

Final Year Project Report

Project Name Hawk Eye



Project Advisor:

Dr. Muhammad Umar Suleman

Submitted by:

Mohammad Abubakar Iftikhar

Wasim Ijaz

Muhammad Abdullah

Saad Atif

Session

2019-2023

University of Management and Technology

C-II Johar Town Lahore Pakistan

Final Approval

- **Head of Department** Dr. Atif Alvi
Department of Informatics & Systems
School of Systems & Technology
UMT Lahore

- **Director (Final Year Projects-CS)** Dr. Jameel Ahmad
Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Supervisor** Dr. Tahir Mushtaq
Department of Informatics & Systems.
School of Systems & Technology
UMT Lahore

- **Co-Supervisor** Dr. Umar Suliman

Project Title	Hawk Eye
Objective	To identify and track the humans
Undertaken by	Mohammad Abubakar Iftikhar Wasim Ijaz Mohammad Abdullah Saad Atif
Supervised by	Dr. Muhammad Umar Suleman
Starting Date	15-09-2022
Completion Date	On Going
Tools Used	C++, React, NodeJs
Operating System	Windows

Acknowledgment

I would like to express my sincerest gratitude to Dr. Umar Suleman and Dr. Tahir Mushtaq for their invaluable guidance, support, and encouragement throughout the duration of this project. His expertise and patience have been instrumental in the successful completion of this project.

I would also like to thank Dr. Jameel for their contributions to this project.

Finally, I would like to acknowledge the support and understanding of my family and friends, who have been a constant source of motivation throughout this journey.

Thank you all for your support.

Abstract

The goal of this project is to develop a system that can detect, identify, and track a person or a vehicle using non-overlapping cameras. The system utilizes computer vision techniques to detect and identify the target object in each camera's field of view. Once the target is detected and identified, the system tracks the target as it moves between cameras by analyzing the motion patterns and identifying unique features of the target. The system then uses this information to seamlessly link the tracks across multiple non-overlapping cameras, providing a comprehensive view of the target's movement. The system is tested on various scenarios with different camera angles and lighting conditions, and the results show that it can accurately detect, identify, and track a person or a vehicle with high precision and recall. The system can be used in various applications, such as surveillance, traffic monitoring, and crowd management.

Table of Contents:

Contents

Acknowledgment	4
Abstract	5
Table of Contents:	6
Definitions and Acronyms.....	8
List of Figures	9
List of Tables	10
1. INTRODUCTION	11
1.1. Motivations:	11
1.2. Project Overview	11
1.3. Problem Statement.	12
1.4. Objectives	12
2. DOMAIN ANALYSIS	13
2.1. Customer	13
2.2. Stakeholders	13
2.3. Affected Groups with social or economic impact	13
2.4. Dependencies/ External Systems	13
2.4.1. Feature Comparison	13
3. REQUIREMENTS ANALYSIS	15
3.1. Requirements	15
3.2. List of Actors	15
3.3. List of use cases	16
3.4. System use case diagram	16
3.5. Extended use cases	17
3.5.1 Login	17
3.5.2 Add A User	18
3.5.3 View Live Camera Footage	19
3.5.4 View Suspicious Humans	19
3.5.5 View Human Path	20
3.6. User interfaces (mock screens)	21
4. DATA FLOW DIAGRAM.....	23

4.1.	Data Flow Diagram Level 0	23
4.2.	Data Flow Diagram Level 1	23
5.	SYSTEM DESIGN.....	24
5.1.	System Architecture Diagram	24
5.2.	Class Diagram	25
5.3.	Sequence Diagrams	26
5.4.	Collaboration Diagrams	27
5.5.	ERD	28
6.	CONCLUSION:.....	29
7.	BIBLIOGRAPHY:.....	30

Definitions and Acronyms

Acronym	Definition
Traffic	All people
ESP32 Cam	A small-sized, low-cost camera module that is compatible with the ESP32 microcontroller
React Application	A JavaScript library for building user interfaces
DFD	Data Flow Diagram
UML	Unified Modeling Language
API	Application Programming Interface
GUI	Graphical User Interface
DB	Database

Table 1- table of Acronyms and Definitions

List of Figures

Figure 1- System Use case	14
Figure 2- Screen 1 (For Mobile)	16
Figure 3- Screen 1 (For Desktop)	16
Figure 4- DFD 0	17
Figure 5- DFD 1	18
Figure 6- Class Diagram	20
Figure 8- Sequence Diagram	21
Figure 9- Collaboration Diagram	22
Figure 10- ERD	24

List of Tables

Table 1- table of Acronyms and Definitions	7
Table 2- Stake Holders	10
Table 3- Feature comparison	11
Table 4- Requirements	12
Table 5- Extended Use cases	15

1. INTRODUCTION

Nowadays, the offensive activities have been increasing and it's difficult to track and identify them. CCTVs are everywhere for monitoring people. Every person is captured by a camera many times a day. It requires a lot of resources and manpower to track each frame of video. Hence, we are creating an automated system with accuracy. Moreover, it will show the frames and the part of it which contains the unusual activity. This System is defined as a real-time surveillance program designed to automatically detect and account for the signs of offensive or disruptive activities immediately. This work plans to use different Deep Learning models to detect and classify levels of high movement in the frame. In this work, videos are categorized into segments. This system can be called an "Intelligent surveillance system". Detection, classification, tracking, identification and path drawing are the main steps of this system. A web application will be developed along with the system to add cameras and analyze the data. This system will count the number of people and the time consumed by a person on the premises of the institute.

1.1. Motivations:

The motivation of this project is the increasing need for advanced systems that not only detect the person but also identify and track the person and vehicle in real time. In traditional systems, which are generally available in the market, overlapping cameras are being used and it is difficult to track the person in all these cameras. This situation makes it difficult to track the movement of traffic in open and large areas like airports, train stations, institutions and shopping malls.

With increasing crime and terrorist activities, it is necessary to shift to an advanced surveillance system that can monitor traffic activities automatically. This system can also be used in traffic monitoring.

1.2. Project Overview

Detection and tracking of humans is useful in many aspects like gender classification, abnormal behavior detection, person identification, counting of individuals, etc. Different methods are already used with different algorithms. We are going to implement a new algorithm. Our goal is to detect a person and then track him in videos. Firstly, we are converting the input video into frames. Then use different techniques to remove noise in images and do foreground and background segmentation, subtracting the background and detecting motion.

In large crowded areas like campuses, due to the crowd, it is difficult to identify and track people. As much of the processing is done by the cameras, the system will be scalable by reducing the load of the server. The goal of this project is to provide cameras with software

which will be an advanced form of IP camera. The end product will be used by any institute or organization.

1.3. Problem Statement.

The problem that this project is going to solve is the limitation of traditional surveillance systems, using overlapping cameras, and limited coverage.

The available systems depend upon manual monitoring, which has drawbacks like time consumption, labour costs, and human error. These limitations make the systems unreliable. The lighting conditions also make the system unreliable.

1.4. Objectives

- Track and display people with path and suspicious behaviour.
- Display Live Video Feed
- Detect the unique persons and keep an image of it
- Send notifications to users depending upon the access type
- Scalable and secure system

2. DOMAIN ANALYSIS

2.1. Customer

We don't have any customers.

2.2. Stakeholders

Stakeholder	Role in System
Project team	They are responsible for project design and development.
Project manager	He is responsible for leading the project team.
End User	He will be using the system on its premises.
Business Analyst	He is responsible for analyzing business opportunities.
Data Scientist	He is responsible for analyzing the data
Engineers	He is responsible for making the project feasible.

Table 2- Stake Holders

2.3. Affected Groups with social or economic impact

- Security staff:
 - They can detect the anomalies easily.
- Surveillance staff
 - They don't need to continue watching recordings for hours.
- Administration staff
 - They can easily identify the path of a suspected person

2.4. Dependencies/ External Systems

Our system depends on:

- ESP-32 cam

2.4.1. Feature Comparison

Data source	Purpose	Image or video clips	Annotation	Environment	Year
MIT	Pedestrian segmentation, detection and tracking	709 pedestrian images 509 training and 200 test images	No annotation	Daylight scenario	2000, 2005
Caltech pedestrian	Detection and tracking of	250,000 frames	2300 pedestrians were annotated	Urban environment	2012

dataset	pedestrian walking on the street				
GM-ATCI	Rear view pedestrian segmentation, detection and tracking	250 video sequences	200K annotated	Both day and night scenarios,	2015
Daimler	Detection and tracking of pedestrian	15,560 pedestrian	2D bounding box overlap criterion	Urban environment	2016
NICTA 2016	Segmentation, pose estimation, learning of pedestrian	25,551 unique pedestrians	2D ground truth image	Urban environment	2016
MS COCO 2018	Object detection, segmentation, keypoint detection, and DensePose detection	2 million instances	5 captions per image	Urban environment	2018
Mapillary vistas dataset 2017	Semantic understanding street scenes	25,000 images	Pixel-accurate and instance-specific human annotations	Urban environment	2017
MS COCO 2017	Recognition, segmentation, captioning	328,124 images	Segmented people and objects	Urban environment	2017
MS COCO 2015	Recognition, segmentation, captioning	328,124 images	Segmented people and objects	Urban environment	2015
ETH	Segmentation, detection, tracking	Videos	Dataset consist of other traffic agents such as different cars and pedestrians	Urban environment	2010
TUD-Brussels	Detection, tracking	1092 image pairs	1776 annotated pedestrian	Urban environment	2009
INRIA	Detection, segmentation	498 images	Annotations are marked manually	Urban environment	2009
CVC-ADAS	Detection, tracking	60,000 frames	7900 annotated pedestrians	Urban environment	2005
PASCAL VOC 2012	Detection, classification, segmentation	11,530 images, 20 object classes	27,450 ROI annotated	Urban environment	2012

Table 3- Feature comparison

3. REQUIREMENTS ANALYSIS

3.1. Requirements

<i>External interface requirements</i>	Our system can be used with existing surveillance systems using APIs provided by us.
<i>Functional Requirements</i>	Security: The system should be secure. Accuracy: The system should be accurate. Compliance: The system should compile with the rules. Interoperability: The system should be interoperable.
<i>Non-Functional Requirements</i>	Performance: The system should perform well. Response: Time: The response time of the system should be minimum. Capacity: The system should be able to store some data. Interface: The interface of the system should be interactive. Usability: The system should be usable on any hardware. Maintainability: The system should be maintained. Backup: The system should have a backup of at least 3 days. Error: The error of the system should be less than 1%. Reliability: Available 27 hours a day. Portability: The system should be portable. Supportability: The system should be supportable. Adaptability: The system should be adaptable Dependency: The system should not be dependent. Efficiency: The system should be efficient. Reporting: The system should provide reports with time stamps. Scalability: The system should be scalable.

Table 4- Requirements

3.2. List of Actors

ESP32-CAM: This is the camera module that captures the video feed and sends it to the Arduino.

System: This is the computer that receives the video feed from the ESP32-CAM, processes it to detect and track the target object, and sends the tracking information and live video feed to the React Application.

React Application: This is the web-based application that receives the tracking information and live video feed from the Arduino, displays them on the screen, and allows the user to access stored data, set parameters, and receive alarms/notifications.

Database: This is the storage system that stores the tracking information and live video feed from the React Application.

User: This is the person who interacts with the React Application to access the stored data, set parameters, and receive alarms/notifications.

3.3. List of use cases

The use cases of People and vehicle detection/identification and tracking have a wide range.

It includes:

- Character Recognition:

It can detect characters on moving vehicles.

- Access control:

It can be used on Airports, offices, Universities etc.

- Security and Surveillance:

It can be used in commercial, and public settings to detect abnormalities.

- Warehouses and factories:

It can be used to safeguard factories.

- Smart Security systems:

It can be used for smart homes as well as smart cities.

- Self-Driving cars:

It can detect when to perform several actions.

3.4. System use case diagram

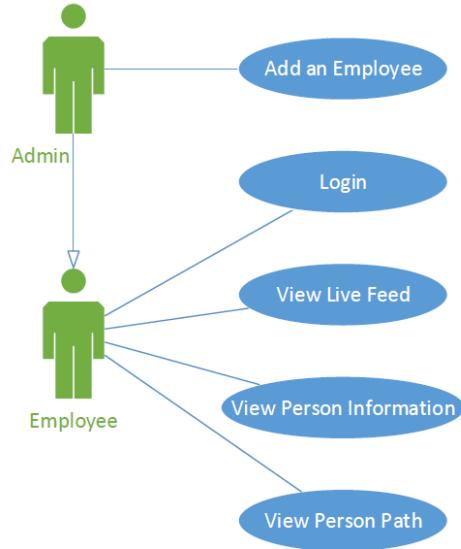


Figure 1- System Use case

3.5. Extended use cases

3.5.1 Login

Use Case ID:	UC-1		
Use Case Name:	View live camera footage		
Created By:	Muhammad Abdullah	Last Updated By:	
Date Created:	15/12/22	Last Revision Date:	
Actors:	User		
Description:	Gives the ability for the user to login to the system.		
Trigger:	An existing user wants to access the system.		
Preconditions:	1. The user should be connected to the internet		
Post conditions:	1. The user is able to login and view the welcome screen.		
Normal Flow:	1. User opens the web application 2. A Welcome screen is shown 3. User opens the login page using login button 4. User enters in their id in the id text box 5. User enters in their password in the password text box 6. User clicks on login 7. The system validates the user id and password 8. System shows welcome screen with options		

Alternative Flows:	
Exceptions:	6a. In step 6 of the normal flow, if the customer enters invalid credentials 1. Error is shown 2. Goes back to step 3 of normal flow
Includes:	
Frequency of Use:	70 times a day
Special Requirements:	The server should be properly functional.
Assumptions:	The user already has login credentials The user is trained to use the system.
Notes and Issues:	

3.5.2 Add A User

Use Case ID:	UC-2		
Use Case Name:	Add a User		
Created By:	Muhammad Abdullah	Last Updated By:	
Date Created:	15/12/22	Last Revision Date:	
Actors:	Admin, User		
Description:	Gives the admin the ability to add a user.		
Trigger:	A new user wants to access the system.		
Preconditions:	1. The admin must be logged in. 2. Admin must be on welcome screen		
Post conditions:	2. A new user has been added.		
Normal Flow:	1. Admin clicks on add user button. 2. Admin enters the new user's name, id and password. 3. Admin clicks on confirm button. 4. New user is added.		
Alternative Flows:			
Exceptions:	4a. In step 4 of the normal flow, if the customer enters invalid credentials 3. Error is shown 4. Goes back to step 3 of normal flow		
Includes:	Normal use cases 1 to 3 and exception 1 to 2 are used whenever the user needs to interact with the system for the first time. And Normal use case 4 is shown whenever a user exits a particular step in any use case functionality.		
Frequency of Use:	70 times a day		

Special Requirements:	The server should be properly functional.
Assumptions:	The user already has login credentials The user is trained to use the system.
Notes and Issues:	

3.5.3 View Live Camera Footage

Use Case ID:	UC-3		
Use Case Name:	View live camera footage		
Created By:	Muhammad Abdullah	Last Updated By:	
Date Created:	15/12/22	Last Revision Date:	
Actors:	User		
Description:	Gives the ability to view live video feed of a particular camera.		
Trigger:			
Preconditions:	<ol style="list-style-type: none"> 1. The user must be logged in. 2. There should be active working cameras 		
Post conditions:	<ol style="list-style-type: none"> 3. The user is able to view live feed of a camera 		
Normal Flow:	<ol style="list-style-type: none"> 1. User selects view camera footage 2. User selects a particular camera 3. Video footage of that camera is shown. 		
Alternative Flows:			
Exceptions:			
Includes:			
Frequency of Use:	70 times a day		
Special Requirements:	The server should be properly functional.		
Assumptions:	The user already has login credentials The user is trained to use the system.		
Notes and Issues:			

3.5.4 View Suspicious Humans

Use Case ID:	UC-4
Use Case Name:	View suspicious humans.
Created By:	Muhammad Abdullah

Date Created:	15/12/22	Last Revision Date:	
Actors:	User		
Description:	Gives the ability to view list of suspicious humans caught by the system		
Trigger:			
Preconditions:	1. The user must be logged in.		
Post conditions:	4. The user is able to view list of suspicious humans		
Normal Flow:	1. Click on suspicious humans from the navigation bar.		
Alternative Flows:			
Exceptions:			
Includes:			
Frequency of Use:	70 times a day		
Special Requirements:	The server should be properly functional.		
Assumptions:	The user already has login credentials The user is trained to use the system.		
Notes and Issues:			

3.5.5 View Human Path

Use Case ID:	UC-4		
Use Case Name:	View path taken by human		
Created By:	Muhammad Abdullah	Last Updated By:	
Date Created:	15/12/22	Last Revision Date:	
Actors:	User		
Description:	Gives the ability to view path of a particular human in the cameras as well as the entry and exit points.		
Trigger:	After the third or fourth use case, the user might want to view the path taken by a human.		
Preconditions:	1. The user should have completed UC-3 or UC-4		
Post conditions:	1. The personal is able to see the path a particular human has taken.		

Normal Flow:	<ol style="list-style-type: none"> 1. User click on path taken dropdown box. 2. User selects the id of the human from drop box. 3. User clicks on path taken button. 4. The system displays the path taken by the particular human and a picture of the human.
Alternative Flows:	
Exceptions:	
Includes:	
Frequency of Use:	70 times a day
Special Requirements:	The server should be properly functional.
Assumptions:	<p>The user already has login credentials</p> <p>The user is trained to use the system.</p>
Notes and Issues:	

3.6. User interfaces (mock screens)

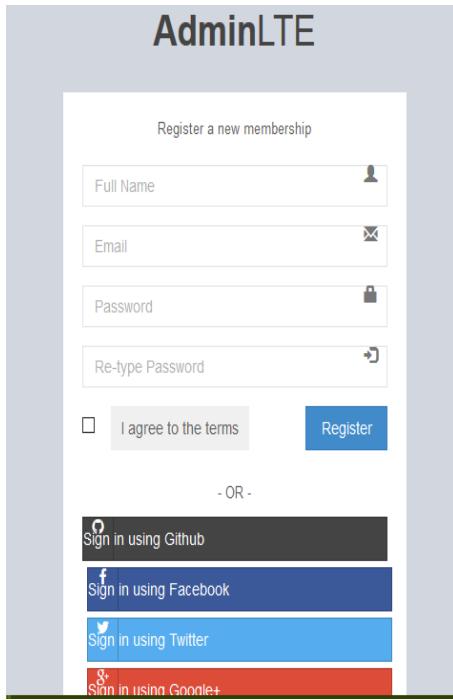


Figure 2- Screen 1 (Mock Screen For Mobile)



Figure 3- Screen 1 (Mock Screen For Desktop)

4. DATA FLOW DIAGRAM

4.1. Data Flow Diagram Level 0

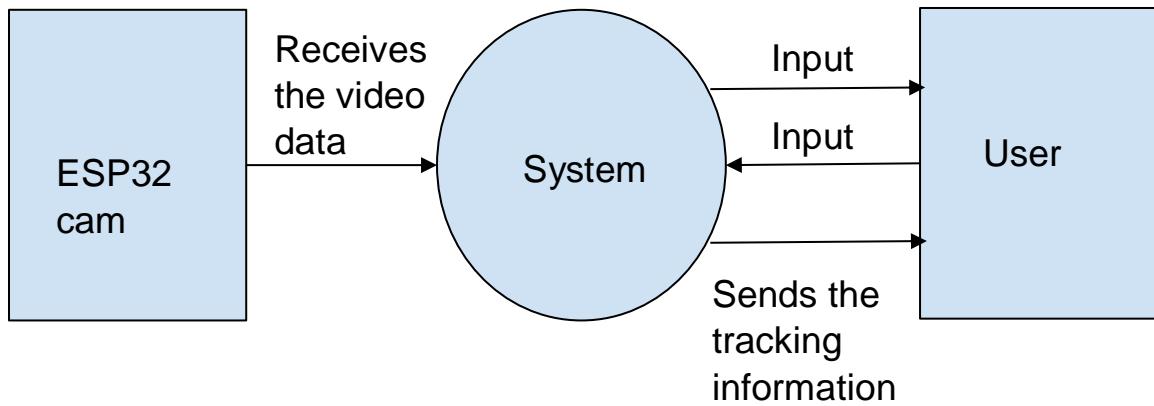


Figure 4- DFD 0

4.2. Data Flow Diagram Level 1

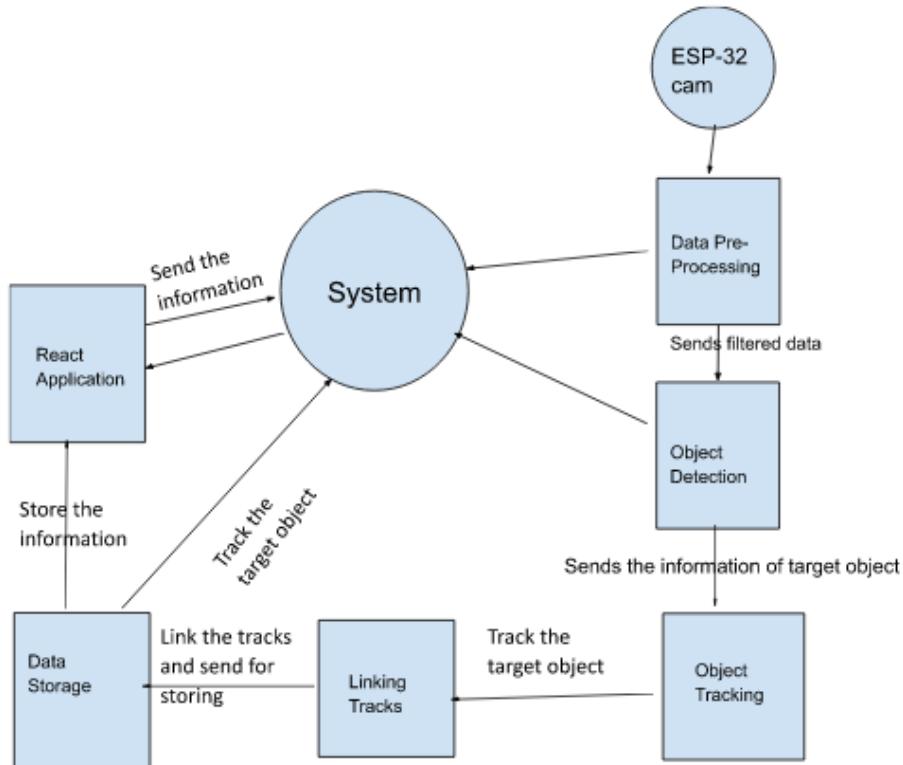


Figure 5- DFD 1



5. SYSTEM DESIGN

5.1. System Architecture Diagram

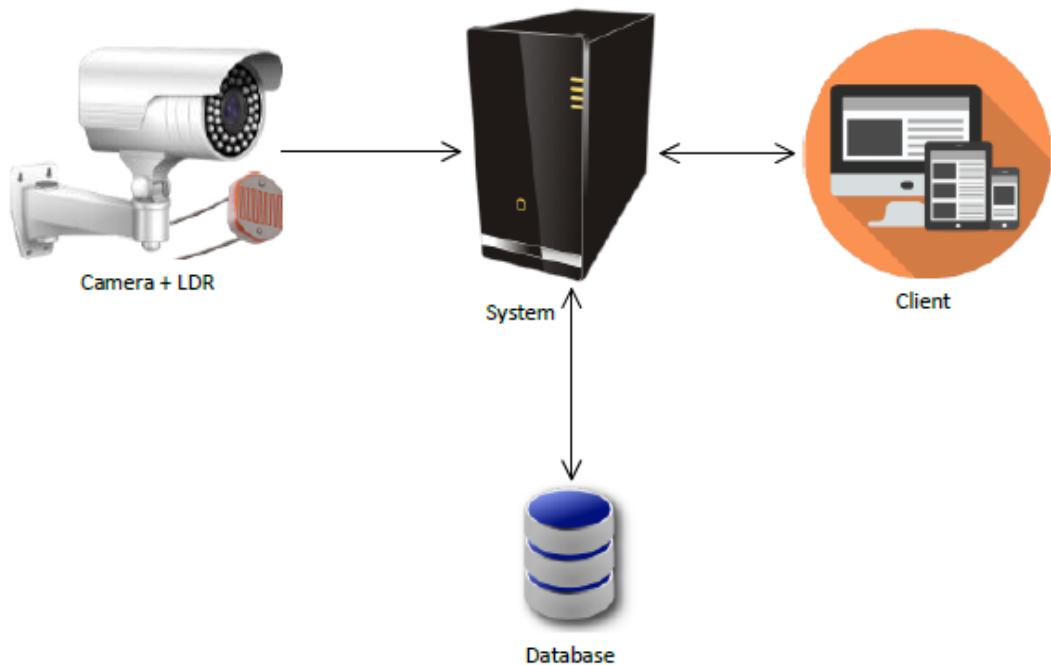


Figure 6- System Architecture Diagram

5.2. Class Diagram

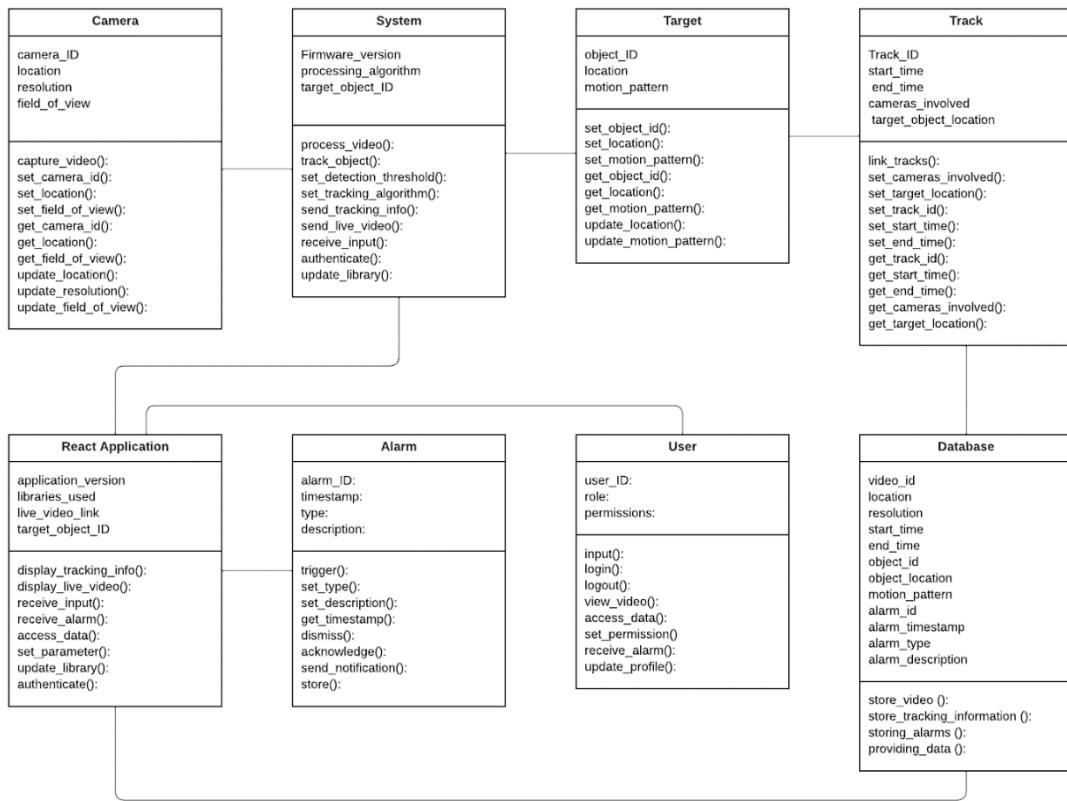


Figure 7- Class Diagram

5.3. Sequence Diagrams

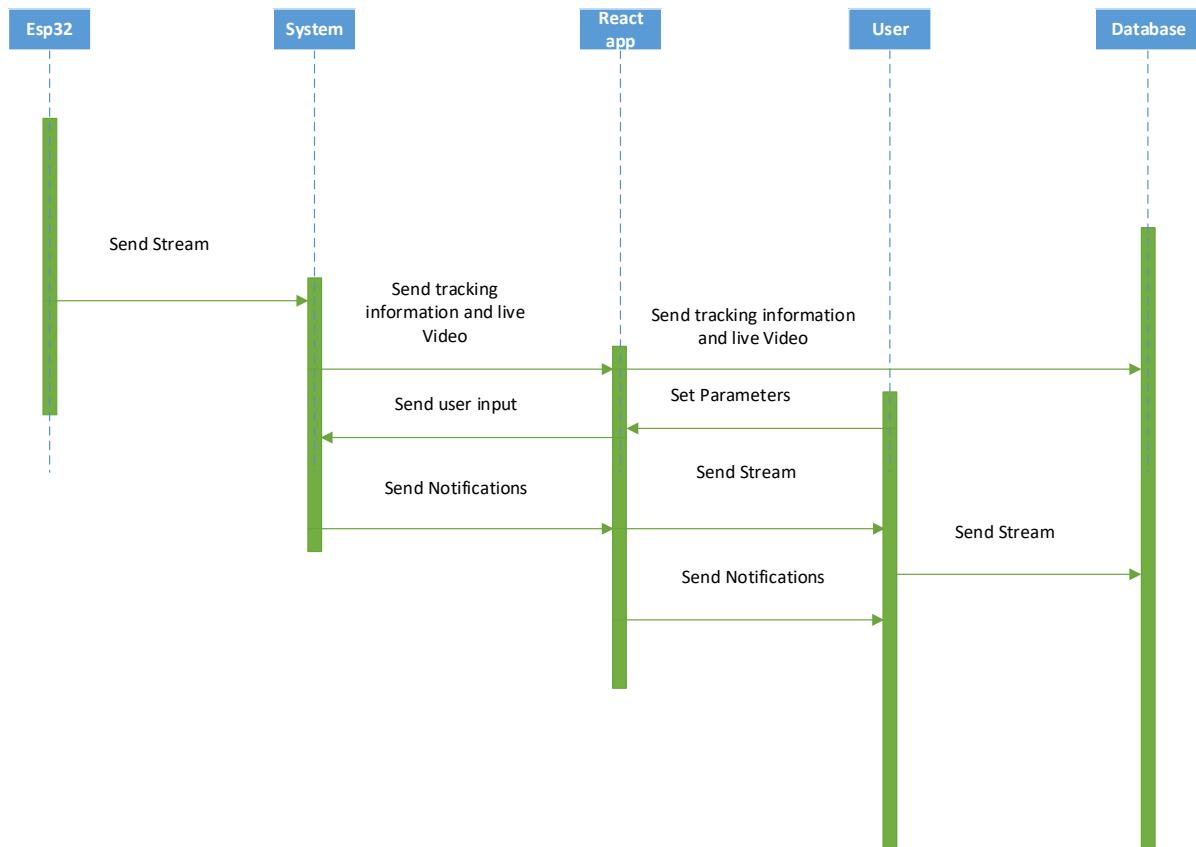


Figure 8- Sequence Diagram

5.4. Collaboration Diagrams

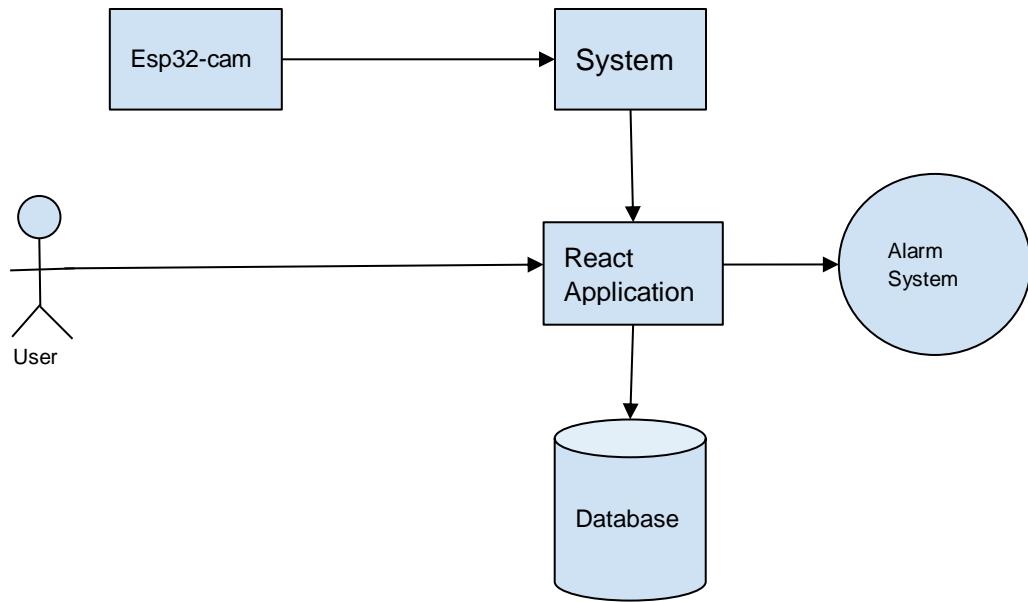


Figure 9- Collaboration Diagram

5.5. ERD

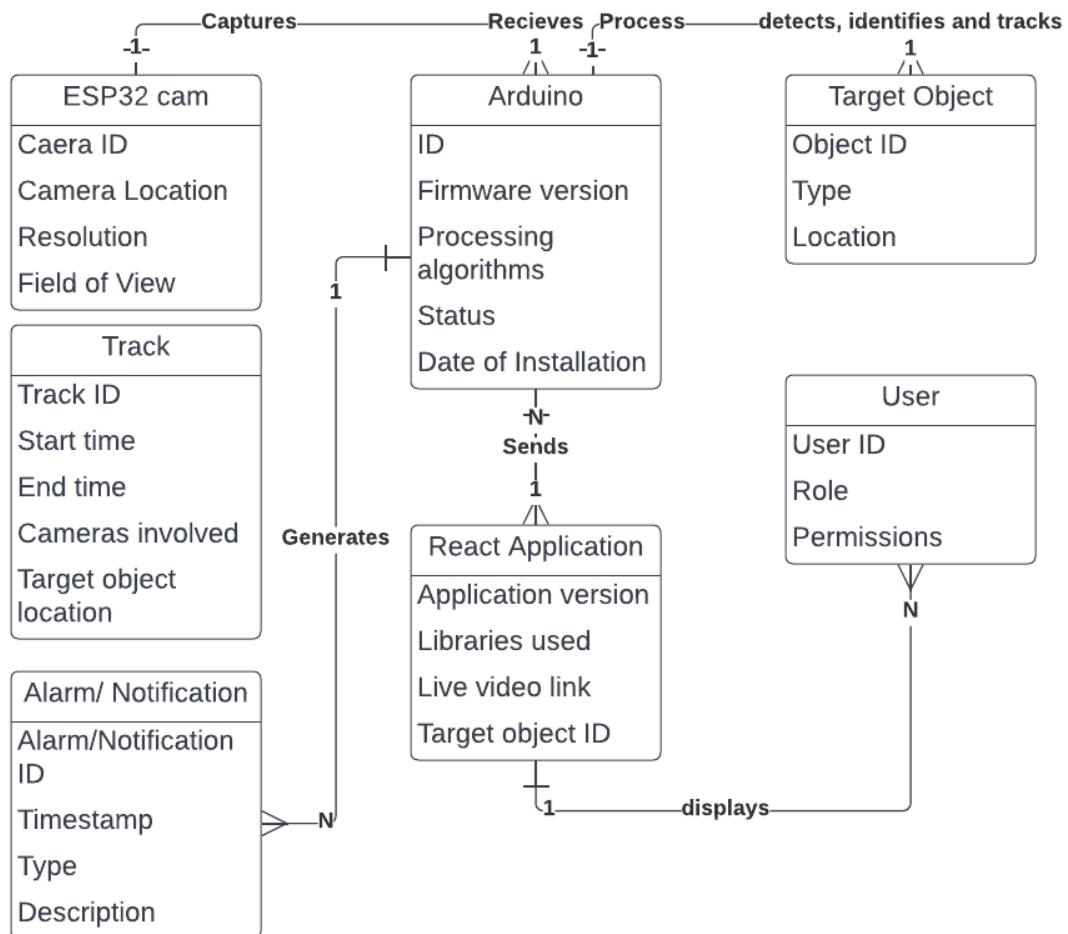


Figure 10- ERD

6. CONCLUSION:

This system is about identifying data points in data that don't fit the normal patterns. It can be useful to solve many problems including fraud detection, medical diagnosis, etc. Machine learning methods allow to automate anomaly detection and make it more effective, especially when large datasets are involved.

7. BIBLIOGRAPHY:

Final Year Project Report
AgriMart: Online Crops Auction Platform



Project Advisor: Miss Fasiha Ashraf

Submitted By:
Abu Bakar S2020266019
Malaika Aziz S2020266044

Session
S2023
University of Management and Technology
C-II Johar Town Lahore Pakistan

Dedication

We dedicate this project to all the hardworking farmers around the world who tirelessly cultivate the land, nourish communities, and provide us with the sustenance we need. Your unwavering commitment to agriculture and the crops you grow inspire us to create AgriMart, an online crops auction platform, to support and empower you. This project is a testament to our dedication to revolutionizing the agricultural industry, fostering fair trade, and connecting farmers with buyers in a transparent and efficient manner. With utmost respect and admiration, we embark on this journey to make a positive impact on the lives of farmers and contribute to the growth of the agricultural sector.

Final Approval

- **Head of Department**

Department of Informatics & Systems
School of Systems & Technology
UMT Lahore

- **Director (Final Year Projects-CS)**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Supervisor**

Department of Informatics & Systems.
School of Systems & Technology
UMT Lahore

- **Co-Supervisor**

Acknowledgment

We would like to express our sincere gratitude to all those who have contributed to the development of AgriMart, an online crops auction platform. We would also like to thank our group member, advisor and mentors for their invaluable guidance and support throughout the project. Furthermore, we are grateful to the farmers and buyers who provided us with valuable insights and feedback to shape AgriMart into a platform that meets their needs. Lastly, we acknowledge the support and encouragement from our friends, family, and colleagues, whose unwavering belief in our vision motivated us to overcome challenges and bring this project to fruition. Without the collective efforts and support of all these individuals, this would not have been possible.

Project Title: AgriMart: Online Crops Auction Platform

Objective: Develop a crops auction platform for efficient and transparent transactions.

Undertaken by: Abu Bakar-S2020266019, Malaika Aziz-S2020266044

Supervised by: Miss. Fasiha Ashraf

Starting Date : 10- April-2023

Completion Date:

Tools Used: MS office, visual paradigm, mermaid, Figma, Vscode, Android Studio, Firebase, Flutter (Framework)

Operating System: Windows 10, macOs, android

Documentation: MS Word

Plagiarism Report

Abstract

AgriMart addresses the huge problem of lack of transparency and fair deals in Agricultural market. The traditional agricultural marketplace faces significant challenges, including limited access to information, lack of transparency, and inefficient transaction processes. To tackle these issues, our project, It introduces a comprehensive crops auction platform. It aims to empower farmers by enabling them to access broader markets and connect with reliable buyers, thus enhancing their economic prospects. Simultaneously, it provides buyers with a platform to source high-quality produce at fair prices. To achieve these objectives, AgriMart implements transparent and standardized auction processes, real-time communication tools, and secure payment and delivery systems. Farmers can list their crops, request quality inspections, and engage in auctions, while buyers can bid on crops, communicate directly with farmers, and track deliveries. Administrators oversee the platform, ensuring fair processes and providing support. The methodology employed involves user-friendly interfaces for registration and login, efficient crop listing with images, and quality inspection requests. Secure online payment systems, real-time chat functionalities, and SMS/email notifications enhance the user experience. The platform also offers predictive analytics, market insights, and a rating system to foster transparency and collaboration.

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
Draft	Abu Bakar	Initial draft created for distribution and review comments	10-April-2023
Preliminary	Malaika Aziz	Second draft incorporating initial review comments, distributed for final review	20-April-2023
Final	Malaika Aziz	First complete draft, which is placed under change control	10-May-2023
Revision 1	Abu Bakar	Revised draft, revised according to the change control process and maintained under change control	28-May-2023

CONTENTS

CONTENTS	I
DEFINITIONS AND ACRONYMS	II
LIST OF FIGURES	III
LIST OF TABLES	IV
1. INTRODUCTION	1
1.1 MOTIVATIONS	1
1.2 PROJECT OVERVIEW	1
1.3 PROBLEM STATEMENT	1
OBJECTIVES	2
2. DOMAIN ANALYSIS.....	3
2.1 CUSTOMER.....	3
2.2 STAKEHOLDERS	3
2.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT	3
DEPENDENCIES/ EXTERNAL SYSTEMS.....	4
2.4 REFERENCE DOCUMENTS.....	4
2.4.1 <i>Related Projects</i>	4
2.4.2 <i>Feature Comparison</i>	4
3. REQUIREMENTS ANALYSIS	6
3.1 REQUIREMENTS	6
3.2 LIST OF ACTORS.....	8
3.3 LIST OF USE CASES	8
3.4 SYSTEM USE CASE DIAGRAM.....	10
3.5 EXTENDED USE CASES.....	11
3.6 USER INTERFACES (MOCK SCREENS)	19
4. DATA FLOW DIAGRAM (OPTIONAL).....	21
4.1 DATA FLOW DIAGRAM LEVEL 0	21
4.2 DATA FLOW DIAGRAM LEVEL 1	22
5. SYSTEM DESIGN	23
5.1 SYSTEM ARCHITECTURE DIAGRAM	23
5.2 CLASS DIAGRAM.....	24
5.3 SEQUENCE DIAGRAMS	25
5.4 ERD.....	32
5.5 DATA DICTIONARY	33

Definitions and Acronyms

Table 0-1: Acronyms and Definitions

Acronym	Definition
UMT	University of Management and Technology
AgriMart	Auction platform related Crops
DFD	Data Flow Diagram
UC	Use case

List of Figures

Figure 3-1: AgriMart Use Case Diagram	10
Figure 3-2: Sign Up screen prototype	19
Figure 3-3: Login Screen prototype	19
Figure 3-4: Crops View Screen Prototype.....	19
Figure 3-5: Crop Detail Screen prototype	19
Figure 3-6: Available Crops Page	20
Figure 3-7: Crops Detail Page	20
Figure 3-8: Farmer's Bid Descision Screen	20
Figure 3-9: Buyer Proceed to Payment Screen.....	20
Figure 4-1: Data Flow Diagram Level 0	21
Figure 4-2: Data Flow Diagram	22
Figure 5-1: System Artitecture Diagram of Agrimart	23
Figure 5-2: Class Diagram of AgriMart	24
Figure 5-3: Buyer Register Sequence Diagram	25
Figure 5-4:Login Sequence Diagram	26
Figure 5-5: Add Crop Sequence Diagram	26
Figure 5-6: Browse Crops Sequence Diagram	27
Figure 5-7: Accept / Reject bid by Farmer.....	27
Figure 5-8: Bid on Crop	28
Figure 5-9: Initialize Chat	28
Figure 5-10: Delivery Tracking.....	29
Figure 5-11: Request Quality Inspection Sequence Diagram.....	29
Figure 5-12: Bid on Crops Sequence Diagram.....	30
Figure 5-13: Make Payment Sequence Diagram	31
Figure 5-14: Entity Relationship Diagram	32

List of Tables

Table 2-1: List of Stakeholders	3
Table 2-2: Feature Comparison	5
Table 3-1: Functional Requirements Table	6
Table 3-2: Non-Functional Requirements	7
Table 3-3: Extended use case Register	11
Table 3-4: Extended use case Login	12
Table 3-5: Extended use case List Crops	13
Table 3-6: Extended use case Browse Crops	13
Table 3-7: Extended use case Bid Crop	14
Table 3-8: Extended Automated Bid Acceptance	14
Table 3-9: Extended use case Real-time Chat	15
Table 3-10: Extended Use case SMS/Email notification	16
Table 3-11: Extended Make Payment	16
Table 3-12: Extended Delivery Tracking	17
Table 3-13: Extended Quality Inspection	18
Table 5-1: Data Dictionary of Agrimart	33

1. INTRODUCTION

1.1 Motivations

The motivation behind choosing this project stems from the challenges faced by farmers and buyers in the traditional agricultural marketplace. Limited access to information, lack of transparency, and inefficient transaction processes have been persistent issues. This project aims to provide a solution that empowers farmers to reach a wider market and enables buyers to connect with reliable sources of agricultural produce. By leveraging technology, we aim to create a platform that streamlines the auction process, facilitates real-time communication, provides market insights, and ensures fair and transparent transactions for all participants.

1.2 Project Overview

This project aims to develop a comprehensive crops auction platform that connects farmers, buyers, and administrators. The platform will enable farmers to list their crops for auction, receive bids from buyers, communicate in real-time, and manage payments and delivery. Buyers will have access to a wide range of crop listings, participate in auctions, and interact directly with farmers. Administrators will oversee the platform, ensure fair auction processes, resolve disputes, and provide analytics and insights. The system will facilitate efficient and transparent transactions, promoting trust and collaboration in the agricultural ecosystem.

1.3 Problem Statement

The current agricultural marketplace faces several challenges that hinder farmers and buyers alike. Lack of transparency, limited access to reliable buyers, and inefficient communication channels create barriers to successful transactions. Farmers struggle to reach wider markets and find trustworthy buyers, while buyers face difficulties in discovering high-quality crops at fair prices. The existing processes are complex and often result in suboptimal outcomes, leading to inefficiencies and missed opportunities for both parties.

To address these issues, our project aims to develop a user-friendly crops auction platform. This platform will revolutionize the way farmers and buyers connect and transact, providing a transparent and efficient marketplace. By leveraging technology and innovation, our solution will create an inclusive platform that bridges the gap between farmers and buyers, fostering a seamless and mutually beneficial trading experience.

One of the key problems in the current agricultural marketplace is the lack of transparency. Farmers often struggle to find reliable buyers who can offer fair prices for their crops, while buyers face challenges in assessing the quality and authenticity of the produce. Our platform will address this problem by implementing transparent and standardized auction processes. Farmers will have the opportunity to list their crops with detailed information, including crop type, quantity, quality, and location. Buyers, on the other hand, will have access to comprehensive crop listings, enabling them to make informed bidding decisions based on their specific requirements.

Another challenge is the limited access to reliable buyers. Many farmers, especially small-scale and rural farmers, find it challenging to reach wider markets and connect with potential buyers who can offer fair prices. Our platform will provide a centralized marketplace where farmers can showcase their crops to a large pool of registered buyers. This expanded reach will enable farmers to maximize their market opportunities and improve their profitability. Buyers, on the other hand, will benefit from a diverse range of crop listings, allowing them to source high-quality produce directly from farmers.

Inefficient communication channels also pose a significant problem in the current agricultural marketplace. Traditional methods of communication, such as phone calls and physical meetings, can be time-consuming and inefficient. Our platform will address this challenge by incorporating real-time chat functionality, enabling direct communication between farmers and buyers. This streamlined communication process will facilitate negotiations, clarify any concerns or queries, and establish trust between the parties involved.

Furthermore, our platform will focus on secure transactions and reliable information exchange. Payment processing will be handled securely within the platform, ensuring that farmers receive their rightful earnings and buyers' payments are held in escrow until the crop is delivered. Additionally, the platform will provide a rating and review system, allowing farmers and buyers to provide feedback and assess each other's performance. This system will enhance transparency, build trust, and encourage accountability among all participants.

By addressing these challenges and providing an intuitive, efficient, and secure marketplace, our project seeks to create a positive impact on the agricultural industry. Our solution aims to empower farmers by expanding their market reach, enabling them to receive fair prices for their crops, and improving their livelihoods. Likewise, buyers will benefit from access to a diverse range of quality crops, simplifying their sourcing processes and promoting sustainable trade practices. Through our platform, we envision a thriving agricultural marketplace that fosters collaboration, transparency, and mutual success for all participants.

Objectives

The objectives of this project are as follows:

- Enable farmers to list their crops, receive competitive bids, and communicate with buyers in real-time.
- Empower buyers to access a diverse range of crop listings, participate in auctions, and interact directly with farmers.
- Provide administrators with tools to oversee the auction process, resolve disputes, and analyze platform performance.
- Facilitate secure and efficient payment and delivery processes for successful transactions.
- Foster trust, transparency, and collaboration in the agricultural marketplace through rating and review systems.
- Provide predictive analytics and market insights to farmers and buyers for informed decision-making.
- Enhance the efficiency and accessibility of the agricultural marketplace, benefiting both farmers and buyers alike.
- Develop a user-friendly crops auction platform that meets the needs of farmers, buyers, and administrators.

2. DOMAIN ANALYSIS

2.1 Customer

Our target customers are individuals and organizations involved in the agricultural industry. This includes farmers, agricultural cooperatives, wholesalers, retailers, and buyers such as restaurants, grocery stores, and food processors. Our platform aims to serve both medium-scale and large-scale agricultural stakeholders seeking efficient and transparent methods of buying and selling crops.

2.2 Stakeholders

Table 2-1: List of Stakeholders

Stakeholder	Role in System
Farmers	They are the primary sellers on the platform, responsible for listing their crops, managing auctions, and delivering the sold produce.
Buyers	They are the primary purchasers on the platform, participating in auctions, placing bids, and coordinating the payment and delivery of crops.
Administrator	They oversee the operations of the platform, ensuring fair auction processes, resolving disputes, and providing support to farmers and buyers.
Payment Gateway Providers	External payment systems that facilitate secure online transactions between buyers and farmers.
Logistic Providers	External logistics services that handle the transportation and delivery of crops from farmers to buyers.

Table 2-1 contains all stakeholders and their role in platform.

2.3 Affected Groups with social or economic impact

Farmers:

Our platform empowers farmers by providing them with a broader market reach, enabling them to sell their crops at competitive prices, improve their livelihoods, and enhance their economic stability.

Buyers:

Buyers benefit from accessing a wider range of crops, competitive pricing, and a transparent marketplace, ensuring the availability of high-quality produce for their businesses.

Local Communities:

By promoting fair trade and supporting local farmers, our platform contributes to the economic development and sustainability of local communities.

Environment:

A more efficient marketplace reduces waste and supports sustainable agricultural practices, benefiting the environment in the long run.

Dependencies/ External Systems

- **Payment Gateways:** We depend on external payment gateway providers to facilitate secure online transactions between buyers and farmers, ensuring reliable and timely payments.
- **Delivery Service Providers:** We may rely on external delivery service providers to handle the transportation and delivery of crops from farmers to buyers, ensuring efficient and reliable logistics.

2.4 Reference Documents

2.4.1 Related Projects

- MandiLinks
- FarmGhar
- Agrofy
- Amazon
- eBay

2.4.2 Feature Comparison

The feature comparison between the different marketplaces are summarized as follows:

MandiLinks:

- More related to crops compared to Amazon and eBay.
- However, it does not have a bid system for crops, limiting the competitive auction aspect.
- It lacks specialized features for crop supply chain management, such as tracking the delivery process and facilitating secure payments.
- There is no mention of analytics capabilities, which could provide valuable insights for farmers and buyers.

Farm Ghar:

- This platform specifically lists crops, indicating a stronger focus on the agricultural industry.
- However, it seems to have a more limited scope as it only provides listing services without offering further processes like bidding, supply chain management, or analytics.

Agrofy:

- More focused on crops compared to Amazon and eBay.
- Similar to MandiLinks, it does not have a bid system for crops, limiting the competitive auction aspect.
- It also lacks specialized features for crop supply chain management and analytics, which could enhance the overall trading experience.

Amazon:

- Huge marketplace with a wide range of products, but not specifically focused on crops.
- Does not have a dedicated bid system for crops.
- Does not provide specialized features for crop supply chain management or analytics.

eBay:

- Another large marketplace, but not specifically focused on crops.
- Similar to Amazon, it lacks a dedicated bid system for crops.
- Does not offer specialized features for crop supply chain management or analytics.

Table 2-2: Feature Comparison

No.	Features	Amazon	eBay	AgriMp	Mandi Links	Agrofy	Farm Ghar	AgriMart
1	Product listing	✓	✓	✓	✓	✓	✓	✓
2	Bid System	✓	✓	✓	✗	✗	✗	✓
3	Supply chain Management	✓	✓	✓	✗	✓	✗	✓
4	Prediction and Analytics	✗	✗	✗	✗	✗	✗	✓
5	Requests for bulk orders	✗	✗	✓	✓	✗	✗	✓
6	Farm Info and Upcoming Crops	✗	✗	✓	✗	✓	✗	✓
7	Secure online payment	✓	✓	✓	✗	✓	✗	✓

Table 2-2 shows that some big online marketplaces like Amazon and eBay are not very good at helping farmers sell their crops. Others like MandiLinks and Agrofy, while they do focus more on crops, are missing important things like a good way for buyers to compete for the crops, a smooth system for getting crops from the farm to the market, and smart tools for understanding what's happening in the crop market. FarmGhar, even though it's more about listing crops, doesn't do all the extra stuff needed for people to easily and safely buy and sell crops.

This is where AgriMart comes in. It's like a special online market just for crops. AgriMart plans to have everything needed to make selling crops work well. There will be a way for buyers to compete for the crops, a smart system for moving crops around efficiently, tools to help people understand the crop market, and a safe way to pay for things. AgriMart wants to be the best place for crop buying and selling, making it easier and better for farmers, buyers, and everyone in the crop business.

3. REQUIREMENTS ANALYSIS

3.1 Requirements

Table 3-1: Functional Requirements Table

Ref#	Functions	Description	Category	Attribute	Detail and Boundary Constraints
FR1	User Management	User registration, KYC verification, login/logout, password recovery	Evident	Security	User interface should be easy to navigate and understand. Password recovery should be secured with two-factor authentication.
FR2	Listing Crops	Farmer can list their crops with details such as crop type, quantity, quality, location, and starting bid price.	Evident	Performance	Listings should load within 2 seconds. Crop images should be optimized for fast loading.
FR3	Bidding and Auctions	Buyers can view and bid on available crops. The system will automatically select the highest bidder when the auction ends.	Evident	Reliability	The system should be able to handle up to 1000 concurrent bids without crashing. Auction timer should be synchronized with the server clock.
FR4	Crop Quality Inspection	A feature for crop quality inspection before listing and delivery. Farmers can request for inspection service and view the inspection report. Buyers can view the inspection report before bidding.	Hidden	Security Reliability	Inspection reports should be stored securely and only accessible to authorized users. The inspection service provider should be verified and trusted.
FR5	Payment and Delivery	Payment will hold in the platform until the crop is delivered. The system will facilitate the delivery process between the farmer and buyer. Farmers and buyers can track the delivery status.	Evident	Security	Payment and delivery information must be stored securely and protected from unauthorized access
FR6	Analytics and Prediction	The system will analyze the supply and demand of crops. Predict future crop prices and demand. Provide farmers and buyers with insights and trends.	Frill	Scalability	The system should be able to handle a large volume of data without compromising performance. Analytics and prediction algorithms should be updated regularly.

FR7	Rating and Feedback	Farmers and buyers can rate each other after the completion of the transaction. Feedback and ratings will be displayed on the user's profile.	Evident	Usability	The rating and feedback system should be easy to use and understand. Users should be able to filter and sort reviews based on various criteria.
FR8	Real-time Chat	Implement real-time chat feature within the platform for farmers and buyers to communicate with each other.	Evident	Usability	Chat interface should be user-friendly and intuitive. Messages should be delivered in real-time without delays.
FR9	SMS and Email Notification	Send SMS and email notifications to remind users of upcoming auctions, bid status updates, and other important events on the platform.	Evident	Reliability	Notifications should be sent out in a timely manner without delay. Users should be able to control their notification preferences.

Table 3-1 represents all functional requirements in AgriMart platform.

Table 3-2: Non-Functional Requirements

Ref#	Functions	Description	Category	Attribute	Detail and Boundary Constraints
NFR1	Usability	User interface should be easy to navigate and understand.	Evident	Usability	The platform should have a clear and intuitive user interface that enables users to easily navigate and understand the functionalities.
NFR2	Performance	Listings should load within 2 seconds. Crop images should be optimized for fast loading.	Evident	Performance	The platform should have fast loading times for crop listings, and the images should be optimized for quick and efficient loading.
NFR3	Reliability	The system should be able to handle up to 1000 concurrent bids without crashing. Auction timer	Evident	Reliability	The platform should be able to handle a high volume of concurrent bids without any system crashes, and the auction timer

		should be synchronized.			should be synchronized accurately.
NFR4	Security	Inspection reports and payment/delivery information must be stored securely and protected from unauthorized access.	Hidden	Security	The platform should have robust security measures in place to protect the confidentiality and integrity of inspection reports, payment, and delivery information.
NFR6	Scalability	The system should be able to handle a large volume of data without compromising performance.	Frill	Scalability	The platform should be designed to handle a significant amount of data efficiently while maintaining optimal performance levels.
NFR7	Usability	The rating and feedback system should be easy to use and provide options to filter and sort reviews based on various criteria.	Evident	Usability	The rating and feedback system should be user-friendly, allowing users to easily provide ratings, filter, and sort reviews as per their preferences.

Table 3-2 represents all non-functional requirements in AgriMart platform.

3.2 List of Actors

Farmer: The primary seller on the platform who lists their crops, manages auction listings, and communicates with buyers to negotiate deals and arrange delivery. Farmers can also provide ratings and reviews for buyers.

Buyer: The primary purchaser on the platform who participates in auctions, places bids on crops, and communicates with farmers to discuss terms, arrange payment, and coordinate delivery. Buyers can also provide ratings and reviews for farmers.

Administrator: The platform's administrator oversees the overall functioning of the system, manages user registrations, resolves disputes, ensures fair auction processes, and provides support to farmers and buyers.

3.3 List of use cases

Register: Users (farmers and buyers) can create their accounts on the platform by providing necessary information like contact details and farming preferences.

Login: Registered users can log into their accounts.

Add Crop: Farmers can list their crops for auction, providing details such as crop type, quantity, quality, and the starting price.

Search Crops: Buyers can browse through available crop listings and filter them based on criteria like crop type, location, and price range.

Bid on Crop: Buyers can participate in crop auctions by placing bids on the crops they're interested in, specifying the bid amount and any additional conditions.

Accept/Reject Bid: The system automatically accepts the highest bids for a particular crop listing, and the farmer can choose to accept any of them, confirming the transaction and initiating further communication with the buyer.

Chat: Provides a communication channel between farmers and buyers to discuss transaction details, negotiate terms, and coordinate logistics.

SMS/Email Notifications: Sends automated notifications to farmers and buyers regarding bid updates, transaction status, and important platform announcements.

Make Payment: Buyers can make secure online payments through integrated payment gateways, ensuring timely and accurate transaction settlements.

Delivery Tracking: Allows farmers and buyers to track the progress of crop delivery, providing real-time updates on shipment status and estimated arrival times.

Request Quality Inspection: Facilitates the scheduling and execution of quality inspections for crops, ensuring compliance with industry standards and certifications.

Provide Rating and Review: Enables farmers and buyers to rate and review each other based on their experiences, contributing to the reputation system and fostering trust within the platform.

View Predictions and Analytics: Farmers and buyers can access predictive analytics and market insights generated by machine learning algorithms, assisting in decision-making and optimizing transactions.

View Bids: Users can view the bids they have placed or received on their crop listings, keeping track of their auction activities.

View Bid Status: Users can check the status of their bids on various crop listings, ensuring transparency throughout the bidding process.

Add Complaint: Users can submit complaints or issues related to transactions for resolution, ensuring a mechanism for dispute resolution.

Approve User: Admin or moderators can approve new user registrations or user-generated content, ensuring the integrity and quality of platform users and content.

Add Quality Report: Users can add reports on the quality of the crops they have received or sold, contributing to a transparent quality assessment system.

Release Payment: Admin can release payments to Farmer after successful delivery of crops and not created any dispute.

Dashboard: Provides users with an overview of their activities and transactions on the platform, offering a convenient way to monitor their engagement and history.

3.4 System use case diagram

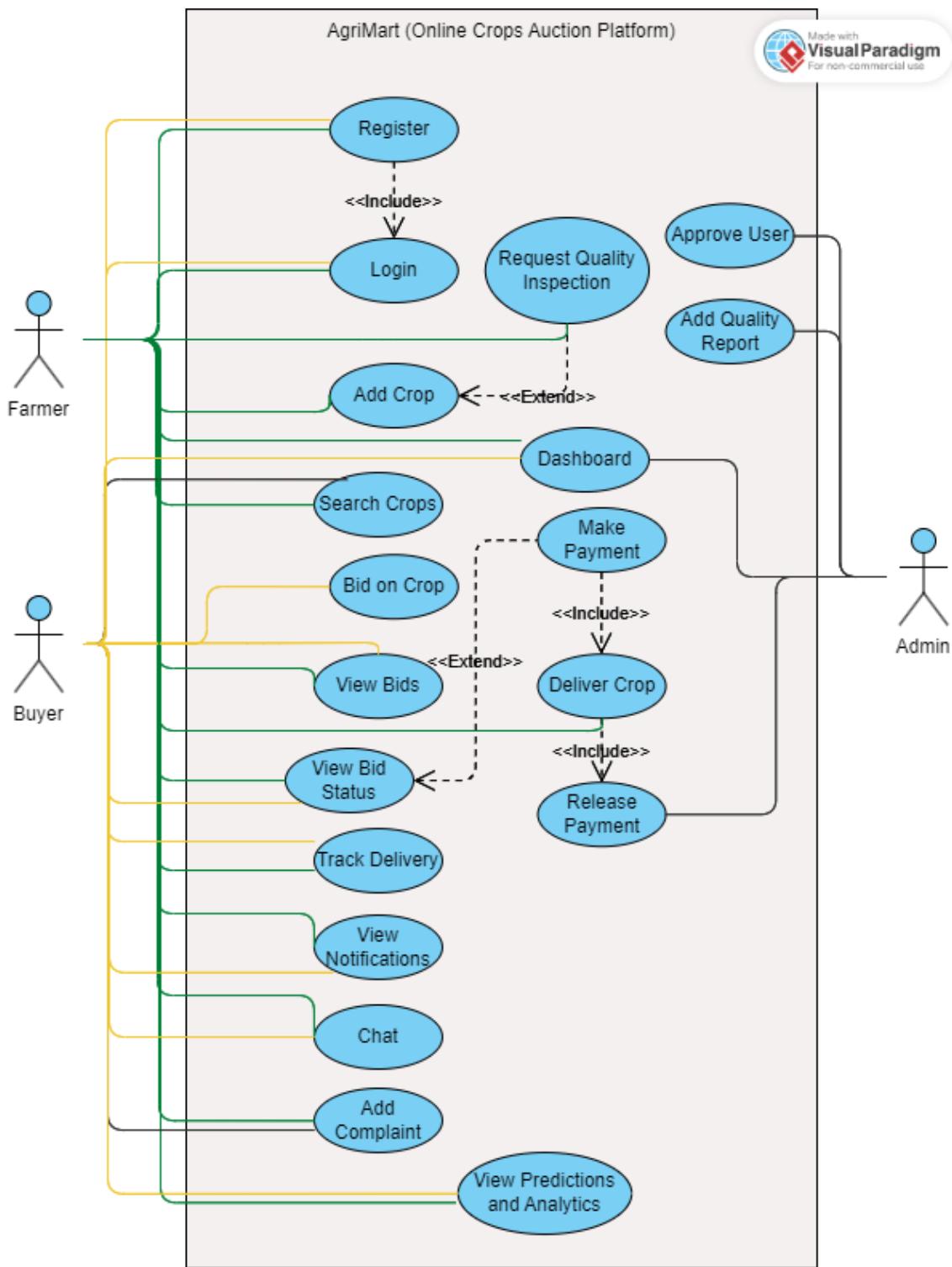


Figure 3-1: AgriMart Use Case Diagram

Figure 3-1 is use case diagram of AgriMart platform which represents all actors and their use cases. It contains actors farmer, buyer and admin and use cases sign up, login , list crops, quality inspection, browse crops, deliver crops . notifications and predictions and analytics.

3.5 Extended use cases

Table 3-3:Extended use case Register

Use Case ID:	UC001		
Use Case Name:	Register		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	10-May-2023	Last Revision Date:	11-May-2023
Actors:	Farmer, Buyer		
Description:	Allows farmers and buyers to register on the platform by providing necessary information CNIC, phone number, location, email address, password and preferences.		
Trigger:	User initiates the register process		
Preconditions:	None		
Post conditions:	1. User account created and user can login to platform		
Normal Flow:	1. User selects the registration option on the platform. 2. System presents the registration form. 3. User fills in the required information, including contact details and preferences. 4. User submits the registration form. 5. System send OTP on phone number. 6. System validates the information provided. 7. System creates a new account for the user. 8. System displays a success message and login on platform.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	In step 5 of the normal flow, if the user does not receive the OTP due to network issues: 1. System prompts the user with an option to resend the OTP. 2. User selects the option to resend the OTP. 3. System resends the OTP to the user's phone number. 4. User enters the received OTP. 5. Use case resumes at step 7 of the normal flow.		
Exceptions:	5a. In step 5 of the normal flow, if the user enters an invalid OTP: 1. System displays an error message to the user, prompting them to re-enter the OTP. 2. User re-enters the correct OTP. 3. Use case resumes at step 7 of the normal flow. 6a. In step 6 of the normal flow, if the information provided by the user is invalid: 1. System displays an error message to the user, indicating that they need to enter valid information. 2. User enters valid information. 3. Use case resumes at step 7 of the normal flow.		
Frequency of Use:	Once per user		

Table 3-3 represents description of use case Register.

Table 3-4: Extended use case Login

Use Case ID:	UC002		
Use Case Name:	Login		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	10-May-2023	Last Revision Date:	11-May-2023
Actors:	Farmer, Buyer		
Description:	Allows registered farmers and buyers to log in to the platform using their email address and password.		
Trigger:	User initiates the login process.		
Preconditions:	1. User has successfully registered on the platform. 2. User account is active.		
Post conditions:	User is successfully logged into the platform.		
Normal Flow:	1. User selects the login option on the platform. 2. System presents the login form. 3. User enters their registered email address and password. 4. User submits the login form. 5. System validates the entered information. 6. If the entered information is valid, the system logs the user into the platform. 7. System displays a success message and grants access to the user's account and dashboard.		
Alternative Flows: [Alternative Flow 1 - Forgotten Password]	In step 3 of the normal flow, if the user forgets their password: 1. User clicks on the "Forgot Password" link. 2. System presents a password reset form. 3. User enters their registered email address. 4. System sends a password reset link to the user's email address. 5. User follows the link to reset their password. 6. Use case resumes at step 3 of the normal flow.		
Exceptions:	4a. In step 4 of the normal flow, if the user enters an incorrect password: 1. System displays an error message to the user, prompting them to re-enter the password. 2. User re-enters the correct password. 3. Use case resumes at step 5 of the normal flow. 4b. In step 5 of the normal flow, if the entered email address is not found in the system: 1. System displays an error message to the user, indicating that the email address is not registered. 2. User is prompted to register if they are a new user or to re-enter their email address. 3. Use case resumes at step 3 of the normal flow.		
Frequency of Use:	Whenever a registered user wants to access their account.		

Table 3-4 represents the description of use case Login

Table 3-5: Extended use case List Crops

Use Case ID:	UC003		
Use Case Name:	Add Crop		
Created By:	Malaika Aziz	Last Updated By:	Malaika Aziz
Date Created:	10-May-2023	Last Revision Date:	13-May-2023
Actors:	Farmer		
Description:	Farmers can list their crops for auction by providing details like images, crop name, type, quantity, quality, starting price.		
Trigger:	Farmer chooses to add crop for auction		
Preconditions:	Farmer is logged in.		
Post conditions:	Crop listing completed and displayed on platform		
Normal Flow:	<ol style="list-style-type: none"> 1. Farmer navigates to the "List Crops" section of the platform. 2. System presents the crop listing form. 3. Farmer provides details such as crop type, quantity, quality, and starting price. 4. Farmer submits the crop listing form. 5. System validates the information provided. 6. System creates a new crop listing. 7. System displays a success message and shows the crop listing on the platform. 		
Exceptions:	<p>5a. In step 5 of the normal flow, if there is a validation error:</p> <ol style="list-style-type: none"> 1. The system displays an error message, prompting the farmer to correct the information. 2. The farmer enters the correct information. 3. The use case resumes at step 6 of the normal flow. 		
Frequency of Use:	Varies per Farmer and crops availability		

Table 3-5 represents description of use case Add Crop.

Table 3-6: Extended use case Browse Crops

Use Case ID:	UC004		
Use Case Name:	Search Crops		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	10-May-2023	Last Revision Date:	12-May-2023
Actors:	Buyer		
Description:	Buyers can browse through the available crop listings on the platform, filtering by criteria such as crop type, location, and price range.		
Trigger:	Buyer initiates the crop browsing process.		
Preconditions:	Buyer is logged in.		
Post conditions:	<ol style="list-style-type: none"> 1. Crops matching the selected criteria are displayed to the buyer. 		
Normal Flow:	<ol style="list-style-type: none"> 1. Buyer selects the option to browse crops on the platform. 2. System presents the crop browsing interface. 3. Buyer applies filters such as crop type, location, and price range to narrow down the search. 		

	4. System retrieves crop listings matching the selected criteria. 5. System displays the filtered crop listings to the buyer.
Frequency of Use:	Varies per buyer's crop browsing activities.

Table 3-6 represents the extended description of Search crops in AgriMart.

Table 3-7: Extended use case Bid Crop

Use Case ID:	UC005		
Use Case Name:	Bid on Crop		
Created By:	Malaika Aziz	Last Updated By:	Malaika Aziz
Date Created:	10-May-2023	Last Revision Date:	12-May-2023
Actors:	Buyer		
Description:	Buyers can participate in auctions by placing bids on the desired crops, indicating the bid amount and additional conditions.		
Trigger:	Buyer initiates the bid placement process		
Preconditions:	Buyer is logged in, and the auction for the desired crop listing is active.		
Post conditions:	Bid is placed on the crop listing.		
Normal Flow:	<ol style="list-style-type: none"> Buyer selects the option to place a bid on a specific crop listing. System presents the bid form for the selected crop listing. Buyer enters the bid amount and any additional conditions, if applicable. Buyer submits the bid form. System validates the bid information. System records the bid on the crop listing. System displays a success message confirming the bid placement. 		
Frequency of Use:	Varies per buyer's crop browsing activities.		

Table 3-7 represents the extended description of use case bid crops.

Table 3-8: Extended Automated Bid Acceptance

Use Case ID:	UC006		
Use Case Name:	Accept/Reject Bid		
Created By:	Malaika Aziz	Last Updated By:	Malaika Aziz
Date Created:	10-May-2023	Last Revision Date:	14-May-2023
Actors:	System, Buyer, Farmer		
Description:	Farmers have the choice to accept or reject bids placed on their crop listings		
Trigger:	Farmer initiates the decision-making process for a received bid.		
Preconditions:	<ol style="list-style-type: none"> Farmer is logged in. At least one bid has been placed on the farmer's crop listing. 		
Post conditions:	The bid is either accepted or rejected by the farmer.		

Normal Flow:	<ol style="list-style-type: none"> 1. The farmer navigates to the "My Auctions" section of the platform. 2. The system presents the list of crop listings with pending bids. 3. The farmer selects a specific crop listing with pending bids. 4. The system displays the details of the bids, including bid amounts and any additional conditions. 5. The farmer reviews the bids and decides whether to accept or reject each bid. 6. For each bid, the farmer selects the corresponding action (accept or reject). 7. The system records the farmer's decision. 8. If the farmer accepts a bid, the system initiates further communication with the buyer regarding payment and delivery details. 9. If the farmer rejects a bid, the system notifies the respective buyer of the rejection.
Frequency of Use:	Varies based on the number of crop listings and bids.

Table 3-8 represents the description of use case Automated bid acceptance.

Table 3-9: Extended use case Real-time Chat

Use Case ID:	UC007		
Use Case Name:	Chat		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	10-May-2023	Last Revision Date:	10-May-2023
Actors:	Farmer, Buyer		
Description:	Provides a communication channel between farmers and buyers to discuss transaction details, negotiate terms, and coordinate logistics in real-time.		
Trigger:	Farmer or buyer initiates a chat conversation.		
Preconditions:	Farmer and buyer are logged in and have selected a specific transaction or crop listing.		
Post conditions:	Real-time chat conversation is established between the farmer and buyer.		
Normal Flow:	<ol style="list-style-type: none"> 1. Buyer selects the option to initiate a chat conversation. 2. System presents the chat interface with the relevant transaction or crop listing details. 3. Farmer or buyer sends a message in the chat. 4. System delivers the message to the recipient. 5. Recipient receives the message in real-time. 6. Recipient can view and respond to the message. 7. Conversation continues in real-time with messages exchanged between the farmer and buyer. 		
Frequency of Use:	Varies based on the interaction between farmers and buyers.		

Table 3-9 represents the description of use case real time chat between buyer and farmer.

Table 3-10: Extended Use case SMS/Email notification

Use Case ID:	UC008		
Use Case Name:	SMS/Email Notifications		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	13-May-2023	Last Revision Date:	16-May-2023
Actors:	System, Farmer, Buyer		
Description:	Sends automated notifications to farmers and buyers regarding bid updates, transaction status, and important platform announcements via SMS or email.		
Trigger:	System generates a notification for a specific event or update.		
Preconditions:	Farmer and/or buyer have provided valid contact information (phone number and/or email address) during registration.		
Post conditions:	Notification is sent to the recipient via SMS or email.		
Normal Flow:	<ol style="list-style-type: none">1. System generates a notification for a specific event or update (e.g., bid update, transaction status change, platform announcement).2. System retrieves the contact information of the relevant farmer(s) and/or buyer(s) associated with the event or update.3. System composes the notification message.4. System sends the notification message via SMS or email to the recipient(s) based on their provided contact information.		
Frequency of Use:	Varies based on the occurrence of events and updates.		
Special Requirements:	Valid contact information (phone number or email address).		

Table 3-10 represents the description of use case sms and email notifications.

Table 3-11: Extended Make Payment

Use Case ID:	UC009		
Use Case Name:	Make Payment		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	13-May-2023	Last Revision Date:	15-May-2023
Actors:	Buyer, Payment Gateway		
Description:	Buyers can make secure online payments through integrated payment gateways, ensuring timely and accurate transaction settlements.		
Trigger:	Buyer initiates the payment process for a completed transaction.		
Preconditions:	Buyer has selected a specific transaction for payment and has a valid payment method linked to their account.		
Post conditions:	Payment is successfully processed, and transaction settlement is confirmed.		
Normal Flow:	<ol style="list-style-type: none">1. Buyer selects the option to make a payment for a completed transaction.2. System presents the payment interface with the transaction details and payment options.3. Buyer selects a payment method from the available options.		

	<ol style="list-style-type: none"> 4. System redirects the buyer to the chosen payment gateway. 5. Buyer enters the required payment details (e.g., credit card information, authentication code). 6. Payment gateway securely processes the payment transaction. 7. Payment gateway provides a response indicating the payment status (success or failure). 8. System receives the payment status from the payment gateway. 9. If the payment is successful, the system confirms the transaction settlement. 10. System updates the transaction status as "Paid" and notifies the buyer and farmer.
Includes:	Payment Gateway Integration
Frequency of Use:	Varies based on the number of completed transactions requiring payment.
Special Requirements:	Integration with secure and reliable payment gateways.
Assumptions:	Buyer has a valid payment method linked to their account.

Table 3-11 represents the description of use case make payment.

Table 3-12: Extended Delivery Tracking

Use Case ID:	UC010		
Use Case Name:	Delivery Tracking		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	15-May-2023	Last Revision Date:	16-May-2023
Actors:	Farmer, Buyer, Shipping Service		
Description:	Allows farmers and buyers to track the progress of crop delivery, providing real-time updates on shipment status and estimated arrival times.		
Trigger:	Farmer or buyer initiates the delivery tracking process for a specific transaction.		
Preconditions:	The transaction involves physical delivery of crops, and a shipping service is engaged.		
Post conditions:	Current shipment status and estimated arrival time are displayed to the farmer and buyer.		
Normal Flow:	<ol style="list-style-type: none"> 1. Farmer or buyer selects the option to track the delivery for a specific transaction. 2. System retrieves the shipment details associated with the transaction. 3. System contacts the shipping service to obtain the current shipment status and estimated arrival time. 4. Shipping service provides the requested information to the system. 5. System displays the current shipment status and estimated arrival time to the farmer and buyer. 		

Table 3-12 represents the description of use case delivery tracking.

Table 3-13: Extended Quality Inspection

Use Case ID:	UC011		
Use Case Name:	Request Quality Inspection		
Created By:	Abu Bakar	Last Updated By:	Abu Bakar
Date Created:	15-May-2023	Last Revision Date:	15-May-2023
Actors:	Farmer, Quality Inspector		
Description:	Facilitates the scheduling and execution of quality inspections for crops, ensuring compliance with industry standards and certifications.		
Trigger:	Farmer requests a quality inspection for a specific crop.		
Preconditions:	Farmer has indicated the need for a quality inspection during the crop listing.		
Post conditions:	Quality inspection is scheduled and performed, and the inspection report is provided to the System.		
Normal Flow:	<ol style="list-style-type: none">1. Farmer requests a quality inspection for a specific crop.2. System prompts the farmer to provide available time slots for the inspection.3. Farmer selects the preferred time slot from the available options.4. System schedules the quality inspection based on the selected time slot and assigns a quality inspector.5. System sends a notification to the assigned quality inspector with the inspection details.6. Quality inspector performs the inspection according to industry standards and certifications.7. Quality inspector generates an inspection report, documenting the findings.8. Quality inspector submits the inspection report to the system.9. System notifies the farmer about the completion of the quality inspection and provides access to the inspection report.		
Frequency of Use:	Varies based on the farmer's request for quality inspections.		
Special Requirements:	Availability of qualified quality inspectors and adherence to industry standards.		

Table 3-13 represents the description of use case quality inspection.

3.6 User interfaces (mock screens)

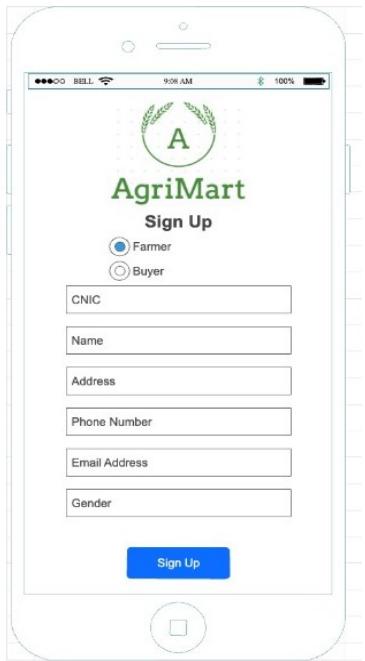


Figure 3-2: Sign Up screen prototype



Figure 3-3: Login Screen prototype

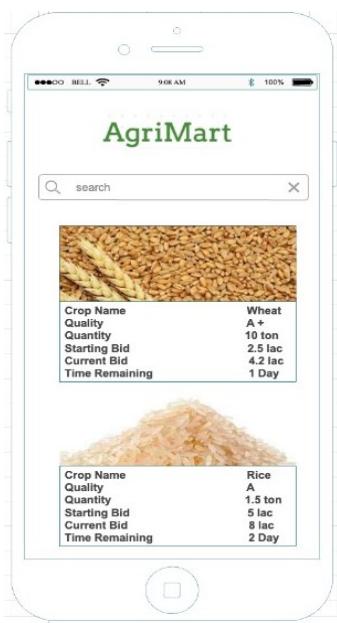


Figure 3-4: Crops View Screen Prototype

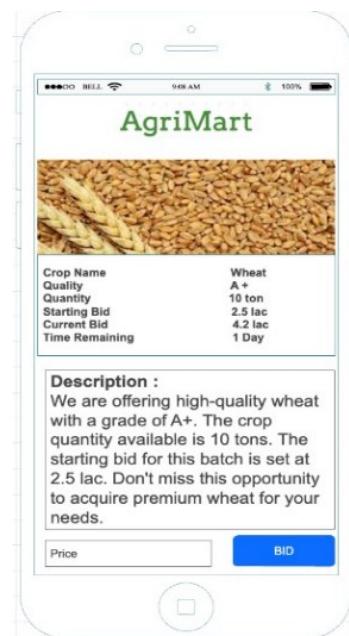


Figure 3-5: Crop Detail Screen prototype

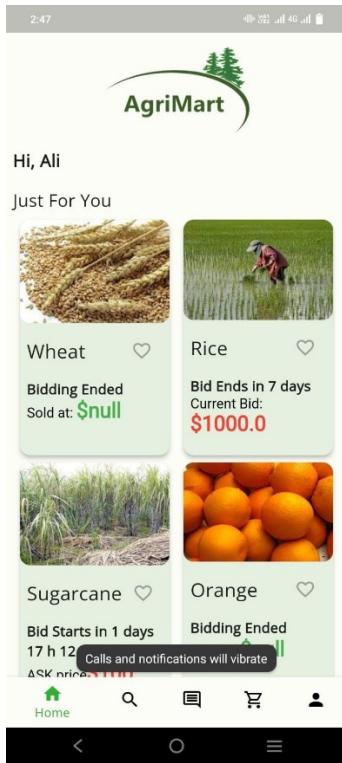


Figure 3-6: Available Crops Page

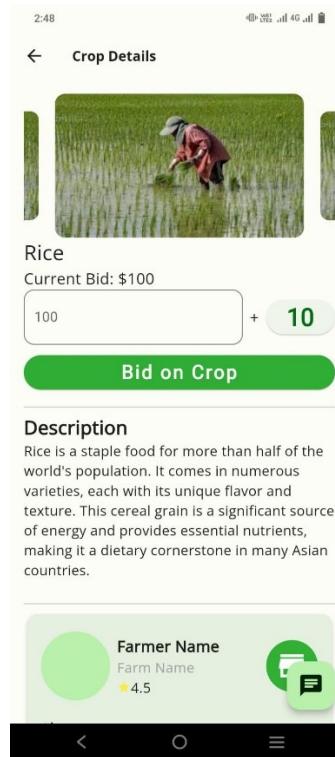


Figure 3-7: Crops Detail Page

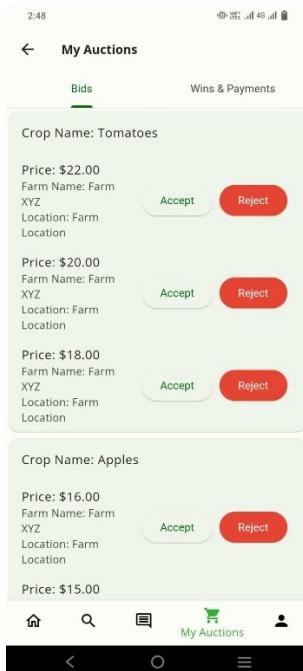


Figure 3-8: Farmer's Bid Decision Screen

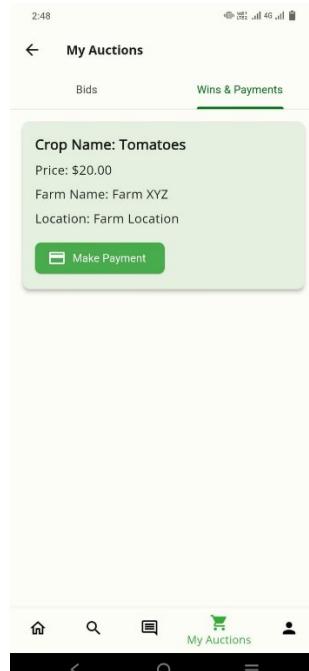


Figure 3-9: Buyer Proceed to Payment Screen

4. DATA FLOW DIAGRAM (OPTIONAL)

4.1 Data Flow Diagram Level 0

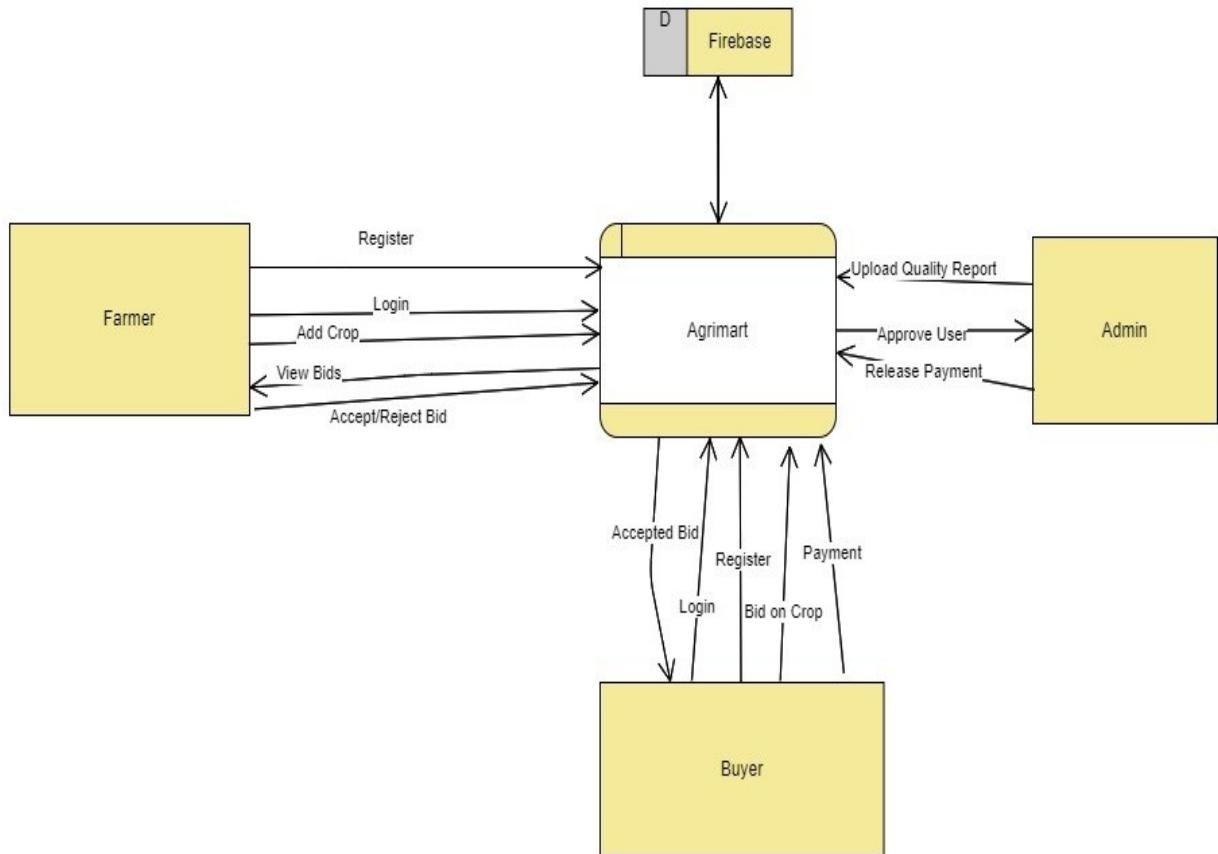


Figure 4-1: Data Flow Diagram Level 0

Figure 4-1 represents the Data Flow Diagram Level 0 of AgriMart, providing an overview of the platform's core features and their data interactions. The diagram showcases essential functions, including user registration, login, crop listing, bidding on crops, viewing bids, and payment processing. Users register on the platform and log in to access its services, with farmers adding crop listings for auction and buyers placing bids. The diagram illustrates the flow of data between these key features, emphasizing the fundamental processes and data exchanges within the AgriMart system, serving as a foundational representation of the platform's high-level functionality.

In this DFD Level 0, the interactions between users, the platform, and key processes are depicted succinctly. Users initiate actions such as registering and logging in, leading to vital functions like adding crop listings, participating in auctions through bids, and managing bids with the ability to view them. The diagram also highlights the integral role of payment processing in facilitating secure online transactions within the AgriMart platform. This visualization serves as a valuable reference for understanding the core data flows and functionalities, laying the groundwork for a comprehensive understanding of AgriMart's operations and interactions.

4.2 Data Flow Diagram Level 1

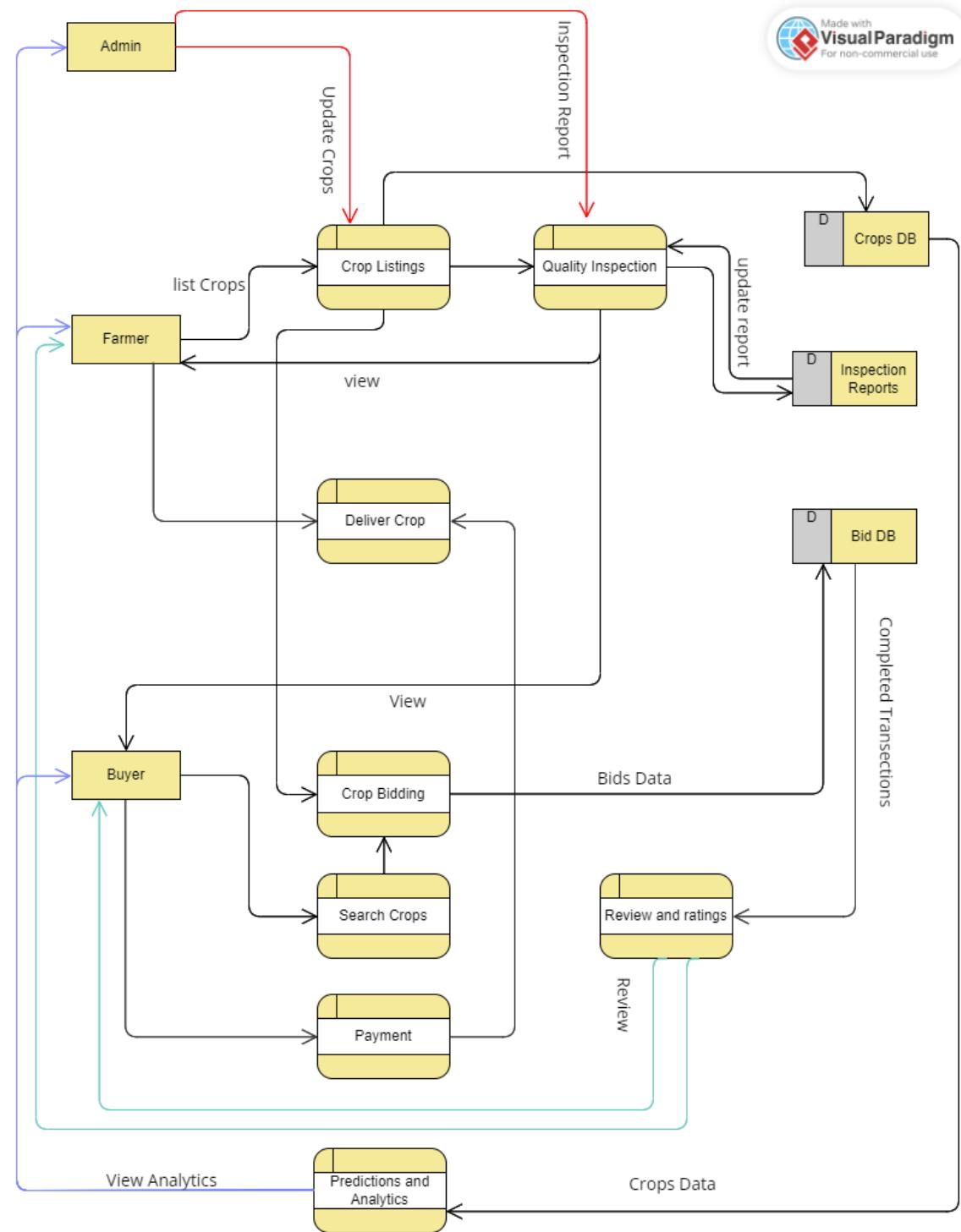


Figure 4-2: Data Flow Diagram

5. SYSTEM DESIGN

5.1 System Architecture Diagram

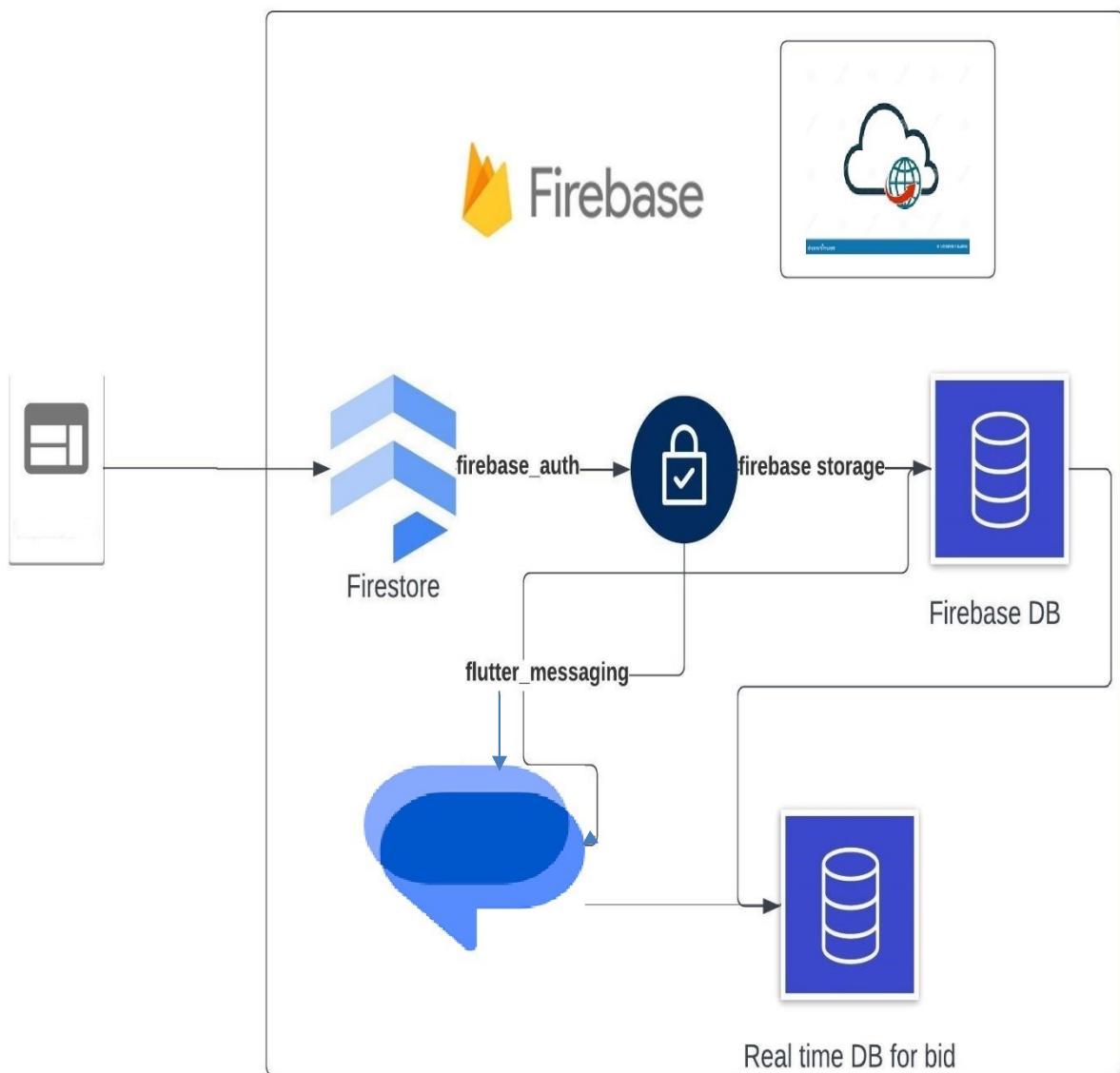


Figure 5-1: System Artitecture Diagram of Agrimart

Figure 5-1 represent how our app will interact with databases on internet. This is system architecture diagram of our platform.

5.2 Class Diagram

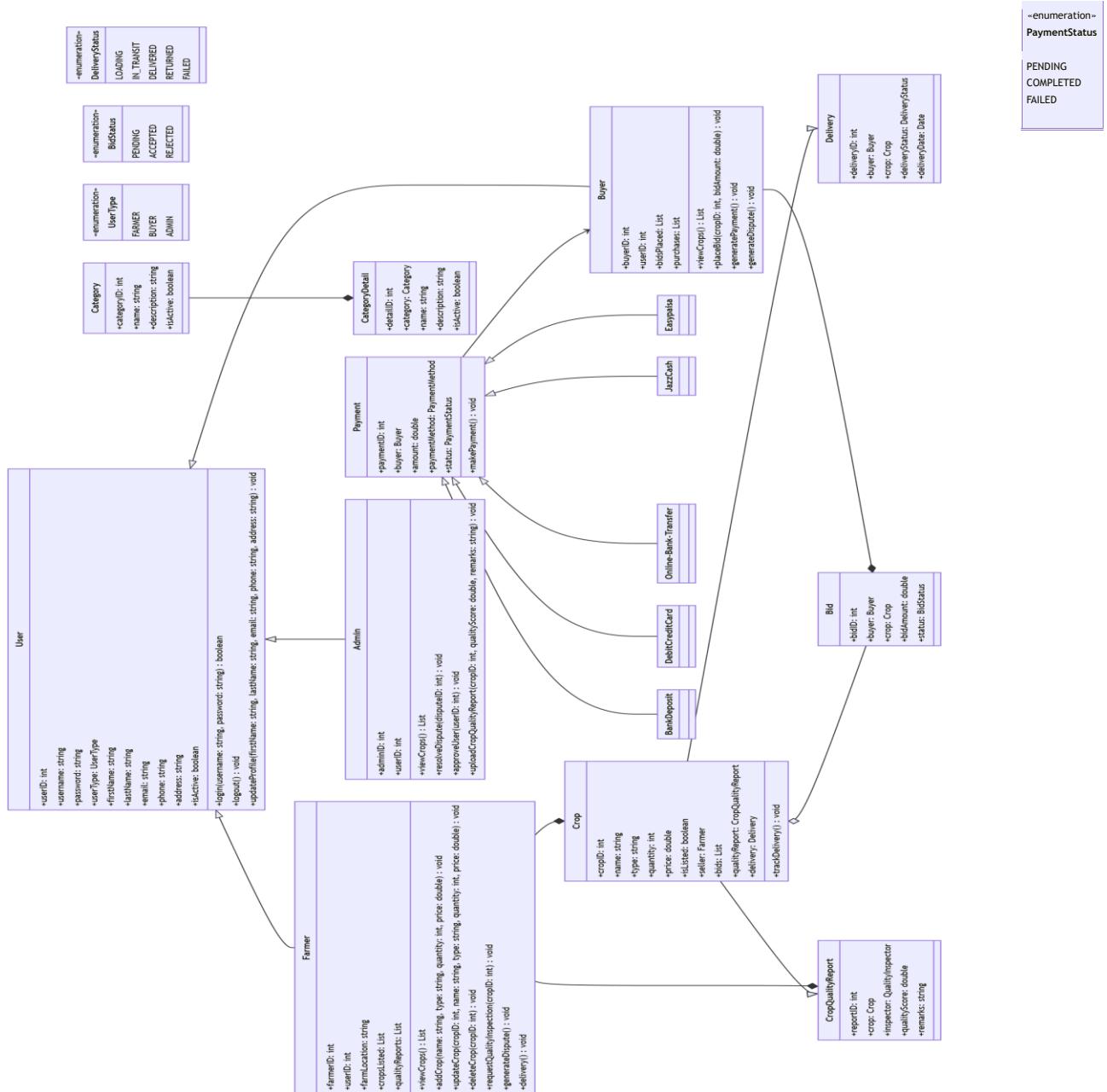


Figure 5-2: Class Diagram of AgriMart

Figure 5-2 represents the different class in AgriMart platform and their relationship with each other.

5.3 Sequence Diagrams

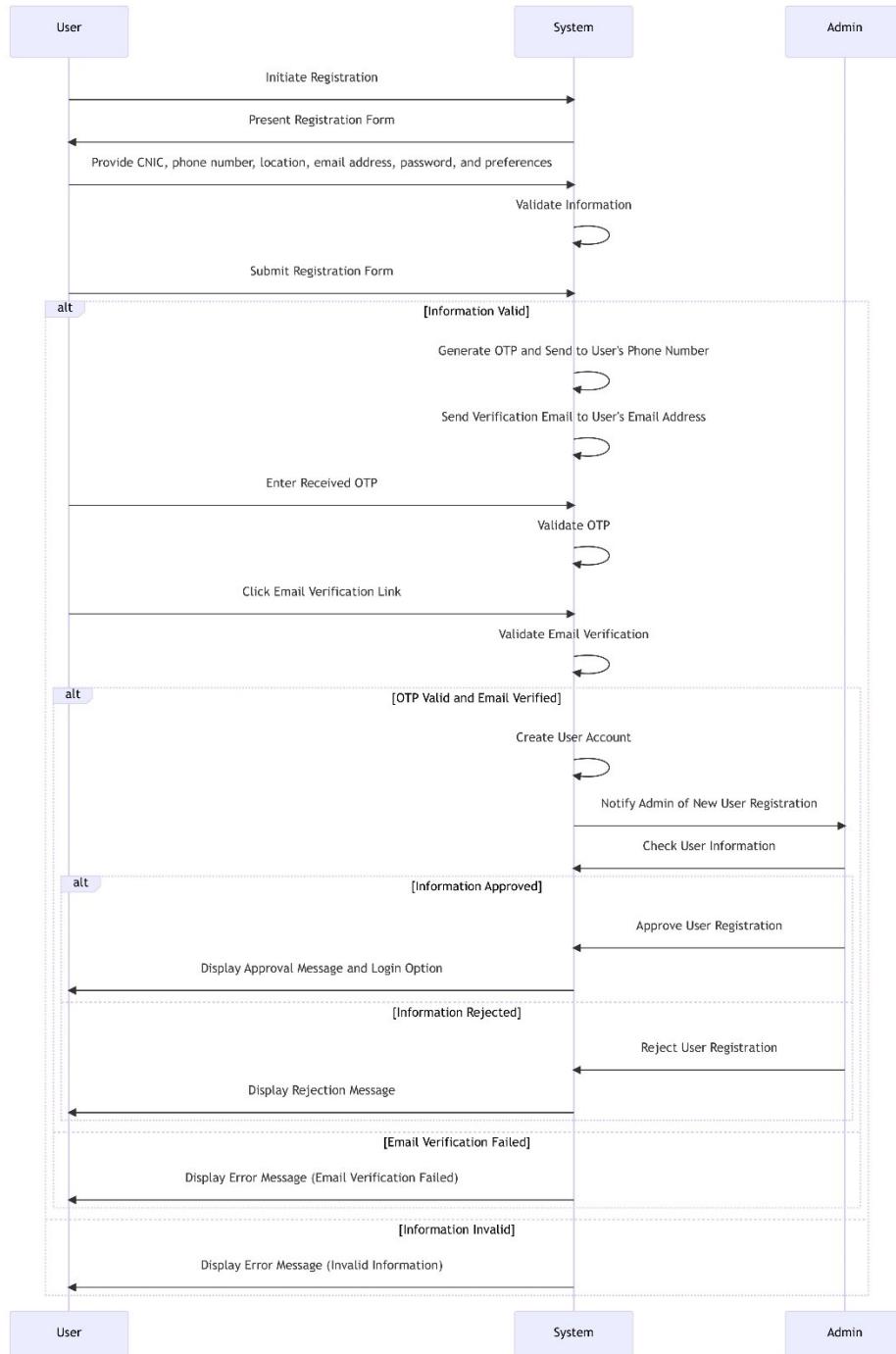


Figure 5-3: Buyer Register Sequence Diagram

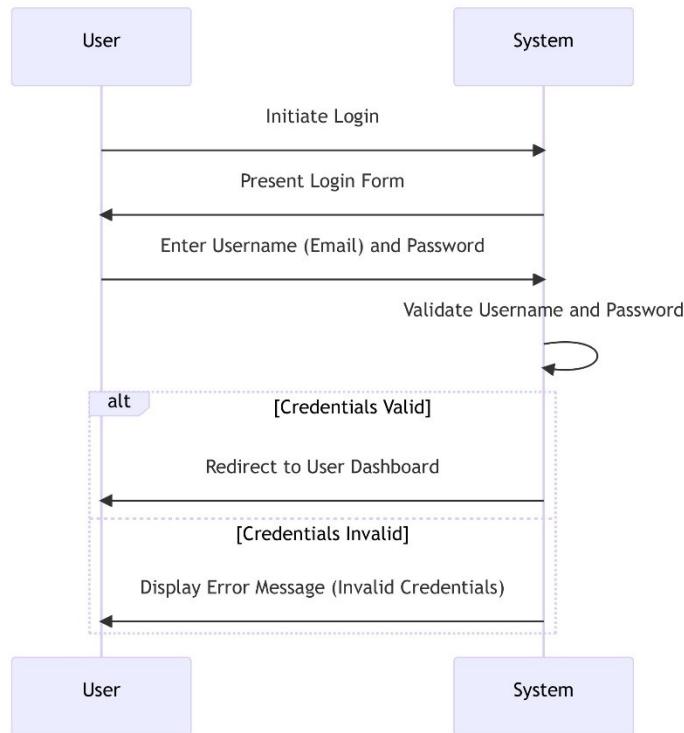


Figure 5-4:Login Sequence Diagram

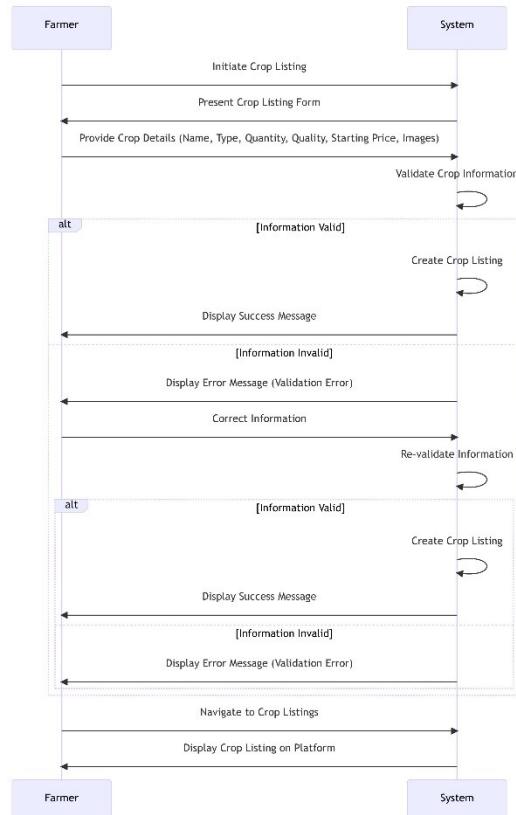


Figure 5-5: Add Crop Sequence Diagram

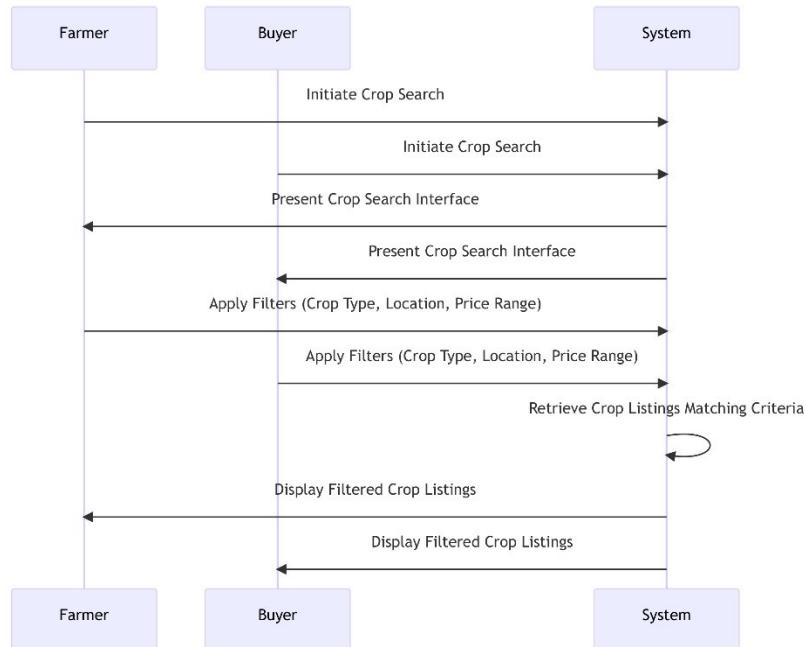


Figure 5-6: Browse Crops Sequence Diagram



Figure 5-7: Accept / Reject bid by Farmer

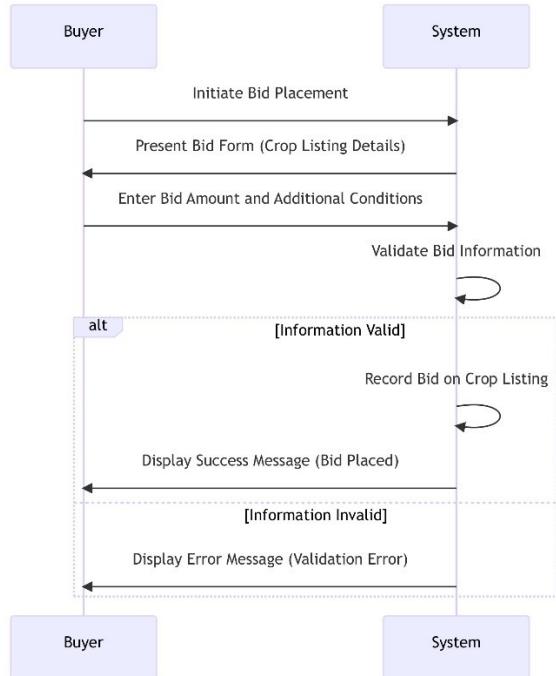


Figure 5-8: Bid on Crop

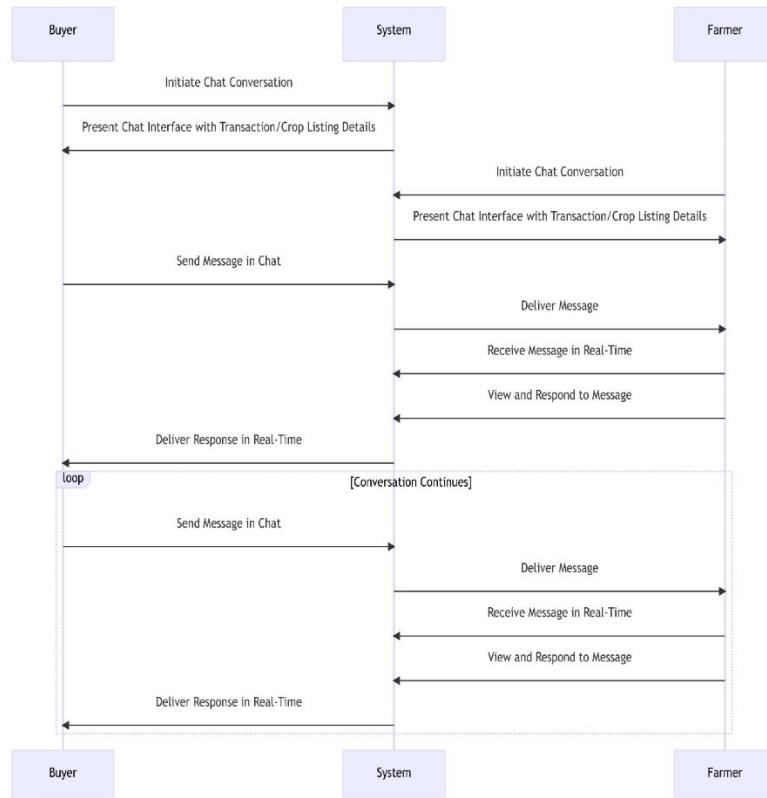


Figure 5-9: Initialize Chat

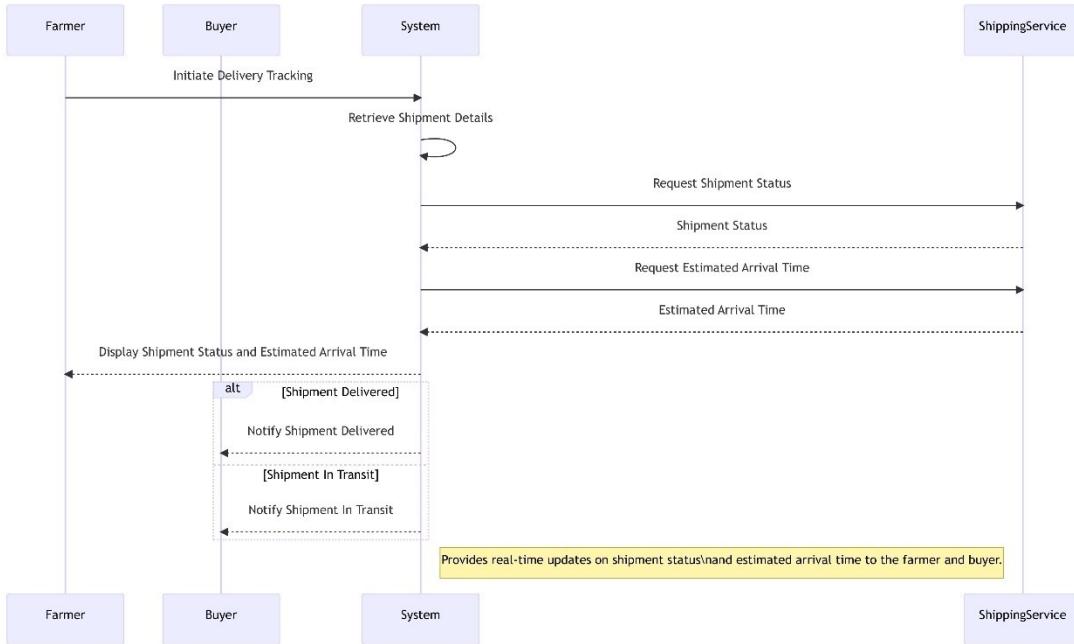


Figure 5-10: Delivery Tracking

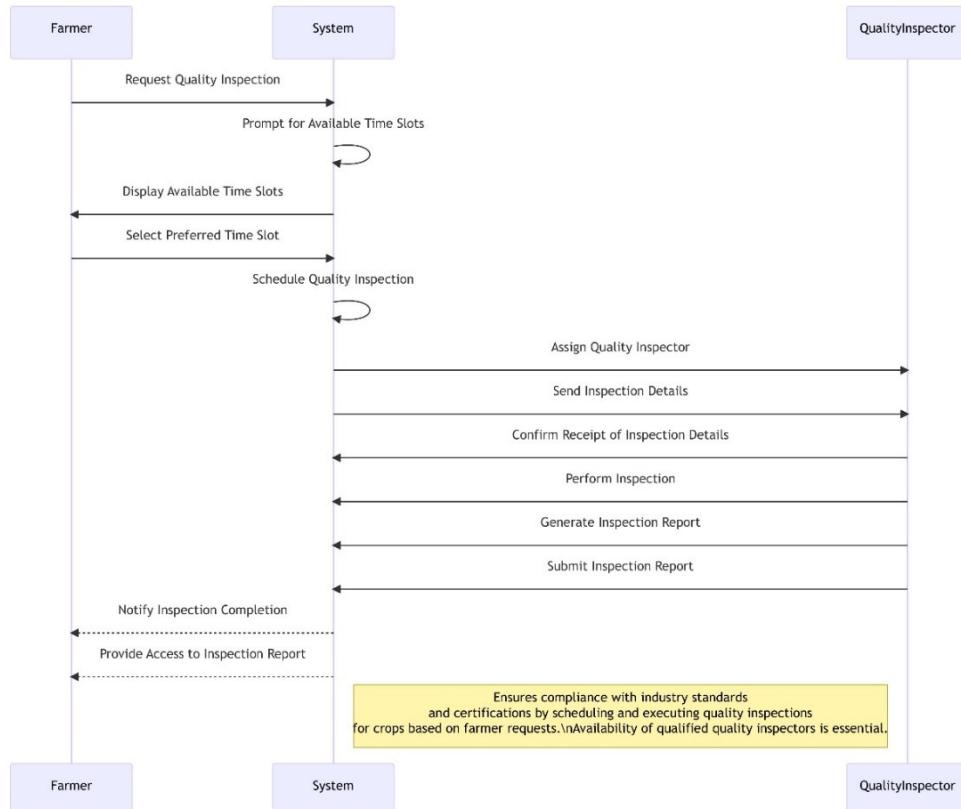


Figure 5-11: Request Quality Inspection Sequence Diagram

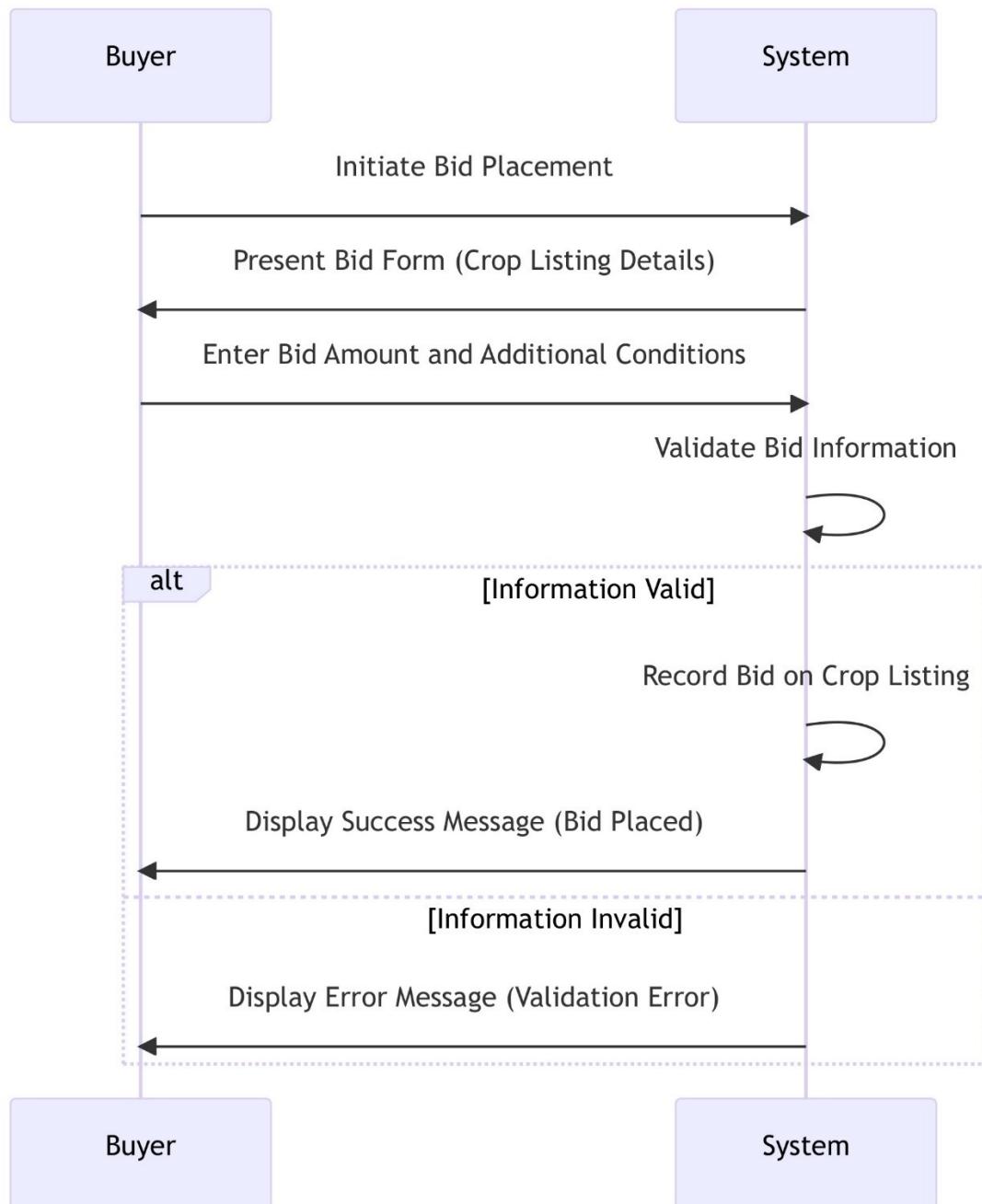


Figure 5-12: Bid on Crops Sequence Diagram

Figure 5-12 shows sequence of Bid process when buyer bid on crop system validated the information and add bid on that crop.

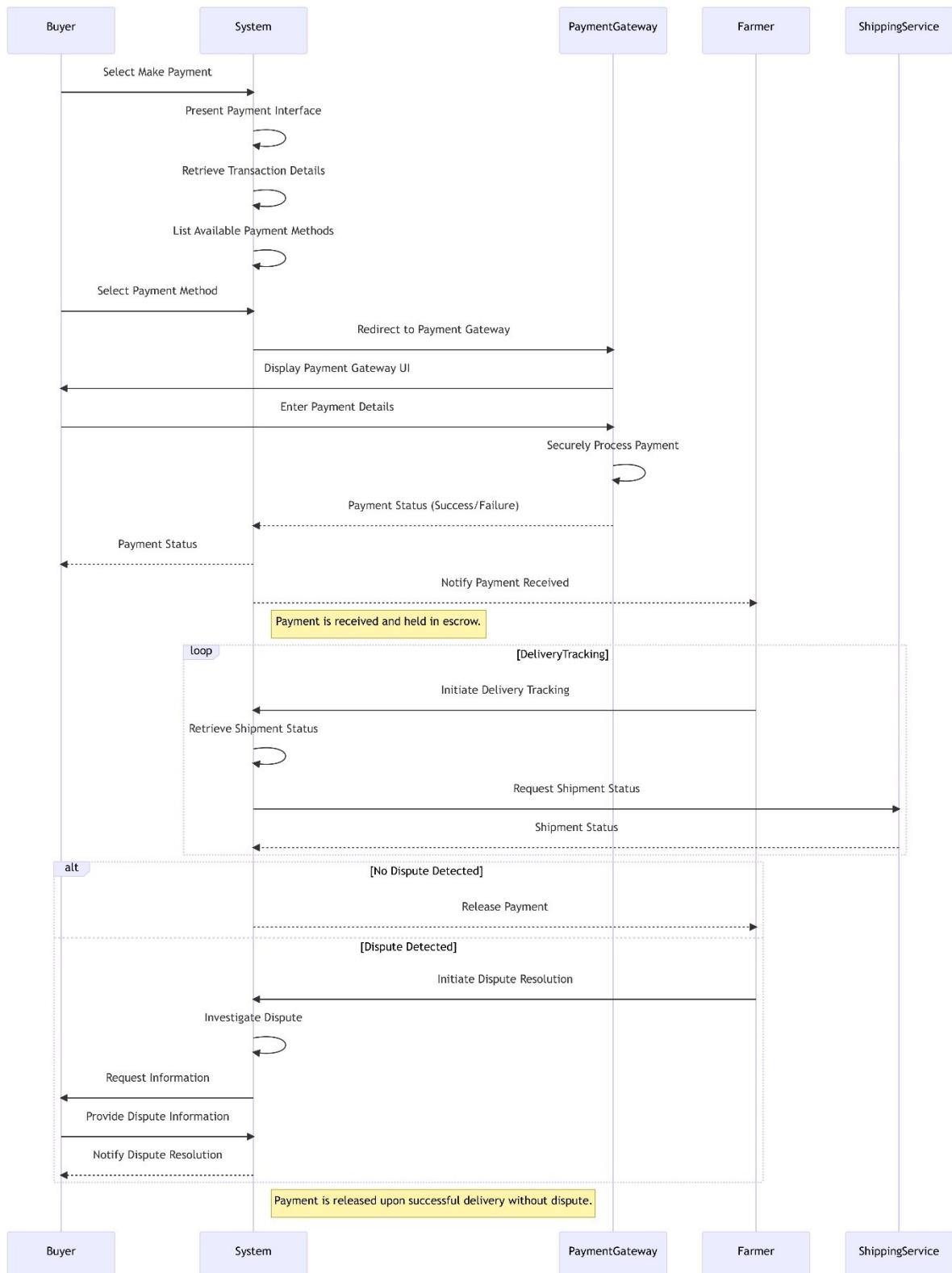


Figure 5-13: Make Payment Sequence Diagram

5.4 ERD

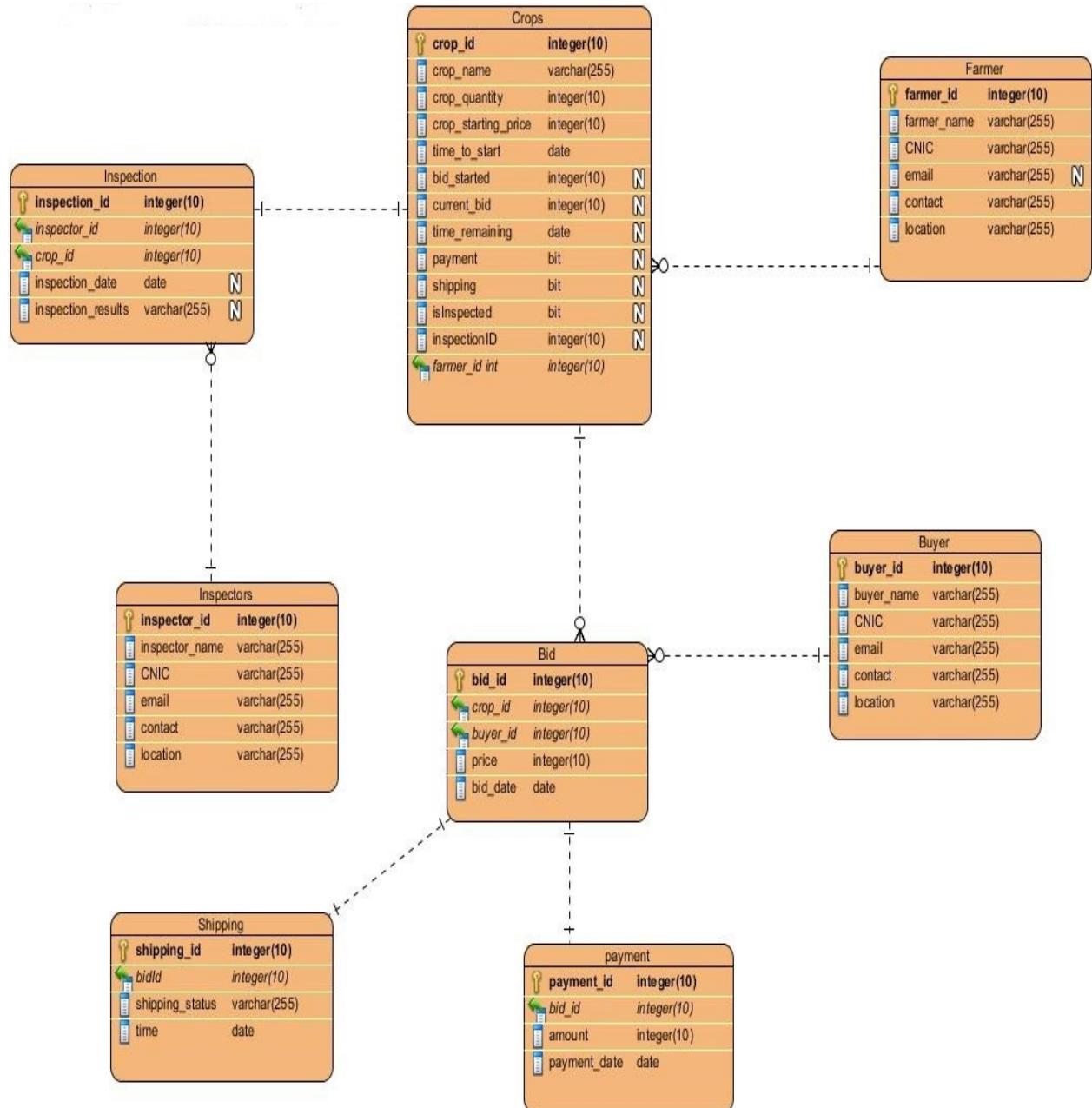


Figure 5-14: Entity Relationship Diagram

Figure 5-14 represents all entities and their relationship with each other in AgriMart platform. It shows how entities are related to each other.

5.5 Data Dictionary

Table 5-1: Data Dictionary of Agrimart

Element Name	Type	Validation	Mandatory	Remarks
Crop ID	Integer	(10)	Yes	Unique identifier for the crop.
Crop Name	Varchar(255)		Yes	Name of the crop.
Crop Quantity	Integer	(10)	Yes	Quantity of the crop available.
Starting Price	Integer	(10)	Yes	Initial price set for the crop.
Bid Start Time	Integer	(10)	Yes	Time in minutes to start the bidding.
Current Bid	Integer	(10)	Yes	Current highest bid for the crop.
Time Remaining	Integer	(10)	Yes	Time remaining in minutes for bidding.
Payment	Bit		Yes	Indicates whether payment is completed (1) or not (0).
Shipping	Bit		Yes	Indicates whether shipping is done (1) or not (0).
Is Inspected	Bit		Yes	Indicates whether the crop is inspected (1) or not (0).
Inspection ID	Integer	(10)	Yes	Unique identifier for crop inspection.
Farmer ID	Integer	(10)	Yes	Identifier for the farmer.
Farmer Name	Varchar(255)		Yes	Name of the farmer.
CNIC	Varchar(255)		Yes	National ID Card number of the farmer.
Email	Varchar(255)		Yes	Email address of the farmer.
Contact	Varchar(255)		Yes	Contact number of the farmer.
Location	Varchar(255)		Yes	Location or address of the farmer.
Inspection Date	Date		No	Date when the crop was inspected.
Inspection Results	Varchar(255)		No	Results of the crop inspection.
Buyer ID	Integer	(10)	Yes	Unique identifier for the buyer.
Buyer Name	Varchar(255)		Yes	Name of the buyer.
Buyer CNIC	Varchar(255)		Yes	National ID Card number of the buyer.

Buyer Email	Varchar(255)		Yes	Email address of the buyer.
Buyer Contact	Varchar(255)		Yes	Contact number of the buyer.
Buyer Location	Varchar(255)		Yes	Location or address of the buyer.
Bid ID	Integer	(10)	Yes	Unique identifier for the bid.
Price	Integer	(10)	Yes	Price offered in the bid.
Bid Date	Date		Yes	Date when the bid was placed.
Shipping ID	Integer	(10)	Yes	Unique identifier for shipping details.
Bid ID (Foreign Key)	Integer	(10)	Yes	Foreign key linking bid to shipping.
Shipping Status	Varchar(255)		Yes	Status of the shipping process.
Time	Date		Yes	Date and time of the bid.
Payment ID	Integer	(10)	Yes	Unique identifier for payment details.
Bid ID (Foreign Key)	Integer	(10)	Yes	Foreign key linking bid to payment.
Amount	Integer	(10)	Yes	Amount of the payment.
Payment Date	Date		Yes	Date when the payment was made.

The data dictionary outlines the essential elements and attributes of a comprehensive agricultural bidding and trading system. It covers various aspects, starting with crop information, including crop ID, name, quantity, pricing details, and bidding-related time parameters. The inspection section offers insights into crop quality, with fields for inspection results and dates, alongside farmer details. Buyer information captures essential data about participants, such as their identification, contact details, and locations. The bid details section logs individual bids, tracking bid IDs, prices, and bid placement dates, along with associated shipping and payment information. Finally, the shipping and payment section outlines the status of shipping processes, payment IDs, amounts, and payment dates. This data dictionary serves as a comprehensive reference guide, ensuring clarity and understanding of the database structure and its vital components within the agricultural trading platform.

Final Year Project Report

Stellar Scholar



Project Advisor: Fasiha Ashraf

Submitted By:

Shameer Zeeshan (S2020266018)

Muhammad Makki (S2020266027)

Saad Ahmad (F2019266069)

Farheen (F2019266073)

Session

S2020-F2023

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

Stellar Scholar is driven by a deep-rooted commitment to propel Pakistan forward in the space race. Inspired by childhood memories of watching science fiction movies, a seed to understand the mysteries of the cosmos was planted, and we went on to explore the development of space exploration technology in our country. These results revealed a sobering reality – Pakistan is lagging, falling decades behind in the global space race.

Motivated by the desire to bridge this gap, we embarked on a mission to make Pakistan self-sufficient in space exploration. Stellar Scholar stands as a significant milestone in Phase Four of Stellar Exploration Technologies. This app plays a vital role in nurturing the curious minds of our young generation, acting as a catalyst to ignite their passion and curiosity in the captivating field of rocket science. Our app aims to build passion and enthusiasm in teenagers, where they would be encouraged to dive deeper into the realm of rocketry and space exploration.

Our dedication to building such an app is directly linked with capturing the curiosity of teenagers and inspiring them to pursue careers in astrophysics, rocket science, and other interconnected disciplines. The primary goal is to empower a wave of scientists. Our primary goal is to empower the next wave of talented scientists and engineers, which would in turn allow Pakistan to build a self-sustaining ecosystem. By doing so, we aspire to establish a self-sustaining ecosystem that will propel Pakistan right beside the leading nations of the world.

Stellar Scholar is not just an app; it represents a collective dream to propel Pakistan to new heights in the space race. It embodies our unwavering commitment to nurturing scientific curiosity, empowering the youth, and fostering self-sufficiency in space exploration.

Final Approval

- **Head of Department**

Department of Computer Science
School of Systems & Technology
UMT Lahore

- **Program Director (Final Year Projects)**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Supervisor**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Co-Supervisor**

Acknowledgment

We would like to express our heartfelt gratitude to everyone who has contributed to the start of the development and realization of Stellar Scholar. This app will not be possible without the dedication, expertise, and support of our instructors. They went above and beyond while guiding us in every aspect of this app.

We would like to extend our sincere gratitude to our esteemed supervisors, Dr. Muzzammil Hussain and Khawaja Ubaid Ur Rehman, for their invaluable guidance, support, and mentorship throughout the development of Stellar Scholar. Their expertise, encouragement, and unwavering dedication have been instrumental in shaping this app into a comprehensive and impactful learning platform.

We would like to acknowledge the University of Management and Technology (UMT) for providing us with the platform and resources to pursue this project. The educational environment and opportunities offered by UMT have been instrumental in our growth and learning.

This app will be a testament to the collaborative effort of an exceptional team, guided by the expertise and support of our supervisors. We are proud to have had the opportunity to work under their mentorship, and we are confident that Stellar Scholar will make a positive impact in the field of space education.

Project Title: Stellar Scholar

Objective

- To explain a unique science to teenagers
- Make them learn the basics of rocketry
- Allow students to think about other career choices
- Pave the way for actual rocket development in the Country

Undertaken by

- Shameer Zeeshan (S2020266018)
- Muhammad Makki (S2020266027)
- Saad Ahmad (F2019266069)
- Farheen (F2019266073)

Supervised by: Dr. Fasiha Ashraf

Starting Date: 27/March/2023

Completion Date: In-Progress

Tools Used

- Unity Engine (For designing the front end)
- Visual Studio (For writing Scripts in C#)
- Blender (For making Models)
- Figma (For UI/UX designing)
- Adobe Illustrator (For UI designing)
- Adobe Photoshop (For Editing)

Operating System: Android

Documentation: Version 1

Plagairism Report

Declaration Form

I have carefully examined the documentation of the Final Year Project titled “*Stellar Scholar*”; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with the user manual and installation guide.

FYP Advisor Name: _____

Signature: _____

Date: _____

Abstract

Stellar Scholar is an extraordinary learning app that takes users on an exciting journey into the captivating realm of rocketry and space sciences. Its immersive gameplay, stunning 3D visuals, and comprehensive educational content combine to create a transformative learning experience for individuals of all ages. With its visually captivating low-poly art style, Stellar Scholar transports users into a virtual rocket lab where they can embark on a voyage of discovery. The app offers two distinctive sections: a learning component and an endless game mode, both designed to complement each other seamlessly. The learning section of Stellar Scholar provides users with an interactive platform to explore the fundamental concepts of rocketry through engaging quizzes and assessments. With a familiar face-on view, reminiscent of traditional learning apps, users can progress through levels, earn rewards, and unlock new content as they showcase their knowledge and comprehension. The thoughtfully curated quizzes serve as checkpoints, offering valuable feedback and reinforcing key principles for effective learning. Complementing the learning section, the game mode in Stellar Scholar allows users to apply their newfound knowledge in a hands-on and thrilling manner. Adopting a third-person perspective, users step into the shoes of rocket scientists, designing, building, and launching their rockets into the virtual skies. Intuitive joystick controls enable precise navigation, while tap-based interactions allow users to select and assemble rocket components, unleashing their creativity and exploring endless possibilities in rocket design. The dynamic camera angles capture breathtaking moments as rockets soar through the atmosphere, immersing users in a truly exhilarating experience. However, Stellar Scholar goes beyond being just a game or a learning app. It fosters a vibrant community of space enthusiasts, providing a platform for connection, collaboration, and knowledge sharing. Users can engage with like-minded individuals, sharing their rocket designs, exchanging strategies, and staying updated with the latest news and advancements in the realm of rocketry. The app's chat rooms and forums create a nurturing space for meaningful discussions and the cultivation of a supportive community. The social impact of Stellar Scholar extends far beyond the app itself. It aspires to inspire the next generation of scientists, engineers, and space explorers, particularly in regions where resources for space exploration are limited, such as Pakistan. By nurturing curiosity and providing users with the knowledge and skills to pursue careers in rocketry, Stellar Scholar aims to ignite a passion for space exploration and drive progress in the global space race. In summary, Stellar Scholar is poised to revolutionize the way we learn about rocketry and space sciences. Through its immersive gameplay, stunning visuals, and comprehensive educational content, the app aims to inspire and empower users, fuel curiosity, and pave the way for future advancements in the field. By unlocking the wonders of rocketry and instilling a sense of wonder, Stellar Scholar propels us toward a future where the boundaries of space exploration are pushed beyond imagination.

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
<i>Draft</i>	TBD	Initial draft created for distribution and review comments	(To be decided) TBD
<i>Preliminary</i>	TBD	A second draft incorporating initial review comments, distributed for final review	TBD
<i>Final</i>	TBD	First complete draft, which is placed under change control	TBD
<i>Revision 1</i>	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
<i>Revision 2</i>	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
<i>Etc.</i>	TBD	TBD	TBD

CONTENTS

CONTENTS	IX
DEFINITIONS AND ACRONYMS.....	XI
LIST OF FIGURES.....	XII
LIST OF TABLES	XIII
1. INTRODUCTION	1
1.1 GAME OVERVIEW	1
1.2 TARGET PLATFORM(S).....	1
1.3 BUSINESS MODEL.....	1
1.4 THEME / SETTING / GENRE	2
1.5 CORE GAMEPLAY MECHANICS	4
1.5.1 <i>Joystick</i>	4
1.5.2 <i>Tap</i>	5
1.5.3 <i>Drag</i>	5
1.6 SERVER / ONLINE MECHANICS	5
1.6.1 <i>Weekly Leaderboard</i>	5
1.7 ART STYLE.....	5
1.7.1 <i>3D</i>	5
1.7.2 <i>Low Poly</i>	6
1.8 LOOK & FEEL/ CAMERA	6
1.8.1 <i>Third Person</i>	6
1.8.2 <i>Face On</i>	7
1.9 PROGRESSION	7
1.9.1 <i>Level Based</i>	7
1.9.2 <i>Endless</i>	8
2. STORY AND GAMEPLAY	9
2.1 STORY.....	9
2.2 CORE GAMEPLAY	9
2.2.1 <i>Learning Phase</i>	9
2.2.2 <i>Upgrade and Customization</i>	9
2.2.3 <i>Rocket Construction</i>	10
2.2.4 <i>Launch and Exploration</i>	10
2.2.5 <i>Performance Evaluation</i>	10
3. DOMAIN ANALYSIS	11
3.1 STAKEHOLDERS.....	11
3.2 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT	11
3.2.1 <i>Students and Teenagers:</i>	11
3.2.2 <i>Parents and Guardians:</i>	11
3.2.3 <i>Educators and Schools:</i>	11
3.2.4 <i>Space Enthusiast Community:</i>	11
3.2.5 <i>Society and Country:</i>	11
3.2.6 <i>App Publishers and Developers:</i>	12
3.2.7 <i>Job Creation:</i>	12
3.2.8 <i>Educational Institutions:</i>	12
3.2.9 <i>Industries related to space:</i>	12
3.2.10 <i>Entrepreneurship and innovation:</i>	12

3.3	DEPENDENCIES/ EXTERNAL SYSTEMS	12
3.4	RELATED PROJECTS	14
3.4.1	<i>Kerbal Space Program:</i>	14
3.4.2	<i>Space Flight Simulator:</i>	14
3.4.3	<i>ISS Docking Simulator:</i>	14
3.4.4	<i>Build Your Rocket:</i>	14
3.4.5	<i>Khan Academy:</i>	14
3.4.6	<i>Duolingo:</i>	14
3.4.7	<i>Photomath:</i>	15
3.4.8	<i>Feature Comparison</i>	15
4.	REQUIREMENTS ANALYSIS.....	17
4.1	REQUIREMENTS.....	17
4.2	LIST OF ACTORS.....	18
4.2.1	<i>User</i>	18
4.2.2	<i>Developer</i>	18
4.3	LIST OF USE CASES.....	18
4.3.1	<i>Authentication</i>	18
4.3.2	<i>Login</i>	18
4.3.3	<i>Signup</i>	18
4.3.4	<i>Main menu</i>	18
4.3.5	<i>New Game</i>	19
4.3.6	<i>Resume</i>	19
4.3.7	<i>Setting</i>	19
4.3.8	<i>Leaderboard</i>	19
4.3.9	<i>Playing mode</i>	19
4.3.10	<i>Inventory</i> :	19
4.3.11	<i>Build your Rocket</i> :	19
4.3.12	<i>Quiz</i> :	19
4.3.13	<i>Learning tutorials</i>	19
4.3.14	<i>Challenges</i>	19
4.3.15	<i>Chat System</i>	19
4.3.16	<i>Achievements</i>	19
4.3.17	<i>Credits</i>	19
4.4	SYSTEM USE CASE DIAGRAM	20
4.5	EXTENDED USE CASES.....	21
5.	USER INTERFACE/ SCREENS.....	34
5.1	MAIN MENU	34
5.2	SETTING	35
5.3	LOADING.....	36

Definitions and Acronyms

Table 0.1: List of acronyms and definitions

Acronym	Definition
UMT	University of Management and Technology
POS	Point of Sale
SFS	Space Flight Simulator
ISS	International Space Station

List of Figures

Figure 1.1: Kerbal Space Program Build	2
Figure 1.2: Kerbal Space Program Gameplay	2
Figure 1.3: SpaceX ISS Docking Simulator.....	3
Figure 1.4: Space Flight Simulator.....	3
Figure 1.5: Space Flight Simulator.....	3
Figure 1.6: Build Your Own Rocket	4
Figure 1.7: Build Your Own Rocket	4
Figure 1.8: Onscreen Joystick	4
Figure 1.9: Tap Mechanic.....	5
Figure 1.10: Drag Mechanic	5
Figure 1.11: 3D Rocket Model	6
Figure 1.12: Low Poly Art Style	6
Figure 1.13: KSP Interface Example.....	6
Figure 1.14: Quiz System Example.....	7
Figure 2.1: Gameplay Loop.....	10
Figure 4.1: System Use-case Diagram	20
Figure 5.1 Main Menu User Interface	34
Figure 5.2 Setting User Interface.....	35
Figure 5.3 Loading Screen	36

List of Tables

Table 0.1: List of acronyms and definitions	XI
Table 3.1 List of Stakeholders.....	11
Table 3.2: Feature Comparison With Existing Games	15
Table 3.3: Feature Comparison With Existing Learning Apps	16
Table 4.1: Functional Requirements.....	17
Table 4.2: Non-Functional Requirements	18
Table 4.3 Extended Use Case 1	21
Table 4.4 Extended Use Case 2	22
Table 4.5 Extended Use Case 3	23
Table 4.6 Extended Use Case 4.....	24
Table 4.7: Extended Use Case 5	24
Table 4.8:Extended Use Case 6.....	25
Table 4.9:Extended Use Case 7	26
Table 4.10: Extended Use Case 8	26
Table 4.11: Extended Use Case 9	27
Table 4.12: Extended Use Case 10.....	28
Table 4.13: Extended Use Case 11.....	28
Table 4.14: Extended Use Case 12.....	29
Table 4.15: Extended Use Case 13	30
Table 4.16: Extended Use Case 14.....	31
Table 4.17: Extended Use Case 15.....	31
Table 4.18: Extended Use Case 16.....	32
Table 4.19: Extended Use Case 17.....	33

1. INTRODUCTION

1.1 Game Overview

Stellar Scholar is an education-based game whose main concept revolves around inspiring children to study and explore rocket and space sciences. This game plans to intrigue young minds to think outside the horizons of traditional sciences. Children usually tend to learn more through games. It keeps their thought process alive. Making them think about the purpose of life at a young age so that they could one day become productive and leading members of society, Pakistan is a country that is lagging in the space race by the day. SUPARCO, which was formed years before ISRO, is now decades behind in terms of technology, mainly because of a lack of resources. If we let it go as it is we will suffer dire consequences. Our app is being developed with the hope that one day these children could become astrophysicists, rocket scientists, and much more, overcoming the lack of resources and opening doors to a new era.

This app will cover the concepts of exploration and life beyond our planet. The main problem with that idea is that to explore we need transport and for a sustainable transport system we need rockets which in turn need rocket scientists. It is hoped that this game paves the way for children to think about the world differently and make the science fiction they see in movies a reality. This game would also build a community of space-enthusiast children that could share their thoughts and talk about their rocket designs in the game. Newsletters would be shared covering the news of the week in related fields.

This app would be a combination of news service, chat rooms, and games that together will form a community that would hopefully jumpstart this country's space program.

1.2 Target Platform(s)

- Stellar Scholar will be supported by Android 10 & above.
- System Requirements would be 2GB of RAM.
- The game will be in Landscape mode.

1.3 Business Model

Stellar Scholar's business model revolves around a freemium structure that incorporates in-app purchases, ad placements, and supplementary educational opportunities. Our app would be free to download through which we aim to achieve inclusivity and accessibility. The revenue stream would begin from in-app purchases which will allow the players to enhance their capabilities to modify their rockets through different means through the in-app store. These optional purchases provide a revenue stream while enabling users to customize their journey.

Ads will play a role in the business model, serving as an additional source of revenue. Thoughtfully integrated into the app, these ads will be seamlessly incorporated to preserve the user experience and generate income.

To further expand revenue streams and provide added value, Stellar Scholar plans to introduce paid educational offerings. Users who reach specific milestones within the game will have the opportunity to participate in exclusive seminars and classes for a fee. Led by subject matter experts, these sessions will delve into deeper aspects of rocket and space sciences, allowing users to gain comprehensive insights.

This business model will allow us to scale our app, enabling us to add the latest content, upgrades, and rewards to keep the community engaged and interested.

1.4 Theme / Setting / Genre

The main theme of Stellar Scholar is centered around a Rocket Lab, where players immerse themselves in the process of developing rocket engines. The game introduces players to the fundamentals of each element involved in rocket engine construction, guiding them to understand their roles and interactions. Through a combination of acquired knowledge and strategic thinking, players engage in assembling these elements to create a fully functional and efficient rocket engine. The game's setting, depicted in visual figures, showcases the immersive Rocket Lab environment, providing players with a realistic and engaging experience. Within this setting, players navigate through various stages of development, from grasping the basics of each element to the final integration and testing phase.

To assess and reinforce player progress, the game incorporates a quiz system that serves as a testing period. By completing quizzes, players advance in the game, unlocking new elements, challenges, and opportunities for further development.

The theme of Stellar Scholar combines education and gameplay, offering an interactive and dynamic experience for players to explore the intricacies of rocket engine development. By combining learning with hands-on application, the game motivates players to acquire knowledge, hone their skills, and progress through the engaging Rocket Lab setting.



Figure 1.1: Kerbal Space Program Build

Figure 1.1 depicts the scene of a rocket lab where the player builds their rocket as discussed above.



Figure 1.2: Kerbal Space Program Gameplay

Figure 1.2 represents the launchpad from where the created rocket would be launched into space.



Figure 1.3: SpaceX ISS Docking Simulator

Figure 1.3 displays the interface of the SpaceX ISS docking simulator and how the player controls the Dragon Capsule to dock with the space station. It mimics the actual maneuver that is performed in orbit.

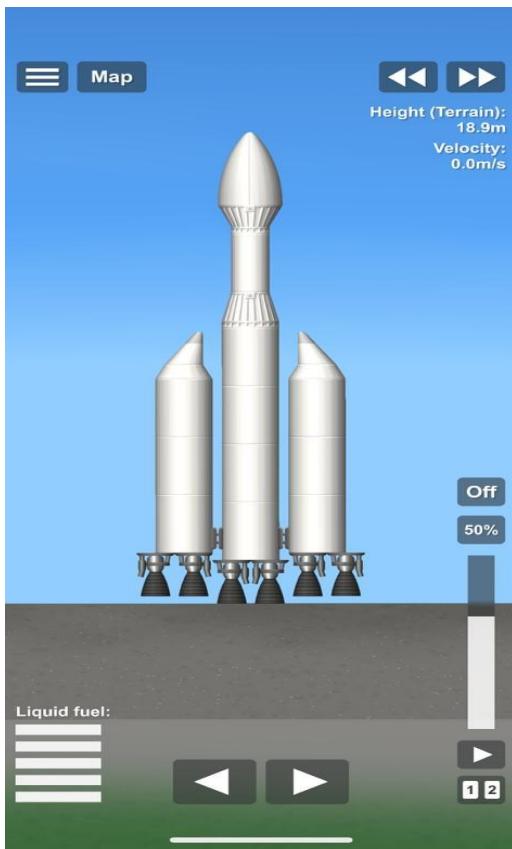


Figure 1.4: Space Flight Simulator

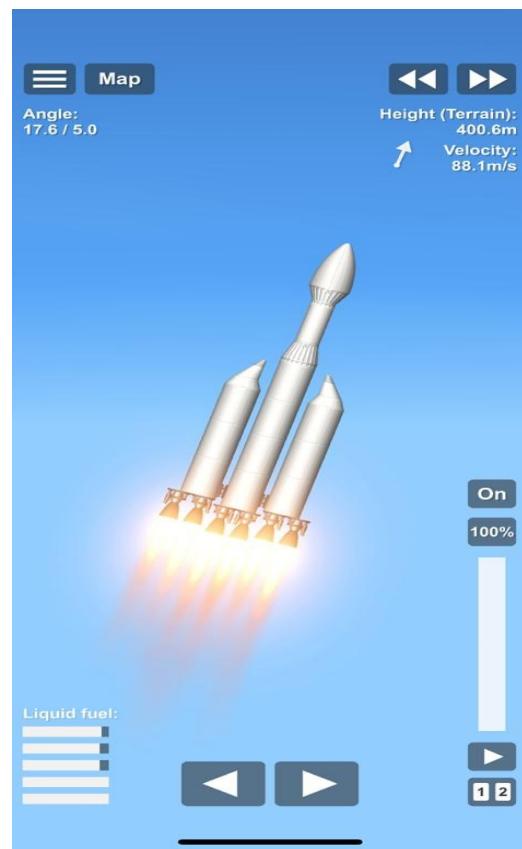


Figure 1.5: Space Flight Simulator

Figures 1.4 & 1.5 are the gameplay of SFS in which a user can build and fly their rockets into orbit using the steering buttons they can increase speed using the slider on the right.



Figure 1.6: Build Your Own Rocket

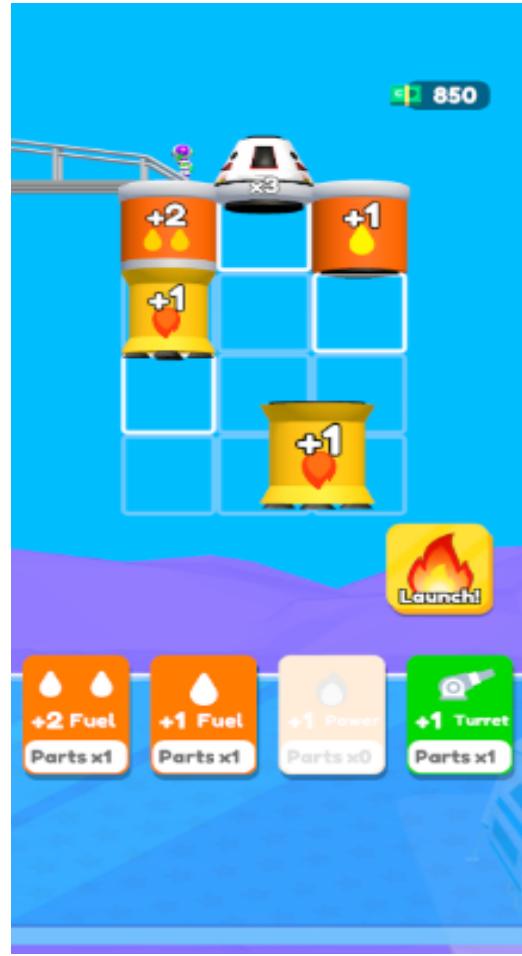


Figure 1.7: Build Your Own Rocket

Figures 1.6 and 1.7 show the interface of a game named Build Your Rocket here the user builds their rocket using the tiles available and then embarks on a journey to explore new worlds.

1.5 Core Gameplay Mechanics

1.5.1 Joystick

A virtual joystick will be displayed on the screen, which would allow the players to control and navigate the rocket throughout the gameplay as shown in Figure 1.8. This on-screen joystick provides a user-friendly interface for players to precisely control the rocket's movements and overcome various challenges encountered during space exploration. In addition to the joystick, other mechanics will complement the gameplay, offering players a diverse and engaging experience throughout their rocketry journey in the game.

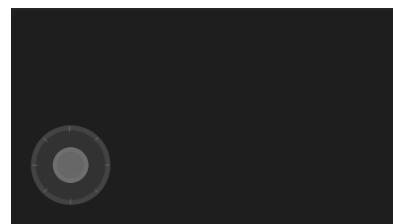


Figure 1.8: Onscreen Joystick

1.5.2 Tap

The selection of specific options or objects within the game will be based on tap interactions. Players can simply tap on the desired object or option on the screen to make a selection as shown in Figure 1.9. Whether it's selecting a rocket component for customization, interacting with a mission objective, or accessing various menu options, a simple tap gesture will allow players to make their choices swiftly and intuitively. This tap-based selection mechanic provides a user-friendly and responsive way for players to navigate and interact with the game's interface, enhancing their overall gameplay experience.

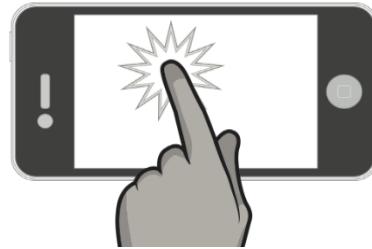


Figure 1.9: Tap Mechanic

1.5.3 Drag

During the construction phase of Stellar Scholar, the dragging mechanic will be utilized to arrange rocket parts which are portrayed in Figure 1.10. Players will have an inventory of various rocket components, and to assemble their rocket, they can drag and drop the desired parts from the inventory to the construction site. By dragging the selected item with their finger or mouse, players can position and place the components in the desired locations within the rocket structure. This drag-based construction mechanic provides a hands-on and interactive experience, allowing players to have precise control over the assembly process and create unique rocket designs.



Figure 1.10: Drag Mechanic

1.6 Server / Online Mechanics

1.6.1 Weekly Leaderboard

This app will be connected to Facebook to save the player's progress and it will update the top player rank and score every week through this players will be able to connect with their friends. It will be a great point to add interest and enjoyment for players. Later, we plan to include other features to save progress.

1.7 Art Style

1.7.1 3D

The rocket models will be designed in 3D as shown in Figure 1.11, allowing for detailed and realistic representations of different types of rockets. Various components such as stages, boosters, engines, and other intricate details to make the rockets visually appealing and accurate would be incorporated. The surrounding terrain, specifically the launch site, will also be developed using 3D art. Buildings, launch pads, control towers, and other structures commonly found at rocket launch sites would be covered in the 3D modeling. The game terrain will be designed in such a manner that it would represent a realistic scenario including vegetation and man-made features.

1.7.2 Low Poly

The rocket models, terrain, and buildings will be represented with a minimal number of polygons, resulting in a simplified and geometric look. Instead of intricate and detailed shapes, the emphasis is on creating recognizable forms using fewer polygons as shown in Figure 1.12. This stylistic choice gives the objects a distinct and stylized appearance.

By combining the low poly art style with the 3D nature of your app, we can achieve a balance between visual simplicity and a realistic representation. The low poly models for rockets, surrounding terrain, and buildings at the launch site will retain their recognizable shapes while providing depth through careful shading and lighting techniques.



Figure 1.11: 3D Rocket Model



Figure 1.12: Low Poly Art Style

1.8 Look & Feel/ Camera

1.8.1 Third Person

In the game section of Stellar Scholar, the perspective and camera view will be set to the third person. This means that the user will have an external view of the rocket and its surroundings, allowing for a more immersive experience as shown in Figure 1.13.

Users will have the ability to drag and align rocket parts in a three-dimensional space. The third-person perspective will provide a comprehensive view of the rocket as it takes shape, allowing users to visualize and adjust the positioning of components for optimal functionality. As the rocket moves through the atmosphere, the angle of the camera will dynamically adjust to follow its trajectory. This dynamic camera movement will provide users with different perspectives and viewpoints, enhancing the sense of immersion and realism.



Figure 1.13: KSP Interface Example

1.8.2 Face On

In the learning app section of Stellar Scholar, the display will feature a face-on view. This means that the user interface will present information and quiz sections in a straightforward and easily accessible manner as shown in Figure 1.14. The quiz sections will be designed to resemble a typical learning app interface. Users will be presented with questions and provided with multiple-choice options. The face-on view will ensure that the questions and options are visible and easy to read, facilitating a seamless learning experience. Alongside the quizzes, the face-on view will also present learning resources such as tutorials, explanations, and additional educational materials. These resources will be easily accessible and organized in a user-friendly manner, enabling users to explore and expand their knowledge of rocketry and space sciences.

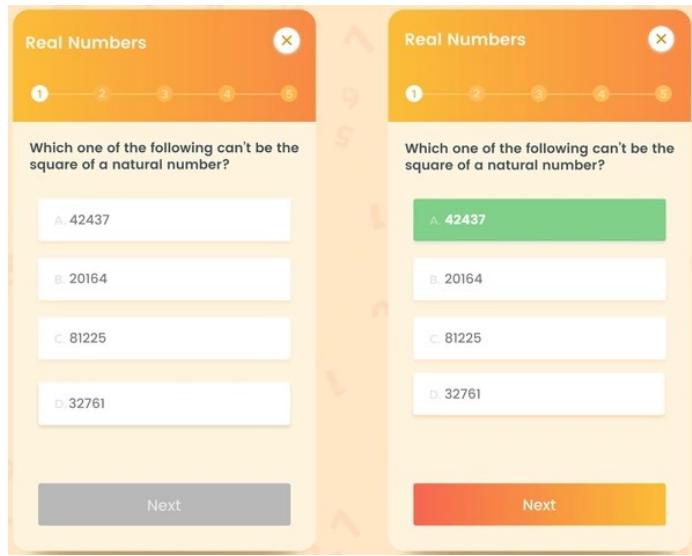


Figure 1.14: Quiz System Example

resources such as tutorials, explanations, and additional educational materials. These resources will be easily accessible and organized in a user-friendly manner, enabling users to explore and expand their knowledge of rocketry and space sciences.

1.9 Progression

1.9.1 Level Based

On the learning side of Stellar Scholar, the game progression follows a level-based structure. As users advance through the levels, they will encounter quizzes that serve as tutorials, providing explanations and key insights into various aspects of rocketry and space sciences. The player assessment begins with the initial tutorial, where users demonstrate their understanding of the fundamental concepts. As they continue to progress, new and more advanced concepts will be introduced, allowing players to expand their knowledge and skills.

Each level will feature quizzes that test the users' comprehension and application of the learned concepts. These quizzes serve as checkpoints for their progress and offer a means to assess their proficiency. By completing the quizzes, players unlock further levels and gain access to more challenging content.

Importantly, the progress made in the quizzes will be rewarded in the game part of Stellar Scholar. The rewards earned from the quizzes, such as points, or in-game items, can be utilized to enhance the player's rocket-building capabilities, unlock new features, or access additional game content. By implementing a level-based progression system for the learning side of the app, users will be motivated to continually improve their knowledge and skills. The quizzes act as educational checkpoints, ensuring that users grasp the foundational concepts before moving on to more advanced topics. This approach provides a structured and rewarding learning experience, keeping users engaged and motivated throughout their journey with Stellar Scholar.

1.9.2 Endless

In the endless game part of Stellar Scholar, the primary objective is to provide users with an immersive experience that enables them to grasp the core dynamics of rocketry. As users progress in the learning app, they will earn upgrades that can be implemented in the game section.

The game revolves around the construction and testing of rockets. Users will have the opportunity to build their rockets, incorporating the knowledge and concepts they have acquired through the learning app. The rocket construction process will allow for customization and experimentation, as users select different components and configurations based on their understanding of rocket science.

Once the rockets are built, users will embark on exciting journeys through the atmosphere. The game mechanics will involve launching the rockets, achieving orbit, and performing various maneuvers. Users will have the chance to experience the challenges and intricacies of spaceflight firsthand as they navigate through different stages of the game.

The endless nature of the game ensures that users can continue exploring and experimenting with different rocket designs and flight strategies. There is no predetermined endpoint, allowing for continuous learning and improvement. Users can set their own goals, whether it's reaching higher altitudes, perfecting orbital maneuvers, or attempting complex space missions. The integration of upgrades earned in the learning app further enhances the gameplay experience. Users can apply their acquired knowledge to enhance their rockets, enabling them to achieve better performance, higher altitudes, and more advanced maneuvers.

2. STORY AND GAMEPLAY

2.1 Story

Rocketry has a captivating history that spans several centuries. It all begins with the use of fire arrows by the Chinese and the Mongols in the 1200s. However, it wasn't considered a scientific field until the late 1700s when Sir Isaac Newton laid its foundation. In the early phases of the late 19th century and early 18th century, rockets were only a means of mass destruction.

Konstantin Tsiolkovsky, a Russian school teacher introduced the groundbreaking concept of space exploration using rockets. It was his vision that changed rocketry forever. His revolutionary idea ignited a sense of possibility and awe, capturing the dreams and aspirations of people worldwide.

Welcome, Stellarites, to the extraordinary realm of space exploration. Prepare yourself for an exceptional journey that will challenge your understanding and ignite your imagination. As you delve into the depths of rocketry and space sciences, you will uncover the enigmatic secrets of the universe while discovering your boundless potential.

Embark on an extraordinary journey as your adventure commences here, where you will be introduced to an immersive game and the captivating story of rocketry. Prepare yourself for an exhilarating and enlightening expedition, as you explore the lives of innovative pioneers who have profoundly influenced the course of rocketry. Witness the astounding accomplishments that have propelled humanity beyond the confines of our planet, and with each step, unravel the enigmatic wonders of the cosmos and behold the remarkable progression of rocketry, from its modest beginnings to its awe-inspiring pinnacle. Brace yourself for an experience that will leave you in awe at every turn of this remarkable voyage.

Are you ready to indulge yourself in a journey through the vastness of space, driven by knowledge and fueled by curiosity? The adventure awaits, Stellarites. Let us embark together on this awe-inspiring journey.

Please note that this storyline would be conveyed in a cut scene at the beginning of the game, introducing the users to the captivating world of rockets. As our application is designed to facilitate user learning through gamification, this narrative effectively encourages exploration and engagement with the fascinating realm of rocketry as described above.

2.2 Core Gameplay

2.2.1 Learning Phase

Users engage in the learning app section, where they progress through levels and quizzes. They are presented with educational content, tutorials, and quizzes to test their understanding of rocketry and space sciences. Successful completion of quizzes rewards them with points or in-app items.

2.2.2 Upgrade and Customization

Users utilize the rewards earned from the learning phase to unlock upgrades, components, and skins for their rockets. They can customize their rockets with different parts and configurations based on their acquired knowledge.

2.2.3 Rocket Construction

Using drag-and-drop mechanics, users assemble their rockets in a three-dimensional space. They align and position the rocket parts to create a functional and optimized design, taking into account the principles of rocket science.

2.2.4 Launch and Exploration

Users launch their constructed rockets into the atmosphere and beyond. They navigate the rocket through the 3D environment, controlling its trajectory and responding to challenges such as gravitational forces and atmospheric conditions.

2.2.5 Performance Evaluation

Performance is evaluated based on altitude, speed, fuel efficiency, and maneuverability. These metrics give feedback to the users allowing them to better understand and refine their designs for future launches.

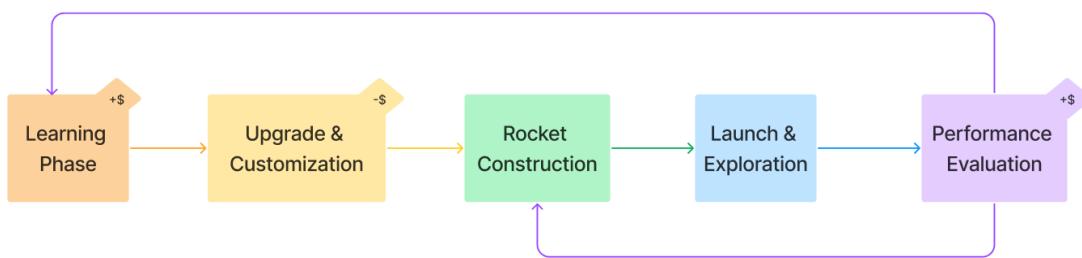


Figure 2.1: Gameplay Loop

Figure 2.1 depicts the entire gameplay loop of Stellar Scholar. It visualizes all the processes explained above, how the learning phase is connected to the upgrade and customization, how from there we can access the rocket construction feature which is linked to the actual game of launching and exploring the space around it, and how afterward our performance is evaluated, and we can go back to the rocket construction or learning phase.

3. DOMAIN ANALYSIS

3.1 Stakeholders

Table 3.1 List of Stakeholders

Stakeholder	Role in System
<i>Animator</i>	<i>Design, and animate all characters, and provide the knowledge and skills to perform the animation</i>
<i>Developer Team</i>	<i>Writing Scripts and implementing the mechanics in the game</i>
<i>Designer</i>	<i>Designing the friendly user interface by designing the UI elements that will interact with the player making the game more enjoyable and intuitive</i>
<i>End Users</i>	<i>Provide Feedback, participate in Beta testing, and build community</i>
<i>Supervisor</i>	<i>Overseeing the entire project by leading the Whole Team</i>

Table 3.1 shows all the stakeholders involved in the development of Stellar Scholar alongside their roles in the system.

3.2 Affected groups with social or economic impact

3.2.1 Students and Teenagers:

Stellar Scholar is like a magical gateway that captures children's hearts and minds with its captivating gaming experience and learning, inspiring them to dream big and pursue awe-inspiring careers in the field of space exploration.

3.2.2 Parents and Guardians:

Stellar Scholar would be a priceless gift for parents, offering them the interest in developing a tool to support their child's educational journey. It becomes a shared adventure where parents engage with their children by doing nurturing conversations about the mysteries of space. Stellar Scholar would raise the general acumen of students as well as their parents.

3.2.3 Educators and Schools:

Stellar Scholar becomes an invaluable platform for educators and schools, providing them with an innovative way to learn science education. By seamlessly integrating the app into classroom activities, educators ignite a spark of curiosity in their students, transforming mundane lessons into thrilling discoveries.

3.2.4 Space Enthusiast Community:

Stellar Scholar aspires to be more than just an app; it aims to build a vibrant community of kind spirits, united by a shared passion for space sciences.

3.2.5 Society and Country:

The secret to change not only individual lives but also entire civilizations and countries lies with the Stellar Scholar. The software transforms into a ray of hope for a better future by inspiring young minds and fostering their aspirations to become astrophysicists, rocket scientists, and leaders in the space industry.

3.2.6 App Publishers and Developers:

The growth and popularity of Stellar Scholar may open up business prospects for publishers and developers of mobile applications. The app has the potential to make money through a variety of channels, like in-app purchases, subscriptions, and in-app advertising, as it attracts users and grows in popularity. The developers and publishers may benefit financially from the app's commercial success, enabling them to increase their product offers or reinvest in the creation of new apps.

3.2.7 Job Creation:

Stellar Scholar's expansion and sustainability may result in the creation of jobs across several industries. There can be a need for qualified individuals in fields like software development, visual design, content creation, marketing, and customer support if the app gets traction. Individuals with knowledge in these sectors may be able to find employment as a result, boosting the local and global economy.

3.2.8 Educational Institutions:

Stellar Scholar adoption at educational institutions may be financially advantageous to the institutions themselves. The software may be incorporated into the curriculum of schools and other educational institutions, improving the standard of science instruction. This can therefore draw in pupils and parents who value cutting-edge, technologically enhanced learning opportunities, thus increasing enrollment and ensuring the financial stability of educational institutions in regional and governmental economies.

3.2.9 Industries related to space:

Stellar Scholar's focus on encouraging young people to pursue careers in space sciences may have a long-term positive impact on the economy of the space industry. The software can help create a qualified workforce by fostering a desire for space exploration in industries like astrophysics, rocket science, engineering, and related ones. As a result, there may be more innovation, research, and development carried out in the nation through enhancing the capacities of the space-related sectors. The expansion of these businesses has the potential to increase employment opportunities and the overall economy.

3.2.10 Entrepreneurship and innovation:

By emphasizing the development of scientific thinking and problem-solving abilities, the app can promote an entrepreneurial and innovative culture. Through their interaction with Stellar Scholar, kids, and students may grow to have an entrepreneurial attitude, which will inspire their creativity and push them to think outside the box. This could result in the birth of fresh entrepreneurs, businesses, and projects.

3.3 Dependencies/ External Systems

In developing this application, we require the following tools:

- Unity Engine (For Front-End)
- Visual Studio (For Scripting)
- Blender (For 3D Models)
- Figma (For UI/UX designing)
- Adobe Illustrator (For UI designing)
- Adobe Photoshop (For Editing)
- Copyright free Sounds
- Personal Computers
- Target Audience

3.4 Related Projects

While researching our domain we analyzed many apps, their details are given below:

3.4.1 Kerbal Space Program:

Kerbal Space Program (KSP) is a popular space flight simulation game developed and published by Squad. It allows players to design, build, and fly their spacecraft and explore a fictional solar system called the Kerbol System. The primary objective of KSP is to build functional spacecraft and complete various missions, such as launching satellites, conducting scientific experiments, exploring other planets and moons, and even establishing space stations and colonies. You can find it at: <https://www.ckarbalaspaceprogram.com/games-kerbal-space-program-2>

3.4.2 Space Flight Simulator:

Space Flight Simulator is a mobile game that simulates and experiences the challenges of space exploration. Players have the opportunity to build and launch their rockets, explore the solar system, and complete various missions. The game offers a sandbox mode where players can freely experiment and create their missions, as well as a career mode with preset missions and objectives. You can find it at: https://store.steampowered.com/app/1718870/Spaceflight_Simulator/

3.4.3 ISS Docking Simulator:

ISS Docking Simulator is a virtual simulation game that allows players to experience the challenges and intricacies of docking spacecraft with the International Space Station (ISS). The game aims to replicate the real-life procedures and techniques used by astronauts during docking missions. The game provides a realistic physics simulation, which means players need to consider factors like gravity, momentum, and orbital mechanics to complete the docking process. You can find it at: <https://iss-sim.spacex.com/>

3.4.4 Build Your Rocket:

Build Your Rocket is a 2d animistic game in which a user can build his rocket by combining the tiles and exploring different planets on that rocket. Start and fly to the farthest planets and reach to end of the galaxy if you can! The higher you go with your rocket to different planets, the better you score and reach new levels. You must beat your competition and fly to all planets before they do. You can find it at: <https://www.crazygames.com/game/build-your-rocket>

3.4.5 Khan Academy:

Khan Academy is a comprehensive online learning platform that offers an extensive range of subjects. It provides video lessons, practice exercises, quizzes, and interactive tools to support learning in math, science, history, economics, computer programming, and more. Khan Academy stands out for its self-paced learning approach and serves as a valuable resource for learners of all ages. You can find it at: <https://www.khanacademy.org/>

3.4.6 Duolingo:

Duolingo is a highly popular app that makes language learning enjoyable and accessible to users. With its gamified approach, Duolingo offers interactive lessons, vocabulary exercises, and speaking practice for various languages. It is known for its user-friendly interface and offers courses suitable for learners of all levels. You can find it at: <https://www.duolingo.com/>

3.4.7 Photomath:

Photomath is a helpful app designed to assist users in solving math problems step-by-step. By utilizing the device's camera, users can scan handwritten or printed math problems, and Photomath provides instant solutions along with detailed explanations. It covers a wide range of math topics, making it beneficial for students seeking a better understanding of math concepts. You can find it at: <https://photomath.com/en>

3.4.8 Feature Comparison

Table 3.2: Feature Comparison With Existing Games

<i>Feature/Aspect</i>	<i>STELLAR SCHOLAR</i>	<i>KERBAL SPACE PROGRAM</i>	<i>SPACEFLIGHT SIMULATOR</i>	<i>ISS SPACE SIMULATOR</i>	<i>BUILD YOUR ROCKET</i>
HARDWARE	2GB RAM	4GB RAM	2GB RAM	NA	3GB RAM
SOFTWARE	ANDROID	PS/PC/ XBOX	IOS/PC/ ANDROID	WEB	WEB/ ANDROID
PROGRESSION	LEVEL & ENDLESS	ENGAGEMENT BASED	FREE WORLD	MISSION-BASED	SCORE BASED
GRAPHIC VISUALS	3D LOW POLY REALISTIC	3D HIGH POLY REALISTIC	2D LOW POLY	3D REALISTIC	2D ANIMATED
GAMEPLAY MECHANICS	JOYSTICK/ TAP/DRAG	JOYSTICK	DRAG/TOUCH	ARROW KEYS	TAP/DRAG
PERFORMANCE & OPTIMIZATION	60 FPS	UPTO 120 FPS	UPTO 144	30 FPS	20 FPS
REALISM	✓	✓	✗	✓	✗
CUSTOMIZATION	✓	✓	✓	✗	✗
IN-APP PURCHASES	✓	✓	✗	✗	✗
IN-APP LEARNING	✓	✗	✗	✗	✗
LEADERBOARD	✓	✗	✗	✗	✗
CHAT SYSTEM	✓	✗	✗	✗	✗

In Table 3.2 we have compared the game element of our app with some existing games. This feature comparison shows our game includes new features.

Table 3.3: Feature Comparison With Existing Learning Apps

Feature/Aspect	Stellar Scholar	Duolingo	Khan Academy	Photomath
Subject Focus	Rocketry and Space Sciences	Language Learning	Various Subjects	Science and Math
Language Support	English Only	Multiple languages	English only	Multiple languages
Learning Resources	Text, Images, Game	Text, Audio	Text, Images, Videos	Text, Images
Community Engagement	✓	✓	✓	✗
Level-Based Progression	✓	✓	✗	✗
Interactive Gameplay	✓	✗	✗	✗
Quiz and Assessment	✓	✓	✓	✓
Real-Time Feedback	✓	✓	✓	✓
3D Graphics	✓	✗	✗	✗
In-App Purchases	✓	✓	✗	✓
Progress Tracking	✓	✓	✓	✓

In Table 3.3 we have compared our game with existing learning apps. This feature comparison shows our app incorporates new features.

4. REQUIREMENTS ANALYSIS

4.1 Requirements

Table 4.1: Functional Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
FR1	User Registration	Functional Requirement	User credentials	Usernames must be unique, passwords should meet security criteria, and email addresses should be valid.
FR2	Tutorial and User Guide:	Functional Requirement	Interactive tutorial and user guide	The tutorial should be easily accessible and understandable for users of varying age groups and levels of knowledge.
FR3	Playing Mode	Functional Requirement	Clear selection of playing difficulty	The user should decide the mode he is using defining the complexity they are going to face.
FR4	Learning Modules and Quizzes	Functional Requirement	Assessment modules to track user progress	The content should be accurate, engaging, and appropriate for the target age group. Quizzes should have a suitable difficulty level and provide timely feedback.
FR5	Progress Tracking and Reporting	Functional Requirement	Progress tracking and reporting system	The tracking system should accurately reflect user progress, and the reporting should be clear and easily understandable by both users and educators.
FR6	Community Features	Functional Requirement	Community engagement tools	Implement appropriate moderation and user guidelines to ensure a safe and respectful community environment.
FR7	Rewards	Functional Requirement	Associated titles alongside achievements	The user should get rewards based on their gameplay using the correct multipliers.
FR8	In-App Purchases	Functional Requirement	In-app purchase functionality	Ensure compliance with payment industry standards and implement safeguards to protect user payment information.
FR9	Offline Mode	Functional Requirement	Offline functionality	The offline mode should have limitations and communicate which features are available offline and which require an internet connection.
FR10	Push Notification	Functional Requirement	Push notification system	Allow users to manage their notification preferences and ensure compliance with platform-specific guidelines for push notifications.

Table 4.1 shows all the functional requirements involved in the system.

Table 4.2: Non-Functional Requirements

NFR1	Gameplay	Non-Functional Requirement	Efficient Code	Ensure a smooth transition between different features of the game, they should be linked together for a smooth story.
NFR2	Version Control and Updates	Non-Functional Requirement	Timely update in response to testing.	Ensure compatibility between different versions of the app and communicate changes effectively to users through release notes or changelogs.
NFR3	Performance Optimization	Non-Functional Requirement	Optimized assets in the system	Maintain compatibility with a range of devices, screen sizes, and operating system versions while delivering a consistent and enjoyable user experience.
NFR4	Feedback and Support	Non-Functional Requirement	Channels to receive feedback and provide support	Set up appropriate channels for addressing user feedback and ensure timely and helpful responses to support inquiries.
NFR5	Compliance	Non-Functional Requirement	Use government-compliant features.	The game must comply with applicable Laws and regulations.

Table 4.2 Shows all the Non-Functional requirements in the system

4.2 List of Actors

4.2.1 User

The user will play the game and give feedback if they face any kind of issue.

4.2.2 Developer

The developer team will be to make all kinds of updates. This team also solves glitches and any problems related to programming and backend issues to give a better response to the user.

4.3 List of use cases

4.3.1 Authentication

In Authentication, we will log in to the app. But if we don't have an account then we will sign up and create an account.

4.3.2 Login

In login the user will enter their credentials and log into their account to save the progress they are making.

4.3.3 Signup

In signup they new user that do not have an account will be able to create their account and then process to login.

4.3.4 Main menu

In the main menu, we are having options i.e., start game, resume game, setting, leader board, quiz, and developer.

4.3.5 New Game

The New Game option will allow the user to start a fresh game, with all the default settings.

4.3.6 Resume

This will allow the user to continue with the all the progress he has made within the app.

4.3.7 Setting

The setting option will allow the user to access different settings inside the app like volume, music, difficulty level and sounds.

4.3.8 Leaderboard

Leaderboard displays the top scorers in the game, the users with the highest stats will be displayed.

4.3.9 Playing mode

This involves a variety of rocket engines. We will have no different rocket engines to be chosen.

4.3.10 Inventory:

Inventory involves different parts, skins, and other required accessories of the rocket.

4.3.11 Build your Rocket:

The players have to assemble different parts to construct a rocket that they are going to be launched.

4.3.12 Quiz:

We will give multiple questions related to rocket working, parts, and orbital physics so that we will be able to know the level of knowledge of the player.

4.3.13 Learning tutorials

If any player is interested in learning more about rockets, they will have the tutorial URL to be accessed on YouTube.

4.3.14 Challenges

Players will have different challenges to complete to jump to test their learning.

4.3.15 Chat System

The users will allow to interact with each other using the chat system. They will form a community to help each other regarding similar topics.

4.3.16 Achievements

Rewarding medals and titles to the player on completing challenges and upgrading levels.

4.3.17 Credits

Credit will display the developers working behind the app.

4.4 System Use Case Diagram

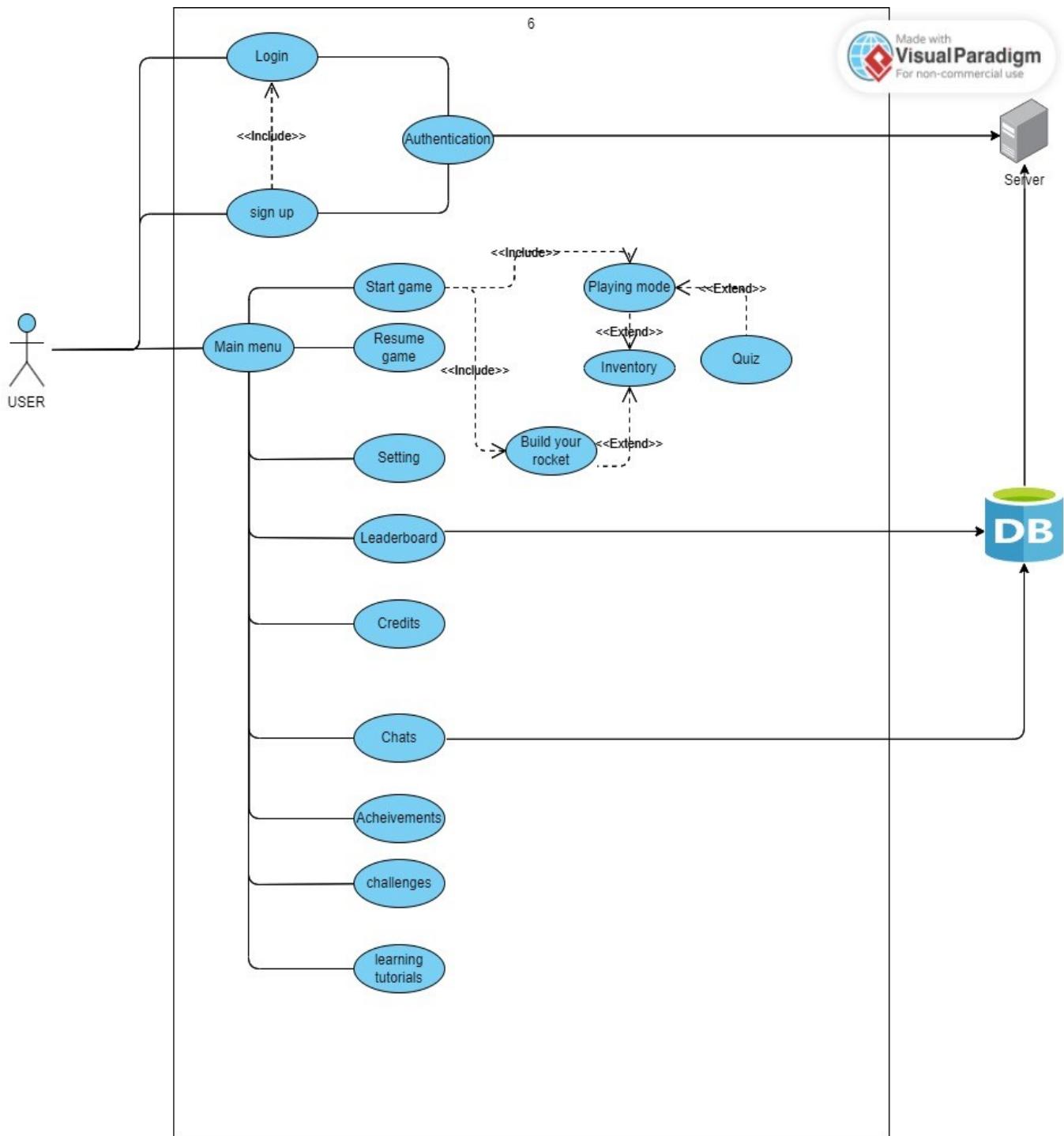


Figure 4.1: System Use-case Diagram

Figure 4.1 shows the working of every feature in the system. It shows how multiple systems are working together.

4.5 Extended use cases

Table 4.3: Extended Use Case 1

Use Case ID:	UC-1		
Use Case Name:	Authentication		
Created By:	Saad Ahmad	Last Updated By:	Saad Ahmad
Date Created:	31 MAY 2023	Last Revision Date:	31 MAY 2023
Actors:	User		
Description:	The user performs the authentication process by providing credentials and undergoing verification to gain access to a system or service.		
Trigger:	When the user will try to log in using his email and password This event prompts the system to prompt the user for their username and password or other required credentials to verify their identity.		
Preconditions:	<ol style="list-style-type: none"> 1. The user must have registered their account before logging in 2. The user's account within the app must be active and not suspended or deactivated. 3. The user must have a stable internet connection. 		
Postconditions:	<ol style="list-style-type: none"> 1. The user will be granted access if the email is successfully verified 2. The user will not be granted access if the login is invalid 		
Normal Flow:	<ol style="list-style-type: none"> 1. The user must have the app installed in the system 2. The user will open the app for logging in 3. The user will click on login if he previously had an account and if not He first has to register 4. Provided login information will be sent to the server for verification. 5. Access will be granted to the user 		
Alternative Flows:	None		
Exceptions:	<ol style="list-style-type: none"> 1. Login is Invalid 2. Please try again, Please Enter a valid email and password 3. Forgotten username password 4. Help reset the password 		
Includes:	None		
Frequency of Use:	Once per Device commonly. Once you have logged in to a Device you don't need to login in again until you log out		
Special Requirements:	Security plays a crucial role in the authentication use case. It focuses on safeguarding user credentials, protecting sensitive information, and preventing unauthorized access. Here are some considerations for security in the authentication use case:		
Assumptions:	Users must consider privacy to secure their account reward titles and chats.		

Table 4.4: Extended Use Case 2

Use Case ID:	UC-2		
Use Case Name:	Main Menu		
Created By:	Saad Ahmad	Last Updated By:	Saad Ahmad
Date Created:	31 MAY 2023	Last Revision Date:	31 MAY 2023
Actors:	User		
Description:	The user can explore different options in the main menu like start game, resume game, change settings, check the leaderboard, and attempt Quiz.		
Trigger:	When a user wants to Play the Game or Change settings or anything related to the app would be done through the Main menu so This event Triggers every time a user opens the app		
Preconditions:	<ol style="list-style-type: none">1. App Installation: The user must have registered their account before logging in2. App Launch: The user's account within the app must be active and not suspended or deactivated3. Internet Connectivity: The user must have a stable internet connection.4. System Compatibility: The user's device should meet the minimum system requirements specified by the app developer to ensure proper functionality. This includes having a compatible operating system version, sufficient storage space, and necessary hardware capabilities.		
Postconditions:	<ol style="list-style-type: none">1. Menu Display: The main menu is successfully displayed on the user's screen, showing the available options, features, or sections of the app.2. User Interaction: The user can interact with the main menu by selecting options, tapping on icons, or performing actions associated with the menu items.3. Feature Accessibility: Upon selecting a specific menu item, the corresponding feature or content associated with that option becomes accessible and visible to the user.		
Normal Flow:	<ol style="list-style-type: none">1. The user must have the app installed in the system2. User will Click on the main menu after logging in3. The main menu will appear in front of the user4. User can choose whatever function feature he wants to use		
Alternative Flows:	None		
Exceptions:	None		
Includes:	None		
Frequency of Use:	Multiple times like every time a user wants to perform quiz different modes or check the leaderboard		
Special Requirements:	None		
Assumptions:	Users understand the main menu's purpose and can navigate through it to access various app features.		
Notes and Issues:	None		

Table 4.5: Extended Use Case 3

Use Case ID:	UC-3		
Use Case Name:	Playing Mode		
Created By:	Saad Ahmad	Last Updated By:	Saad Ahmad
Date Created:	31 MAY 2023	Last Revision Date:	31 MAY 2023
Actors:	User		
Description:	The playing modes use case allows users to select and engage in different game modes		
Trigger:	When the user wants to select the "Play Modes" option from the main menu or in-game menu.		
Preconditions:	<ol style="list-style-type: none">If certain playing modes are locked or require unlocking through gameplay or achievements, the user should meet the necessary criteria or have progressed to the appropriate stage in the game.The required assets and resources for the playing modes should be downloaded, loaded, and available on the user's device.The user's device should meet the minimum system requirements specified by the app developer to ensure proper functionality of the playing modes		
Postconditions:	<ol style="list-style-type: none">The user has successfully selected a specific playing mode from the available options.The app initializes the selected playing mode, setting up the game environment, rules, and any necessary assets or resources.The user engages in the selected playing mode, progressing through the gameplay, achieving objectives, and interacting with game elements.		
Normal Flow:	<ol style="list-style-type: none">The user must have access to the main menu after logging in.Users will select the playing modes feature in the main menu.The user will select the mode according to the difficulty level he wants to choose to play.		
Alternative Flows:	None		
Exceptions:	Mode Not Unlocked		
Includes:	None		
Frequency of Use:	Every time a user wants to experience different modes		
Special Requirements:	Quality		
Assumptions:	Users understand the different playing modes offered by the app and can engage in them without encountering any technical or accessibility issues.		
Notes and Issues:			

Table 4.6: Extended Use Case 4

Use Case ID:	UC-4		
Use Case Name:	Build Your Rocket		
Created By:	Saad Ahmad	Last Updated By:	Saad Ahmad
Date Created:	6 JUNE 2023	Last Revision Date:	6 JUNE 2023
Actors:	User		
Description:	The user will build his rockets by using different parts available in the inventory, they would be stacked together and then used to explore the world.		
Trigger:	When the user moves to play the game they will select this option to continue with the game as they cannot play the game without building the rocket first.		
Preconditions:	<ol style="list-style-type: none"> 1. The user has successfully logged into the app. 2. The user is in the playing mode of the game 3. The user has accessed the build option in the app 		
Postconditions:	The design the user has created will be saved and transferred to the game mode where the player will fly with that rocket		
Normal Flow:	<ol style="list-style-type: none"> 1. The user logs into the app 2. The user selects the game option from the main menu. 3. The user moves to the build option to create 4. The user drags and drops rocket parts from the inventory onto the canvas to design their rocket, adjusting placement and attributes as needed. 5. The user moves to the game to test the created rocket. 6. The user finalizes and saves the rocket design, with the option to share it with others or continue with other activities within the app. 		
Alternative Flows:	None		
Exceptions:	None		
Includes:	None		
Frequency of Use:	This option would be utilized every time a user decides to play the game the user would be brought to the build option before he can play the game.		
Special Requirements:	None		
Assumptions:	The user has accessed the playing mode and understands the functionality of the app.		
Notes and Issues:	None		

Table 4.7: Extended Use Case 5

Use Case ID:	UC-5		
Use Case Name:	Credits		
Created By:	Muhammad Makki	Last Updated By:	Muhammad Makki
Date Created:	6 JUNE 2023	Last Revision Date:	6 JUNE 2023
Actors:	User		

Description:	The user will be able to see the developers working on the game
Trigger:	Clicking the button dedicated to credits in the main menu
Preconditions:	The user has successfully logged into the app.
Postconditions:	The credits would be displayed and the user could exit back to the main menu
Normal Flow:	<ol style="list-style-type: none"> 1. The user logs into the app 2. The user selects the game option from the main menu. 3. Users can access the credits
Alternative Flows:	None
Exceptions:	None
Includes:	None
Frequency of Use:	This option would be utilized every time a user decides to view the credits.
Special Requirements:	None
Assumptions:	The user has accessed the main menu by starting the app and understands English.
Notes and Issues:	None

Table 4.8: Extended Use Case 6

Use Case ID:	UC-6		
Use Case Name:	Setting		
Created By:	Muhammad Makki	Last Updated By:	Muhammad Makki
Date Created:	6 JUNE 2023	Last Revision Date:	6 JUNE 2023
Actors:	User		
Description:	The user will be able to change some preferences of the app		
Trigger:	Clicking the button dedicated to setting in the main menu whenever they need to change their setting in the game		
Preconditions:	The user has successfully logged into the app.		
Postconditions:	The applied setting would be applied to the user's device for the game.		
Normal Flow:	<ol style="list-style-type: none"> 1. The user logs into the app 2. The user selects the setting option from the main menu. 3. Users can change different settings for volume, music, and difficulty. 4. Exit to the main menu 		
Alternative Flows:	Can be accessed from all screens within the game from the dedicated button,		
Exceptions:	None		
Includes:	None		
Frequency of Use:	This option would be utilized every time a user decides to change the settings.		

Special Requirements:	None		
Assumptions:	The user has accessed the main menu by starting the app and understands English		
Notes and Issues:	None		

Table 4.9: Extended Use Case 7

Use Case ID:	UC-7		
Use Case Name:	Resume		
Created By:	Muhammad Makki	Last Updated By:	Muhammad Makki
Date Created:	6 JUNE 2023	Last Revision Date:	6 JUNE 2023
Actors:	User		
Description:	The user will be able to resume the game from a previously saved checkpoint		
Trigger:	Clicking the button dedicated to Resume in the main menu upon startup when they have to load their game		
Preconditions:	<ol style="list-style-type: none"> 1. The user has successfully logged into the app. 2. The user has a pre-saved game 		
Postconditions:	The game would be logged in from a previously saved checkpoint		
Normal Flow:	<ol style="list-style-type: none"> 1. The user logs into the app 2. The user selects the Resume option from the main menu. 3. The game will be loaded from a previously saved checkpoint 		
Alternative Flows:	None		
Exceptions:	None		
Includes:	None		
Frequency of Use:	This option would be utilized every time a user decides to play a game from a previously saved checkpoint.		
Special Requirements:	None		
Assumptions:	The user has accessed the main menu by starting the app and understands the English		
Notes and Issues:	None		

Table 4.10: Extended Use Case 8

Use Case ID:	UC-8		
Use Case Name:	New Game		
Created By:	Muhammad Makki	Last Updated By:	Muhammad Makki
Date Created:	6 JUNE 2023	Last Revision Date:	6 JUNE 2023
Actors:	User		
Description:	The user will be able to start a new game alongside the previously saved game.		

Trigger:	By clicking the button dedicated to New Game in the main menu whenever the user wants to start a new game
Preconditions:	The user has successfully logged into the app.
Postconditions:	The game would be started from scratch with an empty building site.
Normal Flow:	<ol style="list-style-type: none"> 1. The user logs into the app 2. The user selects the New Game option from the main menu. 3. The game would begin from an empty building site without all the unlocked features with all the default settings
Alternative Flows:	None
Exceptions:	None
Includes:	None
Frequency of Use:	This option would be utilized every time a user decides to play a game from scratch.
Special Requirements:	None
Assumptions:	The user has accessed the main menu by starting the app and understands English
Notes and Issues:	None

Table 4.11: Extended Use Case 9

Use Case ID:	UC-9		
Use Case Name:	Inventory		
Created By:	Shameer Zeeshan	Last Updated By:	Shameer Zeeshan
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	Here the user will select a range of skins, rocket parts, and other accessories for the rocket.		
Trigger:	As the user moves to play the game, he can select the inventory button.		
Preconditions:	The user must have selected the game mode before moving into the inventory.		
Postconditions:	The selections user has made while in the inventory menu will be incorporated into the gameplay data and stored accordingly Inventory will manage the used/added items and update them for the particular user.		
Normal Flow:	<ol style="list-style-type: none"> 6. The user must have access to the main menu after logging in. 7. The user will select the playing modes feature in the main menu. 8. Users will have the option to select the inventory when the game begins. 		
Alternative Flows:	1. The player can choose not to change the rocket and go with the preloaded settings.		
Exceptions:	None		
Includes:	None		

Frequency of Use:	Every time a user wants to modify the rocket.
Special Requirements:	None
Assumptions:	The user is in the game phase of the app.
Notes and Issues:	None

Table 4.12: Extended Use Case 10

Use Case ID:	UC-10		
Use Case Name:	Quiz		
Created By:	Shameer Zeeshan	Last Updated By:	Shameer Zeeshan
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	Here the user will assess the knowledge they have gained in the game section		
Trigger:	When a user wants to communicate with their friends or want to send gifts to their friends, they will use the chat box feature		
Preconditions:	The user must have logged into the app and selected the quiz feature.		
Postconditions:	The score and achievement should be incorporated into the user profile. After the quiz, the unlocked options should be enabled for the user.		
Normal Flow:	<ol style="list-style-type: none"> 1. The users log into the app. 2. The main menu engages the playing modes. 3. Selects the quiz feature. 4. Starts the quiz 		
Alternative Flows:	The user can skip the quiz option altogether and move to the game.		
Exceptions:	If the game is at a certain level so the user will not be able to play the game unless completing the quiz		
Includes:	None		
Frequency of Use:	Every time a new concept is introduced and explained it could be tested in the quiz section.		
Special Requirements:	None		
Assumptions:	The user has logged into the app and selected the quiz option		
Notes and Issues:	None		

Table 4.13: Extended Use Case 11

Use Case ID:	UC-11		
Use Case Name:	Challenges		
Created By:	Shameer Zeeshan	Last Updated By:	Shameer Zeeshan
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	The user will have different challenges to test different situations		

Trigger:	In game mode, the user will have access to challenges they can start the challenge by clicking on its dedicated button
Preconditions:	The user should be logged in to the app and be in the game section.
Postconditions:	The rewards of the challenges should be incorporated with the player profile and the data should be synced so that smooth gameplay could be provided.
Normal Flow:	<ol style="list-style-type: none"> 1. The users start the app 2. From the main menu selects the game feature 3. Then select challenges from the sub-menu. 4. Performs the challenge 5. Save the result 6. Return to the main game.
Alternative Flows:	The user can skip the challenge option altogether and move to the game.
Exceptions:	If some portion of the game is played then the user is challenged to perform a certain task in the game
Includes:	None
Frequency of Use:	Every time a user wants to complete the challenge, they can use it. They would be redirected here when they have reached a certain level in the game that needs to be tested.
Special Requirements:	None
Assumptions:	The user has logged into the app to save the record.
Notes and Issues:	None

Table 4.14: Extended Use Case 12

Use Case ID:	UC-12		
Use Case Name:	Chat System		
Created By:	Shameer Zeeshan	Last Updated By:	Shameer Zeeshan
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	It enables the user to communicate and exchange messages in real time within the app, facilitating interaction and collaboration among users.		
Trigger:	When a user wants to communicate with their friends or want to send gifts to their friends they will use the chat box feature		
Preconditions:	The user must be logged into the Game The user must have an internet connection to deliver or receive messages		
Postconditions:	Messages sent by users are successfully delivered to the intended recipients or chat groups. Sent and received messages are displayed in the chat interface, allowing users to view the conversation history.		
Normal Flow:	<ol style="list-style-type: none"> 1. The users log into the app 2. From the main menu select the chat box 3. Now users can start chatting with their friends 		
Alternative Flows:	Users can open the chat box while playing the game also as well as in the main menu		

Exceptions:	A user can skip the chat system without any hesitation it's up to him whether he wants to use it or not
Includes:	None
Frequency of Use:	Every time user wants to communicate
Special Requirements:	None
Assumptions:	Users have a basic understanding of how to use chat systems and can send and receive messages effectively within the app.
Notes and Issues:	None

Table 4.15: Extended Use Case 13

Use Case ID:	UC-13		
Use Case Name:	Achievements		
Created By:	Farheen	Last Updated By:	Farheen
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	Enables the players to unlock and track their progress in the game by accomplishing specific tasks or reaching significant milestones, enhancing their gaming experience and providing a sense of accomplishment.		
Trigger:	Whenever the Player completes a task or reaches a milestone in the game.		
Preconditions:	The user must be logged into the Game Users must have completed specific tasks to unlock achievements and progress.		
Postconditions:	The user successfully unlocks an achievement by completing the required task or reaching a significant milestone in the game. Unlocking achievements may grant the player in-game rewards, such as virtual currency, items, or access to additional content, enhancing the gameplay experience.		
Normal Flow:	1. The users log into the app 2. From the main menu look for the achievements option 3. Select the Achievements feature and now the user can check his rewards and achievements progress		
Alternative Flows:	None		
Exceptions:	If the player fails to complete the required task or reach a milestone within the game, the achievement associated with it remains locked.		
Includes:	None		
Frequency of Use:	Every time user wants to check their progress		
Special Requirements:	None		
Assumptions:	Players are motivated by the prospect of unlocking achievements and will actively engage in the game to complete tasks and reach milestones.		
Notes and Issues:	None		

Table 4.16: Extended Use Case 14

Use Case ID:	UC-14		
Use Case Name:	Leaderboard		
Created By:	Farheen	Last Updated By:	Farheen
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	It allows users to view and compare their game scores or rankings with other players,		
Trigger:	Whenever a user scored well and wants to check their rank in the Leaderboard		
Preconditions:	The user must be logged into the Game The user must have a successful Internet connection. The user must have played the game before setting a score in Leaderboard.		
Postconditions:	The player's score is successfully submitted to the leaderboard. The player's score and corresponding rank are displayed on the leaderboard, allowing them and others to see their position among other players.		
Normal Flow:	1. The users log into the app 2. From the main menu look for the Leaderboard option 3. Select the Leaderboard feature and now the user can check his score and rank among other players		
Alternative Flows:	None		
Exceptions:	If the user hasn't played the game before so the score and ranking would not be displayed in the Leaderboard		
Includes:	None		
Frequency of Use:	Every time user wants to check their ranking		
Special Requirements:	None		
Assumptions:	Players strive to achieve high scores and actively participate in the game to earn a prominent position on the leaderboard, promoting healthy competition and engagement.		
Notes and Issues:	None		

Table 4.17: Extended Use Case 15

Use Case ID:	UC-15		
Use Case Name:	Login		
Created By:	Farheen	Last Updated By:	Farheen
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	It allows users to log in to the game securely by providing their credentials.		
Trigger:	Whenever a user wants to use the app or use a feature that requires authentication.		
Preconditions:	Users have opened the app successfully		

	Users have a secure and stable internet connection User has registered their account before The user entered the correct login details
Postconditions:	User-provided credentials are successfully validated The user has been granted access to the game
Normal Flow:	<ol style="list-style-type: none"> 1. The user must open the app 2. The user clicks on the login option 3. A user provides the credentials 4. The message will be displayed that the user has successfully logged in or failed to login in
Alternative Flows:	None
Exceptions:	If the user hasn't registered before or the user had forgotten the password
Includes:	None
Frequency of Use:	Only once
Special Requirements:	None
Assumptions:	Users have valid credentials and are authorized to access the app, ensuring the security and integrity of user accounts and data.
Notes and Issues:	None

Table 4.18: Extended Use Case 16

Use Case ID:	UC-16		
Use Case Name:	Sign Up		
Created By:	Farheen	Last Updated By:	Farheen
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	It allows users to Register new accounts by providing valid information to get access to the game's features		
Trigger:	Whenever a user wants to login into the game their account must be registered before to access game features and some functionalities.		
Preconditions:	Users have opened the app successfully Users have a secure and stable internet connection Users have a registered email before to link with The user entered the correct details for the registration process.		
Postconditions:	User-provided credentials are successfully validated The user has been granted access to the game		
Normal Flow:	<ol style="list-style-type: none"> 1. The user must open the app 2. Users click on the signup option to register 3. A user provides the correct information for registration 4. The message will appear that the user has successfully registered 		
Alternative Flows:	None		
Exceptions:	If the user-provided information is incorrect		
Includes:	Login		
Frequency of Use:	Every time you need to create a new account		

Special Requirements:	None
Assumptions:	Users provide accurate and valid information during the signup process, allowing for successful account creation and ensuring the integrity of user data within the app.
Notes and Issues:	None

Table 4.19: Extended Use Case 17

Use Case ID:	UC-17		
Use Case Name:	Learning Tutorials		
Created By:	Shameer Zeeshan	Last Updated By:	Shameer Zeeshan
Date Created:	1 JUNE 2023	Last Revision Date:	1 JUNE 2023
Actors:	User		
Description:	Tutorials would be provided to the users They will have access to different videos loaded from YouTube.		
Trigger:	The learning tutorial will be opened as soon as the user selects the option from the menu		
Preconditions:	None		
Postconditions:	Return to the main menu		
Normal Flow:	<ol style="list-style-type: none"> 1. The user starts the app 2. From the main menu selects the tutorial feature 3. Starts to see videos 4. Returns to the main menu. 		
Alternative Flows:	The user can skip the tutorial option altogether and move to the game.		
Exceptions:	None		
Includes:	None		
Frequency of Use:	Every time a new concept is introduced, and the user needs to learn it in depth, they will be referred to here		
Special Requirements:	None		
Assumptions:	The user has logged into the app		
Notes and Issues:	None		

5. USER INTERFACE / SCREENS

5.1 Main Menu

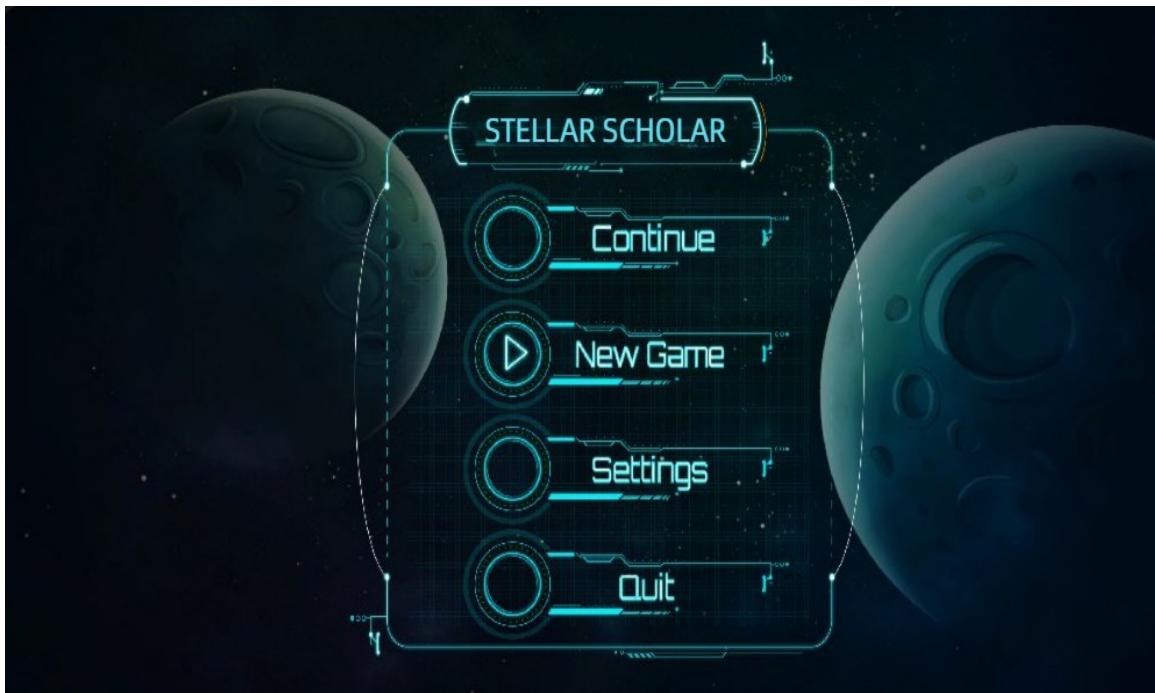


Figure 5.1 Main Menu User Interface

Figure 5.1 shows the main menu of the Stellar Scholar App, The user will be brought here as soon as the game loads, User can then perform a range of operations on the app using the button available

1. Continue

The user will go back to the previously saved checkpoint and continue with the game or quiz.

2. New Game

Will allow the user to open a new space in the game building everything from the beginning. The previously saved blueprints would be removed from history and a new journey would begin.

3. Setting

As the user presses the setting button the screen would be transitioned from the main menu to the setting screen which would be explained next.

4. Quit

This button will allow the user to exit the game and return to their home screen.

5.2 Setting

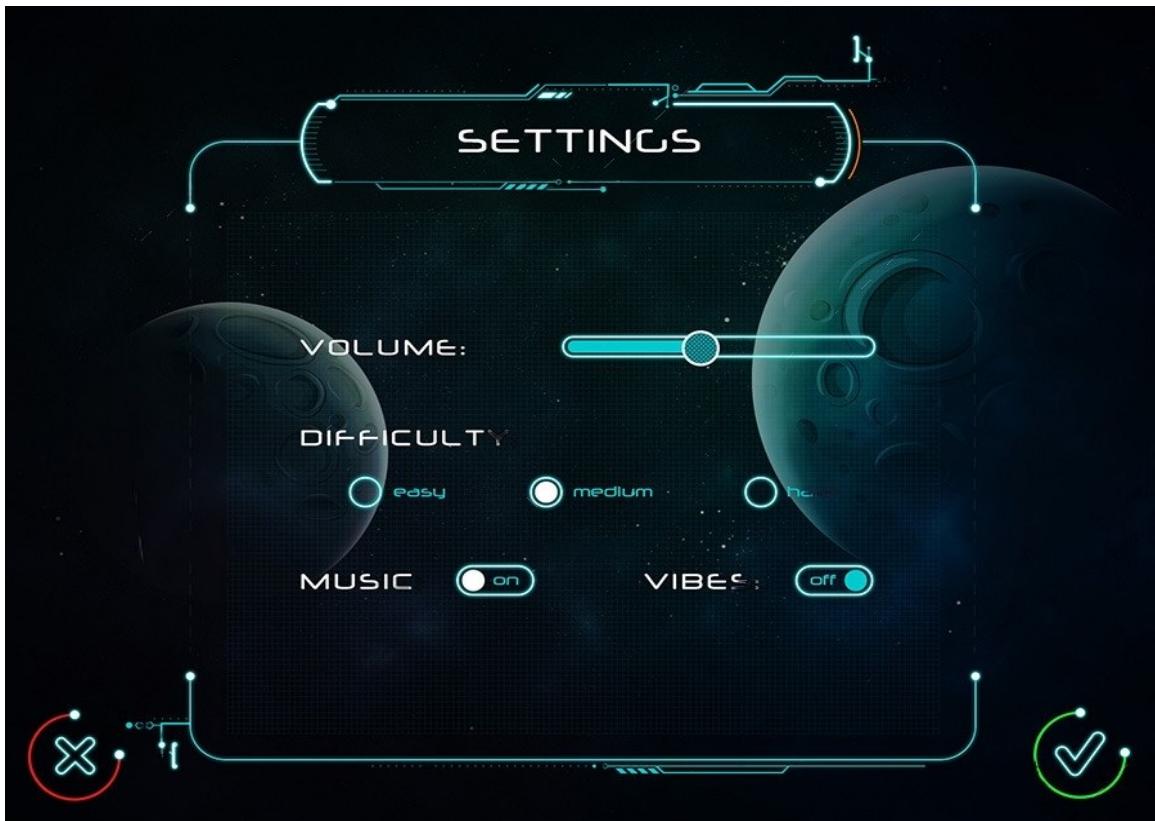


Figure 5.2 Setting User Interface

Different features that would be accessible in the settings screen are shown in Figure 5.2.

1. Volume

Volume Slider will allow the user to set the sound volume of the device inside the game environment.

2. Difficulty

Using the option is difficult the user will select the level of hardness in the app.

3. Music

This toggle button will remove or start the background music from the app.

4. Vibes

This button will remove or start the interactive sounds from the app.

5.3 Loading



Figure 5.3 Loading Screen

Figure 5.3 shows the screen that would be displayed whenever the app is in the loading phase.



Final Year Project Report

Project Evaluation Tool Based on PBL

Project Advisor:

Mr. Rana Marwat Hussain

Project Co-Advisor:

Mr. Noaman Saleem

Submitted By:

Muhammad Daniyal Mazhar	F2019266101
Muhammad Umair Nasir	F2019266148
Abdul Rehman	F2019266127
Abdul Haseeb	F2021105247

Program:

BS(CS)

Session

(2019-2023)

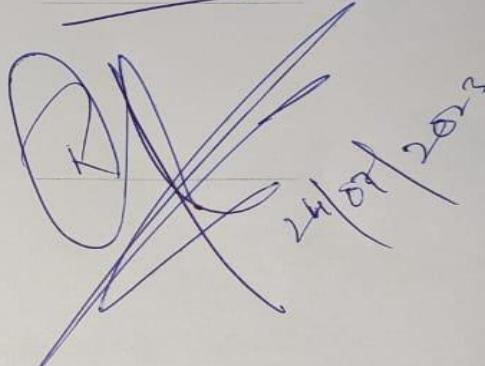
University of Management and Technology

C-II Johar Town Lahore Pakistan

Final Approval

- **Head of Department**
Department of Computer Science
School of Systems & Technology
UMT Lahore
- **Director (Final Year Projects-CS)**
Department of Computer Science,
School of Systems & Technology
UMT Lahore
- **Supervisor**
• Department of Computer Science
• School of Systems & Technology
UMT Lahore

Tal

Tham Almal

24/02/2023

ii

Plagairism Report

University of Management and Technology, Lahore

Similarity Report

Turnitin Originality Report

Project Evaluation Tool Based on PBL by Muhammad Daniyal Mazhar, Muhammad Umair Nasir, Abdul Rehman and Abdul Haseeb

From Reference Desk (LRC info desk)

- Processed on 17-Jul-2023 11:50 PKT
- ID: 2132427863
- Word Count: 8395

Similarity Index

9%

Similarity by Source

Internet Sources:

8%

Publications:

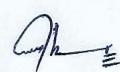
1%

Student Papers:

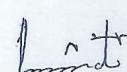
8%

Sources:

1. 1% match (student papers from 23-Jul-2018)
Submitted to Higher Education Commission Pakistan on 2018-07-23
2. 1% match (student papers from 06-Aug-2018)
Submitted to Higher Education Commission Pakistan on 2018-08-06



Checked by



Verified by CLO

Note:

- Sometimes the overall similarity index may be a smaller than the repository percentages combined. This would be due to overlapping text within the repositories.
- It is a system generated report.

Dedication

Dedicated to all our teachers who taught us the way of learning and taught us that how we can make ourselves better every day. It is also dedicated to our families who support us financially. But main credit goes to our respected teachers who remove every hurdle of our path and show us the right way to perform excellent in our curricular and co-curricular activities. Teachers like Mr. Rana Marwat Hussain and all other SST teachers groom us very well so our project is dedicated to them. Furthermore, it is also dedicated to those friends who supported us and helped us in our difficult courses.

Acknowledgment

First of all, we would like to thank ALLAH who gave us health, wealth and an intelligent mind which helped us to reach here. We are here just because of ALLAH because ALLAH gave us ability to do all the things that we have done so far in our academic carrier and we would like to thank our family, friends and most important our teachers who taught us allot. We also want to accept the reality that a good teacher can take you from earth to sky. I would like to thank ALLAH again that ALLAH gave us brilliant teachers who not only taught us about courses but also taught us about life that how we can spend our life generously, patiently and most important truly. We would like to thank our friends also who pushes us toward the doors of success. Our Family also played a big role in our success because they always stood with us in every difficult time.

Project Discription

Project Title

Project Evaluation Tool Based on PBL

Objective

To automate fyp registration process.

Undertaken by

Muhammad Daniyal Mazhar

Muhammad Umair Nasir

Abdul Rehman

Abdul Haseeb

Supervised by

Mr. Rana Marwat Hussain

Starting Date

15 October 2022

Completion Date

10 June 2023

Tools Used

HTML, CSS, Java Script, PHP, Bootstrap

Abstract

The FYP registration procedure is the foundation of our project. It assists in automating the registration procedure. This project is being developed to reduce the amount of time that each person engaged in the FYP registration process needs to spend on it. It will make registering students easier for the committee members. It is challenging for the committee members to remember every student's details during the registration procedure. The committee members' struggles cause the loss of important data. We created a web application to assist committee members in keeping track of registrations and receiving notifications in response to these issues. Members of the committee can reach the advisers to obtain all relevant data regarding the students and project proposal.

REVISION CHART

Table 0.1: Revision Chart

Version	Primary Author(s)	Description of Version	Date Completed
Draft	TBD	Initial draft created for distribution and review comments	(To be decided) TBD
Preliminary	TBD	Second draft incorporating initial review comments, distributed for final review	TBD
Final	TBD	First complete draft, which is placed under change control	TBD
Revision 1	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
Revision 2	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
Etc.	TBD	TBD	TBD

TABLE OF CONTENTS

FINAL APPROVAL.....	II
PLAGAIRISM REPORT.....	III
DEDICATION.....	IV
ACKNOWLEDGMENT.....	V
PROJECT DESCRIPTION.....	VI
ABSTRACT.....	VII
TABLE OF CONTENTS.....	IX
DEFINITIONS AND ACRONYMS.....	XI
LIST OF FIGURES.....	XII
LIST OF TABLES.....	XIII
1. INTRODUCTION.....	1
1.1 MOTIVATIONS.....	1
1.2 PROJECT OVERVIEW.....	1
1.3 PROBLEM STATEMENT.....	1
1.4 OBJECTIVES.....	2
2. DOMAIN ANALYSIS.....	3
2.1 CUSTOMER.....	3
2.2 STAKEHOLDERS.....	3
2.3 AFFECTED GROUPS WITH ECONOMIC IMPACT.....	3
2.4 DEPENDENCIES/ EXTERNAL SYSTEMS.....	3
2.5 REFERENCE DOCUMENTS.....	3
2.5.1 <i>Related Projects</i>	3
2.5.2 <i>Feature Comparison</i>	4
3. REQUIREMENTS ANALYSIS.....	5
3.1 REQUIREMENTS.....	5
3.2 LIST OF REQUIREMENTS.....	5
3.3 LIST OF ACTORS.....	9
3.4 LIST OF ACTORS.....	9
3.5 LIST OF USE CASES.....	9
3.6 SYSTEM USE CASE DIAGRAM.....	10
3.7 EXTENDED USE CASES.....	12
3.8 USER INTERFACES (MOCK SCREENS).....	19
4. DATA FLOW DIAGRAM.....	21
4.1 DATA FLOW DIAGRAM LEVEL 0.....	21
4.2 DATA FLOW DIAGRAM LEVEL 1.....	21
5. SYSTEM DESIGN.....	22
5.1 WORK BREAK DOWN STRUCTURE (WBS).....	22
5.2 ACTIVITY DIAGRAM.....	23
5.3 CLASS DIAGRAM.....	27
5.4 COLLABORATION DIAGRAM.....	28
5.5 SEQUENCE DIAGRAMS.....	32
5.6 DATABASE DIAGRAM.....	36
6. IMPLEMENTATION DETAILS.....	37
6.1 DEVELOPMENT SETUP.....	37

6.2	DEPLOYMENT SETUP.....	37
6.3	CONSTRAINTS.....	37
6.3.1	<i>Assumptions</i>	37
6.3.2	<i>System constraints</i>	37
6.3.3	<i>Restrictions</i>	37
6.3.4	<i>Limitations</i>	37
7.	TESTING.....	38
7.1	EXTENDED TEST CASES.....	38
7.2	DECISION TABLE.....	52
7.2.1	<i>Code snippet</i>	52
7.3	TRACEABILITY MATRIX.....	54
7.3.1	<i>RID vs UCID (requirements vs use cases)</i>	54
7.3.2	<i>Prototypes (RID vs PID)</i>	55
7.3.3	<i>Test Cases (RID vs TID)</i>	56
7.3.4	<i>Coverage (UCID vs TID)</i>	57
8.	RESULTS.....	58
8.1	% COMPLETION.....	58
8.2	% ACCURACY.....	58
8.3	% CORRECTNESS.....	58
9.	CONCLUSION.....	59
10.	FUTURE WORK.....	60

Definitions and Acronyms

Table 0.1: table of acronyms and definitions

Acronym	Definition
UMT	University of Management and Technology
PBL	Project Base Learning

List of Figures

Figure 3.1: Use Case Diagram with Explanation and scope of our system	11
Figure 3.2: Login Page Interface	19
Figure 3.3: Student Portal Interface	19
Figure 3.4: Advisor Interface	20
Figure 3.5: Evaluator Interface	20
Figure 4.1: DFD Level 0 Diagram	21
Figure 4.2: DFD Level 1 Diagram	21
Figure 5.1: Work Break Down Diagram with Explanation	22
Figure 5.2: Activity Diagram for Login/Signup	23
Figure 5.3: Activity Diagram for FYP registration	24
Figure 5.4: Activity Diagram for view and upload marks	25
Figure 5.5: Activity Diagram for view FYP view groups	26
Figure 5.6: Class Diagram with Explanation	27
Figure 5.7: Collaboration Diagram for Student to Sign up	28
Figure 5.8: Collaboration Diagram for a student to Register	29
Figure 5.9: Collaboration Diagram for a student to View Marks	29
Figure 5.10: Collaboration Diagram for Evaluator to Update Marks	30
Figure 5.11: Collaboration Diagram for Evaluator to check all Registered Groups	31
Figure 5.12: Collaboration Diagram for a student to View FYP Criteria	31
Figure 5.13: Sequence Diagram for login/register	32
Figure 5.14: Sequence Diagram for FYP registration	33
Figure 5.15: Sequence Diagram for view and upload marks	34
Figure 5.16: Sequence Diagram for View Registered Groups	35
Figure 5.17: Database Diagram	36
Figure 7.1: Login Page Snapshot	52
Figure 7.2: Dashboard of Advisor Snapshot	52
Figure 7.3: Code Snapshot of Student Portal	53
Figure 7.4: Evaluator Side Snapshot	53

List of Tables

Table 0.1: Revision Chart	viii
Table 0.1: table of acronyms and definitions	xi
Table 2.1: List of Stakeholders	3
Table 2.2: Feature Comparison	4
Table 3.1: Requirements Categories	5
Table 3.2: List of Requirements	5
Table 3.3: Functional Requirement & Non-Functional Requirement	6
Table 3.4: UC-1 (Perform registration)	12
Table 3.5: UC-2 (Make Log In)	12
Table 3.6: UC-03 (Logout)	13
Table 3.7: UC-04 (View Portal)	13
Table 3.8: UC-05 (Title of the Project)	14
Table 3.9: UC-06 (Team Leader Details)	14
Table 3.10: UC-07 (Team Members Details)	14
Table 3.11: UC-08 (Project Evaluation Criteria)	15
Table 3.12: UC-09 (Instructor Evaluation Result)	15
Table 3.13: UC-10 (Project Evaluator Evaluation Result)	15
Table 3.14: UC-11 (Set Registration Time)	16
Table 3.15: UC-12 (Dashboard)	16
Table 3.16: UC-13 (Check User)	16
Table 3.17: UC-14 (Project Evaluation Criterion)	17
Table 3.18: UC-15 (Evaluate Groups)	17
Table 3.19: UC-16 (Project External Evaluator)	17
Table 3.20: UC-17 (Result Updating)	18
Table 3.21: UC-18 (Project Result)	18
Table 3.22: UC-19 (Final Result)	18
Table 7.1: Perform Registration	38
Table 7.2: Make Login	39
Table 7.3: Log out	40
Table 7.4: View Portal	41
Table 7.5: Title of the Project	42
Table 7.6: Team Leader Details	42
Table 7.7: Team Members Detail	43
Table 7.8: Project Evaluation Criteria	43
Table 7.9: Instructor Evaluation Result	44
Table 7.10: Project Evaluator Evaluation Result	44
Table 7.11: Set Registration Time	45

Table 7.12: Dashboard	45
Table 7.13: Check User	46
Table 7.14: Project Evaluation Criterion	47
Table 7.15: Evaluate Groups	48
Table 7.16: Project External Evaluator	49
Table 7.17: Result Updating	50
Table 7.18: Project Result	51
Table 7.19: Final Result	51

1. INTRODUCTION

The FYP registration procedure is the foundation of our project. Every year, UMT develops exceptionally talented software engineers and gives FYP to the students. A time-consuming aspect of the project is the FYP registration process. A computerized system will give UMT students access to a portal where they may control the entire procedure online. This Portal is currently only to check FYP Project Progress Report and to upload All types of Marks of Project Evaluation Criteria by FYP committee members. All the rules and regulations and guidelines will be available.

The whole project is an online portal that is a web application. The individuals who use this portal are Students and FYP committee members. There are two users of the application. FYP Members cannot register by themselves, they will be provided by a Login and Password by the institute. Students will register themselves by a Login and password. Every type of user will perform different functions.

On the FYP committee members side, every member can check the total number of groups registered with their names and ID's and how many groups are evaluated or not. He can also increase the FYP registration date. Committee member receives the project proposal from the student and he will mark him.

On the Students Side the Student Group Leader will interact with the application by entering the details of his and its group members. A unique Project ID will be created for each group. Students will perform registration of his FYP online using this portal. FYP proposal status will be shown on the portal.

1.1 Motivations

The motivation behind this project is that we all live in a world of technology. Currently, every second university offers the FYP online registration system. Therefore, our motivation is to make FYP registration online so that we can provide opportunities for students and teachers. The goal is to provide students with a platform where they can enroll at the click of a button. This will help students complete everything on time. Students do not have to go to another teacher to submit their FYP proposal.

1.2 Project Overview

Many private and public universities in Pakistan provide information on their website but do not have an automated system for graduation projects. We aim to create software that will register FYPs with supervisors online and provide information about their progress to committee members' computers via a web page, saving them time.

- The proposed system will allow to register only those students who are eligible to register in FYP.
- Students can only register for project if he/she successfully completed the 90 credit hours.

1.3 Problem Statement

This project is designed to save time for everyone involved in the FYP registration process. Maintaining information about each student during the registration process is a challenge for committee members. These hurdles faced by committee members lead to the loss of valuable data. With these issues in mind, we decided to develop a web application that would help committee members keep up to date on time and be notified of each registration. The web application provides step-by-step procedures that students and committee members must follow to successfully complete the registration process.

1.4 Objectives

- The system's primary objective is to develop a computer system in fulfilment of established criteria.
- The system offers a web-based interface that allows the entire procedure to be managed online centrally.
- The system we provide is error-free and has an easy-to-use user interface.
- The system will offer a user-friendly online application that takes only a few hours to become familiar with.

2. DOMAIN ANALYSIS

2.1 Customer

The main target market of our project is our UMT students who can register their FYP Project Group by using our platform and the FYP Committee who can mark them.

2.2 Stakeholders

The stakeholders of the system are as follows:

- Admin
- Student
- Evaluators

Table 2.1: List of Stakeholders

Stakeholder	Role in System
Admin	Responsible for the maintenance of the whole Portal by making the website more convenient to the FYP students and FYP committee members and solves all the problems that may occur in the portal.
Students	Students can maintain his/her account after Registering in the portal and can get obtained marks by the FYP Evaluators
Evaluators	Responsible for maintenance of students' record in the portal.

2.3 Affected Groups with economic impact

- **Evaluators**

For Evaluators, the amount of paperwork will be decreased.

- **Students**

For Students, the amount of paperwork will be decreased.

2.4 Dependencies/ External Systems

- **UMT Database**
- **Internet Connection**

2.5 Reference Documents

All of the documents reviewed during the analytical step will be cited in this phase.

2.5.1 Related Projects

We researched a number of comparable systems before developing the PBL Evaluation System. Their details are provided below:

1. Fast University FYP Registration

Developed by Fast University. The working of this was observed from <http://lhr.nu.edu.pk/fsc/fyp/>

2. Comsats University FYP Registration

Developed by Comsats University. The working of this was observed from <http://lhr.nu.edu.pk/fsc/fyp/>

3. University of Hong Kong

Developed by University of Hong Kong. The working of this was observed from <https://wp.cs.hku.hk/2021/fyp21073/overview/>

4. University of Karachi

Developed by University of Karachi. The working of this was observed from <https://uok.edu.pk/faculties/computerscience/docs/fyp-trf.pdf>

2.5.2 Feature Comparison

Table 2.2: Feature Comparison

Features	PBL	Fast University	Comsats University	University of Hong Kong	University of Karachi
User-Friendly Interface	Yes	No	No	Yes	Yes
Supervisor Selection	Yes	Yes	Yes	Yes	No
Project Submission	Yes	Yes	Yes	No	Yes
Team Formation	Yes	Yes	Yes	Yes	Yes
Document Upload	No	No	Yes	Yes	Yes
Check Marks	Yes	No	No	Yes	Yes
Evaluation Process	Yes	No	Yes	No	Yes
Evaluation Criteria Guideline	No	Yes	Yes	No	Yes
Check Registered Groups	Yes	No	Yes	Yes	Yes
Change Registration Date	Yes	No	Yes	Yes	No

Description: The figure 2.3 shows the detailed analysis of related projects that we considered while developing PBL. Our project offers more features than these projects.

3. REQUIREMENTS ANALYSIS

3.1 Requirements

The requirements are based on the performance and functionalities the system must uphold to achieve the objective.

Table 3.1: Requirements Categories

RID	Requirement Type	Description
FR 1.x	Functional Requirements	These requirements are the core requirements of the project
NFR 2.x	Non-Functional Requirements	They play a supporting role for the functional requirements
PR 3.x	Programming Requirements	It describes the tools which are going to be used in this application
DR 4.x	Data Requirements	How the data will be stored in this application

3.2 List of Requirements

Table 3.2: List of Requirements

RID	Requirement Name
FR 1.1	Register
FR 1.2	Login
FR 1.3	Logout
FR 1.4	View FYP Criteria
FR 1.5	Update Marks
FR 1.6	Requests
NFR 2.1	Safety
NFR 2.2	Security
NFR 2.3	Usability
NFR 2.4	Compatibility
NFR 2.5	Responsive
NFR 2.6	Maintainability
NFR 2.7	Availability

Table 3.3: Functional Requirement & Non-Functional Requirement

RID	Description	Category	Attribute	Details & Boundary Constraints
FR 1.1	The system will allow the student to perform registration by entering details.	Functional Requirement	Register interface	<ul style="list-style-type: none"> ● Enter Student Name / ID ● Enter Email ● Enter Password ● Enter Project Title
FR 1.2	The system will allow the student to login into the system with student id and password.	Functional Requirement	Login Interface	<ul style="list-style-type: none"> ● Enter Student Name / ID / Email ● Enter Password
FR 1.3	The system will allow the evaluator to login into the system with email and password.	Functional Requirement	Login Interface	<ul style="list-style-type: none"> ● Enter Instructor Name / ID / Email ● Enter Password
FR 1.4	The system will allow the student to log out from the system.	Functional Requirement	Logout Interface	<ul style="list-style-type: none"> ● Click on the Logout
FR 1.5	The system will allow the Evaluator to log out from the system.	Functional Requirement	Logout Interface	<ul style="list-style-type: none"> ● Click on the Logout
FR 1.6	The system will allow the student to view FYP Criteria.	Functional Requirement	Criteria Interface	<ul style="list-style-type: none"> ● Click on the Project Evaluation Criteria ● View Criteria
FR 1.7	The system will allow the evaluator to view FYP Criteria.	Functional Requirement	Criteria Interface	<ul style="list-style-type: none"> ● Click on the Project Evaluation Criteria ● View Criteria

FR 1.8	The system will allow the evaluator to upload FYP marks.	Functional Requirement	Update Marks Interface	<ul style="list-style-type: none"> ● Go to the Result Updation ● Enter Marks of every criterion ● Click to Update
FR 1.9	The system will allow the student to view marks.	Functional Requirement	View Marks Interface	<ul style="list-style-type: none"> ● Click on the Project Evaluation Result ● View Marks of Every Criteria
FR 1.10	The system will allow the evaluator to view pending requests.	Functional Requirement	View Requests Interface	<ul style="list-style-type: none"> ● Check New Project ID
FR 1.11	The system will allow the evaluator to approve the requests	Functional Requirement	Approve Requests Interface	<ul style="list-style-type: none"> ● Approve New Project ID
NFR 2.1	The system will only allow user to login only after registration.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● Our website will be safe for everyone.
NFR 2.2	Only authorized users who log in via the login section are allowed access by the system.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● Student will not encounter any security issue.
NFR 2.3	Users of the system won't need any training. The system's ui will be responsive.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● Our web will be easy to use.
NFR 2.4	The system will be compatible with all the browsers which have enabled JavaScript.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● Our web will be compatible with any browser.
NFR 2.5	The system must be suitable for all types of end users.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● Our web will be adjustable on any screen size.
NFR 2.6	Because to the software's programme design, it is simple to maintain and fix bugs.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> ● We will maintain the website for future use.

NFR 2.7	The system documentation will be available and maintained for future use.	Non-Functional Requirement	System Functionality	<ul style="list-style-type: none"> Our web will be available for the students all the time
------------	---	----------------------------	----------------------	---

3.3 List of Actors

List of actors are given below:

Admin: All the app activities controlled and monitored by admin.

Student: who will register its project and gets evaluated.

Evaluator: Who checks the project and upload the marks.

3.4 List of Actors

List of actors are given below:

Admin: All the app activities controlled and monitored by admin.

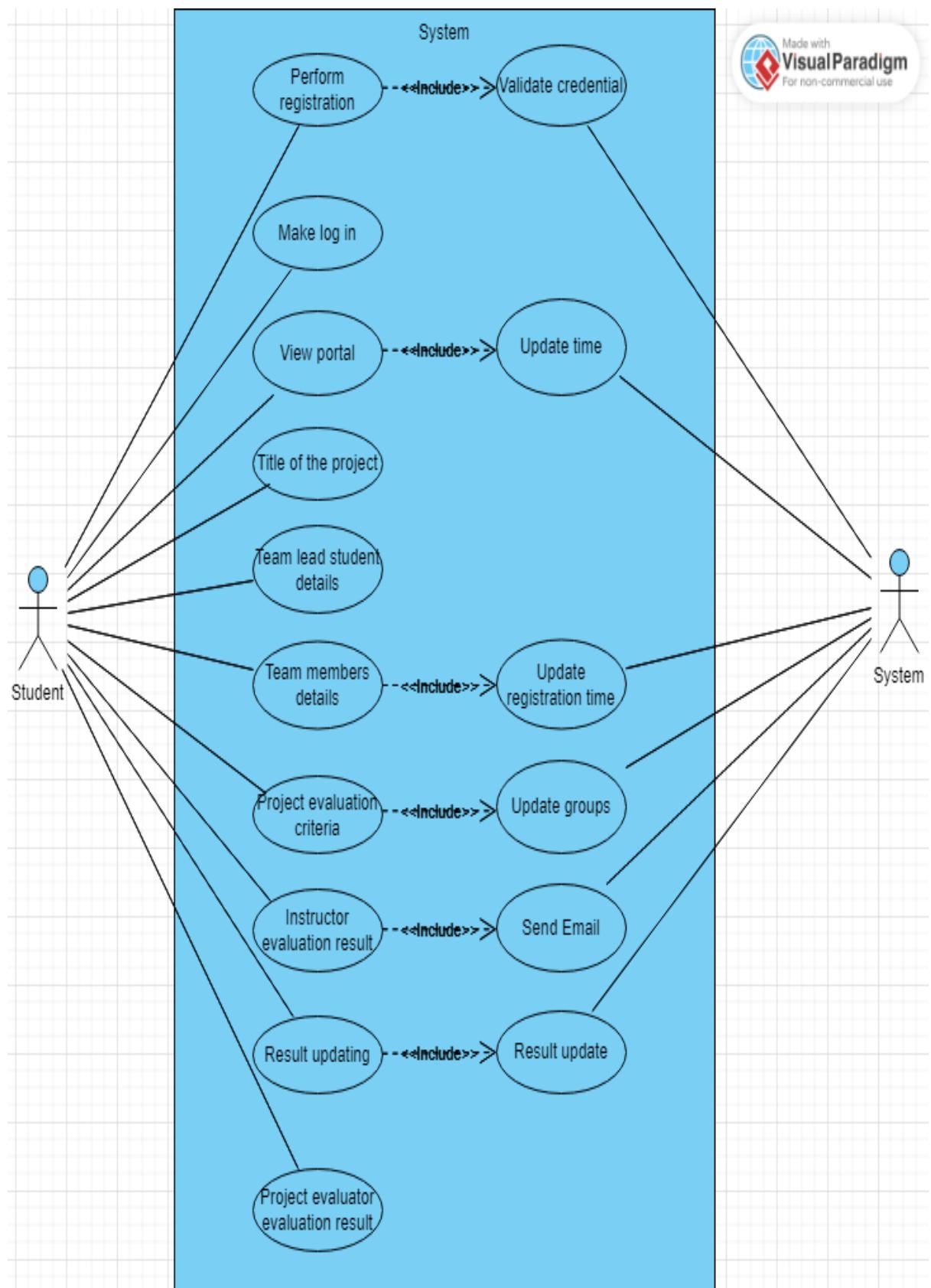
Student: who will register its project and gets evaluated.

Evaluator: Who checks the project and upload the marks.

3.5 List of Use Cases:

- **UC-1** Perform registration
- **UC-2** Make Login
- **UC-3** Logout
- **UC-4** View portal
- **UC-5** Title of the project
- **UC-6** Team leader details
- **UC-7** Team member's details
- **UC-8** Project evaluation criteria
- **UC-9** Instructor evaluation result
- **UC-10** Project evaluator evaluation result
- **UC-11** Set registration time
- **UC-12** Dashboard
- **UC-13** Check user
- **UC-14** Project evaluation criterion
- **UC-15** Evaluate groups
- **UC-16** Project external evaluator
- **UC-17** Result updating
- **UC-18** Project result
- **UC-19** Final Result

3.6 System use case diagram



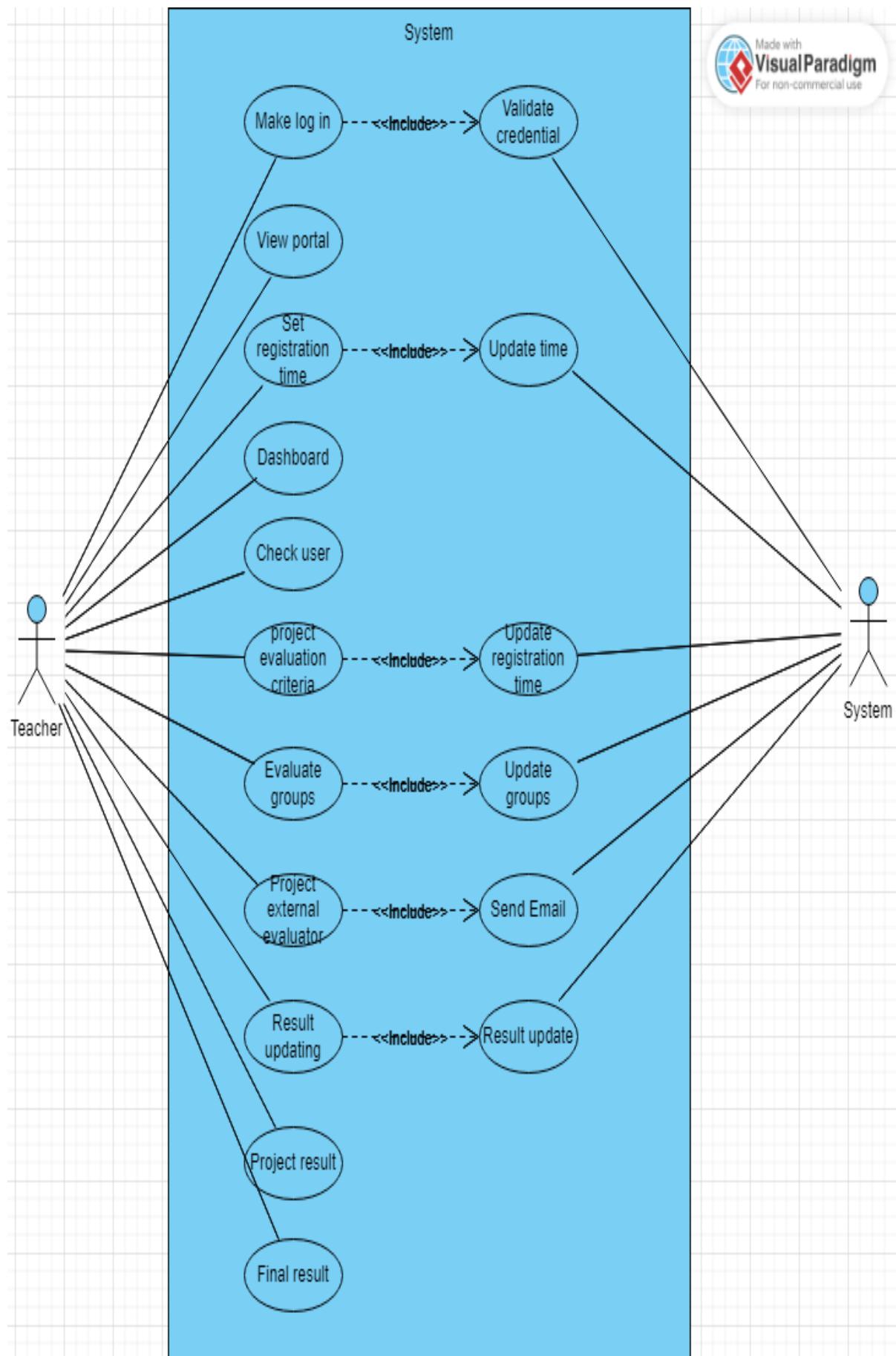


Figure 3.1: Use Case Diagram with Explanation and scope of our system

3.7 Extended Use Cases

Table 3.4: UC-1 (Perform registration)

UC - ID	01
UC - Name	Perform registration
Actor	Student
Description	This process describe that the related actor registers themselves.
Pre-condition	The actor must not register and have no record in database before.
Post-condition	Actor should be registered successfully and have record in database.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will open the website. 2. The actor will click on "Registration" button. 3. The actor enters his/her group detail. 4. The actor will click on register button and will successfully register.
Alternative flow	3a. if actor is already registered then redirect to log in instead.
exception	<<include>>

Description: This process describe that the related actor registers themselves.

Table 3.5: UC-2 (Make Log In)

UC - ID	02
UC - Name	Make Log In
Actor	Students, Supervisors
Description	This use case describes that the related actor log in themselves.
Pre-Condition	The actor needs to be authorized to register.
Post-Condition	The actor should be log in successfully into the system to access certain functionalities.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will open the website. 2. The actor will select "Log in" from the menu. 3. The actor will type in their username and password before hitting the "log in" button. 4. The system will verify id and password and lead actor to his/her profile.
Alternative flow	3a. inform actor if id or password do not match any record in the database.
Exceptions:	<<include>>

Description: This use case describes that the related actor log in themselves.

Table 3.6: UC-03 (Logout)

UC - ID:	03
UC - Name:	Logout
Actors:	Student, instructor
Description:	This describe that the related actor will logout from the system.
Pre - Condition:	The actor has to have a login for the system.
Post - Condition:	The actor needs to successfully log out of the system.
Normal Flow of Events:	<ol style="list-style-type: none">1. The actor will click “Logout” button.2. The actors will logout from the system.3. The system will redirect the actor to the login page.
Alternative flows:	3.a Inform actor if logout will fail.
Exceptions:	None

Description: This use case describe that the related actor will logout from the system.

Table 3.7: UC-04 (View Portal)

UC - ID	04
UC - Name	View Portal
Actor	Student, instructor
Description	This describes that the related actor log in themselves.
Pre-condition	The actor must have an account before.
Post-condition	The actor should be viewed portal.
Normal flow of event	<ol style="list-style-type: none">1. The actor will open the website.2. The actor will click on “Log in” button.3. The actor will enter his/her id and password and click on log in button.4. The system will verify id and password and lead actor to his/her profile.
Alternative flow	3a. inform actor if id or password do not match any record in the database.
exception	None

Description: This use case describes that the related actor log in themselves.

Table 3.8: UC-05 (Title of the Project)

UC - ID	05
UC - Name	Title of the Project
Actor	Student
Description	This describe that the related actor can see their project title.
Pre-condition	The actor must have title of the project.
Post-condition	The actor should be viewed title of the project.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will select "Title of the Project" by clicking. 2. The name and ID of the project will be shown by the system.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see their project title.

Table 3.9: UC-06 (Team Leader Details)

UC - ID	06
UC - Name	Team Leader Details.
Actor	Student
Description	This describe that the related actor can see their profile.
Pre-condition	The actor must have an account before.
Post-condition	The actor should be viewed team lead student details successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click on "Team lead student details". 2. The system will show the title of the project and id of the project.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see their profile.

Table 3.10: UC-07 (Team Members Details)

UC - ID	07
UC - Name	Team Members Details
Actor	Student
Description	This describe that the related actor can see their group members details.
Pre-condition	The actor must have team members.
Post-condition	The actor should be viewed team members details successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click on "Team members' details". 2. The system will show team members' details.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see their group members' details.

Table 3.11: UC-08 (Project Evaluation Criteria)

UC - ID	08
UC - Name	Project Evaluation Criteria.
Actor	Student
Description	This describe that the related actor can see project evaluation criteria specify by the instructor.
Pre-condition	The actor must have an instructor.
Post-condition	The actor should be viewed project evaluation criteria successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Project evaluation criteria”. 2. The system will show project evaluation criteria.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see project evaluation criteria specify by the instructor.

Table 3.12: UC-09 (Instructor Evaluation Result)

UC - ID	09
UC - Name	Instructor Evaluation Result
Actor	Student
Description	This describe that the related actor can see instructor evaluation result.
Pre-condition	The actor must submit his/her project idea.
Post-condition	The actor should be viewed instructor evaluation result successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “instructor evaluation result”. 2. The system will show instructor evaluation result.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see instructor evaluation result.

Table 3.13: UC-10 (Project Evaluator Evaluation Result)

UC - ID	10
UC - Name	Project Evaluator Evaluation Result
Actor	Student
Description	This describe that the related actor can see Project evaluator evaluation result.
Pre-condition	The instructor must approve his/her project idea.
Post-condition	The actor should be viewed instructor evaluation result successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Project evaluator evaluation result”. 2. The system will show Project evaluator evaluation result.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see Project evaluator evaluation result.

Table 3.14: UC-11 (Set Registration Time)

UC - ID	11
UC - Name	Set Registration Time
Actor	instructor
Description	This describe that the related actor can set registration time of project.
Pre-condition	The registration time must not be set before.
Post-condition	Registration time should be updated successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will set date from calendar and press “Update” button. 2. The system will update the registration time.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can set registration time of project.

Table 3.15: UC-12 (Dashboard)

UC - ID	12
UC - Name	Dashboard
Actor	instructor
Description	This describe that the related actor can view total registered groups, total projects and evaluated groups.
Pre-condition	The actor must have registered groups and evaluated groups.
Post-condition	The actor viewed details successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Dashboard”. 2. The system will show registered groups, total projects and evaluated groups.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can view total registered groups, total projects and evaluated groups.

Table 3.16: UC-13 (Check User)

UC - ID	13
UC - Name	Check User
Actor	instructor
Description	This describe that the related actor can view complete detail of registered groups.
Pre-condition	The actor must have registered groups.
Post-condition	The actor viewed details successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Check user”. 2. The system will show complete detail registered groups.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can view complete detail of registered groups.

Table 3.17: UC-14 (Project Evaluation Criterion)

UC - ID	14
UC - Name	Project Evaluation Criterion
Actor	instructor
Description	This describe that the related actor will set project criteria for registration.
Pre-condition	
Post-condition	The project criteria successfully updated.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “project evaluation criterion”. 2. The actor will enter rules and regulations for project and click on “Update” button. 3. The system will update the project evaluation criteria.
Alternative flow	None
exception	None

Description: This use case describe that the related actor will set project criteria for registration.

Table 3.18: UC-15 (Evaluate Groups)

UC - ID	15
UC - Name	Evaluate Groups
Actor	instructor
Description	This describe that the related actor will evaluate the requested groups.
Pre-condition	The related groups must not be registered before.
Post-condition	The groups should be evaluated successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “evaluate groups”. 2. The actor will enter marks (0-5) for project evaluation and click on “Update” button.
Alternative flow	None
exception	None

Description: This use case describe that the related actor will evaluate the requested groups.

Table 3.19: UC-16 (Project External Evaluator)

UC - ID	16
UC - Name	Project External Evaluator
Actor	instructor
Description	This describe that the related actor can invite external evaluator to evaluate student’s project.
Pre-condition	The related evaluator must not be invited before.
Post-condition	The requested email must be sent successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Project external evaluator”. 2. The actor will enter external evaluator details and click “send invite” button. 3. The system sends an email to related evaluator.
Alternative flow	3a. Inform the actor if the email is not sent.
exception	None

Description: This use case describe that the related actor can invite external evaluator to evaluate student’s project.

Table 3.20: UC-17 (Result Updating)

UC - ID	17
UC - Name	Result Updating
Actor	instructor
Description	This describe that the related actor will enter project marks.
Pre-condition	The related project must not be updated.
Post-condition	The related project result should be updated.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Result updating”. 2. The actor will click specific id. 3. The actor will enter marks (0-5) for project result updatation and click on “Update” button.
Alternative flow	3a. Inform the actor if result is not updated.
exception	None

Description: This use case describe that the related actor will enter project marks.

Table 3.21: UC-18 (Project Result)

UC - ID	18
UC - Name	Project Result
Actor	instructor
Description	This describe that the related actor can see the whole result.
Pre-condition	The related project must be updated.
Post-condition	The project result should be viewed successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Project result”. 2. The system will show project result.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see the whole result.

Table 3.22: UC-19 (Final Result)

UC - ID	19
UC - Name	Final Result
Actor	instructor
Description	This describe that the related actor can see the final absolute result.
Pre-condition	The related project must be updated result.
Post-condition	The actor should be viewed final result successfully.
Normal flow of event	<ol style="list-style-type: none"> 1. The actor will click “Final result”. 2. The system will show the final results.
Alternative flow	None
exception	None

Description: This use case describe that the related actor can see the final absolute result.

3.8 User interfaces (mock screens)

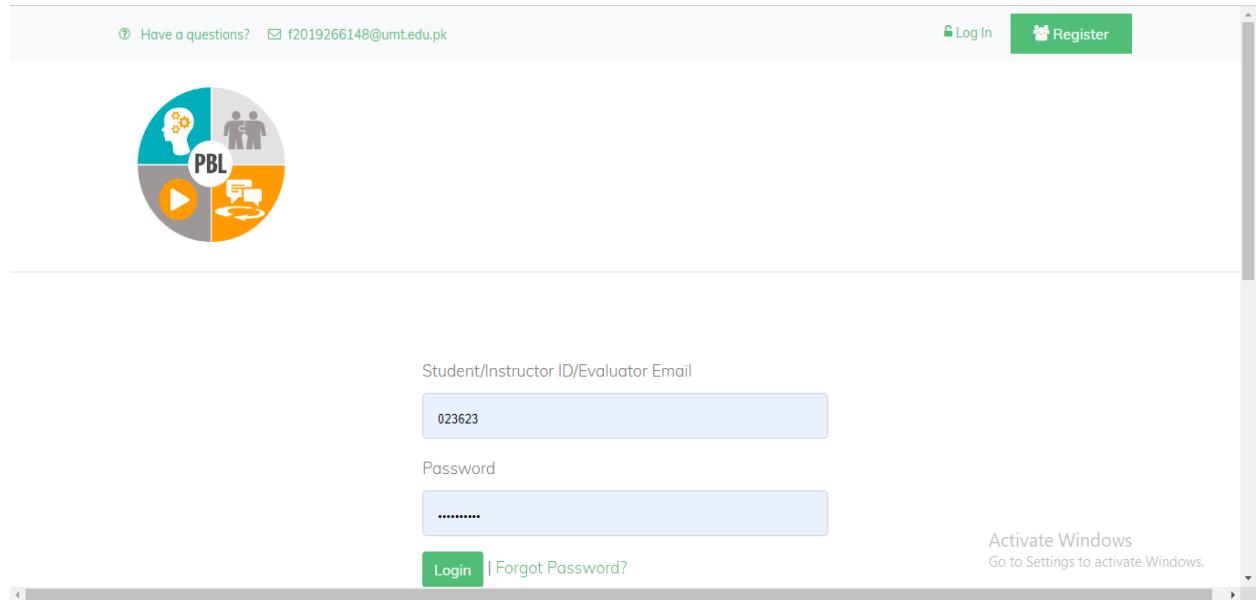


Figure 3.2: Login Page Interface

Description: The above Snapshot is the Login Page of users.

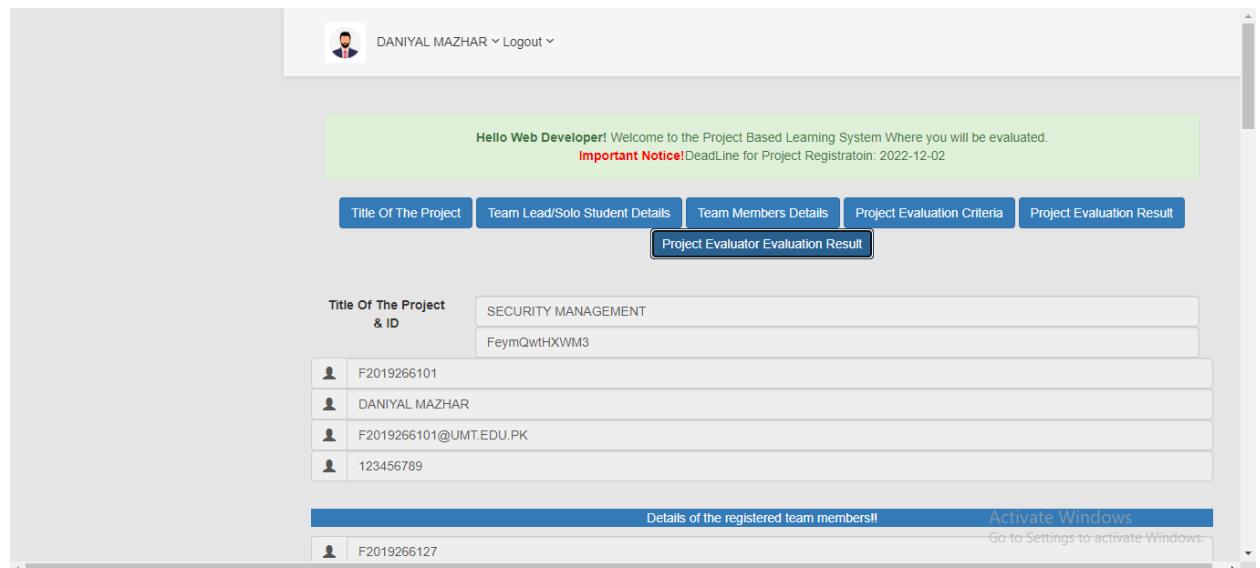


Figure 3.3: Student Portal Interface

Description: The above Snapshot is the Dashboard of the Student Portal side.

Figure 3.4: Advisor Interface

Description: The above Snapshot is the Dashboard of the Advisor side.

Figure 3.5: Evaluator Interface

Description: The above Snapshot is the Dashboard of the Evaluator side.

4. DATA FLOW DIAGRAM

4.1 Data Flow Diagram Level 0

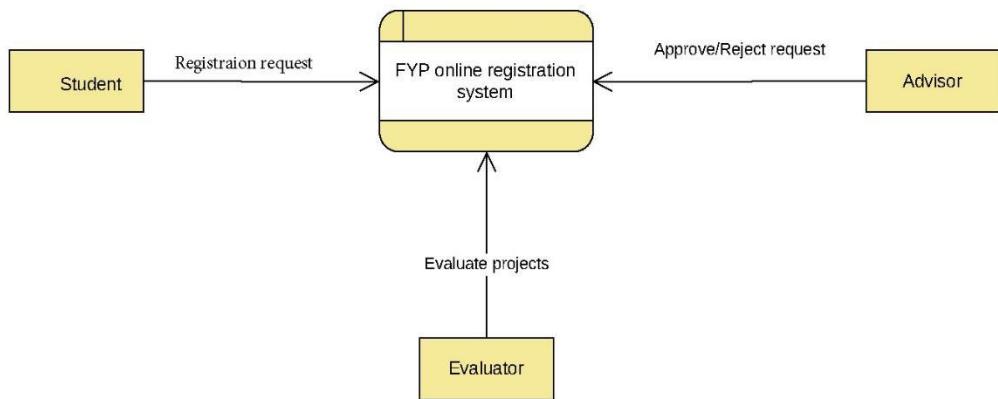


Figure 4.1: DFD Level 0 Diagram

Context Diagram is another name for DFD Level 0. It provides a general summary of the entire procedure or system that is being studied.

4.2 Data Flow Diagram Level 1

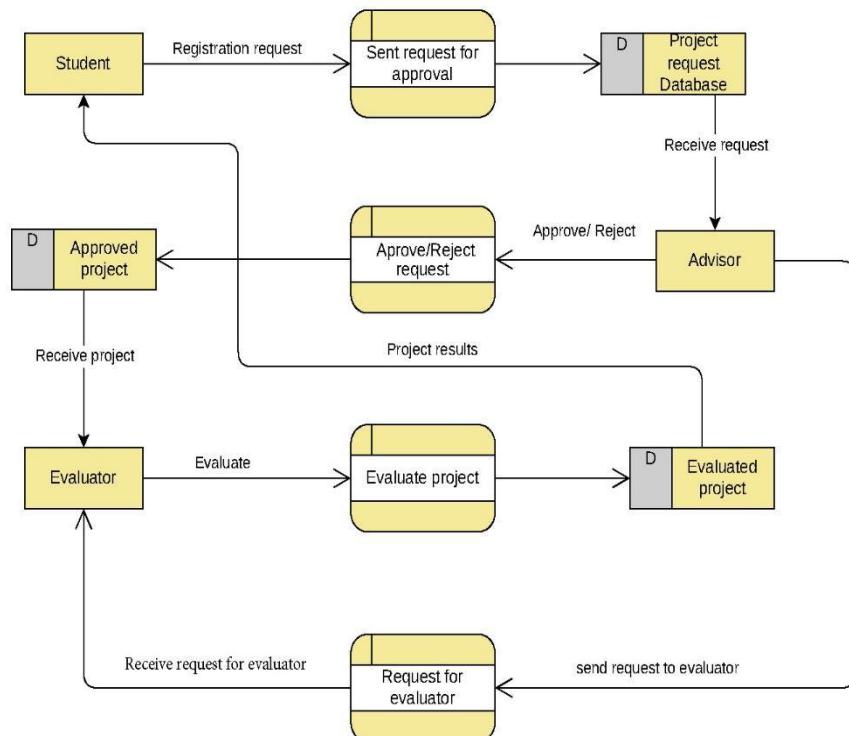


Figure 4.2: DFD Level 1 Diagram

Each of the primary sub-processes that together make up the entire system are listed on a level 1 DFD.

5. SYSTEM DESIGN

5.1 Work Break Down Structure (WBS)

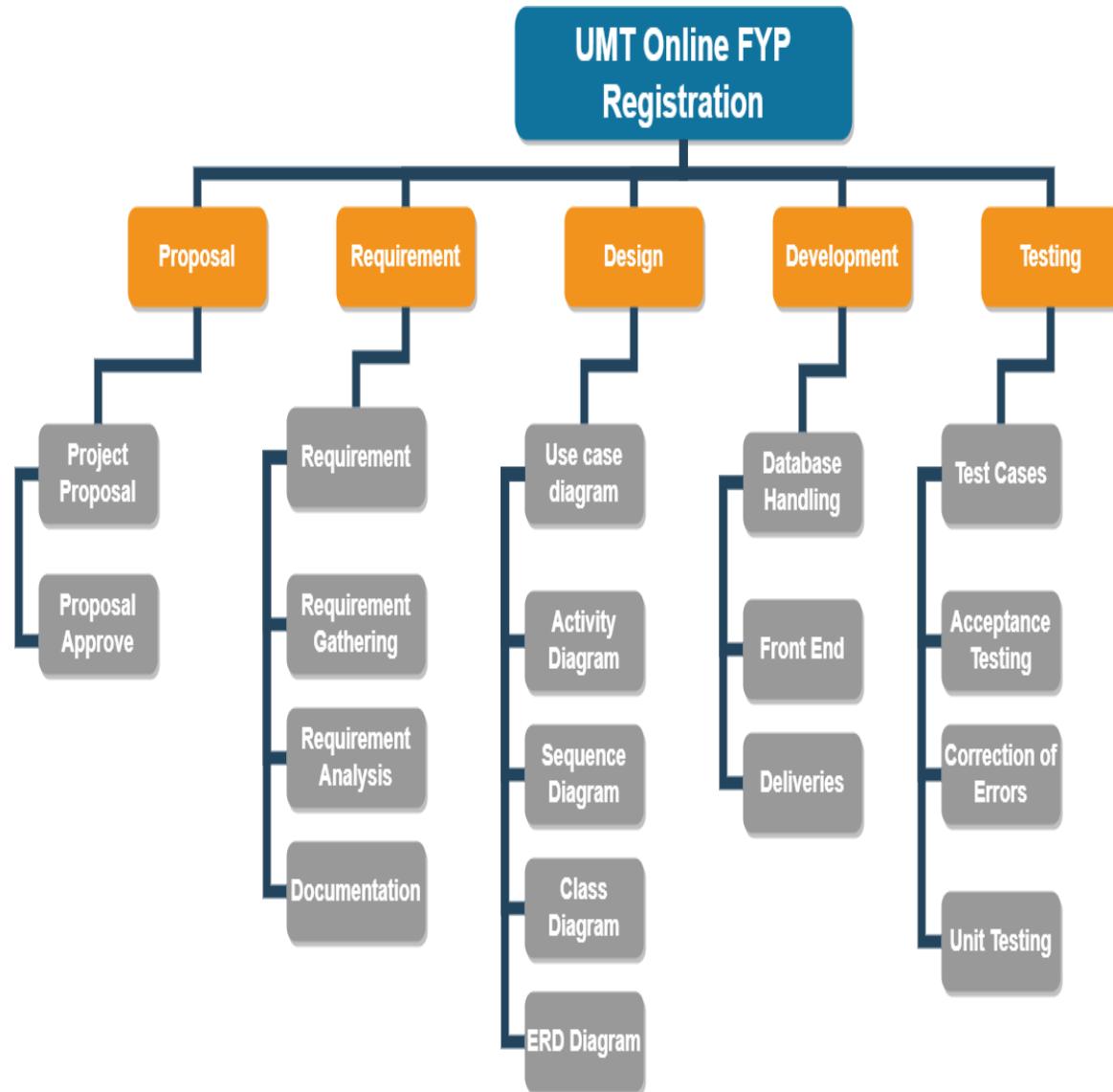


Figure 5.1: Work Break Down Diagram with Explanation

In this WBS Diagram we have broken our work into phases. From first phase of proposal creation, submission to Testing

5.2 Activity Diagram

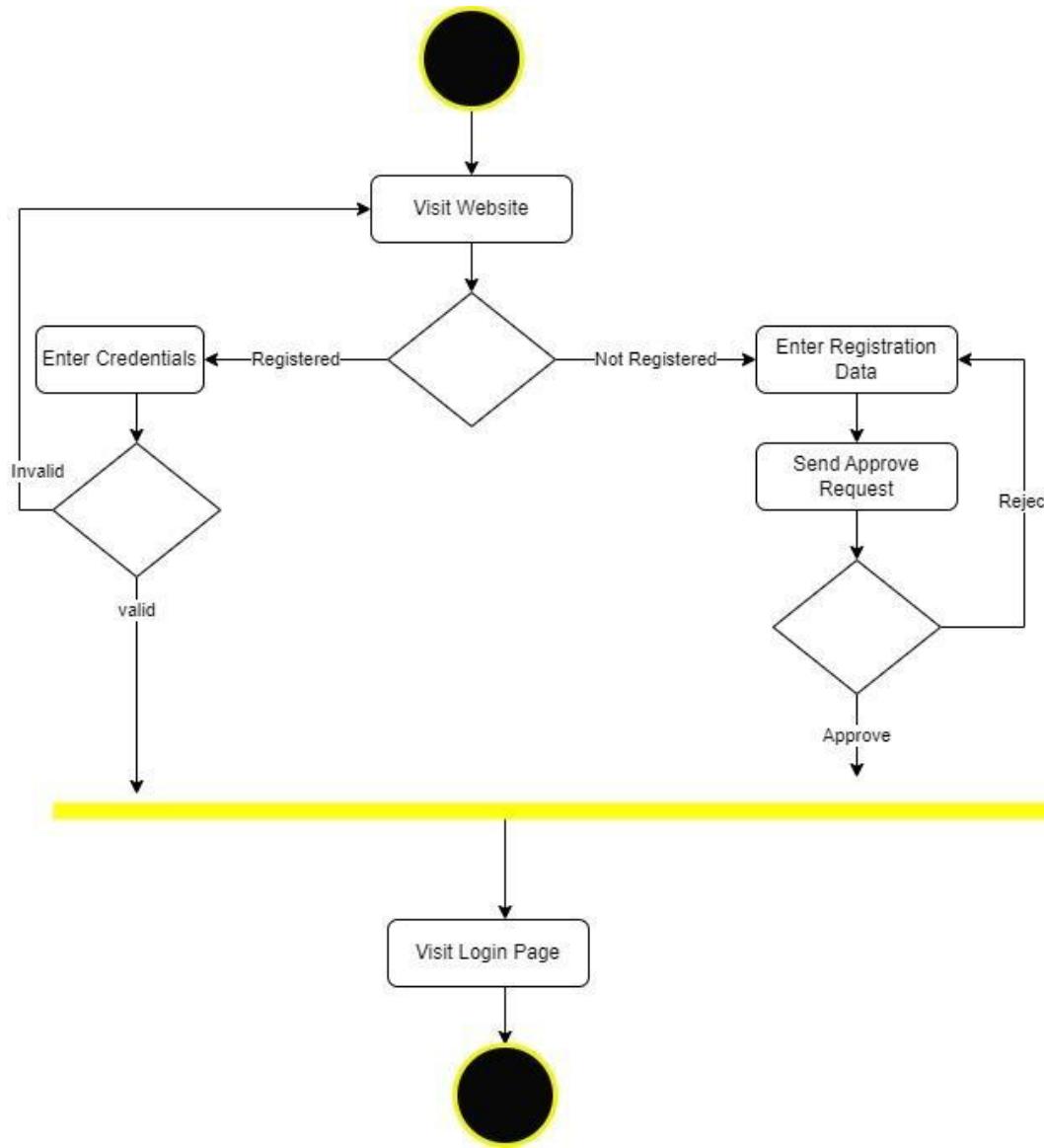


Figure 5.2: Activity Diagram for Login/Signup

In this Activity Diagram it is shown that when a User Visit Website to Register. The system checks if he already registers or not.

If Already Registered, he enters its credentials to Login. Credentials must be valid to Login otherwise not Login. If not Registered, then he enters project data of his and group members name to send project data. The send data gets Approved he can Login Otherwise Reject.

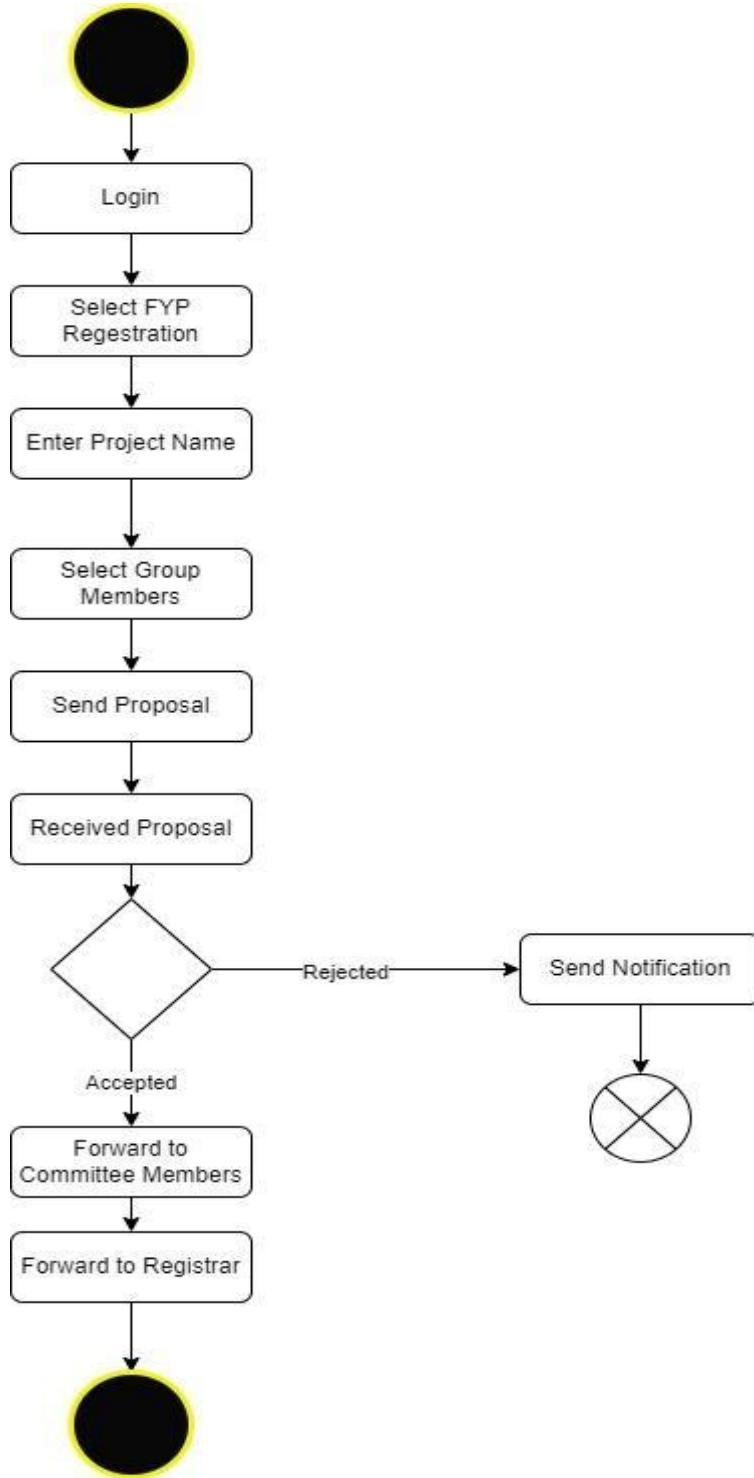


Figure 5.3: Activity Diagram for FYP registration

In this Activity Diagram it is shown that in which steps User gets Registered.

First, He go to Login Page, then Select Register Button to Register. User enters Project Name, Group Members IDs and Names. Then Click Submit Button to Register. On Evaluator side, If Project is approved it is sent it to the other committee members or if rejected a notification is send to the student.

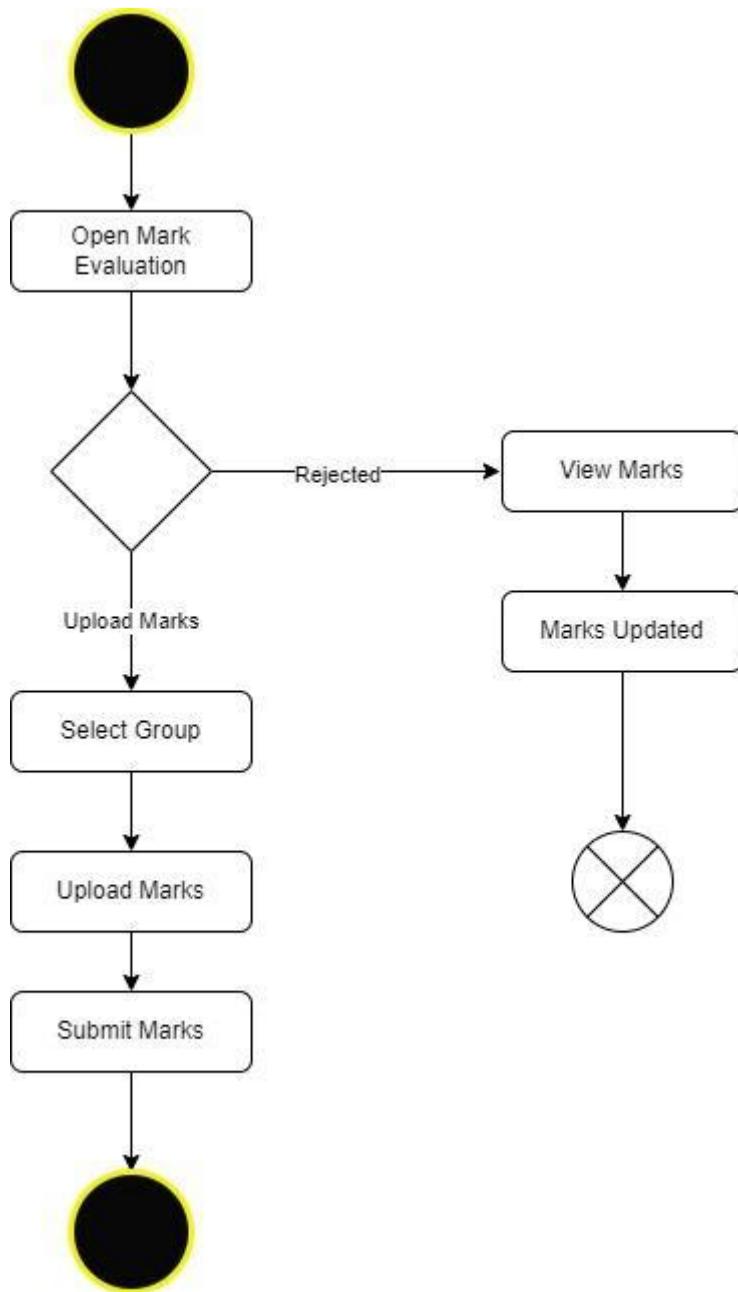


Figure 5.4: Activity Diagram for view and upload marks

In this Activity Diagram it is shown that how can Evaluator upload marks to the groups.

First, He click on the CHECK USER Button. All the groups will be appeared. If he already uploads the Marks, he can update them. If he has not uploaded the marks, he selects a group member. Then Upload the Marks of each Criteria of each group.

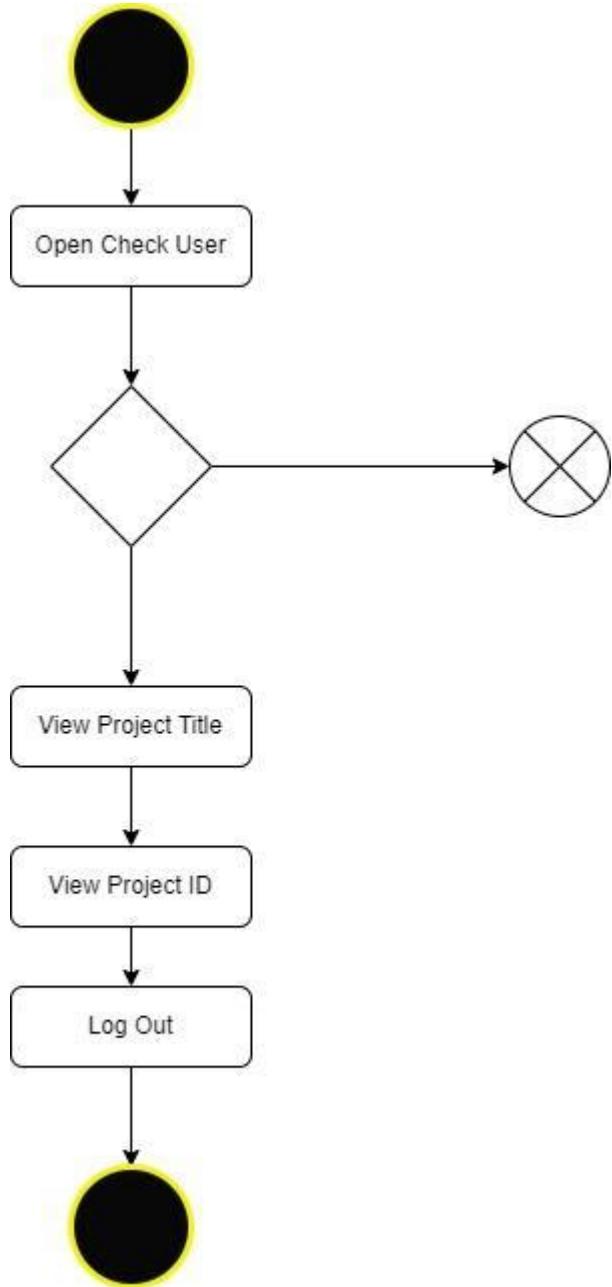


Figure 5.5: Activity Diagram for view FYP view groups

In this Activity Diagram it is shown that how can Evaluator check all the Registered groups.

First, He click on the CHECK USER Button. All the Registered groups will be appeared. Each Group Has Unique Project ID, Project Title, Group Leader Name and ID.

5.3 Class Diagram

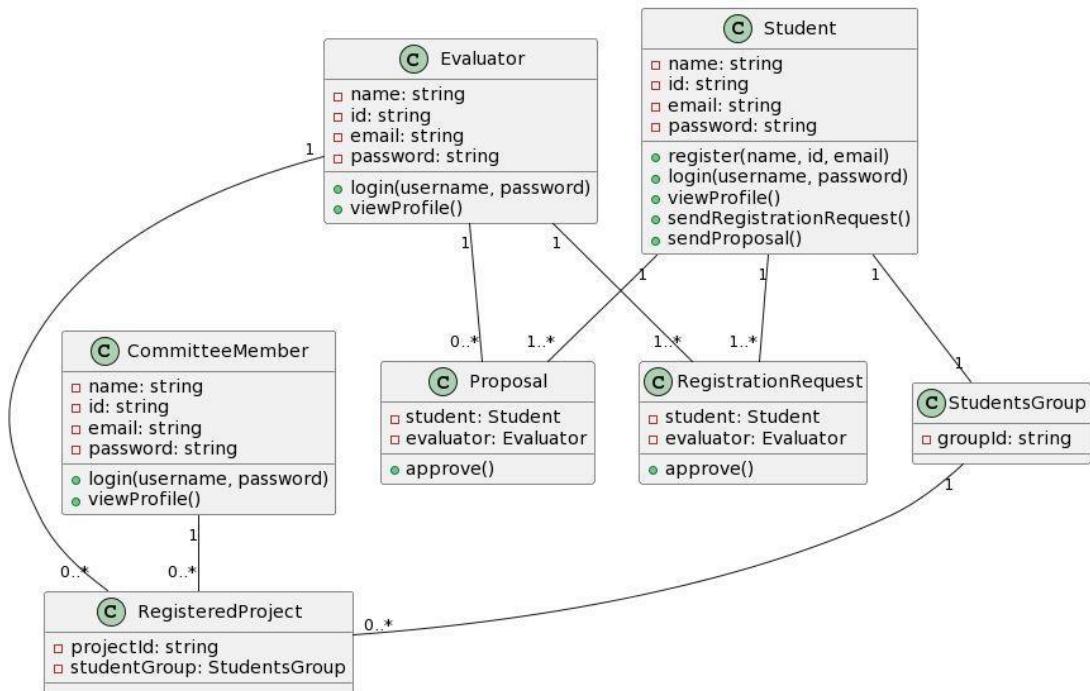


Figure 5.6: Class Diagram with Explanation

In this class diagram dependency of each class shown to other class.

Evaluator with registered name, id, email can view the Profile. Student with name, id, email will perform registration and can Login. Student his Registration Request and will be Approved by Evaluator. Student send his proposal and Evaluator gets it. Student Group gets Auto unique Project Id that can be viewed by both Evaluator and Committee Member.

5.4 Collaboration Diagram

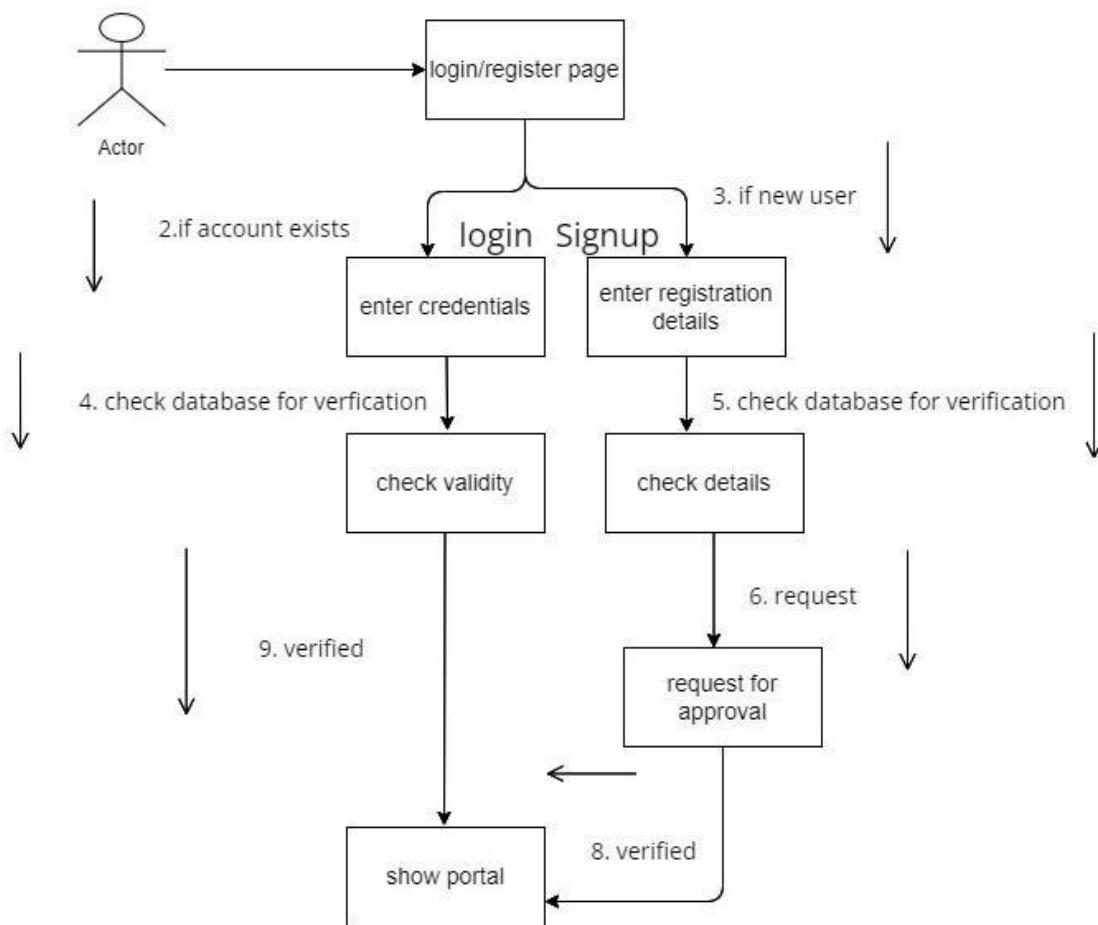


Figure 5.7: Collaboration Diagram for Student to Sign up

In this Collaboration Diagram it is shown that when a User Visit Website to Register. The system checks if he already registers or not.

If Already Account Exists, he enters its credentials to Login. Credentials must be valid to Login otherwise not Login. If Account not exits, then he enters project data of his and group members name to send project data. The send data gets Approved he can Login. Otherwise Reject. Then a pop menu tells you that you are registered.

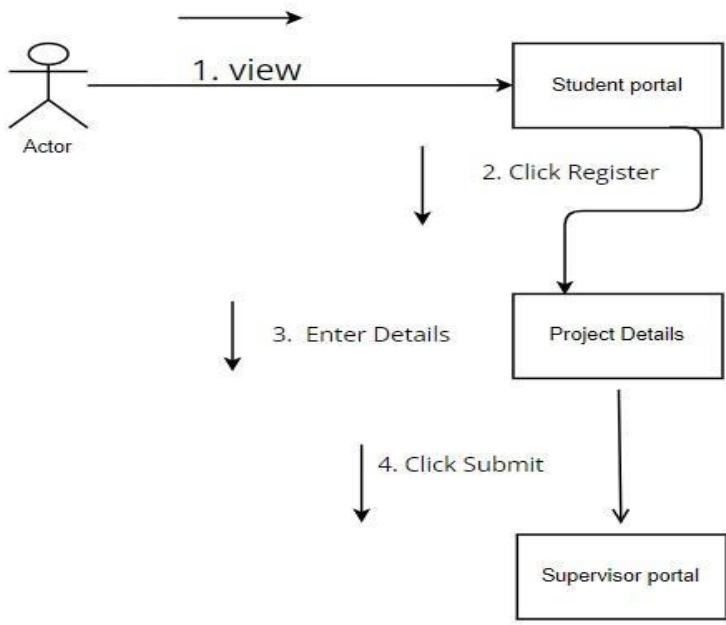


Figure 5.8: Collaboration Diagram for a student to Register

In this Collaboration Diagram it is shown that in which steps User gets Registered.

First, He Select Register Button to Register. User enters Project Name, Group Members IDs and Names. Then Click Submit Button to Register. On Evaluator side, If Project is approved it is sent it to the other committee members.

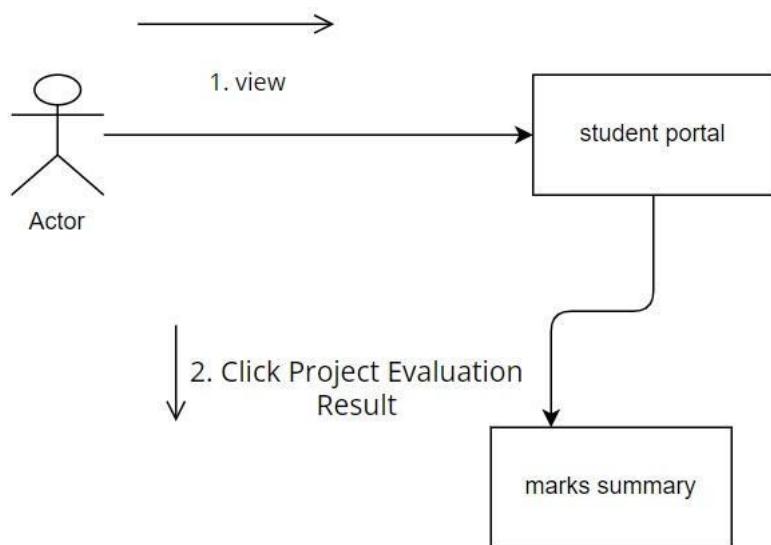


Figure 5.9: Collaboration Diagram for a student to View Marks

In this Activity Diagram it is shown that how can Evaluator upload marks to the groups. First, He click on the Project Results Button. The Evaluated groups will be appeared with their marks. If he has not uploaded the marks, he selects a group member. Then Upload the Marks of each Criteria of each group.

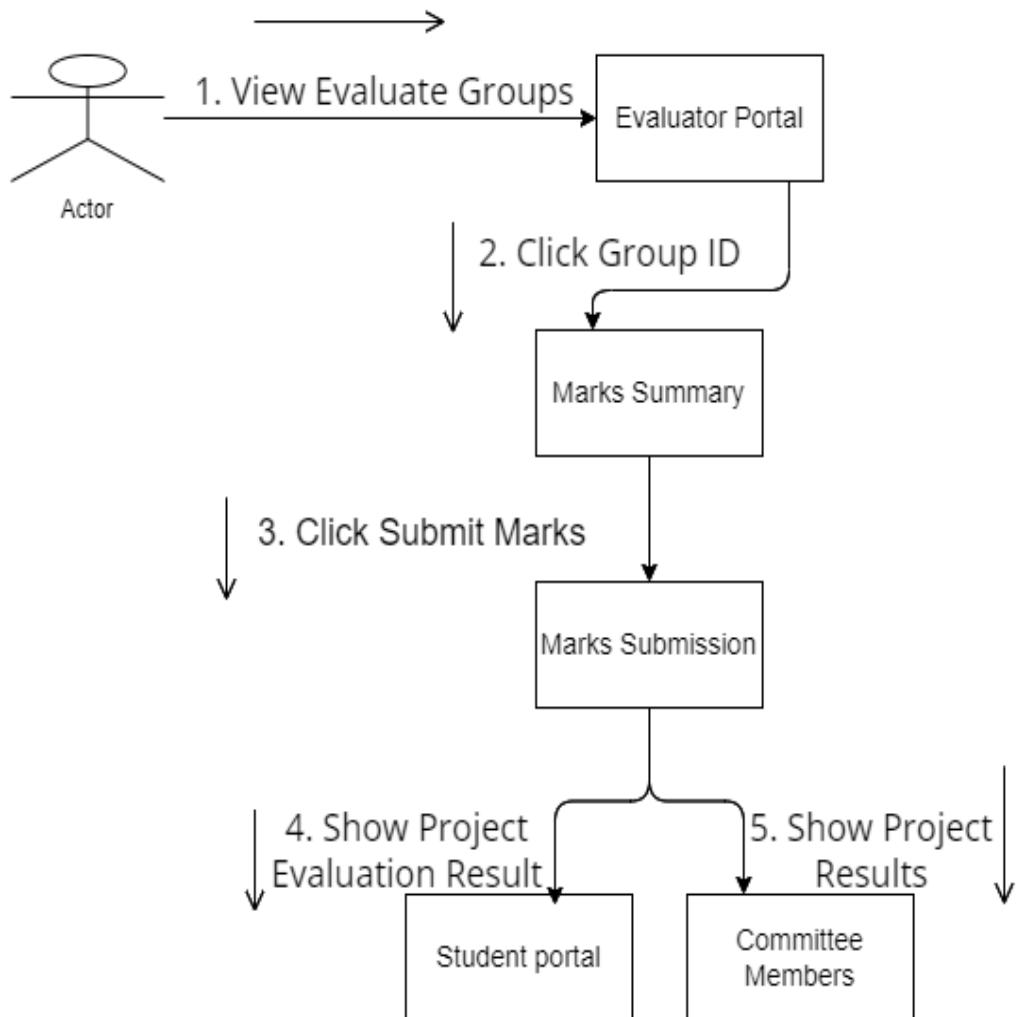


Figure 5.10: Collaboration Diagram for Evaluator to Update Marks

In this collaboration Diagram it is shown how to Evaluator can update Marks.

To update Marks, Evaluator click on the group id and the previous marks summary of each criterion is show. Evaluator enter Marks in it. Click on Update after entering. Updated Result is shown to both student and committee members.

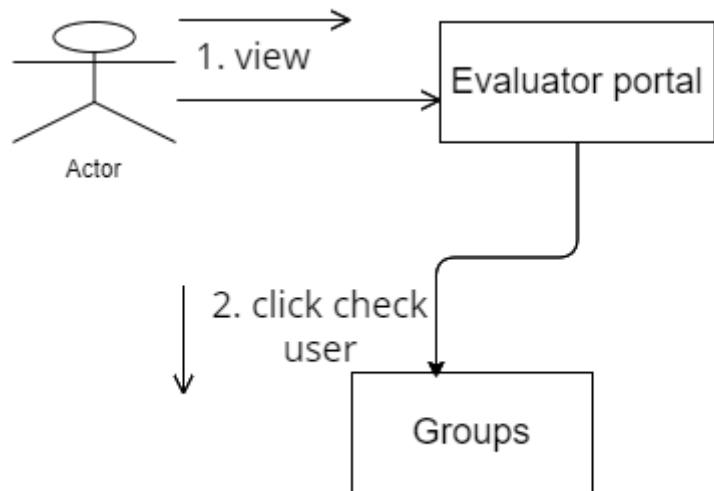


Figure 5.11: Collaboration Diagram for Evaluator to check all Registered Groups

In this Collaboration Diagram it is shown that how can Evaluator check all the Registered groups. First, He click on the CHECK USER Button. All the Registered groups will be appeared. Each Group Has Unique Project ID, Project Title, Group Leader Name and ID.

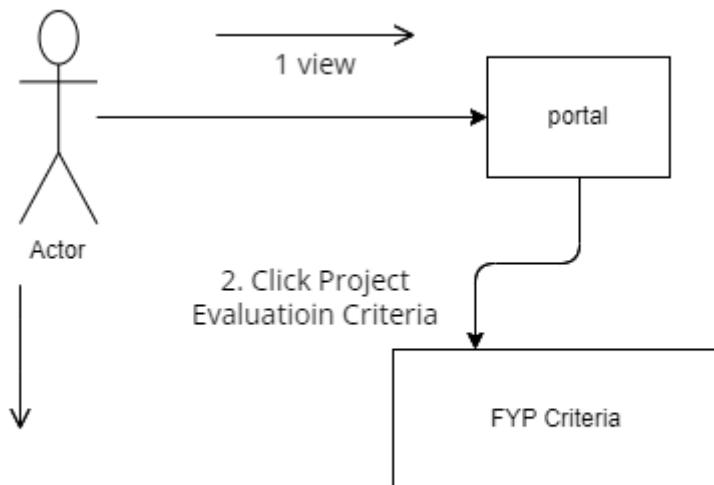


Figure 5.12: Collaboration Diagram for a student to View FYP Criteria

In this Collaboration Diagram it is shown that how Evaluator can check Evaluation Criteria. First, He click on the Project Evaluation Criteria Button. Evaluation Criteria is shown. Evaluator can change it.

5.5 Sequence Diagrams

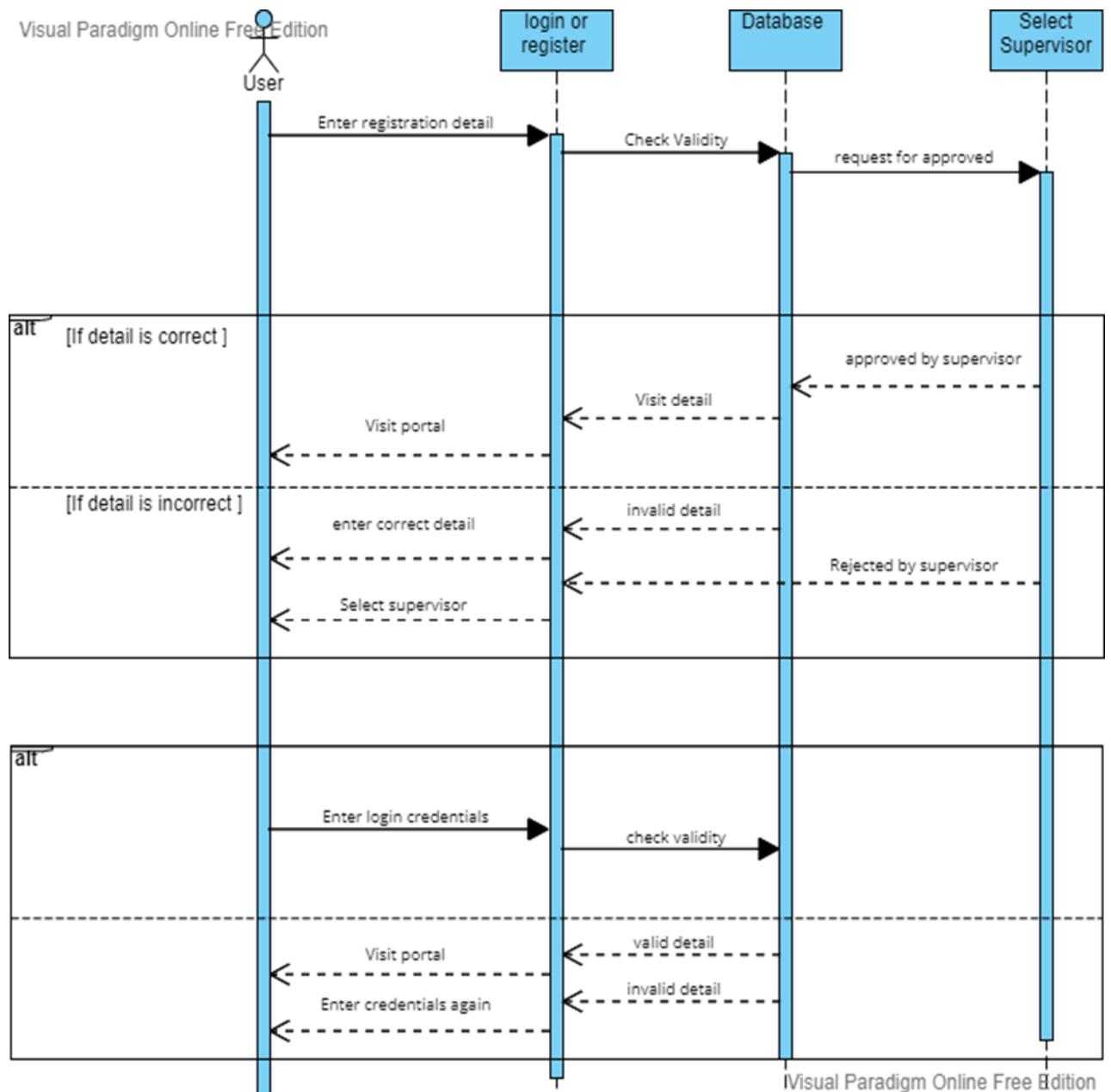


Figure 5.13: Sequence Diagram for login/register

In this Sequence Diagram it is shown that when a User Visit Website to Register. The system checks if he already registers or not.

If Already Registered, he enters its credentials to Login. Credentials must be valid to Login otherwise not Login. If not Registered, then he enters project data of his and group members name to send project data. The send data gets Approved he can Login Otherwise Reject.

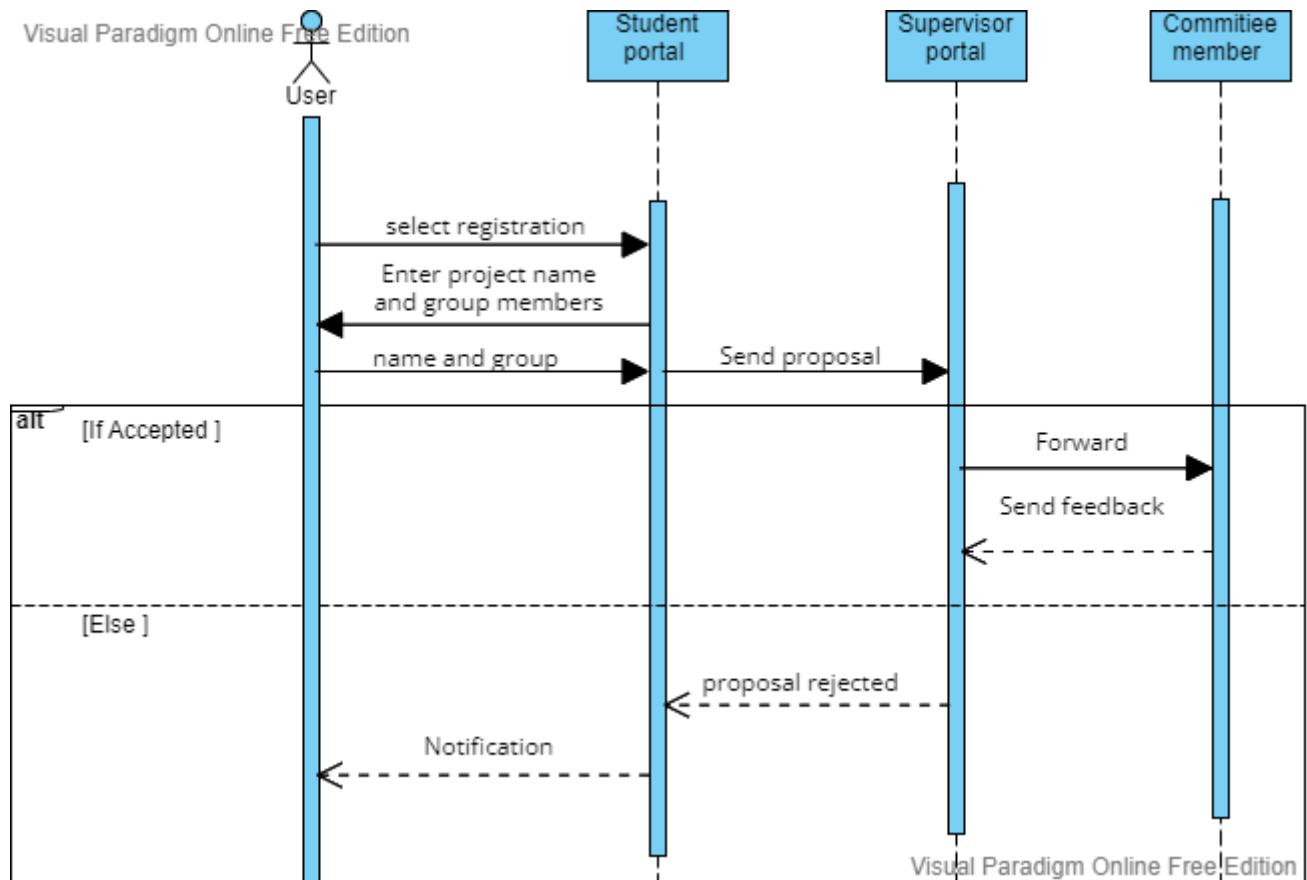


Figure 5.14: Sequence Diagram for FYP registration

In this Sequence Diagram it is shown that in which steps User gets Registered.

First, He go to Login Page. Then Select Register Button to Register. User enters Project Name, Group Members IDs and Names. Then Click Submit Button to Register. On Evaluator side, If Project is approved it is sent it to the other committee members or if rejected a notification is send to the student.

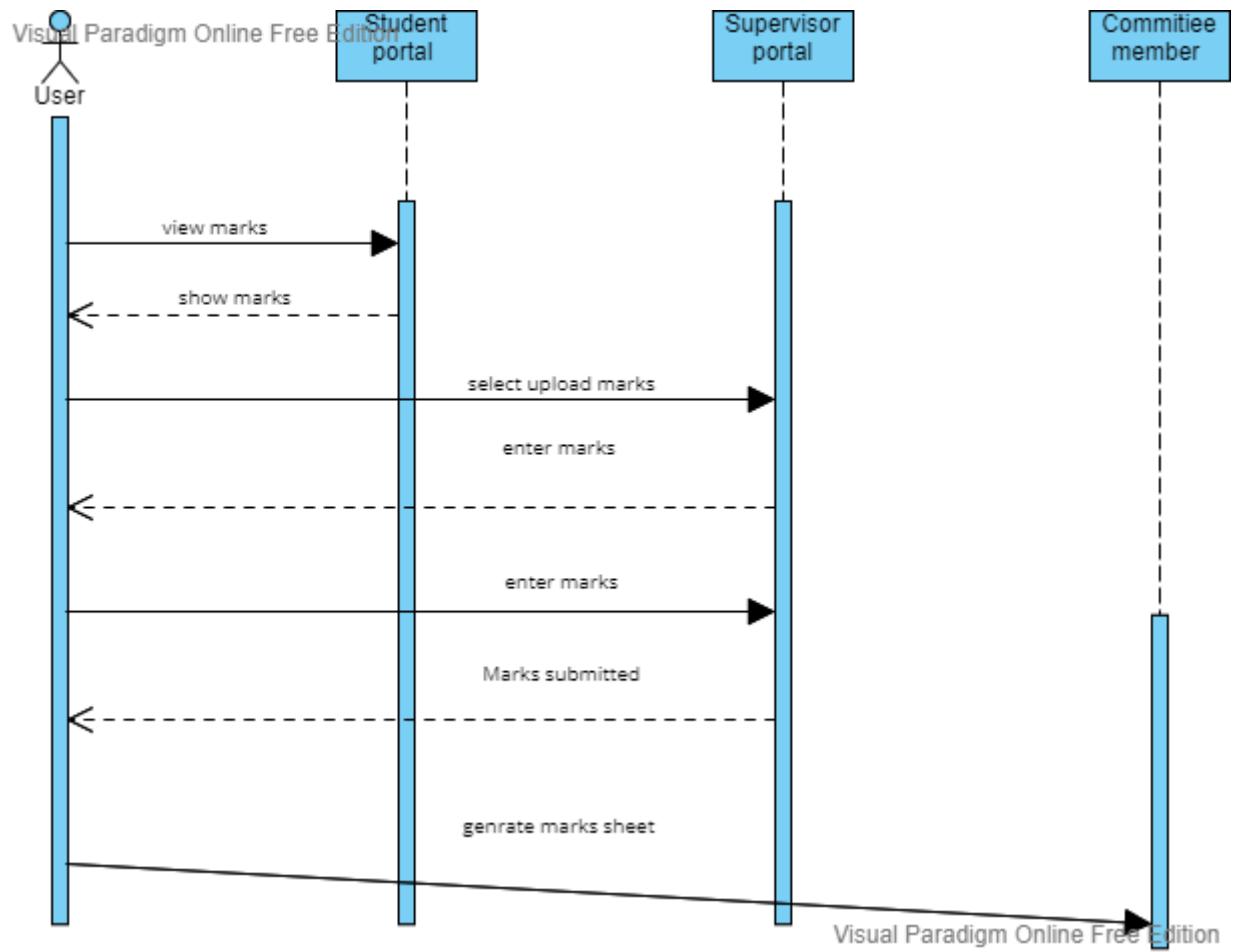


Figure 5.15: Sequence Diagram for view and upload marks

In this Sequence Diagram it is shown that how can Evaluator upload marks to the groups.

First, He click on the CHECK USER Button. All the groups will be appeared. If he already uploads the Marks, he can update them. If he has not uploaded the marks, he selects a group member. Then Upload the Marks of each Criteria of each group.

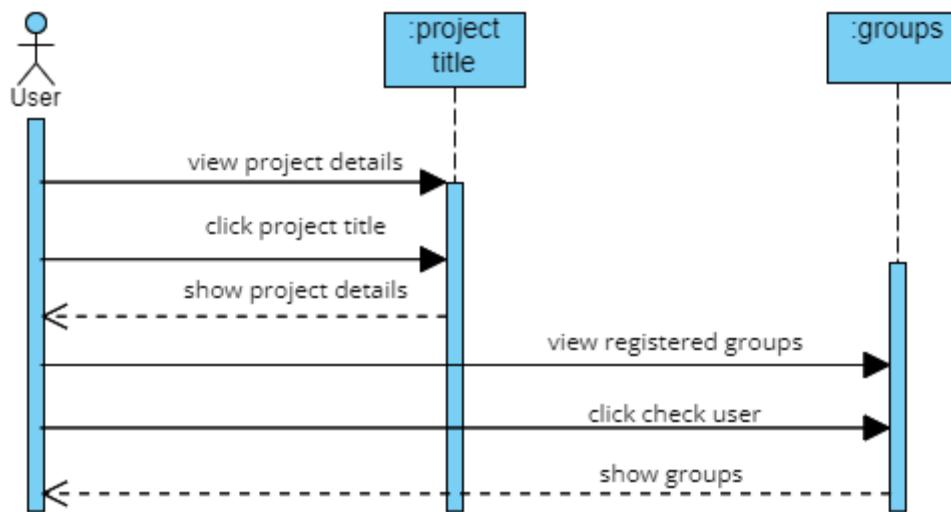


Figure 5.16: Sequence Diagram for View Registered Groups

In this Sequence Diagram it is shown that how can Evaluator check all the Registered groups. To view project details of each group, click on project title. Project title will be shown To view registered groups, click on CHECK USER Button. All the Registered groups will be appeared.

5.6 Database Diagram

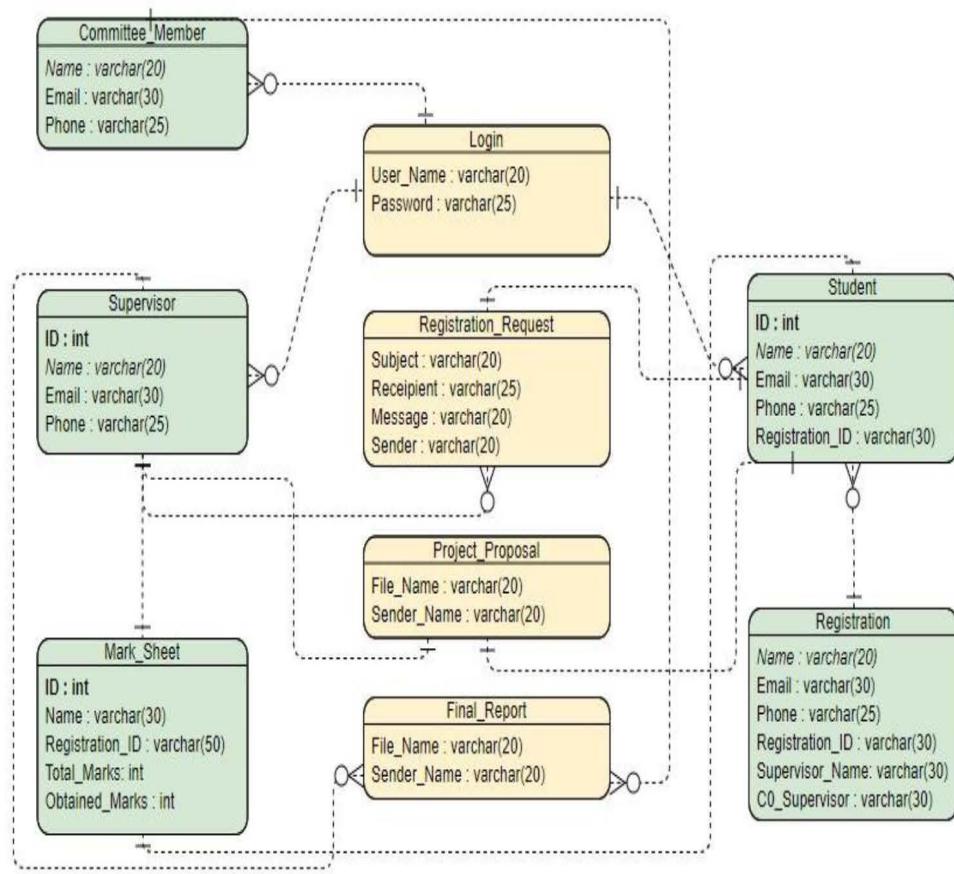


Figure 5.17: Database Diagram

In this database diagram it is shown the type of each element.

6. IMPLEMENTATION DETAILS

6.1 Development Setup

HTML-5/ CSS and java script: HTML and CSS are the two basic technologies for creating websites. HTML provides page structure, CSS (visual and audio) layout for different devices. And Java Script has the ability to update and modify both HTML and CSS. Java script has the ability of handling, calculating, and validating data.

- **Bootstrap** is a free and open-source Front-end web development for mobile devices was the focus of the CSS framework.
- **PHP** is a quick and feature-rich open-source scripting language used to create applications for the web or internet/intranet applications.

MySQL is built on a relational database management system (RDBMS), MySQL is a potent open-source database server that can manage several concurrent database connections.

6.2 Deployment setup

Our upcoming project is going on and we are completing the project soon.

6.3 Constraints

The required system constraints are given below:

- The internet connection must be connected to interact with the website.

6.3.1 Assumptions

Below are the following assumptions:

- The end user must have sufficient and appropriate connectivity to interact with our Services.
- We need a database connection to see the data on the web.

6.3.2 System constraints

The required system constraints are given below:

- An internet connection must be connected to interact with the website.

6.3.3 Restrictions

Below are the following restrictions:

- User cannot add duplicate ID.
- User account authentication must be strong.

6.3.4 Limitations

Below are the following Limitations:

- Only the administrator can manage the entire site.
- User can only read and display information.

7. TESTING

7.1 Extended Test Cases

Table 7.1: Perform Registration

TC-ID: 01		TC- Designed by: Abdul Rehman									
TC-Module Name: Perform registration		TC- Designed Date: 05/18/2023									
TC- Priority: High		TC- Executed by: Abdul Rehman									
TC- Title/Name: To test the registration form		TC- Executed Date: 05/25/2023									
Description: Test that students can register by submitting details.											
Pre - Condition: The student must not be registered and have no record in the database before.											
Dependencies											
Steps	Test - Steps	Test - Data	Expected -Result	Actual -Results	Status -(Pass/Fail)	Note					
1	The student opens the website.				Pass						
2	The student click on “Registration” button.				Pass						
3	The student enters his/her group detail.	The student enter the group members name and student_IDs			Pass						
4	The student click on “Register” button		The registration process is completed after the student clicks the register button.	The student performs registration successfully.	Pass						
Post Condition: The student registered successfully and has a record in the database.											

Description: Test that students can register by submitting details.

Table 7.2: Make Login

TC- ID: 02		TC- Designed by: Abdul Rehman									
TC- Module Name: Make Login		TC- Designed Date: 05/18/2023									
TC- Priority: High		TC- Executed by: Abdul Rehman									
TC- Title/Name: To test the log in process		TC- Executed Date: 05/25/2023									
Description: Test that students and supervisors can log in themselves.											
Pre - Condition: The students and supervisors must be registered.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes					
1	The student opens the website.				Pass						
2	The student click on "Log in" button.				Pass						
3	The student enters his/her id and password and click on log in button.	The student enters his/her id and password.			Pass						
4	The system verifies the id and password and leads the student to his/her profile.		The system will verify id and password and lead student to his/her profile.	The student successfully log in into the system.	Pass						
Post Condition: The students and supervisors log in successfully into the system to access certain functionalities.											

Description: Test that students and supervisors can log in themselves.

Table 7.3: Log out

TC- ID: 03	Test Designed by: Abdul Rehman											
Test Module Name: Log out	TC- Designed Date: 05/18/2023											
TC- Priority: High	TC- Executed by: Abdul Rehman											
TC- Title/Name: To test the log out	TC- Executed Date: 05/25/2023											
Description: Test that the students and instructors will logout from the system.												
Pre-Condition: They must be login into the system.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes						
1	The student/instructor clicks on “Logout” button.		The student/instructor will logout from the system.	The student/instructor logout from the system.	Pass							
Post Condition: The student logout successfully from the system.												

Description: Test that the students and instructors will logout from the system.

Table 7.4: View Portal

TC- ID: 04		TC- Designed by: Abdul Rehman									
TC- Module Name: View Portal		TC- Designed Date: 05/18/2023									
TC- Priority: Low		TC- Executed by: Abdul Rehman									
TC- Title/Name: To test the portal view		TC- Executed Date: 05/25/2023									
Description: Test that the students can log in themselves.											
Pre-Condition: The student must have an account before.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes					
1	The student opens the website.				Pass						
2	The student click on “Log in” button.				Pass						
3	The student enters his/her id and password and click on log in button.	The student enters his/her id and password.			Pass						
4	The system verifies the id and password and leads the student to his/her profile.		The system will verify id and password and lead student to his/her profile.	The student views the portal successfully.	Pass						
Post Condition: The student views the portal.											

Description: Test that the students can log in themselves.

Table 7.5: Title of the Project

TC- ID: 05	TC- Designed by: Abdul Rehman											
TC- Module Name: Title of the project	TC- Designed Date: 05/18/2023											
TC- Priority: Low	TC- Executed by: Abdul Rehman											
TC- Title/Name: To test the title of the project	TC- Executed Date: 05/25/2023											
Description: Test that the student can see his/her project title.												
Pre-Condition: The student must have project title.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes						
1	The student click on “Title of the project”.		The system will show the title of and ID of the project.	The student views the title of the project successfully.	Pass							
Post Condition: The student views the title of the project successfully.												

Description: Test that the student can see his/her project title.

Table 7.6: Team Leader Details

TC- ID: 06	TC- Designed by: Abdul Rehman											
TC- Module Name: Team Leader Details	TC- Designed Date: 05/18/2023											
TC- Priority: Low	TC- Executed by: Abdul Rehman											
TC- Title/Name: To test the team leader details	TC- Executed Date: 05/25/2023											
Description: Test that the student can see his/her profile.												
Pre-Condition: The student must have an account before.												
Dependencies												

Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The student click on “team lead student details”.		The system will show the title of the project and id of the project.	The student views the details of the team lead student successfully.	Pass	
Post Condition: The student views the details of the team lead student successfully.						

Table 7.7: Team Members Detail

TC- ID: 07	TC- Designed by: Abdul Rehman					
TC- Module Name: Team Members detail	TC- Designed Date: 05/18/2023					
TC- Priority: Low	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the team members detail	TC- Executed Date: 05/25/2023					
Description: Test that the student can see their group members detail.						
Pre-Condition: The student must have team members.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The student click on “Team members’ details”.		The system will show team members’ details.	The student views the details of the team members successfully.	Pass	
Post Condition: The student views the details of the team members successfully.						

Description: Test that the student can see their group members detail.

Table 7.8: Project Evaluation Criteria

TC- ID: 08	TC- Designed by: Abdul Rehman
TC- Module Name: Project Evaluation Criteria	TC- Designed Date: 05/18/2023
TC- Priority: High	TC- Executed by: Abdul Rehman
TC- Title/Name: To test the evaluation criteria	TC- Executed Date: 05/25/2023

Description: Test that the student can see project evaluation criteria specify by the supervisor.						
Pre-Condition: The student must have an instructor.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The student click on "Project evaluation criteria".		The system will show project evaluation criteria.	The student views the project evaluation criteria successfully.	Pass	
Post Condition: The student views the project evaluation criteria successfully.						

Description: Test that the student can see project evaluation criteria specify by the supervisor.

Table 7.9: Instructor Evaluation Result

TC- ID: 09	TC- Designed by: Abdul Rehman					
TC- Module Name: Instructor Evaluation Result	TC- Designed Date: 05/18/2023					
TC- Priority: High	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the instructor evaluation result	TC- Executed Date: 05/25/2023					
Description: Test that the student can see instructor evaluation result.						
Pre-Condition: The actor must submit his/her project idea.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The student click on "instructor evaluation result".		The system will show instructor evaluation result.	The student views the instructor evaluation result successfully.	Pass	
Post Condition: The student views the instructor evaluation result successfully.						

Description: Test that the student can see instructor evaluation result.

Table 7.10: Project Evaluator Evaluation Result

TC- ID: 10	TC- Designed by: Abdul Rehman					
TC- Module Name: Project Evaluator Evaluation Result	TC- Designed Date: 05/18/2023					
TC- Priority: High	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the evaluator evaluation result	TC- Executed Date: 05/25/2023					
Description: Test that the student can see Project evaluator evaluation result.						
Pre-Condition: The instructor must approve his/her project idea.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The student click on “evaluator evaluation result”.		The system will show Project evaluator evaluation result.	The student views the evaluator evaluation result successfully.	Pass	
Post Condition: The student views the evaluator evaluation result successfully.						

Description: Test that the student can see Project evaluator evaluation result.

Table 7.11: Set Registration Time

TC- ID: 11	TC- Designed by: Abdul Rehman					
TC- Module Name: Set Registration Time	TC- Designed Date: 05/18/2023					
TC- Priority: High	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the registration time	TC- Executed Date: 05/25/2023					
Description: Test that the instructor can set registration time of project.						
Pre-Condition: The registration time must not be set before.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The instructor set date from calendar and press “Update” button.		The system will update the registration time.	The students view the updated registration time successfully.	Pass	

Post Condition: The students view the updated registration time successfully.

Description: Test that the instructor can set registration time of project.

Table 7.12: Dashboard

TC- ID: 12	TC- Designed by: Abdul Rehman					
TC- Module Name: Dashboard	TC- Designed Date: 05/18/2023					
TC- Priority: High	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the dashboard	TC- Executed Date: 05/25/2023					
Description: Test that the supervisor can view totals registered groups, total projects and evaluated groups.						
Pre-Condition: The supervisor must have registered groups and evaluated groups.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The supervisor clicks on "Dashboard".		The system will show registered groups, total projects and evaluated groups.	The system show registered groups, total projects and evaluated groups.	Pass	
Post Condition: The system show registered groups, total projects and evaluated groups.						

Description: Test that the supervisor can view totals registered groups, total projects and evaluated groups.

Table 7.13: Check User

TC- ID: 13	TC- Designed by: Abdul Rehman					
TC- Module Name: Check User	TC- Designed Date: 05/18/2023					
TC- Priority: Low	TC- Executed by: Abdul Rehman					
TC- Title/Name: To test the users	TC- Executed Date: 05/25/2023					
Description: Test that the instructor can view complete detail of registered groups.						
Pre-Condition: The instructor must have registered groups.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes

1	The supervisor click on "Check user".		The system will show complete detail registered groups.	The system shows complete detail registered groups.	Pass	
Post Condition: The instructor viewed details successfully.						

Description: Test that the instructor can view complete detail of registered groups.

Table 7.14: Project Evaluation Criterion

TC- ID: 14	TC- Designed by: Abdul Rehman
TC- Module Name: Project Evaluation Criterion	TC- Designed Date: 05/18/2023
TC- Priority: High	TC- Executed by: Abdul Rehman
TC- Title/Name: To test the project evaluation criterion	TC- Executed Date: 05/25/2023
Description: Test that the instructor will set project criteria for registration.	
Pre-Condition: The instructor must have an account.	
Dependencies	

Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The supervisor clicks on “project evaluation criterion”.				Pass	
2	The supervisor enters rule and regulations for project and click on “Update” button.	Rule and regulation s for project.	The system will update the project evaluatio n criteria.	The system updates the project evaluation criteria.	Pass	
Post Condition: The project criteria successfully updated.						

Description: Test that the instructor will set project criteria for registration.

Table 7.15: Evaluate Groups

TC- ID: 15	TC- Designed by: Abdul Rehman
TC- Module Name: Evaluate Groups	TC- Designed Date: 05/18/2023
TC- Priority: High	TC- Executed by: Abdul Rehman
TC- Title/Name: To test the evaluate groups	TC- Executed Date: 05/25/2023
Description: This use case describe that the related actor will evaluate the requested groups.	

Pre-Condition: The related groups must not be registered before.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The instructor will clicks on “evaluate groups”.				Pass	
2	The instructor enters marks (0-5) for project evaluation and click on “Update” button.	Enter 0 to 5 marks.	The system will update the marks.	The system updates the marks.	Pass	
Post Condition: The groups evaluated successfully.						

Description: This use case describe that the related actor will evaluate the requested groups.

Table 7.16: Project External Evaluator

TC- ID: 16	TC- Designed by: Abdul Rehman
TC- Module Name: Project External Evaluator	TC- Designed Date: 05/18/2023
TC- Priority: Low	TC- Executed by: Abdul Rehman
TC- Title/Name: To test the Project External Evaluator	TC- Executed Date: 05/25/2023

Description: Test that the instructor can invite external evaluator to evaluate students' project.						
Pre-Condition: The related evaluator must not be invited before.						
Dependencies						
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes
1	The instructor clicks on "Project external evaluator".				Pass	
2	The instructor enters external evaluator details and click "send invite" button.	Name, Email address etc.	The system will send an email to related evaluator.	The system sends an email to related evaluator.	Pass	
Post Condition: The requested email send successfully.						

Description: Test that the instructor can invite external evaluator to evaluate students' project.

Table 7.17: Result Updating

TC- ID: 17	TC- Designed by: Abdul Rehman
TC- Module Name: Result Updating	TC- Designed Date: 05/18/2023

TC- Priority: High		TC- Executed by: Abdul Rehman									
TC- Title/Name: To test the result updates		TC- Executed Date: 05/25/2023									
Description: Test that the instructor will enter project marks.											
Pre-Condition: The related project must not be updated.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes					
1	The instructor clicks on "Result updating".				Pass						
2	The instructor click specific id and enter marks (0-5) for project evaluation and click on "Update" button.	Enter 0 to 5 marks.	The system will update the marks.	The system updates the marks.	Pass						
Post Condition: The related project result updated successfully.											

Description: Test that the instructor will enter project marks.

Table 7.18: Project Result

TC- ID: 18	TC- Designed by: Abdul Rehman											
TC- Module Name: Project Result	TC- Designed Date: 05/18/2023											
TC- Priority: High	TC- Executed by: Abdul Rehman											
TC- Title/Name: To test the project result	TC- Executed Date: 05/25/2023											
Description: Test that the related actor can see the whole result.												
Pre-Condition: The related project must be updated.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes						
1	The instructor clicks on "Project result".		The system will show project result.	The system shows the project result.	Pass							
Post Condition: The project result views successfully.												

Description: Test that the related actor can see the whole result.

Table 7.19: Final Result

TC- ID: 19	TC- Designed by: Abdul Rehman											
TC- Module Name: Final Result	TC- Designed Date: 05/18/2023											
TC- Priority: High	TC- Executed by: Abdul Rehman											
TC- Title/Name: To test the final result	TC- Executed Date: 05/25/2023											
Description: Test that the instructor can see the final absolute result.												
Pre-Condition: The related project must be updated result.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Results	Status (Pass/Fail)	Notes						
1	The instructor clicks on "Final Result".		The system will show Final Result.	The system shows the Final Result.	Pass							
Post Condition: The instructor views the final result successfully.												

Description: Test that the instructor can see the final absolute result.

7.2 Decision Table

In decision table we are going to provide our code snippet the screenshot of our code and a little description about it.

7.2.1 Code snippet

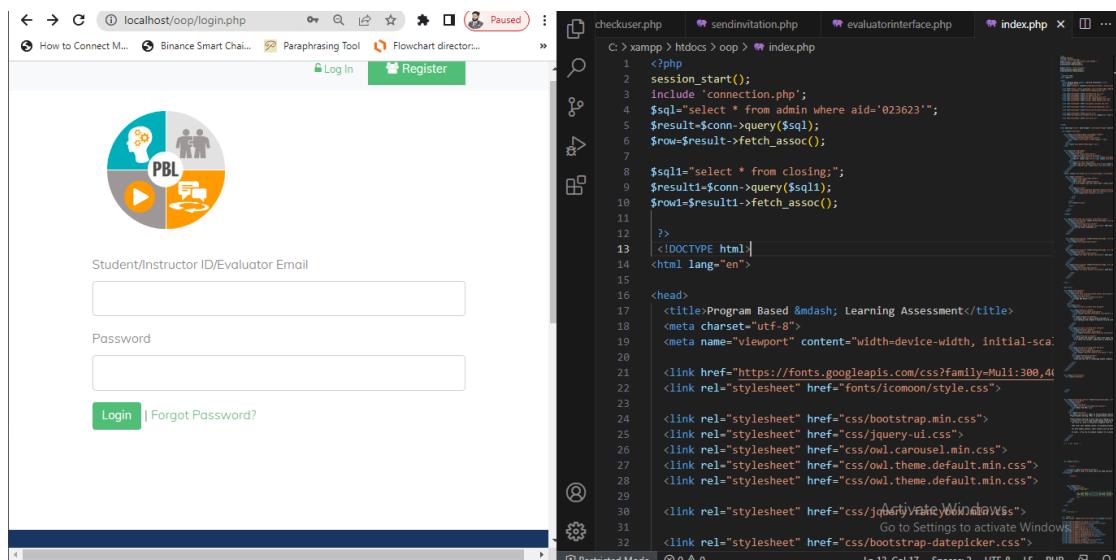


Figure 7.1: Login Page Snapshot

Description: This code snap describes the login page of our website.

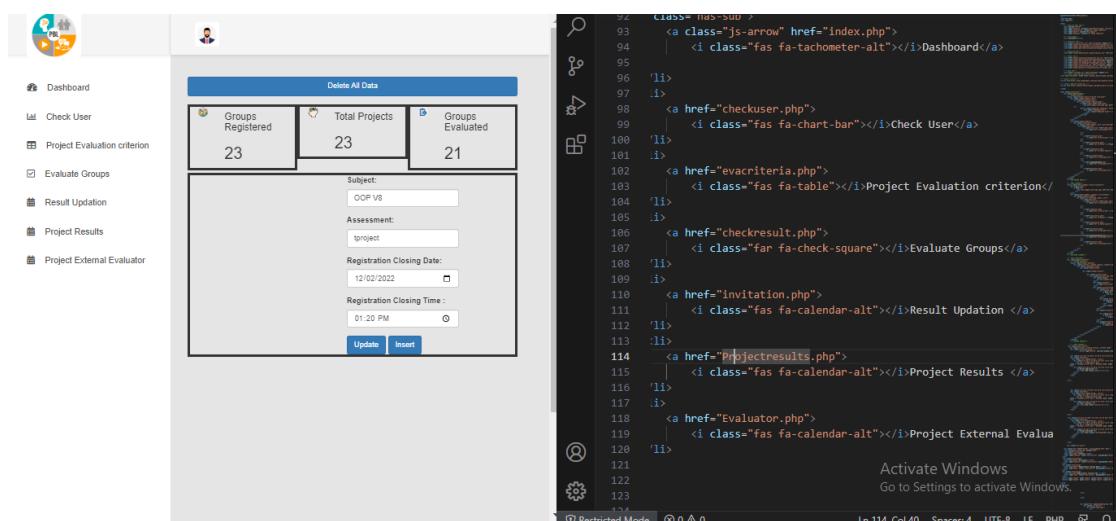


Figure 7.2: Dashboard of Advisor Snapshot

Description: Figure 7.2 shows the dashboard of advisor's side.

The screenshot shows a web application interface for a student portal. At the top, there's a navigation bar with a user profile icon and the text "DANIYAL MAZHAR Logout". Below the navigation bar, a green banner displays a welcome message: "Hello Web Developer! Welcome to the Project Based Learning System Where you will be evaluated. Important Notice! DeadLine for Project Registration: 2022-12-02". The main content area has tabs for "Title Of The Project", "Team Lead/Solo Student Details", "Team Members Details", "Project Evaluation Criteria", "Project Evaluation Result", and "Project Evaluator Evaluation Result". Under "Title Of The Project", it says "SECURITY MANAGEMENT" and "FermQnfrXWMS". Under "Team Lead/Solo Student Details", it lists "F2019266101", "DANIYAL MAZHAR", "F2019266101@UMT.EDU.PK", and "123456789". The "Team Members Details" section shows "Details of the registered team members" with names: "F2019266127", "ABDULREHMAN", "F2019266148", and "M UMAIR NASIR". A note at the bottom states "Your Project Will Be Evaluated Against These Criteria!". On the right side of the screenshot, a portion of the browser's code editor is visible, showing CSS and JavaScript code.

Figure 7.3: Code Snapshot of Student Portal

Description: Figure 7.3 of code show the student side of portal.

The screenshot shows the evaluator interface. At the top, there's a navigation bar with a user profile icon and the text "023624". Below the navigation bar, a section titled "Evaluation Criteria Marks" contains a text input field with the placeholder "0-5 Marks for each Criteria". Another section titled "Evaluation Form for Register Students" includes a note: "Columns Shows the Evaluation Criteria & for each you are given 0-5 marks. Rows shows the Students Details and the Project Details". The main content area displays a table of student projects. The columns include "Project ID", "Student Name", and "Project Title". The rows list various student IDs and their corresponding names and project titles. On the right side of the screenshot, a portion of the browser's code editor is visible, showing PHP and HTML code.

Figure 7.4: Evaluator Side Snapshot

Description: Figure 7.4 shows the evaluator side of the portal.

7.3 Traceability Matrix

Here are we going to make our traceability matrix of RID vs UCID, RID vs PID, RID vs TID & UCID vs TID.

7.3.1 RID vs UCID (requirements vs use cases)

UCID/ RID	R-1	R-2	R-3	R-4	R-5	R-6	R-7	R-8	R-9	R-10	R-11	R-12	R-13
UC-1	✓												
UC-2		✓											
UC-3			✓										
UC-4				✓									
UC-5													
UC-6													
UC-7													
UC-8													
UC-9					✓								
UC-10					✓								
UC-11						✓							
UC-12									✓				
UC-13													
UC-14				✓									
UC-15					✓								
UC-16					✓								
UC-17					✓								
UC-18					✓								
UC-19					✓								

7.3.2 Prototypes (RID vs PID)

Description: Here we are going to make our traceability matrix of RID vs PID.

PID/RI D	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21
P-1	✓																				
P-2	✓	✓																			
P-3			✓	✓			✓	✓	✓		✓		✓	✓	✓						
P-4		✓																✓			
P-5		✓		✓																	
P-6		✓										✓									
P-7		✓	✓						✓			✓									
P-8		✓							✓	✓											

7.3.3 Test Cases (RID vs TID)

Description: Here are we going to make our traceability matrix of RID vs TID.

TID/RID	R -	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13
T-1	✓													
T-2		✓												
T-3			✓											
T-4				✓										
T-5														
T-6														
T-7														
T-8					✓									
T-9														
T-10														
T-11							✓							
T-12														
T-13									✓					
T-14					✓									
T-15						✓								
T-16						✓								
T-17						✓								
T-18						✓								
T-19						✓								

7.3.4 Coverage (UCID vs TID)

Description: Here are we going to make our traceability matrix of TID vs UCID.

UCID/ TID	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9	T 10	T 11	T 12	T 13	T 14	T 15	T 16	T 17	T 18	T 19
UC-1	✓																		
UC-2		✓																	
UC-3			✓																
UC-4				✓															
UC-5					✓														
UC-6						✓													
UC-7							✓												
UC-8								✓											
UC-9									✓										
UC-10										✓									
UC-11											✓								
UC-12												✓							
UC-13													✓						
UC-14														✓					
UC-15														✓					
UC-16															✓				
UC-17																✓			
UC-18																	✓		
UC-19																		✓	

8. RESULTS

1.1 % Completion

Project is 90% completed.

1.2 % Accuracy

Project is 85% accurately working.

1.3 % Correctness

Project has 85% correctness.

2. CONCLUSION

We chose to work on a platform that was unfamiliar to us, therefore the project document was rife with difficulties. Our project was heavily web-based. Although we were familiar with the many modules in the report since we had studied them for the degree, applying all of those principles to our assignment was always difficult. One of the main issues we had as we created a platform for the users interested in the Final Year Project registration process was deciding the features being supplied in our project. The greatest qualities that will make people's lives easier have to be obtained. so that everyone may focus on their task effectively and easily. Another problem was creating the platform because the audience for our product is highly diverse, so we needed to create a design that would work for everyone. The collecting of stakeholder requirements was the second largest hurdle we encountered. It was never easy to arrange meetings with the various stakeholders to learn about their requirements and to use requirement-gathering methodologies.

3. FUTURE WORK

Our main goal was to design a website that would allow students, teachers and committee members to easily navigate the FYP registration process. In the future, we plan to introduce FYP2 assessment and grade updates.

Final Year Project Report

Tron games



Project Advisor: Mr. Owais Khan

Co-advisor: Mr. Rehan Raza

Submitted By: Maaz Saleem (F2017266085)

Session: 2017-2023

Bachelors in Computer Sciences BS-CS

Department of Computer Science

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

Special thanks to:

- **My advisor: Mr. Owais Khan**
- **Respected Teachers: Mr. Noaman Saleem**
- **My parents**
- **The University**

Trongames

Final Approval

- Head of Department
Department of Computer Sciences
School of Systems & Technology
UMT Lahore

Tu

- Director (Final Year Projects-CS)
Department of Computer Science.
School of Systems & Technology
UMT Lahore

Muhammad Ali Shah

16/8/2023

- Supervisor _____
School of Systems & Technology
UMT Lahore

Waqas

- Co-Supervisor _____

Acknowledgment

I wish to register my profound gratitude to Almighty Allah for the guidance and grace in my life. I would also like to extend special regards to my Parents and siblings. May Allah continue showering them with Blessings, Ameen.

I am also grateful to the CS Faculty and staff of university of management and technology for making my bachelors in computer sciences educative and worthwhile.

I appreciate the amazing lecturers in the department for their great and seasoned lectures and guidance. May Allah bless them all.

Project Title: Tron games

Objective: Website for gamers to buy and exchange games.

Undertaken by : Maaz Saleem

Supervised by: M.Owais Khan

Starting Date: Nov-2022

Completion Date: May-2023

Tools Used: Microsoft visual studio code, Cloud database (MongoDB and Amazon S3)

Documentation

Plagairism Report

University of Management and Technology, Lahore

Similarity Report

Turnitin Originality Report
Tron Games by Maaz Saleem

From Reference Desk (LRC info desk)

- Processed on 10-Aug-2023 10:42 PKT
- ID: 2143836567
- Word Count: 4235

Similarity Index

5%

Similarity by Source

Internet Sources:

2%

Publications:

2%

Student Papers:

4%

Sources:

1. 1% match (student papers from 07-May-2012)
Submitted to Higher Education Commission Pakistan on 2012-05-07
2. 1% match (student papers from 23-Jul-2018)
Submitted to Higher Education Commission Pakistan on 2018-07-23



Checked by



For- Verified by CLO

Note:

- Sometimes the overall similarity index may be a smaller than the repository percentages combined. This would be due to overlapping text within the repositories.
- It is a system generated report.

Abstract

The report talks about the project we are working. The project comes from the renowned gaming industry which has been always overlooked in our country. Mostly considered as hindrance in mental growth. But it results in problem solving skills being polished. Because of being overlooked the industry never flourished. The gap in the market presents an opportunity for creating business for dealing in games. As piracy gripped the PC sector, it wasn't able to grip the console sector. Which led us to consider the gap for need of games in Local market as a project for our FYP. As the market is emerging these days and people are moving more and more towards the gaming sector and wanting to buy authentic and issue free games. Sadly, our region isn't given the benefit of accessing this content through our local network. So, this project mostly aims towards eradicating that certain gap which an average user has to face while buying games.

Learning from all the technical electives has led us to implement our knowledge on this.

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
Draft	Maaz Saleem	Initial draft created	15-nov-2022
Initial version	Maaz Saleem	Initial version created with the initial details	20-nov-2022
Mid-semester report v1	Maaz Saleem and M. Sohaib Amir	Mid-semester report created	25-nov-2022
Revision	Maaz Saleem	Revisions made in the report and mock screen snippets added (data flow and requirements to be expanded)	02-dec-2022
Revision-ii	M. Sohaib	Mock screen added with overall changes	15-dec 2022
Revision-iii	Maaz Saleem	Diagrams added	20-dec 2022
Revision-iv	Maaz Saleem	New mock screens	27-dec 2022
Revision-v	Maaz Saleem	Requirements updated	7-jan 2023
Revision-vi	Maaz Saleem	Algorithm screenshot added	15-jan 2023
Revision-vii	Maaz Saleem	Methodology and new age added, Diagrams for system architecture and class diagram added. Extended use cases added	20-feb 2023
Final version	Maaz Saleem	Relevant details updated and finalized	15-June-2023

CONTENTS

CONTENTS.....	I
ACRONYMS AND THEIR DEFINITIONS	III
FIGURES.....	IV
TABLES	V
1. INTRODUCTION	1
1.1 INITIATIVES	1
1.2 PROJECT OUTLINE	1
1.3 PROBLEM DECLARATION	2
1.3.1 <i>Context</i>	2
1.3.2 <i>Problem Statement</i>	2
1.3.3 <i>Digital Age</i>	2
1.4 OBJECTIVES	2
1.5 METHODOLOGY	2
2. DOMAIN EVALUATION	3
2.1 CONSUMER	3
2.2 STAKEHOLDERS.....	3
2.3 GROUPS HAVING ECONOMIC ABD SOCIAL IMPACT.....	3
2.4 DEPENDENCIES	3
2.5 DOCUMENTS FOR REFERNCE.....	3
2.5.1 <i>Related Projects</i>	3
2.5.2 <i>Feature Comparison</i>	4
3. REQUIREMENTS EXPLORATION.....	5
3.1 REQUIREMENT GATHERING.....	5
3.2 STAKEHOLDER/ACTOR LIST	5
3.3 LIST OF USE CASES.....	6
3.4 USE CASE DIAGRAM FOR SYSTEM	7
3.5 USE CASES (EXTENDED).....	7
3.6 USER INTERFACES (MOCK SCREENS).....	20
4. SYSTEM DESIGN	23
4.1 SYSTEM ARCHITECTURE DIAGRAM	23
4.2 CLASS DIAGRAM.....	23
CLASS DIAGRAM	23
4.2.1 <i>Diagram</i>	23
4.3 SEQUENCE DIAGRAMS.....	24
4.4 COLLABORATION DIAGRAMS.....	24
4.5 ERD (SAME AS CLASS DIAGRAM)	25
4.5.1 <i>ERD(figure)</i>	25
5. IMPLEMENTATION DETAILS	26
5.1 DEVELOPMENT SETUP	26
5.2 DEPLOYMENT SETUP.....	26
5.3 ALGORITHMS	26
5.4 CONSTRAINTS	28
5.4.1 <i>Assumptions</i>	28

5.4.2	<i>System constraints</i>	28
5.4.3	<i>Restrictions</i>	28
5.4.4	<i>Limitations</i>	28
6.	TESTING.....	29
6.1	EXTENDED TEST CASES.....	29
7.	RESULTS/OUTPUT/STATISTICS(TRACEABILITY MATRIX)	37
7.1	REQUIREMENTS VS USE CASES.....	37
7.2	TEST CASES (RID vs TID).....	37
7.3	COVERAGE (UCID vs TID)	38
7.4	RESULTS	38
7.4.1	%completion	38
7.4.2	%accuracy	38
7.4.3	%correctness	38
8.	FUTURE WORK	39
9.	BIBLIOGRAPHY	40
9.1	ARTICLES	40
9.2	RESEARCH PAPERS	40
9.3	OTHER REFERENCES	40

Acronyms and their definitions

Acronym	Definition
UMT	University of Management and Technology
POS	Point of Sale
CDkey	Compact disk digital redeemable keys

Table 1: table of acronyms with their definitions

Figures

Table 1: table of acronyms and definitions	iii
Table 2: list of stakeholders.....	3
Figure 1 Use case diagram-1	7
Prototype1: (P1) Main screen	20
Figure 2: Admin panel.....	21
Prototype2: (P2) Product 1	21
Prototype3: Admin	22
Figure 3 Sequence diagram	24
Figure 4 Collaboration.....	24
Figure 5 Collaboration.....	25
Figure 7 Algorithm-ii	26
Figure 6: Main admin page uses stats to show sale and purchase. This is the algorithm which works behind it.....	27
Figure 8 Algorithm-iii	27

Tables

Table 1: table of acronyms with their definitions.....	iii
Table 2: list of stakeholders.....	3
Table 3 UC1	8
Table 4 UC2	9
Table 5 UC3	10
Table 6 UC4	11
Table 7 UC5	12
Table 8 UC6	13
Table 9 UC7	14
Table 10 UC8	15
Table 11 UC9	16
Table 12 UC10	17
Table 13 UC11	18
Table 14 UC12	19
Table 15 UC13	20
Table 16 TC1.....	29
Table 17 Tc2.....	30
Table 18 Tc03.....	31
Table 19 Tc4.....	32
Table 20 Tc5.....	32
Table 21 Tc6.....	34
Table 22 TC7.....	35
Table 23 TC8.....	35
Table 24 TC09.....	36

1. INTRODUCTION

1.1 Initiatives

Real world

The motivation for this project comes from a real-world example. It concludes the example of the project member. As there are markets in only major cities. The user had to visit the market in the isolated corner of the city. It includes certain stores. To buy a specific game which was call of duty. The game's physical copy wasn't available. Only one store had it and the copy been repackaged one and in awful condition. Many hours were wasted for this. There was a gap in the market which could be seen. Users waste hours for a single game and still don't get to buy it. The stores only focus on selling console and buying them. The game market is untouched. So, the motivation came from this example because of the fuss and time wasted for a single product.

1.2 Project Outline

Overview Statement

The overview of the topic is that the project revolves around the gaming needs of console users in the regional market. It provides a platform to the users to buy and exchange games without going out visiting various stores. The project is based on web development. Mainly it introduces a platform where users can easily buy games. These games are available in physical and Digital forms. Users demand physical from and even if they can't get that they prefer the digital forms. There is a shortage of physical form of games in the Pakistani market. They are hard to find and even if the product can be found. Users get a hefty price tag with it. Many users cannot travel to other cities to buy a physical copy of the game. This platform provides those physical copies of games to the users at their doorstep. Another thing the platform introduces is the digital version of games. It is mainly in Cd keys. There aren't any cd/game keys available in the market. These cd/game keys can be bought and inserted into the game store of your console. It straightly downloads it and your account has the ownership for it. The platform provides Cd/game keys to the users. Also, subscription keys are provided in it. Many users play a game and want to exchange the physical copy of it with another game. Mostly stores don't exchange games. Rarely some do and charge extra. This platform enables the users to be able to exchange games from their doorstep with only service charges and cost accounted.

Customer

The customer is the console user as well as a pc user. It is targeted towards the users who want genuine product free from piracy

Goals

The goal here is to make a platform which becomes a famous game retail and exchange platform. Which saves time as well as provide genuine product in terms of DVDs of games and CD keys.

System functions

The system overall includes updating of database, maintain catalogue, user can place an order, user can place a request and login/signup.

System Attributes

Performance

The system performs well on the cloud database used.

Security

The security of the system is based on the cloud platform used.

Modifiability

The system is easy to modify because of noncomplex code of web and already organized database MongoDB

Usability

The usability of this system is kept simple.

1.3 Problem declaration

1.3.1 Context

The problem revolves around the category of gaming industry. The category is of console games. In Pakistani market the console selling and buying is booming. The focus on console games is not that much. It is an untouched market. So, the problem statement focuses on the console games. There is no concept of cd/game key in this region. These can be bought online or from console online stores. But there is a major problem. The regional credit or debit cards aren't applicable.

1.3.2 Problem Statement

The problem statement is that there is a gap in game market. This gap is of console games. Users are unable to get the desired game in the market. This gap costs users a lot of their time and money. Mainly there are stores in only major cities. In order to buy a certain game, the user has to travel to another corner of the city. There is no assuredly that the game will be available. If the user lives in another city, then the user has to bear travel expense and time. This doubles the cost to get a single product. So, this gap is created because of stores focusing only on console selling and purchasing. The problem is of no knowledge of cd/game key.

1.3.3 Digital Age

As the digital age is increasing. Most of the new consoles are jailbreak free and are digital. As far as the piracy is concerned

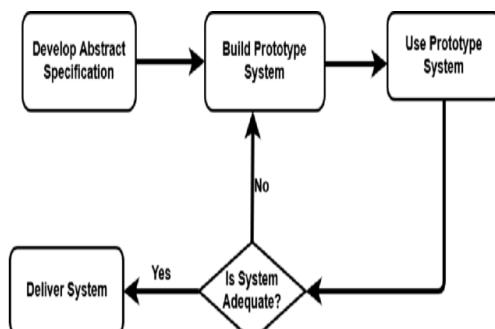
1.4 Objectives

The objectives of this project are

- Providing a one stop platform to buy games.
- Introducing the concept of CD keys in the market which is rare.
- Saving time of users to buy more products.
- Providing reasonable exchange rates for the games.
- Providing genuine tested product in form of DVD's

1.5 Methodology

The methodology used in this project is **Prototyping**. This methodology includes revolutionary prototyping. In this project I have used it. A prototype is created then it will be turned to final product. The final product may vary in design only. The design can be changed looking at the current leading designs or creating one which is easy for user to understand.



2. DOMAIN EVALUATION

2.1 Consumer

No customer contracted

2.2 Stakeholders

Stakeholder	Role in System
Web developer	He/She is responsible for the development of the website
Android developer	He/She is responsible for the development of the Android app
Project manager	Required to coordinate the project
Advisor	Advisor overlooks the project
Supplier	For maintaining the catalogue
User	The end user which interacts with the system

Table 2: list of stakeholders

2.3 Groups having economic abd social impact

It only includes a certain group of people who will be impacted by the system

- Gamers/Console users
People who have consoles or want genuine games to play on their devices can easily save their time through this platform and are thus impacted by it.

2.4 Dependencies

The system depends upon for its completion:

- Mongo DB
It is the database which is a fully automated system which only needs a map to store data
- Payment gateways
Payment gateways for debit/credit payments such as Pay mob.
- Amazon S3
It is used to host images.

2.5 Documents for Reference

2.5.1 Related Projects

The related projects are as follows:

- <https://www.gamestop.com/>
The platform provides trade-ins and various kinds of games for various consoles.
- <https://www.bestbuy.com/>

- Best buy provides various kinds of electronics but also deals in games.
- <https://www.cdkeys.com/>
This platform provides CD keys for various console.
 - Microsoft store
It provides digital games which is an alternative of cdkeys.

2.5.2 Feature Comparison

Sr No.	Comparison Feature	GameStop	Best buy	Remarks
1	Trade in feature	GameStop provides cash for trade in	BestBuy only provides refunds	Evolving the trade in feature into exchanging feature which only costs user small exchange fee and only works as game-to-game exchange
2	Digital games	Cdkey	Microsoft store/Sony/Nintendo	As stores provides digital game and other stores provide digital games in forms of CD keys, these aren't accessible in Pakistan. So, those digital games can be provided through CD keys here.

3. REQUIREMENTS EXPLORATION

3.1 Requirement gathering

The requirements include requirements from users. It also includes the functional and nonfunctional requirements which populate over time as the product nears its completion:

FR1. Users will be able to create an account

FR2. Users will be able to login

FR3. Users will be able to place an order of a certain product

FR4. Users will be able to request a certain product if it not available

FR5. Users will be able to put in a request for certain game

FR6. Users will be able to place order for console subscription codes

FR7. Users will be able to search product

FR8. Admin will be able to post product details

FR9. Admin will be able to modify the product details

FR10. Admin will be able to delete the product

FR11. Admin will be able to approve or disapprove product request

FR12. Users will be able to logout

Nonfunctional requirements

NFR1. The website can handle several requests

NFR2. The website will be expanded to console selling

NFR3. The website will have different sections for games and codes

NFR4. The website will be responsive

NFR5. The database which will be used is efficient

NFR6. The website will have easy navigation.

3.2 Stakeholder/Actor list

Actors	Category	Task
Developers	Primary	They develop the website/app to fulfill the project
Project manager	Primary	PM co-ordinates the developers and leads the project
Users	Primary	Users interact with the system to purchase product
Supplier	Primary	Supplier helps maintain the catalogue (can be any company)
Banks/institutions	Secondary(supporting)	Provide gateway for payments and cash flow

Devices	Secondary(supporting)	Devices on which the app/website runs and help user interact.
Database	Secondary(supporting)	Database which sends and receive queries for the data.
Clients and servers	Secondary(supporting)	Clients and servers which enable the devices and database to communicate.

3.3 List of use cases

The use cases are as follows:

- **Sign-up:** Users will be able to sign up
- **Log-in:** Users will be able to log in their credentials
- **Log-out:** Users will be able to log out their accounts
- **Populate cart:** Users will be able to populate their carts.
- **Preview item:** Users will be able to preview item and its trailer.
- **Buy item:** Users will be able to buy an item and put in purchase request
- **Search product:** Users will be able to search an item through search bar
- **Populate database:** Admin will be able to populate database
- **Delete item:** Admin can delete an item
- **Delete user:** Admin can delete a user
- **Check purchase:** Admin can Check a purchase request by the user.

3.4 Use case diagram for System

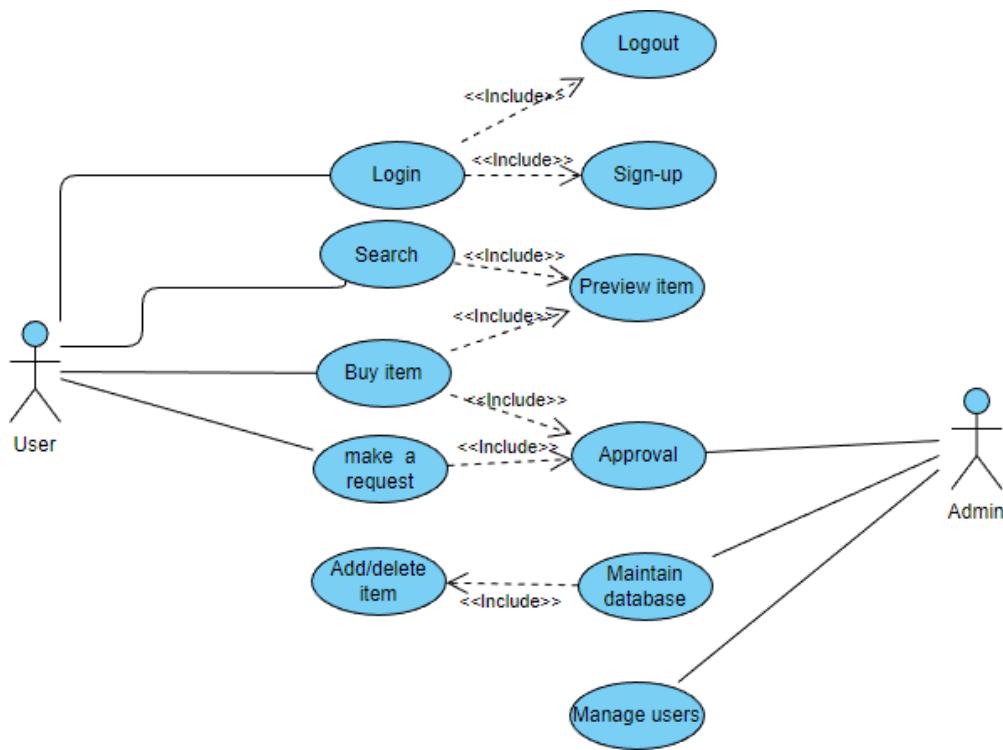


Figure 1 Use case diagram-1

3.5 Use cases (Extended)

Use Case ID:	UC-1		
Use Case Name:	Search product		
Created By:	Maaz	Last Updated By:	Maaz
Date Created:	15-12-22	Last Revision Date:	15-12-22
Actors:	User		
Description:	User searches for a certain product with a keyword in the search bar		
Trigger:	User doesn't want to view the list of product. Instead he/she initiates search		
Preconditions:	<ol style="list-style-type: none"> 1. User must have access to the website 2. Appropriate internet connection 		
Post conditions:	<ol style="list-style-type: none"> 1. User is able to view the desired product 2. User can put in a request if product is not listed 		
Normal Flow:	<ol style="list-style-type: none"> 1. Users inserts website address 2. User reaches website 3. User clicks on the search bar 4. User inserts keyword 5. User looks at the result 6. The product is listed 7. User views the product 		
Alternative Flows:	<p>In step 5 if the product is not listed</p> <ol style="list-style-type: none"> 1. Users accesses request section 2. User puts up a request for a certain product 		
Exceptions:	<ol style="list-style-type: none"> 1. Server doesn't respond 2. Response time becomes high 3. Keyword inserted can be conflicting 		
Includes:	Request use case		
Frequency of Use:	Execution is on demand of the user		
Special Requirements:	Nil		
Assumptions:	The Website is in English language only		
Notes and Issues:	<ol style="list-style-type: none"> 1. Nil 		

Table 3 UC1

Table 4 UC2

Use Case ID:	UC-2
Use Case Name:	Place an order
Created By:	Maaz
Date Created:	15-12-22
Actors:	User
Description:	User places an order of a product
Trigger:	User wants to acquire a new game
Preconditions:	<ul style="list-style-type: none"> 1. User must have access to the website 2. Appropriate internet connection 3. Delivery details(address, phone , zipcode)
Post conditions:	<ul style="list-style-type: none"> 1. User is able to view the desired product 2. User can put in a request if product is not listed 3. User can place order 4. User has paid in cash or has chosen cod
Normal Flow:	<ul style="list-style-type: none"> 1. Users inserts website address 2. User reaches website 3. User looks for the desired product 4. The product is listed 5. User views the product 6. User places an order 7. Confirmation pending
Alternative Flows:	<p>In step 5 if the product is not listed</p> <ul style="list-style-type: none"> 1. Users accesses request section 2. User puts up a request for a certain product <p>In step 7</p> <ul style="list-style-type: none"> 1. The order can be confirmed or rejected
Exceptions:	<ul style="list-style-type: none"> 1. Server doesn't respond 2. Response time becomes high 3. Keyword inserted can be conflicting 4. Payment can be failed
Includes:	Request use case
Frequency of Use:	Execution is on demand of the user
Special Requirements:	Nil
Assumptions:	The Website is in English language only
Notes and Issues:	<ul style="list-style-type: none"> 1. Nil

Table 5 UC3

Use Case ID:	UC-3		
Use Case Name:	Requesting a game		
Created By:	Maaz	Last Updated By:	Maaz
Date Created:	15-12-22	Last Revision Date:	15-12-22
Actors:	User		
Description:	User puts in request for a non-listed game		
Trigger:	User wants to acquire a new game		
Preconditions:	<ol style="list-style-type: none"> 1. User must have access to the website 2. Appropriate internet connection 3. Game details (release year.) 		
Post conditions:	<ol style="list-style-type: none"> 1. User is unable to see the game list 		
Normal Flow:	<ol style="list-style-type: none"> 1. Users inserts website address 2. User reaches website 3. User looks for the desired product 4. The product is not listed 5. User accesses request section 6. User place a request 		
Alternative Flows:	<p>In step 6 if the product is not commercially available</p> <ol style="list-style-type: none"> 1. The request may be confirmed or rejected 		
Exceptions:	<ol style="list-style-type: none"> 1. Server doesn't respond 2. Response time becomes high 3. Keyword inserted can be conflicting 4. Request can be failed register 		
Includes:			
Frequency of Use:	Execution is on demand of the user		
Special Requirements:	Nil		
Assumptions:	The Website is in English language only		
Notes and Issues:	<ol style="list-style-type: none"> 1. Nil 		

Table 6 UC4

Use Case ID:	UC-4		
Use Case Name:	Approval/Non-approval of order on cd-keys		
Created By:	Maaz	Last Updated By:	Maaz
Date Created:	15-12-22	Last Revision Date:	15-12-22
Actors:	Admin		
Description:	User puts in order for a cd-key and admin views the order for approval or non-approval		
Trigger:	User wants to purchase a cd-key		
Preconditions:	<ul style="list-style-type: none"> 1. User must have access to the website 2. Appropriate internet connection 3. Game details (release year) 		
Post conditions:	Nil		
Normal Flow:	<ol style="list-style-type: none"> 1. User places an order of cd-key 2. Admin receives the order 3. Admin approves or dis-approves of the order based on availability of cd-keys. 		
Alternative Flows:	<p>In step 3 if the order is accepted</p> <ol style="list-style-type: none"> 1. Admin approves the order 2. Order is dispatched electronically 		
Exceptions:	<ol style="list-style-type: none"> 1. Server doesn't respond 2. Response time becomes high 		
Includes:			
Frequency of Use:	Execution is on demand of the user		
Special Requirements:	Nil		
Assumptions:	The Website is in English language only		
Notes and Issues:	<ol style="list-style-type: none"> 1. Nil 		

Table 7 UC5

Use Case ID:	UC-5		
Use Case Name:	User signs up		
Created By:	Maaz	Last Updated By:	Maaz
Date Created:	15-12-22	Last Revision Date:	15-12-22
Actors:	User		
Description:	User wants to sign up on the website		
Trigger:	User wants to place an order or view website		
Preconditions:	<ol style="list-style-type: none">1. User must have access to the website2. Appropriate internet connection3. Valid email		
Post conditions:	User account is created		
Normal Flow:	<ol style="list-style-type: none">1. User accesses the website2. User visits sign up forum3. User enters information(password, email, contact, name)		
Alternative Flows:	No alternative flow		
Exceptions:	<ol style="list-style-type: none">1. Server doesn't respond2. Response time becomes high3. User already has an account4. User password is not secure		
Includes:			
Frequency of Use:	Execution is on demand of the user		
Special Requirements:	Nil		
Assumptions:	The Website is in English language only		
Notes and Issues:	<ol style="list-style-type: none">1. Nil		

Table 8 UC6

ID for UC:	Uc-6		
Use Case Heading:	User Login		
Creation by:	Maaz	Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actor:	User		
Information:	User logs in to website		
Trigger:	User wants to login		
Preconditions:	<ol style="list-style-type: none">1. User must have internet2. User must have access to website		
Post conditions:	<ul style="list-style-type: none">• User logged in		
Normal Flow:	<ol style="list-style-type: none">1. User opens website2. User selects account3. User presses login4. Authenticates through google5. User logged in		
Alternative Flows:			
Exceptions:	<ol style="list-style-type: none">1. High response time		
Includes:			
Frequency of Use:	<ol style="list-style-type: none">1. User initiated		
Special Requirements:	nil		
Assumptions:	<ol style="list-style-type: none">1. Website is in English		
Notes and Issues:			

Table 9 UC7

ID for UC:	Uc-7		
Heading for UC	User Logout		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	User		
Information:	User logs out from website		
Trigger:	User wants to logout		
Preconditions:	<ol style="list-style-type: none">1. User must have internet2. User must have access to website		
Post conditions:	<ul style="list-style-type: none">• User logged in		
Normal Flow:	<ol style="list-style-type: none">1. User opens website2. User selects account3. User presses logout4. User logged out		
Alternative Flows:			
Exceptions:	<ol style="list-style-type: none">1. High response time		
Includes:			
Frequency of Use:	<ol style="list-style-type: none">1. User initiated		
Special Requirements:	nil		
Assumptions:	<ol style="list-style-type: none">2. Website is in English		
Notes and Issues:			

Table 10 UC8

ID for UC:	Uc-8		
Heading for UC	User Populates cart		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	User		
Information:	User Populates cart		
Trigger:	User wants to add products to cart		
Preconditions:	<ol style="list-style-type: none">1. User must have internet2. User must have access to website		
Post conditions:	<ul style="list-style-type: none">• User has product in cart		
Normal Flow:	<ol style="list-style-type: none">1. User opens website2. User selects product3. User presses add to cart4. User has products in cart		
Alternative Flows:			
Exceptions:	<ol style="list-style-type: none">1. High response time		
Includes:			
Frequency of Use:	<ol style="list-style-type: none">1. User initiated		
Special Requirements:	nil		
Assumptions:	<ol style="list-style-type: none">1. Website is in English		
Notes and Issues:			

Table 11 UC9

ID for UC:	Uc-9		
Heading for UC	User Views item		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	User		
Information:	User view product		
Trigger:	User wants to view product		
Preconditions:	1. User must have internet 2. User must have access to website		
Post conditions:	<ul style="list-style-type: none"> • User is on website 		
Normal Flow:	1. User opens website 2. User selects product 3. User presses view 4. Product is open in new window		
Alternative Flows:			
Exceptions:	1. High response time		
Includes:			
Frequency of Use:	1. User initiated		
Special Requirements:	nil		
Assumptions:	1. Website is in English		
Notes and Issues:			

Table 12 UC10

ID for UC:	Uc-10		
Heading for UC	User buys product		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	User		
Information:	User buys product		
Trigger:	User wants to buy product		
Preconditions:	1. User must have internet 2. User must have access to website		
Post conditions:	<ul style="list-style-type: none"> • User is on website 		
Normal Flow:	1. User opens website 2. User selects product 3. User adds to cart 4. User check-out 5. User provides card/cod details		
Alternative Flows:			
Exceptions:	1. High response time 2. Payment failure by merchant		
Includes:	Uc-8 , Uc-9		
Frequency of Use:	1. User initiated		
Special Requirements:	nil		
Assumptions:	1. Website is in English		

Notes and Issues:

Table 13 UC11

ID for UC:	Uc-11		
Heading for UC	Admin adds item		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	Admin		
Information:	Admin adds product		
Trigger:	Admin wants to update inventory		
Preconditions:	1. Admin must have internet 2. admin must have authorized access to admin panel		
Post conditions:	<ul style="list-style-type: none"> • Product is on website 		
Normal Flow:	1. Admin opens panel 2. Admin logs in 3. Admin adds product 4. Product is live		
Alternative Flows:			
Exceptions:	1. High response time		
Includes:			

Frequency of Use:	1. Admin initiated
Special Requirements:	nil
Assumptions:	1. Website is in English
Notes and Issues:	

Table 14 UC12

ID for UC:	Uc-12		
Heading for UC	Admin deletes product		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	Admin		
Information:	Admin deletes product		
Trigger:	Admin wants to delete product		
Preconditions:	1. Admin must have internet 2. admin must have access to website		
Post conditions:	Product is removed		
Normal Flow:	1. Admin opens panel 2. Admin selects product 3. Admin presses delete 4. Product is Deleted		
Exceptions:	1. High response time		
Frequency of Use:	1. Admin initiated		
Special Requirements:	nil		
Assumptions:	1. Website is in English		
Notes and Issues:			

Table 15 UC13

ID for UC:	Uc-13		
Heading for UC	Admin edits product		
Creation by:	Maaz	Last Changes made by:	Maaz
Creation Date:	5-4-2023	Date of Revision (Latest):	5-4-23
Actors:	Admin		
Information:	Admin edit product		
Trigger:	Admin wants to edit product		
Preconditions:	3. Admin must have internet 4. admin must have access to website		
Post conditions:	Product is edited		
Normal Flow:	5. Admin opens panel 6. Admin selects product 7. Admin presses edit 8. Product is edited		
Exceptions:	2. High response time		
Frequency of Use:	2. Admin initiated		
Special Requirements:	nil		
Assumptions:	2. Website is in English		
Notes and Issues:			

3.6 User interfaces (mock screens)

Prototype1: (P1) Main screen

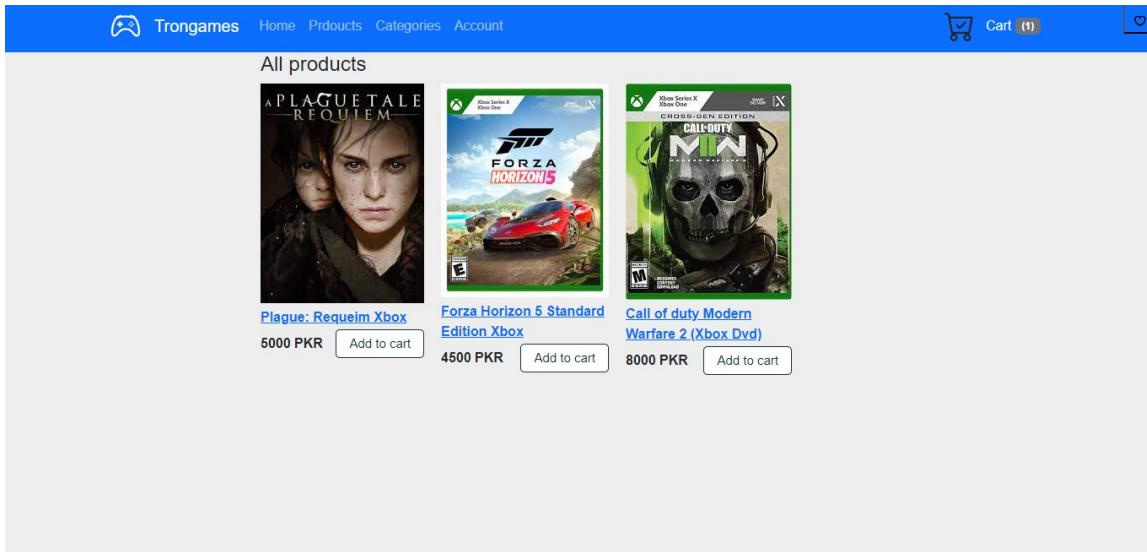
This is the main page for the website



Add new product		
PRODUCT NAME		
Call of duty Modern Warfare 2 (Xbox Dvd)	Edit	Delete
Forza Horizon 5 Standard Edition Xbox	Edit	Delete
Plague: Requiem Xbox	Edit	Delete

Figure 2: Admin panel

Prototype2: (P2) Product 1



Trongames admin

- [Dashboard](#)
- [Products](#)
- [Categories](#)
- [Orders](#)
- [Admins](#)
- [Settings](#)
- [Logout](#)

Hello, **Maaz Saleem**

Maaz Saleem

Orders

TODAY	THIS WEEK	THIS MONTH
0 0 orders today	1 1 orders this week	4 4 orders this month

Revenue

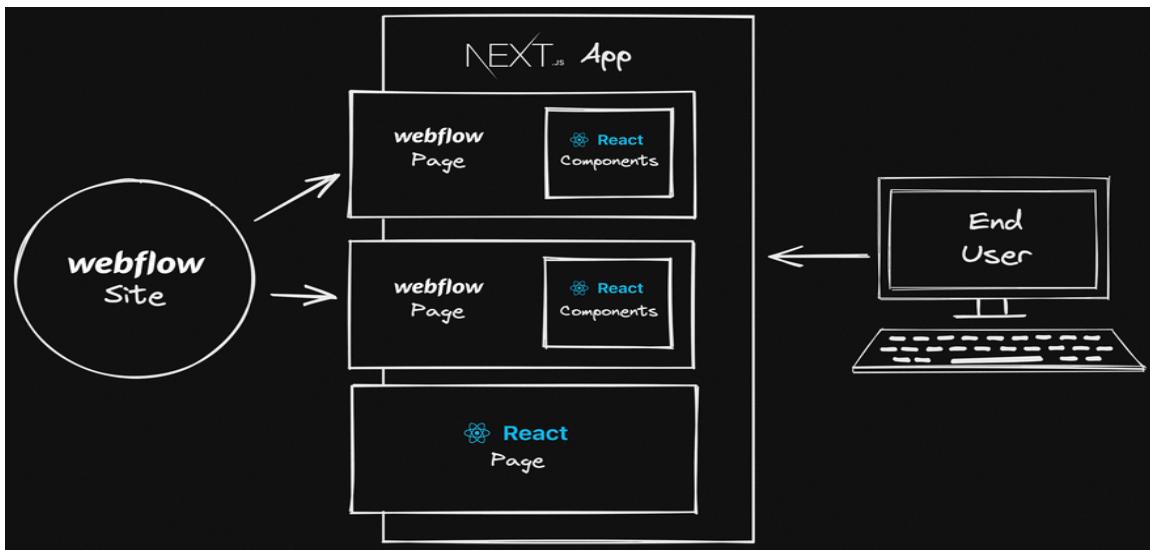
TODAY	THIS WEEK	THIS MONTH
\$ 0 0 orders today	\$ 224 1 orders this week	\$ 1 582 4 orders this month

Prototype3: Admin

4. SYSTEM DESIGN

4.1 System Architecture Diagram

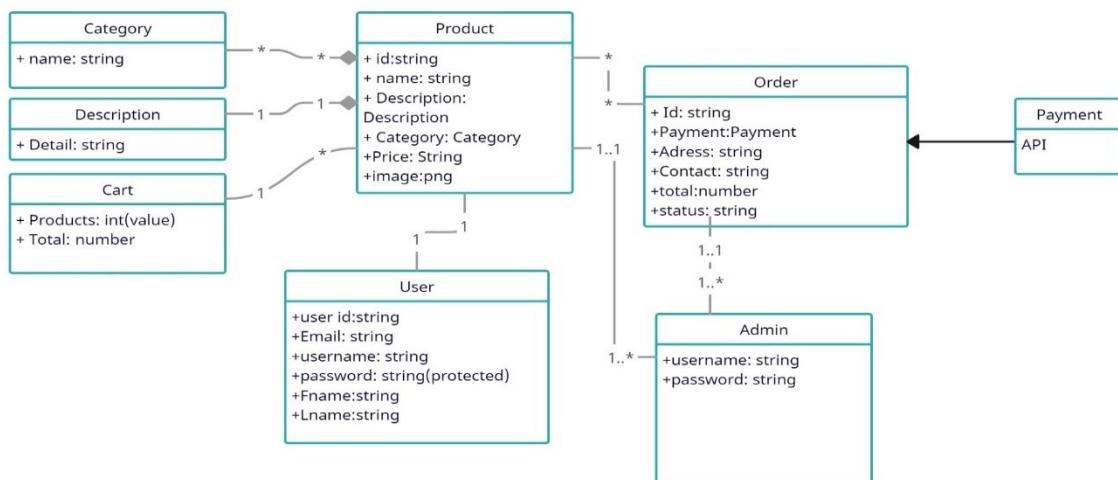
System architecture figure



4.1 Class Diagram

Class diagram

4.1.1 Diagram



4.1 Sequence Diagrams

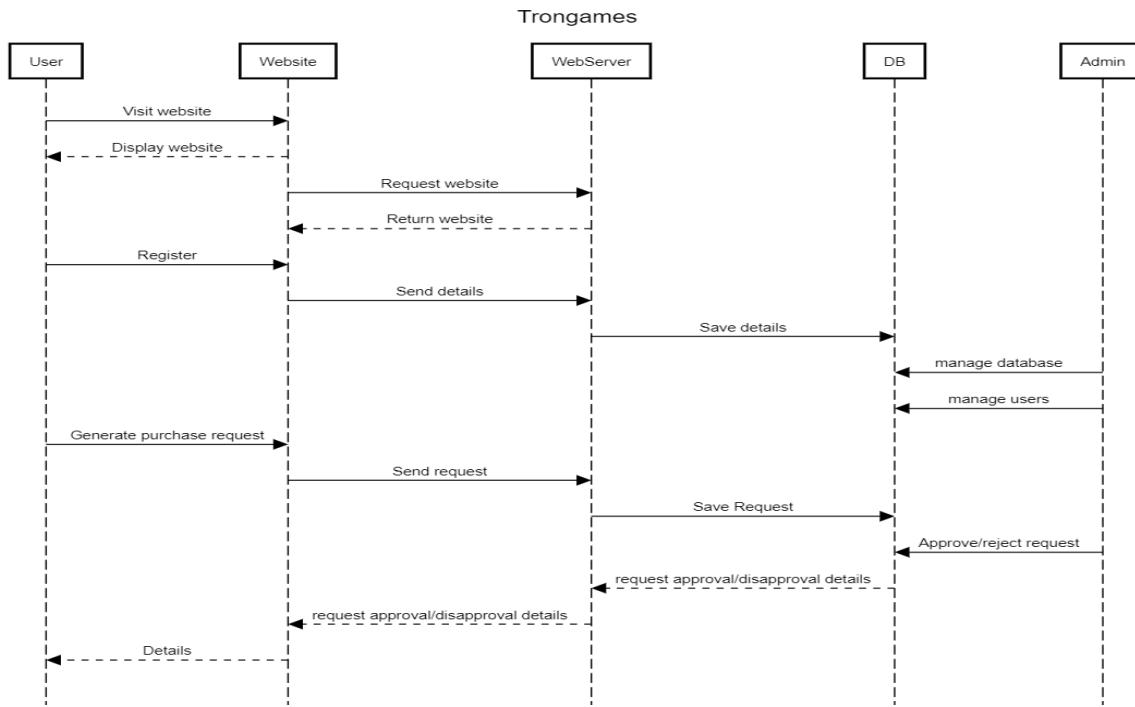


Figure 3 Sequence diagram

4.2 Collaboration Diagrams

1. Purchase request

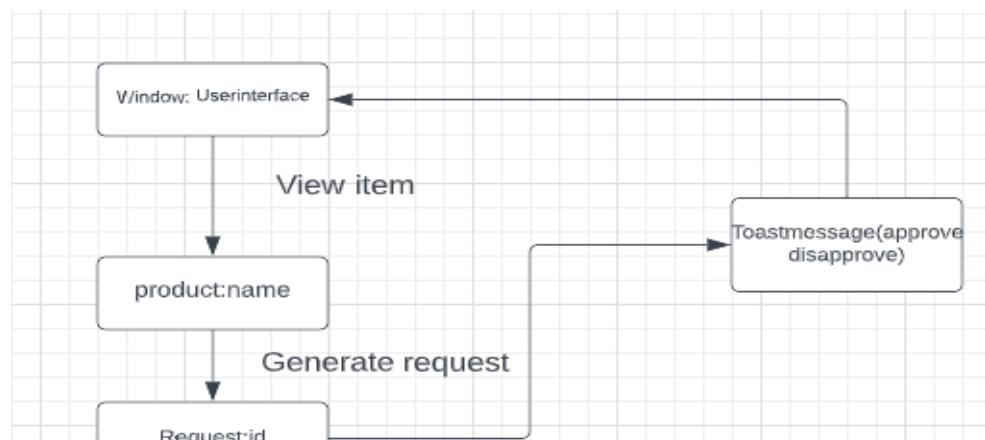


Figure 4 Collaboration

Request for a new product (not in catalogue)

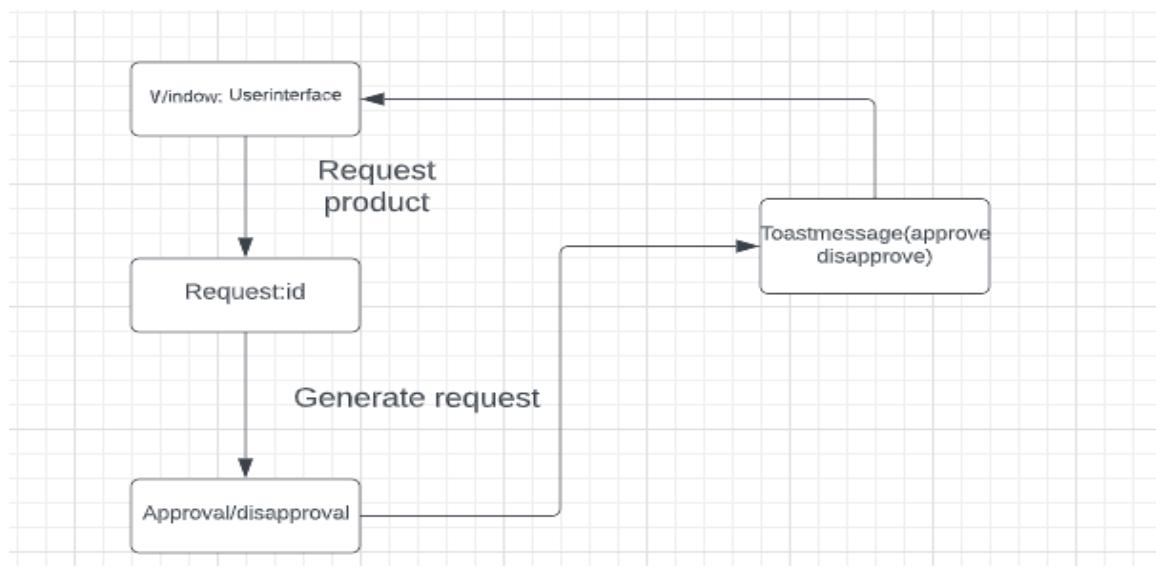
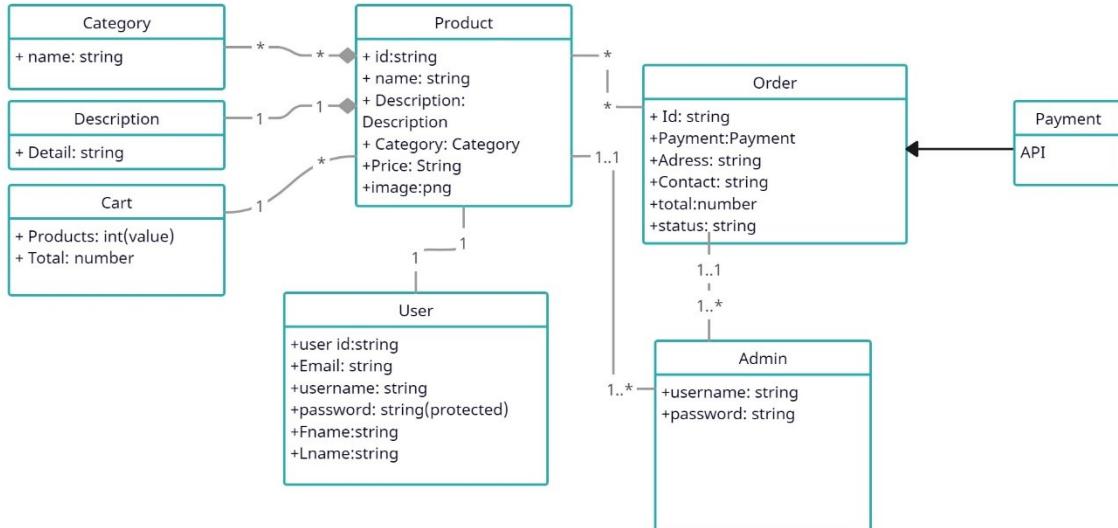


Figure 5 Collaboration

4.1 ERD (same as class diagram)

4.1.1 ERD(figure)



5. IMPLEMENTATION DETAILS

5.1 Development Setup

Tools

The tools used in this are:

Visual studio code: The visual studio code is used to code the website.

MongoDB console: MongoDB console on its website used to monitor the changes and space in the database.

Technology

JavaScript: It is the technology used to code scripts inside the website.

NextJs

Bootstrap

React

Html 5: Html 5 is the technology used. It also includes the protocols used such as FTP, post, get and update.

5.2 Deployment setup

The website is Deployed to Vercel with domain set up for it.

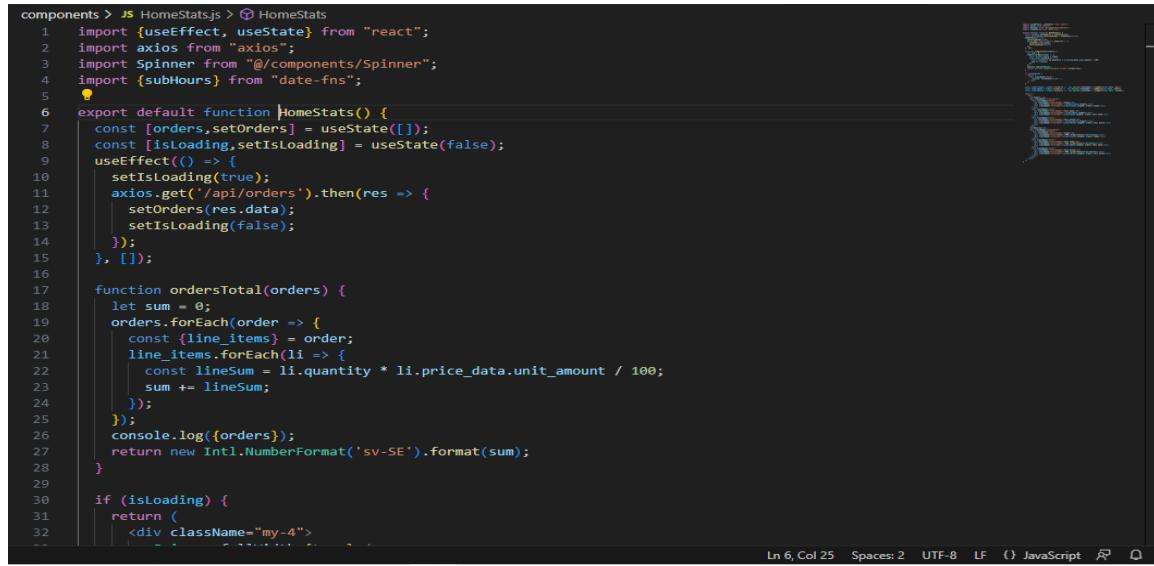
5.3 Algorithms

Figure 6 Algorithm-ii



```
components > JS ProductForm.js > ProductForm
  1 import {useEffect, useState} from "react";
  2 import {useRouter} from "next/router";
  3 import axios from "axios";
  4 import Spinner from "@/components/Spinner";
  5 import {ReactSortable} from "react-sortablejs";
  6
  7 export default function ProductForm({
  8   _id,
  9   title:existingTitle,
 10  description:existingDescription,
 11  price:existingPrice,
 12  images:existingImages,
 13  category:assignedCategory,
 14  properties:assignedProperties,
 15 }) {
 16   const [title,setTitle] = useState(existingTitle || '');
 17   const [description,setDescription] = useState(existingDescription || '');
 18   const [category,setCategory] = useState(assignedCategory || '');
 19   const [productProperties,setProperty] = useState(assignedProperties || []);
 20   const [price,setPrice] = useState(existingPrice || '');
 21   const [images,setImage] = useState(existingImages || []);
 22   const [goToProducts, setGoToProducts] = useState(false);
 23   const [isUploading, setIsUploading] = useState(false);
 24   const [categories, setCategories] = useState([]);
 25   const [categoriesLoading, setCategoriesLoading] = useState(false);
 26   const router = useRouter();
 27   useEffect(() => {
 28     setCategoriesLoading(true);
 29     axios.get('/api/categories').then(result => {
 30       setCategories(result.data);
 31       setCategoriesLoading(false);
 32     })
 33   })
 34 }
```

- The product upload uses image uploading to directly Amazon s3 bucket. Conveniently storing the links in order to directly access when to display the product.



```

components > JS HomeStats.js > HomeStats
1 import {useEffect, useState} from "react";
2 import axios from "axios";
3 import Spinner from "@/components/Spinner";
4 import {subHours} from "date-fns";
5
6 export default function HomeStats() {
7   const [orders, setOrders] = useState([]);
8   const [isLoading, setIsLoading] = useState(false);
9   useEffect(() => {
10     setIsLoading(true);
11     axios.get('/api/orders').then(res => {
12       setOrders(res.data);
13       setIsLoading(false);
14     });
15   }, []);
16
17   function ordersTotal(orders) {
18     let sum = 0;
19     orders.forEach(order => {
20       const {line_items} = order;
21       line_items.forEach(li => {
22         const lineSum = li.quantity * li.price_data.unit_amount / 100;
23         sum += lineSum;
24       });
25     });
26     console.log({orders});
27     return new Intl.NumberFormat('sv-SE').format(sum);
28   }
29
30   if (isLoading) {
31     return (
32       <div className="my-4">
33         <Spinner/>
34       </div>
35     );
36   }
37
38   return (
39     <div>
40       <h2>Total sales: {ordersTotal(orders)}</h2>
41       <table border="1">
42         <thead>
43           <tr>
44             <th>Product</th>
45             <th>Quantity</th>
46             <th>Price</th>
47           </tr>
48         </thead>
49         <tbody>
50           {orders.map((order) => {
51             const {line_items} = order;
52             return line_items.map((li) => {
53               const price = li.price_data.unit_amount / 100;
54               return (
55                 <tr>
56                   <td>{li.product}</td>
57                   <td>{li.quantity}</td>
58                   <td>{price}</td>
59                 </tr>
60               );
61             });
62           })}
63         </tbody>
64       </table>
65     </div>
66   );
67 }

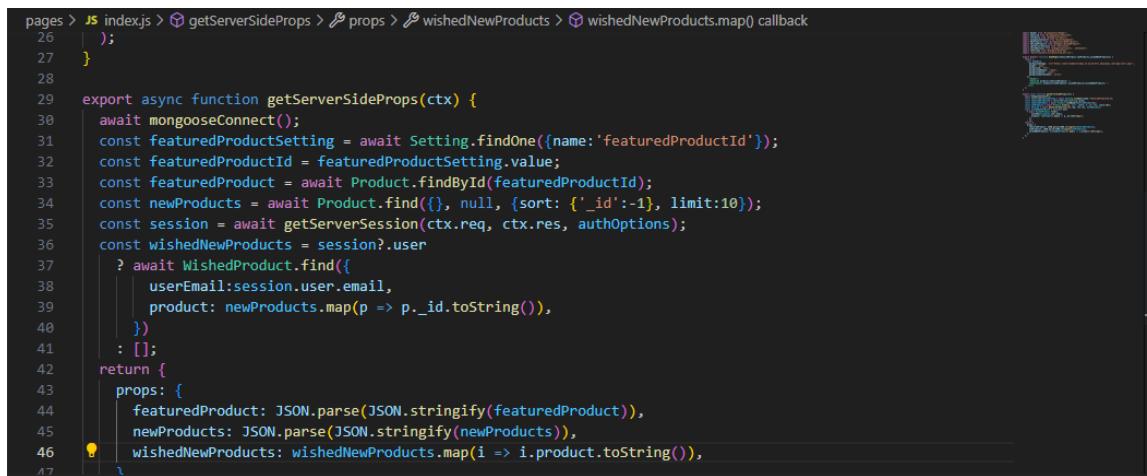
```

Ln 6, Col 25 Spaces: 2 UTF-8 LF () JavaScript

Figure 7: Main admin page uses stats to show sale and purchase. This is the algorithm which works behind it.

Figure 8 Algorithm-iii

- Server-side props to load featured product and information regarding the data beforehand. Taking advantage of SSR (server-side rendering) in NEXT JS



```

pages > JS index.js > getServerSideProps > props > wishedNewProducts > wishedNewProducts.map() callback
26   );
27 }
28
29 export async function getServerSideProps(ctx) {
30   await mongooseConnect();
31   const featuredProductSetting = await Setting.findOne({name:'featuredProductId'});
32   const featuredProductId = featuredProductSetting.value;
33   const featuredProduct = await Product.findById(featuredProductId);
34   const newProducts = await Product.find({}, null, {sort: {'_id':-1}, limit:10});
35   const session = await getSession(ctx.req, ctx.res, authOptions);
36   const wishedNewProducts = session?.user
37     ? await WishedProduct.find({
38       userEmail:session.user.email,
39       product: newProducts.map(p => p._id.toString()),
40     })
41     : [];
42   return {
43     props: {
44       featuredProduct: JSON.parse(JSON.stringify(featuredProduct)),
45       newProducts: JSON.parse(JSON.stringify(newProducts)),
46       wishedNewProducts: wishedNewProducts.map(i => i.product.toString()),
47     }
48   };

```

5.4 Constraints

5.4.1 Assumptions

- Admins will maintain the database
- User can request any type of product in any type of category of genre
- User will interact easily with the interface
- Database will have backup available

5.4.2 System constraints

System constraints includes

- Database restrictions
- Web server responses
- Arising bugs over time

5.4.3 Restrictions

Restrictions applied are as follows:

- Users can request limited number of products in a month
- Request can be declined easily by the admin
- Users can create a single account on one information
- Users can order on guest account too.

5.4.4 Limitations

- It is unable to provide console accessories

6. TESTING

6.1 Extended Test cases

Table 16 TC1

ID for Test Case:	Tc-01	Designed by:	Maaz		
Name for Test Module:	Place Order	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Order placing	Execution date:	5 May, 2023		
Information:	User places an order				
Pre-Condition:	<ol style="list-style-type: none">1. User must have an internet connection.2. User must have access to the website.3. User should have products in the cart				
Dependencies: Cart populated					
Step	Test Step	Test Data			
1.	Open trongames.store				
2.	Select products from the catalogue				
3.	Products can be searched				
4.	Press add to cart				
5.	Click cart				
6.	Enter Email	Saleemmaaz09@gmail.com			
7.	Enter zip code, country	63100, Pakistan			
8.	Enter address	Umt Lahore			
9.	Click on cod/continue to payment				
Expected Result:	Order placed				
Actual Result:	Order gets placed and it shows up in admin panel				
Status:	Pass				
Post Condition:					

Table 17 Tc2

ID for Test Case:	Tc-02	Designed by:	Maaz		
Name for Test Module:	Search product	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Product Search	Execution date:	5 May, 2023		
Information:	User searches a product				
Pre-Condition:	1. User must have an internet connection. 2. User must have access to the website.				
Dependencies:					
Step	Test Step	Test Data			
1.	Open trongames.store				
2.	Press the search icon				
3.	Search page opens up				
4.	Press search bar				
5.	Enter search item	Call of duty			
Expected Result:	Product search				
Actual Result:	Results show up				
Status:	Pass				
Post Condition:					

ID for Test Case:	Tc-03	Designed by:	Maaz		
Name for Test Module:	User sign up	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Sign up	Execution date:	5 May, 2023		
Information:	User signs up				
Pre-Condition:	1. User must have an internet connection. 2. User must have access to the website. 3. User must have a Gmail account.				
Dependencies:					
Step	Test Step	Test Data			
1.	Open trongames.store				
2.	Select account				
3.	Click login				
4.	Authenticate google id				
5.	Input info if not filled already				
6.	Enter Email	Saleemmaaz09@gmail.com			
7.	Enter zip code, country	63100, Pakistan			
8.	Enter address	Umt Lahore			
9.	Click Save				
Expected Result:	User signs up				
Actual Result:	Signed up				
Status:	Pass				
Post Condition:					

Table 18 Tc03

Table 19 Tc4

ID for Test Case:	Tc-04	Designed by:	Maaz
Name for Test Module:	User login	Date of Design:	5 May, 2023
Priority of Test:	High	Execution by:	Maaz
Name of test:	Login User	Execution date:	5 May, 2023
Information:	User Logs in		
Pre-Condition:	1. User must have an internet connection. 2. User must have access to the website. 3. User must be signed up.		
Dependencies: Sign up			
Step	Test Step	Test Data	
1.	Open trongames.store		
2.	Select Account		
3.	Select Login		
4.	User is logged in		
Expected Result:	Login		
Actual Result:	User logged in		
Status:	Pass		

Table 20 Tc5

ID for Test Case:	Tc-05	Designed by:	Maaz
--------------------------	-------	---------------------	------

Name for Test Module:	Place Request	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Product request	Execution date:	5 May, 2023		
Information:	User places a request				
Pre-Condition:	1. User must have an internet connection. 2. User must have access to the website.				
Dependencies: product not existing					
Step	Test Step	Test Data			
1.	Open trongames.store				
2.	Select request				
3.	Place a request through email				
4.	Request placed				
Expected Result:	Order placed				
Actual Result:	Order gets placed and it shows up in admin panel				
Status:	Pass				
Post Condition:					

ID for Test Case:	Tc-06	Designed by:	Maaz		
Name for Test Module:	Product listing	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Product list	Execution date:	5 May, 2023		
Information:	Admin adds a new product				
Pre-Condition:	1. Admin must have an internet connection. 2. Admin must have access to the admin panel.				
Dependencies: Admin should be authorized.					
Step	Test Step	Test Data			
1.	Open Tron games admin				
2.	Select products				
3.	Select add product				
4.	Add name	Call of duty			
5.	Add Information				
6.	Add price	4500			
7.	Select product added				
Expected Result:	Product listed				
Actual Result:	Product shows up on website				
Status:	Pass				
Post Condition:					

Table 21 Tc6

ID for Test Case:	Tc-07	Designed by:	Maaz		
Name for Test Module:	Edit product	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Product edit	Execution date:	5 May, 2023		
Information:	Admin edits a product listing				
Pre-Condition:	1. Admin must have an internet connection 2. Admin must have access to admin panel				
Dependencies: Product should exist					
Step	Test Step	Test Data			
1.	Open Tron games admin				
2.	Select products				
3.	Select edit product				
4.	Edit details				
Expected Result:	Product details changed				
Actual Result:	Product details show up updated				
Status:	Pass				
Post Condition:					

Table 22 TC7

Table 23 TC8

ID for Test Case:	Tc-08	Designed by:	Maaz		
Name for Test Module:	Delete product	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Product deletion	Execution date:	5 May, 2023		
Information:	Admin deletes product				
Pre-Condition:	1. Admin must have an internet connection 2. Admin must have authorization.				
Dependencies: product should exist.					
Step	Test Step	Test Data			
1.	Open Tron games admin				
2.	Select products				
3.	Select delete option				
4.	Approve delete option				
Expected Result:	Product deletion				
Actual Result:	Product gets delisted from website				
Status:	Pass				

Table 24 TC09

ID for Test Case:	Tc-09	Designed by:	Maaz		
Name for Test Module:	Logout	Date of Design:	5 May, 2023		
Priority of Test:	High	Execution by:	Maaz		
Name of test:	Logout	Execution date:	5 May, 2023		
Information:	Logout				
Pre-Condition:	1. User must have logged in				
Dependencies: Logged in					
Step	Test Step	Test Data			
1.	Open Tron games				
2.	Select account				
3.	Select logout				
4.	Logged out				
Expected Result:	Account logged out				
Actual Result:	Page refreshes and logs out				
Status:	Pass				
Post Condition:					

7. RESULTS/OUTPUT/STATISTICS(TRACEABILITY MATRIX)

7.1 requirements vs use cases

UCID/RID	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6	FR 7	FR 8	FR 9	FR 10	FR 11	FR 12
UC 1					★		★					
UC 2		★	★		★							
UC 3					★		★					
UC 4											★	
UC 5	★	★										
UC 6		★										
UC 7												★
UC 11								★				
UC 12									★			

7.2 Test Cases (RID vs TID)

UCID/RID	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6		FR 7	FR 8	FR 9	FR 10	FR 11	F R 12
TC 1			★			★							
TC 2								★					
TC 3	★												
TC 4		★											
TC 5			★	★									
TC 6									★				
TC 7										★			
TC 8											★		
TC 9													★

7.3 Coverage (UCID vs TID)

UCID/TID	T C 1	T C 2	T C 3	T C 4	T C 5	T C 6	T C 7	T C 8
UC 1		★						
UC 2	★							
UC 5			★					
UC 6				★				
UC 9	★							
UC 11						★		
UC12							★	
UC13							★	

7.4 Results

7.4.1 %completion

Completion stands at 91.6 percent.

7.4.2 %accuracy

Accuracy stands at 91 percent

7.4.3 %correctness

Correctness stands at 87 percent.

8. FUTURE WORK

- Future work includes testing of the prototype, expansion of system and creating a mobile app
- Filling the final report with all the left-out details which may come from system analysis
- Adding more requirements upon testing and usage
- The cart option isn't added in the website. It will be added in the coming week because the website design is to be finalized and soon after that the website will reach its final stages
- The login and sign-out option with sessions will be added simultaneously with cart option
- The website will be completed after the addition of few features after the design is finalized
- The application will be soon being developed after the website.
- The prototype of the ui is throw away prototype.
- Wishlist to be added

9. BIBLIOGRAPHY

9.1 Articles

<https://intenta.digital/gaming-industry/video-game-industry-pakistan/>

9.2 Research papers

<https://www.ijert.org/research/comparative-study-of-pc-and-gaming-console-for-video-games-IJERTCONV8IS05060.pdf>

9.3 Other References

- <https://www.gamestop.com/>
- <https://www.bestbuy.com/>
- <https://www.cdkeys.com/>

**Final Year Project Report
E Folder Generator Application**



**Project Advisor:
Sir Rana Marwat Hussain**

**Submitted By:
MOEEZ IMRAN (S2020266072)**

**Session
2020-2023**

**University of Management and Technology
C-II Johar Town Lahore Pakistan**

Dedication

“The world is changed by your example, not by your opinion.” – ***Paulo Coelho***.

Technology is the true supporter and friend of human being if it is used with positive approach. It can minimize the human efforts. Dedicated by this point, a keen effort is made to facilitate the UMT Family in order to reduce human efforts for managing the course folders and related print out pages. As well as to make a better, environment friendly and cost efficient system as compared to printing.

Title of Project: E-folder Application

Student Name : Moeez Imran
and ID S2020266072

Final Approval

1. T. 8/8/23

- Head of Department
Department of Computer Science
School of Systems & Technology
UMT Lahore

Thufail Ahmad
9/8/2023

- Program Director (Final Year Projects)
Department of Computer Science
School of Systems & Technology
UMT Lahore

RJ 8/8/2023

- Supervisor
Department of Computer Science
School of Systems & Technology
UMT Lahore

- Co-Supervisor _____

Acknowledgment

I would like to express my deepest appreciation to **Mr. Rana Marwat** who has helped me to achieve the whole solution of printing problem by clearing my doubts, confusions and concepts related to “Web Technology”. This endeavor would not have been possible without him.

University of Management and Technology, Lahore

Similarity Report

Turnitin Originality Report
E Folder Generator Application by Moeez Imran

From Reference Desk (LRC info desk)

- Processed on 19-Jul-2023 13:21 PKT
- ID: 2133489941
- Word Count: 3117

Similarity Index

2%

Similarity by Source

Internet Sources:

1%

Publications:

0%

Student Papers:

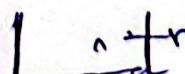
2%

Sources:

- 1% match (student papers from 04-Sep-2020)
Submitted to Higher Education Commission Pakistan on 2020-09-04
- < 1% match (student papers from 27-Jul-2018)
Submitted to Higher Education Commission Pakistan on 2018-07-27



Checked by



Verified by CLO

Note:

- Sometimes the overall similarity index may be smaller than the repository percentages combined. This would be due to overlapping text within the repositories.
- It is a system generated report.

Project Title

E-Folder Generator Application (Web-Based)

Objective

A website based project to replace traditional printing process.

Undertaken by

Sir Rana Marwat Hussain

Supervised by

Sir Rana Marwat Hussain

Starting Date

October 2022

Completion Date

Still Working

Tools Used

- **Programming Language:** PHP Core
- **For Writing & Testing Code:** Visual Code / Sublime Editor (Latest Version)
 - **Compiling & Debugging the Code:** Chrome's V8 Engine
 - **Local Hosting Server:** Xampp Server
 - **Local Backend (Database) Support:** MySQL Server

Operating System

Windows

Documentation

The whole in-depth documentation and report is shared as follow

Declaration Form

I have carefully examined the documentation of the Final Year Project titled “*E Folder Generator Application*”; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with user manual and installation guide.

FYP Advisor Name: Sir Rana Marwat Hussain

Signature: _____

Date: _____

Abstract

It is a Web-Based Digital solution for traditional printing which will minimize the human effort to manage the printed papers. Moreover, it is a more cost efficient and environment friendly process as compared to regular printing. Initially it is just for our UMT family. This whole system will have certain roles with specific and limited functionalities and allowances e.g CODs of respective departments, Professors, Coordinators, Admin etc. COD can enter new users by the request from admin. Admin have all the rights such as to insert, to delete, to update and to search any user. Coordinators can assign courses and sections of a certain subject to a certain teacher. Teachers can upload the respective course content in the E-folder. This will save much time, effort and cost.

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
<i>Draft</i>	Moeez Imran	Initial draft created for distribution and review comments	15/11/2022
<i>Preliminary</i>	Moeez Imran	Second draft incorporating initial review comments, distributed for final review	5/12/2022
<i>Final</i>	Moeez Imran	First complete draft, which is placed under change control	23/1/2023
<i>Revision 1</i>	Moeez Imran	Revised draft, revised according to the change control process and maintained under change control	31/1/2023
<i>Revision 2</i>	Moeez Imran	Revised draft, revised according to the change control process and maintained under change control	7/2/2023

CONTENTS

CONTENTS	1
DEFINITIONS AND ACRONYMS.....	3
LIST OF FIGURES	4
LIST OF TABLES.....	5
1. INTRODUCTION.....	6
1.1 MOTIVATIONS	6
1.2 PROJECT OVERVIEW.....	6
1.3 PROBLEM STATEMENT	7
1.4 OBJECTIVES.....	7
2. DOMAIN ANALYSIS	8
2.1 CUSTOMER	8
2.2 STAKEHOLDERS.....	8
2.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT.....	8
2.4 DEPENDENCIES/ EXTERNAL SYSTEMS	8
2.5 REFERENCE DOCUMENTS	8
2.5.1 <i>Related Projects</i>	9
2.5.2 <i>Feature Comparison</i>	9
3. REQUIREMENTS ANALYSIS.....	10
3.1 REQUIREMENTS	10
3.2 LIST OF ACTORS AND USE CASES	11
3.3 SYSTEM USE CASE DIAGRAM	13
3.4 EXTENDED USE CASES	13
4. DATA FLOW DIAGRAM	15
4.1 DATA FLOW DIAGRAM.....	15
5. SYSTEM DESIGN	16
5.1 SYSTEM ARCHITECTURE DIAGRAM	16
5.2 CLASS DIAGRAM	17
5.3 SEQUENCE DIAGRAMS.....	17
5.4 COLLABORATION DIAGRAMS	18
5.5 ERD	27
6. IMPLEMENTATION DETAILS	29
6.1 DEVELOPMENT SETUP	29
6.2 DEPLOYMENT SETUP	29
6.3 ALGORITHMS.....	29
6.4 CONSTRAINTS.....	30
6.4.1 <i>Assumptions</i>	30
6.4.2 <i>System constraints</i>	30
6.4.3 <i>Restrictions</i>	31
6.4.4 <i>Limitations</i>	31
7. TESTING.....	32
7.1 EXTENDED TEST CASES.....	32
7.2 SOURCE CODE	33
7.2.1 <i>Code snippet</i>	33

8. RESULTS/OUTPUT/STATISTICS.....	44
8.1 %COMPLETION.....	44
8.2 %ACCURACY	44
8.3 %CORRECTNESS.....	44
9. CONCLUSION	45

Definitions and Acronyms

Table 1 List of Definitions

Acronym	Definition
UMT	University of Management and Technology
COD	Chairperson of the Department
E-folder	Electronic Folder
DB	Data bases
e.g	For Example
JS	JavaScript

List of Figures

Table 1 List of Definitions.....	3
Table 2 Stakeholders and their Roles in System.....	8
Figure 1: Use Case Diagram.....	13
Figure 2 : Data Flow Diagram	15
Figure 3: System Architecture	16
Figure 4: Class Diagram	17
Figure 5: Sequence Diagram for LOGIN (ALL ACTORS).....	18
Figure 6: Sequence Diagram for LOGOUT (ALL ACTORS).....	19
Figure 7: Sequence Diagram for ASSIGNING FACULTY (COORDINATOR)	20
Figure 8: Sequence Diagram for UPLOAD CONTENT (TEACHER).....	21
Figure 9: Sequence Diagram for DELETE USER (ADMIN).....	22
Figure 10: Sequence Diagram for INSERT NEW USER (ADMIN).....	23
Figure 11: Sequence Diagram for SEARCH USER (ADMIN AND COD)	24
Figure 12: Sequence Diagram for SEE USERS (ADMIN AND COD)	25
Figure 13: Sequence Diagram for UPDATE DETAILS (ALL ACTORS).....	26
Figure 14: Collaboration Diagram.....	27
Figure 15: Entity Relation Diagram (ERD)	28

List of Tables

Table 1 List of Definitions.....	3
Table 2 Stakeholders and their Roles in System.....	8
Table 3 Test-1 (Login).....	32
Table 4 test-2 (Uploading).....	32
Table 5 Test-3 (Downloading).....	33

1. INTRODUCTION

1.1 Motivations

The day by day increasing prices of printing papers as well as headache and the work load to manage these papers is frustrating. Therefore, there should be such a solution that should do this work cost efficiently just by one click. Also it should be environment friendly as well energy efficient. So, this whole need motivated me to create such a project which have all these capabilities.

1.2 Project Overview

Overall, this whole project and idea is only functional for UMT faculty. A website that will give access to only authorized persons and allow them to perform specific tasks within their limitations. There are respective roles for respective tasks. COD can enter new users by the request from admin. Admin have all the rights such as to insert, to delete, to update and to search any user. Coordinators can assign courses and sections of a certain subject to a certain teacher. Teachers can upload the respective course content in the E-folder. This will save much time, effort and cost.

- PROBLEM TO BE SOLVED**

As we all know that printer is a vital need in almost every sector. But it is not quite environment friendly. The printer pages which are much expenses come from the trees and printer consumes a significant amount of energy. Moreover it emits Carbon dioxide and monoxide leaving a much significant carbon footprint and these pages are much difficult to handle and store.

- CUSTOMER**

UMT faculty (CODs, Teachers, Coordinators etc) and a dedicated Admin.

- GOALS**

To create an easy, cost efficient, environment friendly and secure web based alternative of traditional printing.

- SYSTEM FUNCTIONS**

In this system every user will have a limited access based on his/her role defined. Admin can see, delete, insert and update users. COD can see the faculty as well as add new users on the request of Admin. Coordinators can assign section and courses to respective teachers and teachers can upload the course content in the folders.

1.3 Problem Statement

As we all know that printer is a vital need in almost every sector. But it is not quite environment friendly. The printer pages which are much expenses come from the trees and printer consumes a significant amount of energy. Moreover it emits Carbon dioxide and monoxide leaving a much significant carbon footprint and these pages are much difficult to handle and store. Therefore, a system just like E-folder Generator Application is required to solve such problems.

1.4 Objectives

On completion, our project can give following benefits:

- Expenses for printing pages can be reduced effectively
- Less Effort for managing papers
- Less energy consumption and emission
- Environment friendly system alternative of printing

2. DOMAIN ANALYSIS

2.1 Customer

Our basic customer is UMT itself. In an order to pay tribute to my learning hub I created this project for UMT's faculty and its other staff as well.

2.2 Stakeholders

Table 2 Stakeholders and their Roles in System

Stakeholder	Role in System
Admin	Admin is responsible for all the technical decisions. To insert, delete, see and update users.
COD	COD of respective department is responsible to see users or faculty of his/her own department. He can also add new users on the request form Admin.
Teacher	Teacher of a respective course is responsible to upload the content of the course assigned to him in the E-folder.
Coordinator	Coordinator of a department is responsible to assign courses and the sections of a respective subject to respective teachers.

2.3 Groups affected with economic impact

- FACULTY**

This idea can reduce the paper work for the faculty of UMT and will reduce their work load by making every paper managing related activity easy and secure.

2.4 Dependencies/ External Systems

This project does not require any such external systems for its completion. All we need is just the permission from UMT authorities to implement this system and to make it functional practically.

2.5 Reference Documents

No such documents have been consulted during the analysis phase. As it is purely an idea of mine. Therefore, we need no references that show the consulted documents.

2.5.1 Related Projects

According to my studies, research and observation. No such related idea or related project has been presented by any other person at present. As it is solemnly a new approach therefore it is my good fate to be the first person to represent such an environment friendly, cost and energy efficient and less effort consuming idea.

2.5.2 Feature Comparison

As there is no such related project available at present. Therefore, the feature comparison procedure is not possible until a new project related to this idea is presented by another person.

3. REQUIREMENTS ANALYSIS

3.1 Requirements

The whole data and details of each user are saved in the databases for further actions.

Following are some requirements of our project:

- ExamineFaculty:**

A functional requirement that allows its user to see the faculty members. COD and admin can use this functional requirement only.

- SearchFaculty:**

A functional requirement that allows the user to search a certain member of the faculty of a certain department. It can be used by COD, Admin and coordinator only.

- UpdateFaculty:**

A functional requirement that allows the user to update any details of a certain member of the faculty of a certain department. It can be used by Admin, Coordinator, COD and Faculty (teacher) only.

- InsertNewFaculty:**

A functional requirement that allows the user to insert new faculty members with unique id number and a strong password. It can be used by Admin only.

- DeleteFaculty:**

A functional requirement that allows the user to delete a certain member of the faculty of a certain department. It can be used by Admin only.

- AssignFaculty:**

A functional requirement that allows only the coordinator to assign a certain faculty member of his own department for a specific course with the section allocated.

- **UploadContent:**
A functional requirement that allows the faculty member (teachers) to upload the assigned courses content in the e-folder.
- **Login:**
A functional requirement used by all users that checks if the id and password being entered are present in DB or not with respective action.
- **Logout:**
A functional requirement used by all users to exit their own profiles
- **UpdatePassword:**
A non-functional requirement used by all users to update the password of their own profiles.

3.2 List of Actors and Use Cases

Followings are the actors that are functional in our whole project:

- **COD:**
COD of respective department is responsible to see users or faculty of his/her own department. He can also add new users on the request form Admin.
- **Admin:**
Admin is responsible for all the technical decisions. To insert, delete, see and update users.
- **Coordinator:**
Coordinator of a department is responsible to assign courses and the sections of a respective subject to respective teachers.

- **Faculty Member (Teacher):**

Teacher of a respective course is responsible to upload the content of the course assigned to him in the E-folder.

Followings are the use cases of our project with a brief description:

- **Log In:**

Permit the user to enter account details and gain access to services that are prohibited.

- **Log Out:**

Give the user the option to log out of their personal profile and return to the mock screen.

- **Assigning Faculty:**

Allow the user to assign a specific user a certain course and section.

- **Upload Content:**

Allow the user to upload related content of assigned course.

- **Delete User:**

Allow the user to delete a certain user from DB.

- **Insert New User:**

Allow the user to insert new and registered users in DB.

- **Search User:**

Allow the user to search a registered user in DB.

- **See Users:**

Allow the user to see all the faculty members of a certain department.

- **Update Details:**

Allow the user to update his/her own details excluding some specific details.

3.3 USE Case Diagram

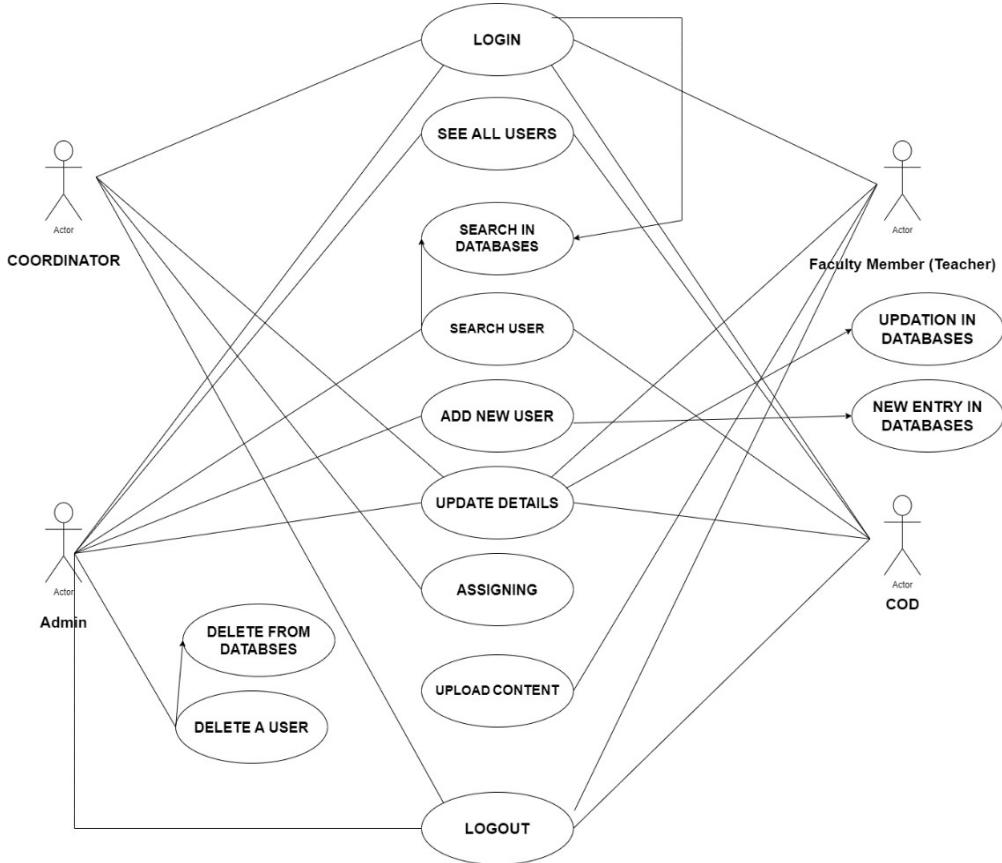


Figure 1: Use Case Diagram of Admin, Coordinator & Faculty

3.4 Extended Use Cases

FIRST USE CASE – LOGIN:

COD, Admin, Faculty Member (Teacher) and Coordinators are the actors for this use case. Every actor can use this use case by entering his/her unique username id and password if the details are matched with the details in DB and after that he/she will be able to use the restricted access according to his/her role in the system. If the details do not match with the DB's details then an ERROR message will be shown to him/her on the screen.

SECOND USE CASE – LOGOUT:

COD, Admin, Faculty Member (Teacher) and Coordinators are the actors for this use case. If the actor have logged in successfully then clicking on the LOGOUT option will bring them out to the main menu or the mock screen.

THIRD USE CASE – ASSIGNING FACULTY:

Coordinator is the actor of this use case. Each coordinator of every department can assign different courses with their sections to respective teachers. No same sections of a course can be allocated to different teachers.

FOURTH USE CASE – UPLOAD CONTENT:

Faculty Member (Teacher) is the actor of this use case. A teacher of a respective course and section can upload all the related material of that course in the E-folder.

FIFTH USE CASE – INSERT USER:

Admin is the actor of this use case. Admin can only insert new users of certain departments only on the approval from COD of that department. Each new user will have a unique username id that cannot be changed and a default password that can be change in future by the will of user.

SIXTH USE CASE – SEARCH USER:

Admin and COD are the actors of this use case. Admin can search each and every user of the system using the unique username id generated to each user. If that matches with DB then searching event will occur successfully. Otherwise an ERROR message will be shown. On the other hand, COD of a specific department can only search a user using the same principal but the user must have to be from that specific department.

SEVENTH USE CASE – SEE USERS:

Admin and COD are the actors of this use case. Admin can see each and every user of the system. On the other hand, COD of a specific department can only see users related to that specific department.

EIGHTH USE CASE – UPDATE DETAILS:

Admin, COD, Faculty Member (Teacher) and Coordinator are the actors of this use case. Every user after logging in successfully will be given an option of updating his/her details except his/her unique username id.

4. DATA FLOW DIAGRAM

4.1 Data Flow Diagram

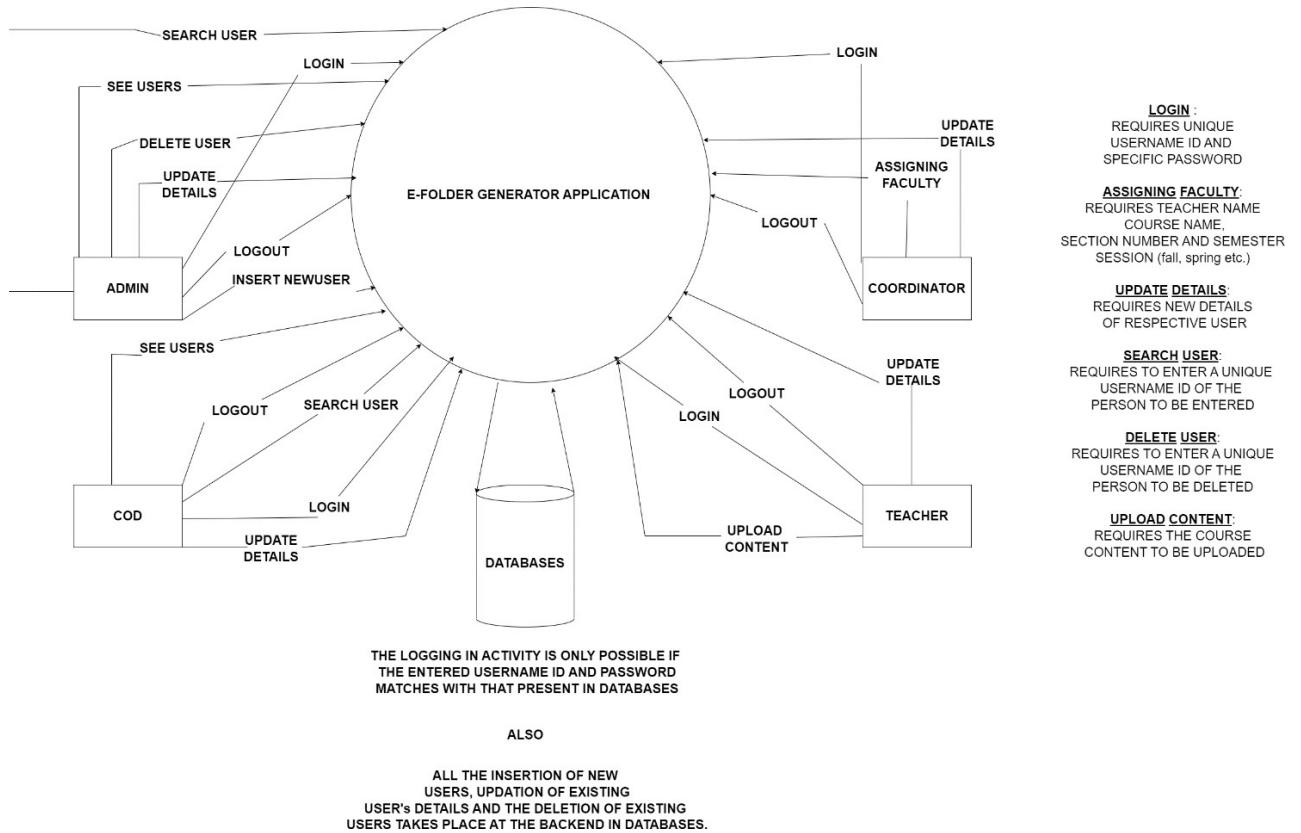


Figure 2 : Data Flow Diagram of E-Folder Application

5. SYSTEM DESIGN

5.1 System Architecture Diagram

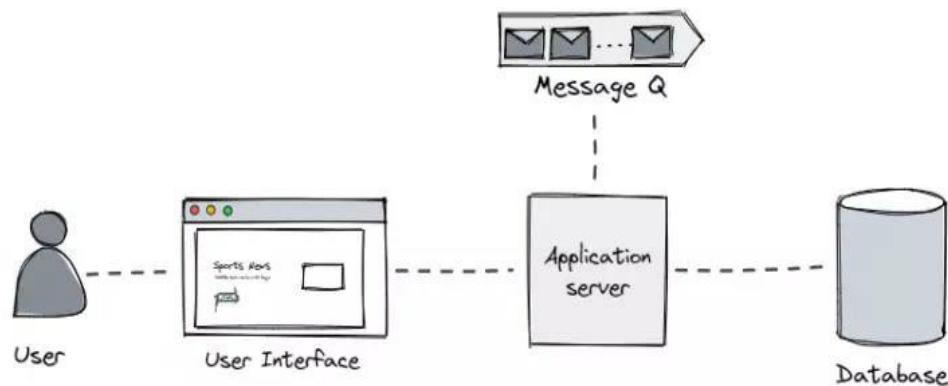


Figure 3: System Architecture of E-Folder Application

5.2 Class Diagram

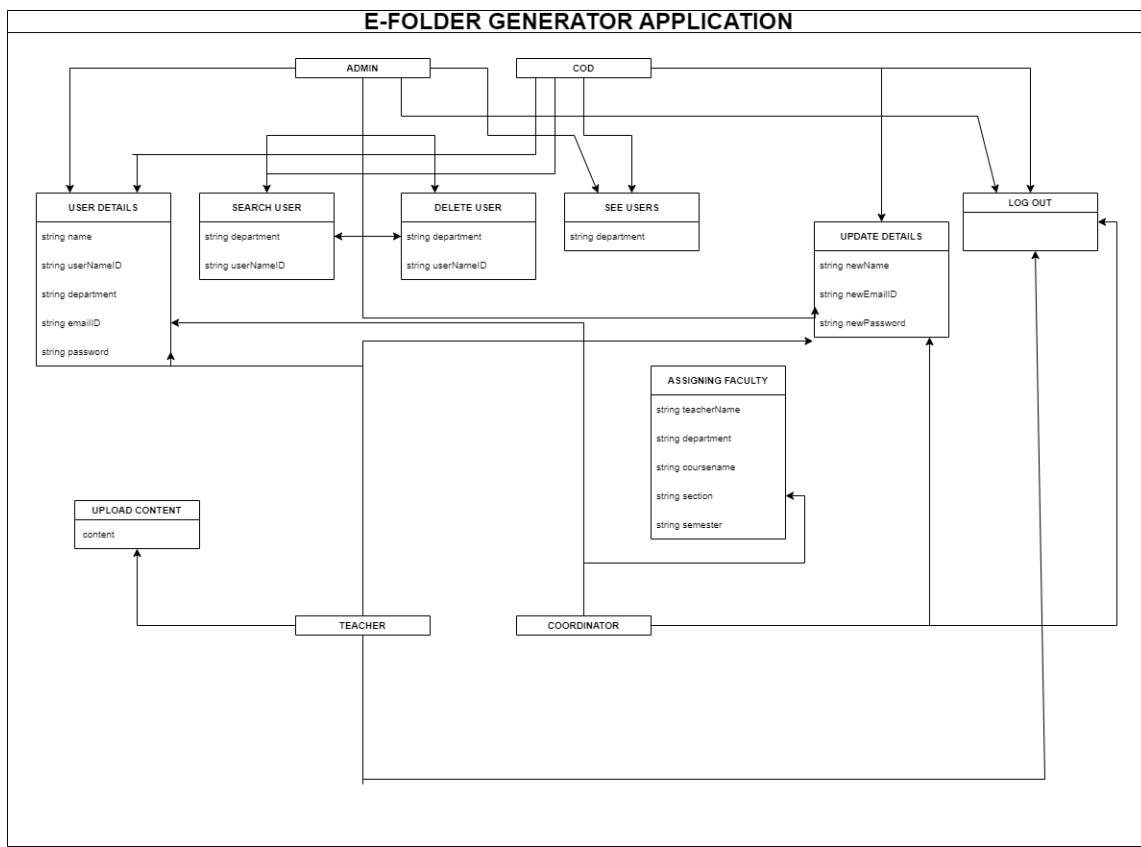


Figure 4: Class Diagram of E-Folder Application

5.3 Sequence Diagrams

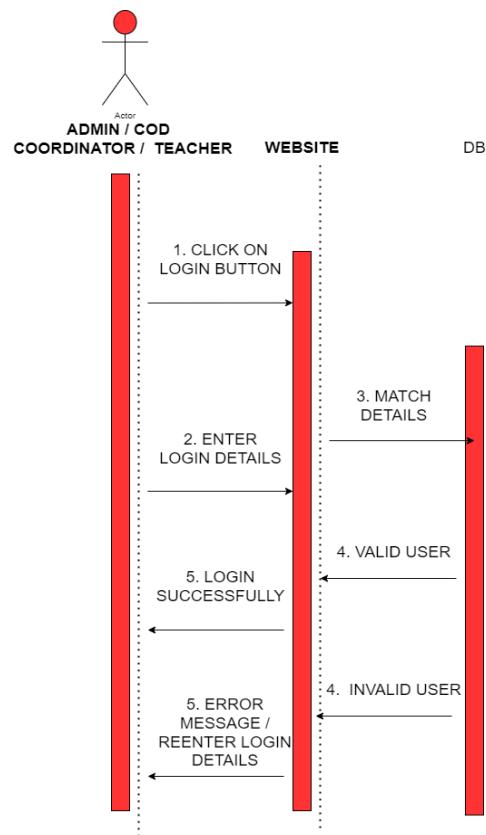


Figure 5: Sequence Diagram for LOGIN (ALL ACTORS) of E-Folder Application

ALL ACTORS AFTER LOGGING IN SUCCESSFULLY
WILL BE PROVIDED WITH A PROMINENT OPTION OF LOG OUT

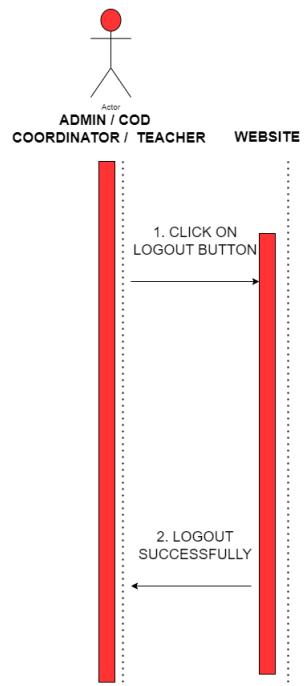


Figure 6: Sequence Diagram for LOGOUT (ALL ACTORS) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY COORDINATOR WILL BE PROVIDED WITH A PROMINENT OPTION OF ASSIGNING FACULTY

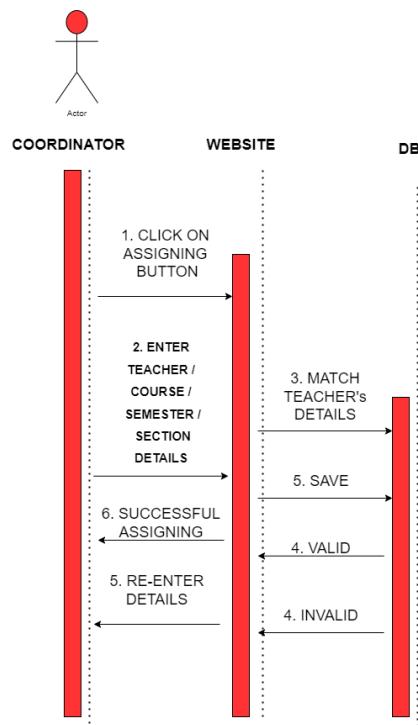


Figure 7: Sequence Diagram for ASSIGNING FACULTY (COORDINATOR) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY TEACHER WILL BE PROVIDED
WITH A PROMINENT OPTION OF UPLOAD CONTENT

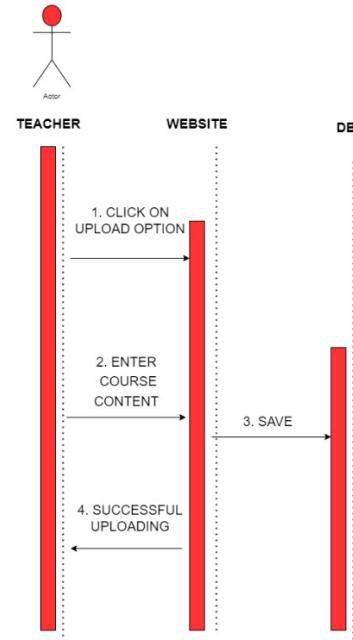


Figure 8: Sequence Diagram for UPLOAD CONTENT (TEACHER) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY ADMIN WILL BE PROVIDED WITH A PROMINENT OPTION OF DELETE USER

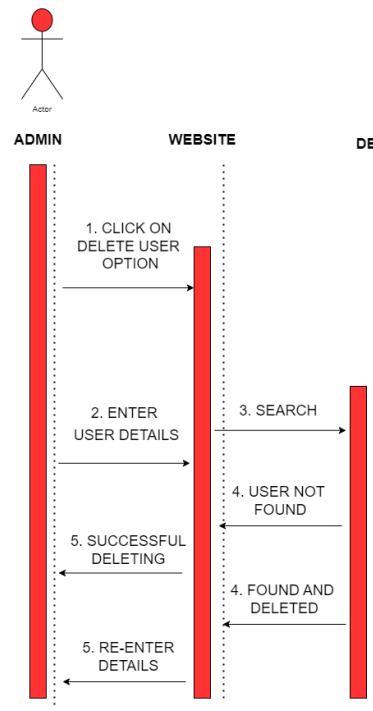


Figure 9: Sequence Diagram for DELETE USER (ADMIN) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY ADMIN WILL BE PROVIDED
WITH A PROMINENT OPTION OF INSERT NEW USER

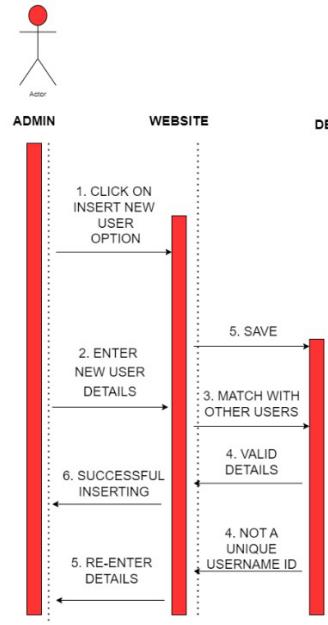


Figure 10: Sequence Diagram for INSERT NEW USER (ADMIN) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY ADMIN AND COD WILL BE PROVIDED WITH A PROMINENT OPTION OF SEARCH USER

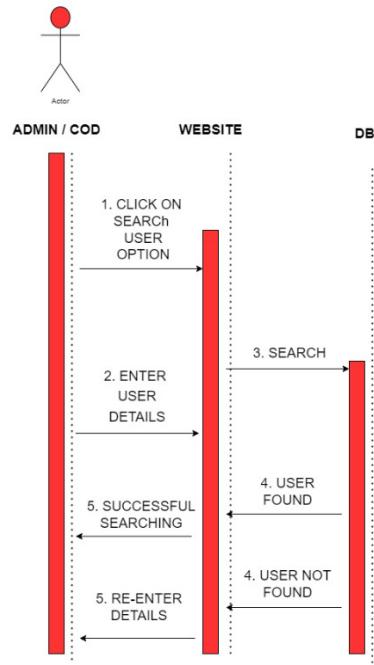


Figure 11: Sequence Diagram for SEARCH USER (ADMIN AND COD) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY ADMIN AND COD WILL BE PROVIDED
WITH A PROMINENT OPTION OF SEE USERS

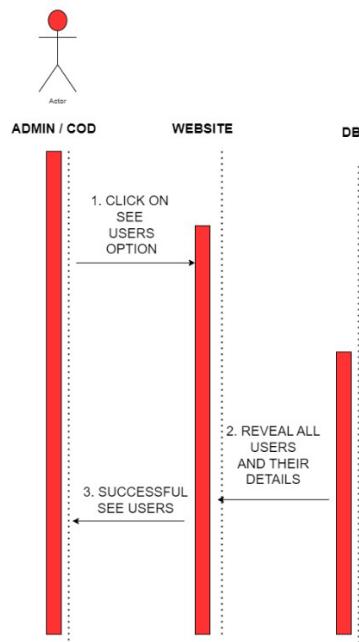


Figure 12: Sequence Diagram for SEE USERS (ADMIN AND COD) of E-Folder Application

AFTER LOGGING IN SUCCESSFULLY ALL USERS (ACTORS) WILL BE PROVIDED WITH A PROMINENT OPTION OF UPDATE DETAILS THROUGH WHICH THEY CAN UPDATE THEIR NAME, EMAIL, PASSWORD etc.

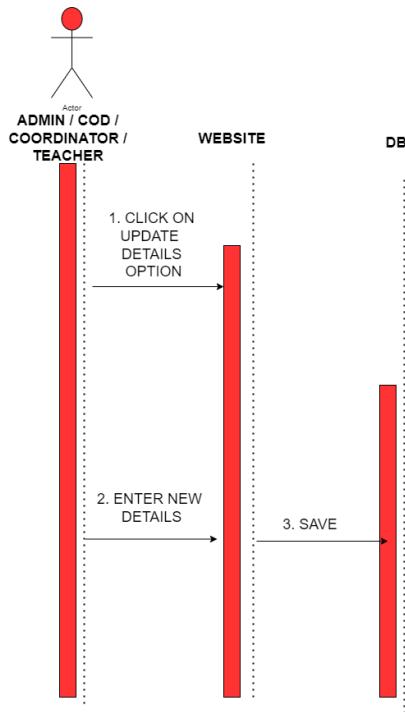


Figure 13: Sequence Diagram for UPDATE DETAILS (ALL ACTORS) of E-Folder Application

5.4 Collaboration Diagrams

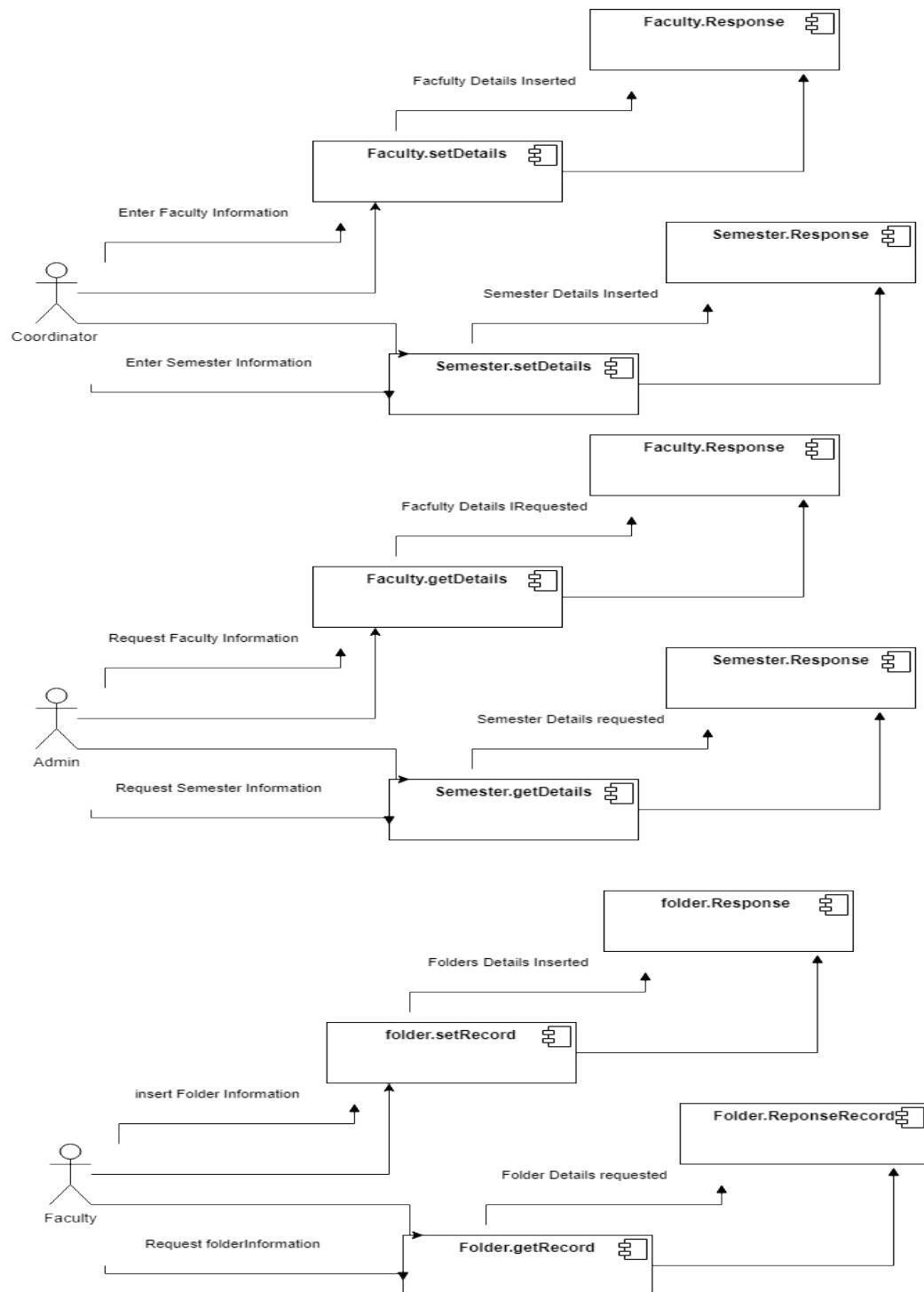


Figure 14: Collaboration Diagram of E-Folder Application

5.5 ERD

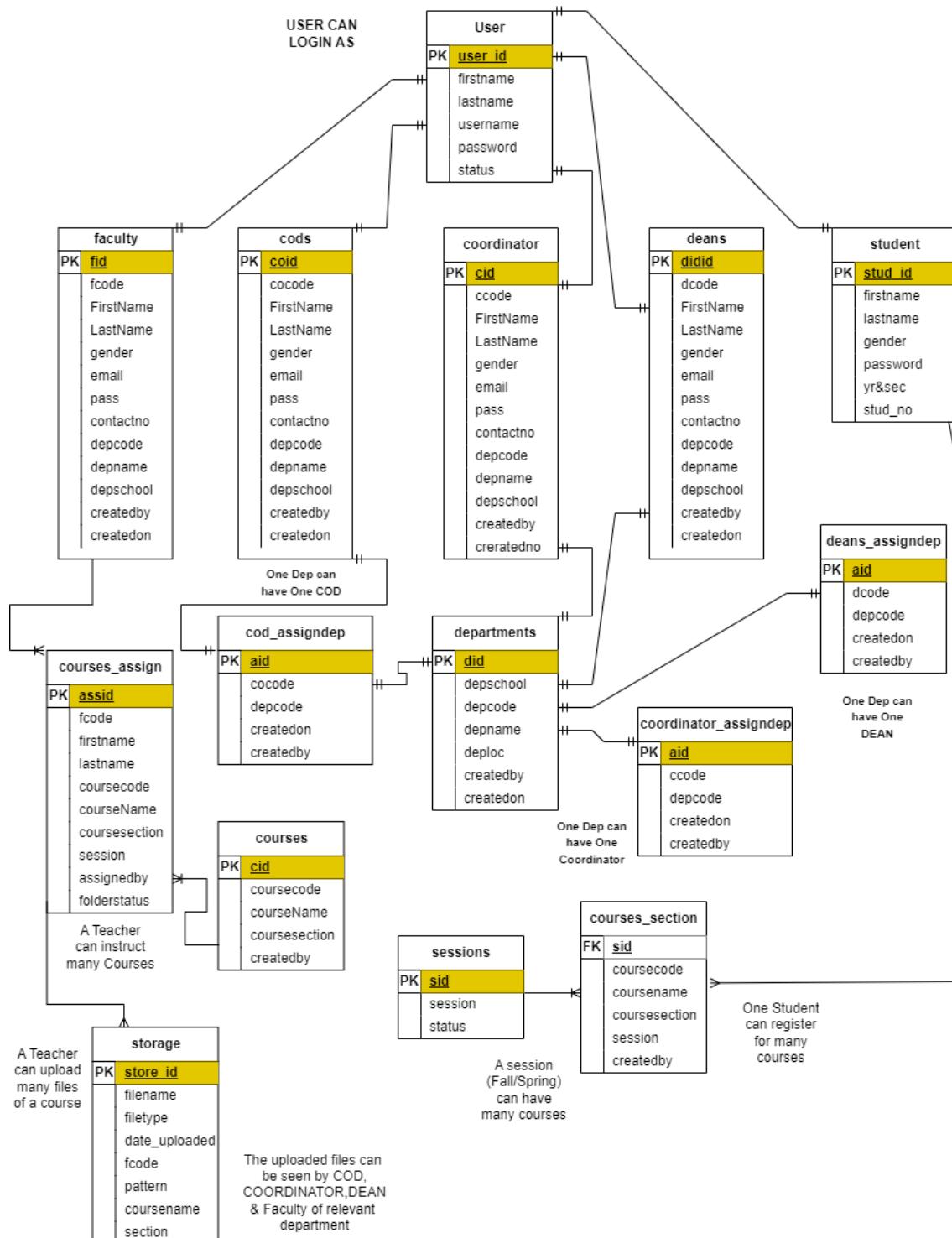


Figure 15: Entity Relation Diagram (ERD) of E-Folder Application

6. IMPLEMENTATION DETAILS

6.1 Development Setup

- **Programming Language:** PHP Core

PHP has been used to develop the front end. Moreover the library fpdf (free pdf) has been used in uploading the course content on E-folder and to assemble it regarding to the pattern provided.

- **For Writing & Testing Code:** Visual Code / Sublime Editor (Latest Version)

The code for this E-folder Generator Application has been written in the VS code and Sublime Editor.

- **Compiling & Debugging the Code:** Chrome's V8 Engine

For accomplishing the tasks in JavaScript (JS) V8 engine has been used.

- **Local Hosting Server:** Xampp Server

For backend and deployment on the local server of this application Xampp Server has been used.

- **Local Backend (Database) Support:** MySQL Server

For building a relational database as a backend MYSQL server has been used through XAMPP server using all the essentials and functional queries.

6.2 Deployment setup

As far as the matter of deployment is concerned, I humbly seek and ask a favor from UMT management to deploy this E-Folder Generator Application in the premises. I have done all this effort and hard-work for this FYP project in an order to present some honor to my institute, by providing a simple solution to a time taking and head aching problem of files management and expensive as well as environment-unfriendly printed papers.

6.3 Algorithms

The entire code has been written by keeping in mind the solution of the problem stated above. Simple PHP with JS has been used to produce a responsive front-end. On the

other hand for back-end databases MySQL and XAMPP has been used. Moreover, for uploading and viewing files being uploaded PHP library FreePDF (fpdf) has been used. Although it is a very simple project but it provides a solution to a attention-seeking problem.

6.4 Constraints

6.4.1 Assumptions

- No person can login without having proper authentication information as username and password.
- All the users will have their limited access according to their roles.
- The faculty members can give uploading pattern to the files that will be seen in the exact pattern in the pdf being downloaded.
- CODs, DEANs, Coordinators and faculty (teacher) can view the uploaded files as pdf related to the respective course section of a relevant department.
- This application will be very helpful for replacing hardcopy printed pages.
- This application can be deployed easily and can be understood by all the actors.
- The application will only work if all the roles are well-defined and each rolling member have unique characteristic information for login and other functions.

6.4.2 System constraints

- **Unique Authentication:**

For proper working and optimal solution all the defined roles above should be fulfilled. Each having proper and unique authenticating usernames and passwords and other vital information.

- **Document Pattern:**

Note that the files will be uploaded same as the uploading pattern provided by the faculty member (teacher) and will be shown in the exact same pattern. So, pattern should be correct. The teacher should know which page will come next after.

- **PDF:**

The document can only be downloaded and viewed in PDF format. So that no one can alter the document.

6.4.3 Restrictions

This E-folder generator Application gives each user restricted actions according to their roles. For Example only teacher has the right to upload a file so all other users are restricted from this functionality. Similarly, every role is well-defined and have proper restrictions according to their roles.

6.4.4 Limitations

This application is well-focused on the documents related to the relevant course. So, videos, audio files cannot be uploaded. This is because we want to make it simple and space efficient.

7. TESTING

7.1 Extended Test Cases of E-Folder Application

Table 3 Test-1 (Login)

Testing Login Button					
TC - Steps	TC - Steps	TC - data	TC - Expected Result	TC - Actual Result	Status (Pass/Fail)
1.	Go to the login section.				Pass
2.	Username and password are required.	Username: 2345 Password: 2345			
3.	Press the button login		User will be able to enter the system	User logged in successfully.	

Table 4 test-2 (Uploading)

Testing Uploading File					
TC - Step	TC - Steps	TC - data	TC - Expected Result	TC - Actual Result	Status (Pass/Fail)
1.	Select the option by clicking the button.				Pass
2.	Select the files From the PC in pattern.	Samplertext1.pdf Samplertext2.pdf			
3.	Click on the Upload button.		The Files should be uploaded to DB	The files uploaded successfully	

Table 5 Test-3 (Downloading)

Testing Downloading/Viewing File					
TC - Step	TC - Steps	TC - data	TC - Expected Result	TC - Actual Result	Status (Pass/Fail)
1.	Choose "view course content" from the menu.		The files should be downloaded in the same uploading pattern in pdf format.	The files get downloaded successfully.	Pass

7.2 Source Code

7.2.1 Code snippet

```
b.appendChild(c);
}
x[i].appendChild(b); a.addEventListener("click", function(e) {
/*when the select box is clicked, close any other select boxes, and open/close the current select box:*/
e.stopPropagation(); closeAllSelect(this);
this.nextSibling.classList.toggle("select-hide"); this.classList.toggle("select-arrow-active");
});
}
function closeAllSelect(elmnt) {
/*a function that will close all select boxes in the document, except the current select box:*/
var x, y, i, xl, yl, arrNo = [];
x = document.getElementsByClassName("select-items");
y = document.getElementsByClassName("select-selected"); xl = x.length;
yl = y.length;
for (i = 0; i < yl; i++) { if (elmnt == y[i]) { arrNo.push(i)
} else {
y[i].classList.remove("select-arrow-active");
}
}
}
for (i = 0; i < xl; i++) { if (arrNo.indexOf(i)) {
x[i].classList.add("select-hide");
}
}
}
/*
if the user clicks anywhere outside the select box, then close all select boxes:*/ document.addEventListener("click",
closeAllSelect);
</script>

</body></html>
```

```

/*style the arrow inside the select element:*/
.select-selected:after { position: absolute; content: "";
top: 14px; right: 10px; width: 0;
height: 0;
border: 6px solid transparent;
border-color: #fff transparent transparent transparent;
}

/*point the arrow upwards when the select box is open (active):*/
.select-selected.select-arrow-active:after {
border-color: transparent transparent #fff transparent; top: 7px;
}

/*style the items (options), including the selected item*/
.select-items div,.select-selected { color: #ffffff;
padding: 8px 16px;
border: 1px solid transparent;
border-color: transparent transparent rgba(0, 0, 0, 0.1) transparent; cursor: pointer;
user-select: none;
}

/*style items (options):*/
.select-items { position: absolute;
background-color: DodgerBlue; top: 100%;
left: 0;
right: 0;
z-index: 99;
<!DOCTYPE html>
<?php
require 'validator.php'; require_once 'conn.php'
?>
<html lang = "en">
<head>
<title>E-folder Application</title>
<meta charset = "utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel = "stylesheet" type = "text/css" href = "admin/css/bootstrap.css" />
<link rel = "stylesheet" type = "text/css" href = "admin/css/jquery.dataTables.css" />
<link rel = "stylesheet" type = "text/css" href = "admin/css/style.css" />
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
/*the container must be positioned relative:*/
.custom-select { position: relative; font-family: Arial;
}

.custom-select select {
display: none; /*hide original SELECT element:*/
}

.select-selected {
background-color: DodgerBlue;
}

```

```

Connection Code:
<?php
$conn = mysqli_connect("localhost", "clearmov_db_sfms", "03347908648@aA", "clearmov_db_sfms");

if(!$conn){
die("Error: Failed to connect to database!");
}
if(isset($_SESSION['faculty'])){
$default_query = mysqli_query($conn, "SELECT * FROM `faculty` where fcode='".$_SESSION['faculty']."'") or die(
    mysqli_error());
$check_default = mysqli_num_rows($default_query);
$row=$default_query->fetch_assoc(); if($check_default === 0){
$encrypted_password = md5('admin');
mysqli_query($conn, "INSERT INTO `user` VALUES('', 'Administrator', '', 'admin', '$encrypted_password', 'administrator')") or die(
    mysqli_error());
return false;
}
}?
}

Code for Faculty Interface:
<?php session_start();
?>

```

8. RESULTS/OUTPUT/STATISTICS

8.1 %completion

This application is 99.9% completed in an order to fulfil all the requirements. From logging in to the uploading of files and viewing those files each and every requirement and function works properly without any chaos and confusion.

8.2 %accuracy

After testing each and every function of this application stated above, we claim that the accuracy level of this system is 99.9%. Every user's as well as functionality's atomicity is fulfilled. This System fulfils all the ACID (Atomicity, Consistency, Isolation and Durability).

8.3 %correctness

After testing the whole system again and again we claim that all the requirements are fulfilled and the correctness level is 99.9%. No ambiguous and irrelevant interface shows up and proper tasks are given to our well-defined rolling members.

9. CONCLUSION

As we all know, nothing in this world is perfect and everything needs some improvement and updating after some specific course of time. But what we can do is that we should propose a currently best and optimal solution to over facing-problem. In my point of view, this E-Folder Generator Application fulfills this requirement of solving the problem stated above. Due to modernization our environment is becoming quite unhealthy. Even a small effort to make it healthy makes a big difference. Moreover, our lives are becoming so busy and tiring. By using technology, we should make it easy and relaxing not tiring. This application fulfils both these requirements by replacing the typical management of printed pages that is a head aching problem. Moreover, it will make our environment a little healthy by shutting down the printer that emits Carbon dioxide as well as Carbon Monoxide at each page it prints. This will also help to save electricity. Hence, according to me it is a very good way to adopt modernization for the sake of mankind.

Final Year Project Report

Nightales in UMT



Project Advisor:

Noaman Saleem

Submitted By:

Muhammad Arqam	F2019266040
Sumyya Anwar	F2019266048
Ahmad Afaaq	F2019266005
Aneeqa Riaz	F2019266009

Session

2019 - 2023

Degree

Bachelors of Science in Computer Science (BsCS)

School

School of Systems and Technology (SST)

University of Management and Technology

C-II Johar Town Lahore Pakistan

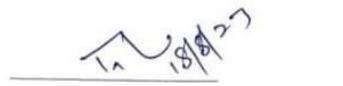
Dedication

At the intersection of technology and entertainment, a dedicated group of creators has embarked on a unique journey, one that is destined to captivate the hearts and minds of gamers, particularly those belonging to the esteemed community of UMTians. With an unwavering passion for gaming and an ardent love for their alma mater, this visionary team has set out to construct an experience like no other – a gaming masterpiece that resonates deeply with the denizens of UMT and all gamers alike. In a world where virtual landscapes have become the canvas for limitless possibilities, these creative minds have recognized a void that yearns to be filled. Amidst the sea of gaming options available, there lies a distinct absence – a void wherein the campus that houses cherished memories, friendships, and a treasure trove of learning experiences remains unexplored. This team has taken upon itself the monumental task of not just addressing this void, but of elevating it into a realm of infinite entertainment. The essence of their mission lies in the understanding that the University of Management and Technology (UMT) is more than just an educational institution; it is a crucible of dreams, a fountainhead of knowledge, and a sanctuary where aspirations take flight. With a profound sense of purpose, they are meticulously constructing a gaming universe that does justice to the grandeur of UMT. From the iconic buildings that have witnessed countless stories unfold to the tranquil pathways that have borne the footsteps of countless scholars, every corner of UMT's sprawling campus is being painstakingly recreated in the digital realm. However, it is not just about the buildings or the geography; it is about the narrative that weaves it all together. This is where their storytelling prowess comes into play. Every cobbled path, every lush green courtyard, and every bustling lecture hall will be imbued with a narrative that intertwines education and adventure. Players will not only get to explore the campus through the lens of a student, but also embark on quests and challenges that mirror the real-life academic journey. From solving puzzles that draw from the subjects taught within these hallowed walls to navigating the intricacies of university life, the game promises to offer an experience that is not just entertaining, but also enlightening. The dedication of this team to their mission is truly commendable. They recognize that their endeavor is not just about creating a game; it is about fostering a sense of belonging and nostalgia. It is about connecting past, present, and future UMTians through a shared virtual experience that resonates with their unique journey. By undertaking this ambitious project, they are bridging the gap between the tangible and the digital, reminding us all that the bonds formed within the physical confines of a university can transcend into the realm of imagination and pixels. In a world where technology is redefining the way we learn, communicate, and experience life, these visionary creators are carving a niche that celebrates both tradition and innovation. The game they are constructing is not just a testament to their technical prowess, but also a testament to their unwavering commitment to their community. Through their efforts, UMTians will not only be able to traverse the halls of their beloved institution virtually, but also share their stories, triumphs, and memories with a global community of gamers. In conclusion, the work undertaken by this dedicated team is nothing short of remarkable. By venturing into uncharted territory and combining the realms of education, nostalgia, and gaming, they are pioneering a new form of digital expression. Their commitment to UMT and the gaming community at large is a beacon of inspiration, displaying the potential of creativity and technology to bridge gaps and create connections. As their virtual masterpiece comes to life, it is bound to leave an indelible mark on the world of gaming and stand as a testament to the power of passionate visionaries.

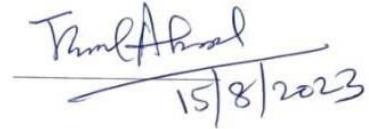
Final Approval

Final Approval

- **Head of Department**
Department of Computer Science
School of Systems & Technology
UMT Lahore



- **Program Director (Final Year Projects)**
Department of Computer Science.
School of Systems & Technology
UMT Lahore


15/8/2023

- **Supervisor**
Department of Computer Science.
School of Systems & Technology
UMT Lahore



Acknowledgment

In the grand tapestry of our journey, as we tread the path that blends innovation and imagination, our hearts are brimming with gratitude to the divine forces that have guided us. With reverence and humility, we extend our sincerest thanks to Allah, the Merciful, for bestowing upon us the wisdom, determination, and resources to embark on this remarkable endeavor. It is through His grace that we find ourselves united in purpose, driven by the shared vision of constructing a gaming experience that stands as a testament to both our creativity and our commitment to the values that define us. Beyond the realm of pixels and code, our journey has been intertwined with the unwavering support of our families and friends. To our parents, whose love and encouragement have been the bedrock of our pursuits, we owe an immeasurable debt of gratitude. Their unwavering belief in our abilities has been the wind beneath our wings, propelling us to reach for the stars. Our friends, too, have been our pillars of strength, standing by us through the highs and lows, celebrating the victories and offering solace in moments of challenge. As we reflect upon the intricate layers of our journey, the figure of our advisor, Sir Noaman Saleem, emerges as a guiding light. His vast reservoir of experience and knowledge has illuminated our path, steering us towards excellence with an unwavering commitment to high standards. It is under his mentorship that we have grown not only as developers but also as individuals striving to make a meaningful impact in the world of gaming. We are indebted to his dedication, patience, and belief in our potential. Additionally, we extend our gratitude to Sir Rana Marwat Hussain, who joined us as a co-advisor during the final stretch of this transformative journey. As we transitioned from students to scholars, his guidance became an invaluable compass, helping us navigate the intricate waters of academia. His wisdom and insights have enriched our perspectives, shaping us into well-rounded individuals prepared to make meaningful contributions. Our journey has been marked by the wisdom imparted by other esteemed advisors as well. To Sir Rehan Raza, Sir Owais Khan, and Sir Khawaja Ubaid-ur-Rehman, we extend our heartfelt appreciation for sharing their expertise and insights, providing us with the scaffolding upon which our project was built. Their dedication to our growth and success has been instrumental in shaping the contours of our achievement. In this symphony of gratitude, our alma mater, the University of Management and Technology, holds a special place. The platform provided by the Imaginary Cup 2022 has been a catalyst, propelling us forward and granting us the privilege to be a part of something larger than ourselves. The opportunities and resources bestowed upon us by the university have ignited our passion and empowered us to transform our dreams into reality. As our journey culminates in this endeavor, we are keenly aware that this tapestry of gratitude extends beyond these words. It is woven into every line of code, every moment of brainstorming, and every instance of perseverance in the face of challenges. It is a testament to the power of collaboration, mentorship, and shared dreams. With hearts full of appreciation and anticipation, we look forward to presenting the world with a gaming experience that stands as a tribute to the forces that have propelled us to this point.

Project Title: Nightales in UMT

Objective: To make a game base on the map of UMT to allow students to have a fun little experience of both horror and puzzles.

Undertaken by :

Muhammad Arqam	F2019266040
Sumyya Anwar	F2019266048
Ahmad Afaaq	F2019266005
Aneeqa Riaz	F2019266009

Supervised by:

Sir Noaman Saleem

Co-Supervised by:

Sir Rana Marwat Hussain

Starting Date:

November 20th, 2022

Completion Date:

July 18th, 2023

Tools Used:

- Visual Paradigm
- Blender
- Figma
- Unity
- Visual Studio
- Photoshop
- Google

Operating System:

- Windows 10/11
- Android 13

Plagiarism Report

University of Management and Technology, Lahore

Similarity Report

Turnitin Originality Report

Nightales in UMT by Muhammad Arqam, Sumyya Anwar, Ahmad Afaaq and Aneeqa Riaz

From Reference Desk (LRC info desk)

- Processed on 09-Aug-2023 16:15 PKT
- ID: 2143496991
- Word Count: 8729

Similarity Index

10%

Similarity by Source

Internet Sources:

7%

Publications:

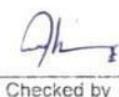
1%

Student Papers:

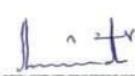
8%

Sources:

1. 2% match (student papers from 18-Aug-2019)
Submitted to Liberal Arts and Science Academy on 2019-08-18
2. 1% match (student papers from 06-Nov-2018)
Submitted to Higher Education Commission Pakistan on 2018-11-05



Checked by



Verified by CLO

Note:

- Sometimes the overall similarity index may be smaller than the repository percentages combined. This would be due to overlapping text within the repositories.
- It is a system generated report.

Figure 1: Plagiarism Report

Declaration Form
(To be submitted by FYP Advisor)

Declaration Form
(To be submitted by FYP Advisor)

I have carefully examined the documentation of the Final Year Project titled "Nightales in UMT"; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with user manual and installation guide.

FYP Advisor Name: Noaman Saleem

Signature: 

Date: 15/08/2023

Abstract

In the ever-evolving landscape of modern entertainment, the gaming market emerges as a colossal force, characterized by its exponential growth and boundless potential. With each sunrise, a new game blossoms, either ready to captivate audiences or steadily marching towards release, embodying the ceaseless creativity and innovation that define this dynamic realm. Games have transcended geographical boundaries, becoming a universal language spoken on mobile devices, consoles, computers, laptops, and gamepads across the globe. The plethora of gaming genres caters to diverse tastes, ranging from the whimsical allure of hyper-casual games to the immersive realism of 3D simulations, from the pulse-pounding thrill of action-packed adventures to the strategic engagement of trivia and puzzle challenges. This kaleidoscope of gaming experiences isn't just an industry; it's a cultural phenomenon that spans generations and backgrounds. A testament to its gravity is the list of luminaries who have invested prodigious resources in shaping the gaming market. Industry giants like Microsoft, Sony, Activision, Unreal Engine, Gameloft, and Unity stand as titans, striving to carve their niche as they push the boundaries of innovation and entertainment. The sheer magnitude of their investments underscores the industry's vast economic potential, an arena where creativity fuels economic growth and technological progress. In this pivotal 21st century, where smartphones have become extensions of our beings, the gaming market finds another avenue to flourish. The ubiquity of smartphones is a testament to how technology has woven itself into the fabric of our lives. Regardless of age, a smartphone is more than just a device; it's a gateway to memories, connections, and, yes, gaming. From the youngest to the eldest, people across generations find solace, joy, and excitement in their mobile games. This burgeoning demand for mobile gaming experiences fuels a cycle of innovation, pushing smartphone manufacturers to consistently improve technology to deliver immersive gaming experiences. As smartphones evolve, so do the expectations of users, who seek increasingly sophisticated games to fill their screens and hearts. This evolution is mirrored by the rise of mobile esports, where players compete in global tournaments, demonstrating their skills and winning substantial prizes. The competitive spirit that has long defined traditional sports is finding a new home in the digital arenas of mobile gaming. Yet, it's important to acknowledge that not every smartphone boasts cutting-edge capabilities. Here, the significance of accessible yet engaging games shines through. This is precisely where our proposed project finds its footing – a game designed not only to entertain but to ensure that every UMT student, regardless of their smartphone's specifications, can partake in the joy of gaming. Our aspiration to bring the UMT campus to life in a virtual gaming environment stems from a desire to enhance the connection students have with their educational haven. While the official UMT app offers an array of features, the absence of a game centered around the UMT map is conspicuous. Our aim isn't just to fill this void but to create a delightful, immersive experience for UMT students. We envision a game that offers a captivating adventure, intertwined with the familiar contours of the UMT campus. Through this gaming experience, we hope to foster a stronger bond between students and their academic environment, transforming navigation into exploration, and perhaps even learning. As we embark on this journey, we are attuned to the pulse of the gaming market, the sway of technological advancements, and the unique needs of the UMT community. Our vision extends beyond creating just another game; it's about sculpting an experience that resonates, an experience that weaves together nostalgia, playfulness, and a sense of belonging. In an era where the digital and the physical coalesce, our project strives to bridge these realms, allowing UMTians to traverse their campus with a new perspective, to venture beyond the mundane and discover a world that combines familiarity with the thrill of discovery. In the grand tapestry of gaming's evolution, our endeavor is but a humble thread, a contribution to a global phenomenon that unites people across borders and backgrounds. It's a testament to the potential of technology to cultivate joy, connection, and a sense of wonder. Through our project, we aim not only to tap into the immense potential of the gaming market but also to honor the spirit of exploration that defines both gaming and education.

REVISION CHART

Table 1: Revision Chart

Version	Primary Author(s)	Description of Version	Date Completed
Mid Term Report	Muhammad Arqam, Sumyya Anwar, Ahmad Afaaq, Aneeqa Riaz	Completed till 5 th Chapter	26 th November, 2022
Final Term Report	Muhammad Arqam, Sumyya Anwar, Ahmad Afaaq, Aneeqa Riaz	Completed	18 th July, 2023

CONTENTS

CONTENTS.....	IX
DEFINITIONS AND ACRONYMS.....	XI
LIST OF FIGURES.....	XII
LIST OF TABLES.....	XIII
1. GAME OVERVIEW	1
1.1 GAME SUMMARY	1
1.2 TARGET PLATFORM(S)	1
1.3 BUSINESS MODEL.....	1
1.4 THEME / SETTING / GENRE	1
1.5 CORE GAMEPLAY MECHANICS.....	1
1.6 SERVER / ONLINE MECHANICS.....	2
1.7 ART STYLE.....	2
1.8 LOOK & FEEL/ CAMERA.....	2
1.9 PROGRESSION.....	3
2. STORY AND GAMEPLAY.....	4
2.1 STORY.....	4
2.2 CORE GAMEPLAY.....	4
3. DOMAIN ANALYSIS	5
3.1 CUSTOMER.....	5
3.2 STAKEHOLDERS	5
3.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT.....	5
3.4 DEPENDENCIES/ EXTERNAL SYSTEMS	5
3.5 REFERENCES	5
3.5.1 <i>Related Projects</i>	6
3.5.2 <i>Feature Comparison</i>	6
4. REQUIREMENTS ANALYSIS	7
4.1 REQUIREMENTS.....	7
4.2 LIST OF ACTORS.....	8
4.3 LIST OF USE CASES	8
4.4 SYSTEM USE CASE DIAGRAM.....	9
4.5 EXTENDED USE CASES	10
5. USER INTERFACE / SCREENS.....	21
5.1 MAIN MENU	21
6. LEVELS	24
6.1 TOOL FOR LEVEL DESIGNING.....	24
6.2 FIRST TIME EXPERIENCE.....	25
7. ASSET LIST	26
8. IMPLEMENTATION DETAILS	28
8.1 DEVELOPMENT SETUP	28
8.2 DEPLOYMENT SETUP.....	28
8.3 ALGORITHMS.....	29
8.4 CONSTRAINTS.....	30

8.4.1	<i>Assumptions</i>	30
8.4.2	<i>System constraints</i>	30
8.4.3	<i>Restrictions</i>	31
8.4.4	<i>Limitations</i>	31
9.	TESTING.....	33
9.1	EXTENDED TEST CASES	33
9.2	TRACEABILITY MATRIX.....	44
9.2.1	<i>RID vs UCID (requirements vs use cases)</i>	44
9.2.2	<i>Prototypes (RID vs PID)</i>	44
9.2.3	<i>Test Cases (RID vs TID)</i>	45
9.2.4	<i>Coverage (UCID vs TID)</i>	46
10.	RESULTS/OUTPUT/STATISTICS.....	47
10.1	%COMPLETION	47
10.2	%ACCURACY	47
10.3	%CORRECTNESS	47
11.	CONCLUSION.....	48
12.	FUTURE WORK.....	49

Definitions and Acronyms

Table 2: Table of Definitions and Acronyms

Acronym	Definition
UMT	University of Management and Technology
POS	Point of Sale
POV	Point of View
FPP	First Person Perspective
TPP	Third Person Perspective

List of Figures

<i>Figure 1: Plagiarism Report</i>	v
<i>Figure 2: Reference Image from the game "Granny"</i>	2
<i>Figure 3: Game Cycle</i>	4
<i>Figure 4: Use case Diagram</i>	9
<i>Figure 5: Main Menu</i>	21
<i>Figure 6: Character</i>	22
<i>Figure 7: Ghost</i>	22
<i>Figure 8: Vehicle</i>	23
<i>Figure 9: Difficulty Graph</i>	24
<i>Figure 10: Code Snippet 1</i>	29
<i>Figure 11: Code Snippet 2</i>	29
<i>Figure 12: Code Snippet 3</i>	30

List of Tables

<i>Table 1: Revision Chart</i>	viii
<i>Table 2: Table of Definitions and Acronyms</i>	xi
<i>Table 3: List of Stakeholders</i>	5
<i>Table 4: Feature Comparison of your Game with existing games</i>	6
<i>Table 5: Function Categories</i>	7
<i>Table 6: Functional Requirements</i>	7
<i>Table 7: Non-Functional Requirements</i>	8
<i>Table 8: Data Requirements</i>	8
<i>Table 9: Extended Use Case UC-1: Login via Google Play</i>	10
<i>Table 10: Extended Use Case UC-2: Play the Game</i>	11
<i>Table 11: Extended Use Case UC-3: Save the game</i>	12
<i>Table 12: Extended Use Case UC-4: Exit the game</i>	13
<i>Table 13: Extended Use Case UC-5: Continue the game</i>	14
<i>Table 14: Extended Use Case UC-6: Settings</i>	15
<i>Table 15: Extended Use Case UC-7: Select Story</i>	16
<i>Table 16: Extended Use Case UC-8: Kill the enemies and complete quests</i>	17
<i>Table 17: Extended Use Case UC-9: Restart</i>	18
<i>Table 18: Extended Use Case UC-10: Return to Main Menu</i>	19
<i>Table 19: Extended Use Case UC-11: Feedback</i>	20
<i>Table 20: Login Button</i>	33
<i>Table 21: Play Button</i>	34
<i>Table 22: Save Button</i>	35
<i>Table 23: Exit Button</i>	36
<i>Table 24: Continue button</i>	37
<i>Table 25: Settings button</i>	38
<i>Table 26: Select Story button</i>	39
<i>Table 27: Kill the enemies and Complete Quests</i>	40
<i>Table 28: Restart Button</i>	41
<i>Table 29: Main Menu Button</i>	42
<i>Table 30: Feedback button</i>	43
<i>Table 31: RID vs UCID</i>	44
<i>Table 32: PID vs RID</i>	44
<i>Table 33: RID vs TID</i>	45
<i>Table 34: UCID vs TID</i>	46

1. GAME OVERVIEW

1.1 Game Summary

The game is based on the UMT Map. The main goal of the player is to retrieve his degree from the University at night while protecting himself from the ghost that wanders at night in UMT, and in order to protect himself he will have to use some items like Sticks, Guns, etc. He will have to complete side quests to get close to the retrieve the degree in the darkest hour.

1.2 Target Platform(s)

- Available on Android phone
 - ⇒ Mobile phone is a basic necessity now a day, everyone around us has a smart phone in their homes which they use to communicate with other people around the world, talk to each other via sending texts or entertain themselves by either listening to music, playing games or taking pictures of themselves or the things or places around them.
 - ⇒ The proposed game that we are working on is available on Android smart phones only.
- Available on Windows
- The game will be in Landscape mode.
 - ⇒ The scenery of the game will be in landscape mode where user will have a full on experience of the UMT Map.

1.3 Business Model

This will be a free to play game with no in-app purchases or side ads. The game is free to from Google playstore. The main goal of the game is give the users a hand on experience of UMT Map in the palm of their hands.

1.4 Theme / Setting / Genre

Following are few examples:

- GTA
- Farcry
- Red Redemption
- Granny
- Residence of Evil
- Slendergirl must die

1.5 Core Gameplay Mechanics

Controls

- **Single Touch:**
 - ⇒ To grab the items
- **Joystick:**
 - ⇒ For character movement

- **Tap:**
⇒ To select options
- **Release:**
⇒ To release item from inventory
- **Drag:**
⇒ To move item in inventory

For Each checked item, please provide an explanation

1.6 Server / Online Mechanics

- **Account with Cloud Save:** The game data will be saved in players Google play account.

1.7 Art Style

Which Art style you will use for your game?

- 3D ✓
- High Poly ✓

1.8 Look & Feel/ Camera

- First Person ✓

The camera will be fixed, the player will be able to see his hands and the item he is holding, his legs, his body but not his face. The camera position will be near his face. For example:

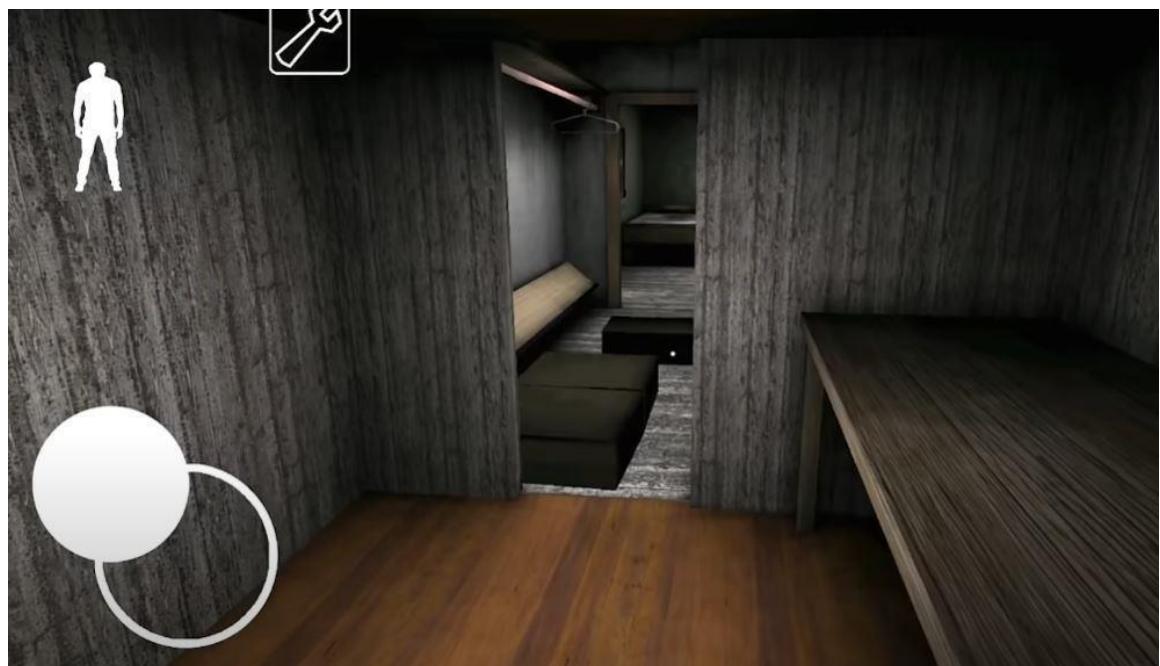


Figure 2: Reference Image from the game "Granny"

1.9 Progression

- **% Complete Based:** The game will progress when the user completes the side quests in order to reach the final task and complete the game before the given time. Once the time is over the play will have to restart from a specific point where they saved the game last time.

2. STORY AND GAMEPLAY

2.1 Story

A UMT Student who is tired of studying and coming to university every day, now he is trying his level best to get out of the university at all cost possible. He is very naughty, and a back bencher. All he does is miss his classes and do pranks on his teachers. He is just trying to get his degree and get out of the university. So, he decided to go on a mission and retrieve his degree from the UMT locker. He knew that it would not be possible to get the degree in day light because it is highly secured and he will be caught red handed easily. He decided to make a plan and drew out a map of UMT and he said he will retrieve the degree at night, but little did he know at night the university is a party place for the ghost and he is going to invade their party and they don't like it when someone interrupts them. He manages to get into the university but once he gets in there is no going out. He decided that he will follow the map and get his degree and get out of the university safely and also have to protect himself from the ghost attacks by using different items stick, gun etc., but he has to do it before sun rise otherwise, he will get caught.

2.2 Core Gameplay

The game loop is determined as the user will start the game, select play from the main menu, complete side quest in limited time if user is unable to complete the mission in limited time, he can either restart or close the game. If user completes the missions, he can save progress and close the game. After closing the user can start the game again.

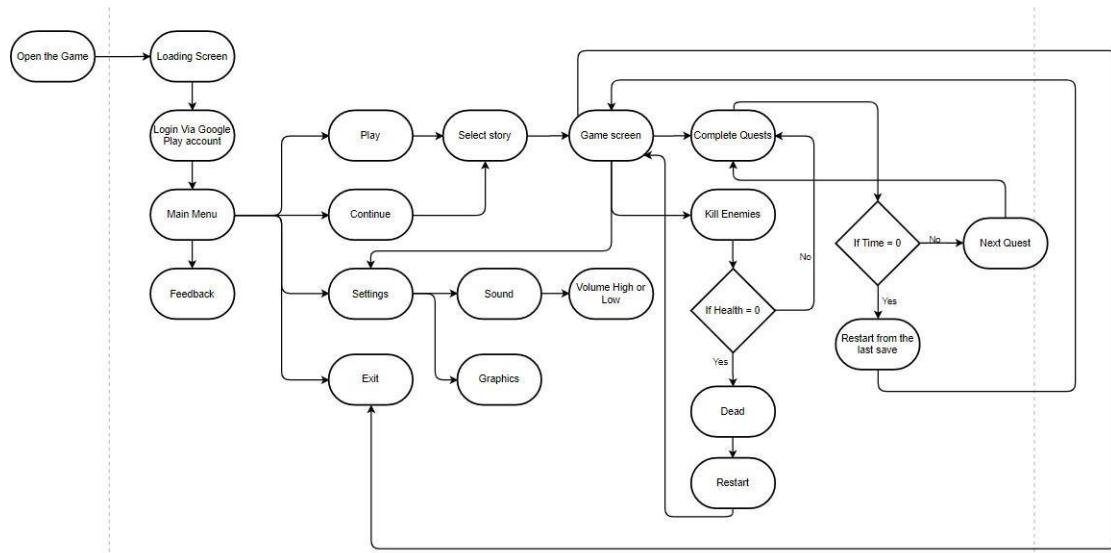


Figure 3: Game Cycle

3. DOMAIN ANALYSIS

3.1 Customer

We are not collaborating with anyone or working with anyone.

3.2 Stakeholders

Table 3: List of Stakeholders

Stakeholder	Role in System
Model Designer	Will create the models of the UMT building
Coder	Will code for the character model i.e., its movement, the grab, drag and drop, etc.
Background Maker	The 2D background that will cover the whole map.
Users	The users will be able to play the game.

Table 3.2 shows the list of stakeholder involved in Nightales in UMT.

3.3 Affected Groups with social or economic impact

The groups that will be impacted by the deployment of this system. They are listed as below:

e.g.

- Development team: Reduced paper work for the team. Also, we will be able to improve the game mechanics as we will look the feedback from users given through the Google playstore.
- Players: The purpose of this project is to target the university audience (especially UMTians) to have an experience of UMT Map model in the palm of their hand with a bit of horror and quest.

3.4 Dependencies/ External Systems

The project is dependent on these systems and products for its completion.

e.g.

- Android Device
- Windows 10/11
- Unity Engine
- Visual Studio code
- Blender
- Photoshop
- Google
- Google play store
- Wikipedia
- Youtube

3.5 References

- GTA 5: [Grand Theft Auto V - Rockstar Games.html](#)

- Granny: [Granny- Apps on Google Play.html](#)

3.5.1 Related Projects

In order to develop First Person Shooter quest and horror style game, we looked up several similar systems. Their details are given below

1. **GTA 5:** The story of three criminals as they have committed daring and profitable heists across the city of Los Santos.
2. **Granny:** This game have feature that has an unnamed protagonist trapped in a house, he has to solve puzzles while avoiding the scary Granny ghost to get out of the house in a time period of only one week.

3.5.2 Feature Comparison

Table 4: Feature Comparison of your Game with existing games

Sr No.	Comparison Feature	GTA 5	Granny	Nightales in UMT
1	Storyline	Has a complete story line	Does not have good story line	Have a decent storyline
2	Graphics	Have stunning graphics	Have stunning graphics	Decent graphics
3	Controls	Great controls via Mouse and Keyboard, or controller.	Does not have much control option	Have control option where user can change the sensitivity and everything
4	Map	Based on the map of South California	Does not have a dedicated map	Based on UMT map and buildings

In above Table 3.2 we have compared our game with existing games. This feature comparison shows our game includes new features.

4. REQUIREMENTS ANALYSIS

4.1 Requirements

Following are the Requirements of the purposed project:

Function Categories

Table 5: Function Categories

Functional Requirements	The user will be able to play the game via Android
Non-Functional Requirements	The functionalities that have no effect on the user's when they play the game
Data Requirements	How the user will save his game progress

Functional Requirements:

Table 6: Functional Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
FR1.1	Login via Google Play	Functional	User response	The system shall allow the users to login into the game via their google play account to save their progress.
FR1.2	Play the game.	Functional	User response	The user shall be able to play the game after connecting his Google play account with the game.
FR1.3	Continue the game	Functional	User response	The user shall be able to continue the game from where he left it.
FR1.4	Save progress	Functional	User response	The user shall be able to save their progress in the game.
FR1.5	Exit the game	Functional	User response	The user shall be able to close the game after playing it.

Non-Functional Requirements:

Table 7: Non-Functional Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
NFR1.1	Loading screen time	Non-functional	System Response time	The system shall take 5 to 10 seconds to load the screen, and it can also depends on the mobile specs.
NFR1.2	Saving time	Non-functional	System Response time	The system shall save the players progress via Google play account in cloud, and it can take a couple of seconds i.e., 10 to 15 seconds.
NFR1.3	Pick up items	Non-functional	User level constraint	The system shall not allow the user to equip more than 10 items at a time.

Data Requirements:

Table 8: Data Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
DR1	Data saving	Data requirement	System Response	The game data can be saved on the user Google playstore cloud, so that the user can play the game after closing and reopening it again.

4.2 List of Actors

Following are the actors involved in this project:

- **Gamers:** The gamer will be able to play the game anytime they want to on their mobile devices and save the progress via their Google play accounts.
- **Developers:** The developers will be able to view the feedback from the users given through via Google play store

4.3 List of use cases

List of use cases in this project is given below:

- **UC-1: Login via Google Play:** The user shall be able to login via Google Play account.
- **UC-2: Play the game:** The user shall be able to play the game after successfully connecting the game with his Google play account.
- **UC-3: Save the game:** The system shall allow the users to save their game progress in the Google play account.
- **UC-4: Exit the game:** The system shall allow the users to exit the game.
- **UC-5: Continue the game:** The system shall allow the users to continue the game where they left it.
- **UC-6: Settings:** The user shall be able to change the setting in the game according to their needs i.e., sound, graphics, etc.

- **UC-7: Select story:** The user shall be able to select the story mode from where they want to begin the game.
- **UC-8: Kill the enemies and Complete quests:** The system shall allow the users to kill the enemies using weapons i.e., Guns, Throw-able, etc.
- **UC-9: Restart:** The user shall be able to restart from a certain point where they died.
- **UC-10: Return to Main Menu:** The user shall be able to return to main menu.
- **UC-11: Feedback:** The system shall allow the users to give feedback via Google play store i.e., ratings and reviews.

4.4 System Use Case diagram

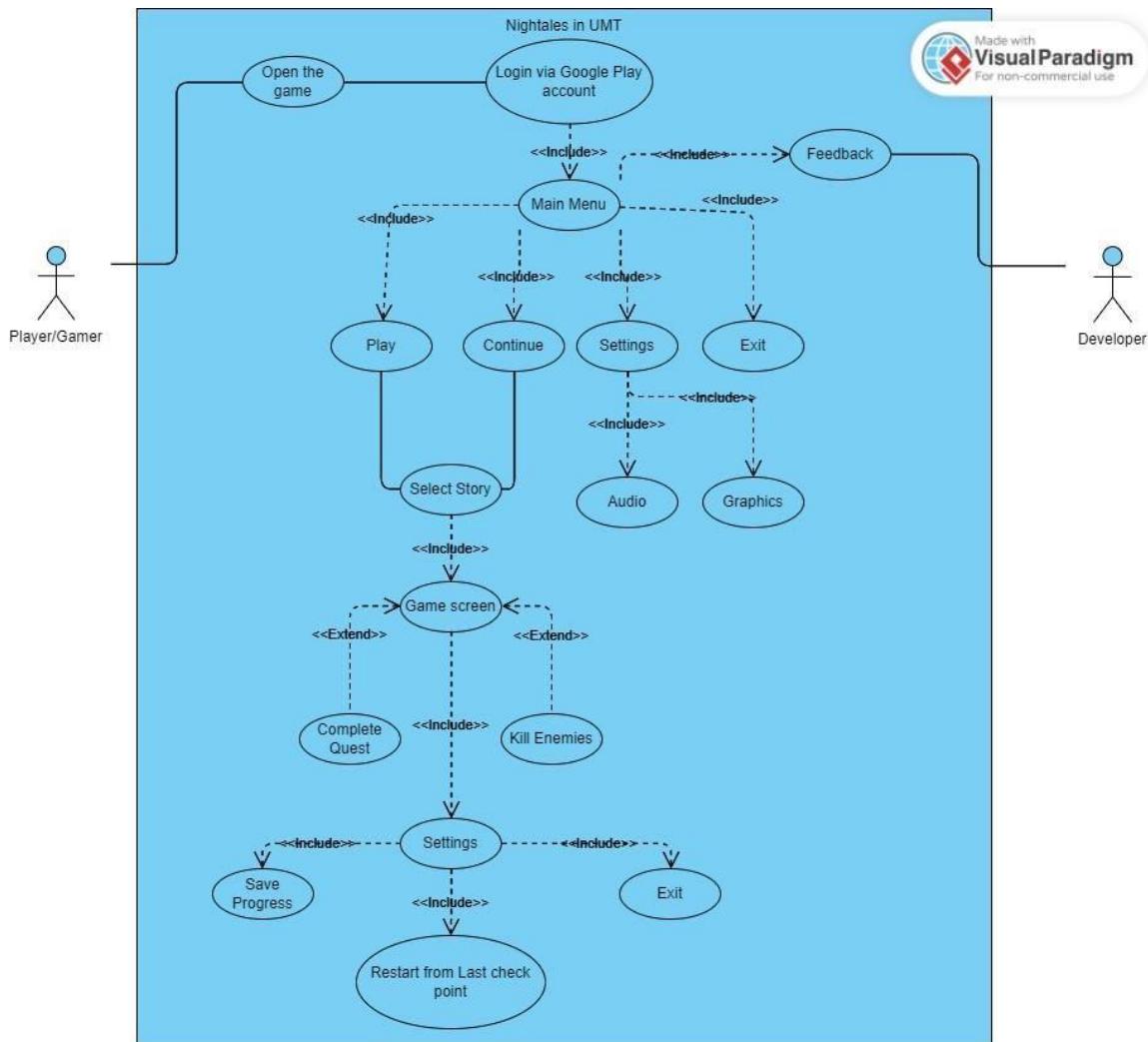


Figure 4: Use case Diagram

In above Figure 4.4.1 we have a gamer who can play the game after logging in and can save the progress in his google play account and can exit it afterwards.

4.5 Extended use cases

Table 9: Extended Use Case UC-1: Login via Google Play

Use Case ID:	UC-1		
Use Case Name:	Login via Google play		
Created By:	Muhammad Arqam, Sumyya Anwar	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The gamer will be able to login in the game via using Google play account what is linked to his/her email.		
Trigger:	The user will have his Google play account linked with the game.		
Preconditions:	Following are the pre-conditions: 1. User will have to need a Gmail account to have access to Google play.		
Post conditions:	Following are the post conditions: 1. User will be able to play the game after linking the game with his/her Google play account		
Normal Flow:	Here's the normal flow: 1. Create a Gmail account 2. Link the Gmail account with Google Playstore 3. Link the Google Playstore account with Google Play 4. Open the game 5. Link the Google play account with the Game 6. Play the Game		
Alternative Flows:	Here's the alternate flow: 5.a. 1: In step 5 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 5.e. 1: In step 5 of normal flow: 1. There is no Internet connection. 2. Unable to connect to Gmail account		
Includes:	UC-2: Play the game, UC-3: Save the game, UC-4: Exit the game, UC-5: Continues the game, UC-7: Settings		
Frequency of Use:	When the game is installed.		
Special Requirements:	Creating a Gmail account.		
Assumptions:	Better understanding of creating a Gmail account.		
Notes and Issues:	Creating a Gmail account, linking it with Google Playstore.		

Table 10: Extended Use Case UC-2: Play the Game

Use Case ID:	UC-2		
Use Case Name:	Play the game		
Created By:	Ahmad Afaaq, Aneeqa Riaz	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The gamer will be able to play the game, and he will also be able to kill the enemies and complete quests.		
Trigger:	The user will be able to play the game when he taps on “Play” in the Main Menu.		
Preconditions:	Following are the pre-conditions: 1. User will have to login with his Google play account		
Post conditions:	Following are the post conditions: 1. User will be able to play the game after linking the game with his/her Google play account		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Play in the Main Menu		
Alternative Flows:	Here's the alternate flow: 5a. In step 5 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 5a.e.1: In step 5 of normal flow: 1. There is no Internet connection.		
Includes:	UC-3: Save the game, UC-6: Select story		
Frequency of Use:	When the user wants to play the game.		
Special Requirements:	Must have Google play account to link the game with it.		
Assumptions:	Understands English		
Notes and Issues:	Network issues.		

Table 11: Extended Use Case UC-3: Save the game

Use Case ID:	UC-3		
Use Case Name:	Save the game		
Created By:	Muhammad Arqam	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to save their progress after they've played the game for several moments		
Trigger:	The user will be able to save the game when he clicks the "Save" button from settings		
Preconditions:	Following are the pre-conditions: 1. User will have to login with his Google play account		
Post conditions:	Following are the post conditions: 2. User will be able to play the game after linking the game with his/her Google play account and can save the game afterwards		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Play in the Main Menu 4. Save the game from the settings, or the game will be auto saved after every quest is completed		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e.1: In step 2 of normal flow: 2. There is no Internet connection.		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user wants to save the game.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 12: Extended Use Case UC-4: Exit the game

Use Case ID:	UC-4		
Use Case Name:	Exit the game		
Created By:	Sumyya Anwar	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to exit the game after playing it.		
Trigger:	The user will be able to exit the game when he clicks the “Exit” button from Main menu		
Preconditions:	Following are the pre-conditions: 1. User will have to open the game first in order to exit.		
Post conditions:	Following are the post conditions: 1. User can reopen the game whenever they feels like it.		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Play the game. 4. Save the progress. 5. Click on Exit in the Main Menu		
Alternative Flows:	There is no Alternate flow.		
Exceptions:	There are no Exceptions.		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user wants to close the game.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 13: Extended Use Case UC-5: Continue the game

Use Case ID:	UC-5		
Use Case Name:	Continue the game		
Created By:	Ahmad Afaaq	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to continue the game where they've left it from.		
Trigger:	The user will be able to continue the game when he clicks the “Continue” button from Main Menu.		
Preconditions:	Following are the pre-conditions: 1. User should have at least played the game once.		
Post conditions:	Following are the post conditions: 1. User will be able to play the game where they've left it (last saved)		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on “Continue” in the Main Menu 4. Start playing the game		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e. 1: In step 2 of normal flow: 1. There is no Internet connection.		
Includes:	UC-3: Save the game, UC-6: Select the Story		
Frequency of Use:	When the user wants to continue the game from where he/she left.		
Special Requirements:	Must have played the game at least once.		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 14: Extended Use Case UC-6: Settings

Use Case ID:	UC-6		
Use Case Name:	Settings		
Created By:	Aneeqa Riaz	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to change the settings like Volume control and Graphics		
Trigger:	The user will be able to change the game settings when he clicks the “Settings” button from Main Menu and In-Game screen as well		
Preconditions:	Following are the pre-conditions: 1. Before changing the setting user will have to open the game.		
Post conditions:	Following are the post conditions: 1. User games setting will be applied after he has changed the settings that he wants.		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Setting in the Main Menu 4. Save settings.		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e. 1: In step 2 of normal flow: 1. There is no Internet connection.		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user wants to adjust the graphics and audio settings		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 15: Extended Use Case UC-7: Select Story

Use Case ID:	UC-7		
Use Case Name:	Select Story		
Created By:	Sumyya Anwar, Muhammad Arqam	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to select the story from where they've left the game i.e., last saved.		
Trigger:	The user can select the story when they click on either “Play” or “Continue” button from the Main menu.		
Preconditions:	Following are the pre-conditions: 1. The user should have a Google Play account in order to select the story.		
Post conditions:	Following are the post conditions: 1. The user will play be able to continue the game from where they've left it.		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Play or Continue in the Main Menu 4. Select Story		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e. 1: In step 2 of normal flow: 1. There is no Internet connection.		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user wants to select what part of the game they want to play.		
Special Requirements:	Must save the game after every quest is complete or it can be auto saved		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 16: Extended Use Case UC-8: Kill the enemies and complete quests

Use Case ID:	UC-8		
Use Case Name:	Kill the enemies and Complete quests		
Created By:	Muhammad Arqam, Sumyya Anwar	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user will be able to kill the enemies and complete quests in order to finish the game in given time.		
Trigger:	The user will be able to kill enemies and complete quests when they click on the “Play” button from the Main menu.		
Preconditions:	Following are the pre-conditions: 1. User should first click on the “Play” or “Continue” Button		
Post conditions:	Following are the post conditions: 1. User can kill enemies and complete quests		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Play or Continue button from the Main menu. 4. Kill enemies and Complete Quests. 5. Save the game. 6. Exit.		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e. 1: In step 2 of normal flow: 1. There is no Internet connection.		
Includes:	No Use cases are involved.		
Frequency of Use:	Kill enemies and complete quests to finish the game and reach the final goal.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 17: Extended Use Case UC-9: Restart

Use Case ID:	UC-9		
Use Case Name:	Restart		
Created By:	Muhammad Arqam, Sumyya Anwar	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	If the user is unable to kill the enemies and dies or is unable to complete the quest with in the given time the player will die and will have to restart.		
Trigger:	The user clicks on the “Restart” button when they die due to enemy’s attacks or unable to complete the quest within the given time.		
Preconditions:	Following are the pre-conditions: 1. The user should complete the quests and kill enemies.		
Post conditions:	Following are the post conditions: 1. Restart the level from the last saved checkpoint and continue playing if the player dies.		
Normal Flow:	Here's the normal flow: 1. Open the game 2. Login via Google Play account 3. Click on Play or Continue button from the Main menu. 4. Kill enemies and Complete Quests. 5. Dies due to enemy’s attacks or fails to complete a quest within a given time. 6. Restart the game from the last saved check point. 7. Save the game. 8. Exit.		
Alternative Flows:	Here's the alternate flow: 2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account. 1. Can play the game offline.		
Exceptions:	Here's the Exception: 2.e. 1: In step 2 of normal flow: 1. There is no Internet connection.		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user is not able to complete the quest in given time or dies to an enemy.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 18: Extended Use Case UC-10: Return to Main Menu

Use Case ID:	UC-10		
Use Case Name:	Return to Main Menu		
Created By:	Ahmad Afaaq, Aneeqa Riaz	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer		
Description:	The user can return to the main menu while playing the game.		
Trigger:	The user when clicks the “Settings” button from the in-game screen while playing the game, they can click on “Return to Main Menu” from the settings.		
Preconditions:	<p>Following are the pre-conditions:</p> <ol style="list-style-type: none"> 1. The user should click on “Play” or “Continue” from the Main menu and be in the game. 		
Post conditions:	<p>Following are the post conditions:</p> <ol style="list-style-type: none"> 1. Can return to the main menu and again play the game if they want to. 		
Normal Flow:	<p>Here's the normal flow:</p> <ol style="list-style-type: none"> 1. Open the game 2. Login via Google Play account 3. Click on Play or Continue button from the Main menu. 4. Click on the settings button from the in-game screen. 5. Click on Return to Main menu. 		
Alternative Flows:	<p>Here's the alternate flow:</p> <p>2.a. 1: In step 2 of the normal flow, if the game is unable to link with the Google Play account.</p> <ol style="list-style-type: none"> 1. Can play the game offline. 		
Exceptions:	<p>Here's the Exception:</p> <p>2.e. 1: In step 2 of normal flow:</p> <ol style="list-style-type: none"> 1. There is no Internet connection. 		
Includes:	No Use cases are involved.		
Frequency of Use:	When the user is done playing the game.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

Table 19: Extended Use Case UC-11: Feedback

Use Case ID:	UC-11		
Use Case Name:	Feedback		
Created By:	Ahmad Afaaq, Aneeqa Riaz	Last Updated By:	Muhammad Arqam
Date Created:	26 th November, 2022	Last Revision Date:	9 th February, 2023
Actors:	Gamer, Developers		
Description:	The user will be able to give feedback to the developers from Google playstore by giving Ratings and Reviews		
Trigger:	The user can give ratings to the developers when the click on the “Feedback” button from the Main menu.		
Preconditions:	<p>Following are the pre-conditions:</p> <ol style="list-style-type: none"> 1. The user should open the game and must have Gmail account linked to Google Play store. 		
Post conditions:	<p>Following are the post conditions:</p> <ol style="list-style-type: none"> 1. Leave a rating and review for the Developer to see so they can improve the game. 		
Normal Flow:	<p>Here's the normal flow:</p> <ol style="list-style-type: none"> 1. Open the game 2. Login via Google Play account 3. Click on the Feedback button from the Main menu. 4. Leave ratings and reviews on Google playstore. 		
Alternative Flows:	<p>Here's the alternate flow:</p> <p>1.a. 1: In step 1 of the normal flow, if the game is unable to link with the Google Play account.</p> <ol style="list-style-type: none"> 1. The user can go to Google playstore and search the game there and give ratings from there directly. 		
Exceptions:	<p>Here's the Exception:</p> <p>1.e: In steps of normal flow:</p> <ol style="list-style-type: none"> 1. There is no Internet connection. 		
Includes:	No Use cases are involved.		
Frequency of Use:	When there is a bug or issue and want to point them out, or user is being generous to leave remarks about the game.		
Special Requirements:	No Special Requirements		
Assumptions:	No Assumptions		
Notes and Issues:	No Notes and Issues		

5. USER INTERFACE / SCREENS

5.1 Main Menu

- “Play” Button - Goes to Select story Screen
- “Options” Button – Goes to Select Character screen
- “Quit”

Here's a figure of the main menu:



Figure 5: Main Menu



Figure 6: Character



Figure 7: Ghost



Figure 8: Vehicle

6. LEVELS

In this game, you'll have to defend yourself and survive from the attacks of ghosts and complete several quests along the way in order to win the game. The more you progress in the game the more you the tasks will get harder and so on.

Beginner Task:

Task: Get a Gun

Objective: The objective of this task is that the player should get a weapon from a room in order to kill the ghost and protect himself/herself

Description: The beginning will be fairly easy as it will require the gamer to find a gun and protect himself from the attacks of the ghosts. For this the user will have to move himself/herself using the arrow keys and find the room where the gun is located.

Complex Task:

Task: Kill 10 Ghosts

Objective: The objective of this task is that the player should kill 10 ghosts.

Description: This task is fairly tough as the player will have to not only protect himself from the ghosts attack but also survive and kill 10 ghosts to protect himself/herself.

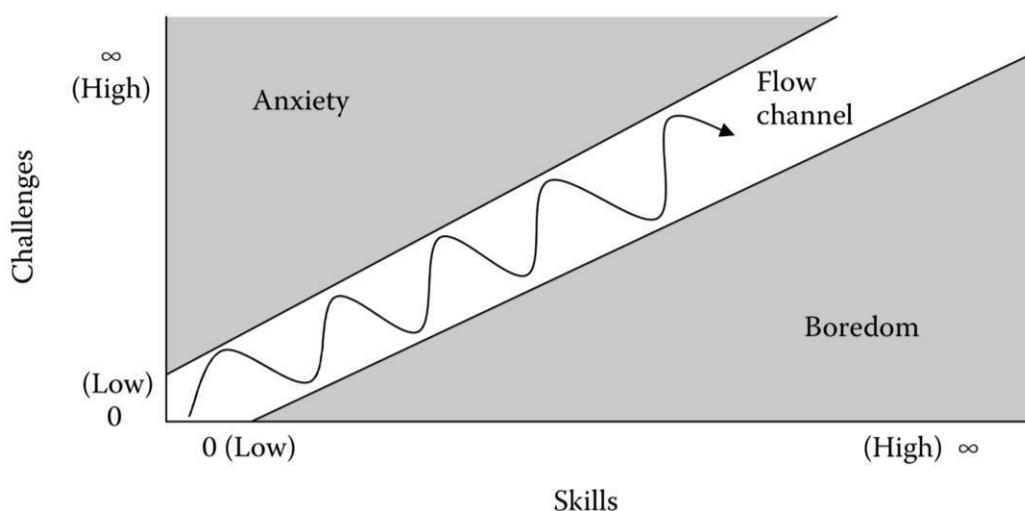


Figure 9: Difficulty Graph

6.1 Tool for Level Designing

There are several tools used for level designing in Nightales in UMT game.

Unity:

Unity is a popular game development engine that provides a comprehensive set of tools for designing levels. It offers a visual editor with drag-and-drop functionality, allowing you to create

and modify the game environment easily. Unity also supports scripting, which enables you to add interactive elements and implement gameplay mechanics.

Photoshop:

Textures and Materials: Photoshop can be used to create custom textures and materials for the surfaces in your game levels. You can design and edit images for ground, walls, objects, and other elements that make up the environment.

2D Level Layouts: You can use Photoshop to sketch and create 2D level layouts or maps. It allows you to create visually appealing diagrams and plan the overall structure of your levels.

Figma:

UI Elements and HUD Design: Figma excels in creating Player interfaces (UI) and designing HUD (Heads-Up Display) elements for your game. You can design buttons, icons, menus, and other interactive elements that enhance the Player experience in your levels.

Wireframes and Prototyping: Figma design and prototyping features can help you create wireframes and interactive prototypes of your levels. This can be useful for testing and iterating on level layouts and gameplay mechanics.

Blender:

Blender is a free and open-source 3D modeling and animation software. While it is primarily used for asset creation, it also offers level design capabilities. You can use Blender to create and manipulate 3D models, design terrains, and create intricate level layouts.

6.2 First Time Experience

The first time experience for the gamer is that they will be able to explore a somewhat similar map to UMT, where they have to survive from the attacks of the ghosts and complete the task in order to win and complete the game. As a first time experiencer of the game the game will be somewhat of medium difficult for new players and they'll get used to it as the task passes by. This game will give the players a sense of playing in the UMT map and survival game.

7. ASSET LIST

- Art
 - 3D Game
 - Player
 - The player will be a first person character.
 - Ghosts
 - The ghosts will be of different types that will attack the player.
 - Gun
 - Different varieties of guns for the player to use to kill the ghosts.
 - Environment
 - Use of textures to depict surrounding elements like trees, foliage, or bushes.
 - Incorporate textures for skyboxes or backgrounds to create visually appealing atmosphere.
 - Signage and Markings:
 - Implement textures for gun, ray tracing, and proper help guide for ease of player.
 - Lightning and Shadow textures
 - Use textures to enhance lighting effects, such as sunlight or artificial light sources.
 - Apply textures for shadows to create depth and realism within the game environment.
 - Player Interface Textures
 - Design textures for menus, buttons, and other UI elements.
 - Ensure consistency with the overall visual style of the game.
- Environment Art
 - Blood animation
 - Kill Animation
 - Item picking Animation
- UI - Icons, Buttons, Pop Ups
 - Main Menu
 - Play Button
 - Exit Button
 - Settings Button
 - Continue Button
 - Feedback Button
- Game Objects
 - Buildings
 - UMT like Map
 - Characters
 - A first person character

- Weapons
 - Pistol
 - ARs
 - SMGs
- Tools
 - Flashlight
 - Keys
- Particle FX
 - Flash light
 - Sky light
- Sound Effects
 - Damage taken sound
 - Shooting sound
 - Enemies dying sound
 - Door sounds
 - Horror sounds
- Music
 - Background Music
- All Writing
 - Quest Scripts
 - Story ‘Screenplay’
- Level Design Documents

8. IMPLEMENTATION DETAILS

8.1 Development Setup

Unity Engine (for designing front end): Unity empowers users with the capability to design games and immersive encounters in both 2D and 3D dimensions. The engine provides a core scripting API primarily utilizing C# through Mono. This API serves both the Unity editor via plugins and the actual games, complemented by intuitive drag-and-drop functionalities.

Visual Studio (for writing scripts in C#): Visual Studio empowers developers to seamlessly manage the entire development cycle within a single integrated environment. For instance, you can perform tasks such as editing, debugging, testing, version control, and cloud deployment all within Visual Studio. The rich array of features and programming languages available in Visual Studio facilitates a progressive transition from initial coding experiences to working on diverse project types.

Blender (for making models): Blender, often stylized as "blender," stands as a no-cost and open-source collection of software tools for 3D computer graphics. Its applications span from crafting animated films, visual effects, and artistic creations to producing 3D-printed models, motion graphics, interactive 3D apps, virtual reality content, and, in the past, even video games.

Adobe Photoshop (for Editing): Adobe Photoshop serves as a widely utilized software renowned for its capabilities in raster image editing, graphic design, and digital art creation. Through its innovative layering system, it enables a multi-dimensional and adaptable approach to design and editing. Additionally, the software offers a robust selection of editing tools that, when combined, grant the potential for a vast array of creative possibilities.

8.2 Deployment setup

Unity Engine: Unity engine is used for the front end designing we face some problems during the installation process due to the specifications of our system then we discuss it with our advisor and then we resolve that issue with installing the right version of unity.

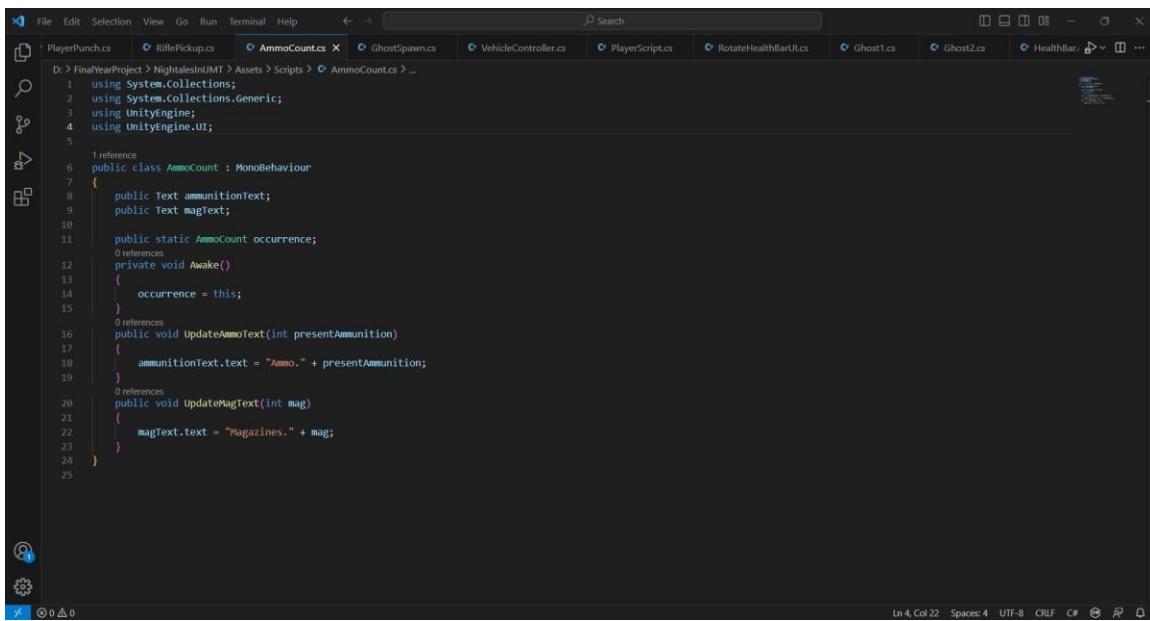
Visual Studio: Visual studio is used for the writing of C# script we face the issues for connection of visual studio with the unity, as we did not link the visual studio with the unity during the installation process. Later we find the way to resolve it by discussion with our advisor.

Blender: Blender is used for making models at start we are not use to this platform and facing some issues in template then we discuss with our advisor and resolve it.

Figma: Figma is used for UI/UX design at start we are not use to this platform and facing some issues in template then we discuss with our advisor and resolve it.

Adobe Photoshop: This is used for the sake of editing at start we are unable to get subscription and face some issues regarding to the tools later we find help from the advisor and YouTube.

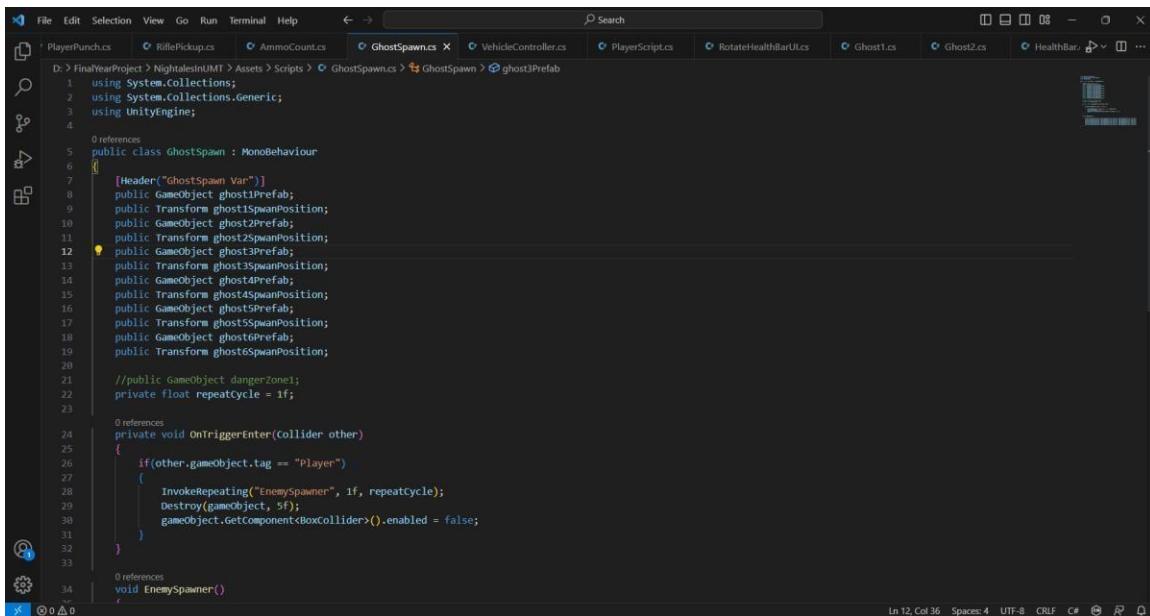
8.3 Algorithms



A screenshot of a code editor window showing the `AmmoCount.cs` script. The window title is "AmmoCount.cs". The code defines a class `AmmoCount` that implements the `MonoBehaviour` interface. It contains two text components: `ammunitionText` and `magText`. A static variable `occurrence` is used to store a reference to the current instance. The `UpdateAmmoText` method updates the `ammunitionText` with the string "Ammo." followed by the value of `presentAmmunition`. The `UpdateMagText` method updates the `magText` with the string "Magazines." followed by the value of `mag`.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 [reference]
7 public class AmmoCount : MonoBehaviour
8 {
9     public Text ammunitionText;
10    public Text magText;
11
12    public static AmmoCount occurrence;
13
14    private void Awake()
15    {
16        occurrence = this;
17    }
18
19    public void UpdateAmmoText(int presentAmmunition)
20    {
21        ammunitionText.text = "Ammo." + presentAmmunition;
22    }
23
24    public void UpdateMagText(int mag)
25    {
26        magText.text = "Magazines." + mag;
27    }
28}
```

Figure 10: Code Snippet 1



A screenshot of a code editor window showing the `GhostSpawn.cs` script. The window title is "GhostSpawn.cs". The code defines a class `GhostSpawn` that implements the `MonoBehaviour` interface. It contains several public transform variables for spawning ghost1 through ghost6. It also contains a private float variable `repeatCycle` and a private void method `OnTriggerEnter` that checks if the colliding object is a player and then destroys it after a repeating invoke call.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [reference]
6 public class GhostSpawn : MonoBehaviour
7 {
8     [Header("GhostSpawn Var")]
9     public GameObject ghost1Prefab;
10    public Transform ghost1SpawnPosition;
11    public GameObject ghost2Prefab;
12    public Transform ghost2SpawnPosition;
13    public GameObject ghost3Prefab;
14    public Transform ghost3SpawnPosition;
15    public GameObject ghost4Prefab;
16    public Transform ghost4SpawnPosition;
17    public GameObject ghost5Prefab;
18    public Transform ghost5SpawnPosition;
19    public GameObject ghost6Prefab;
20    public Transform ghost6SpawnPosition;
21
22    //public GameObject dangerZone;
23    private float repeatCycle = 1f;
24
25    private void OnTriggerEnter(Collider other)
26    {
27        if(other.gameObject.tag == "Player")
28        {
29            InvokeRepeating("EnemySpawner", if, repeatCycle);
30            Destroy(gameObject, 5f);
31            gameObject.GetComponent<BoxCollider>().enabled = false;
32        }
33    }
34
35    void EnemySpawner()
36    {
37    }
38}
```

Figure 11: Code Snippet 2

The screenshot shows the Unity Editor's code editor with the file `PlayerPunch.cs` open. The code implements a `PlayerPunch` component that punches objects in its range. It uses `Physics.Raycast` to find objects, and then checks if they are `ObjectToHit`, `ghost1`, or `ghost2` components. If found, it calls `ObjectToHit.ObjectHitDamage`, `ghost1.ghostHitDamage`, or `ghost2.ghostHitDamage` respectively, each passing `giveDamageOf` as a parameter.

```
D:\FinalYearProject>NightmaresInIMT>Assets>Scripts>PlayerPunch.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5     0 references
6 public class PlayerPunch : MonoBehaviour
7 {
8     [Header("Player Punch Var")]
9     public Camera cam;
10    public float giveDamageOf = 10f;
11    public float punchingRange = 5f;
12
13    0 references
14    public void Punch()
15    {
16        RaycastHit hitInfo;
17
18        if(Physics.Raycast(cam.transform.position, cam.transform.forward, out hitInfo, punchingRange))
19        {
20            Debug.Log(hitInfo.transform.name);
21
22            ObjectToHit objectToHit = hitInfo.transform.GetComponent<ObjectToHit>();
23            Ghost1 ghost1 = hitInfo.transform.GetComponent<Ghost1>();
24            Ghost2 ghost2 = hitInfo.transform.GetComponent<Ghost2>();
25
26            if(objectToHit != null)
27            {
28                objectToHit.ObjectHitDamage(giveDamageOf);
29            }
29            else if(ghost1 != null)
30            {
31                ghost1.ghostHitDamage(giveDamageOf);
32            }
32            else if(ghost2 != null)
33            {
34                ghost2.ghostHitDamage(giveDamageOf);
35            }
35        }
36    }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
135
136
137
138
139
139
140
141
142
143
144
145
145
146
147
148
149
149
150
151
152
153
154
155
155
156
157
158
159
159
160
161
162
163
164
165
165
166
167
168
169
169
170
171
172
173
174
175
175
176
177
178
179
179
180
181
182
183
184
185
185
186
187
188
189
189
190
191
192
193
194
195
195
196
197
198
199
199
200
201
202
203
204
205
205
206
207
208
209
209
210
211
212
213
214
214
215
216
217
218
218
219
220
221
222
223
223
224
225
226
227
227
228
229
229
230
231
232
233
233
234
235
235
236
237
238
238
239
239
240
241
242
242
243
244
244
245
246
246
247
247
248
248
249
249
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
14
```

Figure 12: Code Snippet 3

8.4 Constraints

8.4.1 Assumptions

- i. **Assumption:** Nightales in UMT will accurately represent the physics and mechanics of a real person walking, running, picking items.
 - ii. **Assumption:** The gun sounds, muzzle flash, ammo, and reload will be similar to real life animations.
 - iii. **Assumption:** The Player will have a fun time playing the game and exploring the map which is similar to UMT.
 - iv. **Assumption:** The Game will be visually appealing, fun to play, horror and based on a survival story.
 - v. **Assumption:** Sounds effects of shooting, enemies dying, will be implemented to give it a real life experience.

8.4.2 System constraints

System constraints for “Nightales in UMT” in a game can encompass various aspects of the development process. Here are some system constraints to consider:

- i. **Performance Constraints:** Nightales in UMT requires optimal performance to maintain smooth gameplay and visual fidelity. Constraints may include achieving a target frame rate, minimizing input lag, and optimizing rendering, physics, and collision detection algorithms.
 - ii. **Memory Constraints:** The size and complexity of characters, textures, and audio assets can impact memory usage. Constraints may involve managing memory efficiently, optimizing asset sizes, and considering memory limitations of the target platform(s).
 - iii. **Processing Power Constraints:** The computational demands of a Nightales in UMT can strain the processing capabilities of the target platform. Constraints may include optimizing code efficiency, minimizing CPU load, and utilizing parallel processing where applicable.

- iv. **Platform Constraints:** Different platforms (PC, consoles, mobile) have varying hardware capabilities and performance limitations. Constraints involve developing the simulation to meet the specific platform requirements, including graphics APIs, input methods, and memory restrictions.
- v. **Realism Constraints:** Achieving a realistic environment involves striking a balance between accuracy and performance. Constraints may involve deciding on the task, detail, determining physics parameters, and implementing realistic character and gun behavior within the limitations of real-time computation.
- vi. **Input Constraints:** The available input devices and their capabilities can impact the control and interaction with the Nightales in UMT. Constraints may involve designing the simulation to support different input methods, such as gamepads, keyboards, or touchscreens, and ensuring responsive and intuitive controls.
- vii. **Storage Constraints:** The storage space available on the target platform can limit the amount of content and assets that can be included in the game. Constraints involve managing storage efficiently and optimizing asset sizes.

8.4.3 Restrictions

The constraints applied to Nightales in UMT by the client can vary depending on their specific requirements and expectations. Here are constraints that clients might impose:

- i. **Time constraints:** The client may have a specific deadline or time frame within which they expect the game to be completed. It is important to manage development process efficiently to meet these time constraints.
- ii. **Performance constraints:** The client might require the game to run smoothly on a specific set of hardware or devices. Optimizing the game's performance, such as frame rate and memory usage, becomes crucial to meet these constraints.
- iii. **Design constraints:** The client may have specific design guidelines or aesthetic preferences that you need to adhere to. This could include the overall look and feel of the game, Player interface design, or branding requirements.
- iv. **Gameplay constraints:** The client might have specific gameplay mechanics or features they want to include or exclude in the game. It's important to understand and implement these constraints while designing and developing the gameplay experience.
- v. **Platform constraints:** The client may require the game to be developed for a specific platform or multiple platforms (such as PC, mobile, consoles). Ensuring compatibility and optimizing the game for the target platform) is essential.
- vi. **Budget constraints:** The client may have budget limitations that impact the resources and scope of the project. It's important to consider these constraints while planning the development process and managing the allocated resources effectively.

8.4.4 Limitations

When developing a Nightales in UMT, there are several limitations to consider. Here are limitations you may encounter:

- i. **Realism:** Achieving absolute realism in Nightales in UMT is challenging due to various factors such as limited computational power, simplified physics models, and approximations. The simulation may not perfectly replicate every aspect of real-world map of UMT.

- ii. **Hardware Constraints:** The performance of the game can be limited by the hardware capabilities of the target platform. Older or less powerful devices may struggle to handle complex 3D models, advanced physics calculations, or high-resolution textures, leading to reduced performance or visual quality.
- iii. **Development Time and Resources:** The available development time and resources can impose limitations on the level of detail and complexity that can be achieved in the simulation. Balancing the desired features and the available time and resources is essential.
- iv. **Processing Power:** Complex simulations with high-fidelity graphics and physics calculations can be computationally demanding. Lower-end devices may not be capable of handling such calculations in real-time, resulting in degraded performance or the need for optimization trade-offs.
- v. **Player Input and Controls:** The accuracy and precision of Player input devices, such as controllers or keyboards, can affect the control and handling of the character. Limitations in input devices may impact the realism and responsiveness of the game.
- vi. **Content Size and Optimization:** Large file sizes of 3D models, textures, and audio assets can limit the amount of content that can be included in the game. Optimization techniques, such as texture compression and level-of-detail (LOD) systems, are often necessary to manage the file sizes and maintain performance.
- vii. **Development Expertise:** Creating a high-quality Nightales in UMT requires expertise in areas such as game physics, 3D modeling, and programming. Limitations in the development team's skills and experience may impact the quality and scope of the simulation.
- viii. **Compatibility:** Ensuring compatibility across different platforms and devices can be challenging due to variations in hardware, operating systems, and graphics APIs. Limitations may arise in terms of supported platforms, performance optimizations, and feature availability.

9. TESTING

9.1 Extended Test Cases

Table 20: Login Button

Case Id: TC_1	Designed By: Muhammad Arqam											
Module Name: Login button	Design Date: 12/6/2023											
Priority: High	Executed By: Sumyya Anwar											
Name: To test the Sign in	Executed Date: 12/6/2023											
Description: Test the Login button												
Pre-Conditions: The Player shall open the game and click on Sign up with Google												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes						
1	Navigate to Google play				Pass							
2	Click the Sign In		Player moves to the next screen									
Post condition: The Player move to next screen and plays the game												

Table 21: Play Button

Case Id: TC_2		Designed By: Muhammad Arqam									
Module Name: Play button		Design Date: 15/6/2023									
Priority: High		Executed By: Ahmad Afaaq									
Name: To test the Play button		Executed Date: 15/6/2023									
Description: A player can play the game when clicks on Play button											
Pre-Conditions:											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to Play Button				Pass						
2	Click on Play		Player moves to the next screen								
3											
Post condition:											

Table 22: Save Button

Case Id: TC_3		Designed By: Sumyya Anwar									
Module Name: Save button		Design Date: 15/6/2023									
Priority: High		Executed By: Aneeqa Riaz									
Name: To test the save button		Executed Date: 15/6/2023									
Description: The player clicks on the save button to save their current progress											
Pre-Conditions: Launch the game and access the save button											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to Save button				Pass						
2	Click the Save button		The game is saved screen shall appear	The game is saved screen appears							
3											
4											
Post condition:											

Table 23: Exit Button

Case Id: TC_4	Designed By: Muhammad Arqam											
Module Name: Exit button	Design Date: 15/6/2023											
Priority: High	Executed By: Sumyya Riaz											
Name: To test the exit button	Executed Date: 15/6/2023											
Description: Test the exit button in game so that players can close the game.												
Pre-Conditions: Launch the game and access the exit button												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes						
1	Navigate to exit button				Pass							
2	Click the exit button		The game shall close	The game closes								
Post condition: The Player move to next screen and Play the game.												

Table 24: Continue button

Case Id: TC_5	Designed By: Muhammad Arqam											
Module Name: Continue button	Design Date: 15/6/2023											
Priority: High	Executed By: Sumyya Riaz											
Name: To test the Continue button	Executed Date: 15/6/2023											
Description: Test the continue button in game access continue button to resume where the player left the game.												
Pre-Conditions: Launch the game and play the game.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes						
1	Navigate to Continue button				Pass							
2	Click the Continue button		Player shall move to the play screen	Player move to the game.								
Post condition:												

Table 25: Settings button

Case Id: TC_6	Designed By: Muhammad Arqam											
Module Name: Settings	Design Date: 15/6/2023											
Priority: High	Executed By: Sumyya Riaz											
Name: To test the Settings button	Executed Date: 15/6/2023											
Description: Test the settings button in game and player can change the settings according to their needs												
Pre-Conditions: Launch the game and play it.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes						
1	Navigate to settings button				Pass							
2	Click the settings button		Player shall move to the setting screen	Player moved to the setting screen								
Post condition:												

Table 26: Select Story button

Case Id: TC_7	Designed By: Muhammad Arqam											
Module Name: Select Story button	Design Date: 15/6/2023											
Priority: High	Executed By: Sumyya Riaz											
Name: To test the Select story button	Executed Date: 15/6/2023											
Description: Test the select story button in game where player can select which story mode they want to play												
Pre-Conditions: Launch the game and play the game.												
Dependencies												
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes						
1	Navigate to Continue button				Pass							
2	Click the Continue button		Player shall move to the play screen	Player move to the game.								
Post condition:												

Table 27: Kill the enemies and Complete Quests

Case Id: TC_8		Designed By: Muhammad Arqam									
Module Name: Kill Enemies and Complete Quests		Design Date: 15/6/2023									
Priority: High		Executed By: Sumyya Riaz									
Name: To test the functionalities of shootings and completing quests		Executed Date: 15/6/2023									
Description: Test the functionalities of shooting and destroying the enemies and completing the quests											
Pre-Conditions: Launch the game and play the game.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Click on the left mouse button to shoot the enemies		The enemy shall take damage	The enemy takes damage	Pass						
2	Complete the quests by doing certain tasks		Player shall complete tasks by doing challenges	Player completes the challenges.							
Post condition:											

Table 28: Restart Button

Case Id: TC_9		Designed By: Muhammad Arqam									
Module Name: Restart button		Design Date: 15/6/2023									
Priority: High		Executed By: Sumyya Riaz									
Name: To test the restart button		Executed Date: 15/6/2023									
Description: Test the restart button in game where if the player dies he/she can click the restart button and play the game again											
Pre-Conditions: Launch the game and play the game.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to Restart button				Pass						
2	Click the Restart button		The game shall restart from the last checkpoint	The game restarts from the last check point							
Post condition:											

Table 29: Main Menu Button

Case Id: TC_10		Designed By: Muhammad Arqam									
Module Name: Main Menu button		Design Date: 15/6/2023									
Priority: High		Executed By: Sumyya Riaz									
Name: To test the main menu button		Executed Date: 15/6/2023									
Description: Test the main menu button in game where if the pauses the game or dies from enemies attack, the player can click the main menu to return to the start page of the game.											
Pre-Conditions: Launch the game and play the game.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to Main Menu button				Pass						
2	Click the Main Menu button		The player shall move from game screen to main menu screen.	The player moves from game screen to main menu screen.							
Post condition:											

Table 30: Feedback button

Case Id: TC_11		Designed By: Muhammad Arqam									
Module Name: Feedback button		Design Date: 15/6/2023									
Priority: High		Executed By: Sumyya Riaz									
Name: To test the Feedback button		Executed Date: 15/6/2023									
Description: Test the feedback button in game where the player can gives its feedback about the game.											
Pre-Conditions: Launch the game and play the game.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to Feedback button				Pass						
2	Click the Feedback button		The player shall move from game screen to feedback screen.	The player moves from game screen to feedback screen.							
Post condition:											

9.2 Traceability Matrix

9.2.1 RID vs UCID (requirements vs use cases)

Table 31: RID vs UCID

UCID/RID	R 1	R 2	R 3	R 4	R 5
UC 1	✓				
UC 2		✓			
UC 3				✓	
UC 4					✓
UC 5			✓	✓	
UC 6					
UC 7					
UC 8					
UC 9					
UC 10					
UC 11					

9.2.2 Prototypes (RID vs PID)

Table 32: PID vs RID

PID/RID	R 1	R 2	R 3	R 4	R 5
PID 1	✓				
PID 2		✓	✓		
PID 3					✓
PID 4				✓	

9.2.3 Test Cases (RID vs TID)

Table 33: RID vs TID

TID/RID	R 1	R 2	R 3	R 4	R 5
TID 1	✓				
TID 2		✓			
TID 3				✓	
TID 4					✓
TID 5			✓		
TID 6					
TID 7					
TID 8					
TID 9					
TID 10					
TID 11					

9.2.4 Coverage (UCID vs TID)

Table 34: UCID vs TID

TID/UCID	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11
TID 1	✓										
TID 2		✓									
TID 3			✓								
TID 4				✓							
TID 5					✓						
TID 6						✓					
TID 7							✓				
TID 8								✓			
TID 9									✓		
TID 10										✓	
TID 11											✓

10. RESULTS/OUTPUT/STATISTICS

10.1 %completion

Fulfilled requirements: 14

Total requirements: 15

$$\% \text{completion} = (14/15) * 100 = 93.33\%$$

10.2 %accuracy

Implemented requirements: 13

Total requirements: 15

$$\% \text{accuracy} = (13/15) * 100 = 86.67\%$$

10.3 %correctness

Tested requirements: 13

Total requirements: 15

$$\% \text{accuracy} = (13/15) * 100 = 86.67\%$$

11. CONCLUSION

In conclusion, the game “Nightales in UMT” was a fun experience as a final year project. We did face some difficulties along the way of making the game i.e., budget, gear problem, processing power, memory issues, time management, communication gap. After all this, we were able to deliver the game and present it as our final year project.

The primary goal of the game was successfully achieved which was to create a character that has to survive from the attacks of the ghost and protect itself from the enemies attack and complete side task along the way.

The second goal was to create the map of the UMT, which we are still working on, because we do not have enough resources to make the exact replicate of the map and the equipment that is required is over the budget for us, but still somehow managed to make a replicate of the UMT Map and delivered the core objective of the game.

The projects main purpose was to create a fun little game that will cover the basic logic of a survival game, with a slight bit touch of horror, puzzle solving, tasks, map awareness, fun environment, proper game experience and much more. The map that we replicated as UMTs Map took us the most time, because we wanted the users to have a fun experience while playing the game and do not feel bounded while playing or get bored.

Along the way of making this project all, the group members showed their absolute best in order to complete the project. No doubt, there were some difficulties along the way but we still managed to do it within the given time limit. In addition, the period that was given for the final year project was less but still we managed to do the project within the given period.

Yes, there were some issues in-between the group members regarding the working process and time but we figured all of the problems and their solutions respectively which not only made us strong but independent and make us work even harder for us to complete the project and this also helped us build a strong team and work accordingly and developed a sense of Team management and leadership and Time management as well.

We also would thank our advisor Sir Noaman Saleem, who accepted us and helped us throughout the project and was very collaborative and helpful with giving wonderful ideas and he divided the tasks between the members which made the project go through like a breeze.

We also would like to thank our co-advisor Sir Rana Marwat Hussain, who gave us the idea of making a game based on the UMT map. He also helped us throughout the project and always checked up on us when were in need of help.

Lastly we would like to thank UMT for providing us with the best possible faculty that a student can ask for, We would like to thank our FYP Department Head Sir Jameel Ahmad and Ma'am Sundas Rashid and also we would like to thank Ma'am Ayesha Asmat, Sir Rana Marwat Hussain, Sir Rehan Raza, Sir Owais Khan, Sir Khawaja Ubaid-ul-Rehman, and other faculty members as well for teaching us and helping us throughout this project.

12. FUTURE WORK

For future work we are planning on doing the following things:

- **Refining the map even more:** We want to refine the map and make it as much replicated to UMT Map as much as possible.
- **Add more stories:** We want to add more stories along the way and make it more fun for the end user to play.
- **Expand the map even more:** We want to expand the map to other various places like Fortress, DHA, Bahria, etc., for the players to have a vast variety of maps to play in and explore more new places.
- **Add more vehicles:** Adding vehicles for the players to travel in between the map
- **Expanding the story line:** Expanding the story line to add more drama and suspense and fun little tasks for the user to finish and win the game
- **Update on regular bases:** We want to keep this game up-to-dated so that every time there is a bug in the game we solve it first thing.
- **Add more guns:** Add more guns and artillery for the user to use and enjoy with.
- **Enhance the Graphics:** We want to enhance the user experience even more by making it realistic for the players to play so that they can enjoy the game even more
- **Making it Multiplayer:** Implement a cooperative mode where players can team up with friends or other online players to survive and solve various task. This could involve designing cooperative missions, implementing communication features, or integrating a matchmaking system for finding other players.

Final Year Project Report
Project Name “Infected Zombie City”



Project Advisor: Mr. Khawaja Ubaid Ur Rehman

Submitted By:

MUHAMMAD NOUMAN	F2018266300
MUHAMMAD DANYAL	F2018266206
AHMAD MOBEEN	F2018266381

Session
2018-2023
BS-CS

School of Science and Technology
University of Management and Technology
C-II Johar Town Lahore Pakistan

Dedication

By the grace of Allah, our creator; our source of Knowledge, we are able to complete this project. Our teachers and mentors played a vital role in the making of this project. They provided their valuable feedback and shared their valuable experience giving us their precious time with which we proceeded.

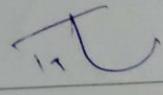
So, we would like to dedicate this project to our parent's mentors, also the project advisor ***Khawaja Ubaid ur Rehman*** who showed us the right way of doing things, showed us how simple things can be made pretty interesting. If it were not for him this project would not be what it is now. We are very grateful to all of them for putting their faith in us for believing in us in developing such a game (**Infected zombie city**). While designing harry in grave game for android we employ three key qualities like dedication, determination and discipline.

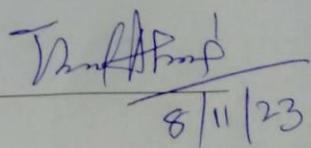
Final Approval

OPPO A52

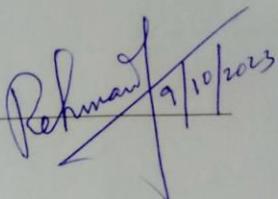
Final Approval

- **Head of Department**
Department of Computer Science
School of Systems & Technology
UMT Lahore
- **Program Director (Final Year Projects)**
Department of Computer Science.
School of Systems & Technology
UMT Lahore
- **Supervisor**
Department of Computer Science.
School of Systems & Technology
UMT Lahore
- **Co-Supervisor**





8/11/23.



Rehman 9/10/2023



Acknowledgment

Our deepest gratitude goes to Allah Almighty who has provided us Knowledge, Wisdom, Strength to cover this project to completion. I appreciate the School of System and Technology (UMT) for providing us the tools and technology which was crucial in making this project.

Furthermore, we would like to appreciate Mr. Ibrahim Murad for carrying on the legacy of his father, this university which is the source of our education for providing us various opportunities to enhance ourselves educationally moreover professionally. We would also like to appreciate the dean of SST for running things smoothly for the students working on their final year projects.

Finally, we are very grateful to Khawaja Ubaid ur Rehman for supervising this project. All credit goes to him. He is a true leader the way he guided us throughout the project is phenomenal. Under his supervision, we were able to complete our project Blood bank application.

We acknowledge his love, timely assistance, and guidance, which paved a way for us to follow. If it was not for him, we would not be able to even start the project. Our every step is supervised by him and we are very grateful to him for that.

Project Title:

- INFECTED ZOMBIE CITY
- **Objective:**
 - Provide an online horror game for game lovers.
 - To spread awareness about how to survive in death threatening crises like covid-19
 - Even a child of 10-13 years of age can also play this game.
- **Undertaken by:**

NAME, s	ID
MUHAMMAD NOUMAN	F2018266300
MUHAMMAD DANYAL	F2018266206
AHMAD MOBEEN	F2018266381

- **Supervised by:** Mr. Khawaja Ubaid ur Rehman
- **Starting Date:** 10-June-2022

- **Completion Date:**

In Progress

- **Tools Used:**

Unity Engine (For designing front end)
Visual Studio (For writing Scripts in C#)
Blender (For making Models)
Photoshop (For UI design)
Online sounds websites

- **Front End:**

Unity: Unity is, simply put, the world's most popular game engine. It packs a ton of features together and is flexible enough to make almost any game you can imagine. With unrivaled cross-platform features, Unity is popular with both hobby developers and AAA studios. It's been used to create games like Pokémon Go, Hearthstone, Rim world, Cup head, and plenty more.

Blender: We use this tool to make models and characters.

Photoshop: We use this tool to design the UI of the game.

- **Back End:**

C Sharp (C#): C# (pronounced C sharp) is one of the most popular programming languages in the world. Developed by Microsoft more than 20 years ago, C# is built on

top of the dot NET (.NET) Framework which is a free open-source, cross-platform developer platform for building different types of apps. C# is an object-oriented, statically typed, general-purpose programming language widely used for: Internet of Things and Game development for PC, Mac, Consoles, Mobile, and VR|AR with Unity

- **Operating System:**
 - Android
- **Documentation:**
 - Version 1

Plagiarism Report

University of Management and Technology, Lahore

Similarity Report

Turnitin Originality Report
Infected Zombie City by Muhammad Nouman, Muhammad Danyal and Ahmad Mobeen

From Quick Submit (Quick Submit)

- Processed on 04-Oct-2023 15:43 PKT
- ID: 2185317323
- Word Count: 10590

Similarity Index

19%

Similarity by Source

Internet Sources:

11%

Publications:

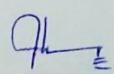
1%

Student Papers:

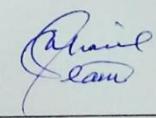
17%

Sources:

1. 4% match (student papers from 30-Jul-2018)
Submitted to Higher Education Commission Pakistan on 2018-07-30
2. 4% match (student papers from 31-Jul-2020)
Submitted to Higher Education Commission Pakistan on 2020-07-31



Checked by



For Verified by CLO

Note:

- Sometimes the overall similarity index may be smaller than the repository percentages combined. This would be due to overlapping text within the repositories.
- It is a system generated report.

Declaration Form

(To be submitted by FYP Advisor)

Declaration Form

(To be submitted by FYP Advisor)

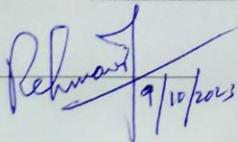
I have carefully examined the documentation of the Final Year Project titled "INFECTED ZOMBIE CITY"; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with user manual and installation guide.

FYP Advisor Name: *Khawaja Ubaid ur Rehman*

Signature:



A handwritten signature in blue ink, appearing to read "Rehman". To the right of the signature, the date is written vertically as "9/10/2023".

Date:

9/10/2023

Abstract

In recent times, Gaming industry is booming at a very high rate. It is expected that the gaming industry will be worth 320 billion \$ by the end of 2026. There are hundreds of thousands of games right now but these days most of the people find it very difficult to find horror games for their Childs. There are many reasons behind that. Most games are made for mature audiences (18+). We are going to make a game which anyone can play, even kids of 14-15 years of age. Our main focus will be the Childs who want to play horror games without any fear. The story of this game will be very attractive towards the new generation.

● REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
Draft	NOUMAN	Scope and Work flow of the object was defined	11-06-2022
Preliminary	DANYAL	Use cases were decided	15-06-2022
Preliminary 2	DANYAL	Core gameplay diagram	
Final	NOUMAN	Grammar has been checked and suggestions for improvement have been made.	25-06-2022
Revision 3	DANYAL	Class diagram	10-07-2022
Revision 4	DANYAL	Screenshots of the mockup designs.	15-07-2022

● CONTENTS

Dedication	i
Final Approval.....	ii
Acknowledgment.....	iii
Project Title:	iv
Plagiarism Report	vi
Declaration Form.....	vii
Abstract	viii
● REVISION CHART	IX
● CONTENTS	X
DEFINITIONS AND ACRONYMS	XII
LIST OF FIGURES	XIII
LIST OF TABLES	XIV
1. GAME OVERVIEW	1
1.1 GAME SUMMARY	1
1.2 TARGET PLATFORM(S).....	1
1.3 BUSINESS MODEL	1
1.4 THEME / SETTING / GENRE	1
1.5 CORE GAMEPLAY MECHANICS	1
1.6 SERVER / ONLINE MECHANICS.....	1
1.7 ART STYLE	1
1.8 LOOK & FEEL/ CAMERA.....	2
1.9 PROGRESSION	2
2. STORY AND GAMEPLAY.....	3
2.1 STORY.....	3
2.2 CORE GAMEPLAY.....	3
3. DOMAIN ANALYSIS.....	4
3.1 CUSTOMER.....	4
3.2 STAKEHOLDERS	4
3.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT	4
3.4 DEPENDENCIES/ EXTERNAL SYSTEMS.....	4
3.5 REFERENCES	4
3.5.1 Related Projects.....	4
4. REQUIREMENTS ANALYSIS	5
4.1 REQUIREMENTS	5
4.2 LIST OF ACTORS.....	7
4.3 LIST OF USE CASES	7
4.4 SYSTEM USE CASE DIAGRAM.....	8
4.5 EXTENDED USE CASES.....	9
5. USER INTERFACE / SCREENS.....	31
5.1 MAIN MENU.....	31
5.2 GAMEPLAY SCREEN	32
5.3 PAUSE MENU	33
5.4 EXIT MENU.....	34
5.5 SETTING MENU.....	35
5.6 LEVEL FAIL	35
	36

5.7	CONGRATULATION MENU	37
6.	LOADING SCREEN.....	38
7.	LEVELS	39
❖	FIRST TIME EXPERIENCE.....	40
8.	ASSET LIST	41
9.	IMPLEMENTATION DETAILS.....	42
9.1	DEVELOPMENT SETUP.....	42
9.2	DEPLOYMENT SETUP.....	42
9.3	ALGORITHMS	42
9.4	CONSTRAINTS	44
9.4.1	<i>Assumptions.....</i>	44
•	<i>First, you will get a user interface where you can select the button you want to select e.g. Play to start the game, settings, and exit, leaderboard and customization options.</i>	44
•	<i>When you click the Play button, an interface will appear where you can select the weapon you want to play with.</i>	44
9.4.2	<i>System constraints</i>	44
9.4.3	<i>Restrictions</i>	44
9.4.4	<i>Limitations</i>	44
10.	TESTING.....	45
10.1	EXTENDED TEST CASES	45
10.2	DECISION TABLE.....	57
10.2.1	<i>Code snippet.....</i>	58
10.2.2	<i>Decision coverage table</i>	60
10.3	TRACEABILITY MATRIX	69
10.3.1	<i>RID vs UCID (requirements vs use cases)</i>	69
10.3.2	<i>Prototypes (RID vs PID)</i>	70
	<i>Table 10.7: Prototypes (RID vs PID).....</i>	70
10.3.3	<i>Test Cases (RID vs TID).....</i>	71
	<i>Table 10.8: Test Case (RID vs TID).....</i>	71
10.3.4	<i>Coverage (UCID vs TID)</i>	72
	<i>Table 10.9: Coverage (UCID vs TID).....</i>	72
11.	RESULTS/OUTPUT/STATISTICS	74
11.1	%COMPLETION	74
11.2	%ACCURACY	74
11.3	%CORRECTNESS	74
12.	CONCLUSION.....	75
13.	FUTURE WORK	76
14.	BIBLIOGRAPHY	77
14.1	OTHER REFERENCES	77
15.	APPENDIX	78
15.1	GLOSSARY OF TERMS	78
15.2	PRE-REQUISITES	78

Definitions and Acronyms

Acronym	Definition
FPS	First person shooter
3D	3-dimension view
LWRP	Lightweight render pipeline
SRS	Software requirement specification
NPC	Non-player character

List of Figures

Figure 2.1: Core Gameplay Diagram.....	3.
Figure 4.1: System Use Case Diagram.....	8
Figure 5.1: Main Menu Diagram.....	30
Figure 5.2: Game Gameplay Diagram.....	31
Figure 5.2: Game pause menu.....	32
Figure 5.2: Game exit menu.....	33
Figure 5.2: Game setting menu.....	34
Figure 5.2: Game level fail menu.....	35
Figure 5.2: Game congratulation menu.....	36
Figure 6.1: Loading Screen Diagram.....	37
Figure 7.1 complexity of levels.....	38
Figure 9.1: Code snippet of player control.....	41
Figure 9.1: Code snippet of weapon control.....	42
Figure 9.1: Code snippet of zombie control.....	42
Figure 9.1: Code snippet of bullet hit.....	57
Figure 9.1: Code snippet of resume game.....	58
Figure 9.1: Code snippet of player death.....	58

List of Tables

INFECTED ZOMBIE CITY

List of Tables

Table 1: List Of Acronyms and Definitions	xii
Table 3.1: List of Stakeholders	4
Table 4.1: Requirements	5
Table 4.5: Extended Use case.....	9
Table 10.1: Extended Test case.....	44
Table 10.2: Decision Table.....	56
Table 10.2.2: Decision Coverage Tabl.....	59
Table 10.3.1: RID vs UCID.....	68
Table 10.3.2: Prototype (RID vs PID).....	69
Table 10.3.3: Test Case (RID vs TID).....	70
Table 10.3.4: Coverage (UCID vs TID).....	71

1. GAME OVERVIEW

1.1 Game Summary

Our game is based on the frictional horror story. In the game we have a cut scene in which we see the story of how the virus starts spreading and infects people. We also see that the government of the country takes strict action toward the city by locking all entries and by deploying black ops to eliminate zombies and find the cure of infection. They also advise other city residents to stay at home and wear protective masks to lower the chance of getting infective. On the other hand, our black ops reached the lab and fought with the boss to get the cure location. When they reach the place where the cure is located gameplay will end and another cut scene in which we show how they release cure in air and also show the effect of cure on infected people then our game will end.

1.2 Target Platform(s)

- Modern Android tablets and phones.
- The game will be in Landscape mode.

1.3 Business Model

- Our game earns revenue from Integrate Rewarded Ads which give the player some coin in case he watches the ad. The watch ad option will come after you complete or fail the game.

1.4 Theme / Setting / Genre

Our game theme is based on survival and horror games. Our game is based on the frictional horror story. This is the story of the brave soldiers who fought and survived the zombie outbreak to save humanity and cure infected people.

1.5 Core Gameplay Mechanics

Controls

- Single Touch
- Joystick
- Tap
- Slide
- Swipe
- Drag

1.6 Server / Online Mechanics

- We also added server mechanics like weekly leaderboard by making our game an online multiplayer game and using a photon server. Our game supports solo, duo and squad gameplay style.

1.7 Art Style

- **3D:**

For Art style we use 3D art style for map and for characters we use 3D art.

1.8 Look & Feel/ Camera

- **Third Person:**

Over game camera mechanism is a third person viewpoint that means the player can see the whole character from behind.

1.9 Progression

- **Engagement Based:**

Game's progression is tied to the player's engagement with the story and their progress towards completing tasks and reaching the main objective. As the player progresses through the game, the story advances and they get closer to the end.

2. STORY AND GAMEPLAY

2.1 Story

This is the story of the brave soldiers who fought and survived the zombie outbreak to save humanity and cure the infected people. The story begins with experiments releasing a top secret virus that spreads very quickly in the laboratory and turns everyone in the laboratory into zombies. It then begins to infect itself through the air outside the laboratory, turning large numbers of people in City X into man-eating monsters. Given the situation, the country's government has issued a red alert and advised people to wear masks and stay indoors. To find the cure in the lab, the government deployed its best black ops unit to deal with the zombies and release the vaccine into the air like the virus, so that the vaccine releases the zombie virus into the air and in the Body system of the infected killed people. Our secret service soldiers enter the city and make their way to the laboratory. They are also attacked by zombies and viruses begin to infect their bodies, but they can inoculate themselves with small doses found in small branches of the main laboratory. Somehow they manage to get to the main lab, where they fight the main boss and the first infected zombie. When he dies, he tells us the correct location of the vaccine in the building and how to release it into the air. Upon arrival at the vaccination site, our Soldiers release the vaccine by firing a small rocket into the air. So the vaccination spreads throughout the country, the infected are cured day by day and the city returns to its original state. Our heroes receive huge rewards and the government apologizes to its people and destroys all records and investigative documents so that this terrible story can now be repeated in the future.

2.2 Core Gameplay

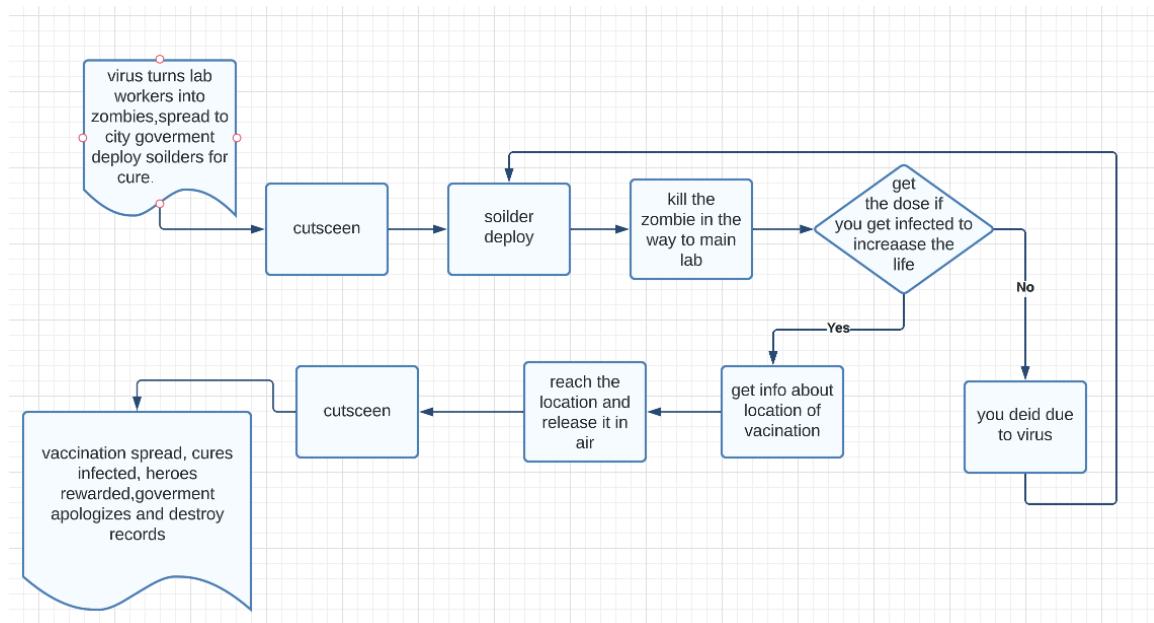


Figure 2.1: core gameplay

3. DOMAIN ANALYSIS

3.1 Customer

Our main customers are going to be the younger generation especially children of the 10-13 age group as young people are more attracted towards video games.

3.2 Stakeholders

Table 3.1: List of stakeholders

Stakeholder	Role in System
Developer team	The developer team will be able to make any kind of updates. They also look and solve any glitch and problem in the game so we can give better experience in the game.
player	The player will be able to play and give feedback if they saw any glitch and bug.

3.3 Affected Groups with social or economic impact

- The general population, who would be informed on how to protect themselves during a deadly virus outbreak
- People who may have been directly affected by the COVID-19 pandemic or other similar crises, such as losing loved ones or experiencing economic hardship
- Healthcare workers and first responders who are on the frontlines of dealing with outbreaks
- The government and public health organizations responsible for managing and responding to outbreaks
- The economy as outbreaks can lead to disruptions in businesses and industries

3.4 Dependencies/ External Systems

Our system is depending upon:

- Admin
- Android devices.

3.5 References

- World War Z.
- Last Of US

3.5.1 Related Projects

- Resident evil
- World War Z

4. REQUIREMENTS ANALYSIS

4.1 Requirements

Table 4.1: List of requirements

RID	description	Category	Attribute	Details & Boundary Constraints
FR1	MAIN MENU	functional	Function required for main menu.	<ul style="list-style-type: none"> • Players should be able to start the game. • Players should be able to select different options. • Players should be able to exit the game. • Player should be able to load a game from where he recently exits the game
FR2	PAUSE MENU	functional	Function required for pause menu.	<ul style="list-style-type: none"> • Players should be able to resume the game. • Players should be able to select different options like audio setting and control setting etc. • He should be able to exit the game.
FR3	GAMEPLAY MODE	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> • Players should be able to start the game. • Players should be able to use shooting abilities to kill zombies. • Players should be able to move left, right, forward and backward. • Players can die. • Players should be able to exit the game.
Fr4	Play game	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> • Players can start the game by pressing the buttons.
Fr5	Gun selection	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> • Players can select different types of guns.

Fr6	Health	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players can monitor his character's health through a health bar.
Fr7	Stamina	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players can monitor his stamina through the stamina bar.
Fr8	Sprint	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players should be able to run fast left, right, forward and backward. Players are able to run fast.
Fr9	Jump	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players can jump from obstacles.
Fr10	Move	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players should be able to move left, right, forward and backward. Players are able to walk.
Fr11	Fire	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players should be able to use shooting abilities to kill zombies
Fr12	Settings	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players can change game settings according to his liking.
Fr13	Exit	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players should be able to exit the game.
Fr14	Customization	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> Players can customize his character costume and gun skins.
Fr15	Chase	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> The NPC chased the player when he came near them.
Fr16	Attack	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> The NPC attacked the player when he came near them.
Fr17	Sound	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> NPC make sounds so they can create a fearful environment.

Fr18	Walk	Functional	Function required for gameplay mode.	<ul style="list-style-type: none"> • NPC walk in around their specific position.
NFR1	Usability	Non-functional	Usability requirements for Infected Zombie City.	<ul style="list-style-type: none"> • All the elements of the game must be designed in accordance with a naive user perspective; however, the gameplay requires beginner level knowledge of FPS games. • The game should provide key points or instructions on how to play the game.
NFR2	Performance	Non-functional	Performance requirement for game.	<ul style="list-style-type: none"> • The game should run at a minimum of 30 frames per second. • The latency overall should be less than 200 milliseconds.

4.2 List of Actors

Player: Can play games and alert about bugs by feedback.

Zombie: they are like NPC but they can attack and kill you if you do not protect yourself.

4.3 List of use cases

Player: -

1. **Play game:** Players can start the game by clicking on the option play game.

I. **Gun selection:** Player can select his gun which he wants to use in game.

II. **Player status:**

i. **Health:** It shows the player remaining life.

ii. **Stamina:** It shows the stamina left for a person to run.

III. **Game mechanics:**

i. **Movement:**

I. **Sprint:** player can sprint by pressing the sprint button.

II. **Jump:** player can jump by pressing the jump button.

III. **Move:** Players can move in every direction in the map of the game by pressing the arrow key or joystick.

IV. **Fire:** player can fire by pressing the fire button from the gun.

2. **Settings:** He accesses other setting options by clicking on it.

3. **Exit game:** By this option, the player can close the game when he wants to.
 4. **Pause option:** Players use this option to return to the game or restart the game.
 5. **Customize:**
 - I. **Buy gun and costume:** Players can buy guns and costumes for their character.
 - II. **Customize gun and costume:** Players can select and equip different guns and costumes.
- 6. Zombie: -**
- I. **Chase:** They can chase you when you go near them.
 - II. **Attack:** They also attack players in their area and cause infection.
 - III. **Sound:** They have a scream sound so you can alert yourself.
 - IV. **Walk:** They just walk around until you go near them.

4.4 System Use Case diagram

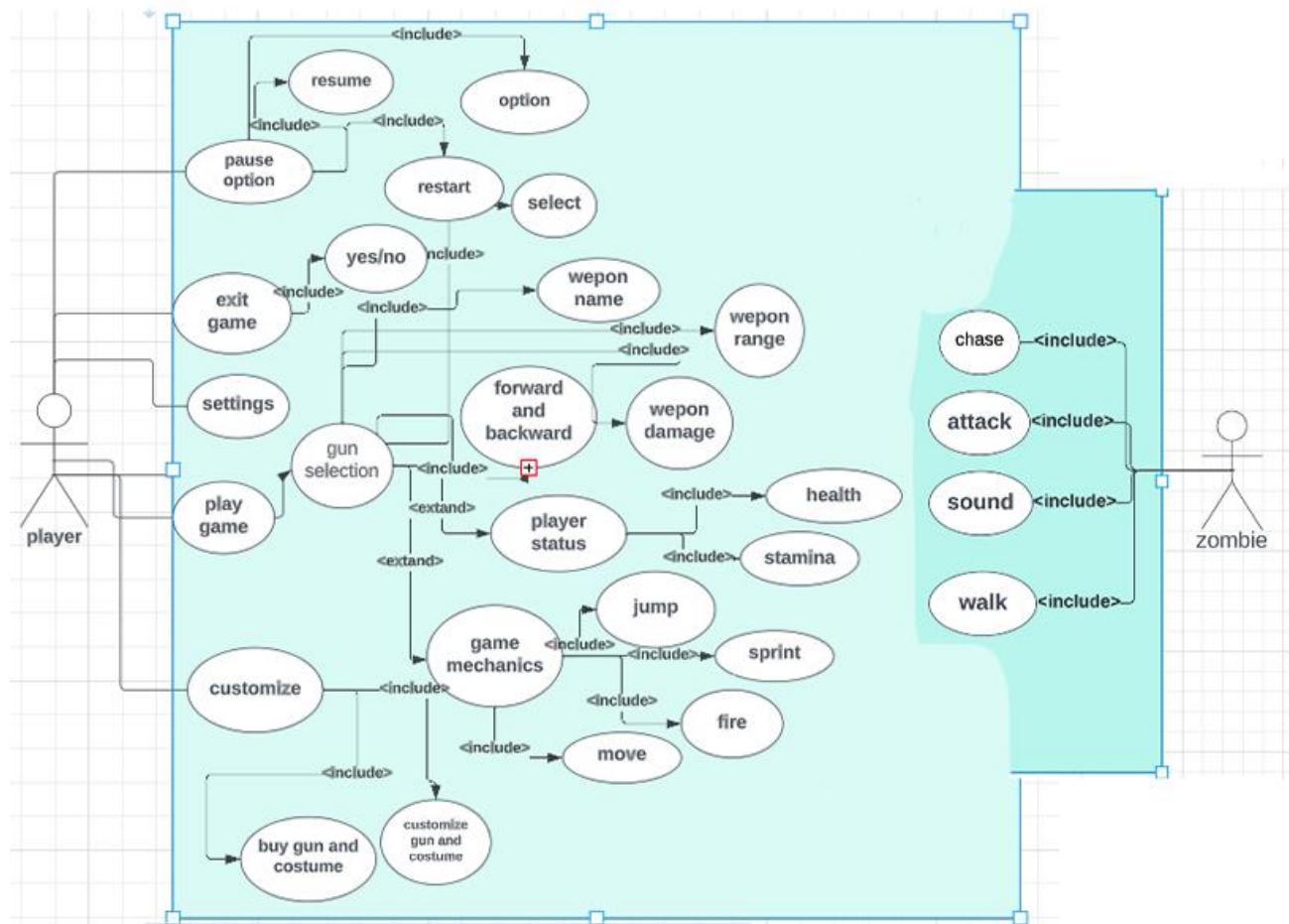


Figure 4.1: Use case diagram of Infected Zombie City

4.5 Extended use cases

Table 4.2: Play game:

Use Case ID:	UC-1.		
Use Case Name:	Play games.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Gamer/Player		
Description:	Players can start the game by clicking on the option play game.		
Trigger:	As the user opens this application, it will choose to play or not.		
Preconditions:	<ul style="list-style-type: none"> • The application should have launched correctly 		
Post conditions:	<ul style="list-style-type: none"> • The game starts 		
Normal Flow:	<ol style="list-style-type: none"> 1. On the title screen, the play game button populates on the screen. 2. Upon pressing the button, the application should take the user to the gameplay screen. 		
Alternative Flows:	<ul style="list-style-type: none"> • User exits the application by pressing the back or home button of the Android device. • Or a game crash due to some error. 		
Includes:	This use case starts the game for the user to play.		
Frequency of Use:	Every time a user opens this application.		
Assumptions:	It is assumed that the user understands the English and Urdu language		
Notes and Issues:	Press the play game button.		

Table 4.3: Gun selection:

Use Case ID:	UC-2		
Use Case Name:	Gun selection.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Players can select guns.		
Trigger:	Press on play game button then it takes you to gun selection.		
Preconditions:	<ul style="list-style-type: none">• Players can see different guns and their damage range by pressing the left and right button.		
Post conditions:	<ul style="list-style-type: none">• The player can select a gun.		
Normal Flow:	<ul style="list-style-type: none">• On pressing the play game button then it takes you to gun selection.		
Includes:	The use case helps users to select guns.		
Frequency of Use:	Users can select different weapons.		
Assumptions:	User should have some knowledge of fps games		
Notes and Issues:	Press the select button.		

Table 4.4: Player status:

Use Case ID:	UC-3		
Use Case Name:	Player status.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Player can see his status of health and stamina.		
Trigger:	When the player starts the game.		
Preconditions:	<ul style="list-style-type: none">● The player is playing a game.		
Post conditions:	<ul style="list-style-type: none">● The player dies or gets injured and starts running in the game.		
Normal Flow:	<ul style="list-style-type: none">● After getting hit by the zombies and after running.		
Includes:	The use case shows the player's health and stamina status.		
Frequency of Use:	When a user dies or gets injured and starts running in the game.		
Assumptions:	Users should have some knowledge of fps games.		

Table 4.5: Health:

Use Case ID:	UC-4		
Use Case Name:	Health.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Player can see his health status.		
Trigger:	When the player starts the game.		
Preconditions:	<ul style="list-style-type: none">• The player is playing a game.		
Post conditions:	<ul style="list-style-type: none">• The player is dead or gets injured in the game.		
Normal Flow:	<ul style="list-style-type: none">• After getting hit by the zombies.		
Includes:	The use case shows the player's health status.		
Frequency of Use:	When a user dies or gets injured.		
Assumptions:	Users should have some knowledge of fps games.		

Table 4.6: Stamina:

Use Case ID:	UC-5		
Use Case Name:	Stamina.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Characters can become tired by continuing running.		
Trigger:	When the player presses the run button.		
Preconditions:	<ul style="list-style-type: none">The player is playing a game.		
Post conditions:	<ul style="list-style-type: none">The player's stamina bar starts to decrease in game.		
Normal Flow:	<ul style="list-style-type: none">After running for a while.		
Includes:	The use case shows the player stamina status.		
Frequency of Use:	When a user wants to run.		
Assumptions:	Users should have some knowledge of fps games.		

Table 4.7: Game mechanics:

Use Case ID:	UC-6		
Use Case Name:	Game mechanics.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	It defines the functionality and physics of the game.		
Trigger:	When the player starts the game.		
Preconditions:	<ul style="list-style-type: none">• The player should be in the game.		
Normal Flow:	<ul style="list-style-type: none">• all the physics and functionality should be applied		
Includes:	The use case defines the functionality and physics of the game.		
Frequency of Use:	When the player starts the game.		
Assumptions:	Developers should have some knowledge of fps games physics and functionality.		

Table 4.8: Movement:

Use Case ID:	UC-7		
Use Case Name:	Movement.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Players can move in the game world.		
Trigger:	Press and move your finger to move.		
Preconditions:	<ul style="list-style-type: none">● The player should be in the game.		
Normal Flow:	<ul style="list-style-type: none">● Players can move around the world by moving the joystick.		
Includes:	The use case uses to move players.		
Frequency of Use:	When users walk or run.		
Assumptions:	Users should have some knowledge of fps games.		
Notes and Issues:	Press and move the joystick button.		

Table 4.9: Fire:

Use Case ID:	UC-8		
Use Case Name:	Fire.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Players can kill zombies.		
Trigger:	Press on the fire button.		
Preconditions:	<ul style="list-style-type: none">The player holding a gun.		
Post conditions:	<ul style="list-style-type: none">The player fires bullets from a gun.		
Normal Flow:	<ul style="list-style-type: none">On pressing the firing button, the button player shoots from a gun to kill zombies.		
Includes:	The use case helps users to kill zombies.		
Frequency of Use:	When a user wants to attack.		
Assumptions:	User should have some knowledge of fps games		
Notes and Issues:	Press the fire button.		

Table 4.10: Jump:

Use Case ID:	UC-9		
Use Case Name:	Jump.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Players can jump.		
Trigger:	Press on the jump button.		
Preconditions:	<ul style="list-style-type: none">There is an obstacle in front of the player.		
Post conditions:	<ul style="list-style-type: none">The player can jump up to clear the obstacle.		
Normal Flow:	<ul style="list-style-type: none">On pressing the jump button, the button player jumps from the ground.		
Includes:	The use case helps users to jump.		
Frequency of Use:	When a user wants to jump.		
Assumptions:	User should have some knowledge of fps games		
Notes and Issues:	Press the jump button.		

Table 4.11: Sprint:

Use Case ID:	UC-10		
Use Case Name:	Sprint.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	player		
Description:	Players can run.		
Trigger:	Press on the sprint button.		
Preconditions:	<ul style="list-style-type: none">● Player is moving forward.		
Post conditions:	<ul style="list-style-type: none">● The player can run fast.		
Normal Flow:	<ul style="list-style-type: none">● On pressing the sprint button, players start running fast.		
Includes:	The use case helps users to run.		
Frequency of Use:	When the user wants to run.		
Assumptions:	User should have some knowledge of fps games		
Notes and Issues:	Press the sprint button.		

Table 4.12: Settings:

Use Case ID:	UC-11		
Use Case Name:	Settings.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can change settings to its liking in the setting menu.		
Trigger:	Press on the setting button.		
Preconditions:	<ul style="list-style-type: none">The application should have launched correctly		
Post conditions:	<ul style="list-style-type: none">The change option functionality works properly.		
Normal Flow:	Upon pressing the button, the application should take the user to the setting screen.		
Alternative Flows:	<ul style="list-style-type: none">Users can change between playing games.		
Includes:	The use case helps the player to change the settings according to his liking.		
Frequency of Use:	When a user wants to change something.		
Assumptions:	It is assumed that the user understands the English and Urdu language.		
Notes and Issues:	Press the settings button.		

Table 4.13: Exit game:

Use Case ID:	UC-12		
Use Case Name:	Exit game.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can exit the game.		
Trigger:	Press on the exit button.		
Preconditions:	<ul style="list-style-type: none">● The application should be at the title screen or the game should have been paused.		
Post conditions:	<ul style="list-style-type: none">● The game process is killed.		
Normal Flow:	<ul style="list-style-type: none">● When the pause button is pressed the exit button appears,● Upon pressing the exit button, the application should kill all the objects and the bodies and it should end the gameplay● The application closes.		
Alternative Flows:	<ul style="list-style-type: none">● User exits the application by pressing the back or home button of the Android device.		
Includes:	The use case kills the game process.		
Frequency of Use:	When the user wants to close the game.		
Assumptions:	It is assumed that the user understands the English and Urdu language.		
Notes and Issues:	Press the Exit button.		

Table 4.14: Pause option:

Use Case ID:	UC-13		
Use Case Name:	Pause option.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can pause the game by pressing the pause button.		
Trigger:	Press the button to go to the pause menu.		
Preconditions:	<ul style="list-style-type: none">• The application should be in the main game section.		
Post conditions:	<ul style="list-style-type: none">• The game is running.		
Normal Flow:	<ul style="list-style-type: none">• When the player wants to pause the game or wants to interact with pause options.		
Includes:	The use case helps the user to pause the game.		
Frequency of Use:	When the user wants to pause the game.		
Assumptions:	It is assumed that the user understands the English and Urdu language.		
Notes and Issues:	Press the Pause option button.		

Table 4.15: Customize:

Use Case ID:	UC-14		
Use Case Name:	Customize.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can go to the customize menu from the running game.		
Trigger:	Press on the customize button.		
Preconditions:	<ul style="list-style-type: none">The application should be a running game.		
Post conditions:	<ul style="list-style-type: none">The application on its main menu.		
Normal Flow:	<ul style="list-style-type: none">When the user presses the customize button. The player goes to the customization section.		
Includes:	The use case helps to customize the player.		
Frequency of Use:	When the user wants to customize his character.		
Assumptions:	Users should have some knowledge of fps games.		
Notes and Issues:	Press customize button.		

Table 4.16: Buy a gun and costume:

Use Case ID:	UC-15		
Use Case Name:	Buy a gun and costume.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can go to the buy gun and costume menu from the running game.		
Trigger:	Press on the buy a gun and costume button.		
Preconditions:	<ul style="list-style-type: none"> • The application should be a running game. 		
Post conditions:	<ul style="list-style-type: none"> • The application on it's customize menu. 		
Normal Flow:	<ul style="list-style-type: none"> • When the user presses the buy a gun and costume button.. 		
Includes:	The use case helps to customize players.		
Frequency of Use:	When the user wants to buy skins for their gun or character.		
Assumptions:	Users should have some knowledge of fps games.		
Notes and Issues:	Press the buy gun and costume button.		

Table 4.17: Customize gun and costume:

Use Case ID:	UC-16		
Use Case Name:	Customize guns and costumes.		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	Player		
Description:	Players can go to the customized gun and costume menu from the running game.		
Trigger:	Press on the customize gun and costume button. <ul style="list-style-type: none">● The application should be a running game.		
Preconditions:	<ul style="list-style-type: none">● The application on it's customize menu.		
Post conditions:	<ul style="list-style-type: none">● When the user presses the customize gun and costume button.		
Normal Flow:			
Includes:	The use case helps to customize players.		
Frequency of Use:	When the user wants to equip skins for their gun or character.		
Assumptions:	Users should have some knowledge of fps games.		
Notes and Issues:	Press customize gun and costume button.		

Table 4.18: Zombie:

Use Case ID:	UC-17		
Use Case Name:	Zombie		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	NPC		
Description:	They are the NPC of the game who do actions according to the algorithm.		
Trigger:	When the player starts the game.		
Preconditions:	<ul style="list-style-type: none">● Player starts the game.		
Post conditions:	<ul style="list-style-type: none">● NPC can move, attack, chase and scream according to the algorithm.		
Normal Flow:	<ul style="list-style-type: none">● NPC can move, attack, chase and scream.		
Includes:	The use case helps to produce a fear environment for the player.		
Frequency of Use:	Whenever every user starts the game.		
Assumptions:	Developers should have some knowledge NPC while making fps games.		

Table 4.19: Chase:

Use Case ID:	UC-18		
Use Case Name:	Chase		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	NPC		
Description:	NPC chase when they see any player in their attack circle according to the algorithm.		
Trigger:	When the player comes near the zombies. <ul style="list-style-type: none">● Player is standing or walking near the NPC.		
Preconditions:	<ul style="list-style-type: none">● NPC chase the player.		
Post conditions:			
Normal Flow:	<ul style="list-style-type: none">● When an NPC sees any player in their chase circle according to the algorithm.		
Includes:	The use case helps to produce a fear environment for the player.		
Frequency of Use:	When an NPC sees any player in their attack circle according to the algorithm.		
Assumptions:	Developers should have some knowledge NPC while making fps games.		

Table 4.20: Attack:

Use Case ID:	UC-19		
Use Case Name:	Attack		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	NPC		
Description :	NPC attack when they see any player in their attack circle according to the algorithm.		
Trigger:	When the player comes near the zombies. <ul style="list-style-type: none">● Player is standing or walking near the NPC.		
Preconditions:	<ul style="list-style-type: none">● NPC attack the player.		
Post conditions:	<ul style="list-style-type: none">● When an NPC sees any player in their attack circle according to the algorithm.		
Normal Flow:			
Includes:	The use case helps to damage the player.		
Frequency of Use:	When an NPC sees any player in their attack circle according to the algorithm.		
Assumptions:	Developers should have some knowledge NPC while making fps games.		

Table 4.21: Sound:

Use Case ID:	UC-20		
Use Case Name:	Sound		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	NPC		
Description:	Npc make a loud sound to make a horror environment in the game.		
Trigger:	When the player starts the game.		
Preconditions:	<ul style="list-style-type: none">● Player is standing or walking.		
Post conditions:	<ul style="list-style-type: none">● NPC scream in horrific voices to make a horror environment in the game.		
Normal Flow:	<ul style="list-style-type: none">● To make a horror environment in the game.		
Includes:	The use case helps to produce a fear environment for the player.		
Frequency of Use:	When a user starts the game.		
Assumptions:	Developers should have some knowledge NPC while making fps games.		

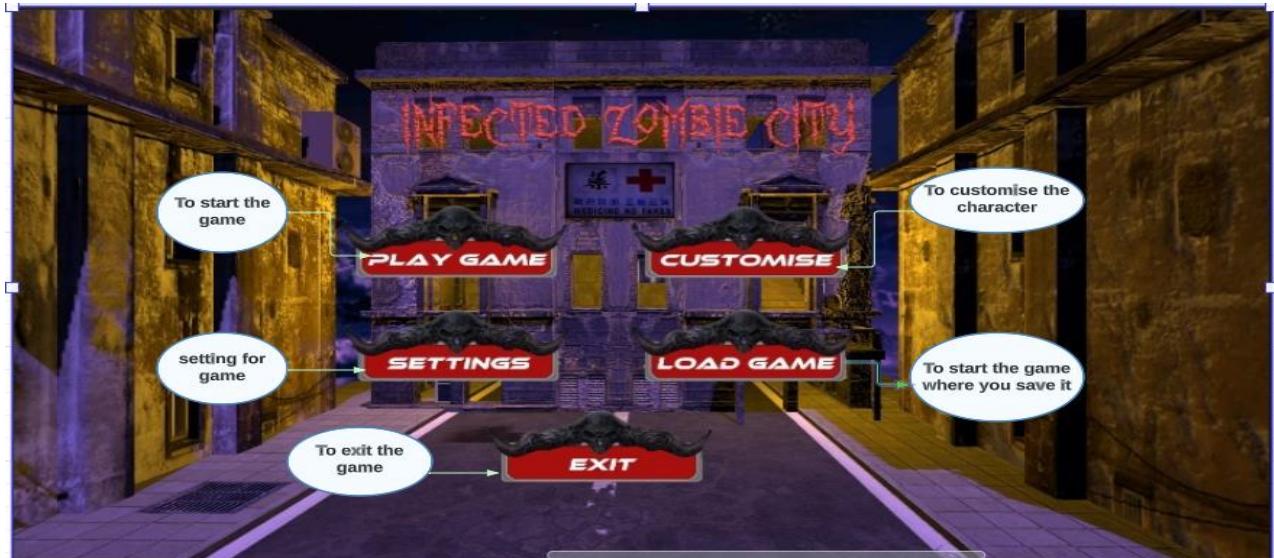
Table 4.22: Walk:

Use Case ID:	UC-21		
Use Case Name:	Walk		
Created By:	MUHAMMAD DANYAL	Last Updated By:	MUHAMMAD NOUMAN
Date Created:	10/6/2022	Last Revision Date:	10/6/2022
Actors:	NPC		
Description:	NPC can move slowly forward, backward and left, right according to the algorithm.		
Trigger:	When the player starts the game. <ul style="list-style-type: none">● Player starts the game.		
Preconditions:	<ul style="list-style-type: none">● NPC can move slowly forward, backward and left, right.		
Post conditions:	<ul style="list-style-type: none">● Movement according to the algorithm.		
Normal Flow:			
Frequency of Use:	Whenever a user starts the game.		
Assumptions:	Developers should have some knowledge NPC while making fps games.		

5. USER INTERFACE / SCREENS

5.1 Main Menu

This is the main Menu screen which will be shown to the player as soon as he launches the game after showing the main menu screen the player will have to press any button on the screen to perform a specific functionality. Each button has a different function and work.



P1: Main Menu of Infected Zombie City

- **"Play Game"**

After clicking the play button, another screen will appear. In which you have to select the gun which you are going to use in the game first.

- **"Customize"**

When you press the customize button, you will be redirected to the player customize menu where you can see multiple things like clothes, gun attachments, in-game currency and many more things which you can purchase using cash. You can also update your current avatar depending upon the style you want.

- **"SETTING"**

After pressing the Settings button, you will be shown a new screen. In that screen you will be shown different settings related to the game like Music setting, Sound setting, sensitivity setting, Control layout setting etc.

- **"Load Game"**

In this load game section you will be able to continue the game where you save the game and quit the game.

- **"Exit"**

If a player presses the exit option, he will be able to exit the game.

5.2 Gameplay Screen

This is the gameplay screen which will be shown to the player as soon as he selects play button game on the main menu screen the player will have to press multiple buttons on the screen to operate the character. Each button has a different function and works.



P2: gameplay of Infected Zombie City

- **Joystick:**

Let's talk about the joystick button, as shown in P2. It is used to control the character's movement. With the help of this button we can go left, right, forward and backward.

- **Auto:**

This button sets the weapon mode, for example if you want to fire a single bullet or perform a quick shot with the selected weapon.

- **Aim:**

Use this button to aim at the sights of your weapon. This means that if you don't want to shoot the weapon from the hip, you can use this function to open the sights and aim accurately.

- **Reload:**

With this button you can reload the weapon's ammunition or change the magazine.

- **Crouching, running and jumping:**

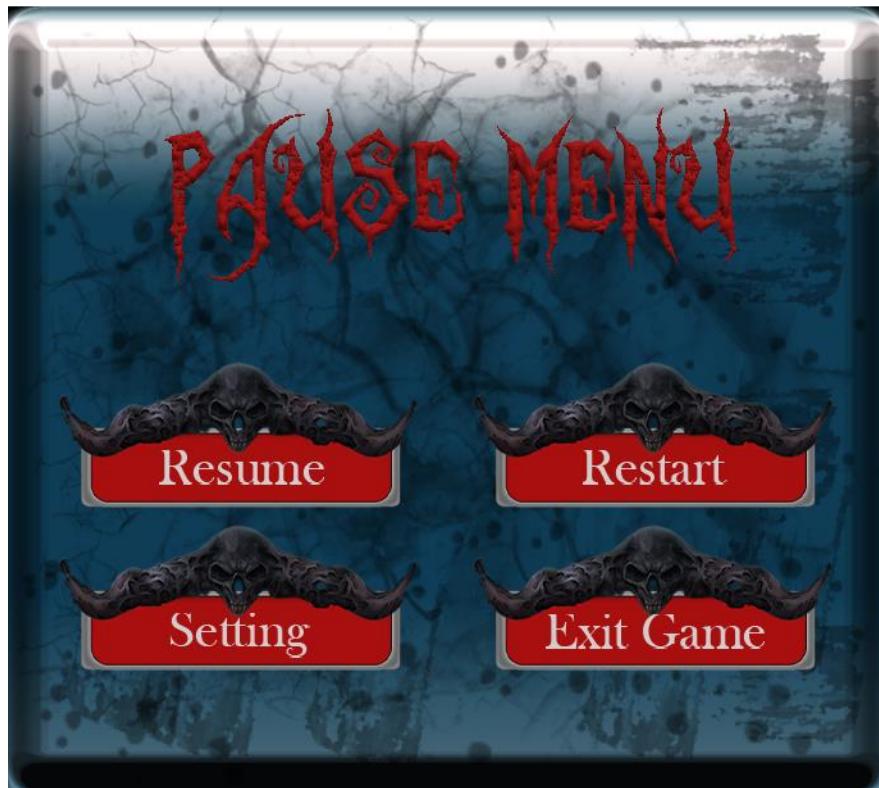
This button is used for various functions. First, crouching causes your character to stand up. Secondly, sprint is used for running and thirdly, jumping as the name suggests is used for jumping.

- **Map and Pause:**

The map's sole purpose is to provide the player with information about the character's environment so that he can navigate and enjoy the best gaming experience. The pause button is used to pause the game and also has some options of its own like restart and resume. and reset buttons.

5.3 Pause menu

This is the pause menu screen it targets when we press the pause button. It pause the game flow and the game starts when we resume the game.



P3: Pause menu of Infected Zombie City

- **Resume:**

This button is used to resume the game where it stopped and the player can enjoy the game again from the same position of the character.

- **Restart:**

This button is used to restart the game if you don't able to continue further.

- **Setting:**

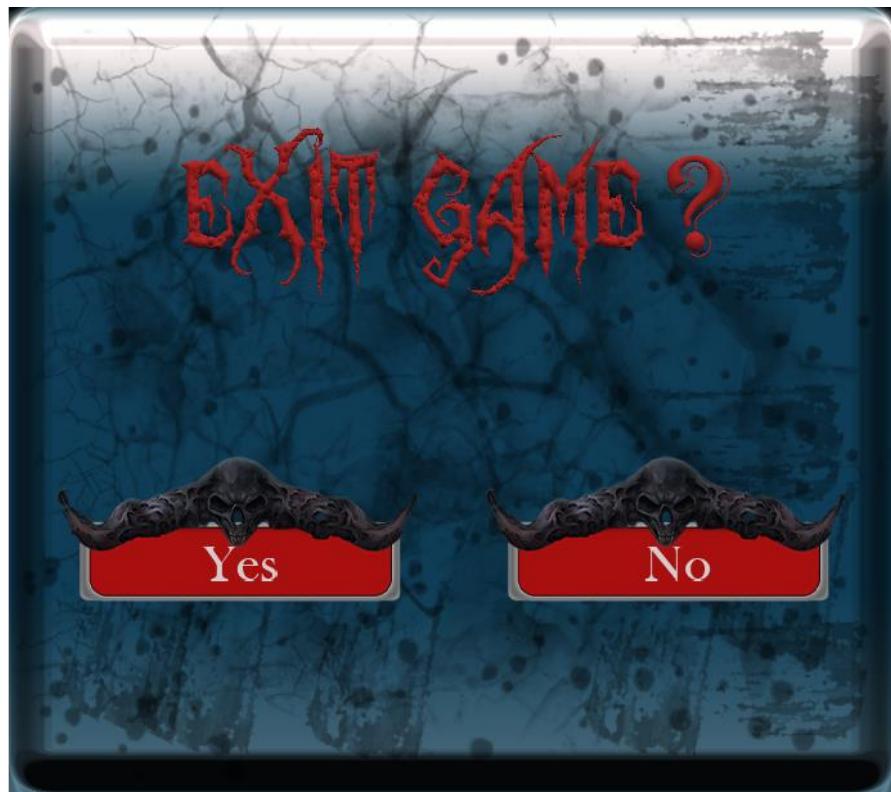
This button takes you to the settings menu. Where you can change settings according to your liking.

- **Exit game:**

This button is used to exit the game when you get bored or tired.

5.4 Exit menu

This menu shows up when you want to exit the game. It has two simple options: yes or no, choose whatever you like according to the situation.



P4: Exit menu of Infected Zombie City

- **Yes:**

If you want to exit the game use the yes option.

- **No:**

If you want to not exit the game use the no option.

5.5 Setting menu

This menu shows up when you want to change default game settings according to your liking.



P5: Setting of Infected Zombie City

- **Volume:**

This option is used to lower or increase the volume of the game. Just press + or – button you can adjust the volume to your liking.

- **Pointer size:**

This option is used to lower or increase the size of the aim pointer of the game. Just press + or – button you can adjust the aim pointer size to your liking.

- **Reset:**

This option is used to reset the change options to the default settings just press reset button.

- **Apply:**

This option is used to apply change options in the setting menu and exit the setting menu just press the apply button.

5.6 Level fail

This menu shows up when you die in the game. It also has two options: main menu and restart.



P6: Died of Infected Zombie City

- **Main menu:**

This is the main menu button that will take you to the main menu screen. Just press the main menu button.

- **Restart:**

This is the restart button that will restart the game from the start of level. Just press restart button.

5.7 Congratulation menu

This menu shows up when you complete the story of the game. It also has two options: main menu and restart.



P7: Congratulations of Infected Zombie City

- **Main menu:**

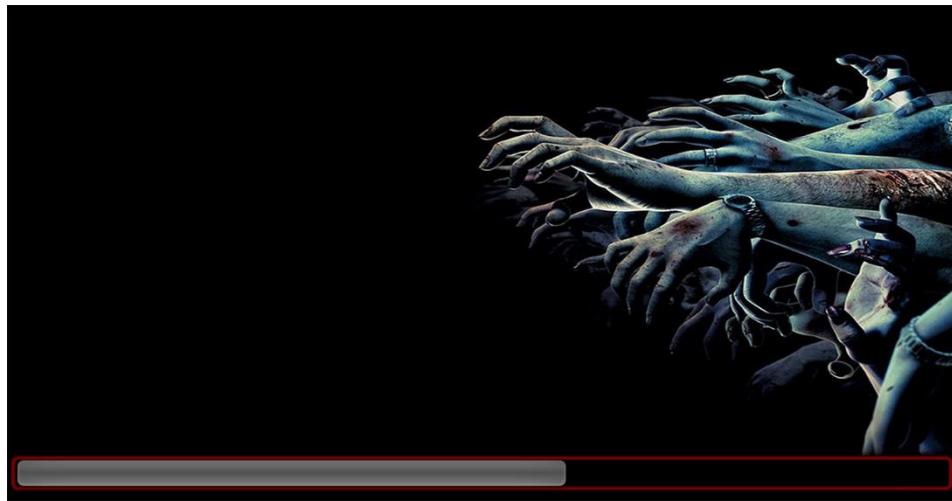
This is the main menu button that will take you to the main menu screen. Just press the main menu button.

- **Restart:**

This is the restart button that will restart the game from the start of level. Just press restart button.

6. LOADING SCREEN

This is the loading screen of our game, which appears whenever we enter the game screen or menu. This screen will show you how long you have to wait until the game is fully playable. Behind it, all functions for the game screen or menu are taken care of.



P8: loading screen of Infected Zombie City

7. LEVELS

Details about how many levels in your game?

There are three main missions in our game.

- First the beginner mission where the player comes to know how he is going to play the game. How he moves, shoots, jumps and sprints. After that the player gets the mission to find the main lab.
- Second mission is to fight the zombie boss and defeat him and ask the location of the vaccination.
- Third mission is to go to the location of the vaccination and fire it in the air so everybody gets better.

How mission will vary?

The mission will vary according to the story. First will be the beginner level difficulties but as we go near the lab the zombie behavior becomes vicious and dangerous and the damage also increases and then comes the boss level which will be very much difficult to complete because the boss is very powerful. Then comes the vaccination location mission. It will be hard too because the behavior of the zombies is very dangerous.

An example of a beginner level.

- First the beginner mission where the player comes to know how he is going to play the game. How he moves, shoots, jumps and sprints. After that the player gets the mission to find the main lab.

An example of a complex level.

The boss level will be very much difficult to complete because the boss is very powerful .Then comes the vaccination location mission it will be hard too because the behavior of the zombies is very dangerous.

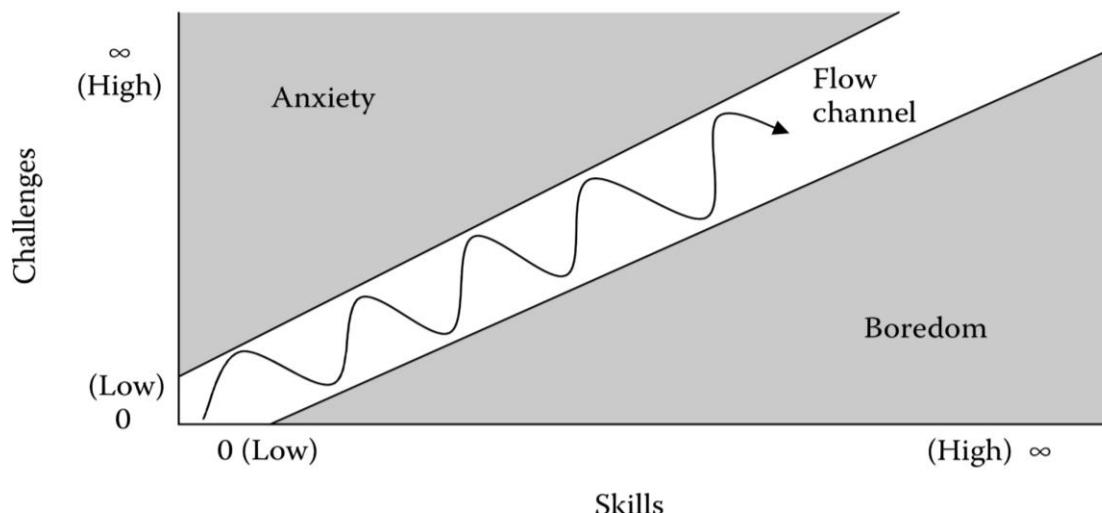


Figure 7.1: Complexity of levels

❖ First Time Experience

- First, the player needs to download and install the game on their device.
- Once the game has started, the player is presented with a tutorial that walks them through the basic mechanics of the game, such as: Movement, interaction with objects and use of skills.
- As the player progresses through the tutorial, they will learn how to navigate the game's menus and interface.
- Once the tutorial is completed, the player can start playing the game's missions.
- As you play, you will encounter various challenges and obstacles that will require you to use the skills learned in the tutorial to overcome them.
- The game may also introduce new mechanics and abilities as the game progresses, which the player must learn in order to progress.
- The player can also access the game's help menu to get more information and tips to help them with difficult parts of the game.
- The more experience the player gains, the more difficult challenges they can overcome and unlock new content.

8. ASSET LIST

- **Art:**

- The game is in 3D art.
- 3D Details is on Textures.

- **Environment Art:**

- All Associated Animations

- **UI - Icons, Buttons, Pop Ups:**

All design of UI - Icons, Buttons, and Pop Up on Photoshop

- **Game Objects:**

- **Buildings:**

Is taken from <https://www.incern.org/2022/06/city-slum-building-pack.html>

- **Characters:**

Zombie model is taken from <https://unityassets4free.com/zombies-pack-v2/>

Characters model is taken from

<https://www.mixamo.com/#/?page=1&query=solider&type=Character>

- **Weapons:**

Is taken from unity asset store and link is

<https://assetstore.unity.com/packages/3d/props/guns/modern-weapons-pack-14233>

- Particle FX

- Sound Effects

- Music

- **All Writing**

- **Quest Scripts**

Written by Danyal shah khan.

- **Story ‘Screenplay’**

Design on Blender

- Level Design Documents

9. IMPLEMENTATION DETAILS

9.1 Development Setup

The list of tools used in building the game Infected Zombie City are given below.

- **Unity hub and visual studio:**

Unity hub is used to set the game environment and visual studio is used for the code of the game.

- **Photoshop:**

Photoshop is used to design UI of the game like buttons, health etc.

- **Blender:**

Blender is used for designing 3D buildings and characters.

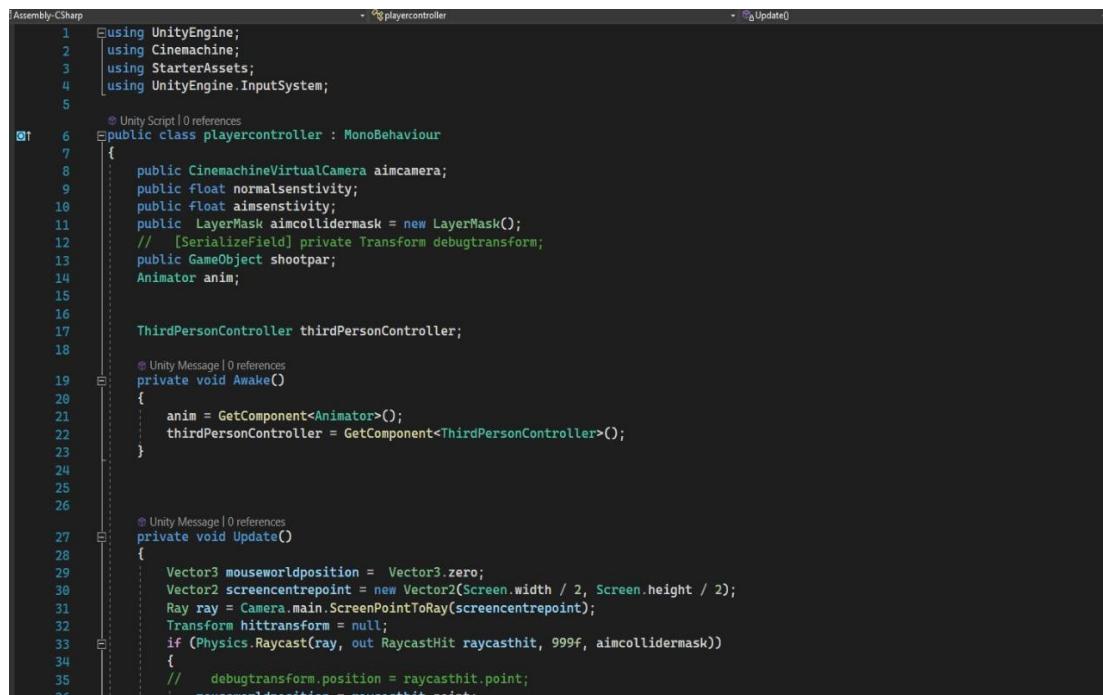
9.2 Deployment setup

For now, we deploy our game at azure servers and in the future, we will deploy it at play store.

9.3 Algorithms

These are some main algorithms of our game.

Player Controller:

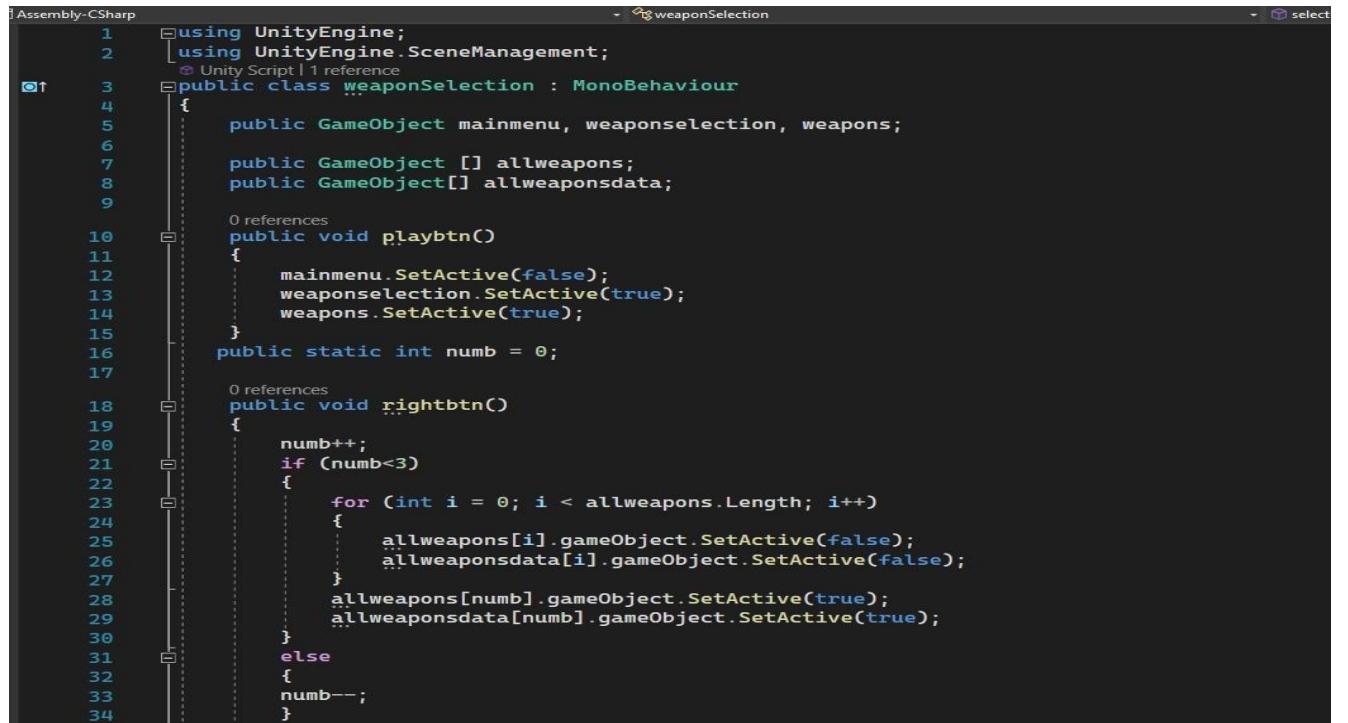


The screenshot shows a Unity C# code editor with the script 'playercontroller.cs' open. The code defines a MonoBehaviour class named 'playercontroller'. It includes fields for CinemachineVirtualCamera, float variables for normalsensitivity and aim sensitivity, a LayerMask for aimcollidermask, and components for shootpar and animator. It also includes fields for ThirdPersonController and Animator. The Awake() method initializes these components. The Update() method calculates mouse world position, performs a raycast from the camera to that point, and handles raycast hits. The code uses Unity's standard namespaces and includes comments explaining the logic.

```
Assembly-CSharp
1  Using UnityEngine;
2  Using Cinemachine;
3  Using StarterAssets;
4  Using UnityEngine.InputSystem;
5
6  @Unity Script | 0 references
7  Public class playercontroller : MonoBehaviour
8  {
9      public CinemachineVirtualCamera aimcamera;
10     public float normalsensitivity;
11     public float aimsensitivity;
12     public LayerMask aimcollidermask = new LayerMask();
13     // [SerializeField] private Transform debugtransform;
14     public GameObject shootpar;
15     Animator anim;
16
17     ThirdPersonController thirdPersonController;
18
19     @Unity Message | 0 references
20     Private void Awake()
21     {
22         anim = GetComponent<Animator>();
23         thirdPersonController = GetComponent<ThirdPersonController>();
24     }
25
26
27     @Unity Message | 0 references
28     Private void Update()
29     {
30         Vector3 mouseworldposition = Vector3.zero;
31         Vector2 screencentrepoin = new Vector2(Screen.width / 2, Screen.height / 2);
32         Ray ray = Camera.main.ScreenPointToRay(screencentrepoin);
33         Transform hittransform = null;
34         if (Physics.Raycast(ray, out RaycastHit raycasthit, 999f, aimcollidermask))
35         {
36             // debugtransform.position = raycasthit.point;
37             mouseworldposition = raycasthit.point;
38         }
39     }
40 }
```

Figure 9.1: Code snippet of player control

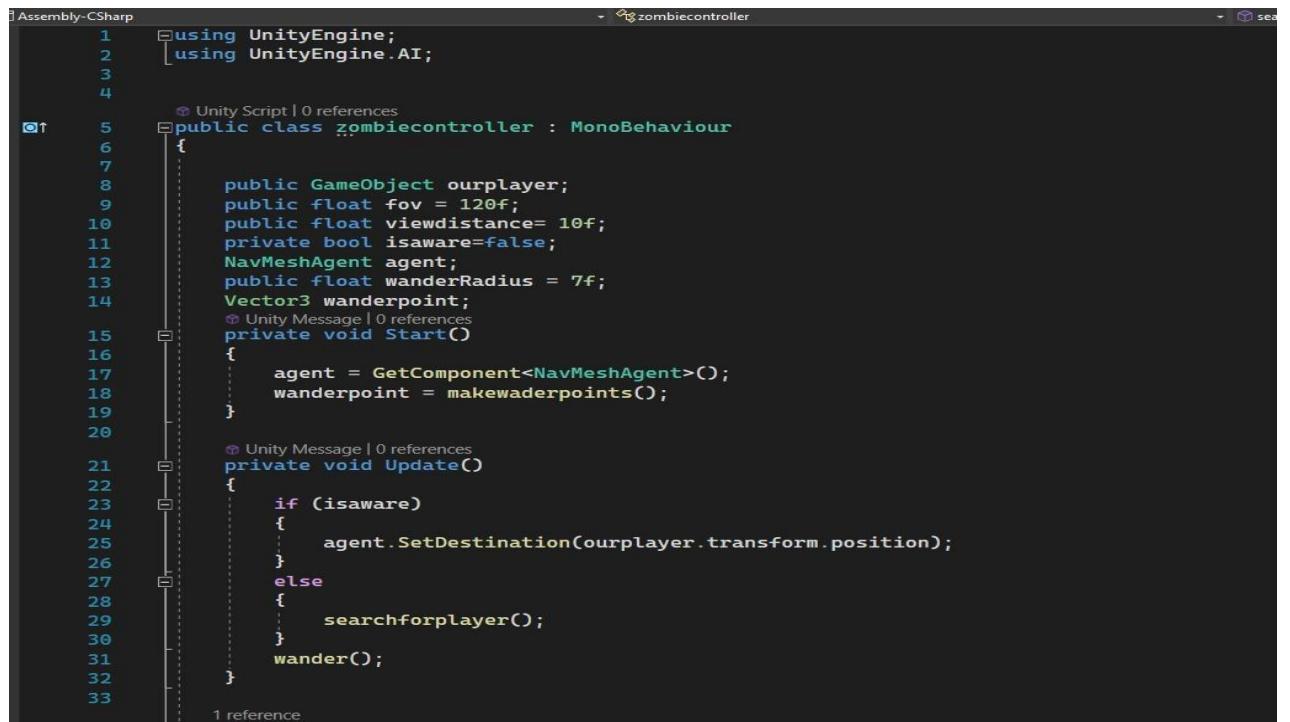
Weapon selection:



```
Assembly-CSharp weaponSelection
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3  public class weaponSelection : MonoBehaviour
4  {
5      public GameObject mainmenu, weaponselection, weapons;
6
7      public GameObject [] allweapons;
8      public GameObject[] allweaponsdata;
9
10     public void playbtn()
11     {
12         mainmenu.SetActive(false);
13         weaponselection.SetActive(true);
14         weapons.SetActive(true);
15     }
16     public static int numb = 0;
17
18     public void rightbtn()
19     {
20         numb++;
21         if (numb<3)
22         {
23             for (int i = 0; i < allweapons.Length; i++)
24             {
25                 allweapons[i].gameObject.SetActive(false);
26                 allweaponsdata[i].gameObject.SetActive(false);
27             }
28             allweapons[numb].gameObject.SetActive(true);
29             allweaponsdata[numb].gameObject.SetActive(true);
30         }
31         else
32         {
33             numb--;
34         }
35     }
36 }
```

Figure 9.2: Code snippet of weapon control

Zombie controller:



```
Assembly-CSharp zombiecontroller
1  using UnityEngine;
2  using UnityEngine.AI;
3
4
5  public class zombiecontroller : MonoBehaviour
6  {
7
8      public GameObject ourplayer;
9      public float fov = 120f;
10     public float viewdistance= 10f;
11     private bool isaware=false;
12     NavMeshAgent agent;
13     public float wanderRadius = 7f;
14     Vector3 wanderpoint;
15     private void Start()
16     {
17         agent = GetComponent<NavMeshAgent>();
18         wanderpoint = makewaderpoints();
19     }
20
21     private void Update()
22     {
23         if (isaware)
24         {
25             agent.SetDestination(ourplayer.transform.position);
26         }
27         else
28         {
29             searchforplayer();
30         }
31         wander();
32     }
33 }
```

Figure 9.3: Code snippet of zombie control

Git repository link given below:

<https://github.com/danyal880/Infected-Zombie-City>

9.4 Constraints

9.4.1 Assumptions

Things we assume will be true.

- First, you will get a user interface where you can select the button you want to select e.g. Play to start the game, settings, and exit, leaderboard and customization options.
- When you click the Play button, an interface will appear where you can select the weapon you want to play with.
- When you press the select button you will see the player and some buttons where you can see the joystick, jump, run and shoot.

9.4.2 System constraints

A constraint specifies how the system must operate or how it must be built

9.4.3 Restrictions

There will be no restrictions on the player; he can enjoy the game as much as he wants.

9.4.4 Limitations

The main software limitation is server stability.

10. TESTING

10.1 Extended Test Cases

Table 10.1: Play Game

Test Case Id: TC_1		Test Design By: Danyal									
Test Module Name: Play button to start game		Test Design Date: 12/6/2023									
Test Priority: High		Test Executed By: Nouman									
Test Name: To test the play button		Test Executed Date: 13/6.2023									
Description: Test the play button in game											
Pre-Conditions: The user Shall open our game and click on play button to start the game											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to main screen				Pass						
2	Click the play button		User moves to the next screen								
Post condition: The user move to next screen and select level											

Table 10.2: Gun selection

Test Case Id: TC_2		Test Design By: Danyal									
Test Module Name: Gun select button to select		Test Design Date: 12/6/2023									
Test Priority: High		Test Executed By: Nouman									
Test Name: To test the Gun selection button		Test Executed Date: 13/6.2023									
Description: Test the Gun selection menu in game											
Pre-Conditions: The user Shall open our game and click on play button and then select the gun to start the game											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to main screen				Pass						

2	Click the play button		User moves to the next screen			
3	Click the select button		User moves to the next screen			
Post condition: The user move to next screen and select level						

Table 10.3: Setting

Test Case Id: TC_3			Test Design By: Danyal							
Test Module Name: Setting button to do modification			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the setting button			Test Executed Date: 13/6.2023							
Description: A User accesses the setting options to modify his default settings.										
Pre-Conditions:										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Navigate to Setting button	Modification volume and pointer size.			Pass					
2	Click the setting button		User moves to the next screen a							
3	Click on the apply		Change settings will apply	Volume and pointer size change according to the user taste						
Post condition:										

Table 10.4: Customize

Test Case Id: TC_4		Test Design By: Danyal									
Test Module Name: Customize button to do modification in your gun and character		Test Design Date: 12/6/2023									
Test Priority: High		Test Executed By: Nouman									
Test Name: To test the customize button		Test Executed Date: 13/6.2023									
Description : A User accesses the customize option to modify his character.											
Pre-Conditions:											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					
1	Navigate to customize button	Modification of character costume and gun etc.			Pass						
2	Click the customize button		User moves to the next screen a								
3	Click on the modifications		character shall be modify	character is modified							
Post condition: users can buy and customize gun skins and clothes.											

Table 10.5: Buy gun and costume

Test Case Id: TC_5		Test Design By: Danyal									
Test Module Name: Buy gun and costume button to buy skin and costume		Test Design Date: 12/6/2023									
Test Priority: High		Test Executed By: Nouman									
Test Name: To test the Buy gun and costume button		Test Executed Date: 13/6.2023									
Description: Test the Buy gun and costume button in game											
Pre-Conditions: A User accesses the Buy gun and costume option to buy gun and costume for his character.											
Dependencies											
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes					

1	Navigate to buy a gun and costume.				Pass	
2	Click the Buy a gun and costume button		User shall move to the buy a gun and costume menu	User move to the buy a gun and costume		

Post condition: The user can buy a gun and costume.

Table 10.6: Customize gun and costume

Test Case Id: TC_6			Test Design By: Danyal							
Test Module Name: Customize gun and costume button to customize skins and clothes.			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the Customize gun and costume button			Test Executed Date: 13/6.2023							
Description: Test the test the Customize gun and costume button in game										
Pre-Conditions: launch the game and access the customize button than Customize gun and costume.										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Navigate to test the Customize gun and costume button				Pass					
2	Click the test the Customize gun and costume button		User shall move to the Customize gun and costume menu	User move to the Customize gun and costume						

Post condition: The user can customize guns and costumes.

Table 10.7: Pause option

Test Case Id: TC_7			Test Design By: Danyal							
Test Module Name: pause button to pause game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the pause button			Test Executed Date: 13/6.2023							
Description: Test the pause button in game										
Pre-Conditions: launch the game and access the pause button										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Navigate to pause button				Pass					
2	Click the pause button		User shall move to the pause menu	User move to the pause menu						
Post condition: The user can resume, restart and exit to the main menu.										

Table 10.8: exit button

Test Case Id: TC_8			Test Design By: Danyal							
Test Module Name: Exit button to end game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the exit button			Test Executed Date: 13/6.2023							
Description: Test the exit button in game										
Pre-Conditions: launch the game and access the exit button										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Navigate to exit button				Pass					
2	Click the exit button		Game shall be closed	Game is closed						
Post condition: The user exits the game.										

Table 10.9: player status

Test Case Id: TC_9			Test Design By: Danyal							
Test Module Name: check the player status in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the player status			Test Executed Date: 13/6.2023							
Description: Test the player status in game										
Pre-Conditions: launch the game and check the player status										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	Check the player status on running or damage		User status shall decrease as he get damage or run	User status decrease as he get damage or run in game						
Post condition: The user player status decrease as he get damage or run.										

Table 10.10: health

Test Case Id: TC_10			Test Design By: Danyal							
Test Module Name: check the player health in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the health status			Test Executed Date: 13/6.2023							
Description: Test the health status in game										
Pre-Conditions: launch the game and check the health status in game										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	Check the player health status on damage		User health status shall decrease as he get damage in game	User health status decrease as he get damage in game						

Post condition: The user player health status decrease as he get damage					

Table 10.11: stamina

Test Case Id: TC_11			Test Design By: Danyal							
Test Module Name: check the player stamina in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the stamina status			Test Executed Date: 13/6.2023							
Description: Test the stamina status in game										
Pre-Conditions: launch the game and check the stamina status in game										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	Check the player stamina status on running		User stamina status shall decrease as he run in game	User stamina status decrease as he run in game						
Post condition: The user player's stamina status decreases as he runs.										

Table 10.12: game mechanics

Test Case Id: TC_12			Test Design By: Danyal							
Test Module Name: check the game mechanics in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the game mechanics			Test Executed Date: 13/6.2023							
Description: Test the game mechanics in game										
Pre-Conditions: launch the game and check the game mechanics										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	Check the game mechanics in game		User can check the game	All game mechanics are applied						

			mechanics in game			

Post condition: The user can check game mechanics by sprinting, jumping and moving, firing.

Table 10.13: movement

Test Case Id: TC_13			Test Design By: Danyal							
Test Module Name: check the movement of the player in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the movement			Test Executed Date: 13/6.2023							
Description: Test the movement in game										
Pre-Conditions: launch the game and check the movement of the player in game										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check the movement of the player in game		Character shall move in direction of joystick	Character move in direction of joystick						
Post condition: The Character can move in the direction of the joystick.										

Table 10.14: sprint

Test Case Id: TC_14			Test Design By: Danyal							
Test Module Name: check the sprint button			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the sprint button in game			Test Executed Date: 13/6.2023							
Description: Test the sprint button in game										
Pre-Conditions: launch the game and check the character sprint or not.										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check the sprint button in game		Character shall run fast in game	Character run fast in game						

Post condition: The character runs fast in the game.					

Table 10.15: jump

Test Case Id: TC_15			Test Design By: Danyal							
Test Module Name: check the jump button			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the jump button in game			Test Executed Date: 13/6.2023							
Description: Test the jump button in game										
Pre-Conditions: launch the game and check the character jump or not.										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check the jump button in game		Character shall jump in game	Character jump in game						
Post condition: The character can jump in the game.										

Table 10.16: move

Test Case Id: TC_16			Test Design By: Danyal							
Test Module Name: check the move button			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the move button in game			Test Executed Date: 13/6.2023							
Description: Test the move button in game										
Pre-Conditions: launch the game and check the character move or not.										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check the move button in game		Character shall move in game	Character move in game						
Post condition: The Character can move in the direction of the joystick.										

Table 10.17: fire

Test Case Id: TC_17			Test Design By: Danyal							
Test Module Name: check the fire button			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the fire button in game			Test Executed Date: 13/6.2023							
Description: Test the fire button in game										
Pre-Conditions: launch the game and check the character fire or not.										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check the fire button in game		Character shall fire in game	Character fire in game						
Post condition: The character can fire in game.										

Table 10.18: zombie

Test Case Id: TC_18			Test Design By: Danyal							
Test Module Name: zombies exists in game			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the zombies exists in game			Test Executed Date: 13/6.2023							
Description: Test the zombies exists in game										
Pre-Conditions: launch the game and check zombies exists in game										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check zombies exists in game		zombies shall present in game	zombies are present in game						
Post condition: The user can find zombies in game.										

Table 10.19: chase

Test Case Id: TC_19			Test Design By: Danyal							
Test Module Name: check chase of zombie.			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the chase of zombies in game			Test Executed Date: 13/6.2023							
Description: Test the chase of zombie in game.										
Pre-Conditions: launch the game and check that zombie chasing the player										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check that zombies chasing the player		Zombies shall chase the player when he comes near the zombie.	Zombies chase the player when he come near the zombie.						
Post condition: Zombie chasing the player when he came near the zombie.										

Table 10.20: attack

Test Case Id: TC_20			Test Design By: Danyal							
Test Module Name: check attack of zombie			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the attack of zombie in game			Test Executed Date: 13/6.2023							
Description: Test the attack of zombies in the game.										
Pre-Conditions: launch the game and check that zombie attack the player										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check that zombie attack the player		Zombies shall attack the player when he comes near	Zombies attack the player when he comes near						

			the zombie.	the zombie.		
Post condition: Zombie attacks the player when he comes near the zombie.						

Table 10.21: sound

Test Case Id: TC_21			Test Design By: Danyal							
Test Module Name: zombie make sounds to make horror environment			Test Design Date: 12/6/2023							
Test Priority: High			Test Executed By: Nouman							
Test Name: To test the zombie make sounds to make horror environment			Test Executed Date: 13/6.2023							
Description: Test the zombie make sounds to make horror environment										
Pre-Conditions: launch the game and check that zombie make sounds to make horror environment										
Dependencies										
Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes				
1	Start the game				Pass					
2	check that make sounds to make horror environment		Zombie shall make sounds to make horror environment	Zombie make sounds to make horror environment						
Post condition: Zombie make sounds to make horror environment.										

Table 10.22: walk

Test Case Id: TC_22		Test Design By: Danyal
Test Module Name: check the walk of zombie		Test Design Date: 12/6/2023
Test Priority: High		Test Executed By: Nouman
Test Name: To test the walk of zombie in game		Test Executed Date: 13/6.2023
Description: Test the walk of zombie in game		
Pre-Conditions: launch the game and check the walk of zombie		
Dependencies		

Step	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail	Notes
1	Start the game				Pass	
2	check that zombie walk the player		Zombies shall walk around the specific place.	Zombies walk around the specific place.		
Post condition: Zombies walk around the specific place.						

10.2 Decision Table

Table 10.5

Decision/Branch	Condition(s)	Outcome(s)
Click the play button	User is on the main screen	User moves to the next screen
Click the setting button	User is on the main screen	User moves to the setting screen
Click on the customize	User is on the main screen	User moves to the customize screen
Click the pause button	User is on the game screen	User shall move to the option which will help you to go different screen
Click the Exit button	User is on the main screen	Game shall be closed
Click the buy gun and custom	User is on the customize screen	User moves to the buy gun and custom screen
Click customize gun and costume	User is on the customize screen	User moves to the customized gun and custom.
Click select button	User is on gun selection screen	User moves to main gameplay.
Character status	User is on main game play screen	Character health and stamina status change.
Character health	User is on main game play screen	Character health status change.
Character stamina	User is on main game play screen	Character stamina status change.
Character movement	User is on main game play screen	Character moves in different direction
Character sprint	User is on main game play screen	Character moves fast.
Character jump	User is on main game play screen	Character jump.

Character move	User is on main game play screen	Character moves
Character fire	User is on main game play screen	Character moves
Zombie	User is on main game play screen	NPC attack, chase, walk, sound and chase.
Zombie attack	User is on main game play screen	NPC attack the character.
Zombie walk	User is on main game play screen	NPC walks around the specific point
Zombie sound	User is on main game play screen	NPC make horror sounds.
Zombie chase	User is on main game play screen	NPC chase the character.

10.2.1 Code snippet

```

1 reference
public void OnBulletHit(PlayerBullet bullet, float damageMultiplier)
{
    if (currentState == deathState) return;

    hp -= bullet.damage * damageMultiplier;
    onHpChanged?.Invoke();

    if (bloodEffect != null)
    {
        bloodEffect.transform.position = bullet.transform.position;
        bloodEffect.Stop();
        bloodEffect.Play();
    }

    if (hp <= 0)
    {
        ChangeState(deathState);
    }
}

1 reference
public void OnGrenadeHit(AbstractGrenade grenade)
{
    ChangeState(deathState);
}

```

Figure 10.1: Code snippet of bullet hit

```

1 reference
private void ResumeGame()
{
    Time.timeScale = 1;

    IsGamePaused = false;
    Events.GameResumed.Call();
}

1 reference
private void FinishGame()
{
    IsGameFinished = true;
    Events.GameFinished.Call();
}

1 reference
private void Replay()
{
    Time.timeScale = 1;
    Events.GameReplay.Call();
    StartCoroutine(LoadScene(SceneManager.GetActiveScene().buildIndex));
}

1 reference
private void LoadHomeScene()
{
    Time.timeScale = 1f;
    Events.GameLoadHomeScene.Call();
    StartCoroutine(LoadScene(0));
}

```

Figure 10.2: Code snippet of resume game

```

Unity Message | 0 references
private void Awake()
{
    Events.PlayerDied += OnPlayerDied;
}

Unity Message | 0 references
private void Start()
{
    idleState = new IdleState(this);
    patrolState = new PatrolState(this);
    searchState = new SearchState(this);
    chaseState = new ChaseState(this);
    attackState = new AttackState(this);
    deathState = new DeathState(this);
    playerDiedState = new PlayerDiedState(this);

    animator = GetComponent<Animator>();
    characterController = GetComponent<CharacterController>();
    navmeshAgent = GetComponent<NavMeshAgent>();

    player = PlayerBehaviour.GetInstance();

    animator.applyRootMotion = false;
    navmeshAgent.autoBraking = false;
}

```

Figure 10.3: Code snippet of player died

10.2.2 Decision coverage table

Table 1: Main Screen Navigation and Play Button Coverage

Test Cases	Navigate to main screen	Click the play button	User moves to the next screen
TC_1	✓	✓	✓

Table 2: Setting Button Coverage

Test Cases	Navigate to setting button	Click the setting button	User moves to the setting screen
TC_2	✓	✓	✓

Table 3: Pause button Coverage

Test Cases	Navigate to Pause button	Click the Pause button	User shall move to the Pause menu
TC_3	✓	✓	✓

Table 4: Exit Button Coverage

Test Cases	Navigate to exit button	Click the exit button	User shall quit the game
TC_4	✓	✓	✓

Table 5: Click on the customize button Coverage

Test Cases	Navigate customize button	Click the customize button	User shall move to customize menu
TC_5	✓	✓	✓

Table 6: Click the buy gun and custom Coverage

Test Cases	Navigate buy gun and custom button	Click the buy gun and custom button	User shall move to buy gun and custom menu
TC_6	✓	✓	✓

Table 7: Click customize gun and custom Coverage

Test Cases	Navigate customize gun and custom button	Click the customize gun and custom button	User shall move to customize gun and custom menu
TC_7	✓	✓	✓

Table 8: Click select button Coverage

Test Cases	Navigate to select button	Click the select button	User shall have a selected gun in game
TC_8	✓	✓	✓

Table 9: Character status Coverage

Test Cases	Character health and stamina decrease	Character health and stamina decrease when he run or get damage	Character health and stamina shall decrease when he run or get damage
TC_9	✓	✓	✓

Table 10: Character health Coverage

Test Cases	Character health decrease when he get hit by zombies	Character health decrease when he get hit by zombies	Character health shall decrease when he get hit by zombies
TC_10	✓	✓	✓

Table 11: Character stamina Coverage

Test Cases	Character stamina decrease or increase according to his movement	Character stamina decrease when he run	Character stamina shall decrease when he run
TC_11	✓	✓	✓

Table 12: Character movement Coverage

Test Cases	Character move according to input command	Character move around	Character shall move, jump and sprint in the game
TC_12	✓	✓	✓

Table 13: Character sprint Coverage

Test Cases	Navigate to sprint button	Click the sprint button	character shall sprint in game
TC_13	✓	✓	✓

Table 14: Character jump Coverage

Test Cases	Navigate to jump button	Click the jump button	character shall jump in game
TC_14	✓	✓	✓

Table 15: Character move Coverage

Test Cases	Navigate to move joystick	Click the move joystick	Character shall move in the direction of joystick
TC_15	✓	✓	✓

Table 16: Character fire Coverage

Test Cases	Navigate to fire button	Click the fire button	character shall fire the gun
TC_16	✓	✓	✓

Table 17: Zombie Coverage

Test Cases	NPC attack ,chase ,sound and walk	When the game starts	NPC shall attack ,chase ,sound and walk
TC_17	✓	✓	✓

Table 18: Zombie attack Coverage

Test Cases	NPC attack the player	When Player go near NPC	NPC shall attack the player
TC_18	✓	✓	✓

Table 19: Zombie walk Coverage

Test Cases	NPC move around a specific point	When player start the game	NPC shall move around a specific point
TC_19	✓	✓	✓

Table 20: Zombie sound Coverage

Test Cases	NPC makes horror sound	When player start the game	NPC shall makes horror sound when game starts
TC_20	✓	✓	✓

Table 21: Zombie chase Coverage

Test Cases	NPC chase the player	When Player go near NPC		NPC shall chase the player
		When Player go near NPC	NPC shall chase the player	
TC_21	✓	✓	✓	✓

10.3 Traceability Matrix

10.3.1 RID vs UCID (requirements vs use cases)

Table 10.6: RID vs UCID (requirements vs use cases)

CID/ RID	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	
UC 1	✓		✓	✓																✓	✓
UC 2			✓	✓	✓															✓	✓
UC 3			✓	✓		✓	✓													✓	✓
UC 4			✓	✓		✓														✓	✓
UC 5			✓	✓				✓												✓	✓
UC 6	✓		✓	✓					✓	✓	✓	✓					✓	✓	✓	✓	✓
UC 7	✓		✓	✓					✓	✓	✓									✓	✓
UC 8	✓		✓	✓					✓	✓		✓								✓	✓
UC 9	✓		✓	✓					✓	✓	✓	✓								✓	✓
UC 10	✓		✓	✓					✓	✓		✓								✓	✓
UC 11	✓		✓	✓	✓								✓							✓	✓

UC 12	✓		✓							✓							✓	✓
UC 13	✓		✓								✓						✓	✓
UC 14	✓	✓	✓														✓	✓
UC 15	✓		✓								✓						✓	✓
UC 16	✓		✓								✓						✓	✓
UC 17	✓		✓								✓						✓	✓
UC 18	✓		✓									✓	✓	✓	✓	✓	✓	✓
UC 19	✓		✓									✓					✓	✓
UC 20	✓		✓									✓					✓	✓
UC 21	✓		✓										✓				✓	✓
UC 22	✓		✓														✓	✓

10.3.2 Prototypes (RID vs PID)

Table 10.7: Prototypes (RID vs PID)

RID/PID	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
R1	✓					✓	✓	
R2		✓	✓					
R3		✓				✓	✓	
R4	✓	✓	✓			✓	✓	
R5	✓							

R 6		✓																			
R 7		✓																			
R 8		✓																			
R 9		✓																			
R 10		✓																			
R 11		✓																			
R 12	✓		✓	✓																	
R 13	✓		✓	✓																	
R 14	✓																				
R 15		✓																			
R 16		✓																			
R 17		✓																			
R 18		✓																			
R 19	✓	✓																			
R 20		✓																			

10.3.3 Test Cases (RID vs TID)

Table 10.8: Test Case (RID vs TID)

RID/ TID	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9	T 10	T 11	T 12	T 13	T 14	T 15	T 16	T 17	T 18	T 19	T 20	T 21	T 22
R1	✓	✓	✓	✓	✓	✓			✓													
R 2								✓	✓													

10.3.4 Coverage (UCID vs TID)

Table 10.9: Coverage (UCID vs TID)

11. RESULTS/OUTPUT/STATISTICS

11.1 %completion

Fulfilled requirements: 20

Total requirements: 20

$$\% \text{completion} = (20/20) * 100 = 100\%$$

11.2 %accuracy

Implemented requirements: 20

Total requirements: 20

$$\% \text{accuracy} = (20/20) * 100 = 100\%$$

11.3 %correctness

Tested requirements: 20

Total requirements: 20

$$\% \text{accuracy} = (20/20) * 100 = 100\%$$

12. CONCLUSION

In summary, our goal was to create a game that not only educates society about the necessary precautions during a pandemic but also provides a fun and enjoyable learning environment. Additionally, our goal was to foster connections between people through our game's multiplayer mode.

By combining education and entertainment, we hope to raise awareness of pandemic safety measures in an engaging way. Our multiplayer mode encourages social interaction and collaboration, fostering a sense of togetherness even during times of physical distancing.

With this innovative approach, we want to make a positive contribution to public health awareness and community connectivity and ultimately help society to cope with pandemic situations more effectively and resiliently.

13. FUTURE WORK

We have many plans for future development. For example, adding new maps and locations, additional characters and playable factions, more types of zombies, special events and challenges, new weapons, equipment, stories and missions. We also thought about adding multiplayer modes like Team Survival Rounds and Manhunt. We are also fixing future bugs and making the game more balanced for the gaming experience. The most important thing is that we will also release this game for PC.

14. BIBLIOGRAPHY

Use IEEE or ACM format for citations

14.1 Other References

- World War Z.
- Last Of US
- Resident evil
- World War Z

15. APPENDIX

15.1 Glossary of terms

FPS	First person shooter
3D	3-dimension view
LWRP	Lightweight render pipeline
SRS	Software requirement specification
NPC	Non-player character

15.2 Pre-requisites

Requirements (Minimum):

Android Baseband & Kernel V7.1, 2-4 GB RAM, Memory 300 MB.

Final Year Project Report

Project Name: AI Based Life Saving System



Project Advisor:

Mahmood Hussain

Submitted by:

Daniyal Javaid (f2019266078)

Umair Malik (f2019266096)

Muhammad Husnain (f2019266076)

Ayesha Muhktar (f2019266093)

Session 2019-23

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

We (Umair Malik-F2019266096, Daniyal Javaid-F2019266078, Muhammad Husnain-F2019266076, Ayesha Muhktar-F2019266093) would dedicate our work to our respected supervisor who assisted us and taught us about the things which we should add in our system and along with we dedicate our work to the people who are suffering from the blood disease such as thalaseemia and we would dedicate our work to the humanity.

Final Approval

- **Head of Department**

Department of Informatics & Systems
School of Systems & Technology
UMT Lahore

- **Director (Final Year Projects-CS)**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Supervisor**

Department of Informatics & Systems.
School of Systems & Technology
UMT Lahore

- **Co-Supervisor**

Acknowledgment

I would want to convey my gratitude to everyone who helped with this project in some form and who gave unshakable support from the project's inception. Perseverance, hard effort, patience, and the right direction are the four ingredients of a successful undertaking. I'd want to thank Sir Mahmood Husnain, who has always been an inspiration to me.

He helped ensure that the project was completed successfully. I'd want to take this chance to thank Muhammad Husnain for being so kind as to come up with an idea and act as our project manager. In addition, I thank Umair Malik, Ayesha Mukhtar, and Daniyal Javaid for their assistance.

I appreciate your help with the project documentation. I am grateful to the lecturers in the computer science department for sharing their knowledge and providing advice on how to make our project better. Finally, I want to thank everyone who has supported us, including our friends, family, and extended family as well as anybody else who has helped in any way to make our initiative a success.

Project Title:

AI Based Life Saving System

Objective :

Provision of blood donor using web portal and Mobile App

Undertaken by:

- Daniyal Javaid (f2019266078)
- Umair Malik (f2019266096)
- Muhammad Husnain (f2019266076)
- Ayesha Mukhtar (f2019266093)

Supervised by:

Sir Mahmood Husnain

Starting Date:

03/10/2022

Completion Date:

20/05/2023

Tools Used:

React JS,Django, React native,Neo4j

Operating System:

Windows and Android

Documentation :

Plagiarism Report

Umair

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	5%
2	Submitted to Myongji University Graduate School Student Paper	4%
3	Submitted to Pontificia Universidad Catolica del Ecuador - PUCE Student Paper	3%
4	Submitted to University of London External System Student Paper	1%
5	suraj.lums.edu.pk Internet Source	1%
6	www.coursehero.com Internet Source	1%
7	escholar.umt.edu.pk:8080 Internet Source	1%
8	Submitted to Asia Pacific International College Student Paper	<1%

Abstract:

The proposed project aims to develop a web portal and mobile application for finding blood donors in emergency situations. The main motivation behind the project is to help humanity and address the issue of the public's inability to provide patients with blood in a timely manner due to a shortage of donors. The problem statement includes a lack of proper system to find donors, superstitions, and lack of awareness about blood donation. The objectives of the project include a search bar for finding donors, a donor registration form, and automated messages in case of emergency, collaboration with NGOs, and collaboration with hospitals. The target customers are people in need of blood donations and the stakeholders include the administrator, donors/attendants, and employers. The system affects patients, NGOs, and hospitals and has dependencies on the Google Maps API and mobile messaging. There are 60,000 fewer teenage donors today than there were ten years ago, according to a 2011 BBC World Blood Day Study., with 2-3 Out of 10 Needing blood of the same blood type and finding blood nearby and readily available. An area that is getting very difficult to find. Most People died because they didn't donate blood in time. That's why I decided to create a web portal and mobile application where anyone can find nearby blood donors. This makes access to donors easier and allows people to contact donors as soon as possible to save lives. In the future, we plan to include NGOs in this web portal and connect hospitals to the portal. As a result, we create the right platform that can help people when they need blood in an emergency of any kind. The website and application is using Django and neo4j as back end and react and react native for front ends of both website and the application for a clean and structured look

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
Draft	Umair	Initial draft created for distribution and review comments	20-11-2022
Preliminary	Husnian	Second draft incorporating initial review comments, distributed for final review	05-01-2023
Final	Umair	First complete draft, which is placed under change control	20-04-2023
Revision 1	Daniyal	Revised draft, revised according to the change control process and maintained under change control	02-05-2023

CONTENTS

CONTENTS.....	1
DEFINITIONS AND ACRONYMS.....	3
LIST OF FIGURES.....	4
LIST OF TABLES.....	5
1. INTRODUCTION.....	6
1.1 MOTIVATIONS.....	6
1.2 PROJECT OVERVIEW.....	6
1.3 PROBLEM STATEMENT.....	6
1.4 OBJECTIVES.....	7
2. DOMAIN ANALYSIS.....	8
2.1 CUSTOMER.....	8
2.2 STAKEHOLDERS.....	8
2.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT.....	8
2.4 DEPENDENCIES/ EXTERNAL SYSTEMS.....	8
2.5 REFERENCE DOCUMENTS.....	9
2.5.1 <i>Related Projects</i>	9
2.5.2 <i>Feature Comparison</i>	9
3. REQUIREMENTS ANALYSIS.....	10
3.1 REQUIREMENTS.....	10
3.1.1 <i>Hardware/Software Requirements: -</i>	12
3.2 LIST OF ACTORS.....	12
3.3 LIST OF USE CASES.....	13
3.4 SYSTEM USE CASE DIAGRAM.....	14
3.5 EXTENDED USE CASES.....	15
3.5.1 <i>Use Case donor registration</i>	15
3.5.2 <i>Use Case update stock</i>	16
3.5.3 <i>Use Case donation requests</i>	17
3.6 USER INTERFACES (MOCK SCREENS).....	18
4. DATA FLOW DIAGRAM (OPTIONAL).....	24
4.1 DATA FLOW DIAGRAM LEVEL 0.....	24
4.2 DATA FLOW DIAGRAM LEVEL 1.....	25
5. SYSTEM DESIGN.....	26
5.1 SYSTEM ARCHITECTURE DIAGRAM.....	26
5.2 CLASS DIAGRAM.....	27
5.3 SEQUENCE DIAGRAMS.....	28
5.4 COLLABORATION DIAGRAMS.....	30
5.5 ERD.....	31
5.6 DATA DICTIONARY.....	32
6. IMPLEMENTATION DETAILS.....	33
6.1 DEVELOPMENT TECHNOLOGIES.....	33

6.2	DEPLOYMENT SETUP.....	33
6.3	ALGORITHMS.....	34
6.4	CONSTRAINTS.....	38
6.4.1	<i>Assumptions</i>	38
6.4.2	<i>System constraints</i>	38
6.4.3	<i>Restrictions</i>	39
6.4.4	<i>Limitations</i>	39
7.	TESTING.....	40
7.1	EXTENDED TEST CASES.....	40
7.2	DECISION TABLE.....	43
7.2.1	<i>Code snippet</i>	43
7.2.2	<i>Decision coverage table</i>	48
7.3	TRACEABILITY MATRIX.....	49
7.3.1	<i>RID vs UCID (requirements vs use cases)</i>	49
7.3.2	<i>Prototypes (RID vs PID)</i>	49
7.3.3	<i>Test Cases (RID vs TID)</i>	50
7.3.4	<i>Coverage (UCID vs TID)</i>	50
8.	RESULTS/OUTPUT/STATISTICS.....	51
8.1	%COMPLETION.....	51
8.2	%ACCURACY.....	51
8.3	%CORRECTNESS.....	51
9.	CONCLUSION.....	52
10.	FUTURE WORK.....	53
11.	BIBLIOGRAPHY.....	54
12.	APPENDIX.....	55
12.1	GLOSSARY OF TERMS.....	55
12.2	PRE-REQUISITES.....	55

Definitions and Acronyms

e.g

Acronym	Definition
UMT	University of Management and Technology
POS	Point of Sale
BDMS	Blood donation management system
GUBBMS	Ganpat university blood bank management system
SMIBDMS	Sindh Madressatul Islam University blood donation management system

Table 1: table of acronyms and definitions

List of Figures

Figure 1: sample use case diagram with explanation	14
Figure 2 website register page	18
Figure 3 website Login Page	19
Figure 4 website blood request Form	20
Figure 5 website request for donor	21
Figure 6 website welcome page	22
Figure 7 Profile	23
Figure 8 Login/Register	23
Figure 9 mobile GUI	23
Figure 10 DFD level 0	24
Figure 11 DFD level 1	25
Figure 12 System Architecture Diagram	26
Figure 13 Class Diagram	27
Figure 14 Sequence Diagram (donor)	28
Figure 15 Sequence Diagram (Attendant)	29
Figure 16 collaboration diagram	30
Figure 17 ERD	31
Figure 18 Donor Finding	34
Figure 19 Order Blood	35
Figure 20 Order Blood	36
Figure 21 Order Blood	37
Figure 22 User Authentication	38
Figure 23 Login Screen	43
Figure 24 Login Screen	44
Figure 25 Password Verification	44
Figure 26 Password Verification	45
Figure 27 Maps Integration	45

List of Tables

Table 1: table of acronyms and definitions	3
Table 2: list of stakeholders	8
Table 3 feature comparison	9
Table 4 Basic Requirements	13
Table 5 Basic Requirements	13
Table 6 Extended use case (1)	16
Table 7 Extended use case (2)	17
Table 8 Extended use case (3)	18
Table 9 Test Id 01	35
Table 10 Test Id 02	36
Table 11 Test Id 03	37
Table 12 Test Id 04	38
Table 13 Decision Table	38
Table 14 RID/UCID	39
Table 15 PID/FRID	39
Table 16 TID/FRID	40
Table 17 UCID/TID	40
Table 18 Glossary	45

1. INTRODUCTION

Now a day's health issues are very common in our society and people are suffering from different kinds of health problems. Sometimes those health issues are so serious that people need blood immediately while their patients are in operation theatre. According to the survey in BBC 2011 to mark World Blood Day there are 60,000 fewer young donors than a decade ago and out of every 10 people 2 to 3 people needs blood of the similar blood group and it becomes very difficult to find blood instantly in the nearby areas. Mostly people lost their lives because they don't get blood on time, so we have decided to make a web portal and mobile application where anyone can find blood donors in their nearby areas. This will make it access to the donors easily and people can reach out the donor as soon as possible and save lives. In future, we will add the NGOs to this web portal and we will also link hospitals to the portal too. In result we will make a proper platform where we will be able to connect people which facilitate people if they need blood in any kind of emergency.

1.1 Motivations

Main reason to choose this project was simply to help humanity in any way possible and in our view blood donation management was the best way to serve humanity as donating blood is considered a bad omen and people avoid to donate blood our system will mainly focus on providing blood to the needy and also give guidance to the people who don't donate blood as those making this project have been facing the fact that finding blood in case of emergency is not an easy tasks that's why we have dedicated to build this project for them,

- Thalassemia patients
- Hospitals
- Immediate response groups

1.2 Project Overview

Our proposed concept, "A.I. based lifesaving system," addresses the urgent issue of the public's inability to provide patients with blood in a timely manner due to a shortage of donors. According to Dr. Fauzia Saeed, in charge of the Punjab Red Crescent Society's blood bank, 20 out of 1000 people do not provide blood. We are hoping to locate more donors, and our system is focused on giving blood to Pakistan's almost 10 million thalassemia sufferers. Moreover, our system will succeed

- blood provision
- Advocacy for better health
- Awareness for different blood diseases
- fulfill SDGs

1.3 Problem Statement

- No proper system to find donors on time
- People tend not to donate blood due to superstitions
- Blood banks not creating a proper system to provide blood

- hospitals either don't have proper blood banks or they sale blood bags for their own benefits
- no awareness to the people that blood donation is a healthy process and everyone should donate blood every 4 months for better health and to help others

1.4 Objectives

1. Website and the app will have a search bar to search for the donors nearby by giving location access, contact info and their desired blood type
2. The website will also have a donor registration form where different donors will give out all of their personal information like name, NIC, Contact info, address and blood types
3. Incase of emergency the users looking for donors when they will search for blood the AI system will generate automated messages to all the users contacts and nearby blood donors will be shown on the portal with their contact info's
4. Collaborating with NGOs to provide blood donations so they can give blood to thalassemia patients monthly
5. Collaborating with hospitals so they can access our system and share their blood bank and donation data with us for better efficacy.

2. DOMAIN ANALYSIS

2.1 Customer

In case of a blood donation emergency the person in need looks for donors our system will provide donors to the person in need and there are many NGOS that work in social welfare that also provide people with blood donors and many other facilities to help the needy and our system will also give access to different hospitals who mainly call for blood donations

2.2 Stakeholders

Stakeholder	Role in System
Administrator	He is responsible for handling backend work regarding database and also he is responsible for generating reports. If system faces any kind of bug, then he is going to fix the system
Donor / attendant	Donor can be of any type who wants to donate blood of any group and the system will find for him the attendant who the person in need of a specific blood group at a specific location
Employer	Employer is responsible for managing all the donors / attendants enquiries, it is just like a helpdesk which will be provided to the donors / attendants.

Table 2: list of stakeholders

2.3 Affected Groups with social or economic impact

This system affects the people all over Pakistan that need blood donations or are currently searching for donors in their nearby locations.

- Patients
A person that has been suffered from an injury that needs blood
- NGOS
Certain institutes that monthly collect blood for related deceases that require monthly blood transfusions like blood cancer or thalassemia.
- Hospitals
Hospitals that have blood banks that store blood in case of emergencies

2.4 Dependencies/ External Systems

Our proposed system has dependency to google maps API we need user location access to give attendants their locations and we need to access mobiles messages so we can generate messages for donations through our app

- Google maps api
- Message inbox access
- Contacts inbox access
- Cloud servers for hosting the application

2.5 Reference Documents

We had studied through different resources about the related project people done before and through different organization in which people are still working

2.5.1 Related Projects

List of all the documents/ projects that you have looked up as reference material for this project along with their links/references.

1. SMI blood donation management system (BDMS)

Developed by SyedMoiz Shah Working of this project was observed from studocu.com no relevant documentation was available

2. GU Blood bankManagement System (GUBBMS)

Developed by RinkuS.Patel. the working of this management software was observed from academia.com no relevant documentation was available.

2.5.2 Feature Comparison

Table 3 feature comparison

Sr No.	Comparison Feature	GUBBMS	SRS,BDMS	Remarks
1	Donor management	GUBBMS had a proper database for static and logged in donor management	BDMS does not support feature of static data in database	Using the GUBBMS feature we will create static donor data and also give relational data from contacts of our users
2	Location searching	GUBBMS gives static location of the donors	BDMS does not support this but gives contact info	Using the GUBBMS feature and BDMS feature we will give both contact info and dynamic location using GPS
3	Request management	GUBBMS has donor request after login	BDMS has static data of donors for information	Using GUBBMS donor request feature we will create and emergency donor request without login
4	Admin panel	GUBBMS has a admin panel authorized person and is difficult to access	BDMS has an admin panel for static records handling	Using GUBBMS feature we will create a Django admin which be easy to manage all data

3. REQUIREMENTS ANALYSIS

3.1 Requirements

Functional Requirements

- **Signup**
The system allows donors to store their personal information so that public can easily access the required blood group.
- **Online Appointment**
The system allows public to access and reserve their required blood on time in case of any emergency.
- **Blood Request**
The system allows users to request their desired blood type for which the cross match can be carried out.
- **No Installation**
The web application can be simply access by the browser. It doesn't need any installation or up gradation whereas the mobile application need installation and can be upgraded to its new version for its steady use.
- **Donor Records**
The data retrieved from signups of donors and from the mobile application will be stored in our database.

Non-Functional Requirements

- **Availability**
System is a web and mobile based and it will be available on the internet easily.
- **Security**
System allows public to sign up and access it but only authorized personals are allowed to access the all data.
- **Usability**
System can be easily accessible by the person who have internet on their smart phones.
- **Performance**
System will perform efficiently because of its advanced technologies and advance frameworks using AI and will be much optimized.
- **Modifiability**
Authorize Personals can access the whole system and they will be able to modify the system in case of any bugs or crashes.

System:

- **React Js. (Front End)**
 - React Js is a front-end JavaScript framework that is free and open-source for creating user interfaces based on UI components. It saves time for developers as they don't have to write various codes for the same features. It was developed by Facebook, who also use its platform to create Instagram.[1]

- There are some other frameworks of frontend like Angular, Vue or Bootstrap but the best of them is React Js because it is stronger framework because of its ability to break down the complex interface and allow user to work on individual components.
 - We will use React Js in this project front-end in which we will build an interactive user interface and website with JavaScript code and we will connect it with the mobile app.[2]
- **Django (Back End)**
 - It is a python based back-end framework that enables the secure development and it maintain website. It takes care much hassle in web development and it is used for the fast and clean web development.
 - We also have different another framework like node js, express, larval and ruby. All have their own specification and its own characteristics but the main reason why we are using Django is that it uses python and it will help us by using AI frameworks for our databases and it somehow much efficient.
 - In our website we will be using Django in such way that we will code the backend in python and after that we will use the AI technology in our backend which will connects the data base in such a way that it can take some decisions from its own and also helpful for the other AI features. [3]
- **React Native (Mobile App)**
 - React native is advance mobile development framework which uses JavaScript we are using it because we will be able to make android and iOS based mobile apps using same code and we will easily sync it with website.
 - We have many other frameworks like flutter, ionic, Rho Mobile and many more but we are using React Native because it is quite faster and flexible than other along with it we are using React Js in front end so it is better to use react native for mobile app development so we can easily sync and integrate the both website and app.
 - In this we will work on the react native in such way that we will connect the mobile app and through database we will share the data. In mobile application we will also allow user to find the people nearby with the help of GPRs [4]
- **Neo4j (Database)**
 - Neo4j is the database which is used to build the graph base relations. In neo4j we have the nodes and properties instead of rows and columns. As in our Project we have decided to use the graph database with the AI technology so for this Neo4j is best database which we can use.
 - We also have different databases type which we can use in our project like relational data base which consist of tables. In which we have like Mango DB, NoSQL, SQL but if we compare the relational and graph data base so for this the graph is much better because it much faster, more reliable and more efficient.

- We will use Neo4j as our project database in which we connect the data base with such a way that the mobile app and the website have the same database connected with each other and both will save their data and get their required data from the same database. Data base will be function in that way that it can work more efficient using AI technology and it will retrieve the data from the users using AI

3.1.1 Hardware/Software Requirements: -

Table 4 Basic Requirements

Required Ram	500 MB or more
Processor Type	Pentium IV or above
Operating System	Windows 7,8,10 , android or ios
Input Device Required	Keyboard, Mouse, touchpad
Output Device Required	Monitor, LCD, LED, basic smartphones
Front End	React js React native
Back End	Django Neo4j

Table 5 Basic Requirements

3.2 List of Actors

The system boundary and list all actors with the use cases are given below:

Administrator:

He is responsible for handling backend work regarding database and also he is responsible for generating reports. If system faces any kind of bug, then he is going to fix the system

Donor / attendant:

Donor can be of any type who wants to donate blood of any group and the system will find for him the attendant who the person in need of a specific blood group at a specific location

Employer:

Employer is responsible for managing all the donor's / attendants enquiries, it is just like a helpdesk which will be provided to the donors / attendants.

3.3 List of use cases

Login and logout: to get personal details of donors and store in database

Change password: to change password if forgotten

Blood requests near donor: to give donor blood donations near them

Donor finder: gives details of donor to attendant

Blood request: create blood request for applications so donors can donate in emergency

Donor current location: gives current location to the attendant of donor

Manage user full application: manages all the application if any error occur admin solves

3.4 System use case diagram

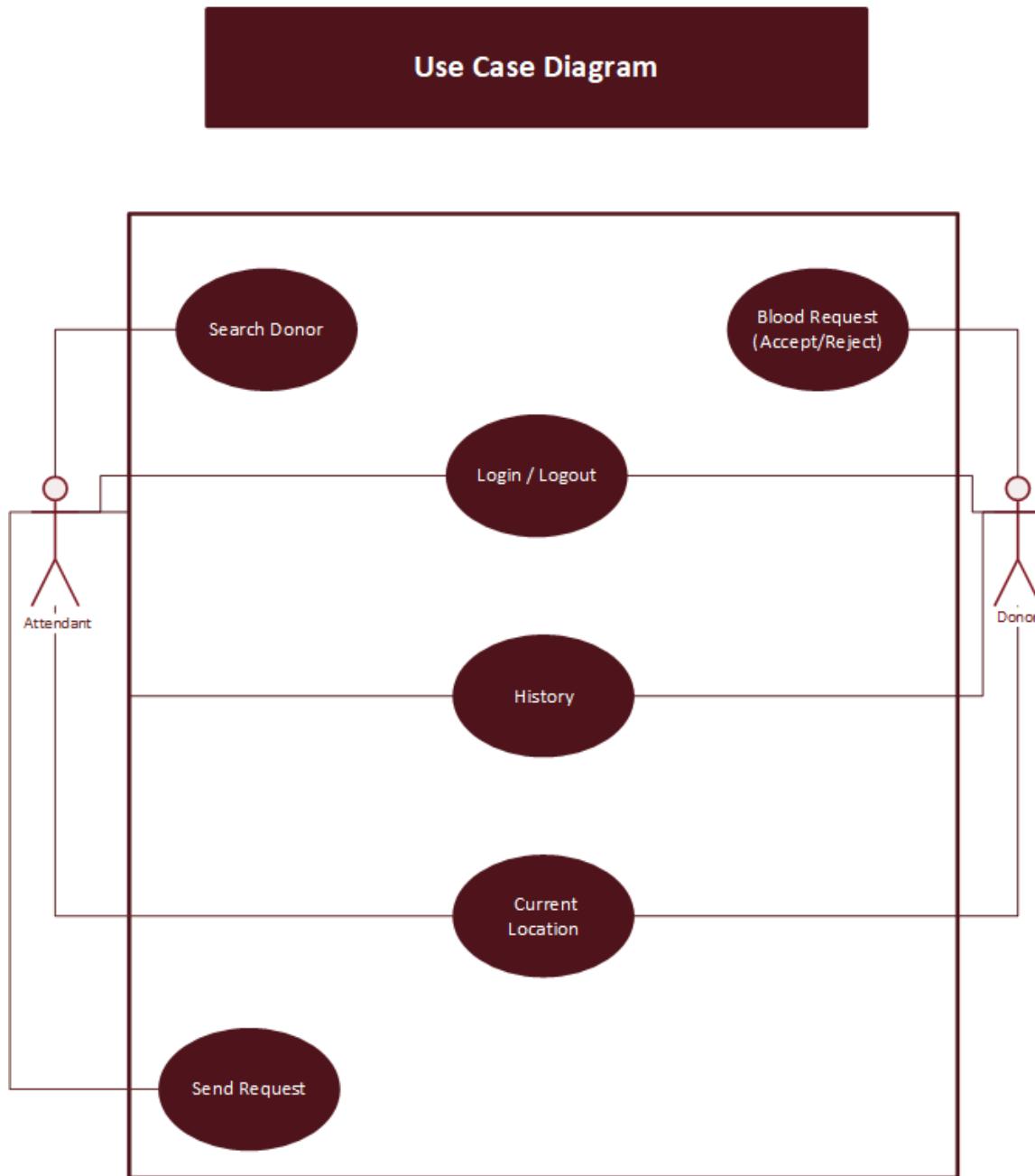


Figure 1: sample use case diagram with explanation

3.5 Extended use cases

1.1.1 Use Case donor registration

Document each use case. This can have completed using the tables provided below:

Section: Main			
Name:	Registration		
Actors:	Blood donor, user, admin		
Purpose:	For donor registration.		
Description:	To register a donor so that they can donate blood or look into the blood bank inventory.		
Cross References:	Functions: blood group name A donor registers for donation we will give take donation after knowing their blood group and details		
Pre-Conditions	Donor selected a form to fill		
Successful Post-Conditions	Information is valid it save in database.		
Failure Post-Conditions	Invalid info error message will be displayed		
Typical Course of Events			
Actor Action		System Response	
1	This use case begins when a donor opens website and starts the login process		
2	System determines if the info is right	3	
4	Info determined to be right	5	Donor gets login password
7	Informs that they can access the inventory		
8	Donor searches for blood groups .		
		10	System shows all the blood group details

Alternative Course	
Step 2:	Wrong info given.
Step 3:	No blood group details shown.

Table 6 Extended use case (1)

1.1.2 Use Case update stock

Document each use case. This can have completed using the tables provided below:

Section: Main			
Name:	Update stock		
Actors:	Admin and user		
Purpose:	For updating blood bank stocks		
Description:	A user or an admin updates stocks after a donation is made or the blood is taken for the patients.		
Cross References:	Functions: blood taken, donation made Use Cases: a user or an admin manages and updates if the blood is taken or donated.		
Pre-Conditions	User selected the blood bank module as a blood bank instead of a user		
Successful Post-Conditions	If internet is good stock updates		
Failure Post-Conditions	Internet bad then updating fails.		
Typical Course of Events			
Actor Action		System Response	
1	This use case begins when blood is donated or is taken		
2	User checks if any blood is missing or a donation has been made .		

4	If either condition Is true				
7	User updates the details of blood	8	Updates are made		
Alternative Course					
Step 2:	Unable to update blood bank details.				
Step 3:	Miss place of blood bags .				

Table 7 Extended use case (2)

1.1.3 Use Case donation requests

Document each use case. This can have completed using the tables provided below:

Section: Main			
Name:	Viewing donation requests		
Actors:	User, donor		
Purpose:	For view the blood requests		
Description:	A request has been made for donation or receiving request for blood then the user takes a look at the requests		
Cross References:	Functions: Sensor donation request form Use Cases: donor must have given right info at the request form and tell their details for donations or to get blood		
Pre-Conditions	The user has selected requests page		
Successful Post-Conditions	If internet Is good requests must be found		
Failure Post-Conditions	If internet bad the backup screen will be displayed		
Typical Course of Events			
Actor Action		System Response	
1	This use case begins when a donor or receiver launches a request		

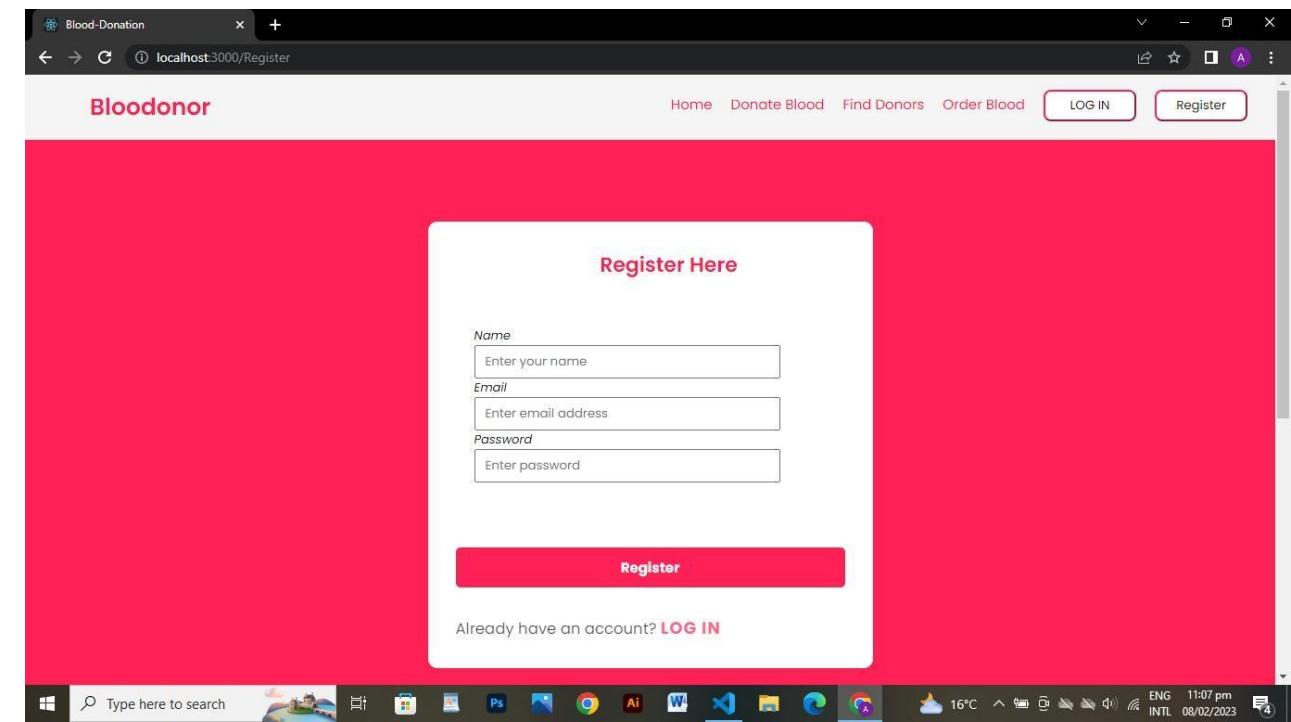
2	User opens the request tab	3	Systems displays all requests.		
4	Users responses to requests	5	Systems marked the request approved and sent to process		
6	Check all requests		System gives all requests		
Alternative Course					
Step 2:	No requests available .				
Step 3:	Requests are all completed.				

Table 8 Extended use case (3)

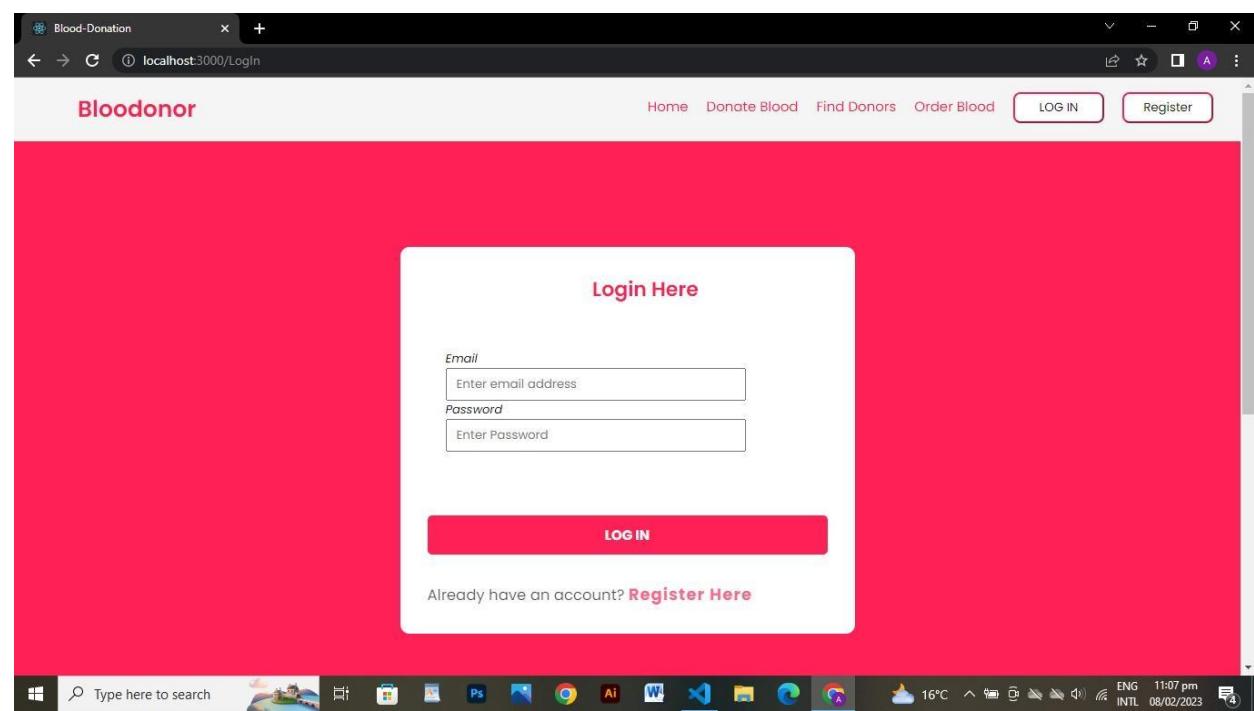
1.2 User interfaces (mock screens)

Website:

Register Page:



Login Page:



Blood Request Form:

The screenshot shows a web browser window titled "Blood-Donation" with the URL "localhost:3000/OrderBlood". The page has a header with the title "Blooddonor" and navigation links for "Home", "Donate Blood", "Find Donors", "Order Blood", "LOG IN", and "Register". Below the header is a section titled "BLOOD Order" containing several input fields:

- Full Name: A text input field with placeholder text "Enter your name".
- Email: A text input field with placeholder text "email@example.com".
- Phone: A text input field with placeholder text "Phone Number".
- Address: A text input field with placeholder text "Enter your address".
- Blood Group: A dropdown menu currently set to "A+".

At the bottom of the form is a red "Order" button.

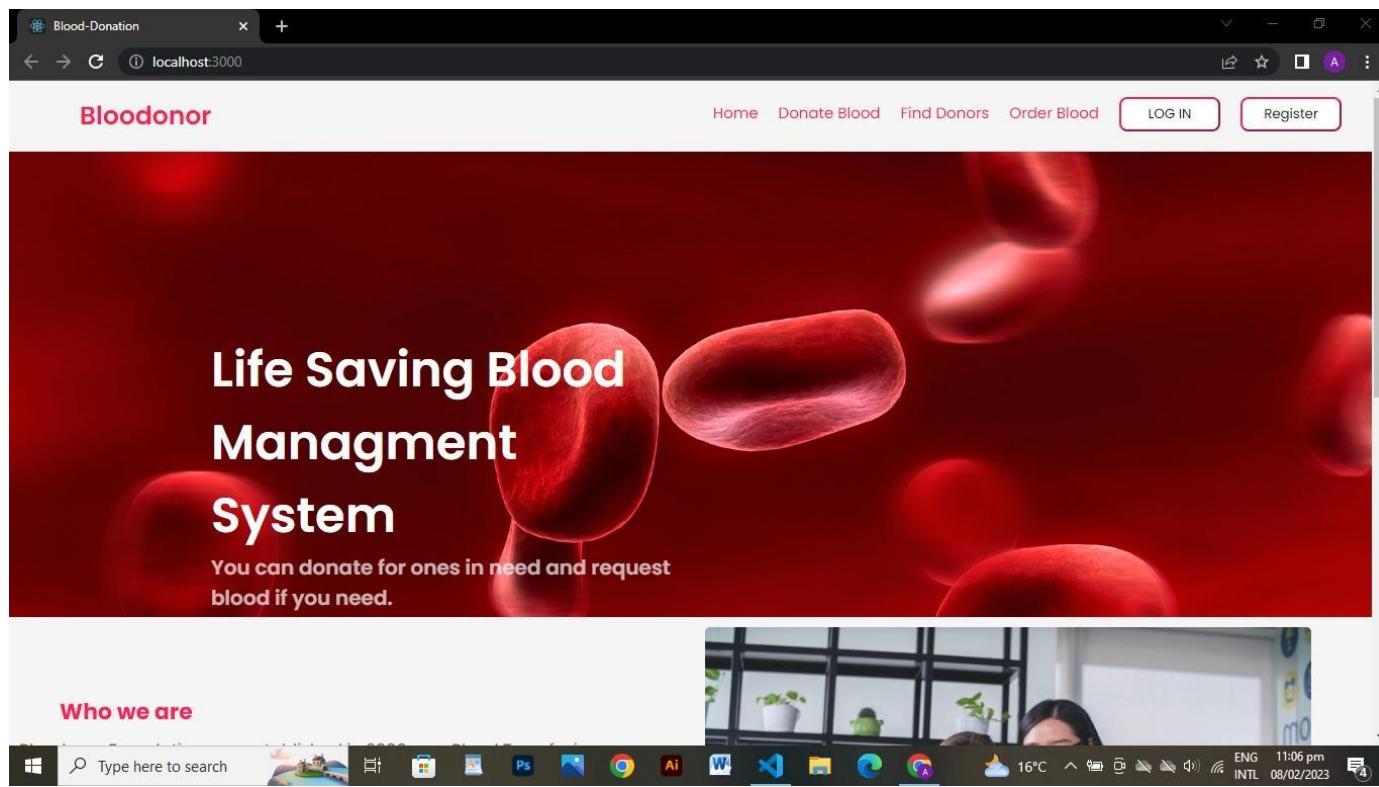
The browser's taskbar at the bottom shows various application icons and system status indicators, including the date and time (08/02/2023, 11:07 pm), battery level (4), and network connection.

Request for Donor:

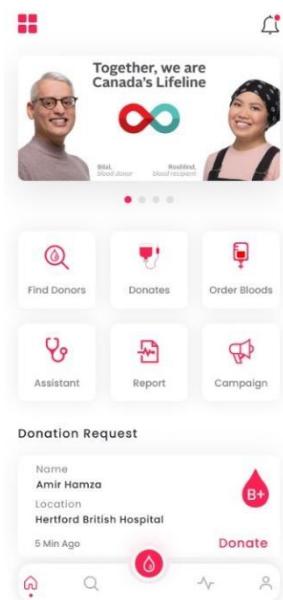
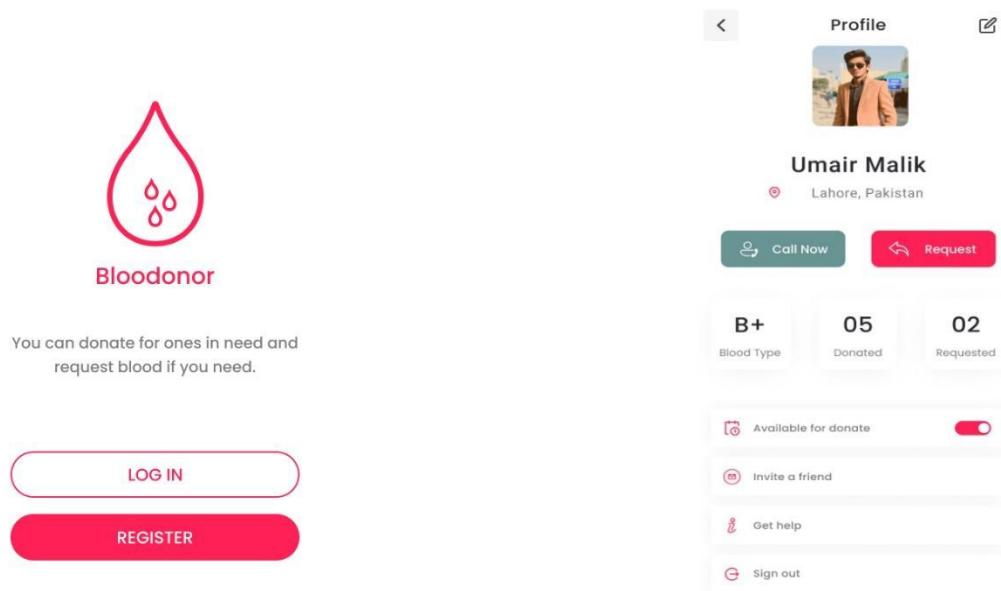
The screenshot shows a web browser window titled "Blood-Donation" with the URL "localhost:3000/FindDonors". The page is titled "Bloodonor" and features a "Donor Request" section. This section displays two separate requests, each enclosed in a red-bordered box. The first request is for "ayesha" with number "03306178305" at "Wapda town, G2" and blood group "A+". The second request is for "Asad Saeed" with number "03306178305" at "Umt Lahore" and blood group "A+". The browser's toolbar and taskbar are visible at the bottom.

Name	Number	Location	Blood Group
ayesha	03306178305	Wapda town, G2	A+
Asad Saeed	03306178305	Umt Lahore	A+

Website GUI:



Mobile App:



2. DATA FLOW DIAGRAM (OPTIONAL)

2.1 Data Flow Diagram Level 0

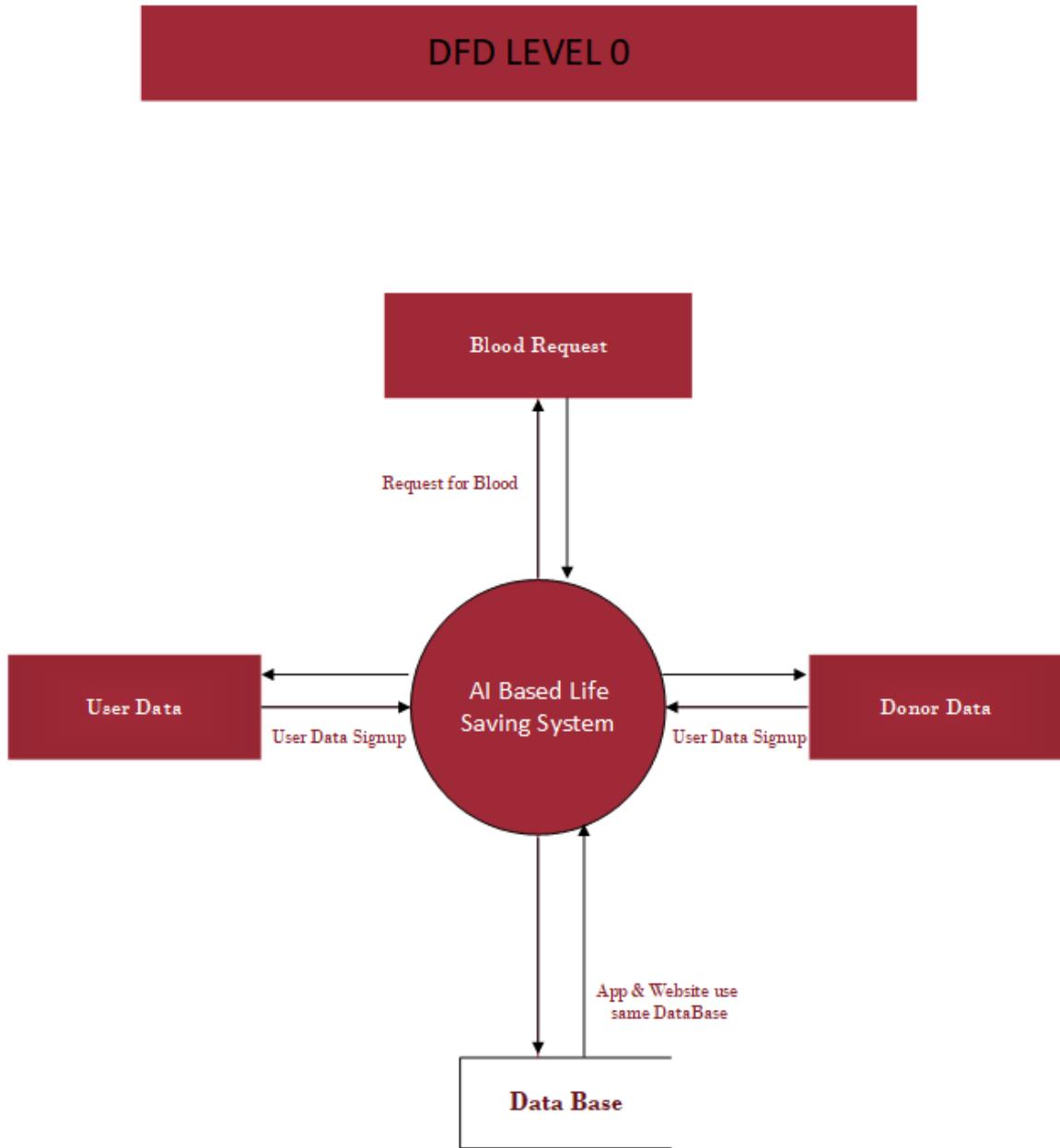


Figure 10 DFD level 0

2.2 Data Flow Diagram Level 1

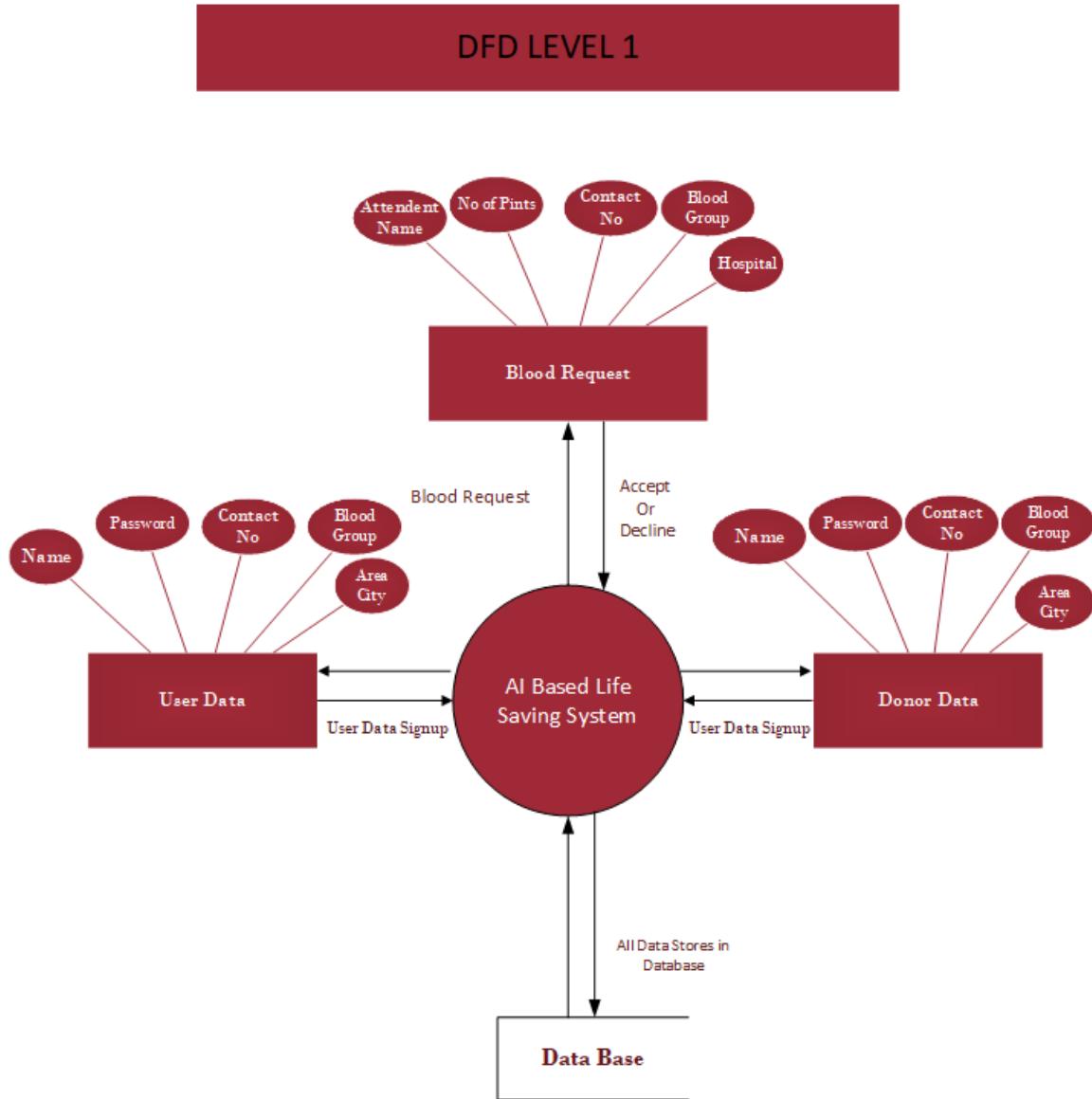
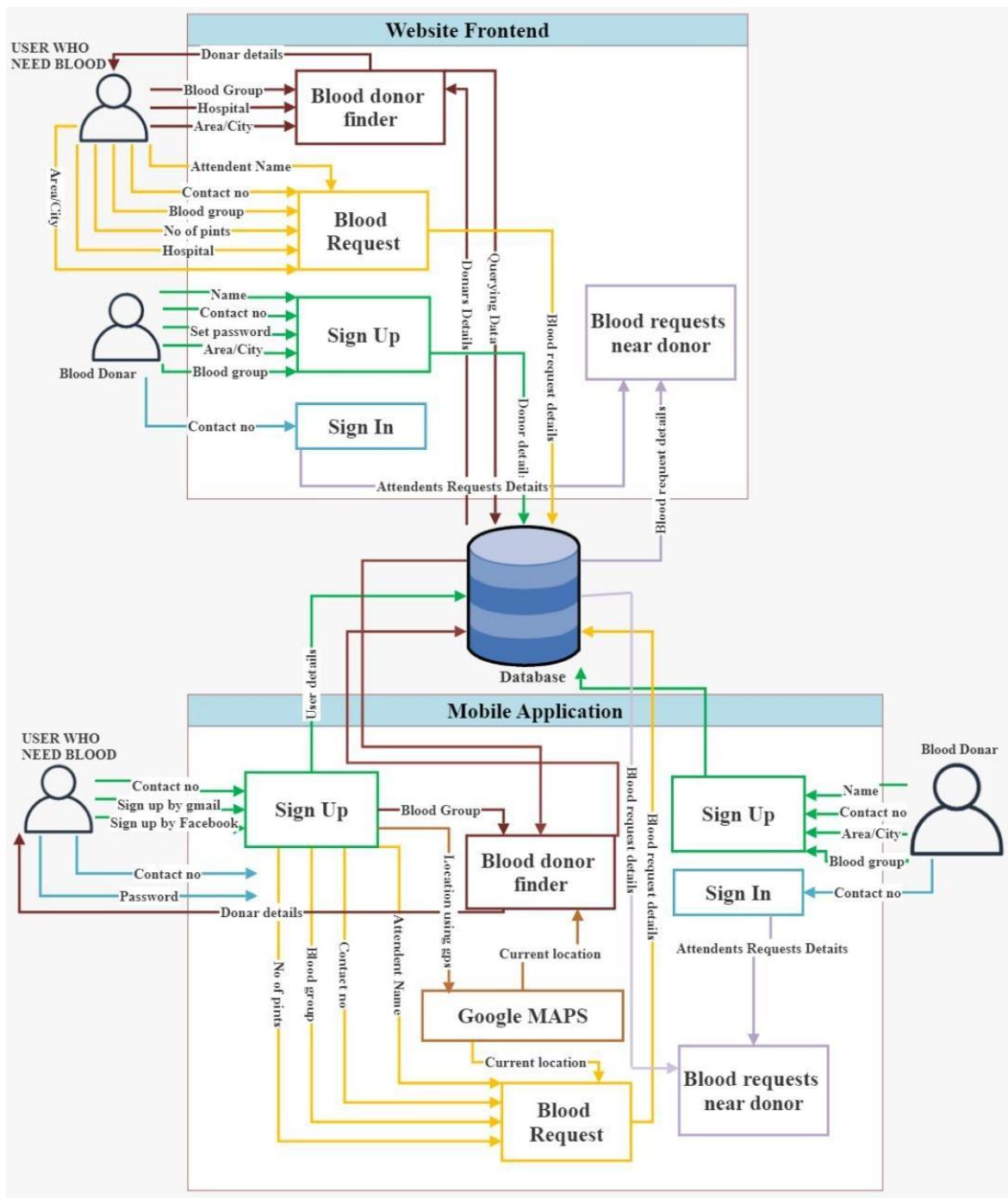


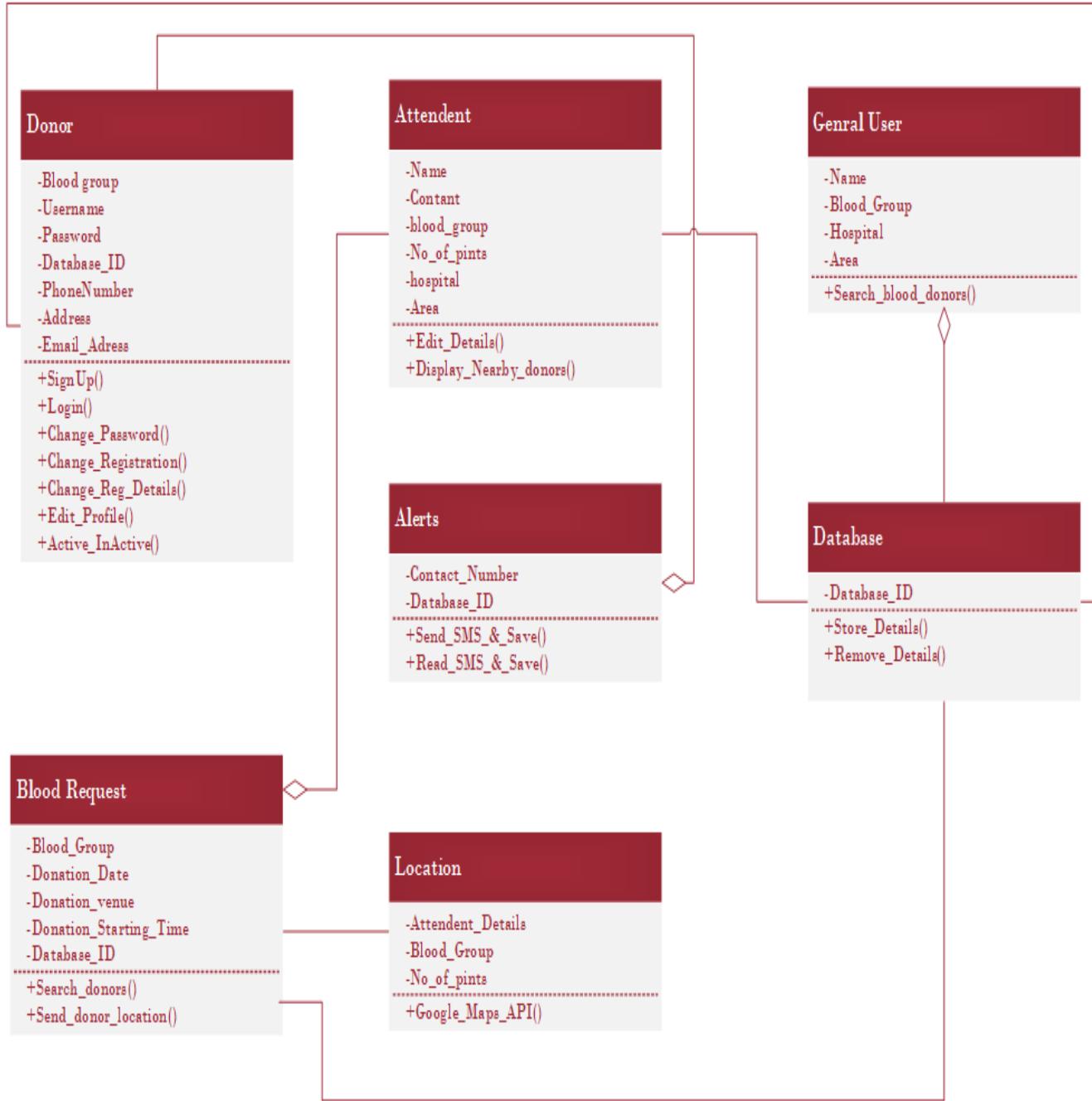
Figure 11 DFD level 1

3. SYSTEM DESIGN

3.1 System Architecture Diagram



3.2 Class Diagram



3.3 Sequence Diagrams

Donor

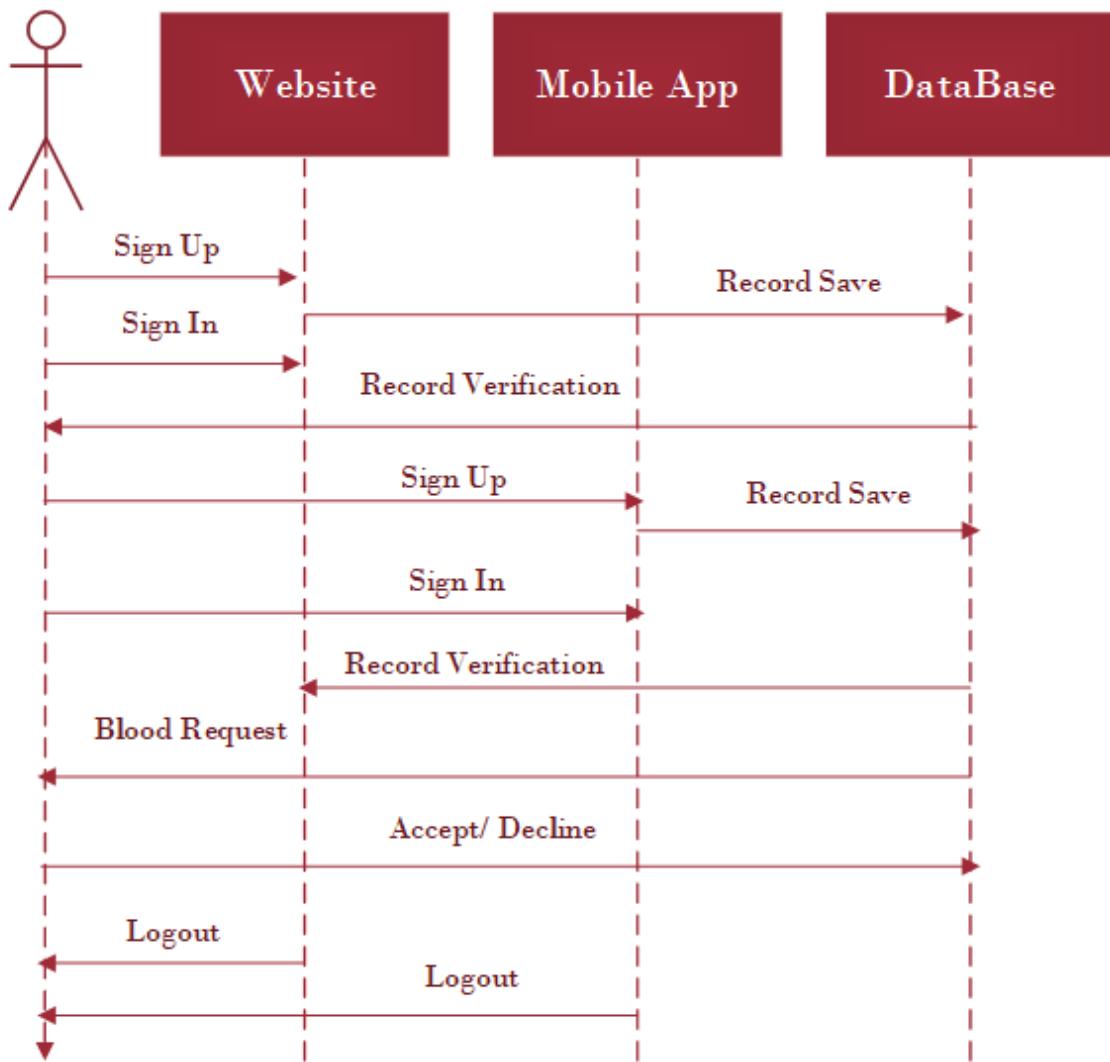


Figure 14 Sequence Diagram (donor)

Attendant

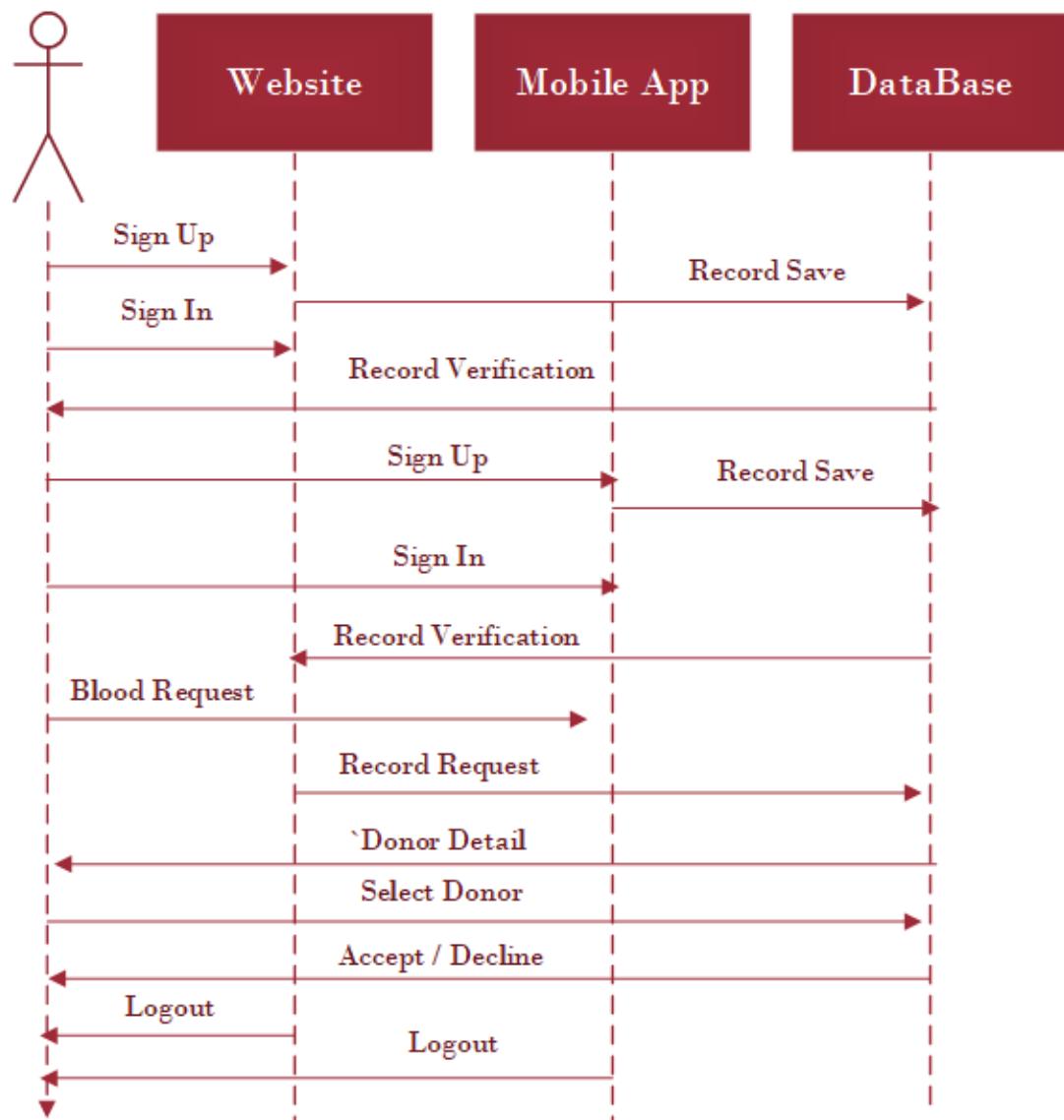
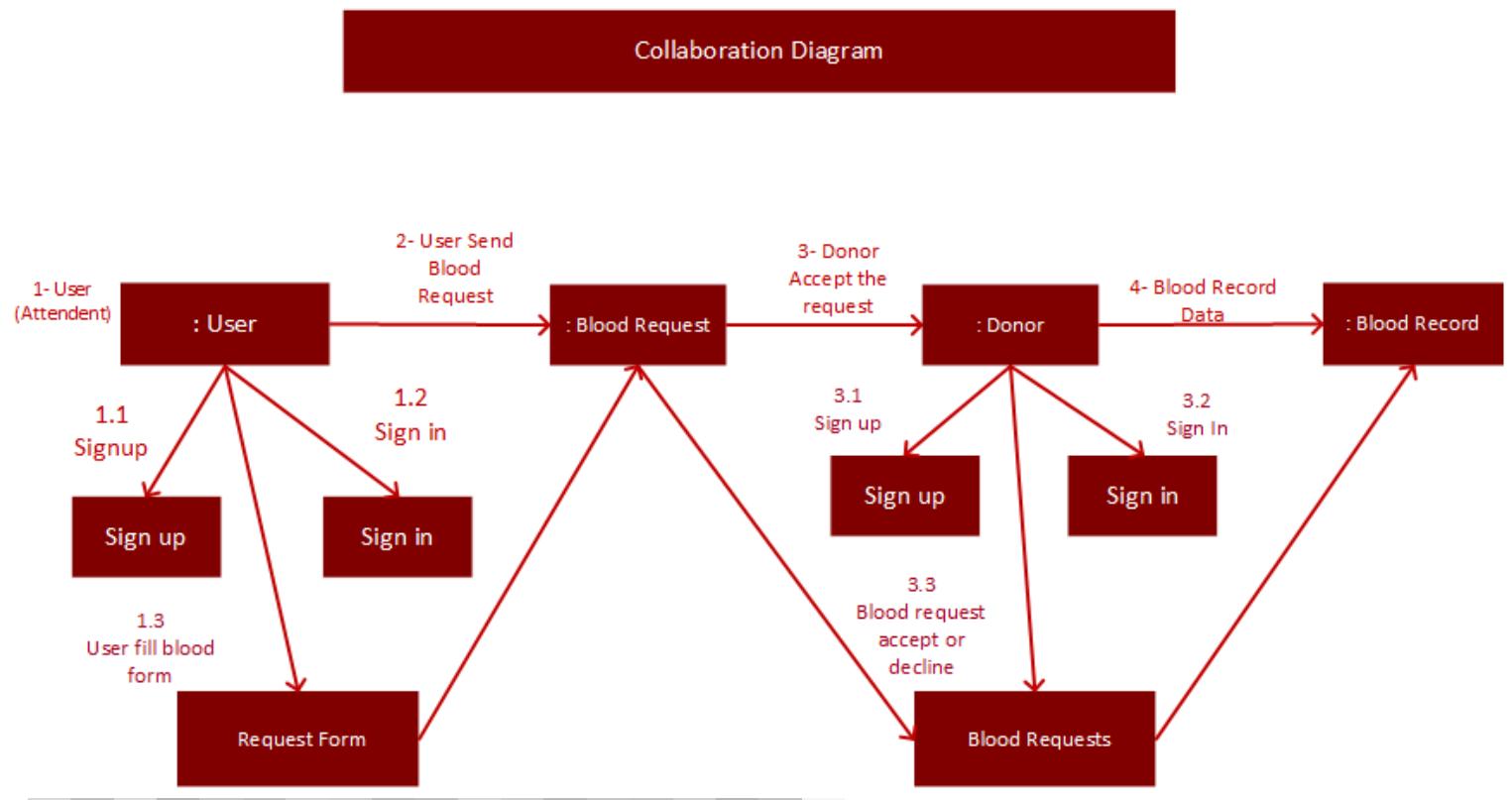


Figure 15 Sequence Diagram (Attendant)

3.4 Collaboration Diagrams



3.5 ERD

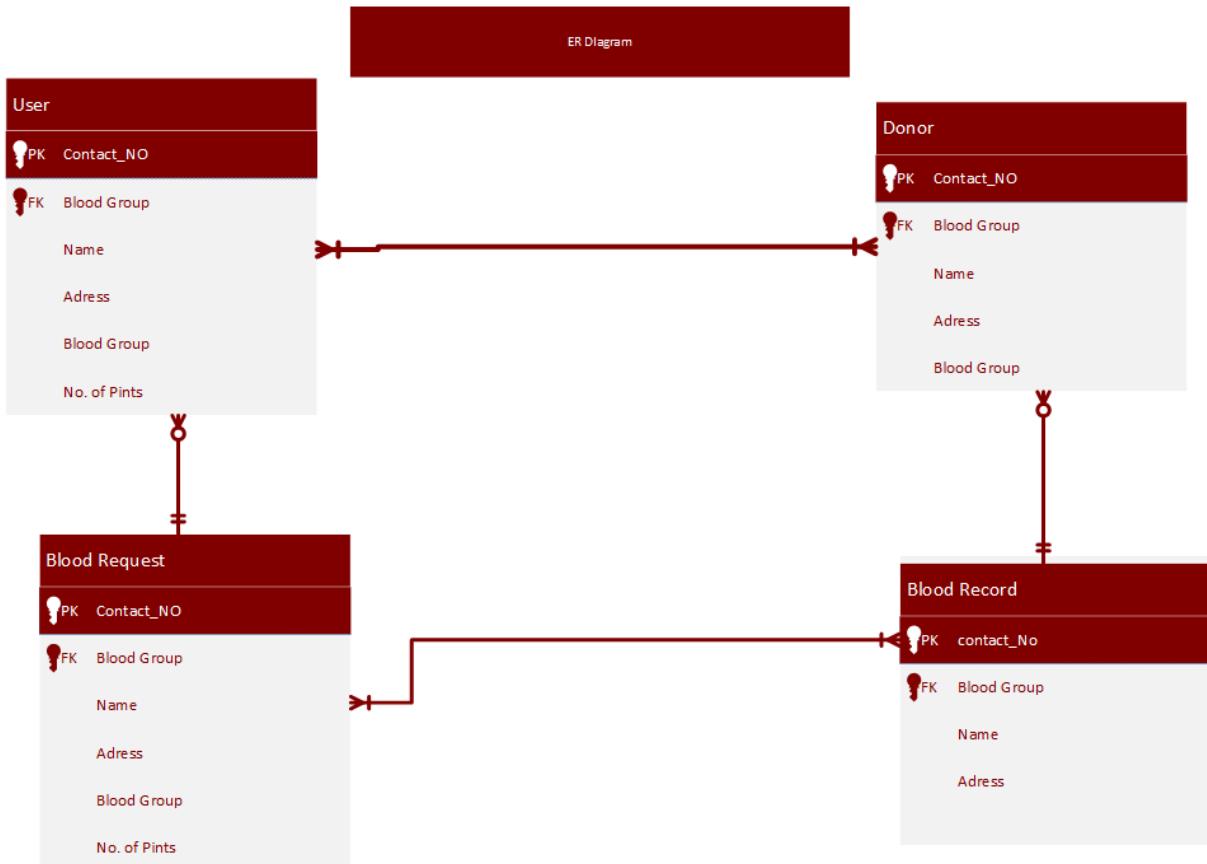


Figure 17 ERD

3.6 Data Dictionary

Element Name	Type	Validation	Mandatory	Remarks
BloodGroup	String	yes	(O+,B+)	Blood groups must be in the following format
name	String	No	Should be a proper string	Without username no login will be allowed
Password	Password	Yes	Should be more than 8 digits	No login without password
DataBase_ID	String	Yes	NOT NULL	To assign specific data
Address	String	No	NOT NULL	it should be well defined address
Email_Adress	String	Yes	(djavaid81@gmail.com)	To store emails for registrations and messages
Contact	Integer	Yes	(03481491327)	To store contact info for messages
Hospital	String	No	NOT NULL	To store hospital info
Attendent_details	String	No	NOT NULL	To store attendant info
No_Of_Pints	Integer	No	NOT NULL	To store the per donor pints data

4. IMPLEMENTATION DETAILS

4.1 Development Technologies

React - Front End Development:

We are using React js in this project for front end development for creating the GUI of the website because it makes creating dynamic and interactive user interfaces easier. Website consists of different pages which can help people to find their desired blood requirements easily. There is a log in page in website where people can register when they will donate blood and their information will be saved for later use. There is a request form where people who are in need of blood can request blood by telling their blood group and number of pints so that donors can see the request and accept it for donating blood.

Django – Back End Development:

Django connects website and application with Neo4j and performs data allocation and send data back to the application. It allows to save the data which it receives from website and mobile application in the form of node in Neo4j and it authenticates the whole data which is entered by user from website and mobile app.

React Native – Mobile App Development:

We are using react native for front end of mobile application using react native we made a different forms and different portal where the user can find the donor data and for the blood request along with this all we are using the react native to send the data back to the Django where it will work as a backend of the mobile app and react native further sends the things to perform action on using Django.

Neo4j – Database:

We are using Neo4j as our database it is almost considered as the new launched database in the market which is basically a graph base database in which we have the nodes and we can create the relationship between the node in the form of graph so like that in our project we are using this database because this database will create the relation of our donor between each other in this our donor will count as a node and through this database the relationships will be created which help us to know that which user has relation with which user or donor or which donor has relation with which donor or user.

4.2 Deployment setup

- Visual Studio: It will be used for the front end coding and mobile app coding and it uses Java script language and for web development we use the react and for mobile app development we used react native.
- Django: It will be used as backend coding and linking of the website mobile app and database and it uses the python language
- Neo4j: It will be used for the database and it uses the cipher language which is used to make a nodes and relation between the nodes

4.3 Algorithms

```
4
5 const FindDonors = () => {
6   const [donors, setDonors] = useState([]);
7
8   const Card = ({ item }) => {
9     console.log(item);
10    return (
11      <div className="donor-data">
12        <p>Name</p> {item?.name}
13        <p>Number</p> {item?.phone} <p>Location</p> {item?.address} <p>Blood Group</p> {item?.blood}
14      </div>
15    );
16  };
17
18  useEffect(() => {
19    const starCountRef = ref(database, "BloodDonationRequest");
20    onValue(starCountRef, (snapshot) => {
21      let data = snapshot.val();
22      data = Object.keys(data).map((key) => data[key]);
23
24      // No implementation found
25    });
26  }, []);
27
28  return (
29    <div>
30      <div className="FindDonors-heading">
31        <h2> Donor Request </h2>
32      </div>
33
34      <div>
35        {donors.map((donor) => (
36          <Card item={donor} />
37        )));
38      </div>
39    </div>
40  );
41}
42
43 export default FindDonors;
44
```

Figure 18 Donor Finding

This code shows us that how the donor can be search when someone will request for finding a donor

```

1 import React, { useState } from "react";
2
3 const OrderBlood = () => [
4   const [userData, setUserData] = useState({
5     name: "",
6     email: "",
7     phone: "",
8     address: "",
9     blood: ""
10   });
11
12   let name, value;
13   const postUserData = (event) => {
14     name = event.target.name;
15     value = event.target.value;
16     setUserData({ ...userData, [name]: value });
17   };
18   // firebase
19
20   const submitData = async (event) => {
21     event.preventDefault();
22     const { name, email, phone, address, blood } = userData;
23
24     if (name && email && phone && address && blood) {
25       const res = await fetch("https://bloodoner-1-default.firebaseio.com/Orders.json", {
26         method: "POST",
27         headers: [
28           "Content-Type": "application/json",
29         ],
30         body: JSON.stringify({
31           name,
32           email,
33           phone,
34           address,
35           blood,
36         })
37       });
38       if (res) {
39         setUserData({
40           name: "",
41           email: "",
42           phone: "",
43           address: "",
44           blood: ""
45         });
46     }
47   };
48 
```

Figure 19 Order Blood

```
45     });
46     alert("Successfully Order");
47   } else {
48     alert("Fill the data");
49   }
50 } else {
51   alert("Fill the data");
52 }
53 );
54
55 return (
56   <>
57   <form method="POST">
58     <div className="order-heading ">
59       <h2>BLOOD Order</h2>
60     </div>
61     <div className="order-form">
62       <div className="inner-box">
63         <label>Full Name</label>
64         <input
65           type="text"
66           name="name"
67           placeholder="Enter your name"
68           value={userData.name}
69           onChange={postUserData}
70           autoComplete="off"
71           required
72         />
73
74         <label>Email</label>
75         <input
76           type="text"
77           name="email"
78           placeholder="email@example.com"
79           value={userData.email}
80           onChange={postUserData}
81           autoComplete="off"
82           required
83         />
84
85         <label>Phone</label>
86         <input
87           type="text"
88           name="phone"
89           placeholder="Phone Number"
```

Figure 20 Order Blood

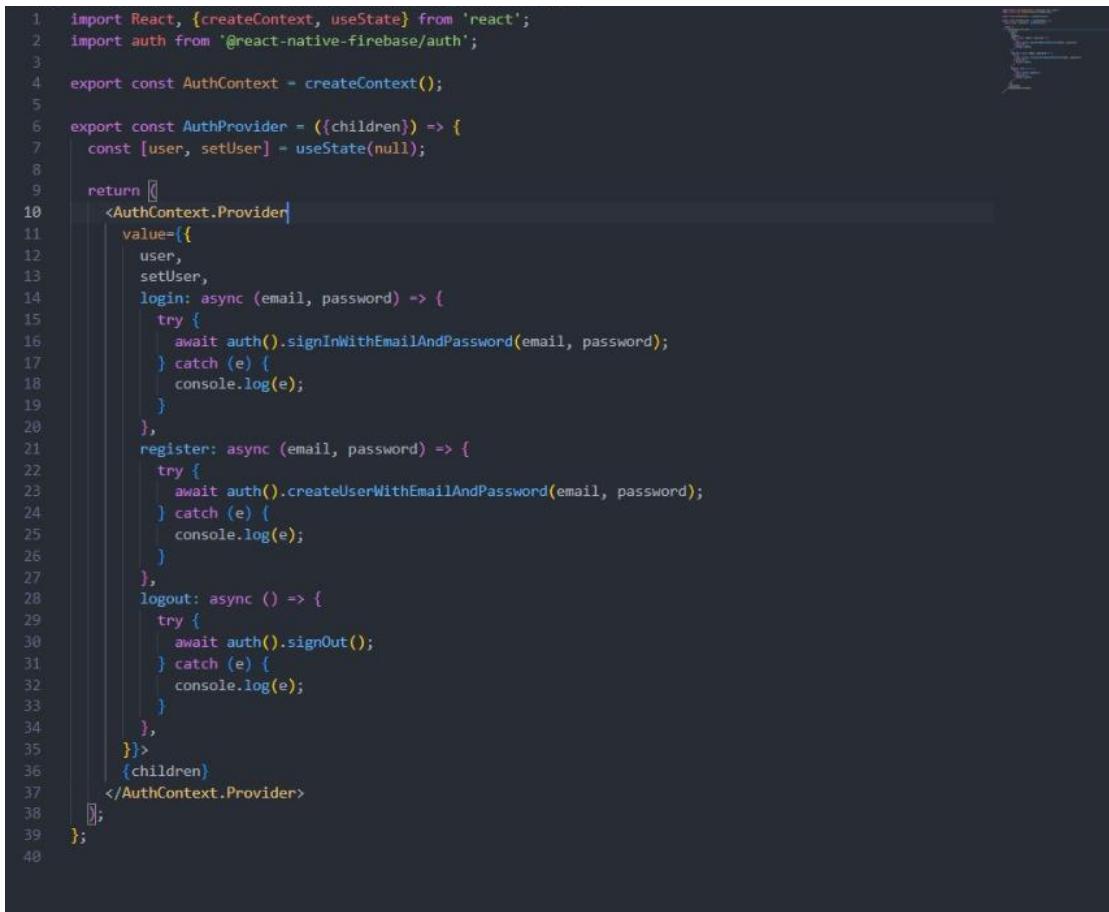
```

84
85             <label>Phone</label>
86             <input
87                 type="text"
88                 name="phone"
89                 placeholder="Phone Number"
90                 value={userData.phone}
91                 onChange={postUserData}
92                 autoComplete="off"
93                 required
94             />
95
96             <label>Address</label>
97             <input
98                 type="text"
99                 name="address"
100                placeholder="Enter your address"
101                value={userData.address}
102                onChange={postUserData}
103                autoComplete="off"
104                required
105            />
106
107             <label>Blood Group</label>
108             <input
109                 type="text"
110                 name="blood"
111                 placeholder="A+"
112                 value={userData.blood}
113                 onChange={postUserData}
114                 autoComplete="off"
115                 required
116             />
117
118             <button type="submit" onClick={submitData}>
119                 Order
120             </button>
121             </div>
122         </div>
123     </form>
124   </>
125 );
126 ];
127
128 export default OrderBlood;

```

Figure 21 Order Blood

This code shows us that how the person will request for the blood and enters his detail for requesting for blood



```
1 import React, {createContext, useState} from 'react';
2 import auth from '@react-native-firebase/auth';
3
4 export const AuthContext = createContext();
5
6 export const AuthProvider = ({children}) => {
7   const [user, setUser] = useState(null);
8
9   return [
10     <AuthContext.Provider>
11       value={(
12         user,
13         setUser,
14         login: async (email, password) => {
15           try {
16             await auth().signInWithEmailAndPassword(email, password);
17           } catch (e) {
18             console.log(e);
19           }
20         },
21         register: async (email, password) => {
22           try {
23             await auth().createUserWithEmailAndPassword(email, password);
24           } catch (e) {
25             console.log(e);
26           }
27         },
28         logout: async () => {
29           try {
30             await auth().signOut();
31           } catch (e) {
32             console.log(e);
33           }
34         },
35       )}>
36     {children}
37   </AuthContext.Provider>
38 ];
39 };
40
```

Figure 22 User Authentication

This Code is use for the user authentication in which it authorize the email and password from the user who login or register.

4.4 Constraints

4.4.1 Assumptions

- User should have internet facility to access website and mobile app.
- User need to Sign in the Mobile App to use it

4.4.2 System constraints

- Any device which is connected to the internet can access the system
- Mobile app is only working on the Smart Phones which have Play Store
- Device Should have the storage of 200 MB for installing this app

4.4.3 Restrictions

- Only admin can access the database

4.4.4 Limitations

- Offline Mode
- Call System

5. TESTING

5.1 Extended Test Cases

Test Case ID: 01		Test Design By: Ayesha Muhktar									
Test Module Name: Login/Logout Submit		Test Design Date: 09/01/23									
Test Priority: High		Test Executed By: Umair Malik									
Test Name: Test Login/Logout Button		Test Executed Date: 8/04/23									
Description: For accessing app and website portal											
Pre-Condition: Must Connected with internet											
Dependencies: None											
Step No.	Test step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes					
1	Connect with internet and open app				Pass						
2	Enter your details		Home page will open after authorization	Home page will open after authorization	Pass	Enter Valid Data					
Post Condition: App portal will login											

Table 9 Test Id 01

Test Case ID: 02		Test Design By: Ayesha Muhktar									
Test Module Name: Search Donor		Test Design Date: 09/01/23									
Test Priority: High		Test Executed By: Umair Malik									
Test Name: Searching Donor		Test Executed Date: 08/04/23									
Description: for searching the donor											
Pre-Condition: Enter Valid Data											
Dependencies: None											
Step No.	Test step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes					
1	Go to main menu and enter details		Blood Donor data will be shown	Blood Donor data will be shown	Pass						
Post Condition: You can access the donor details											

Table 10 Test Id 02

Test Case ID: 03		Test Design By: Muhammad Husnian									
Test Module Name: Blood Request		Test Design Date: 22/01/23									
Test Priority: High		Test Executed By: Umair Malik									
Test Name: Blood Request		Test Executed Date: 08/04/23									
Description: Generate blood request											
Pre-Condition: Enter Valid Patient details											
Dependencies: Must Login											
Step No.	Test step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes					
1	Open Blood Request Panel		Request Form will open	Request Form will open	Pass						
2	Enter Patient Details		Form will be submitted	Form will be submitted	Pass						
Post Condition: Request for blood will be generated											

Table 11 Test Id 03

Test Case ID: 04		Test Design By: Muhammad Husnain									
Test Module Name: Location		Test Design Date: 22/01/23									
Test Priority: Low		Test Executed By: Daniyal Javaid									
Test Name: Location		Test Executed Date: 15/04/23									
Description: Location Testing											
Pre-Condition: Allow Location access											
Dependencies: Internet Connection											
Step No.	Test step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes					
1	Enter your Location and Check nearby Donor		Nearby donor are shown	Nearby donor are shown	Pass	Must Allow Location Tracking					
Post Condition: You can see donor nearby you											

Table 12 Test Id 04

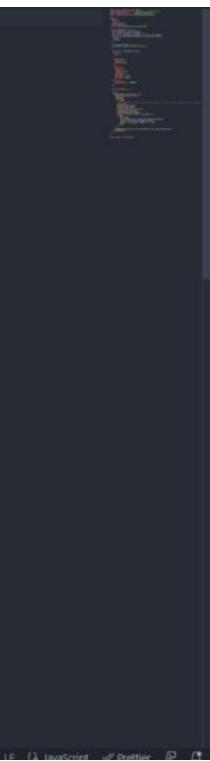
5.2 Decision Table

```
1 import React, {useState, useContext} from 'react';
2 import {
3   Text,
4   TouchableOpacity,
5   Image,
6   StyleSheet,
7   ScrollView,
8 } from 'react-native';
9 import FormInput from '../../components/FormInput';
10 import FormButton from '../../components/FormButton';
11 import {AuthContext} from '../../navigation/AuthProvider';
12
13 const LoginScreen = ({navigation}) => {
14   const [email, setEmail] = useState();
15   const [password, setPassword] = useState();
16
17   const {login} = useContext(AuthContext);
18
19   return (
20     <ScrollView contentContainerStyle={styles.container}>
21       <Image source={require('../../assets/logo.png')} style={styles.logo} />
22
23       <FormInput
24         labelValue={email}
25         onChangeText={(userEmail) => setEmail(userEmail)}
26         placeholderText="Email"
27         iconSource={require('../../assets/mail.png')}
28         keyboardType="email-address"
29         autoCapitalize="none"
30         autoCorrect={false}
31       />
32
33       <FormInput
34         labelValue={password}
35         onChangeText={(userPassword) => setPassword(userPassword)}
36         placeholderText="Password"
37         iconSource={require('../../assets/lock.png')}
38         secureTextEntry={true}
39       />
40
41       <FormButton buttonTitle="Log In" onPress={() => login(email, password)} />
42
43       <TouchableOpacity onPress={() => navigation.navigate('ForgotPassword')}>
44         <Text style={styles.forgotText}>Forgot Password?</Text>
45       </TouchableOpacity>
46     </ScrollView>
47   );
48 }
49
50
51
52
53
54
55
56
57 export default LoginScreen;
```

Figure 23 Login Screen

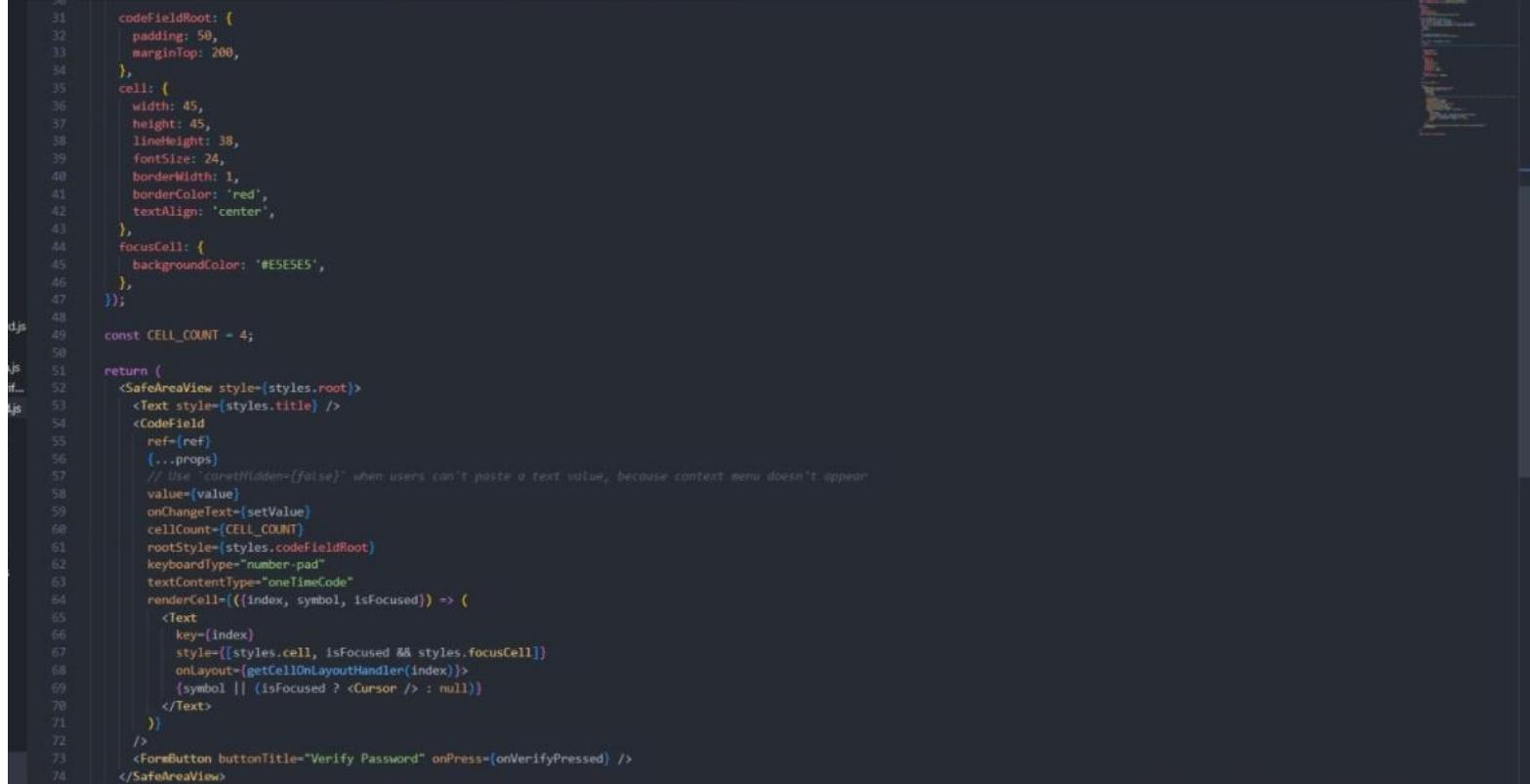
```
48       <TouchableOpacity onPress={() => navigation.navigate('RegisterScreen')}>
49         <Text style={styles.navButtonText}>
50           Don't have an account? Create here
51         </Text>
52       </TouchableOpacity>
53     </ScrollView>
54   );
55 }
56
57 export default LoginScreen;
```

Figure 24 Login Screen



```
1 import React, {useState} from 'react';
2 import {SafeAreaView, Text, StyleSheet} from 'react-native';
3 import FormButton from '../components/FormButton';
4 import {useNavigation} from '@react-navigation/native';
5
6 import {
7   CodeField,
8   Cursor,
9   useBlurOnFulfill,
10  useClearByFocusCell,
11 } from 'react-native-confirmation-code-field';
12
13 const VerifyPassword = () => {
14   const navigation = useNavigation();
15   const [value, setValue] = useState('');
16   const ref = useBlurOnFulfill({value, cellCount: CELL_COUNT});
17   const [props, getCellOnLayoutHandler] = useClearByFocusCell({
18     value,
19     setValue,
20   });
21
22   const onVerifyPressed = () => {
23     navigation.navigate('SuccessfullyVerify');
24   };
25
26   const styles = StyleSheet.create({
27     root: {
28       flex: 1,
29     },
30
31     codeFieldRoot: {
32       padding: 50,
33       marginTop: 200,
34     },
35     cell: {
36       width: 45,
37       height: 45,
38       lineHeight: 38,
39       fontSize: 24,
40       borderWidth: 1,
41       borderColor: 'red',
42       textAlign: 'center',
43     },
44     focusCell: {
45       backgroundColor: '#E5E5E5',
46     }
47   });
48
49   return (
50     <View style={styles.root}>
51       <CodeField
52         ref={ref}
53         value={value}
54         cellCount={CELL_COUNT}
55         keyboardType="number"
56         duration={200}
57         stepTime={15}
58         onLayout={getCellOnLayoutHandler}
59       />
60       <FormButton
61         title="Verify"
62         onPress={onVerifyPressed}
63       />
64     </View>
65   );
66 }
67
68 const CELL_COUNT = 4;
```

Figure 25 Password Verification



A screenshot of a code editor displaying a React component for password verification. The component uses the `CodeField` component from the `react-native-safe-area-view` library. The code includes styling for the root container and individual cells, handling of focused cells, and a button to verify the password.

```
26
27   codeFieldRoot: {
28     padding: 50,
29     marginTop: 200,
30   },
31   cell: {
32     width: 45,
33     height: 45,
34     lineHeight: 38,
35     fontSize: 24,
36     borderWidth: 1,
37     borderColor: 'red',
38     textAlign: 'center',
39   },
40   focusCell: {
41     backgroundColor: '#E5E5E5',
42   },
43 });
44
45 const CELL_COUNT = 4;
46
47 return (
48   <SafeAreaView style={styles.root}>
49     <Text style={styles.title} />
50     <CodeField
51       ref={ref}
52       {...props}
53       // Use `caretHidden={false}` when users can't paste a text value, because context menu doesn't appear
54       value={value}
55       onChangeText={setValue}
56       cellCount={CELL_COUNT}
57       rootStyle={styles.codeFieldRoot}
58       keyboardType="number-pad"
59       textContentType="oneTimeCode"
60       renderCell={({index, symbol, isFocused}) => (
61         <Text
62           key={index}
63           style={[styles.cell, isFocused && styles.focusCell]}
64           onLayout={getCellOnLayoutHandler(index)}
65           {symbol || (isFocused ? <Cursor /> : null)}
66         </Text>
67       )}
68     />
69     <FormButton buttonTitle="Verify Password" onPress={onVerifyPressed} />
70   </SafeAreaView>
71 );
```

Figure 26 Password Verification

```

40  const getLocation = async () => {
41    fetch(
42      `https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial&origins=${origin.description}&destinations=${destination.description}`
43    )
44      .then(res => res.json())
45      .then(data => {
46        dispatch((data.rows[0].elements[0]));
47      });
48  };
49  [origin, destination, GOOGLE_MAPS_APIKEY];
50
51  if (!origin) {
52    return null;
53  }
54
55  const coordinate = destination
56  ? {
57    latitude: destination.location.lat,
58    longitude: destination.location.lng,
59  }
60  : [
61    latitude: origin.location.lat,
62    longitude: origin.location.lng,
63  ];
64
65  return (
66    <MapView
67      ref={mapRef}
68      style={tw` flex-1`}
69      mapType="mutedStandard"
70      initialRegion={{
71        latitude: origin.location.lat,
72        longitude: origin.location.lng,
73        latitudeDelta: 0.005,

```

Figure 27 Maps Integration

5.2.2 Decision coverage table

Condition	Internet Access	Main Portal	Login	Location	Logout
Login	T	F	T	F	F
Main portal	T	T	T	F	F
Search Donor	T	F	T	T	F
Request Portal	T	F	T	T	F
Log Out	T	F	F	F	T

Table 13 Decision Table

5.3 Traceability Matrix

5.3.1 RID vs UCID (requirements vs use cases)

UCID/RI D	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9
UC 1	✓	✓	✓		✓	✓		✓	✓
UC 2	✓	✓	✓	✓				✓	
UC 3	✓	✓	✓		✓	✓	✓	✓	✓
UC 4	✓	✓		✓		✓		✓	
UC 5	✓		✓		✓		✓		✓
UC 6	✓	✓				✓			
UC 7	✓	✓	✓		✓		✓		✓
UC 8	✓			✓		✓			✓
UC 9	✓	✓			✓				✓
UC 10		✓		✓		✓			

Table 14 RID/UCID

5.3.2 Prototypes (RID vs PID)

PID/FRI D	FR1	FR2	FR3	FR4	FR5
PI1	✓	✓	✓	✓	✓
PI2	✓	✓	✓		
PI3	✓		✓	✓	✓
PI4	✓	✓	✓		✓
PI5	✓	✓	✓	✓	
PI6	✓	✓	✓		✓
PI7		✓	✓		
PI8	✓	✓		✓	

Table 15 PID/FRID

5.3.3 Test Cases (RID vs TID)

TID/FRID	FR1	FR2	FR3	FR4	FR5
T1	✓	✓			✓
T2	✓	✓	✓	✓	✓
T3	✓	✓	✓		✓
T4	✓	✓	✓		✓
T5	✓	✓		✓	✓
T6	✓				
T7	✓		✓		✓
T8	✓	✓		✓	✓

Table 16 TID/FRID

5.3.4 Coverage (UCID vs TID)

TID/UCID	UC1	UC2	UC3	UC4	UC5	UC6	UC7
T1	✓	✓	✓	✓	✓	✓	✓
T2	✓	✓		✓		✓	
T3	✓	✓		✓	✓	✓	✓
T4	✓		✓			✓	
T5		✓	✓	✓	✓		✓
T6	✓	✓					
T7	✓		✓	✓	✓	✓	
T8		✓			✓		✓

Table 17 UCID/TID

6. RESULTS/OUTPUT/STATISTICS

6.1 %completion

Full Filled Requirement: 31

Total Requirement: 40

% Completion: 77.5%

6.2 %accuracy

Implemented Requirement: 29

Total Requirement: 40

% Accuracy: 72.5%

6.3 %correctness

Tested Requirement: 48

Total Requirement: 56

% Correctness: 85.7%

7. CONCLUSION

The conclusion for a final year project on AI based lifesaving system that consists of a website and a mobile could include the following points:

Achievement of Project Objectives: This system will help people to find the desire blood at their nearby location and it can be also helpful that using the Graph based database you can find the required blood by your nearby people or mutual people and you just have to put the request of the desire blood group and you will find that blood. You have to put the request and the donor who is registered as a donor will see that request and he will accept your blood request and you can contact that donor after he accept your blood request. In both website and App you can generate a request in which you need to describe your desired blood group and your location and it will be shown on the portal

Technology Integration and Suitability: We tried to use the latest technologies in our project such as react for the website because it is the most demanding in our society and it will full fill our needs and for back end we are using python Django which will help our website to work more efficiently with React and for mobile app we are using the React Native so we can easily link our website and Mobile App and for database we used Neo4j because for our project it will be the best database we find in our surroundings because we need to build a connections with the users and donor so neo4j provides us a facility in which we can create a connections and proper relationship using the Graph. Along with this all we can make our project more sustainable and for long term it can also be helpful if we get load on our website or app then this technology will help us to handle the load on both.

User Experience and Interface Design: We made a simple but effective design so that a common man who just know how to use the mobile or laptop can access our website and app because emergency can be occur to anyone. We didn't consider it that this app will be used by a Literate person it can be use by anyone maybe a illetatre person so that why our design for the app is so simple that anyone can easily understand that and use it.

Impact and Contribution: We tried our best to make this system more easy for everyone so that anyone can use this system and it can help people in their bad time instead of calling the different people you just need to enter the detail of patient and blood group and this will help you to find your desire blood group. We tried to contribute in the humanity so that we can save the lifers of people using this system many people just die because of lack of blood so we just give a small try and small contribution to humanity so that we can save the lives of people using this system.

In conclusion, the project successfully developed AI based Life Saving system. This system demonstrated effective integration of these technologies to provide a user-friendly interface, robust backend functionality, and efficient database management. The project holds potential to make a positive impact on the healthcare sector and presents opportunities for further development and collaboration in the future.

8. FUTURE WORK

In Future we have some goals to achieve and make it more common for daily use we will try to reach more audience using different marketing techniques and make it more likely common so that people can use this app as much as they can whenever they are in any emergency. Furthermore we will connect NGOs and Hospitals with this portal and Create achievement system for the user. We will also try to fulfill the SDGs Goal by adding the more things. We will try to fulfill the SGD-17 by signing the different MOUs with the Different Company and as per the other apps we will try to add the points for the person who donates more blood will receive a reward in the form of points which he or she can redeem it for its personal use. Other than that our Mission is to make this app more helpful for the people who required blood. We will provide service for the NGOs to organize the blood thalassemia camps and share it through our app so that as much as people know about thalassemia and they donate bloods in the camps moreover we will try to make a forum where the issues related to the blood disease can be discussed and when it is connected to the Hospital then the Dr. will be a part of this complain and they will guide the People about these disease and along that we have decided to run some awareness complain online through the App form and those who will be a part of it they can achieve some points that already discussed before that they can use them for their personal use. Such as if we sign MOUs with the labs and hospitals that person can redeem the special discount through those points and as we will achieve our goals we will add more things and more requirements according to the situation of that time.

9. BIBLIOGRAPHY

- [1] C. Gackenheimer, Introduction to React, 2015.
- [2] javinpaul, "The 2022 React JS Developer RoadMap," 19 10 2019. [Online]. Available: <https://medium.com/javarevisited/the-2019-react-js-developer-roadmap-9a8e290b8a56>. [Accessed 19 10 2022].
- [3] "Django Introduction," 08 14 2019. [Online]. Available: https://www.w3schools.com/django/django_intro.php.
- [4] javinpaul, "2022 React JS Developer RoadMap," 19 10 2019. [Online]. Available: <https://medium.com/javarevisited/the-2019-react-js-developer-roadmap-9a8e290b8a56>. [Accessed 6 10 2022].
- [5] m. shah, "stiddocu," 25 10 2019. [Online]. Available: <https://www.studocu.com/row/document/sindh-maddressatul-islam-university>.
- [6] G. Sidelnikov, Learning React JavaScript Library from Scratch, Independently Published, March 17, 2017.
- [7] J. R. Briggs, Python for Kids, No Starch Press, 2013.
- [8] S. B. A. B. Ben Shaw, Web Development with Django, Packt Publishing, January 25, 2021.
- [9] B. Eisenman, Learning React Native, O'Reilly Media, October 23 ,2017.
- [10] C. Kemper, Beginning Neo4j, Apress, December 30, 2015.

10. APPENDIX

10.1 Glossary of terms

UI	User Interface
UX	User Experience
FR	Functional Requirement
NFR	Non Functional Requirement
SDGs	Sustainable Development Goals
BG	Blood Group
DB_ID	Databased
UC	Use Case
TID	Transaction ID
PID	Prototype ID
RID	Requirement ID

Table 18 Glossary

10.2 Pre-requisites

Requirements (Minimum):

- Device with internet connection
- Smart phone with 1-2GB RAM , Memory 200 MB

Final Year Project Report

The Smart Garage



Project Advisor:

Dr Saeed Ahmad

Submitted By:

Muhammad Hussnain Butt

F2016266216

Session

2016-2023

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

This final year project report has been presented in Bachelor of Computer Science. We would like to dedicate our project to our parents and more especially to our Project advisors as they are behind the completion and success of this project. They guided us in a right direction and gave us their precious time. Whenever we discussed our project with them, they always gave us better suggestions to improve or enhance our project. Every time a person has to open the Garage, he/she has to come out of the car and open it. This project will save a lot of time

Final Approval

- **Head of Department**

Department of Computer Science
University of Management and Technology
Lahore

- **Program Director (Final Year Projects)**

Department of Computer Science
University of Management and Technology
Lahore

- **Supervisor**

Department of Computer Science
University of Management and Technology
Lahore

- **Co-Supervisor**

(If any)

Acknowledgment

We would like to appreciate and express our gratitude to all the people who were the key support for us and make it possible for us to complete this report. We are thankful to our Final Year Project advisor, **Dr Saeed Ahmad**, who help, motivate, manage and support us to complete our project and our final report writing. We would also like to thank all the teammates who deserves to be appreciated as their hard work make this project possible to be completed.

Project Title:

The Smart Garage

Objective:

To provide the facility of saving time and money for the user

Undertaken by:

Muhammad Hussnain Butt

F2016266216

Supervised by: Dr Saeed Ahmad

Starting Date:

jan 1, 2021

Completion Date:

August 2, 2023

Tools Used:

Jupyter Notebook, Google colab, Tensor Flow object detection API, EasyOCR, Faster_r-cnn_resnet101_v1_1024*1024.

Operating System:

Windows 10

Plagairism Report

Declaration Form

I have carefully examined the documentation of the Final Year Project titled “*Project title*”; and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

The document has been checked for plagiarism through Turnitin software available in UMT Library. The similarities of the document are within acceptable range.

Moreover, the accompanying CDs contain PDF of the documentation, as well as the source code and binaries with user manual and installation guide.

FYP Advisor Name: _____ **Dr Saeed Ahmad** _____

Signature: _____

Date: _____

Abstract

In today's world everyone is busy with their work and humankind is working to make things easier and quicker for themselves so here comes Smart Garage System. This Systems redefines of how garages were, providing an automatic door opening system based upon sticker (unique code) recognition using deep learning. This would be done by using a camera which will capture the front side of the car further an algorithm will be used to judge the position of the sticker then recognize the sticker using optical character recognition and deep learning after that the system will check that weather the recognized sticker is present in the database, depending upon that it will either turn on green or red light (Green if it is present and red if it's not). This smart garage will increase the security and environmental safety. This system will also help the people to save time and money that they spend on hiring gate keepers. This system totally based on "Deep Learning".

REVISION CHART

Task	Group person available in meeting	Details	Complete date
Elicitation	Muhammad Husnain Butt	Advice to start elicitation Read related research papers To study related apps Advised to study risks, benefits, difficulties for our project.	Agu 2, 2021

Task	Group person available in meeting	Details	Complete date
Proposal	Muhammad Husnain butt	Advised to update proposal, add new features, to decide a platform that will be used for implementation tools and technologies	jul 4,2023
Use cases(Revision 1)	Muhammad Husnain butt	Use cases designed After meeting advised to correct pointed errors in use cases. To add common and specialized functionality.	jul 9,2023
Front end	Muhammad Husnain butt	Front end screens designed	feb 15,2023

Task	Group person available in meeting	Details	Complete date
Sequence diagrams	Muhammad Husnain butt	After meeting advised to make class diagrams Also discussed different tools to make diagrams.	Nov 30,2022
Dfd	Muhammad Husnain butt	Dfd 0 and 1, 2 designed.	Dec 5,2022
Revision 1	Muhammad Husnain butt	Revised use extended use cases	Dec 10,2022
Revision 2	Muhammad Husnain butt	Revised ERD diagram	Dec 22,2022
Revision 3	Muhammad Husnain butt	Revised system architecture diagram	Jan 5,2023

Task	Group person available in meeting	Details	Complete date
Revision 4	Muhammad Husnain butt	Revised test cases	July 7,2023
Revision 6	Muhammad Husnain butt	Advised to change some interface	July 22,2023
Final Release	Muhammad Husnain butt	Final System Test	August 1,2023

CONTENTS

DEDICATION	II
ACKNOWLEDGE.....	iv
DECLARATION	vii
CONTENTS.....	XIII
DEFINITIONS AND ACRONYMS.....	XV
LIST OF FIGURES.....	XVI
LIST OF TABLES.....	XVII
1. INTRODUCTION	1
1.1 MOTIVATIONS.....	1
1.2 PROJECT OVERVIEW.....	1
1.3 PROBLEM STATEMENT	1
1.4 OBJECTIVES:	1
2. DOMAIN ANALYSIS	2
2.1 CUSTOMER:	2
2.2 STAKEHOLDERS.....	2
2.3 DEPENDENCIES/ EXTERNAL SYSTEMS	2
2.4 REFERENCE DOCUMENTS	3
2.4.1 Feature Comparison.....	4
3. REQUIREMENTS ANALYSIS.....	5
3.1 REQUIREMENTS.....	5
3.2 LIST OF ACTORS:	7
3.3 LIST OF USE CASES:.....	7
3.4 SYSTEM USE CASE DIAGRAM.....	8
3.5 EXTENDED USE CASE	9
3.6 USER INTERFACES (MOCK SCREENS).....	10
4. DATA FLOW DIAGRAM (OPTIONAL).....	13
4.1 DATA FLOW DIAGRAM LEVEL 0	13
4.2 DATA FLOW DIAGRAM LEVEL 1	13
4.3 DATA FLOW DIAGRAM LEVEL 2	14
5. SYSTEM DESIGN	15
5.1 SYSTEM ARCHITECTURE DIAGRAM.....	15
5.2 SEQUENCE DIAGRAMS.....	16
5.3 ERD	17
5.4 DATA DICTIONARY.....	18
6. IMPLEMENTATION DETAILS	19
6.1 DEVELOPMENT SETUP	19
6.2 DEPLOYMENT SETUP.....	19
6.3 ALGORITHMS	19
6.4 CONSTRAINTS	20
6.4.1 Assumptions.....	20
6.4.2 System constraints	20

6.4.3	<i>Limitations</i>	22
7.	TESTING	23
7.1	EXTENDED TEST CASES	23
	TABLE 7.1: TEST CASE ID 4.....	26
7.2	DECISION TABLE.....	32
7.2.1	<i>Code snippet</i>	32
7.2.2	<i>Decision coverage table</i> :.....	36
7.3	TRACEABILITY MATRIX	37
7.3.1	<i>RID vs TID (requirements vs TEST cases)</i>	37
8.	RESULTS/OUTPUT/STATISTICS	38
8.1	%COMPLETION	38
8.2	%ACCURACY.....	38
8.3	%CORRECTNESS	38
9.	CONCLUSION	39
10.	FUTURE WORK	40
11.	BIBLIOGRAPHY	41
11.1	BOOKS	41
11.2	JOURNALS.....	41
11.3	ARTICLES.....	41
12.	42
12.1	RESEARCH PAPERS	42
12.2	OTHER REFERENCES	42

Definitions and Acronyms

Acronym	Definition
API	Application Programming Interface
ERD	Entity Relationship Diagram
UML	Unified Modeling Language
GUI	Graphical User Interface

Table 1: Table of acronyms and definitions

List of Figures

Figure 3.4: Use Case Diagram.....	08
Figure 3.6: Login Screen (mock screen).....	10
Figure 3.6: Dashboard Screen (mock screen).....	11
Figure 3.6: Add Car (mock screen)	11
Figure 3.6: Delete Car (mock screen)	12
Figure 4.1: Data Flow Diagram (Level 0)	13
Figure 4.2: Data Flow Diagram (Level 1)	13
Figure 4.3: Data Flow Diagram (Level 2)	14
Figure 5.1: System Architecture	15
Figure 5.2: Sequence Diagram.....	16
Figure 5.3: ERD Diagram	17
Figure 6.3: Faster R-cnn algo.....	20
Figure 6.4.2: Camera	21
Figure 6.4.2: Arduino	21
Figure 6.4.2: Sticker	22

List of Tables

Table 2.2: List Of Stakeholders.....	02
Table 3.5: UC-1	09
Table 5.4: Data Dictionary	18
Table 7.1: Test Case 1	23
Table 7.1: Test Case 2	24
Table 7.1: Test Case 3	25
Table 7.1: Test Case 4	26
Table 7.1: Test Case 5	27
Table 7.1: Test Case 6	28
Table 7.1: Test Case 7	29
Table 7.1: Test Case 8	30
Table 7.1: Test Case 9	31
Table 7.2.2: Decision Coverage Table	36
Table 7.3.1: RID vs TID.....	37

List of Codes

Figure 7.2.1: Code snippet	32
Figure 7.2.1: Code snippet	32
Figure 7.2.1: Code snippet	33
Figure 7.2.1: Code snippet	33
Figure 7.2.1: Code snippet	34
Figure 7.2.1: Code snippet	35
Figure 7.2.1: Code snippet	36

1. INTRODUCTION

1.1 Motivations

The main purpose of this project named smart garage is to create a system that works for door opening for save money and safe time.

1.2 Project Overview

In the modern world, almost every person is trying to get everything in automation for their easiness. Generally, the number plates are easily readable for the human being because of their intelligence but it is a difficult task for the computers to do so. We have made this project to increase the security at the gate entrance and to save time and money of the user. The designs of number plates are not convenient because of the variation of parameters like size of numbers, background and type of fonts etc., so we replace this problem by designing a special type of sticker along with a unique code on it. The working principle of smart garage is that it will take a picture of the sticker using a camera and then the captured picture will be transferred to be verified in the database. After the verification, if the data will exist in the database, then the Arduino installed in the smart garage will generate a signal towards the LED light and the LED light will generate a green signal so that the door will be open. For the installation of smart garage, we have also designed the LOGIN and HOMEPAGE. This installation process will be supervised by the ADMIN. In this way, the user will not be required to do any work for running the smart garage.

1.3 Problem Statement

Opening and closing of door is a monotonous job especially for the people who are disabled, it is a very difficult to them to open the door. Secondly in today's world, everyone wants to save their time, so this system will facilitate our user to use their time in an efficient way. Moreover, smart garage will also save their money that the people are using for hiring the gate keepers for security and opening gate nowadays

1.4 Objectives:

The main objective of our system is to completely facilitate the user. Smart garage will let the people save their money and time. This system will also provide us more safety and security.

2. DOMAIN ANALYSIS

2.1 Customer:

Our customers are the users those who will use this smart garage to save time and only authorized user used this system. This is the one-time registration in the database.

2.2 Stakeholders

USER	Most important stakeholder of smart garage are its users. As users will make it successful by connecting this system and fulfill the mission behind this system and we will make it more advanced and helpful for our users in future.
ADMIN	One of our prime users will be admin who will manage this system and enter user information.

Table 2.2: List of Stakeholders

In this table we are discussing our project's stakeholders. A stakeholder is a person who is interested in using our project. In this project we have two stakeholders of our project, a user who use this project in their garages to use the facilities of our project. An Admin who will control all the systematical work of our project.

2.3 Dependencies/ External Systems

- ADMIN
- USER
- DATABASE

2.4 Reference Documents

1. Parking Gate Control Base:

This project is to develop an automatic gate control application which recognizes license plates from cars at entrance gate and take an action to let cars enter or not.

2. Smart garage using android application:

This paper is about developing a gate by the implementation of android applications. This project was controlled by using android application on a phone that was connected by WIFI. The project is limited with a smart gate prototype made of acrylic glass of base size 48cm * 20cm and a DC motor was to open the gate or close the gate.

3. Design and construction of an automatic gate:

This project is on a design and construction of an automatic gate. The gate will react whether a human or a vehicle is coming close to it. The project is 90 % accurate and, in this project, there is no detection component but a sensor is used.

2.4.1 Feature Comparison

Sr.no	Comparison Feature	Parking Gate Control Base	Smart garage
1	Detection of object	This application is basically automatic gate control which recognizes license plates from cars.	In this project a special type of sticker is required. For recognition
2	Gate Opening	This project uses Android Application for opening the gate	Smart Garage will automatically open the gate upon recognition and authentication of sticker
3	Gate opening	This project uses a sensor to open gate upon someone reaching near it regardless of a car or person	This project only opens gate when sticker is identified and recognized

3. REQUIREMENTS ANALYSIS

3.1 Requirements

Introduction:

This chapter of our project tells us about the requirement for making the smart garage. So smart garage is a development that can be used in our daily lives for saving the user time and the money that is used for hiring new gate guard for opening the gate for the user. The following step are the basic scenario for using the smart garage.

- User has to register his name and car sticker unique code in our Arduino.
- A camera is used for identifying the unique code
- Then the code will be transferred for checking in our database, that whether the code is stored in our database or not.
- Then after verification of that unique code the signals from our Arduino will be transferred to a LED light
- The LED Light will be attached to the user gate, by getting the signal it will produce a green signal to open the gate for a required time.
- If the user information is not found in the database, then the door will be closed in that case.

Functional categories,

Functional requirement	<p>There is list of required services for user to use smart garage</p> <p>FR 1: LOGIN</p> <p>An official Admin ID and password is to be entered in order to excess the system.</p> <p>FR 2: Dashboard</p> <p>There will be five components in dashboard that is ADD car, DELETE car, counter of total car registered, START button and END button. The admin can ADD/DELETE car in the system according to the user requirement. Admin can see the total number of registered cars. Admin can start smart garage as well as stop it.</p> <p>FR 3: Unique code sticker</p>
-------------------------------	--

	<p>A special type of sticker is required in using smart garage because it is the main unique code that every user is having for opening the gate. Without the code the smart garage will not work.</p> <p>FR 4: Camera</p> <p>Camera is a basic component for using smart garage because it is used to scan the picture of the unique code for the verifications.</p> <p>FR 5: Arduino</p> <p>Arduino is the second mandatory instrument that is used to enter the unique code and also to verify it from the database so that it can open the gate for the user to enter.</p> <p>FR 6: LED Light</p> <p>When the signal is generated signal through Arduino then the LED light will be turn on red to green</p>
Non-functional requirements	<p>NFR 1: Usability</p> <p>Smart garage is very easy to learn and operate, simply the user will only enter the unique code and the setup is done.</p> <p>NFR 2: Efficiency of use</p> <p>The smart garage is a time efficient development. After the Installation and verification, the smart garage will open the gate within seconds. Basically, it is designed for facilitating in time efficiency for user</p> <p>Secondly, it will also facilitate the user in saving the money. The user will not be spending money in hiring gate guard after using smart garage.</p> <p>NFR 3: Intuitiveness</p> <p>The working principle of smart garage is very easy to understand for a user. The user can easily install the setup because there is a minor installation process.</p> <p>NFR 4: Security</p> <p>Smart garage provides the proper security for a user. It is designed in such a way that there will be no harm in the security requirement. It is much secured because every user of smart garage will be assigned a unique code.</p> <p>NFR 5: Reliability</p> <p>The system of smart garage is very reliable because the user can do unlimited iterations for opening gate, there is no limit on that and the smart garage will not produce any error.</p>

	<p>NFR 6: Performance</p> <p>The quality of smart garage will be assured because it will give a quick response on the user command.</p> <p>NFR 7: Maintenance</p> <p>Smart garage can be maintained easily because if a bug will be generated in it, it will be resolved by the developers very easily.</p>
Data requirement	For data gathering of smart garage, a database is required which will keep the installed code sticker of car so that the user will not do the installation again and again every time. This will make the smart garage more time efficient
Constraints	-----
External interface requirement	The Arduino will be the medium that will be used to transfer the signal to the DC motor after the verification

3.2 List of Actors:

This is the character that are used in smart garage.

- Admin
- User

3.3 List of use cases:

1. Log In: This use case will provide the admin to access the admin site
2. Home page: This use case will provide the admin to change or add or delete a unique sticker code.

3.4 System use case diagram

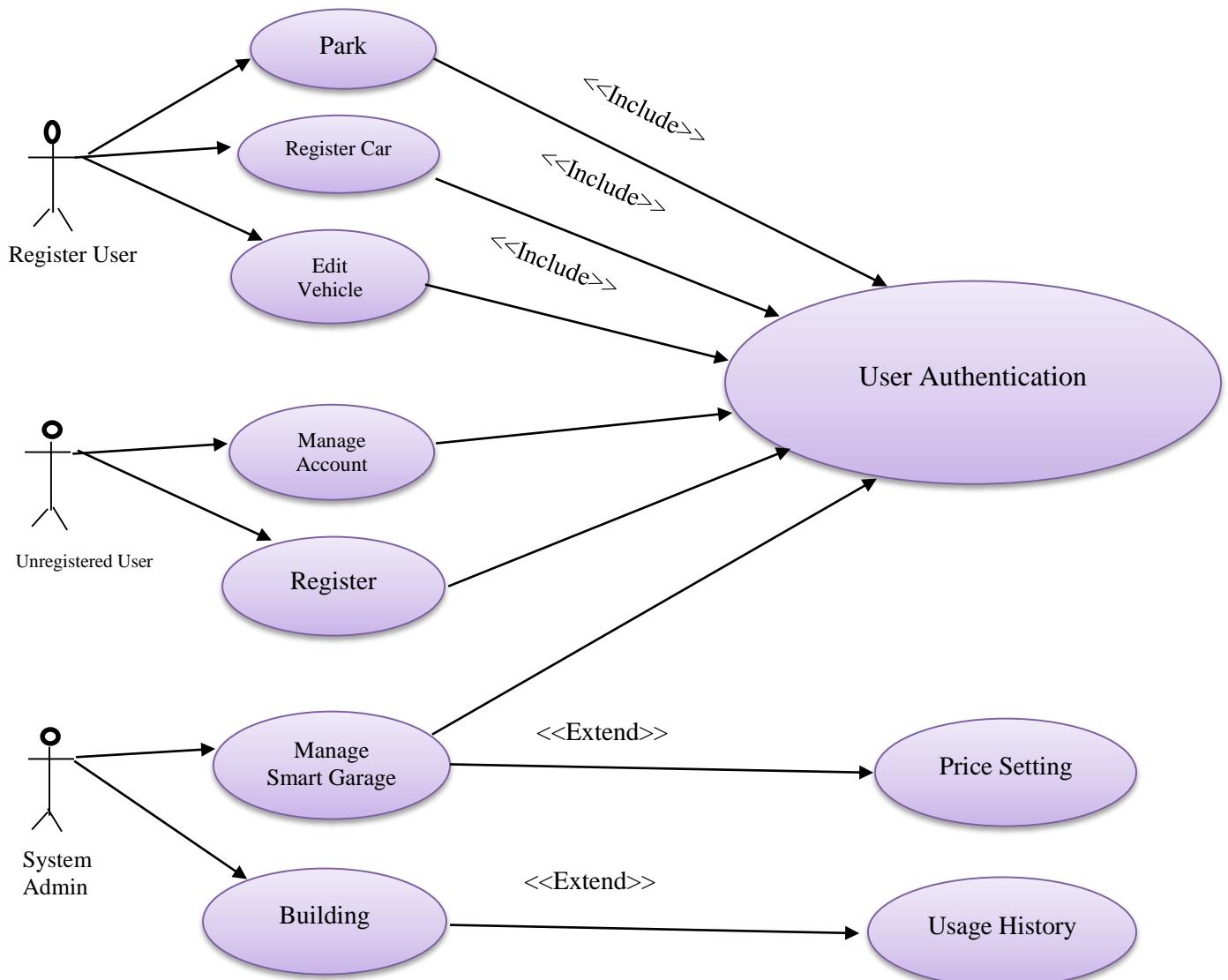


Figure 3.4: Use case diagram

Figure 3.4 shows the use case diagram that is describing all the interaction of the characters with our project.

3.5 Extended use case

Use Case ID:	UC-1		
Use Case Name:	Smart Garage		
Created By:	Sameer Wahid	Last Updated By:	Sameer Wahid
Date Created:		Last Revision Date:	
Actors:	Admin, User		
Description:	The Admin must enter the unique code sticker		
Trigger:	Admin will first login and then after entering the unique code, the user can use the smart garage for that particular code.		
Preconditions:	<ol style="list-style-type: none"> 1. The Admin must have to register the unique code first. 2. The user must agree to all the terms and conditions. 		
Post conditions:	The user can easily use the smart garage after getting registered in it.		
Normal Flow:	<ol style="list-style-type: none"> 1. The admin will first login the system 2. The admin will add the unique code of sticker 3. The code will be registered in the database 4. The system will show a message of “successfully registered” 5. Then the user can get the access through smart garage 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	-----		
Exceptions:	<ol style="list-style-type: none"> 1. If the Admin will not enter any code, it will not register the car for smart garage 2. If the admin entered the wrong code, the system will not recognize it and the gate will not open. 		
Includes:	As it is the start for system so no other use case is required in this use case.		
Frequency of Use:	Unlimited time.		
Special Requirements:	<ol style="list-style-type: none"> 1. Must have a Arduino device 2. Must have a camera 3. Must have a LED Light 		
Assumptions:	The user will not face any difficulty in using smart garage because there is no work required from the user. All the work is linked with the Admin.		

Table 3.5: UC-1

3.6 User interfaces (mock screens)

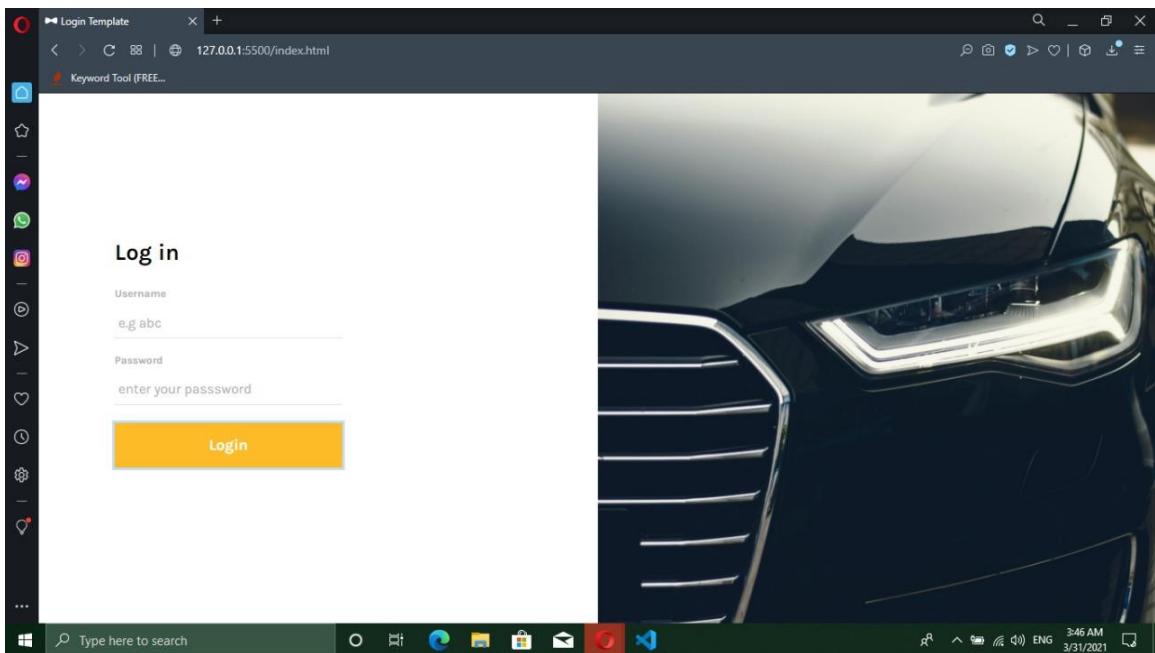


Figure 3.6: Login Screen

This is the login screen of our smart garage for the login of our admin. Figure 3.6 contains a username and password in it. If the admin will put the username and password correctly then he/she can access the smart garage.

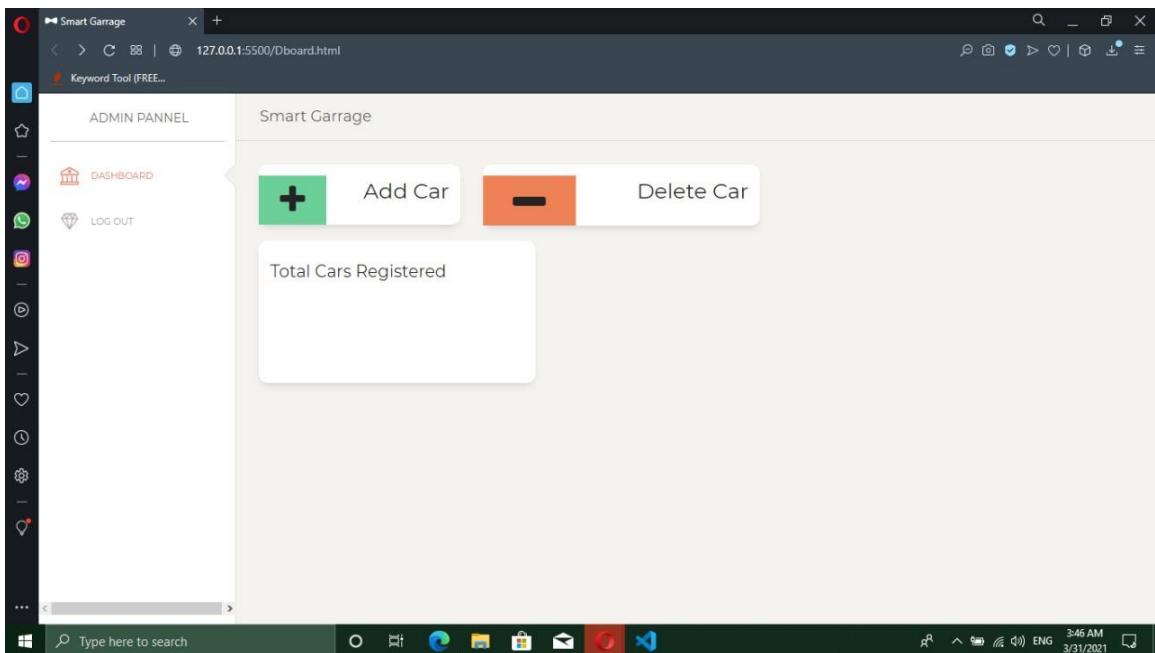


Figure 3.6: DASHBOARD Screen

This is the dashboard here the admin can perform multiple actions. This figure shows two different buttons and one counter screen named as total car register. There is also an option to logout the dashboard.

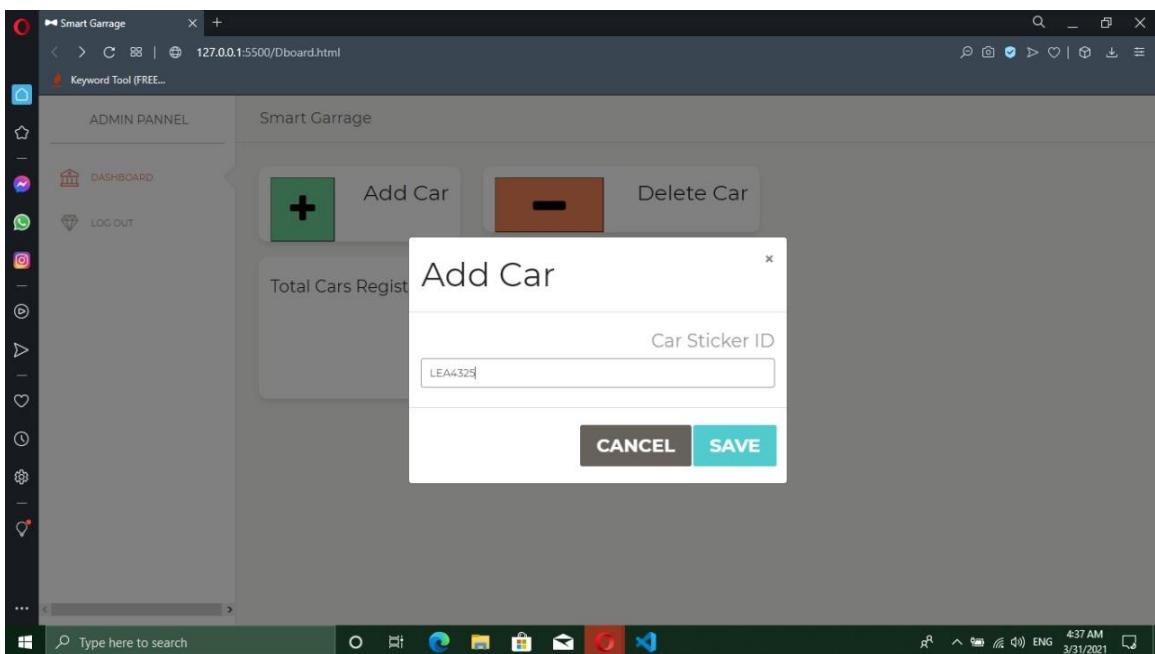


Figure 3.6: ADD CAR

Here add car button functionality is that Admin can enter sticker id and add it in the database. He also can cancel the operation by a cancel button given.

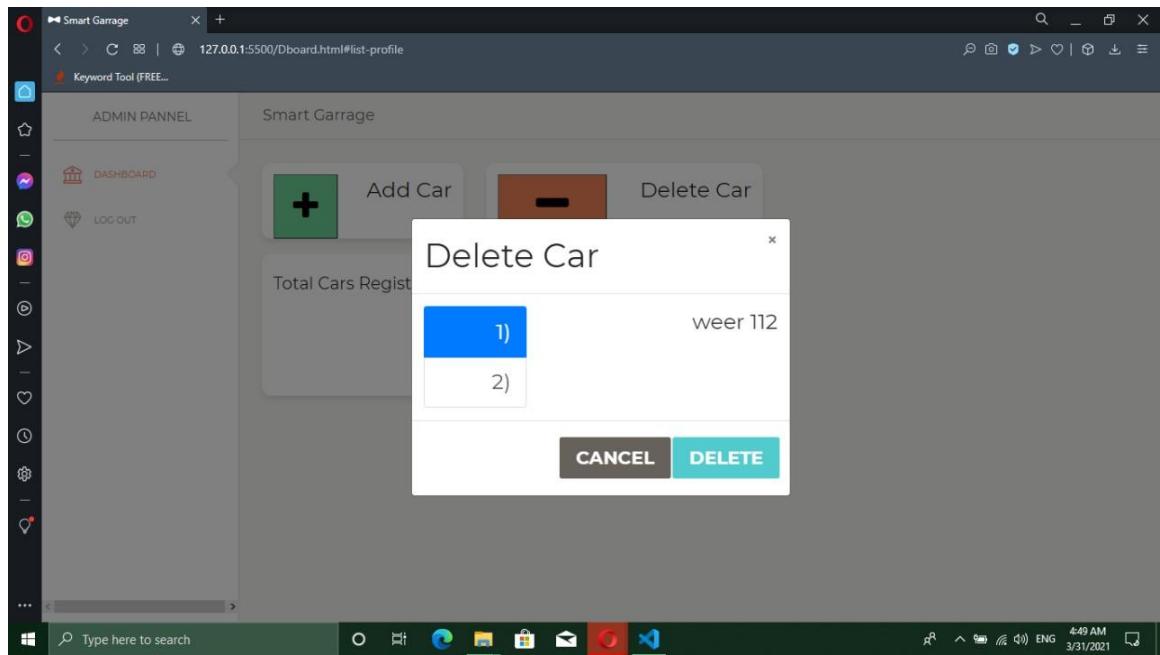


Figure 3.6: DELETE CAR

Here delete car button functionality is that the admin can delete from database and registered cars would be displayed here.

4. DATA FLOW DIAGRAM (OPTIONAL)

4.1 Data Flow Diagram Level 0

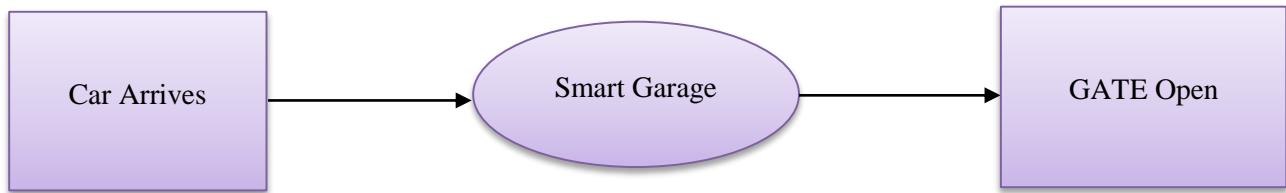


Figure 4.1: Data Flow Diagram

4.2 Data Flow Diagram Level 1

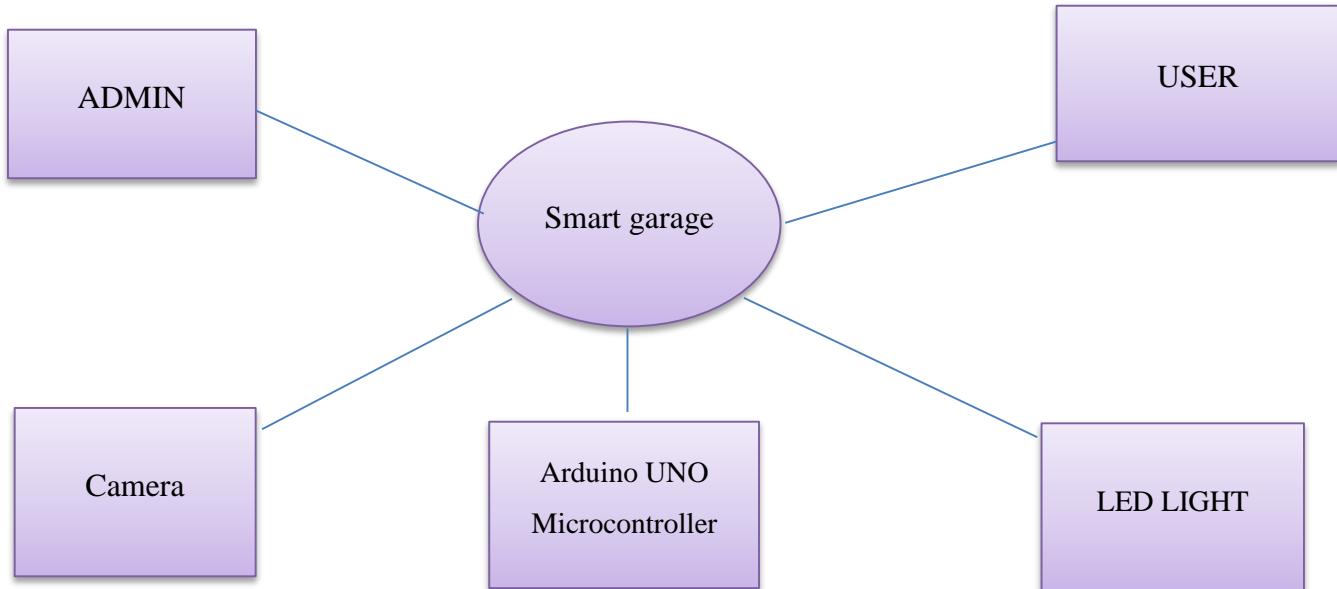


Figure 4.2: Data Flow Diagram

4.3 Data Flow Diagram Level 2

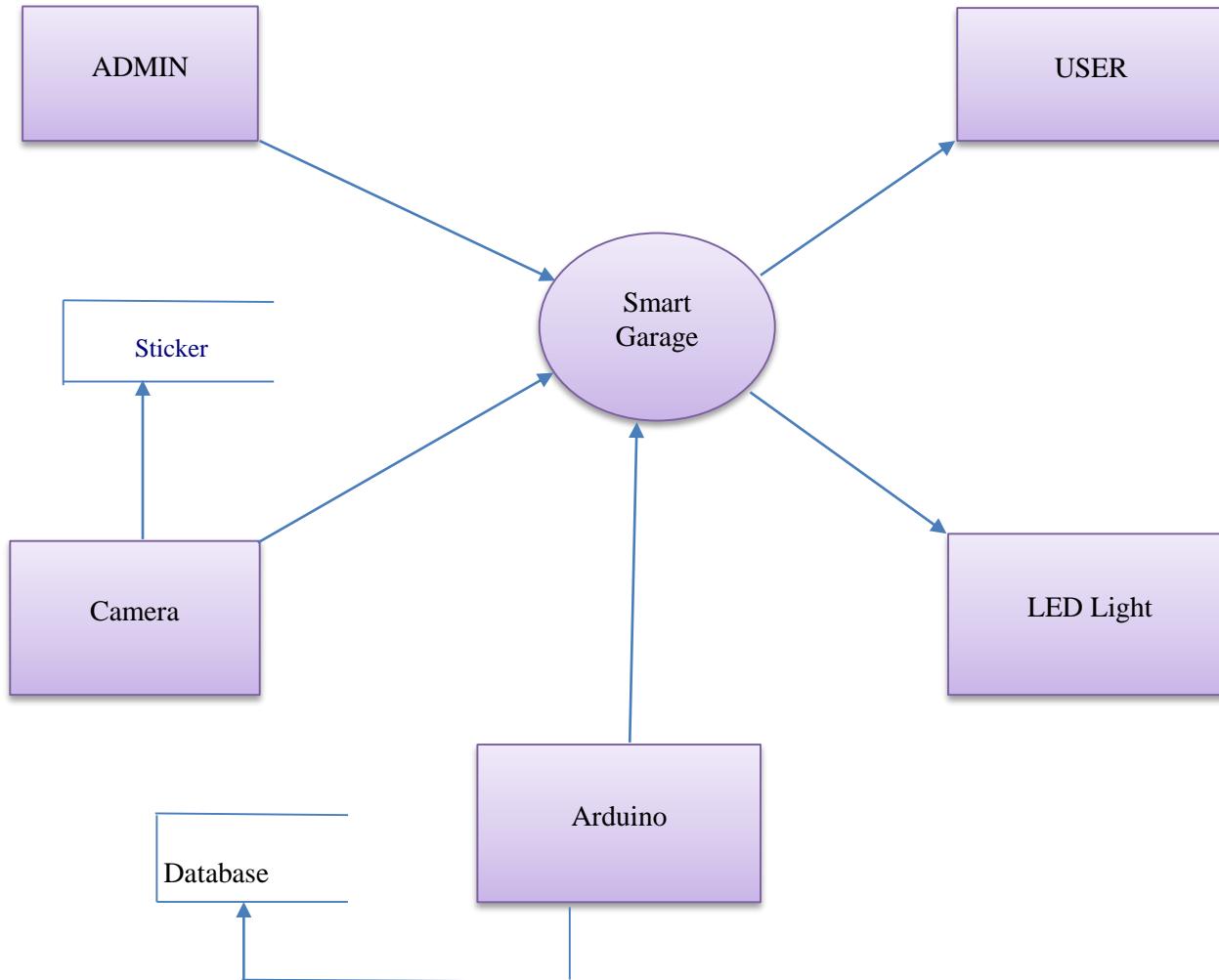


Figure 4.3: Data Flow Diagram

Figure 4.1, 4.2 and 4.3 shows the data flow diagrams, which are representing the flow of our smart garage. These Data flow diagrams are also providing the information about the outputs and inputs of every entity of our project.

5. SYSTEM DESIGN

5.1 System Architecture Diagram

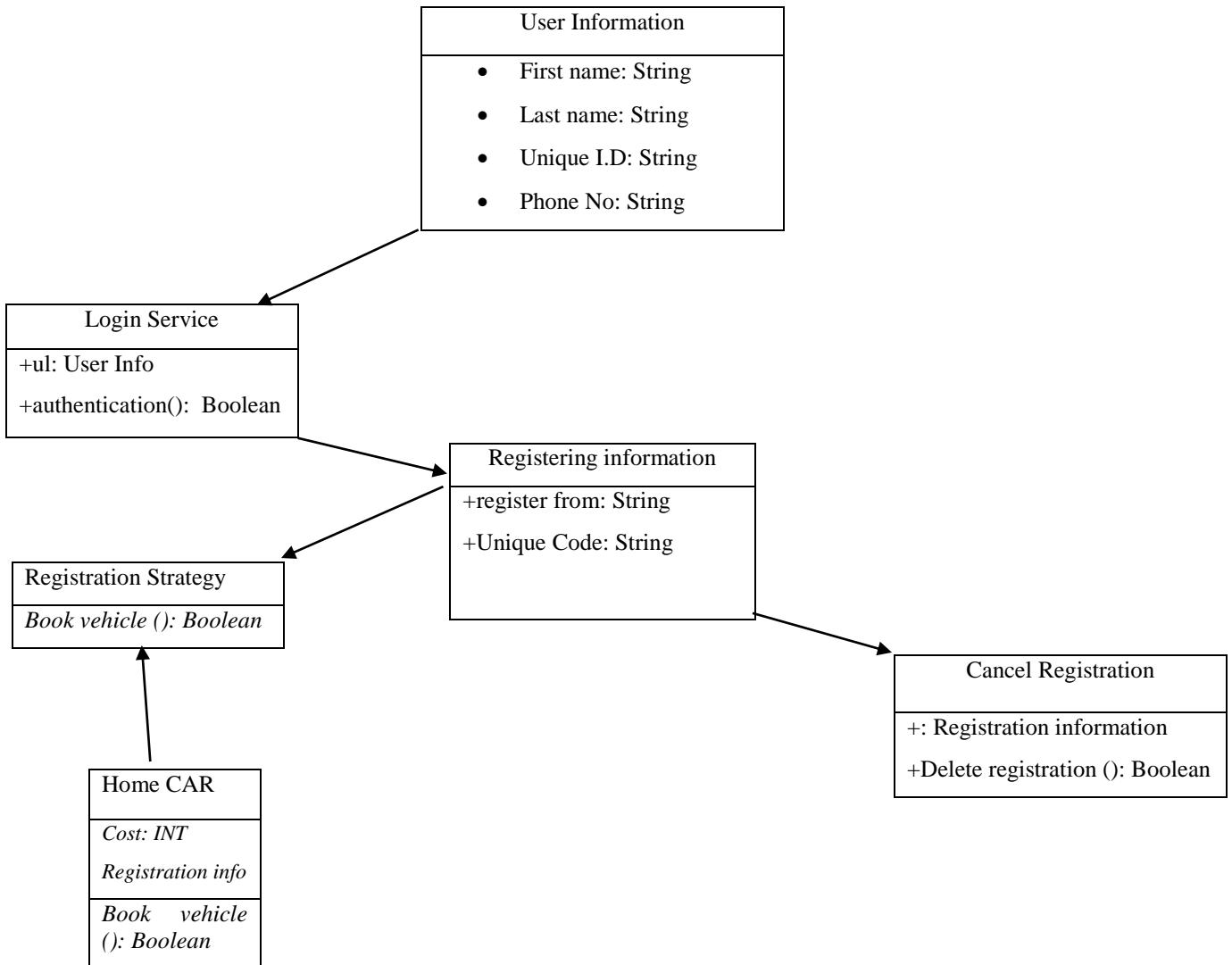


Figure 5.1: System Architecture

5.2 Sequence Diagrams

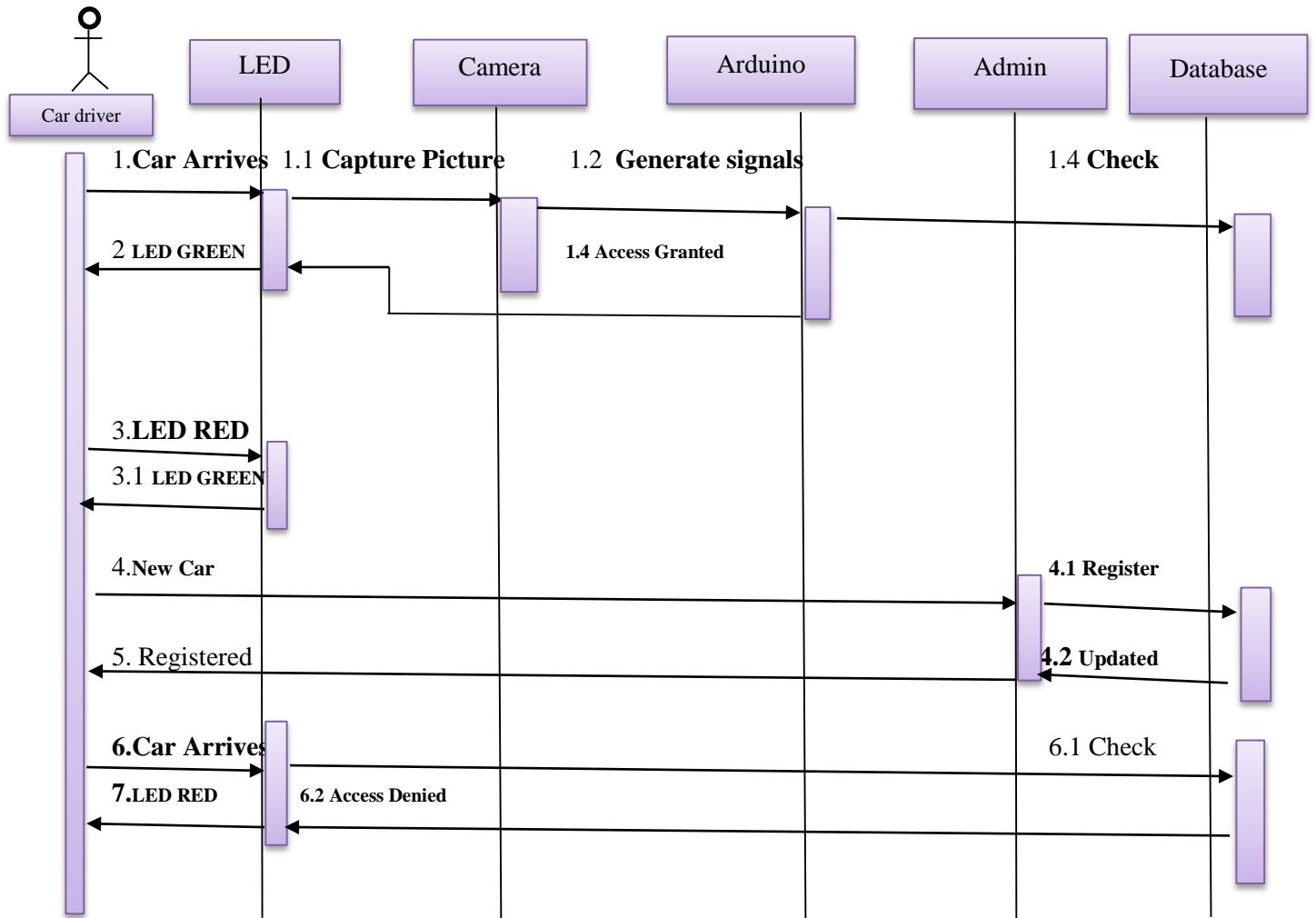


Figure 5.2: Sequence diagram

Figure 5.2 shows the sequence diagram that shows all the interaction of the objects with the sequence of time. All the functionality are arranged with the messages. In this diagram every condition is full-filling the functions of our Smart Garage.

5.3 ERD

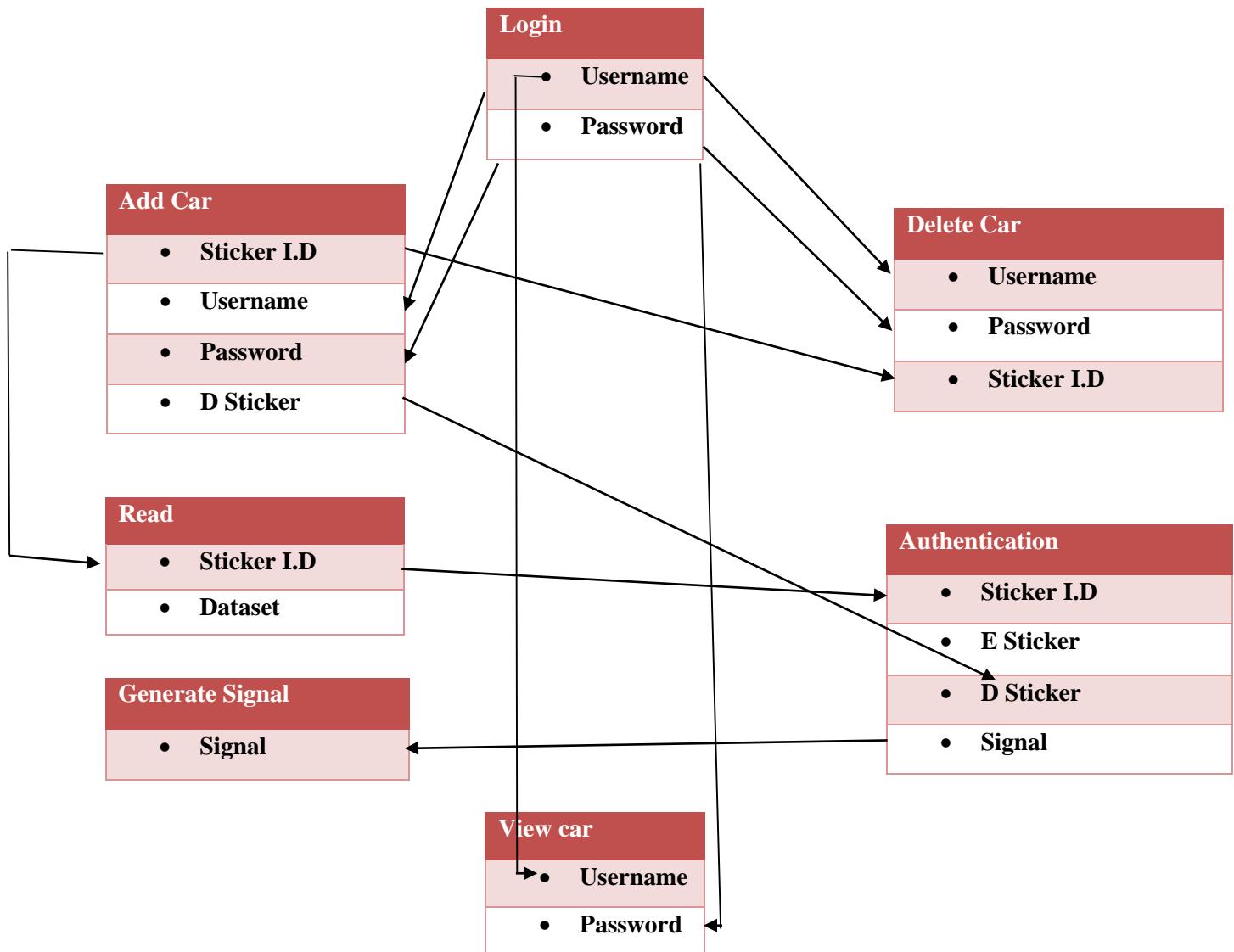


Figure 5.3: ERD diagram

Figure 5.3 shows the Entity- relationship model that is illustrating all the ‘entities’ such as people or objects relate to each other within a system.

5.4 Data Dictionary

Element Name	Type	Validation	Mandatory	Remarks
Username	String	Present	Yes	This is username
Password	String	Range(11<pass>6)	Yes	User's password
Phone	Numbers	Range (11 digits)	Yes	User's contact
Sticker I.D	String	Present	Yes	User's Unique code
D Sticker	String	Present	Yes	User's D Sticker
E Sticker	String	Present	Yes	User's E Sticker
Dataset	String	Present	Yes	This is user's Dataset
Signal	String	Present	Yes	This is Signal

Table 5.4: Data Dictionary

Figure 5.4 shows the data dictionary of all the collection of names, definition and attributes about data elements that are being used in a database.

6. IMPLEMENTATION DETAILS

6.1 Development Setup

Programming Language: python, C++.

Hardware Interface: Personal Computer with GPU

Database: MySQL

Tools: Jupyter Notebook, Google colab, Tensor flow object detection API, EasyOCR.
Faster_r-cnn_resnet101_v1_1024*1024.

6.2 Deployment setup

Smart Garage will be used to automate houses it is a step towards future. Every time a person has to open the garage, he/she has to come out of the car and open it. This project will save a lot of time. This Systems redefines of how garages were, providing an automatic door opening system based upon sticker (unique code) recognition using deep learning. This would be done by using a camera which will capture the front side of the car further an algorithm will be used to judge the position of the sticker then recognize the sticker using optical character recognition and deep learning after that the system will check that whether the recognized sticker is present in the database, depending upon that it will either turn on green or red light (Green if it is present and red if it's not). This smart garage will increase the security and environmental safety and help people in their day to day lives.

6.3 Algorithms

- Transfer learning
- Fast R-cnn

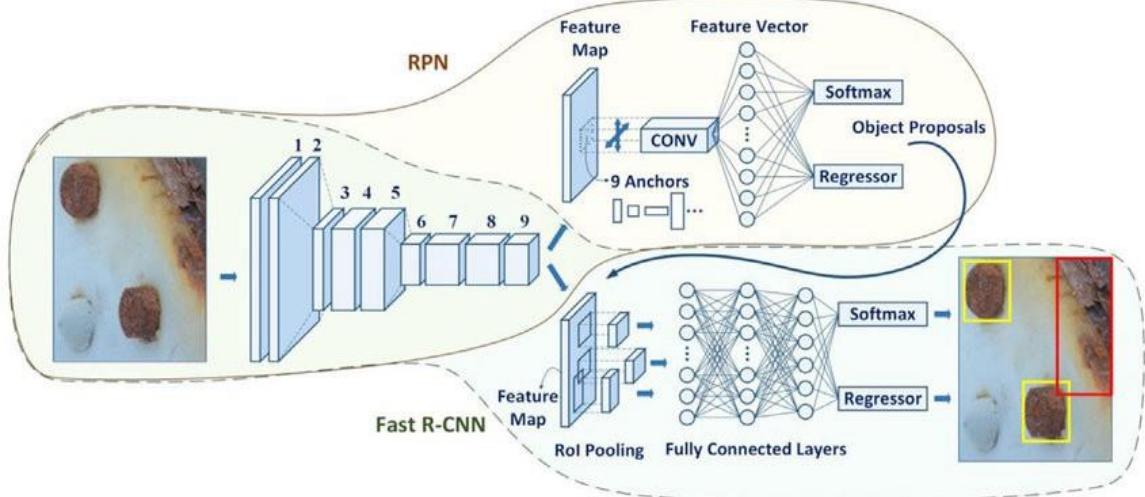


Figure 6.3: faster r-cnn algorithm architecture

Figure 6.3 shows the Faster r-cnn algorithm model that is used for object detection. It has 32 layers. It uses these 32 layers for detection the object and it is used for training the model in detection.

6.4 Constraints

6.4.1 Assumptions

- User have personal computer with GPU installed.
- User have paid the official team for installation.

6.4.2 System constraints

- Windows or Linux
- Database
- Camera



Figure 6.4.2: camera

Figure 6.4.2 shows the Camera, which is the main hardware that will be used for the most important work because it will take the picture of the unique code sticker for recognition.

- **Arduino**



Figure 6.4.2: Arduino

Figure 6.4.2 shows the Arduino Uno that is an open-source microcontroller board based on the Microchip Atmega328P microcontroller and developed by Arduino.cc. The Arduino is used for generating the signal for turning the LED Green from Red.

- **Sticker**



Figure 6.4.2: Code Sticker

6.4.3 Limitations

- Smart garage team is installing the setup only in Pakistan.
- Lighting conditions effects accuracy of the model.
- Lower GPU machines will have performance issues.

7. TESTING

7.1 Extended Test Cases

Table 7.1: Test Case ID

Test Case ID: 1		Test Design By: Mohammad Khan									
Test Module Name: Sign in button on interface		Test Design Date: 7-07-21									
Test Priority: High		Test Executed By: Mohammad khan									
Test Title/Name: To test Sign in button		Test Executed Date: 7-07-21									
Description: To test if sign in button is working fine on official account											
Pre-Condition: Interface must be running, and admin must be signed in on official account											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Navigate to sign in page				pass	admin must enter correct information to get access, and if that email is not available in database it will not access.					
2	Click on sign in text field										
3	Fill the form according to the format	Email: test1@gmail.com Password: log123									
4	Click on sign in button		The admin is logged into interface as his/her record is already available in database	admin signed in successfully.							
Post Condition: Admin has logged in, and has full access to smart garage.											

Table 7.1: Test Case ID 2

Test Case ID: 2		Test Design By: Mohammad Khan									
Test Module Name: selecting add car		Test Design Date: 7-07-21									
Test Priority: High		Test Executed By: Mohammad Khan									
Test Title/Name: selecting add car test		Test Executed Date: 7-07-21									
Description: Admin can add a new car.											
Pre-Condition: Admin must sign it to the interface.											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Navigate to add car				Pass	The admin can only enter the car.					
2	Select add car to add a new car		Open dialogue box	opened successfully							
3	We have to enter the unique code sticker										
4	click on ok button to add car on screen		We want to add a car	Entered successfully							
Post Condition: The Admin has successfully entered the car.											

Table 7.1: Test Case ID 3

Test Case ID: 3		Test Design By: Mohammad Khan									
Test Module Name: Selecting delete car		Test Design Date: 7-07-21									
Test Priority: High		Test Executed By: Mohammad Khan									
Test Title/Name: Selecting delete car test		Test Executed Date: 7-07-21									
Description: Admin can delete a car.											
Pre-Condition: Admin must sign it to the interface.											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Navigate to delete car				Pass	The admin can only delete the car.					
2	Select delete car to delete a car		Open dialogue box	opened successfully							
3	We want to delete car from the list										
4	click on bin button to delete car		We want to delete a car	Deleted successfully							
Post Condition: The Admin has successfully entered the car.											

Table 7.1: Test Case ID 4

Test Case ID: 4		Test Design By: Mohammad Khan									
Test Module Name: Total car registered		Test Design Date: 7-07-21									
Test Priority: High		Test Executed By: Mohammad khan									
Test Title/Name: Total car registered		Test Executed Date: 7-07-21									
Description: Admin can see the total number of registered cars.											
Pre-Condition: Admin must sign it to the interface.											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Navigate to total car registered				Pass	The admin can see the count of total cars registered.					
2	To find total car registered after adding/deleting		Count of Total cars registered	Calculated successfully							
Post Condition: The Admin can see the correct count of total car registered successfully.											

Table 7.1: Test Case ID 5

Test Case ID: 5		Test Design By: Mohammad khan									
Test Module Name: Starting smart garage system		Test Design Date: 7-07-21									
Test Priority: High		Test Executed By: Mohammad khan									
Test Title/Name: Starting smart garage test		Test Executed Date: 7-07-21									
Description: Admin can run the smart garage system.											
Pre-Condition: Admin must sign it to the interface.											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Navigate to start the system				Pass	The admin can start/run the system.					
2	To start smart garage		Start the system	Started successfully							
Post Condition: The Admin started the smart garage system successfully.											

Table 7.1: Test Case ID 6

Test Case ID: 6	Test Design By: Mohammad khan									
Test Module Name: Ending the smart garage system	Test Design Date: 7-07-21									
Test Priority: High	Test Executed By: Mohammad khan									
Test Title/Name: Ending the smart garage system.	Test Executed Date: 7-07-21									
Description: Admin can end the smart garage system.										
Pre-Condition: Admin must sign it to the interface.										
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note				
1	Navigate to log out				Pass	The admin can stop/close the system.				
2	Log out system		Close the system	Closed successfully						
Post Condition: The Admin has ended the smart garage successfully.										

Table 7.1: Test Case ID 7

Test Case ID: 7		Test Design By: Sameer Wahid									
Test Module Name: Recognizing Unique Code Sticker		Test Design Date: 8-07-21									
Test Priority: High		Test Executed By: Sameer Wahid									
Test Title/Name: To recognize unique code sticker		Test Executed Date: 8-07-21									
Description: To test whether the sticker is recognized correctly or not.											
Pre-Condition: Interface must be running.											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Click on the start button on dashboard				pass	Admin must be in the dashboard					
2	Object detection script will start running										
3	Navigate into smart garage directory										
4	Navigate to the file sticker.csv										
5	Open and check the data stored in csv file		The csv file has the value in it	The csv file is containing the value							
Post Condition: Csv file is now containing the sticker value.											

Table 7.1: Test Case ID 8

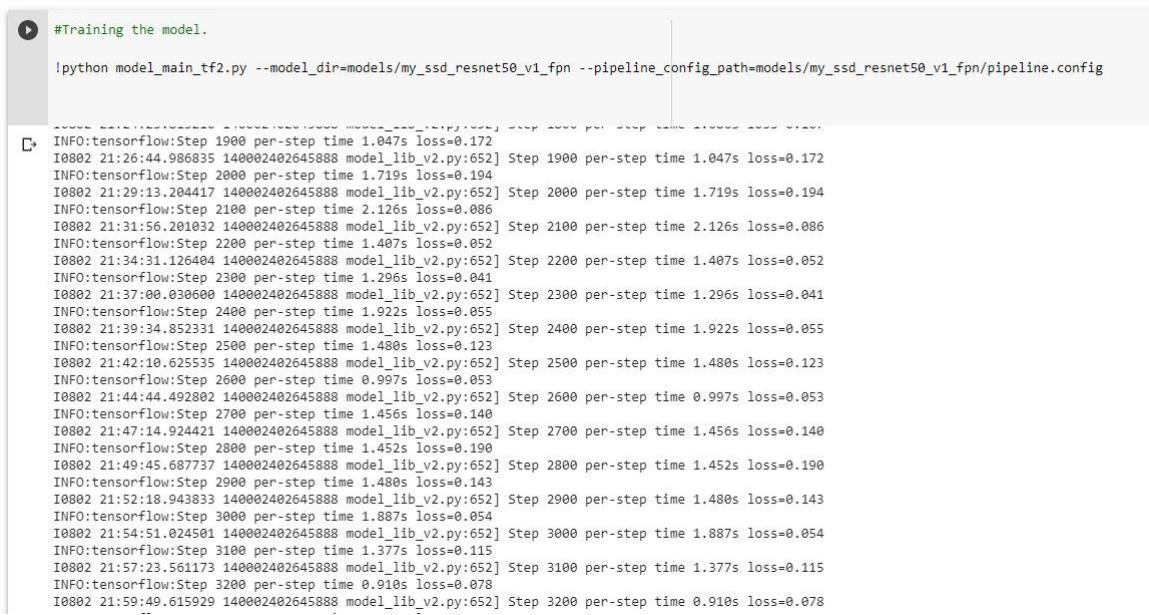
Test Case ID: 8		Test Design By: Sameer Wahid									
Test Module Name: Checking the functioning of camera		Test Design Date: 8-07-21									
Test Priority: High		Test Executed By: Sameer Wahid									
Test Title/Name: To test the functionality of Camera.		Test Executed Date: 8-07-21									
Description: To test if the Camera is functioning properly or not											
Pre-Condition: Interface must be running and the start button is Clicked											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Click on the start button				pass						
2	Camera is on		Camera should be running	Camera is running							
Post Condition: The smart garage has started recognizing sticker.											

Table 7.1: Test Case ID 9

Test Case ID: 9		Test Design By: Sameer Wahid									
Test Module Name: Testing the Arduino		Test Design Date: 8-07-21									
Test Priority: High		Test Executed By: Sameer Wahid									
Test Title/Name: To test Arduino		Test Executed Date: 8-07-21									
Description: To test if the Arduino is generating the signal to turn the LED green.											
Pre-Condition: Interface must be running, and camera is working properly											
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Note					
1	Object detection model has recognize the sticker and stored it in the csv file				pass						
2	Arduino script will read the sticker csv file and authenticate sticker value		The Arduino will generate the green light	The Arduino generated green light							
Post Condition: No post condition											

7.2 Decision Table

7.2.1 Code snippet

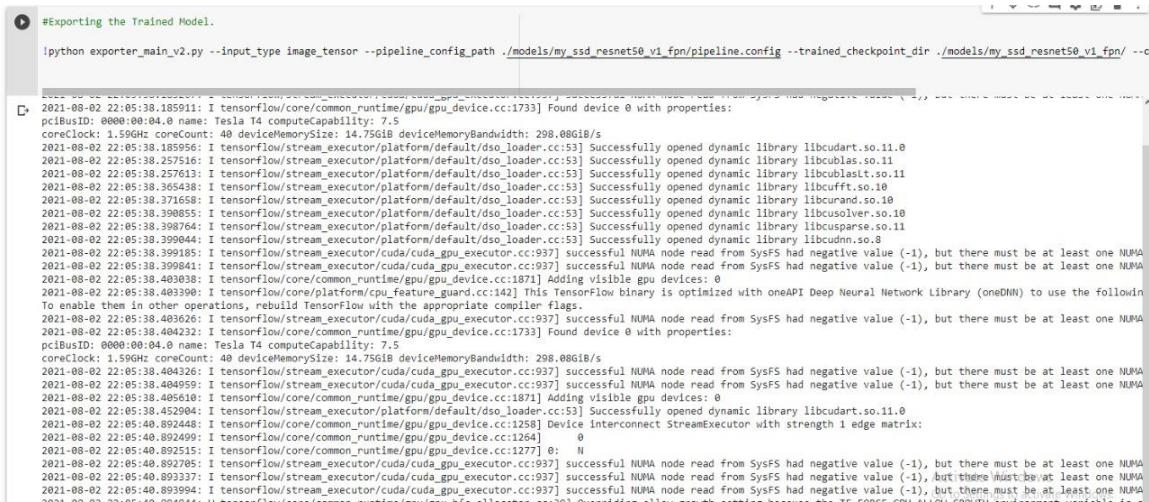


```
#Training the model.

!python model_main_tf2.py --model_dir=models/my_ssd_resnet50_v1_fpn --pipeline_config_path=models/my_ssd_resnet50_v1_fpn/pipeline.config

INFO:tensorflow:Step 1980 per-step time 1.047s loss=0.172
I0802 21:26:44.986835 140002402645888 model_lib_v2.py:652] Step 1900 per-step time 1.047s loss=0.172
INFO:tensorflow:Step 2000 per-step time 1.719s loss=0.194
I0802 21:29:13.204417 140002402645888 model_lib_v2.py:652] Step 2000 per-step time 1.719s loss=0.194
INFO:tensorflow:Step 2100 per-step time 2.126s loss=0.086
I0802 21:31:56.201032 140002402645888 model_lib_v2.py:652] Step 2100 per-step time 2.126s loss=0.086
INFO:tensorflow:Step 2200 per-step time 1.407s loss=0.052
I0802 21:34:31.126404 140002402645888 model_lib_v2.py:652] Step 2200 per-step time 1.407s loss=0.052
INFO:tensorflow:Step 2300 per-step time 1.296s loss=0.041
I0802 21:37:00.030000 140002402645888 model_lib_v2.py:652] Step 2300 per-step time 1.296s loss=0.041
INFO:tensorflow:Step 2400 per-step time 1.92s loss=0.055
I0802 21:39:34.852331 140002402645888 model_lib_v2.py:652] Step 2400 per-step time 1.92s loss=0.055
INFO:tensorflow:Step 2500 per-step time 1.480s loss=0.123
I0802 21:42:10.625535 140002402645888 model_lib_v2.py:652] Step 2500 per-step time 1.480s loss=0.123
INFO:tensorflow:Step 2600 per-step time 0.997s loss=0.053
I0802 21:44:44.492802 140002402645888 model_lib_v2.py:652] Step 2600 per-step time 0.997s loss=0.053
INFO:tensorflow:Step 2700 per-step time 1.456s loss=0.140
I0802 21:47:14.924421 140002402645888 model_lib_v2.py:652] Step 2700 per-step time 1.456s loss=0.140
INFO:tensorflow:Step 2800 per-step time 1.452s loss=0.190
I0802 21:49:45.687737 140002402645888 model_lib_v2.py:652] Step 2800 per-step time 1.452s loss=0.190
INFO:tensorflow:Step 2900 per-step time 1.480s loss=0.143
I0802 21:52:18.943833 140002402645888 model_lib_v2.py:652] Step 2900 per-step time 1.480s loss=0.143
INFO:tensorflow:Step 3000 per-step time 1.887s loss=0.054
I0802 21:54:51.024501 140002402645888 model_lib_v2.py:652] Step 3000 per-step time 1.887s loss=0.054
INFO:tensorflow:Step 3100 per-step time 1.377s loss=0.115
I0802 21:57:23.561173 140002402645888 model_lib_v2.py:652] Step 3100 per-step time 1.377s loss=0.115
INFO:tensorflow:Step 3200 per-step time 0.910s loss=0.078
I0802 21:59:49.615929 140002402645888 model_lib_v2.py:652] Step 3200 per-step time 0.910s loss=0.078
```

Figure 7.2.1: Training the Model



```
#Exporting the Trained Model.

!python exporter_main_v2.py --input_type image_tensor --pipeline_config_path ./models/my_ssd_resnet50_v1_fpn/pipeline.config --trained_checkpoint_dir ./models/my_ssd_resnet50_v1_fpn/ --c

INFO:tensorflow:Found device 0 with properties:
pcibusID: 0000:00:04.0 name: Tesla T4 computeCapability: 7.5
coreClock: 1.59GHz coreCount: 48 deviceMemorySize: 14.75GiB deviceMemoryBandwidth: 298.08GiB

2021-08-02 22:05:38.185911: I tensorflow/core/common_runtime/gpu/device.cc:1733] Found device 0 with properties:
pcibusID: 0000:00:04.0 name: Tesla T4 computeCapability: 7.5
coreClock: 1.59GHz coreMemorySize: 14.75GiB deviceMemoryBandwidth: 298.08GiB
2021-08-02 22:05:38.185911: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcudart.so.11.0
2021-08-02 22:05:38.185911: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcudaz.so.11
2021-08-02 22:05:38.257612: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcublas.so.11
2021-08-02 22:05:38.305438: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcufft.so.11
2021-08-02 22:05:38.371658: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcurland.so.10
2021-08-02 22:05:38.390855: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcusolver.so.10
2021-08-02 22:05:38.398764: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcusparse.so.10
2021-08-02 22:05:38.399044: I tensorflow/core/common_runtime/gpu/device.cc:1733] Successfully opened dynamic library libcudnn.so.8
2021-08-02 22:05:38.399185: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.399841: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.403038: I tensorflow/core/common_runtime/gpu/device.cc:1733] Adding visible gpu devices: 0
2021-08-02 22:05:38.403390: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following To enable them in other operations, rebuild TensorFlow with the appropriate compile flags.
2021-08-02 22:05:38.403626: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.404232: I tensorflow/core/common_runtime/gpu/device.cc:1733] Found device 0 with properties:
pcibusID: 0000:00:04.0 name: Tesla T4 computeCapability: 7.5
coreClock: 1.59GHz coreCount: 48 deviceMemorySize: 14.75GiB deviceMemoryBandwidth: 298.08GiB

2021-08-02 22:05:38.404232: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.404527: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.404807: I tensorflow/core/common_runtime/gpu/device.cc:1733] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:38.405108: I tensorflow/core/common_runtime/gpu/device.cc:1733] successfully opened dynamic library libcudart.so.11.0
2021-08-02 22:05:38.405204: I tensorflow/core/common_runtime/gpu/device.cc:1733] successfully opened dynamic library libdevice.so.1.0
2021-08-02 22:05:38.405204: I tensorflow/core/common_runtime/gpu/device.cc:1738] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-08-02 22:05:40.892448: I tensorflow/core/common_runtime/gpu/device.cc:1264] 0: N
2021-08-02 22:05:40.892515: I tensorflow/core/common_runtime/gpu/device.cc:1277] 0: N
2021-08-02 22:05:40.892765: I tensorflow/core/common_runtime/gpu/device.cc:1277] 0: N
2021-08-02 22:05:40.893337: I tensorflow/core/common_runtime/gpu/device.cc:1277] 0: N
2021-08-02 22:05:40.893337: I tensorflow/core/common_runtime/gpu/device.cc:1277] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:40.893337: I tensorflow/core/common_runtime/gpu/device.cc:1277] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
2021-08-02 22:05:40.893994: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node
```

Figure 7.2.1: Exporting the Trained Model



```
#Loading the saved_model
import tensorflow as tf
import time
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils

PATH_TO_SAVED_MODEL="/content/gdrive/My Drive/TensorFlow/workspace/training_demo/exported-models/my_model/saved_model"
#PATH_TO_SAVED_MODEL="/content/gdrive/My Drive/TensorFlow/workspace/training_demo/models/my_ssd_resnet50_v1_fpn"

print('Loading model...', end='')
# Load saved model and build the detection function
detect_fn=tf.saved_model.load(PATH_TO_SAVED_MODEL)

print('Done!')
Loading model...Done!
```

Figure7.2.1: Loading the Saved Model



```
[ ] #Step 18- Testing the Model.

#Loading the label_map
category_index=label_map_util.create_category_index_from_labelmap("/content/gdrive/My Drive/TensorFlow/workspace/training_demo/annotations/label_map.pbtxt",use_display_name=True)

#category_index=label_map_util.create_category_index_from_labelmap([path_to_label_map],use_display_name=True)

[ ] #Step 19- Testing the Model.

#Loading the image
img=['/content/gdrive/MyDrive/TensorFlow/workspace/training_demo/images/T4.jpg','/content/gdrive/MyDrive/TensorFlow/workspace/training_demo/images/T6.jpg']
print(img)

#list containing paths of all the imag
['/content/gdrive/MyDrive/TensorFlow/workspace/training_demo/images/T4.jpg', '/content/gdrive/MyDrive/TensorFlow/workspace/training_demo/images/T6.jpg']
```

Figure 7.2.1: Testing the Model

```
▶ import numpy as np
import tensorflow as tf
from PIL import Image
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore') # Suppress Matplotlib warnings

def load_image_into_numpy_array(path):
    """Load an image from file into a numpy array.

    Puts image into numpy array to feed into tensorflow graph.
    Note that by convention we put it into a numpy array with shape
    (height, width, channels), where channels=3 for RGB.

    Args:
        path: the file path to the image

    Returns:
        uint8 numpy array with shape (img_height, img_width, 3)
    """
    return np.array(Image.open(path))

for image_path in img:
    print('Running inference for {}...'.format(image_path), end='')
    image_np=load_image_into_numpy_array(image_path)

    input_tensor=tf.convert_to_tensor(image_np)

    input_tensor=input_tensor[tf.newaxis, ...]
```

Figure 7.2.1: Detecting the sticker

```
detections=detect_fn(input_tensor)

num_detections=int(detections.pop('num_detections'))
detections={key:value[0,:num_detections].numpy()
            for key,value in detections.items()}
detections['num_detections']=num_detections

detections['detection_classes']=detections['detection_classes'].astype(np.int64)

image_np_with_detections=image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections,
    detections['detection_boxes'],
    detections['detection_classes'],
    detections['detection_scores'],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=1,      #max number of bounding boxes in the image
    min_score_thresh=.70,     #min prediction threshold
    agnostic_mode=False)
%matplotlib inline
plt.figure()
plt.imshow(image_np_with_detections)
print('Done')
plt.show()
```

Figure 7.2.1: Detecting the Sticker

```

import easyocr
import cv2
detection_threshold = 0.7

image = image_np_with_detections
scores = list(filter(lambda x: x > detection_threshold, detections['detection_scores']))
boxes = detections['detection_boxes'][:len(scores)]
classes = detections['detection_classes'][:len(scores)]

width = image.shape[1]
height = image.shape[0]

for idx, box in enumerate(boxes):
    roi = box*[height, width, height, width]
    region = image[int(roi[0]):int(roi[2]),int(roi[1]):int(roi[3])]

    im = cv2.cvtColor(region,cv2.COLOR_BGR2GRAY)

    # histogram equalization
    equ = cv2.equalizeHist(im)
    # Gaussian blur
    blur = cv2.GaussianBlur(equ, (5, 5), 1)

    # manual thresholding
    th2 = 70
    equ[equ>=th2] = 255
    equ[equ<th2] = 0

#easy ocr
reader = easyocr.Reader(['en'])
ocr_result = reader.readtext(equ)
plt.imshow(region)
print(ocr_result)

```

Figure 7.2.1: Reading the Sticker

7.2.2 Decision coverage table:

Conditions	Username	Blank	Blank	Blank	Invalid	Invalid	Invalid	Valid	Valid	Valid
Conditions	Password	Blank	Invalid	Valid	Blank	Invalid	Valid	Blank	Invalid	Valid
Actions	Expected Result	Error: please enter username	Error: please enter username	Error: please enter username	Error: please enter Valid username	Error: Login failed	Error: please enter valid username	Error: please enter password	Error: Login failed	
Actions	Login page	Login page	Login page	Login page	Login page	Login page	Login page	Login page	Login page	Dashboard

Table 7.2.2: Decision Coverage Table

7.3 Traceability Matrix

7.3.1 RID vs TID (requirements vs TEST cases)

TID / RID	R1	R2	R3	R4	R5
TC-01	✓				✓
TC-02		✓	✓		
TC-03		✓	✓		
TC-04		✓			
TC-05	✓	✓	✓	✓	✓
TC-06	✓	✓	✓	✓	✓
TC-07	✓	✓	✓		
TC-08	✓	✓		✓	
TC-09	✓	✓			✓

Table 7.3.1 : RID vs TID

8. RESULTS/OUTPUT/STATISTICS

8.1 %completion

We have one use case and 6 requirements in our project. 95% of requirements are completed by the use case.

8.2 %accuracy

When adding a new car our project is validating for empty fields. Almost 96% of our work is accurate.

8.3 %correctness

Our system results are 98% accurate.

9. CONCLUSION

Smart Garage will be used to automate houses it is a step towards future. Every time a person has to open the Garage, he/she has to come out of the car and open it. This project will save a lot of time. This Systems redefines of how garages were, providing an automatic door opening system based upon sticker (unique code) recognition using deep learning. This would be done by using a camera which will capture the front side of the car further an algorithm will be used to judge the position of the sticker then recognize the sticker using optical character recognition and deep learning after that the system will check that weather the recognized sticker is present in the database, depending upon that it will either turn on green or red light (Green if it is present and red if it's not). This smart garage will increase the security and environmental safety and help people in their day to day lives. The Smart garage will bring innovation.

10. FUTURE WORK

In future we want to add more additional features in this project to enhance it for the betterment of parking lots because we can take our smart garage in big parking lots projects. This project is much related to the parking lots requirement because this project fulfills the requirement of a parking lot.

Secondly, smart garage can also help in electronic challan technology. Our project will need a lit bit more development for completing the needs of electronic challan system.

11. BIBLIOGRAPHY

11.1 Books

- i. Deep Learning with Python
Book by François Chollet

11.2 Journals

NONE,

11.3 Articles

- i. How to use AMD GPU's for machine learning on windows
<https://medium.com/swlh/how-to-use-amd-gpus-for-machine-learning-on-windows-96ace916e97>
- ii. Google colab free GPU tutorial
<https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>
- iii. A complete guide to google colab for deep learning
<https://www.kdnuggets.com/2020/06/google-colab-deep-learning.html>

12.

12.1 Research papers

- i. bin Mohd Zafie, Mohamad Afiq Afifi. "Smart Gate Using Android Applications." *Journal of Physics: Conference Series*. Vol. 1755. No. 1. IOP Publishing, 2021.
- ii. Ikpeze, Onyinye Florence, et al. "Design and construction of an automatic gate." *ABUAD J. Eng. Res. Dev* 2.2 (2019): 123-131.
- iii. Nguyen, Nhat-Duy, et al. "An evaluation of deep learning methods for small object detection." *Journal of Electrical and Computer Engineering* 2020 (2020).

12.2 Other References

NONE.



Estd. 1990

**Final Year Project Report
Project Name: NFT Connect
Project Advisor:
Dr. Saeed Ahmed
Submitted By:**

**Abdullah Tanvir (F2019266268)
Noureen Butt (F2019266313)
Durrez Ahmad (F2019266275)
Danish Ahmed (F2019266359)**

Session (2019-2023)

**University of Management and Technology
C-II Johar Town Lahore Pakistan**

Dedication

This project is dedicated to Allah, our Creator and Prophet ﷺ who blessed us with the required resources that led us to the completion of this project. We would like to mention Dr. Saeed Ahmed who gave us the opportunity to work under her guidance. Lastly, we dedicate this project to our parents who were always there to support & provide us with the resources and help we needed throughout this project.

Final Approval

- **Head of Department**

Department of Informatics & Systems
School of Systems & Technology
UMT Lahore

- **Program Director (Final Year Projects)**

Department of Informatics & Systems.
School of Systems & Technology
UMT Lahore

- **Supervisor** _____

- Department of Informatics & Systems.
- School of Systems & Technology
UMT Lahore

- **Co-Supervisor** _____

Acknowledgment

We would like to acknowledge our advisor Dr. Saeed Ahmed who assisted us during this project. He kept us updated about the latest technologies to use and helped us to get through any ambiguities throughout the project. Next, we would like to acknowledge our evaluators and directors, their feedback and remarks helped us to evolve our project immensely. Lastly, we would like to acknowledge our university UMT for providing us with a platform to work on a project in the form of a team.

Project Title: NFT connect

Objective

- User friendly and eye catchy interface to provide excellent experience to the user.
- Create, buy and sell NFTs.
- A secure and immutable platform introduced to ensure the integrity of data.

Undertaken by

Abdullah Tanvir (f2019266268)
Noureen Butt (f2019266313)
Durrez Ahmad (f2019266275)
Danish Ahmed (f2019266359)

Supervised by Dr. Saeed Ahmed

Starting Date: October 10, 2022

Completion Date: January 25, 2023

Tools Used ReactJS CSS, NodeJS, MongoDB, JavaScript, Ethereum/Solana, Solidity

Operating System Windows 10

Documentation

Abstract

The world of technology has been changing with time. The most recent revolution in technical world is introduced which is the accouplement of web 3. Data is replicated over the blockchain network and each block is cryptographically connected to the one before it. Because the block's hash changes as the data in the block is changed, it is nearly impossible to temper data in a blockchain. NFT Connect is a single-page web application built on the Web 3.0 standards and made possible by MERN and Blockchain technologies.

NFT Connect, a digital marketplace for NFT transactions. Users of this platform can create, sell and buy the NFTs of their choice. Non-fungible tokens, or NFTs, have been the focus of discussion for a while. Users are further encouraged to think about the uses of NFTs because they have been made available at previously unheard-of rates. NFT Connect is currently helping artists get the respect and attention they have long deserved. NFT holders are assured ownership rights. A single digital item known as NFT is kept on the chain. NFT can be any type of digital material, including video, photo, gifs, and 3D models. It makes use of blockchain technology, which offers a safe, decentralized database for record-keeping.

REVISION CHART

Version	Suggestions	Description of Version	Date Completed
<i>Capstone - 1</i>	Meta verse Technology	At the time of proposal presentation, it was suggested to use the meta verse technology in the project. Then later on it was advised by the advisor not to use this technology in the project. In this report we have worked on the introduction, motivation and objectives clearing chapter#1. Then we covered chapter #2, where we have made the requirement analysis. Covering chapter#3, we did Domain analysis, and made use cases and prototypes of the proposed system. Then we cover chapter#4, creating DFDs of level 0, 1 and 2. Lastly we made the system diagrams covering system architecture, Sequence diagrams, collaboration diagram, activity diagram, ERD and data dictionary	27/1/2023
<i>Capstone - 2</i>	NFT Connect	The metaverse technology is not used as per advisor's consent. The test cases and the traceability matrix created. Accuracy, completeness and correctness was checked. The report was finalised.	6/6/2023

CONTENTS

CONTENTS

1. INTRODUCTION:	1
1.1 MOTIVATION:.....	1
1.2 PROJECT OVERVIEW:.....	1
1.3 PROBLEM STATEMENT:	2
1.4 OBJECTIVES:.....	2
2. DOMAIN ANALYSIS	3
2.1 CUSTOMERS:	3
2.2 STAKEHOLDERS:	3
2.3 DEPENDENCIES/EXTERNAL SYSTEM:	4
2.4 REFERENCE DOCUMENTS:	4
2.4.1 Related Projects.....	4
2.4.2 Critical Analysis:.....	5
2.4.3 Feature Comparison Table:	6
3. REQUIREMENTS ANALYSIS	7
3.1 REQUIREMENTS:.....	7
3.2 BUSINESS REQUIREMENT:.....	7
3.3 PRODUCT REQUIREMENT:.....	7
3.4 USER REQUIREMENT:.....	7
3.5 DESIGN CONSTRAINTS:	7
3.6 PROGRAMMING LANGUAGE:.....	7
3.7 INTERFACE REQUIREMENTS:.....	8
3.7.1 User Interface:.....	8
3.7.2 Hardware Interfaces:	8
3.8 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	9
3.8.1 Non-functional Requirements:.....	10
3.9 LIST OF ACTORS:	11
3.10 LIST OF USE CASES:	12
3.11 USE CASE DIAGRAM	13
3.12 EXTENDED USE CASES	14
3.13 USER INTERFACES	29
3.13.1 Dashboard	29
3.13.2 Create Collection.....	31
3.13.3 Create NFT	34
3.13.4 Sell NFT.....	35
3.13.5 Buy NFT.....	36
3.13.6 Help Center	38
4. DATA FLOW DIAGRAM.....	39
4.1 DFD LEVEL 0:.....	39
4.2 DFD LEVEL 1:.....	39
4.3 DFD LEVEL 2:.....	40
5. SYSTEM DESIGN	41
5.1 SYSTEM ARCHITECTURE DIAGRAM:.....	41
5.2 CLASS DIAGRAM:.....	42
5.3 SEQUENCE DIAGRAM:	43

5.3.1 Connect Wallet:	43
5.3.2 Create Collection:	44
5.3.3 Submit Query:	45
5.3.4 Create NFT:	46
5.3.5 Sell NFT:	47
5.3.6 Buy NFT:	48
5.4 COLLABORATION DIAGRAM	49
5.4.1 Create Collection:	49
5.4.2 Create NFT:	49
5.4.3 Sell NFT:	50
5.4.4 Buy NFT:	51
5.5 ACTIVITY DIAGRAM	52
5.5.1 Create Collection.	52
5.5.2 Create NFT	53
5.5.3 Sell NFT.	54
5.5.4 Buy NFT.	55
5.6 STATE DIAGRAM	56
5.6.1 Create Collection.	56
5.6.2 Buy NFT.	56
5.6.3 Create NFT	57
5.6.4 Sell NFT.	58
5.7 ENTITY RELATIONSHIP DIAGRAM	59
5.8 DATA DICTIONARY	60
5.8.1 User Dictionary:	60
5.8.2 Collection Dictionary	60
5.8.3 NFT Dictionary	61
5.8.4 Sell Dictionary	62
5.8.5 Buy Dictionary	62
6. IMPLEMENTATION DETAILS:	63
6.1 DEVELOPMENT SETUP:	63
6.1.1 ReactJS:	63
6.1.2 NodeJS:	63
6.1.3 MongoDB:	63
6.1.4 ExpressJS:	63
6.1.5 Solidity:	63
6.1.6 Visual Studio Code:	64
6.2 DEPLOYMENT SETUP:	64
6.2.1 ReactJS:	64
6.2.2 NodeJS:	64
6.2.3 Blockchain:	64
6.3 ALGORITHMS:	64
6.4 CONSTRAINTS:	65
6.4.1 Assumptions:	65
6.4.2 System Constraints:	66
6.4.3 Restrictions:	67
6.4.4 Limitations:	68
7. TESTING:	69

7.1 EXTENDED TEST CASES:	69
7.1.1 Connect Crypto Wallet	69
7.1.2 Create Collection.....	71
7.1.3 Create NFT:.....	72
7.1.4 Sell NFT:	74
7.1.5 Buy NFT:.....	76
7.1.6 Edit NFT:.....	80
7.1.7 Delete NFT	81
7.1.8 Transfer NFT:.....	82
7.2 DECISION TABLE:	83
7.2.1 Code Snippets:.....	83
7.2.2 Decision Table:	87
7.3 TRACEABILITY MATRIX:	88
7.3.1 Requirements vs Use Cases:.....	88
7.3.2 Requirements VS Test Cases:	89
7.3.3 Use Cases vs Test Cases:	90
7.3.4 Requirements vs Prototypes:	91
8. RESULTS.....	92
8.1 COMPLETENESS:.....	92
8.2 ACCURACY:.....	93
8.3 CORRECTNESS:.....	93
9. CONCLUSION	95
10. FUTURE WORK.....	96
10.1 FRACTIONAL OWNERS:.....	96
IT IS OBSERVED THAT ONLY ONE USER CAN OWN AN NFT. WORK IS BEING DONE TO INTRODUCE FRACTIONAL OWNERS. IT WILL ALLOW MORE THAN ONE USERS TO OWN SINGLE NFT. THIS WILL MAKE IT FEASIBLE FOR ALMOST EVERYONE TO OWN HIGH-VALUE ASSETS.	96
10.2 IMPROVEMENT IN SUSTAINABILITY:	96
10.3 AN IMPROVED USER EXPERIENCE:	96
10.4 NFTs AND AI:.....	96
10.5 NFTs IN METAVERSE:.....	96
11. BIBLIOGRAPHY.....	97
11.1 ARTICLES:	97
12. APPENDIX:	98
12.1 GLOSSARY:	98
12.2 PRE-REQUISITES:.....	99
12.2.1 NodeJS:.....	99
12.2.2 VS-Code:.....	99
12.2.3 External Crypto Wallet:	99

DEFINITIONS AND ACRONYMS

Acronym	Definition
UMT	University of Management and Technology
NFT	Non-Fungible Token
FCFS	First Come First Serve
DFD	Data Flow Diagram
ASAP	As Soon As Possible
ERD	Entity Relationship Diagram
CSS	Cascade style sheet
JS	JavaScript
ERC	Ethereum Request for Comments
IDE	Integrated Development Environment

LIST OF FIGURES

Figure 3.1 Use Case Diagram	13
Figure 3.2 Dashboard Screen.....	29
Figure 3.3 Explore screen	29
Figure 3.4 Stats screen	30
Figure 3.5 Resource screen.....	30
Figure 3.6 Profile screen	30
Figure 3.7 My collection Screen.....	31
Figure 3.8 create collection screen.....	32
Figure 3.9Create Collection screen.....	32
Figure 3.10 create Collection screen.....	33
Figure 3.11 Create collection	33
Figure 3.12 Create NFT screen.....	34
Figure 3.13 Create NFT screen.....	34
Figure 3.14 Create NFT screen.....	35
Figure 3.15 Sell NFT at FCFS	36
Figure 3.16 Sell NFT at auction.....	36
Figure 3.17 Buy NFT screen.....	37
Figure 3.18 Add to cart screen.....	37
Figure 3.19 Submit Query screen	38
Figure 4.1 DFD Level 0	39
Figure 4.2 DFD Level 1	39
Figure 4.3 DFD Level 2	40
Figure 5.1 System Architecture Diagram	41
Figure 5.2 Class Diagram	42
Figure 5.3 Connect wallet sequence diagram	43
Figure 5.4 Create Collection Sequence Diagram.....	44
Figure 5.5 Submit Query sequence diagram.....	45
Figure 5.6 Create NFT sequence Diagram	46
Figure 5.7 Sell NFT Sequence diagram.....	47
Figure 5.8 Buy NFT Sequence diagram	48

Figure 5.9 Create collection collaboration diagram.....	49
Figure 5.10 Create NFT collaboration diagram.....	49
Figure 5.11 Sell NFT Collaboration diagram	50
Figure 5.12 Buy NFT Collaboration Diagram.....	51
Figure 5.13 Activity Diagram Create Collection.....	52
Figure 5.14 Activity diagram Create NFT	53
Figure 5.15 Activity diagram Sell NFT	54
Figure 5.16 Activity Diagram Buy NFT	55
Figure 5.17 State Diagram Create Collection.....	56
Figure 5.18 State Diagram buy NFT.....	56
Figure 5.19 State Diagram Create NFT	57
Figure 5.20 State Diagram Sell NFT	58
Figure 5.21 ERD	59
Figure 7.1 Loading.....	83
Figure 7.2 Dashboard.....	83
Figure 7.3 Create Collection	84
Figure 7.4 Create Collection	84
Figure 7.5 My collection before collection created	85
Figure 7.6 My Collection after collection created	85
Figure 7.7 Create NFT	86

LIST OF TABLES

Table 2.1 List of Stakeholders	3
Table 2.2 Feature Comparison Table.....	6
Table 3.1 Functional Requirements	9
Table 3.2 Non Functional Requirements	10
Table 3.3 List of Actors	11
Table 3.4 List of Use Cases	12
Table 3.5 Extended Use Case Connect Crypto Wallet	14
Table 3.6 Extended Use Uase to Create Collection	15
Table 3.7 Extended Use Case to Create NFT	16
Table 3.8 Extended Use Case to Sell NFT	17
Table 3.9 Extended Use Case to Sell NFT at Auction.....	18
Table 3.10 Extended Use Case to Sell NFT at FCFS Approach	19
Table 3.11 Extended Use Case to Buy NFT	20
Table 3.12 Extended Use Case to Edit NFT	21
Table 3.13 Extended Use Case to Delete NFT	22
Table 3.14 Extended Use Case to Transfer NFT	23
Table 3.15 Extended Use Case to View Profile.....	24
Table 3.16 Extended Use Case to add NFT to cart.....	25
Table 3.17 Extended Use Case to use Help Center	26
Table 3.18 Extended Use Case to View About us	27
Table 3.19 Extended Use Case to View NFT in full Screen Mode	28
Table 3.20 View NFT in full screen extended use case.....	28
Table 5.1 User Data Dictionary	60
Table 5.2 Collection Data Dictionary	60
Table 5.3 NFT Data Dictionary	61
Table 5.4 Sell Data Dictionary.....	62
Table 5.5 Buy Data Dictionary	62
Table 7.1 Test Case to Connect Valid Wallet Account	69
Table 7.2 Test Case to Connect Invalid Wallet Account.....	70
Table 7.3 Test case to create Collection	71

Table 7.4 Test Case to Create NFT with Collection.....	72
Table 7.5 Test Case to Create NFT without Collection.....	73
Table 7.6 Test Case to Sell NFT at Fixed Price.....	74
Table 7.7Test Case to Sell NFT at Auction	75
Table 7.8 Test Case to Buy NFT at Fixed Rate with Sufficient Funds	76
Table 7.9 Test case to Buy NFT at Fixed Rate Without Sufficient Funds	77
Table 7.10 Test Case to Buy NFT at Auction With Sufficient Funds	78
Table 7.11 Test Case to Buy NFT with Insufficient Funds	79
Table 7.12 Test Case to Edit NFT	80
Table 7.13 Test Case to Delete NFT.....	81
Table 7.14 Test Case To Transfer NFT	82
Table 7.15 Decision Table	87
Table 7.16 Requirements vs Use Cases Traceability Matrix	88
Table 7.17 Requirements vs Test Cases Traceability Matrix	89
Table 7.18 Use Cases vs Test Cases Traceability Matrix	90
Table 7.19 Requirements vs Prototypes Traceability Matrix	91

1. Introduction:

Blockchain technology has brought revolution to world of technology. It serves as an immutable ledger of events where data is replicated over the blockchain network and each block is cryptographically linked to the next block. Tempering the data in blockchain is almost impossible as the hash of the block changes with the change in the data. This technology has introduced the concept of web 3.0. NFT Connect is web 3.0 based single page web application developed using the MERN technologies and Blockchain technology.

NFT Connect is a digital marketplace to deal in NFTs. This platform allows its users to sell and buy NFTs of their choice. NFTs or non-fungible tokens, have long been the subject of debate. NFTs have been sold at unheard-of prices, which give users additional motivation to consider their applications. Currently, NFTs are assisting artists in receiving the value and attention they have long deserved. Since NFTs are created on a blockchain, the NFT holder is guaranteed ownership rights. NFT is a unique digital item stored on the chain. NFT can be any digital asset like videos, images, gifs and 3D models etc. It uses blockchain technology which provides secure decentralized database to keep track of the record.

1.1 Motivation:

As we all know that we live in the world of technology and today we need to implement web 3.0 in almost every management system we need to keep data in soft form instead of keeping it in hard form because for any organization their records matters a lot and for any person his/her documents worth a lot. Any kind of hard document or paper work can get lost, it's very easy for someone to burn it. So, with the help of web 3.0 we can keep our important records and documents on a secure data base so that no un authorized person can have his hands on our important documents. Web 3.0 also gives you ownership of any digital asset of yours so that no one can destroy it or misuse it.

1.2 Project Overview:

NFT is a unique digital item which cannot be substituted and is stored on the blockchain network. The things you can do with tangible stuff can also be done with NFT. You can sell it, present it as a gift to your friend and delete it (just like you throw tangible stuff) if you don't want it. NFT Connect provides you marketplace to trade in NFTs. The marketplace is build keeping in note the Ethereum protocol standards ERC-721. ERC-721 Ethereum standard enables our application to work with non-fungible tokens.

This platform provides its users with the ownership of their digital assets. Using this platform, users can convert any of their assets into NFT (digital record of ownership). It is a unique digital item stored on a blockchain. NFT can be anything like images, audio, video and

3D models. This project uses Blockchain technology. Blockchain is a secure decentralized network where making changes to the data is not easy. It ensures the integrity of data by consensus method. NFT connects provides a large range of categories to ease buyers and creators. The creator can set category for their asset so the buyer willing to purchase the asset in specific category can easily find their artwork. The creator can also set their NFT to auction, in this method different buyers can bid the auction and the creator can choose to whom he wants to sale his artwork to.

A single block in a blockchain contains data, unique hash of the block and the hash of the previous block. If anyone changes the data then the hash for that block changes resulting in invalid data and all the blocks in the chain after that block will also become invalid. Each block on the blockchain is cryptographically linked to the other block. Integrity of the data is also ensured cryptographically. The art piece is made non-fungible by allotting a non-fungible token. There is also a track record of the owners (previous and current) of the digital asset which is accessible by any user. If a user buys an NFT then the authorization will be transferred to the buyer and he/she will own the asset officially. If you are an NFT creator and are unsure of the best venue for showcasing your digital product, welcome to NFT Connect.

1.3 Problem Statement:

The Problem faced with digital assets was that they were fungible. Anyone could use those assets without authorization. Anyone could copy the digital artwork of any creator and there was no way to find out which piece is original. This platform has provided ease to the creators and buyers. Another matter faced now a day is that when anyone buys an art piece of a collector, its authenticity comes with a paper certificate which you have to take care of all the time. It can be easily forgotten or destroyed.

1.4 Objectives:

- User friendly and eye catchy interface to provide excellent experience to the user.
- Create, buy and sell NFTs.
- A secure and immutable platform introduced to ensure the integrity of data.

2. Domain Analysis

NFT connect provides a large range of categories to ease buyers and creators. The creator can set category for their asset so the buyer willing to purchase the asset in specific category can easily find their artwork. The creator can also set their NFT on auction, in this method, different buyers can bid the auction and the creator can choose to whom he wants to sale his NFT.

2.1 Customers:

Creators, artists, celebrities and brands are the main target audience of the project. Otherwise, if a person is willing to deal in NFTs can use this website as it allows its user to collect, create, sale, transfer and buy NFTs with user-friendly Interface.

2.2 Stakeholders:

Following is the list of stakeholders of NFT Connect:

- Clients/users
- Admin
- Minors
- Developer

Table 2.1 List of Stakeholders

Stakeholder	Role	Description
Client/User	Creator	The users or clients can create NFTs.
Client/User	Seller	Users can sell their created NFTs by either of the ways and earn Crypto currency.
Client/User	Buyer	Users can buy NFTs they are interested in.
Admin	Administrator	Admin will manage the user's queries and make sure system's maintenance.
Miners	Contract deployment	Miners will deploy the smart contracts and charge gas fee for it.
Developer	Developer	Maintains the System and add new features if required.

Stakeholders are the people that are economically or socially effected by the project. Table 2.1 shows a list of stakeholders of <NFT Connect>. Client/user are effected socially and socially as

they are known due to their artwork/NFT and socially effected as they earn by selling their NFTs.

2.3 Dependencies/External System:

- ***External Crypto Wallet*** –Our proposed application works via an external wallet. The user creates an account on any wallet, which can store NFT and crypto currency and then connect it to our platform to continue their journey at NFT Connect.
- ***Internet Connection*** –The participants willing to use the platform must have a strong and reliable internet connection.

2.4 Reference Documents:

2.4.1 Related Projects

2.4.1.1 Magic Eden:

Magic Eden (Sidney Zhang, 2021) was founded by a group of four friends Sidney Zhang, Jack Lu, Zuoqun Yin and Zhouji Zhou which were learning about NFTs and its enormous potential. With expertise in crypto, consumer internet companies, and management consulting, they found gaps in the market for existing NFTs and came up with a brand-new solution.

2.4.1.2 Nifty Gateway:

A worldwide-founded platform developed in 2018. The twin brothers, Duncan and Griffin Cock Foster developed Nifty Gateway (Foster, Nifty Gateway, 2018), a digital art online auction platform for non-fungible token art. Nifty Gateway has sold NFTs by well-known NFT artists as Beeple, Pak, Refik Anadol, and others. A small number of platform users were compromised in March 2021. The fact that none of the people who were compromised had 2-factor authentication turned on.

2.4.1.3 Binance NFT:

The Binance NFT (Binance NFT, 2021) Marketplace was designed for trading a variety of digital collectibles and artworks. It is cost-free to join up for, use, and has modest marketplace and mint costs for exchanging NFTs. However, because Binance NFT only gives a 1% royalty fee, producers may not make that much money from secondary sales. The platform might be fantastic for collectors, but it's not so wonderful for artists who want to make money off of their works.

2.4.1.4 Rarible:

Rarible (Falin, 2022) is a blockchain-secured NFT marketplace and issuance platform that enables users to produce and trade digital collectibles. Alex Salnikov and Alexei Falin launched Rarible in 2020. Making NFT available to everyone is one of the goals of Rarible. In the long run, Rarible wants to be a community-owned, decentralised, autonomous organisation

(DAO). Six months after the project's beginning, in July 2020, the team unveiled the platform's RARI governance token. For all transactions, Rarible levies a 1% fee on the buyer as well as the seller.

2.4.1.5 Async Art:

Async Art (Conlan Rios, 2020) is a platform for creating, buying, and selling programmable NFT artwork. Async Art is creating a brand-new type of digital artwork that is created as a collection of live, changeable "Layers." The ownership of such Layers can be retained by the artists, sold to collectors, or programmed to change automatically according on various factors, such as the time of day or shifting stock market prices.

2.4.1.6 Makersplace:

Makersplace (Makersplace, 2018) is marketplace for nonfungible token artists and collectors. Users can buy, sell, and produce digital art to be sold using blockchain technology, just like other NFT markets. MakersPlace was established in 2016. Since MakersPlace is non-custodial, using it is very safe. Only the buyer and seller have access to cryptocurrencies and other digital assets thanks to smart contracts. Simply serving as a service to facilitate transactions, MakersPlace.

2.4.2 Critical Analysis:

After a detailed analysis of some well-known NFT marketplaces i.e, OpenSea, Magic Eden, Nifty Gateway, Rarible, Coinbase NFT, Binance NFT, SuperRare, Async Art, Makers Place, KnownOrigin, Mintable, Foundation, BakerySwap and Zora etc. We concluded that OpenSea is the current best NFT marketplace with almost every major feature and requirement, it isn't easy for anyone to find any flaws on their platform. However, after doing a detailed study of OpenSea we found some flaws in their system. One of the major flaw is that their interface is not very user friendly. For example, if a beginner visits their website it will be quite hard for them to navigate their way through buying and selling NFTs. Moreover, they don't have a wide range of categories and things are not organized very well. Furthermore, hardly any NFT marketplace provides a NFT social media platform. Therefore, our major goal is to create a NFT marketplace which provides a user friendly interface so that every user, whether its a beginner or an expert NFT collector. We aim to provide our users with a marketplace with a large range of categories which are organized and easy to navigate. To conclude it, we aim to make a NFT marketplace that resembles OpenSea but does not have the flaws that OpenSea has.

2.4.3 Feature Comparison Table:

Table 2.2 Feature Comparison Table

MarketPlace		Magic Eden	Nifty Gateway	Binance NFT	Rarible	Async Art	Marker's Place
Sale Style		Auction, Fixed price	Auction, Fixed Price	Auction, Fixed Price, Offer	Auction, Fixed Price	Auction, Fixed Price	Auction
Gas fee	Initial Contract	No	Yes	No	NO	Yes	Yes
	Minting	No	NO	Yes	NO	No	NO
	Listing	No	Yes	Yes	NO	Yes	Yes
	Auction Settlement	Yes	Yes	Yes	Yes	Yes	Yes
Supported Format	Img	Yes	Yes	Yes	Yes	Yes	Yes
	Video	Yes	No	Yes	Yes	No	No
	Audio	Yes	No	Yes	Yes	Yes	No
	SVG	No	Yes	No	No	No	Yes
	3D	Yes	No	No	No	Yes	No
Blockchain		Polygon	Ethereum	Ethereum	Ethereum, polygon	Ethereum	Ethereum

Table 2.2 Shows a comparison between the features of those marketplaces. Table 2.2 gives details about what are the charges scheme, the supported format of NFTs, the selling types and the blockchain used by the respective marketplaces.

3. Requirements analysis

3.1 Requirements:

The main goal that we hope to achieve is the saving of time. When all the work will be done on one place instead of going place to place.

3.2 Business requirement:

- Provide everything at one place.
- Make it easy to use and time saving.

3.3 Product requirement:

- Sales and market tracking of NFTs
- NFT Collections
- NFT Social Media
- User Profile Management

3.4 User requirement:

- Any system with web browser having internet connection and a NFT Wallet.
- End user: Anyone can use it provided it is more than 18 years old.
- Performance requirements: Time to load the page, time to load data, minimum loading time. Secure and Decentralized Network.

3.5 Design constraints:

- User will land on the Dashboard on accessing NFTConnect.
- User will signup/login using unique/valid credentials.
- User can Buy/ Sell NFTs and also explore different collections.

3.6 Programming language:

The following languages will be used in our project

- ReactJS
- CSS3
- HTML5
- JQuery
- NodeJS
- JavaScript
- Solidity
- Ethereum/Solana
- MongoDB

3.7 Interface requirements:

3.7.1 User Interface:

Application will be accessed through a Browser Interface. User would be able to explore NFTs but won't be able to trade in NFTs without logging in and connecting a NFT Wallet to their account.

3.7.2 Hardware Interfaces:

The minimum hardware requirements are as follows

3.7.2.1 Server Side:

- Operating System: Windows XP/7/8/10, Windows ME
- Processor: Pentium 3.0 GHz or higher
- RAM: 2 GB or more
- Hard Drive: minimum 10 GB or more

3.7.2.2 Client side:

- Operating System: Windows 9x or above, MAC or UNIX.
- Processor: Pentium III or 2.0 GHz or higher.
- RAM: 2GB or more

3.7.2.3 Software Interfaces:

- User Interface: HTML5, CSS3, ReactJs, jQuery, JavaScript
- ServerSide: NodeJs, ExpressJs, JavaScript
- Database: MongoDB
- Blockchain: Solidity, Ethereum

3.8 Functional and Non-Functional Requirements

Table 3.1 Functional Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
FR1	Sign Up	Evident	Registering into the system	The users should be able to register using their unique and valid email ids and a strong password. The credentials of each user should be saved in MongoDB so the user can be authenticated while logging in.
FR2	Logging In	Evident	System Access	Users should be able to log in to their account using valid credentials, so that they can access all the features of NFTConnect. The credentials of the logged in user should be stored in MongoDB so that we can track the active sessions on the system.
FR3	Wallet Authentication	Evident	Third party access	The address of user's wallet should be linked with user's account so that users don't have to connect their wallet every time.
FR4	Create Collections	Evident	Current user	Users should be able to create their own collections in which they can store the NFTs which they have created or bought. The collections should be saved in user's profile so he can access it whenever he wants.
FR5	Create NFTs	Evident	Blockchain access	Users should be able to register their digital assets as NFTs in the Blockchain network. Newly created NFTs should be minted in the blockchain network as a block.
FR6	Sell NFTs	Evident	market trading	Users should be able to buy the NFTs listed for sale or sell the NFTs they own. After buying or selling NFTs, the ownership is transferred to the new owner.
FR7	Dashboard	Evident	Current user	users should be able to see different categories of NFTs on the dashboard and explore all the NFTs that are listed for sale or auction.

				NFTs should be extracted from the blockchain and displayed on the dashboard for the users to explore.
FR8	Buy NFT	Evident	Current user	Users should be allowed to buy the listed NFTs from the marketplace. They pay the required amount and then the NFT ownership is transferred to the buyer.
FR9	Payment processing from Wallet	Evident	Third party access	Users should be able to pay in ethereum from their NFT wallets while buying NFTs from NFTConnect. users should be provided a secure payment gateway.

Table 3.1 Provides us with the Functional requirements of <NFT Connect>. These requirements demonstrate the functionalities provided by <NFT Connect> according to the latest trends and requirements. Table 3.1 Comprehends the usage of <NFT Connect>.

3.8.1 Non-functional Requirements:

Table 3.2 Non Functional Requirements

RID	Description	Category	Attribute	Details & Boundary Constraints
NFR1	Authentication	Hidden	System access	users should only able to access the system using valid credentials.
NFR2	Security	Hidden	Is Admin	A secure system and two step verification provide security for any account.
NFR2	Upgradable	Hidden	According to market requirements	It should be simple to upgrade the system in accordance with new requirements.

NFR3	Scalability	Hidden	Quality maintenance	The system shouldn't freeze or hang.
NFR4	Data Integrity	Hidden	Reliability of data	The system ought to be linked to a database to preserve this integrity.
NFR5	Performance:	Hidden	Reliability of data	System processing must be instantaneous and without latency. It takes a stable internet connection to use this system.

Table 3.2 comprehends the non-functional requirements provided by <NFT Connect>. The purpose of these requirements is to ensure a reliable and up-to-date system to the users. Table 3.2 Comprehends the usage of <NFT Connect>.

3.9 List Of Actors:

Table 3.3 List of Actors

Actors	Description
User/Client	User connect their wallet to the NFT connect to the marketplace and can create, Buy, transfer and sell NFTs.
Admin	Admin checks the queries of the user or client and respond with a satisfactory solution.

Table 3.3 Contains the list of actors that will be using <NFT Connect>. User/ Client can deal in NFT in the marketplace. They can convert any of their digital asset into The user can ask about any difficulty they face using <NFT Connect> which will be responded by the admin.

3.10 List of use cases:

Table 3.4 List of Use Cases

ID	Primary Actors	Use case
1	Client/User	Connect Crypto Wallet
2	Client/User	Create Collection
3	Client/User	Create NFT
4	Client/User	Sell NFT(FCFS, Auction)
5	Client/User	Auction
6	Client/User	FCFS
7	Client/User	Buy NFT
8	Client/User	Bidding
9	Client/User	FCFS
10	Client/User	Edit NFT
11	Client/User	Delete NFT
12	Client/User	Transfer NFT
13	Client/User	View Profile
14	Client/User	Add to cart
15	Client/User	View Help Center
16	Client/User	View NFT in full screen Mode
17	Admin	Respond to Queries

Table 3.4 contains List of use cases. These use cases show how the user will be interacting with the platform. The user can use <NFT Connect> to create NFT, to sell NFT and to buy NFT. This platform provide two ways to sell NFTs i.e. at auction or at fixed price. The user can also transfer the NFT to anyone they want.

3.11 Use Case diagram

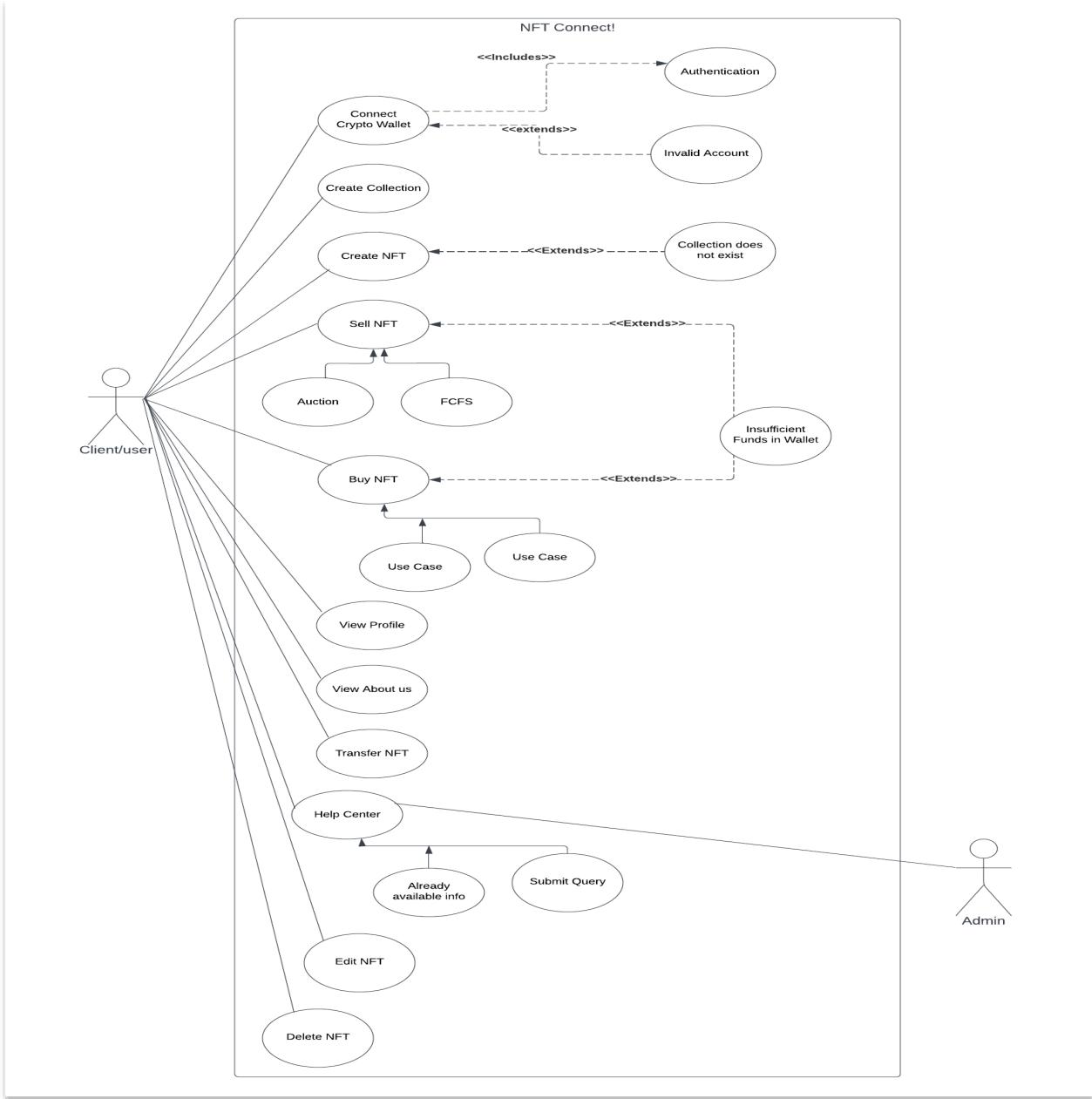


Figure 3.1 Use Case Diagram

Figure 3.1 Shows Use Case diagram of <NFT Connect>. This diagram shows the behavior of the marketplace. The figure 3.1 shows that in what ways the user can interact with the system and in what ways the admin can interact with the system.

3.12 Extended Use Cases

Table 3.5 Extended Use Case Connect Crypto Wallet

Use Case ID	NFT_UC_001	
Name	Connect Crypto Wallet	
Actors	Client	
Purpose	Allows the user to use the marketplace to create, buy and sell NFTs.	
Description	A program designed to buy, create and store the NFTs and crypto currency. Users who want to use the marketplace need to connect their external wallet to the marketplace.	
Pre-Conditions	A stable internet connection required.	
Successful Post-Conditions	User can create, sell and buy NFTs.	
Failure Post-Conditions	User cannot deal in NFTs.	
Typical Course of Events		
Actor Action		System Response
1	The user will connect their external wallet to the marketplace.	Wallet connected.
2	The user can now create, sell and buy NFTs if have sufficient balance in the wallet.	...
Alternative Course		
Step 1	The user not able to connect their wallet can only visit the marketplace and cannot deal in NFTs. So if one is interested in dealing in NFTs at the marketplace, they should try connecting their crypto wallet.	

Table 3.5 explains the procedure to connect the crypto wallet to the marketplace. A user cannot deal in NFTs until they connect their Crypto wallet to <NFT Connect>. <NFT Connect> Support a crypto wallet which can store the cryptocurrency and the NFT.

Table 3.6 Extended Use Case to Create Collection

Use Case ID	NFT_UC_002				
Name	Create Collection				
Actors	Client				
Purpose	Collections define category of NFT clients/users create.				
Description	In order to create NFT, it is mandatory for a client to create collection as it defines the category of the NFT. A single collection can have one or more than one NFTs.				
Pre-Conditions	A stable internet connection and a wallet with sufficient balance connected to the marketplace is required.				
Successful Post-Conditions	User can create NFTs.				
Failure Post-Conditions	User cannot deal in NFTs.				
Typical Course of Events					
Actor Action		System Response			
1	This use case begins when a user connect their wallet to the marketplace.	Wallet connected.			
2	The user now selects the collections tab from the dashboard.	User is lead to collections page.			
3	The user fills in the required information and clicks the create collection button.	Collection created.			
4	Now the user can use their collection to create NFTs.	...			
Alternative Course					
Step 1	Try Connecting a valid crypto wallet account.				

Table 3.6 explain the procedure to create collection. At <NFT Connect> it is mandatory for a user to create a collection in order to create NFT because collection provides category to NFT. Creation of collection is free as no gas fee is charged for it.

Table 3.7 Extended Use Case to Create NFT

Use Case ID	NFT_UC_003	
Name	Create NFT	
Actors	Client	
Purpose	Clients/users or users can create NFTs of their digital assets.	
Description	Using this feature, clients/users can create NFTs of their digital asset. An NFT can be anything like image, video, 3D models or gifs etc.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and a collection is required.	
Successful Post-Conditions	User can create NFTs and can do anything with it.	
Failure Post-Conditions	User cannot deal in NFTs.	
Typical Course of Events		
Actor Action		System Response
1	This use case starts by connecting user's crypto wallet to the marketplace.	Wallet connected.
2	The user then selects the create NFT feature from the navbar at the dashboard.	User lands at create NFT page.
3	The user fills in the required information, selects the corresponding collection and then creates the NFT.	NFT created.
4	The user then decides if he wants to keep the NFT or sell it.	...
Alternative Course		
Step 1	Try Connecting a valid crypto wallet account. Else the user should create collection or check the internet connection.	

Table 3.7 contains the procedure to create an NFT at <NFT Connect>. In order to create NFT the user must connect their wallet to the marketplace. Creating NFT requires collection. So if a user have not created collection, they should first create collection and then come to create NFT.

Table 3.8 Extended Use Case to Sell NFT

Use Case ID	NFT_UC_004	
Name	Sell NFT	
Actors	Client	
Purpose	Clients/users or users can sell their created NFTs.	
Description	Using this feature, clients/users can sell their NFTs. User can either put their NFT on selling by placing a fix price for it or by putting it at auction. In case of auction, the user can choose whom he wants to sell his NFT to depending upon the price offered for the respective NFT.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can sell and earn cryptocurrency.	
Failure Post-Conditions	User cannot deal in NFTs.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	The user select the NFT to sell at its profile page.	User provided with option to edit their NFT or to sell it.
3	The user then select sell option in order to sell their selected NFTs.	Lands to the page where user selects the selling method.
Alternative Course		
Step 1	Try Connecting a valid crypto wallet account.	

Table 3.8 provides steps to sell NFT at <NFT Connect>. A user can sell their created/ collected NFT. Selling an NFT costs some gas fee so a user must have sufficient funds in their account to sell NFT.

Table 3.9 Extended Use Case to Sell NFT at Auction

Use Case ID	NFT_UC_005	
Name	Sell at Auction	
Actors	Client	
Purpose	Clients or users can sell their created NFTs.	
Description	Using this feature, clients can sell their NFTs at auction. The client fills in the required fields and then put it on auction. The buyers willing to purchase the NFT bid on it. The client can then sell their NFT to whom they want to sell it depending upon the bidding amount.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can sell and earn cryptocurrency.	
Failure Post-Conditions	User cannot sell their NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	The user select the NFT to sell at its profile page.	User provided with option to edit their NFT or to sell it.
3	The user then select sell option in order to sell their selected NFTs.	Lands to the page where user selects the selling method.
4	The user chooses to sell their NFT at auction.	The user is lead to the UCTION page.
5	The user then fills in their wallet address in order to receive the selling amount of their NFT and then click the ok button.	Your NFT is placed at auction for selling.
Alternative Course		
Step 1	The user should have funds in the wallet for gas fee.	

Table 3.9 contains course of actions to sell an NFT at auction. Auction is one of the way to sell NFT at <NFT Connect>. The seller put their NFT on auction and multiple buyers place bid for that NFT. The seller can then sell their NFT to anyone from the buyers who placed bid.

<NFT Connect>

Table 3.10 Extended Use Case to Sell NFT at FCFS Approach

UseCase ID	NFT_UC_006	
Name	First come first serve approach	
Actors	Client	
Purpose	Clients or users can sell their created NFTs.	
Description	This feature allows the user to sell the NFT at the price they want to sell their NFT. The user fills in the required fields along with the selling price. The first person willing to buy the NFT gets the NFT.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can sell and earn cryptocurrency.	
Failure Post-Conditions	User cannot sell their NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	The user select the NFT to sell at its profile page.	User provided with option to edit their NFT or to sell it.
3	The user then select sell option in order to sell their selected NFTs.	Lands the user to the page where they select the selling method.
4	The user chooses to sell their NFT at auction.	The user is lead to the UCTION page.
5	The user then fills in their wallet address in order to receive the selling amount of their NFT and the price the user want to sell their NFT at and then select the sell	Your NFT is set for selling at the NFT Connect.
Alternative Course		
Step 1	Try Connecting a valid crypto wallet account.	

Table 3.10 provides with the course of actions required to sell NFT at FCFS approach. FCFS is another way of selling NFT at <NFT Connect>. It is selling the NFT at a fixed rate. In this approach the seller can sell the NFT at their desired price.

Table 3.11 Extended Use Case to Buy NFT

Use Case ID	NFT_UC_007	
Name	Buy NFT	
Actors	Client	
Purpose	Clients or users can buy NFTs.	
Description	This feature allows the user to buy NFTs they are willing to. You can select the category of the NFTs so that you can easily buy the NFT of your choice.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace is required.	
Successful Post-Conditions	User owns the bought NFT.	
Failure Post-Conditions	User cannot buy NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	The homepage contains a wide range for NFTs	...
3	User can select the category of NFT they are searching for at the dashboard.	Provides user with the category of NFTs they selected.
4	The user then selects the NFT they want to buy.	User lands at buying page.
5	The user fills in the required info. If the NFT is at auction then they can place a bid on that at the price they want to buy it. Otherwise, the user just need to fill in the amount the seller requires.	The ownership rights transferred to the buyer.
Alternative Course		
Step 1	Try Connecting a valid crypto wallet account with sufficient funds.	

Table 3.11 contains the method to buy NFT at <NFT Connect>. If the NFT is at auction then the user place a bid and wait for the seller's response. If the NFT is at FCFS then the user pay the amount to the seller and gets ownership of the NFT. Buying NFT charges gas fee.

<NFT Connect>

Table 3.12 Extended Use Case to Edit NFT

Use Case ID	NFT_UC_008	
Name	Edit NFT	
Actors	Client	
Purpose	Clients or users can edit their created NFTs.	
Description	This feature allows the users to edit their created NFTs. They can change the digital asset, the name, description and collection of their category.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can edit their NFT.	
Failure Post-Conditions	User cannot edit their NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	From the dashboard the user comes at profile page.	Lands at profile.
3	Then the user select their NFT to be deleted it.	...
4	Then select the edit button at the NFT
5	From there a pop up will appear which will contain the details of NFT.	...
6	Edit the details and click ok.	NFT edited.
Alternative Course		
Step 1	Add sufficient funds to wallet and Check internet Connection.	

Table 3.12 provides the steps to edit the NFT at <NFT Connect>. The user can edit their NFT and editing an NFT is a cost free process. The user can edit any information about their NFT. If the user has mistakenly done something they can easily edit the details.

Table 3.13 Extended Use Case to Delete NFT

Use Case ID	NFT_UC_009	
Name	Delete NFT	
Actors	Client	
Purpose	Clients or users can delete their NFTs.	
Description	This feature allows the users to delete their NFTs.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can delete their NFT.	
Failure Post-Conditions	User cannot delete their NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	From the dashboard the user comes at profile page.	Lands at profile.
3	Then the user select their NFT to be deleted it.	...
4	Then select the edit button at the NFT	...
5	From there a pop up will appear which will contain the details of NFT and a button to delete the NFT.	...
6	Select the Delete button and the NFT deleted.	NFT deleted.
Alternative Course		
Step 1	Check Internet connection.	

In order to delete an NFT the user can take help from Table 3.13 as it provides course of action to delete an NFT at <NFT Connect>. Deleting an NFT at <NFT Connect> is a cost free process as it does not require the user to pay gas fee.

Table 3.14 Extended Use Case to Transfer NFT

Use Case ID	NFT_UC_010	
Name	Transfer NFT	
Actors	Client	
Purpose	Clients or users can transfer their NFTs.	
Description	This feature allows the users to transfer their NFTs to whomever they want to transfer.	
Pre-Conditions	A stable internet connection, a wallet with enough balance connected to the marketplace and an NFT is required.	
Successful Post-Conditions	User can transfer their NFT.	
Failure Post-Conditions	User cannot transfer their NFT.	
Typical Course of Events		
Actor Action		System Response
1	The user connects their crypto wallet to the marketplace.	Wallet connected.
2	The user select the NFT to be transferred.
3	The user then transfer the NFT by entering the wallet address to whom the NFT is being transferred.	NFT transferred.
Alternative Course		
Step 1	The user should check their internet connection.	

Table 3.14 provides the strategies to transfer NFT to anyone at <NFT Connect>. Transferring NFT requires a user to pay gas fee. The user just needs to enter the account of the user to which they want to transfer their NFT.

Table 3.15 Extended Use Case to View Profile

Use Case ID	NFT_UC_011	
Name	Profile	
Actors	Client/users	
Purpose	By visiting their profile, users can see their activity at the marketplace.	
Description	This feature allows users to see the collection of NFTs created and the NFT purchased until the date. It also contains the range of NFTs favorite by the users.	
Pre-Conditions	A stable internet connection is required.	
Successful Post-Conditions	User can explore their gallery.	
Failure Post-Conditions	User cannot see their gallery of NFT items.	
Typical Course of Events		
Actor Action		System Response
1	The user selects the profile tab from the dashboard.	The user lands at the profile page.
2	The user can see the range of collections and NFT created.	...
3	User can see the field of NFT bought.	...
4	User can also see their activity at the marketplace.	...
Alternative Course		
Step 1	The user cannot see their profile if their wallet is not connected.	

If the user wants to view their profile at <NFT Connect>, they can follow the procedure provided in Table 3.15. At the profile the user can see their collected or created NFTs, their created collections and their activity at the marketplace.

Table 3.16 Extended Use Case to add NFT to cart

Use Case ID	NFT_UC_012
Name	Add to cart
Actors	Admin, Client
Purpose	All the clients/users can add their selected NFTS to the cart.
Description	This feature enables the users to add the NFT they want to buy in the cart.
Pre-Conditions	A stable internet connection and wallet connected to the marketplace is required.
Successful Post-Conditions	Every user can access the NFTs created at the marketplace.
Failure Post-Conditions	User having unstable internet connection cannot access it.
Typical Course of Events	
Actor Action	
1 Select the item the user is interested in buying.	...
2 Add the selected NFT to the cart.	NFT added to cart
Alternative Course	
Step 1	Unstable or No connection to the internet results in error.

The user can follow Table 3.16 in order to add an item to the cart. This use case begins when a user wants to buy an NFT at FCFS approach. The use adds the NFT to cart and then checks out to buy it.

Table 3.17 Extended Use Case to use Help Center

Use Case ID	NFT_UC_013	
Name	Help center	
Actors	Client	
Purpose	To provide our clients with the best of experience at the marketplace, this feature is launched.	
Description	This feature enables our users to get guidance for using marketplace. If the user still feels confused even after reading the guidance provided at this page, they can submit their queries in the form of text. Our team will resolve your query as soon as possible.	
Pre-Conditions	A stable internet connection is required.	
Successful Post-Conditions	User can ask for any difficulty they are facing in using the marketplace.	
Failure Post-Conditions	User is lead to the dashboard.	
Typical Course of Events		
Actor Action		System Response
1	From the dashboard, the user selects the help center feature.	User is lead to the help center page.
2	The user can read out the guidance provided to the queries at the hand.	...
3	If the user still feels uncertain about something or they don't get solution to their problem in the already provided guidance, they can submit their query by filling in the required fields along with their email address.	Our teams will response to your query at the provided email ASAP.
Alternative Course		
Step 1	Check internet Connection	

Table 3.17 provides the user with procedure to use Help center at <NFT Connect>. If a user is facing problem using the platform they can use Help center. Information is provided at this page to resolve the problems. If the user does not get the solution there, they can submit their query which will be responded by the admin.

Table 3.18 Extended Use Case to View About us

Use Case ID	NFT_UC_014	
Name	View About us	
Actors	Client	
Purpose	To provide our clients with the best of experience.	
Description	This feature allows user to know about NFT Connect.	
Pre-Conditions	A stable internet connection is required.	
Successful Post-Conditions	User understands the basic terms at the NFT Connect.	
Failure Post-Conditions	User is lead to the dashboard.	
Typical Course of Events		
Actor Action		System Response
1	From the dashboard, the user selects the About us handle.	About us page appears.
2	User can understand about how NFT Connect works and about its terms and conditions.	...
Alternative Course		
Step 1	Check internet connection.	

Table 3.18 contains course of action to know about us at <NFT Connect>. If the user wants to know about the basic terms and conditions at the marketplace they can visit about us page to know about the marketplace.

Table 3.19 Extended Use Case to View NFT in full Screen Mode

Use Case ID	NFT_UC_015	
Name	View NFT in full screen Mode	
Actors	Admin, Client	
Purpose	All the clients/users can view the NFT in full screen mode.	
Description	Allows the user to view the NFT in full screen so that they can clearly see the NFT.	
Pre-Conditions	A stable internet connection required.	
Successful Post-Conditions	Every user can access the NFTs created at the marketplace.	
Failure Post-Conditions	User having unstable internet connection cannot access it.	
Typical Course of Events		
Actor Action		System Response
1	The user selects any NFT (Either from dashboard or from their own collections).	NFT with feature description opens.
2	The user then clicks the full screen icon.	Full screen view of NFT.
Alternative Course		
Step 1	Unstable or No connection to the internet results in error. So try connecting your system to a stable internet connection.	

Table 3.20 View NFT in full screen extended use case

Table 3.19 Provides the course of action to view the NFT at full screen at <NFT Connect>. If a user wants to see the details of the NFT at the platform they can use this use case and can analyze the NFT at full screen view.

3.13 User Interfaces

3.13.1 Dashboard

Whenever a user comes at <NFT Connect>, the first page they land at is dashboard.

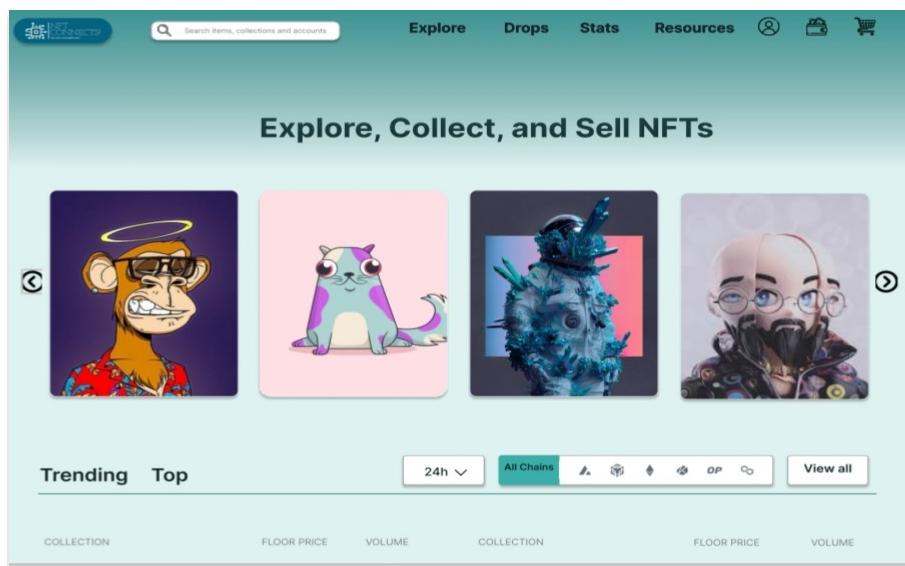


Figure 3.2 Dashboard Screen

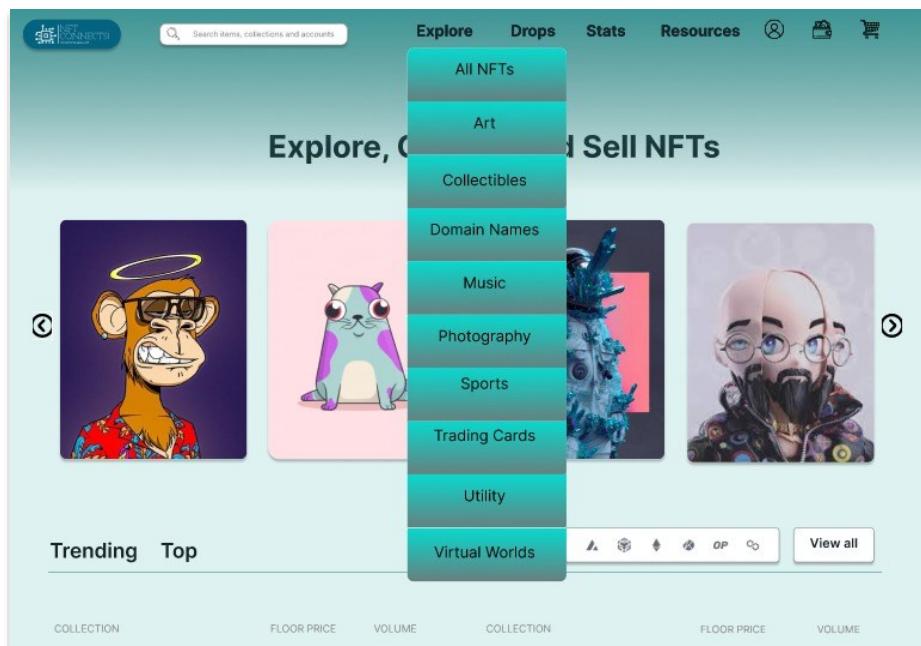


Figure 3.3 Explore screen

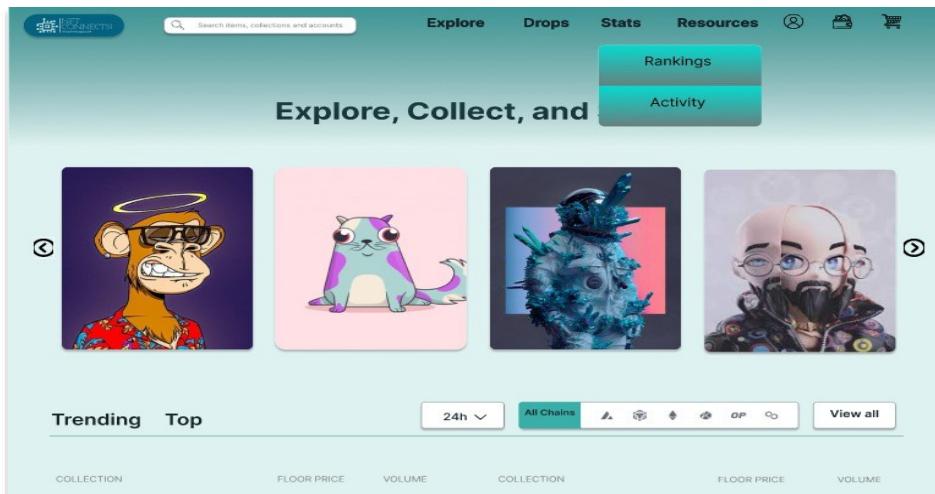


Figure 3.4 Stats screen

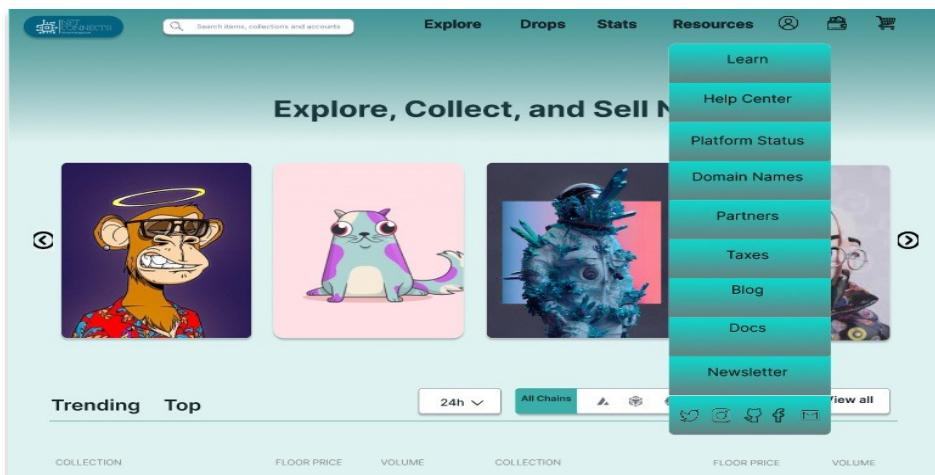


Figure 3.5 Resource screen

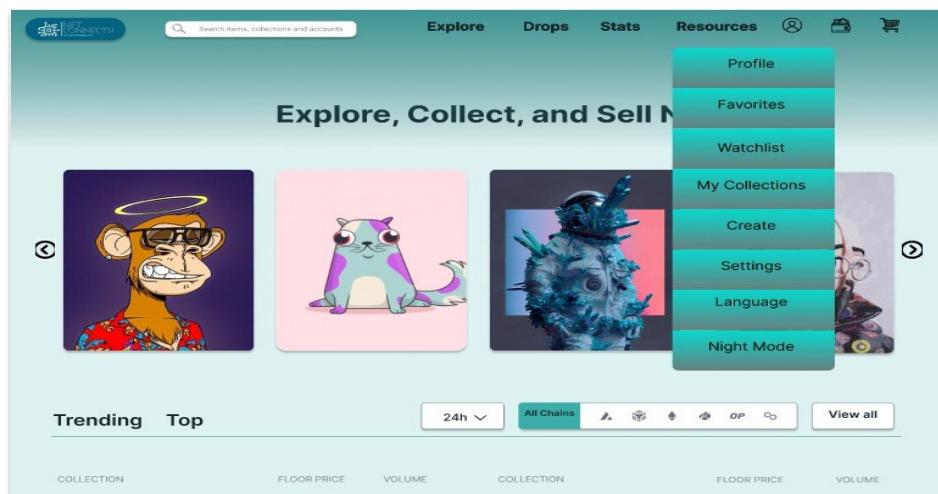


Figure 3.6 Profile screen

Figure 3.2 shows dashboard screen of <NFT Connect>. Dashboard contains a navbar consisting of Figure 3.3 explore tab where users can explore NFTs category wise. Drop tab shows the NFT calendar wise. Figure 3.4 is Stats tab shows the top ranking of the NFTs at the Marketplace. Figure 3.5 is Resource tab using which user can further go to about us page and help center. The profile tab Figure 3.6 which shows a drop down on hover containing the list of create collection, create NFT, favorite, watched and settings. The wallet tab to connect wallet to the marketplace. The dashboard screen contain NFTs listed for selling.

3.13.2 Create Collection

In order to create NFT the user needs to follow to following course of actions:

1. From the Dashboard, the user hovers at the resource tab where a dropdown appears.
2. Then go to My collection
3. At my collection page the user can see the collection if created otherwise a blank screen with a create button.
4. Go to create.
5. Provide the required data for creating collection.
6. Then click create button.
7. Your collection created.

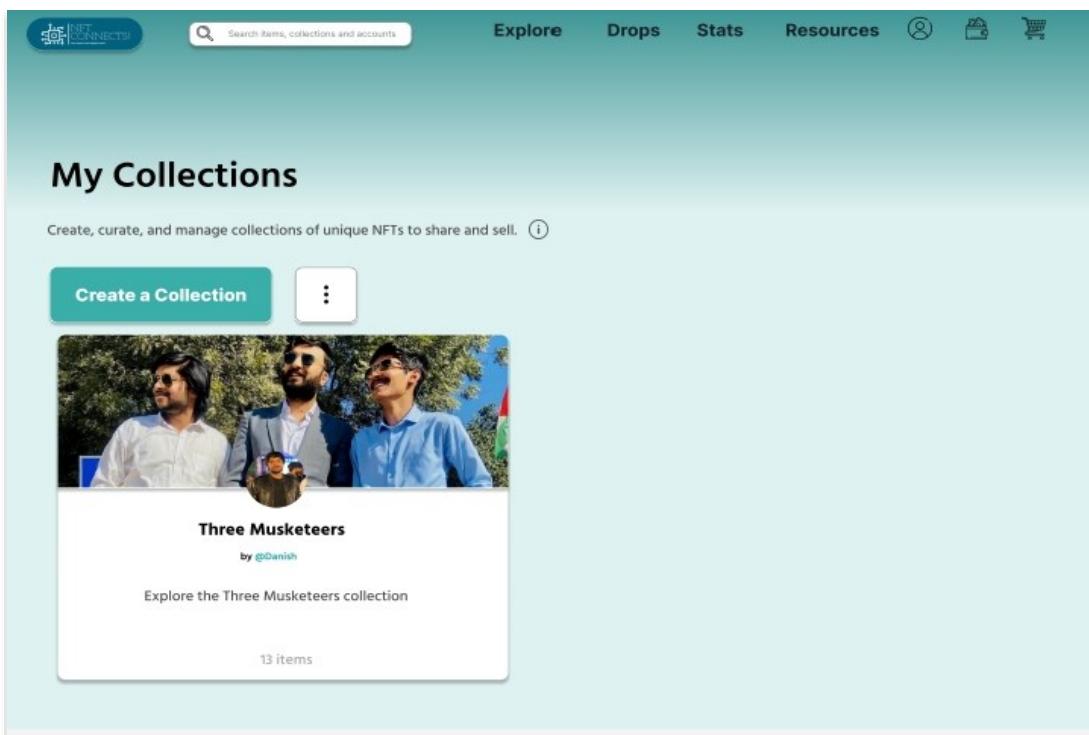


Figure 3.7 My collection Screen

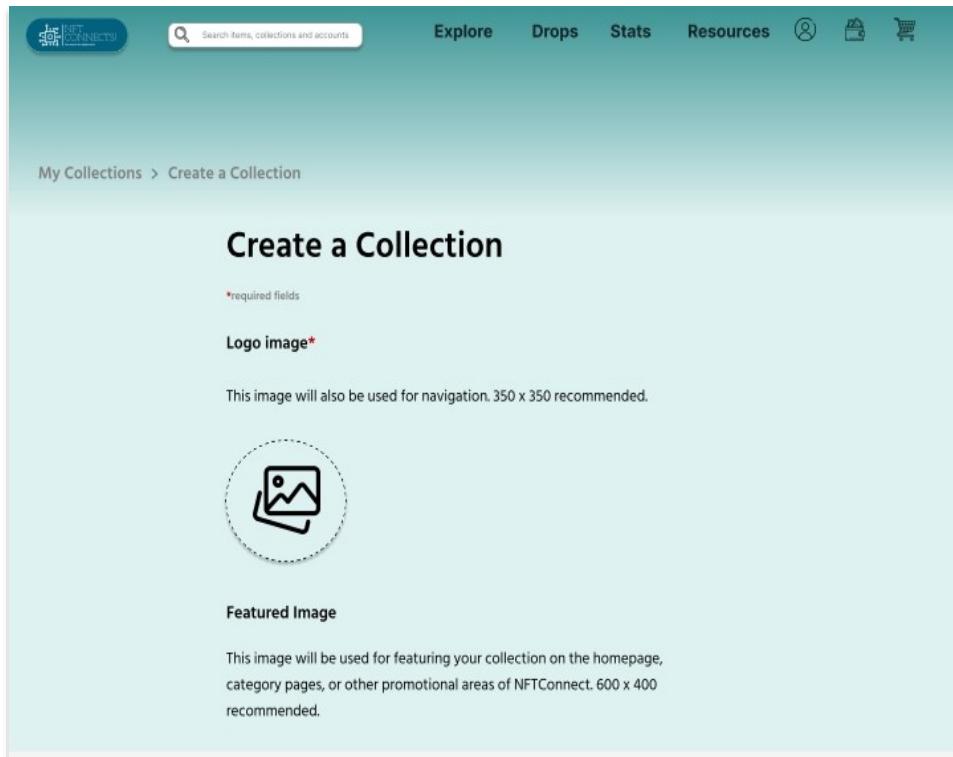


Figure 3.8 create collection screen

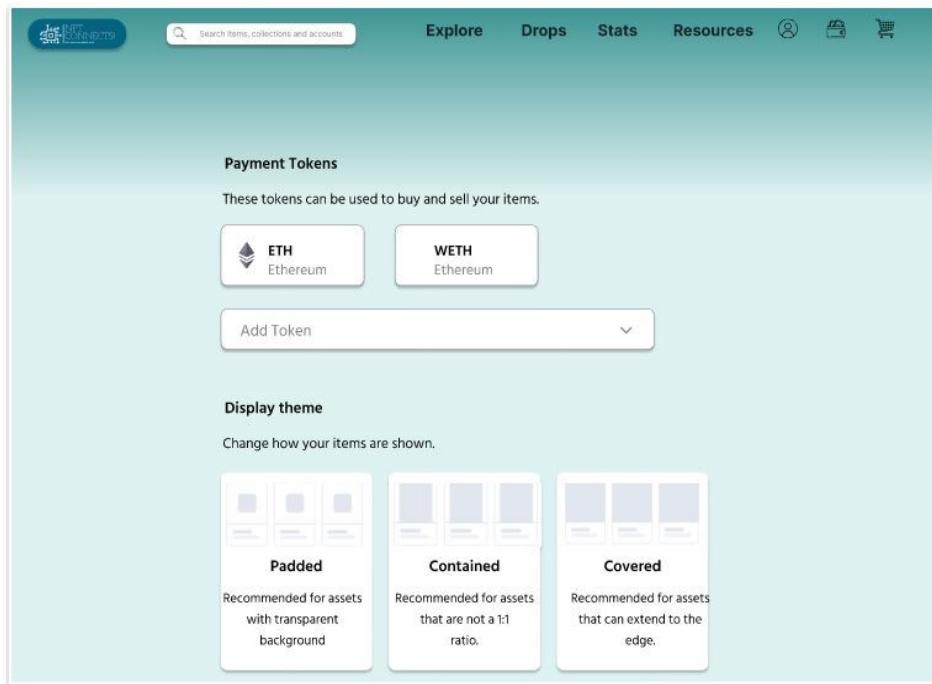


Figure 3.9Create Collection screen

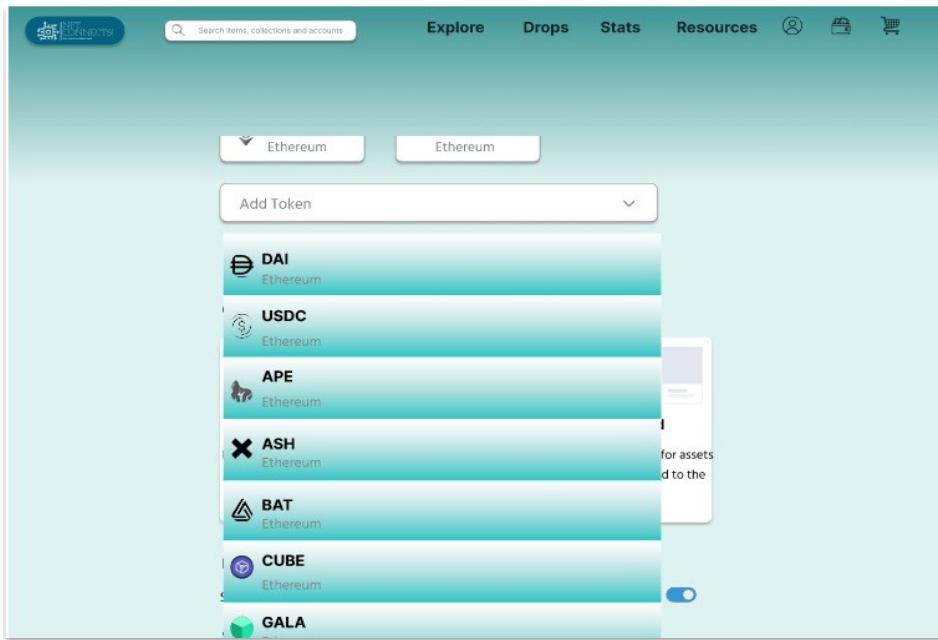


Figure 3.10 create Collection screen

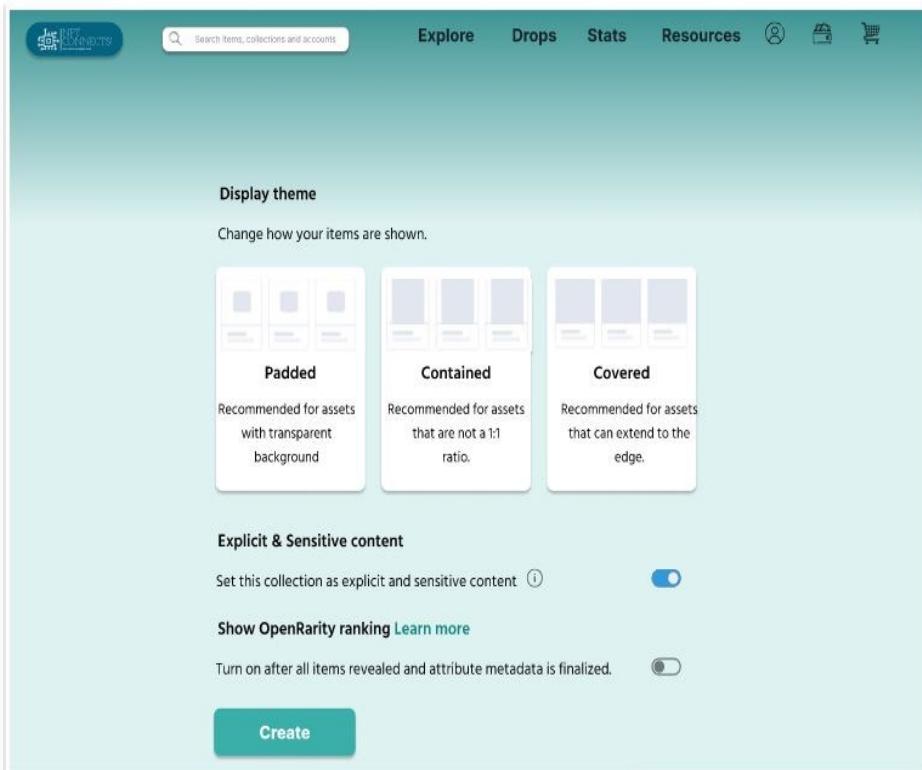


Figure 3.11 Create collection

Creating a collection requires the user to connect the wallet to the marketplace. The user can create collection only if their wallet is connected to the marketplace. If a user wants to create NFT, it is mandatory to create collection because a collection gives category to the NFT. Creating collection is a cost free process, it does not charge any gas fee.

3.13.3 Create NFT

User can create NFT at <NFT Connect> by following steps:

1. From the dashboard, hover at profile which show a drop down.
2. From the dropdown list select create
3. Provide the required data to create NFT
4. If the collection does not exist then the NFT cannot be created.
5. If the collection exist, click create
6. Your NFT created.

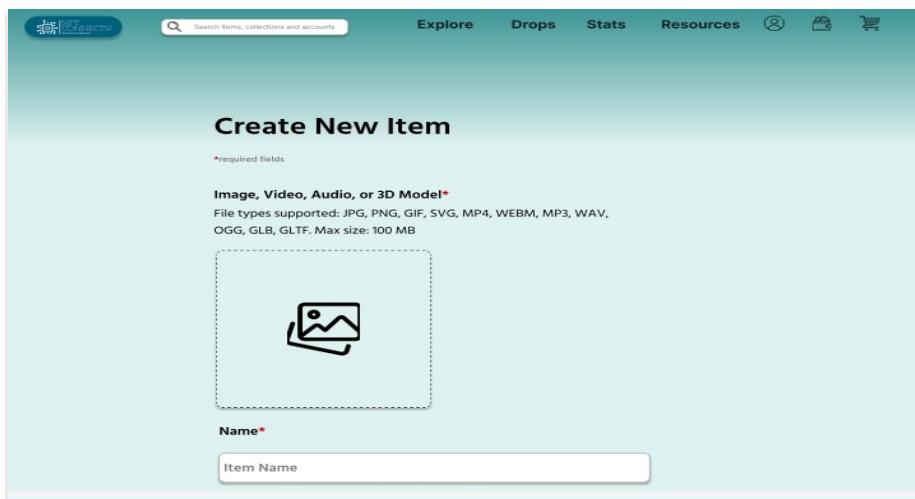


Figure 3.12 Create NFT screen

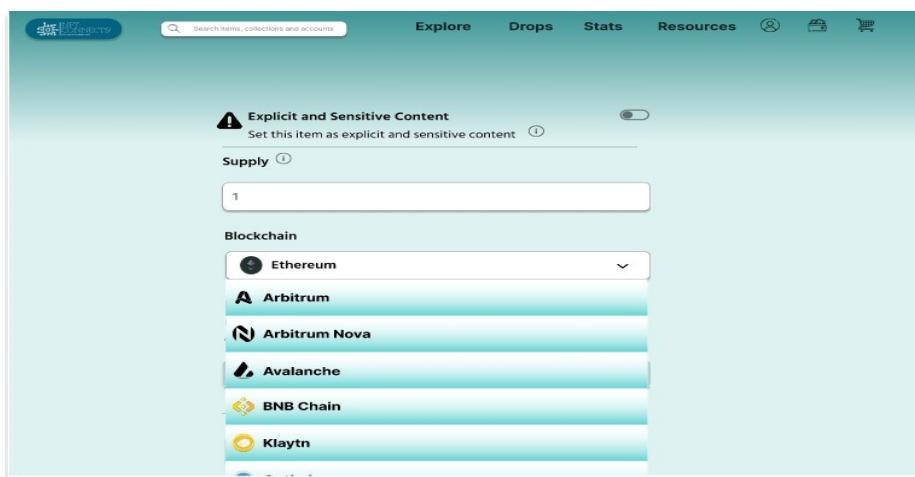


Figure 3.13 Create NFT screen

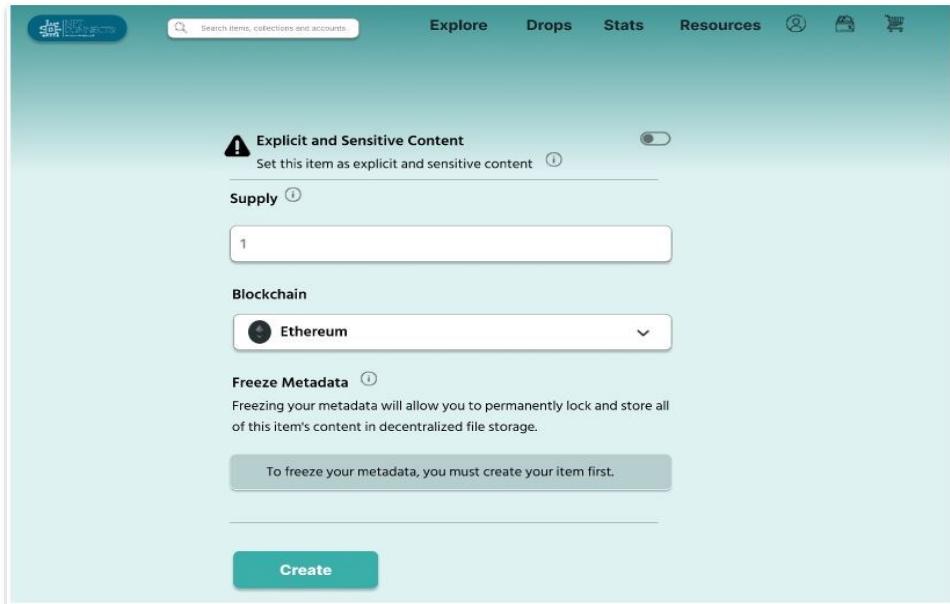


Figure 3.14 Create NFT screen

Creating an NFT is a cost free process as it does not require the user to pay gas fee. User can convert any of their digital asset into an NFT. An NFT can be an image, a 3D mode, a video, an audio or a GIF. NFT is a record of ownership of the NFT. Before creating NFT, make sure that user have created collection.

3.13.4 Sell NFT

Following are the course of action for selling NFT.

1. From the dashboard, hover at profile which show a drop down
2. From the dropdown list select profile
3. Go to collections
4. Select the NFT you want to sell
5. Select sell NFT button
6. Now enter the data according to the selected method
7. If selected auction, enter seller account number only
8. If selected FCFS, enter the price for NFT and seller account number.
9. Then select confirm
10. Your NFT listed for selling

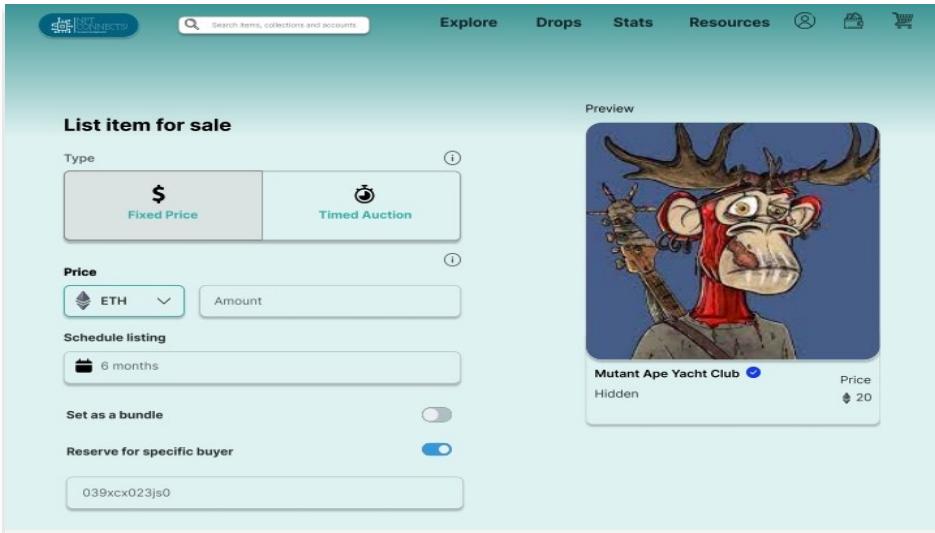


Figure 3.15 Sell NFT at FCFS

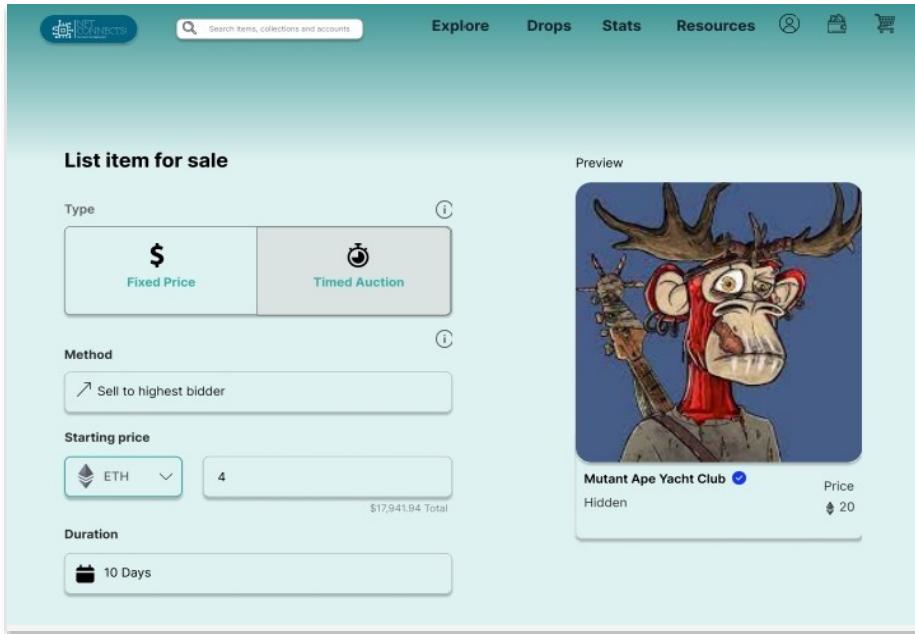


Figure 3.16 Sell NFT at auction

Listing an NFT for selling costs gas fee. At <NFT Connect>, user can sell their NFT in two ways. Figure 3.15 is selling the NFT at fixed price. The other way is selling at auction as shown in figure 3.16. At auction , the seller can sell to any one buyer from many buyers who placed bid at their NFT.

3.13.5 Buy NFT

Follow the following procedure to buy NFT

1. The dashboard contain the range of NFTs listed for selling
2. In order to get the item category wise select the category from the explore tab at dashboard.

3. User can also search for the NFT.
4. Select the NFT
5. If the NFT is at auction then place a bid and wait for the seller's response
6. If the NFT is at fixed price, add it to the cart
7. Complete the purchase
8. NFT ownership transferred to the buyer.

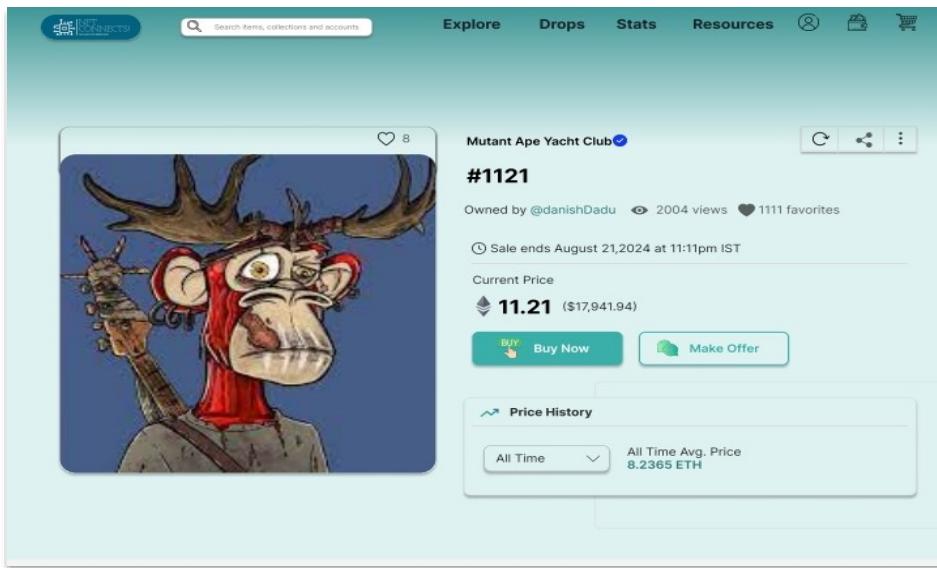


Figure 3.17 Buy NFT screen

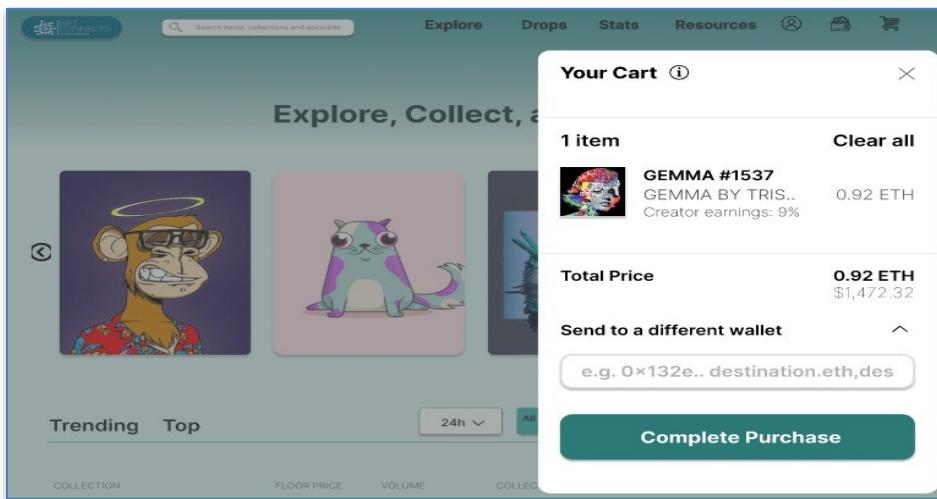


Figure 3.18 Add to cart screen

Figure 3.17 and 3.18 are the screens to buy NFT at <NFT Connect>. If the NFT is at auction then the user place a bid and wait for seller's response. Otherwise, the user can directly buy the NFT by paying the gas fee and the purchasing cost of NFT. Buy NFT costs gas fee.

3.13.6 Help Center

Following is the procedure to get to the help center page

1. From the dashboard, hover at the resource tab
2. From the dropdown, select learn
3. A page will open which will contain a range of solution to the commonly faced problems at the marketplace.
4. If the user does not find the solution to their problem, they can submit a query.
5. Select submit query from help center page
6. Select the type of query and provide the required information
7. Click submit button
8. Your query submitted

The screenshot shows the 'Help Center' section of the NFT Connect website. On the left, there is a sidebar with several options: 'My Account' (selected), 'Wallet and Transaction Issues', 'Buying and Selling NFTs', 'Developer Help', 'Report a Bug or Error Message', and 'Report Fraudulent Activity'. Below this is a link to 'Join our community on Twitter and Instagram'. On the right, the main form area has fields for 'Your Email Address *', 'Subject *', 'Description *', and an 'Attachments (optional)' section with a file upload input. A 'Submit' button is located at the bottom right of the form.

Figure 3.19 Submit Query screen

Help center is provided to ease the clients at the <NFT Connect>. If the user is facing any problem using the marketplace and need help then they can come at help center and get their issues resolved. If the solution to the problem is not available in the already provided solutions, they can submit a query as shown in Figure 3.19. Then their query will be resolved ASAP.

4. Data Flow Diagram

4.1 DFD Level 0:

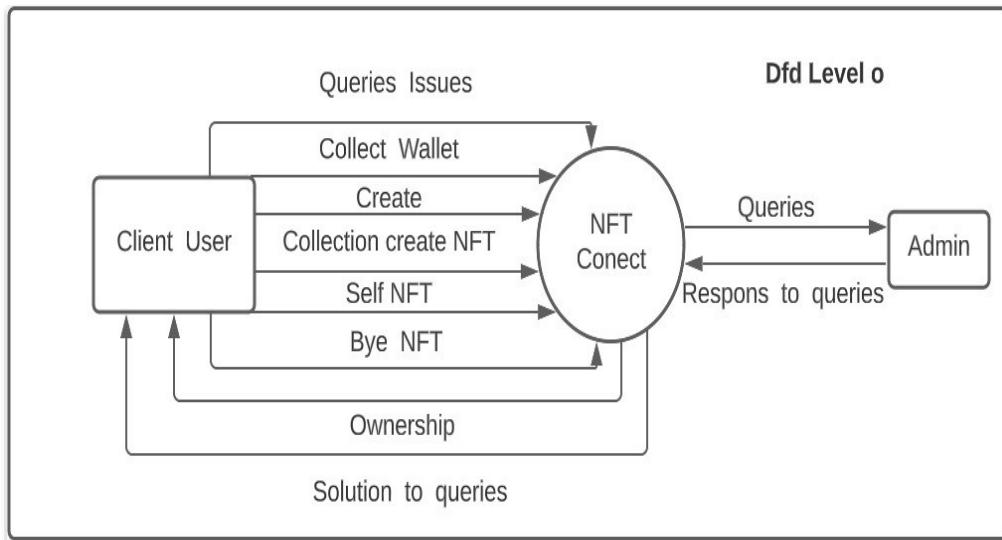


Figure 4.1 DFD Level 0

Figure 4.1 shows the data flow diagram level 0, which shows that with what sort of data the system deals. Figure 4.1 provides the very basic flow of data at <NFT Connect>.

4.2 DFD Level 1:

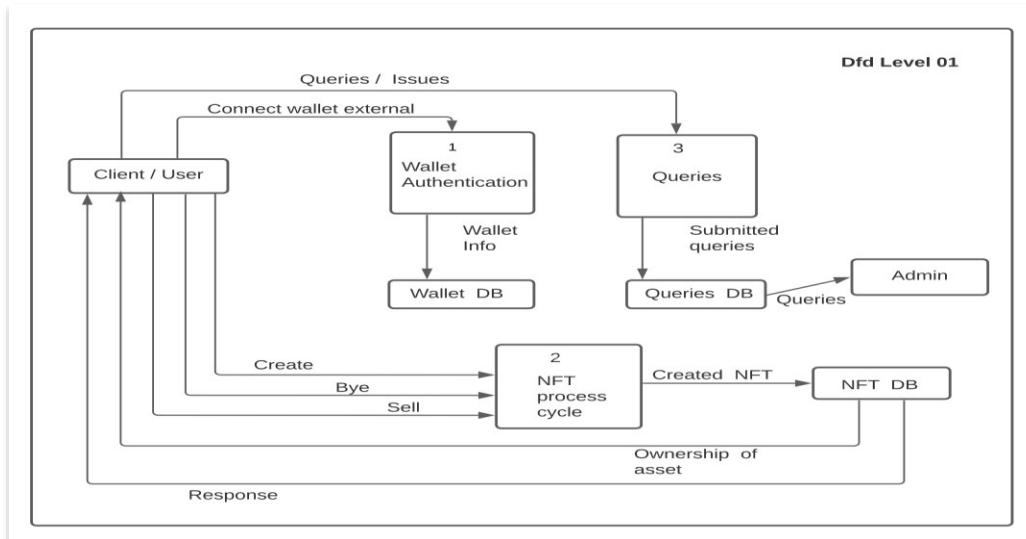


Figure 4.2 DFD Level 1

Figure 4.2 shows data flow diagram level 1. Figure 4.2 shows the flow of data at <NFT Connect>. It contains the numerous sub-processes the data flows through, data inputs and outputs, and data storage.

4.3 DFD Level 2:

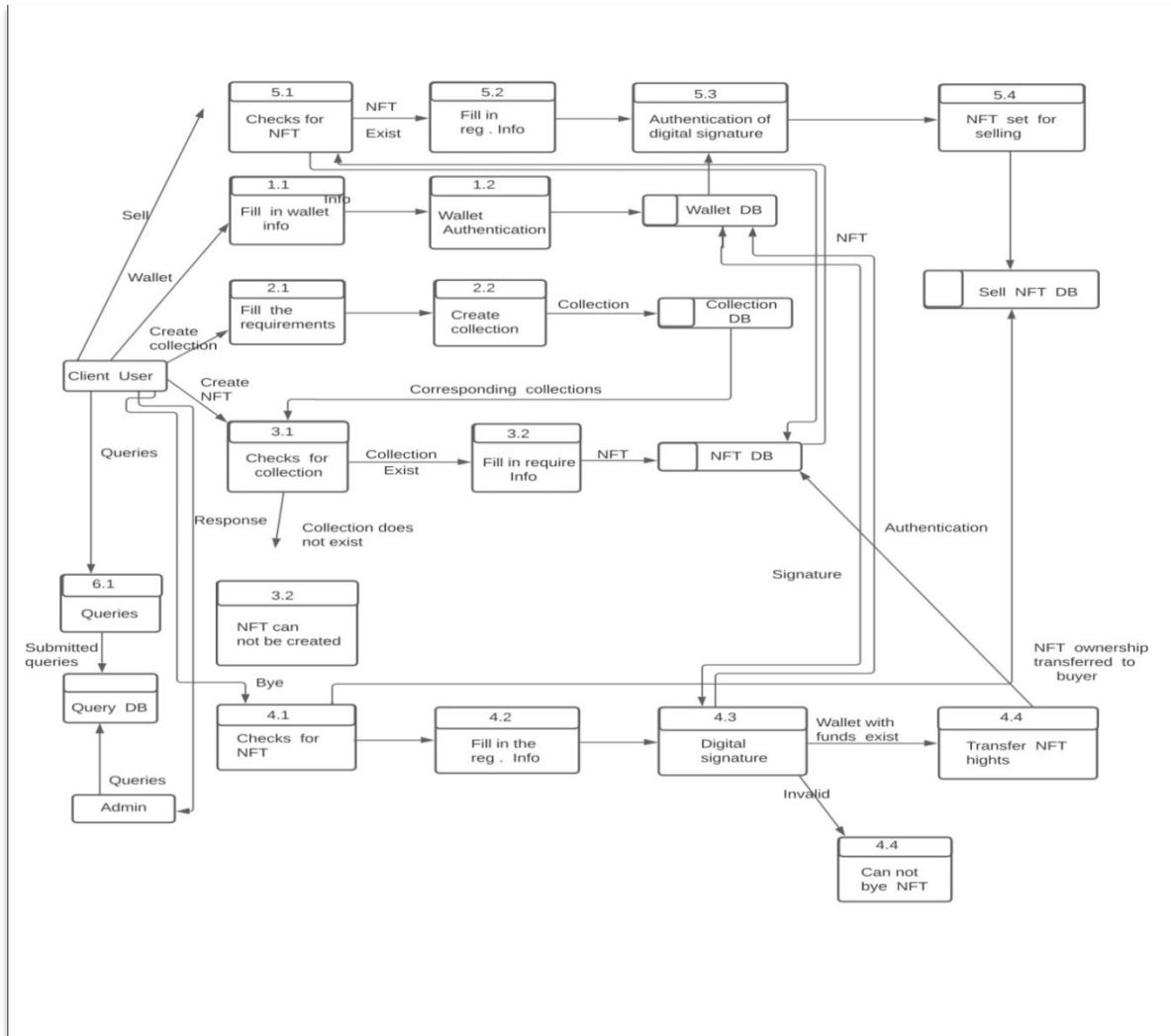


Figure 4.3 DFD Level 2

Figure 4.3 shows data flow diagram level 2. Figure 4.3 shows the flow of data at <NFT Connect>. This level two data flow diagram template illustrates the movement of information within a system and can be shared with various parties (stakeholders).

5. System Design

5.1 System Architecture Diagram:

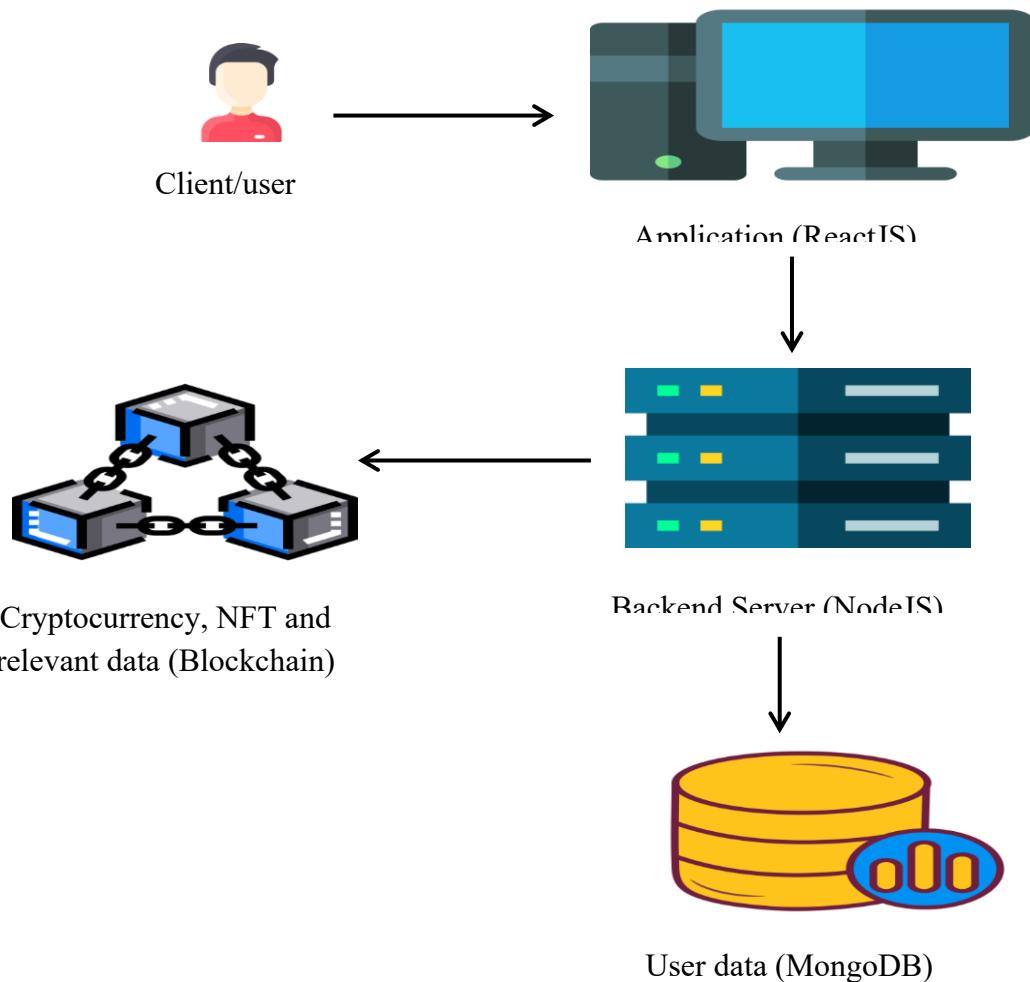


Figure 5.1 System Architecture Diagram

Figure 5.1 shows system architecture diagram that abstracts the connections, constraints, and divisions among software system components. It offers a thorough breakdown of the software system's physical deployment and development plan.

5.2 Class Diagram:

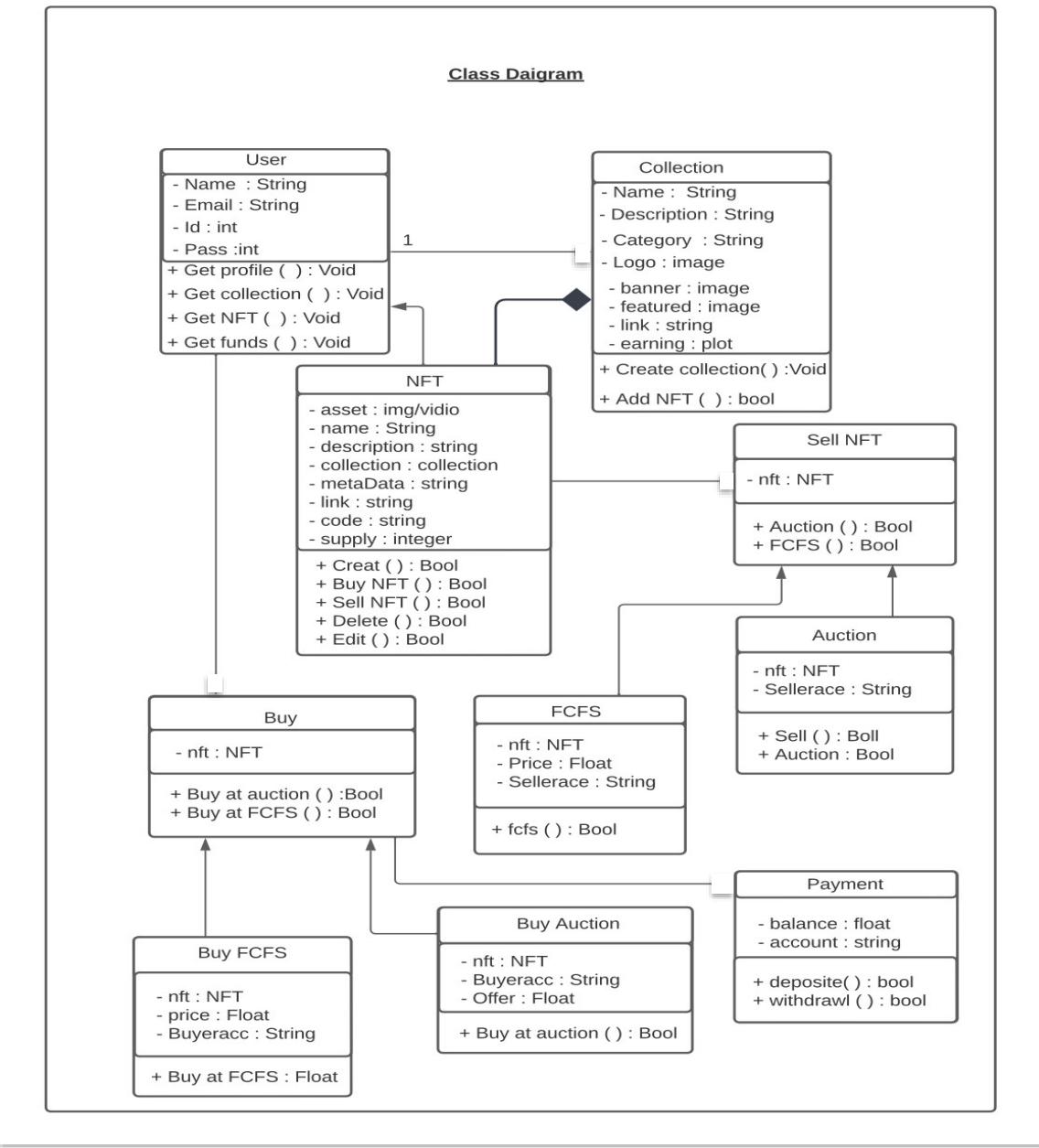


Figure 5.2 Class Diagram

Figure 5.2 shows the class diagram that is defining <NFT Connect>. It shows the functionalities and the relationship between objects. It shows the services provided by <NFT Connect>. It defines the code representing an entity.

5.3 Sequence Diagram:

5.3.1 Connect Wallet:

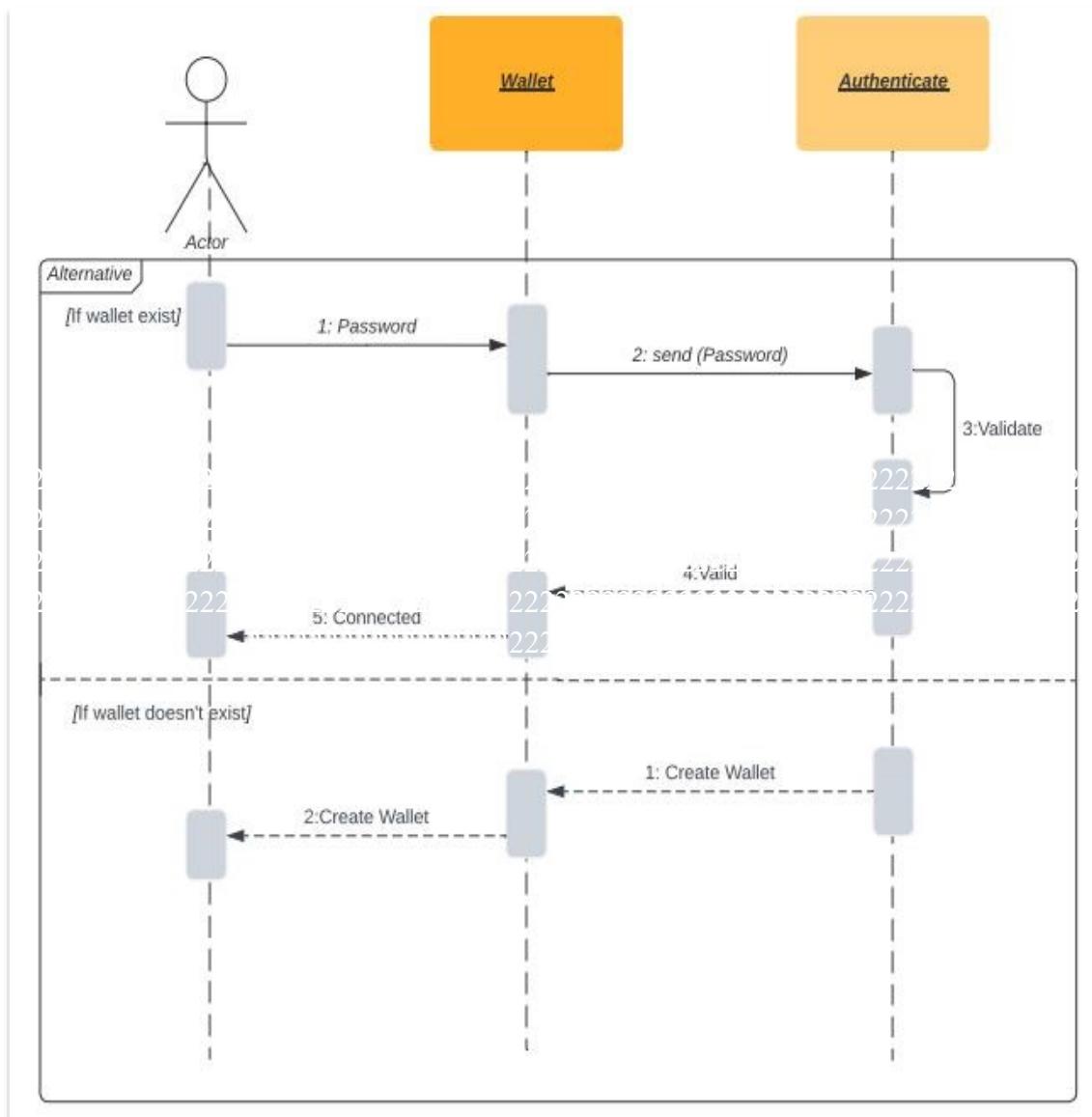


Figure 5.3 Connect wallet sequence diagram

Figure 5.3 shows sequence diagram. It contains the course of actions of the user to connect the wallet to <NFT Connect> and the response of the system to the user's actions. The user can connect their wallet only if they have a wallet account. Otherwise, a user is required to create wallet account and then connect it to the marketplace.

5.3.2 Create Collection:

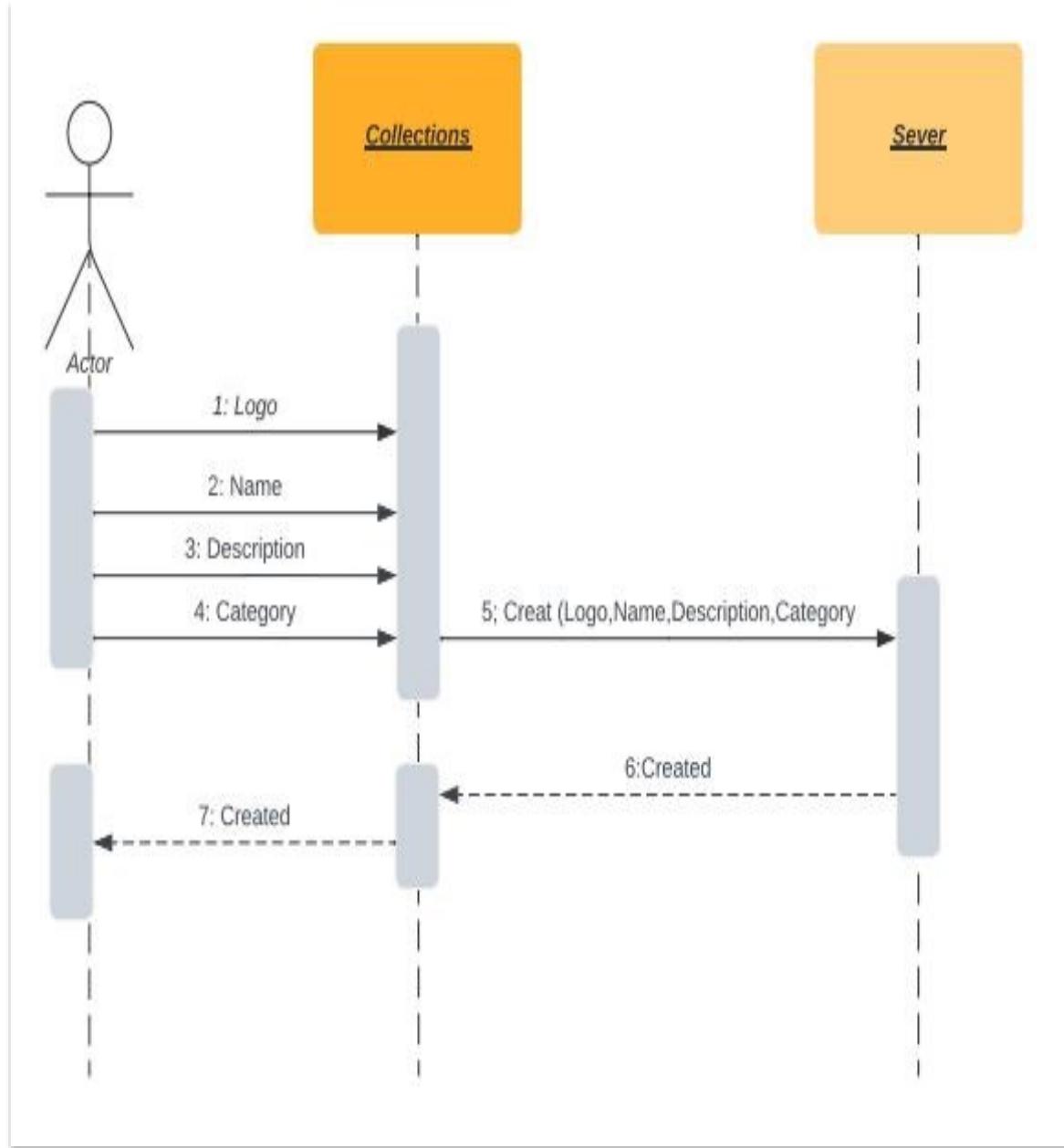


Figure 5.4 Create Collection Sequence Diagram

Figure 5.4 shows sequence diagram to create collection at <NFT Connect>. The user is required to connect their wallet to the marketplace in order to create collection. Creating a collection is mandatory in order to create NFT.

5.3.3 Submit Query:

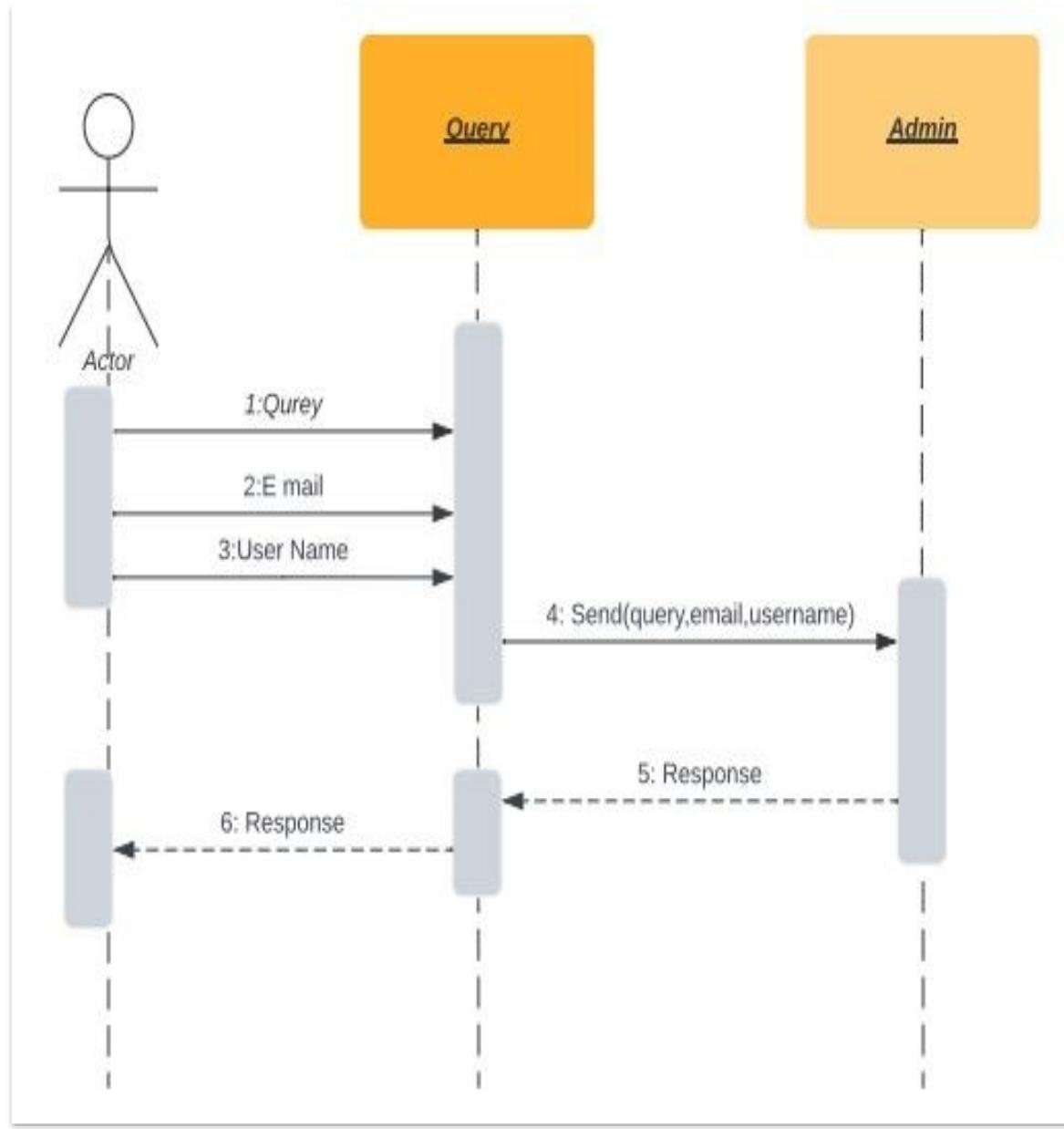


Figure 5.5 Submit Query sequence diagram

Figure 5.5 is a sequence diagram to submit a query in case of any problem at <NFT Connect>. If a user is facing problem in using and they don't get their solution at help center they can submit a query to which the admin will response with a suitable solution ASAP. Figure 5.5 demonstrates the sequence of steps to submit query.

5.3.4 Create NFT:

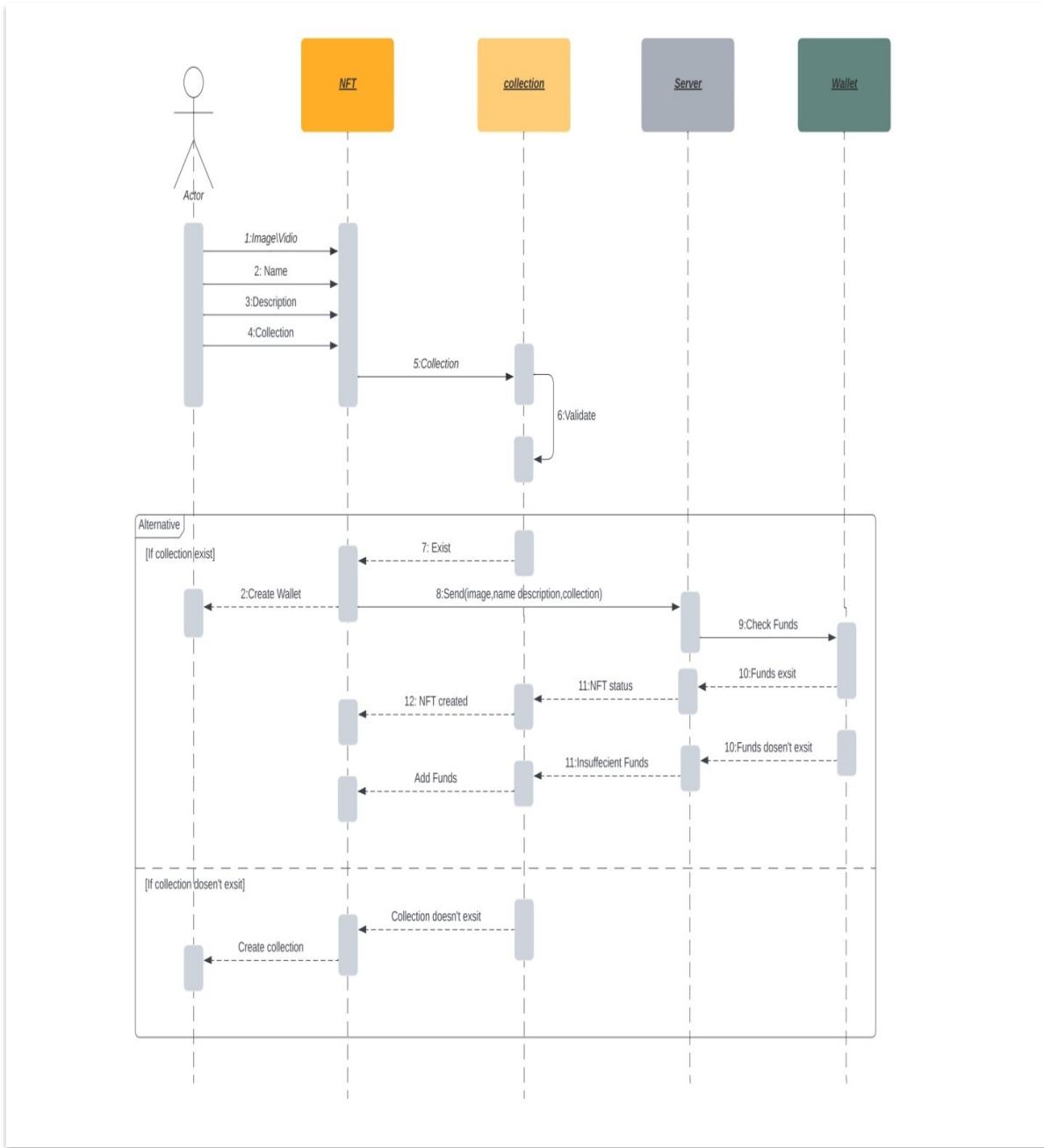


Figure 5.6 Create NFT sequence Diagram

Figure 5.6 is sequence diagram to create NFT at <NFT Connect>. It provides sequence of actions to a user perform to create NFT and the system's response to those course of actions. Creating an NFT is a cost free process.

5.3.5 Sell NFT:

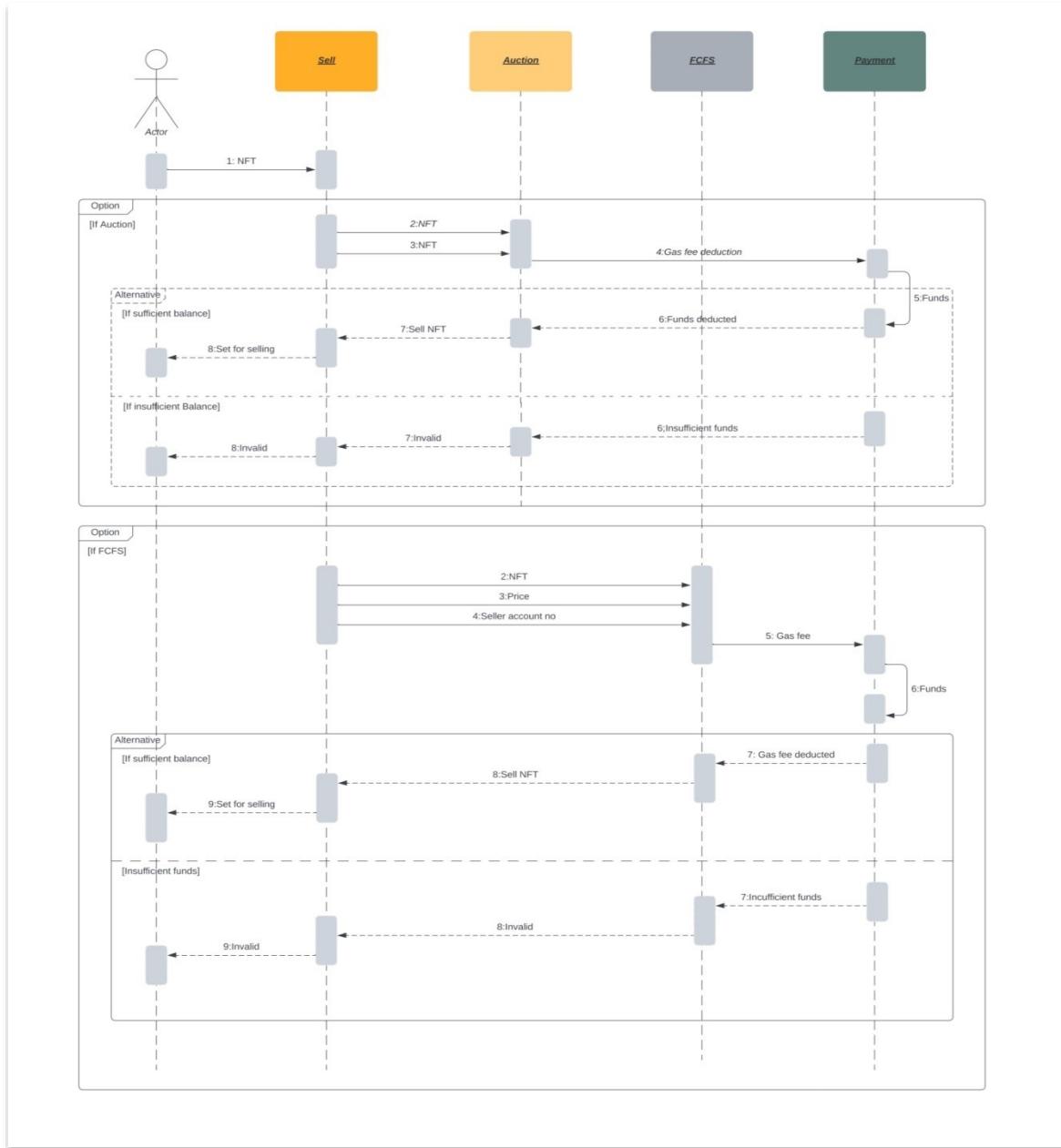


Figure 5.7 Sell NFT Sequence diagram

Figure 5.7 is sequence diagram to sell NFT at <NFT Connect>. It provides sequence of actions to a user perform to sell NFT and the system's response to those course of actions. Selling NFT can be done in two ways i.e. either at auction or at FCFS approach.

5.3.6 Buy NFT:

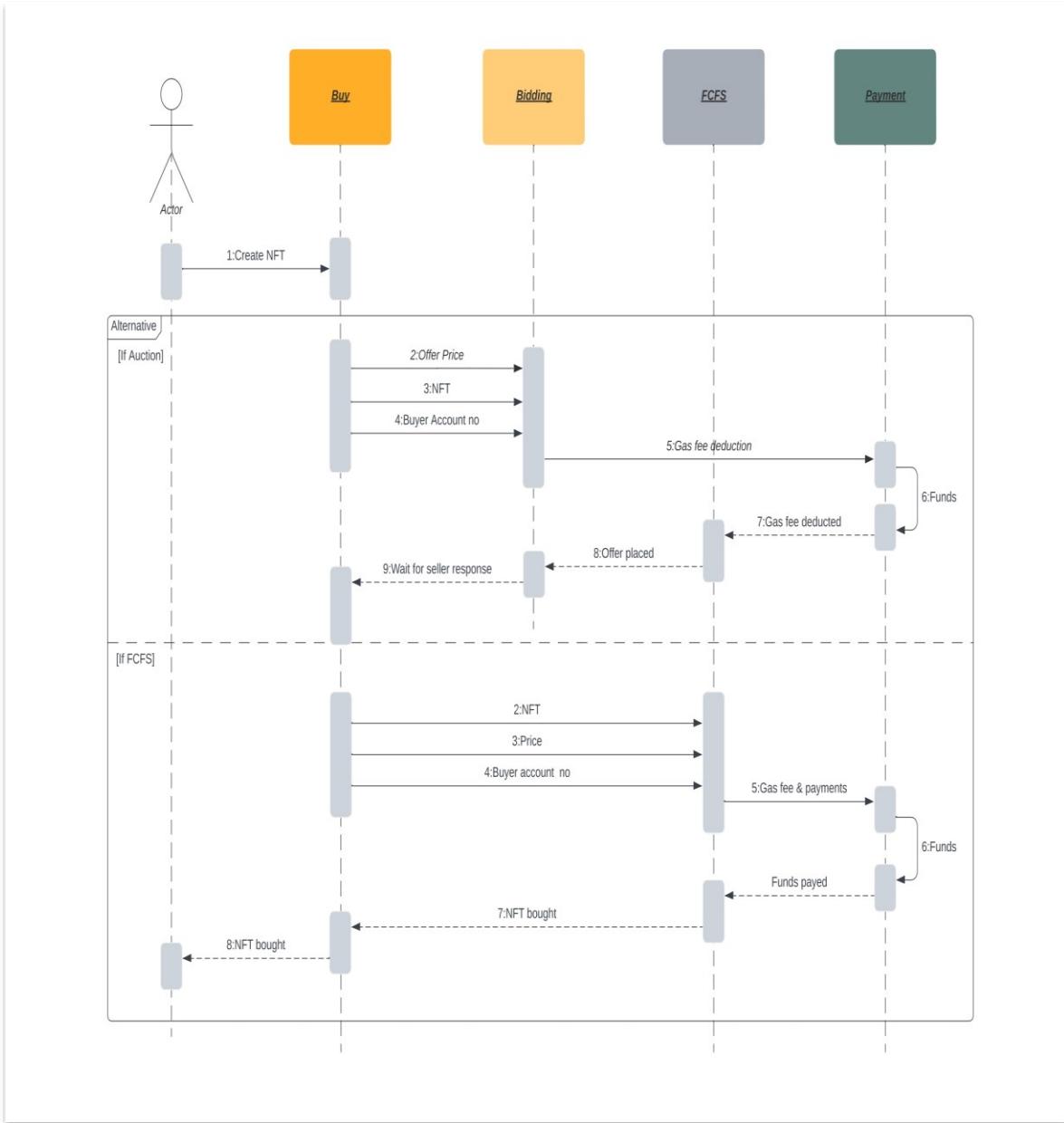


Figure 5.8 Buy NFT Sequence diagram

Figure 5.8 is sequence diagram to buy NFT at <NFT Connect>. It provides sequence of actions to a user perform to buy NFT and the system's response to those course of actions. If the NFT is at auction then the user place a bid and wait for seller's response. Otherwise, the user can directly buy the NFT by paying the gas fee and the purchasing cost of NFT.

5.4 Collaboration Diagram

5.4.1 Create Collection:

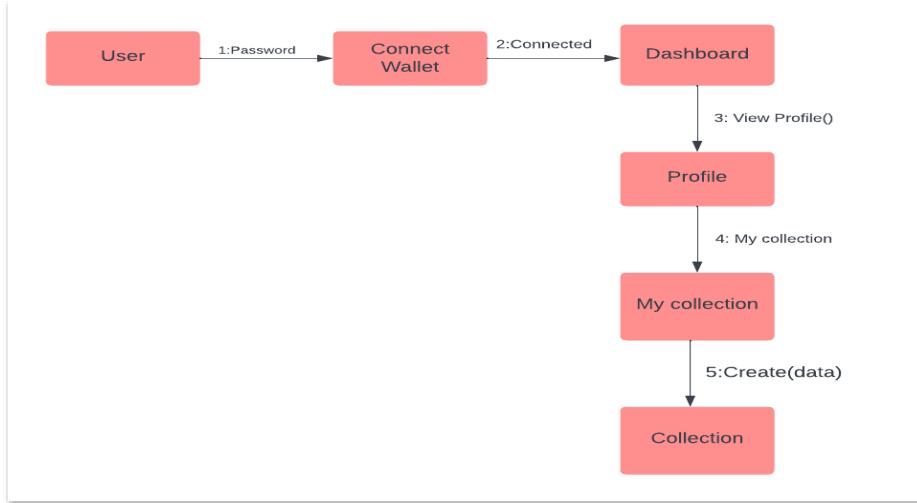


Figure 5.9 Create collection collaboration diagram

Figure 5.9 is a collaboration diagram showing the flow towards creating collection. It shows the structure of the objects and their interactions with the marketplace. Creating a collection is mandatory to create NFT.

5.4.2 Create NFT:

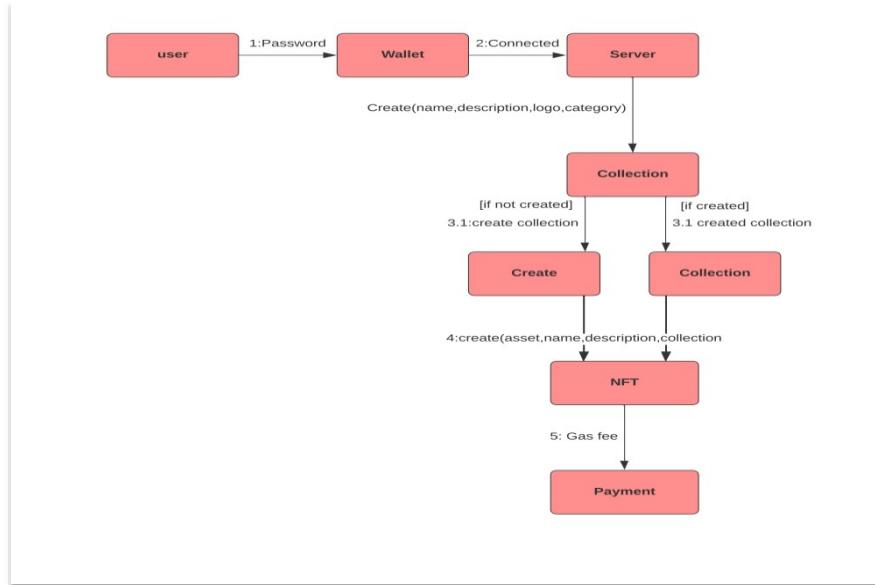


Figure 5.10 Create NFT collaboration diagram

Figure 5.10 is a collaboration diagram to create NFT at <NFT Connect>. It provides structure and organization towards creating an NFT. Creating an NFT requires a collection. So, if a user have not created an NFT, they should first create collection and then NFT.

5.4.3 Sell NFT:

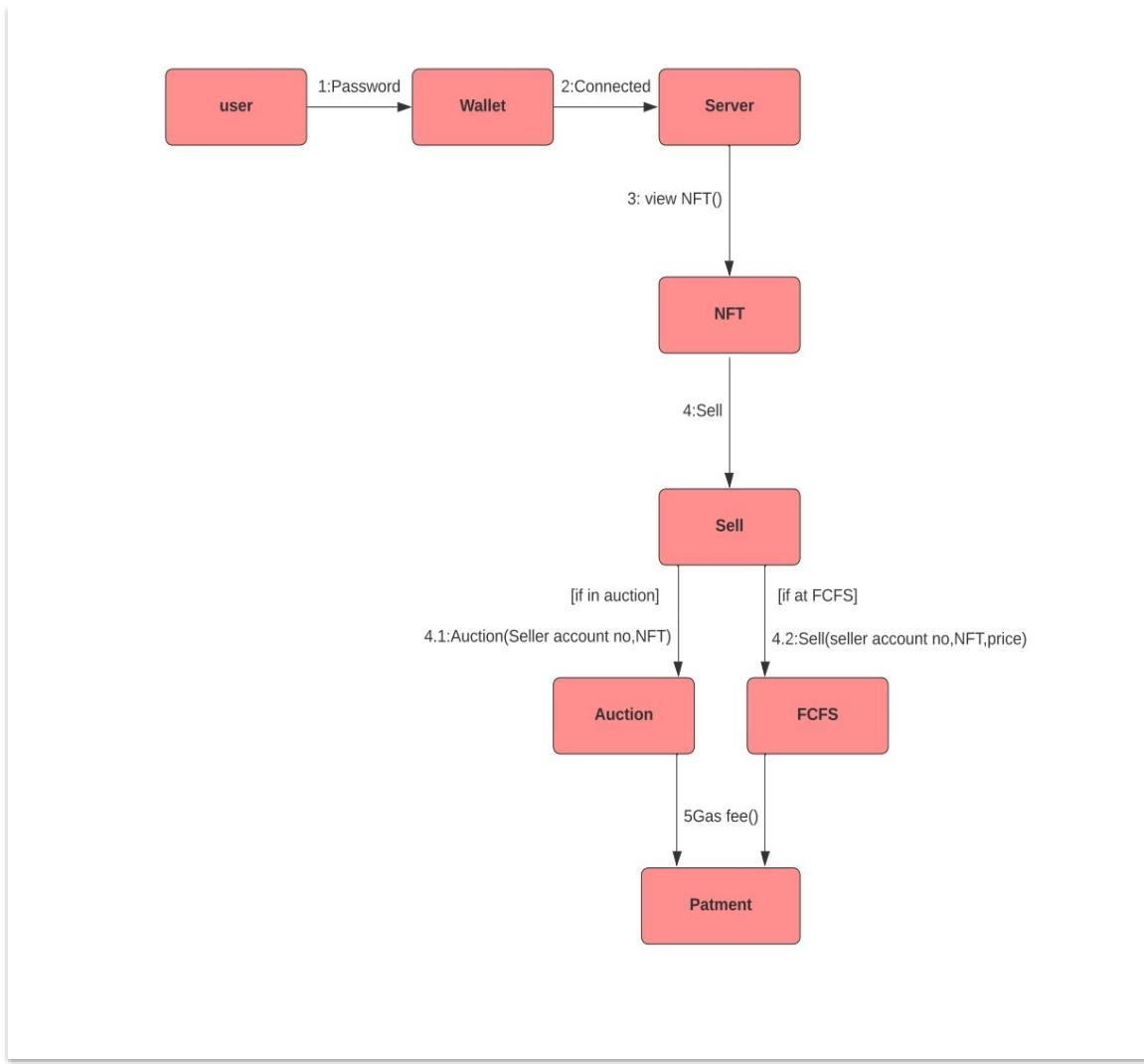


Figure 5.11 Sell NFT Collaboration diagram

Figure 5.11 is a collaboration diagram to sell NFT at <NFT Connect>. It provides structure and organization towards selling an NFT. At <NFT Connect> a user can sell their NFT in two ways. Either at auction or at FCFS approach. Selling NFT costs gas fee.

5.4.4 Buy NFT:

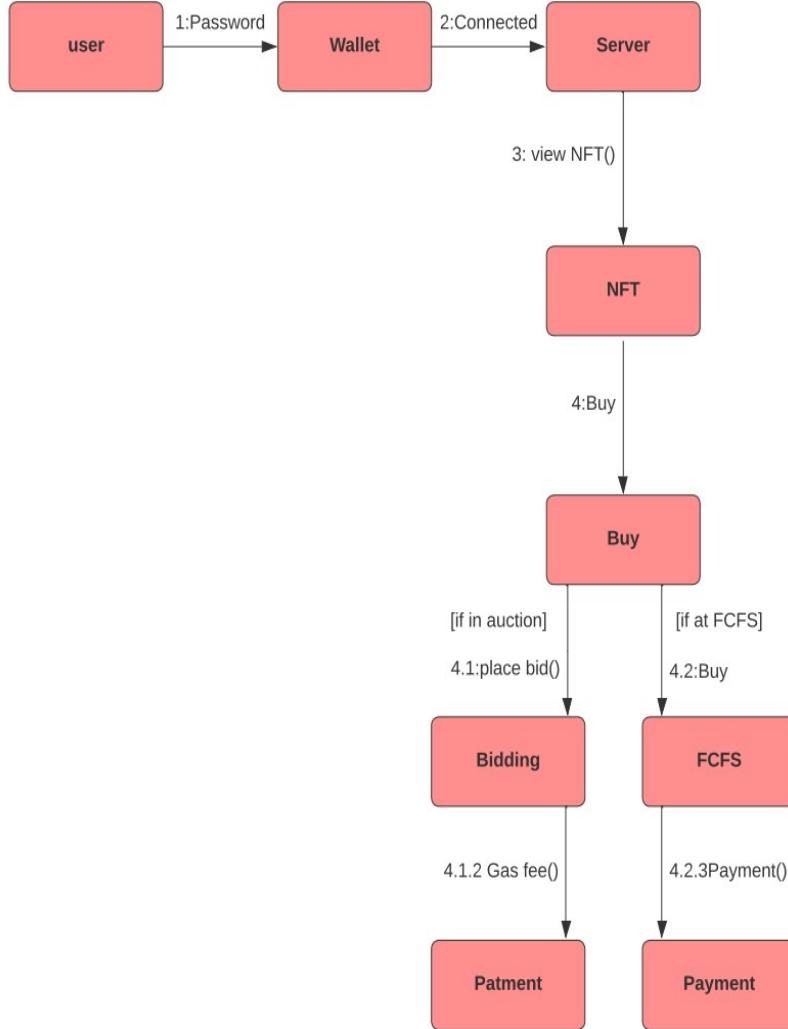


Figure 5.12 Buy NFT Collaboration Diagram

Figure 5.12 is a collaboration diagram to buy NFT at <NFT Connect>. It provides structure and organization towards buying an NFT. If the NFT is at auction then the buyer place a bid and wait for seller's response. Otherwise, the buyer just needs to pay the gas fee and the purchase amount of NFT.

5.5 Activity Diagram

5.5.1 Create Collection

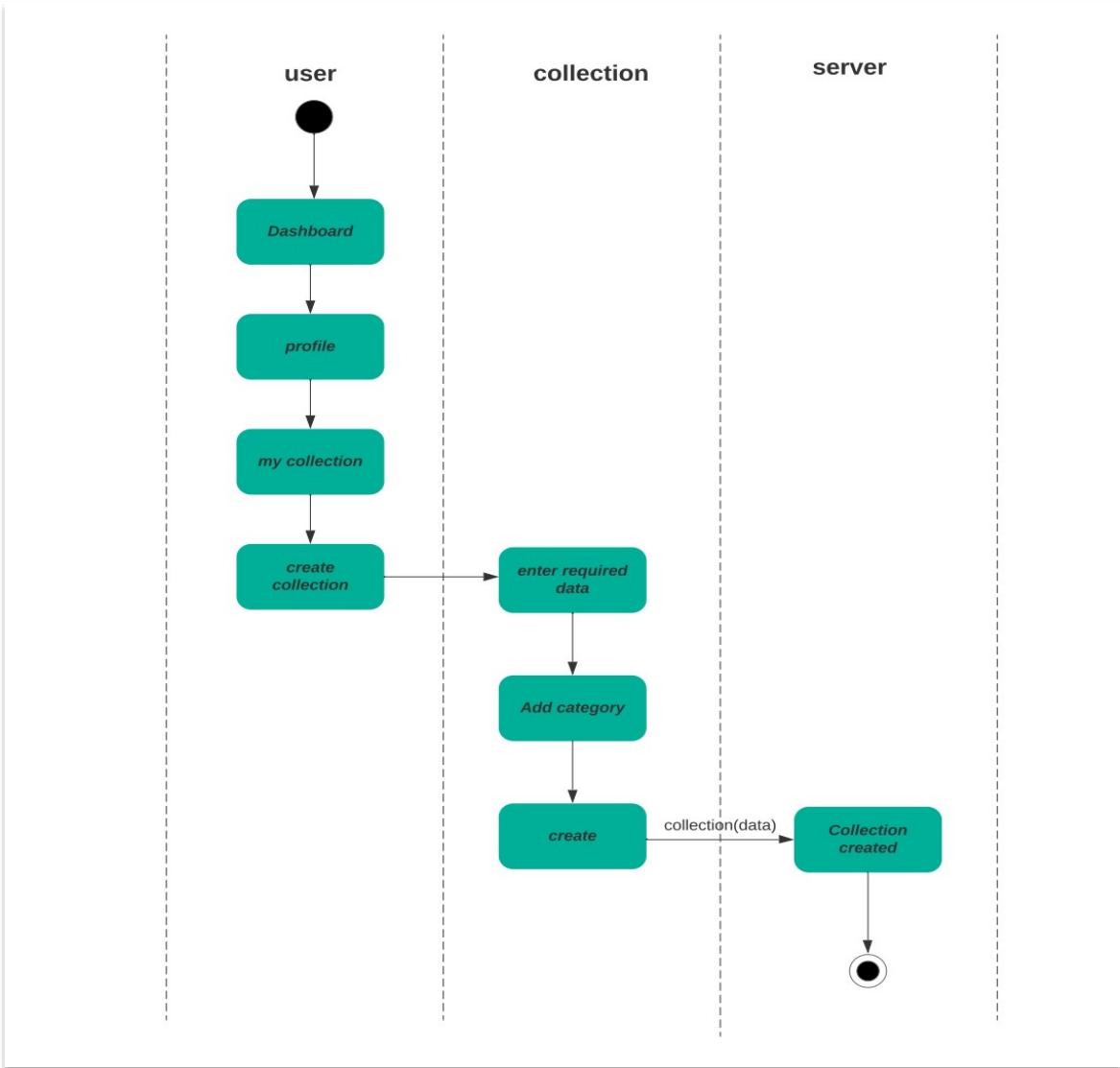


Figure 5.13 Activity Diagram Create Collection

Figure 5.13 is an activity diagram towards creating collection at <NFT Connect>. It shows the activity of user for creating collection. Creating a collection is mandatory for user as it contains the category of the NFT that will be associated with it.

5.5.2 Create NFT

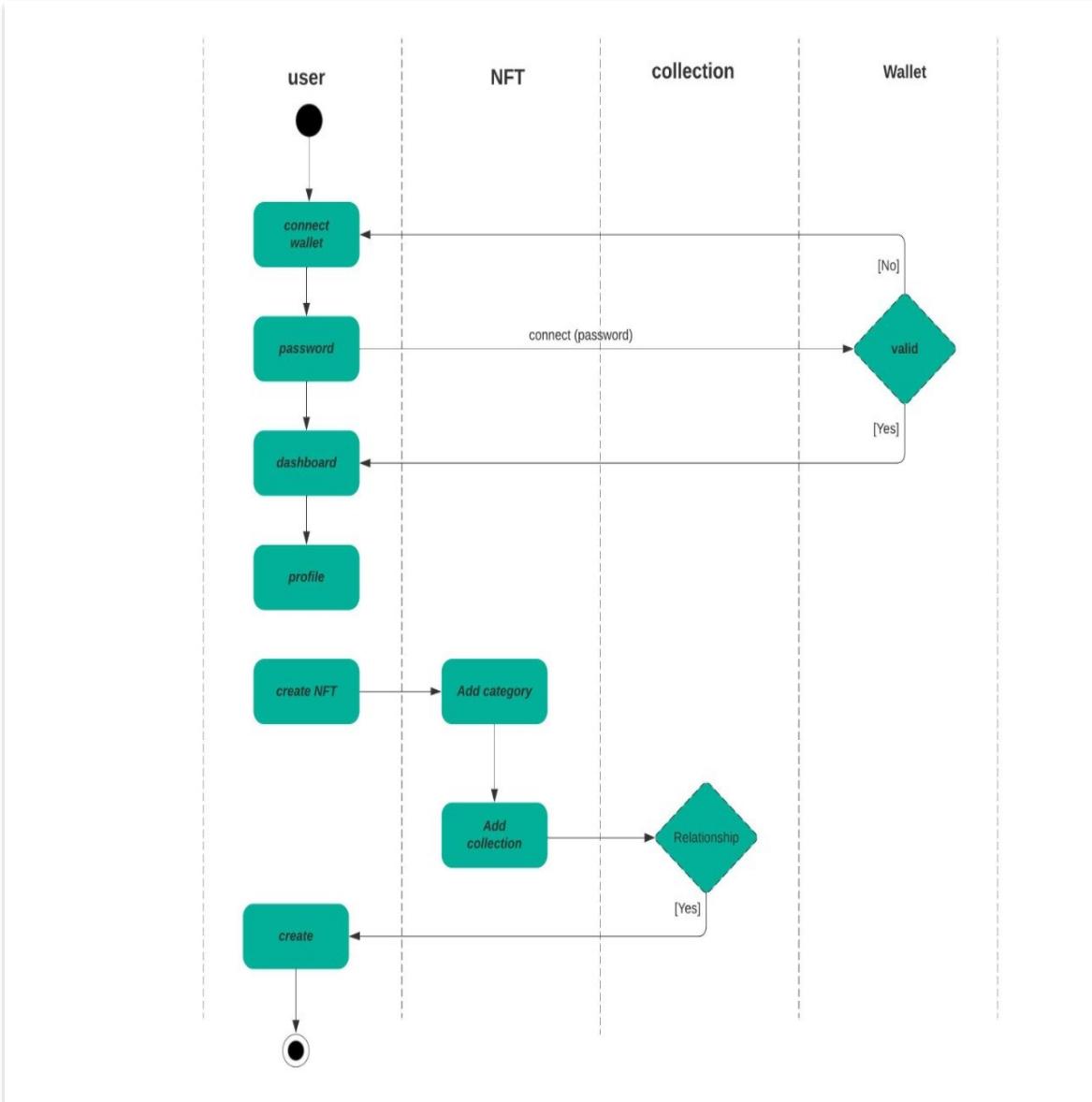


Figure 5.14 Activity diagram Create NFT

Figure 5.14 is an activity diagram towards creating NFT at <NFT Connect>. It shows the activity of user for creating NFT. Creating a collection is mandatory for user as it contains the category of the NFT that will be associated with it. NFT cannot be created if user have not created collection.

5.5.3 Sell NFT

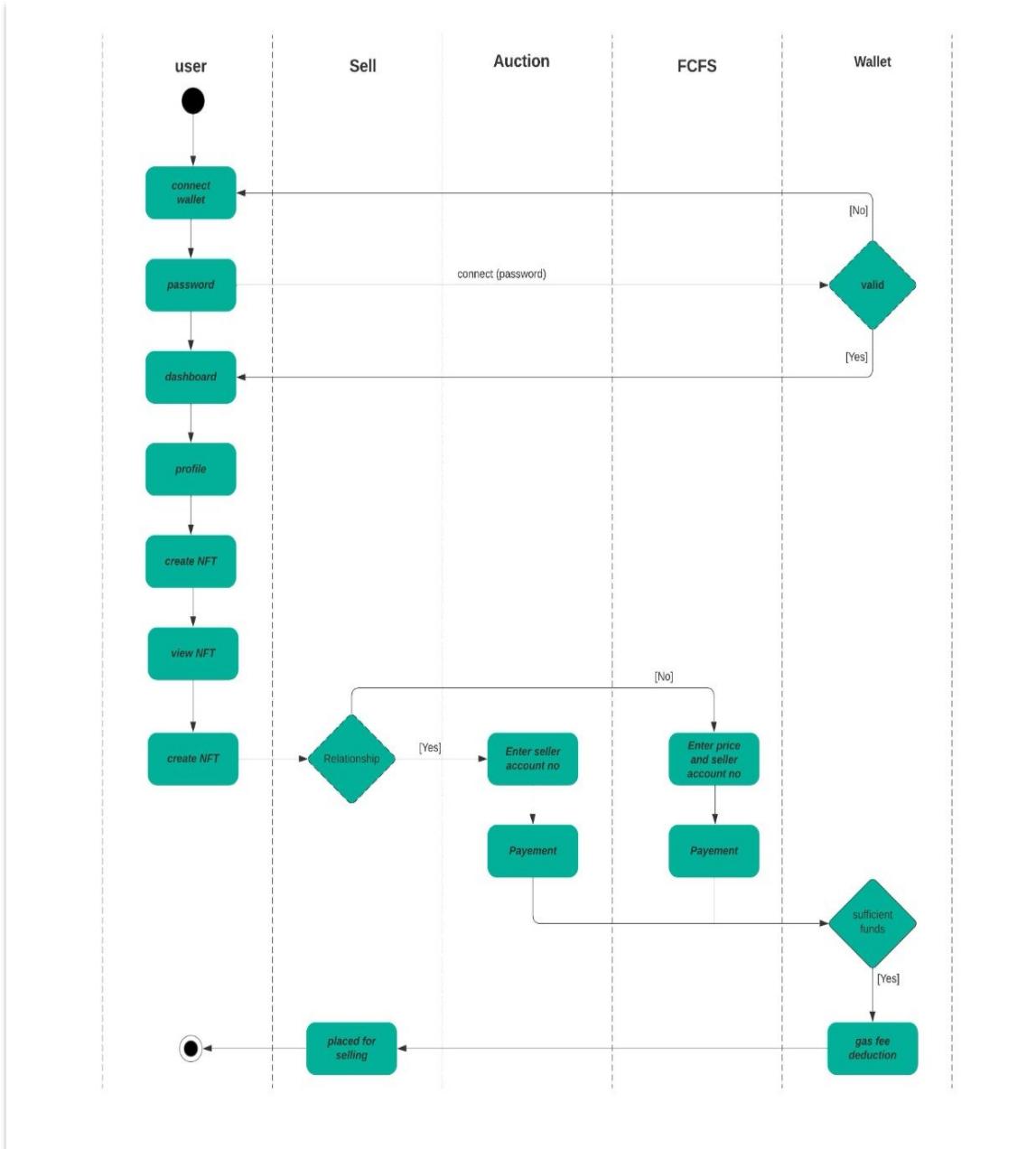


Figure 5.15 Activity diagram Sell NFT

Figure 5.15 is an activity diagram towards Selling NFT at <NFT Connect>. It shows the activity of user for Selling NFT. At <NFT Connect> a user can sell their NFT in two ways. Either at auction or at FCFS approach. Selling NFT costs gas fee.

5.5.4 Buy NFT

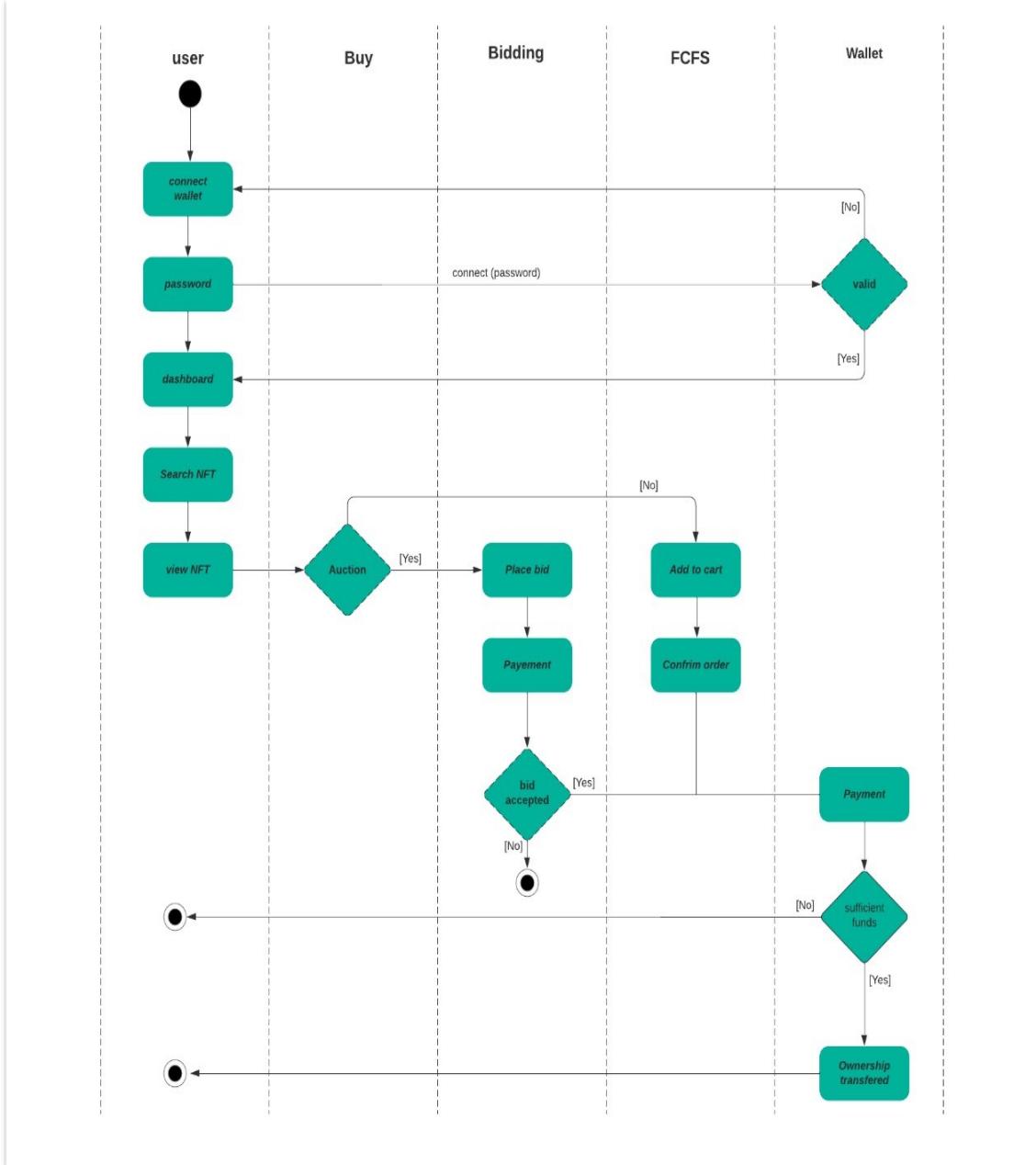


Figure 5.16 Activity Diagram Buy NFT

Figure 5.16 is an activity diagram towards Buying NFT at <NFT Connect>. It shows the activity of user for buying NFT. If the NFT is at auction then the buyer place a bid and wait for seller's response. Otherwise, the buyer just needs to pay the gas fee and the purchase amount of NFT.

5.6 State Diagram

5.6.1 Create Collection

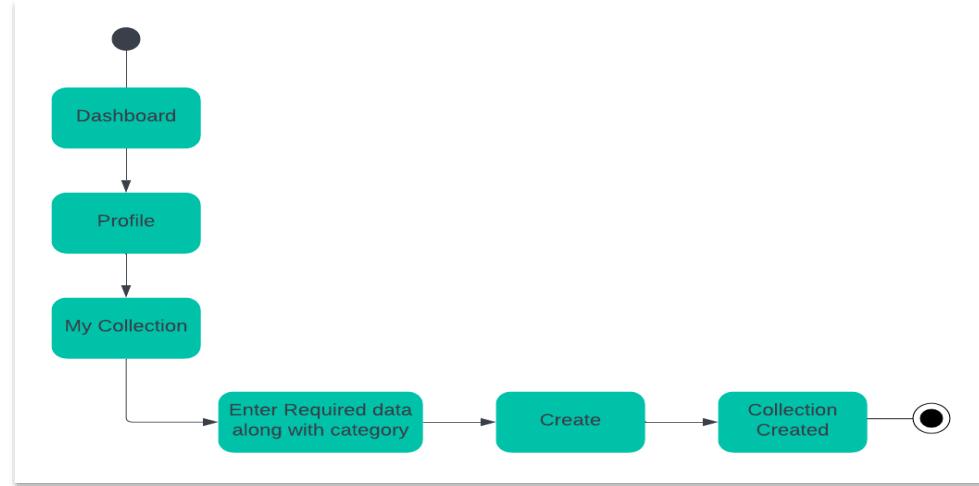


Figure 5.17 State Diagram Create Collection

Figure 5.17 is a state diagram towards creating collection at <NFT Connect>. It shows the change in state of the system as the user proceeds towards creating collection. Creating a collection is mandatory for user as it contains the category of the NFT that will be associated with it.

5.6.2 Buy NFT

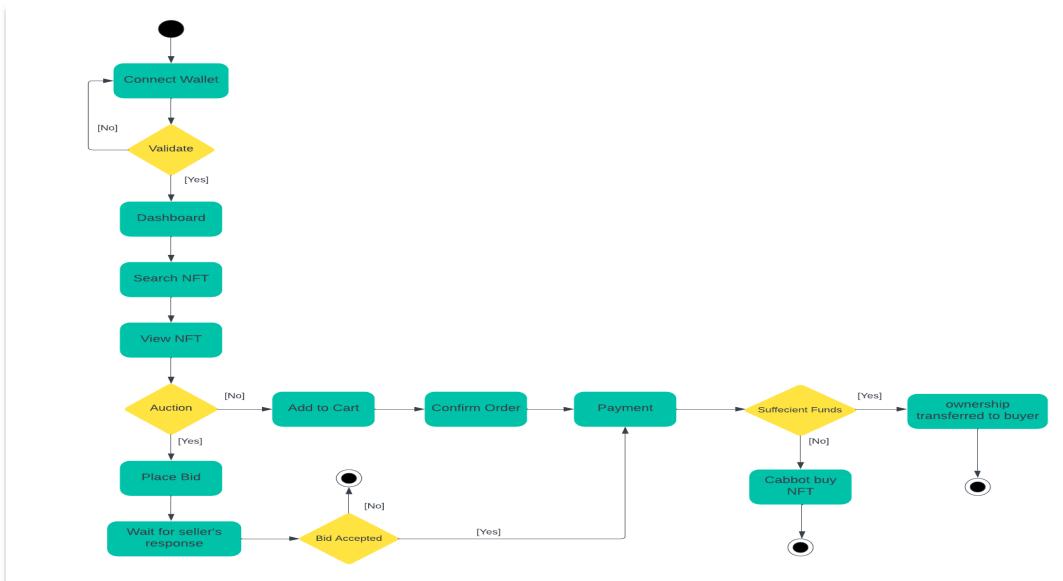


Figure 5.18 State Diagram buy NFT

Figure 5.18 is a state diagram towards buying NFT at <NFT Connect>. It shows the change in state of the system as the user proceeds towards buying NFT. If the NFT is at auction then the buyer place a bid and wait for seller's response. Otherwise, the buyer just needs to pay the gas fee and the purchase amount of NFT.

5.6.3 Create NFT

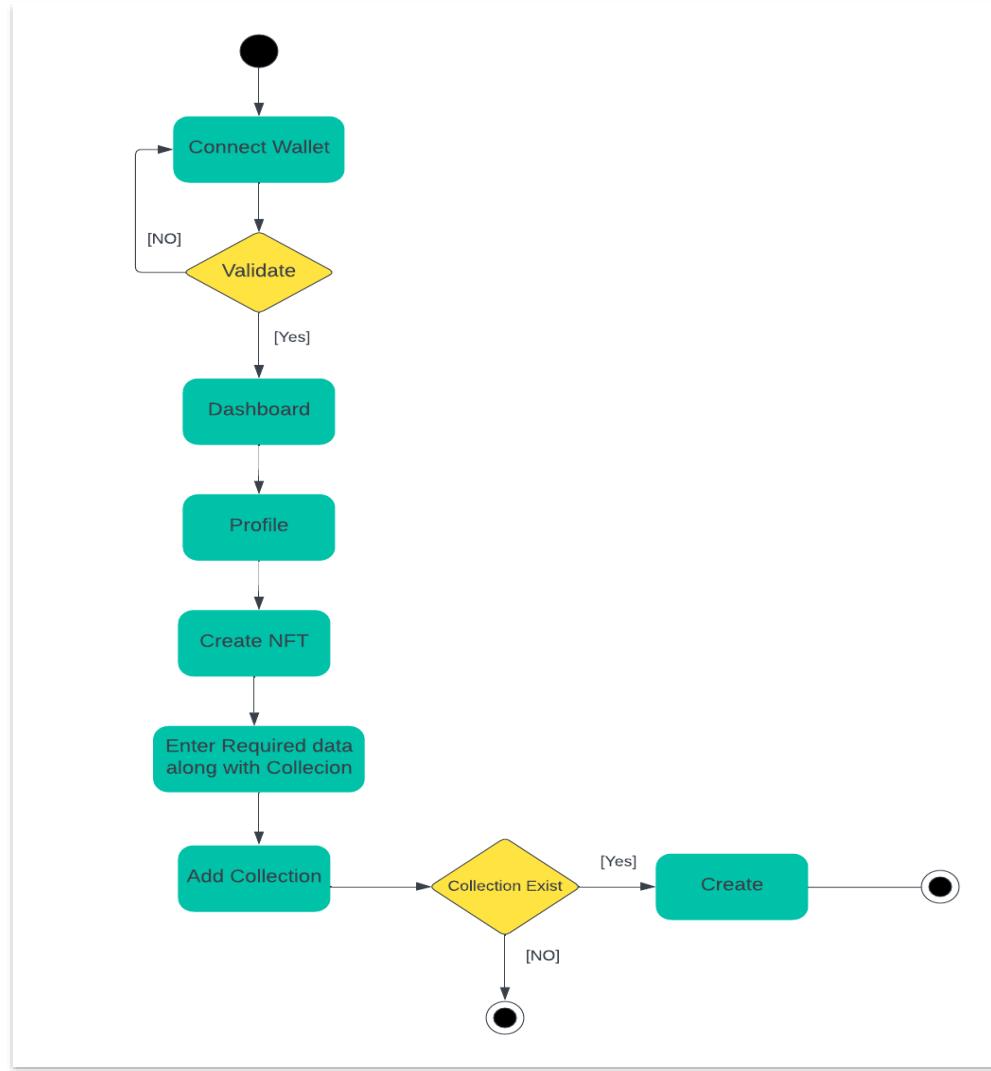


Figure 5.19 State Diagram Create NFT

Figure 5.19 is a state diagram towards creating an NFT at <NFT Connect>. It shows the change in state of the system as the user proceeds towards creating NFT. Creating a collection is mandatory for user as it contains the category of the NFT that will be associated with it. NFT cannot be created if user have not created collection.

5.6.4 Sell NFT

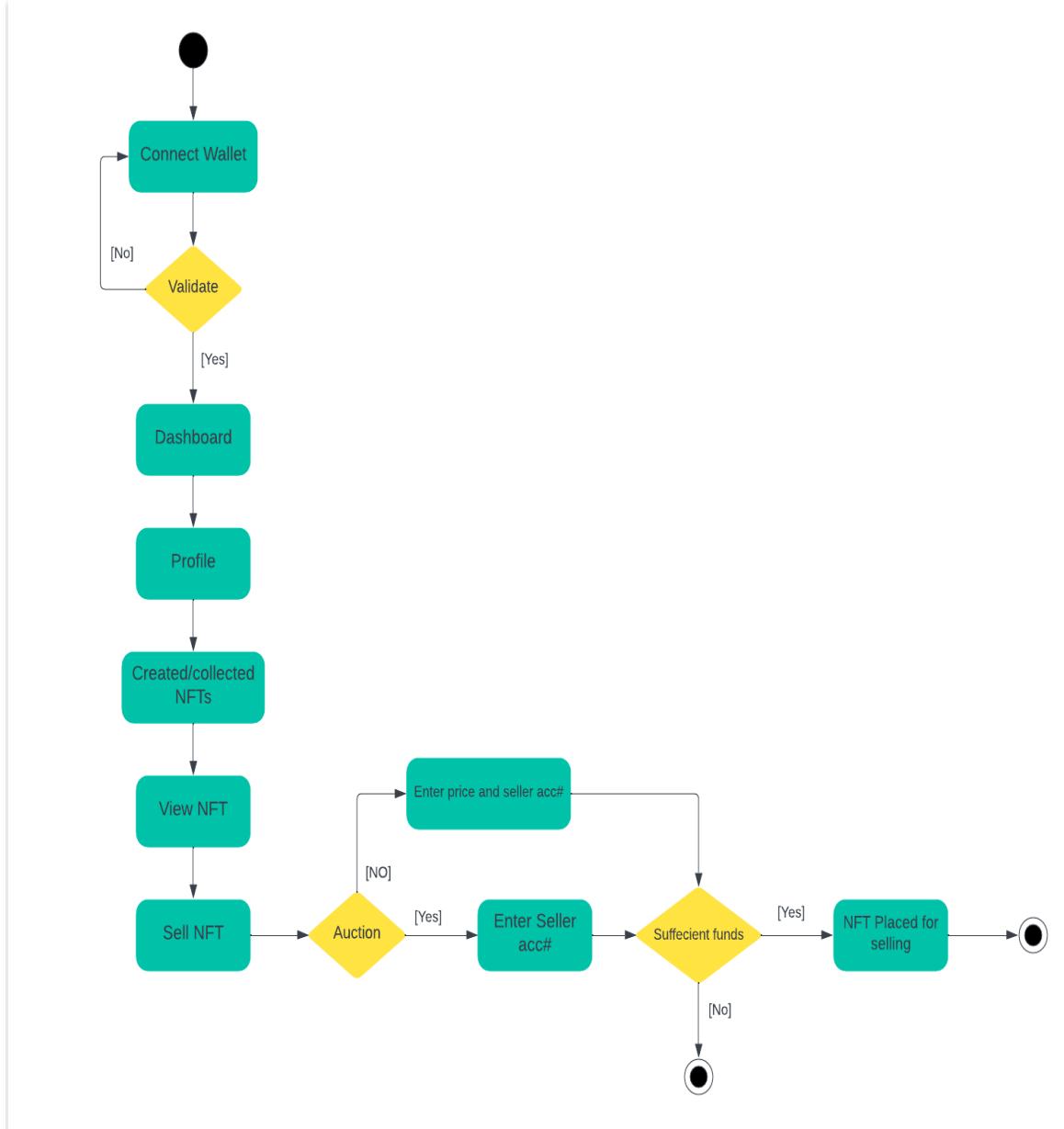


Figure 5.20 State Diagram Sell NFT

Figure 5.20 is a state diagram towards selling an NFT at <NFT Connect>. It shows the change in state of the system as the user proceeds towards selling NFT. At <NFT Connect> a user can sell their NFT in two ways. Either at auction or at FCFS approach. Selling NFT costs gas fee.

5.7 Entity Relationship Diagram

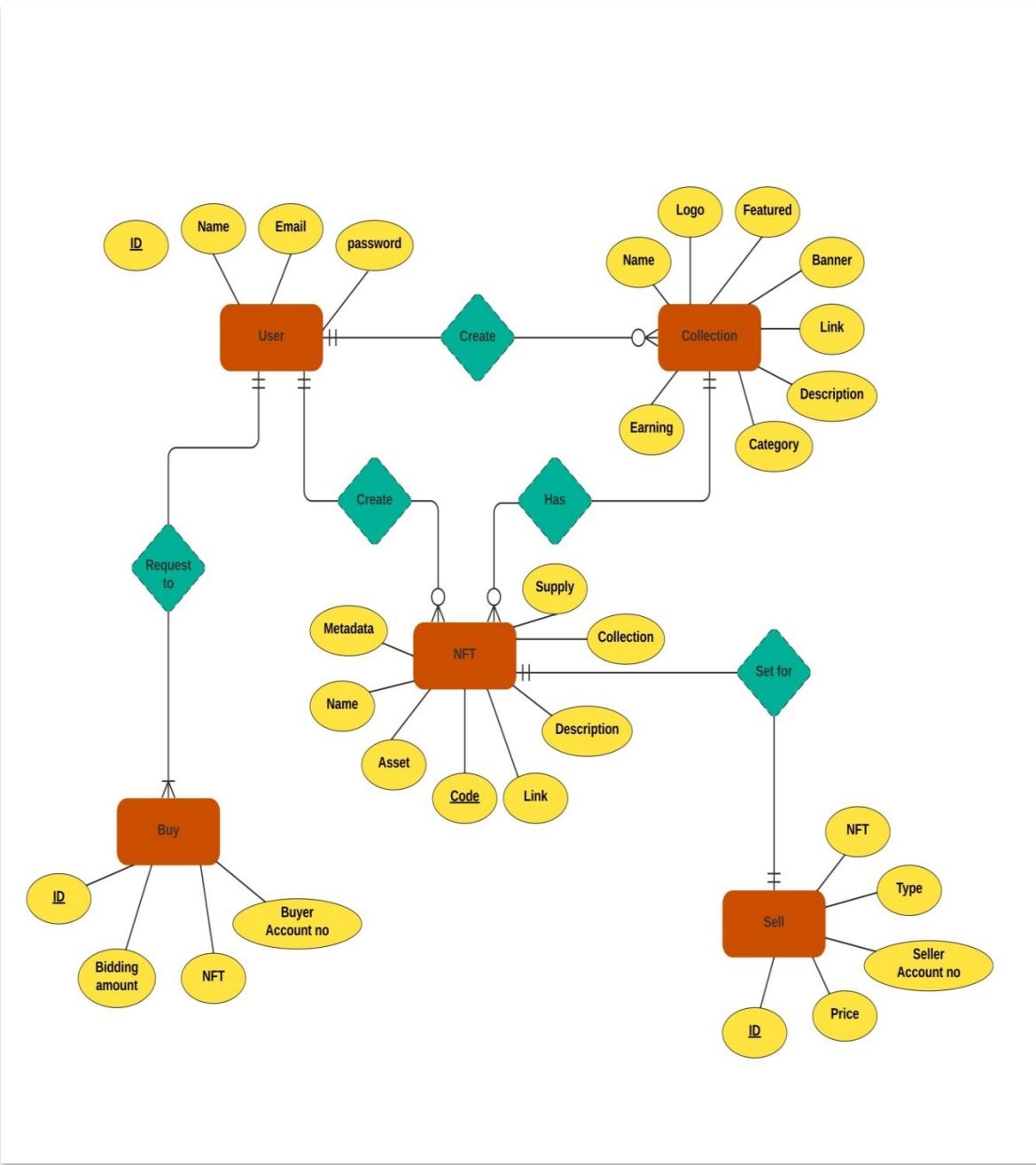


Figure 5.21 ERD

Figure 5.21 is an Entity relationship diagram of <NFT Connect>. Entity Relationship Diagram is a type of graphic that shows the connections between many elements of a system, including people, things, locations, ideas, and events.

5.8 Data Dictionary

5.8.1 User Dictionary:

Table 5.1 User Data Dictionary

Field Name	Type	Size	Constraint	Required	Description
Id	Int	10	Primary Key	Yes	Unique id of each user at the marketplace
Name	String	20	Not Null	Yes	Name of the user
Email	String	50	Not Null	Yes	Email id of the user
Password	String	50	Not Null	Yes	Password of account

The user using the platform must provide their data. Table 5.1 contains the data dictionary of the user. Data dictionary is metadata which means data about data. Table 5.1 gives complete information about user data.

5.8.2 Collection Dictionary

Table 5.2 Collection Data Dictionary

Field Name	Type	Size	Constraint	Required	Description
Featured	Img	100MB(max)	Jpg/png	No	This image will be the feature image of collection
Banner	Img	100MB(max)	Jpg/png	No	Banner image for the collection
Logo	Img	100mb(max)	Jpg/png	Yes	Logo of the collection
Link	String	50	Not Null	No	Link to the complete detail about the collection
Name	String	30	Not Null	Yes	Suitable name of the collection
Description	String	200	Not Null	No	Description of the collection

Earning	Float	2	Not Null	No	Revenue everytime the item in collection Is resaled
Category	String	20	Not Null	Yes	Category of the collection

The user willing to create an NFT at <NFT Connect> must create collection. Table 5.2 contains the data dictionary of collection. Data dictionary is metadata which means data about data. Table 5.2 gives complete information about collection data.

5.8.3 NFT Dictionary

Table 5.3 NFT Data Dictionary

Field Name	Type	Size	Constraint	Required	Description
Code	String	20	Primary Key	Yes	Unique identification code for each NFT
Asset	Img/vid/gif	100MB	Not Null	Yes	The digital asset to convert into NFT
Name	String	50	Not Null	Yes	Name of NFT
Link	String	50	Not Null	No	Link for the detail about NFT
metaData	String	100	Not Null	Yes	Data about the block on the blockchain
Description	String	150	Not Null	No	Description of NFT
Collection	Collection	602	Not Null	Yes	Already created collection
Supply	Int	2	Not Null	No	No of copies of NFT

The user willing to deal in NFT at <NFT Connect> first needs to create NFT. Table 5.3 contains the data dictionary of NFT. Data dictionary is metadata which means data about data. Table 5.3 gives complete information about NFT data.

5.8.4 Sell Dictionary

Table 5.4 Sell Data Dictionary

Field name	Type	Size	Constraint	Required	Description
Id	Int	20	Primary Key	Yes	Unique id of each nft on selling list
Nft	NFT	NFT	Not Null	Yes	NFT to sell
accountNo	String	80	Not Null	Yes	Seller account number
Price	Double	20	Not Null	No(for auction)	Price of NFT
Type	String	30	Not Null	Yes	Selling type, auction or FCFS

The user at <NFT Connect> willing to sell NFT provide relevant data. Table 5.4 contains the data dictionary of Sell NFT. Data dictionary is metadata which means data about data. Table 5.3 gives complete information about Sell data.

5.8.5 Buy Dictionary

Table 5.5 Buy Data Dictionary

Field name	Type	Size	Constraint	Required	Description
Id	Int	20	Primary Key	Yes	Unique id of each nft to be bought
Nft	NFT	NFT	Not Null	Yes	NFT to buy
accountNo	String	80	Not Null	Yes	Buyer account number
BiddingAmount	Double	20	Not Null	No(for auction)	bid of NFT
type	String	30	Not Null	Yes	Selling type, auction or FCFS

The data NFT being bought is stored in the data base. Table 5.5 contains the data dictionary of buy. Data dictionary is metadata which means data about data. Table 5.5 gives complete information about Buy data.

6. Implementation Details:

6.1 Development Setup:

6.1.1 ReactJS:

ReactJS is a popular JavaScript library for creating user interfaces. ReactJS's component-based design is one of its best qualities. Front-end of the NFT Connect is build using ReactJS. React function based components are used to create and implement the front-end. The Marketplace's user interface is broken up into smaller, independent components that offer a dynamic experience.

6.1.2 NodeJS:

A JavaScript runtime environment called NodeJS was created to support server side coding. The v8 engine in Chrome is used to construct NodeJS. NodeJS was created primarily to implement JavaScript outside of a web browser. In NFT Connect, is use to develop server side code to process the user information.

6.1.3 MongoDB:

MongoDB is a document-oriented, non-relational database management system. In NFT Connect, we are dealing with the data in JSON format and MongoDB best fit to provide data management for JSON objects. It stores the user data as well as the data of the NFTs. It stores the information of the current owner of the NFT.

6.1.4 ExpressJS:

ExpressJS is a web app framework for NodeJS build to create web servers and API's. NFT Connect used Express to create API's as it is implemented on the top of the NodeJS.

6.1.5 Solidity:

A high level language called Solidity was created to implement blockchain smart contracts. At NFT Connect, the self-executing agreements of the NFT are known as smart contracts and are encoded straight into code. NFT Connect is using Ethereum blockchain and ERC-721 token that is use to implement NFTs within smart contract. All these are implemented using solidity.

6.1.6 Visual Studio Code:

Visual Studio Code is an IDE used to that we used to develop front-end and the back-end of the marketplace. The development code of front-end in ReactJS and backend in NodeJS, ExpressJS and MongoDB are performed in this IDE.

6.2 Deployment Setup:

The pre-requisites for deploying the set-up include downloading and installing NodeJS. Deploy all the files mentioned below simultaneously.

6.2.1 ReactJS:

- Open the source code file in Visual studio code (vs code).
- Open the terminal in vs code
- Run the command “npm start”

6.2.2 NodeJS:

- Open the source code file in Visual studio code (vs code).
- Open the terminal in vs code
- Run the command “node main.js”

6.2.3 Blockchain:

- Set-up the development environment (vs code) and install dependencies : ethers, hardhat, @nomiclabs/hardhat-waffle, dotenv:
 - Npm install ethers
 - Npm install hardhat
 - Npm install nomiclabs/hardhat-waffle
 - Npm install dotenv
- Run the command ”yarn hardhat deploy –network sepolia/mainnet”

6.3 Algorithms:

Proof of work:

In order to protect the network and verify transactions, using Proof of Work (PoW) or Proof of Stake (PoS).

Authentication:

It produce key pairs using asymmetric key techniques like RSA where the private key is kept by the asset owner and the public key is used to confirm possession.

Searching and discovery:

In order to assist users in discovering NFTs based on their interests, related assets, or current trends. These algorithm incorporate content-based filtering, collaborative and filtering.

Auction:

NFT Connect uses decentralised auctions algorithm to set fair pricing based on supply and demand dynamics.

6.4 Constraints:

6.4.1 Assumptions:

NFT Connect is an NFT marketplace that enables its users to own their digital assets and to secure their assets for a lifetime. Following are the assumptions made for the development of NFT Connect:

6.4.1.1 NFTs:

NFT Connect assumes that any digital asset like artwork, collectibles, pictures etc. can be converted into an NFT. NFT connect allows its users to deal in NFTs.

6.4.1.2 Blockchain:

NFT Connect implements blockchain to keep the transactions secure and to maintain the integrity of the data. The marketplace uses Ethereum blockchain and is implemented in solidity.

6.4.1.3 ERC-721:

ERC-721(Ethereum request for comments -721) symbolizes the possession of a unique digital item. These are the standard set of rules to convert a digital item into an NFT. NFTs are non-fungible which means any other item cannot replace a unique digital item.

6.4.1.4 MetaData:

NFTs at NFT Connect are assumed to have metadata. Title, description, subtitle and other related data of NFTs is also maintained.

6.4.1.5 Features:

The main functions performed at the NFT Connect includes sell, buy and transfer NFTs. Selling an NFT can be done in two ways at the marketplace i.e. at fixed price and at auction. The ownership rights are transferred to the one who buy the NFT.

6.4.1.6 Transparent transactions:

The transactions carried out at the marketplace are transparent and secure. Ownership rights are trustable. Payments are managed correctly by maintaining the transaction records.

6.4.1.7 User wallet:

The user is assumed to have a crypto wallet account e.g. MetaMask that is compatible for managing NFTs and cryptocurrency.

6.4.2 System Constraints:

System constraints affect the performance of the marketplace. Following are some system constraints to optimize the performance:

6.4.2.1 Transaction cost:

In order to carry out smart contracts and transactions on a public blockchain network, it requires gas fee. These expenses can impose restrictions to the usage and affordability of the marketplace. To deal with such a constraint, the marketplace should be designed in a way to optimize gas usage and clear charge structures for customers should be taken into account.

6.4.2.2 Storage space:

NFT Connect serves its users by storing the digital asset/NFT and its corresponding metadata. To accommodate the expanding NFT ecosystem, enough storage and bandwidth resources are required. The ideal place to store NFTs is a wallet that support NFTs.

6.4.2.3 Cryptography Methods:

Computational overhead may be introduced by the use of cryptography such as hashing, digital signatures, and encryption. Consideration should be given to the performance of cryptographic operations to guarantee efficient processing, particularly in situations with high user concurrency and transaction volumes.

6.4.2.4 External dependencies:

The integration of NFT Connect with external platforms such as payment gateways, digital wallets, or external APIs, is a necessity to store cryptocurrency and NFT. The functionality and overall user experience of the marketplace may be restricted by the compatibility and availability of these connections.

6.4.2.5 Maintenance:

To guarantee the marketplace's operation, security, and compliance with emerging standards and protocols, ongoing maintenance and frequent upgrades are required. To

minimize disturbance to current operations during repair activities, careful planning is required.

6.4.3 Restrictions:

Following are certain limits and restrictions that are taken into account while developing NFT Connect.

6.4.3.1 Gas and transaction size restrictions:

Due to gas caps and block size restrictions, public blockchain networks like Ethereum place restrictions on the volume and complexity of transactions. These restrictions may make it difficult to trade larger NFTs or carry out complicated transactions.

6.4.3.2 Financial limitations:

Users are facilitated by listing gas fee at NFT Connect. The users not having enough cryptocurrency in their wallet to pay listing fee of the NFT, can list their NFT for selling without paying the fee. But once that NFT is sold the fee is deducted from the seller's wallet.

6.4.3.3 Crypto Wallet:

The users at NFT Connect are required to connect their external crypto wallet (a wallet supporting cryptocurrency and NFT) to the marketplace. Without connecting the crypto wallet, no one can deal in NFTs. One can see the listed NFTs on the marketplace but cannot buy it. Buying, selling, creating or transferring NFT requires the user to connect their wallet.

6.4.3.4 Regularity compliance:

NFT Connect abides by all relevant laws and rules, including financial rules, consumer protection, data privacy, and intellectual property rights. Complying with the juristic nature of the marketplace, the user's private data is kept confidential, only the NFT related data is made public. Every time user makes a transaction at the marketplace, a digital signature to the crypto wallet is required to carry out secure transactions.

6.4.3.5 Smart Contract:

At NFT Connect, smart contract is designed to carry out a variety of functions, such as token production, ownership monitoring, and transactions.

6.4.4 Limitations:

6.4.4.1 Interoperability limitations:

NFT Connect is built on the single blockchain network i.e Ethereum. It does not allow to transfer the NFTs to other blockchain networks. Therefore, the transferability and liquidity of the NFTs is limited.

6.4.4.2 Scalability issues:

NFT Connect is built on Ethereum blockchain, which has scalability issues. NFTs' rising popularity strains the network, increasing transaction fees and slowing down transaction times. The user experience may suffer as a result.

6.4.4.3 Limited payment methods:

NFT Connect does not provide its users a variety of payment channels. It allows payment through crypto wallet and in ethers only. No cryptocurrency, other than ethers, is supported. On the other hand, it doesn't allow to pay through credit cards, bank accounts or debit cards. Transactions are only carried out through crypto wallets.

7. Testing:

7.1 Extended Test cases:

7.1.1 Connect Crypto Wallet

Table 7.1 Test Case to Connect Valid Wallet Account

Test Case ID	TC_NFT_001		Designed by	Noureen Butt					
Test Module Name	Connecting valid external wallet			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Wallet account authentication			Execution Date	5/30/2023				
Description	Testing wallet connection to the marketplace.								
Pre-condition	The user is required to have an external crypto wallet account that support both cryptocurrency and NFT.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status			
1	Open the marketplace								
2	Click the wallet button								
3	Fill the required fields		Password:b477N96U						
4	Click unlock button			Wallet connected	Wallet connected	pass			
Post Condition		The user wallet is authenticated, now the user can deal in NFT.							

In order to connect the wallet to the marketplace, the user must have an external crypto wallet account. If the user does not possess an account then they should first create the account and then connect it to the marketplace. Table 7.1 completely defines the case when the user connected a valid account.

Table 7.2 Test Case to Connect Invalid Wallet Account

Test Case ID	TC_NFT_002		Designed by	Noureen Butt					
Test Module Name	Connecting invalid external wallet			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Wallet account authentication			Execution Date	5/30/2023				
Description	Testing wallet connection to the marketplace.								
Pre-condition	The user is required to have an external crypto wallet account that support both cryptocurrency and NFT.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status			
1	Open the marketplace								
2	Click the wallet button								
3	Fill the required fields		Password:b477N96U						
4	Click unlock button			Wallet not connected	Wallet not connected	pass			
Post Condition		The user wallet is not authenticated, the user can only view the NFTs.							

Table 7.2 shows a test case where an invalid crypto wallet account is entered. The user without connecting a valid account can only view NFTs. He/She neither can deal in NFTs nor can avail all the features of the marketplace Without a valid account to crypto wallet.

7.1.2 Create Collection

Table 7.3 Test case to create Collection

Test Case ID	TC_NFT_003		Designed by	Noureen Butt						
Test Module Name	Create Collection		Design Date	5/15/2023						
Test Priority	High		Executed by	Durrez Ahmad						
Test Title	Create collection test		Execution Date	5/30/2023						
Description	Testing create Collection.									
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.									
Dependencies	External Crypto Wallet									
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes				
1	Open the marketplace									
2	Check if the wallet is connected. If not then connect it first									
3	Click the my collection tab from the navbar and navigate to the create collection page.									
4	Fill in the required fields	Logo image: image.png Name: My collection Description: this is the description of the collection Category: art								
5	Click create button		Collection created	Collection created	pass					
Post Condition		The user is navigated to the My Collection tab.								

Table 7.3 shows the successful test case to create collection at NFT Connect. The pre-requisites for creating collection is connecting a valid crypto wallet to the marketplace.

7.1.3 Create NFT:

Table 7.4 Test Case to Create NFT with Collection

Test Case ID	TC_NFT_004		Designed by	Noureen Butt					
Test Module Name	Create NFT			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Create NFT with collection test			Execution Date	5/30/2023				
Description	Testing create NFT.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace and a collection.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status			
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Click create button from the navbar and navigate to the create collection page.								
4	Fill in the required fields		Image: myImage.png Name: aesthetic world Description: this is the description of the NFT Collection: My Collection						
5	Click create button			NFT created	NFT created	pass			
Post Condition		The user is navigated to the NFT page where they can see their created NFT.							

Table 7.4 shows the successful test case when the user provide the required information along with the image to be converted into an NFT.

Table 7.5 Test Case to Create NFT without Collection

Test Case ID	TC_NFT_005		Designed by	Noureen Butt					
Test Module Name	Create NFT		Design Date	5/15/2023					
Test Priority	High		Executed by	Durrez Ahmad					
Test Title	Create NFT without collection test			Execution Date	5/30/2023				
Description	Testing create NFT.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace and a collection.								
Dependencies	External Crypto Wallet								
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes			
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Click create button from the navbar and navigate to the create collection page.								
4	Fill in the required fields	Image: myImage.png Name: aesthetic world Description: this is the description of the NFT Collection: Not Created							
5	Click create button		Collection required	Collection required	pass				
Post Condition	The user is navigated to the NFT page where they can see their created NFT.								

Table 7.5 shows test case where NFT is tried to create without creating a collection. Collection is a pre-requisite for creating an NFT. Without collection, NFT cannot be created.

7.1.4 Sell NFT:

Table 7.6 Test Case to Sell NFT at Fixed Price

Test Case ID	TC_NFT_006		Designed by	Noureen Butt					
Test Module Name	Sell NFT at fixed rate		Design Date	5/15/2023					
Test Priority	High		Executed by	Durrez Ahmad					
Test Title	Sell NFT at fixed rate test		Execution Date	5/30/2023					
Description	Testing Sell NFT at fixed rate.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace and a created NFT.								
Dependencies	External Crypto Wallet								
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes			
1	Check if the wallet is connected. If not then connect it first								
2	Click My Collection Button								
3	Select the collection and then the NFT you are willing to sell								
4	Click list to sell button								
5	Select sell at fixed rate								
6	Fill the required fields	Price: 4 Eth							
7	Provide digital signature to wallet account	authenticated							
8	Click List item button		Listed for selling	Listed for selling	pass				
Post Condition	The user is navigated to the NFT page where they can see their listed NFT.								

Table 7.6 shows the test case to sell the NFT at a fixed rate. The pre-requisites for the module are a valid crypto account connected to the marketplace and an NFT. The user then can sell any of their NFTs.

Table 7.7 Test Case to Sell NFT at Auction

Test Case ID	TC_NFT_007		Designed by	Noureen Butt						
Test Module Name	Sell NFT at auction		Design Date	5/15/2023						
Test Priority	High		Executed by	Durrez Ahmad						
Test Title	Sell NFT at auction test		Execution Date	5/30/2023						
Description	Testing Sell NFT at auction.									
Pre-condition	The user is required to connect their external crypto wallet to the marketplace and a created NFT.									
Dependencies	External Crypto Wallet									
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes				
1	Check if the wallet is connected. If not then connect it first									
2	Click My Collection Button									
3	Select the collection and then the NFT you are willing to sell									
4	Click list to sell button									
5	Select sell at auction									
6	Provide the digital signature to your wallet account	authenticated								
7	Click List item button		Listed for selling	Listed for selling	pass					
Post Condition	The user is navigated to the NFT page where they can see their listed NFT.									

Table 7.7 shows the test case to sell NFT at auction. It requires the user to select the NFT and provide digital signature to their wallet account.

7.1.5 Buy NFT:

Table 7.8 Test Case to Buy NFT at Fixed Rate with Sufficient Funds

Test Case ID	TC_NFT_008		Designed by	Noureen Butt					
Test Module Name	Buy NFT at fixed rate			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Buy NFT at fixed rate with sufficient funds test			Execution Date	5/30/2023				
Description	Testing Buy NFT at fixed rate with sufficient funds in the account.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status			
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Select the NFT to buy								
4	Select Buy now button								
5	Fill in the required fields		Fill in the address of the crypto wallet						
6	Select continue			NFT bought	NFT bought	pass			
Post Condition		The user is navigated to the NFT page where they can see their bought NFT.							

Table 7.8 shows the test case to buy NFT at fixed rate with sufficient funds in the account. After paying the required amount, the ownership rights are transferred to the buyer.

Table 7.9 Test case to Buy NFT at Fixed Rate Without Sufficient Funds

Test Case ID	TC_NFT_009		Designed by	Noureen Butt					
Test Module Name	Buy NFT at fixed rate		Design Date	5/15/2023					
Test Priority	High		Executed by	Durrez Ahmad					
Test Title	Buy NFT at fixed rate with insufficient funds test			Execution Date	5/30/2023				
Description	Testing Buy NFT at fixed rate with insufficient funds in the account.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.								
Dependencies	External Crypto Wallet								
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes			
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Select the NFT to buy								
4	Select Buy now button								
5	Fill in the required fields	Fill in the address of the crypto wallet							
6	Select continue		Cannot proceed	Cannot proceed	pass				
Post Condition	The user restricted to proceed ahead in buying NFT.								

Table 7.9 shows the test case to buy NFT at fixed rate with insufficient funds in the account. The user is not allowed to proceed ahead if they have insufficient funds in their account.

Table 7.10 Test Case to Buy NFT at Auction With Sufficient Funds

Test Case ID	TC_NFT_010		Designed by	Noureen Butt					
Test Module Name	Buy NFT at auction			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Buy NFT at auction with sufficient Funds in account/wallet test			Execution Date	5/30/2023				
Description	Testing Buy NFT at auction with sufficient funds in account/wallet.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status	Notes		
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Select the NFT to buy								
4	Select Offer button								
5	Fill in the required fields		Offered Price: 0.3 ETH Wallet address :XXXXXX						
6	Select continue			Offer placed	Offer placed	pass			
Post Condition		The user is navigated to the NFT page where they can see their bought NFT.							

Table 7.10 shows the test case to buy NFT at auction. The user is required to fill in the price they want to offer to buy the NFT along with their wallet address.

Table 7.11 Test Case to Buy NFT with Insufficient Funds

Test Case ID	TC_NFT_011		Designed by	Noureen Butt					
Test Module Name	Buy NFT at auction			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Buy NFT at auction without sufficient Funds in wallet test			Execution Date	5/30/2023				
Description	Testing Buy NFT at auction with sufficient funds in wallet.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.								
Dependencies	External Crypto Wallet								
Step	Test Step		Test Data	Expected Result	Actual Result	Status	Notes		
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Select the NFT to buy								
4	Select Offer button								
5	Fill in the required fields		Offered Price: 0.3 ETH Wallet address :xxxxxx						
6	Select continue			Cannot Proceed	Cannot proceed	pass			
Post Condition		The user restricted to proceed ahead in buying NFT.							

Table 7.11 shows the test case to buy NFT at auction with insufficient funds. The user is not allowed to proceed in the buying process if they have insufficient balance.

7.1.6 Edit NFT:

Table 7.12 Test Case to Edit NFT

Test Case ID	TC_NFT_012		Designed by	Noureen Butt						
Test Module Name	Edit NFT		Design Date	5/15/2023						
Test Priority	High		Executed by	Durrez Ahmad						
Test Title	Edit NFT test		Execution Date	5/30/2023						
Description	Testing edit NFT feature.									
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.									
Dependencies	External Crypto Wallet									
Step	Test Step		Test Data	Expected Result	Actual Result	Status	Notes			
1	Open the marketplace									
2	Check if the wallet is connected. If not then connect it first									
3	Select the NFT from collection to edit									
4	Select Edit item									
5	Edit the Detail of the NFT you want to change		Name: Utilities							
6	Select save changes			NFT Edited	NFT Edited	pass				
Post Condition		NFT edited.								

Table 7.12 shows the test case to edit NFT at NFT Connect. The pre-requisites for editing an NFT include connecting a crypto wallet to the marketplace and NFT owned.

7.1.7 Delete NFT

Table 7.13 Test Case to Delete NFT

Test Case ID	TC_NFT_013		Designed by	Noureen Butt						
Test Module Name	Delete NFT		Design Date	5/15/2023						
Test Priority	High		Executed by	Durrez Ahmad						
Test Title	Delete NFT test		Execution Date	5/30/2023						
Description	Testing Delete NFT feature.									
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.									
Dependencies	External Crypto Wallet									
Step	Test Step		Test Data	Expected Result	Actual Result	Status				
1	Open the marketplace									
2	Check if the wallet is connected. If not then connect it first									
3	Select the NFT from collection to Delete									
4	Select Edit item									
5	Scroll down the page and select Delete button		Name: Utilities	Deleted	Deleted	pass				
Post Condition		NFT edited.								

Table 7.13 shows a successful test case to delete an existing NFT from the collection. The prerequisites for deleting an NFT include connecting the wallet to the marketplace.

7.1.8 Transfer NFT:

Table 7.14 Test Case To Transfer NFT

Test Case ID	TC_NFT_014		Designed by	Noureen Butt					
Test Module Name	Transfer NFT			Design Date	5/15/2023				
Test Priority	High			Executed by	Durrez Ahmad				
Test Title	Transfer test			Execution Date	5/30/2023				
Description	Testing Transfer NFT feature.								
Pre-condition	The user is required to connect their external crypto wallet to the marketplace.								
Dependencies	External Crypto Wallet								
Step	Test Step	Test Data	Expected Result	Actual Result	Status	Notes			
1	Open the marketplace								
2	Check if the wallet is connected. If not then connect it first								
3	Select the NFT from collection to transfer								
4	Select transfer								
5	Fill in the required field	Destination address:YYYYYY	Transferred	Transferred	pass				
Post Condition		NFT edited.							

Table 7.14 shows a successful test case to Transfer an existing NFT from the collection to some destination address. The pre-requisites for transferring an NFT include connecting the wallet to the marketplace.

7.2 Decision Table:

7.2.1 Code Snippets/UI: Loading Screen:

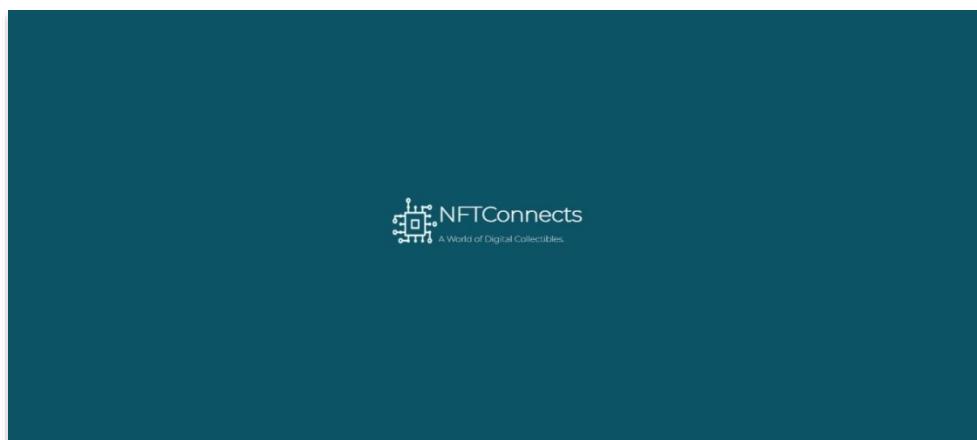


Figure 7.1 Loading

Figure 7.1 shows loading page. Whenever a user comes at the marketplace, loading page is the first interface they come across.

Dashboard:

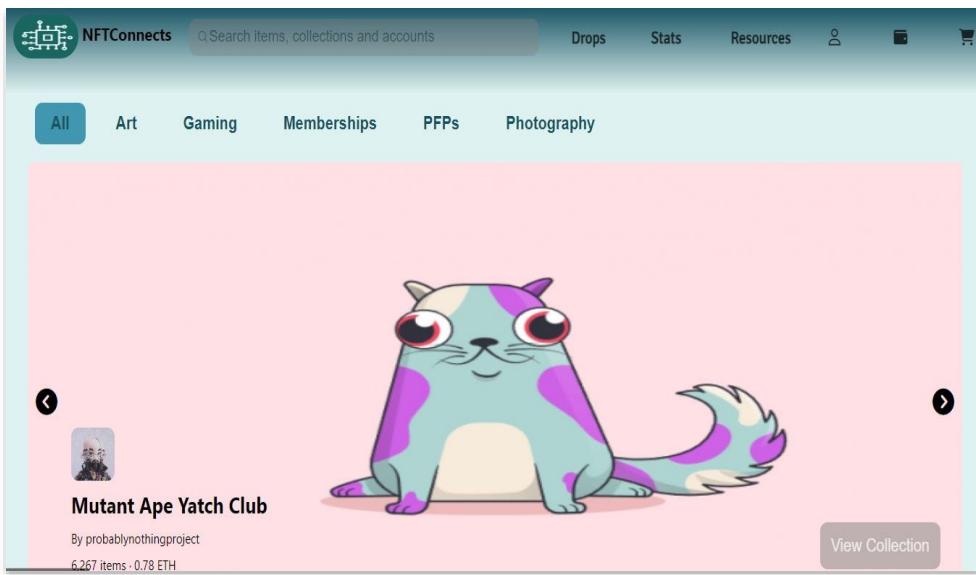


Figure 7.2 Dashboard

Figure 7.2 is the dashboard of NFT Connect. After the loading page, the user come in contact with the dashboard where they can see the NFTs listed for selling and can navigate through the website.

Create collection:

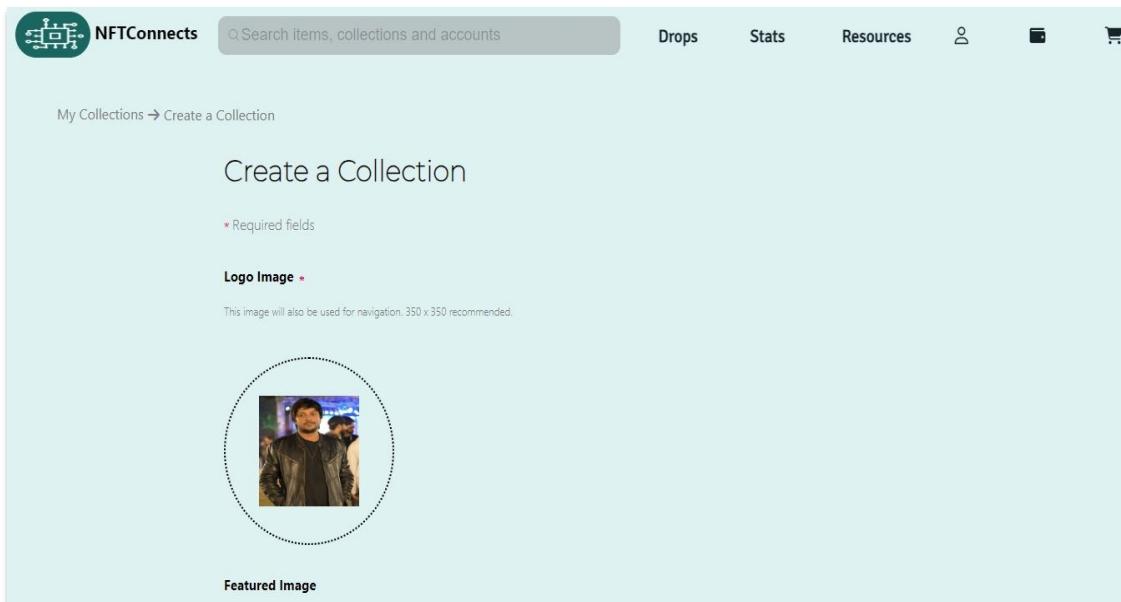


Figure 7.3 Create Collection

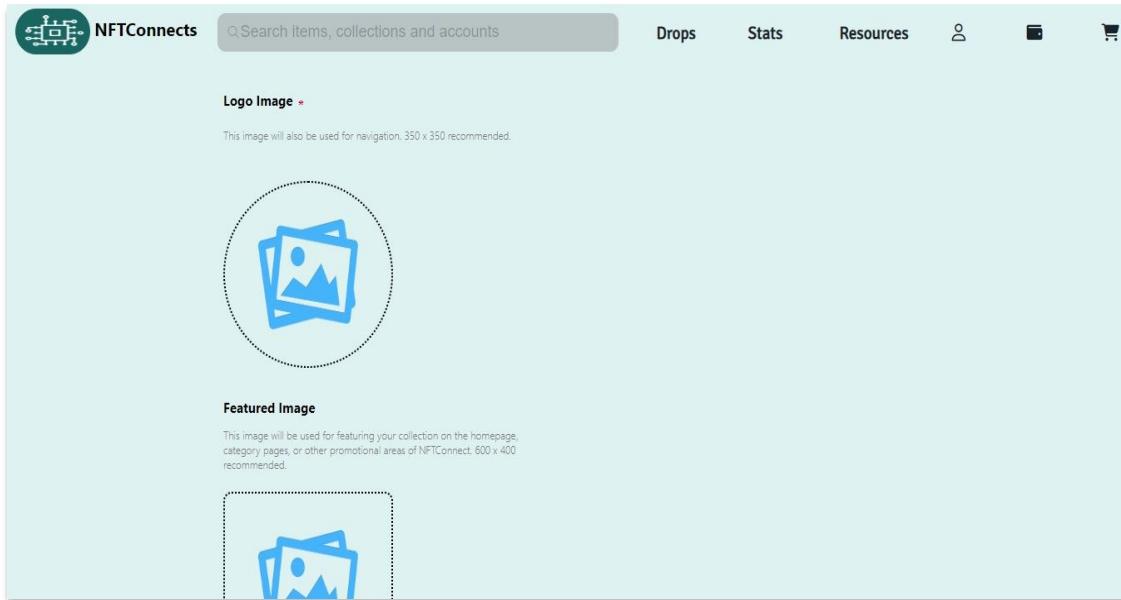


Figure 7.4 Create Collection

Figure 7.3 and 7.5 are the create collection interfaces. In order to create an NFT, the user must create collection first.

My collection:

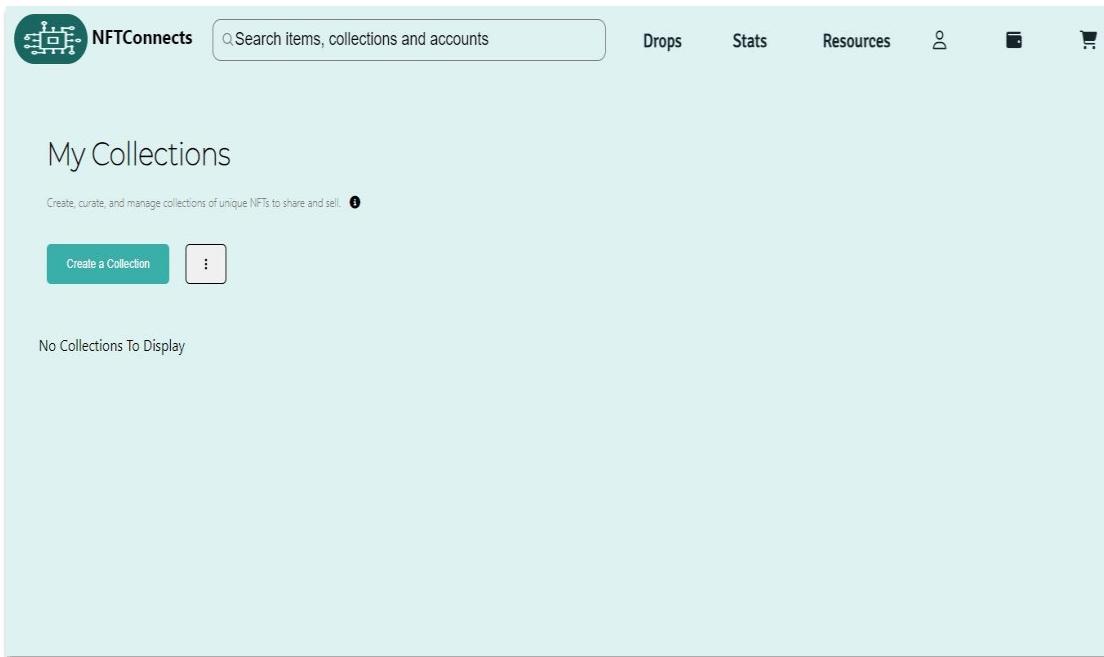


Figure 7.5 My collection before collection created

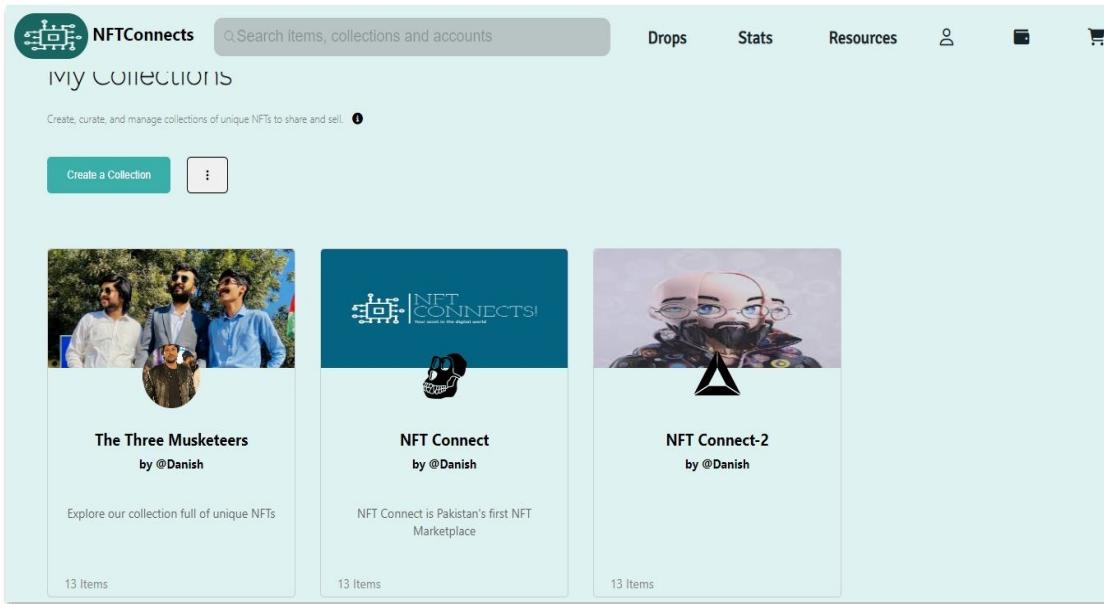


Figure 7.6 My Collection after collection created

Figure 7.5 is the my collection when no collection existed/created. Figure 7.6 is the my collection when the user created a collection.

Create NFT:

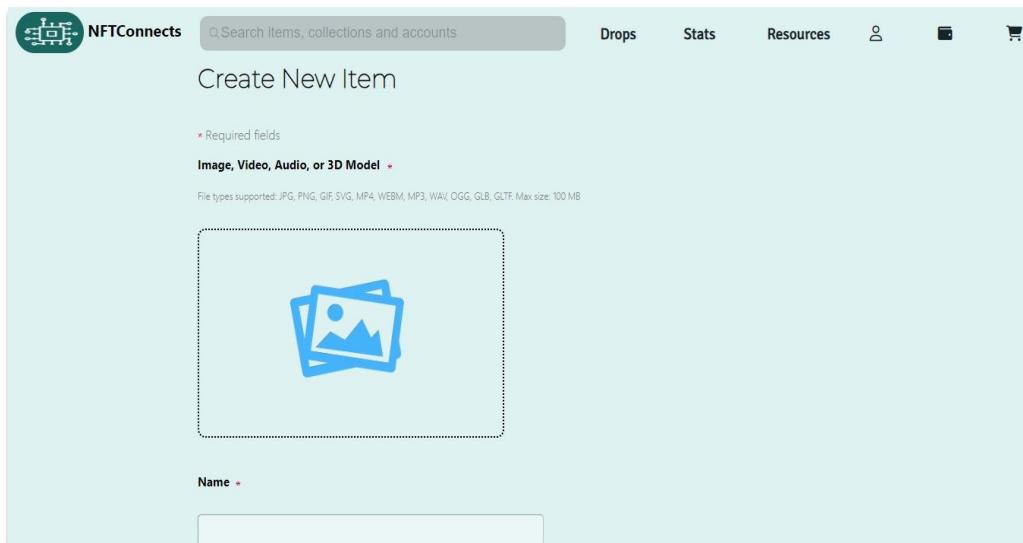


Figure 7.7 Create NFT

Figure 7.7 is the create NFT page. The user is required to have already created collection to associate their NFT to. The user fills in the required data and then the NFT is created.

7.2.2 Decision Table:

Table 7.15 Decision Table

Conditions	R1	R2	R3	R4	R5	R6	R7	R8	R9
Connect Wallet	F	T	T	T	T	T	T	T	T
Sufficient funds in wallet	F	F	T	F	F	T	T	F	T
Collection	F	F	F	T	T	T	F	F	T
NFT	F	F	F	F	T	F	T	T	T
Actions									
Create Collection	F	T	T	T	T	T	T	T	T
Create NFT	F	F	F	T	T	T	T	T	T
Buy NFT at Auction	F	F	T	F	F	T	T	F	T
Buy NFT at Fixed Rate	F	F	T	F	F	T	T	F	T
Sell NFT at auction	F	F	F	F	T	F	T	T	T
Sell NFT at fixed rate	F	F	F	F	T	F	T	T	T
Edit NFT	F	F	F	F	T	F	T	T	T
Delete NFT	F	F	F	F	T	F	T	T	T
Transfer NFT	F	F	F	F	T	F	T	T	T

Table 7.15 represents Rules and conditions at which the system shows specific behavior. NFT connect follow the above conditions and rules to operate as a marketplace.

7.3 Traceability Matrix:

7.3.1 Requirements vs Use Cases:

Table 7.16 Requirements vs Use Cases Traceability Matrix

Requirements/ Use Case	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
NFT_UC_001	✓	✓	✓						
NFT_UC_002				✓					
NFT_UC_003					✓				
NFT_UC_004						✓			✓
NFT_UC_005						✓			✓
NFT_UC_006						✓			✓
NFT_UC_007								✓	✓
NFT_UC_008					✓				
NFT_UC_009					✓				
NFT_UC_010					✓				
NFT_UC_011							✓		
NFT_UC_012									✓
NFT_UC_013							✓		
NFT_UC_014							✓		
NFT_UC_015					✓				

Table 7.16 shows traceability matrix between requirements of the system and use cases of the system. It can be seen that all the requirements are met by the use cases developed.

7.3.2 Requirements VS Test Cases:

Table 7.17 Requirements vs Test Cases Traceability Matrix

Requirements/ Test Cases	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
TC_NFT_001	✓	✓	✓						
TC_NFT_002	✓	✓	✓						
TC_NFT_003				✓					
TC_NFT_004					✓				
TC_NFT_005					✓				
TC_NFT_006						✓			✓
TC_NFT_007						✓			✓
TC_NFT_008								✓	✓
TC_NFT_009								✓	✓
TC_NFT_0010								✓	✓
TC_NFT_011								✓	✓
TC_NFT_012					✓				✓
TC_NFT_013					✓				
TC_NFT_014						✓		✓	

Table 7.17 shows traceability matrix between requirements of the system and test cases created for the system. It can be seen almost all the requirements and the major features are tested.

7.3.3 Use Cases vs Test Cases:

Table 7.18 Use Cases vs Test Cases Traceability Matrix

Requirements / Test Cases	NFT_UC_001	NFT_UC_002	NFT_UC_003	NFT_UC_004	NFT_UC_005	NFT_UC_006	NFT_UC_007	NFT_UC_008	NFT_UC_009	NFT_UC_010	NFT_UC_011	NFT_UC_012	NFT_UC_013	NFT_UC_014	NFT_UC_015
TC_NFT_001	✓														
TC_NFT_002	✓														
TC_NFT_003		✓													
TC_NFT_004			✓												
TC_NFT_005			✓												
TC_NFT_006				✓		✓									
TC_NFT_007				✓	✓										
TC_NFT_008							✓					✓			
TC_NFT_009							✓					✓			
TC_NFT_0010							✓					✓			
TC_NFT_011							✓					✓			
TC_NFT_012								✓							
TC_NFT_013									✓						
TC_NFT_014										✓					

Table 7.18 shows traceability matrix between use cases of the system and test cases created for the system. It can be seen almost all the use cases and the major features are tested.

7.3.4 Requirements vs Prototypes:

Table 7.19 Requirements vs Prototypes Traceability Matrix

Requirements/ Prototypes	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9
Dashboard							✓		
Create Collection				✓					
Create NFT					✓				
Sell NFT						✓			✓
Buy NFT								✓	✓
Help Center							✓		

Table 7.18 shows the traceability matrix to trace the work relation between the prototypes and the requirements of the system i.e. NFT Connect.

8. Results

8.1 Completeness:

- The functional requirements FR1, FR2 and FR3 are fulfilled by the use case id NFT_UC_001. The said use case requires the user to create an external wallet account and then connect it to the marketplace. The functional requirement FR1 and FR2 are completed by the creation of external wallet account that support both cryptocurrency and NFT. On the other hand, connecting the wallet to the marketplace fulfills the FR2 and FR3 requirement i.e. authenticating wallet.
- The functional requirement FR4 is satisfied by the use case NFT_UC_002. It states that the user can create their collection at the marketplace but the pre-requisite include connecting the wallet to the marketplace.
- Functional requirement FR5 states that creating an NFT is required at NFT Connect. The use case NFT_UC_003 fulfills the requirement of creating an NFT. For creating NFT at marketplace, the user is required to have a crypto wallet connected to the marketplace and a collection must exist before creating NFT. If the user has not created collection then they need to first create collection and then they will create NFT to the corresponding collection.
- Functional requirement FR6 is that the user should be allowed to sell their NFT. Use case NFT_UC_004, NFT_UC_005 and NFT_UC_006 states that the user at NFT Connect is allowed to sell their NFTs. NFT_UC_004 says that user is allowed to sell their NFT in two ways i.e. at fixed rate or at auction. NFT_UC_005 demonstrates the steps to sell NFT at auction. On the other hand, NFT_UC_006 demonstrates the procedure to sell NFT at fixed price.
- Functional Requirement FR7 suggests that a dashboard must exist which contains other features of the marketplace along with the NFTs listed for selling. Use case NFT_UC_011, NFT_UC_013 and NFT_UC_014 fulfills the requirement.
- Functional requirement FR8 is about buying NFT. The requirement says that user must be allowed to buy an NFT from the marketplace. The use case NFT_UC_007 meet the requirement. It demonstrates the way to buy an NFT at the marketplace. The pre-requisite for buying an NFT include connecting a crypto wallet to the marketplace and there must be sufficient funds in the wallet.
- Since NFT Connect is a marketplace to deal in NFTs, payment processing is one of its essential requirements. FR9 states that a secure and reliable payment process should be introduced. The use cases NFT_UC_004, NFT_UC_005, NFT_UC_006, NFT_UC_007 AND NFT_UC_0011 are the cases where the payment process occurs.

The completeness of the marketplace is 100% since all the requirements of the system are fulfilled by the use cases.

8.2 Accuracy:

- Test Cases TC_NFT_001 and TC_NFT_002 are designed to test the requirements FR1, FR2 and FR3 i.e. to authenticate the user crypto wallet. The test cases pass the
- The requirement FR4 creating collection is tested by the test case TC_NFT_003. The test case was passed for the said requirement.
- The test cases TC_NFT_004 and TC_NFT_005 are designed for the functional requirement FR5 i.e. create NFT. The test cases pass which means that the use case for creating NFT is designed the way it was required.
- For functional requirement FR6 which is sell NFT the test cases TC_NFT_006 and TC_NFT_007 were designed. Both the test cases pass for the requirement.
- The functional requirement FR7 says that a dashboard should be designed for the ease of the users. No test case was designed for the said requirement.
- The test cases TC_NFT_008, TC_NFT_009, TC_NFT_010 and TC_NFT_011 were designed to test the functional requirement FR8. The test cases TC_NFT_008 and TC_NFT_009 were to test the requirement of buy NFT at fixed rate and both the tests were passed. The test cases TC_NFT_010 and TC_NFT_011 tested the functionality of selling NFT at auction and were a success.
- The test cases TC_NFT_006, TC_NFT_007, TC_NFT_008, TC_NFT_009, TC_NFT_010, TC_NFT_011 and TC_NFT_012 test the FR9.

The accuracy of the system is 98% since all the features of the system operate the way they were required. The 2% goes for dashboard since no test case was designed for FR7. But it was not needed.

8.3 Correctness:

- The use case NFT_UC_001 was to connect crypto wallet to the marketplace. The test cases TC_NFT_001 and TC_NFT_002 were designed to test the said use case. Both the test cases pass for the use case as per requirement.
- The test case TC_NFT_003 was designed to test the use case NFT_UC_002 which was creating the collection. The use case passed the test case.

- The use case NFT_UC_003 was to create NFT at the marketplace. Test cases TC_NFT_004 and TC_NFT_005 were designed to check the correctness of the use case. Both the test cases were passed for the requirement.
- The use case NFT_UC_004 states the ways to sell NFTs at the marketplace. The use case NFT_UC_005 demonstrates the way to sell NFT at auction on the other hand NFT_UC_006 says the way to sell the NFT at fixed price. The test case TC_NFT_007 was designed to test selling at auction and was a success. On the other hand, TC_NFT_006 satisfies the use case for selling NFT at fixed price.
- The use case NFT_UC_007 is to buy an NFT at the marketplace. The test cases TC_NFT_008, TC_NFT_009, TC_NFT_010 and TC_NFT_011 were designed to test the said test case and was a success.
- The test case TC_NFT_012 was designed to test the use case NFT_UC_008 i.e. edit NFT. The test case was passed.
- The use case NFT_UC_009 states the procedure to delete an NFT. The test case TC_NFT_013 was designed to test the requirement and was a success.
- The use case NFT_UC_010 demonstrates the transferring NFT at marketplace. The requirement was test using TC_NFT_014 and was passed.
- The use case NFT_UC_012 was adding the item (NFT) to cart. The test cases TC_NFT_008, TC_NFT_009, TC_NFT_010 and TC_NFT_011 includes the procedure to test the use case.
- No test cases were designed to test the use cases NFT_UC_011, NFT_UC_013, NFT_UC_014 and NFT_UC_015.

Since all the designed test cases pass for the use cases as required the correctness of the system developed is 100%. There are some use cases for which tests are not designed. So their correctness is not known.

9. Conclusion

The NFT industry is a fast developing field that has recently attracted a lot of attention and interest. New possibilities for digital ownership, provenance, and value exchange have been made possible by non-fungible tokens (NFTs). At the moment, digital art and collectibles dominate the NFT market.

Blockchain technology has brought revolution to world of technology. It serves as an immutable ledger of events where data is replicated over the blockchain network and each block is cryptographically linked to the next block. Tempering the data in blockchain is almost impossible as the hash of the block changes with the change in the data. As we all know that we live in the world of technology and today we need to implement web 3.0 in almost every management system we need to keep data in soft form instead of keeping it in hard form because for any organization their records matters a lot and for any person his/her documents worth a lot. The Problem faced with digital assets was that they were fungible. Anyone could use those assets without authorization. Anyone could copy the digital artwork of any creator and there was no way to find out which piece is original. This platform has provided ease to the creators and buyers.

In conclusion, NFT Connect has provided its users with every key feature needed to ensure the secure transactions of NFTs. NFT Connect, a digital marketplace for NFT transactions. Users of this platform can create, sell and buy the NFTs of their choice. Non-fungible tokens, or NFTs, have been the focus of discussion for a while. Users are further encouraged to think about the uses of NFTs because they have been made available at previously unheard-of rates. NFT Connect is currently helping artists get the respect and attention they have long deserved. NFT holders are assured ownership rights. All the key features are provided at the marketplace to facilitate its users.

10. Future Work

The universe of Non-Fungible Tokens (NFTs) is expected to change and grow in a variety of ways in the future (work). Following are some potential areas where future work will be done:

10.1 Fractional Owners:

It is observed that only one user can own an NFT. Work is being done to introduce fractional owners. It will allow more than one users to own single NFT. This will make it feasible for almost everyone to own high-value assets.

10.2 Improvement in sustainability:

It can be observed that a lot of energy consumption is associated with the blockchain in consensus method i.e. proof-of-work. Future work could focus on developing new methods of consensus to reduce energy consumption.

10.3 An improved user experience:

Since the popularity of NFTs is being increasing day by day, the need of the hour is to introduce user-friendly interfaces. So that any user, willing to deal in NFTs, don't face any problem while using the platform.

10.4 NFTs and AI:

The integration of AI with NFT provides the audience with a unique and dynamic experience. AI is used to generate tokens with the help of algorithms. It can prove to have strong impact as NFTs now can be updated.

10.5 NFTs in metaverse:

Metaverse gives the user a virtual world experience. It requires some avatar to recognize a person in the virtual world. NFTs can be used as avatar. New Tokens are introduced to create copies of NFTs and use them as avatar in the metaverse.

The above are some of the examples of the future work being done on NFTs. There is much more to come out for NFTs in the time ahead.

11. Bibliography

11.1 Articles:

Refrence:

(2018). From Makersplace:

<https://makersplace.com/about/#:~:text=100K%2B%20collectors,creations%20to%20a%20global%20audience>.

(2021, 6). From Binance NFT: <https://buynft.com/marketplaces/binance-nft/>

Conlan Rios, L. L. (2020, 1 1). From Asyn Art: <https://www.finder.com/async-art>

Falin, A. S. (2022, 3). From Rarible: <https://golden.com/wiki/Rarible-MN5B93G>

Foster, D. a. (2018). From Nifty Gateway.

Foster, D. a. (2018). From Nifty Gateway: <https://nymag.com/intelligencer/2020/03/duncan-and-griffin-cock-foster-nifty-gateway-gemini.html>

Sidney Zhang, J. L. (2021, 9 17). From Magic eden:

https://www.google.com/search?q=magic+eden+nft&rlz=1C1GCEB_enPK1017PK1017&oq=magi&aqs=chrome.0.69i59j69i57j0i433i512j0i131i433i512j0i433i512j69i61j69i60l2.1419j0j7&sourceid=chrome&ie=UTF-8

work, F. (n.d.). Future work on NFTs. https://www.cryptopolitan.com/the-future-of-nfts-a-2023-market-outlook/#The_marriage_between_NFTs_and_AI.

12. Appendix:

12.1 Glossary:

NFT:

Non fungible Token. A unique token assigned to digital asset.

FCFS:

First come first serve. It is an approach for buying NFTs at the marketplace at fixed rate.

DFD:

Data flow diagram. It represents the flow of data through the developed system.

ASAP:

As soon as possible.

ERD:

Entity relationship diagram, represents relationship between entities. An entity can be a person or an object or anything that has characteristics performs some task.

IDE:

Integrated Development Environment, any software application that helps the programmers in developing code.

ERC-721:

Ethereum request for comments, a standard implemented for non-fungible tokens.

Auction:

Placing an offer to buy NFT at your desired rate.

JS:

JavaScript, a programming language implemented on client side.

12.2 Pre-Requisites:

12.2.1 NodeJS:

In order to deploy the application on your system, you must need to have NodeJS installed at your pc.

12.2.2 VS-Code:

VS-code is an IDE required to deploy the application.

12.2.3 External Crypto Wallet:

The user willing to use the marketplace is required to have an external crypto wallet account which support both NFTs and crypto currency.

Final Year Project Report

Augmented Reality E-Commerce Mobile App



Project Advisor:

Mr. Amjad Ali

Submitted By:

Ramsha Noor

[F2019065190]

Sundas Sarfraz Cheema

[F2019065037]

Session

Fall 2022 - Spring 2023

**University of Management and Technology
C-II Johar Town Lahore Pakistan**

Dedication

It is dedicated to my parents, my nation, my institution, and the worthy professors who have guided me throughout my life.

Final Approval

- **Head of Department**

Department of Computer Science
& Systems School of Systems & Technology
UMT Lahore

- **Director (Final Year Projects-CS)**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

- **Supervisor**

Department of Computer Science.
School of Systems & Technology
UMT Lahore

Acknowledgment

Even this simple mobile app requires the combined efforts of a huge number of people. When we first started out creating mobile applications, we learned this the hard way. We appreciate everyone's early efforts and continued backing tremendously.

Training helps individuals become ready for their first jobs by instructing them in the practical application of previously learned ideas. Every everyone participating in the production of an app, whether directly or indirectly, must have a same goal.

As young software engineers, we appreciate having the chance to learn from the expertise of the people who helped us create this program. They have earned our deepest gratitude.

To all who have helped us along the way, we are eternally grateful. In order to succeed, one needs the support of his or her whole team and the firm as a whole. From those of us here in the class and the professor who supervised this assignment, we appreciate you taking the time to read this letter.

Project Title:**Augmented Reality E-Commerce Mobile App****Objective:**

The project's ultimate goal is to make it easier for people all around the globe to try on new garments before buying them. We're developing an AR-powered smartphone app that will allow you to visually try on garments in 3D. Because of this, users will have more leeway in putting on garments, and they will be able to test out a wider variety of options in less time

Undertaken by

Ramsha Noor [F2019065190]

Sundas Sarfraz Cheema [F2019065037]

Supervised by

Mr. Amjad Ali

Starting Date

October 3, 2022 (Fall 2022)

Completion Date

May 28, 2023

Tools Used

Android Studio, Google Firebase

Google ML kit, Google Scenform SDK

Google ARCore, Android Smart Phone 8.1 or above

Operating System

Android

Documentation

Software Requirement Specification

Plagiarism Report

Abstract

The fashion accessories business in developing nations suffers from lagging advertising and promotion. Our research examined how Augmented Reality (AR) in low-income apparel bazaars affects customers' perceptions, behaviors, and interactions. This positivist study assumes causal linkages are stable like everything else. Experiential marketing may gather cross-sectional data following a stimulus. Business innovation concepts like uses and gratification and user experience models guide the experimental design for this study (within group). Hedonic, aesthetic, pragmatic, and functional qualities characterize the user experience (utility). After a positive app experience, users are more likely to buy. Augmented reality software competence affected users' perceptions and interactions. This study has several new aspects. The Ray-Ban smart lab was used for digital reflection marketing. Second, producers of augmented reality experiential marketing events considered users' kinesthetic, euphoric, aesthetic, and utilitarian needs. It should work well, be easy to use, look good, feel good, and satisfy the customer's innermost desires. This is the first study of its type in Pakistan, and it examines the effects of augmented reality on consumer proficiency and the following consequences on attitude and pleasure in the fashion accessory industry. This study reveals that familiarity with the application is the most important mediator between attitude and an improved user experience made possible by augmented reality, as opposed to previous research that has concentrated on gender and age. Future researchers may find the proposed final model useful in their own study on fashion brands in both developed and developing countries, and this has important theoretical consequences. Since the findings of the study may be used to enhance the marketing strategy of brands, the participation of brand managers and marketing managers is also of major managerial value.

REVISION CHART

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ramsha Noor Sundas Sarfraz	<p>This is the completed draft of the first capstone project. There are a total of 5 chapters.</p> <ol style="list-style-type: none">1. Introduction2. Domain analysis3. Requirement Analysis4. Data Flow Diagram5. System Design	25-01-2023
2.0	Ramsha Noor Sundas Sarfraz	<p>The 2.0 version is to complete draft Of capstone 2. In these we are complete 7 chapters of document that include:</p> <ol style="list-style-type: none">6. Implementation7. Testing8. Results9. Conclusion10. Future work11. References12. Appendix	28-05-2023

CONTENTS

CONTENTS.....	1
1.1 DEFINITIONS AND ACRONYMS	4
1.2 LIST OF FIGURES	5
1.3 LIST OF MOCKUPS.....	6
1.4 LIST OF TABLES	6
1. INTRODUCTION.....	8
1.1 MOTIVATIONS.....	8
1.1.1 <i>Customer Engagement</i>	9
1.1.2 <i>Product Personalization</i>	9
1.1.3 <i>Improves Customer Satisfaction</i>	9
1.1.4 <i>Augmented Reality- A Standout Feature</i>	9
1.2 PROJECT OVERVIEW	10
1.2.1 <i>What is Augmented Reality</i>	10
1.2.2 <i>Augmented Reality Is Portable</i>	10
1.3 PROBLEM STATEMENT	11
1.4 OBJECTIVES	11
2. DOMAIN ANALYSIS.....	12
2.1 CUSTOMER.....	12
2.2 STAKEHOLDERS	12
2.3 AFFECTED GROUPS WITH SOCIAL OR ECONOMIC IMPACT	12
2.4 REFERENCE DOCUMENTS.....	13
2.4.1 <i>Related Projects</i>	14
2.4.2 <i>Feature Comparison</i>	15
3. REQUIREMENTS ANALYSIS	17
3.1 REQUIREMENTS	17
3.1.1 <i>Non_functional Category Table</i>	17
3.1.2 <i>Functional Requirement Category And Attribute Table</i>	20
3.2 LIST OF ACTORS.....	23
3.3 LIST OF USE CASES	23
3.4 SYSTEM USE CASE DIAGRAM.....	24
3.5 EXTENDED USE CASES.....	25
3.5.1 <i>Sign up</i>	25
3.5.2 <i>Login</i>	26
3.5.3 <i>View and Try products</i>	27
3.5.4 <i>Cart</i>	28
3.5.5 <i>Place order</i>	29
3.5.6 <i>View Order Status</i>	31
3.5.7 <i>View Order History</i>	31
3.5.8 <i>Manage Product</i>	32
3.5.9 <i>Rate Item</i>	34
3.5.10 <i>View Payment</i>	35
3.5.11 <i>Change password</i>	35
3.5.12 <i>Manage order</i>	36
3.5.13 <i>View Feedback</i>	37

3.5.14	<i>Update Profile</i>	38
3.5.15	<i>Logout</i>	39
3.5.16	<i>Update order Status</i>	40
3.5.17	<i>Search Products</i>	41
3.6	USER INTERFACES (MOCK SCREENS)	42
3.6.1	<i>Splash page</i>	42
3.6.2	<i>Sign In</i>	43
3.6.3	<i>Sign up</i>	43
3.6.4	<i>Customer Home page</i>	44
3.6.5	<i>Search Product</i>	44
3.6.6	<i>Try Product</i>	45
3.6.7	<i>Cart</i>	46
3.6.8	<i>Change Password</i>	46
3.6.9	<i>My Account</i>	47
4.	DATA FLOW DIAGRAM	48
4.1	DATA FLOW DIAGRAM LEVEL 0	48
4.2	DATA FLOW DIAGRAM LEVEL 1	49
4.3	DATA FLOW DIAGRAM LEVEL 2	50
5.	SYSTEM DESIGN	51
5.1	SYSTEM ARCHITECTURE DIAGRAM	51
5.2	CLASS DIAGRAM.....	52
5.3	SEQUENCE DIAGRAMS	53
5.3.1	<i>System Sequence diagram</i>	53
5.3.2	<i>Obeject Sequence diagram</i>	54
5.4	COLLABORATION DIAGRAMS.....	55
5.4.1	<i>Customer Collaboration Diagram</i>	55
5.4.2	<i>Admin Collaboration diagram</i>	56
5.5	ACTIVITY DIAGRAM	57
5.5.1	<i>Customer Activity Diagram</i>	57
5.5.2	<i>Admin Activity Diagram</i>	58
5.6	STATE CHART DIAGRAM.....	59
5.7	DATABASE DESIGN	60
5.7.1	<i>Entity Relationship Diagram</i>	60
5.7.2	<i>Entity Relationship Model</i>	61
5.8	DATA DICTIONARY	62
6.	IMPLEMENTATION DETAILS	65
6.1	DEVELOPMENT SETUP.....	65
6.1.1	<i>Development Tools:</i>	65
6.1.2	<i>Programming Languages:</i>	65
6.1.3	<i>Augmented Reality Frameworks:</i>	65
6.1.4	<i>3D Modeling and Design Tools:</i>	65
6.1.5	<i>UI/UX Design Tools:</i>	65
6.1.6	<i>Backend and APIs:</i>	65
6.1.7	<i>Testing and Debugging:</i>	65
6.2	DEPLOYMENT SETUP	66
A.	ALGORITHMS	66
i.	<i>Algorithm for Try shoes in Real Environment</i>	66

ii. <i>Algorithm for Try Clothes in Real Environment</i>	67
B. CONSTRAINTS	68
i. <i>Assumptions</i>	68
ii. <i>System constraints</i>	69
iii. <i>Restrictions</i>	69
iv. <i>Limitations</i>	69
7. TESTING	70
7.1 EXTENDED TEST CASES	70
7.1.1 <i>Sign-Up</i>	70
7.1.2 <i>Login</i>	71
7.1.3 <i>View and Try Product</i>	72
7.1.4 <i>Cart Functionality</i>	73
7.1.5 <i>Place Order</i>	74
7.1.6 <i>View Order Status</i>	76
7.1.7 <i>Manage Product</i>	77
7.1.8 <i>Manage Order</i>	78
7.1.9 <i>Logout</i>	79
7.1.10 <i>Update Order Status</i>	80
7.2 TRACEABILITY MATRIX	81
7.2.1 <i>RID vs UCID (requirements vs use cases)</i>	81
7.2.2 <i>Prototypes (RID vs PID)</i>	82
7.2.3 <i>Test Cases (RID vs TID)</i>	82
8. RESULTS/OUTPUT/STATISTICS.....	84
8.1 COMPLETION	84
8.2 ACCURACY AND CORRECTNESS.....	85
9. CONCLUSION	87
10. FUTURE WORK	88
10.1 BRAND ACQUISITION AND COLLABORATION:.....	88
10.2 PRODUCT CATALOG EXPANSION:.....	88
10.3 BRAND PROMOTION AND MARKETING:.....	88
10.4 SALES AND CONVERSION OPTIMIZATION:.....	88
10.5 PERFORMANCE TRACKING AND ANALYTICS:.....	88
10.6 REVENUE SHARING AND PROFIT GENERATION:.....	88
10.7 CUSTOMER SATISFACTION AND FEEDBACK:.....	89
11. BIBLIOGRAPHY	90
11.1 BOOKS	90
11.2 JOURNALS	90
11.3 ARTICLES	90
11.4 RESEARCH PAPERS	90
12. APPENDIX	91
12.1 GLOSSARY	91

Definitions and Acronyms

Acronym	Definition
UMT	University of Management and Technology
B2C	Business to Consumer
E-Commerce	Electronic Commerce
AR	Augmented Reality
SMA	Sephora Mobile App
NFA	Nike Fit App
HMA	Houzz Mobile App
GUI	Graphic User Interface
FR	Functional Requirements
NFR	Non-Functional Requirements
DFD	Data Flow Diagrams
2-D	Two Dimensions
3-D	Three Dimensions
API	
UC	Use Case

Table 0.1 Definitions and Acronyms

List of Figures

Figure 3.1: Use Case Diagram	24
Figure 4.1: Data Flow Diagram Level 0	48
Figure 4.2: Data Flow Diagram Level 1	49
Figure 4.3: Data Flow Diagram Level 2	50
Figure 5.1: System Architecture Diagram	51
Figure 5.2: System Class Diagram.....	52
Figure 5.3: System Sequence Diagram.....	53
Figure 5.4: System Object Sequence Diagram	54
Figure 5.5: Customer Collaboration Diagram	55
Figure 5.6: Admin Collaboration Diagram.....	56
Figure 5.7: Customer Activity Diagram	57
Figure 5.8: Admin Activity Diagram.....	58
Figure 5.9: System State Chart Diagram	59
Figure 5.10: Entity Relationship Diagram	60
Figure 5.11: Entity Relationship Model.....	61

List of Mockups

Prototype 1: Splash UI	42
Prototype 2: Sign in/Log in.....	43
Prototype 3: Sign up.....	43
Prototype 4: Customer Home Page.....	44
Prototype 5: Search Products	44
Prototype 6: Try Product.....	45
Prototype 7: Customer Shopping Cart	46
Prototype 8 Update Password	46
Prototype 9 Manage Account.....	47

List of Tables

Table 0.1 Definitions and Acronyms	4
Table 2.1 List of Stakeholder	12
Table 2.2: Feature Comparison of Related Paragraph	16
Table 3.1: Non-Functional category Requirement Table	20
Table 3.2: Functional Requirement attribute table	22
Table 3.3: Use Case Of Signup	26
Table 3.4: Use Case of Login	27
Table 3.5: Use Case of Try and View Product	28
Table 3.6: Use Case of Customer Cart.....	29
Table 3.7: Use Case of Place Order	30
Table 3.8: Use Case of Order Status	31
Table 3.9: Use Case of View Order History	32
Table 3.10: Use Case of Manage Product.....	34
Table 3.11: Use Case of Rate Item	35
Table 3.12: Use Case of View Payment	35
Table 3.13: Use Case of Change Password	36
Table 3.14: Use Case of Manage Order.....	37
Table 3.15: Use Case of View Feedback	38

Table 3.16: Use Case of Update Profile.....	39
Table 3.17: Use Case of Logout	40
Table 3.18: Use Case of Update Status.....	41
Table 3.19: Use Case of Search Product.....	42
Table 5.1: Data Dictionary.....	64
Table 7.1: Test Case for Sign-Up	71
Table 7.2: Test Case for Login	72
Table 7.3: Test Case for View and Try Product	73
Table 7.4: Test Case for Cart Functionality.....	74
Table 7.5: Test Case for Place Order.....	76
Table 7.6: Test Case for View Order Status	77
Table 7.7: Test Case for Manage Product.....	78
Table 7.8: Test Case for Manage Order.....	79
Table 7.9: Test Case for Log Out.....	80
Table 7.10: Test Case for Update Order Status	81
Table 7.11: RID vs UCID	82
Table 7.12: RID vs PID	82
Table 7.13: RID vs TID	83

1. INTRODUCTION

E-commerce, is known as electronic commerce or just e-commerce, refers to business dealings conducted entirely through the Internet. Online commerce is the buying and selling of physical goods and services as well as the exchange of digital information between parties through the internet. Doing business through the Internet and the electronic exchange of information, goods, and services among several parties, such as businesses, their customers, and other interested parties, is what is known as e-commerce or electronic commerce.

In contrast, "business-to-consumer e-commerce" (or "B2C e-commerce") describes online exchanges between firms and end users. Numerous companies, like Amazon.com, now engage in online sales. When shopping online, a buyer's computer is linked to a seller's computer, often over the internet.

There is no intermediary service available. Online marketplaces like Amazon.com have made it possible to do business in an entirely digital, interactive, and real-time manner. When a third party facilitates the exchange of goods and services through the Internet, this is known as electronic commerce.

1.1 Motivations

The use of augmented reality has the ability to bridge this gap and reveal previously untapped and uniquely human potential. Although augmented reality is still in its infancy, it has already begun to make inroads into the mainstream, with over \$60 billion expected to be spent on AR technology by 2020. In addition to the academic and nonprofit worlds, augmented reality will impact every industry. It will affect how we learn, make decisions, and interact physically in the next months and years. It will change the way businesses interact with customers, train employees, design and produce products, oversee supply chains, and compete in the marketplace.

Augmented reality's power comes from the way in which information is interpreted by users. There is a distinct way in which each of our five senses gathers data. The human ability to see is unparalleled. Between eighty and ninety percent of what we learn is via sight.

Humans have limited learning and knowledge. "Cognitive load" is mental strain. One-at-a-time thinking reduces multitasking ability.

Any information that's tough to grasp causes mental strain. To read instructions on a computer screen requires more brainpower than hearing them orally. Separating processing from presentation may increase mental strain. Consider the risks of using a smartphone while driving. The driver must maintain the screen's information in working memory and apply it to the road ahead. What you see on television isn't always effective in practice. Physical seclusion requires effort.

AR is useful. Superimposing contextual information and instructions provides a breakthrough GUI. Augmented reality reduces the need for out-of-context 2-D information on pages and displays. AR might increase revenue. Augmented reality doubled conversions, according to Shopify. Consumers may buy and sample anything off the shelf (even their own homes).

AR influences buyer behavior. Improved product-market connection. Positive (virtual) connection enhances consumer loyalty. Anger. How would you characterize winning a branded video game?

Imagine trying on your favorite label's item in a virtual fitting room. Positive brand perception leads to higher sales.

The term "augmented reality" refers to a technology that enables users to superimpose digital data onto genuine locations with the purpose of creating an experience that is more immersive and engaging. When online businesses incorporate augmented reality into their business practices, everyone, including the customers of those online stores, stands to benefit. It establishes a reliable alliance that boosts the overall quality of the customer experience, as well as revenue and visibility levels for both businesses.

We have developed a list of five different ways in which augmented reality is beneficial to online shops.

1.1.1 Customer Engagement

although it yields the greatest returns, experiential marketing has traditionally been the most challenging and expensive form of advertising to implement. Because of AR, all forms of advertising are rapidly becoming interactive experiences. Successful brands are those that actively involve and respond to their target audience, and Augmented Reality provides a hands-free mechanism for doing so.

1.1.2 Product Personalization

Personalization adds to a product's special allure. It's crucial to the smooth operation of any online shop. The retail sector has been increasingly characterized by its emphasis on customized service for customers. Augmented reality allows customers to digitally try on and assess products, and it also helps businesses make the most of customization. Therefore, it's beneficial for making well-informed purchases in a more personalized environment.

1.1.3 Improves Customer Satisfaction

The partnership between E-Commerce software and large companies has resulted in a significant improvement in the level of convenience offered by online shopping. Customers who prefer to make their purchases online may now use augmented reality technology to see the items as if they were a model, giving them the same kind of shopping experience as if they were really in the store. That is wonderful news for the bottom line of the firm as well as its reputation among the happy clients it already has.

1.1.4 Augmented Reality- A Standout Feature

Every single online business, at some point in the process of getting started on the path to becoming an established online merchant, will unavoidably come up against the obstacle of competition. Customers are, without a doubt, spoiled by the options that are accessible to them, and they have more options than they could ever expect to employ all at once; as a result, it may be tough to identify methods to make your company stand out in comparison to the competition that is out there. Augmented reality makes shopping online into more of an experience, which gives online merchants a greater opportunity to differentiate their brand from the brands of their competitors.

Project Overview

Because there is nothing tactile in front of you when you shop online, it may be difficult to get a good feel for a product and determine whether or not it will meet your needs. It is impossible to acquire the same feel for a couch while purchasing it online as you would when purchasing it from a real shop, and the same is true in reverse.

Augmented reality (AR) applications, on the other hand, make it possible for you to provide customers detailed and exhaustive information about your products without needing them to leave the convenience of their own homes.

What is Augmented Reality

Especially when it comes to e-commerce, the retail industry has a variety of prospects for augmented reality. Even if many people have become used to the concept of shopping online, there are still a great number of transactions for which we would want a little more information on the context. This is the case despite the fact that a large number of individuals have become acclimated to the concept. As a direct result of this, it is probable that some categories of products may have a more difficult time being sold over the internet.

Because of this, we developed an augmented reality app that gives customers the ability to try on and examine our 3D models of clothing and accessories on their own

Augmented Reality Is Portable

Virtual reality requires the use of a headset and, in certain cases, the use of hand-held controllers. You must also take steps to prevent harm to the physical world on your account. (You don't want to be that person that punches their TV while doing virtual reality boxing, do you?)

To experience augmented reality, however, you need just your smartphone or, in certain cases, specialized glasses. Due to your heightened state of awareness, you won't need to alter anything about your environment in order to get the most out of the experience.

AR allows us to provide clients with a great deal more information about a product than we are able to do so with a simple still picture or even a video. We can illustrate what a product would appear like in a consumer's area in addition to providing 3D images of it. This decreases the likelihood that the customer would buy the goods only to discover that it does not look as they had expected when they received it.

When compared to the other product presentations, we give successful communication advantages by producing higher novelty, immersion, enjoyment, and utility. This, in turn, results in good attitudes toward the medium and increased purchase intention using augmented reality (AR).

The Use of Augmented Reality Customers may not only virtually try out items when shopping online, but also use realistically modeled user interfaces to virtually browse for them. More customers are drawn into the sales funnel, leading to higher conversion rates, since stores may remove barriers to entry and make all product features available to them.

The line between reality and the online world blurs as a result. It helps people have more positive perceptions of the real world and have more satisfying interactions with it. Its widespread use in the realm of online retail has made consumers' life easier. This facilitates and improves the efficiency of online shopping

Problem Statement

One of the major problems with e-commerce is trying to convey the depth and texture of a physical product in a two-dimensional computer rendering. Augmented reality has the potential to bridge the gap between in-store and online purchasing by simplifying the process of portraying things and giving the buyer with a fuller understanding of the object they're buying.

Internet consumers may now inspect and try out products and services in their own time and location by using augmented reality. Customers are more inclined to buy anything after using augmented reality to preview it.

There is a possibility that conversion rates will decrease as a direct result of worries over this specific consequence. Consumers are wary and want to be certain that they are getting precisely what it is that they want from their purchases. In the event that they go ahead and make the purchase nonetheless, but are ultimately dissatisfied with it, your rate of returns will go higher.

Customers shopping on the internet are able to see and feel how things will operate before making a purchase thanks to augmented reality (AR).

Objectives

Contextual enhancement with augmented reality with the help of e-commerce, we can provide our consumers a really unique purchasing experience.

- Features that eventually will alleviate the issues of online shopping include those that enable consumers to see what clothing and accessories look like on them, that display what sizes would suit your body, or that show the real size of a product in your area.
- Customers may be able to see themselves in all of the clothes instead of seeing models wearing the items at an online shop.
- There will be a new degree of customization that is predicted to boost online business conversion rates and sales.
- The Use of Augmented Reality Customers may not only virtually try out items when shopping online, but also uses realistically modeled user interfaces to virtually browse for them.
- The line between reality and the online world blurs as a result. It helps people have more positive perceptions of the real world and have more satisfying interactions with it.

2. DOMAIN ANALYSIS

2.1 Customer

Anyone may utilize this system since it is public. For companies catering to tech-savvy customer segments, augmented reality experiences that let consumers virtually try on, modify, or envision things like apparel, shoes, beauty, and furniture might revolutionize social commerce. There will be a meteoric rise in sales because customers will be able to visually alter, try on, or otherwise picture the products they are buying.

2.2 Stakeholders

Following are the stakeholders of this app

Stakeholder	Role in System
Admin	The admin of this system can handle all categories and product. He will be adding, deleting and modifying new products and categories and manage whole app.
Users/Customer	The customer/user is a one who can buy a product from our app and use our app as an end user

Table 2.1 List of Stakeholder

2.3 Affected Groups With Social Or Economic Impact

- **Increase Conversion Rate:** Exposure of products to prospective customers is essential to the growth of any business. If augmented reality can be kept up, this may be a viable option for online stores. Customers who see the virtual product as a flat image get the most realistic experience. Customers whose perspectives can be shifted to include the entire breadth of ecommerce will be enticed to make several visits to the shopping website, which will eventually lead to an increase in the conversion rate.
- An increase in Internet use young people and college students make up the **fastest-growing population**.
- **Innovation:** There is no other demographic that makes more use of cutting-edge software and the internet than this one.
- E-commerce, mobile commerce, and digital advertising have all seen significant expansion as a result of the exploding number of people using the Internet.
- The promise of increased efficiency and lower costs in electronic commerce is another draw for would-be business owners to invest in cutting-edge computer technology.
- The market need for e-commerce has increased greatly as a result of the following factors: The rise in scholastic requirements across all levels.
- Changes in online consumer behavior The online shopping experience is always being enhanced to make it more convenient for the customer.
- With augmented reality, online retailers can better educate and interact with their clientele, all the way through and beyond the transaction.

- Millennials and Gen Zers want stores to provide greater opportunities for interaction.
- Superimposition-based augmented reality (AR) may be used by online retailers to show clients how an item of furniture, for instance, would look in their home. To place an object, the user just opens the appropriate website or app and directs the camera to the desired spot. When they point their smartphone in that direction, a digital overlay of the product displays, giving them a better idea of whether or not it would be a good match.
- When a customer clicks on an item from the "product" part of the catalog, they are given the opportunity to see a preview of the object that is both realistic and accurate to its size.
- **Reduce returns:** AR makes it feasible to give customers with far more information on a product than is possible with a simple still photo or even a video. This information may be presented in a variety of formats, including augmented reality. You are able to present, in addition to offering views of a product in 3D, what the product would seem like in the space of a consumer. As a result, there is less of a chance that the consumer would buy the product, get it, and then be dissatisfied when they see that it does not look as they had imagined it would.
- **Capture the attention** of prospective new buyers: in today's noisy environment, the only way to get people to hear what you have to say is to generate some excitement about it. New clients may be won over in this way. Establishing an engaging AR marketing strategy is one option that may be used.
- This is a great way to show off a new product since it gives customers a preview of how it will appear when worn by them. The method is thus very efficient.
- Because of the **one-of-a-kind nature of AR filters**, **they may increase audience engagement** and inspire people to tag you in the content they create, a win-win for all parties involved.
- An element of "**wow**" may help your business stand out from the crowd of competitors on social media by drawing attention to its unique selling points. One reason for this is because doing so highlights your individuality.

2.4 Reference Documents

<https://www.emarketer.com/content/for-some-shoppers-ar-looks-like-a-way-to-avoid-stores>

<https://www.drugstorenews.com/retail-news/ulta-beauty-acquires/>

<https://www.thedrum.com/news/2018/06/18/lor-al-wants-marry-ar-and-personalisation-with-facetime-style-beauty-assistant>

<https://finance.yahoo.com/news/ulta-beauty-invests-technology-keep-010200205.html>

<https://techcrunch.com/2018/03/20/wayfairs-android-app-now-lets-you-shop-for-furniture-using-augmented-reality/>

<https://techcrunch.com/2017/10/24/target-adds-ar-shopping-to-its-mobile-website/>

<https://www.recode.net/2017/11/16/16666662/williams-sonoma-outward-acquisition-ar-augmented-reality-3d-images-pottery-barn-west-elm>

<https://money.cnn.com/2017/11/17/technology/williams-sonoma-outward/index.html>

<https://www.mobilemarketer.com/news/forrester-home-depot-sephora-top-retail-mobile-app-experiences/550040/>

<https://blog.hubspot.com/marketing/augmented-reality-examples>

<https://corporate.homedepot.com/newsroom/5-technologies-changing-how-we-shop>

<https://www.engadget.com/2019/05/09/nike-fit-augmented-reality-right-fit-size-shoes/>

<https://www.emarketer.com/content/virtual-and-augmented-reality-users-2019>

<https://corporate.homedepot.com/newsroom/project-color-app-paint>

2.4.1 Related Projects

Following is the related Projects.

1. Nike Fit App (NFA)

Augmented reality technology was created by Nike and is included into the Nike Fit smartphone app. Customers will be able to ascertain the correct size of shoe they need thanks to this feature. We were able to get an understanding of the inner workings of the management software used by Nike, Inc. by using AppBrain.com.

2. Sephora Mobile App (SMA)

The cutting-edge smartphone app that Sephora has created includes a Virtual Artist. Customers are able to virtually try on things by using face recognition technology, which is provided by The Virtual Artist. Nike, Inc. is Responsible for the Development of the Sephora Mobile App. This application is being developed by the company Live Typing, specifically under the supervision of the project manager "Ilya pomazcov." Livetyping.com provided the vantage point from which the program was evaluated. No corresponding document is available.

3. Houzz Mobile App (HMA)

It's an app for decorating your home. Apple, Inc. is the company responsible for developing the Houzz Mobile App. This application is being developed by the Company Live Typing's Team Lead, Guy Shaviv, who is responsible for overseeing the project. It was discovered on apple.com that the program was there. No corresponding document is available.

2.4.2 Feature Comparison

The following table show the features of different apps in market and how our app is differ from them.

Sr No.	Comparison Feature	NFA	SMA	HMA	Remarks
1	Try product	Users of the Nike app just need to aim the camera on their device at their feet for the software to identify the appropriate size shoe for them. All of this may be accomplished in under a minute by the consumer without leaving the convenience of their own home.	Clients use the app to both post a selfie and scan their face. After that, customers may use the app to virtually test out various colors of cosmetics.	You can easily check whether that purple sofa you've had your eye on will fit in your living room by using the app's View in My Room function. Digitally exploring an object in a space under varied illumination is also possible.	Instead of taking a photo and uploading it to Sephora's "Try product" function to "try on" cosmetics, this algorithm uses a real-time camera to scan the user's face.
2	Online shop	Once a customer's shoe size has been confirmed, the information is saved inside the app. By scanning a QR code created by the shopper's smartphone camera, sales associates may quickly and simply access the customer's shoe size via the app.	If a user loves what they see in the virtual mirror, they can simply make a purchase inside the app.	There is no chance to purchase test out Furniture using the smartphone app.	Using the feature of "online shop" from Nike and Houzz improving it by adding shop online of tried product so customer feel eases their life by using app

3	Products and categories	Foot print and size of foot are both provided by Nike via their app, allowing customers to purchase products in-store.	Sephora provides wide range of perfumes and cosmetics/makeup products to try out and buy online	Houzz app just provide home décor products such as couch, paintings, dinning accessories, and etc.	Taking the "Product and Categories" features from each of these apps, we can create one comprehensive fashion, clothes, home decor, and shoe app. Several companies have partnered with this app to offer their products for sale with a "try it before you buy it" option. This means that rather than having to download a separate app for each product category, users just need to learn how to navigate one platform.
---	-------------------------	--	---	--	---

Table 2.2: Feature Comparison of Related Paragraph

3. REQUIREMENTS ANALYSIS

3.1 Requirements

Following table show the categories of non-functional requirements.

3.1.1 Non_functional Category Table

To assure the overall quality and efficacy of a system, the Non-functional Category Table offers a thorough overview of non-functional needs, including performance, security, scalability, and usability.

Category	Description
Non-Functional Requirements	<p>Confidentiality</p> <p>NFR_1: Unauthorized users should be denied access to all data. During transmission, it must not be sabotaged in any way.</p> <p>Integrity</p> <p>NFR_2: All data shall be sent securely without any modifications.</p> <p>Availability</p> <p>NFR_3: Within a certain timeframe, information shall be accessible wherever it is needed.</p> <p>Authenticity</p> <p>NFR_4: A user's identity shall be verified before they are granted access to sensitive data.</p> <p>Encryption</p> <p>NFR_5: Only the intended recipient should be able to decode the information.</p> <p>Auditability</p> <p>NFR_6: It is important that data shall be captured in a fashion that allows for an audit to be performed to ensure that it meets all integrity standards.</p> <p>Navigation and structure:</p> <p>NFR_7: Customers shall be able to easily navigate to the content they need, browse available options, and complete their purchases with a minimal number of mouse clicks.</p> <p>Readability of Text:</p> <p>NFR_8: Clients need sight to read product descriptions.</p> <p>Availability:</p> <p>NFR_9: The system must be available 24/7.</p>

	<p>Performance:</p> <p>NFR_10: Logging in should take less than one second total. Aside from the aforementioned, the whole site should load in under a minute on the glass. Under SLA, you're obligated to handle up to 25 requests for pages per second.</p> <p>Security</p> <p>NFR_11: The system shall lock a user's account after four consecutive successful logins attempts within a period of five minutes.</p> <p>Maintainability</p> <p>NFR_12: The function of module shall not be greater than 15 lines.</p> <p>Portability</p> <p>NFR_13: The system shall support minimum android version 8.0 or above</p> <p>Reusability</p> <p>NFR_14: The Registration module shall be designed in a way that it can be used in other applications in the future.</p> <p>Compatibility</p> <p>NFR_15: The application shall be compatible with android smart phones</p>
Data Requirements	<p>DR_1: All information must be stored in Google Firebase (Realtime database).</p> <p>DR_2: A customer's password shall never be seen in the application's user interface. Special characters denoting textual characters will always be used to repeat this.</p> <p>DR_3: A customer's credit card number shall never be shown in the application after being retrieved from the database. The last four digits of the credit card number are required to be shown at all times.</p> <p>DR_4: Only authorized users, such as system administrators, shall be able to access the system's back-end servers.</p> <p>DR_5: Encryption of system databases is required.</p>
Constraints	<p>Const_1: Memory Constraints</p> <p>Const_1.1: The system shall have at least 200 MB in internal storage of mobile to install app.</p> <p>Const_2: Hardware Constraints</p>

	<p>Const_2.1: The user shall have active camera in their android phone to try product.</p> <p>Const_2.2: The hardware interface for the system will consist only of the hardware necessary to connect to the internet, since the program must be deployed online. Example: a modem, a wide area network to a local area network, or a cross-cable for an Ethernet network.</p> <p>Const_3: Software Constraints</p> <p>Const_3.1: The user shall have android phone with 8.0 version or above of android operating system.</p> <p>Const_3.2: A general knowledge of basic mobile skills shall require to use the product</p> <p>Const_4: Operational Constraints</p> <p>Const_4.1: The system shall automatically log out all customers after a customer install updated version of app from play store.</p> <p>Const_5: User Interface Constraints</p> <p>Const_5.1: It will take a 1 hours of training to user to understand the application</p> <p>Const_5.2: The system shall require a maximum of 0.5 seconds for loading and logging into the system</p>
External Interface requirements	<p>Software Interface</p> <p>SI_R_1: The database will be consulted by the online shop software in order to determine the various product types that may be offered.</p> <p>SI_R_2: In order to get product details, availability, and sales, the online shop must get in touch with the administration.</p> <p>Hardware Interfaces</p> <p>HI_R_1: The hardware interface for the system will consist only of the hardware necessary to connect to the internet, since the program</p>

	<p>must be deployed online. Example: a modem, a wide area network to a local area network, or a cross-cable for an Ethernet network,</p> <p>User Interfaces</p> <p>UI_R_1: Any Android phone running Android 8.0, Android 10, or Android 12 should be able to utilize the software's user interface, so users may use the camera to test out items.</p> <p>Communication Interface</p> <p>CI_R_1: For online interaction, the system will make use of the Google API.</p>
--	---

Table 3.1: Non-Functional category Requirement Table

3.1.2 Functional Requirement Category And Attribute Table

Functional Requirement Category and Attribute Table classifies functional needs according to their traits and qualities. It aids in compiling and organizing the needs for clear stakeholder communication and comprehension.

RID	Description	Category	Attribute	Details & Boundary Constraints
FR_1.0	admin shall login into system by using his credentials and	functional	Verification	Admin shall not be able to log in to system after entering wrong credentials
FR_1.1	Customers shall login in system after successfully registration	functional	Verification	Customer shall not be able to log in to system after entering wrong credentials
FR_1.2	Customers shall be complete its registration for the very first time.	functional	Application Registration page	User shall view Signup page with details and save required field in database correctly
FR_1.3	Admin shall add 3D products, their description and details.	functional	Save in db.	After admin adding products and details the information successfully save on database at same time. Customer shall not be able to add products.
FR_1.4	Admin shall delete 3D products, their	functional	Delete from db.	After admin deleting products and details from system the

	description and details.			information successfully deletes on database at same time. Customer shall not be able to delete products.
FR_1.5	Admin shall update 3D products, their description and details.	functional	update db.	After admin updating products and details the information successfully update on database at same time. Customer shall not be able to update products.
FR_1.6	Customer shall view 3D products, their description and details.	functional	Load products from database	Customer shall view 3D products, their description and details on their home page which are load from database in 3 sec.
FR_1.7	Admin shall manage orders requests.	functional	Order status	Admin shall update order status so customer view their order status
FR_1.8	Admin shall be removed, cancel and complete order	functional	Manage order	
FR_1.9	The customer shall login with verified email and password	functional	Authentication	
FR_2.1	Admin shall be able to view feedback from customer.	functional	Visibility	The feedback of product given by user shall be visible to admin
FR_2.2	Admin shall be able to view all payments.	functional	Maintainability	The system shall maintain the record of all order payment for admin
FR_2.3	Customers shall be changing his/her password.	functional	Secured	The customer can change his/her password once in a week with OTP.
FR_2.4	Customers shall be able to view all categories available to shop.	functional	View Category	The products must be lying under some categories.
FR_2.5	Customers shall be able to try our products online with help of augmented reality.	functional	Try product	Whenever customer click to try product. The phone camera shall open with product in real space.

FR_2.6	Customers shall be able to add products to cart which they want to buy.	functional	Add product into cart	The cart shall be automatically saved in database with respective customer id
	Customers shall be able to view products in cart which they want to buy.	functional	Manage cart	System shall support only one cart for user at a time
FR_2.7	Customer shall be able to place order after adding products to carts.	functional	Checkout	Customer shall place only one order at a time
FR_2.8	Customer shall be able to view his/her order status. Is its pending, completed or dispatched	functional	Order status	The order status updated by admin shall visible to user.
FR_3.0	Customer shall be able to view history of their all orders.	functional	Order History	All completed order with respected id must be maintain on database
FR_3.1	Customer shall be able to change his information at any time.	functional	Manage Profile	The updated information successfully updated on database.
FR_3.2	Customer and admin shall be able to log out to app any time.	functional	Logout	After logout from system both admin and user need their credential to again log into system.
FR3.3	System shall be able to calculate shipping fee according to area	functional	Shipment fee	The system shall calculate shipping fee according to requirement
FR_3.4	Customer shall be able enter required details to complete their order	functional	Shipment details	The details enter by user is saved in database with order details. So, admin complete the order.

Table 3.2: Functional Requirement attribute table

3.2 List of Actors

There are two actors of this Application

Admin: Admin can manage application by managing orders, customers, products and **categories**.

Customer: Customer is the one tries products online in real environment and who can buy product.

3.3 List of use cases

Following are the use cases of this system

- **Login:** admin login into system by using his credentials and Customers can login in system after successfully registration
- **Sign-Up:** Customers should be complete its registration for the very first time.
- **Manage Products:** Admin can add, delete, update, 3D products and their description and details.
- **Manage order:** admin manage orders requests.
- **View Feedback:** Admin can view feedback from customer.
- **Manage payments:** Admin manage all payments.
- **Change Password:** Customers can be changing its password any time.
- **View 3D products:** Customers can view all 3D products available to shop.
- **Try Products:** Customers can try our products online with help of augmented reality.
- **Cart:** Customers can add products to cart which they want to buy.
- **Place Order:** Customer can be place order after adding products to carts.
- **View Order Status:** Customer can view his/her order status. Is its pending or dispatched.
- **View All Categories:** Customer can view all categories of products. For Example, women wear, men's wear, shoes and accessories.
- **Rate Items:** Customer can rate and give feedback about our products.
- **Order History:** Customer can view history of their all orders.
- **Update Profile:** Customer should be able to change his information at any time.
- **Log-Out:** Customer and admin should be able to log out to app any time.

3.4 System use case diagram

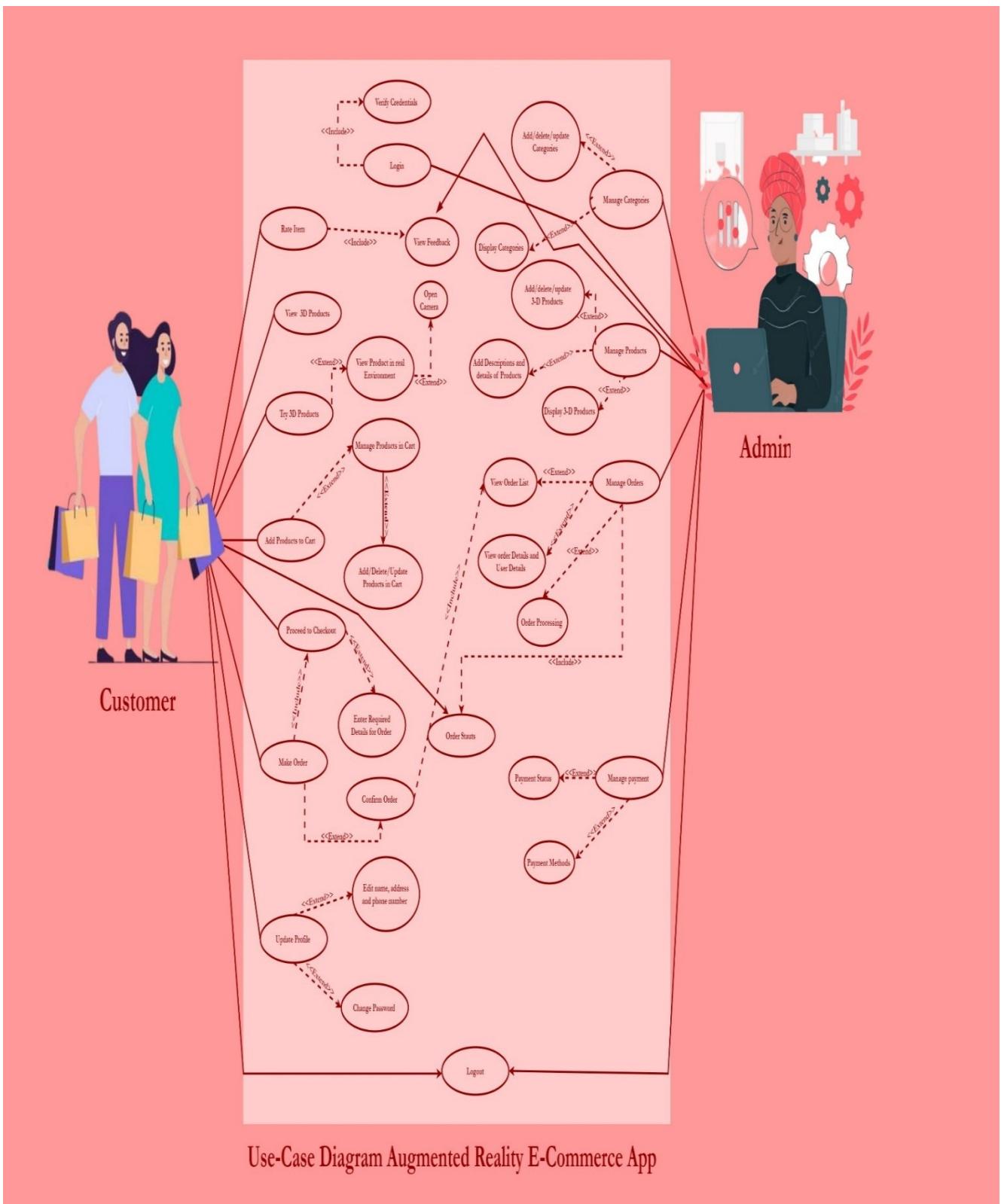


Figure 3.1: Use Case Diagram

3.5 Extended use cases

Here is the all extend use cases of app, that include Login, Sig-Up, Shopping Cart, Try Products, Order and others.

3.5.1 Sign up

Use Case ID:	UC_01		
Use Case Name:	Sign up		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	Customer sign up into system with their required credential.		
Trigger:	Customer open app to sign up		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. Customer installed the app from google play store. 2. Customer not have account in system. 		
Post conditions:	<p>Following is the postcondition.</p> <ol style="list-style-type: none"> 1. Customer successfully registers into system. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Open mobile app. 2. Click on “register” button. 3. Enter name, email, password and all required details. 4. Enter password again to confirm it. 5. Enter captcha. 6. Customer click to register button. 7. Customer successfully create account. 		
Alternative Flows:	<p>4a. The alternative flow is start from step 4 of normal flow. If user not enter same password in confirm password field.</p> <ol style="list-style-type: none"> 1. If customer not enter same password in confirm password field, the system show message “the confirm password is not matched with password”. 2. Customer enter same password. 3. Flow resume to step 4 of normal flow <p>5a. If customer not enter captcha.</p> <ol style="list-style-type: none"> 1. If customer not enter the captcha, the system show message “verify you’re not a robot”. 2. Customer enter captcha. 3. The flow will resume from step 5 of normal flow. 		
Exceptions:	<p>3a. In step 3 of the normal flow, if the customer enters credentials like email or name and it match from another user.</p> <ol style="list-style-type: none"> 1. System will show pop up message to customer with message “you have account with these credentials please log in”. 		

	2. The flow resume to step 3 of normal flow.
Includes:	None
Frequency of Use:	500 user per day
Special Requirements:	<ol style="list-style-type: none"> The user's password must never be shown in plain text anywhere in the application's UI. Textual characters are always indicated by special characters. The user's credentials must not be duplicated in the system.
Assumptions:	The Customer shall understand English.
Notes and Issues:	<ol style="list-style-type: none"> The customer enter must be email in format of abc@example.com The password must have at least 1 special character, 1 capital letter, alphanumeric and at least have 8 characters.

Table 3.3: Use Case Of Signup

3.5.2 Login

Use Case ID:	UC 02		
Use Case Name:	login		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Admin and Customer		
Description:	Customer and admin login into system with their verified credential.		
Trigger:	Customer and admin open app to login		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> Both actors installed the app from google play store. Admin must have active account. Customer must register in system. 		
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> Customer successfully login into system and view customer home page. Admin successfully login into system and view Admin home page. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> Open mobile app. Click on login. Enter email and password. Click on login button. System validates the actor have account. A successfully login message shown on screen. Actors view their respective home page. 		

Alternative Flows:	<p>5a. The alternative flow is start from step 5 of normal flow. If user not have account</p> <ol style="list-style-type: none"> 1. If customer not have account the system show message “Please register first”. 2. Customer click to new account button. 3. System creates new account successfully. 4. Customer enter email and password and flow resume to step 3 of normal flow <p>5b. If customer not create new account.</p> <ol style="list-style-type: none"> 1. the system show message “Please register first”. 2. Customer not create new account 3. Login failed. 4. System redirected to login page.
Exceptions:	<p>3a. In step 3 of the normal flow, if the customer enters and invalid credentials.</p> <ol style="list-style-type: none"> 1. System will show pop up message to actors with message “Please Enter Right Credentials”. 2. Actors enter right credentials 3. The flow resume to step 5 of normal flow.
Includes:	UC_01: Sign up
Frequency of Use:	500 user per day
Special Requirements:	<ol style="list-style-type: none"> 1. The user's password must never be shown in plain text anywhere in the application's UI. Textual characters are always indicated by special characters. 2. The system shall log in the customer till the next update.
Assumptions:	The Customer shall understand English.
Notes and Issues:	<ol style="list-style-type: none"> 1. The customer enter must be email in format of <u>abc@example.com</u> 2. The password must have at least 1 special character, 1 capital letter, alphanumeric and at least have 8 characters.

Table 3.4: Use Case of Login

3.5.3 View and Try products

Use Case ID:	UC_03		
Use Case Name:	View and Try products		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	Customer view and try product from different categories after successfully login in to system		
Trigger:	Customer open app		
Preconditions:	Following is the precondition.		

	<ol style="list-style-type: none"> Customer must login in system.
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> Customer successfully view customer home page. Customer try products (clothes, watches, glasses) by opening their camera.
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> Open mobile app. Go to home page. Select category for specific product. Click on product 3D image. System shows a pop-up permission message to “allow camera access”. Customer allow access to app to open a camera. Customer visually try product by open their camera in real environment.
Alternative Flows:	<p>5a. The alternative flow is start from step 5 of normal flow. If user not allowed system to open camera</p> <ol style="list-style-type: none"> If customer not allow camera access to app. The system won't open product in real environment to try.
Exceptions:	Null
Includes:	None
Frequency of Use:	Customer can try 1 product at a time.
Special Requirements:	<ol style="list-style-type: none"> The customer's android phone shall have camera and The customer shall have android version at least 8.1 to try product. Customer shall have stable internet connection.
Assumptions:	The Customer shall understand English.
Notes and Issues:	The customer shall have android version at least 8.1 to try products.

Table 3.5: Use Case of Try and View Product

3.5.4 Cart

Use Case ID:	UC 04		
Use Case Name:	Cart		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	Customer add products to cart to buy them.		
Trigger:	Customer click on “add to cart” button right below product image.		

Preconditions:	Following is the precondition. 1. Customer select specific product from specific category.
Post conditions:	Following is the post condition. 1. Customer successfully add product of related category in cart. 2. The pop-up message shows with text “item successfully added to cart”.
Normal Flow:	The normal flow is: 1. Click on “view” button to view product. 2. Select quantity of product want to buy. 3. Click “add to cart” button. 4. The pop-up message shows with text “item successfully added to cart”. 5. Customer add & delete product from cart any time.
Alternative Flows:	2a. The alternative flow is start from step 2 of normal flow. If user not select quantity of product. 1. If customer click “add to cart” button without selecting quantity of product. 2. the system pop-up message shown to customer with text “Please enter quantity”. 3. Customer add quantity. 4. flow resume to step 3 of normal flow
Exceptions:	2a. In step 2 of the normal flow, if the customer enters quantity and that product is out of stock. 2. System will show pop up message to customer with message “the selected product is out of stock”. 3. The system re directed customer to the product page.
Includes:	UC_03: view and try products
Frequency of Use:	1 cart for 1 user.
Special Requirements:	Following are the special requirements. 1. The customers shall be able to change his/her cart any time. 2. Customer shall add more than one item in cart.
Assumptions:	The Customer shall understand English.
Notes and Issues:	1. Customer shall have only one cart at time 2. Customer cart products is auto saved in card against his /her user id.

Table 3.6: Use Case of Customer Cart

3.5.5 Place order

Use Case ID:	UC_05		
Use Case Name:	Place order		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		

Description:	Customer place order of product that they added into cart.
Trigger:	Customer click on button “Checkout” from cart page.
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. Customer must have products in cart. 2. The total bill of all product with shipment fee is calculated and display on customer screen.
Post conditions:	<p>Following is the postcondition.</p> <ol style="list-style-type: none"> 1. Customer successfully login into system and view customer home page. 2. Admin successfully login into system and view Admin home page.
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Click on cart button. 2. View total bill. 3. Click on “checkout” button. 4. Enter shipping address and details on detail page. 5. Select payment methods. 6. Enter payment address (if different). 7. Click on “Confirm Order Button”. 8. System start countdown of 30 sec to cancel order. 9. If customer not cancel the order is placed. 10. The pop-up message with order id shown and which confirmed order.
Alternative Flows:	<p>8a. The alternative flow is start from step 8 of normal flow. If user cancel the order.</p> <ol style="list-style-type: none"> 1. If customer click to “cancel” button with in 30 sec the order is cancelled. 2. The order details Is not saved on database. 3. System re direct customer to “cart” page.
Exceptions:	<p>3a. In step 3 of the normal flow, if the customer has empty cart.</p> <ol style="list-style-type: none"> 1. After clicking checkout button system validated the cart products. 2. If user have empty cart (not have any product in cart). 3. The system pop-up the message with text “you cart is empty, please add some product to make order”. 4. If customer add products to cart the flow resume to step 2 of normal flow.
Includes:	UC_12: Manage order
Frequency of Use:	1 order at 1 time
Special Requirements:	<p>Following are special requirements.</p> <ol style="list-style-type: none"> 1. The customer’s order must be saved against the user id.
Assumptions:	The Customer shall understand English.
Notes and Issues:	None

Table 3.7: Use Case of Place Order

3.5.6 View Order Status

Use Case ID:	UC_06		
Use Case Name:	View order status		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	Customer view his/her order status after placing an order.		
Trigger:	Customer click the status button from my account		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> Customer shall have processing order in queue. 		
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> Customer view his/her order's current status. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> Click on my account icon. Go to order status tab View order status 		
Alternative Flows:	<p>2a. The alternative flow is start from step 2 of normal flow. If user not have any processing order.</p> <ol style="list-style-type: none"> If customer not have any processing order the message shown "there is no pending order" 		
Exceptions:	None		
Includes:	UC_12: Manage order		
Frequency of Use:	None		
Special Requirements:	None		
Assumptions:	The Customer shall understand English.		
Notes and Issues:	None		

Table 3.8: Use Case of Order Status

3.5.7 View Order History

Use Case ID:	UC_07		
Use Case Name:	View order History		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023

Actors:	Customer
Description:	Customer view his/her order history of completed order.
Trigger:	Customer and admin open app to login
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> Customer shall have at least one completed order.
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> Customer view his/her order's history.
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> Click on account icon. Go to order history tab View order history Customer view all completed order.
Alternative Flows:	<p>2a. The alternative flow is start from step 2 of normal flow. If user not have any completed order.</p> <ol style="list-style-type: none"> If customer not have any completed order the message shown "there is no order"
Exceptions:	None
Includes:	None
Frequency of Use:	None
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	None

Table 3.9: Use Case of View Order History

3.5.8 Manage Product

Use Case ID:	UC_08		
Use Case Name:	Manage Product		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Admin		
Description:	Admin add, update, delete and display product of related category.		
Trigger:	Admin open manage product tab.		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> Admin shall be logged in the system with his/her credentials. 		

	<p>2. Some products must already exist, either in the system or in supplementary materials (in case of modification or deletion).</p>
Post conditions:	<p>Following is the postcondition.</p> <ol style="list-style-type: none"> 1. Admin shall add product successfully with description and details. 2. Admin shall update product successfully with description and details. 3. Admin shall delete product successfully with description and details 4. Admin shall display product successfully with description and details
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Admin click on manage product button. <ul style="list-style-type: none"> A. To add products. 1. Click on add product button. 2. Select product image from internal storage. 3. Add name. 4. Add description. 5. Add price. 6. Select category. 7. Click on save button to add. B. To update previous products. 1. Click on update product tab. 2. Enter product name to search. 3. the details open to relative product name. 4. admin select product which want to update. 5. The relative field is filled with previous text and details. 6. The admin enters relative data (want to update). 7. Click on save button to save. C. To delete products. 1. Click on delete product tab. 2. Enter product name to search. 3. the details open to relative product name. 4. admin click on delete button. 5. Product successfully deleted from database
Alternative Flows:	<p>3a. The alternative flow is start from step 3 from B of normal flow. If admin enter product name.</p> <ol style="list-style-type: none"> 1. If admin enter such name which are not a part of system, the message shown “No Product Found or Recheck your Spelling”. <p>2a. The alternative flow is start from step 2 from A of normal flow. If admin miss any required field.</p> <ol style="list-style-type: none"> 1. If admin not enter any of required field, the system show message “please enter required field”.
Exceptions:	Unfortunately, a malfunction in the database prevents the modifications from being permanently stored at this time.

Includes:	None
Frequency of Use:	None
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	Adding, deleting and updating operations cannot be undone.

Table 3.10: Use Case of Manage Product

3.5.9 Rate Item

Use Case ID:	UC 09		
Use Case Name:	Rate item		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	A customer rate in to comment on a recent purchase or the purchasing procedure.		
Trigger:	The customer wants to share their thoughts about a service or product.		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. The customer must be logged in. 2. the consumer must have placed at least one order that he or she is willing to rate. 		
Post conditions:	<p>Following is the postcondition.</p> <ol style="list-style-type: none"> 1. The system saves and forwards feedback for further processing. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. The customer provide rating by clicking/adding stars. 2. The customer click on “send” button to save the feedback. 		
Alternative Flows:	<p>2a. The alternative flow is start from step 2 of normal flow.</p> <ol style="list-style-type: none"> 1. If user not have any completed order, instead of sending, it can be cancelled. 		
Exceptions:	A database error prevents the system from storing the user comments.		
Includes:	UC_13: View Feedback		
Frequency of Use:	None		
Special Requirements:	None		

Assumptions:	The Customer shall understand English.
Notes and Issues:	The feedback will not be available by the customer, only by the administrator

Table 3.11: Use Case of Rate Item

3.5.10 View Payment

Use Case ID:	UC_10		
Use Case Name:	View Payment		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Admin		
Description:	Admin view all the payment that received from completed orders.		
Trigger:	The customer payment that they pay for their order		
Preconditions:	None.		
Post conditions:	Following is the postcondition. 1. Admin view all details successfully.		
Normal Flow:	The normal flow is: 1. Click on “manage payment”. 2. View all payments.		
Alternative Flows:	None		
Exceptions:	There is no payment list due to bad internet or database failure.		
Includes:	None		
Frequency of Use:	None		
Special Requirements:	None		
Assumptions:	The Customer shall understand English.		
Notes and Issues:	None		

Table 3.12: Use Case of View Payment

3.5.11 Change password

Use Case ID:	UC_11
Use Case Name:	Change Password

Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer		
Description:	Customer change his/her password.		
Trigger:	Customer wants more secure password		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. Customer shall have logged in in the system. 		
Post conditions:	<p>Following is the postcondition</p> <ol style="list-style-type: none"> 1. Customer changed his/her password successfully. 2. Customer automatically logout from all devices 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Click on my account icon. 2. Go to privacy tab. 3. Click change password option. 4. Enter current password. 5. Enter new password. 6. Confirm new password. 7. Click save button. 8. OTP generate to verified mobile number. 9. Verify OTP. 10. Confirmation message pop-up. 		
Alternative Flows:	<p>2a. The alternative flow is start from step 2 of normal flow. If user forget their current password.</p> <ol style="list-style-type: none"> 1. If user forget his/her current password the system asks to forget password by generating reset password link. 		
Exceptions:	OTP is not generated		
Includes:	None		
Frequency of Use:	None		
Special Requirements:	None		
Assumptions:	The Customer shall understand English.		
Notes and Issues:	None		

Table 3.13: Use Case of Change Password

3.5.12 Manage order

Use Case ID:	UC 12
Use Case Name:	Manage Order

Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Admin		
Description:	Admin manage order by dispatching, or completing order		
Trigger:	Admin process the order		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. Pending order must be in queue from customer. 		
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> 1. Admin dispatched order successfully. 2. Admin successfully complete order. 3. Admin successfully cancel order 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Admin click to “order” tab. 2. View all pending order. 3. Ready order. 4. Dispatched it 5. Update order status. 		
Alternative Flows:	<p>2a. The alternative flow is start from step 2 of normal flow. If Admin not have any order.</p> <ol style="list-style-type: none"> 1. If there is no pending order in list. 2. The system show message “there is no pending order in list”. 		
Exceptions:	None		
Includes:	UC_05: Place order		
Frequency of Use:	None		
Special Requirements:	None		
Assumptions:	The Customer shall understand English.		
Notes and Issues:	None		

Table 3.14: Use Case of Manage Order

3.5.13 View Feedback

Use Case ID:	UC_14		
Use Case Name:	View feedback		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023

Actors:	Admin
Description:	The admin view feedbacks from customers.
Trigger:	Admin wants to open feedbacks given by the customers.
Preconditions:	Following is the precondition. 1. Admin must be logged in.
Post conditions:	Following is the post condition. 1. Admin view feedback successfully.
Normal Flow:	The normal flow is: 1. the admin opens the "Process Feedbacks" dialog. 2. He/she reads the feedbacks
Alternative Flows:	None
Exceptions:	None
Includes:	UC_09: Rate item
Frequency of Use:	None
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	None

Table 3.15: Use Case of View Feedback

3.5.14 Update Profile

Use Case ID:	UC_15		
Use Case Name:	Update Profile		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer and admin		
Description:	Admin and Customer edit his/her personal information any time.		
Trigger:			
Preconditions:	Following is the precondition. 1. Customer shall have register account.		
Post conditions:	Following is the post condition. 1. Customer's profile has been either created or updated		

Normal Flow:	The normal flow is: 1. Users selects "Edit Profile" 2. System displays categories of profile (personal, preferences). 3. Users selects category which he/she want to change 4. App open detail 5. Users' updates detail, click to "save" button 6. System validates data as required, updates customer profile.
Alternative Flows:	None
Exceptions:	The actor's information matches a customer profile. The use case will proceed once the system asks the actor to confirm their profile.
Includes:	None
Frequency of Use:	None
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	None

Table 3.16: Use Case of Update Profile

3.5.15 Logout

Use Case ID:	UC_16		
Use Case Name:	Logout		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Customer, Admin		
Description:	The actors used this to terminated their sessions.		
Trigger:	Actor is doing this by using the app		
Preconditions:	Following is the precondition. 1. The user is logged in 2. The user no longer wants to be logged in		
Post conditions:	Following is the post condition. The user is logged out.		
Normal Flow:	The normal flow is: 1. The user clicks on the logout button 2. The app logs the user out 3. The app redirects to the default login page.		

Alternative Flows:	None
Exceptions:	None
Includes:	None
Frequency of Use:	Whenever the user wants to log out.
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	None

Table 3.17: Use Case of Logout

3.5.16 Update order Status

Use Case ID:	UC_17		
Use Case Name:	Update order status		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	Admin		
Description:	The admin updates the status of pending orders.		
Trigger:	The status of an order needs manual interference.		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. The admin shall be logged in. 2. There is must be any pending order in list 		
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> 1. New order status will be saved in the system. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 2. Admin click on "Update Order Status" dialog. 3. He/she selects the order which needs to be updated from the list. 4. The new status to be set is chosen from drop down menu. 5. Admin saves the changes. 		
Alternative Flows:	None		
Exceptions:	The system cannot save the new states due to a database failure.		
Includes:	UC_06: view order status		

Frequency of Use:	None
Special Requirements:	None
Assumptions:	The Customer shall understand English.
Notes and Issues:	Changes can't be undone.

Table 3.18: Use Case of Update Status

3.5.17 Search Products

Use Case ID:	UC_18		
Use Case Name:	Search product		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	28-12-2022	Last Revision Date:	19-05-2023
Actors:	customer		
Description:	Browse and/or search among products (clothes, glasses, shoes and etc.).		
Trigger:	The customer wants to search product.		
Preconditions:	<p>Following is the precondition.</p> <ol style="list-style-type: none"> 1. The customer opens an app on his/her phone 		
Post conditions:	<p>Following is the post condition.</p> <ol style="list-style-type: none"> 1. The system presents a result list of products (which meets with the given search conditions in case of searching) to the customer. 		
Normal Flow:	<p>The normal flow is:</p> <ol style="list-style-type: none"> 1. Customer opens app on phone. 2. The consumer enters a search term and pushes "OK." 3. The system detects goods that match the search criteria and shows the consumer. 4. Customer chooses a product. 5. Customer hits "Details." 6. A popup window displays product details. 		
Alternative Flows:	None		
Exceptions:	<ol style="list-style-type: none"> 1. The search terms are wrong. 2. The system cannot load products(database unavailable). 		
Includes:	None		
Frequency of Use:	None		
Special Requirements:	None		

Assumptions:	The Customer shall understand English.
Notes and Issues:	If none of the products meets the given search conditions, the system returns an empty list.

Table 3.19: Use Case of Search Product.

3.6 User interfaces (mock screens)

3.6.1 Splash page



Prototype 1: Splash UI

3.6.2 Sign In

← **Sign In**

Please fill E-mail & password to login your Shopy application account.

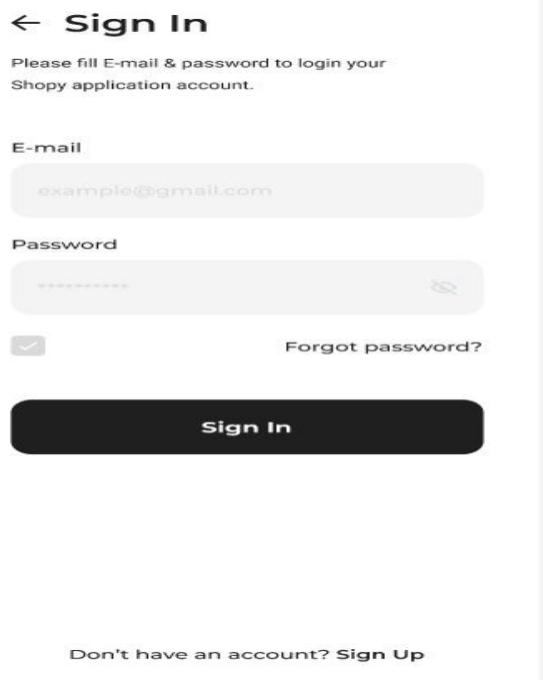
E-mail

Password

[Forgot password?](#)

Sign In

Don't have an account? [Sign Up](#)



Prototype 2: Sign in/Log in

3.6.3 Sign up

← **Sign Up**

Name

E-mail

Password

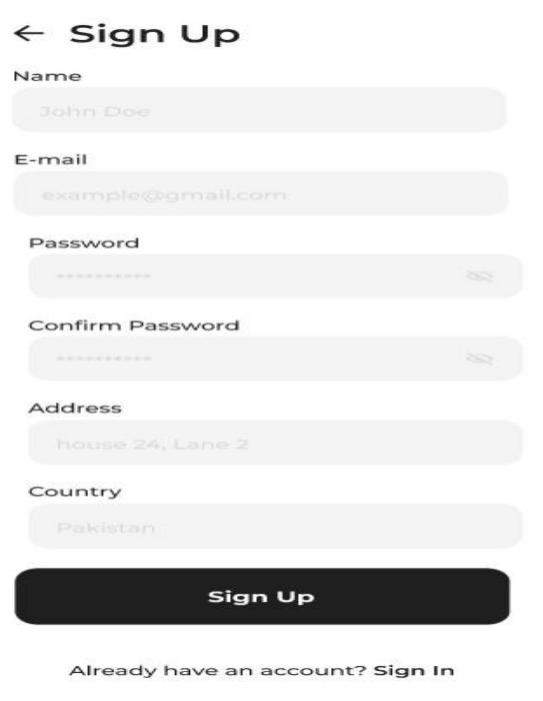
Confirm Password

Address

Country

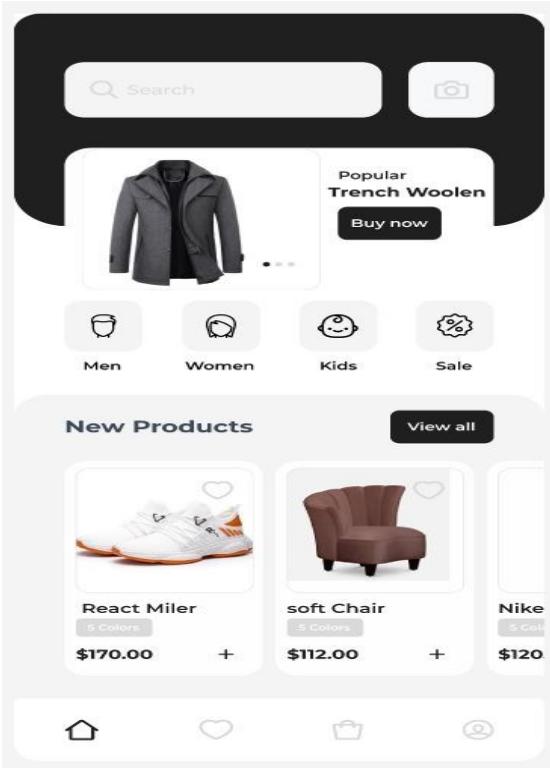
Sign Up

Already have an account? [Sign In](#)



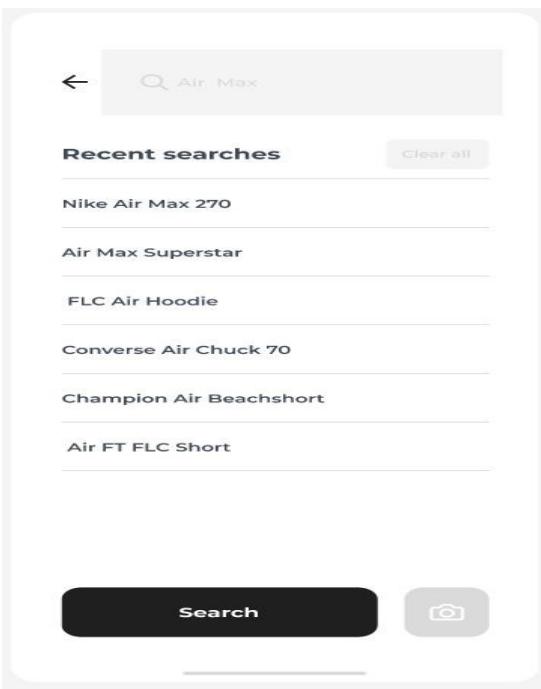
Prototype 3: Sign up

3.6.4 Customer Home page



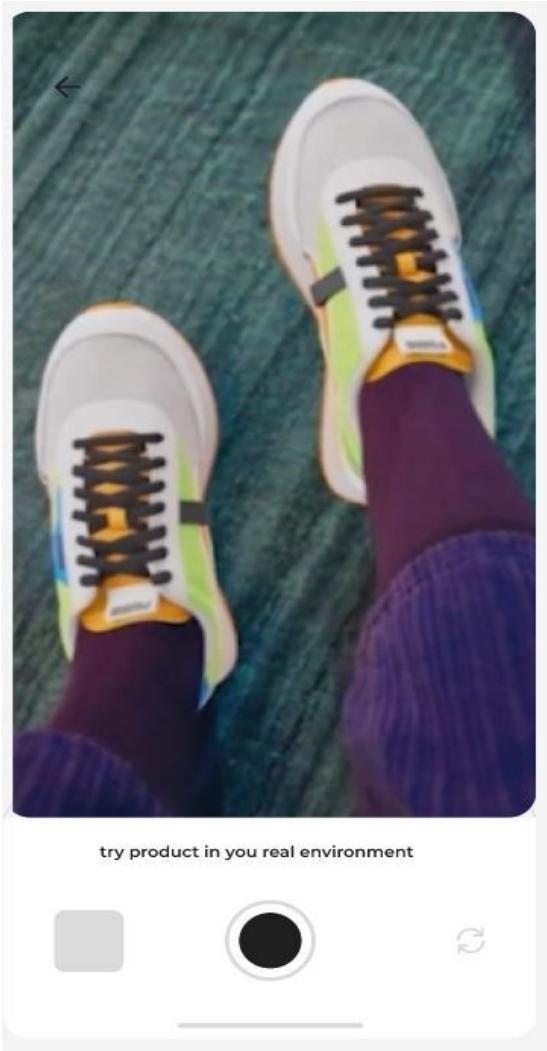
Prototype 4: Customer Home Page

3.6.5 Search Product



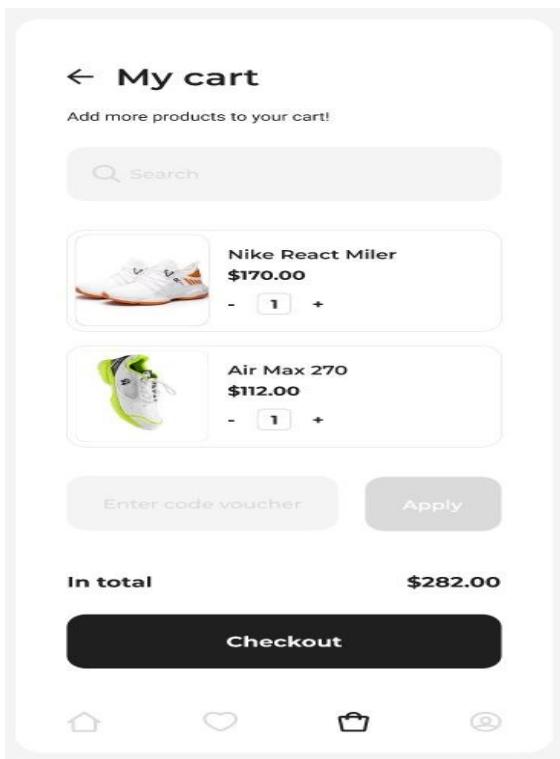
Prototype 5: Search Products

3.6.6 Try Product



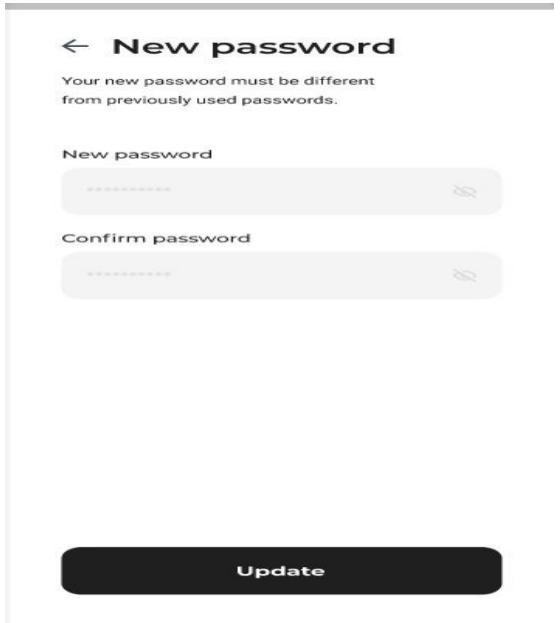
Prototype 6: Try Product

3.6.7 Cart



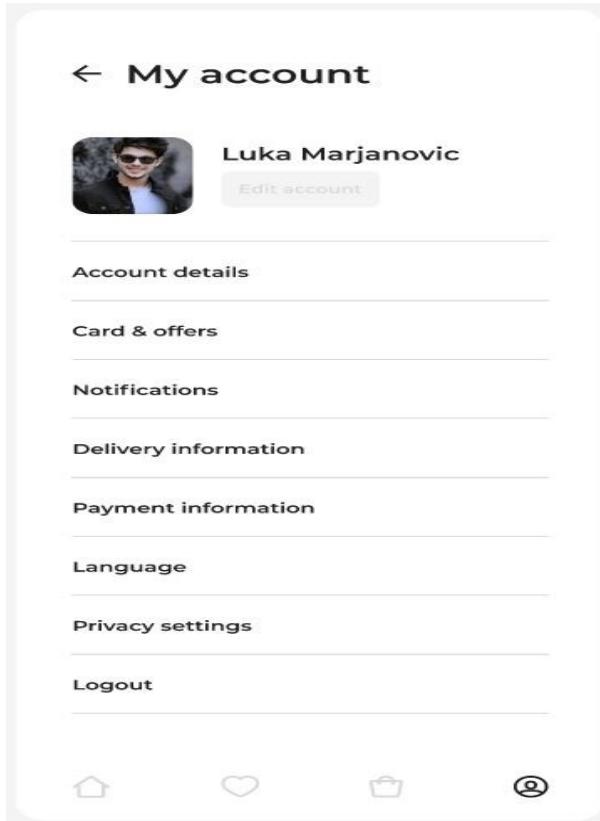
Prototype 7: Customer Shopping Cart

3.6.8 Change Password



Prototype 8: Update Password

3.6.9 My Account



Prototype 9: Manage Account

4. DATA FLOW DIAGRAM

4.1 Data Flow Diagram Level 0

The simplified Level 0 data flow diagram for an AR e-commerce app shows the information flow between external entities (users, customers), the system's fundamental components (inventory management, order processing), and the overall system's high-level activities.

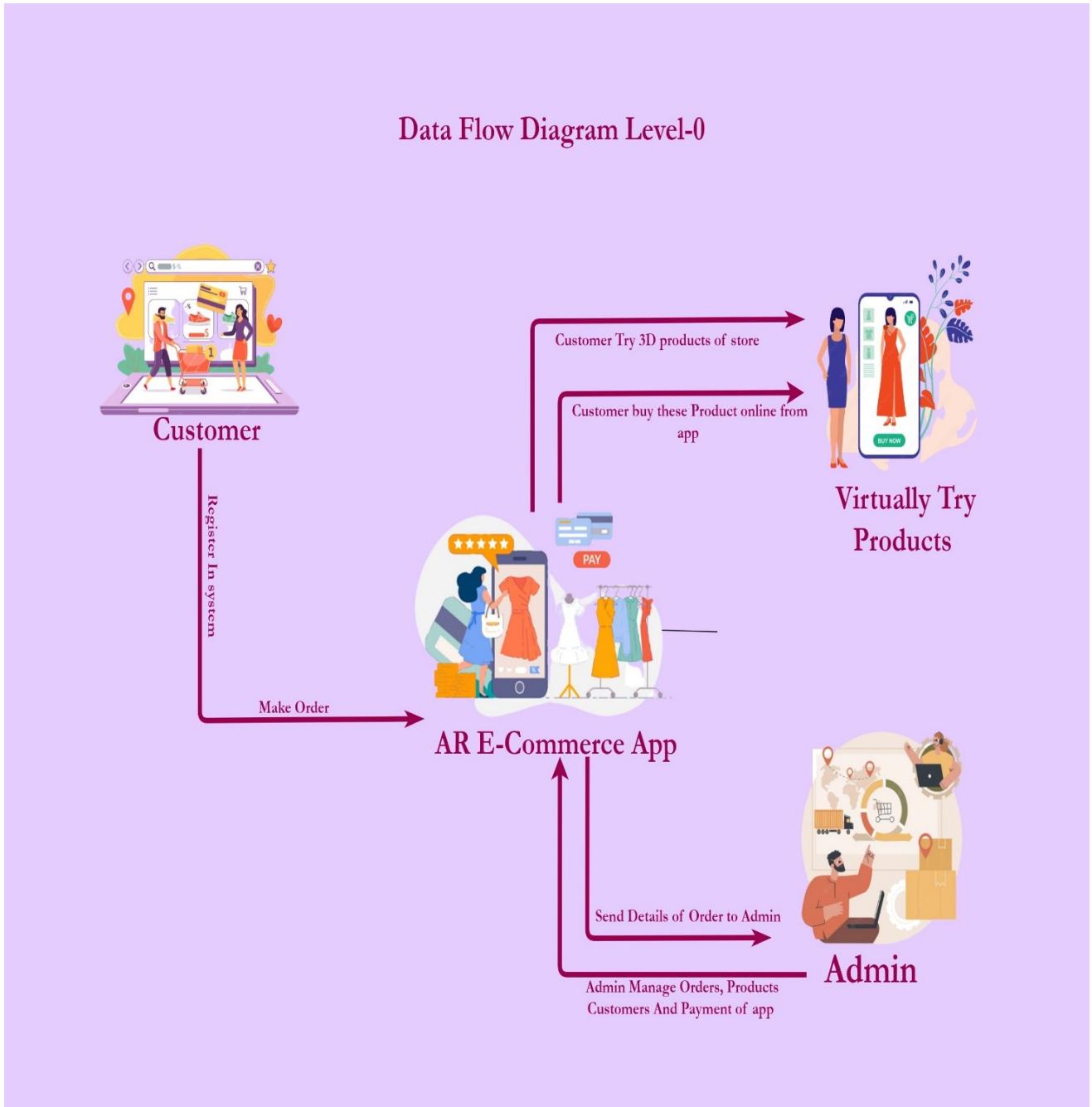


Figure 4.1: Data Flow Diagram Level 0

4.2 Data Flow Diagram Level 1

The Level 1 data flow diagram (DFD) of an e-commerce app illustrates the flow of information between the user, the system, and external entities during the process of placing an order and completing a transaction.

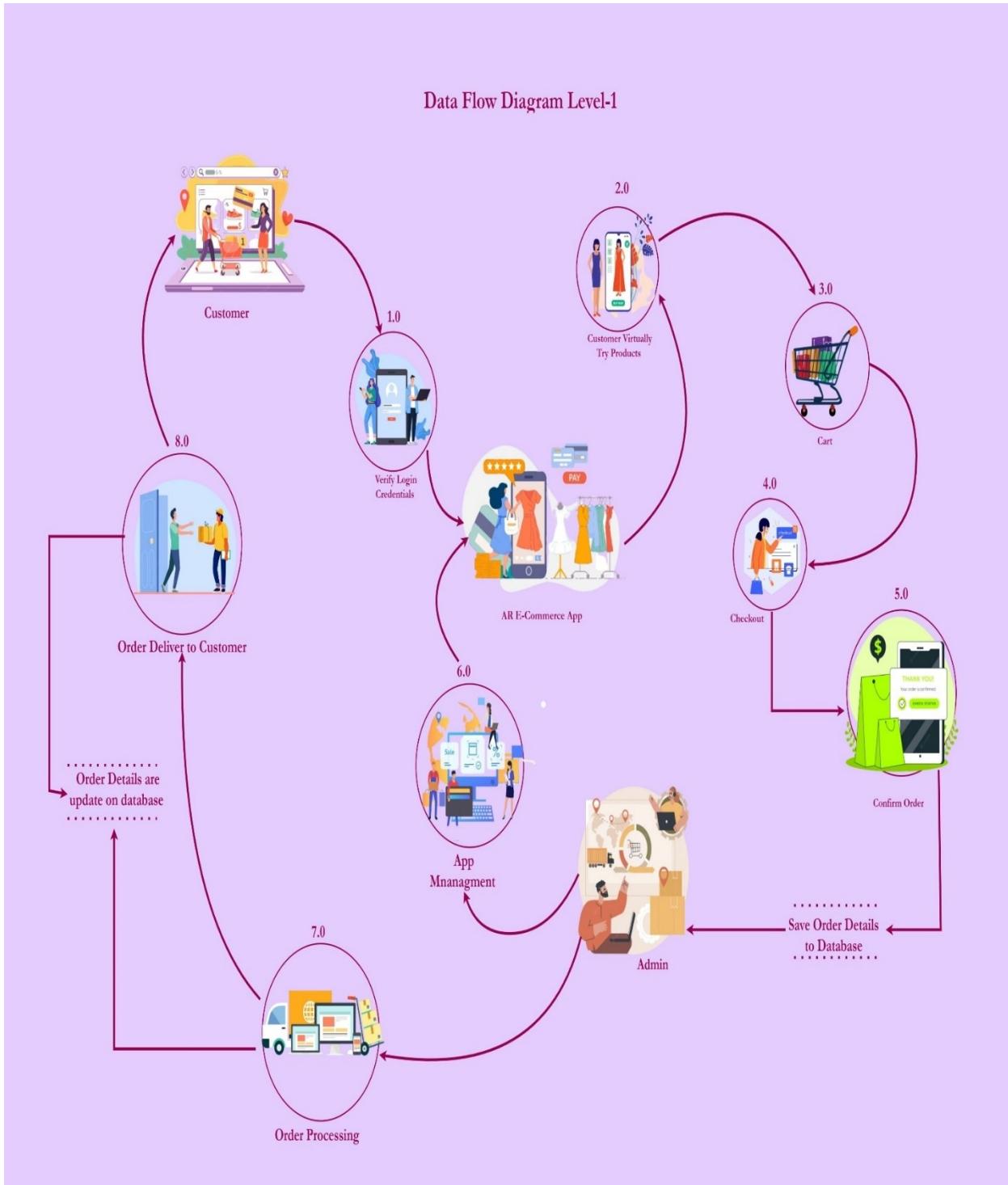


Figure 4.2: Data Flow Diagram Level 1

4.3 Data Flow Diagram Level 2

The Level 2 Data Flow Diagram (DFD) of an e-commerce app illustrates the flow of information how customers select products, update cart, and complete orders while integrating with backend systems.

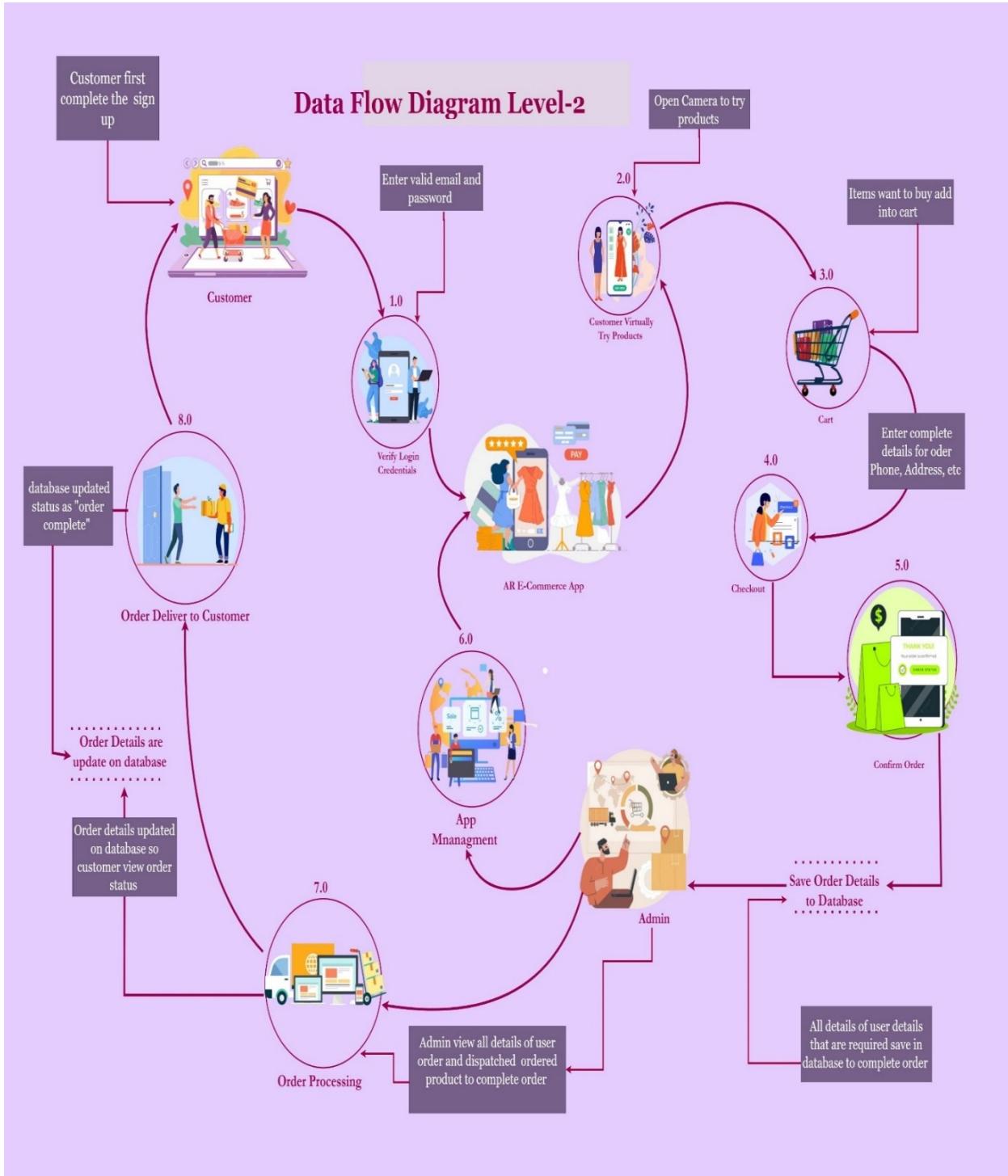


Figure 4.3: Data Flow Diagram Level 2

5. SYSTEM DESIGN

The system design diagram of the AR e-commerce app showcases the architecture and components involved in the app's functioning, illustrating how different modules such as user interface, server, database, and external services interact to provide a seamless online shopping experience.

5.1 System Architecture Diagram

The client-side applications, web servers, application servers, databases, and external services are just a few of the modules that interact to provide users with seamless online shopping experiences, as shown in the system architecture diagram of the e-commerce app.

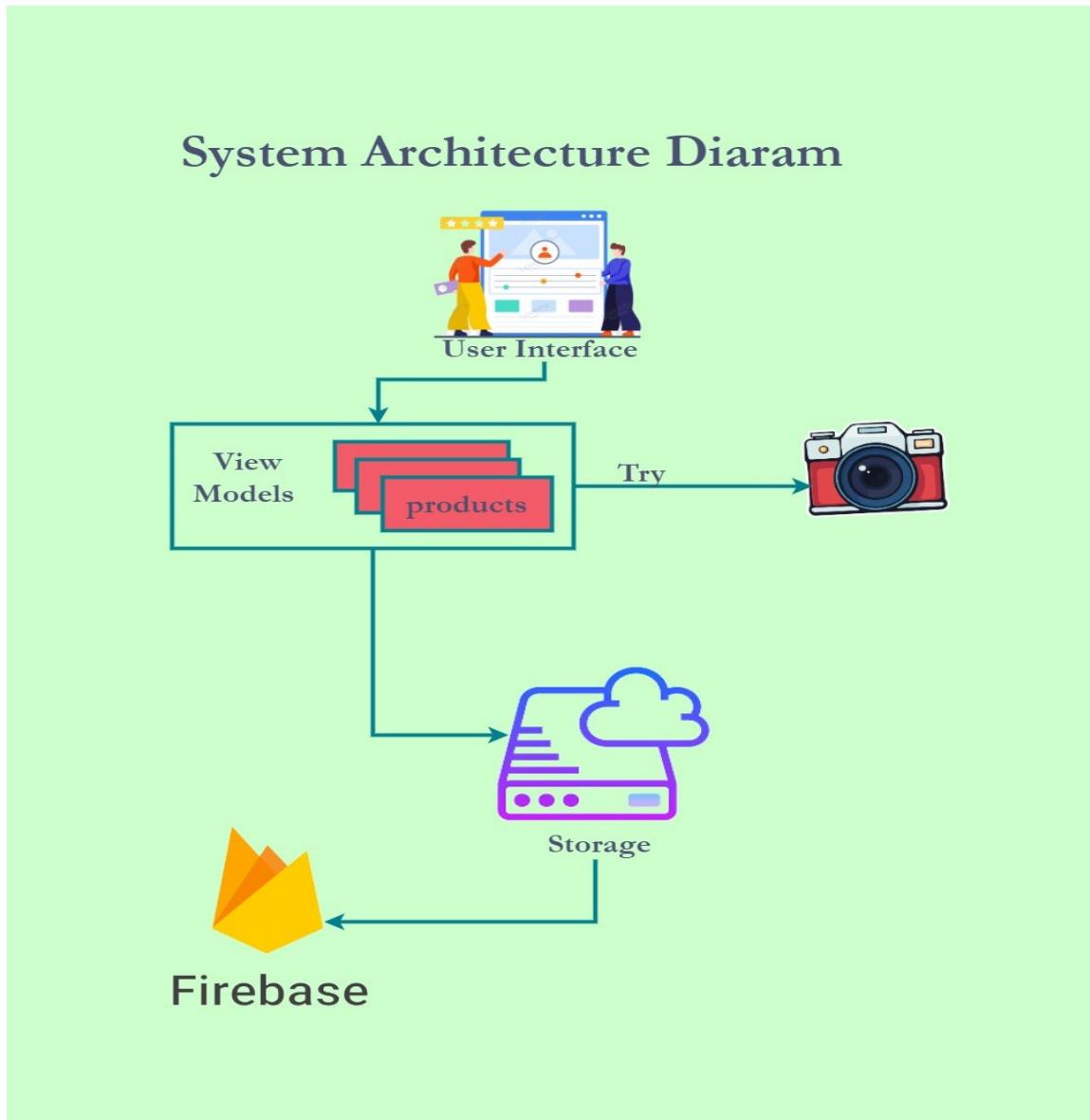


Figure 5.1: System Architecture Diagram

5.2 Class Diagram

The class diagram for an e-commerce app represents the structure and relationships between classes, depicting entities such as User, Product, Cart, and Order, showcasing how they interact and collaborate to facilitate online shopping functionality.

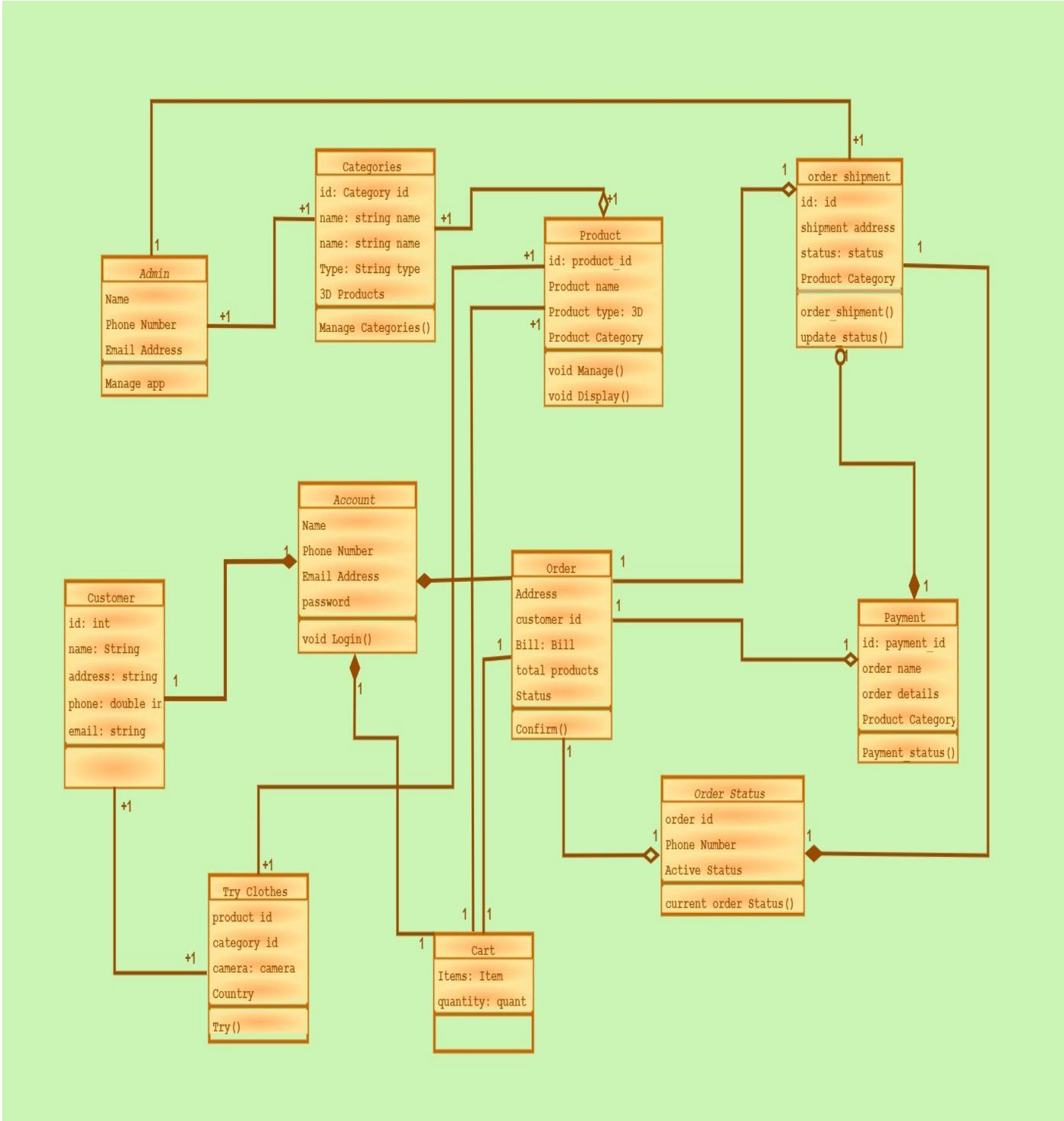


Figure 5.2: System Class Diagram

5.3 Sequence Diagrams

Sequence Diagram for E-commerce App: Illustrates the flow of interactions between actors and the system, showcasing steps such as user login, product search, adding items to the cart, and completing the purchase process.

5.3.1 System Sequence diagram

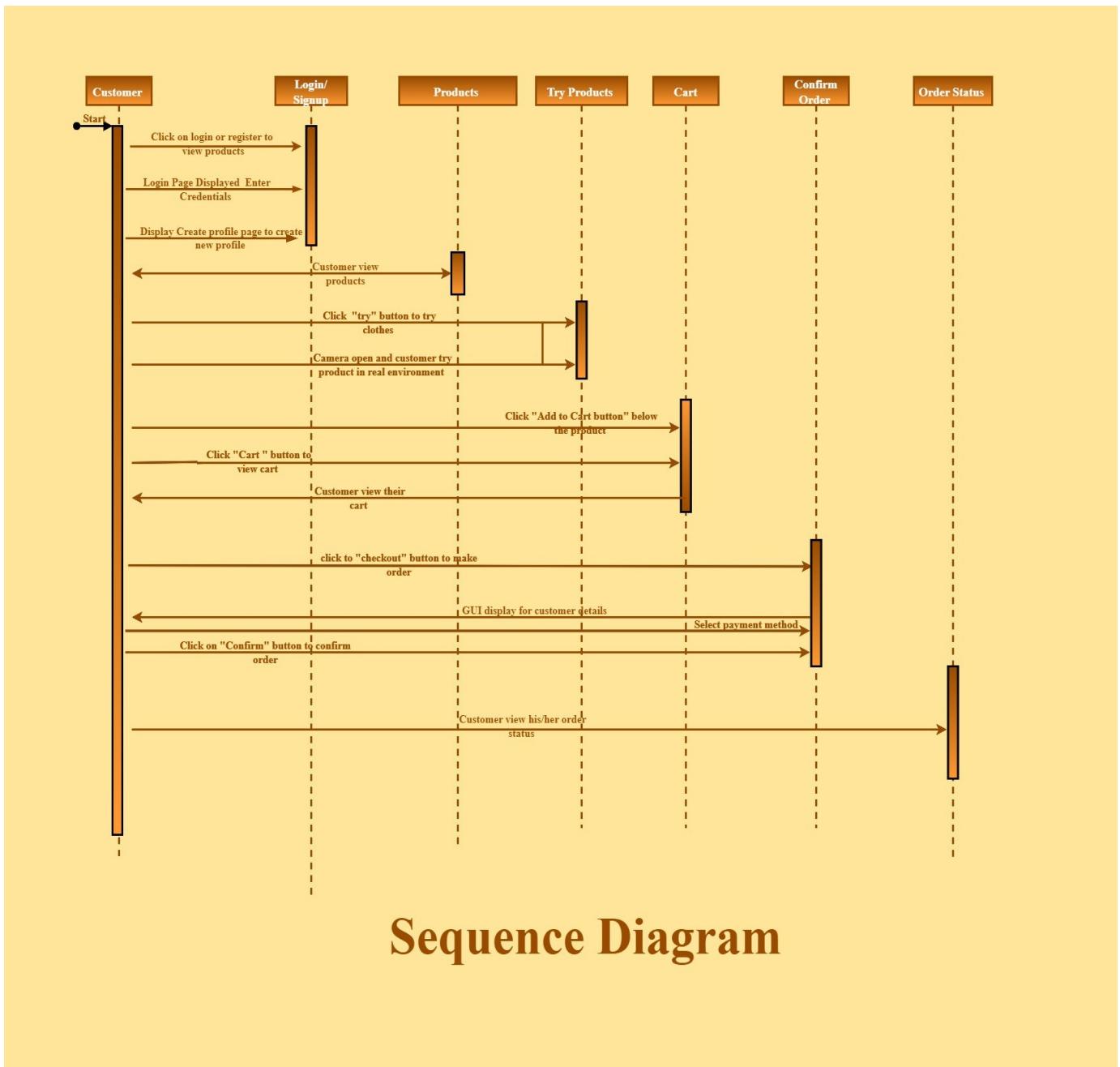
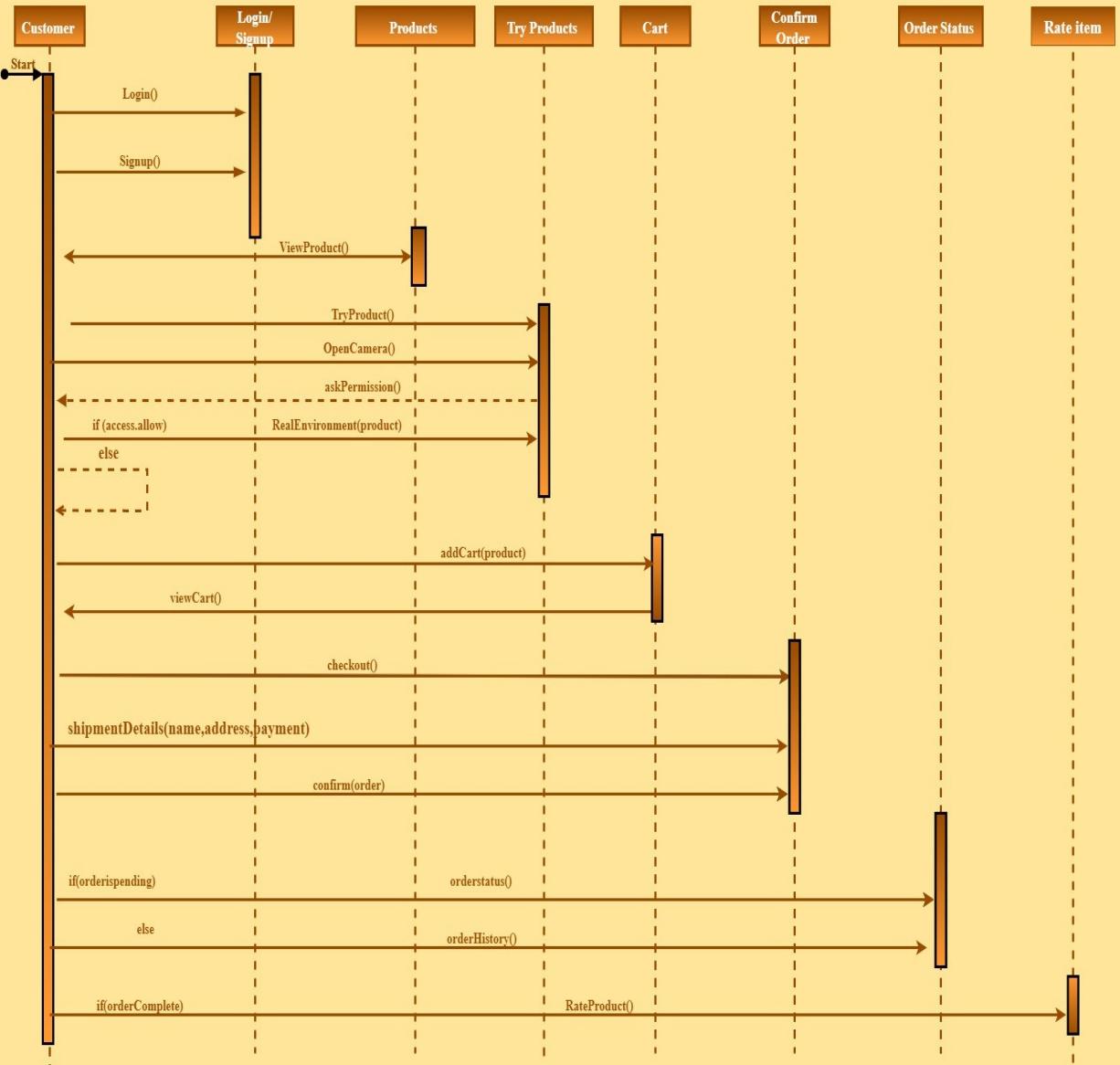


Figure 5.3: System Sequence Diagram

5.3.2 Object Sequence diagram



Object Sequence Diagram

Figure 5.4: System Object Sequence Diagram

5.4 Collaboration Diagrams

A collaboration diagram for an e-commerce app shows how different elements, such as users, the shopping cart, the payment gateway, and the inventory system, communicate with one another to enable smooth online buying experiences.

5.4.1 Customer Collaboration Diagram

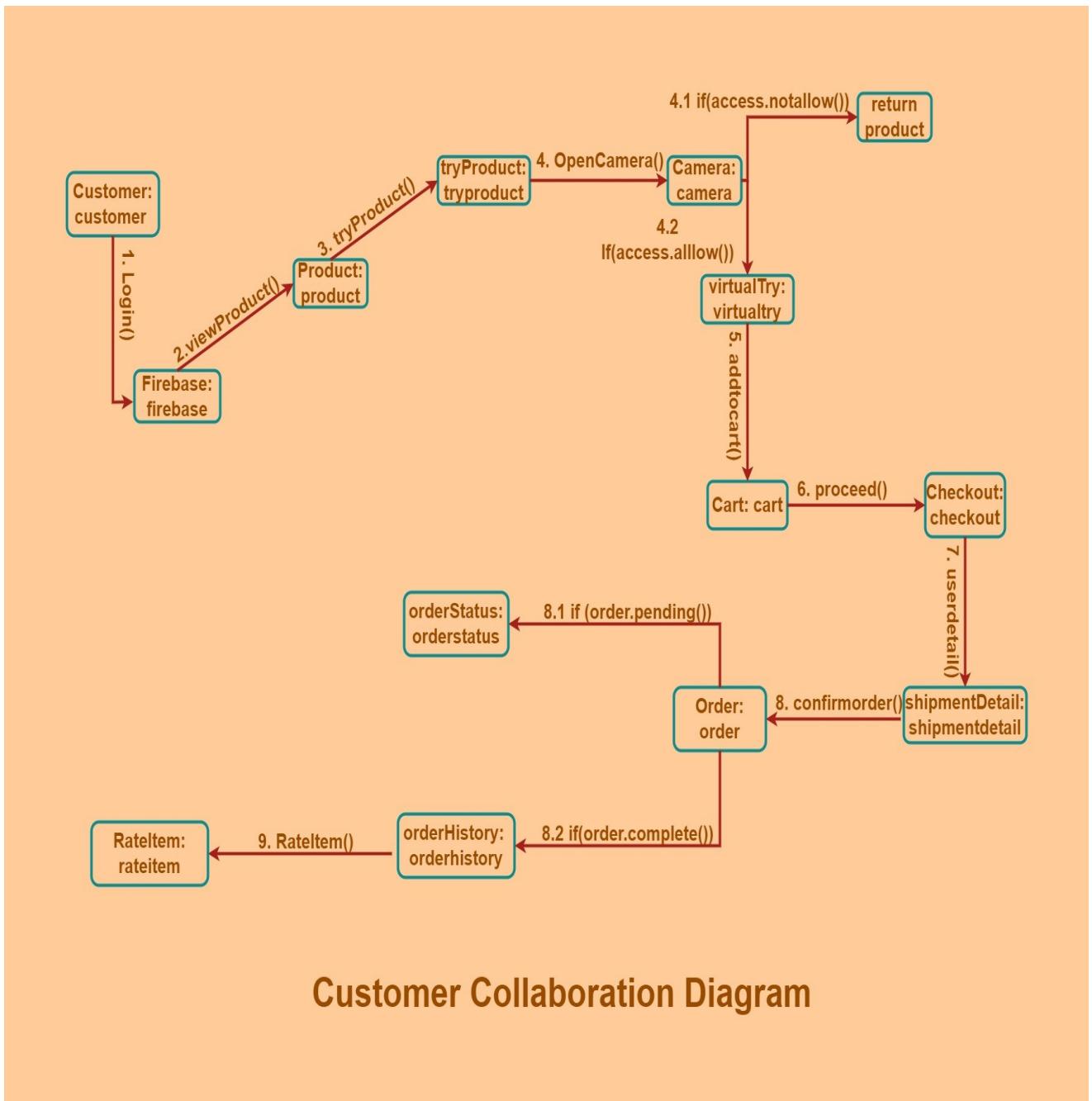


Figure 5.5: Customer Collaboration Diagram

5.4.2 Admin Collaboration diagram

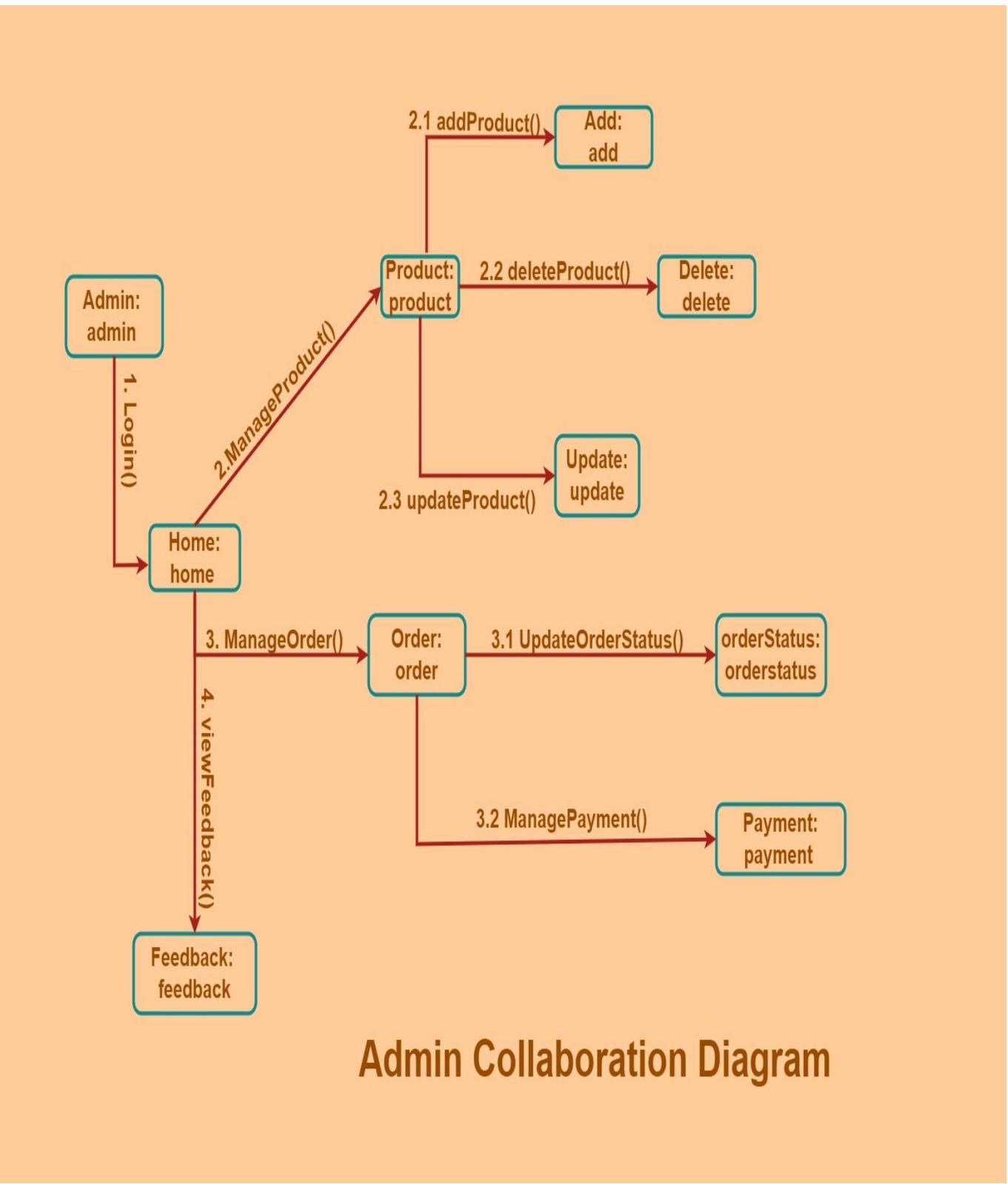


Figure 5.6: Admin Collaboration Diagram

5.5 Activity Diagram

A visual representation of the app's essential features and user interactions, an activity diagram for an e-commerce app depicts the sequence of actions, such as user registration, product browsing, adding things to the cart, and going to checkout.

5.5.1 Customer Activity Diagram

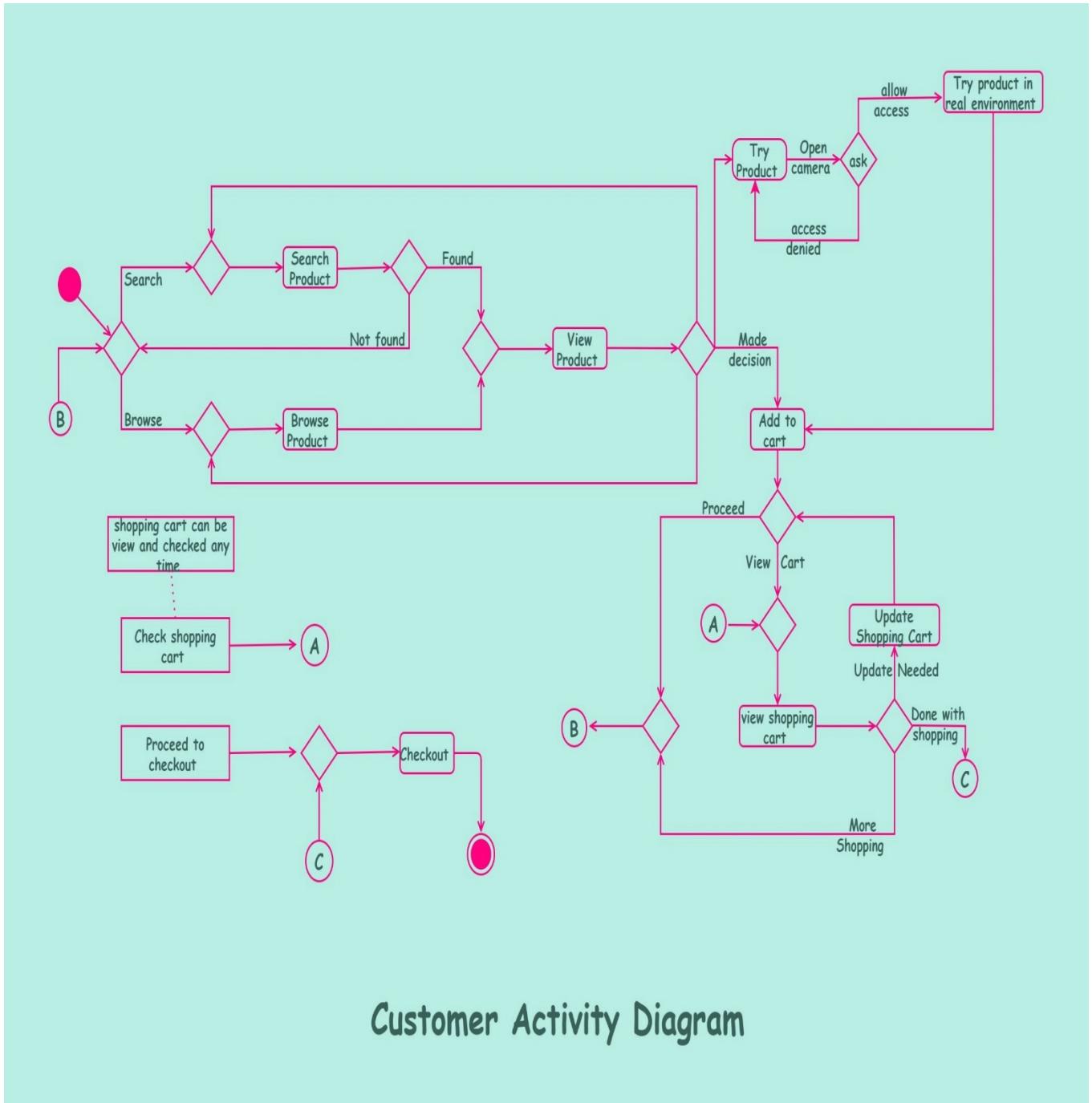
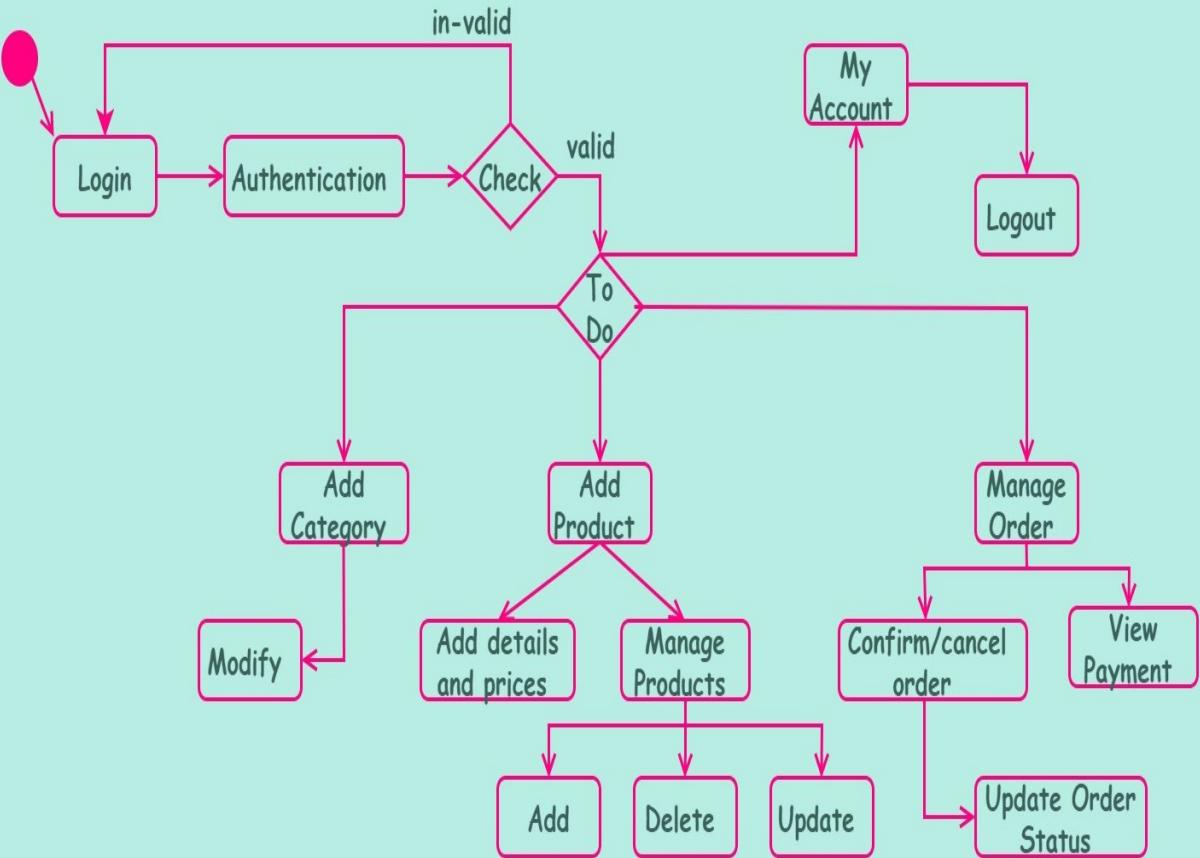


Figure 5.7: Customer Activity Diagram

5.5.2 Admin Activity Diagram



Admin Activity Diagram

Figure 5.8: Admin Activity Diagram

5.6 State Chart Diagram

An activity diagram for an e-commerce app shows the sequence of operations, such as user registration, product browsing, adding items to the basket, and proceeding to checkout. It is a visual depiction of the program's key features and user interactions.

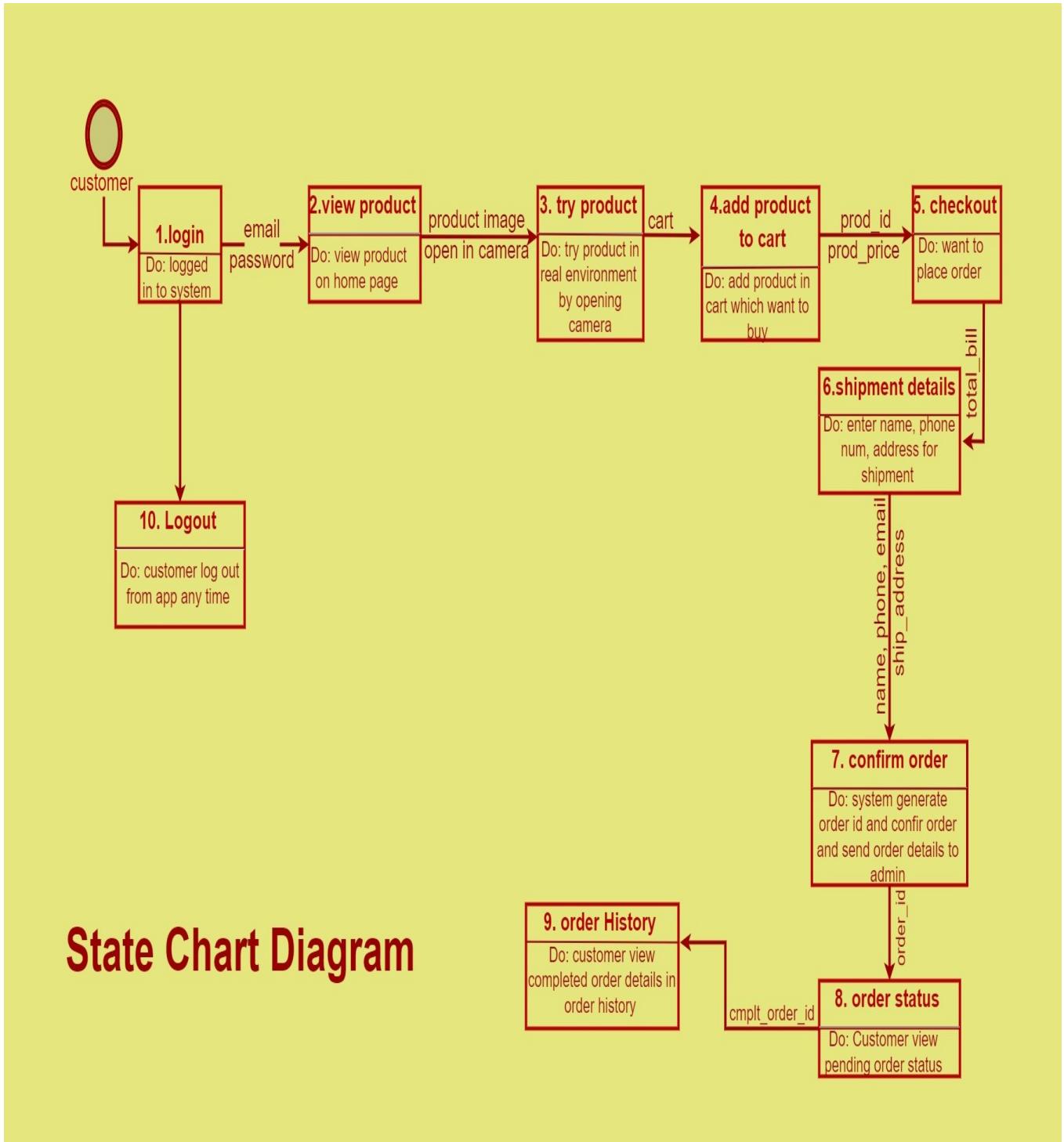


Figure 5.9: System State Chart Diagram

5.7 Database Design

The database design diagram for the e-commerce software provides a visual depiction of how the data is arranged and integrated inside the system by highlighting the structure and connections between various data entities, including customers, items, orders, and payments.

5.7.1 Entity Relationship Diagram

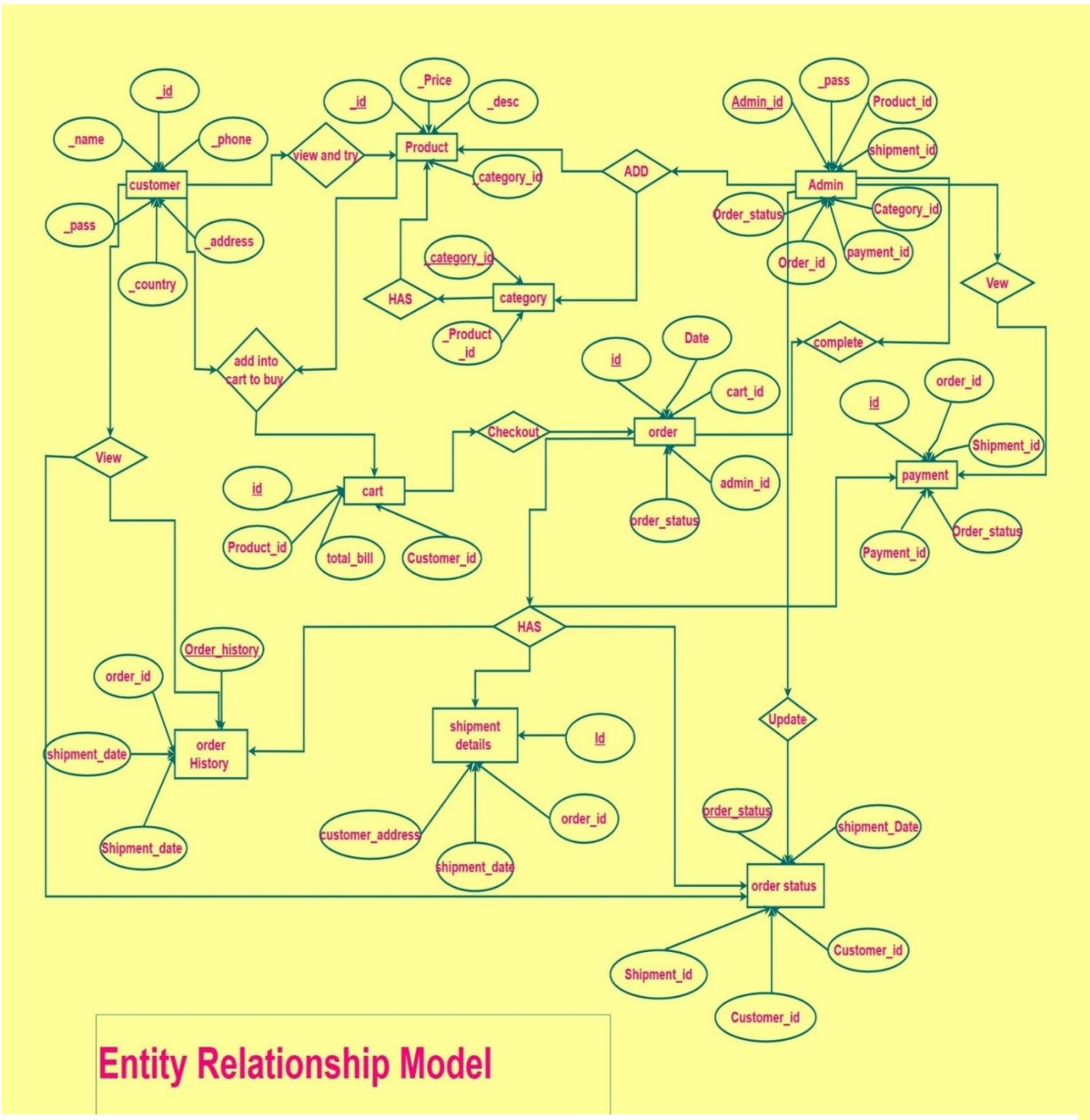


Figure 5.10: Entity Relationship Diagram

5.7.2 Entity Relationship Model

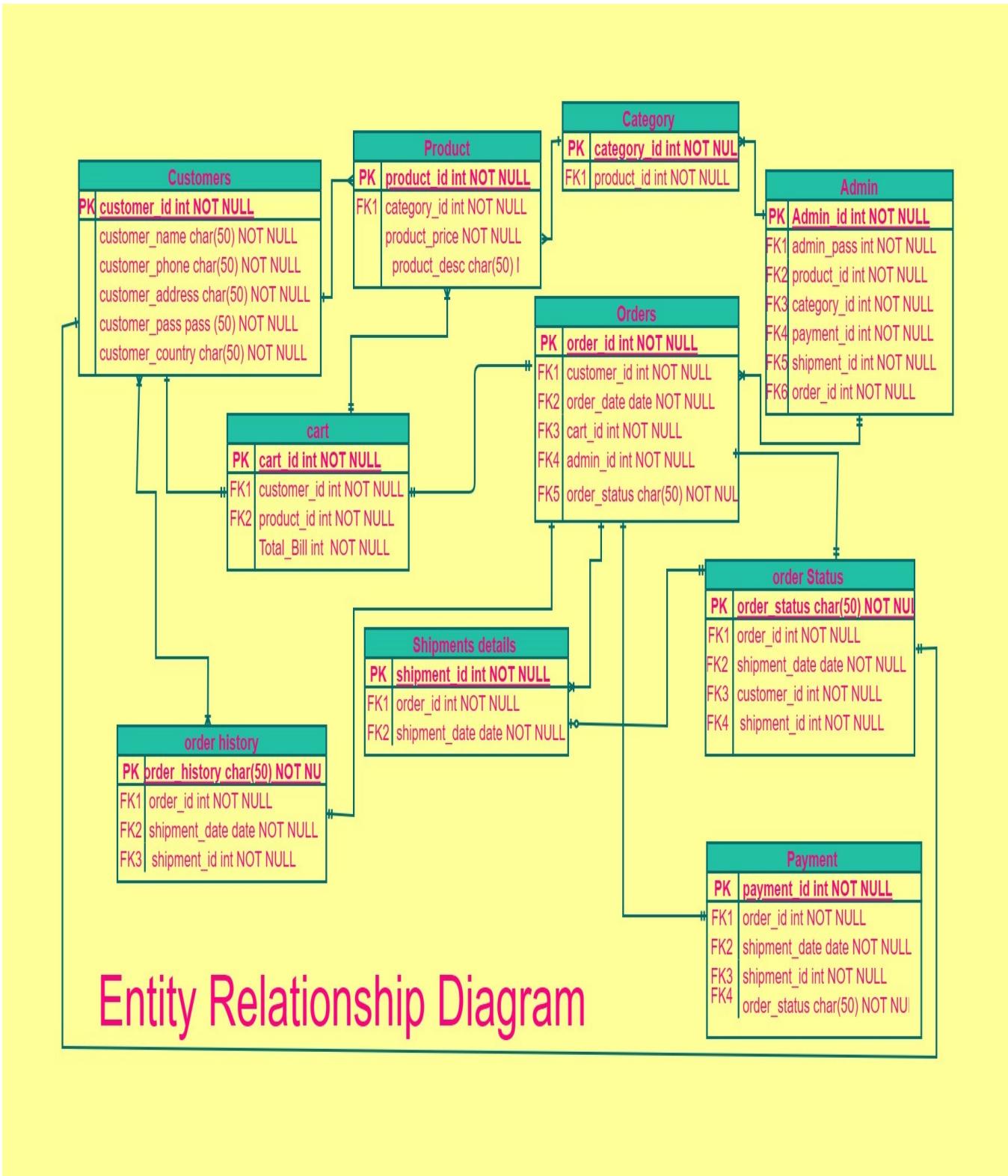


Figure 5.11: Entity Relationship Model

5.8 Data Dictionary

The data dictionary table in a database provides a comprehensive description of the data elements, their attributes, and their relationships within the database. It serves as a centralized reference guide for developers and users, aiding in data understanding, integrity, and efficient data management.

Element Name	Type	Validation	Mandatory	Description
Customer_id	Int	@NotNull	Yes	Customer must have a specific unique id for their further actions
Customer_name	Nvarchar(50)	@NotNull	No	Customer have some name that he/she will use while using app
Customer_address	Nvarchar(50)	@NotNull	Yes	Customer have some address that he/she use for order
Customer_phone	Char(50)		No	Customer have phone number that they use for order
Customer_country	Country	@NotNull	Yes	The shipment facility is available for some specific country. So it's a mandatory for order details of customer.
Product_id	Int	@NotNull	Yes	Some product has same name and categories, so unique id of product distinguishes between same products.
Product_name	Nvarchar(50)		No	Whenever admin add some product, they shall have some name. So, customer search product by name.
Product_price	Int	@NotNull	Yes	Price is mandatory field; every product has some price.

Product_desc	Nvarchar(50)		No	Product description is helpful for Customer understanding that what it is.
Admin_id	Int	@Not Null	Yes	Admin shall have unique id so he/she perform important task.
Admin_Pass	password	@Not Null	Yes	Admin password is most important element, because he/she manage all ap by entering correct credentials.
Cart_id	Int	@Not Null	Yes	Every time customer add some product in his/her cart that cart have some unique id against the customer id.
Total_Bill	Int	@Not Null	Yes	The total bill is generated when user want to check out the cart. The system adds up all product prices, tax and shipment fee to calculate total bill.
Category_id	Int @Not Null	@Not Null	Yes	Every category added by admin have some unique id so products are enlisted according to category id.
Order_id	Int	@Not Null	Yes	Every order from customer has some unique id.
Order_date	Date	@Not Null	Yes	When customer place any order, the date is automatically saved on db

Order_status	Char(50)	@Not Null	Yes	Every pending order have some status like confirm dispatched or deliver. It is update by Admin
Shipment_id	Int @Not Null	@Not Null	Yes	When ever admin get an order from customer and it is ready to ship. Admin give a unique id for shipment to keep record of shipment.
Shipment_details	Nvarchar(50)	@Not Null	Yes	Shipment details that user enter to complete order is managed by admin. He use this information to complete order.
Shipment_date	Date	@Not Null	Yes	Shipment date is update on database when admin ship the order.
Order_History	Char(50)	@Not Null	Yes	Every completed order saved on customer order history.
Payement_id	Int @Not Null	@Not Null	Yes	Every order has unique payment id, it helps admin to keep record of order payments.

Table 5.1: Data Dictionary

6. IMPLEMENTATION DETAILS

6.1 Development Setup

To develop an Augmented Reality (AR) E-Commerce Android app, you need a combination of tools and technologies. Here are some commonly used ones:

6.1.1 Development Tools:

Android Studio: The official IDE for Android app development, providing a comprehensive environment for coding, debugging, and testing.

AR Core: Google's AR platform that enables building AR experiences on Android devices.

6.1.2 Programming Languages:

Java: The primary programming language for Android app development.

6.1.3 Augmented Reality Frameworks:

AR Core: A framework that allows developers to build AR experiences, including object recognition, tracking, and environmental understanding.

6.1.4 3D Modeling and Design Tools:

Blender: A free and open-source 3D modeling software for creating 3D assets.

Maya: A professional 3D modeling and animation software widely used in the industry.

SketchUp: A user-friendly 3D modeling tool suitable for quick prototyping.

6.1.5 UI/UX Design Tools:

Adobe XD: A design and prototyping tool for creating user interfaces and interactive experiences.

Sketch: A vector graphics editor with plugins available for designing mobile app interfaces.

6.1.6 Backend and APIs:

Firebase: A comprehensive platform that provides services such as authentication, real-time database, storage, and cloud functions.

6.1.7 Testing and Debugging:

Android Debug Bridge (ADB): A command-line tool for debugging and interacting with Android devices.

Android Emulator: Built-in with Android Studio, it allows you to test your app on virtual Android devices.

6.2 Deployment setup

Here's an algorithmic overview of how you can implement the "Try Shoes with AR" feature in an Android app:

6.2.1 Algorithms

6.2.1.1 Algorithm for Try shoes in Real Environment

Set up the AR Environment:

- Initialize the AR Core session in Android app.
- Configure the camera view and projection matrix for rendering the augmented scene.
- Enable plane detection to identify the ground surface where the shoes will be placed.

Detect and Track Feet:

- Use image recognition or computer vision algorithms to detect and track the user's feet in real-time.
- Apply pose estimation techniques to estimate the position and orientation of each foot.
- Continuously update the foot tracking information as the user moves or rotates their feet.

Selecting and Displaying Shoes:

- Implement a user interface for browsing and selecting different shoe styles and sizes.
- Retrieve the 3D models or assets for each shoe style from your local storage or a remote server.
- Display a virtual representation of the selected shoe model in the AR scene, aligned with the detected foot positions.

Aligning Shoes with Feet:

- Use the tracked foot positions and orientations to align the virtual shoes with the user's feet.
- Apply transformation matrices to position and rotate the virtual shoe models accordingly.
- Implement interactive controls to adjust the position, rotation, and scaling of the shoes relative to the feet.

Realistic Rendering and Lighting:

- Apply appropriate materials, textures, and shaders to the shoe models for realistic rendering.
- Consider the lighting conditions in the real environment and adjust the virtual lighting to match.
- Implement techniques such as shadows and reflections to enhance the realism of the virtual shoes.

User Interaction:

- Enable user interaction to toggle between different shoe styles or sizes.
- Allow users to customize shoe attributes like color or material if applicable.
- Implement intuitive gestures or controls for adjusting the virtual shoes on the user's feet.

Performance Optimization:

- Implement occlusion techniques to hide portions of the virtual shoes that would be hidden behind real-world objects.
- Use level of detail (LOD) techniques to optimize the rendering of the virtual shoe models based on their distance from the camera.
- Optimize the AR rendering pipeline to maintain a smooth and responsive experience on the user's device.

Finalize and Complete Purchase:

- Once the user has selected the desired shoe style and size, provide options to complete the purchase within the app.
- Integrate with a payment gateway to process the transaction securely.
- Handle order fulfillment and provide necessary confirmation to the user.

This algorithm provides a high-level overview of the steps involved in implementing an AR-based "Try Shoes" feature. The specific implementation details and algorithms depending on the chosen AR framework, 3D modeling tools, and the features and interactions you want to include in your app.

6.2.1.2 Algorithm for Try Clothes in Real Environment

Here's an algorithmic overview of how you can implement the "Try Clothes with AR" feature in an Android app:

Set up the AR Environment:

- Initialize the AR Core session in your Android app.
- Configure the camera view and projection matrix for rendering the augmented scene.
- Enable plane detection to identify the surfaces where the clothes will be placed.

Detect and Track Body:

- Use image recognition or computer vision algorithms to detect and track the user's body in real-time.
- Apply pose estimation techniques to estimate the position and orientation of the body.
- Continuously update the body tracking information as the user moves or rotates.

Selecting and Displaying Clothes:

- Implement a user interface for browsing and selecting different clothing styles and sizes.
- Retrieve the 3D models or assets for each clothing item from your local storage or a remote server.
- Display a virtual representation of the selected clothing item in the AR scene, aligned with the detected body position.

Aligning Clothes with Body:

- Use the tracked body position and orientation to align the virtual clothing item with the user's body.
- Apply transformation matrices to position and rotate the virtual clothing model accordingly.
- Implement interactive controls to adjust the position, rotation, and scaling of the clothes relative to the body.

Realistic Rendering and Lighting:

- Apply appropriate materials, textures, and shaders to the clothing models for realistic rendering.
- Consider the lighting conditions in the real environment and adjust the virtual lighting to match.
- Implement techniques such as shadows and reflections to enhance the realism of the virtual clothes.

User Interaction:

- Enable user interaction to toggle between different clothing styles or sizes.
- Allow users to customize clothing attributes like color or pattern if applicable.
- Implement intuitive gestures or controls for adjusting the virtual clothes on the user's body.

Performance Optimization:

- Implement occlusion techniques to hide portions of the virtual clothes that would be hidden behind real-world objects or the user's body.
- Use level of detail (LOD) techniques to optimize the rendering of the virtual clothing models based on their distance from the camera.
- Optimize the AR rendering pipeline to maintain a smooth and responsive experience on the user's device.

Finalize and Complete Purchase:

- Once the user has selected the desired clothing item, provide options to complete the purchase within the app.
- Integrate with a payment gateway to process the transaction securely.
- Handle order fulfillment and provide necessary confirmation to the user.
- Remember, this algorithm provides a high-level overview of the steps involved in implementing an AR-based "Try Clothes" feature. The specific implementation details and algorithms may vary depending on the chosen AR framework, 3D modeling tools, and the features and interactions you want to include in your app.

6.2.2 Constraints

Here is the constraints of system.

6.2.2.1 Assumptions

- Users have compatible Android devices capable of running augmented reality applications.
- Users have access to a stable internet connection for seamless browsing and purchasing.
- Users are familiar with basic smartphone operations and app navigation.
- Users have an understanding of the concept of augmented reality and its limitations
- Users have registered accounts with valid personal information to complete purchases and track orders.
- Users are willing to share their location information for delivery purposes, subject to privacy regulations and permissions.

- Users have a secure payment method and are comfortable entering payment details within the app.

6.2.2.2 System constraints

- The app should be compatible with a range of Android devices, considering various screen sizes, processing power, and memory capacities.
- The app should be optimized to run smoothly and efficiently, minimizing resource consumption and battery usage.
- The app should adhere to the guidelines and restrictions imposed by the Android operating system, ensuring compatibility and security
- The app should implement secure user authentication and data encryption to protect user accounts and personal information.
- The app should integrate with trusted payment gateways and adhere to their security guidelines.
- The app should comply with data protection regulations and ensure proper handling of user data

6.2.2.3 Restrictions

- The accuracy and stability of the augmented reality experience may vary depending on environmental factors such as lighting conditions and surface textures.
- The app may require a significant amount of storage space for 3D models, textures, and other assets used in the augmented reality experience.
- The performance of the app may be impacted by the complexity and number of 3D models rendered simultaneously.
- The app's functionality and augmented reality features may be limited by the capabilities and constraints of the AR framework and APIs used.
- The availability of specific products and inventory may be subject to limitations imposed by suppliers or third-party integration.

6.2.2.4 Limitations

- The app should implement secure user authentication and data encryption to protect user accounts and personal information.
- The app should integrate with trusted payment gateways and adhere to their security guidelines.
- The app should comply with data protection regulations and ensure proper handling of user data

7. TESTING

7.1 Extended Test Cases

Here are all the extended test cases of the app, which include Login, Sign-Up, Shopping Cart, Try Products, Order and others.

7.1.1 Sign-Up

Test Case ID:	TC_01		
Test Case Name:	Sign-up		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Admin, Customer		
Description:	All users shall be sign-up for the first time to interact with the app.		
Pre-Conditions	Following is the pre-condition. <ul style="list-style-type: none">• Ensure that the app is installed on the customer's device.• Verify that the customer does not have an existing account in the system.• Confirm that the customer has a stable internet connection.		
Actions	The normal flow is: <ul style="list-style-type: none">• Open the mobile app.• Click on the "Register" button.• Enter a valid name, email, password, and all required details.• Enter the same password again in the confirm password field.• Enter the captcha code.• Click on the "Register" button.		
Expected Result	<ul style="list-style-type: none">• The app should validate the entered information and register the customer successfully.• The app should display a success message indicating that the account has been created.• The customer should be redirected to the login page to log in with the newly registered credentials.		
Post Conditions	<ul style="list-style-type: none">• Verify that the customer's account is created and stored securely in the system.• Confirm that the customer can now log in using the registered email and password.• The customer's personal information should be saved accurately in the system.		
Test Data	Valid name, email, and password. Matching confirm password. Valid captcha code.		

Test Environment	<ul style="list-style-type: none"> • Ensure that the app is running on a compatible Android device. • Confirm that the app has necessary permissions and access to the required resources. • Mock or test server environment to simulate the registration process.
Test Case Execution	<ul style="list-style-type: none"> • Execute the test steps as outlined above. • Observe the app's behavior during the sign-up process. • Verify that the expected results are met
Test Result	<ul style="list-style-type: none"> • Capture any observed anomalies, errors, or unexpected behavior during the test execution. • Record the test results as Pass or Fail based on whether the expected results were achieved
Test Closure	<ul style="list-style-type: none"> • After completing the test, ensure that the app is in a stable state and the customer's account is created successfully. • Document any issues or notes related to the test execution
Actual Result	Same as expected

Table 7.1: Test Case for Sign-Up

7.1.2 Login

Test Case ID:	TC_02		
Test Case Name:	Login		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Description:	Admin and Customer shall login after registration.		
Pre-Conditions	<p>Following is the post condition.</p> <ul style="list-style-type: none"> • Ensure that the app is installed on the actor's device. • Verify that the actor has a registered account in the system. • Confirm that the actor has a stable internet connection. 		
Actions	<ul style="list-style-type: none"> • Open the mobile app. • Click on the "Login" button. • Enter a valid email and password for the registered account. • Click on the "Login" button. 		
Expected Result	<ul style="list-style-type: none"> • The app should validate the entered email and password. • If the entered credentials are valid, the app should log in the actor successfully. • The app should display a success message indicating the successful login. • The actor should be redirected to their respective home page based on their role (customer or admin). 		
Post Conditions	<ul style="list-style-type: none"> • Verify that the actor is logged in and can access the features and functionalities specific to their role. • The actor's session and authentication data should be maintained until the next update or logout 		

Test Data	<ul style="list-style-type: none"> Valid email and password for the registered account
Test Environment	<ul style="list-style-type: none"> Ensure that the app is running on a compatible Android device. Confirm that the app has necessary permissions and access to the required resources. Mock or test server environment to simulate the login process
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Observe the app's behavior during the login process. Verify that the expected results are met.
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. Record the test results as Pass or Fail based on whether the expected results were achieved
Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the app is in a stable state and the actor is successfully logged in. Document any issues or notes related to the test execution
Actual Result	Same as Expected

Table 7.2: Test Case for Login

7.1.3 View and Try Product

Test Case ID:	TC 03		
Test Case Name:	View and Try Product		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Customer		
Description:	Customer try product From app by browsing desire product.		
Pre-Conditions	<ul style="list-style-type: none"> Following is the post condition. Ensure that the customer is logged in to the app. Verify that the customer's device has a camera. Confirm that the customer has a stable internet connection. Ensure that the customer's device meets the minimum Android version requirement (at least 8.1). 		
Actions	<ul style="list-style-type: none"> Open the mobile app. Navigate to the home page. Select a category for a specific product. Click on the 3D image of the product. Allow the app to access the camera by granting the necessary permission 		
Expected Result	<ul style="list-style-type: none"> The app should display the selected product in a 3D image view. The app should show a pop-up message requesting permission to access the camera. 		

	<ul style="list-style-type: none"> If the customer grants camera access, the app should open the camera in real-time to overlay the product in the customer's environment. The customer should be able to visually try the product using the augmented reality feature.
Post Conditions	<ul style="list-style-type: none"> Verify that the customer can view and try the selected product using the app's augmented reality feature. The app should handle the camera access and overlay the product accurately in the customer's environment. The customer should be able to interact with the augmented reality view to rotate, resize, or move the product.
Test Data	<ul style="list-style-type: none"> Valid login credentials for the customer account. A product selected from a specific category.
Test Environment	<ul style="list-style-type: none"> Ensure that the app is running on a compatible Android device with a camera. Confirm that the app has necessary permissions and access to the camera. Mock or test server environment to simulate the product and augmented reality features.
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Observe the app's behavior during the viewing and trying of the product. Verify that the expected results are met
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. Record the test results as Pass or Fail based on whether the expected results were achieved.
Test Closure	<ul style="list-style-type: none"> After completing the test, ensure that the app is in a stable state and the customer can successfully view and try products using the augmented reality feature. Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.3: Test Case for View and Try Product

7.1.4 Cart Functionality

Test Case ID:	TC_04		
Test Case Name:	Cart Functionality		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Customer		
Description:	Customer shall add item into cart which they want to buy.		
Pre-Conditions	<ul style="list-style-type: none"> Ensure that the customer is logged in to the app. Verify that the customer has selected a specific product from a specific category. 		

	<ul style="list-style-type: none"> Ensure that the selected product is in stock
Actions	<ul style="list-style-type: none"> Click on the "View" button to view the product details. Select the desired quantity of the product. Click on the "Add to Cart" button
Expected Result	<ul style="list-style-type: none"> The product details should be displayed. The customer should be able to select the desired quantity of the product. After clicking "Add to Cart," a pop-up message should appear with the text "Item successfully added to cart." The product should be added to the customer's cart
Post Conditions	<ul style="list-style-type: none"> Verify that the product is successfully added to the customer's cart. The pop-up message confirming the successful addition should be displayed. The customer's cart should reflect the added product.
Test Data	<ul style="list-style-type: none"> Valid login credentials for the customer account. A product selected from a specific category. Quantity of the product to be added to the cart.
Test Environment	<ul style="list-style-type: none"> Ensure that the app is running on a compatible device. Verify that the customer's cart is empty before executing the test case
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Observe the app's behavior during the cart functionality. Verify that the expected results are met.
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. Record the test results as Pass or Fail based on whether the expected results were achieved
Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the cart functionality is working as expected. Verify that the added product is correctly reflected in the customer's cart. Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.4: Test Case for Cart Functionality

7.1.5 Place Order

Test Case ID:	TC_05		
Test Case Name:	Place Order		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Customer		
Description:	After adding items to cart customer place order by proceeding.		

Pre-Conditions	<ul style="list-style-type: none"> • Ensure that the customer is logged in to the app. • Verify that the customer has added products to their cart. • The total bill, including the shipment fee, should be calculated and displayed on the customer's screen.
Actions	<ul style="list-style-type: none"> • Click on the "Cart" button. • View the total bill displayed on the cart page. • Click on the "Checkout" button. • Enter the shipping address and details on the order details page. • Select the desired payment method. • If applicable, enter a different payment address. • Click on the "Confirm Order" button.
Expected Result	<ul style="list-style-type: none"> • The customer should be able to access the cart page. • The total bill should be displayed accurately. • After clicking "Checkout," the customer should be directed to the order details page. • The customer should be able to enter the shipping address and details. • The customer should be able to select the preferred payment method. • If applicable, the customer should be able to enter a different payment address. • After clicking "Confirm Order," a countdown of 30 seconds should start. • If the customer does not cancel the order within 30 seconds, the order should be successfully placed. • A pop-up message should appear with the order ID and confirmation of the order.
Post Conditions	<ul style="list-style-type: none"> • Verify that the order is successfully placed and saved against the customer's user ID. • The pop-up message confirming the order placement should be displayed. • The order details should be saved in the database.
Test Data	<ul style="list-style-type: none"> • Valid login credentials for the customer account. • Products added to the customer's cart. • Shipping address and details. • Selected payment method.
Test Environment	<ul style="list-style-type: none"> • Ensure that the app is running on a compatible device. • Verify that the customer's cart contains products. • The customer's account and order history are accessible.
Test Case Execution	<ul style="list-style-type: none"> • Execute the test steps as outlined above. • Observe the app's behavior during the order placement process. • Verify that the expected results are met.
Test Result	<ul style="list-style-type: none"> • Capture any observed anomalies, errors, or unexpected behavior during the test execution. • Record the test results as Pass or Fail based on whether the expected results were achieved.

Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the order placement functionality is working as expected. Verify that the order is successfully placed and the order details are accurately saved. Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.5: Test Case for Place Order

7.1.6 View Order Status

Test Case ID:	TC_06		
Test Case Name:	View Order Status		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Customer		
Description:	Customer view their order status of pending order.		
Pre-Conditions	<ul style="list-style-type: none"> Ensure that the customer is logged in to the app. Verify that the customer has at least one processing order in the queue. 		
Actions	<ul style="list-style-type: none"> Click on the "My Account" icon. Navigate to the "Order Status" tab. 		
Expected Result	<ul style="list-style-type: none"> The customer should be able to access the "My Account" section. The "Order Status" tab should be visible and accessible. The customer should be able to view the status of their order(s) that are currently in the processing queue. 		
Post Conditions	<ul style="list-style-type: none"> Verify that the customer is able to view the order status without any issues. The order status should accurately reflect the current status of the customer's processing order(s). 		
Test Data	<ul style="list-style-type: none"> Valid login credentials for the customer account. At least one processing order in the customer's queue. 		
Test Environment	<ul style="list-style-type: none"> Ensure that the app is running on a compatible device. Verify that the customer is logged in with a valid account. The customer's processing order(s) are visible and accessible in the order queue. 		
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Observe the app's behavior while accessing the "My Account" section and navigating to the "Order Status" tab. Verify that the customer can view the order status of their processing order(s). 		
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. 		

	<ul style="list-style-type: none"> Record the test results as Pass or Fail based on whether the expected results were achieved.
Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the order status functionality is working as expected. Verify that the customer can view the order status of their processing order(s) accurately. Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.6: Test Case for View Order Status

7.1.7 Manage Product

Test Case ID:	TC_07		
Test Case Name:	Manage product		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Admin		
Description:	Admin manage product by adding, deleting and updating them.		
Pre-Conditions	<ul style="list-style-type: none"> Ensure that the admin is logged in to the system with valid credentials. Verify that the system already has existing products or categories 		
Actions	<ul style="list-style-type: none"> Click on the "Manage Product" button. Click on the "Add Product" button. Select a product image from the internal storage. Enter the product name. Add a product description. Enter the product price. Select a category for the product. Click on the "Save" button to add the product. 		
Expected Result	<ul style="list-style-type: none"> The admin should be able to access the "Manage Product" section. The "Add Product" form should be visible and accessible. The admin should be able to successfully add a new product with all the required information. The product should be saved and stored in the database. 		
Post Conditions	<ul style="list-style-type: none"> Verify that the product has been added successfully. The added product should be visible and accessible in the system's product listing. The product details should include the correct name, description, price, and category. 		
Test Data	<ul style="list-style-type: none"> Valid login credentials for the admin account. Existing product categories in the system. Required information for adding a new product (name, description, price, category). 		

Test Environment	<ul style="list-style-type: none"> • Ensure that the system is running and accessible. • Verify that the admin is logged in with a valid account. • Existing products or categories are available in the system
Test Case Execution	<ul style="list-style-type: none"> • Execute the test steps as outlined above. • Fill in the necessary fields to add a new product. • Save the product and observe the system's response. • Verify that the product has been added and stored correctly.
Test Result	<ul style="list-style-type: none"> • Capture any observed anomalies, errors, or unexpected behavior during the test execution. • Record the test results as Pass or Fail based on whether the expected results were achieved.
Test Closure	<ul style="list-style-type: none"> • After completing the test, ensure that the product addition functionality is working as expected. • Verify that the added product is displayed correctly in the system's product listing. • Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.7: Test Case for Manage Product

7.1.8 Manage Order

Test Case ID:	TC_08		
Test Case Name:	Manage Order		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Admin		
Description:	Admin manage order by dispatching, or completing order		
Pre-Conditions	<ul style="list-style-type: none"> • Ensure that the admin is logged in to the system with valid credentials. • Verify that there is at least one pending order in the system 		
Actions	<ul style="list-style-type: none"> • Click on the "Manage Order" tab. • View the list of pending orders. • Select an order from the list. • Mark the order as ready for dispatch. • Update the order status to "Dispatched". 		
Expected Result	<ul style="list-style-type: none"> • The admin should be able to access the "Manage Order" section. • The list of pending orders should be visible and accessible. • The admin should be able to select an order from the list. • The admin should be able to mark the order as ready for dispatch. • The order status should be successfully updated to "Dispatched". 		
Post Conditions	<ul style="list-style-type: none"> • Verify that the order has been successfully dispatched. • The dispatched order should no longer be listed under pending orders. 		

	<ul style="list-style-type: none"> The order status should reflect the change to "Dispatched".
Test Data	<ul style="list-style-type: none"> Valid login credentials for the admin account. At least one pending order in the system.
Test Environment	<ul style="list-style-type: none"> Ensure that the system is running and accessible. Verify that the admin is logged in with a valid account. At least one pending order is available in the system.
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Locate a pending order in the system. Mark the order as ready for dispatch and update the status. Observe the system's response and any displayed messages or notifications.
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. Record the test results as Pass or Fail based on whether the expected results were achieved.
Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the order dispatch functionality is working as expected. Verify that the dispatched order is no longer listed under pending orders. Check that the order status has been correctly updated to "Dispatched". Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.8: Test Case for Manage Order

7.1.9 Logout

Test Case ID:	TC 09		
Test Case Name:	Log Out		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Admin, customer		
Description:	Admin and Customer logout from system when they no need system anymore.		
Pre-Conditions	<ul style="list-style-type: none"> Ensure that the user is logged in to the app. Verify that the logout button is available and accessible to the user. Confirm that the app is in a stable state 		
Actions	<ul style="list-style-type: none"> Click on the logout button in the app. 		
Expected Result	<ul style="list-style-type: none"> The app should successfully log out the user. The app should redirect the user to the default login page. The user's session and authentication data should be cleared. 		

Post Conditions	<ul style="list-style-type: none"> Verify that the user is no longer logged in and cannot access restricted features without logging back in. Confirm that the app does not retain any personal information or session data of the logged-out user.
Test Data	<ul style="list-style-type: none"> User is logged in with valid credentials.
Test Environment	<ul style="list-style-type: none"> Ensure that the app is running on a compatible Android device. Confirm that the app has necessary permissions and access to the required resources.
Test Case Execution	<ul style="list-style-type: none"> Execute the test steps as outlined above. Observe the app's behavior during the logout process. Verify that the expected results are met.
Test Result	<ul style="list-style-type: none"> Capture any observed anomalies, errors, or unexpected behavior during the test execution. Record the test results as Pass or Fail based on whether the expected results were achieved.
Test case Closure	<ul style="list-style-type: none"> After completing the test, ensure that the app is in a stable state and the user is successfully logged out. Document any issues or notes related to the test execution.
Actual Result	Same as Expected

Table 7.9: Test Case for Log Out

7.1.10 Update Order Status

Test Case ID:	TC_10		
Test Case Name:	Update Order Status		
Created By:	Ramsha Noor	Last Updated By:	Sundas Sarfraz
Date Created:	14-03-2022	Last Revision Date:	29-05-2023
Actors:	Admin		
Description:	Admin update order status		
Pre-Conditions	The admin is logged in to the system. There is at least one pending order in the list.		
Actions	Admin clicks on the "Update Order Status" dialog. Admin selects the order with ID 123456 from the list. Admin chooses "Shipped" from the drop-down menu as the new order status. Admin saves the changes.		
Expected Result	In step 4, the system should successfully save the new order status for order ID 123456.		
Post Conditions	The post condition is that the new order status ("Shipped") is saved in the system.		

Test Data	Order ID: 123456 New Order Status: "Shipped"
Test Environment	None
Test Case Execution	Test the test case in the development environment.
Test Result	The system successfully saves the new order status for order ID 123456. The post condition is satisfied, and the new order status is updated in the system.
Test case Closure	The test case has been executed and passed successfully. No defects or issues were identified. The test case can be closed.
Actual Result	Same as Expected

Table 7.10: Test Case for Update Order Status

7.2 Traceability Matrix

A traceability matrix for an eCommerce app ensures that each requirement or functionality is linked to the corresponding test cases, enabling comprehensive testing coverage and efficient tracking of software quality.

It also provides a clear mapping between requirements and implemented features, aiding in easy identification of any gaps or missing functionalities

7.2.1 RID vs UCID (requirements vs use cases)

UCID/R ID	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21	
UC 1	✓	✓	✓																			
UC 2		✓	✓																			
UC 3							✓	✓							✓							
UC 4											✓	✓				✓						
UC 5	✓	✓	✓		✓							✓					✓	✓				
UC 6					✓	✓									✓	✓			✓	✓	✓	
UC 7	✓		✓						✓							✓			✓	✓	✓	
UC 8	✓			✓						✓				✓	✓		✓	✓				
UC 9			✓		✓						✓			✓		✓	✓					
UC 10	✓					✓						✓										
UC 11			✓		✓			✓								✓	✓					
UC 12		✓		✓		✓		✓														

UC 13	✓		✓		✓										✓		✓	✓				
UC 14										✓												
UC 15						✓									✓							✓
UC 16					✓						✓											
UC 17															✓							

Table 7.11: RID vs UCID

7.2.2 Prototypes (RID vs PID)

PID/R ID	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21
PID 1																					
PID 2	✓	✓																			
PID 3			✓																		
PID 4							✓			✓	✓				✓						
PID 5																	✓	✓			
PID 6					✓	✓									✓	✓			✓	✓	✓
PID 7	✓	✓	✓						✓							✓			✓	✓	✓
PID 8						✓				✓			✓	✓			✓	✓			
PID 9																✓	✓				
PID 10							✓							✓							
PID 11	✓																				✓
PID 12	✓							✓						✓	✓						✓

Table 7.12: RID vs PID

7.2.3 Test Cases (RID vs TID)

UCID/R ID	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21
TC 1	✓	✓																			
TC 2		✓	✓																		
TC 3								✓	✓						✓						
TC 4											✓					✓					

TC 5					✓					✓			✓	✓	✓	✓		
TC 6				✓		✓					✓	✓		✓		✓		
TC 7					✓			✓				✓		✓		✓		
TC 8				✓				✓			✓		✓	✓				
TC 9	✓																✓	
TC 10	✓														✓	✓		

Table 7.13: RID vs TID

8. RESULTS/OUTPUT/STATISTICS

8.1 Completion

The completion of the specified requirements (FR_1.0 to FR_3.4) has been achieved in the system. The system successfully includes all the necessary functionalities and features outlined in the requirements, ensuring a comprehensive and fully-functional application.

- The login feature for both the admin and customers is complete, allowing them to securely access their respective accounts (FR_1.0, FR_1.1, FR_1.9). The registration process for customers is also fully implemented, ensuring that they can complete their registration during their initial login (FR_1.2).
- The admin has complete control over the management of 3D products, including adding, deleting, and updating their descriptions and details (FR_1.3, FR_1.4, FR_1.5). This empowers the admin to effectively maintain the product inventory.
- Customers have the complete ability to view 3D products, including their descriptions and details (FR_1.6). They can browse the available products and make informed decisions based on the provided information.
- The admin has complete control over managing order requests, including the ability to remove, cancel, and complete orders (FR_1.7, FR_1.8). This ensures that the admin can efficiently handle order management tasks.
- The system provides complete functionality for the admin to view feedback from customers (FR_2.1) and access information about all payments (FR_2.2). This allows the admin to monitor customer satisfaction and track payment records effectively.
- Customers have complete control over changing their passwords (FR_2.3) and have access to view all available product categories (FR_2.4). This allows them to personalize their account settings and easily navigate through different product categories.
- The augmented reality feature for trying products online is fully implemented, providing customers with a complete and immersive virtual trial experience (FR_2.5). Customers can add products to their cart and view the cart, ensuring a complete shopping experience (FR_2.6, FR_2.7).
- Customers have complete visibility into the status of their orders, whether they are pending, completed, or dispatched (FR_2.8). This allows them to track the progress of their orders accurately.
- Customers can fully rate and provide feedback about products (FR_2.9), enabling them to share their opinions and contribute to the improvement of the system.
- The system maintains a complete history of all customer orders (FR_3.0), allowing customers to review their past purchases accurately.
- Customers have complete flexibility to change their information at any time (FR_3.1) and can log out of the system whenever they want (FR_3.2).
- The system accurately calculates shipping fees according to the customer's area (FR_3.3), ensuring transparent and accurate pricing.
- Customers can enter the required details to complete their orders accurately, ensuring a seamless and complete checkout process (FR_3.4).

Overall, the system has achieved the completion of all specified requirements, providing users with a comprehensive and fully-functional augmented reality e-commerce android app.

8.2 Accuracy and Correctness

The system has been designed to meet the specified requirements (FR_1.0 to FR_3.4). The following paragraphs provide an evaluation of the system's compliance with each requirement, focusing on correctness and accuracy.

- The system ensures that the admin can securely log in using their credentials (FR_1.0). Similarly, customers are able to log in after successfully registering and verifying their email and password (FR_1.1, FR_1.9). The login functionality has been implemented correctly and accurately, ensuring that only authorized users can access the system.
- Customers are required to complete their registration process during their initial login (FR_1.2). This requirement has been implemented correctly, ensuring that customers provide all necessary information for their registration.
- The admin has the capability to add, delete, and update 3D products along with their descriptions and details (FR_1.3, FR_1.4, FR_1.5). This functionality has been implemented correctly and accurately, allowing the admin to manage the product inventory efficiently.
- Customers can view 3D products along with their descriptions and details (FR_1.6). The system presents the product information accurately, allowing customers to make informed decisions while browsing the available products.
- The admin can effectively manage order requests (FR_1.7). They are able to remove, cancel, and complete orders accurately, ensuring smooth order management within the system (FR_1.8).
- The admin can view feedback from customers (FR_2.1) and access information about all payments (FR_2.2) accurately. This functionality provides the admin with valuable insights into customer satisfaction and payment records.
- Customers can change their passwords (FR_2.3) and view all available product categories (FR_2.4) correctly. These features allow customers to personalize their account settings and navigate the product catalog efficiently.
- The system enables customers to try products online using augmented reality technology (FR_2.5) accurately. This functionality enhances the customer experience by providing a virtual trial experience.
- Customers can add products to their cart and view the cart accurately (FR_2.6, FR_2.7). This allows customers to easily select and purchase their desired products.
- Customers can view the status of their orders accurately, whether they are pending, completed, or dispatched (FR_2.8). This feature provides customers with real-time updates on their orders, ensuring transparency and reliability.
- Customers can rate and provide feedback about products accurately (FR_2.9). This functionality allows customers to share their opinions and experiences, contributing to the overall improvement of the system and product offerings.
- Customers can change their information accurately at any time (FR_3.1). This flexibility allows customers to update their details as needed.
- Both customers and admin users can log out of the system at any time accurately (FR_3.2). This functionality ensures that users can securely log out of the system, maintaining the privacy and security of their accounts.

- Customers can enter the required details accurately to complete their orders (FR_3.4). The system validates and processes the customer's order details correctly, facilitating a smooth and error-free checkout process.
- The system has successfully implemented and fulfilled all specified requirements (FR_1.0 to FR_3.4) with a high level of correctness and accuracy. The functionalities and features have been designed and implemented to meet the desired standards, ensuring that the system functions reliably and accurately, providing users with a seamless.

9. CONCLUSION

In conclusion, the development and implementation of an augmented reality (AR) e-commerce Android app offer tremendous opportunities to revolutionize the online shopping experience. Through the integration of AR technology, this app enables customers to visualize products in their real environment, enhancing their decision-making process and providing a more immersive and engaging shopping experience.

The AR e-commerce app offers numerous benefits for both customers and businesses. For customers, it eliminates the guesswork associated with online shopping by allowing them to virtually try on or visualize products, such as clothes, accessories, or furniture, before making a purchase. This not only enhances customer satisfaction but also reduces the likelihood of returns and improves overall customer loyalty.

From a business perspective, the app provides a competitive edge by offering an innovative and interactive shopping experience. By incorporating AR technology, businesses can differentiate themselves in the crowded e-commerce market and attract a larger customer base. The app also opens up opportunities for cross-selling and upselling, as customers can explore complementary products and accessories within the AR environment.

The successful development and implementation of an AR e-commerce app require careful consideration of various factors. This includes selecting the appropriate AR development platform, integrating seamless user interfaces and experiences, ensuring robust product catalogs, and maintaining a stable and reliable app performance. Furthermore, it is crucial to prioritize data security and privacy to instill trust and confidence in customers.

Looking ahead, the future of AR e-commerce apps holds great potential. As technology continues to advance, we can anticipate further enhancements, such as improved object recognition, more realistic visualizations, and personalized recommendations based on user preferences and behavior. Additionally, the integration of social sharing features can allow customers to seek feedback and recommendations from their social networks, further enhancing the app's value and engagement.

10. FUTURE WORK

In order to expand and enhance our business model, we can explore the possibility of establishing collaborative partnerships with different brands. This approach involves working closely with various brands to sell their products through our platform, thereby generating profits through mutually beneficial relationships. The following outlines the potential future work and strategies for executing this approach:

10.1 Brand Acquisition and Collaboration:

Identify potential brands that align with our platform's vision, target market, and customer preferences. Establish partnerships by approaching brands and showcasing the benefits of collaborating with us. This may involve negotiating commission structures, revenue-sharing models, or other mutually agreed-upon terms.

10.2 Product Catalog Expansion:

Develop a comprehensive product catalog by including a wide range of brands and their products. This would allow us to offer customers a diverse selection of high-quality products across different categories. Continuously update and refresh the catalog to ensure a current and attractive inventory.

10.3 Brand Promotion and Marketing:

Implement robust marketing strategies to promote the brands and products we are collaborating with. This can involve targeted advertising campaigns, social media marketing, influencer collaborations, and other promotional activities. The goal is to create brand awareness, increase customer engagement, and drive sales.

10.4 Sales and Conversion Optimization:

Implement strategies to optimize sales and conversion rates. This can include improving the user experience on our platform, streamlining the purchasing process, offering personalized recommendations, and leveraging data analytics to understand customer preferences and behavior. Continuously analyze and optimize these strategies to maximize revenue generation.

10.5 Performance Tracking and Analytics:

Utilize data analytics tools to track the performance of each brand and product. Monitor key metrics such as sales volume, conversion rates, customer feedback, and customer satisfaction. Regularly share performance reports with brand partners to foster transparency and enable data-driven decision-making.

10.6 Revenue Sharing and Profit Generation:

Establish a fair and transparent revenue-sharing model with brand partners. This can be based on a percentage of sales, fixed fees, or other mutually agreed-upon terms. Regularly analyze revenue generation and profitability to ensure sustainable growth and mutually beneficial partnerships.

10.7 Customer Satisfaction and Feedback:

Prioritize customer satisfaction by providing excellent customer service, efficient order fulfillment, and hassle-free return/exchange processes. Encourage customers to provide feedback on their experiences with different brands and products, allowing us to continuously improve and refine our offerings.

By pursuing these strategies, we can position ourselves as a trusted platform for both customers and brands, facilitating profitable collaborations while meeting the needs and preferences of our customer base. This future work requires careful planning, effective execution, and ongoing evaluation to ensure the success and sustainability of the collaborative brand partnerships and profit generation initiatives.

11. BIBLIOGRAPHY

These are the references which we use to collect some data.

11.1 Books

- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385.
- Billinghurst, M., & Kato, H. (Eds.). (2002). *Augmented reality: Placing artificial objects in real scenes*. CRC Press.
- Craig, A. B. (2013). *Understanding augmented reality: Concepts and applications*. Morgan Kaufmann.

11.2 Journals

- Azuma, R. T. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 34-47.
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1995). Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351, 282-292.
- Bimber, O., & Raskar, R. (2005). *Spatial augmented reality: Merging real and virtual worlds*. A K Peters/CRC Press.

11.3 Articles

- R. T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 34-47.
- DiVerdi, S., Bolas, M., & Riedl, M. (2013). Augmented reality for the sciences. *Communications of the ACM*, 56(12), 62-70.
- Kato, H., Billinghurst, M., & Poupyrev, I. (2000). ARToolKit: A tracking library for augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 48-56.

11.4 Research papers

- R., & Bishop, G. (1994). Improving static and dynamic registration in an optical see-through HMD. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 197-204.
- Fuchs, H., Livingston, M. A., & Raskar, R. (1999). Augmented reality visualization for laparoscopic surgery. *Proceedings of the 2nd international workshop on Augmented environments for medical imaging*, 59-68.
- Navab, N., Genc, Y., Geiger, B., Wang, C., & Hager, G. (2005). Augmented reality in computer-assisted interventions. *IEEE Computer Graphics and Applications*, 25(6), 48-57.

12. APPENDIX

12.1 Glossary

A. Term We Use

Admin: Refers to the administrator or system administrator who has privileged access and control over the augmented reality e-commerce Android app.

Customer: Denotes a user of the app who registers and interacts with the platform to browse and purchase products.

Augmented Reality (AR): A technology that overlays digital content, such as 3D models, onto the real-world environment, enhancing the user's perception and interaction with the physical world.

E-commerce: Electronic commerce, which involves buying and selling of goods or services over the internet.

SDK: Software Development Kit, a collection of software tools, libraries, and documentation that aids developers in building applications for a specific platform or technology.

B. User Roles

Admin: The admin is responsible for managing the overall system, including adding, updating, and deleting 3D products, managing orders, viewing customer feedback, and handling customer support.

Customer: The customer interacts with the app, views and searches for 3D products, adds products to the cart, places orders, views order status, provides feedback, and manages their profile information.

C. Development/Deployment Setup

Operating System: Windows, macOS, or Linux.

Integrated Development Environment (IDE): Android Studio.

Software Development Kit (SDK): Android SDK.

Programming Language: Java or Kotlin.

Version Control System: Git.

Database: Relational database management system (e.g., NoSQL database (e.g., firebase).

Augmented Reality SDK: ARCore for Android.

Testing Framework: JUnit or Espresso.

D. Use Case Diagram

The use case diagram illustrates the interactions between the actors and the system's functionality. It showcases the following use cases:

Admin-related use cases:

Login into System, add 3D Products, Delete 3D Products, Update 3D Products, Manage Order Requests, Remove Order, Cancel Order, Complete Order, View Feedback from Customer and View Payments

Customer-related use cases:

Register, Login, View 3D Products, Try Products with Augmented Reality, Add Products to Cart, View Cart, Place Order, View Order Status, Rate and Give Feedback, View Order History, Change Personal Information and Logout

F. Class Diagram

The class diagram represents the various classes and their relationships within the augmented reality e-commerce Android app. It encompasses entities such as User, Admin, Customer, Product, Order, Cart, Payment, Feedback, and others. The relationships between these entities depict the associations, dependencies, and inheritance within the system.

G. Database Schema

The database schema defines the structure of the database that stores important data for the app. It includes tables



Leaf Disease Detection

FYP-II (2019-2023)





Muhammad Habib
(F2019266362)

Hadayat Ullah
(F2019266317)



Abdullah Jabbar
(F2019266368)



Syed Ali Shah
(F2019266375)



Table of contents

01

Introduction

02

Motivations

03

Problem Statement

04

Solution

05

Implementation

06

References

Introduction

Agriculture plays a vital role in the economy of Pakistan. Cotton is one of Pakistan's main crops, and cotton-made products are exported to generate revenue. But the production was badly affected due to a lack of technical research and methods. The major influencing factor that affects cotton productivity is crop loss due to plant diseases, which reduces production by approximately 20 to 30%.





Motivations

As the economy of Pakistan is based on agriculture. Pakistan occupies the fourth position among the cotton growers of the world. In respect of raw cotton exports, Pakistan holds the third position and is the fourth in consumption (about 30 to 40 percent of its production). But unfortunately, due to a lack of use of technology and modern methods, 20% to 30% of production decreases. This issue can be solved by creating a model that can detect plant disease early on and take appropriate action.

Problem vs. Solution

Problem

Temperature fluctuations, diseases, and pests make the cotton plant vulnerable to a variety of disorder attacks. More than 10% of production losses occurred due to different cotton leaf diseases. Detecting these diseases withbare eyes increased the complexity of cotton crops' productivity, which decreased the accuracy of their identification.

Solution

The use of computer vision technology for disease identification is a meaningful way to resolve the previously mentioned issues, which cause many problems in the early identification of plant diseases. Computer vision can be used not only to obtain information about cotton diseases quickly and accurately but also to make accurate spraying decisions according to the severity of the disease and the area of cotton affected.

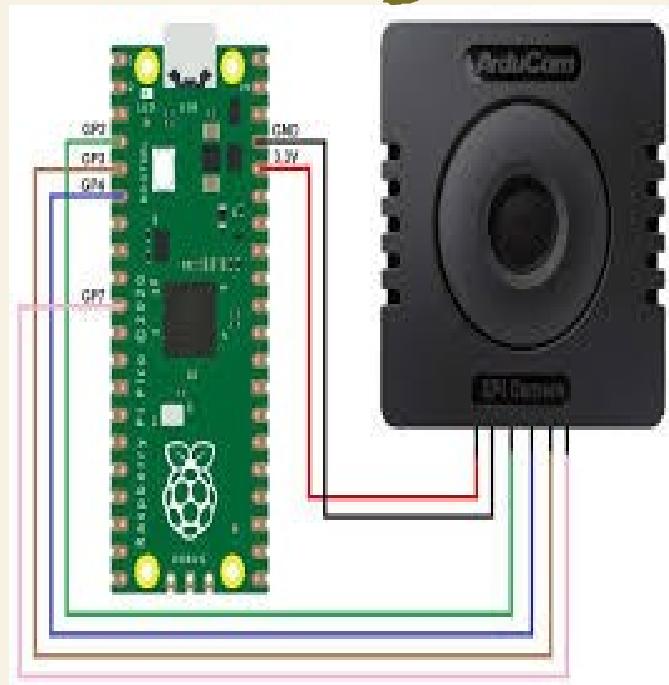
Implementation Details

To implement this solution using Raspberry Pi, we can use a Raspberry Pi camera module to capture digital images of cotton leaves. These images can be processed and classified in real-time using the trained CNN model, allowing for quick and efficient detection of leaf spot disease.



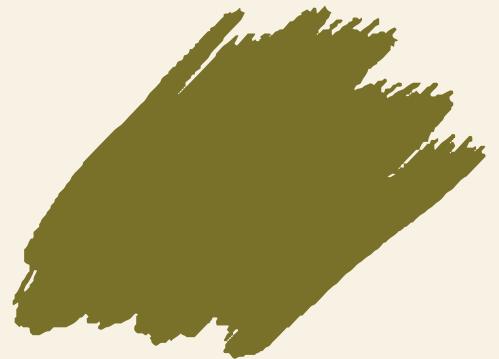
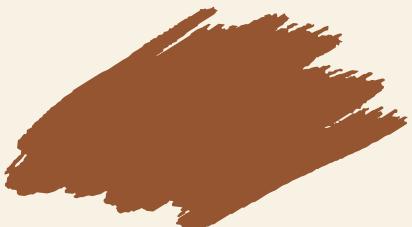
How to implement

1. Develop and train a machine learning model for cotton leaf disease detection using a dataset of labeled images.
2. Install the required software, libraries, and dependencies on the Raspberry Pi.
3. Connect a camera module to the Raspberry Pi and capture digital images of cotton leaves.
4. Process and analyze the captured images using the pre-trained machine learning model to detect leaf spot disease.
5. Display the results of the analysis, such as the percentage of healthy and diseased leaves, on a connected display.

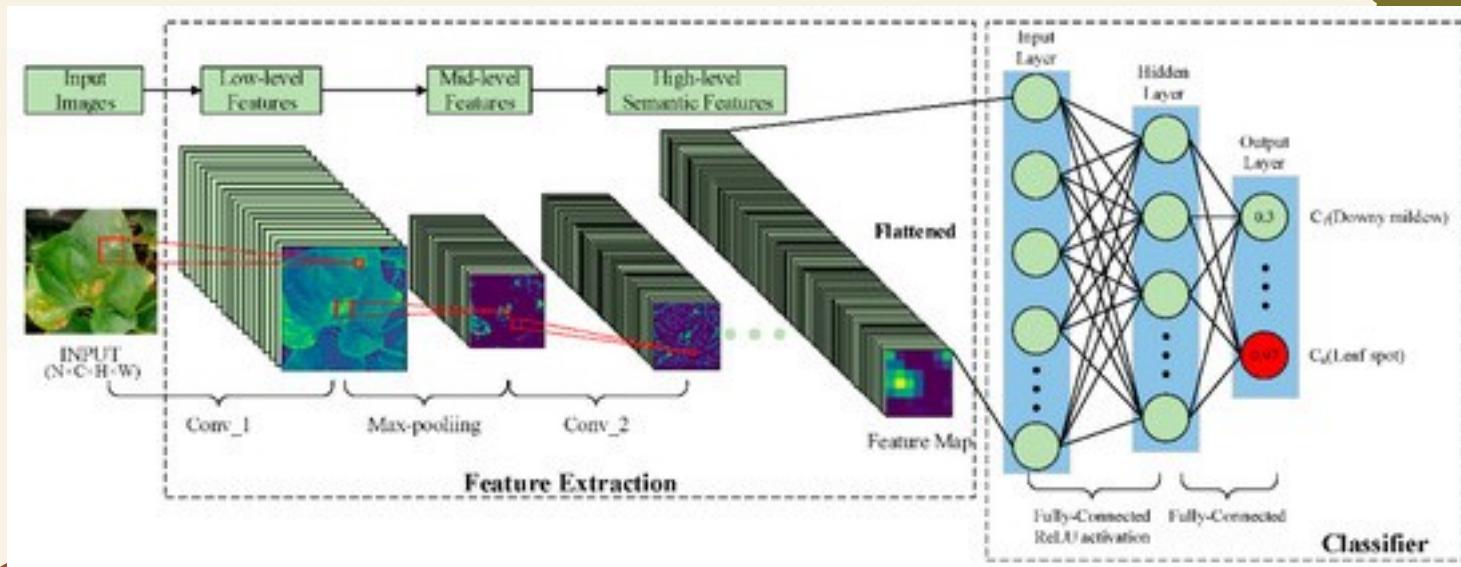


Implementation Details(CNN)

Our proposed solution for cotton leaf disease detection using machine learning involves training a convolutional neural network (CNN) model on a dataset of labeled images of healthy and diseased cotton leaves. The trained model can then be used to classify new images as either healthy or diseased with high accuracy.



CNN Architecture Diagram



Dataset Details

The dataset used for training and evaluating our machine-learning model consists of labeled images of healthy and diseased cotton leaves. The images were collected from various sources, such as cotton fields and research institutions, and were carefully labeled by experts to ensure accuracy.



Code snippets

```
# Build the CNN model
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(input_size[0], input_size[1], 3)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=validation_generator.n // batch_size
)

# Evaluate the model on the test set
_, accuracy = model.evaluate(test_generator)
print("Test accuracy: {:.2f}%".format(accuracy * 100))

# Make predictions on the test set
predictions = model.predict(test_generator)
predicted_classes = tf.argmax(predictions, axis=1)
```

Results/Outputs

```
Epoch 1/40
126/126 [=====] - 13s 102ms/step - loss: 1.4584 - accuracy: 0.4848 - val_loss: 1.3385 - val_accuracy: 0.4836
Epoch 2/40
126/126 [=====] - 13s 102ms/step - loss: 0.8828 - accuracy: 0.6742 - val_loss: 1.2889 - val_accuracy: 0.5987
Epoch 3/40
126/126 [=====] - 13s 105ms/step - loss: 0.7558 - accuracy: 0.7141 - val_loss: 1.5446 - val_accuracy: 0.5822
Epoch 4/40
126/126 [=====] - 13s 106ms/step - loss: 0.7037 - accuracy: 0.7491 - val_loss: 1.0741 - val_accuracy: 0.6382
Epoch 5/40
126/126 [=====] - 14s 108ms/step - loss: 0.6700 - accuracy: 0.7621 - val_loss: 1.0911 - val_accuracy: 0.5921
Epoch 6/40
126/126 [=====] - 14s 110ms/step - loss: 0.6212 - accuracy: 0.7721 - val_loss: 1.0015 - val_accuracy: 0.6250
Epoch 7/40
126/126 [=====] - 14s 109ms/step - loss: 0.5759 - accuracy: 0.7836 - val_loss: 0.8256 - val_accuracy: 0.6283
Epoch 8/40
126/126 [=====] - 14s 112ms/step - loss: 0.5833 - accuracy: 0.7911 - val_loss: 0.9859 - val_accuracy: 0.6414
Epoch 9/40
126/126 [=====] - 15s 121ms/step - loss: 0.5247 - accuracy: 0.8026 - val_loss: 0.9430 - val_accuracy: 0.6414
Epoch 10/40
126/126 [=====] - 16s 123ms/step - loss: 0.4897 - accuracy: 0.8216 - val_loss: 0.8091 - val_accuracy:
```

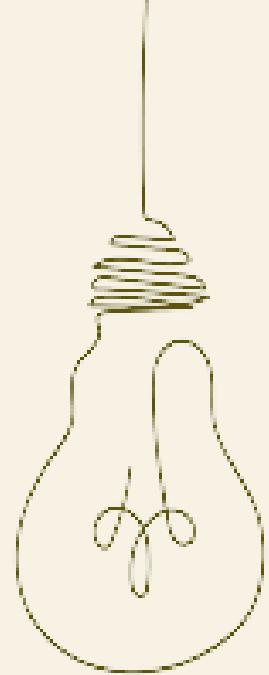
Results/Outputs

```
--> [1] >> 2023-05-05
```

Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Angular leaf spot - Solution: Use of resistant varieties
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Cercospora leaf spot - Solution: Removal of infected leaves
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Cercospora leaf spot - Solution: Removal of infected leaves
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Cercospora leaf spot - Solution: Removal of infected leaves
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Angular leaf spot - Solution: Use of resistant varieties
Predicted: Angular leaf spot - Solution: Use of resistant varieties
Predicted: Angular leaf spot - Solution: Use of resistant varieties
Predicted: Ramularia leaf spot - Solution: removal of infected plant debris
Predicted: Angular leaf spot - Solution: Use of resistant varieties
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides
Predicted: Alternaria leaf spot - Solution: Application of fungicides

Future work

- User Interface and Accessibility (Web/Mobile application)
- Real-time Monitoring and Alert System
- Disease Database Expansion
- Refinement of Disease Detection Algorithms



Conclusion

By using a Raspberry Pi to implement our solution, we can create a portable, low-cost, and efficient tool for detecting leaf spot disease in cotton. The success of our project depends on the quality of the dataset used, which was carefully collected and labeled to ensure accuracy and generalization. Through our project, we hope to contribute to the field of agriculture by providing a reliable and efficient tool for detecting cotton leaf diseases.

Failure

Unfortunately due to an electric short circuit!



Failure

Unfortunately due to an electric short circuit!



Failure

Unfortunately due to an electric short circuit!



References

- [1] Navin V. Dumare and Prof. S. S. Mungona, "Identification of Cotton Leaf Diseases Using Raspberry Pi," Semantic Scholar, 2017. [Online]. Available:
<https://www.semanticscholar.org/paper/Identification-of-Cotton-Leaf-Diseases-Using-Pi%3A-A-Dumare-Mungona/1321f146865156dbf8b152e7911122ad2f7e019c>
- [2] S. Tripathy, "Detection of Cotton Leaf Disease Using Image Processing Techniques," Journal of Physics: Conference Series, vol. 2062, p. 012009, 2021. doi: 10.1088/1742-6596/2062/1/012009.
<https://iopscience.iop.org/article/10.1088/1742-6596/2062/1/012009/pdf>
- [3] G. Rama Mohan Reddy, Nettam Sai Sumanth, and N. Sai Preetham Kumar, "Plant Leaf Disease Detection using CNN and Raspberry Pi," International Journal of Advance Scientific Research and Engineering Trends, vol. 6, no. 1, pp. 79-86, February 2020. ISSN: 2456-0774.
http://www.ijasret.com/VolumeArticles/FullTextPDF/343_5.PLANT_LEAF_DISEASE_DETECTION_G_CNN_AND_RASPBERRY_Pi.pdf



Thanks!