**Full Stack NFT Marketplace**

|  |
| --- |
|  |

**By:**

**Usman Ayub**

**27952**

**Muhammad Afzaal Hameed**

**23733**

**Mussab Saeed**

**27660**

**Supervised by:**

**Mr. Tajamul Shahzad**

**Faculty of Computing**

**Riphah International University, Islamabad**

**Spring/Fall 2024**

**A Dissertation Submitted To**

**Faculty of Computing,**

**Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Award of the Degree of**

**Bachelors of Science in Computer Science**

**Faculty of Computing**

**Riphah International University, Islamabad**

Date: [date of final presentation]

**Final Approval**

This is to certify that we have read the report submitted by ***Muhammad Afzaal Hameed (23733), Usman Ayyub (27952), Mussab Saeed (27660)***for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSCS).

**Committee:**

|  |  |
| --- | --- |
| **1** | Mr. Tajamul Shahzad  (Supervisor) |
|  |  |
| **2** | Dr. Muhammad Musharraf  (Head of Department) |

**Declaration**

We hereby declare that this document “**Full Stack NFT Marketplace**” neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **Mr. Tajamul Shahzad**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Uaman Ayub**

**27952**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Muhammad Afzaal Hameed**

**23733**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mussab Saeed**

**27660**

**Dedication**

Insert dedication Our final year project is dedicated to our parents, friends and teachers, whose love and support have been our pillars of strength. To our professors and especially supervisor"**Mr. Tajamul Shahzad**", your guidance has shaped our academic journey.

**Acknowledgement**

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We extend our heartfelt gratitude to our project supervisor “**Mr. Tajamul Shahzad”**, whose unwavering support, invaluable guidance, and continuous mentorship were indispensable to the successful completion of this project. Their dedication and commitment have been a driving force behind our work.

Furthermore, we want to say a big thank you to our family and friends. They have been our constant source of support and motivation, always encouraging us to do our best and be honest and hardworking.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Uaman Ayub**

**27952**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Muhammad Afzaal Hameed**

**23733**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mussab Saeed**

**27660**

Table of Contents

[List of Tables vii](#_Toc167027218)

[Abstract 1](#_Toc167027219)

Chapter 1………………………………………………………………………..

[Introduction 2](#_Toc167027220)

[1. Goals and Objectives 2](#_Goals_and_Objectives)

[1.1.1 Goals 2](#_Toc167027222)

[1.1.2 Objectives 3](#_Toc167027223)

[1.2 Scope of the Project 3](#_Toc167027224)

Chapter 2………………………………………………………………………………………..

[Literature Review 4](#_Toc167027225)

[2.1 Introduction 4](#_Toc167027226)

[2.2 Background and Problem Elaboration 4](#_Toc167027227)

[2.3 Detailed Literature Review 4](#_Toc167027228)

[2.4 Literature Review Summary Table 4](#_Toc167027232)

[2.5 Research Gap 5](#_Toc167027233)

[2.6 Problem Statement 5](#_Toc167027234)

Chapter 3

[Requirements and Design 6](#_Toc167027235)

[3.1 Requirements 6](#_Toc167027236)

3.1.1 [Functional Requirements 6](#_Toc167027237)

[3.1.2 Non-Functional Requirements 6](#_Toc167027238)

[3.1.3 Hardware and Software Requirements 6](#_Toc167027239)

[3.2 Proposed Methodology 6](#_Toc167027240)

[3.3 System Architecture 6](#_Toc167027241)

[3.4 Use Cases 6](#_Toc167027242)

[3.4.1 User/Customer 6](#_Toc167027243)

[3.5 Fully Dressed use cases……………………………………………………………...6](#_3.5_Fully_Dressed)

3.5.1 Register new user……………………………………………………………….

3.5.2 Login……………………………………………………………………………

3.5.3 Generate NFT listing……………………………………………………………

3.5.4 Search NFTs……………………………………………………………

3.5.5 Add to Favorites……………………………………………………………

3.5.6 Remove fromFavorites……………………………………………………………

3.5.7 Purchase NFT……………………………………………………………

3.5.8 Browse NFT Listings…………………………………………

3.5.9 Manage NFT Collection…………………………………………

3.5.10 Update NFT Details…………………………………………

3.5.11 Contact Us…………………………………………

3.5.12 Meta Wallet Connect…………………………………………

3.5.13 Viewing Popular NFTs…………………………………………

3.5.14 Placing a Bid on an NFT…………………………………………

3.5.15 Buying an NFT with MetaMask……………………………………

[1.11 Database Design *(Optional)* 7](#_Toc167027244)

[1.12 Class Diagram (*Optional)* 7](#_Toc167027245)

[1.13 Sequence diagram *(Optional)* 7](#_Toc167027246)

[1.14 Any Other Artifact… 7](#_Toc167027247)

[1.15 GUI Graphical User Interfaces (*Optional)* 7](#_Toc167027248)

[Implementation and Test Cases 8](#_Toc167027249)

[1.16 Implementation 8](#_Toc167027250)

1. [1.16.1 Implementation of First Component/Algorithm 8](#_Toc167027251)

[1.17 Test case Design and description 8](#_Toc167027252)

1. [1.17.1 Sample Test case No.1 8](#_Toc167027253)
2. [1.17.2 Sample Test case No.2 8](#_Toc167027254)

[1.18 Test Metrics 9](#_Toc167027255)

1. [1.18.1 Sample Test case Matric.No.1 9](#_Toc167027256)
2. [1.18.2 Sample Test case Metric.No.2 9](#_Toc167027257)
3. [1.18.3 Sample Test case Metric.No.3 9](#_Toc167027258)

[Experimental Results and Analysis 10](#_Toc167027259)

[Conclusion and Future Directions 11](#_Toc167027260)

[References 12](#_Toc167027261)

[Appendix 13](#_Toc167027262)

[Appendix A: Guidelines 13](#_Toc167027263)

[Appendix B: Heading of Sample Appendix B 13](#_Toc167027264)

[Formatting Guidelines 14](#_Toc167027265)

[Chapter 1: Heading 1 14](#_Toc167027266)

[1.1 Heading 2 14](#_Toc167027267)

1. [1.1.1 Heading 3 14](#_Toc167027268)

[Tables and Figures 15](#_Toc167027269)

[Equations 16](#_Toc167027270)

[Header/Footer 16](#_Toc167027271)

[Other Formatting Guidelines 16](#_Toc167027272)

[References 16](#_Toc167027273)

# 

List of Tables

[Table 1: Literature Review 6](#_Toc165128403)

[Table 2: Market Survey 7](#_Toc165128404)

[Table 3: This is Sample table caption 40](#_Toc165128405)

[Table 4: This is Sample table caption 40](#_Toc165128406)

**List of figures**

[Figure 1: Architecture Diagram 12](#_Toc165132973)

[Figure 2: Farmer Use Case 13](#_Toc165132974)

[Figure 3: Admin Use Case 14](#_Toc165132975)

[Figure 4: Database Design 28](#_Toc165132976)

[Figure 5: Methodology Diagram 29](#_Toc165132977)

[Figure 6: List of Styles 40](file:///C:\Users\DELL\Desktop\Tomato%20Care_CS_Mr.Muhammad%20Usman%20Karim.docx#_Toc165132978)

[Figure 7: IEEE Reference style 41](#_Toc165132979)

# Abstract

The project, titled **“The Full Stack NFT (Non-Fungible Token) Marketplace”** aims to create a comprehensive platform for trading digital assets, offering a seamless experience for creators and collectors alike. Built on a foundation of **Next.js, Solidity, Hardhat, Node.js, and MetaMask,** the marketplace provides a robust ecosystem for minting, buying, and selling NFTs. The platform prioritizes security, scalability, and user-friendliness, ensuring that users can confidently engage with the marketplace. Through innovative features and a focus on emerging blockchain technologies, the project seeks to revolutionize the way digital assets are traded, opening up new avenues for creators to monetize their work and for collectors to discover and acquire unique pieces. With a commitment to excellence and a vision for the future of digital asset trading, the Full Stack NFT Marketplace project represents a significant step forward in the evolution of blockchain-based marketplaces.

# 

# Introduction

Non-fungible tokens (NFTs), a product of blockchain technology, have totally changed the way that ownership was predicted in the digital realm. NFTs are clear digital assets that substitute for the ownership of a specific device or work of material, along with music, art, collectibles, and more. As they are kept on a blockchain, their ownership history, scarcity, and legitimacy are insured.

We are designing a "Full Stack NFT Marketplace" include in our Final Year Project (FYP) which will further utilize in blockchain technology to make it simple to create, purchase, and sell NFTs. With the help of this marketplace, artists will be able to mint their digital works as NFTs and market and sell their masterpieces to a worldwide audience of collectors and enthusiasts.

Various kinds of technologies, including Next.js for the front end, Solidity for building smart settelments, Hardhat for Ethereum development, Node.js for backend development, and MetaMask for wallet integration, will be used in the building of the marketplace. With the use of these technologies, we will be able to develop an NFT platform that is secured, inflatable, and simple to use for both manufacturer and purchaser.

With the help of this project, we desired to examine how NFTs might convert digital ownership and give a platform that facilitate workers to review their work in advanced and modern ways.

# Goals and Objectives

# Goals

* Build a customer friendly platforms for selling and purchasing NFTs
* Confirm the safe transactions by using blockchain technology
* Issue a broad area of NFT group for the customers to explore
* Apply a filter system and strong search for comfortable browsing
* Build coordination with creators and experts to encourage the platform
* Allow customers to design and organize the contributions of NFT
* Must give the access to retailers for checking their NFTs performance
* Provide a smooth incorporation with famous wallets of cryptocurrency
* Apply a receptive draft for greatest customer experience over devices
* Make sure the acceptance with official and regulatory demands linked with NFTs

# Objectives

* Build a front end interface for buying and browsing NFTs
* Apply unique agreements on the blockchain to control the transactions of NFTs
* Develop a customer verification system for accurate registration and login
* Integrate payment channels for cryptocurrencies to purchase NFTs
* Develop a back end system to control customer account and transactions
* Apply the filter and search utility for NFTs postings
* Provide a dashboard where consumers may oversee their transactions and NFT collections.
* Conduct extensive testing on the platform to guarantee security and functionality.
* Improve the platform for better performance and scalability
* Develop a marketing initiative to publicize the platform and engage the users

## Scope of the Project

# 

# Literature Review

## Introduction

Non-fungible tokens (NFTs) are a revolutionary development in the quickly developing field of blockchain technology. They allow for the digital depiction of distinctive assets and completely alter ideas about provenance and ownership. A examination of the literature indicates that NFTs are becoming more and more popular in a variety of industries, such as art, gaming, and collectibles. Talks about how these technologies may affect copyright, authenticity verification, and applications related to decentralized finance (DeFi) are particularly noteworthy. Developing a full-stack NFT marketplace requires a thorough understanding of critical technologies, which must come first. A dynamic and user-friendly interface is ensured by using frontend frameworks like Next.js and React.js, while the backend is powered by Node.js for scalability and reliable operation. Solidity makes it easier to create smart contracts, while resources like Hardhat offer a more efficient Ethereum development environment. While using libraries like OpenZeppelin guarantees the dependability and security of smart contracts, integration with cryptocurrency wallets like MetaMask improves user accessibility and security. This technological synthesis makes it possible to build a unified and effective platform for the smooth exchange of digital assets within the developing NFT ecosystem.

## Background and Problem Elaboration

## Detailed Literature Review

## 

## Literature Review Summary Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Title | Author | Year | Source | Summary | Reference |
| 1. | NFT MARKET PLACE | Chirag Chaudhari, Kunal Girme | 2023 | IRJMETS | The paper explains how NFT marketplaces use blockchain to ensure the authenticity and ownership of digital assets, offering secure transactions and monetization opportunities for creators, while also noting challenges like fraud and scalability. | https://www.irjmets.com/uploadedfiles/paper/issue\_10\_october\_2023/45692/final/fin\_irjmets1698589914.pdf |
| 2 | NFT Marketplace Design and Market Intelligence | Pavel Kireyev | 2022 | Papers SSRN | This paper likely examines the strategic design and market analysis essential for NFT marketplace success. It probably delves into user-friendly interfaces, blockchain integration, and market trends, offering valuable insights for developers and analysts. | https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=4002303 |
| 3 | Non-fungible Token(NFT) Markets on the Ethereum Blockchain | Lennart Ante | 2021 | Tandfonline | This study looks at the NFT market from 2017 to 2021, focusing on 14 major submarkets on Ethereum. It finds that while transaction numbers have dropped, traded value has risen. The research shows interconnectedness among submarkets, indicating ongoing evolution and potential inefficiencies in NFT markets. | https://www.tandfonline.com/doi/abs/10.1080/10438599.2022.2119564 |
| 4 | NFT Marketplace | Piyush Batra | 2023 | Arvix | This project creates a dApp for secure NFT management, combining blockchain and deep learning for features like wallet connections, NFT generation, and marketplace, showcasing their potential in digital asset management. | https://arxiv.org/abs/2304.10632 |

## 2.5 Research Gap

## 2.6 Problem Statement

**2.6.1 Security Breaches**:

* + Unapproved access to customer accounts results of stealing the NFTs or particular information
  + Smart agreements susceptibility results of manipulation of loopholes.
    1. **Scalability Issues**:

● The incapacity to manage a substantial amount of transactions during peak hours, leading to sluggish system performance or failures.   
● Difficulty in expanding the platform's user base and meeting the rising demand for NFTs.

* + 1. **User Experience Challenges**:
  + Complex transactions and confusing customer interface results of discouragement and perplexing among users
  + Unavailability of mobile upgradation, makes the customer unsuitable to approach the platforms on tablets and smart phones
    1. **Legal and Regulatory Compliance**:
  + Unreliability and uncertainty with regard to NFTs legal status and the essential management for controlling an NFT marketplace
  + Valid legal argument from ownership disputes, copyright infringement, and non compliance with financial rules
    1. **Market Manipulation and Fraud**:
  + Scheme to tackle NFT prices with the help of artificially inflating demand and fake bids
  + Faked or copied NFTs being vending on the marketplace, cheat the customer and weaken the trust in platforms
    1. **Payment and Transaction Problems**:
* Delaying and failures in payments leads to ineffective transactions and funds loss
* Insufficient support for various payment process or currencies, difficulty for users to approach in different regions

# 

# Chapter 3

# Requirements and Design

# 3.1 Requirements

For an NFT marketplace, requirement engineering is a thorough process of determining, recording, verifying, and overseeing the requirements that specify the capabilities, limitations, and performance of the system. The first step in this process is to interact with stakeholders—such as artists, collectors, investors, and platform operators—to learn about their requirements and expectations. A wide range of elements, including the development of NFTs, token standards, smart contract functionality, marketplace user interfaces, search and discovery systems, payment gateways, and security measures, may be included in these requirements. Scalability, compatibility with other blockchains, regulatory compliance, and data protection concerns must also be taken into account. Flexibility and adaptability are critical in the continuously evolving NFT arena, necessitating a detailed examination of market trends and user input. To provide as the basis for the development process, needs must be obtained, prioritized, verified, and clearly and concisely documented. An NFT marketplace's requirement engineering is an iterative process that makes sure the finished solution satisfies stakeholders' expectations and fits into the dynamic NFT ecosystem.

### Functional Requirements

### Non-Functional Requirements

### Hardware and Software Requirements

## 3.2 Proposed Methodology

## 3.3 System Architecture

## Use Case

# 3.5 Fully Dressed use cases

# UC ID: UC001

|  |  |
| --- | --- |
| **UC Name**: | **Registration of User** |
| **Primary Actor**: | New User |
| **Stakeholders and Interests**: | Users (who wants to connect with marketplace), Admin (involved in controlling new registrations) |
| **Pre-condition**: | User has approach to the registration page. |
| **Post condition**: | Successfully created User account. |
| **Success Guarantee**: | A confirmation mail must send to User for login account. |
| **Main Flow**: | * + 1. User cover the registration Page     2. User enter all the required information or data     3. User consent the registration form     4. System support the information     5. If required, system make a new user account     6. User secure a verification mail |
| **Alternative Flow**: | The system shows an error message, if the given email is already registered. |

* + 1. **UC ID: UC002**

|  |  |
| --- | --- |
| **UC Name**: | User Login |
| **Primary Actor**: | Registered User |
| **Stakeholders and Interests**: | Users (want to access their account), Admin (Security concerns) |
| **Pre-condition**: | registered account of User. |
| **Post-condition**: | User is logged into the system. |
| **Success Guarantee**: | User is assigned to the dashboard leads to successful login |
| **Main Flow**: | * + 1. User move to the login page.     2. User gives their email and password.     3. User presents the login form.     4. System supports the credentials.     5. If correct, user is logged in and turns over to the dashboard. |
| **Alternative Flow**: | System shows an error message, if credentials are invalid. |

* + 1. **UC ID: UC003**

|  |  |
| --- | --- |
| **UC Name:** | Generate NFT listing |
| **Primary Actor:** | Seller |
| **Stakeholders and Interests:** | Sellers, Buyers, Platform Administrators |
| **Pre-condition:** | Seller is validated and has digital resources for listing. |
| **Post-condition:** | Create NFT listing which is in view on the marketplace. |
| **Success Guarantee:** | NFT is successfully listed with correct metadata. |
| **Main Flow:** | * + 1. Retailer logs into the platform.     2. Seller starts the process to make a new NFT listing.     3. Seller uploads digital resources and Seller uploads digital asset and fill up in metadata details.     4. Seller fixes prices and other listing guidelines.     5. Seller finds out and submits the listing. |
| **Alternative Flow:** | None. |

* + 1. **UC ID: UC004**

|  |  |
| --- | --- |
| **UC Name** | : Search NFTs |
| **Primary Actor**: | User |
| **Stakeholders and Interests**: | Users (need to search the particular NFT), Artists (Want their NFTs to be discoverable) |
| **Pre-condition**: | User is on the NFT marketplace page. |
| **Post-condition** | Specific search results are presented to User. |
| **Success Guarantee** | User observed the specific NFTs |
| **Main Flow**: | * + 1. User enters a search query.     2. System filters NFTs based on the query.     3. System shows the filtered results to the User. |
| **Alternative Flow**: | None. |

* + 1. **UC ID: UC005**

|  |  |
| --- | --- |
| **UC Name**: | Add to Favorites |
| **Primary Actor**: | User |
| **Stakeholders and Interests**: | Users (need a bookmark NFTs), Artists (Want their NFTs to be favorite) |
| **Pre-condition**: | User is logged in and looking an NFT. |
| **Success Guarantee**: | User gets a verification of successful addition. |
| **Post-condition**: | NFT is added to the favorite list of User. |
| **Main Flow**: | * 1. User clicks on the "Add to Favorites" button.   2. System adds the NFT to the user's favorites list. |
| **Alternative Flow**: | * 1. If the NFT is already added in user’s favorite list, the system shows a message specifying the same. |

* + 1. **UC ID: UC006**

|  |  |
| --- | --- |
| **UC Name** | Remove from Favorites |
| **Primary Actor**: | User |
| **Stakeholders and Interests** | Users (need to hide the NFT from favorites), Artists (convert their NFT into un-favorite). |
| **Pre-condition**: | User is logged in and looking their favorites list. |
| **Post-condition**: | NFT is hiding from the favorite list of User. |
| **Success Guarantee**: | User receives a confirmation of successful removal. |
| **Main Flow**: | User taps on the "Remove from Favorites" button next to an NFT.  System removes the NFT from the user's favorites list. |
| **Alternative Flow**: | * 1. If the NFT is not in the user's favorites, the system displays a message indicating the same. |

* + 1. **UC ID: UC007**

|  |  |
| --- | --- |
| **UC Name:** | Purchase NFT |
| **Primary Actor:** | Buyer |
| **Stakeholders and Interests:** | Buyers, Sellers, Platform Administrators, Payment Processor. |
| **Pre-condition:** | Buyer is validated and has enough balance. |
| **Post-condition:** | Buyer keeps the purchased NFT. |
| **Success Guarantee:** | Buyer successfully completes the purchase transaction. |
| **Main Flow:** | * + 1. Buyer searching the marketplace and select NFT for purchase.     2. Buyer analyzes the listing details and settled the purchase.     3. Buyer chooses the payment method and finished the transactions.     4. The ownership of NFT is shifted to the buyer. |
| **Alternative Flow:** | Failure in payment, NFT already sold out. |

* + 1. **UC ID: UC008**

|  |  |
| --- | --- |
| **UC Name:** | Browse NFT Listings |
| **Primary Actor:** | User (Buyer/Seller) |
| **Stakeholders and Interests:** | Buyers, Sellers, Platform Administrators |
| **Pre-condition:** | User is validated and on the marketplace homepage. |
| **Post-condition:** | User has looked all the available NFT listings. |
| **Success Guarantee** | User successfully cut across through listings. |
| **Main Flow:** | * + 1. User accesses the NFT marketplace.     2. User exploring through present NFT listing.     3. User look details of specific selected listings. |
| **Alternative Flow:** | None |

* + 1. **UC ID: UC009**

|  |  |
| --- | --- |
| **UC Name:** | Manage NFT Collection |
| **Primary Actor:** | User (Buyer/Seller) |
| **Stakeholders and Interests:** | Buyers, Sellers |
| **Pre-condition:** | User is validated and has uploaded NFTs. |
| **Post-condition:** | User well ordered and control their NFT collection. |
| **Success Guarantee:** | User successfully fulfills activity on their NFTs. |
| **Main Flow:** | * + 1. User accesses their profile/dashboard.     2. User covers the collection section of NFT.     3. User arranges, improve metadata, or remove the NFT as desired. |
| **Alternative Flow:** | None. |

* + 1. **UC ID: UC010**

|  |  |
| --- | --- |
| **UC Name**: | Update NFT Details |
| **Primary Actor**: | Artist |
| **Stakeholders and Interests**: | Artists (need to improve their NFT information), Buyers (Want correct information ) |
| **Pre-condition**: | Artist is logged in and owns the NFT. |
| **Post-condition**: | NFT details are modernized |
| **Success Guarantee**: | Updated details are displayed to users |
| **Main Flow**: | * + 1. Artist covers the improved NFT page.     2. Artist selects the NFT to update.     3. Artist improves the details (name, description, and price).     4. Artist submits the update form.     5. System upgrades the NFT details. |
| **Alternative Flow**: | * + 1. The system rejects the updates, if the artists no longer maintain the NFT. |

* + 1. **UC ID: UC011**

|  |  |
| --- | --- |
| **UC Name**: | Contact Us |
| **Primary Actor**: | User |
| **Stakeholders and Interests** | Users (Want to contact the supporter), Admin (Want to direct the support requests) |
| **Pre-condition**: | User is on the "Contact Us" page. |
| **Post-condition** | User's message is submitted to the support team. |
| **Success Guarantee**: | User get a verification email with a support ticket number. |
| **Main Flow**: | 1. User navigates to the "Contact Us" page. 2. User fills in the contact form with their name, email, and message. 3. User submits the contact form. 4. System validates the form data. 5. If valid, system sends a confirmation email to the user and assigns a support ticket. |
| **Alternative Flow**: | If the form data is invalid, the system displays an error message and prompts the user to correct it. |

* + 1. **UC ID: UC012**

|  |  |
| --- | --- |
| **UC Name**: | Meta Wallet Connect |
| **Primary Actor**: | User |
| **Stakeholders and Interests**: | Users (Need to join their Meta Wallet), Admin (Want to control wallet connections) |
| **Pre-condition** | User is logged in and wants to link their Meta Wallet. |
| **Post-condition**: | User's Meta Wallet is successfully linked. |
| **Success Guarantee**: | User can access Meta Wallet features on the marketplace. |
| **Main Flow**: | * + 1. User negotiates to the Meta Wallet connection page.     2. User taps on the "Connect Meta Wallet" button.     3. System produces the user access to select their Meta Wallet provider.     4. User selects their Meta Wallet provider and accepts the connection.     5. System linked with Meta Wallet of User to their marketplace account. |
| **Alternative Flow**: | * + 1. The system must show a message of cancellation, if the user cancels the link task. |

* + 1. **UC ID: UC013**

|  |  |
| --- | --- |
| **Use Case Name:** | Viewing Popular NFTs |
| **Primary Actor:** | User/Collector |
| **Stakeholders and Interests:** | User/Collector: Interested in viewing and discovering famous NFTs.  NFT Creators: Great concern in their NFTs being featured as famous, attaining more clearness and developing sales. |
| **Pre-condition:** | The user is logged into the NFT marketplace. |
| **Post-condition:** | The user has checked famous NFTs and may select to connect with or buy them. |
| **Success Guarantee:** | The user can access a list of famous NFTs and check their details and media. |
| **Main Flow** | 1. User negotiates to the “Popular NFTs” section of the marketplace. 2. System recovers a list of famous NFTs based on criteria such as views, likes, or sales volume. 3. System shows the list of famous NFTs to the user. 4. User selects a specific NFT to view its details. 5. System shows the details of specific NFT, such as title, description, media, creator, and present price. 6. User can linked with the NFT, such as view it, liking it, add it into favorite list, or buying it. |
| **Alternative Flow** | If there is no famous NFT available at that moment, the system tell the user and suggest searching other NFTs group. |

* + 1. **UC ID: UC014**

|  |  |
| --- | --- |
| **Use Case Name:** | Placing a Bid on an NFT |
| **Primary Actor:** | Collector/Buyer |
| **Stakeholders and Interests:** | Collector/Buyer: Concerned in buying NFT through bidding.  NFT Creator/Artist: Interested in accepting bids and selling their NFTs.  Marketplace Platform: Interested in fostering the bidding process. |
| **Pre-condition** | The purchaser must register on the platform and logged in. |
| **Post-condition:** | The bid is successfully set on the NFT, and the purchaser receives a verification of bid. |
| **Success Guarantee:** | The bid is successfully put on the NFT, and the buyer can track the level of their bid. |
| **Main Flow** | 1. Purchaser navigates to the specific NFT listing and selects the option to place a bid. 2. System moves the buyer to enter their bid amount. 3. Collector/Purchaser enters the bid amount and decides the bid. 4. System records the bid and updates the bid status for the NFT listing.   NFT artist/creator is informed for the new bid. |
| **Alternative Flow** | If the amount of bid is less than that of present highest bid, the system informed the collector/purchaser that their bid was not successful and produces them to place a greater bid if needed. |

* + 1. **UC ID: UC015**

|  |  |
| --- | --- |
| **Use Case Name:** | Buying an NFT with MetaMask |
| **Primary Actor:** | Buyer/Collector |
| **Stakeholders and Interests:** | Buyer/Collector: By using MetaMask, interesting in buying NFTs safely.  Seller: Interested in selling their NFTs to buyers.  Marketplace Platform: Interested in fostering safe transactions between purchaser and sellers. |
| **Pre-condition:** | Buyer/Collector has a MetaMask wallet set up and linked to the marketplace.  The preferred NFT is listed for sale on the marketplace. |
| **Post-condition:** | The MetaMask wallet of buyer is removed with the purchase amount, and NFT is shifted to the buyer’s wallet**.** |
| **Success Guarantee:** | By using MetaMask, the buyer successfully buys the NFT, and the NFT is shifted to their wallet. |
| **Main Flow** | 1. Collector/Buyer chooses the preferred NFT for purchase. 2. System shows the buy details, including seller information and price. 3. Collector/Buyer chooses the option to purchase and confirms the purchase. 4. System makes the buyer to link their MetaMask wallet if no already linked. 5. Collector/Buyer linked their MetaMask wallet and confirms the transaction. 6. System checks the amount of purchase from the buyer’s MetaMask wallet and shifted the NFT to the buyer’s wallet.   System improves the record of owner for the NFT. |
| **Alternative Flow** | If the MetaMask wallet of purchaser is not sufficient, the system mentions the buyer to refill their wallet before go on to the purchase. |

## Database Design *(Optional)*

## Class Diagram (*Optional)*

## Sequence diagram *(Optional)*

## Any Other Artifact…

## GUI Graphical User Interfaces (*Optional)*

This section should give the GUI dumps of each screen, with reference to the user. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

# Implementation and Test Cases

**For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.**

## Implementation

Whatever implementation that you have done so far, please elaborate here.

Give clear details of the algorithms that were implemented along with the platform and the APIs which were used. **For FYP-1, this chapter can be changed to description of prototype developed.**

### Implementation of First Component/Algorithm

Write implementation of first component of your system here.

## **Test case Design and description**

**This section will be added in FYP-II.** Summarize the common attributes of test cases. This may include input constraints that must be true for every input in the set of associated test cases, any shared environmental needs, any shared special procedural requirements, and any shared case dependencies. The following scheme is recommended for describing test cases in detail.

### Sample Test case No.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<Software component Name>** | | | | | |
| **<Reference>** | | | | | |
| Test Case ID: | | *Reference Number* | Test Date: | | *Date* |
| Test case Version: | | *Version number* | Use Case Reference(s): | | *Relation to use cases* |
| Revision History: | | *Refer to previous test case identity (if any)* | | | |
| Objective | | *Need and scope of the testing* | | | |
| Product/Ver/Module: | | *Refer to overall system being built and the place of this test case in it.* | | | |
| Environment: | | *Necessary and desired properties of the test environment. (hardware/software)* | | | |
| Assumptions: | | *Assumptions that might affect the testing process.* | | | |
| Pre-Requisite: | | *Necessary condition that needs to be fulfilled prior to the test case.* | | | |
| Step No. | Execution description | | | Procedure result | |
|  | *Events being tested.* | | | *Mention software response.* | |
| Comments: | | | | | |
| *Passed* *Failed* *Not Executed* | | | | | |

### Sample Test case No.2

.

.

.

## Test Metrics

Summarize here the common ground of attributes of test case metrics.

### Sample Test case Matric.No.1

|  |  |
| --- | --- |
| Metric: | Purpose |
| Number of Test Cases: | Total number of test cases that you have developed for your system. |
| Number of Test Cases Passed: | The number of test cases that successfully passed |
| Number of Test Cases Failed: | The number of test cases that failed |
| Test Case Defect Density: | (No of test cases failed \* 100)  No of test cases executed |
| Test Case Effectiveness: | No of defects detected using test cases \*100  Total number of defects detected |
| Traceability Matrix: | Traceability is the ability to determine that each feature has a source in requirements and each requirement has a corresponding implemented feature. |

### Sample Test case Metric.No.2

### Sample Test case Metric.No.3

.

.

.

# Experimental Results and Analysis

**This chapter will be added in FYP-II.** Give proper analysis and discussion of experimental results (in plain English text) along with tables of results. **For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.**

# Conclusion and Future Directions

**This chapter is mandatory.** Give conclusions and summary of the work done. What were your findings and what were the results? Discuss in detail whether the scope of your project was entirely covered or not and whether the objectives of the project were met or not. What challenges did you face and what has been left out and why?

Sum up all the conclusions of all the chapters here to make a conclusion chapter. Do not repeat any text, just summarize it in different words.

Give recommendations for future work also. How your project can be further enhanced or improved? Future recommendations if someone wants to work on it. **For FYP-1 it is mandatory to list down a plan of the work to be done for FYP-2.**

# References

List all important sources of information which have been consulted for this project

# Appendix

## Appendix A: Guidelines

This section should include all supporting information from the project that was not included in the body of the report.  You should include surveys, complex statistical calculations, certain detailed tables and other such information in an appendix.  The information presented in this section is important to support the work presented in the body of the report but would make it more difficult to read and understand if presented within the body of the report.

Cite the appendix items in the report narrative (write "see Appendix A") and organize appendices (e.g., Appendix A, Appendix B,

Any tables, figures, forms, or other materials that are not totally central to the analysis but that need to be included are placed in the Appendix.

## Appendix B: Heading of Sample Appendix B

Following is a sample code with “code” style format.

Void SampleFunction(){

Print “Hello World.”;

}

# Formatting Guidelines

This document also serves as style guide for final year project reports. In order to give a similar high-quality appearance to all final year software project reports this template uses a collection of predefined Microsoft Word formatting styles. **These styles should be used without modification or replacement.** Font in the document is ***“Time New Roman”.*** This template provides following styles:

* **Title** – the main title style
* **Title2** – the subtitle style
* **Body Text** – style for paragraphs
* **Caption** – the style for a figure or table caption
* **Table Description** – the style for description of table, it must be added after caption.
* **Figure Description** - the style for description of figure, it must be added after caption.
* **Code** – the style for program source code

**int x** = 10; // Writing important code

* **Table Header Row** – Style for the header row of table
* **Table Grid** – the style for the data rows in the tables
* **Reference** – The style for references
* **Bullets** – The style for the bullet lists
* **Numbered** **List**– Style for numbered lists

All Heading styles with different level numbers are listed below.

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Heading 7

Heading 8

Heading 9

## Tables and Figures

Tables and figures should be centered horizontally. The caption button should be used to insert caption for both the figures and tables. All figures and tables must be numbered properly. Always refer to tables and figures according to their numbers. A table or figure can be cited as follows: ‘see Table1’ or ‘as shown in Table1’. The caption of table should be centered above the table and figure caption should be centered below the figure. Place the tables/figures close to their reference. Use “Table Header Row” and ‘Table Grid’ style for table’s header and data rows respectively. It is compulsory to provide brief description of table/figure after its caption. Styles for table and figure descriptions are “Table Description” and “Figure Description” respectively.

Press Ctrl+Shift+S to see list of styles mentioned above. Figure 1 shows the Apply Style window displaying the list of styles. Select any text then press Ctrl+Shift+S, the Apply Style window will show you the current style applied on that text and if required, you can change the style by selecting any other style from the “Style Name” dropdown.

This is brief description of above figure.

Figure 1: List of Styles

Table 1: This is Sample table caption

This is brief description of following Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Header row | Header row | Header row | Header row |
| Row1 col1 | Row1 col2 | Row1 col3 | Row1 col4 |
| Row2 col1 | Row2 col2 | Row2 col3 | Row2 col4 |

Table 2: This is Sample table caption

This is brief description of following Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Header row | Header row | Header row | Header row |
| Row1 col1 | Row1 col2 | Row1 col3 | Row1 col4 |
| Row2 col1 | Row2 col2 | Row2 col3 | Row2 col4 |

## Equations

Use equation editor to write equations in this report. Use last button of the custom tool bar to invoke equation editor. Similar to tables and figures, equations should also be aligned centered horizontally. Number all equations and insert them in parenthesis. Below is a sample equation and its reference number. An equation can be referenced like this: ‘it is clear from (1)’.

 (1)

## Header/Footer

Notice the headers in this document, before Introduction (i.e. the main content of this document) page numbers are in roman numerals. The page numbers of the actual content start with Arabic numerals i.e. 1, 2, 3 and so on. All of the **odd numbered pages** contain title of your project while the **even numbered pages** contain the section heading (i.e. chapter’s name) in the headers.

## Other Formatting Guidelines

* Keep 2-4 GUIs in one page. Consume as much space as possible. Do not leave most of page blank unnecessarily.
* Do not break tables (or use cases) in multiple pages unless the table is too large to fit in one page.
* Re-arrange the content i.e., text, images, and tables properly to meet above two guidelines.

## References

Always refer to the source of information by inserting the reference number in square brackets like this [5]. The reference numbers can either be added at the end of the sentence or within the sentence without changing the punctuation of sentence. A reference can also be cited as follows: ‘as Ruskey [2] mentioned’. List each source only once on your reference page.



Figure 2: IEEE Reference style

This figure represents the styling information for adding references in IEEE format

**Following is a list of sample reference for various typed of sources in IEEE format.**

1. P.M. Morse and H. Feshback, *Methods* of *Theoretical Physics*. New York: McGraw Hill, 1953. **//Format for Book**
2. S.K. Kenue and J.F. Greenleaf, “Limited angle multifrequency diffiaction tomography,” *IEEE Trans. Sonics Ultrason*., vol. SU-29, no. 6, pp. 213-2 17, July 1982. **//Format for Journal Article**
3. B. Tsikos, “Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation,” Ph.D. dissertation, Univ. of Pennsylvania, BCE Dept., Philadelphia, 1987. [Add if applicable: University Microfilms, Inc., University of Michigan, Ann Arbor, Michigan.] **//Format for Dissertation or thesis**
4. R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, “An overview of AL, programming system for automation,” in *Proc. Fourth Int. Joint Conf Artif. Intell*., pp. 758-765, Sept. 3-7, 1975. **//Format for Proceedings paper**
5. “Technology threatens to shatter the world of college textbooks, *The Wall Street Journal*, vol 91, pp. Al, A8, June 1, 1993. **//Format for Newspaper article**
6. R. Cox and J. S. Turner, “Project Zeus: design of a broadband network and its application on a university campus,” Washington Univ., Dept. of Comp. Sci., Technical Report WUCS-91-45, July 30, 1991. **//Format for Technical Report**
7. M. Janzen, *Instant Access Accounting*. Computer software. Nexus Software, Inc IBM-PC, 1993. **//Format for** **Software**
8. Fuminao Okumura and Hajime Takagi, “Maglev Guideway On the Yamanashi Test Line,” *http://www.rtri.or.jp/rd/maglev2/okumura.html*, October 24, 1998. **//Format for** **World Wide Web** (give author and title if named)
9. “AT&T Supplies First CDMA Cellular System in Indonesia,” http://www.att.com/press/1095/951011.nsa.html, Feb 5, 1996. **//Format for World Wide Web**