SAVIFY

By:

Muzammil Arif
35747
Farhan Ahmed
32621

Supervised by:
Mr. Syed Hassaan Ali Shah

Faculty of Computing
Riphah International University, Islamabad
Spring 2025

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: December, 2024

Final Approval

This is to certify that we have read the report submitted by ***Muzammil Arif (35747)***, ***Farhan Ahmed (32621)*** 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 ______________________________

Syed Hassaan Ali Shah
(Supervisor)

2 ______________________________

Dr. Musharraf Ahmed
(Head of Department/chairman)

Declaration

We hereby declare that this document "**Savify**" 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 **Syed Hassaan Ali Shah**. 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.

Muzammil Arif
35747

Farhan Ahmed
32621

Dedication

Our project is dedicated to our parents, teachers, friends, and our supervisor " Syed Hassaan Ali Shah" who has been our mentor and inspiration throughout out educational journey. We are pleased to dedicate our project to such motivational and inspiring people.

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 are greatly indebted to our project supervisor “Syed Hassaan Ali Shah”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Muzammil Arif
35747

Farhan Ahmed
32621

Abstract

In today’s digital age, online shopping has become an integral part of consumer lifestyle, yet many existing platforms fall short of delivering an engaging and intuitive experience. Shoppers often face challenges like limited **interaction**, and the **inability to bargain**, which diminishes the appeal of online shopping compared to traditional markets. Moreover, searching for products can be **time-consuming**, especially for users who want to browse visually or use natural language rather than text search. In response to these limitations, there is a need for an e-commerce platform that combines the **convenience** of online shopping with features that closely mimic the traditional shopping experience. This approach would provide consumers with greater **flexibility**, **personalized interaction**, and a more efficient way to find products tailored to their **preferences**.

Table of Contents

List of Tables

List of Figures

Introduction

Chapter 1: Introduction

Savify is a web-based multi-vendor platform designed to create a more interactive and personalized online shopping experience. The platform allows sellers to upload product information, while buyers can explore, negotiate, and search for items using advanced search functionalities. As current e-commerce platforms often lack personalization and effective engagement features, customers frequently experience impersonal transactions, limited communication options, and challenges in finding relevant products.

A significant issue addressed by *Savify* is the limited interaction between buyers and sellers, as well as the lack of real-time communication tools, which often results in a detached shopping experience to address these gaps, *Savify* incorporates mechanisms to improve buyer-seller interaction, enhance search capabilities, and maintain a respectful online environment through AI-driven moderation.

Savify aims to provide a transparent, secure, and efficient shopping experience, enhancing customer satisfaction and promoting innovation in online retail. Ultimately, *Savify* supports a more dynamic e-commerce environment, contributing to the growth and engagement of digital marketplaces.

.

1.1 Goals and Objectives

The primary objective of *Savify* is to develop an interactive e-commerce platform where customers can browse, search, and negotiate confidently, creating an experience that feels as engaging as traditional shopping.

<u>Goals:</u>

- To provide a web-based platform enabling shoppers to negotiate on product and get in their desired price.
- To enhance the online shopping experience by enabling visual base search which makes search process easier.

Objectives:

- Create a platform that allows multiple vendors to list and sell products seamlessly, providing a user-friendly experience for both sellers and buyers.

- Develop an AI-driven feature that enables real-time price negotiations between customers and vendors, offering a personalized and interactive shopping experience.
- Incorporate image and voice search capabilities to enhance product discovery and improve user convenience.

Savify aims to set a new standard in e-commerce by addressing gaps in personalization, interaction, and accessibility that are prevalent in current platforms.

1.2 Scope of the Project

- Our Website will be developed on MERN.

- Create a web-based system that allows multiple vendors to register, list, and manage their products, including inventory, pricing, and order fulfillment.

- Implement secure user registration and login functionalities for both customers and vendors, along with profile management features.

- Integrate image search functionality enabling users to search for products using images, and incorporate voice search to enhance user experience and accessibility.

- Develop an AI-driven bargaining feature that allows customers to negotiate prices in real-time, providing a dynamic and personalized shopping experience.

- Include features for customer reviews, ratings, and a support system to address inquiries and enhance overall service quality.

- Problem will be solved with Machine Learning.

  The platform would be accessible and user-friendly, and simplified for both buyer and seller.

Chapter 2: Literature Review

2.1 Introduction

Savify is a web-based multi-vendor e-commerce platform developed to bring a traditional shopping experience into the digital space. While e-commerce has seen rapid growth globally, many platforms **lack interactive** and **personalized features** that could improve user experience and buyer-seller engagement. *Savify* is tailored specifically to provide customers with a more dynamic and connected shopping journey. The platform enables sellers to easily share product details and interact with buyers, addressing common issues such as limited negotiation options, impersonal interactions, and difficulty in finding products intuitively.

The primary challenges addressed by *Savify* include the lack of personalized interaction and buyer engagement, which often reduces customer satisfaction on existing platforms. By integrating features like real-time bargaining, *Savify* has taken steps to bridge this gap, making online shopping both interactive and efficient.

2.2 Background and Problem Elaboration

While multi-vendor e-commerce platforms have broadened online selling opportunities, they often lack features that enhance personalization and user interaction. Traditional platforms miss the personalized negotiations and interactive experiences of physical stores, leading to less customer engagement.

Text-based search functionalities can be limiting due to language barriers or vague descriptions, making product discovery frustrating. By integrating an **AI Bargaining**

System, **image search**, and **voice search**, the platform can simulate in-store experiences and improve accessibility.

This project aims to create a web-based multi-vendor e-commerce platform that addresses these shortcomings by incorporating advanced AI features to enhance user satisfaction and streamline the shopping experience.

2.3 Detailed Literature Review

2.3.1 Definitions

2.3.2 Related Research Work 1

2.3.3 Related Research Work 2

2.4 Literature Review Summary Table

The columns in the table depend upon your problem and should be specific to your project.

Table 1: History of Computing Devices

The summary of various computing devices invented in the past from 1833-1901 is presented here.

No.	Name, reference	Inventor	Year	Input	Output	Description
1.						

2.5 Research Gap

Despite advancements in e-commerce technologies, there is a notable gap in integrating advanced AI features into multi-vendor platforms to enhance user interaction and personalization. Current multi-vendor e-commerce platforms often lack the incorporation of AI-driven bargaining systems that allow for dynamic price negotiations, a feature that could simulate the personalized experience of physical store shopping. While some platforms have implemented image or voice search independently to improve product discovery, there is limited research and practical application combining these functionalities within a single platform.

Existing studies have primarily focused on the individual implementation of AI bargaining agents, image search algorithms, or voice recognition systems in e-commerce settings. However, they do not address the challenges and benefits of integrating these technologies holistically in a multi-vendor environment. This integration could bridge the gap between the impersonal nature of online shopping and the interactive experiences of brick-and-mortar stores.

The research gap lies in developing a comprehensive, web-based multi-vendor e-commerce platform that seamlessly incorporates an AI bargaining system alongside advanced image and voice search capabilities. Such an integration remains underexplored in academic research and commercial applications. Addressing this gap can lead to a more engaging and accessible shopping experience, meeting modern consumer expectations and providing vendors with innovative tools to enhance customer satisfaction and loyalty.

2.6 Problem Statement

Despite the proliferation of multi-vendor e-commerce platforms, many lack advanced features that provide personalized and interactive shopping experiences akin to physical stores. Traditional platforms often miss opportunities for real-time price negotiations, leading to reduced customer engagement and satisfaction. Additionally, reliance on text-based search functionalities presents challenges for users facing language barriers or when product descriptions are insufficient, making product discovery cumbersome. There is a pressing need for an innovative e-commerce solution that integrates an AI-driven bargaining system, image search, and voice search capabilities to enhance user interaction, accessibility, and overall satisfaction in the online shopping experience

Chapter 3: Requirements and Design

In this chapter, we have developed the functional requirements for our actors, i.e., **Buyer, Seller, and Admin**. The requirements are specifically designed for the **Savify** platform.

Savify is a web-based e-commerce platform designed to provide an interactive and efficient way for customers and sellers to connect and engage with each other. The platform is user-friendly, easy to navigate and search, and offers features such as AI bargaining system, image search, AI assistant, and speech-to-text support. These functionalities ensure a convenient and seamless experience for all users.

We created system use cases based on each functional requirement and developed corresponding use case diagrams. Additionally, we prepared fully dressed use cases for the main actors, i.e., **Buyer, Seller, and Admin**, ensuring that each role's interactions and responsibilities are clearly outlined within the Savify system.

3.1 Requirements

3.1.1 Functional Requirements

Buyer:

ID	**Requirements**
FR-1.1	Buyer shall be able to sign up on website.
FR-1.2	Buyer shall be able to login to the website.
FR-1.3	Buyer shall be able to edit their profile.
FR-1.4	Buyer shall be able to recover passwords.
FR-1.5	Buyer shall be able to add product to cart.
FR-1.6	Buyer shall be able to delete product from cart.
FR-1.7	Buyer shall be able to buy product.
FR-1.8	Buyer shall be able to add review to product.
FR-1.9	Buyer shall be able to bargain from seller.
FR-1.10	Buyer shall be able to view products
FR-1.11	Buyer shall be able to view orders.
FR-1.12	Buyer shall be able to view cart.
FR-1.13	Buyer shall be able to chat with seller.
FR-1.14	Buyer shall be able to view chats.
FR-1.15	Buyer shall be chat with savify.

Seller:

ID	Requirements
FR-2.1	Seller shall be able to register their account.
FR-2.2	Seller shall be able to login to their account.
FR-2.3	Seller shall be able to edit their profile.
FR-2.4	Seller shall be able to recover passwords.
FR-2.5	Seller shall be able to add products.
FR-2.6	Seller shall be able to view products.
FR-2.7	Seller shall be able to delete products.
FR-2.8	Seller shall be able to edit products.
FR-2.9	Seller shall be able to view orders.
FR-2.10	Seller shall be able to manage orders.
FR-2.11	Seller shall be able to reply to customers.
FR-2.12	Seller shall be able to chat with savify.

Admin:

ID	Requirements
FR-3.1	Admin shall be able to login to account.
FR-3.2	Admin shall be able to edit profile.
FR-3.3	Admin shall be able to view sellers.
FR-3.4	Admin shall be able to add sellers.
FR-3.5	Admin shall be able to delete sellers.
FR-3.6	Admin shall be able to edit sellers.
FR-3.7	Admin shall be able to view buyers.
FR-3.8	Admin shall be able to add buyers..
FR-3.9	Admin shall be able to delete buyers.
FR-3.10	Admin shall be able to edit buyers.
FR-3.11	Admin shall be able to reply to buyer and seller.

3.1.2 Non-Functional Requirements

- **Data Security and Privacy:** The platform ensures the confidentiality and security of user data through robust encryption protocols and compliance with data protection standards.

3.1.3 Hardware and Software Requirements

Hardware Requirements:

- **Server:** Server should run windows 10-11 for the latest requirements.
- **Storage:** Moderate Storage to save all the data during and after project completion.
- **Processors:** High performance Processors such as GPUs to efficiently compute the projects.
- **Camera:** Webcam for product Search through image detection.

Software Requirements:

- **Operating System:** Operating system such as Windows, Linux or MacOS.
- **Database:** We used MongoDB as our Database for storage purpose.
- **Programming Languages:**
  - The website can be built using the MERN stack, which includes:
    - ➢ JavaScript: for server-side and client-side scripting.
    - ➢ Node.js: A JavaScript runtime environment for server-side development.
    - ➢ Express.js: a web application framework for building the server-side application.
    - ➢ React.js: A JavaScript library for building the client-side user interface.
  - Python: Trained YOLOv11 latest model on product images.
- **Development Tools:** Development tools such as Google Colab, Visual Studio code to run and debug codes. Furthermore, we used Roboflow to annotate images of dataset.
- **Version Control:** A version control system like Git to manage source code and collaborate with multiple developers.

3.2 Proposed Methodology

Savify is a web-based platform designed to connect customers and sellers easily. The platform is tailored specifically for e-commerce, providing features that facilitate seamless interaction between both parties. Sellers can upload their products, and customers can browse, search, and purchase items conveniently.

As current e-commerce platforms in Pakistan often lack features like real-time bargaining and AI-assisted shopping, Savify addresses these gaps by creating a user-friendly and innovative experience. The platform allows sellers to showcase their products with images and descriptions, while customers can negotiate prices using the built-in bargaining system.

3.3 System Architecture

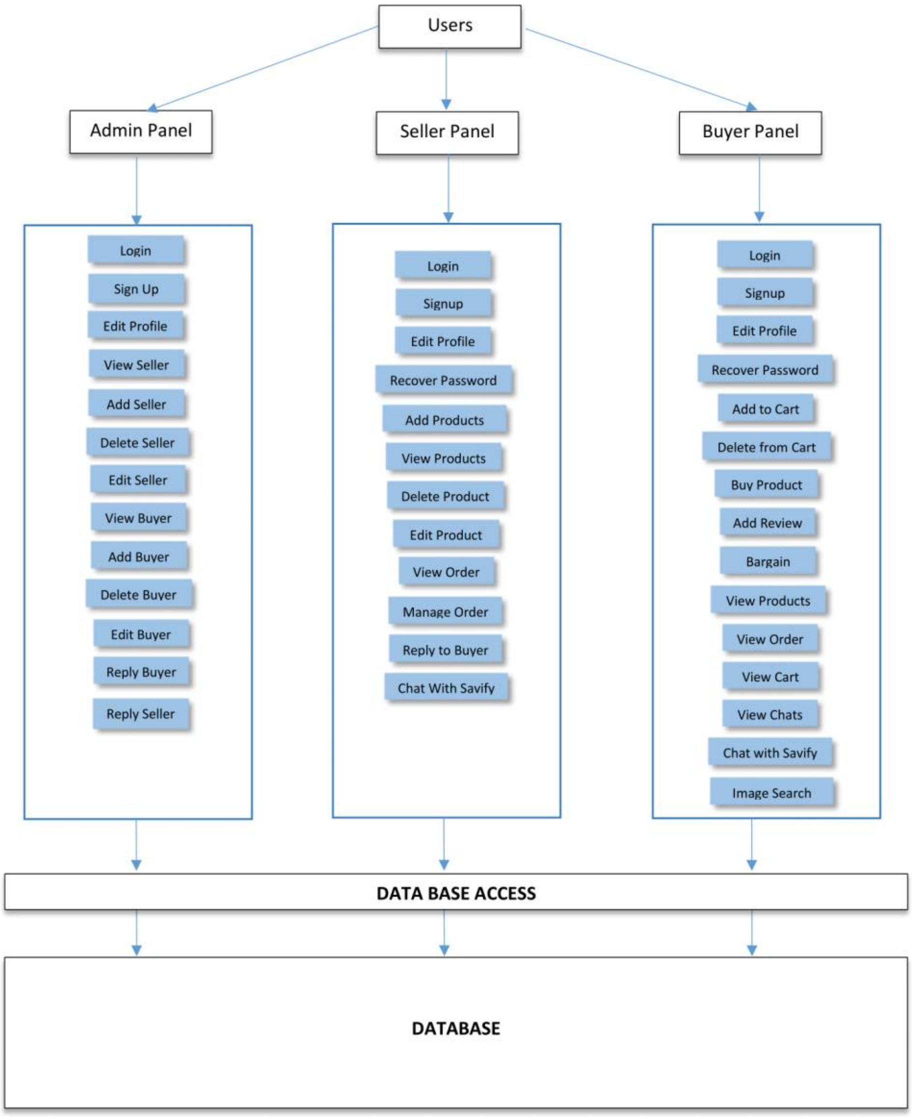


Figure 3.3. 1 : System Architecture

3.4 Use Cases

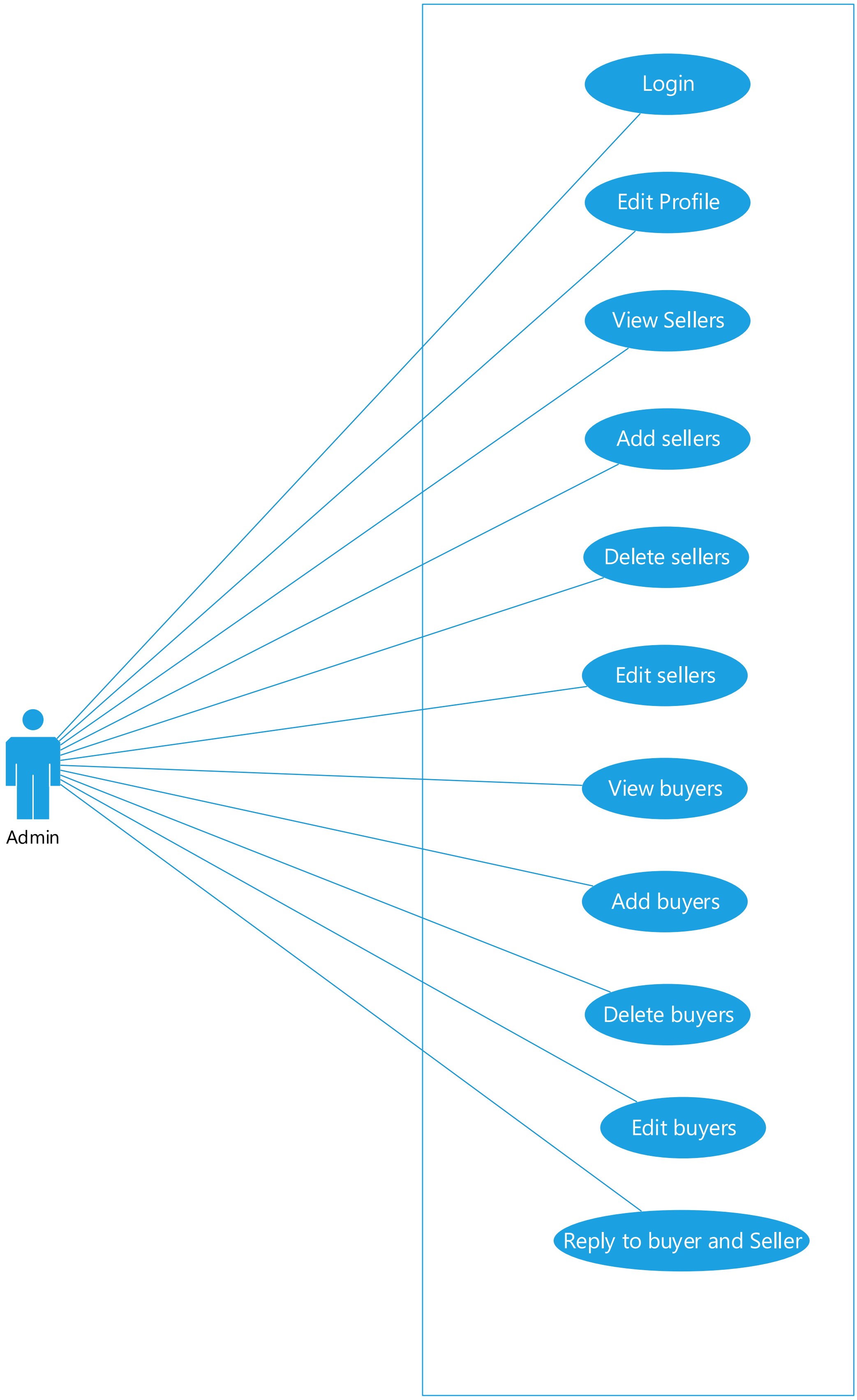


Figure 3.4. 1 : Admin Usecase

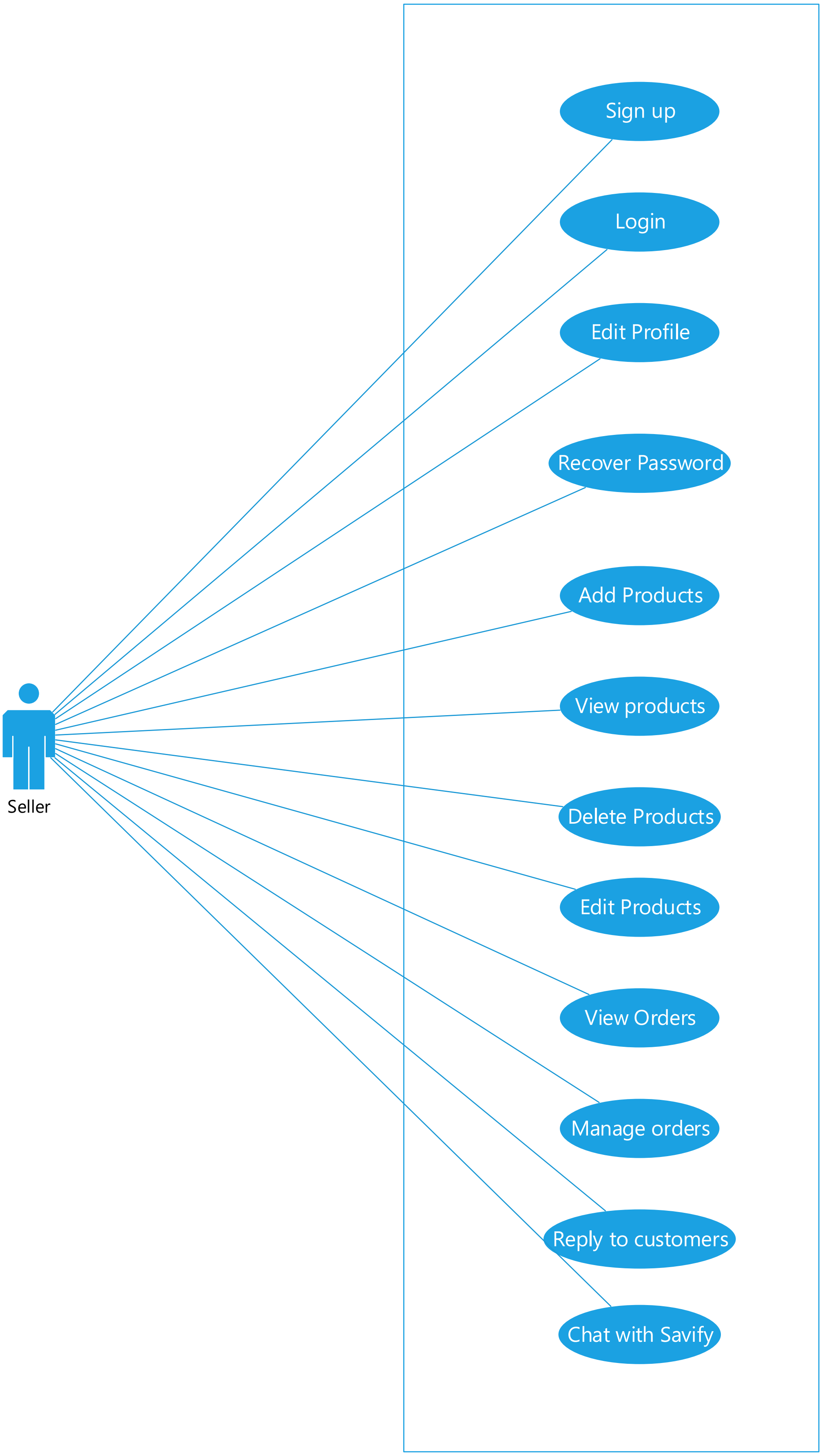


Figure 3.4. 2 : Seller Usecase

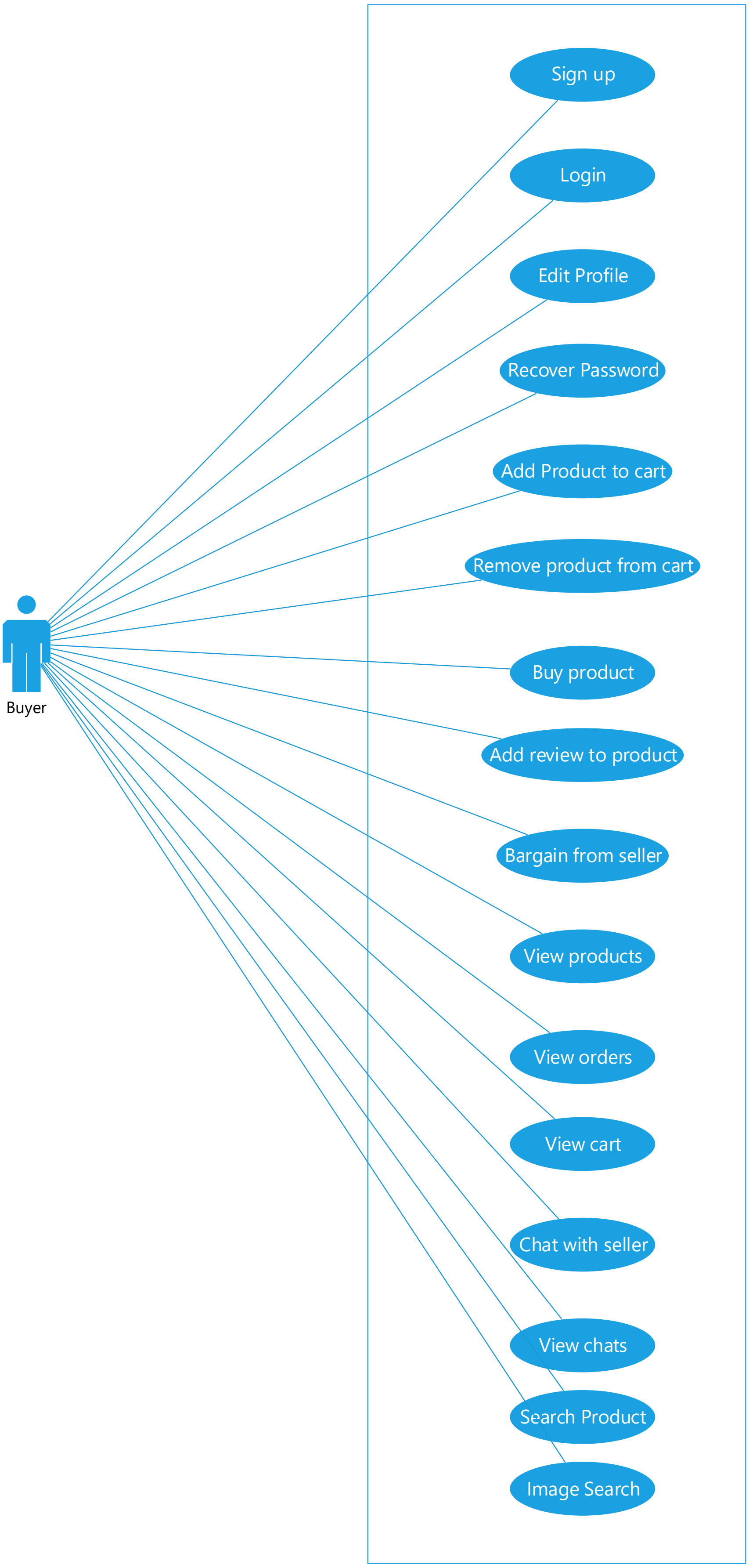


Figure 3.4. 3 : Buyer Usecase

Fully-Dressed Use Cases:

3.4.1 Login

<table>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Login to System</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Admin, Seller, Buyer</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">The user provides their login credentials. If valid, they are granted access to the system.</td></tr>
<tr><td colspan="2"><b>Pre-Conditions</b></td><td colspan="2">• The user must be registered in the system database.
• The user must not already be logged in.</td></tr>
<tr><td colspan="2"><b>Post-Conditions</b></td><td colspan="2">• The user’s session is initiated.
• The user is redirected to their respective dashboard.</td></tr>
<tr><td colspan="2"><b>Special Requirements</b></td><td colspan="2">• Ensure encryption of passwords during verification.
• Provide feedback for invalid credentials.</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
<tr><td>1</td><td>The user opens the login page.</td><td>2</td><td>The login page is displayed asking for email and password.</td></tr>
<tr><td>3</td><td>The user enters valid email and password.</td><td>4</td><td>The system verifies the credentials, starts the session, and redirects to the appropriate dashboard.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>3</td><td>The user enters invalid email or password.</td><td>4-A</td><td>The system displays an error message: "Invalid email or password."</td></tr>
</table>

Table 3.4. 1 : Login to system

3.4.2 Sign Up

<table>
<tr><td colspan="2"><b>Name</b></td><td colspan="2">Sign up</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td colspan="2">Seller, Buyer</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td colspan="2">The user registers for an account by providing necessary details.</td></tr>
<tr><td colspan="2"><b>Pre-Conditions</b></td><td colspan="2">• The user must not already have an account.
• All required fields in the sign-up form must be valid.</td></tr>
<tr><td colspan="2"><b>Post-Conditions</b></td><td colspan="2">• A new account is created.
• The user is redirected to the login page for authentication.</td></tr>
<tr><td colspan="2"><b>Special Requirements</b></td><td colspan="2">• Validate email format.
• Check for duplicate email addresses during registration.</td></tr>
<tr><td colspan="4"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="2"><b>System Response</b></td></tr>
</table>

1	The user navigates to the sign-up page.	1	The user navigates to the sign-up page.
3	The user fills out the form and submits it.	3	The user fills out the form and submits it.
Alternative Flow			
4.1	The user provides invalid or duplicate information.	4.2	The user provides invalid or duplicate information.

Table 3.4. 2 : Sign Up

3.4.3 Edit Profile Information

Name	Edit Profile Information		
Actors	Admin, Seller, Buyer		
Summary	A logged-in user updates their profile details.		
Pre-Conditions	• The user must be logged in. • The user must have access to the "Edit Profile" section.		
Post-Conditions	• The user’s updated details are stored in the database. • Changes are reflected in the user’s account.		
Special Requirements	• Validation for email format, phone numbers, etc. • Real-time feedback for successful updates.		
Basic Flow			
Actor Action		**System Response**	
1	The user navigates to "Edit Profile."	2	The system displays the user’s current profile information.
3	The user modifies the desired fields and submits the form.	4	The system validates inputs and updates the user’s profile in the database, displaying a success message.
Alternative Flow			
4.1	The user enters invalid data (e.g., invalid email format).	4.2	The system highlights errors in the form and prompts the user to fix them.

Table 3.4. 3 : Edit Profile

3.4.4 : Add Sellers

Name	Add Sellers
Actors	Admin
Summary	The Admin can add a new seller to the platform by providing seller details such as name, email, contact number etc. and store information. Upon successful addition, the seller is registered in the system.
Pre-Conditions	• The Admin must be logged into the system with valid credentials. • The Admin must have the necessary privileges to manage sellers.

	• The required seller details (e.g., name, email, contact number, and store name) must be available.		
Post-Conditions	The new seller is successfully added to the database.		
Special Requirements	The system must validate that the seller's email is unique.		
Basic Flow			
Actor Action		**System Response**	
1	The Admin selects the "Add Seller" option in the system.	2	The system displays a form for entering seller details (name, email, contact number, store name, etc.).
3	The Admin enters the seller's details and submits the form.	4	The system validates the input (e.g., unique email, valid contact number).
5	If the input is valid, the Admin confirms the action.	6	The system saves the seller details in the database.
		7	The system displays a success message confirming the seller has been added.
Alternative Flow			
4.1	The Admin submits incomplete or invalid seller details (e.g., duplicate email or invalid contact number).	4.2	The system displays an error message indicating the validation failure and highlights the invalid fields.

Table 3.4. 4 : Add Sellers

3.4.5 : View Sellers

Name	View Seller		
Actors	Admin		
Summary	The admin views the details of a registered seller, including their profile information, products, sales, and other relevant data.		
Pre-Conditions	• The admin must be logged into the system with appropriate permissions. • There must be at least one seller registered in the system.		
Post-Conditions	• The admin successfully views the details of the selected seller. • No data is modified during this process.		
Special Requirements	• The system should load seller details quickly. • The interface should provide comprehensive details, including seller profile, products, and performance metrics. • Only authorized admins should have access to seller details.		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the "Manage Sellers" section in the admin dashboard.	2	The system displays a list of all registered sellers.
3	The admin selects a specific seller to view.	4	The system retrieves and displays the seller's details, including profile information, a list of products, sales

			statistics, and feedback received from buyers.
Alternative Flow			
3	If there are no sellers in the system, the system displays a message indicating that no sellers are available to view.	4-A	If the selected seller's account has been deleted or is inaccessible, the system displays an error message.

Table 3.4. 5 : View Sellers

3.4.6 : Edit Sellers

Name	Edit Seller		
Actors	Admin		
Summary	The admin updates the details of a registered seller, such as their profile information, account status, or permissions.		
Pre-Conditions	• The admin must be logged into the system with the appropriate permissions. • The seller's account exists in the system. • The admin must know the details that need to be updated.		
Post-Conditions	• The seller's information is updated in the database. • The seller is notified about the changes made to their account.		
Special Requirements	• Validation for updated information (e.g., valid email format, unique usernames, etc.). • Notifications must be sent to the seller upon changes. • Changes to critical data (e.g., permissions or account status) must require additional confirmation from the admin.		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the "Manage Sellers" section in the admin dashboard..	2	The system displays a list of all registered sellers
3	The admin selects a specific seller to edit.	4	The system retrieves and displays the current details of the seller in an editable form.
5	The admin modifies the necessary details (e.g., name, email, account status) and submits the changes.	6	The system validates the entered information and updates the seller’s details in the database.
		7	The system notifies the seller of the changes made to their account.
Alternative Flow			
3	If the admin enters invalid or incomplete data (e.g., invalid email format), the system displays an error	4-A	If the admin decides not to proceed with the changes, they can cancel the action, and no updates are made to the seller's

	message and prompts the admin to correct the information.		account.

Table 3.4. 6: Edit Sellers

3.4.7 : Delete Sellers

Name	Delete Seller		
Actors	Admin		
Summary	Admin Deletes Sellers.		
Pre-Conditions	• Admin must be logged in. • The seller must be registered in the system.		
Post-Conditions	• The seller’s account is deleted from the database. • The seller's products and associated data are either archived or removed based on platform policies.		
Special Requirements	• Admin must confirm deletion to prevent accidental removals. • The system should notify the seller of their account deletion via email or notification. • Compliant with data retention and privacy laws (e.g., GDPR).		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to "Manage Sellers."	2	The system displays a list of registered Sellers.
3	The admin selects a buyer to Delete.	4	The system performs the action and confirms the changes.
Alternative Flow			
3	The admin attempts to delete a non-existent Seller.	4-A	The system displays an error message: "Seller not found."

Table 3.4.7 : Delete Sellers

3.4.8 : Add Buyer

Name	Add Buyer
Actors	Admin
Summary	The admin manually adds a new buyer to the system, providing their account details and profile information
Pre-Conditions	• The admin must be logged into the system with appropriate permissions. • The required buyer details (e.g., name, email, and password) must be available.
Post-Conditions	• The buyer is successfully added to the system. • The buyer receives a notification or email with their account credentials and login instructions.

Special Requirements	• The system must validate the buyer’s email for uniqueness. • The system must securely encrypt the buyer’s password. • The system should provide feedback to the admin on the success or failure of the operation.		
Basic Flow			
Actor Action		**System Response**	
1	The admin navigates to the "Manage Buyers" section in the admin dashboard	2	The system displays an option to add a new buyer.
	The admin selects "Add Buyer" and fills in the required details (e.g., name, email, contact number, and password).		The system validates the entered details for completeness and uniqueness (e.g., checks if the email is already in use).
	The admin confirms the addition of the buyer.		The system creates the buyer account, stores the details in the database, and generates a notification or email to send to the buyer.
			The system displays a success message to the admin.
Alternative Flow			
3	If the admin enters invalid data (e.g., email format errors, duplicate email), the system displays an error message and prompts the admin to correct the input.	4-A	If the admin cancels the action at any point, the buyer’s account is not created, and the system returns to the "Manage Buyers" page.

Table 3.4.8 : Add Buyer

3.4.9 : Delete Buyers

Name	Delete Buyers
Actors	Admin
Summary	Admin Deletes Buyers.
Pre-Conditions	• Admin must be logged in. • The buyer must be registered in the system.
Post-Conditions	• The buyer’s account is deleted from the database. • The buyer’s products and associated data are either archived or removed based on platform policies.
Special Requirements	• Admin must confirm deletion to prevent accidental removals. • The system should notify the seller of their account deletion via email or notification. • Compliant with data retention and privacy laws (e.g., GDPR).
Basic Flow	
Actor Action	**System Response**

<table>
<tr><td>1</td><td>The admin navigates to "Manage Buyers."</td><td>2</td><td>The system displays a list of registered Buyers.</td></tr>
<tr><td>3</td><td>The admin selects a buyer to Delete.</td><td>4</td><td>The system performs the action and confirms the changes.</td></tr>
<tr><td colspan="4"><b>Alternative Flow</b></td></tr>
<tr><td>3</td><td>The admin attempts to delete a non-existent Buyer.</td><td>4-A</td><td>The system displays an error message: "Buyer not found."</td></tr>
</table>

Table 3.4.9 : Delete Buyers

3.4.10 : View Buyer

<table>
<tr><td><b>Name</b></td><td colspan="4">View Buyer</td></tr>
<tr><td><b>Actors</b></td><td colspan="4">Admin</td></tr>
<tr><td><b>Summary</b></td><td colspan="4">The admin views the details of a registered buyer, including their profile information, order history, and other relevant data.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="4">• The admin must be logged into the system with appropriate permissions.
• There must be at least one buyer registered in the system.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="4">• The admin successfully views the details of the selected Buyer.
• No data is modified during this process.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="4">• The system should load Buyer details quickly.
• The interface should provide comprehensive details, including buyer profile, products, and performance metrics.
• Only authorized admins should have access to buyer details.</td></tr>
<tr><td colspan="5"><b>Basic Flow</b></td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td colspan="3"><b>System Response</b></td></tr>
<tr><td>1</td><td>The admin navigates to the "Manage Buyers" section in the admin dashboard.</td><td>2</td><td colspan="2">The system displays a list of all registered Buyers.</td></tr>
<tr><td>3</td><td>The admin selects a specific buyer to view.</td><td>4</td><td colspan="2">The system retrieves and displays the buyer’s details, including profile information.</td></tr>
<tr><td colspan="5"><b>Alternative Flow</b></td></tr>
<tr><td>3</td><td>If there are no buyer in the system, the system displays a message indicating that no buyer are available to view.</td><td>4-A</td><td colspan="2">If the selected buyer account has been deleted or is inaccessible, the system displays an error message.</td></tr>
</table>

Table 3.4.10 : View Buyers

3.4.11 : Edit Buyer

Name	Edit Buyer
Actors	Admin

Summary	The admin updates the details of a registered buyer, such as their profile information, account status, or permissions.
Pre-Conditions	• The admin must be logged into the system with the appropriate permissions. • The buyer's account exists in the system. • The admin must know the details that need to be updated.
Post-Conditions	• The buyer’s information is updated in the database.
Special Requirements	• Validation for updated information (e.g., valid email format, unique usernames, etc.).

<table>
<tr><th colspan="4">Basic Flow</th></tr>
<tr><th colspan="2">Actor Action</th><th colspan="2">System Response</th></tr>
<tr><td>1</td><td>The admin navigates to the "Manage Buyers" section in the admin dashboard.</td><td>2</td><td>The system displays a list of all registered buyers.</td></tr>
<tr><td>3</td><td>The admin selects a specific buyer to edit.</td><td>4</td><td>The system retrieves and displays the current details of the buyer in an editable form.</td></tr>
<tr><td>5</td><td>The admin modifies the necessary details (e.g., name, email, account status) and submits the changes.</td><td>6</td><td>The system validates the entered information and updates the buyer’s details in the database.</td></tr>
<tr><th colspan="4">Alternative Flow</th></tr>
<tr><td>3</td><td>If the admin enters invalid or incomplete data (e.g., invalid email format), the system displays an error message and prompts the admin to correct the information.</td><td>4-A</td><td>If the admin decides not to proceed with the changes, they can cancel the action, and no updates are made to the buyer’s account.</td></tr>
</table>

Table 3.4.11 : Edit Buyers

3.4.12 : Reply to Buyer or Seller

Name	Reply to Buyer or Seller
Actors	Admin, Buyer, Seller
Summary	The admin responds to messages, queries, or complaints from buyers and sellers via the platform's communication system.
Pre-Conditions	• The admin must be logged into the system with the appropriate permissions. • The buyer or seller must have sent a message or query through the platform. • The communication system must be functional.
Post-Conditions	• The buyer or seller receives the admin’s reply. • The conversation thread is updated in the system and stored for future reference.

<table>
<tr><td><b>Special Requirements</b></td><td colspan="3">The system must provide a clear interface for message management.</td></tr>
<tr><td colspan="4" align="center"><b>Basic Flow</b></td></tr>
<tr><td colspan="2" align="center"><b>Actor Action</b></td><td colspan="2" align="center"><b>System Response</b></td></tr>
<tr><td>1</td><td>The admin navigates to the "Messages" or "Support Center" section in the admin dashboard.</td><td>2</td><td>The system displays a list of incoming messages or queries from buyers and sellers.</td></tr>
<tr><td>3</td><td>The admin selects a specific message or query to respond to.</td><td>4</td><td>The system displays the conversation thread and provides a text input area for the admin to type a reply.</td></tr>
<tr><td>5</td><td>The admin types a response and submits it.</td><td>6</td><td>The system sends the reply to the buyer or seller and updates the conversation thread.</td></tr>
<tr><td colspan="4" align="center"><b>Alternative Flow</b></td></tr>
<tr><td>3</td><td></td><td>4-A</td><td></td></tr>
</table>

Table 3.4.12 : Reply to Buyer or Seller

3.4.13 : Recover Password

<table>
<tr><td><b>Name</b></td><td colspan="3">Recover Password</td></tr>
<tr><td><b>Actors</b></td><td colspan="3">Seller, Buyer</td></tr>
<tr><td><b>Summary</b></td><td colspan="3">A seller or buyer can recover their password by initiating a password reset process through the platform.</td></tr>
<tr><td><b>Pre-Conditions</b></td><td colspan="3">• The seller or buyer must have an active account on the platform.
• The user must have access to the registered email or phone number associated with their account.</td></tr>
<tr><td><b>Post-Conditions</b></td><td colspan="3">• The system sends a password reset link or code to the registered email or phone number.
• The user successfully resets their password and regains access to their account.</td></tr>
<tr><td><b>Special Requirements</b></td><td colspan="3">• The reset process must be secure to prevent unauthorized access.
• Password reset links or codes must have an expiration time (e.g., 15 minutes).</td></tr>
<tr><td colspan="4" align="center"><b>Basic Flow</b></td></tr>
<tr><td colspan="2" align="center"><b>Actor Action</b></td><td colspan="2" align="center"><b>System Response</b></td></tr>
<tr><td>1</td><td>The user navigates to the login page and clicks "Forgot Password."</td><td>2</td><td>The system prompts the user to enter their registered email.</td></tr>
<tr><td>3</td><td>The user enters their registered email and submits the request.</td><td>4</td><td>The system validates the input and sends a password reset link or code to the provided email.</td></tr>
<tr><td>5</td><td>The user clicks the link or enters the code and navigates to the password reset page.</td><td>6</td><td>The system prompts the user to enter a new password.</td></tr>
</table>

7	The user enters and confirms the new password, then submits the form.	8	The system validates the new password, updates the user's credentials in the database, and confirms the reset.
Alternative Flow			
3	If the user enters an unregistered email.	4-A	The system displays an error message and prompts for a valid input.
	If the user attempts to use an expired reset link or code		The system notifies the user and prompts them to request a new one

Table 3.4. 13 : Recover Password

3.4.14 : Add Products to cart

Name	Add Products to Cart		
Actors	Buyer		
Summary	The buyer adds selected products to their shopping cart.		
Pre-Conditions	The buyer must be logged in.		
Post-Conditions	• The selected product is added to the cart. • The buyer’s cart is updated in real time.		
Special Requirements	Allow the buyer to specify quantity before adding to the cart.		
Basic Flow			
Actor Action		**System Response**	
1	The buyer clicks "Add to Cart" for a product.	2	The system adds the product to the cart and updates the cart’s total.
Alternative Flow			
3	The buyer tries to add an out-of-stock product	4-A	The system displays a message: "This product is currently out of stock."

Table 3.4.14 : Add to cart

3.4.15 : Remove product from cart

Name	Remove product from cart
Actors	Buyer
Summary	The buyer can remove a product from their shopping cart.
Pre-Conditions	• The buyer must be logged in. • The buyer must have at least one product in the cart.
Post-Conditions	The selected product is removed from the cart.
Special Requirements	
Basic Flow	

Actor Action		System Response	
1	The buyer navigates to the cart page.	2	The system displays all products in the buyer's cart.
3	The buyer clicks the "Remove" button next to a product.	4	The system removes the product from the cart and updates the total price.
Alternative Flow			
3		4-A	

Table 3.4.15 : Remove Product from cart

3.4.16 : Add Review to Product

Name	Add Review to Product		
Actors	Buyer		
Summary	The buyer can add a review to a product.		
Pre-Conditions	The buyer must be logged in.		
Post-Conditions	The review is saved and associated with the product		
Special Requirements			
Basic Flow			
Actor Action		**System Response**	
1	The buyer navigates to the product page.	2	The system displays the product details and a "Write a Review" section.
3	The buyer writes a review and submits it.	4	The system validates the review and saves it to the database.
		5	The system displays a success message: "Your review has been submitted."
Alternative Flow			
3		4-A	

Table 3.4.16 : Add Review

3.4.17 : View Products Page

Name	View Products Page
Actors	Buyer
Summary	The buyer can browse and view details of products.
Pre-Conditions	The buyer must be logged in.

Post-Conditions	The buyer can view product details.		
Special Requirements			
Basic Flow			
Actor Action		System Response	
1	The buyer navigates to the "Shop" page.	2	The system displays a list of products
3	The buyer clicks on a product.	4	The system displays the product details, including images, price, and description.
Alternative Flow			
3		4-A	

Table 3.4.17 : View Products

3.4.18 : View Cart

Name	View Cart		
Actors	Buyer		
Summary	The buyer can view their shopping cart and the products it contains.		
Pre-Conditions	• The buyer must be logged in. • The buyer must have added at least one product to the cart.		
Post-Conditions	The cart content is displayed		
Special Requirements			
Basic Flow			
Actor Action		System Response	
1	The buyer clicks the "Cart" icon or link.	2	The system displays the cart page with all added products, their quantities, and the total price.
Alternative Flow			
3		4-A	

Table 3.4.18 : View Cart

3.4.19 : Search for Product

Name	Search for Product
Actors	Buyer
Summary	The buyer can search for a product using a keyword, such as the product name, category, or brand.
Pre-Conditions	The buyer must be on the platform's main page or a search bar must be accessible.

Post-Conditions	The system displays a list of products that match the search query.		
Special Requirements			
Basic Flow			
Actor Action		**System Response**	
1	The buyer enters a keyword into the search bar.	2	The system processes the query and searches the database for relevant products.
3	The buyer clicks the "Search" button or presses Enter.	4	The system displays a list of matching products with images, names, prices, and availability.
Alternative Flow			
3	If no products match the search query	4-A	The system displays a message: "No products found. Try a different keyword.

Table 3.4.19 : Search Product

3.4.20 : Image Search

Name	Image Search		
Actors	Buyer		
Summary	The buyer can upload or live detect an image to search for similar products using image recognition technology.		
Pre-Conditions	The buyer must have access to webcam access or access to files.		
Post-Conditions	The system displays a list of products that match the uploaded image.		
Special Requirements			
Basic Flow			
Actor Action		**System Response**	
1	The buyer clicks the "Image Search" button.	2	The system opens a file uploader or camera option.
3	The buyer uploads an image or takes a photo.	4	The system processes the image using image recognition technology.
		5	The system displays a list of similar products with images, names, and prices.
Alternative Flow			
3	If no products match the uploaded image.	4-A	The system displays a message: "No matching products found. Please try another image.

3.4.21 : AI Bargain by Buyer

Name	View Cart		
Actors	Buyer, AI System		
Summary	The buyer uses the AI system to negotiate the price of a product.		
Pre-Conditions	• The buyer must be logged in. • The product must be eligible for AI Bargain.		
Post-Conditions	The AI suggests a negotiated price, which the buyer can accept or reject.		
Special Requirements			
Basic Flow			
Actor Action		**System Response**	
1	The buyer clicks the "Bargain Now" button on the product page.	2	The system opens an AI chat interface.
3	The buyer provides their offer to the AI.	4	The AI evaluates the offer and responds with a counteroffer.
5	The buyer accepts or rejects the counteroffer.	6	If accepted, the AI applies the negotiated price to the product.
Alternative Flow			
3		4-A	

3.4.22 : Complete Checkout

Name	Complete Checkout		
Actors	Buyer		
Summary	The buyer finalizes their order by providing payment and shipping details.		
Pre-Conditions	• The buyer must be logged in. • The buyer must have items in their cart.		
Post-Conditions	• The order is placed successfully. • Payment is processed and confirmation is sent to the buyer.		
Special Requirements	Allow buyers to review and modify their order before confirming.		
Basic Flow			
Actor Action		**System Response**	
1	The buyer clicks "Checkout" in the cart.	2	The system displays a summary of the cart items and the total price.
3	The buyer provides payment and shipping details.	4	The system validates the input, processes the payment, and displays an order confirmation.

Alternative Flow			
3	The buyer provides invalid shipping details	4-A	The system highlights the errors and prompts the buyer to correct them.

3.4.23 : Add New Product

Name	Add New Products
Actors	Seller
Summary	The seller adds new products to the platform by providing details such as name, price, description, and images.
Pre-Conditions	• The seller must be logged in. • Required product details must be available.
Post-Conditions	• The product is successfully added to the system. • The product becomes available for buyers to view and purchase.
Special Requirements	• Validate input fields (e.g., price, name, and description length). • Allow multiple images to be uploaded.

Basic Flow			
Actor Action		**System Response**	
1	The seller navigates to the "Add Products" page.	2	The system displays a form for entering product details (name, price, description, and images)
3	The seller fills out the form and uploads images.	4	The system validates the input, saves the product, and displays a success message.
Alternative Flow			
3	The seller provides incomplete or invalid details.	4-A	The system highlights the errors and prompts the seller to correct them.

3.4.24 : View Product

Name	View Products
Actors	Seller
Summary	A seller views the detailed information of their own products, including product status, inventory, price, and product performance (e.g., views and orders).
Pre-Conditions	The seller must be logged into their account. - The product must exist in the seller's inventory.
Post-Conditions	- The product details are displayed successfully. - The seller can make further actions like editing or managing inventory.
Special Requirements	The system must ensure that the seller can only view products they own.
Basic Flow	

Actor Action		System Response	
1	The seller logs in and navigates to the "Manage Products" section in their dashboard.	2	The system displays a list of all products the seller has listed, including summary information such as name, price, and stock status.
3	The seller selects a specific product to view more details.	4	The system retrieves and displays detailed information about the selected product, such as product description, price, stock levels, product status (e.g., active, inactive),
Alternative Flow			
3	If the seller has not listed any products	4-A	The system displays a message indicating that no products are available and suggests adding a new product.

3.4.25 : Delete Product

Name	Delete Products
Actors	Seller
Summary	The seller can delete a product listed in their store. Upon confirmation, the product is removed from the store, and the database is updated.
Pre-Conditions	• The seller must be authenticated and logged into their account. • The seller must have at least one product listed in their store.
Post-Conditions	The product is successfully removed from the database and is no longer visible to buyers.
Special Requirements	• The system should validate that the seller can only delete their own products. • There must be a confirmation step before the deletion to avoid accidental actions.
Basic Flow	

Actor Action		System Response	
1	The seller logs in and navigates to the "Manage Products" section.	2	The system displays a list of products added by the seller.
3	The seller clicks the "Delete" button for a specific product.	4	The system displays a confirmation prompt: "Are you sure you want to delete this product?
5	The seller confirms the deletion.	6	The system deletes the product from the database and updates the product list. The system displays a success message: "Product deleted successfully."
Alternative Flow			
3	The seller clicks "Cancel" on the confirmation prompt.	4-A	The seller clicks "Cancel" on the confirmation prompt.

3.4.26 : Reply to Customers

Name	Reply to Customers		
Actors	Seller		
Summary	The seller can respond to customer inquiries or reviews regarding their products through the system. Sellers can view messages or comments and provide replies.		
Pre-Conditions	• The seller must be authenticated and logged into their account. • The seller must have received at least one inquiry or review from customers.		
Post-Conditions	• The customer's inquiry is marked as responded to. • The customer receives the reply through the system.		
Special Requirements			
Basic Flow			
Actor Action		**System Response**	
1	The seller logs into their account.	2	The system displays the seller's dashboard with a "Messages/Reviews" section.
3	The seller navigates to the "Messages/Reviews" section.	4	The system displays a list of customer inquiries or reviews related to the seller's products.
5	The seller clicks on a specific inquiry or review.	6	The system displays the detailed inquiry/review along with a text box for the reply.
Alternative Flow			
3		4-A	

3.4.27 : View and Manage Buyer Orders

Name	View and Manage Buyer Orders
Actors	Seller
Summary	The seller reviews and manages orders placed by buyers for their products.
Pre-Conditions	• The seller must be logged in. • Orders for the seller's products must exist in the system.
Post-Conditions	• The seller updates the status of orders (e.g., confirmed, shipped). • Buyers are notified of order status changes.
Special Requirements	• Display order details clearly, including buyer information.

	• Allow filtering orders by name		
Basic Flow			
Actor Action		**System Response**	
1	The seller navigates to the "Manage Orders" page.	2	The system displays a list of orders for the seller's products.
3	The seller selects an order and updates its status.	4	The system updates the order status and notifies the buyer of the changes.
Alternative Flow			
3	The seller attempts to update an invalid order.	4-A	The system displays an error message: "Unable to update order. Please try again."

3.4.28 : Edit Products

Name	Edit Products		
Actors	Seller		
Summary	The seller modifies details of their existing products.		
Pre-Conditions	• The seller must be logged in. • The product to be edited must exist in the system and belong to the seller.		
Post-Conditions	• The updated product details are saved in the system. • Buyers see the updated product details immediately.		
Special Requirements	• Validate all updated fields (e.g., price must be numeric, description must meet length criteria). • Ensure no duplicate product names within the seller's product list.		
Basic Flow			
Actor Action		**System Response**	
1	The seller navigates to the "Edit Products" page.	2	The system displays a list of products added by the seller.
3	The seller selects a product to edit.	4	The system displays the product details in an editable form.
5	The seller updates the details and submits the form.	6	The system validates the input, saves the changes, and displays a success message.
Alternative Flow			
5	The seller enters invalid or duplicate product details.	6-A	6-A. The system highlights errors and prompts the seller to fix them.

3.5 Database Design *(Optional)*

3.6 Class Diagram (*Optional)*

3.7 Sequence diagram *(Optional)*

3.8 Any Other Artifact…

3.9 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.

Chapter 4: Implementation and Test Cases

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

4.1 Implementation

We have successfully implemented a Seller Dashboard and a Buyer Dashboard, seamlessly integrating advanced functionalities to enhance user experience. A standout feature is the Image Search capability, powered by the state-of-the-art YOLOv11 model. This model has been meticulously custom-trained on a specialized dataset, ensuring high accuracy and relevance in image-based queries. The Seller Dashboard enables efficient management of listings, while the Buyer Dashboard provides intuitive navigation and interaction for shoppers. Together, these components leverage cutting-edge AI technologies to redefine how sellers and buyers connect in a digital marketplace.

4.1.1 Implementation of First Component/Algorithm

The first component, Image Search, is a critical functionality of the platform, leveraging a robust machine learning model for accurate and efficient results. The YOLOv11 model was trained using a structured approach outlined in the diagram provided. Here's a detailed description of the training process:

1.Dataset Collection: Data collection began with three primary datasets: Hammer Dataset, Remote Dataset, and Lighter Dataset. These datasets were sourced to ensure a diverse and comprehensive range of images relevant to the search functionality.

2.**Merging Datasets:** The individual datasets were consolidated into one unified dataset. This step ensured that the model had access to a sufficiently large and varied dataset, enhancing its ability to generalize effectively.

3.Dataset Preprocessing: Preprocessing involved cleaning and standardizing the dataset, addressing issues like inconsistent image sizes, resolution mismatches, or irrelevant data. This step optimized the dataset for model training.

4.Dataset Labeling: Proper annotations were added to the dataset during this stage. Each image was labeled with bounding boxes and relevant class information, a critical step to teach the YOLOv11 model to identify objects accurately.

5.Dataset Augmentation: To further enrich the dataset, augmentation techniques were applied. This included transformations such as rotation, scaling, flipping, and brightness adjustments, enhancing the model's robustness to real-world scenarios.

6.YOLOv11 Model Training: With the preprocessed and augmented dataset, the YOLOv11 model was trained. This stage involved optimizing the model’s architecture and hyper parameters to achieve high performance in object detection tasks.

7.Trained Model: The final trained model is now equipped to perform image searches with high precision, enabling users to search for items by simply uploading an image. This capability is essential for an intuitive and seamless user experience on the platform.

This systematic approach ensures the image search functionality is powered by a well-trained, high-performance YOLOv11 model, delivering accurate and reliable results.

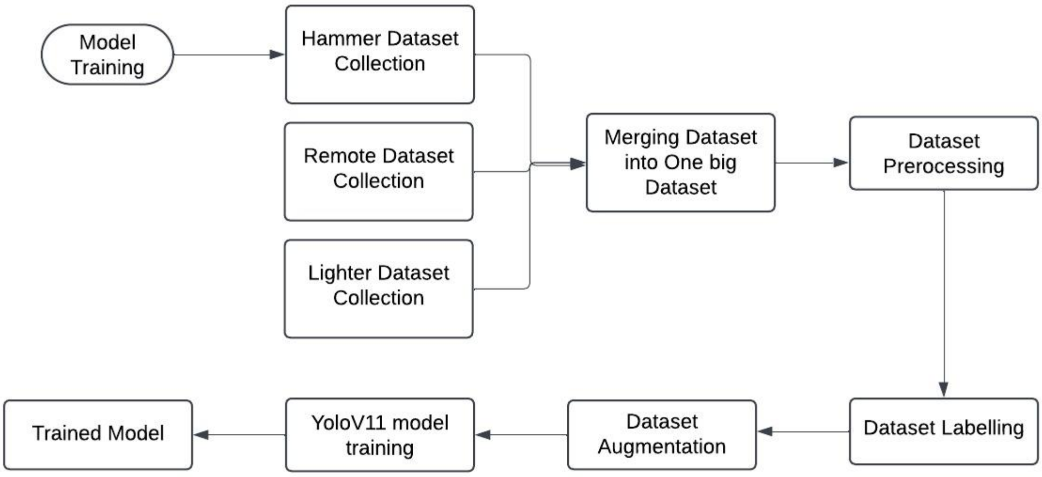

Model Training
Hammer Dataset Collection
Remote Dataset Collection
Lighter Dataset Collection
Merging Dataset into One big Dataset
Dataset Perocessing
Dataset Labelling
Dataset Augmentation
YoloV11 model training
Trained Model

Figure 4.1.1 : Model Training

4.2 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.

4.2.1 Sample Test case No.1

<Software component Name>			
<Reference>			
Test Case ID:	*Reference*	**Test Date:**	*Date*

	Number		
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*			

4.2.2 Sample Test case No.2

.

.

.

4.3 Test Metrics

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

4.3.1 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.

4.3.2 Sample Test case Metric.No.2

4.3.3 Sample Test case Metric.No.3

.

.

.

Chapter 5: 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.**

Chapter 6: 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.

Chapter 1: Heading 1

1.1 Heading 2

1.1.1 Heading 3

1.1.1.1 Heading 4

1.1.1.1.1 Heading 5

1.1.1.1.1.1 Heading 6

1.1.1.1.1.1.1 Heading 7

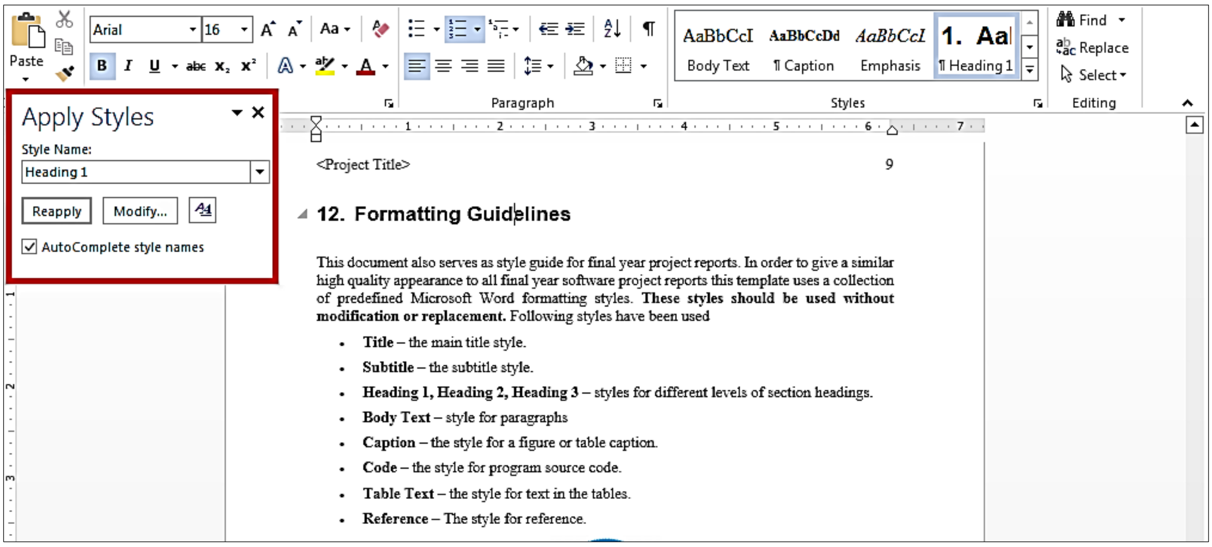
1.1.1.1.1.1.1.1 Heading 8

1.1.1.1.1.1.1.1.1 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.



Style Name: Heading 1

☑ AutoComplete style names

<Project Title> 9

12. 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.** Following styles have been used

- **Title** – the main title style.
- **Subtitle** – the subtitle style.
- **Heading 1, Heading 2, Heading 3** – styles for different levels of section headings.
- **Body Text** – style for paragraphs
- **Caption** – the style for a figure or table caption.
- **Code** – the style for program source code.
- **Table Text** – the style for text in the tables.
- **Reference** – The style for reference.

Figure 1: List of Styles

This is brief description of above figure.

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

Table 3: 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)’.

$$\sum_{\forall v \in V(G)} \deg(v) = 2\left|E(G)\right| \tag{1}$$

Header/Footer

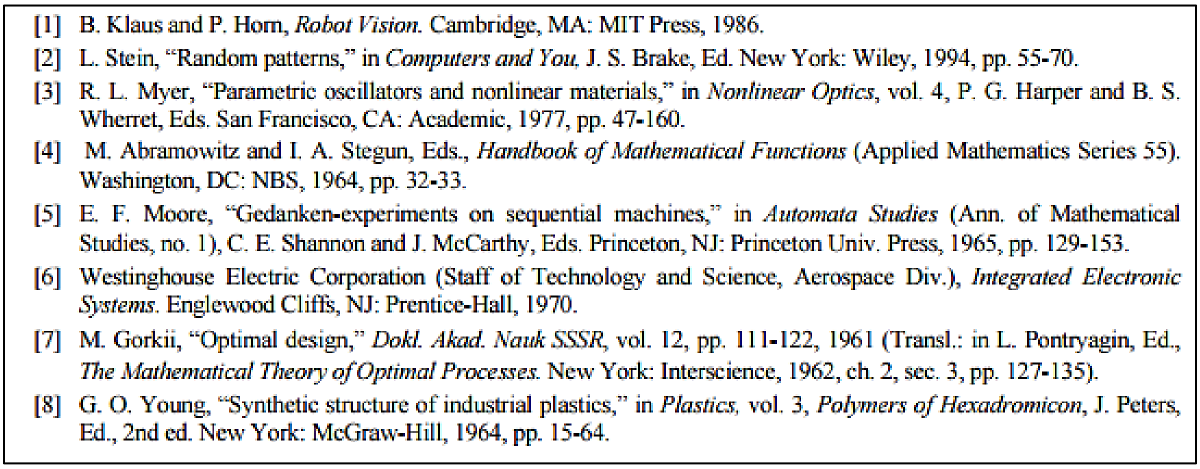
Notice the headers in this document, before Introduction (i.e. the main content of this document) page numbers are in roman numerals. The page numbers of the actual content start with Arabic numerals i.e. 1, 2, 3 and so on. All of the **odd numbered pages** contain title of your project while the **even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



[1] B. Klaus and P. Horn, *Robot Vision.* Cambridge, MA: MIT Press, 1986.

[2] L. Stein, “Random patterns,” in *Computers and You,* J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

[3] R. L. Myer, “Parametric oscillators and nonlinear materials,” in *Nonlinear Optics*, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.

[4] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions* (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.

[5] E. F. Moore, “Gedanken-experiments on sequential machines,” in *Automata Studies* (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.

[6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), *Integrated Electronic Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1970.

[7] M. Gorkii, “Optimal design,” *Dokl. Akad. Nauk SSSR,* vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., *The Mathematical Theory of Optimal Processes.* New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).

[8] G. O. Young, “Synthetic structure of industrial plastics,” in *Plastics,* vol. 3, *Polymers of Hexadromicon*, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.

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**