



SnapMart

Submitted by:

- **Abdel Aziz Ayman** (42010193)
- **Abdel Hay Soliman** (42010360)
- **Basma Ashraf** (42010682)
- **Mai Mahmoud** (42010610)
- **Mohamed Ibrahim** (42010235)
- **Salma Atef** (42010652)

**A dissertation submitted in partial fulfillment of the requirements for the degree of
Bachelor of computer science and information technology**

Supervised by:

- **Dr. Shady Mohammed**

Committee Report

We certify we have read this graduation project report as examining committee, examine the student in its content and that in our opinion it is adequate as a project document for "**SnapMart**".

Supervisor:

Name: **Dr. Shady Mohammed**

Signature:

Date: 11 / 2/2024

Chairman:

Name:

Signature:

Date: 11/2/2024

Examiner:

Name:

Signature:

Date: 11 /2 /2024

Intellectual Property Right Declaration

This is to declare that the work under the supervision of Dr. Shady Mohammed Elsaïd having title “SnapMart” carried out in partial fulfillment of the requirements of Bachelor of Science in Computer Science and Information Technology is the sole property of the Ahram Canadian University and the respective supervisor. It is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/ organizational usage, etc. with the permission of the university and respective supervisor. This above statement applies to all students and faculty members.

Names:

Abdel Aziz Ayman

Abdel Hay Soliman

Basma Ashraf

Mai Mahmoud

Mohamed Ibrahim

Salma Atef

Supervisor: **Dr. Shady Mohammed**

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of Dr. Shady Mohammed Elsaid having title "SnapMart" is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Names:

Abdel Aziz Ayman

Abdel Hay Soliman

Basma Ashraf

Mai Mahmoud

Mohamed Ibrahim

Salma Atef

| | |
|--|-----------|
| chapter1. Introduction..... | 6 |
| 1.1 Overview..... | 6 |
| 1.2 Motivation..... | 6 |
| 1.3 Objective..... | 7 |
| 1.4 Aim..... | 8 |
| 1.5 Scope..... | 8 |
| 1.6 General constraints..... | 9 |
| 1.7 Organization of the dissertation..... | 10 |
| Chapter2. Background and Previous work..... | 11 |
| 2.1 Background..... | 11 |
| 2.2 previous..... | 14 |
| Chapter3. Planning and Analysis..... | 23 |
| 3.1 planning..... | 23 |
| 3.1.1 Feasibility study and estimated cost..... | 26 |
| 3.1.2 Gantt chart..... | 28 |
| 3.2 Analysis and limitations of existing system..... | 30 |
| 3.3 Need for new system..... | 34 |
| 3.4 Analysis of new system..... | 35 |
| 3.4.1 User Requirements..... | 35 |
| 3.4.2 System Requirements..... | 35 |
| 3.4.3 Functional Requirements..... | 35 |
| 3.4.4 Non- Functional Requirements..... | 35 |
| 3.4.5 Domain Requirements..... | 38 |
| 3.5 Advantage of new system..... | 40 |
| 3.6 User Characteristics..... | 42 |

Chapter4. Design.....44

| | |
|---|----|
| <i>4.1 Design and Implementation Constraints.....</i> | 44 |
| <i>4.2 Assumptions and dependencies.....</i> | 46 |
| <i>4.3 Design of database ERD.....</i> | 49 |
| 4.3.1 Entity Relationship Diagram..... | 49 |
| 4.3.2 Mapping of Entity Relationship Diagram..... | 50 |
| <i>4.4 Class diagram.....</i> | 51 |
| <i>4.5 Use case diagram.....</i> | 52 |
| 4.5.1 Primary Use-case Diagram..... | 52 |
| 4.5.2 Use-case Scenarios..... | 52 |
| <i>4.6 Activity diagram.....</i> | 53 |
| <i>4.7 Sequence diagram.....</i> | 54 |
| <i>4.8 User interface.....</i> | 87 |

Chapter5. Implementation(semester||)

Chapter6. Testing(semester||)

6.1 TDD

Conclusion

Future work

References.....90

Chapter 1

introduction

1.1 Overview

Our project aims to meet the needs of individuals who prefer traditional shopping over online methods. The basic concept revolves around an easy-to-use app where users can either take a photo of the desired product or upload it. The program uses advanced algorithms to match the product with products in the store closest to the user, prioritizing similarity, and then proceeds to locate the nearest store that sells the product. Our focus is not only to find the most similar product but also to ensure proximity and cost effectiveness. For these factors, our platform provides seamless connectivity between offline shoppers and local retailers, revolutionizing the traditional shopping experience.

1.2 Motivation

1-Convenience and time-saving: The application simplifies the search process by eliminating the need for users to visit multiple stores physically or spend a significant amount of time browsing online platforms. It centralizes the search in one place, making it more convenient and efficient.

2-Increased accessibility: The application helps users discover local stores they may not have been aware of before. It expands their options by presenting a broader range of places where they can find the desired item, increasing the chances of finding it quickly and easily.

3-Personalized recommendations: By utilizing user-uploaded pictures, the application can provide tailored recommendations based on the specific item the user wants to buy. This personalization enhances the overall user

experience and increases the likelihood of finding the exact item they are looking for.

4-Supporting businesses: The application promotes businesses by directing users to nearby stores that offer similar items. This can benefit both users and merchants by encouraging community engagement and supporting the economy.

1.3 Objective

The project is an application through which customer he photographs a specific product and application tells me the nearest branch found it. customer can see it with his eyes before he buys it. Therefore, this is useful because The person can see the product and evaluate it before he buys it without falling into any of the types fraud like when you buy any product online.

The project must be easy to use as many suitable products can be provided to people who need them due to the use of artificial intelligence, as it anyone can use it, as it can be available for all systems, And it is important that there is an evaluation after search to the product to see how good the research is. And client patients.

It will not be risk in implementing this project because we asked more than one person in different places about the idea, and the majority of them agreed at a rate of 90%. And we asked more than one vendor about the idea, and the majority of them agreed on this idea, and ready they gave us the data regarding the products.

1.4 Aim

The aim of our application to help people who don't like online shopping and faces a lot of problem with it in our application we can solve some of problems that you face in online shopping like when you order an item and you receive a wrong item or you receive your item but you found it in bad quality. In our application we make it easy so you can find your item easily and go check it by yourself we ease on you the process of find a product

1.5 Scope

The scope of our project is to develop a mobile app that allows users to take pictures of clothes and find nearby stores that carry the same products. The app will use location detection to provide customized results based on the user's current location. It aims to make shopping for specific clothing items more convenient and efficient.

Here are some more details for our project scope:

- 1. Image Recognition:** Implement advanced image recognition technology to accurately identify the t-shirt in the images taken by users.
- 2. Search Functionality:** Develop a search feature that matches the captured t-shirt image with similar products available in nearby stores.
- 3. Store Integration:** Integrate your app with store databases to provide real-time information on the availability, pricing, and location of the t-shirt in nearby stores.
- 4. Location Detection:** Utilize location detection to customize search results and display the nearest stores that carry the desired t-shirt.
- 5. Store Details:** Provide users with detailed information about the stores, including their addresses, contact information, and operating hours.

1.6 General constraints

Time: It refers to the project's schedule and the deadlines that need to be met. Managing time effectively is crucial for project success.

Cost: This constraint involves the project's budget and the financial resources allocated to it. Managing costs involves ensuring that the project stays within budget and optimizing resource allocation.

Risks: Risks are unforeseen events or circumstances that can impact the project. It is important to identify potential risks and develop risk management strategies to mitigate their impact on the project.

Scope: The scope defines the project's objectives, deliverables, and boundaries. Managing the scope involves ensuring that the project stays focused on its intended goals and does not expand beyond its original scope.

Resources: Resources include the people, equipment, and materials necessary for project execution. Managing resources involves allocating them effectively to ensure smooth progress and avoid resource shortages.

Quality: Quality refers to meeting the specified standards and expectations for project deliverables. Managing quality involves implementing processes and controls to ensure that the project meets the desired level of quality.

1.7 Organization of the dissertation

Chapter 1: provides an overview of our project, its motivation, its goal, its constraints, and its scope.

Chapter 2: give the background and the history of our project and our prior experience.

Chapter 3: Speaking of project planning, analysis, and limitations, this chapter will cover our functionality, user characteristics, and Gantt chart.

Chapter 4: It includes our UML diagrams as well as the assumption and design of the database.

Chapter 2

Background and previous work

2.1 Background

Online shopping application. Its purpose is to provide the customer with the thing he wants by photographing the product, then the application begins searching for it in the store closest to the customer, then tells him the location of the store and display all the details of item that the customer search for.

Online shopping refers to the process of buying goods or services over the internet. It enables consumers to browse products, add them to a virtual cart, and make purchases online. "E-commerce Get It Right!" provides practical advice for creating successful online businesses, covering topics such as website design, product presentation, secure payment methods, customer service, marketing, analytics, and legal compliance. The book offers insights to enhance the online shopping experience and optimize e-commerce platforms for better performance and customer satisfaction. (Wikipedia, ...)

E-commerce This model involves buying and selling products or services online. It can include business-to-consumer (B2C) e-commerce, where businesses sell directly to individual customers, or business-to-business (B2B) e-commerce, where businesses trade with other businesses online. There are various models in E-commerce one of the most common model is storefront model.
(techtarget, n.d.)

There are various models in E-commerce one of the most common model is storefront model.

In a storefront model, the e-commerce app serves as the virtual storefront or digital representation of the physical store. It typically includes features such as product catalogs, order processing, secure payment, timely order fulfillment, and customer reviews. The app is designed to showcase products or services and provide a user-friendly interface for customers to browse, select, and purchase items.

In the storefront model of e-commerce, an "online shopping mall" It is designed to provide customers with a one-stop destination for browsing and purchasing products or services from various sellers or brands the online shopping mall operates similarly to a physical shopping mall, where multiple stores are located under one roof. (Deitel, 2001)

In the book "The Long Tail: Why the Future of Business Is Selling Less of More" by Chris Anderson, the concept of the "Long Tail" refers to the phenomenon where the demand for niche or less popular products collectively surpasses the demand for popular mainstream products. This concept is particularly relevant in the context of online shopping and e-commerce.

In traditional brick-and-mortar retail, physical shelf space is limited. Therefore, stores tend to stock products that have proven to be popular and sell quickly. However, in the online world, retailers can offer a vast variety of products due to virtually unlimited digital shelf space. As a result, consumers have access to a wide range of niche, specialized, or less mainstream products that might not be readily available in physical stores.

Online shopping, in the context of the Long Tail theory, means the ability for consumers to explore and purchase these less popular or niche products. Online retailers, especially e-commerce platforms, leverage the Long Tail by offering diverse product selections, catering to various interests and preferences. This approach allows consumers to find products that align with their unique tastes, hobbies, or needs, even if these products are not bestsellers in the traditional sense.

in summary, online shopping, as described in "The Long Tail," means the ability for consumers to access a vast array of products, including niche and less popular items, thanks to the extensive product offerings of online retailers. It enables customers to explore a diverse range of options and make purchases based on their specific interests and requirements. (investopedia, n.d.)

2.2 previous work

2.2.1 Amazon

The biggest online retailer in the world as well as a well-known supplier of cloud services is Amazon (3.Amazon.com).

Amazon began as an online bookseller but has now evolved into a web-based company primarily focused on offering services related to digital streaming, cloud computing, e-commerce, and artificial intelligence (AI).

With an enormous product selection and inventory, the company uses an Amazon-to-buyer sales strategy, allowing customers to purchase nearly anything, including apparel, cosmetics, fine dining, jewelry, books, movies, electronics, pet supplies, furniture, toys, garden supplies, and household goods.

With its global headquarters located in Seattle, Amazon operates separate websites, customer support centers, software development centers, data centers, and fulfillment centers. (4.techttarget, n.d.)

Under a program known as Fulfillment by Amazon (FBA), vendors can use Amazon's fulfillment facilities to store their inventory. On behalf of traders, Amazon manages product delivery, packing, and storage. With FBA, vendors may take use of Amazon's extensive logistical network, which offers consumers prompt and dependable delivery. (5.sell.amazon, n.d.)

One criticism leveled against Amazon is the issue of fake products being sold on its platform. It has been alleged that some sellers are selling fake goods that look like well-known brands, raising questions about the reliability of the products, consumer safety, and product quality.

(6.criticism_of_Amazon, n.d.)

Similarities:

1. Product Search: Both SnapMart and the Amazon app involve searching for specific items.
2. Location-Based Services: Both apps utilize location-based services to provide relevant information to users. SnapMart aims to detect the user's location.
3. Product Information: Both apps provide product information to users. They can display details such as product descriptions, specifications, prices.
4. Visual Recognition: Both apps may incorporate visual recognition technology to identify products based on images.

Differences:

1. Platform Scope: SnapMart focus on connecting users with nearby physical stores that have the desired item. The Amazon app, primarily operates as an online marketplace, connecting users with sellers and facilitating online purchases.
2. Authentication: SnapMart provides an opportunity for users to physically inspect and authenticate the products before making a purchase. This reduces the risk of fake items.
3. Trust and Reliability: SnapMart aims to foster trust and reliability by relying on the reputation and credibility of stores. Users can have more confidence in the authenticity and quality of the products they find through the app, as they are purchasing from established physical retailers.
4. Economy Support: By promoting stores, SnapMart helps support the economy and encourages consumers to contribute to the growth of businesses in their community. This approach can be appealing to users who prioritize supporting businesses over online marketplaces.

(7.Wikipedia, ...)

2.2.2 AliExpress

Similarities:

1. Retail E-commerce Platform: Both SnapMart and AliExpress serve as online platforms for retail E-commerce, allowing users to browse and purchase products from multiple vendors.
2. Product Listings: Similar to AliExpress, SnapMart would feature product listings, including images, descriptions, and pricing.
3. Search and Filtering: Both platforms offer users the ability to search for specific products and filter results based on various criteria.
4. Visual Search Emphasis: SnapMart and AliExpress places emphasis on visual search, enabling users to find products by uploading images.

Differences:

1. AI-Powered Image Recognition: One key difference is the use of AI in SnapMart to analyze and provide detailed information about clothing items based on images. AliExpress doesn't offer such image recognition capabilities.
2. Data and Analytics: SnapMart includes advanced data analysis to provide the best results about products .(8.The Lean Startup - googling ., n.d.).
3. Visual Search Emphasis : SnapMart works to find a place to sell the product, unlike AliExpress, which works to display the product only on the APP

2.2.3 Taobao

Taobao is one of the largest and most popular online shopping platforms in China, owned by the Alibaba Group. It was launched in 2003 and has since become a leading e-commerce platform in China and various other regions. The Taobao app allows users to shop for a wide range of products, including fashion, electronics, home goods, beauty products, and more.

(9.Saporedicina.com, n.d.)

Similarity:

1. Product Search: Both SnapMart and the Taobao app involve searching for specific items.

2. Image recognition and visual search:

Both SnapMart and Taobao use image recognition and visual search technologies. Users can take photos of products they are interested in, and the app searches for similar products. This approach enhances the user experience by simplifying the search process.

3. Explore specialized products:

SnapMart and Taobao cater to the Long Tail market, offering a wide range of products, including specialty and specialty items. Users can explore unique, less mainstream products based on their specific interests, hobbies, or needs.

4. Local Shopping:

Both SnapMart and Taobao concepts focus on connecting users with stores. SnapMart aims to find the nearest store that sells the desired product, enhancing convenience for users who prefer traditional shopping. Likewise, Taobao allows users to discover products from local sellers, promote local businesses and enable local shopping experiences.

5. Enhance user engagement:

By integrating image recognition and localization technologies, both concepts enhance user engagement. Users can actively participate in the shopping process by taking photos of products they like, enhancing interactivity and personalization in their shopping experience.

Differences:

1. Focus on user experience:

SnapMart App: Your concept emphasizes the simplicity of taking a photo of the desired product and finding it in the nearest store. The primary focus is on the convenience of users who prefer traditional shopping methods.

Taobao App: Taobao offers an online shopping experience, allowing users to browse a range of products, compare prices and make purchases directly through the app. It meets the needs of users who prefer online shopping.

2. Shopping method:

SnapMart App: Your concept bridges the gap between online and offline shopping by using image recognition to locate products in stores. It targets users who want instant shopping but need help finding specific items.

Taobao App: Taobao is an online marketplace that allows users to shop for products from various categories, sellers and brands. It is designed for users who prefer to browse and purchase items online.

3. The target audience:

SnapMart App: Targets individuals who prefer traditional shopping methods, value proximity, and want to quickly locate specific products without extensive online browsing.

Taobao App: Caters to an audience of users who enjoy online shopping, want access to a wide range of products, and are willing to explore different options and sellers.

4. Geographic focus:

SnapMart: Primarily focused on helping users find products in nearby physical stores, with an emphasis on shopping experiences.

Taobao App: Mostly serves users in China, offering products from sellers.
(10World.Taobao, n.d.)

2.2.4 Walmart

Walmart is a multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores. It is one of the largest retail companies in the world (11Walmart, n.d.)

Walmart offers a wide range of products, including groceries, electronics, clothing, home goods, furniture, toys, sporting goods, and more. It is known for its "everyday low prices" strategy, aiming to provide affordable products to consumers. Walmart operates both physical stores and an online platform, allowing customers to shop in-store or make purchases online for home delivery or store pickup.

Walmart stores are typically large and offer a one-stop shopping experience, fulfilling various consumer needs. Many Walmart stores also have additional services like pharmacies, automotive centers, and financial services.

Similarities:

1. Wide Product Selection: Walmart and SnapMart offers a vast range of products across various categories, apparel, and more. This wide selection attracts customers looking for a convenient one-stop shopping experience.
2. Search and Filtering: Both platforms offer users the ability to search for specific products and filter results based on various criteria.
3. Store Network: With numerous physical stores located in various regions, this widespread presence allows to reach a large customer base and offer convenience through local accessibility.

Differences:

1. SnapMart provides easy discovery of items and their different types through product photography.
2. SnapMart provides the closest place where the product you are looking for is available.
3. SnapMart offers a unique feature where it provides AI-Powered Image Recognition: One key difference is the use of AI in SnapMart to analyze and provide detailed information about clothing items based on images.

2.2.5 Jumia

Jumia is a Pan-African technology company that is built around a marketplace, logistics service and payment services. The logistics service enables the delivery of packages through a network of local partners while the payment services facilitate the payments of online transactions within Jumia's ecosystem. It has partnered with more than 100,000 active sellers and individuals and it competes at various levels.

Jumia, Egypt's most convenient online shopping mall, serves you in the best way possible. Our aim is to provide you with the best-in-class products whether you are in Cairo, Alexandria, or any other city in Egypt. On Jumia, you can find a wide range of products from the best brands at the best price and outstanding quality. Because Jumia is an online shop for the whole family, here you can find everything you need for your kids.

Easy-to-use website makes your shopping experience unique and very convenient. Jumia, Egypt's great online market, offers you all the products you might need; high-tech, home and kitchen appliances, mobiles, home decorations, perfumes, watches, games, computers, and tablets beside our big fashion store in which you can shop for clothing, shoes, accessories, sunglasses and more (12.sellercenter, n.d.)e.

Similarities

1. Product variety: Jumia provides a wide range of products across various categories. Users can browse through a diverse selection of items and choose the ones that suit their needs.
3. Search and Filtering: Both platforms offer users the ability to search for specific products and filter results based on various criteria.
4. Product Search: Both platforms involve searching for specific items.
5. Retail E-commerce Platform: Both platforms serve as online platforms for retail E-commerce, allowing users to browse and purchase products from multiple vendors.
6. Product Listings: Both platforms would feature product listings, including images, descriptions, and pricing.

Differences:

1. Confirmation of the product: our product provides you the experience of confirmation of the product to make sure that is the right product you want before you buy it.
2. Image Recognition: we use artificial intelligence to analyze and provide detailed information about clothing items based on images that the customer provides us with it
3. Locations and nearby places: our application provides you with the necessary information about the item you search for where you can find it and its specific place

2.2.6 Noon

Similarities:

- 1.Photo Capture: Both SnapMart and Noon involve capturing photos of products. In SnapMart , users capture photos to upload and find similar products, while in Noon, users can also upload product photos to search for similar items.
- 2.Recommendation System: Both SnapMart and Noon utilize recommendation systems. SnapMart recommends local and global shops that have similar products based on user-uploaded photos and shop data. Noon, on the other hand, recommends products based on user preferences, browsing history, and other data.
- 3.Product Matching: Both SnapMart and Noon involve matching products based on their visual similarity. SnapMart matches user-uploaded photos with similar products in shops, while Noon matches user-uploaded photos with similar products available on their platform. (13.NOON, n.d.)e.

Differences:

- 1.Scope: SnapMart focuses on recommending shops that have the same product as the one uploaded by the user. Noon, on the other hand, is a comprehensive e-commerce platform that offers a wide range of products across different categories, including electronics, fashion, home appliances, and more.
- 2.Data Source: In SnapMart , the data of shops that users enter is used to recommend the nearest shops with similar products. In contrast, Noon has its own inventory of products and relies on sellers to list their items on the platform.

Chapter 3

Planning and Analysis

3.1 Planning

The Software Development Life Cycle (SDLC) offers an organized method for creating and delivering the software solution for my project. An outline of the standard steps in the SDLC is provided below:

1. System Analysis: During this phase, you analyze the gathered requirements in detail. This includes identifying potential challenges, defining system architecture, and determining the technical feasibility of the project. You may also create system design specifications and establish project timelines.

- 1 User requirement**
 - 1.1 System requirement**
 - 1.2 Functional requirement**
 - 1.3 Non-functional requirement**

2. System Design: In this phase, you translate the requirements into a detailed system design. This involves creating high-level and low-level designs, specifying the system's structure, modules, interfaces, and data flow. It may also include database design, user interface design, and defining the technology stack.

- 1. Database design**
- 2. ERD**
- 3. Class diagram**
- 4. Use case diagram**
- 5. Sequence diagram**
- 6. Interface design**
- 7. Activity diagram**

3.Coding: The development phase involves coding and implementation based on the design specifications. Programmers and developers write the necessary code, perform unit testing, and integrate the individual components together to build the complete system.

1. UI Client
2. DB Development
3. API Development
4. Data Science
5. Security

4.Testing: The testing phase is crucial for ensuring the quality and reliability of the software solution. It includes various types of testing, such as unit testing, integration testing, system testing, and acceptance testing. Bugs and issues are identified, documented, and fixed during this phase.

1. TDD
2. Test levels
 - 2.1 test planning
 - 2.2 test analysis
 - 2.3 test design
 - 2.4 test implementation
 - 2.5 test execution
 - 2.6 test completion
3. Test types
 - 3.1 Black box Testing (functional, Non-functional, regression)
 - 3.2 White box

5.Deployment: Once the software solution passes all testing phases, it can be deployed to the production environment. This involves installing the system, configuring the necessary hardware and software, and making it available for end-users.

6.Maintenance: After deployment, the system requires ongoing maintenance and support. This phase involves monitoring the system's performance, addressing any issues or bugs that arise, and making necessary updates or enhancements based on user feedback or changing requirements.

| 1 | task | Predecessors | Estimated time "wide Band Delphi" | | | | | | |
|----|--|---------------|-----------------------------------|----------|-------|-----|---------|-------|-------|
| | | | Abdelaziz | Abdelhay | Basma | Mai | Mohamed | Salma | Total |
| 2 | Task Name | | 60 | 39 | 45 | 60 | 50 | 45 | 50 |
| 3 | system analysis | | | | | | | | |
| 4 | UR1 | | 9 | 11 | 10 | 9 | 10 | 13 | 10 |
| 5 | System requirement | UR1 | 2 | 4 | 4 | 3 | 2 | 3 | 3 |
| 6 | 1.1.1 login | UR1 & FR1.2.1 | 2 | 4 | 4 | 3 | 2 | 3 | 3 |
| 7 | Functional testing | UR1 | 1 | 2 | 3 | 4 | 3 | 5 | 3 |
| 8 | 1.2.1 create new accounts | UR1 & SR1.1.1 | 1 | 2 | 3 | 4 | 3 | 5 | 3 |
| 9 | non-function requirements | UR1 & FR1.1 | 1 | 1 | 3 | 5 | 2 | 2 | 2 |
| 10 | 1.3.1 security | UR1 & FR1.1 | 1 | 1 | 3 | 5 | 2 | 2 | 2 |
| 11 | UR2 | UR1 | 10 | 20 | 24 | 17 | 19 | 31 | 20 |
| 12 | system requirements | UR1 | 3 | 4 | 6 | 7 | 2 | 3 | 4 |
| 13 | 2.1.1 image recognition | FR2.2.1 | 7 | 1 | 1 | 1 | 1 | 1 | 1 |
| 14 | 2.1.2 location | FR2.2.3 | 1 | 1 | 4 | 1 | 1 | 1 | 1 |
| 15 | 2.1.3 mapping service | FR2.2.3 | 1 | 1 | 1 | 1 | 1 | 6 | 1 |
| 16 | function requirements | UR2 | 11 | 12 | 8 | 9 | 6 | 8 | 9 |
| 17 | 2.2.1 image capture and upload | UR1 | 3 | 3 | 2 | 4 | 1 | 5 | 3 |
| 18 | 2.2.2 image recognition | UR1 & FR2.2.1 | 3 | 3 | 2 | 4 | 1 | 5 | 3 |
| 19 | 2.2.3 integration with a mapping service | UR1& FR2.2,2 | 2 | 1 | 1 | 3 | 2 | 5 | 2 |
| 20 | non function requirements | FR2.2 | 6 | 8 | 5 | 5 | 9 | 4 | 6 |
| 21 | 2.3.1 responsiveness | FR2.2 | 1 | 6 | 1 | 1 | 1 | 1 | 1 |
| 22 | 2.3.2 high accuracy | FR2.2 | 1 | 1 | 1 | 7 | 1 | 1 | 1 |
| 23 | 2.3.3 accessibility of DB | FR2.2 | 1 | 1 | 4 | 1 | 1 | 1 | 1 |
| 24 | 2.3.4 reliability of API | FR2.2 | 1 | 1 | 1 | 1 | 1 | 4 | 1 |
| 25 | 2.3.4 protection and privacy regulations | FR2.2 | 1 | 1 | 1 | 1 | 1 | 4 | 1 |
| 26 | UR3 | UR1 | 11 | 19 | 17 | 31 | 24 | 20 | 20 |
| 27 | system requirements | UR1 | 7 | 9 | 3 | 8 | 9 | 4 | 7 |
| 28 | 3.1.1 Add items | FR3.2.1 | 1 | 3 | 2 | 1 | 2 | 6 | 2 |
| 29 | 3.1.2Categorize items | SR3.1.1 | 1 | 3 | 2 | 1 | 2 | 6 | 2 |
| 30 | 3.1.3 manage availability | SR3.1.1 | 1 | 3 | 2 | 1 | 2 | 6 | 2 |
| 31 | Function requirements | UR1 | 8 | 2 | 5 | 6 | 8 | 5 | 6 |
| 32 | 3.2.1 create inventory | UR1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 |
| 33 | 3.2.2 add new items | FR3.2.1 | 2 | 2 | 4 | 1 | 3 | 1 | 2 |
| 34 | 3.2.4 categorization feature | FR3.2.2 | 2 | 2 | 4 | 1 | 3 | 1 | 2 |
| 35 | 3.2.5manage the availability | FR3.2.2 | 1 | 1 | 1 | 5 | 1 | 1 | 1 |
| 36 | non-function requirements | FR3.2 | 7 | 3 | 6 | 6 | 5 | 3 | 5 |
| 37 | 3.3.1 efficiency | FR3.2 | 1 | 1 | 1 | 6 | 1 | 1 | 1 |
| 38 | 3.3.2 flexibility | FR3.2 | 1 | 1 | 5 | 1 | 1 | 1 | 1 |
| 39 | 3.3.3 availability | FR3.2 | 3 | 5 | 4 | 3 | 2 | 2 | 3 |
| 40 | System design | | 30 | 35 | 25 | 30 | 20 | 40 | 30 |
| 41 | Database Design | | 10 | 6 | 9 | 11 | 6 | 7 | 8 |
| 42 | Class Diagram | | 4 | 8 | 5 | 7 | 5 | 7 | 6 |
| 43 | Usecase Diagram | | 4 | 8 | 5 | 7 | 5 | 7 | 6 |
| 44 | Sequence Diagram | | 4 | 8 | 5 | 7 | 5 | 7 | 6 |
| 45 | Interface Design | | 3 | 5 | 7 | 1 | 5 | 3 | 4 |
| 46 | coding | | 70 | 80 | 80 | 60 | 80 | 70 | 75 |
| 47 | UI Client | | 10 | 10 | 9 | 10 | 14 | 10 | 10 |
| 48 | DB Development | | 25 | 15 | 28 | 22 | 25 | 30 | 25 |
| 49 | API Development | | 15 | 15 | 16 | 16 | 14 | 11 | 15 |
| 50 | Data Science | | 19 | 25 | 20 | 20 | 21 | 15 | 20 |
| 51 | Security | | 6 | 7 | 10 | 7 | 8 | 10 | 8 |
| 52 | Testing | | 50 | 60 | 55 | 55 | 60 | 40 | 55 |
| 53 | TDD | | 11 | 11 | 7 | 9 | 15 | 9 | 10 |
| 54 | Test Levels | | 20 | 50 | 49 | 45 | 41 | 41 | 44 |
| 55 | Test planning | | 7 | 8 | 5 | 8 | 6 | 7 | 7 |
| 56 | Test analysis | | 7 | 8 | 5 | 8 | 6 | 7 | 7 |
| 57 | Test design | | 12 | 9 | 13 | 9 | 9 | 10 | 10 |
| 58 | Test implementation | | 10 | 28 | 25 | 35 | 17 | 10 | 20 |
| 59 | Test excitation | | 5 | 5 | 4 | 5 | 7 | 5 | 5 |
| 60 | Test completion | | 12 | 9 | 13 | 9 | 9 | 10 | 10 |
| 61 | Maintenance | | | | | | | | |
| 62 | Launch | | | | | | | | |

3.1.1 Feasibility study and estimated cost

Feasibility Study:

1. Technical Feasibility: Evaluate the technical aspects of the project, including the availability of image recognition algorithms, APIs for store and location data, and the required infrastructure for hosting and scaling the system.
2. Market Feasibility: Conduct market research to assess the demand for such a system. Determine if there is a target audience and if they would find value in the proposed solution.
3. Financial Feasibility: Assess the financial viability of the project. Consider the potential revenue streams, such as advertisements, partnerships with stores, or premium features. Evaluate the projected costs and potential return on investment.vc
4. Legal and Ethical Feasibility: Ensure compliance with privacy laws and regulations regarding the handling of user data. Consider any ethical implications related to image recognition and location tracking.
5. Operational Feasibility: Evaluate the operational requirements of the project, including staffing, maintenance, and ongoing support. Consider if the necessary resources and expertise are available to successfully implement and manage the system.

Estimated Cost:

1. Development Costs: This includes expenses related to hiring developers, designers, and other technical personnel, as well as the cost of software development tools and licenses. (Team members are the developers)
2. Infrastructure Costs: Consider the costs associated with hosting the system, including servers, storage, and network infrastructure. Cloud-based solutions may incur ongoing costs based on usage (\$5).
3. Data Costs: If you plan to utilize external data sources or APIs for store and location information, consider any associated costs for data access or subscriptions (($\$0.50 - \5) per thousand requests).
4. Integration Costs: If you need to integrate with mapping services, social media platforms, or other third-party systems, consider any associated integration costs (\$20).
5. Testing and Quality Assurance:

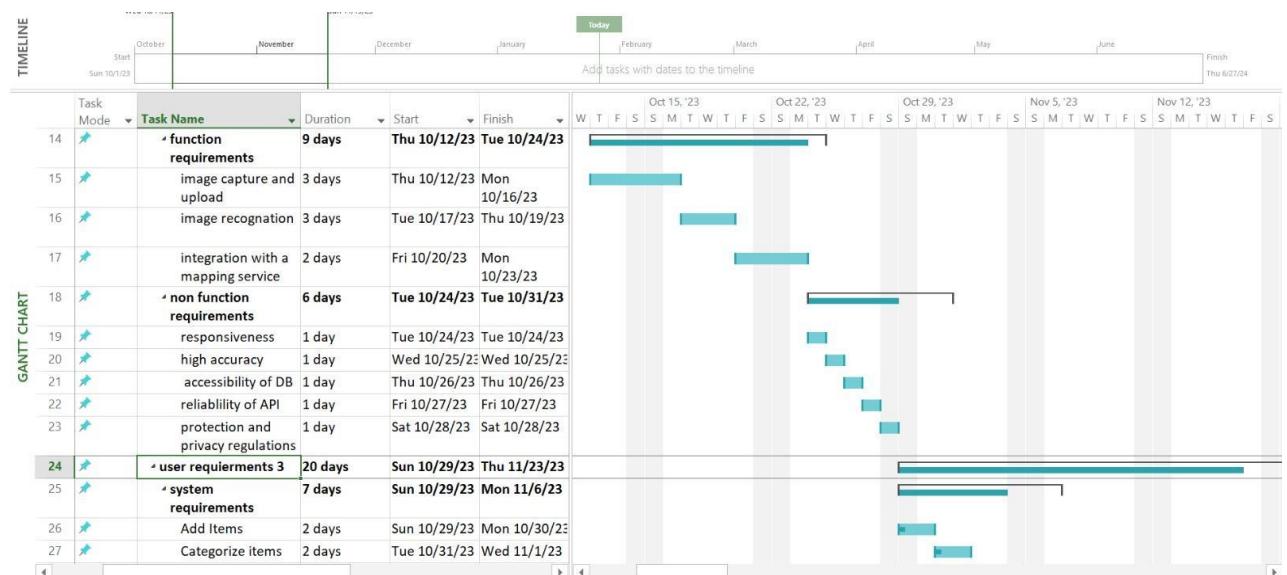
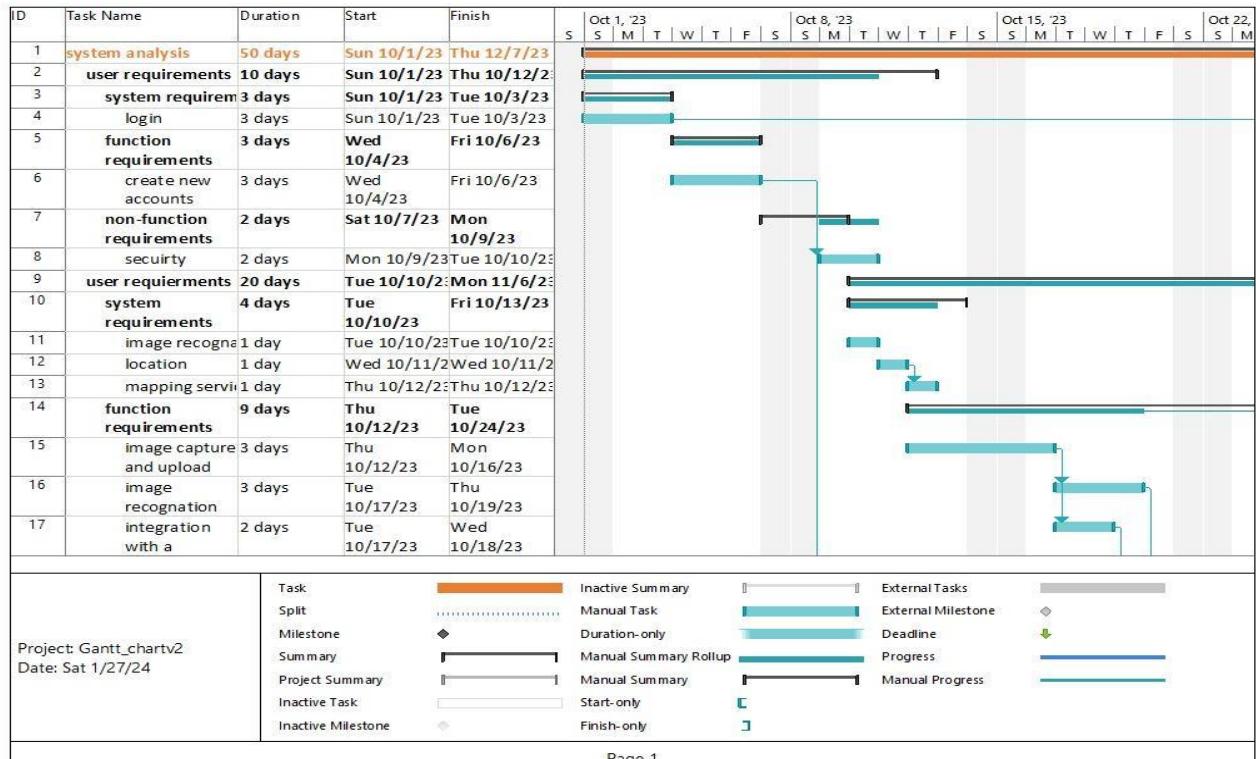
Comprehensive testing, including unit testing, integration testing, and user acceptance testing, is crucial. The extent of testing required will affect the development timeline and cost (\$20).

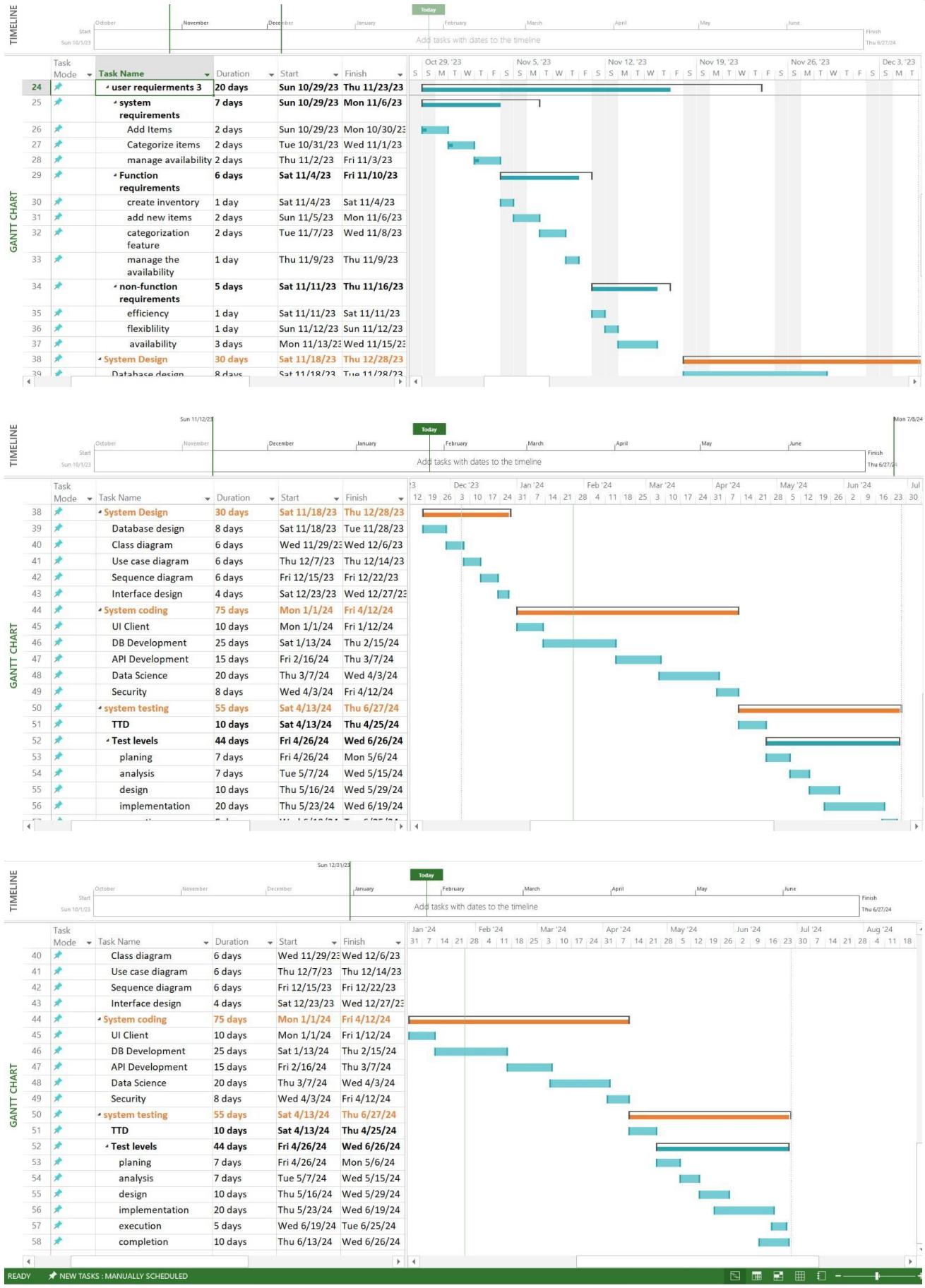
6. Maintenance and Support Costs: Factor in ongoing expenses for system maintenance, updates, bug fixes, and user support. This may include personnel costs or fees for outsourcing support services. (\$100)
7. Marketing and Promotion Costs: Consider expenses related to marketing and promoting the application to attract users and drive adoption. (\$45)
8. Project Management:

Efficient project management is essential for successful development. Costs may include project management tools, communication platforms, and the time invested by project managers. (\$150)

Total cost: \$345 (Actual costs for each of these categories can vary)

3.1.2 Gantt chart





3.2 Analysis and limitations of existing system

1. Functional Analysis:

- Analysis: Evaluate the functionality of the SnapMart e-commerce platform, including features such as product search, product listings and management.
- Limitation: The platform does not contain online shopping or any transaction process because it acts as an interface between user and vendor

2. Technical Analysis:

- Analysis: Assess the technical aspects of the SnapMart, including its performance, scalability, and reliability. Evaluate factors such as application loading times, uptime, and responsiveness.
- Limitation: Network Connectivity and Speed of the mobile application heavily relies on network connectivity to upload images, process them, and retrieve search results. Limited or unreliable network connectivity can impact the user experience and the speed at which search results are delivered.

3. Usability Analysis:

- Analysis: Evaluate the user experience and interface design of SnapMart, considering factors such as ease of navigation, product search functionality, and clarity of product information.
- Limitation: Limited Mobile Responsiveness for mobile applications, limitations in responsive design and optimization for different devices can impact usability. Inconsistent layouts, poor touch interactions, and difficulty in viewing or interacting with content on smaller screens can affect the user experience.

4. Security Analysis:

- Analysis: Assess the security measures implemented on SnapMart, including data encryption, user authentication, and protection of personal information.

- limitation: Complexity of Third-Party Integrations: Integrating with external systems, APIs, or third-party libraries introduces potential security risks. However, conducting a security analysis for these integrations can be challenging due to limited access to their source code or documentation. Depending on the level of cooperation from third parties, it may be difficult to assess the security posture fully.

5. Customer Service Analysis:

- Analysis: Evaluate the effectiveness and responsiveness of SnapMart customer service, including channels for contacting support, issue resolution timeframes, and overall customer satisfaction.

- limitation: Difficulty in Measuring Customer Satisfaction Customer service analysis often aims to measure customer satisfaction, which can be a complex and multifaceted concept. It may be challenging to define and quantify customer satisfaction accurately, as it can be influenced by various factors, including expectations, prior experiences, and individual perceptions.

6. Expansion Analysis:

- Analysis: Assess Amazon's expansion strategies and capabilities, including the availability of services like SnapMart.

- Limitation: Not all SnapMart services and features are available in every country.

7. Product Diversity and Curation:

- Analysis: Evaluate the diversity and curation of products available on SnapMart, including the range of categories, brands, and product quality.
- Limitation: customers may report challenges in finding niche or specialized products, as the platform tends to prioritize popular or mainstream items, potentially limiting options for specific customer needs.

8. Ethical and Sustainability Considerations:

- Analysis: Assess SnapMart's practices and initiatives related to ethical sourcing, environmental sustainability, and social responsibility.
- limitation: Privacy Concerns The app's functionality, such as taking photos and analyzing them, raises privacy concerns. Users may be hesitant to grant access to their camera or share images due to potential misuse or unauthorized access to their personal data. Balancing the need for functionality with privacy protection can be challenging.

9. Accessibility and Inclusivity:

- Analysis: Assess the accessibility features and inclusivity efforts of the SnapMart,
- Limitation: Some users may report challenges with website accessibility, such as issues with screen reader compatibility

10. Data Privacy and Personalization:

- Analysis: Evaluate SnapMart's data privacy practices and the balance between personalized recommendations and customer privacy.
- limitation: Third-Party Integrations If the app integrates with external systems, APIs, or third-party services, it may involve sharing user data with those entities. Ensuring the privacy practices and data handling policies of third parties align with your app's privacy standards can be challenging.

11. Competitive Landscape:

- Analysis: Assess the competitive landscape in which SnapMart operates, considering the presence of other e-commerce platforms and potential disruptors in the market.
- Limitation: Increasing competition from other major e-commerce players and emerging direct-to-consumer.

12. Geographic Reach and Infrastructure:

- Analysis: Evaluate SnapMart's infrastructure, including the availability and reliability of services in different regions.
- Limitation: may have limited access to certain services or face logistical challenges due to infrastructure constraints, impacting delivery times and customer experience.

13. Innovation and Differentiation:

- Analysis: Assess SnapMart's ability to innovate and differentiate its offerings, including the development of new services or technologies.
- limitation: User Expectations Users often have specific expectations and preferences based on their previous experiences with similar apps. Striking a balance between meeting user expectations and offering something different can be tricky, as too much deviation from established patterns may lead to user confusion or resistance.

3.3 Need for new system

Object Recognition:

Object recognition refers to the ability of an AI system to identify and classify objects within an image. By implementing object recognition algorithms, you can train your system to recognize specific items or products in the images uploaded by users. This involves using deep learning techniques, such as convolutional neural networks (CNNs), to analyze and classify the objects present in the images accurately.

Image Processing and Enhancement:

Image processing involves applying various techniques and algorithms to manipulate and enhance digital images. In the context of your project, image processing can be used to improve the quality and clarity of the uploaded images. Techniques such as noise reduction, image denoising, contrast adjustment, and image sharpening can be applied to enhance the details and make the object recognition algorithms more effective.

Product Availability and Pricing:

Integrating your system with e-commerce platforms or product databases allows you to retrieve real-time information about the availability and pricing of the identified item. By connecting to these platforms, users can receive up-to-date information on where they can purchase the item and compare prices from different sellers. This feature provides convenience and helps users make informed decisions about their purchases.

Geolocation and Store Mapping:

Leveraging geolocation data, your system can identify the user's location and provide information about nearby physical stores that carry the desired item. By integrating store mapping functionality, you can guide users to the nearest physical stores that have the item in stock. This feature helps users find convenient and accessible locations to make their purchases.

3.4 Analysis of new system

User Requirement 1:

I want to be able to login or sign up to the app and capture an image of an object using the mobile app and receive a list of the nearest stores that have the same item in stock based on their current location. The app should provide store details, including the store name, address, contact information, and distance from the user's location. Additionally, the user should have the ability to navigate to the selected store using integrated mapping or navigation tools.

| System requirement |
|--|
| SR 1.1 The system should provide a login and sign up page for user. |
| Functional requirement |
| FR 1.1 The system should allow users to create new accounts by providing necessary information such as username, email address, and password. And have authenticated registered users by verifying their credentials during the login process. |
| Non-functional requirement |
| NFR 1.1 The system should securely store and encrypt user passwords to protect sensitive user data. |

User Requirement 2:

I want to capture an image of an object using the mobile app and receive a list of the nearest stores that have the same item in stock based on their current location. The app should provide store details, including the store name, address, contact information, and distance from the user's location. Additionally, the user should have the ability to navigate to the selected store using integrated mapping or navigation tools

| System requirement |
|--|
| SR 2.1 The system should provide a user interface that allows users to capture and upload images. |
| SR 2.2 The system should have an image recognition algorithm capable of accurately identifying objects in uploaded images. |

| |
|---|
| SR 2.3 The system should have access to a database of store locations and inventory information |
| SR 2.4 The system should integrate with a mapping service or API to retrieve distance and navigation information. |
| SR 2.5 The system should implement robust security measures to protect user data. |
| Functional requirement |
| FR 2.1 The system should have an image capture feature that enables users to take photos within the app. |
| FR 2.2 The system should utilize an image recognition algorithm to analyze and identify objects in the uploaded images by integrating with the database. |
| FR 2.3 The system should integrate with a mapping service or API to retrieve distance and navigation information between the user's location and the nearest stores. |
| Non-functional requirement |
| NFR 2.1 The image capture and upload process should be fast and responsive to provide a smooth user experience. |
| NFR 2.2 The image recognition algorithm should have a high accuracy rate in identifying objects. And the process should be efficient and provide quick results. |
| NFR 2.3 The database should be accessible and responsive to ensure that store information and inventory data are up to date. And should be able to handle a large number of store locations and inventory entries. |
| NFR 2.4 The mapping service or API should provide accurate and reliable distance and navigation information. And the integration with the mapping service or API should be seamless and provide a smooth user experience. |
| NFR 2.5 The system should comply with industry standards and best practices to ensure the confidentiality, integrity, and availability of user data. And should adhere to relevant data protection and privacy regulations. |

User Requirement 3:

The vendor should be able to add new items to their inventory, update existing item information, and manage stock availability using the mobile app. The app should provide a user-friendly interface for the vendor to easily categorize their items.

| System requirement |
|--|
| SR 3.1 The system should provide a login and sign up page for vendors. |
| SR 3.2: The system should allow vendors to add new items to their inventory by providing information such as item name, description, price, and quantity. And can update items. |
| SR 3.3: The system should allow vendors to categorize their items into different categories or tags |
| SR 3.4 The system should provide a mechanism for vendors to manage the availability of their items in real-time. |
| Functional requirement |
| FR 3.1 The system should allow vendors to create new accounts by providing necessary information such as username, email address, and password. And have authenticated registered vendors by verifying their credentials during the login process. |
| FR 3.2 The system should provide an interface for vendors to add new items to their inventory by entering information such as item name, description, price, and quantity. And should be able to update existing item information, including price, description, and quantity. |
| FR 3.3 The system should provide a categorization feature where vendors can assign items to different categories or tags. And should be able to create and manage their own categories or tags for organizing their inventory. |
| FR 3.4 The system should provide functionality for vendors to manage the availability of their items in real-time. |
| Non-functional requirement |
| NFR 3.1 The system should securely store and encrypt vendor passwords to protect sensitive user data. |
| NFR 3.2 The system should handle the addition and update of items efficiently, even with a large inventory. And The item management interface should be intuitive and easy to navigate for vendors |
| NFR 3.3 The system should support a flexible categorization system that allows vendors to create and manage their own categories or tags. |
| NFR 3.4 The system should update stock availability information in real-time to ensure accurate information for vendors and customers. |

Domain requirement

Domain Requirement 1:

The app should support a wide range of product categories.

Domain Requirement 2:

The app should have access to an extensive database of store locations, including both large retail chains and local businesses.

Domain Requirement 3:

The app should utilize accurate and up-to-date geolocation data to determine the user's current location.

Domain Requirement 4:

The app should integrate with popular mapping services, such as Google Maps or Apple Maps, to provide accurate store locations and directions to users.

Domain Requirement 5:

The app should leverage advanced image recognition technologies, such as machine learning algorithms or computer vision techniques, to accurately identify and match objects in uploaded images.

Domain Requirement 6:

The app should continuously update and expand its database of items and store information to ensure comprehensive and reliable search results.

Domain Requirement 7:

The app should comply with relevant data protection and privacy regulations, ensuring that user data is securely stored and processed.

Domain Requirement 8:

The app should consider regional or country-specific factors, such as different store chains, availability of certain products, or local shopping preferences, to provide relevant search results to users in different locations.

Domain Requirement 9:

The app should consider potential variations in product availability, pricing, and promotions across different stores, ensuring that users receive accurate and updated information.

Domain Requirement 10:

The app should provide support for multiple languages and localization features to cater to users from different regions and cultural backgrounds.

3.5 Advantages of new system

1. Convenience for Users:

- Easy Access: Users can quickly access information about a product by taking a

Picture, eliminating the need for manual searches.

- Time Savings: Users save time by avoiding the need to manually input product details or search for information online.

2. Real-time Information:

- Up-to-Date Data: Users can receive real-time information about product Availability, prices, and locations, ensuring that the data is current and accurate.

3. Location-Based Services:

- Geolocation: The system can provide location-based results, helping users find

Nearby stores or e-commerce options where the product is available.

- In-Store Navigation: For physical stores, the system can guide users to the exact location of the product within the store.

4. Cost Comparison:

- Price Transparency: Users can easily compare prices from different sources, helping them make informed purchasing decisions.
- Deals and Discounts: The system can also highlight any ongoing promotions or discounts available for the product.

5. Improved Shopping Experience:

- Enhanced Product Discovery: Users may discover new products and alternatives when searching for specific items.
- Reduced Friction: Simplified processes reduce the friction associated with shopping, leading to a better overall experience.

6. Business Benefits:

- Increased Sales: Businesses can benefit from increased visibility and easier access to their products, potentially leading to higher sales.
- Customer Insights: The system can collect data on user preferences and behavior, providing valuable insights for businesses.

7. Sustainability:

- Reduced Wastage: By helping users find products more easily, the system can contribute to reducing wastage and overconsumption.

8. Competitive Advantage:

- Businesses that offer such a system can gain a competitive edge by providing a unique and innovative service.

3.6 User characteristics:

Mobile users:

Attributes: Users who primarily access applications through smartphones and tablets.

Consideration: Design a mobile-friendly interface with responsive layouts and intuitive touch gestures for smooth navigation.

Photography lovers:

Features: Users feel comfortable taking and uploading high-quality photos.

Consideration: Make sure the app accommodates high-resolution image uploads

and provides immediate feedback on image recognition accuracy

Preference for traditional shopping:

Attributes: Users who prefer visiting physical stores rather than shopping online.

Consideration: Providing a seamless bridge between online and offline shopping

experiences, with an emphasis on ease of locating products in nearby stores.

Local Shoppers:

Features: Users who prefer to shop from nearby physical stores.

Consideration: Prioritize search results based on proximity, availability of the products.

Non-tech users:

Attributes: Users with limited experience using complex digital applications.

Consideration: Create an easy-to-use interface with straightforward navigation, minimal steps, and clear instructions to improve accessibility.

Impatient users:

Features: Users are looking for fast and efficient solutions without unnecessary delays.

Consideration: Improving the speed of the application, providing fast image recognition results, and quick access to store information

Privacy-conscious users:

Attributes: Users concerned about the privacy and security of their personal data.

Consideration: Clearly communicating app privacy policies, providing subscription

features for location services, and ensuring the secure handling of user information

Chapter 4

Design

4.1 Design and Implementation Constraints

Technological Constraints:

Image Recognition Technology: The accuracy and efficiency of the image recognition algorithm may be a constraint, as it directly affects the app's ability to identify the item in the picture.

Compatibility with Mobile Devices: The app must be compatible with various mobile devices, considering different screen sizes, resolutions, and operating systems.

Database and Integration Constraints:

Product Database: The app relies on a comprehensive and up-to-date product database. Ensuring the accuracy and completeness of this database may pose a challenge.

Integration with Store Databases: Integration with store databases to check for item availability requires collaboration with different retailers, and their varying data formats can be a constraint.

Location Services:

Accuracy of Location Services: The effectiveness of finding the nearest store depends on the accuracy of location services. Issues like GPS accuracy and signal strength can be constraints.

Privacy Regulations: Compliance with privacy regulations and user consent for location tracking may pose constraints.

Usability Constraints:

User Interface Design: Designing an intuitive and user-friendly interface, especially for the image capture feature, is crucial for user adoption.

Performance Constraints:

Response Time: The app should provide quick responses to user queries. Slow performance could impact user satisfaction.

Bandwidth and Data Usage: Optimizing the app to work efficiently with limited bandwidth and minimizing data usage is important, especially for users with limited data plans.

Security Constraints:

Data Security: Ensuring the security of user data, especially images, and transaction details is critical. Secure communication with store databases is also essential.

Authentication and Authorization: Implementing secure user authentication and authorization mechanisms is crucial.

Regulatory Constraints:

Data Privacy Regulations: Compliance with data protection and privacy regulations is a constraint that needs careful consideration.

Location-based Regulations: Adhering to local regulations regarding location-based services and data collection is important.

Integration with External APIs:

API Availability: Relying on external APIs for store information may be a constraint if these APIs are not always available or reliable.

Cost Constraints:

Development Costs: Developing and maintaining the app within a budget may be a constraint, especially if you need to license third-party technologies or services.

4.2 Assumptions and dependencies

Assumptions:

User Cooperation:

Assumption: Users will be willing to take pictures of items and use the app to find them in nearby stores.

Impact: If users are not comfortable with the image capture feature or find it inconvenient, it could affect the app's adoption and success.

Accurate Image Recognition:

Assumption: The image recognition technology will accurately identify items based on user-submitted pictures.

Impact: If the image recognition algorithm is not reliable, users may get incorrect search results, leading to dissatisfaction.

Up-to-Date Product Database:

Assumption: The product database is regularly updated with accurate information about available items in various stores.

Impact: If the database is outdated or incomplete, users may not find the items they are looking for, affecting the app's usefulness.

Stable Internet Connection:

Assumption: Users will have a stable internet connection for real-time querying of store databases.

Impact: Poor connectivity could result in slow responses or failure to fetch updated store information.

User Location Accuracy:

Assumption: Location services will provide accurate information about the user's location.

Impact: Inaccurate location data may lead to incorrect store recommendations, impacting the app's effectiveness.

Compliance with Privacy Regulations:

Assumption: The app's privacy policies and data handling practices comply with relevant regulations.

Impact: Non-compliance could lead to legal issues and damage the app's reputation.

Dependencies:

Third-Party APIs:

Dependency: The app relies on external APIs to fetch store information.

Impact: Changes or disruptions to these APIs could affect the app's functionality.

Collaboration with Retailers:

Dependency: Collaboration with retailers is required for integrating the app with their databases.

Impact: Delays or difficulties in securing collaborations could affect the app's ability to provide accurate store information.

Technological Dependencies:

Dependency: The app depends on specific technologies for image recognition, location services, and other functionalities.

Impact: Changes or advancements in these technologies may require updates to the app.

Budget and Resource Dependencies:

Dependency: The project is dependent on the availability of budget and resources for development, marketing, and maintenance.

Impact: Insufficient resources may lead to delays or compromises in the project.

Regulatory Changes:

Dependency: The app is subject to existing and future regulations related to privacy and data protection.

Impact: Changes in regulations may necessitate updates to the app to ensure compliance.

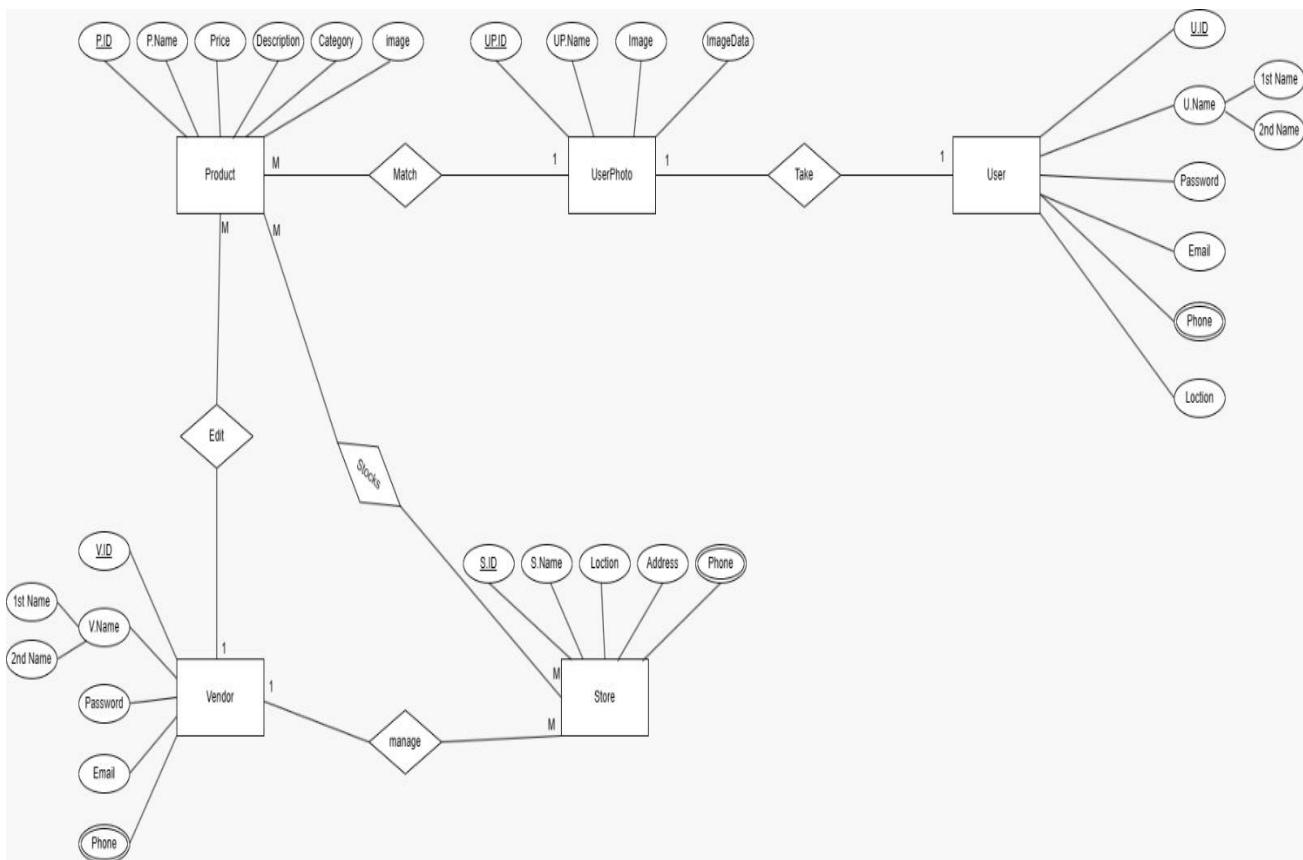
Device Compatibility:

Dependency: The app needs to be compatible with various mobile devices.

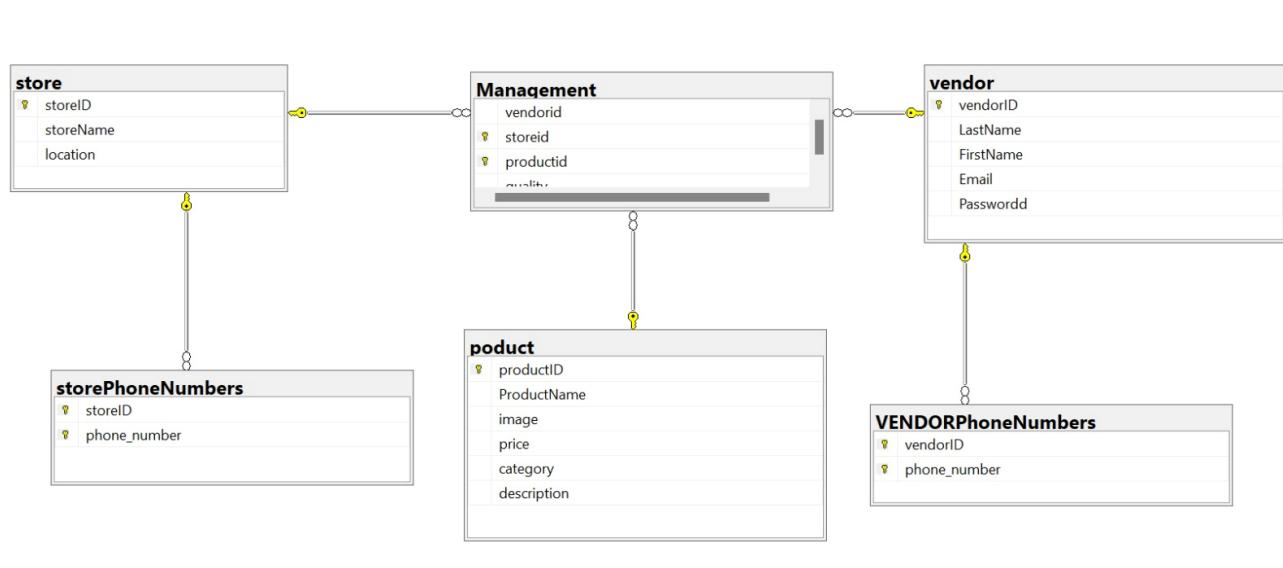
Impact: Changes or updates to operating systems may require adjustments to the app.

4.3 Design of database ERD

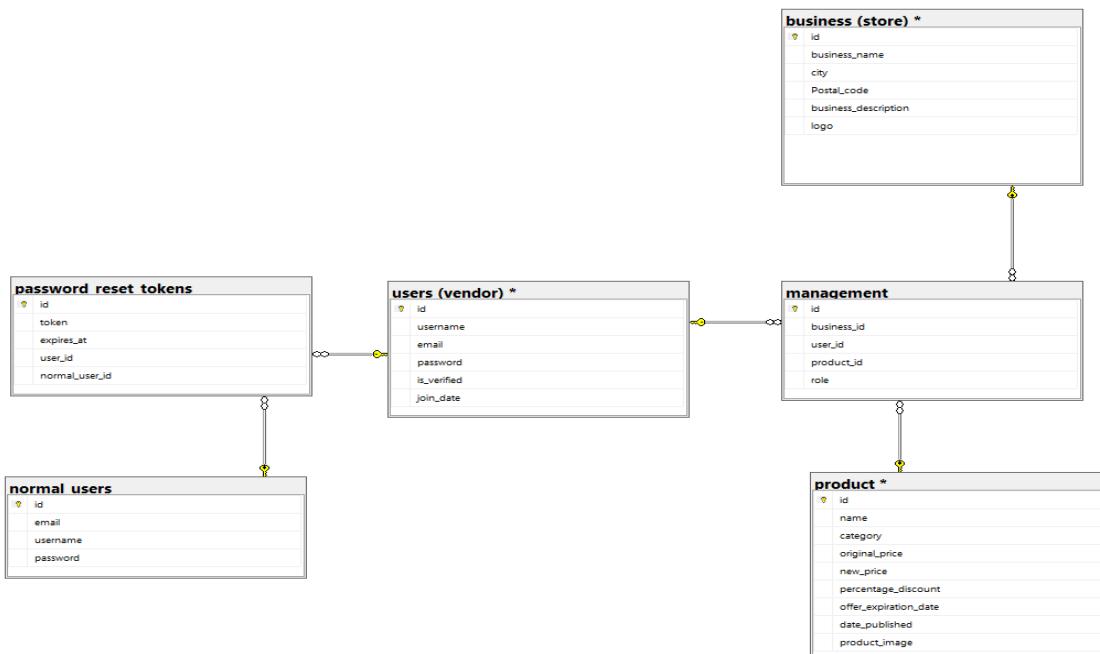
4.3.1 Entity Relationship Diagram



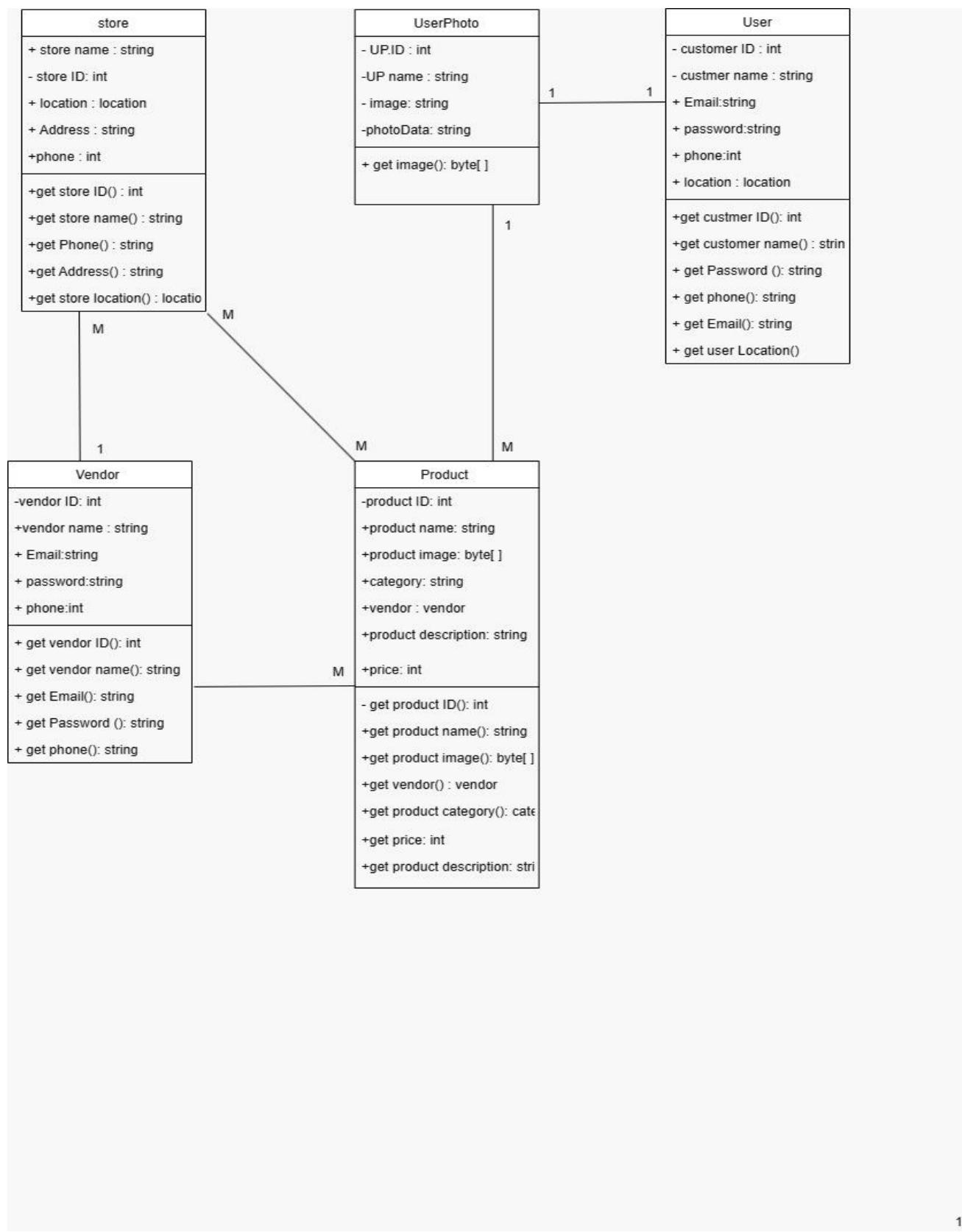
4.3.2 Mapping of Entity Relationship Diagram(planned)



Actual Database:



4.4 Class diagram (planned)



4.5 Use case & Sequence & Activity Diagrams

1 Use-case Scenarios FOR SIGNUP FUNCTION FOR USER AND VENDOR (FR1.1)

| | |
|--------------------|--|
| Use-case Name | signup |
| Use-case ID | UC001 |
| Description | This use case describes the process of a user signing up for a new account. |
| Primary actor | user |
| Secondary actor | none |
| Precondition | The user does not have an existing account. |
| Main Scenario | <ol style="list-style-type: none"> 1. User opens the application. 2. User navigates to the signup page. 3. User enters their desired username, email address, and password. 4. User submits the signup form. 5. The system validates the entered information, ensuring that all required fields are filled and the username and email address are unique. 6. If the entered information is valid and unique, the system creates a new account for the user. 7. The system sends a confirmation message or email to the user to verify their account. 8. User receives the confirmation message or email and follows the provided instructions to verify their account. 9. Once the account is verified, the user can proceed to log in using their credentials. |
| Alternate Scenario | If the entered information is invalid or already exists, the system displays an error message and prompts the user to enter valid and unique information. |
| Post condition | The user successfully creates a new account and can proceed with logging in. |

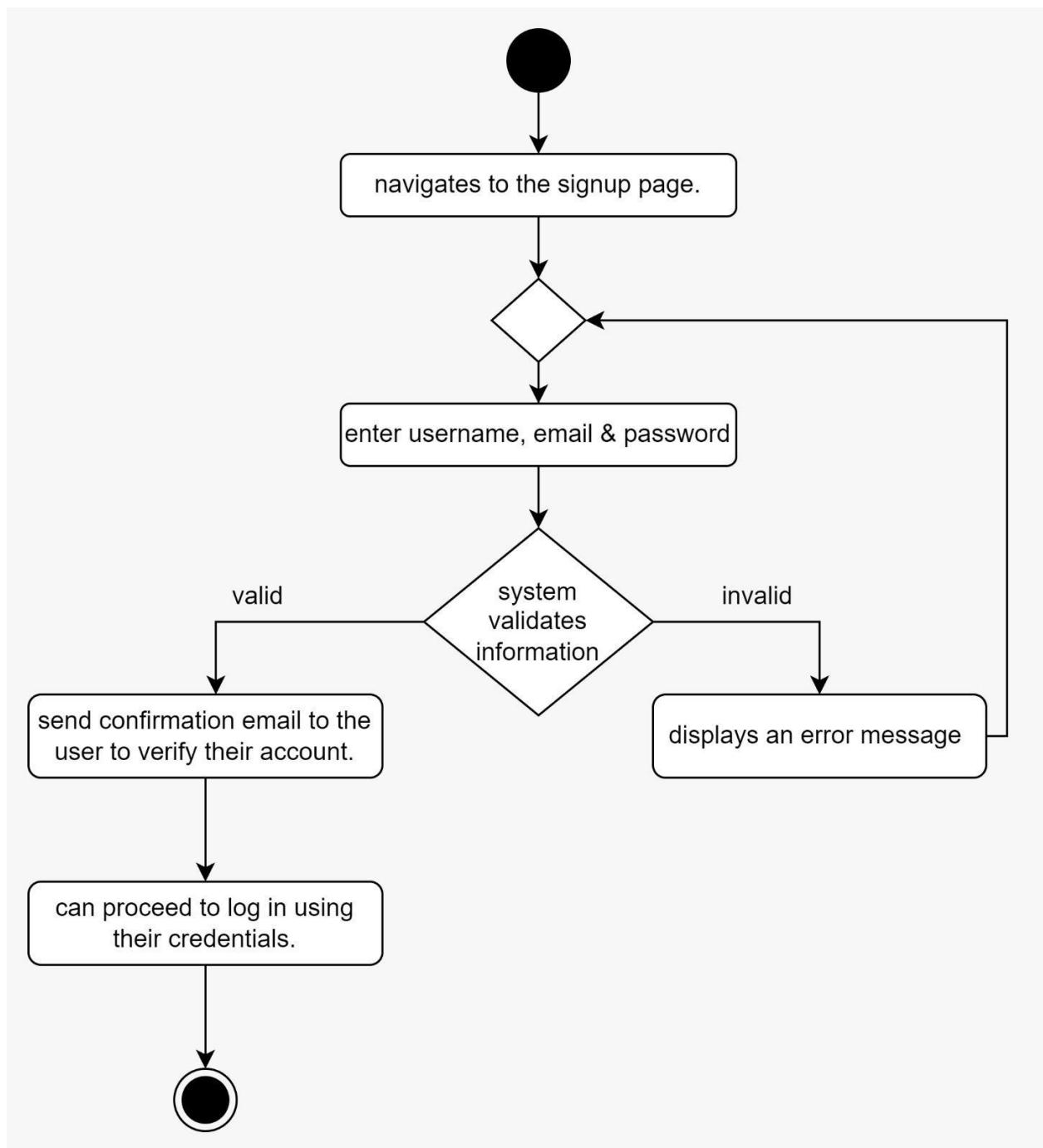


Figure 1.1 ACTIVITY DIAGRAM FOR SIGNUP

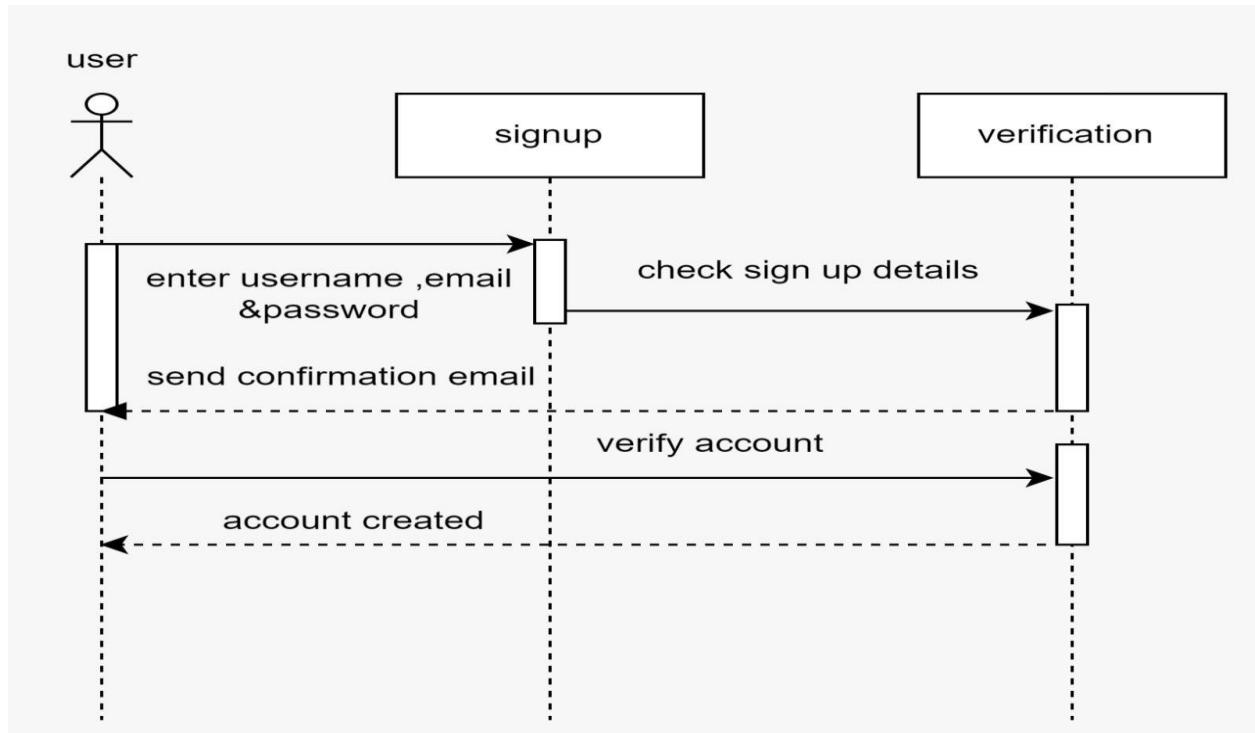


Figure1. 2 SEQUENCE DIAGRAM FOR SIGNUP

**2 Use-case Scenarios FOR LOGIN FUNCTION FOR USER AND VENDOR
(FR1.1)**

| | |
|--------------------|--|
| Use-case Name | login |
| Use-case ID | UC002 |
| Description | This use case describes the process of a user logging into the system. |
| Primary actor | user |
| Secondary actor | none |
| Precondition | The user must have a registered account |
| Main Scenario | <ol style="list-style-type: none"> 1. User opens the application. 2. User navigates to the login page. 3. User enters their username or email and password. 4. User submits the login form. 5. The system verifies the user's credentials by comparing them to the stored user information. 6. If the entered credentials are valid, the system grants access to the user. 7. User is redirected to the home page. 8. If the entered credentials are invalid, the system displays an error message indicating incorrect username or password. <p>User can retry entering the correct credentials or choose the "Forgot Password" option to reset their password.</p> |
| Alternate Scenario | <ol style="list-style-type: none"> 1. The system presents the user with a password recovery form or prompts them to enter their registered email address. 2. User enters their email address or provides the required information in the password recovery form. 3. The system verifies the provided information and sends a password reset link or instructions to the user's registered email address. 4. User receives the password reset link or instructions. 5. User follows the provided instructions to reset their password. <p>Once the password is successfully reset, the user can log in using their new password.</p> |

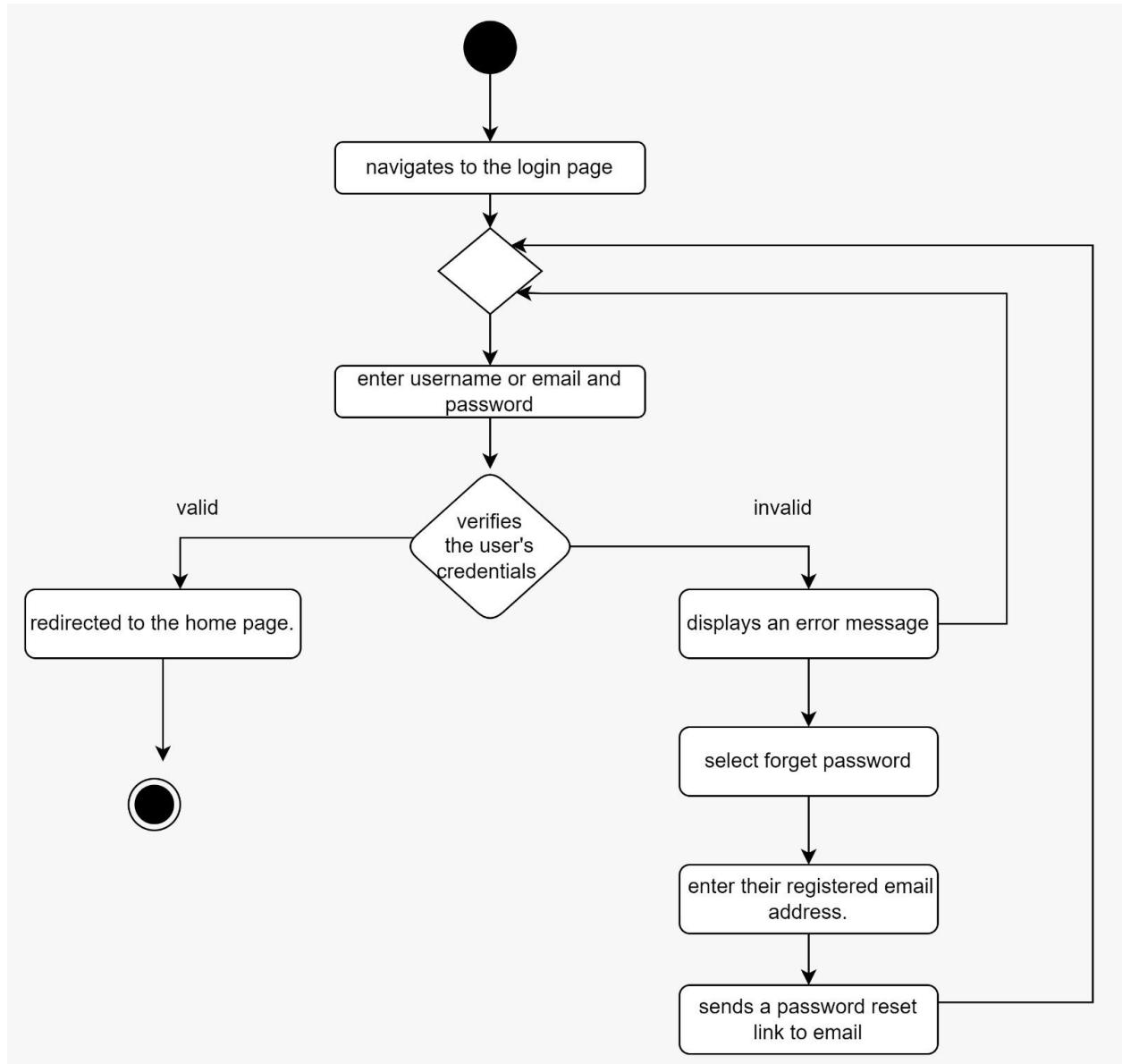


Figure 2.1 ACTIVITY DIAGRAM FOR LOGIN

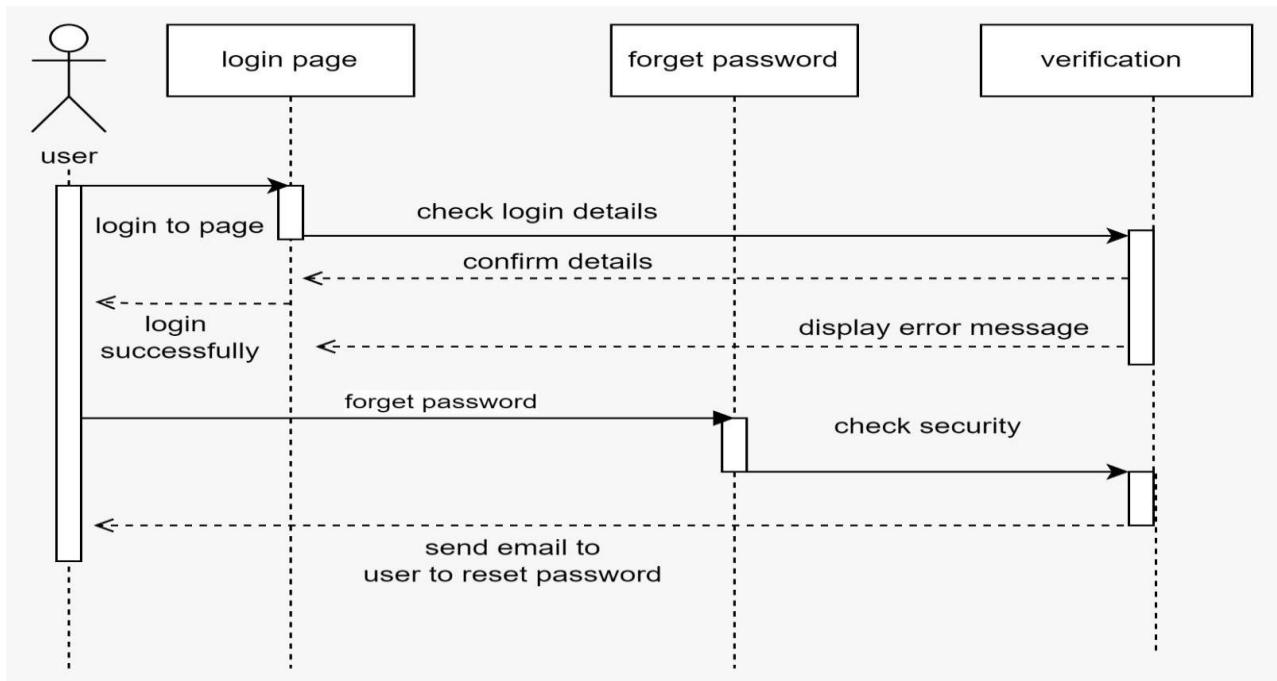


Figure 2.2 SEQUENCE DIAGRAM FOR LOGIN

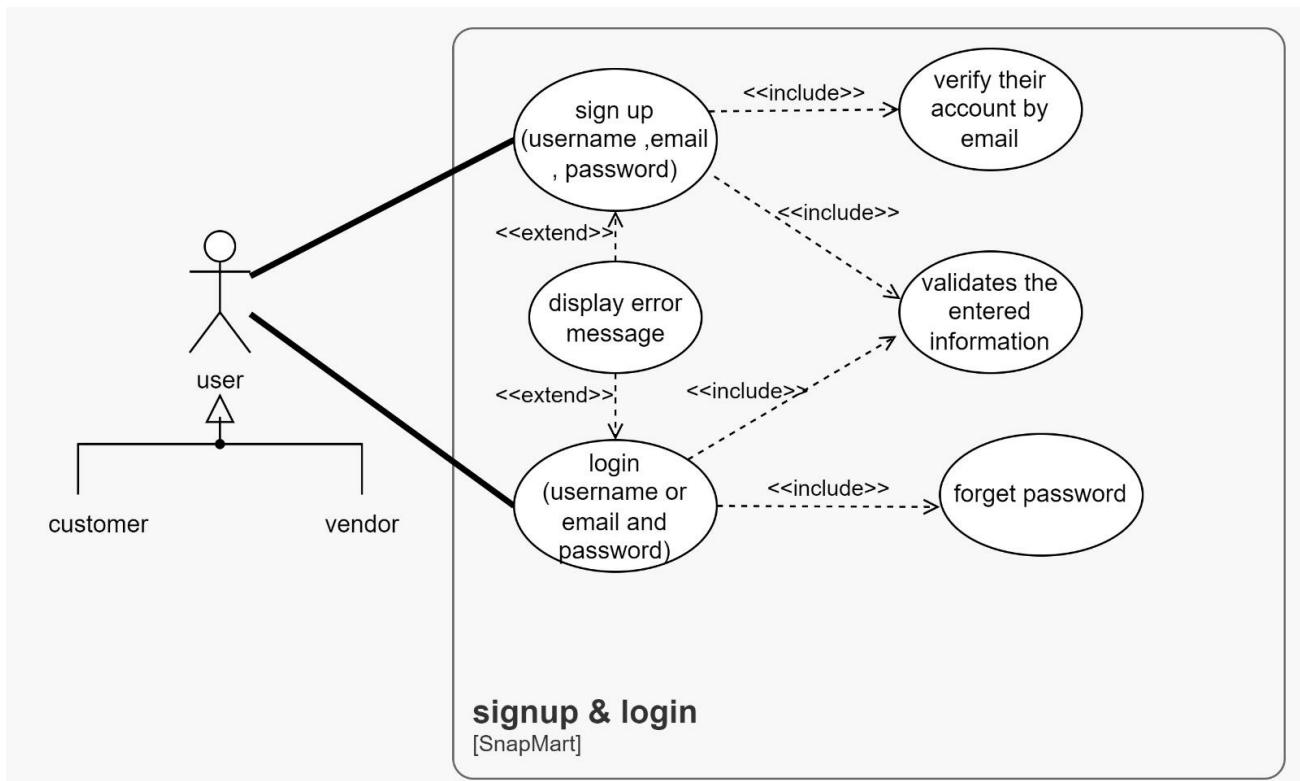


Figure 3 USECASE DIAGRAM FOR LOGIN & SIGNUP

3 Use-case Scenarios FOR IMAGE CAPTURE FUNCTION (FR2.1)

| | |
|--------------------|--|
| Use-case Name | Image capture |
| Use-case ID | UC003 |
| Description | The use-case describes the functionality of the system to have an image capture feature that allows users to take photos within the application |
| Primary actor | User |
| Secondary actor | Device |
| Precondition | The user has opened the application and has access to a device with a camera |
| Main Scenario | <ol style="list-style-type: none"> 1. User selects the image capture feature from the application's interface. 2. The system activates the device's camera functionality. 3. User frames the desired subject within the camera viewfinder. 4. User captures a photo by pressing the capture button. 5. The system processes and saves the captured image to the designated location within the application. 6. User selects the upload photo feature from the application's interface. 7. The system prompts the user to select the destination for uploading the photo. 8. User selects the desired destination for uploading the photo. 9. The system uploads the processed image to the specified destination. |
| Alternate Scenario | If user cancel image capture the user is returned to the application interface. |

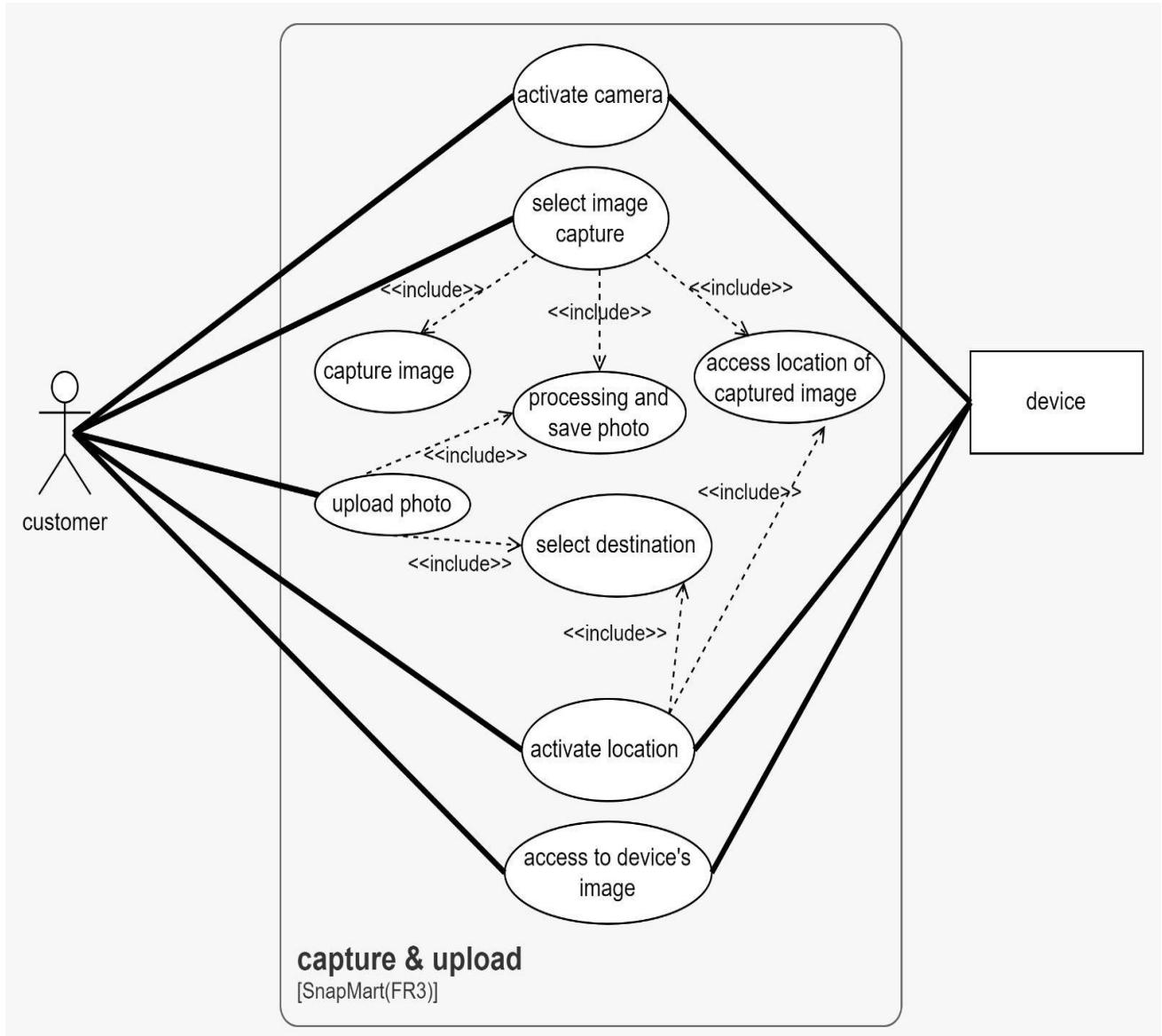


Figure 3.1 USE CASE DIAGRAM FOR CAPTURE IMAGE

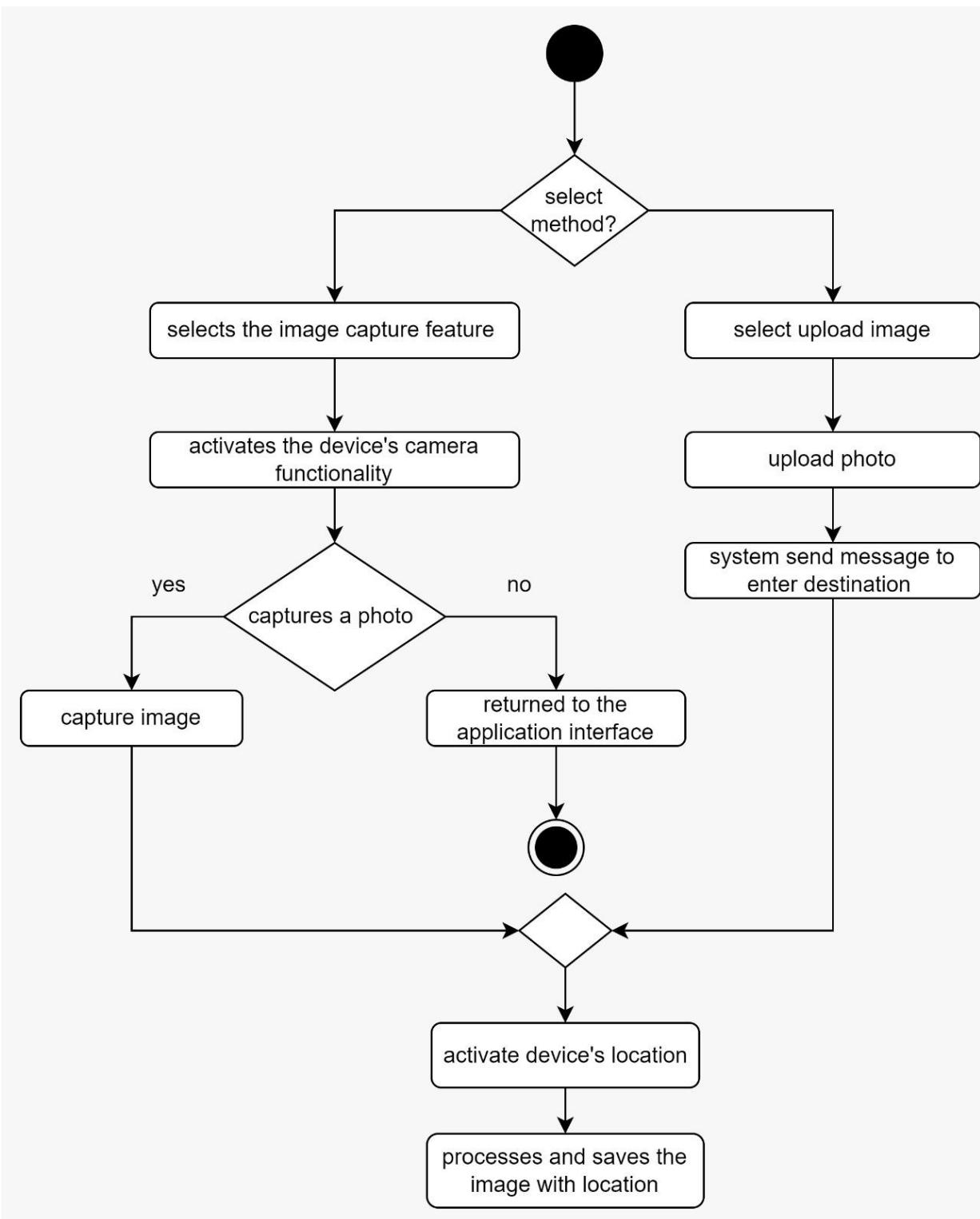


Figure 3.2 ACTIVITY DIAGRAM FOR CAPTURE IMAGE FUNCTION

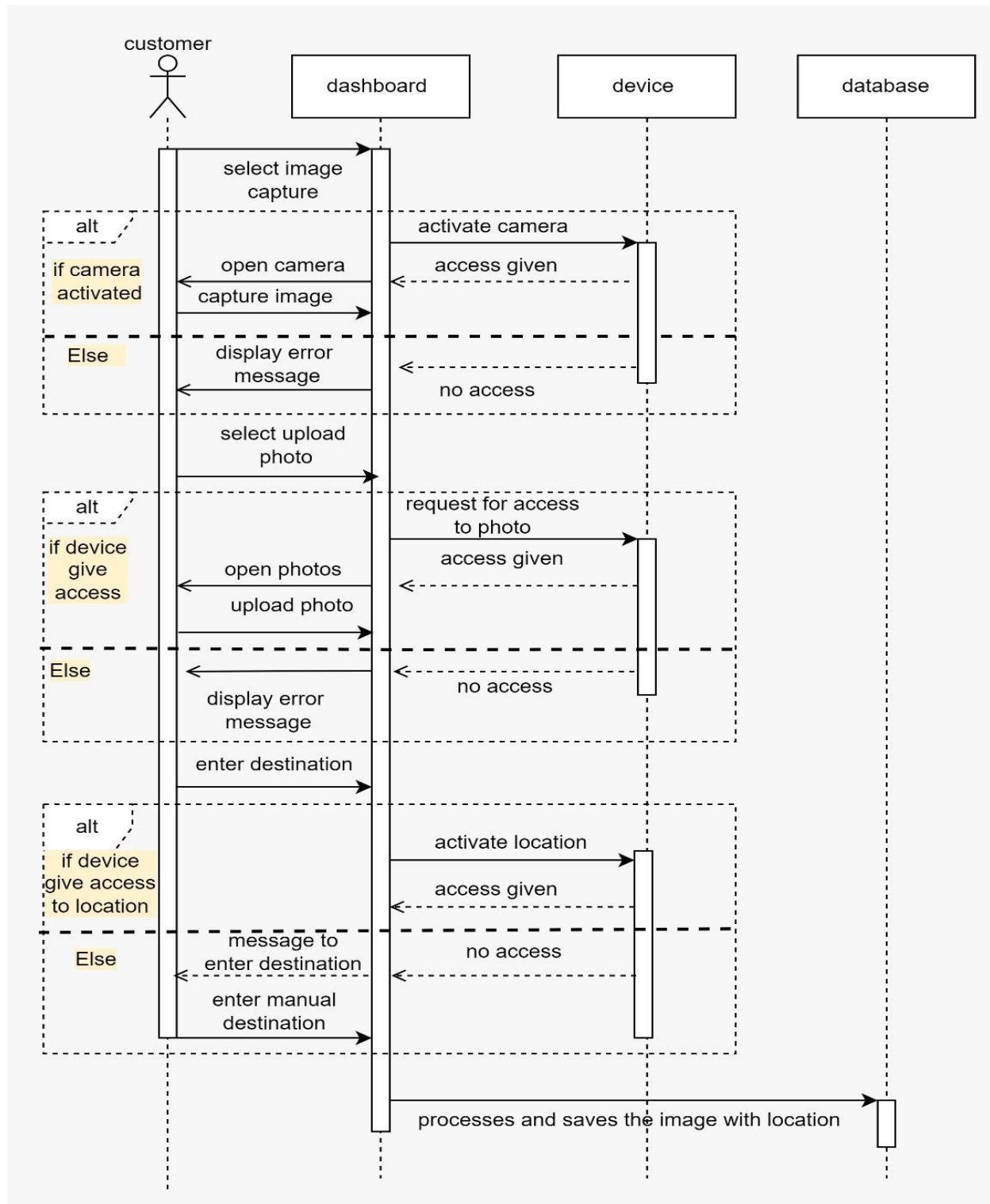


Figure 3.3 SEQUENCE DIAGRAM FOR CAPTURE IMAGE FUNCTION

4 use case scenario FOR IMAGE RECOGNITION FUNCTION (FR2.2)

| | |
|--------------------|--|
| Use-case Name | Image Recognition |
| Use-case ID | UC004 |
| Description | The use-case describes the functionality of the system to utilize an image recognition algorithm to analyze and identify objects in the uploaded images |
| Primary actor | System |
| Secondary actor | None |
| Precondition | The user has uploaded an image within the application |
| Main Scenario | <ol style="list-style-type: none"> 1. Upon user interaction, the system receives the uploaded or captured image saved in database. 2. The system performs preprocessing tasks on the uploaded image, such as resizing, normalization, or enhancement, to prepare it for the image recognition algorithm. 3. The system applies image recognition algorithm to analyze the content of the uploaded image. 4. The algorithm identifies and extracts features, patterns, and objects within the image. 5. The system searches the database for images that contain similar objects to those identified in the uploaded image. 6. The search may involve comparing object features, patterns, or other relevant characteristics stored in the database. 7. Based on the results of the database search, the system generates a list of images that share similar objects with the one captured in the uploaded image. 8. The list includes relevant data such as image names, location, and other details. 9. The system presents the generated list of images to the user through the application interface. <p>The user can view the list, explore details of each matched image.</p> |
| Alternate Scenario | If object not found displays an error message |

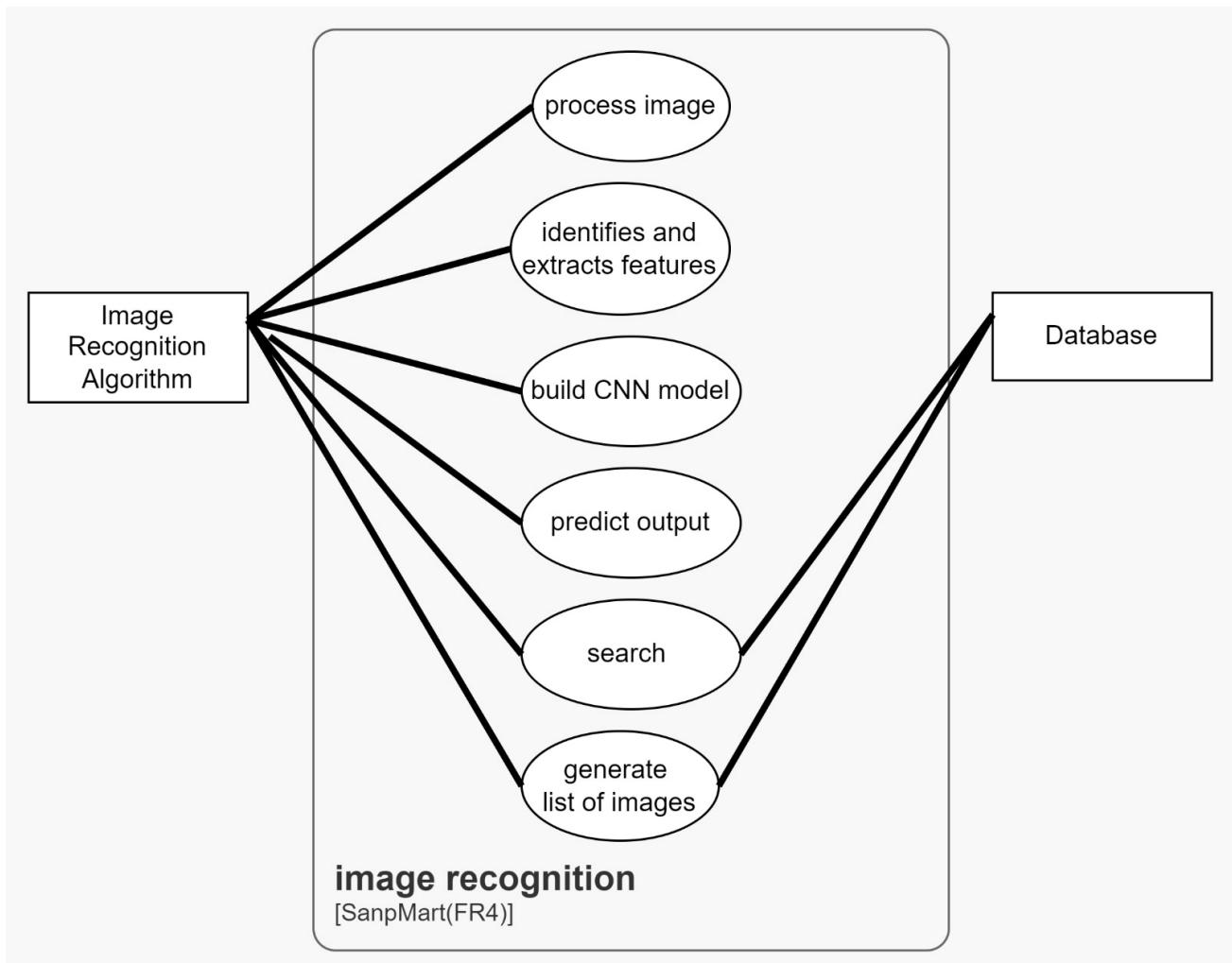


Figure 4.1 USE CASE DIADRAM FOR IMAGE RECOGNITION

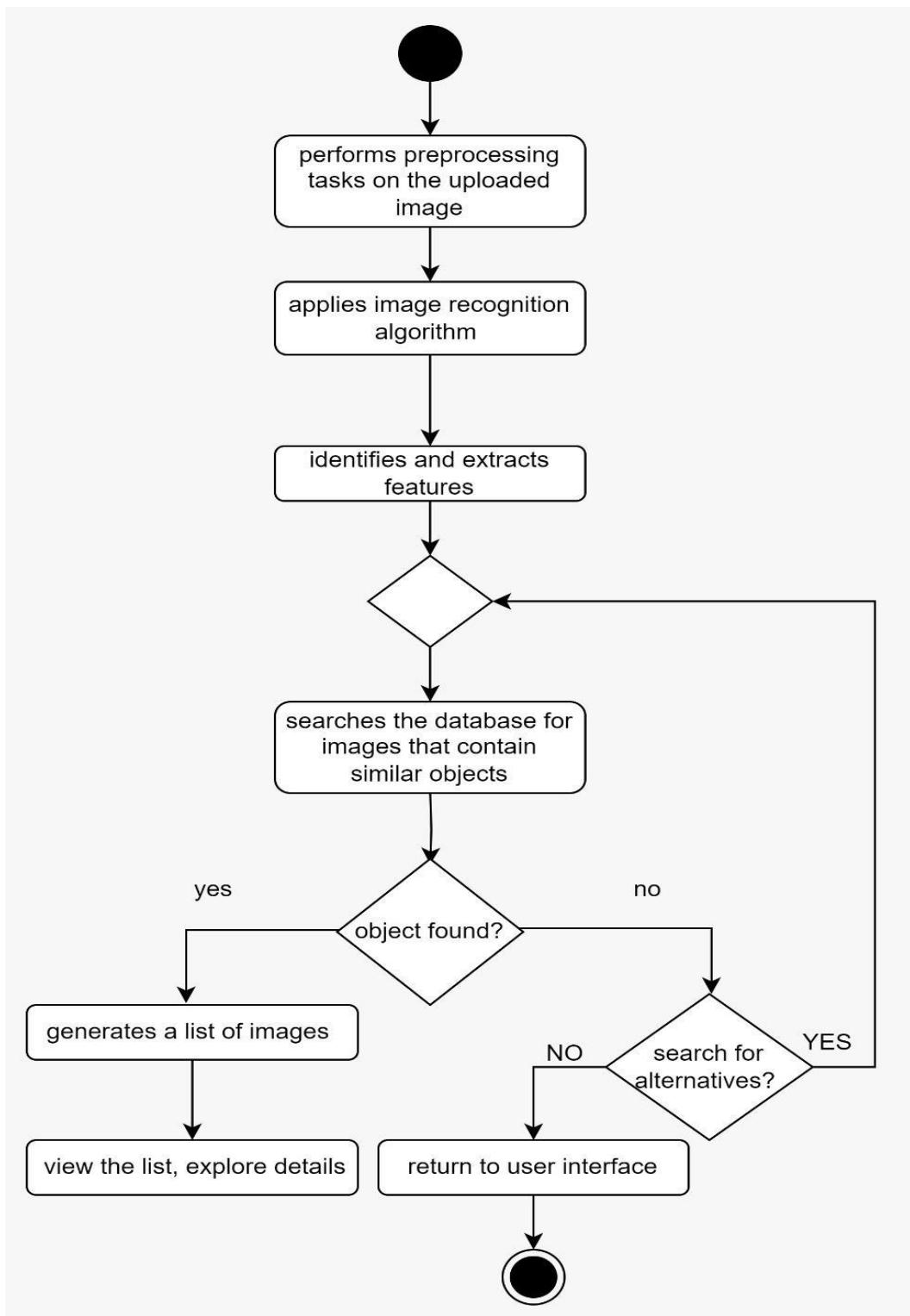


Figure 4.2 ACTIVITY DIAGRAM FOR IMAGE RECOGNITION

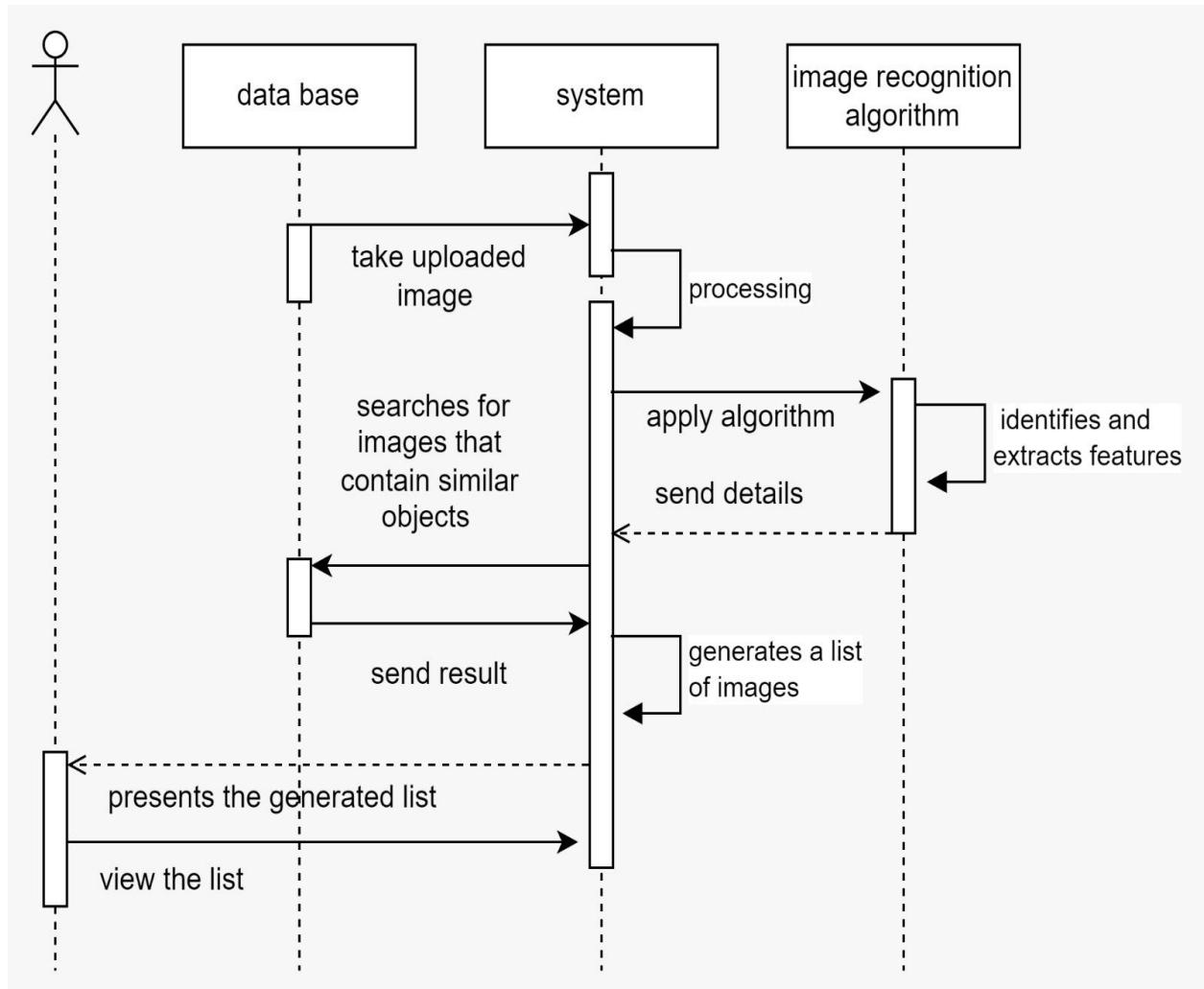


Figure 4.3 SEQUENCE DIAGRAM FOR IMAGE RECOGNITION

5 use case scenario FOR MAPPING INTEGRATION FUNCTION (FR2.3)

| | |
|--------------------|--|
| Use-case Name | Mapping integration |
| Use-case ID | UC005 |
| Description | The use-case describes the functionality of the system to integrate with a mapping service or API to retrieve distance and navigation information between the user's location and the nearest stores. |
| Primary actor | system |
| Secondary actor | none |
| Precondition | The system is connected to the mapping service/API |
| Main Scenario | <ol style="list-style-type: none"> 1. The system establishes a connection with the mapping service/API. 2. The system retrieves the user's location or prompts the user to provide their location. 3. The system queries the mapping service/API to retrieve the nearest store locations based on the user's location. 4. The system receives the distance and navigation information from the mapping service/API. 5. The system displays the distance and navigation information to the user, providing directions or relevant details for reaching the nearest stores. |
| Alternate Scenario | None |

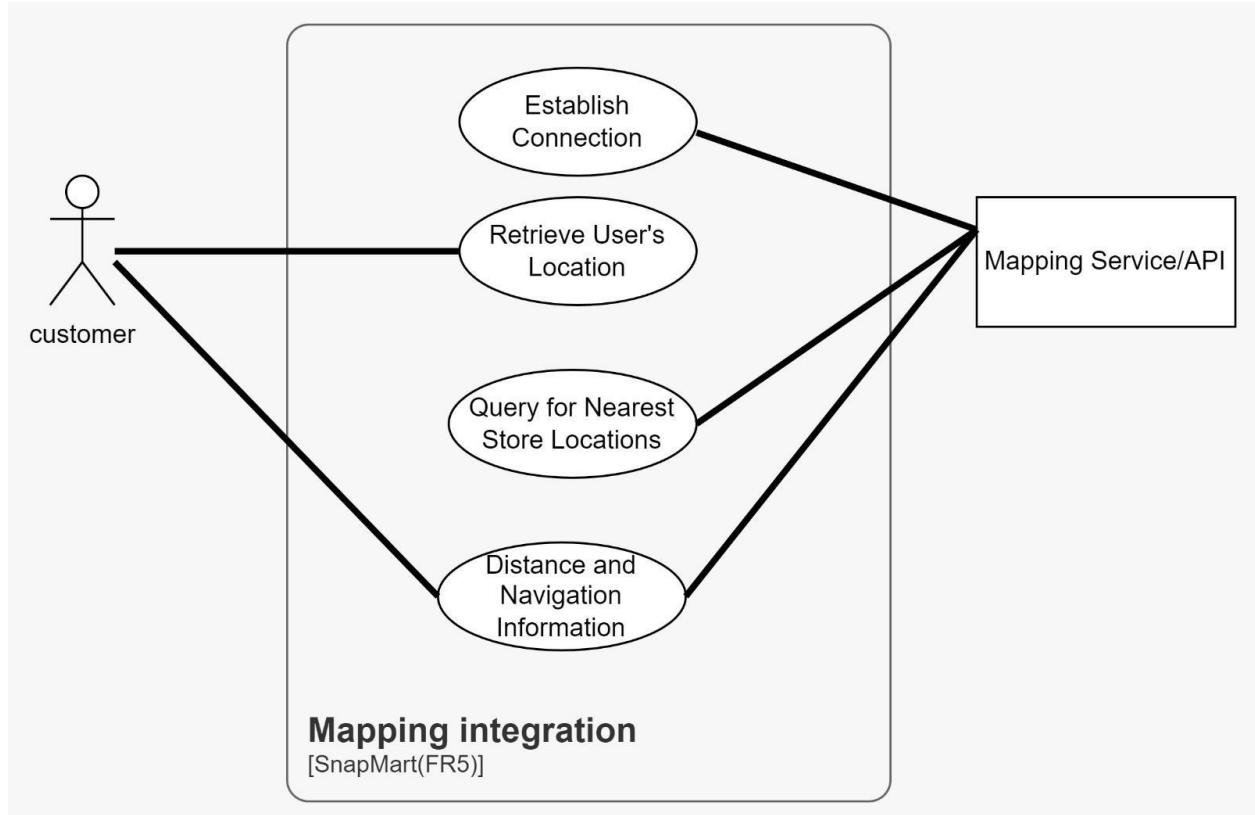


Figure 5.1 USE CASE DIAGRAM FOR MAPPING INTEGRATION

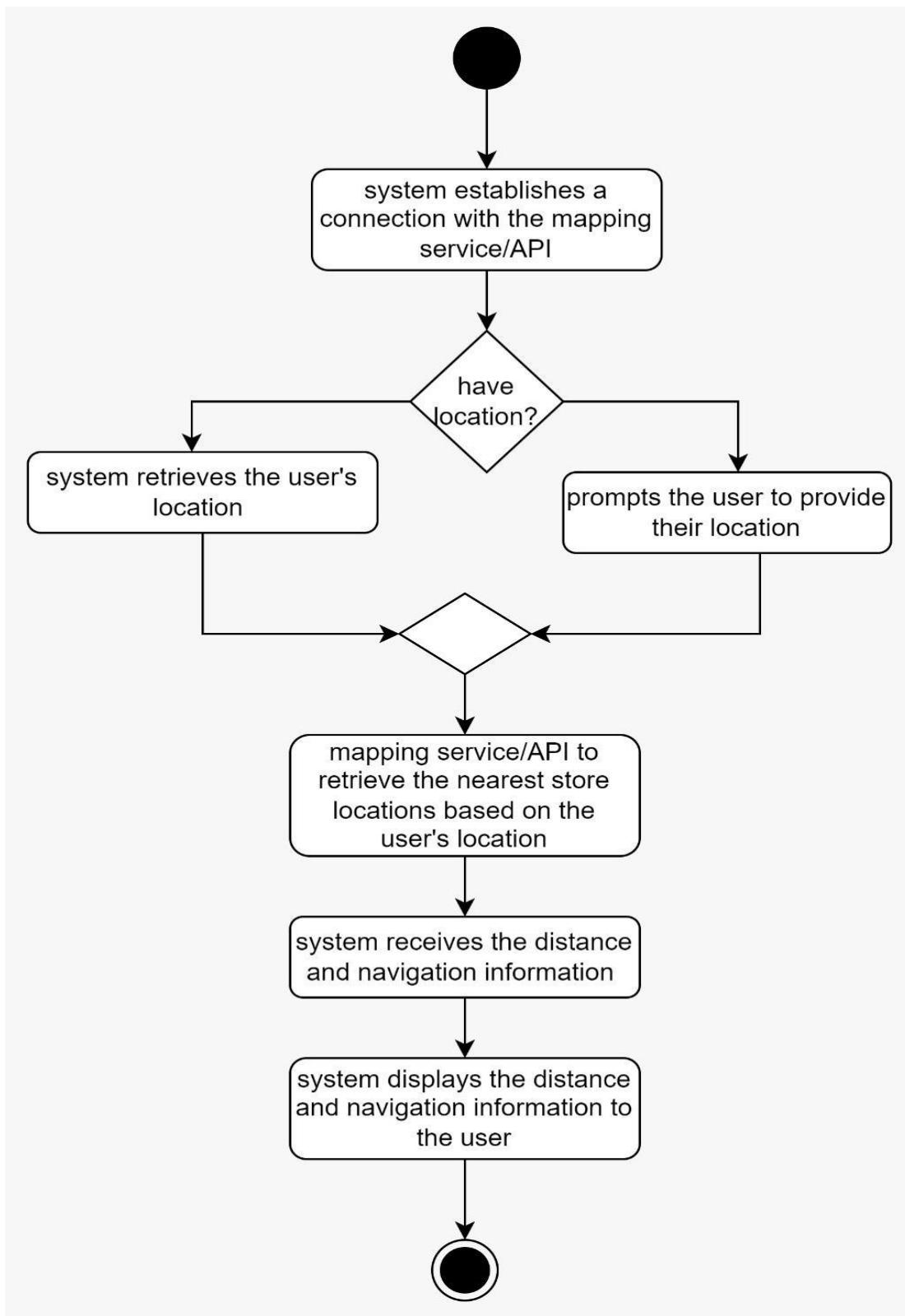


Figure 5.2 ACTIVITY DIAGRAM FOR MAPPING INTEGRATION

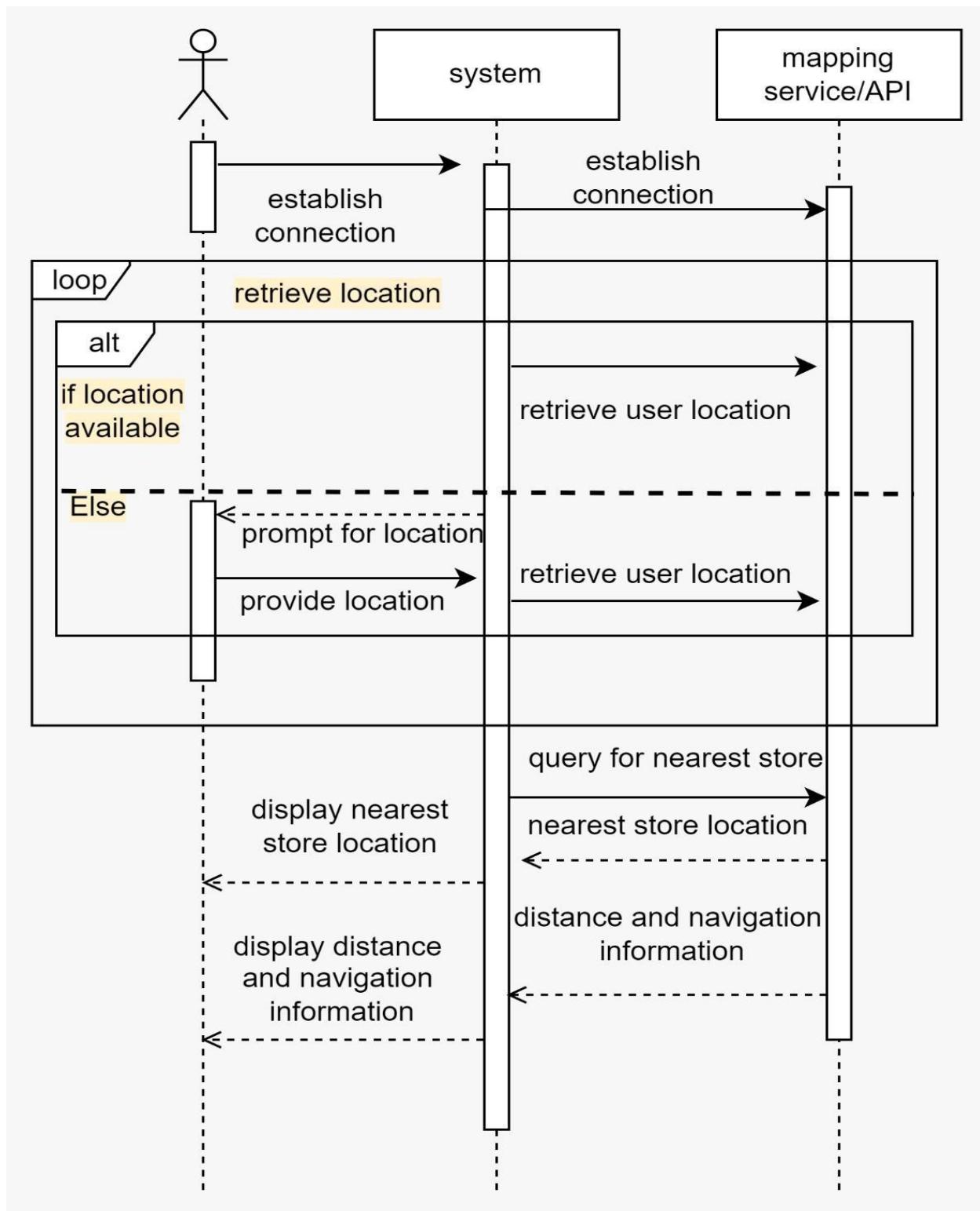


Figure 5.3 SEQUENCE DIAGRAM FOR MAPPING INTEGRATION

6 use case scenario FOR CREATE INVENTORY BY VENDOR (FR 3.3)

| | |
|--------------------|---|
| Use-case Name | Create inventory |
| Use-case ID | UC006 |
| Description | The vendor adds items from scratch to build the inventory that only the vendor can provide this inventory with items. |
| Primary actor | Vendor |
| Secondary actor | None |
| Precondition | <p>1- The vendor must have valid login credentials to access the system.</p> <p>2- The vendor's account must have the necessary permission to add items to the inventory.</p> <p>The inventory management system must be operational and accessible.</p> |
| Main Scenario | <p>1- The vendor logs into the system using their credentials.</p> <p>2- Once authenticated, the vendor accesses the inventory management section.</p> <p>3- The vendor selects the option to "Add New Item" to initiate the process.</p> <p>4- The system prompts the vendor to input details such as item name, description, quantity, price, and any relevant specifications.</p> <p>5- The vendor submits the information, indicating the completion of the item addition process.</p> <p>6- The system validates the entered data and adds the new item to the inventory database.</p> <p>The inventory database is now updated to include the newly added item, and it is made available for customers to view and purchase</p> |
| Alternate Scenario | If the vendor provides incomplete or invalid information during the item addition process, the system displays an error message. The vendor is prompted to correct the errors or provide the missing information. After making the necessary corrections, the vendor resubmits the item details. The system revalidates the information and, upon successful validation, adds the item to the inventory. |

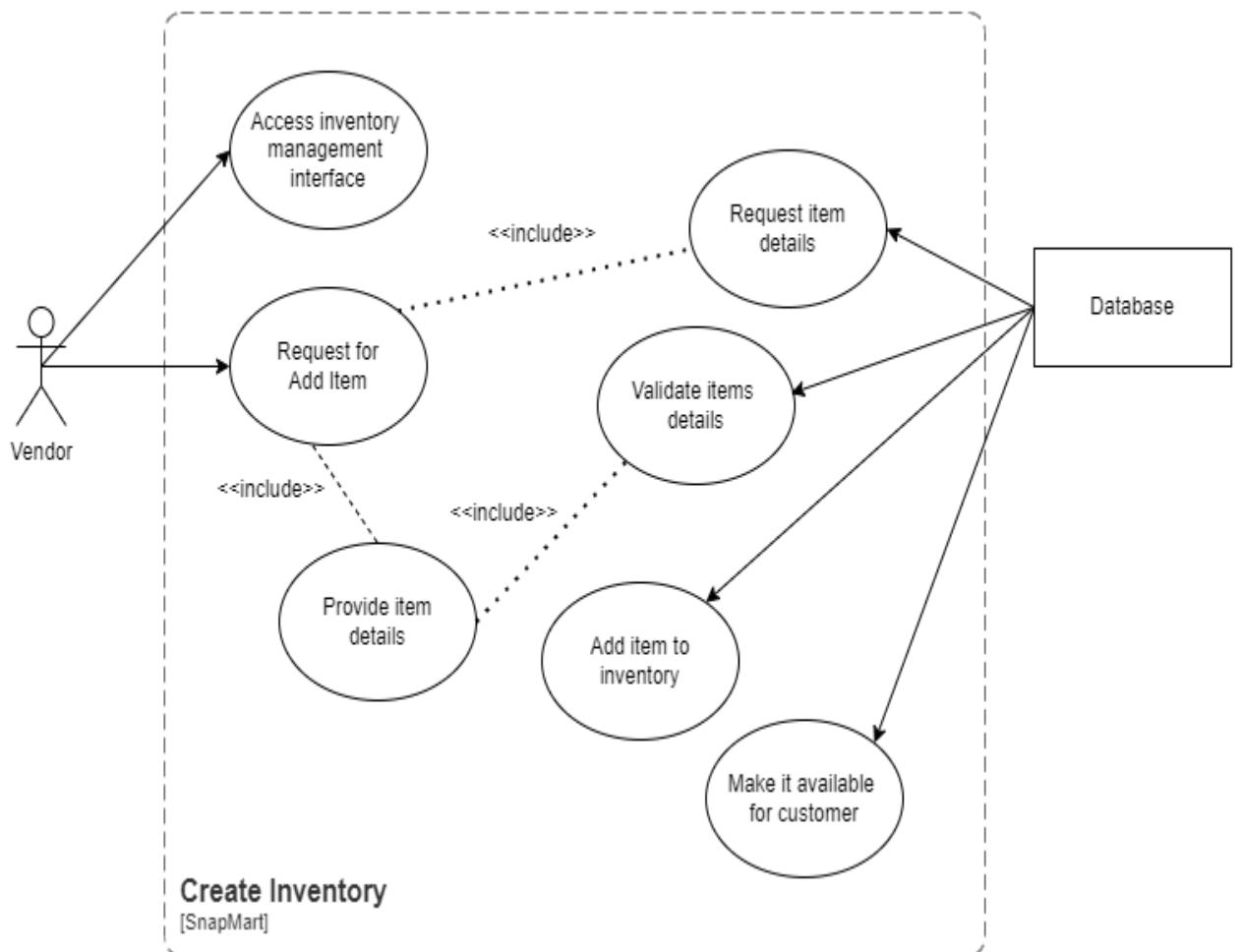


Figure 6.1 USE CASE DIAGRAM FOR CREATE INVENTORY

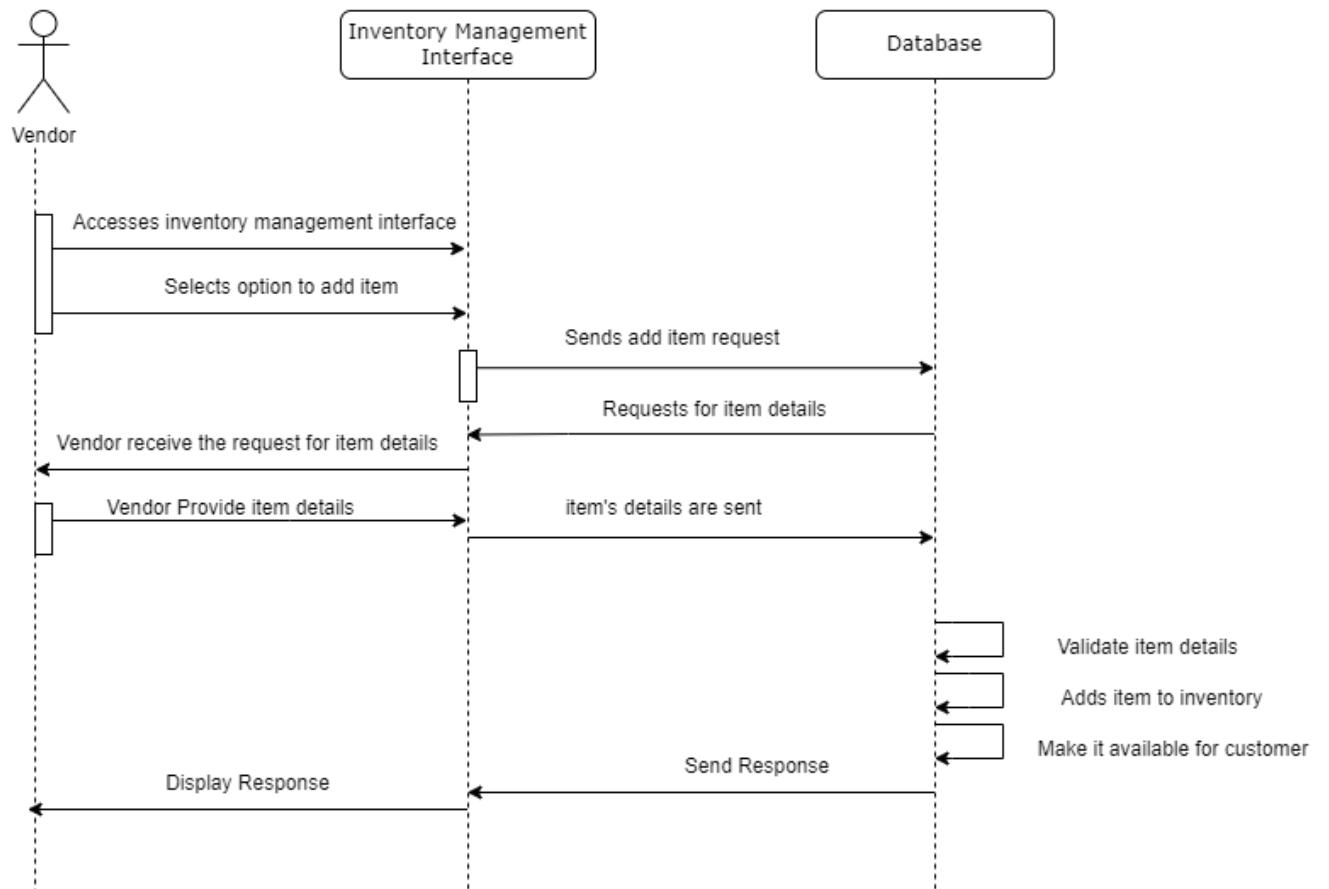


Figure 6.2 SEQUENCE DIAGRAM FOR CREATE INVENTORY

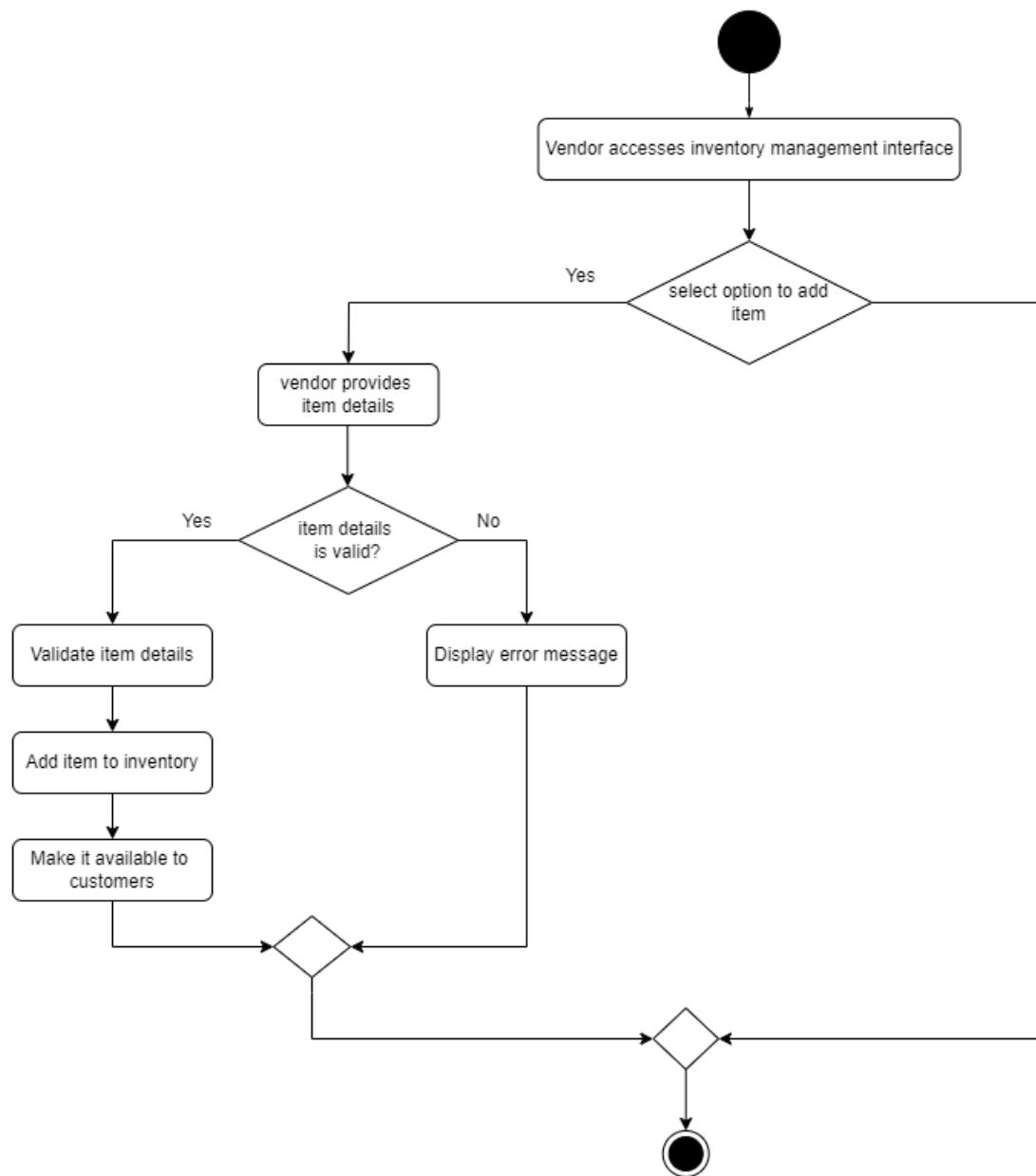


Figure 6.3 ACTIVITY DIADRAM FOR CREATE INVENTORY

7 use case scenario FOR ADDING NEW ITEM (FR3.2)

| | |
|--------------------|--|
| Use-case Name | Adding New Items |
| Use-case ID | UC007 |
| Description | The system should provide an interface for vendors to add new items to their inventory by entering information such as item name, description, price, and quantity. And should be able to update existing item information, including price, description, and quantity. |
| Primary actor | Vendor |
| Secondary actor | None |
| Precondition | <ul style="list-style-type: none"> 1- The system is up and running. 2- Vendors have valid accounts and authentication credentials. 3- The vendor has the necessary permissions to add or update items in the inventory. |
| Main Scenario | <ul style="list-style-type: none"> 1- The vendor accesses the system's inventory management interface. 2- The vendor selects the option to add a new item and provides information such as item name, description, price, and quantity. 3- The vendor selects the option to update an existing item and modifies information such as price, description, or quantity. 4- The vendor confirms the changes made. |
| Alternate Scenario | <ul style="list-style-type: none"> 1- The vendor provides invalid or incomplete information while adding or updating an item. 2- The vendor attempts to add or update an item without the necessary permissions. |
| Post condition | <ul style="list-style-type: none"> 1- New items are successfully added to the inventory. 2- Existing items are updated with the modified information. |

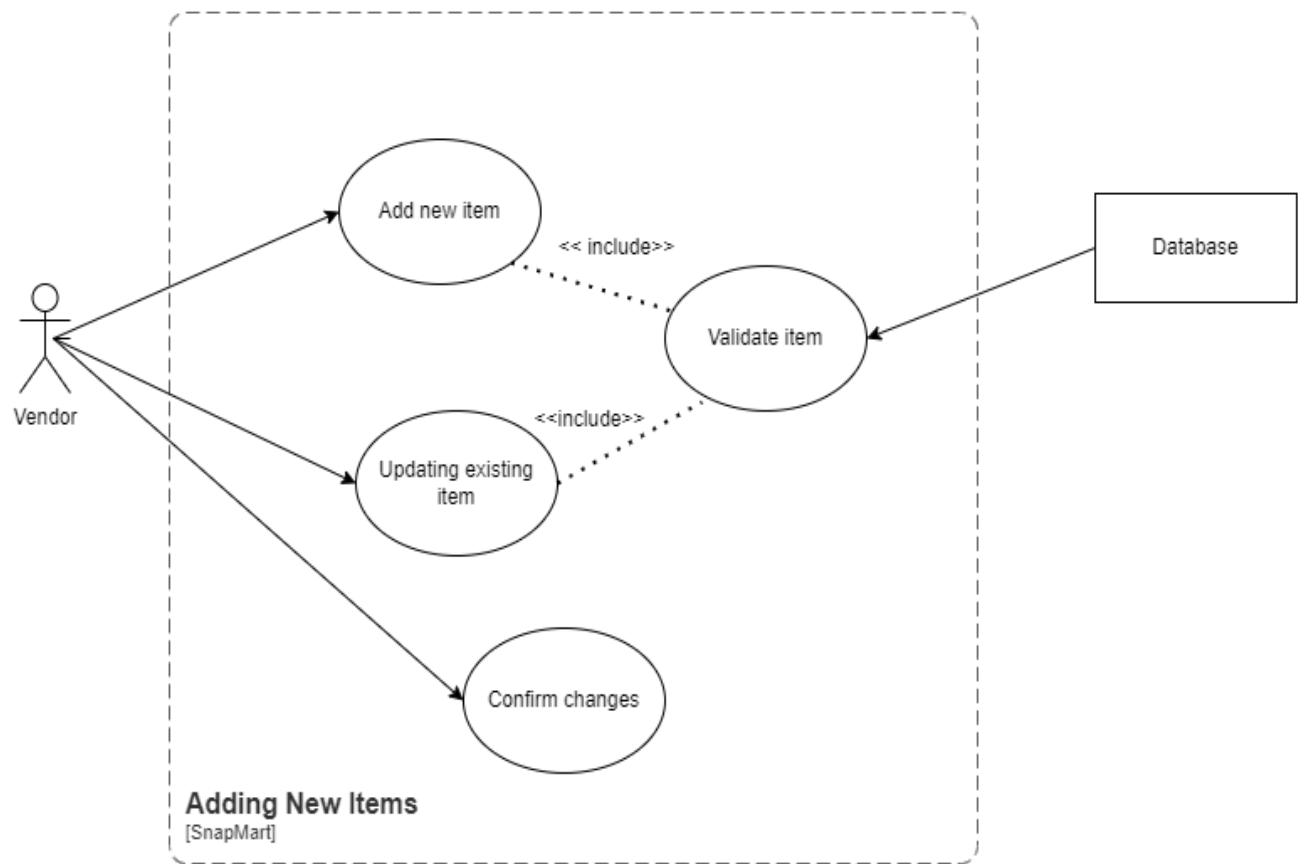


Figure 7.1 USE CASE DIAGRAM FOR ADDING NEW ITEM

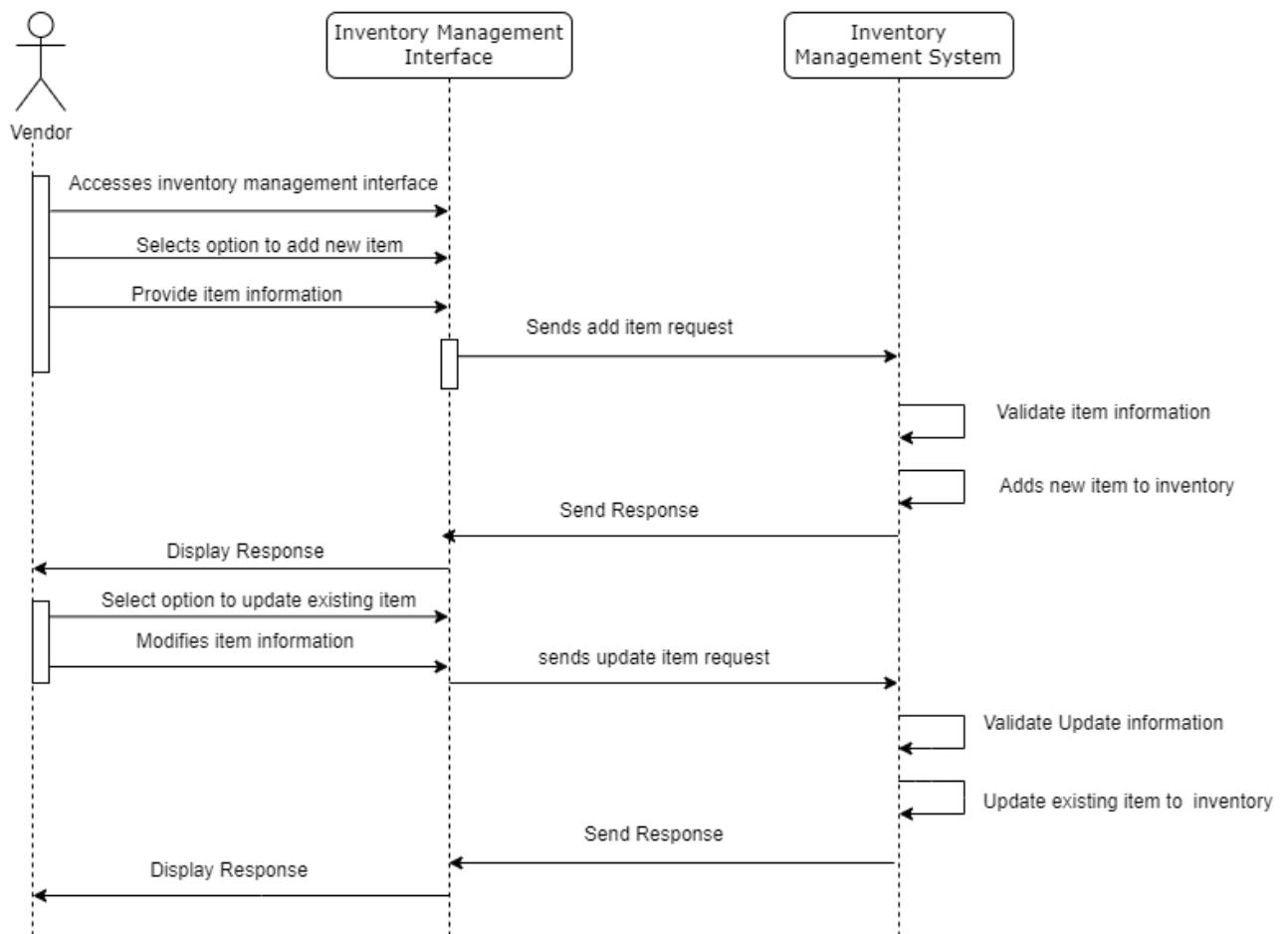


Figure 7.2 SEQUENCE DIAGRAM FOR ADDING NEW ITEM

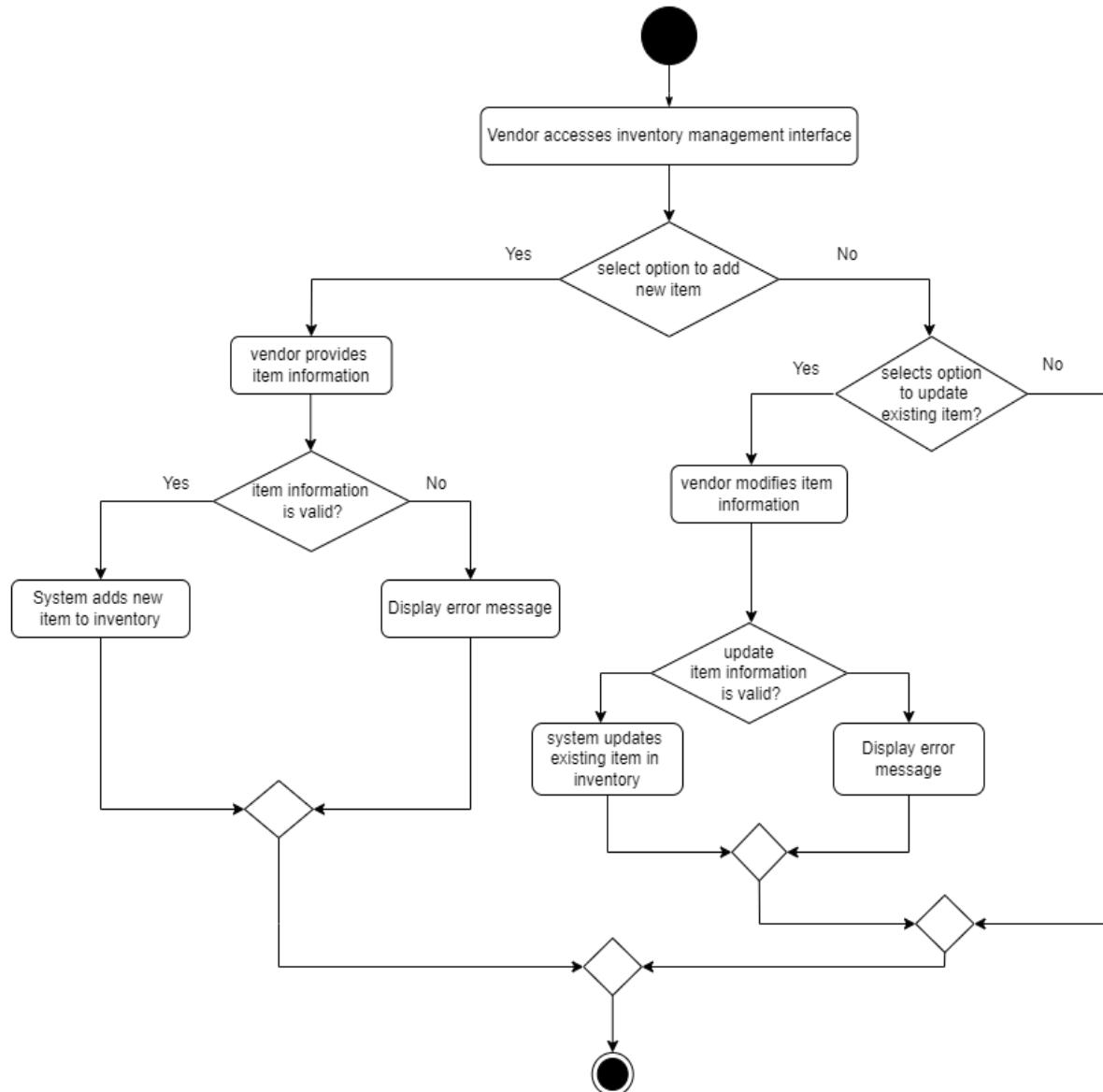


Figure 7.3 ACTIVITY DIADRAM FOR ADDING NEW ITEM

8 use case scenario FOR CATEGORIZING FEATURES (FR3.4)

| | |
|--------------------|--|
| Use-case Name | Categorizing Features |
| Use-case ID | UC007 |
| Description | The system should provide a categorization feature where vendors can assign items to different categories or tags. And should be able to create and manage their own categories or tags for organizing their inventory |
| Primary actor | Vendor |
| Secondary actor | None |
| Precondition | <p>1- The system is operational and accessible.</p> <p>2- The user has a valid account and is logged in.</p> <p>The user has items in their inventory that need categorization.</p> |
| Main Scenario | <p>1- The vendor logs into the system using their credentials.</p> <p>2- Upon login, the vendor is directed to a dashboard where they can view an overview of their inventory.</p> <p>3- The vendor navigates to the categorization feature within the system.</p> <p>4- The vendor selects an option to create a new category or tag. They enter the name, description, and any relevant details for the new category or tag.</p> <p>5- After entering the information, the vendor saves the new category or tag, and it is immediately applied to the system.</p> <p>6- The vendor goes through their inventory list and selects items that belong to the newly created category or tag. They can assign multiple items to the same category/tag.</p> <p>The vendor can view, edit, or delete existing categories or tags. They can also reassign items to different categories or tags as needed.</p> |
| Alternate Scenario | <p>1- Duplicate Category or Tag Name so the system prevents the creation and prompts the user to choose a unique name.</p> <p>2- Invalid User Access and the system will deny access and prompt the user to log in with valid credentials.</p> <p>Empty Category or Tag Assignment so the system prompts the user to select at least one category or tag before proceeding.</p> |

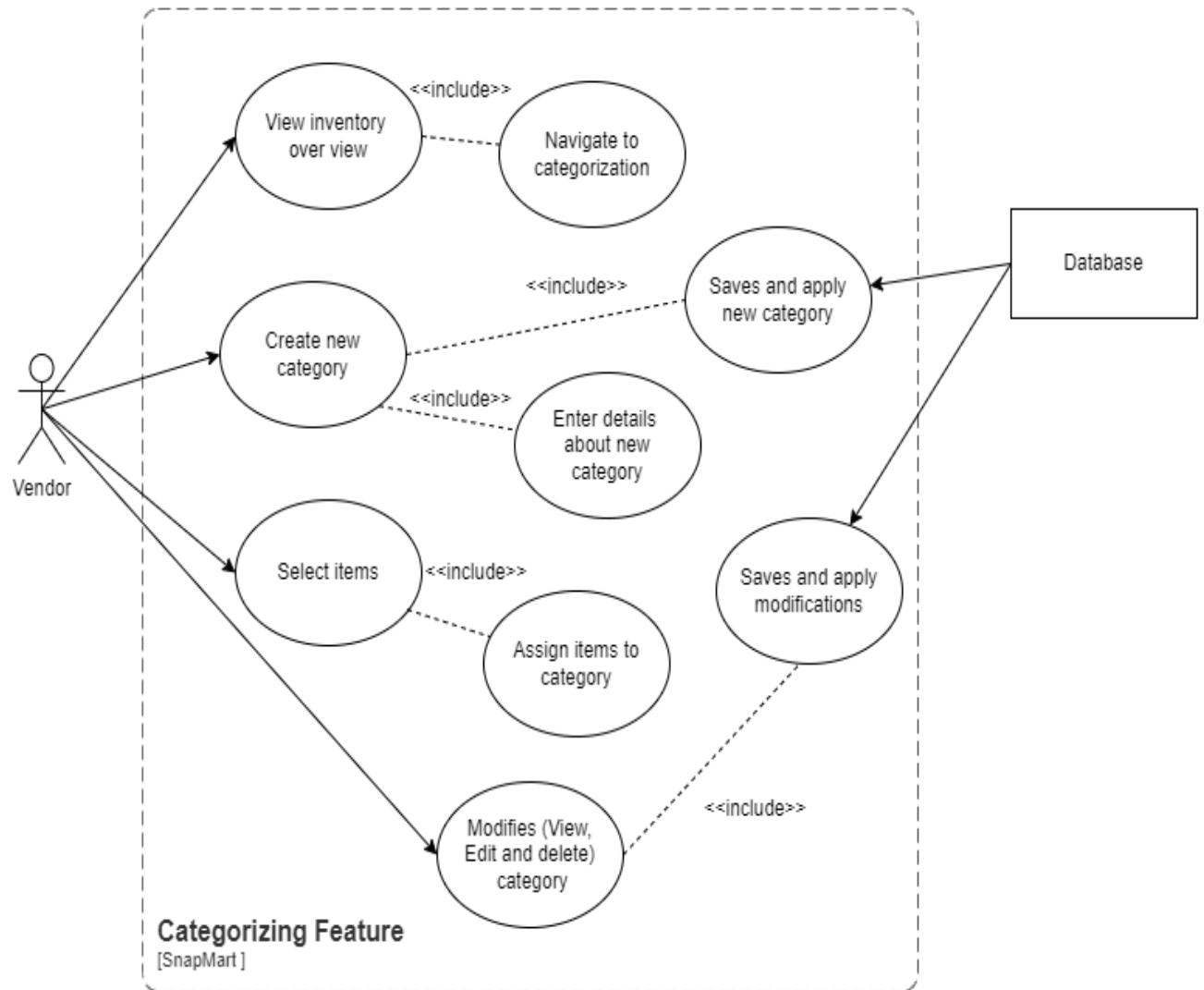


Figure 8.1 USE CASE FOR CATEGORIZING ITEMS FOR VENDOR

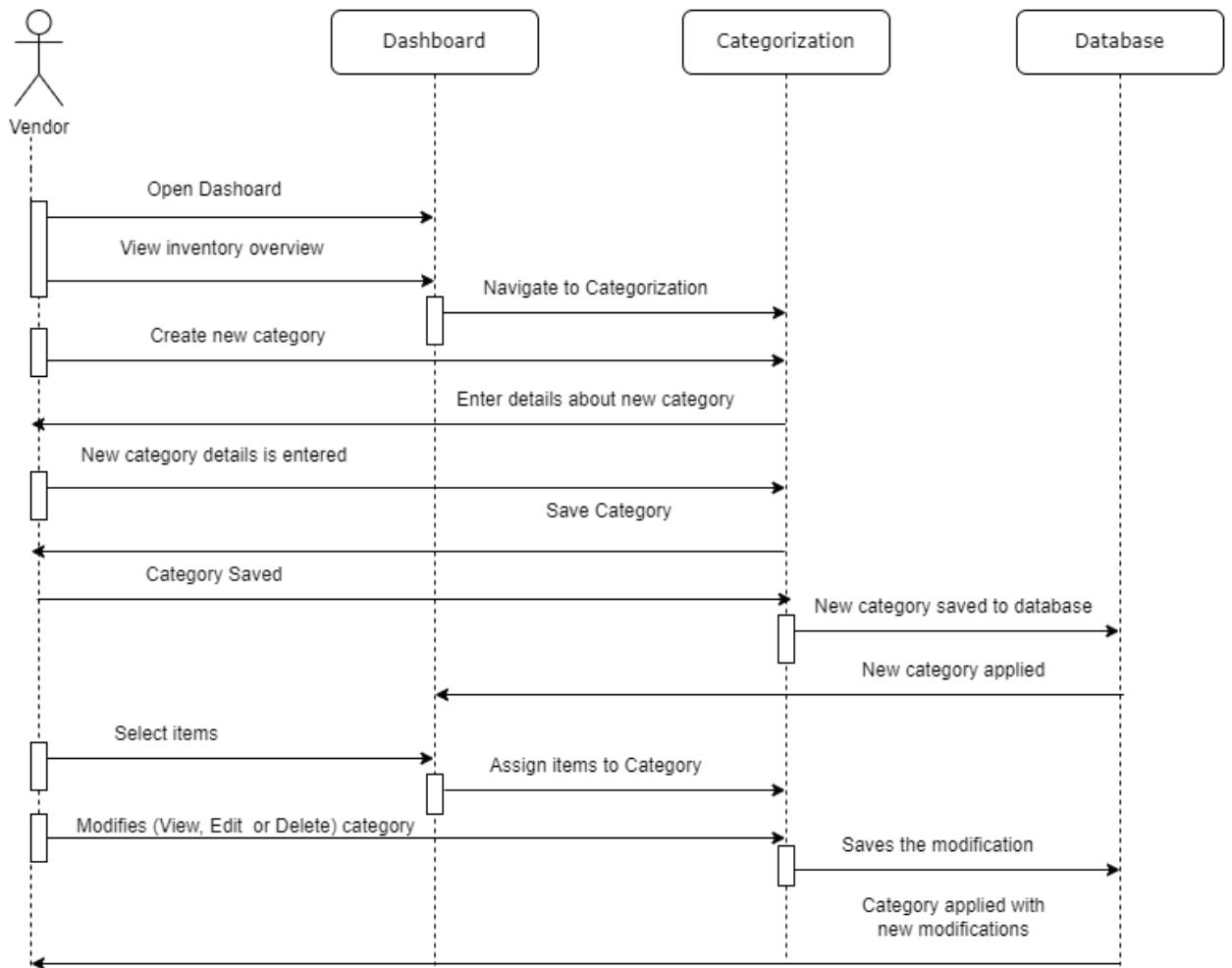


Figure 8.2 SEQUENCE DIAGRAM FOR CATEGORIZING ITEMS FOR VENDOR

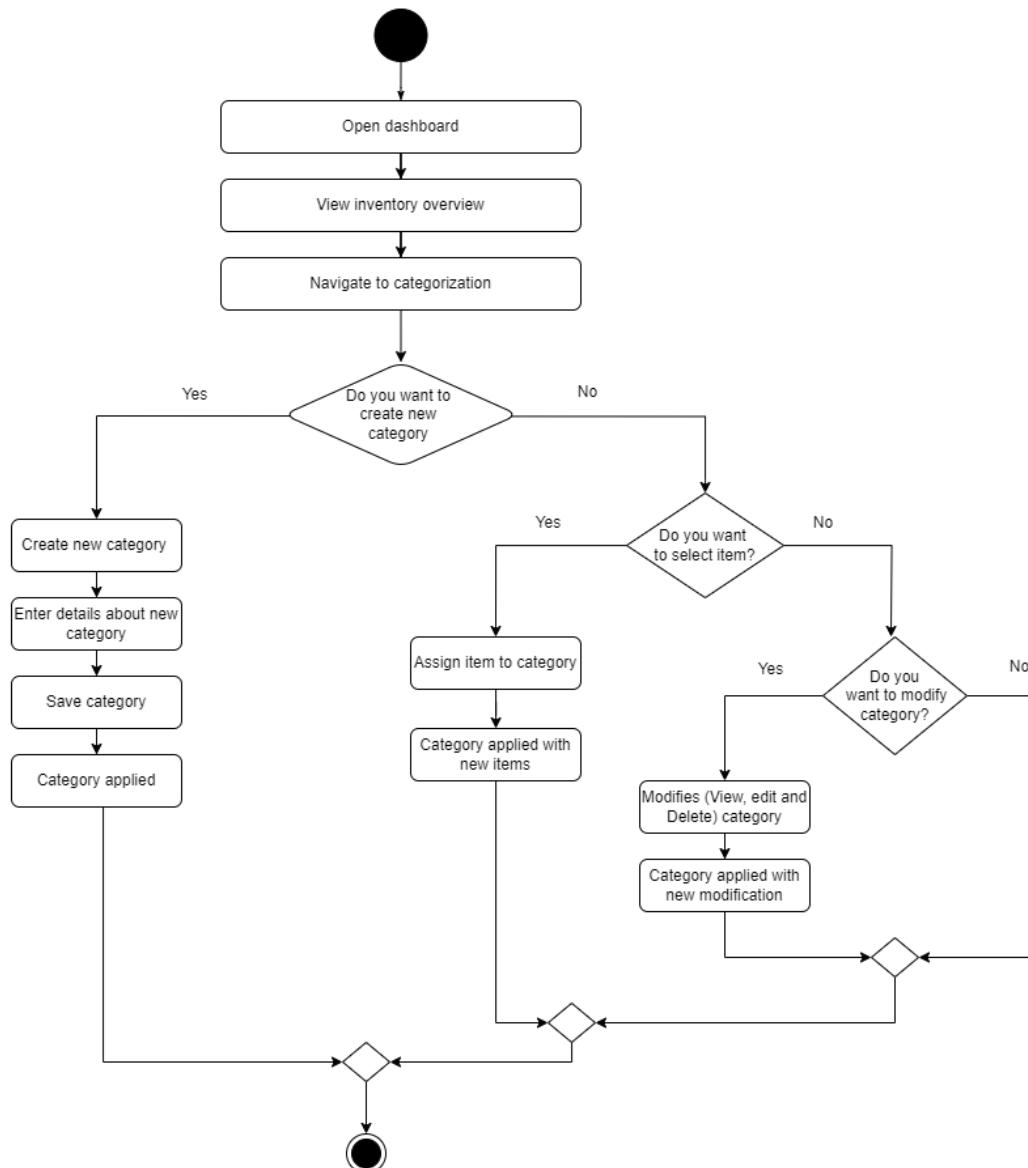


Figure 8.3 ACTIVITY DIAGRAM FOR CATEGORIZING ITEMS FOR VENDOR

9 use case scenario FOR MANAGE AVAILABILITY OF ITEMS (FR3.5)

| | |
|--------------------|--|
| Use-case Name | Manage Availability |
| Use-case ID | UC008 |
| Description | The system should provide functionality for vendors to manage the availability of their items in real-time. |
| Primary actor | Vendor |
| Secondary actor | None |
| Precondition | <ul style="list-style-type: none"> 1- The system is properly installed and configured. 2- Vendors have valid accounts and credentials. 3- The vendor's items are properly registered in the system. |
| Main Scenario | <ul style="list-style-type: none"> 1- The vendor logs into the system using their username or email and password. 2- The system displays the current inventory of the vendor's items. 3- The vendor selects a specific item and updates its availability status (e.g., in stock, out of stock, limited stock). 4- The system immediately reflects the changes in real-time for customers browsing the platform. 5- The vendor logs out of the system. |
| Alternate Scenario | The system notifies the vendor of the incorrect login and prompts them to enter valid credentials. |
| Post condition | <ul style="list-style-type: none"> 1- Changes made by the vendor to item availability are immediately reflected on the platform. 2- The system logs all updates to item availability for audit purposes. 3- Vendors can access historical sales and analytics data |

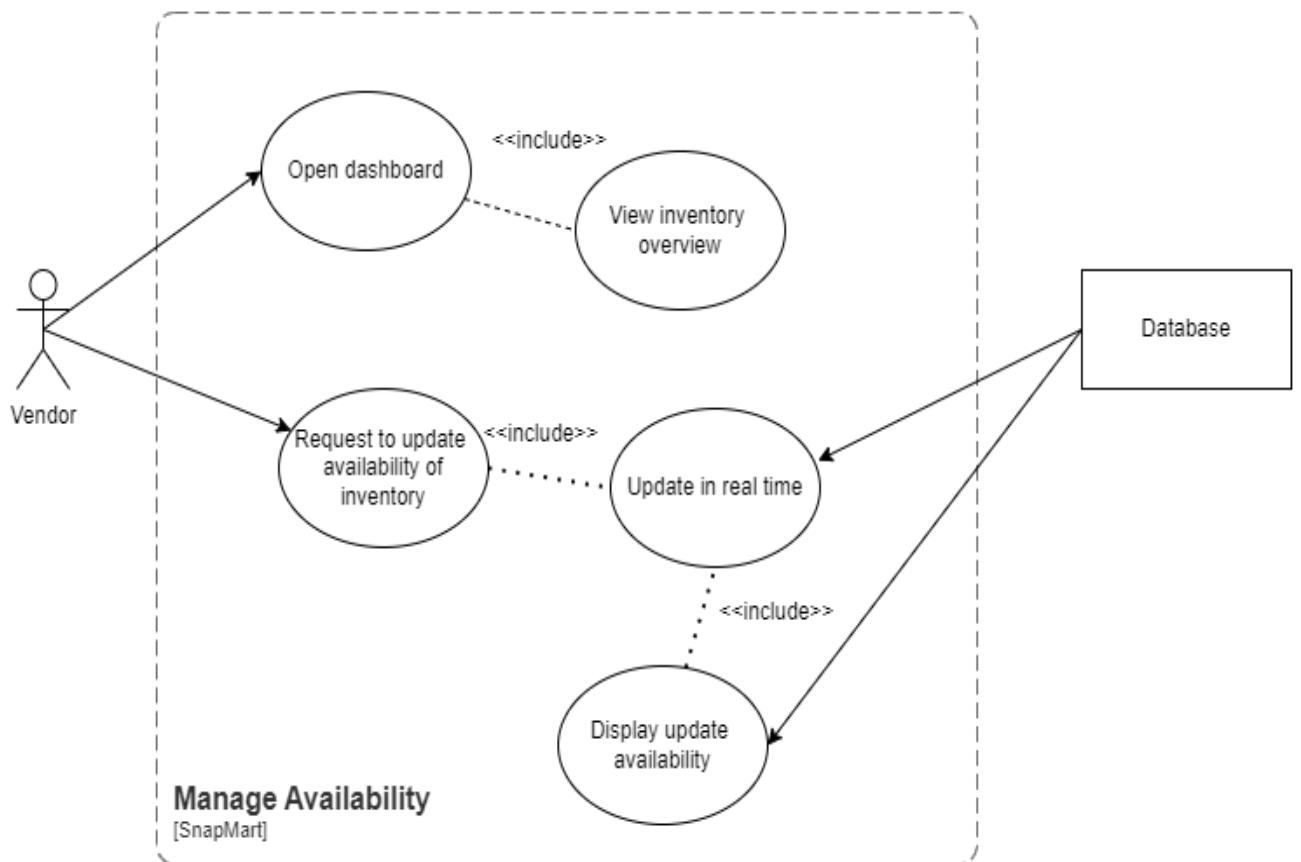


Figure 9.1 USE CASE DIAGRAM FOR MANAGE AVAILABILITY OF ITEMS

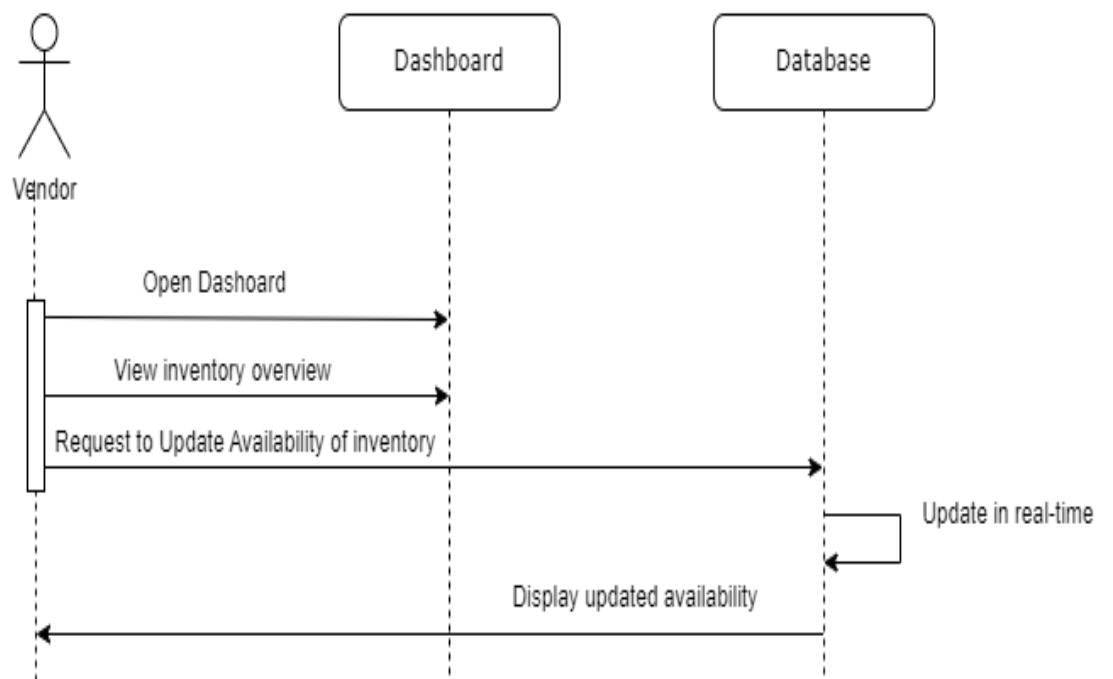


Figure 9.2 SEQUENCE DIAGRAM FOR MANAGE AVAILABILITY OF ITEMS

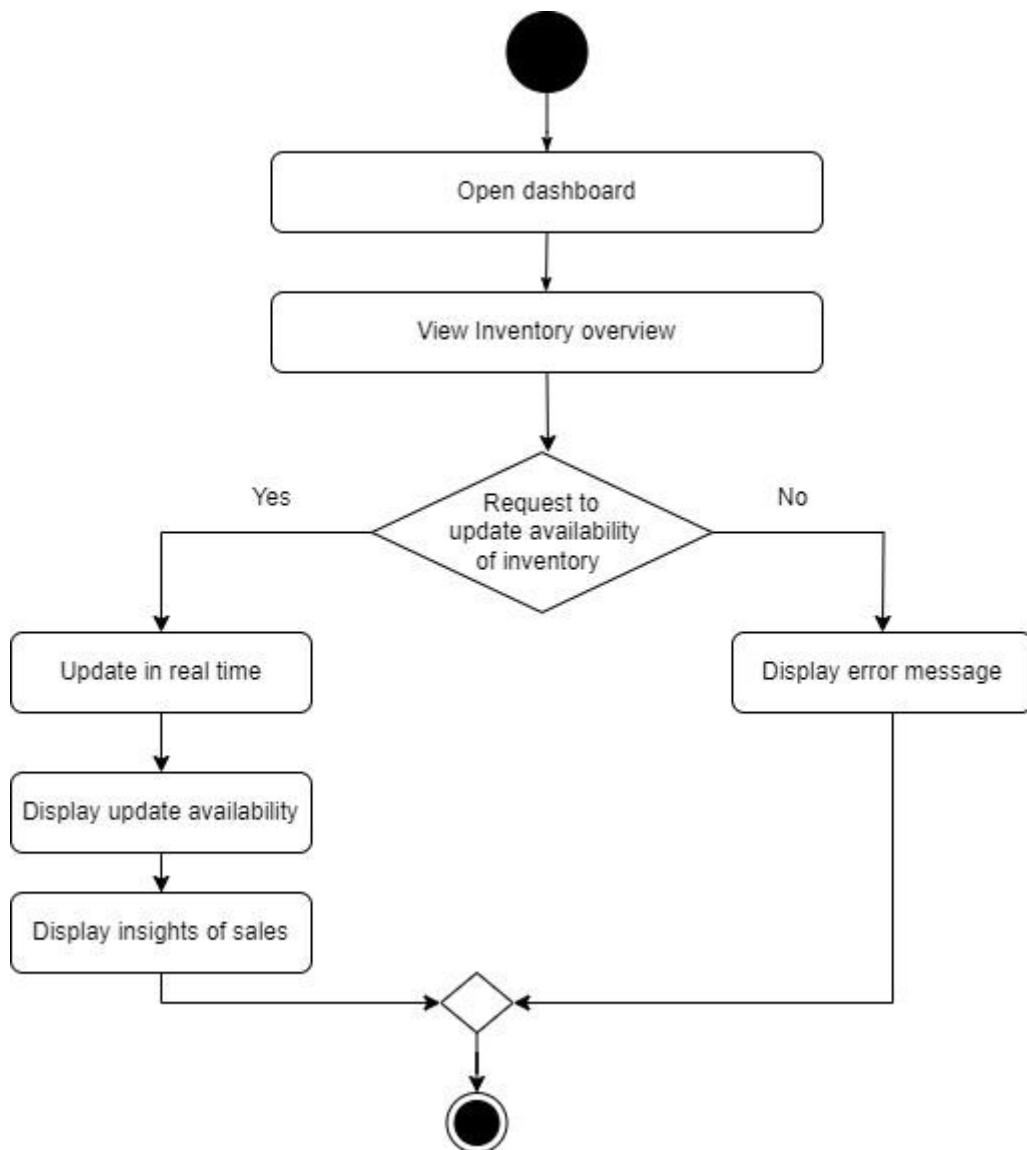
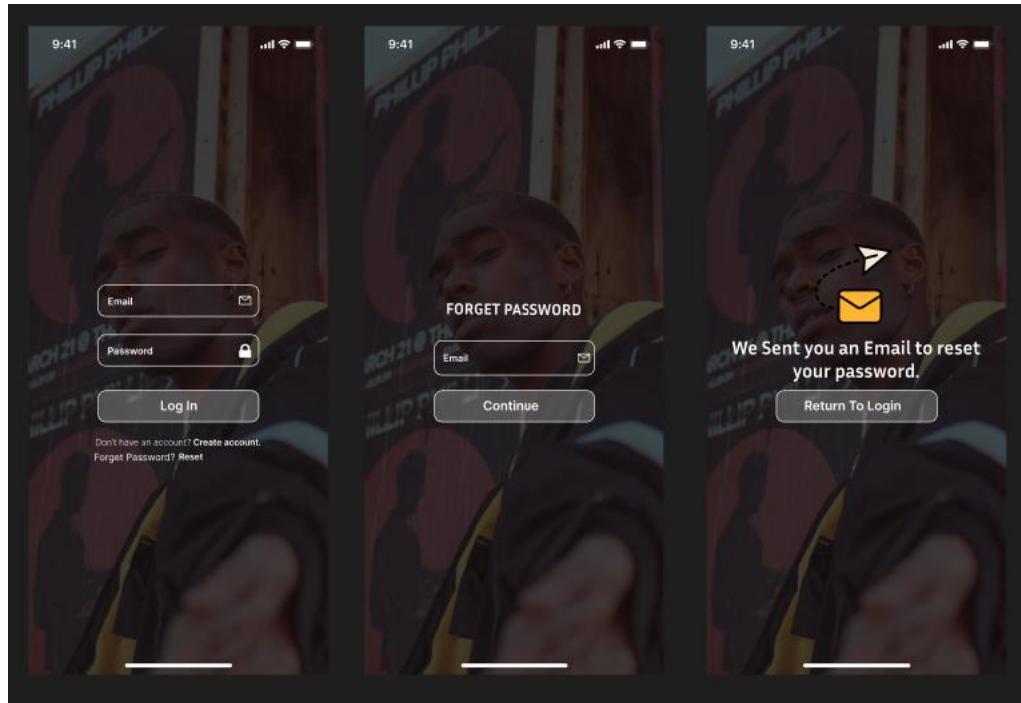
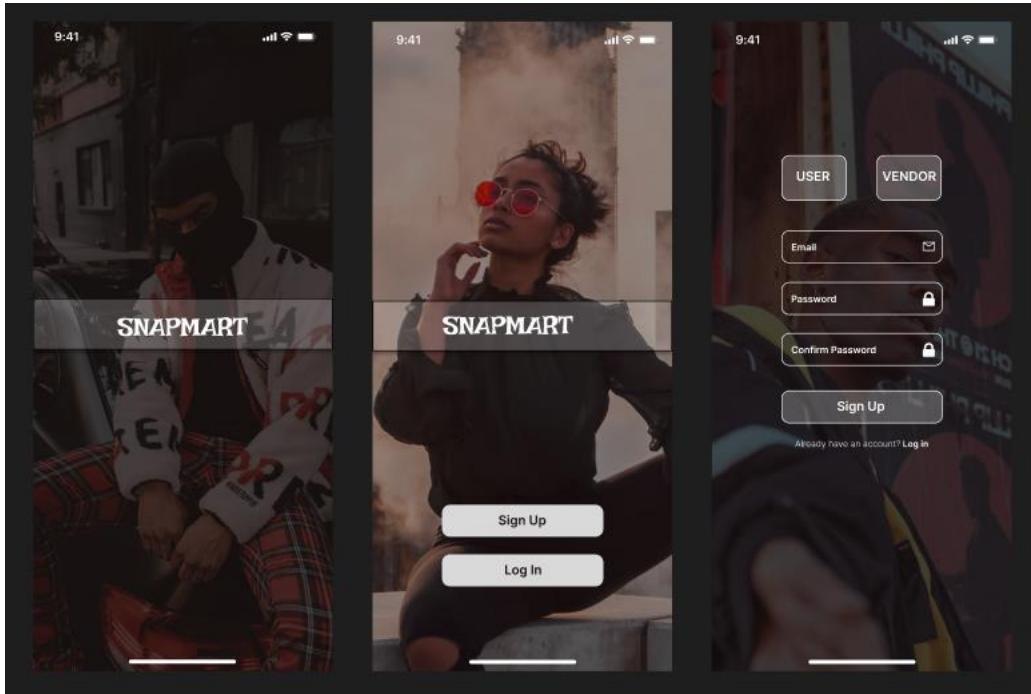
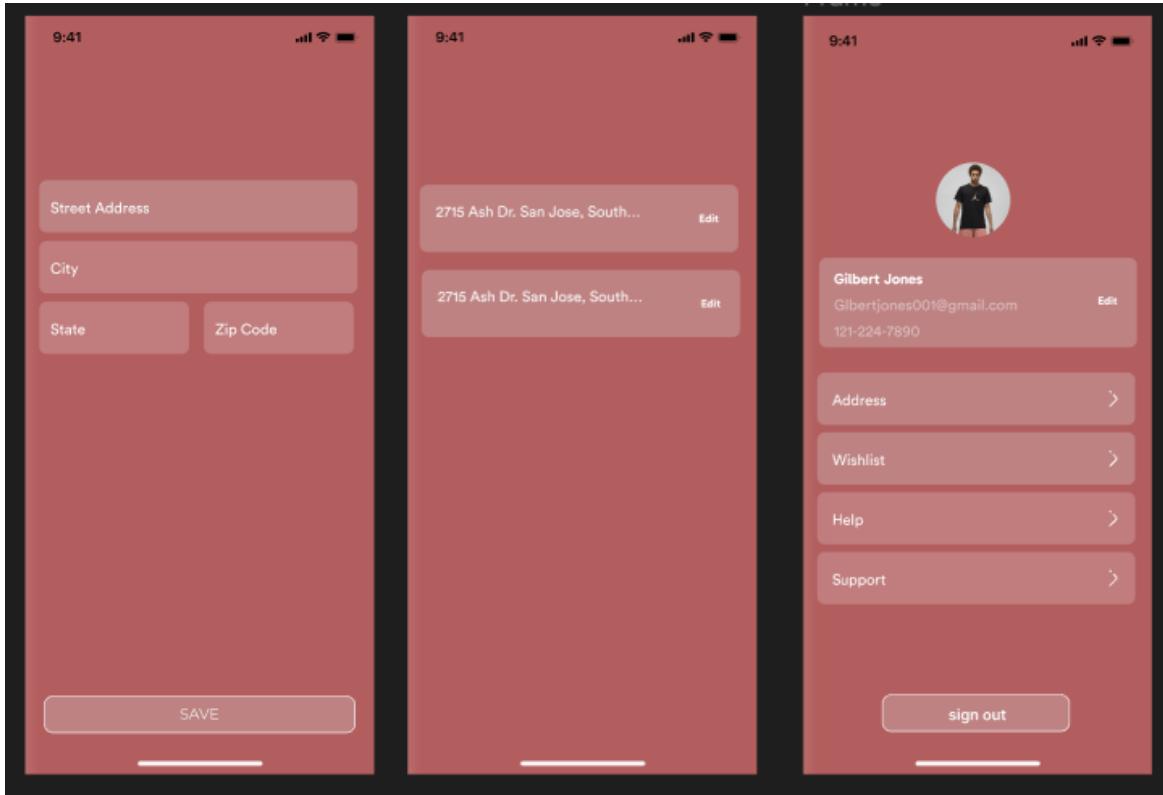
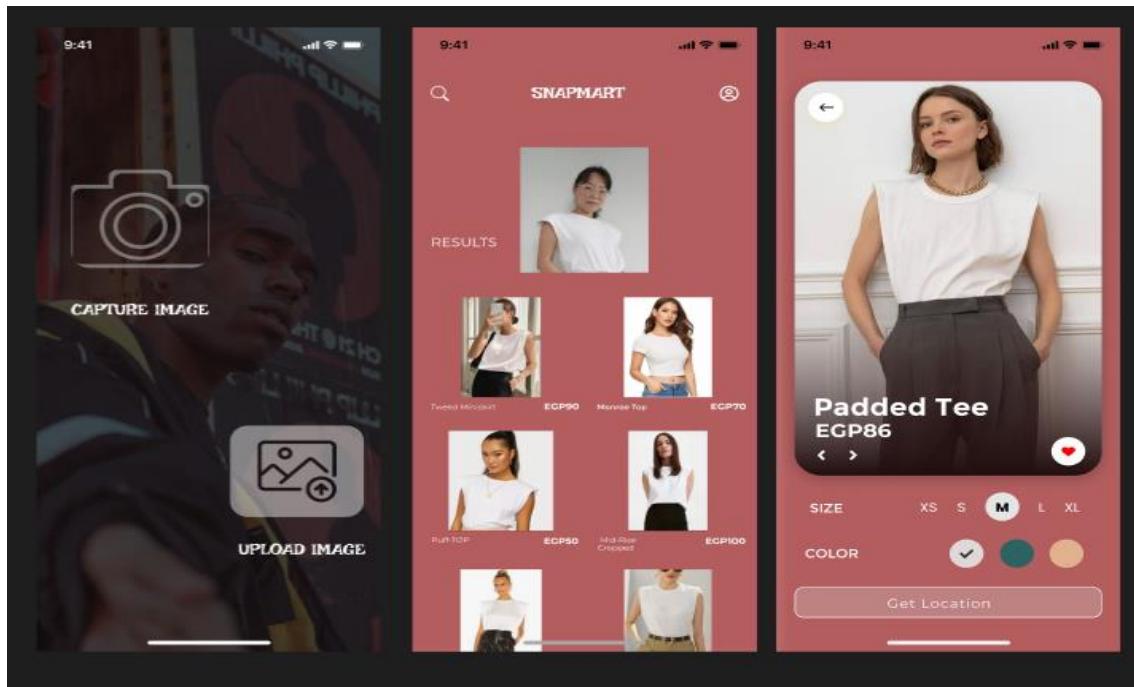
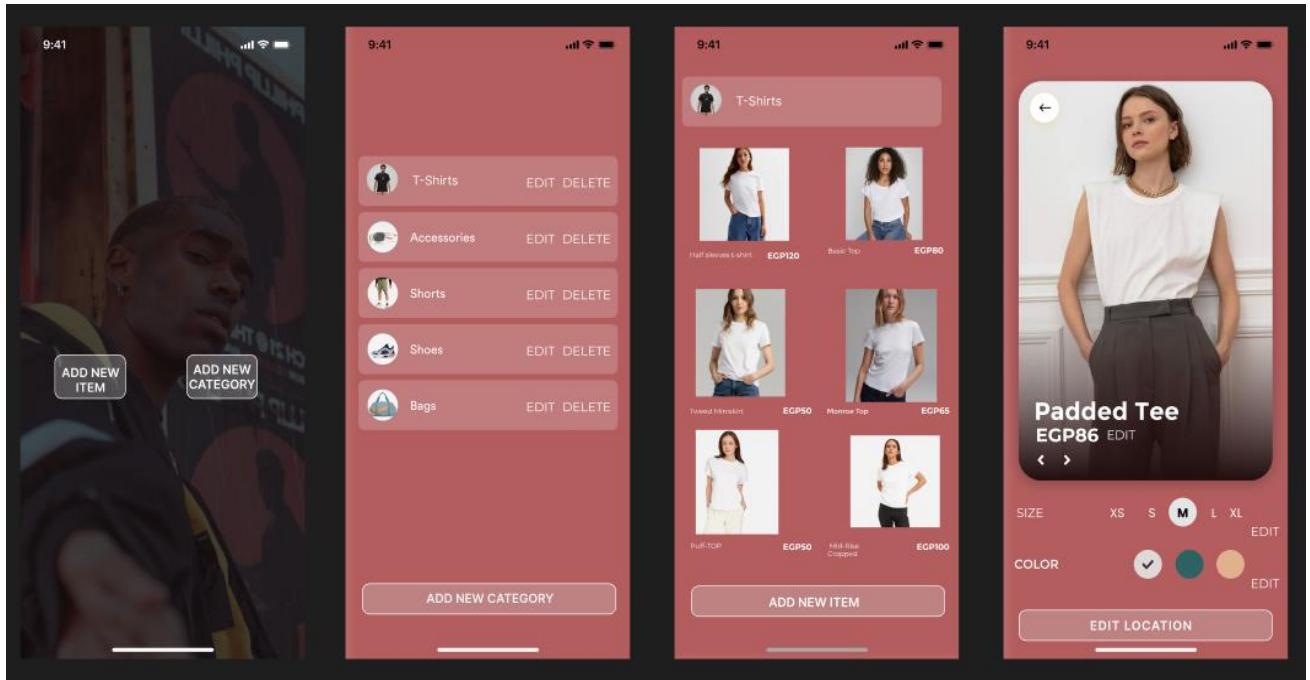


Figure 9.3 ACTIVITY DIAGRAM FOR MANAGE AVAILABILITY OF ITEMS

4.6 User interface





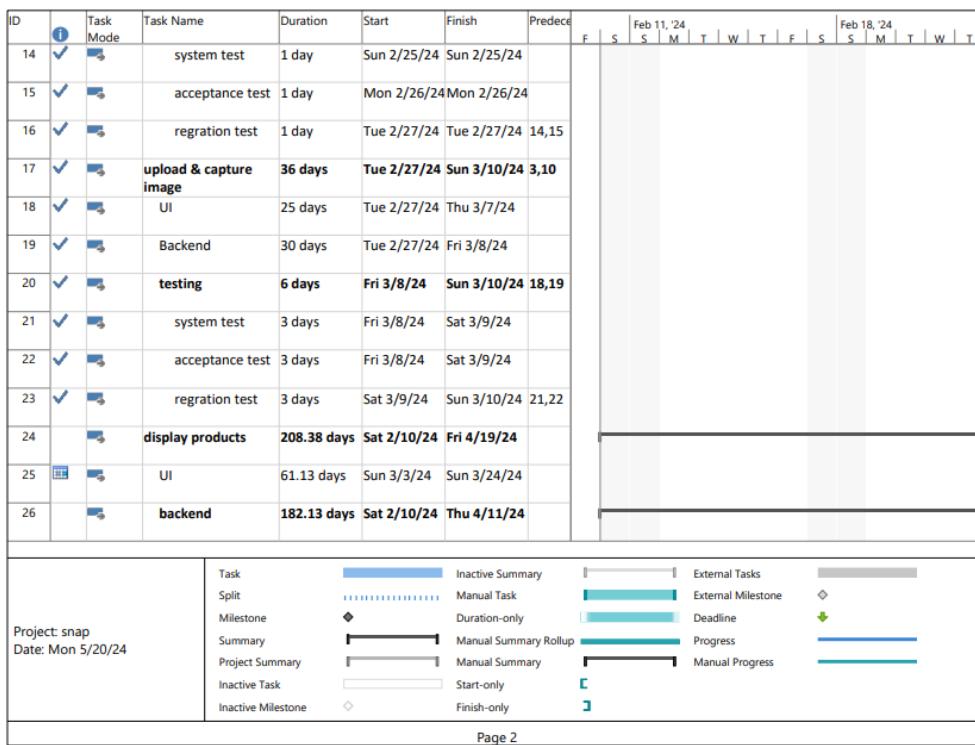
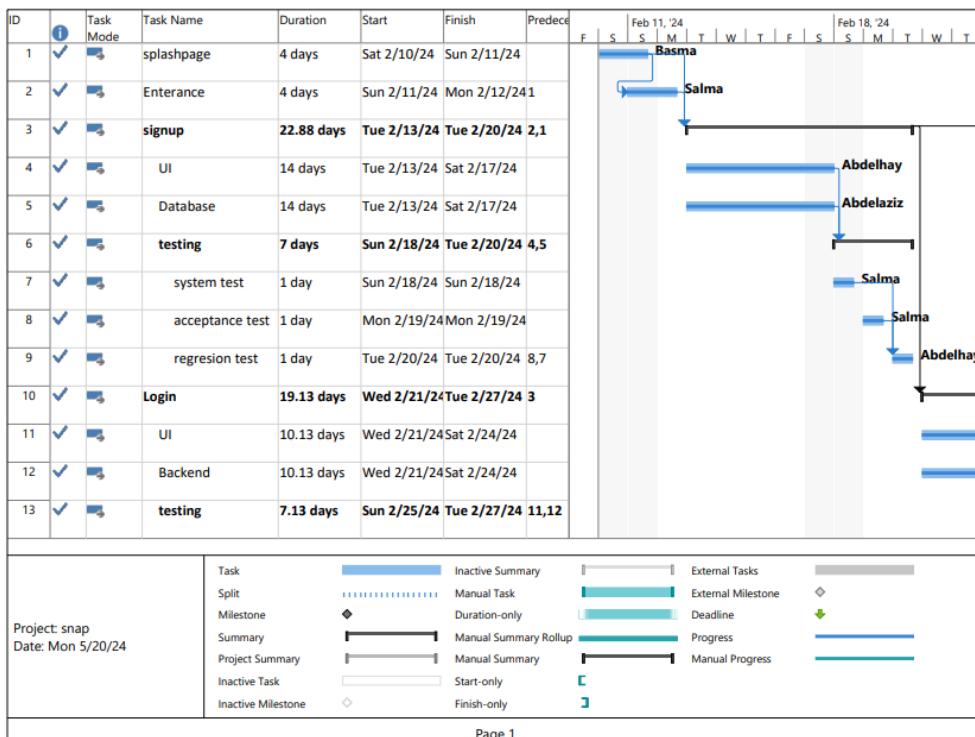


- [https://www.figma.com/file/MmULET28g6a1bjJHux8VXB/Shopping-E-commerce-app-design-\(Community\)?type=design&node-id=0%3A1&mode=design&t=NOzdGI3fpTZ3OZuS-1](https://www.figma.com/file/MmULET28g6a1bjJHux8VXB/Shopping-E-commerce-app-design-(Community)?type=design&node-id=0%3A1&mode=design&t=NOzdGI3fpTZ3OZuS-1)

Chapter 5

Implementation

1. planning



| ID | Task Mode | Task Name | Duration | Start | Finish | Predece | F | S | S | M | T | W | F | S | S | M | T | W | F | | | |
|----|-----------|---|------------|-------------|-------------|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|
| 27 | | get all products | 35 days | Sat 2/10/24 | Mon 3/11/24 | | | | | | | | | | | | | | | | | |
| 28 | | get specific prod | 55 days | Mon 3/11/24 | Thu 4/11/24 | 27 | | | | | | | | | | | | | | | | |
| 29 | | testing | 26.25 days | Thu 4/11/24 | Fri 4/19/24 | 25,26 | | | | | | | | | | | | | | | | |
| 30 | | system test | 6 days | Thu 4/11/24 | Sat 4/13/24 | | | | | | | | | | | | | | | | | |
| 31 | | acceptance test | 10.13 days | Sat 4/13/24 | Tue 4/16/24 | | | | | | | | | | | | | | | | | |
| 32 | | regretion test | 10.13 days | Tue 4/16/24 | Fri 4/19/24 | 30,31 | | | | | | | | | | | | | | | | |
| 33 | | profile customer | 34 days | Sun 3/10/24 | Fri 3/22/24 | 17 | | | | | | | | | | | | | | | | |
| 34 | | UI | 20 days | Sun 3/10/24 | Fri 3/22/24 | | | | | | | | | | | | | | | | | |
| 35 | | display user profile (backend) | 20 days | Sun 3/10/24 | Sun 3/17/24 | | | | | | | | | | | | | | | | | |
| 36 | | add location for customer | 61.88 days | Fri 3/22/24 | Thu 4/11/24 | 17,33 | | | | | | | | | | | | | | | | |
| 37 | | UI | 3 days | Sun 3/24/24 | Mon 3/25/24 | | | | | | | | | | | | | | | | | |
| 38 | | Locationbased integration testing | 15 days | Fri 3/22/24 | Wed 3/27/24 | | | | | | | | | | | | | | | | | |
| 39 | | | 46.88 days | Wed 3/27/24 | Thu 4/11/24 | 37,38 | | | | | | | | | | | | | | | | |

Project: snap
Date: Mon 5/20/24

Page 3

| ID | Task Mode | Task Name | Duration | Start | Finish | Predece | F | S | S | M | T | W | F | S | S | M | T | W | F | | | |
|----|-----------|---------------------|------------|-------------|-------------|---------|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|
| 40 | | system test | 2 days | Wed 3/27/24 | Wed 3/27/24 | | | | | | | | | | | | | | | | | |
| 41 | | acceptance test | 2 days | Wed 4/10/24 | Thu 4/11/24 | | | | | | | | | | | | | | | | | |
| 42 | | regretion test | 2 days | Thu 4/11/24 | Thu 4/11/24 | 40,41 | | | | | | | | | | | | | | | | |
| 43 | ✓ | profile vendor | 38.13 days | Mon 3/4/24 | Sun 3/17/24 | 10,33 | | | | | | | | | | | | | | | | |
| 44 | ✓ | UI | 17 days | Mon 3/4/24 | Sun 3/17/24 | | | | | | | | | | | | | | | | | |
| 45 | ✓ | display vendor prof | 17 days | Mon 3/4/24 | Sun 3/17/24 | | | | | | | | | | | | | | | | | |
| 46 | | add category | 45 days | Sun 3/17/24 | Mon 4/1/24 | 33,43 | | | | | | | | | | | | | | | | |
| 47 | | UI | 21 days | Fri 3/22/24 | Fri 3/29/24 | | | | | | | | | | | | | | | | | |
| 48 | ✓ | Backend | 38 days | Sun 3/17/24 | Sat 3/30/24 | | | | | | | | | | | | | | | | | |
| 49 | ✓ | add | 23 days | Sun 3/17/24 | Mon 3/25/24 | | | | | | | | | | | | | | | | | |
| 50 | ✓ | update | 15 days | Mon 3/25/24 | Sat 3/30/24 | 49 | | | | | | | | | | | | | | | | |
| 51 | ✓ | delete | 15 days | Mon 3/25/24 | Sat 3/30/24 | 49 | | | | | | | | | | | | | | | | |
| 52 | | testing | 10 days | Fri 3/29/24 | Mon 4/1/24 | 47 | | | | | | | | | | | | | | | | |

Project: snap
Date: Mon 5/20/24

Page 4

| ID | Task Mode | Task Name | Duration | Start | Finish | Predecessors | F | S | S | M | T | W | T | F | S | S | M | T | W | T |
|----|-----------|-----------------|------------|-------------|-------------|--------------|---|---|---|---|-------------|---|---|---|---|-------------|---|---|---|---|
| | | | | | | | | | | | Feb 11, '24 | | | | | Feb 18, '24 | | | | |
| 53 | Normal | system test | 2 days | Fri 3/29/24 | Fri 3/29/24 | | | | | | | | | | | | | | | |
| 54 | Normal | acceptance test | 2 days | Fri 3/29/24 | Fri 3/29/24 | | | | | | | | | | | | | | | |
| 55 | Normal | regration test | 8 days | Fri 3/29/24 | Mon 4/1/24 | 53,54 | | | | | | | | | | | | | | |
| 56 | Normal | add item | 55.13 days | Sat 3/30/24 | Wed 4/17/24 | 446 | | | | | | | | | | | | | | |
| 57 | Normal | UI | 22 days | Mon 4/1/24 | Mon 4/8/24 | | | | | | | | | | | | | | | |
| 58 | Normal | Backend | 38 days | Sat 3/30/24 | Thu 4/11/24 | | | | | | | | | | | | | | | |
| 59 | Normal | add | 23 days | Sat 3/30/24 | Sat 4/6/24 | | | | | | | | | | | | | | | |
| 60 | Normal | update | 15 days | Sat 4/6/24 | Thu 4/11/24 | 59 | | | | | | | | | | | | | | |
| 61 | Normal | delete | 15 days | Sat 4/6/24 | Thu 4/11/24 | 59 | | | | | | | | | | | | | | |
| 62 | Normal | testing | 17.13 days | Thu 4/11/24 | Wed 4/17/24 | 57,58 | | | | | | | | | | | | | | |
| 63 | Normal | system test | 1 day | Thu 4/11/24 | Fri 4/12/24 | | | | | | | | | | | | | | | |
| 64 | Normal | acceptance test | 1 day | Thu 4/11/24 | Fri 4/12/24 | | | | | | | | | | | | | | | |
| 65 | Normal | regration test | 3 days | Tue 4/16/24 | Wed 4/17/24 | 63,64 | | | | | | | | | | | | | | |

Project: snap
Date: Mon 5/20/24

Task: Inactive Summary, Split: Manual Task, Milestone: External Tasks, Duration-only: External Milestone, Summary: Manual Task Rollup, Project Summary: Deadline, Inactive Task: Progress, Inactive Milestone: Manual Progress.

Page 5

| ID | Task Mode | Task Name | Duration | Start | Finish | Predecessors | F | S | S | M | T | W | T | F | S | S | M | T | W | T |
|----|-----------|---------------------------|----------|-------------|-------------|--------------|---|---|---|---|-------------|---|---|---|---|-------------|---|---|---|---|
| | | | | | | | | | | | Feb 11, '24 | | | | | Feb 18, '24 | | | | |
| 66 | Normal | Machine learning | 200 days | Sat 2/10/24 | Wed 4/17/24 | | | | | | | | | | | | | | | |
| 67 | Normal | integration | 80 days | Sat 2/10/24 | Fri 3/8/24 | | | | | | | | | | | | | | | |
| 68 | Normal | data collection | 40 days | Sat 2/10/24 | Fri 2/23/24 | | | | | | | | | | | | | | | |
| 69 | Normal | preprocessing | 40 days | Fri 2/23/24 | Fri 3/8/24 | 68 | | | | | | | | | | | | | | |
| 70 | Normal | inovation and integration | 120 days | Fri 3/8/24 | Wed 4/17/24 | 67 | | | | | | | | | | | | | | |
| 71 | Normal | training | 80 days | Fri 3/8/24 | Wed 4/3/24 | | | | | | | | | | | | | | | |
| 72 | Normal | model developm | 40 days | Wed 4/3/24 | Wed 4/17/24 | 71 | | | | | | | | | | | | | | |
| 73 | Normal | Location based service | 50 days | Sat 2/10/24 | Tue 2/27/24 | | | | | | | | | | | | | | | |
| 74 | Normal | system test | 26 days | Wed 4/17/24 | Thu 4/25/24 | 66 | | | | | | | | | | | | | | |

Project: snap
Date: Mon 5/20/24

Task: Inactive Summary, Split: Manual Task, Milestone: External Tasks, Duration-only: External Milestone, Summary: Manual Task Rollup, Project Summary: Deadline, Inactive Task: Progress, Inactive Milestone: Manual Progress.

Page 6

2. implementation in details

-Front End part

Introduction

The front-end implementation of SnapMart, a mobile e-commerce application, plays a critical role in delivering an engaging and intuitive user experience. Developed using Flutter, a robust open-source framework by Google, and Dart, a versatile programming language, the SnapMart app leverages modern mobile development practices to provide a seamless shopping experience on both iOS and Android platforms.

Flutter was chosen for its ability to facilitate cross-platform development, allowing us to write code once and deploy it across multiple platforms with a consistent look and feel. This choice not only streamlined the development process but also ensured that the app could reach a wider audience without the need for separate codebases for iOS and Android.

Dart, as the underlying programming language for Flutter, brings performance benefits by compiling to native code, ensuring smooth animations and quick load times. Additionally, Dart's concise syntax and powerful features like asynchronous programming enabled us to build a responsive and efficient application.

Project Organization

The SnapMart mobile e-commerce app is organized in a way that promotes modularity, readability, and maintainability. The overall structure of the Flutter project includes the following main directories and files:

- **lib/**: This is the primary directory containing all the Dart source code files. It is organized into several subdirectories to separate different aspects of the app.
 - **screens/**: Contains all the screen widgets representing different pages of the app. Each screen is defined in its own Dart file.
 - `Splashpage.dart`: initial screen shown to the user while the app is loading.
 - `onboarding.dart`: A series of screens to introduce the app's features.
 - `login_screen.dart`: Defines the login screen where users can sign in.
 - `signup_screen.dart`: Defines the signup screen where new users can create an account.
 - `Enter.dart`: have the search by photo (camera or upload) or text
 - `Display_item.dart`: display the item matched to the uploaded item
 - `profile.dart`: have details of user
 - `addaddress.dart`: have addresses to search by
 - `help_page.dart`: make user to contact with us if he have any issue
 - `vendor_details.dart`: have brand name and the items that customer add
 - `additem.dart`: Help vendor to add item and details of this item
- **assets/**: This directory contains static resources such as images, fonts, and other files needed by the app.
 - **test/**: Contains unit and widget tests to ensure the app's functionality and stability.

- `main.dart`: The entry point of the Flutter application. This file contains the `main` function and sets up the app's root widget.
- `pubspec.yaml`: The configuration file for the Flutter project. It lists dependencies, assets, and other settings required by the app.

Reusable Components

In the SnapMart mobile e-commerce app, several custom widgets were created to ensure a consistent look and feel throughout the app. These reusable components help to maintain a clean and modular codebase, making it easier to manage and update the app's UI. Below are some of the key reusable components:

1. Custom Buttons:

- **PrimaryButton**: A custom button widget designed to be the primary call-to-action button throughout the app. It features a consistent color scheme, rounded corners, and a text style that aligns with the app's branding

2. Input Fields:

- **CustomTextField**: A reusable text input field with customizable hint text, icon, and validation logic. It ensures a consistent design for all input fields across the app.

Custom AppBar

One of the standout features of the SnapMart app is the unique design of the AppBar, achieved using a custom clipper called `RoundedAppBarClipper`. This custom AppBar design provides a distinctive and visually appealing header for the app.

Networking and API Integration

HTTP Requests

In the SnapMart mobile e-commerce app, we utilized the `http` package to facilitate communication with our backend API. This package allows us to perform various tasks such as fetching product data, submitting orders, and managing user accounts. Here's an overview of how we implemented HTTP requests in our app:

1. **Fetching Product Data**: To display products in the app, we need to fetch product data from the backend. We perform a GET request to the API endpoint that provides the product details.
2. **Managing User Accounts**: User authentication and management are critical aspects of the app. We handle user login by sending a POST request with the user's credentials to the API endpoint.

2-Machine learning part

Importing Libraries:

Libraries such as NumPy and pandas are fundamental for data manipulation and analysis.

TensorFlow and Keras are crucial for deep learning tasks, including image processing and neural network modeling. Matplotlib is used for visualizations, while Plotly offers interactive plotting capabilities. Threading libraries facilitate parallel execution, enhancing performance in certain tasks.

2. Loading Data:

Loading data from CSV files (images.csv and styles.csv) into pandas DataFrames allows for structured handling of tabular data. This step is foundational for subsequent data cleaning, merging, and analysis.

3. Data Cleaning and Merging:

- Extracting 'id' column: From the filenames in the images DataFrame, extracting unique identifiers ('id') helps link image data with style information.
- Merging DataFrames: Combining styles and images based on the extracted 'id' column creates a unified dataset. This integration enhances the dataset's richness by associating detailed style attributes with each image.

4. Filtering and Processing:

- Column Dropping: Removing irrelevant columns ('season', 'link', 'file_found', 'year') streamlines the dataset, focusing on essential attributes.
- Article Type Filtering: Selecting specific article types (e.g., Shirts, Tshirts, Watches) tailors the dataset for targeted analysis or modeling, excluding less relevant categories.

5. Visualization:

- Bar Plot: Visualizing the distribution of selected article types provides insights into their frequency within the dataset. This graphical representation aids in understanding data balance or imbalance across different categories.

6. Article Type Listing:

- Printing unique article types confirms the effectiveness of filtering, ensuring that only desired categories are retained for further processing.

7. Exclusion of Certain Subcategories:

- Removing subcategories like 'Free Gifts' or 'belts' refines the dataset to focus exclusively on primary fashion items. This step enhances dataset purity and relevance to specific modeling objectives.

8. Assigning Random Prices:

- Price Range Definition: Establishing price ranges for different subcategories adds a pricing dimension to the dataset.
- Random Price Assignment: Randomly assigning prices within defined ranges introduces variability, mimicking real-world pricing diversity among fashion products.

9. Adding Location and Vendor Information:

- Random Assignment: Randomly assigning vendors and locations introduces variability, reflecting diverse sourcing and distribution patterns in retail environments.
- Consistent Attributes: Ensuring consistency (e.g., fixed location for 'Blazers') maintains data integrity, crucial for reliable analysis and modeling outcomes.

10. Adding Available Size Information:

- Size Information Inclusion: Incorporating available size data enhances dataset completeness, crucial for sizing-related analyses or user-facing applications.

11. Counting Articles by Type:

- Grouping and Counting: Grouping by 'articleType' and counting occurrences provides an overview of dataset distribution, highlighting dominant and minority categories.

12. Handling Missing Values:

- Null Value Check: Verifying null values in the 'Price' column ensures data completeness before subsequent analysis or model training.
- Data Cleansing: Removing rows with null prices enhances dataset cleanliness, eliminating potential biases or inaccuracies in subsequent analyses.

13. Final Dataset Preparation:

- Cleaned Dataset: The final dataset integrates cleaned, augmented, and enriched data attributes (e.g., prices, sizes, vendors).

- **Readiness for Modeling:** Prepared dataset is primed for machine learning models, offering structured data inputs and minimizing preprocessing overhead during model training.

Data Augmentation:

- **ImageDataGenerator:** TensorFlow's `ImageDataGenerator` facilitates on-the-fly augmentation, augmenting training data with variations in rotation, shift, shear, zoom, and flip. This technique enhances model robustness and generalization.

Feature Extraction Using VGG16:

- **Pre-trained Model:** VGG16, pretrained on ImageNet, provides a robust feature extraction framework without the need for extensive training.
- **Feature Extraction Function:** A custom function extracts deep features from images, leveraging VGG16's architecture to capture high-level image representations effectively.

Main Loop for Copying Images:

- **Image Copying Process:** Iteratively copying images from a source directory to a destination folder based on data availability ensures dataset completeness for subsequent analysis or model training.
- **Error Handling:** Handling non-existent images (`skipped_images`) ensures data integrity and prevents processing errors.

Purpose and Usage:

- **GitHub Installation:** Installing a package directly from a GitHub repository extends functionality beyond standard libraries, accessing cutting-edge features or bug fixes.

Machine Learning Part:

- **Data Splitting:** Partitioning data into training and testing sets ensures unbiased model evaluation and validation.
- **Model Training:** `RandomForestClassifier` is chosen for its ensemble learning approach, robustness to overfitting, and applicability to multi-class classification tasks.
- **Model Evaluation:** `Accuracy_score` quantifies model correctness, while `classification_report` provides comprehensive class-wise metrics (precision, recall, F1-score) for detailed performance assessment.

Overall Accuracy:

- **Accuracy Metric:** High accuracy indicates model proficiency in predicting test set labels, crucial for assessing model reliability and effectiveness.

Classification Report:

- **Detailed Metrics:** Precision (accuracy of positive predictions), recall (ability to detect positives), and F1-score (harmonic mean of precision and recall) offer granular insights into model performance across different classes.

Visual Display:

- **Product Comparison:** Displaying top 5 similar products with overlaid details (vendor, price, location, product name, size) enhances user understanding and aids decision-making in fashion retail scenarios.

By providing deep insights into each section, including rationale, methods, and implications, this structured approach ensures comprehensive understanding of your data preparation, augmentation, feature extraction, and machine learning model implementation workflow. This clarity supports effective application and adaptation of these methods in diverse data science and machine learning projects.

Random Forest

Concept: Random Forest is an ensemble learning method used for classification and regression tasks. It operates by constructing multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of individual trees. The "random" in Random Forest refers to:

- **Random Sampling:** Each tree in the forest is trained on a random subset of the training data (bootstrap samples).
- **Random Subset of Features:** At each split in a decision tree, a random subset of features is considered as potential candidates for splitting.

Advantages:

- **High Accuracy:** Random Forests generally yield high accuracy due to the diversity of trees and their averaging effect.
- **Robust to Overfitting:** The ensemble nature and randomness in feature selection help mitigate overfitting, making it more robust compared to individual decision trees.

- **Feature Importance:** It provides a measure of feature importance, which helps in understanding which features contribute most to predictions.

Applications:

- **Classification and Regression:** Commonly used in classification tasks such as spam detection, medical diagnosis, and customer segmentation. Also applied in regression tasks like predicting housing prices and stock market trends.
- **Anomaly Detection:** Identifying outliers or anomalous data points in various domains.
- **Feature Selection:** Used to determine the most significant features in a dataset.

Implementation Considerations:

- **Parameter Tuning:** Important parameters include the number of trees (`n_estimators`), depth of each tree (`max_depth`), and the number of features considered for splitting (`max_features`).
- **Computational Efficiency:** Efficient for large datasets due to parallelization and the ability to handle high-dimensional data.

VGG16

Concept: VGG16 (Visual Geometry Group 16) is a deep convolutional neural network architecture designed by the Visual Geometry Group at the University of Oxford. It was first introduced in the context of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. Key features include:

- **Depth:** It consists of 16 convolutional and fully connected layers.
- **Uniform Architecture:** VGG16 uses 3x3 convolutional filters throughout the network, maintaining a simple and uniform architecture.

Advantages:

- **Effective Feature Extraction:** VGG16 is adept at learning hierarchical features from images, capturing both low-level features (edges, textures) and high-level features (object parts, whole objects).
- **Transfer Learning:** Pretrained on ImageNet, VGG16 serves as a powerful feature extractor. Transfer learning with VGG16 involves using its learned representations for tasks other than the original ImageNet classification.

Applications:

- **Image Classification:** VGG16 is widely used for image classification tasks due to its effectiveness in extracting discriminative features from images.
- **Object Detection:** Used as a backbone network in object detection frameworks like Faster R-CNN and YOLO (You Only Look Once).
- **Feature Extraction:** Features extracted from VGG16 can be used for tasks such as image retrieval, style transfer, and content-based image retrieval.

Implementation Considerations:

- **Computational Resources:** Training VGG16 from scratch requires substantial computational resources due to its depth and number of parameters.
- **Fine-tuning:** Adjusting pretrained VGG16 for a specific task involves freezing some layers (to retain learned features) and fine-tuning others (to adapt to new data).

Integration in the Project

By incorporating Random Forest for robust classification tasks and leveraging VGG16 for effective image feature extraction, your graduation project showcases a blend of ensemble learning and deep learning techniques. These methods not only enhance the accuracy and robustness of your models but also provide meaningful insights into complex datasets, such as fashion product images and attributes.

Attempted Models: CNN, RNN, and KNN

Previous Attempts: In the initial stages of the project, several machine learning models were experimented with to classify and analyze fashion product images. These attempts included Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and K-Nearest Neighbors (KNN).

1. Convolutional Neural Network (CNN):

- **Concept:** CNNs are well-suited for image classification tasks, leveraging hierarchical feature extraction through convolutional layers.
- **Implementation:** A CNN architecture was designed to process fashion product images directly, extracting relevant features such as textures, shapes, and patterns.
- **Challenges:** Despite intensive hyperparameter tuning and architectural adjustments, the CNN model struggled to achieve satisfactory accuracy. The complexity of fashion product

classification, including subtle variations in style and appearance, posed challenges in feature extraction and generalization. The accuracy was 26%.

1. Recurrent Neural Network (RNN):

- **Concept:** RNNs are effective for sequential data processing, suitable for tasks involving time series or sequential dependencies.
- **Implementation:** Attempted to model sequential aspects in fashion product attributes or styles using RNNs, expecting to capture temporal dependencies or style trends.
- **Outcome:** The RNN model encountered difficulties in effectively capturing the diverse and often non-sequential nature of fashion product characteristics. As a result, it did not yield the desired accuracy levels required for robust classification. The accuracy was 100% and this overfitting.

1. K-Nearest Neighbors (KNN):

- **Concept:** KNN is a simple yet effective algorithm for classification tasks based on similarity measures.
- **Implementation:** Utilized KNN to classify fashion products based on their feature vectors, aiming to leverage similarities in attribute space.
- **Limitations:** Despite its simplicity and intuitive nature, KNN faced challenges in scaling with the dataset size and handling high-dimensional feature spaces efficiently. This limited its effectiveness in accurately classifying fashion products with diverse attributes. The accuracy was 62%.

Comparative Analysis and Challenges

Performance Comparison:

- **Accuracy Insights:** Each model (CNN, RNN, KNN) was evaluated based on accuracy metrics, precision, recall, and F1-score. Comparative analysis revealed that these models individually struggled to achieve the desired performance levels due to inherent complexities in fashion product classification.
- **Complexity of Fashion Attributes:** Fashion products often exhibit nuanced differences in style, color, texture, and shape, which posed challenges for traditional machine learning models like CNN, RNN, and KNN to generalize effectively.

Conclusion:

- **Need for Ensemble and Deep Learning Approaches:** The limitations encountered with CNN, RNN, and KNN underscored the necessity of leveraging ensemble methods (such as Random

Forest) and deep learning techniques (like VGG16) for robust feature extraction, hierarchical learning, and model generalization in fashion product classification.

- **Insights for Model Selection:** The comparative analysis provided valuable insights into the strengths and limitations of various machine learning approaches, guiding the selection of Random Forest and VGG16 for their respective strengths in ensemble learning and deep feature extraction.

3-Back End part

Chapter 6

Testing

| | | | |
|-------------------------|--------------------------|-----------------------|--------------------------|
| Test Scenario ID | Login-1 | Test Case ID | Login-1A |
| Test Description | Login-positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Correct Email in English Language in email section, and Correct Password in English Language in the password section and button. | Email: vendorsnapmart.1@gmail.com Password: Vendor003213 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |

| | | | |
|-------------------------|--------------------------|-----------------------|--------------------------|
| Test Scenario ID | Login-2 | Test Case ID | Login-2A |
| Test Description | Login-positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Correct Email in English Language in email section, and Correct Password in English Language in the password section and hit login button. | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-3 | Test Case ID | Login-3A | | | |
|------------------------------|--|---|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Incorrect Email in English Language in email section, and Correct Password in English Language in the password section and hit login button. | Email: Admin368@gmai.com Password: Admin003213 | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| | | | |
|-------------------------|--------------------------|-----------------------|--------------------------|
| Test Scenario ID | Login-4 | Test Case ID | Login-4A |
| Test Description | Login-positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|----------------------------------|----------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Correct Email in English Language in email section, Enter Incorrect Password in English Language in the password section and hit login button. | Email: Admin.1@gmail.com Password: Addmin38907 | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-5 | Test Case ID | Login-5A | | | |
|------------------------------|--|--|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Incorrect Email in English Language in email section, and Correct Password in English Language in the password section and hit login button. | Email: User3892@gmai.com Password: User097538 | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-6 | Test Case ID | Login-6A | | | |
|------------------------------|--|--|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter correct Email in English Language in email section, and Incorrect Password in English Language in the password section and hit login button. | Email: usersnapmart.1@gmail.com Password: Usserr49383 | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-7 | Test Case ID | Login-7A | | | |
|------------------------------|--|-----------------------------------|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Leave email section empty , Leave password section and hit login button. | Email: none Password: none | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-8 | Test Case ID | Login-8A | | | |
|------------------------------|--|---|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Leave email section empty , Enter correct Password in English Language in the password section and hit login button. | Email: none Password: User097538 | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-9 | Test Case ID | Login-9A | | | |
|------------------------------|--|---|----------------------------------|----------------------------------|--------------|-------------|
| Test Description | Login-positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter Correct Email in English Language in email section, Leave password section empty and hit login button. | Email: usersnapmart.1@gmail.com Password: none | Fail to Login to the application | Fail to Login to the application | Mobile App | Pass |

| Test Scenario ID | Login-10 | Test Case ID | Login -10A | | | |
|------------------------------|---|---|--------------------------------------|--------------------------------------|--------------|-------------|
| Test Description | Login -positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Email section an email exceed 50 character | Email: Userapplicationfor SnapMartjobTo ApplyInThisNow@gmail.com | Do not accept more than 50 character | Do not accept more than 50 character | Mobile App | Pass |

| Test Scenario ID | Login-11 | Test Case ID | Login -11A | | | |
|------------------------------|--|-------------------------------------|--------------------------------------|--------------------------------------|--------------|-------------|
| Test Description | Login -positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Password section a password exceed 20 character | Password: User8Xq4JnB5wzLm3T9pwr | Do not accept more than 20 character | Do not accept more than 20 character | Mobile App | Pass |

| Test Scenario ID | Sign up-1 | Test Case ID | Sign up -1B | | | |
|------------------------------|-------------------------------|-------------------------------|--------------------------|-----------------------|--------------|-------------|
| Test Description | Sign up -positive test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in name section numbers | Name: 274830 | Do not accept numbers | Do not accept numbers | Mobile App | Pass |

| | | | |
|-------------------------|-----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Sign up-2 | Test Case ID | Sign up -2B |
| Test Description | Sign up -positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|--------------------------------------|--------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Email section in an email exceed 50 character | Email: Userapplicationfor SnapMartjobTo ApplyInThisNow@gmail.com | Do not accept more than 50 character | Do not accept more than 50 character | Mobile App | Pass |



| | | | |
|-------------------------|-----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Sign up-3 | Test Case ID | Sign up -3B |
| Test Description | Sign up -positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|--|--------------------------------------|--------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Name section a Correct Name section, Enter in email Section a correct email, Enter in password section correct password , enter in confirm password another password and hit sign up button | Name: User Email: usersnapmart.1@gmail.com Password: User097538 Confirm password: User098667 | Cannot accept to create this account | Cannot accept to create this account | Mobile App | Pass |

| Test Scenario ID | Sign p-4 | Test Case ID | Sign up - 4B | | | |
|------------------------------|--|-------------------------------|------------------------------------|---|--------------|-------------|
| Test Description | Sign up-Negative test case | Test Priority | High | | | |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application | | | |
| Test Execution Steps: | | | | | | |
| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in name section in Arabic language | Name: المستخدم | Accept to write in Arabic language | Do not accept to write in Arabic language | Mobile App | Fail |

| | | | |
|-------------------------|----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Search-1 | Test Case ID | Search -1C |
| Test Description | Search -positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|--|----------------------------------|----------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Search Bar Name of Item in English language | Search : Enter the name of the required item | Success To find the Correct Item | Success To find the Correct Item | Mobile App | Pass |

| | | | |
|-------------------------|----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Search-2 | Test Case ID | Search -2C |
| Test Description | Search -Negative test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|--|-----------------------------------|----------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Enter in Search Bar Name of Item in Arabic language | Search: Enter the name of the required item in Arabic language | Success To find the Correct brand | write Fail to in Arabic language | Mobile App | Fail |

| | | | |
|-------------------------|----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Search-3 | Test Case ID | Search -1C |
| Test Description | Search -positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---------------------------|------------------------------|------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Hit Upload Photos button and select a photo from gallery | Choose photo from gallery | Success To upload this photo | Success To upload this photo | Mobile App | Pass |

| | | | |
|-------------------------|----------------------------|-----------------------|--------------------------|
| Test Scenario ID | Search-4 | Test Case ID | Search -4C |
| Test Description | Search -positive test case | Test Priority | High |
| Pre-Requisite | Open SNAPMART App | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|--|--------------------------------------|--------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Hit Take Photo button and Try to Capture a photo for the required product | Capture the photo for required product | Success To upload the captured photo | Success To upload the captured photo | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|--------------------------|
| Test Scenario ID | Profile-1 | Test Case ID | Profile -1D |
| Test Description | Profile-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|---|--|--|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Hit on profile picture to upload photo ,select a photo from gallery and choose it to be a profile picture | Choose photo from gallery | Success To upload this photo to be a profile picture | Success To upload this photo to be a profile picture | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|--------------------------|
| Test Scenario ID | Profile-2 | Test Case ID | Search -2D |
| Test Description | Search -positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|---|---|---|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Hit Edit Name button and enter the correct name in English language | Name: User | Success To change the name into new one | Success To change the name into new one | Mobile App | Pass |

| | | | |
|-------------------------|---------------------------------------|-----------------------|--------------------------|
| Test Scenario ID | Profile-3 | Test Case ID | Search -3D |
| Test Description | Search -positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Login to the application |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|---|--|--|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Hit Edit Email button and enter the correct email in English language | Email : snapmart0@gmail.com | Success To change the email into new one | Success To change the email into new one | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|----------------|
| Test Scenario ID | Add-Address-1 | Test Case ID | Add-Address-1E |
| Test Description | Add-Address-1-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Add address |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Enter in New Address Section Correct address in English Language, Enter in City Section correct city in English Language, Enter in Postal Code section correct postal code and hit Save button | New Address: Ahram Canadian University City: 6th October Postal Code: 3220023 | Success To add new address | Success To add new address | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|----------------|
| Test Scenario ID | Add-Address-2 | Test Case ID | Add-Address-2E |
| Test Description | Add-Address-2-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Add address |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Leave New Address Section Empty, Leave City Section Empty, Leave Postal Code section Empty and hit Save button | New Address: City: Postal Code: | Fail To add new address | Fail To add new address | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|----------------|
| Test Scenario ID | Add-Address-3 | Test Case ID | Add-Address-3E |
| Test Description | Add-Address-1-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | Edit address |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | hit Edit icon, Edit address in New Address Section and put new one in English Language, Edit city in City Section and put new one in English Language, Edit postal code in Postal Code section and put new one and hit Save button | New Address: Street 9 City: El Mukattem Postal Code: 11571 | Success To Edit address | Success To Edit address | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|-----------------|
| Test Scenario ID | Add-New-Item-1 | Test Case ID | Profile -1F |
| Test Description | Add-New-Item-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | To Add new item |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|---|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Enter product name in product name section in English Language, Enter price in price section, Enter size in size section, Enter Location in English Language in the location section, hit button choose image and choose the required one and hit Add Product button. | Product Name: white-shirt Price: 537 Size: 31 Location: El sheikh Zayed Choose photo from gallery | Success To add new item | Success To add new item | Mobile App | Pass |



| | | | |
|-------------------------|---------------------------------------|-----------------------|-----------------|
| Test Scenario ID | Add-New-Item-2 | Test Case ID | Profile -2F |
| Test Description | Add-New-Item-positive test case | Test Priority | High |
| Pre-Requisite | To have an account on the application | Post-Requisite | To Add new item |

Test Execution Steps:

| S.No | Action | Inputs | Expected Output | Actual Output | Test Browser | Test Result |
|------|--|---|---------------------------------------|---------------------------------------|--------------|-------------|
| 1 | Launch Application. | App Store: Play Store: | SNAPMART Home | SNAPMART Home | Mobile App | Pass |
| 2 | Login to the application | Email: usersnapmart.1@gmail.com Password: User097538 | Successfully Login to the application | Successfully Login to the application | Mobile App | Pass |
| 3 | Leave product name section Empty, Leave price section Empty, Leave size section empty, Leave Location section Empty, do not choose image | Product Name: Price: Size: Location: | Fail To add new item | Fail To add new item | Mobile App | Pass |

Future work:

1. Advanced Search Features

AI-Powered Recommendations: Implement advanced AI algorithms to provide more accurate and personalized recommendations based on user preferences and browsing history.

Voice Search: Integrate voice recognition to allow users to search for items using voice commands.

Enhanced Image Recognition: Improve the image recognition capabilities to better identify and match clothing items with those in the database.

2. User Experience Enhancements

UI/UX Improvements: Continuously refine the user interface and user experience based on user feedback and usability testing.

Dark Mode: Implement a dark mode to provide users with a more comfortable viewing experience, especially in low-light conditions.

Accessibility Features: Enhance accessibility by adding features like screen reader support, larger text options, and high-contrast themes.

References:

1. Deitel, H. M. (2001). *e-Business and e commerce for managers*. United States of America: Prentice Hall. Retrieved from https://drive.google.com/drive/folders/1JCXCC44ykkpllcWJqo22hJI_zuhG2B1
2. <https://www.investopedia.com/terms/l/long-tail.asp>
3. <https://www.amazon.eg/>
4. <https://www.wikipedia.org/>
5. <https://en.wikipedia.org/wiki/E-commerce>
6. <https://www.techtarget.com/searchcio/definition/e-commerce>
7. <https://www.investopedia.com/terms/e/ecommerce.asp>
8. <https://www.the-future-of-commerce.com/2020/01/19/what-is-e-commerce-definition-examples/>
9. <https://www.bigcommerce.com/articles/ecommerce/>
10. <https://www.alibaba.com/>
11. <https://www.jumia.com.eg/>
12. <https://www.noon.com/>
13. <https://flutter.dev/>
14. <https://dart.dev/>
15. <https://code.visualstudio.com/>
16. <https://developer.android.com/>
17. <https://www.microsoft.com/en-us/sql-server/>
- 18.