

SmartFit: Augmented Reality-Integrated Footwear Retail And Customization Portal With Ai-Powered Assistance

SmartFit: Augmented Reality-Integrated Footwear Retail and Customization Portal is a web-based system that enhances the AR shopping experience while also managing transactions and inventory. It allows customers to try on shoes virtually and explore various styles while helping to manage stock and track sales more efficiently.

The goal of this project is to simplify shoe retail by managing orders, tracking inventory, helping employees, and improving customer experience.

Objectives of the Study

The main objective of the study was to develop and implement an Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance, which aims to manage footwear shops and enable customers to virtually fit and customize shoes through AR model viewing and an AI chatbot.

Specifically, the study aims to:

- Design a portal that is capable of:
 - a. Allowing all users to register through the registration module;
 - b. Providing shop owners and add employees with an inventory management module to update stock levels and manage the product catalog through inventory module;

- c. Enabling customers to access the augmented reality website that lets them try on the shoes virtually;
 - d. Offering a basic customization option for shoes that allows customers to personalize their selections and receive cost estimates through the customization module;
 - e. Facilitating customers to place shoe orders easily through the ordering module;
 - f. Implementing a tracking system that allows customers to monitor the status of their shoe orders in real time;
 - g. Integrating AI-powered customer support to assist customers with their inquiries;
 - h. Establishing a feedback system where users can submit ratings and reviews; and
 - i. Creating a report generation to collect data from the system for easy report creation.
- Create a system using Java EE (Jakarta EE), DeepAR, Firebase, an internet connection, and Windows OS (10 or 11) as software requirements and a desktop computer, laptop, printer, router, and a mobile device as a hardware requirements.
 - Test and improve the system in terms of functional suitability, usability, maintainability, and security.

- Evaluate the performance of the system based on ISO 25010 characteristics such as functional suitability, maintainability, usability, performance efficiency, compatibility, reliability, security, and portability.

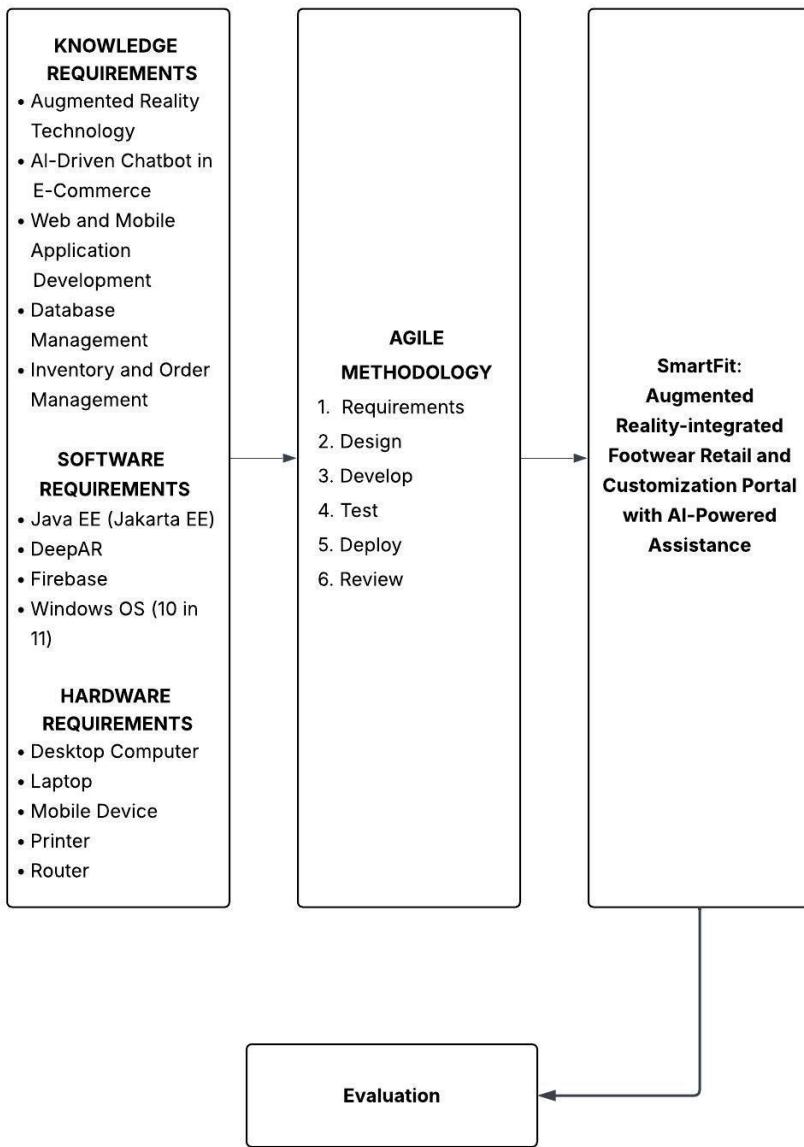


Figure 1.0 Conceptual Model of the Study

Figure 1 represents the conceptual model of the study which includes four phases, namely Input, Process, Output, and Evaluation.

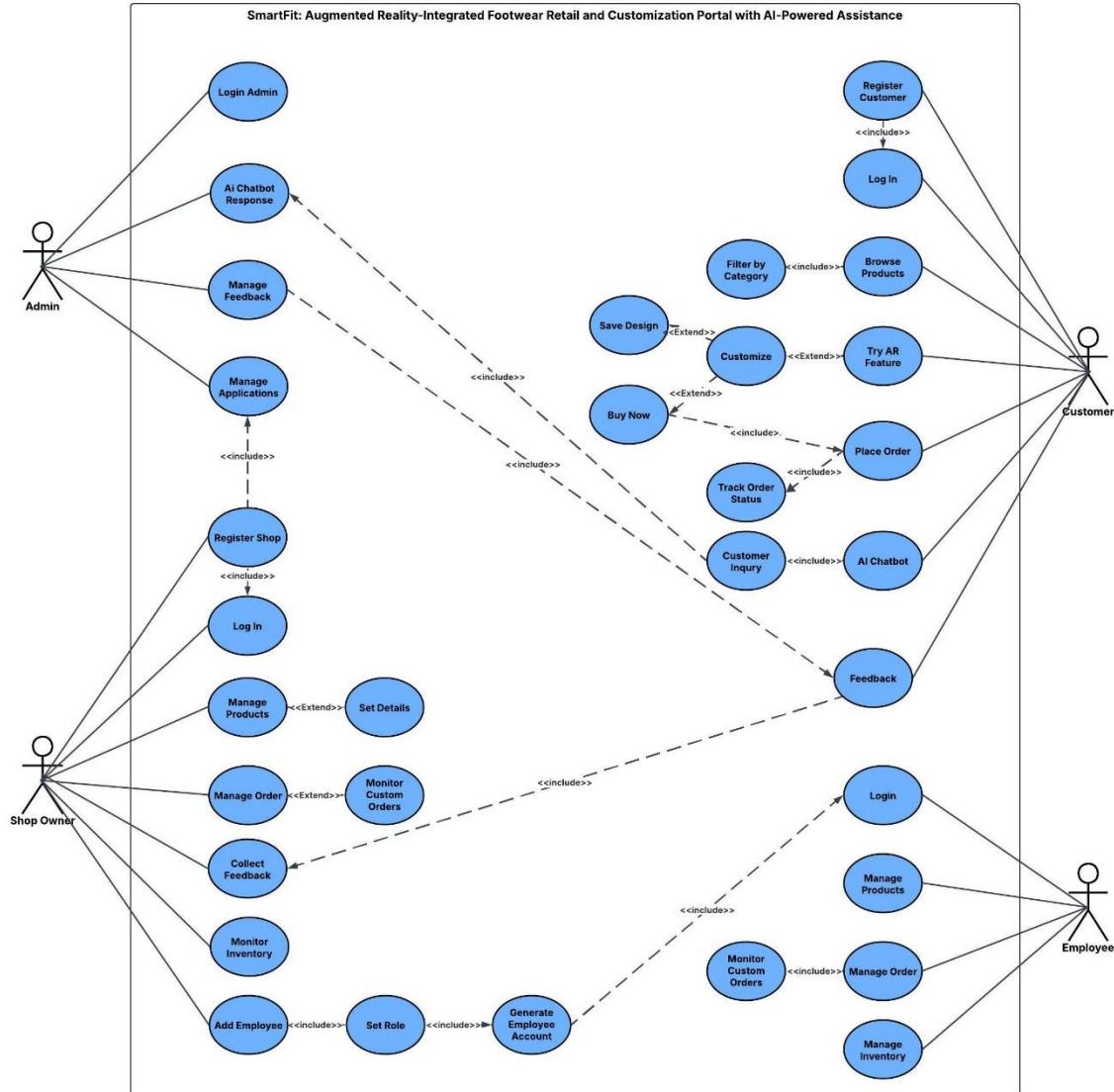


Figure 2.0 Use Case Diagram

Figure 2 shows the use case diagram of the SmartFit platform which is an augmented reality integrated footwear retail and customization portal with AI enhanced support. There are four key actors accommodated by the system; Customer, Shop Owner, Employee, Admin. Customers are able to register or log in, browse and filter products, make use of the AR feature through the website, place and track orders, manage their cart and profile, provide feedback and

engage the AI chatbot. Shop Owners can register their shop, upload documents to be approved and once approved can operate on products, orders, inventory, receive low stock alerts and add employees. Employees have the same set of functionalities that a shop owner would have, such as product admin, order management and stock monitoring. But managers only have access to sales records. Also, employees are not allowed to add other employees. The Admin helps to check and approve or reject shop owners' applications with correct notification, administers AI chatbot responses by adding, editing or deleting entries to provide smooth communication in the platform.

Activity Diagram

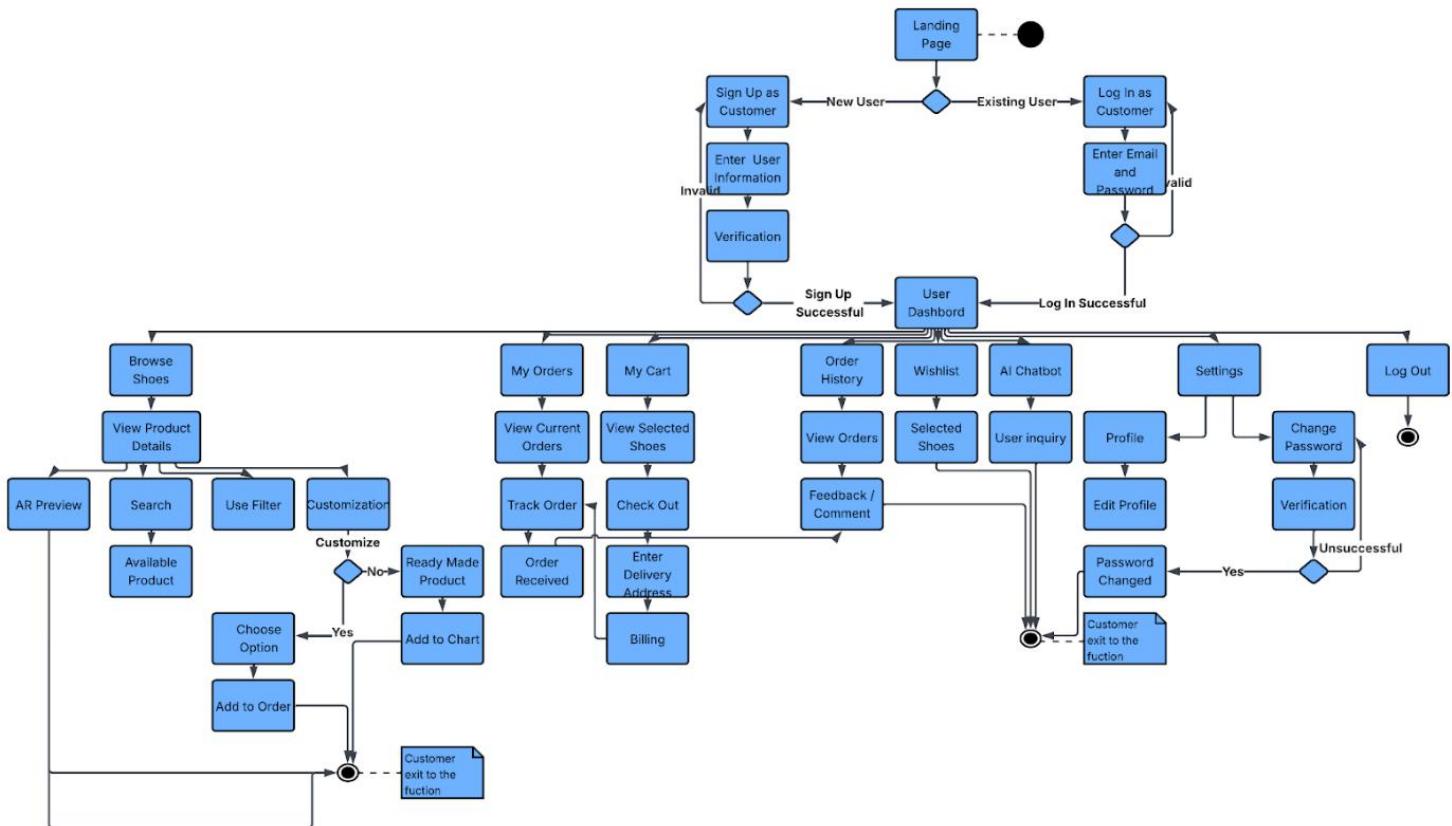


Figure 3.0 Activity Diagram (Customer Side)

In figure 3, In order to access the developed system, customers start from the landing page where they have an option to register or login. The new customer provides necessary information and gets verified through email before accessing the customers dashboard. Current customers can log in by using their credentials. Customers from the dashboard can explore offered footwear, read product descriptions, available stock, use filters and use augmented reality previews with the direct link to web browser. For customizable products, customers input choice options like color that is available, then place adds to cart or place order directly. Customers are able to see and track current orders, access checkout, enter delivery details, and choose payment methods using QR code. Upon successful payment a digital receipt was provided. Failed transactions trigger a retry option. Other features include order's history viewing, placement of feedback, wish list, customer inquiries using AI chatbot. Account settings enable uploading profiles and changing passwords and verify. There is a logout function which safely closes the session.

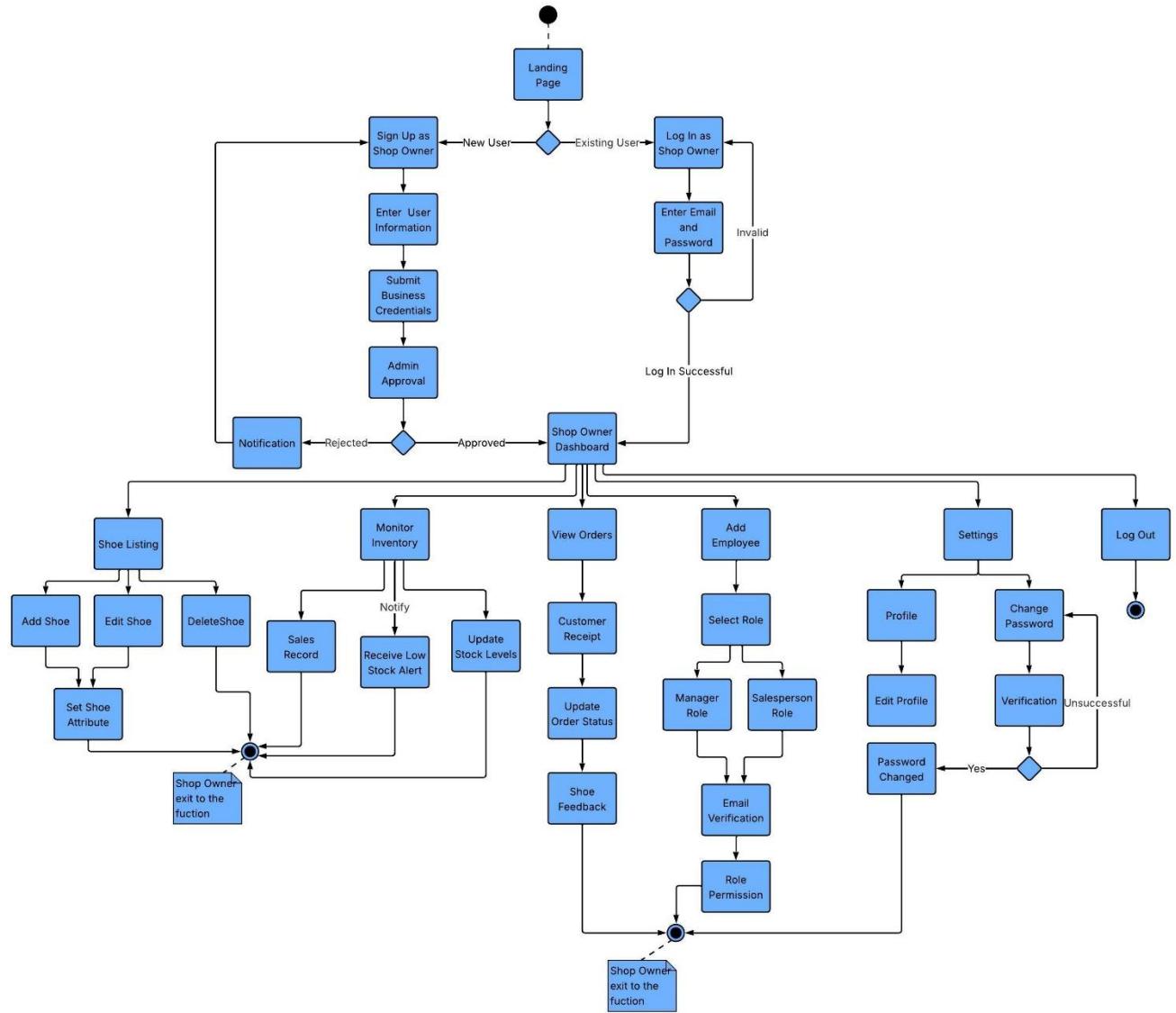


Figure 4.0 Activity Diagram (Shop Owner Side)

In figure 4, the shop owner may either register by providing personal information and business credentials for admin approval or log in using authorized credentials. After approval or successful login, the shop owner is able to enter the dashboard. In the dashboard, the shop owner can place products by adding,

editing, or deleting shoes, whereby one can even detail a particular product specification. It supports inventory monitoring, sales records, low stock notifications are received, and update stock levels. Also, the shop owner can control customer orders, view order details, issue receipts, update order status, and view customer feedback. The shop owner can also manage employees by assigning them roles such as manager or salesperson, with permission attached to roles through email verification. In addition, account settings enable the shop owner to edit profile information and reset passwords but only on verification. The activity closes once the shop owner logs-out or leaves the function.

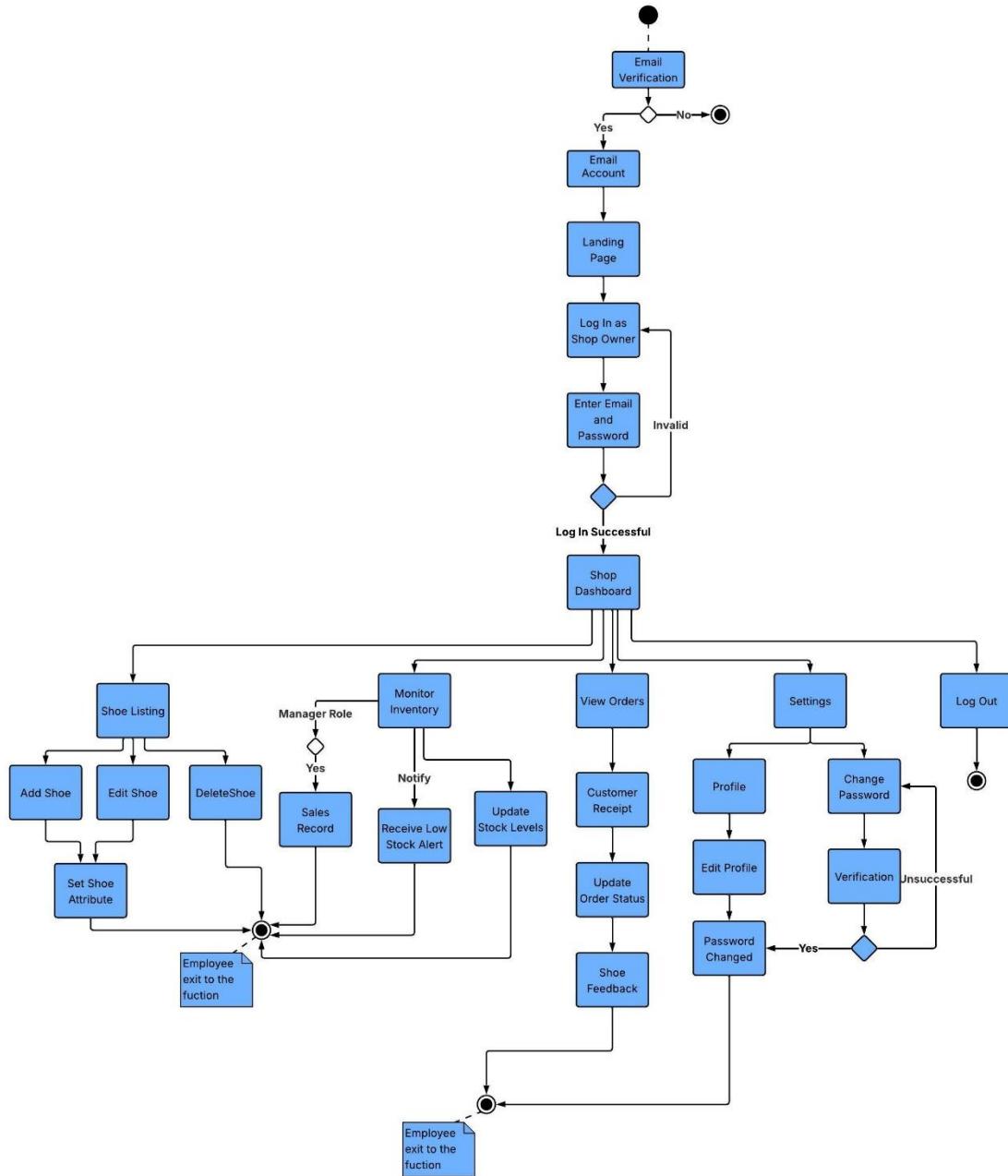


Figure 5.0 Activity Diagram (Employee Side)

In Figure 5, the employee begins with email verification. Once they have been verified, they have to log in using their credentials. After logging in, it directed to the shop dashboard. The employees can control shoe listings, adding, editing, or deleting shoes. Employees can also monitor stock, have alerts on low stock and

update stock levels. Employees can view and update customer orders, accessing customer receipts and respond to customer feedback. In settings, they may update their profile and change their password after verification. However, only employees having a manager role can access the sales record. The session is ended when the employee signs out or exits the function.

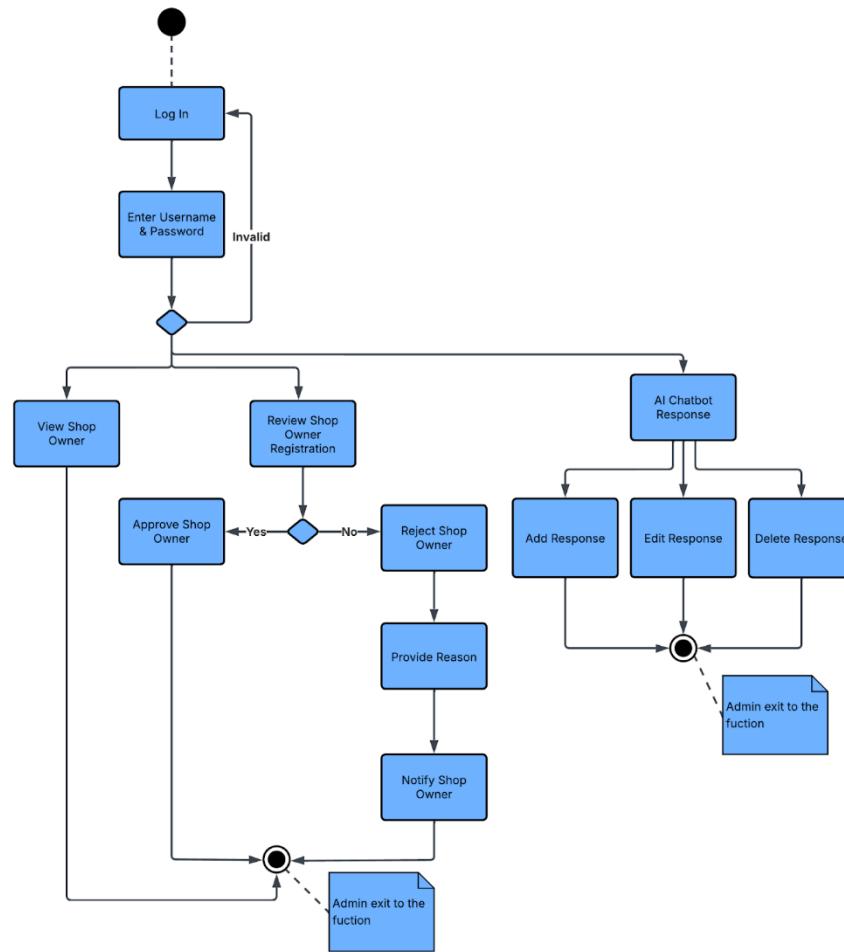


Figure 6.0 Activity Diagram (Admin Side)

In figure 6, to log into the admin side within the developed system, the admin first login using credentials. The admin was made to open up system functionalities such as viewing the registered shop owners and viewing the shop owners'

registration requests from new ones. While going through the review, the admin evaluates the sent business credentials and documents. The administrator approves the shop owner then on the registration if it meets the appropriate requirements. Otherwise, the registration is denied, and a reason is given and a formal notification is given to the shop owner. The admin controls AI chatbot responses such as add, edit, or delete the predefined system answer to customer questions. The administrator can then move on to perform other tasks, or when done exiting the administrative interface he can exit the entire session.

Database Design

The database design contains the entity-relationship models and data dictionaries that were used to develop the system. Some of the major tables are Customer, Shoe, and Shop Table.

Entity Relationship Diagram

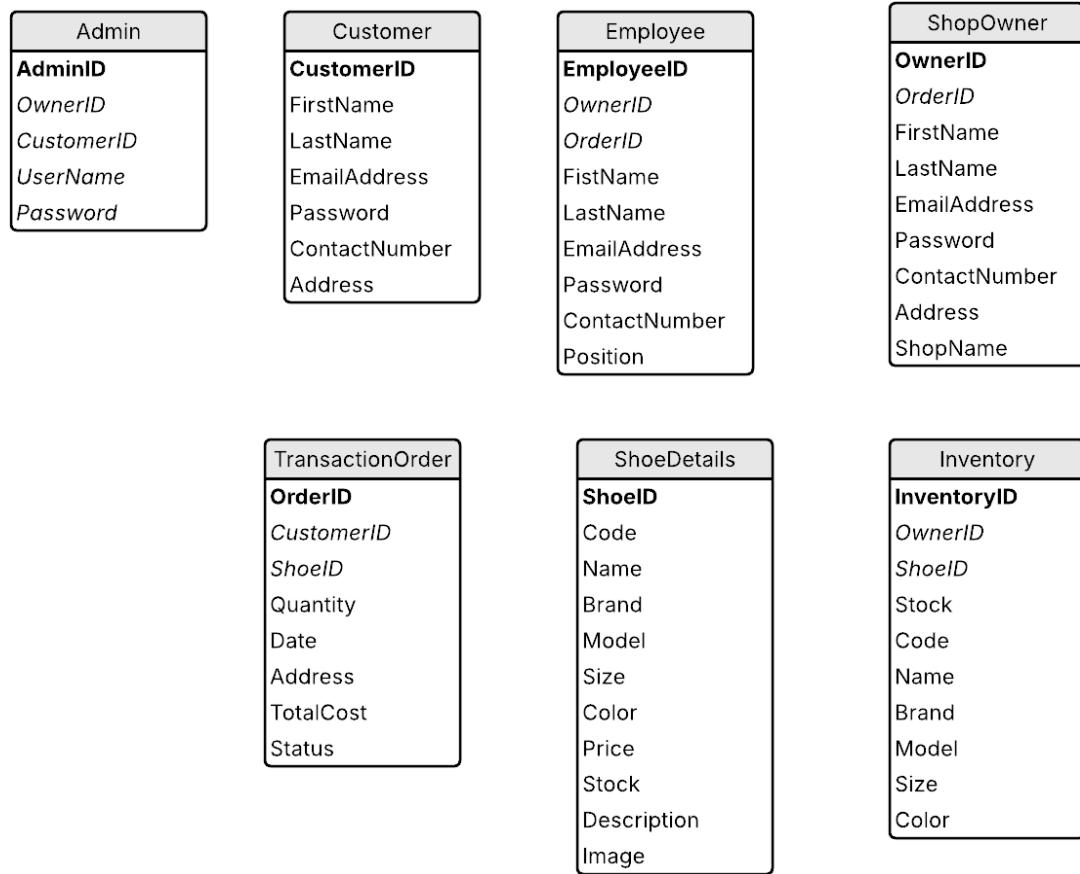


Figure 7.0 Entity Relationship Model

Figure 7 illustrates the NoSQL data model of SmartFit: A Developed Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance uses Firebase Firestore to fund shoe sale operations by controlling inventory, handling user-accounts, and transactions. The database is designed into various collections of Admins, Customers, ShopOwners,

Employees, ShoeDetails, Inventory, and TransactionOrder all of which contain documents containing the appropriate information.

The management of user accounts is carried out through the admins collection, which can work with the documents of both the Customers and ShopOwners collections. Customers collection relates to TransactionOrders collection where data such as purchase records of customers is stored. Equally, the ShopOwners collection refers to the Employees and Inventory collection to trace their store staff and product stocks.

Each transaction is saved to the TransactionOrders collection and can have transaction reference to the employee who placed the order and embedded or linked shoe records of the ShoeDetails collection. The availability and quantity of products are handled in the Inventory collection that keeps shop-specific data and references to products in ShoeDetails. This system enables every shop owner to have complete access to his inventory as well as enabling scalability and real-time data access and updating across the platform.

Data Dictionary

This explains what each data element means and how to use it correctly. It includes metadata about data elements, which helps define their scope and characteristics and the rules for how they should be used and applied.

Table 1.0 Customer Table

System Name: Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance				
Subject: Customer Table				
PK	FK	Field Name	Data Type	Description
Yes	No	customer_id	INT	Customer ID
No	No	fname	VARCHAR	First Name
No	No	lname	VARCHAR	Last Name
No	No	email_address	VARCHAR	Email Address
No	No	password	VARCHAR	Password
No	No	contact_number	INT	Contact Number
No	No	address	VARCHAR	Address

Table 1.0 displays the Customer table of the Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance. This table holds the data of the customer. This table's fields include: customer_id, fname, lname, email_address, password, contact_number, and address.

Table 2.0 Shoe Table

System Name: Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance				
Subject: Shoe Table				
PK	FK	Field Name	Data Type	Description
Yes	No	shoe_id	INT	Shoe ID
No	No	code	INT	Code
No	No	name	VARCHAR	Name
No	No	brand	VARCHAR	Brand
No	No	model	VARCHAR	Model
No	No	size	INT	Size
No	No	color	VARCHAR	Color
No	No	price	INT	Price
No	No	stock	INT	Stock
No	No	description	VARCHAR	Description
No	No	image	IMAGE	Image

Table 2.0 presents the Shoe table of the Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance. This table holds the data of the shoes. This table's fields include: shoe_id, code, name, brand, model, size, color, price, stock, description, and image.

Table 3.0 Shop Table

System Name: Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance				
Subject: Shop Table				
PK	FK	Field Name	Data Type	Description
Yes	No	shop_id	INT	Shop ID
No	Yes	order_id	INT	Order ID
No	No	fname	VARCHAR	First Name
No	No	lname	VARCHAR	Last Name
No	No	email_address	VARCHAR	Email Address
No	No	password	VARCHAR	Password
No	No	contact_number	INT	Contact Number
No	No	address	VARCHAR	Address
No	No	shop_name	VARCHAR	Shop Name

Table 3.0 shows the Shop table of the Smartfit: Augmented Reality-Integrated Footwear Retail and Customization Portal with AI-Powered Assistance. This table holds the data of the shop. This table's fields include: shop_id, order_id, fname, lname, email_address, password, contact_number, address, and shop_name.

Project Development

The system that was developed have used an Agile-based approach, focusing on flexibility, teamwork, and continuous deployment. The use of Agile allows the development team to deliver functionality in cycles; for example, the Augmented Reality capability, Artificial Intelligence (AI) assistance, and shoe

design customizing are divided into smaller, more manageable pieces and delivered in Sprints. The Agile methodology allows for continuous feedback from users and quick responses, so the system can be adjusted quickly to meet changing goals and user expectations. Even though there was a degree of planning done initially to define the project's scope, the focus remained on adaptability and responsiveness; therefore, the development and testing activities were performed concurrently so that the team could maximize efficiency while continuously refining the system via real-time feedback and incremental enhancements.

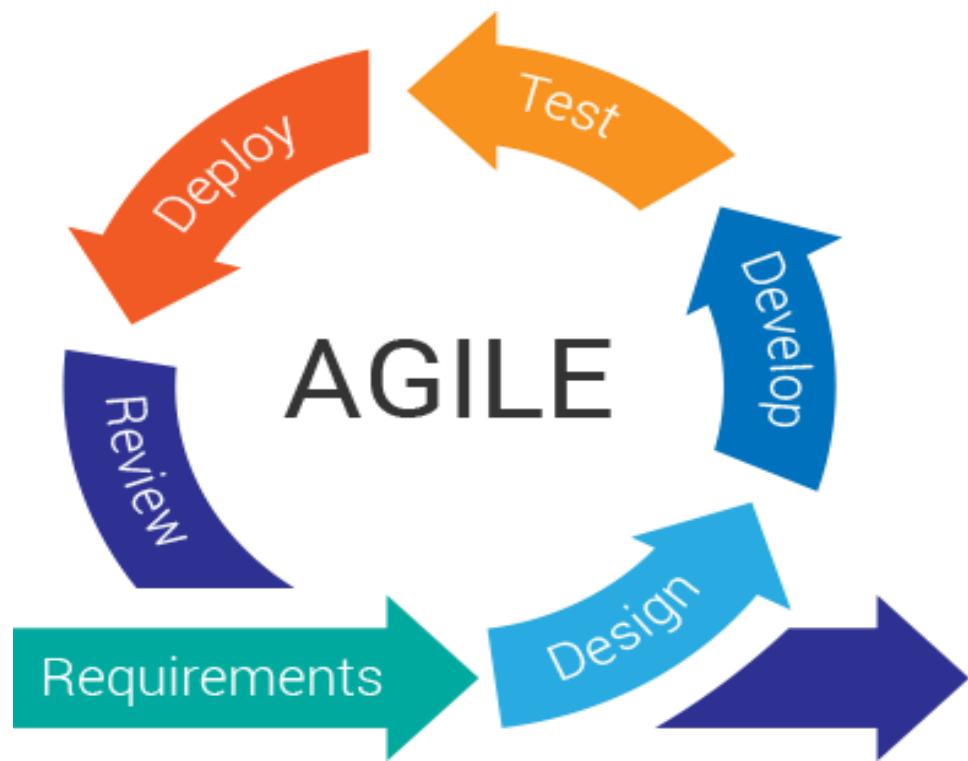


Figure 8.0 Agile Model

Reference: <https://medium.flutterdevs.com/what-is-agile-methodology-in-mobile-app-development-fa83ed6ac09>

Gantt Chart

This section enable activities to be clustered in phases to view the organization of the project better and to acquaint the project teams with the total timetable and workflow.

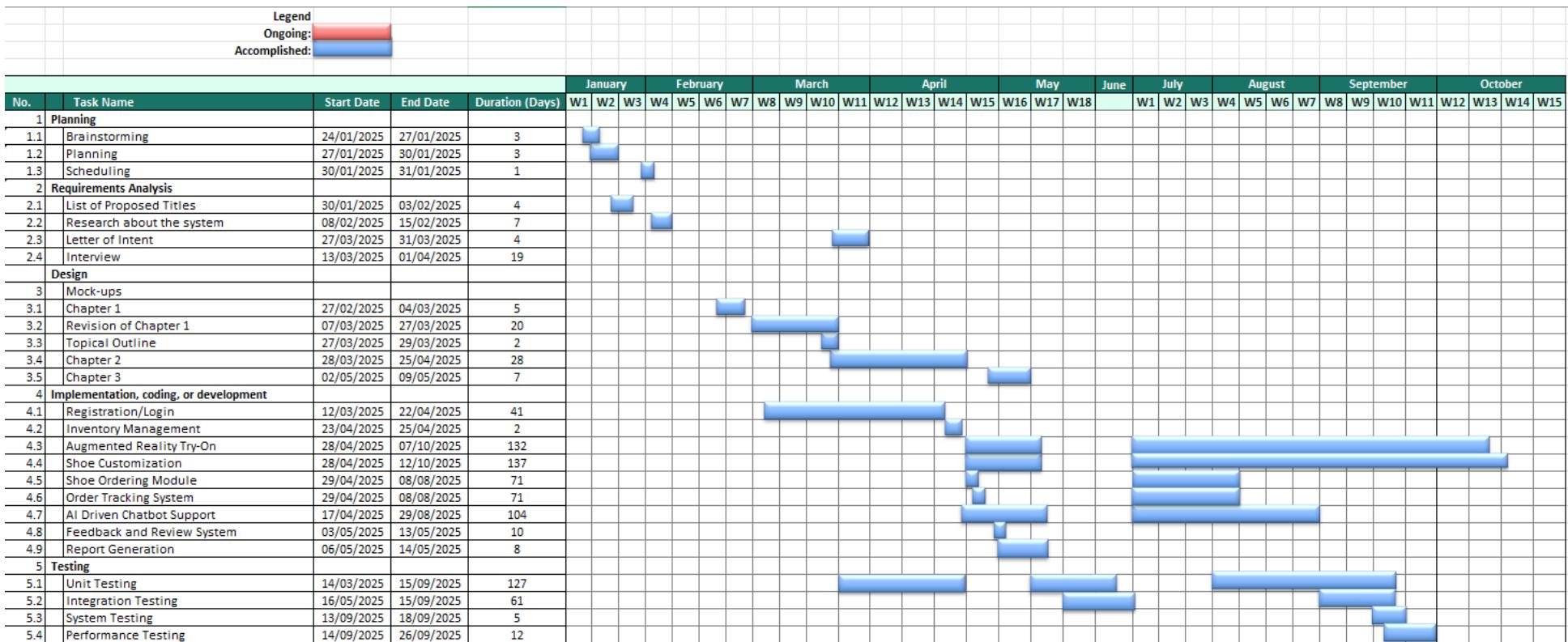


Figure 9.0 Gantt Chart

Figure 9.0 represents the Gantt chart that shows the timeline of activities, or phases listed, which consist of planning, requirements analysis, design, development, and testing. The system development started on January 27, 2025, to October 20, 2025. The proponents started the development with brainstorming, deciding which title or topic to propose based on some circumstances. When the final title was approved, the proponents started planning the project proposal and scheduling the development process of the system. In gathering data, they started preparing a letter of intent addressing the company's approval of their research, followed by conducting an interview with the company. After gathering the data needed, the proponents started developing the system. They began initially working with the documents so that they could plan and organize the process they followed and serve as a basis after deciding what to do in developing the web and application. After completing the whole system, the proponents conducted a series of testing procedures to ensure that the system is working properly starting with the unit testing, followed by the integration testing, then the system testing, and lastly, the performance testing.