Kingdom of Saudi Arabia
Ministry of Education
King Faisal University
College of Computer Sciences & Information Technology

Razzah

(Traditional Saudi Males' Clothing System)

Prepared by

Name	Student ID
Deema Algreeshah	219009773
Sara Aljaafar	219040101
Noura Almosined	219010636
Latifah Alhatlan	219029556

Supervised by

Dr. Majed Al Saeed

Section 61

February, 2023

Table of Contents

Table of Figures

1. Introduction

Nowadays, people suffer from getting some of their needs manually, so the requests for online systems have increased. Therefore, the availability of an online shopping system turned out to be the best way to buy things such as food, skin care products, and clothes so people can have them from any place in the world. Although there are various applications for the mentioned categories, it rarely provides choices for male, especially for clothes like Thobes, which is the traditional wear for male in Saudi Arabia. Hence, Razzah, a Traditional Saudi Males' Clothing System, will be developed to satisfy customer needs. The system will serve all age groups, with different options and sizes. So, the purchase process becomes comfortable for customers and their requirements.

1.1 Purpose

The purpose of this document is to present a detailed description of Traditional Saudi Males' Clothing System. Therefore, it will illustrate the system's functions, interfaces, requirements, features, and overall descriptions as well as how the system reacts to external and internal events. Thus, both the system developer and the stakeholders will benefit from this document.

1.2 Project Scope

This software system will be a Razzah Online System for Saudi male customers.

- The system is designed to satisfy male needs in Thobe, Bisht, and Shemagh clothing. Also, it provides various sizes and other options.
- The system is designed to allow the vendor to manage and communicate with male customers, so they can purchase Thobe clothes online.
- The system enables the customer to save the desired item for further purchase.
- The goal of the software is to give male customers the ability to determine when and from where they can order from the software.

2. Overall Description

2.1 Product Perspective

The user interface of the software is appropriate. Additionally, the system of the software make use of a centralized database that houses all of the consumer data.

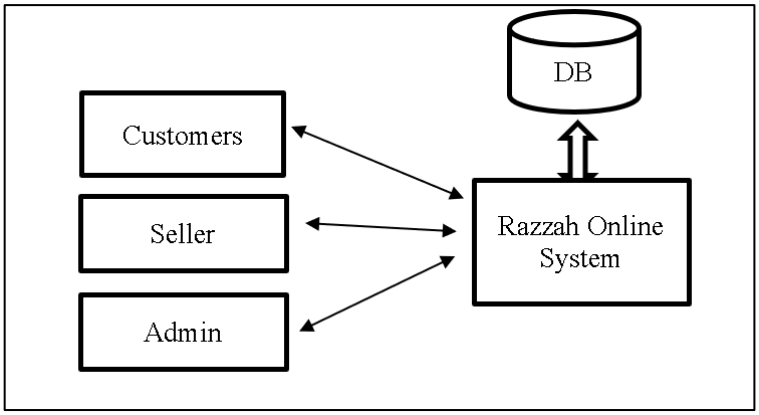

Figure 1 Product Perspective

2.2 System Framework

In our mobile application, we are going to use Flutter. It is an open-source framework. With a single codebase, frontend and full-stack developers may create an application's user interface (UI) on a variety of platforms using Flutter. When Flutter first debuted in 2018, it primarily assisted the building of mobile apps. Six platforms are now supported by Flutter for application development: IOS, Android, the web, Windows, MacOS, and Linux [1]. Our application is Cross-platform app development. With cross-platform app development, programmers may create applications for several platforms using just one programming language and one codebase. Cross-platform app development is faster and less expensive than native app development if you're releasing an application for numerous platforms. Additionally, this procedure enables developers to provide users with a more unified experience across platforms. Compared to developing native apps, this strategy may have disadvantages, such as restricted access to native device features. Flutter, on the other hand, includes characteristics that facilitate faster and more effective cross-platform app development.

2.3 Software tools

Software:

- Android Studio for building the app.
- Firebase for system's database.
- Uizard for the UI design of the app.

2.4 Product Features

The product features can be described as follows:

- **Product management:** manage the product in the system based on the features.
- **Shopping cart:** the customer can add/delete more than one product to the shopping cart until the customer is being sure to make the purchase.
- **Wishlist:** the customer can save items, so they can purchase faster when it becomes possible.
- **Payment:** the customer can make payment only on cash after reviewing the items in the shopping cart.
- **Account management:** allows users to create an account, log in, and log out, and reset password in case it is forgotten.

2.5 User Classes and Characteristics

Customers: can create an account and login. Also, customers can determine the specifications of the clothes, save favorite items, add/remove the items from the shopping cart, view the shopping cart, and place order.

Admin: having all rights to add /remove some specifications of the items and view reports. Also, admin can have the right to show users information.

Seller: the seller who provides some specifications of the items to be available in the shop to add them by the admin.

3. System Features

This section discusses the features and functional requirements of the system in detail.

3.1 Register

- New users shall be able to register to the system.
  1. The system shall allow the user to input personal information.
  2. The system shall identify the user via their email.
  3. The system shall create an account for the registered user if email wasn't used.
  4. The system shall display an error message if email is already used.

3.2 Login

- The user can log into the system.
  1. The system shall enable the user to enter email and password.
  2. The system shall direct the user to home page if the email/password are correct.
  3. The system shall display an error message if the email/password are wrong.
  4. The system shall allow the user to reset the password if it is forgotten.

3.3 View Items

- The user shall be able to browse system pages and view items.
  1. The system shall display all sections.
  2. The system shall move to the required section.
  3. The system shall display products' information.
  4. The system shall support the user with favorite page to save needed items.

3.4 View/ Update Cart

- The user shall be able to add and view items of the cart.
  1. The system shall display the details of added items in the cart.
  2. The system shall show the total price of cart's items.
  3. The system shall enable the user to update the cart either by adding or deleting items.
  4. The system shall allow the user to place order.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

1. The system shall respond to requests no longer than 10 sec.
2. The system shall be able to handle multiple customers logged in at the same time.
3. The system shall be able to add items to the shopping cart in less than 2 second.
4. The system shall be able to store several customer records.
5. The system shall be able to view the item and its information for customers.

4.2 Security Requirements

1. The system shall detect consecutive failed login attempts.
2. The system shall be able to encrypt all sensitive information.
3. The system shall keep the user's privacy and information secure.

4.3 Software Quality Attributes

1. **Availability:** The system shall have an availability of 99.99%.
2. **Usability:** system shall be easy to use, system shall write an error message in a way that customers understand it.
3. **Maintainability:** The system shall be easily updatable for fixes and patches and create files to record all changes, updated, fixes that happen on the system.
4. **Testability:** system shall be able to run under any circumstance.
5. **Efficiency:** The system shall validate user id existence if it will prevent it from registering again.

5 Overall System Design

5.1 System Architecture

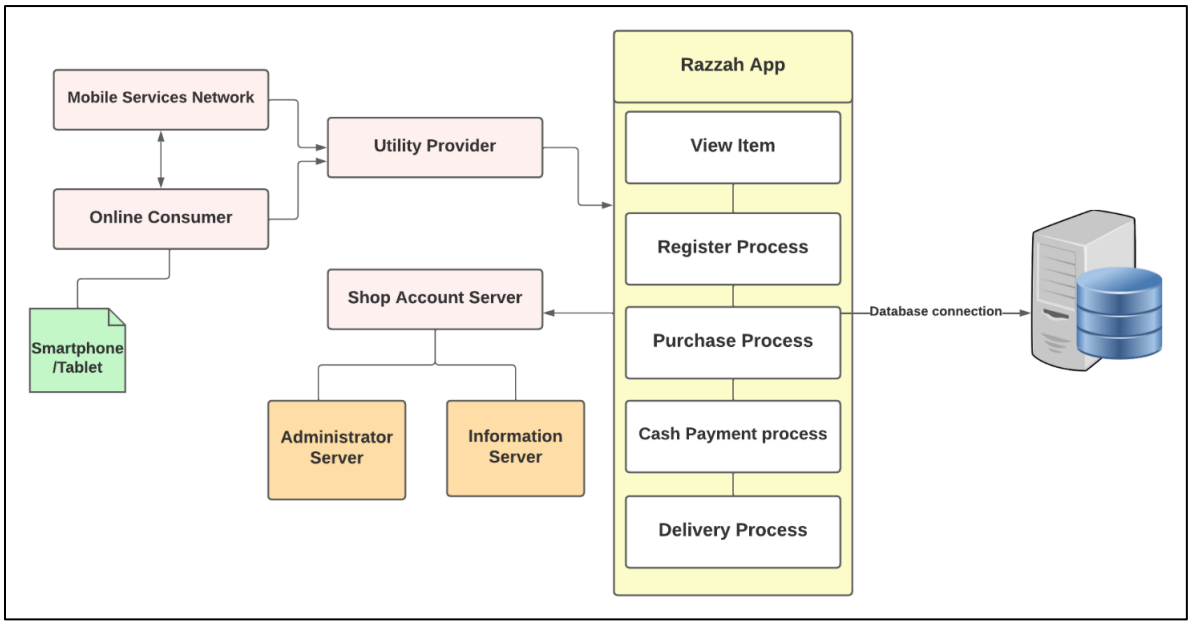

Figure 2 System Architecture

Razzah App aims to facilitate virtual shopping on mobile devices like tablets and smartphones. Figure 2 illustrates the design of system architecture. The device needs network services in order to connect to an application. The main functionalities of the app are register process, viewing item, purchase process, cash payment process, and delivery process. Moreover, it connects to firebase database to store and retrieve the data to provide dynamic pages and user authentication.

5.2 MVC Pattern

One of the most popular and widely accepted web development frameworks for building scalable and flexible projects is Model-View-Controller (MVC) [2]. An application is divided into three primary logical components using the MVC architectural pattern: the model, the view, and the controller. Each of these parts is designed to handle particular application development facets. Model, all the user's data-related logic is represented by the Model component. This could be any other data related to business logic or the data that is being passed between the View and Controller components. A Customer object, for instance, can obtain customer information from a database, edit it, and then update the data back into the database or utilize it to render data. View, the application's entire UI functionality is implemented in the View component. For instance, the Customer view will have every UI element that the end user interacts with, such as text boxes, dropdown menus, etc. Controller, in order to handle the business logic and incoming requests, manipulate data using the Model component, and interact with Views to generate the output, controllers serve as an interface between the Model and View components. Using the Customer Model to update the database, the Customer controller, for instance, will manage all interactions and inputs from the Customer View. The Customer data will be seen using the same controller. In this document, we apply MVC pattern for cart as in Figure 3, product as in Figure 4, user authentication as in Figure 5, and order as in Figure 6. The steps for all are common as follows:

1. The button is clicked.
2. A low-level UI event will be produced. The component's event handler calls a method on the presentation model instance (often injected into the view) that manages View and Controller interactions.
3. The model receives an application message from view to pass it into controller to execute the command.
4. Before passing the model to other result handlers, the command's result handler may alter it or store it.
5. The view is updated typically via binding to presentation model properties.

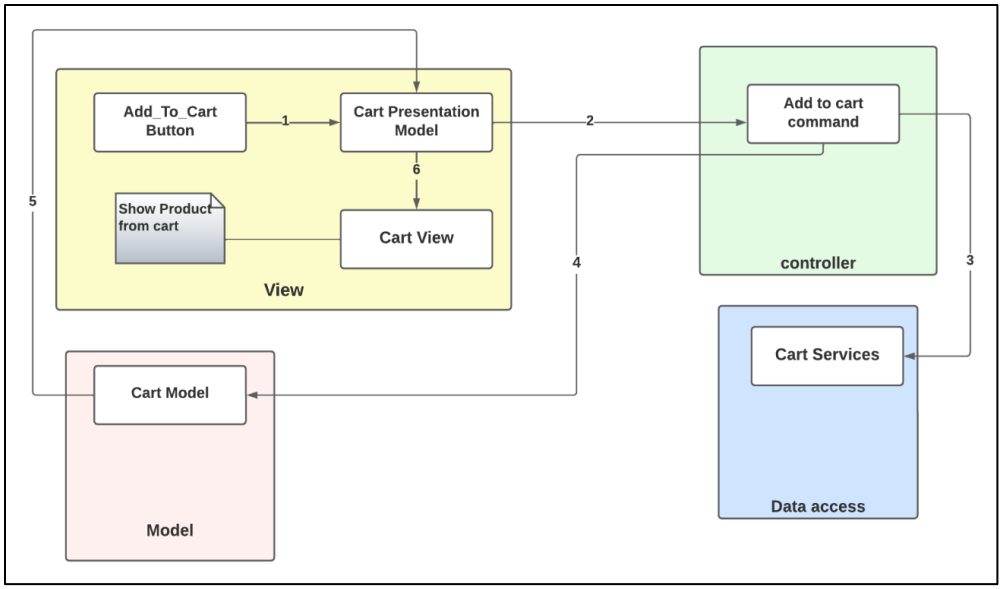

Figure 3. MVC Pattern for Cart

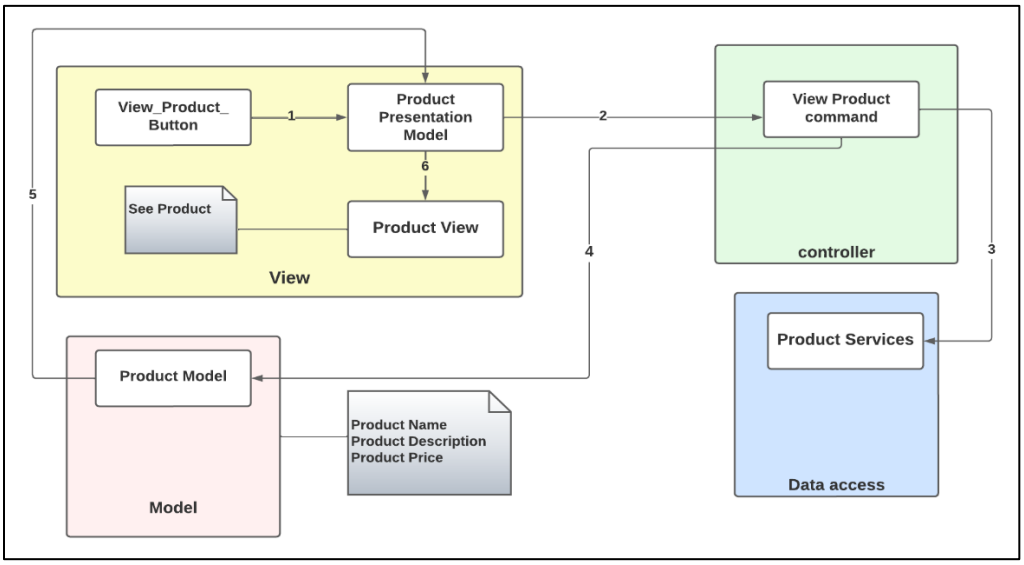

Figure 4. MVC Pattern for Product

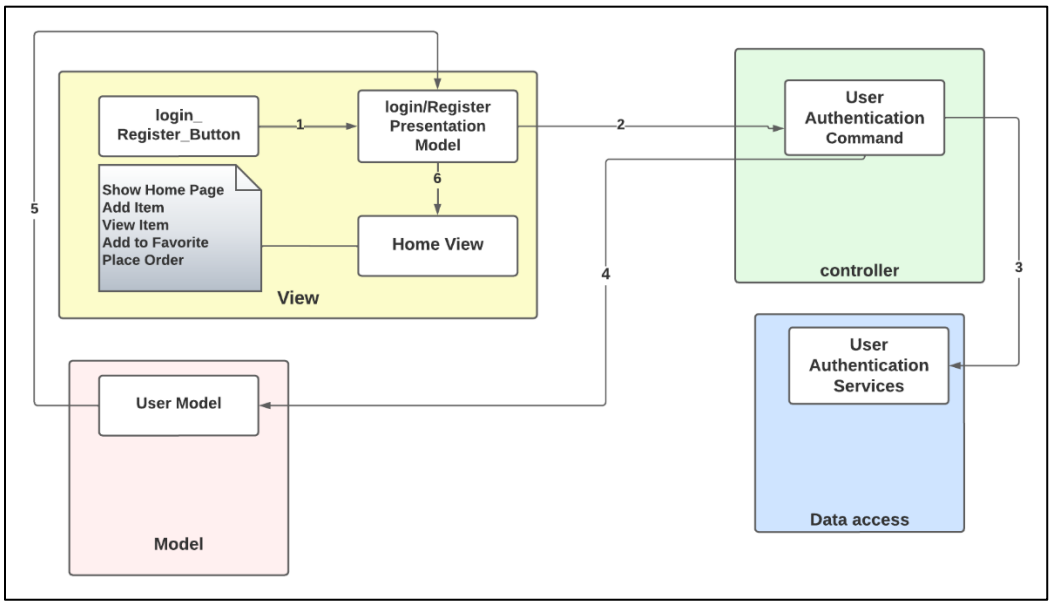

Figure 5. MVC Pattern for User Authentication

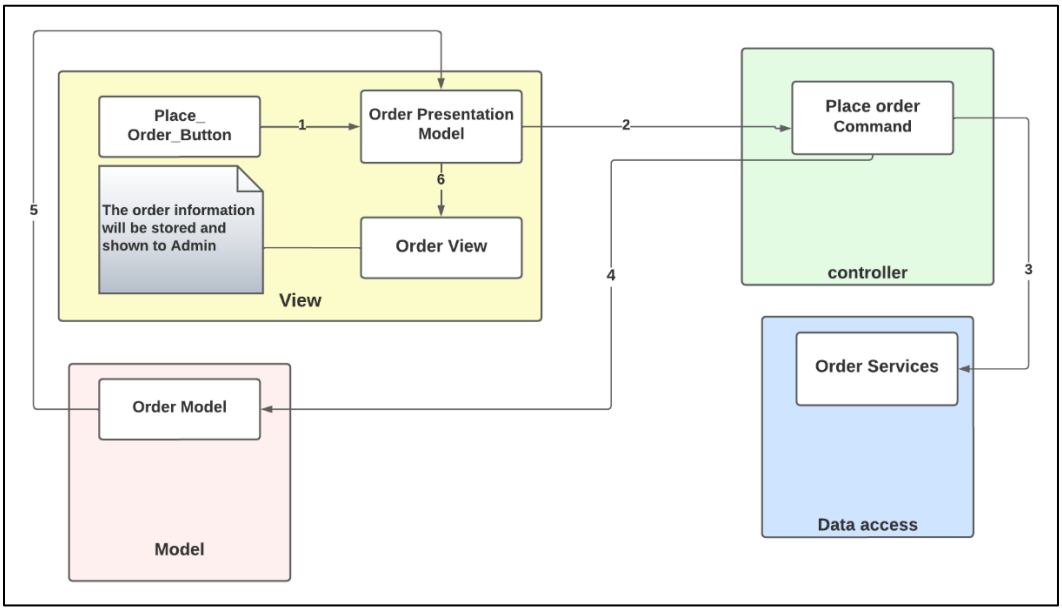

Figure 6. MVC pattern for Order

5.3 Software Design

5.3.1 Class Diagram

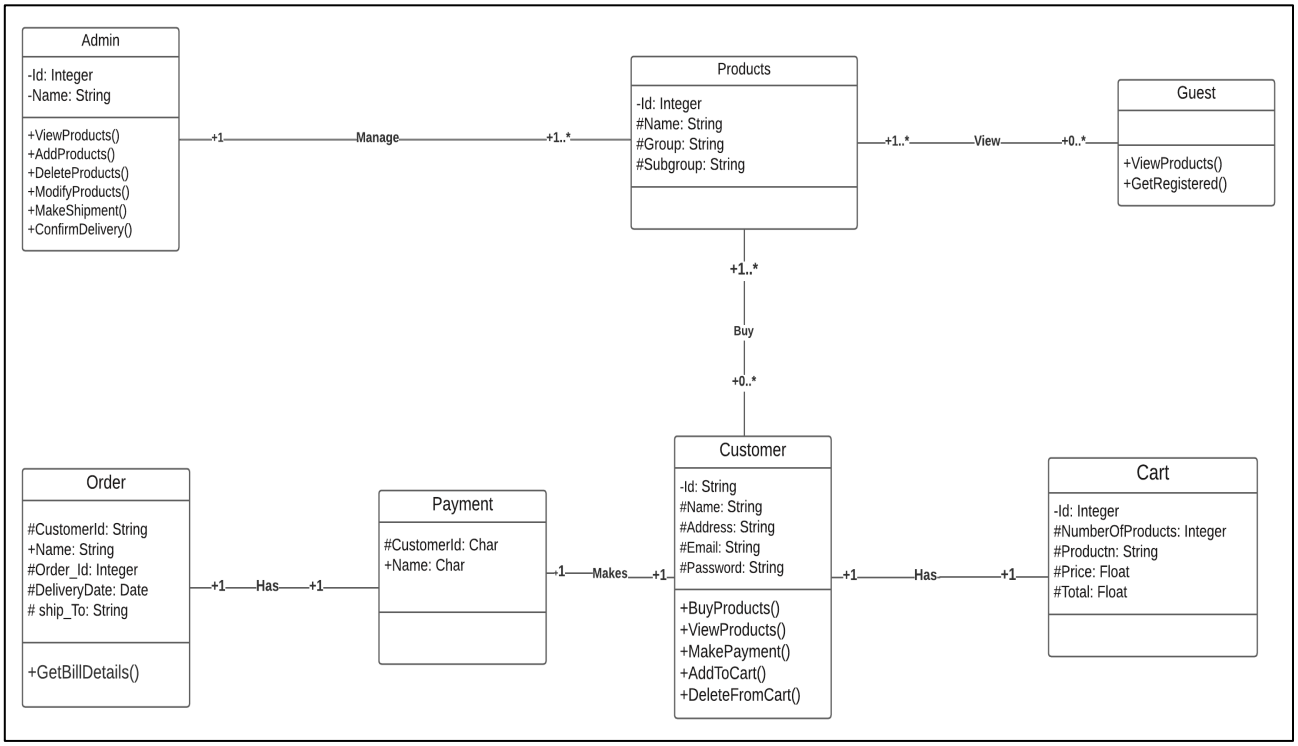

Figure 7 Class Diagram

The Razzah App could serve more than one customer and every customer must have a name, an Id, an email, and an address. Customers can register in the system using their Id and Password and then used it to log in to the system. The customer can order more than one order and each order has a number, Total price which it gets from payment once the order is confirmed, delivery Date, and address which the order will be shipped to it. Each order will have products and each product will have an ID, a name, group. The group, it can be Thob, Bisht, and Shemagh. The customers should pay with cash only.

5.3.2 Use Case Diagram

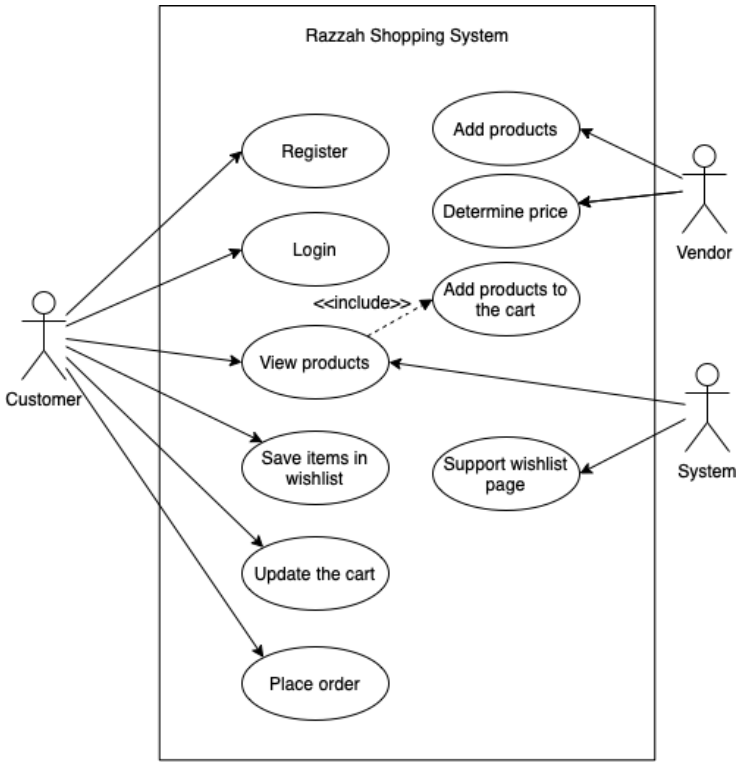

Figure 8 Use Case Diagram

The Razzah Shopping System provides functionality for stakeholders. The system will allow a customer to create an account by registering and login into the system. Both customer and system can view products then the customer can update his/her cart when he/she wants and confirm the order when he/she is finished. The system allows vendors to add and determine the price of the products. Also, it provides Wishlist page for customers. So, customers then can save the desired items in them.

5.3.3 Sequence Diagram

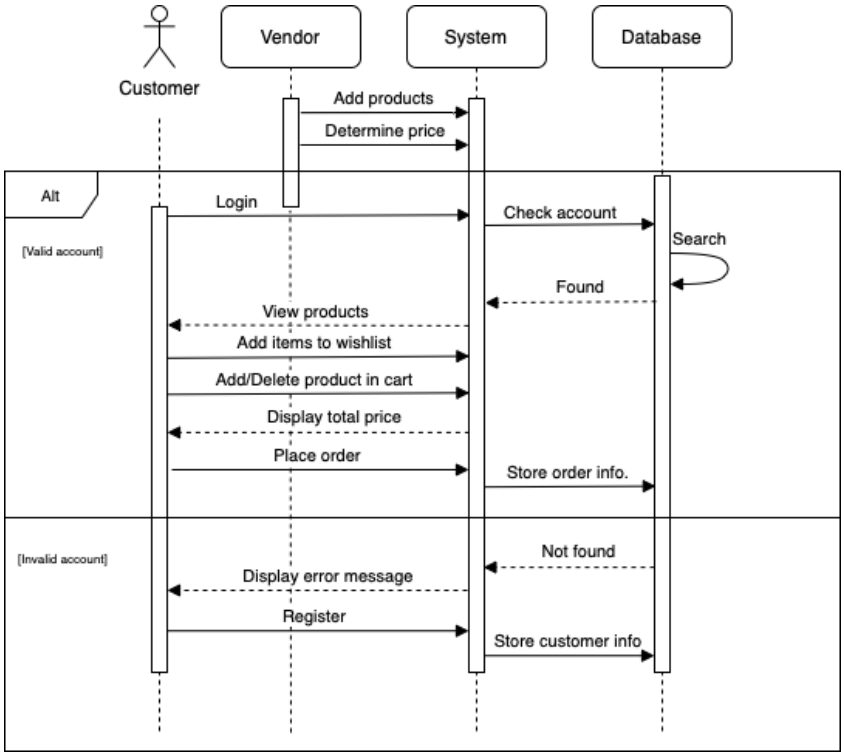

Figure 9 Sequence Diagram

The system starts when the vendor adds products to it and determines their prices. Then, the customer logins to the system, the system will check whether the account is found in the database or not by searching inside the database, it will view products to the customer if the account was found. Moreover, the customer can add the items that they liked to their Wishlist also add and delete items to the cart and place the order once they are done. Afterward, the system will store the order information into its database. If the account was invalid in login, the system displays an error message to the customer to register to the system then the system will store the customer's information in the database.

5.3.4 Activity Diagram

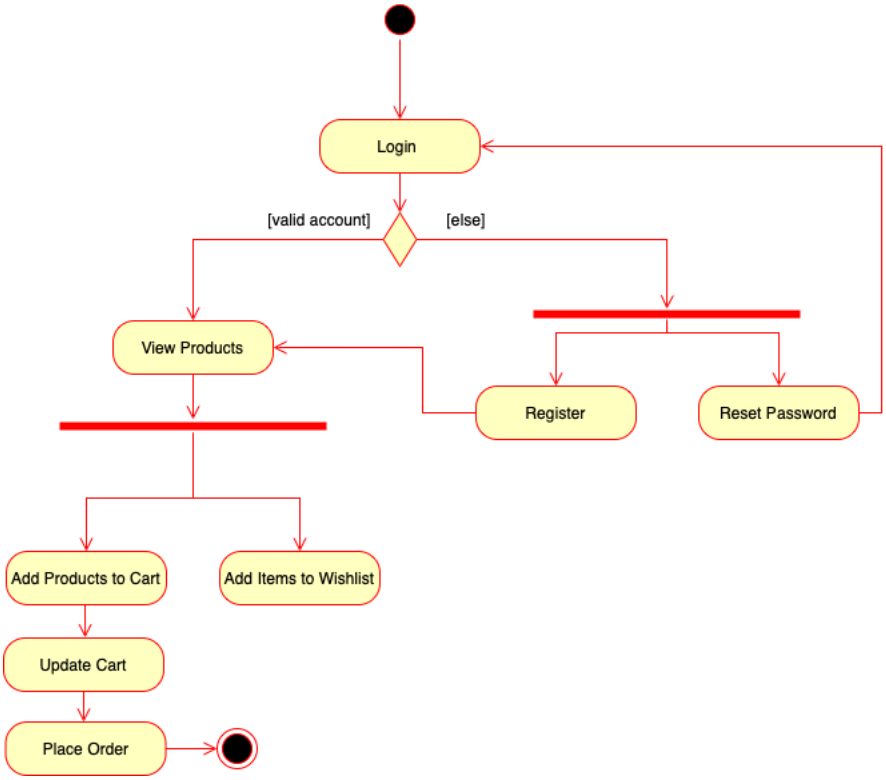

Figure 10 Activity Diagram

This diagram begins when the customer logs onto the Razzah Shopping System and enters his/her credentials. The system verifies whether the email and password are valid or not. If they were invalid, the customer either resets the password or registers with his/her information and then directed to the items in the main page. Whereas if the email and password are valid, the customer can view the products and add them either to the cart or the Wishlist. In case the customer adds products to the cart, he/she can update the cart by adding more products or deleting unwanted products then placing the order after his/her shopping is done.

5.4 Database Design

Razzah shopping system is based on a shopping database that helps to keep system information safe. This system has 3 entities: customer, product, and shopping cart as shown in the Figure 11.

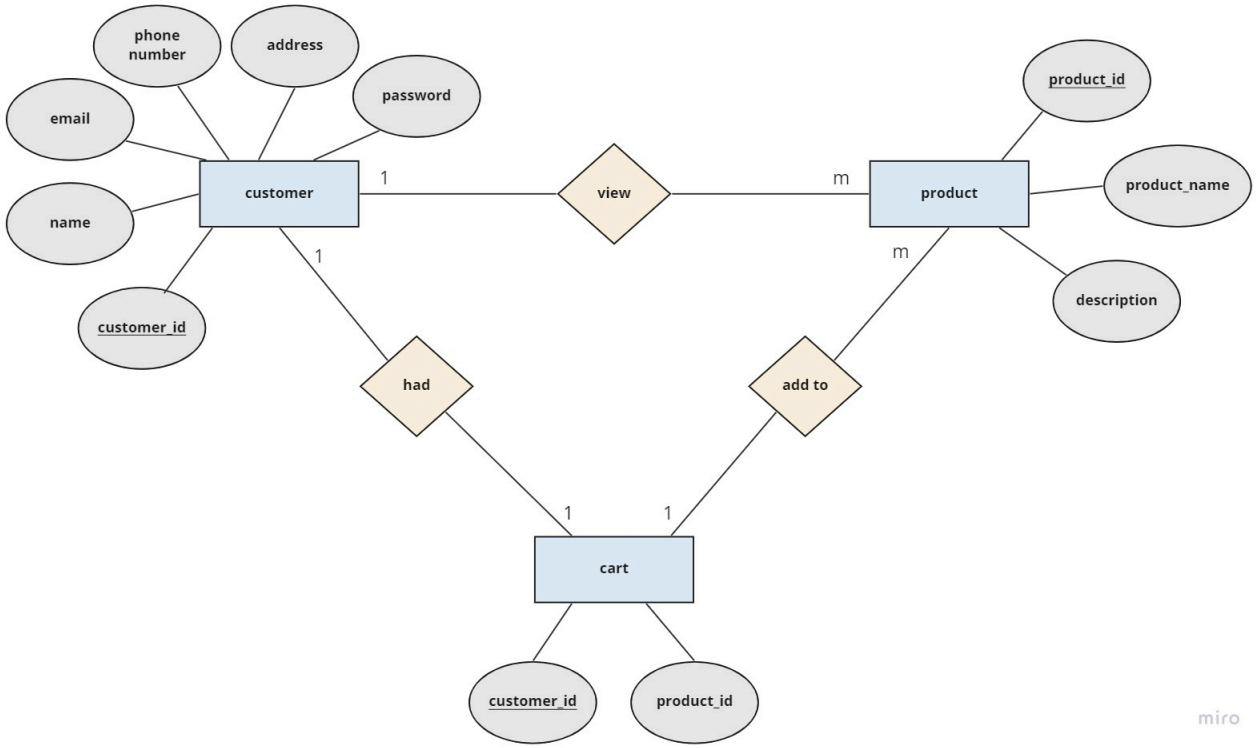

Figure 11 Entity Relation Diagram

5.5 Graphical User Interface Design

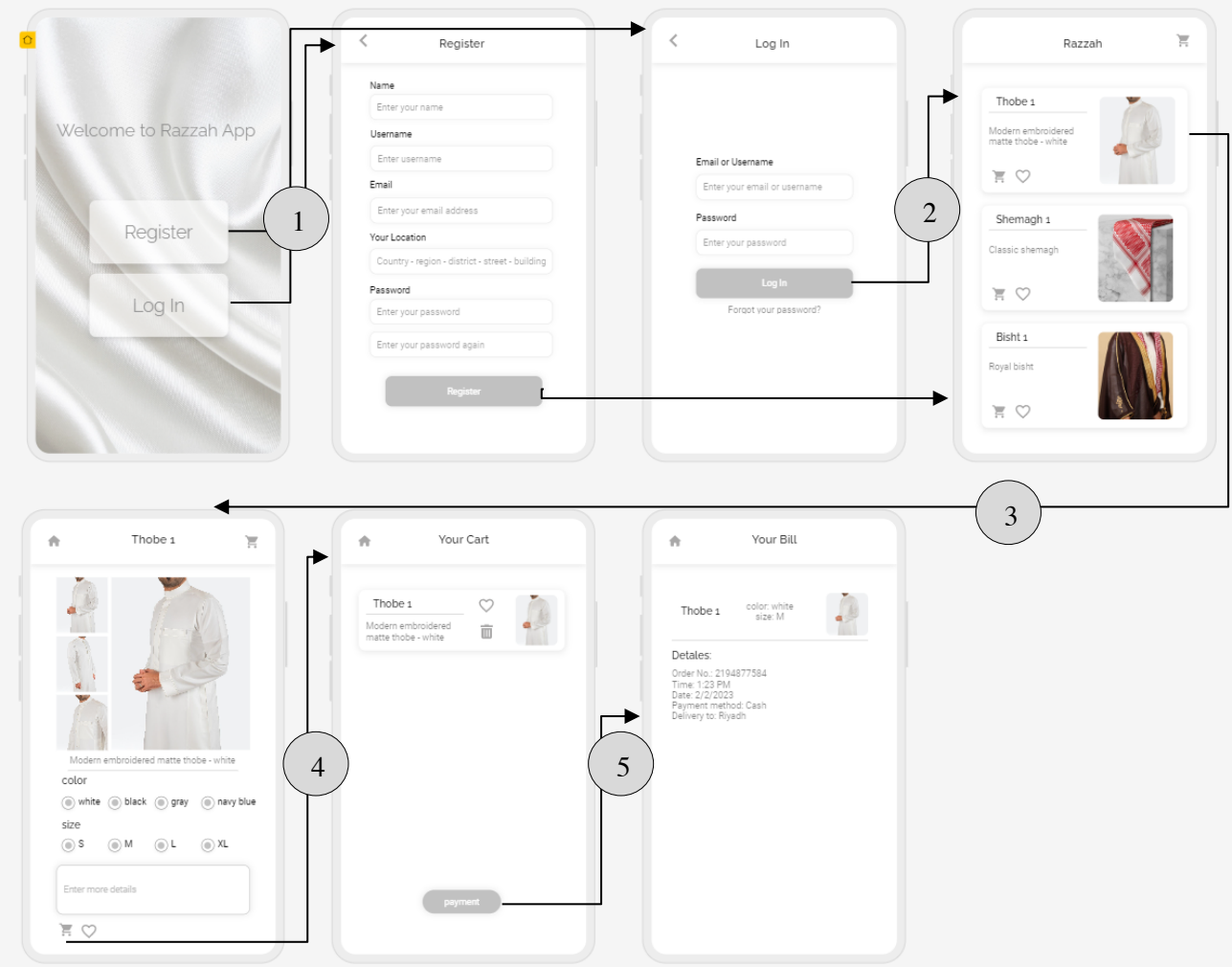

Figure 12 Graphical User Interface Design

6 Implementation

In this section, we show the interfaces of the program that we have created and a simplified explanation of how it works.

Figure 13. Splash Screen

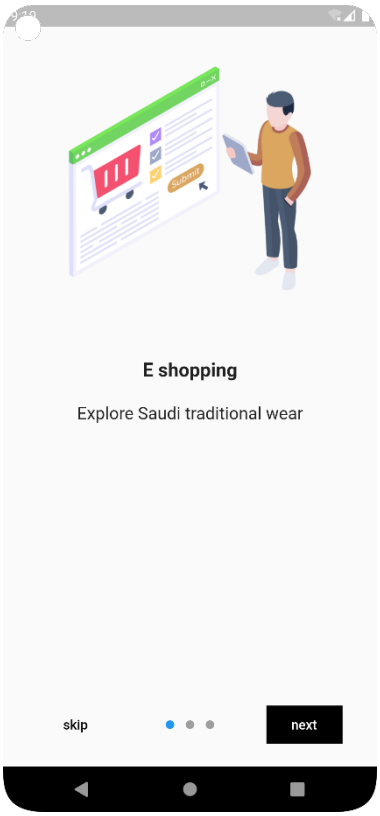

Figure 14. Onboarding-1

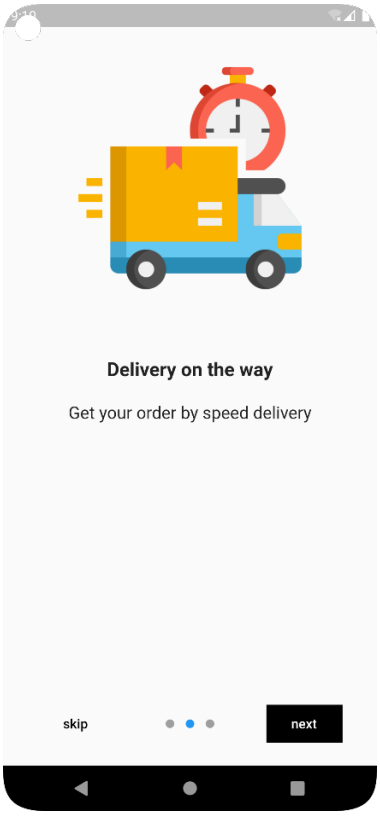

Figure 15 Onboarding-2

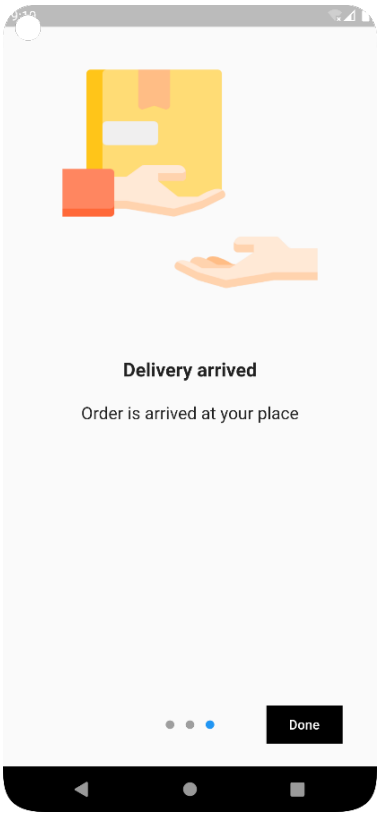

Figure 16 Onboarding-3

The application starts with these interfaces that define its services.

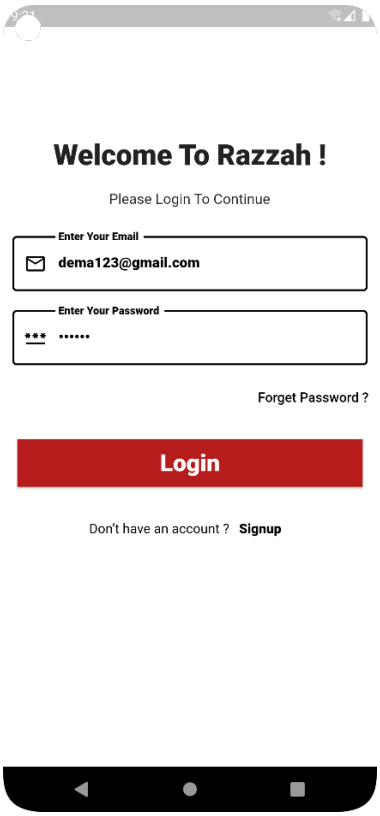

Figure 17 Login Page

Then, you will be taken to this page, which is the login page. On this page, the user enters their data if they have an account for successful authentication. In case the user forgets the password, they press on “Forgot Password” to go to that page. Whereas if the user does not have an account, they click on “Signup” to go to the new account registration page.

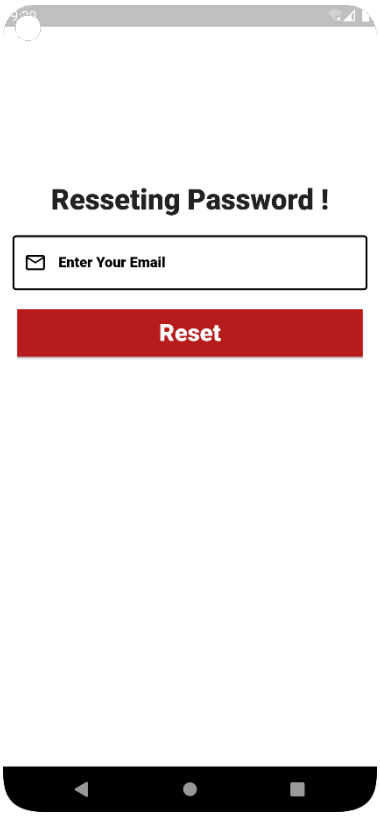

Figure 18 Resetting Password Page

On this page, the user is verified to reset the password by sending a verification code to the email.

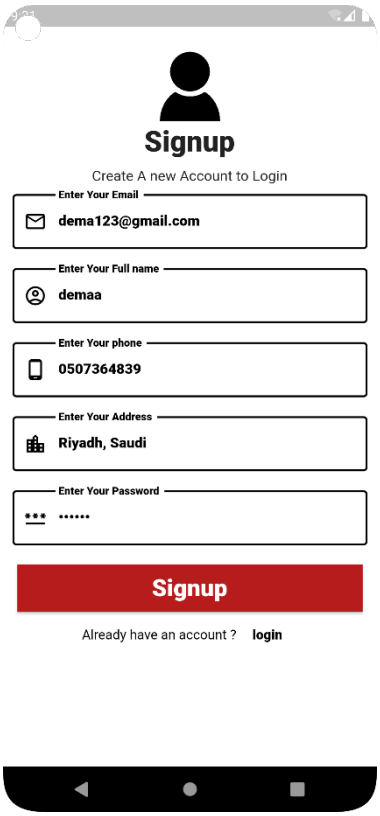

Figure 19 Signup Page

The user creates a new account on this page by entering their personal information, which is email, full name, mobile number, address, and password followed by clicking on "Signup" to go to the products page. If the user had an account, they press “Login” to go to that page.

Figure 20 Home Page (Thobe Section)

This is the home page, as you can see at the top of the page, there are three buttons that navigate to the favorite list, shopping cart, and logout. Also, it contains all the products, divided into three categories' pages: a Thobe, a Bisht, and a Shemagh. As shown in Figures 20, 21, and 22, respectively. Where all products are displayed, and each product is displayed with its name, price, and pictures. You can add the product directly to the shopping cart, or add it to your favorite list, or even delete it from the cart.

Figure 21 Home Page (Bisht Section)

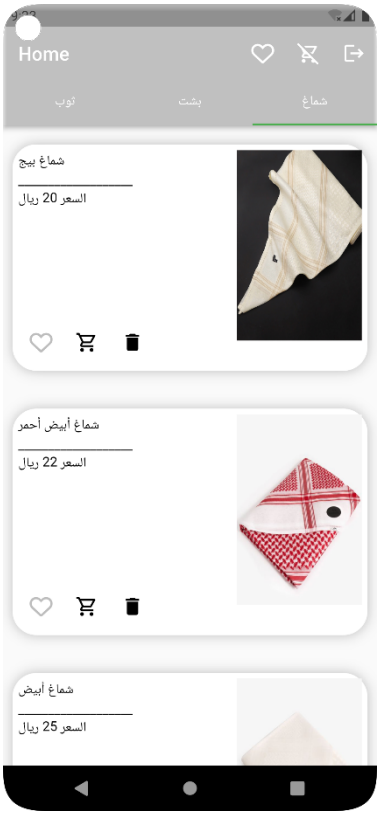

Figure 22 Home Page (Shemagh Section)

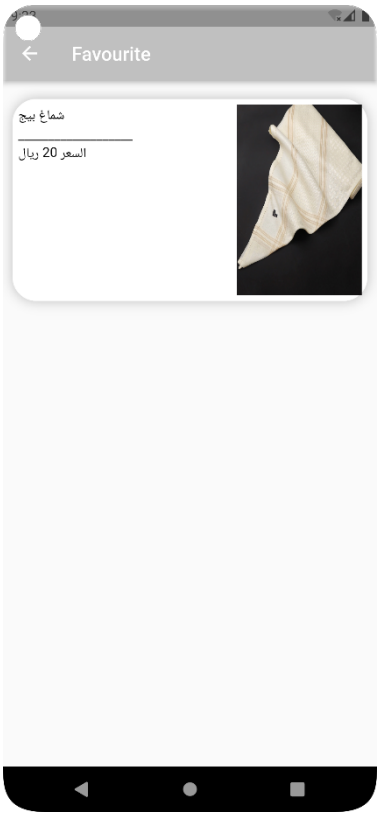

Figure 23 Favorite Page

This is the list of favorite items that can be accessed from the home page.

Figure 24 Shopping Cart Page

This is the shopping cart that can be accessed from the home page, and several products can be added and removed whenever you want.

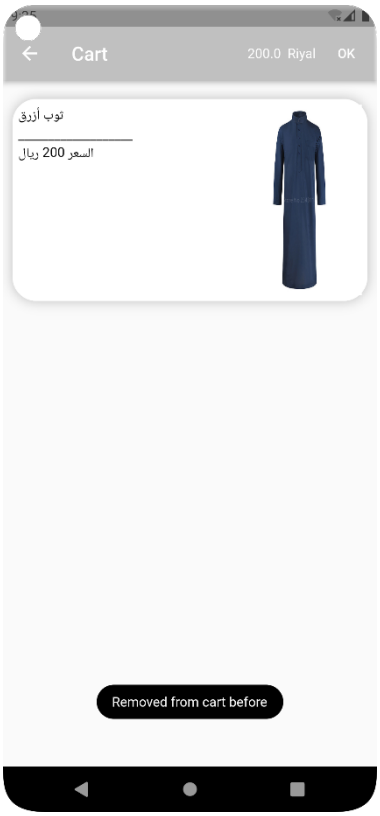

Figure 25 Shopping Cart Page (with product removing)

When a product is removed, a message "Removed from cart before" appears. It also shows the total price of the products at the top, and if you click on “OK”, the system confirms the order.

7 Testing

Software testing is the process of assessing and confirming that a software application or product performs as intended. Testing has advantages such as bug prevention, lower development costs, and better performance. The tables 1,2,3, and 4 test main functionality of Razzah App.

Table 1 Test Case # 1

Project Name: Razzah					
Test Case ID: Login 1-1			**Test Designed by:** All the member of the project		
Test Priority: High			**Test Designed date:**		
Module Name: Login Screen			**Test Executed by:** All the member of the project		
Test Title: Verify login with valid username and password			**Test Execution date:**		
Description: Test the login page					
Pre-conditions: User has valid username and password					
Dependencies:					
Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigate to login page				
2	Provide valid email	Dema123@gmail.com	Login successfully		
3	Provide valid password	demmaa			
4	Click on login button		User should be able to login	User is navigated to home page	Pass
Post-conditions: User is validated with database and successfully login to account. The account session details are logged in database.					

Table 2 Test Case # 2

Project Name: Razzah	
Test Case ID: Sign up	**Test Designed by:** All the member of the project
Test Priority: High	**Test Designed date:**
Module Name: Sign up Screen	**Test Executed by:** All the member of the project
Test Title: Verify sign up with name, username, password, email, and location.	**Test Execution date:**
Description: Test the sign up page	
Pre-conditions: User does not have account in database means username not duplicated	
Dependencies:	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to sign up page				
2	Provide full name	Ali mohammed			
3	Provide phone number	05039033			
4	Provide password	Bref23m			
5	Provide email	alimmg@gmail.com			
6	Provide address	Alahsa-9189			
12	Click on sign up button		User should have a new account in this application	User is registered successfully then navigate to home page	Pass

Post-conditions: Account information is recorded in the database.

Table 3 Test Case # 3

Project Name: Razzah	
Test Case ID: Add to cart 3-1	**Test Case ID:** Add to cart 3-1
Test Priority: High	**Test Priority:** High
Module Name: Add to cart	**Module Name:** Add to cart
Test Title: Verify adding to cart	**Test Title:** Verify adding to cart
Description: Test the button (add to cart)	
Pre-conditions:	
Dependencies:	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Start with home page				
2	Click on add to cart button		Add item to cart	Add item to cart	Pass

Post-conditions: Record the order details in the database as incomplete.

Table 4 Test Case # 4

Project Name: Razzah	
Test Case ID: Place order 4-1	**Test Case ID:** Place order 4-1
Test Priority: High	**Test Priority:** High
Module Name: Checkout Screen	**Module Name:** Checkout Screen
Test Title: Verify checkout	**Test Title:** Verify checkout
Description: test checkout page	
Pre-conditions: User most have items in a cart.	
Dependencies:	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Start with home page				
2	Click on cart button		Move to cart page	Opened cart page	Pass
3	Click on place order button		Store order information	Store order information	Pass

Post-conditions: Record the order details in the database as complete.

Table 5 Test Case # 5

Project Name: Razzah	
Test Case ID: Favorite 5-1	**Test Designed by:** All the member of the project
Test Priority: High	**Test Designed date:**
Module Name: favorite Screen	**Test Executed by:** All the member of the project
Test Title: Verify favorite	**Test Execution date:**
Description: test favorite page	
Pre-conditions:	
Dependencies:	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Start with home page				
2	Click on favorite button		Display all favorite items	Display all favorite items	Pass

Post-conditions: Record the order details in the database as complete.

8 References

[1] What is Flutter? - Flutter App Explained - AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/flutter/

[2] TutorialsPoint. (2021, July 28). MVC Framework - Introduction. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm