QAU Huts Meal Ordering System

Final Year Project

Supervisor

Dr. Umer Rashid

Submitted By

Sumbal Batool

MSC Computer Science

Department of Computer Science Quaid-I-Azam

University, Islamabad

A report submitted to the Department of Computer Science, Quaid-I-Azam University, Islamabad. As a partial fulfilment of the requirements for the award of the Master's degree in Computer Science.

Abstract

Qau Huts Meal Ordering System is designed primarily for use in the food delivery. This system allows customers to quickly and easily manage an online menu, also they can browse and place orders with just few clicks. The purpose of my system is that it greatly simplifies the ordering process for customers. When the customer visits the ordering Application, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After selecting, the item is then added to their order. The customer can review the details of their order at any time before checking out. This provides instant visual confirmation of what was selected and ensures that the desired items are placed in order queue. Simply this system is user friendly and easy to use without taking much time. Themain problem of the staff and students is that they have to go to Huts physically in order to get their desired food which is quite hectic and time consuming and it is also not possible to visit hut every time. In existing system for giving any orders users should visit hotels or restaurants to know about food items and them give order and pay. In this methodtime and manual work is required. The customers need a system which provide easy, efficient and accurate way to order food. To solve this problem, I plan to design a "Qau Huts Meal Ordering System application". This application allows users to order food from any restaurant within the premises of QAU without going to the restaurants physically.

Table of Contents

List of Tables

List of Diagrams

List of Figures

CHAPTER 1

SOFTWARE PROJECT MANAGEMENT PLAN

1. Introduction

Software Project Management Plan is an art and science of planning and leading the software projects. It is a sub-discipline of project management in which project is planned, implemented, monitored and controlled. QHMOS is application that present Food ordering system in Quaid-I-Azam University. I am proposing here greatly simplifies the ordering process for both the customer and the Huts. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the tray. In this chapter project management of QHMOS is described. This chapter provide the information that how the project of the application defined above is to manage i.e. work schedules, deliverables, tools required to build this project etc.

1.1 Project Overview

Qau huts meal ordering system is android application is the focus of the project which includes all the required documents, figures and diagrams, chart and table, tools and techniques etc. The project encompasses both server-side and client-side functionalities so, both aspects are covered in this document.

1.2 Project Deliverables

1. Software Project Management Plan (SPMP)

A document that describe tools & techniques, milestones and distribution of modules of the projects and resources to be used.

2. Software requirement Specifications (SRS)

A document that describes the scope of project, overview, major inputs & outputs and the required functionalities in the project.

3. Software Design Description (SDD)

This document defines the structure and working of project through descriptive diagrams.

4. Software Test Documentation (STD)

Test document presents test cases for the required functionalities of product, expected results & actual results.

5. Implementation

A working product having all functionalities as defined in project description.

1.3 Problem Statement

The main problem of the staff and students is that they have to go to Huts physically in order to get their desired food which is quite hectic and time consuming and it is also not possible to visit hut every time.

1.3.1 Background and Existing system

In existing system for giving any orders users should visit hotels or Huts to know about food items and then give order and pay. In this method time and manual work is required.

Following are some of the apps somehow resembling our project.

- An android application through which customer can place their order and food panda is responsible for delivering this order to the customer. This app is performing as a third party. When a customer can open this app, it searches the customer location and shows restaurant nearer to the customer location.

1.3.2 Problems in Existing System

Some of the problems in the existing system are as follows:

These apps act as a third party, if customer order any item on these apps, these apps than forwarding this order to the restaurant chosen by the customer. Then they deliver the food to the customer but their services are too slow and they take several times for delivering the meal. Rates of items are too much high and then these apps also charge extra money on the time of delivery than that of the actual price so this is also one of the problems of using these apps. One of the other problems is Customer cannot directly communicate with the restaurant in this system.

1.3.3 Proposed System

The proposed system is an Android-based application that would at the start be implemented for Quaid-e-Azam university with a mobile app, customers can easily browse all the dishes. The restaurant has available, customize dishes to their requirements and place an order. It can also save their favorite orders allowing them to easily re-order that in the future. The Qau Huts Meal Ordering System enables the end users to register online, Select the food from the Graphical User Interface menu and order food online by just selecting the food that the user wants to have.

1.4 Audience

The following people shall be the main audience of the QAU Meal Ordering System.

- Faculty of Quaid-e-Azam university
- Staff of Quaid-e-Azam university.
- Students of Quaid-e-Azam university.
- Visitors at of Quaid-e-Azam university

1.5 Project Features

- Following are the major Features of the application:
- Online menus are available in this application.
- Easy lookup of QAU huts and cafe.
- Check ratings and review of the huts.
- Simple fast and convenient ordering of food.
- Prior Knowledge of time for delivery helps prepare and provide better service.
- Prices are manageable for customers.

1.6 Requirement Analysis

The following section presents the complete set of functional and non-functional requirements identified for the subject Qau huts MOS. Functional requirements are listed first, according to their relationship to the overall system. The non-functional requirements that pertain to safety, security, the interface, human engineering, qualification, operation, maintenance and performance are subsequently presented. The functional requirements have been specified using a natural language description and as such, the reader is directed to Section 4 (UML Analysis Models) for further detail.

1.6.1 Functional Requirements

This subsection presents the identified functional requirements for the subject QAU Huts MOS. Initially, general requirements that pertain to the whole system are given. Where possible, subsequent requirements have been demarcated based on their relevance to the users of the system.

- Customer shall be able to login to his account through username and Password.
- Customer shall be able to choose a restaurant from a list of restaurants.
- Customer shall be able to see the menu of a restaurant.
- Customer shall be able to order from the menu of a restaurant.

- Customer shall be able to add an item to favorite list.
- Customer shall be able to delete an item from the favorite list.
- Customer shall be able to cancel order.
- Customer shall be able to Review order.
- Customer shall be able to Place order.
- Customer shall be able to Provide delivery address.
- Customer shall be able to Check out.
- Customer shall be able to log out.
- System shall be able to update the menu list.
- System shall be able to manage users.
- System shall be able to manage orders.

1.6.2 Non-Functional Requirements:

This subsection presents the identified non-functional requirements for the subject Qau Huts MOS.

The subcategories of non-functional requirements given are safety, security, Performance, Reliability, Usability, Flexibility.

- **Availability**: - Application should be able to provide menu items at any time.
- **Performance**: - Application must be lightweight and provide better performance.
- **Reliability**: - Application maintain consistency. Application must be reliable for the customers.
- **Usability**: - GUI should be simple and interactive. The application must satisfy a maximum number of customer needs.
- **Flexibility**: - Application must be flexible to add new menu items.

1.7 Assumption and Dependencies

User must be trained for basic android functionalities. User must have the basic knowledge of English. Customer will be required internet access to order. Riders will be available to deliver the food for nearby hostels and university area. The memory usage of the app will have to be constrained by the devices it is intended to run. Since most tablets, Android phones may have limited memory.

1.8 Project Management Plan

This section describes how the project will be managed, its tasks, deliverables and milestones.

1.8.1 Tasks

- **Requirements Analysis Phase**

  Identify Requirements

  Define Use Cases

  Develop Analysis Model

  Develop SRS

- **Design Phase**

  Develop Design

  Evaluate Design

  Develop Software Test Documentation

- **Implementation Phase**
- **Integration Phase**
- **Testing Phase**

2 Requirements Analysis Phase

This section includes task and their details for problem analysis.

1.8.2 Identify Requirements

Description

The initial step in the development of project is the identification of requirements. The requirements include both functional and non-functional requirements.

Deliverables and Milestones

Requirements are collected and reviewed.

Resources Needed

People: Sumbal Batool, Supervisor

Software: MS Word

Hardware: Laptop

Dependencies and Constraints

None

Risks and Contingencies

None

1.8.3 Define Use Cases

- **Description**

  This task includes defining and writing use cases and making use case diagram.

- **Deliverables and Milestones**

  Use cases are written down and reviewed.

- **Resources Needed**

  People: Sumbal Batool, Supervisor

  Software: MS Word, Draw.io

  Hardware: Laptop

- **Dependencies and Constraints**

  None

- **Risks and Contingencies**

  None

1.8.4 Develop Analysis Model

Description

This task includes making domain model for the system.

Deliverables and Milestones

Domain model is reviewed.

Resources Needed

People: Sumbal Batool, Supervisor

Software: MS Word, Draw.io

Hardware: Laptop

Dependencies and Constraints

None

Risks and Contingencies

None

1.8.5 Develop SRS

Description

This task includes making Software Requirements Specification document which contains the description of functional and non-functional requirements.

Deliverables and Milestones

SRS document is reviewed.

Requirement Analysis phase is complete.

Resources Needed

People: Sumbal Batool, Supervisor

Software: MS Word, Draw.io

Hardware: Laptop

Dependencies and Constraints

None

Risks and Contingencies

None

1.9 Design Phase

Develop Design

- **Description**

  This task includes the development of architectural design and detailed design of the system.
- **Deliverables and Milestones**

  Architectural and detailed design is reviewed.
- **Resources Needed**

  People: Sumbal Batool, Supervisor

  Software: MS Word, Draw.io

  Hardware: Laptop
- **Dependencies and Constraints**

The development of designs requires the completion of requirements analysis phase.

- **Risks and Contingencies**

  None

Evaluate Design

- **Description**

  This task includes evaluation and verification of the design.

- **Deliverables and Milestones**

  Design phase is completed.

- **Resources Needed**

  People: Sumbal Batool, Supervisor

  Software: MS Word, Draw.io

  Hardware: Laptop

- **Dependencies and Constraints**

  The development of designs requires the completion of requirements analysis phase.

- **Risks and Contingencies**

  None

1.10 Develop Software Test Documentation

- **Description**

  This task includes defining test cases for the system.

- **Deliverables and Milestones**

  Test cases are reviewed.

- **Resources Needed**

  People: Sumbal Batool, Supervisor

  Software: MS Word

  Hardware: Laptop

- **Dependencies and Constraints**

  To define the test cases, design phase should be complete.

- **Risks and Contingencies**

  None

1.11 Testing Phase

System Testing

- **Description**

  This phase includes the testing of application based on different testing parameters.

- **Deliverables and Milestones**

  Application is tested.

- **Resources Needed**

  People: Sumbal Batool, Supervisor

  Hardware: Android Cell Phone

- **Dependencies and Constraints**

  For the testing of application, integration phase should be complete.

- **Risks and Contingencies**

  None

CHAPTER 2

Software Requirement Specification

2.1 Introduction

QAU Huts Meal Ordering System is designed primarily for use in the food delivery. This system allows customers to quickly and easily manage an online menu, also they can browse and place orders with just few clicks. The purpose of my system is that it greatly simplifies the ordering process for customers. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After selecting, the item is then added to their order. The customer can review the details of their order at any time before checking out. This provides instant visual confirmation of what was selected and ensures that the desired items are placed in order queue. Simply this system is user friendly and easy to use without taking much time.

2.2 Purpose

The purpose of this document is to give a detailed description of the requirements for Project. This document covers the functional as well non-functional requirements of the system that happen to be the essential features of the system that are fixed, known, and agreed to be delivered. In addition to the basic requirements, it will also describe the external interface requirements.

2.3 Objective

Objectives of this system are

- To provide Facility and fastest way to both the customer and system with proper database e and information.

- To save time taken in ordering process.
- To provide safe and time saving payment mechanism.

2.4 Acronyms and Abbreviations

Table 1:Acronyms and Abbreviations

Acronym/ Abbreviations	**Description**
QHMOS	**Qau Huts Meal Ordering System**
SRS	Software Requirement Specification
SPMP	Software Project Management Plan
UC	Use Case
SSD	System Sequence Diagram
SD	Sequence Diagram
DB	Database
UI	User Interface

2.5 Use cases list

- Sign up
- Login
- Display Menu
- Add Menu Category
- Delete Category
- Update Menu
- Select Category
- Display Items
- Display Huts
- Browse Huts
- Select Huts

- Select items
- Add items
- Delete items
- Update Items
- Cancel order
- Place order
- Add delivery address
- Check out
- Display Cart
- Add to Cart
- Check Orders
- Deliver order
- Total Earnings
- Log out

2.6 Use Case Diagram

Use case diagram Figure 2.5.2 presents a use case diagram for the subject QAU Huts MOS. The key interactions between the end-users of the system and the system itself are depicted.

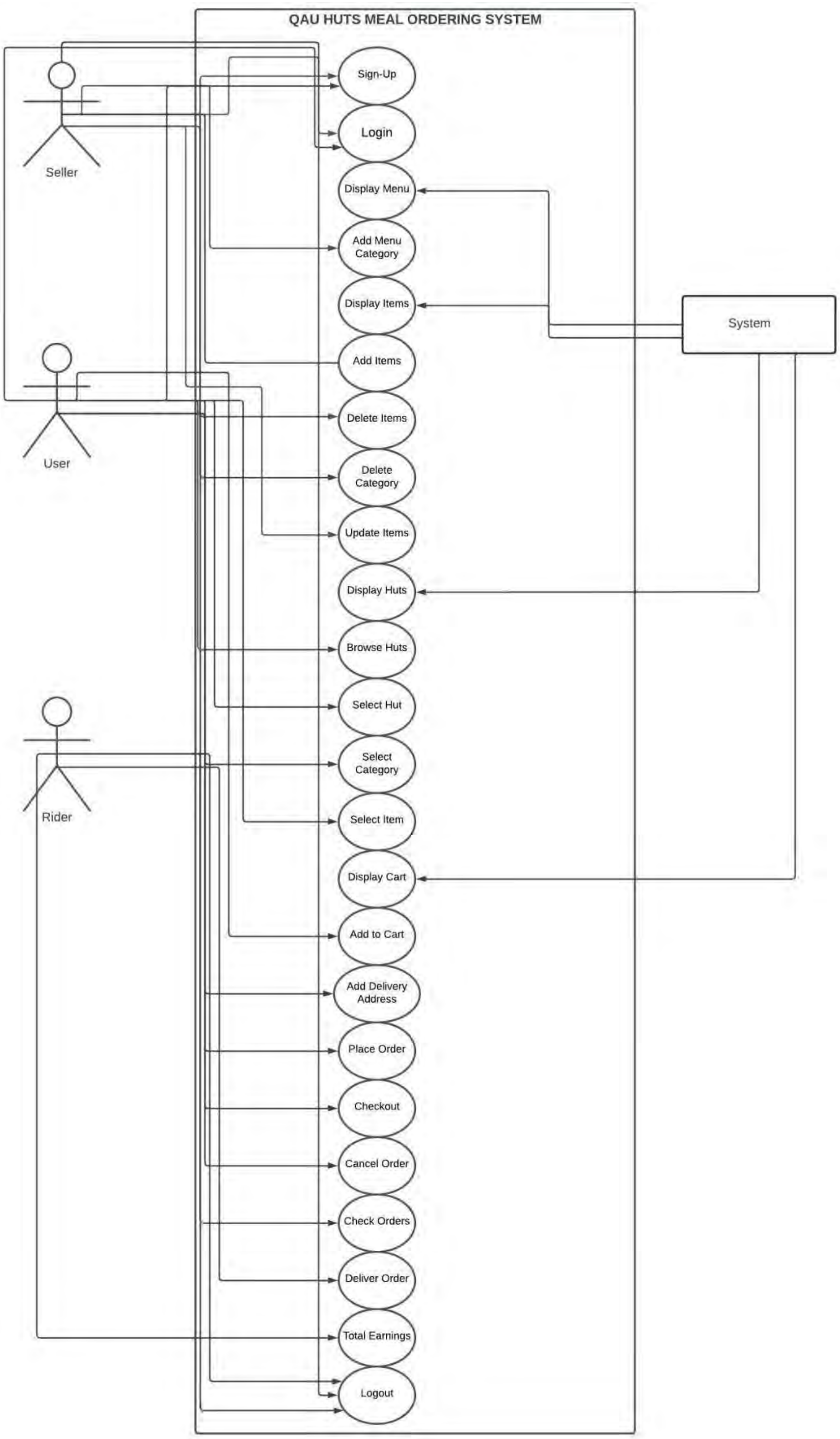


Diagram 1:Use Case diagram for QHMOS

2.7 Use Case Description: -

1. Login

Table 2 presents the Log in use case description to show the interaction between acustomer and an android when logging into the system

Use case Id	**UC-1 login**
Primary actor	Customer
Pre-condition	User first sign up to the application.
Post-condition	If login was successful, the customer is now logged into the system. If not the system state is unchanged.
Priority	High
Frequency of use	one time
Secondary Actor	System
Scenario	1. The system requests that the customer enter his/her name and password. 2. The customer enters his/her name and password. 3. The system validates the entered name and password and logs the customer into the system.
Alternate scenario	Invalid Name / Password If in the scenario the customer enters an invalid name and/or password, the system displays an error message. customer can choose to either return to the beginning of the scenario or cancel the login.

Table 2 UC-1: Login

2 Register User /Sign Up

Table 3 presents the Log in use case description to show the interaction between acustomer and an android when logging into the system**.**

Use case Id	**UC-2 Sign- up**
Primary actor	Customer /user
Pre-condition	User not able to sign in or visit app for first time.
Post-condition	User registered successfully and application home page will display.
Priority	High
Frequency of use	one time
Secondary Actor	System
Success Scenario	1. User click on "Signup" button. 2. User enters his/her username, email Id, password. 3. User select his/her profile picture 4. User enters a password of his/her choice. 5. User presses "Sign in" button. 6. Message displayed "Account created successfully" to user 7. Home Page will be displayed to user.
Alternate scenario	2a. User enter invalid info. a) User enters wrong email format refresh page 3a. Password mismatched Users enter password that is mismatched. Repeat step 3. 3b. Invalid password length User enters password of length less than desired length, repeat step 3. 4a. Empty fields a) User did not fill any mandatory fields, repeat step 2 to

Table 3 UC-2: Sign-up

3 Display Menu

Table 4 presents the Display menu use case description to show the interaction between a customer and an android.

Use case Id	**UC-3 Display menu**
Primary actor	User
Pre-condition	customer selects hut from the available huts.
Post-condition	System displays menu of selected Hut that customer select
Priority	High
Frequency of use	many time
Secondary Actor	System
Success Scenario	1. Search bar appear on homepage of app 2. The customer selects the hut from the available huts. 3. Now customer click on menu button
Alternate scenario	1. No menu available 2. The system displays the message that no hut available in that area 3. The system will show the available menu of selected hut to customer

Table 4 UC-3: Display menu

4 Browse Huts

Table 5 presents the browse Huts use case description to show the interactionbetween a customer and an android.

Use case Id	UC-4 Browse huts
Primary actor	Customer
Pre-condition	customer enters area or turn on location
Post-condition	System displays list of Huts that customer search
Priority	High
Frequency of use	many time
Secondary Actor	System
Success Scenario	4. Search bar appear on homepage of app 5. The customer enters the hut name or place name 6. Now customer click on search button
Alternate scenario	3. No hut available 4. The system displays the message that no hutavailable in that area 3. The system will show theavailable huts to customer nearby that area.

Table 5UC-4: Browse huts

5 Add Menu Category

Table 6 presents the add menu category use case description to show the interaction between a customer and an android**.**

Use case Id	**UC-5** Add Menu Category
Primary actor	Seller
Pre-condition	Menu are displayed
Post-condition	Category are added into menu
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	1. The seller observes the menu items. 2. seller then selects the desired food items. 3. The customer then adds the selected food items to the category by clicking the button **'Add to Category'.**
Alternate scenario	1. In case the menu is not available to the seller, or the system fails to display menu to the seller. 1a. the seller searches their desired items in search bar and select the items. 2. If the seller selects the items and add them to "menu category but not mentioned their quantity. 2a: The system generates a message to ask seller for enter quantity of item.

Table 6 UC-5: Add Menu Category

6 Delete Category

Table 7 presents the delete menu category use case description to show the interaction between a seller and an android**.**

Use case Id	**UC-6** Delete Category
Primary actor	Seller
Pre-condition	Menu Category are displayed
Post-condition	Category are deleted from menu
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	4. The seller observes the menu categories. 5. seller then selects the desired category. 6. The seller then deletes the selected category by clicking the button **'Delete Category'.**
Alternate scenario	3. In case the menu is not available to the seller, or the system fails to display menu to the seller. 1a. the seller searches their desired category in search bar and delete the category. 4. If the seller selects the category and delete it from "menu category but not mentioned their items. 2a: The system generates a message to ask seller for enter items.

Table 7 UC-6: Delete Category

7 Select Category

Table 8 presents the select category use case description to show the interactionbetween a seller and an android**.**

Use case Id	**UC-7** Select Category
Primary actor	Seller
Pre-condition	Menu category are displayed
Post-condition	category is selected from menu
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	**1.** The seller observes the menu categories. **2.** seller then browse the available categories. **3.** The seller then selects the desired category by clicking the button '**Select Category'**
Alternate scenario	5. In case the menu is not available to the customer, or the system fails to display menu to the customer. 1a. the customer searches their desired category in search bar and select the category. 6. If the customer selects the items and add them to "my cart" but not mentioned their items. 2a: The system generates a message to ask customer for enter item.

Table 8 UC-7: Select Category

8 Select Item

Table 9 presents the select menu items use case description to show the interactionbetween a customer and an android**.**

Use case Id	**UC-8** Select Menu Items
Primary actor	Customer
Pre-condition	Menu items are displayed
Post-condition	item is selected and add into cart
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	4. The customer observes the food items. 5. customer then selects the desired food items. 6. The customer then adds the selected food items to the cart by clicking the button **'Add to my cart'.** 7. In '**My cart**' customer provides details of selected items. 8. Details include Quantity of items.
Alternate scenario	7. In case the menu is not available to the customer, or the system fails to display menu to the customer. 1a. the customer searches their desired items in search bar and select the items. 8. If the customer selects the items and add them to "my cart" but not mentioned their quantity. 2a: The system generates a message to ask customer for enter quantity of item.

Table 9 UC-8: Select Menu Items

9 Add items

Table 10 presents the add item use case description to show the interaction between a customer and an android.

Use case Id	UC-9 Add item to cart
Primary actor	Customer
Pre-condition	The Customer selected items from menu list of hut
Post-condition	Menu items have been added to cart
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	1. Customer should access to system 2. Customer selects items from menu list 3. Customer select "add to cart" Cart will show final items with bill and total amount of each item
Alternate scenario	1. The old item may be not updated in cart which customer wants to replace 2. Customer wants to decrease amount of item

Table 10 UC-9: Add items to cart

10 Update items

Table 11 presents the update items use case description to show the interactionbetween a user and an android**.**

Use case Id	**UC-10 Update item to cart**
Primary actor	user
Pre-condition	The user selected items from menu list of hut
Post-condition	Menu items have been updated in to cart
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	4. Customer should access to system 5. Customer selects items from menu list to update 6. Customer select "update item" Cart will show final updated items with bill and total amount of each item
Alternate scenario	3. The old item may be not updated in cart which customer wants to replace 4. Customer wants to decrease amount of item

Table 11 UC-10: Update item to cart

11 Display items

Table 12 presents the Display items use case description to show the interaction between a customer and an android**.**

Use case Id	**UC-11 Display items to cart**
Primary actor	Customer
Pre-condition	The Customer selected items from menu list of huts
Post-condition	Menu items have been displayed to cart
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	7. Customer should access to system 8. Customer selects items from menu list 9. Customer select "add to cart" Cart will show final items with bill and total amount of each item
Alternate scenario	5. The old item may be not updated in cart which customer wants to replace 6. Customer wants to decrease amount of item

Table 12 UC-11: Display items to cart

12 Display cart

Table 13 presents the Display cart use case description to show the interaction between a customer and an android**.**

Use case Id	**UC-12 Display cart to user**
Primary actor	user
Pre-condition	The Customer selected items from menu list of huts
Post-condition	cart has been displayed to user
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	10. Customer should access to system 11. Customer selects items from menu list 12. Customer select "add to cart" system will show final items with bill and total amount of each item in carts
Alternate scenario	7. The old item may be not updated in cart which customer wants to replace 8. Customer wants to decrease amount of item

Table 13 UC-12: Display cart to user

13 Delete items Use Case

Table 14 presents the delete items use case description to show the interaction between a customer and an android**.**

Use case Id	UC-13 delete items from cart
Primary actor	Customer
Pre-condition	The Customer place selected items into cart
Post-condition	Item will be removed from cart and system displays the final list of selected items
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	1. The customer clicks on “view cart” 2. Cart will show final items with bill and total amount of each item 3. Negative sign will indicate the sign of remove item 4. The customer select item and click on negative sign to remove the item from cart 5. Now the quantity of item and bill will be decreased or item will be removed from cart 6. Now the customer can click on “confirm order’’.
Alternate scenario	1. System may respond late because of connectivity issue 2. Customer click on remove item to decrease quantity accidently two times now no item will be found in cart.

Table 14 UC-13: Delete items from cart

14 Display Huts

Table 15 presents the Display Huts use case description to show the interactionbetween a customer and an android**.**

Use case Id	**UC-14 Display Huts to user**
Primary actor	Customer
Pre-condition	The Customer logged in to the system
Post-condition	Huts have been displayed to user
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	13. Customer should access to system 14. Customer enable location or enter in the app search bar system will show the available huts in that area to user
Alternate scenario	9. The system did not obtain the user location 10. Customer did not logged in.

Table 15 UC-14: Display huts to user

15 Select Huts

Table 16 presents the select huts use case description to show the interaction betweena customer and an android.

Use case Id	**UC-15** Select Huts
Primary actor	User
Pre-condition	User is logged in to the system.
Post-condition	Huts are selected by the user.t
Priority	High
Frequency of use	More than 500 times a day, mostly used between 1pm to 4pm
Secondary Actor	System
Success Scenario	9. The customer observes the available huts. 10. customer then selects the desired hut
Alternate scenario	9. In case the huts are not available to the customer, or the system fails to display huts to the customer. 1a. the customer searches their desired huts in search bar and select the hut.

Table 16 UC-15: Select huts

16 Place order Use Case

Table 17 presents the place order use case description to show the interactionbetween a customer and an android.

Use case Id	UC- 16 Place order
Primary actor	User
Pre-condition	The Customer should be log in and system is configured to take input.
Post-condition	Order has been made/is placed.
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	Customer browses for favorite huts Customer selects “View Menu” from the main screen Customer selects from the menu items Customer selects “Add to cart” Customer selects “View cart” Customer clicks on Confirm order
Alternate scenario	Customer enters wrong order and wants to go back to main menu Customer accidently press on confirm order button

Table 17 UC-16: Place order

17 Cancel Order Use Case

Table 18 presents the cancel order use case description to show the interactionbetween a customer and an android.

Use case Id	UC- 17 Cancel Order
Primary actor	Customer
Pre-condition	Customer must be logged in and have placed order in the cart.
Post-condition	Customer has removed the order from the cart successfully
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	Customer selects "My cart" option. Customer selects an order. Customer selects remove order. System generates confirmation message. Customer confirms the message and press ok.
Alternatescenario	5a. Customer cancel confirmation a) Customer cancelthe confirmation ask by the system, repeat all steps again.

Table 18 UC-17: Cancel Order

18 Add delivery address Use Case

Table 19 presents the Add delivery address use case description to show the interaction between a customer and an android**.**

Use case Id	**UC-18 Add delivery address**
Primary actor	user
Pre-condition	The order has been confirmed.
Post-condition	Customer's address has been saved in system Success.
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	1. The customer should have logged in to system 2. The customer selects items from menu for order 3. The customer confirms the order 4. Screen will display interface to enter delivery address in respective fields 5. The customer enters address for delivery in respective field and click to proceed
Alternate scenario	1. The customer wants to cancel the order and doesn't provide delivery address 2. The customer accidently provides wrong address 3. The customer wants to change the delivery address.

Table 19 UC-18: Add delivery address

19 Add to Cart Use Case

Table 20 presents the Add to Cart use case description to show the interaction between a customer and an android.

Use case Id	**UC-19** Add to cart address
Primary actor	user
Pre-condition	The order has been confirmed.
Post-condition	Customer has successfully added items to cart
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	5. The customer should have logged in to system 6. The customer selects items from menu for order 7. The customer confirms the order 8. Screen will display interface to select food items from menu. 5. The customer select items and select the add to cart option.
Alternate scenario	4. The customer wants to update the order and doesn't provide items 5. The customer accidently provides wrong items 6. The customer wants to change the menu category.

Table 20 UC-19: Add to cart address

20 Check out Use Case

Table 21 presents the make payment use case description to show the interaction between a customer and an android**.**

Use case Id	**UC-** 20 Check out
Primary actor	Customer
Pre-condition	Login, Customer must have an order in cart.
Post-condition	1. Customer makes payment for the order he select. 2. Receipt has been generated.
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	1. After Customer make an order. Customers click on proceed to payment button 2. System shows a message “your order has been placed”. 3. System generates print of receipt order.
Alternate scenario	2a. Customer cancel confirmation b) Customer cancel the confirmation ask by the system, repeat all steps again.

Table 21 UC-20: Check out

21 Update Menu Use Case

Table 22 presents the Update menu use case description to show the interaction between system and customer.

Use case Id	UC- 21 Update menu
Primary actor	System
Pre-condition	System checks the menu list and add some new items.
Post-condition	Menu list has been updated.
Priority	Medium
Frequency of use	Once or twice in months.
Secondary Actor	System
Success Scenario	1.System enter the system 2.System makes required changes 3.System save changes.
Alternate scenario	1. Some of the menu might not need any change. 2.system may not update the changes.

Table 22 UC-21: Update menu

22 Check orders Use Case

Table 23 presents the use case Check order description to show the interactionbetween rider and customer.

Use case Id	**UC- 22** Check orders
Primary actor	Rider
Pre-condition	Customer has to booked order.
Post-condition	1.Rider checks order details. 2.Rider checks the order.
Priority	High
Frequency of use	Many times,
Success Scenario	1. Rider check Number of items 2. Rider check delivery address of customer 3. Rider checks the no of orders by selecting check order option
Alternate scenario	1 Rider check order option is not available.

Table 23 UC-22: Check orders

23 Deliver order Use Case

Table 24 presents the use case deliver order description to show the interaction between system and customer.

Use case Id	**UC- 23** Deliver order
Primary actor	Rider
Pre-condition	Customer has to booked order.
Post-condition	1.Rider checks order details. 2.Rider delivered the order.
Priority	High
Frequency of use	Many times,
Success Scenario	4. Rider check Number of items and make payment. 5. Check delivery address of customer 6. Deliver order
Alternate scenario	1 Customer selected items are not available.

Table 24 UC-23: Deliver order

24 Total Earnings Use Case

Table 25 presents the use case Total earnings description to show the interactionbetween system and customer.

Use case Id	**UC- 24** Total Earnings
Primary actor	Rider
Pre-condition	Rider has to logged in to rider app.
Post-condition	1. Rider checks order details. 2. Rider view the total earnings.
Priority	High
Frequency of use	Many times,
Success Scenario	1. Rider must have access to the rider app. 2. Rider selects the total earning options from the interface 3. System displays the total earnings to the rider.
Alternate scenario	1 The total earnings option is not available.

Table 25 UC-24: Total Earnings

25 Log out Use Case

Table 26 presents the Log Out use case description to show the interaction between awaiter and a tablet when logging out of the system.

Use case Id	**UC- 25** logout
Primary actor	User
Pre-condition	User must log in to the system.
Post-condition	Users log out to system
Priority	High
Frequency of use	Many times,
Secondary Actor	System
Success Scenario	1. System display logout option. 2. User selects the option.
Alternate scenario	None b) Customer cancel the confirmation ask by the system, repeat all steps again.

Table 26 UC-25: Logout

2.8 System Sequence Diagram

This section provides pictorial representation of sequenced followed in the interaction between user and application for the available application functionalities

1 UC-1: Create Account /Sign Up

The diagram below demonstrates the sequence of functions perform in the interaction of User and Application to create new account. After installing application user create his/her account by Signing Up into application.

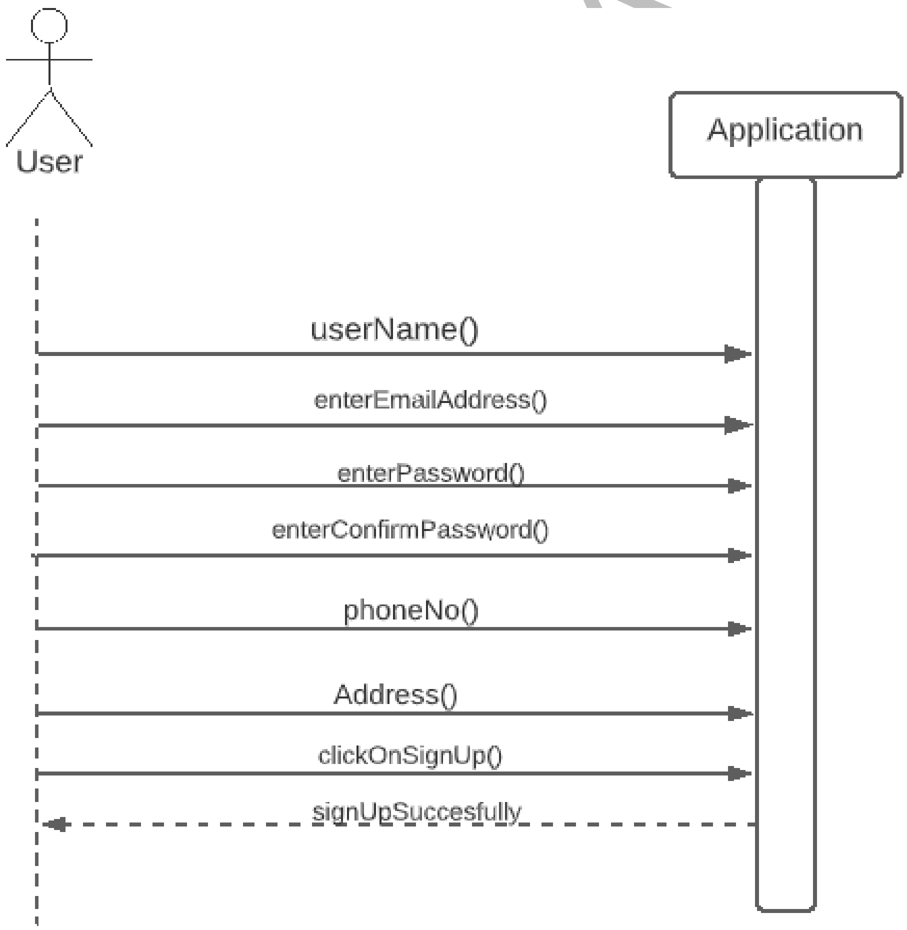


Diagram 2: SSD of Sign Up in QHMOS

2 2.7.2 UC-2: Log In

The following diagram demonstrates the interaction between application and system or registered user in order to log in to the application. User login to the application using email and password and system validate from database.

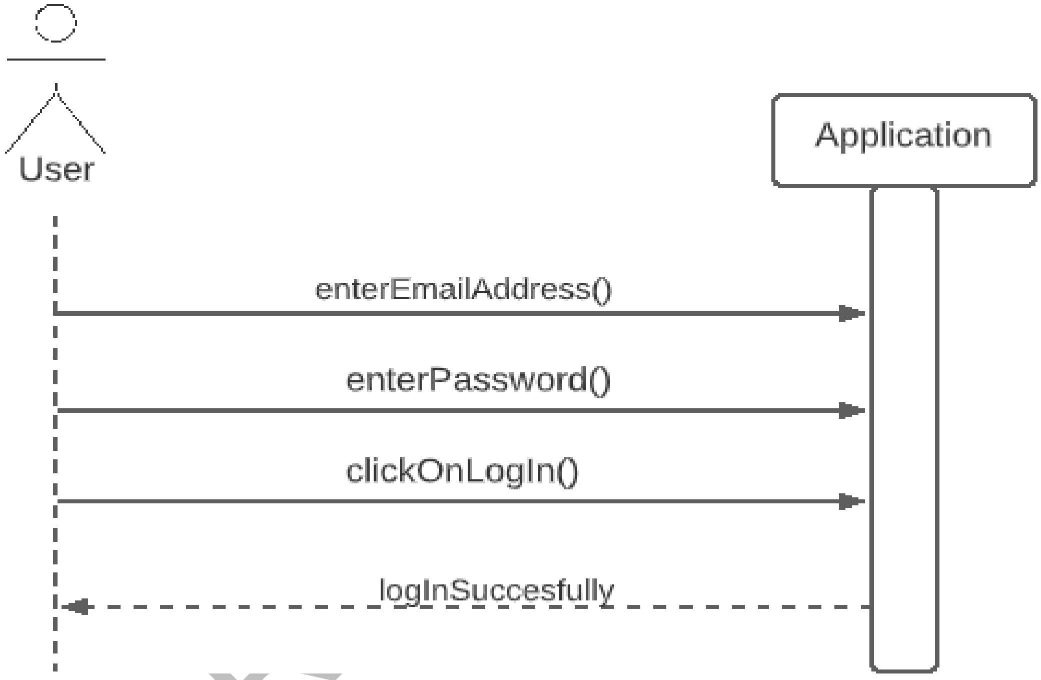


Diagram 3: SSD of Log in Use Case

3 UC-3: Browse Hut*s*

The following diagram demonstrates the interaction between application and customer in order to browse the huts. After login home page will be displayed to user. User can browse huts and navigate to available huts.

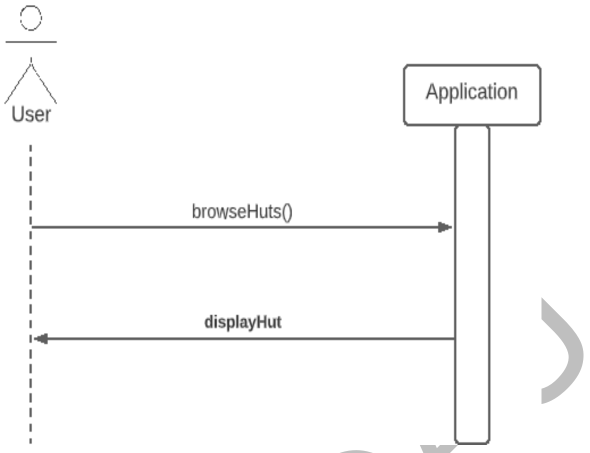


Diagram 4: SSD of Browse Huts in QHMOS

4 UC-4: Select items

The following diagram demonstrates the interaction between application or customer in order to select Menu items. Customer can choose hut and menu list of selected huts will be displayed.Customer select menu item from the list

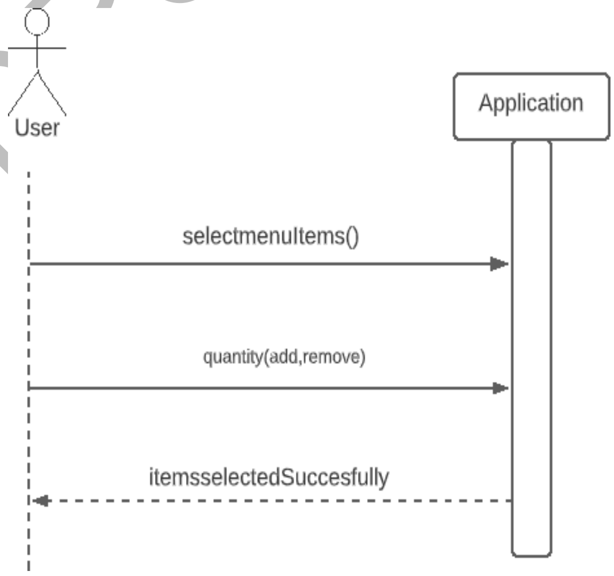


Diagram 5: SSD of Select menu Items in QHMOS

5 UC-5: Add items

The following diagram demonstrates the interaction between application and customer in order to add the items. Customer can select menu item and add items to cart for further process to make order

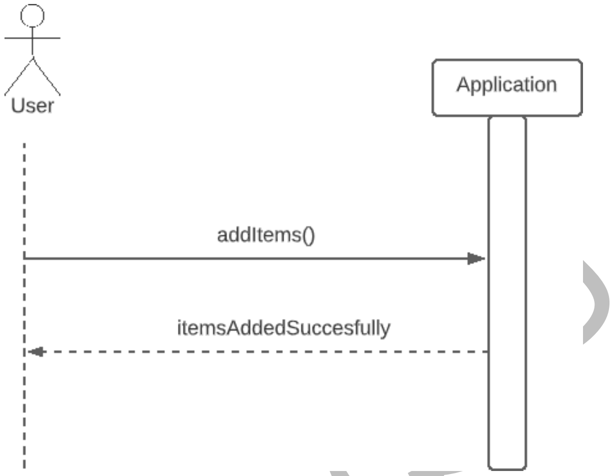


Diagram 6: SSD of Add Items in QHMOS

6 UC-6: Delete items

The following diagram demonstrates the interaction between application or customer in order to remove the items. Customer can select menu item from the cart and remove items to cancel selected item

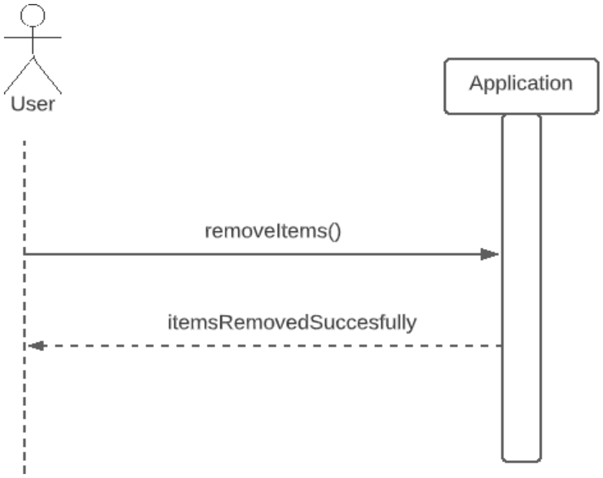


Diagram 7: SSD of Delete items in QHMOS

7 UC-7: Display Items

The following diagram demonstrates the interaction between application or customer in displaying items. Customers are allowed to see items before checkout after select items from menu.

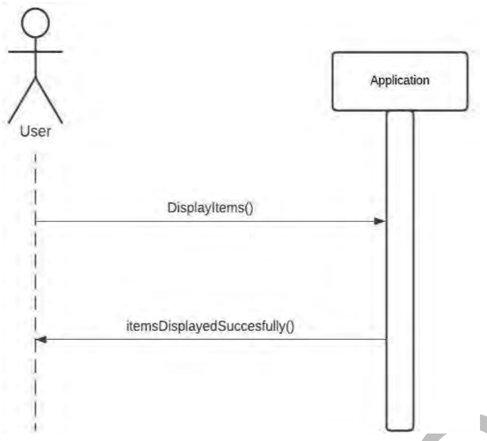


Diagram 8: SSD of Display items in QHMOS

8 UC-8: Update items

The following diagram demonstrates the interaction between application or customer in order to update the items. Customer can select menu item from the cart and update them.

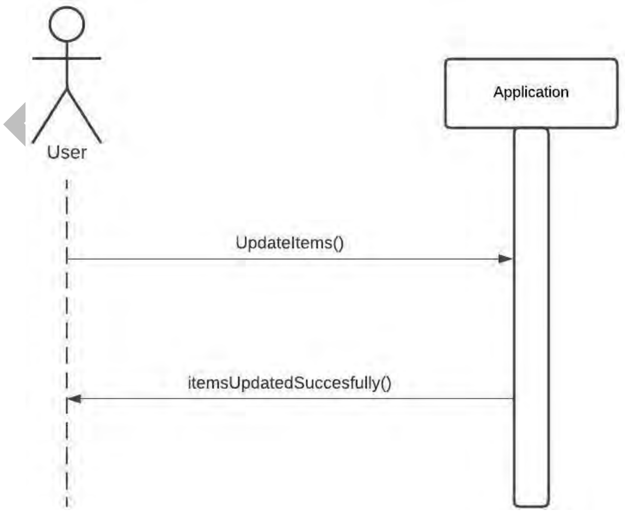


Diagram 9: SSD of Update items in QHMOS

9 UC-9: Display Menu

The following diagram demonstrates the interaction between application or customer in displaying menu. Customers are allowed to see menu before checkout after select items from menu.

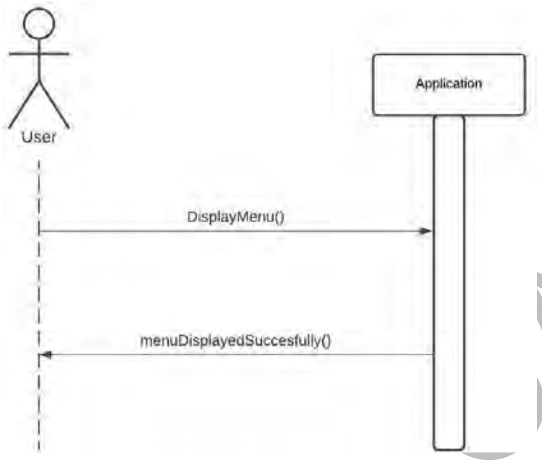


Diagram 10: SSD of Display Menu in QHMOS

10 UC-10: Cancel order

The following diagram demonstrates the interaction between application or customer in canceling order. Customers are allowed to cancel order before checkout after select items

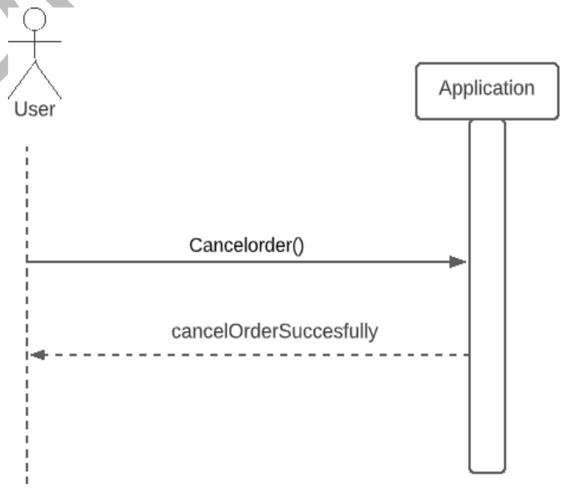


Diagram 11: SSD of Cancel Order in QHMOS

11 UC-11: Add Menu Category

The following diagram demonstrates the interaction between application or seller in adding menu category. seller is allowed to add menu category before checkout after select items

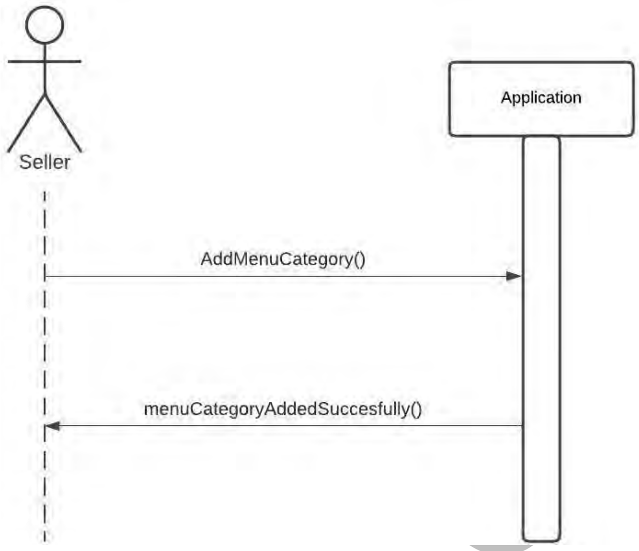


Diagram 12: SSD of Add Menu Category in QHMOS

12 UC-12: Delete Category

The following diagram demonstrates the interaction between application or seller in deleting menu category. seller is allowed to delete menu category after adding items in menu.

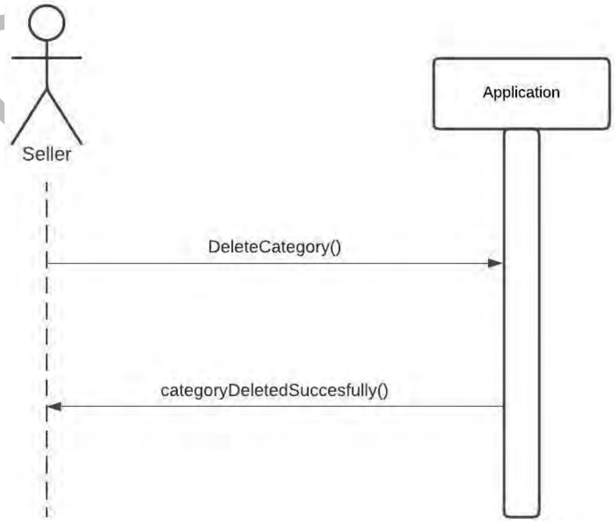


Diagram 13: SSD of Delete Category in QHMOS

13 UC-13: Select Category

The following diagram demonstrates the interaction between application or seller in order to select category. seller is allowed to select menu category after logging in to application.

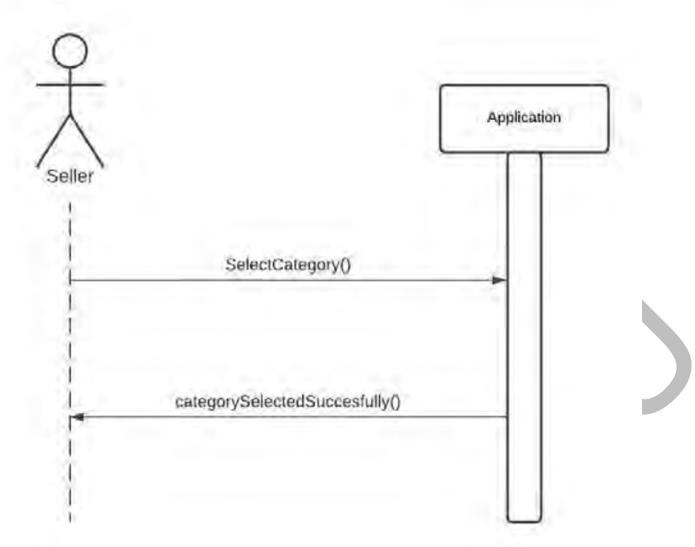


Diagram 14: SSD of Select Category in QHMOS

14 UC-14: Place order

The following diagram demonstrates the interaction between application or customer in order to place. Customer can confirm order after selecting hut and items and proceed to check out

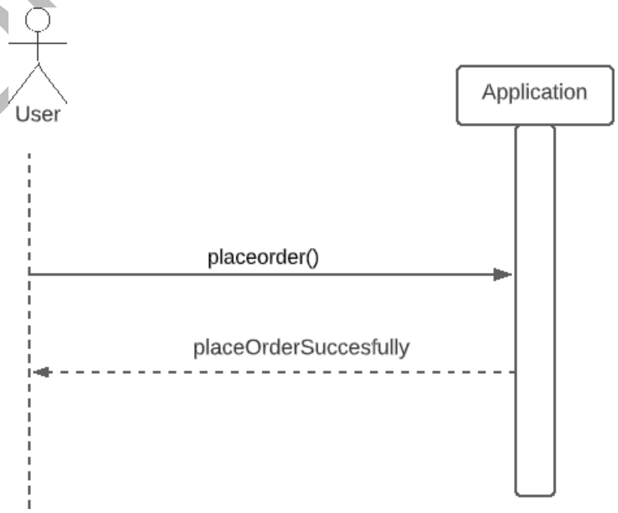


Diagram 15: SSD of Place Order in QHMOS

15 UC-15: Select Huts

The following diagram demonstrates the interaction between application or user in order to select huts. User is allowed to select huts after logging in to application.

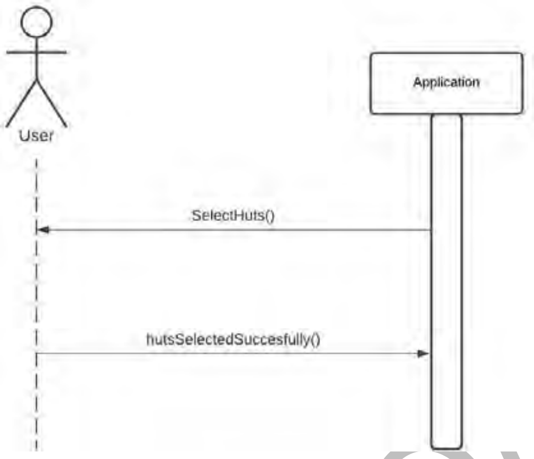


Diagram 16: SSD of Select Huts in QHMOS

16 UC-16: Display Huts

The following diagram demonstrates the interaction between application or user in order to display huts. User is allowed to see huts after logging in to application.

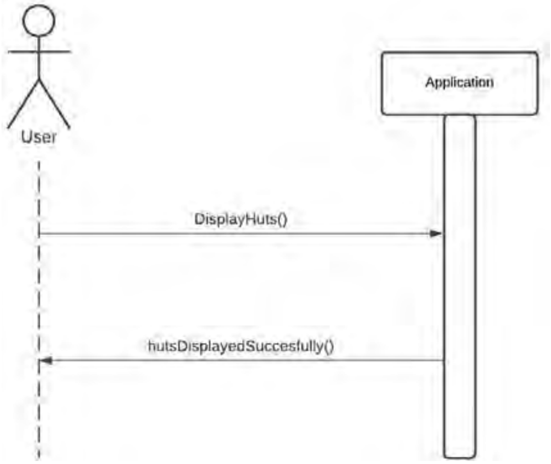


Diagram 17: SSD of Display Huts in QHMOS

17 UC-17: Add delivery Address

The following diagram demonstrates the interaction between application and customer in order to add delivery address. Customer can confirm order and provide delivery address to receive order.

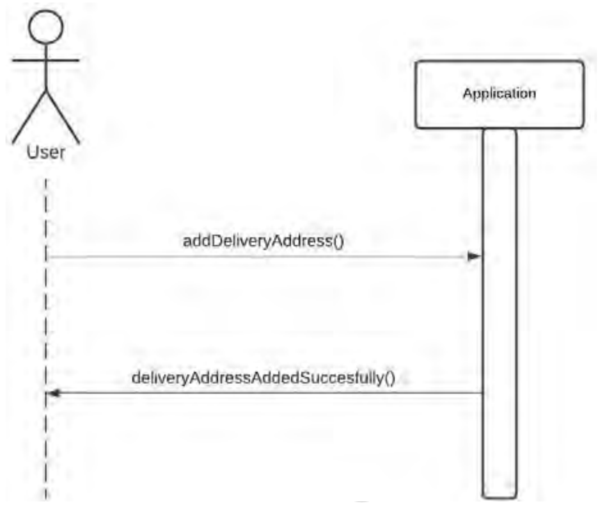


Diagram 18: SSD of Add Delivery Address in QHMOS

18 UC-18: Check Out

The following diagram demonstrates the interaction between application or customer in order to Check out. Customer can place order and proceed to check out to make payment and place order

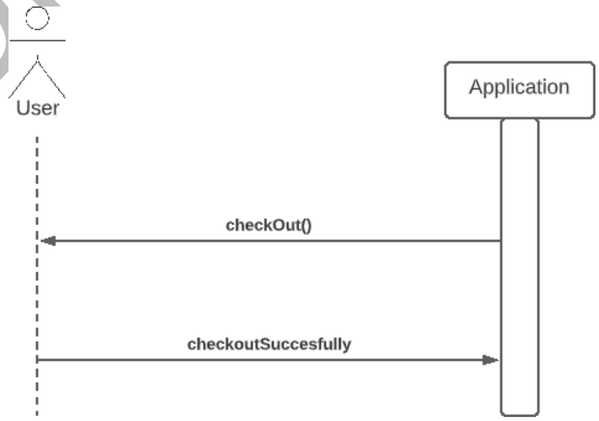
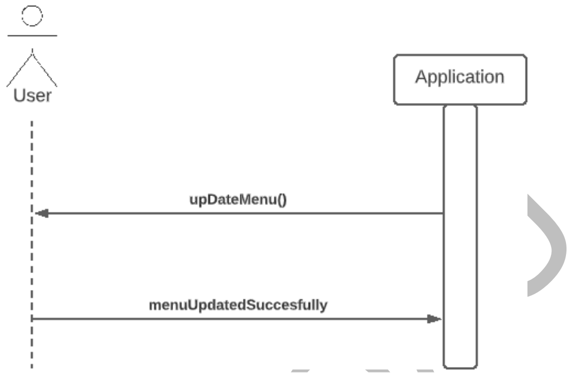


Diagram 19: SSD of Check Out in QHMOS

19 2.5.19 UC-19: Update Menu

The following diagram demonstrates the interaction between application or customer in order to Update menu. Admin is allowed to update menu list and system display updated menu list



0

Diagram 20: SSD of Update Menu in QHMOS

20 UC-20: Deliver Order

The following diagram demonstrates the interaction between application or customer in order to deliver order. Customer successfully place order and order should be delivered to customer

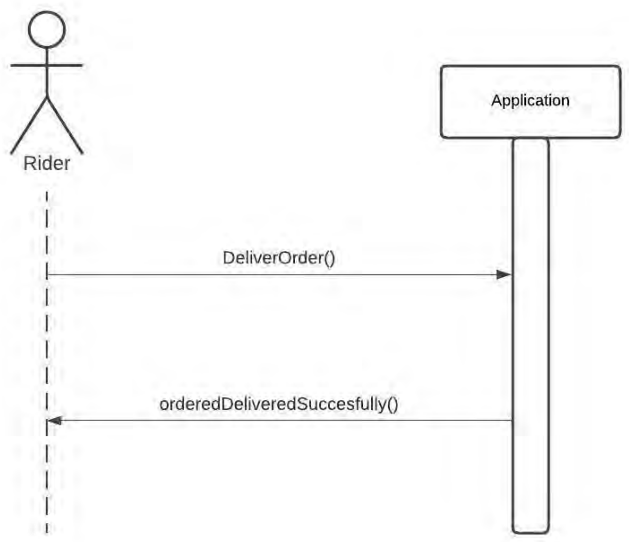


Diagram 21: SSD of Deliver Order in QHMOS

21 UC-21: Display Cart

The following diagram demonstrates the interaction between application or user in order to display cart. User is allowed to see cart after adding items from menu in to cart.

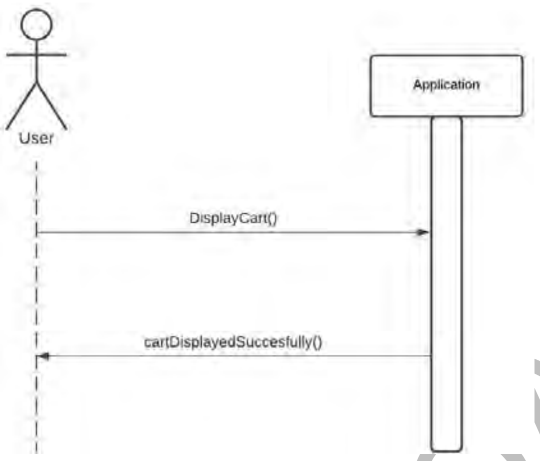


Diagram 22: SSD of Display Cart in QHMOS

22 UC-22: Add to Cart

The following diagram demonstrates the interaction between application or user in order to adding items in to cart. User is allowed to add items to cart after selecting add to cart option.

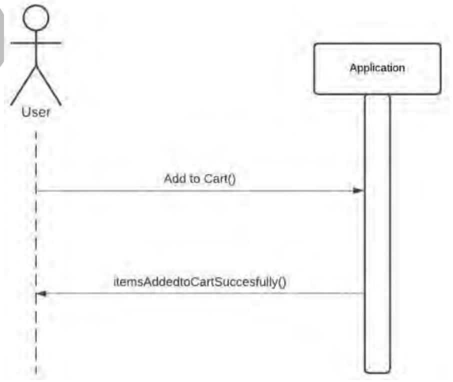


Diagram 23: SSD of Add to Cart in QHMOS

23 UC-23: Check Orders

The following diagram demonstrates the interaction between application or rider in order to check orders. Rider is allowed to check orders before delivering.

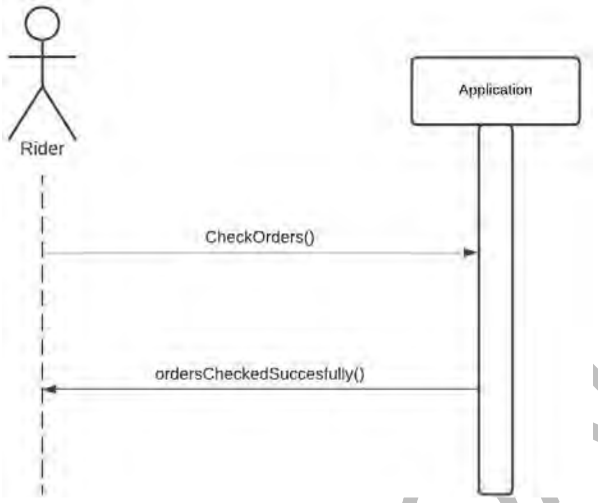


Diagram 24: SSD of Check Orders in QHMOS

24 UC-24: Total Earnings

The diagram below is to visually describe interaction of System and Rider to obtains the total earnings.

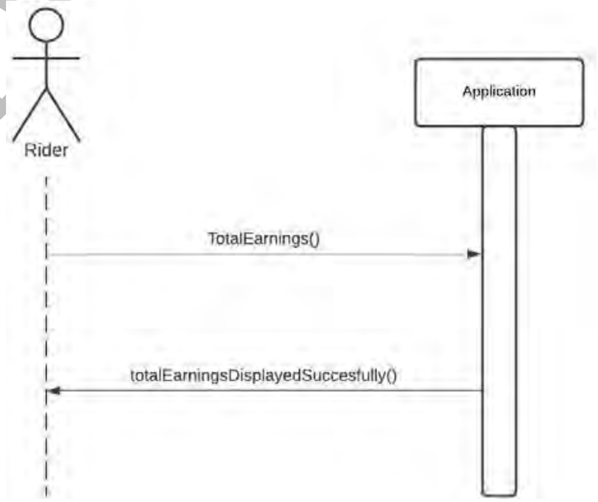


Diagram 25: SSD of Total Earnings in QHMOS

25 UC-25: Log Out

The diagram below is to visually describe interaction of System and User to log out user from application and save user’s data.

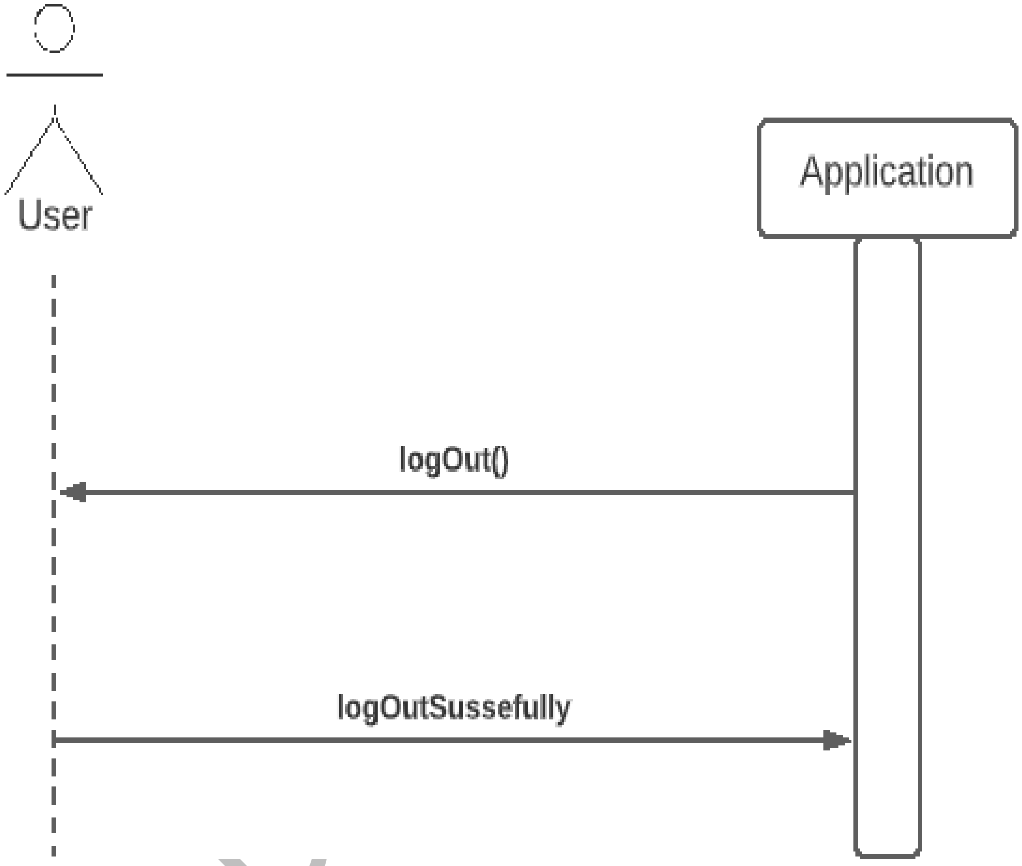


Diagram 26: SSD of Log Out in QHMOS

2.9 Domain Model

Domain model is used for representation of conceptual classes. Domain model contains classes their association, interaction of objects and convert design components into classes.

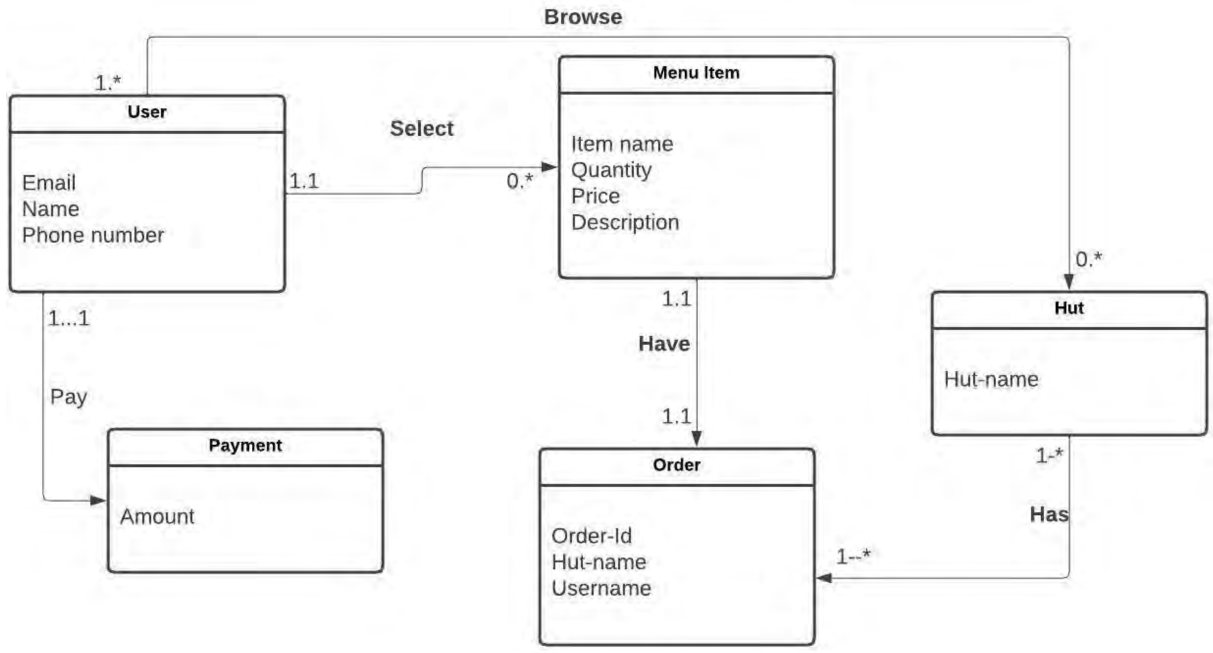


Diagram 27: Domain Model of Qau Huts Meal Ordering System

2.10 Summary

The above chapter has described the purpose of for QHMOS project, its scope, overview and major inputs & outputs. Software, hardware, application and user interfaces are described along with product functions. Furthermore, user interactions with the product are demonstrated via Use Case Diagrams and descriptions and System Sequence Diagram and Domain Model.

CHAPTER 3

SYSTEM DESIGN DOCUMENT

3.1 Introduction

Software Design Description (SDD) is a document which provides the complete description of the design of the software to be developed before the actual development. The SDD document describes the system architecture design in detail and provides a complete description of the different components. It also describes how the different components will communicate with each other. The SDD document also contains the architecture diagram, sequence diagrams and class diagram.

3.2 Design Overview

In the design phase of the system we describe the system's architecture and complete structure of the system. We discuss how user interacts with the system. In this phase we design interaction diagrams of system which includes class diagram that and sequence diagram which shows interaction between objects.

3.3 Sequence Diagrams

Sequence diagrams in the UML are primarily used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves. The following sequence diagrams are in our system that demonstrate the overall functionality of system and interaction between different objects.

1 Create Account/Sign Up

The diagram below demonstrates the sequence of functions perform in the interaction of User and Application to create new account. After installing application user create his/her account by Signing Up into application.

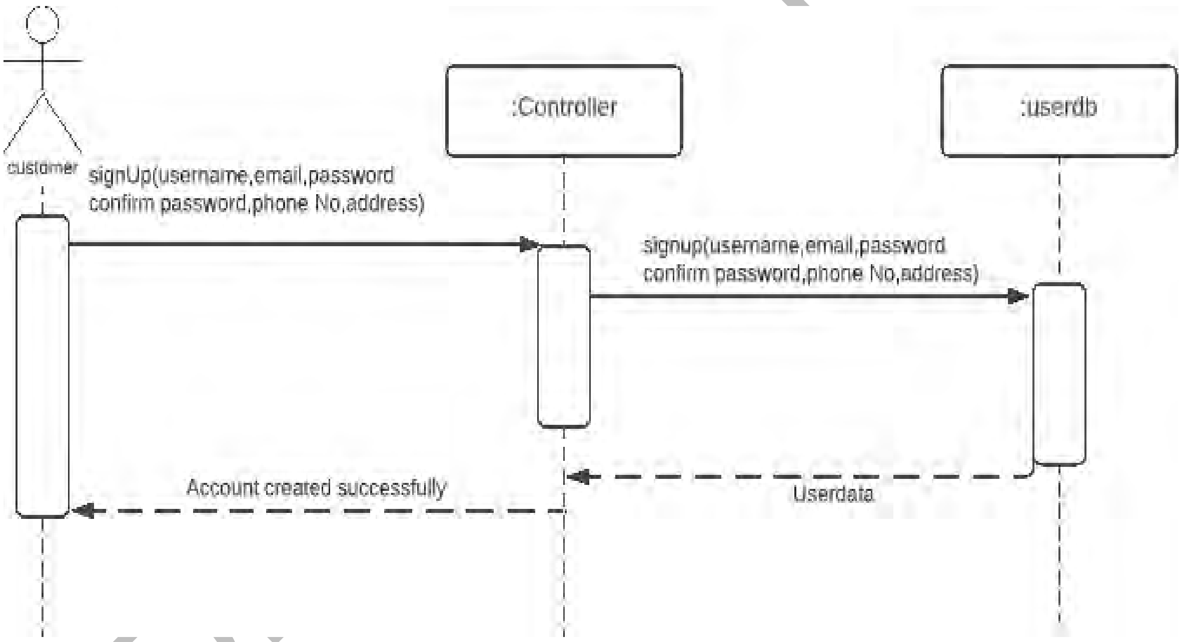

customer
:Controller
:userdb
signUp(username,email,password confirm password,phone No,address)
signup(username,email,password confirm password,phone No,address)
Userdata
Account created successfully

Diagram 28: Sequence Diagram of creating new account in QHMOS

2 Login

The following diagram demonstrates the interaction between application and system or registered user in order to log in to the application. User login to the application using email and password and system validate from database.

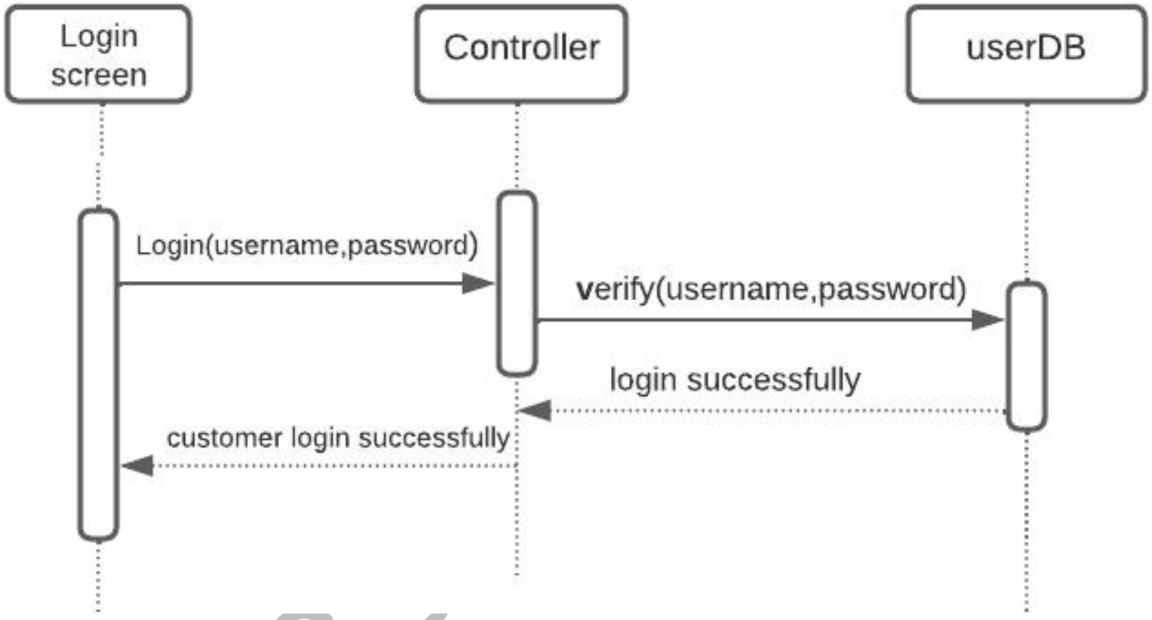


Diagram 29: Sequence Diagram of login account in QHMOS

3 Browse Hut

The following diagram demonstrates the interaction between application and customer in order to browse the huts. After login home page will be displayed to user. User can browse huts and navigate to available huts.

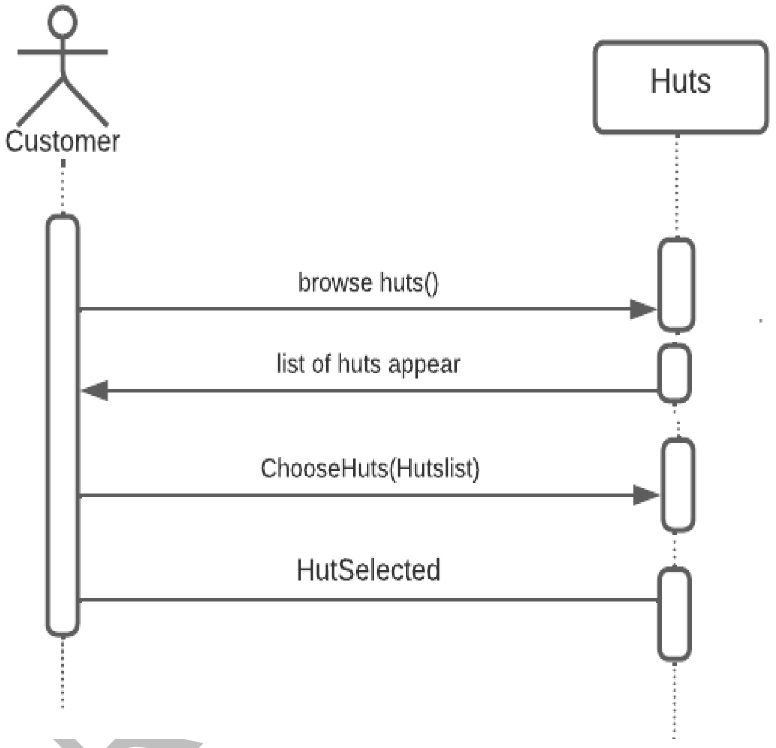


Diagram 30: Sequence Diagram of Browse Huts in QHMOS

4 Select Items

The following diagram demonstrates the interaction between application or customer in order to select Menu items. Customer can choose hut and menu list of selected hut will be displayed. Customer select menu item from the list

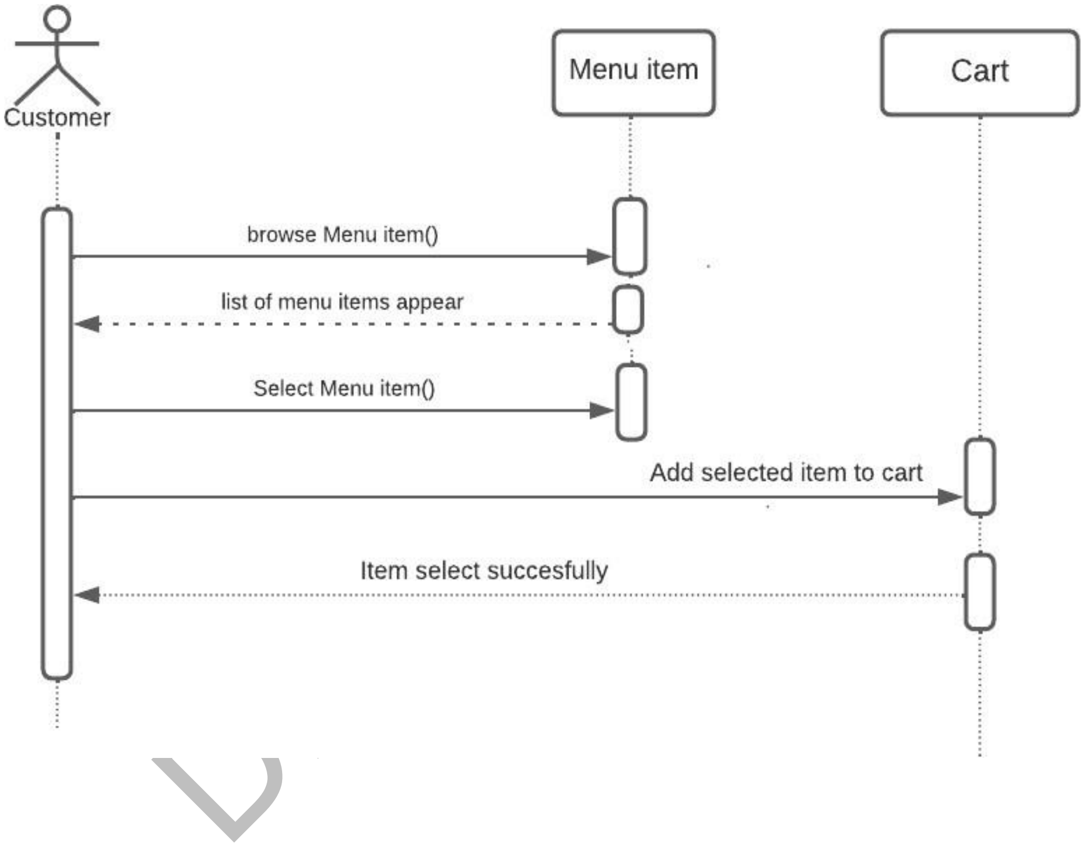


Diagram 31: Sequence Diagram of select Menu Items in QHMOS

5 Add to cart

The following diagram demonstrates the interaction between application and customer in order to add the items. Customer can select menu item and add items to cart for further process to make order

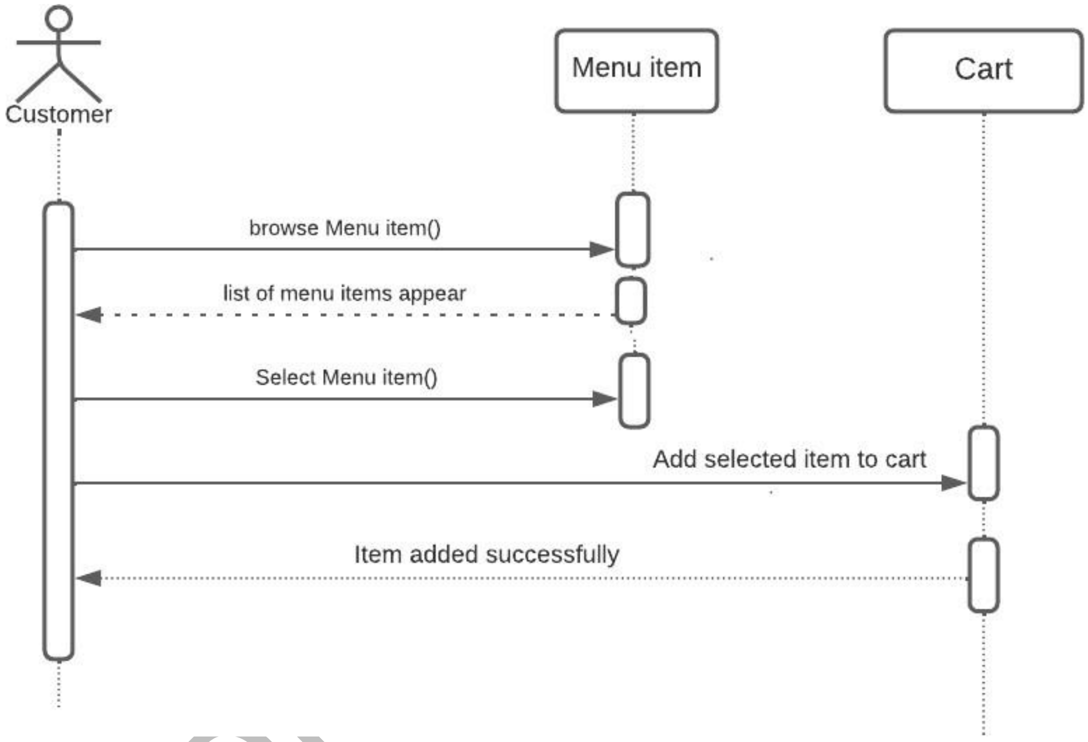


Diagram 32: Sequence Diagram of Add to cart in QHMOS

6 Delete items

The following diagram demonstrates the interaction between application or customer in order to remove the items. Customer can select menu item from the cart and remove items to cancel selected item

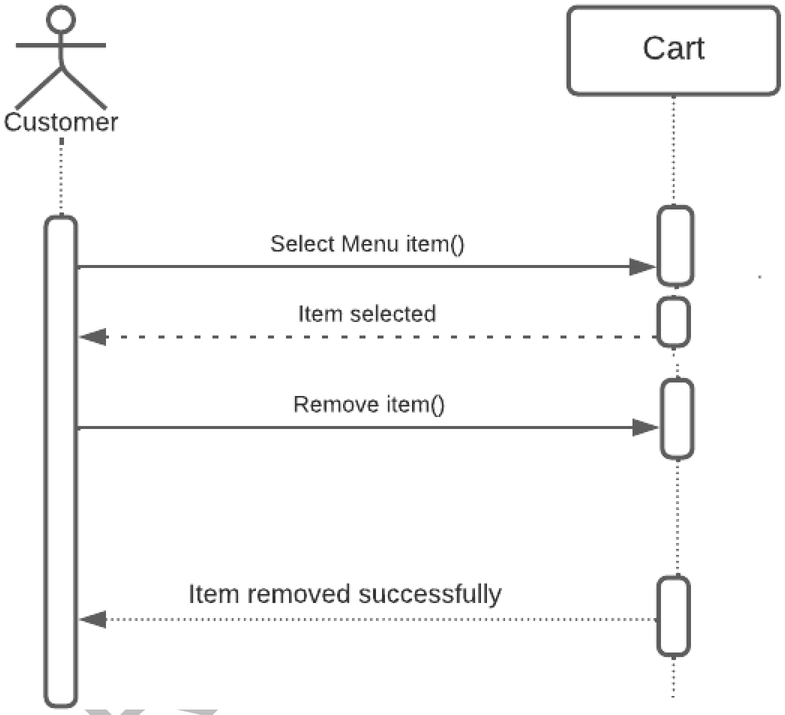


Diagram 33: Sequence Diagram of Delete items in QHMOS

7 Review cart:

The following diagram demonstrates the interaction between application or customer in order to remove the items. Customer can review menu item from the cart and add or remove item quantity

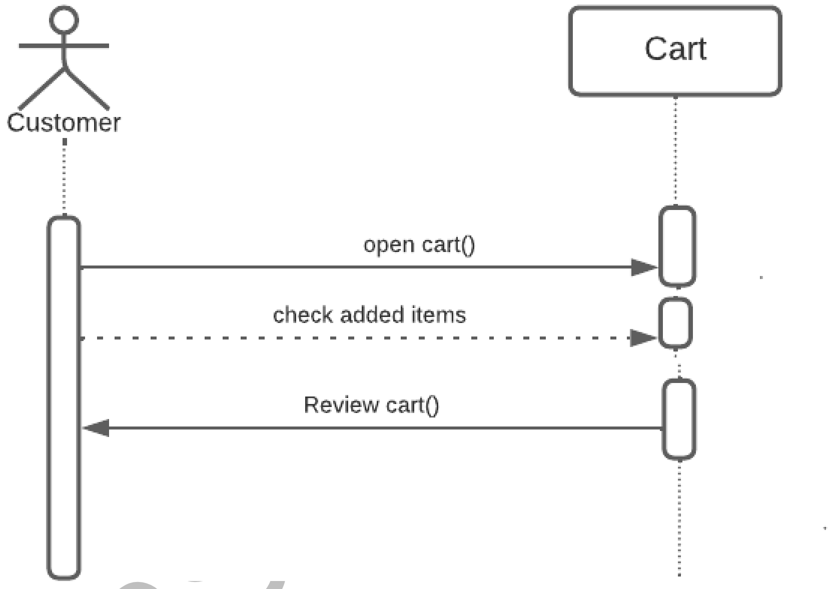


Diagram 34: Sequence Diagram of Review cart in QHMOS

8 Confirm Order:

The following diagram demonstrates the interaction between application or customer in order to place. Customer can confirm order after selecting hut and items and proceed to check out

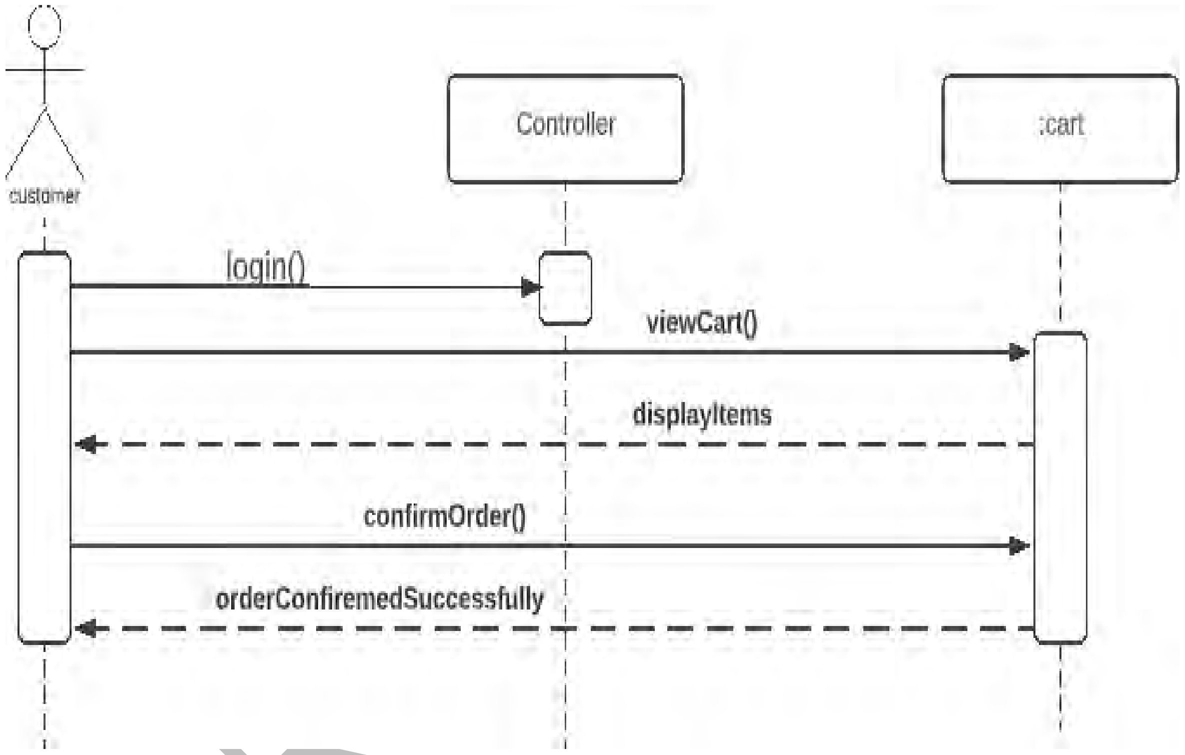


Diagram 35: Sequence Diagram of Confirm Order in QHMOS

9 Place Order:

The following diagram demonstrates the interaction between application or customer in order to place. Customer can confirm order after selecting hut and items and proceed to check out

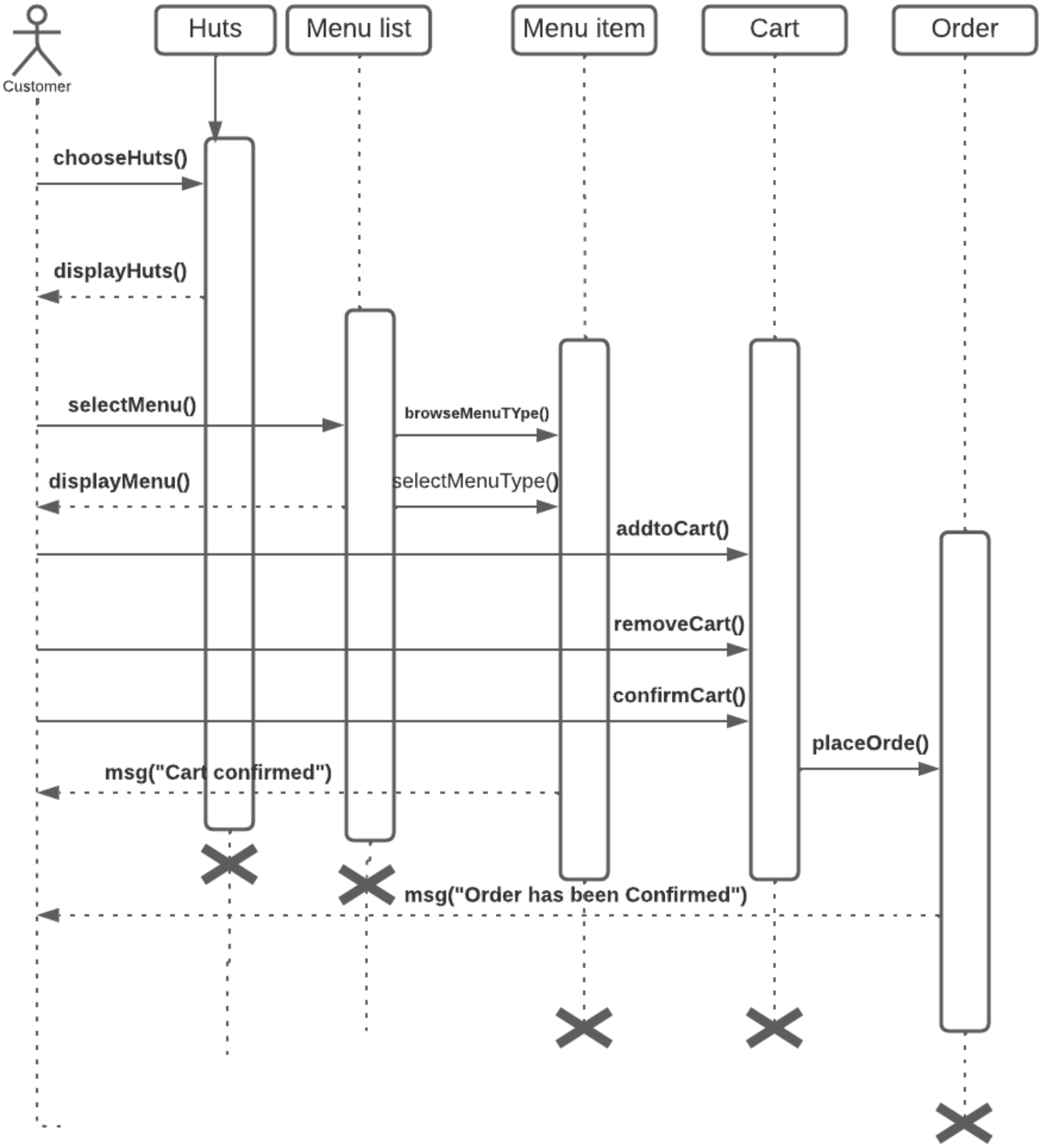


Diagram 36: Sequence Diagram of Place Order in QHMOS

10 Check Out:

The following diagram demonstrates the interaction between application or customer in order to Check out. Customer can place order and proceed to check out to make payment and place order

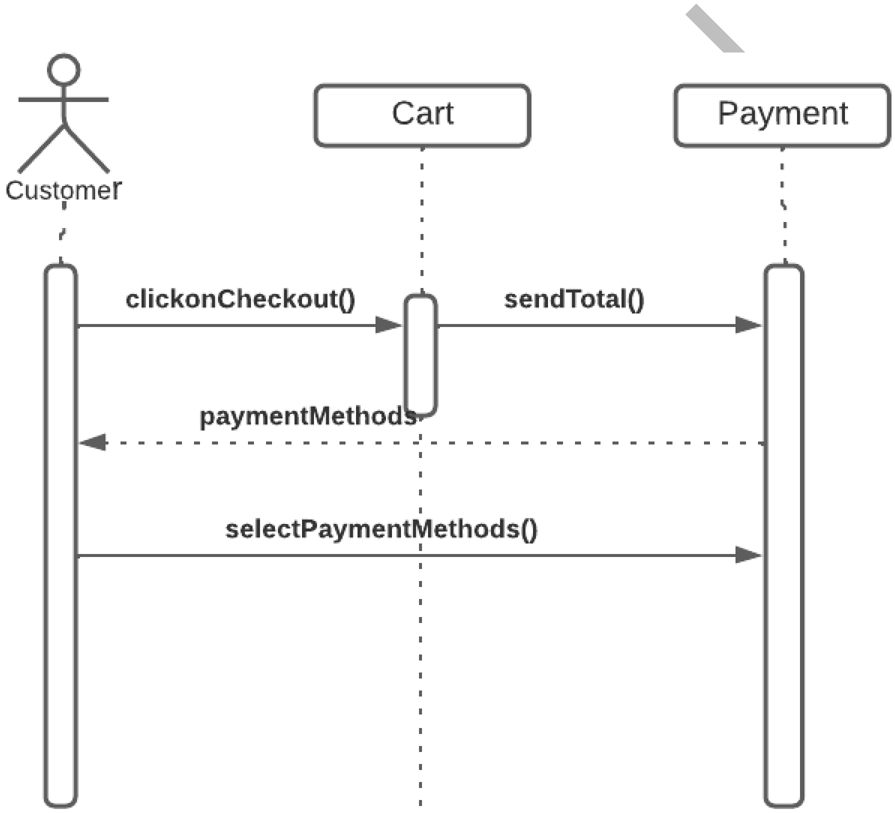


Diagram 37: Sequence Diagram of check out in QHMOS

11 Update Menu

The following diagram demonstrates the interaction between application or customer in order to Update menu. Admin is allowed to update menu list and system display updated menu list

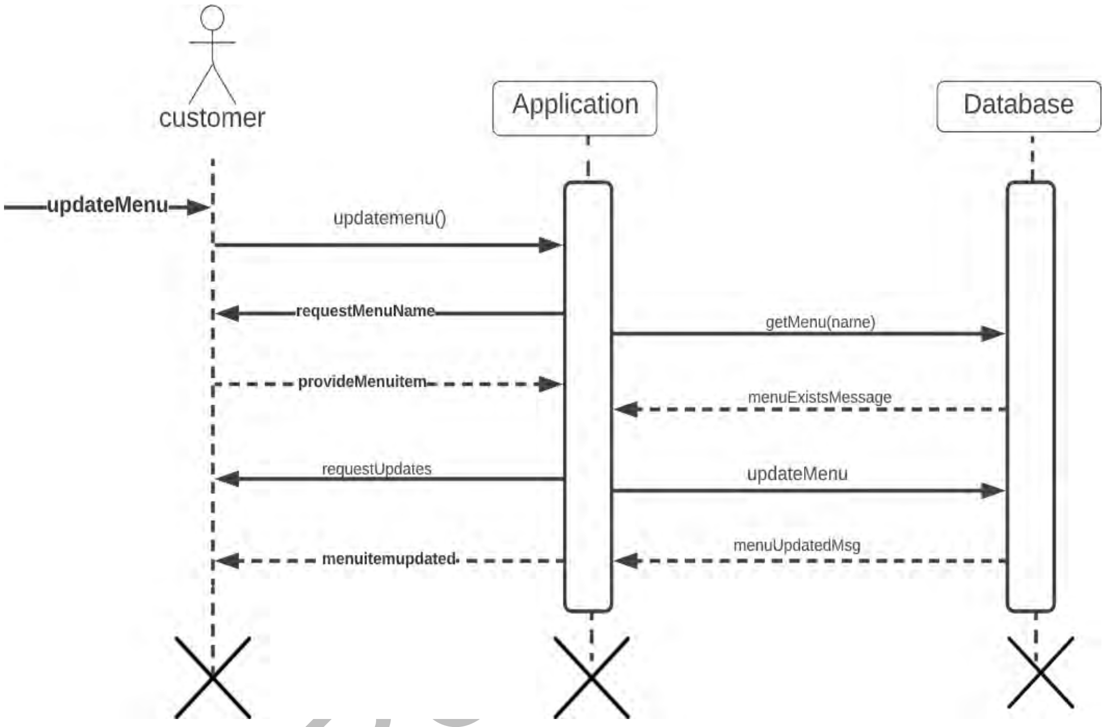


Diagram 38: Sequence Diagram of Update menu in QHMOS

12 Provide Bill

The following diagram demonstrates the interaction between application and customer in order to provide Bill. Bill of menu items are calculated and display to customer.

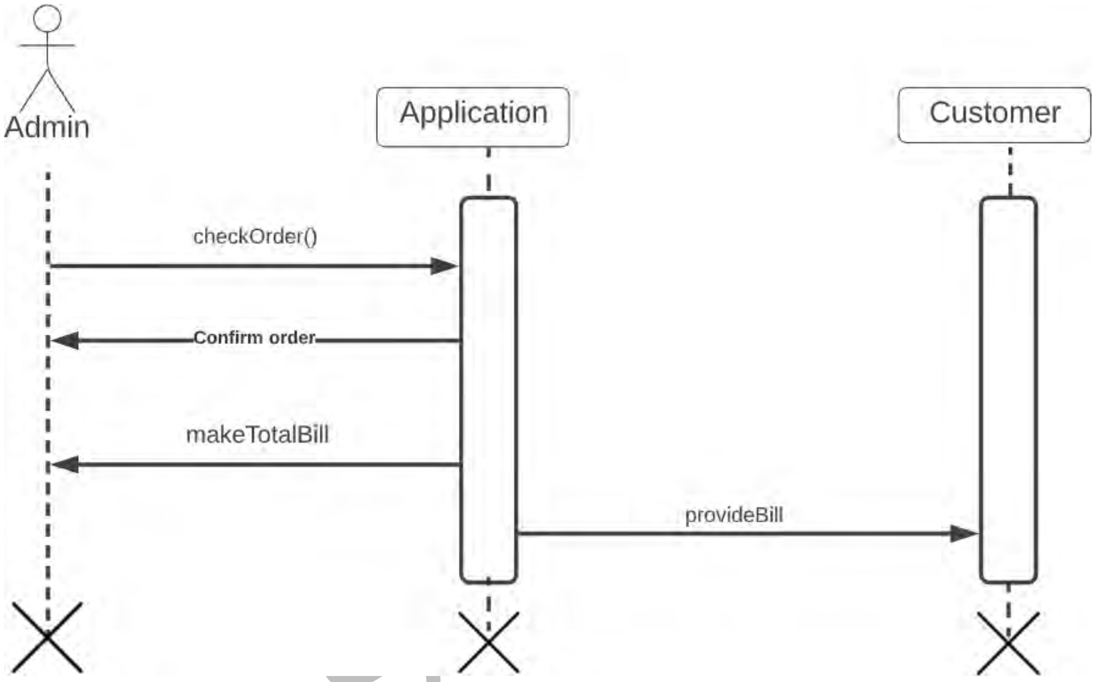


Diagram 39: Sequence Diagram of Provide Bill in QHMOS

13 Receive Payment

The following diagram demonstrates the interaction between application or customer in order to Receive payment Admin receive payment after successfully delivered order to customer

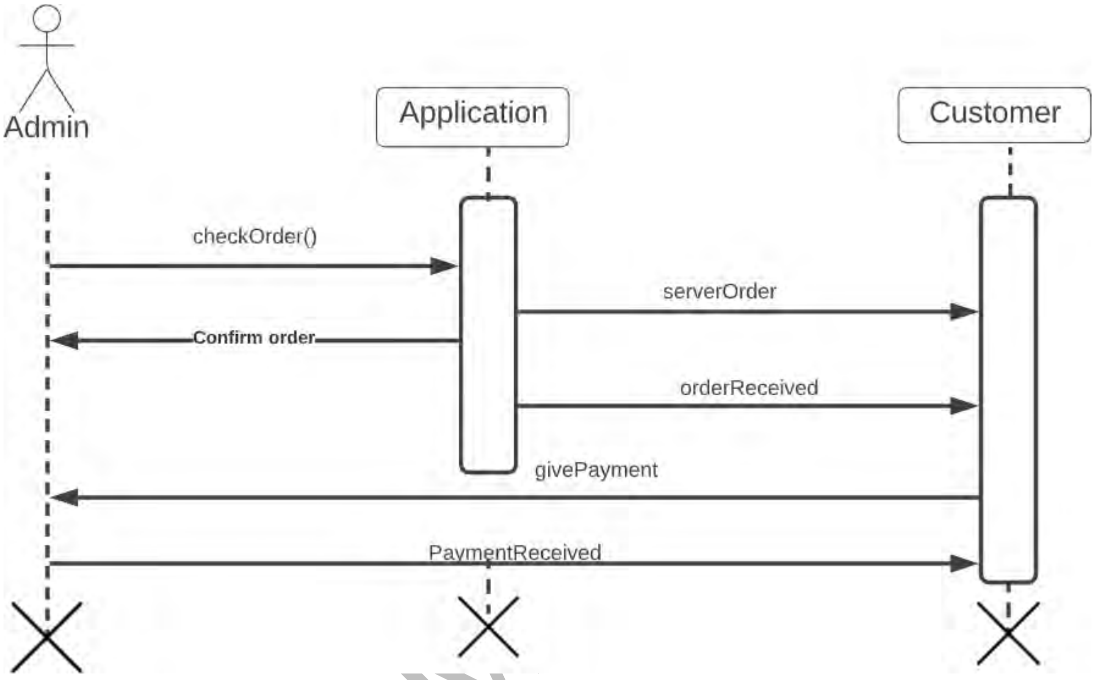


Diagram 40: Sequence Diagram of Receive Payment in QHMOS

3.4 Class Diagram

This is the representation of actual software classes of the system. A class diagram represents both, the structural features i.e. attributes and the behavioral features i.e. functions of the software classes. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

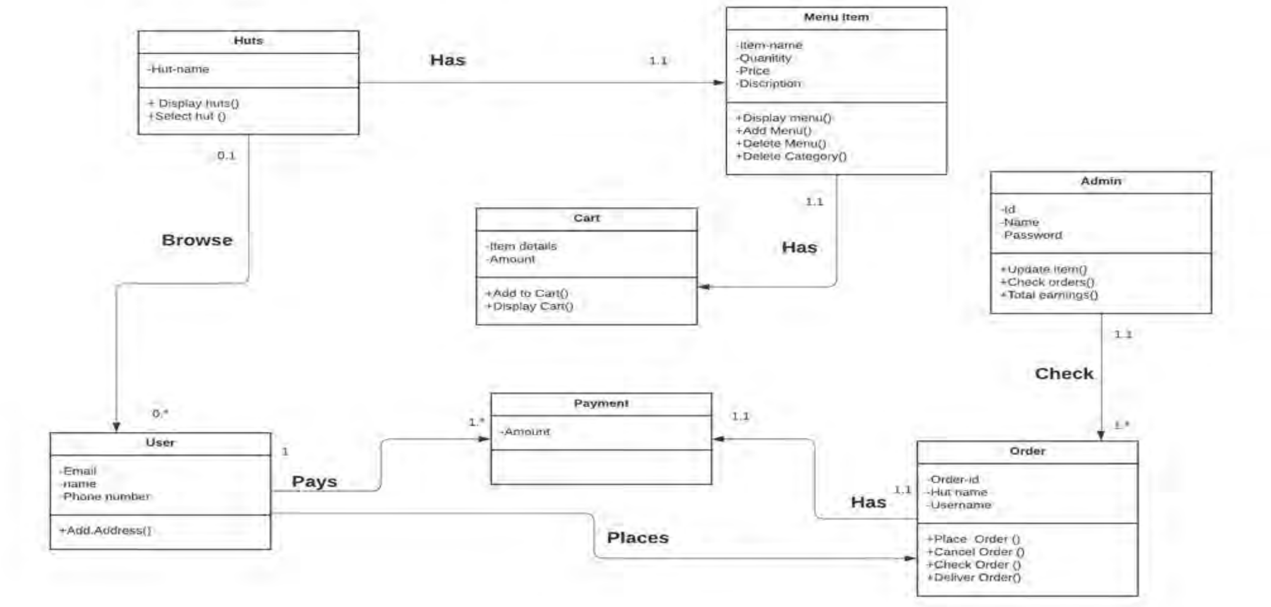


Diagram 41: Class Diagram of QHMOS

3.5 Summary

In this chapter, system design of the QHMOS project is described the flow of data within the system is defined with sequence diagram. The Class Diagram provides the classes & modules to be implemented in project and their functions along with their relations with each other.

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION

Testing is executing a system in order to identify any gaps, errors or missing requirements in contrast to actual requirements. Software test document is a document under which tester will determine whether a system under test satisfies requirements or works correctly or not. Testing analyzes a software item and evaluates its features.

4.2 Programming Language Selection

The first step in implementation is to select a suitable language. Suitable means that the language must be flexible enough to support design.

1 Dart

"**Dart**" is a programming language for developing fast applications on any platform which includes coding and applications and designing interfaces.

2 Dart Features

- Open-Source Programming language
- Platform Independent
- Object-Oriented
- Flexible Compilation
- Objects
- Light-weight
- Browser Support
- Easy to learn

4.3 Software Used

There are many software or platforms are available for the development of application. For the implementation of the project "Collaborative Multimedia Content Sharing Portal" **visual studio code** has been selected as platform for coding of the project. Microsoft word has been selected for documentation purpose. **Luicdchart** has been selected for designing diagrams

1 Visual Studio

Visual studio code is a standalone source code editor which support development operations like debugging, task running and code refactoring. Developers used Visual studio code because it is easy to use and allow quick development of applications.

2 Database

Database is used to store and retrieve information and also provide storage. Firebase database is used for this application to store and retrieve multimedia contents. Firebase saved information under collection and also allocate some storage. It allows to store user information, uploaded contents and then retrieve to display to other users.

4.4 User Interface

A user interface is the visual presentation of a produced project, and the user can communicate with the application by user interface through the hardware devices. User interfaces are designed in such a way that a user can do maximum tasks in the minimum number of steps. Following are screen images of the “High fidelity prototype.”

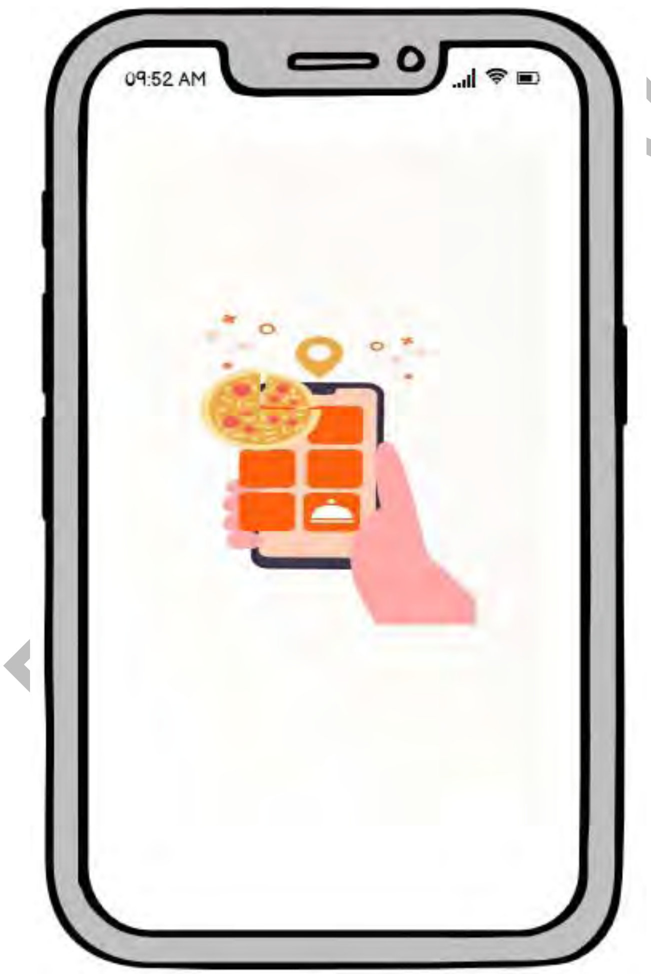


Figure 1: User-App: Flash Screen

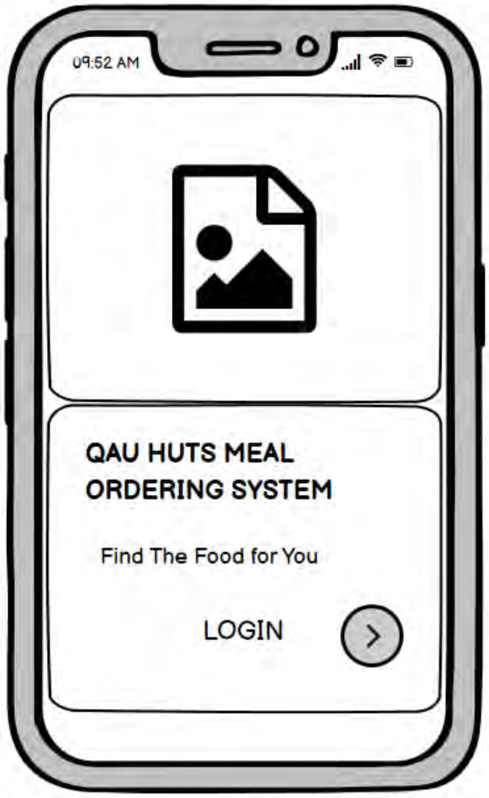


Figure 2: User-App: Main Screen

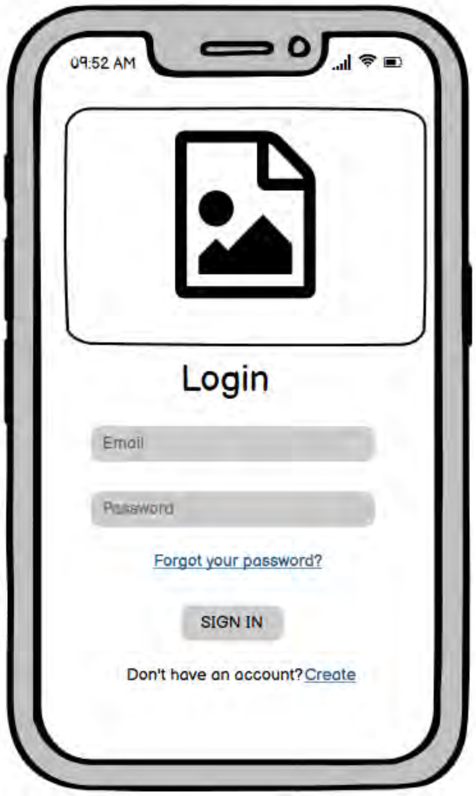


Figure 3: User-App: Login Screen

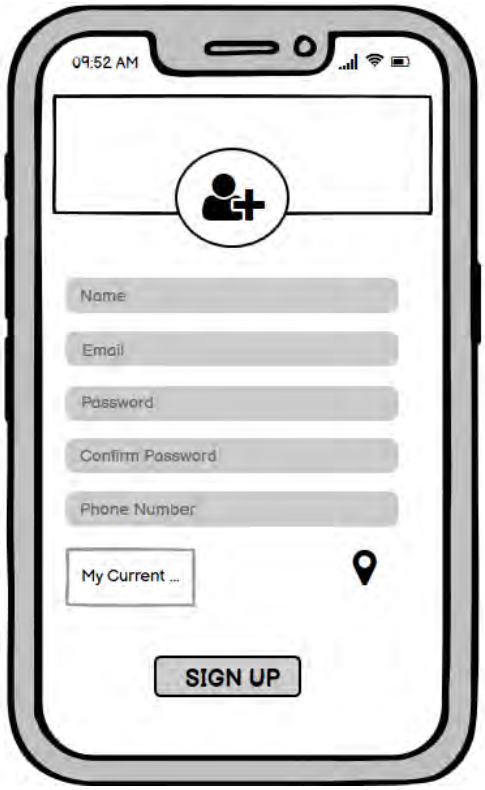


Figure 4: User-App: Sign Up Screen

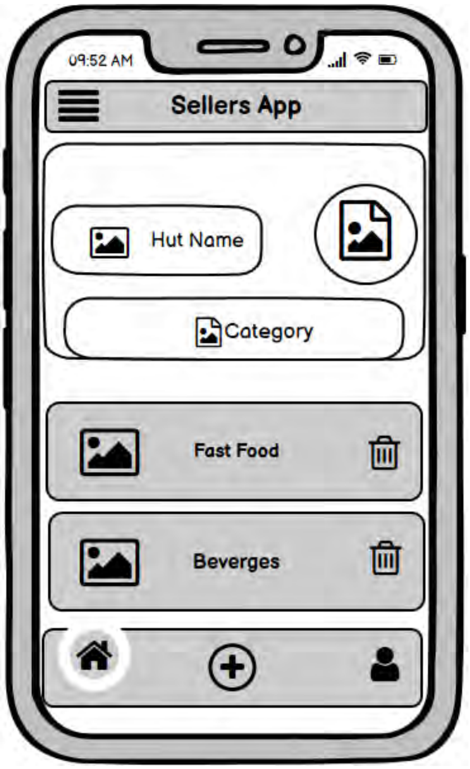


Figure 5: User-App: Home Screen

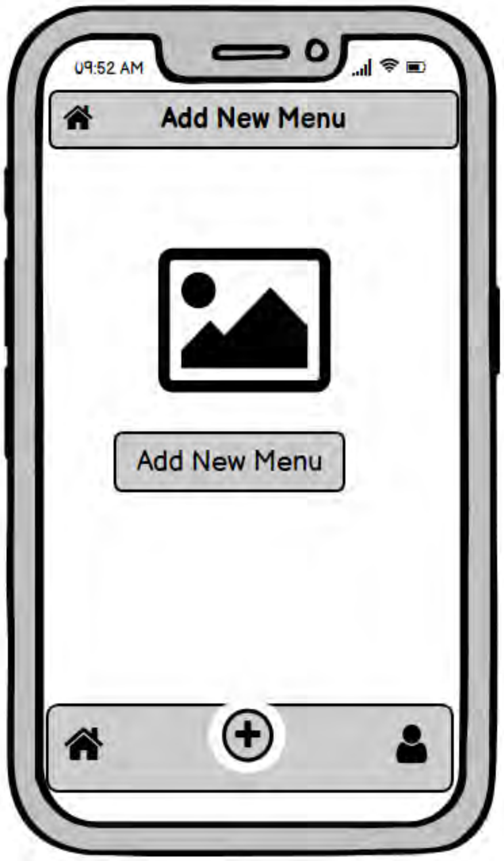


Figure 6: User-App: Add New Menu Screen

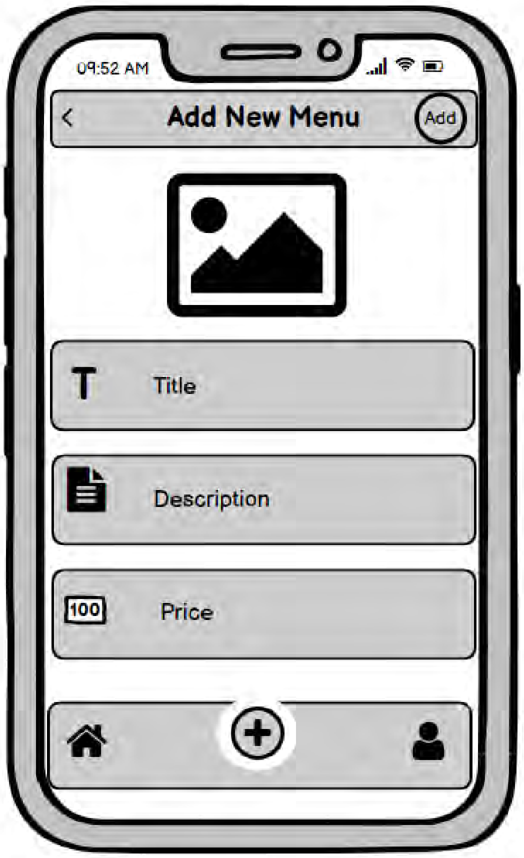


Figure 7: User-App: Add New Menu Item Screen

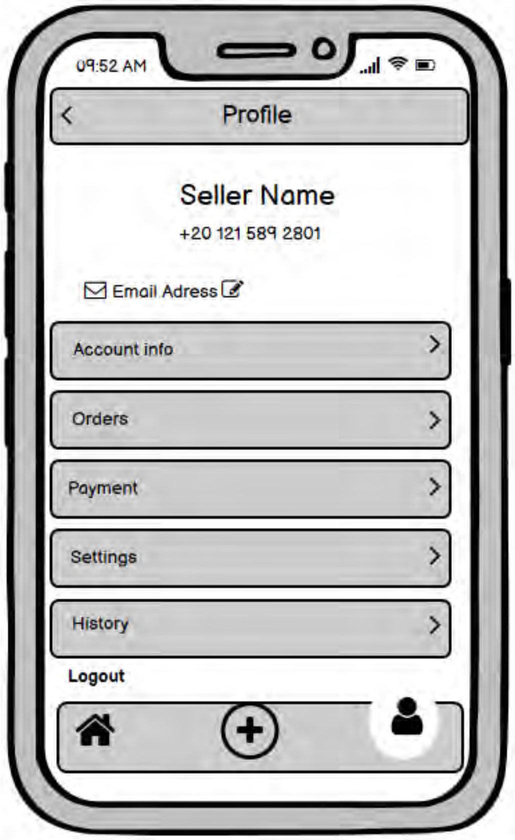


Figure 8:User-App: Profile Screen

Figure 9: Rider-App: Flash Screen

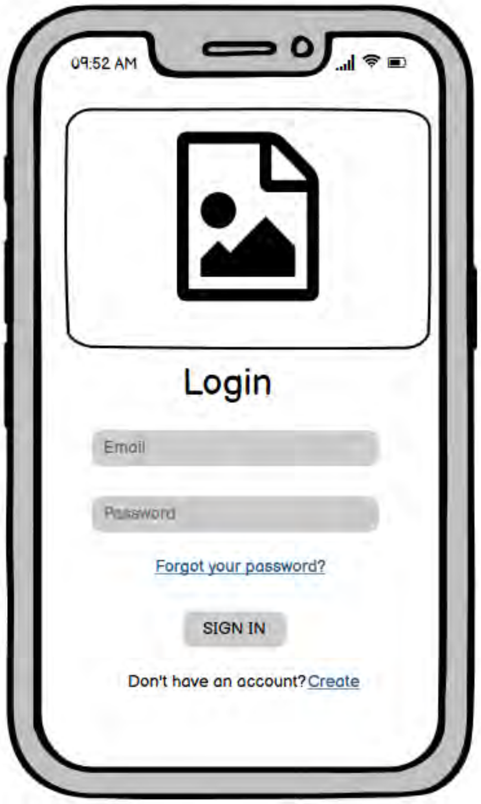


Figure 10: Rider-App: Login Screen

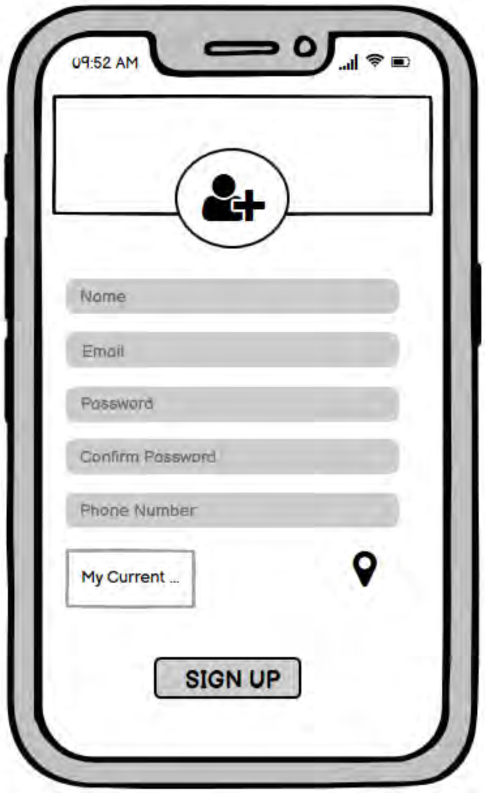


Figure 11:Rider-App: Signup Screen

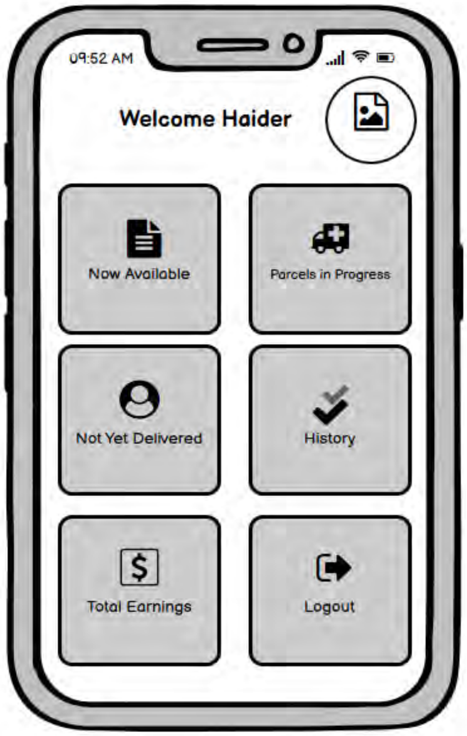


Figure 12: Rider-App: Home Screen

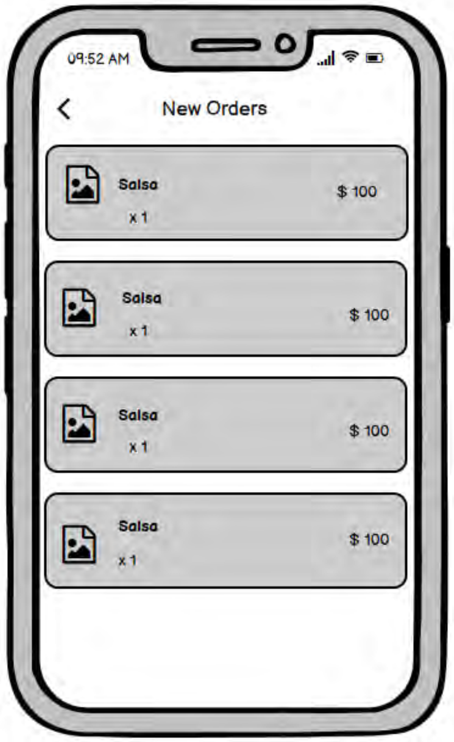


Figure 13:Rider-App: Order Screen

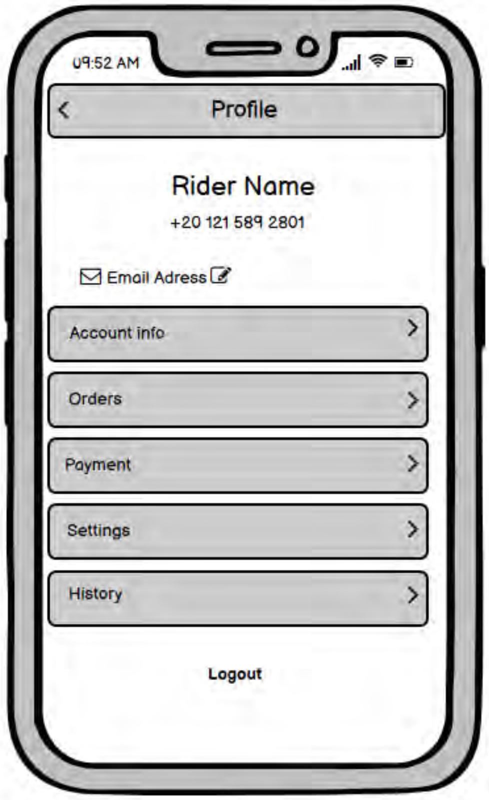


Figure 14:Rider-App: Profile Screen

Figure 15: User-App: Profile Screen

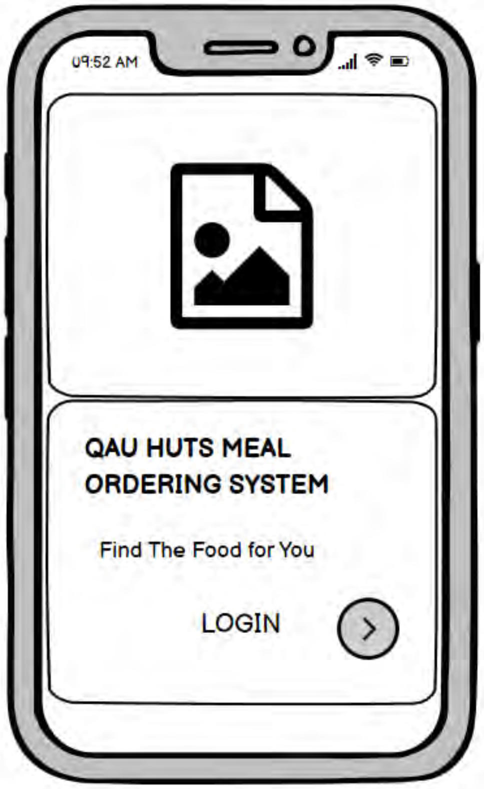


Figure 16: User-App: Main Screen

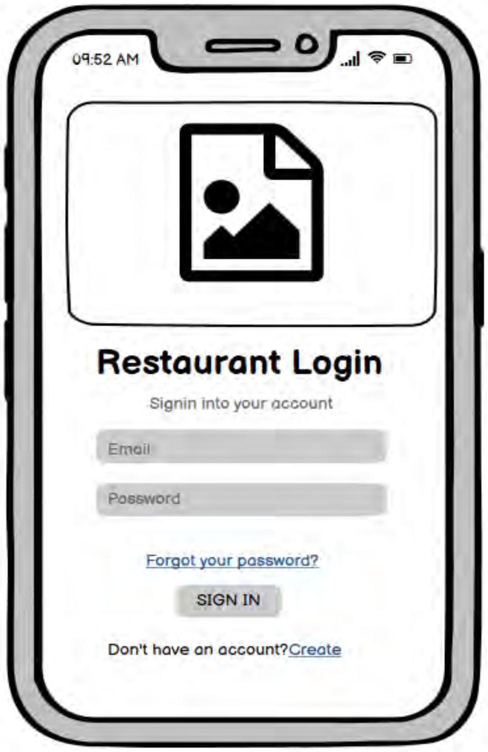


Figure 17:User-App: Login Screen

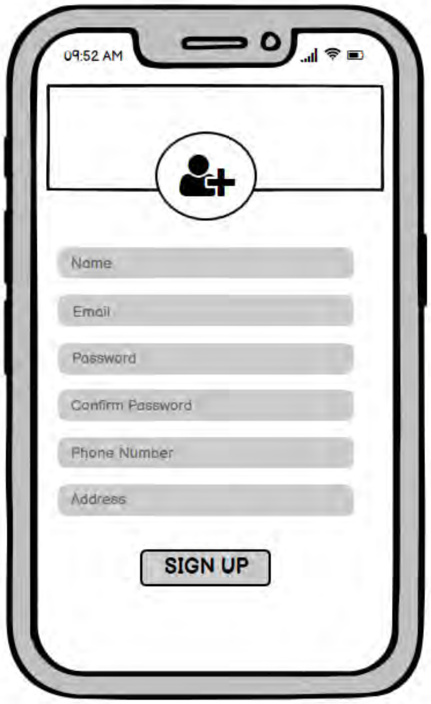


Figure 18: User-App: Signup Screen

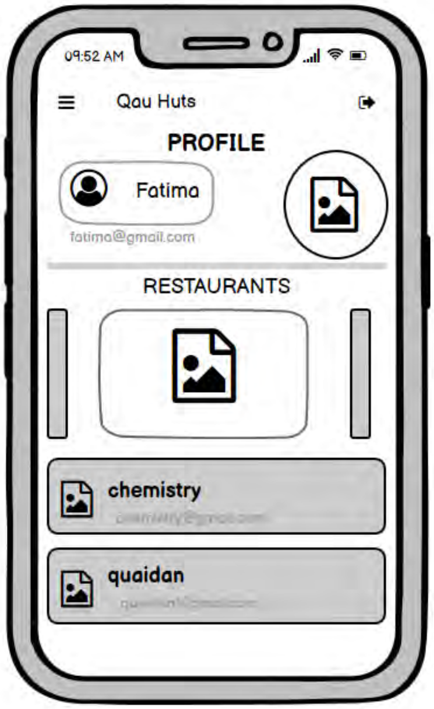


Figure 19: User-App: Home Screen

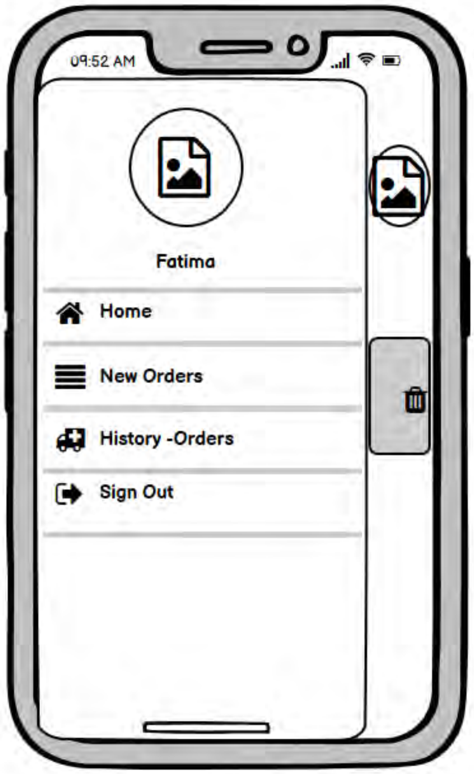


Figure 20: User-App: Hover Screen

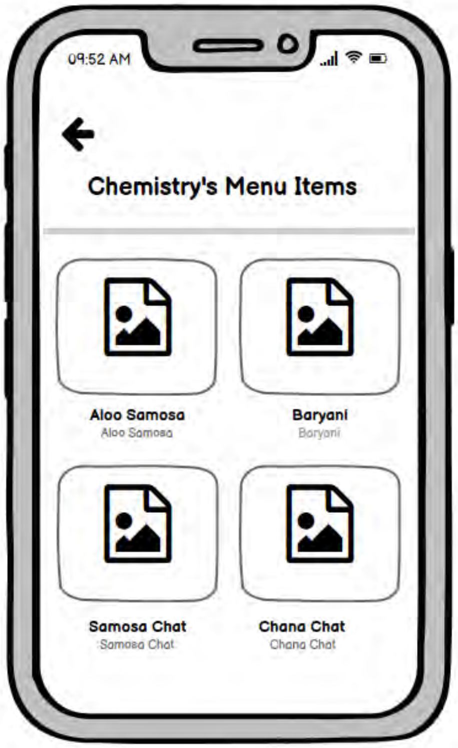


Figure 21: User-App: Hut Menu Screen

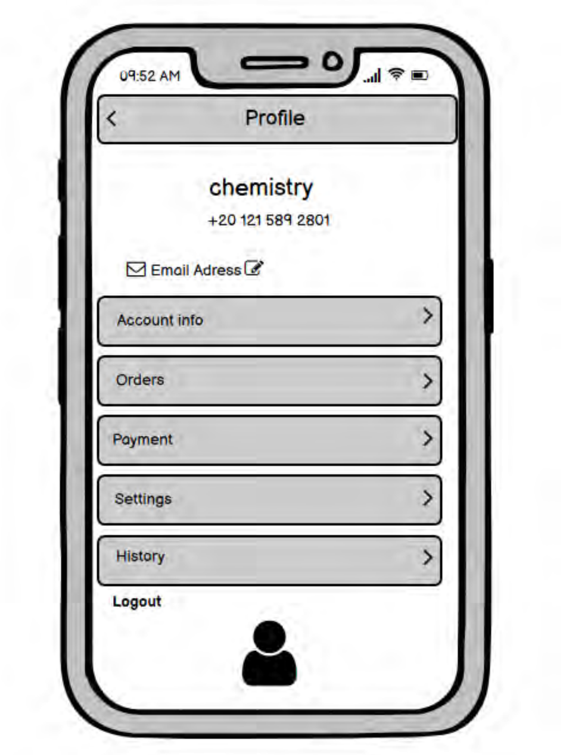


Figure 22: User-App: User Screen

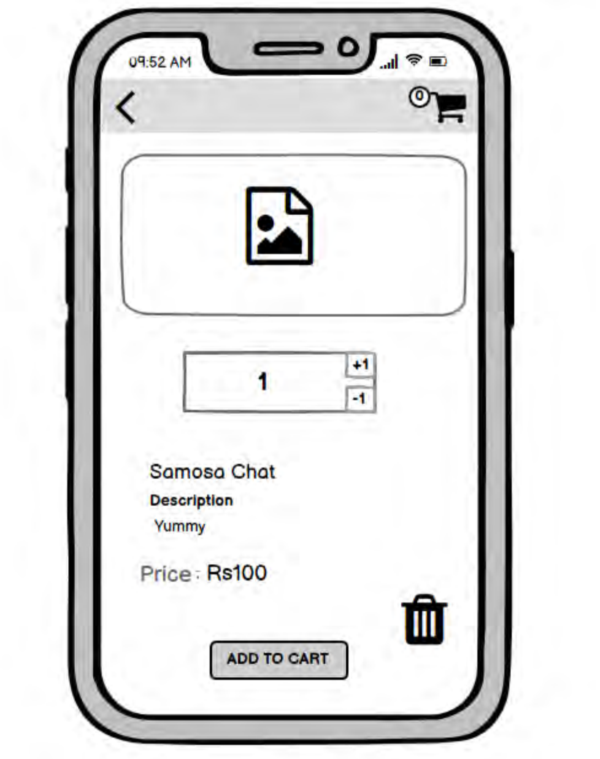


Figure 23: User-App: Add to Cart Screen

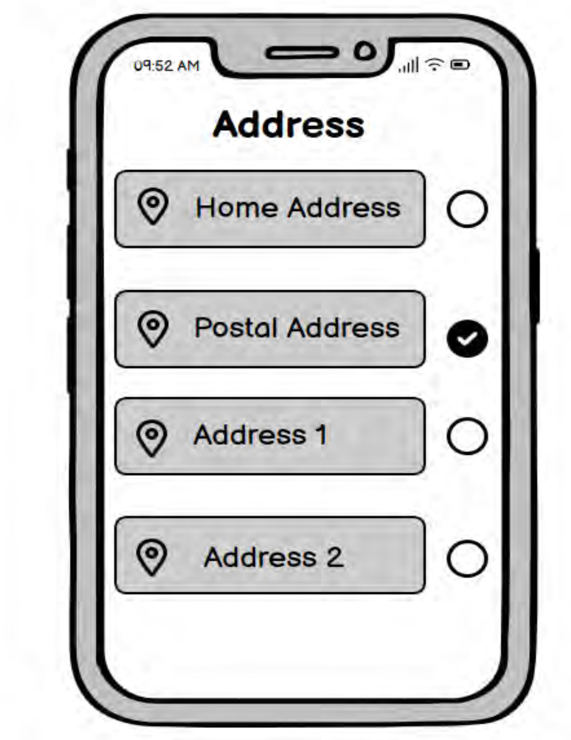


Figure 24: User-App: Add Address Screen

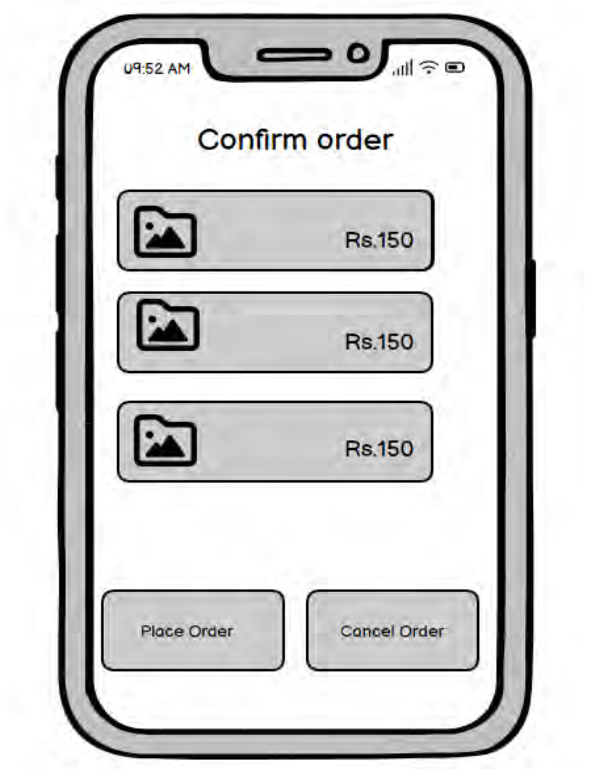
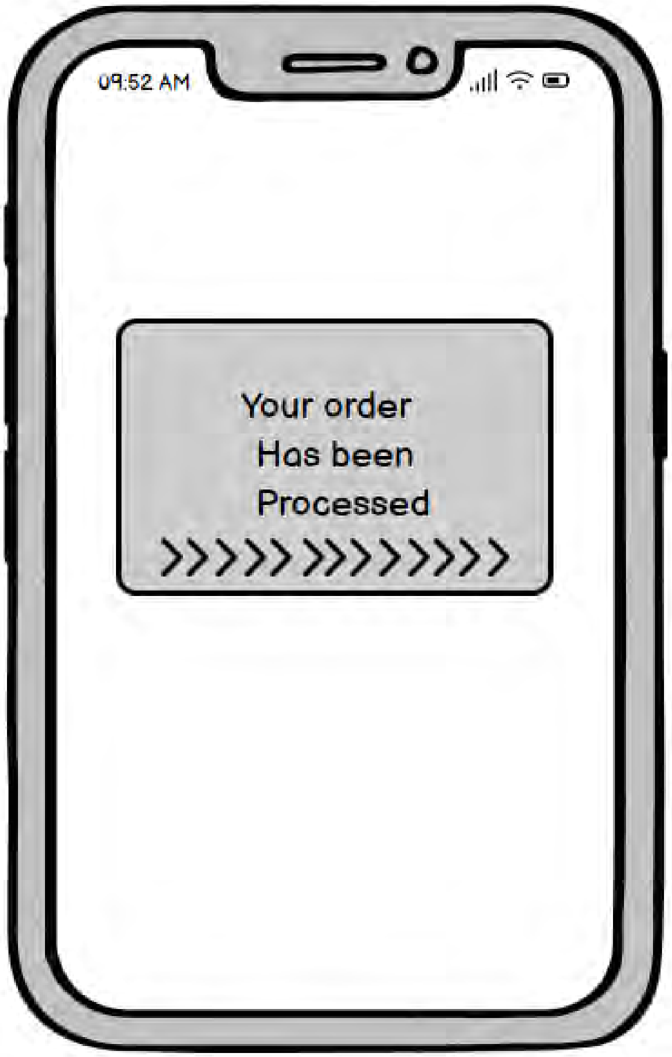


Figure 25: User-App: Confirm Order Screen



Your order
Has been
Processed

Figure 26: User-App: Order confirmation screen

CHAPTER 5

SOFTWARE TEST DOCUMENTATION

5.1 INTRODUCTION

Testing is executing a system in order to identify any gaps, errors or missing requirements in contrast to actual requirements. Software test document is a document under which tester will determine whether a system under test satisfies requirements or works correctly or not. Testing analyzes a software item and evaluates its features.

5.2 System Overview

QHMOS is an android based application The system presents an interactive and up-to-date menu with all available options in an easy-to-use manner. Customer can choose one or more items to place an order which will land in the **Cart**. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time.

5.3 Test Approach

Two testing techniques are used for this system.

- User Acceptance Testing consists of a process of verifying that a solution works for the user. It is not System Testing (ensuring that software does not crash or meets documented requirements) but rather ensures that the solution will work for the user (i.e., tests that the user accepts the solution).

- Usability testing is a way to see how easy to use a software product by testing it with real users.

5.4 TEST PLANS

5.4.1 Features to be tested

Features to be tested are all according to user perspective. So, manual testing will be performed. Following are the features to be tested

- Login
- Register User
- Browse Huts
- Select Menu Items
- Remove Menu Items

5.5 Test Cases

A test case describes an input, action, or an event and an expected response, to determine if a feature of a software application is working correctly.

1 TC-1 Login

Id	**TC-1**
Actor	User
Description	User Login to the system
Setup	Authenticate user from firebase Home screen displayed to user
Instruction	1. User enters email “sumbalbatool56@gmail.com”. 2. Enters password “asghar56”. 3. Click on Login button.
Expected Result	User should login and Home screen will be displayed
Actual Result	User login successfully and Home screen displayed to user.
Verdict	Pass

Table 27 Test case for UC: Login

2 TC-2 Register User

Id	**TC-2**
Actor	User
Description	User wants to register himself/herself
Setup	User install application User fill registration form
Instruction	**1.** User enter username “Sumbal” into input text field **2.** User enter email “areej@gmail.com” into input text field **3.** User enter password **4.** User fill confirm password field **5.** User enter phone number **6.** User enter address. **7.** User click on “Sign Up” button
Expected Result	User’s account created into firebase
Actual Result	Account created successfully
Verdict	Pass

Table 28: Test case for UC: Register User

3 TC-3 Browse Huts

Id	**TC-3**
Actor	User
Description	User Browse different Huts by name.
Setup	Available Hut are displayed.
Instruction	**1.** User can browse different Huts. **2.** User can search Huts typing name of Hut into search bar **3.** User click on “Search” button.
Expected Result	Search Huts will be shown to user into Home Screen.
Actual Result	Search Huts will be shown to user into Home Screen.
Verdict	Pass

Table 29: Test case for UC: Browse Huts

4 TC-4 Select Menu Items

Id	**TC-4**
Actor	Customer
Description	Customer select menu items from menu list.
Setup	Selected items are displayed.
Instruction	1. Customer can select Huts and browse menu items. 2. Customer select menu items from menu list. 3. Customer click on “Confirm order “button
Expected Result	Selected items should add into cart and shown on cart Screen
Actual Result	Selected items are adding into cart shown on cart Screen.
Verdict	Pass

Table 30: Test case for UC: Select Menu items

5 TC-6 Remove Items from cart

Id	**TC-6**
Actor	Customer
Description	Customer remove items from cart.
Setup	Remove items are displayed.
Instruction	1. Customer can select items from menu list. 2. Customer can Remove items from cart. 3. Customer click on “remove item “button.
Expected Result	Selected items are removed from cart.
Actual Result	Selected items are removed from cart.
Verdict	Pass

Table 31: Test case for UC: Remove items from cart

Chapter 6

Conclusion and Future Enhancements

6.1 Introduction

This section contains an overall summary of the entire system behavior and the functionalities the system is offering which are briefly described in all previous sections and holds a description of what new advancements can be made in this field in the future.

6.2 Conclusion

QHMOS allow users to select the food according to their interest. Users can easily search for the desired food from the available huts and can do the order of these food with only just one click so the hazard of going individually to the huts has been decreased.

6.3 Future Enhancements

In the future application can be enhanced.

- As this application is developed only for “Quaid-I-Azam University” it can be developed for the whole country restaurant.
- Increase the Number of sources.
- Different clustering algorithms can be used.
- System Accuracy and complexity can also be checked in usability evaluation.

REFERENCES

[1] Firebase: https://firebase.flutter.dev/docs/overview

[2] Visual Studio Code: https://code.visualstudio.com/docs

[3] LucidChart: https://www.lucidchart.com/

[4] Rogar S. (2020). Pressman. Software Engineering: Apratitioner's Approach. McGraw-Hill Education,2020 ISBN:1260423298,9781260423298

[5] Larman, C. (2004). Applying UML and Patterns: An Introduction to Object- Oriented Analysis and Design and Iterative Development (3rd Edition). Upper Saddle River, NJ, USA: Prentice Hall PTR. ISBN: 01314