



MULTIMEDIA UNIVERSITY

Cyberjaya Campus
Faculty of Computing and Informatics

Trimester 1 2019/2020

TSE 2101 – Software Engineering Fundamentals



STAYIESTA

Homestay Management System

Developed by **ArrobaInc.**

—

Developers:
Ahmad Muzaffar bin Abdul Rahim – *Project Leader*
Wan Izz Naeem bin Wan Ismail – *Lead Programmer*
Muhammad Faris bin Abdul Bashir – *Head of Quality Assurance*
Abdul Malik Ridhwan bin Adam Malek – *Head of Documentation*

Tutorial Section: TT07
Lecturer Name: Chua Fang Fang
Submission Date: 2nd October 2019

Copyright © 2019 ArrobaInc.

TABLE OF CONTENTS

1.	Acknowledgements.....	1
2.	Problem Statements	2
3.	User Stories.....	3
4.	Process Model Chosen.....	4
5.	Functional Requirements	5
6.	Use Case Diagrams.....	6
7.	Use Case Description Tables.....	8
8.	Flowcharts	20
9.	Business Rules	40
10.	Sequence Diagram	41
11.	Class Diagram	49
12.	ER Diagram	50
13.	Software Architectural Diagram.....	51
14.	Application Architectural Diagram	52
15.	Component Diagram.....	53
16.	Deployment Diagram	54
17.	Interface Design.....	55
18.	Quality Attributes.....	68
19.	Data Dictionary	69
20.	Test cases	75

1. Acknowledgements

We would like to extend our gratitude and thank you to Dr. Chua Fang Fang for giving us assistance and support throughout our times in developing this project. This is a homestay management system that will ease a homestay company in managing the homestays that they own. This is also a software that will be the medium between hosts and customers.

Thousands apology for any inconvenience caused throughout the making of this software and if our deliverables did not meet one's expectation. This is **Stayiesta Version 1** and we hope we can develop this program in a better manner in future.

2. Problem Statements

- a. Fake details**

There is no guarantee that the details of a homestay uploaded by a host is legit.

- b. Security risk**

Transactions system between hosts and customers to be secure so that both parties are satisfied.

- c. Outdated info**

Information about a homestay should be updated regularly by the host so that customers can get the latest information about their homestay.

- d. Last minute cancellation**

No options are given for customers if their booking got cancelled by the host.

- e. Loyal rewards**

No rewards are provided for a loyal customer.

3. User Stories

- As a user, I want to make sure that the details provided by the host are legitimate so that I won't be cheated.
- As a developer, I want the transactions between hosts and customers to be secure so that both parties satisfied.
- As a tech support, I want to make sure that all hosts update their homestay information regularly so that customers can get the latest information about their homestay.
- As a user, I want to be given options of available home-stays so that I receive a form of protection in the case that I am left stranded after last minute cancellation by host.
- As investors, I want to provide benefits and rewards for my customers so that we can retain the loyal customers.

4. Process Model Chosen

- a. Type of Process Patterns**

We are using stage pattern because it focuses to a problem associated with each phase. Thus, we can directly detect them and fix the problem right away.

- b. Prescriptive Process Model Being Chose**

We pick Incremental Process Model as a guideline to accomplish our tasks because this model is more flexible and require less cost to change scope and requirements.

- c. Specialized Process Model**

We add unified process to our framework as it needs less time for integration as the process of integration goes on throughout the development process.

5. Functional Requirements

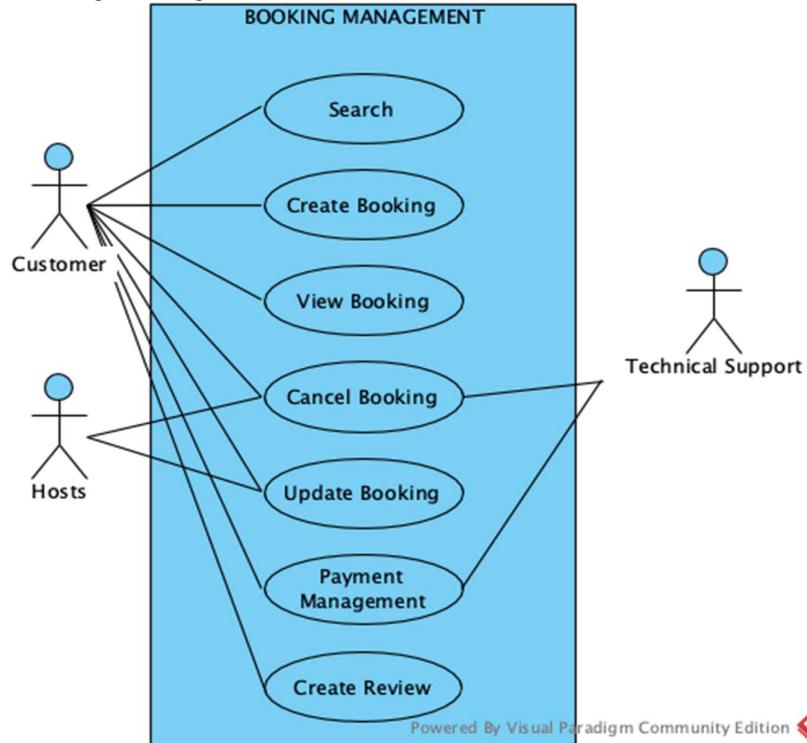
- a. The system Stayiesta shall allow users to search for available homestay.
- b. The system Stayiesta shall allow user to create booking for homestay.
- c. The system Stayiesta shall allow user to view booking details.
- d. The system Stayiesta shall allow user to update their booking details.
- e. The system Stayiesta shall allow user to cancel their bookings.
- f. The system Stayiesta shall allow host to cancel user's booking for homestay.
- g. The system Stayiesta may provide homestay alternatives for customers if hosts cancel booking at the last minute.
- h. The system Stayiesta shall allow host to verify updated booking details.
- i. The system Stayiesta shall allow users to make payments for their homestay bookings.
- j. The system Stayiesta shall be responsible in verifying payments.
- k. The system Stayiesta shall allow hosts to add their houses@place to stay as homestay.
- l. The system Stayiesta shall allow host to update homestay details.
- m. The system Stayiesta shall allow technical support verify the legitimacy of homestay details based on certain criteria before publishing the details.
- n. The system Stayiesta shall allow technical support to verify updated homestay details.
- o. The system Stayiesta shall allow host to delete their homestay information.
- p. The system Stayiesta shall allow user to update their profile information.
- q. The system Stayiesta shall allow user to delete profile.
- r. The system Stayiesta shall allow user to update profile.
- s. The system Stayiesta shall allow technical support to suspend host accounts.
- t. The system Stayiesta shall be responsible for suspending accounts that are inactive for 12 months.
- u. The system Stayiesta shall allow user to view profile.
- v. The system Stayiesta shall allow user to view rewards page.
- w. The system Stayiesta shall allow technical support to provide loyalty rewards program to regular customers who has good booking record for a month.
- x. The system Stayiesta shall allow users to create an account.
- y. The system Stayiesta shall allow users to log in into the system.
- z. The system Stayiesta shall allow users to log out of the system.
- aa. The system Stayiesta shall allow users to report any related issues by opening a ticket.
- bb. The system Stayiesta shall verify the credentials entered by users.
- cc. The system Stayiesta shall notify host three times max to update their homestay information.

Non-functional requirements

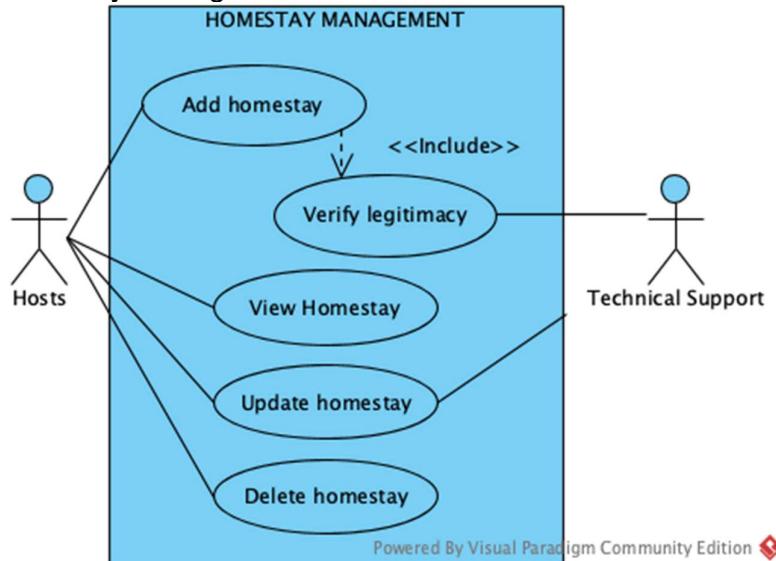
- a. The system Stayiesta should be easy to navigate between pages.
- b. The system Stayiesta should load a page in less than 5 seconds.

6. Use Case Diagrams

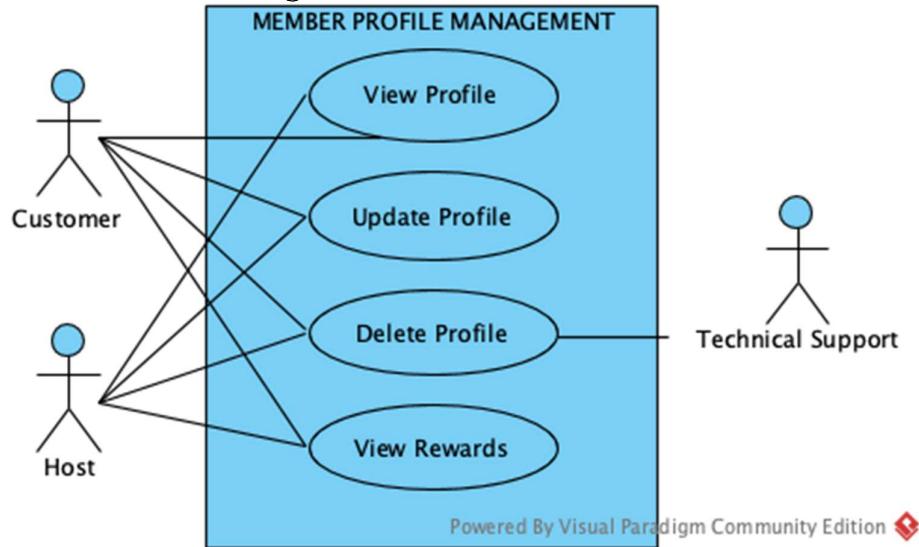
- Booking Management



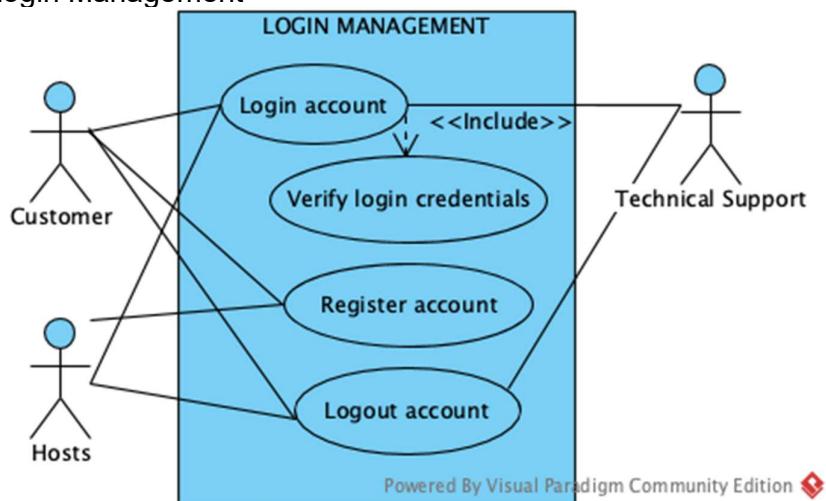
- Homestay Management



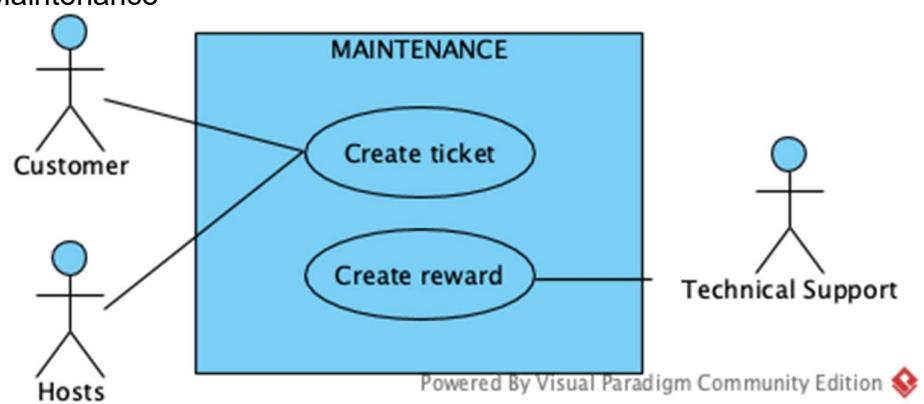
- Member Profile Management



- Login Management



- Maintenance



7. Use Case Description Tables

a. Booking Management

Use Case ID/Name	BM01 / Search
Precondition	User must logged in as customer
Post Condition	List of homestay according to customer's need will be listed.
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all homestay ID and necessary information such as it's location, type, number of bedrooms, number of bathrooms, basic amenities vector of dates of availability for the homestay and review ID with its details. 2. User inputs desired location. 3. User inputs desired date of check in and check out. 4. System displays available homestays. 5. User selects one of the homestays. 6. System displays homestay ID, homestay name, homestay type, number of bedrooms, number of bathrooms, basic amenities vector, review ID, review details and stars. 7. User choose between to proceed with homestay or change homestay.
Alternate Flow	<p>After step 3, the user can choose to narrow the results by filtering them according to filters provided by the system.</p> <p>If there are no homestays available after filtering, the system will display "No homestay available" and will return to the search query again.</p>
Exception Flow	-

Use Case ID/Name	BM02 / Create Booking
Precondition	User has picked their desired homestay
Post Condition	Booking successfully created and added into database
Main Flow	<ol style="list-style-type: none"> 1. After selecting homestay, system stores the selection into a temporary storing area. 2. System retrieve all details on user. 3. System stores details into temporary storing area.

	<ol style="list-style-type: none"> 4. User enters check-in time. 5. System displays summary of booking for user's validation and reference. 6. System displays terms and conditions. 7. User enters option Y or N for validation. 8. System stores all data into database. 9. Booking successfully created.
Alternate Flow	If user enters invalid option, system will loop until it gets valid input.
Exception Flow	-

Use Case ID/Name	BM03/ View Booking
Precondition	User must be logged in as customer
Post Condition	List of successful booking will appear
Main Flow	<ol style="list-style-type: none"> 1. System retrieves booking ID associated with the user. 2. System displays all bookings with details such as its booking ID, homestay ID, homestay name, location, booking date and payment status.
Alternate Flow	If there is no booking exist in the database, system will display "No booking found" and prompts user to return to main page.
Exception Flow	-

Use Case ID/Name	BM04/ Cancel Booking
Precondition	User must be logged in as customer User must have at least one existing booking in database
Post Condition	Chosen booking details will be deleted from the account's database
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all booking ID associated to the account. 2. User enters booking ID to be deleted. 3. User enters his/her password for verification. 4. User enters Y or N for confirmation. 5. System deletes the booking details and display a successful message.

Alternate Flow	<p>Step 2, if user entered invalid booking ID, system will loop until it gets a valid input or user request to cancel operation.</p> <p>Step 3, if user entered invalid password, deletion will be cancel and system will display an error "Password is incorrect. Deletion of homestay will be cancel"</p> <p>Step 4, if user entered invalid option, system will loop until it gets a valid input or user request to cancel operation.</p>
Exception Flow	-

Use Case ID/Name	BM05/ Update Booking
Precondition	User must have at least one successful booking in database
Post Condition	Chosen booking details successfully updated
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all bookings ID associated to account. 2. User enters booking ID to be updated 3. System displays every field with its corresponding value. 4. User enters a new value according to the field. 5. System will store updated booking details in database. 6. System displays message "Booking successfully updated"
Alternate Flow	<p>Step 2, if the user enters an invalid booking ID, system will loop until it gets the right input or user request to cancel operation.</p> <p>Step 3, if user decided not to change the value, system will accept a blank field input and treat it as no change. System will directly move to the next field.</p> <p>Step 3, if user enters invalid input for corresponding field, system will loop until it gets the right input.</p>
Exception Flow	-

Use Case ID/Name	BM06/ Payment Management
Precondition	User must have at least one booking with “pending” payment status
Post Condition	Payment status become “paid” and a successful page will be displayed.
Main Flow	<ol style="list-style-type: none"> 1. User view the list of booking made displayed by system. 2. User enters booking ID which its payment status is false or “pending”. 3. System displays booking invoice for user’s validation and reference. 4. User enters input to agree or disagree with the price and to checkout. 5. System will retrieve reward ID associated to the account. 6. System displays all rewards available associated with the user 7. User enters reward ID available for them. 8. System displays a new booking invoice with discounted price for user’s validation and reference. 9. User enters input to agree or disagree with the price. 10. User enters card number, card expiry date and CVV number. 11. System will generate a 6-digit code 12. User is required to enter the 6-digit code again for verification purposes. (Dummy version of TAC). 13. If user input is correct, payment made is successful.
Alternate Flow	<p>Step 2, if user enters booking ID that has payment status “true”, an error message will appear with message “Booking has been paid”</p> <p>Step 2, if user enters invalid booking ID, system will loop the query until valid booking ID is entered.</p> <p>For every input option, if user entered invalid input, system will loop the query until user enters valid input.</p> <p>Step 7, if user decided not to use any rewards, user can enter a blank field and it will skip to step 10.</p> <p>Step 7, if user entered an invalid reward ID, system will loop</p>

	<p>until it gets a valid input.</p> <p>Step 7, if user entered a reward ID that is not associated with the account, system will display error message “Reward does not exist for user” and asks user to re-enter or cancel the operation (reward operation).</p> <p>Step 10, for every field, if user entered invalid input, system will loop until valid input is entered.</p> <p>Step 12, if user entered wrong 6-digit code, system will automatically cancel the payment and payment is failed.</p>
Exception Flow	-

Use Case ID/Name	BM07/ Create Review
Precondition	User must have at least one booking
Post Condition	Review successfully created
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all booking IDs associated to the account. 2. User enters booking ID to leave a review. 3. System retrieves homestay ID. 4. User enters star for review. 5. User enters review details. 6. System stores data into database.
Alternate Flow	<p>If booking ID entered is invalid, system will loop until user input a valid booking ID.</p> <p>If user entered invalid value for variable star, system will loop until user input a valid booking ID.</p>
Exception Flow	-

b. Homestay Management

Use Case ID/Name	HM01 / Add Homestay
Precondition	User must be logged in as a Host.
Post Condition	Homestay is added
Main Flow	<ol style="list-style-type: none"> 1. User enters all necessary homestay information such as name, type, number of bedrooms, number of bathrooms, price, basic amenities. 2. User submits the application. 3. System adds the homestay into the database. 4. Technical support manually verifies legitimacy of homestay information.
Alternate Flow	If user enters invalid input for any field, the system will loop the query until it gets the valid input.
Exception Flow	-

Use Case ID/Name	HM02 / View Homestay
Precondition	1. User must be logged in as a Host
Post Condition	List of homestays associated to the account will be displayed.
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all homestay ID associated to the account. 2. System will display the list of homestays with its details.
Alternate Flow	If no homestays are registered under the user, the system will display "No homestays are registered under your account." and return to main page.
Exception Flow	-

Use Case ID/Name	HM03 / Update Homestay
Precondition	<ol style="list-style-type: none"> 1. User must be logged in as a Host. 2. There must be at least one homestay added.
Post Condition	Information of the homestay updated.
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all homestay ID associated to the account. 2. User enters homestay ID to be updated. 3. System will display every field with its current value. 4. User enters a new value according to field. 5. System store the updated details into the database. 6. System displays "Homestay updated successfully"
Alternate Flow	<p>Step 2, if the user entered an invalid homestay ID, system will loop until it gets a valid ID or user request to cancel operation.</p> <p>Step 3, if the user decided not to edit the current value in the field, the user can enter a blank input and system will ignore the value entered.</p>
Exception Flow	-

Use Case ID/Name	HM04 / Delete Homestay
Precondition	<ol style="list-style-type: none"> 1. User must be logged in as a Host 2. Homestay information must exist.
Post Condition	A successful page will display with message "Homestay deleted successfully".
Main Flow	<ol style="list-style-type: none"> 1. System retrieves all homestay ID associated to the account. 2. System displays all homestay information in the host account. 3. User enters homestay ID of the homestay to be deleted. 4. System delete the homestay.
Alternate Flow	If user enters invalid homestay ID, the system will display error message and prompt the user to either cancel operation or re-enter homestay ID.
Exception Flow	-

c. Member Profile Management

Use Case ID/Name	MPM01 / View Profile
Precondition	User must be logged in as any type of accounts
Post Condition	Details of the user will be listed down
Main Flow	<ol style="list-style-type: none"> 1. System retrieves necessary information of that user ID from database. 2. System displays user ID, username, full name, IC number, birthdate, email, address, and phone number.
Alternate Flow	-
Exception Flow	-

Use Case ID/Name	MPM02 / Update Profile
Precondition	User must be logged in as any type of account
Post Condition	User's profile updated
Main Flow	<ol style="list-style-type: none"> 1. System retrieves current details of account from database. 2. User input new detail accordingly 3. System saves the updated data in database
Alternate Flow	If user decided to skip a field, user can just leave the field blank and enter so data will not change.
Exception Flow	

Use Case ID/Name	MPM03 / Delete Profile
Precondition	User must be logged in as any type of accounts
Post Condition	User's account deleted and system return to login page.
Main Flow	<ol style="list-style-type: none"> 1. System retrieves current details of account. 2. User input password for verification. 3. User input Y or N when system requested for confirmation.

Alternate Flow	If user input password incorrectly, system will loop until user decided to cancel deletion.
Exception Flow	-

Use Case ID/Name	MPM04 / View Rewards
Precondition	User must be logged in as customers account
Post Condition	Rewards offered for users displayed accordingly
Main Flow	<ol style="list-style-type: none"> 1. System retrieves necessary information from the account like reward ID, reward name, reward type, reward value and reward details. 2. System displays rewards associated with account.
Alternate Flow	If there is no rewards associated to the account, system will display "No rewards available" and user will prompt to leave the page.
Exception Flow	

d. Login Management

Use Case ID/Name	LM01 /Login account
Precondition	User must have a registered account.
Post Condition	User is logged in.
Main Flow	<ol style="list-style-type: none"> 1. User entered the username and password. 2. System verifies the login credentials entered by user. 3. If username exist and password matches with the username, access is granted and user enters the system.
Alternate Flow	<ol style="list-style-type: none"> 1. If the username or password is invalid, the system prompts the user to try again or cancel the operation.
Exception Flow	-

Use Case ID/Name	LM02 / Register account
Precondition	User must exist.
Post Condition	Account created.
Main Flow	<ol style="list-style-type: none"> 1. User chooses between Customer account or Host account. 2. User enters preferred username and password. 3. User enters information such as name, NRIC, birth date, mailing address, phone number, e-mail. 4. User submits application 5. System will verify and directly create the account.
Alternate Flow	<ol style="list-style-type: none"> 1. If the username exists, the system prompts the user to enter another username or cancel the operation. 2. If the user enters the wrong input at [1], the system loop at the same query until valid input is entered. 3. If user enters invalid input for any information requested, the system will loop the same query until it fulfills the requirements for each query.
Exception Flow	-

Use Case ID/Name	LM03 / Logout account
Precondition	User is logged in.
Post Condition	User is logged out.
Main Flow	<ol style="list-style-type: none"> 1. User selects the logout option. 2. System logs out the user and return to the login page.
Alternate Flow	-
Exception Flow	-

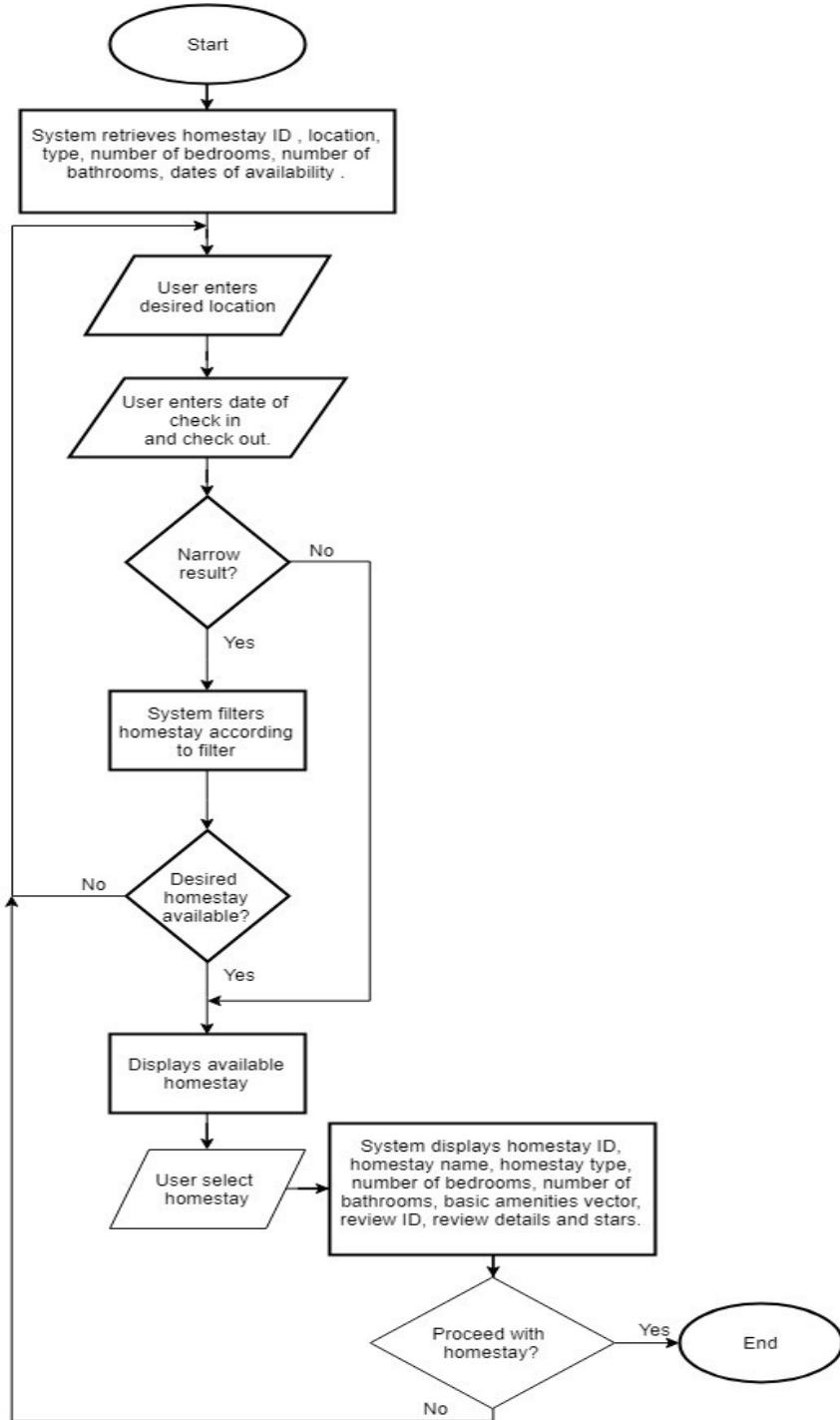
e. Maintenance

Use Case ID/Name	M01 / Create Ticket
Precondition	User must be logged in as hosts or customers
Post Condition	A successful page will appear if ticket submitted successfully.
Main Flow	<ol style="list-style-type: none"> 1. System retrieve account information like user ID, name, e-mail address, account type 2. User enters ticket type. 3. User enter ticket details. 4. User submit the ticket.
Alternate Flow	If user input invalid option for ticket type, system will keep on looping until user input the right option.
Exception Flow	-

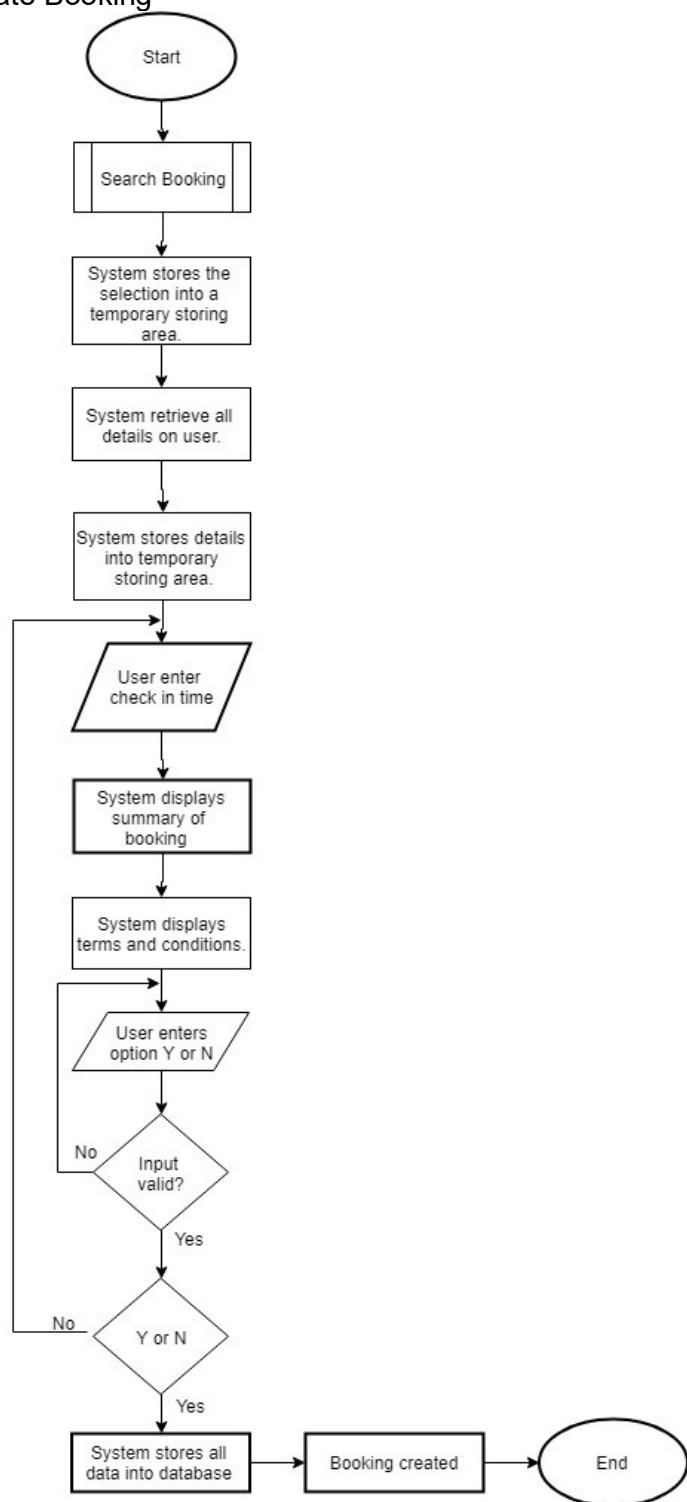
Use Case ID/Name	M02 / Create Reward
Precondition	User must be logged in as technical support
Post Condition	A successful page will appear if reward submitted successfully and a reward ID will be generated.
Main Flow	<ol style="list-style-type: none"> 1. System will retrieve account information like user ID, name, e-mail address, account type. 2. User pick reward type either discount or price-off. 3. User enters value for reward. 4. User enters details of reward 5. User submits reward 6. System will verify and create the reward.
Alternate Flow	If information entered incorrectly, user will be prompted to re-enter the information again
Exception Flow	-

8. Flowcharts

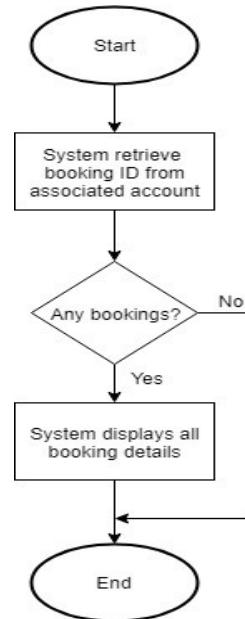
a. Booking Management BM01 – Search



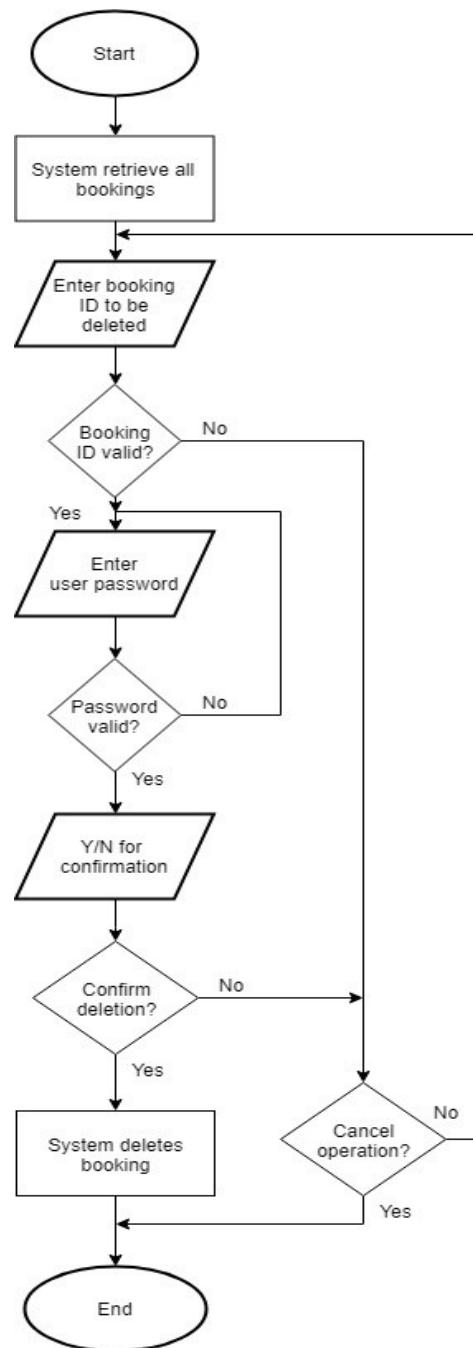
BM02 – Create Booking



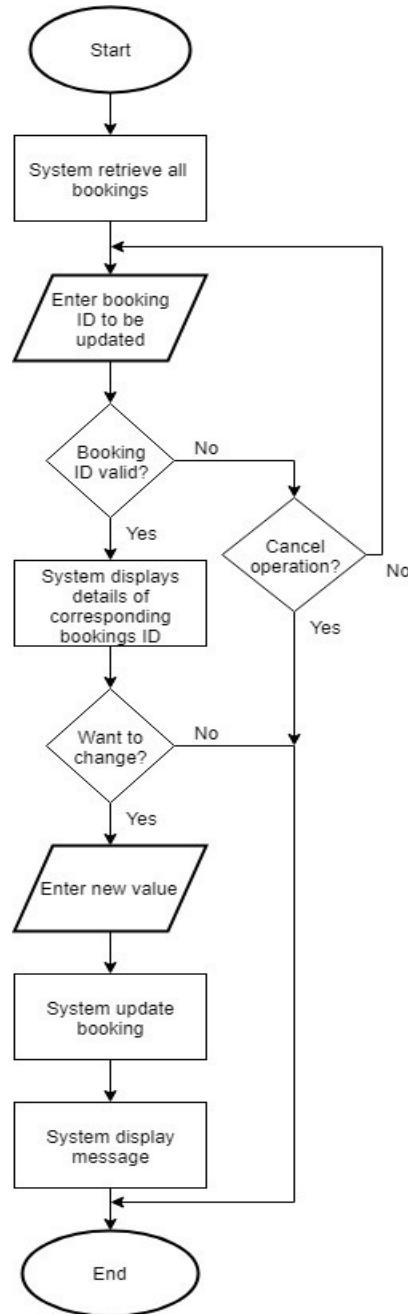
BM03 – View Booking



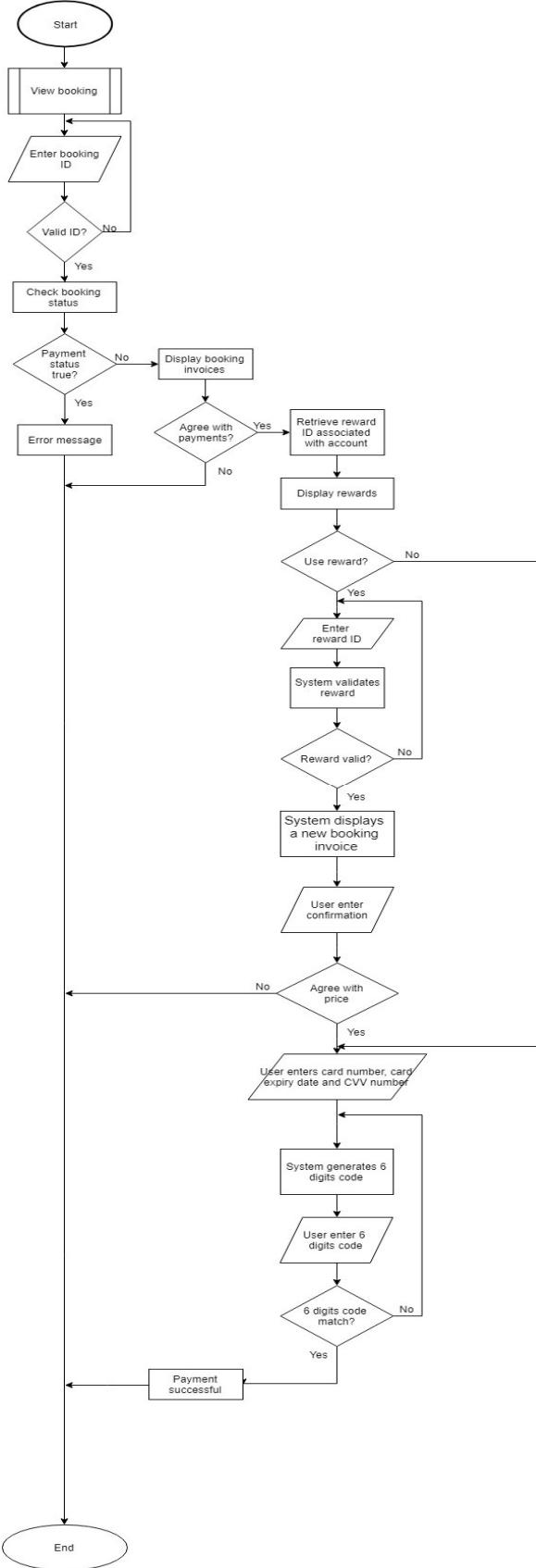
BM04 – Cancel Booking



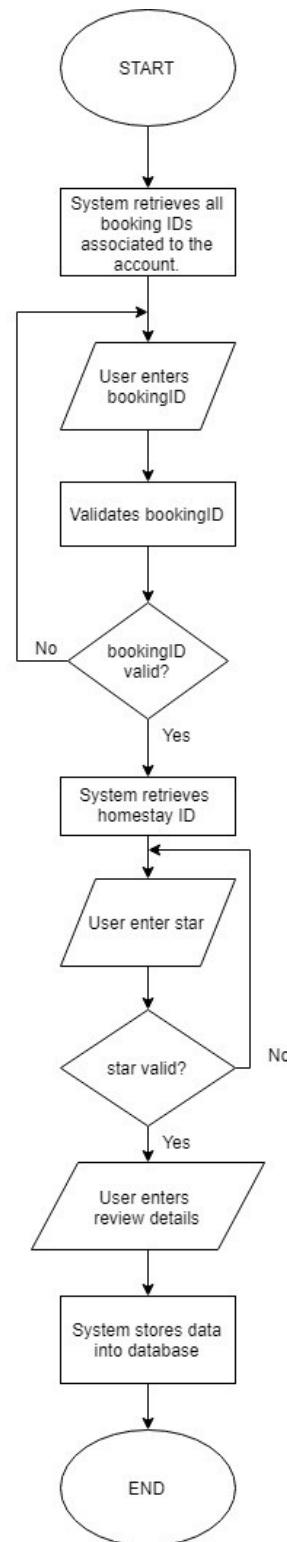
BM05 – Update Booking



BM06 – Payment Management



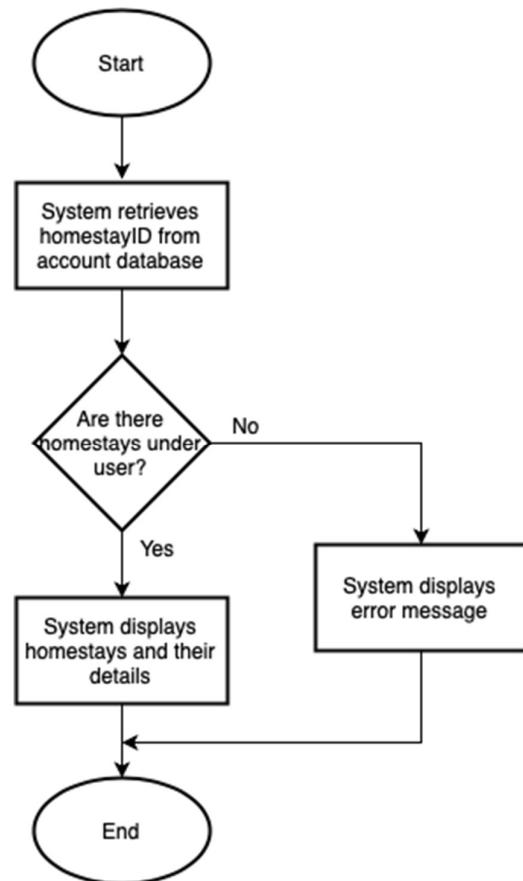
BM07 – Create Review



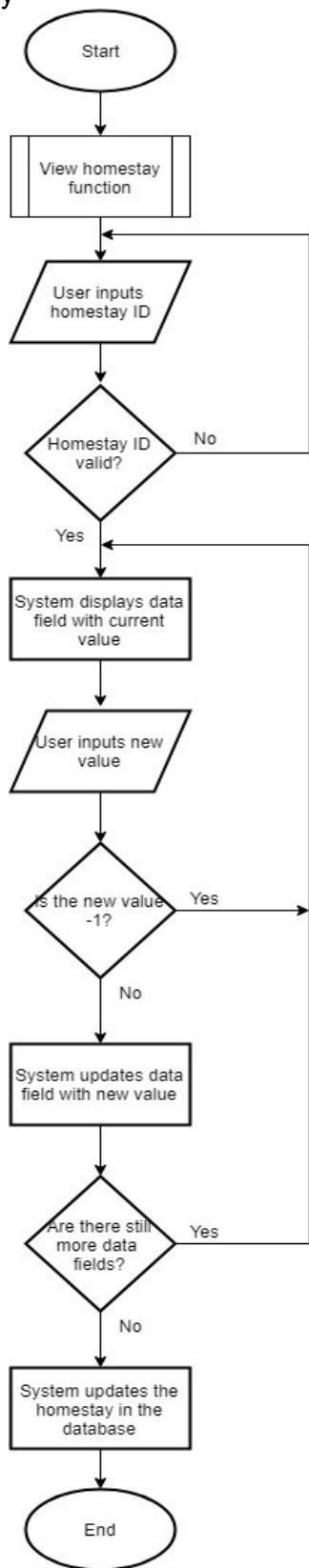
b. Homestay Management
HM01 – Add Homestay



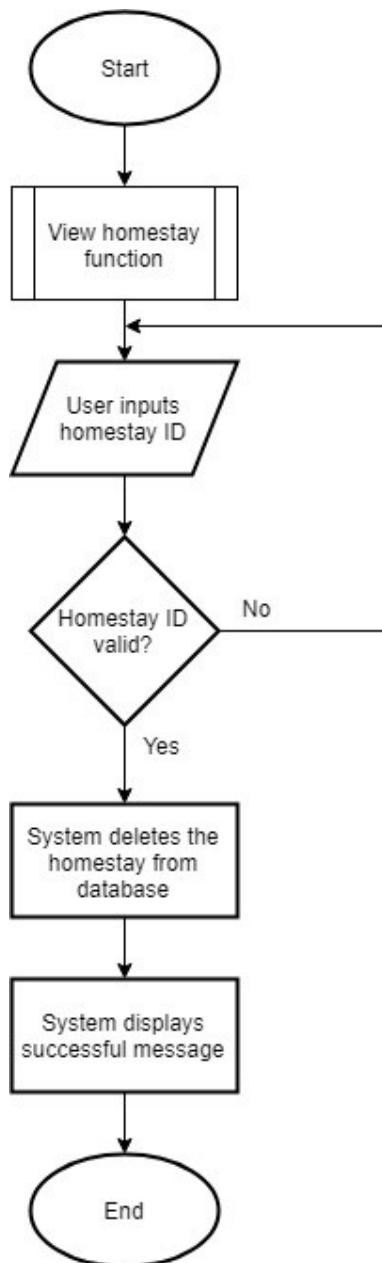
HM02 – View Homestay



HM03 – Update Homestay

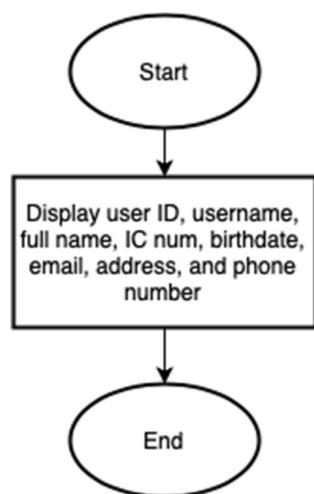


HM04 – Delete Homestay

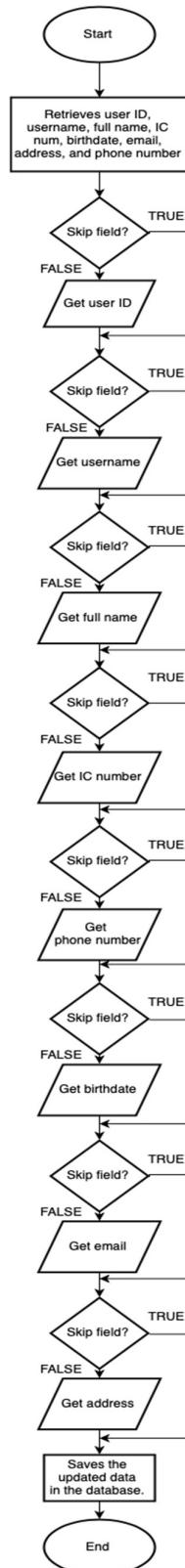


c. Member Profile Management

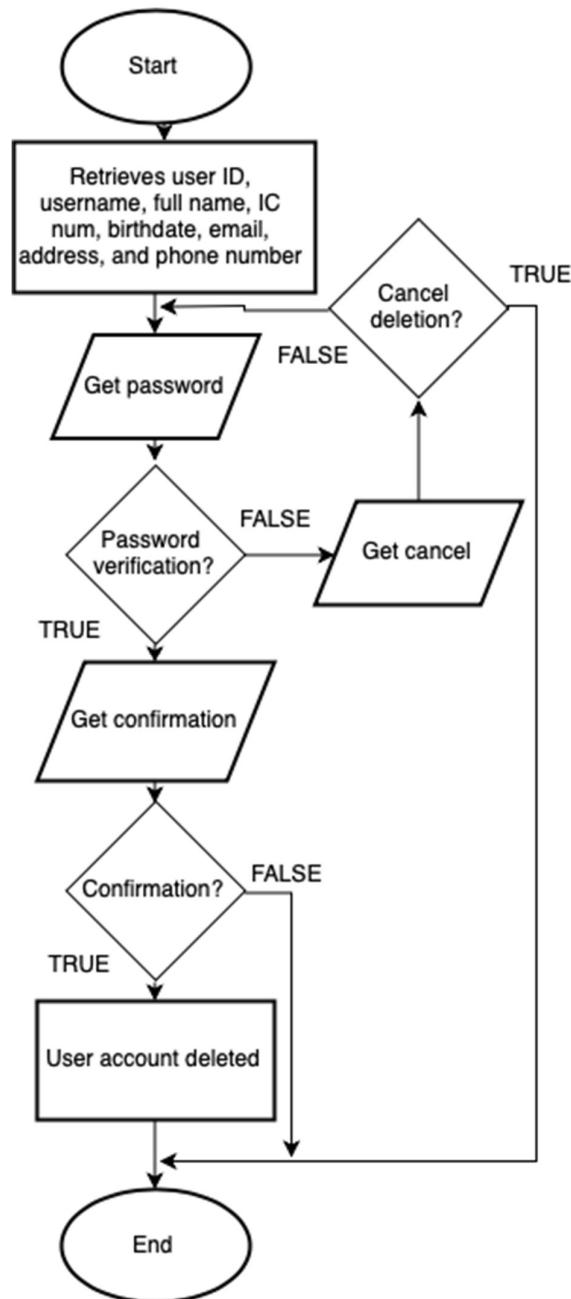
MPM01 – View Profile



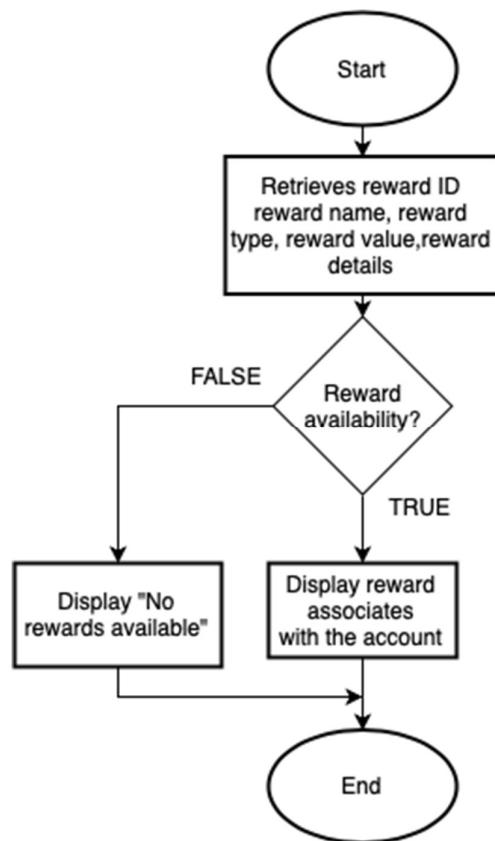
MPM02 – Update Profile



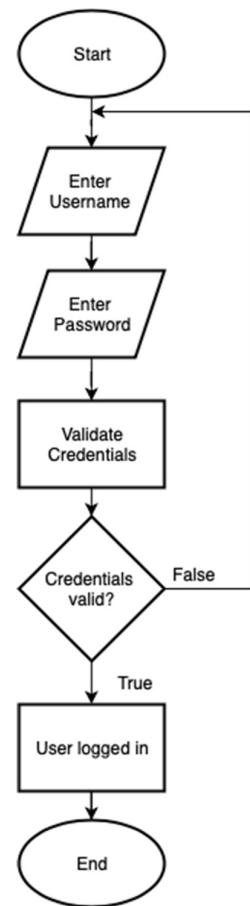
MPM03 – Delete Profile



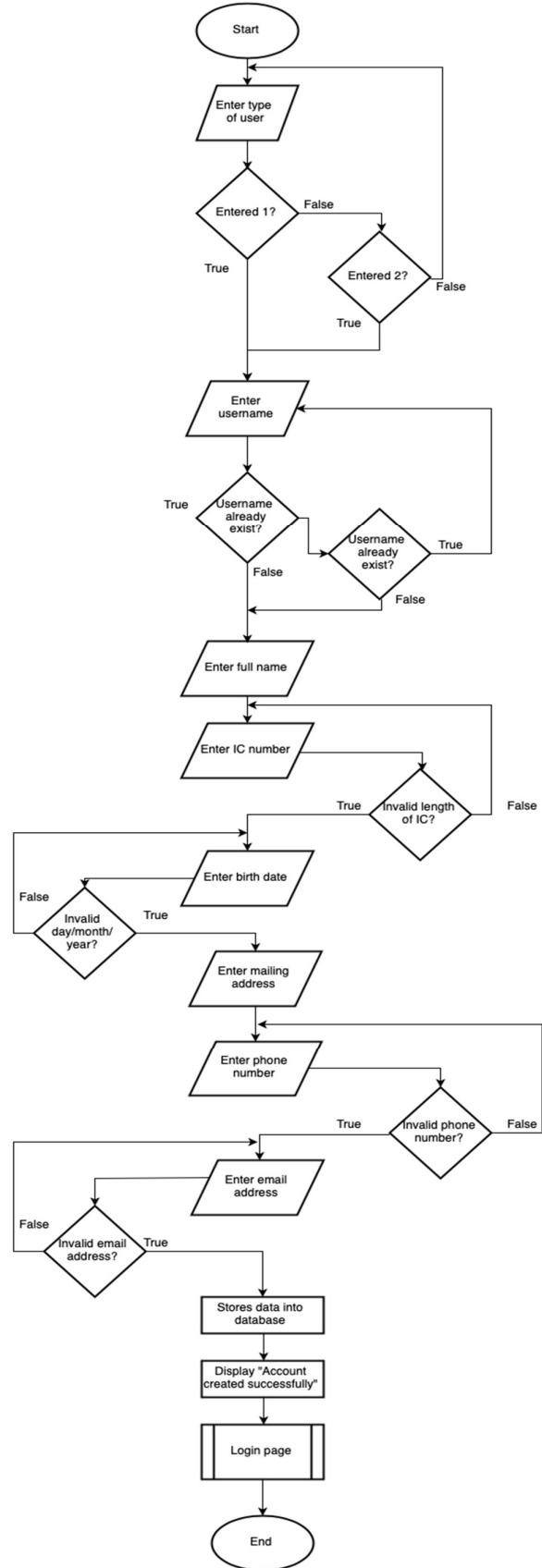
MPM04 – View Rewards



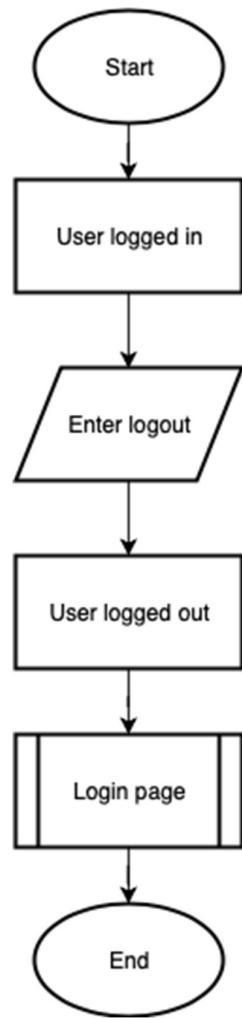
d. Login Management
LM01 – Login Account



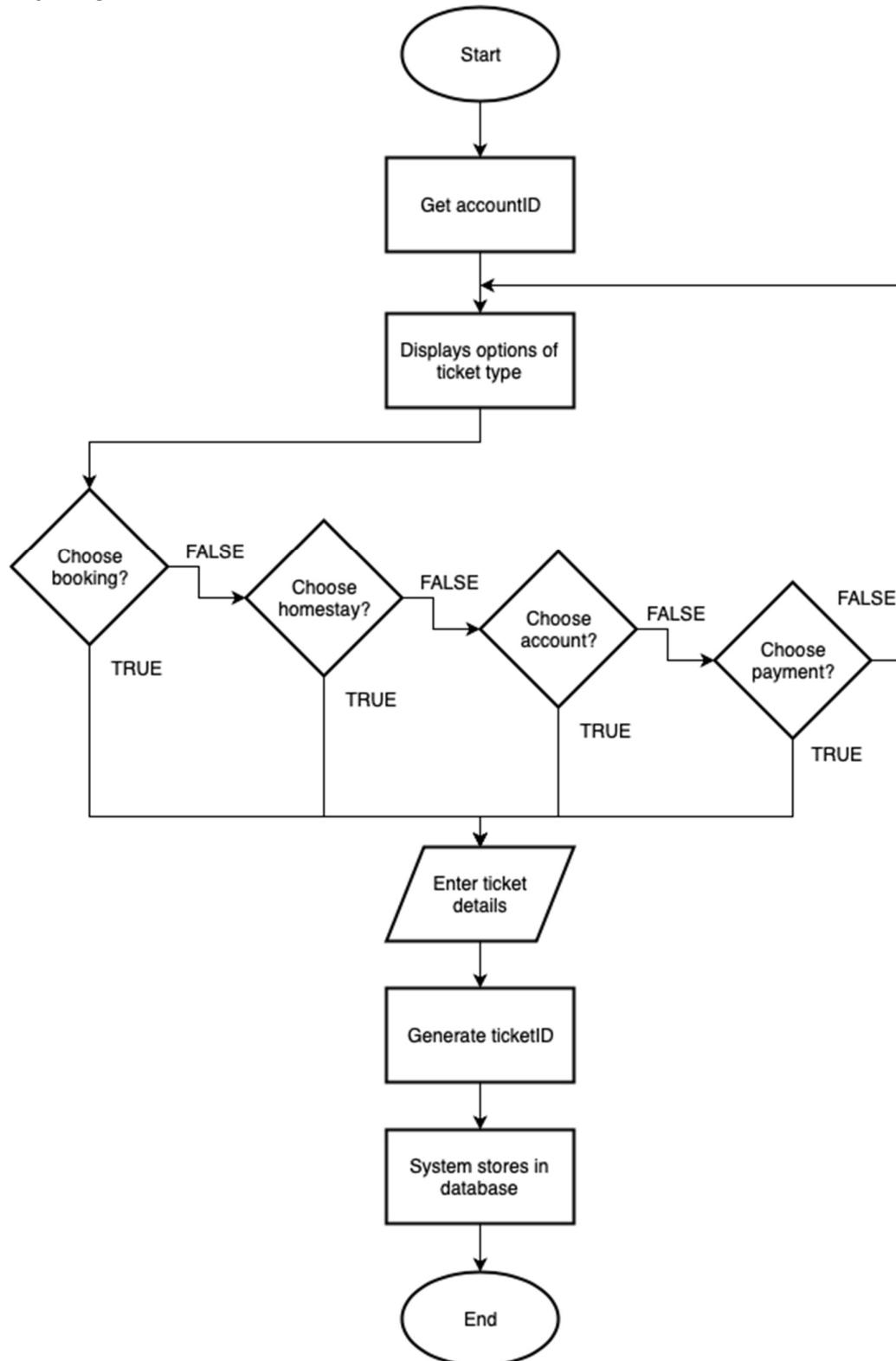
LM02 – Register Account



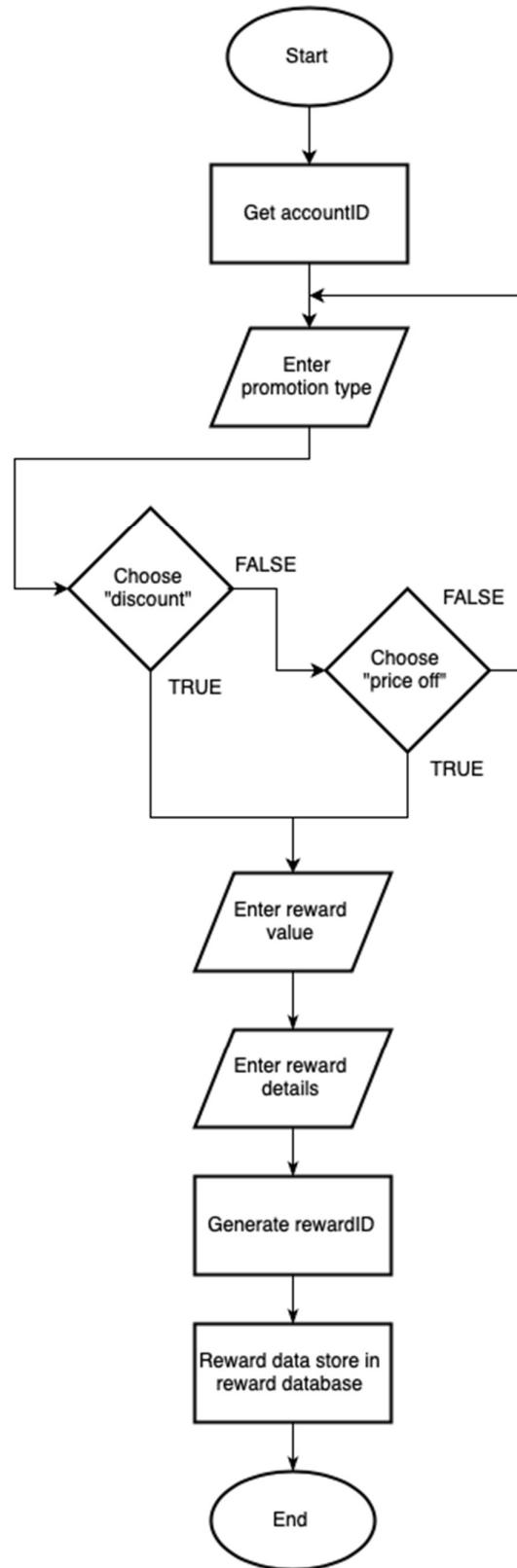
LM03 – Logout Account



e. Maintenance
M01 – Create Ticket



M02 – Create Reward



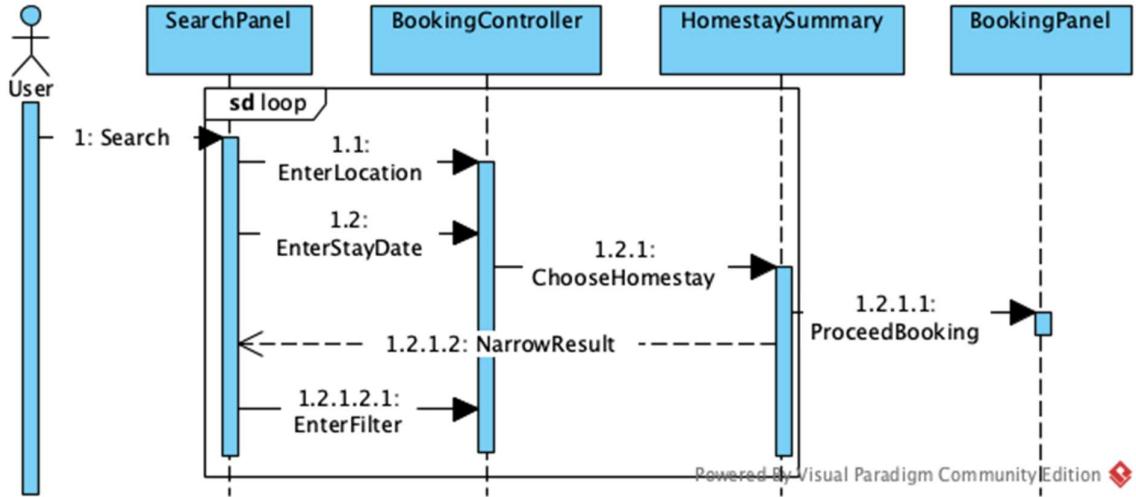
9. Business Rules

- a. Each customer may **create** booking(s).
- b. Each booking can be **created by** only one customer.
- c. Each host may **host** homestay(s).
- d. Each homestay can be **hosted by** only one host.
- e. Each booking can **book** only one homestay.
- f. Each homestay can be **booked by** only one booking.
- g. Each customer may **own** reward(s).
- h. Each reward may be **owned by** customer(s).
- i. Each customer may **make** payment(s).
- j. Each payment can be **made** by only one customer.
- k. Each booking can be **associated with** only one payment.
- l. Each payment can be **associated with** only one booking.
- m. Each payment can be **applied with** only one reward.
- n. Each reward can be **applied to** only one payment.
- o. Each customer may **make** review(s).
- p. Each review can be **made by** only one customer.
- q. Each homestay may **have** review(s).
- r. Each review can be **associated with** only one homestay.
- s. Each customer may **submit** ticket(s).
- t. Each ticket can be **submitted by** only one customer.
- u. Each host may **submit** ticket(s).
- v. Each ticket can be **submitted by** only one host.

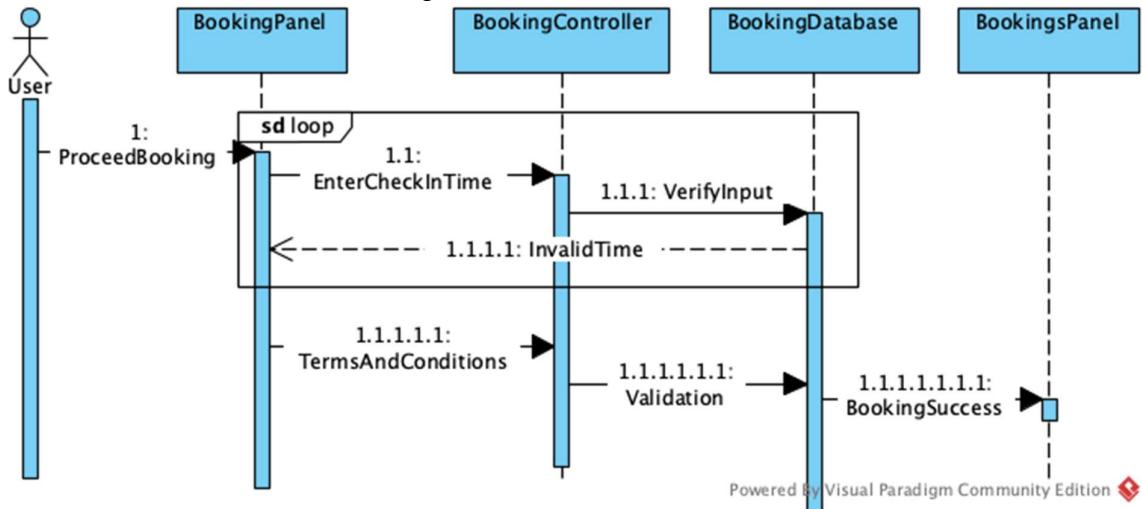
10. Sequence Diagram

a. Booking Management

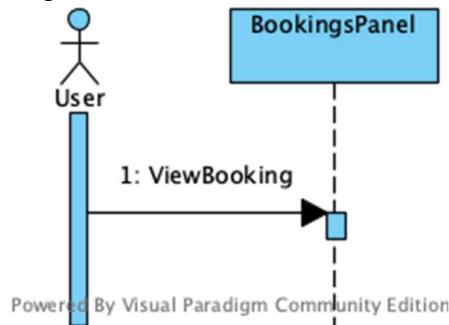
- BM01 - Search booking



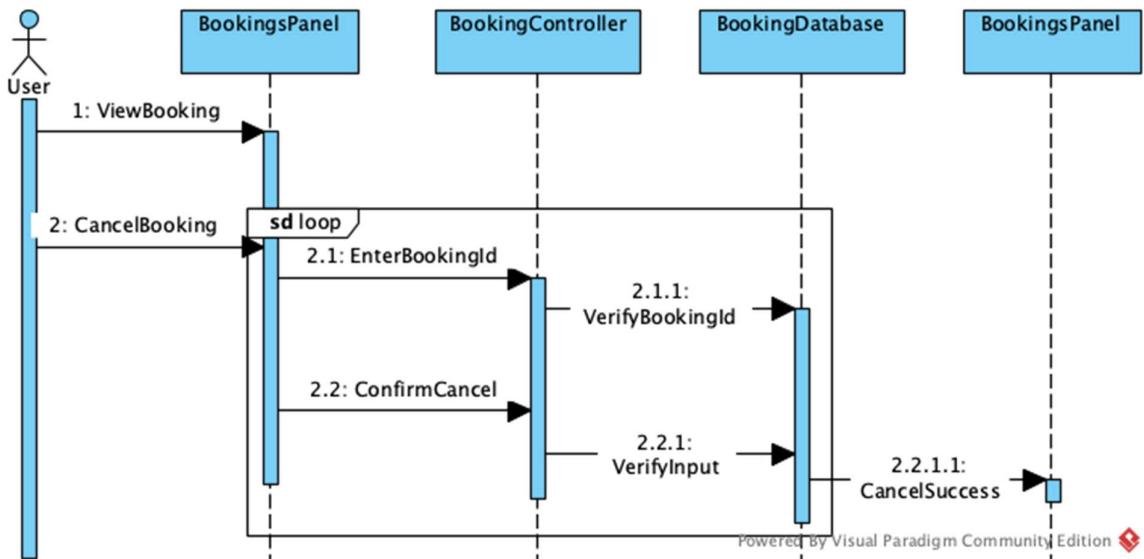
- BM02 - Create booking



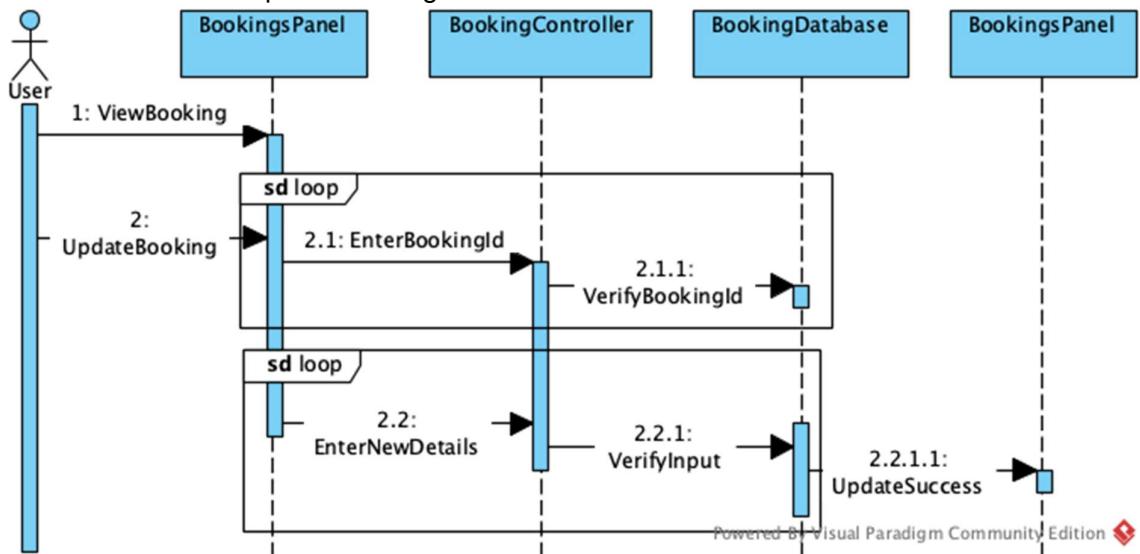
- BM03 - View booking



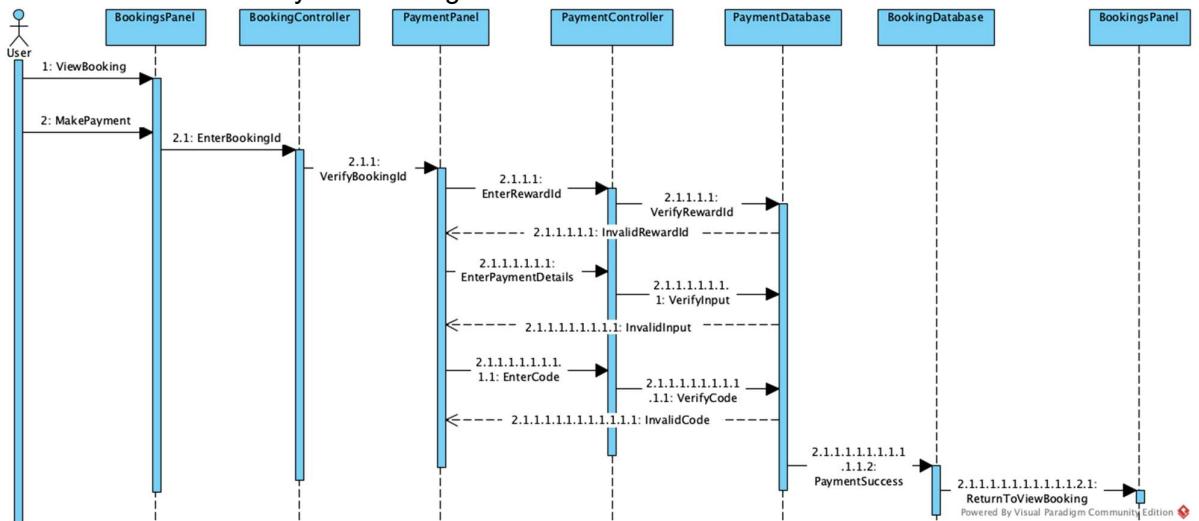
- BM04 - Cancel booking



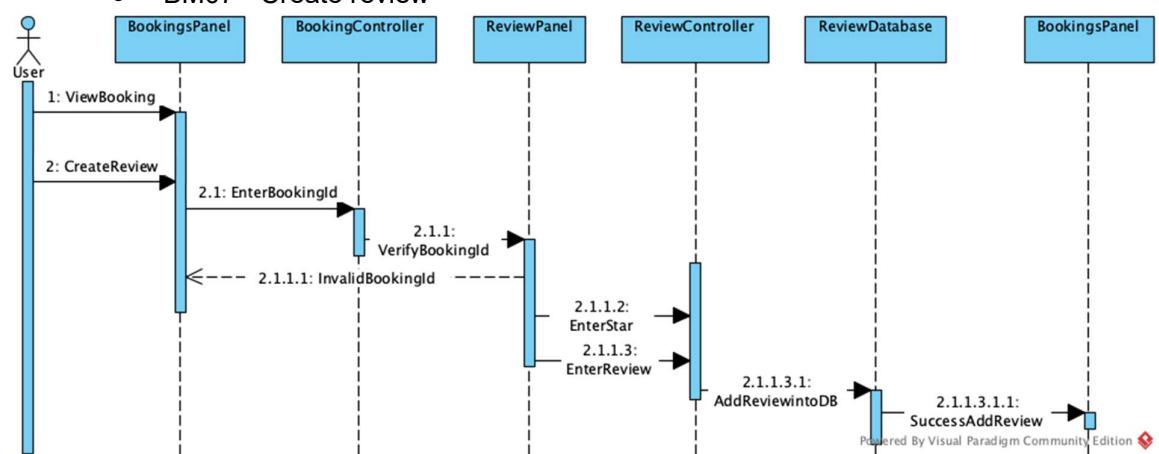
- BM05 - Update booking



- BM06 - Payment management

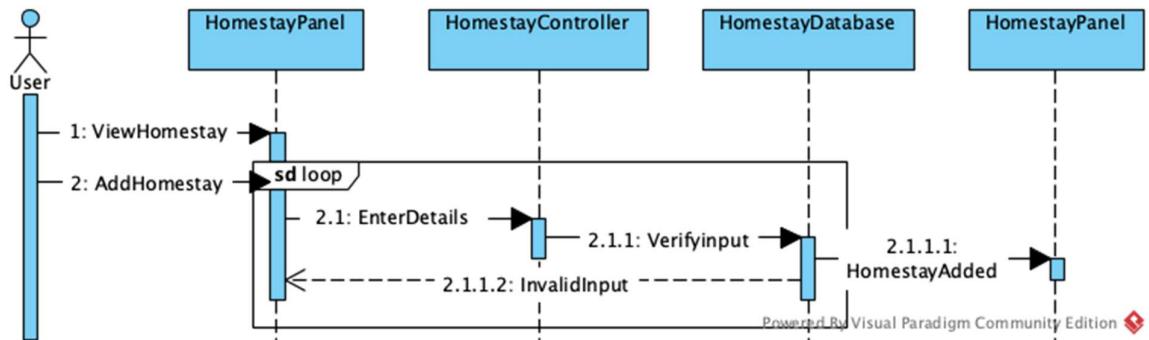


- BM07 - Create review

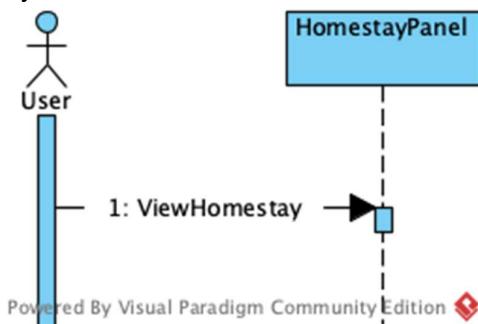


b. Homestay Management

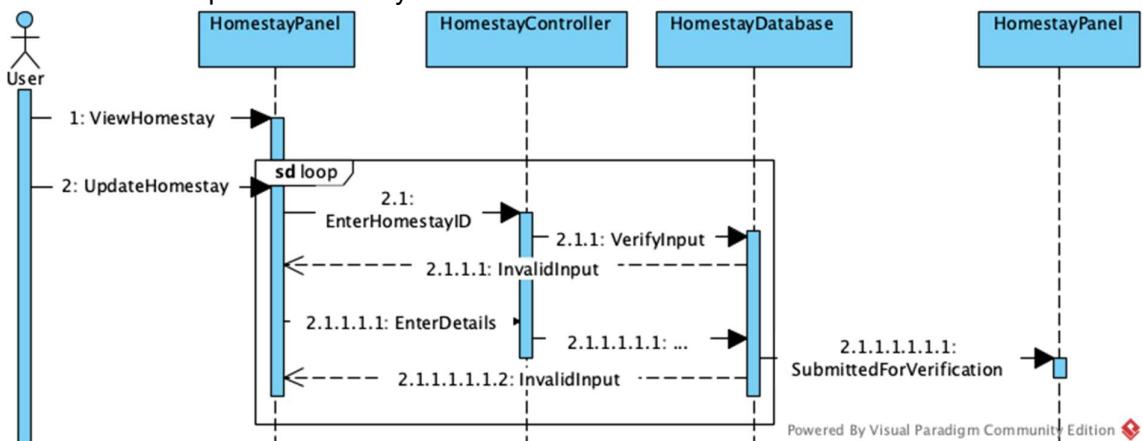
- HM01 - Add homestay



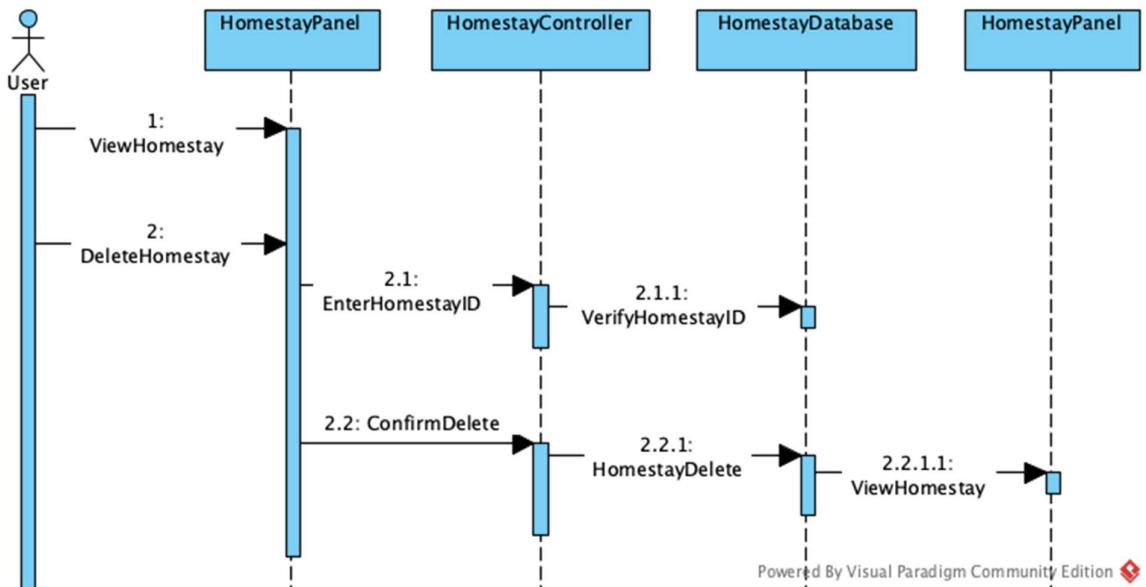
- HM02 - View homestay



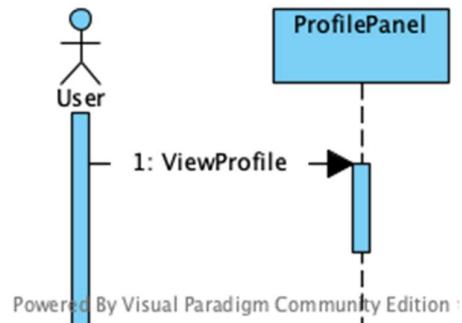
- HM03 - Update Homestay



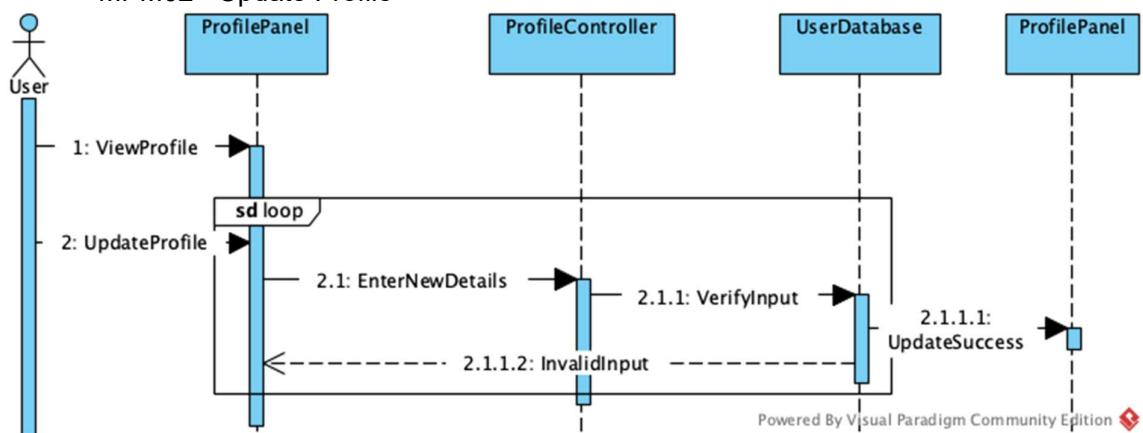
- HM04 - Delete Homestay



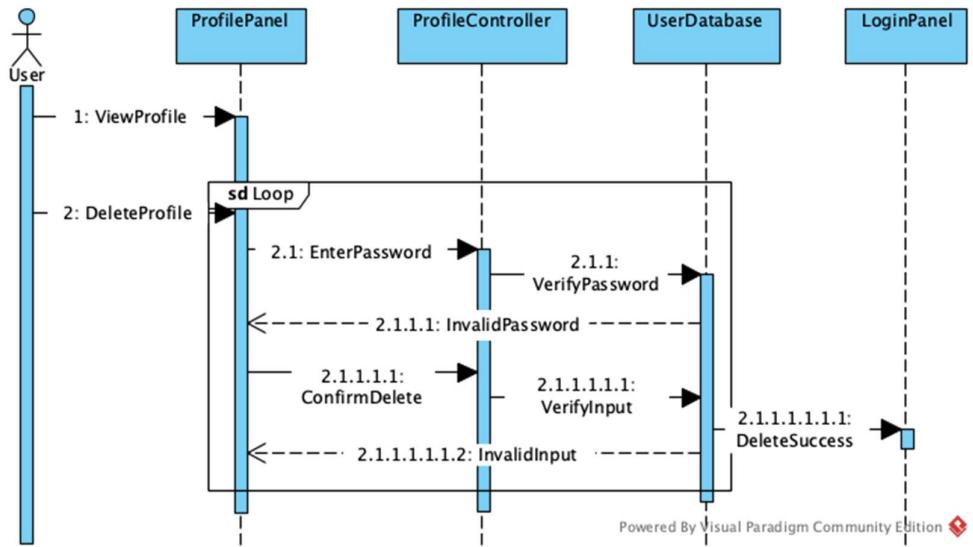
- c. Member Profile Management
 - MPM01 - View Profile



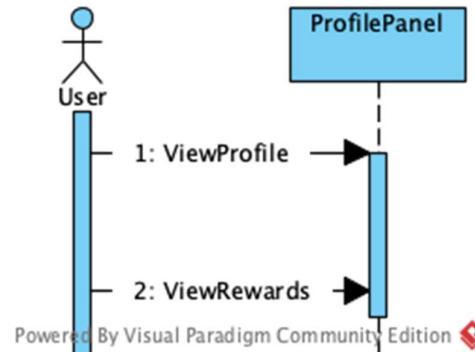
- MPM02 - Update Profile



- MPM03 - Delete profile

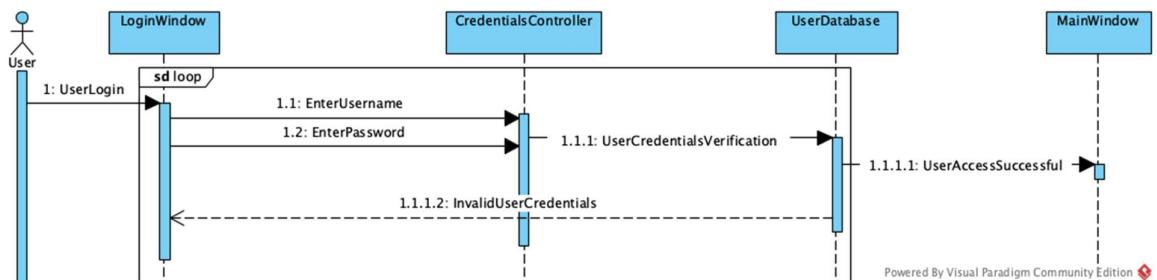


- MPM04 - View rewards

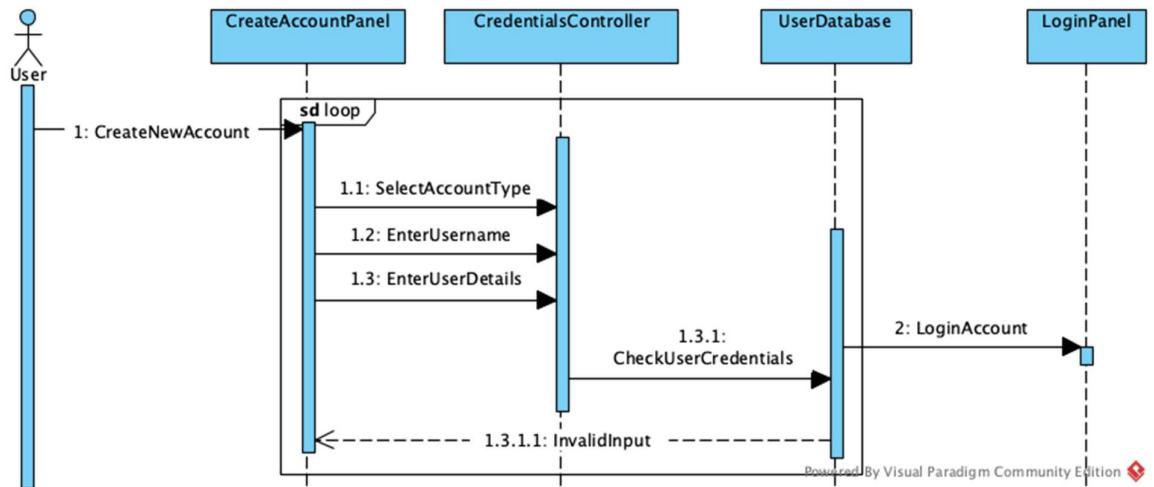


d. Login Management

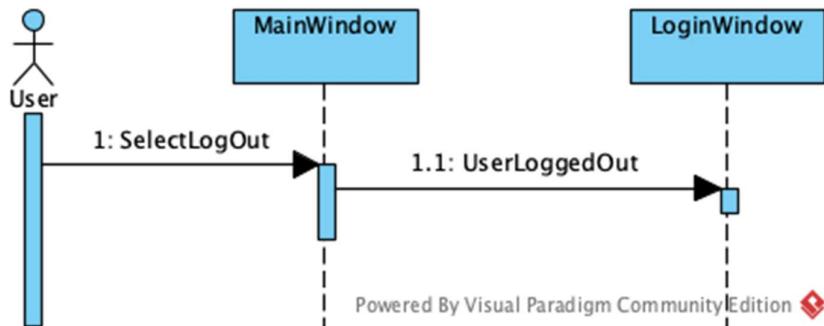
- #### • LM01 - Login Account



- LM02 - Register Account

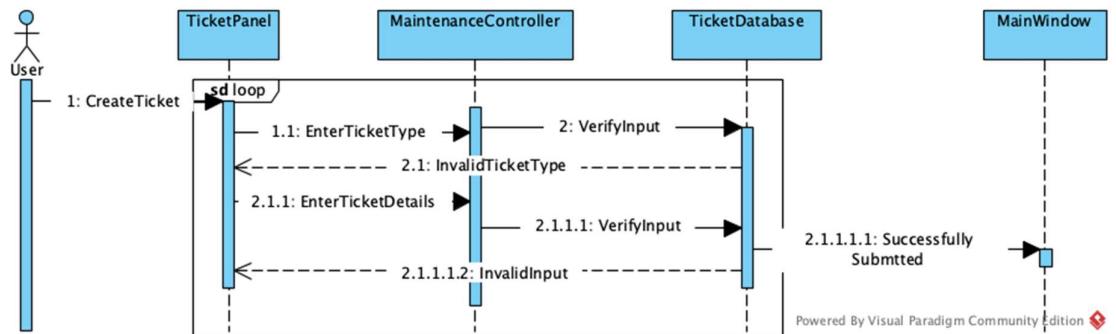


- LM03 - Logout Account

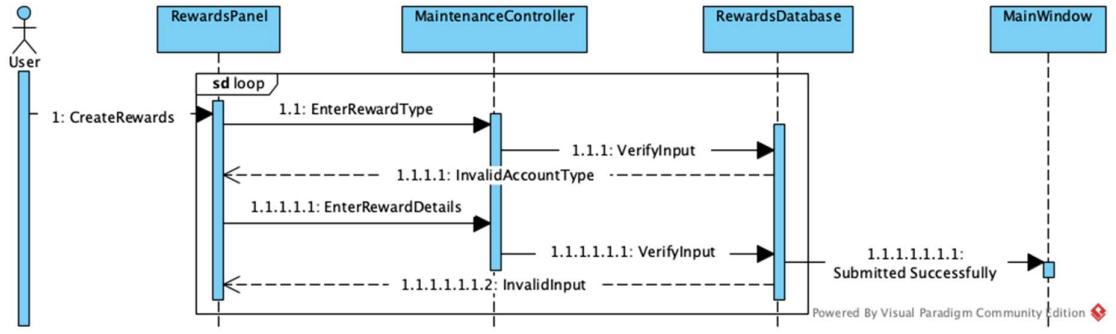


e. Maintenance

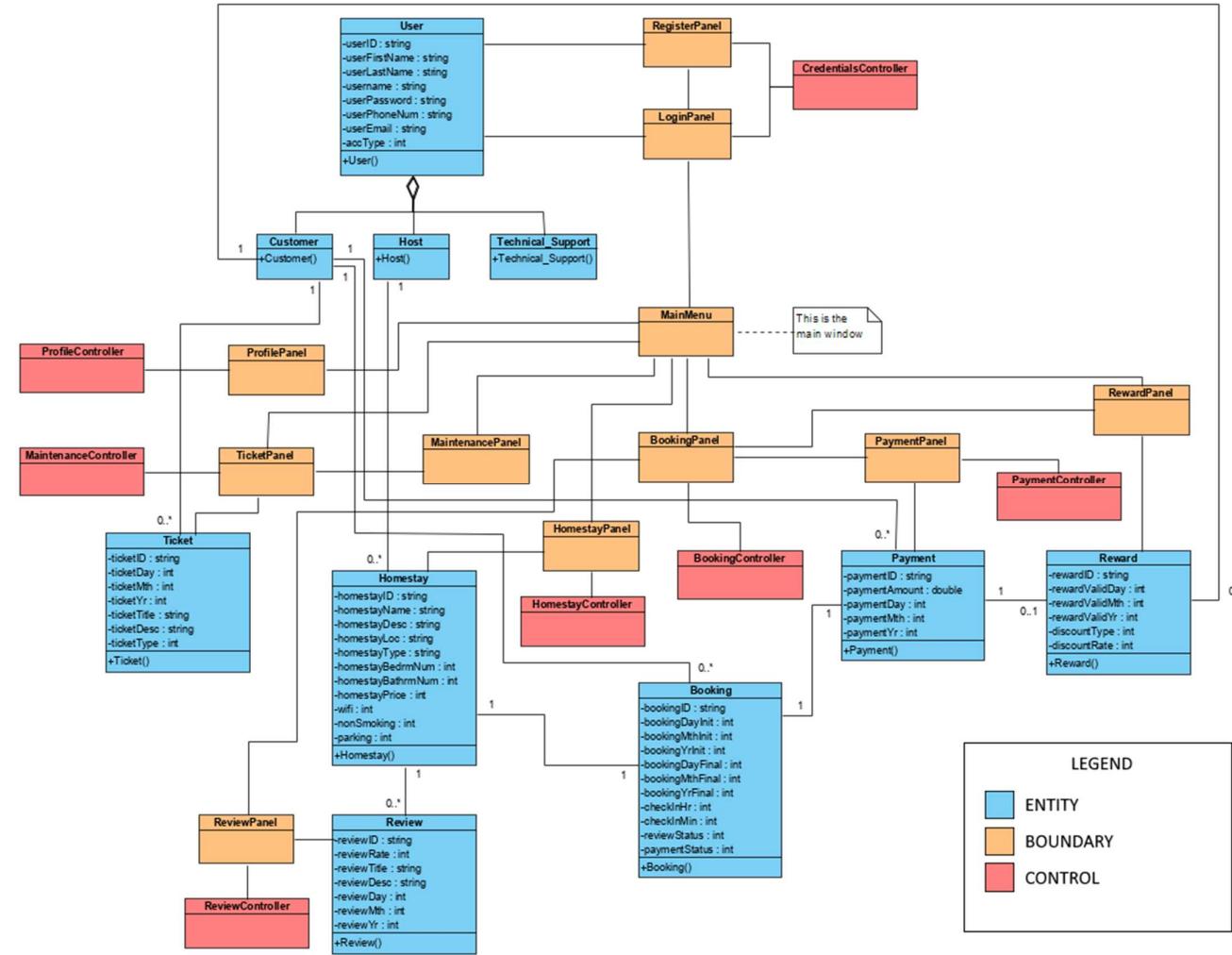
- M01 – Create Ticket



- M02 – Create Reward

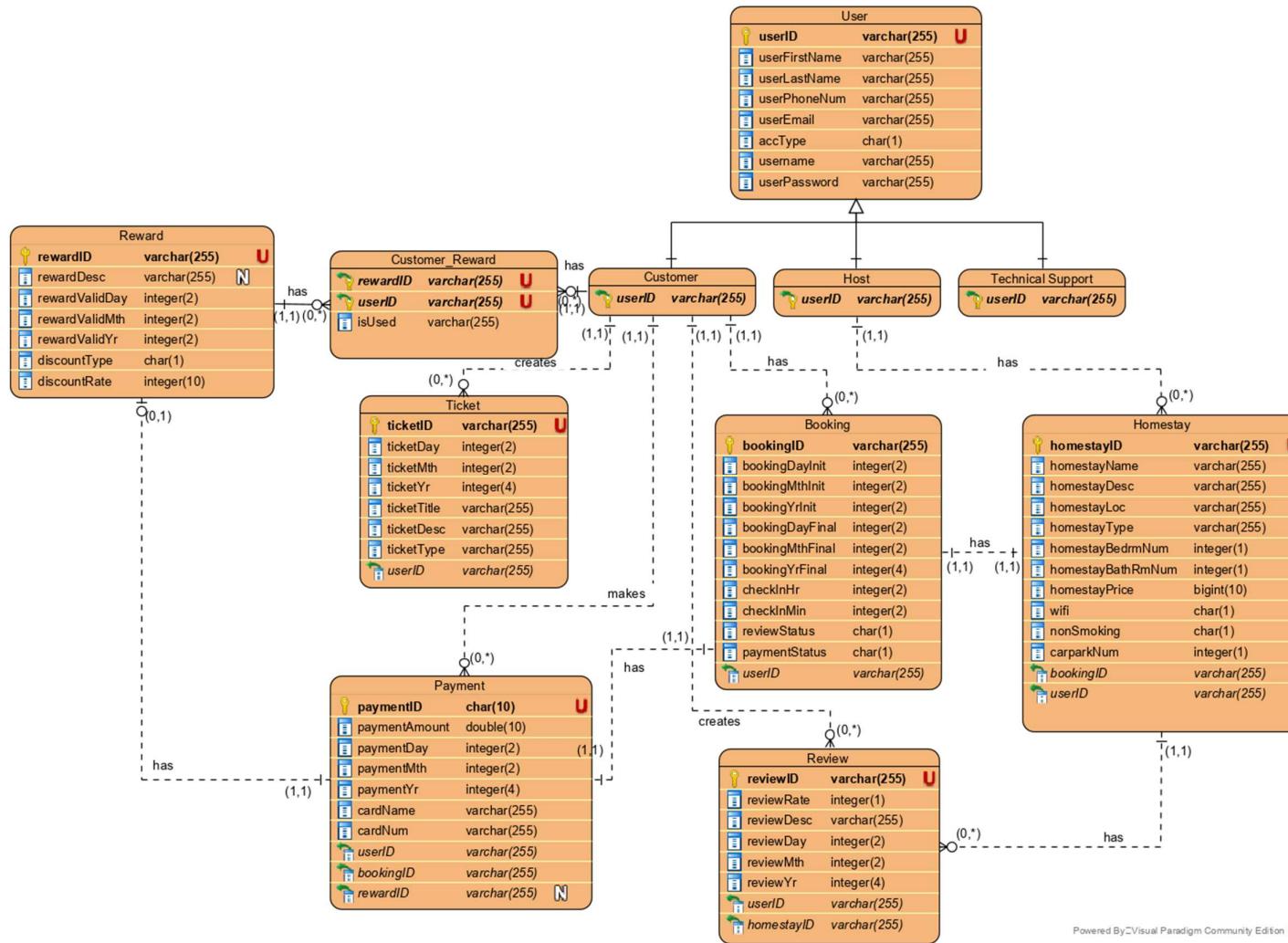


11. Class Diagram



The diagram above shows the relationship and connections between entity, boundary and control classes of this software. All boundary classes will associate with a control class for error-checking purposes.

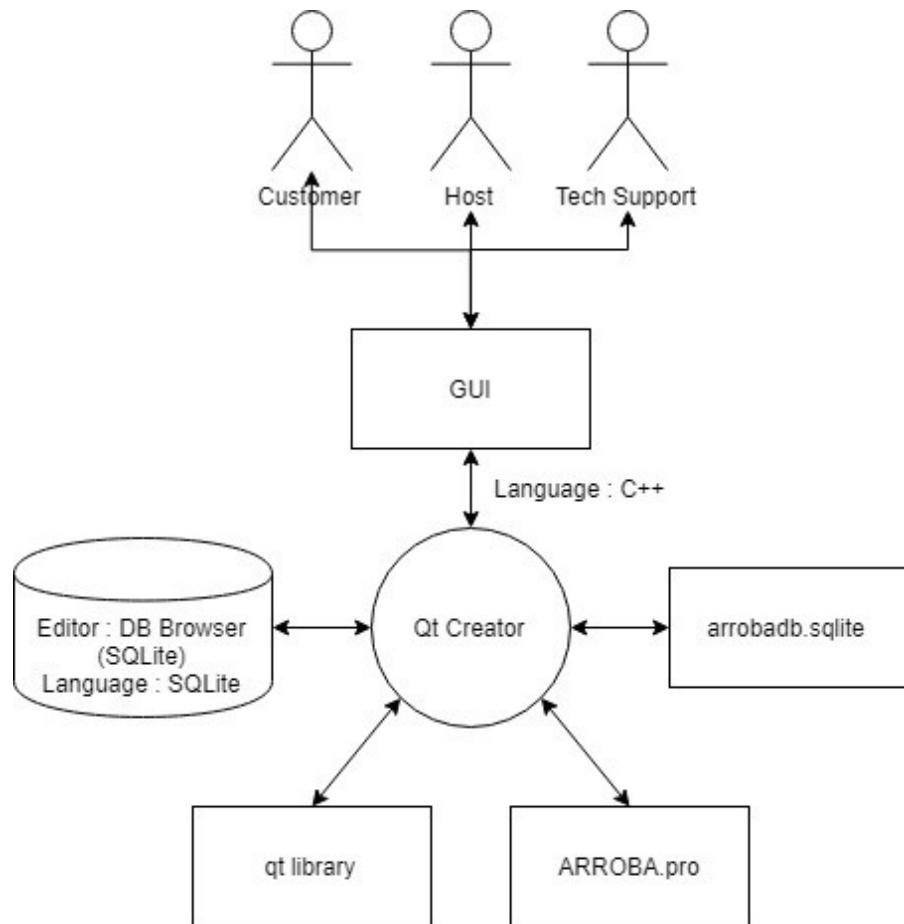
12. ER Diagram



Powered By Visual Paradigm Community Edition

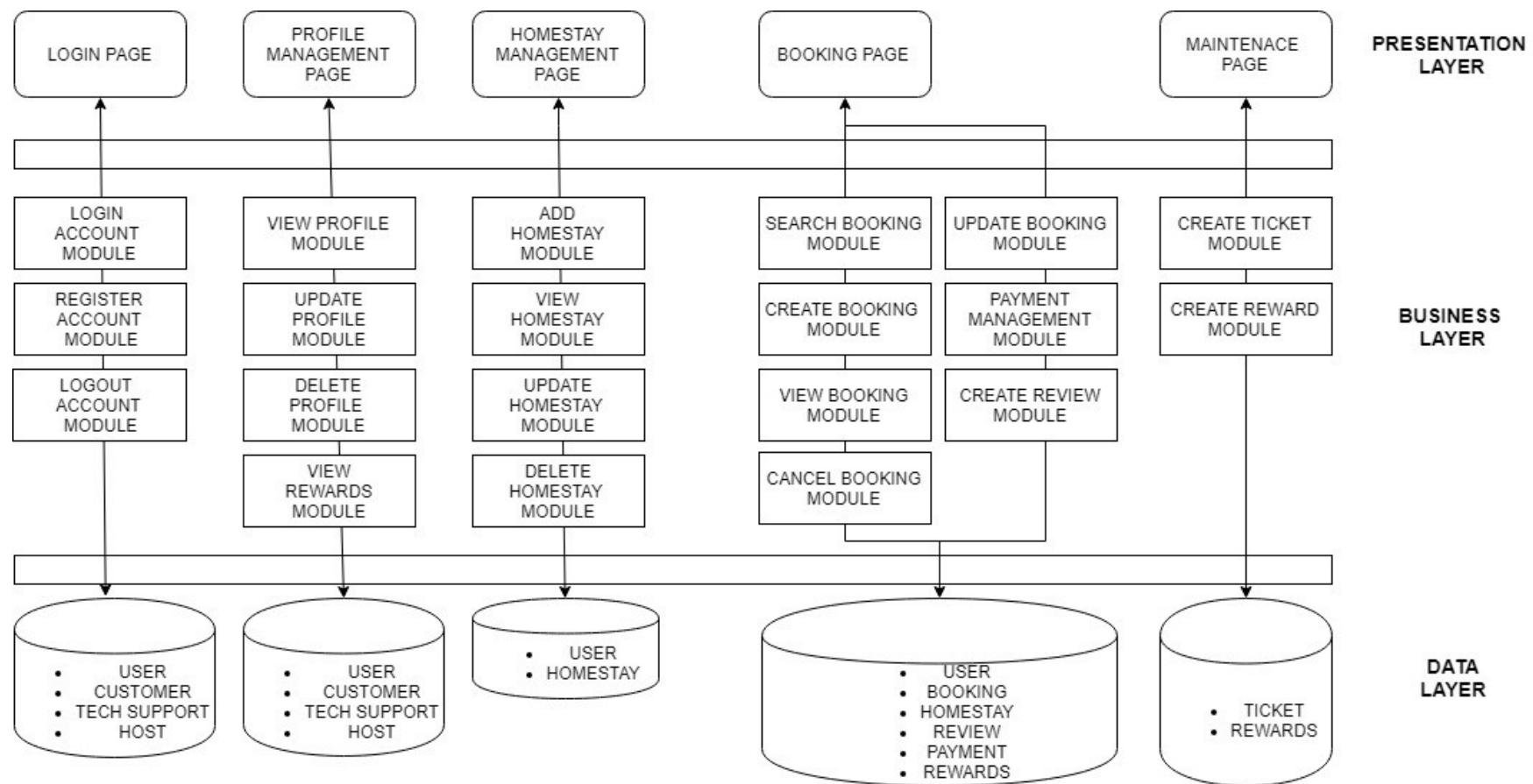
The diagram above shows the list of attributes that will be used to store all the data in the database. Note that we follow the naming convention that has been suggested.

13. Software Architectural Diagram



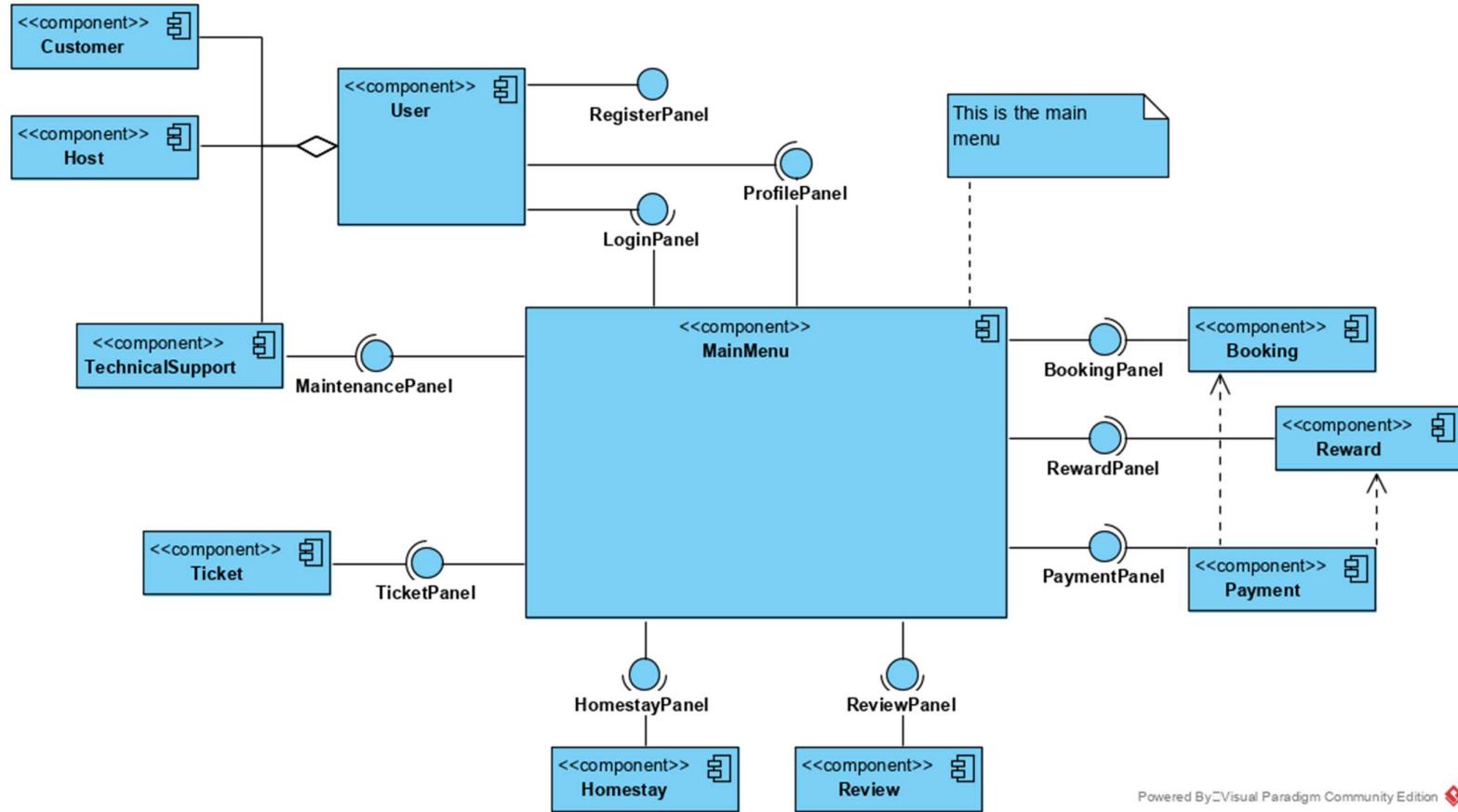
We will be using **C++** as our programming language for this software and uses **Qt** to create the UI. Our software will be connected to a database named **SQLite**.

14. Application Architectural Diagram



Our software consists of these sections according to the layers.

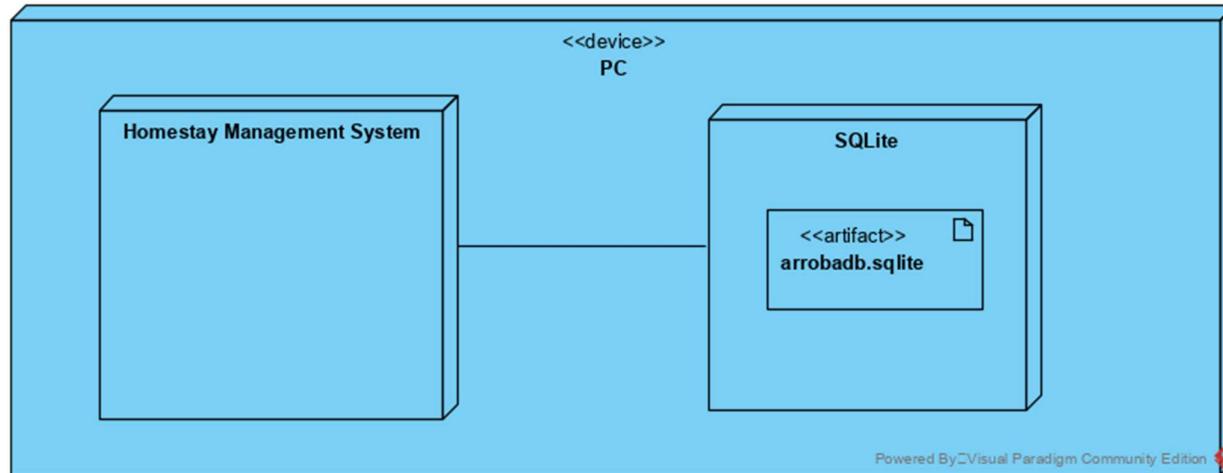
15. Component Diagram



All components will be linked to a main menu that will act as the intermediator between two components.

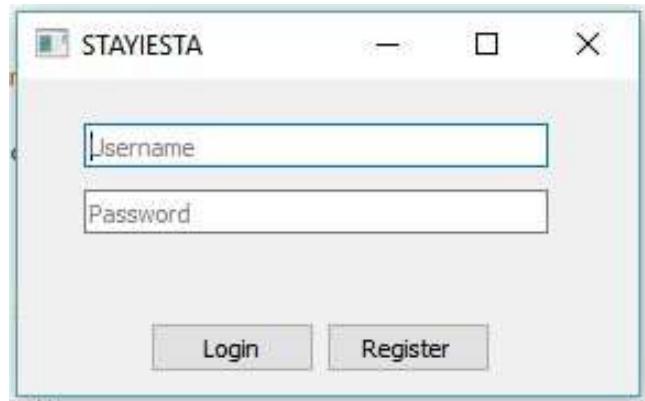
All actions will be carried out when there's a call from the **MainMenu**.

16. Deployment Diagram



*Our software is connected to a local host supplied by **SQLite**. All data can be retrieved although you are not connected to the internet.*

17. Interface Design



When user first launched the application, user will be greeted with this login panel. At this stage,

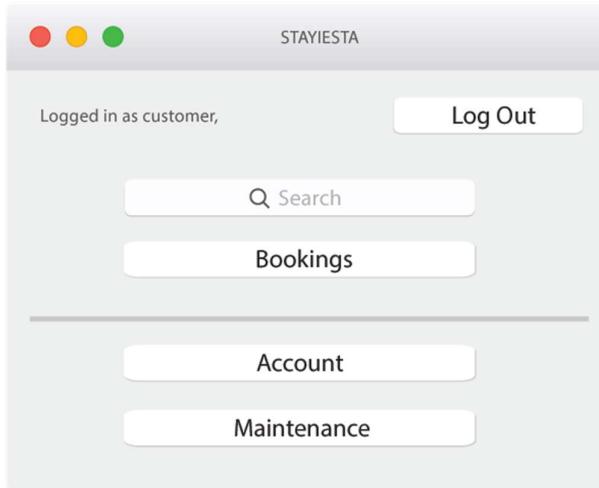
- i) User can enter their login credentials; or

Two side-by-side screenshots of a "Registration" window. The left screenshot shows the registration form with fields for First Name*, Last Name*, Username*, Password*, Phone Number*, and Email Address*. A note says "(*)" indicates required field. Below are checkboxes for Host (unchecked) and Customer (checked). Buttons at the bottom are "Register" and "Go Back". The right screenshot shows the same form filled with "test" in all fields, except for Password which is masked as "****". A note says "(*)" indicates required field. Below are checkboxes for Host (unchecked) and Customer (checked). A message "Registration successful" is displayed above the buttons "Register" and "Go Back".

- ii) User can click "Register" and register new account

When their login credentials are correct, they will enter the home page according to their account type:

i) Customer



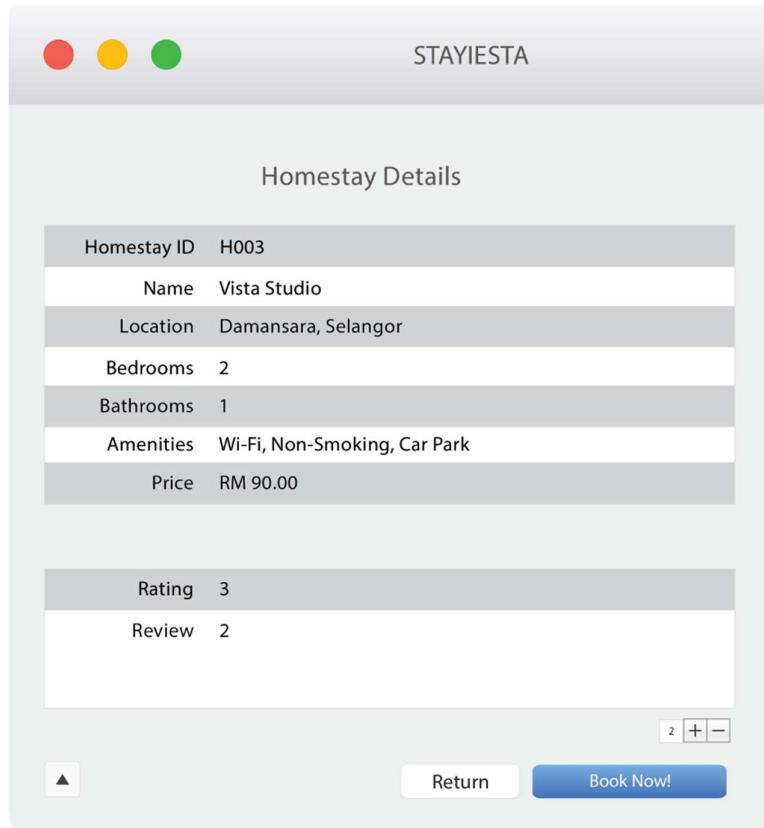
Customer Home Page

A screenshot of a Mac OS X-style window titled "STAYIESTA". The window contains the following elements:

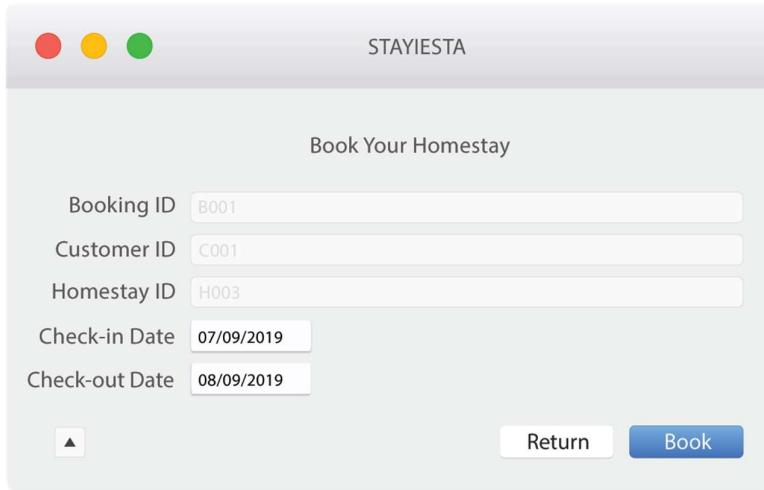
- Top right: Red, yellow, and green window control buttons.
- Top center: The word "STAYIESTA".
- Middle top: Text "Search Results".
- Middle left: A "Filters:" section with dropdown menus for "Bedrooms" and "Bathrooms", and checkboxes for "Carpark", "Wi-Fi", "Non-Smoking" (which is checked), and a "Filter" button.
- Middle center: A table showing search results:

ID	Name	Bedrooms	Bathrooms	Price	Carpark	Non-Smoking	Wi-Fi
015	Villa Anggerik	3	2	80.00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
003	Vista Studio	2	1	130.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
- Bottom left: A small upward arrow icon.
- Bottom right: "Details..." and "Book" buttons.

When user input their location at the search box, this will be the result page.



This will be the detailed page for the homestay that the user chose. Reviews made also include here.



Once user click "Book Now", user needs to enter the necessary details.

The screenshot shows a booking summary page titled "Booking Summary". At the top, there are three colored circles (red, yellow, green) and the word "STAYIESTA". Below the title, the booking details are listed in a table:

Booking ID	B02
Customer ID	C03
Name	Syakirah Mansor
E-mail	syakirah.m@mail.com
Homestay ID	H13
Name	The Rumah
Location	Hulu Langat, Selangor
Bedrooms	3
Bathrooms	2
Amenities	Wi-Fi, Non-Smoking, Carpark
Check-in Date	05/09/2019
Check-out Date	06/09/2019
Price	RM 180.00

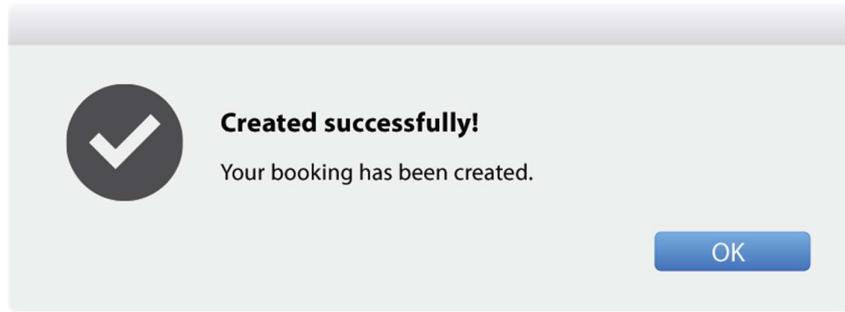
At the bottom of the page are navigation buttons: a small triangle icon, a "Return" button, and a "Next" button.

A create booking summary page will be shown.

The screenshot shows a terms and conditions page titled "Terms and Conditions". At the top, there are three colored circles (red, yellow, green) and the word "STAYIESTA". Below the title, a text box contains the following text: "This will be the terms and conditions for Stayiesta Homestay Management System developed by ArrobalInc." A vertical scroll bar is visible on the right side of the text box.

At the bottom of the page, there is a checkbox followed by the text: "I accepted to the terms and conditions stated above; and hereby giving consent to Stayiesta to manage my bookings." Below the checkbox are navigation buttons: a small triangle icon, a "Return" button, and an "Agree" button.

Terms and conditions page for user's reference



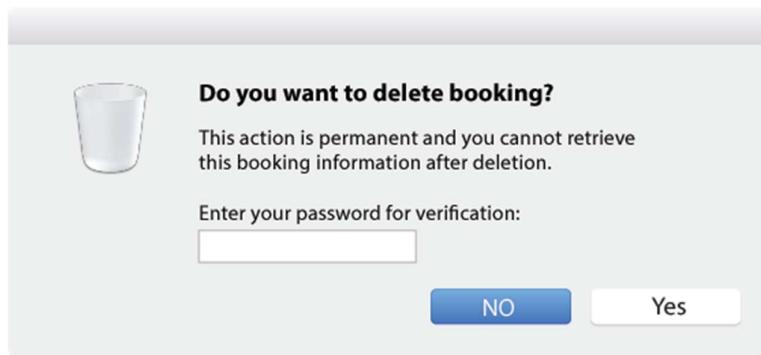
An action box will pop-out when user agrees to the terms and conditions that indicates the booking has been created successfully.

A screenshot of a web-based application titled "STAYIESTA". At the top, there are three colored window control buttons (red, yellow, green) on the left and the title "STAYIESTA" on the right. Below the title is a sub-header "Booking Management". A table lists bookings with the following data:

ID	HID	Homestay Name	Location	Check-in Date	Check-out Date	Price	Paid?	Reviewed?
001	015	Villa Anggerik	Petaling Jaya, Selangor	01/09/2019	02/09/2019	80.00	<input checked="" type="checkbox"/>	<input type="checkbox"/>
002	003	Vista Studio	Damansara, Selangor	15/09/2019	16/09/2019	130.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

At the bottom of the interface are four buttons: "Cancel", "Update", "Payment", and "Review".

When user clicked "Bookings" at the home page, this "View Bookings" page will appear.



User highlighted a booking and clicked "Cancel", this authorization page will appear.

STAYIESTA

Leave a Review

Customer ID: C001

Booking ID: B103

Homestay ID: H004

Name: Vista Suria

Rating: 4

Review: Homestay is good.

Booking is highlighted and user click “Review”, this create review page will appear.

Booking is highlighted and user click “Payment”,

STAYIESTA

Booking Summary

Booking ID	B02
Customer ID	C03
Name	Syakirah Mansor
E-mail	syakirah.m@mail.com
Homestay ID	H13
Name	The Rumah
Location	Hulu Langat, Selangor
Bedrooms	3
Bathrooms	2
Amenities	Wi-Fi, Non-Smoking, Carpark
Check-in Date	05/09/2019
Check-out Date	06/09/2019
Price	RM 180.00

Reward Code: R02

Payment summary page will appear with input for reward code.

STAYESTA

Payment Gateway

Customer ID: C001

Booking ID: B103

Homestay ID: H004

Payment Type: Vista Suria

Cardholder Name: Abu Bakar Siddiq

Card Number: 4833-7322-1234-2122

Expiry Date: 10/22

CVV: 232

When user proceed to payment, a payment gateway page will appear.

Authorizing Payment

Please enter the digit below for authorization:
B34WYQ

Payment authorization action box

STAYESTA

Create Ticket

Name: Text Field

ID: Text Field

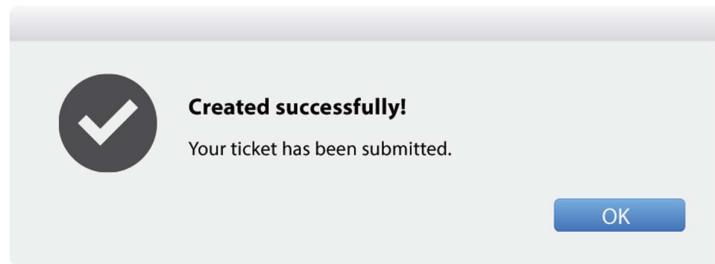
Type: Text Field

E-mail: Text Field

Ticket Type: General Enquiry

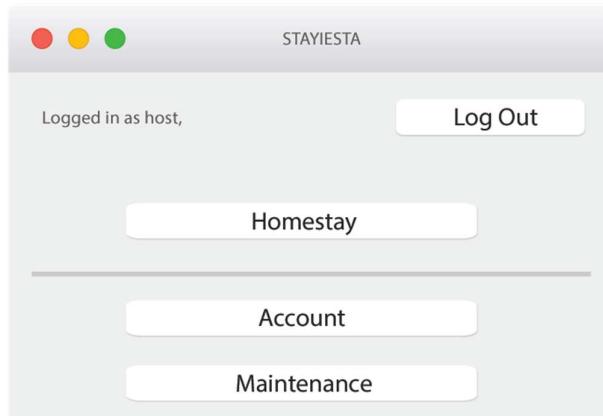
Description:

When user click "Maintenance" at home page, a create ticket page will pop-out.



An action box will pop-out when the ticket submitted successfully.

ii) Host



Home Page

The screenshot shows the 'STAYIESTA' application window with the title 'Homestay Management'. It displays a table of homestays:

ID	Name	Bedrooms	Bathrooms	Price	Carpark	Non-Smoking	Wi-Fi
015	Villa Anggerik	3	2	80.00	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
003	Vista Studio	2	1	130.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

At the bottom of the table are buttons for 'Delete', 'Update', and 'Add'.

When user clicked "Homestay", a list of homestays will appear.

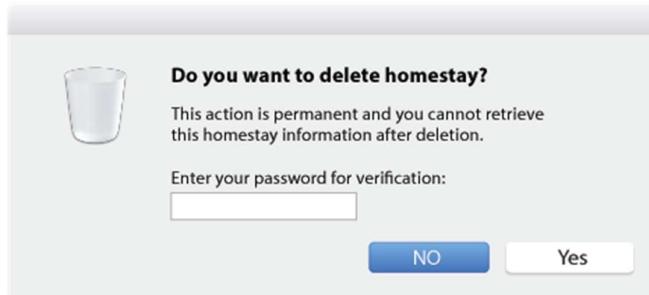
When user highlighted a homestay:

The screenshot shows the 'STAYIESTA' application window with the title 'Homestay Management'. The 'Vista Studio' row from the previous table is selected. The form fields are as follows:

- Name: Vista Studio
- ID: 003
- Location: Damansara, Selangor
- Bedrooms: 3
- Bathrooms: 1
- Price: 150.00
- AMENITIES:
 - Carpark:
 - Wi-Fi:
 - Non-Smoking:

At the bottom are buttons for 'Reset' and 'Save'.

a) And click “Update”, user will be prompt to update necessary information



b) And click “Delete”, user will be ask to enter password for authorization.

Homestay Management

Name: Condominium Melati

ID: 003

Location: Petaling Jaya, Selangor

Bedrooms: 3

Bathrooms: 2

Price: 130.00

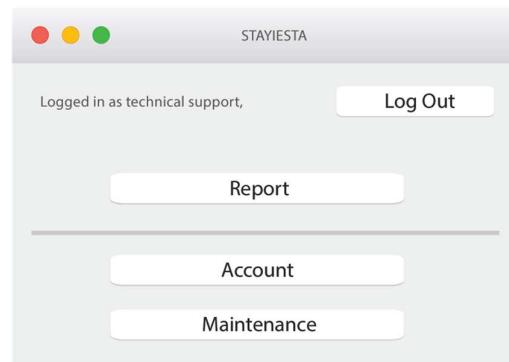
AMENITIES

Carpark Wi-Fi Non-Smoking

▲ Reset Save

When user click “Add”, user will be ask to enter necessary information.

iii) Technical Support

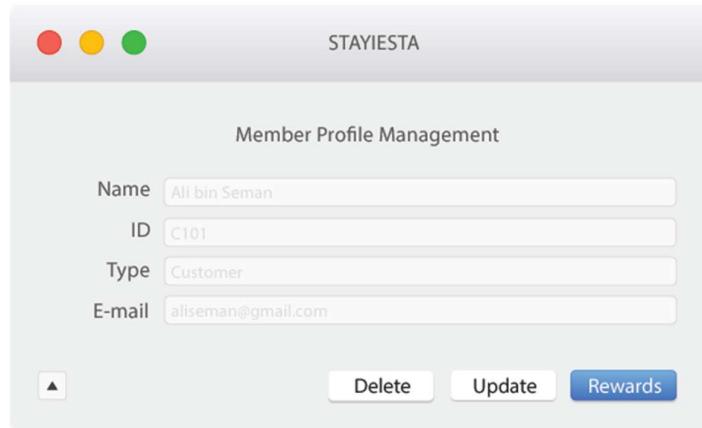


Home Page

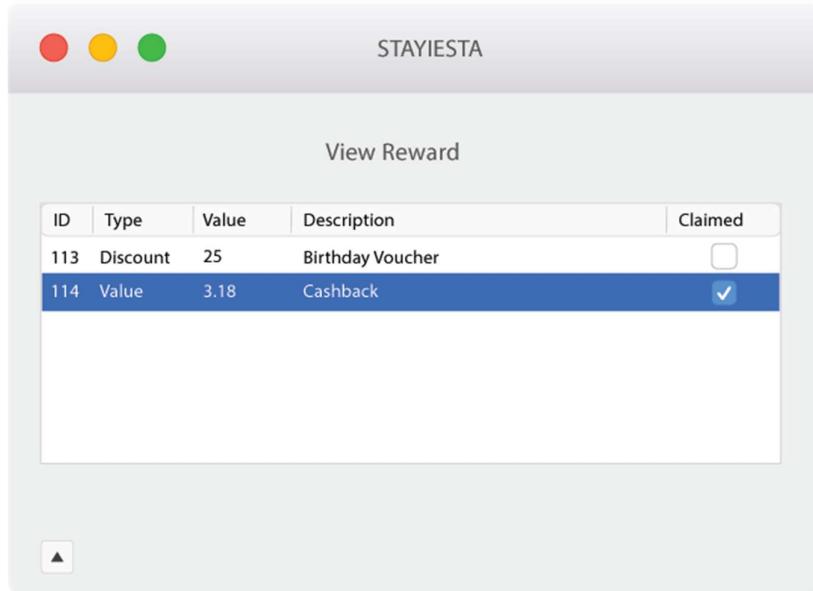
The screenshot shows a window titled "STAYIESTA" with the title "Create Reward". It contains several input fields: "Name" (Text Field), "ID" (Text Field), "Type" (Text Field), "E-mail" (Text Field), "Reward Type" (dropdown menu set to "Discount"), "Value" (Text Field), and "Description" (Text Area). At the bottom are "Reset" and "Submit" buttons.

When user click "Maintenance", user can input the details to add reward.

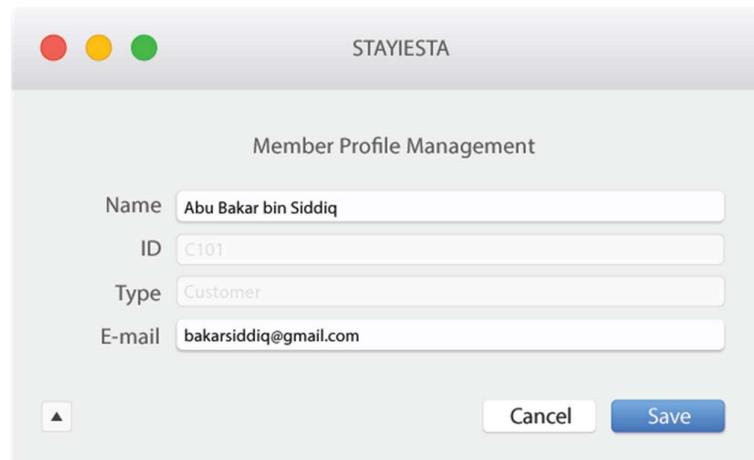
For all users,



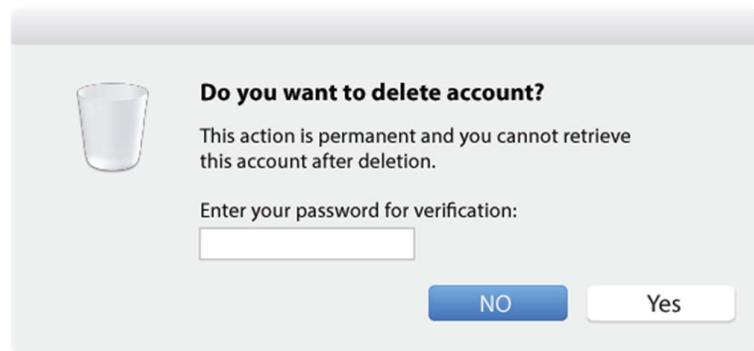
An account management will pop-out when they click “Account”



When user click “Rewards”, they can view list of rewards they owned with the status.



When user click “Update”, they can update necessary information.



When user click “Delete”, they can delete their profile and input the password for authorization.

18. Quality Attributes

Software Quality: Ease of Navigations

- In our software, there will be a home button in **every window except login panel, register panel and dialogs; and popups**. This is to increase functionality and user-friendliness.

Presentation Quality: Simplicity in Presenting the software

- We will use **simple words but in a proper manner** to demonstrate our software. This will increase understanding and decrease time usage in presentation.

19. Data Dictionary

Class/Function Name	Variable Name	Variable Type
<i>Login Management Modules</i>		
LoginPanel	lineEditPath lineEditUser logStatus picLbl buttonLogin buttonReg	QLineEdit QLineEdit QLabel QPixmap QPushButton QPushButton
RegisterPanel	checkBoxCust checkBoxHost lineEditAddr lineEditFName lineEditLName lineEditPass lineEditPhone lineEditUser regStatus buttonback buttonRegister	QCheckBox QCheckBox QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit QCheckBox QPushButton QPushButton
PopupRegisterPanel	picLbl buttonOK aLabel	QPixmap QPushButton QLabel
MainMenuCust	buttonLogout loginAsLabel picLbl buttonAccount buttonBooking buttonTicket lineEditSearch	QPushButton QLabel QPixmap QPushButton QPushButton QPushButton QLineEdit
MainMenuHost	buttonLogout loginAsLabel picLbl buttonHomestay buttonAccount buttonMaint	QPushButton QLabel QPixmap QPushButton QPushButton QPushButton
MainMenuTech	buttonLogout loginAsLabel picLbl buttonAccount buttonReward buttonTicket	QPushButton QLabel QPixmap QPushButton QPushButton QPushButton

Booking Modules		
SearchResult	listEditName buttonFilter buttonReset comboBoxWifi comboBoxBobMore comboBoxCar comboBoxSmoke comboBoxWifi buttonBook buttonDetail buttonHome	QString QPushButton QPushButton QComboBox QComboBox QComboBox QComboBox QComboBox QPushButton QPushButton QPushButton
HomestayDetail	tableView buttonNextRev buttonPrevRev lineEditRate textEditReview buttonBook buttonHome buttonReturn	QTableView QPushButton QPushButton QLineEdit QTextEdit QPushButton QPushButton QPushButton
BookingCreate	dateEditIn dateEditOut lineEditBookID lineEditHomeID lineEditUserID buttonBook buttonHome buttonReturn	QDateEdit QDateEdit QLineEdit QLineEdit QLineEdit QPushButton QPushButton QPushButton
BookingSummary	tableView buttonBook buttonHome buttonReturn	QtableView QPushButton QPushButton QPushButton
PopupBookingCreate	picLbl buttonOK	QPixmap QPushButton
BookingView	tableView buttonCancel buttonHome buttonPay buttonReview buttonUpdate	QTableView QPushButton QPushButton QPushButton QPushButton QPushButton
BookingDelete	lineEditPass	QLineEdit

	buttonCancel buttonConfirm aLabel picLbl	QPushButton QPushButton QLabel QPixmap
<i>Payment Module</i>		
PaymentSummary	lineEditReward buttonApply buttonCancel buttonCheckout buttonHome tableView	QLineEdit QPushButton QPushButton QPushButton QPushButton QTableView
PaymentGateway	lineEditBookID lineEditHomeID lineEditUserID lineEditPrize comboBoxType lineEditCvv lineEditCard lineEditDate lineEditName buttonAutho buttonCancel buttonHome	QLineEdit QLineEdit QLineEdit QLineEdit QComboBox QLineEdit QLineEdit QLineEdit QLineEdit QPushButton QPushButton QPushButton
PaymentAuthorization	buttonAutho labelCaptcha lineEditAutho buttonCancel picLbl	QPushButton QLabel QLineEdit QPushButton QPixmap
PopupPaymentSuccess	picLbl buttonOK	QPixmap QPushButton
PopupRewardApplied	picLbl buttonOK	QPixmap QPushButton
PopupRewardApplyFailed	picLbl buttonOK	QPixmap QPushButton
<i>Homestay Modules</i>		
HomestayView	buttonAdd buttonUpdate buttonDelete buttonHome tableView	QPushButton QPushButton QPushButton QPushButton QTableView
HomestayAdd	lineEditName	QLineEdit

	lineEditLoc lineEditBed lineEditBath comboBoxType lineEditPrize lineEditCar aLabel checkBoxWifi checkBoxSmoke buttonAdd buttonReset buttonHome	QLineEdit QLineEdit QLineEdit QComboBox QLineEdit QLineEdit QLabel QCheckBox QCheckBox QPushButton QPushButton QPushButton
PopupHomestayAdd	picLbl buttonOK	QPixmap QPushButton
HomestayUpdate	lineEditName lineEditLoc lineEditBed lineEditBath comboBoxType lineEditPrice lineEditCar aLabel checkBoxWifi checkBoxSmoke buttonUpdate buttonReset buttonHome	QLineEdit QLineEdit QLineEdit QLineEdit QComboBox QLineEdit QLineEdit QLabel QCheckBox QCheckBox QPushButton QPushButton QPushButton
HomestayDelete	lineEditPass aLabel buttonCancel buttonConfirm picLbl	QLineEdit QLabel QPushButton QPushButton QPixmap
PopupHomestayUpdate	picLbl buttonOK	QPixmap QPushButton
PopupHomestayDelete	aLabel picLbl buttonOK	QLabel QPixmap QPushButton
<i>Member Profile Modules</i>		
ProfileView	lineEditAddr lineEditFName lineEditLName lineEditPass lineEditPass_2 lineEditPhone	QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit

	lineEditType lineEditUser buttonReward buttonUpdate buttonDelete aLabel	QLineEdit QLineEdit QPushButton QPushButton QPushButton QLabel
ProfileDelete	aLabel lineEditpass buttonReward buttonConfirm picLbl	QLabel QLineEdit QPushButton QPushButton QPixmap
PopupProfileUpdate	picLbl buttonOK	QPixmap QPushButton
PopupProfileDelete	aLabel picLbl buttonOK	QLabel QPixmap QPushButton
RewardView	buttonHome tableView	QPushButton QLabelView
<i>Maintenance Modules</i>		
TicketCreate	lineEdit lineTitle lineType textEdit ticketType buttonHome buttonReset buttonSubmit	QLineEdit QLineEdit QLineEdit QTextEdit QComboBox QPushButton QPushButton QPushButton
RewardCreate	lineDay lineMonth lineYear lineID lineValue rewardType textEdit buttonHome buttonReset buttonSubmit	QLineEdit QLineEdit QLineEdit QLineEdit QLineEdit QComboBox QTextEdit QPushButton QPushButton QPushButton
ReviewCreate	lineBook rating buttonHome	QLineEdit QComboBox QPushButton

	buttonReset buttonSubmit lineDay lineYear lineMont description	QPushButton QPushButton QLineEdit QLineEdit QLineEdit QLineEdit
TicketView	tableView buttonHome	QTableView QPushButton

20. Test cases

Action	Input Values	Expected Output	Actual Output	Pass/Fail	Remarks
Login Panel	Username: mus0 Password: abc123	Entered customer main panel	Logged in as mus0.	✓	
Register	First name: Muhd Last name: Mus E-mail: mm@gmail.com Phone number: 023456789 Username: mus0 Password: abc123 Account type: customer	Registered successfully and data entered In database	Registration successful popup appear.	✓	
Search	Cyberjaya	All homestays with location Cyberjaya	All homestay wil location Cyberjaya appear	✓	
Filter search	Tick with available wifi.	All homestays with wifi available at Cyberjaya	All homestay with available wifi appear	✓	
Create Booking	Check-in date: 2/10/2019 Check-out date: 3/10/2019	Booking is created successful and	Booking is created successfully	✓	

		view booking panel will appear			
View Booking	Click “Bookings” button on Main Menu	Show booking panel of the selected homestay	Booking panel appeared	✓	
Create Payment	Click “Payment” while one of bookings are highlighted	Booking summary will appear	Booking summary appeared	✓	
Apply Reward	Enter the reward id that is available for the user, DQ8U	New booking summary with discounted price appeared.	Popup appeared and price has been discounted	✓	
Payment Gateway	Cardholder Name: Ali bin Abu Card number: 1234567812345678 Expiry Number: 11/22 CVV: 050	Payment Authorization dialog will appear	Payment authorization dialog appeared	✓	
Payment Authorisation	Entered the same value produced by the CAPTCHA	Payment success dialog will appear and data submitted will be inserted into database	Payment successful	✓	

Create Review	Rating: 5 Description: Good place to stay. Recommended to family	Create review success dialog will appear and data submitted will be inserted into database.	Create rating failed	X	run-time error occurred
Delete Booking	Click delete booking and verify password	Booking popup will appear and data in database will be deleted as well.	Delete process failed	X	Delete button disabled.
Create Ticket	Title : Surrounding Type : Homestay Description : Loud surrounding	Create ticket success dialog will appear and data submitted will be inserted into database	Ticket is created and popup appeared	✓	
View Profile	Click "Account"	First name: Muhd Last name: Mus E-mail: mm@gmail.com Phone number: 023456789 Username: mus0	All details are correctly displayed	✓	

		Password: abc123 Account type: customer			
Update Profile	First name: Muhd Last name: Mus E-mail: mm@gmail.com Phone number: 023456789 Username: mus0 Password: abc456 Account type: host	Popup will appear and data in database will be changed to the new data inserted.	Update successfully and popup appeared	✓	
Delete Profile	Enter password : abc456	profile has been deleted successfully and popup will appear	Profile has been deleted successfully and popup appeared	✓	
Login Panel	Username: host Password: host	Entered host main panel	Logged in as host	✓	
View homestays	Click view homestay	List of homestay owned by the host will appear: MutaraVille, Cyberjaya	Shows table of owned homestays	✓	

Add homestay	Name: Arc Homestay ID: autogenerated Location: Cyberjaya Number of bedrooms : 4 Number bathrooms:2 Carpark: 1 Price : 100 Amneties: No wifi	Popup will appear and data will be saved in database.	Created successfully and popup appeared	✓	
Update homestay	Chosen: Arc Homestay Number of bedrooms : 4 Number bathrooms:2 Carpark : 2 Amneties: Wifi available	Popup will appear and new data will be saved in database	Update process failed	✗	Variable does not match.
Delete homestay	Enter password host	Deleted successfully and popup will appear	Deleted successfully and popup appeared	✓	

Create ticket	Title : Jam Type : Booking Description : Booking keep failing	Create ticket success dialog will appear and data submitted will be inserted into database	Ticket submitted	✓	
Login Panel	Username: admin Password: admin123	Entered technical support main panel	Logged in as admin	✓	
Create Reward	Expiry Date: 1/12/2019 RewardType: Discount Value: 10% Description : Enjoy your reward!	Reward is successfully created and popup will appear.	Reward created successfully and popup appeared	✓	