AMERICAN UNIVERSITY OF SHARJAH

College of Engineering

Spring 2024

COE 420 Project Group 9

Final Report

Name	ID
Warda Ul Hasan	g00090299
Hafsah Tahir	g00089492
Jood Shinawi	g00088178
Mohamed Desouki	b00085623
Harsh Surana	b00088424

Instructor: Dr. Hicham Hallal

Submission Date: 12th May 2024

Table of Contents

Detailed Description

The project's objective is to use a web application to automate a laundromat. Initially, the dryers and washers will be upgraded into smart appliances by installing the necessary sensors, which will provide insight on power usage, water levels, and machine status (on/off, cycle, locked door). The sensors will connect to a central system via Wi-Fi or Bluetooth, enabling real-time data collection. This data will enable remote machine health monitoring and offer insights into machine availability, cycle progress, and any problems.

Second, a thorough security system will be put in place. The entire facility will have security cameras installed so that the owner may monitor it in real time. Through the internet, customers can also remotely access their own machines to monitor the status of their cycles.

Both one-time visitors and regular members can book washers and dryers at the automated laundromat for up to 2 consecutive cycles. Initially, users will be prompted to select their preferred date and time for using a washer or dryer. Following this selection, they will be asked to specify their choice between washers and dryers, leading to the display of available machines for the chosen time slot. Users can then pick their desired machine. At this point, the system will verify against the database to ensure that the selected machine hasn't been booked for two consecutive cycles immediately before the new booking request. If such a booking exists, the user will receive a notification advising them to select a different machine or to cancel the booking altogether to prevent exceeding the two-cycle limit. On the other hand, if no conflicting booking is found, the user will be allowed to proceed with reserving the machine for up to two consecutive cycles. Upon successful booking, the user will be notified, and the database will be updated to reflect the new booking status of the machine. This process ensures efficient use of the laundromat's resources while maintaining equitable access for all users.

For payment, users will be provided with two different options with how they can pay for their laundry service: a coin-based method and an online payment method. The client will be asked to choses their payment method when they make their reservation. The coin-based method allows the client to pay for their load on-site and through an interface that is installed on the machines themselves. Before the client is able to start the load on their booked machine, if the payment hasn't already been made, the machine will prompt the user to pay using coins. Once the user inserts the correct amount, the machine will allow the user to start the load.

The other method is the online payment method using Credit/Debit Card. This method will allow users to complete their booking and payment online. After selecting a washer or dryer for a specific time slot, users are redirected to a secure online payment portal. Here, they can enter their credit or debit card details, including card number, expiry date, and CVV. The portal will ensure the security and confidentiality of this sensitive information. Once the user confirms the payment, the system will process the transaction in real time. Following this the machine will be reserved for the specified time slot and the database that will store information about the machine will also update.

The number of times a user has used the service will be kept track of. Clients who have used the web service more than 5 times will be offered rewards in the form of discounts. The value of the discounts can vary depending on how long the client has been a regular user, and the discounts will increase the longer that they have been a regular client. There will be a maximum discount level that a client could receive. These discounts will not be consistently available but loyal clients will have access to them on a rolling basis. The users will be notified when such discounts are available to them. A client does not need to be part of the loyalty program to receive these discounts.

Loyalty Program: Clients engaging with the laundromat can choose to be either one-time visitors or regular members. Regular members can sign up to a rewarding system structured around accumulating points after each usage. As patrons use the facility, whether for washing or drying services, their participation contributes to point accumulation. Regular members benefit from this system as points accumulate with every visit, gradually unlocking exclusive discounts, special offers, or other perks. This incentivizes ongoing patronage and fosters a sense of loyalty among customers. Additionally, the program serves as a means of appreciation for frequent users, encouraging them to continue utilizing the services while simultaneously enhancing overall customer satisfaction and retention.

Information about machines, clients, and utility bills will be stored in the Firebase Realtime database. There will be three tables: Machine, Client, and Utility_Bills. The Machine table will be designed to track how frequently the machine is used per day, the times when the machine will be sent for maintenance, and how much revenue is earned by a machine. This table will include the following fields: ‘Bookings’, ‘Usage_Statistics’, ‘Maintenance_Schedules’, and ‘Revenue_Generated.’ The Client table will include information about the users of the laundromat system, and it will also differentiate between one-time visitors and regular members. The fields in this table will include ‘Client_ID’, ‘Client_Type’ (indicating visitor or regular member), ‘Name’, ‘Contact_Details’, and ‘Reward_Points’ (for the regular visitors). The Utility_Bills table will include details on expenditures on water and electricity. This table will include fields: ‘Bill_Type’, ‘Amount’, and ‘Billing_Date’ columns, providing vital insights into the laundromat's operational expenses.

The facility will guarantee uninterrupted internet service through connections from different providers and high-availability network equipment. Continuous monitoring will detect and resolve any connectivity issues promptly. This reliable internet access is crucial for real-time machine monitoring, booking, payment processing, security camera surveillance, and customer interaction through the web application. By prioritizing reliable internet infrastructure, the laundromat ensures seamless service delivery and customer satisfaction.

Lastly, once a customer has paid for the services they have utilized, the website will allow them to leave reviews on the machines that they used, the facility and their experience with the service as a whole. After the user has submitted the review, they will have the option to modify or delete their review. These reviews will be posted publicly on the website and will list the username of the user, as well as the date they left the review and the machine that they used. Any administrator to the system will be notified of any review that is left on the site and will be able to respond to the reviews left on the website.

Functional and Non-Functional Requirements

Functional Requirements:

FR1. The system should allow the client to book a specific machine (washer and or dryer).

FR2. The system should allow clients to pay using one of two modes of payment: classical coin-based method and online payment.

> **FR2.1.** The system should provide a payment interface for the coin-based method.
>
> **FR2.2.** The system should provide an online payment interface for using a credit/debit card.
>
> **FR2.3.** The system should provide a payment confirmation after payment is done.
>
> **FR2.3.** The system should wait until payment is made before the machine can be used.

FR4. The system should keep a database of information on machines (records of usage, maintenance, revenue, etc.), clients, and bills (water, electricity).

FR5. The system should work based on a reward system (points awarded to clients, specifically members, after each usage).

FR6. The system should allow clients to provide feedback on the machines and the facilities.

> **FR6.1** The system should allow the administration to respond to reviews.

FR7. The system should provide the availability of the machines' statuses.

> **FR7.1.** The system should have each machine equipped with the necessary sensors to read its status (ON/OFF).
>
> **FR7.2.** The system should have each machine equipped with the necessary sensors to read the amount of power consumed.
>
> **FR7.3.** The system should have each machine equipped with the necessary sensors to read locked/unlocked.
>
> **FR7.4.** The system should have each machine that provides a payment confirmation after payment is done through a monitor interface.

FR8. The system should display notifications to clients for reservation confirmations and machine availability.

FR9. The system should not allow a client to book more than 2 consecutive cycles per machine.

Non-Functional Requirements:

NFR1. The facility should be under 24/7 surveillance using cameras.

NFR2. The system should support both one-time visitors and registered members.

NFR3. The system should provide all users with web-based access to the system.

> **NFR3.1.** The system should provide role-based access to the system (administrators and clients).
>
> **NFR3.2.** The system should provide login and logout functionalities for all users.

NFR4. The system should be accessible from all popular web browsers (Chrome, Firefox, Internet Explorer).

NFR5. The system should be accessible from all popular mobile devices.

NFR6. The system should be able to support at least 25 new registrations per hour.

NFR7. The user interface should be responsive to laptops, mobile phones, and tablets with different screen sizes.

NFR8. The retrieval of data from the database should be within 6 seconds.

NFR8. The system should have uninterrupted internet service.

NFR9. The system should be available 24/7

> **NFR9.1.** The system will not be available during scheduled maintenance.
>
> **NFR9.2** The system should be covered by an uninterrupted internet service to allow continuous access to the facility's services.

NFR10. The website must have a user-friendly interface.

Use Case Diagram and Descriptions

Use Case Diagram

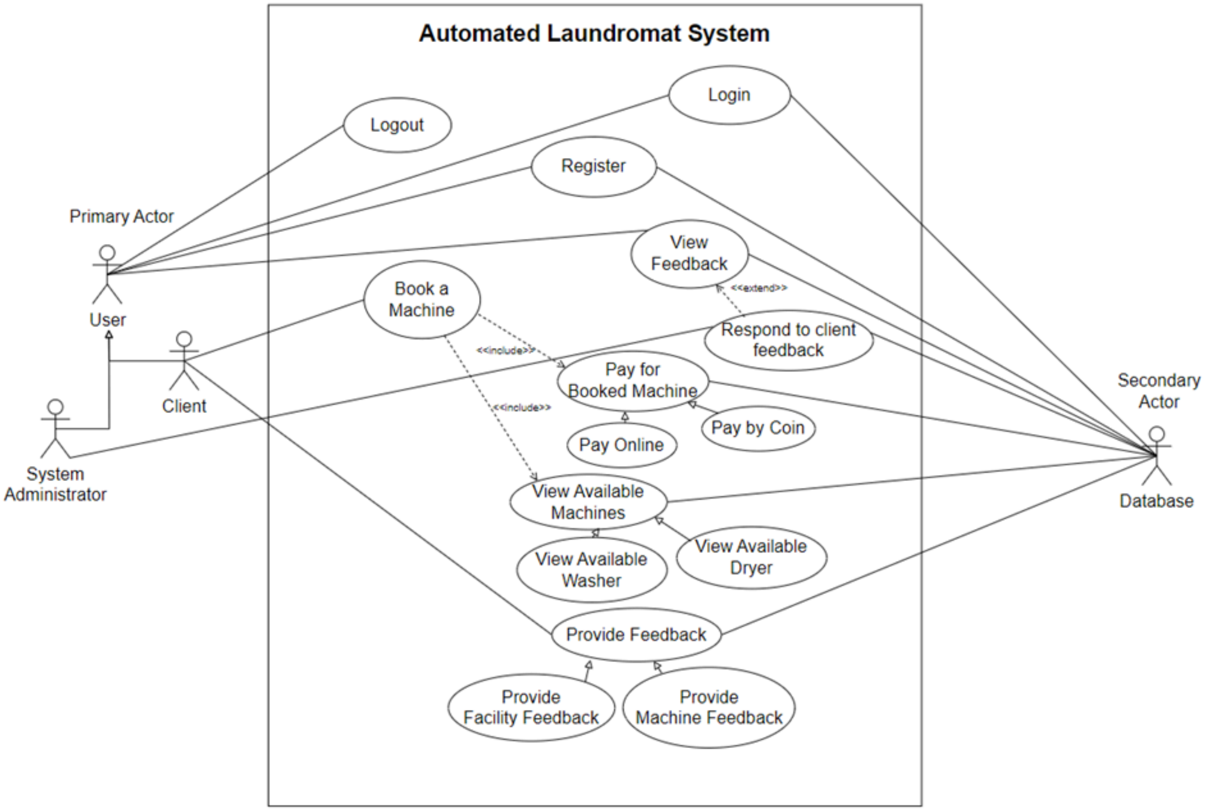


Figure 1: Use Case Diagram

Use Case Descriptions

1. Login

Use-Case ID:	001		
Use-Case Name:	Login		
Created by:	Jood	Last Updated By:	Jood
Date Created:	09/03/2024	Date Last Updated:	09/03/2024

Actors:	User, Database
Description:	This use case describes the process by which a user logs into the system.
Preconditions:	The user/system administrator must have a registered account on the website.
Postconditions:	Once the user/system administrator enters the right username/email and password , the user is redirected to their account.
Normal Flow:	1.1 The user/system administrator navigates to the login page of the washing machine booking website. 1.2 The system displays input fields for email and password. 1.3 The user/system administrator enters their registered email and password. 1.4 The entered email and password are then verified by through the user details stored in the Database. 1.5 The user/system administrator clicks on the Login button. 1.6 The system verifies the entered credentials. 1.7 If the credentials are correct, the system allows access to the user/system administrator.
Alternative Flow:	No Alternative Flows.
Exceptions Flow:	1. If the user/system administrator enters an incorrect email format, email, or password: a. The system displays an error message to the user indicating invalid credentials. b. The user/system administrator is asked to re-enter their username/email and password. c. The flow returns to step 3 of the normal flow.

2. Logout

Use-Case ID:	002		
Use-Case Name:	Logout		
Created by:	Jood	Last Updated By:	Jood
Date Created:	09/03/2024	Date Last Updated:	09/03/2024

Actors:	User
Description:	This use case describes the process by which a logged-in user logs out from the system.
Preconditions:	The user/system administrator must be logged into their account on the website.
Postconditions:	Once the user/system administrator logs out, he/she is redirected to the login page.
Normal Flow:	2.1 The user is currently logged into their account. 2.2 The user navigates to the logout button, found by clicking on their profile icon. 2.3 The system terminates the user's session and logs them out. 2.4 The system redirects the user to the login page.
Alternative Flow:	No Alternative Flows.
Exceptions Flow:	No Exceptions.

3. View Available Machines

Use-Case ID:	003		
Use-Case Name:	View Available Machines		
Created by:	Hafsah	Last Updated By:	Hafsah
Date Created:	18/03/2024	Date Last Updated:	18/03/2024

Actors:	Client, Database
Description:	The user will be able to view any available washing/drying machines at the facility at any given time
Preconditions:	The user should be logged into their their account, if they are a registered user, else they should select the option to continue as a guest
Postconditions:	Once they have viewed the available machines, they will have the options to book the machine
Normal Flow:	3.1. The user will choose the time for which they would they would like to see the availability of the machines 3.2 The availability of the machines at the selected date and time are retrieved from the Database. 3.3. The website will display all of the machines that are available at the time selected 3.4. The user will be able to select a machine and see how long it is available for
Alternative Flow:	No alternate flow
Exceptions Flow:	No exceptions

4. Book a Machine (Schedule a Booking)

Use-Case ID:	004		
Use-Case Name:	Book a Machine		
Created by:	Warda	Last Updated By:	Warda
Date Created:	12/03/2024	Date Last Updated:	31/03/2024

Actors:	Primary: Client Secondary: Database System
Description:	This use case describes the process by which a user books a specific machine (washer or dryer) for up to 2 consecutive cycles.
Preconditions:	1. The user must be logged into the system (registered member) or proceed as a guest (one-time visitor). 2. The machine must be available for booking. 3. The Database is accessible and up-to-date on the machine's usage, maintenance, and revenue records.
Postconditions:	1. The machine is booked for a specified time and for at most 2 consecutive cycles. 2. Once the booking is confirmed the user receives a notification of their reservation. 3. The booking details are stored in the Database.
Normal Flow:	4.0 The user selects the “Book a Machine” option on the homepage. 4.1 The system displays a calendar for the user to choose a date and time for the booking. 4.2 The user selects their preferred date and time. 4.3 The user will select to view the available machines. 4.4 The system prompts the user to choose between a Dryer or a Washer. 4.5 The user selects their preferred machine. 4.6 The system checks the Database for available machines during the selected date and time. 4.7 The system displays the available machines to the user. 4.8 The user picks a machine from the displayed options. 4.9 The system prompts the user to select at most 2 consecutive cycles. 5.0 The user inputs the desired number of consecutive cycles (1 or 2). 5.1 The system notifies the user that the machine is booked and proceeds to the next step to pay for the booked machine.
Alternative Flow:	1. When the user chooses to view available machines for their preferred machine type (either Dryer or Washer), the website indicates that there are no available machines of their preferred type for the selected time slot.

	2. The user then selects a different date and time to find an available machine, and the process described in step 4.2 is repeated.
Exceptions Flow:	4.0.E.1 If the user inputs a number other than 1 or 2 for the number of cycles, the website prompts the user to input either 1 or 2, as those are the only valid options.

5. Provide Feedback

Use-Case ID:	005		
Use-Case Name:	Provide Feedback		
Created by:	Hafsah	Last Updated By:	Hafsah
Date Created:	19/03/2024	Date Last Updated:	19/03/2024

Actors:	Client, Database
Description:	After using the web service, the client will be able to leave a review on their experience with the machine, facility and user interface as well.
Preconditions:	Clients should have utilized the system to book a washing/drying machine and have paid for the service before they can leave a review.
Postconditions:	The review will be posted on the website.
Normal Flow:	5.1. Click on “Submit feedback” 5.2. User will be prompted to choose between machine feedback 5.3. User will be prompted to enter the booking reference 5.3. They will fill in their name 5.4. They will write their review 5.5. They will submit the review 5.6 The review will then be stored in the database
Alternative Flow:	5.2. User will be prompted to choose between facility feedback
Exceptions Flow:	No exceptions

7. Pay for Booked Machine(s)

Use-Case ID:	007		
Use-Case Name:	Pay for Booked Machine(s)		
Created by:	Harsh	Last Updated By:	Harsh
Date Created:	10/03/2024	Date Last Updated:	10/03/2024

Actors:	Primary: Client Secondary: Database
Description:	This use case details how a user pays for booked laundry machines using a payment interface located on each machine. The user can choose between coin payment and online payment options. Payment details are processed and recorded in the database.
Preconditions:	1. User must have acceptable payment means. 2. The machine's payment interface should be operating properly. 3. The Database is accessible and up-to-date on the machine's records.
Postconditions:	1. Successful payment grants the user access to the machine for a specified duration.
Normal Flow:	7.1. User approaches the machine and initiates the payment interface. 7.2. The payment amount is displayed. 7.3. User selects payment method and completes the transaction. 7.4. The interface confirms the transaction. 7.5. The database records the transaction and updates the user's reservation status and reward points if applicable. 7.6. User accesses the machine.
Alternative Flow:	1. Insufficient coins lead to payment through the web application.
Exceptions Flow:	7.0.E.1 If the payment is not successful: The system notifies the user of the unsuccessful payment and prompts them to try again. 7.0.E.1. Booking issues trigger error signals and customer support options.

8. Pay Online

Use-Case ID:	008		
Use-Case Name:	Pay Online		
Created by:	Harsh	Last Updated By:	Harsh
Date Created:	10/03/2024	Date Last Updated:	10/03/2024

Actors:	Primary: Client Secondary: Database
Description:	Users complete online payments for services or products. The process is logged and tracked in the database.
Preconditions:	1. The user must have access to an internet-connected device and a valid payment method (credit/debit card, online banking, e-wallet, etc.) 2. The website or app must support online transactions.
Postconditions:	1. The payment is processed successfully, and the database records the transaction 2. The user receives confirmation of the transaction.
Normal Flow:	8.1. User selects a product/service and chooses online payment. 8.2. User is redirected to the payment gateway. 8.3. Payment method and details are entered and confirmed. 8.4. Transaction processed and confirmed to the user. 8.5. Database updates transaction and reservation details, along with user rewards if registered.
Alternative Flow:	1. If the online payment method is not available: The user needs to proceed to pay by coin.
Exceptions Flow:	8.0.E.1 If the transaction fails (due to insufficient funds, incorrect details, etc.): 1- The payment gateway displays an error message. 2- The user is prompted to re-enter payment details or choose a different payment method.

9. Pay by Coin

Use-Case ID:	009		
Use-Case Name:	Pay by Coin		
Created by:	Harsh	Last Updated By:	Harsh
Date Created:	30/03/2024	Date Last Updated:	30/03/2024

Actors:	Primary: Client Secondary: Database
Description:	Users pay for laundry machines using coins at each machine. Transactions and machine access are managed and recorded by the database.
Preconditions:	1. The user must have an adequate amount of coins acceptable by the machine 2. The payment interfaces at the laundromat must be operational and capable of accepting and processing coin payments
Postconditions:	1. The payment is processed successfully 2. The user gains access to the booked laundry machine for a specified duration.
Normal Flow:	9.1 The user approaches the payment interface with the booked machine number. 9.2 The user selects the option to "Pay by Coin" on the payment interface. 9.3 The machine displays the amount due for the booked machine. 9.4 The user inserts the required coins into the machine. 9.5 The payment interface validates the coins and processes the payment. 9.6 The machine confirms the successful transaction. 9.7 Database updates transaction and reservation details, along with user rewards if registered. 9.8 The user is granted access to use the booked laundry machine for a specified duration. 9.9 The payment machine updates the booking status on the laundromat's system.
Alternative Flow:	1- If the user does not have enough coins: 1.1. The user opts to pay through the web application. 1.2. The flow then continues from the relevant step in the "Pay Online" use case.

Exceptions Flow:	9.0.E.1 If the payment machine does not accept the coins (due to invalid coins, machine error, etc.): a. The machine rejects the payment and prompts the user to try again with different coins or use the web application. b. The flow returns to step 9.4 or redirects to the "Pay Online" use case.

Class Diagram

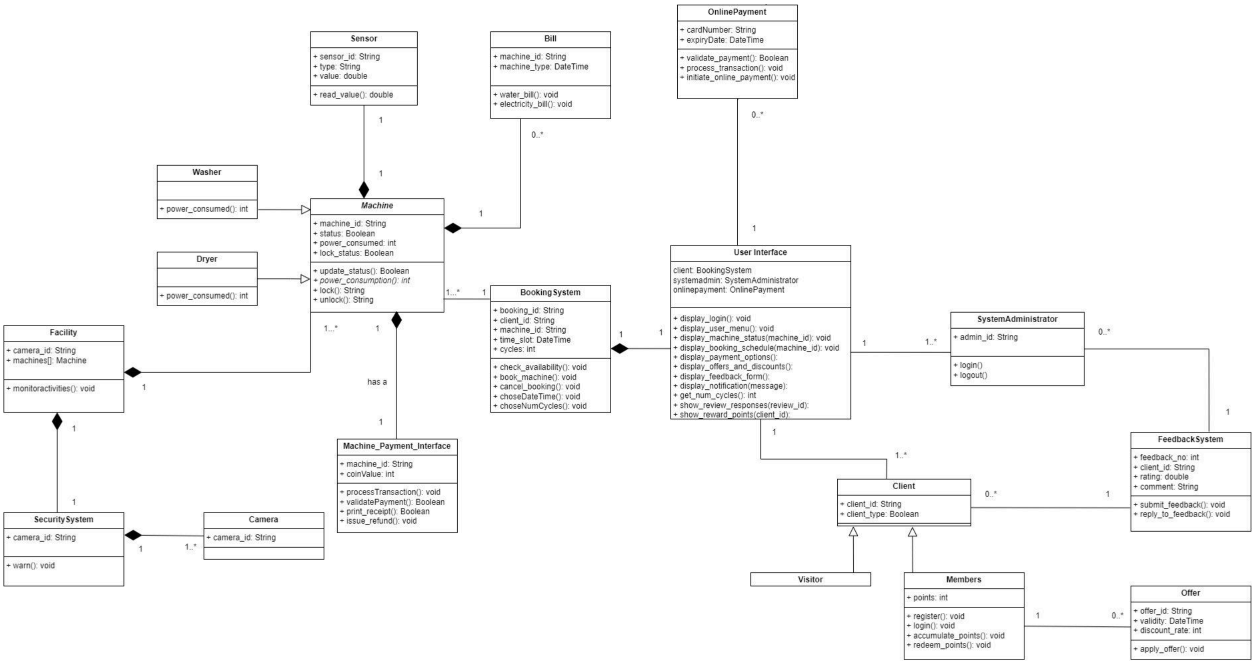


Figure 2: Class Diagram

Domain Model

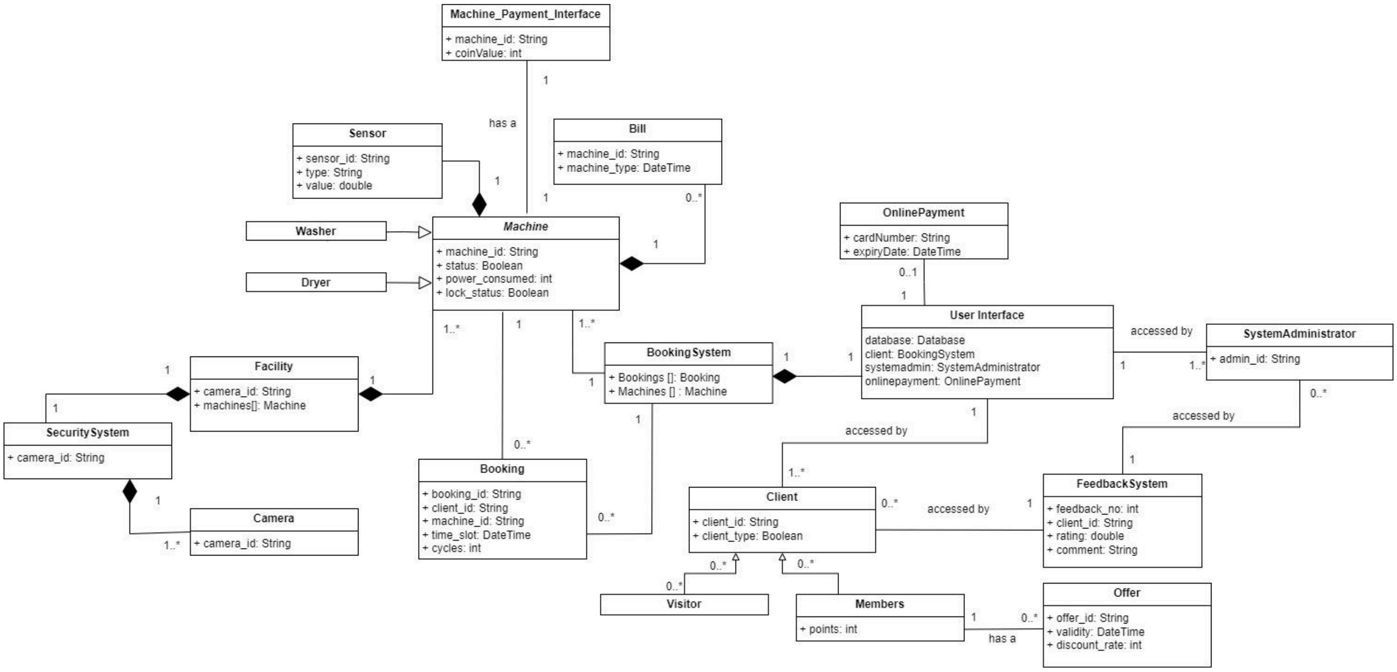


Figure 3: Domain Model

Sequence Diagrams

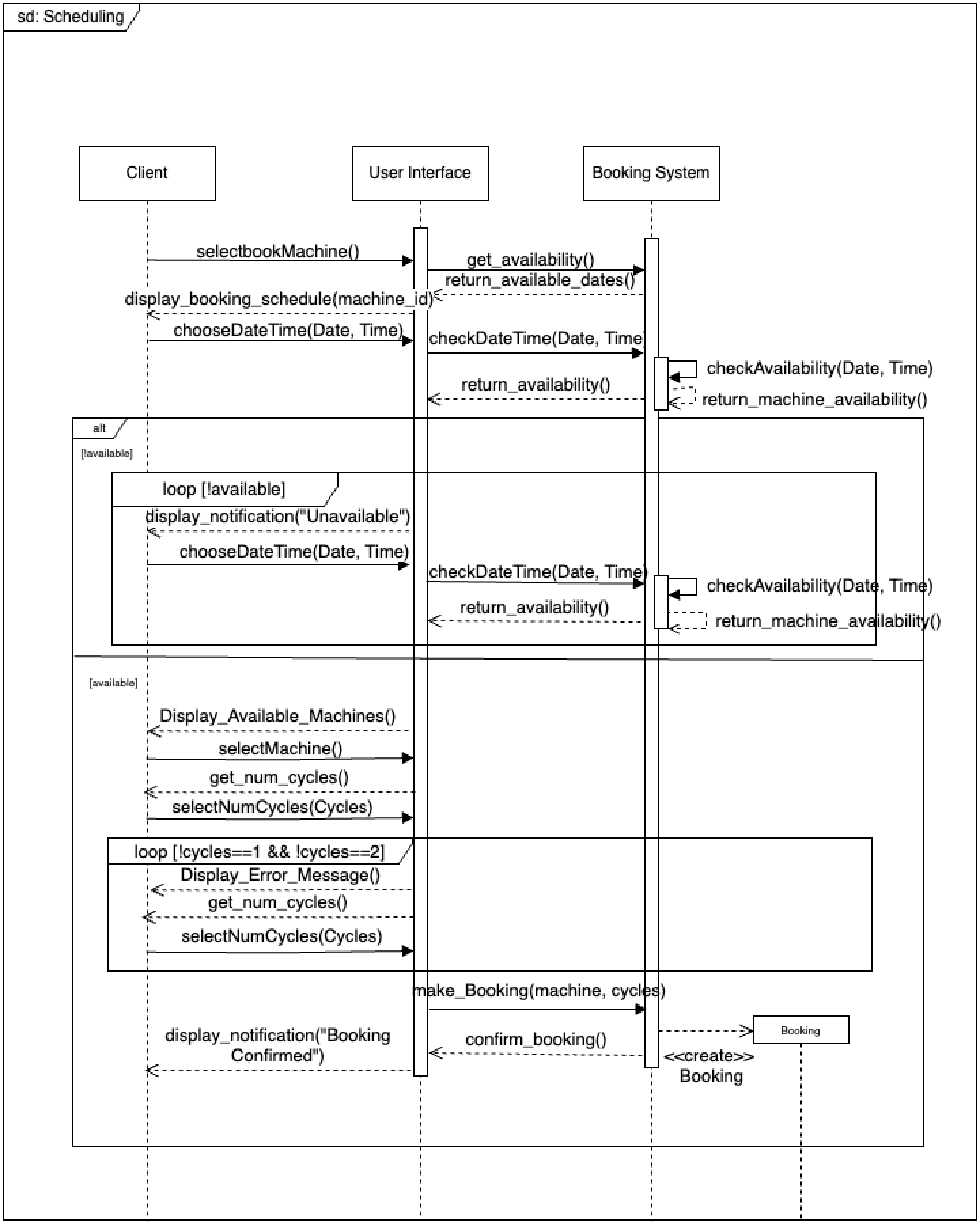

sd: Scheduling
Client
User Interface
Booking System
selectbookMachine()
get_availability()
return_available_dates()
display_booking_schedule(machine_id)
chooseDateTime(Date, Time)
checkDateTime(Date, Time)
checkAvailability(Date, Time)
return_availability()
return_machine_availability()
alt
[!available]
loop [!available]
display_notification("Unavailable")
chooseDateTime(Date, Time)
checkDateTime(Date, Time)
checkAvailability(Date, Time)
return_availability()
return_machine_availability()
[available]
Display_Available_Machines()
selectMachine()
get_num_cycles()
selectNumCycles(Cycles)
loop [!cycles==1 && !cycles==2]
Display_Error_Message()
get_num_cycles()
selectNumCycles(Cycles)
make_Booking(machine, cycles)
display_notification("Booking Confirmed")
confirm_booking()
Booking
<<create>> Booking

Figure 4: Sequence Diagram for Scheduling

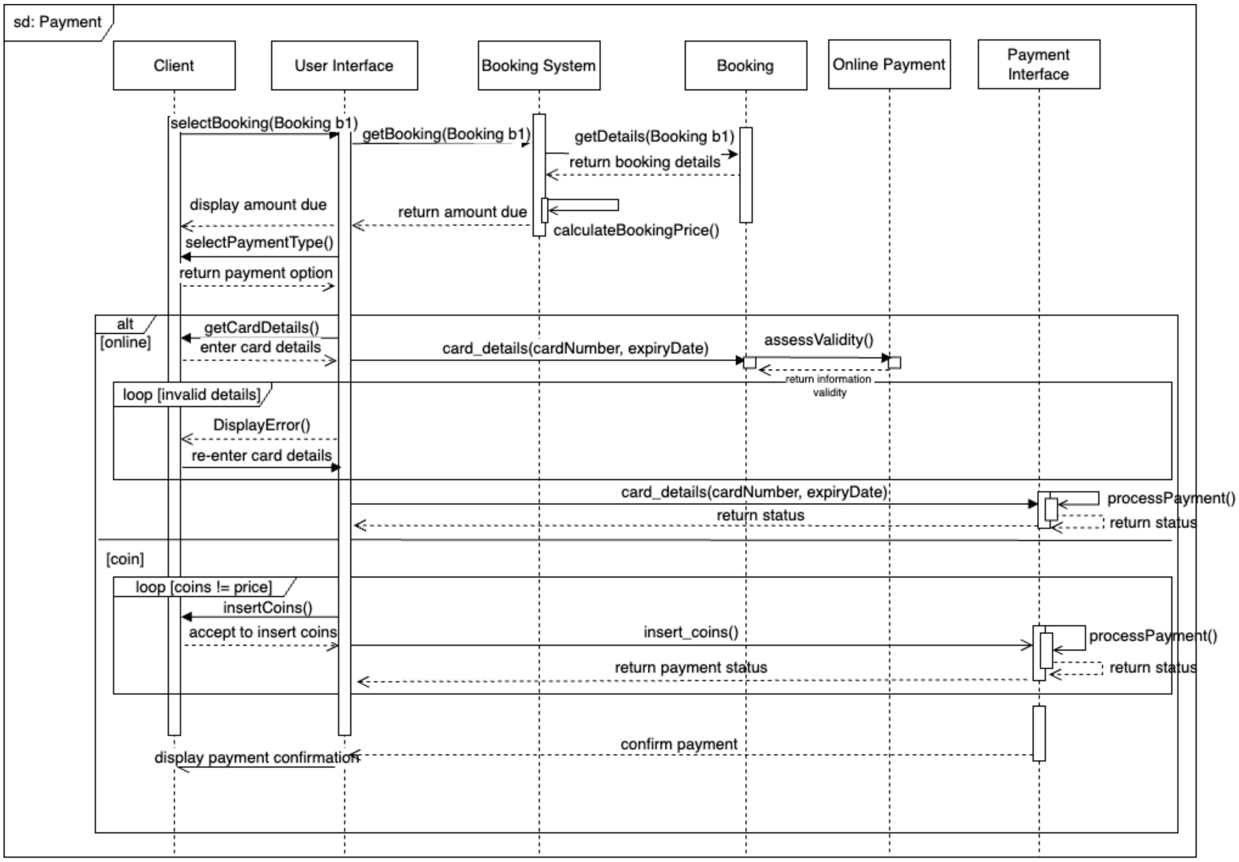


Figure 5: Sequence Diagram for Payment

Testing Plan

Test Cases

1. Login

Test Case ID: TC001

- **Objective**: To verify that a user can successfully log into their account
- **Precondition**: None
- **Test Steps**:
    1. Enter user credentials
    2. Click sign in
    3. User will come to the homepage of the system
- **Expected Result**: The user will be signed in and will be at the homepage of the website.

2. Booking a Machine

Test Case ID: TC002

- **Objective**: To verify that a user can successfully book a machine for up to 2 consecutive cycles.
- **Precondition**: User must be logged in, machine must be available.
- **Test Steps**:
    1. Navigate to the "Book a Machine" section.
    2. Select an available machine and a valid date and time.
    3. Select 1 consecutive cycle.
    4. Confirm the booking.
- **Expected Result**: The booking is confirmed and the user receives a notification; database updates with the new booking.
    1.
- **Expected Result**: The user is informed that the machine is unavailable and prompted to choose another machine or time.

Test Case ID: TC003

- **Objective**: To test the system's handling of a booking request when the machine is not available.
- **Precondition**: User must be logged in, desired machine is already booked for the selected time.
- **Test Steps**:

    1. Log in as a user.
    2. Navigate to the "Book a Machine" section.
    3. Select a machine that is not available at the desired time.
    4. Attempt to book.
- **Expected Result**: The user is informed that the machine is unavailable and prompted to choose another machine or time

3. Payment Confirmation

Test Case ID: TC004

- **Objective**: To verify that the revenue is updated in the database after a payment is made.
- **Precondition**: Booking is made, payment is due.
- **Test Steps**:
    1. Complete a booking that requires payment.
    2. Select the payment method and complete the transaction.
    3. Check the database for the revenue update related to the machine.
- **Expected Result**: The revenue data for the machine in the database should reflect the payment made.

4. Reward Points Allocation

Test Case ID: TC005

- **Objective**: To ensure that reward points are correctly allocated to frequent users after completing a transaction.
- **Precondition**: User has completed multiple bookings and transactions.
- **Test Steps**:
    1. Log in as a user who has completed the threshold number of transactions to qualify for rewards.
    2. Complete a booking and payment.
    3. Check user profile for updated reward points.
- **Expected Result**: Reward points are correctly updated in the user's account according to the loyalty program rules.

5. Leave Feedback

Test Case ID: TC006

- **Objective**: for clients to leave feedback on machines
- **Precondition**: User has completed bookings and transactions.
- **Test Steps**:
    1. Log in as a user
    2. Navigate to the “Submit Feedback” page.

   3. Enter name.
   4. Enter booking ID.
   5. Submit feedback.
- **Expected Result:** Feedback will be saved on the website.

6. Only Admins will be able to respond to Feedback

Test Case ID: TC007

- **Objective**: for only admins to respond to feedback and other users (visitors and regular members) will not be able to respond.
- **Test Steps**:
   1. Log in as an admin user
   2. Navigate to the “View Feedback” page.
   3. Click on the Respond button.
   4. Enter response.
   5. Response displayed with the feedback.
   6. Log in as a normal user (visitor or regular member)
   7. Navigate to the “View Feedback” page.
   8. There will be no Respond button for the user

Implementation Details

We developed the Automated Laundromat website using Flutter and integrated it with a Firebase database for managing the data. In the source_code_COE420.zip folder that is attached along with our submission, there are multiple elements:

1. ***.dart_tool, .flutter-plugins, .flutter-plugins-dependencies, .metadata, analysis_options.yaml, automated_laudromat.iml, pubspec.yaml, and pubspec.lock*** are Flutter and Dart configuration files.
2. ***android, ios, linux, macos, windows, and web*** contain code for building the app on different platforms.
3. ***.firebaserc, firebase.json, database.rules.json, firestore.indexes.json, firestore.rules, and storage.rules*** are Firebase configuration files.
4. Next, are the code and resource directories:
    a. ***lib***, which contains the Dart Code for the Flutter App.
    b. ***fonts***, which is a directory for custom fonts used in the app
    c. ***test***, which contains test files for the project.
5. Other Files and Directories include ***functions***, which contains the Firebase Cloud Functions, ***build***, which contains output files from building the app. ***README.md***, which is a markdown file with information about the project and ***.gitignore***, which specifies intentionally untracked files to ignore.

For this section, we will dive deeper into the Dart Code used for the Flutter App, found in the ***lib*** directory:

1. ***"featurewidget.dart":*** The FeatureItem class defined is a Flutter widget that uses the Material Design elements to display a feature as an icon followed by a title. The FeatureItem has two must-have properties: a text field and an icon. The constructor initializes these properties and, if necessary, accepts a key and passes it to the superclass. In the build method, the widget is composed using a Column, which vertically stacks its child widgets. It shows an icon at the top that is 40 units in size and has the provided icon data. Just below the icon, there is a vertical spacer (SizedBox) with a height of 10 units, and then the Text widget that displays the title with a font size of 16.
2. ***"BookingPage":*** Defines a BookingPage Flutter widget that facilitates machine bookings for a laundromat by allowing users to select a date, time, and machine type through a series of interactive dialogs. The BookingPage Class extends StatefulWidget, meaning that its state can change over time. The BookingPage class initializes its state in the _BookingPageState class, which is where the logic of the BookingPage widget is handled. It defines various properties such as selectedDate, selectedTime, focusedDay, and others that manage the state related to the booking details.The _pickTime method displays a dialog with selectable times. When a time is selected, it updates the selectedTime state and clears any previously available machines.The widget uses TableCalendar for date selection.After selecting a date, customers are presented with another dialog box (implemented in _selectMachineType) where they can choose the type of machine (dryer or washer). Using the chosen date, time, and machine type, the _viewAvailableMachines method queries the Firebase database to find any available machines. If no machines are available, it informs the user; otherwise, it displays the

available machines. In the _selectCycles method, users can choose a machine and a cycle count. Then, in order to complete the booking, this method calls the _bookMachine method, which will increment the booking count and update the Firebase database to reflect the booked machine. By interacting with Firebase via the FirebaseDatabase.instance.ref() function, the application can access and modify booking data that is kept in a structured fashion. Firebase is used to maintain booking counters, machine availability, and booking details. The _showMessage method uses ScaffoldMessenger to manage problems and show the user any messages or errors that occur, like when an incorrect amount of cycles is input or when there are no machines available. Following a machine booking that is successful, the user is sent to the PaymentPage and receives a confirmation message containing the booking ID. The user interface is made up of a calendar for choosing dates, buttons to view available machines and change the time and date, and dynamic sections that show the selected time and date.

3. ***"Feedback.dart":*** The FeedbackPage Flutter widget is designed to collect and manage feedback from users regarding facilities and machines. To maintain its state and deal with the dynamic nature of user interactions, this widget makes use of a stateful structure. The widget uses a post-frame callback to initiate _showFeedbackOptions after initializing the state in _FeedbackPageState. This guarantees that the dialog appears following the completion of the widget builds. Feedback Type Selection: The _showFeedbackOptions method displays a dialog where users can choose between 'Facility Feedback' or 'Machine Feedback'. Depending on the choice, it will additionally ask for user input via _getPublicName, letting users type a name that will be shown with their comments in public. Users are asked for feedback regarding either the facility or a particular machine, depending on whatever feedback type they have chosen. Users then need to provide a booking ID in order to obtain the unique machine ID linked to their reservation when requesting machine feedback. Feedback for both kinds is entered into a Firebase database once the required data has been gathered, including the booking ID, public name, and feedback text. To store feedback and control counters for feedback entries, the widget communicates with Firebase. It maintains feedback details under structured paths that include the feedback type, user's public name, and special details like machine ID for machine feedback. It retrieves and updates feedback counts using routes like Facility_Feedback_Count and Machine_Feedback_Count. After submitting feedback, users are shown a confirmation message via a snackbar.
4. ***"FeedbackListAdminPage.dart" & "FeedbackListPage.dart"*** : Admin users can view and reply to user feedback by using the `FeedbackListAdminPage` widget. The `FeedbackListPage` widget is intended to show user input about the machines and facilities on the website. For real-time data handling, feedback entries are collected from two different categories: Facility Feedback and Machine Feedback, using Firebase. Through a `StreamBuilder` that monitors changes in Firebase database nodes, the widget is designed to enable administrators to view feedback in real-time. Every feedback entry is shown as a `FeedbackItem`, a customized widget that displays the user's input together with the type of feedback (about a particular machine or a facility) and any administrative responses that have been given. It defaults to displaying "No admin response yet" if no response has been received. Administrators can reply directly to user input from within the app using dialogues that allow them to enter and submit their responses, which are then updated in the Firebase database.

5. ***"DefaultFirebaseOptions":*** The `DefaultFirebaseOptions` class serves as a setup tool for configuring Firebase on various platforms. For web, Android, iOS, macOS, and Windows platforms, this class incorporates predefined Firebase options by specifying parameters like the API key, app ID, messaging sender ID, project ID, auth domain, database URL, and storage bucket. These parameters are essential to link the app to the appropriate Firebase project instance. Depending on the platform identified at runtime, the `currentPlatform` getter method dynamically returns the appropriate Firebase settings. This function makes use of Flutter's `defaultTargetPlatform} and `kIsWeb} to determine the application's current operating environment. The method throws an `UnsupportedError` and prompts to specify Firebase options through the FlutterFire CLI again if the platform is unsupported (like Linux, in this example) or if a default option is not defined for the platform.
6. ***"LoginPage.Dart":*** Users wishing to access an Automated Laundromat service can utilize the 'LoginPage' widget as an authentication interface. It provides two separate login experiences: one for visitors and one for regular members. Regular members use Firebase Authentication to validate their password and registration email when logging in. The user is forwarded to the 'RegularMemberPage' after successful authentication. In the event of authentication problems, such as an incorrect password, an invalid email address, or a user not found, the widget uses a SnackBar to display specific error messages. For visitors, the system assigns them a unique visitor ID based on the timestamp and directs them to the 'VisitorPage'. Moreover, the page offers a registration link that takes new regular members to the `RegistrationPage'.
7. ***"main.dart":*** The `main.dart` file establishes the application's entry point and initializes and configures Firebase services. The first step is to make sure the Flutter bindings are initialized correctly to allow for appropriate platform interaction. A generic initialization serves all other platforms, but the Firebase initialization is conditionally designed for web platforms with particular settings. The 'MyApp' widget is used when Firebase has been configured. The theme and home widget of the program are specified by the 'MyApp' class, which specifies the application root. To maintain user sessions, the home widget employs a `StreamBuilder` to monitor changes in the state of Firebase Authentication. Whether or not a user is logged in determines which user type (Visitor or Regular Member) the stream pulls from Firebase Realtime Database, or if no user is identified, it redirects to the 'LoginPage'. Depending on their type (which can be determined from the database records), users are led to either the 'RegularMemberPage' or the 'VisitorPage'.
8. ***"PaymentPage":*** The `PaymentPage` widget manages the automated laundromat payment process. When it first initializes, it retrieves and shows booking information such as the machine ID and cycle count. It also determines the payment amount based on the type of machine (dryer or washer). Users have the option of paying online with a card or with coins. When making a coin payment, the widget counts the coins inserted and prompts the user until the required amount is entered. Once the required amount is entered, the payment is processed, the machine's revenue is updated, and reward points are awarded based on the number of cycles. Else, for online payments, a dialog collects card details (card number, expiry date, and CVV). When this validation is successful, it updates the revenue and reward points and proceeds to process the payment in a similar manner. Both payment options then update the Firebase database to reflect the extra revenue and adjust the user's reward points. Through snack bars, the widget also offers

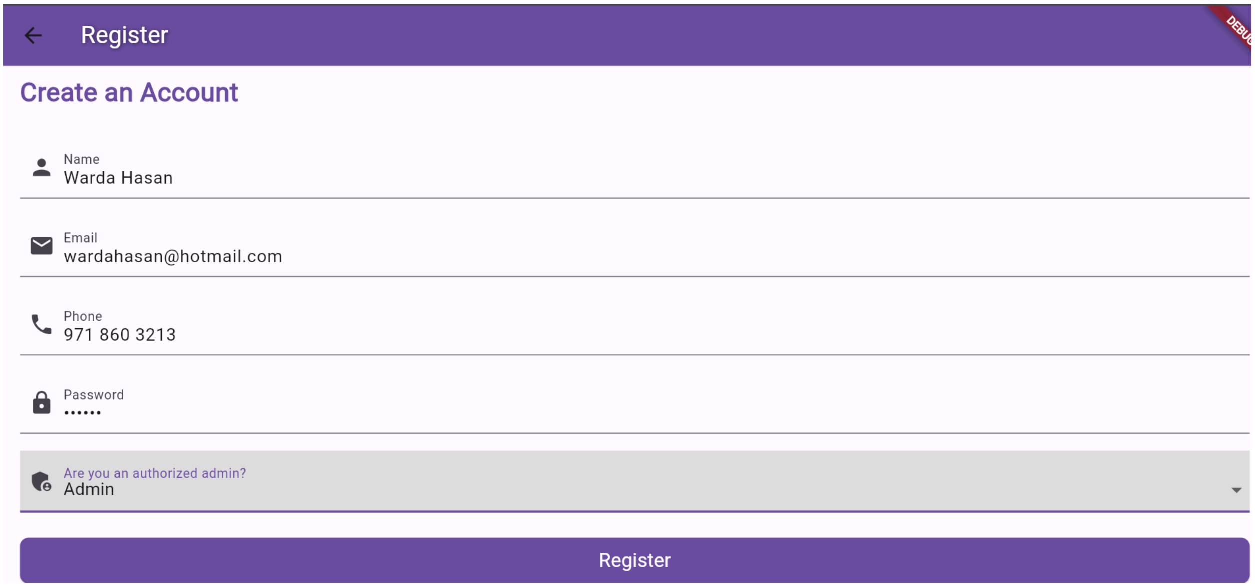
the user instantaneous feedback for a variety of actions, such as successful payments or data entry problems.

9. ***"RegistrationPage.Dart":*** The RegistrationPage widget makes it easier for users to register. By entering their name, email address, phone number, and password, new users can create an account. Firebase Authentication facilitates the establishment of user credentials, and the Firebase Realtime Database stores user information such as the generated unique client ID and initial reward points. Users can also select their client type, choosing between 'Regular Member' and 'Admin'. This choice is reflected in the database entry, allowing for differentiated access levels within the application based on user roles. A counter in the database is increased by the _generateClientId method to guarantee that every user is assigned a distinct client ID that is zero-padded and prefixed for uniformity. After a successful registration, the user is automatically logged in with the credentials they have specified and their information is saved under a new database entry. Errors are logged to the console if any step of the registration procedure is unsuccessful.
10. ***"RegularPage":*** The primary user interface for regular members. It displays the user's name, reward points, and user type, offering a personalized experience. Firebase Realtime Database is used to retrieve user-specific data, while Firebase Authentication is used for user management.The name and reward points of the logged-in user are shown on the page. Additionally, it has buttons for scheduling a machine, leaving comments, and viewing feedback (admins are directed to 'FeedbackListAdminPage', while regular members are directed to 'FeedbackListPage'). It also highlights key aspects of the laundry service, such as quick booking, various payment choices, and round-the-clock monitoring using the 'FeatureItem' widgets, along with pricing details per machine. Furthermore, users can sign out using an integrated logout feature, which returns them instantly to the login page and guarantees a proper session termination.
11. ***"VisitorPage":*** The primary user interface for visitors. Upon entry, the page welcomes the visitor with a greeting and displays buttons for interaction. It includes buttons for scheduling a machine, leaving comments, and viewing feedback (visitors are directed to 'FeedbackListPage'). It also highlights key aspects of the laundry service, such as quick booking, various payment choices, and round-the-clock monitoring using the 'FeatureItem' widgets, along with pricing details per machine. Furthermore, visitors can sign out using an integrated logout feature, which returns them instantly to the login page and guarantees a proper session termination.

Testing Results

1. Login – Test Case ID: TC001

First, a user has to create an account and fill in their credentials and whether they are logging in as an admin or user. Figure 6 shows this. Once the account has been created, the database will record the credentials. This is shown in figure 7.



Register

Create an Account

Name: Warda Hasan

Email: wardahasan@hotmail.com

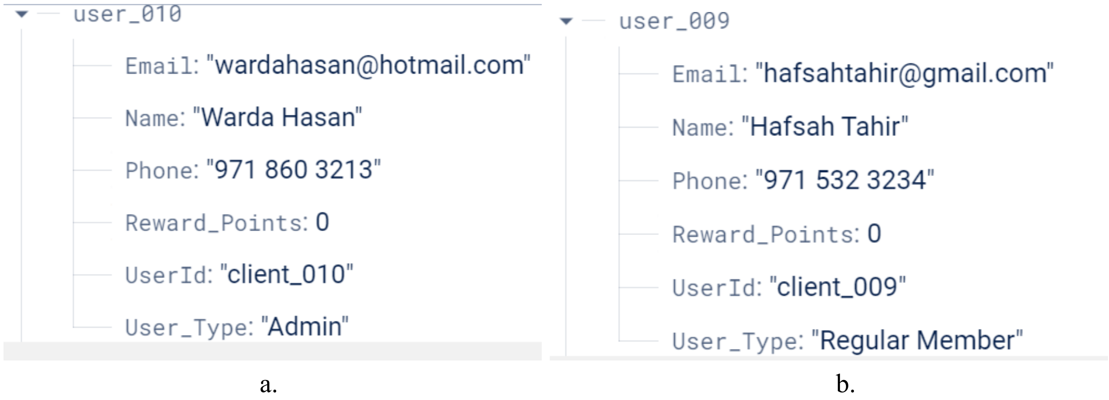
Phone: 971 860 3213

Password: ••••••

Are you an authorized admin? Admin

Register

Figure 6: registering new user



```
user_010
    Email: "wardahasan@hotmail.com"
    Name: "Warda Hasan"
    Phone: "971 860 3213"
    Reward_Points: 0
    UserId: "client_010"
    User_Type: "Admin"
```

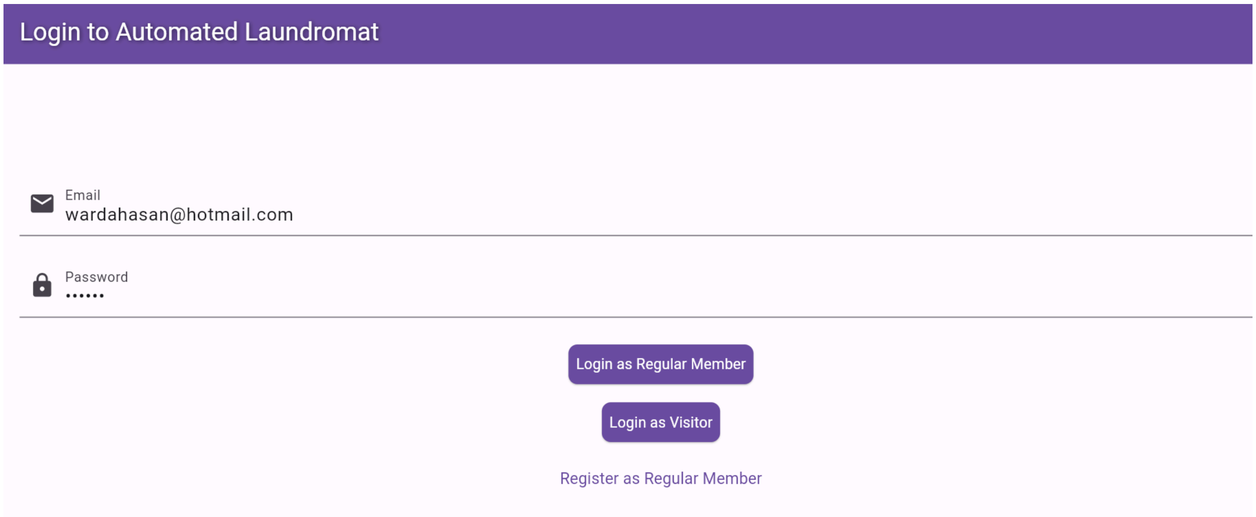
a.

```
user_009
    Email: "hafsahtahir@gmail.com"
    Name: "Hafsah Tahir"
    Phone: "971 532 3234"
    Reward_Points: 0
    UserId: "client_009"
    User_Type: "Regular Member"
```

b.

Figure 7: database recording newly registered users

The user was then prompted to log in using the credentials they had signed up with. The log in page has the option to log in as a registered user or a visitor. When logging in as a visitor, there is no need to enter credentials. Figure 8 shows the login page.



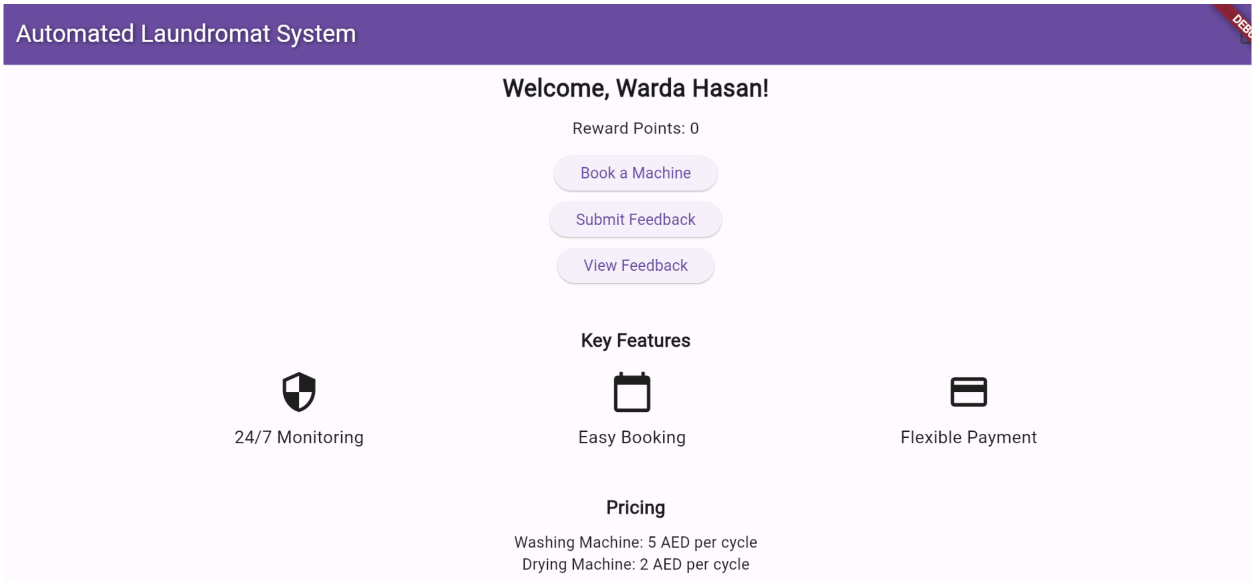
Login to Automated Laundromat

Email: wardahasan@hotmail.com

Password: ••••••

Figure 8: login page

After logging in, the user is greeted with the login page, as seen in figure 9 below.



Automated Laundromat System

Welcome, Warda Hasan!

Reward Points: 0

Key Features

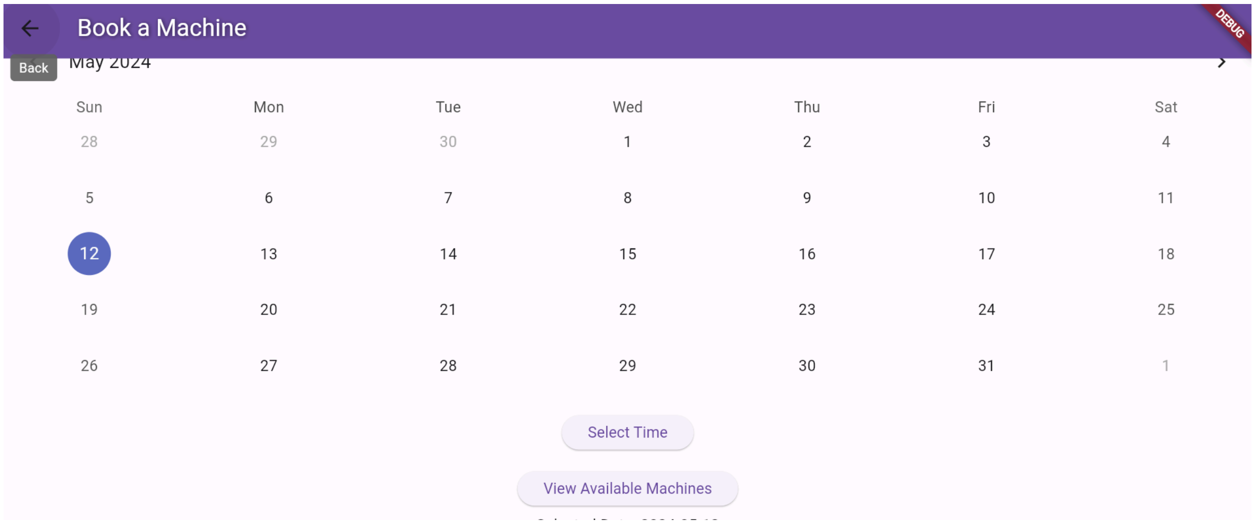
Pricing

Washing Machine: 5 AED per cycle
Drying Machine: 2 AED per cycle

Figure 9: homepage after logging in

2. Booking a Machine –Test Case ID: TC002

The user will navigate to the "Book Machine" button on the home page, after which they will be greeted with a calendar page in which they can select the date that they want to book the machine. This page is shown in figure 10.

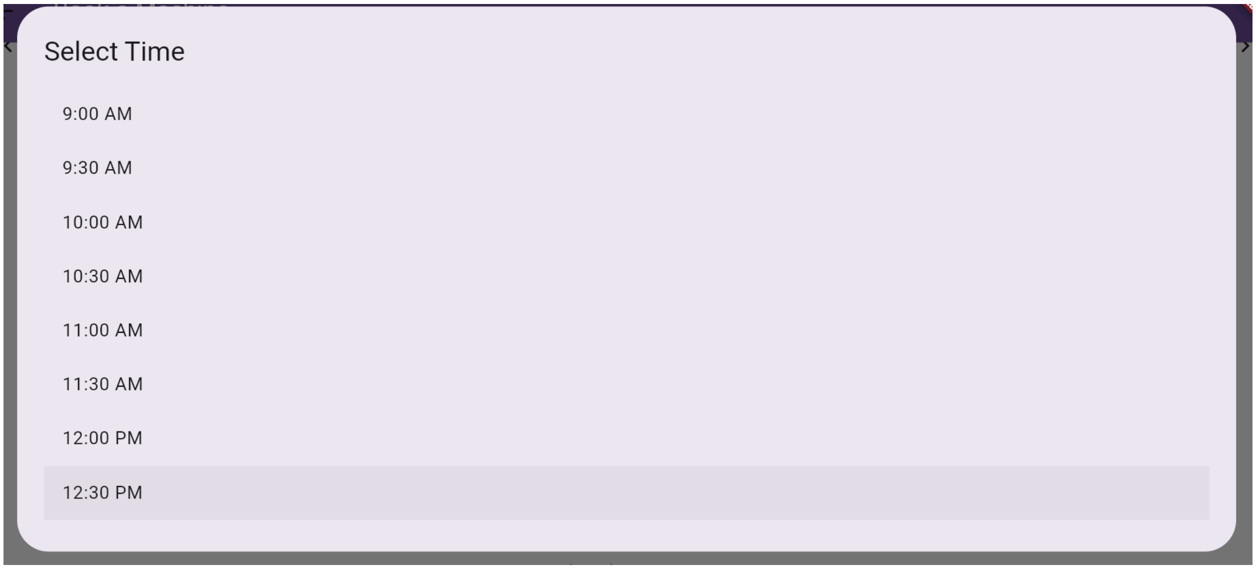


May 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

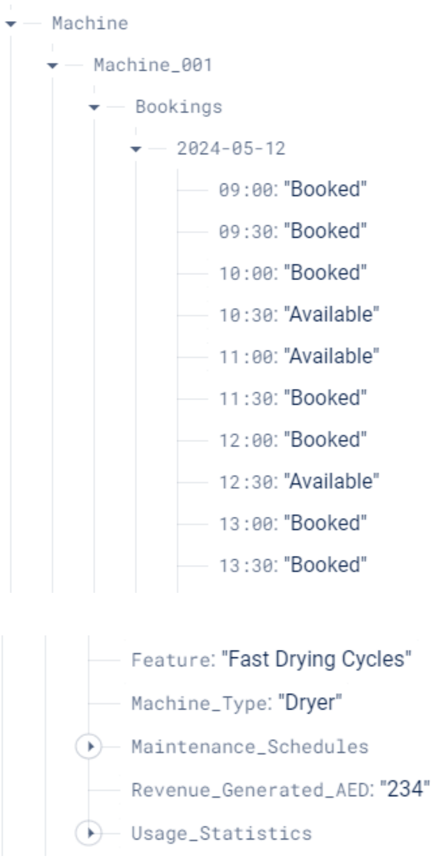
Figure 10: Book a machine calendar page

After this, the user will be prompted to select the time slot that they want to book, as well as the type of machine that they want to book. Once they have selected these things, the system will check to see if machines are available at that time, and if they are it will show the user what machines are available at that particular date and time.


Select Time

9:00 AM
9:30 AM
10:00 AM
10:30 AM
11:00 AM
11:30 AM
12:00 PM
12:30 PM

Figure 11: Selecting a time slot for selected time

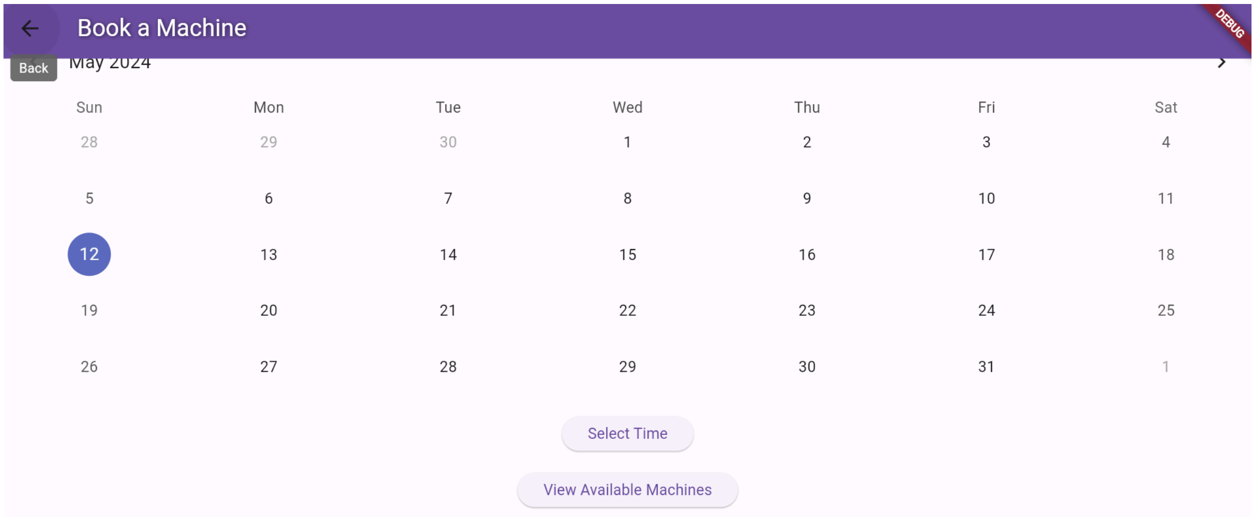


```
Machine
  Machine_001
    Bookings
      2024-05-12
        09:00: "Booked"
        09:30: "Booked"
        10:00: "Booked"
        10:30: "Available"
        11:00: "Available"
        11:30: "Booked"
        12:00: "Booked"
        12:30: "Available"
        13:00: "Booked"
        13:30: "Booked"

    Feature: "Fast Drying Cycles"
    Machine_Type: "Dryer"
    Maintenance_Schedules
    Revenue_Generated_AED: "234"
    Usage_Statistics
```

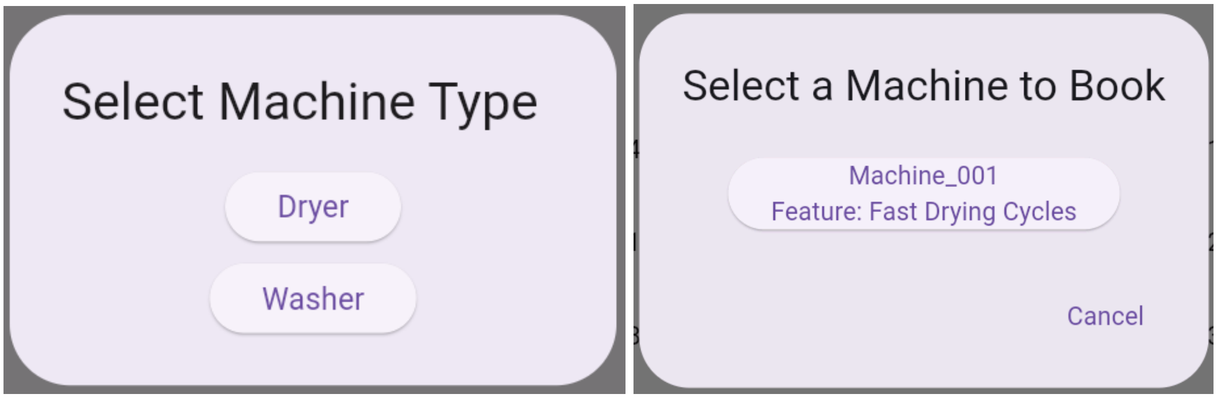
Figure 12: database showing machine record

From the database, we can see that the machine Machine_001 is booked on 12 May 2024 at 09:00, and it is available at 12:30. It is a Dryer machine.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

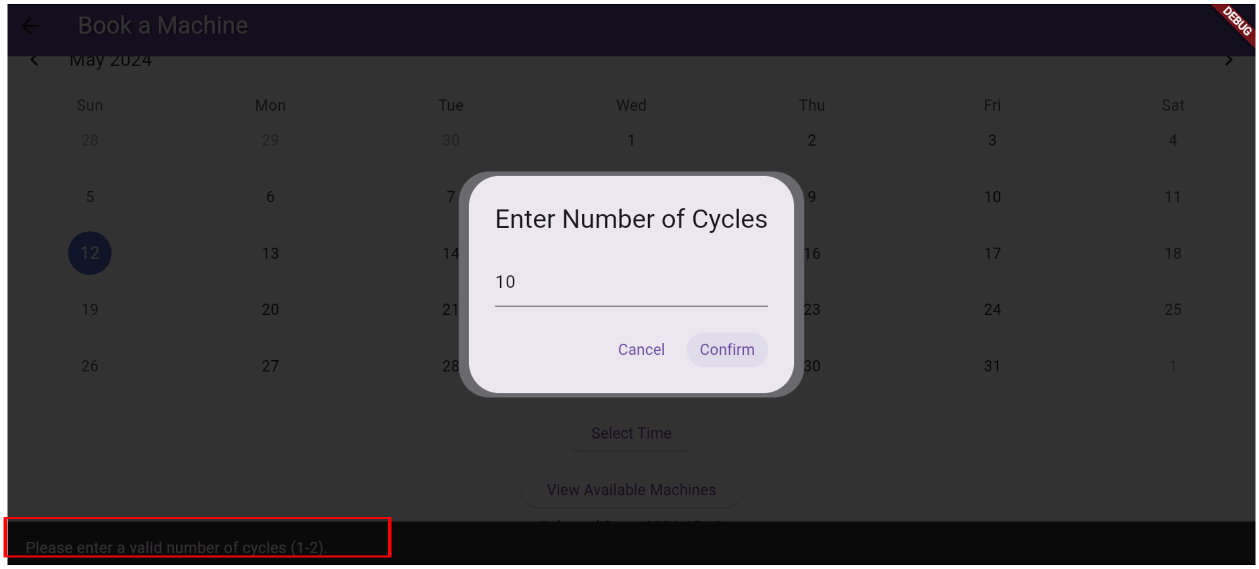
Figure 13: confirming the date and time



a. *b.*

Figure 14 a,b: selecting machine type and specific machine.

After selecting the machine, the user will be prompted to enter the number of cycles they want to book the machine for. If the user enters a number greater than 2 then they will see an error message and be prompted to enter a number between 1-2. After they select the number of cycles (eg. 1), then they are forwarded to the payment page. The booking will also be recorded in the database.

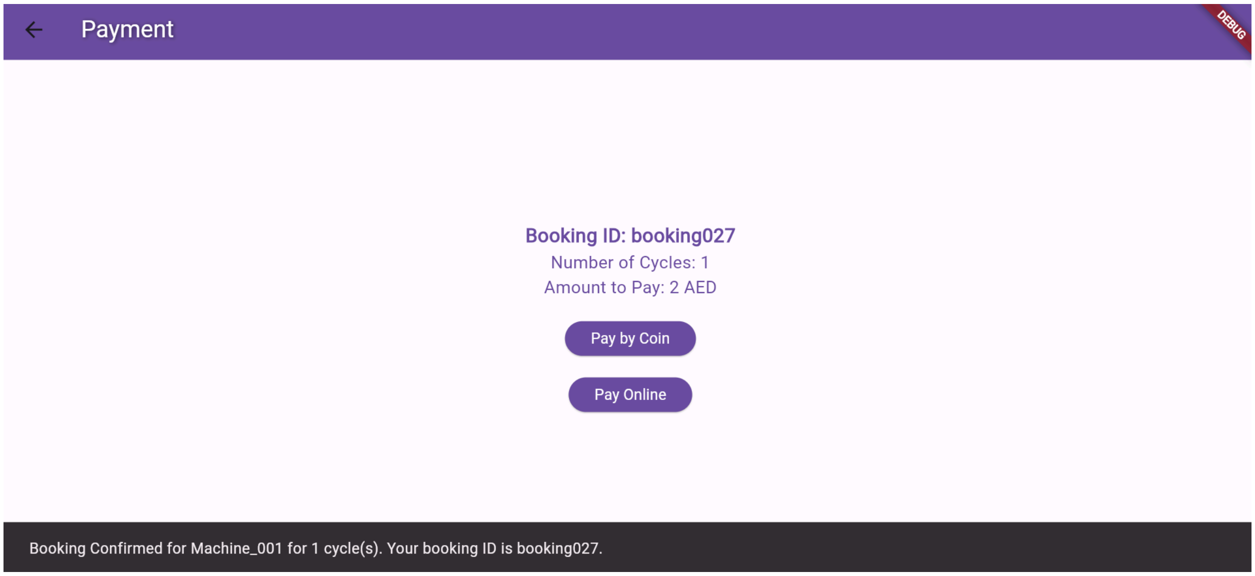


Enter Number of Cycles

10

Please enter a valid number of cycles (1-2).

Figure 15: Incorrect number of cycles entered and error message shown



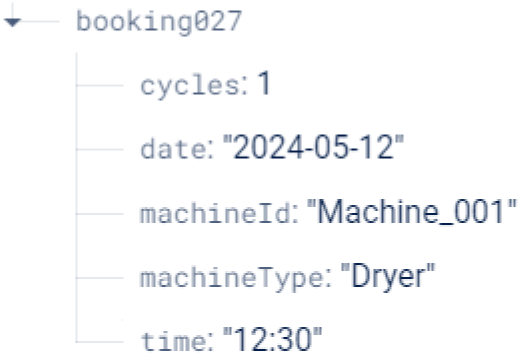
Booking ID: booking027

Number of Cycles: 1

Amount to Pay: 2 AED

Booking Confirmed for Machine_001 for 1 cycle(s). Your booking ID is booking027.

Figure 16: correct number of cycles entered and forwarded to payment page

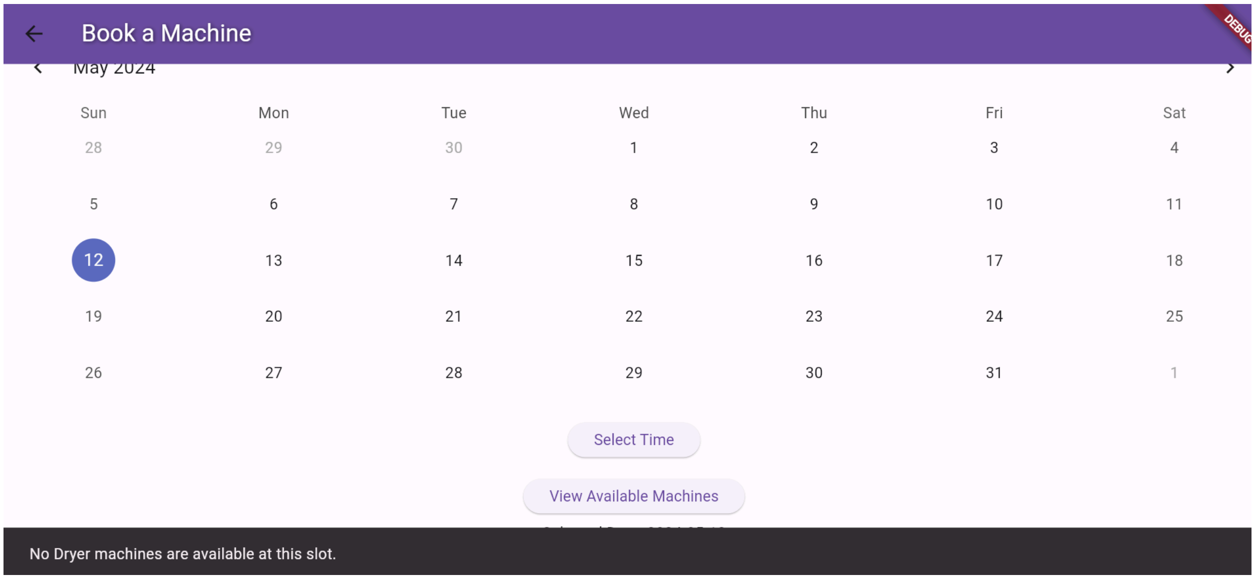


```
booking027
    cycles: 1
    date: "2024-05-12"
    machineId: "Machine_001"
    machineType: "Dryer"
    time: "12:30"
```

Figure 17: booking recorded in the database

Test Case ID: **TC003**

In this test case, we needed to check if the system would stop the user from double booking the same machine in the same time slot of the same day. Once the user selected their preferred date and time and the system checked the database and all of the machines were booked, the system would show an error message. As seen in figure 12, the machine is booked on 12th May at 9:00, after selecting the View Available Machine options it shows that there is no dryer machine available in that slot.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

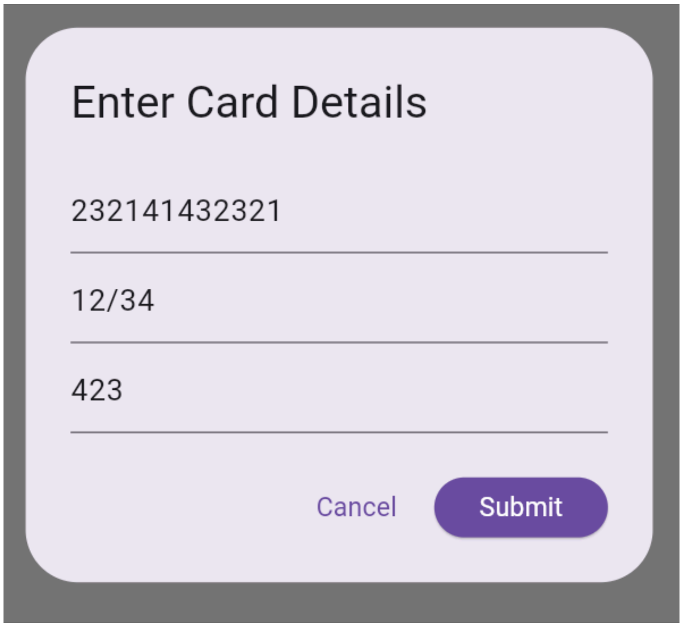
No Dryer machines are available at this slot.

Figure 18: no machine available in selected time slot so cannot book

3. Payment Confirmation

Test Case ID: TC004

From the Payment interface shown in figure 16 (which shows up after the booking has been made), the user is prompted to select their payment method. Choosing payment by card will forward them to the card payment interface where they can enter their details.



Enter Card Details

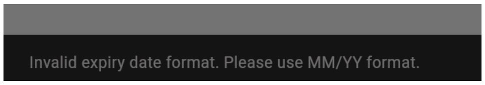
232141432321

12/34

423

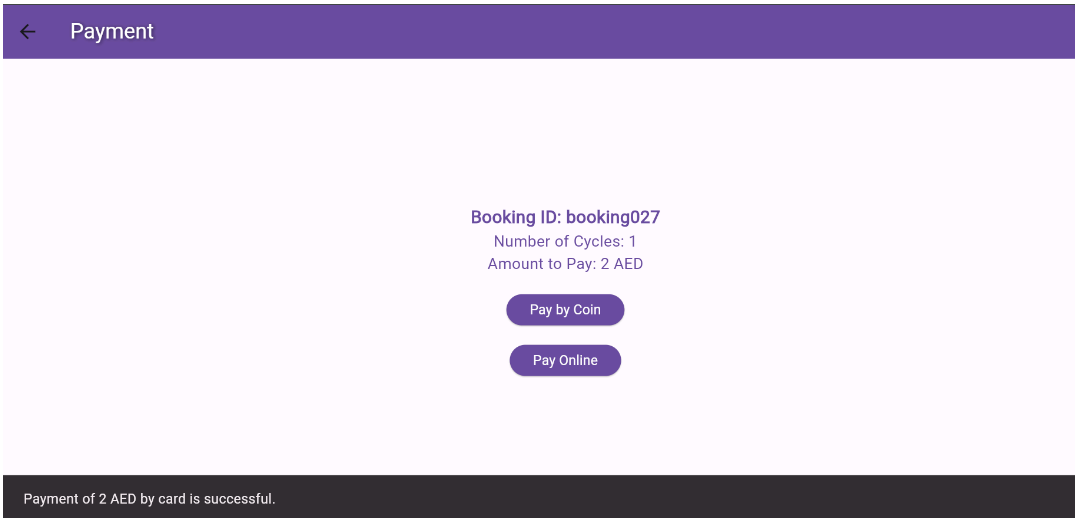
Figure 19: entering card details

If details are incorrect, then an error message shows up. Otherwise, the payment will be confirmed.



Invalid expiry date format. Please use MM/YY format.

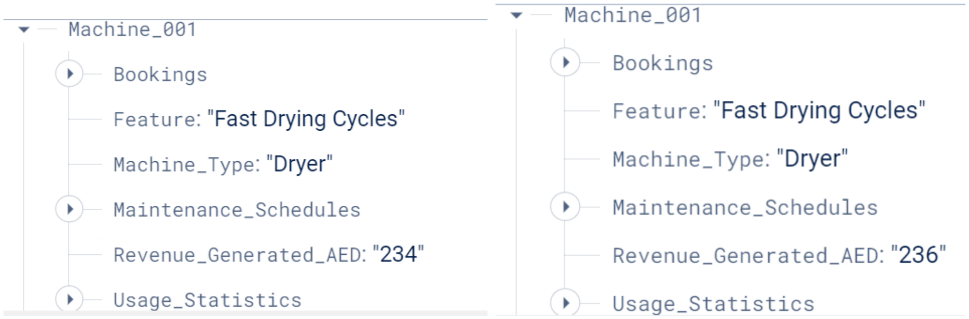
Figure 20: incorrect details entered



Booking ID: booking027
Number of Cycles: 1
Amount to Pay: 2 AED

Payment of 2 AED by card is successful.

Figure 21: payment confirmed

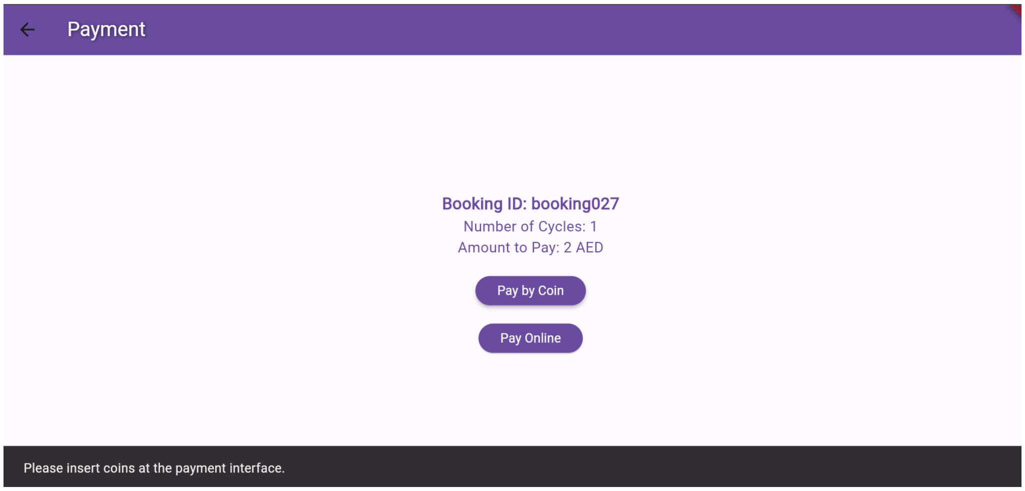


```
Machine_001
    Bookings
    Feature: "Fast Drying Cycles"
    Machine_Type: "Dryer"
    Maintenance_Schedules
    Revenue_Generated_AED: "234"
    Usage_Statistics
```

```
Machine_001
    Bookings
    Feature: "Fast Drying Cycles"
    Machine_Type: "Dryer"
    Maintenance_Schedules
    Revenue_Generated_AED: "236"
    Usage_Statistics
```

Figure 22: Revenue Generated by the machine is updated in the Database

If coin payment is selected, the user will be prompted to enter coins on the machine.



Booking ID: booking027
Number of Cycles: 1
Amount to Pay: 2 AED

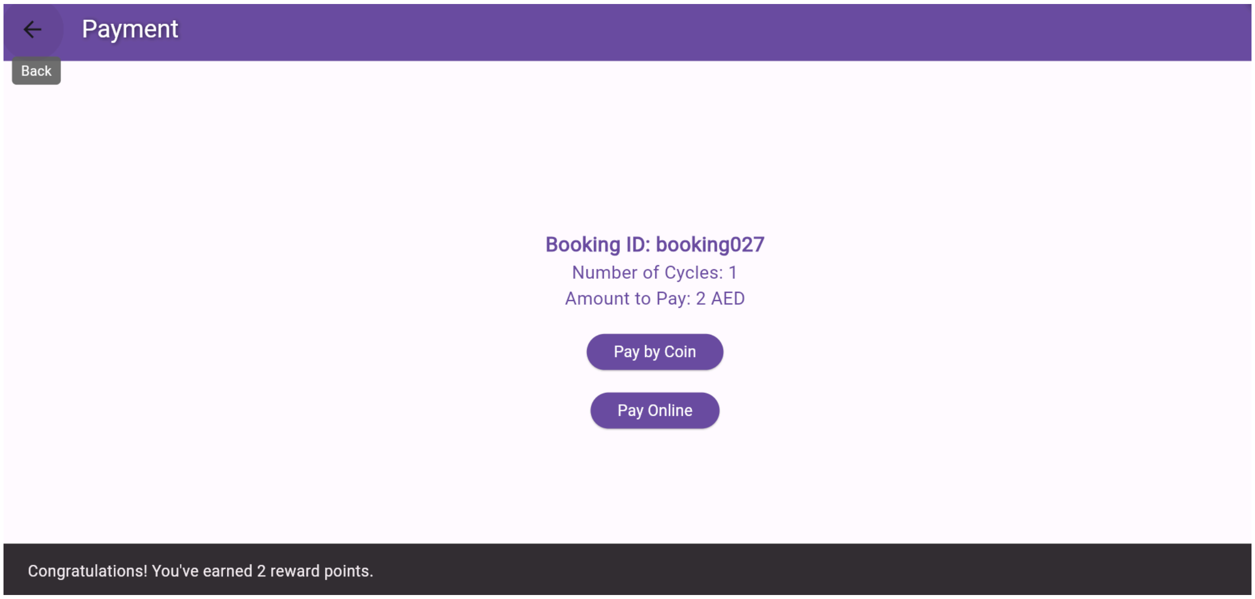
Please insert coins at the payment interface.

Figure 22: coin payment prompt

4. Reward Points Allocation

Test Case ID: TC005

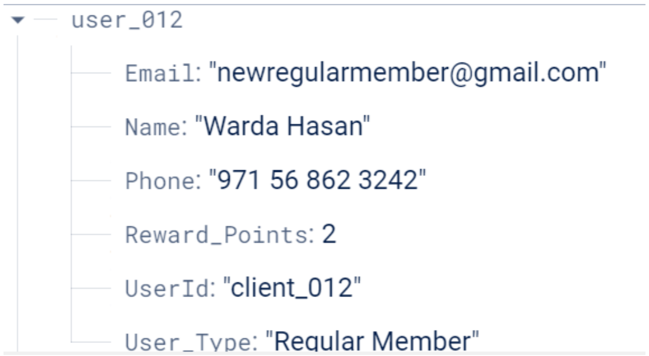
Once a payment has been made the points for the transaction are automatically added to the user's account in the database. This is also confirmed on the payment page and the home page.



Booking ID: booking027
Number of Cycles: 1
Amount to Pay: 2 AED

Congratulations! You've earned 2 reward points.

Figure 23: Points confirmed on payment page



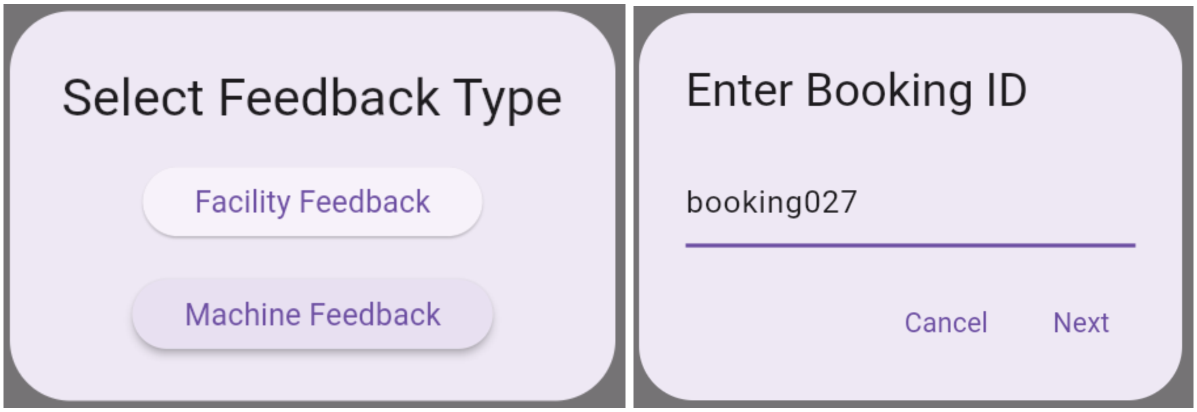
```
user_012
    Email: "newregularmember@gmail.com"
    Name: "Warda Hasan"
    Phone: "971 56 862 3242"
    Reward_Points: 2
    UserId: "client_012"
    User_Type: "Regular Member"
```

Figure 24: points added in the database

5. Leave Feedback

Test Case ID: TC006

Users will be able to leave feedback on their experience of the machines that they used. For this, they navigated to “Submit Feedback” and selected the “Machine Feedback” option. They will be asked to enter the booking ID they would like to leave reviews for so that the system can extract which machine was used in this transaction. This is seen in figure 26 a and 26 b respectively.



Select Feedback Type

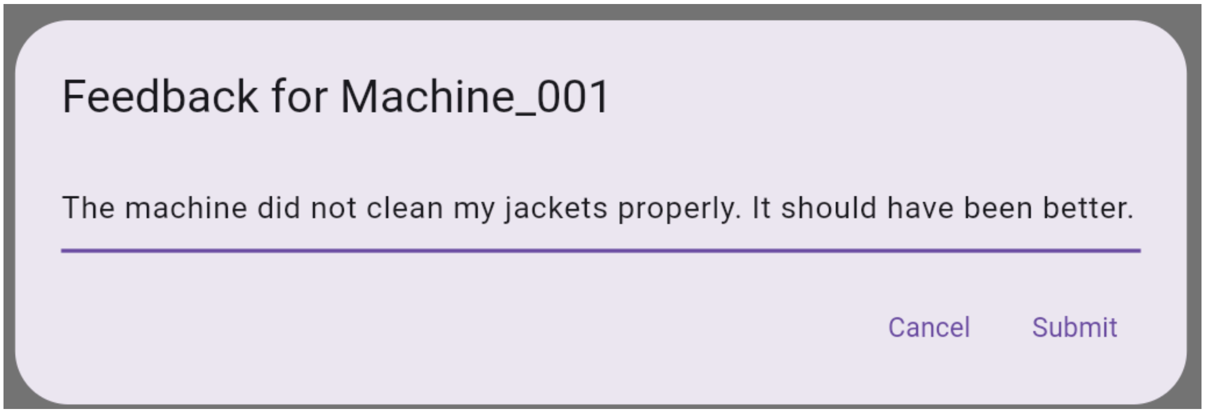
Enter Booking ID

booking027

a. *b.*

Figure 26 a, b: selecting machine feedback and entering booking ID

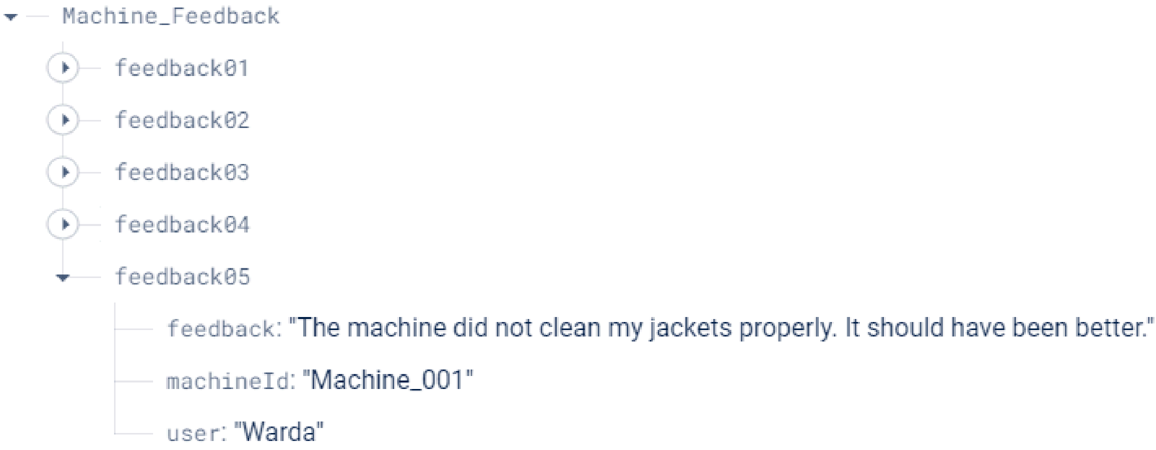
After this, the user will be asked to enter the feedback, which will then be saved in the database and displayed on the site in the "View Reviews" page.



Feedback for Machine_001

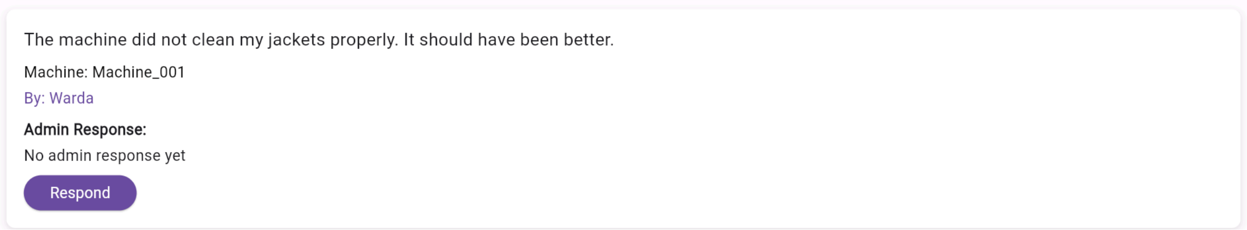
The machine did not clean my jackets properly. It should have been better.

Figure 27: Leaving feedback



```
Machine_Feedback
    feedback01
    feedback02
    feedback03
    feedback04
    feedback05
        feedback: "The machine did not clean my jackets properly. It should have been better."
        machineId: "Machine_001"
        user: "Warda"
```

Figure 28: Saving review in database



The machine did not clean my jackets properly. It should have been better.

Machine: Machine_001

By: Warda

Admin Response:

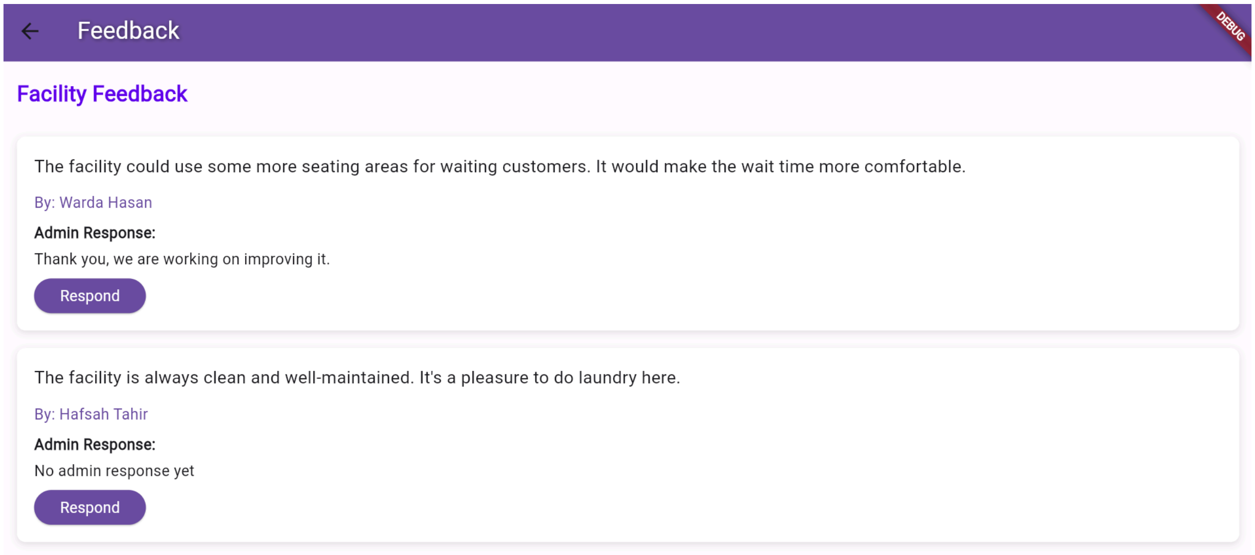
No admin response yet

Figure 29: Displaying review on the website

6. Only Admins have the option of responding to feedback

Test Case ID: TC007

The user was registered as an Admin as seen in figure 6.



Facility Feedback

The facility could use some more seating areas for waiting customers. It would make the wait time more comfortable.

By: Warda Hasan

Admin Response:

Thank you, we are working on improving it.

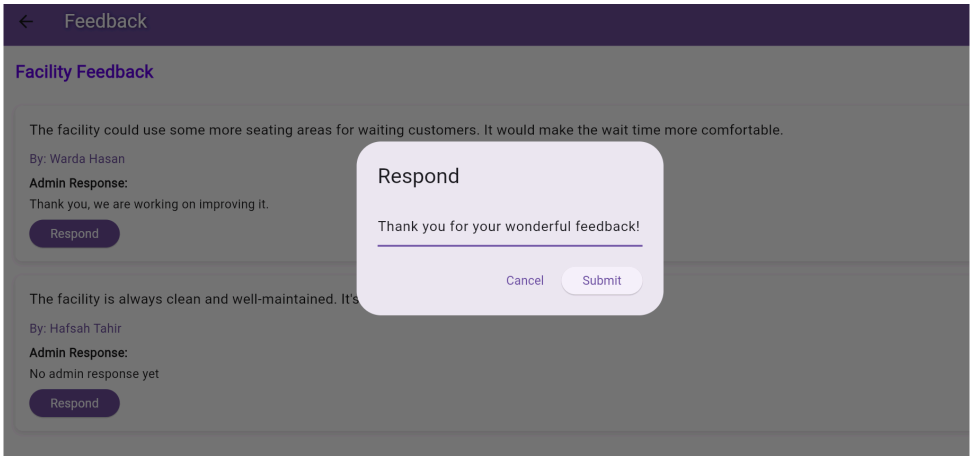
The facility is always clean and well-maintained. It's a pleasure to do laundry here.

By: Hafsah Tahir

Admin Response:

No admin response yet

Figure 30: Respond button for the users who are Admins



Facility Feedback

The facility could use some more seating areas for waiting customers. It would make the wait time more comfortable.

By: Warda Hasan

Admin Response:

Thank you, we are working on improving it.

The facility is always clean and well-maintained. It's

By: Hafsah Tahir

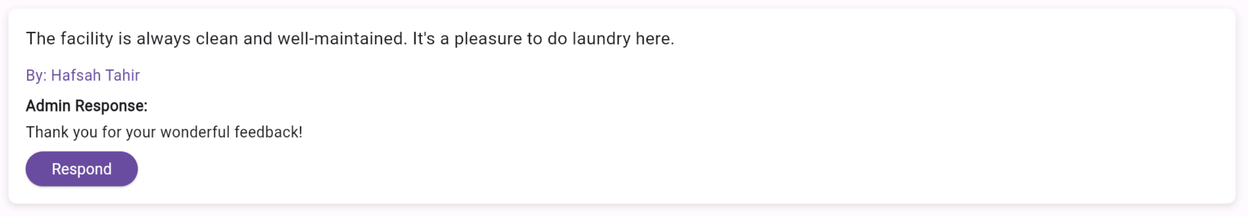
Admin Response:

No admin response yet

Respond

Thank you for your wonderful feedback!

Figure 31: Admin response to a feedback



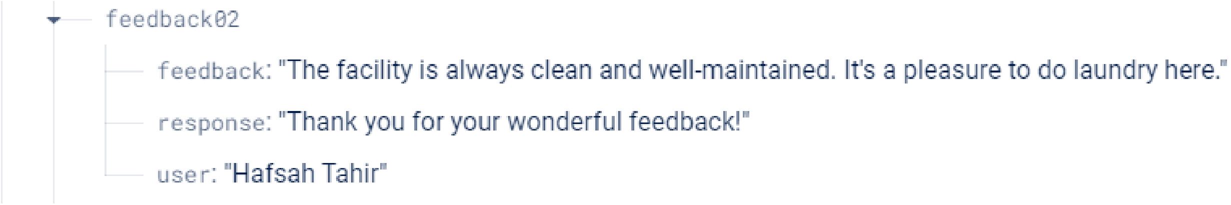
The facility is always clean and well-maintained. It's a pleasure to do laundry here.

By: Hafsah Tahir

Admin Response:

Thank you for your wonderful feedback!

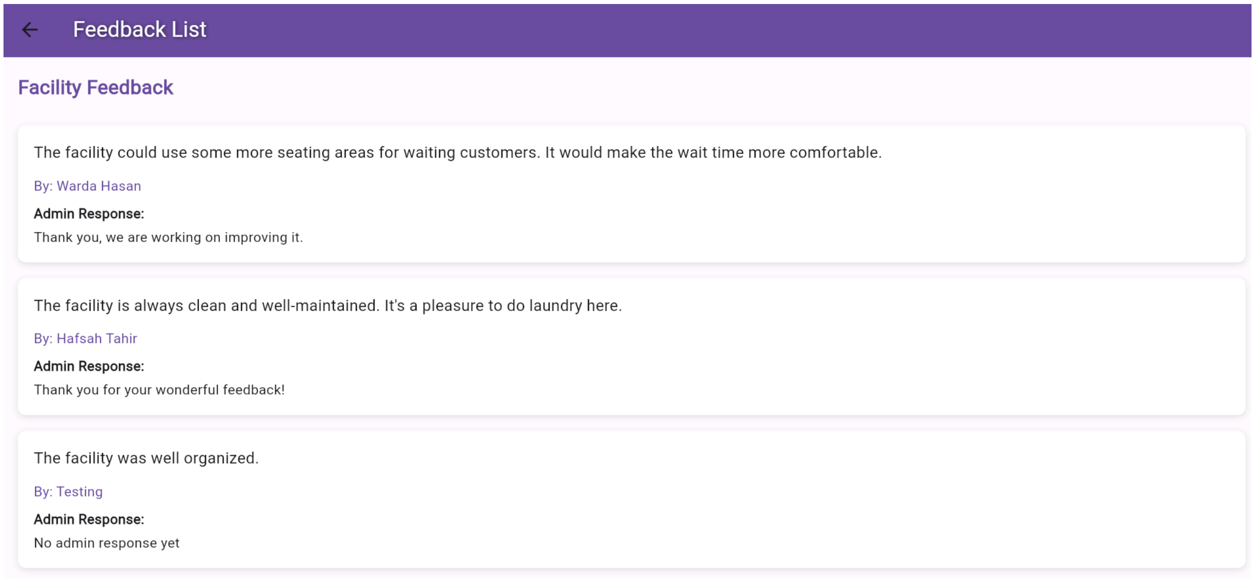
Figure 32: Admin response updated in the website



```
feedback02
    feedback: "The facility is always clean and well-maintained. It's a pleasure to do laundry here."
    response: "Thank you for your wonderful feedback!"
    user: "Hafsah Tahir"
```

Figure 33: Admin response updated in the database

In the screenshot below, the user was a visitor and they do not have the “Respond” to feedback option.



Facility Feedback

The facility could use some more seating areas for waiting customers. It would make the wait time more comfortable.

By: Warda Hasan

Admin Response:

Thank you, we are working on improving it.

The facility is always clean and well-maintained. It's a pleasure to do laundry here.

By: Hafsah Tahir

Admin Response:

Thank you for your wonderful feedback!

The facility was well organized.

By: Testing

Admin Response:

No admin response yet

Figure 33: Normal users (visitors or regular members) without Respond option