

Kereta Sewa Jimat: Car Rental System with Fuel Update

Muhammad Aliff Muhamad Shukri¹, Mohd Zanes Sahid^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: zanes@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2024.05.02.066>

Article Info

Received: 16 July 2024

Accepted: 30 October 2024

Available online: 15 December 2024

Keywords

Car Rental, Fuel, Application, Global Positioning System (GPS)

Abstract

Car rental services provide a solution for everyone to temporarily use a car for diverse purposes. However, it includes problems that might occur like human error during booking process, incomplete rental car information, and oversight in refuel when using the rental car. Objectives of this system are to design the car rental system using an object-oriented approach, develop it as a mobile application, and test it using User Acceptance Testing. Furthermore, the scope includes several modules, like the booking module, the rental car catalog module, and fuel update module. This system adheres to the prototyping model methodology, and some procedures have been followed to develop the system, which are communication, quick planning, modelling, prototyping, and deployment with feedback. The testing and feedback collection have been completed to understand user experiences with the system. This can streamline the car rental booking process for customers and improves efficiency in management for staff.

1. Introduction

Car rental service is a convenient and flexible transportation solution allowing individuals to lease vehicles for various purposes temporarily. Car rental services are commonly used for personal travel, business trips, vacations, and temporary transportation. Kereta Sewa Jimat is a car rental service business under Infinity Global Network in Parit Raja, Johor. This rental car service is also available for areas close to Parit Raja, such as Ayer Hitam, Batu Pahat and Universiti Tun Hussein Onn Malaysia (UTHM). They are using spreadsheet software like Microsoft Excel and a messenger application like WhatsApp to manage their car rental service process.

A booking system for car rentals can streamline various operations, including booking and managing reservations. This results in enhanced operational efficiency for the business, ultimately saving time and simplifying customer and organization processes [1]. The current car rental process is complicated and packed with difficulties. Customers frequently encounter a booking problem like double booking which is caused by human error, and they may receive the least information about the rental car, and rental durations could be more flexible. Customers sometimes do not have responsibilities about the fuel when they use the rental car, as they always forget or do not want to refill the fuel, which is one of the policies in car rental service where the customer needs to refill the fuel of the rental car when they want to return it. This application will offer a user-friendly platform and a mobile app with fuel updates, so this application will be developed to overcome the problems which can make the whole experience efficient and cost-effective for customers and staff.

2. Literature Review

2.1 Car Rental Service

Car rental services are businesses that allow people to use cars for cash temporarily. Through this service, people and companies can hire cars for short periods ranging from an hour to a few days. Due to this, car rental services are top-rated because they allow people to use vehicles without having committed to their ownership indefinitely. Contemporary travelers rely on car rental services that offer flexibility, accessibility, affordability, comfort, security, and varied car choices. They allow travelers to make discoveries in comfort without the aid of public transportation and cars. Many types of rental car companies offer are suitable for business trips or vacations, depending on their needs and budgets [2].

2.2 Mobile Application

A mobile application is a type of software that is intended to operate on portable electronics like tablets and smartphones. It is the outcome of current advancements in technology. The combination of media, information technology, the Internet, and cutting-edge technologies has given rise to mobile apps [3]. Many people prefer to perform their tasks or go online through mobile devices. It is because using mobile applications on mobile devices is more accessible because all tasks can be done at their fingertips only and can be done anywhere. Users will become more dependable on the mobile application to do some tasks or solve problems such as making a reservation for renting a car.

2.3 Flutter Framework

Google developed the open-source Flutter framework utilizing the Dart programming language to create cross-platform apps for various desktop operating systems, Android, iOS, macOS, and Google Fuchsia, all from a single code script. It provides a native experience like other platforms or apps. It enhances the user experience, guarantees security, facilitates quicker updates, makes application maintenance more accessible, and offers many other benefits that come with every smoothly running program [4]. The Flutter framework will be used to develop this Car Rental System application because Flutter has many benefits, such as allowing the widgets to be easily modified or customized and being rich with libraries. It also provides the User Interface (UI) widgets for design requirements.

2.4 Global Positioning System (GPS)

Devices are used in satellite-based GPS tracking to track the position. GPS position tracking uses satellites, receivers, transmitters, and other pertinent equipment. When using satellite-based GPS tracking, one must be concerned about GPS signal deterioration that might result in the tracking target being lost. In this application, phone-based GPS tracking in the customer's phone provides real-time data about the car's location and movement. It allows staff to monitor the car's usage throughout the rental period, tracking its current position and route. This real-time monitoring helps ensure that the car is being used appropriately and assists with administrative tasks, such as verifying the car's location after it is returned. Additionally, it can provide safety and security features, such as alerting the car rental service if the vehicle is moved without authorization.

2.5 Study of Proposed System

Fig. 1 below shows the flowchart of the Kereta Sewa Jimat: Car Rental System with Fuel Update. The flowchart of the system only includes the customer's perspective when using the system. For a new customer they need to register first by filling in the registration form. Once done, their details need to be verified by the staff whether the customer is eligible to rent a car or not. As for existing customer and a customer who just got verified by the staff, they can login the application. Then, the customers can start the booking process by choosing the date. If there is no available time on that date, they need to choose another date. If there is available time, they could choose their desired time. If there is no available rental car, they need to choose another time. If there is an available rental car, they could choose the rental car, then view the rental car details. Once satisfied with the choice, customers can choose the rental period and confirm. The system will display the booking summary and customer need to upload the payment receipt by referring to the company bank account number. During the rental day, customers should click "START" in the application to start their rental period. Before customers end their rental period, they need to update the status of the fuel of the rental car. After that they finally can click "END" in the application to end the rental period.

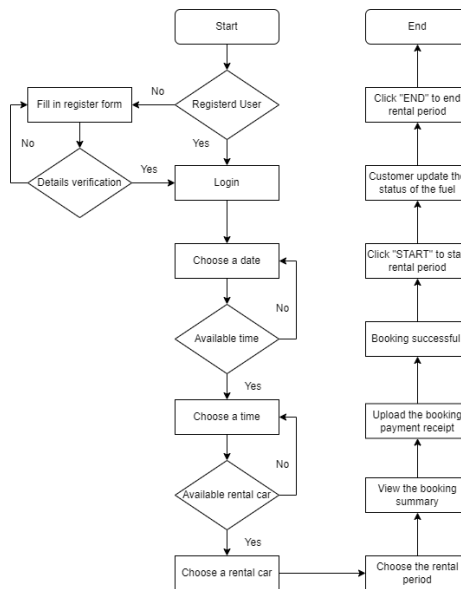


Fig. 1 Flowchart of the Kereta Sewa Jimat: Car Rental System with Fuel Update

2.6 Study of Existing Related Systems

A study of a similar system has been done on existing systems. This study helps develop a plan that meets the needs of users. In addition, study and assessments conducted will serve as a guideline to correct deficiencies and make improvements and changes to the system to be developed. The system also has studied the similarities that can be associated with the system to be developed. The existing system are Hertz Malaysia, SOCAR Malaysia, and GoCar Malaysia. Table 1 below shows the comparison between the existing system and proposed system.

Table 1 Comparison between existing system and proposed system

Features	Hertz Malaysia	SOCAR Malaysia[5]	GoCar Malaysia	Proposed system
Car Reservation	Make a booking to rent a car	Make a booking to rent a car	Make a booking to rent a car	Make a booking to rent a car
Car Catalog	View the available car near customer	View the available car near customer	View the available car near customer	View the available car based on date and time
Customer Profile	Customer can view and edit profile	Customer can view and edit profile	Customer can view and edit profile	Customer can view and edit profile
Track rental car during rental period	Cannot track the rental car	Can track the rental car	Cannot track the rental car	Can track the rental car
Fuel Update/Report	No update or report needed for fuel of the rental car	Can claim the money for refueling the rental car	Need to update the fuel status before and after	Need to confirm the fuel status and fill in the form

3. Methodology/Framework

3.1 Prototyping Model

The Prototype Model has been chosen to develop the Car Rental System with Fuel Update. A prototype model is a common approach for producing a working replica of any concept, piece of software, or system that needs more development [6]. Since the Prototype Model will offer a smaller version of the projected final product, precise customer feedback must be incorporated into the development procedure. The prototype Model could help determine whether it can or cannot satisfy the customer's needs. The user experiences the system through prototyping and offers suggestions for any needed changes. If the requirement is not met or fulfilled, the prototype is refined and may undergo multiple iterations before being integrated into the implementation phase.

3.2 Evolutionary Prototyping

Evolutionary Prototyping has been chosen. This model involves a series of prototyping refinements. Designing and splitting the system into several separate modules is the initial task. After each module has the features it needs, customers view it. Following the collection of feedback from customers, the model is once again refined. This procedure continues constantly until a final, workable prototype is produced. It will eventually get to the point where both you and the customers agree that the product is useful and beneficial [7]. The implementation of Evolutionary Prototype in developing the car rental system will ensure that the system aligns closely with the changing needs of both customers and the rental service, resulting in a more user-friendly and responsive solution.

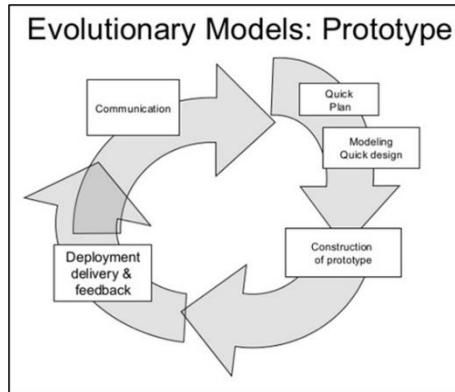


Fig. 2 Evolutionary Prototyping [8]

There are total of five phases from the evolutionary prototyping model. As shown in Table 2 below, each phase has its own assignment and output that are needed to produce during the entire project development.

Table 2 Software development activities and their task

Phase	Activity	Deliverable
Communication	<ul style="list-style-type: none"> - Gather the data - Have an interview session with the owner of the car rental service. 	<ul style="list-style-type: none"> - Proposal - Questionnaire a form of the interview
Quick Plan	<ul style="list-style-type: none"> - Study the data that has already been gathered and translate into meaningful information. - Outlined the scopes, significance, objective, and identify. - Work out requirements for the system 	<ul style="list-style-type: none"> - Use Case Diagram - Use Case Specification - Class Diagram - Activity Diagram - Sequence Diagram - Functional and Non-Functional Requirement of the system
Modelling Quick Design	<ul style="list-style-type: none"> - Design the User Interface (UI) by referring to the wireframe. - Design the database of the system, 	<ul style="list-style-type: none"> - Wireframe of the system - User Interface - Database Design
Construction of Prototype	<ul style="list-style-type: none"> - Write the program code of the system based on the requirements. 	<ul style="list-style-type: none"> - Program Code
Deployment Delivery and Feedback	<ul style="list-style-type: none"> - Test the prototype which modules have been integrated. - Find the bugs and figure out the part where does not meet the requirements. - The process will be redefined back to prototype phase based on user’s feedback, to meet the user requirements. 	<ul style="list-style-type: none"> - Test Cases

The first sub-phase of this analysis is determining user requirements. The developers must work with the users to determine how the current system functions and what the users want from the new system. It is also crucial that future users are involved in the process early on so that their views of the system are considered. Table 3 below shows the user requirements of the proposed application.

3.3 User Requirements Analysis

Table 3 *User Requirements of the proposed application*

No	User Requirement
1	Customer should be able to register by fill in the registration form
2	Customer should be able to view the status of the account verification
3	Customer should be able to login with correct phone number and password
4	Customer should be able to edit their profile details
5	Customer should be able to insert the date and time for booking the rental car
6	Customer should be able to view the available rental car based on the date and time.
7	Customer should be able to select the available rental car
8	Customer should be able to view the summary of the booking
9	Customer should be able to upload receipt as evidence of booking payment
10	Customer should be able to view the current and past booking
11	Customer should be able to upload the receipt as evidence of extra fee payment
12	Customer should be able to set the start and end the period of renting the car
13	Customer should be able to view the confirmation message about fuel status
14	Customer should be able to confirm the status of the fuel
15	Customer should be able to fill in the fuel form
16	Staff should be able to view the customer's registration form
17	Staff should be able to verify the customer's registration form
18	Staff should be able to add the rental car and its details
19	Staff should be able to edit the rental car and its details
20	Staff should be able to delete the rental car and its details
21	Staff should be able to add the period and price for the rental car
22	Staff should be able to view the current and past customer's booking
23	Staff should be able to track the car of the customer in real-time during their rental session
24	Staff should be able to select the type of report and frequency
25	Staff should be able to view the generated report

3.4 Functional and Non-functional Requirements

Functional requirements are product features or functions that developers must incorporate in order to allow users to complete their tasks. Non-functional requirements specify how the system should operate rather than anything to do with its functionality. They frequently have an impact on the whole user experience and are essential for guaranteeing the system's usability, dependability, and effectiveness.

Table 4 *Functional requirements of the proposed system*

Module	Description
Authentication	- The application should allow customer to register and log in once verified
Profile	- The application should allow customer to view and edit profile details
Availability	- The application should allow customer to search the available rental car based on date and time
Rental Catalog	- The application should allow customer to select and choose the period of the selected rental car
Car	
Payment	- The application should allow customer to view the booking summary and upload the receipt of the booking payment
Booking	- The application should allow customer to view all the booking with status
Rental Session	- The application should allow customer to start and end the rental session - The application should allow staff to track customer's location during rental period
Fuel Update	- The application should allow customer to confirm the status of the fuel and update it by fill in the fuel form
Report	- The application should allow staff to select the type of report and frequency - The application should allow staff to view the generated report

Table 5 *Non-functional requirements of the proposed system*

Requirement	Description
Capacity	- The database of the application should be able to store at least 200 000 records without a significant impact on performance.
Performance	- The system should respond to customer login time with a maximum latency of 2 seconds.
Security	- Every customer needs to have an account and can access the application when the phone number and password are correct or matching. - Enables registered customers to log in securely using their credentials. - Implements security measures such as password encryption to protect customer data. - Ensures that only authorized users can access the application's features and services.
Operational	- The application is available for Android operating system mobile devices with 6.0 version and above

3.5 Use Case Diagram

Use case diagrams are used to demonstrate how users interact with applications or systems to accomplish tasks. It displays the way in which the user interacts with the application and its features. The suggested application's use case diagram is depicted in **Fig. 3** to demonstrate how the user and the application interact.

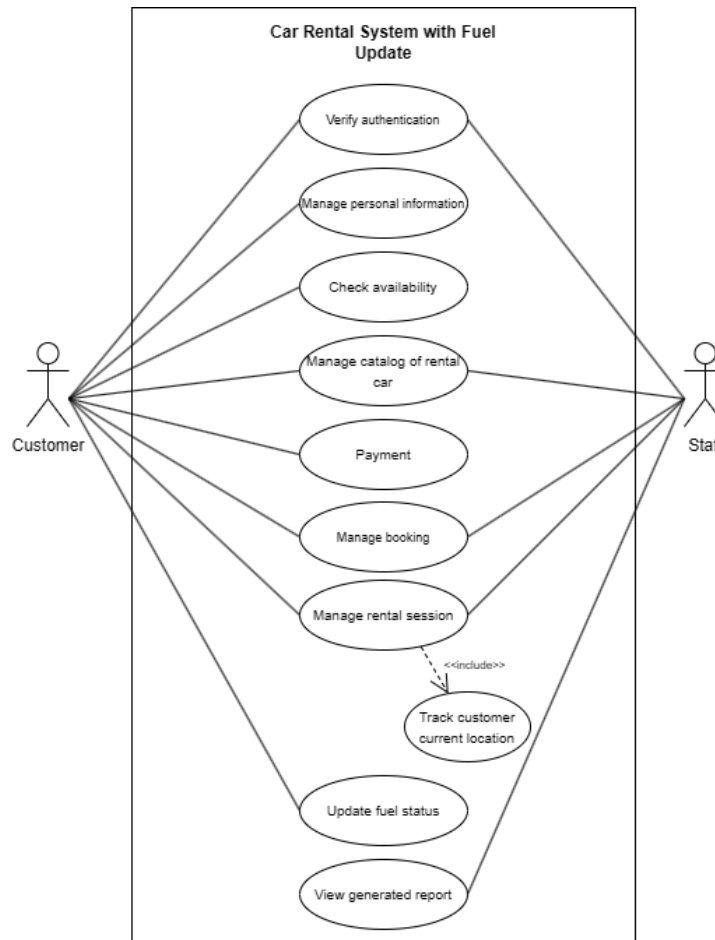


Fig. 3 Use Case Diagram of Kereta Sewa Jimat: Car Rental System with Fuel Update

There are two actors involved in the use case diagram which are customer and staff. As for the customer, there are eight (8) use case involved. First is “Verify Authentication” where customer need to verify their account by register and then can log in into the system. Next is “Manage Personal Information” where customer can view and edit their personal information. As for “Check Availability” use case. Customer can check the available rental car for booking by inserting the desired date and time. After that is “Manage Catalog of Rental Car” where customers can view details of selected rental car and choose how long they want to rent the car. Then, “Payment” is where customers view their booking summary and make a payment for the booking by uploading the receipt of the payment. In the “Manage Booking”, customer can view all the booking that they made with current status. “Manage Rental Session” is where customer notify the staff about the start and end of the rental session. Lastly, “Update Fuel Status” is for customer to update the fuel status of the rental car before the end of the rental session.

On the other hand, there are five (5) use case involved for admin. First is “Verify Authentication” where staff need to review and verify the customer’s registration. Next is “Manage Catalog of Rental Car” where staff can add, edit, and delete the rental car with its period and price. As for “Manage Booking”, staff can view all the customer’s bookings with the status. “Manage Rental Booking” is where staff can track customer during their rental session. The last use case for staff is “View Generated Report” where staff can choose the type of report with frequency and view the generated report.

3.6 Sequence Diagram

A sequence diagram depicts how objects interact with one another and describes how they function sequentially. The illustrated sequence diagram is useful for understanding the requirements well for the proposed system or application. The sequence diagram of all use cases will be shown in **Appendix A**.

3.7 Database Design – Class Diagram

Database design is the process of structuring and organizing data to efficiently store, manage, and retrieve information in a systematic way. In a database system, it entails designing tables, relationships, and constraints to guarantee data integrity and accuracy.

Based on class diagram in **Fig. 4**, there are seven entities involved which are Customer, Profile, RentalCar, PeriodPrice, Booking, CustLocation, and Fuel. Customer entity have a one-to-one relationship with Profile, and RentalCar, entities, one to one or more relationships with Booking entity. Next, RentalCar entity has a one-to-one relationship with Booking entities, while have a one to one or more relationships with PeriodPrice entity. Lastly, booking entity have a one-to-one relationship with CustLocation, and Fuel entities.

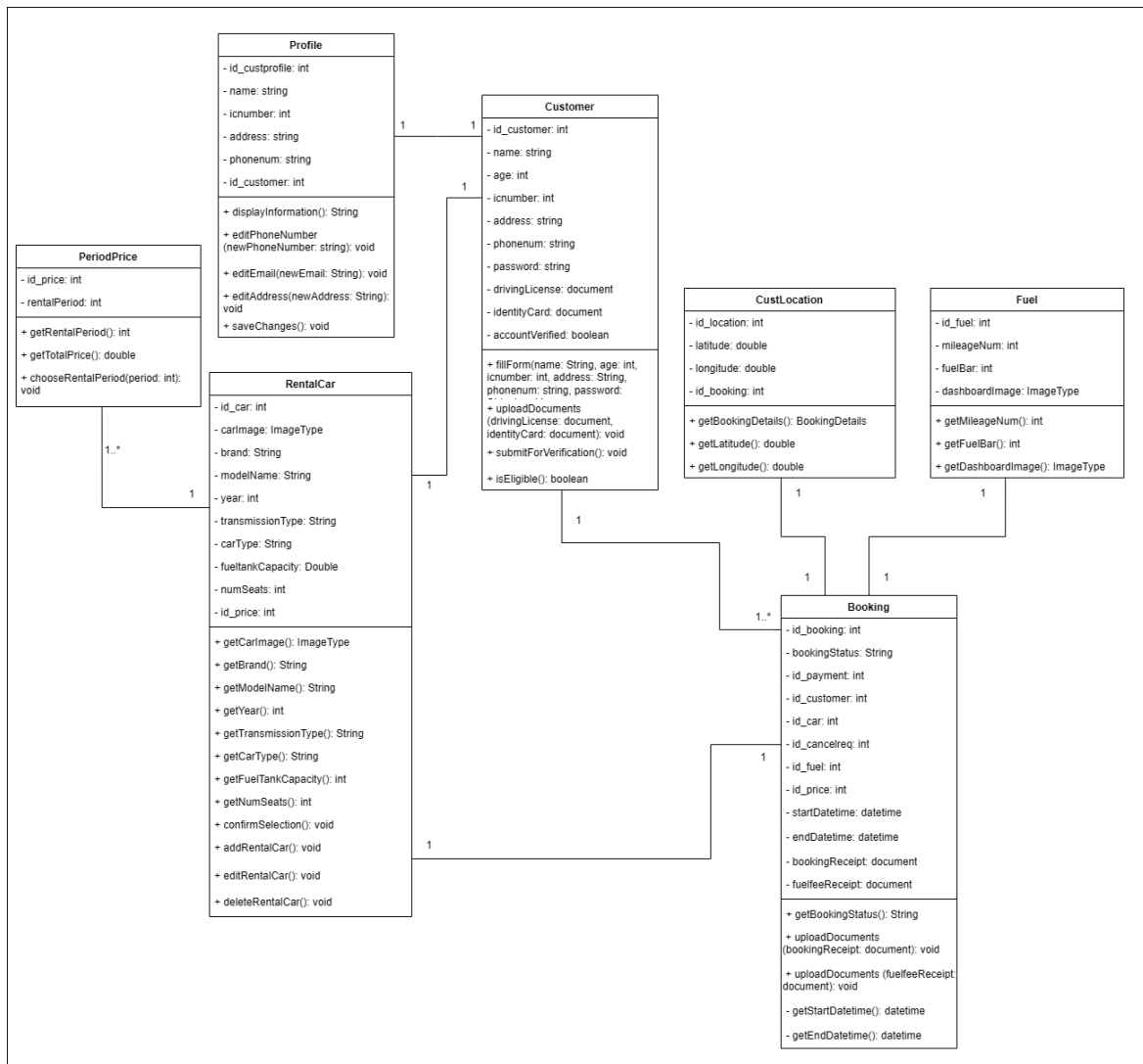


Fig. 4 Class Diagram for Kereta Sewa Jimat: Car Rental System with Fuel Update

3.8 User Interface Design

User interface design refers to the collection of design elements that maximize smooth interaction and create enjoyable user experiences when using the system. Good user interface (UI) will lead the users to using it well without any problems. **Fig. 5** below depicts some of the design of the UI of this system that includes for customer and staff.

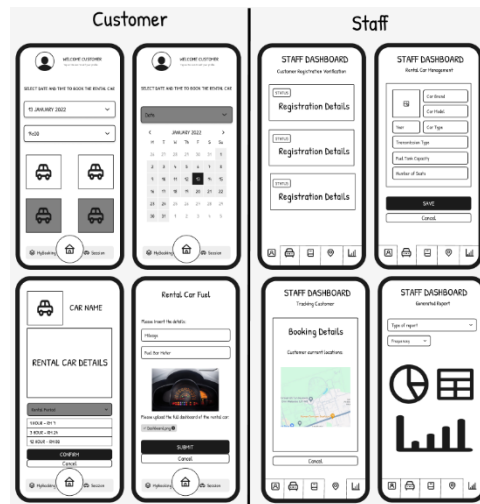


Fig. 5 User Interface Design for Kereta Sewa Jimat: Car Rental System with Fuel Update

3.9 Implementation of the System

The implementation of the Car Rental System with Fuel Update is being highlighted. The system is developed by using Visual Studio Code as the coding environment. As for the database, a No-SQL database which is Firebase has been used to integrate with the system to store, retrieve, update, and delete the information or data. The framework involved is Flutter that uses Dart language as the programming language to develop this system as an Android application.

Fig. 6 depicts the interfaces that are involved in the customer authentication process. When customer go through the application, they will first see the login page that includes email and password field, the role, login button, track account status button, and the register link. Customer needs to register by filling in the registration form and submit it. Then, customer can track their account verification status by filling in the email and check the status.

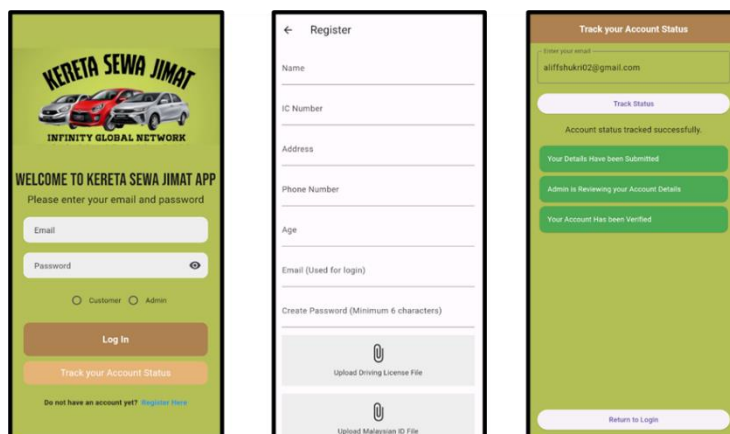


Fig. 6 Customer Authentication Interface

Fig. 7 depicts the page where staff can verify or reject the customer account. Staff can see the details of customer and review it. Then, the staff can decide whether the staff wants to verify or reject the customer account.

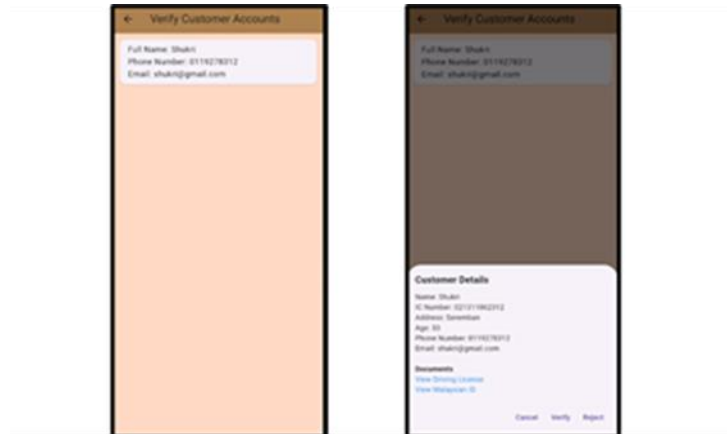


Fig. 7 Verification of Customer Account Interface

Fig. 8 shows the home page where customer can select their desired date and time to book the rental car. Then the system will show only the available rental for customer to choose. Then, customers can select the rental car to view more details of the rental car. Customer also can check the price based on their desired rental period either in hours or days. Then, they can click 'Checkout' button to navigate them to Checkout Page.

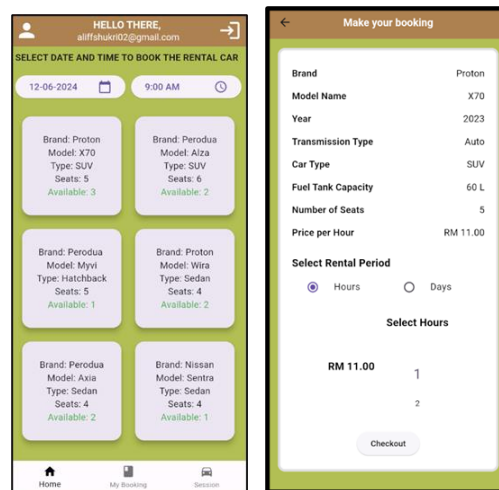


Fig. 8 Check Availability and Rental Car Details Interface

Fig. 9 shows rental car management interfaces where staff can add, edit, and delete the rental car. Staff can add the car with the plus '+' icon on the bottom right of the page, where it requires an staff to fill in the details about the rental car that will be added. Then the rental car will be added to the system and display on the list. Staff can edit the existing rental car details by tapping on pencil icon. The edit page will show details and it is editable. Trash icon is for staff remove or delete the car from the system.

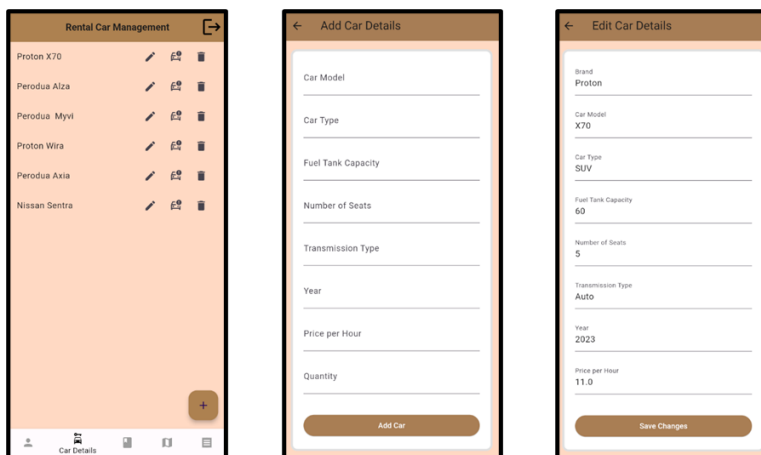


Fig. 9 Rental Car Management Interface

Fig. 10 depicts the interfaces where staff can manage the plate number by tapping on the car icon. The staff can view the list of plate numbers that has been assigned by each number of rental cars. Staff can add, edit, and delete the plate numbers. The add and delete function also affects the quantity of the rental car for that brand.

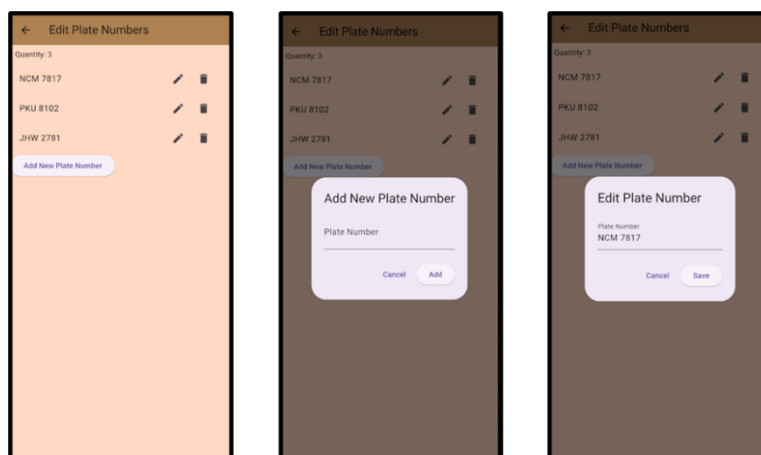


Fig. 10 Plate Number Management Interface

Fig. 11 shows the MyBooking Interfaces for customers to view their bookings with the status of that booking. Customers can view their bookings details by selecting any booking from the bookings list. Customers also need to pay for the penalty for the status Completed (Pending Penalty Payment) as they do not refill the fuel after end the rental session. As for staff, they also have the same interface to view all the bookings of their customers with status.

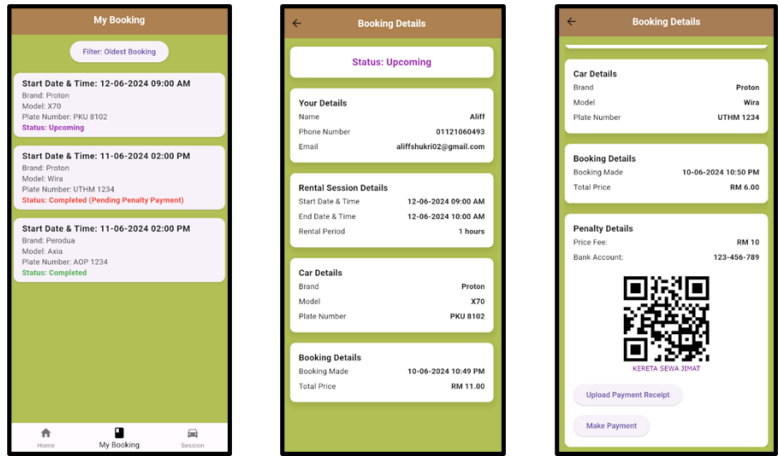


Fig. 11 Manage Bookings Interface

Fig. 12 depicts Rental Session page where customers can initiate and end their rental session. On this page, there is a countdown to show how many times are left before their rental session starts. The booking card below is the nearest upcoming booking that the customer will start the rental session.

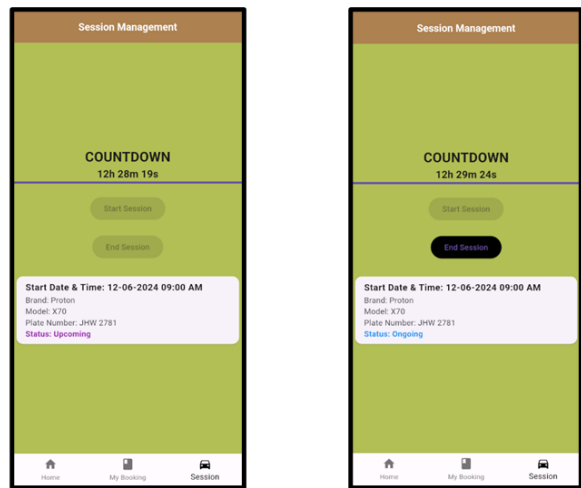


Fig. 12 Rental Session Interface

Fig. 13 shows the page where staff can track the location of their customers during their rental session. The booking with status Ongoing only will be displayed on this page, and staff able to track those customers.

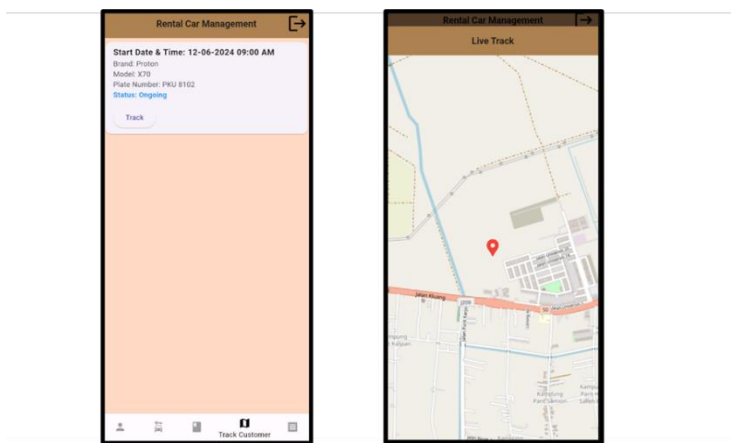


Fig. 13 Track Customer Location Interface

Fig. 14 shows the Fuel Status Form that customers need to fill in after ending their rental session. They need to fill in the details, select whether they refuel the car or not, and have to upload the picture of the dashboard meter of the rental car.

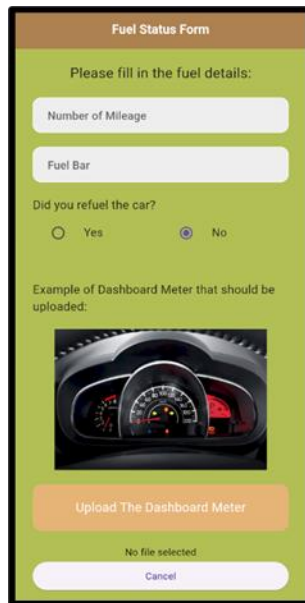


Fig. 14 Fuel Status Form Interface

Fig. 15 depicts a page where staff can generate the report about the data from this car rental system. The staff needs to choose the report type to be generated and click generate report. This report also can be downloaded as a Portable Document Format (PDF) file.



Fig. 15 Generated Report Interface

4. Result and Discussion

Table 6 below indicate the test case for nine (9) modules involved in this system, which are The modules that have been involved in this system are Verify Authentication (TEST_100), Manage Personal Information (TEST_200), Check Availability (TEST_300), Manage Catalog of Rental Car (TEST_400), Payment (TEST_500), Manage Booking (TEST_600), Manage Rental Session (TEST_700), Update Fuel Status (TEST_800), and View Generated Report (TEST_900).

Table 6 List of Test Cases

No.	Test Cases	Description	Result
TEST_100			Fail/Pass
1.	TEST_100_001	Customer can submit the register after filling in the registration form and tick the Terms and Condition	Pass
2.	TEST_100_002	Registered customer can track the status of account verification after enter email in Track Status Page.	Pass
3.	TEST_100_003	Verified customer can use the credentials and choose Customer role to login into the system (redirect to Home Page)	Pass
4.	TEST_100_004	Staff login to the system with staff credentials and choose Admin role	Pass
5.	TEST_100_005	Staff can view all the customer details and the list of customers waiting the account to be verified	Pass
6.	TEST_100_006	Staff can verify the account by click on Verify button and reject by click on Reject button.	Pass
7.	TEST_100_007	Account that has been disabled by staff or system cannot log in into the system.	Pass
TEST_200			Fail/Pass
1.	TEST_200_001	The details of the customer will be showed when customer enter the Profile section	Pass
2.	TEST_200_002	The details of the customer can be updated	Pass
3.	TEST_200_003	The details of the customer can be canceled when to update	Pass
TEST_300			Fail/Pass
1.	TEST_300_001	Time selection filter the time that has passed from the actual time	Pass
2.	TEST_300_002	Show only available car based on date and time selected	Pass
TEST_400			Fail/Pass
1.	TEST_400_001	Show the details of the available rental car	Pass
2.	TEST_400_002	The price was displayed after customer choose the rental period of that rental car either in Hours or Days	Pass
3.	TEST_400_003	The new rental can be added by staff after filling in the rental car details	Pass
4.	TEST_400_004	The details of the rental car can be edit	Pass
5.	TEST_400_005	The existing rental car in the system can be delete	Pass
6.	TEST_400_006	The plate number of each rental car can be added	Pass
7.	TEST_400_007	The plate number of each rental car can be edit	Pass
8.	TEST_400_008	The plate number of each rental car can be deleted	Pass
TEST_500			Fail/Pass
1.	TEST_500_001	Show the summary of booking to the customer which include Customer Details, Rental Session Details, Car Details, and Booking Details.	Pass
2.	TEST_500_002	The file of booking payment receipt can be uploaded	Pass
3.	TEST_500_003	The booking can be made after successfully make the payment by uploading the booking payment receipt	Pass
TEST_600			Fail/Pass
1.	TEST_600_001	The system shows all the bookings made by customer with different status	Pass
2.	TEST_600_002	The bookings can be filter with filter option	Pass
3.	TEST_600_003	More details about the booking is shown after click on the bookings card	Pass
4.	TEST_600_004	Penalty payment can be made for booking status: Completed (Pending Penalty Payment)	Pass
5.	TEST_600_005	Staff view the receipt of booking payment and penalty payment	Pass
TEST_700			Fail/Pass
1.	TEST_700_001	Session page show the countdown of the nearest upcoming booking	Pass
2.	TEST_700_002	Initiate the rental session by click on Start Session and stop the rental session by click on End Session	Pass

Table 6 (Cont.)

3.	TEST_700_003	Staff track the location of the customer during rental session by view the map	Pass
4.	TEST_700_004	Confirmation dialog was showed when customer click End Session button	Pass
TEST_800			Fail/Pass
1.	TEST_800_001	The Fuel Status form show all the details that customer need to fill	Pass
2.	TEST_800_002	The image of dashboard can be uploaded	Pass
3.	TEST_800_003	Redirect to Home Page after submitting the Fuel Status form	Pass
TEST_900			Fail/Pass
1.	TEST_900_001	The report is generated when staff choose the report type	Pass
2.	TEST_900_002	The report can be downloaded	Pass

A total of 37 test cases had been conducted to test the Car Rental System with Fuel Update. The system has passed successfully. **Table 7** shows the overall result of the test cases.

Table 7 Overall Result of Test Cases

Test Case ID	Total Test Cases	Total Pass	Total Fail
TEST_100	7	7	0
TEST_200	3	3	0
TEST_300	2	2	0
TEST_400	8	8	0
TEST_500	3	3	0
TEST_600	5	5	0
TEST_700	4	4	0
TEST_800	3	3	0
TEST_900	2	2	0
Total	37	37	0

5. Conclusion

In short, the project of Kereta Sewa Jimat: Car Rental System with Fuel Update was successful in achieving what it wanted by creating a user-friendly mobile app. This can help resolve booking errors, effective management of fuel, and complete information about cars to enhance customer satisfaction by increasing operational efficiency. Even tt lacks the features of notifications, separate apps for customers and staff, and the track option works only if the customer activates the application, the system also have some majorly resulting advantages include full details of cars, fewer errors in booking, better management, tracking security issues from the customers' side, and avoiding losses to the firm because of overlooking fuel. Hopefully, it will enhance users' experience with renting cars and, to the car rental industry, work toward optimization of operations and quality services.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

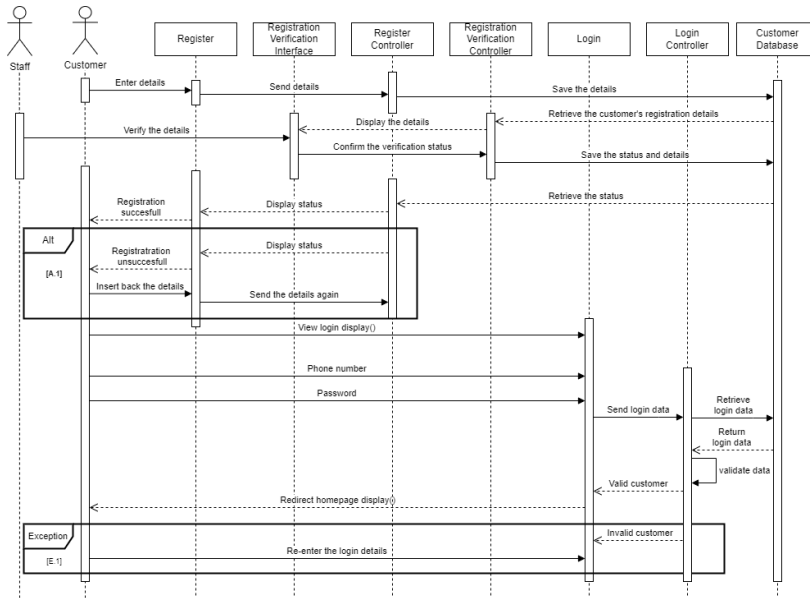
Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

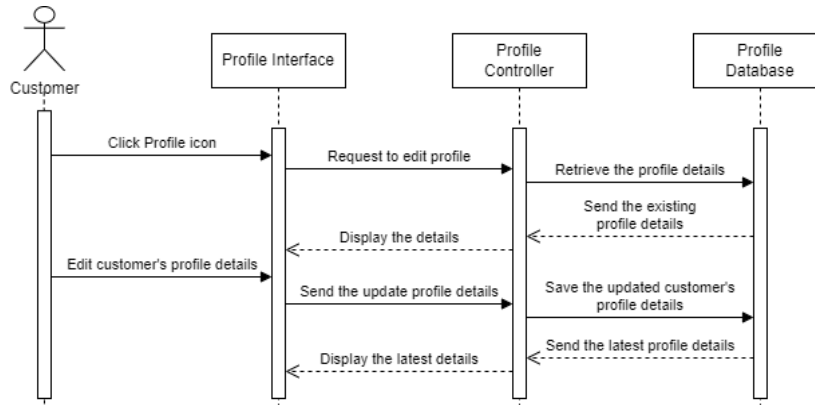
Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

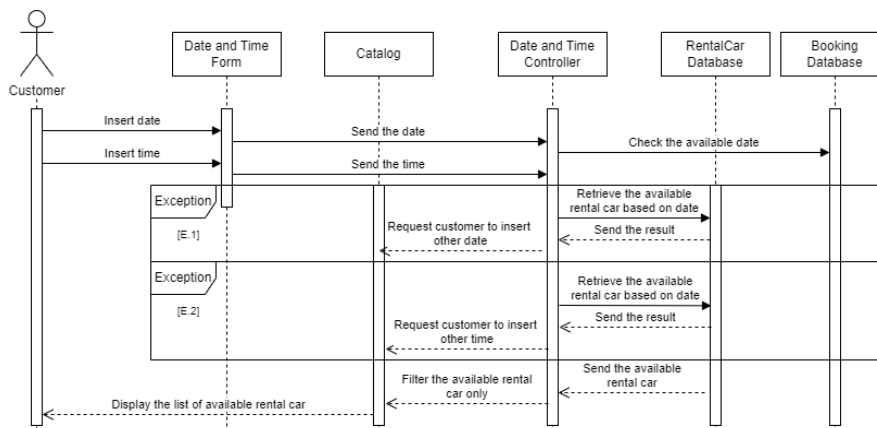
Appendix A: Sequence Diagram



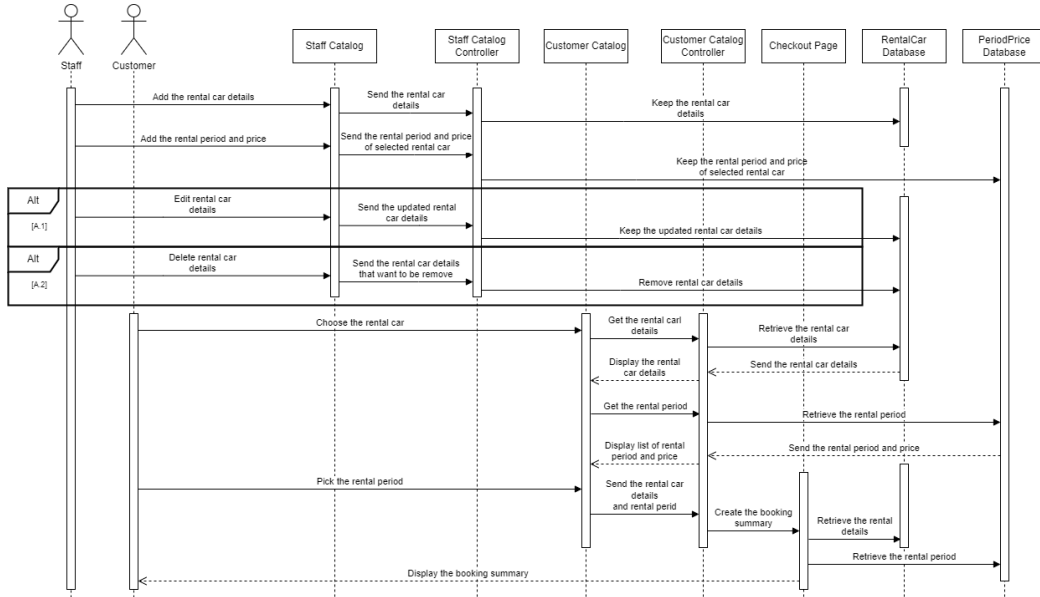
Appendix A.1: Sequence Diagram for Verify Authentication



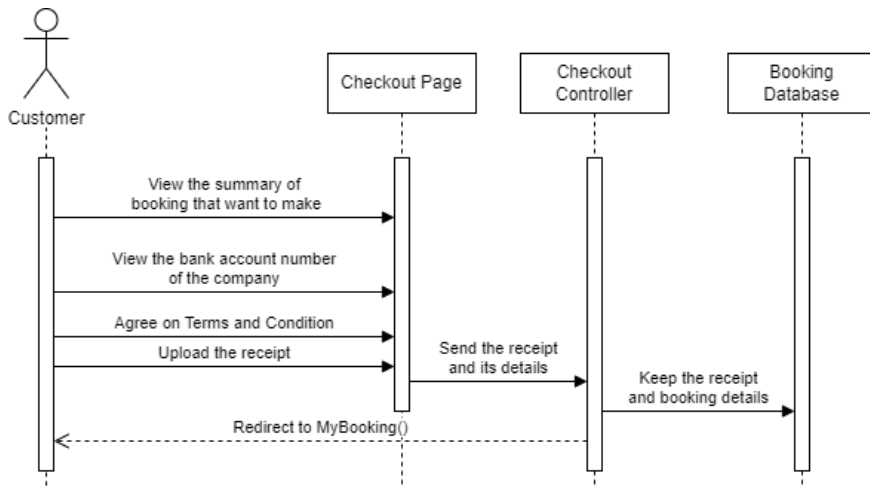
Appendix A.2: Sequence Diagram for Manage Personal Information



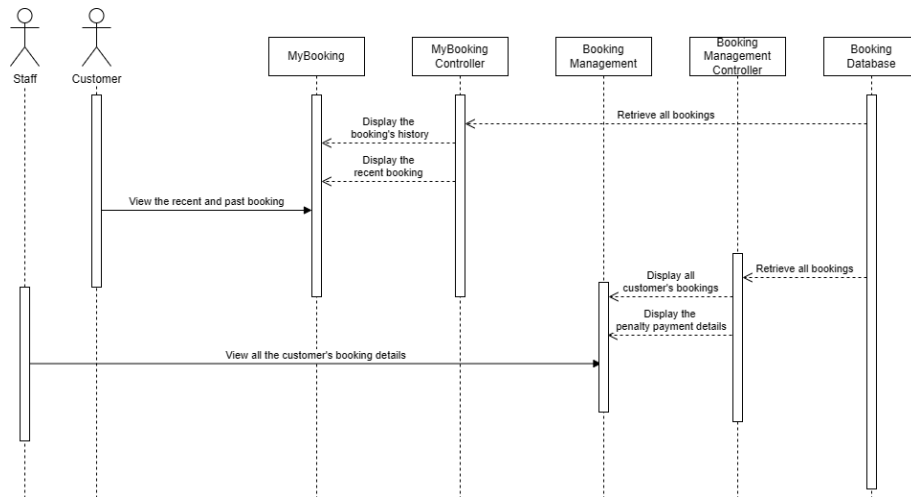
Appendix A.3: Sequence Diagram for Check Availability



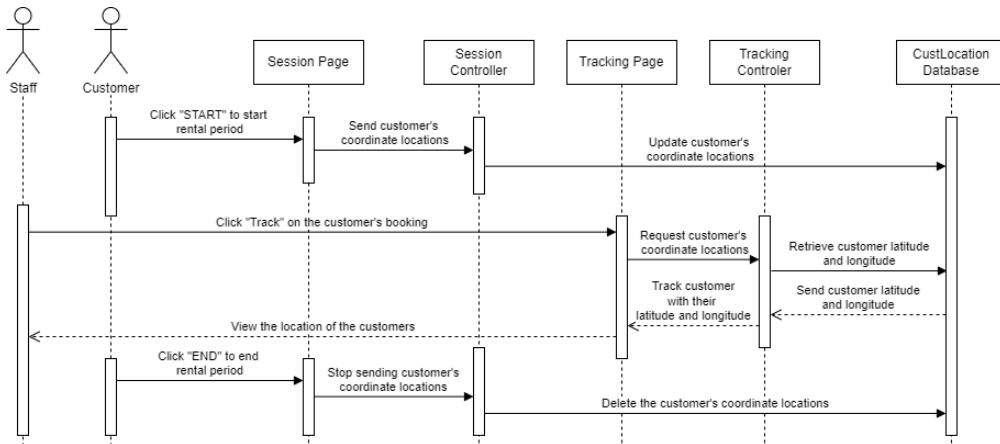
Appendix A.4: Sequence Diagram for Manage Catalog of Rental Car



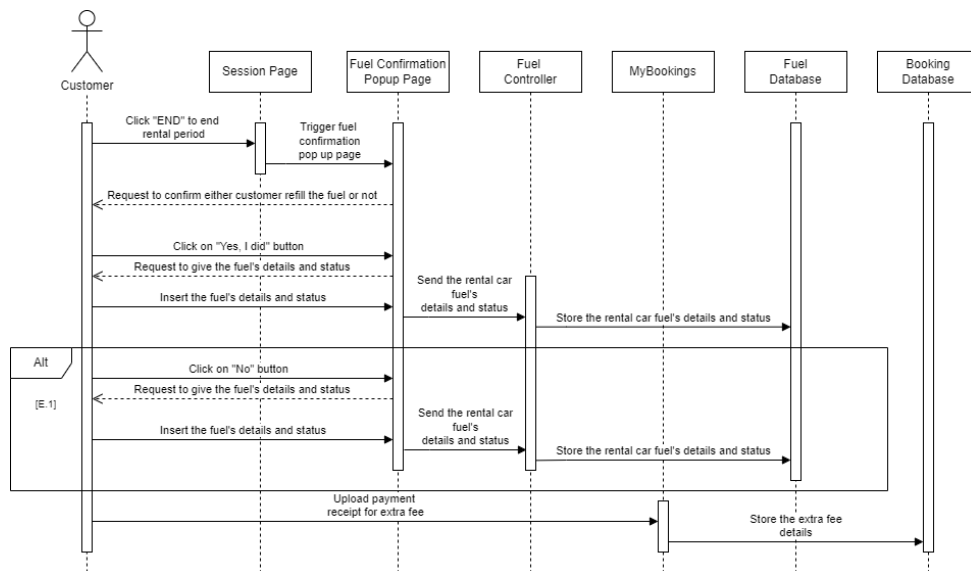
Appendix A.5: Sequence Diagram for Payment



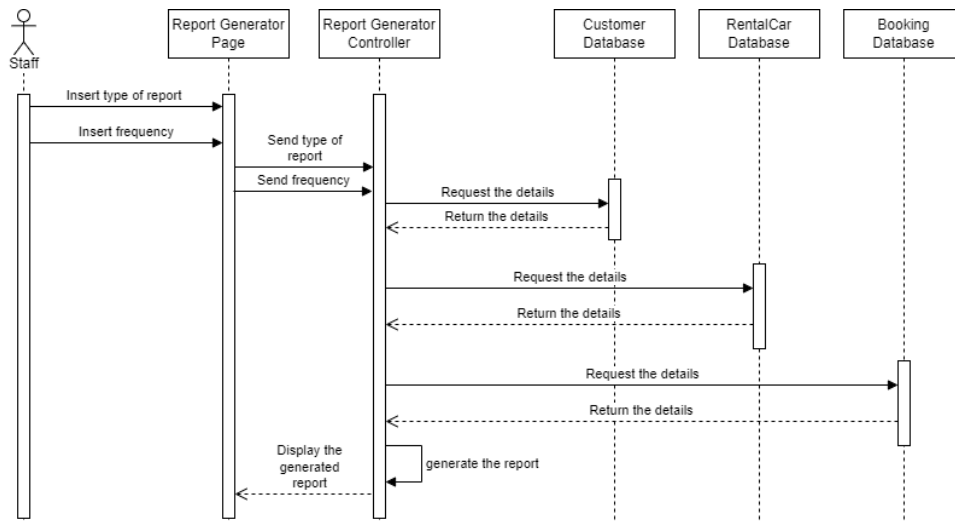
Appendix A.6: Sequence Diagram for Manage Booking



Appendix A.7: Sequence Diagram for Manage Rental Session



Appendix A.8: Sequence Diagram for Manage Update Fuel Status



Appendix A.9: Sequence Diagram for Manage View Generated Report

References

- [1] C. Rafati. (2021, November 5). "What are the benefits of online car rental software?" Rentrax. <https://rentrax.com/what-are-the-benefits-of-online-car-rental-software/>
- [2] MeltAdmin. (2023, March 3). "The Importance of Car Rental Services for Travelers," AAA Vehicle Rental. <https://aaavehiclerental.com/2023/03/31/the-importance-of-car-rental-services-for-travelers/>
- [3] A. Phongtraychack and D. Dolgaya, "Evolution Of Mobile Applications," MATEC Web of Conferences, vol. 1, pp. 1-7, 2018. https://www.matec-conferences.org/articles/mateconf/pdf/2018/14/mateconf_imet2018_01027.pdf
- [4] Moteria. (n.d.). "What is Flutter? Get Familiar With Its Features And Advantages," Elluminati Inc. <https://www.elluminatiinc.com/what-is-flutter/>
- [5] SOCAR. (n.d.). "SOCAR - Free yourself. Own the experience". <https://socar.my/>
- [6] S. Kumar. (2016, February 16). "Prototype Model in Software Engineering," Scaler. <https://www.scaler.com/topics/software-engineering/prototype-model-in-software-engineering/>
- [7] P. Martinez. (2024, June 6). "What is Evolutionary Prototype?" Mockitt. <https://mockitt.wondershare.com/prototyping/evolutionary-prototyping.html>
- [8] MBA Skool. (2024, February 1). "Prototype - Definition & Meaning," MBA Skool. <https://www.mbaskool.com/business-concepts/it-and-systems/13712-prototype.html>
- [9] BasuMallick. (2022, June 3). "What Is GPS (Global Positioning System)? Meaning, Types, Working, Examples, and Applications," Spiceworks. <https://www.spiceworks.com/tech/iot/articles/what-is-gps/>
- [10] Dennis, B. H. Wixom, and D. P. Tegarden, System Analysis & Design, an Object-Oriented Approach with UML, Wiley, 2020.
- [11] K. Pohl and C. Rupp, "Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam; foundation Level - IREB compliant," Rocky Nook, 2015.