Applied Information Technology And Computer Science Vol. 2 No. 1 (2021) 337-375
© UniversitiTun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Rizqi Car Rental Booking System Based on Mobile Applications

Nur Suhana Alid, Ruhaya Ab. Aziz*

Faculty of Computer Science & Information Technology, University Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2021.01.01.026
Received 31 March 2021; Accepted 11 April 2021; Available online 31 May 2021

Abstract: Rizqi Car Rental Booking System based on Mobile Application developed to enable customers to book rental cars online that can provide a wealth of information related to car rental reservations. In addition, the system was developed to facilitate the car rental information management of Rizqi Car Rental. Based on observation, this system offer a user feedback module compared to the existing system that does not offer a user feedback module. In order to make the Rizqi Car Rental Booking System based on Mobile Application, the waterfall development process model is used as a process model that acts as a guideline that covers several phases namely planning, analysis, design, and implementation. At the end of the system development, customers can make online car rental reservations that are seen to provide comfort and satisfaction to customers and help the management of Rizqi Car Rental Company booking to run smoothly.

Keywords: booking system, mobile application,online, car rental

1. Introduction

A rental car is a vehicle that can be used temporarily with payment for a certain period. Using a rental car can help people who do not have access to their own personal vehicle or do not own a vehicle. Car Rental Companies offer different types of cars and reasonable prices to their customers. Therefore, customers can choose the type of car they want according to the travel distance and customer's cost. These advantages can help save costs such as car fuel and car maintenance.

Rizqi Car Rental Company is a car rental agency located in Parit Raja. The agency is owned by a married couple and managed by them along with several staff members. The agency operates daily from 8 a.m. to 12 a.m. The management of this agency is managed manually and still does not have an online car rental management system. Rizqi Car Rental Company is very popular among UTHM students who live in Perwira and Bestari Residential College because not all students have their own vehicles to move from residential college to other places without worrying about transportation. In addition, the services offered by this company are very good and are located close to Perwira Residential College and Bestari Residential College.

Rizqi Car Rental Booking System based on Mobile Application is a system developed to make staff easier to manage customer's data and information. This car rental system will help Rizqi Car Rental

*Corresponding author: ruhaya@uthm.edu.my

Company run smoothly. Information management for Rizqi Car Rental Company can also be managed regularly and consistently. Staff also can enter the information to keep in the system and the stored information can be updated and deleted from the system. In addition, the system will help staff more easily to deal with customers. Using this system can make customers easier to booka rental car online.

2. Literature Review

Literature Review is a method of gathering information obtained from scientific sources such as books, magazines, articles, newspapers, and other sources suitable for this study. The information obtained is usually related to the title of the study being conducted. This method involves the process of selection of ideas, collection, and exchange of information. Next, this study is conducted to make a comparison of existing systems and problems encountered as well as analyze important information to develop and suggest better systems.

According to [1], a literature review is the search and evaluation of availableliterature in selected subjects or topics. It is a document for the subject or topic being studied. Next, according to [2], literary study is the process of analyzing information in books, scientific articles, and other sources related to the topic that provides the basis of knowledge about a topic.

This study was conducted to help improve the level of access efficiency and quality in the developed system. In the process of developing this system, several studies on existing systems have been implemented for the data collection process. These studies are conducted to ensure that the objectives of the system to be developed toachieve the desired characteristics.

2.1 Information and Booking System

An information system is a set of combined components in the data storage process to collect, manage, receive, and store information to support the management and decisions made by certain organizations. According to [3], information systems are a combination of information technology and user activities to support the operation and management of networks that complement hardware and software and organizations that use information technology to manage data. According to [4], information systems are the glue that unites all management planning, organizing, directing, and controlling in one overall activity.

Booking was defines as something that has been booked [5]. In the proposed system, users book a rental car through the online system to get a rental carand set the desired time and date.

Information and booking system are very important for an organization because it can reduce operating costs. In addition, the information and booking system provides facilities to the organization to manage information and reservations and facilitates customers to deal with the organization. Next, the data management process becomes more efficient and can help solve the problems encountered. This system can also help organizations improve marketing and produce the best marketing strategies.

2.2 Mobile Application

A mobile application is a software or a series of programs that run on a mobile device and perform specific tasks for its users. Mobile applications are a new and rapidly evolving software development segment in Information and Communication Technology. Mobile applications are very easy to use, user-friendly, inexpensive, downloadable, and can work on all types of mobile devices.

Rizqi Car Rental Booking System based on Mobile Application can make it easier for users to make reservations with only through mobile devices. Users just need to download this application on the smartphone and can book a rental car anywhere. This Mobile Application Car Rental Booking System is developed with user-friendly features. This can help users more easily use this system.

2.3 Study of Equivalent System

This study was conducted on several equivalent systems to identify the advantages and disadvantages of the developed system. In addition, this study was conducted to collect important information for the proposed system, namely the Rizqi Car Rental Booking System based on Mobile Application.

2.4 Comparison of Study Results

Based on the studies and comparisons that have been done, there are similarities and differences between the equivalent system and the proposed system. This comparison is made by evaluating the features found in the equivalent system and the proposed system.

Table 1: Comparison of system equivalent and proposed system

Characteristics	KLeZCAR	SOCAR	KARLOOP	Rizqi Car Rental Booking System
Platform	Based on Web	Based on Mobile Application	Based on Web	Based on Mobile Application
System Module / Menu:				
-User registration	Yes	Yes	No	Yes
-Make a booking	Yes	Yes	Yes	Yes
-Booking Period	Yes	Yes	Yes	Yes
-Rental car list view	Yes	Yes	No	Yes
-User feedback	No	No	No	Yes
-Logout	Yes	Yes	No	Yes
Online Payment	No	No	No	No
Update data	No	Yes	No	Yes

Table 1 shows the similarities and differences of the features found in the equivalent system and the proposed system. As a result, the proposed system offers user feedback modules that are not available on all three equivalent systems. In addition, all three equivalent systems and the proposed system do not offer online payment modules.

3. Methodology

The methodology is a method or technique used for design, analysis, and data collection to achieve the objectives and goals of the study. The methodology helps the project journey to be more systematic and organized. The selection of methodology is very important to ensure the development of the project according to a set schedule.

According to [6], a methodology is defined as a system that includes methods and principles used in an activity or discipline. According to [7], a methodology is a set of methods used to research a particular study subject.

The software process model used in the Rizqi Car Rental Booking System based on this Mobile Application projectis a waterfall development process model. This section will describe the phases found in the waterfall development process model and the hardware and software requirements for this system.

3.1 Waterfall Development Process Model

The waterfall development process model is the first process model in the methodology. Using this process model, the analyst or user continues the phase-in sequence from one phase to another. The process model is known as waterfall development because this process model moves forward from one phase to another in the same way like a waterfall. The advantages of using this process model is users can identify system requirements before programming started and minimize changes of requirements while the project is ongoing. The disadvantage of using a waterfall development process model are that the design must be fully determined before programming started and take a long time for the solution of the proposed system in the analysis and delivery phase of the system. Figure 1 shows the activities carried out in the waterfall development process model.

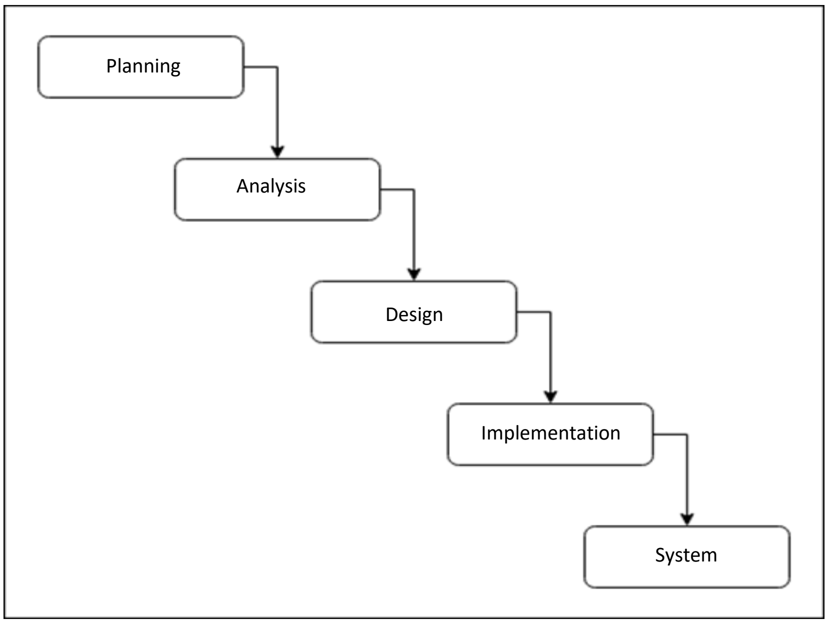


Figure 1: Waterfall Development Process Model [8]

Based on figure 1, this waterfall development process model has four phases namely planning phase, analysis phase, design phase, and implementation phase.

The planning Phase is the main phase in the development of the Rizqi Car Rental Booking System based on Mobile Application. This phase aims to provide an understanding of how the system will be developed. In addition, problems, objectives, and scope can be identified in this phase. Data collection from various types of sources can also be done in this phase. Next, this phase is very important to give an initial overview of the system requirements to be developed. Gantt charts and proposal papers are also produced to ensure the development of this system runs smoothly according to the set time.

The analysis phase is the phase where information is analyzed to determine the advantages and disadvantages of the system to be developed. This phase aims to understand the needs of the system and determine the aspects needed to develop this system. Next, the booking system can be developed according to the features required by the user. Therefore, an interview session with the car rental company Rizqi and the project supervisor was conducted to get their opinion on the proposed system. Furthermore, the information obtained is studied in more detail to facilitate the development process of the Rizqi Car Rental Booking System based on Mobile Application. In addition, hardware and software are required to support the development of Rizqi Car Rental Booking System based on Mobile Applications.

In the design phase, there are three types of design in this phase, namely interface design, database design, and process design. The interface design is a visual layout element that allows users to interact with the system. The user-friendly and interactive interface is designed for the Rizqi Mobile Car Rental Booking System based on the Mobile Application. Database design, on the other hand, is organizational data according to a database model that involves data classification and identifying relationships. Process design is the act of changing the vision and goals of the organization as well as existing resources into a visible and achievable way to achieve the vision of the organization. This design phase helps produce a cost-effective booking system.

The implementation phase is carried out based on the objectives and needs of the users that have been identified. The system is developed with the involvement of the programming language implemented into the system. This aims to produce a real system design using programming languages such as Java, PHP, and so on. In order to develop a good system, the use of an appropriate programming language is essential. Next, implement the system interface and functions using the selected programming language. The programming language that has been used to develop the Rizqi Car Rental Booking System based on Mobile Application is Java.

Hardware and software are requirements that support the process of building a Car Rental Booking System. The following is a list of hardware and software that will facilitate and streamline the system development process. The hardware and software used in developing this system are:

List of hardware requirements:

I. Personal Computer - ACER Aspire E1 -421

II. Processor – AMD Dual Core Processor E2 1800 (1.7 GHz)

III. Operating System – Windows 8

IV. RAM – 8.00 GB

V. Input Device – Keyboard, mouse

List of software requirements:

I. Android Studio 3.5

II. Microsoft Word 2010

III. Microsoft Power Point 2010

IV. SQL server

V. Windows 8

4. Analysis and Design

The analysis is the description of a situation or problem from various aspects in detail. The design is the process of defining architecture, modules, interfaces, and data to meet the requirements of a defined system. Analysis and design are important phases in system development. All information and data collected will be analyzed to ensure that the system meets the objectives, various guidelines, and methods used in the work implementation phase have been provided.

This section will explain the analysis and design of the Rizqi Car Rental Booking System based on Mobile Application. Some diagrams are built based on object-oriented. This section will also describe the requirementstraceability matrix in the Rizqi Car Rental Booking System based on Mobile Application.

4.1 System Analysis

Unified Modeling Language (UML) is a modern approach to modeling and documenting software. UML is a diagram that aims to represent the system with main actors, roles, actions, artifacts, or classes, to better understand, modify, maintain, or document information about the system.

4.1.1 Use Case Diagram

A use case diagram is a dynamic or behavioral diagram in UML. Use case diagrams work by using actors and use cases. A use case diagram is a set of actions, services, and functions that a system needs to perform. Figure 2 is a use case diagram for the Rizqi Mobile Car Rental Booking System based on Mobile Application.

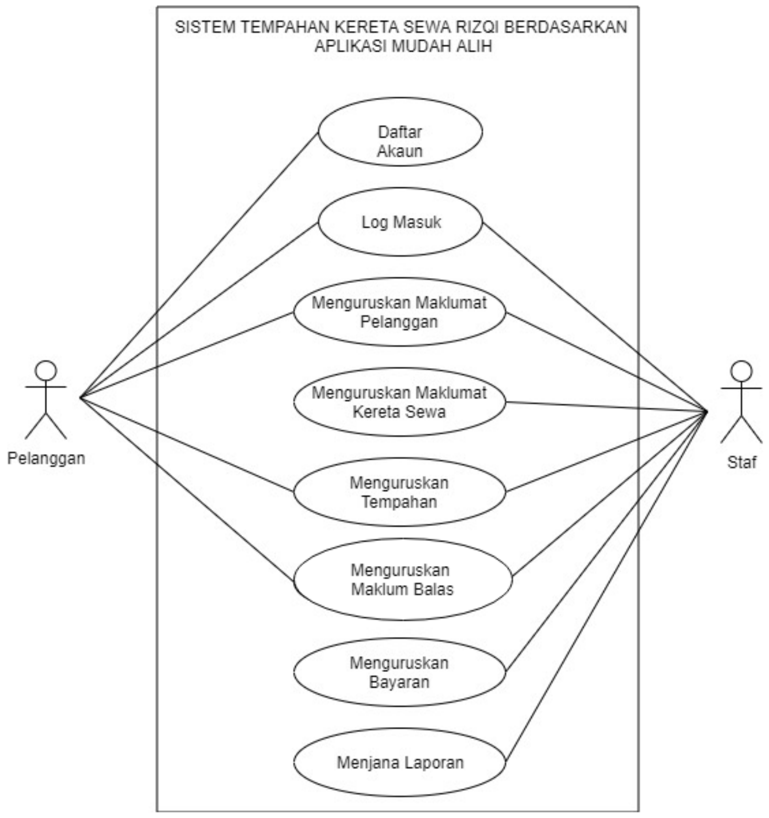

SISTEM TEMPAHAN KERETA SEWA RIZQI BERDASARKAN APLIKASI MUDAH ALIH
Daftar Akaun
Log Masuk
Menguruskan Maklumat Pelanggan
Menguruskan Maklumat Kereta Sewa
Menguruskan Tempahan
Menguruskan Maklum Balas
Menguruskan Bayaran
Menjana Laporan
Pelanggan
Staf

Figure 2: Use Case Diagram forRizqi Mobile Car Rental Booking System based on Mobile Application

Figure 2 shows the use case diagram for Rizqi Mobile Car Rental Booking System based on Mobile Application. In this system, there are two users involved, namely staff and customers. The system has eight modules namely register, login, manage customer information, managerental car information, manage bookings, manage feedback, manage payment, and generate reports. The use case specification table based on each module stated in the use case diagram can be referenced in Appendix A.

4.1.2 Activity Diagram

An activity diagram is a UML diagram that focuses on the implementation and behavioral flow of a system and not its implementation. Activity diagrams are also known as object-oriented data streams. Activity diagrams consist of actions used for behavior modeling technology. The activity diagram can be referenced in Appendix B.

4.1.3 Sequence Diagram

Sequence diagrams are diagrams that show the interaction of objects arranged in a time sequence. The reference diagram describes the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to perform the scenario function. Sequence diagrams can be referenced in Appendix C.

4.1.4 Class Diagram

The class diagram shows the static structure of the object-oriented model, this includes the object class, its internal structure, and the relationship between the classes. In Unified Modeling Language (UML), a rectangle with three sections separated by a horizontal line represents the class. The class name is at the top of the square, while the list of attributes is in the middle and the list of operations is at the bottom of the square. Figure 3 is a class diagram for Rizqi Mobile Car Rental Booking System based on Mobile Application.

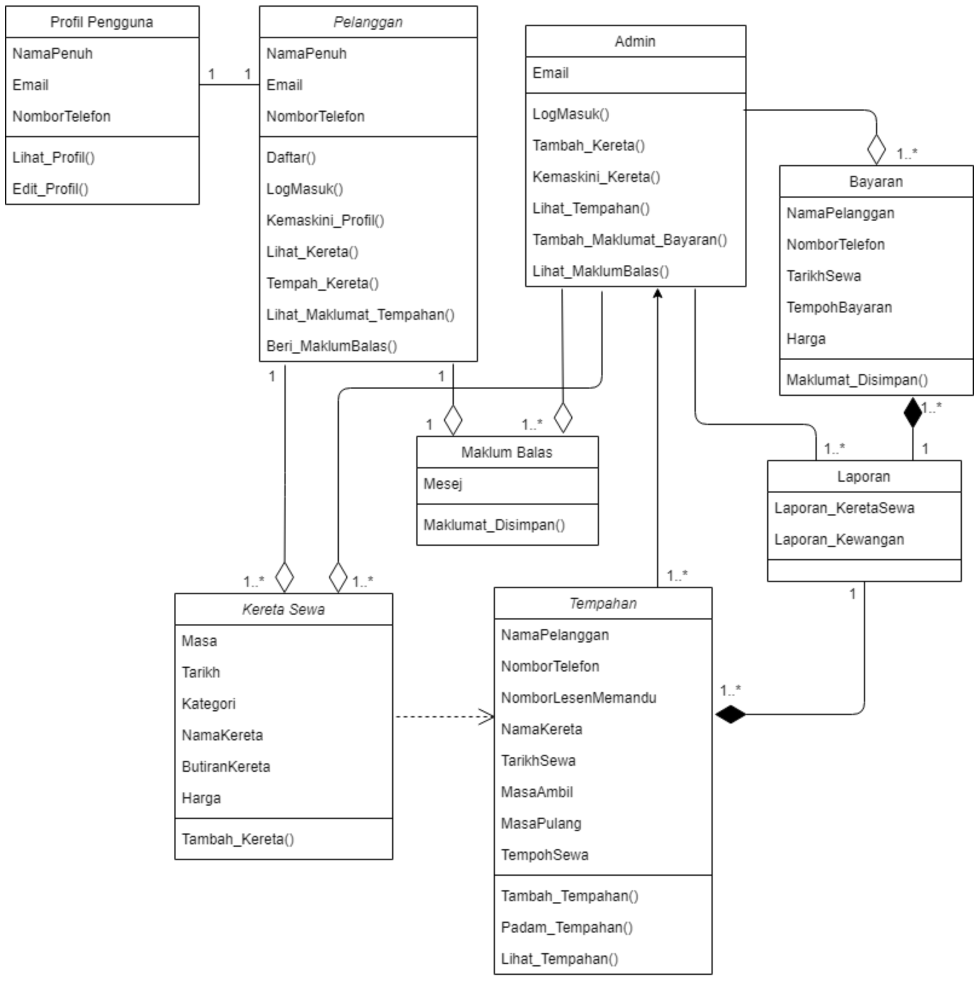


Figure 3: Class DiagramRizqi Mobile Car Rental Booking System based on Mobile Application

Figure 3 shows the static structure of the Rizqi Car Rental Booking System based on Mobile Application. The system has eight classes along with a list of attributes and a list of operations.

4.2 System Design

System design is important in developing a system because it can give a clear picture of the data entered and the results to be released.

4.2.1 Interface Design

Interface design aims to provide an overview of the system developed for system users. The interface design of the Rizqi Car Rental Booking System based on Mobile Application can be referenced in Appendix D.

4.3 Requirements Traceability Matrix

The requirements traceability matrix is a table showing that each requirement has its own test case or case to ensure that all requirements specified for the system are tested in a test protocol. Table 2 showsrequirements traceability matrix for Rizqi Car Rental Booking System based Mobile Application.

Table 2: Requirements Traceability Matrix Rizqi Car Rental Booking System based on Mobile Application

ID	Description
REQ_100	**Register**
SRS_REQ_101	Customerselect register new account button.
SRS_REQ_102	Customer fill in the required information such as full name, email, password, and phone number.
SRS_REQ_103	Customer selectthe register button.
REQ_200	**Login**
SRS_REQ_201	Customer fill in email and password.
SRS_REQ_202	Customer select login button.
SRS_REQ_203	Customer successfully logged into the system.
SRS_REQ_204	Waiting for the customer to fill in the correct email and password.
REQ_300	**Manage Customer Information**
SRS_REQ_301	Customer select the update button on the display customer information.
SRS_REQ_302	Customer updates the required information.
REQ_400	**Manage Rental Car Information**
SRS_REQ_401	Staff select the rental car information button.
SRS_REQ_402	Staff fill in rental car information.
REQ_500	**Manage Booking**
SRS_REQ_501	Customer select the booking button.
SRS_REQ_502	Customer choose a rental car.
SRS_REQ_503	Customer fill in the booking information such as driver name, phone number, start and end travel dates as well as start and end times.
SRS_REQ_504	Customer select the booking button
SRS_REQ_505	Customerchoose a rental car again.
REQ_600	**Manage Feedback**
SRS_REQ_601	Customer select the feedback button.
SRS_REQ_602	Customer gives feedback.
SRS_REQ_603	Customer selectthe finish button.
REQ_700	**Manage Payments**
SRS_REQ_701	Staff select the payment button
SRS_REQ_702	Staff fill in payment information such as customer name, telephone number, rental period, rental date, and price.
SRS_REQ_703	Staff selects the view data button.
SRS_REQ_704	Staff select save button to save payment information.
REQ_800	**Generate Reports**
SRS_REQ_801	Staff select the report button

Table 2 shows the requirements traceability matrix for each module in the Rizqi Car Rental Booking System based on Mobile Application. Each module has its own test case or case to ensure that the requirements of each module are tested according to the test protocol.

4.3 Data Dictionary

A data dictionary is a set of database tables used to store information about the definition of a database. This data dictionary contains information about database objects such as tables, indices, columns, data types, and views. The user table for each module available in the Rizqi Car Rental Booking System based on Mobile Application can be referenced in Appendix E.

5. Implementation and Testing

The implementation phase is carried out based on the objectives and needs of each module that has been identified. The system will be developed with the involvement of the programming language implemented into the system and software that has been determined. The testing phase is carried out to test and evaluate the system developed to repair the deficiencies found in the system. Every function of the developed system is tested to avoid errors that will affect the system.

This chapter describes the implementation and testing carried out on the Rizqi Car Rental Booking System based on Mobile Applications. The program code in this system will also be explained based on the modules available in this system.

5.1 Implementation

The purpose of implementation is to build the right work system, install it in the organization, replace old systems and work methods, complete system and user documentation, train users and provide support systems to help users. The implementation of the system has six main activities namely coding, testing, installation, documentation, training, and support. This step aimsis to transform physical system specifications into functional and reliable software and hardware. The implementation of the Rizqi Car Rental Booking System can be found in Appendix F.

5.1.1 Register Module

This register module is used by customers to register a new account. Customers only need to fill in information such as full name, email, password, and phone number to register a new account.

5.1.2 Login Module

This Login Module has two parts, namely the customer and admin section. This module is used by users logging in and logging out of the system. Users need to enter email and password into the system.

5.1.3 Manage Customer Information Module

This Manage Customer Information Module is used by customers to view and update information such as full name, email, and phone number.

5.1.4 Manage Rental Car Information Module

This Manage Rental Car Information Module is used by staff to add rental car information into the system. Information such as car photos, car name, car details, and car price for an hour of rental is required when filling out rental car information.

5.1.5 Manage Booking Module

This Manage Booking Module is used by customers to make rental car bookings. Customers will choose the type of car and make a reservation by filling out the booking form. Staff will accept and confirm orders made by customers.

5.1.6 Manage Feedback Module

The Manage Feedback Moduleis used by customers. Customers can provide suggestions or comments on the services provided by the Car Rental Company only through the system. Feedback sent by customers will be displayed in the admin system as a reference toRizqi Car Rental Company to improve the system or service.

5.1.7 Manage Payments Module

The Payment Management Module is used by staff. Staff will fill in information such as customer name, customer phone number, rental period, total price, and rental date manually into the system when the customer makes the payment. Staff can also view the display of data that has been entered into the system. The payment process made at Rizqi Car Rental Company is in cash.

5.1.8 Generate Reports Module

The Generate Reports Module is used by staff. Staff can choose whether to view financial reports or booking reports.

5.2 Testing

System testing is a verification activity performed on the system to ensure that the system is developed according to the specifications of the requirements and design of the system. System testing is done to observe how the system works and test whether the application is easy to use or otherwise. In addition, this system is also tested to find out how this system works if the input entered is correct and an error occurs if the input entered is incorrect. The table of test cases according to the modules in the Rizqi Car Rental Booking System can be found in Appendix G.

5.1. Conclusion

In conclusion, Rizqi Car Rental Booking System based on Mobile Application was developed to make customers easier to book rental cars online. In addition, this system also helps to solve the problem of information and booking management of Rizqi Car Rental Company.

Acknowledgement

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

References

[1] Fund, R. L. (2019). Literature Review. Retrieved from https://www.rlf.org.uk/resources/what-is-a-literature-review/.

[2] Adibah, U. (2019). Apa itu Kajian Literatur? Retrieved from https://www.pascasiswazah.com/apa-itu-kajian-literatur/.

[3] Raditya (2012). Information System. Retrieved from http://raditya- pw.blogspot.com/.

[4] Diwan, P. (1999). Information Management System. Malaysia: Golden Book Centre SDN. BHD.

[5] Pustaka. (2017). Dewan Bahasa dan Pustaka. Retrieved from http://prpm.dbp.gov.my/Cari1?keyword=mendefinisikan.

[6] KamusDewanEdisiKeempat (2005). Kuala Lumpur: Dewan Bahasa danPustaka.

[7] Hornby, A. S. (1995). Oxford Advanced Learner’s Dictionary. Oxford: Oxford University Press.

[8] David, T. Alan, D. and Barbara (2013). System Analysis and Design with UML Version 2, 4th Edition.

Appendix A: Use Case Specification Table

Table 3: Use Case Specification Table for Register Module

Use Case	Register
Use Case ID	1.0
Simple Description	New user register an account for the first time.
Actor	Customer
Situation Before	Display a login interface that contains the register button option for new users.
Main Flow	1. Customer select register new account button. (SRS_REQ101) 2. Customer fill in required information such as full name, email, password and phone number. (SRS_REQ102) 3. Customer select register button. (SRS_REQ103) 4. The system stores customer information.
Situation After	Display main page.
Alternative Flow	No

Table 4: Use Case Specification Table for Login Module

Use Case	Login
Use Case ID	2.0
Simple Description	User login into the system
Actor	Customer and Staff
Situation Before	Display login interface.
Main Flow	1. Customer fill in email and password. (SRS_REQ201) 2. Customer select login button. (SRS_REQ202) 3. If wrong email or password: (A-1: Wrong Information) 3.1 The system will ask the customer to refill the correct email and password. 4. Customer successfully logged into the system. (SRS_REQ203)
Situation After	Display main menu.
Alternative Flow	Waiting for the customer to fill in the correct email and password.

Table 5: Use Case Specification Table forManage Customer Information Module

Use Case	Manage Customer Information Module
Use Case ID	3.0
Simple Description	System updatescustomer’s personal information.
Actor	Customer and Staff
Situation Before	Display customer information
Main Flow	1. System display current personal information to customer. 2. Customer select the update button on the display customer information. (SRS_REQ301) 3. Customer updates the required information. (SRS_REQ302) 4. System update customer’s information.
Situation After	Customer’s information updated
Alternative Flow	No

Table 6: Use Case Specification Table for Manage Rental Car Information Module

Use Case	Manage Rental Car Information

Use Case ID	4.0
Simple Description	Users manage car rental information.
Actor	Staff
Situation Before	Display menu.
Main Flow	1. Staff select the rental car information button. (SRS_REQ401) 2. System asks staff to select the add button. 3. If Staff select add button: 3.1 The system displays the rental car information form. 3.2 Staff fill in rental car information. (SRS_REQ402) 4. System stores car rental information.
Situation After	Car rental information has been stored.
Alternative Flow	No

Table 7: Use Case Specification Table for Manage Booking Module

Use Case	Manage Booking
Use Case ID	5.0
Simple Description	System allows users to manage booking information.
Actor	Customer and Staff
Situation Before	Display main menu.
Main Flow	1. Customer select the booking button. (SRS_REQ501) 2. System displays available rental car information. 3. Customer choose a rental car (SRS_REQ502) 4. System display booking information form. 5. Customer fill in the booking information such as driver name, phone number, start and end travel dates and start and end times. (SRS_REQ503) 6. Send booking information. 7. If booking ussuccessful: (B-1:Booking Unsuccessful) 7.1 The system will ask the customer to make another car rental option. 8. Customer select the booking button. (SRS_REQ504) 9. Booking successful.
Situation After	Booking successful.
Alternative Flow	B-1 Booking Unsuccessful (SRS_REQ505) Customer choose a rental car again.

Table 8: Use Case Specification Table for Manage Feedback Module

Use Case	Manage Feedback
Use Case ID	6.0
Simple Description	The system allows users to gives feedback.
Actor	Customer and Staff
Situation Before	Display main menu
Main Flow	1. Customer select the feedback button. (SRS_REQ601) 2. Customer gives feedback. (SRS_REQ602) 3. Customer select the finish button. (SRS_REQ603) 4. System receives feedback.
Situation After	Feedback accepted
Alternative Flow	No

Table 9: Use Case Specification Table for Manage Payments Module

Use Case	Manage Payments
Use Case ID	7.0
Simple Description	System allows users to manage rental car information.
Actor	Staff
Situation Before	Display main menu.
Main Flow	1. Staff select the payment button (SRS_REQ701) 2. System displays the payment information form. 3. Staff fill in payment information such as customer name, telephone number, rental period, rental date and price. (SRS_REQ702) 4. Staff selects the view data button. (SRS_REQ703) 5. System stores payment information. 6. Staff select save button to save payment information. (SRS_REQ704) 7. System display payments information.
Situation After	Payment information has been stored into the system.
Alternative Flow	No

Table 10: Use Case Specification Table for Generate Reports Module

Use Case	Generate Reports
Use Case ID	8.0
Simple Description	The system allows users to generate reports.
Actor	Staff
Situation Before	Display main menu.
Main Flow	1. Staff select the report button (SRS_REQ801) 2. System displays the information that has been summarized.
Situation After	Report generated.
Alternative Flow	No

Appendix B: Activity Diagram

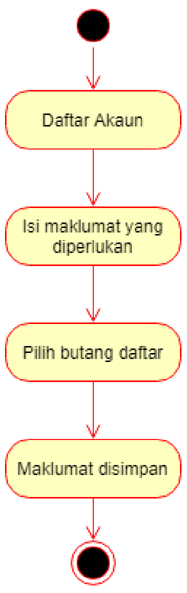


Figure 4: Activity Diagram for Register Module

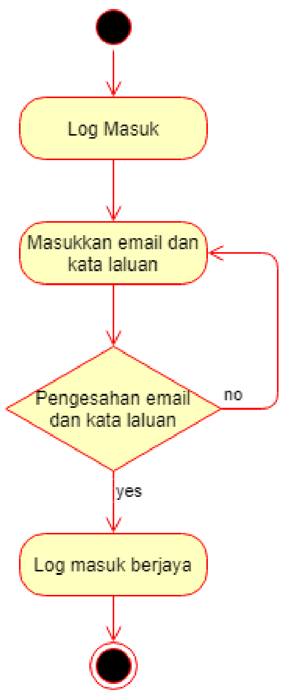

Log Masuk
Masukkan email dan kata laluan
Pengesahan email dan kata laluan
no
yes
Log masuk berjaya

Figure 5: Activity Diagram for Login Module

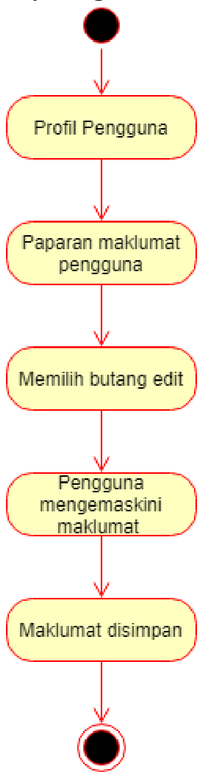


Figure 6: Activity Diagram Aktiviti for Manage Customer Information Module

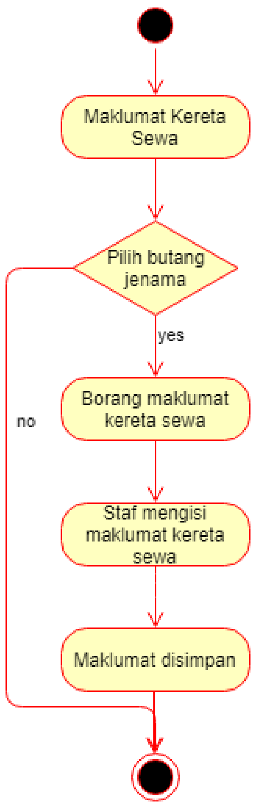


Figure 7: Activity Diagramfor Manage Rental Car Information Module

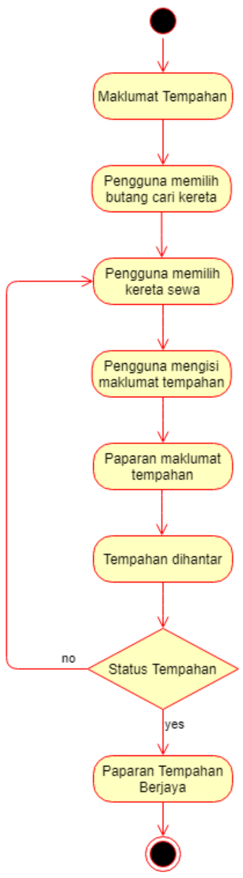

Maklumat Tempahan
Pengguna memilih butang cari kereta
Pengguna memilih kereta sewa
Pengguna mengisi maklumat tempahan
Paparan maklumat tempahan
Tempahan dihantar
no
Status Tempahan
yes
Paparan Tempahan Berjaya

Figure 8: Activity Diagramfor Manage Booking Module

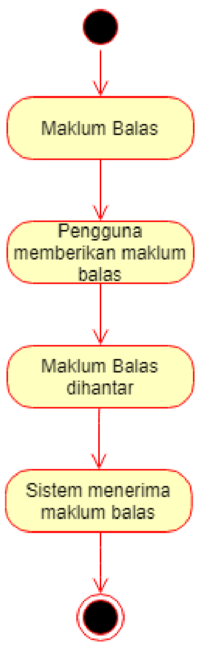

Maklum Balas
Pengguna memberikan maklum balas
Maklum Balas dihantar
Sistem menerima maklum balas

Figure 9: Activity Diagramfor Manage Feedback Module

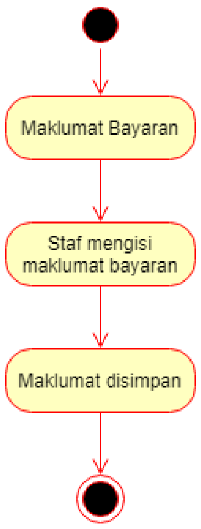


Figure 10: Activity Diagramfor Manage Payments Module

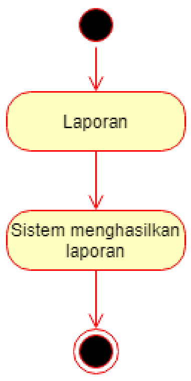


Figure 11: Activity Diagramfor Generate Reports Module

Appendix C: Sequence Diagram

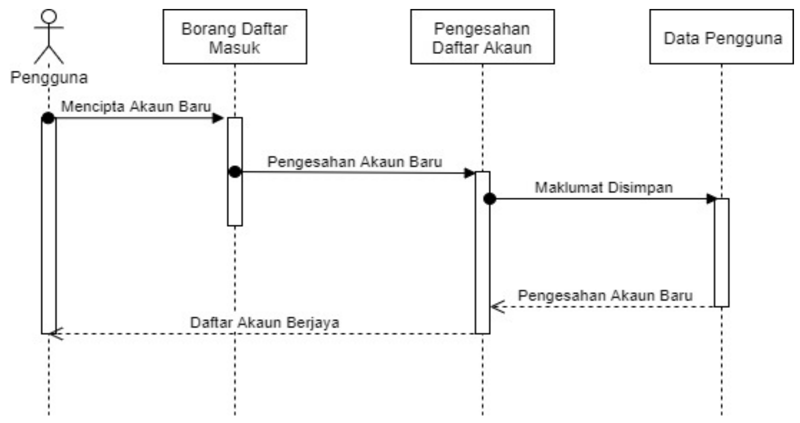


Figure 12: Sequence Diagramfor Register Module

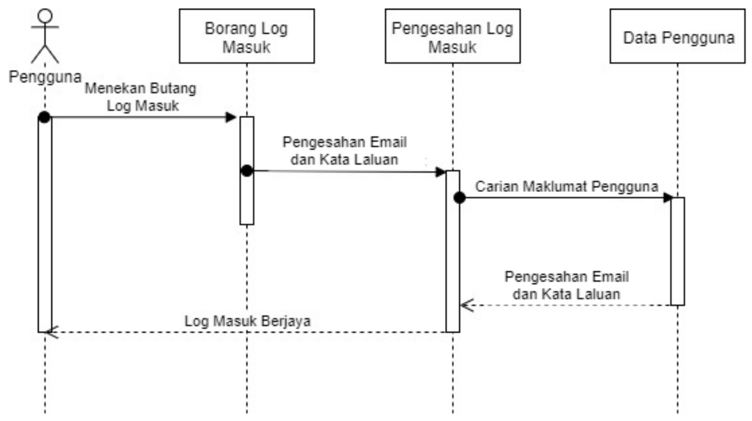


Figure 13: Sequence Diagramfor Login Module

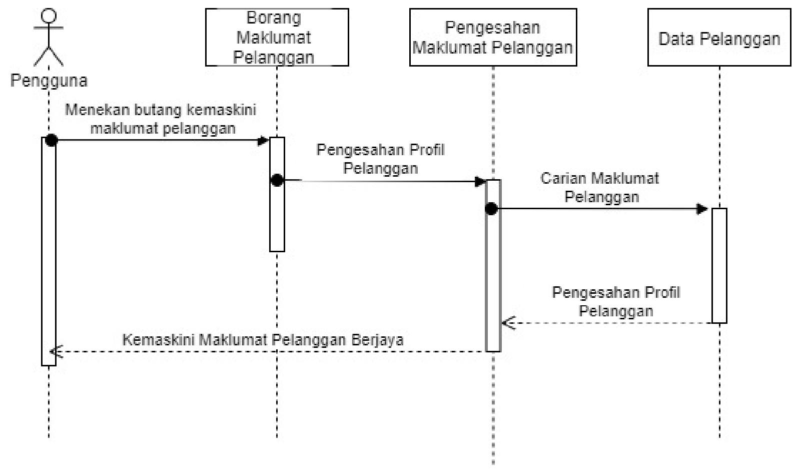

Pengguna
Borang Maklumat Pelanggan
Pengesahan Maklumat Pelanggan
Data Pelanggan
Menekan butang kemaskini maklumat pelanggan
Pengesahan Profil Pelanggan
Carian Maklumat Pelanggan
Pengesahan Profil Pelanggan
Kemaskini Maklumat Pelanggan Berjaya

Figure 14: Sequence Diagramfor Manage Customer Information Module

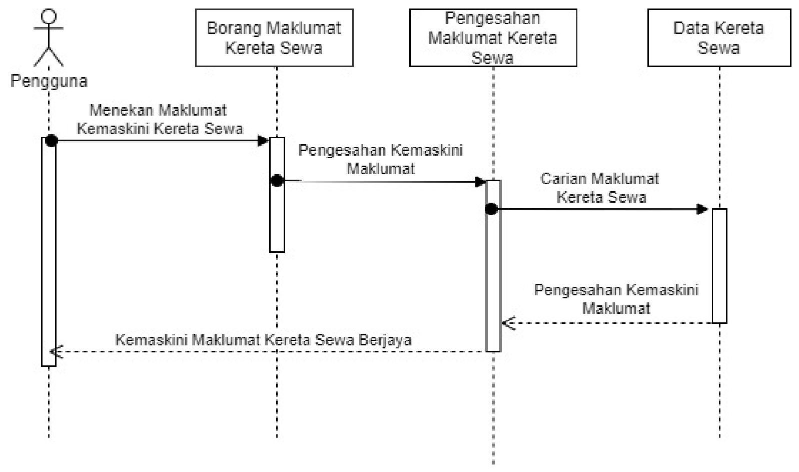

Pengguna
Borang Maklumat Kereta Sewa
Pengesahan Maklumat Kereta Sewa
Data Kereta Sewa
Menekan Maklumat Kemaskini Kereta Sewa
Pengesahan Kemaskini Maklumat
Carian Maklumat Kereta Sewa
Pengesahan Kemaskini Maklumat
Kemaskini Maklumat Kereta Sewa Berjaya

Figure 15: Sequence Diagram for Manage Rental Car Information Module

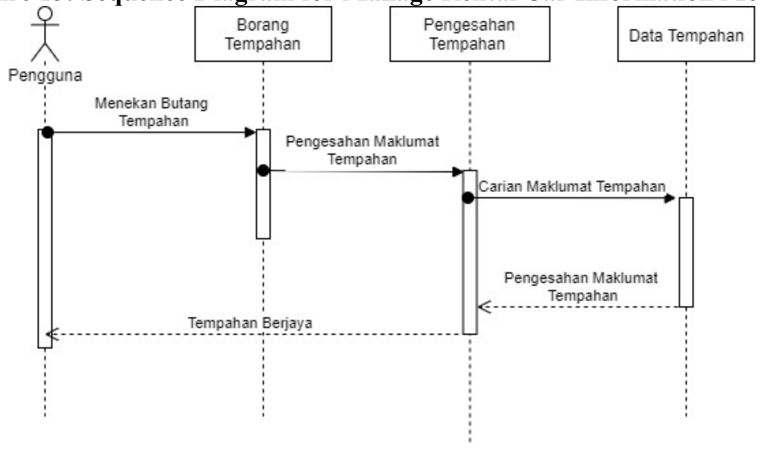


Figure 16: Sequence Diagram for Manage Booking Module

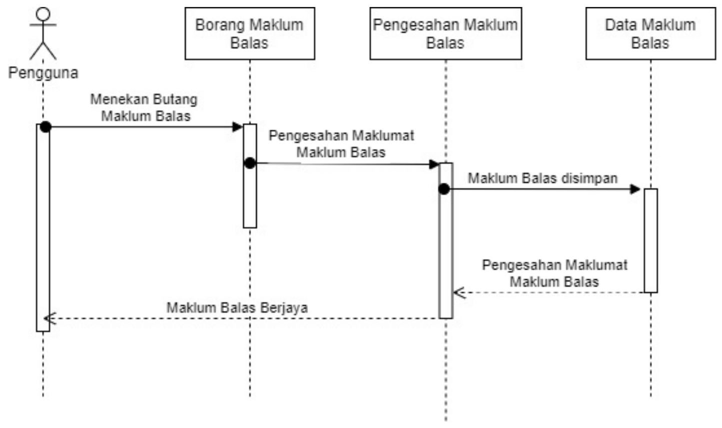


Figure 17: Sequence Diagram for Manage Feedback Module

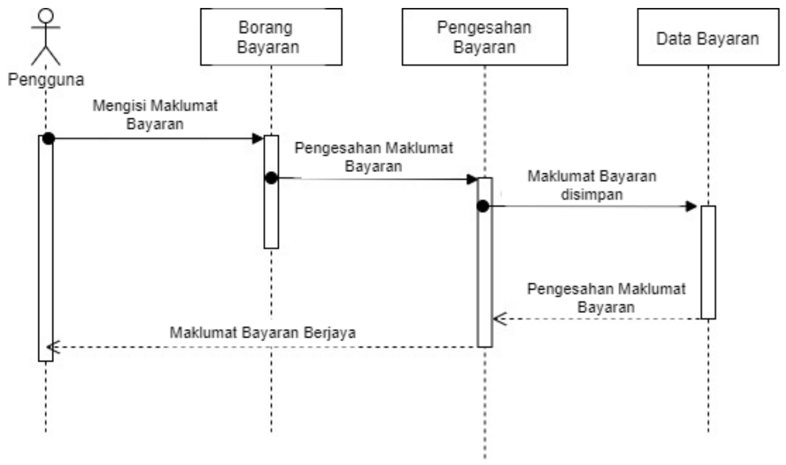


Figure 18: Sequence Diagram for Manage Payments Module

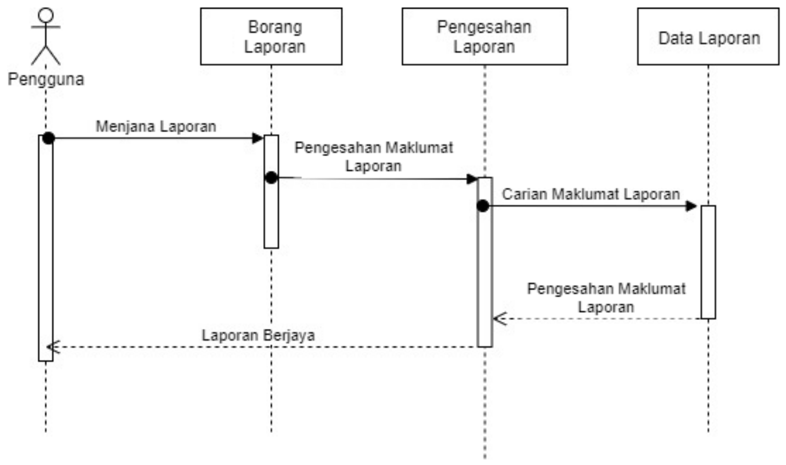


Figure 19: Sequence Diagram for Generate Reports Module

Appendix D: Rizqi Car Rental Booking System Based on Mobile Application Interface

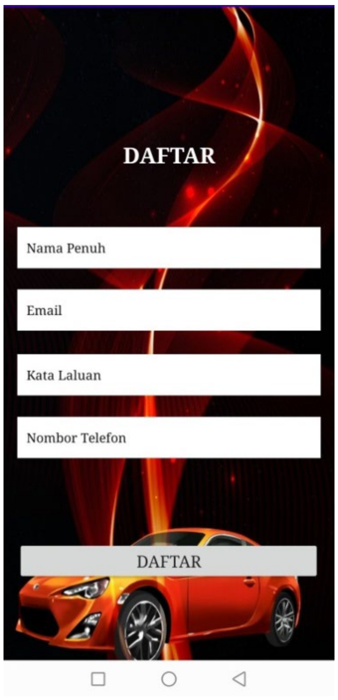


Figure 20: Interface for Register

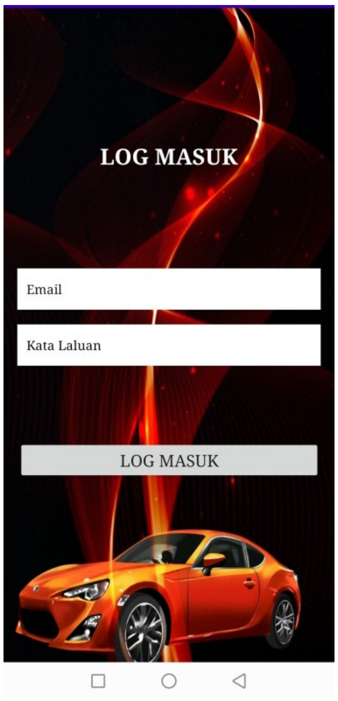
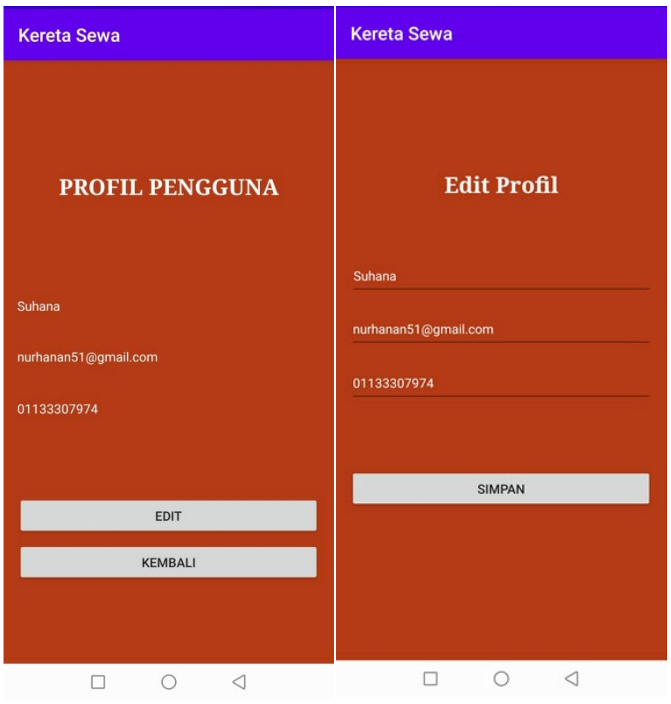


Figure 21: Interface for Login


Kereta Sewa

PROFIL PENGGUNA

Suhana

nurhanan51@gmail.com

01133307974

Kereta Sewa

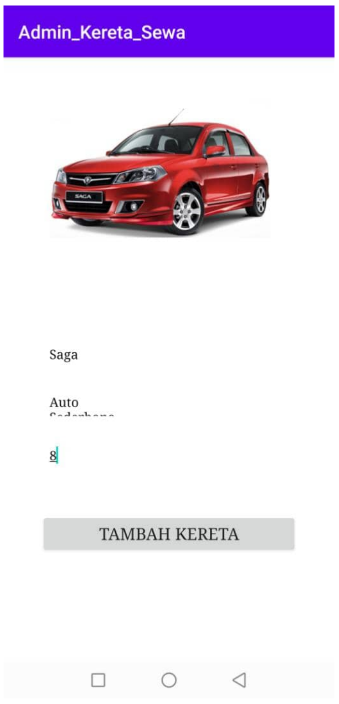
Edit Profil

Suhana

nurhanan51@gmail.com

01133307974

Figure 22: Interface for Manage Customer Information

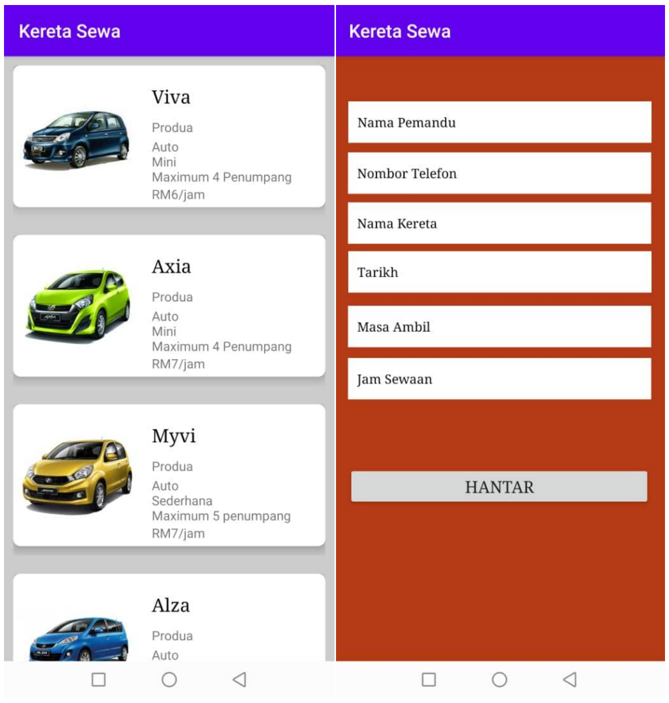

Admin_Kereta_Sewa

Saga

Auto

8

Figure 23: Interface for Manage Rental Car Information



Kereta Sewa

Viva
Produa
Auto
Mini
Maximum 4 Penumpang
RM6/jam

Axia
Produa
Auto
Mini
Maximum 4 Penumpang
RM7/jam

Myvi
Produa
Auto
Sederhana
Maximum 5 penumpang
RM7/jam

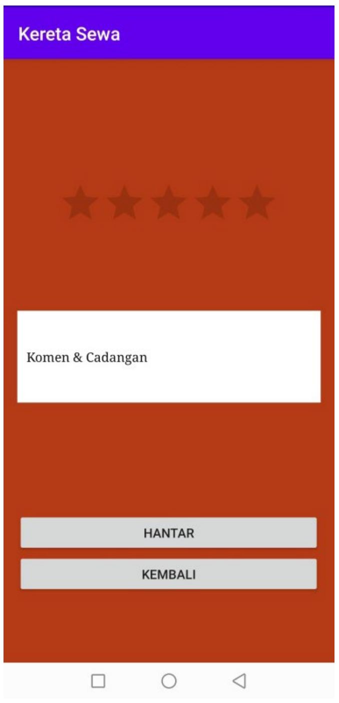
Alza
Produa
Auto

Kereta Sewa

Nama Pemandu
Nombor Telefon
Nama Kereta
Tarikh
Masa Ambil
Jam Sewaan

HANTAR

Figure 24: Interface for Manage Booking



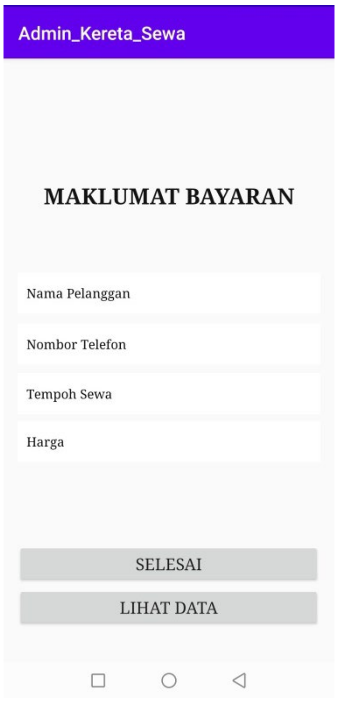
Kereta Sewa

Komen & Cadangan

HANTAR

KEMBALI

Figure 25: Interface for Manage Feedback


Admin_Kereta_Sewa

MAKLUMAT BAYARAN

Nama Pelanggan

Nombor Telefon

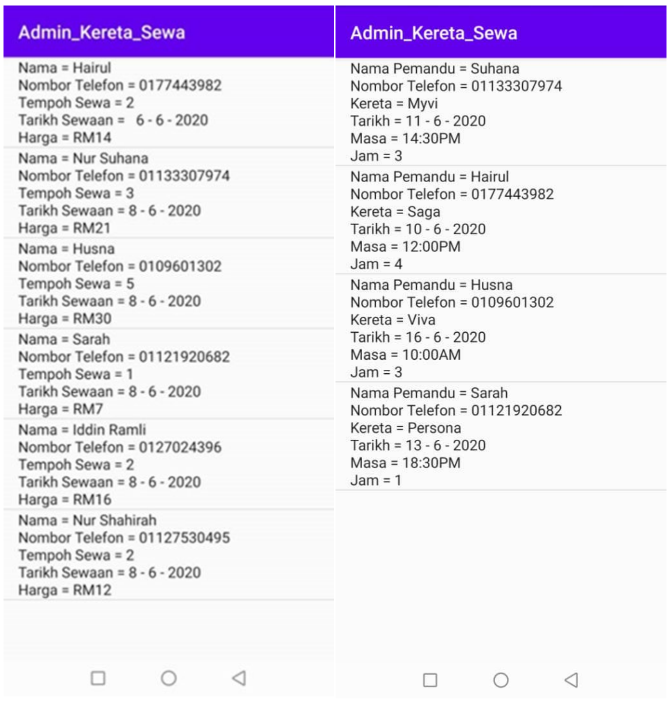
Tempoh Sewa

Harga

SELESAI

LIHAT DATA

Figure 26: Interface for Manage Payments


Admin_Kereta_Sewa

Nama = Hairul
Nombor Telefon = 0177443982
Tempoh Sewa = 2
Tarikh Sewaan = 6 - 6 - 2020
Harga = RM14

Nama = Nur Suhana
Nombor Telefon = 01133307974
Tempoh Sewa = 3
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM21

Nama = Husna
Nombor Telefon = 0109601302
Tempoh Sewa = 5
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM30

Nama = Sarah
Nombor Telefon = 01121920682
Tempoh Sewa = 1
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM7

Nama = Iddin Ramli
Nombor Telefon = 0127024396
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM16

Nama = Nur Shahirah
Nombor Telefon = 01127530495
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM12

Admin_Kereta_Sewa

Nama Pemandu = Suhana
Nombor Telefon = 01133307974
Kereta = Myvi
Tarikh = 11 - 6 - 2020
Masa = 14:30PM
Jam = 3

Nama Pemandu = Hairul
Nombor Telefon = 0177443982
Kereta = Saga
Tarikh = 10 - 6 - 2020
Masa = 12:00PM
Jam = 4

Nama Pemandu = Husna
Nombor Telefon = 0109601302
Kereta = Viva
Tarikh = 16 - 6 - 2020
Masa = 10:00AM
Jam = 3

Nama Pemandu = Sarah
Nombor Telefon = 01121920682
Kereta = Persona
Tarikh = 13 - 6 - 2020
Masa = 18:30PM
Jam = 1

Figure 27: Interface for Generate Report

Appendix E: Data Dictionary Table

Table 11: User Table for Register

Attribute	Data Type	Description
User_ID	String	User Id
Full Name	String	User full name
Email	String	User Email
Password	String	User Password
PhoneNo	String	User phone number

Table 12: User Table for Login

Attribute	Data Type	Description
User_ID	String	User id
Email	String	User email
Password	String	User password

Table 13: User Table for Manage Customer Information

Attribute	Data Type	Description
User_ID	String	User id
FullName	String	User full name
Email	String	User email
PhoneNo	String	User phone number

Table 14: User Table for Manage Rental Car Information

Attribute	Data Type	Description
Car_ID	String	Car rental id
Category	String	Car brand
Date	String	Date of rental car information added to system
Time	String	Time of rental car information added to system
CarName	String	Car rental name
CarDetails	String	Car rental detail
CarPrice	String	Car rental price for one hour

Table 15: User Table for Manage Booking Module

Attribute	Data Type	Description
DriverName	String	Driver full name
PhoneNo	String	Driver phone number
CarName	String	Car name
Date	String	Rental date
StartTime	String	Start time
EndTime	String	End time
RentalPeriod	String	Rental hours

Table 16: User Table for Manage Feedback Module

Attribute	Data Type	Description
User_ID	String	User id
Message	String	User feedback

Table 17: User Table for Manage Payments Module

Attribute	Data Type	Description
CustomerName	String	Customer name
PhoneNo	String	Customer phone number
RentalPeriod	String	Rental hours
RentalDate	Date	Date the customer made the payment
Price	String	Rental total price

Table 18: User Table for Generate Reports Module

Attribute	DataType	Description
CarRentalReport	String	Car rental report
PaymentReport	String	Payments report

Appendix F: Implementation

Register Module

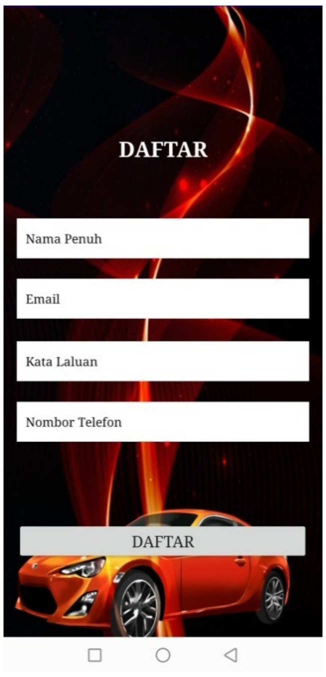
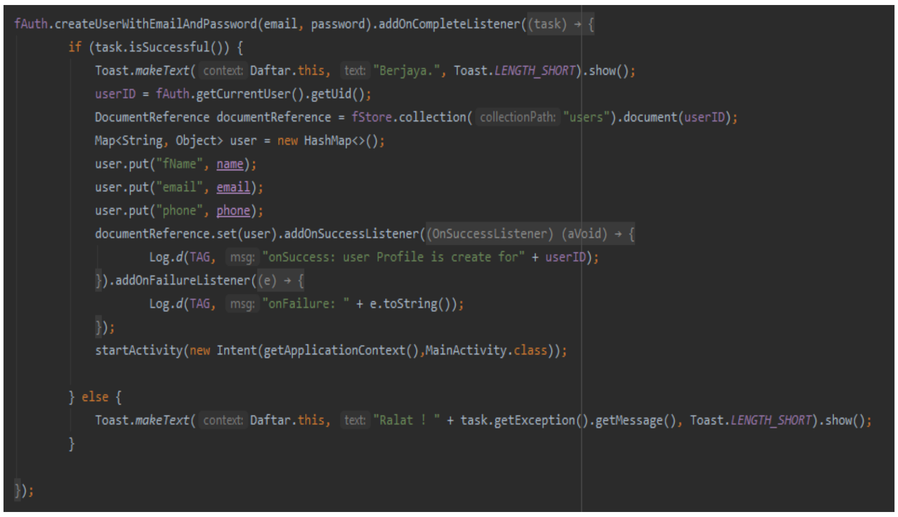


Figure 28: Interface for Register

Figure 28 shows the result of the register interface. Users need to fill in information such as full name, email, password, and phone number to log in to the system.



```
fAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) → {
        if (task.isSuccessful()) {
            Toast.makeText( context: Daftar.this,  text: "Berjaya.", Toast.LENGTH_SHORT).show();
            userID = fAuth.getCurrentUser().getUid();
            DocumentReference documentReference = fStore.collection( collectionPath: "users").document(userID);
            Map<String, Object> user = new HashMap<>();
            user.put("fName", name);
            user.put("email", email);
            user.put("phone", phone);
            documentReference.set(user).addOnSuccessListener((OnSuccessListener) (aVoid) → {
                    Log.d(TAG,  msg: "onSuccess: user Profile is create for" + userID);
            }).addOnFailureListener((e) → {
                    Log.d(TAG,  msg: "onFailure: " + e.toString());
            });
            startActivity(new Intent(getApplicationContext(),MainActivity.class));

        } else {
            Toast.makeText( context: Daftar.this,  text: "Ralat ! " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();
        }

});
```

Figure 29: Program Section for Register

Figure 29 shows theregister section program where the fName, email, and phone attributes will be recorded in the database. When the system successfully storage the information entered by the user. The system will display a "Successful" message.

Login Module

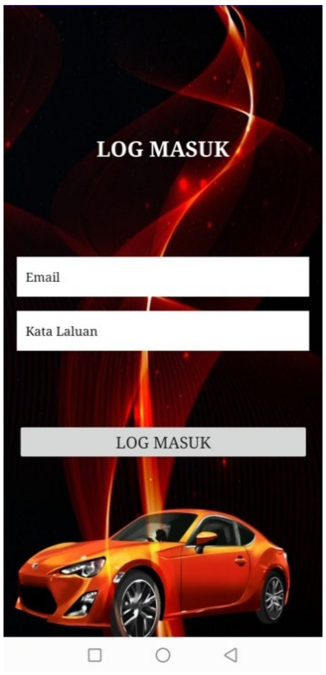
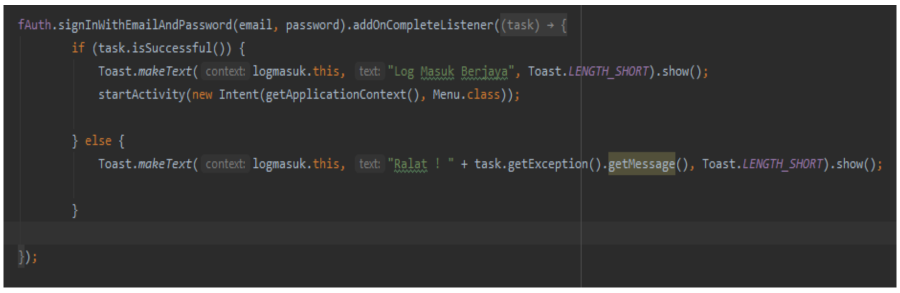


Figure 30: Interface for Login

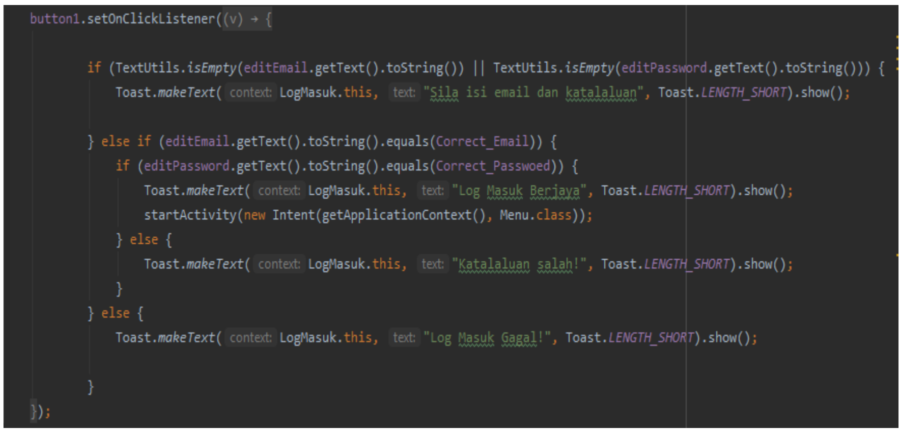
Figure 30 shows the result of the login interface. Users need to fill in information such as email and password to enter the system.



```
fAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener((task) → {
        if (task.isSuccessful()) {
            Toast.makeText( context: logmasuk.this,  text: "Log Masuk Berjaya", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(getApplicationContext(), Menu.class));

        } else {
            Toast.makeText( context: logmasuk.this,  text: "Ralat ! " + task.getException().getMessage(), Toast.LENGTH_SHORT).show();

        }

});
```

Figure 31: Program Section for Customer Login

Figure 31 shows the login section program for the customer where the signInWithEmailAndPassword method is used in this program. When a successful login, the system will display a "Successful Login" message.



```
button1.setOnClickListener((v) → {

        if (TextUtils.isEmpty(editEmail.getText().toString()) || TextUtils.isEmpty(editPassword.getText().toString())) {
            Toast.makeText( context: LogMasuk.this,  text: "Sila isi email dan katalaluan", Toast.LENGTH_SHORT).show();

        } else if (editEmail.getText().toString().equals(Correct_Email)) {
            if (editPassword.getText().toString().equals(Correct_Passwoed)) {
                Toast.makeText( context: LogMasuk.this,  text: "Log Masuk Berjaya", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getApplicationContext(), Menu.class));
            } else {
                Toast.makeText( context: LogMasuk.this,  text: "Katalaluan salah!", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText( context: LogMasuk.this,  text: "Log Masuk Gagal!", Toast.LENGTH_SHORT).show();

        }
});
```

Figure 32:Program Section for Admin Login

Figure 32 shows the login section program for the admin where emails and passwords are included in this program. For this admin login, staff only need to use the same email and password to log into the system. When a successful login, the system will display a "Successful Login" message.

Manage Customer Information Module

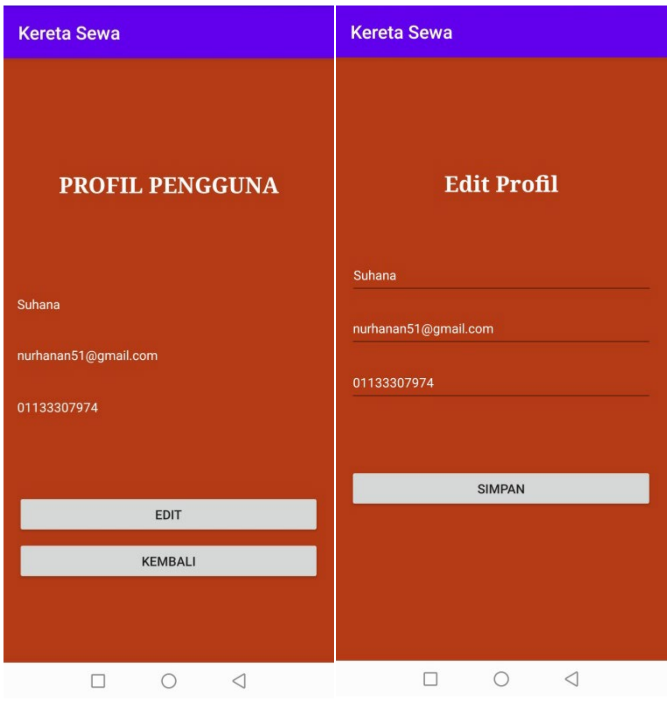
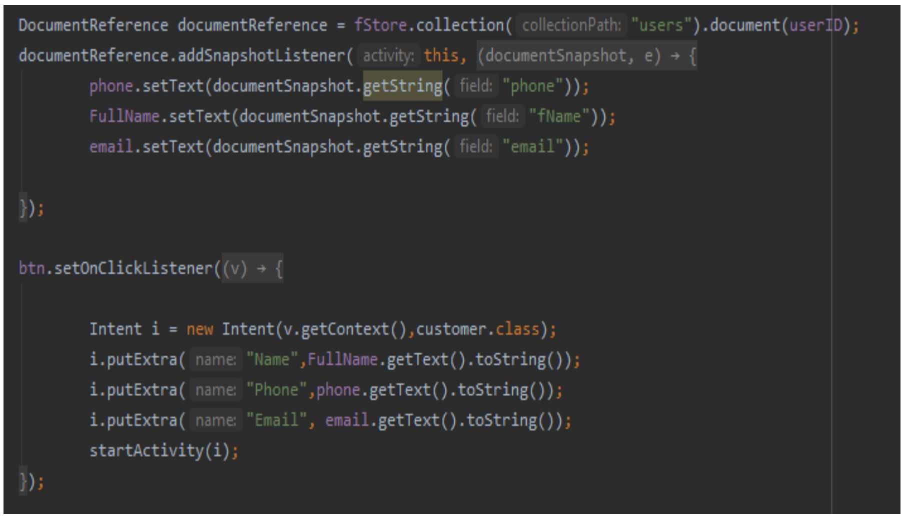


Figure 33: Interface for Manage Customer Information

Figure 33 is an interface to manage customer information. The manage customer information module has two parts, customers can view the personal information in the user profile and the customer can choose to update the information in the edit profile.



```
DocumentReference documentReference = fStore.collection( collectionPath: "users").document(userID);
documentReference.addSnapshotListener( activity: this, (documentSnapshot, e) → {
        phone.setText(documentSnapshot.getString( field: "phone"));
        FullName.setText(documentSnapshot.getString( field: "fName"));
        email.setText(documentSnapshot.getString( field: "email"));

});

btn.setOnClickListener((v) → {

        Intent i = new Intent(v.getContext(),customer.class);
        i.putExtra( name: "Name",FullName.getText().toString());
        i.putExtra( name: "Phone",phone.getText().toString());
        i.putExtra( name: "Email", email.getText().toString());
        startActivity(i);
});
```

Figure 34: Program Section for User Profile

Figure 34 is a user profile section program. The DocumentReference method is used to read the information contained in the system database. Information such as full name, email, and phone number will be displayed on the user profile.

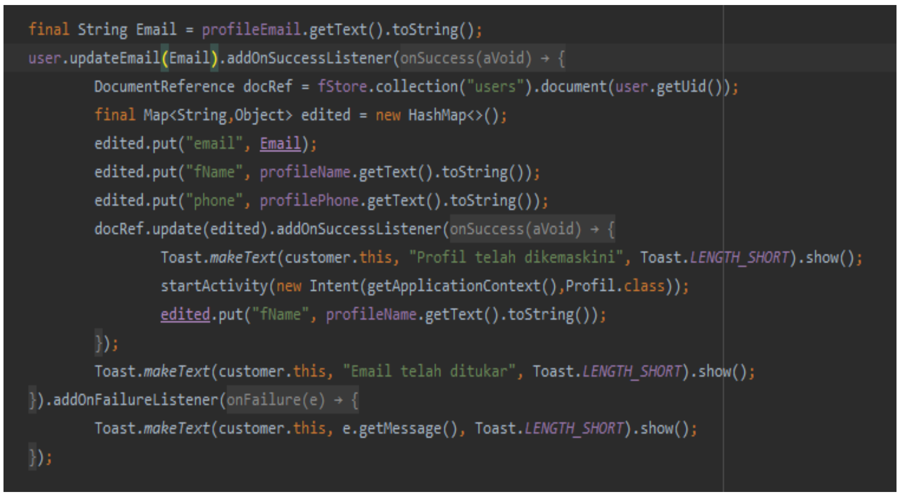

```
final String Email = profileEmail.getText().toString();
user.updateEmail(Email).addOnSuccessListener(onSuccess(aVoid) → {
        DocumentReference docRef = fStore.collection("users").document(user.getUid());
        final Map<String,Object> edited = new HashMap<>();
        edited.put("email", Email);
        edited.put("fName", profileName.getText().toString());
        edited.put("phone", profilePhone.getText().toString());
        docRef.update(edited).addOnSuccessListener(onSuccess(aVoid) → {
                Toast.makeText(customer.this, "Profil telah dikemaskini", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getApplicationContext(),Profil.class));
                edited.put("fName", profileName.getText().toString());
        });
        Toast.makeText(customer.this, "Email telah ditukar", Toast.LENGTH_SHORT).show();
}).addOnFailureListener(onFailure(e) → {
        Toast.makeText(customer.this, e.getMessage(), Toast.LENGTH_SHORT).show();
});
```

Figure 35: Program Section for Edit Profile

Figure 35 is a profile edit section program. The update Email (Email) and Map methods are used to update information and store information into the system.

Manage Rental Car Information Module

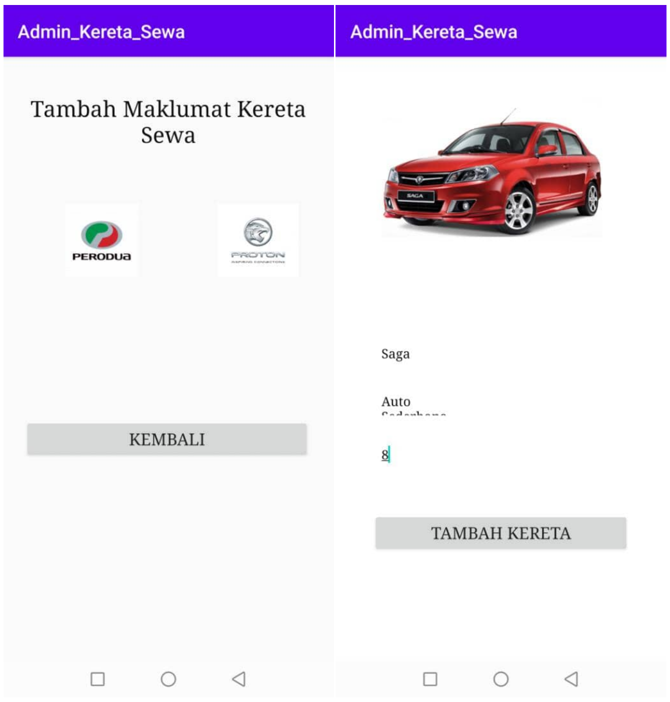
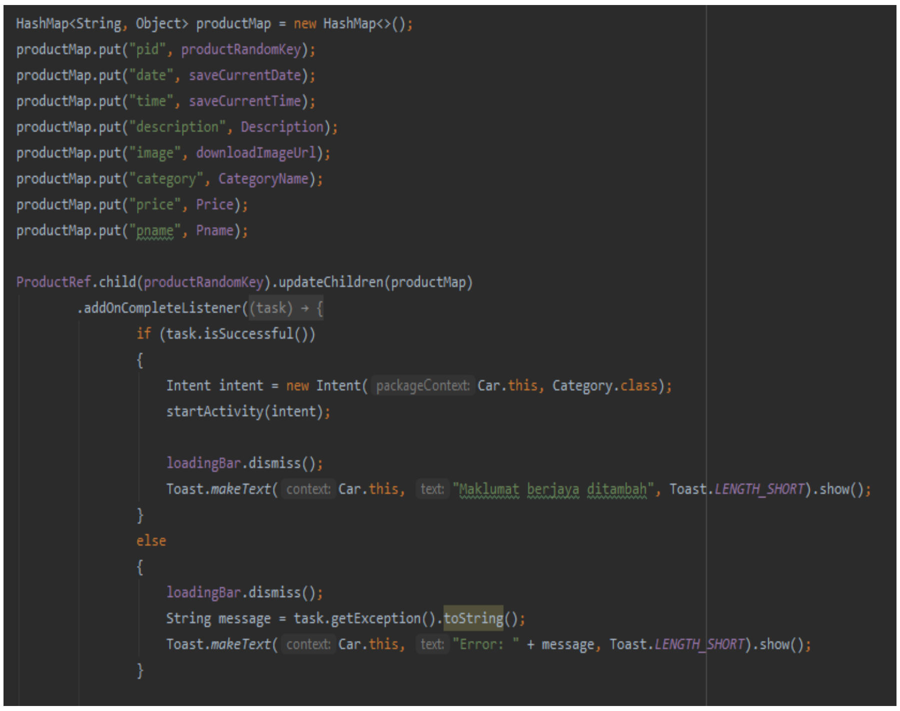


Figure 36: Interfacefor Manage Rental Car Information Module

Figure 36 is the result rental car information interface. To manage rental car information, staff need to select the car brand button to fill in the information on the rental car form. Staff needs to upload a photo of the car and fill in information such as car name, car details, and price for an hour of rental.

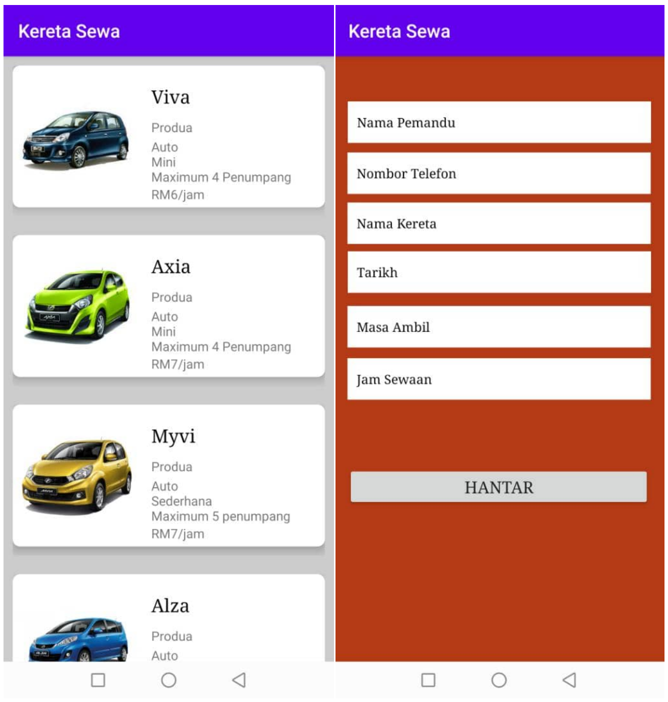


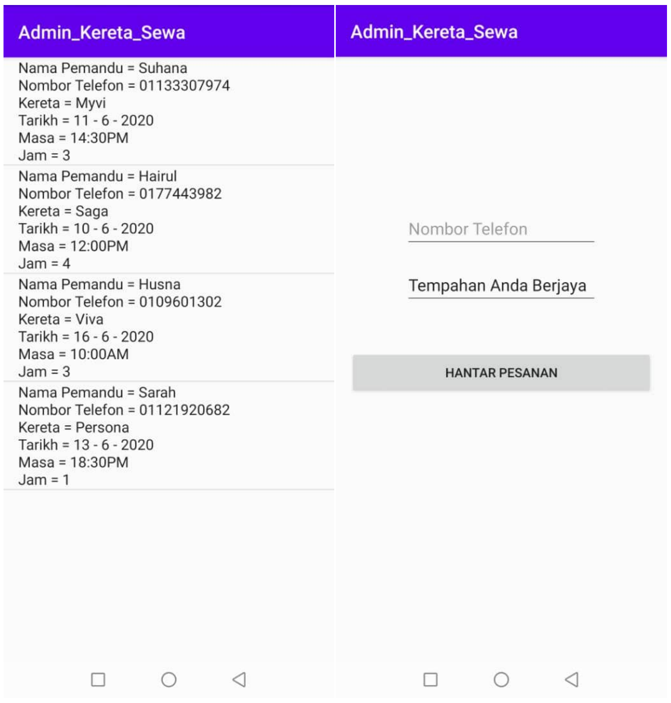
```
HashMap<String, Object> productMap = new HashMap<>();
productMap.put("pid", productRandomKey);
productMap.put("date", saveCurrentDate);
productMap.put("time", saveCurrentTime);
productMap.put("description", Description);
productMap.put("image", downloadImageUrl);
productMap.put("category", CategoryName);
productMap.put("price", Price);
productMap.put("pname", Pname);

ProductRef.child(productRandomKey).updateChildren(productMap)
        .addOnCompleteListener((task) → {
                if (task.isSuccessful())
                {
                    Intent intent = new Intent( packageContext: Car.this, Category.class);
                    startActivity(intent);

                    loadingBar.dismiss();
                    Toast.makeText( context: Car.this,  text: "Maklumat berjaya ditambah", Toast.LENGTH_SHORT).show();
                }
                else
                {
                    loadingBar.dismiss();
                    String message = task.getException().toString();
                    Toast.makeText( context: Car.this,  text: "Error: " + message, Toast.LENGTH_SHORT).show();
                }
```

Figure 37: Program Section for Manage Rental Car Information Module

Figure 37 is a program section of the rental car information form. The productMap method is used to storeinformation into the system. If the process of storing rental car information is successful, the system will display a message "Information successfully added".

Manage Booking Module





Admin_Kereta_Sewa

Nama Pemandu = Suhana
Nombor Telefon = 01133307974
Kereta = Myvi
Tarikh = 11 - 6 - 2020
Masa = 14:30PM
Jam = 3

Nama Pemandu = Hairul
Nombor Telefon = 0177443982
Kereta = Saga
Tarikh = 10 - 6 - 2020
Masa = 12:00PM
Jam = 4

Nama Pemandu = Husna
Nombor Telefon = 0109601302
Kereta = Viva
Tarikh = 16 - 6 - 2020
Masa = 10:00AM
Jam = 3

Nama Pemandu = Sarah
Nombor Telefon = 01121920682
Kereta = Persona
Tarikh = 13 - 6 - 2020
Masa = 18:30PM
Jam = 1

Admin_Kereta_Sewa

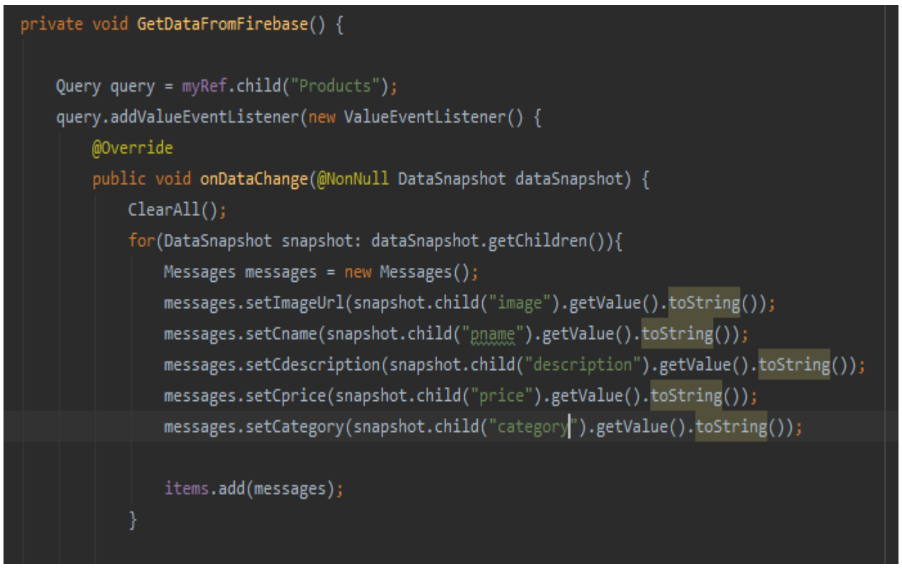
Nombor Telefon

Tempahan Anda Berjaya

HANTAR PESANAN

Figure 38: Interface for Manage Booking

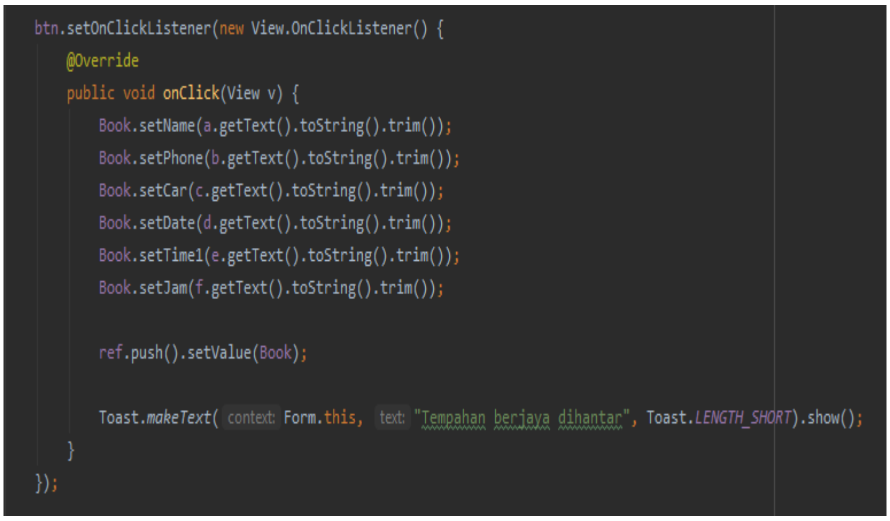
Figure 38 shows the result manage booking interface. Customers can view the list of rental cars available in the system and fill in the booking information on the booking form.



```
private void GetDataFromFirebase() {

    Query query = myRef.child("Products");
    query.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            ClearAll();
            for(DataSnapshot snapshot: dataSnapshot.getChildren()){
                Messages messages = new Messages();
                messages.setImageUrl(snapshot.child("image").getValue().toString());
                messages.setCname(snapshot.child("pname").getValue().toString());
                messages.setCdescription(snapshot.child("description").getValue().toString());
                messages.setCprice(snapshot.child("price").getValue().toString());
                messages.setCategory(snapshot.child("category").getValue().toString());

                items.add(messages);
            }
```

Figure 39: Program Section for List of Car Rental Information

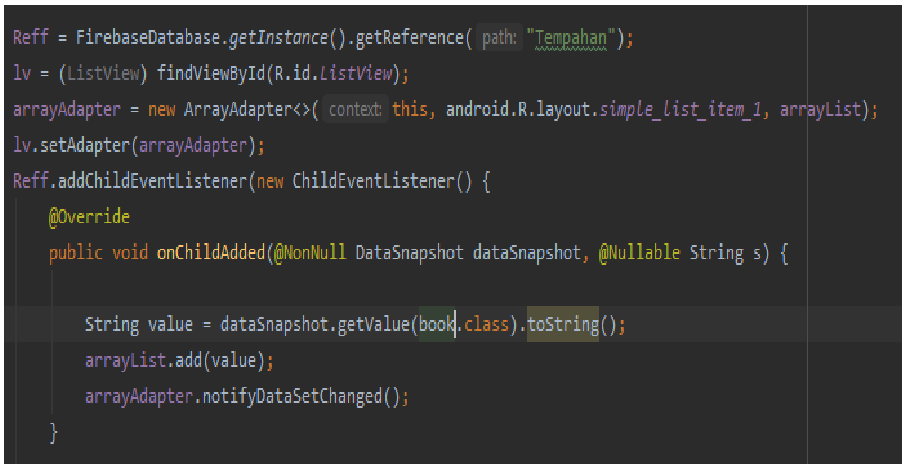
Figure 39 shows the program section for a list of rental car information. The addValueEventListener() method is used to read data from the product class. The dataSnapshot method is used because this method contains data in the Firebase database.



```
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Book.setName(a.getText().toString().trim());
        Book.setPhone(b.getText().toString().trim());
        Book.setCar(c.getText().toString().trim());
        Book.setDate(d.getText().toString().trim());
        Book.setTime1(e.getText().toString().trim());
        Book.setJam(f.getText().toString().trim());

        ref.push().setValue(Book);

        Toast.makeText( context: Form.this,  text: "Tempahan berjaya dihantar", Toast.LENGTH_SHORT).show();
    }
});
```

Figure 40: Program Sectionfor Booking Form

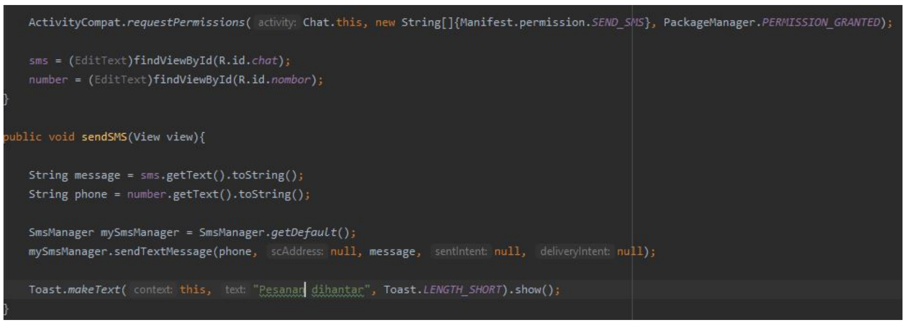
Figure 40 is a program section of the booking information form. The setName, setPhone, setCar, setDate, setTime, and setJam methods are used to record the data in the database. If the information storage process is successful, the system will display a message "Booking Successfully Sent".



```
Reff = FirebaseDatabase.getInstance().getReference( path: "Tempahan");
lv = (ListView) findViewById(R.id.ListView);
arrayAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, arrayList);
lv.setAdapter(arrayAdapter);
Reff.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        String value = dataSnapshot.getValue(book.class).toString();
        arrayList.add(value);
        arrayAdapter.notifyDataSetChanged();
    }
```

Figure 41: Program Section for Booking Information

Figure 41 is a program section of the booking information display. Data recorded in the booking class will be displayed. The arrayList method is used to store information in list form.

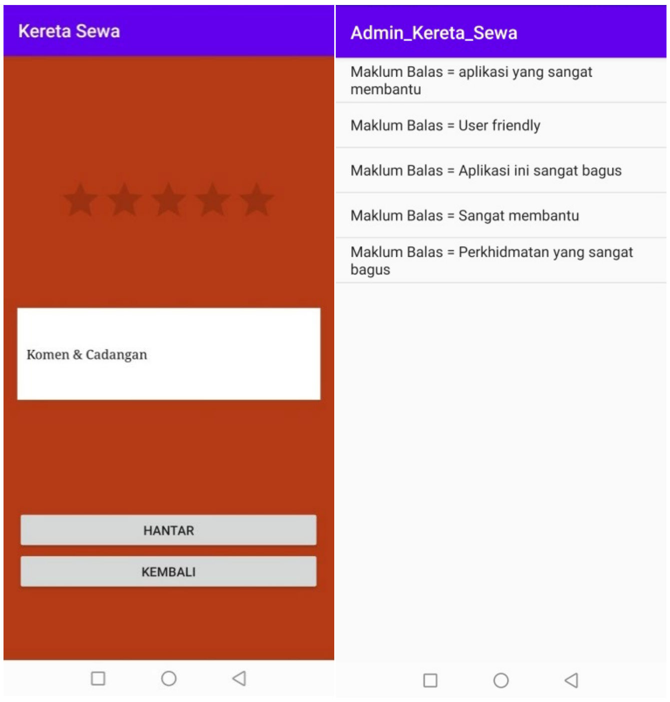


```
    ActivityCompat.requestPermissions( activity: Chat.this, new String[]{Manifest.permission.SEND_SMS}, PackageManager.PERMISSION_GRANTED);

    sms = (EditText)findViewById(R.id.chat);
    number = (EditText)findViewById(R.id.nombor);
}

public void sendSMS(View view){

    String message = sms.getText().toString();
    String phone = number.getText().toString();

    SmsManager mySmsManager = SmsManager.getDefault();
    mySmsManager.sendTextMessage(phone,  scAddress: null, message,  sentIntent: null,  deliveryIntent: null);

    Toast.makeText( context: this,  text: "Pesanan dihantar", Toast.LENGTH_SHORT).show();
}
```

Figure 42: Program Section for Booking Confirmation

Figure 42 shows a program section of the booking confirmation. Manifest.permission.SEND_SMS method is used to allow the system to send short messages. If a short message is successfully sent, the system will display a message "Message sent".

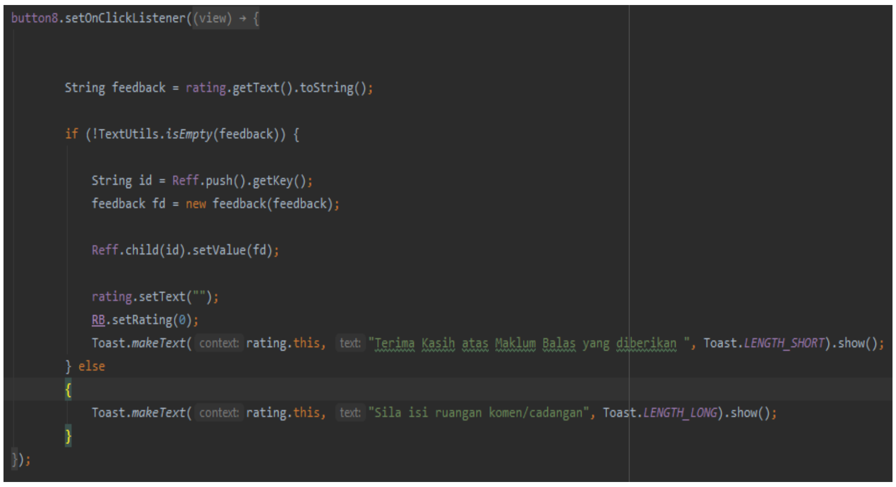
Manage Feedback Module



Maklum Balas = aplikasi yang sangat membantu
Maklum Balas = User friendly
Maklum Balas = Aplikasi ini sangat bagus
Maklum Balas = Sangat membantu
Maklum Balas = Perkhidmatan yang sangat bagus

Figure 43: Interface Manage Feedback

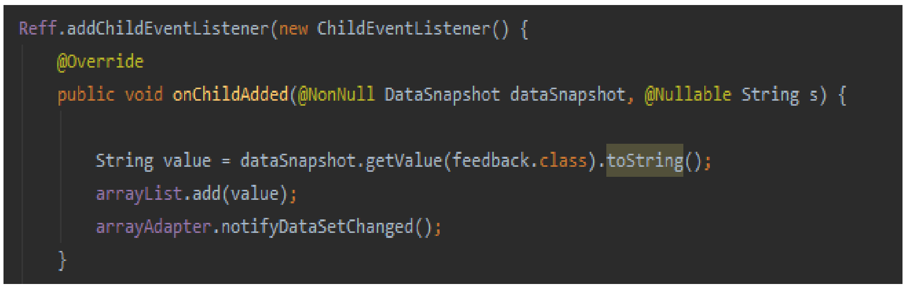
Figure 43 shows the result manage feedback interface. For feedback forms, customers can givestars and fill in the comments and suggestions field. For feedback display, staff can view feedback sent by customers through the system.



```
button8.setOnClickListener((view) → {


    String feedback = rating.getText().toString();

    if (!TextUtils.isEmpty(feedback)) {

        String id = Reff.push().getKey();
        feedback fd = new feedback(feedback);

        Reff.child(id).setValue(fd);

        rating.setText("");
        RB.setRating(0);
        Toast.makeText( context: rating.this,  text: "Terima Kasih atas Maklum Balas yang diberikan ", Toast.LENGTH_SHORT).show();
    } else
    {
        Toast.makeText( context: rating.this,  text: "Sila isi ruangan komen/cadangan", Toast.LENGTH_LONG).show();
    }
});
```

Figure 44: Program Sectionfor Feedback Form

Figure 44 is a program section of the feedback form. Feedback attributes will be recorded in the database. If the information storage process is successful, the system will display a message "Thank you for the feedback given".

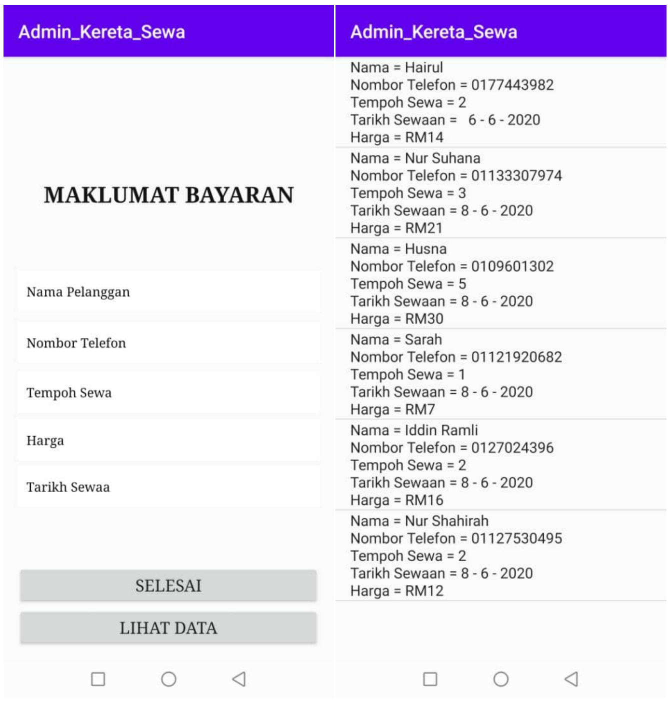


```
Reff.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        String value = dataSnapshot.getValue(feedback.class).toString();
        arrayList.add(value);
        arrayAdapter.notifyDataSetChanged();
    }
```

Figure 45: Program Section for Feedback Display

Figure 45 is a feedback display section program. The data recorded in the feedback class will be displayed. The arrayList method is used to store information in list form.

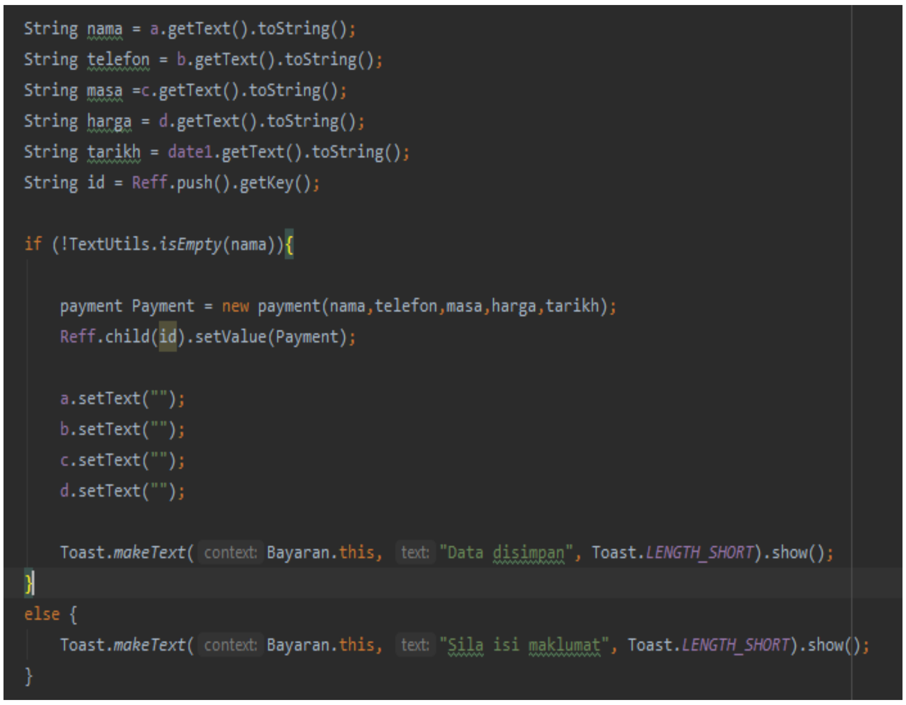
Manage Payments Module



```
Nama = Hairul
Nombor Telefon = 0177443982
Tempoh Sewa = 2
Tarikh Sewaan =  6 - 6 - 2020
Harga = RM14
Nama = Nur Suhana
Nombor Telefon = 01133307974
Tempoh Sewa = 3
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM21
Nama = Husna
Nombor Telefon = 0109601302
Tempoh Sewa = 5
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM30
Nama = Sarah
Nombor Telefon = 01121920682
Tempoh Sewa = 1
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM7
Nama = Iddin Ramli
Nombor Telefon = 0127024396
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM16
Nama = Nur Shahirah
Nombor Telefon = 01127530495
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM12
```

Figure 46: Interface for Manage Payments

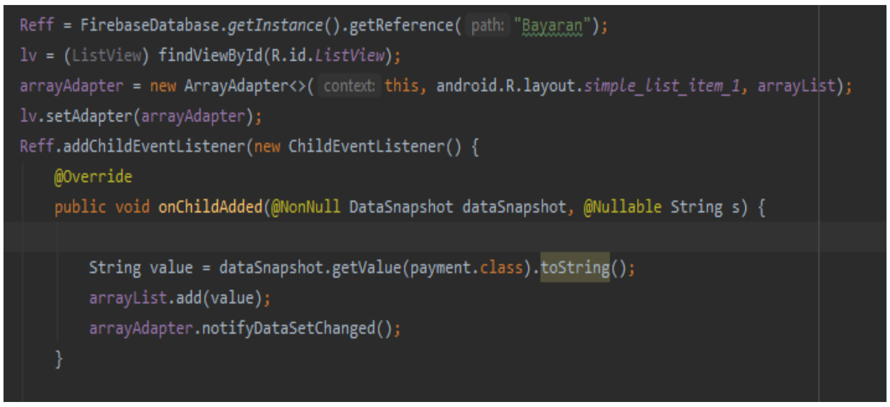
Figure 46 shows the resultinginterface. For payment information forms, staff can fill in the information of customers who have already made payment. For payment information display, staff can view the data that has been filled into the system.



```
String nama = a.getText().toString();
String telefon = b.getText().toString();
String masa =c.getText().toString();
String harga = d.getText().toString();
String tarikh = date1.getText().toString();
String id = Reff.push().getKey();

if (!TextUtils.isEmpty(nama)){

    payment Payment = new payment(nama,telefon,masa,harga,tarikh);
    Reff.child(id).setValue(Payment);

    a.setText("");
    b.setText("");
    c.setText("");
    d.setText("");

    Toast.makeText( context: Bayaran.this,  text: "Data disimpan", Toast.LENGTH_SHORT).show();
}
else {
    Toast.makeText( context: Bayaran.this,  text: "Sila isi maklumat", Toast.LENGTH_SHORT).show();
}
```

Figure 47: Program Section for Payment Form

Figure 47 is a payment form program section. Attributes of name, phone, time, price, and date will be recorded in the database. If the information storage process is successful, the system will display a message "Data stored".

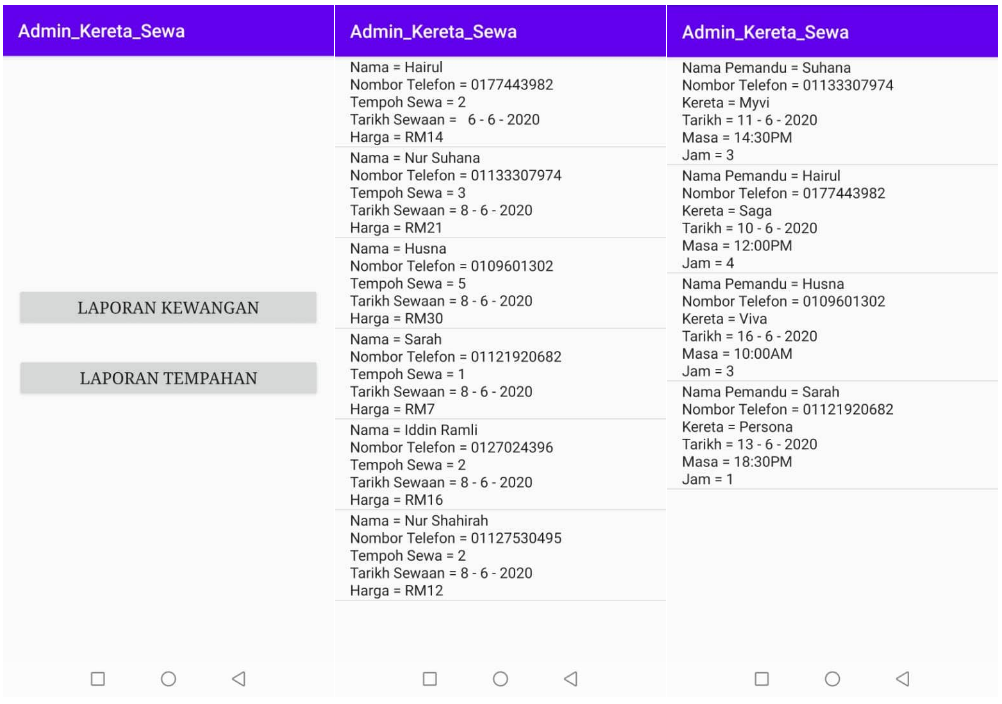


```
Reff = FirebaseDatabase.getInstance().getReference( path: "Bayaran");
lv = (ListView) findViewById(R.id.ListView);
arrayAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, arrayList);
lv.setAdapter(arrayAdapter);
Reff.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        String value = dataSnapshot.getValue(payment.class).toString();
        arrayList.add(value);
        arrayAdapter.notifyDataSetChanged();
    }
```

Figure 48: Program Section for Payment Information Display

Figure 48 is a program of payment information display section program. The data recorded in the payment class will be displayed. The arrayList method is used to store information in list form.

Generate Reports Module



Admin_Kereta_Sewa

Admin_Kereta_Sewa

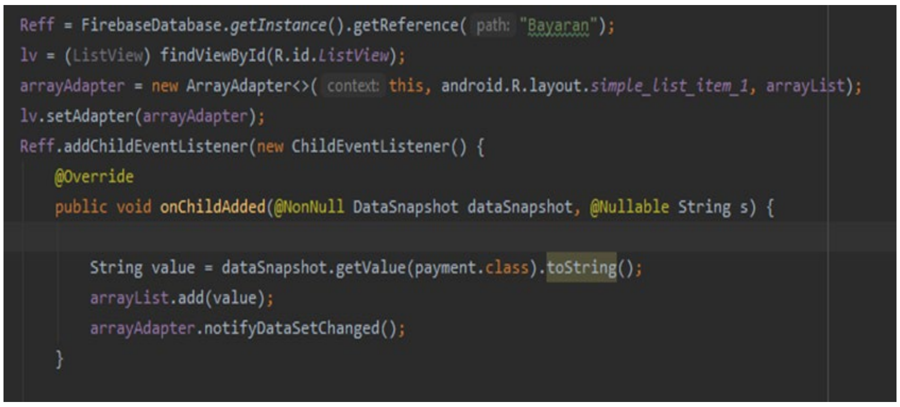
```
Nama = Hairul
Nombor Telefon = 0177443982
Tempoh Sewa = 2
Tarikh Sewaan =  6 - 6 - 2020
Harga = RM14
Nama = Nur Suhana
Nombor Telefon = 01133307974
Tempoh Sewa = 3
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM21
Nama = Husna
Nombor Telefon = 0109601302
Tempoh Sewa = 5
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM30
Nama = Sarah
Nombor Telefon = 01121920682
Tempoh Sewa = 1
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM7
Nama = Iddin Ramli
Nombor Telefon = 0127024396
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM16
Nama = Nur Shahirah
Nombor Telefon = 01127530495
Tempoh Sewa = 2
Tarikh Sewaan = 8 - 6 - 2020
Harga = RM12
```

Admin_Kereta_Sewa

```
Nama Pemandu = Suhana
Nombor Telefon = 01133307974
Kereta = Myvi
Tarikh = 11 - 6 - 2020
Masa = 14:30PM
Jam = 3
Nama Pemandu = Hairul
Nombor Telefon = 0177443982
Kereta = Saga
Tarikh = 10 - 6 - 2020
Masa = 12:00PM
Jam = 4
Nama Pemandu = Husna
Nombor Telefon = 0109601302
Kereta = Viva
Tarikh = 16 - 6 - 2020
Masa = 10:00AM
Jam = 3
Nama Pemandu = Sarah
Nombor Telefon = 01121920682
Kereta = Persona
Tarikh = 13 - 6 - 2020
Masa = 18:30PM
Jam = 1
```

Figure 49: Interface Generate Reports

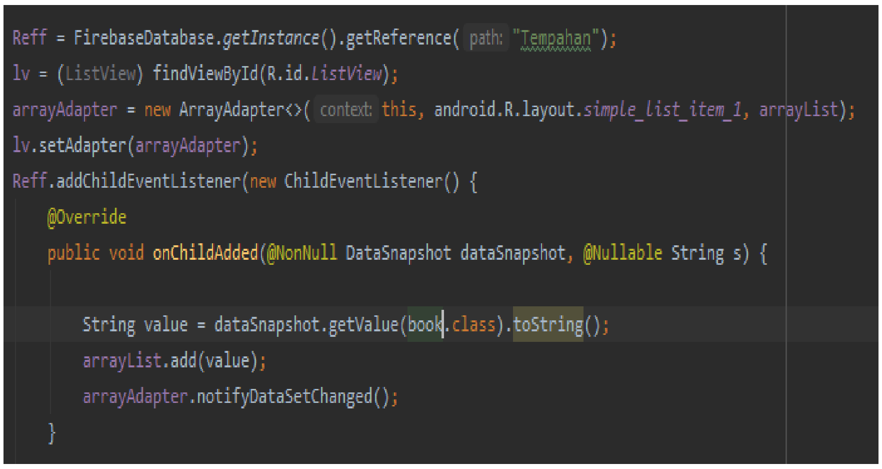
Figure 49 shows the resulting interface. Staff can select the report options button to view the report. Details of information such as customer name, phone number, and date can be seen in the report.



```
Reff = FirebaseDatabase.getInstance().getReference( path: "Bayaran");
lv = (ListView) findViewById(R.id.ListView);
arrayAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, arrayList);
lv.setAdapter(arrayAdapter);
Reff.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        String value = dataSnapshot.getValue(payment.class).toString();
        arrayList.add(value);
        arrayAdapter.notifyDataSetChanged();
    }
```

Figure 50: Program Section for Payment Report

Figure 50 is a payment report display section program. The data recorded in the payment class will be displayed. The arrayList method is used to store information in list form.



```
Reff = FirebaseDatabase.getInstance().getReference( path: "Tempahan");
lv = (ListView) findViewById(R.id.ListView);
arrayAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, arrayList);
lv.setAdapter(arrayAdapter);
Reff.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        String value = dataSnapshot.getValue(book.class).toString();
        arrayList.add(value);
        arrayAdapter.notifyDataSetChanged();
    }
```

Figure 51: Program Sectionfor Booking Report

Figure 51 is the program section of the booking report display. Data recorded in the booking class will be displayed. The arrayList method is used to store information in list form.

Appendix G: Testing

Table 19: Testing for Register

No	Testing Case	Expected Result	Actual Result
1.	Select the account registerationbutton.	The registration form is displayed	Registration form successfully displayed.
2.	Fill in the full name, email, password, and phone number.	User: Account registration is successful and the homepage is displayed.	Successfully registered a new account and displayed the homepage.
3.	The information entered is incomplete.	User: Account registration unsuccessful and a message will be displayed.	Failed to register a new account. Email and password are required.
4.	Select the register button	User: Register successful	Register successful

Table 19 shows the testing of the register module tested to the user through the system. The user has a register module to use this system and the information entered by the user will be stored in the database.

Table 20: Testing for Login

No	Testing Case	Expected Result	Actual Result
1.	Select the login button	The login form is displayed	Login form successfully displayed.
2.	Fill in email and password	User: Login successful and main menu displayed.	Successfully logged in and displayed the main menu.
3.	The information entered is incorrect.	User: Login unsuccessful and message will appear. Email and password need to be refilled.	Login unsuccessful. No accounts recorded. Email and password need to be refilled.

Table 20 shows the testing of the login module tested to the user through the system. Users have a login module to use this system using the email and password that have been registered in the system.

Table 21: Testing for Manage Customer Information

No	Testing Case	Expected Result	Actual Result
1.	Selects the update button on the customer information display.	A user profile is displayed.	Successful user profiles are displayed.
2.	Updating information	User: Information is successfully updated and user information is displayed.	Successfully updated information and displayed updated information.
3.	Not updating information.	User: Information is not updated and does not change.	Displays information that has not been updated.

Table 21 shows the testing manage customer informationmodule tested to the user through the system. Users who successfully register an account and login into the system can update the information in the user profile.

Table 22: Testing for Manage Rental Car Information

No	Testing Case	Expected Result	Actual Result
1.	Select the rental car information button	The rental car information form is displayed.	The rental car information form was successfully displayed.
2.	Add car rental information.	User: Car rental information successfully added.	Successfully added car rental information.
3.	Did not add car rental information.	User: Car rental information is not added.	Unable to add rental car information to the database.

Table 22 shows the testing manage rental car information module tested to users through the system. Users who successfully log into the admin system can add information.

Table 23: Testing for Manage Booking

No	Testing Case	Expected Result	Actual Result
1.	Select the booking button	A list of rental car is displayed	A list of rental cars was successfully displayed.
2.	Choosing a rental car	User: The rental car was successfully selected and the booking form is displayed.	Users chooses a car by entering the car name in the car rental information form.
3.	Do not choose a rental car	User: Booking form is not displayed.	Booking unsuccessful.
4.	Fill in the booking information.	User: Booking successfully sent and recorded.	Booking are sent and recorded in the admin system.
5.	Not fill in booking information.	User: Booking unsuccessfully sent and recorded.	Bookingis not sent and not recorded in the admin system.

Table 23 shows the testingmanage booking module tested to the user through the system. Users who successfully login to the system can book a rental car by selecting a rental car and fill in the booking form.

Table 24: Testing for Manage Feedback

No	Testing Case	Expected Result	Actual Result
1.	Select the feedback button	The feedback form is displayed.	Feedback form successfully displayed.
2.	Fill in the comments and suggestions	User: Feedback was successfully sent and recorded.	Successfully sent the feedback and successfully recorded it into the admin system.
3.	Do not fill in comments and suggestions	User: Feedback was not sent successfully and recorded.	Failed to send feedback and did not encode in the admin system.

Table 24 shows the testing manage feedback module tested to users through the system. Users who successfully loginto the system can provide feedback such as comments or suggestions on the system and services received.

Table 25: Testing for Manage Payments

No	Testing Case	Expected Result	Actual Result
1.	Select the payment button	The payment information form is displayed.	Payment information form successfully displayed.
2.	Fill in payment information	User: Payment information was successfully recorded and displayed.	Successfully recorded payment information and displayed a list of payment information.
3.	Does not fill in payment information	User: Payment information was not successfully added and recorded into the system.	Failed to add payment information and not recorded in the admin system.
4.	Select the view data button.	Payment information is displayed.	Payment information successfully displayed.

Table 25 shows the testing manage payments module tested to users through the system. Users who successfully log into the admin system can fill in the payment information and the information will be recorded in the admin system.

Table 26: Testing for Generate Reports

No	Testing Case	Expected Result	Actual Result
1.	Selectthe report button	A selection of report information is displayed.	Financial report and booking report button display.
2.	Generating reports	User: Report successfully generated.	Display of report information.
3.	Does not generating reports	User: Report not generated successfully.	No report information display.

Table 26 shows the testing generate reports tested to the user through the system. Users who successfully log into the admin system can view reports.