Car Rental Mobile Apps (WeGoo)

SWE / 143 / 23B

GERMAN-MALAYSIAN INSTITUTE

JULY / 2025

PROJECT TITLE **SWE / 143 / 23B**

I hereby declare that the project paper or thesis has been read and I have the opinion that the project paper is appropriate in terms of scope coverage and quality for awarding a Diploma in Software Engineering.

Signature :

Name of Supervisor : Ms. Ainin Sofiya Hisham

Date : 16 / 11 / 2025

WeGoo Application

SWE / 143 / 23B

Group Members
Puteri Nuur Binti Azhar
Muhammad Noor Danish Bin Noor Irwan
Adam Iskandar Bin Azwan Zamani
Ahmad Ammar Bin Abdul Rauzi

This project report is submitted as a partial fulfilment of the requirement for the **Diploma in Software Engineering.**

Project Supervisor
Ms. Ainin Sofiya Hisham

Semester 5
Computer Information Department
German-Malaysian Institute

July / 2025

We hereby declare that this project paper is authentic and group members are :-

Signature : 

Name of Group Members : Puteri Nuur Binti Azhar

Signature : 

Name of Group Members : Muhammad Noor Danish Bin Noor Irwan

Signature : 

Name of Group Members : Adam Iskandar Bin Azwan Zamani

Signature : 

Name of Group Members : Ahmad Ammar Bin Abdul Rauzi

Date : 16 / 11 / 2025

DEDICATION

We dedicate this report is intended for those people who have inspired and supported us during this journey. Your belief in our abilities and your support has always been inspiring not only for our personal and professional growth, as well as for our academic advancement. This final year report is dedicated to our family as a token of appreciation for their support throughout the writing process. Your all-time support and sacrifices have been contributed to our success. To friends and classmate, it is difficult to express enough for they have been the power in our existence and joy. We would like to thank you for your contribution to the final year project and for adding that team work and togetherness factor to an already wonderful programme. To our supervisor and TTO, words cannot express how much we are grateful for what you have taught me and your support. Your sheer commitment in motivating us or provoking us to develop reasoning abilities and encouraging beyond capabilities definitely have benefited our general group development. Last but not the least, the authors are grateful to all the participants who were involved in the preparation of this final year report and all the resources used. Your input of information, ideas and Material have been useful in completing this project and I thank you for your contribution.

ACKNOWLEDGEMENT

We would like to express our deep indebtedness to all the people that helped to bring into being this final year project titled WeGoo. In the first and foremost we would like to be grateful to our Supervisor Ms. Ainin Sofiya Hisham for her support, her knowledge and motivation which she has provided us to do the project. To the friends and classmates, we would like to extend our sincere thanks for the comments they have offered for the construction of this work and negative criticisms to aspiring novels. For us, the group signifies the sum of each part, thus we would like to thank everyone and their contribution. Through the passion that they hold for their work and the different backgrounds that the members possess, they was able to present a new perceptive to the project which is beneficial to the productivity of the task. Finally, in the last note, the authors care to express their appreciation to their families and friends for being helpful in the work. These have in particular involved beliefs or capabilities, incentives in difficult context, and some words to spur in all those doing. Allow us one final time to thank all of you for your support and for your confidence in our team. We would like to begin by extending our appreciation, due to your assistance, it was impossible for us to embark on this project sans you.

ABSTRACT

The main purpose of this project is to develop an efficient car rental application that allows users to easily book, compare, and manage vehicles through a single digital platform. During development, we identified several issues, including the lack of centralized systems among rental providers, limited comparison and secure payment features in existing apps, and the need for a more transparent method to verify vehicle conditions. To address these challenges, our objectives focus on creating a complete rental system, providing secure booking and comparison features, and integrating OpenCV technology to detect scratches on vehicles. The application is developed using Android Studio, and we adopt the Scrum model for both development and testing, allowing continuous iterations with unit testing, integration testing, and system testing. Our feasibility study includes primary sources such as surveys and interviews with rental providers and users, as well as secondary sources involving research on existing car rental platforms and related articles. We also assessed the required hardware and software, confirming that the project can be completed within the planned budget. With clear UML diagrams and structured design, creating the mockups became more organized and effective. In conclusion, the WeGoo application aims to provide a modern, transparent, and user-friendly solution that improves the overall car rental process for both users and service providers.

TABLE OF CONTENT

Chapter	Content	Page Number
Chapter 1	Introduction	1
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Objectives	3
	1.4 Project Scope	4
	1.4.1 Types of Systems	4
	1.4.2 Targeted Users	4
	1.4.3 Features	5
	1.4.4 Testing Elements	5
Chapter 2	Literature Review	6
	2.1 Feasibility Studies	6
	2.1.1 Surveys	6
	2.2 Primary Sources	14

2.2.1 Researching Related Application 14

2.3 Secondary Sources 24

2.3.1 Observation 24

2.4 Product Comparison 27

2.4.1 Comparison Table 27

Chapter 3 Project Methodology 29

3.1 Software Requirement 29

3.1.1 SDLC Process 29

3.1.2 Use Case Diagram 32

3.1.3 Use Case Specification 33

3.1.4 Sequence Diagram 62

3.1.5 Activity Diagram 68

3.1.6 Class Diagram 74

3.2 Project Specification 75

3.2.1 Software Specification 75

3.2.2 Hardware Specification 77

	3.2.3 Database Specification	81
	3.2.4 Design Specification	82
	3.3 Project Features	83
	3.4 Project Costing	85
Chapter 4	Testing Methodology	86
	4.1 Testing Model	86
	4.1.1 Scrum Model	86
	4.1.2 Feature To Be Tested	88
	4.2 Testing Tools	89
	4.2.1 Application Use For Testing	89
	4.2.2 Testing Type	94
Chapter 5	Finding And Analysis	96
	5.1 Project Mockup	96
	5.1.1 User GUI System Proposed	96
	5.1.2 Admin GUI System Proposed	104
	5.2 Problem Encounter And Countermeasure	111
	5.3 Result And Discussion	112

Chapter 6 Conclusion 113

6.1 Conclusion 113

Reference 115

LIST OF TABLES

Tables	Description	Page Number
2.4.1.1	Product Comparison Table	27
3.1.3.1	Login Use Case Description	33
3.1.3.2	Rental Transport Use Case Description	42
3.1.3.3	Redirect to Online Banking Use Case Description	45
3.1.3.4	Comparison Tools Use Case Description	48
3.1.3.5	OpenCV Scratch Detector Use Case Description	50
3.1.3.6	Update App Management Use Case Description	52
3.1.3.7	Manage Application Use Case Description	54
3.1.3.8	Manage Data User And Admin Use Case Description	56
3.2.1.1	Software Specifications	74
3.2.2.1	Hardware Specifications	76
3.2.3.1	Database Specifications	80
3.2.4.1	Design Specifications	81

3.4.1	Project Costing	84
4.2.1.1	Application Use For Testing	88
5.2.1	Problem Encounter And Countermeasure	110

LIST OF FIGURES

Figures	Description	Page Number
2.1.1.1.1	Pie Chart WeGoo Survey	6
2.1.1.1.2	Bar Chart WeGoo Survey	7
2.1.1.1.3	Bar Chart WeGoo Survey	7
2.1.1.1.4	Bar Chart WeGoo Survey	8
2.1.1.1.5	Pie Chart WeGoo Survey	9
2.1.1.1.6	Bar Chart WeGoo Survey	9
2.1.1.1.7	Suggestion Section WeGoo Survey	10
2.1.1.1.8	Suggestion Section WeGoo Survey	11
2.1.1.1.9	Comment Section WeGoo Survey	12
2.1.1.1.10	Comment Section WeGoo Survey	12
2.1.1.1.11	Comment Section WeGoo Survey	13
2.2.1.1	Homepage SOCAR	14
2.1.2.2	SOCAR Option	15

2.1.2.3	SOCAR Booking Details	15
2.1.2.4	SOCAR Booking History	15
2.2.1.2	SOCAR Booking Page	15
2.1.2.6	SOCAR My Promos Page / SOCAR Redeem Page	16
2.2.1.5	TREVO Homepage	18
2.1.2.9	TREVO Option Page	21
2.1.2.10	TREVO Booking Details	21
2.1.2.11	TREVO Booking History	21
2.2.1.7	TREVO Host page	21
2.2.1.8	TREVO Users Profile	22
2.3.1.1	Articles from BytePlus	24
2.3.1.2	Articles from BytePlus	25
3.1.3.9	Update app management	54
3.1.5.1	Login Activity Diagrams	67
3.1.5.2	WeRent Activity Diagrams	68
3.1.5.3	OpenCV Activity Diagrams	69
3.1.5.4	Admin Update Application Management Activity Diagrams	70

3.1.5.5	Admin Manage User Activity Diagrams	71
3.1.5.6	Invalid Password Activity Diagrams	72
3.1.6.1	WeGoo Class Diagram	73
3.2.1.1	Google Maps API	74
3.2.1.2	OpenCV	74
3.2.1.3	Visual Studio Code (VS Code)	74
3.2.1.4	Android Studio	75
3.2.2.1	Iphone 13	75
3.2.2.2	MSI 27 Inch LED Flat Pro 4K	76
3.2.2.3	Mac Mini i7	77
3.2.2.4	Fantech (SP42) X42	77
3.2.2.5	Redragon K552 Kumara	78
3.2.2.6	Jbl Portable Speaker	78
3.2.2.7	Monitor Arm	79
3.2.3.1	Firebase	80
3.2.4.1	Figma	81
3.2.4.2	Capcut	81

5.1.1.2	User Sign Up Page	97
5.1.1.3	User Forgot Password Page	98
5.1.1.4	User Homepage	99
5.1.1.5	User Search Page	100
5.1.1.6	User Booking Detail	101
5.1.1.7	User Profile Page	102
5.1.1.8	User Edit Profile Page	103
5.1.2.1	Admin Login Page	104
5.1.2.2	Admin Homepage	104
5.1.2.3	Add and Edit Vehicle Page Admin	105
5.1.2.4	Sign Up Page for Admin and Users	106

CHAPTER I

INTRODUCTION

1.1 PROJECT BACKGROUND

Over time, we've seen remarkable transformations in how we live, from the way we dress to how we move from place to place. What once required walking long distances can now be done with just a few taps on a screen. Renting cars has become an increasingly popular and convenient option, allowing people to travel freely without the long-term commitment of ownership. However, many still face challenges in finding reliable rental services and selecting the right car that suit their needs, lifestyle, and budget.

Introducing **WeGoo**, the all-in-one app designed to revolutionize how you handle everything related to car rentals. Whether you're planning to rent or explore, WeGoo combines innovation and simplicity to make every process effortless. Designed with a user-friendly interface, It also enables smooth vehicle management and profile customization, ensuring a convenient and personalized experience for all types of users. The latest version of WeGoo now features **OpenCV** technology, which brings intelligent image detection and recognition capabilities to enhance user interaction and functionality.

In addition, WeGoo connects users with trusted rental services, guaranteeing reliability and transparency in every transaction. Through its user-friendly interface, users can easily search, compare, and book car that meet their needs in just a few taps. With WeGoo, car renting and planning become smarter, faster, and more accessible than ever before, bridging the gap between technology and transportation to make mobility not only convenient but also intelligent and efficient for everyone.

1.2 PROBLEM STATEMENT

- **Lack of proper digitized mobile application**: Existing car rental system are outdated and not fully user-friendly, making it hard for users to browse, book, and manage rentals smoothly.

- **Existing system may not offer reliable booking with integrated payment features.** : Many current platforms lack a stable booking system and secure in-app payment options, causing users to face uncertainties during reservations and transactions.

- **Several apps ignore the use of advanced technologies like OpenCV to detect vehicle damages accurately. :** Most car rental apps do not integrate modern image-processing technologies, resulting in less accurate or unreliable detection of vehicle scratches and damages.

1.3 OBJECTIVE

- **To develop a car booking and management mobile application called WeGoo. :** To focuses on creating a user-friendly app that allows users to easily book and manage car rentals.

- **To provide reliable booking, comparison, and payment features.** : To ensures users can safely compare car options, make bookings, and complete secure payments within the app.

- **To test the functionality of the application and the OpenCV scratch detection feature** : To ensure OpenCV detection features inside the WeGoo application functioning well and successfully running without any critical error.

1.4 PROJECT SCOPE

The goal of developing the WeGoo application is to provide a complete and efficient digital platform where users can easily book, compare, and manage car rentals. This system aims to support both customers and rental providers by offering a centralized space for vehicle listings, secure bookings, and clear rental processes. Through the integration of OpenCV scratch detection, WeGoo ensures transparency in vehicle condition reporting while helping reduce disputes between users and providers. This approach strengthens trust, improves service quality, and enhances the overall rental experience, all within a simple and user-friendly mobile environment.

1.4.1 TYPES OF SYSTEMS

WeGoo is an Android based application that was developed from the ground up using present day programming tools and languages specifically designed for android operating systems.

1.4.2 TARGETED USERS

The WeGoo application is designed to cater to a wide range of users, including fresh graduate students who need affordable and flexible transportation options, large families who require convenient booking and vehicle comparison features, and work or travel agencies that manage multiple rental arrangements for their clients. Additionally, WeGoo supports car rental service providers by offering a centralized digital platform where they can manage vehicle listings, bookings, and customer interactions more efficiently.

1.4.3 FEATURES

- Login / Sign up system
- OpenCv
- Update/Manage Vehicle
- History
- View Customer
- Profile Setting
- Booking
- Compare Table
- Payment Gateway
- Contact Provider

1.4.4 TESTING ELEMENTS

WeGoo will use types of testing to test its functionality:

- Functionality Testing
- Portability Testing

CHAPTER II

LITERATURE REVIEW

2.1 FEASIBILITY STUDIES

2.1.1 SURVEYS

A primary source is a source that takes information directly from surveys, public suggestions, and research, providing original and firsthand data on the topic.

2.1.1.1 SURVEY

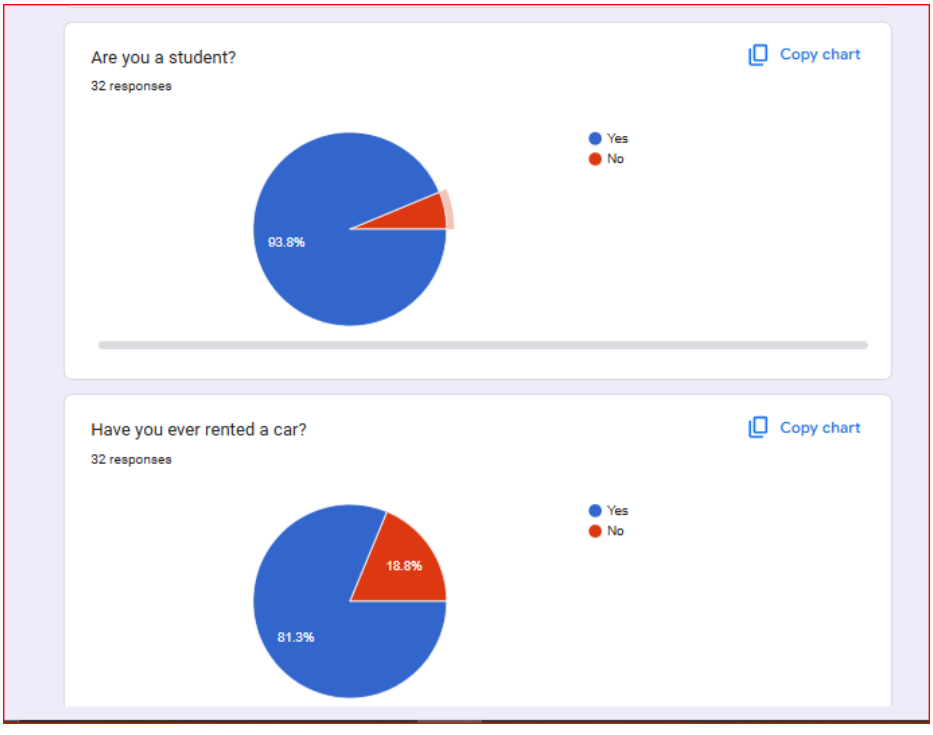


Figure 2.1.1.1.1 : Pie Chart WeGoo Survey

This survey aims to target individuals who have rented a car.

For this survey, we targeted 30 respondents consisting of fresh graduates, big families, and the general public. In this pie chart, we can see that 93.8% of the respondents are students, while 6.2% are from families and the general public. Additionally, another pie chart shows that 81.3% of the respondents have rented a car, while 18.8% have not.

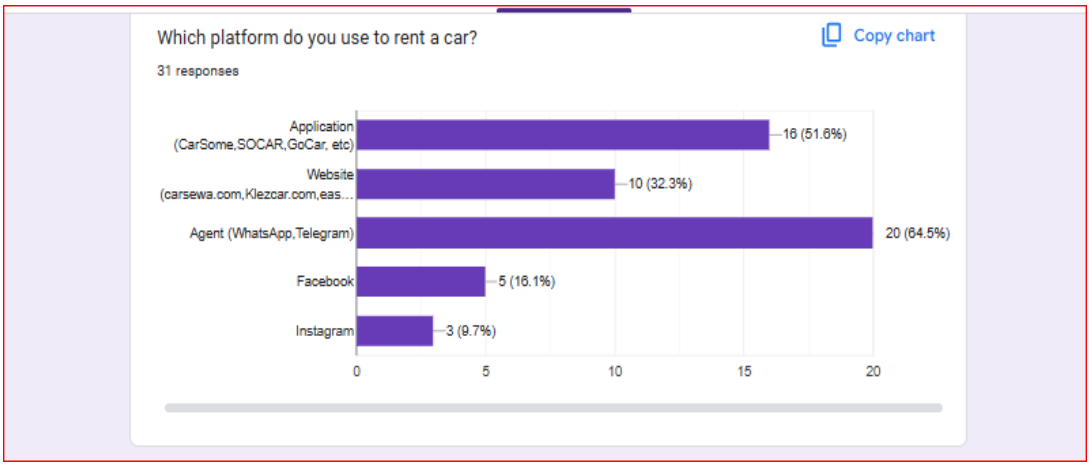


Figure 2.1.1.1.2 : Bar Chart WeGoo Survey

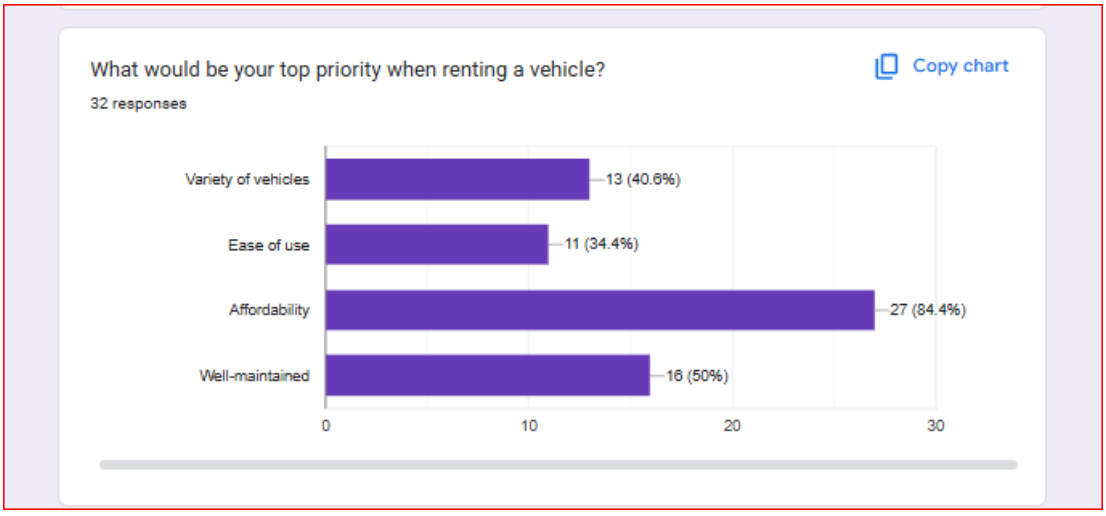


Figure 2.1.1.1.3 : Bar Chart WeGoo Survey

Based on a stacked bar chart, 64.5% respondents prefer the Agent (Whatsapp,Telegram) platform to use to rent a car. Only 9.7% respondents prefer Instagram as a platform to use to rent a car. Hence, a top priority for respondents to rent a vehicle is Affordability that comes in first place, meanwhile ease of use ranks last.

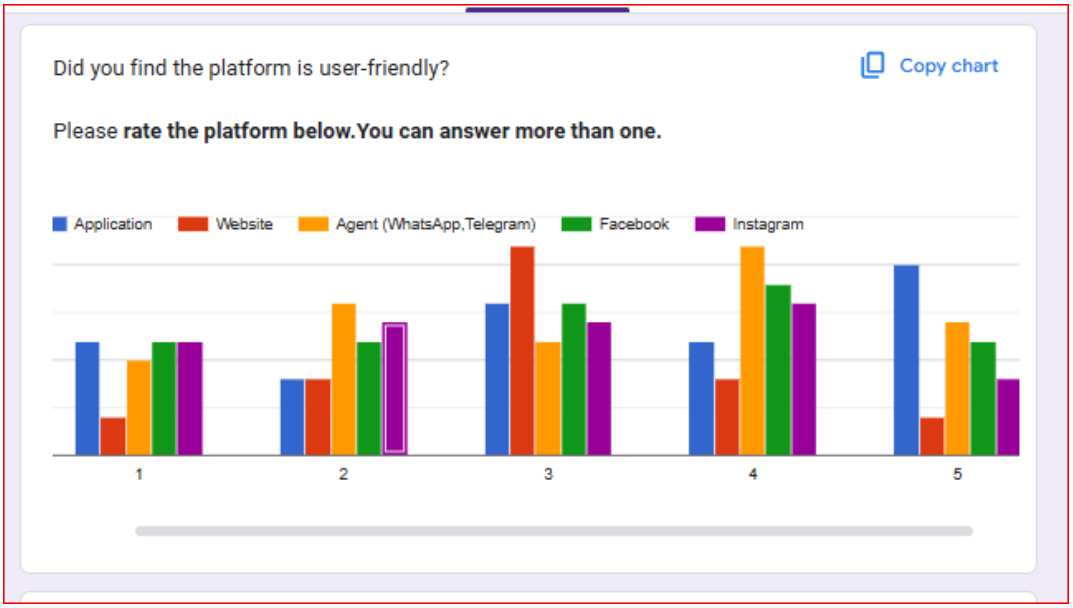


Figure 2.1.1.1.4 : Bar Chart WeGoo Survey

This chart shows user ratings for how user-friendly different platforms are. The Application Agent platforms are seen as the most user-friendly.Website is the least preferred.

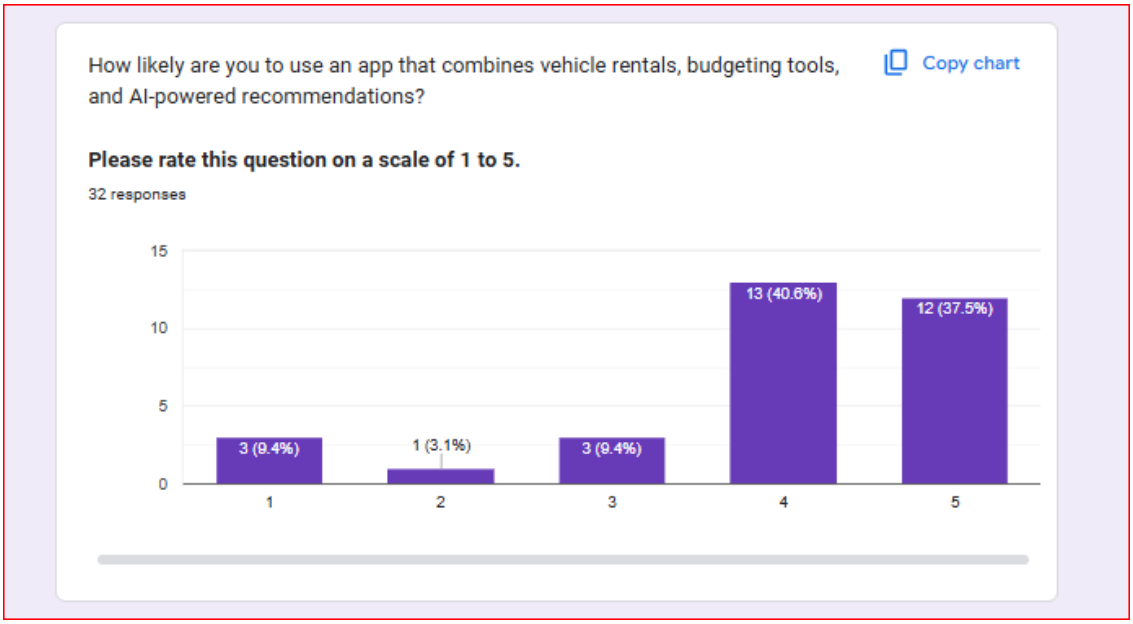


Figure 2.1.1.1.5 : Bar Chart WeGoo Survey

This pie chart shows that 40.6% of respondents preferred to use an app that combines vehicle rentals, budgeting tools, and AI- powered recommendations and only 3.1% of them not preferred.

Did you find the OpenCV feature helpful for users to detect scratch? Why?

10 responses

Yes, it provided consistent, objective results that reduced human error.

Not very helpful because it sometimes marked shadows as scratches.

Somewhat helpful, it worked for major scratches but missed some minor ones.

Yes, the detection was quick and saved me time compared to manually inspecting images.

Yes, the feature was easy to use and didn't require technical knowledge.

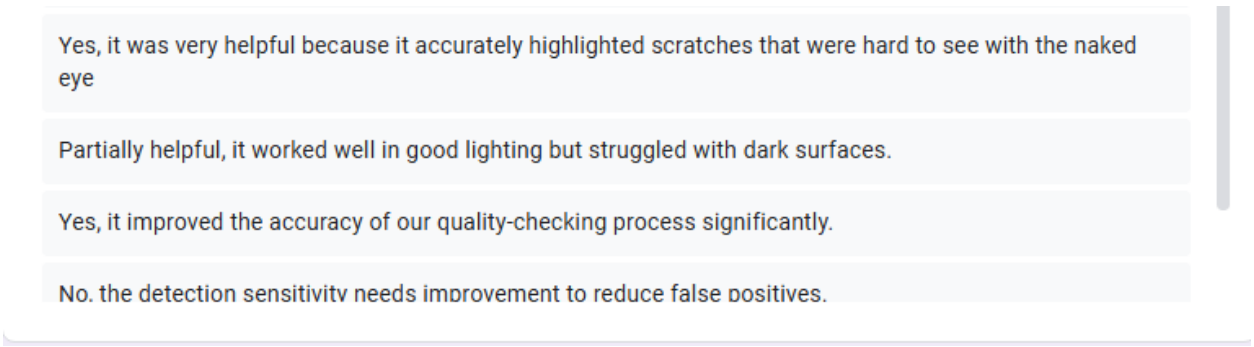


Figure 2.1.1.1.6 : Suggestion Section WeGoo Survey

The form asks users whether they found the OpenCV scratch-detection feature helpful and why. The responses reveal mixed but generally positive feedback, highlighting both strengths and areas for improvement.

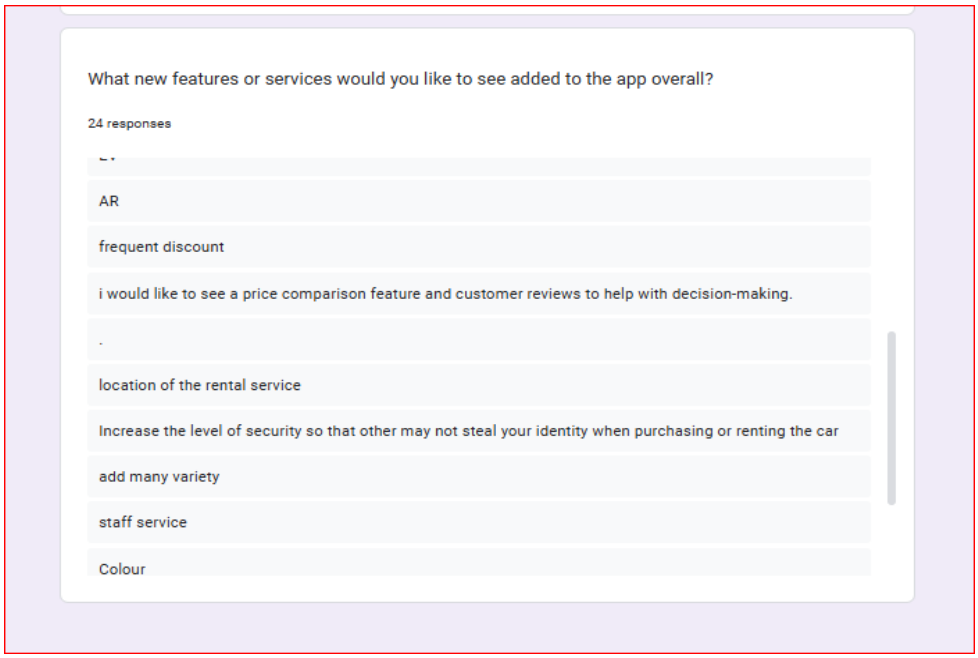


Figure 2.1.1.1.7 : Comment Section WeGoo Survey

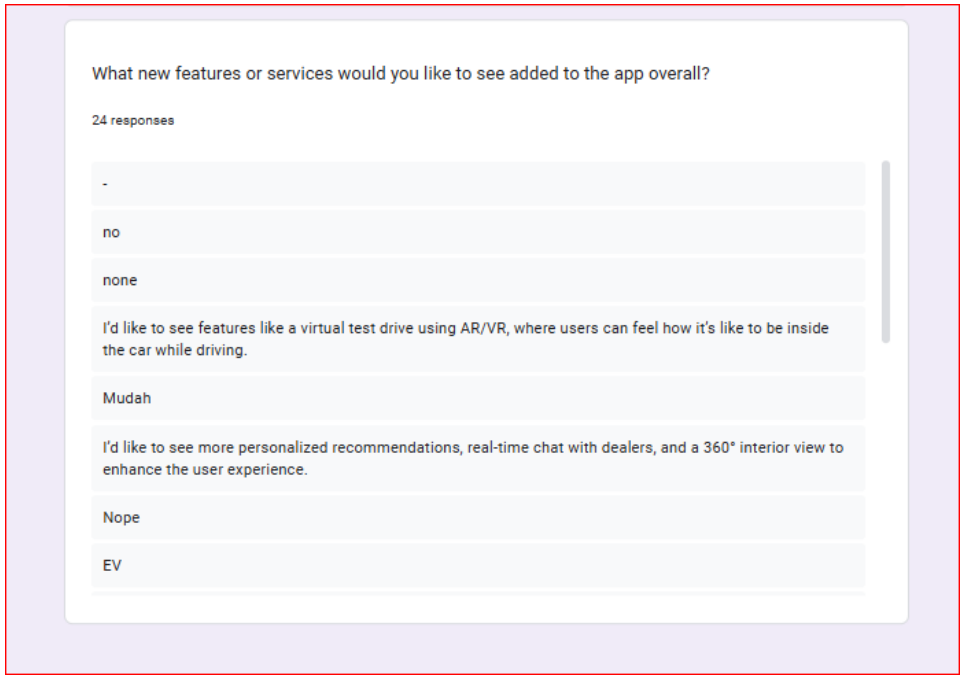


Figure 2.1.1.1.8 : Comment Section WeGoo Survey

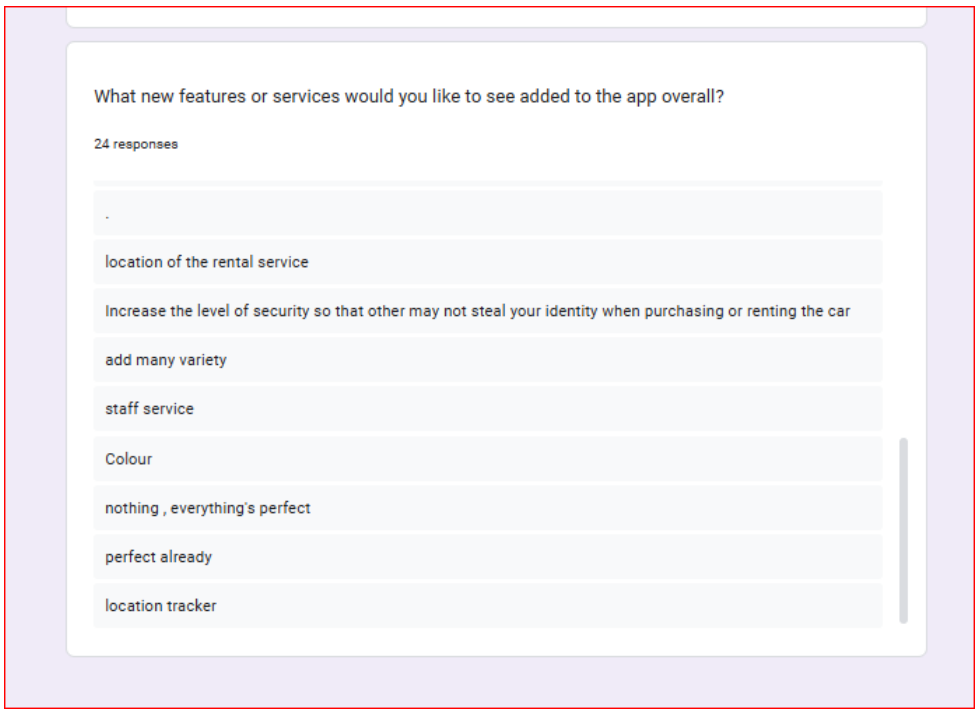


Figure 2.1.1.1.9 : Comment Section WeGoo Survey

This the response of the question which is, what new features or services would they like to see added to the app overall.

2.2 PRIMARY SOURCES

Experimentation on processes, material or any related. Observation on process on going and do market research or survey for checking on usage either this application or software have demand based on proposed project.

2.2.1 RESEARCHING OF RELATED APPLICATION

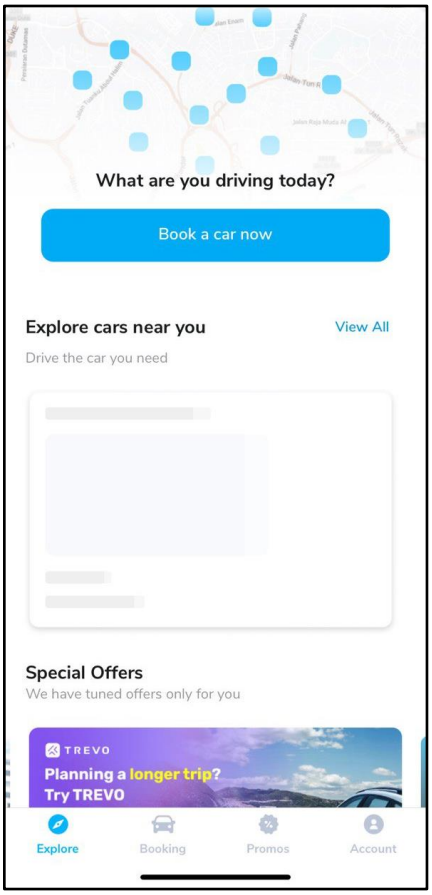


Figure 2.2.1.1 : Homepage SOCAR

This is the homepage of the SOCAR car-sharing app. At the top, there's a map showing available cars nearby with blue markers. Just below it, there's a question asking, "What are you driving today?" followed by a big blue button that says "Book a car now."

Next is a section called “Explore cars near you,” where users can see a preview of cars available nearby and tap “View All” to see more options. Below that is the “Special Offers” section with a banner promoting longer trips using TREVO.

At the bottom, there’s a menu bar with four tabs: Explore, Booking, Promos, and Account. The Explore tab is currently selected. The overall design is clean and easy to use.

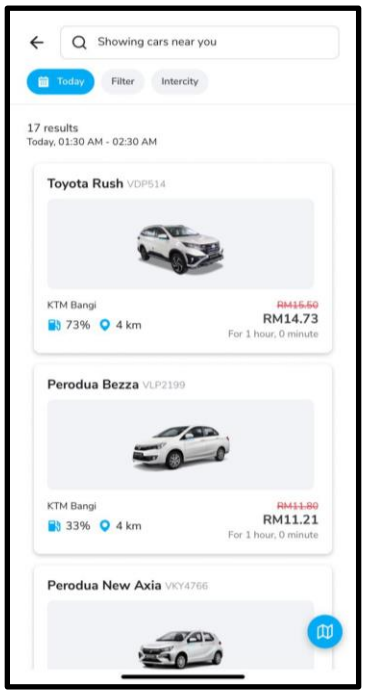


Figure 2.1.2.2 : SOCAR Option

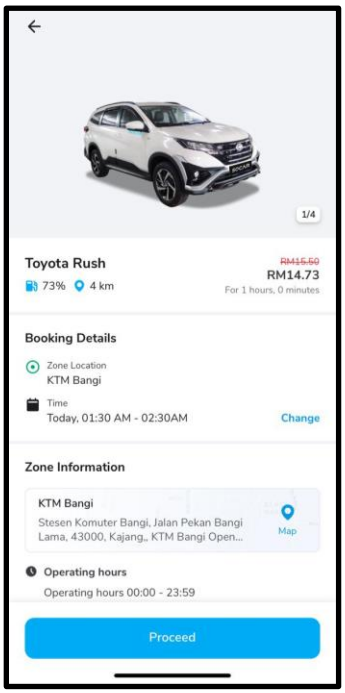


Figure 2.1.2.3: SOCAR Booking Details

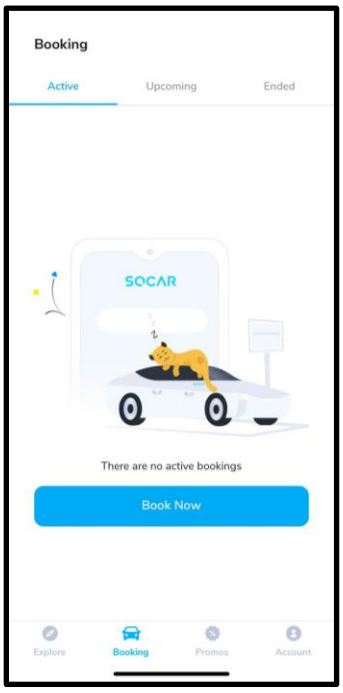


Figure 2.1.2.4 : SOCAR Booking History

Figure 2.2.1.2 : SOCAR Booking Page

This screen shows the booking page of the app. It has tabs for Active, Upcoming, and Ended bookings. There’s also a blue "Book Now" button that lets users start a new booking. This screen displays a list of cars available nearby. Users can filter the search by date, location, and other

options. Each car listing shows the car model, its location, distance from the user, battery level, and price. There are clear images of the cars and useful details to help users choose the right vehicle.

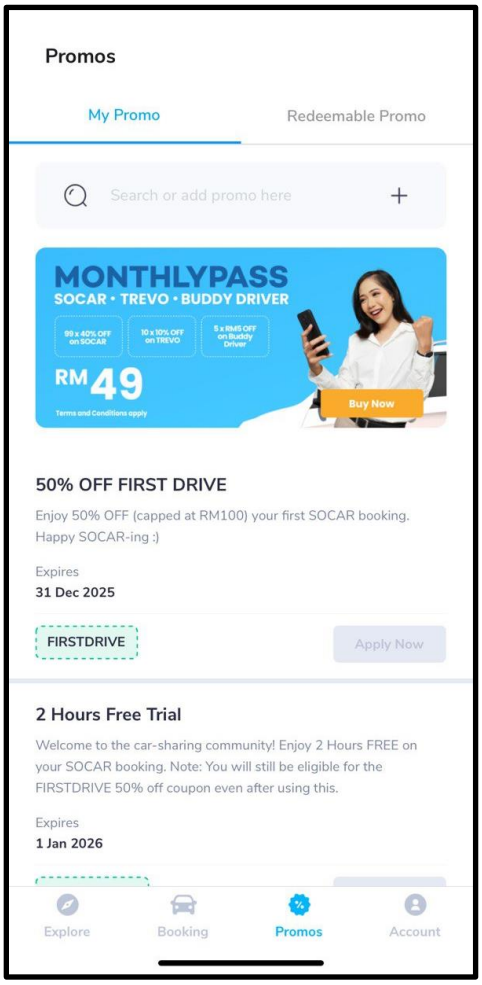


Figure 2.1.2.6 : SOCAR My Promos Page

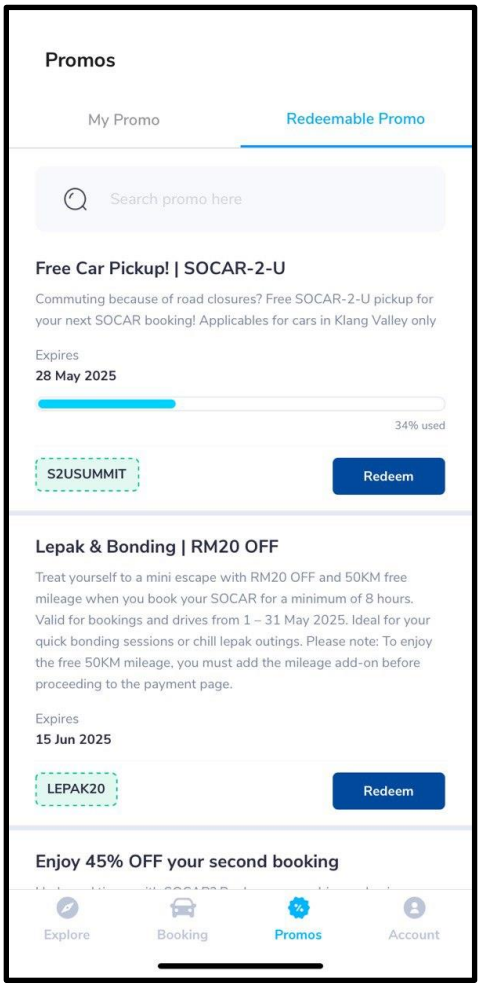


Figure 2.1.2.6 : SOCAR Redeem Page

Figure 2.2.1.3 : SOCAR Promos Page

Presenting various promotional offers to the user. The promotions displayed include a "MONTHLY PASS" for car-sharing and ride services at a discounted price, along with specific offers such as "50% OFF FIRST DRIVE" for new users and a "2 Hours Free Trial." Each promotion provides details on its benefits, terms, and expiry dates, suggesting a well-organized system for managing user incentives. The presence of a search bar and navigation options at the bottom, including

"Explore," "Booking," "Promos," and "Account," indicates a comprehensive app designed for mobility or related services.

These include a "Free Car Pickup!" service designed for convenience, particularly in situations like road closures, and a "Lepak & Bonding" deal that offers a discount and free mileage for longer car bookings. Each promotion is clearly presented with its specific terms, expiry date, and details on how to avail the offer.

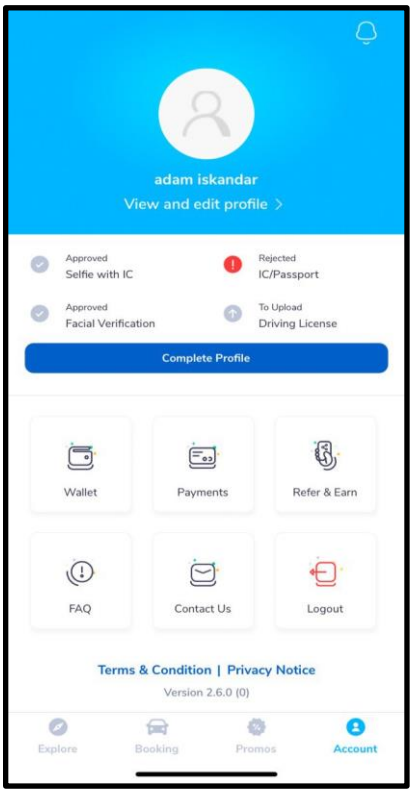


Figure 2.2.1.4 : SOCAR Users Profile

The image shows the mobile app's **Account/Profile** section. At the top, there's a profile area with a picture placeholder, username, and edit option. Below is a **verification status** section showing identity and facial verification progress, with indicators for approved, rejected, or pending actions (e.g., uploading a driving license). A **"Complete Profile"** button prompts the user to finalize info. Further down, a **grid of buttons** provides quick access to features like Wallet, Payments, Refer & Earn, FAQ, Contact Us, and Logout.

- **TREVO**

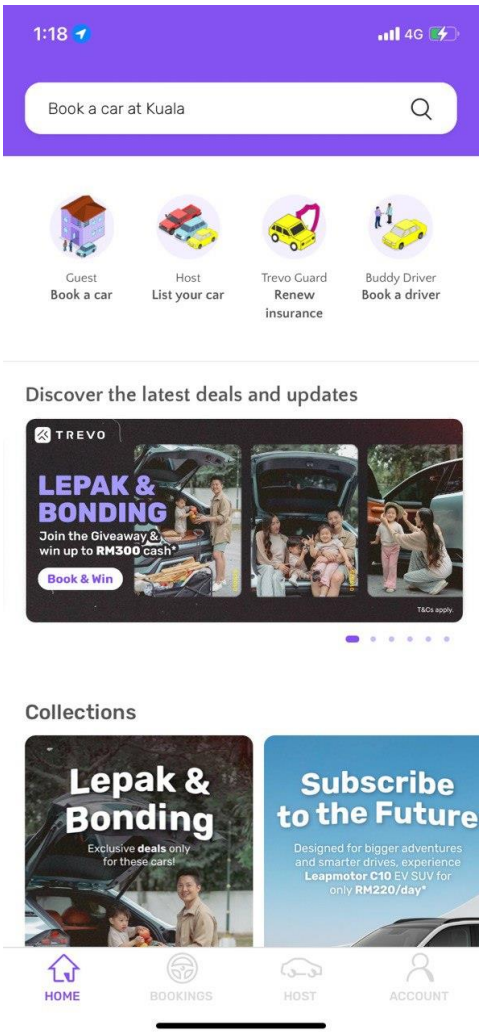


Figure 2.2.1.5 : TREVO Homepage

This image displays the main interface of the Trevo mobile application, characterized by its modern design featuring a clean white background accented with purple elements. The top of the screen includes a search bar, prompting users to book a car indicating its function as a car-sharing or rental platform with location-specific services. Below the search bar, a set of clearly labeled icons represents the app's core functionalities: enabling users to "Book a car" as a guest, "List your car" as a host, "Renew insurance" through Trevo Guard, or "Book a driver" via Buddy Driver.

A significant portion of the screen is dedicated to "Discover the latest deals and updates," featuring a prominent banner promoting a "LEPAK & BONDING" campaign from Trevo.

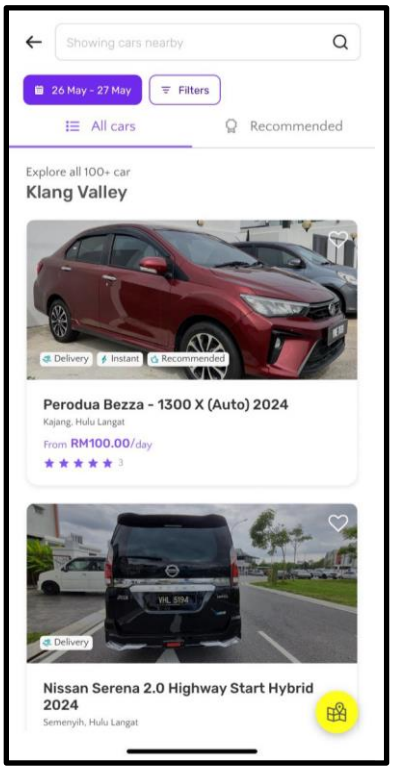


Figure 2.1.2.9: TREVO Option Page

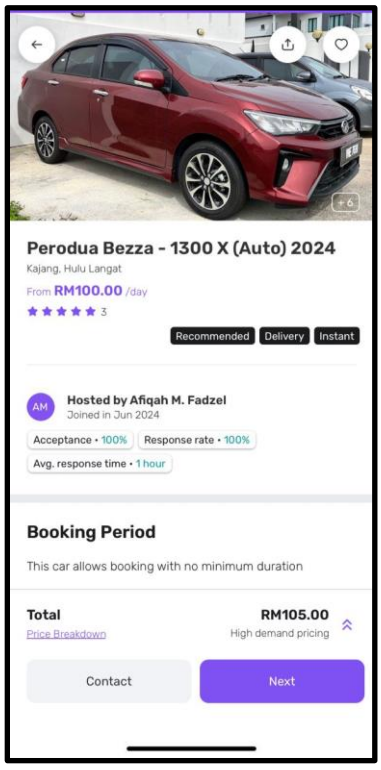


Figure 2.1.2.10: TREVO Booking Details

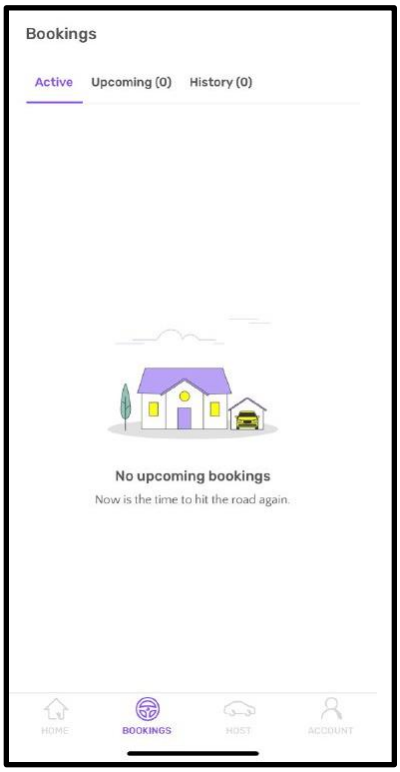


Figure 2.1.2.11: TREVO Booking History

Figure 2.2.1.6 : TREVO Booking Page

"Bookings" section, specifically highlighting an empty "Upcoming" reservations tab.This screen also indicates the presence of "Active" and "History" tabs for managing different stages of car rentals.

The subsequent two images showcase the car selection and booking process. The second image displays a list of vehicles available for rent, sorted by proximity to the user's location. It features alongside its daily rental price, location, and key features like "Delivery" and "Instant" booking. The interface also includes filters for dates and car types, allowing for customized searches. It provides a more detailed, single-car view of the "Perodua Bezza," offering multiple perspectives of the vehicle and crucial information about the car's host, including their acceptance and

response rates. Furthermore, it details the "Booking Period" and the "Total" estimated price, with clear options to either "Contact" the host or proceed to the "Next" step for booking.

Taken together, it demonstrates a well-designed and intuitive user flow within the application, enabling users to seamlessly check their booking status, browse available vehicles with comprehensive information, and efficiently initiate the reservation process.

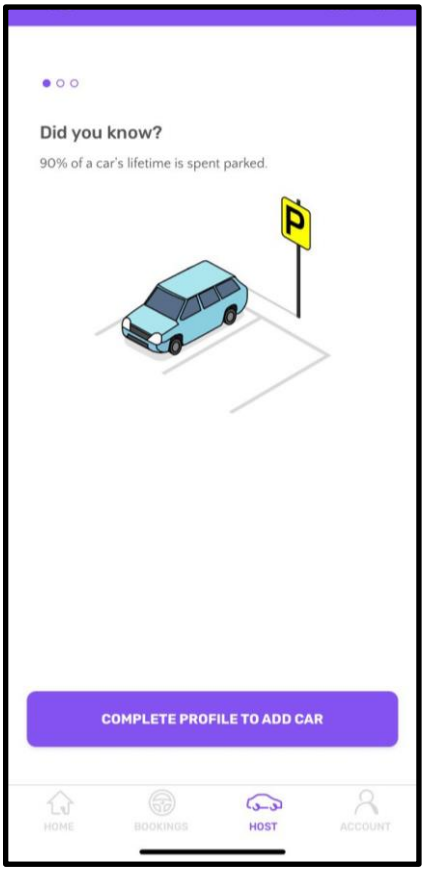


Figure 2.2.1.7 : TREVO Host page

For some users that want to generate income or become a host for a car, they need to complete their profile to add their car.

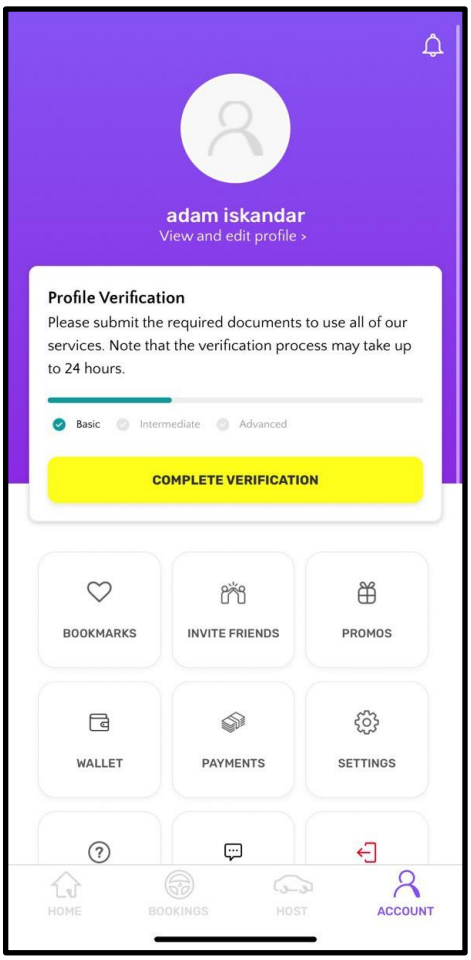


Figure 2.2.1.8 : TREVO Users Profile

This image presents the "Account" or user profile management screen of a mobile application. At the top, a circular icon serves as a placeholder for the user's profile picture, accompanied by the username and an option to "View and edit profile." It features "Profile Verification," instructing the user to submit necessary documents to unlock full app functionality.

Below this verification section, a grid of organized, white square buttons offers direct access to key application features, including "Bookmarks," "Invite Friends," "Promos," "Wallet," "Payments," and "Settings." The bottom of the screen features a consistent navigation bar, providing quick access to other main sections like "Home," "Bookings," "Host," and the highlighted "Account" tab, indicating a well-structured application designed for user management and access to various services.

OpenCV (Open Source Computer Vision Library) plays an important role in the automotive marketing field by improving how vehicles are presented and promoted to customers. With its ability to analyze and process images and videos OpenCV helps businesses detect scratches, dents or damages on vehicles automatically. This increases transparency and trust between customers and sellers or rental providers. In addition, OpenCV can support augmented reality (AR) features that allow users to visualize cars in real-world environments or view different color and design options directly from their mobile devices. These will create a more interactive and engaging experience.

Moreover, OpenCV helps marketing teams manage and enhance vehicle images effectively. It can automatically sort and tag vehicle photos based on color, model or condition making online listings more organized and user-friendly. The technology can also improve image quality by adjusting brightness, removing reflections and sharpening details before publishing. With its ability to analyze customer engagement and visual preferences, OpenCV provides valuable insights that help automotive marketers understand what attracts users ultimately leading to more effective marketing strategies and improved customer satisfaction.

Conclusion

In conclusion, based on the research we conducted on all these applications, we found that there are several aspects we can improve and upgrade to make them more beneficial for users. However, there are also some specific features we can adopt and enhance, such as the loan calculator.

2.3 SECONDARY SOURCES

2.3.1 OBSERVATION

A secondary source is a source that takes from observation,analysis or evaluation of primary sources, rather than presenting original, firsthand data.

We chose observation as our secondary resource to analyze and compare the effectiveness, strengths, and limitations of existing platforms related to car rental services, Update / Manage vehicle and OpenCV visualization. These observations provide valuable insights for the development of WeGoo.

2.3.1.1 ONLINE ARTICLE

- **OpenCV in Automotive Marketing**

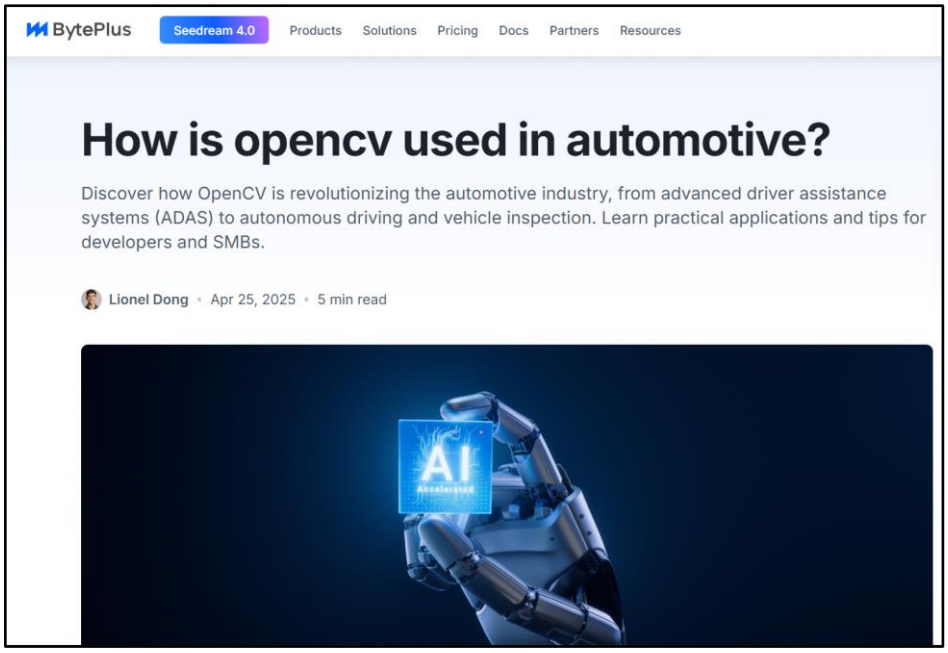


Figure 2.3.1.1 : Articles from BytePlus

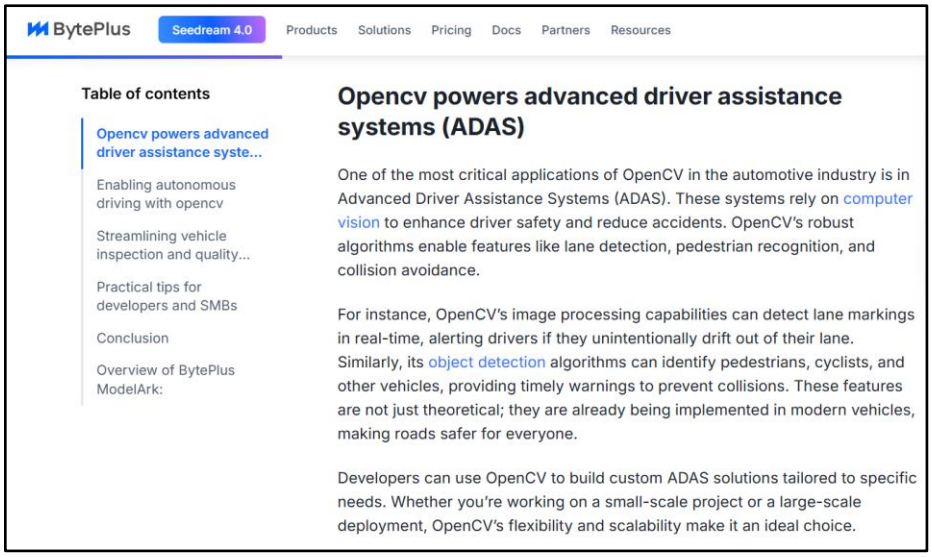


Figure 2.3.1.2 : Articles from BytePlus

- **Car Rental Services**

  For car rentals, we looked at the point of view of both the customer and the rental car agent. We found that as a customer, we need to communicate through the given platform, such as the WhatsApp application. We also have to make a rental payment, which includes a deposit, to confirm the booking. After that, we as customers need to pick up the rented vehicle at the designated location and take photos of the vehicle's condition before using it.

  From the agent's point of view, they need to deliver and communicate with the customer to ensure the rental process runs smoothly. The agent is also responsible for verifying the deposit payment made by the customer and preparing any important documents or information related to the vehicle. In addition, the agent must ensure that the vehicle is in good condition before handing it over to the customer, and take photos of the vehicle's condition at the time of delivery for reference. Once the rental period ends, the agent also needs to recheck the condition of the vehicle upon return and manage the remaining deposit if applicable.

- **Update / Manage Vehicle**

  For the update and manage vehicle process, we observed how rental platforms and agents typically handle vehicle information throughout the rental lifecycle. From the agent's perspective, they are responsible for keeping the vehicle database accurate and up to date. This includes adding new vehicles with complete details such as model, plate number, price, availability status, and images. Agents must also update vehicle information whenever there are changes, such as maintenance schedules, temporary unavailability, or price adjustments. They also need to remove vehicles that are no longer in service to prevent booking issues. Additionally, agents monitor the condition of each vehicle and record any damages, mileage updates, or service requirements. These updates ensure transparency, reduce customer complaints, and help maintain smooth operation of the rental service. Through this observation, we identified the importance of a centralized, user-friendly system like WeGoo that simplifies vehicle management and reduces manual errors.

- **OpenCV (Vehicle Scratch Detection)**

  OpenCV technology enhances user experience by enabling automatic detection of scratches or damages on vehicles through image processing. Using the smartphone camera, the system can analyze vehicle surfaces and identify visible scratches before and after a rental. This feature helps ensure transparency between customers and agents by verifying the vehicle’s condition accurately. However, OpenCV detection requires good lighting and high-quality images to function effectively, and its accuracy may be limited by camera resolution or environmental factors such as reflections and shadows.

2.4 PRODUCT COMPARISON

2.4.1 COMPARISON TABLE

FEATURES	WeGOO	SOCAR	TREVO
Car-sharing model	/	/	X
Peer-to-peer rentals	/	X	/
Standard vehicle variety	/	/	X
Luxury/unique vehicle access	/	X	/
Hourly rental option	/	/	X
Daily/long-term rental	/	/	/
Smart cost calculator	/	X	X
Fuel, toll, discount estimator	/	X	X
Instant booking	/	/	X
Host approval system	/	X	/
Keyless unlock via app	/	/	X
Insurance included	/	/	/
In-app customer support	/	/	/
Can detect scratch on vehicles	/	X	X
Best for tech-savvy users	/	X	X

Figure 2.4.1.1

CHAPTER III

PROJECT METHODOLOGY

3.1 SOFTWARE REQUIREMENT SPECIFICATION

3.1.1 SDLC PROCESS

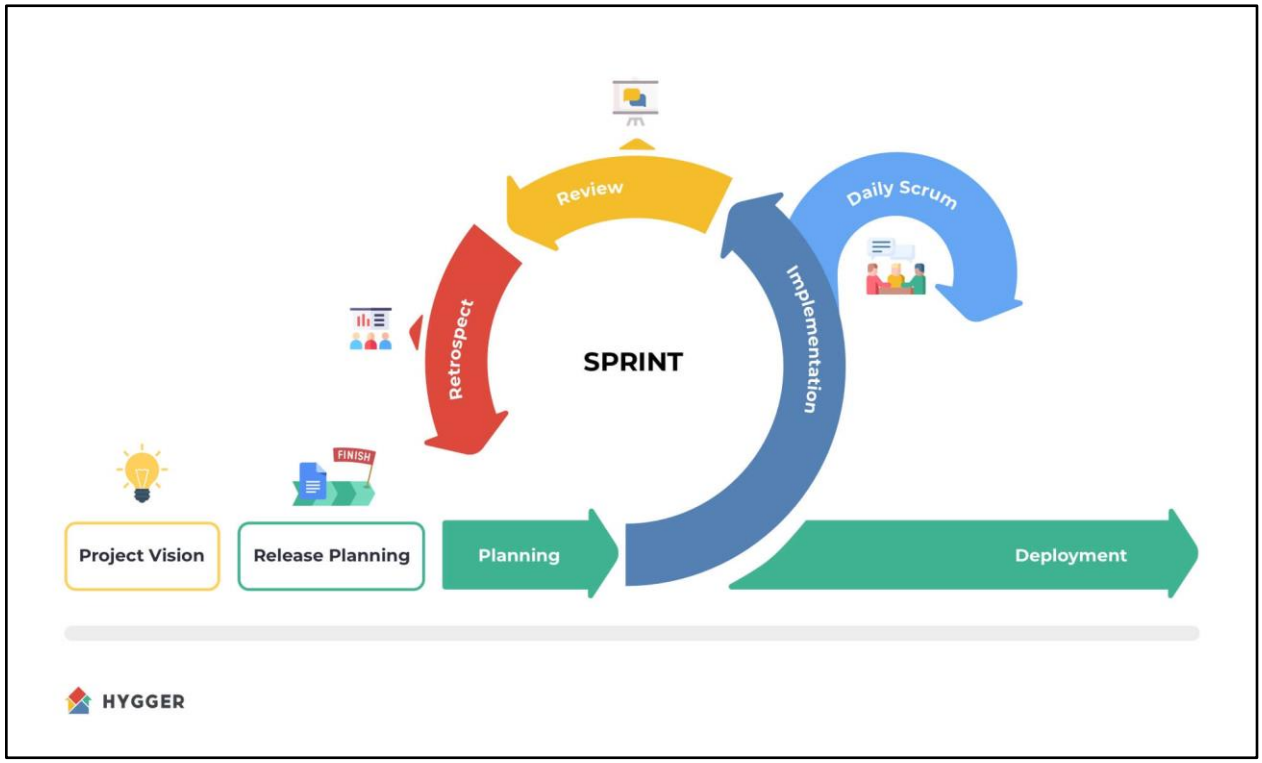


Figure 3.1.1.1 : Sprint SDLC Process

1. Planning Meetings

We begin by organizing planning meetings where all stakeholders, including project managers and team members, come together to define the goals, scope, and timeline of the project. During this phase, we discuss feasibility, assign roles, identify risks, and lay the foundation for a successful development cycle.

2 . Gathering Requirements and Analysis

Next, we gather and analyze the requirements by engaging with users and stakeholders through interviews, surveys, and research. We work closely to ensure all functional and non-functional requirements are clear, complete, and aligned with business needs before moving forward.

3. Design

Once the requirements are finalized, we move into the design phase, where we create the architecture and system models. We design everything from user interfaces to databases and system flows, ensuring that we have a clear blueprint to guide development and maintain consistency throughout the project.

4. Implementation

With a solid design in place, we begin the implementation phase by writing code and developing features according to the specifications. We collaborate as a development team, following best practices and version control to build the system efficiently and accurately.

5. Review or Testing

After building the system, we conduct thorough testing to ensure functionality, performance, and security. We perform unit tests, integration tests, and user acceptance tests. Here, we identify bugs or mismatches with the requirements and fix them before the software goes live.

6. Retrospect

Once testing is completed, we take time to reflect on the project by holding a retrospective meeting. We discuss what went well, what challenges we faced, and how we can improve in future cycles. This phase helps us grow as a team and deliver better results moving forward.

7. Deployment

Finally, we deploy the software into the live environment where users can access and use it. We ensure the system is stable, monitor performance, and provide support as needed. With deployment complete, we deliver real value to users and complete the development cycle.

3.1.2 USE CASE DIAGRAM

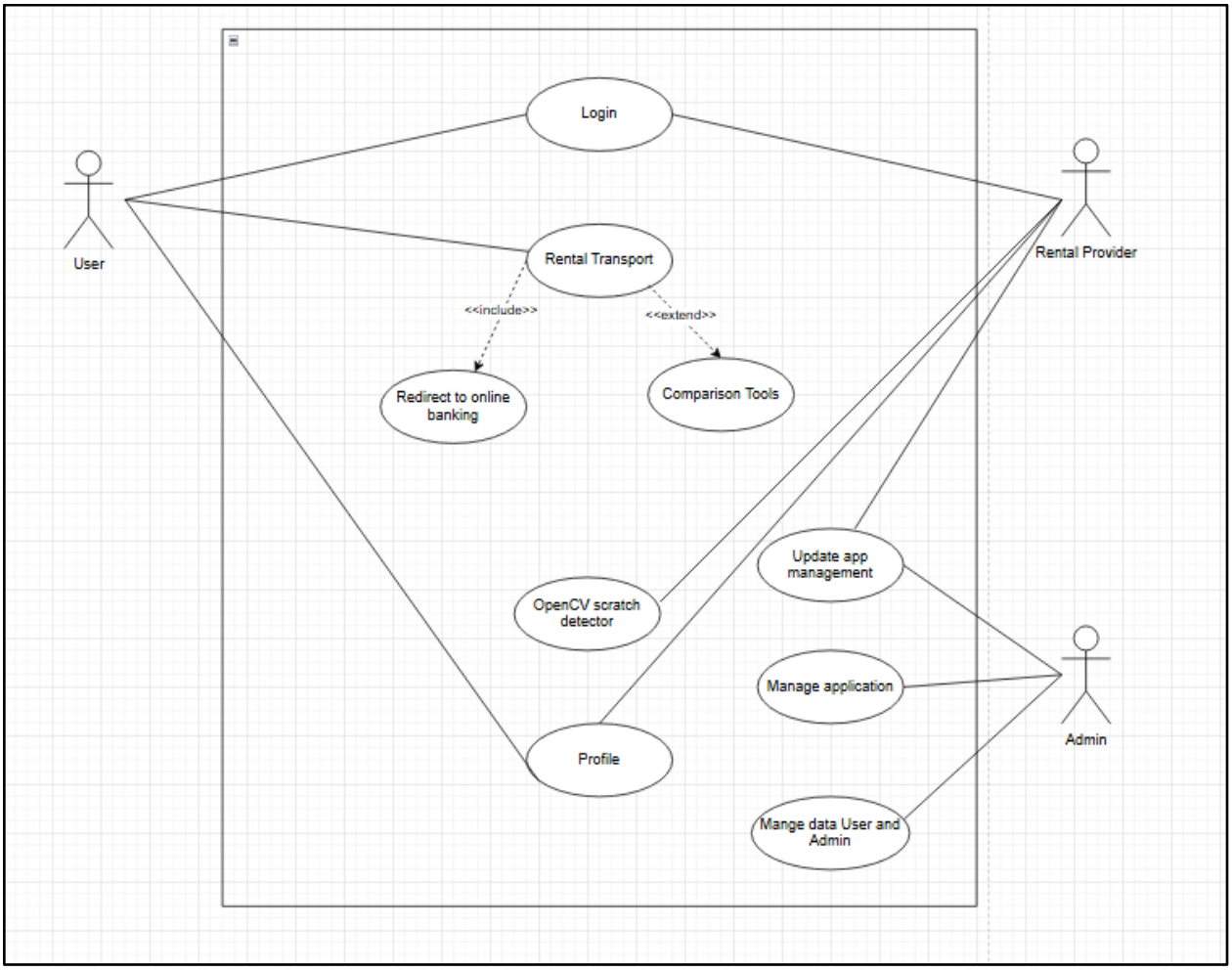


Figure 3.1.2.1 : Use Case Diagram

3.1.3 USE CASE SPECIFICATION

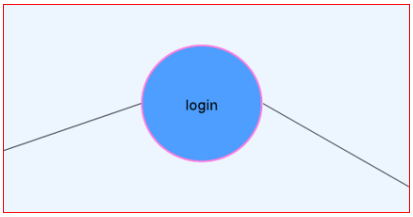


Figure 3.1.3.1 : Login

Use case name	Login
Description	This use case describes the process by which a User or Admin authenticates into the WeGoo Application to access financial tools, maintenance services, and administrative functions.
Actors	Rental Provider and User
Preconditions	1. User is logged in. 2. Financial calculation features (UC2) are enabled.
Basic flow	1. User opens the app. 2. System shows login screen. 3. User enters credentials. 4. System verifies them. 5. If valid, system checks role. 6. System loads the correct dashboard: 7. User Dashboard 8. Admin Dashboard 9. System logs login activity.

Table 3.1.3.1 : Login Continue to page 33

Alternate flow	Invalid Credentials Forgotten Password 1. The user clicks "Forgot Password?" 2. The system sends a password reset link to the registered email. 3. The user follows the link to reset their password securely. Account Lockout (Security Measure) 1. After 3 failed login attempts: 2. The system temporarily locks the account (5-minute cooldown). 3. Displays: "Account locked. Try again later or reset your password." Session Expired (Inactivity) 1. .If the user is inactive for 30 minutes: 2. The system automatically logs them out. 3. Requires re-authentication for security
Post condition	Successful Login: 1. The user gains access to role-specific features. 2. A secure session token is generated. Failed Login: 1. The system logs the attempt for security auditing. 2. If locked out, the user must wait or reset their password.

Table 3.1.3.1 : Login

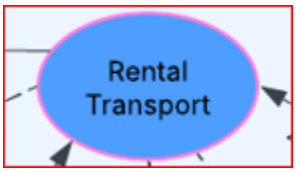


Figure 3.1.3.2 : Rental Transport

Use case name	Rental Transport
Description	This use case allows users to track and manage recurring expenses, specifically Rental and Transport costs, as part of their personal financial calculations.
Actors	User
Preconditions	1. User must be logged into the system. 2. User must have financial calculation features enabled (from UC2). 3. Basic Flow (Main Success Scenario) 4. User selects "Expense Tracking" from the financial tools menu. System displays expense categories: 1. Rental 2. Transport 3. User chooses a category (e.g., Rental): For Rental: 1. System prompts for monthly rent amount, lease duration, and additional costs (utilities, maintenance). For Transport: 1. System asks for fuel costs, public transport fees, vehicle loan/lease payments, and insurance.

Table 3.1.3.2 : Rental Transport continue to the 33

	2. User enters expense details. 3. System validates inputs (ensures numerical values, checks for logical ranges). 4. System saves and categorizes expenses. User can: 1. View a summary of tracked expenses. 2. Compare expenses against income (affordability check). 3. Export data for budgeting.
Alternate flow	Invalid Input (e.g., Negative Values) 1. If user enters invalid data (e.g., negative rent): 2. System rejects input and displays: 3. "Invalid amount. Please enter a positive value." 4. Highlights the incorrect field in red. Partial Data Entry 1. If user skips required fields (e.g., no rent amount provided): 2. System prevents submission and shows: 3. "Missing required information. Please fill all fields." Edit/Delete Existing Expense User can: 1. Modify previously entered expenses. 2. Delete outdated entries.

Post condition	Successful Entry: 1. Expense data is saved and integrated into financial reports. 2. Available for future calculations (e.g., Affordability Check in UC2). Failed Entry: 1. System retains unsaved data for correction. 2. User can retry or cancel.

Table 3.1.3.2 : Rental Transport

Figure 3.1.3.3: Online Banking

Use case name	Redirect to online banking
Description	This use case describes the process where the WeGoo Application securely redirects users to their online banking portal for transaction verification, payment processing, or account linking.
Actors	User
Preconditions	1. User must be logged into WeGoo. 2. User must have selected a banking-related action (loan payment, account sync). 3. User's bank must support third-party redirects (OAuth, secure APIs).

Basic flow	1. User selects **Pay via Bank** or **Link Account**. 2. System shows disclaimer and asks for confirmation. 3. User confirms. 4. System creates secure token/redirect URL. 5. User is redirected to bank login page. 6. User logs in and approves the action. 7. Bank sends success/failure back to WeGoo. 8. System shows **Payment Successful** or error. 9. User returns to dashboard.
Alternate flow	Server Reload 1. If user enters wrong bank credentials: 2. Payment cancelation will pop up
Post condition	**Success:** 1. Transaction/account link confirmed. 2. Financial data updated. **Failure:** 1. No changes made. 2. Error logged

Tabl3.1.3.3: Online Banking

Figure 3.1.3.4: Maintenance and Services

Use case name	Update maintenance & services
Actors	User and Vehicle Provider
Preconditions	1. The user must be logged into the WeGoo application. 2. The user must have an active rental or vehicle registered under their profile. 3. Rental Transport process must have been initiated.
Description	This use case allows the user to see update information related to maintenance and services of their rental transport or owned vehicles within the WeGoo application.
Basic flow	1. User logs into the WeGoo application. 2. User navigates to "Rental Transport" section. 3. User selects “Update Maintenance & Services.” 4. System displays the list of vehicles/rentals and associated maintenance records. 5. User selects a maintenance record to update. 6. User updates maintenance details (e.g., maintenance type, service date, notes). 7. User submits the update request. 8. System validates the submitted information. 9. System saves the updated record. 10. System displays a success confirmation to the user.

Table 3.1.3.4: Maintenance and Services *Continue to page 36*

Alternate flow	**Invalid Data:** 1. System asks user to correct incomplete/invalid input. **Record Not Found:** 1. System shows error and redirects to maintenance overview.
Post condition	**Success:** 1. Maintenance records updated. 2. Dashboard shows new info. 3. (Optional) Confirmation email sent. **Failure:** 1. Records not updated. 2. Dashboard shows old data. 3. (Optional) Email not sent.

Table 3.1.3.4: Maintenance and Services

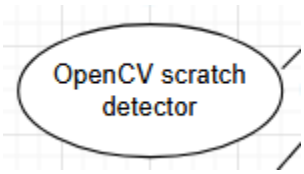


Figure 3.1.3.5 : OpenCV scratch detector

Use case name	OpenCV scratch detector
Description	Allows users to view rental locations and service centers through an embedded Google Maps inside the app.
Actors	Rental Provider
Preconditions	1. Users and Vehicle Provider must be logged in. 2. User's and Vehicle Provider device must have an active internet connection.
Basic flow	1. Rental Provider navigates to "OpenCV scratch detector" feature. 2. System display OpenCV detection test . 3. OpenCV opened.

Table 3.1.3.5 : OpenCV scratch detector *Continue to page 38*

Alternate flow	Internet Connection Error: 1. System shows a message: "Unable to load map. Please check your connection." Restart error: 1. System need to reopened to use it again
Post condition	Success Outcomes: 1. OpenCV loads properly. 2. Camera function. Failure Outcomes: 1. The camera failed to load. 2. OpenCV output failed to display.

Table 3.1.3.5 : OpenCV scratch detector

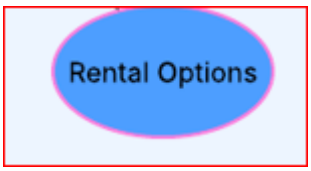


Figure 3.1.3.6 : Rental Option

Use case name	Rental Options
Description	User and Vehicle Provider browses available rental vehicles and selects an option.
Actors	User and Vehicle Provider
Preconditions	User and Vehicle Provider must be logged in.
Basic flow	1. User and Vehicle Provider navigates to "Rental Transport" section. 2. System displays list of available vehicles. 3. User and Vehicle Provider browses and filters options. 4. User and Vehicle Provider selects a vehicle and proceeds to book.
Alternate flow	No results found: 1. Inform user. Error loading rentals: 1. Show error.

Post condition	Successful Postcondition: 1. Rental option selected and booking initiated. Failure Postcondition: 1. Rental selection failed; no booking initiated.

Table 3.1.3.6 : Rental Option

Figure 3.1.3.7 : Comparison Tools

Use case name	Comparison Tools
Description	User and Vehicle Provider compares rental vehicles/services side-by-side
Actors	User and Vehicle Provider
Preconditions	User must be logged in.
Basic flow	1. User and Vehicle Provider selects vehicles for comparison. 2. System displays a comparison chart.
Alternate flow	1. No data available: Show message. 2. Comparison load failure: Display error.
Post condition	Successful Postcondition: 1. Comparison chart displayed successfully. Failure Postcondition: 1. No comparison result shown; user sees error.

Table 3.1.3.7 : Comparison Tools

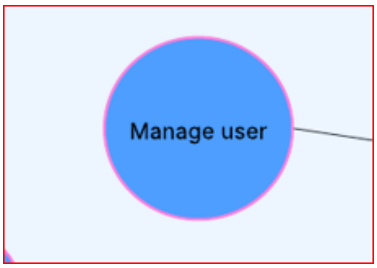


Figure 3.1.3.8 : Manage User

Use case name	Manage user
Description	Admin creates, modifies, or deletes user accounts and permissions.
Actors	Admin
Preconditions	1. Admin is logged in with superuser privileges 2. User management module is accessible
Basic flow	1. Admin opens **Manage Users**. 2. System shows user list + filters + Add/Edit/Deactivate options. 3. Admin selects an action: a. **Add:** Enter email, role, temp password. b. **Edit:** Update profile/privileges. c. **Deactivate:** Mark account inactive. 4. System validates (e.g., unique email). 5. System sends confirmation email to the user.
Alternate flow	Duplicate User 1. System blocks creation, suggests merge Last Admin Deactivation 1. System prevents action to maintain admin access

Post condition	1. User database updated 2. Audit log records changes

Table 3.1.3.8 : Manage User

Figure 3.1.3.9 : Update app management

Use case name	Update app management
Description	Admin deploys app updates and manages feature releases. Rental Provider update vehicles and set up
Actors	Admin and Rental Provider
Preconditions	New app version/features are staged in dev environment
Basic flow	1. Admin and Rental Provider accesses "Update App Management" 2. System shows: - Version control panel - Feature toggle dashboard - Emergency rollback option 3. Admin and Rental Provider: - Schedules version rollout (phased or full) - Enables/disables features via toggles 4. System: - Pushes updates - Notifies users (in-app banner)
Alternate flow	Failed Update 1. Auto-rollback to last stable version Critical Bug 1. Emergency patch deployment

Table 3.1.3.9 : Update app management *Cotinue to page 43*

Post condition	Successful Update Deployment: 1. New app versions/features are live for target users. 2. Version history is logged (timestamp, admin ID, changes made). 3. Users receive in-app notifications about updates (if configured). Feature Toggles Modified: 1. Enabled/disabled features are immediately reflected in production. 2. A/B test groups are updated per new configuration. Rollback (If Triggered): 1. System reverts to the last stable version without downtime. 2. Incident report is generated for root-cause analysis.

Table 3.1.3.9 : Update app management

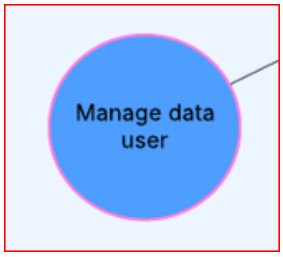


Figure 3.1.3.10: Manage data user

Use case name	Manage data user
Description	Admin handles data compliance and user data requests.
Actors	Admin
Preconditions	Data privacy module enabled
Basic flow	1. Admin selects "Manage Data Use" 2. System provides: - Data Dashboard: Storage metrics, retention policies - User Requests: Export/delete data petitions 3. Admin actions: - Processes GDPR requests (export/erase) - Adjusts retention periods (auto-delete after 2y) - Configures third-party data sharing 4. System executes changes with confirmation
Alternate flow	Legal Hold 1. System exempts specified data from deletion Bulk Request 1. Background processing with progress tracking
Post condition	1. Data policies enforced 2. Compliance reports generated

Table 3.1.3.10: Manage data user

Figure 3.1.3.11: Profile

Use case name	Profile
Description	Users can view and update personal info in WeGoo, including name, contact details, payment preferences, rental history, and saved preferences.
Actors	User and Vehicle Provider
Preconditions	• The user must be logged in to the WeGoo application. • The application must have access to the user database.
Basic flow	1. User logs in and opens **Profile**. 2. System shows current info. 3. User selects **Edit**. 4. User updates details. 5. User saves changes. 6. System validates input. 7. System updates the database. 8. Shows: **"Profile updated successfully."**
Alternate flow	• **Invalid Input:** • System shows error and asks for correction. • **Cancel Update:** • User cancels; no changes saved. • **System Error:** • System alerts user; asks to retry later.
Post condition	**Successful Update:** • Profile updated in database. • UI shows new info. **Unsuccessful Update:** • No changes saved. • Old data kept. • User notified.

Table 3.1.3.11: Profile

3.1.4 SEQUENCE DIAGRAMS

- **LOGIN**

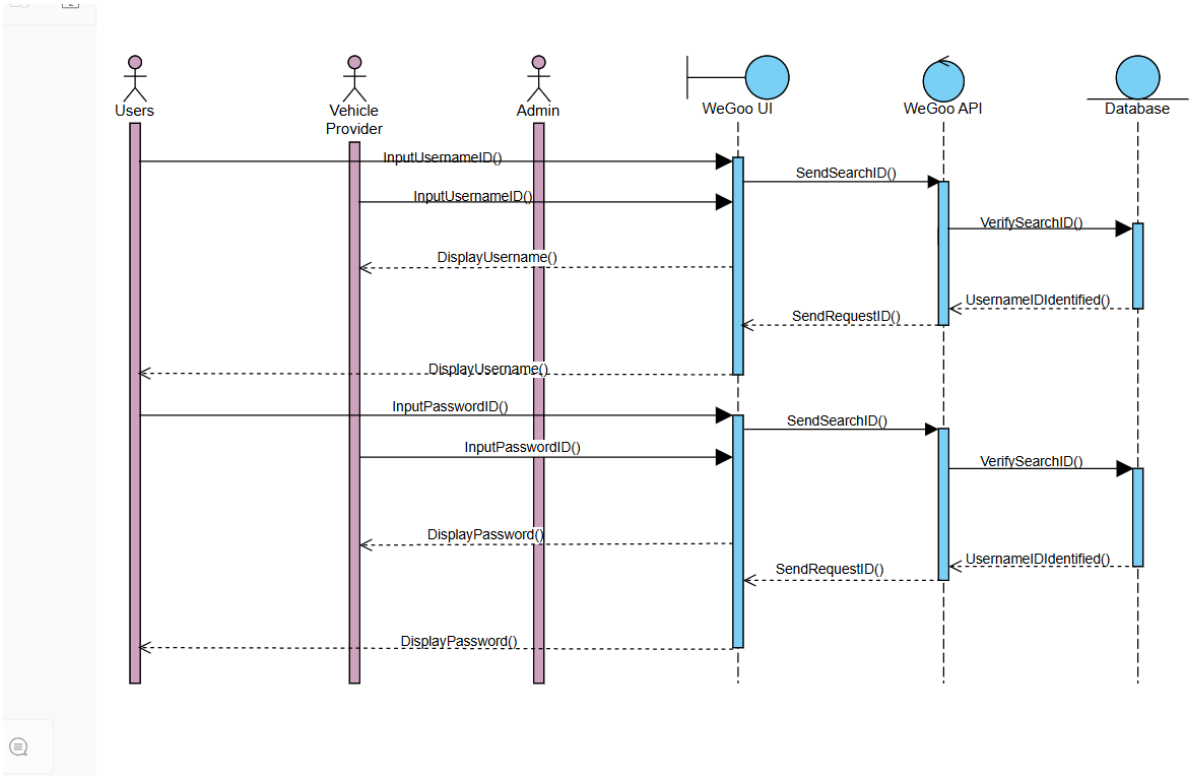


Figure 3.1.4.1 : Login Sequence Diagrams

- **INVALID PASSWORD**

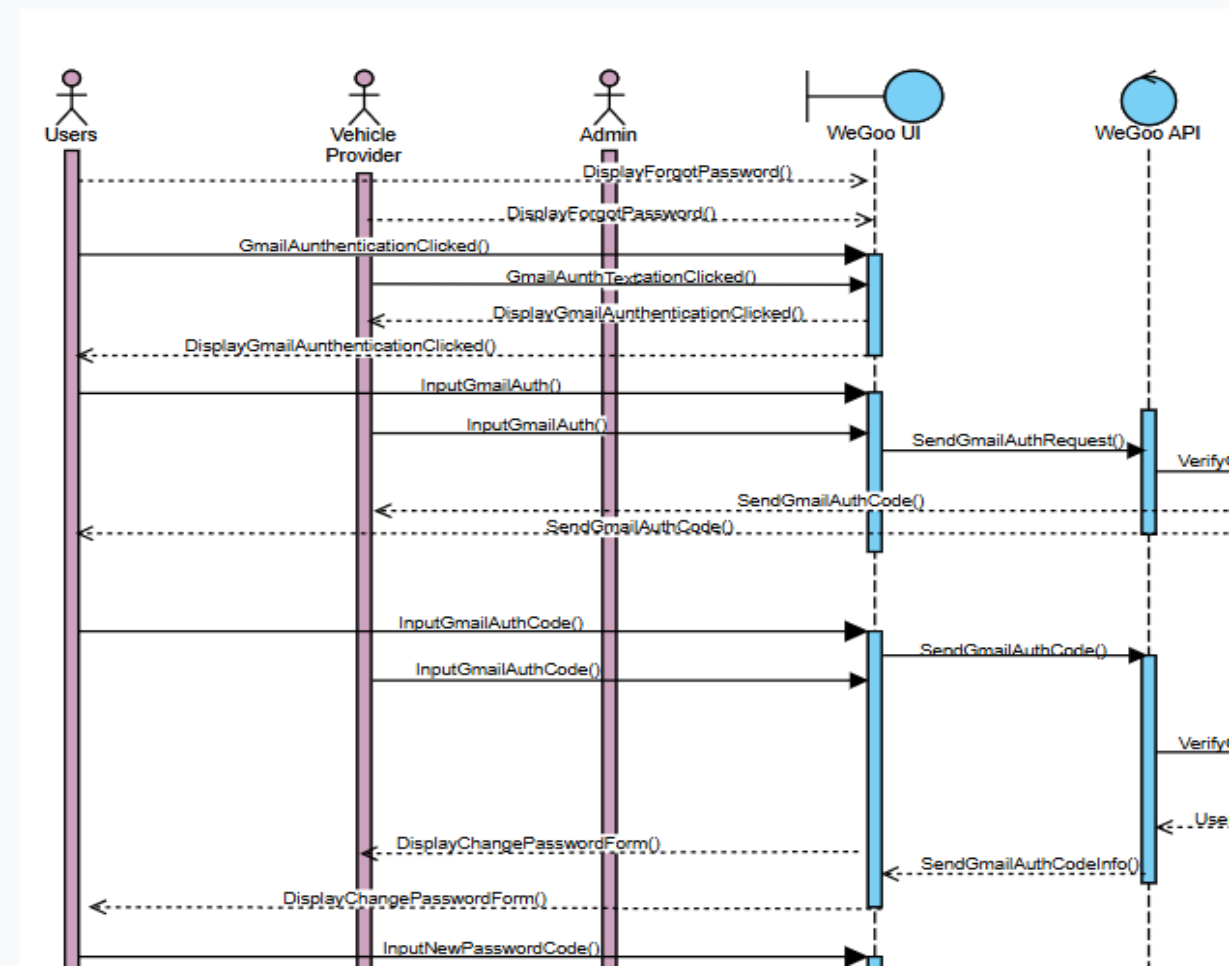


Figure 3.1.4.2 : Invalid Password Sequence Diagrams

- **WeRent**

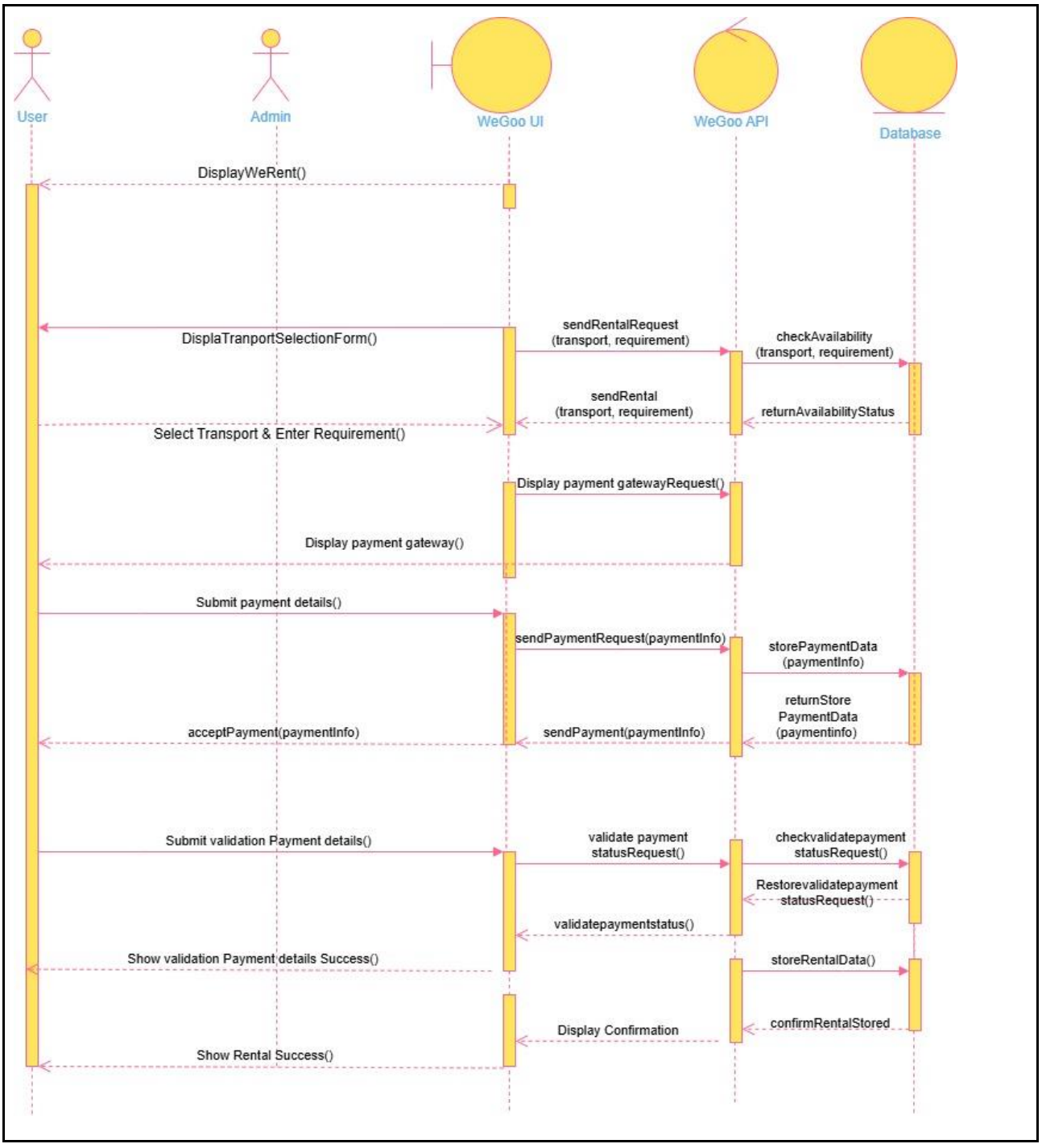


Figure 3.1.4.3: WeRent Sequence Diagrams

- **OpenCV**

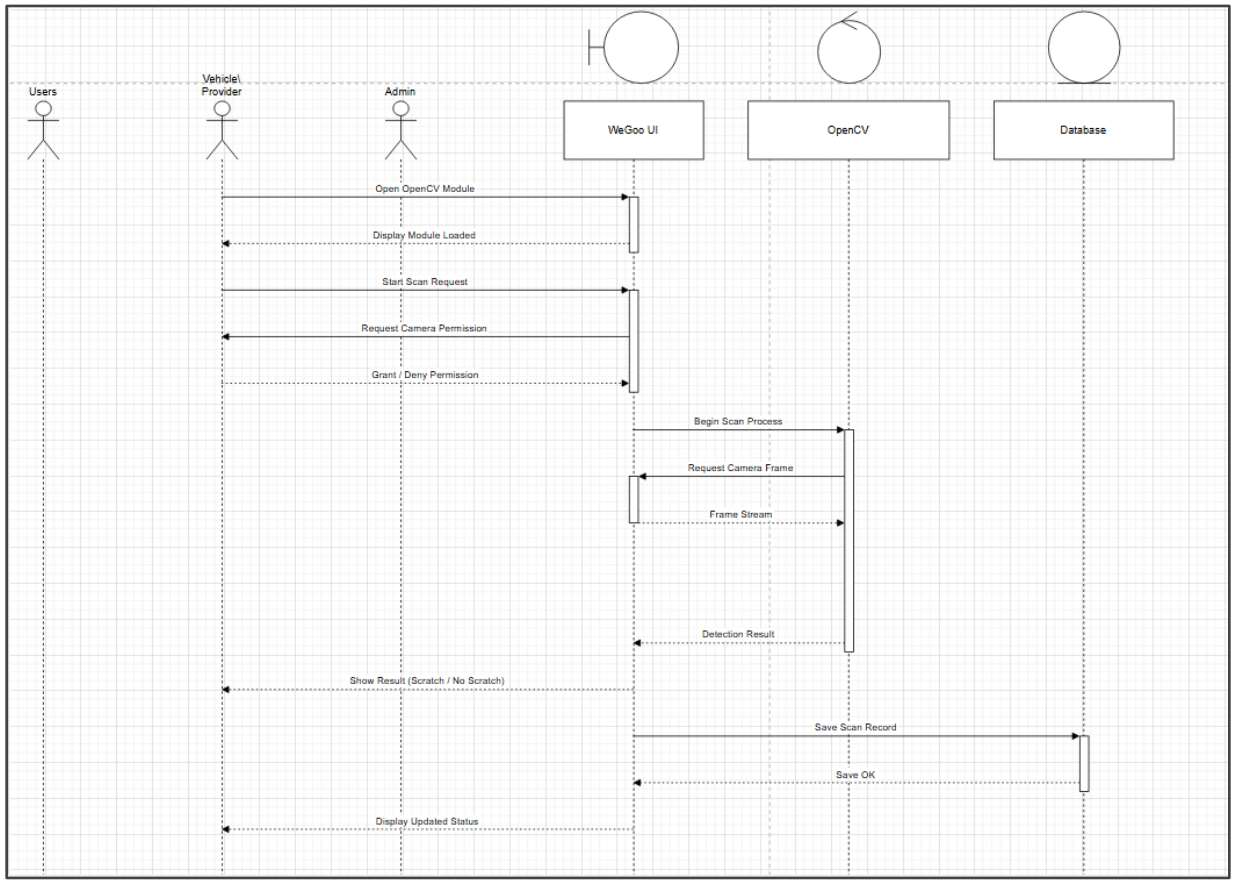


Figure 3.1.4.4 : OpenCV Sequences Diagrams

- **Admin Update Application Management**

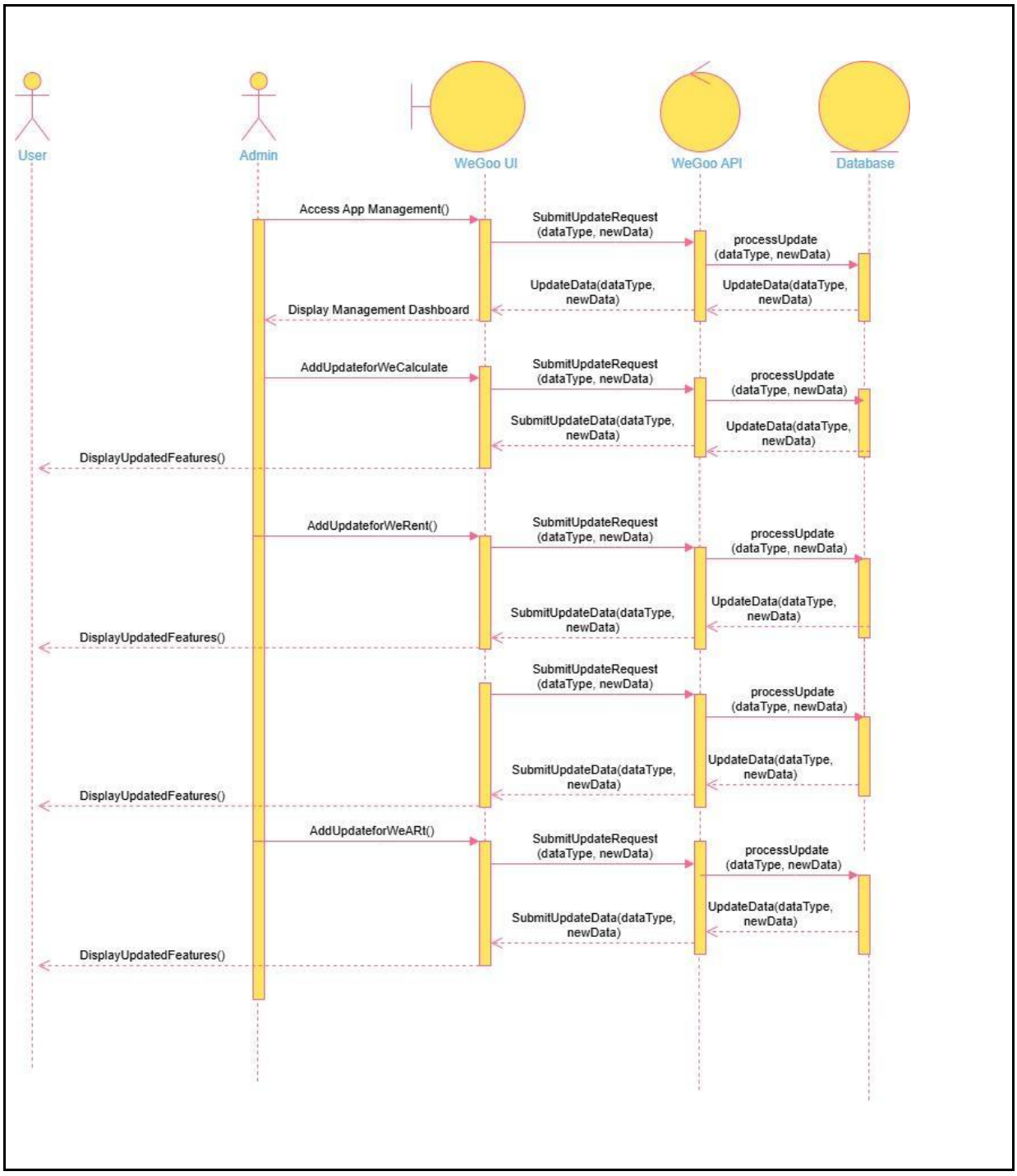


Figure 3.1.4.5 : Admin Update Application Management Sequence Diagrams

- **Admin User Management**

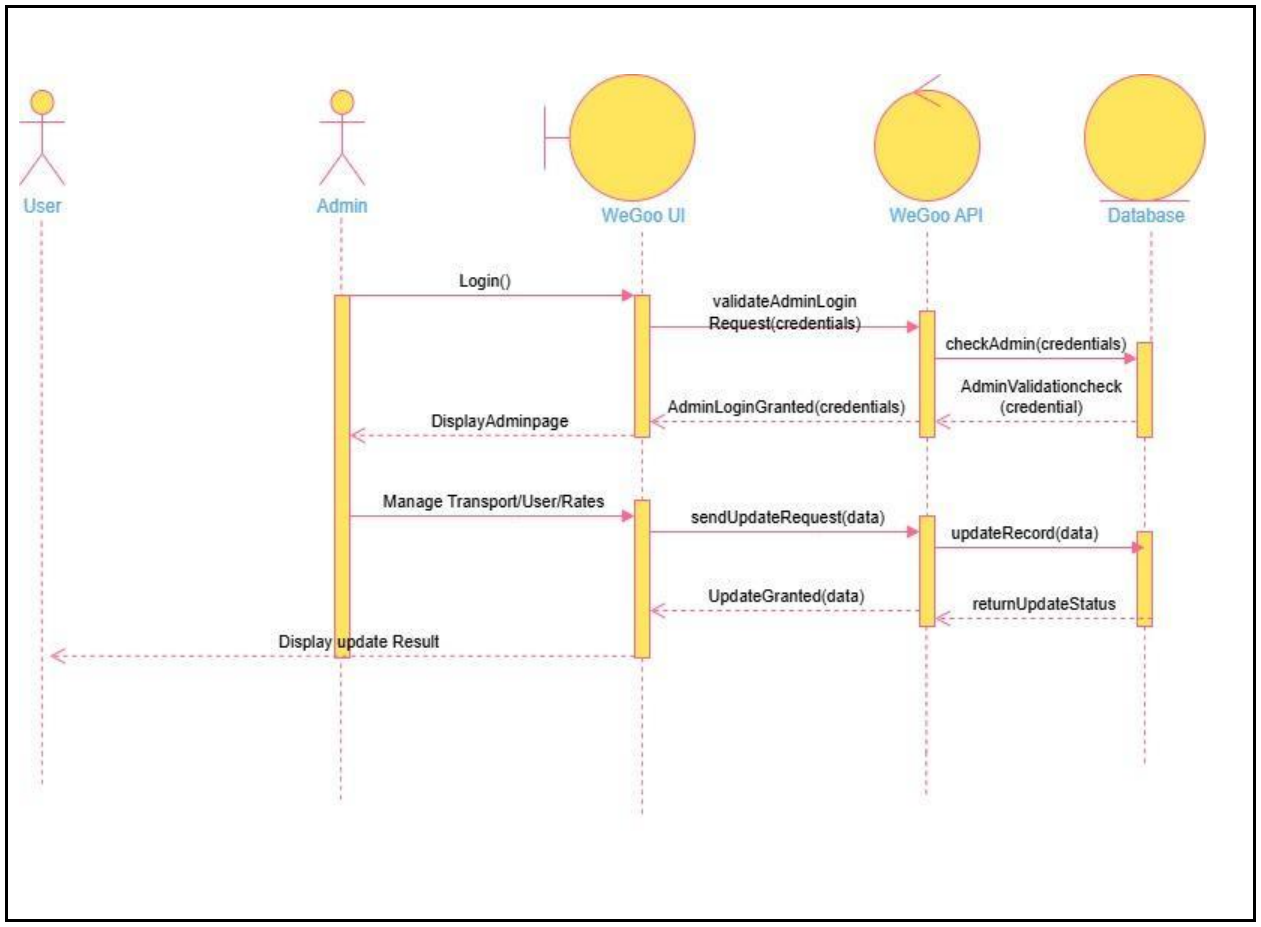


Figure 3.1.4.6 : Admin User Management Sequence Diagrams

3.1.5 ACTIVITY DIAGRAM

- **Login**

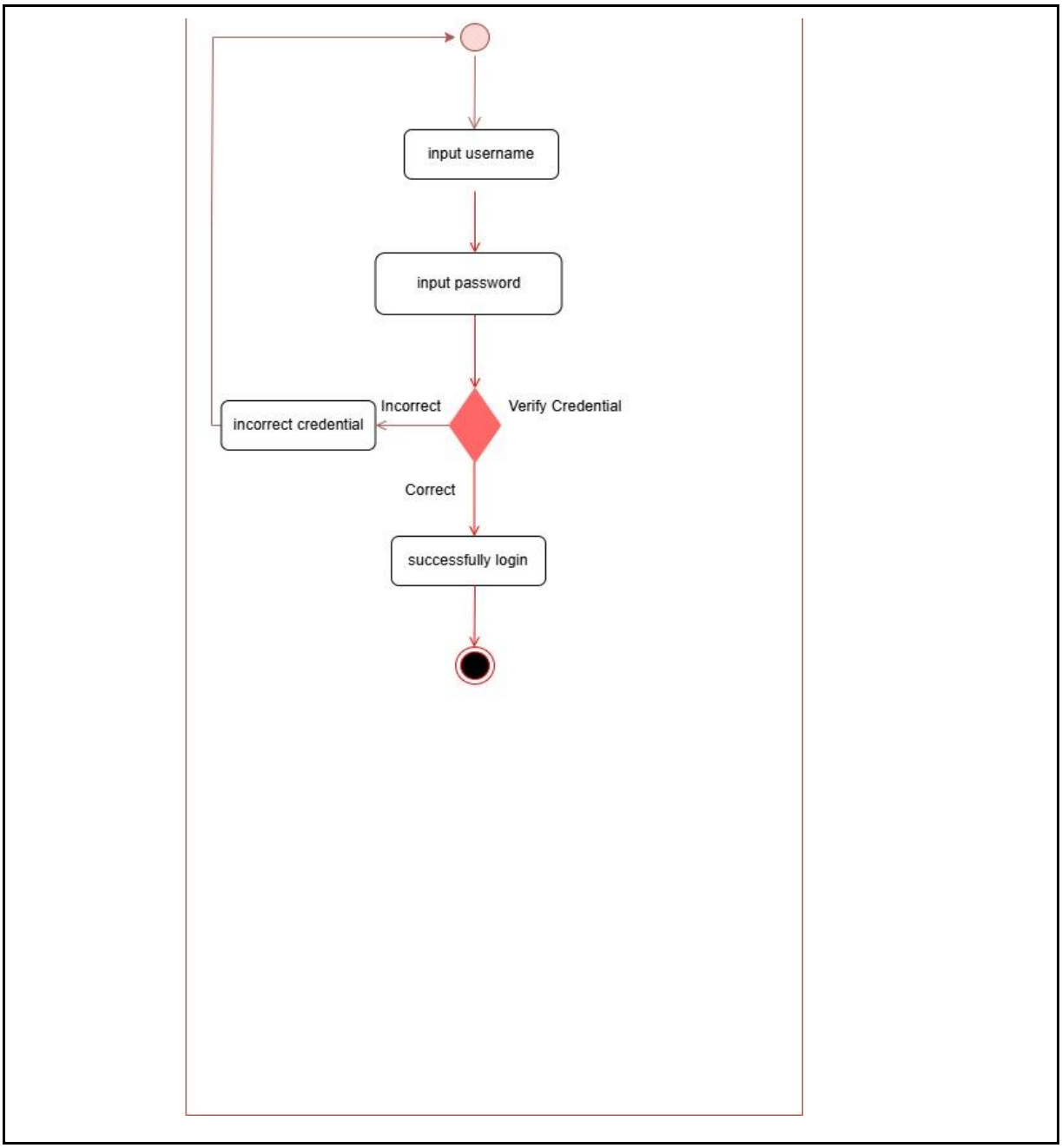


Figure 3.1.5.1 : Login Activity Diagrams

- **WeRent**

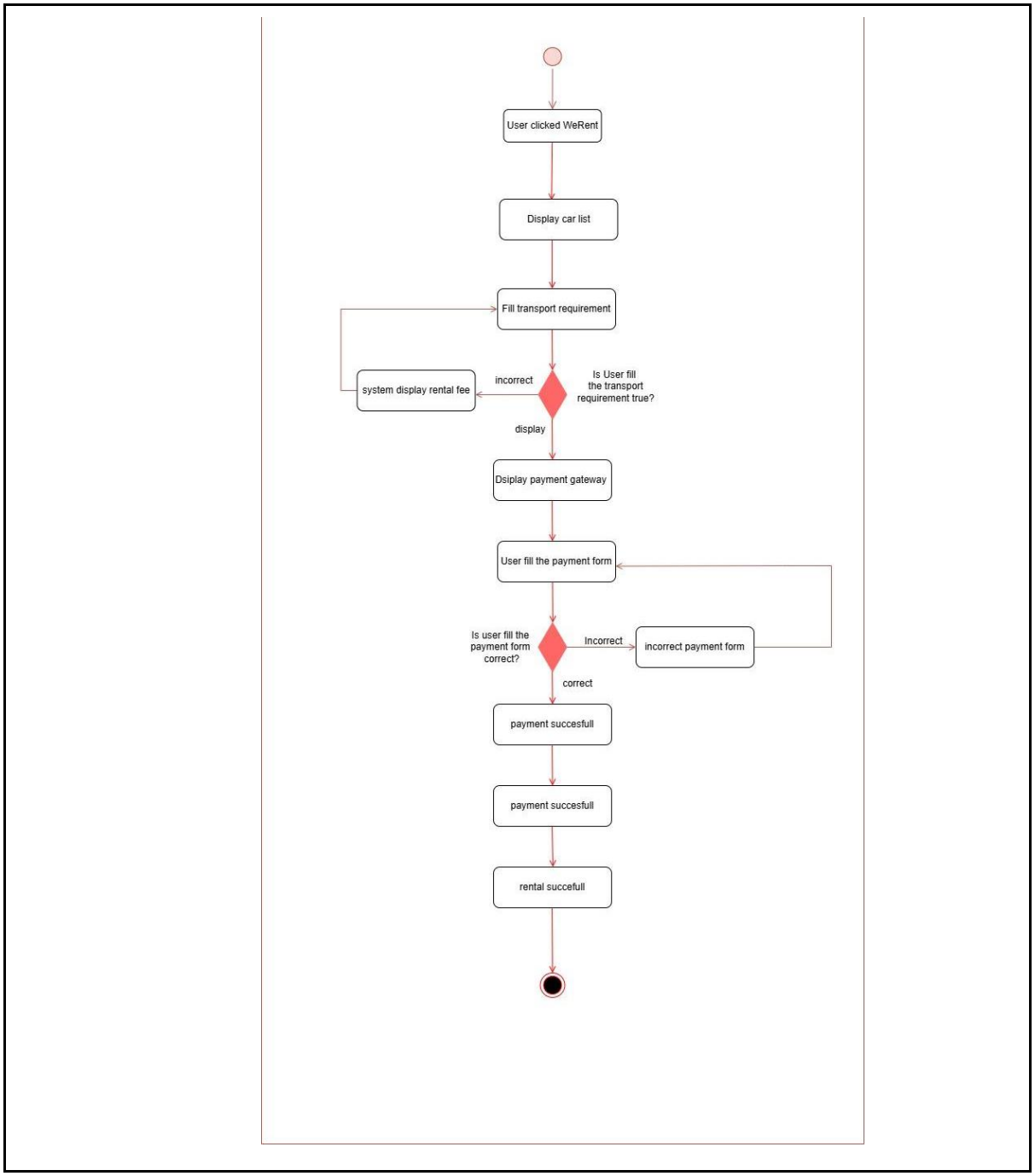


Figure 3.1.5.2 : WeRent Activity Diagrams

- **OpenCV**

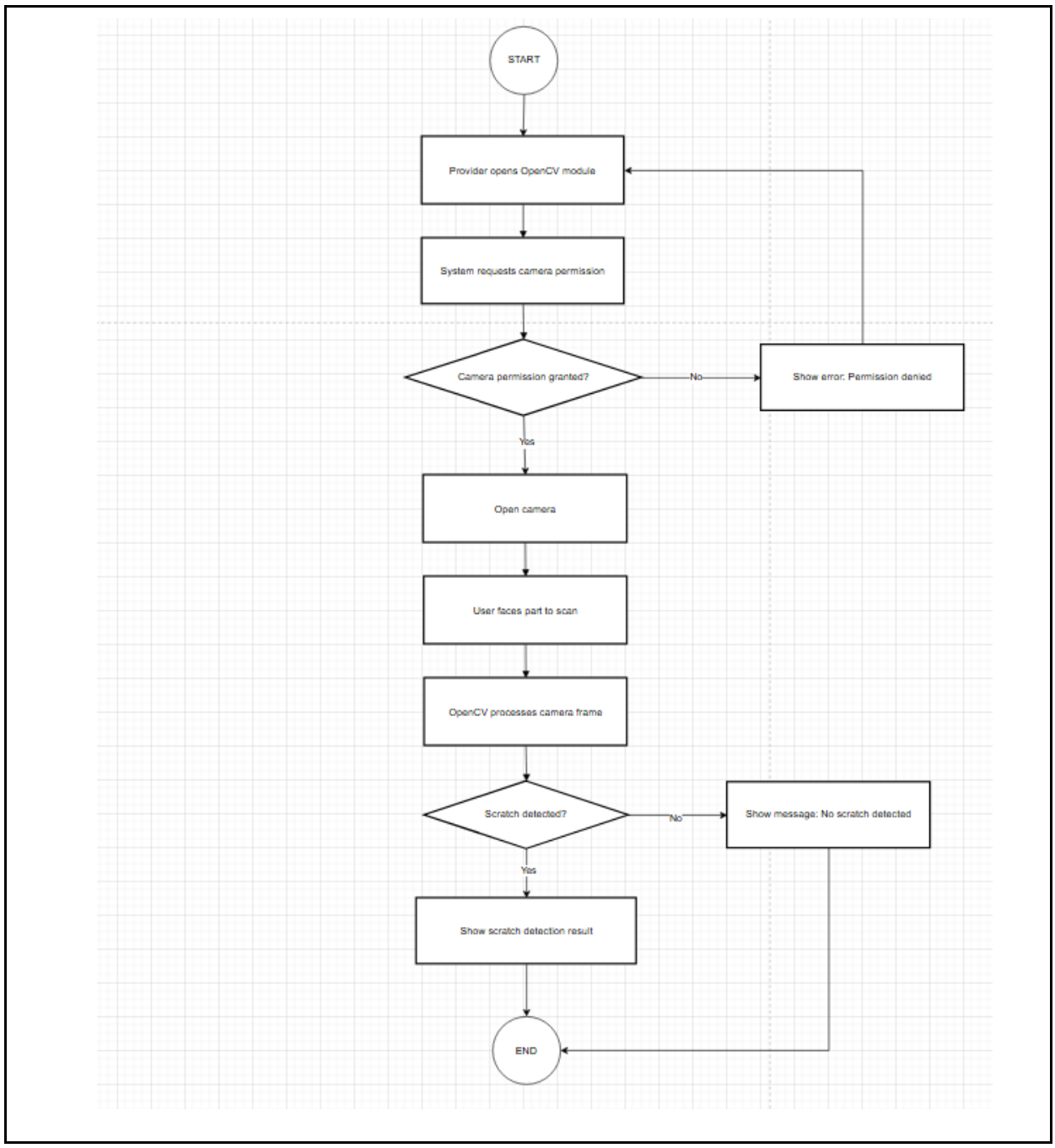


Figure 3.1.5.3 : OpenCV Activity Diagrams

- **Admin and Rental Provider Update Application Management**

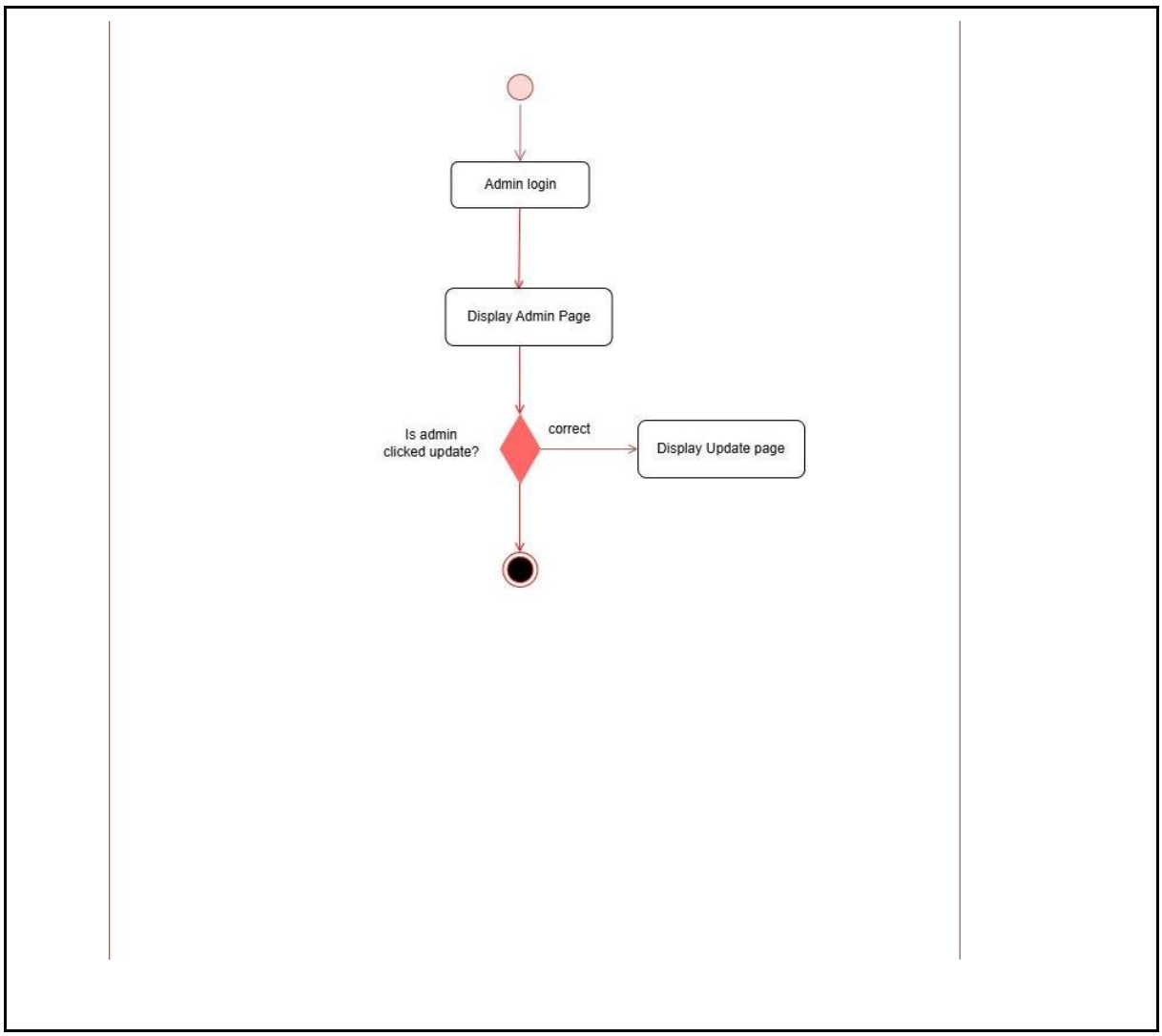


Figure 3.1.5.4 : Admin Update Application Management Activity Diagrams

- **Admin Manage User**

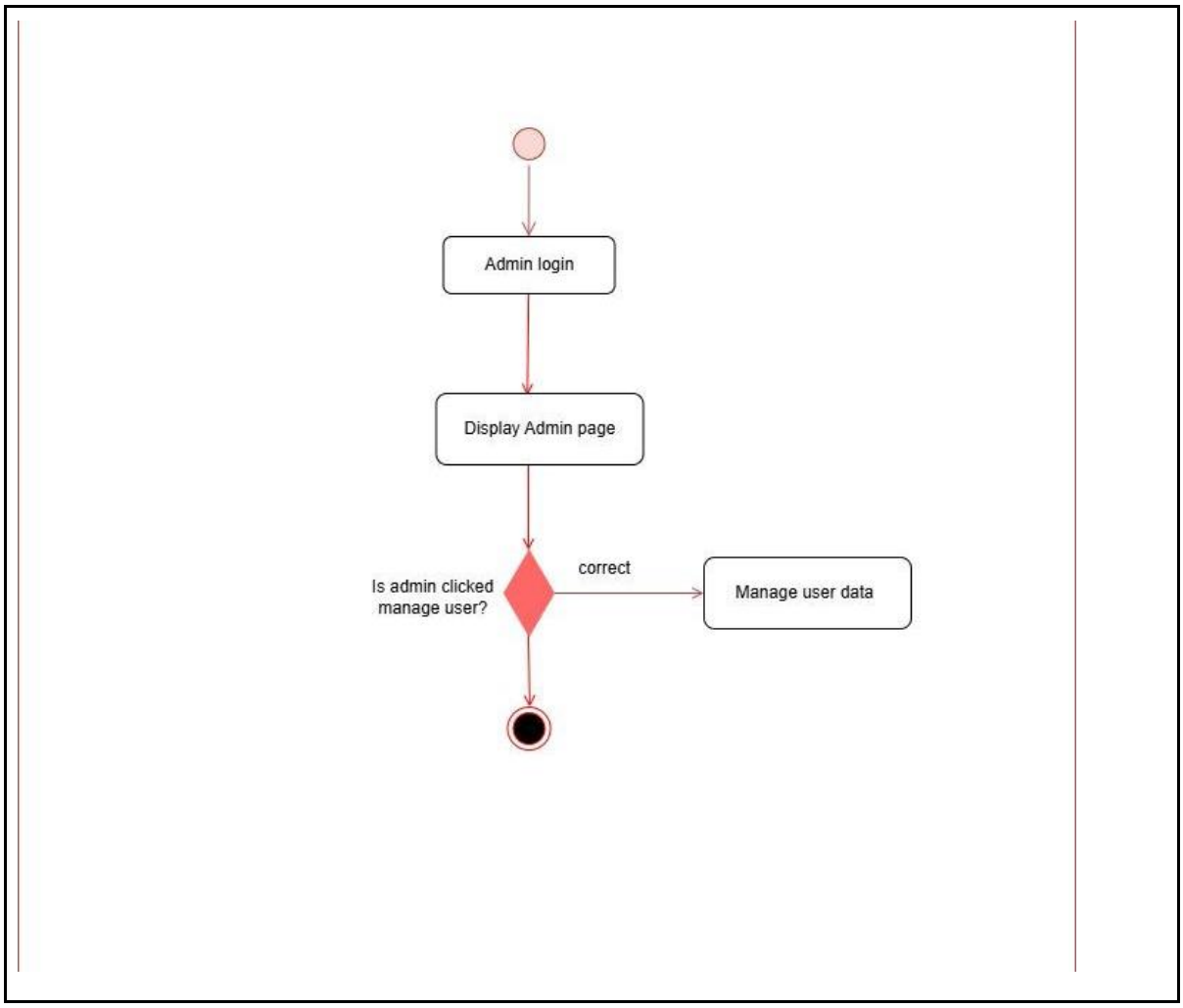


Figure 3.1.5.5 : Admin Manage User Activity Diagrams

- **Invalid Password**

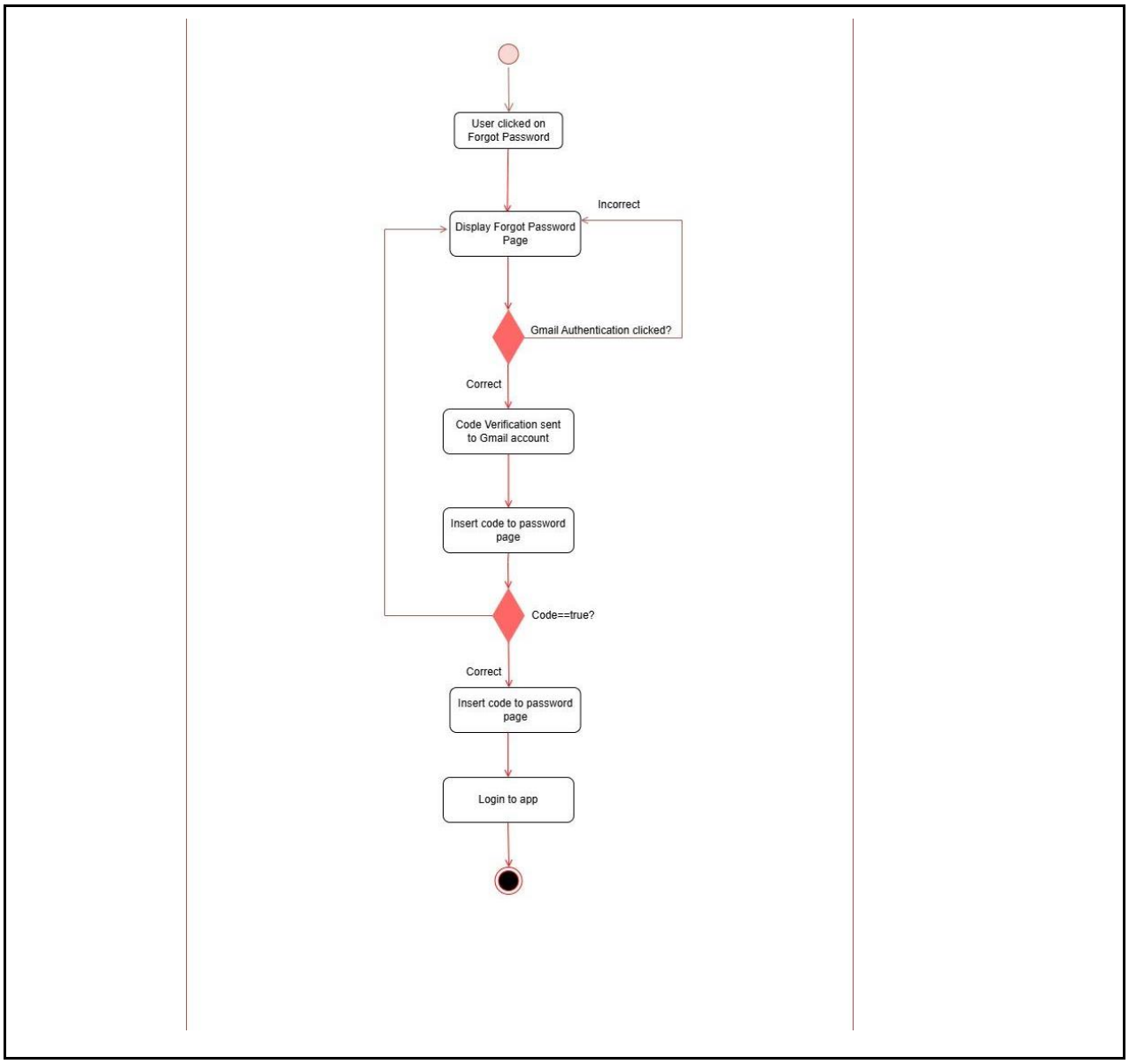


Figure 3.1.5.6 : Invalid Password Activity Diagrams

3.1.6 CLASS DIAGRAM

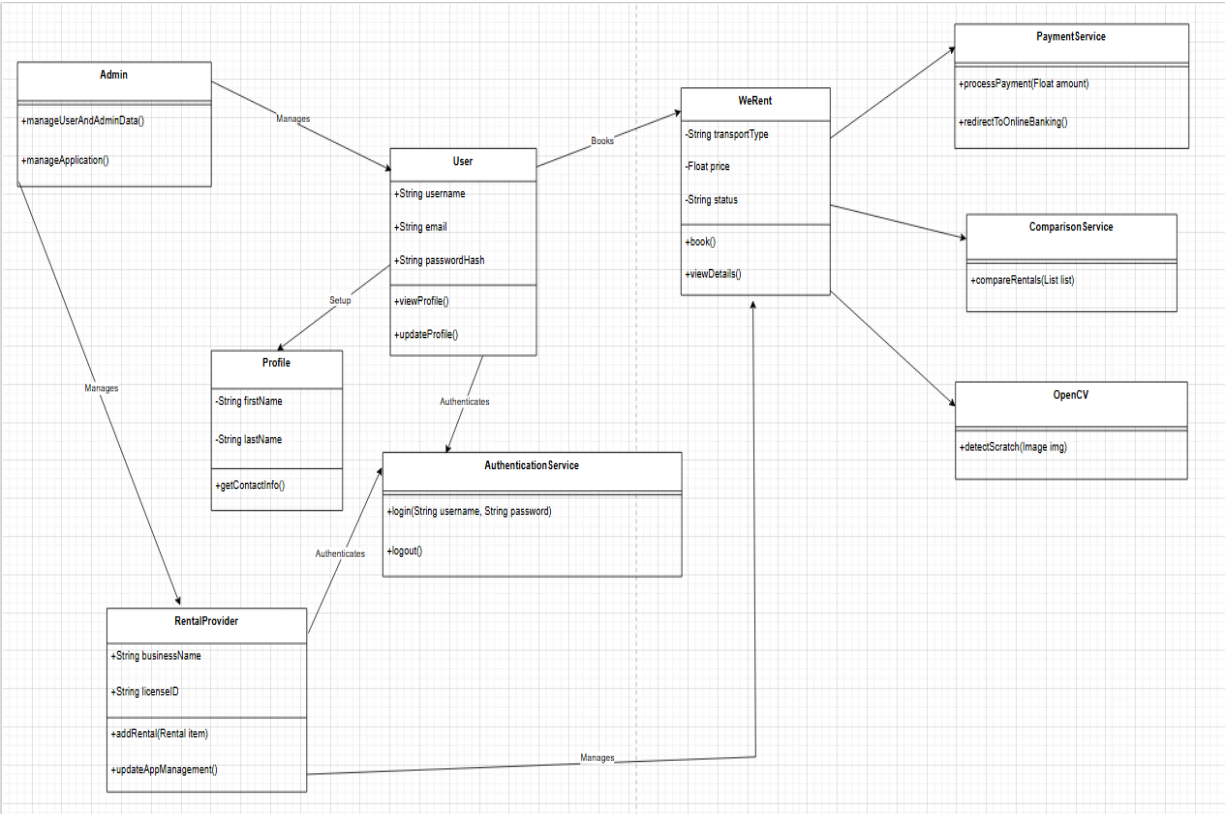


Figure 3.1.6.1 : WeGoo Class Diagram

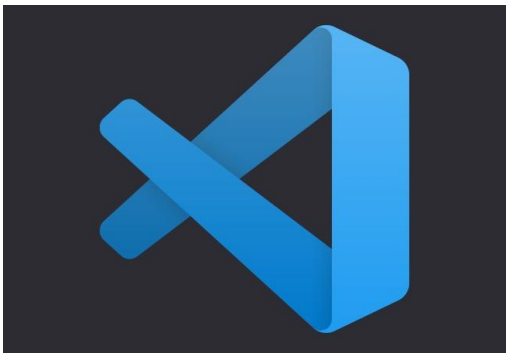
3.2 PROJECT SPECIFICATION

3.2.1 SOFTWARE SPECIFICATION

<table>
<tr>
<td> 
Figure 3.2.1.1 : Google Maps API</td>
<td>Descriptions :

• To add GPS features inside WeGoo application and desktop.</td>
</tr>
<tr>
<td colspan="2">Google Maps API</td>
</tr>
<tr>
<td> 
Figure 3.2.1.2 : OpenCV</td>
<td>Descriptions :

• To detect scratch on the vehicles.</td>
</tr>
<tr>
<td colspan="2">OpenCV</td>
</tr>
</table>

 *Figure 3.2.1.3 : Visual Studio Code (VS Code)*	**Descriptions :** • To implement coding proposed for WeGoo application
Visual Studio Code (VS Code)	
 *Figure 3.2.1.4 : Android Studio*	**Descriptions:** • To use android virtual devices during development phase on WeGoo application
Android Studio	

3.2.2 HARDWARE SPECIFICATION

<table>
<tr>
<td> 

Figure 3.2.2.1 : Iphone 13</td>
<td>Descriptions :

● To implement WeGoo mobile application in iOS operating system.</td>
</tr>
<tr>
<td colspan="2">Iphone 13</td>
</tr>
<tr>
<td> 

Figure 3.2.2.2 : MSI 27 Inch LED Flat Pro 4K</td>
<td>Descriptions :

● To display, test and implement WeGoo in desktop view.</td>
</tr>
<tr>
<td colspan="2">AOC U27B35 27 INC ULTRA HD IPS VIEW</td>
</tr>
</table>

<table>
<tr><td> 
Figure 3.2.2.3 : Mac Mini i7</td><td>Descriptions :

● To implement WeGoo desktop application in iOS and Android operating system..</td></tr>
<tr><td colspan="2">Mac Mini i7</td></tr>
<tr><td> 

Figure 3.2.2.4 : Fantech (SP42) X42</td><td>Descriptions :

● Accessory for PC.</td></tr>
<tr><td colspan="2">FANTECH (SP42) X42</td></tr>
</table>

<table>
<tr><td> 
Figure 3.2.2.5 : Redragon K552 Kumara</td><td>Descriptions :

● Accessory for PC.</td></tr>
<tr><td colspan="2">Redragon K552 KUMARA</td></tr>
<tr><td> 
Figure 3.2.2.6 : Jbl Portable Speaker</td><td>Descriptions :

● Accessory for commercial videos</td></tr>
<tr><td colspan="2">JBL PORTABLE SPEAKER</td></tr>
</table>

<table>
<tr>
<td> 
Figure 3.2.2.7 : Monitor Arm</td>
<td>Descriptions :

● Extension for monitor</td>
</tr>
<tr>
<td colspan="2">MONITOR ARM</td>
</tr>
</table>

3.2.3 DATABASE SPECIFICATION

 *Figure 3.2.3.1 :Firebase*	**Descriptions :** ● To save user data in the database. ● To observe user data.
Firebase	

3.2.4 DESIGN SPECIFICATION

<table>
<tr><td> 
Figure 3.2.4.1 : Figma</td><td>Descriptions :

● To design the WeGoo application.</td></tr>
<tr><td colspan="2">Figma</td></tr>
<tr><td> 
Figure 3.2.4.2 : Capcut</td><td>Descriptions :

● To create animation videos and demo for WeGoo applications.</td></tr>
<tr><td colspan="2">After Effect</td></tr>
</table>

3.3 PROJECT FEATURES

- Login & Sign Up (both admin and user)
  Allows both users and admins to create accounts and securely access the system with personalized roles and permissions.

- Booking System
  Enables users to easily select, reserve, and confirm vehicle rentals through a fast and streamlined booking process.

- OpenCV Powered
  Integrates advanced image-processing technology to automatically detect, analyze, and highlight vehicle scratches or damages. This feature enhances accuracy, reduces manual inspection errors, and provides users with a transparent and reliable way to assess vehicle conditions before and after rentals.

- Search and Filter
  Allows users to quickly find vehicles based on specific criteria such as type, price, availability, or features, making the selection process faster and more convenient.

- Build in Google Maps
  Integrates Google Maps to help users locate rental vehicles, view routes, and navigate to pick-up or drop-off points easily.

- Rental Options
  Provides users with a variety of car types and rental plans, allowing them to choose the option that best fits their needs and budget.

- Compare Table
  Allows users to easily compare multiple car side by side based on features, price, and availability to make informed rental decisions.

- Online banking
  Enables secure in-app payments and transactions, allowing users to pay for rentals conveniently without leaving the application.

- Contact for Provider
  Allows users to directly communicate with vehicle providers for inquiries, support, or special requests, ensuring smooth and effective communication.

3.4 PROJECT COSTING

Bil.	HARDWARE & SOFTWARE	PRICE	QUANTITY
1.	Iphone 13	RM1999	1
2.	Mac mini	RM0	1
3.	AOC U27B35 27 INCH	RM850	1
4.	SPEAKER JBL PORTABALE	RM165	1
5	MONITOR ARM	RM80	1
TOTAL		RM3094	

Figure 3.4.1

CHAPTER IV

TESTING METHODOLOGY

4.1 TESTING MODEL

4.1.1 SCRUM MODEL

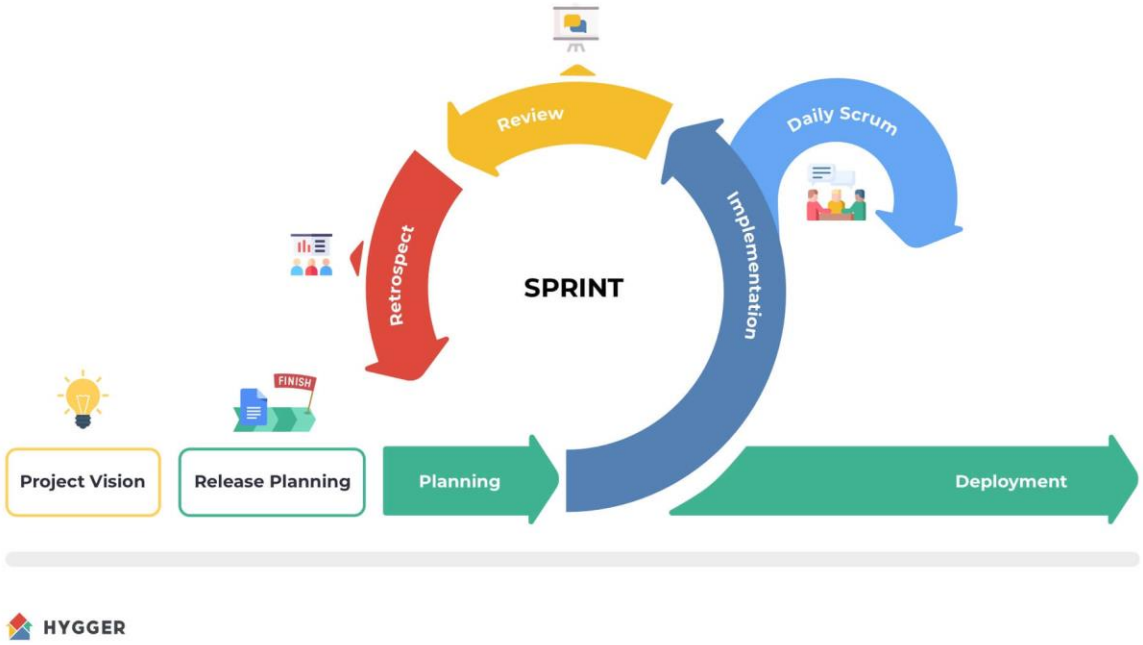


Figure 4.1.1.1 : Scrum Model

Scrum, an agile framework, integrates testing seamlessly within its iterative development cycles. It emphasizes continuous testing throughout the development process, ensuring high-quality deliverables. Test-driven development (TDD) and acceptance test-driven development (ATDD) are commonly employed within Scrum teams. Testing in Scrum is collaborative, with testers closely working with developers and stakeholders to validate and refine

product increments. Automation plays a vital role, enabling quick feedback and efficient regression testing within short iterations.

The Scrum framework offers distinct advantages:

- Accelerates time-to-market and enhances project success rates by fostering iterative, adaptive workflows.
- Early detection and resolution of issues through frequent inspection and adaptation, preventing them from escalating.
- Methodical progress and deployment planning through regular sprint cycles and sprint reviews.
- Ideal for managing medium-sized tasks by breaking them into manageable increments, promoting flexibility and responsiveness.
- Simplifies oversight with transparent goals and objectives set for each sprint, facilitating effective monitoring and adjustment.

4.1.2 FEATURE TO BE TESTED

Feature testing is important for providing customers with a fast, reliable, and secure online experience to ensure our application functions correctly and meets the specified requirements. There are several feature in WeeGoo that needs to be tested.

- Login/Sign Up system
  - To test validation username/password authentication
- Homepage
  - To test page load performance and responsiveness
- User Profile
  - To test viewing and editing personal details
- Search
  - To test result accuracy and relevance
- Settings
  - To test privacy settings
- OpenCV
  - To test the detection scratch feature.
- Rental Fee Calculator
  - To test the calculation accuracy
- Payment Gateway
  - To test supported payment methods
- Build in Google Maps
  - To test the user location detection

4.2 TESTING TOOLS

4.2.1 APPLICATION USE FOR TESTING

 *Figure 4.2.1.2 : Postman*	**Descriptions :** ● To test the WeGoo API application. Advantages : Here are some key **advantages of using Postman as a testing tool**: 1. **User-Friendly Interface** ○ Postman has a very intuitive and easy-to-use GUI, which makes creating, sending, and testing APIs fast even for beginners. **2. Detailed Reports and Logs** ○ It provides clear response outputs, status codes, response times, and even detailed logs, which are helpful for debugging. **3. Collaboration Features** ○ Teams can easily share collections, environments, and APIs, thanks to Postman's

	workspace and cloud features.
Postman	

4.2.2 TESTING TYPE

1. Functionality Testing

Functionality testing is considered one of the most critical types of software testing. It focuses on verifying that the application behaves according to the defined requirements and meets the intended business needs. Functionality testing ensures that all features and functions of the software work properly under normal and expected conditions. Our decision to include functionality testing in our application development process is based on the need to deliver a reliable and user-friendly product. We aim to thoroughly test each function of the application to identify any potential defects early, thereby improving overall system quality and minimizing issues during later testing phases.

2. Portability Testing

Portability testing is a critical aspect of software quality assurance, particularly when the goal is to ensure that an application functions seamlessly across various platforms, devices, and operating environments. It is an essential process to verify that the software can be deployed and utilized efficiently, regardless of the hardware or software configuration. The primary focus of portability testing is to assess the application’s adaptability and identify potential issues related to different operating systems, browsers, or network environments. Our decision to integrate portability testing into our development cycle is influenced by the necessity to ensure a consistent user experience across diverse platforms.

CHAPTER V

FINDING AND ANALYSIS

5.1 PROJECT MOCKUP

5.1.1 USER GUI SYSTEM PROPOSED

5.1.1.1

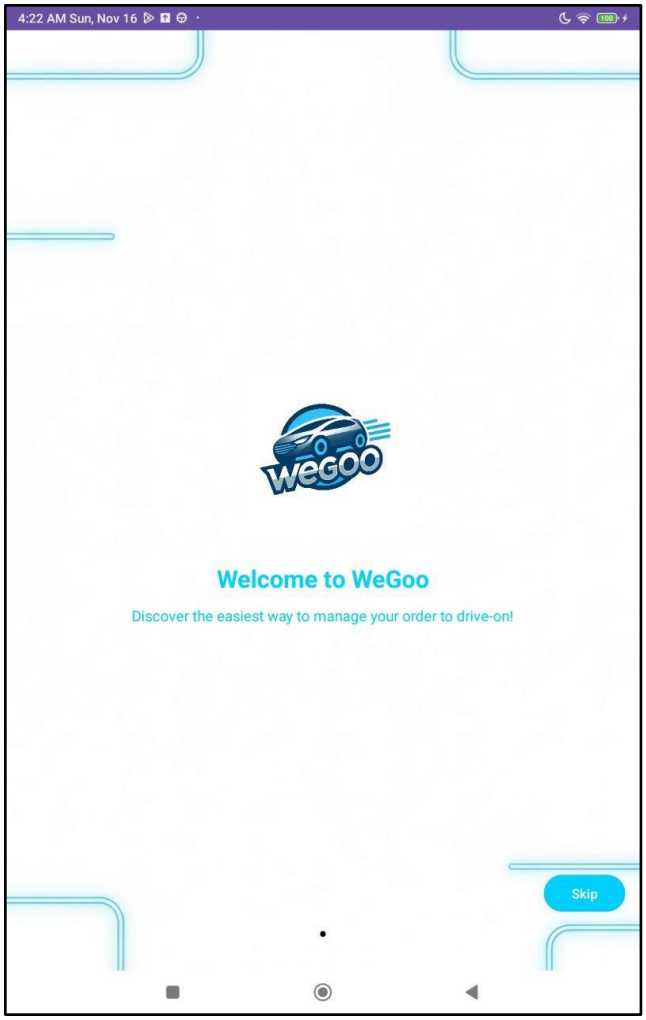


Figure 5.1.1.1 :

5.1.1.2 User Login

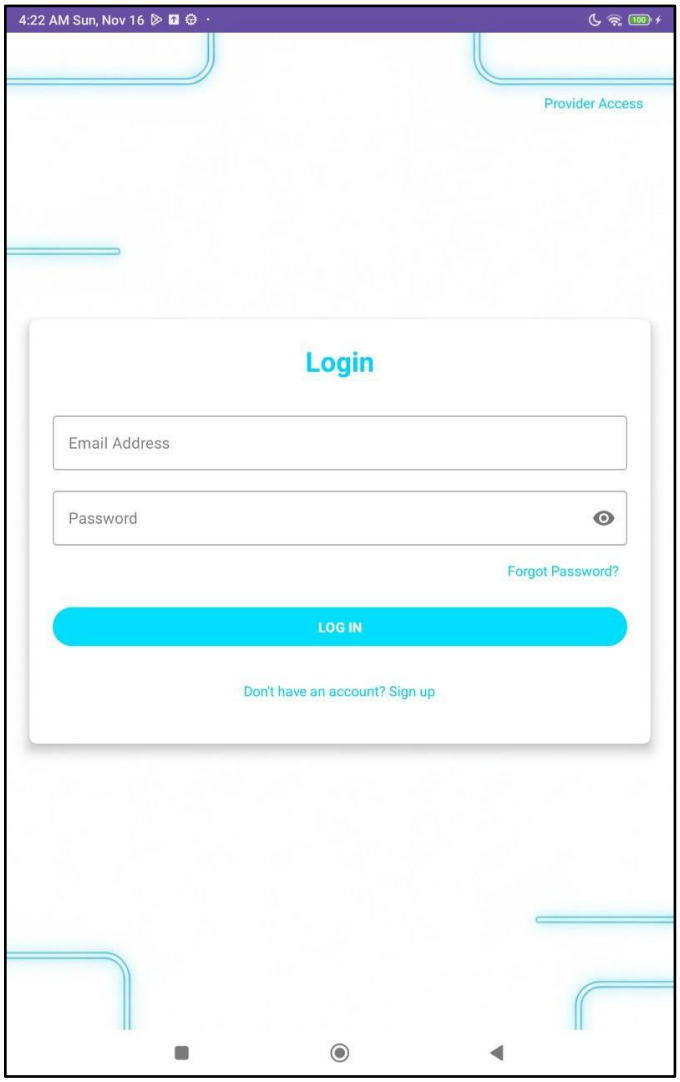


Figure 5.1.1.2 : User Login

5.1.1.3 User Sign Up

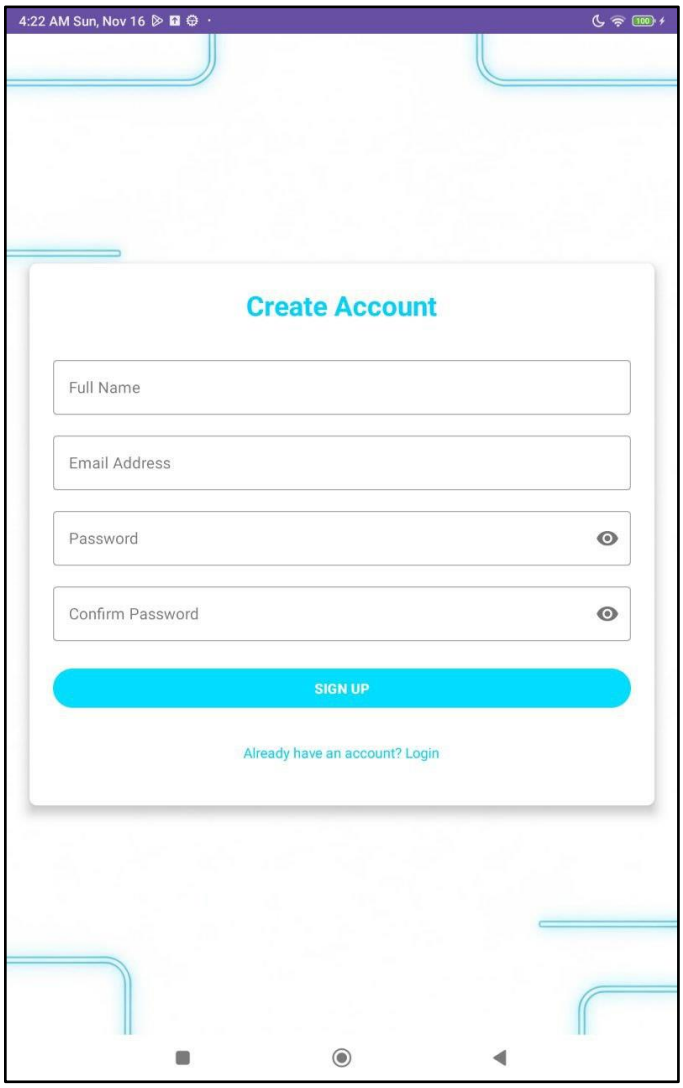


Figure 5.1.1.3 : User Sign Up

5.1.1.4 Payment Option

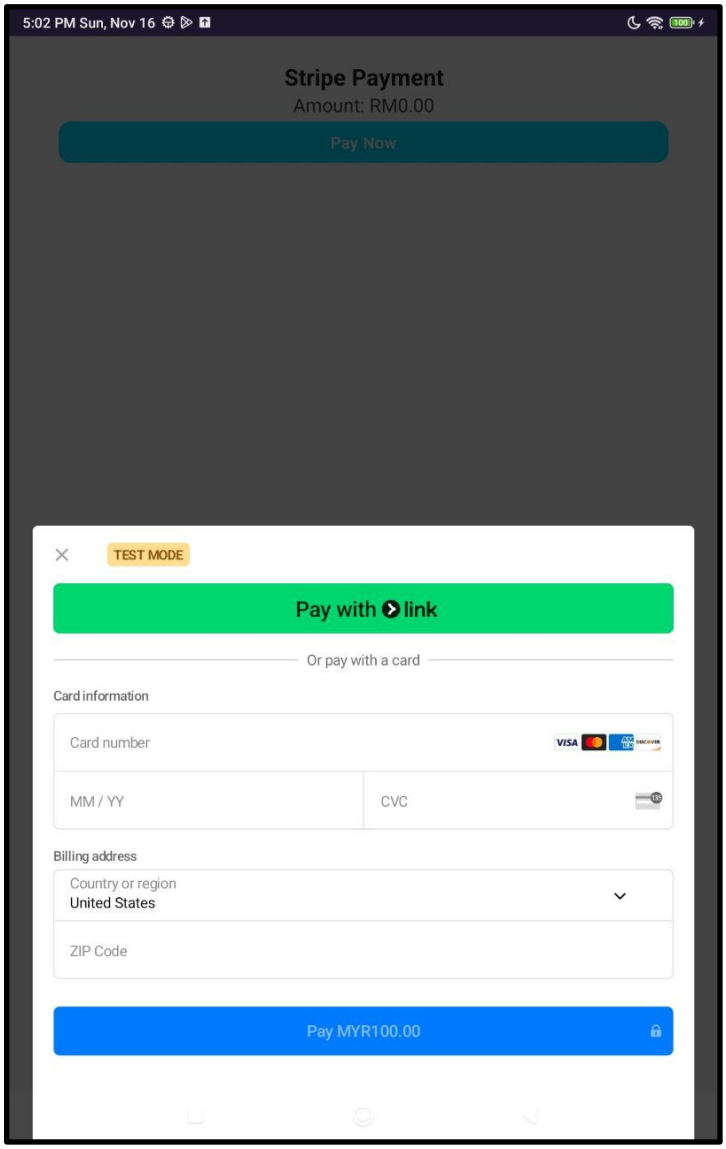


Figure 5.1.1.4 : Payment Option

5.1.1.5 OpenCV Detection

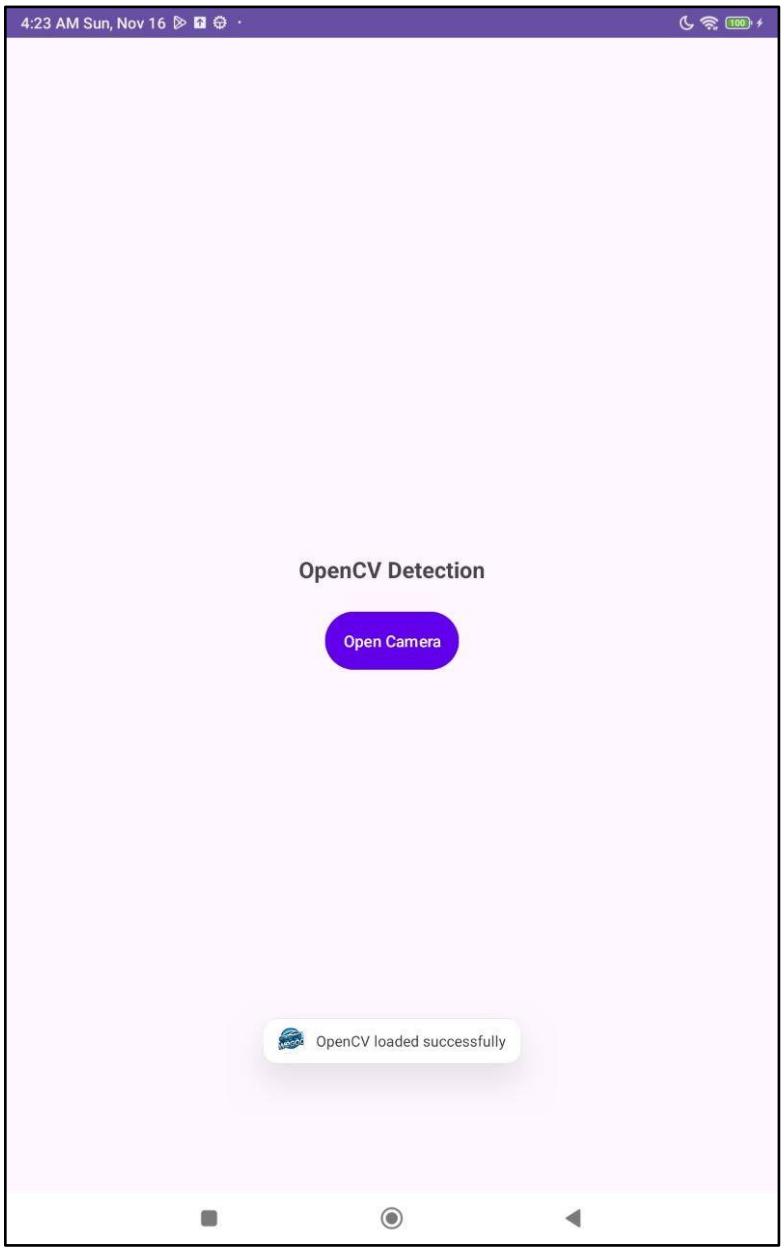


Figure 5.1.1.5 : OpenCV Detection

5.1.1.6 User Profile Page

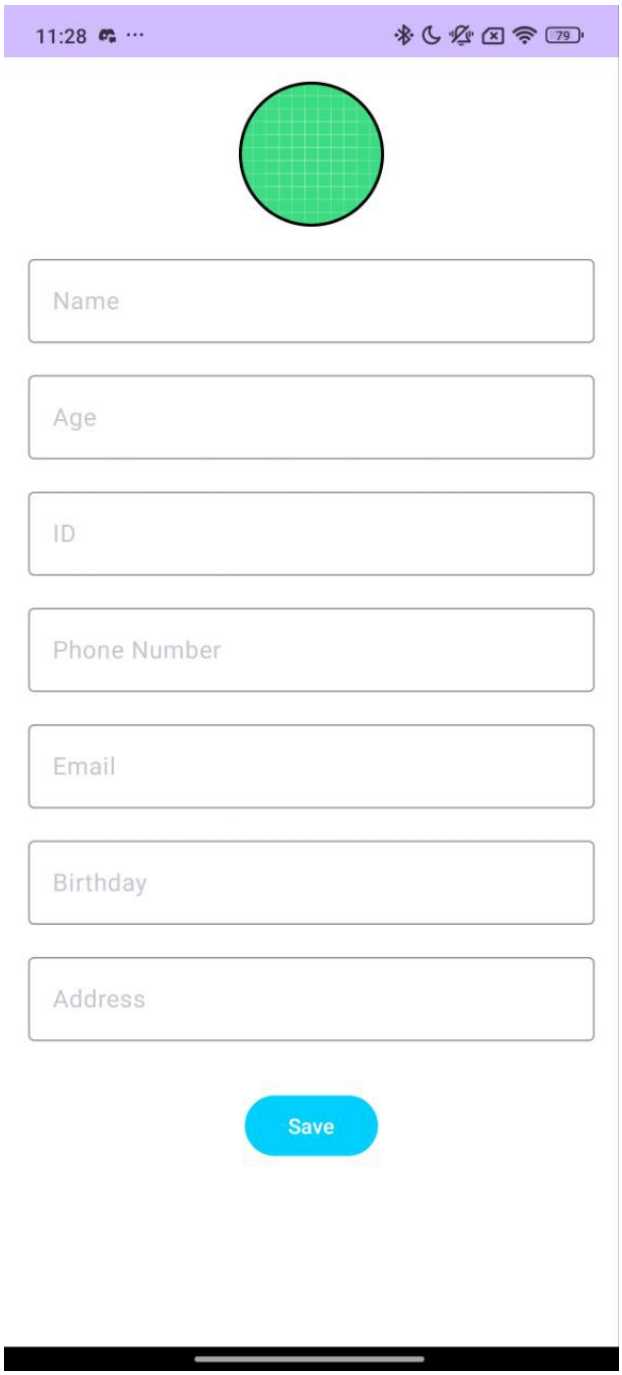


Figure 5.1.1.6 : User Profile Page

5.1.1.8 Payment Page

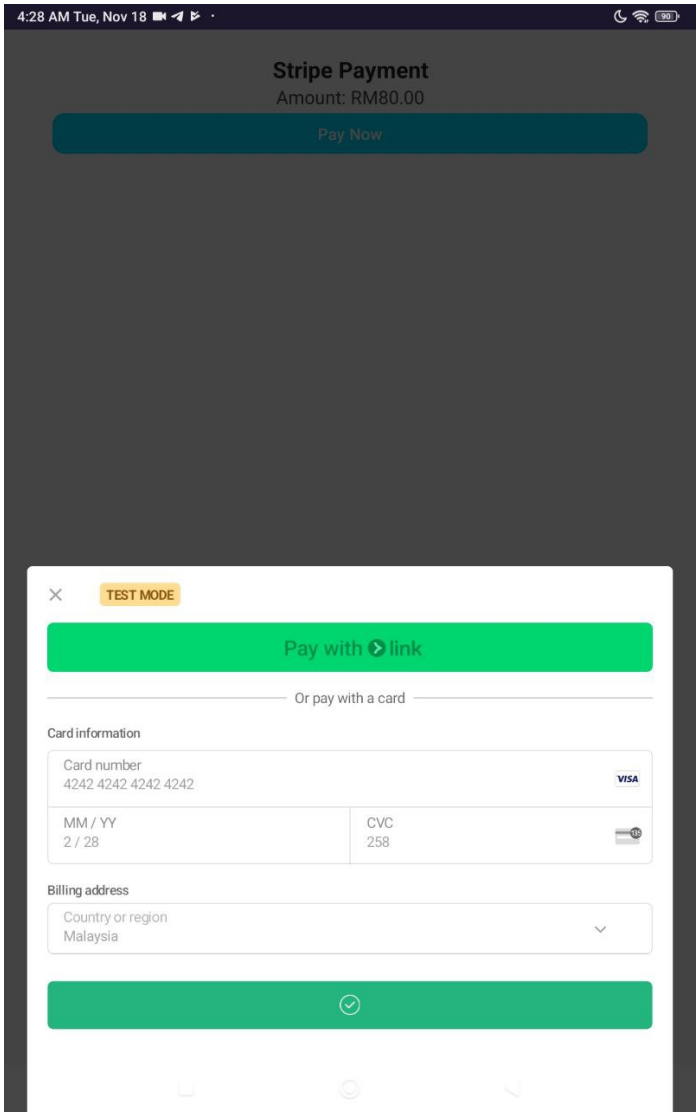


Figure 5.1.1.8 : Payment Page

5.1.2 ADMIN GUI SYSTEM PROPOSED

5.1.2.1 Login Admin Page

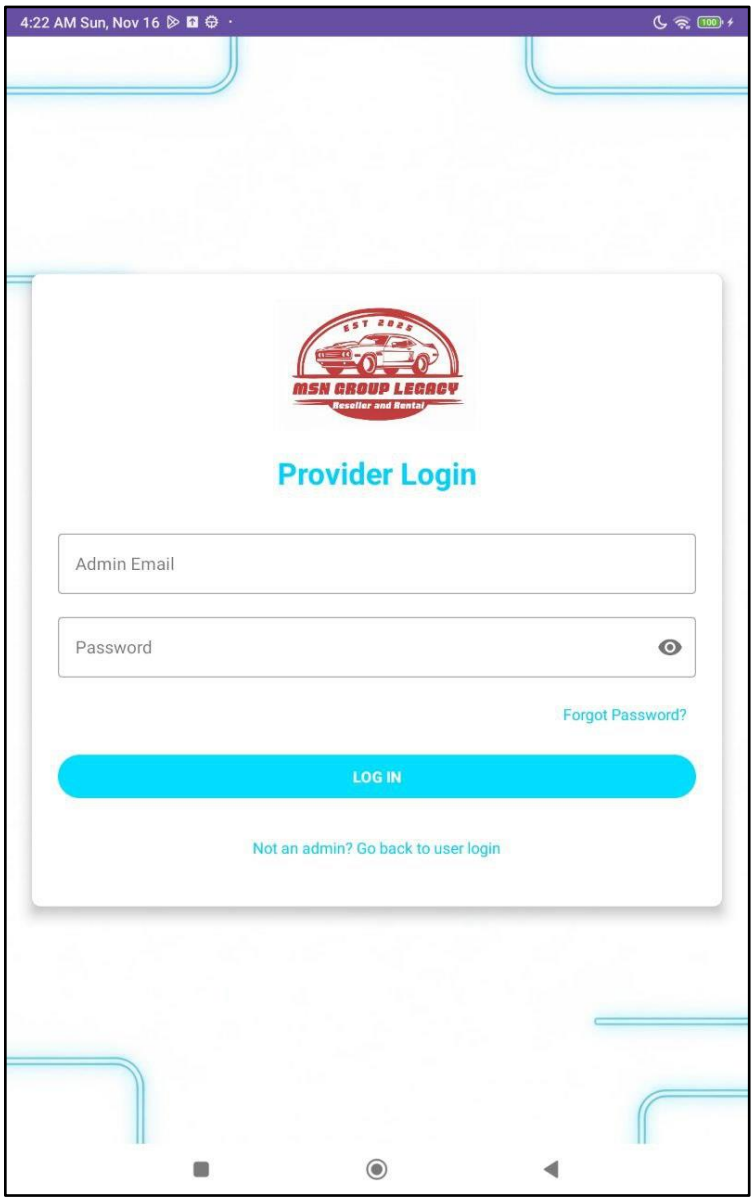


Figure 5.1.2.1 : Login Admin Page

5.1.2.2 Admin Mainpage

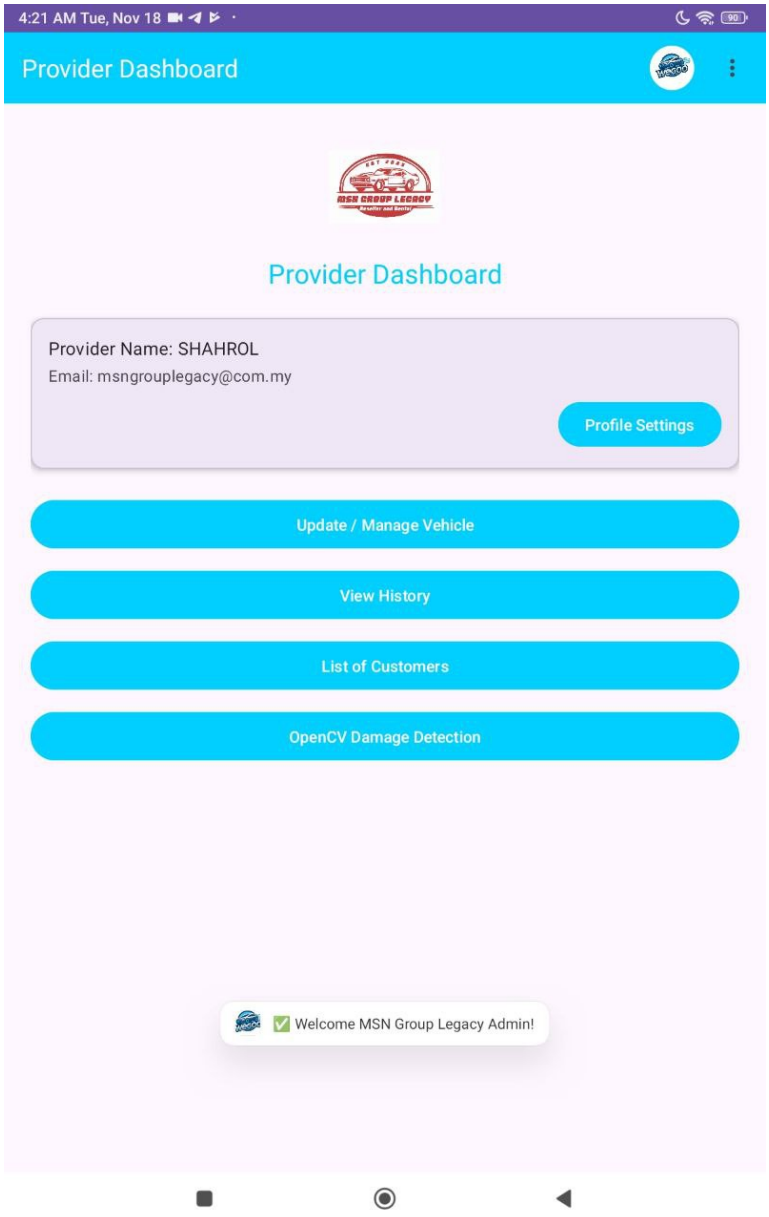


Figure 5.1.2.2 : Admin Mainpage

5.1.2.3 Add and Edit Vehicle Page Admin

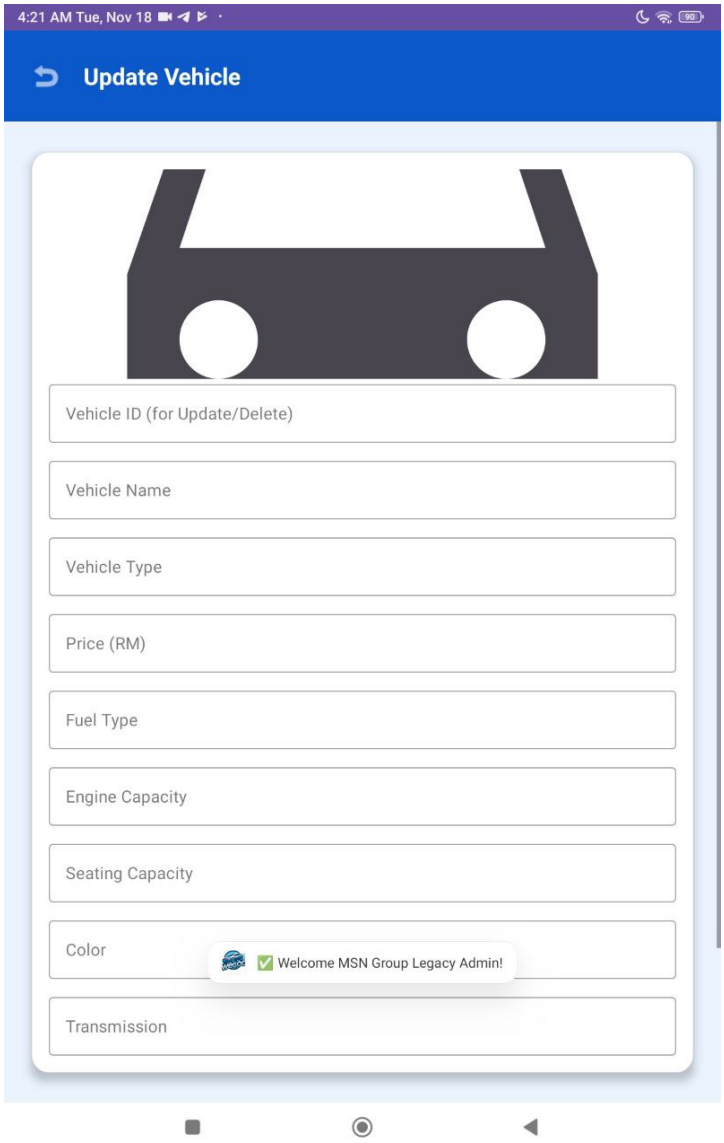


Figure 5.1.2.3 : Add and Edit Vehicle Page Admin

5.1.2.4 Sign Up Page for Admin and Users

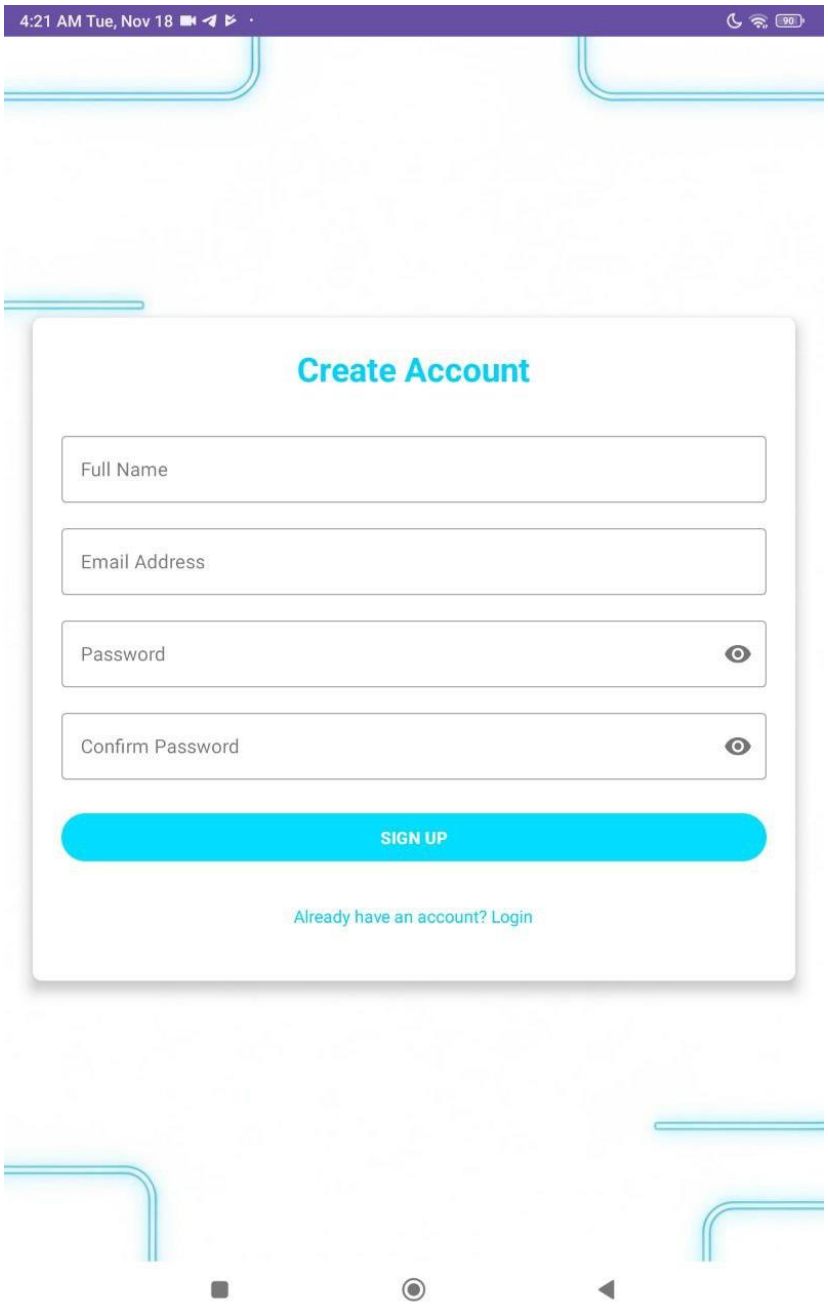


Figure 5.1.2.4 : Sign Up Page for Admin and Users

5.2 PROBLEM ENCOUNTER AND COUNTERMEASURE

The problem encountered that is fade by WeGoo and its countermeasure are as follow:

Problem Encountered	Countermeasure
Difficulty Detecting Scratches on Certain Car Colors (e.g., black or reflective surfaces)	Apply pre-processing techniques such as edge enhancement and grayscale conversion to improve the visibility of scratches on darker or reflective surfaces.
Communication Breakdown.	Maintain clear communication among team members, stakeholders and users.
Ensuring WeGoo works smoothly across multiple Android devices with different screen sizes and performance levels..	Optimize the app using responsive layout design, test across various Android screen resolutions using Android Studio device emulators, and implement adaptive UI scaling to maintain consistent performance and appearance on all Android devices.
Our stakeholder does not have any digitized system for their car rental services.	Introduce WeGoo as a simple digital platform to help the stakeholder manage bookings and vehicles more easily.

Figure 5.2.1

5.3 RESULT AND DISCUSSION

The result of the project mockup for WeGoo shows the interface for the rental app. The aim is to enhance users' experience and provide a friendlier user interface. All the standards of an application interface are successfully followed when designing our user interface for the app.

For instance, our home page shows directly the needs for users that include nearby providers across Malaysia. The home page also showcases the interaction menu, profile, logo, booking detail, and explore option.All the subsections on the main page can be clicked, making it more convenient for users to navigate through the app.

The admin page is provided with user management and personal customization information. It allows the admin to open their virtual rental shop inside the app.

Next, the built- in google maps really help users to find the location of the rental shop location. The features called WeCarculate in our WeGoo app also really can help users to calculate their budget and they will automatically get the recommendation for their needs. Last but not least, augmented reality (AR) is a feature to help users to determine the size of the car that they want to rent. Either it will fit in their car park or they can go through narrow roads.

Overall, the project mockup demonstrates a well-designed app that helps users find the vehicle they need and it also showcases key features, accessibility, and flexibility of the app, ensuring a positive user experience while using WeGoo.

CHAPTER VI

6.1 CONCLUSION

In conclusion, WeGoo represents a significant improvement in the way car rental services are accessed and managed. By addressing the core issues found in existing platforms such as inefficient booking processes, limited comparison features, and lack of reliable damage verification, we have successfully developed a system that enhances convenience, transparency, and trust for both users and rental providers.

Through the integration of essential features such as secure booking, user profile management, built-in Google Maps, and comparison tools, WeGoo offers a streamlined experience that simplifies the entire rental journey from start to finish. The OpenCV technology further strengthens the functionality of the system, allowing the app to accurately detect scratches and damages on vehicles. This not only improves fairness in the rental process but also reduces disputes between users and providers, ensuring a more transparent and reliable rental experience.

Additionally, WeGoo serves as a modern solution for stakeholders who previously relied on manual or non-digitized systems. By providing a centralized, easy-to-use platform, the app helps rental providers manage vehicles, bookings, and customer interactions more efficiently. This digital transformation supports better decision-making, reduces operational errors, and enhances overall service quality.

Overall, WeGoo successfully meets its objectives by offering a user-friendly, efficient, and technologically enhanced car rental application. With its strong features and improved scratch detection capabilities, WeGoo stands as a comprehensive system designed to support both renters and rental providers while contributing to a more organized and trustworthy rental ecosystem.

6.2 RECOMMENDATION

1. To achieve the outlined objectives and address the identified problems, the following recommendations are proposed:To achieve the outlined objectives and address the identified problems, the following recommendations are proposed:

2. Integrated Payment Gateway: A smooth and secure checkout process is important. If we offer various payment methods, this will increase the conversion rates.

3. Advanced Search and Filter: A good filtering system makes browsing much easier. It significantly improves the user experience and decision making to make decision faster.

4. User ratings and reviews: Future users are more likely to book if they can see honest reviews from past customers. Reviews also give us idea for improving service quality. Encourage verified renters to leave feedback by offering small support.

5. Modern UI/UX Design: Invest in a good designed interface. This will ensure the users interested to use our app with a good designed interface. A poorly designed interface can causes user to leave the application.

6. Insurance & Damage Reporting: This protects both the company and the customer. It is particularly useful for self-drive rentals where damage can occur.

REFERENCE

YouTube Videos:

1. Tokmakov, A. [@andretokmakov]. (2023, June 22). *Create a car rental application using flutte*r. YouTube. https://youtu.be/RKrWgdCUP1U

2. Bradski, G., & Kaehler, A. (2024). *OpenCV (Version 4.8) [Computer software]*. https://opencv.org/

3. Simplified Coding. (2021, February 21). *Android unit test tutorial – writing your first unit test* [Video]. YouTube. https://youtu.be/rM25aJ0o3Us

---

Websites and Tools:

4. Android Developers. (n.d.). *Android Studio*. Google. https://developer.android.com/studio

5. Microsoft. (n.d.). *Visual Studio Code*. https://code.visualstudio.com/

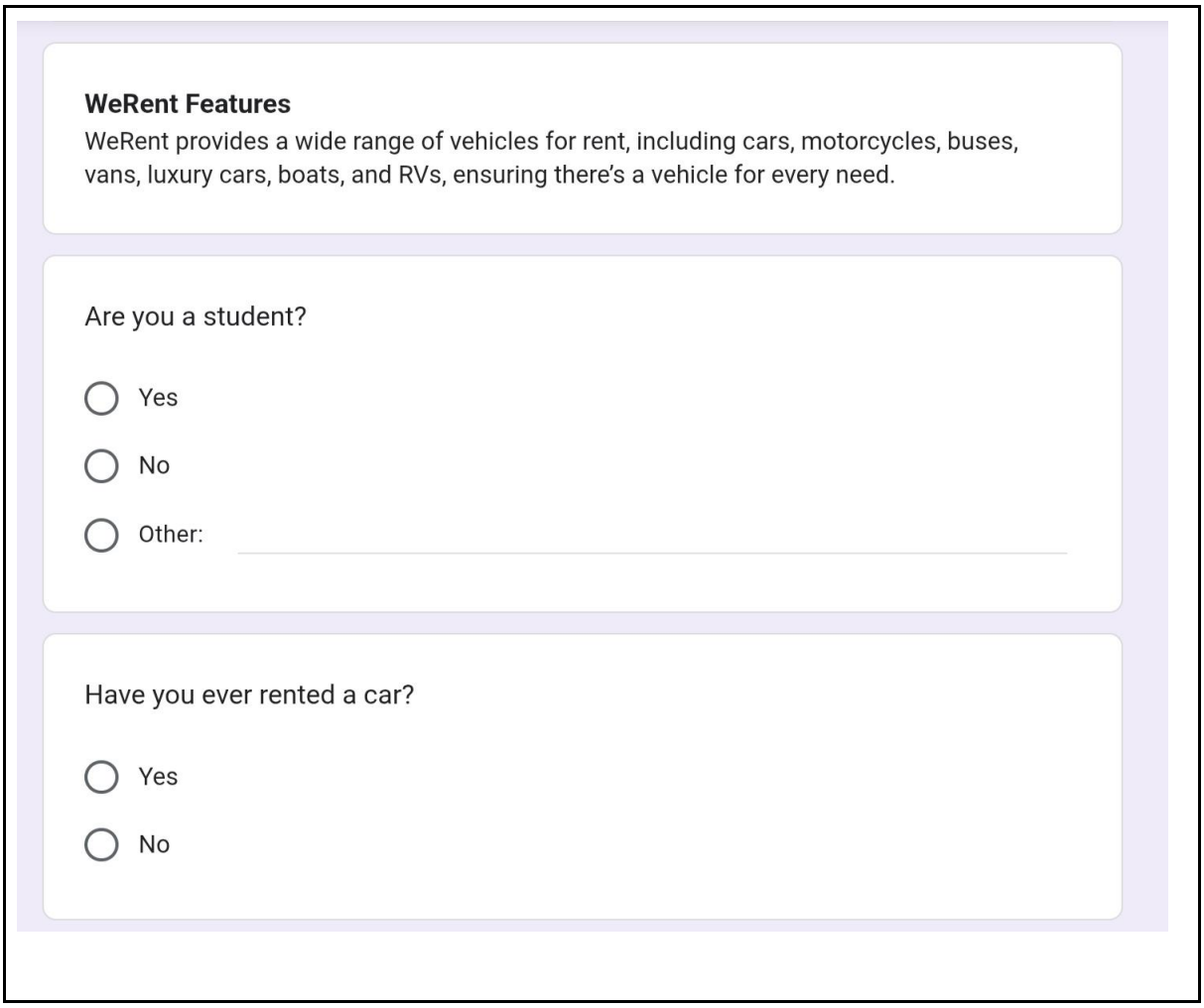
6. Google Maps Platform. (n.d.). *Maps APIs*. https://developers.google.com/maps

7. Google. (n.d.). *Flutter – Build apps for any screen*. https://flutter.dev/

8. OpenCV. (n.d.). *Open Source Computer Vision Library*. https://opencv.org/

APPENDIX A

- **SURVEY**



WeRent Features
WeRent provides a wide range of vehicles for rent, including cars, motorcycles, buses, vans, luxury cars, boats, and RVs, ensuring there's a vehicle for every need.

Are you a student?

- ◯ Yes
- ◯ No
- ◯ Other: _____

Have you ever rented a car?

- ◯ Yes
- ◯ No

Which platform do you use to rent a car?

- [ ] Application (CarSome,SOCAR,GoCar, etc)
- [ ] Website (carsewa.com,Klezcar.com,easybook.com,etc)
- [ ] Agent (WhatsApp,Telegram)
- [ ] Facebook
- [ ] Instagram

What would be your top priority when renting a vehicle?

- [ ] Variety of vehicles
- [ ] Ease of use
- [ ] Affordability
- [ ] Well-maintained

Did you find the platform is user-friendly?

Please **rate the platform below.You can answer more than one.**

	Application	Website	Agent (WhatsApp,Telegram)	Facebook	Instagram
1	☐	☐	☐	☐	☐
2	☐	☐	☐	☐	☐
3	☐	☐	☐	☐	☐
4	☐	☐	☐	☐	☐
5	☐	☐	☐	☐	☐

Augmented Reality (AR)

Augmented Reality (AR) is a technology that integrates digital information, such as images, text, or audio, into a user's view of the real world.

How effectively did the **Augmented Reality** (AR)feature allow you to visualize the vehicle's size, design, and features?

- ◯ Very Effective
- ◯ Effective
- ◯ Neutral
- ◯ Ineffective
- ◯ Very Ineffective

APPENDIX B

- **INTERVIEW STAKEHOLDER**

APPENDIX C

	FYP 2 Scrum Meeting	**30 days**	**Thu 10/7/25**	**Wed 20/8/25**
	Displaying project idea	1 day	Fri 18/7/25	Fri 18/7/25
	Submitting purchase	30 days	Thu 10/7/25	Wed 20/8/25
	Sprint 1: Documentating	**81 days**	**Fri 1/8/25**	**Fri 21/11/25**
	Chapter 1–3	22 days	Fri 1/8/25	Mon 1/9/25
	Chapter 4–6	25 days	Thu 18/9/25	Wed 22/10/2
	Sprint 2: Development	**40 days**	**Wed 23/7/25**	**Tue 16/9/25**
	Building the app	40 days	Wed 23/7/25	Tue 16/9/25
	Sprint 3: Testing	**20 days**	**Thu 18/9/25**	**Wed 15/10/2**
	Designing Menu (U	20 days	Thu 18/9/25	Wed 15/10/2
	Sprint 4: Maintenance	**8 days**	**Thu 23/10/25**	**Mon 3/11/25**
	Debugging	7 days	Thu 23/10/25	Fri 31/10/25
	Documentation Report	5 days	Thu 23/10/25	Wed 29/10/25
	Scrum Meeting FYP 2	**58 days**	**Wed 17/9/25**	**Fri 5/12/25**
	Progress Presentation 1	1 day	Wed 17/9/25	Wed 17/9/25
	Progress Presentation 2	1 day	Mon 20/10/25	Mon 20/10/25
	Internal Presentation 1	1 day	Tue 18/11/25	Tue 18/11/25
	Internal Presentation 2	1 day	Wed 19/11/25	Wed 19/11/25
	External Presentation	2 days	Tue 2/12/25	Wed 3/12/25