Software Requirements Specification

for

<CCRental>

Version 1.0 approved

Prepared by <Lim Ke En>

<Nanyang Technological University, Top G>

<9/3/2023>

Table of Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**

Lim Ke En	9/3/2023	Initial Write up	1.0
Lim Ke En	17/3/2023	Refine	1.1
Lim Ke En	21/3/2023	Add User Interface	1.2
Lim Ke En	23/3/2023	Refine use case description and sequence diagram	1.3

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

This Software Requirement Specification (SRS) document is intended for the **CCRental** web application, build version 1.0. The purpose of this SRS document is to describe the requirements specifications for the **CCRental** web application to facilitate the development and production process for all stakeholders. This document will include the system features, limitations, functional and non-functional requirements, as well as use case descriptions. The **CCRental** web application aims to help users to calculate the cheapest real time prices from different car rental platforms and allow users to make informed choices from there. This **CCRental** web application is built using the following frameworks, front-end framework being React JS, back-end framework being Node JS as well as the usage of MongDB as a database system.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

This section describes all standards and typographical conventions that we will be following when writing the SRS document.

Refer to appendix A for the list of definitions (Data dictionary) for special terms used during the description of our report.

This document follows the IEEE standards. Priorities of higher-level requirements are inherited by detailed-level requirements.

Font	Arial
Heading	Times, Bold, Size 18
Subheading	Times, Bold, Size 14
Text	Size 11

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This section will describe the different types of readers that the document is intended for and will provide suggestions to the sequence for reading the document.

This SRS document is meant for all stakeholders such as **CCRental** users, **CCRental** development team, **CCRental** testing team as well as **CCRental** project manager.

This document begins with the Introduction where it will state the purpose and product scope about the project. Thereafter, it will be followed by the overall description of the product which includes the application functionalities, several design constraints and assumptions of application. Then it will be followed by the introduction of both system functionalities and non-functional requirements of the product. This document will end with an appendix that encompasses the glossary and analysis models.

All stakeholders should begin reading the document from the introduction section - 1.1 Purpose, 1.2 Document Conventions as well as Appendix A (Data dictionary) to get a glimpse of what the product is about and have a brief introduction to the product.

The **CCRental** development team can then proceed with reading section 2. Section 2 will provide the development team with a high-level description of product perspective, functionalities that will be useful in helping the development team to build their product. After which, section 4 (System Feature) in which developers can have a better understanding about the functionalities and features of the application.

On the other hand, the **CCRental** users, testing team and project managers can read through this document in sequential order.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

This section will provide a short description of the product’s purpose, benefits, objective as well as goals.

In Singapore, car ownership and maintenance is a hustle and compared to the world, expensive too. As a result, car renting has become an economical way for Singaporeans to get the experience of owning a car for a period of time without all the hassles and financial burden that comes with car ownership. Also, car renting services have since become increasingly popular among Singaporeans families and young adults. Furthermore, with Singapore’s highly connected transport network, some Singaporeans may only require the use of a car for a certain period of time, hence allowing them to resort to rental cars instead of owning a private vehicle.

There are several growing platforms and companies where users are able to rent cars such as Getgo, CarLite and TribeCar. Each company has its own way of charging users with various different prices at different timings. As a customer, we are always on the lookout for the cheapest and best price in the market and the same goes for renting cars. Also, with the increasing cost of living in Singapore, Singaporeans are always sourcing for the cheapest service. However, comparing prices in rental cars is not always as easy as it seems, it requires the comparison of different applications and calculation of different prices.

As such, our team intends on bridging this gap by helping users calculate the cheapest real time prices from different car rental platforms to compare and choose what is best for the user.

The **CCRental** web application will come in handy in helping users to calculate their time needed for renting the car and recommending a range of cheapest rental cars. Users will be able to get the results of the rental car without needing to go through different websites when comparing the prices.

In addition, our target users of this application are residents of Singapore who would like to rent cars. People who have attained a driving license as well as families who need a car to travel during weekends are our target audience for this application.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

- Best Car Sharing Singapore: https://blog.moneysmart.sg/transportation/car-sharing-singapore-guide/#getgo-review
- React JS Documentation: https://reactjs.org/docs/getting-started.html
- Node JS Documentation: https://nodejs.org/en/docs
- MongoDB Documentation: https://www.mongodb.com/docs/
- Google Maps API: https://developers.google.com/maps/documentation/javascript
- GetGo Website: https://getgo.sg/
- CarLite Website: https://www.carlite.sg/
- TribeCar Website: https://www.tribecar.com/

2. Overall Description

2.1 Product Perspective

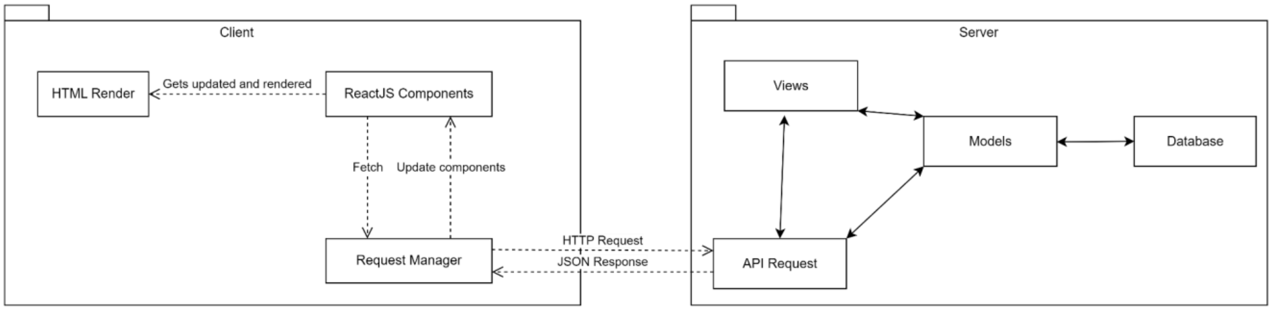
<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, ***self-contained product.*** *If the SRS defines a component of a larger system,*

relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

CCRental web application is a new, standalone, self-contained web application. It is a new web application that aims to help users to find the cheapest and nearest rental car services. The application can help to recommend an ideal location to pick up the car and the cheapest price according to the number of hours they need to rent the car for. The application also uses the Google Maps API to allow users to input their location in order to better estimate the amount of time needed for renting a car. The application will then sort the results based on the cheapest to the most expensive car renting service for users to refer to.

The application is inspired by applications that compare taxi prices and we hope to be able to create and programme the same functionalities for the renting services.

An overall system diagram depicting the operation of the application is as shown below:



2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

This section will summarise the major functions that the product must perform or must let the user perform.

The **CCRental** web application product functionalities will be broken down into three main sub-categories - Accounts, ComparePrice, CalculateDistance.

A. Accounts

○ Users can register for an account
○ Users can login to their registered account
○ Users can retrieve their login credentials
○ Users can recover their password in the case if users forgot their password.

B. ComparePrice

○ Users can input their preference of the options
○ Results will be filtered accordingly and displayed from the cheapest to most expensive rental service
○ Users can also view ratings of the different rental services
○ Users can save their favorite rental car into their account

C. CalculateDistance
○ Users can input all the locations they want to go
○ System can automatically calculate and provide an estimation of the amount of time they need to rent the car.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

This section will describe the user classes and characteristics of the applications that are based on frequency of use, subset of product function used, technical expertise, security or privilege levels. The main users for our application are People with driving licenses and also families who are on the lookout for car rental service.

A. People with driving licence

Attributes	**Description**
Frequency of use	High
Subset of product functions used	All
Technical expertise	Low
Characteristics	These are people who have attained a driving licence but do not have ownership of private vehicles. Hence, **CCRental** will come into good use where it allows them to obtain the cheapest rental car service from the platform.

B. Families who wants to rent car

Attributes	**Description**
Frequency of use	High
Subset of product functions used	All
Technical expertise	Medium

Characteristics	Having a car will increase convenience for family outings etc. Hence, families are more likely to rent a car for their family outing. Therefore, the system will provide recommendations of cars that are more pro-family and suited for their use.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

This section will provide the description of the environment in which the software will operate, including the hardware platform, operating system and versions together with other software components that must peacefully coexist.

Product Environment of **CCRental**:

The version of the web operating system must be at least _______ .

Development Environment of **CCRental**:

Development Environment	Description
Front-end: React.js	React.js framework is an open-source JavaScript framework and library developed by Facebook. It is used for building interactive user interfaces and web applications efficiently. React.js allows us to develop our application by creating reusable components. Hence, React.js is suitable to be used to design and build web applications. ReactJS (version 18.2.0)
Back-end: Node JS	Node.js is an open source server environment which is found on multiple platforms such as Windows, Linux, Mac OS etc. It is an asynchronous event-driven JavaScript runtime which is designed to build scalable network applications. NodeJS (version 18.15.0)
Database: MongoDB	MongoDB is a document database with the scalability and flexibility to allow querying and indexing the way developers want. MongoDB stores data in flexible, JSON-like documents. The document model maps to the objects in the application code allows for data to be easily utilised.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

- Limitations

Limitations	Details
Google Maps API	Time taken to retrieve location or data from Google Maps API might take longer than expected due to the use of free service.
Web Scraping limitations	

- Design Standards

Design Standards	Details
Programming standards	● Each frontend component needs to have its own folder that consists of .js file and .css file for better management of different components. ● Avoid deep nesting in codes. This will make it easier to follow and read as well as to debug. ● Use proper indentation to improve readability of code. ● Leave comments to describe code function so that it can guide other developers when working on the code. ● All non-class variables must adopt the camelCase naming conventions. ● All class variables must adopt the PascalCase naming conventions.
User Interface Standards	● All User Interface design must adopt the same colour and theme scheme throughout all the pages. ● All User Interface design needs to adhere to the pre-approved or considered user interface design.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

A live demo video on the flow of the application will be shown.

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

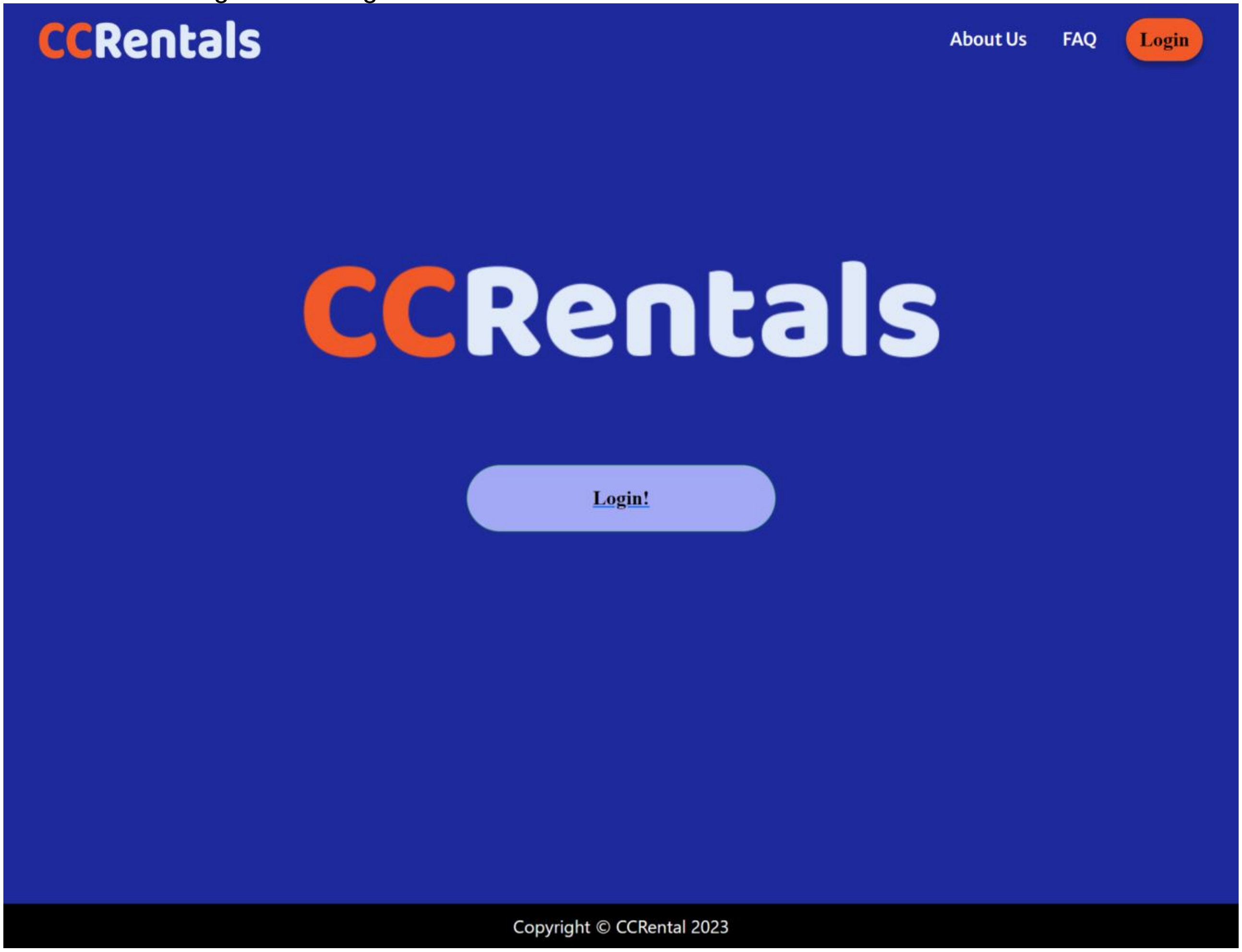
3. External Interface Requirements

3.1 User Interfaces

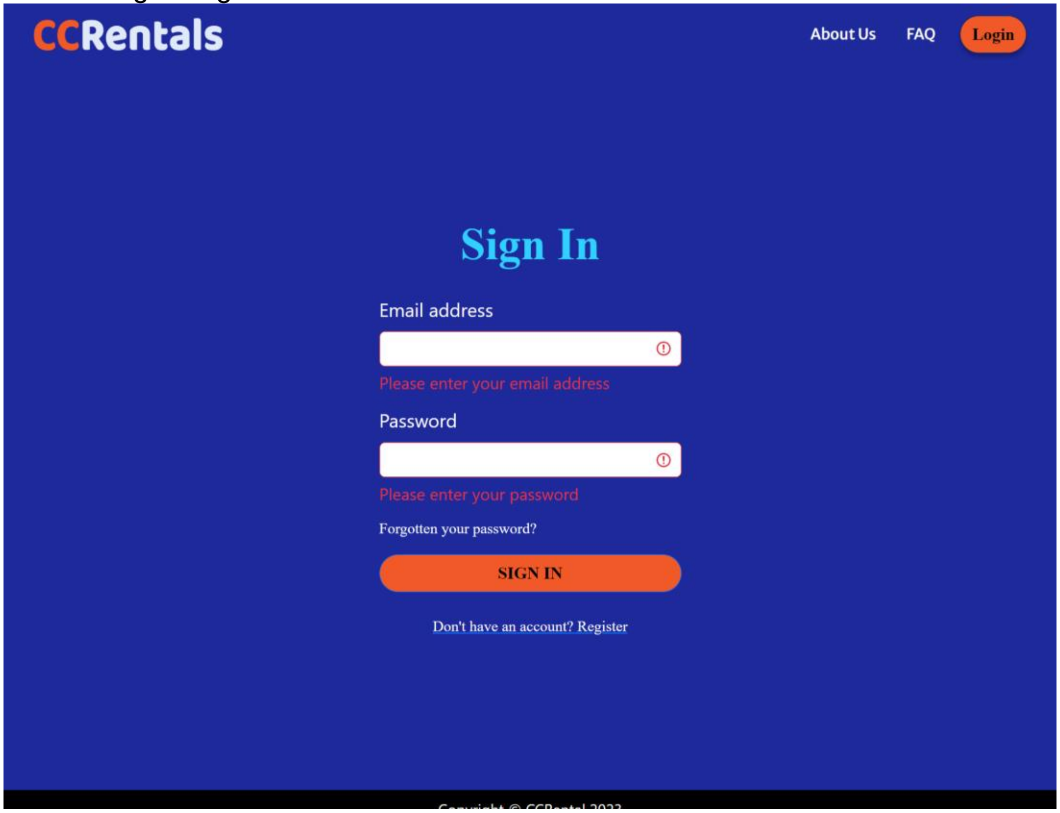
<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

This segment will provide a description of the logical characteristic of each interface. Sample screen images and user interfaces will be shown for better illustration.

A. Home Page before login



B. Login Page

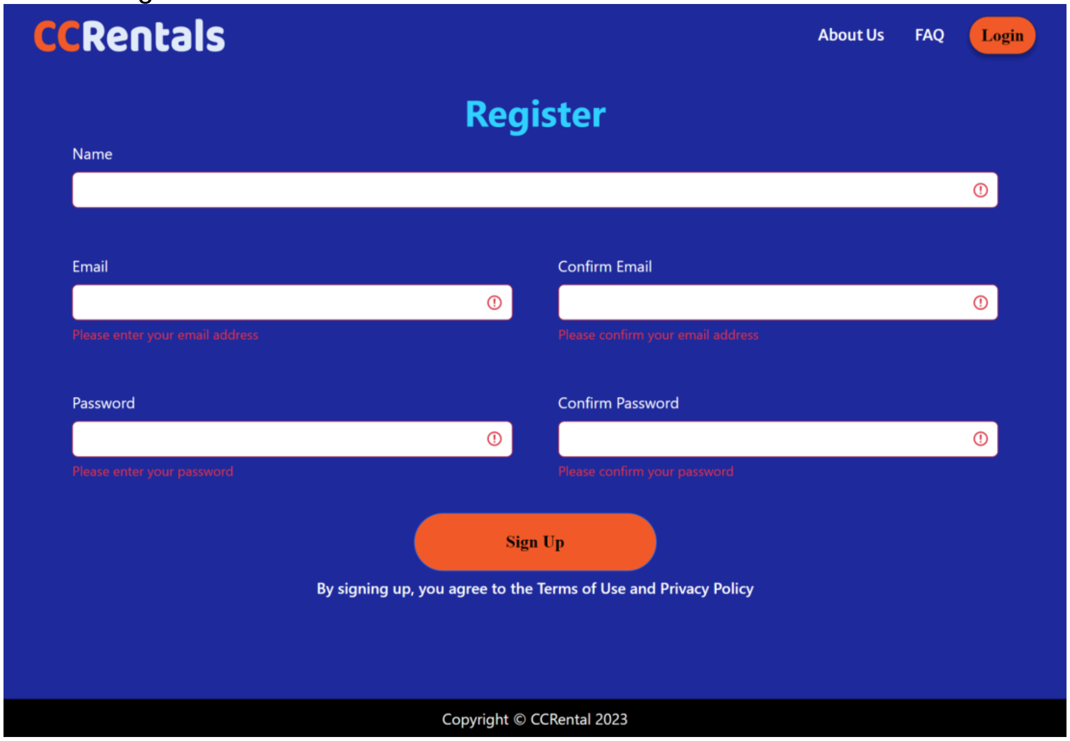


Please enter your email address

Please enter your password

Copyright © CCRental 2023

C. Register Account



Please enter your email address

Please confirm your email address

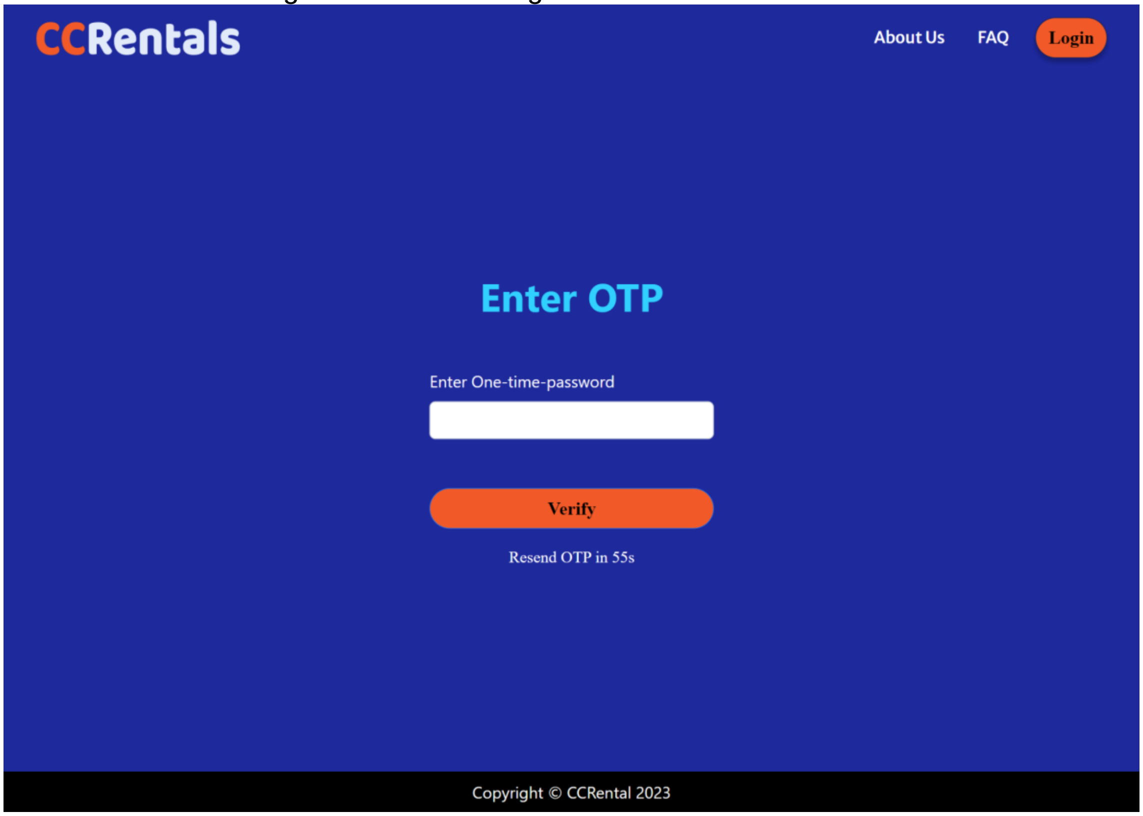
Please enter your password

Please confirm your password

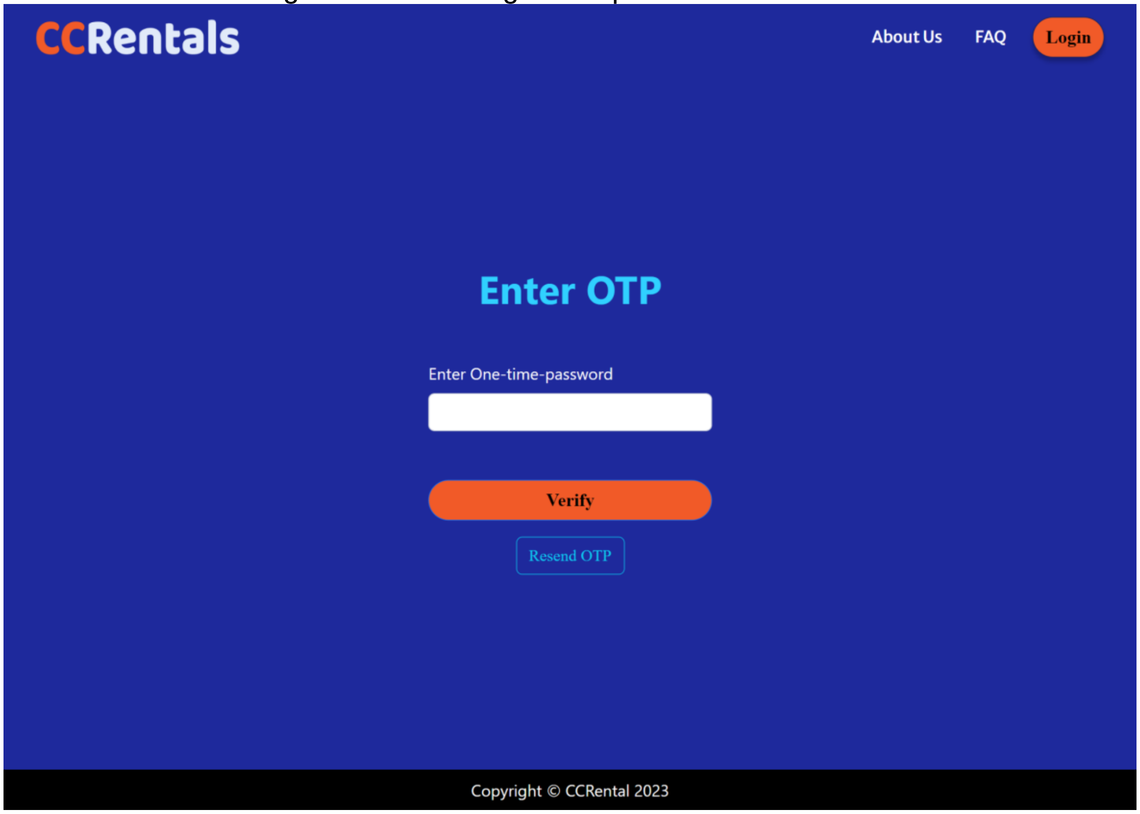
By signing up, you agree to the Terms of Use and Privacy Policy

Copyright © CCRental 2023

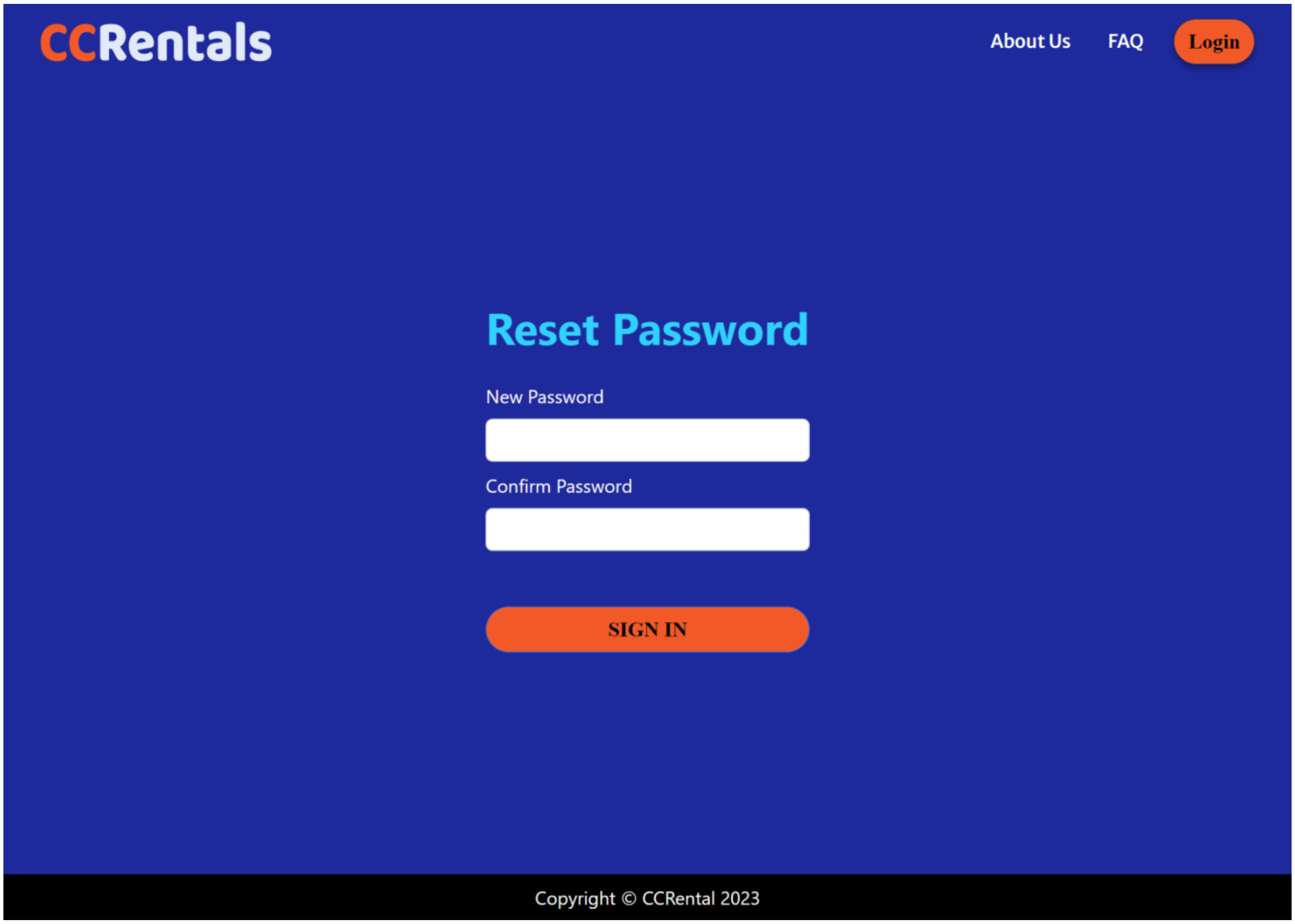
D. Enter OTP Page before resending OTP



E. Enter OTP Page after resending OTP option is available



F. Reset Password Page



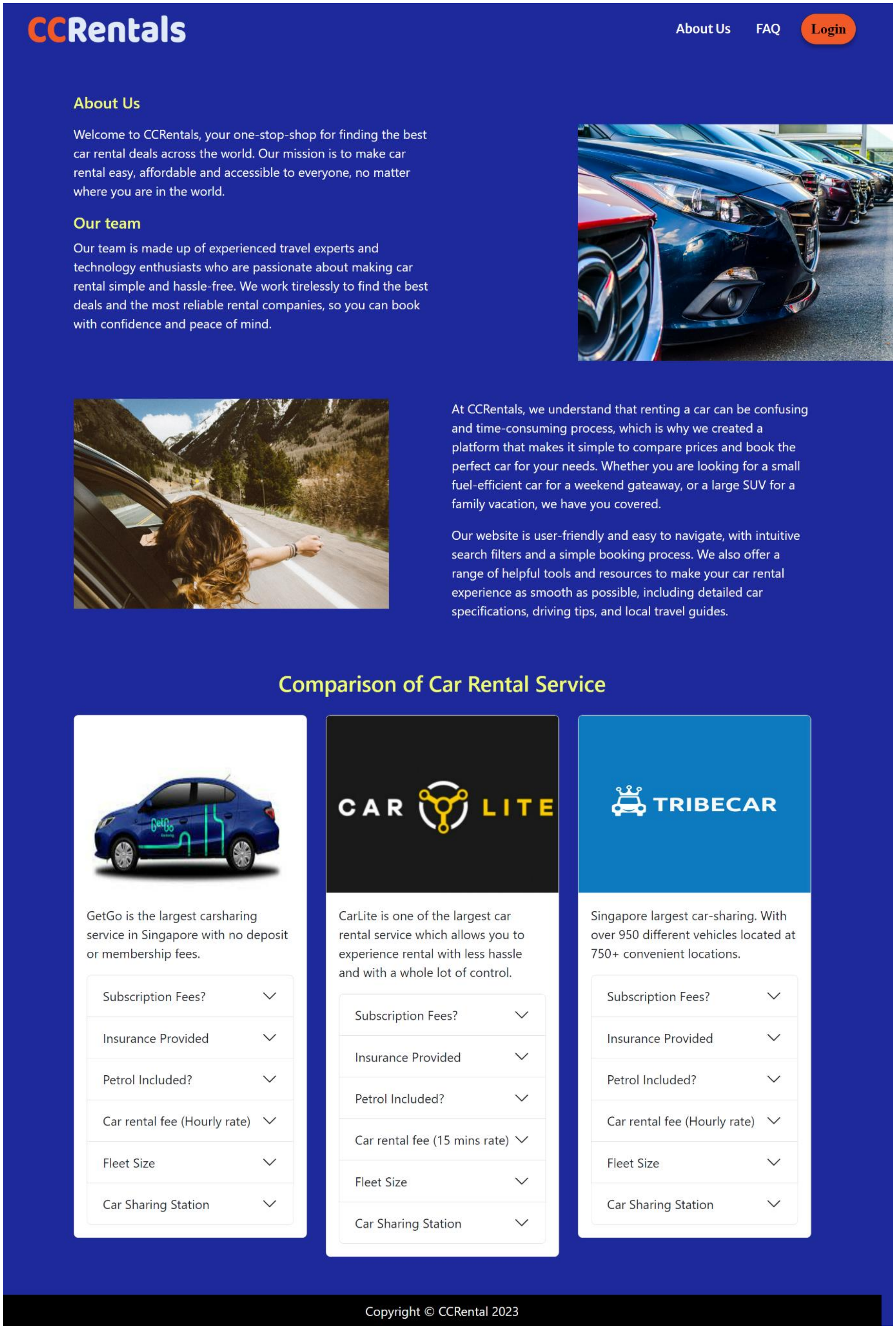
Reset Password

New Password

Confirm Password

Copyright © CCRental 2023

G. About Page



CCRentals

About Us FAQ Login

About Us

Welcome to CCRentals, your one-stop-shop for finding the best car rental deals across the world. Our mission is to make car rental easy, affordable and accessible to everyone, no matter where you are in the world.

Our team

Our team is made up of experienced travel experts and technology enthusiasts who are passionate about making car rental simple and hassle-free. We work tirelessly to find the best deals and the most reliable rental companies, so you can book with confidence and peace of mind.

At CCRentals, we understand that renting a car can be confusing and time-consuming process, which is why we created a platform that makes it simple to compare prices and book the perfect car for your needs. Whether you are looking for a small fuel-efficient car for a weekend gateaway, or a large SUV for a family vacation, we have you covered.

Our website is user-friendly and easy to navigate, with intuitive search filters and a simple booking process. We also offer a range of helpful tools and resources to make your car rental experience as smooth as possible, including detailed car specifications, driving tips, and local travel guides.

Comparison of Car Rental Service

GetGo is the largest carsharing service in Singapore with no deposit or membership fees.

- Subscription Fees?
- Insurance Provided
- Petrol Included?
- Car rental fee (Hourly rate)
- Fleet Size
- Car Sharing Station

CAR LITE

CarLite is one of the largest car rental service which allows you to experience rental with less hassle and with a whole lot of control.

- Subscription Fees?
- Insurance Provided
- Petrol Included?
- Car rental fee (15 mins rate)
- Fleet Size
- Car Sharing Station

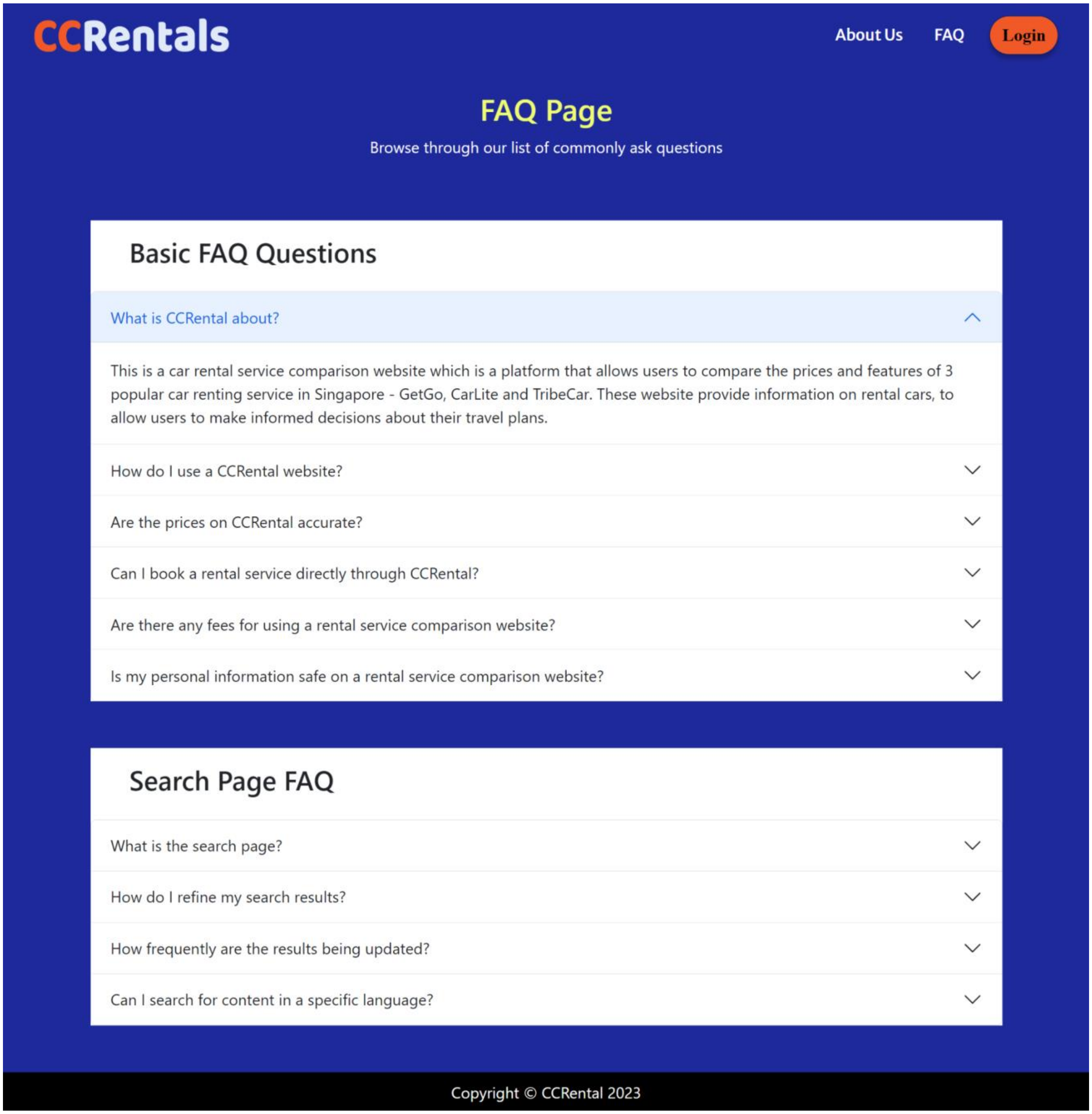
TRIBECAR

Singapore largest car-sharing. With over 950 different vehicles located at 750+ convenient locations.

- Subscription Fees?
- Insurance Provided
- Petrol Included?
- Car rental fee (Hourly rate)
- Fleet Size
- Car Sharing Station

Copyright © CCRental 2023

H. FAQ Page



FAQ Page

Browse through our list of commonly ask questions

Basic FAQ Questions

What is CCRental about?

This is a car rental service comparison website which is a platform that allows users to compare the prices and features of 3 popular car renting service in Singapore - GetGo, CarLite and TribeCar. These website provide information on rental cars, to allow users to make informed decisions about their travel plans.

How do I use a CCRental website?

Are the prices on CCRental accurate?

Can I book a rental service directly through CCRental?

Are there any fees for using a rental service comparison website?

Is my personal information safe on a rental service comparison website?

Search Page FAQ

What is the search page?

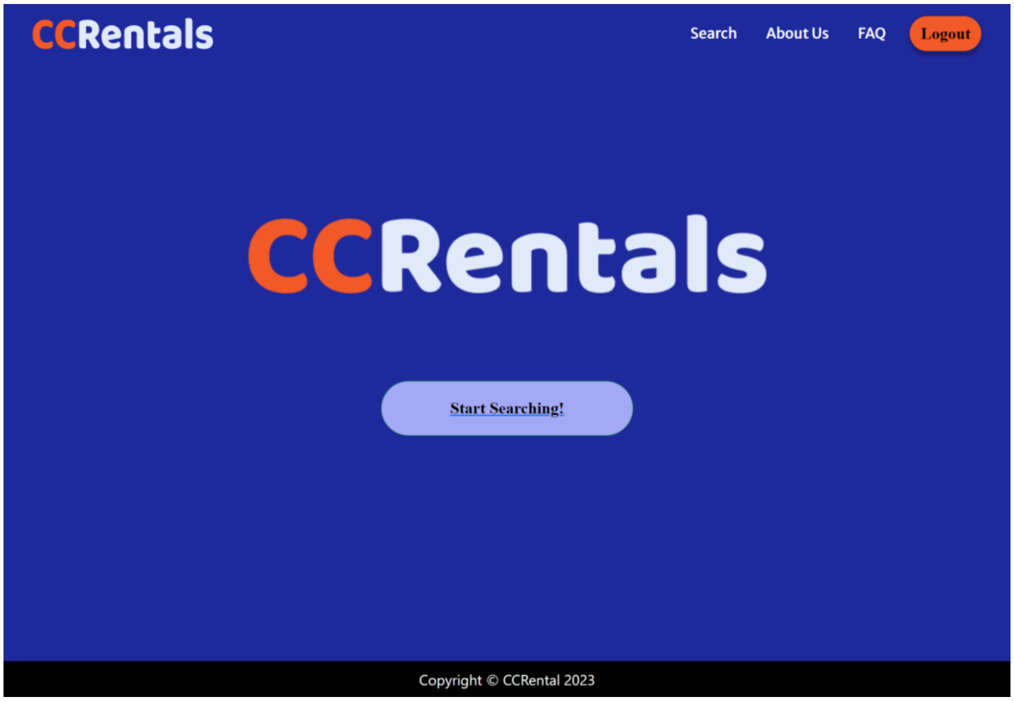
How do I refine my search results?

How frequently are the results being updated?

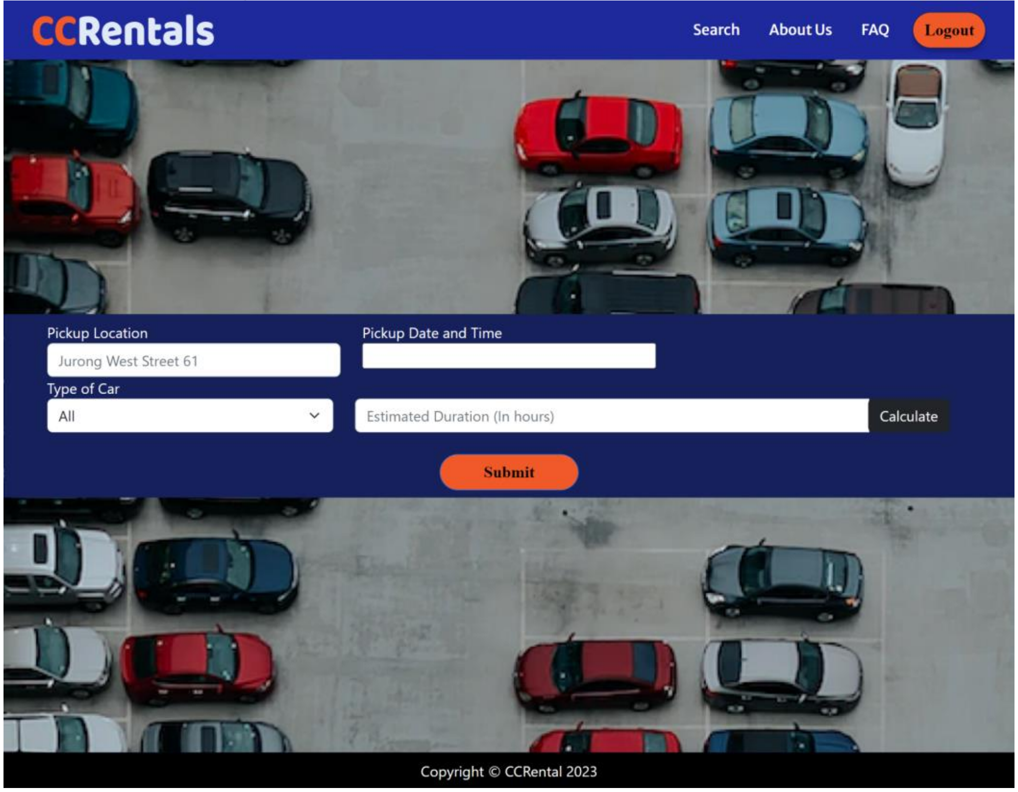
Can I search for content in a specific language?

Copyright © CCRental 2023

I. Home Page after user logs in



J. Search Page



Pickup Location

Jurong West Street 61

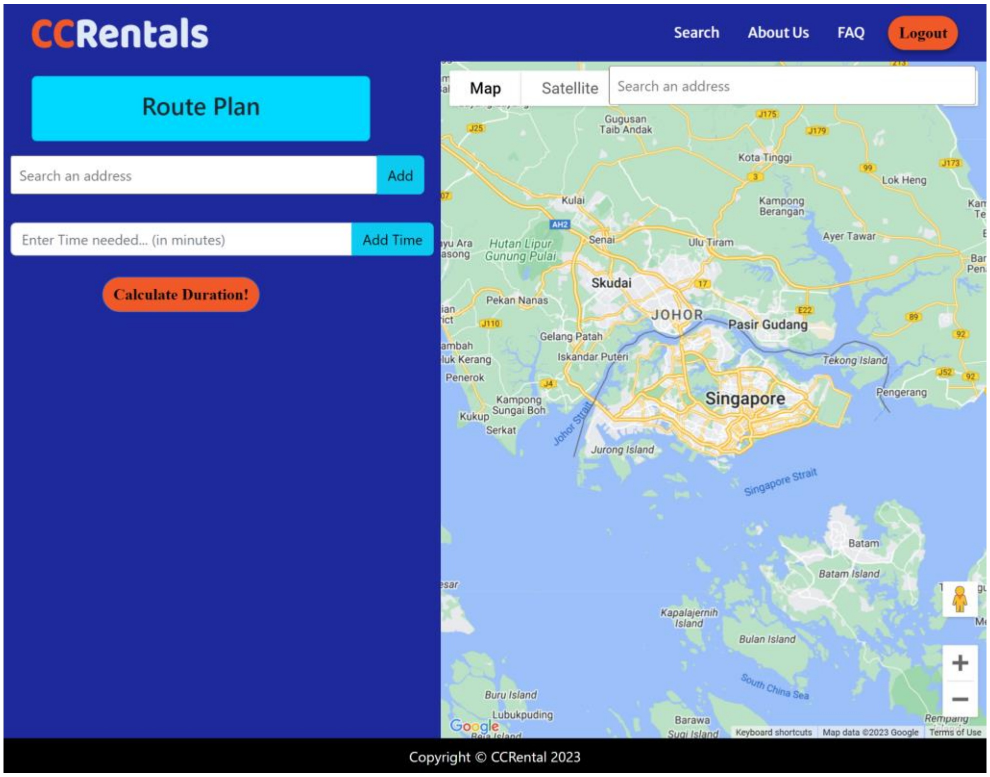
Pickup Date and Time

Type of Car

All

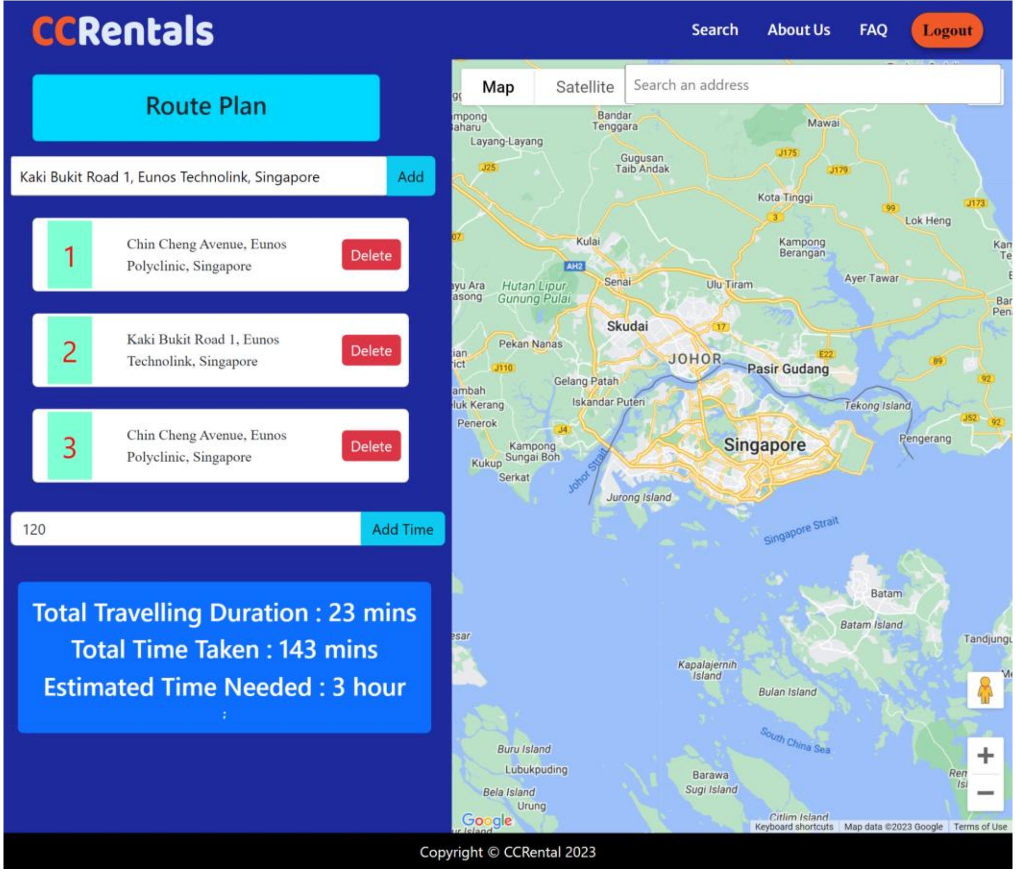
Estimated Duration (In hours)

K. Maps Page



Copyright © CCRental 2023

L.	Maps Page - After calculating estimated time



Kaki Bukit Road 1, Eunos Technolink, Singapore

1 Chin Cheng Avenue, Eunos Polyclinic, Singapore

2 Kaki Bukit Road 1, Eunos Technolink, Singapore

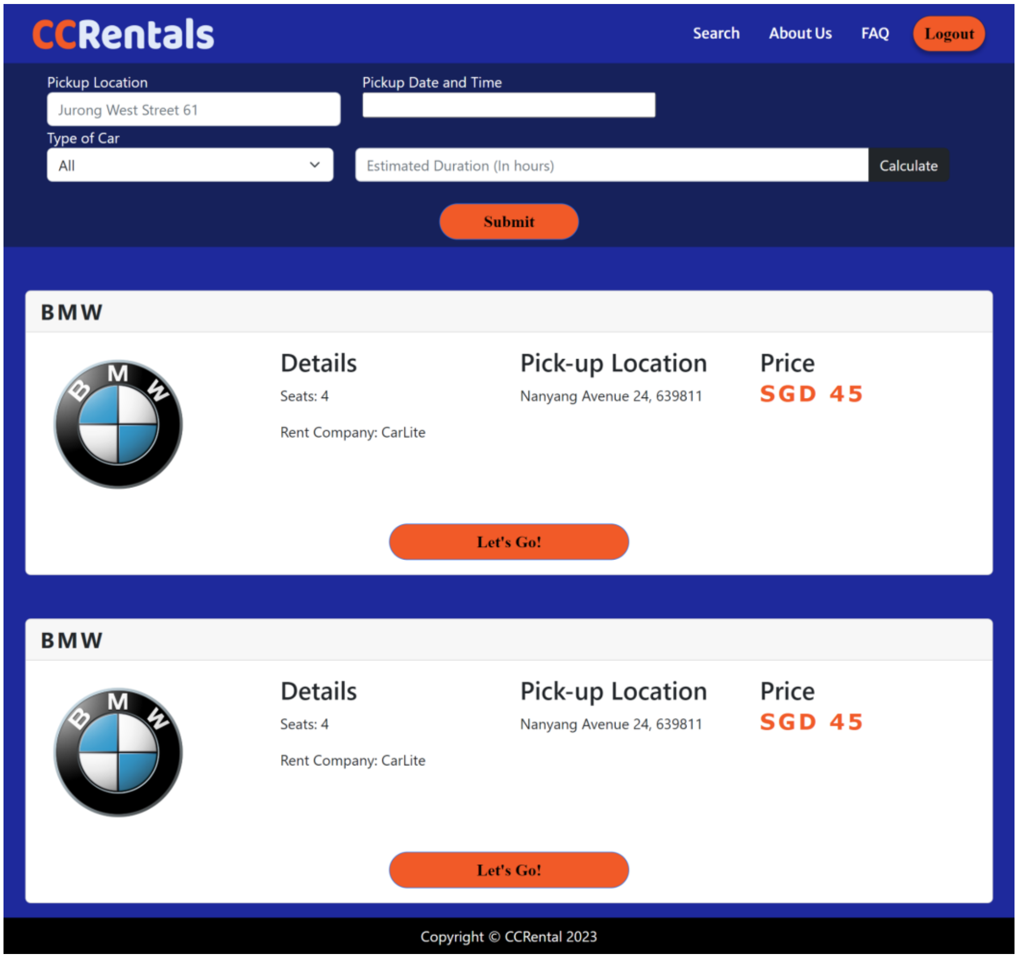
3 Chin Cheng Avenue, Eunos Polyclinic, Singapore

120

Total Travelling Duration : 23 mins
Total Time Taken : 143 mins
Estimated Time Needed : 3 hour

Copyright © CCRental 2023

M.	Results Page



Pickup Location: Jurong West Street 61

Pickup Date and Time:

Type of Car: All

Estimated Duration (In hours)

BMW

Details
Seats: 4
Rent Company: CarLite

Pick-up Location
Nanyang Avenue 24, 639811

Price
SGD 45

BMW

Details
Seats: 4
Rent Company: CarLite

Pick-up Location
Nanyang Avenue 24, 639811

Price
SGD 45

Copyright © CCRental 2023

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

This section will cover all hardware interface requirements for the CCRental application to achieve its desired functionalities. This includes the supported device types that are needed, the nature of data and control interactions between the software and hardware as well as communication protocols being used.

A. Client-side Requirements

The **CCRental** web application will support all desktop computers or laptops as well as being compatible with web services on smartphones. This device must support the usage of a desktop browser such as GoogleChrome or Firefox.

B. Server-side Requirements

The **CCRental** backend server must be hosted and run on a server-computer. The backend server will perform basic functionalities such as Create, Read, Update, Delete (CRUD) operations as well as connection using REST API with the client service on the back-end database. The database must be connected and hosted on the local computer.

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

This segment will describe the software interfaces.

A. Software Components

The **CCRental** web application is using MongoDB as a back-end database to handle all the querying of data in the web application. The back-end server is implemented using the node.js framework which will help to perform functionalities, the front-end that is implemented using React JS and are connected using Rest API.

B. Software Architecture

The **CCRental** software architecture is following the Model-View-Controller design pattern.

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronisation mechanisms.>

This segment will discuss all the associated communication interfaces for this web application to be working.

The **CCRental** web application is following the REpresentational State Transfer which is an architectural style for allowing easier communication between systems. This allows the implementation of client and server to be done independently. In the REST architecture, clients will send requests to retrieve or modify data, and servers send responses to these requests.

Communication from client to server will invoke the GET and POST requests.

4. System Features

<This template illustrates organising the functional requirements for the product by system features, the major services provided by the product. You may prefer to organise this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Register

1.1.1.4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Description	First time users can register for an account to create their account by simply clicking on the 'Register' button.
Priority	High
Frequency of Use	20 times per day

*1.1.2.*4.1.2 Stimulus/Response Sequences

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	29 January 2023	Date Last Updated:	21 March 2023

Actor:	User (Initiating Actor), Database
Description:	First time users can register for an account to create their account by simply clicking on the 'Register' button.
Preconditions:	1. The user must be connected to the internet 2. The user does not have an account prior to registration - Chosen email address does not exist in the database. 3. The user has navigated to the Login Interface

Postconditions:	1. The user has successfully registered an account for the application with a unique username and password and their account is added into the system database. Or 2. The user is notified of the reason(s) why the registration of the account is unsuccessful.
Flow of Events:	1. At the homepage of the website, the user can click onto the “Login” button and the system will redirect the user to the login page. 2. Since the user is a new user and would like to create an account to access the functions, he would need to register for a new account. 3. The user can click on “Register as a new user” which is situated at the bottom of the login page and he will be redirected to the registration page. 4. The user would then need to input a valid email address, username. 5. The user will need to input a valid password that contains at least 8 characters which includes an upper-case, lower-case letter, numerical digits and a special character. 6. The user will need to input the password once more to confirm his password. 7. At the end of the registration form, the user would need to check the checkbox of “I agree to the Terms of Use and Privacy Policy”. 8. The user will click on the “Sign up” button to register their account. 9. The system validates if there is an identical email existing in the system 10. The system will verify if the password satisfies all requirements 11. Upon verification, the system will store all information in the database securely. 12. Once registration is successful, the system will automatically help the user to log into his account and go into the home page after logging in.
Alternative Flows:	AF-S5: The user inputs a password that does not satisfy all the requirements set. 1. The system displays the message “Password does not meet all requirements, please try again!” under the password field. 2. The system returns to Step 5 and waits for inputs from the user. AF-S5: The user inputs a mismatched email address. 1. The system displays the message “Email does not match, please try again!” 2. The system returns to Step 5 and waits for inputs from the user.

	AF-S6: The user inputs a mismatched password. 3. The system displays the message “Passwords do not match, please try again!” below the password field. 4. The system returns to Step 5 and waits for inputs from the user. AF-S7: The user did not check the checkbox of “I agree to the Terms of Use and Privacy Policy.” 1. Upon clicking on the “Sign up” button, the system will display the message “Please check the checkbox for acknowledging the Terms of Use and Privacy Policy!” at the top of the registration form. 2. The system will return to Step 4 and wait for input from the user. AF-S8: The user did not complete all of the input fields. 1. Upon clicking on the “Sign up” button, the system will display the message “Please check that all fields are filled up!” at the top of the registration form 2. The system will return to Step 4 and wait for user to complete all inputs. AF-S9: The user inputs an email address that had already been registered. 1. The system displays the message “Email address has already been registered, please input another email address!” under the email address field. 2. The system returns to Step 4 and waits for input from the user.
Exceptions:	EX-1: The user did not receive the OTP in his email. 1. Users can click on the “Resend OTP” button that is made available after 60 seconds. 2. Once the user clicks on the “Resend OTP” button, a new OTP will be generated by the system and sent to the email address. 3. The system will return to Step 4 and wait for the user to input again. EX-2: The user request for more than three OTP request. 1. On the fourth time the user requests to generate a new OTP, the system will generate the message “Please try again with a different email” at the top of the registration form.

	2. The system returns to Step 4 and waits for input from the user.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

*1.1.3.*4.1.3 Functional Requirements

1.Users must be able to create an account on the system.
1.1.The system must display text fields for the user to enter his credentials
1.1.1.Text field must consists of username
1.1.2.Text field must consist of an email address and an confirm email address field
1.1.3.Text field must consists of a password and a confirm password field
1.2.The users must fill in all of the fields before clicking on the 'Sign up' button.
1.3.The system must verify all the fields filled in by the user before success creation of the account.
1.3.1.The system must make sure that the email address and confirm email address fields which the users inputted are the same.
1.3.2.The system must make sure that the password and confirm password fields which the users inputted are the same.
1.3.3.The email address that the users entered must not exist before.
1.3.4.The system needs to provide an error message to describe the reasons that the user's information is rejected.
1.4.The system must create an account for the user upon verification and store it in the database
1.5.The system must log the user to the main page of the system after registration.

4.2 Login

4.2.1 Description and priority

Description	Users can login to his/her account with the correct credentials that are inputted when users register.
Priority	High
Frequency of Use	20 times per day

4.2.2 Stimulus/ response sequence

Use Case ID:	002		
Use Case Name:	Login		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	29 January 2023	Date Last Updated:	7 February 2023

Actor:	User (Initiating Actors), Database
Description:	The user can login to his/her account with the correct credentials that are inputted when users register.
Preconditions:	1. The user must be connected to the Internet. 2. The user has a registered account.
Postconditions:	1. The User has successfully logged into his/her own account. OR 2. The user is notified of the reason(s) why he is unable to login into his account. Eg. You have input the wrong email address/password
Flow of Events:	1. At the homepage of the website, the user can click onto the “Login” button and the system will redirect the user to the login page. 2. At the login page, the system requests the input of both the email address and password. 3. The user inputs his registered email address and his password. 4. The user clicks on the “Login” button 5. The system verifies the credentials (Email and Password) provided with the database *VerifyCredentials* function 6. If the Email and Password are correct and verified, the user will be directed into his account.
Alternative Flows:	AF-S4: The user left the input field blank. 1. Upon clicking onto the “Login” button, the system displays the following message “Please ensure that all fields are filled up!”. 2. The system will prompt the user to fill up all the fields. 3. The system returns to Step 3 and waits for the user to fill in all the fields. AF-S5: The user inputs an incorrect email address or password.

	1. Upon clicking onto the “Login” button, the system displays the following message: “Invalid email address and/or password!” at the top of the login page using the *LoginError* function. 2. The system returns to Step 3 and waits for the user to fill in the fields again.
Exceptions:	EX-1: The user inputs incorrect email address and password for more than five times 1. After five attempts, the system will display the message “More than five tries, please try again after 10 minutes or click on forget your password” 2. The system will only accepts registration hence returning to Step 3 after 10 minutes EX-2: The user forgets his username and/or password 1. The user clicks on “Forget Password” that is situated below the login button. 2. The user can recover his/her account using the extended use case *RecoverAccount*
Includes:	
Extends	*RecoverAccount*
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.2.3 Functional requirements

1.User must be able to register for an account and login using the correct credentials

1.1.The system must display text fields for the user to enter his/her information

1.1.1.Text fields must consists of a email address

1.1.2.Text fields must consists of a password

1.2.Users must fill in all required text fields before pressing on the ‘login’ button

1.2.1.System must verify that all fields are being filled up for the users to be able to log in to the system

1.3.System must be able to log the user in if the information from the text fields are verified and correct

1.3.1.Email address must be found in the database

1.3.2.Password entered must match the password that is set with the email address in the database

1.4.System must be able to log the user into the home page of the website.

4.3 RecoverAccount

4.3.1 Description and Priority

Description	The user can request for help if he forgets his password and this use case can help to recover the account.
Priority	Medium
Frequency of Use	Approximately 5 times a day

4.3.2 Stimulus/Request Response

Use Case ID:	003		
Use Case Name:	RecoverAccount		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	29 January 2023	Date Last Updated:	15 February 2023

Actor:	User (Initiating Actors), Database
Description:	The user can request for help if he forgets his password and this use case can help to recover the account.
Preconditions:	1. The user must be connected to the internet 2. The user has forgotten his/her login credentials (Password of his account)
Postconditions:	1. The user has successfully recovered his/her account by changing his password. OR 2. The user has sought further assistance and support from the customer service.
Flow of Events:	1. The user clicks on “Forget your password” on the login page 2. The system displays the recover account page 3. User will input the registered email address and clicks on “Recover Account” 4. The system will verify the email address to ensure that the email address exist and sends an email to the email address that is given 5. The system then automatically generates a One-time Password (OTP) and emails it into the user's inbox. 6. User will input the OTP.

	7. Upon successful inputs of the OTP, the system will redirect users to a page where users can type in their new password with certain requirements. 8. The system then validates and verifies the new password and updates into the database. 9. Once the password is changed successfully in the database, users will be informed of the change in password and be redirected back to the login page.
Alternative Flows:	AF-S4: The user inputs the incorrect email address and email address is not registered 1. The system will display the following message: “Not a registered email, Please try again!” at the top of the page. 2. The system will returns to Step 3 and waits for user to input the registered email address AF-S7: The user inputs an incorrect OTP 1. The system displays the message “Incorrect OTP! Please try again!” at the top of the registration form 2. The system will re-generate a new OTP and send it to their email address. 3. The system returns to Step 4 and waits for the user to input again. AF-S8: The user inputs a password that does not satisfy all the requirements set. 1. The system displays the message “Password does not meet all requirements, please try again!” under the password field. 2. The system returns to Step 7 and waits for inputs from the user. AF-S8: The user inputs a mismatched password. 1. The system displays the message “Passwords do not match, please try again!” below the password field. 2. The system returns to Step 7 and waits for inputs from the user.
Exceptions:	EX-1: The user forgets his registered email 1. The user clicks on “Contact Support” button 2. The system will show the support email address 3. The user will then contact the support via email to recover his/her email address

	EX-2: The user did not receive the OTP in his email. 1. Users can click on the “Resend OTP” button that is made available after 60 seconds. 2. Once the user clicks on the “Resend OTP” button, a new OTP will be generated by the system and sent to the email address. 3. The system will return to Step 4 and wait for the user to input again. EX-3: The user request for more than three OTP request. 1. On the fourth time the user requests to generate a new OTP, the system will generate the message “Please try again with a different email” at the top of the registration form. 2. The system returns to Step 4 and waits for input from the user. EX-4: The user did not receive any email from the website despite multiple tries. 1. Users can click onto the “Help” button and seek help from the customer service. 2. User then waits for the customer service to attend to his problem. 3. The system returns to Step 4 and waits for further input from the user.
Includes:	
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.3.3 Functional Requirements

1.System must be able to ask user to input their registered email address

1.1.System needs to check if the email text field is being inputted

2.System must verify that the email address that the user registered exist in the database and verify that the email address exist.

3.System must be able to sent an OTP to the email address and verify it thereafter.

4.4 Search

4.4.1 Description and Priority

Description	Users will be able to search for the location they want to rent the car, pick up time, drop off time as well as filter the type of car that the users prefer
Priority	High
Frequency of Use	20 times a day

4.4.2 Stimulus/Request Response

Use Case ID:	004		
Use Case Name:	Search		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	30 January 2023	Date Last Updated:	21 March 2023

Actor:	User (Initiating Actor), CarRentingData (Database)
Description:	Users will be able to search for the location they want to rent the car, pick up time, drop off time as well as filter the type of car that the users prefer
Preconditions:	1. Users must have logged in 2. Users must be connected to an internet
Postconditions:	Users will obtain a list of car renting services that is sorted from the lowest price to the highest price.
Flow of Events:	1. Users can navigate through the navigation bar and click onto the “Start Searching” button that is situated on the navigation bar or the home page to start their searching process 2. Users can first type in their postal code of the current location 3. After that, users can enter the date of renting the car, start time of renting, drop off time, type of car (e.g. 5 seater car, electric car) depending on their preference 4. The system will retrieve the information of the preferred choice. 5. The system will display the results of the car renting services sorted from the lowest price to the highest price using the ComparePrice use case.
Alternative Flows:	AF-S2: Users key in invalid postal code of their location

	1. The system will generate the following message: "Please enter a valid postal code" at the bottom of the text box for the location 2. The system will return back to step 2 and wait for further input from the user. AF-S3: Users did not field in all the blanks of input 1. The system will generate the following message: "Please input all the fields" at the bottom of the text box 2. The system will return back to step 2 and waits for further input from the user AF-S3: Users do not know the amount of time they need in renting the car 1. Users can click on the "Need estimate time?" button below the drop off time section. 2. System will then redirect the users to the Maps page where users can input the starting location and their destination of travel using the extend *Maps* use case
Exceptions:	EX-1: The website loses internet connection while searching 1. A page which shows the following message "Internet connection is lost, please try again later." which informs the user of the problem. 2. The system will return to step 1 for users to key in the inputs.
Includes:	ComparePrice
Extends	Maps
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.4.3 Functional Requirements

2.User shall be able to search for their current location or area to rent the car in the search bar, and results of the nearest car rental location should be shown

2.1.System will be able to verify if users have inputted all text field.

2.2.System needs to verify if the current location or area of location is valid.

2.3.System will filter all the information and display all results.

3.User shall be able to get his current location using the GPS feature or manually input his current location using postal code

3.1.System will retrieve the current location using GPS feature of the Google Maps API

4.User shall be able to enter type of car from the given suggestions

5. User shall be able to enter the what time do they like to start renting the car and the time duration for the car rental

5.1. System will verify the inputs of the field, for instance the duration of car rental should be in numeric values.

4.5 Maps

4.5.1 Description and Priority

Description	Users will use the google maps API to input the locations they are travelling to and the system will calculate the total estimated travelling time users require.
Priority	Medium
Frequency of Use	10 times a day

4.5.2 Stimulus/Request Response

Use Case ID:	005		
Use Case Name:	Maps [Calculate distance]		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	30 January 2023	Date Last Updated:	23 March 2023

Actor:	User (Initiating Actor), Google Maps
Description:	Users will use the google maps API to input the list of locations they are travelling to and the system will calculate the total estimated travelling time the users require.
Preconditions:	1. Users are logged into the account 2. Users must be connected to an Internet 3. Users must have searched for the nearest renting location and set it as the starting point. 4. Google Maps API needs to be set up and needs to be able to be used
Postconditions:	Users will obtain the total estimated travel time and are able to estimate the total amount of time needed to rent the car.

Flow of Events:	1. After knowing the start location, users will input the number of locations they will be travelling to. 2. Users will then input all the addresses of the location by adding each of the locations one by one using the autocomplete function from Google Maps API. 3. The system will then compute the total estimated travelling time using the Google Maps Geocoding and Geolocation API, this will allow users to estimate the amount of time they require to rent the car. 4. Users can enter the total amount of time needed to stay at each location in total in one input. 5. The system will then display the total amount of estimated travelling time.
Alternative Flows:	AF-S2: User inputs an invalid address 1. The system will display the following message "Please enter a valid address" under the invalid address 2. The system will return to step 2 and wait for further inputs from the user. AF-S3 User did not add any location into the blanks or only include 1 location 1. The system will display the following message "Please enter more locations" as an alert to alert the users to include more locations. 2. The system will return to step 2 and wait for further inputs from the user.
Exceptions:	EX-2: Google Maps API is not being set up properly 1. A page which shows the following message "Error with Maps, Maps is not set up." which informs the user of the problem. 2. The system will return to step 1 for users to key in the inputs.
Includes:	
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

5.1.1.

4.5.3 Functional Requirement

1. User shall be able to click on various car park locations and check the distance and time from their current location to the carpark clicked

1.1. System must be able to verify that the car park location exists.

2. User shall be able to see the directions from their current location to the nearest car park they want to travel from
3. Users shall be able to select the location in which they are travelling to.

3.1. System must verify that the location that users are going is valid.

3.2. System must be able to calculate the estimated travelling time

4. Users shall be able to check the estimated time required for them to rent a car.

4.1. System must be able to calculate all the estimated travelling time and suggest the time in which users require to rent their car.

4.6 ComparePrice

4.6.1 Description and Priority

Description	Users will be provided with the results that is sorted from the lowest to the highest price
Priority	High
Frequency of Use	20 times a day

4.6.2 Stimulus/Request Response

Use Case ID:	006		
Use Case Name:	ComparePrice (Results Page)		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	30 January 2023	Date Last Updated:	7 February 2023

Actor:	User (Initiating Actor), CarRentingData
Description:	Users will be provided with the results that is sorted from the lowest to the highest price
Preconditions:	1. Users is connected to the internet 2. Users have login to their registered account 3. Users have used the search use case to search and filter about their preference 4. System shall be able to calculate the renting prices from all three companies
Postconditions:	Users will be provided with the list of cheapest rental service and users can make informed decision

Flow of Events:	1. Users will be directed to a page where the system will display the information according to user's preference that is sorted from lowest to highest price 2. The system will display information such as location of the car renting place, total rental amount, colour of car and type of car 3. The system will also show the rating of the car where users can refer to those ratings that is shown in the CarRating use case 4. At the side panel, users can filter more information such as the inclusivity of fuel using the filter use case
Alternative Flows:	AF-S4: User can choose not to select any filter. 1. The system will show all results sorted from the lowest to the highest price without applying any filter.
Exceptions:	
Includes:	*Filters*, *CarRatings*
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.6.3 Functional Requirement

1. User must be able to check the various prices of different car rentals for their travel

1.1. System must display more information about the various car rentals

2. System shall be able to be provided with the best deal according to the parameters set by the user

4.7 CarRatings

4.7.1 Description and Priority

Description	Users will be able to read reviews from other users about the different car renting companies and users can add reviews if they want
Priority	Medium
Frequency of Use	10 times a day

4.7.2 Stimulus/Request Response

Use Case ID:	007		
Use Case Name:	CarRatings		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	30 January 2023	Date Last Updated:	7 February 2023

Actor:	Users (Initiating Actor), Database
Description:	Users will be able to read reviews from other users about the different car renting companies and users can add reviews if they want
Preconditions:	1. Users is connected to the internet 2. Users have login to the registered account
Postconditions:	1. Users have read about the reviews from other users which can help them in making more informed choices OR 2. Users have added their reviews and ratings for the different car renting companies.
Priority:	Medium
Frequency of Use:	10 times a day
Flow of Events:	Reading Reviews 1. System will display the top 5 reviews 2. Users can read about the reviews and see the rating of the car renting company 3. If users wants to read more about the company, the link to the car renting company will be included in the reviews page Adding Reviews 1. Below the reviews section, the system will display a comment bar where users can add in their review of the car renting company 2. Users can add in their reviews and add their rating towards the company 3. After users are ready to add their review and rating, users can press the “Submit” button 4. The system will store the reviews and ratings in the database
Alternative Flows:	AF-S2 (Adding Reviews): Users left the fields blank and press the “Submit” button

	1. The system will generate the following message: “Please enter text into the fields” at the bottom of the page 2. The system will return to step 2 and wait for further inputs from the users.
Exceptions:	EX-1: No reviews about the car renting company 1. System will show the following message: “No reviews added” at the reviews section
Includes:	
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.7.3 Functional Requirement

1. Users are able to review and see rating of the cars
2. Users must be provided with the correct ratings and review of the car.

4.8 Filters

4.8.1 Description and Priority

Description	Include filters that filter choices such as inclusivity of fuel as well as if you can travel to malaysia.
Priority	Medium
Frequency of Use	10 times a day

4.8.2 Stimulus/Request Response

Use Case ID:	008

Use Case Name:	Filter		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	31 January 2023	Date Last Updated:	7 February 2023

Actor:	Users (Initiating Actor), Database
Description:	Include filters that filter choices such as inclusivity of fuel as well as if you can travel to malaysia.
Preconditions:	1. Users is connected to the internet 2. Users have login to the registered account 3. Users have used the search use case to search 4. Users have view the list of prices in the comparePrices use case
Postconditions:	Users will be able to filter their choices according to their preference for a more filtered list of suggestion
Flow of Events:	1. At the side panel, users can tick onto the check box that are applicable to their choices 2. Users are able to filter by the following selection: Fuel type and car condition 3. The system will regenerate the choices as follows 4. The system will generate a new list of choices according to the filtered choice 5. Users can now view the more detailed list according to their own preference
Alternative Flows:	
Exceptions:	
Includes:	
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

4.8.3 Functional Requirement

1. User should be able to filter their choices according to their own preference.

4.9 ViewInformation

4.9.1 Description and Priority

Description	Users can view more information about the specific car rental.
Priority	Medium
Frequency of Use	10 to 15 times a day

4.9.2 Stimulus/Request Response

Use Case ID:	009		
Use Case Name:	ViewInformation		
Created By:	Lim Ke En	Last Updated By:	Lim Ke En
Date Created:	12 February 2023	Date Last Updated:	16 February 2023

Actor:	Users (Initiating Actor), Database
Description:	Users can view more information about the specific car rental upon clicking on the view button
Preconditions:	1. A detailed list of car rental have been generated for comparison that is arranged from the lowest price to the highest price
Postconditions:	1. Users will be able to view more information about the car such as the terms and conditions 2. Users can be redirected to the car renting page if they choose to book the specific car rental service
Priority:	Medium
Frequency of Use:	10 to 15 times
Flow of Events:	1. Upon wanting to know more information, users can click on the view button. 2. After clicking the view button, the detailed information about the car will be shown. 3. Full detailed information about the car would be listed in that page. 4. Information such as the terms and condition of the car renting service and some (Frequently Ask Question) FAQ will be listed for the user

	5. If users wants to book the certain car renting service, they can be redirected to the specific website or application
Alternative Flows:	
Exceptions:	
Includes:	
Extends	
Special Requirements:	
Assumptions:	
Notes and Issues:	

5. Other Nonfunctional Requirements

5.1 Usability Requirements

5.1.1 System should display prices up to 2 decimal points and in SGD currency.
5.1.2 System should use 12 hour clock to describe current time
5.1.3 System should be able to show time in hour and minutes for time required in travel or time of duration of car rental
5.1.4 System should show FAQ information in the local language according to the user's locale
5.1.5 System must be able to display all the information in the language preferred by the user
5.1.6 Users must not spend more than 5 minutes to create an account
5.1.7 If network connection is not available, the website must display and informative pop-up box telling the users to try again later.

5.2 Performance Requirements

5.2.1 After searching, users should get up to 20 car rental recommendations that match their preference within 30 seconds.
5.2.2 System must be able to locate the user's current location within 5 seconds if the user choose to use the current location finder by Google API.
5.2.3 System must be able to calculate the total estimated time needed to rent the car within 15 seconds.
5.2.4 System must not be shut down for more than 10 hours in a year
5.2.5 Website should be fully functional within 15-30 seconds.

5.3 Supportability Requirements

5.3.1 Website must be able to run and are compatible with most web browsers
5.3.2 Website should be able to be tested, adapt, maintain, configure, scale and localise

5.4 Security Requirements

5.4.1 The system must not disclose the user’s country and email address to other users
5.4.2 Other users must not be able to view other user’s password, email addresses.
5.4.3 Other users must not be able to change other user’s passwords and email addresses.
5.4.4 The system should not be disclosing the user's email address and password for any personal or commercial use.

5.5 Interface Attributes

5.5.1 System should interact with the API of google maps and other web-scraping websites

5.6 Overall Maintenance

5.6.1 Updating the database of the website regularly
5.6.2 Constant updates for website design or changes
5.6.3 Perform backups regularly and stored off site
5.6.4 Update website’s software and plugins
5.6.5 Checking and removing spam reviews, comments and user accounts.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Created by	Li jiaqian
Date created	29 January 2023
API	API stands for Application Programming Interface, which is a set of protocols for two or more computer programs to communicate with each other
User account	A means by which app users can get access to

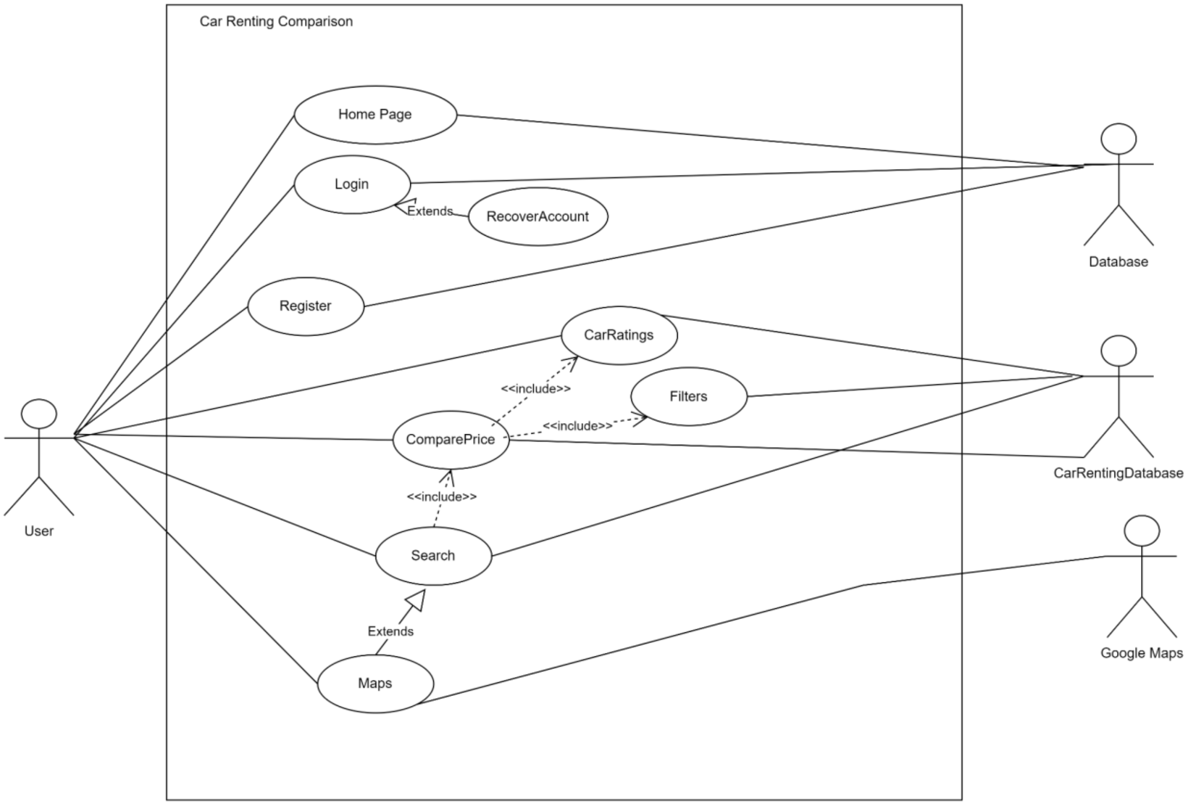
	the platform. It includes a username as an identification set by user and an registered email address or contact number where the platform will send OTP for users to log in. The users can also log in with the password they set when registering.
Database	A spreadsheet that contains all the users' accounts including username, registered emails or phone numbers, passwords and their Favourite Starting Point etc.
Starting point	A box where users can enter their starting location. They can directly select their current position or their favourite starting point.
Favourite starting point	A location that is most often selected by a user. Platform will record the starting points a user has selected and sort them by number of times. The most often selected one will be marked as Favourite starting point and provided to users to select without entering location in detail
Nearest rental points	A set of rental points that belong to different renting companies and are closest to the starting point entered by users.
Rating	A scaling system implemented by our platform for users to rate and review for their car-renting experience. The scale is from one star to five stars. More stars stand for better user experience. The rating of a renting company is the average of the users' rating. The rating system will be offered to a user after a renting service.
One time password (OTP)	A six-digit number which will be sent to a user's email address or phone number via email or SMS for the user to log in the platform. An OTP will expire in 3 minutes and the user can request another OTP after 1 minute.
Destinations	A series of boxes where users can enter the locations where they want to go in order. If the number of boxes provided is not enough, users can also add more boxes for them to enter their destinations
Calculated time spent	The time that is calculated by using Google

	Maps by summing the time needed to reach destinations. Since the companies in our platform only provide “from A to A” service, the return time is also counted.
Estimated lease time	The time that is estimated by users themselves. They should refer to the calculated time and then include other time they will spend.
Estimated renting fee	The fee that is estimated through multiplying the estimated lease time offered and unit price of a company.
Filter	A selection bar where users can choose the car by filtering price, categories,types and features

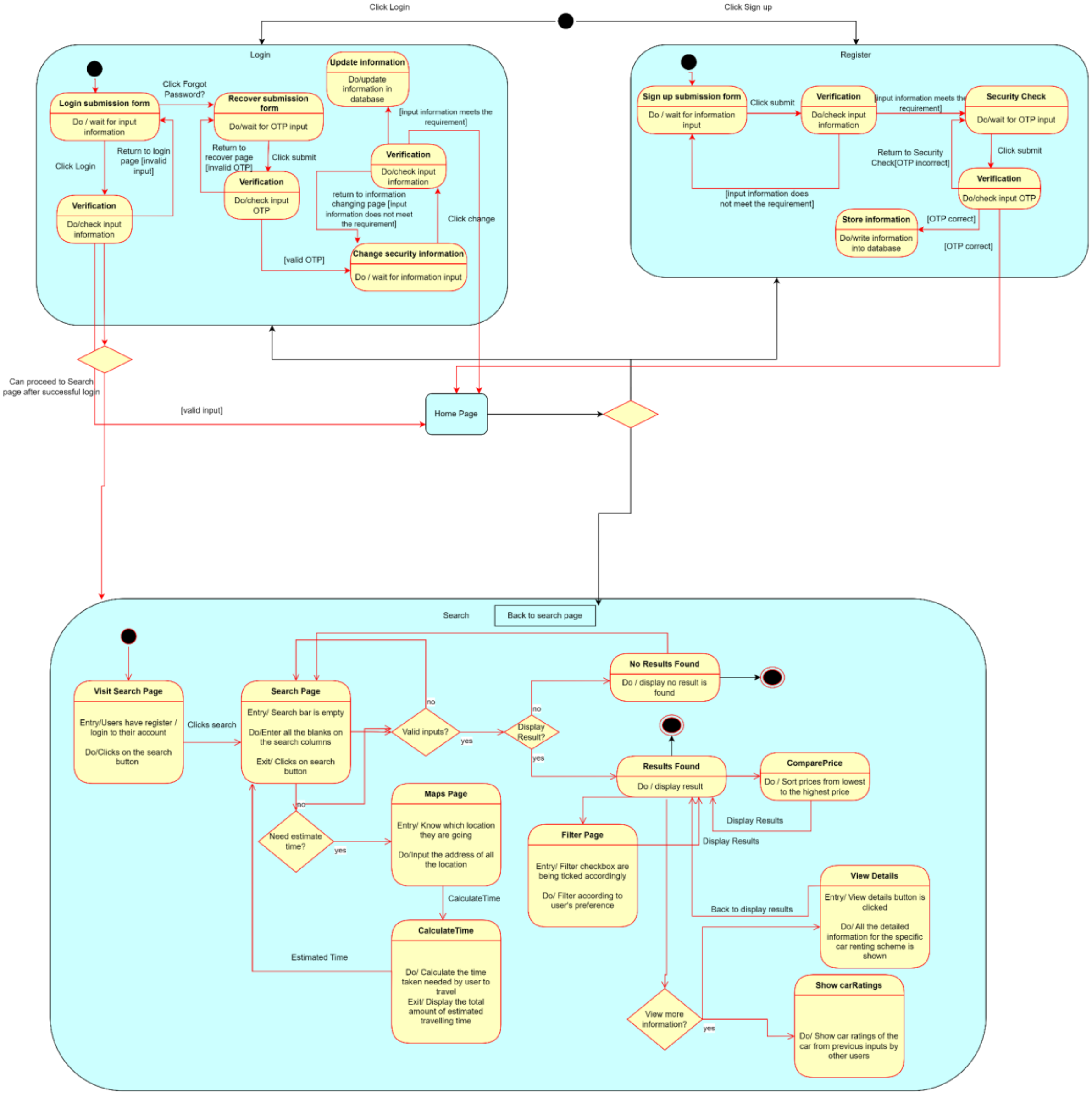
Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

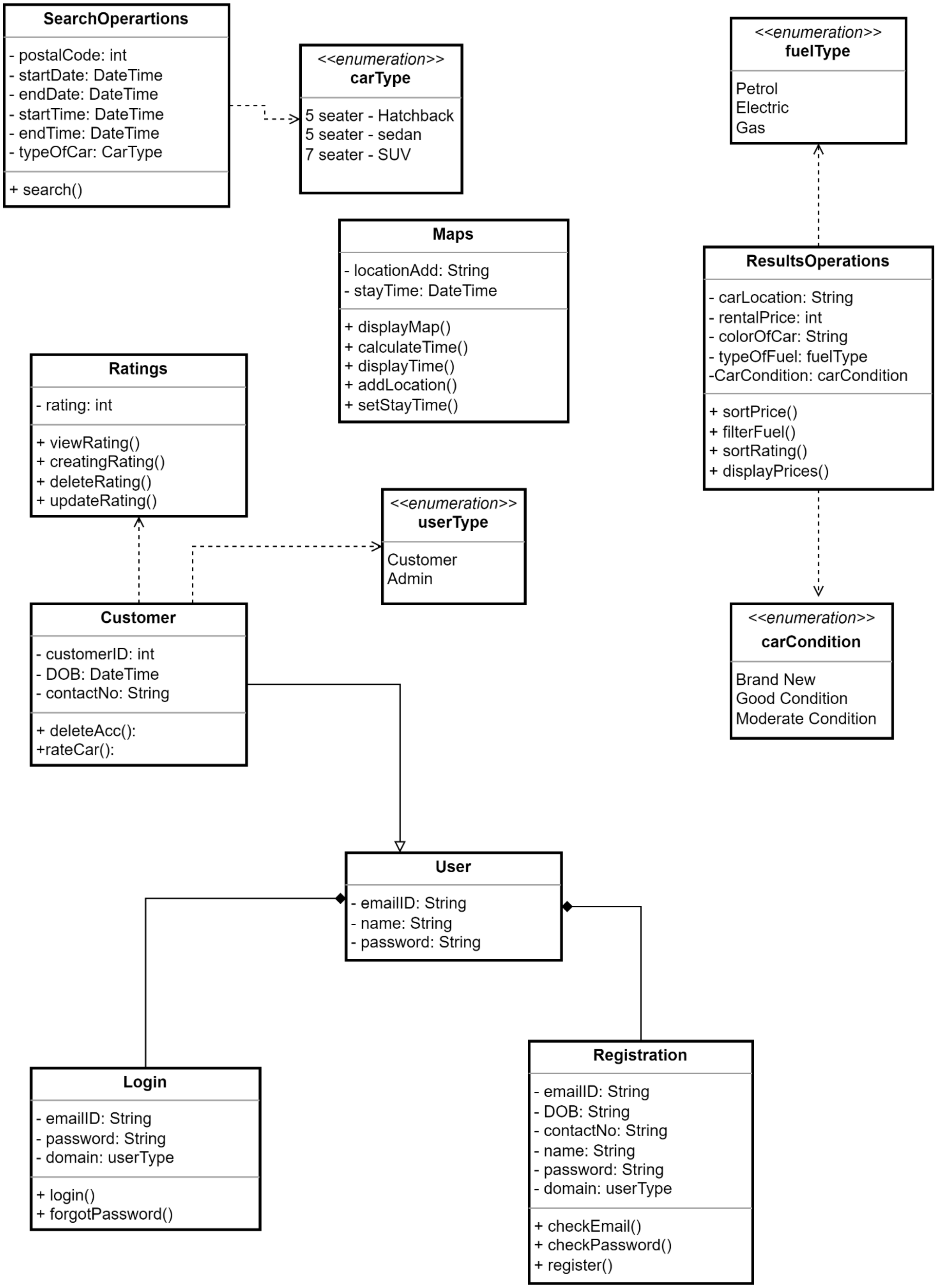
- Use Case Diagram



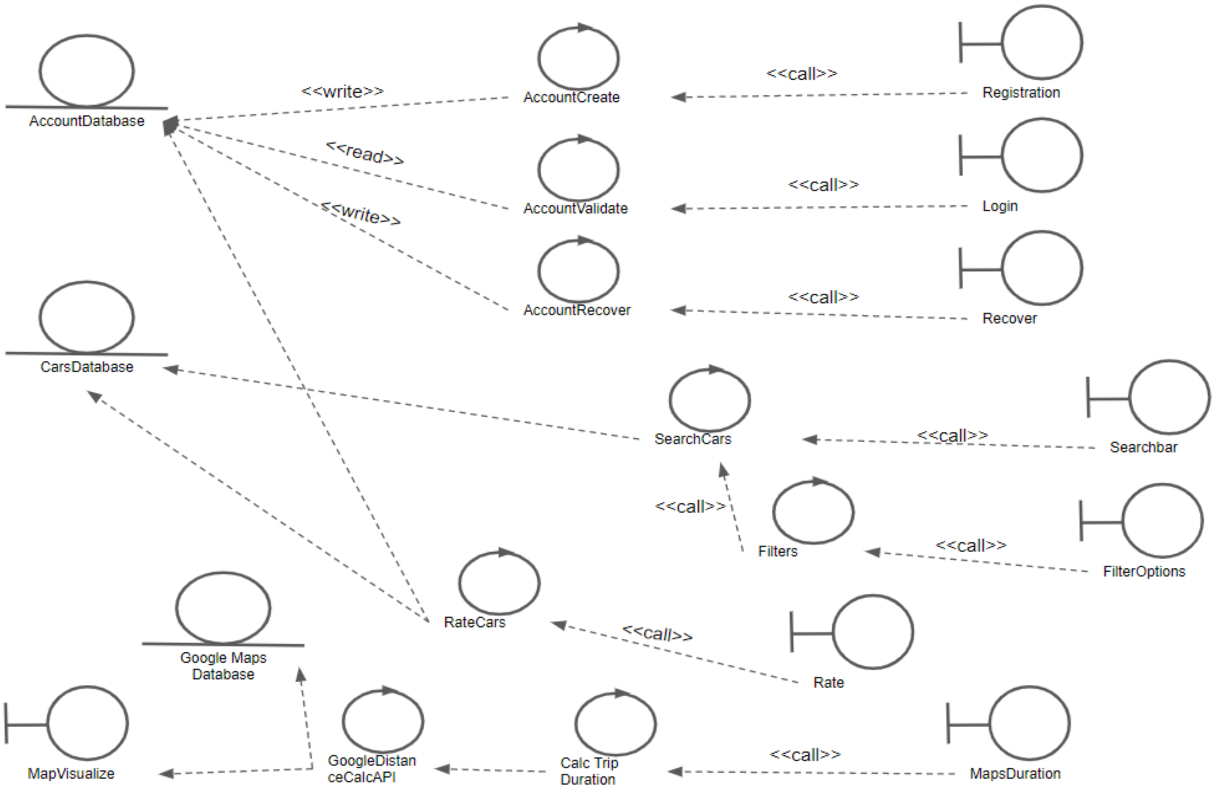
- Dialog Map (State Machine Diagram)


Click Login
Click Sign up
Login
Update information
Do/update information in database
Click Forgot Password?
Login submission form
Do / wait for input information
Recover submission form
Do/wait for OTP input
[input information meets the requirement]
Return to login page [invalid input]
Click Login
Return to recover page [invalid OTP]
Click submit
Verification
Do/check input information
Verification
Do/check input OTP
return to information changing page [input information does not meet the requirement]
Verification
Do/check input information
Click change
[valid OTP]
Change security information
Do / wait for information input
Register
Sign up submission form
Do / wait for information input
Click submit
Verification
Do/check input information
[input information meets the requirement]
Security Check
Do/wait for OTP input
Return to Security Check[OTP incorrect]
Click submit
[input information does not meet the requirement]
Verification
Do/check input OTP
Store information
Do/write information into database
[OTP correct]
[OTP correct]
Can proceed to Search page after successful login
[valid input]
Home Page
Search
Back to search page
Visit Search Page
Entry/Users have register / login to their account
Do/Clicks on the search button
Clicks search
Search Page
Entry/ Search bar is empty
Do/Enter all the blanks on the search columns
Exit/ Clicks on search button
no
Valid inputs?
yes
no
Display Result?
yes
No Results Found
Do / display no result is found
Results Found
Do / display result
ComparePrice
Do / Sort prices from lowest to the highest price
no
Need estimate time?
yes
Maps Page
Entry/ Know which location they are going
Do/Input the address of all the location
Filter Page
Entry/ Filter checkbox are being ticked accordingly
Do/ Filter according to user's preference
Display Results
Display Results
CalculateTime
CalculateTime
Do/ Calculate the time taken needed by user to travel
Exit/ Display the total amount of estimated travelling time
Estimated Time
View Details
Entry/ View details button is clicked
Do/ All the detailed information for the specific car renting scheme is shown
Back to display results
View more information?
yes
Show carRatings
Do/ Show car ratings of the car from previous inputs by other users

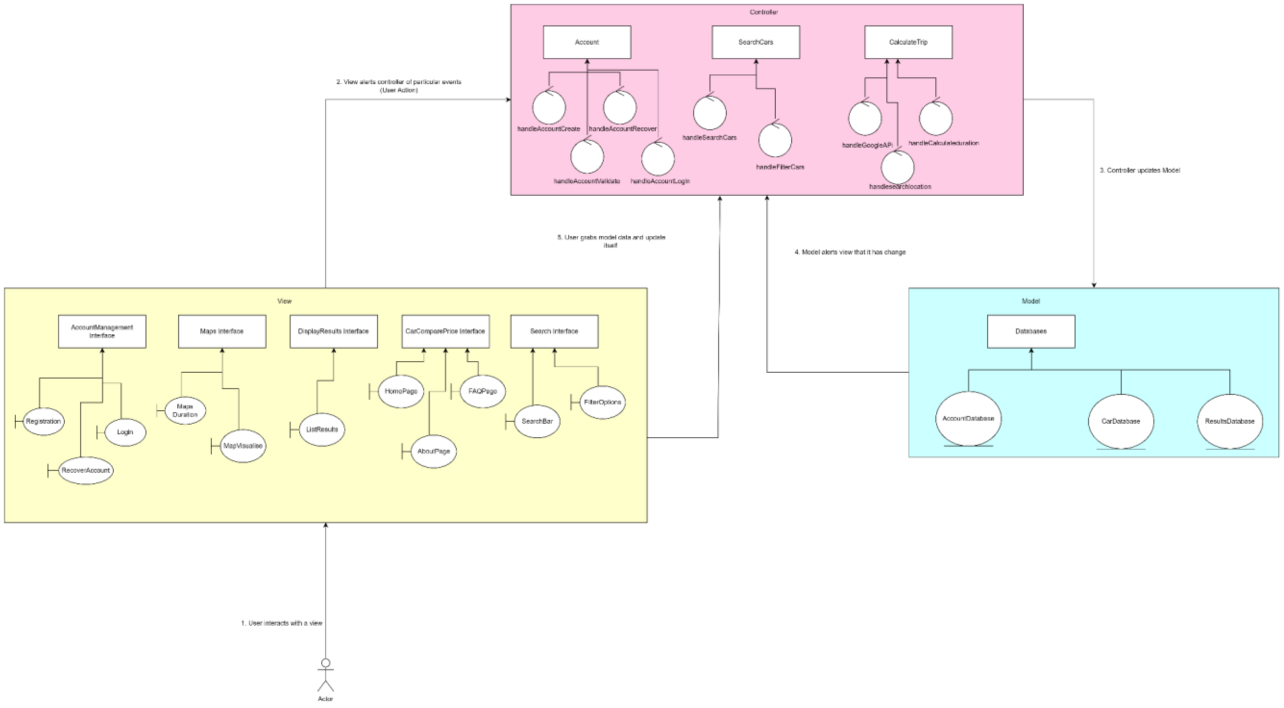
- UML Class Diagram



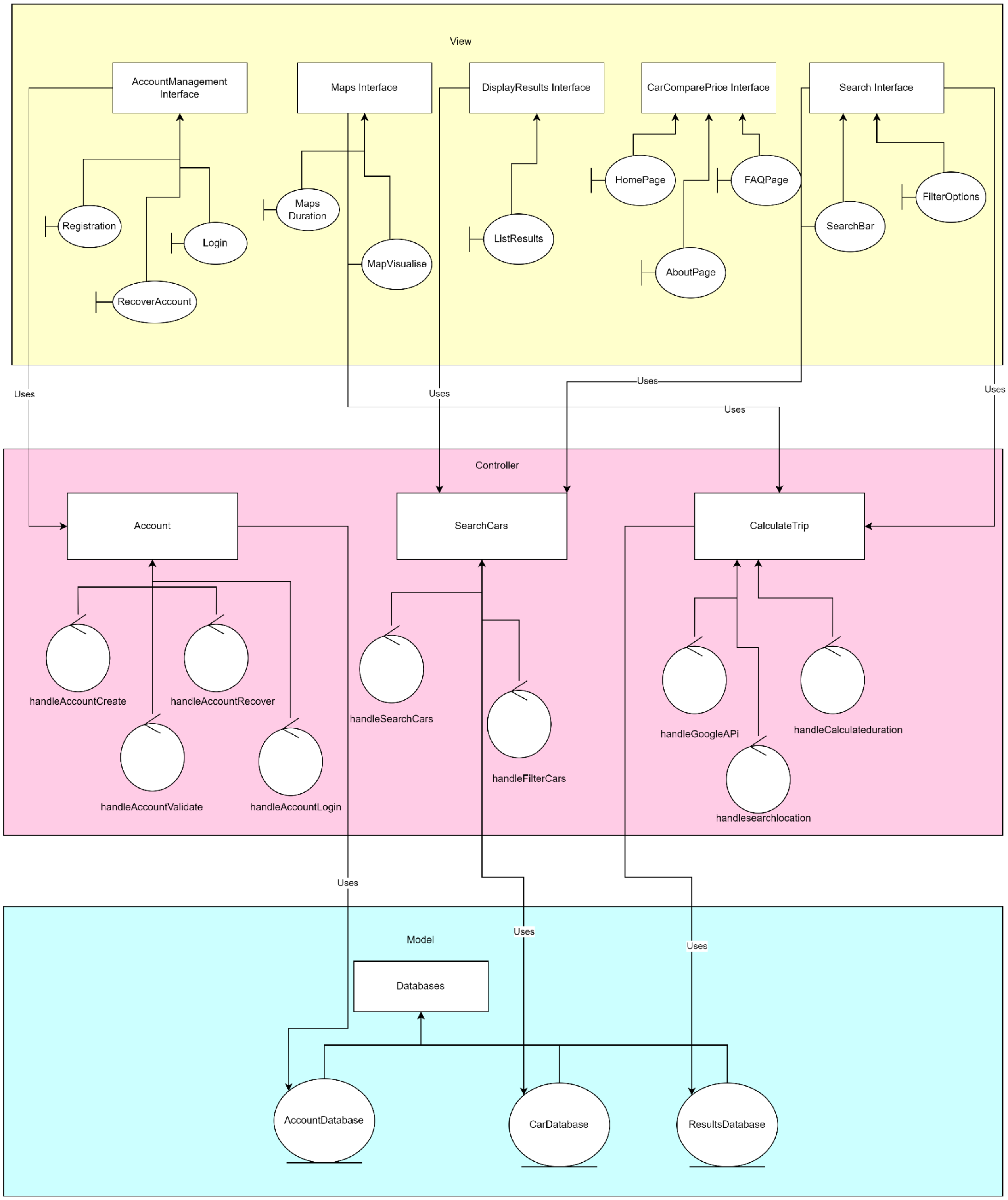
- Key boundary Classes and Control Classes



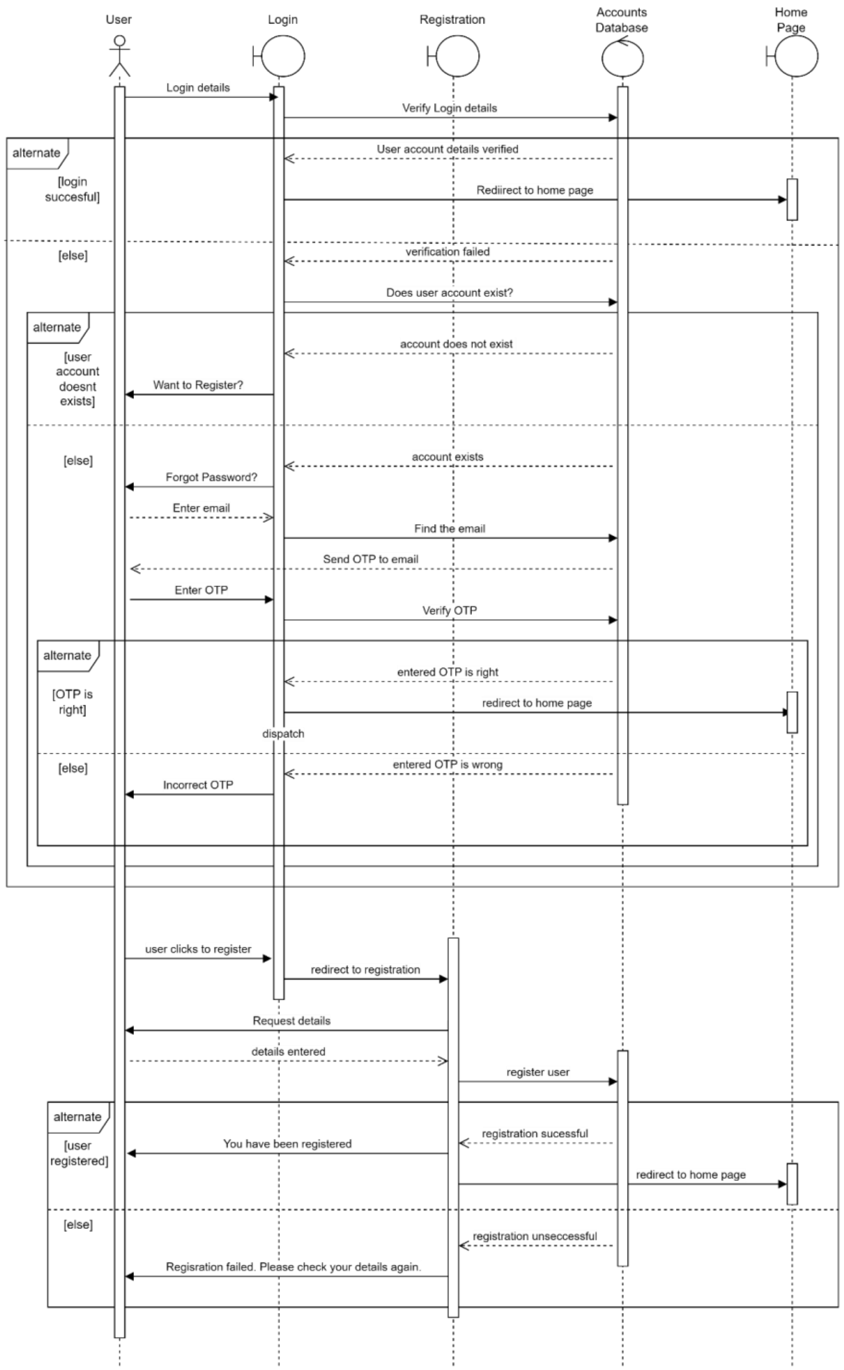
- Software Architecture (Model, View, Controller Architecture)



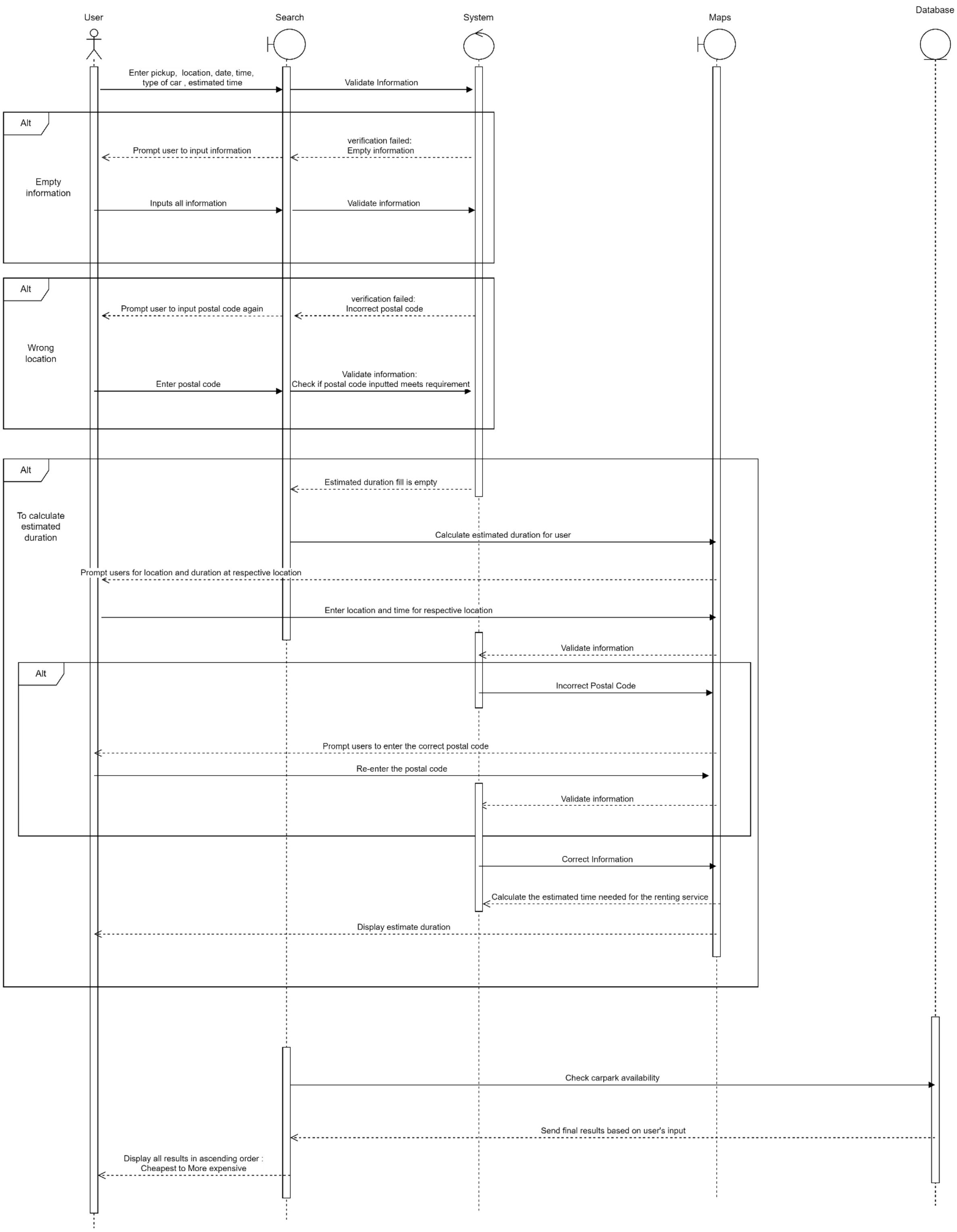
- Software Architecture (3-Layered Architecture)



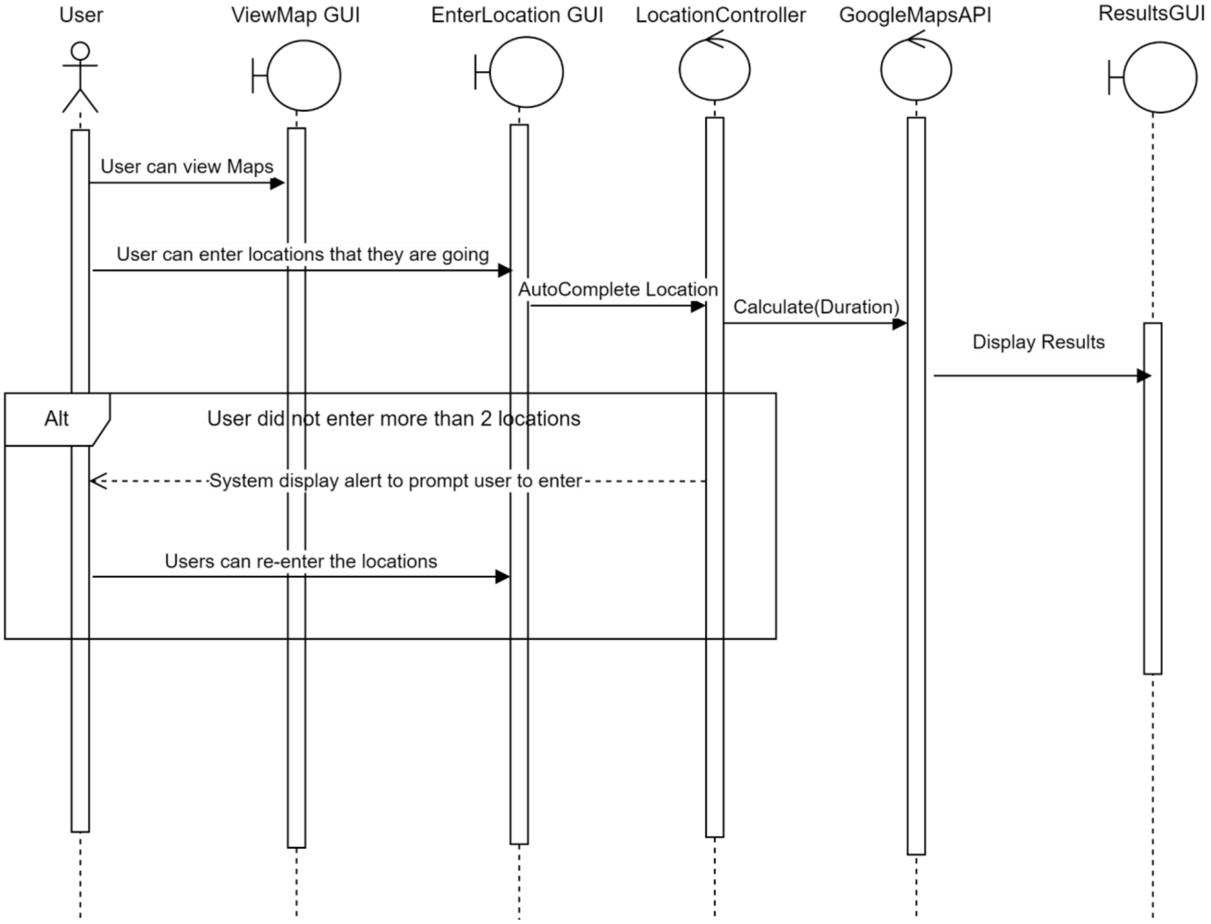
- Sequence Diagram
  - Login and Register Sequence Diagram



- ○ Search Sequence Diagram


User
Search
System
Maps
Database
Enter pickup, location, date, time, type of car , estimated time
Validate Information
Alt
verification failed: Empty information
Prompt user to input information
Empty information
Inputs all information
Validate information
Alt
verification failed: Incorrect postal code
Prompt user to input postal code again
Wrong location
Enter postal code
Validate information: Check if postal code inputted meets requirement
Alt
Estimated duration fill is empty
To calculate estimated duration
Calculate estimated duration for user
Prompt users for location and duration at respective location
Enter location and time for respective location
Validate information
Alt
Incorrect Postal Code
Prompt users to enter the correct postal code
Re-enter the postal code
Validate information
Correct Information
Calculate the estimated time needed for the renting service
Display estimate duration
Check carpark availability
Send final results based on user's input
Display all results in ascending order : Cheapest to More expensive

- Maps Sequence Diagram [Calculate Duration]


User
ViewMap GUI
EnterLocation GUI
LocationController
GoogleMapsAPI
ResultsGUI
User can view Maps
User can enter locations that they are going
AutoComplete Location
Calculate(Duration)
Display Results
Alt
User did not enter more than 2 locations
System display alert to prompt user to enter
Users can re-enter the locations

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>