Software Requirements Specification

for

Parking4TheWin

Version 1.0 approved

Prepared by:
Chen Ruxing, Ivan Yuen,
Joanna Lim, Min Khant,
Tan Jun Xiong, Qiu Zhen

4TheWin

04 April 2023

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

1. Introduction

1.1 Purpose

This document provides a detailed description of a Carpark Availability Checking System. This system is a website application of the real-time number of available parking lots in HDB carparks in Singapore, providing convenience for drivers to find available places to park their vehicles. The system also encourages drivers to indicate interest through rewards, thus providing a more accurate indication of the vacancy of carparks.

1.2 Document Conventions

For the readability of this article, various conventions and abbreviations have been used in streamlining the document. Some conventions used throughout this document are as follows:

- Website refers to the application/software developed for this project.
- All features were developed based on the context of Singapore. This assumption persists throughout the entire development process.
- All users are referenced with he/his/him, with no intended bias or reference to the gender identity of the user. This was used for a simple matter of consistency.
- Drivers refer to any user who wishes to use the website to find parking vacancies and/or claim rewards.
- Employees refer to users representing corporate entities who wish to use the website to create rewards and attract customers.
- Once a driver has registered his COE successfully, his account is said to be authenticated.
- The priorities of the System Features are categorized as follows:
  Necessary | This feature must be completed. All other features depend on the completion of this feature.
  High | This feature is highly recommended to be completed.
  Medium | This feature greatly benefits the driver when completed.
  Low | This feature provides some benefits to the driver.
- For the functional requirements of System Features, the following abbreviation tags are used, in the exact order. Numbers following the tag simply help denote the different conditions.
  PRE - Preconditions are necessary before executing the feature.
  ALT - Alternate flows or paths taken by the user in executing the feature.
  EXC - Exceptions or errors by the user that deviate from the manner of the intended usage.
  POS - Post-conditions necessary upon the completion of the feature.

REQ - Other software requirements or assumptions necessary for successful execution.

- Any use of the term “bubble” refers to the visual tags used on our map canvas for identifying unique carpark locations.

For a full list of keyword definitions, refer to “Appendix A: Glossary”.

1.3 Intended Audience and Reading Suggestions

This document contains a detailed explanation of the Carpark Availability Checking System. It will explain the purpose and features of the system, the interfaces of the system, the functional and non-functional requirements of the system, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

<Drivers>
The primary users of our app are drivers who would like to make an informed decision on the details and availability of carparks close to their intended destination before commencing their journey. Drivers should refer to Section 2.6 for a user manual on the application.

Drivers may view the specifics of a feature in Section 4 if they believe the website is not functioning as intended.

<Corporate>
The secondary users of our app are companies to give out vouchers for those who actively contribute to indicate their interest in going to specific carparks. This allows companies to advertise themselves and attract potential customers. Similarly, refer to Section 2.6 for a user manual on the application and Section 4 for feature specifics.

<Developers>
Developers who intend to continue developing the application should refer to this document. Read the document (and other references) in the following order:

- Section 1: for a quick understanding of the document outline.
- Section 2: for a top-level understanding of the development process.
- User Manuals (both): refer to the manuals and run the program to grasp the functionalities first-hand.
- Section 3: to internalise the interface requirements and the system architecture supporting all features.
- Section 4: for detailed response sequences between users and the website.
- Section 5: any other non-functional requirements the developer may want to consider.

1.4 Product Scope

This application is based on the real-time available parking lots in HDB carparks data from Data.gov.sg.

The drivers can get information about available HDB parking lots near their destination. This will help them by making it easier to find places to park before they depart. The drivers can also

indicate their interest in going to a specific carpark. This statistic will assist other drivers in having a clearer judgement of the carpark's vacancy upon their arrival.

The drivers can get points when they arrive at the interested carpark and redeem rewards with the points they get. Corporations can use this system to give out vouchers for the drivers, promoting themselves and attracting potential customers through our website.

1.5 References

The README file provided along with this document specifies the location of all documents. Other documents that may be referred to alongside the Software Requirements Specification (SRS) are:

- User Manual (Drivers)
- User Manual (Corporate)
- Use Case Diagram
- Use Case Sequence Diagrams
- Class Diagrams
- System Architecture
- Test Cases

2. Overall Description

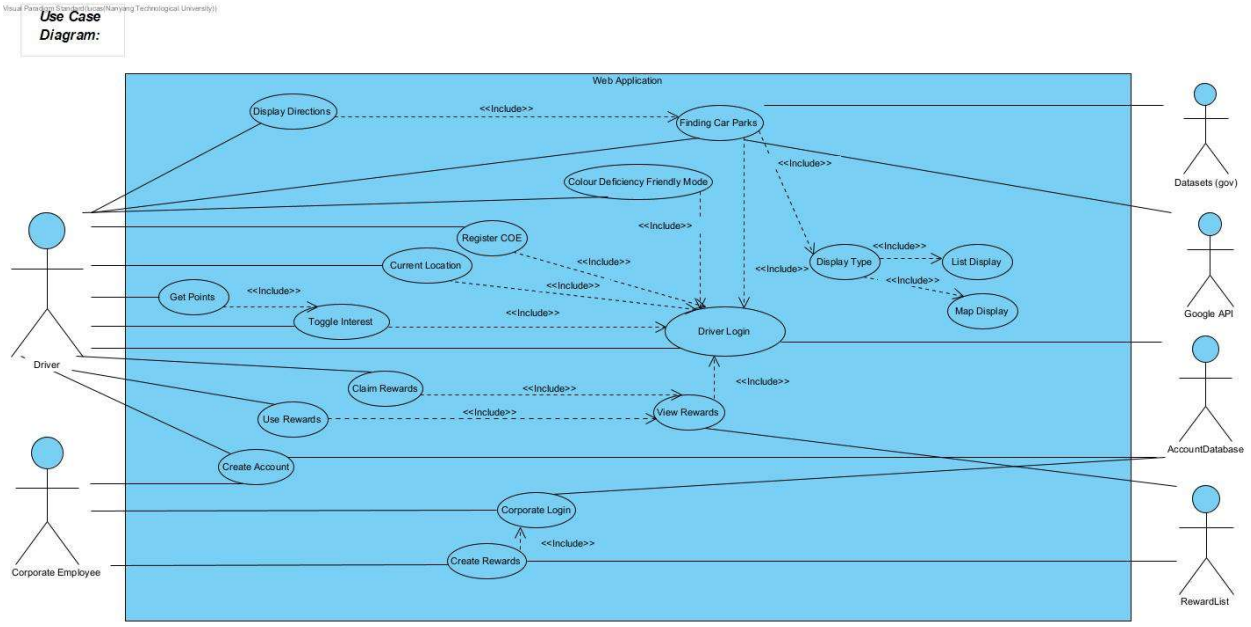
2.1 Product Perspective

The website is a self-contained product designed to enable drivers to locate available parking spots within a specified radius of their desired destination. The primary purpose is to help drivers find available parking spots in a specified radius of their desired destination. The system must allow drivers to query all carparks within a specified radius and display the results in the form of bubbles on a map or a list, which can be sorted by distance and vacancy.

The system must contain two different account types: Driver and Corporate. Drivers will register for an account using a unique email address, and password. Registration of their Certificate of Entitlement (COE) is optional.

The system must allow drivers who have registered their COE to indicate interest in a particular carpark. The system will display the expected demand for a carpark based on the Government database and indicated interests of drivers. The system must reward drivers who park at the lots they indicated an interest in and the rewards system will be point-based.

The system must allow Corporate accounts to create rewards, which can be claimed by drivers based on their points. The system will provide an interface for Corporate accounts to create and manage rewards.



Use Case Diagram

2.2 Product Functions

Listed below are all the functions accessible to a user of the website. Most features are intended for drivers, and some features are intended for corporate employees.

1. **Account Creation for Drivers**
   Drivers should create their own accounts to use the map. COE Registration is still needed to access Points, Rewards, and Interest Indication. Each account must be created with a unique email address.
2. **Driver Login**
   The driver will log in to his account, after which he can save destinations and indicate interest in going to carparks.
3. **COE Registration**
   Drivers with accounts are entitled to exclusive rewards and discounts. We want to ensure that drivers won't abuse the system. To do this, we must check that they own a vehicle. We can do this by verifying their COE.
4. **Colour Vision Deficiency Mode**
   The driver can change the colour scale used for bubbles on the map. A suitable colour scale is available for drivers with red-green colour vision deficiency.
5. **Querying carparks**
   The driver obtains information about carparks near his destination.
6. **Updating Current GPS Location**
   Drivers must update the current location as seen by the website to view accurate navigation directions from their actual current location to the selected carpark.
7. **Interest Indication & Navigation**
   Drivers must indicate interest and park at the indicated location to receive reward points. Indicating an interest in the carpark provides a more accurate judgement on the vacancy of the selected carpark. Navigation to the selected carpark will be displayed after interest has been indicated.
8. **Reward Points Collection**
   To get reward points after the driver has indicated his interest in parking at a parking location and successfully parked at that location.
9. **Account Creation for Corporate**
   Corporates must create their accounts to add Rewards for drivers to redeem. Each account must be created with a unique business UEN. The corporate employee will be the actor of this use case.
10. **Corporate Login**
    Corporate employees must log in to the Corporate Account to create rewards.
11. **Create Rewards for Corporate**
    Employees can create rewards to be issued by the corporate, for drivers to redeem with their points.
12. **View Rewards**
    The driver views available rewards that can be claimed with reward points, and also view the rewards he has claimed.
13. **Claim Reward**
    The driver claims rewards using his reward points.

14. **Use Reward**
    The driver uses a claimed reward.

2.3 User Classes and Characteristics

The most important user class for this product is the drivers as they are the primary users and the reason for the existence of the website. Corporate employees are less important but still crucial for the purposes of the website. The different user classes are as follows:

1. Drivers:
   The drivers are primary users of the website, and they will use it to search for available parking spots. They will require a basic level of technical expertise to be able to navigate the website and search for parking spots. The drivers will need to register for an account, provide personal information such as their email address and COE, and indicate their interest in a particular parking lot. They will also need to upload a picture of their vehicle for verification purposes and receive rewards for parking at the lots they indicated an interest in.

2. Corporate Employees:
   This user class will represent companies that will use the website for reward-related purposes. They will be creating and managing rewards for website users. The corporate employees will log into their accounts by providing their company's UEN and password.

2.4 Operating Environment

Our system's components are Python, the web server and the web server gateway interface. The Nginx web server and Python support Windows, Linux and macOS, but Gunicorn doesn't support Windows. To overcome this, we can use WSGI or Docker to run an isolated virtual environment for the server. An amd64 system is preferable to an Arm system as some docker images are known to support only the amd64 version.

2.5 Design and Implementation Constraints

Given that there are hardware and network constraints, we have designed our software to meet the following requirements:

1. **Web Browser**
   The website must be able to run on modern web browsers and be minimally backwards compatible with old web browsers.
2. **Database Management System**
   The website must be designed to work with SQLite and SQLAlchemy.
3. **Bandwidth and Server Capacity**

The website must be designed to handle a potentially large number of simultaneous user requests, 10,000 requests, and should be optimised for efficient use of bandwidth and server resources.

4. **Security and Privacy**
   Hashing must be used for password storage to protect user data and prevent unauthorised access to sensitive information.
5. **Third-Party APIs**
   The website needs to integrate with MapBox and Data.gov.sg provided by the Government Technology Agency, Singapore.
6. **Mobile Responsiveness**
   The website must be designed to be mobile responsive, providing an optimal user experience across various mobile devices and screen sizes, this includes sidebar optimisation.
7. **Accessibility**
   Font sizes must be appropriate, as well as sufficient contrast for readability.

2.6 User Documentation

The website will come with a Step-by-Step User Manual for Drivers and corporate employees separately, provided along with this document. The manuals include FAQs for the users. Refer to the documents “User Manual (Drivers).pdf” and “User Manual (Corporate).pdf” respectively. The location of all documentation is listed in the README file in the home directory.

2.7 Assumptions and Dependencies

Assumptions

1. **COE Registration**
   It is assumed that drivers registering with their COE are acting in good faith and will not provide false information.
2. **Receive Reward Points**
   It is assumed that drivers act in good faith and actually park at the carpark that he has indicated an interest in.

Dependencies

1. **Availability of rewards**
   The website is dependent on having sufficient rewards available to incentivise drivers to indicate their interest in parking at a carpark.
2. **Device compatibility**
   The website is dependent on the driver having a working camera for uploading photos.
3. **Server uptime**
   The website is dependent on the server being up and running to process user actions and store data.

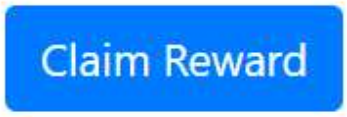
3. External Interface Requirements

3.1 User Interfaces

We have included a few features to improve the User Experience and User Interface (UX/UI). Users may interact with the website wherever necessary. Bootstrap v5.0 is used for the Frontend Styling Framework, and Mapbox API is used for the Map Canvas Rendering, Geocoding and Route calculation. UX/UI features are as follows:

- For all interactive buttons, there will be a shift in colour tone when the user hovers over the button. A border will be generated upon clicking the button. All buttons clicked will trigger a series of relevant and appropriate responses in the website.

  Without hovering:

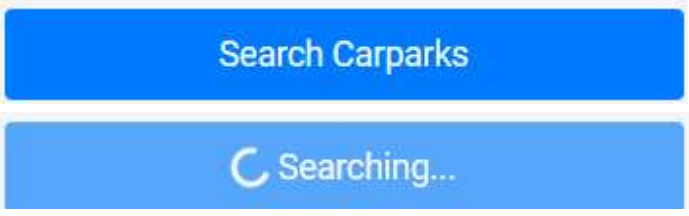
  With hovering:

  When clicking:

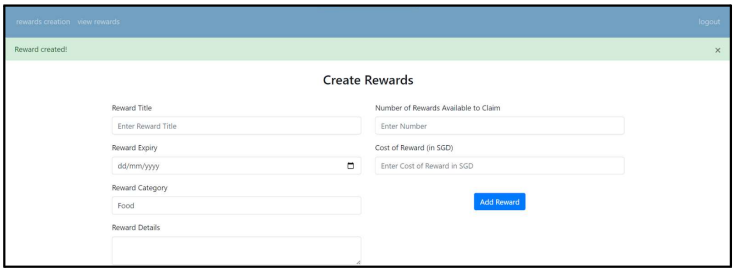
  

  When clicking on button with async or long-running task:

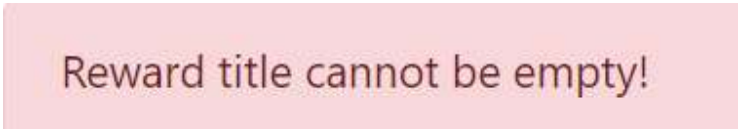
  

- All flash messages will be shown at the top of the page for clear visibility. The messages are colour coded to differentiate between error messages and successful completion messages.

Top of page, successful message:

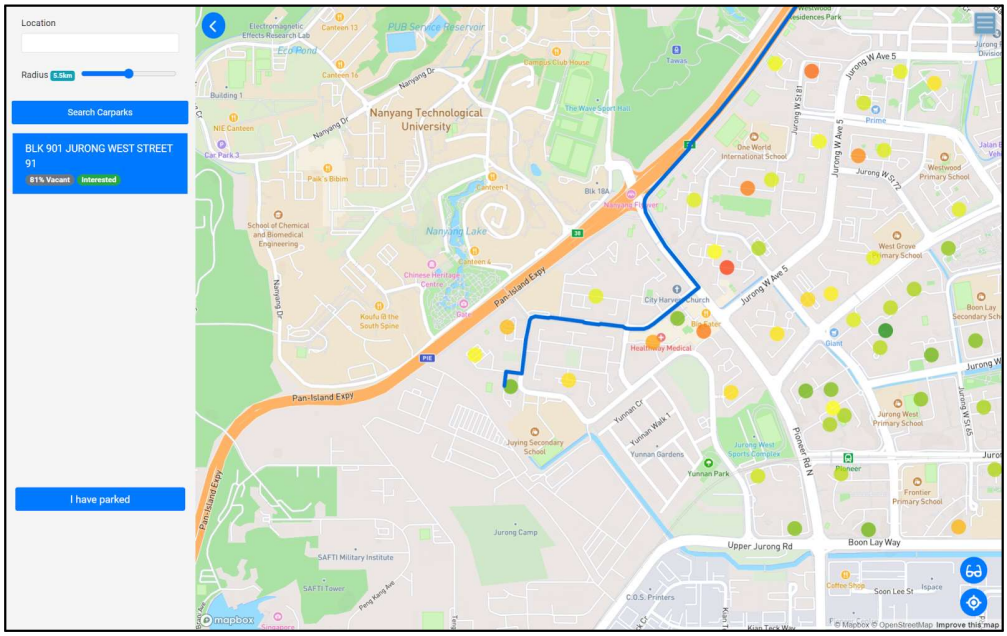


Error message:

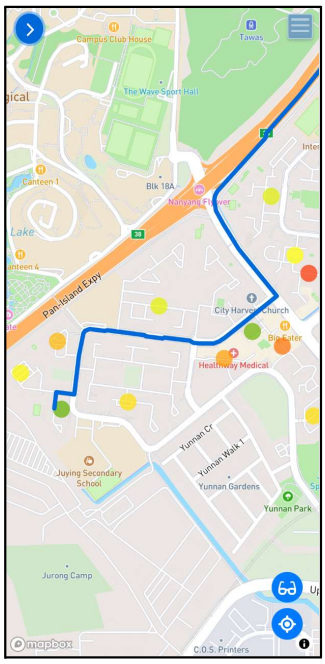


- The Search Query Sidebar can be minimized for clear view of map on all screen layouts.

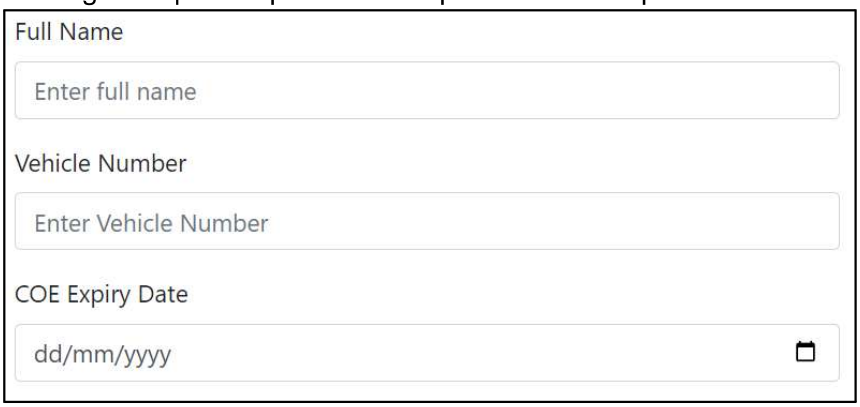
Sidebar not minimized:



Sidebar minimized, on a mobile phone:

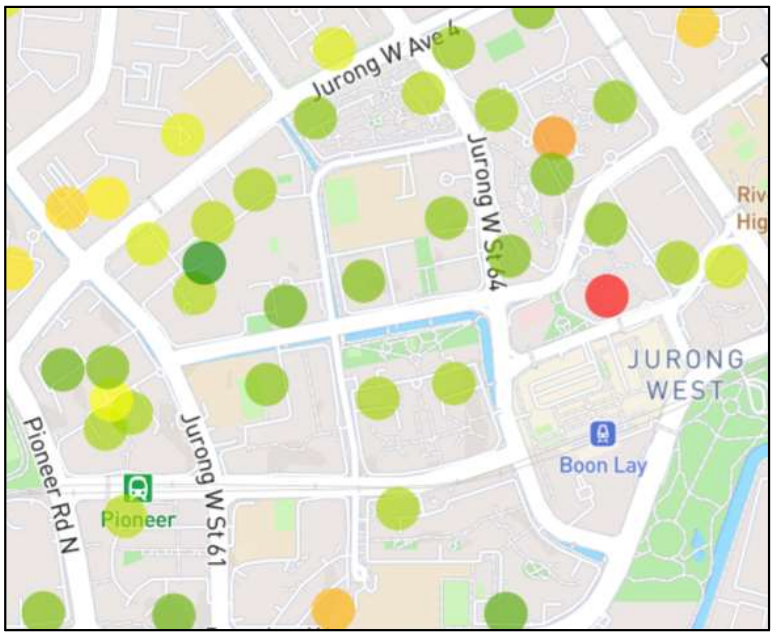


- Guiding Prompts are provided in input boxes to help the user avoid invalid inputs.

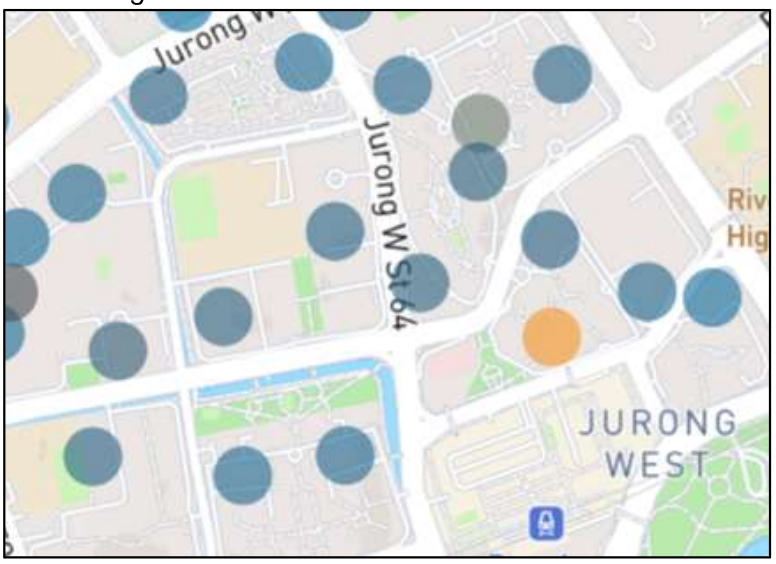


- Different colour schemes are provided to accommodate drivers with colour vision deficiency.

  Red-Green scale:

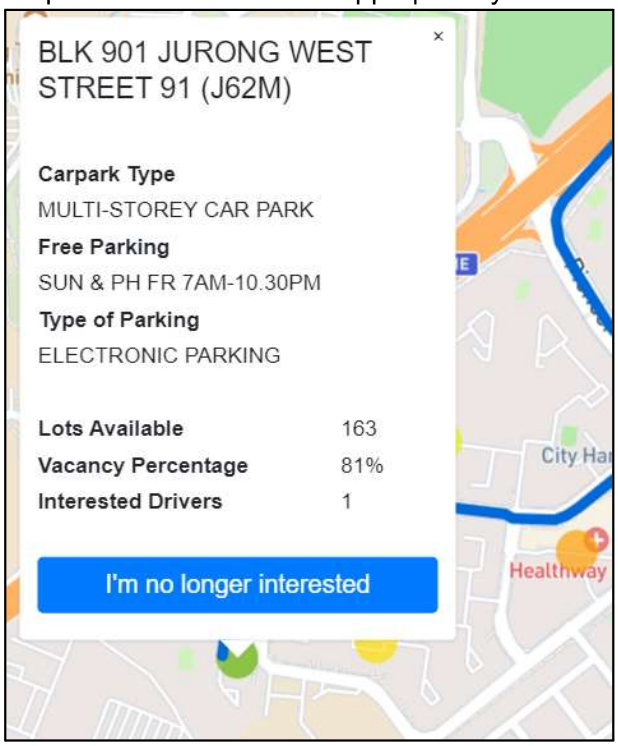


Blue-Orange scale:



- Icons are intuitive and easy to understand icons regardless of the users' literacy.

- Carpark details are shown appropriately for clear reference to the selected carpark.



3.2 Hardware Interfaces

Listed below are some of the hardware requirements:

Driver

- Has a functioning camera
- Has a builtin GPS
- Able to connect to the Internet
- Able to run a modern Internet browser, such as Google Chrome, Microsoft Edge, Mozilla Firefox, Safari

Corporate

- Able to connect to the Internet

- Able to run a modern Internet browser, such as Google Chrome, Microsoft Edge, Mozilla Firefox, Safari

Server
- Multi-threaded or multi-core CPU is capable of handling a large amount of background threads simultaneously
- High performance NIC card able to handle large amounts of network traffic with low latency and high throughput

3.3 Software Interfaces

Refer to the System Architecture Diagrams and Class Diagrams in "Appendix B: Analysis Models" for visualization of specifics detailed in this section.

Client
Our application is served as a web application, meaning it can be viewed by an appropriate web browser. The only possible issues with this is that some web browsers might not support
- **the latest SSL/TLS encryption**: this could lead to some security issues, or the client not being to access the website at all.
- **JavaScript ES6**: the client can access the website but some/all features dependent on JS ES 6 might not work

Therefore, clients should use browsers that fully support the two features mentioned above.

Server

For software components, we would have to consider the frontend and backend aspect:

	Function	Chosen
Frontend	Client-side scripting	JavaScript ES6
	Frontend (Styling) Framework	Bootstrap v5.0
	Helper	jQuery
	Map Canvas Rendering, Geocoding, Route Calculation	Mapbox API
Backend (Python 3)	Web Server	Nginx
	Web Server Gateway Interface	Gunicorn
	Web Framework	Flask
	SQL toolkit and object relational mapper	SQLAlchemy

	SQL Database	SQLite

Our web application uses **Flask**, a lightweight **Python** micro-framework that is ideal for creating Internet-ready applications. Although Flask lacks some features, such as a built-in database, its extension system makes it easy to integrate additional functionalities. We have integrated **SQLite** and **SQLAlchemy** using Flask-SQLAlchemy, a comprehensive package that seamlessly incorporates these tools into the Flask framework.

While Flask does have a built-in server, it is not recommended for production-level applications due to its limited capabilities and lack of security features. Therefore, an appropriate web server and WSGI server is needed to serve the Flask application to the internet. A web server is a software application that receives and responds to HTTP requests sent by web clients, such as browsers, to deliver web content over the internet.

A WSGI server is a type of server software that implements the Web Server Gateway Interface, enabling web applications written in Python to communicate with web servers and be served over the internet. They work with each other to ensure that the Flask application can communicate with the client via the Web. An appropriate choice of web server and WSGI server is **Nginx** and **Gunicorn.**

To program the frontend pages of our web application, we utilise **JavaScript**, which is a versatile scripting language widely used for web development. Specifically, we use **JavaScript 6.0** to leverage features such as the let and const keywords, Promises, fetch, and modules. **Bootstrap** is a highly regarded front-end framework that enables developers to create responsive and mobile-first web applications with ease. Alongside Bootstrap, we use **jQuery**, which is a fast, small, and feature-rich JavaScript library that provides functions helpful in streamlining web development.

The major component of our front-end, more specifically the map/index page, is the map itself. An interactive and responsive map canvas is needed and the **Mapbox API** provides just that. Moreover, it also provides geocoding services, which translate search terms into coordinates, and also route calculation.

3.4 Communications Interfaces

The application's networking communication can be categorised into the following:

1. **Initial request and response**: The client initiates the communication by sending an initial request to the web server. The server responds with an appropriate response in the form of a webpage that the client can display.
2. **Subsequent requests and responses**: After the initial page has been loaded, the client may make subsequent requests to the server for more data. The server responds with the requested data, which the client can display or use.
3. **Content delivery networks**: The client may also make requests to appropriate content delivery networks (CDN) to obtain bootstrap and jQuery modules+styling (CSS).
4. **Mapbox API**: The client may request services from the Mapbox API to obtain relevant mapping data and use its routing and geocoding services
5. **Periodic requests to Data.gov.sg CKAN API**: The server may periodically make requests to the CKAN action API implemented by Data.gov.sg to update carpark information and vacancy data. This ensures that the information provided to clients is always up to date.

HTTP Encryption

All communications mentioned above are accomplished through HTTP requests. However, HTTP is inherently insecure, which means that SSL/TLS encryption must be implemented to secure these communications. This encryption process is what results in HTTPS. When it comes to the client side, it is assumed that the user's web browser will handle SSL/TLS encryption. This means that HTTPS will be used automatically for all requests made from the client's browser to the server.

For the sake of simplicity, we will assume that the APIs that the client or our server uses (CDNs, Mapbox, Data.gov.sg) are encrypted (HTTPS). Therefore, when the client or our server makes a request to these APIs, their browser automatically changes it to HTTPS.

For serving requests from the client, our server can also implement SSL/TLS encryption. This can be done by the web server, which in our case is Nginx.

Implementing REST API

REST API, which stands for Representational State Transfer Application Programming Interface, is a widely used convention for building web services and applications that are flexible, scalable, and easy to maintain. By using REST API in our web application, we hope to standardise the way requests are made to the server and allow anyone with the correct credentials to query our database. This makes our application more extensible and transparent, as other developers can easily build upon our work or integrate with it.

Authentication and control access details have been omitted.

Route	Method	Description
/driver	GET	Get all drivers or specified drivers.
	PUT	Modify specified driver (his interested carpark).
/vehicles	GET	Get all registered vehicles of a specified driver.
	DELETE	Delete a registered vehicle of a specified driver.
/rewards	GET	Get all rewards that are available
	POST	Create a reward
	DELETE	Delete specified reward.
/rewards/claim	POST	Add a claimed reward and modify the appropriate rewards.
/rewards/use	DELETE	Delete a claimed reward.
/points	GET	Get points of a specified user.
	PUT	Modify points of a specified user.

/carparks	GET	Get all car park data.

4. System Features

This section has been organised by use case, with features differentiated for drivers and corporate employees.

4.1 Account Creation for Drivers

4.1.1 Description and Priority

Drivers should create their own accounts to use the map. COE Registration is still needed to access Points, Rewards, and Interest Indication. Each account must be created with a unique email address. Priority is Necessary.

4.1.2 Stimulus/Response Sequences

1. The driver fills the following inputs in the form, with no specific order: Email, First Name, Password, and Re-enter Password.
2. The driver clicks on the submit button.
3. All inputs are filled in and valid. The system creates a valid Driver Account entity.
4. The driver is directed to the home page and logged in.

4.1.3 Functional Requirements

PRE-1: The driver can access the account creation page.
PRE-2: The driver is on the account creation page.
EXC-1: If the inputs are invalid, an error message will be displayed. The driver will have to re-enter the inputs of the form.
POS-1: A new Driver Account with the correct information is created in the user base.
POS-2: The driver can log in to the page if he gives valid credentials.
REQ-1: Each driver can create an account using a unique Email Address. Each driver Account is identified by its unique Email Address.

4.2 Driver Login

4.2.1 Description and Priority

The driver will log in to his account, after which he can save destinations and indicate interest in going to carparks. Priority is Necessary.

4.2.2 Stimulus/Response Sequences

1. The driver enters the Email and Password into the provided text input boxes.
2. The driver clicks on the “Login” button.
3. The website queries the account Database with the input information.
4. The website obtains the driver’s account and stores the information as a dynamic memory of the user’s current session.
5. The website redirects the driver to the home page.

4.2.3 Functional Requirements

PRE-1: The driver has created an account.
PRE-2: The driver is looking at the default page of the website, which is the login page.
EXC-1: If the website does not obtain any account from the database at Step 4, there will be an error flash message and the driver is redirected back to Login Page (i.e. Step 1).
POS-1: The driver is logged in and redirected to the main page.

4.3 COE Registration

4.3.1 Description and Priority

Drivers with accounts are entitled to exclusive rewards and discounts. We want to ensure that drivers won't abuse the system. To do this, we must check that they own a car. We can do this by verifying their COE. Priority is High.

4.3.2 Stimulus/Response Sequences

1. The driver inputs the fields for Full Name, Vehicle Number and COE expiry date.
2. The driver clicks the Submit button.
3. All fields are filled and valid. The system validates the input and redirects to the driver's registered vehicle page.

4.3.3 Functional Requirements

PRE-1: The driver has an account and is logged in.
PRE-2: The driver is on the COE registration page.
ALT- 1: After Step 3, the driver may navigate to the "Registered Vehicles" page to view his registered vehicles.
EXC-1: If the driver clicks the Submit button before completing all input fields, an alert box pops up in the browser, containing names of invalid inputs. The driver will need to redo Step 1.
EXC-2: If the Vehicle Number is already registered, stating "Vehicle with this car plate has already been registered!" will be flashed. The driver will need to redo Step 1.
EXC-3: If the COE expiry date provided has already passed, "COE date input has expired!" will be flashed. The driver will need to redo Step 1.
POS-1: The driver can toggle Interest and access Rewards.

4.4 Colour Vision Deficiency Mode

4.4.1 Description and Priority

The driver can change the colour scale used for bubbles on the map. A suitable colour scale is available for drivers with red-green colour vision deficiency. Priority is Low.

4.4.2 Stimulus/Response Sequences

1. The driver clicks on the button with a spectacles icon.
2. The website changes the colour scale from red-green to blue-orange, updating the colours of all bubbles on the map.

4.4.3 Functional Requirements

PRE-1: The driver has created an account and is logged in.
PRE-2: The driver is looking at the home page of the website.
ALT -1: The driver may repeat Step 2 any number of times, toggling between the 2 colour scales.
POS-1: The colour scheme of the website is updated.

4.5 Querying Carparks

4.5.1 Description and Priority

The driver obtains information about carparks near his destination. Priority is Necessary.

4.5.2 Stimulus/Response Sequences

1. The driver enters the destination postal code and shifts the radius slider to an acceptable radius of his choice.
2. The driver clicks on the “Search Carparks” button.
3. The website queries the Dataset for a list of nearby carparks.
4. The website updates the Map Display by shifting to the specified destination, with an appropriate size zoom to show all bubbles (representing carparks at the respective locations) within the requested radius. The website also shows a List Display of carparks. Each entry of this List includes the carpark name, vacancy, and distance from the destination.
5. The driver selects a bubble to view the details of the carpark.
6. The website displays information about the carpark in a text box pointing at the respective bubble.

4.5.3 Functional Requirements

PRE-1: The driver has created an account and is logged in.
PRE-2: The driver is looking at the home page of the website.
ALT -1: Instead of using the postal code in Step 1, the driver may use the name or description of the location. The website will use the closest match in its database and query the relevant carparks.
ALT -2: The driver may select a carpark by clicking an entry in the list instead of a bubble at Step 5. Step 6 proceeds without changes.
ALT -3: After completing Step 6, the driver may continue viewing other carparks by repeating Steps 5 to 6. ALT-1 remains a viable step for viewing the carpark details.
ALT -4: After completing Step 6, the driver queries about carparks near other destinations by repeating Step 1.
EXC-1: If the destination cannot be found, a message stating “You have entered an invalid location or we were unable to find your location. Please try again.” The Map and List displays will not be updated. The driver may repeat Step 1, choosing another destination.

POS-1: The Map Display has been adjusted to show the relevant bubbles, and the List is updated with the relevant carpark entries.
REQ-1: The radius slider has a lower bound of 1km and an upper bound of 10km. This ensures that there will be at least 1 relevant carpark (in Singapore's context), and the website can render the relevant carparks.

4.6 Updating Current GPS Location

4.6.1 Description and Priority

Drivers must update the current location as seen by the website to view accurate navigation directions from their actual current location to the selected carpark. Priority is Medium.

4.6.2 Stimulus/Response Sequences

1. The driver clicks on the button with a crosshair target icon.
2. The website asks for permission to access the device's GPS location.
3. The driver enables location access.
4. The website obtains the GPS location and updates the location of the geolocation pin icon on the map. The pin represents the location retrieved from the device.

4.6.3 Functional Requirements

PRE-1: The driver has created an account and is logged in.
PRE-2: The driver is looking at the home page of the website.
EXC-1: The driver does not provide permission for location access at Step 3. The website is unable to retrieve the device's GPS location and the location of the geolocation pin will not be updated.
POS-1: The geolocation pin is now pointing at the device's GPS location on the map.
REQ-1: The device has a functional GPS that can be accessed by the website.

4.7 Interest Indication & Navigation

4.7.1 Description and Priority

Drivers must indicate interest and park at the indicated location to receive reward points. Indicating an interest in the carpark provides a more accurate judgement on the vacancy of the selected carpark. Navigation to the selected carpark will be displayed after interest has been indicated. Priority is High.

4.7.2 Stimulus/Response Sequences

1. The driver selects a carpark from the carparks identified by the website after the carpark query.
2. The driver clicks on "I'm interested".
3. The website updates the driver's interest and updates the total number of drivers who are interested in visiting the carpark.

4. The website displays the directions from the current location to the selected carpark on the map.

4.7.3 Functional Requirements

PRE-1: The driver's has been authenticated.
PRE-2: The driver is logged in.
PRE-3: The driver has queried about carparks near his destination.
PRE-4: There is at least 1 carpark identified by the system based on the parameters provided by the driver during his query.
ALT -1: After Step 4, the driver may click on "I'm no longer interested". The number of interested drivers to that carpark will be updated and the navigation directions will be removed from display. Step 2 may be repeated again.
ALT -2: After Step 4, the driver may repeat Steps 1 to 2 for a different carpark. The number of interested drivers for the relevant carparks will be updated and the navigation directions on the map will also be updated to the most recent carpark of interest.
EXC-1: If the driver is not authenticated, the interest indication button will not be available.
POS-1: The website will display an updated count of the number of drivers who have also indicated an interest in the same carpark. The driver's interest will be saved in the system.
POS-2: Navigation directions will be displayed.
POS-3: The website will display the "I have parked" button.
REQ-1: For accurate navigation directions, the driver must have completed task 4.5, Updating Current GPS Location. Otherwise, navigation directions will be displayed from some default location set on the map, in the middle of Singapore.

4.8 Reward Points Collection

4.8.1 Description and Priority

To get reward points after the driver has indicated his interest in parking at a parking location and successfully parked at that location. Priority is Medium.

4.8.2 Stimulus/Response Sequences

1. The driver clicks on "I have parked".
2. The website redirects the driver to the file upload page, "Verify Parking".
3. The driver chooses and uploads a photo of his car parked at that location from his device.
4. The driver clicks on the "Submit" button
5. The website displays a flash message, "You have received 1 point!", and redirects the driver to the home page.

4.8.3 Functional Requirements

PRE-1: The driver has been authenticated.
PRE-2: The driver is logged in.
PRE-3: The driver has indicated an interest in parking at the location.
EXC-1: If the driver exits the "Verify Parking" page without selecting the

"Submit" button, no points will be awarded and the driver needs to re-indicate his interest at the selected carpark.

POS-1: The driver receives Reward Points.

REQ-1: The driver's hardware is capable of uploading image files.

REQ-2: If the driver is not authenticated, the interest indication button will not be available. This feature cannot be accessed.

REQ-3: If the driver has not indicated interest at any carpark, the "I have parked" button will not be available. This feature cannot be accessed.

REQ-4: The "Submit" button cannot be clicked if no image file has been uploaded.

4.9 Account Creation for Corporate

4.9.1 Description and Priority

Corporates must create their accounts to add Rewards for drivers to redeem. Each account must be created with a unique business UEN. The corporate employee will be the actor of this use case. Priority is High.

4.9.2 Stimulus/Response Sequences

1. The corporate employee fills the following inputs in the form, with no specific order:
   UEN, Company Name, Password, Re-enter Password.
2. The employee clicks on the submit button.
3. All inputs are filled in and valid. The website creates a valid Corporate account entity.
4. The website redirects the employee to the Reward Creation page. A message stating "Corporate Account successfully created!" is flashed.

4.9.3 Functional Requirements

PRE-1: The employee can access the account creation page.

PRE-2: The employee is on the account creation page.

EXC-1: If the UEN already has an associated account, an error message stating "An account has already been created with this UEN!" will be displayed. The employee must re-enter the inputs of the form.

EXC-2: If the passwords are invalid, such as not being equal or not being sufficiently secure, an error message will be displayed.The employee must re-enter the inputs of the form.

POS-1: A new Corporate Account with the correct information is created in the user base.

POS-2: The Corporate Account is logged in and on the Rewards Creation page.

REQ-4: Each corporate can create an account using a unique UEN. Each corporate Account is identified by its unique UEN.

4.10 Corporate Login

4.10.1 Description and Priority

Corporate employees must log in to the Corporate Account to create rewards. Priority is High.

4.10.2 Stimulus/Response Sequences

1. The employee enters UEN and password.
2. Database checks if UEN exists.
3. If UEN exists, the database checks if the password is correct.
4. If the password is correct, login is successful and the Rewards Creation page is rendered.

4.10.3 Functional Requirements

PRE-1: Corporate account already exists.

EXC-1: If UEN does not exist in the database, an error message is flashed, prompting the employee to re-enter details.

EXC-2: If the password is incorrect, login is unsuccessful, and the login page prompts the employee to re-enter details.

POS-1: Corporate Account is logged in and the employee is redirected to the rewards creation page.

4.11 Create Rewards for Corporate

4.11.1 Description and Priority

Employees can create rewards to be issued by the corporate, for drivers to redeem with their points. Priority is Medium.

4.11.2 Stimulus/Response Sequences

1. Employee completes the following fields before clicking “Add Reward”: Reward Title, Reward Expiry, Reward Category, Reward Details, Number of Rewards Available to Claim, and Cost of Reward (in SGD).
2. Website verifies the eligibility of the Reward by the fields.
3. If the Reward is eligible, the Reward is posted to the Rewards Page. A message stating “Reward Created!” is flashed.

4.11.3 Functional Requirements

PRE-1: Employee account exists and is logged in.

ALT- 1: After Step 3, the employee may continue to create rewards, repeating Steps 1 to 3.

ALT- 2: After Step 3, the employee may navigate to “View Rewards” to view the rewards created by the account. Rewards can be deleted by clicking on the “x” button of each reward.

EXC-1: If the Reward is ineligible, the Reward creation page prompts the employee to recreate a Reward.

POS-1: The Reward will be posted to the rewards page.

4.12 View Rewards

4.12.1 Description and Priority

The driver views available rewards that can be claimed with reward points, and also view the rewards he has claimed. Priority is Medium.

4.12.2 Stimulus/Response Sequences

1. The driver logs into his account, using his username and password.
2. The driver selects "View Rewards", which displays his reward point, the available rewards and his claimed rewards.
3. The driver exits from the “Rewards” page and returns to the “Home” page.

4.12.3 Functional Requirements

PRE-1: The driver has an authenticated account and is logged in.
PRE-2: The driver is on the default page.
POS-1: There are no changes to the reward points balance.

4.13 Claim Reward

4.13.1 Description and Priority

The driver claims rewards using his reward points. Priority is Medium.

4.13.2 Stimulus/Response Sequences

1. The driver clicks on the “Claim Reward” button to claim a reward from the list of available rewards.
2. The reward is added to the list of claimed rewards and the reward points balance is updated. A message stating “You have successfully claimed the reward is flashed”.

4.13.3 Functional Requirements

PRE-1: The driver has an authenticated account and is logged in.
PRE-2: The driver is on the “View Rewards” page.
ALT -1: The driver chooses not to claim any reward on Step 1. The driver is viewing available rewards. This is feature 4.12.
EXC-1: The driver does not have sufficient points to claim the reward at Step 1. A message stating “Sorry, you do not have enough points to claim this reward!” is flashed. The driver may repeat from Step 1 for a different reward.
POS-1: The driver receives the reward he chose and his reward points balance is updated.

4.14 Use Reward

4.14.1 Description and Priority

The driver uses a claimed reward. Priority is Medium.

4.14.2 Stimulus/Response Sequences

1. The driver selects "Use Reward" on a claimed reward.
2. The website displays a QR code.
3. The driver scans the QR code, effectively using the reward.
4. The driver clicks on the “x” button of the box containing the QR code.
5. A message stating “You have successfully used this reward!” is flashed. The reward is removed from the list of claimed rewards.

4.14.3 Functional Requirements

PRE-1: The driver has an authenticated account and is logged in.
PRE-2: The driver has claimed at least 1 reward.
POS-1: The used reward will be removed from the list of claimed rewards.
REQ-1: The driver can only use the reward when the QR code is shown. The driver can only see the QR code once, upon clicking “Use Reward”. Thereafter, the claimed reward will be considered used and can no longer be used again.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

We aim for this website application to be a quick and convenient tool for drivers to find suitable carparks. The nonfunctional requirements specified below are necessary for achieving this goal.

1. The system must cater to different screens and layouts for accessibility and convenience of users.
    a. There must not be unwanted overlapping of display objects
    b. Different layouts include and are not limited to mobile phones, tablets or computer monitors, not restricted to the operating system.

2. The application must be responsive, easy-to-learn and easy-to-use.
    a. 85% of first-time users must be able to make a query within 5 minutes of entering the website, after login.
    b. The website must be able to handle multiple users simultaneously. The website must continue functioning without errors while 100 users run the application concurrently.

3. The application must be accessible to those with disabilities.
    a. Font sizes must be appropriate, as well as sufficient contrast for readability.
    b. Suitable colour scale used for drivers with Colour Vision Deficiency, specifically Red-Green deficiency (due to the default colour scale).

4. The website must be able to run on modern web browsers and be minimally backward compatible with old web browsers.

5. The system must provide an assistive guide for drivers.
    a. This will come in the form of a step-by-step guide and an FAQ page.

5.2 Security Requirements

The following security requirements must be complied with to ensure the security of user data, the protection of soure code and the integrity of software functionality and software memory.

1. The system must comply with all relevant data privacy regulations, including the Personal Data Protection Act (PDPA).
2. The system must store user data securely and use it only for the purposes of system functionality. No data should be accessed where unnecessary. No data should be shared or sold without the authorisation of the user. Sensitive attributes are stored as private attributes and can only be accessed and mutated through necessary function calls.
3. The system must allow drivers to delete their data from the system.
4. The system must urge drivers to set strong passwords to ensure that their accounts are more secure. No account can be created without a password of at least 8 characters long, consisting of alphanumeric characters and at least 1 special character, such as ‘!’, ‘@’ etc.
5. The system must implement measures to prevent unauthorised access to user accounts. Hashing is used on all passwords before storage in the database.

6. The system must implement preventative measures to prevent a buffer overflow attack. Users are not allowed to input fields related to account login / sign up with more than 150 characters, preventing an overflow of the buffer into vulnerable code memory.

5.3 Software Quality Attributes

We can elicit the software quality attribute requirements based on the various stakeholders and users:

Stakeholder	Quality Attributes	Elaboration
User (Driver)	Correctness/Reliability	- The car parking information should be reliable and should also be updated frequently. Currently, the vacancy of a carpark is updated every 5 minutes with the API provided by GovTech. - The “indicate interest feature” should allow the driver to navigate to the carpark. - The points that users gain should not be taken away from the user - The user should be able to use their points to redeem their rewards and also use the redeemed rewards with no technical difficulty.
	Portability	The user should be able to use this web application on any device, anywhere. This requirement is mostly related to the responsive UI requirement.
	Accessibility	The needs of more people are considered when using the application. - Adjustment of color-coding is supported for people with colorblindness when using the application - Font of the website can be adjusted.
	Security	The user data should be secure. This is concerned with 2 aspects: security of data transfer over different components and security of data storage.
User (Corporate)	Correctness/Reliability	- There should be no bug or malfunction which could allow the drivers to misuse the points/ rewards - All rewards should display the correct information and be shown to every driver
	Portability/Accessibility	Same as mentioned in the User (Driver) section.
Developer	Readability	- Proper documentation should be written to describe the web application

	- This document, the system architecture, class diagram, etc should serve as ample documentation of the web application - The code should be properly organised to make it easier for the developer to navigate it - The code should follow a proper style guideline (e.g. Google Style Guides) - The code should be also be documented properly
Maintainability	The code should use Object-Oriented Designs or any other suitable design to reduce code repetition, increase reuse of code and for extensibility. The code should be refactored frequently to increase maintainability.
Extensibility	There should be a way to use our services/database, possibly via an API.
Testability	Each use case has a set of suitable test cases to verify the program with expected results, regardless of future updates and optimization.

5.4 Business Rules

The following rules are adhered to, to prevent any blatant misuse of the website, such as claiming rewards where ineligible or claiming of multiple rewards by the same physical individual. Misuse of the website will damage the trust of corporations in our website, and inaccurate representation of interests in some carparks, and an overall failure of the project.

- The application should only be used when the user uses a valid email address to register for an account.
- The application should only display the number of available parking lots in HDB parking lots.
- The driver must verify COE before they can claim points.
- The number of interested drivers should be displayed separately from the actual number of available parking lots to avoid confusion.
- Points should only be given to drivers if the photos they upload proving they have parked at where they indicated interest are verified.
- Corporates must have a valid Unique Entity Number (UEN) to register an account.
- The vouchers provided should be verified before being made available for the drivers to redeem.

- The security of users' data should be protected in accordance with relevant data privacy regulations, including the Personal Data Protection Act (PDPA).

6. Other Requirements

This project was completed in alignment with the Smart Nation movement and the Data-Driven Smart Nation Competition launched by ZEA. While some of our features assume users are acting in good faith, we believe that the collaborations with the Government and other stakeholders can help reduce the need of our assumptions. Furthermore, our dataset is limited to HDB carparks. Listed below are some of the possible extension that can improve our project.

1. Validity of COE Registration:
   Tentatively, we have no way of verifying the authenticity of a registered COE. A possible solution would require the assistance of the Land Transport Authority (LTA). LTA can utilise their database, and together with the use of Singpass, verify the driver's COE.

2. Validity of Parked Car image:
   Tentatively, we cannot verify that the image uploaded feature the driver's car parked at the correct location. By collaborating with LTA, we can potentially use the vehicles' In-Vehicle Unit (IU) to detect entry of the vehicle into the carpark. IUs are currently used for payment at Electronic Parking System (EPS) carparks and Electronic Road Pricing (ERP) gantries. Furthermore, the use of IUs can allow us to provide accurate information at existing carparks, and also carparks of malls that are not governed by HDB.

3. Availability of more Carparks:
   By collaborating with more companies and/or organisations, the website will be able to feature more carparks, and thus allow drivers to choose a more convenient location for parking. Currently, drivers visiting a mall will not be shown the mall's carpark since it is not in our available dataset. We could potentially collaborate with CapitaLand to obtain the carpark information of all CapitaLand malls.

Appendix A: Glossary

Definitions of relevant terms are as specified (in alphabetical order):

Account	The entity in the code that represents a user. It stores and accesses the relevant information of the user from the database. Driver accounts and Corporate accounts are distinct from each other with different function access.
API	An abbreviation for Application Programming Interface. APIs in this project are used to sync datasets used in real-time.
Authenticated	Any driver account that has successfully registered his COE on the website.
Bubbles	Visual tags pointing to specific points on the map, which can be clicked on to reveal more information.
Carpark	Carparks in HDB buildings, popular shopping centres, tourist destinations, or hotels.
Claim	Analagous to Redeem.
COE	An abbreviation for Certificate of Entitlement. The COE is legally required for each owned vehicle in Singapore for the vehicle to be permitted for use on the road.
Colour Scale	A set of colours that are connected bidirectionally in the colour spectrum.
Company	Analagous to Corporation.
Corporate/ Corporations	A type of user of the website. A corporate is expected to use the website for the purposes of creating rewards to be claimed by drivers, as a means of publicity and outreach to potential customers in their business.
Current Location	The GPS location obtained by the website from the driver's device. This location will be treated as the initial point of the driver's journey.
Default Page	Refers to the driver login page that renders upon loading the website.
Destination	The target location of the user. This will be in any vehicle-accessible area of Singapore.
Developer	Any person or group working on the source code of this software, with the purpose of improving, upgrading, or developing the website.
Directions	The route displayed on the map canvas, from the driver's current location to his destination.
Distance	Most often refers to the distance between the driver's destination and some referenced carpark.
Driver	A type of user of the website. A driver is expected to use the website for the purposes of finding a suitable carpark near their destination, obtain points,

	claim rewards from the website and use the rewards in their accounts.
Employee	A person who uses the Corporate account. A corporate employee is assumed to have the permission of the company he represents to access and use the corporate account. All rewards created by the employee through the corporate account will be deemed as actions decided by the company as a whole.
External Stimuli	Actions taken by actors on to the website.
Government	The Government of Singapore.
GPS	An abbreviation for Global Positioning System.
HDB	Housing Development Board, a statutory board responsible for public housing in Singapore. HDB is also responsible for HDB carparks.
Home Page	Refers to the page that renders immediately after the driver logs in, containing the map.
Interest	An indication of the driver's intent to park at a chosen carpark.
Journey	To travel from the current location to the carpark of interest.
Location	Refers to some selected carpark. Not to be confused with "Current Location".
Lots	Refers to the parking lots of a carpark.
Map	A geographical representation of Singapore. Used commonly for the locality of the user's destination.
Navigation	Analagous to Directions.
PDPA	An abbreviation for Personal Data Protection Act. This Act is written by the Government of Singapore to protect the safety and security of users' personal data in some particular setting.
Points	The currency used in the software, rewarded to drivers who have completed a set of requirements.
(Potential) Customers	The target audience of the company. This may be subgroups of the drivers who use the website, and may be interested in the products and/or services of the mentioned company.
Radius	The linear distance from the destination, for all 360°.
Redeem	To obtain some reward in exchange for points.
Reward	An incentive given to drivers, in exchange for their points. Rewards can created by corporates who use our website.
Stakeholder	Relevant actors of the website.

System Analagous to the Website.

UEN An abbreviation for Unique Entity Number. All registered companies in Singapore has a UEN.

User Person who is using the website. This may refer to either the driver or the corporate employee.

Vacancy The percentage of parking lots in a carpark that are available for drivers to park in.

Vouchers A type of reward. This could be discounts or coupons that can be used when consuming a product and/or service by the corporation.

Website The online pages where the intended features are available and accessible.

Appendix B: Analysis Models

1. Use Case Diagram:
   A top-level diagram showing all use cases of the software, and the relevant actors on the website. Refer to the README to locate timage file of the Use Case Diagram.

2. Sequence Diagrams:
   Each use case or system feature has its associated Sequence Diagram. Refer to the README to locate all image files, one for each Sequence Diagram.

3. Class Diagrams:
   A static representation of all entities in code. The diagram shows how different objects and aspects of the code interact with each other.
   2 Class Diagrams have been used: one to represent the Front-End Development, another to represent the Back-End Development. You are recommended to reference the System Architecture along with Section 3.3, Software Interfaces. Refer to the README to locate all image files, one for each Class Diagram.

4. System Architecture:
   A visual representation of the code framework. The system architecture states the technical specifications, requirements and protocols used in development. You are recommended to reference the System Architecture along with Section 3.3, Software Interfaces. Refer to the README to locate the image file of the System Architecture.

5. Test Cases:
   A list of test cases to verify the functionalities of the website. The test cases can be used whenever the system is upgraded, ensuring the core functionality remains intact. Refer to the README to locate the PDF document containing all the test cases.