Software Requirements Specification

for

CarParkGoWhere

**Version 1.0 approved**

**Prepared by Ong Jing Jie, Seow Jing Woon, Cheng Yi Feng,  
Jumana Haseen, Damien Lee Han Wei, Haoren**

**Nanyang Technological University**

**09/09/2024**

**Table of Contents**

[**1. Introduction 1**](#_heading=h.1fob9te)

[**1.1 Purpose**](#_heading=h.3znysh7) **1**

[**1.2 Document Conventions 1**](#_heading=)

[**1.3 Intended Audience and Reading Suggestions 1**](#_heading=)

[**1.4 Product Scope 2**](#_heading=)

[**1.5 References 2**](#_heading=)

[**2. Overall Description 2**](#_heading=)

[**2.1 Product Perspective** 2](#_heading=h.2s8eyo1)

[**2.2 Product Functions 3**](#_heading=)

[**2.3 User Classes and Characteristics 3**](#_heading=)

[**2.4 Operating Environment 4**](#_heading=)

[**2.5 Design and Implementation Constraints 4**](#_heading=)

[**2.6 User Documentation 4**](#_heading=)

[**2.7 Assumptions and Dependencies 5**](#_heading=)

[**3. External Interface Requirements 5**](#_heading=)

[**3.1 User Interfaces 5**](#_heading=h.2jxsxqh)

[**3.2 Hardware Interfaces 5**](#_heading=)

[**3.3 Software Interfaces 5**](#_heading=)

[**3.4 Communications Interfaces 5**](#_heading=)

[**4. System Features 6**](#_heading=)

[**4.1 Car Park Search & Routing 6**](#_heading=h.dsj5hx1wfqig)

[**4.2 Car park Map Display & Sidebar Sorting/Recommendation 7**](#_heading=h.or4a374sskge)**-8**

[**4.3 Car Park Filtering**](#_heading=) **8-9**

[**4.4 Account System**](#_heading=h.pvdsxrexzpcl) **9-10**

[**5. Other Nonfunctional Requirements 11**](#_heading=h.3whwml4)

[**5.1 Performance Requirements**](#_heading=h.2bn6wsx) **11**

[**5.2 Safety Requirements 11**](#_heading=)

[**5.3 Security Requirements 11**](#_heading=)

[**5.4 Software Quality Attributes 12**](#_heading=)

[**5.5 Business Rules 12**](#_heading=)

[**6. Other Requirements 12**](#_heading=)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Ong Jing Jie | 4/9/2024 | Initial Version | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

With the high vehicle ownership numbers in Singapore, Singapore faces a significant parking problem, particularly in densely populated areas such as the Central Business District area. To attempt to resolve this issue, Team 46 aims to develop a website that allows users to find the nearest, cheapest or fastest car park based on their destination. This project will be considered complete when the website is tested and approved for release by the ZEA. This project supports the ZEA in eliciting innovative applications in support of the Smart Nation initiative.

## Document Conventions

The body text of this document is written in standard font Arial, 11-point size for readability. All headings and subsections are written in bold and numbered hierarchically e.g. 1, 1.1, 1.1.1. Code snippets, user inputs and program outputs are written in font roboto mono, 11-point size. Prioritisation is explicitly stated for each requirement, if no priority for the detailed requirement is explicitly declared then it is assumed that they inherit the priority of the higher-level requirement. Requirements have separate parameters where each will be ranked from one to ten, ten being of higher weightage. Each functional requirement will have a unique code in the form “REQ-1” to facilitate referencing and traceability. Each sub-requirement will follow the parent requirement with an additional number in the form “REQ-1.1”. Each version of the SRS document is assigned a version number that will be incremented with each revision. The version number is listed under the project name on the first page.

## Intended Audience and Reading Suggestions

*This document is intended for the reading of developers, testers and project managers from ZEA.*

*Organization of the SRS Document:*

* *Introduction*
* *Overall Description*
* *External Interface Requirement*
* *System Features*
* *Other Nonfunctional Requirements*
* *Other Requirements*

*The suggested sequence for reading the document is from the start of the document to the end of the document, from section 1 to section 6 in chronological order.*

## Product Scope

The purpose of the software is to allow users to use an online map and GPS coordinates of the vehicle to find the most suitable parking lots in car parks near the user’s final destination. We aim to reduce the hassle of users such as:

* Having to drive to the car parks to check the prices
* Arriving at car parks only to find no available spaces and facing long wait times.
* The trouble of finding the cheapest and fastest route to the car park.

The software aims to provide users with parking rates at the start of the journey, assist the user in choosing the car parks based on the user's destination, price and distance of the car park from the destination, and provide the fastest route to the car park.

## 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.>*

# Overall Description

## Product Perspective

This website has been developed as a response to the need for a comprehensive parking solution in Singapore since motorists face difficulty finding affordable and nearby parking spaces that are available. Although existing solutions such as Park&Go @SG, Breeze and SGCarMart’s Carpark Rates App provide similar functionalities, they still have certain limitations. For example, some of these applications may lack real-time traffic data accuracy or seamless user experience.  
  
This product is not a follow-on member of this existing product family but it could be considered an enhancement over their systems. The purpose of this system is to build an all-in-one platform, which integrates and improves their strengths such as more accurate real-time data provision and cost calculation, while addressing their weaknesses such as lack of the need for more intuitive user interface or lack of alternative suggestions.

The website is designed to function as a central hub with aggregated parking information from multiple authoritative data sources such as the Urban Redevelopment Authority (URA), Housing Development Board (HDB) and map data providers. These data feed our system with accurate real-time data, which will then analyse and provide comprehensive information to users.

## Product Functions

Major Functions the product must perform:

1. Location-based services
   * Display a list of car parks that are near the destination within a certain distance radius
   * Display the fastest route to the chosen car park
2. Price comparison
   * Display pricing rates of the car parks near the destination sorting by lowest to highest
3. Search and filtering
   * Allow user to search for car parks near their destination
   * Allow user to filter for prices of car park and nearest car park from destination
4. Real-time availability
   * Display real-time availability of parking spaces
   * Allow users to refresh parking availability information
5. System Navigation Support
   * Show users the best car park based on the shortest distance to the destination from the car park and the cheapest price.

## User Classes and Characteristics

There are several user classes that we can differentiate, namely

2.3.1. Frequent users

| Frequency of Use | Daily or multiple times a day |
| --- | --- |
| Product Functions Use | Nearest Car park, Fastest Car park |
| Technical Expertise | Moderate to high, familiar with navigation tools |
| Importance | High |

2.3.2. Casual Users

| Frequency of Use | Weekly or less |
| --- | --- |
| Product Functions Use | Nearest Car Park, cheapest Car Park |
| Technical Expertise | Basic to moderate |
| Importance | Medium as they form a significant portion of the Users but may not be using the tool frequently |

2.3.3. Tourists or Visitors

| Frequency of Use | High but short-term |
| --- | --- |
| Product Functions Use | Nearest Car Park, cheapest Car Park |
| Technical Expertise | Varied, but the app should be user-friendly and intuitive |
| Importance | Medium |

2.3.4. Business Users

| Frequency of Use | Regular to frequent usage depending on business needs |
| --- | --- |
| Product Functions Use | Cheapest Car Park |
| Technical Expertise | High |
| Importance | High, as they may represent a significant revenue stream |

## Operating Environment

The website operates in Windows Vista and above, Linux, macOS 12 that supports Chrome or Firefox.

## 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 organisation will be responsible for maintaining the delivered software).>*

## 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.>*

## 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).>*

# External Interface Requirements

## 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.>*

## 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.>*

## 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.>*

## 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.>*

# System Features

## 4.1 Car Park Search & Routing

4.1.1 Description and Priority

This allows users to input their current/starting location and destination to search for available nearby car parks. The system will then display the route from the starting location to the car park. It will also display the walking route from the car park to the intended destination. This feature is of High priority.

| **Components** | **Rating (out of 10)** | **Description** |
| --- | --- | --- |
| **Benefit** | 10 | Core functionality for users to search and visualise the route of their journey. |
| **Penalty** | 10 | Without this feature, users would not be able to find suitable car parks at their destination. |
| **Cost** | 7 | Requires integration with OneMap, HRA and HDB car park APIs to dynamically update results. |
| **Risk** | 8 | High technical complexity due to the need for real-time map updates and integration with location and car park data. |

4.1.2 Stimulus/Response Sequences

Stimulus 1:

User: Input their current and destination locations (address, postal code or coordinates) manually or automatic detection of users’ GPS coordinates

System: Retrieves the route from the current location/ starting location to the suitable car park and displays it as a highlighted route on the map. The walking route from the car park to the destination will be highlighted in a different colour from the driving route.

4.1.3 Functional Requirements

REQ-1: Users must be able to query the car park system. The query can either be of address line, coordinates or postal code.

REQ-2: The system must display the driving route from the current location to the car park along with the walking route from the car park to the destination.

## 

## 4.2 Car park Map Display & Sidebar Sorting/Recommendation

4.2.1 Description and Priority

This will display the available car parks on the map and mark available car parks with an identifier. This will also sort and rank the car parks based on set users’ filter parameters in the sidebar. The first car park shown at the top of the sidebar will be recommended by the system. This feature is of High priority.

| **Components** | **Rating (out of 10)** | **Description** |
| --- | --- | --- |
| **Benefit** | 10 | Helps users easily find the most suitable car park at a centralised sidebar. |
| **Penalty** | 10 | Without this feature, users would not be able to visualise where the car parks are. |
| **Cost** | 6 | Moderate complexity, mainly requiring sorting algorithms and integration with the filtering mechanism. |
| **Risk** | 3 | Low risk since the sorting logic is straightforward, though usability must be carefully considered. |

4.2.2 Stimulus/Response Sequences

Stimulus 1:

User: Car park search & Routing (Stimulus 1)

System: Displays an interactive map that marks each available car park with a coded identifier. The sidebar will display all detailed information of the car park. The sidebar also displays only car parks that are at the current level of detail and is sorted by default settings, in ascending order.

Stimulus 2:

User: Clicks on a marker on the interactive map

System: Highlights the marker by changing to a different colour and the respective sidebar dropdown of the car park will be expanded to display the detailed information.

Stimulus 3:

User: Selects to sort by options; hourly rate, driving distance to car park, walking time to destination, driving time to car park.

System: Sorts and ranks according to the feature that the user has selected; in ascending or descending order

# 

# 4.2.3 Functional Requirements

REQ-1: Maps displayed must be interactive.

REQ-1.1: The user must be able to zoom in on the map

REQ-1.2: The user must be able to zoom out on the map

REQ-1.3: User must be drag the map in order to move in all directions

REQ-2: Available car parks must be marked with a unique identifier e.g. location marked in green or labelled “CP01”.

REQ-3: Selecting a car park will highlight the marker on the map and expand the respective car park dropdown to show hourly rate, driving distance to the car park, walking time to destination, driving time to the car park

REQ-3.1: Selecting a marker will highlight the marker and expand the car park details in the drop-down at the sidebar.

REQ-3.2: Expanding the car park drop-down in the sidebar will highlight the respective car park marker.

REQ-4: Sidebar allows users to set sorting parameters.

REQ-4.1: Allows users to specify either “ascending” or “descending” order.

REQ-4.2: Allows users to choose according to either the hourly rate, walking distance to the destination or number of available parking lots.

REQ-5: Recommended car park toggle and settings.

REQ-5.1: The recommended car park will be based on a formula of the sum of weightage multiplied by the components.

REQ-5.1.1: Weights will be set by default but can be adjusted by the user.

REQ-5.1.2: There will be a toggle to turn off/on recommendations by using the filter feature.

REQ-5.1.3: The sum of weights will always be of integer 1.

## 4.3 Car Park Filtering

4.3.1 Description and Priority

This allows users to filter car park search results. The filtering ensures that users can quickly narrow down to the most suitable car park. This is a Medium-priority feature.

| **Components** | **Rating (out of 10)** | **Description** |
| --- | --- | --- |
| **Benefit** | 8 | Ensures users find suitable car parks efficiently. |
| **Penalty** | 6 | Without this feature the system can still function, but may result in users manually sifting through too many options, increasing the time to find the most suitable car park. |
| **Cost** | 4 | Low development complexity. |
| **Risk** | 4 | Low risk, as the feature involves straightforward input filtering. |

4.3.2 Stimulus/Response Sequences

Stimulus 1:

User: Car park Search and Routing (Stimulus 1)

System: Presents a list of car parks based on default filter settings

Stimulus 2:

User: Select and set filter options; hourly rate, driving distance to car park, walking time to destination, driving time to car park

System: Updates the car park results list in real-time based on filter preference

Stimulus 3:

User: Modifies one or more filters

System: Dynamically refreshes the available car parks to match the new criteria.

4.3.3 Functional Requirements

REQ-1: Given the car parks at the current level of detail, only those that fulfil the filter parameters will be displayed on UI.

REQ-1.1: There must be a toggle button to display hidden car parks with greyed-out font.

REQ-2: Filter parameters must be able to be toggled and set with numeric maximum and minimum.

REQ-2.1: The hourly rate must be accurate to double the nearest $0.10.

REQ-2.2: The driving distance to the car park must be accurate to the nearest 0.1km.

REQ-2.3: Walking time must be accurate to the nearest minute.

REQ-2.4: Driving time to the car park must be accurate to the nearest minute.

## 4.4 Account System

4.4.1 Description and Priority

This feature allows users to create personal accounts to save preferences, and store recent parking and search history. It is of Low priority, providing added convenience for frequent users.

| **Components** | **Rating (out of 10)** | **Description** |
| --- | --- | --- |
| **Benefit** | 7 | Enhances user experience by personalising the system and reducing the need to re-enter information. |
| **Penalty** | 3 | Without this feature, users may have to input preferences and locations repeatedly, leading to frustration. |
| **Cost** | 5 | Standard complexity, requiring user authentication and basic account management functions. |
| **Risk** | 4 | Low risk, but protection of sensitive data such as passwords must be handled securely. |

4.4.1 Stimulus/Response Sequences

Stimulus 1:   
 User: Creates a new account.  
 System: collects user details and creates a profile.  
 Stimulus 2:   
 User: Log in to their account  
 System: retrieves preferences, recent parking and search history.  
 Stimulus 3:   
 User: Updates their saved preferences  
 System: Saves updated preferences  
 Stimulus 4:  
 User: Guest Account  
 System: Does not load any previous preferences or history

4.3.3 Functional Requirements

REQ-1: The system must allow users to create new accounts.  
 REQ-1.1: The user must input a string password from 8 to 20 characters.  
 REQ-1.2: Passwords must not include blank spaces.  
 REQ-1.3: Passwords must include at least one upper and lower case letter.  
 REQ-1.4; Passwords must include at least one special character.  
 REQ-1.5: Passwords must include at least one number.

REQ-2: The system must allow users to log in to their accounts.

REQ-3: The system must allow users to update their saved preferences.  
 REQ-3.1: Saved preferences allow users to adjust weightage to parameters.

REQ-4: The system must allow users to use the application without logging in.

REQ-5: The system must not allow identical emails.

REQ-6: The system must be able to validate corresponding username and password.

# Other Nonfunctional Requirements

## Performance Requirements

* + 1. The product should provide search results within a certain timeframe. The search for the nearest/cheapest/fastest car park should take less than 10 seconds and less than 15 seconds for detailed information such as the price of the car park, estimated time of arrival to the carpark as well as the availability of the carpark

* + 1. The page load time should not exceed 2 seconds under normal load conditions and should not exceed 3 seconds under high load conditions.

* + 1. The system should also have an uptime of 99.9% or higher, ensuring that it is available most of the time. The system should handle unexpected errors gracefully and provide reliable access to parking information.

* + 1. After a system reboot, full system functionality should be achieved within 10 seconds.

## Safety Requirements

* + 1. The user interface of the system should require minimal actions from the user. E.g. the map should have the location of the user centred at all times so that the user does not need to adjust the map manually*.*

## Security Requirements

* + 1. The system must protect user data such as names, contact details and location data through various means such as encryption and comply with data protection regulations such as the Personal Data Protection Act (PDPA). The application should only collect data that is necessary for the functionality of the service to ensure that data retention policies are adhered to.

* + 1. The system should be regularly updated to address security vulnerabilities and improve safety features.

* + 1. The system should obtain relevant safety certification such as the ISO/IEC 27001 for information security management as well as comply with industry standards for app development and data security, such as Open Web Application Security Project (OWASP) guidelines

## Software Quality Attributes

* + 1. The user interface should have an intuitive and user-friendly interface. 80% of first-time users should be able to use the application effectively on their first try.

* + 1. The system must be able to display information using four of Singapore's most common languages: English, Malay, Tamil and Chinese.

1. The default language of the system is English
2. Users must be able to change the system language on the settings page.

## Business Rules

* + 1. Users
       1. Normal users should be able to search and view car park information
    2. System Administrators
       1. System administrators configure system settings, manage API integration and oversee the general system performance.

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalisation requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**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 organisation, and just include terms specific to a single project in each SRS.>*

| **Term** | **Definition** |
| --- | --- |
| System | This refers to the web application. |
| Aggregated Parking Information | Collected information on car park prices, availability of parking lots, location of the car park that is available for users to read through. |
| Identifier | An easily seen marker/point for users. |
| UI | User Interface |
| GPS | Global Positioning System |
| URA | Urban Redevelopment Authority |
| HDB | Housing & Development Board |
| Location services | Services which provide GPS coordinates such as current coordinates and destination coordinates. |
| Query | Sending a data packet requesting information from various databases. |
| Filter | The process of selectively extracting information that meets predefined criteria |
| API | Application Programming Interface is the software middle man that allows two programs to talk to each other. |
| Algorithm | A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. |

**Appendix B: Analysis Models**

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

**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.>*

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc