# Project Mission Statement

Cirno Team will develop a parking app, ParkWhere, to allow motorists to look up carpark information. The project is complete when the app is tested and approved by the ZEA (CZ2006/CE2006 Authority). This project supports the Data-Driven Smart Nation Competition to exploit publicly available government data to help motorists find suitable carparks easily.

# Functional Requirements

1. Main Menu
   1. User must be able to input a search query into the main search bar.
   2. When the user taps the “Search” button, the system must parse the user’s search query and conduct a search based on that query.
      1. The system must be able to parse the user’s search query into a search type. Search types include:
         1. Full address
         2. Partial address (only road name or general location)
         3. Postal Code
         4. Carpark number
      2. The system must be able to conduct a search based on a search query.
         1. The search query will be listed on the Search Listing page.
2. Search Listing
   1. The system must display the resolved search query on the Search Listing page.
   2. The system must display, in pages, the nearest carparks to the searched location in the order of closest to furthest.
   3. The system must display at most 10 carpark listings per page in a table format.
      1. In each row of the table, the system must display each carpark listing’s Basic Information.
         1. The system must show the carpark listing as greyed out if the carpark’s available lots is zero.
      2. When the user taps a carpark listing, the system must show that carpark’s Advanced Information.
      3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Favorites list.
   4. When the user taps the “Next Page” button, the system shows the next 10 carpark listings in order.
   5. The system must be able to filter search results returned from a search query.
   6. When the user taps the “Filter” button, the system must show the filtering options. The filtering options are:
      1. Maximum distance to carpark
      2. Minimum number of available lots
      3. Parking System (Electronic/Coupon)
      4. Short Term Parking availability
      5. Free Parking availability
      6. Night Parking availability
   7. When any of the search filters are applied, the system must repopulate the search listing with carparks that meet the search filters’ criteria.
3. Favorites Listing
   1. The user must be able to add and remove carparks to a Favorites list. (see 2.3.3)
   2. The system must display, in pages, the Favorite carparks in alphabetical order of carpark number.
   3. The system must display at most 10 carpark listings per page in a table format.
      1. In each row of the table, the system must display each carpark listing’s Basic Information.
         1. The system must show the carpark listing as greyed out if the carpark’s available lots is zero.
      2. When the user taps a carpark listing, the system must show that carpark’s Advanced Information.
      3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Favorites list.
   4. When the user taps the “Next Page” button, the system shows the next 10 carpark listings in order.
4. Blacklist
   1. The user must be able to add and remove carparks to a Blacklist. (see 2.3.3)
   2. The system must display, in pages, the Blacklisted carparks in alphabetical order of carpark number.
   3. The system must display at most 10 carpark listings per page in a table format.
      1. In each row of the table, the system must display each carpark listing’s Basic Information.
         1. The system must show the carpark listing as greyed out if the carpark’s available lots is zero.
      2. When the user taps a carpark listing, the system must show that carpark’s Advanced Information.
      3. When the user taps a carpark listing, the user must be able to add or remove that carpark to their Blacklist.
   4. When the user taps the “Next Page” button, the system shows the next 10 carpark listings in order.

# Non-Functional Requirements

1. System must not crash.
2. System must not hang.
3. Search must return result within 5 seconds.
4. 90% of users must be able to get the search results within 2 minutes upon accessing the system.

# Data Dictionary

Carpark – A building or infrastructure that allows cars to be parked legally.

Carpark Number – The unique alpha-numeric identifier for a singular carpark.

Carpark Location – The details of where the carpark is located. This may include street name, block number, postal code.

Carpark Lots – Individual spaces for a single car to be parked.

Available Lots in Carpark – The current number of carpark lots that are not occupied by a parked car.

Total Lots in Carpark – The maximum number of carpark lots that a carpark accommodates.

Carpark Basic Information – A collection of basic information about a carpark, that is displayed in the Search Listing. This includes: Carpark Number, Carpark Distance from queried location, Carpark Location, Available Lots in Carpark, Total Lots in Carpark

Carpark Advanced Information – A collection of advanced information about a carpark, that is displayed when the user selects a carpark. This includes: Carpark Rates, Parking System (Electronic/Coupon), Short Term Parking Times, Free Parking Availability, Night Parking Availability, HDB/URA/LTA affiliation

# Use Case Diagram

Diagram, shape, rectangle

Description automatically generated

# Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Search For Carparks | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | System, User | | |
| Description: | Search for nearby carparks based on search query. | | |
| Preconditions: | 1. App must be connected to WiFi/Mobile Data | | |
| Postconditions: | 1. User is able to view a list of nearby carparks | | |
| Priority: | High | | |
| Frequency of Use: | 0-10 times per day | | |
| Flow of Events: | 1. User enters search query into search bar 2. System parses search query into a resolvable location 3. System finds at most 10 (or other preset value) nearest carparks to the queried location 4. System displays at most 10 (or other preset value) nearest carparks to the queried location, with each carpark’s Basic Information. | | |
| Alternative Flows: | AF-S1: User taps “Back” button while system is searching.   1. Return to step 1.   AF-S2a: User does not enter search query into search bar and commences search.   1. System assumes queried location to be current location.   AF-S2b: Search query cannot be resolved to a location.   1. System assumes queried location to be current location.   AF-S4: User taps “Next Page” button   1. If there are more carpark listings, system repopulates search listing with next 10 (or other preset value) carpark listings, continue from step 3.   AF-S4: User taps “Previous Page” button   1. System repopulates search listing with previous 10 (or other preset value) carpark listings, continue from step 3. | | |
| Exceptions: | EX-AF-S2a/b: Current location cannot be determined   1. System displays error message “Cannot find current location, please allow and turn on GPS”   EX-AF-S2: Internet connection gets cut off   1. System displays error message “No internet connection, please check if your mobile data or Wifi is turned on.” 2. Return to step 1. | | |
| Includes: | 1. Query Basic Carpark Information | | |
| Special Requirements: | - | | |
| Assumptions: | 1. Real-time Basic Information for found carparks is retrieved successfully. | | |
| Notes and Issues: | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Query Basic Carpark Information | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |
| Actor: | System, API | | |
| Description: | Query basic carpark information from API | | |
| Preconditions: | 1. App must be connected to WiFi/Mobile Data | | |
| Postconditions: | 1. Basic Information of queried carparks is retrieved | | |
| Priority: | High | | |
| Frequency of Use: | 0-10 times per day | | |
| Flow of Events: | 1. System requests Basic Information of specific carparks 2. API returns Basic Information of queried carparks to System | | |
| Alternative Flows: | AF-S1: Query is interrupted by user navigation   1. Stop query and drop existing results. | | |
| Exceptions: | EX-1: API does not return Basic Information of queried carparks   1. System displays error message “Server error, try again later”. 2. Return to search bar page. | | |
| Includes: | - | | |
| Special Requirements: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

# UI Mockups









