Carpark NF

Pineapple
Potala, Jin Min, Nicholas, Wei Hong,
Haozheng, Jia Ying

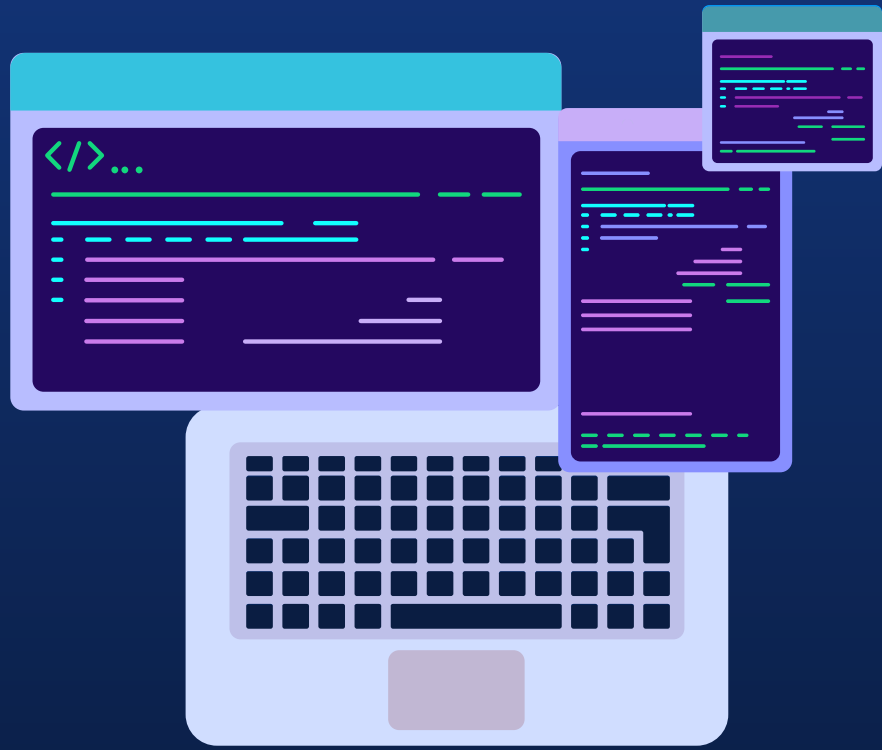


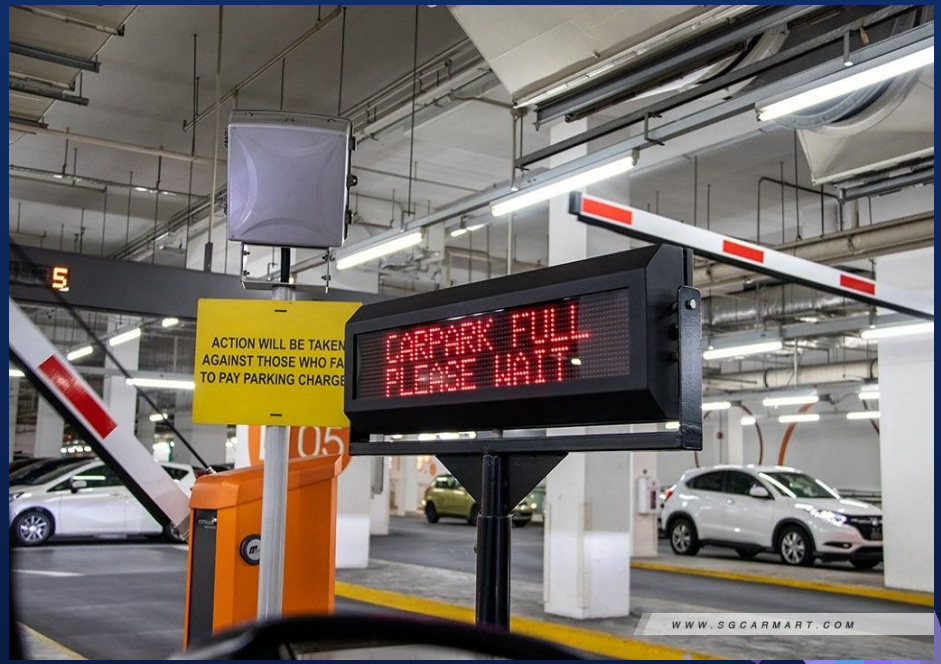
TABLE OF CONTENTS

01

Overview

Introduction of Carpark NF

See you at 12pm. If you are late then we are **OVER**!



Carpark NF

- Web application that allows users to search for a location and then showing them a list of carparks nearby (within 500m) that location
    - Shows the availability of the carpark so that users can plan their journey beforehand
- Allows users to favourite their frequently visited carparks, and navigate to them
- Target/Expected users of our application is drivers

02

Requirements

Functional requirements & non-functional requirements

Functional Requirements

1. General
    a. The system must be able to display a map with locations of carparks.

2. Search
    a. The system must allow user to search for a location.
    b. The user must be able to sort the list of carparks by distance or percentage of availability.

3. Favourite
    a. The user must be able to favourite carparks.
    b. The user must be able to view a list of their favourite carparks.
    c. The user must be able to remove a favourited carpark.

Functional Requirements

4. Select carpark to visit
    a. The user must be able to select a carpark to visit from his/her list of favourite carparks or list of carparks from searching for a destination.
    b. The user must be able to unselect a carpark that he/she has selected to visit.

5. Retrieve carpark availability information
    The system must be able to retrieve carpark availability information via Data.gov.sg's Carpark Availability API.

6. Retrieve carpark information
    a. The system must be able to retrieve a carpark's information using the carpark number by using the list of HDB Carpark Information by Data.gov.sg.

Functional Requirements

7. Locations
    a. The system must be able to retrieve locations via Google Maps API.

8. Navigation
    a. The user must be able to search for a route to a selected carpark.
    b. The system must be able to display the route searched for by the user on the map.

9. Carpark rates
    a. The system must be able to retrieve carpark rates for user's selected carpark from the URA API.

Non-Functional Requirements

1. Usability
    a. The system and user must have internet connection.
    b. The user must be allowed to favourite a carpark by clicking on a star icon

2. Reliability
    a. The system must update the carpark availability information every 1 minute.
    b. After a system reboot, the full system functionality must be restored within 5 minutes.

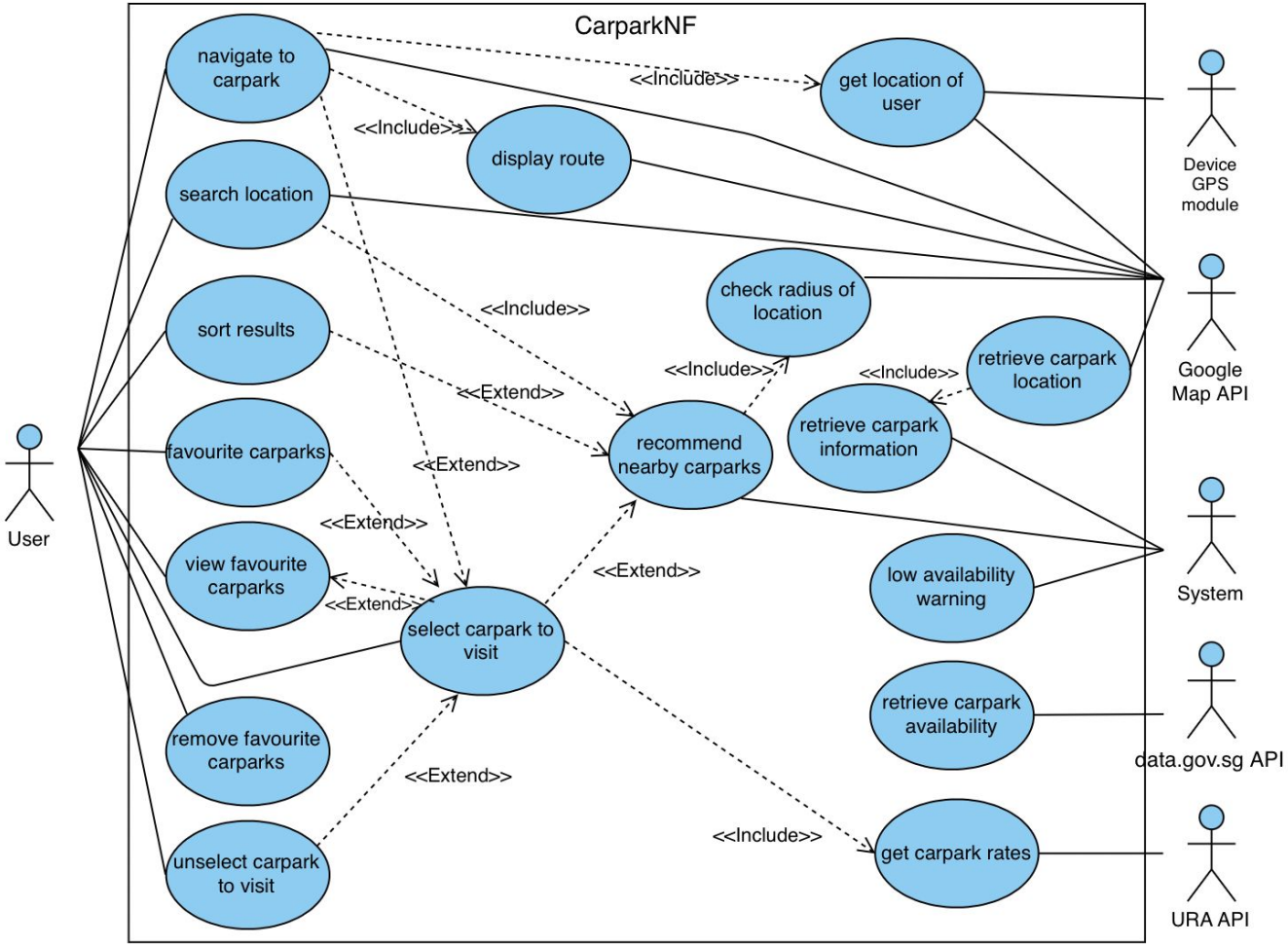
Non-Functional Requirements

3. Performance
    a. The system must not crash when the user opens the application.
    b. The user must be able to use the application within 20 seconds of opening the application.
    c. The system must be able to return the search results to the user within 10 seconds.

4. Supportability
    a. The user must be able to access the application from web browsers on their mobile device.

02

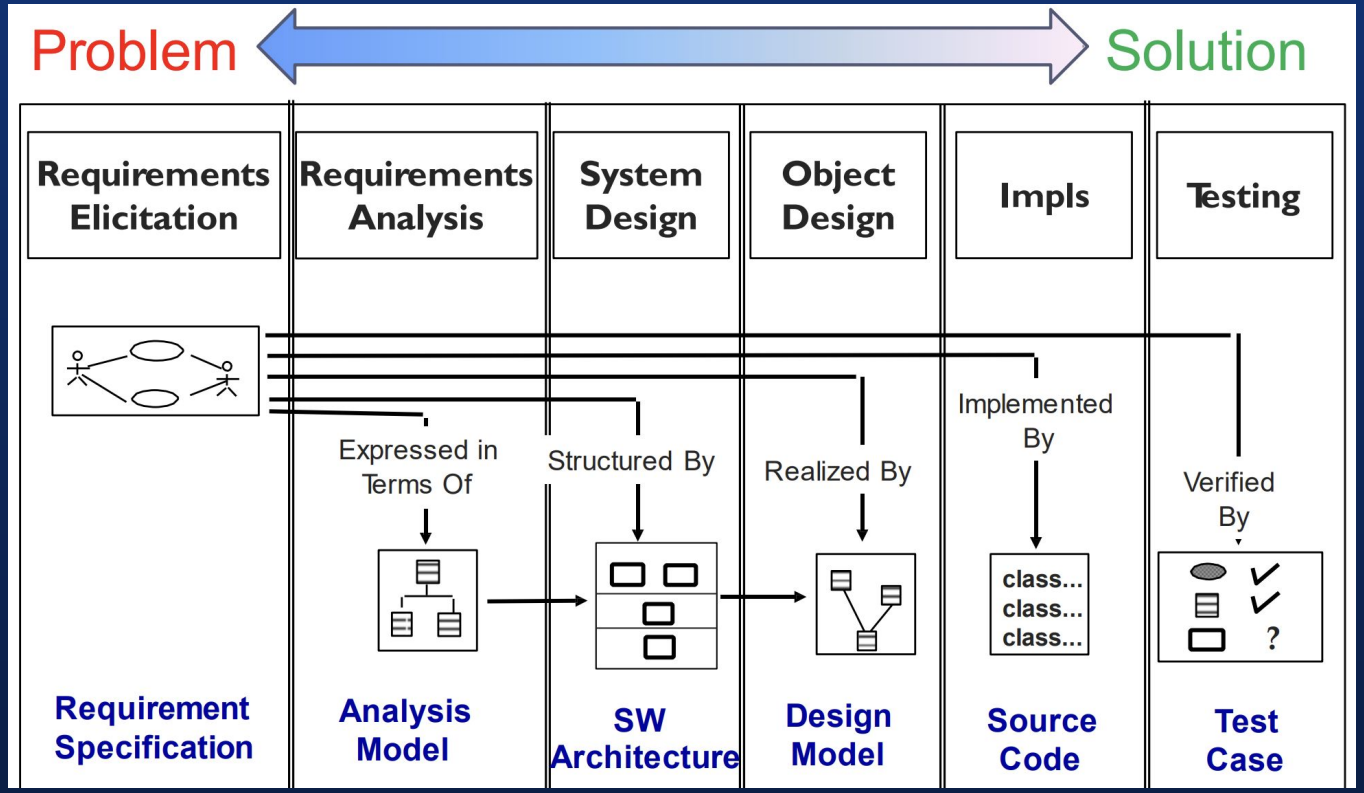
Use Case Model


CarparkNF
navigate to carpark
search location
sort results
favourite carparks
view favourite carparks
remove favourite carparks
unselect carpark to visit
display route
check radius of location
recommend nearby carparks
select carpark to visit
get location of user
retrieve carpark location
retrieve carpark information
low availability warning
retrieve carpark availability
get carpark rates
<<Include>>
<<Extend>>
User
Device GPS module
Google Map API
System
data.gov.sg API
URA API

03

SE Practices

Software Development Lifecycle



Design Practices

Single Responsibility

- Easier to maintain
- Reusability

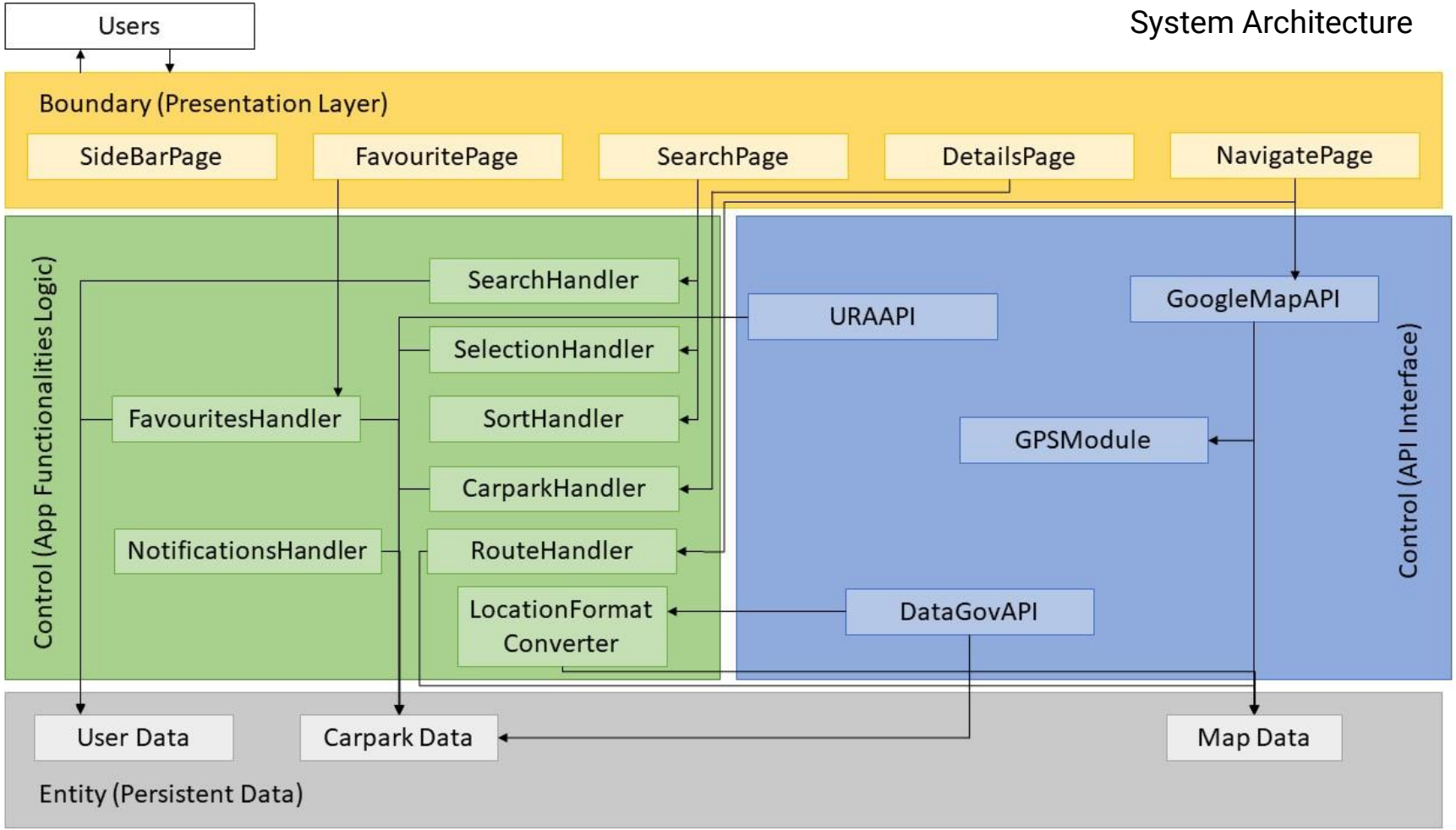
Dependency Injection

- Allows ease of testing, reusability of code, modularity
- Helps with loose coupling

04

System Design

System Architecture



Design Patterns

Facade Pattern

- Makes code easier to use and understand
- Reduces dependencies on classes

Observer Pattern

- Enables loose coupling
- Classes are only linked when necessary
- When one class needs to be changed, only a few will be affected - making codes reusable

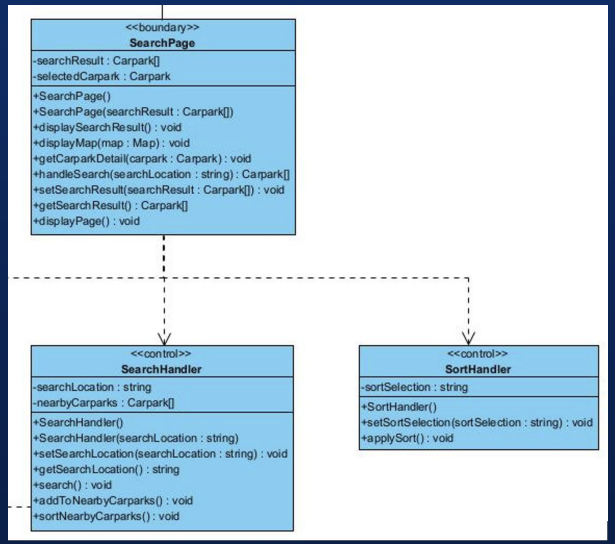
05

Traceability

Traceability

- Throughout the SDLC, we ensured our requirements are atomic such that it can be verifiable, traceable and unambiguous
- Example: Search functionality requirement

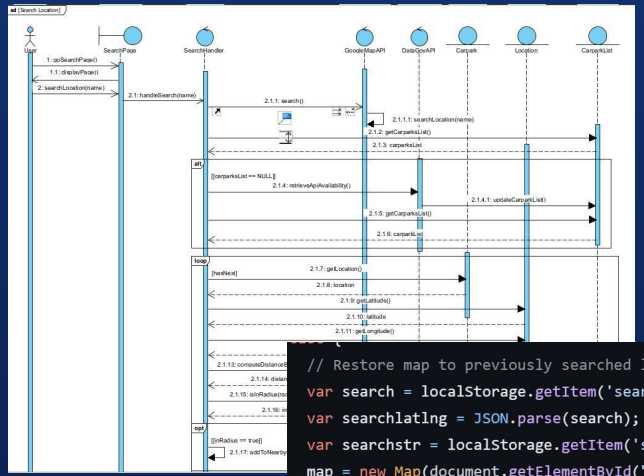
2. Search
    a. The system must allow user to search for a location.
        i. The user must be able to search for a location by entering a 6-digit postal code or a street name.
        ii. The system must be able to recommend a list of carparks within a radius of 500 metres of the searched location.
            1. Each carpark in the list of carparks displayed must contain information related to that carpark.
                a. Information includes:
                    i. Carpark distance from searched location
                    ii. Carpark address
                    iii. Carpark availability
                    iv. Carpark rate
        iii. The user must be able to sort the list of carparks by ascending order of distance from searched location or descending order of availability.
            1. By default, the list of carparks is sorted by ascending order of distance from searched location.



Traceability

Use Case ID:	1		
Use Case Name:	Search Location		
Created By:	Haozheng	Last Updated By:	Jin Min
Date Created:	30/08/2023	Date Last Updated:	13/09/2023

Actor:	User, Google Maps API
Description:	To search for carparks near the user's searched location
Preconditions:	1. Device must be connected to Wi-Fi/Mobile Data
Postconditions:	1. Carparks that fall within the radius of the searched location will be found and saved.
Priority:	High
Frequency of Use:	0-20 times per day
Flow of Events:	1. User navigate to the Search page from the sidebar 2. User will enter the 6-digit postal code or street name of his desired location 3. System will search for carparks that are near the location 4. Nearby carparks will be saved as a search result.
Alternative Flows:	-
Exceptions:	-
Includes:	Recommend nearby carpark
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

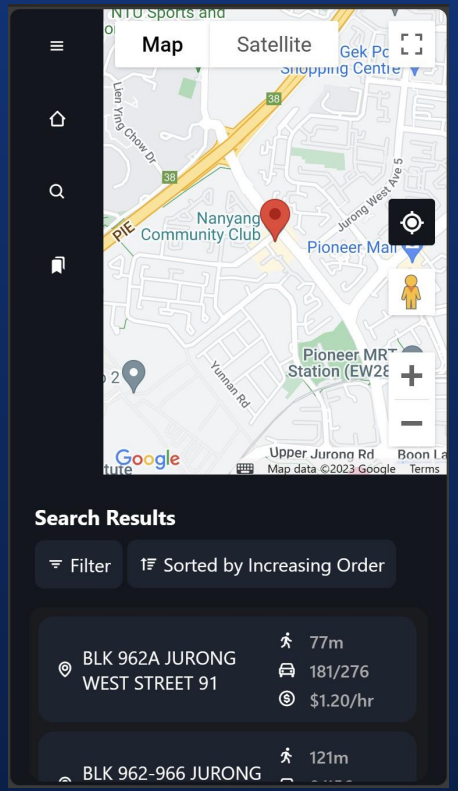



```
// Restore map to previously searched location by retr
var search = localStorage.getItem('searchlocation');
var searchlatlng = JSON.parse(search);
var searchstr = localStorage.getItem('searchstring');
map = new Map(document.getElementById("map"), {
  zoom: 15,
  center: searchlatlng,
  mapId: "mapid",
});
resetSearchResults();
hideScreen();
getLocation(searchstr);
ProcessData(searchstr);
```

Traceability

Test ID	Scenario	Expected Result	Actual Result
1	Search by 6-digit postal code (searched location)	List of carparks within 500 metres radius of searched location is displayed to the user	List of carparks within 500 metres radius of searched location is displayed to the user
2	Search by street name (searched location)	List of carparks within 500 metres radius of searched location is displayed to the user	List of carparks within 500 metres radius of searched location is displayed to the user
3	Search empty string	Popup stating "No results found for search."	Popup stating "No results found for search."

Search string	Expected Result	Actual Result
"530115" (Kovan)	List of carparks within 500 metres radius of an HDB block in Kovan is displayed to the user	List of carparks within 500 metres radius of an HDB block in Kovan is displayed to the user
"Nanyang CC"	List of carparks within 500 metres radius of Nanyang CC is displayed to the user	List of carparks within 500 metres radius of Nanyang CC is displayed to the user
"NTU"	"No results found" message displayed on results page. (No carparks nearby)	"No results found" message displayed on results page. (No carparks nearby)
"abcdefg"	Popup stating "No results found for search."	Popup stating "No results found for search."
"" (empty string)	Popup stating "No results found for search."	Popup stating "No results found for search."

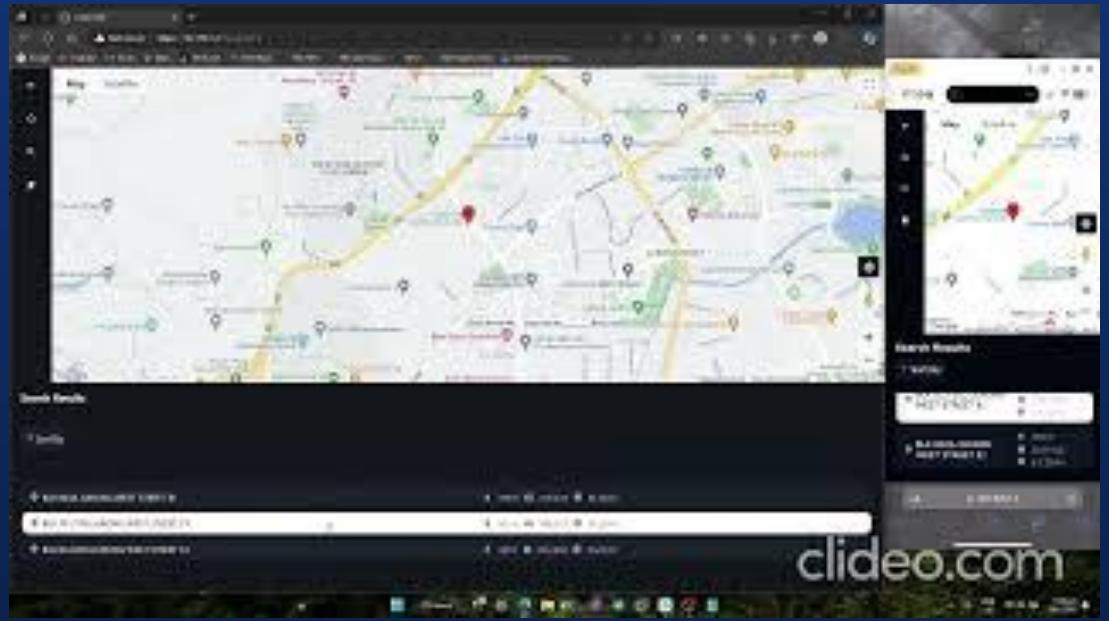


06

Demo

Flow

1. Enter app
2. Search
    a. Gibberish
    b. “Nanyang CC”
    c. “310256”
3. Sort
    a. By descending availability
    b. By ascending distance
4. Select carpark (from search results)
5. Navigate
6. Favourite carpark
7. Go to favourites page
8. Click on carpark
9. Unfavourite carpark
10. Back to favourites page



Try it for yourself!

Thank you