

Good morning everyone. My name is _____, and my team members are _____. We are going to present a demo of ParkWhere, a car park finder application

Let's start by taking a look at the use case diagram for our project.

Imagine you've just accessed the app, registered, and landed on the home screen. Now, let's say you're getting ready to leave the house to go to JEM, but you know the carpark there is often full. You'd like to find out where you could potentially park, along with estimated duration, estimated distance and price. Simply access our interface and find the nearest carpark.

Or imagine yourself outside, driving, and unable to find parking. The first thing you can do is use the option to search using your current location. With just a tap, you'll see all the available carparks within a 2km radius from where you are.

Alternatively, if you're headed to a different location, simply enter the address. After inputting the destination and clicking search, you'll be presented with all nearby carparks within a 2km radius. Click on the markers to access detailed information and directions to your desired parking spot.

Now, let's talk about saving time for frequent visits. Suppose you often go to the same carpark, like the one in Bukit Batok, from your home. Instead of searching every time, simply favorite the carpark. In the future, if you'd like to go there from your home, no need to search again. Just head over to "Favorite Carparks" and execute your trip from there. Convenience at your fingertips!

That covers the core functionality of our app. Now, without further ado, let's dive into a live demo. We'll showcase our application interfaces and guide you through how seamlessly you can navigate ParkWhere.

<Live demo>

Login

This is our homepage, where users will log in to the app. Users can log in either with email or Google. If user does not have an account with us, they can sign up here [Sign Up Page]

[Forget Password]

Email: test@gmail.com

Password: password1

-> wrong password

Here, you can see that I have forgotten the password. Let's reset the password [Reset Password]

[Go back to login] Once they log in, [Log In] they are brought to the main page [map]

Change Password

They can change their password if choose to do so, like this. [Change password to Password123!]

Search by Name

On our home page, we can search for car parks and see their information.

First, we type in the location that we want to go to, lets say lot one

Location: lot one

Then, we click find car park. Now, we can see the car parks around the area in a 2km radius

From here, we can see car parks that are available, which are the normal blue ones,[Click into a blue car park]

And the car parks which have limited availability, or unavailable, which are the red ones

[Click into red car park]

From here we can see all the car park information, such as availability, operating times and rates.

Search by Current Location

Another feature we have added is to search car parks by current location, for those that have already reached their destination and want to look for car parks nearby.

We can do so by clicking on this icon here [Click on icon]. From here, we can see the car parks in a 2 km radius from where we are currently

Favourite Car Park

Next, we can favourite our frequently used car parks by clicking on this. [Favourite a few car parks][show icon change]. We can see our favourite car parks here [Go to favourite carpark page]. From here, we can see all the car park information

Rename Carpark

We can also rename our favourite car parks [Rename Carpark]

New Name: HOME

If you wish to rename a carpark, simply follow these steps, navigate to the "Favourite Car Parks" section. Locate the carpark you want to rename. Click on the "Rename Carpark" option next to the carpark you wish to rename. Enter the new name for the car park in the designated field. Click "Ok" to confirm the new name. Your carpark will now be updated with its new name.

Delete Carpark

To remove a carpark from your favourites list, here's what you need to do: Go to the "Favourite Car Parks" page. Find the carpark you want to delete from your favourites. Click on the "Delete" option next to the carpark you wish to remove. Confirm the deletion when prompted. The car park will be removed from your favourites, and you will not see it in the "Favourite Car Parks" page anymore

Get Directions

Need directions to your carpark? Follow these simple steps: Select the carpark you want directions to either from your favourites list or from the marker rendered on google maps. Click on the "Get Directions" option. You'll be redirected to Google Maps, where the route to the selected carpark will be displayed. From there, you can view the route details, including the estimated travel time and distance. Feel free to customise the route or add multiple stops

if needed. Once you're ready, you can start navigation to the car park directly through Google Maps.

One software engineering technique we applied is modular design, which involves breaking down the code into smaller, reusable components or modules. This approach enhances flexibility, simplifies maintenance, and facilitates testing. For instance, we organised separate modules for database interactions, user interface components, and application logic. Each module had a specific responsibility, making it easier to detect and resolve issues when they occurred.

Additionally, we employed version control, utilising Git, to collaborate on the code and track changes over time. This practice allowed us to work on different features or aspects of the app independently, without concerns about conflicts or accidental overwrites. With a comprehensive commit history, our development process became more efficient, reducing the risk of introducing bugs or errors.

Moving on to design patterns, we adopted the Model-View-Controller (MVC) pattern. This pattern divides the code into three layers: the model (handling data), the view (managing the user interface), and the controller (handling user input and communicating between model and view). Implementing MVC promoted high cohesion and low coupling, enabling team members to work on the app simultaneously with minimal interference.

Furthermore, we utilised the strategy design pattern, which separates functions into different classes to support low coupling. This approach also aligns with the Open-Closed Principle, allowing for future extension while minimising modifications. Each class adheres to the Single Responsibility Principle, focusing on a specific function and enhancing the application's agility.