Flow:

Intro
App overview, problem and purpose
Main features overview (Use case models, Sequence Diagram, System Architecture)
Live Demo
Good SE practices and Design Principles

Intro

introduce group members

introduce problem statement and target users

how Parknow addresses the problem

Good Morning everyone. My name is Caleb. With me today are Alan, Ye Wint Myint Myat (Ye wint), Keith, and Joel, and together, we are Team Overflow

Next slide

Have you ever tried to find cheap parking in the Central Business District but found it hard and have to circle the area many times?

Next slide
Well, even though Singapore is known to be a relatively advanced and smart nation, the current experience of finding parking in the CBD remains a significant challenge. Outdated information and traffic issues persist, leading to frustration and confusion for drivers in our seemingly sophisticated urban environment.

Next slide

Introducing ParkNow, a user-centric solution dedicated to revolutionising the parking experience for drivers in Singapore, with a particular focus on the bustling Central Business District (CBD).

Next slide

For our app, we predict that the main target users will be drivers in Singapore, seeking a convenient and efficient solution for parking in high-demand areas.

Next slide

Now, I will be sharing more about the 5 main features of our app.

Next slide

First we have the login and register, each driver will have their own unique account to access the app.

Next slide

Next, we have the function to search for carparks near your destination. Drivers will be able to search for the destination they have in mind. After that, ParkNow will display cheap public parking near the location, all for the users to choose from!

Next slide

Next, we have the navigation feature of the app, which will provide the shortest, fastest route to your destination.

Next slide

We also have one of our key features, which is the display of the availability and rates of carparks for you to choose from.

Next slide

Finally, we have the favourite carparks option. This allows users to save carparks which they have found and allow to find them easily, at greater convenience.

Next slide

Now that we have identified the main features, we will proceed to draw our use case diagrams.

Next slide

This is an overview of ParkNow with all the use cases. Every user will have to register first to use our app. After registering for an account, users will then be able to log in using their usernames and password in the subsequent uses. After a successful login, the user will then be able to access all of our app features, such as the search and navigate functions.

Next slide

Here is our overall class diagram for ParkNow, which shows all of the functions for the respective classes and

Next slide

Here is the class diagram with each class in its entity, boundary, controller form. For this presentation, we will go more in depth for two of our features.

Next slide

The first feature will be the enquiry for carpark details, and we will explain the feature with the help of our sequence diagram. The user will first click on the carpark they want to travel to, and the SearchUI will redirect the user to the carpark UI. The carparkUI will then query for details from the SearchController which communicates with the carpark Availability API and the URA API. Upon receiving the response, the search controller will then return the results for the carparkUI to display the carpark details, such as the availability and rates of the carpark.

Next slide

And this is the class diagram for the enquiry of carpark details.

Next slide

The next feature we will be going through is the navigation feature. As shown in the sequence diagram, it will be a continuous loop while the user has not reached their destination. The mapsUI will query from the navigatecontroller for the details of the route and the controller will then send the request to the GoogleMapsAPI. The GoogleMapsAPI will then return the route details to the controller which will then pass the results for the MapsUI to display. The user will also have the option to terminate the navigation prematurely and this will redirect the user from the MapsUI to the CarparkUI.

Next slide

And this is the class diagram for the navigation feature.

Next slide

And to wrap up the explanation of our use cases, we have our system architecture. For the boundary classes, we coded them using React Native, for the controller classes we used django and finally for the entities, we stored the information in mySQL database.

Next slide

I will now pass the time to Ye Wint Myint Myat who will share with you more on the software engineering practices, and design principles that we adopted.

Slide 20

Thank you Caleb. I will now be going through the good software engineering practices and design principles that we have applied during the development of our app.

Slide 21

Firstly, we used GitHub as a platform to track contributions and also store our code changes in case something happens to our app. Secondly, we worked in pairs to code the front end and trio to code the back end of the project in order to develop common ownership of code, improving refactoring of code and sharing knowledge to debug and fix broken code and ensure traceability. Thirdly, we planned ahead to deliver a set amount of work by a certain time. We have a checklist of what we need to do by the end of every 2 weeks and each person would take on one or more tasks. At the start, we finished the functional and non-functional requirements and then finished our diagrams and finally finalising our app and testing it out.

Slide 22

Firstly for good design principles, we ensure the single responsibility principle which states that a class should only have 1 responsibility. For instance, "Register," "Login," and "View

Profile Page" are separate use cases, each with a distinct job. Secondly, we ensure loose coupling which ensures each class is as independent as possible from the others to make the system more maintainable and scalable. For instance, "Search for Nearest Car Parks" and "Add Car Park to Favorites" interact with the APIs but are independent of each other's internal details.

Slide 23

Thirdly, we ensure the modularity of controllers and functions so that they can be easily reused across different parts of the app. For example, searchController could be used by various parts of the application that need to implement search functionality.
Lastly, we ensure Encapsulation / Principle of Least Knowledge. For instance, SearchController may offer a method like searchNearbyCarParks, and other classes would call this method when they need to perform a search. However, they wouldn't know how SearchController interacts with the GoogleMapsAPI or CarparkAvailabilityAPI to get the search results

And now keith and joel will be demonstrating the live demo

Live Demo

Thank you Caleb. For our demo, we will be mirroring our android device so that everyone can see it. So the moment we open our app, we will first come across a splash screen followed by our introduction screens. These introduction screens inform new users the key features they can gain from using our app.

Firstly, they can easily find convenient parking spots right from their phones. Secondly, they can seamlessly navigate to their chosen carpark with directions provided. Lastly, they can remain informed on the prices of different car parks.

Once we're past the introduction screens, we enter our entry page. Here, new users can choose to sign up for an account. They will need to fill their email, username, password and confirmation password. Our app will check whether their inputs are satisfactory.

For instance, if they are using an email that is already used by someone else, it will return an error. Similarly, if their username is already taken, it will also not allow them to use it. Alternatively, their email also needs to be in the right format to be accepted. Once they have entered acceptable inputs, a confirmation OTP will be sent to the email they have provided. After keying in the correct OTP, they will then be redirected to the login page.

Here, the user is expected to key in their username or password. If for example, they key in a wrong username, they will be informed that the user does not exist. If they use the right username but wrong password, they will also be informed to try again.

Once they login with the correct credentials, they will be brought to our home page. What this home page does, is it takes their current location and displays all the carparks that are within a certain radius of their location. For the purpose of this demo, we have set this radius to be very large. But in actual practice, this radius will be much smaller or even something the user can set themselves.

Here, we can see each carpark's name, as well as the number of slots available and the distance of the carpark from us. These information are important as users will likely be interested in carparks that are not only near to them, but also have an abundant number of empty slots. If however, there are no parking lots, the app will display 'no slots available'

From this page, the user can choose carparks they are interested in to get more specific details. For instance, once I click in here, I can see this carpark's specific details, such as its address, availability and parking rates for different times of the week. Sometimes, users might want to share the location of the carpark with other people, such as to meet someone at a particular carpark. To do this, they can press the copy button, which copies the address to their clipboard. Also. If the user likes a carpark and wants to use it regularly, they can also press the favourites button to add it into their favourites folder. For instance, I will add the first few carparks here into my favourites folder. Later on, I will be able to view these carparks from my favourites folder. Now, I will pass the time to Joel.

Thank you Keith
We can search a location by clicking on the search bar at the top, then typing a new location. The app will try to complete what I type by giving suggestions. If we select marina bay, it will bring us to this list of car parks near to the location. The carpark list will display the distance of the carpark from the user's current location. When we go into one of the carparks, it will show us the same details as before. If we want to navigate to this carpark we can click on the "See Nearest Route" button. This will show us the fastest route given by Google API from the user's current location to the chosen carpark. Once we are done we can click on the cancel button to exit the navigation.

On the profile page, when we can see the user's username, email and password.
We can change the registered email by clicking on the button next to the email. This will lead us to this page, where we will key in our new email. The app will then prompt us for an OTP. Once we key in the correct OTP, the app will reflect our new email.
We can also change the password. If we enter the wrong existing password, the app will show an error message. If the new passwords do not match, another error message will show. Once all the inputs are correct. We will have to log back into the app.

If we click on the favourites page, we can see all the car parks that keith previously favourited. When we click on one of these car parks we can see the same details as well as the navigation function. So instead of having to search for frequently visited car parks each time we can add them to our favourites for easy access.