Use Cases

for

NAP

Version 1.0 approved

Prepared by

HENG CHOON KANG,
MUHAMMAD IQSHAN BIN MOHD SA'AD,
PAN ZAIYU, RYAN,
TAN WEI HONG

NANYANG TECHNOLOGICAL UNIVERSITY, Team Foresight

1/9/2023

Revision History

Name	Date	Reason For Changes	Version

Use Case ID:	UC1.1		
Use Case Name:	Search carparks (Map)		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do an automatic search around the user of 0.5km
Preconditions:	The user must launch the application
Postconditions:	The nearest carparks around the user will be displayed on the map
Priority:	High
Frequency of Use:	
Flow of Events:	1. User opens the application 2. The system will do a scan around the user to find the nearest carparks
Alternative Flows:	1. The user can choose to adjust the search radius
Exceptions:	NIL
Includes:	Carpark Availability
Special Requirements:	NIL
Assumptions:	1. The user must have enabled location service on their device 2. The user must be connected to the internet
Notes and Issues:	NIL

Use Case ID:	UC1.2		
Use Case Name:	Search carparks (List)		
Created By:	Ryan Pan	Last Updated By:	
Date Created:	2/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do an automatic search around the user of 0.5km
Preconditions:	The user must launch the application
Postconditions:	The nearest carparks around the user will be displayed in a list
Priority:	High
Frequency of Use:	
Flow of Events:	1. User opens the application 2. The system will do a scan around the user to find the nearest carparks 3. The system will display the results in a list
Alternative Flows:	1. The user can choose to adjust the search radius
Exceptions:	NIL
Includes:	Carpark Availability
Special Requirements:	NIL
Assumptions:	1. The user must have enabled location service on their device 2. The user must be connected to the internet
Notes and Issues:	NIL

Use Case ID:	UC2		
Use Case Name:	Search carparks around an area		
Created By:	Choon Kang	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User
Description:	The system will do a manual search around the specific area of 0.5km
Preconditions:	The user must launch the application.
Postconditions:	The nearest carpacks around the searched area will be displayed on the map or list.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. User opens the application 2. User search for a specific postal area 3. The system will scan around the area and find the nearest carpark within the set radius
Alternative Flows:	
Exceptions:	NIL
Includes:	Carpark Availability
Special Requirements:	NIL
Assumptions:	1. The user must have enabled location service on their device 2. The user must be connected to the internet
Notes and Issues:	NIL

Use Case ID:	UC1.3		
Use Case Name:	Carpark Availability		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	Data.gov API
Description:	The system will extract the carpark information from the API
Preconditions:	The system must have already conducted a radius search around the user or a specific point that the user has already decided
Postconditions:	The carparks around the user will be colour-coded according to availability
Priority:	High
Frequency of Use:	
Flow of Events:	1. The system will extract the relevant information from the API a. Available number of carpark lots 2. The system will classify the availability of the carpark lots through a color-coded system a. Green: 0-24% full b. Yellow: 25-49% full c. Red: 50-74% full d. Black: 75-100% full 3. The system will show the number of the available car park lots
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. Data provided by the government regarding parking lot availability is assumed to be correct and true 2. User must be connected to the internet
Notes and Issues:	NIL

Use Case ID:	UC4		
Use Case Name:	Navigating to carpark		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	Google Map API, User
Description:	Google Map will find the shortest route to the carpark that the user has decided
Preconditions:	User should have already made a selection regarding which car park they intend to visit
Postconditions:	User has successfully arrived at his destination
Priority:	High
Frequency of Use:	Once, each time the user wants to travel to their preselected car park
Flow of Events:	1. The user selects a carpark as their destination 2. The system utilizes Google Maps' API to determine the closest and most efficient route to the user's chosen car park
Alternative Flows:	AF-S1: If the user changes their mind about the car park destination enroute: 1. The system can recalculate the route based on its new selection
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User must be connected to the internet at all times 2. User has their location service enabled at all times for the Google Maps API to accurately determine their current location 3. User's device must be compatible with Google Maps API for it to function effectively 4. The system is properly integrated with Google Maps API which allows for seamless communication and route calculations
Notes and Issues:	NIL

Use Case ID:	UC5		
Use Case Name:	Save carpark		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	System will save the carpark the user parks in
Preconditions:	User must reach the destination and have his car parked
Postconditions:	User saves the location of the car park he parked his car at for future reference. (if he visits the place often)
Priority:	Low
Frequency of Use:	Once each time the user parks in a carpark
Flow of Events:	1. User decides if he is going to visit the place often 2. If yes, saves the location of the car park 3. Carpark can be found easily by the user once saved.
Alternative Flows:	AF-S1 User does not frequent to this carpark 1. User does not need to save carpark 2. User selects the cancel button
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	1. User must be connected to the internet at all times 2. User has their location service enabled at all times for the Google Maps API to accurately determine their current location 3. User's device must be compatible with Google Maps API for it to function effectively 4. The system is properly integrated with Google Maps API which allows for seamless communication and route calculations
Notes and Issues:	NIL

Use Case ID:	UC5.1		
Use Case Name:	View saved carpark		
Created By:	Ryan Pan	Last Updated By:	
Date Created:	2/9/2023	Date Last Updated:	

Actor:	User
Description:	The user will view a list of carparks that they previously saved
Preconditions:	The user must click on “Saved”
Postconditions:	The system will display a list of carparks that the user saved
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	1. User enters the main page 2. User clicks on “Saved” 3. System retrieves a list of saved carparks from the user’s account and displays them in a list
Alternative Flows:	AF-S1: If user has no saved carparks: 1. System displays “No saved carparks” 2. System returns to step 1
Exceptions:	NIL
Includes:	NIL
Special Requirements:	Database for User information
Assumptions:	1. The user must be connected to the internet
Notes and Issues:	NIL

Use Case ID:	UC5.2		
Use Case Name:	Save parking lot		
Created By:	1/9/2023	Last Updated By:	
Date Created:	Weihong	Date Last Updated:	

Actor:	User, System
Description:	User will enter the carpark lot that he parked his car at
Preconditions:	User has reached the car park After user has decided whether he wants to save the carpark for future use User has an account
Postconditions:	User saves the carpark lot
Priority:	Low
Frequency of Use:	Once
Flow of Events:	1. System will prompt the user to enter the carpark lot 2. User enters the carpark lot 3. System saves the information
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	UC3		
Use Case Name:	Filter carpark		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	System filter out carparks that does not meet the user requirements
Preconditions:	User must know what carpark he wants
Postconditions:	User finds car parks that meets his requirements
Priority:	Medium
Frequency of Use:	Each time the user enters the app
Flow of Events:	1. User selects requirements (distance, availability, parking rates) 2. System filters out car parks that does not meet those requirements 3. System displays remaining car parks on the map 4. User has a list of car parks that meets his requirements to choose from. 5. User chooses the desired carpark and System navigates
Alternative Flows:	1. User does not have a car park that he desires 2. User does not use the car park filtering function and just navigates to the nearest car park regardless of availability/rates.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	User must be connected to the internet at all times User has their location service enabled at all times for the Google Maps API to accurately determine their current location User's device must be compatible with Google Maps API for it to function effectively The system is properly integrated with Google Maps API which allows for seamless communication and route calculations
Notes and Issues:	NIL

Use Case ID:	UC6		
Use Case Name:	Pay Parking fees		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System, Parking.sg
Description:	User use the app to pay for parking fees for carpark without gantries
Preconditions:	User reached the car park
Postconditions:	User pays his fees
Priority:	Medium
Frequency of Use:	Each time User parks in a car park that has no gantry
Flow of Events:	1. User parks his car 2. User pays his parking fees using a link sent to him through System app
Alternative Flows:	AF-S1: User parks his car at carparks with electronic gantries: 1.
Exceptions:	NIL
Includes:	Integration with Parking.sg
Special Requirements:	NIL
Assumptions:	1. User visits a coupon parking carpark instead of the ones with electronic gantries 2. User must be connected to the internet at all times 3. User must have already created an account with Parking.sg 4. User must have already save their credit card information with Parking.sg 5. System is properly integrated with Parking.sg to allow seamless transition from one app to another
Notes and Issues:	NIL

Use Case ID:	REG 1		
Use Case Name:	Registration		
Created By:	Muhd Iqshan	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	First time users can register for an account. After registration, the user can save carparks that they frequent to
Preconditions:	The user should not have registered an account previously
Postconditions:	The user has successfully created the account and logged in
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. User clicks on 'Sign Up' in the 'Log In' page 2. System prompts the user to enter username, password and email address 3. User inputs the required information and clicks 'Register' 4. System checks whether the information submitted is valid and sufficient 5. System stores these information into a database 6. User is logged in
Alternative Flows:	AF-S3 If username is taken by another user: 1. System displays "Username already exists. Please enter another username" 2. System returns to Step 2 AF-S3 If password do not meets the requirements: 1. System displays "Passwords do not meet the necessary requirements" 2. System returns to Step 2 AF-S3 If email address is invalid 1. System displays "Please enter a valid email address" 2. System returns to Step 2
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	User should be connected to the internet when registering
Notes and Issues:	NIL

Use Case ID:	L1		
Use Case Name:	Account login		
Created By:	Weihong	Last Updated By:	
Date Created:	1/9/2023	Date Last Updated:	

Actor:	User, System
Description:	User use the app to pay for parking fees for carpark without gantries
Preconditions:	User has an account
Postconditions:	User can use the app
Priority:	High
Frequency of Use:	Once for every user
Flow of Events:	1. User keys in username and password to log in 2. User is free to use the app
Alternative Flows:	AF-S1 If user does not have a registered account: 1. User can click on the 'Register' button to register an account AF-S1 If user forgots their username/password: 1. User can click on the 'Forgot Password' button to initiate a password reset process and receive a password reset link or instructions sent to their registered email address. AF-S1 If user selects 'Remember my login info' before logging in: 1. The system will securely store their username and password, ensuring smoother subsequent logins AF-S1 If username/password is incorrect: 1. System displays "Username/password is incorrect. Please try again" 2. Go back to Step 1
Exceptions:	NIL
Includes:	Database for User information
Special Requirements:	NIL
Assumptions:	1. User must be connected to the internet at all times
Notes and Issues:	NIL