Use Cases

for

SC2006 Software Engineering

Version 1.0 approved

Prepared by Glenn Lee, Glynis Looi, Jan Chen Jie, Lee Yun Jia, Manasi Singh, Mayukhi Nimesha

SCSD C5, Nanyang Technological University

28/01/2025

Revision History

Name	Date	Reason For Changes	Version

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use

case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by "EX" to indicate "Exception". Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case Template

Use Case ID:	UC-001		
Use Case Name:	Login		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	31 Jan 2025	Date Last Updated:	1 Feb 2025

Actor:	User
Description:	User logs into the system to access personalized features.
Preconditions:	1. User has an existing account. 2. User has a valid email and the corresponding password registered in the system.
Postconditions:	1. Success: User is authenticated and gains access to the system's features. 2. Failure: The email and/or the password is invalid
Priority:	High
Frequency of Use:	Very frequent (multiple times a day)
Flow of Events:	1. User opens the application. 2. The system displays a login interface requesting a username and password. 3. User enters their credentials. 4. The system validates the credentials by comparing them with the database. 5. If the credentials are correct, the system grants access to user’s dashboard.
Alternative Flows:	AF-S5: If the credentials are incorrect, 1. The interface displays the message “Invalid email and/or password. Please try again!” for 2 seconds. 2. The display returns to the step 2. 3. If credentials are still incorrect after 3 tries, the option “Forget Password” is displayed. 4. The system prompts the user to enter their registered email. 5. A password reset link or OTP is sent to the registered email. 6. User sets a new password and proceeds to log in.
Exceptions:	Account locked: If multiple login attempts fail, the account may be temporarily locked
Includes:	Change Username and Password
Special Requirements:	- The system must ensure secure communication for login (e.g., HTTPS and encrypted passwords). - The system must lock accounts after a predefined number of failed attempts to ensure security.
Assumptions:	1. User has a stable internet connection.

Notes and Issues:	1. Consider adding multi-factor authentication for enhanced security

Use Case ID:	UC-002		
Use Case Name:	Search Destination		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	31 Jan 2025	Date Last Updated:	1 Feb 2025

Actor:	User, and Guest User
Description:	Allows users to search for parking destinations and apply various filters to find suitable parking locations. Users can view recommended car parks, detailed parking information, and access specific features like pricing, EV parking, sheltered parking, and distance information.
Preconditions:	1. User is logged in to account or accessing as guest user 2. Has location services (GPS) enabled 3. System is operational and accessible
Postconditions:	1. User can view filtered list of parking lots matching their criteria 2. User can access detailed information about specific car parks 3. User can view recommended parking options.
Priority:	High
Frequency of Use:	Very frequent (multiple times a day)
Flow of Events:	1. User selects "Search Destination" option from the dashboard 2. System displays filter options interface 3. User selects desired filters: 3.1. Pricing preferences 3.2. EV parking requirements 3.3. Sheltered parking needs 3.4. Distance preferences 4. System processes filter criteria 5. System displays filtered results including: 5.1. Recommended car parks 5.2. Detailed car park information 6. User can select specific car park to view detailed information
Alternative Flows:	1. If no results match filter criteria: 1.1. System displays message indicating no matches 1.2. Suggests modifying filter settings 2. If location services are disabled: 2.1. System prompts user to enable location services 2.2. Offers manual location input option
Exceptions:	1. System cannot access location services 1.1. Display error message 1.2. Prompt user to check device settings 2. Connection failure 2.1. Display error message 2.2. Offer retry option
Includes:	1. Filter Features 2. Get Pricing Information

	3. Get EV Parking Information 4. Get Sheltered Parking Information 5. Get Distance Information 6. Show Recommended Car Park 7. View Car Park Details
Special Requirements:	1. Real-time parking availability data 2. Accurate GPS functionality 3. Up-to-date pricing information 4. Current EV charging station status
Assumptions:	1. User has stable internet connection 2. Device has GPS capability 3. Parking data is regularly updated 4. User understands basic filtering concepts
Notes and Issues:	1. Filter combinations should be optimized for user experience 2. Consider adding save favorite filters feature 3. Real-time updates for parking availability 4. Consider adding sorting options for search results

Use Case ID:	UC-003

Use Case Name:	Filter Features		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Enables users to apply specific filtering criteria to narrow down parking options based on various parameters including pricing, EV parking availability, sheltered parking options, and distance preferences.
Preconditions:	1. User has initiated a destination search 2. Search destination functionality is active 3. Filter options are available and loaded
Postconditions:	1. Filtered results are displayed to user 2. All selected filter criteria are applied 3. Results show only relevant parking options
Priority:	High
Frequency of Use:	High (Multiple times per search)
Flow of Events:	1. System displays available filter options to user 2. User selects one or more filter criteria: 2.1. Price range 2.2. EV parking requirements 2.3. Sheltered parking needs 2.4. Distance preferences 3. System processes selected filters 4. System applies filters to available parking options 5. System displays filtered results 6. User can modify or clear filters as needed
Alternative Flows:	1. User clears all filters: 1.1. System resets to default view 1.2. Shows all available options 2. User modifies existing filters: 2.1. System updates results in real-time 2.2. Displays newly filtered options
Exceptions:	1. No results match filter combination: 1.1. Display "No matches found" message 1.2. Suggest broadening filter criteria 2. Filter system error: 2.1. Display error message 2.2. Offer reset option
Includes:	1. Get Pricing Information 2. Get EV Parking Information 3. Get Sheltered Parking Information 4. Get Distance Information
Special Requirements:	1. Responsive filter interface 2. Real-time filter application 3. Multiple filter selection capability

	4. Filter combination logic
Assumptions:	1. All filter options are functioning correctly 2. Data for each filter category is available 3. Filters can be applied in any combination 4. System can handle multiple simultaneous filters
Notes and Issues:	1. Consider adding filter presets for common combinations 2. Implement filter history for quick reuse 3. Consider adding "popular filter combinations" based on usage 4. Ensure filter performance doesn't impact overall system speed 5. Consider adding tooltips for filter explanations

Use Case ID:	UC-004		
Use Case Name:	Get Pricing Information		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Retrieves and displays detailed pricing information for selected parking locations, including hourly rates, daily rates, special rates (if any), and payment methods accepted.
Preconditions:	1. Filter Features functionality is active 2. Pricing data is available in the system 3. User has accessed the filter options
Postconditions:	1. Pricing information is displayed to user 2. User can view different rate categories 3. Payment method information is accessible
Priority:	High
Frequency of Use:	Very frequent (Multiple times each search)
Flow of Events:	1. System retrieves pricing data for available parking locations 2. User selects pricing filter options: 2.1. Set maximum price willing to pay 2.2. Select preferred payment methods 2.3. Choose rate type (hourly/daily) 3. System displays: 3.1. Standard parking rates 3.2. Peak hour rates (if applicable) 3.3. Special rates (weekend/holiday) 3.4. Available payment methods 4. User can sort results by price 5. System updates results based on price criteria
Alternative Flows:	1. User modifies price range: 1.1. System updates results immediately 1.2. Displays new price-filtered options 2. Special rate periods: 2.1. System displays alternative pricing 2.2. Shows applicable time periods
Exceptions:	1. Pricing information unavailable: 1.1. Display "Price information temporarily unavailable" 1.2. Show last updated timestamp 2. Invalid price range selected: 2.1. Display error message 2.2. Suggest valid price range
Includes:	None
Special Requirements:	1. Real-time price updates 2. Currency conversion capability 3. Clear price breakdown display

	4. Support for multiple payment methods
Assumptions:	1. Pricing data is current and accurate 2. System has access to real-time pricing updates 3. Payment method information is current 4. Users understand local currency
Notes and Issues:	1. Consider implementing price alerts 2. Add historical price trends 3. Include peak vs. off-peak pricing comparison 4. Consider adding price estimation calculator 5. Include information about parking passes or subscriptions 6. Ensure pricing information is clearly formatted for user understanding

Use Case ID:	UC-005		
Use Case Name:	Get EV Parking Information		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Provides users with detailed information about Electric Vehicle (EV) parking facilities, including charging station availability, types of chargers, charging rates, and real-time availability status.
Preconditions:	1. Filter Features functionality is active 2. EV parking data is available in the system 3. User has accessed the filter options
Postconditions:	1. EV parking information is displayed to user 2. Charging station availability status is shown 3. User can view detailed charging facility information
Priority:	High
Frequency of Use:	Frequent (Daily by EV owners)
Flow of Events:	1. System retrieves EV parking data 2. User selects EV parking filter options: 2.1. Charger type preferences (Type 1, Type 2, CCS, CHAdeMO) 2.2. Charging speed requirements (slow/fast charging) 2.3. Power output preferences (kW) 3. System displays: 3.1. Available EV parking spots 3.2. Charging station types 3.3. Real-time availability 3.4. Charging rates 4. User can view detailed information for each location 5. System updates availability in real-time
Alternative Flows:	1. No EV charging stations available: 1.1. System shows nearest alternative locations 1.2. Displays estimated waiting times 2. Charging station maintenance: 2.1. Display maintenance schedule 2.2. Show alternative charging options
Exceptions:	1. Real-time data unavailable: 1.1. Display last known status 1.2. Show timestamp of last update 2. Charging station malfunction: 2.1. Display alert message 2.2. Provide customer service contact
Includes:	None
Special Requirements:	1. Real-time charging station status updates 2. Integration with EV charging networks 3. Power output information display 4. Charging time estimations
Assumptions:	1. EV charging station data is current

	2. System can track charging station status 3. Users understand EV charging terminology 4. Network connectivity for real-time updates
Notes and Issues:	1. Consider adding charging session booking feature 2. Implement notification system for available chargers 3. Add user reviews for charging stations 4. Include charging speed information 5. Consider adding payment integration for charging 6. Track and display charging station reliability metrics

Use Case ID:	UC-006		
Use Case Name:	Get Sheltered Parking Information		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Provides users with comprehensive information about sheltered parking facilities, including covered parking spots, basement parking, multi-story car parks, and weather protection features.
Preconditions:	1. Filter Features functionality is active 2. Sheltered parking data is available in the system 3. User has accessed the filter options
Postconditions:	1. Sheltered parking options are displayed to user 2. Type of shelter/coverage is clearly indicated 3. Availability status of sheltered spots is shown
Priority:	High
Frequency of Use:	Frequent (Daily, especially during adverse weather - very applicable to Singapore which has prolonged monsoon seasons)
Flow of Events:	1. System retrieves sheltered parking data 2. User selects sheltered parking filter options: 2.1. Type of shelter (covered/basement/multi-story) 2.2. Level preferences (if applicable) 2.3. Direct building access requirements 3. System displays: 3.1. Available sheltered parking locations 3.2. Type of shelter provided 3.3. Current availability 3.4. Access routes to connected buildings 4. User can view detailed shelter information 5. System updates availability in real-time
Alternative Flows:	1. No sheltered parking available: 1.1. System shows nearest alternatives 1.2. Suggests peak vs. off-peak timing 2. Partial shelter availability: 2.1. Display partially covered options 2.2. Show distance to nearest full shelter
Exceptions:	1. Shelter status unavailable: 1.1. Display last known status 1.2. Show timestamp of last update 2. Maintenance or closure: 2.1. Display alert message 2.2. Show alternative sheltered options
Includes:	None
Special Requirements:	1. Real-time availability updates 2. Clear shelter type categorization 3. Building connection information 4. Weather protection details
Assumptions:	1. Shelter status data is current

	2. System can differentiate shelter types 3. Building access information is updated 4. Weather protection features are maintained
Notes and Issues:	1. Consider adding weather alerts integration 2. Implement peak-hour availability predictions 3. Add information about lighting conditions 4. Include security feature information 5. Consider adding height restrictions 6. Track and display maintenance schedules

Use Case ID:	UC-007		
Use Case Name:	Get Distance Information		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest, Location Services
Description:	Provides users with accurate distance and travel time information between their location and parking facilities, including walking distance to final destination and different route options.
Preconditions:	1. Filter Features functionality is active 2. Location services (GPS) are enabled 3. User has accessed the filter options 4. Destination location is specified
Postconditions:	1. Distance information is displayed to user 2. Walking routes and times are shown 3. Alternative routes are available if applicable
Priority:	High
Frequency of Use:	Very frequent (Multiple times per search)
Flow of Events:	1. System retrieves current location data 2. User selects distance filter options: 2.1. Maximum walking distance preferred 2.2. Route type preferences 2.3. Time constraints 3. System displays: 3.1. Distance to parking location 3.2. Estimated driving time 3.3. Walking distance to final destination 3.4. Alternative route options 4. User can view detailed route information 5. System updates travel times based on traffic
Alternative Flows:	1. Location services disabled: 1.1. Prompt to enable location services 1.2. Allow manual location entry 2. Multiple route options: 2.1. Display alternative routes 2.2. Show time differences
Exceptions:	1. Unable to calculate route: 1.1. Display error message 1.2. Suggest alternative locations 2. Traffic disruptions: 2.1. Show traffic alerts 2.2. Provide alternate routes
Includes:	None
Special Requirements:	1. Real-time GPS integration 2. Traffic data integration 3. Accurate walking route calculation 4. Multiple route calculation capability
Assumptions:	1. GPS services are available 2. Road and pathway data is current

	3. Traffic information is available 4. Walking routes are accessible
Notes and Issues:	1. Consider adding public transport integration 2. Implement favorite routes saving 3. Add accessibility route options 4. Include covered walkway information 5. Consider adding time-based recommendations 6. Track historical traffic patterns

Use Case ID:	UC-008		
Use Case Name:	Show Recommended Car Park		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Displays personalized car park recommendations based on user preferences, historical data, and current conditions including availability, distance, pricing, and facility features.
Preconditions:	1. Search Destination functionality is active 2. User location is available 3. Car park data is accessible 4. Filter preferences are set (if any)
Postconditions:	1. Recommended car parks are displayed 2. Recommendations are ranked by relevance 3. Key information for each recommendation is visible
Priority:	High
Frequency of Use:	Frequent (Multiple times per search)
Flow of Events:	1. System analyzes available car parks based on: 1.1. Current location 1.2. User preferences (if available) 1.3. Real-time availability 1.4. Historical usage patterns 2. System displays recommended car parks showing: 2.1. Distance and estimated travel time 2.2. Current availability 2.3. Pricing information 2.4. Special features (EV, Sheltered, etc.) 3. User can sort recommendations by different criteria 4. System updates recommendations in real-time
Alternative Flows:	1. No suitable recommendations available: 1.1. Display closest alternatives 1.2. Suggest modified search criteria 2. Limited availability: 2.1. Show next best alternatives 2.2. Display wait time estimates
Exceptions:	1. Unable to generate recommendations: 1.1. Display error message 1.2. Show general car park list instead 2. System overload: 2.1. Show cached recommendations 2.2. Display last update timestamp
Includes:	None
Special Requirements:	1. Machine learning integration for personalization 2. Real-time availability updates 3. Historical data analysis capability 4. Quick response time for recommendations
Assumptions:	1. Sufficient data available for recommendations 2. User preferences can be determined

	3. Real-time updates are possible 4. Car park information is accurate
Notes and Issues:	1. Consider implementing user feedback system 2. Add recommendation explanation feature 3. Include peak time warnings 4. Consider weather impact on recommendations 5. Implement regular recommendation refresh 6. Track recommendation accuracy

Use Case ID:	UC-009		
Use Case Name:	View Car Park Details		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest
Description:	Provides comprehensive details about a specific car park including real-time availability, facilities, entrance/exit points, operating hours, security features, and payment options.
Preconditions:	1. Search Destination functionality is active 2. Car park data is available 3. User has selected a specific car park 4. System has access to real-time data
Postconditions:	1. Detailed car park information is displayed 2. All available features are listed 3. Real-time status is shown 4. User can access navigation options
Priority:	High
Frequency of Use:	Frequent (a few times every search)
Flow of Events:	1. User selects a specific car park 2. System retrieves comprehensive information: 2.1. Current availability 2.2. Operating hours 2.3. Entrance/exit locations 2.4. Height restrictions 2.5. Payment methods accepted 2.6. Security features 3. System displays detailed information: 3.1. Floor plans (if available) 3.2. Photos of entrances 3.3. Special instructions 4. User can: 4.1. View real-time updates 4.2. Access navigation directions 4.3. Save car park to favorites 4.4. Share location details
Alternative Flows:	1. Information partially unavailable: 1.1. Display available information 1.2. Show when full details will be updated 2. Temporary facility changes: 2.1. Display alert messages 2.2. Show duration of changes
Exceptions:	1. Unable to retrieve details: 1.1. Display error message 1.2. Show basic information only 2. System maintenance: 2.1. Show cached information 2.2. Display last update time

Includes:	None
Special Requirements:	1. Real-time data integration 2. Image display capability 3. Interactive floor plans 4. Multi-language support
Assumptions:	1. Car park data is regularly updated 2. Images and floor plans are available 3. Real-time status can be monitored 4. Payment system information is current
Notes and Issues:	1. Consider adding virtual tour feature 2. Implement user reviews section 3. Add historical availability patterns 4. Include accessibility information 5. Consider adding booking integration 6. Track frequent user patterns

Use Case ID:	UC-010		
Use Case Name:	Check Username and Password		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User
Description:	Validates the username and password entered by the user during login to ensure they match the registered credentials.
Preconditions:	1. The user has registered an account in the system. 2. The user is on the login screen.
Postconditions:	**Success:** The user is authenticated and gains access to their account. **Failure:** The system informs the user of invalid credentials and prompts them to re-enter or reset their password.
Priority:	High
Frequency of Use:	Very frequently (a few times a day)
Flow of Events:	1. The user enters their username and password on the login screen. 2. The system retrieves the stored credentials for the entered username. 3. The system compares the entered password with the stored password. 4. If the credentials match, the user is logged in.
Alternative Flows:	**Incorrect Credentials:** 1. System informs the user that the credentials are incorrect. 2. The user is prompted to re-enter their credentials or use the "Forget Password" option.
Exceptions:	1. Unregistered Username: System informs the user that the username is not found in the database. 2. System Error: System fails to validate credentials due to backend issues.
Includes:	None
Special Requirements:	1. Ensure secure storage and retrieval of credentials. 2. Enforce strong encryption for passwords. 3. Include a "Show Password" toggle to help users avoid typos while typing their password. 4. Provide an option to remember the username for faster logins
Assumptions:	1. The system has access to the account database. 2. The system's user database is up to date
Notes and Issues:	1.

Use Case ID:	UC-011

Use Case Name:	System Register Account		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User
Description:	Allows a user to create a new account by providing necessary details such as username, password, and other relevant information. This enables the user to access personalized features of the system.
Preconditions:	1. The user is on the registration page. 2. The system is connected to the database for storing account details.
Postconditions:	**Success:** The user's account is successfully created, and they are redirected to the login page or logged in automatically. **Failure:** The system informs the user of any errors (e.g., invalid input, username already taken) and provides guidance to resolve the issue.
Priority:	High
Frequency of Use:	Infrequent (only used when a new user registers)
Flow of Events:	1. The user navigates to the "Register" page. 2. The user enters required information, including: a. Username b. Password c. Additional information (e.g., email, phone number, etc.) 3. The system validates the input: a. checks if the username is unique b. checks if the password fits the security requirements c. ensure all mandatory fields are filled 4. If all inputs are valid: a. The system stores the account details in the database. b. A confirmation message is displayed.
Alternative Flows:	AF-S4: If the inputs are invalid, it could be due to one of these 2 **Username Already Taken:** 1. The system informs the user and suggests alternative usernames. 2. The user enters a new username. **Weak Password:** 1. The system provides feedback on password strength. 2. The user updates the password to meet requirements.

Exceptions:	**Database Error:** The system encounters an issue storing the account details in the database.
Includes:	None
Special Requirements:	1. Provide optional or mandatory email or phone verification to confirm account ownership. 2. Offer the registration page in multiple languages for better accessibility.
Assumptions:	1. The user has access to a valid email address or phone number for account creation. 2. The system database is functional and can store new account details. 3. The user understands the registration process and inputs data accurately.
Notes and Issues:	1. Simplify the registration form by limiting the number of mandatory fields.

Use Case ID:	UC-012		
Use Case Name:	Forget Password		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User
Description:	Enables users to securely reset their forgotten password through a verification process, allowing them to regain access to their account.
Preconditions:	1. User has an existing account in the system 2. User has access to their registered email/phone 3. Login page is accessible
Postconditions:	1. User's password is successfully reset 2. User can log in with new password 3. System sends confirmation of password change
Priority:	Low
Frequency of Use:	Occasional (Rarely needed by most users)
Flow of Events:	1. User clicks "Forget Password" option on login page 2. System prompts for user identification: 2.1. Registered phone number 2.2. Registered email address 3. System verifies user exists 4. System generates and sends verification code: 4.1. Via SMS to registered phone 4.2. Via email to registered email 5. User enters verification code 6. System validates code 7. System prompts for new password: 7.1. New password entry 7.2. Password confirmation 8. System validates password requirements 9. System updates password 10. System confirms successful password reset
Alternative Flows:	1. Invalid user identification: 1.1. Display error message 1.2. Allow retry 2. Incorrect verification code: 2.1. Allow code resend 2.2. Provide retry option
Exceptions:	1. Account locked: 1.1. Display lock duration 1.2. Provide customer support contact 2. System unable to send verification: 2.1. Offer alternative verification methods 2.2. Show error message
Includes:	None
Special Requirements:	1. Secure password reset process 2. Strong password requirements 3. Time-limited verification codes 4. Multiple verification methods

Assumptions:	1. User has access to registered contact methods 2. Database system is operational 3. Communication services (email/SMS) are available 4. Password encryption is in place
Notes and Issues:	1. Consider implementing security questions 2. Add account recovery options 3. Track failed reset attempts 4. Implement password strength meter 5. Consider adding 2FA option during reset 6. Log all password reset activities

Use Case ID:	UC-013		
Use Case Name:	Get Navigation		
Created By:	Manasi	Last Updated By:	Manasi
Date Created:	1 Feb 2025	Date Last Updated:	1 Feb 2025

Actor:	User, Guest, Location Service
Description:	Provides real-time navigation guidance to selected parking locations, including route optimization, turn-by-turn directions, alternative routes, and real-time traffic updates. Integrates with external location services for accurate navigation.
Preconditions:	1. User has selected a destination car park 2. GPS/Location services are enabled 3. Device has internet connectivity 4. Location Service is operational 5. User has granted necessary location permissions
Postconditions:	1. Navigation route is displayed to user 2. Turn-by-turn directions are available 3. Real-time updates are active 4. Route is optimized based on current conditions
Priority:	High
Frequency of Use:	Very Frequent (Multiple times daily)
Flow of Events:	1. User requests navigation to selected car park 2. System initiates connection with Location Service 3. Location Service validates current location4 4. System requests route calculation: 4.1. Multiple route options 4.2. Traffic conditions 4.3. Road works/closures 4.4. Time estimates 5. Location Service returns route data 6. System displays navigation interface: 6.1. Map view with current location 6.2. Selected car park location 6.3. Recommended route 6.4. Alternative routes 6.5. Estimated Arrival Time (ETA) 7. System begins turn-by-turn navigation: 7.1. Voice guidance (if enabled) 7.2. Visual instructions 7.3. Distance to next turn 7.4. Lane guidance 8. System provides continuous updates: 8.1. Real-time traffic updates 8.2. Route recalculation if needed 8.3. ETA adjustments 9. System monitors approach to destination: 9.1. Entrance approach guidance 9.2. Final parking instructions 9.3. Pedestrian navigation to final destination

Alternative Flows:	1. Loss of GPS signal: 1.1. Cache last known location 1.2. Attempt to reestablish connection 1.3. Provide offline navigation support 2. Better route becomes available: 2.1. Alert user of faster route 2.2. Offer route switch option 3. User deviates from route: 3.1. Automatic route recalculation 3.2. Update navigation instructions
Exceptions:	1. Location Service unavailable: a. Switch to offline navigation mode b. Display cached maps if available c. Show static directions 2. Invalid destination coordinates: a. Request destination reconfirmation b. Suggest nearby valid locations 3. No route available: a. Display error message b. Suggest alternative destinations
Includes:	None (But interfaces with Location Services)
Special Requirements:	1. Real-time GPS integration 2. Traffic data integration 3. Multiple mapping service support 4. Offline navigation capability 5. Voice guidance system 6. Multiple language support 7. Battery optimization features
Assumptions:	1. Device has sufficient GPS accuracy 2. Location Service is reliable 3. Map data is current 4. Device has adequate processing power 5. Audio output is available for voice guidance
Notes and Issues:	1. Consider implementing: 1.1. AR navigation features 1.2. Parking level specific guidance 1.3. Indoor navigation capabilities 1.4. Integration with car park systems 1.5. Historical traffic pattern analysis 1.6. Battery consumption warnings 1.7. Navigation sharing features 2. Address: 2.1. GPS accuracy in urban canyons 2.2. Underground parking navigation 2.3. Multiple entrance handling 2.4. Signal loss recovery 2.5. Battery optimization 2.6. Data usage optimization

Use Case ID:	UC-014		
Use Case Name:	History		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	2 Feb 2025	Date Last Updated:	2 Feb 2025

Actor:	User
Description:	Allows users to view their past parking sessions, including details such as parking locations, timestamps, duration, and fees (if applicable). Users can filter, sort, or export history for reference.
Preconditions:	1. User is logged into the app. 2. Past parking session data exists in the system. 3. Device has internet connectivity (for real-time data retrieval).
Postconditions:	1. User successfully views parking history. 2. History is displayed in an organized manner. 3. Users can filter and sort past records. 4. Users can access session details when needed.
Priority:	Medium
Frequency of Use:	Occasional
Flow of Events:	1. User selects the "History" option from the app menu. 2. System retrieves parking history from stored records. 3. System displays a list of past parking sessions, including: a. Date and time of parking b. Parking location c. Duration of parking d. Parking fee 4. User can apply filters (e.g., by date, location) to refine search results (if needed). 5. User selects a specific record to view more details. 6. System displays detailed information for the selected parking session.
Alternative Flows:	1. No Parking History Available: a. System displays message: "No parking history available." 2. User Applies Filters: a. System updates the displayed records based on selected filters.
Exceptions:	1. System Failure: a. System displays error: "Unable to retrieve history. Please try again later." 2. No Internet Connection: a. System may display cached history (if available). b. Otherwise, show message: "Internet required to access full history."
Includes:	None
Special Requirements:	1. Secure storage of parking history. 2. User-friendly filtering and sorting options.

	3. Compliance with data privacy regulations (e.g., allow users to delete history).
Assumptions:	1. Parking session data is accurately recorded. 2. Users have necessary permissions to access history.
Notes and Issues:	1. Consider implementing: a. Frequent Locations feature based on history. b. Data visualization for insights into parking habits. c. History-based recommendations for parking spots.

Use Case ID:	UC-015		
Use Case Name:	Settings		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	2 Feb	Date Last Updated:	2 Feb

Actor:	User, Guest
Description:	Allows both guests and registered users to configure and personalize app preferences. Guests have access to basic settings such as theme customization and language selection, while registered users can manage account-related settings, privacy options, and notifications.
Preconditions:	1. The user or guest has accessed the settings menu. 2. System settings are available for modification.
Postconditions:	1. Preferences are successfully updated and saved. 2. System applies changes immediately or after a restart if necessary.
Priority:	High
Frequency of Use:	Occasional
Flow of Events:	1. Guest/User navigates to the "Settings" section in the app menu. 2. System displays available settings categories based on role: a. Guest Settings: i. Language Selection ii. Theme Customization (Light/Dark mode) iii. Basic Location Preferences b. User Settings: (Includes all Guest settings + additional options) i. Account Settings (Change password, email, phone number, delete account) ii. Notification Preferences (Enable/disable alerts, reminders) iii. Privacy Settings (Manage data sharing, location permissions) iv. Advanced Location Services (Enable/disable GPS tracking, set accuracy level) 3. Guest/User selects a setting to modify.

	4. System displays available options for the selected setting. 5. Guest/User updates the setting and confirms the changes. 6. System applies and saves the updated settings. 7. System notifies the guest/user of successful changes (if required).
Alternative Flows:	1. Settings Require App Restart: a. System informs the guest/user that certain changes (e.g., language selection) will take effect after restarting the app. 2. Guest Upgrades to Registered User: a. Additional settings become available after account creation. 3. User Cancels Changes: a. No modifications are made; system retains previous settings.
Exceptions:	1. System Error Prevents Updates: a. System displays error message: "Unable to update settings. Please try again later." 2. Network Dependency for Certain Settings: a. If settings require an internet connection (e.g., account changes), system displays a message: "Internet connection required to update this setting."
Includes:	None
Special Requirements:	1. Secure storage of user preferences. 2. Immediate feedback when settings are changed. 3. Accessibility support (e.g., larger fonts, voice assistance). 4. Multi-language support.
Assumptions:	1. Guests have limited settings access. 2. Users understand available settings and their impact. 3. System permissions allow modification of settings.
Notes and Issues:	1. Consider implementing: a. Backup & Restore Settings to sync preferences across devices. b. Customizable Quick Settings for frequently changed options. c. Privacy-focused options to comply with data protection regulations.

Use Case ID:	UC-016		
Use Case Name:	Guest Login		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	2 Feb	Date Last Updated:	2 Feb

Actor:	Guest
Description:	Allows a guest to access the app without creating an account. Guests can explore basic features such as searching for parking

	slots, viewing maps, and adjusting basic settings but will have limited access compared to registered users.
Preconditions:	1. The app is installed and launched.
Postconditions:	1. Guest successfully enters the app without an account. 2. Guest can access permitted features. 3. Guest mode restrictions are enforced.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	1. Guest opens the app. 2. System displays the Login/Sign Up screen with a "Continue as Guest" option. 3. Guest selects "Continue as Guest". 4. System provides access to guest-permitted features, such as: a. Viewing available parking locations b. Searching for car parks c. Navigating to a selected car park d. Adjusting basic settings (language, theme) 5. System displays a notification or prompt indicating that some features require an account (e.g., booking/reserving a parking slot, saving preferences, accessing history). 6. Guest can continue exploring the app or choose to register for full access.
Alternative Flows:	1. Guest Tries to Access a Restricted Feature: a. System prompts guest to sign up or log in to access that feature. 2. Guest Upgrades to a Registered User: a. Guest selects the option to create an account, and the system transitions to the registration process.
Exceptions:	1. App Requires Internet Connection: a. If the app needs real-time data (e.g., live parking availability) and the guest is offline, the system displays a message: "Internet connection required for real-time updates."
Includes:	None
Special Requirements:	1. Guest mode should have clear UI indicators to show limited access. 2. Guest access data should be temporary (e.g., session-based). 3. The app should encourage registration without forcing it.
Assumptions:	1. Guests will have fewer privileges than registered users. 2. Some features are only available to registered users. 3. Guest access does not store personal data permanently.
Notes and Issues:	1. Consider implementing: a. Seamless transition to account creation (e.g., if a guest tries to book a parking spot, they can sign up without losing progress). b. Temporary data storage (e.g., guest session expires after app restart).

	c. Guest session tracking to understand usage patterns.

Use Case ID:	UC-017		
Use Case Name:	Parking Location Memory Feature		
Created By:	Mayukhi	Last Updated By:	Mayukhi
Date Created:	4 Feb	Date Last Updated:	4 Feb

Actor:	User
Description:	Allows users to save their parking location by either taking a photo or writing a note within the app. This helps users remember where they parked, especially in large or crowded parking areas. The stored information can be accessed later when retrieving the car.
Preconditions:	1. User is logged into the app. 2. User has granted necessary permissions (camera, storage, location).
Postconditions:	1. Parking location is successfully saved. 2. User can retrieve the saved location, photo, or note when needed. 3. The data is stored until the user deletes it or a session expires.
Priority:	Medium
Frequency of Use:	Occasional
Flow of Events:	1. User selects the "Save Parking Location" option in the app. 2. System provides two options: a. Take a Photo of the parking spot. b. Write a Note (e.g., "Level B2, Zone C, Near Exit 3"). 3. User chooses an option and inputs the information. 4. System stores the saved parking details along with the timestamp and optional GPS location. 5. User can later access the saved location via the "Find My Car" section.
Alternative Flows:	1. User Takes Multiple Photos: a. System allows users to upload or capture multiple images for better reference. 2. User Edits or Deletes the Saved Location: a. System provides an option to update or remove the saved parking details.
Exceptions:	1. Camera or Storage Access Denied: a. System displays an error message and prompts the user to enable permissions. 2. Insufficient Storage: a. System warns the user and suggests freeing up space. 3. User Exits Before Saving:

	a. System discards unsaved data and returns to the main screen.
Includes:	None
Special Requirements:	1. Secure and temporary storage for saved locations. 2. Quick access to saved parking details. 3. Easy-to-use interface for capturing and retrieving information.
Assumptions:	1. Users may need to store multiple parking locations temporarily. 2. Saved data should not persist indefinitely unless manually deleted. 3. The system does not automatically track the car's location (manual input required).
Notes and Issues:	1. Consider implementing: a. Parking Timer Reminder to notify users about time limits. b. Offline Support for saving and retrieving notes/photos without an internet connection.