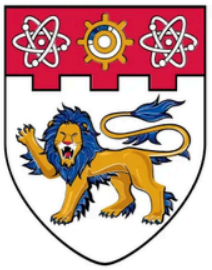


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 – Software Engineering

Lab 2 Deliverables

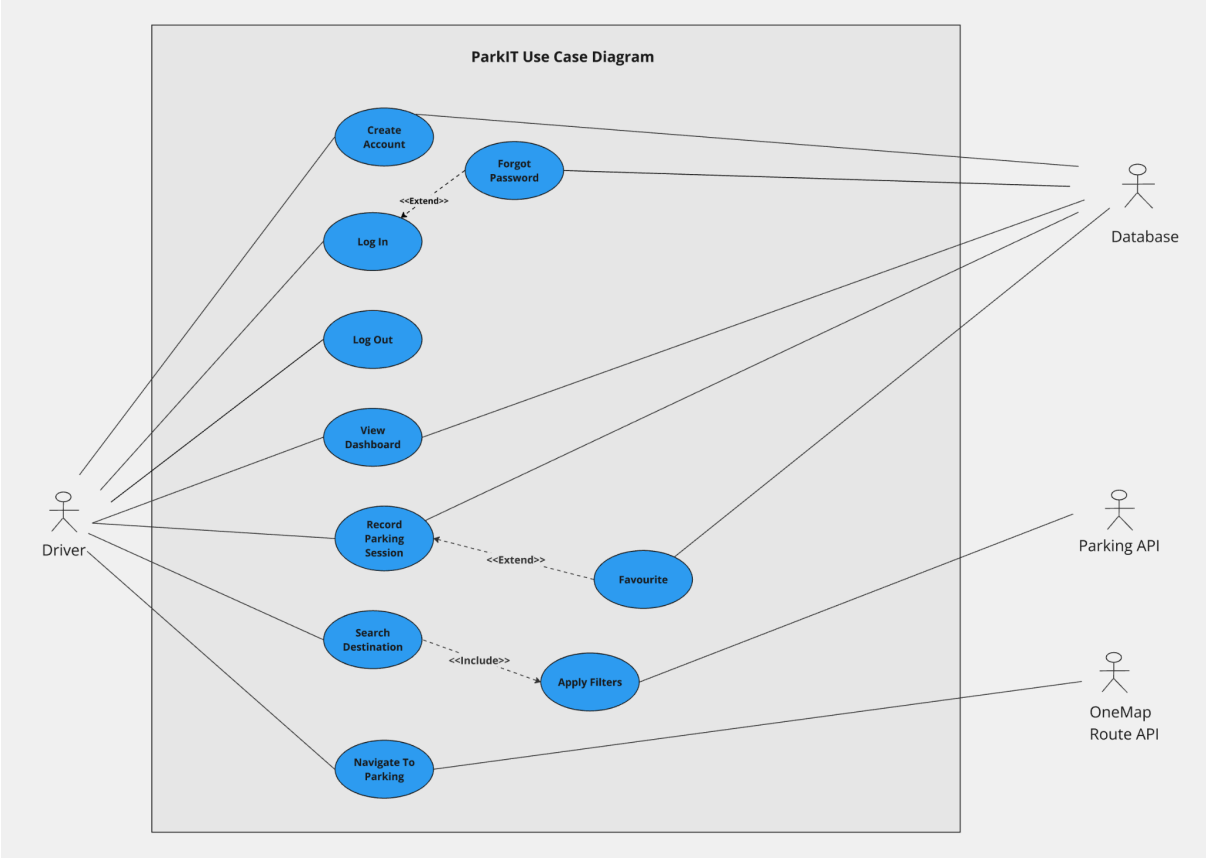
Lab Group	SCSF
Team	4 (Fantastic Four)
Members	Rashi Ojha (U2323323D) Sheth Shaivi Sachin (U2323665B) Dylan Quek Zhi En (U2321239L) Lee Zhuo Yang, Nicholas (U2423112F) Richard Chong Wing Liong (U2222047G)

Table of contents

Using UML (Unified Modeling Language) diagrams such as Use Case Diagrams, Sequence Diagrams, Class Diagrams, Boundary Class Diagrams, and Dialog Maps is crucial for designing, developing, and understanding a software system. These diagrams help visualize system structure, user interactions, and workflows, ensuring clarity among developers, designers, and stakeholders.

1. Use Case Diagrams & Descriptions

Use Case Diagram:



We used the use case diagram to define the system's scope by identifying key users and functionalities. It ensured all essential features were captured before development, validated requirements, and aligned the ParkIT app's functions like "Search Destination," "Apply Filters," and "Navigate to Parking" with user needs.

Use Case Descriptions:

1. Account Management

1.1. Create Account

Use case ID	AM1
Use Case	Create Account
Primary Actor	Driver
Description	This use case describes the process by which a driver registers for a new account on the ParkIT platform by clicking the “Sign Up” button on the registration UI.
Pre Conditions	1. The driver has navigated to the ParkIT website. 2. The driver has selected the option to create an account.
Post Conditions	1. A new driver account is created in the system with the role “Driver.” 2. The system shall add the driver’s verified email address and password to the database. 3. The driver shall be logged into the system. 4. The system shall display the Homepage.
Priority	High
Frequency of Use	Low
Flow of Events	1. When the app is launched, the Main page is displayed. 2. The driver clicks the “Sign up” button on the Main Page. 3. The Registration Form UI is loaded. 4. The system prompts the driver to input their email address. 5. The driver enters their email address then clicks the “verify email” button. 6. The system sends a verification link to the driver’s email id. 7. The system waits for the verification link to be clicked by the driver. 8. The user enters the password 9. The system verifies that the password is strong - more than 8 characters and a combination of numbers and strings. 10. The system logs the driver into their new account and redirects them to the Home Page.

Alternate Flow	**AF-S5 The driver clicks “Verify email” button without entering the email** 1. The system displays the message “Please enter a valid email address” 2. The system returns to Step 4 and waits for inputs from the us **AF-S5 The user inputs an email address that is already associated with an existing account** 1. The system displays the message “Email address is already associated with an account.” 2. The system returns to the Main Page. **AF-S7 The user has not verified their email address within the time limit** 1. The system displays the message “link has expired.” 2. The user can click the “Resend verification link” button and the system will send another verification link to the user’s email address. 3. The system returns to Step 6 and waits for the link to be verified. **AF-S9 The password entered isn’t strong** 1. The system displays the message “ password weak. Ensure password is 8 characters long” 2. The system returns to step 8 and waits for password input from driver.
Exceptions	**EX-1 The user did not receive the verification link** 1. The user can click the “Resend verification link” button and the system will send another verification link to the user’s email address. 2. The system returns to Step 6 and waits for the link to be verified.
Includes	None
Special Requirements	**For security, passwords must conform to a specific password policy:**

	1. Minimum length of 8 characters 2. At least 1 uppercase character 3. At least 1 lowercase character 4. At least 1 special character
Assumptions	None
Notes and Issues	None

1.2. Log in

Use Case ID	AM2
Use case	Log In
Primary Actor	Driver
Description	Driver can log in to their account to access the features of the app by using their username and password
Pre Conditions	1. The driver has navigated to the Main Page of the application. 2. The driver must have an existing account already registered in the system.
Post Conditions	1. The driver is authenticated and logged into the system 2. The system displays Home Page
Priority	High
Frequency of Use	High
Flow of Events	1. The driver launches the application and the Main Page is displayed 2. The driver clicks on the “Login” button and is then redirected to the login page 3. The driver inputs their username and password then clicks the “Log In” button. 4. The system validates the provided information using the database. 5. The system redirects the user to the Home Page.

Alternate Flow	**AF-S3 The user clicks the “Log In” button before completing all fields** 1. The system displays “Please complete all fields” error prompt. 2. The system returns to Step 3 and waits for inputs from the user. **AF-S4: The user inputs a non-existent username** 1. The system displays “No account has been created with this username. Please sign up instead.” error prompt. 2. The system returns to the Main Page. **AF-S4: The user inputs a wrong password** 1. The system displays “Incorrect password” error prompt. 2. The system returns to Step 3 and waits for inputs from the user.
Exceptions	None
Includes	(Extends) AM3 : Forgot Password
Special Requirements	None
Assumptions	None
Notes and Issues	None

1.3. Forgot Password

Use case ID	AM3
Use case	Forgot Password
Primary Actor	Driver
Description	This use case allows a driver to reset their password if forgotten.

Pre Conditions	1. The driver has navigated to the Login Page of the application. 2. The driver must have an existing account already registered in the system.
Post Conditions	1. The driver's account password is changed to their new password in the system database. 2. The driver is logged into their account.
Priority	High
Frequency of Use	Moderate
Flow of Events	1. The driver navigates to the Login Page. 2. The driver clicks the "Forgot Password?" button. 3. The system redirects the driver to the Forgot Password Page. 4. The driver enters their email address, then clicks the "Send email verification link" button. 5. The system waits for the email to be verified. 6. The system redirects the user to the Change Password Page. 7. The driver enters their new password and confirms their password,then clicks the "Change Password" button. 8. The system validates the provided information, then redirects the driver to the Home Page.
Alternate Flow	**AF-S4 The driver clicks "Send email verification link" button without entering the email** 1. The system displays the message "Please enter a email address" 2. The system returns to Step 4 and waits for inputs from the us **AF-S4: The driver inputs an email address that is not associated with an existing account** 1. The system displays "No account has been created with this phone number. Please sign up instead." error prompt. 2. The system returns to the Login Page. **AF-S7: The user clicks the "Change Password" button**

	before completing all fields 1. The system displays “Please complete all fields” error prompt. 2. The system returns to Step 7 and waits for inputs from the driver. **AF-S7: The password and confirm password fields do not match** 1. The system displays “Passwords do not match” error prompt. 2. The system returns to Step 7 and waits for inputs from the driver. **AF-S7: The driver enters a password that does not meet the strength requirements** 1. The system displays “Please use a stronger password” error prompt. 2. The system returns to Step 7 and waits for inputs from the user.
Exceptions	**EX-1 The user did not receive the verification link** 1. The user can click the “Resend verification link” button and the system will send another verification link to the user’s email address. 2. The system returns to Step 4 and waits for the link to be verified.
Includes	None
Special Requirements	For security, passwords must conform to a specific requirement: 1. Minimum length of 8 characters 2. At least 1 uppercase character 3. At least 1 lowercase character 4. At least 1 special character
Assumptions	None
Notes and Issues	None

1.4. Logout

Use case ID	AM4
Use case	Logout
Primary Actor	Driver
Description	Allows driver to log out of their session.
Pre Conditions	1. The driver is at the Home Page 2. The driver is already logged into their account
Post Conditions	1. The user is logged out of their account. 2. The system displays the Main Page.
Priority	High
Frequency of Use	Medium
Flow of Events	1. The driver navigates to the Home Page 2. The driver clicks the “Logout “ button. 3. The system logs the driver out of their account 4. Driver is redirected to the Main Page.
Alternate Flow	None
Exceptions	None
Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

2. Parking Search and Navigation

2.1. Search Destination

Use Case ID	PS1
Use case	Search Destination

Primary Actor	Driver
Description	This use case allows a driver to search for available car parks near a specified destination. The driver enters a destination, and the system retrieves and displays a list and map view of nearby car parks along with details such as availability, pricing, and parking type.
Pre Conditions	1. The driver is on the Search Parking Destination Page 2. The search bar is empty
Post Conditions	1. The system stores the chosen parking location for navigation.
Priority	High
Frequency of Use	High
Flow of Events	1. The driver navigates to the Search Parking Destination Page 2. The search bar is empty. 3. The driver enters their destination. 4. The system validates the destination. 5. The system loads all available parking spaces near the entered destination. 6. The driver can click on the “Apply Filters” button and the system will adjust the results based on the included use case Apply Filters. 7. The driver selects a parking spot and confirms it. 8. The system stores the location for navigation.
Alternate Flow	**AF-S3: The location entered does not exist** 1. The system displays message “ Location not found” 2. The driver is prompted to re-enter the destination as the system returns to step 2 **AF-S5 No parking spaces are available within the default 3km radius** 1. The System displays message “ Parking not found” 2. The System displays message “ Apply filters to increase the search radius”
Exceptions	None
Includes	(Includes) Apply Filters (FM1)
Special Requirements	None
Assumptions	None

Notes and Issues	None

2.2 Navigate to Parking

Use case ID	PS2
Use case	Navigate to parking
Primary Actor	Driver
Description	This use case enables the system to provide real-time navigation instructions to guide the driver to the selected carpark using the OneMap API. It also provides the estimated time to arrival on a real-time basis.
Pre Conditions	The driver has selected a carpark.
Post Conditions	The system displays the travel route from the user's chosen start location to destination on a map.
Priority	High
Frequency of Use	High
Flow of Events	1. The driver selects the Navigate button. 2. The system redirects the driver to Navigation Page 3. The system displays the location of the selected car park. 4. The driver clicks the "Find Route" button. 5. The system validates the provided information, then displays the travel route from the start location to the destination on the map. 6. The system will constantly update the driver's ETA by using the OneMap API.
Alternate Flow	**AF-S4 The driver clicks on "Find Route" but no car park was selected** 1. The system displays " Please Select a parking" 2. The driver is redirected to the Search Parking Destination Page **AF-S3: Geolocation tracking is disabled in the driver's device settings** 1. The system requests the driver to "Turn ON GPS" settings. 2. The system returns to step 2
Exceptions	None

Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

2.3 Record parking session

Use case ID	PS3
Use case	Record Parking Session
Primary Actor	Driver
Description	This use case handles the storage and management of parking information, such as cost, location, and duration.
Pre Conditions	1. The driver is on the Navigation UI. 2. The driver has reached the desired parking space.
Post Conditions	1. Parking session information, including total duration, cost and location, is stored in a database. This information will be accessible for future reference from the driver's dashboard.
Priority	High
Frequency of Use	High
Flow of Events	1. The driver has reached the desired parking spot. 2. The driver clicks on the button “Start Parking Session” 3. The system continues timer and waits for the driver to “Stop Parking Session” 4. The driver clicks on the “Stop Parking Session” button. 5. The system records the time spent on parking and calculates the total cost. 6. The system displays the information and the driver is given the option to add the parking space as a favourite using the favourite use case (SF1) 7. The information is recorded into the database after the driver clicks on the “OK” button.
Alternate Flow	**AF-S3 Driver forgot to press the “Stop Parking Session” manually** 1. The system detects prolonged inactivity (exceeding a predefined limit of 12 hours) and automatically

	stops the session. 2. The system will not record this session because the parking duration is inaccurate. 3. The driver is notified about the auto-stopped session.
Exceptions	None
Includes	(Extends) Save Favourites
Special Requirements	None
Assumptions	None
Notes and Issues	None

3. Filter Management

3.1. Apply Filter

Use case ID	FM1
Use case	Apply Filter
Primary Actor	Driver
Description	This use case allows the driver to apply filters such as filter by parking type, filter by distance radius and filter by cost to optimise based on their preferences.
Pre Conditions	1. The driver has searched for the destination. 2. The destination has been validated
Post Conditions	1. The parking list updates based on the chosen preference
Priority	High
Frequency of Use	High

Flow of Events	1. The driver opens the filter menu. 2. The system displays available filter options, including: Price, Parking type and Distance radius 3. The driver selects and applies one or more filters. 4. The system validates the filter input and updates the search results accordingly. 5. The system re-displays the filtered parking spaces
Alternate Flow	**AF-S4: No Matching Results:** 1. The system displays a message indicating that no car parks match the selected filter criteria. 2. The system prompts the driver to adjust the search criteria
Exceptions	None
Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

4. Save Favourites

Use case ID	SF1
Use case	Favorites
Primary Actor	Driver
Description	This use case allows drivers to save their favorite parking locations for easy access in the future. Any favourites stored will be highlighted when the driver searches for the same location.
Pre Conditions	1. Drivers must be logged in to their account. 2. The driver must have completed navigation. 3. The parking session must have been completed
Post Conditions	1. Selected parking location is saved to the driver's favorite list. 2. Saved locations will appear as favorites in future searches. 3. Drivers can jump to these locations during a search.
Priority	Medium

Frequency of Use	Moderate
Flow of Events	1. Driver views the recorded time spent in the parking area. 2. Driver selects the “Add to Favorites” button. 3. The system validates the location. 4. System saves the location as a favorite. 5. The system displays a confirmation message indicating the location has been added successfully.
Alternate Flow	None
Exceptions	None
Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

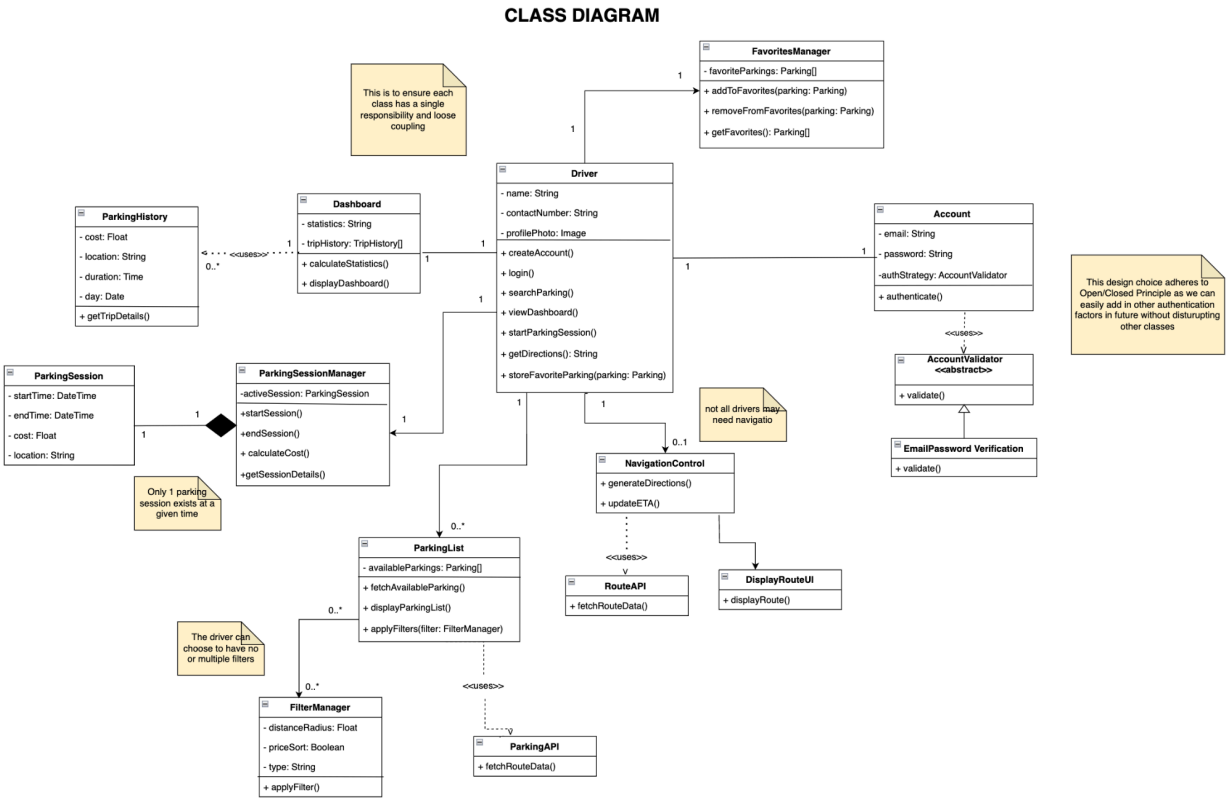
5. View Dashboard

Use case ID	VD1
Use case	View Dashboard
Primary Actor	Driver
Description	This use case enables a driver to review detailed statistics of their parking history. The dashboard displays data such as session durations, total costs, and monthly summaries, often using visual aids like bar charts for clarity.
Pre Conditions	1. The driver has completed one or more parking sessions, and the system has been tracking parking history. 2. The driver has logged into the system.
Post Conditions	1. The driver is presented with detailed statistics on their parking usage, including monthly summaries and visual representations (bar charts).
Priority	Medium
Frequency of Use	Moderate
Flow of Events	

	1. The driver selects the “View Dashboard” option from the menu. 2. The system retrieves the driver’s parking history information from the database 3. The system processes the retrieved data to calculate monthly statistics. 4. The system then displays the statistics using bar charts.
Alternate Flow	**AF-S2: No parking Session has been recorded** 1. The system displays message “No session recorded” 2. The driver is redirected to the dashboard page
Exceptions	None
Includes	None
Special Requirements	None
Assumptions	None
Notes and Issues	None

2. Class Diagrams

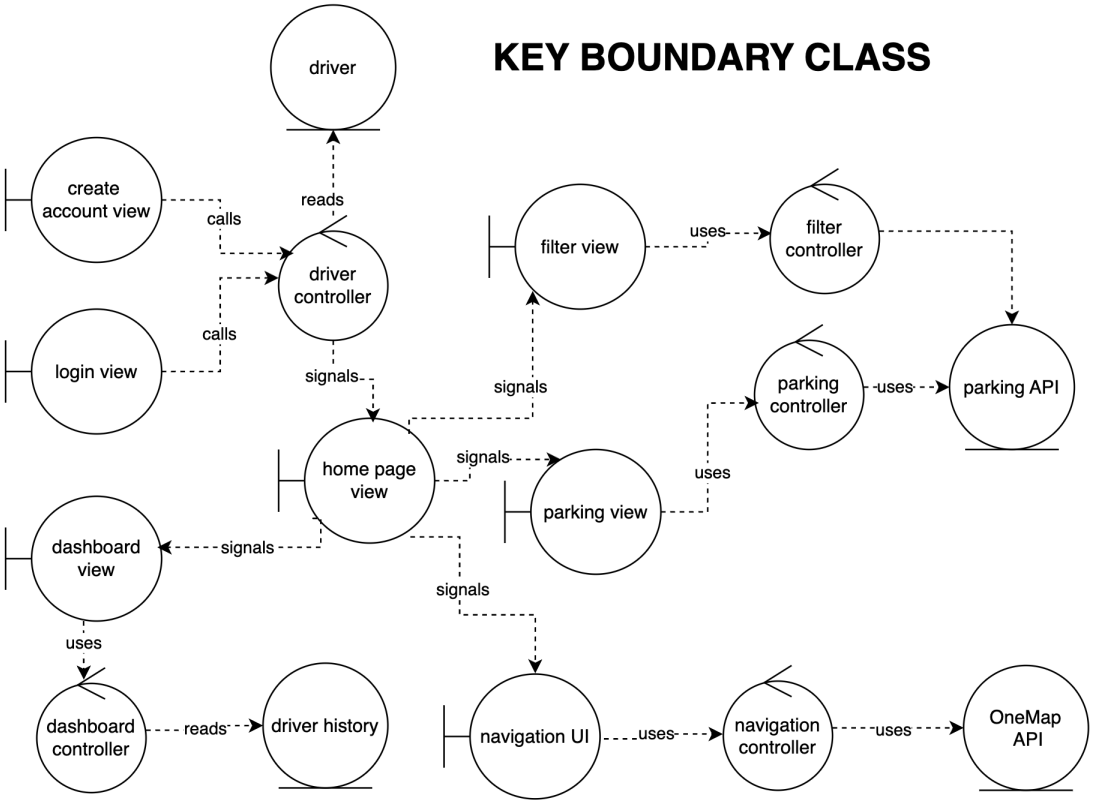
We used the class diagram to define the system’s structure by outlining classes, attributes, methods, and relationships. It ensured a modular design in the ParkIT system by establishing entities like "Driver," "ParkingSessionManager," "FavoritesManager," and "NavigationControl," aiding in database and architecture planning.



ParkingSession entity is recording the current ongoing parking whereas the ParkingHistory contains record of all the recorded parking sessions.
We have applied the OOP concepts (SOLID) that we learnt to better divide the entity classes and adhere to all design principles.

3. Key Boundary Class Diagrams

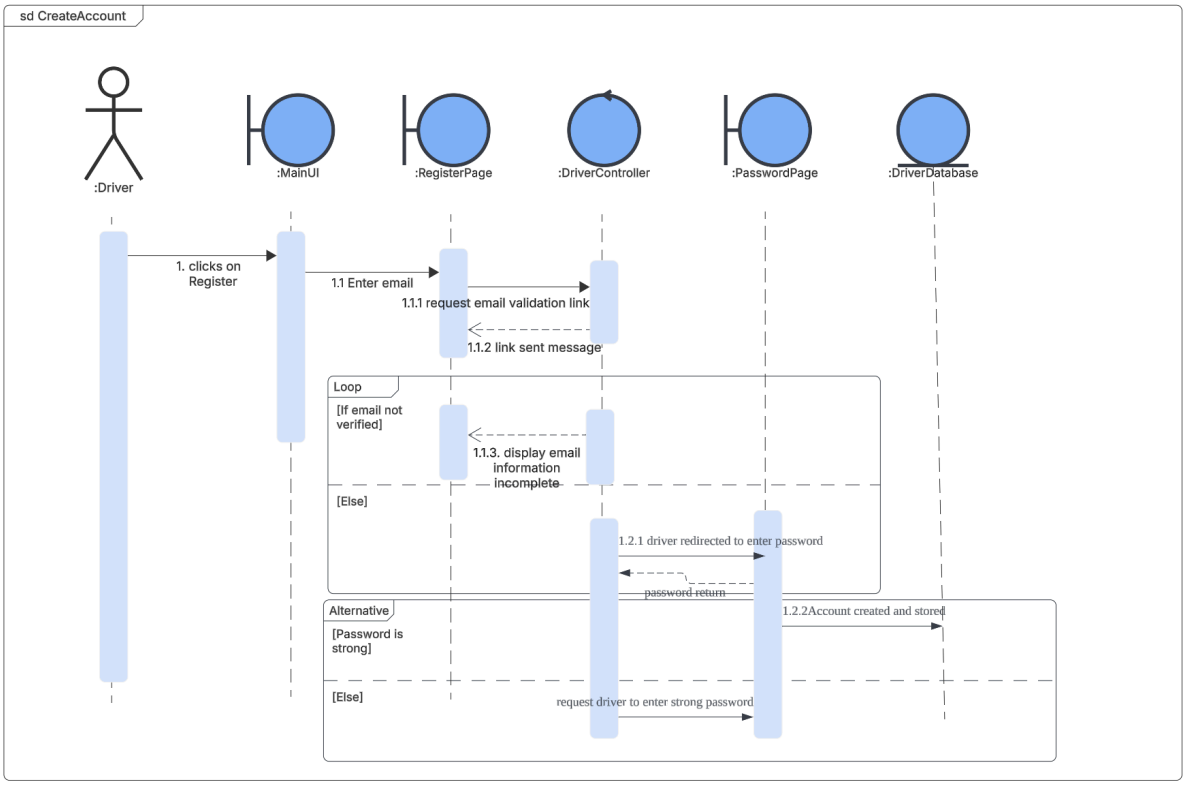
We used the boundary class diagram to illustrate interactions between the UI, controllers, and backend. In the ParkIT system, it showed how views like the "Home Page View" connect with controllers like the "Driver Controller" to fetch relevant data.



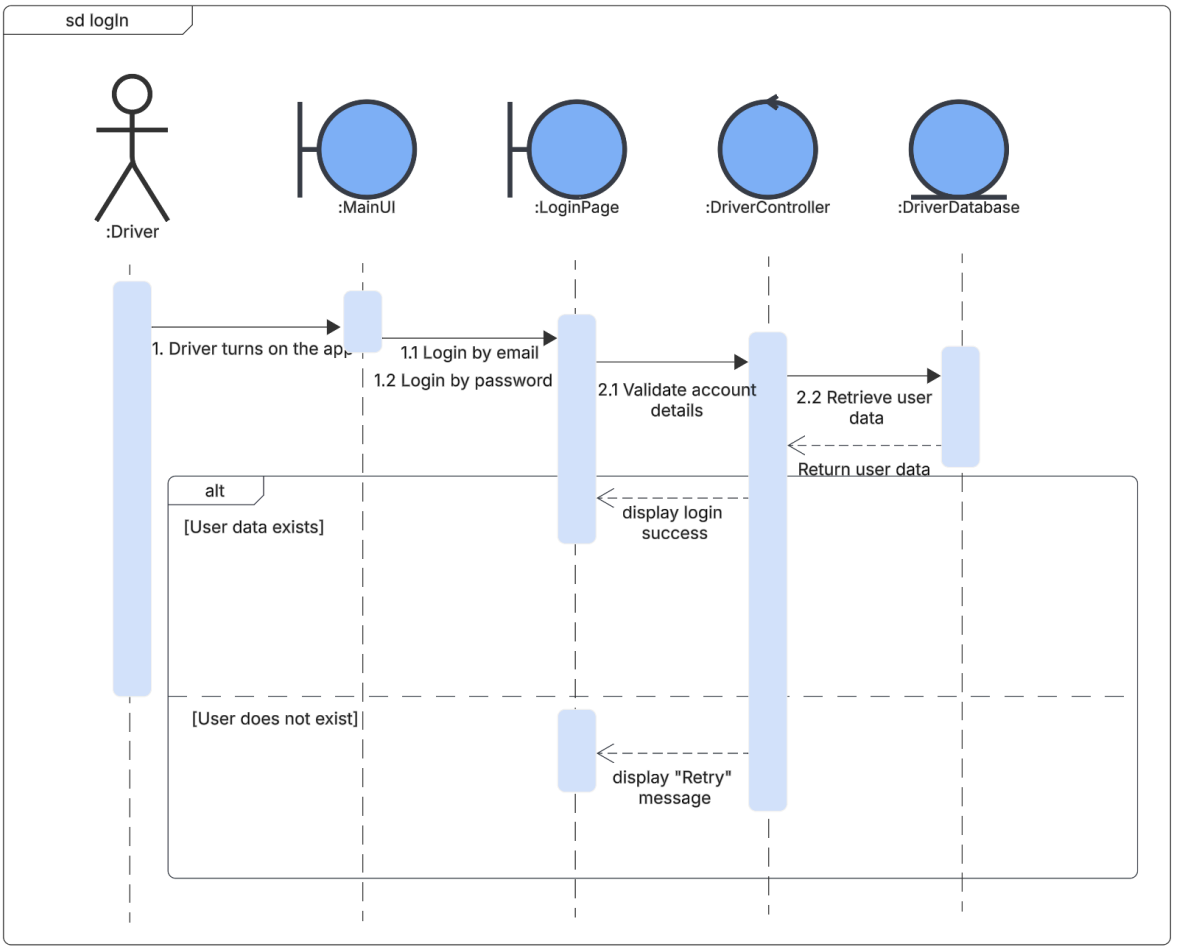
4. Sequence Diagrams

We used the sequence diagram to visualize the flow of interactions between system components over time.

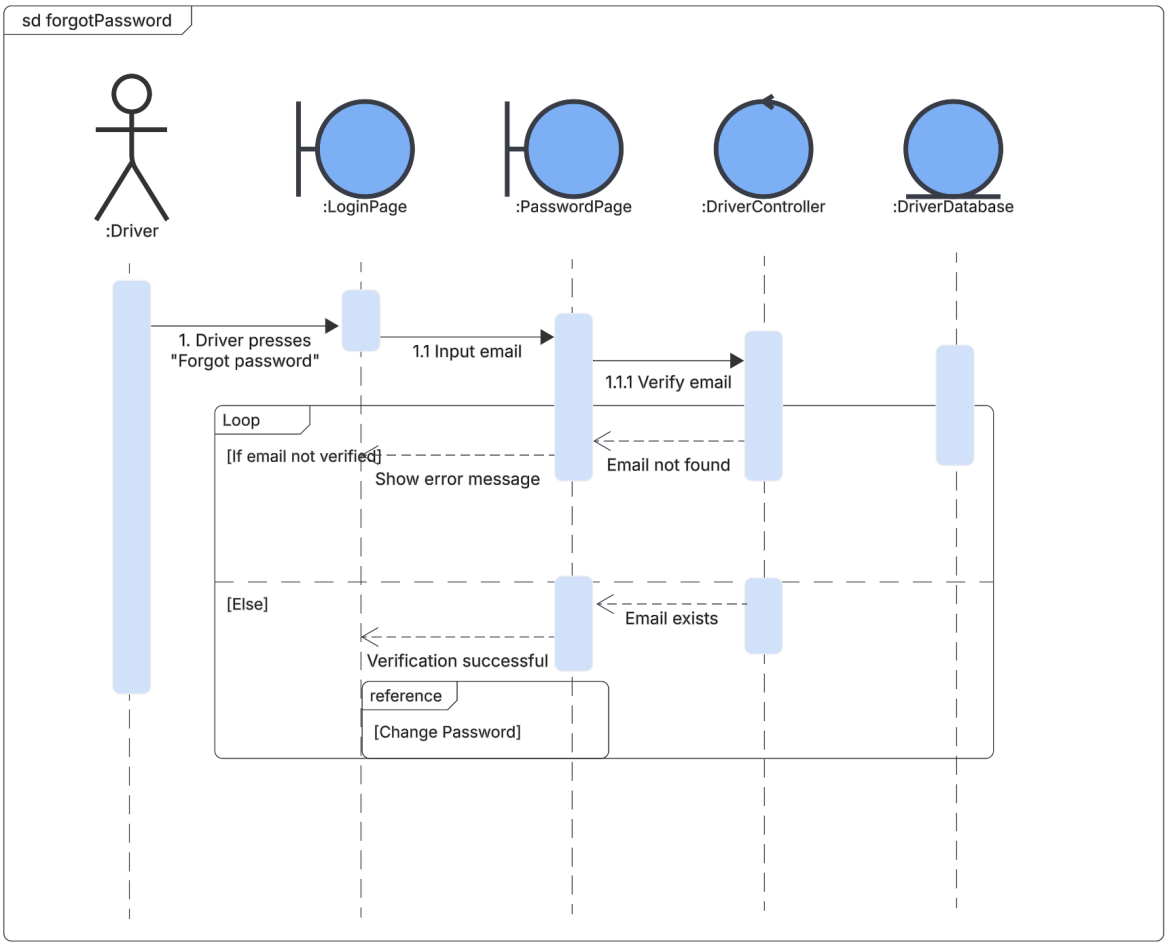
1. Create Account


sd CreateAccount
:Driver
:MainUI
:RegisterPage
:DriverController
:PasswordPage
:DriverDatabase
1. clicks on Register
1.1 Enter email
1.1.1 request email validation link
1.1.2 link sent message
Loop
[If email not verified]
1.1.3. display email information incomplete
[Else]
1.2.1 driver redirected to enter password
password return
Alternative
[Password is strong]
1.2.2Account created and stored
[Else]
request driver to enter strong password

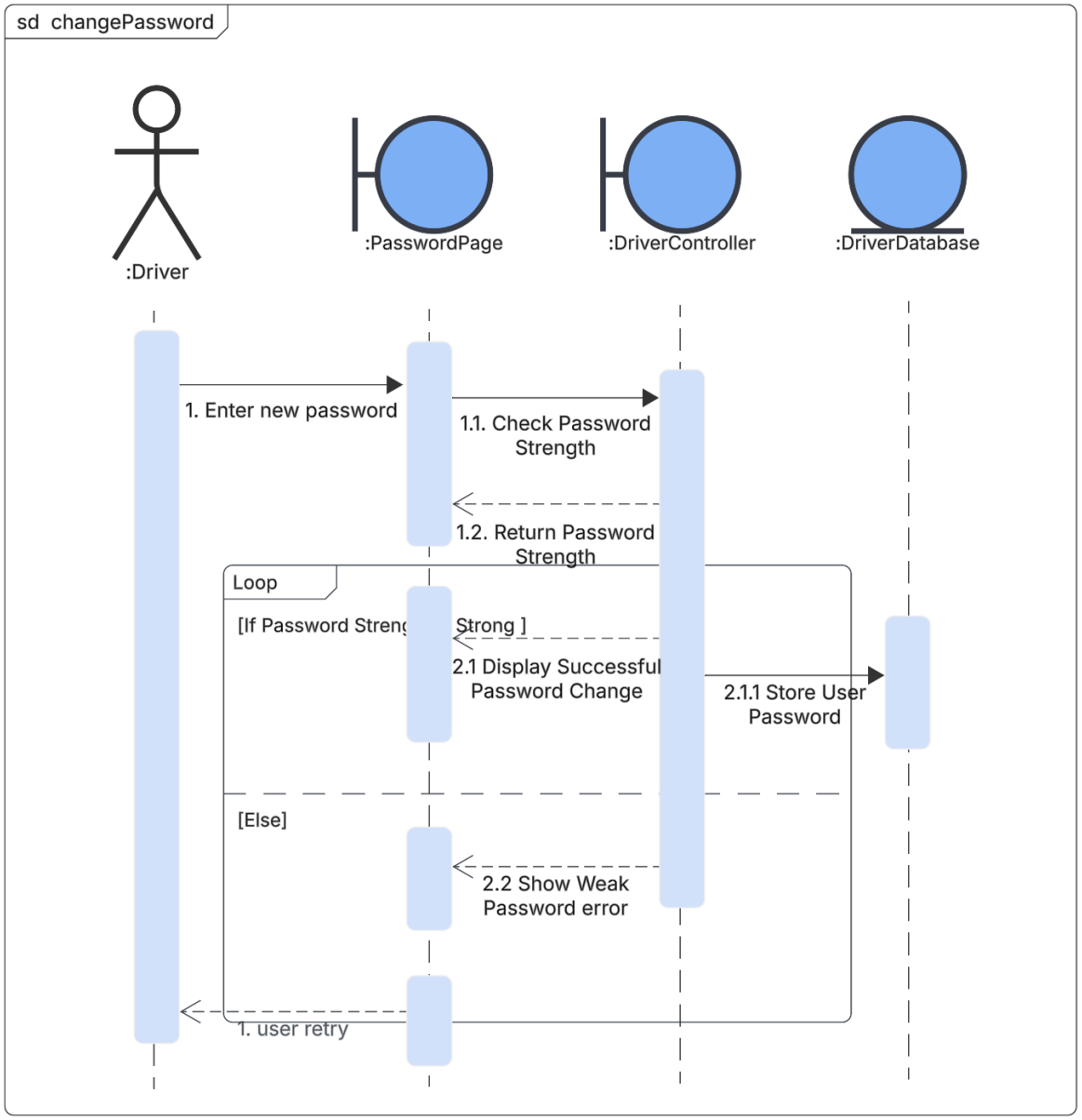
2. Log In



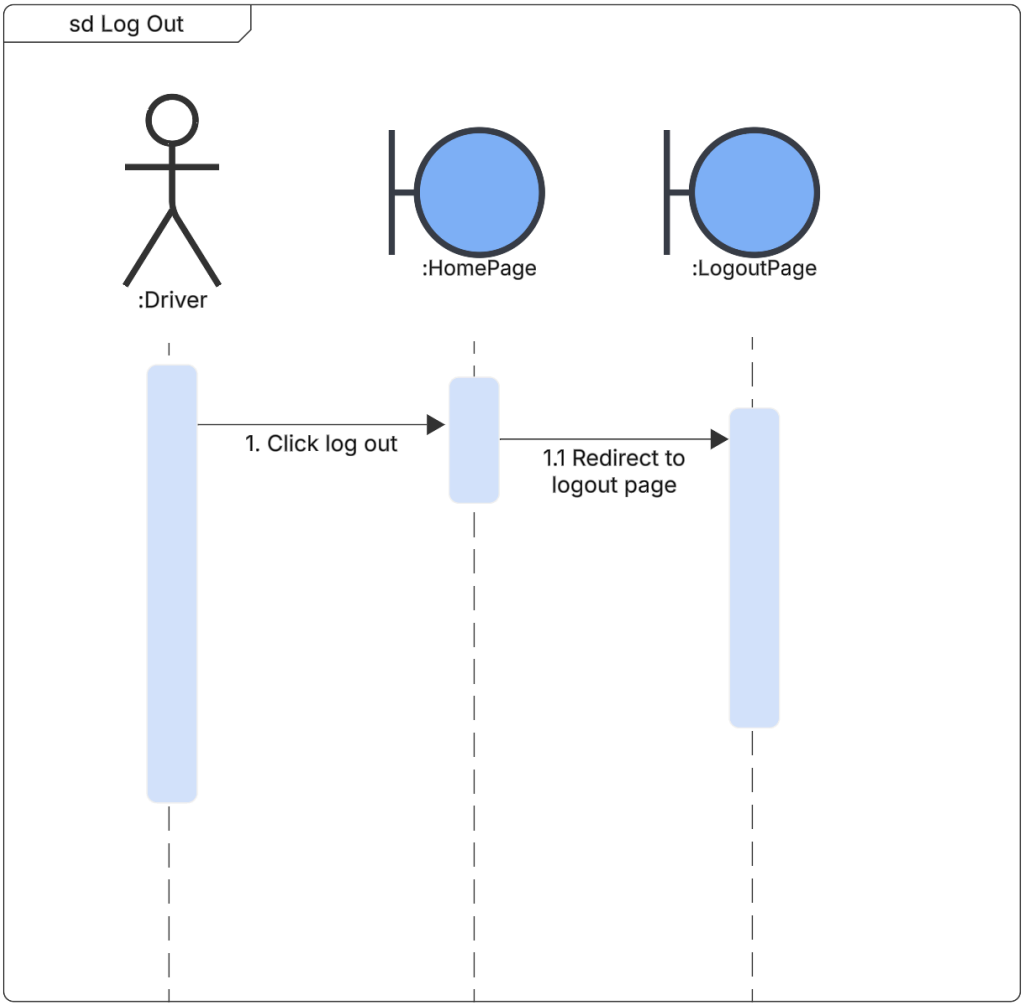
3. Forgot Password


sd forgotPassword
:Driver
:LoginPage
:PasswordPage
:DriverController
:DriverDatabase
1. Driver presses "Forgot password"
1.1 Input email
1.1.1 Verify email
Loop
[If email not verified]
Show error message
Email not found
[Else]
Email exists
Verification successful
reference
[Change Password]

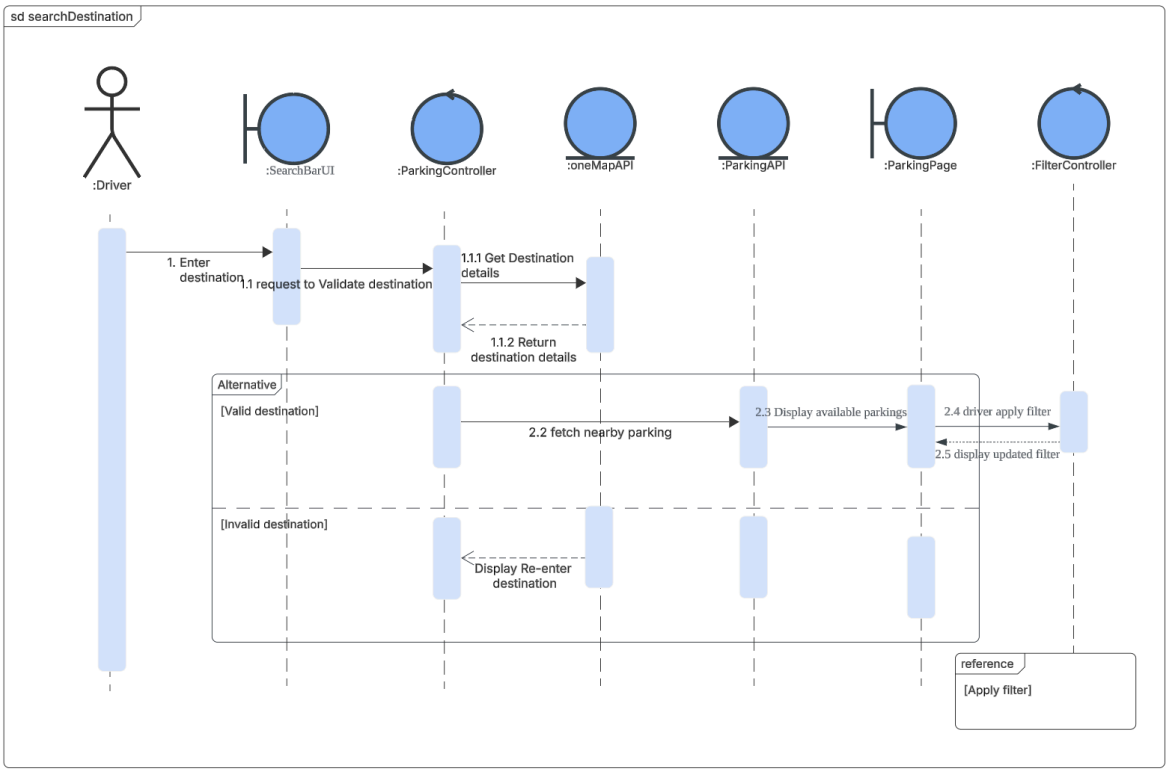
4. Change Password



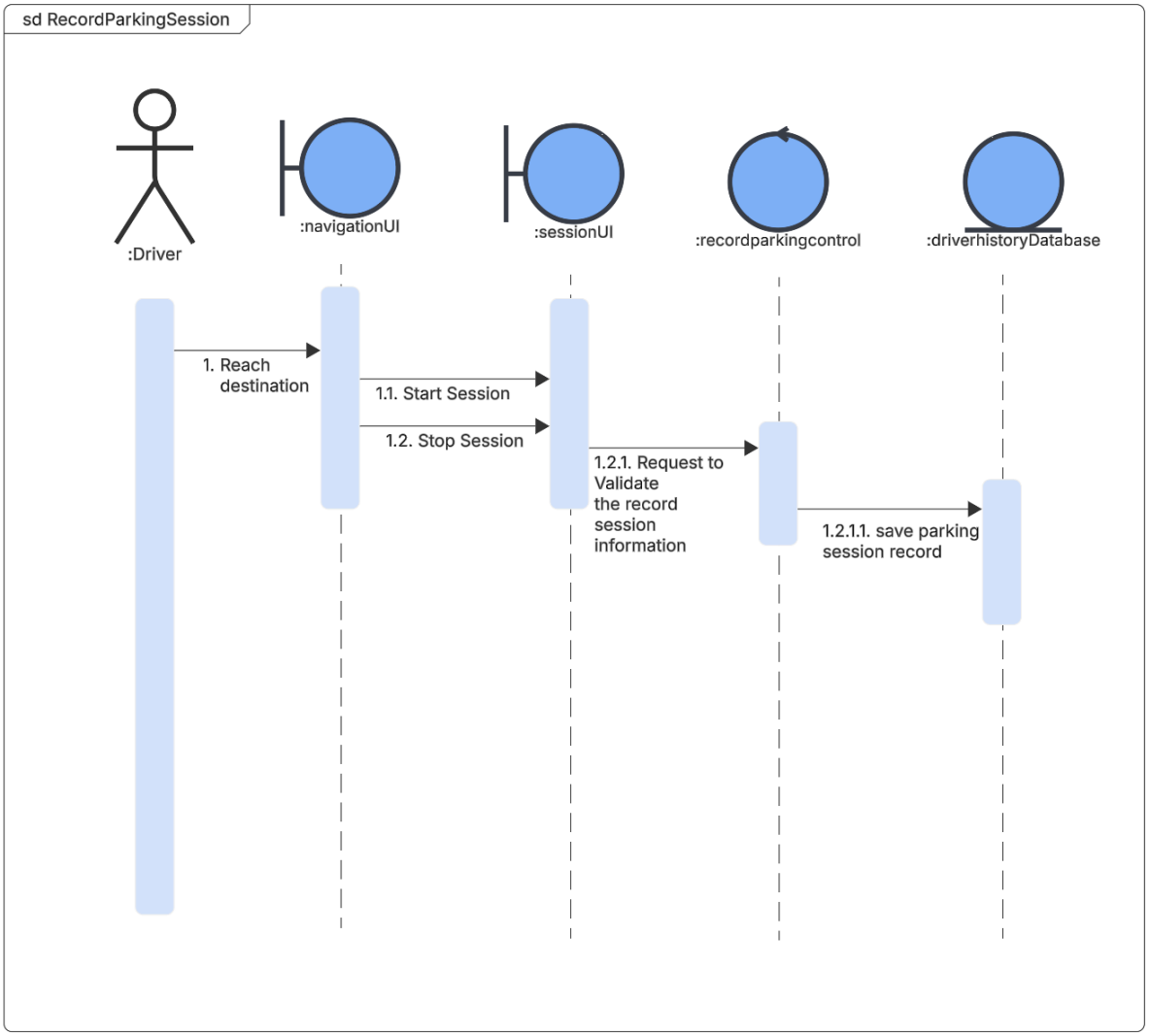
5. LogOut


sd Log Out
:Driver
:HomePage
:LogoutPage
1. Click log out
1.1 Redirect to logout page

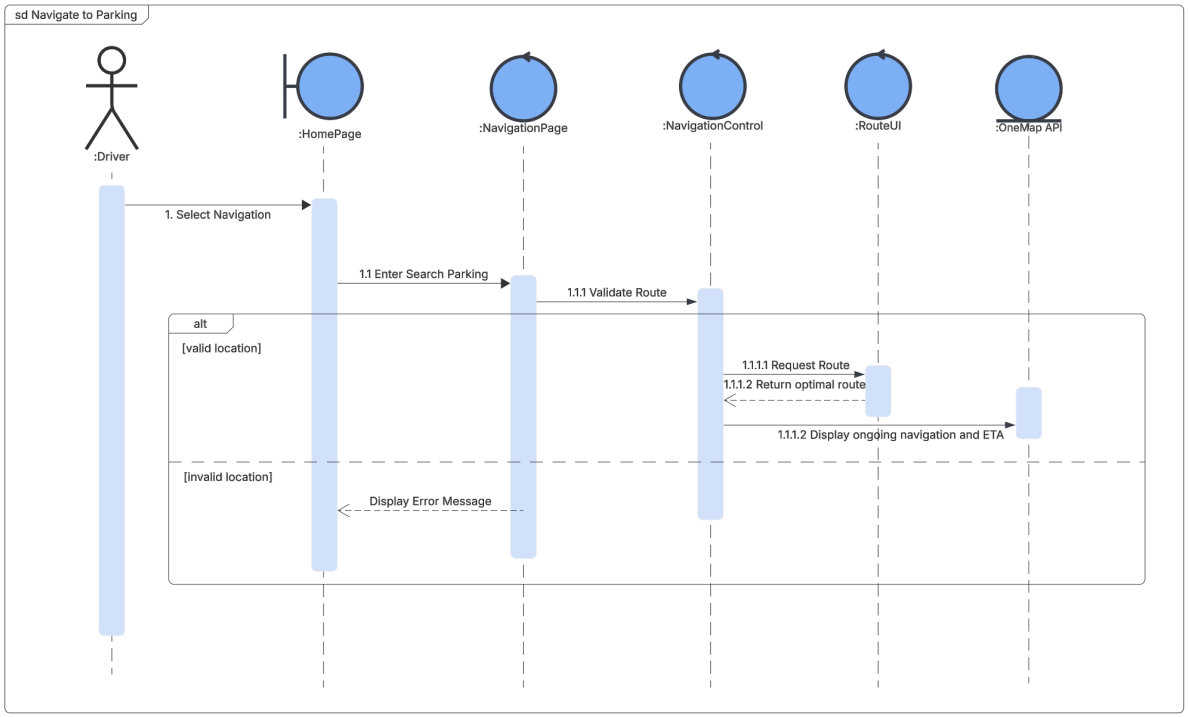
6. Search Destination



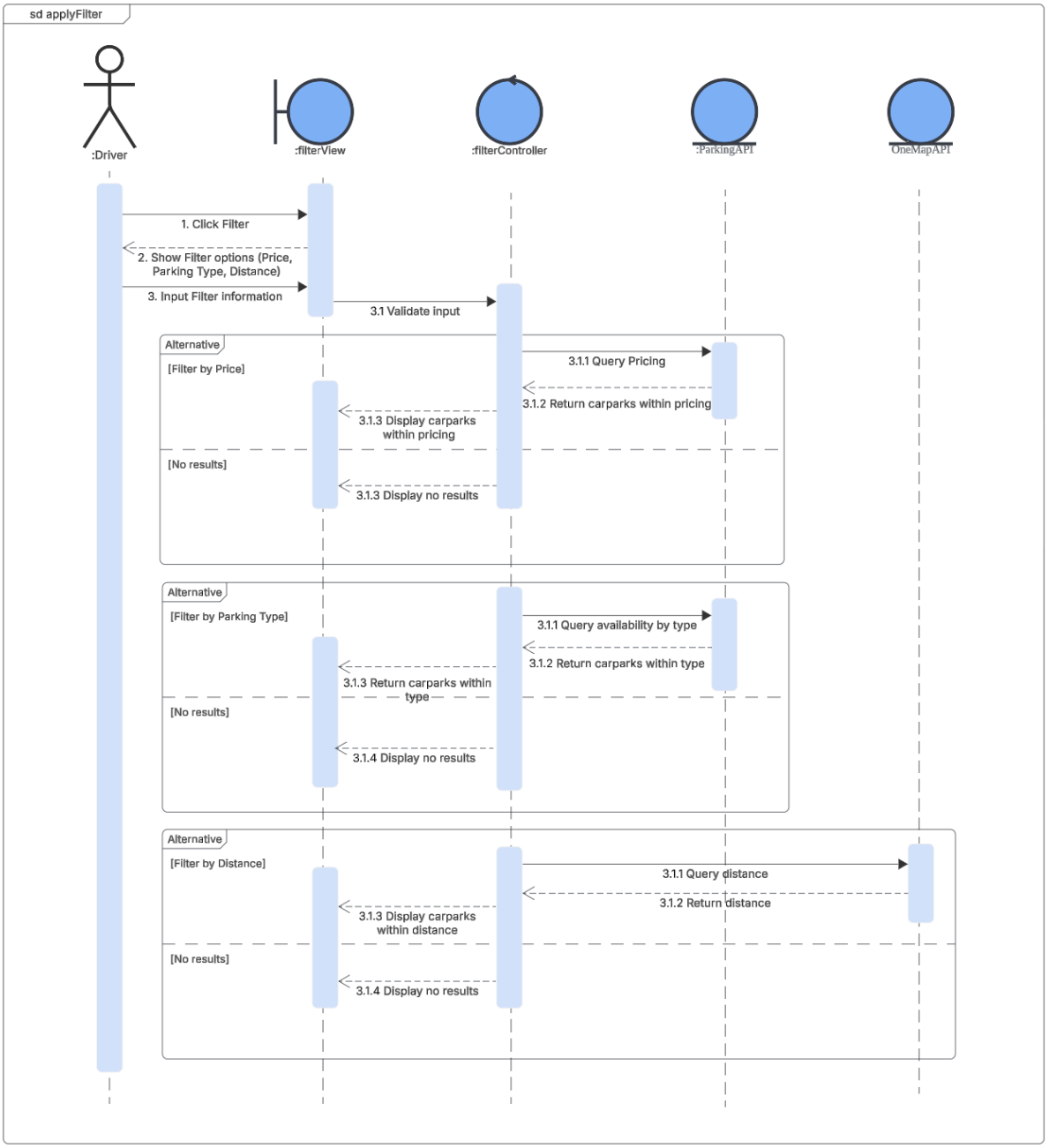
7. Record Parking Session



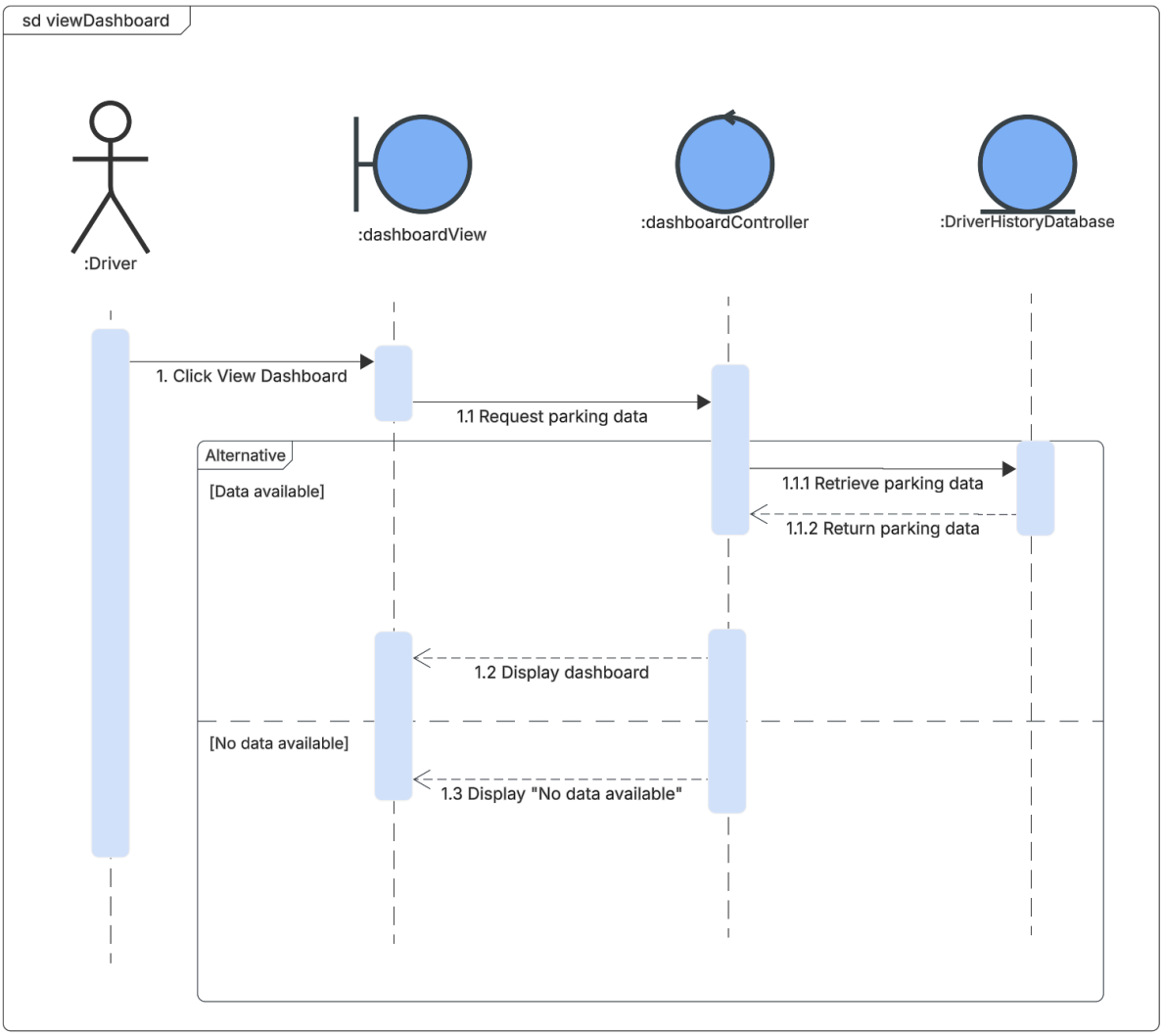
8. Navigate to Parking



9. Apply Filter



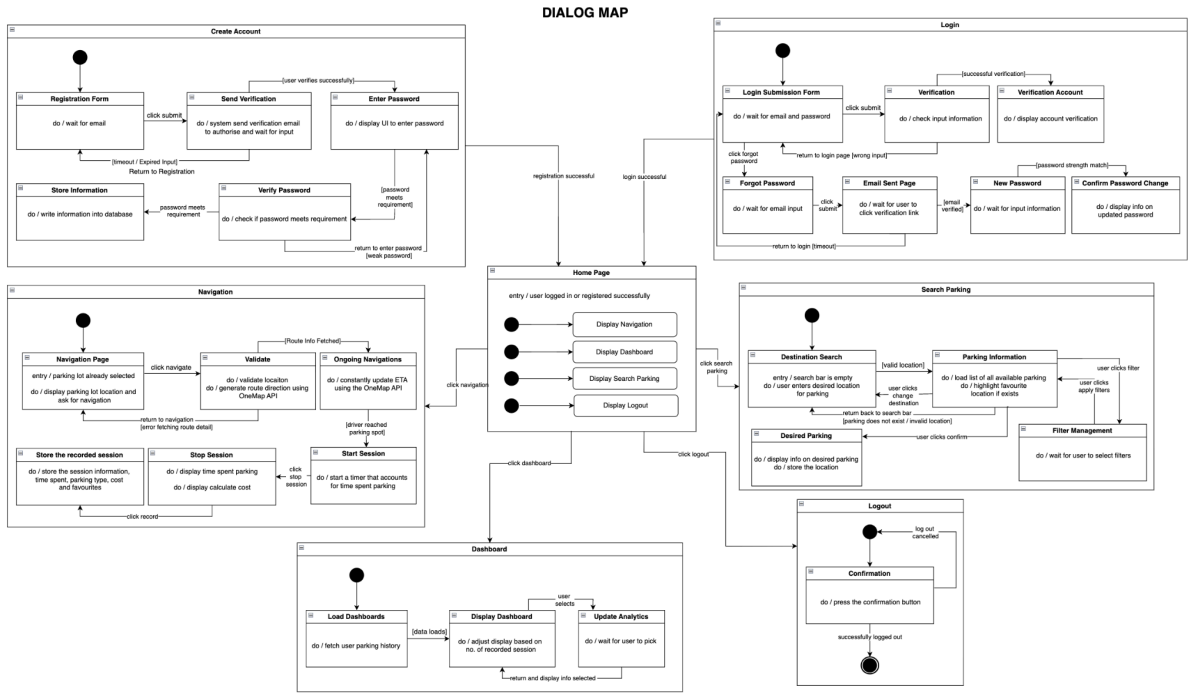
10. View Dashboard



For better clarity of diagrams (sequence diagrams) → https://lucid.app/lucidchart/4f92e4c0-23b4-48ca-82bb-b1ddb6d0ccdb/edit?invitationId=inv_513a4004-5dc9-466c-a8e2-cd08250ef220&page=6CyUaPecTbe1#/documents?folder_id=398320405

5. Dialog Map

Lastly, creating the dialog map helped us in finding missing or incorrect navigation pathways. This also helped us spot opportunities to reuse and redundancies in the user interface without worrying about the details of screen design prematurely.



For better clarity of diagrams (dialog map, class diagram & boundary class) → https://app.diagrams.net/#G19d_mkqy9IssZ8j5j-Ubi_7HGHIX3Qn6Y#%7B%22pageId%22%3A%22g7e6bmTCwa9WViMW9q3T%22%7D