



# ParkMate

## PRESENTATION

BRYANT GOH JUN XIAN U2423462J  
GUAN YIBIN U2423353E  
GOH JIN LONG ABDILLAH U2321634L  
HARSHIL GUPTA U2421166J  
KUMAR PREETHAM U2422986F

---

# Agenda

1

INTRODUCTION

2

SOLUTION

3

SYSTEM  
DESIGN

4

FUNCTIONAL  
REQUIREMENT  
TRACEABILITY

5

FURTHER  
IMPROVEMENTS

6

CONCLUSIONS

# Introduction

# Why our app?

NUMBER OF EVS CONTINUE TO SURGE AS THE YEARS GO BY



EV OWNERS STRUGGLE TO FIND PARKING AND CHARGING SPOTS ON THE GO



FINDING PARKING/CHARGING SPOTS JUST TO FIND OUT THAT THEY ARE OVERPRICED, FULL OR INCOMPATIBLE



PARKING APPS TREAT EV CHARGING AS AN AFTERTHOUGHT, LEAVING DRIVERS WITH FRAGMENTED INFORMATION AND NO REAL-TIME VISIBILITY



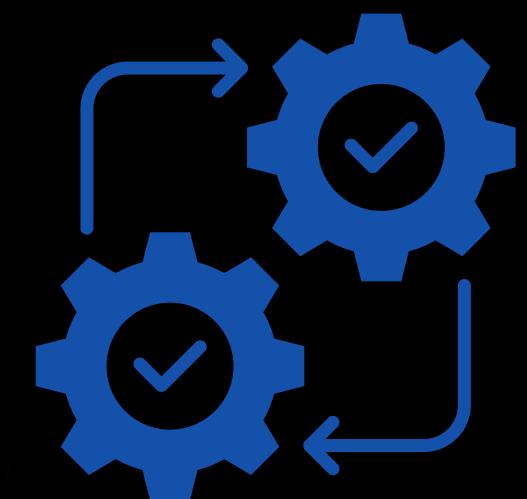
# Our initiative

updates

LIVE UPDATES ON PARKING  
AVAILABILITY, NEARBY CHARGING  
PORTS, RATES AND CONNECTOR TYPES



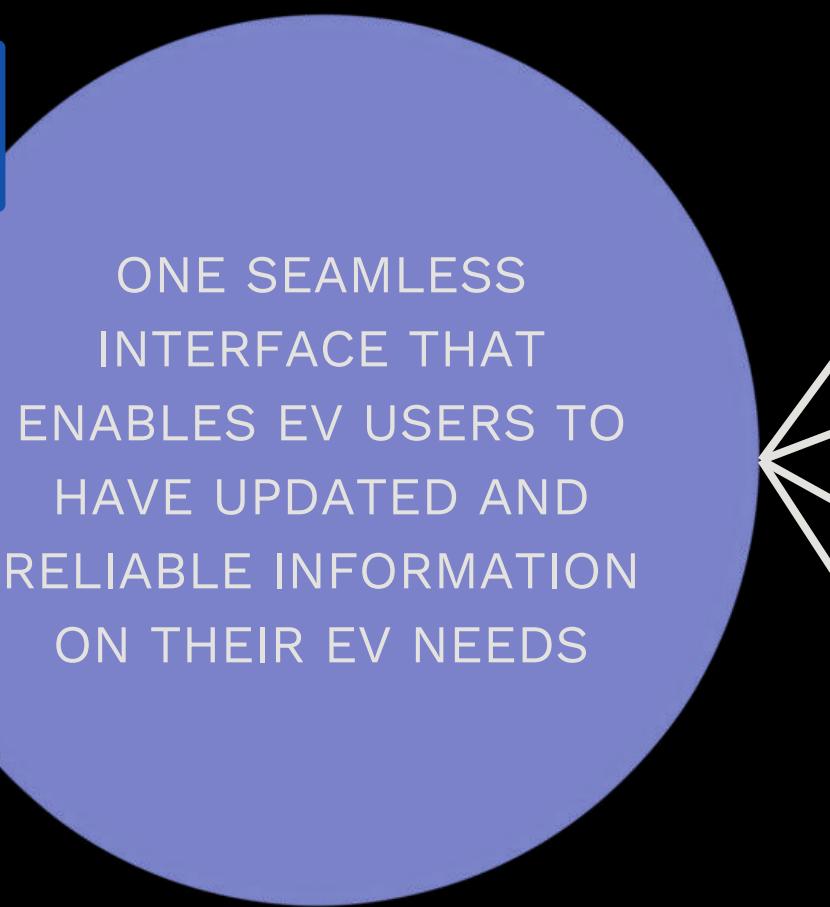
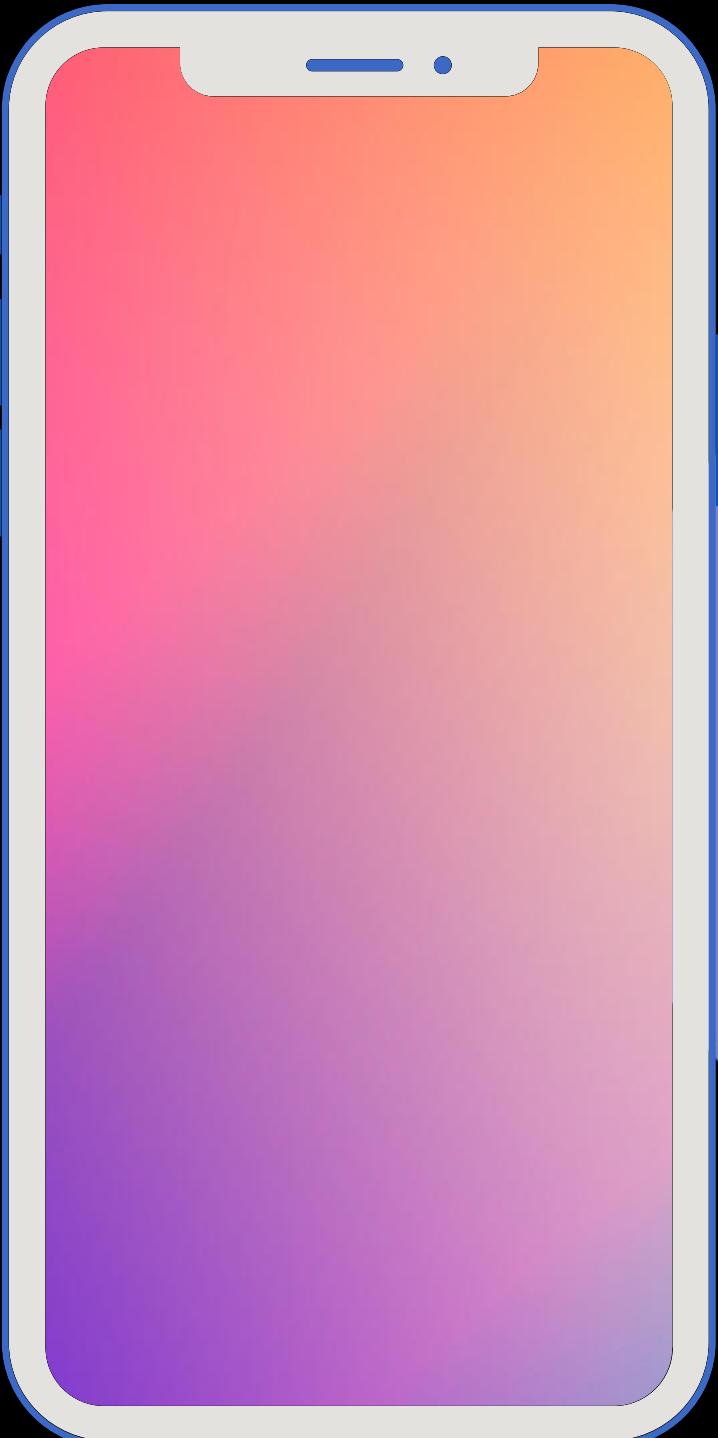
FASTER, SMARTER PARKING  
DECISIONS, REDUCING STRESS,  
TRAFFIC CONGESTION AND  
RANGE ANXIETY



INTEGRATES EVERYTHING INTO  
ONE SEAMLESS INTERFACE

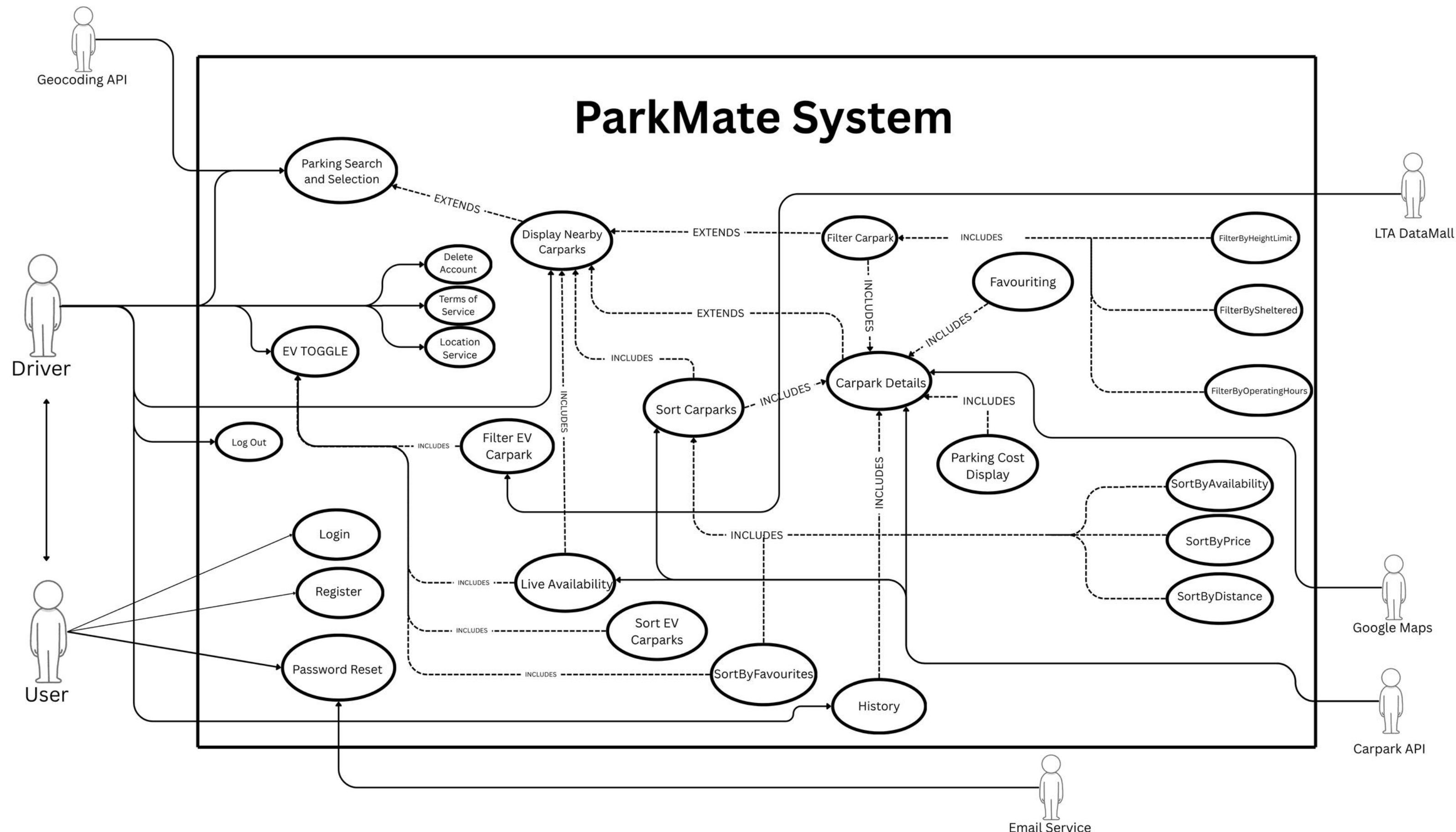
# Solution

# ParkMate



- CARPARK SEARCH FUNCTIONS**  
 OUR APP PROVIDES LIVE UPDATES ON PARKING SPOTS AROUND THE USER, INCLUDING AVAILABILITY AND COST FOR ALL DRIVERS
- ASSISTED NAVIGATION**  
 AFTER FINDING A CARPARK OR CHARGING SPOT, CLICKING ON NAVIGATE WILL BRING THE USER TO AN EXTERNAL NAVIGATION SITE FOR NAVIGATION.
- SATISFIES THE NEEDS OF EV USERS**  
 LIVE UPDATES ON CHARGING AND PARKING SPOT AVAILABILITIES, RATES AND CHARGER TYPE AROUND THE USERS USING REAL TIME DATA, MAKING IT CONVINIENT.
- FILTERING AND PREFERENCES**  
 ALLOWS USERS TO SEARCH ACCORDING TO THEIR PREFERENCES, LIKE SHELTERED PARKING, TYPE OF CHARGER AND PRICE.

# Use Case Diagram



# DATASETS AND APIs

**API:**

Housing & Development Board. (2018). Carpark Availability (2025) [Dataset]. data.gov.sg. Retrieved November 11, 2025 from  
[https://data.gov.sg/datasets/d\\_ca933a644e55d34fe21f28b8052fac63/view](https://data.gov.sg/datasets/d_ca933a644e55d34fe21f28b8052fac63/view)

Housing & Development Board. (2015). HDB Carpark Information (2025) [Dataset]. data.gov.sg. Retrieved November 11, 2025 from  
[https://data.gov.sg/datasets/d\\_23f946fa557947f93a8043bbef41dd09/view](https://data.gov.sg/datasets/d_23f946fa557947f93a8043bbef41dd09/view)

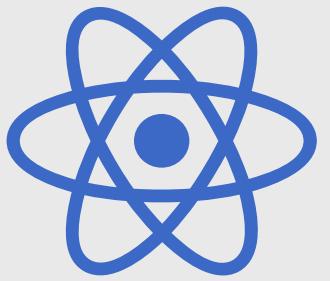
**Google Map API:**

<https://visgl.github.io/react-google-maps/>

**EV Charging point CSV:**

[https://datamall.lta.gov.sg/content/dam/datamall/datasets/Facts\\_Figures/Electric-Vehicle-Charging-Network/Electric\\_Vehicle\\_Charging\\_Points.zip](https://datamall.lta.gov.sg/content/dam/datamall/datasets/Facts_Figures/Electric-Vehicle-Charging-Network/Electric_Vehicle_Charging_Points.zip)

PRES



REACT



TYPESCRIPT



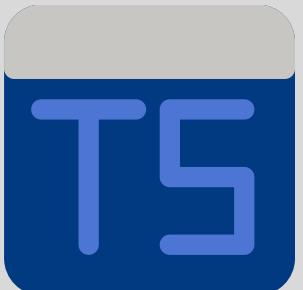
VITE

APPLICATION



NODE.JS

express

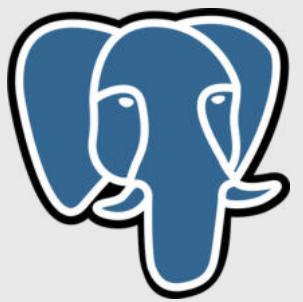


EXPRESS

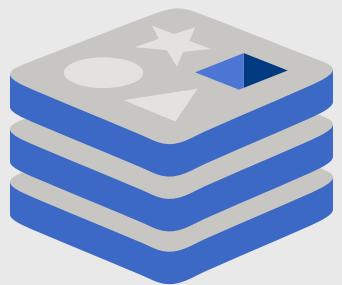
TYPESCRIPT

# Tech Stack

DATABASE



POSTGRESQL



REDIS

# System Design

# Design Principles

## SINGLE RESPONSIBILITY PRINCIPLE (SRP)

- 1.EACH DOMAIN IS ITS OWN MODULE WITH SEPARATE CONTROLLER, SERVICE, AND ADAPTER.
- 2.UI PIECES HAVE ONE JOB
- 3.CHANGES TOUCH ONLY THE EV SERVICE — NO RIPPLE TO UI OR OTHER DOMAINS

## OPEN-CLOSED PRINCIPLE (OCP)

- 1.ADAPTER + STRATEGY PATTERNS NORMALIZE EXTERNAL PROVIDERS
- 2.FILTERS/SORTING USE A PLUGGABLE RULE SET; NEW CRITERIA EXTEND VIA NEW RULE OBJECTS
- 3.FEATURE TOGGLES PLUG INTO THE FEATURE REGISTRY WITHOUT MODIFYING EXISTING MODULES

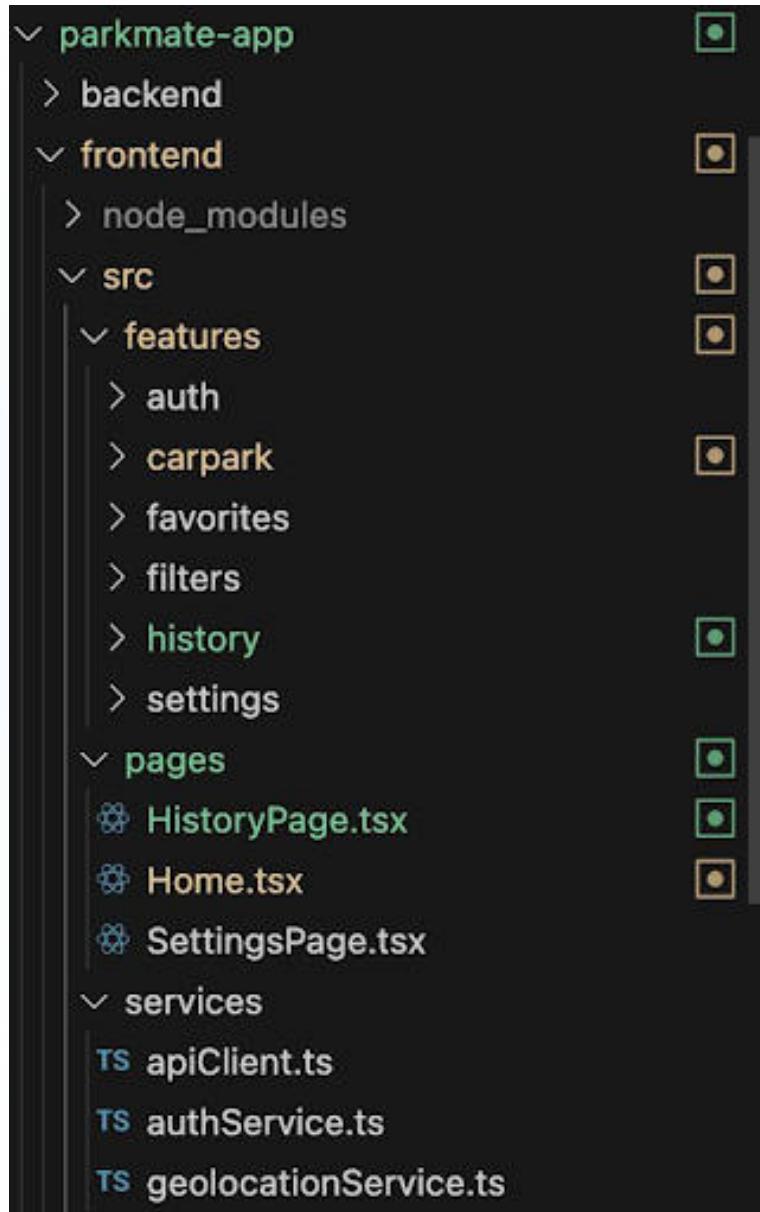
## INTERFACE SEGREGATION PRINCIPLE (ISP)

- 1.CONTROLLERS DEPEND ONLY ON WHAT THEY USE
- 2.CANONICAL DTOS ARE SMALL AND PURPOSE-BUILT
- 3.CLIENTS CALL NARROW INTERFACES—REDUCED COUPLING AND SAFER REFACTORS

## DEPENDENCY INVERSION PRINCIPLE (DIP)

- 1.CONTROLLERS DEPEND ON ABSTRACTIONS (INTERFACES) AND RECEIVE THEM VIA CONSTRUCTOR INJECTION (OR DI CONTAINER)
- 2.PROVIDERS, CACHE, AND LOGGER ARE SWAPPABLE WITHOUT CODE CHANGES
- 3.PROMOTES A LOOSELY COUPLED APP: EASIER TESTING, SAFER UPGRADES

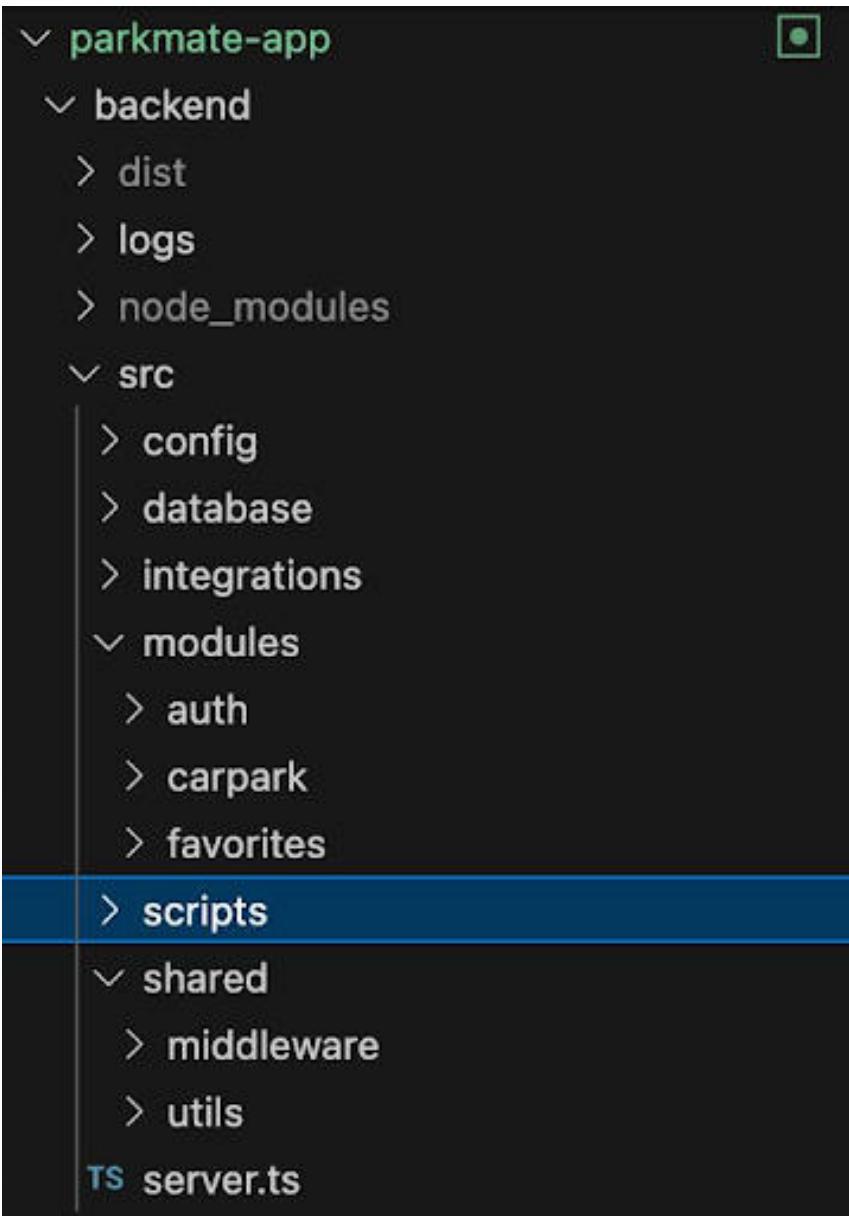
# Design Patterns



## FRONTEND

- FEATURE-BASED ORGANIZATION: EACH DOMAIN (AUTH, CARPARK, ETC.) IS SELF-CONTAINED, ALLOWING PARALLEL DEVELOPMENT.
- REUSABLE COMPONENTS: SHARED UI AND LOGIC (BUTTONS, CARDS, API CALLS) ARE ABSTRACTED UNDER ‘SERVICES’ AND ‘COMPONENTS’.
- ROUTING & PAGES SEPARATION: HIGH-LEVEL ROUTE PAGES ARE KEPT DISTINCT FROM INTERNAL FEATURE LOGIC.
- SCALABILITY: NEW FEATURES (E.G., PAYMENTS, RESERVATIONS) CAN EASILY BE ADDED UNDER ‘FEATURES’ WITH MINIMAL DISRUPTION.
- PERFORMANCE: VITE PROVIDES LIGHTNING-FAST BUILDS AND HOT MODULE RELOADING FOR IMPROVED DEVELOPER PRODUCTIVITY.

# Design Patterns

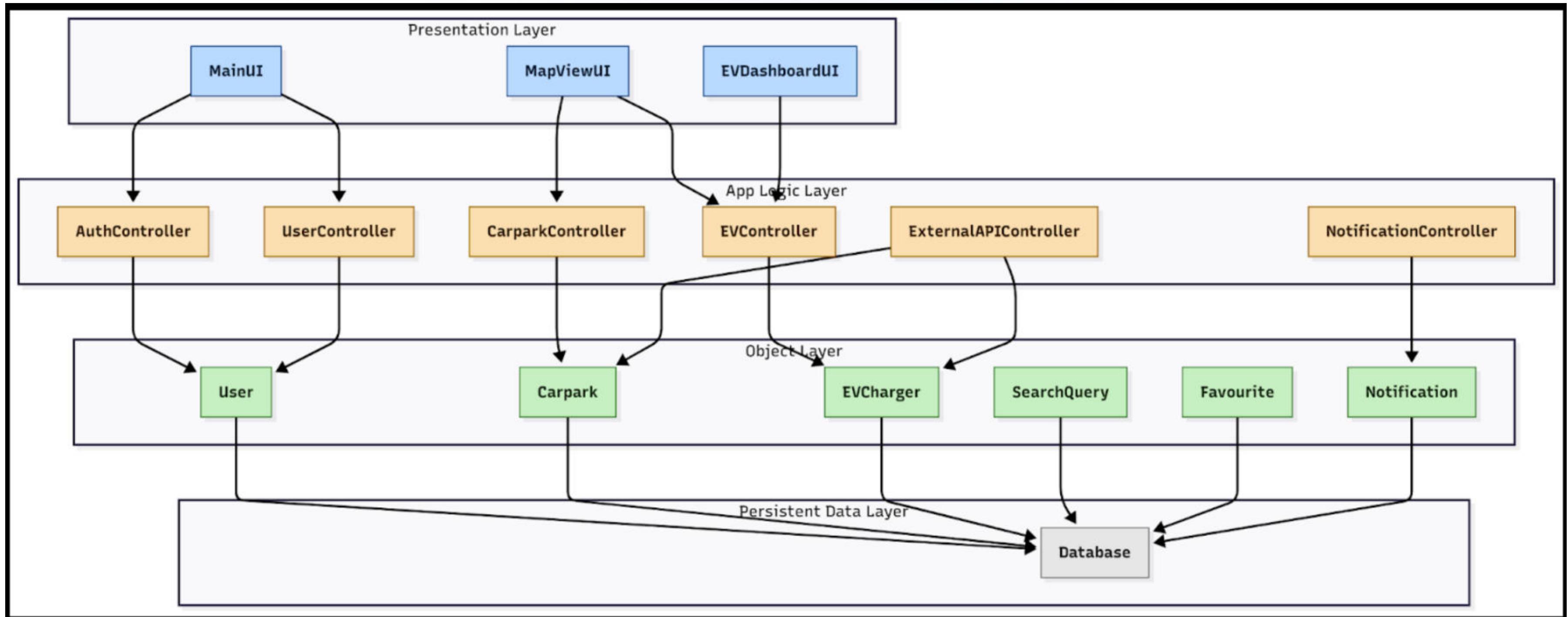


```
parkmate-app
└── backend
    ├── dist
    ├── logs
    ├── node_modules
    └── src
        ├── config
        ├── database
        ├── integrations
        └── modules
            ├── auth
            ├── carpark
            └── favorites
        └── scripts
            └── shared
                ├── middleware
                └── utils
        └── server.ts
```

## BACKEND

- MODULAR STRUCTURE: EACH CORE MODULE (AUTH, CARPARK, FAVORITES) ENCAPSULATES ROUTES, CONTROLLERS, AND SERVICE LOGIC.
- INTEGRATIONS FOLDER: HANDLES EXTERNAL DATA SOURCES LIKE LTA DATAMALL, EMA EVSE, AND ONEMAP API FOR LIVE UPDATES.
- DATABASE & CACHING: COMBINES POSTGRESQL (FOR PERSISTENT STORAGE) AND REDIS (FOR REAL-TIME EV AND CARPARK DATA).
- SHARED MIDDLEWARE: COMMON REQUEST VALIDATION, AUTHENTICATION, AND LOGGING ENSURE CONSISTENCY ACROSS ROUTES.
- SCALABLE DESIGN: ADDITIONAL MODULES (E.G., RESERVATION, PAYMENT) CAN BE SEAMLESSLY INTEGRATED FOR FUTURE IMPLEMENTATIONS.

# System Architecture



# Non-Functional Requirements

EFFICIENCY

ROBUSTNESS

MAINTAINABILITY

RELIABILITY

SECURITY

```
// Smart filtering based on zoom level to prevent performance degradation
if (bounds) {
  // Only show chargers within viewport + buffer
  const ne = bounds.getNorthEast();
  const sw = bounds.getSouthWest();

  // Add 20% buffer to bounds for smooth panning
  const latBuffer = (ne.lat() - sw.lat()) * 0.2;
  const lngBuffer = (ne.lng() - sw.lng()) * 0.2;

  filteredEvChargers = filteredEvChargers.filter(charger => {
    const lat = charger.latitude;
    const lng = charger.longitude;

    return lat >= (sw.lat() - latBuffer) &&
      lat <= (ne.lat() + latBuffer) &&
      lng >= (sw.lng() - lngBuffer) &&
      lng <= (ne.lng() + lngBuffer);
  });
}

// Additional limit based on zoom level to prevent overload
let maxMarkers = 3000;
if (currentZoom < 11) maxMarkers = 500; // Zoomed out: fewer markers
else if (currentZoom < 13) maxMarkers = 1500; // Medium zoom
// Zoomed in (13+): show up to 3000

if (filteredEvChargers.length > maxMarkers) {
  // Sample evenly distributed chargers
  const step = Math.ceil(filteredEvChargers.length / maxMarkers);
  filteredEvChargers = filteredEvChargers.filter((_, index) => index % step === 0);
}
```

DETERMINISTIC DOWNSAMPLING KEEPS MAP RENDERING  $\leq$  MAXMARKERS FOR SMOOTH PANNING (E.G., 3,000  $\rightarrow$  ~500 MARKERS).

# Non-Functional Requirements

EFFICIENCY

ROBUSTNESS

MAINTAINABILITY

RELIABILITY

SECURITY

```
// Transaction helper
export const transaction = async (callback: (client: any) => Promise<void>) => {
  const client = await pool.connect();
  try {
    await client.query('BEGIN');
    await callback(client);
    await client.query('COMMIT');
  } catch (error) {
    await client.query('ROLLBACK'); // Critical: Rollback on any error
    throw error;
  } finally {
    client.release(); // Always release connection back to pool
  }
};
```

ON ANY ERROR WE ROLLBACK, PREVENTING  
PARTIAL WRITES AND PRESERVING DATA INTEGRITY.

# Non-Functional Requirements

EFFICIENCY

ROBUSTNESS

MAINTAINABILITY

RELIABILITY

SECURITY

```
import Joi from 'joi';

export const registerSchema = Joi.object({
  email: Joi.string().email().required().messages({
    'string.email': 'Please provide a valid email address',
    'any.required': 'Email is required',
  }),
  password: Joi.string().min(8).required().messages({
    'string.min': 'Password must be at least 8 characters long',
    'any.required': 'Password is required',
  }),
  name: Joi.string().min(2).max(100).required().messages({
    'string.min': 'Name must be at least 2 characters long',
    'string.max': 'Name cannot exceed 100 characters',
    'any.required': 'Name is required',
  }),
});

export const loginSchema = Joi.object({
  email: Joi.string().email().required(),
  password: Joi.string().required(),
});

export const forgotPasswordSchema = Joi.object({
  email: Joi.string().email().required(),
});

export const resetPasswordSchema = Joi.object({
  token: Joi.string().required(),
  newPassword: Joi.string().min(8).required(),
});
```

SCHEMAS ARE EXPORTED AND REUSED ACROSS ROUTES/TESTS, SO RULE CHANGES HAPPEN IN ONE FILE—NO DUPLICATION.

# Non-Functional Requirements

EFFICIENCY

ROBUSTNESS

MAINTAINABILITY

RELIABILITY

SECURITY

```
const validateForm = (): boolean => {
  const errors: Record<string, string> = {};

  if (!formData.name.trim()) {
    errors.name = 'Name is required';
  } else if (formData.name.trim().length < 2) {
    errors.name = 'Name must be at least 2 characters';
  }

  if (!formData.email) {
    errors.email = 'Email is required';
  } else if (!/\S+@\S+\.\S+/.test(formData.email)) {
    errors.email = 'Email is invalid';
  }

  if (!formData.password) {
    errors.password = 'Password is required';
  } else if (formData.password.length < 8) {
    errors.password = 'Password must be at least 8 characters';
  } else if (!/(?=.*[a-z])(?=.*[A-Z])(?=.*\d)/.test(formData.password)) {
    errors.password = 'Password must contain uppercase, lowercase, and number';
  }

  if (!formData.confirmPassword) {
    errors.confirmPassword = 'Please confirm your password';
  } else if (formData.password !== formData.confirmPassword) {
    errors.confirmPassword = 'Passwords do not match';
  }

  setValidationErrors(errors);
  return Object.keys(errors).length === 0;
};
```

SINGLE GATE: FORM ONLY SUBMITS  
WHEN ALL VALIDATIONS PASS (NO  
PARTIAL/DIRTY DATA).

# Non-Functional Requirements

EFFICIENCY

ROBUSTNESS

MAINTAINABILITY

RELIABILITY

SECURITY

```
async login(credentials: LoginDto): Promise<AuthResponse> {
    // Find user
    const user = await this.authRepository.findByEmail(credentials.email);
    if (!user) {
        throw new Error('Invalid credentials'); // Generic error (security)
    }

    // Verify password
    const isPasswordValid = await bcrypt.compare(credentials.password, user.passwordHash)
    if (!isPasswordValid) {
        throw new Error('Invalid credentials'); // Generic error (security)
    }

    // Generate tokens
    const accessToken = this.generateAccessToken(user.id);
    const refreshToken = this.generateRefreshToken(user.id);

    // Remove password hash from response
    const { passwordHash: _passwordHash, ...userWithoutPassword } = user;
    return {
        user: userWithoutPassword,
        accessToken,
        refreshToken,
    };
}
```

SENSITIVE FIELDS ARE REMOVED SERVER-SIDE BEFORE  
SERIALIZATION, PREVENTING ACCIDENTAL LEAKAGE.

# Functional Requirement Traceability

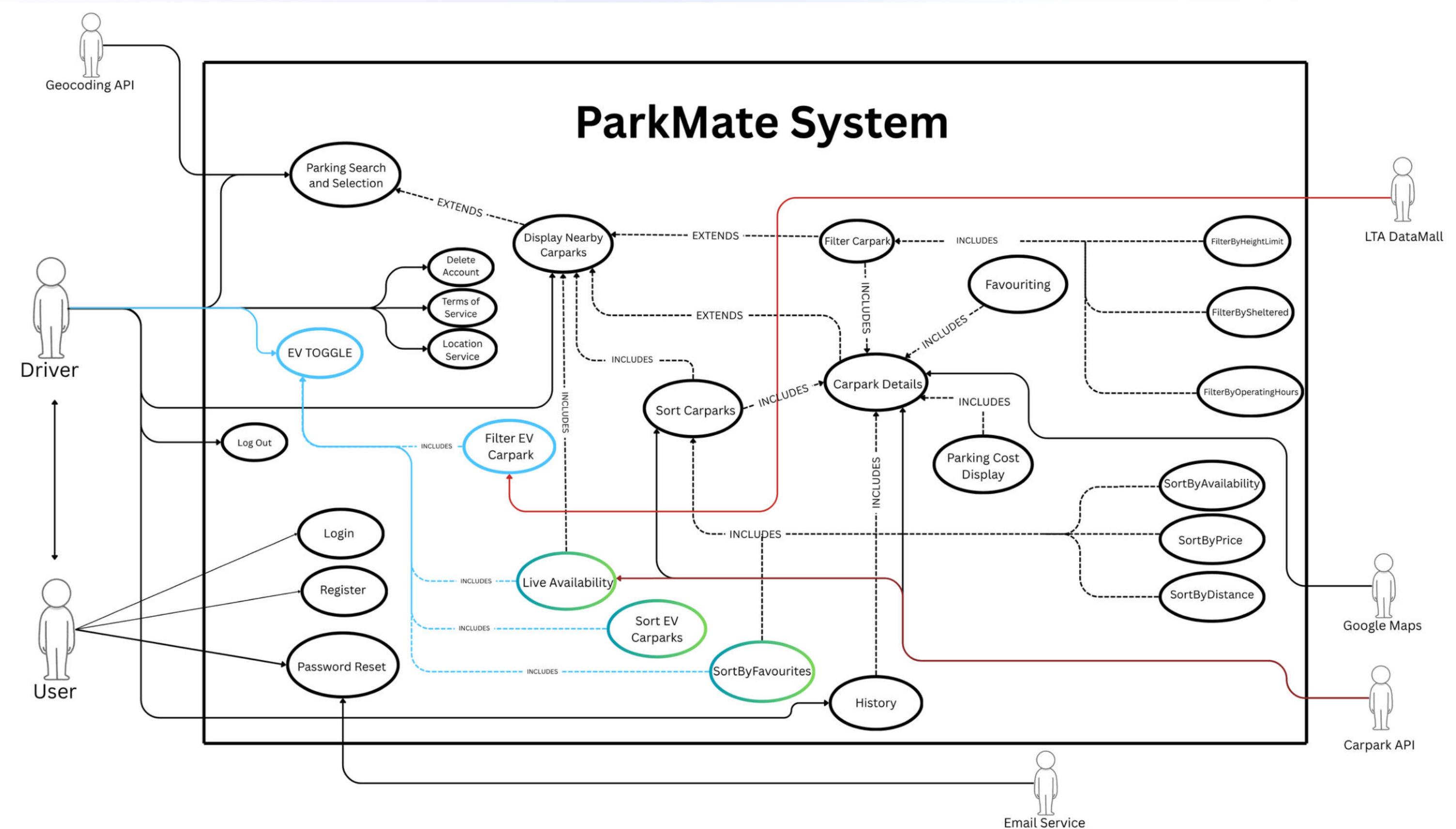
# Functional Requirement #6

Actor:	User (EV Driver), ParkMate System
Description:	This use case lets EV users filter and sort carparks or charging stations based on their vehicle's plug type and preferred charging speed (Type 2, CCS 2 (Fast), CHAdeMO, CCS2, CC2). It helps users quickly find chargers that are compatible with their car and reduce waiting time by showing only stations that match their needs.
Preconditions:	<ol style="list-style-type: none"> <li>1. <u>User</u> has opened the Search Results or Map View after entering a destination.</li> <li>2. Connector type and power information are included in the dataset</li> </ol>
Postconditions:	Map updates to show only stations that match the selected connector type and speed.
Priority:	Medium (Enhances user personalization and usability for EV drivers)
Frequency of Use:	Medium (Users are likely to apply filters during every search session)
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user opens the Filter menu on the search or map screen.</li> <li>2. ParkMate shows a section labeled "EV Charger Filters."</li> <li>3. The user selects a preferred charging speed</li> <li>4. The user taps Apply Filters.</li> <li>5. ParkMate updates the map and list instantly, showing only chargers that meet the criteria.</li> <li>6. If no stations match, the system displays a message suggesting to clear or adjust filters.</li> </ol>
Alternative Flows:	<ol style="list-style-type: none"> <li>1. To clear filters, <u>The</u> user taps "Reset All." and ParkMate resets the results to show all carparks and EV stations again.</li> <li>2. When live data refreshes, ParkMate keeps the active filters and updates only the visible chargers that match them.</li> </ol>
Exceptions:	If a refresh fails while filters are active, cached data remains visible until the next update.
Includes:	NIL
Special Requirements:	<ul style="list-style-type: none"> <li>• Results update immediately when filters are applied, no need to reload the whole page.</li> </ul>

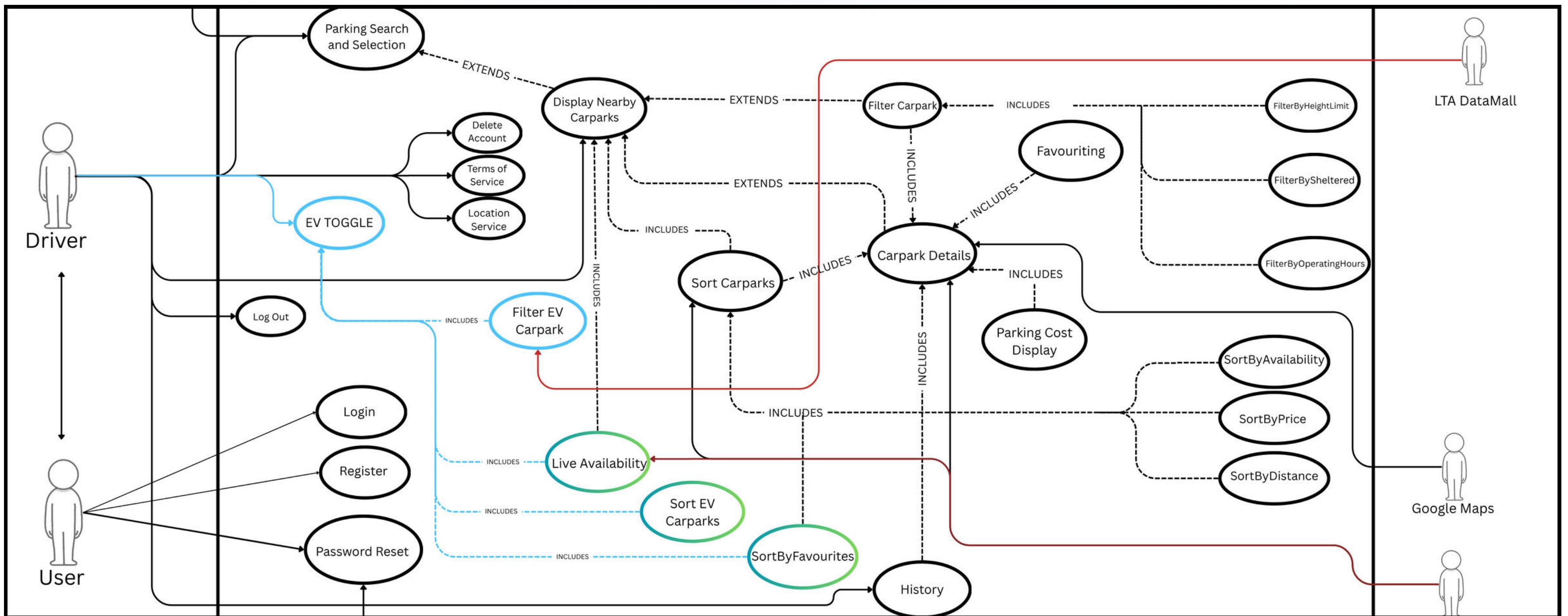
	<ul style="list-style-type: none"> <li>• ParkMate remembers the user's last chosen connector and speed preferences for the next session.</li> <li>• Users can select multiple connector types and charging speeds simultaneously.</li> <li>• Each filter option includes both icons and text for accessibility.</li> <li>• Filters rely on standardized connector names from LTA or operator datasets.</li> </ul>
Assumptions:	<ul style="list-style-type: none"> <li>• The app has access to structured metadata for each EV station (connector type, speed).</li> <li>• At least one filterable EV attribute (type or speed) is available for every listed station.</li> </ul>
Notes and Issues:	NIL

# Functional Requirement #6

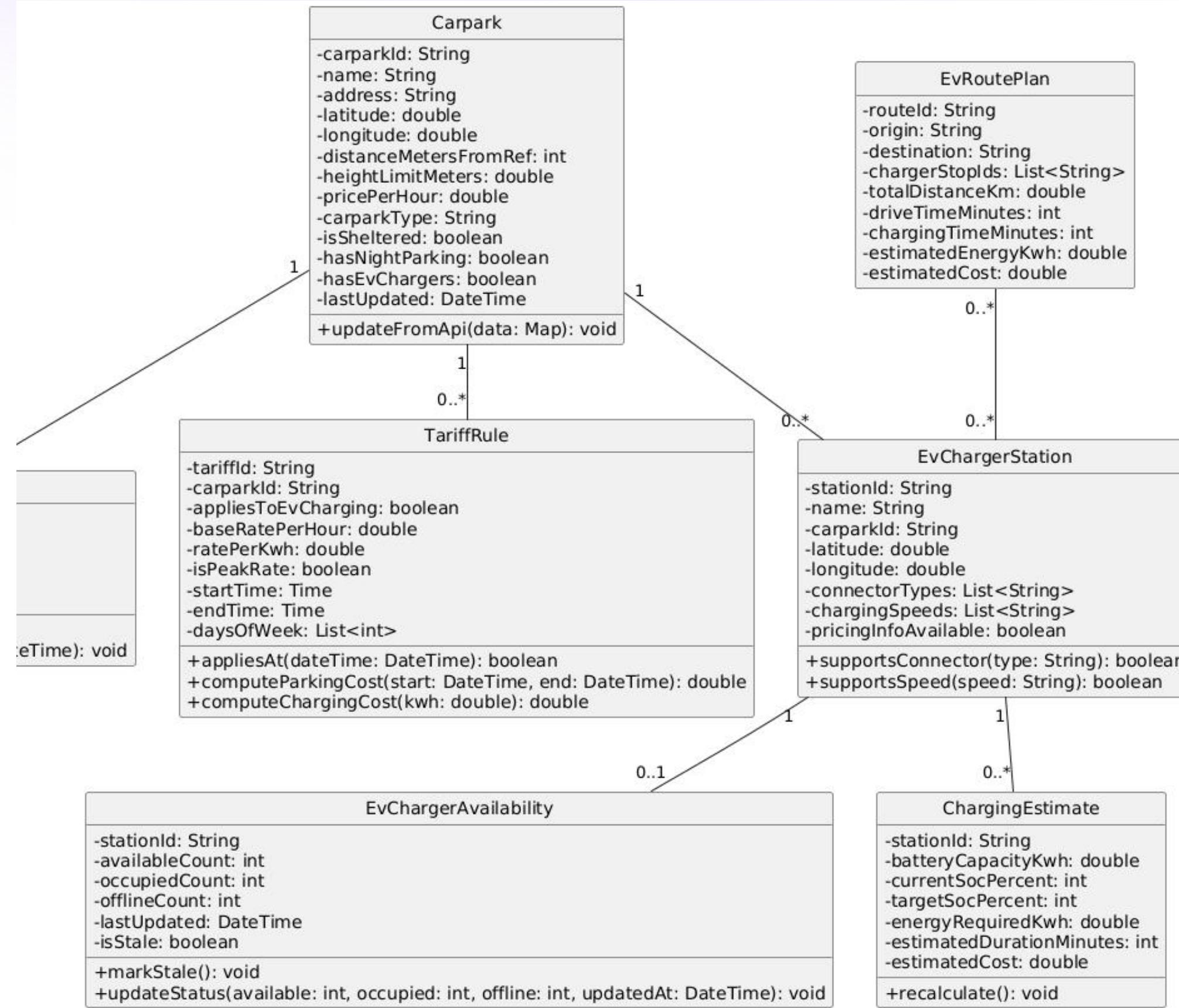
# USE CASE DIAGRAM



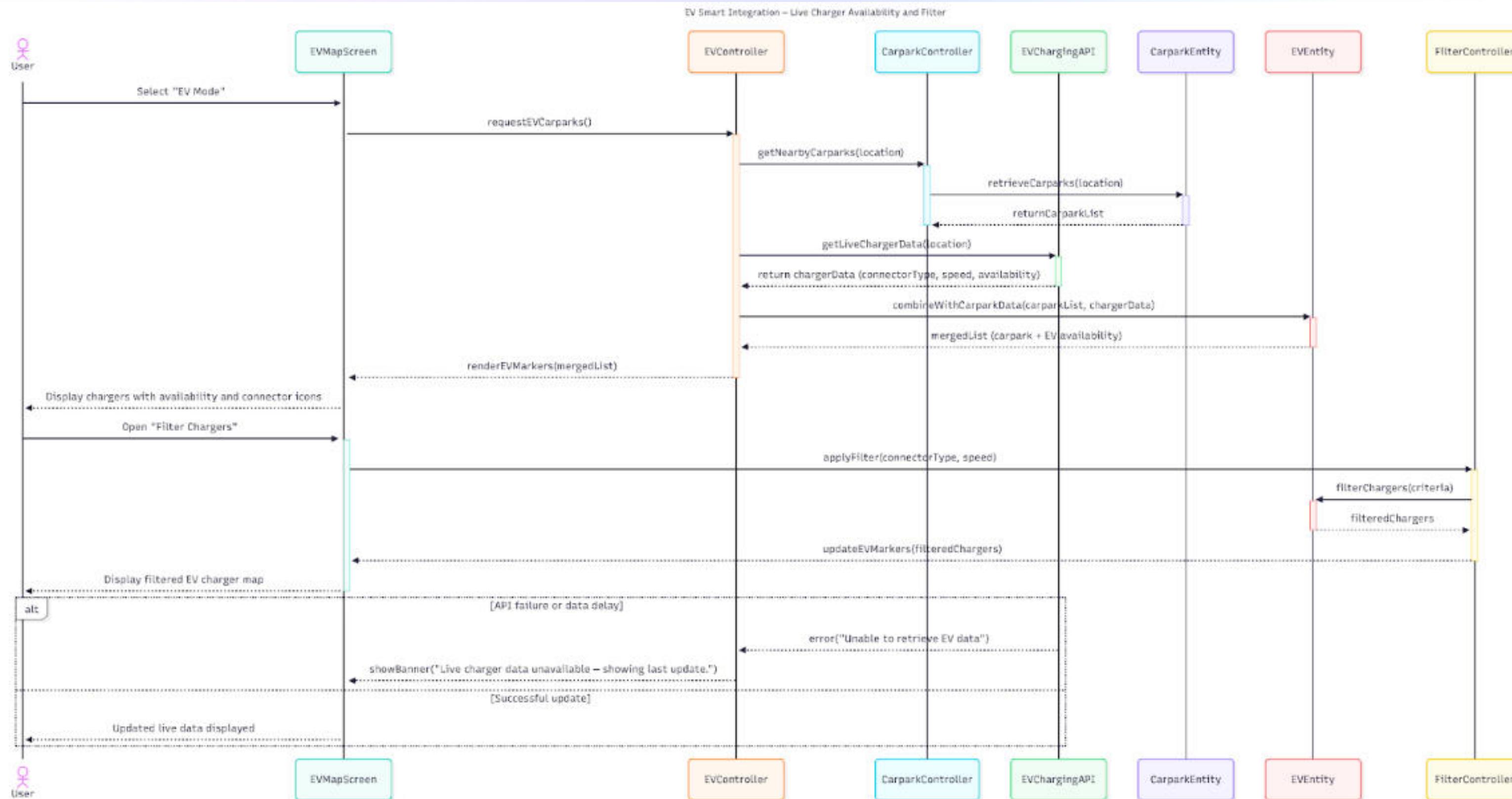
# Traceability



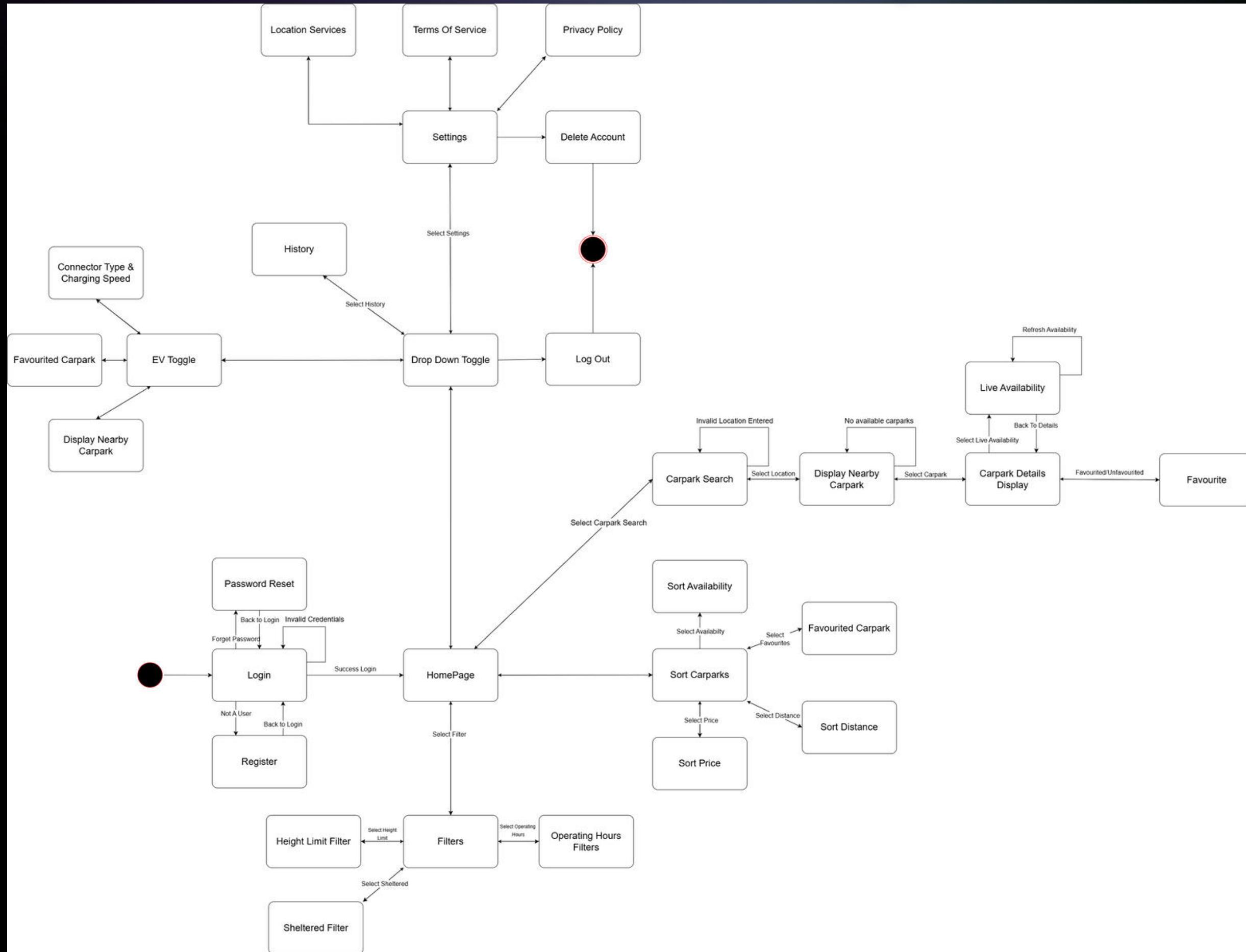
# REQUIREMENTS ANALYSIS (CLASS DIAGRAM)



# REQUIREMENTS ANALYSIS (SEQUENCE DIAGRAMS)



# REQUIREMENTS ANALYSIS (DIALOG MAP)



# BLACK BOX TESTING

# TESTING STAGES

USER  
AUTHENTICATION  
TESTING

DESTINATION  
TESTING

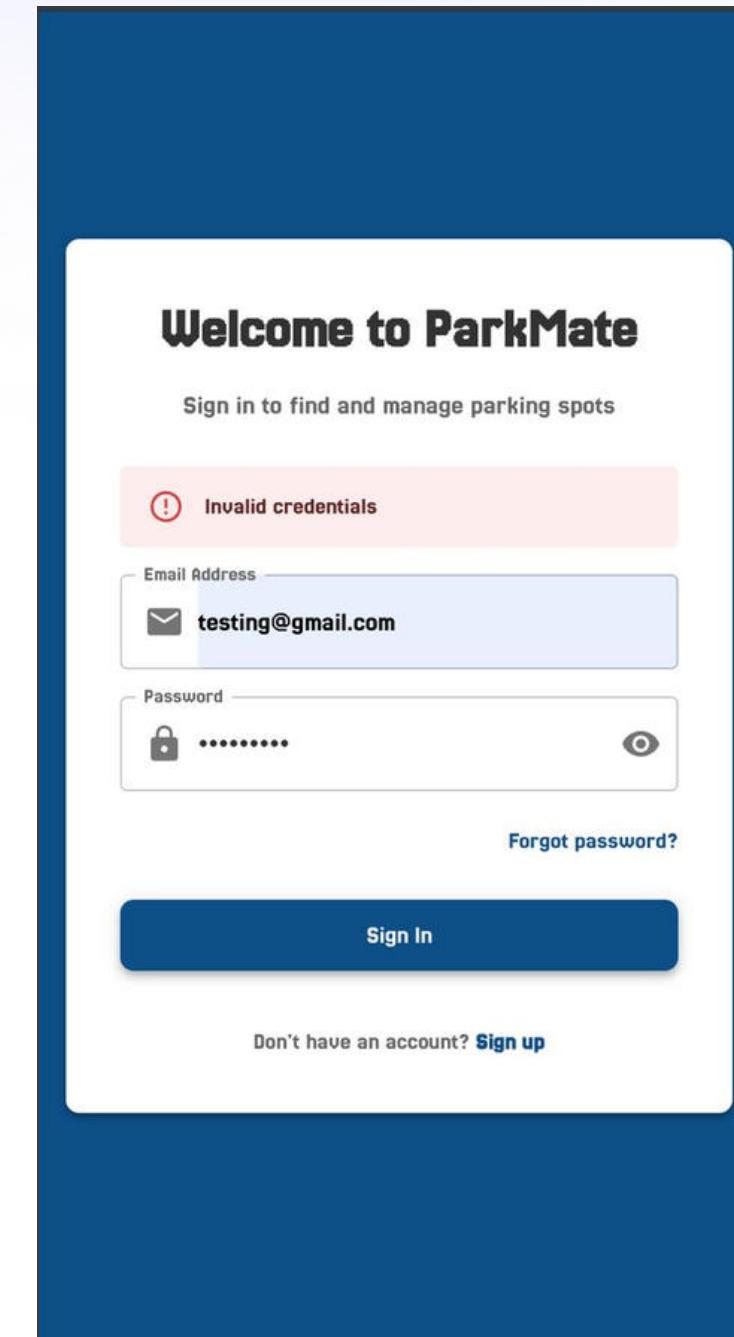
CARPARK  
DISPLAY  
TESTING

EV DISPLAY  
TESTING

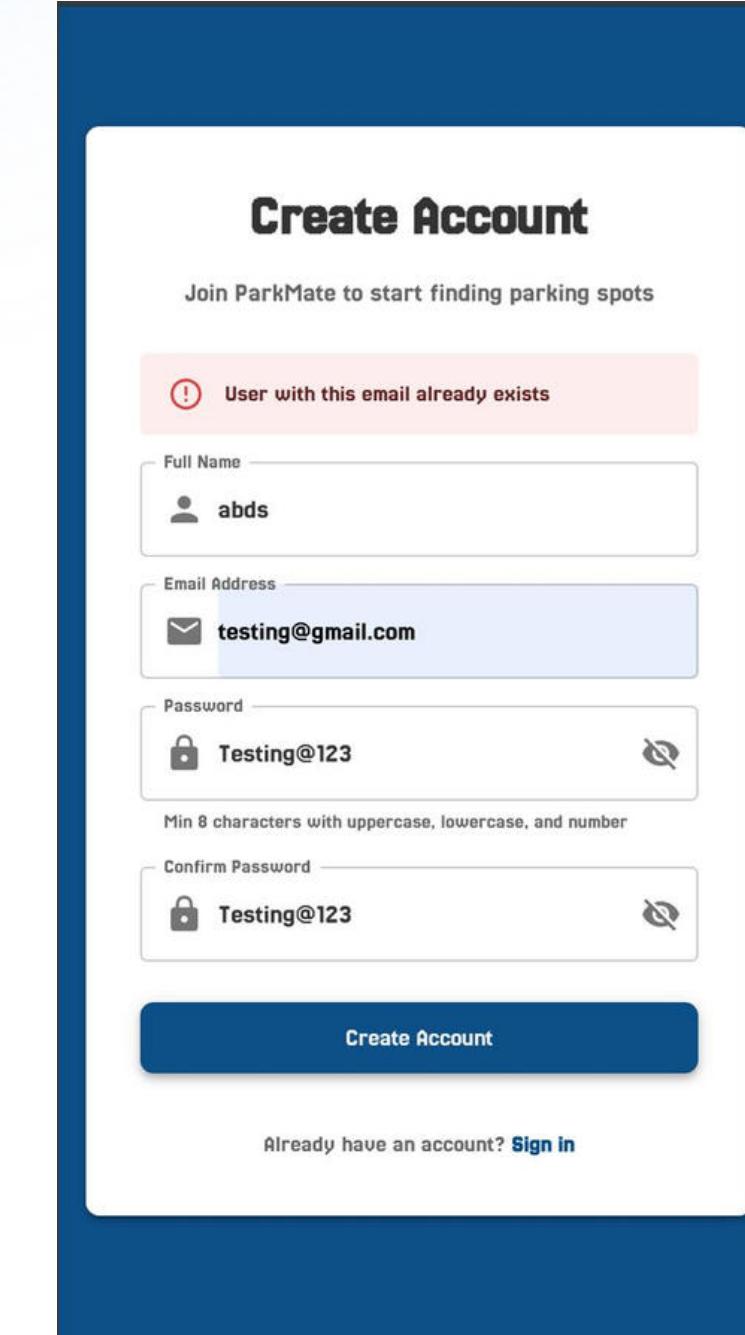
EV CHARGER  
TESTING

# Testing (User Authentication)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	(All Valid Inputs) Name: "Zerius Heng" Email: <a href="mailto:zerius0139@gmail.com">zerius0139@gmail.com</a> Password: ParkMate!2025 Confirm Password: ParkMate!2025	Successful signup; userID created, account can log in thereafter	Successful signup; userID created, account can log in thereafter	Yes
2.	(All valid except Name) Name: ""	System notifies "Name is required."	System notifies "Name is required."	Yes
3.	(All Valid except Email) Email: ""	System notifies "Please enter a valid email."	System notifies "Please enter a valid email."	Yes
4.	(All Valid except Password) Password: ""	System notifies "Please fill in all required fields."	System notifies "Please fill in all required fields."	Yes
5.	(Weak Password) Password: password Confirm: password	System notifies "Password does not meet the requirements."	System notifies "Password does not meet the requirements."	Yes
6.	(Mismatch Password) Password: ParkMate!2025 Confirm: ParkMate!2024	System notifies "Passwords do not match."	System notifies "Passwords do not match."	Yes
7.	(Duplicate Email) Email already registered	System notifies "Email already in use."	System notifies "Email already in use."	Yes
8.	(Valid Inputs) with transient DB error	System notifies "Service unavailable. Please try again later."	System notifies "Service unavailable. Please try again later."	Yes



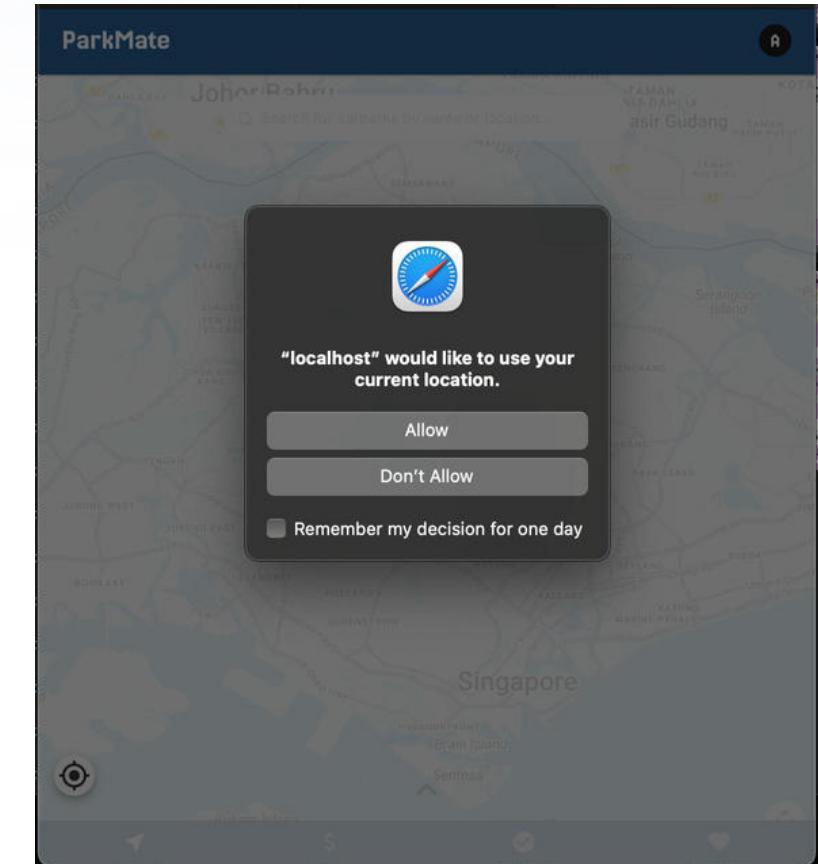
TEST CASE 4



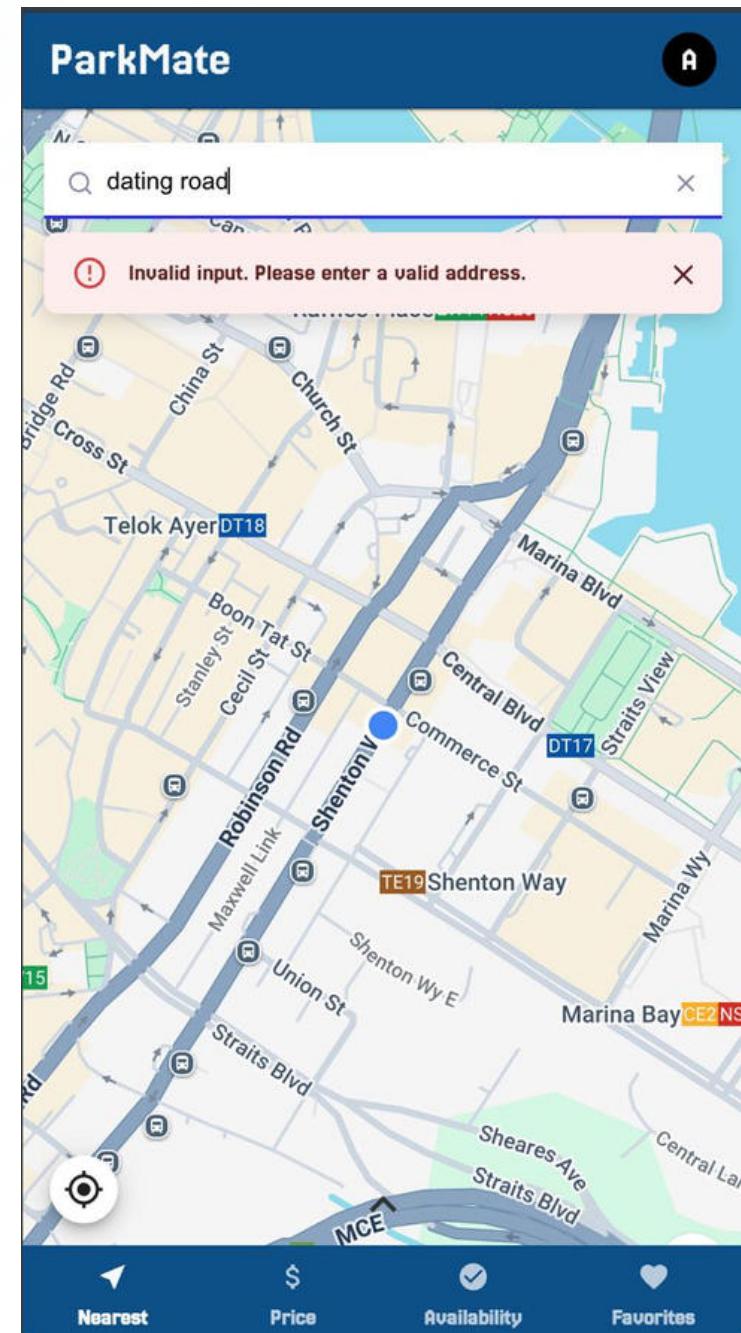
TEST CASE 7

# Testing (Destination)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	(Valid) Query: "NTU North Spine"	Geocoding suggestions shown; user selects a suggestion; destination set	Geocoding suggestions shown; user selects a suggestion; destination set	Yes
2.	(Valid) Use "Current Location" (permission granted)	Destination set to current coordinates	Destination set to current coordinates	Yes
3.	(Invalid) Query: "#@\$%"	System notifies "No matching locations found."	System notifies "No matching locations found."	Yes
4.	(Service Down) Geocoding unavailable	System notifies "Unable to retrieve location data. Please try again later."	System notifies "Unable to retrieve location data. Please try again later."	Yes
5.	(Permission Denied) Current Location denied; empty query	System prompts to enter a destination text	System prompts to enter a destination text	Yes



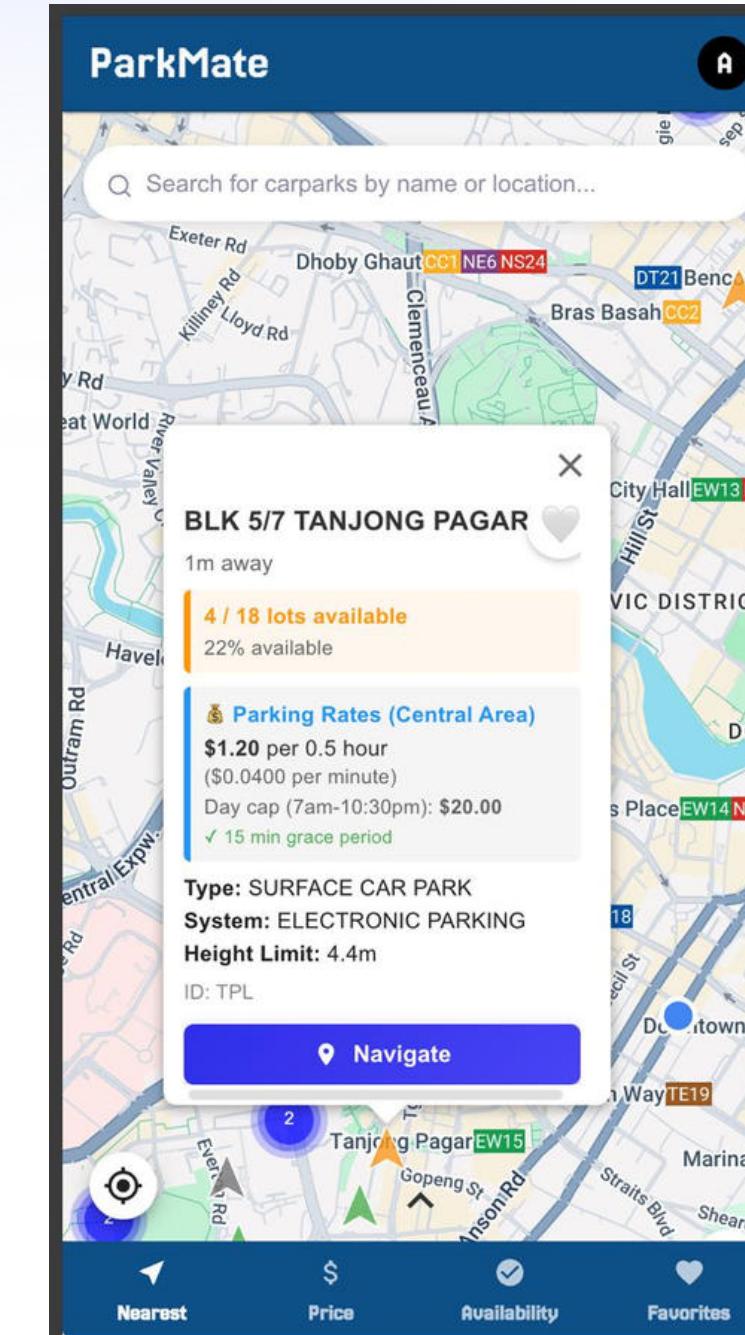
TEST CASE 2



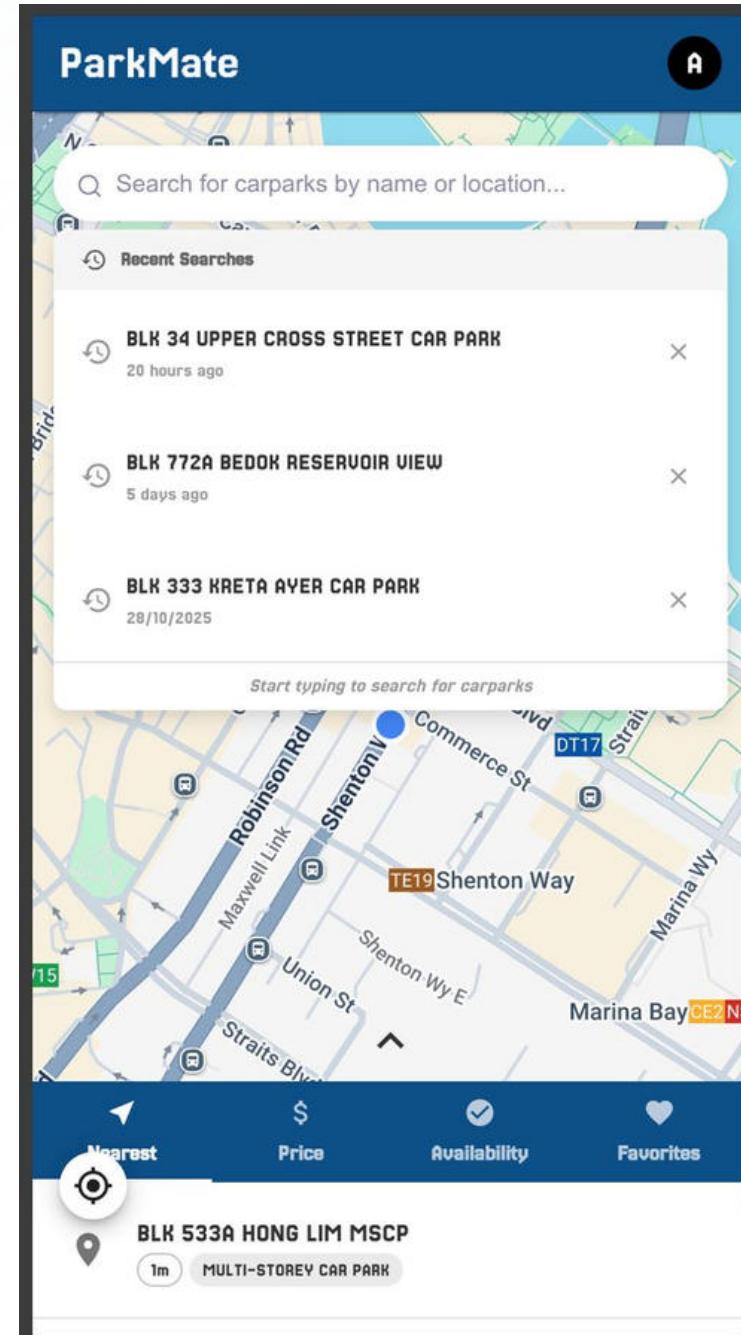
TEST CASE 3

# Testing (Carpark Display)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	(Valid) Destination set to "NTU North Spine"	Nearby carparks displayed with name, distance, price, height limit, live availability, last updated.	Nearby carparks displayed with name, distance, price, height limit, live availability, last updated.	Yes
2.	(Zero Results) Area with no carparks	System shows "No nearby carparks found" with the option to adjust the destination	System shows "No nearby carparks found" with the option to adjust the destination	Yes
3.	(Data Stale) Provider returns cached data	Results shown with "Last updated..." freshness indicator; no crash	Results shown with "Last updated..." freshness indicator; no crash	Yes
4.	(Select Item) Tap marker/list item	Carpark details open and saved to history	Carpark details open and saved to history	Yes
5.	(Service Down) Carpark data API unavailable	System notifies "Carpark information unavailable. Please try again later."	System notifies "Carpark information unavailable. Please try again later."	Yes



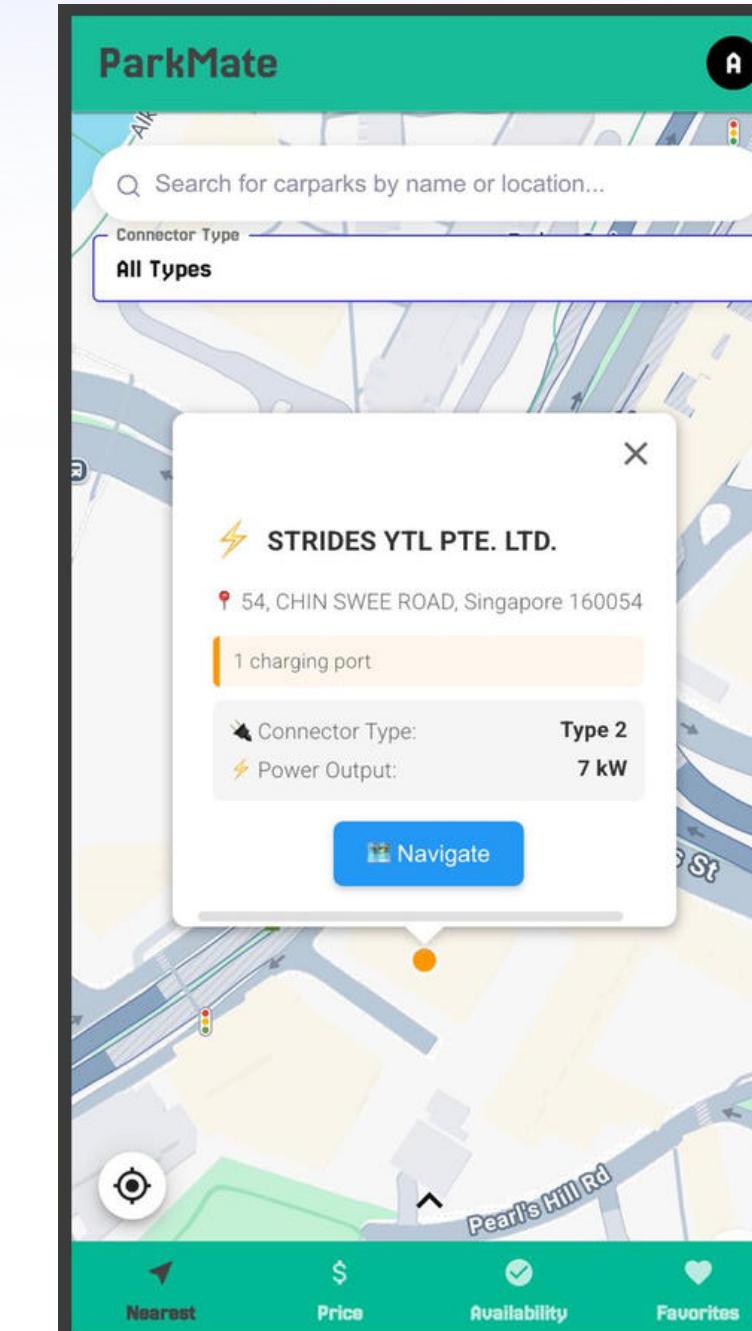
TEST CASE 1



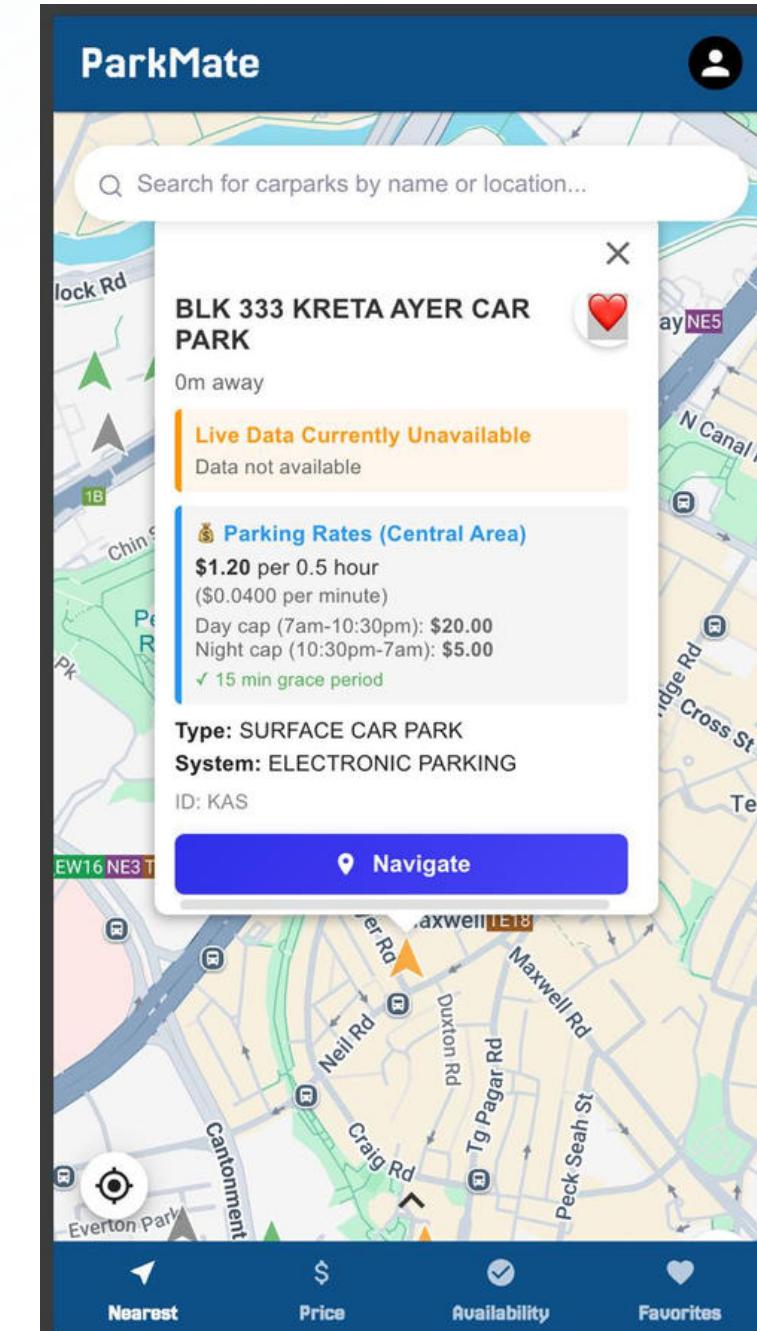
TEST CASE 4

# Testing (EV Display)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	(Valid) Destination with chargers	Show stations with availability state, connector type, charging speed, and last updated	Show stations with availability state, connector type, charging speed, and last updated	Yes
2.	(Auto Refresh) Wait 1 minute	Data refreshes; last updated time changes accordingly	Data refreshes; last updated time changes accordingly	Yes
3.	(No Chargers) Area without chargers	The system shows "No EV chargers found nearby" guidance	The system shows "No EV chargers found nearby" guidance	Yes
4.	(Feed Down) EV data source offline	Show last known results with a clear notice "Live data temporarily unavailable."	Show last known results with a clear notice "Live data temporarily unavailable."	Yes
5.	(Manual Refresh) Pull to refresh immediately after auto refresh	No errors; consistent latest timestamp	No errors; consistent latest timestamp	Yes



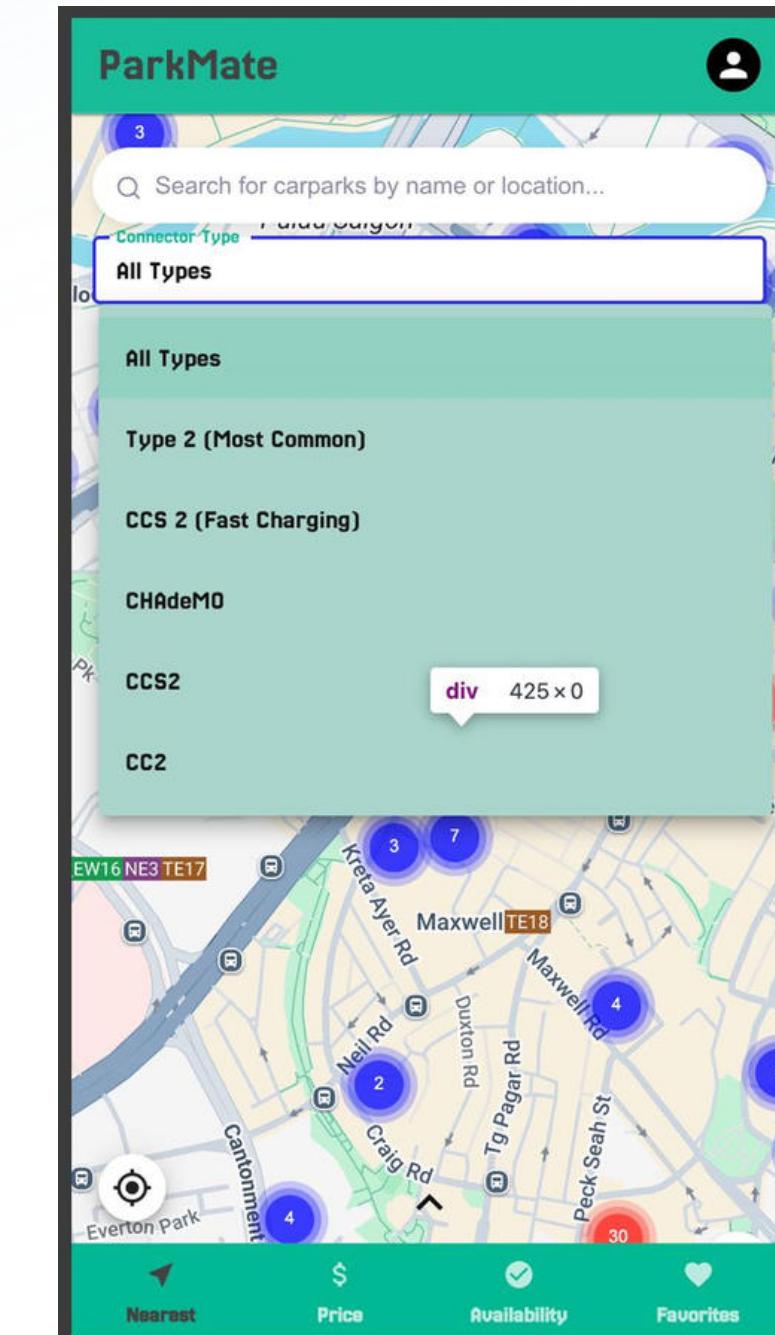
TEST CASE 1



TEST CASE 3

# Testing (EV Filters)

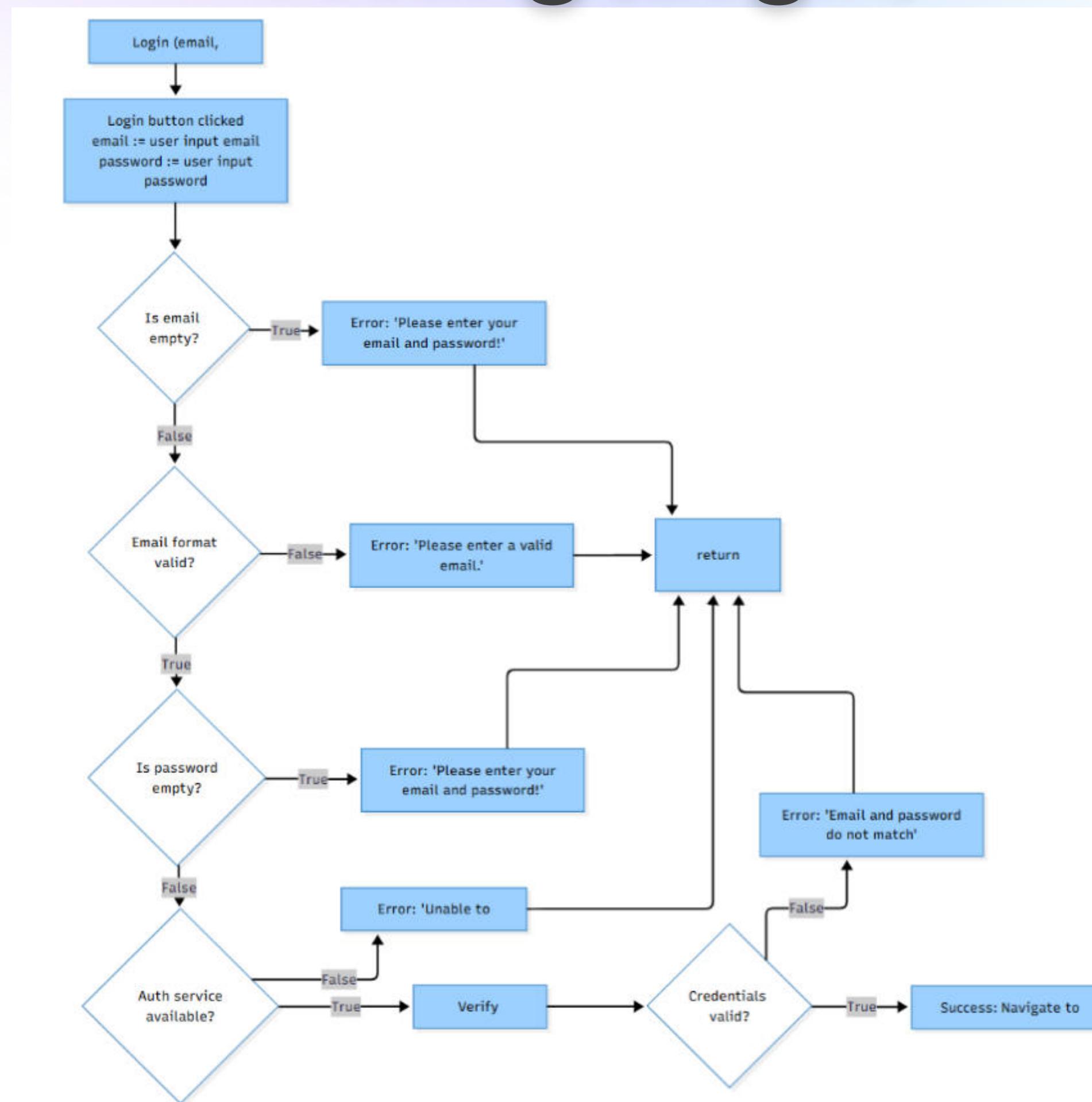
No.	Test Input	Expected Output	Actual Output	Pass?
1.	(Valid) Connector Types: {Type 2, CCS2} Charging Speed: "Fast"	List/map shows only matching stations; last update retained	List/map shows only matching stations; last update retained	Yes
2.	(Valid) Require Available: true	Results show only stations with at least one available connector	Results show only stations with at least one available connector	Yes
3.	(Contradictory) Filters yield empty set	Empty state with guidance to clear or adjust filters	Empty state with guidance to clear or adjust filters	Yes
4.	(Preference Memory) Apply filter, navigate away, return	Prior EV filter preferences are remembered and applied	Prior EV filter preferences are remembered and applied	Yes



TEST CASE 1

# WHITE BOX TESTING

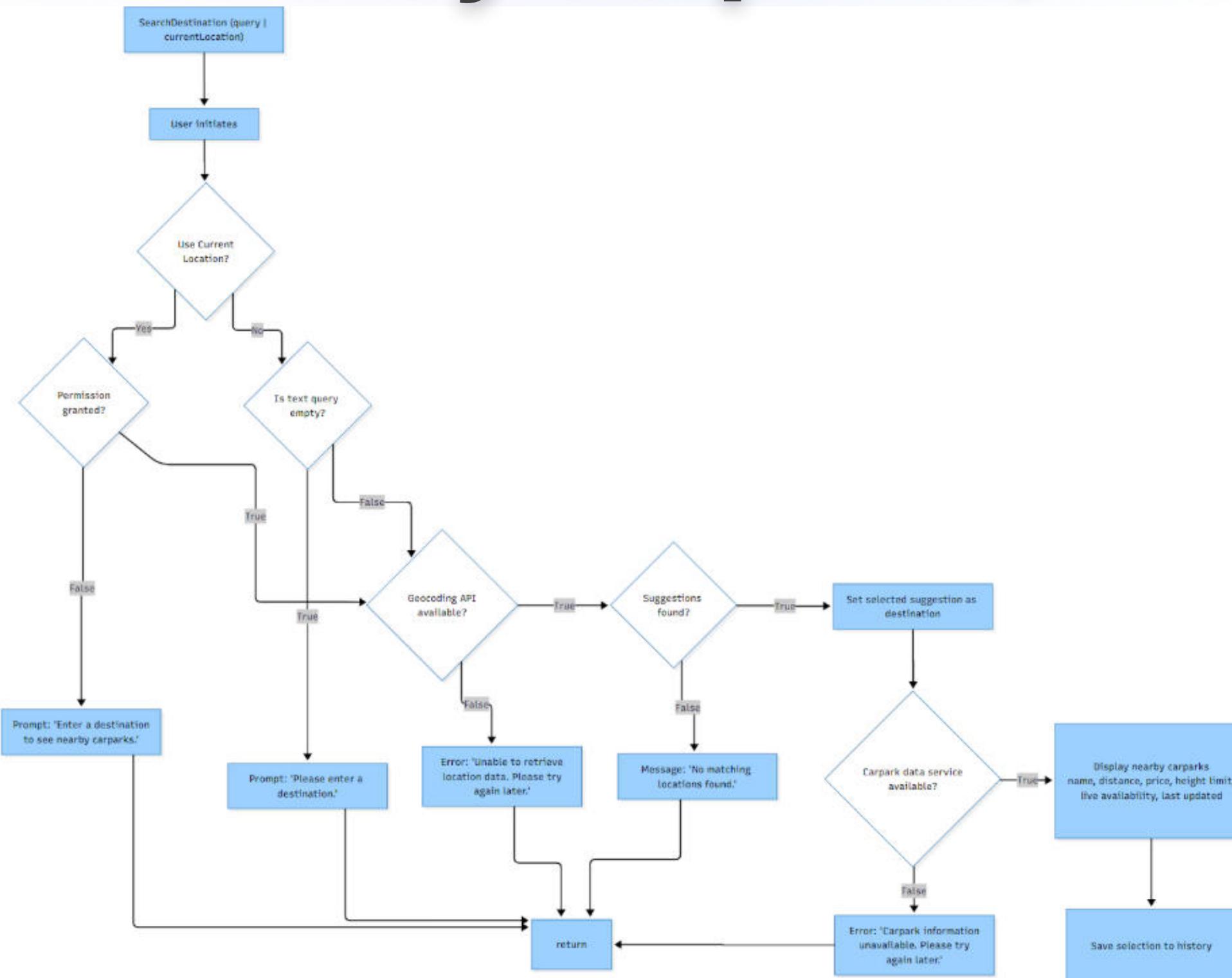
# Testing (Login)



# Testing (Login)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	email = <a href="mailto:user1@parkmate.app">user1@parkmate.app</a> password = Correct#2025	Log in successfully → navigate to Home	Log in successfully → navigate to Home	Y
2.	email = "" password = Correct#2025	"Please enter your email and password!"	"Please enter your email and password!"	Y
3.	email = admin1 password = Correct#2025	"Please enter a valid email."	"Please enter a valid email."	Y
4.	email = admin1@ password = Correct#2025	"Please enter a valid email."	"Please enter a valid email."	Y
5.	email = <a href="mailto:user1@parkmate.app">user1@parkmate.app</a> password = ""	"Please enter your email and password!"	"Please enter your email and password!"	Y
6.	email = <a href="mailto:ghost@parkmate.app">ghost@parkmate.app</a> password = Any#1234	"Email and password do not match"	"Email and password do not match"	Y
7.	email = <a href="mailto:user1@parkmate.app">user1@parkmate.app</a> password = Correct#2025 <b>Auth service down</b>	"Unable to connect"	"Unable to connect"	Y

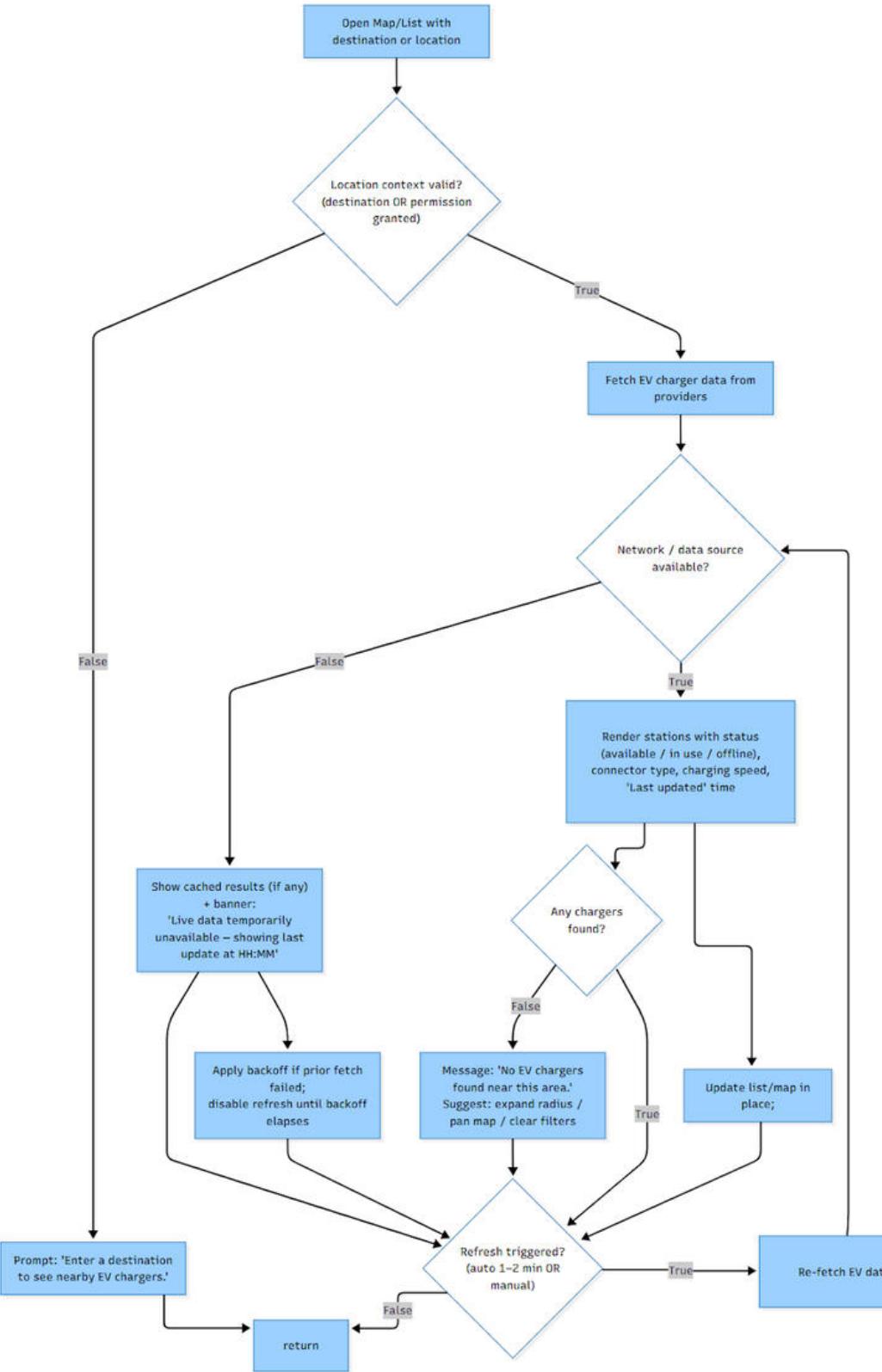
# Testing (Destination Search & Nearby Carparks)



# Testing (Destination Search & Nearby Carparks)

No.	Test Input	Expected Output	Actual Output	Pass?
1.	Query mode: "NTU North Spine"	Suggestions shown → user selects → destination set → nearby carparks displayed with name, distance, price, height limit, live availability, last updated; selection saved to history	Suggestions shown → user selects → destination set → nearby carparks displayed with name, distance, price, height limit, live availability, last updated; selection saved to history	Y
2.	Query mode: ""	Prompt "Please enter a destination"	Prompt "Please enter a destination"	Y
3.	Query mode: "@@@@###"	"No matching locations found"	"No matching locations found"	Y
4.	Query mode: "NTU North Spine", Geocoding down	"Unable to retrieve location data. Please try again later."	"Unable to retrieve location data. Please try again later."	Y
5.	Current Location mode (permission granted)	Destination set to device location → nearby carparks displayed with full details	Destination set to device location → nearby carparks displayed with full details	Y
6.	Current Location mode (permission denied), no query	Prompt "Enter a destination to see nearby carparks."	Prompt "Enter a destination to see nearby carparks."	Y
7.	Destination set, Carpark data service unavailable	"Carpark information unavailable. Please try again later."	"Carpark information unavailable. Please try again later."	Y

# Testing (EV Live Charger Availability)



No.	Test Input	Expected Output	Actual Output	Pass?
1.	Valid Destination; network/data source available; area has chargers	Stations rendered with status (available / in use / offline), connector type, charging speed, and Last updated timestamp	Stations rendered with status (available / in use / offline), connector type, charging speed, and Last updated timestamp	Y
2.	No destination and location permission denied	Prompt "Enter a destination to see nearby EV chargers"	Prompt "Enter a destination to see nearby EV chargers"	Y
3.	Valid destination; data source down	Show cached results (if any) with banner "Live data temporarily unavailable — showing last update at HH:MM."	Show cached results (if any) with banner "Live data temporarily unavailable — showing last update at HH:MM."	Y
4.	Valid Destination; no chargers in area	Message "No EV chargers found near this area." with guidance to expand radius, pan map, or clear filters	Message "No EV chargers found near this area." with guidance to expand radius, pan map, or clear filters	Y
5.	Auto refresh occurs ~1-2 minutes after case #1	Data is re-fetched; Last updated time advances; list/map updates in place without losing scroll or selection	Data is re-fetched; Last updated time advances; list/map updates in place without losing scroll or selection	Y



Live Demo

# THE WAY AHEAD

1

## IMPROVEMENT 1

INTEGRATE THE NAVIGATION INTO THE APP ITSELF FOR THE USER TO HAVE A MORE SEAMLESS EXPERIENCE.

2

## IMPROVEMENT 2

INCORPORATE AI INTO HELPING THE USERS COMPARE THE OPTIONS AROUND THEM, LISTING OUT THE PROS AND CONS OF EACH CHARGER

3

## IMPROVEMENT 3

FURTHER EXPANSION OF OUR APP TO BE USED IN CARPLAY FOR CONVENIENCE, AND INCREASING SAFETY SINCE THE USERS WILL NOT HAVE TO USE THEIR PHONES

# Thank You

---