1.Scenario: Planning a Route

Starting Point

- **User is authenticated** and lands on the main screen after logging in.

User Profile

- **Regular Commuter**: This user frequently travels between set locations, like home and work, and prioritizes eco-friendly, efficient options. The application can accommodate wheelchair accessibility or limited mobility needs as well.

Task Description

- **Objective**: Plan a cross-city route using multiple transport options (bike-sharing, bus, subway), while factoring in real-time weather and accessibility.
- **Motivation**: Find a cost-effective, quick route that also minimizes environmental impact.

Context

- This is typically done during rush hours, making route speed and efficiency crucial as well as adaptability to traffic and transport changes.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user begins on the **main screen** after logging in, where the **"Plan a Route"** button is visible.
2. **UI Elements**:
    - The user taps **"Plan a Route"**.
    - They enter **starting and destination addresses** in designated text fields.
    - Options for **transportation modes** (bike-sharing, bus, subway, etc.) appear as icons for selection.
    - A **weather icon** provides real-time updates, affecting the system's choice of transport options.
    - For accessibility needs, the user can toggle an **"Accessibility Mode"** icon that highlights wheelchair-friendly routes.
3. **System Reactions**:
    - Upon confirming selections, the application calculates the optimal route, displaying the travel time and cost for each segment (e.g., bike-sharing to subway).
    - The map view updates to display the route, using colored lines for different modes (e.g., green for bike, blue for bus).
    - Accessible paths are marked with a distinct color or icon if **Accessibility Mode** is on.
4. **Success / Error Cases**:

    - **Success**: The user receives the full route with time and cost estimates, adjusting for weather if needed.
    - **Error**: If a transport option is unavailable (e.g., no bikes nearby due to high demand), the app alerts the user and suggests alternative routes.
    - **Other Errors**: The app displays an error message if there's an issue with the user's location input, prompting them to re-enter it.
5. **Design Motivation**:
    - This design prioritizes **user efficiency** by reducing the steps needed to plan a route and provides clear, real-time information on weather and transport availability. Icons for modes of transport and accessibility improve **usability** and **accessibility** for various user profiles.
    - Leveraging principles of **responsive design** and **contextual adaptation** from mobile navigation systems, this app mirrors successful practices seen in city mobility applications.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user selects only public transport options due to rainy weather, and the system suggests a bus-to-subway route, avoiding outdoor modes.
    - **Alternative 2**: For a sunny day, the app highlights bike-sharing as the first leg of the commute, with the user switching to a subway midway

2. Scenario: Reserving an Electric Bike from a Nearby Station

Starting Point

- **User has opened the app**, enabled location services, and can see nearby transport options, including bike-sharing stations.

User Profile

- **Occasional Urban Cyclist**: This user typically prefers short, eco-friendly trips around the city and uses electric bikes when available, especially during busy times when public transport is inconvenient or delayed.

Task Description

- **Objective**: Find and reserve a nearby electric bike from a bike-sharing station for a quick, eco-friendly trip.
- **Motivation**: The user needs a convenient and sustainable transportation method to navigate the city when other options are less efficient.

Context

- This task is commonly done in daytime hours, during periods of high traffic or in densely populated areas where bike availability may fluctuate, necessitating a reservation system.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user opens the app to the **main map view**, showing current location and nearby transport options.
    - The app automatically detects location via **location services** and displays **nearby bike-sharing stations**.
2. **UI Elements**:
    - The user taps on an **electric bike icon** at a nearby station on the map, which reveals bike availability and station details.
    - A **"Reserve Bike"** button appears in the station's pop-up window, along with **estimated time to the station**and **battery level** of each available bike.
    - The user selects a bike with sufficient battery and taps **"Reserve"** to confirm.
3. **System Reactions**:
    - The app processes the reservation and locks the selected bike, updating the station's availability and marking the bike as "Reserved" in the system.
    - A **confirmation message** appears with reservation details, including the bike number, reservation time limit, and walking directions to the station.
    - The **map view updates** to show a route from the user's location to the reserved bike station, using visual cues (like a highlighted line).
4. **Success / Error Cases**:
    - **Success**: The user successfully reserves the bike, and the app provides a timer to track the reservation duration, ensuring they reach the station within the allocated time.
    - **Error**: If no bikes are available, the app notifies the user and suggests alternative stations with available electric bikes.
    - **Other Errors**: If the reservation fails (e.g., due to network issues), the app displays an error message and prompts the user to retry or select a different bike.
5. **Design Motivation**:
    - This design focuses on a **fast, intuitive reservation process**, with minimal steps to secure a bike, making it highly suitable for users in a rush.
    - The app's **map-based UI** and real-time **availability updates** improve user awareness, reducing frustration when bikes are in high demand. Color-coded battery levels help users choose a suitable bike at a glance.
    - Following best practices from successful bike-sharing apps, the design ensures **easy navigation** with on-screen guidance and location-based updates, addressing urban cyclists' needs for speed and eco-friendly transport.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user tries to reserve a bike at a nearby station, but all are reserved or in use. The app redirects them to another station within walking distance with available bikes.
    - **Alternative 2**: Due to a delay, the user's reservation time is about to expire. The app sends a notification offering the option to extend the reservation by a few minutes, ensuring they don't lose access to the bike.

3. Scenario: Checking Real-Time Public Transport Delays

Starting Point

- **User is authenticated** and has a planned trip saved or has selected a specific route in the app.

User Profile

- **Public Transport User**: This user frequently uses buses or trains and relies on real-time updates to avoid delays and manage transfers, especially during busy times or in case of disruptions.

Task Description

- **Objective**: Check the real-time status of a planned trip to see if the bus or train is on time or delayed.
- **Motivation**: Avoid unnecessary waiting time, minimize delays, and prevent missed transfers.

Context

- This task is usually performed during rush hours or in situations where unexpected delays, weather issues, or service disruptions may impact public transport schedules.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user starts on the **main dashboard** where their planned trips are listed, along with quick access to real-time updates for each trip.
2. **UI Elements**:
    - The user taps on a **saved trip** or **selected route** for the current journey.
    - Within the trip details view, there's an option to check **"Real-Time Status"** with live updates.
    - **Schedule Info**: The app displays estimated arrival times for each stop along the route, showing any delays with colored indicators (e.g., green for on-time, red for delays).
    - **Notifications Option**: The user can toggle on **real-time notifications** for their selected route to receive updates on further delays or schedule changes.
3. **System Reactions**:
    - Upon selection, the app retrieves the latest data from the transport system, updating the arrival times and displaying any delays in a clear format.
    - If the user's bus or train is delayed, the app displays an **estimated delay time** (e.g., "+10 min") next to each affected stop on the route.
    - The app may suggest alternative routes if delays are significant, with an option to **view alternate routes** or transfer points.
4. **Success / Error Cases**:

    - **Success**: The user sees up-to-date arrival times, confirming that the transport is either on time or delayed, and can adjust their plans if needed.
    - **Error**: If real-time data is unavailable (e.g., due to connectivity issues), the app displays an error message explaining that real-time updates can't be retrieved, and shows the scheduled time as a fallback.
    - **Other Errors**: If the user selects a route without real-time data support, the app notifies them and suggests turning on notifications for any changes to scheduled times.
5. **Design Motivation**:
    - This design aims to streamline **real-time access** to transport information, allowing the user to quickly see if they need to adjust their plans.
    - Using **color-coded indicators** for delays and **clear estimated times** follows best practices in UX for transport apps, making it easy to understand delays at a glance.
    - The alternative route suggestion feature, combined with real-time notifications, improves **user control and flexibility**, reducing stress and enhancing trust in the app's reliability.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user receives a notification of a delay and uses the app's suggested alternative route to reach their destination on time.
    - **Alternative 2**: Upon checking the real-time status, the user sees that their train is delayed but opts to stay on their planned route, setting up notifications to monitor any further delays

4. Scenario: Booking a Shared Electric Car for an Evening Ride

Starting Point

- **User is authenticated** and currently browsing available shared electric vehicles on the app.

User Profile

- **Evening Commuter or Social Traveler**: This user typically books shared vehicles for evening or weekend activities, such as going to dinner or meeting friends, and prefers eco-friendly options to minimize emissions.

Task Description

- **Objective**: Find and book a shared electric car for an evening trip across the city.
- **Motivation**: The user seeks a convenient, comfortable, and eco-friendly travel solution that eliminates parking and traffic concerns.

Context

- This task is commonly performed during non-peak hours, such as evenings or weekends, when users value flexibility and convenience for social or personal outings.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user opens the **vehicle selection map** in the app, which shows all available shared electric cars nearby.
2. **UI Elements**:
    - On the map, **car icons** represent the locations of available electric vehicles. The user taps a nearby icon to view details.
    - A **vehicle information panel** displays key details for each car, including battery level, estimated range, and cleanliness rating.
    - The user taps the **"Book Vehicle"** button within the panel to proceed with the booking.
3. **System Reactions**:
    - Upon booking, the app reserves the car and displays a **confirmation screen** with the car's details, the reservation time limit, and estimated travel time to the car's location.
    - The app provides **walking directions** from the user's current location to the car's location, highlighted on the map.
    - The app also gives an option to **start a pre-trip checklist**, ensuring the user checks the car's battery, tire condition, and cleanliness upon arrival.
4. **Success / Error Cases**:
    - **Success**: The user successfully books the car, receives directions, and completes the pre-trip checklist. The app starts a countdown to track reservation duration, ensuring they reach the car within the set time.
    - **Error**: If the selected car becomes unavailable before the booking is confirmed, the app notifies the user and suggests other nearby vehicles.
    - **Other Errors**: If the user arrives at the car and finds it in unacceptable condition, they can report it through an option in the app. The app then suggests alternative nearby vehicles or provides customer support contact information.
5. **Design Motivation**:
    - This design focuses on a **quick, intuitive booking process** with minimal steps, perfect for users looking to arrange transportation on the go.
    - Real-time **vehicle status indicators** and **battery level display** help the user make informed choices, enhancing convenience and reliability.
    - Using **pre-trip checklists** and real-time updates ensures vehicle quality, inspired by best practices from popular car-sharing services, which enhance user trust and satisfaction.
    - A clean, map-based UI with **directions to the car** improves usability, especially in unfamiliar areas, by guiding the user seamlessly from reservation to travel.
6. **Alternative Scenarios**:

- **Alternative 1**: The user reserves a car but, due to unforeseen delay, is unable to reach the car before the reservation expires. The app sends a reminder and offers to extend the reservation time if possible.
- **Alternative 2**: Upon arrival at the car, the user realizes they need a larger vehicle for additional passengers. The app allows them to cancel the reservation without a fee and suggests available nearby cars that meet their updated requirements.

5. Scenario: Identifying the Most Eco-Friendly Travel Option for a Daily Commute

Starting Point

- **User is authenticated** and has previously set up their **daily commute profile**, including their typical commute route and preferences.

User Profile

- **Environmentally-Conscious Commuter**: This user is focused on adopting more sustainable transportation options and is highly motivated to reduce their carbon footprint. They may be tracking emissions or looking to shift away from single-occupancy vehicle use.

Task Description

- **Objective**: Compare various modes of transport—bike, walking, and public transit—and determine which one produces the lowest carbon footprint for their daily commute.
- **Motivation**: The user seeks to minimize their environmental impact while commuting to and from work.

Context

- This task is typically performed during regular commuting hours, or when the user is trying to make more eco-friendly decisions, especially as part of a **green commute initiative** or to start a more sustainable lifestyle.

Detailed Scenario Steps

1. **Starting Point in the UI**:
   - The user begins on the **dashboard** screen after logging in, which displays their daily commute route and a prominent button labeled **"Compare Eco-Friendly Travel Options"**.
2. **UI Elements**:

    - The user taps **"Compare Eco-Friendly Travel Options"**.
    - They are presented with options to select transportation modes: **Bike**, **Walking**, and **Public Transit** (with icons for each mode).
    - A slider or dropdown allows the user to choose the **date and time** of their commute to account for variations in transport availability.
    - **Carbon Footprint Indicator**: Each mode of transport is associated with an **estimated carbon footprint**(e.g., kg CO2 per trip).
    - **Efficiency Indicator**: Displays the **total time and distance** for each option.
3. **System Reactions**:
    - Upon user selection, the app displays a **side-by-side comparison** of each mode's carbon footprint, cost, and travel time.
    - The **most eco-friendly option** (i.e., lowest carbon footprint) is highlighted in green, while less eco-friendly options are displayed with a yellow or red icon.
    - The app may also suggest **alternative routes** or combinations (e.g., bike-to-train) that could further reduce emissions.
    - Users can click on **"More Info"** to see detailed statistics on the carbon emissions for each mode.
4. **Success / Error Cases**:
    - **Success**: The user successfully identifies the most eco-friendly option, receives a recommended mode of transport for the day, and is presented with an alternative in case of weather or public transport issues.
    - **Error**: If no options are available (e.g., bike-sharing is out of service), the app displays a message informing the user and suggests alternative eco-friendly options, like walking or public transit.
    - **Other Errors**: If the user enters an invalid route or location, the app prompts them to enter a valid commute destination and automatically updates the carbon footprint calculations based on the new route.
5. **Design Motivation**:
    - The design focuses on presenting data clearly and encouraging environmentally-friendly choices through intuitive interface elements like icons, sliders, and side-by-side comparisons. This approach reduces cognitive load, helping the user quickly assess their options.
    - **Visual Clarity**: The use of color (green for eco-friendly, red for higher carbon emissions) ensures the app is easy to navigate and emphasizes sustainability. This follows **best practices for eco-conscious design** seen in similar apps and is aligned with principles of **green UX design**.
    - The option to see detailed statistics supports **user education**, which is important for environmentally-minded users who may want to track their impact over time.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user selects a route with **public transit**, but due to train delays or overcrowding, they switch to **bike-sharing**, leading to a slight increase in carbon footprint but still remaining eco-friendly.
    - **Alternative 2**: On a particularly **rainy day**, the user opts for **public transit** instead of walking or biking, but the app shows that biking would have still been more eco-friendly had it not been for weather conditions.

6. Scenario: Checking the Availability of Park-and-Ride Facilities

Starting Point

- **User is authenticated** and has entered their destination in the app to plan a mixed-mode trip that involves both driving and public transit.

User Profile

- **Suburban Car Commuter**: This user drives from a suburban area to the city center but prefers to park at a park-and-ride station and use public transport for the latter part of their journey to avoid heavy traffic and parking challenges in the city.

Task Description

- **Objective**: Locate an available parking spot at a park-and-ride facility near a public transit station.
- **Motivation**: Minimize inner-city driving, reduce travel stress, and lower commuting costs by combining driving with public transit.

Context

- This task is often performed during peak hours by commuters coming from suburban areas to city centers, seeking a convenient way to integrate car and public transit travel.
-

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user begins on the **trip planning screen**, where they have entered their destination. The app suggests nearby park-and-ride facilities along the route.
2. **UI Elements**:
    - A **map view** shows available park-and-ride locations along the route, with icons indicating parking availability (e.g., green for available, red for full).
    - The user taps on a specific park-and-ride facility icon to view **availability details**, including current open spots, distance from the user's location, and transit options from that location.
    - The app displays a **"Reserve Parking"** button for park-and-ride stations that support reservations, along with an **"Add to Route"** option to integrate the facility into the trip.
3. **System Reactions**:
    - When the user selects a facility, the app retrieves real-time parking availability data and displays it on the screen, updating in real-time if capacity changes.
    - For stations with reservation support, the app confirms the reservation and provides a **QR code or digital pass** for parking access.

    - The app also provides **directions to the facility**, showing the estimated driving time and the transit options available from the station to the user's final destination.
4. **Success / Error Cases**:
    - **Success**: The user finds an available spot, reserves it if possible, and the app provides directions to the facility. Upon arrival, they scan the QR code or show the digital pass to access parking.
    - **Error**: If the chosen facility is full before they confirm the reservation, the app displays an alert and suggests nearby park-and-ride locations with availability.
    - **Other Errors**: If real-time parking data cannot be retrieved, the app displays an estimated capacity based on past data and prompts the user to check availability upon arrival.
5. **Design Motivation**:
    - This design emphasizes **seamless integration of car and transit travel**, making it convenient to switch between the two modes.
    - The use of **real-time availability indicators** enhances user control and reduces stress by allowing them to make quick, informed decisions.
    - Allowing reservations and providing **digital parking passes** improves accessibility, especially for users who commute during peak hours and rely on secured parking options.
    - The combination of **map view** and **real-time data updates** helps users visually plan their journey, inspired by best practices from car-sharing and transit apps, enhancing usability and confidence in the app.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user sees that their preferred park-and-ride facility is full but opts to park at a nearby alternate facility suggested by the app.
    - **Alternative 2**: The user decides to reserve a parking spot at a facility that is slightly farther but offers better transit options, prioritizing convenience for the final leg of their journey.

7. Scenario: Offering a Personal Travel History Report

Starting Point

- **User is authenticated** and accessing their account dashboard, where they can view travel insights and history.

User Profile

- **Regular Traveler Interested in Sustainability**: This user frequently uses the service and wants to track their travel habits for greater efficiency and to reduce environmental impact.

Task Description

- **Objective**: Review a monthly report summarizing travel history, including routes, transportation modes, and environmental impact (e.g., CO2 emissions).
- **Motivation**: Gain insights into travel patterns to make more informed, eco-friendly decisions and optimize travel habits.

Context

- This task is usually performed at the end of the month or periodically when the user wishes to assess their travel efficiency, environmental impact, or set sustainability goals.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user starts from the **account dashboard**, where a section labeled “Your Monthly Travel Report” is displayed.
2. **UI Elements**:
    - A **report summary card** provides a high-level overview, displaying total trips, distances, CO2 emissions, and transportation modes used.
    - The user taps a **"View Full Report"** button to access detailed information, including visualizations such as charts and maps showing route frequency and CO2 emissions per trip.
    - There is an **"Environmental Impact"** section with comparisons (e.g., CO2 savings compared to driving alone) and personalized tips for reducing emissions based on the user's travel history.
3. **System Reactions**:
    - When the user taps “View Full Report,” the app retrieves and loads monthly data, presenting it in **graphical formats** (e.g., bar charts for trip frequency by mode, line graphs for emissions trends).
    - A **route map** highlights frequently taken routes, with markers showing each trip’s distance and CO2 emissions.
    - The app also generates **personalized insights**, such as “Switching to bike for short trips could save an additional 5% in emissions,” based on the user’s travel patterns.
4. **Success / Error Cases**:
    - **Success**: The user successfully views the report and understands the breakdown of travel modes and their environmental impact. They can use tips provided by the app to improve future travel choices.
    - **Error**: If data for certain trips cannot be retrieved, the app notifies the user and indicates missing data points in the report, ensuring transparency while still providing the rest of the report.
    - **Other Errors**: If the app encounters a loading issue, it displays a “Try Again” button and offers troubleshooting tips (e.g., checking internet connection).
5. **Design Motivation**:
    - This design prioritizes **data visualization and personalization**, helping users understand their habits through clear and accessible charts, inspired by best practices in fitness tracking and sustainability apps.
    - By displaying **CO2 emissions** and providing **practical recommendations**, the app encourages eco-friendly decisions, supporting users in meeting sustainability goals.

    - Visual elements like **route maps and graphs** make data engaging and understandable, aligning with the design principles of user-centered transparency and motivation for behavioral change.
    - The **comparison feature** provides context, showing users their impact in a relatable way (e.g., saved emissions), which is a proven strategy for promoting sustainable behavior.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user checks their travel report mid-month to get preliminary insights. The app displays data for the first half of the month, along with a note that a full report will be available at month's end.
    - **Alternative 2**: The user notices high CO2 emissions for their most-used routes. The app suggests alternative, eco-friendlier routes and highlights public transit or bike options, helping them make a proactive change.

8. Scenario: Sharing a Trip Plan with Friends for a Group Outing

Starting Point

- **User is authenticated** and has created a trip itinerary with details such as departure times, meeting points, and chosen transportation modes.

User Profile

- **Socially Active User Planning a Group Event**: This user often organizes outings with friends or colleagues and values efficient coordination and eco-friendly transportation options.

Task Description

- **Objective**: Create and share a detailed trip plan, allowing friends to view meeting times, locations, and transport modes for easy coordination.
- **Motivation**: Ensure everyone in the group arrives on time, ideally using similar or sustainable transport options.

Context

- This task is typically performed before social events like dinners, concerts, or team outings, when efficient planning and coordination are crucial.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user starts from the **trip itinerary screen**, where they have already created a plan with stops, times, and transport modes.
2. **UI Elements**:
    - The **itinerary summary** displays the planned route, stops, estimated travel times, and transport modes, along with meeting points for each stop.

    - A **"Share Trip Plan"** button allows the user to send the itinerary to friends. When tapped, the user can choose how to share (e.g., via link, messaging apps, email).
    - The app also provides a **"Send Notifications"** toggle, enabling the user to allow friends to receive updates if any changes are made to the plan (e.g., new meeting point, adjusted times).
3. **System Reactions**:
    - When the user taps “Share Trip Plan,” the app generates a shareable link that includes the trip details and invites friends to view and join the trip.
    - The app confirms successful sharing and displays a **notification icon** for the user to track if friends have opened the link.
    - If friends join the trip through the link, the app shows a **participant list** under the itinerary, allowing the user to see who has agreed to the plan.
4. **Success / Error Cases**:
    - **Success**: The user successfully shares the trip plan with friends, who receive a notification or link, allowing them to view the itinerary and join.
    - **Error**: If the shareable link fails to generate, the app prompts the user to retry and offers alternative sharing methods (e.g., copy-pasting details to messaging apps).
    - **Other Errors**: If a participant cannot view the trip plan due to compatibility issues, the app provides a “view on web” option to access the itinerary via a web browser.
5. **Design Motivation**:
    - This design prioritizes **social and collaborative features**, inspired by calendar-sharing and group coordination tools, making it easy for users to share and synchronize plans.
    - The **real-time participant tracking** and **notifications** for updates enhance transparency and reduce confusion, ensuring everyone in the group is informed of any changes.
    - The **toggle feature for notifications** respects users’ preferences by allowing them to opt-in or out of updates, maintaining a balance between functionality and user control.
    - Integrating **environmental impact suggestions** subtly encourages eco-friendly travel choices, supporting the app’s sustainability goals without imposing restrictions.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user creates a group chat within the app to discuss the itinerary. Friends receive the itinerary link in the chat and can reply with questions or changes directly in the group.
    - **Alternative 2**: The user sets up the trip plan with a public transport preference, and the app shows friends which buses or trains to take. Friends have the option to opt for different transport modes while still keeping track of meeting points and times.

9. Scenario: Finding Charging Stations for an Electric Vehicle

Starting Point

- **User has opened the app** and activated the map function for EV-related services, showing a map view of nearby charging stations.

User Profile

- **Electric Vehicle (EV) Owner**: A user who drives an electric car and needs to plan charging stops, especially for long trips.

Task Description

- **Objective**: Locate available charging stations on the route or near the destination to ensure the vehicle has sufficient charge for the remainder of the trip.
- **Motivation**: Avoid running out of charge mid-trip by planning stops at conveniently located stations.

Context

- This task is typically done before or during longer trips, when users need to account for the availability of charging stations along their route or at their destination.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user starts from the **map screen** with the EV services layer activated, displaying icons for nearby charging stations.
2. **UI Elements**:
    - The **map view** shows available charging stations as pins or icons, with details like distance, charger type, and availability status.
    - A **search bar** allows the user to enter their destination or set filters (e.g., fast charging, compatibility with their EV model).
    - When the user taps a station icon, a **station detail card** appears, showing information like current availability, pricing, estimated charging time, and distance.
3. **System Reactions**:
    - As the user scrolls the map, the app dynamically updates the station icons, displaying relevant options along the user's route or near their destination.
    - When the user selects a station, the app provides an **"Add to Route"** option, integrating the station as a stop on their trip.
    - The app also suggests **alternative stations** along the route in case the user's first choice becomes unavailable, with push notifications for real-time updates.
4. **Success / Error Cases**:
    - **Success**: The user finds an available station, views details, and adds it to their route, confident they'll have a reliable charging option.
    - **Error**: If no stations are available in the area, the app notifies the user, showing alternative options within a feasible detour distance.

    - **Other Errors**: If the station data fails to load, the app prompts the user to retry and provides a message to check the internet connection or refresh the map view.
5. **Design Motivation**:
    - This design prioritizes **real-time data and convenience**, inspired by existing EV and navigation apps that emphasize station availability and integration into trip planning.
    - The **"Add to Route"** feature reduces the user's effort in managing multiple apps for navigation and charging, making for a seamless experience in a single app.
    - Visual indicators and **up-to-date status information** reduce the uncertainty around availability, a common pain point for EV users.
    - By suggesting **alternative charging stations**, the app helps the user stay flexible, allowing them to adjust plans without the risk of delays or low battery levels.
6. **Alternative Scenarios**:
    - **Alternative 1**: The user filters stations by "fast charging only" and "available now." The app shows only real-time options matching these criteria along their route.
    - **Alternative 2**: The user sets a charging station as a preferred stop. If the station becomes unavailable, the app reroutes them to the nearest alternative and updates their trip plan automatically.

10. Scenario: Reporting an Issue with a Transport Service (e.g., Broken Bike, Delayed Bus)

Starting Point

- **User is authenticated** and either currently in transit or has just completed their journey.

User Profile

- **General Commuter or Shared Transport User**: A user who regularly uses public or shared transportation and may encounter service issues during their travels.

Task Description

- **Objective**: Report an issue (e.g., broken bike, delayed bus) to notify the service provider for quick resolution.
- **Motivation**: Ensure that the problem is logged for either immediate fixing or for alternative route suggestions, allowing smoother journeys for all users.

Context

- This task is usually performed right when the disruption occurs, enabling timely reporting and ideally helping the user or others avoid similar issues on their journey.

Detailed Scenario Steps

1. **Starting Point in the UI**:
    - The user starts from the **active journey screen** if in transit, or from the **completed journey history** if reporting after the trip.
2. **UI Elements**:
    - The user selects the **"Report Issue"** button within the trip details screen, which opens an issue report form.
    - The **issue form** has fields like “Type of Issue” (dropdown options: broken bike, bus delay, route problem), “Description,” and an **optional photo upload** for visual proof.
    - The **location and time** fields auto-populate based on the journey information, reducing manual input and ensuring accurate data.
3. **System Reactions**:
    - When the user submits the form, the app confirms the report with a **“Thank You”** screen and provides an estimated response time if applicable.
    - The app also offers an **“Alternative Route Suggestions”** button in case the disruption requires an immediate travel adjustment, leading the user to view similar routes.
    - The app sends a **push notification** if the issue gets resolved or if any further action is required by the user.
4. **Success / Error Cases**:
    - **Success**: The user successfully submits the issue report and receives confirmation with a case ID for reference.
    - **Error**: If the report fails to send due to a network issue, the app informs the user with a retry option and prompts them to check connectivity.
    - **Other Errors**: If the auto-populated location or time fields are incorrect, the user can manually adjust them, with the app providing a tooltip on how to edit.
5. **Design Motivation**:
    - This design focuses on **user efficiency and transparency**, inspired by successful support features in apps like Uber, where users can report issues with minimal friction.
    - By **auto-populating details** such as time and location, the design minimizes user effort and ensures accurate reporting.
    - The **alternative route suggestions** provide real-time support, reducing frustration by helping the user adjust plans promptly.
    - Offering **follow-up notifications** for resolution status aligns with user expectations for transparency and keeps them informed without needing to recheck the app.
6. **Alternative Scenarios**:
    - **Alternative 1**: If the user’s trip is delayed, they can tap “Alternative Routes” directly from the journey screen, which then auto-generates a report to notify the provider of the delay.
    - **Alternative 2**: For broken shared vehicles (e.g., bikes or scooters), the user can select “Report Issue” directly from a map icon, even if they’re not actively on a trip, making it easier to report problems with idle shared vehicles.