**Use Cases**

**Use Case 1: Creating a Travel Schedule**

**Actors:**

* Registered User

**Preconditions:**

* User is logged into the application.
* The city database and Google Places API are accessible.

**Basic Flow:**

1. The user selects the **"Create Schedule"** option from the navigation bar.
2. The system prompts the user to select a **city**.
3. The user selects a city, and the system retrieves its top attractions.
4. The system prompts the user to select a **travel period** via a calendar interface.
5. The system retrieves weather forecasts for the selected period.
6. The system prompts the user to **enter or select a hotel location**.
7. The user searches and selects activities from the suggested list.
8. The user **drags and drops** activities into a structured itinerary.
9. The system calculates and displays travel time between activities.
10. The system generates an **optimized daily route map**.
11. The user saves the schedule, which is stored in their profile.

**Alternative Flows:**

* If the selected city is unavailable, the system displays that maybe the city name is not spelled corectly.
* If the user tries to add an activity that is closed during the selected time, the system prevents it and suggests alternatives.
* If the internet connection is lost, the system temporarily saves progress and syncs when reconnected.

**Postconditions:**

* The schedule is successfully saved and accessible in the "View Schedules" section.

**Use Case 2: Modifying an Existing Schedule**

**Actors:**

* Registered User

**Preconditions:**

* The user has an existing schedule.

**Basic Flow:**

1. The user selects **"View Schedules"** from the navigation bar.
2. The system displays the list of saved schedules.
3. The user selects a schedule and chooses **"Modify"**.
4. The system opens the schedule in edit mode.
5. The user can:
   * Change the **hotel location**.
   * Remove or add **activities**.
   * Adjust **activity time slots**.
6. The system recalculates travel times and updates the itinerary.
7. The user saves changes, and the system updates the schedule.

**Alternative Flows:**

* If the user removes an activity, the system suggests replacements based on time availability.
* If an activity is unavailable, the system notifies the user and suggests alternatives.

**Postconditions:**

* The schedule is successfully updated and saved.

**Use Case 3: Viewing and Navigating a Schedule**

**Actors:**

* Registered User

**Preconditions:**

* The user has at least one saved schedule.

**Basic Flow:**

1. The user selects **"View Schedules"** from the navigation bar.
2. The system displays the list of saved schedules.
3. The user selects a schedule to view details.
4. The system displays:
   * The **structured itinerary** with time slots.
   * The **weather forecast** for each day.
   * The **map with the planned route**.
   * A link to **Google Maps for real-time navigation**.
5. The user can choose to:
   * Modify the schedule.
   * Start navigation using Google Maps.

**Alternative Flows:**

* If the user has no schedules, the system suggests creating one.
* If internet connectivity is lost, the system displays the last saved version.

**Postconditions:**

* The user successfully views and navigates their planned itinerary.

**Use Case Diagrams**

### ****Diagram 1: Travel Schedule Management****

#### ****Actors:****

* Registered User

#### ****Use Cases:****

1. **Create a Travel Schedule**
2. **Modify an Existing Schedule**
3. **View and Navigate a Schedule**
4. **Delete a Schedule**

#### ****Relationships:****

* The **Registered User** interacts with all use cases.
* "Modify an Existing Schedule" and "Delete a Schedule" are only available for existing schedules.
* "View and Navigate a Schedule" depends on "Create a Travel Schedule."

### ****Diagram 2: Activity and Location Selection****

#### ****Actors:****

* Registered User
* Google Places API

#### ****Use Cases:****

1. **Search for a City**
2. **Select Activities & Locations**
3. **Filter Activities (Budget, Type, etc.)**
4. **View Activity Details**
5. **Drag & Drop Activities into the Schedule**
6. **View Map & Routes**

#### ****Relationships:****

* The **Registered User** initiates all actions.
* The **Google Places API** provides activity details, location data, and filters.
* "View Map & Routes" depends on the selected activities.
* "Drag & Drop Activities" depends on "Select Activities & Locations."

These descriptions define the core functionality of the diagrams. Next, I can help you create the actual visual diagrams or refine the descriptions further. Let me know how you'd like to proceed! 🚀