# Use Case Document: YatriSimplified

**Domain:** Travel

**Use Case Name: YatriSimplified**

**Use Case Description:** This use case describes the process of planning, optimizing, and booking a complete journey using public transportation within the travel coordination system. The system allows users to input a source and destination and receive a customized, end-to-end travel plan using available transportation modes such as buses, trains, cabs, and flights. The system analyses available routes, suggests optimized paths based on cost and time, and facilitates a single-point payment for the entire trip.

**Use case Goal:** To allow a user to search, view, and book an optimized journey using a combination of public transportation (train, bus, cab, flight) from a source city to a destination, with a single payment.

**Actor:** Customer (Traveler)

**Pre-Conditions**

- User must be registered and logged into the system.
- Database must contain transport route data (train, bus, cab, flight).
- Cities should be listed and valid.

**Post Conditions**

- User receives a confirmed itinerary containing selected travel segments.
- Booking is marked as "paid" in the system.
- The entire journey plan is stored in the database for future reference.

**Main Flow (Basic Path)**

1. Initiate: The user navigates to the Plan Journey section after logging into the travel platform.
2. Action Selection: The user selects the desired action
    - Enter Source and Destination
    - View Available Routes

    - Choose Recommended Option
    - Proceed to Book and Pay

3. Data Entry/Selection
    - **Enter Journey Details**: The user enters the following
        - Source city (e.g., Davangere)
        - Destination city (e.g., Kashmir)
    - **View Travel Options**: The system displays various combinations of travel routes using
        - Trains
        - Buses
        - Flights
        - Cabs
    - **Choose Route**: The user selects a journey option, either
        - Recommended **Fastest Route**
        - Recommended **Cheapest Route**
        - **Custom Combination**

4. **Validation:** The system checks if transport options exist between the selected cities and ensures availability for the chosen date. It verifies the total cost and duration, avoiding any duplicate or conflicting travel segments.
5. **Confirmation:** A detailed summary is displayed showing transport modes, timings, and the total fare. The user reviews the plan and confirms the booking to proceed.
6. **Processing:** The system processes the payment and marks the journey as confirmed. Itinerary details are saved and options to download or view the journey are provided.

**Alternate Flows**

- No Transport Available: If no transport options are available between the entered cities, the system shows a message like “No routes found”. The user is asked to modify source/destination or choose a flexible date.
- Segment Unavailable: If one segment (e.g., flight or train) is fully booked or unavailable for the selected date. The system suggests alternate routes with different transport combinations or dates.

- Invalid Input: If the user enters an invalid or unsupported city name. The system shows an error and prevents proceeding until corrected.
- User Cancels Before Confirmation: If the user changes their mind before confirming the journey. They can go back, edit their route, or cancel without any charges.

**Possible Test Cases**

| Test Case ID | Description | Input Data | Expected Outcome |
|---|---|---|---|
| TC01 | User registration with valid details | Name, email, password | Account created successfully, redirected to login |
| TC02 | User login with correct credentials | Email, password | User is logged in and redirected to homepage |
| TC03 | User login with incorrect password | Email, wrong password | Error message: "Invalid credentials" |
| TC04 | Plan journey with valid source & dest | Source: Davangere, Destination: Kashmir | Shows transport options with cost and time estimates |
| TC05 | Plan journey with no transport found | Source: Unknown, Destination: Remote location | Message displayed: "No transport options available" |
| TC06 | Suggest fastest route | Valid route entered | System displays the fastest route with time details |
| TC07 | Suggest cheapest route | Valid route entered | System displays the cheapest route with cost details |
| TC08 | Journey summary confirmation | User selects a route | Displays summary with total price and travel time |
| TC09 | Payment processing (success) | Valid payment details | Payment successful, booking confirmed |
| TC10 | Payment processing (failure) | Invalid payment / network error | Error message shown, retry option available |
| TC11 | View itinerary after booking | Logged-in user with confirmed booking | Displays full journey plan with download option |

| | | | |
|---|---|---|---|
| TC12 | Session timeout before confirmation | User inactive on confirmation page | Session expired, redirected to login |
| TC13 | Invalid city input | Source: "123", Destination: "@" | Error message: "Invalid input" |
| TC14 | Journey cancellation before payment | User cancels before confirming | No booking saved, redirected to home |
| TC15 | Check input field validations | Blank fields during registration | Warning: "Fields cannot be empty" |
| TC16 | Duplicate registration | Email already registered | Error: "Email already in use" |
| TC17 | View journey history | Logged-in user with past bookings | Displays list of previous journeys |
| TC18 | Change transport preference | User modifies route from fastest to cheapest | System reprocesses and shows updated recommendations |
| TC19 | Partial segment unavailable | Train available, but flight segment full | Suggests alternate transport for unavailable segment |

**Expected Output**

1. **User Authentication**: Successful registration and login redirects the user to the homepage, while incorrect credentials or duplicate accounts show proper error messages.

2. **Journey Planning**: Upon entering valid source and destination, the system displays all transport options, along with fastest and cheapest recommendations.

3. **Input Validation**: Invalid or missing data prompts relevant error messages, ensuring proper data entry before proceeding.

4. **Booking Confirmation**: A detailed journey summary (segments, time, and cost) is shown before confirmation, allowing the user to review or cancel.

5. **Payment & Processing**: On successful payment, booking is confirmed and saved; failures trigger retries options and error prompts.

6. **Post-Booking Actions**: Users can view or download their itinerary, check journey history, or get suggestions for alternate transport if some segments are unavailable.

**Technologies Required**

- Frontend: HTML5, CSS3, JavaScript
- Backend: Python, Flask (Web framework)
- Testing Tools: Postman
- Other Useful Tools: Git + GitHub