Use Case UC1: Load Records from CSV

Primary Actor: System

Supporting Actors: CSV Data Source

Stakeholders and Interests:

- Admin: Wants to upload train route data without errors
- User: Wants accurate railway connection data
- System: Needs to load the data accurately to allow further operations

Preconditions:

- The system is operational.
- CSV Data Source is present and available to use in the correct structure.
- Admin has access to the data source and permissions to load data.

Success Guarantee (Postconditions):

- Railway connection data is loaded into the system from the CSV file.
- System is ready for search operations.

Main success scenario:

1. System accessed the CSV file.
2. System reads and loads the connection records.
3. Data is available in the system for search operations to be performed.

Extensions

- If the data source is not a csv file, indicate error and prompt the user to select a new file.
- If the CSV file cannot be read, stop the read operation, and prompt the user to select a new file.
- If a record in the file is invalid (i.e. does not have all the necessary parameters), skip the record, and continue processing the other entries.

Special Requirements

None

Technology and Data Variations List

Data is provided in a CSV file.

Open Issues: **None**

Use Case UC2: Search for Connection

Primary Actor: User

Supporting Actors: N/A

Stakeholders and Interests:

- User: Wants to find connections between 2 cities with a set of search parameters.

Preconditions:

- The system is operational.
- CSV Data has been loaded into the system.

Success Guarantee (Postconditions):

- The system displays a list of connections that match the search parameters from user input

Main success scenario:

1. The User provides search parameters.
2. The system validates the user input.
3. The system finds all direct connections that match the parameters.
4. The system calculates the trip duration for all connections found.
5. The system displays a list of all connection matches.

Extensions

- If any input is invalid, the system displays an error and user is prompted to re-enter a valid value

- If no direct connections are found:
    - The system provides indirect connections with either one or two stops in between the arrival and destination cities.

    - The trip duration calculation will include the time to change trains for the indirect connections.

- If no direct or indirect connections are found, the system indicates that there were no matches.

Special Requirements

- The search operations should work for any public parameter except for route ID (not-public).

Technology and Data Variations List:

- Days of operation can be represented as “Daily” or in the following formats: “Day, Day”, or “Day-Day”
- Arrival time may include (+1d)

Open Issues: **None**

Use Case UC2.1: Sort Results

Primary Actor: User

Supporting Actors: N/A

Stakeholders and Interests:

- User: Wants to sort the displayed results in a desired order (by price or by duration).

Preconditions:

- List of search results is displayed for the user

Success Guarantee (Postconditions):

- The system displays an updated list of connections that match with the sort option selected by the user

Main success scenario:

1. User chooses to sort by trip duration or price.
2. The system sorts the list.
3. The system displays the newly sorted list.

Extensions: **None**

Special Requirements

- Sorting option is only by duration and price

Technology and Data Variations List: **None**

Open Issues: **None**

Use Case UC3: Book Trip

Primary Actor: User

Supporting Actors: N/A

Stakeholders and Interests:

- User: Wants to book a trip for one or more travellers
- System: Needs to create valid booking and store the reservation records

Preconditions:

- System is operational
- Connection data is loaded
- Trip has been searched and selected
- Client enters valid information

Success Guarantee (Postconditions):

- A booked trip is created with a unique ID
- One or more reservations are created (maximum one per traveller)
- Each Reservation is assigned with a Ticket that has a unique ID
- All bookings are saved in the TripCollection
- Client records are maintained
- Booking confirmation is displayed

Main success scenario:

1. The user selects a trip from the search results.
2. The user specifies the number of travellers.
3. For each traveller user enters the person's first name, last name, age, and ID.
4. The system checks for duplicate reservations.
5. The system created a BookedTrip with a unique ID
6. For each traveller, the system creates a Reservation with a unique ID and generates a Ticket with unique ID.
7. The system saved the BookedTrip to the TripCollection.
8. The system displays booking confirmation with trip ID and the ticket IDs.

Extensions:

- Invalid user information
    - System displays error messages
    - Prompts the user to re-enter information

- Reservation already exists
    - System stops the booking
    - Displays error message informing the user that the person whose information was entered already has a reservation for that connection (duplicate ID)

Special Requirements

- Trip ID must be unique
- Reservation ID must be unique
- Ticket ID must be unique
- Only one reservation per client per connection

Technology and Data Variations List:

- Data for the bookings must be kept in memory for the duration of the session

Open Issues: **None**

Use Case UC4: View Trips

Primary Actor: User

Supporting Actors: N/A

Stakeholders and Interests:

- User: Wants to view their current trips and past trips.
- System: Needs to retrieve and display the trip information

Preconditions:

- System is operational and maintains records of trips
- TripCollection contains booked trips
- Client enters valid credentials

Success Guarantee (Postconditions):

- The system displays a list of all trips for the client
- Trips are marked as either past or current (today/future)
- All trip details are visible

Main success scenario:

1. The user selects View Trips.
2. The user enters last name and id.
3. The system validates the user information.
4. The system searches the TripCollection for matching trips.
5. The system categorizes the trips into current and past.
6. The system displays current trips.
7. The system displays past trips.
8. For each trip the system displays it for the user.

Extensions:

- Invalid credentials
    - System displays error message
    - Prompts the user to re-enter information

- No trips found
    - System displays message indicating that no trips were found

Special Requirements

- Search must only be by last name and ID
- Current and past trips must be distinguishable for the user to see

Technology and Data Variations List: **None**

Open Issues: **None**