Operation Contract: findTrip

Operation: findTrip(parameterType, parameterValue)

Precondition:
• None
• Routes loaded from the CSV

Postcondition:
• A list of Trip objects, *t*, is created.
• Zero, one, or many Trip objects may be created based on search criteria.
• The created Trip objects are appended to the list *t*.
• The system displays list *t*.
• All indirect Trip objects in *t* satisfy the active LayoverPolicy (max daytime/night layover limits).

Operation Contract: sortTrip

Operation: sortTrip(parameterType)

Precondition:
• A list of trips exists.
• The Trip list originates from database-loaded routes and may include numeric Trip IDs.

Postcondition:
• The same Trip objects are returned in a new sorted order.
• The sorted list is displayed.

Operation Contract: selectTrip

Operation: selectTrip(tripId)

Precondition:
• Trip exists.
• Trip must be valid and its tripId is a unique numeric identifier generated by the system.
• If Trip is indirect, its layovers already passed LayoverPolicy validation.

Postcondition:
• Trip details are displayed.
• The selected Trip object is temporarily stored for possible booking and later persistence.

Operation Contract: provideTravellers

Operation: provideTravellers(listOfClients)

Precondition:
• Trip object has been selected.
• User provides client information (first name, last name, id, age).
• Client data must conform to database schema types and unique constraints (clientId exists or is assignable).

Postcondition:
• List of Client objects is created one for each traveller.
• List of clients is stored and ready to be used in the booking process.
• Client records are validated against database or inserted if new.

Operation Contract: bookTrip

Operation: bookTrip(listOfClients, tripId)

Precondition:
• List of clients is not empty.
• Trip exists.
• User is authenticated.
• All layovers for an indirect Trip comply with the active LayoverPolicy.
• Database connection available for database operations.
• Trip object has numeric tripId.

Postcondition:
• New BookedTrip object is created
• For each Client, a Reservation object is created and linked to a new Ticket.
• All Reservation objects are aggregated into the BookedTrip.
• Each object (Trip, Client, Reservation, Ticket, BookedTrip, TripHistory) is persisted to the Database

• Confirmation message and booking details are returned to user.

Operation Contract: authenticateOrIdentify

Operation: authenticateOrIdentify(lastName, id)

Precondition:
• The user provides their information (last name and client ID).
• Credentials must match a Client record stored in the Database.

Postcondition:
• Session is established for that client.
• Validated Client object is retrieved from the Database for subsequent operations (view history, book trip).

Operation Contract: viewTripHistory

Operation: viewTripHistory(lastName, id)

Precondition:
• The client is authenticated.

Postcondition:
• A list of BookedTrip records associated with given client id is retrieved from the database
• Zero, one, or many BookedTrip objects are returned.