

Project Specification for Group _TUT5101-22_____

Team Name: Global Flight

Domain: Flight information and travel planning

A flight tracking and information platform that allows users to quickly search for real-time flight details (departure and arrival times), check ticket prices, find direct flights, and view baggage policies using data from the flighttrader24 API and other supporting APIs(if needed).

User Stories:

User story 1: As a user, I want to search for a flight by flight number, so that I can see its departure and arrival times easily.

User story 2: As a user, I want to add a flight to “My Favorite”, so that I can quickly check my preferred flights.

User story 3: As a user, I want to enter a flight number to check the ticket prices, so that I can compare and plan my budget.

User story 4: As a user, I want to input departure and destination airports to check if there are available flights, so that I can choose a suitable flight plan.

User story 5: As a user, I want to view recent flights operated by a specific airline, so that I can see its current operations.

User story 6: As a user, I want to track the live status of a flight in real time, so that I can see whether it's on time, delayed, or already landed.

Use Cases:

Use case 1: Search flight details

Main Flow:

- User enters the flight number.
- System retrieves flight details (departure, arrival, times, and airline).
- System displays the information in a readable format.

Alternative Flow:

- Invalid flight number -> display "Flight not found".

Use case 2: Favourite a flight

Main Flow:

- User clicks "Add to my favourite" button.
- System saves this flight in the user's local favourite list.

Alternative Flow:

- Flight already saved -> display "Flight already in My Favourite".

Use case 3: Check ticket prices

Main Flow:

- User enters a flight number.
- System retrieves the ticket prices data from a secondary API.
- System displays the ticket prices.

Alternative Flows:

- Invalid flight number -> display "Flight not found".
- Price not found -> display "Ticket prices unavailable".

Use case 4: Search flights

Main Flow:

- User enters a departure airport and/or a destination airport.
- System calls the Flightradar24 API to retrieve flights matching those criteria.
- System filters and displays flights that connect the two airports.
- User views available flights.

Alternative Flows:

- Invalid or unknown airport code -> display "Invalid airport input".
- No flights found for that route -> display "No flight available for this route".

Use case 5: View recent flights of an airline

Main Flow:

- User enters an airline name, e.g. Air Canada.
- System calls the Flightradar24 API to retrieve a list of recent or active flights operated by that airline.
- System displays each flight's number and its route (departure → arrival).

Alternative Flows:

- Invalid airline name -> display "Airline not found".
 - No recent flights -> display "No recent flights".
-

Use case 6: Track flight status

Main Flow:

- User enters a flight number.
- System calls the Flightradar24 API to retrieve live status data for this flight, such as estimated arriving time and current position.
- System displays a status summary.

Alternative Flow:

- Invalid flight number -> display “Flight not found”.
 - Flight is not currently active -> display “This flight is not currently active”.
-

MVP:

| Lead | Use Case [title of use case from previous section] | User Story [optionally, include this column to indicate which user story this use case is associated with] |
|-------------------------|--|--|
| Zhaotong, Frank | Use Case 1 | User Story #1 |
| Frank, Alex | Use case 2 | User Story #2 |
| Yanfei, David | Use case 4 | User Story #4 |
| Zhaotong, Alex | Use case 5 | User Story #5 |
| Zhaotong, Yanfei, David | Use Case 6 | User Story #6 |

Use case 3 is not part of the MVP but are planned for later development, time permitting.

Proposed Entities for the Domain:

Flight (Used in most use cases)

Instance variables:

- String flight_number
- String departure_airport
- String arrival_airport
- String status
- String departure_time
- String arrival_time

AirlineFlight extends Flight (Used in Use Case 5)

Instance variables:

- String airline
- Instance variables inherited from class Flight

Favourite (Used in Use Case 2)

Instance variables:

- String user_id
- String flight_number

FlightStatus (Used in Use Case 6)

Instance variables:

- String flight_number
- String estimated_arrival_time
- double latitude
- double longitude
- double altitude
- double speed

Proposed API for the project:

- API Name: Flighttrader24
- Link: <https://fr24api.flightradar24.com/docs/getting-started>
- Main service provided: Provides real-time and historical flight data, including flight number, route, schedule, status, and live tracking information such as location, altitude, and speed.
- Status: Not yet testing, will try to call the API as soon as possible.