Interactive Flight Destination Explorer

Requirements Analysis Document

---

1. Project Overview

The Interactive Flight Destination Explorer is a web-based application that helps travelers discover and book flights by visually indicating where they can fly directly to. Users are presented with a world map where they can select their city's airport, and the system will display all destinations reachable by direct flight from the selected departure point, using flight data retrieved via external APIs and stored in a backend database.

Users can interact with the map to explore destinations or connecting airports and, upon selecting one, are presented with flight options and the ability to book through an external site. Flights can be filtered based on travel date, airline, and price. The system is designed for travelers, travel agents, business users, and administrators who manage the backend flight information.

---

2. Functional Requirements (Use Cases)

Name of Use Case:	Admin		
Created By:	Spencer	**Last Updated By:**	Spencer
Date Created:	06/19/25	**Last Revision Date:**	06/19/25
Description:	Admins manage flight database, routing information, and any other flight related information needed.		
Actors:	Admin, flight database		
Preconditions:	• Admin has login credentials and appropriate permissions • Database is operating correctly		
Postconditions:	• Flight database is updated with current information		
Flow:	1. Admin logs into the dashboard 2. System displays current database and sync status 3. Admin selects a management function (update, delete, add data) 4. Admin enters new or updated information 5. System validates the data 6. Admin reviews the pending changes 7. Admin commits the changes 8. System updates the database and logs the action 9. Admin logs out		
Alternative Flows:	• Admin cancels changes before committing		
Exceptions:	Database connection failure, Invalid data		
Requirements:	Admin dashboard, database access, logging mechanism		

Name of Use Case:	Browse & obtain flight to book		
Created By:	Gil	**Last Updated By:**	Gil
Date Created:	06/19/25	**Last Revision Date:**	06/19/25
Description:	User finds a flight and goes to an external site to book		
Actors:	User, flights backend, vendor modules		
Preconditions:	• Flights exist from vendors for purchase • Flights database populated and active		
Postconditions:	• Vendor sites are active		
Flow:	1. User filters by departure and return flight dates via dropdown. 2. GET request from backend for flights occurring on those dates. Display map of all matching cities. 3. User clicks on departure city on map. This displays a map of places to directly fly from that airport. 4. User clicks on arrival city from the map. 5. GET request from backend for flights that exist between the two cities. Display available flights on-screen in new page. 6. User clicks on flight. 7. Redirect to corresponding vendor website that offers that flight.		
Alternative Flows:	1. User does not specify specific dates. Display map for any flights in database between cities. 2. User clicks on departure city on map. This displays a map of places to directly fly from that airport. 3. User clicks on arrival city from the map. 4. GET request from backend for flights that exist between the two cities. Display nearest available flights on-screen in new page. 5. User clicks on flight. 6. Redirect to corresponding vendor website that offers that flight.		
Exceptions:	None specified		
Requirements:	Backend flight database, external vendor integration, airport metadata, autocomplete search, flight route data, map rendering module, filter UI, backend filtering logic, flight details UI		

Name of Use Case:	Search and Recommend Destinations		
Created By:	Brian	**Last Updated By:**	Brian
Date Created:	06/19/25	**Last Revision Date:**	06/19/25
Description:	Travel agent uses the system to recommend destinations to clients.		
Actors:	Travel Agent		
Preconditions:	• System is accessible • Agent has access to search tools		
Postconditions:	• Agent compiles recommendations		
Flow:	1. Agent logs in 2. Agent selects filters and map options 3. Agent exports results for client use		
Alternative Flows:	• Agent prints page of selected destinations		
Exceptions:	Data not available		
Requirements:	Search tools, print/export		