

Requirement Analysis Phase

Objective:

To define and document all **functional**, **non-functional**, and **system** requirements for FlightFinder, ensuring alignment with user needs and technical feasibility.

1. Stakeholder Identification :

Stakeholder	Role/Interest
End Users (travelers)	Need accurate, fast, and customizable flight search
Project Owner (You)	Define vision, design, and development
Airline APIs/Partners	Provide real-time flight data
UI/UX Designer (if any)	Responsible for app design

2. Functional Requirements

These are the core **features** your app must perform.

Requirement ID	Description
FR-1	User can search flights by source, destination, and date.
FR-2	Flights are fetched from real-time airline APIs or data providers.
FR-3	Users can filter results by price, duration, stops, airline, and time.
FR-4	Sort flights by best match, price, or duration.
FR-5	View detailed flight info (layovers, baggage policy, aircraft type, etc.)
FR-6	Price alerts for selected routes and dates.
FR-7	Option to bookmark/save flights for comparison.
FR-8	Redirection to booking partner or airline site.

3. Non-Functional Requirements

These ensure performance, usability, and reliability.

Requirement ID	Description
NFR-1	App should return search results in < 3 seconds.
NFR-2	High availability and uptime of 99.5%.
NFR-3	Secure API communication using HTTPS.
NFR-4	Mobile-responsive and user-friendly UI.
NFR-5	Should support 500+ concurrent users without performance drop.
NFR-6	GDPR-compliant data handling.

4. System Requirements :

Requirement Type	Description
Hardware	Web server (cloud or VPS), optional CDN for fast delivery
Software	Frontend (React/Angular), Backend (Node.js/Django), Database (MongoDB/PostgreSQL)
APIs	Skyscanner API / Amadeus API / RapidAPI flight data sources
Hosting	AWS / Heroku / Vercel / Firebase

5. Assumptions & Constraints

- Assumption: Users have internet access and basic travel knowledge.
- Constraint: Real-time API data access may be rate-limited or paid.
- Constraint: Some airline partners may not support full data integration.

6. Success Criteria

- App fetches and displays flight results correctly from live APIs.
- User can filter, sort, and view details with minimal load time.
- Bookmarked flights persist across sessions (optional login).
- At least 80% user satisfaction in test group (if testing is planned).