# Software Requirements Specification (SRS) Project: Real-Time Global Flight Tracker Author: Anith Joy Date: 05/09/2025

## Introduction

### Project Overview

This project is a web application that allows users to track real-time flight data globally. It will fetch live aircraft positions using the OpenSky Network API and visualize them on an interactive world map.

### Scope

- Display global aircraft positions in real time.

- Allow users to retrieve flight details such as callsign, origin country, position, and speed.

- Provide search/filter capabilities by aircraft ID or country.

- Be accessible via a web browser on both desktop and mobile platforms.

### Future Scope

- Support user accounts and persistent data storage for favorites or historical data.

## System Overview

This is a full-stack web application with the backend built in Java Spring Boot and the frontend using a modern JavaScript framework. Real-time flight data will be fetched periodically from the OpenSky Network and rendered on a map interface. Users will be able to interact with the data via search and filter operations.

## Functional Requirements

### Real-Time Flight Display

- The system shall display aircraft as markers on a world map.

- Each marker shall represent a real aircraft fetched from the OpenSky API.

### Flight Details View

- The system shall display details (callsign, country, speed, altitude) when a marker is clicked.

### Search and Filter

- The system shall allow users to search flights by:

- ICAO24 identifier

- Country of origin

### Data Polling

- The system shall fetch live flight data every 10 seconds.

- (Optional) WebSocket-based real-time updates if scaling is required.

### User Account (Future Scope)

- The system may allow user login for saving favorite flights.

## Non-Functional Requirements

Performance

Usability

Reliability

Security

## System Architecture

### Backend

- Java Spring Boot application

- REST API layer to serve flight data

- Optional PostgreSQL database for persistent storage

### Frontend

- React.js + Leaflet.js for map rendering

- Calls backend API endpoints to fetch flight data

## API Integration

- OpenSky REST Endpoint: `?`

## Constraints

## Milestones

## Acceptance Criteria

## Future Enhancements

- User Login

- Save favorite flights.

- Show flight path history using stored data.

- Create analytics dashboards for air traffic trends.