Airline Fleet Management Desktop App - Project Scope

Project and Objectives

The main goal is to provide an easy and fast tool that can be helpful when making critical decisions regarding fleet operations. The application aims to streamline the entire aircraft management process, offering a comprehensive solution that empowers users to optimize resource utilization, ensure regulatory compliance, and enhance overall operational efficiency.

The main key functionalities of the app are:

1. User Roles and Permissions:

- Identify different user roles (e.g., administrator, pilot, maintenance personnel).
- Define permissions and access levels for each role.

2. Core Features:

Aircraft Management:

- Add, edit, and delete aircraft details.
- Track aircraft specifications, registration, and maintenance history.

Flight Scheduling:

- Create and manage flight schedules.
- Assign specific aircraft to flights.

Maintenance Tracking:

- Record and track maintenance activities for each aircraft.
- Schedule and prioritize routine and unscheduled maintenance.

Fuel Management:

- Monitor fuel consumption.
- Log fueling activities and costs.

3. Notifications and Alerts:

- Implement a notification system for upcoming maintenance tasks, expiring licenses, or scheduled flights.
- Allow users to set custom alerts based on specific criteria.

4. Reporting and Analytics:

- Generate reports on fleet performance, fuel efficiency, maintenance costs, etc.
- Provide visual analytics for better decision-making.

5. Integration with External Systems:

- Integrate with weather services for real-time weather updates.
- Connect with aviation authorities for regulatory compliance checks.

6. Security:

- Implement secure authentication and authorization mechanisms.
- Encrypt sensitive data to ensure confidentiality.

7. User-friendly Interface:

- Design an intuitive UI with a focus on usability.
- Use graphical elements to represent data effectively (e.g., charts, maps).

10. Performance Optimization:

- Optimize database queries and application logic for performance.
- Handle large datasets efficiently.

Technology and Development Environment

For technology aspects, it will be used:

a. Backend Development:

- Language: C#
- Framework: .NET Core
- Database: PostgreSQL

b. Frontend Development:

- UI Framework: Windows Presentation Foundation (WPF)

c. Integration and Scripting:

- Scripting Language: Python

d. Development Environment:

- IDE: Visual Studio
- Database Management: PostgreSQL
- Design Tools: Figma
- Version Control: Git (GitHub)

The goal is to aim for simplicity, an easy development, and integration with common tools for collaboration and monitoring. Adjustments can be made based on specific project requirements and preferences.