Software Requirements Specification

for

Aviation Safety System

Version 1.0

Prepared by: Anushri Viraj Sakhardande, Sindhu, Uppaluri Jahnavi

Manipal Institute of Technology, Manipal

28th, February 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product specified in this Software Requirements Specification (SRS) document is the "**Aviation Safety System**." This document outlines the software requirements for developing the Aviation Safety System, **version 1.0.** This SRS covers the entire system, including the passenger and administrator modules and the database management, analytical tools, and software requirements necessary for the system's implementation.

1.2 Document Conventions

The SRS follows these conventions: Each requirement statement has its priority for clarity. Standard formatting distinguishes headings, subheadings, and text. Key terms are highlighted for emphasis. Consistent language usage ensures clarity.

The SRS is organised as follows:

1. Introduction: Provides an overview of the project, the conventions, reading suggestions and scope.
2. Overall Description: Offers a system description, including user roles, functionality, and hardware/software requirements.
3. External Interface Requirements: Specifies interfaces with external systems or users.
4. System Features: Describes individual system features in detail.
5. Nonfunctional Requirements: Considers other aspects such as security, reliability, and maintainability.
6. Appendices: Includes supplementary information, such as glossary and references.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for various stakeholders in developing, implementing, testing, and managing the Aviation Safety System. The primary audience for the aviation safety system is the passengers of flights. These individuals can directly benefit from the information the system provides regarding safety based on the associated database.

Suggested sequence for reading:

1. Overview sections for all readers.
2. Developers: Detailed requirements (Specific Requirements, System Features).
3. Project managers: Overall Description, Resource Requirements.
4. Testers: Specific Requirements (testing-related sections).
5. Documentation writers: Overall Description, System Features.

1.4 Product Scope

The Aviation Safety System aims to achieve the following objectives:

1. Safety Assurance: Provide passengers with accurate and up-to-date information on the safety conditions of their intended flights, allowing them to make informed decisions about their travel.
2. Record Keeping: Maintain a comprehensive database of flight incidents, including damage criteria, to facilitate analysis and improve safety standards.

3. Predictive Analysis: Utilize historical flight data to predict potential risks and safety parameters for current and future flights, enhancing proactive safety measures.
4. Efficient Data Management: Ensure efficient handling of flight data, allowing for easy retrieval, analysis, and reporting.
5. Continuous Improvement: Facilitate ongoing analysis of aviation safety trends, enabling stakeholders to identify areas for improvement and implement corrective actions effectively.

1.5 References

1. ML.NET:
   - Documentation: ://docs.microsoft.com/en-us/dotnet/machine-learning/
   - GitHub Repository: https://github.com/dotnet/machinelearning
   - NuGet Package: ://www.nuget.org/packages/Microsoft.ML/
2. NumSharp (NumPy for .NET):
   - Documentation: https://github.com/SciSharp/NumSharp/wiki
   - GitHub Repository: https://github.com/SciSharp/NumSharp
   - NuGet Package: https://www.nuget.org/packages/NumSharp/
3. Font Awesome:
   - Documentation: https://fontawesome.com/v5.15/icons?d=gallery&p=2
   - GitHub Repository: https://github.com/FortAwesome/Font-Awesome
   - Official Website: https://fontawesome.com/

2. Overall Description

2.1 Product Perspective

The Aviation Safety System specified in this SRS is a **new, self-contained** product developed to address the need for enhanced safety assurance in air travel. It is not a part of an existing product family or a replacement for any specific current systems but rather a standalone solution to improve aviation safety standards. The Aviation Safety System operates independently, providing passengers real-time safety information and analysis to support their travel decisions. While it may interact with external systems for data retrieval or communication purposes, it is designed to function autonomously to fulfil its primary objectives.

The system's major components include:

1. User Interface: Allows passengers to access safety information and input flight data.
2. Database Management: Stores and manages records of flight incidents, safety ratings, and historical data.
3. Analytical Engine: Processes data to generate safety ratings, perform predictive analysis, and identify safety trends.
4. Administrative Tools: Enable system administrators to manage the database, update information, and perform system maintenance tasks.

2.2 Product Functions

The Aviation Safety System must perform the following significant functions or allow users to serve them:

1. User Registration and Authentication: Users must be able to register themselves into the system and authenticate their identity for access.
2. Data Management: The system should efficiently manage and store records of flight incidents, safety ratings, and other relevant data.
3. Safety Analysis: Conduct flight data analysis to determine safety ratings, identify trends, and perform predictive analysis for future flights.
4. User Interface: Provide a user-friendly interface for passengers to access safety information, input flight data, and receive safety ratings.
5. Administrative Tools: Offer administrative functionalities for system management, including adding, deleting, and updating database entries.
6. Reporting: Generate reports on safety statistics, trends, and analysis findings for passengers and administrators.
7. Security: Ensure data security measures are in place to protect sensitive information and prevent unauthorised access or tampering.

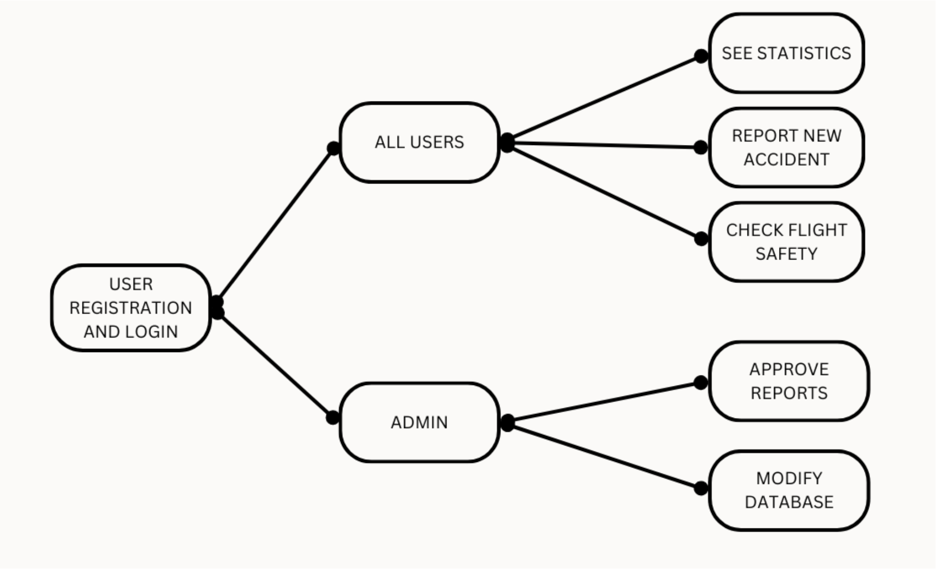

Fig 2.2.1. Data Flow Diagram for the Aviation Safety System

2.3 User Classes and Characteristics

The Aviation Safety System has the following user classes:

1. Passengers:
   - Characteristics: Varied educational levels, minimal technical expertise required, frequent system users.
   - Functions: Access safety information, input flight data, receive safety ratings based on various filters, and view reports.
2. Administrators:
   - Characteristics: Users with higher technical expertise are responsible for system management and maintenance.
   - Functions: Add, delete, and update database entries, generate reports, and oversee system functionality.

2.4 Operating Environment

The Aviation Safety System operates in the following environment:

1. Hardware Platform:
   - Processor: Intel dual core or above
   - Processor Speed: 1.0GHz or above
   - RAM: 1 GB RAM or above
   - Hard Disk: 20 GB hard disk or above
2. Operating System:
   - Windows: Versions compatible with the C# programming language and Oracle database, such as Windows 7, Windows 8, Windows 10, and their server equivalents.
3. Software Components:
   - Language: C# for application development.
   - Database: Oracle for data storage and management.
   - Additional software components may be required for data analysis, reporting, and interfacing with external systems. These components should be compatible with the chosen hardware platform and operating system.

2.5 Design and Implementation Constraints

Several items or issues can limit the options available to the developers of the Aviation Safety System:

1. Hardware Limitations: The specified hardware requirements, including processor speed, RAM, and hard disk space, impose constraints on system performance and scalability.
2. Software Components: Using specific technologies, tools, and databases, such as C# for application development and Oracle for data storage, limits alternative options.
3. Security Considerations: Implementing robust security measures to protect sensitive data and prevent unauthorised access or tampering imposes system architecture and design constraints.
4. Design Conventions and Standards: Adherence to design conventions, programming standards, and best practices ensures maintainability and compatibility with existing systems but may limit flexibility in design decisions.
5. Interfacing with External Systems: Compatibility and interoperability with external applications and systems, such as flight data providers and passenger booking platforms, require adherence to predefined communication protocols and data exchange formats.
6. Parallel Operations: Ensuring smooth parallel operations and avoiding conflicts between multiple system users or processes imposes constraints on concurrency control and data integrity mechanisms.
7. Customer Maintenance Responsibility: If the customer's organisation is responsible for maintaining the delivered software, the system design should consider ease of maintenance, documentation standards, and user support requirements.

2.6 User Documentation

A **user manual** will be created along with a **step-by-step usage video tutorial** detailing the functionalities available in the system. This tutorial will provide users with a hands-on walkthrough of critical tasks and workflows within the Aviation Safety System, allowing them to follow along and learn at their own pace.

2.7 Assumptions and Dependencies

The following assumptions and dependencies are for the system:

1. Third-Party Libraries: The Aviation Safety System relies on the FontAwesome for C# library for iconography and ML.NET framework for machine learning capabilities. Assumptions are made

regarding these third-party components' availability, compatibility, and support throughout the project lifecycle. Any changes in licensing, updates, or discontinuation of support for these libraries could impact system functionality and development timelines.

2. Integration with ML.NET: Integrating the ML.NET framework introduces dependencies on machine learning models and algorithms for predictive analysis within the Aviation Safety System. Assumptions are made regarding these models' accuracy, performance, and adaptability to the aviation safety domain. Model accuracy, data requirements, or algorithm performance changes may necessitate system architecture or feature prioritisation adjustments.
3. Development Environment Compatibility: Assumptions are made regarding the compatibility of the Font Awesome for C# library and ML.NET framework with the development environment, including the chosen programming language (C#) and IDE (Integrated Development Environment). Any incompatibilities or conflicts with development tools could impede development progress and require resolution.
4. Dependencies on Machine Learning Models: Integrating the ML.NET framework introduces dependencies on specific machine learning models and datasets for predictive analysis tasks within the Aviation Safety System. Assumptions are made regarding these dataset's availability, quality, and relevance for training and evaluation purposes. Data availability or quality changes could impact the accuracy and performance of predictive analysis functionalities.

3. External Interface Requirements

3.1 User Interfaces

The Aviation Safety consists of the following interfaces:

1. Login Page:
   - Interface Component: Login form for user authentication.
   - Characteristics: Simple and intuitive layout with fields for username and password entry. Error message display for invalid login attempts.
2. User Stats Page:
   - Interface Component: Dashboard displaying user statistics and relevant information.
   - Characteristics: Clear presentation of user stats. The navigation menu is positioned on the screen's side to access other pages easily. Standard buttons are present for logging out and accessing help resources.
3. Navigation Menu:
   - Interface Component: Side menu for navigating between different pages.
   - Characteristics: Consistent layout and design across all pages. Labelled menu items have been created for easy navigation. Highlight or indicate the current page to provide visual feedback to users.
4. Other Pages:
   - Interface Components: Various pages for system functionalities (e.g., data input, report generation, administrative tasks).
   - Characteristics: Each page follows consistent design principles and layout guidelines. Standard buttons for common functions such as saving, deleting, and printing. Error message display in case of invalid inputs or system errors.
5. GUI Standards and Style Guides:
   - The user interface design adheres to established GUI standards and product family style guides for consistency and usability.
   - Sample screen images and layout constraints are provided in the User Interface Specification document.
6. Error Message Display:

- Error messages are displayed prominently and clearly, indicating the nature of the error and suggesting possible solutions.

3.2 Hardware Interfaces

1. Supported Device Types:
   - The Aviation Safety System is a desktop application installed on local systems, including desktop computers and laptops.
2. Nature of Data and Control Interactions:
   - Data Interactions: The application interacts with a cloud database to retrieve, store, and manage data related to flight incidents, safety ratings, and user information. This includes querying database records, updating data entries, and synchronising local data with the cloud.
   - Control Interactions: The application controls data transmission and synchronisation processes between the local system and the cloud database. This involves initiating data transfers, managing database connections, and handling error conditions.

3.3 Software Interfaces

1. Database:
   - Database Management System: Oracle (version 21c)
   - Purpose: The Aviation Safety System interacts with the Oracle database to store, retrieve, and manage data related to flight incidents, safety ratings, user information, and system configurations.
   - Data Items: Includes flight data, safety ratings, user credentials, and system settings.
   - Messages: Involves SQL queries for data retrieval, insertion, and updates.
   - Communication: JDBC (Java Database Connectivity) API for Java applications or ADO.NET (ActiveX Data Objects for. NET) for .NET applications.
   - Referenced Documents: The Oracle Database Documentation for detailed API protocols and database schema.
2. Operating System:
   - Supported Operating Systems: Windows (versions 7, 8, 10)
   - Purpose: The Aviation Safety System runs on the supported operating systems to provide users with a platform for accessing and using the application.
3. Tools and Libraries:
   - Font Awesome for C# (version 6.3) ML.NET Framework (version 3.0)
   - Purpose: Font Awesome for C# is used for iconography within the user interface, while the ML.NET framework provides machine learning capabilities for predictive analysis tasks.
   - Communication: Integration of library functions and APIs within the application codebase.
   - Referenced Documents: The Font Awesome Documentation and ML.NET Documentation for usage guidelines and API protocols.
4. Shared Data:
   - Flight data, safety ratings, user credentials, and system configurations are shared between users.
   - Implementation Constraint: Secure communication protocols are mandated for transmitting sensitive data between the Aviation Safety System and external software components, ensuring data confidentiality and integrity.

3.4 Communications Interfaces

1. Local Database Communication:
   - Purpose: The Aviation Safety System communicates with a local database for data storage, retrieval, and management.

- Message Formatting: Structured query language (SQL) for database queries and commands.
- Communication Standards: Direct database connections using JDBC (Java Database Connectivity) for Java applications or ADO.NET (ActiveX Data Objects for . NET) for .NET applications.
- Security and Encryption: Database access controls and encryption mechanisms are implemented at the database level to ensure data security.

2. Internal Application Communication:

- Purpose: Internal communication within the Aviation Safety System for data processing, system events, and user interactions.
- Message Formatting: Structured data formats (e.g., JSON, XML) for inter-component communication.

4. System Features

4.1 Safety Rating Prediction

4.1.1 Description and Priority

This feature uses historical flight data and predictive analysis techniques to forecast the safety rating of future flights. By analysing weather conditions, aircraft type, and past incidents, the system predicts the safety rating of upcoming flights, providing passengers with valuable information to make informed travel decisions.

Priority: High

Priority Component Ratings:

- Benefit: 9
- Penalty: 7
- Cost: 8
- Risk: 8

Benefit: Predicting safety ratings for future flights provides passengers with crucial information to ensure their safety and make informed travel choices. This feature enhances the overall safety assurance provided by the Aviation Safety System.
Penalty: If the safety rating predictions are inaccurate or unreliable, it could undermine passenger trust in the system and lead to dissatisfaction. Ensuring the accuracy and reliability of forecasts is essential to mitigate potential penalties.
Cost: Developing and implementing predictive analysis algorithms, as well as integrating them into the system, may require significant resources and investment in terms of time, expertise, and technology.
Risk: There is a risk of inaccuracies in safety rating predictions, which could lead to incorrect assessments of flight safety and potential safety incidents. Additionally, the complexity of predictive analysis algorithms may introduce technical challenges and risks during implementation.

4.1.2 Stimulus/Response Sequences

1. The user inputs flight details such as departure location, destination, date, and aircraft type. The system validates the input data and prepares it for predictive analysis.

2. The user initiates the safety rating prediction process. The system triggers the predictive analysis algorithm to analyse historical flight data and predict the safety rating for the specified flight.

3. The user waits for the prediction results. The system performs the predictive analysis and predicts the specified flight's safety rating.

4. The user views the safety rating prediction displayed on the system interface. The system presents the safety rating prediction and relevant information, such as factors considered in the analysis and confidence level.

5. The user assesses the safety rating prediction and decides whether to book the flight. The system provides options for further actions, such as securing the flight, viewing additional details, or exploring alternative flight options.

4.1.3 Functional Requirements

REQ-1: The system shall integrate with external datasets from Kaggle to access historical flight data for predictive analysis.

REQ-2: The system shall utilise NumPy.NET and ML.NET decision tree algorithms to perform predictive analysis on historical flight data and forecast safety ratings for future flights.

REQ-3: The system shall preprocess and clean the dataset from Kaggle to ensure data integrity and accuracy for predictive analysis.

REQ-4: In case of an invalid or incomplete dataset from Kaggle, the system shall log an error and notify the administrator for manual intervention.

REQ-5: The system shall manage missing or incomplete data points within the dataset by employing suitable data imputation techniques to ensure accurate predictive analysis results.

REQ-6: If the ML.NET decision tree algorithms encounter errors during predictive analysis, the system shall log the error details and provide an error message to the user indicating the issue.

REQ-7: The system shall provide users with a confidence level or probability score associated with each safety rating prediction, indicating the reliability of the forecasted rating.

REQ-8: The system shall allow users to input flight details, such as departure location, maximum elevation, date, and aircraft type, for safety rating prediction.

REQ-9: The system shall validate user-input flight details to ensure completeness and accuracy before initiating the safety rating prediction process.

REQ-10: The system shall present the safety rating prediction results to the user in a clear and understandable format.

4.2 Statistics of past flights

4.1.1 Description and Priority

This feature involves presenting statistical information and insights derived from the analysis of past flight data. Users can access various metrics and trends related to previous flights, such as accident rates, aircraft performance, weather patterns, and safety ratings. Displaying these stats enables users to understand historical flight conditions better and make informed decisions about their travel plans.

Priority: High

Priority Component Ratings:

- Benefit: 7
- Penalty: 5
- Cost: 6
- Risk: 6

Benefit: Providing users with access to statistics of previous flights allows them to assess historical safety performance and make informed decisions about future travel, enhancing overall safety awareness and confidence.
Penalty: Inaccurate or misleading statistics could lead to misinterpretation and incorrect assumptions about flight safety, potentially undermining user trust in the system.
Cost: Implementing the feature to display stats of previous flights may require moderate resources in terms of development time, data processing, and system integration.
Risk: There is a risk of presenting biased or incomplete statistics if the data analysis process is not thorough or if there are limitations in the availability or quality of historical flight data. Ensuring data accuracy and reliability is crucial to mitigate this risk.

4.1.2 Stimulus/Response Sequences

The system presents the selected statistical metrics clearly and organised, using charts, graphs, or tables for visualisation. The user can view various statistical metrics, such as accident rates, safety ratings, weather patterns, and aircraft performance.

4.1.3 Functional Requirements

REQ-1: The system shall integrate with external datasets from Kaggle to access historical flight data for statistical analysis.

REQ-2: The system shall preprocess and clean the dataset from Kaggle to ensure data integrity and accuracy for statistical analysis.

REQ-3: The system shall filter statistical data based on parameters such as date range, aircraft type, location, and other relevant factors.

REQ-4: The system shall calculate and present various statistical metrics related to previous flights, including but not limited to accident rates, safety ratings, weather patterns, and aircraft performance.

REQ-5: The system shall use appropriate data visualisation techniques, such as charts, graphs, or tables, to present the statistical metrics clearly and understandably.

REQ-6: The system shall dynamically update the displayed statistics in real time based on user interactions and adjustments to the parameter filters.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. Real-time Interaction Performance:
   - Real-time responsiveness to user interactions enhances user engagement and facilitates interactive statistical data exploration. Users expect smooth and immediate feedback when interacting with the stats display interface.
2. Dynamic Update Performance:
   - Users should not experience delays or lags when viewing updated statistical metrics.
3. Visualization Rendering Performance:
   - The system shall render data visualisations, such as charts, graphs, or tables, with smooth transitions and minimal loading time. Users expect data visualisations to load quickly and display seamlessly. Smooth and fast rendering of visualisations enhances the user experience by providing a visually appealing and responsive interface.

5.2 Safety Requirements

1. Risk Mitigation for Incorrect Safety Ratings
   - The system shall provide clear disclaimers to users that safety ratings are predictions based on historical data and are not guarantees of future safety.
   - Implement prominent warnings and educational materials within the user interface to inform users about the limitations and uncertainties associated with safety ratings.
   - Adhere to aviation regulatory guidelines regarding presenting safety information to passengers and travellers.
   - Compliance with safety communication standards specified by relevant aviation authorities.
2. Prevention of Misleading Information
   - The system shall ensure that statistical information presented to users is accurate, relevant, and unbiased to prevent misinformation or misinterpretation that could lead to safety concerns.
   - Implement data validation and verification processes to detect and correct inaccuracies or biases in statistical data.
   - Adherence to aviation safety regulations governing the accuracy and transparency of safety-related information provided to passengers and travellers.

5.3 Security Requirements

- The system shall protect user privacy and confidential information, such as personal data and travel itineraries, from unauthorised access, disclosure, or misuse.
- Implement robust data encryption, access controls, and authentication mechanisms to safeguard user information against unauthorised access or breaches.
- The system must satisfy relevant aviation safety regulatory authorities' data privacy and security requirements.

5.4 Software Quality Attributes

1. Reliability
   - Requirement: The system shall maintain reliability ensuring minimal downtime and service interruptions.

- Verification: Conduct regular reliability testing and track system uptime to validate the reliability metric.

2. Usability
   - Requirement: The system shall achieve user satisfaction, indicating high user acceptance and ease of use.
   - Verification: Conduct usability testing with representative users to evaluate user satisfaction and usability metrics.
3. Testability
   - Requirement: The system shall have built-in testability features and comprehensive test suites to facilitate thorough testing and debugging.
   - Verification: Evaluate the effectiveness of test suites in detecting and resolving software defects through testing coverage and defect resolution metrics.

5.5 Business Rules

1. User Roles and Permissions
   - Only registered users with valid credentials can access the Aviation Safety System.
   - Administrators have the authority to add, delete, or update information in the system database, while passengers can only view and search for flight data.
   - Passengers must log in to access personalised features like viewing safety ratings for specific flights or searching historical flight data.
2. Data Access and Privacy
   - Access to sensitive data, such as personal passenger information or detailed flight records, is restricted to authorised personnel only.
   - Passenger data privacy is strictly enforced, and access to personal information is limited to essential system functionalities.
   - System administrators are responsible for ensuring compliance with data privacy regulations and safeguarding user information from unauthorised access or misuse.
3. Error Handling and Notification
   - The system automatically logs errors and irregularities during data processing or analysis for administrative review and resolution.
   - Users are notified of any system errors or disruptions that may affect their ability to access or interact with the system, along with instructions for troubleshooting or reporting issues.
4. System Maintenance and Updates
   - Regular system maintenance and updates are scheduled during off-peak hours to minimise user access and system availability disruption.
   - Administrators coordinate and implement system updates, ensuring minimal downtime and seamless transition to new software versions or features.
5. Compliance with Regulations
   - The Aviation Safety System must comply with relevant aviation safety regulations, data privacy laws, and industry standards governing flight-related information collection, storage, and dissemination.
   - Compliance with regulatory requirements is the responsibility of system administrators, who must always ensure that the system adheres to applicable laws and guidelines.

6. Other Requirements

Currently all requirements have been explicitly explained in the previous sections.