# AVIATION INDEX
# PROJECT REQUIREMENTS

Jesse Dalton
CST 452
Grand Canyon University
Professor Scott Webster

Revision: 2.1

June 29, 2025

# ABSTRACT

Aviation Index is a full-stack, web-based study platform created for pilots who need a fast, structured way to master aviation knowledge. The project aimed to turn a sprawling collection of scattered notes and flashcards into a simple, organized, and concise application. The application organizes aviation information into Topics and Questions and tracks each user's progress as they study the material. From a user's perspective, it feels like an adaptive question bank.

The application is built using the Spring Boot framework with session-based authentication. Data is persisted using a MySQL database. Client web pages are generated using the Thymleaf Templating engine. Key features include secure password management with BCrypt, a user-friendly library of aviation-specific information, enum-driven study-status tracking, and tailored filters for managing session content.

The entire software development process included phases for design, implementation, testing, and deployment. Design plans included user stories, a complete requirements list, a comprehensive data dictionary, a low-level architecture design, and detailed implementation and testing plans. Implementation was carried out according to the design plans and guided by the requirements list with testability in mind. To track project success, a traceability matrix was used to map requirements to final project artifacts and tests.

# FUNCTIONAL REQUIREMENTS

*** See Functional Requirements Document (separate)

## NON-FUNCTIONAL REQUIREMENTS

| ID | Use Case or User Story | Justification |
|---|---|---|
| 1 | The client should be accessible on desktop as well as mobile devices. | Access on multiple devices increases usability and provides uninterrupted sessions. |
| 2 | All user interfaces must be intuitive and easy to understand and use. | Users should be able to create an account and immediately start focusing on studying instead of learning how to use the app. |
| 3 | All aviation information must be current and accurate. | Safety-critical information should always be effectively maintained. |
| 4 | All aviation information must be applicable and comprehensive. | Application content should be meticulously curated to provide the most useful experience for pilots. |

## TECHNICAL REQUIREMENTS

| ID | Use Case or User Story | Justification |
|---|---|---|
| 1 | The application will be developed using the Spring Boot framework. | Spring Boot is a powerful and extensive framework often used for web development. |
| 2 | Security will be managed by Spring Boot Security. | Spring Boot Security is a trusted and safe security option. |
| 3 | Web-facing endpoints will be handled by Spring Controllers. | The application should follow an idiomatic N-tier monolithic architecture. |
| 4 | Business logic will be processed by Spring Services. | The application should follow an idiomatic N-tier monolithic architecture. |
| 5 | Database operations will be managed by Spring JPA Repositories. | The application should follow an idiomatic N-tier monolithic architecture. |
| 6 | Data will be modeled by POJO classes that also map to database entities. | The application should follow an idiomatic N-tier monolithic architecture. |
| 7 | All front-end user interfaces will be generated using the Thymeleaf Templating Engine. | The application should follow an idiomatic N-tier monolithic architecture. |
| 8 | Styles will use basic CSS. | The application will be simple enough not to warrant the need for more dedicated CSS libraries or dialects. |
| 9 | Data will be persisted using a MySQL database. | The application should follow an idiomatic N-tier monolithic architecture. |
| 10 | Initial DDL and DML scripts will be provided to instantiate a MySQL database prior to launch. | The application should follow an idiomatic N-tier monolithic architecture. |