

Airport Management Documentation

*29th June 2025
Midterm Sprint*

Khoa Pham

Sprint Reflection – Airport Management Project

This sprint was a valuable learning experience as I developed both a Spring Boot REST API and a Java-based command-line interface (CLI) application to interact with the API. Working solo, I was responsible for building the full stack from backend services to the frontend CLI, and this helped me understand the entire flow of data and structure in a full application.

What Went Well

- **API Functionality:** I successfully implemented core CRUD operations for all entities—Aircraft, Airport, City, and Passenger. Each entity was tied together through clear relationships, and I structured the project with proper use of controllers, services, and repositories.
- **CLI Integration:** The CLI application communicates smoothly with the API using HttpClient. I implemented functions that answered real-world queries like "Which aircraft has each passenger flown on?" and "What airports do aircraft take off from and land at?".
- **Code Organization:** I kept the code modular and readable. Each class had a clear responsibility, which made debugging and testing easier.

- **Testing & Mocking:** I implemented some unit tests using JUnit and Mockito to ensure CLI output behaved as expected based on mocked API responses.

What Could be Improved

- **Time Management:** I underestimated how long it would take to integrate all the relationships and fully test each feature, especially the CLI interactions.
- **Test Coverage:** While I added some tests for the CLI, I didn't fully test every service or controller method in the API. With more time, I would improve backend test coverage.
- **Error Handling:** The API could use more detailed validation and error messages. I would also like to handle edge cases more gracefully in the CLI.
- **User Experience:** The CLI could be improved with better formatting and possibly added interactivity for editing or filtering results.

What I Would Do Differently

If I were to do this sprint again:

- I would **start testing earlier**, writing tests alongside the feature implementation.
- I would also consider **splitting the sprint into mini-deadlines** to avoid last-minute rushes.

- Lastly, I would try adding a simple **frontend UI** to see how the API performs beyond just CLI.