Purpose

The system design is documented in the System Design Document (SDD). It describes additional design goals set by the software architect, the subsystem decomposition (with UML class diagrams), hardware/software mapping (with UML deployment diagrams), data management, access control, control flow mechanisms, and boundary conditions. The SDD serves as the binding reference document when architecture-level decisions need to be revisited.

Audience

The audience for the SDD includes the system architect and the object designers as well as the project manager.

Table of Contents

Document History

Rev.	Author	Date	Changes

1. Introduction

We used the Java MVC Spring Framework to implement REST architecture.

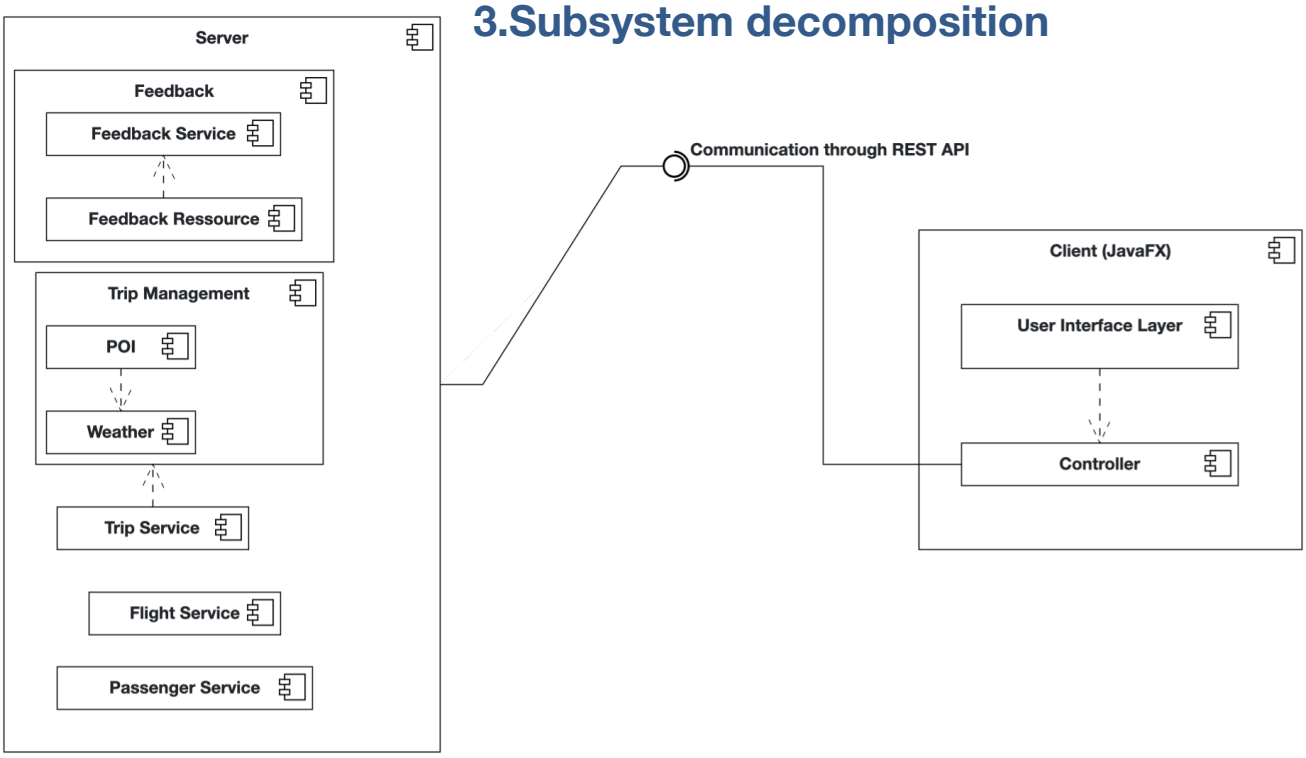
1. Overview

The Model classes are implemented in the package Common which is used by both Server and Client classes. Ressource, as well as Service classes are implemented in the package Server. The Client package contains the View and the Controller classes.

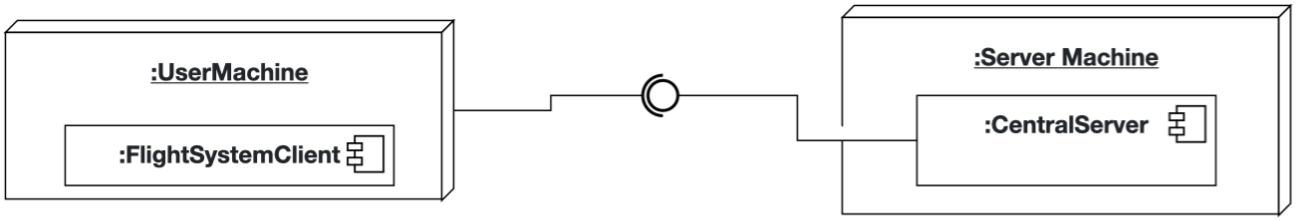
2. Design Goals

We prioritised usability over functionality. Every functionality implemented in the GUI is accessible with less than four clicks.

3.Subsystem decomposition



4. Hardware/software mapping



5. Persistent data management

We mainly used a simple storage scheme, when it came to storing flightIds for trips or passengerIds in a flight. We used ArrayLists for this purpose in the Service classes. Only Service classes can directly operate on the stored data; other classes (mainly Ressource classes) must delegate storage-, find- and update-operations to the corresponding Service classes.

6. Access control and security

As described in Section 6, we implemented access restrictions to storage components, however we did not make use of access matrices or restriction control lists, as the User only communicates with a GUI allowing him/her to access, what should be visible for the User. Other Information are hidden.

7. Global software control

As pictured in the deployment diagram in Section 4, we opted for a centralised control-flow in our project.

8. Boundary conditions

The user has to press a button to start and exit the application. He/she also always has to press corresponding buttons to add a feedback/view security instructions..