Technische Universität München

Teamproject

EIST SoSe 22

42er

Flightsystem

Submitted by:

Carlo Bortolan

Fabian Fritz

Corrected by:

Alexandra Cara Marquardt

Carlo Bortolan
Fabian Fritz

FLIGHTSYSTEM 42

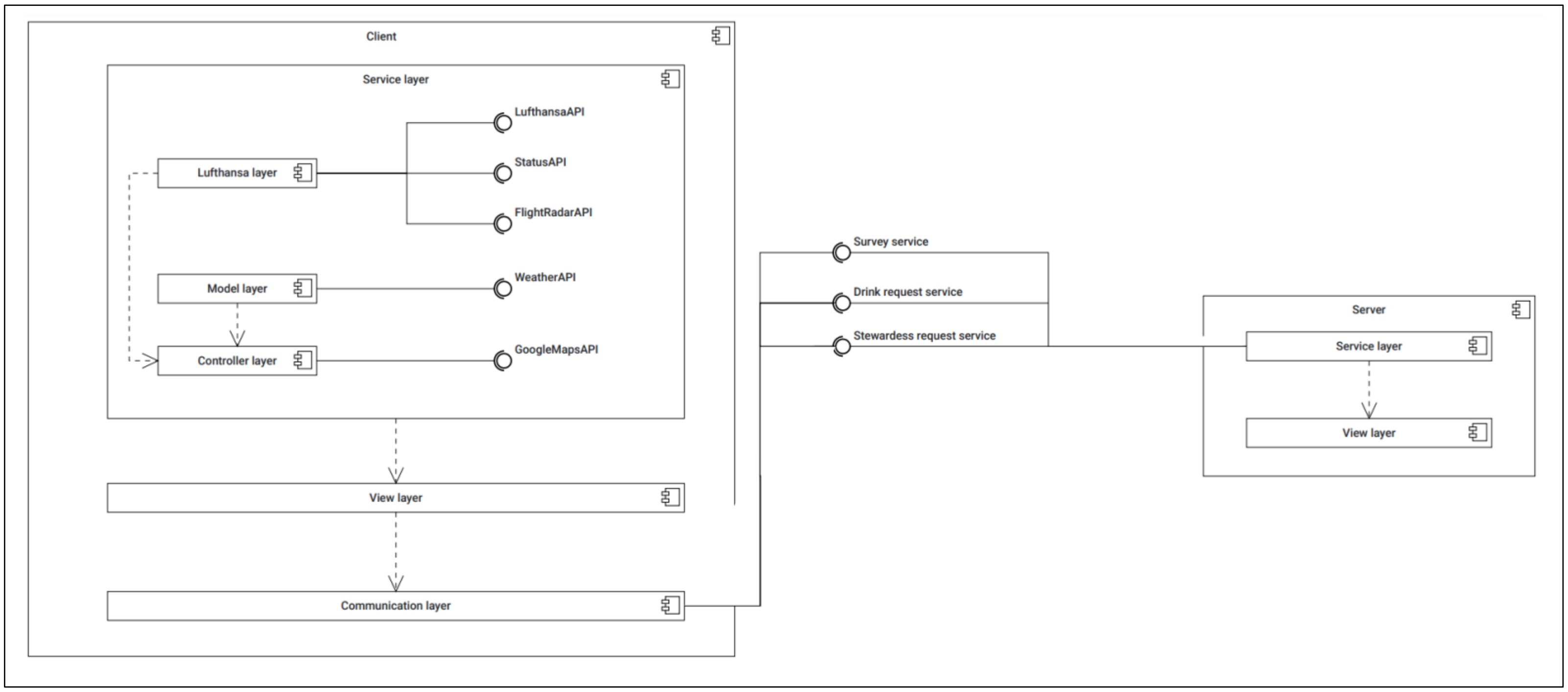
1. Introduction

The purpose of the Introduction is to provide a brief overview of the software architecture. It also provides references to other documents.

2. Design Goals

This section describes the design goals and their prioritization (e.g. usability over extensibility). These are additional nonfunctional requirements that are of interest to the developers. Any trade-offs between design goals (e.g., usability vs. functionality, build vs. buy, memory space vs. response time), and the rationale behind the specific solution should be described in this section. Also the rationale of all other decisions must be consistent with described design goals.

3. Subsystem decomposition



4. Hardware/software mapping

This section describes how the subsystems are mapped onto existing hardware and software components. A UML deployment diagram accompanies the description. The existing components are often off-the-shelf components. If the components are distributed on different nodes, the network infrastructure and the protocols are also described.

5. Persistent data management

- *unavailable*

6. Access control and security

- Public getters and setters, private attributes
- UI FXML methods mostly private
- The low coupling – high cohesion principle is followed as the classes mostly interact with other classes of the same package and only in rare cases delegate method calls to classes that are contained in other packages.

7. Global software control

Besides the obvious use of multithreading inside the given UI components, all processes are executed sequentially.

8. Boundary conditions

To start the FlightSystem App you only need to run the main-method of the class FlightSystem.java (main/java/com/example/eist22t02zweiundvierziger2022/FlightSystem.java). This will automatically start the app and the server. The loading of the app should normally take from 5 up to 90 seconds, but may vary depending on your device (and if you have just cloned it or already started it once before).

The Spring Boot server has three relevant windows that you can access by copying and pasting the respective link in a browser. (In case you change the client/server port you will have to update the link)

View submitted surveys (http://localhost:8080/Surveys)
View drink orders (http://localhost:8080/Drinks)
View current in-flight (requests http://localhost:8080/Requests)
The server application runs on the default port (8080). In case this port is already in use you can change the port:

First by updating the client port:

- replace “.baseUrl("http://localhost:8081/")” with “.baseUrl("http://localhost:<newHost>/")” at line 36 of the server/Client.java class

Secondly by changing the port of the application:

- Go to “FlightSystem.java” [RUN] -> [Edit Configurations ...] -> [Modify options] -> [Add VM options] -> input “-Dserver.port=<newHost>” in the VM Options field.