Student: IEREMIAȘ Viorel

Group: 244

Assignment no.: 1

Assignment topic: Servlets

**Requirement**

*“Tema Servlet presupune instalarea şi configurarea containerelor servlet Tomcat si Jetty si a serverelor de aplicatii (AS) WildFly si GlassFish. Realizarea unei aplicaţii simple dar relevante ce foloseste servleturi. Sunt necesare minimum două servleturi care să comunice între ele, eventual unul pe post de dispecer. Aplicatia va trebui sa fie portabila / instalabila pe oricare dintre cele patru containere de mai sus si prin toate modalitatile de deploy: arhive war pe cele patru AS; în plus, pentru Tomcat şi Jetty deploy cu context extern (fără war) şi aplicaţii embedded. Realizarea presupune inclusiv download-ul distributiilor corespunzătoare şi instalarea acestora.”*

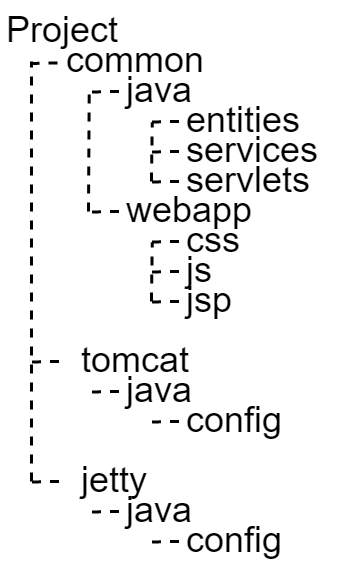
**Proposed solution**

A flight management and tracking system. The main functionality consists in displaying information about flights. The filtering capacity allows a refined search, that meets the criteria relevant for the user: departure and arrival location.

The filter options are dynamically set, i.e. after the user selects the desired destination country, the selection tool for the destination city in the filter area is automatically filled with the available destination cities in the country and nothing more.

This feature improves on the one hand the experience of the user, and on the other hand, the ability of the system to integrate the constant changes that appear in flight routes.

**Architecture**

Despite the need to generate build artifacts for different deployment environments, the *business cod*e of the application should not be duplicated. With this in mind, the **common** submodule contains exactly the before mentioned part – the Java code and the web application resources that make up the functionality. The additional **tomcat** and **jetty** submodules only contain Java configuration classes specific to the Tomcat, respectively Jetty servlet container, required for the embedded Jar deployment of the application.

**Gradle** is used as buildsystem for its flexibilty in configuring custom build pipelines. The **war** deployment is built from the **common** submodule. The embedded **jar** deployments are build from the specific modules, both of which include the **common** submodule as dependency.

**Installation**

The build script at the root level of the projects coordinates the execution of the separate builds and collects the artifacts at the root-level **build/libs** directory. The build is initiated by dispatching **gradle assigBuildAll** in the root of the project.