# Chapter 3.

# MAIN APPLICATION AND USER PORTAL DESIGN

## Description of architecture in process of development.

The following system modules has to be developed and defined (Fig.3.1.):

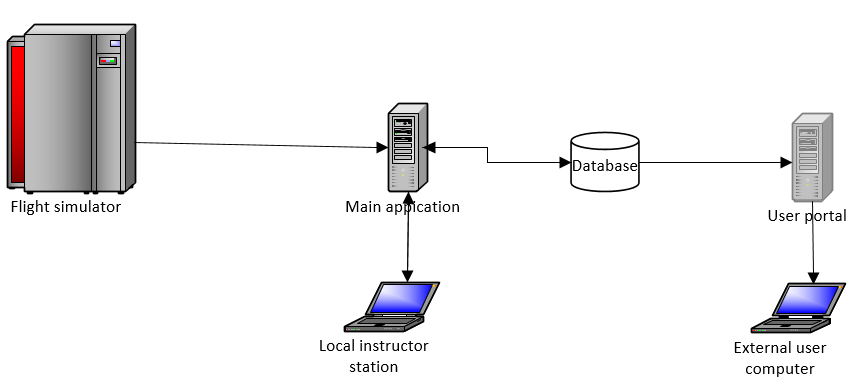
1. Flight simulator – system that provides flight simulation, gathering data and data transition.
2. Main application server – server with deployed main application WAR file and possibility to connect to database.
3. Local station or laptop – station or laptop with active connection to local network and connection to main application.
4. Database – contains data of all entities and simulation record values.
5. User portal server – server with deployed user portal WAR file and possibility to receive data from database.
6. Common user laptop or computer – computer with internet connection and internet browser.

Fig.3.1. Structure architecture diagram

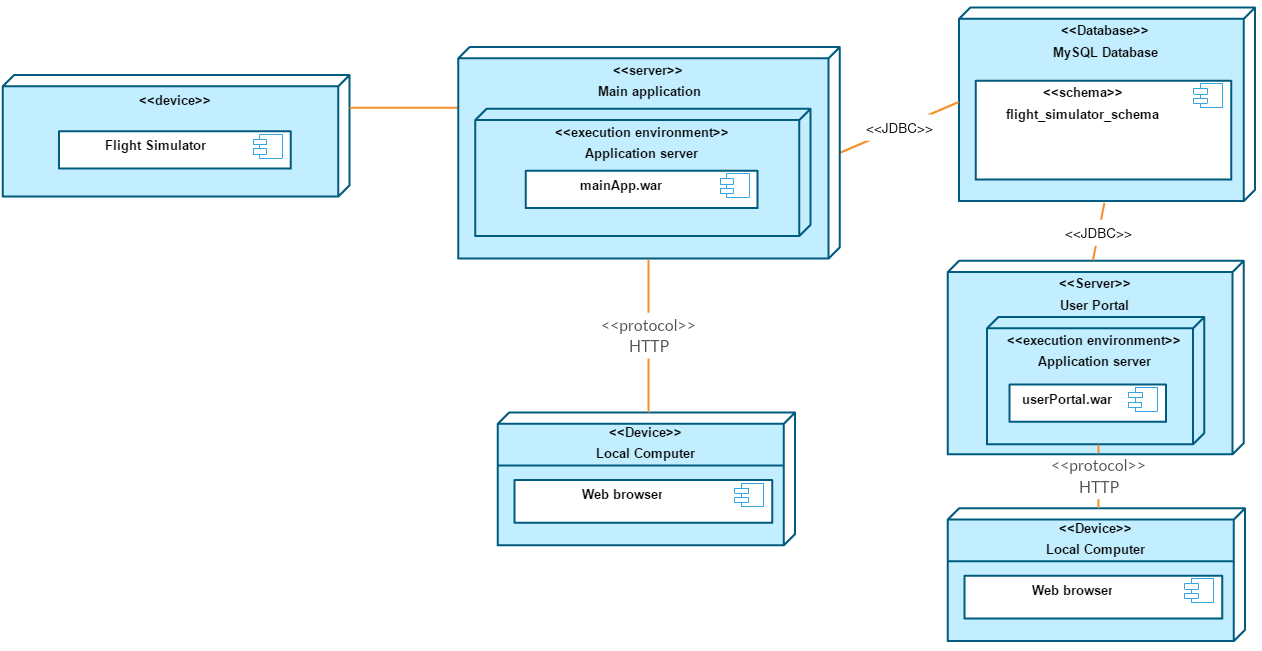


Fig.3.2. Deployment diagram of the system

## Database design.

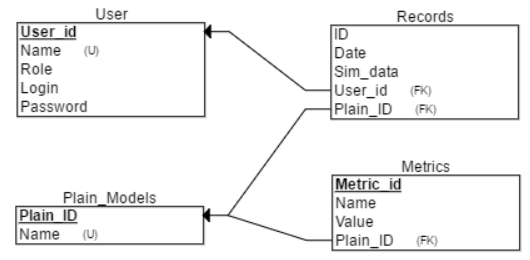
After investigation were ensured that database should store users accounts, plain models description, metrics for each plain model and records from flight simulation. Records have to have relative connection with plain model and user names. Metrics have relation connection with plain model.

Fig.3.3. ER model of database.

## Description of packages of main application.

A Java package organizes Java classes into namespaces, providing a unique namespace for each type it contains. Classes in the same package can access each other's package-private and protected members. Java packages can be stored in compressed files called JAR files, allowing classes to be downloaded faster as groups rather than individually.

In general, a package can contain the following kinds of types: classes, interfaces, enumerations, and annotation types. A package allows a developer to group classes (and interfaces) together. These classes will all be related in some way – they might all have to do with a specific application or perform a specific set of tasks. Programmers also typically use packages to organize classes belonging to the same category or providing similar functionality. In case of main application of system will be created Package will consist Users entity, Plain\_model entity, Metric entity, Records entity and one enum to store roles in String type.

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

### Entity package.

Entity – is a POJO-class, connected with database with help of annotation @Entity or with XML. This kind of class is required to have current requirements:

* 1. Must have empty constructor.
  2. Cannot be inner class, interface or enum.
  3. Cannot be final and contain final fields.
  4. Must contain @Id field.

Entity can:

* 1. Contain non-empty constructors.
  2. Inherit or be inherited.
  3. Contain other methods and implement interfaces.

Entities can be related with each other with one-to-one, one-to-many and many-to-many relations.

### 3.3.2. Class diagram for “entity” package.

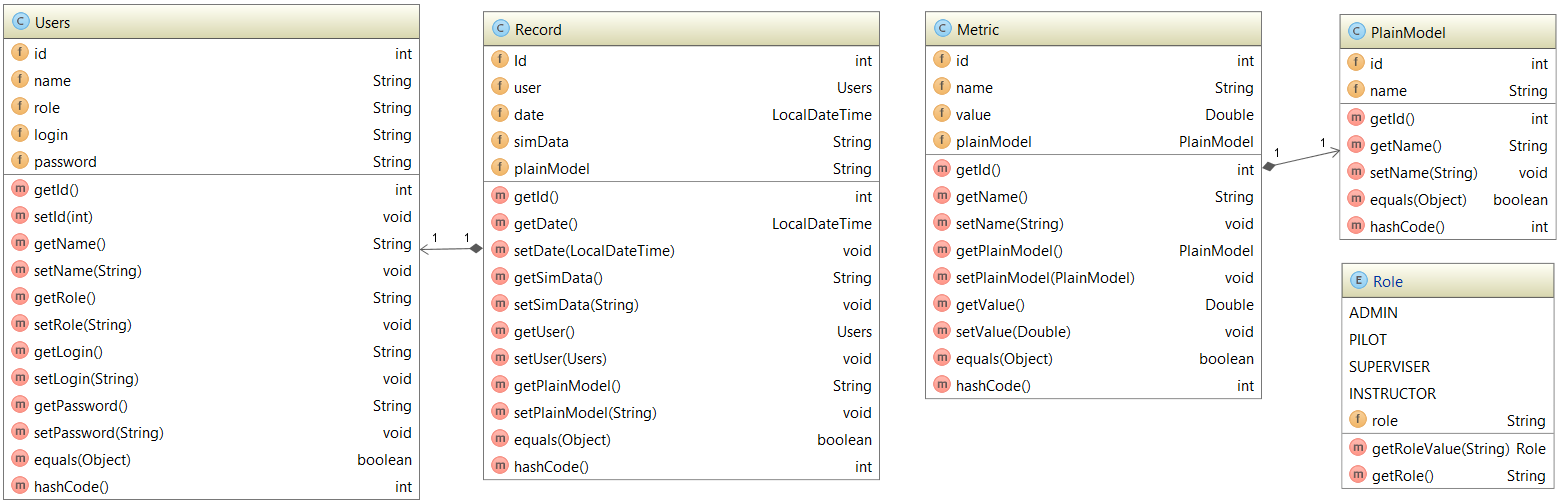


Fig.3.4. Class diagram for entity package.

### Class diagram for “DAOImplementation” package.

Fig.3.5. Class diagram for implementation classes of DAOs.

### Class diagram for “rest”.

Fig.3.6. Class diagram for rest package with socket listener.

## Graphical user interface design.

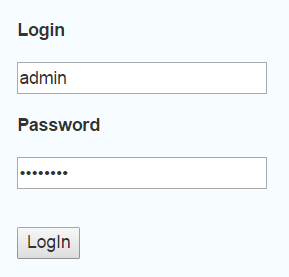
Initial page of main application is login page. With granted credentials user can sign up with different rights. The login page is represented on Fig.3.7.

Fig.3.7. Login page

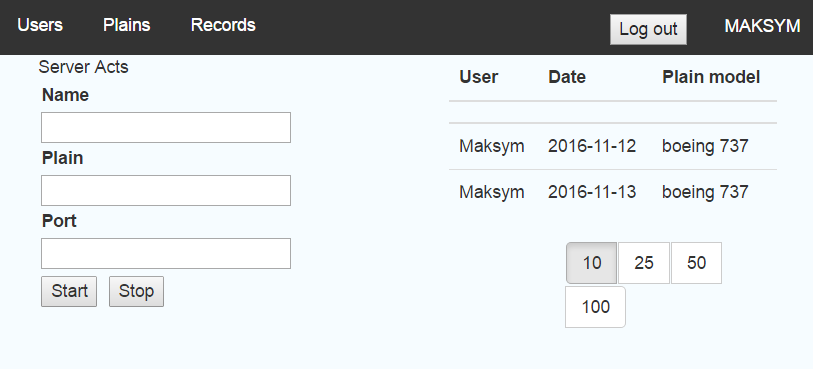
After that user will be relocated on main page with listening process initiation configuration and start-stop buttons. It is necessary to input name of pilot, plain model and port for which servler will be listening simulator data input. On the right side will be visualized list of last simulation records (Fig.3.8.).

Fig.3.8. Main page.

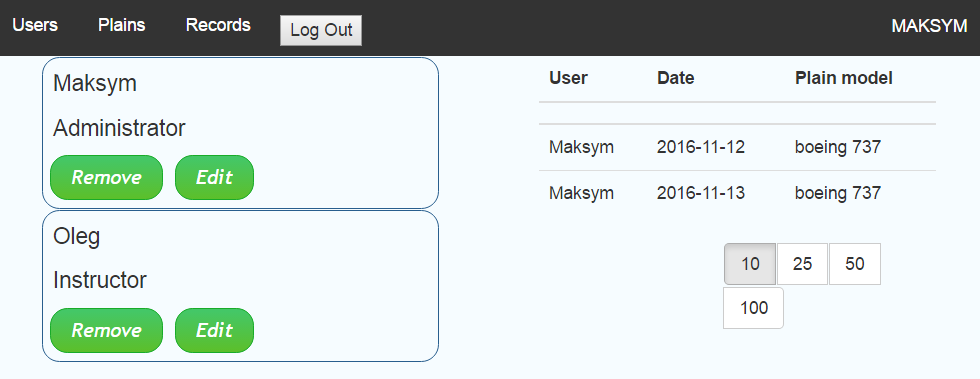
User can brows main application with menu on the top of page. First item in menu is Users page. Clicking on User item the table with records for this peculiar user will be opened (Fig.3.9.). For dispaing tables will be used AngularJS factory ngTable. In angularjs if we want to bind data to tables and implement functionalities like sorting, paging and filtering it’s better to use ng-table module. By using ng-table module in angularjs applications we can achieve functionalities like showing data in table format, sorting, filtering and paging without writing much code.

Fig.3.9. Users page

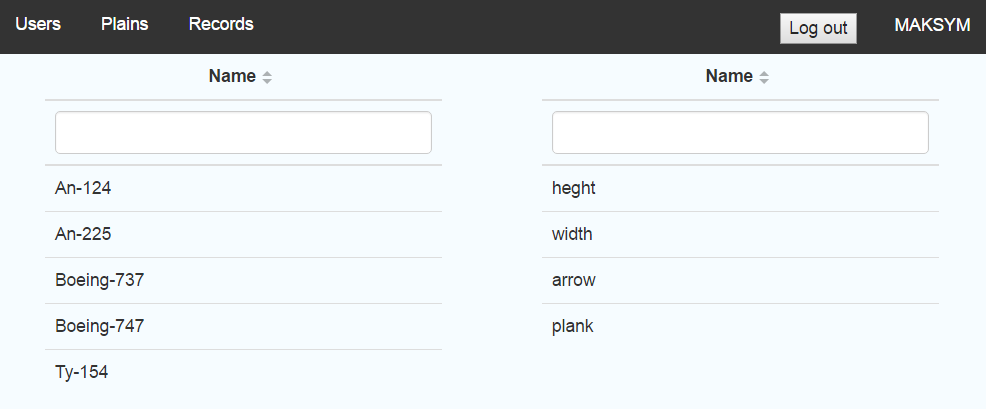
Next page is Plains. The list of plain models will be visualized in table view and if plain model item will be clicked the plain model metrics will be viewed alongside it, like was designed according to the requirements specification (Fig.3.10).

Fig.3.10. Plains page