

# **Program systems development**

## **Compulsory programs 2018**

### 1. Medical data handler

- Create a NodeJS server, which is able to store the doctors and patients with different rights. There are patients ordered to the doctors and there are measurements and treatments ordered to the patients.
- Create an Angular 2 browser app, which makes available a login functionality both for doctors and patients. The doctors can select the proper patient from their list and can check the uploaded data on different dashboards (e.g.: blood pressure, weight, etc...). The patients get a simple environment after login. This view is used to check their earlier uploads and send new data to the server.

### 2. Quiz game

- Create a NodeJS server, which is able to store users as administrators and players (the players can have scores). Define arbitrarily long quiz series where all the correct answers are defined.
- Create an Angular 2 browser app where just the guests can sign up. If someone log in as an admin, he/she can create new questions, and delete the created ones. The registered players can play and can earn points. From the points you have to create a toplist which can be seen by both the admins and the users.

### 3. Hotel booking system

- Create a NodeJS server, which stores the informations about the hotels with the number of the free rooms.
- Create an Angular 2 browser app which makes available a login site for the admins in order to query all the hotels and their informations. The admins can add new hotels or modify the informations. The users can query the available hotels and book rooms.

### 4. Forum

- Create a NodeJS server that stores the users, the topics and the posts. There is an admin user who can work as a moderator (moderate the users, the topics and the posts). The users can delete nor topics, neither posts. They can modify only their own posts.
- Create an Angular 2 browser app that represents the forum and has login and register sites. Only the registered users can create new topics, but the unregistered users can also send new posts (as a guest). There are LIKE and DISLIKE buttons below each posts and the number of likes and dislikes are shown next to the posts.

#### 5. Exchange (bourse) simulator

- Create a NodeJS server, which stores the users (there are no admins) and the users have capitals and stock exchanges. On the server execute a value changing script with a timer. The script can create new share, changes the existing shares' value (use tendencies, not just random numbers). A share has current value, name, type (or category).
- Create an Angular 2 browser app which shows the logged in user's balance, the statuses of the shares, the tendencies and the new and the best shares. The client should refresh once in every 30 seconds in order to communicate up-to-date data. Extend the functionalities with share selling and buying functions.

#### 6. Like collector

- Create a NodeJS server which stores entities and all of the entities have a like number, an id and a picture (pictures can be stored with GridFS in MongoDB). The pictures should have similar sizes (not exactly the same).
- Create a Cordova app that queries two random pictures from the server and shows them. The user can select the proper picture by tapping and the tapped picture's entity earns one like. The pictures are coming till the user does not quit the application.

#### 7. Own idea

- The own ideas should be send with their detailed specifications to the teacher.
- The own ideas should contain a Backend and a Frontend part
  - Backend: NodeJS server
  - Frontend: Angular 2 or Cordova