Assignment 1		Project Summary		
Course		Full-stack Development with Node.js and React.js - 2016		
Project author				
№	First name, last name		E-mail	
1	Miroslav Mirchev		mirchev1977@yahoo.com	

Project name	Online School System (OSS)

1.Short project description (Business needs and system features)

The Online School System is an excellent opportunity for students to receive distance education online.

There are various subject which are provided by the school system rendered via video recordings of easily digestible information chunks in the form of main topics and sub topics.

Teachers can create courses on different subjects and provide materials for them. Teachers can also create test materials in the form of quizzes and code challenges to make it possible for the students to follow their own progress.

By each test passed, the student receives points and rating respectively.

If anonymous, the student has the opportunity to access the free of charge courses, get informed about them, and review their materials, pass the tests, but not gain points and respectively get rated between the rest of the students. Registration gives students opportunity to participate in the paid courses and follow their own progress by gaining points and be rated among the other course participants.

Points can be awarded only once for course test passed. If students decide to review a course material in order to consolidate the material inside and pass a test again, they gain no points.

The students get access to the courses and their materials via a dash board where they can register and unregister to various courses, be displayed information about courses, materials, points and ratings.

Teachers and administrators have also their dashboard where they can administer students, update materials and give support to users about their accounts.

OSS has also administrators of the school management who can manage the online system and its users.

The system will be developed as a *Single Page Application (SPA)* using **React.js** as front-end, and **Node.js + hapi** as backend technologies.

Main User Roles

Anonimous Student

Can view the courses information pages and participate in the free courses with no possibility for following progress, gaining points for tests passed and be rated among the other students.

Registered User

Administrator

Extends Registered User. Can manage (create, edit user data and delete) all *Registered Users*, as well as *create courses, course materials and tests*.

Teacher

Extends Registered User. Can create courses, materials to them and tests.

Student

Extends Registered User. Can subscribe to various courses, follow their own progress, gain points for passing tests and be rated among the other course participants. They gain no points for passing a test more than once.

2.Main Use Cases / Scenarios		
Use case name	**Brief Descriptions**	**Actors Involved**
2.1Browse all courses courses, learn from free courses and do tests, but not follow progress and be rated	The *User* can browse the information views (Home, Courses, About) in OSS SPA, and can learn from free courses and do tests, but no follow progress and be rated.	All users
2.2 Register	*Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. By default all new registered users have *Student* role. *Administrator* can register new users by entering *User Data* and choosing a Role (*Student, Teacher*, or *Administrator*).	*Anonymous User, Administrator*
2.3 Change Data	*Registered User* can view and edit her personal *User Data*. *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Student, Teacher*, or *Administrator*.	*Registered User, Administrator*
2.4 Manage Users	*Administrator* can browse and filter users based on different criteria: first and last name, email, Role. *Administrator* can choose a *User* to manage, and can manage the chosen User - edit, or delete. *Administrator* can create a new user.	*Administrator*
2.5 Browse Courses and Points	Students can browse the courses they have completed and points gained for each of them. Teachers can browse courses they have created.	*Student, Teacher*
2.6 Complete Course / Topic	Registered Student or Instructor browses available courses and materials and can sort and filter them using different metadata fields. Student or Teacher chooses a course / material to complete. OSS presents the Course materials and tests one by one to the Student or Teacher, who follow them. On Course / Material / Test completion the results are saved by OSS.	*Student, Teacher*

2.7 **Add / Edit course / material**	Teacher or Administrator specifies/edits Course / Material meta-data such as: name, subject, description, and tags. Teacher or Administrator can create/edit or delete specific materials supplied to course.	*Teacher, Administrator*
2.8 **Manage Courses**	Instructor can browse their Courses and course materials, add new Courses and materials to them using Add/Edit at the dash board, and delete a Course / Material, as well as view the Students' progress. Administrator can browse courses of all Teachers, edit and delete them.	*Teacher, Administrator*

3.Main Views (SPA Frontend)		
View name	**Brief Descriptions**	**URI**
3.1 **Home**	Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register.	*/* */user*
3.2.**Courses** 3.2 .1 **Available Courses**	Presents courses available according to *User's Role* and identity. Offers abilities to create, read, update, delete (CRUD), as well as browse *Courses. Display Points Gain in individual courses.*	*/courses* */user/courses*
3.2.2 **Completed Courses**	Display User's completed Courses and points.	*/user/courses/completed*
3.2.3 **Courses in Progress**	Display User's Courses in progres and points.	*/user/courses/inprogress*
3.3 **User Data** 3.3.1 **Personal Data**	Presents ability to view and edit personal *User Data*, as well as deregister from *OSS*.	*/user/personal*
3.3.2 **Rating**	Display User's points and rating in regard to the rest of the course partakers	*/user/rating*
3.4 **Users**	Presents ability to manage (CRUD) Users and their User Data (available for Administrators only, as described in UCs).	*/users*
3.5 **About**	Presents information about the *OSS* project and its owner.	*/about*

4.API Resources (Node.js Backend)		
View name	**Brief Descriptions**	**URI**
4.1 Users	GET *User Data* for all users, and POST new *User Data* (Id is auto-filled by *OSS* and modified entity is returned as result from POST request). Available only for *Administrators*.	*api/users*
4.2 User	GET, PUT, DELETE *User Data* for *User* with specified *userId*.	*api/users{userId}*
4.3 Courses	GET courses (according to *User's Role* and identity) and POST new *Course* (Id is auto-filled by *OSS* and modified entity is returned as result from POST request).	*api/courses*
4.4 Course	GET, PUT, DELETE *Course*	*api/courses/{courseId}*
4.5 Course Materials	GET all Course Materials, be able to post new material	*api/courses/ {courseId}/courseMaterials*
4.6 Course Material	*GET, POST, UPDATE and DELETE Specific Course Material*	*api/courses/ {courseId}/courseMaterials/ {courseMaterialId}*
4.7 Material Tests	GET all Material Tests, be able to post new test	*api/courses/ {courseId}/courseMaterials/ {courseMaterialId}/tests*
4.8 Test	GET, POST, UPDATE and DELETE Specific Material Test	*api/courses/ {courseId}/courseMaterials/ {courseMaterialId}/tests/{testId}*