

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2020

Project author		
№	Pseudonym	Face-to-face/ online
1	MB	face-to-face

Project name	ReviewIt
--------------	----------

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

Information has become a very important resource. That is why it is crucial for everyone to have access to it in a structured system. **ReviewIt** is a system for sharing experience and opinion that allows students studying in the Faculty of Mathematics and Informatics to share their thoughts about elective courses. The system will be developed as a *Single Page Application (SPA)* using **React.js** as front-end, and **Node.js + express** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using **React Router**. The backend will be implemented as a **REST/JSON API** using JSON data serialization. There will be also a real-time event streaming from the server to the web client using **Socket.IO** and **Server Sent Events (SSE)/WebSocket** in order to maintain a real-time communication between the users. The main user roles (actors in UML) are:

- *Anonymous User – can only view the information course pages*
- *Student (extends Registered User) – can read, write comments and create course pages*
- *Administrator (extends Registered User) – can manage (create, edit user data and delete) all registered users and course pages.*

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. 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.	Anonymous User
2.2. Browse Course pages	<i>Any Student can browse through the catalog of available courses in the system and read more information and reviews about the course. Reviews can be sorted/filtered by date</i>	<i>Student</i>
2.3. Change User Data	<i>Registered User can view and edit her personal User Data.</i> <i>Administrator can view and edit User Data of all Users.</i>	<i>Student, Administrator</i>
2.4. Suggest new courses	<i>Any Student can suggest a new course to be added in the course catalog. He/she has to provide main course data. Then the administrator can approve or decline the suggestion.</i>	<i>Student</i>
2.5. Review actions	<i>Any Student can add reviews to existing course, like/dislike/report a review.</i>	<i>Student</i>
2.6. Follow courses	Any Student can follow/unfollow a course to get/stop getting notifications when new reviews are added to a course	<i>Student</i>

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.	/
3.2. Registration	Presents a view allowing anonymous users to register to the system.	/register
3.3. Login	Presents a view allowing the users to login	/login
3.4. Courses	Presents a view displaying all available courses in the system with the ability to sort/filter/search for courses	/courses
3.5. Course page	Presents a view displaying main course information, number of students who has	/course/id

	followed the course as well as an option to follow/unfollow the course and view reviews	
3.6. Notifications	Presents a view displaying all notifications about courses the Student has followed.	/notifications
3.7. Dashboard	Presents a view displaying most reviewed courses recently	/dashboard
3.8. Account information	Presents a view displaying user's personal data allowing him/her editing it.	/account
3.9. Course Suggestions	Viewing all course suggestions and presenting the ability to create course/delete suggestion (available for <i>Administrators</i> only)	/course-suggestions

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
4.1. Users	GET <i>User Data</i> for all users, and POST new <i>User Data</i> (Id is auto-filled by OKTS and modified entity is returned as result from POST request). Available only for <i>Administrators</i> .	/api/users
4.2. User	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i>	/api/users/{userId}
4.3. Login	POST <i>User Credentials</i> (e-mail address and password) and receive a valid <i>Security Token</i> to use in subsequent API requests.	/api/login
4.4. Logout	POST a logout request for ending the active session and invalidating the issued <i>Security Token</i> .	/api/logout
4.5. Courses	Get all available courses on the system; POST new course - only administrator	/api/courses
4.6. Course suggestion	Post course information.	/api/suggestions
4.7. Reviews	POST new review; PUT/DELETE own review	/api/courses/{courseId}/reviews /api/courses/{courseId}/reviews/{reviewId}
4.8. Vote up/down/report	PUT existing review	/api/courses/{courseId}/reviews/{reviewId}

