

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2023

Project author		
	Name	FN
	Sonya Nikolova	82213

Project name	University Information System
--------------	-------------------------------

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

In this day and age it is crucial to have easy access to the information regarding your education process. The **University Information System (UIS)** provides the ability for students to easily check their grades, the subjects they are taking, if they have paid their semesters, etc. The system allows teachers to fill in their students grades quicker and keep track of the subjects they have to teach in the current semester. The admins of the system can create new student and teachers accounts, they can also introduce new subjects to the information system and check the statuses of the payments of the students. The system will be developed as a Single Page Application, also known as *SPA* using React.js as frontend, Node.js + express as backend technologies. Each view will have a unique URL and the routing between the different routes will be done using React Router. The backend will be implemented as a REST/JSON API using JSON data serialization. The main user roles will be:

- **User** - registered user without a role assigned
- **Admin (extends User)** - can view the information about the students, the teachers, the students' payment of semestrial fees, create new users - students and teachers alike, create new subjects.
- **Student (extends User)** - can view the information about their grades, subjects, personal information
- **Teacher (extends User)** - can view the information about the subjects they are teaching, the students which are enrolled in their classes

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. Register	The <i>Administator</i> can register new users in the system by providing valid User Data and assigning a role to the newlu created user.	Administrat or
2.2. Change user's data	A <i>User</i> .can view and edit their personal User data. The <i>Administrator</i> can view and edit the data of all the other users and assign them roles.	<i>User</i> , <i>Administrat</i> <i>or</i>
2.3. Manage Users	An <i>Administrator</i> can browse and filter users based on different criteria: first and last name, email, Role. An <i>Administrator</i> can choose a <i>User</i> to manage, and can manage the chosen User - edit or delete. An <i>Administrator</i> can also create a new user.	<i>Administrat</i> <i>or</i>
2.4. Manage Students and their Majors	An <i>Administrator</i> can browse and filter <i>Studens</i> based on different criteria: name of <i>Major</i> , creation date, etc. An <i>Administrator</i> can create a new <i>Major</i> , and assign a <i>Major Key</i> to it so students can be enrolled in it later on. An <i>Administrator</i> can choose a <i>Major</i> to manage, and can add or remove <i>Students</i> to the <i>Major</i> , or delete it if necessary.	<i>Administrat</i> <i>or</i>
2.5. Manage Subjects	A <i>Student</i> can browse the subjects they are taking, enroll in a new one, drop a subject if they wish. They can also view the results of their exams in a given subjects. A <i>Teacher</i> can browse the exam results of students which are taking their subject, edit and delete them. An <i>Administator</i> can browse the subjects and add new ones.	<i>Student</i> , <i>Teacher</i> , <i>Administrat</i> <i>or</i>
2.6. Edit Subject	An <i>Administator</i> or a <i>Teacher</i> specifies and edits the data regarding a subject - such as description, name, number of enrolled students allowed, exam date, allowed Majors and semesters, etc.	<i>Teacher</i> , <i>Administrat</i> <i>or</i>

2.7. Enter students grades	A <i>Teacher</i> can browse their subjects and enter the grades for their students. The UIS presents the students with their grades afterwards.	<i>Teacher</i>
2.8. Check payment status and issue a payment	<i>Students</i> can check their payment status and issue a payment in the system and <i>Administrators</i> can approve the payment (change the status of it)	<i>Student, Administrator</i>
2.9. Propose a subject	<i>Teachers</i> can propose a new subject and wait for the administrators to approve it. If the subject is approved, they will automatically be assigned to it.	<i>Teacher</i>
2.10. Approve a subject	<i>Administrators</i> can approve proposed subjects by <i>Teachers</i> and include them in the subjects list. When approving a subject, the <i>Teacher</i> that has proposed is being assigned to lead it.	<i>Administrator</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. Login	Present the Login form. If login is successful, redirects to Subjects view.	/login
3.2. Subjects	Presents the subjects that the user has depending on their role. Offers abilities to browse, choose, create, read, update, delete (CRUD) Subjects (for <i>Administrators</i> and <i>Teachers</i> only).	/subjects
3.3. Enter grades	Presents a view where a <i>Teacher</i> can enter the grades for a particular subject for the students that are taking it.	/subject-name/ /grades
3.4. Payments	Presents <i>Payments</i> available according to role of the user. A <i>Student</i> can issue a payment from this view and see previous ones. An <i>Administrator</i> can approve or decline a payment.	/payments
3.5. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register in OKTS, as well as to enroll to a <i>Student Group</i> by presenting a <i>Token Key</i> for that group.	/register

3.6. User Data	Presents ability to view and edit personal <i>User Data</i> .	<i>/user-details</i>
3.7. Majors	Presents ability to browse, filter and manage <i>Majors</i> (for <i>Administrators</i> and <i>Teachers</i>), and assign <i>Students</i> to them (for <i>Administrators</i> only), as described in UCs.	<i>/majors</i>
3.8. Users	Presents ability to manage (CRUD) Users and their User Data (available for <i>Administrators</i> only).	<i>/users</i>
3.9. About	Presents information about the UIS project and its owner.	<i>/about</i>

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 <i>UIS</i> and modified entity is returned as result from POST request). Available only for <i>Administrators</i> .	<i>/api/users</i>
4.2. User	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> , according to restrictions described in UCs.	<i>/api/users/{userId}</i>
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.	<i>/api/login</i>
4.4. Logout	POST a logout request for ending the active session with <i>UIS</i> , and invalidating the issued <i>Security Token</i> .	<i>/api/logout</i>
4.5. Majors	GET <i>Majors</i> , and POST new <i>Major</i> (Id is auto-filled by <i>UIS</i> and modified entity is returned as result from POST request), according to <i>User's Role</i> and identity security restrictions.	<i>/api/majors</i>
4.6. Subjects	GET subjects (according to <i>User's Role</i> and identity) and POST new <i>Subject</i> (Id is auto-filled by <i>UIS</i> and modified entity is returned as result from POST request).	<i>/api/subjects</i>
4.7. Subject	GET, PUT, DELETE <i>Subject Data</i> for <i>Subject</i> with specified <i>testId</i>	<i>/api/subjects/{subjectId}</i>
4.8. Payments	GET <i>Payments</i> (according to <i>User's Role</i> and identity), and POST a new <i>payment</i> (Id is	<i>/api/payments</i>

	auto-filled by <i>UIS</i> and modified entity is returned as result from POST request).	
4.9. Grades	GET <i>Grades</i> (according to <i>User's Role</i> and identity)) for <i>Subject</i> with specified <i>subjectId</i> , and POST a new <i>Grade</i> (Id is auto-filled by <i>UIS</i> and modified entity is returned as result from POST request). PUT, DELETE are also available.	<i>/api/subjects/{subjectId}/grades</i>
4.10. Grade for subject	GET , <i>Grade</i> for <i>Subject</i> with specified <i>subjectId</i> and for <i>User</i> with with specified <i>userId</i> .	<i>/api/subjects/{subjectId}/grades/ {userId}</i>