

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

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

Project name	Expenses manager
--------------	------------------

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

Expenses manager is a web application that allows users to add incomes and expenses on their account, it also shows history, statistics and graphs based on the user's expenses. In addition to that it allows users to register, and administrators to manage them. 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. The authentication and db service will be provided by **mongodb**. The main user roles (actors in UML) are:

- *Anonymous User* – can only view the home/login/register page
- *Registered User* – can add incomes, expenses and see his balance and statistics
- *Administrator* – can access the admin panel where he can see information about all users

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors
---------------	--------------------	--------

		Involved
2.1. Browse information	The <i>User</i> can browse the information views (Home, About).	<i>All users</i>
2.2. Register	<p><i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name, and choosing a password.</p> <p><i>Administrator</i> can register new users by entering <i>User Data</i>.</p>	<i>Anonymous user</i>
2.3. Login/ Logout	<i>Registered user /Admin</i> can login and logout from the system.	<i>Registered user</i>
2.4. Manage users	<i>Administrator</i> can browse and filter users based on different criteria and see their current situation.	<i>Administrator</i>
2.5. Modify info	<i>Registered users</i> can view and edit expenses.	<i>Registered user</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. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register in <i>Expenses manaher system</i> .	/register
3.3. Login	Presents a view allowing the users to login.	/login
3.4. Users	Presents ability to manage Users and their User Data (available for <i>Administrators</i> only)	/users
3.5. View profile	Profile viewer.	/profile/{userID}
3.6. Incomes	Presents the registered user's incomes and lets him add new incomes or categories.	/income

3.7. Expenses	Presents the registered user's expenses and lets him add new ones or categories.	/expense
3.8. Balance	Presents the history of the registered user's incomes and expenses in the calendar with chart of the categories he spends money in.	/balance
3.9. My groups	Select a group you participate in. People in groups share goals and do taxes together.	/groups
3.10. Group	Contribute to a group's goals or pay taxes.	/groups/{groupId}
3.11. Kickstarter	Check out different kickstarters and choose to contribute to one.	/kickstarter/{kickstarterID}
3.12. Kickstarters	Browse all public kickstarters.	/kickstarters
3.13. About	Presents information about the <i>Expenses manager</i> project and his owner.	/about

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>Expenses management system</i> 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).	/api/login
4.4. Logout	POST a logout request for ending the active session with <i>the Expenses management system</i> .	/api/logout
4.5. Expenses	GET, PUT payments for the user.	/api/expenses/{userID}/{paymentID}
4.6. Incomes	GET, PUT incomes for the user.	/api/incomes/{userId}/

		<i>{incomeID}</i>
4.7. Groups	GET, PUT, DELETE groups	<i>/api/groups/{groupID}</i>
4.8. Kickstarters	GET, PUT, DELETE kickstarters information.	<i>/api/kickstarter/{kickstarterID}</i>
4.9. Goals	GET, PUT, DELETE goals	<i>/api/groups/goals</i>
4.10. Payment	GET, PUT payments	<i>/api/groups/goals/{paymentID}</i>
4.11. Kickstarter payment	GET, PUT kickstarter payment	<i>/api/kickstarter/{kickstarterID}/{paymentID}</i>