

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2017

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

Project name	Online Income/Expenses Management System
--------------	--

1. Short project description (Business needs and system features)
<p>The Internet age changed the way we earn and spend money. The financial management can be the difference between poor and very successful people. The Online Income/Expenses Management System provides its users with the ability to keep track on their account balance. In addition to that it allows users to create a saving plan in order for them to save even more money. The system will be developed as a <i>Single Page Application (SPA)</i> using React.js as front-end, and Node.js + express as backend technologies.</p> <p>The main user roles (actors in UML) are:</p> <ul style="list-style-type: none"> • <i>Registrated User</i> – can add earnings and expenses and see reports about his account balance. Can also create saving plans • <i>Premium Registrated User</i> – has all of the benefits of the normal users, but in addition to that, they can administrate users

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. Register	<i>Anonymous User</i> 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 <i>Registered User</i> role.	<i>All users</i>
2.2. Log In	All of the users can log in.	<i>All users</i>
2.3. Change user data	Every user can change his personal information	<i>All users</i>
2.4. Add income and expenses	Every user can add income and expenses	<i>All users</i>
2.5. Add periodical transactions	Every user can add periodical transactions	<i>All users</i>
2.6. Edit/Delete income and expenses	Every user can edit and delete income and expenses	<i>All users</i>
2.7. Account history	Every user can see the history of his spendings/earnings	<i>All users</i>
2.8. Create/Edit/Delete savings plans	The user can create his own savings plan	<i>All users</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
3.1. Dashboard	Presents the reports about all the account information	/

3.2. Add expenses/earnings	Presents the interface to add expenses/earning	<i>/transactions</i>
3.3. Savings plan	Presents the plan's detail	<i>/plans</i>
3.4. User Data	Presents ability to view and edit personal <i>User Data</i> , as well as deregister from the system	<i>/personal</i>

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
4.1. User	GET, POST, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> .	<i>/api/users</i>
4.2. Transactions	GET users and POST Transaction(Id is auto-filled by the system and modified entity is returned as result from POST request).	<i>/api/transactions</i>
4.3. Authentication	POST method for login authentication	<i>/api/auth</i>
4.4. Plan	GET, PUT, DELETE <i>Savings Plan</i> (according to <i>User's Role</i> and identity)) and POST new <i>Plans</i> (Id is auto-filled by the system and modified entity is returned as result from POST request).	<i>/api/planDetail</i>