

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

Project author			
№	First name, last name	E-mail	Face-to-face/ online
1	Bilyana Gospodinova	gospodinova_beki@mail.bg	face-to-face
2	Daniel Shushkov	shushkovdaniel@gmail.com	face-to-face

Project name	Pass FMI
--------------	----------

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

As regular students we all need books to increase our knowledge. The *Pass FMI* system provides the possibility students to offer their old books (to exchange or sell them) to other students that might need them. In addition, it allows users to register in order to offer a book and administrators to approve the offering and manage all users and offerings. The system will be developed as *Single Page Application (SPA)* using **React.js** as front-end and **Node.js + Express.js** as backend technologies.

The main user roles (actors in UML) are:

- *Anonymous User* - can view the offerings and search them by some criteria.
- *Student* (extends Anonymous User) - can create an offering and can edit his own offerings so far.
- *Administrator* (extends Student) - can manage users; accepts or rejects the newly created offerings by the students; can create, edit, delete an existing offering.

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1 Browse information and see items	The <i>User</i> can browse the information views (Home, Products, About) in Pass FMI SPA, and can search by subject and categories.	All Users
2.2 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>User</i> role.	<i>Anonymous User</i> , <i>Administrator</i>

	<i>Administrator</i> can register new users by entering <i>User Data</i> and choosing a Role (<i>Student</i> or <i>Administrator</i>).	
2.3 Change User Data	<p><i>User</i> can edit his own personal details - first name and last name.</p> <p><i>Administrator</i> can view and edit the <i>User Data</i> of all <i>Students</i> and assign them <i>Roles</i>: <i>Student</i> or <i>Administrator</i>.</p>	<i>Student</i> , <i>Administrator</i>
2.4 Manage Users	<i>Administrator</i> can modify a user role - increase or decrease it (remove <i>student</i>).	<i>Administrator</i>
2.5 Add/Edit Items	<i>Administrator</i> can create, edit and delete existing items.	<i>Student</i> , <i>Administrator</i>
2.6 Filter Items	<i>Anonymous user</i> , <i>Student</i> and <i>Administrator</i> browse available items and can sort and filter them using different items metadata fields.	<i>Anonymous user</i> , <i>Student</i> , <i>Administrator</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
3.1 Home	Presents introductory information for the purpose of the system. Offers an ability to register.	/
3.2 Items	Presents all of the items that are currently for sell or exchange by the students. Offers the ability to create, update or delete (CRUD) and browse the items according to user's role as defined by UCs.	/items
3.3 User data	Presents the ability to view and edit personal data, as well as deregister.	/personal
3.4 Users	Presents the ability to manage (CRUD) users and their user data (available only for Administrators as described in the UCs).	/users
3.5 About	Presents information about 'Pass FMI' project and its 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>FMIRMS</i> and modified	/api/users

	entity is returned as result from POST request). Available only for <i>Administrators</i> .	
4.2 User	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> .	<i>/api/users/{userId}</i>
4.3 Items	GET items (according to <i>User's Role</i> and identity) and POST new <i>Item</i> (Id is auto-filled by <i>Pass Fmi system</i> and modified entity is returned as result from POST request).	<i>/api/items</i>
4.4 Item	GET, PUT, DELETE <i>Item Data</i> (including <i>Capacity and etc.</i>) for <i>Item</i> with specified <i>itemId</i> .	<i>/api/items/{itemId}</i>
4.5 Departments	GET departments and POST new <i>Departments</i> for <i>Item</i> (Id is auto-filled by <i>Pass FMI</i> and modified entity is returned as result from POST request).	<i>/api/items/departments</i>
4.6 Department	GET, PUT, DELETE department for <i>Item</i> with specified <i>departmentId</i> .	<i>/api/items/departments/{departmentId}</i>
4.7 Comments	GET comments and POST new <i>Comments</i> for <i>Item</i> (Id is auto-filled by <i>Pass FMI</i> and modified entity is returned as result from POST request).	<i>/api/items/{itemId}/comments</i>
4.8 Comment	GET, PUT, DELETE Comment for specified <i>Item</i> with specified <i>commentId</i> .	<i>/api/items/{itemId}/comments/{commentId}</i>
4.9 Subjects	GET subjects (according to <i>User's Role</i> and identity) and POST new <i>subject</i> (Id is auto-filled by <i>Pass Fmi system</i> and modified entity is returned as result from POST request).	<i>/api/subjects</i>
4.10 Subject	GET, PUT, DELETE subject <i>Data</i> (including <i>Department and etc.</i>) for <i>Subject</i> with specified <i>SubjectId</i> .	<i>/api/subjects/{subjectId}</i>