Assignment 1	Project Summary
Course	**Web Application & Service Development with Spring Framework v5 - 2018**

Project author

№	First name, last name	E-mail	Face-to-face/ online
1	Simeon Angelov	simeon.angelov9@gmail.com	face-to-face
2	Boris Arsov	kuce.maca@gmail.com	face-to-face

Project name	CV Builder

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

Since people need a CV for almost every new job they apply for, such a generator/builder can be quite useful. CV World will be an online CV and cover letter generator.It will feature guide lines and articles too, that will help the user in the process. The system will be developed using *Spring 5 Application Development Framework*. It will implement web-based front-end client using *Thymeleaf/JSP* templates and *jQuery* or *React.js/Vue.js/Angular* JavaScript/TypeScript libraries for asynchronous page data updates. Each page will have a distinct URL, and the routing between pages will be done server side using *SpringMVC/ WebFlux*. The backend will be implemented as a *REST/JSON API* using JSON data serialization. The main user roles (actors in UML) are:

• *Guest* – can read articles and build a CV

• *User* - same as guest, but data can be stored on server so he can build his CV easier next time

• *Administrator* – can manage (create, edit user data and delete) all *Users*

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1. Build CV	*Users/Guests/Administrators* can build a CV	*Guest/User/Administrator*
2.2. Register	*User:* Can register in order to manage his data easier (it will be remembered). Valid email and names will be required *Guest:* Can use the functionality of the site, but data will not be stored ot server (maybe) *Administrator:* Only the ones who manage the site	*Guest*
2.3. Change User Data	*Administrator and User* can view and edit only their own *User Data*	*User/ Administrator*
2.4. Manage Users	*Administrator* can choose a *User* to manage, and can manage the chosen User - edit or delete.	*Administrator*
2.5. Login	*Administrator* or User who already have an account can log in	*Administrator/User*
2.6. Build cover letter	*Users/Guests/Administrators* can generate a Cover letter	*Guest/User/Administrator*
2.7. Save CV/Cover letter	*Users and Administrators only* can save created CVs for later in their profile	*User/Administrator*

2.8. Read articles	*Users/Guests/Administrators can read the articles on the site*	*Guest/User/Administrator*

3. Main Views		
View name	**Brief Descriptions**	**URI**
3.1. Home	Presents an overview of the site, with navigation in header and footer and link to some of the articles	*/*
3.2. CV	Presents the functionality for generating a CV	*/builder*
3.3. Cover letter	Presents the functionality for generating a cover letter	*/cover-letter*
3.4. Tips	Presents *articles related to the topic*	*/tips*
3.5. User Registration	Presents a view allowing guests to register	*/register*
3.6. Login	Presents a view allowing the users to login.	*/login*
3.7. Profile	Presents ability to view and edit personal *User Data*	*/personal*
3.8. Users	Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs).	*/users*
3.9. About	Presents information about the site.	*/about*

.

4. API Resources		
View name	**Brief Descriptions**	**URI**
4.1. Register	GET *User Data* for all users, and POST new *User Data* (Id is auto-filled t). Available only for *Administrators*.	*/api/users*
4.2. User	GET, PUT, DELETE *User Data* for *User* with specified *userId*. Available only for *Administrators*.	*/api/users/{userId}*
4.3. Login	POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests.	*/api/login*
4.4. Logout	POST a logout request for ending the active session with *the site,* and invalidating the issued *Security Token*.	*/api/logout*
4.5. CV	GET *Cv for a user*, and POST new *CV, DELETE CV*	*/api/{userId}/cv/{cvId}*
4.6. Cvs	Get all cv's for a user	*/api/{userId}/cv*
4.6. Cover letter	GET, PUT, DELETE *Student Group* (including assigned students) for *Student Group* with specified *groupId*.	*/api/{userId}/coverLetter/{coverletterId}*
4.7. Tips	GET articles	*/api/tips*
4.8. Tip	GET, PUT, DELETE, POST *article*	*/api/tips/{tipsId}*

.