

Assignment 1	Project Summary
Course	Web Application & Service Development with Spring Framework v5 - 2019

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

Project name	Gift Me
--------------	---------

1. Short project description (Business needs and system features)
<p>The Internet has a huge impact on people to acquire goods and services in ever-increasing amounts. But it can also be used for challenging consumerism and encouraging reusing inventory and stocks.</p> <p>The Gift Me system provides the ability for people to gift to others their unused stocks and items as well as choose and receive gifts from others.</p> <p>Gift Me also allows users to register, create friendships (one to one friendships and group friendships) and post items which they want to give away, and administrators to manage users and their content.</p> <p>The system will be developed using Spring 5 Application Development Framework. It will implement web-based front-end client using Thymeleaf templates and jQuery JavaScript libraries for asynchronous page data updates.</p> <p>Each page will have a distinct URL, and the routing between pages will be done server side using SpringMVC. The backend will be implemented as a REST/JSON API using JSON data serialization.</p>

There will be also a real-time event streaming from the server to the web client using **WebFlux** and **Server Sent Events (SSE)/WebSocket** in order to allow the users to get notifications for friendship/group invitations or for someone being interested in a gift they have posted, in real time. The main user roles (actors in UML) are:

- *Anonymous User* – can only view the Home Page of the application (which reveal its intent) and login/register.
- *User* (extends *Registered User*) –. can add friends, create groups and manage the members, post gift items and choose who to take it, click “Want it” on their friends gifts. Also *User* can search for people, groups or items by keyword..
- *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as *Groups* and *Items* .

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. See information	The <i>User</i> can see the information views(Gifts Feed, My Active Gifts, Given Gifts History, Received Gifts History, User Data, My friends, My groups).	<i>Registered User, Administrator</i>
2.2. Register	<i>Anonymous User</i> can register in the system by providing a valid e-mail address and phone number, first and last name, and choosing password. <i>Administrator</i> can register new by entering <i>User Data</i> .	<i>Anonymous User, Administrator</i>
2.3. Change User Data	<i>Registered User</i> can view and edit her personal <i>User Data</i> .	<i>Registered User, Administrator</i>

	<i>Administrator can view and edit User Data of all Users and assign delete their posts or remove them from groups, delete profiles.</i>	
2.4. Manage Users	<p><i>Administrator can browse and filter users based on different criteria: first and last name, email, Role.</i></p> <p><i>Administrator can choose a User to manage, and can manage the chosen User - edit (using Change User Data UC) or delete.</i></p> <p><i>Administrator can create a new user using Register UC.</i></p>	<i>Administrator</i>
2.5. Create and Manage Groups	<p><i>User can create a group and accept/reject group's membership request(only if he/she is the creator).</i></p> <p><i>Administrator can browse and filter Groups based on different criteria: name of Group, creation date, etc.</i></p> <p><i>Administrator or User can create a new Group.</i></p> <p><i>Only Administrator and User, who is group owner, can accept group membership requests and remove members from the group.</i></p> <p><i>Administrator can delete a group if it is necessary.</i></p>	<i>Administrator, User</i>
2.6. Search information	<i>The User/Administrator can search for users/groups/items by providing keywords and selecting what he/she is searching for- users, groups or items(gifts) and see the results on Search Result page.</i>	<i>User, Administrator</i>

	<p>Items which are found are only the visible one for the user.</p> <p>E.g. items posted from user's friends or in some of the groups of which the user is member.</p>	
2.7. Manage friends by sending friendship requests or accept/reject received requests	<p>The <i>User</i> can send friendship request to friends found via Search(Using Search information UC).</p> <p>The <i>User</i> can accept/reject friendship requests sent to him/her, which can be seen on <i>Friendships View</i>.</p> <p><i>Administrator</i> can manage the friendships of all users.</p>	<i>User, Administrator</i>
2.8. Manage oneself group membership	<p><i>User</i> can choose to accept or reject group membership requests, which can be seen on <i>Groups View</i>.</p> <p>Any <i>User</i> who is group member can suggest adding new members.</p> <p><i>User</i> can leave the group at any time.</p>	<i>User</i>
2.9. Post gifts and manage them	<p>The <i>User</i> can post gift items on the news feed or in a group, which they participate in.</p> <p>If the gift item is posted on the Gifts Feed it can be seen only from the friends of the user (it will appear on their Gifts Feed).</p> <p>When a gift item is posted in a group page it can be seen only</p>	<i>User, Administrator</i>

	<p>from the other group members.</p> <p><i>Administrator</i> can remove gift items, which are not legit or make an item inactive.</p>	
2.10. Attempt to take a gift	<i>User</i> can attempt to take a gift item by clicking 'Want it' on it.	<i>User</i>
2.11. Accept gift receiver	<i>User</i> can see in real-time who wants to take particular gift choose one receiver by clicking 'Choose' on his/her name in	<i>User</i>

3. Main Views (Frontend)		
View name	Brief Descriptions	URI
3.1. Home	Presents the introductory information for the purpose of the system as well as detailed instructions on how to start using it. Prominently offers ability to register.	/
3.2. Gifts Feed	Presents currently non-taken (active) gifts posted by user's friends. Offers abilities to browse, choose and attempt to take by clicking 'Want it' button.	/gifts-feed
3.3. My Active Gifts	<p>Presents to the user his/her active gifts as well as the list of gifts' "wanters" and the exact time when any of them has clicked "Want it".</p> <p>Only the user can see the list of people who wants his gift. Friends of the user can only see the active gifts without seeing who wants them.</p>	/active-gifts

	The owner of the gifts can click 'Choose' on anyone who wishes the gift. This will make the gift inactive and will send notification to all the "wanters" of the gift which reveals if they are accepted. Other users can only click 'Want it'.	
3.4. Given Gifts History	Presents <i>Test Results</i> available according to <i>User's Role</i> and identity as defined by <i>UCs</i> .	<i>/test-results</i>
3.5. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register in <i>OKTS</i> , as well as to enroll to a <i>Student Group</i> by presenting a <i>Token Key</i> for that group.	<i>/register</i>
3.6. Login	Presents a view allowing the users to login.	<i>/login</i>
3.7. User Data	Provides ability to view and edit personal <i>User Data</i> , enroll to more <i>Student Groups</i> by presenting a <i>Token Key</i> for particular <i>Student Group</i> , as well as deregister from <i>OKTS</i> .	<i>/personal</i>
3.8. Student Groups	Presents ability to browse, filter and manage <i>Student Groups</i> (for <i>Administrators</i> and <i>Instructors</i>), and assign Students to them (for <i>Administrators</i> only), as described in <i>UCs</i> .	<i>/groups</i>
3.9. Dashboard	Presents in real time the Students' progress on active Tests, subject to Role restrictions described in <i>UCs</i> .	<i>/dashboard</i>
3.10. Users	Presents ability to manage (CRUD) Users and their User Data (available for <i>Administrators</i> only, as described in <i>UCs</i>).	<i>/users</i>
3.11. About	Presents information about the <i>OKTS</i> project and his 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>OKTS</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>OKTS</i> , and invalidating the issued <i>Security Token</i> .	<i>/api/logout</i>
4.5. Student Groups	GET <i>Student Groups</i> , and POST new <i>Student Group</i> (Id is auto-filled by <i>OKTS</i> and modified entity is returned as result from POST request), according to <i>User's Role</i> and identity security restrictions.	<i>/api/groups</i>
4.6. Student Group	GET, PUT, DELETE <i>Student Group</i> (including assigned students) for <i>Student Group</i> with specified <i>groupId</i> .	<i>/api/groups/{groupId}</i>
4.7. Tests	GET users (according to <i>User's Role</i> and identity) and POST new <i>Test</i> (Id is auto-filled by <i>OKTS</i> and modified entity is returned as result from POST request).	<i>/api/tests</i>
4.8. Test	GET, PUT, DELETE <i>Test Data</i> (including <i>Questions</i> and <i>Answers</i>) for <i>Test</i> with specified <i>testId</i> .	<i>/api/tests/{testId}</i>
4.9. Test Results	GET <i>Test Results</i> (according to <i>User's Role</i> and identity)) for <i>Test</i> with specified <i>testId</i> , and POST new <i>Test Result</i> (Id is auto-filled by <i>OKTS</i> and modified entity is returned as result from POST request).	<i>/api/tests/{testId}/results</i>
4.10. Test Result	GET, PUT, DELETE <i>Test Result</i> (according to <i>User's Role</i> and identity) for <i>Test</i> with specified <i>testId</i> and <i>Test Result</i> with specified <i>testResultId</i> .	<i>/api/tests/{testId}/results/{testResultId}</i> /
4.11. Active Tests	SSE event streaming of Students' progress on currently active <i>Tests</i> (separate event pushed for each <i>Question Answer</i>), according to <i>User's Role</i> and identity security restrictions.	<i>/api/active-tests</i>