

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

Project author			
No	First name, last name	E-mail	Face-to-face/ online
1	Georgi Bojinov	georgi.bojinov@hotmail.com	face-to-face
2	Dragomir Proychev	dragproychev@gmail.com	face-to-face

Project name	Jitter – next generation social network
--------------	---

1. Short project description (Business needs and system features)
<p>Social media has become an important factor in people's lives, reaching the point where they can't live without it. Everything from communication to sharing content, keeping up with all kinds of people and even job hunting, social media can do. As a result, their complexity has risen exponentially, and the number of features can be overwhelming for users. The need for more intuitive and straightforward social networks has arisen and as such, Jitter aims to focus on just that – a simple, intuitive UI, having the essential features but building them in a simple, easy to understand way – a chat system that is fast and efficient and doesn't use external applications, a like and comment system that makes sure you want to like what you're liking, a post system that allows for full editing and deletion of content, and a simple user system without multiple role splitting. Jitter is simple, straightforward and intuitive – just what a social network has to be.</p>

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. News feed	The <i>User</i> can browse his news feed and see all recent posts by friends and his/her own posts.	All users

<b>2.2. Register</b>	<i>Anonymous User</i> can register in the social network by providing a valid e-mail address, first and last name, choosing a password and providing profile data – location, education, interests.	<i>Anonymous User</i>
<b>2.3. Change Profile</b>	<i>User</i> can view and edit his/her personal profile and choose what is private, can only be seen by friends or is public.	<i>User</i>
<b>2.4. Add/Edit Post</b>	<i>User</i> can make a post on his/her personal page, update, delete, and view it and select whether it is private or not.	<i>User</i>
<b>2.5. Add/Edit Comment</b>	<i>User</i> can add a comment to a post if commenting is allowed and if the user can view the post. The user can also edit and/or delete the comment.	<i>User</i>
<b>2.6. Add/Remove Like</b>	<i>User</i> can like a comment or a post and remove his/her own likes on posts and comments.	<i>User</i>
<b>2.7. Add/Remove Friend</b>	<i>User</i> can add or remove other users from his/her friends list. He/she can also accept or reject friend requests from other users.	<i>User</i>
<b>2.8. Send chat message</b>	<i>User</i> can send a message to their friends or to another user if he/she has permitted that.	<i>User</i>
<b>2.9. Permit chat message</b>	<i>User</i> can specify who can send him/her chat messages – only friends or everybody.	<i>User</i>
<b>2.10. View/Archive messages</b>	<i>User</i> can view all his/her messages and message history with anybody. He/she can also choose to archive any messages that are obsolete.	<i>User</i>
<b>2.11. Manage Friends list</b>	<i>User</i> can view, add or delete friends from his/her friends list.	<i>User</i>
<b>2.12. Manage activity log</b>	<i>User</i> can view, modify or delete activity logs, make them private,	<i>User</i>

	hidden, etc.	
--	--------------	--

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
<b>3.1. Home</b>	Displays the news feed of the currently logged in user, including the posts and recent activity of the user's friends.	/
<b>3.2. Login</b>	Allows an anonymous user to log into the system	/user/login
<b>3.3. Register</b>	Allows an anonymous user to create a new account	/user/register
<b>3.4. Profile</b>	Displays the profile and the activity of the user with the given id	/profile/{id}
<b>3.5. Messages</b>	Displays a list of all conversations for the currently logged in user	/messages
<b>3.6. About</b>	Displays additional information about the user	/profile/{id}/about
<b>3.7. Conversation</b>	Displays conversation history for the user with the given id	/messages/{id}
<b>3.8. Activity log</b>	Displays a log of the user's recent actions and activities	/activity/{id}

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
<b>4.1. Users</b>	POST Create a new user with the given credentials	/api/users
<b>4.2. User</b>	GET, PUT, DELETE Information about user. Available only for the current user.	/api/users/{id}
<b>4.3. Profile</b>	GET Information about a user's profile. PUT Update your own profile by modifying the existing information.	/api/profile/{id}

<b>4.4. Posts</b>	POST Create a new post	<i>/api/posts</i>
<b>4.5. Post</b>	GET A post with the given id. PUT, DELETE A post the logged in user has created.	<i>/api/posts/{id}</i>
<b>4.6. Comments</b>	POST Create a new comment for a post with the given id. GET All comments for the corresponding post	<i>/api/comments/{postId}</i>
<b>4.7. Comment</b>	PUT, DELETE A comment the logged in user has posted.	<i>/api/comments/{commentId}</i>
<b>4.8. Likes</b>	POST Give a like to a post with postId. GET All likes for an existing post. DELETE The user's like of a post if the logged in user has already liked it	<i>/api/likes/{postId}</i>
<b>4.9. Messages</b>	GET All messages for the conversation between the logged in user and user with userId. POST Send a new message for this conversation.	<i>/api/messages/{userId}</i>
<b>4.10. Message</b>	PUT Modify an existing message. DELETE Delete a message	<i>/api/messages/{messageId}</i>
<b>4.11. Friends</b>	GET Returns all friends of the user with given id  POST Responds to a friend request or sends a friend request to user  DELETE removes a user from the logged user's friends list	<i>/api/friends/{userId}</i>