|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Multimedia Technologies with Angular and TypeScript - 2020 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/online |
| 1 | AcidRogue | |  |

|  |  |
| --- | --- |
| Project name | WebChatRoom |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Online ChatRoom is a web chatroom where users can interact with each other using messages, images and emojis. You simply subscribe to a chatroom from a list with different chatrooms, and you can start talking to other people! You can send messages that other people can see, you can also see other people’s profiles. You can switch between your subscribed servers to see different messages and people. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Register** | The User can register, providing a username and password | Anonymous users |
| * 1. **Login** | The User can login using their username and password | All users |
| * 1. **Change User Data** | The User can view and edit their personal data. |  |
| * 1. **Browse Servers** | The User can browse servers from a menu of different servers | All users |
| * 1. **Subscribe to Servers** | The User can subscribe to servers to see their messages and other people that are subscribed to it as well | All users |
| * 1. **Unsubscribe from a server** | The User can unsubscribe from a subscribed server | All users |
| * 1. **Send and Read Messages** | The User can send and read messages in a subscribed server | All users |
| * 1. **Edit Message and Delete Message** | The User that sent a message, can decide to edit it after sending it | All users |
| * 1. **Search messages in a Server** | The User can search for messages in the server | All users |
| * 1. **View Other User’s Profile** | The User can see other users’ profiles with information, such as username, subscribed servers, etc. | All users |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Login** | Login page | /login |
| * 1. **Register** | Register page | /register |
| * 1. **Home** | Presents your chat dashboard with all of your subscribed servers on the left By default, the first server from your subscribed list is selected | /dashboard |
| * 1. **User** | Presents your website data and allows you to edit | /user/:uid |
| * 1. **See User** | Allows you to see information about a user that has a mutually subscribed server | /dashboard |
| * 1. **Create a new server** | Allows the user to create a new server | /new-server |
| * 1. **Servers** | Presents a list with all the servers from which you can subscribe to. | /servers |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET and POST User Data. Accessible only for Administrators. | */api/users* |
| * 1. **User** | GET, PUT and DELETE a User. | */api/user/{userid}* |
| * 1. **Messages** | GET and POST Messages. All users can send messages. The messages have a sender id and server id. | *api/messages* |
| * 1. **Search Messages** | GET Messages from a user. | *api/messages/search?user={userid}* |
| * 1. **Search Messages** | GET Messages from a search term. | *api/messages/search?term={searchTerm}* |
| * 1. **Search Messages** | GET Messages from a server. | *api/messages/search?server={serverid}* |
| * 1. **Search Messages** | GET Messages from all search options. | *api/messages/search?server={serverid}&user={userid}&search={searchTerm}* |
| * 1. **Message** | GET and DELETE a message. | *api/message/{messageid}* |
| * 1. **Servers** | GET and POST Servers. Different servers have different messages and subscribed users. Only Admins can create new Servers. | *api/servers* |
| * 1. **Server** | GET and DELETE a Server. If DELETE is used, all messages connected to the server will be deleted as well. | *api/server/{serverid}* |