|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Angular frontend and Node.js backend | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | KM | | face-to-face |

|  |  |
| --- | --- |
| Project name | Virtual Gift |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Everyone wants to make the perfect gift for a loved one, friend or a co-worker. The Virtual Gift provides you the ability to create a media content for someone (videos or photos).  A person (the event creator) will be able to create an event, choosing from several different events. He will be able to invite other contributors to the event. Everyone will be able to upload a short video or a photo/s to the event. After everyone invited commits their content, the event will generate (by a template based on the event) a content that then can be sent to the person who you will want to surprise. The system will be developed as a *Single Page Application (SPA)* using ***Angular*** as front-end, and ***Node.js + Express*** as backend technologies.  The main user roles (actors in UML) are:  • *View only User* – can only view the final content.  • *User* –can create an Event, can contribute with content, can invite other users to contribute.  • *Administrator* – can manage (create, edit user data and delete) all *Registered Users.* |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Create Event** | A user can create and manage an event. Set end date for the event, invite other contributors. | All users |
| * 1. **Register** | *User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. | *User, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, email.  *Administrator* can choose a *User* to manage, and can manage the chosen User - edit (using Change User Data UC) or delete.  Administrator can add/delete a user to event, can create, edit and delete an event.  *Administrator* can create a new user using *Register UC*. | *Administrator* |
| * 1. **View Content** | Viewer can only view the end content | *User* |
| * 1. **Add multimedia to content (Photo, Video)** | If you are user subscribed to an event, you can contribute to the multimedia for the event. The user can delete only content that he submitted for a current event. | *User* |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Login and Register Page** | Login Page in which you will provide valid username and password . Register Page to register a user | <host:port>/login  <host:port>/register |
| **Events view** | List of all events that you participate to and information for them | <host:port>/events |
| * 1. **Event Content view** | The current content of a particular event. There you will be able to add more content and invite people, or remove a user (if you are the creator of the event) | <host:port>/eventContent |
| * 1. **About View** | Information about the current user and avatar | <host:port>/about |
| * 1. **Create new Event View** | View for creating a new event. You will be shown the minimum fields needed to start an event | <host:port>/createEvent |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| **Authenticate** | POST request for aquiring auth token. | <host:port>/users/authenticate |
| **Register** | POST request for registering a user. | <host:port>/users/register |
| **Event invite** | POST request for inviting existing user to contribute to event. | <host:port>/users/invite |
| **Current user** | GET request to get information for current user | <host:port>/users/current |
| **Create event** | POST request for creating an event | <host:port>/users/:userId/events |
| **Delete event** | DELETE request for deleting an event | <host:port>/users/:userid/delete/:eventId |

|  |  |  |
| --- | --- | --- |
| **Update Event** | Update the current event (Add more user to contribute) | <host:port>/updateEvent/:eventId |
| **Upload multimedia** | POST request uploading content (photos, videos) for particular event. Need to provide eventId | <host:port>/upload/:eventId |
| **Get Multimedia** | GET request for accessing the multimedia for a current user and paricular event | <host:port>/users/:userid/content/:eventId |
| **Delete Multimedia** | DELETE request for deleting multimedia for a particular user and particular event. You can specify a name of a particular resource that you want to delete as an optional query param in the url | <host:port>/users/:userid/deleteContent/:eventId?name={name} |