|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignment 1 | | Project Summary | | |
| Course | | Full-stack Development with Node.js and React.js - 2016 | | |
|  | | | | |
| Project author | | | | |
| № | First name, last name | | E-mail | Face-to-face/ online |
| 1 | Anatoly Pachev | | anatoly.pachev@gmail.com | Online |

|  |  |
| --- | --- |
| Project name | A Simple Customer Management System |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| A simple Customer Management System (CRM) catering primarily to the needs of small organizations and business individuals.  The main system functionality revolves around the aspects of storing, organizing and looking up the contact details of companies and individuals that are somehow related to or are of interest to the administrators of the CRM.  The backend of the CRM will be built on top of *node.js* and *hapi.js* with data management done in *MongoDB*; the frontend will be executed primarily in *React* and *jQuery*, leveraging the frameworks’ abilities to provide a full and seamless SPA experience to the end user.  In order to make use of the application’s functionality, one has to be a registered user. Anonymous user access will be limited to the CRM’s main/ login view and the About view. Registered users will come in either one of two categories/ roles:   * **CRM Manager** – a user whose privileges allow him or her to create new points of contact and supply or update data for such contacts; Managers are responsible for managing the contacts that they have themselves created or that have been assigned to them through an Administrator-class user. * **CRM Administrator** – a user-class that has unlimited access and rights towards managing CRM Manager accounts within the system as well as accessing/ creating/ updating/ deleting contact information. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Access the Home/ Login view** | All users can access the application’s entry view where they can attempt to log into the application. | *All users* |
| * 1. **Add a new user** | *CRM Administrators* are the only users who can create new user accounts (CRM Admins and CRM Managers) for accessing the database’s resources. | *CRM Administrator* |
| * 1. **Change User Data** | *CRM Managers* can view/ update their personal user details and account data with the exception of their usernames and system privileges.  *CRM Administrators* can view/ update their own user data as well as that of other registered users. | *CRM Manager/ CRM Administrator* |
| * 1. **Manage Users** | *CRM Administrators* can browse and filter users based on different criteria: first and/ or last name, email, etc.  *CRM Administrators* can edit a user’s account/ personal data (including upgrading or downgrading the user’s role, but cannot edit the username of an existing user), can park a user’s account (thus making it unusable until reactivated) or mark an account for deletion (permanent deletion of user account is not allowed on the DB-level). | *CRM Administrator* |
| * 1. **Create a New Contact** | Both *CRM Managers* and *Administrators* cancreate new contact points within the system and supply the necessary contact details. | *CRM Manager/ CRM Administrator* |
| * 1. **View/ Update Existing Contact Information** | The active management of existing contact data is generally the responsibility of the CRM Manager that created the contact and who is, thus, deemed the contact owner responsible for subsequent updates to that data.  *CRM Managers* can update the details of those contacts that they have initially registered in the system and those that have been assigned to them by a *CRM Administrator*.  *CRM Administrators* can update the data belonging to any existing contact, including the CRM Manager’s account to which a particular contact is assigned. | *CRM Manager/ CRM Administrator* |
| * 1. **Search for a Contact or Browse All Contacts** | Registered users have access to all contact details stored in the database. Available contacts can be searched for using filters on contact details such as names, telephones, emails, etc. | *CRM Manager/ CRM Administrator* |
| * 1. **Mark a Contact for Deletion** | Contact records can be marked for deletion by both CRM Managers and Administrators. Administrators can mark any existing contact for removal from the active contact base while Managers can only invalidate contacts that are assigned to them for managing.  Contacts that have been marked for deletion are only marked as invalid while their data remain intact in the database. Such contact records are filtered out of CRM Managers’ contact search results while still available to Administrators. | *CRM Manager/ CRM Administrator* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home/ Login** | The entry point for starting using the system. The Home screen signals the users that the application is online and ready and points them in the direction of a simple login form. | */* |
| * 1. **Contact Creation form** | Allows CRM Managers to add new contacts and contact data to the organizational database. | */new-contact* |
| * 1. **Contact Search form** | Allows registered users to search the database for an existing contact point. | */search* |
| * 1. **Contact Details Visualization and Update form** | Allows CRM Managers to access a contact point’s available data and update this data as the circumstances require. Also available to CRM Administrators. | */contact* |
| * 1. **User Management view** | Allows CRM Administrators to create new users as well as update user details, park user accounts or mark user accounts for deletion. | */users* |
| * 1. **About** | Displays information about the project and its owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users and POST new *User Data* (Id is auto-filled by *the CRM*, and a modified entity is returned as a result from the POST request). Available only to *CRM* *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*. | */api/users/{userId}* |
| * 1. **Contacts** | GET *Contacts Data* for all existing contacts and POST new *Contact Data* (Id will be supplied by the app). Available to all registered users. | */api/contacts/* |
| * 1. **Contact** | GET, PUT, DELETE a single *Contact Record* as per the supplied *contactId*. | */api/contacts/{contactId}* |