|  |  |  |
| --- | --- | --- |
| **Section** | **Function/ requirement** | **Description** |
| 1. API | Create user | Post user details to the data base containing username, password, user type, name, surname, email and, contact number |
| Delete user | Deletes the current user logged in. |
| Update user | 2 functions: get user details from DB and post back user details and overwrite current details with new. |
| Create new post | Creates a new post in the relevant section and updates the user id with the current user logged in. only users with user types 1 and 2 will be allowed to create posts. (user types 1 and 2 are admin and medical professional) |
| Update user post | Updates current post in view by posing back to db with post id. Over writes old user id and date |
| Delete user post | Removes current post from db using post id. No log is kept of deletions |
| Retrieve posts | Pulls list of all posts from all sections, detail and id’s from all 4 sections. Each section to be pulled separately. |
| Log in | Takes user provided password and username and checks in database if the user exists and if the password is correct (encrypted) and provides an authentication token to be used for the session. |
| Authentication token | Random string that allows access the db and api functions from either the backend or mobile application. This is generated by the api and stored in the database alongside the user’s details. |
| **Section** | **Function/ requirement** | **Description** |
| 1. Admin Back end | Create user | Receive details of user as per API requirements and send them through to the API. The inputs are first checked for validity. Only unique usernames are allowed. |
| Delete user | Using the currently logged in user, send a request to the API to delete the current user. |
| Update user | Using the currently logged in user, retrieve all the details of the current user and display. After updates, capture all details from the form and send back into the API to be updated using the current users ID. |
| Log in | On landing page send user name and password into API to receive an authentication token and user ID. |
| Create post | Sends heading and post details to the API along with the user id and current date. |
| Delete post | Sends a request to the api to delete a current post. This is done by clicking a delete button. |
| Update post | Having a review page with all the post details and header. All the fields can be edited. Once edited it should send back to the API along with the current users ID and current date |
| Select over all post view | This is a display of all the posts that can be viewed. Upon clicking on one of the posts it would open up and show all the detail of the post. This is subdivided by the section that the post falls into. |
| View post time line | This is a simple report that shows all the posts and when they were created/updated |
| **Section** | **Function/ requirement** | **Description** |
| 1. Mobile app\*\* | Create user | Receive details of user as per API requirements and send them through to the API. The inputs are first checked for validity. Only unique usernames are allowed. |
| Delete user | Using the currently logged in user, send a request to the API to delete the current user. |
| Update user | Using the currently logged in user, retrieve all the details of the current user and display. After updates, capture all details from the form and send back into the API to be updated using the current users ID. |
| Log in | On landing page send user name and password into API to receive an authentication token and user ID. |
| Create post | Sends heading and post details to the API along with the user id and current date. |
| Delete post | Sends a request to the api to delete a current post. This is done by clicking a delete button. |
| Update post | Having a review page with all the post details and header. All the fields can be edited. Once edited it should send back to the API along with the current users ID and current date |
| Select over all post view | This is a display of all the posts that can be viewed. Upon clicking on one of the posts it would open up and show all the detail of the post. This is subdivided by the section that the post falls into. |

\*\* app will function the same as the backend simply without the report.