|  |  |
| --- | --- |
| **Feature** | Login |
| **Introduction** | All users will be able to login with their username and password, on first screen of the App. |
| **Inputs** | 1-User enters username.  2-User enters password.  3-User press submit button |
| **Processing** | on “Submit” button click in the form, authVerify() function will be called. It will check if the username and password combination match the one in the database. |
| **Outputs** | User enters main menu based on his/her permission level. |

|  |  |
| --- | --- |
| **Feature** | Register |
| **Introduction** | All users must have account in order to get accesses the app If a user doesn't have an account, they will be asked to create one. |
| **Inputs** | 1-User creates username.  2-User enters valid email address.  3-User creates password.  4-User press register button |
| **Processing** | on “Register” button click a registerUser() function will verify that all mandatory fields are filled out with the proper values, and none are left empty and confirmation email send to user. |
| **Outputs** | User can access the app and use all features based on her/his permission level. |

|  |  |
| --- | --- |
| **Feature** | Forgot Password |
| **Introduction** | If user forgets a username or password, he/she can reset it with email address that was provide during registration. |
| **Inputs** | 1-User enters email (that was provide during registration)  2-User press reset button |
| **Processing** | on “Reset” button click the forgotPassword() function will check if the provided email exists in the database, then the user will be emailed instructions on how to create a new password. |
| **Outputs** | User can access the app again and use it. |

|  |  |
| --- | --- |
| **Feature** | Deactivate Account |
| **Introduction** | If a user no longer wants to use the app, they might have their account deactivated. |
| **Inputs** | 1-User enters in text-box reason why they don’t want use app anymore.  2-User enters password.  3-User press deactivated button |
| **Processing** | on “Deactivated” button click the deactivateAccount() function will be called. It will delete account related to the user form the database. |
| **Outputs** | User account and all information related to account will be deleted. |

|  |  |
| --- | --- |
| **Feature** | Start recording |
| **Introduction** | When a user presses the button while they are walking or running, the system starts to record the way that the user is going through, time that user spend for running and other function will record. |
| **Inputs** | 1-User already in exercise screen  2-User press start button |
| **Processing** | on “Start” button click the startRecord() function will be called. It will start process of tracking user activities. |
| **Outputs** | The user will be able to see his/her tracking history. |

|  |  |
| --- | --- |
| **Feature** | Save record |
| **Introduction** | User press save button in order to save their workout progress. |
| **Inputs** | 1-User previously pressed start” Record”  2-User press save button |
| **Processing** | on “Submit” button click the saveRecord() function writes a record to the database table. The record is essentially a set of coordinates that describe the user’s route. |
| **Outputs** | User saves their progress that in future review it. |

|  |  |
| --- | --- |
| **Feature** | Delete activity |
| **Introduction** | User can delete some activities recorded that they don’t want to be associated with their account. |
| **Inputs** | 1-User press History.  2-User select record that he\she want to delete.  3-User press delete button. |
| **Processing** | on “Delete” button click the deleteActivity() method that allows users to delete unused or old activities to make some space for new ones. This method actually deletes a database record, so there’s no way to restore the information. |
| **Outputs** | User delete record that is no longer associated with his/her account. |

|  |  |
| --- | --- |
| **Feature** | Schedule Workout |
| **Introduction** | User have option to schedule a workout and add it to calendar. |
| **Inputs** | 1-User press calendar.  2-User press new event.  3-User select workout that he/she want to schedule and time and date as well.  4-User press save button. |
| **Processing** | on “Save” button click the saveSchedule() function will be called it writes a record to the database table and adds event to calendar. |
| **Outputs** | User can see all scheduled workouts in calendar and can be reminded about upcoming events. |

|  |  |
| --- | --- |
| **Feature** | Notifications |
| **Introduction** | User will be automatically reminded by the app about upcoming event. |
| **Inputs** | \*User must give permission to the app “To Send Notifications” in order to receive them.  1- User must already schedule workout.  2-User press settings.  3-User select notifications.  4-User set time frame that will remind him/her about upcoming event.  5 User press Turn-On. |
| **Processing** | on “Turn-ON” button click the OnNotific() function will be called it adds a record to the database table and adds reminder to the system. |
| **Outputs** | User will receive notifications about upcoming workout before certain period of time (based on his/her selection). |