# **1. Use Case Descriptions**

# 

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Login to User Account | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor), Firebase Authenticator API |
| Description: | Users need to login to use the app. |
| Preconditions: | 1. User has downloaded the app 2. User has registered an account successfully 3. There is stable internet communication available |
| Postconditions: | 1. App’s main menu page is loaded |
| Priority: | - High. Users would not be able to use the application if they are not logged in. |
| Frequency of Use: | - Whenever the user enters the app. |
| Flow of Events: | 1. User will enter his/her email address and password to log in. 2. User will click on the “LOGIN” button. 3. Application system will then verify the username and password with the database. 4. Upon successful login, the main menu page of the application will be shown |
| Alternative Flows: | AF-S3: If the user inputs incorrect password   1. If a user inputs the incorrect password, an error message “Username / Password entered is invalid, Please try again”. will be shown. 2. The user can input password again.   AF-S3: If the user inputs incorrect username   1. If a user inputs the incorrect username, an error message “Email / Password entered is invalid, Please try again”. will be shown. 2. The user can input username again.   AF-2: If the user clicks on the “FORGOT PASSWORD” button   1. User is led to the reset password page. |
| Exceptions: | 1. The user can choose to forget the password or register an account (if the account does not exist). |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | 1. Login is in English |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Search for location(s) of interest having specific type of exercise or healthy food | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor), Exercise/Food database |
| Description: | Users can search for a list of locations that have a specific type of exercise or healthy food by typing it in the search box. |
| Preconditions: | 1. User login successfully. |
| Postconditions: | 1. The Food & Exercise interface is loaded. 2. A list of sports venues for the searched exercise or restaurant that sells that specific healthy food will be loaded. |
| Priority: | - High. |
| Frequency of Use: | - Whenever the user tries to look for the location list of a particular type of exercise/healthy food |
| Flow of Events: | 1. User enters the name/type of the exercise or healthy food into the search box. 2. System searches for locations of interest from its database that have the type of exercise or healthy food matching the input received from the user. 3. Search returns a list of locations with the matching type of exercise or food. 4. User selects one of the locations from the result list obtained. |
| Alternative Flows: | AF-S2: If the searched exercise/food type is not found   1. No results will be shown. 2. If the search query is spelt wrongly, the user will be prompted to go to a new results page with the correctly spelt search query. 3. The user returns to step 2 if the new results page is not selected or cannot be generated |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | 1. User enters valid exercise/food type |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | Check Crowd Level at Location | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (initial actor), Thread API |
| Description: | User can check crowd level at location of interest previously found in search query |
| Preconditions: | 1. User login successfully. 2. Valid exercise/food type has been entered in the search query 3. There exists a match for location for valid exercise/food type in the database |
| Postconditions: | 1. Crowd level status is shown under the location of interest searched |
| Priority: | - High |
| Frequency of Use: | - Whenever the user wants to check the crowd level status of a particular location of interest. |
| Flow of Events: | 1. User clicks on one of the locations of interest returned by the search query 2. Crowd level status and other relevant information regarding the location of interest is displayed |
| Alternative Flows: | AF-S4 : If the crowd level status cannot be fetched by the application or does not exist in database API   1. Users will be prompted that the crowd level data is currently unable to be fetched at the current time 2. The user returns to the search results page. |
| Exceptions: | N/A |
| Includes: | Use Case 2 |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Check Weather Condition | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can check weather condition of the location |
| Preconditions: | 1. User login successfully. 2. Valid exercise/food type has been entered in the search query 3. There exists a match for location for valid exercise/food type in the database |
| Postconditions: | 1. Location’s weather condition is shown under the location of interest. |
| Priority: | - High |
| Frequency of Use: | - Whenever the user wants to check the weather condition of a particular location of interest. |
| Flow of Events: | 1. User clicks on one of the locations of interest returned by the search query 2. Weather condition and other relevant information regarding the location of interest is displayed |
| Alternative Flows: | AF-S4 : If the weather condition cannot be fetched by the application or does not exist in database API   1. Users will be prompted that the weather condition data is currently unable to be fetched at the current time 2. The user returns to the search results page. |
| Exceptions: | N/A |
| Includes: | Use case 2 |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Check people in vicinity | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can check people in the vicinity. |
| Preconditions: | 1. User login successfully. 2. Access to location is enabled. 3. Other users registered an account. 4. Other users’ access to location is enabled. |
| Postconditions: | 1. A list of profiles of other users are shown. |
| Priority: | - High |
| Frequency of Use: | - Whenever the user wants to check people in the vicinity. |
| Flow of Events: | 1. User clicks on Find a partner 2. List of profiles of other users in the vicinity are shown. |
| Alternative Flows: | AF-S2 : If there are no users in the vicinity.   1. Users will be prompted that there are currently no users in the vicinity. 2. The user will return to the main menu page. |
| Exceptions: | N/A |
| Includes: | Use case 1 |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | Chat | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can chat with other people in the vicinity. |
| Preconditions: | 1. User login successfully. 2. Access to location is enabled. 3. Other users registered an account. 4. Other users’ access to location is enabled. |
| Postconditions: | 1. Chat interface is shown. |
| Priority: | - High |
| Frequency of Use: | - Whenever the user wants to chat with people in the vicinity to work out/eat together. |
| Flow of Events: | 1. User clicks on Find a partner 2. List of profiles of other users in the vicinity are shown. 3. User clicks on the user's profile that the user wants to chat with. 4. Chat interface is shown. |
| Alternative Flows: |  |
| Exceptions: | N/A |
| Includes: | Use Case 6 |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | Access to User Location | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can allow access to location in the app. |
| Preconditions: | 1. User login successfully. |
| Postconditions: | 1. A confirmation of consent will be shown. |
| Priority: | - High |
| Frequency of Use: | - Whenever the user wants to query location or check people in the vicinity. |
| Flow of Events: | 1. If access to location has not been turned on, a pop up will be shown. 2. User can click on Disallow or Allow button. 3. A confirmation display will be shown. |
| Alternative Flows: |  |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 8 | | |
| Use Case Name: | Register Account | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor), Database API |
| Description: | Users can register an account. |
| Preconditions: | 1. User has downloaded the app. 2. User does not has registered an account with the email provided. |
| Postconditions: | 1. App login page is loaded |
| Priority: | - High. User would be unable to login without an account. |
| Frequency of Use: | - When they first use the application. |
| Flow of Events: | 1. User clicks on the “Register” button. 2. User will enter their full name, email address, date of birth, gender, mobile number and password to create an account. 3. User will click on the “Register” button. 4. System will check if the requirements are valid to create a new user. 5. The new user registration will be successful. 6. The login page of the application will be shown. |
| Alternative Flows: | AF-S4: If the email key in is not in the correct format   1. Error message: “Valid email is required” is displayed. 2. User return to Step 2 to re-enter a valid email.     AF-S4: If the password fails the checker check   1. Error message “Password too weak” is displayed. 2. User return to Step 2 to re-enter stronger password.     AF-S4: If the mobile number key in is not 8 digit   1. Error message “Please enter a valid phone number” is shown. 2. User return to Step 2 to re-enter a valid mobile number.     AF-S4: If the email is already in use   1. Error message “User already registered with this email” is displayed. 2. User return to Step 2 to re-enter a valid email address.   AF-S4: If the password and confirm password not tally   1. Error message “Password key-in is not the same” is displayed. 2. User return to Step 2 to re-enter password and confirmation password. |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 9 | | |
| Use Case Name: | Reset Password | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can reset their password if they forget.. |
| Preconditions: | 1.User has registered an account. |
| Postconditions: | 1. User’s password will reset. 2. User will be able to login to the application. |
| Priority: | - Low. |
| Frequency of Use: | - Only when the user forgets his/her password, which is rare in most cases. |
| Flow of Events: | 1.User clicks on “FORGOT PASSWORD?” in the login page.  2.User will be prompted to enter his/her registered email address.  3. User clicks on “RESET PASSWORD” and an email will be sent to the user’s registered email.  4. User will receive the reset password instructions which contains a link that allows users to enter their new password. |
| Alternative Flows: | AF-S3: If user enter invalid email   1. Error message “User does not exist or is no longer available” is displayed. 2. User return to Step 2 to re-enter valid email address |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | User has access to the email linked to his/her account. |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 10 | | |
| Use Case Name: | Upload profile picture | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor), Database API |
| Description: | User can upload/update their profile picture. |
| Preconditions: | 1. User login successfully. 2. User has downloaded the pictures of interest. |
| Postconditions: | 1. User’s profile picture will be uploaded successfully. |
| Priority: | - Low. |
| Frequency of Use: | - Whenever the user wants to change the profile picture |
| Flow of Events: | 1. User clicks on the “Settings” button. 2. The settings page will be loaded and the user can click on the ‘Upload profile picture’ button. 3. User will be directed to a “Upload your profile picture” page with the current profile picture shown. 4. Upon clicking the “Choose picture” button, the user can select a picture from the downloads page. 5. User will be directed back to the “Upload your profile picture” page with the chosen profile picture. 6. User clicks on the ‘Upload’ button to confirm. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 11 | | |
| Use Case Name: | Change Password | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | User can change their password. |
| Preconditions: | 1. User login successfully. |
| Postconditions: | 1. “Password reset successful” message will be printed and the user will be directed back to the settings page. |
| Priority: | - Low. |
| Frequency of Use: | - Whenever they want to change the password. |
| Flow of Events: | 1. User clicks on the “Settings” button. The settings page will be loaded. 2. Upon clicking the “Change password” button, a “Change your password” page will be loaded. 3. User will enter his/her new password and click on the ‘Proceed’ button to confirm. 4. Application system will then verify if the password is valid and update the new password. |
| Alternative Flows: | AF-S4:If the user inputs a password that does not contain at least 1 upper case character and one special character   1. Error message “Password enter is too weak, please try again” will be displayed. 2. The user will be redirected back to Step 3 to re-enter a valid password.   AF-S4:If the user does not input a password   1. Error message “New password field is left empty” will be displayed. 2. The user will be redirected back to Step 3 to re-enter a valid password. |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 12 | | |
| Use Case Name: | Change Username | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor),Database API |
| Description: | Users can change their username |
| Preconditions: | 1.User login successfully. |
| Postconditions: | 1. “Username updated” message will be printed and the user will be directed back to the settings page. |
| Priority: | - Low |
| Frequency of Use: | - Whenever the user wants to change the username. |
| Flow of Events: | 1.User clicks on the “Settings” button. The settings page will be loaded.  2.Upon clicking the “Change username” button, a “Change your username” page will be loaded.  3.User will enter his/her new username and click the “Proceed” button to confirm. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 13 | | |
| Use Case Name: | Delete Account | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User (Initiating actor), Database API |
| Description: | Users can delete their user account. |
| Preconditions: | 1.User login successfully. |
| Postconditions: | 1.User’s account will be deleted and will be directed back to the Welcome Page of the app.  2. Message “Account Deleted” will be shown to user |
| Priority: | - Low. |
| Frequency of Use: | - Not Frequent. |
| Flow of Events: | 1.User clicks on the “Settings” button. The settings page will be loaded.  2.Upon clicking the “Delete Account” button, a “Are you sure” page will be loaded.  3.User will click on the “Delete” button to confirm. |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Includes: | N/A |
| Special Requirements: | N/A |
| Assumptions: | N/A |
| Notes and Issues: | NIL |