**Use Cases**

**for**

**WITFIT**

**Version 0.2**

**Prepared by WomenInTech**

**WomenInTech (CZ2006 BS3)**

**6 Feb 2022**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| WomenInTech | 31/01/2022 | Initial documentation | 0.1 |
| WomenInTech | 06/02/2022 | Merging and Completion of Initial Draft | 0.2 |

[**UC001**](#_heading=h.1xqdgfmnc59t) **4**

[**Create main app account**](#_heading=h.eyf4o3tpvkiv) **4**

[**UC002**](#_heading=h.v0sm89gkeq) **6**

[**Login to main app account**](#_heading=h.7ipxf7oc5wje) **6**

[**UC003**](#_heading=h.whiuxyaac5m8) **7**

[**View gym map**](#_heading=h.4irrgwujgo0g) **7**

[**UC004**](#_heading=h.98dgh2vxvr91) **8**

[**Search for gyms**](#_heading=h.47yvgys8un1h) **8**

[**UC005**](#_heading=h.y95yid159yid) **9**

[**View gym directions**](#_heading=h.k80mpz9pxmnd) **9**

[**UC006**](#_heading=h.ikel3yy165bn) **10**

[**View gym details**](#_heading=h.tiudakq66qal) **10**

[**UC007**](#_heading=h.d5tkldu4iul9) **12**

[**View gym reviews**](#_heading=h.jpt0tdb8qez2) **12**

[**UC008**](#_heading=h.nl4rdbunjiu7) **13**

[**Submit review**](#_heading=h.vsjy0ro0yz8j) **13**

[**UC009**](#_heading=h.9zrmddo5px3) **15**

[**Update personal information**](#_heading=h.mjmb66azrmbz) **15**

[**UC010**](#_heading=h.sr5yfzwhad2u) **16**

[**Select fitness goals**](#_heading=h.a1213hdhxakh) **16**

[**UC011**](#_heading=h.vke0jqniznov) **17**

[**Recommend personalised workout**](#_heading=h.o2ks597gczpl) **17**

[**UC012**](#_heading=h.ren0t5fgb99d) **18**

[**Search for exercises**](#_heading=h.1ii3unvz54ut) **18**

[**UC013**](#_heading=h.covuhl9lgfra) **19**

[**Add exercises**](#_heading=h.5jzsyx6bz188) **19**

[**UC014**](#_heading=h.3bj13he13ywb) **20**

[**Conduct workout**](#_heading=h.46xpgzkws2c9) **20**

[**UC015**](#_heading=h.6eh0zkppppxz) **21**

[**Create customised workout routine**](#_heading=h.qyjle2dnymfw) **21**

[**UC016**](#_heading=h.3tn0znbv2nc5) **22**

[**View exercise details**](#_heading=h.whvkn59wy5) **22**

[**UC017**](#_heading=h.6a4u6ftz9nuo) **23**

[**Pick your music**](#_heading=h.uw2jqsu9oorq) **23**

[**UC018**](#_heading=h.v4z1hksxfdnv) **24**

[**Display workout summary**](#_heading=h.bfr0fapukszf) **24**

[**UC019**](#_heading=h.7ovlvzzfczdg) **25**

[**Add workout to progress calendar**](#_heading=h.chgqtqvwdw31) **25**

[**UC020**](#_heading=h.3fwu6wvvxvx8) **26**

[**Display workout history**](#_heading=h.8jruzj325lgp) **26**

[**UC021**](#_heading=h.2y9rtioyqadk) **27**

[**Create gym buddy account**](#_heading=h.ddnns2geb821) **27**

[**UC022**](#_heading=h.58q5znefo1bz) **28**

[**Login to gym buddy account**](#_heading=h.pnu93k61e6bq) **28**

[**UC023**](#_heading=h.n0d157fmt5dy) **29**

[**Update account preferences**](#_heading=h.nikb339lg9xw) **29**

[**UC024**](#_heading=h.tip88ib7c0f4) **30**

[**View suggested buddies**](#_heading=h.m6975k44lcin) **30**

[**UC025**](#_heading=h.rm784jfosqko) **31**

[**View specific buddy details**](#_heading=h.q1l6cur9e246) **31**

[**UC026**](#_heading=h.lvq8ox5y9a8l) **32**

[**Request to pair up with buddy**](#_heading=h.qx4ffa6yexmm) **32**

[**UC027**](#_heading=h.do8b8xlf8sx0) **33**

[**View suggested buddies photo**](#_heading=h.uuim2tff49dv) **33**

[**UC028**](#_heading=h.es24foleve8u) **34**

[**Chat with buddy**](#_heading=h.z0hzsf115cl1) **34**

[**UC029**](#_heading=h.amuztgppa6w) **35**

[**Remove buddy**](#_heading=h.lv74jb1opa8e) **35**

[**UC030**](#_heading=h.9133zt7yae5q) **36**

[**View all chats**](#_heading=h.dx1funtvpjz6) **36**

| Use Case ID: | UC001 | | |
| --- | --- | --- | --- |
| Use Case Name: | Create main app account | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating actor), App Database |
| --- | --- |
| Description: | First time users are to create a WITFIT account to use the application. |
| Preconditions: | 1. This use case extends UC002 Login to the main app account. 2. User does not have an existing account. 3. Users must have an email. 4. An internet connection is required. |
| Postconditions: | 1. User has an account created with a username. 2. Users will be able to see the homepage of their account. 3. The account information will be stored in the app database. |
| Priority: | High.  Users will not be able to login to access any of the application functionalities without an account. |
| Frequency of Use: | Rarely..  The first time the user uses the app. |
| Flow of Events: | 1. User clicks “I don’t have an account” 2. Sign up page will be displayed 3. Users will be prompted to enter First Name, Last Name, Email, Birthday, Gender, Username, Password and Confirm Password before pressing the signup button. 4. If the user’s credentials are valid, the user will be redirected to a page which will prompt the user for their height and weight. 5. If the user’s height and weight are valid, the app will be redirected to the next page. 6. The user will be prompted about whether they have any physical injuries or pre-existing health conditions. 7. If the user answers “No” to both questions in Step 6, the user will be redirected to the page on fitness goals. 8. The user will be prompted to enter their fitness goals and will click next to confirm. 9. Upon clicking next, the user will be redirected to a welcome page, which is indicative of a successful account creation. |
| Alternative Flows: | AF-S4: If the user’s credentials are already in use.   1. App displays the error message “Username already in use“. 2. The app returns to step 3.   AF-S5: If the user’s height is invalid   1. App displays the error message “Please enter a valid height” 2. The app returns to step 4.   AF-S5: If the user’s weight is invalid   1. App displays the error message “Please enter a valid weight” 2. The app returns to step 4.   AF-S7: If the user answers “Yes” they have physical injuries.   1. The app will redirect the user to a page to know more about their physical injuries. 2. The user will be prompted to enter their type of injury. 3. The app will proceed to Step 8.   AF-S7: If the user answers “Yes” they have pre-existing medical conditions.   1. The app will redirect the user to a page to know more about their pre-existing medical condition. 2. The user will be prompted to enter their medical condition. 3. The app will proceed to Step 8.   AF-S7: If the user answers “Yes” to both questions.   1. The app will redirect the user to a page to know more about their physical injuries. 2. The user will be prompted to enter their type of injury. 3. Upon clicking next, the user will proceed to the next page. 4. The user will be prompted to enter their pre-existing medical condition. 5. Upon clicking next, the app will proceed to step 8.   AF-S9: Upon clicking next, if the user is a female.   1. The user will be prompted to enter the date of their last period. 2. The user can select “Not sure” or “I am pregnant”. 3. The user selects “FINISH” 4. The app will proceed to step 9. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | The user’s email address is valid. |
| Notes and Issues: | - |

| Use Case ID: | UC002 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login to main app account | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User(Initiating Actor), App Database |
| --- | --- |
| Description: | User logs into app using their WITFIT account |
| Preconditions: | 1. Users must have an existing account for successful login. 2. An internet connection is required. |
| Postconditions: | 1. App will store the user's login information. 2. Users will see a welcome homepage. |
| Priority: | High.  Use cases UC003 View gym map, UC006 View gym details, UC008 Submit review, UC009 Update personal information, UC014 Conduct workout, UC015 Create customised workout routine, UC018 Display workout summary and UC021 Create gym buddy account will not be available without logging in. |
| Frequency of Use: | Rarely.  The first time the user uses the app.  The only instance other than this is when the user chooses to sign out. |
| Flow of Events: | 1. User launches the app. 2. The app prompts the user for their username and password. 3. User enters username and password. 4. User presses the login button. 5. If the user's credentials are valid, the app logs the user in and displays a welcome message, indicating successful login. |
| Alternative Flows: | AF-S4: If user’s credentials are invalid   1. App displays invalid password/username. 2. The app returns to step 2.   AF-S3: User presses “Forget Password”   1. App prompts the user for their email. 2. The app will send the user instructions via email to reset the password. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | The user has a valid email address.  The app will send a temporary “new password” to the email which the user can use. |
| Notes and Issues: | The app does not limit the number of attempts the user has to input their credentials. |

| Use Case ID: | UC003 | | |
| --- | --- | --- | --- |
| Use Case Name: | View gym map | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (initiating actor), Maps API |
| --- | --- |
| Description: | The user will use this function to view all the gyms on a map. |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login. 2. Invocation as an inclusion of use case UC004 Search for gyms. 3. Users must have enabled location sharing with the application. 4. An internet connection is required. |
| Postconditions: | The app must display an interactive map with the locations of various gyms marked. |
| Priority: | High.  Having an interactive map of the locations of gyms would provide users with an aerial, top-down view of their gym choices at a glance. |
| Frequency of Use: | High.  The interactive map will occupy considerable space on the UI and provide valuable information to users. |
| Flow of Events: | 1. The app uses the included use case UC002 Login to log the user into the application. 2. On successful login, the app retrieves information about all gyms in Singapore, such as their name and geographic location from the Maps API. 3. The user uses the included use case UC004 Search for gyms to filter the list of gyms. 4. The app displays each gym in the list of gyms as markers on the map. 5. The app displays the user and a circle indicating the search radius on the map. 6. If the user selects a gym on the map, the app invokes the extended use case UC005 View gym directions. |
| Alternative Flows: | AF-S1: If location sharing is not accessible   1. The app will prompt the user to enable location sharing permissions with the app 2. The app returns to step 1. |
| Exceptions: | EX1: If the Maps API is down   1. The app displays the message ‘The Maps API is down’. 2. The map displays a static image of Singapore. |
| Includes: | UC002 Login, UC004 Search for gyms. |
| Special Requirements: | - |
| Assumptions: | The location data returned by the Maps API for all the gyms is valid and accurate.  The user’s location sharing functionality is accurate. |
| Notes and Issues: | - |

| Use Case ID: | UC004 | | |
| --- | --- | --- | --- |
| Use Case Name: | Search for gyms | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User(Initiating actor), Maps API, App Database |
| --- | --- |
| Description: | Filter the gyms based on user input. |
| Preconditions: | 1. Users must enable location sharing with the application. 2. An internet connection is required. |
| Postconditions: | The app must modify the list of gyms to only include those that satisfy the user’s search criteria and display them accordingly. |
| Priority: | High.  Being able to filter search results is important in bringing value to our users by showing only the most relevant data. |
| Frequency of Use: | High.  This functionality will always be invoked by use cases UC003 View gym map, UC006 View gym details, UC008 Submit review. |
| Flow of Events: | 1. The app inputs a default value of 5km for the distance filter. 2. The app filters the list of gyms based on the defined distance filter. 3. The app inputs a default value of ‘closest’ for the sort filter. 4. The app sorts the list of gyms based on the defined sort filter. 5. If the user inputs text into the search bar, the app filters the list of gyms based on the search term. |
| Alternative Flows: | AF-S1: If user selects the Distance filter   1. The app will display a dropdown of distances to select, from 1-10. 2. The user will make their selection. 3. The app returns to step 2.   AF-S3: If user selects the sort filter   1. The app will display a dropdown of sorts to select: ‘closest’, ‘alphabetical ascending’ ‘alphabetical descending’, ‘highest reviews’ 2. The user will make their selection. 3. The app returns to step 4. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | The location data returned by the Maps API for all the gyms is valid and accurate.  The user’s location sharing functionality is accurate. |
| Notes and Issues: | The sorting for highest reviews will rely on data fetched directly from the app database instead of Map sAPI. |

| Use Case ID: | UC005 | | |
| --- | --- | --- | --- |
| Use Case Name: | View gym directions | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User(Initiating actor), Maps API |
| --- | --- |
| Description: | Provide directions from the user’s current location to the destination gym |
| Preconditions: | 1. This use case extends UC003 View gym map and use case UC006 View gym details. 2. Invocation as an inclusion of use case UC003 View gym map. |
| Postconditions: | App must display the recommended path(s) to travel from the source to the destination on a map. |
| Priority: | High.  Being able to show users how to get to their desired destination is an extremely important feature that will make it easier for users to initiate their workout, which is one of the main purposes of the app. |
| Frequency of Use: | High.  This functionality will often be invoked by use cases UC003 View gym map and UC006 View gym details. |
| Flow of Events: | 1. The app retrieves the possible paths from the user’s current location to the desired destination from the Maps API. 2. The app displays all the possible paths as options. 3. The app selects the preferred method of travel based on the recommended path from the Maps API. 4. The app uses the included use case UC003 View gym map to display the recommended path. |
| Alternative Flows: | AF-S2: If user selects the preferred method of travel   1. The app retrieves the user selection. 2. The app updates the preferred method of travel accordingly. 3. The app returns to step 4. |
| Exceptions: | EX1: If the Maps API is down   1. The app displays the message ‘The Maps API is down’. 2. The map displays a static image of Singapore. |
| Includes: | UC003 View gym map |
| Special Requirements: | - |
| Assumptions: | The location data returned by the Maps API for all the gyms is valid and accurate.  The user’s location sharing functionality is accurate. |
| Notes and Issues: | The app shall display only a visual of the recommended path. If the user wishes to navigate to the gym, the app shall prompt the user to open Google Maps or a preferred navigation app of the user’s choice. |

| Use Case ID: | UC006 | | |
| --- | --- | --- | --- |
| Use Case Name: | View gym details | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (initiating actor), Maps API |
| --- | --- |
| Description: | The user can use this functionality to view more detailed information about the gym. |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. Invocation as an inclusion of use case UC004 Search for gyms. 3. Users must have enabled location sharing with the application. 4. Users must enable clipboard access with the application. 5. An internet connection is required. |
| Postconditions: | The app must display an interactive list of gyms with their details. |
| Priority: | Medium.  Having an interactive list of gyms with their details would provide users with knowledge of their gym choices at a glance. |
| Frequency of Use: | Medium  The interactive list will serve to supplement use case UC003 View gym map, as a more detailed way to display valuable information to users. |
| Flow of Events: | 1. The system uses the included use case UC002 Login to the main app account to log the user into the application. 2. On successful login, the app retrieves information about all gyms in Singapore, such as their name, address, and operating hours from the Maps API. 3. The user uses the included use case UC004 Search for gyms to filter the list of gyms displayed. 4. If the user selects the ‘copy to clipboard’ text on a specific gym, the app will copy the gym’s information to the user’s clipboard. 5. If the user selects the ‘see reviews’ text on a specific gym, the app invokes the extended use case UC007 View gym reviews. 6. If the user selects the ‘get directions’ text on a specific gym, the app invokes the extended use case UC005 View gym directions. |
| Alternative Flows: | AF-S1: If location sharing is not accessible   1. The app will prompt the user to enable location sharing permissions with the app 2. The app returns to step 1.   AF-S4: If clipboard is not accessible   1. The app will prompt the user to enable clipboard access with the app 2. The app returns to step 4. |
| Exceptions: | EX1: If the Maps API is down   1. The app displays the message ‘The Maps API is down’ 2. The map displays a static image of Singapore |
| Includes: | UC002 Login to main app account, UC004 Search for gyms |
| Special Requirements: | - |
| Assumptions: | The location data returned by the Maps API for all the gyms is valid and accurate.  The user’s location sharing functionality is accurate. |
| Notes and Issues: | - |

| Use Case ID: | UC007 | | |
| --- | --- | --- | --- |
| Use Case Name: | View gym reviews | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User(Initiating actor)  App Database |
| --- | --- |
| Description: | Retrieve the review details for the different gyms. |
| Preconditions: | 1. This use case extends UC006 View gym details. 2. An internet connection is required. |
| Postconditions: | The app must display the review information of all gyms |
| Priority: | Low  As most gyms will offer enough equipment for users to fulfil their workout routine, reviews will mostly be for quality of life and minor details about the gym and its absence should not majorly impede our app in serving its purpose. |
| Frequency of Use: | Medium  This functionality will frequently be invoked by use case UC006 View gym details. |
| Flow of Events: | 1. The app retrieves a list of all reviews from App Database 2. The app calculates the average total rating from 1 to 5 stars by gym 3. The app displays this average total rating. 4. The app calculates the total number of reviews in each rating scale from 1 to 5 stars by gym. 5. The app displays this distribution of the total number of reviews in each rating scale. 6. The app inputs a default value of ‘Most Recent’ for the Sort filter. 7. The app sorts the list of reviews based on the defined Sort filter. |
| Alternative Flows: | AF-S6: If user selects the Sort filter   1. The app will display a dropdown of sorts to select: ‘most recent’, ‘highest rating’ 2. The user will make their selection. 3. The app returns to step 7. |
| Exceptions: | EX1: No reviews exist for a gym   1. The app displays the message ‘No reviews for this gym so far.’ 2. The app displays a button to submit a review. 3. If the button is pressed, use case UC008 Submit review is invoked. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC008 | | |
| --- | --- | --- | --- |
| Use Case Name: | Submit review | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (initiating actor)  App Database |
| --- | --- |
| Description: | Writing and submitting a review for a gym. |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. Invocation as an inclusion of use case UC004 Search for gyms. 3. An internet connection is required. |
| Postconditions: | The user’s review is submitted and stored in the App Database as a new entry.  These reviews must be displayed in future uses of UC007 View Gym Reviews. |
| Priority: | Low  As most gyms will offer enough equipment for users to fulfil their workout routine, reviews will mostly be for quality of life and minor details about the gym and its absence should not majorly impede our app in serving its purpose. |
| Frequency of Use: | Low  This functionality will only be invoked when the user takes the initiative to submit a review for the benefit of other users. |
| Flow of Events: | 1. The app uses the included use case UC002 Login to log the user into the application. 2. On successful login, the app uses the included use case UC004 Search for gyms to prompt the user to search for and select the gym they wish to review. 3. The app displays the user’s username, today’s date, and the selected gym. 4. The app prompts the user to select a review rating from 1 to 5 stars. 5. The user selects a rating. 6. The app displays the selected rating. 7. The app prompts the user to write a comment for the review. 8. The user writes a comment. 9. The app displays the comment. 10. The user selects the ‘Post review’ button 11. The app displays a summary of the review. 12. The app displays a confirmation button. 13. If the user selects ‘Yes’, the review is stored as a new record on App Database. |
| Alternative Flows: | AF-S8.1: If the user does not write a comment   1. The app displays a message to the user: ‘You have not written a comment. Users are highly encouraged to leave a comment’. 2. The app displays an acknowledgement button. 3. The user presses ‘Ok’ on the acknowledgement button. 4. The app returns to step 10.   AF-S8.2: If the user exceeds 1024 characters in the comment   1. The app displays a message to the user: ‘Your comment has exceeded the word limit. Please keep to within 1024 characters.’ 2. The app displays an acknowledgement button. 3. The user presses ‘Ok’ on the acknowledgement button. 4. The app returns to step 8. |
| Exceptions: | EX1: The review is unable to be added to App Database   1. The app displays the message ‘Sorry, there was an error submitting your review. Please try again later.’ |
| Includes: | UC002 Login to main app account, UC004 Search for Gyms |
| Special Requirements: | - |
| Assumptions: | Included use cases UC002 Login to main app account and UC004 Search for gyms are functional. |
| Notes and Issues: | - |

| Use Case ID: | UC009 | | |
| --- | --- | --- | --- |
| Use Case Name: | Update personal information | | |
| Created By: | Gabriel Tang | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description: | User logs into the app and input personal information |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. An internet connection is required. |
| Postconditions: | App stores update name and profile picture in the App Database |
| Priority: | Low.  The user can still use all other functionalities without updating personal information. |
| Frequency of Use: | Low.  This function will only be invoked if the user wants to edit their personal information. |
| Flow of Events: | 1. The user clicks “Edit Profile” 2. The app will display pre-filled information. 3. The app prompts the user to edit their personal information. 4. If the user wishes to changes their fitness goals, the app invokes the extended use case UC009 Select fitness goals 5. The user clicks on confirm settings after all fields are filled. 6. The app displays a confirmation message at the user profile, indicating that the update of personal information is successful. |
| Alternative Flows: | AF-S5: User clicks on confirm settings when not all fields are filled   1. The app will display a red text and prompt the user to fill up all fields. 2. The app will return to step 3. |
| Exceptions: | - |
| Includes: | UC002 Login to main app account |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID:⠀ | UC010 | | |
| --- | --- | --- | --- |
| Use Case Name:⠀ | Select fitness goals | | |
| Created By:⠀ | Gabriel Tang | Last Updated By: | Foo Zhi Kai |
| Date Created:⠀ | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description: | User uses the app and selects his/her fitness goals |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. This use case extends UC009 Update personal information 3. An internet connection is required |
| Postconditions: | The app shall update the user’s selected fitness goals into the app database. |
| Priority: | Medium.  This functionality allows users to define and work towards their own objectives, an important factor in fitness ensuring continued motivation and discipline in achieving one’s fitness goal. |
| Frequency of Use: | Low.  This function will only be invoked if the user chooses to change their fitness goals. Users will not change their fitness goals frequently. |
| Flow of Events: | 1. The app will display the pre-selected fitness goal. 2. The app will display a list of available fitness goals. 3. The app prompts the user to select a new fitness goal. 4. If the user has selected a new fitness goal, the user can click on confirm settings. 5. The app displays a confirmation message at the user profile, indicating that the update of fitness goals is successful. |
| Alternative Flows: | AF-S1: If the user does not have a preselected fitness goal   1. The app will display ‘No fitness goal selected’ 2. The app will return to step 2.   AF-S4: If the user did not select a new fitness goal   1. The app will display an option to return back to the previous step. 2. If the user selects the option to return back to the previous step, the app will return to UC009. |
| Exceptions: |  |
| Includes: | UC002 Login to main app account |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC011 | | |
| --- | --- | --- | --- |
| Use Case Name: | Recommend personalised workout | | |
| Created By: | Gabriel Tang | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User, App Database |
| --- | --- |
| Description: | User selects a recommended personalised workout that will be generated by the app |
| Preconditions: | 1. This use case extends UC015 Create customised workout routine 2. Invocation as an inclusion of use case UC002 Login to main app account. 3. An internet connection is required |
| Postconditions: | 1. User will see their overall workout selected (Duration + exercises) 2. View past workouts |
| Priority: | High.  User selects workout based on their personal information input hence it is important for the app to recommend a workout that is suitable for the user |
| Frequency of Use: | Medium.  Everytime the user uses the app. |
| Flow of Events: | 1. The app displays recommended personalised workout 2. User then selects recommended music 3. If user selects customise workout, app prompt user to add exercise from database 4. Once completed, user taps on “Start Workout“ 5. If the operation was a success, a toast message will appear, indicating the success/start 6. Workout begins with timer 7. After workout is over, app displays view past workout option |
| Alternative Flows: | AF-S1: If the operation fails   1. A toast message will be displayed indicating the error 2. App returns to step 1 |
| Exceptions: | - |
| Includes: | UC002 Login to main app account |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC012 | | |
| --- | --- | --- | --- |
| Use Case Name: | Search for exercises | | |
| Created By: | Bryan Leow | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (initiating actor), App Database |
| --- | --- |
| Description: | Filter our database of exercises based on a search criteria. |
| Preconditions: | 1. This use case extends UC013 Add exercises or UC016 View exercise details. 2. An internet connection is required |
| Postconditions: | A filtered list of exercises matching the search criteria is returned to the calling use case. |
| Priority: | High.  Being able to search our database for exercises is crucial as it enables users to select the exercises they desire, whether due to personal preference in-line with their training goals or due to previous injuries preventing some exercises from being carried out. |
| Frequency of Use: | High.  It is invoked as an extension of two other highly used use cases UC013 Add exercises and UC016 View exercise details |
| Flow of Events: | 1. The app invokes this use case as an extension of one of two other use cases, UC013 Add exercises or UC016 View exercise details 2. The app parses the search criteria passed to it. 3. The app retrieves the filtered list of exercises from the App Database. 4. The app returns the filtered list to the use case that invoked it. |
| Alternative Flows: | AF-S2: If the search criteria cannot be parsed:   1. The user is prompted with the message ‘The app does not understand your search’. 2. The app sets the filtered list to be empty. 3. The app returns to step 4. |
| Exceptions: | EX1: The App Database cannot be accessed:   1. The user is prompted with the message ‘The app database is currently unavailable, please try again later’ |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC013 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add exercises | | |
| Created By: | Bryan Leow | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description: | User selects exercise to add from workout API |
| Preconditions: | 1. This use case extends UC012 Search for exercises |
| Postconditions: | The exercise is added to a customised workout routine and stored in the App Database. |
| Priority: | Medium.  This functionality allows users to customise their own workout routines if they wish to do so instead of following the recommended workout. Customisation allows users to better tailor the app towards achieving their fitness goals. |
| Frequency of Use: | Medium.  This functionality will only be invoked when users have decided to customise their workout, and want to add a specific exercise. |
| Flow of Events: | 1. The app invokes this use case as an extension of use case UC012 Search for exercises, when the user selects the ‘Add exercise’ button. 2. The app obtains information about the selected exercise from use case UC012 Search for exercises and information about the selected workout routine from use case UC015 Create customised workout routine. 3. The app updates the customised workout routine with the selected exercise in the App Database. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC014 | | |
| --- | --- | --- | --- |
| Use Case Name: | Conduct workout | | |
| Created By: | Bryan Leow | Last Updated By: | Bryan Leow |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (initiating actor), App Database |
| --- | --- |
| Description: | Facilitates the overall flow of a workout routine. |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. This use case extends UC011 Recommend personalised workout |
| Postconditions: | A workout routine is completed. |
| Priority: | High.  This is one of the key functionalities of the app to facilitate users to exercise and is thus extremely important. |
| Frequency of Use: | High.  Users will almost always use this functionality when wanting to exercise, which ranges from 3-7 times per week per user. |
| Flow of Events: | 1. On successful login, the app uses the included use case UC011 Recommend personalised workout to generate a workout routine for the user. 2. The app displays information about the workout routine, such as title, description, duration, and location. 3. Users press ‘start workout’ to begin the workout 4. The app displays a warm-up video from the App Database, along with a timer. 5. Users press ‘start warm up’ to begin the warm-up. 6. Users press ‘finish warm up’ to end the warm-up. 7. The app uses the extended use case UC016 View exercise details to conduct each exercise in the routine. |
| Alternative Flows: | AF-S2:   1. The user selects their own customised workout routine. 2. The app returns to step 3. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | UC002 Login to main app account, UC011 Recommend personalised workout |
| Notes and Issues: | - |

| Use Case ID: | UC015 | | |
| --- | --- | --- | --- |
| Use Case Name: | Create customised workout routine | | |
| Created By: | Gladys Loh | Last Updated By: | Gladys Loh |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating actor), Workout API |
| --- | --- |
| Description: | Users are able to customise a workout and edit their recommended routine. |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. |
| Postconditions: | The app will update the workout routine according to what the user specified to the app database. |
| Priority: | Medium.  This is a main functionality - that is to allow users to have creativity in customisation to suit their needs. |
| Frequency of Use: | Medium.  This function will only be invoked if users want to edit their customised routine. |
| Flow of Events: | 1. After successful login in UC002 Login to the main app account, the user can select to customise their workout routine. 2. The user can make changes to the exercises, sets or reps. 3. If the user selects “Search for exercises to add”, UC012 Search for exercises will be invoked. |
| Alternative Flows: | AF-S1: After completing a workout, the user can edit their customised workout.   1. The app will proceed to Step 2.   AF-S3: If the user selects “Recommend me a personalised workout”.   1. UC011 Recommend personalised workout will be invoked. |
| Exceptions: | - |
| Includes: | UC002 Login to main app account |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC016 | | |
| --- | --- | --- | --- |
| Use Case Name: | View exercise details | | |
| Created By: | Joshua Khoo | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), Workout API |
| --- | --- |
| Description: | User is able to view exercise details before and after the workout |
| Preconditions: | 1. This use case extends UC012 Search for exercises 2. An internet connection is required |
| Postconditions: | The app shall display detailed information about the exercise selected, such as the recommended number of sets and reps. |
| Priority: | High.  Users need to be able to view exercise details before they can start a workout. |
| Frequency of Use: | High.  This functionality will be invoked in UC014 Conduct workout, this functionality will be used each time the user wants to start a workout. |
| Flow of Events: | 1. The app fetches videos from the Workout API. 2. The app then displays the specific exercise details that are part of the recommended personalised workouts. 3. User views the details of the recommended exercises. 4. If the user selects to start workout after reading through the details, the app invokes the extended use case UC014 Conduct Workout. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC017 | | |
| --- | --- | --- | --- |
| Use Case Name: | Pick your music | | |
| Created By: | Gladys Loh | Last Updated By: | Gladys Loh |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating actor), Music API |
| --- | --- |
| Description: | Users are recommended a workout music playlist but are allowed to change if they want. |
| Preconditions: | 1. This use case extends UC015 Create customised workout routine 2. The user must be connected to their spotify account. 3. An internet connection is required. |
| Postconditions: | The app must allow the user to select music. |
| Priority: | Low.  This feature is merely to boost user experience. |
| Frequency of Use: | Medium.  High likelihood that users would want to play music during their workout. |
| Flow of Events: | 1. The app will collate relevant information such as workout intensity from the customised workout created from UC015 Create customised workout routine 2. The app fetches relevant music data such as bpm and name of the song from the Music API. 3. The app uses a matching algorithm to match the exercises with music choices. 4. The app creates a personalised recommended music playlist. 5. The app will prompt the user if they wish to conduct the workout with the recommended music playlist. 6. If Yes, the app will return to UC014 Conduct workout along with the recommended music playlist. |
| Alternative Flows: | AF-S6: If the user selects No,   1. The app will prompt the user to edit their recommended music playlist. 2. If the user chooses to edit the playlist, the app will launch the spotify app and allow them to make edits to the playlist. 3. If the user chooses not to edit, the app will return to UC014 Conduct workout without any music playlist selected. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | User has a valid spotify account. |
| Notes and Issues: | - |

| Use Case ID: | UC018 | | |
| --- | --- | --- | --- |
| Use Case Name: | Display workout summary | | |
| Created By: | Foo Zhi Kai | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating actor), App Database |
| --- | --- |
| Description: | Display summary of completed workout |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. Invocation as an inclusion of use case UC014 Conduct workout. 3. An internet connection is required. |
| Postconditions: | The app must display a summary of the workout |
| Priority: | Medium.  A key component of the user experience. |
| Frequency of Use: | Medium.  The summary will be displayed every time the user completes a workout. |
| Flow of Events: | 1. The user uses the included use case UC014 to start conducting a workout. 2. The user completes the workout. 3. On successful completion, the app displays a summary of the workout. 4. The app will display the time and date the workout was completed 5. The app will also display some basic statistics like total sets, total calories, and duration of workout |
| Alternative Flows: | AF-S3: If the workout was done at the gym,   1. The app will prompt the user to submit a review for the gym 2. If the user selects the option to select to submit the review, the app will return to UC008 Submit review |
| Exceptions: | - |
| Includes: | UC002 Login to main app account, UC014 Conduct workout |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC019 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add workout to progress calendar | | |
| Created By: | Foo Zhi Kai | Last Updated By: | Foo Zhi Kai |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description: | User adds workout to their progress calendar |
| Preconditions: | 1. This use case extends UC018 Display workout summary. 2. An internet connection is required. |
| Postconditions: | 1. App stores workout summary into user’s progress calendar 2. App displays an updated progress calendar |
| Priority: | Medium.  A key component of the user experience and motivates the user to continue using the app. |
| Frequency of Use: | Low  This functionality will only be invoked when the user chooses to add their workout to their progress calendar whenever the user completes a workout. |
| Flow of Events: | 1. The app uses the included use case UC018 when the user has completed their workout. 2. The app will prompt the user if the user wishes to add the workout to their progress calendar. 3. The app calendar will display the date the workout was done. 4. The app calendar will display the workout summary displayed in use case UC018 that was completed on that date. |
| Alternative Flows: | AF-S2: If the user does not wish to add the workout,   1. The app will return to UC2018.   AF-S4: If the user is a female,   1. The app will retrieve information regarding the user’s menstruation cycle from the App Database 2. The app will display this information on the App Calendar |
| Exceptions: | - |
| Includes: |  |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC020 | | |
| --- | --- | --- | --- |
| Use Case Name: | Display workout history | | |
| Created By: | Joshua Khoo | Last Updated By: | Gabriel Tang |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description: | User is able to access and view past workouts |
| Preconditions: | 1. This use case extends UC019 Add workout to progress calendar 2. The user has done at least a workout |
| Postconditions: | 1. The app must display a summary of past workouts 2. The app must display a history of all previous workouts |
| Priority: | Medium.  A key component for users to track their progress. This functions as an important motivating force for them to continue using the app. |
| Frequency of Use: | Low.  The workout history will only be displayed when the user selects this option on the progress calendar. Users are unlikely to check their history often. |
| Flow of Events: | 1. The app will retrieve information regarding the user’s previous workouts from the App Database 2. The app will display a summary of past workouts using the information retrieved 3. The app will display a history of past workouts using the information retrieved |
| Alternative Flows: | - |
| Exceptions: | EX1 User has not done any workouts   1. The app will display an error message “You have not performed any workout”. 2. The app will prompt them to conduct a workout. 3. If the user selects the option to conduct a workout, the app will return to UC014. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC021 | | |
| --- | --- | --- | --- |
| Use Case Name: | Create gym buddy account | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | Create user profile for gym buddy application function |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. This use case extends UC022 Login to gym buddy account 3. The user must have an internet connection. |
| Postconditions: | The app will store the user gym buddy data in the application database.  The user will be interfaced to the homepage, indicating successful account creation. |
| Priority: | High.  Users will not be able to login to access any of the “find a gym buddy” functionalities without an account. |
| Frequency of Use: | Once.  The first time the user tries to use the “find a gym buddy” functionality. |
| Flow of Events: | 1. The user clicks on the sign up for gym buddy button 2. The app prompts the user for a brief introduction, their preferred workout time, preferred buddy gender, their goals, areas of expertise, style of training and preference of training location. The app also prompts the user to upload a photo of themselves 3. The user clicks on create buddy profile only after all fields are filled. 4. The app displays the user’s gym buddy profile, indicating successful signup. |
| Alternative Flows: | AF-S3: User clicks on create buddy profile when not all fields are filled.   1. The app will display a red text, please fill up all fields. 2. The app returns to step 2. |
| Exceptions: | - |
| Includes: | UC002 Login to main app account. |
| Special Requirements: | - |
| Assumptions: | The user has unlimited tries to fill in all the fields. |
| Notes and Issues: | - |

| Use Case ID: | UC022 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login to gym buddy account | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | Login to gym buddy to access gym buddy functionalities |
| Preconditions: | 1. Invocation as an inclusion of use case UC002 Login to main app account. 2. The user must have an internet connection. 3. The user must have a gym buddy account. |
| Postconditions: | The app will store the user data in the application database. |
| Priority: | High.  Use cases UC023 Update account preferences, UC024 View suggested buddies, UC025 View specific buddy details, UC026 Request to pair up with buddy, UC027 View suggested buddies photo, UC028 Chat with buddy, UC029 Remove buddy and UC030 View all chats will not be available without logging in. |
| Frequency of Use: | Rarely.  This function will only be used if the user logs out from the “find a gym buddy” functionality. |
| Flow of Events: | 1. The user clicks on the gym buddy button. 2. The user selects ‘I already have an account’. 3. The app displays the user’s gym buddy profile, indicating successful login. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | UC002 Login to main app account |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | UC023 | | |
| --- | --- | --- | --- |
| Use Case Name: | Update account preferences | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | This functionality allows the user to update their account preferences with regards to their gym buddy account. |
| Preconditions: | 1. Invocation as an inclusion of use case UC022 Login to gym buddy account. 2. The user must have an internet connection. |
| Postconditions: | The app will update the user data that was edited in the application database. |
| Priority: | Low.  The user can still use all other functionalities in the “find a gym buddy” interface even without updating account preferences. |
| Frequency of Use: | Low.  This function will only be invoked if the user wants to edit their preferences. |
| Flow of Events: | 1. The user clicks “Edit Profile” 2. The app will display pre-filled information. 3. The app prompts the user to edit their brief introduction, their preferred workout time, preferred buddy gender, their goals, areas of expertise, style of training and preference of training location. The app also prompts the user to upload more photos of themselves. 4. The user clicks on confirm settings after all fields are filled. 5. The app displays a confirmation message at the user profile, indicating that the update of preferences is successful. |
| Alternative Flows: | AF-S4: User clicks on confirm settings when not all fields are filled   1. The app will display a red text and prompt the user to fill up all fields. 2. The app will return to step 3. |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC024 | | |
| --- | --- | --- | --- |
| Use Case Name: | View suggested buddies | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database, Maps API |
| --- | --- |
| Description. | View a list of suggested buddies |
| Preconditions: | 1. Invocation as an inclusion of use case UC022 Login to gym buddy account. 2. The user must have an internet connection. 3. The user must have accepted the app permission to access the user’s location. |
| Postconditions: | The app displays a list of suggested buddies. |
| Priority: | Medium.  This functionality will enable users to find buddies to match. It is a core feature of the gymm buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals. |
| Frequency of Use: | Medium.  This function will only be invoked if the user chooses to utilise the gym buddy functionality. |
| Flow of Events: | 1. The user is in the gym buddy interface and clicks find a buddy. 2. The app uses the App Database to fetch gym buddy information from other users. 3. The app uses the Maps API to fetch location details from other users. 4. The app uses a matching algorithm to suggest buddies. 5. The app will display suggested buddies one by one, with an option to swipe left and swipe right to view more buddies. 6. If the user swipes left or right, a new suggested buddy will be shown. |
| Alternative Flows: | AF-S5: There are no more suggested buddies for the user to match.   1. The app will display a message “No users to match at the moment”. 2. The app will return to the gym buddy interface. |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC025 | | |
| --- | --- | --- | --- |
| Use Case Name: | View specific buddy details | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor)  App Database |
| --- | --- |
| Description. | User clicks into a specific buddy and is shown more details of the selected buddy. |
| Preconditions: | 1. Invocation as an inclusion of use case UC022 Login to gym buddy account. 2. This use case extends UC024 View suggested buddies |
| Postconditions: | The app shall display detailed information about the buddy selected, such as their style of training and traits. |
| Priority: | Low.  This functionality will allow users to view more specific details about a buddy should they choose to do so. It is not required for users to match with buddies. |
| Frequency of Use: | Low.  This function will only be invoked if the user chooses to utilise the gym buddy functionality, and chooses to view the details of a specific buddy while in use case UC024 View suggested buddies. |
| Flow of Events: | 1. The user clicks on the suggested buddy’s profile to view more details. 2. The app fetches specific details of the buddy from the App Database. 3. The app then displays specific details of the buddy from the App Database. 4. If the user taps on the view more photos button, the app invokes the extended use case UC027 View suggested buddies photo. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC026 | | |
| --- | --- | --- | --- |
| Use Case Name: | Request to pair up with buddy | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | The user selects to pair up with the suggested buddy. |
| Preconditions: | 1. This use case extends UC025 View Specific Buddy Details 2. Invocation as an inclusion of use case UC022 Login to gym buddy account. 3. The user must have an internet connection. 4. The user must have accepted the app permission to access the user’s location. |
| Postconditions: | The app will update pairing information in the app database.  The buddy will appear in the user’s buddy list, the next time the user navigates there. |
| Priority: | Medium.  This functionality will allow users to request to pair up with a buddy. It is a core feature of the gym buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals. |
| Frequency of Use: | Low.  This function will only be invoked if the user chooses to utilise the gym buddy functionality, and chooses to view the details of a specific buddy while in use case UC024 View suggested buddies. |
| Flow of Events: | 1. The user clicks on the “green tick” to pair up with the suggested buddy. 2. If the other buddy has already chosen to pair up, the app will display a message “You have successfully matched with <The other user’s name>”. 3. The app will return to use case UC024 View suggested buddies. |
| Alternative Flows: | AF-S2 If the other buddy has yet to click the green tick or the other buddy has selected the “red cross”.   1. The app will display a message “pending match” and the suggested buddy will be removed from the list. 2. The app will display the next suggested buddy from the list. 3. The app will return to step 3. |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC027 | | |
| --- | --- | --- | --- |
| Use Case Name: | View suggested buddies photo | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | The user chooses to view more photos of the suggested gym buddy |
| Preconditions: | 1. This use case extends UC025 View Specific Buddy Details 2. Invocation as an inclusion of use case UC022 Login to gym buddy account. 3. The user must have an internet connection. 4. The user must have accepted the app permission to access the user’s location. |
| Postconditions: | The app shall display a gallery of photos arranged in ascending order of date uploaded. |
| Priority: | Very low.  This functionality allows users to view the appearance of their buddy. However, as users should be matching based on their fitness goals, viewing photos would be a minor, if not negligible, factor in determining a match and is hence not crucial to this functional requirement. |
| Frequency of Use: | Very low.  This function will only be invoked if the user chooses to utilise the gym buddy functionality, chooses to view the details of a specific buddy, and chooses to view their photo. |
| Flow of Events: | 1. The user taps on to view more photos 2. The user can scroll through the photos uploaded by the other gym buddy. |
| Alternative Flows: | AF-S1: The suggested buddy only has 1 photo.   1. The app will display a message “no more photos to view”. 2. The app returns to UC025 View specific buddy details |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC028 | | |
| --- | --- | --- | --- |
| Use Case Name: | Chat with buddy | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database, Buddy |
| --- | --- |
| Description. | The user chooses to start a chat with a buddy after a match is found. |
| Preconditions: | 1. Invocation as an inclusion of use case UC026 Request to pair up with buddy. 2. Invocation as an inclusion of use case UC022 Login to gym buddy account. 3. There must be a match between the user and the suggested buddy. 4. There must be an existing internet connection. 5. This use case extends UC030 View all chats. |
| Postconditions: | The app will consistently update chat information into the app database.  The app must display chat messages sent by the user onto the buddy’s chat display.  The app must display chat messages sent by the buddy onto the user’s chat display. |
| Priority: | Medium.  This functionality will allow users to chat with a matched buddy. It is a core feature of the gym buddy functional requirement without which it will not work to fulfil its purpose. Having a workout partner is important to motivate users to continue working out and keep each other accountable for their fitness goals. |
| Frequency of Use: | Very Low.  This function will only be invoked if the user chooses to utilise the gym buddy functionality, chooses to request to pair up with buddy and the buddy has done the same, and one of the users wants to send a message. |
| Flow of Events: | 1. The user selects a buddy to start a chat. 2. The user sends a message. 3. The message will appear on the buddy’s chat. |
| Alternative Flows: | AF-S2 The user receives a message.   1. The message will appear on the user’s chat with that buddy. |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account, UC026 Request to pair up with buddy |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout.  For messages to be received and sent concurrently, both the user and the buddy must have a stable internet connection. |
| Notes and Issues: | The app shall display the message sent on the buddy’s chat within 1 one second.  The app shall display the message sent by the buddy on the user’s chat within one second. |

| Use Case ID: | UC029 | | |
| --- | --- | --- | --- |
| Use Case Name: | Remove buddy | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | Remove gym buddy from buddy list |
| Preconditions: | 1. Invocation as an inclusion of use case UC022 Login to gym buddy account. 2. Invocation as an inclusion of use case UC026 request to pair up with buddy. 3. There must be an existing match between the user and the suggested buddy. |
| Postconditions: | 1. The app will remove the match from the user’s buddy list and the app database will be updated. 2. The app will also remove the match from the buddy’s user list and the app database will be updated accordingly. 3. The app shall display a message indicating successful removal of buddy. |
| Priority: | Medium.  This function is necessary to enhance overall user experience, allowing users to remove buddies that they do not desire, or feel uncomfortable chatting with. |
| Frequency of Use: | Very Low.  This function will only be invoked if the user chooses to utilise the gym buddy functionality, chooses to request to pair up with buddy and the buddy has done the same, and one of the users wants to remove gym buddy |
| Flow of Events: | 1. The user selects a chat with a buddy. 2. The app displays the chat with the selected buddy. 3. The user selects the option “Remove buddy” 4. The app will display a confirmation message “Are you sure?” 5. If the user selects Yes, the app displays the message “Buddy has been removed”. |
| Alternative Flows: | AF-S4: The user selects No   1. The app will return to step 2. |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account, UC026 request to pair up with buddy |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |

| Use Case ID: | UC030 | | |
| --- | --- | --- | --- |
| Use Case Name: | View all chats | | |
| Created By: | Joshua Khoo | Last Updated By: | Joshua Khoo |
| Date Created: | 31/1/2022 | Date Last Updated: | 6/2/2022 |

| Actor: | User (Initiating Actor), App Database |
| --- | --- |
| Description. | The user can view all existing chats with the user’s matches. |
| Preconditions: | 1. Invocation as an inclusion of use case UC022 Login to gym buddy account. 2. The user must have an internet connection. |
| Postconditions: | The app shall display a list of all chats with all existing matches the user has. |
| Priority: | Medium.  This function represents the main platform for users to interact with their buddies. |
| Frequency of Use: | High.  This function will be invoked whenever the user wants to chat with other buddies in UC028 or view their buddy list. |
| Flow of Events: | 1. The user selects “Buddy List” 2. The app will display a list of chats with all buddies that have been matched. 3. If the user selects a specific buddy to chat with, the app invokes the extended use case UC028 Chat with buddy |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | UC022 Login to gym buddy account |
| Special Requirements: | - |
| Assumptions: | The user’s location is kept on throughout. |
| Notes and Issues: | - |