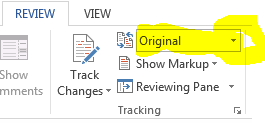
***Notes to the Team***

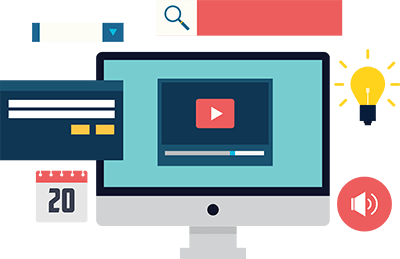
This is a template for the **Elaboration Document** for your project with explanatory notes included.

**When using this template, the following steps are recommended:**

1. Replace all text enclosed in angle brackets (e.g., <Team Name>) with the correct values. These angle brackets appear either in the body of the document or in headers and footers.
2. To update the Table of Contents, right-click on it and select “Update field” and choose the option - “Update entire table”.
3. Turn off Tracking. Go to the Review tab; make sure Display for Review is set to “Original”.



1. Before submission of this document, delete this instruction section “Notes to the Team”.



Stats and Stacks

Elaboration Document

BucketList

Team Members:

|  |
| --- |
| 217143512, Bernard McDonald  217198392, Mu’Minah Daniels  213348292, Tiffany de Vos |

Document Due Date: 20 June 2019

**TABLE OF CONTENTS**

[1 FUNCTIONAL REQUIREMENTS 4](#_Toc8676236)

[1.1 Analysis Use Case Model 4](#_Toc8676237)

[1.2 Use Case Glossary and Responsibilities 5](#_Toc8676238)

[2 UI Prototypes 7](#_Toc8676239)

[2.1 Team UI Guidelines 7](#_Toc8676240)

[2.2 UI Designs & Updated Analysis Use Case Narratives 7](#_Toc8676241)

[2.2.1 Designed by <Insert team member name> 7](#_Toc8676242)

[2.2.2 Designed by <Insert team member name> 7](#_Toc8676243)

[2.2.3 Designed by <Insert team member name> 8](#_Toc8676244)

[2.2.4 Designed by <Insert team member name> 8](#_Toc8676245)

[3 DATA REQUIREMENTS 9](#_Toc8676246)

[3.1 Domain Class Diagram 9](#_Toc8676247)

[3.2 Implementation Ready Class Diagram 10](#_Toc8676248)

# FUNCTIONAL REQUIREMENTS

## Analysis Use Case Model

[Replace this text with the Analysis Use Case diagram – this must be drawn in Visio.]

## Use Case Glossary and Responsibilities

|  |  |
| --- | --- |
| **Team Member Responsible: Mu’Minah** | |
| **Use Case Id** | **Use Case Name** |
| B0100 | Login/Logout |
| B0200 | Sign up |
| B0300 | Maintain account |
| B0400 | View profile |
| A0100 | Create challenge |
| A0200 | Approve/Reject suggested challenge |
| A0300 | Maintain challenge |
| **Queries/Reports** | |
| A0400 | View established challenges |
| A0500 | View suggested challenges |

|  |  |
| --- | --- |
| **Team Member Responsible: Bernard** | |
| **Use Case Id** | **Use Case Name** |
| A0600 | Add challenge to own list |
| A0700 | Suggest challenge |
| A0800 | Mark challenge as completed |
| A0900 | Upload photo |
| A1000 | View ratings/reviews |
| A1100 | Rate challenge |
| A1200 | Send challenge invite |
| **Queries/Reports** | |
| A1300 | View own challenges |
| A1400 | View new challenges |

|  |  |
| --- | --- |
| **Team Member Responsible: Tiffany** | |
| **Use Case Id** | **Use Case Name** |
| C0100 | Send friend request |
| C0200 | Accept/Reject friend request |
| C0300 | Remove friend |
| C0400 | View friend’s profile |
| C0500 | Accept/Reject challenge invite |
| C0600 | Verify friend’s challenge completion |
| C0700 | Calculate points |
| C0800 | Check Achievements |
| **Queries/Reports** | |
| C0900 | View friends’ activity |
| C1000 | View friend list |

# UI Prototypes

## Team UI Guidelines

**Navigation & Control**

Moving between pages should be easy and intuitive. The user should always know where to click to move to the next desired page and each page should never be more than 3 clicks away. This principle is enforced by having

* The header with the logo, which acts as a link to the home page, always visible (INSERT SCREENSHOT)
* Navigation options in footer of each page
* Consistently styled and clearly visible buttons that indicate that they link to another page

Controls should be consistently styled and used effectively. Meaning, that the suitable control should be used for the specific user input (dropdowns for lists of selectable items, checkboxes for true or false input etc.). These controls will be prominent in the ‘Filter’ section (seen in the Discover Challenges, My Friends and My Challenges pages) as multiple controls can be used to filter each query.

**Support for User Efficiency**

The user can only access the website once logged in. Once the user is logged in they will always see the home page first, making the starting point of each visit familiar to the user. Each page will only show the necessary information in an organized and simplistic manner as to not confuse or overwhelm the user. Each page serves towards a single function, such as viewing the user’s challenges, managing their friends etc. Making use of the most suitable controls (as mentioned above) provides user efficiency as these controls are intuitive.

**Recognition rather than recall**

Reinforcement of user actions, such as friend added. System lets user knows friend has been added.

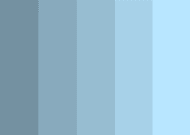
Recognition of user actions, such as buttons change color when user hovers over them etc.

FINISH UP

**Aesthetic design**

Each page should be simplistic in design, while providing the user all the necessary information and functionality. This principle can be enforced by efficiently choosing the key visual elements of your website: colors, fonts and graphics.

* We have decided on a simple pastel pallet as it is ‘easy on the eyes’ and compliments the idea of simplicity.



* FONT?
* Graphics should only be used when helping the user accomplish a task or when it enforces familiarity and simplicity. Meaning, using globally recognized graphics to convey specific information such as a check mark for a completed task, or a question mark for a pending task etc.

Each page should share a consistency that makes the user more comfortable with the website as they progressively use it. This will be enforced by using consistent styling and the same general template for each page:

(INSERT SCREENSHOT)

**Error prevention/recovery**

Errors are prevented by giving the user only the necessary amount of control. Meaning, only controls that are accessible to the user at that current point in time are displayed and suitable controls are used for user input. Errors can also be prevented by validating user input before it is sent to the server i.e. check that an entered email address is in the correct format before allowing the user to submit it.

Errors are recovered by making use of pop-ups once an error has occurred and displaying enough information about the error (without getting too technical) to make the user understand what went wrong and what steps that can take to rectify the error.

## UI Designs & Updated Analysis Use Case Narratives

### Designed by Mu’Minah

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0100 | Login/Logout | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin, User | |  |
| **Description** | The admin/user can log into their account using their credentials or log out of their account. | |
| **Pre-Conditions** | The admin/user must have created an account. | |
| **Triggers** | Admin/User selects the “Login”/” Logout” option or user has just signed up. | |
| **Post-Conditions** | The admin/user is logged into or logged out of their account. | |
| **Basic Flow of Events** | 1. Admin/User selects “Login” in option on the sign-up page 2. Admin/User enters account credentials 3. Admin/User is logged in | |
| **Alternate Flow** | * 1. Admin/User selects “log out” option in the header   2. Admin/User is logged out   OR   * 1. Called from B0200 (Sign up)   2. Return to step 3. | |
| **Initial UI design** | A screenshot of a cell phone  Description automatically generated | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0200 | Sign up | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can create an account of their own. | |
| **Pre-Conditions** |  | |
| **Triggers** | The user clicks on a link to BucketList signup | |
| **Post-Conditions** | A new user account is created. | |
| **Basic Flow of Events** | 1. System displays new account form 2. User fills in form containing all information regarding account 3. User clicks “Sign Up” 4. Account is created 5. System calls B0100 (Login) | |
| **Alternate Flow of events** |  | |
| **Initial UI design** | Sign-up page | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0300 | Maintain account | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can make changes to their account. | |
| **Pre-Conditions** | The user must be logged in. | |
| **Triggers** | The user selects “Account” located in the header. | |
| **Post-Conditions** | The changes to the account are saved. | |
| **Basic Flow of Events** | 1. System displays “Maintain Account” form 2. User edits form containing all information regarding account 3. User selects “Save” option 4. Changes are saved | |
| **Alternate Flow of events** | * 1. User selects “Cancel” option   2. Changes are cancelled | |
| **Initial UI design** | Maintain account page | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0400 | View own profile | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can view his/her points and achievements | |
| **Pre-Conditions** | The user must be logged in. | |
| **Triggers** | The user selects his/name username located in the header. | |
| **Post-Conditions** | The user’s points and achievements are displayed | |
| **Basic Flow of Events** | 1. User selects his/her username 2. The system displays user’s achievements and points | |
| **Alternate Flow of events** |  | |
| **Initial UI design** | Account page | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0100 | Create challenge | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin | |  |
| **Description** | The admin creates a challenge that is made available to users to add to their bucket list. | |
| **Pre-Conditions** | The admin must be logged in. | |
| **Triggers** | The admin selects the “Create Challenge” option on either the “Maintain Challenges” tab or the “Review Challenges” tab | |
| **Post-Conditions** | A challenge is created and saved. | |
| **Basic Flow of Events** | 1. System displays new challenge form 2. Admin fills in form containing all information regarding challenge 3. Admin selects “Save” option 4. Challenge is saved and added under “Maintain Challenges” tab | |
| **Alternate Flow of events** | * 1. Admin selects “Cancel” option   2. Challenge is cancelled | |
| **Initial UI design** | create challenge page | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0200 | Approve/Reject suggested challenge | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin | |  |
| **Description** | The admin approves or rejects a challenge that has been suggested by a user. The admin can edit the challenge before approving it. | |
| **Pre-Conditions** | The admin must be logged in and a challenge must be suggested by a user. | |
| **Triggers** | A challenge is suggested by a user (A0700) and the admin selects the “Review Challenges” tab. | |
| **Post-Conditions** | The challenge is either approved, saved and made available to users for use or rejected and discarded. | |
| **Basic Flow of Events** | 1. Admin views the suggested challenges (A0400) 2. Admin selects “Approve” button for a challenge 3. Challenge is saved and added under “Maintain Challenges”. | |
| **Alternate Flow** | * 1. Admin selects “Edit” button for a challenge   2. A0300 is called   OR   * 1. Admin selects “Reject” button for a challenge   2. Challenge is discarded | |
| **Initial UI design** | A screenshot of a cell phone  Description automatically generated | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0300 | Maintain challenge | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin | |  |
| **Description** | The Admin may make changes to an existing or suggested challenge. | |
| **Pre-Conditions** | The admin must be logged in and the challenge must already exist (created, approved or suggested). | |
| **Triggers** | The admin selects the “Edit” option. | |
| **Post-Conditions** | The changes to the challenge are saved. | |
| **Basic Flow of Events** | 1. Admin views the established challenges (A0400) 2. Admin selects “Edit” option for the challenge he/she wishes to edit 3. System displays edit challenge form 4. Admin edits form containing all information regarding the challenge 5. Admin selects “Save” option 6. Changes are saved 7. A0400 is called | |
| **Alternate Flow of events** | * 1. Called from A0200 or A0400 or A0500   2. Return to step 3   3. Admin selects “Cancel” option   4. Changes are cancelled  1. A0500 is called | |
| **Initial UI design** | Edit Challenge form | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0400 | View established challenges | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin | |  |
| **Description** | Admin can view and filter through all established challenges. Established challenges are all challenges either created by the Admin or approved by the Admin. | |
| **Pre-Conditions** | Admin must be logged in and at least one established challenge must exist | |
| **Triggers** | Admin clicks on the “Maintain Challenges” option in the header. | |
| **Post-Conditions** | All established challenges are displayed | |
| **Basic Flow of Events** | 1. User selects “Maintain Challenges” option 2. System displays all established challenges. 3. User may Filter or Edit the established challenges | |
| **Alternate Flow of events** | 1. User selects “Filter” button 2. System displays filter form 3. User fills in form containing all information regarding filtering the challenges 4. System applies filter. 5. Return to step 2.   OR   1. User selects “Edit” 2. A0300 (Maintain Challenge) is called | |
| **Initial UI design** | A screenshot of a cell phone  Description automatically generated | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0500 | View suggested challenges | |
| **Primary Business Actors** | | **Other participating Actors** |
| Admin | |  |
| **Description** | Admin can view and filter through all suggested challenges. Suggested challenges are all challenges which users have suggested and have not been approved by the Admin | |
| **Pre-Conditions** | Admin must be logged in and at least one suggested challenge must exist | |
| **Triggers** | Admin clicks on the “Review Challenges” option in the header. | |
| **Post-Conditions** | All suggested challenges are displayed | |
| **Basic Flow of Events** | 1. Admin selects “Review Challenges” option 2. System displays all established challenges. 3. Admin may filter or edit suggested challenges | |
| **Alternate Flow of events** | 1. Admin clicks “Filter” button 2. System displays filter form 3. Admin fills in form containing all information regarding filtering the challenges 4. System applies filter 5. Return to step 2 6. Admin clicks “Edit” 7. A0300 (Maintain Challenge) called | |
| **Initial UI design** | A screenshot of a cell phone  Description automatically generated | |

### Designed by Bernard

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0600 | Add challenge to own list | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user adds a challenge to their bucket list. | |
| **Pre-Conditions** | User must be logged in and the challenge must exist. | |
| **Triggers** | The user selects the “Add to BucketList” option when viewing new challenges. | |
| **Post-Conditions** | The challenge is added to their bucket list. | |
| **Basic Flow of Events** | 1. User views the new challenges (A1400) 2. User selects “Add to BucketList” option for the challenge user wishes to add to own list. 3. Challenge is added to their bucket list | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0700 | Suggest challenge | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | Admin |
| **Description** | The user creates a challenge suggestion which is either approved or declined by an admin. | |
| **Pre-Conditions** | The user must be logged in. | |
| **Triggers** | The user selects the create challenge suggestion option on the “New Challenges” tab. | |
| **Post-Conditions** | The suggested challenge is saved and made available to the admin for review (A0200). | |
| **Basic Flow of Events** | 1. User views the new challenges (A1400) 2. User selects the “Suggest Challenge” option 3. System displays new challenge form 4. User fills in form containing all information regarding challenge 5. User selects “Save” option 6. Challenge is saved | |
| **Alternate Flow of events** | 5.1) User selects “Cancel” option  5.2) Operation is cancelled | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0800 | Mark challenge as completed | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user marks a challenge as complete and it is made available for verification by friends. | |
| **Pre-Conditions** | The user must be logged in, the challenge must exist, and the challenge must have been added to their bucket list. | |
| **Triggers** | The user completes the challenge and selects the “Mark as Complete” option on the “My Challenges” tab | |
| **Post-Conditions** | The challenge is marked as complete and is made available for verification by friends. | |
| **Basic Flow of Events** | 1. User views his/her own challenges (A1400) 2. User selects “Mark as Complete” option for the desired challenge 3. Completed challenge is made available for verification by friends (C0600) 4. The option is given to rate/review challenge (A1100) | |
| **Alternate Flow** | * 1. If challenge requires photo, system calls A0900. Return to step 3 | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0900 | Upload photo | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user uploads a photo of them completing a challenge. | |
| **Pre-Conditions** | The user must be logged in, the challenge must exist, the challenge must have been added to their bucket list, and the user must have a photo of them completing the challenge. | |
| **Triggers** | The user completes the challenge (A0800) and selects the upload photo option (some challenges require a photo, others do not). | |
| **Post-Conditions** | The photo is uploaded, and the completed challenge is made available for verification by friends (C0500). | |
| **Basic Flow of Events** | 1. User marks a challenge as complete (A0800) 2. System displays a dialog box to select the photo to upload 3. User selects the photo 4. Photo is uploaded | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A1000 | View ratings/reviews | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can view the reviews for any existing challenges | |
| **Pre-Conditions** | The user must be logged in, the challenge must exist | |
| **Triggers** | The user selects the “See more” option next to a challenge while viewing own challenges, new challenges or friends’ challenges. | |
| **Post-Conditions** | The system displays the reviews and rating for a selected challenge | |
| **Basic Flow of Events** | 1. User views the new challenges (A1400) 2. User selects “See more” next to the rating for the challenge the user wishes to see 3. System displays all ratings and reviews made by users for that challenge | |
| **Alternate Flow** | * 1. User views his/her own challenges (A1300). Return to step 2   2. User views friends’ challenges (C0900). Return to step 2 | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A1100 | Rate challenge | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can rate a challenge that they have completed. | |
| **Pre-Conditions** | The user must be logged in, the challenge must exist, and the user must have marked the challenge as complete (A0600). | |
| **Triggers** | The user selects the “Rate” option either when viewing a completed challenge in his/her challenges or upon completing a challenge. | |
| **Post-Conditions** | The challenge rating and review is saved. | |
| **Basic Flow of Events** | 1. User views his/her own challenges (A1300) 2. User selects “Rate” option for the completed challenge user wishes to rate 3. System displays rate challenge form 4. User fills in form containing all information regarding rating and reviewing challenge 5. Rating is saved | |
| **Alternate Flow** | * 1. User marks a challenge as complete (A0800)   2. User selects “Rate” option for challenge. Return to step 3. | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A1200 | Send challenge invite to friend | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can invite a friend to a group challenge. | |
| **Pre-Conditions** | The user must be logged in, the challenge must exist and be allowed to be a group challenge and the other user must be a friend. | |
| **Triggers** | The user selects the “Invite Friend” option when viewing own challenges | |
| **Post-Conditions** | The friend is added to the group challenge. | |
| **Basic Flow of Events** | 1. User views own challenges (A1300) 2. User selects “Invite Friend” option for the challenge they wish to invite a friend to 3. User selects friend from friends list 4. Challenge invite is sent to friend | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A1300 | View own challenges | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can view and filter through the challenges that they have added to their bucket list. | |
| **Pre-Conditions** | The user must be logged in. | |
| **Triggers** | The user selects the “My Challenges” option. | |
| **Post-Conditions** | The challenges that were added to the users bucket list are displayed. | |
| **Basic Flow of Events** | 1. User selects “My Challenges” option 2. System displays challenges added to the users bucket list | |
| **Alternate Flow of events** | * 1. User selects “Filter” option   1.2) System displays filter form  1.3) User fills in form containing all information regarding filtering the displayed bucket list  1.4) User selects “apply filter” option   * 1. System applies filter. Return to step 2. | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A1400 | View new challenges | |
| **Primary Business Actors** | | **Other participating Actors** |
|  | |  |
| **Description** | The user can view and filter through the challenges that they have not added to their bucket list. | |
| **Pre-Conditions** | The user must be logged in. | |
| **Triggers** | The user selects the “Discover Challenges” option. | |
| **Post-Conditions** | The challenges (not yet added to the users bucket list) are displayed. | |
| **Basic Flow of Events** | 1. User selects “My Challenges” option 2. System displays challenges not yet added to the users bucket list | |
| **Alternate Flow of events** | * 1. User selects “Filter” option   1.2) System displays filter form  1.3) User fills in form containing all information regarding filtering the displayed bucket list  1.4) User selects “apply filter” option  1.5) System applies filter. Return to step 2. | |
| **Initial UI design** |  | |

### Designed by Tiffany

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0100 | Send friend request | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can attempt to add another user to their friends list by sending a friend request. | |
| **Pre-Conditions** | The user must be logged in, the other user must exist, and the user must know the other user’s username or name and surname | |
| **Triggers** | Called from use case C1000 | |
| **Post-Conditions** | The other user is added to the user’s friends list. | |
| **Basic Flow of Events** | 1. User searches for the user by username or name and surname using the search bar 2. System displays search results 3. User locates user they wish to befriend in results 4. User selects “Send Friend Request” option for that user 5. Friend request is sent | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0200 | Accept/Reject friend request | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user accepts or rejects a friend request from another user. | |
| **Pre-Conditions** | The user must be logged in and must have received a friend request from another user. | |
| **Triggers** | The user receives a friend request in their “Notifications”. | |
| **Post-Conditions** | If the friend request is accepted, the other user is added to the user’s friend list. If rejected, the friend request is discarded. | |
| **Basic Flow of Events** | 1. User selects “Notifications” option 2. System displays all invites (friend requests and challenge invites) 3. User locates the invite they wish to accept/decline 4. If user wants to accept the friend request, user selects “Accept” option for that friend request and other user is added to their friends list | |
| **Alternate Flow** | * 1. If user wants to reject the friend request, user selects “Decline” option for that friend request and friend request is discarded | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0300 | Remove friend | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can maintain their friends list by viewing and removing friends. | |
| **Pre-Conditions** | The user must be logged in and have another user added to their friends list. | |
| **Triggers** | Called from use case C1000 or use case C0800 | |
| **Post-Conditions** | Changes to the user’s friends list is saved. | |
| **Basic Flow of Events** | 1. User locates the friend user wants to remove in the list of friends. 2. User selects “Remove Friend” option for that friend 3. That user is removed from the friends list | |
| **Alternate Flow** | * 1. User locates “Remove Friend” option on the friend’s profile and selects it.   2. The user is removed from the friends list | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0400 | Accept/Reject challenge invite from friend | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can accept an invite to a group challenge. | |
| **Pre-Conditions** | The user must be logged in, and the user must’ve received a challenge invite from a friend (A1200). | |
| **Triggers** | The user receives an invite to a group challenge from a friend in “Notifications” | |
| **Post-Conditions** | If the challenge invite is accepted, the challenge is added to the user’s bucket list. If rejected, the challenge invite is discarded. | |
| **Basic Flow of Events** | 1. User selects “Notifications” option 2. System displays all notifications (friend requests, challenge invites) 3. User locates the invite they wish to accept/decline and selects it 4. System displays the information for the challenge. 5. If user wants to accept a challenge request, user selects “Accept” option for that request and challenge is added to user’s bucket list | |
| **Alternate Flow** | * 1. If user wants to reject a challenge request, user selects “Decline” option for that request and challenge request is discarded | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0500 | Verify friend’s challenge completion | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | The user can verify that a friend has completed a challenge. | |
| **Pre-Conditions** | The user must be logged in, the other user must be added as a friend, and the friend must’ve marked the challenge as completed (A0800). | |
| **Triggers** | Called from C0900 | |
| **Post-Conditions** | The challenge completion is verified. | |
| **Basic Flow of Events** | 1. User located the completed challenge they would like to verify 2. User selects “Verify Challenge” option next to that completed challenge 3. System calls C0600 | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0600 | Calculate points | |
| **Primary Business Actors** | | **Other participating Actors** |
| System | |  |
| **Description** | System allocates points to user for a verified completed challenge. | |
| **Pre-Conditions** | The user must’ve marked a challenge as complete and a friend must’ve selected the “Verify Challenge” option on the challenge (C0500). | |
| **Triggers** | Called from C0500 | |
| **Post-Conditions** | The system allocates points to the user. | |
| **Basic Flow of Events** | 1. System calculates the amount of points that should be allocated to user 2. System allocates points to user 3. Changes are saved 4. Use case C0700 is called. | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0700 | Check achievements | |
| **Primary Business Actors** | | **Other participating Actors** |
| System | |  |
| **Description** | System allocates earned achievements to user after a verified completed challenge. | |
| **Pre-Conditions** | The user must’ve marked a challenge as complete, a friend must’ve selected the “Verify Challenge” option on the challenge (C0500) and the system must have calculated the points (C0600). | |
| **Triggers** | Called from C0600 | |
| **Post-Conditions** | The system allocates any earned achievements to the user. | |
| **Basic Flow of Events** | 1. System checks whether any achievements should be allocated to user 2. System allocates earned achievements to user 3. Changes are saved | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

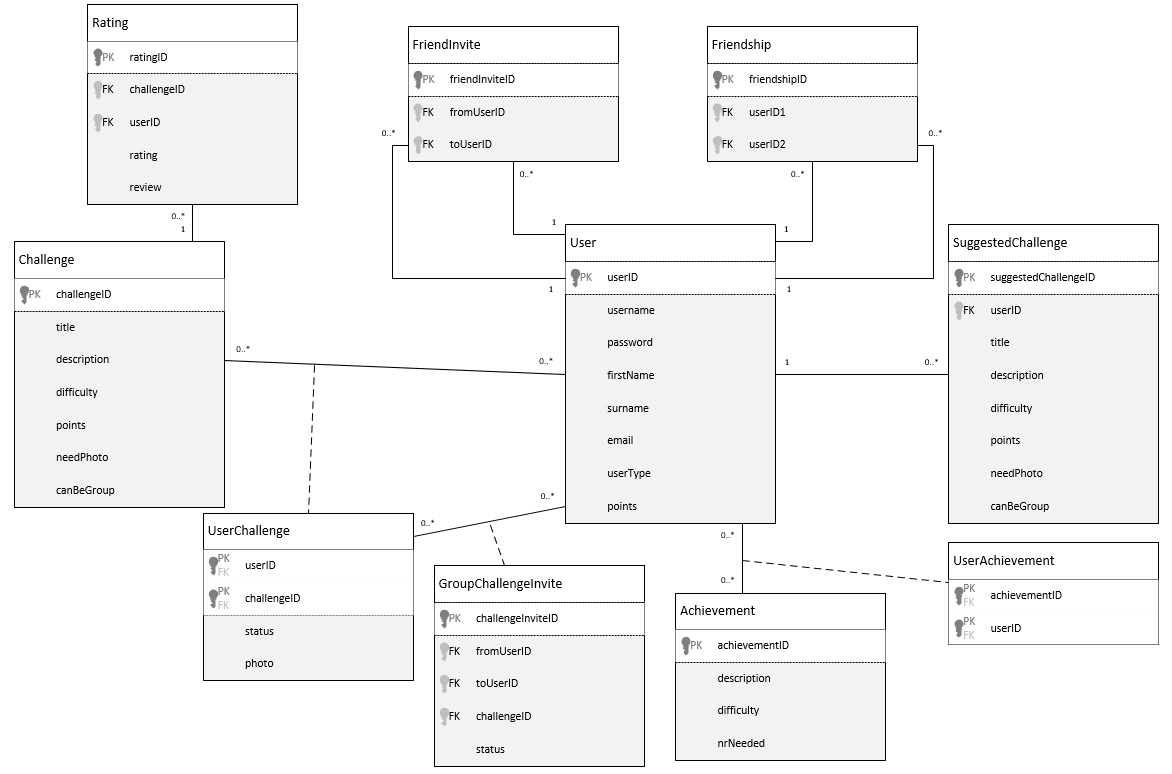
|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0800 | View friend’s profile | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | User views the points and achievements for a friend. | |
| **Pre-Conditions** | User must be logged in and be friends with another user. | |
| **Triggers** | Called from C0900 or C1000 | |
| **Post-Conditions** | System displays the points and achievements for a friend. | |
| **Basic Flow of Events** | 1. User locates name of friend whose profile they wish to view 2. User selects name of that friend 3. System display the points, achievements and a remove option for that friend. | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0900 | View friend’s list | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | User views their list of friends | |
| **Pre-Conditions** | User must be logged in and be friends with at least one other user. | |
| **Triggers** | Called from C0900 | |
| **Post-Conditions** | System displays the user’s list of friends | |
| **Basic Flow of Events** | 1. User selects “Add/Remove Friends” option 2. System displays all friends with the options to add friends or remove friends. | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C1000 | View friends’ challenges | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | User views friends’ challenges and their statuses. | |
| **Pre-Conditions** | User must be logged in and be friends with at least one another user. | |
| **Triggers** | User selects the “Friends’ Challenges” option. | |
| **Post-Conditions** | System displays the user’s friends’ challenges and their statuses. | |
| **Basic Flow of Events** | 1. User selects the “Friends’ Challenges” option 2. System display all the challenges of friends and their statuses 3. User can optionally filter challenges by friend name, challenge name, challenge status, challenge difficulty, challenge rating and whether the challenge is a group challenge. | |
| **Alternate Flow of events** |  | |
| **Initial UI design** |  | |

# DATA REQUIREMENTS

## 3.1 Domain Class Diagram



## Implementation Ready Class Diagram

