ClubLife

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software Requirements Document

**Brandon Poole**

**Jonathan Merklin**

**Sasha Hedges**

**Zach Babka**

**9/22/2017**

## Change History

|  |  |  |
| --- | --- | --- |
| **Date** | **Name** | **Change** |
| 8/25/16 | ClubLife Team | Initial Version |
| 9/13/16 | Brandon Poole | Formatting |
| 9/13/16 | Zachary Babka | Introduction |
| 9/15/16 | ClubLife Team | Created and Inserted UML |
| 9/15/16 | Brandon Poole | Improved Introduction |
| 9/20/16 | Brandon Poole | Formatting/Revising |
| 9/20/16 | Jonathan Merklin | General Revising |
| 9/20/16 | Sasha Hedges | Formatting/Revising |
| 9/20/16 | Zachary Babka | Revisions |
| 9/21/16 | Brandon Poole | ToC Numbering/Formatting |
| 9/21/16 | Jonathan Merklin | Small Font Fix |

## Table of Contents

Introduction - 3

Key Definitions - 3

Project Description - 4

Summary - 4

Functional Requirements - 4

Nonfunctional Requirements - 4

Diagrams - 5

Class Diagram - 5

Use Case Diagram - 6

Activity Diagrams - 7

Find a Club - 7

View User Profile - 8

Edit Own Profile - 9

Find an Event - 10

Edit Member - 11

Make Club Post - 12

CRUD Event - 13

Edit Club Info - 14

Conclusion - 15

## 

## **Introduction**

Keeping up with the student organizations you are in can be a hassle. While there currently exists a mobile application for viewing and following clubs at the University of Alabama called Corq, the app is widely unknown and not a popular choice for organizing clubs. Our app, ClubLife, will be a dynamic source of communication between students and the organizations to which they belong. It will allow for easier management by both students seeking to be active in their clubs and officers trying to keep their club organized.

Major goals include developing the app to be cross-platform through the use of Xamarin, providing user-centered functionality that make it actually worth using, and having an easy-to-adopt setup for clubs.

### Key Definitions

Student Organization - A group of students along with a leading faculty member that meets regularly about a particular field/area/subject of interest.

Club - Effectively synonymous with student organization.

Officer - A member of a student organization with more responsibility but increased status within the organization; the position acquired often by election or appointment.

Club Leader - Similar to an officer, but with leadership over the student organization as a whole. (i.e. President/Chair)

Dynamic - Live updates including push notifications instead of requiring the user to refresh and manually check.

Cross-Platform - Reusable components across multiple platforms. (i.e. Android, iOS, and Windows).

(Motivation, Purpose, Scope, Goals. Also a list of key definitions that apply to project)

## 

## **Project Description**

Create a cross-platform application for both Android and iOS that will function as a dynamic, interactive repository of all clubs and student organizations at the University of Alabama. Users will be able to subscribe to desired organizations and receive notifications/updates about events related to said organizations. From an administrative standpoint, the officers of each club will be able to maintain the club’s information and alerts that are sent out, thus facilitating better communication and organization between clubs and members.

## **Summary**

### Functional Requirements

* Login Interface (MyBama integration?)
* User Profile Page (settings, list of clubs, personal information, edit option for specific user)
* Edit Profile Page (only available to specific user via edit button, turns all fields editable, save changes button)
* Club/Organization pages (Club Name, description, meeting locations/times, upcoming events, club documents, club tags, edit options for club officers/leaders)
* Edit Club/Organization Page (Only available to club officers/leaders, officers can edit most but not all fields, leaders can edit all fields)
* Create an Event (Time, place, description, affiliated organizations [opt in], event tags)
* Search events (Sort by name/upcoming, filter by affiliated club/event tags)

### Nonfunctional Requirements

* Pull from main database once a day to populate local database
* Testable interface that allows for efficient navigation through the app
* User permissions
* Automatic updating from SOURCE database
* User privacy
* Usability
* Releasing on iOS app store and android play store

## **Diagrams**

### Class Diagram

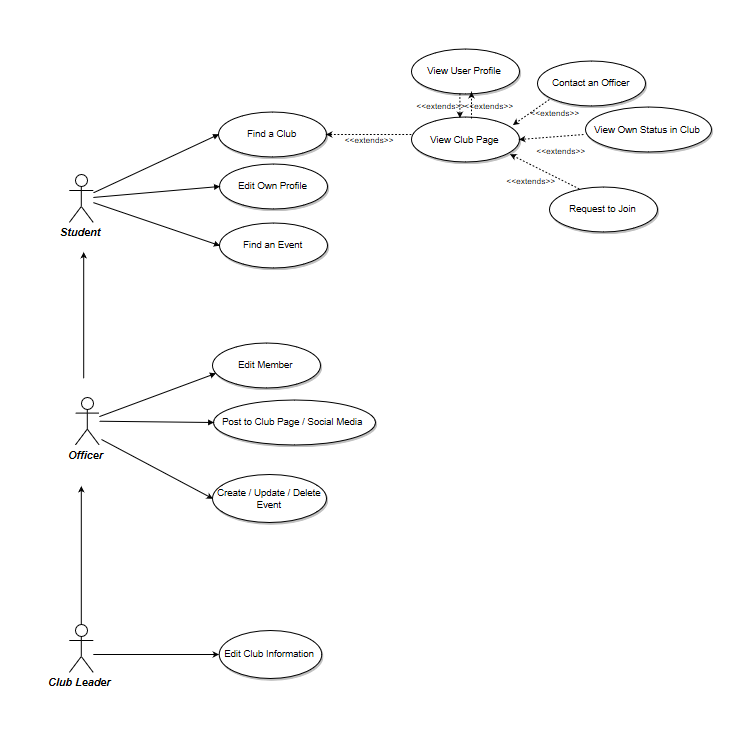
CS495ClassDiagramFinal.png

The generalized high-level class diagram (i.e. without the expected method signatures) is shown above. There are six primary classes at the heart of the ClubLife application. At the core of the application are students and clubs, which have a many-to-many relationship. Students are the users of the application, and they can be in zero or more clubs. Clubs are the main point of the application, and clubs can consist of zero or more students. Officers are students with special privileges for interacting with clubs; each club has zero or more officers. Club leaders are those instances of officers that lead the club, and therefore have the most features available to them.

Officers for clubs can create posts which appear on the club page - one club has many posts, hence the aggregation relationship they share. Events are another primary focus of the application - clubs can host events (which may be public or private, though that particular fact is not represented in this diagram).

### 

### Use Case Diagram



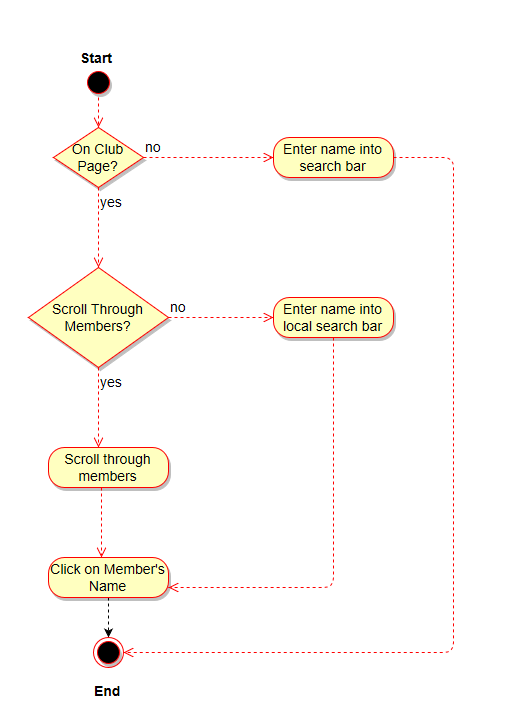
The club leader has the highest privileges, with all the abilities of a club leader, officer and a student. Similarly, an officer has the next highest privileges, with all the abilities of an officer and a student. A student can find a club and then has the options to view the club page, contact an officer, view their status in the club, request to join, and view user profiles. A student can also edit their own profile and find an event throughout all clubs. An officer has the ability to edit members’ of the club, post to the club page or to a social media page and create, update and delete events that the club created. A club leader has the ability to edit club information.

### Activity Diagrams:

### GiantFindClubActivityDiagramFINAL.png

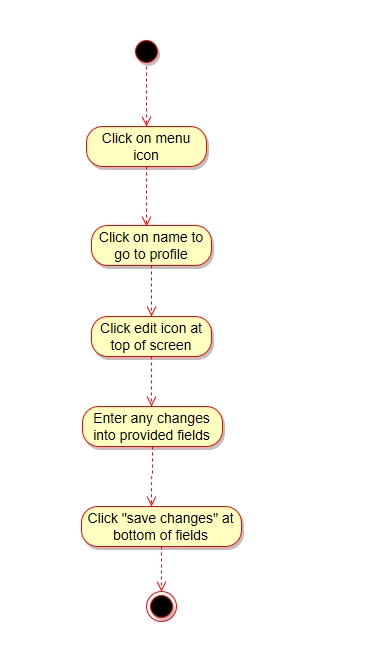
#### Find a Club

This activity diagram represents the information including and extending from the “Find a Club” use case. When a user searches for a club, ideally multiple clubs will be listed. From here, they may (optionally) view the page for a specific club. Once on the club page, there are a multitude of activities that can occur, depending on the user’s choice (and membership status within the club). Sometimes they may click on the membership profile of members visible from the club page. If they are not a member, they may request to join. If they are a member, they can view their status (or properties) within the club. They may also contact an officer of the club as that information is visible on the club page.



#### View User Profile

If the user is not on a club page, then they would click on the main search bar, enter the name of the desired user, and click on resulting member’s name. If the user is on the club page, they have the option of scrolling through a list of members or searching in the local search bar through the member’s to find the desired user, and then click on the resulting member’s name.



#### Edit Own Profile

To edit their profile, the user would click on the menu icon on the top of the screen and then click on their own name to navigate to their profile. The user can then click on the edit icon on the top of their profile and enter any changes into the provided fields. The user would then click save changes or cancel to stop editing their profile.

Untitled Diagram.png

#### Find an Event:

To find an event from a club page, a user would click on the “Events” tab and then either scroll through the listed events or enter the name or keyword into the search bar and click on the desired events for more information. If the user wanted to find a general event from any club, they could click the “Find Events” field on the main menu and are given the option to filter by club, subject and other options. They are also given the ability to search through the filtered list for an event with those selected options.



#### Edit Member

An officer can edit a member’s status by tapping on an edit icon next to the member’s name in a user list of those already in the club and those who have applied to the club. If the user is not a member, they can approve their application and add them to the club roster. If the user is already a member, the officer can update certain club-related member properties such as the status of dues paid and club titles. The officer can also choose to revoke a member’s status and remove them from the club.



#### Make Club Post

An officer can choose to either create a new post or edit an old post by selecting from a list of recent posts. Editing an old post would open up the post in edit mode where the officer could then choose to save any edits via the “Publish” button or cancel their edits. When creating a new post the application would pull up a text box where the officer could type out their new post. Once written, the officer can choose whether to publish the post to the club page in the application, to social media accounts, or both.

CRUD Event.png

#### CRUD Event

Officers can either create new public/private events or select an existing event and then either edit it or delete it.

Edit Club Info.png

#### Edit Club Info

Club Leaders can edit additional aspects of their club that a normal officer cannot. They can choose to update club properties or edit the officers.

## 

## 

## **Conclusion**

At its core, ClubLife is an organization management application that is similar to some other products that are already on the market, specifically targeted towards university clubs. The technical challenges stem from the goal of making a cross-platform product to appeal to a broad user base; from an architectural standpoint most of the work is that of a tried-and-true “create, read, update, display” application. From the functional and nonfunctional requirements one can ascertain that the application will be modeled around specific views for the users (students), as well as the clubs, with more functionality available to specific users within certain clubs.