

# Crimson Clubs

---

## Software Requirements Specifications

Max Livingston  
Maclane May  
Adam Gabriel  
Travis Redfield  
Taylor Meads

3/1/2018

Revision History			
Date	Version	Description	Author
2-19-2018	0.1	Initial Draft	Max Livingston
2-26-2018	0.1	Added More to ReqDoc	Max Livingston
2-28-2018	0.1	Added Diagrams	Max Livingston
3-1-2018	0.1	Added Activity Diagram Descriptions	Travis Redfield
3-1-2018	0.1	Updated DB Diagram Updated Class and DB diagram descriptions	Maclane May

## Table of Contents

1. Preface .....	3
2. Introduction .....	3
3. Glossary .....	3
4. User Requirements Definition .....	3
4.1 Functional Requirements .....	3
4.2 Non-Functional Requirements .....	4
5. Competition .....	4
6. Diagrams .....	4
6.1 Normal User Use Case Diagram .....	4
6.2 Admin User Use Case Diagram .....	5
6.3 Add Stats Activity Diagram .....	6
6.4 Add User Activity Diagram .....	6
6.5 Create Event Activity Diagram .....	7
6.6 Create Profile Activity Diagram .....	8
6.7 Leave Group Activity Diagram .....	8
6.8 Look at Info Activity Diagram .....	9
6.9 Modify Info Activity Diagram .....	9
6.10 Personal Calendar Activity Diagram .....	10
6.11 Remove User Activity Diagram .....	10
6.12 Search Clubs Activity Diagram .....	11
6.13 View Event Stats Activity Diagram .....	12
6.14 Database Diagram .....	12
6.15 Class Diagram .....	13

## 1. Preface

---

This is version 0.1 of the requirements documentation for the Android-based Crimson Clubs application. This document is intended for developers and user to guide them through the process of development during the course of the project.

## 2. Introduction

---

Crimson Clubs is an Android App/framework that will allow clubs and intramurals at universities keep track of their statistics and schedules. In addition to this functionality, we intend to include tools that will allow admins and referees to update and track past / previous games.

## 3. Glossary

---

Android: a software stack for mobile devices that includes an operating system, middleware, and key applications.

Application framework: framework that enables reuse and replacement of components

Framework: an abstraction in which software providing generic functionality can be selectively changed by user code, providing application specific software.

Admin User: a user that has extra permissions with that club or team. They function as a normal user except with the additional abilities when dealing with the specific club they are admin for.

Normal User: a user that has only standard permissions within the club they are viewing. They may have admin privileges in a different club.

## 4. User Requirements Definition

---

### 4.1 Functional Requirements

#### 4.1.1 Admin Users

1. An admin user shall be able to create events.
2. An admin user shall be able to add other people or players / approve people.
3. An admin user shall be able to add custom push notifications.
4. An admin user shall be able to remove people.
5. An admin user shall be able to modify descriptions, info's, and tags.

#### 4.1.2 Normal Users

6. A normal user shall be able to request to join clubs, then once approved able to join club.
7. A normal user shall be able to search clubs.

8. A normal user shall be able to look at personal calendar.
9. A normal user shall be able to leave clubs.
10. A normal user shall be able to create a profile.
11. A normal user shall be able to view event stats.

#### 4.1.3 Referee Users

12. A referee user shall gain access to stopwatch asset.
13. A referee user shall be able to upload scores.
14. A referee user shall be able to update statistics.

## 4.2 Non-Functional Requirements

1. The system should be able to operate navigation outside of a network.
2. The system must not be a significant drain on battery life.
3. The database will be T-SQL.
4. The backend will be C# .NET using MVC.
5. The frontend website will use AJAX calls instead of form submissions, to provide a faster, more seamless experience to the user.
6. Performance is critical, any visible delays or appearance of inefficiency will need to be corrected so that the user has a quick, smooth interaction with the software.
7. Role base access control will be implemented.

## 5. Competition

---

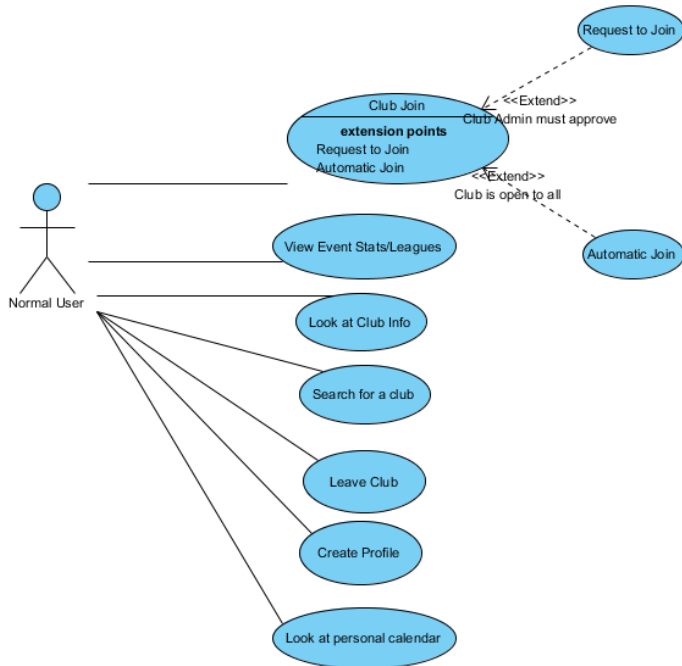
	User-Friendliness	Schedule	Nonintrusive Advertising	Efficiency	Statistics
SquadFusion	Some	Full	Some	Full	Some
IMLeagues	None	Full	None	None	Partial
Crimson Clubs	Full	Full	Full	Full	Full

## 6. Diagrams

---

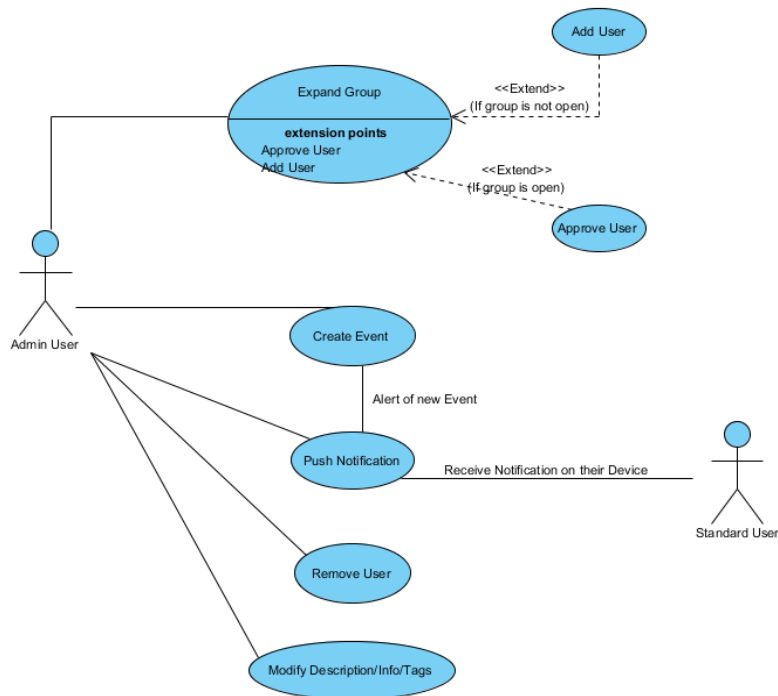
### 6.1 Normal User Use Case Diagram

This use case explores all the functionality of the normal users and what they can do when looking at the main page. They can search for clubs, see their personal calendar, view events and descriptions, join more clubs, and change details on their personal



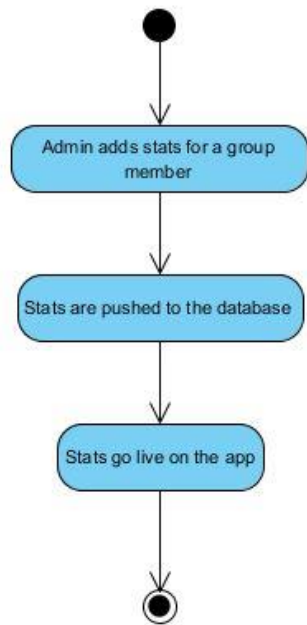
## 6.2 Admin User Use Case Diagram

This use case looks at what the admin of a club can do. They can approve or add users to the club, update information about the club, add events and push notifications to all of the members of the club.



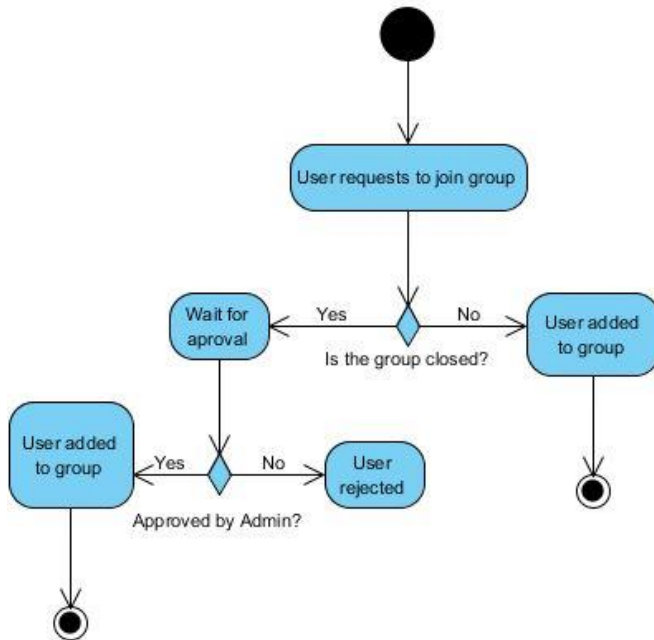
### 6.3 Add Stats Activity Diagram

This functionality is intended to be used by the referees during games or by an admin from the group after the game, so admin privileges are required. The admin would enter the stats into a form and then send the form to the database and the app and website will pull the stats from the database.



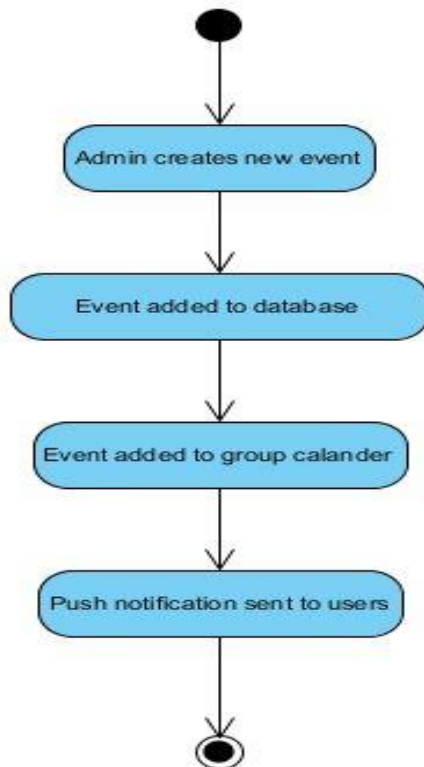
### 6.4 Add User Activity Diagram

There are two different types of groups in the app: open and closed groups. Closed groups are used for group that are not publicly available or have some sort of requirements to join, such as intramural teams, honor societies, sororities, or fraternities. For open groups once a user requests to join the group they are immediately added to the group and can access all of its content. For a closed group the user requests to join the group and then must wait for approval by an admin from that group before they are added and can access the group's content. The admin has the choice to approve or deny any request.



## 6.5 Create Event Activity Diagram

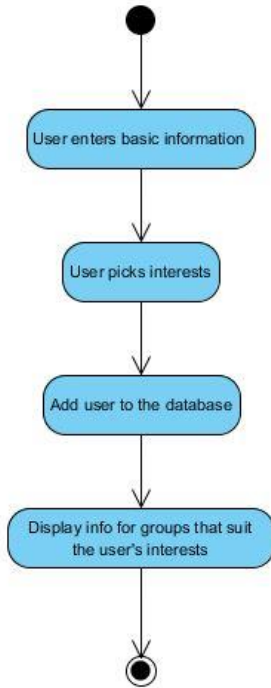
Creating events is reserved for admin users. The admin fills out a form with the information about the event and then sends the information to the database. The event is added to the group's calendar and a push notification is sent out to the group members to inform them of the new event.





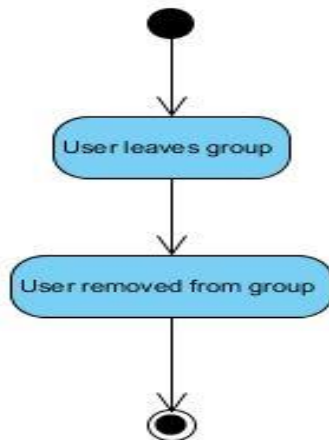
## 6.6 Create Profile Activity Diagram

When a user starts the app for the first time they will be prompted to fill out a profile. They will fill out their basic information along with selecting topics that interest them. This information will be sent to the database and then a list of clubs that match with the user's interests will be displayed. In the future we may extend the interests section somehow to better match users to clubs instead of just basic tags.



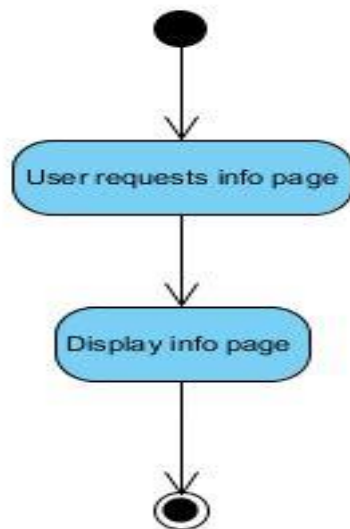
## 6.7 Leave Group Activity Diagram

At any time a user can select to leave a group. Once they choose to leave the group they will be removed from that group and they will no longer receive notifications from the group or have the group's events displayed on their personal calendar.



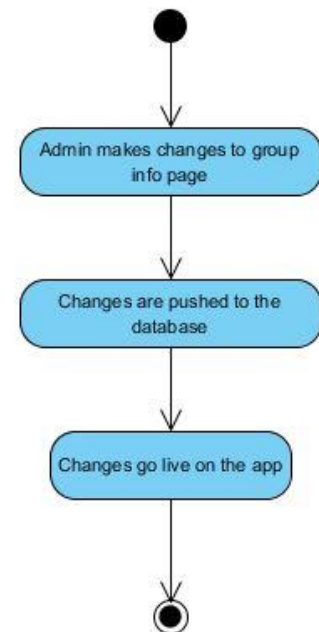
## 6.8 Look at Info Activity Diagram

After a user has searched for clubs they can select a club's page to view more detailed information about the club. Once they select the club the page displaying that club's information will be pulled from the database and displayed.



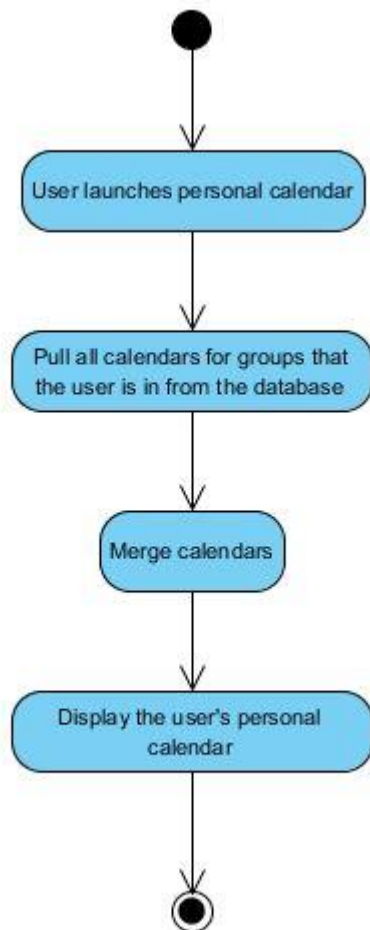
## 6.9 Modify Info Activity Diagram

An admin for a group can select to modify the group's information page. Once selected they will be prompted with a form populated with the current information for the group. After they have made all the changes that they want the new information is sent back to the database and the new information is shown when someone pulls up the page on the app or website.



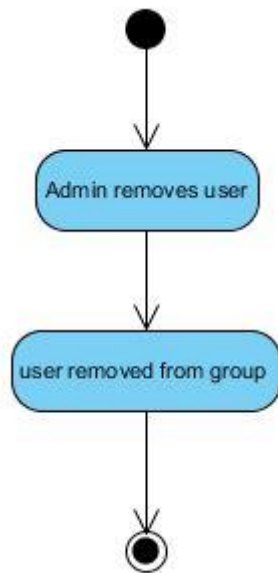
### 6.10 Personal Calendar Activity Diagram

At any time a user can select to launch their personal calendar. Once launched all of the calendars for groups that they are a member of will be pulled from the database and merged into one calendar where they can view all of their events.



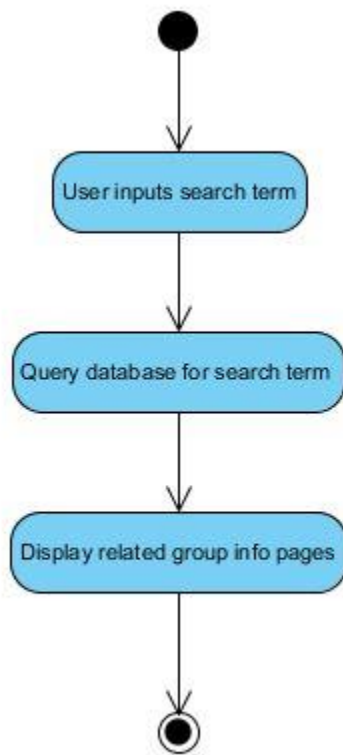
### 6.11 Remove User Activity Diagram

An admin from the group can remove a specific user from their group. The functionality is very similar to a user leaving a group. The admin selects the user from the list of users and selects to remove them from the group. They are removed from the list and will no longer receive notifications or see group events on their personal calendar.



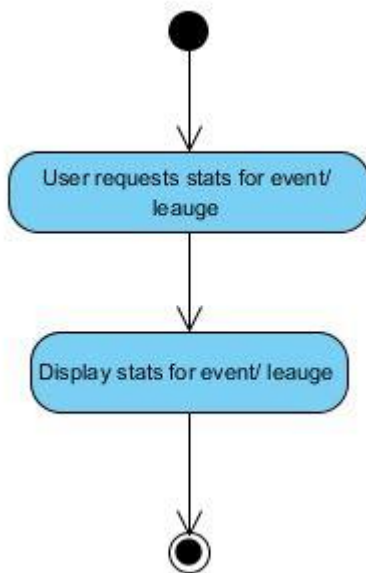
## 6.12 Search Clubs Activity Diagram

The user inputs a search term and the app will query the database for groups that have tags that match the search term. These groups will then be displayed for the user to view. In the future we may implement a more complex system to better match users to groups that they may be interested in.



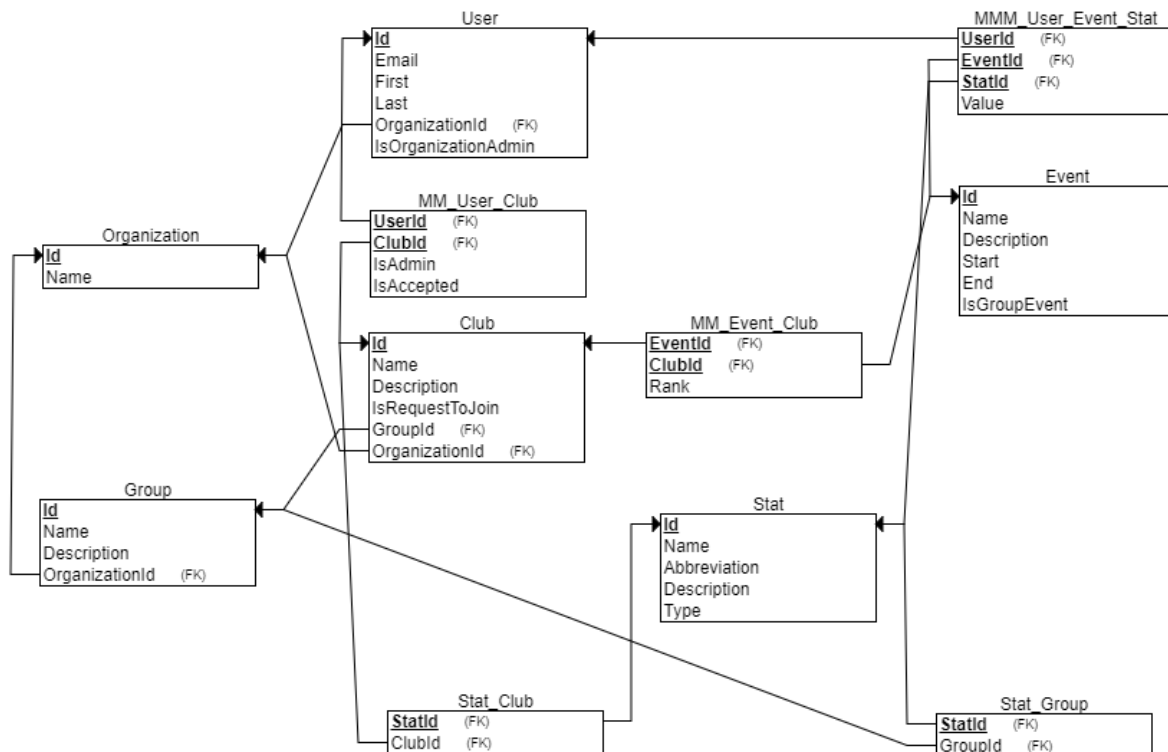
### 6.13 View Event Stats Activity Diagram

A user can request the stats for a specific event/ league. This functionality is designed for the intramural teams and not general clubs. The user will navigate to the specific event/ league and then select to view stats. The stats are then pulled from the database and displayed for the user.



### 6.14 Database Diagram

A first pass at the database. This model attempts to provide all of the features as simply as possible.



## 6.15 Class Diagram

This class diagram shows the expected classes that will be used in the Android application. After a login activity, there will be a main activity with a navigation drawer. The navigation drawer will have different options that change the main activity's current fragment so the user can have a smoother experience while completing the possible tasks. Managing clubs and creating events will be in their own activities. The managing club activity will be activated from within the view club fragment, and the create event will be active from the managing club activity.

