|  |
| --- |
|  |
| Crimson Clubs |
| Software Requirements Specifications |
|  |
|  |
| **Max Livingston**  **Maclane May** |
| **Adam Gabriel**  **Travis Redfield**  **Taylor Meads** |
| **2/27/2018** |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Preface 3](#_Toc507667536)

[2. Introduction 3](#_Toc507667537)

[3. Glossary 3](#_Toc507667538)

[4. User Requirements Definition 3](#_Toc507667539)

[4.1 Functional Requirements 3](#_Toc507667540)

[4.2 Non-Functional Requirements 4](#_Toc507667541)

[5. Competition 4](#_Toc507667542)

[6. Diagrams 4](#_Toc507667543)

[6.1 Normal User Use Case Diagram 4](#_Toc507667544)

[6.2 Admin User Use Case Diagram 5](#_Toc507667545)

[6.3 Add Stats Use Case Diagram 6](#_Toc507667546)

[6.4 Add User Use Case Diagram 6](#_Toc507667547)

[6.5 Create Event Use Case Diagram 7](#_Toc507667548)

[6.6 Create Profile Use Case Diagram 7](#_Toc507667549)

[6.7 Leave Group Use Case Diagram 8](#_Toc507667550)

[6.8 Look at Info Use Case Diagram 8](#_Toc507667551)

[6.9 Modify Info Use Case Diagram 9](#_Toc507667552)

[6.10 Personal Calendar Use Case Diagram 9](#_Toc507667553)

[6.11 Remove User Use Case Diagram 10](#_Toc507667554)

[6.12 Search Clubs Use Case Diagrams 10](#_Toc507667555)

[6.13 View Event Stats Use Case Diagram 11](#_Toc507667556)

[6.14 Database Diagram 11](#_Toc507667557)

[6.15 Class Diagram 12](#_Toc507667558)

# 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.

# 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.

# 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.

# User Requirements Definition

## 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

1. A normal user shall be able to request to join clubs, then once approved able to join club.
2. A normal user shall be able to search clubs.
3. A normal user shall be able to look at personal calendar.
4. A normal user shall be able to leave clubs.
5. A normal user shall be able to create a profile.
6. A normal user shall be able to view event stats.

4.1.3 Referee Users

1. A referee user shall gain access to stopwatch asset.
2. A referee user shall be able to upload scores.
3. A referee user shall be able to update statistics.

## 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 with 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 a quick, smooth interaction with the software.
7. Role base access control will be implemented.

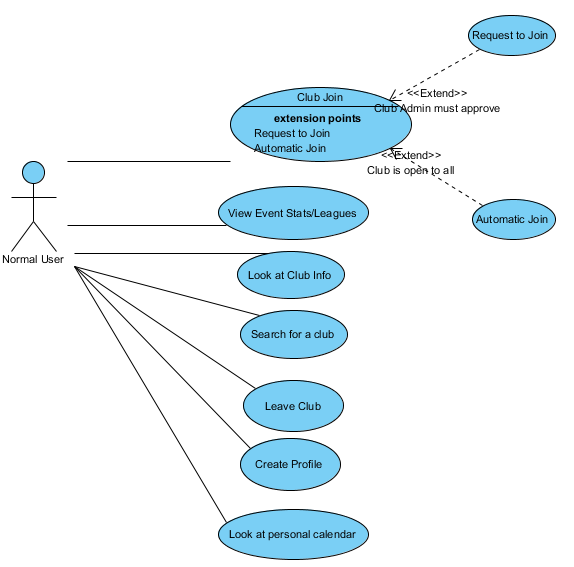
# Competition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

# Diagrams

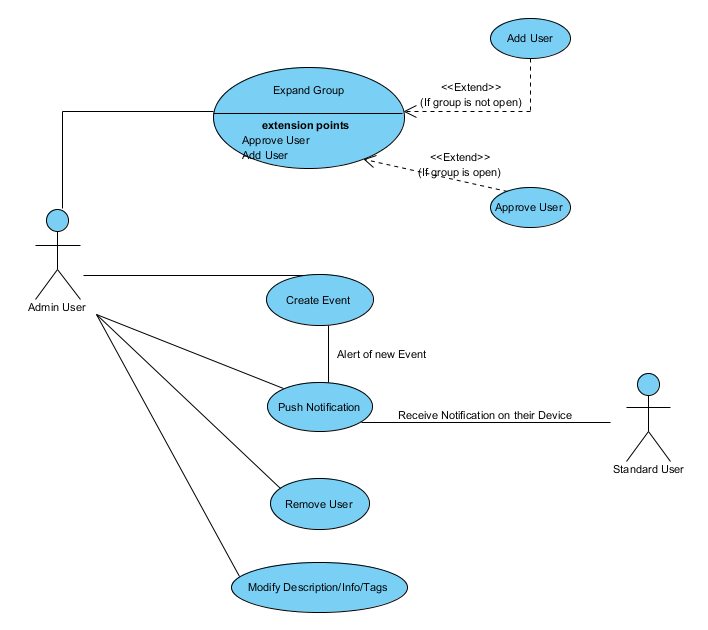
## Normal User Use Case Diagram

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

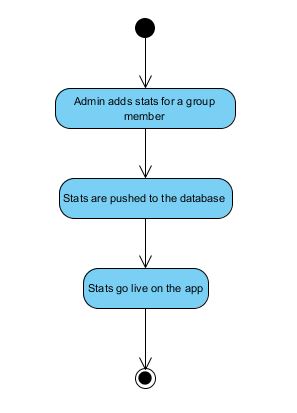


## Admin User Use Case Diagram

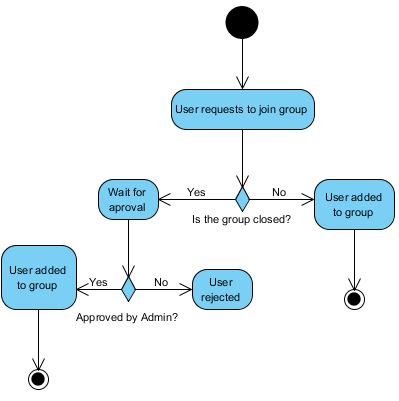
This use case looks at what the admin of a club can do. They can approve or add users to the club, update information about the club, add events and push notifications to all of the members of the club.



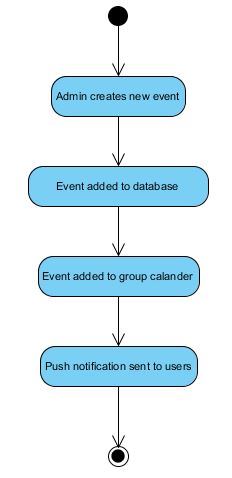
## Add Stats Use Case Diagram



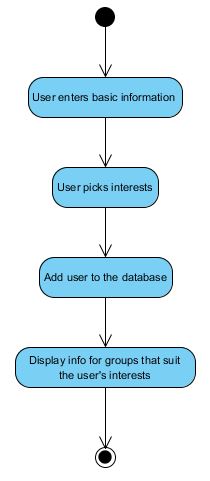
## Add User Use Case Diagram



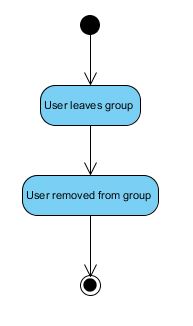
## Create Event Use Case Diagram



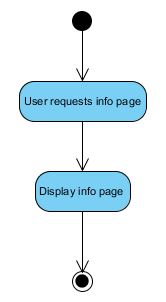
## Create Profile Use Case Diagram



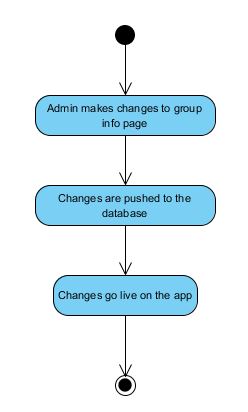
## Leave Group Use Case Diagram



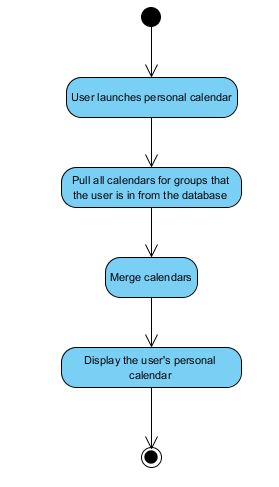
## Look at Info Use Case Diagram



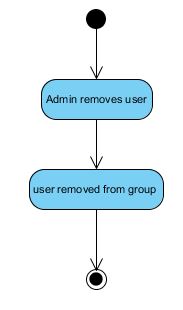
## Modify Info Use Case Diagram



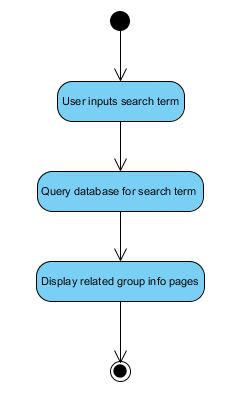
## Personal Calendar Use Case Diagram



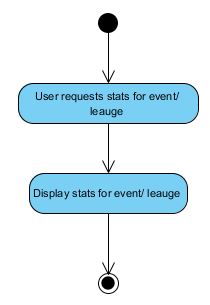
## Remove User Use Case Diagram



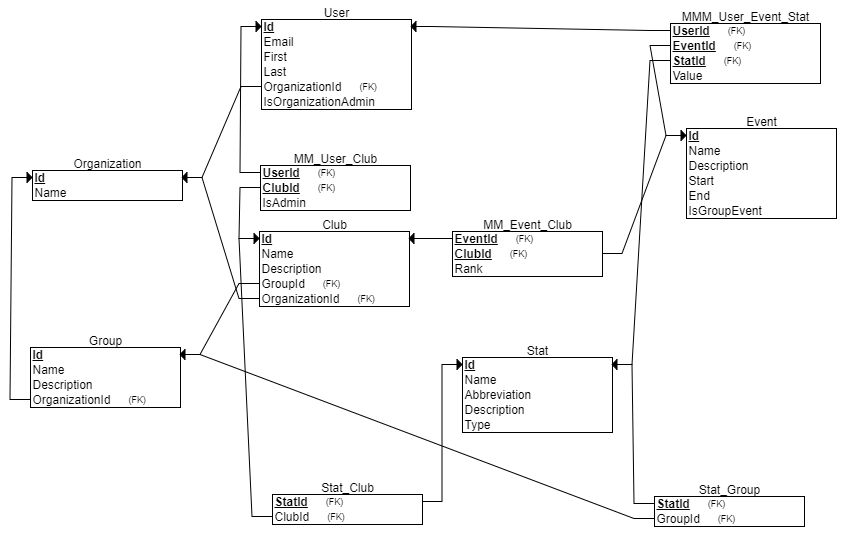
## Search Clubs Use Case Diagrams



## View Event Stats Use Case Diagram



## Database Diagram



## Class Diagram

