

Crimson Clubs

Software Requirements and Design Specifications

**Max Livingston
Maclane May
Adam Gabriel
Travis Redfield
Taylor Meads**

4/3/2018

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
3-29-2018	0.2	Fixed issues from Requirements document	Adam Gabriel
3-31-2018	0.2	Added sequence diagrams and descriptions for some of them	Adam Gabriel
4-1-2018	0.2	Added the rest of the sequence diagram descriptions	Travis Redfield
4-2-2018	0.2	Updated Format and fixed some sequence diagrams	Adam Gabriel
4-3-2018	0.2	Final format updates specifically to class diagram section	Adam Gabriel/Taylor Meads

Table of Contents

1.	Preface	4
2.	Introduction	4
2.1	Purpose	4
2.2	Scope	4
2.3	Goals	4
2.4	Glossary	4
2.5	Project Description	5
2.5.1	Normal User	5
2.5.2	Admin User	5
3	User Requirements Definition	6
3.1	Functional Requirements	6
3.1.1	Admin User	6
3.1.2	Normal User	6
3.1.1	Referee User	6
3.2	Non-Functional Requirements	7
4	Competition	7
5	Activity Diagrams	7
5.2	Normal User Use Case Diagram	7
5.3	Admin User Use Case Diagram	8
5.4	Add Stats Activity Diagram	8
5.5	Add User Activity Diagram	9
5.6	Create Event Activity Diagram	10
5.7	Create Profile Activity Diagram	11
5.8	Leave Group Activity Diagram	11
5.9	Look at Info Activity Diagram	12
5.10	Modify Info Activity Diagram	12
5.11	Personal Calendar Activity Diagram	13
5.12	Remove User Activity Diagram	13
5.13	Search Clubs Activity Diagram	14
5.14	View Event Stats Activity Diagram	15
5.15	Database Diagram	15

6	Class Diagram	16
6.1	Main Class Diagram.....	16
6.2	Main Activity Section	17
6.3	Fragment Section	18
6.4	Managing Club Section	19
6.5	Database Diagram	20
7	Sequence Diagrams.....	21
7.1	Create Profile	21
7.2	User Login	21
7.3	Join Club	22
7.4	Search Club	23
7.5	Look at Calendar	23
7.6	Leave Club.....	24
7.7	View Event Statistics	25
7.8	Create Club.....	25
7.9	Create Event.....	26

1. Preface

This is the second version of the document for design and requirements for the application “Crimson Clubs”. This is an android and web-based application which is focused on helping clubs, organizations, intramurals, and their members in anything related to that organization. This document is designed to help developers and anyone else interested in understanding the project at all levels, from requirements, to design, to implementation.

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.

2.1 Purpose

The purpose of this application is to provide a service to clubs and intramurals of universities to help them with organizing events and members, and to help members of those clubs to plan their schedules and manage the things that they are a part of. Furthermore, specifically for intramurals and other sports, it will allow members to view their specific leagues and schedules and the statistics within that sport. Overall, it is meant to be something that will make things easier for both standard members and administrators of a club or organization.

2.2 Scope

The scope of this project includes many things that will be useful to those people within those organizations. It will include the ability to send and receive push notifications and emails to your phone, the ability to search and join clubs that you might want to join. Also, events can be created and sent to a calendar for the members of an organization and those members will be able to view a single club’s calendar of events, or also merge all of their club’s calendars into one personal calendar.

2.3 Goals

The goals of this app and website are to create a holistic experience that will cover everything that an organization would need to be able to do to help themselves and their members to have all the information in one place. There will no longer be a need to look through group chats or go online to each group’s website to find out what you need to know, it will all be available after a couple taps or clicks. And for administrators, there is no need to keep reminding with emails or messages, because it will be easy to just send a notification to the phone of each of your members to remind them of upcoming events.

2.4 Glossary

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.

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.

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

2.5 Project Description

2.5.1 Normal User

Profile

A user will be able to create a profile with interests and use that profile to look for organizations that they may want to be a part of.

Search

A user will be able to search for clubs based on search terms or tags and then receive the results based on relevance.

Calendar

A normal user will be able to view a calendar of any public group or also view a personal calendar that combines the calendars and events of all the groups that they are in.

Join/Leave

A normal user will be able to join any group that is open, and will be able to request entrance into any group that is currently closed, and will be added pending the acceptance of an administrator.

A user will also be able to leave a group at any time regardless of whether or not the group is closed.

2.5.2 Admin User

Normal User

An admin user has all the same privileges as a normal user, but becomes an admin user when they go to the club page of a page that they are an admin for.

Edit Description

An admin user will be able to edit the information/tags/description of any group that they have administrator access to.

Add Event

The user will be able to add an event for the group and then that event can be added to the group calendar and also push a notification to the group member's phones.

Approve/Kick

An admin user can approve a user that has requested to join the club.

An admin user can also kick any user from the club.

Delete Club

An admin user can also delete a club.

3 User Requirements Definition

3.1 Functional Requirements

3.1.1 Admin User

(High Priority)

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.

3.1.2 Normal User

(High Priority)

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.

(Medium Priority)

10. A normal user shall be able to create a profile.
11. A normal user shall be able to view event stats.

3.1.1 Referee User

(Low Priority)

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.

3.2 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. 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.
4. Role base access control will be implemented.

4 Competition

	User-Friendliness	Schedule	Nonintrusive Advertising	Performance of Application	Statistics
SquadFusion	Some	Full	Some	Full	Some
IMLeagues	None	Full	None	None	Partial
Crimson Clubs	Full	Full	Full	Full	Full

5 Activity Diagrams

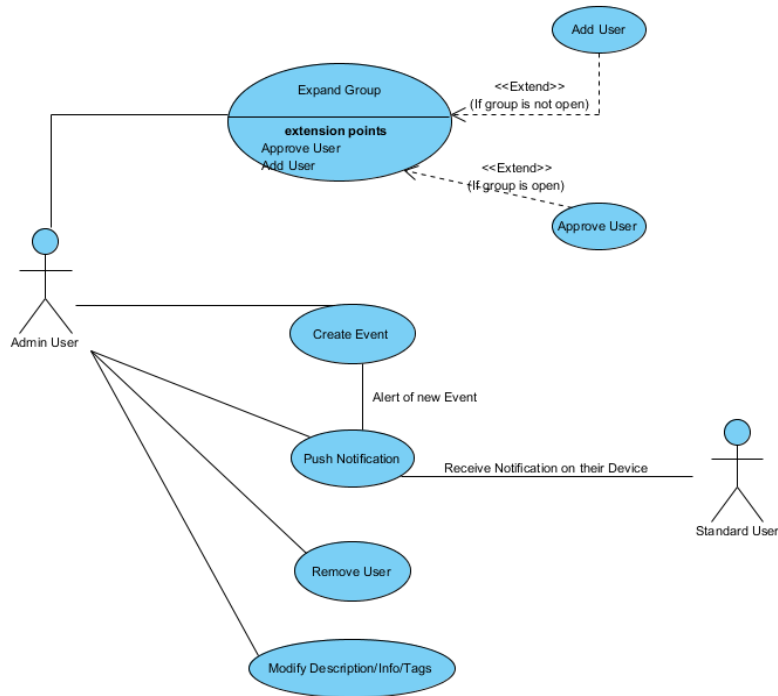
5.2 Normal User Use Case Diagram

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 profile.



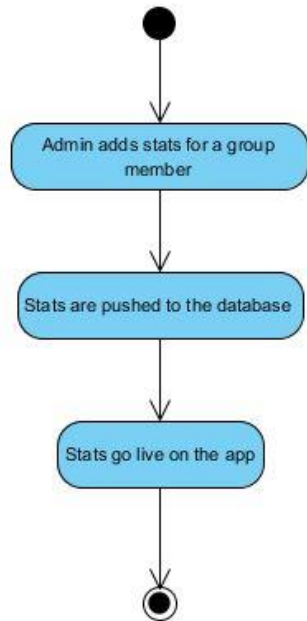
5.3 Admin User Use Case Diagram

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.



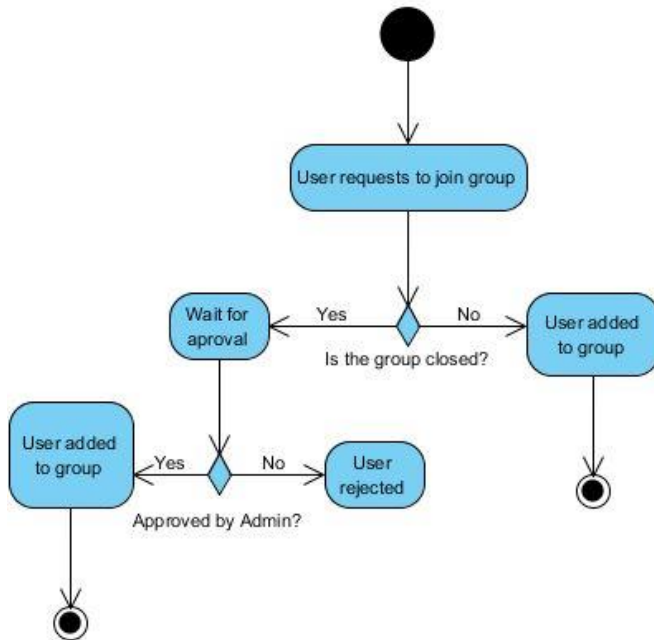
5.4 Add Stats Activity Diagram

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.



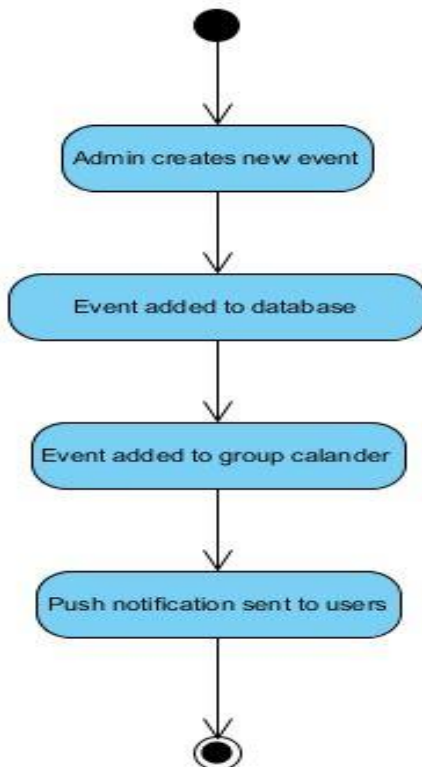
5.5 Add User Activity Diagram

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.



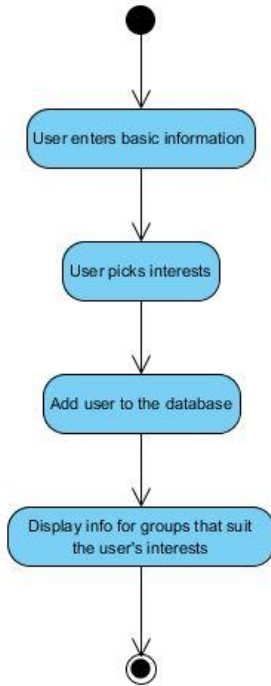
5.6 Create Event Activity Diagram

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.



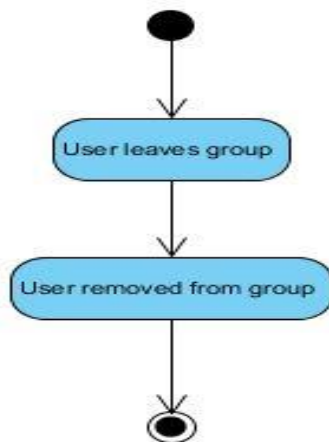
5.7 Create Profile Activity Diagram

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.



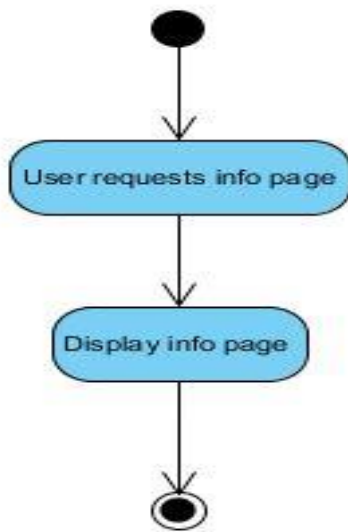
5.8 Leave Group Activity Diagram

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.



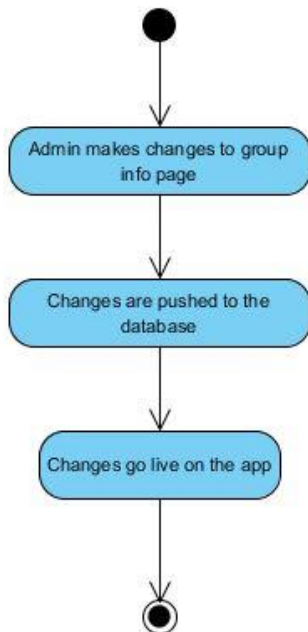
5.9 Look at Info Activity Diagram

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.



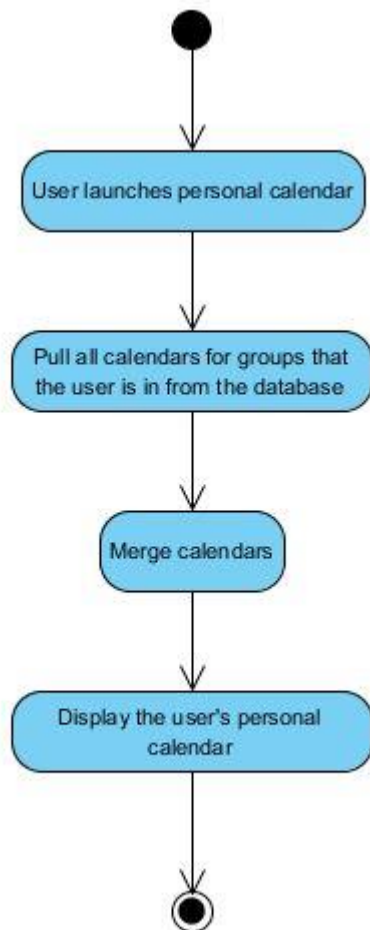
5.10 Modify Info Activity Diagram

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.



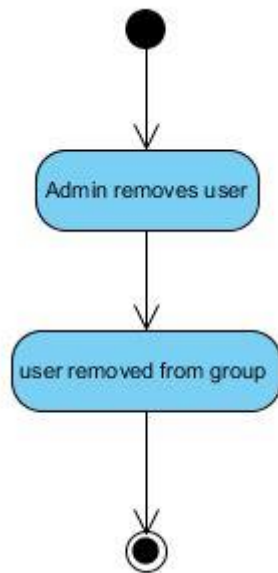
5.11 Personal Calendar Activity Diagram

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.



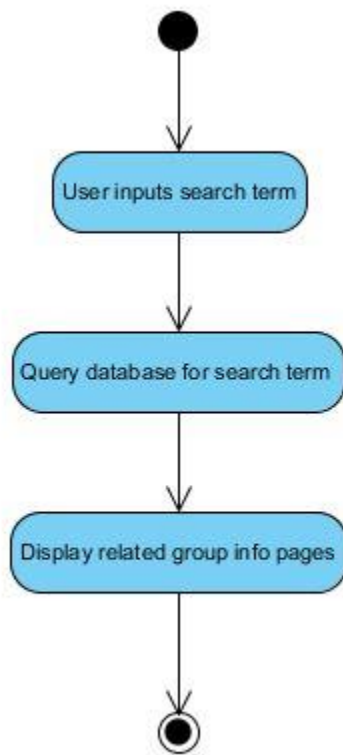
5.12 Remove User Activity Diagram

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.



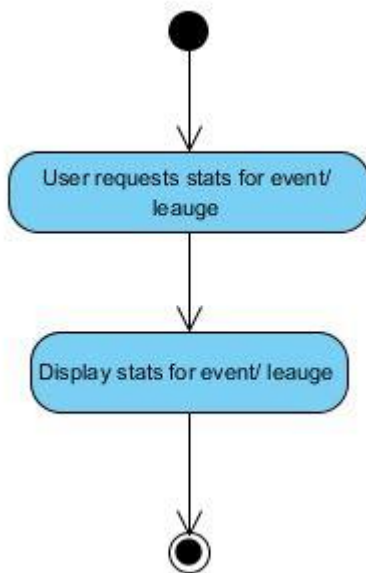
5.13 Search Clubs Activity Diagram

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.



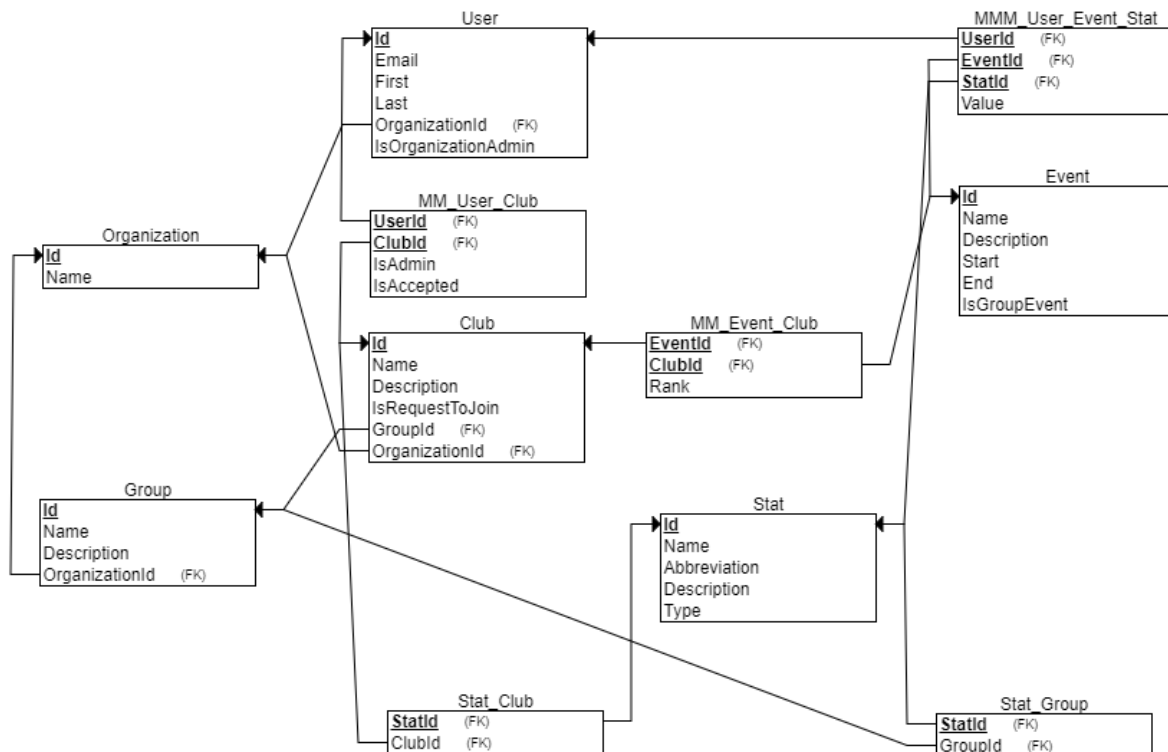
5.14 View Event Stats Activity Diagram

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.



5.15 Database Diagram

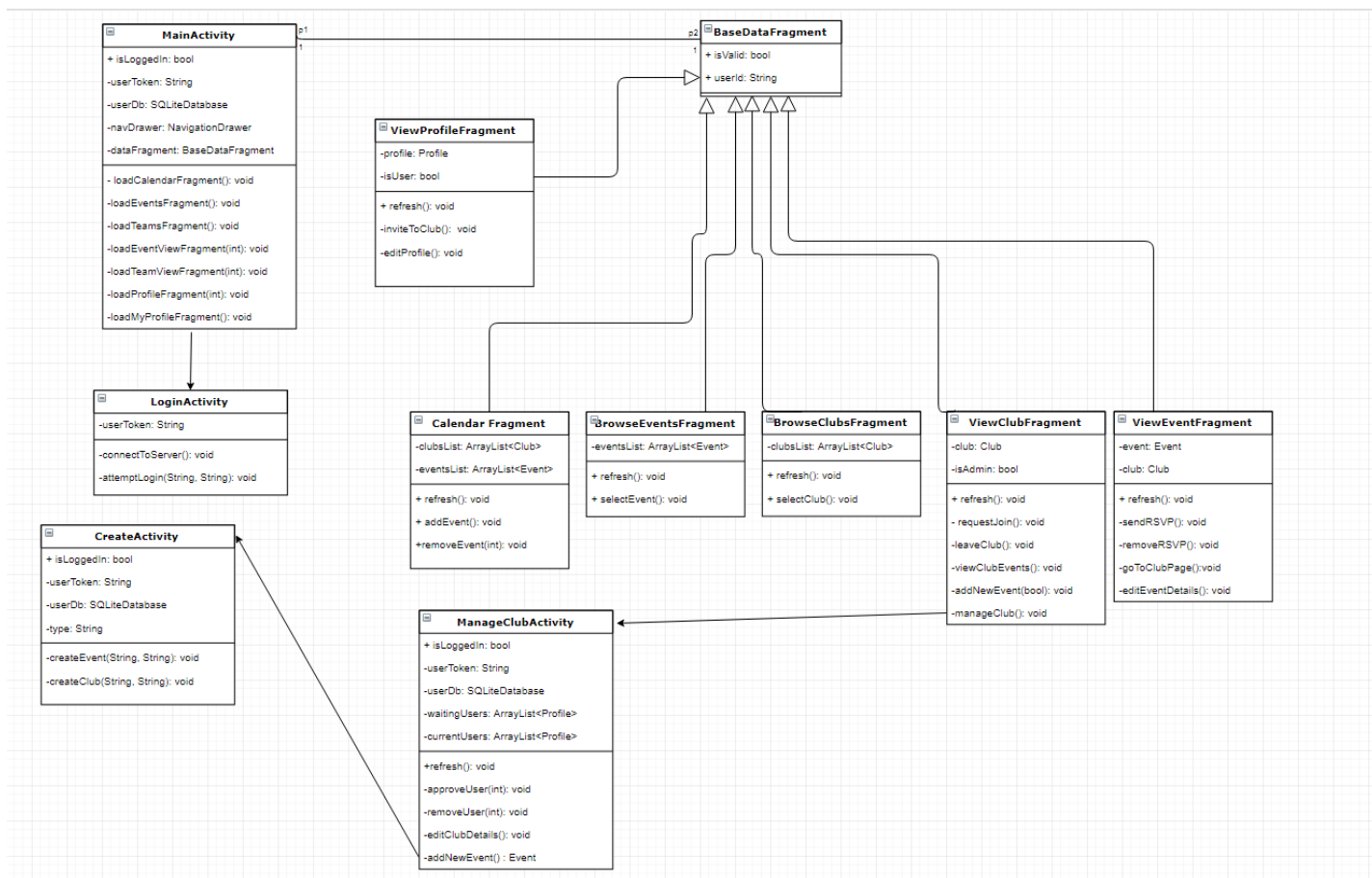
A first pass at the database. This model attempts to provide all of the features as simply as possible.



6 Class Diagram

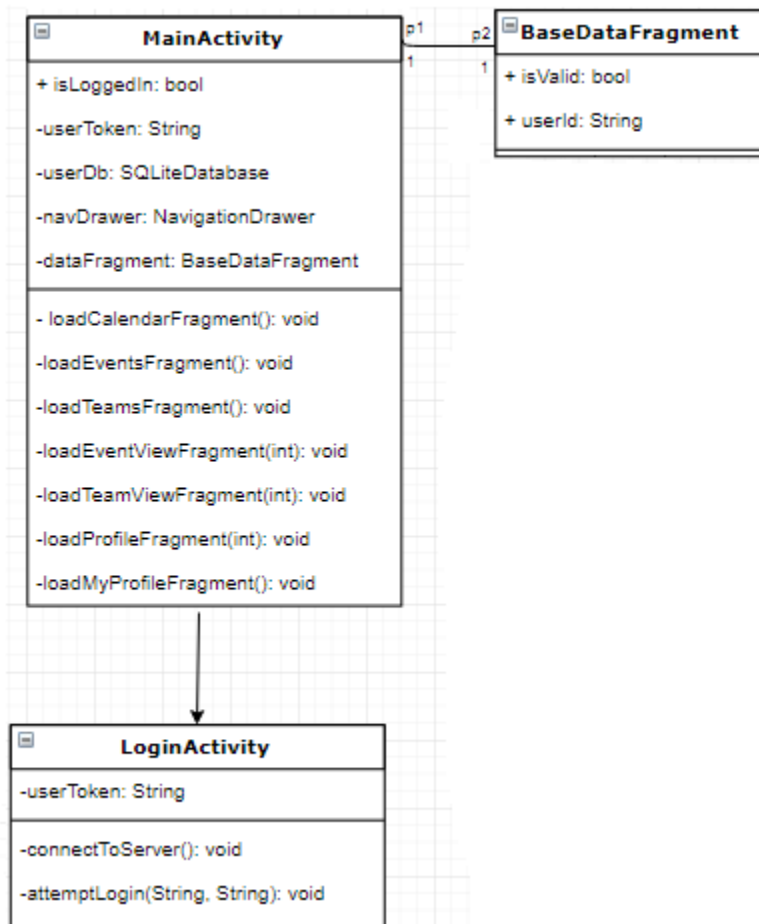
6.1 Main Class Diagram

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. Below, the class diagram is broken up and explains each section in more detail.



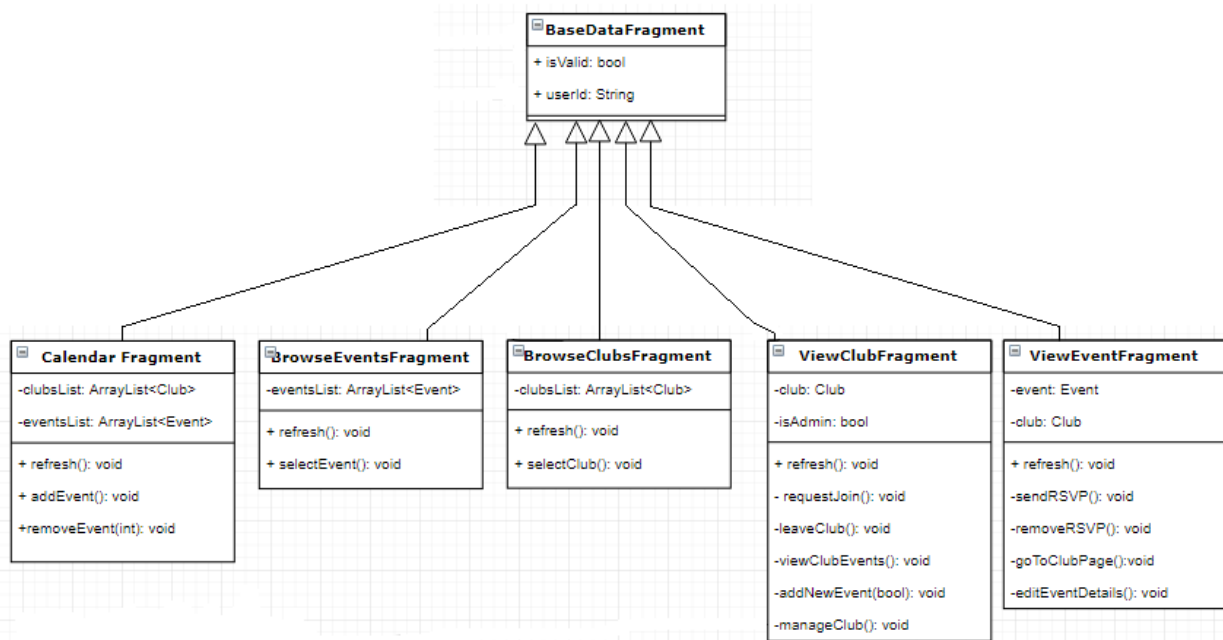
6.2 Main Activity Section

This section of the class diagram primarily deals with setting up and initializing the rest of the application. The MainActivity is brought up at launch, and that is used to load all of the other fragments that will perform most of the actual functions of the application. Furthermore, the main activity handles login by calling the LoginActivity and checking whether the user is logged in when they open the application. The login activity will then be used to log the user in if they are not currently logged in.



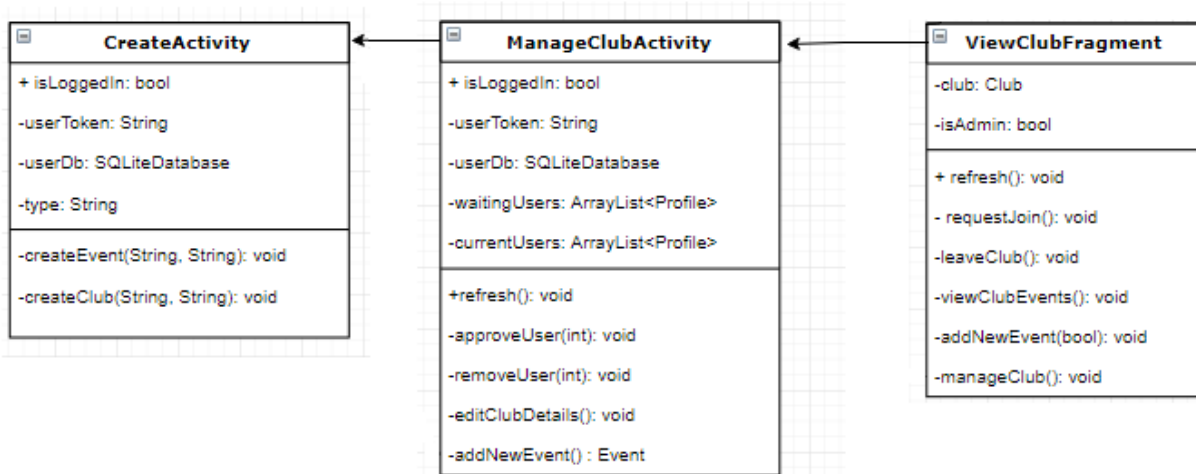
6.3 Fragment Section

This section of the class diagram is based on the BaseDataFragment and all of the other fragments base off of that. The fragments will be the main thing that handles all of the processes of the application. The calendar fragment will handle adding and removing events from the calendar for the specific group, and also for each person. After that, the browse events fragment will allow users to search through their events and the events of a specific organization. The Browse Clubs fragment will have the same functionality, but transferred over to deal with clubs, and not events. Lastly, View club and view event fragments will work nearly the same, just with looking at the specifics of a club or an event. These are all subfragments of the BaseDataFragment and will work to control much of the main functionality of the application.



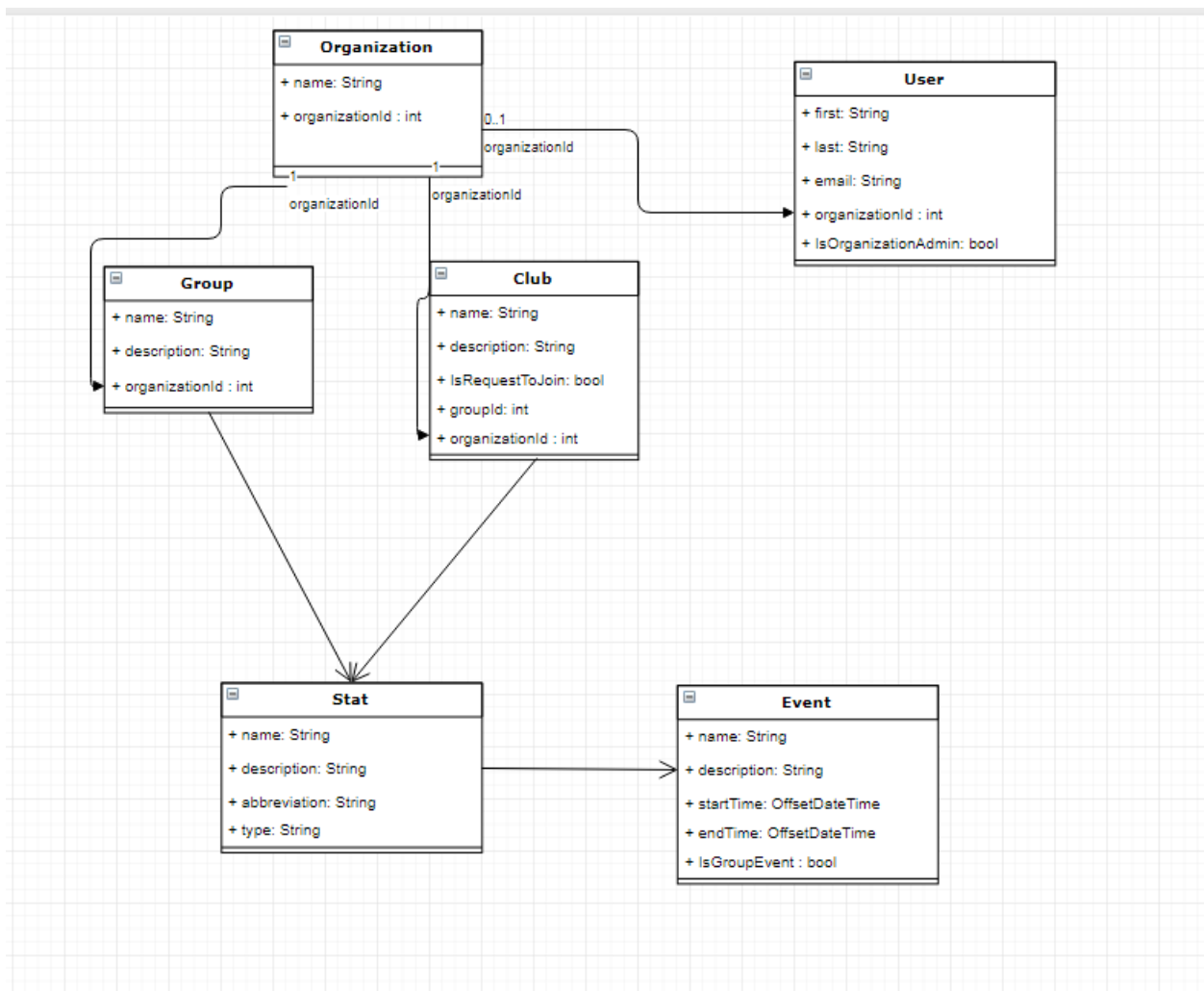
6.4 Managing Club Section

This section of the class diagram mostly pertains to the managing of a club within the application. If a user is an administrator for that specific club, then they will be redirected to the manage club activity if they click that button from the view club fragment. From there, they will be able to edit club details, add and remove users within the club, add new events for that club, and anything that an administrator would want to do to help run that organization they are managing. If they wish to create a new event or club, then they will be redirected to the create activity, which will help them to create a new event or a new club if they wish.



6.5 Database Diagram

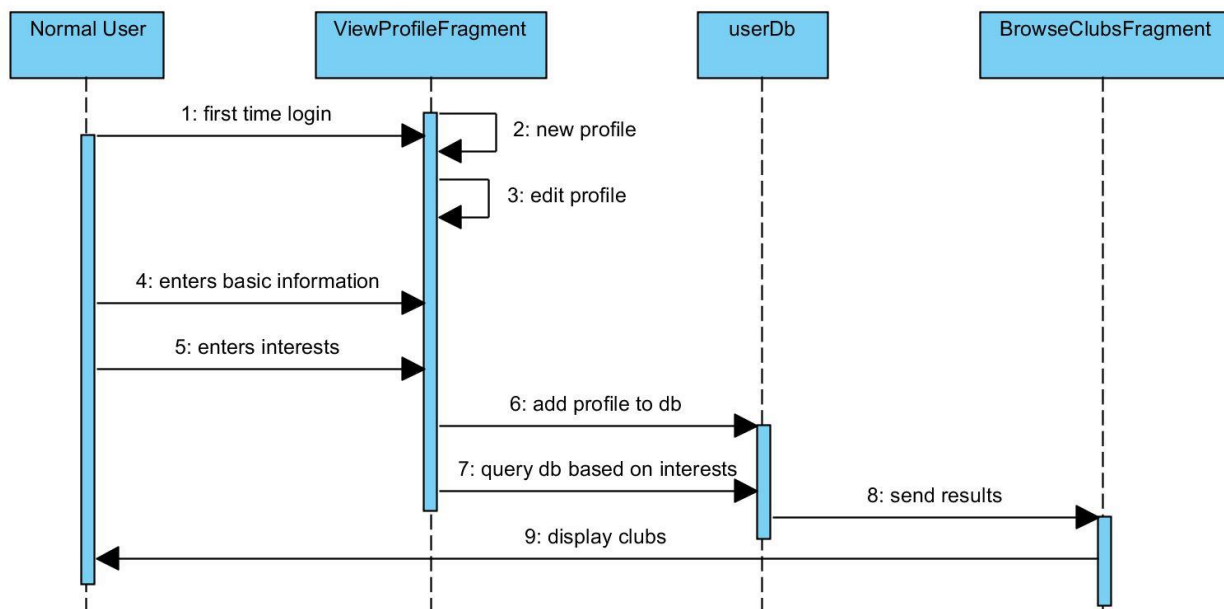
Next is a breakdown of how the information and data is going to be stored internally within the app. There will be a base Organization class, which will have subclasses group, club, and user. These will store information like names, descriptions, Id's, and any other common information that is necessary. After that, "Stat" is a subclass that inherits from both Group and Club which will handle whatever kind of statistics that are tracked; things such as size, data created, officers, etc. This can then also be used for intramurals to track things related to the team or league; such as record and any stats pertaining to specific sport that they are participating in. Finally, Event is a subclass of statistic and will deal with all the different kind of events that will be created by ay group or team.



7 Sequence Diagrams

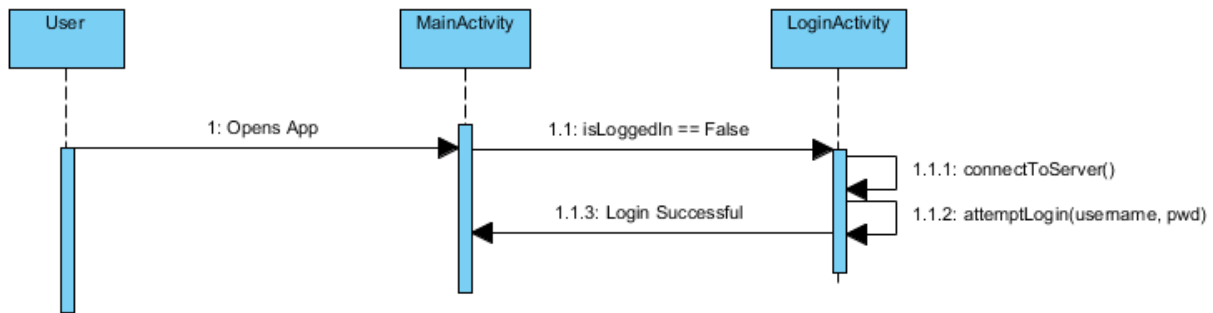
7.1 Create Profile

The first time that a starts the app they will be prompted to login or create a new account. After selecting to create a new account they will be prompted for a username and password. After this they will be prompted for some basic information such as: name, major, year in school, etc. The user will then be prompted to fill out some sort of survey or select topics that they are interested in. This information will be used to match the new user with clubs and organizations that they may be interested in. These will be displayed for the user after they enter their information. The interest selection and pairing with clubs and organizations is only a medium priority, so it is not certain what this will look like in the final build of the app.



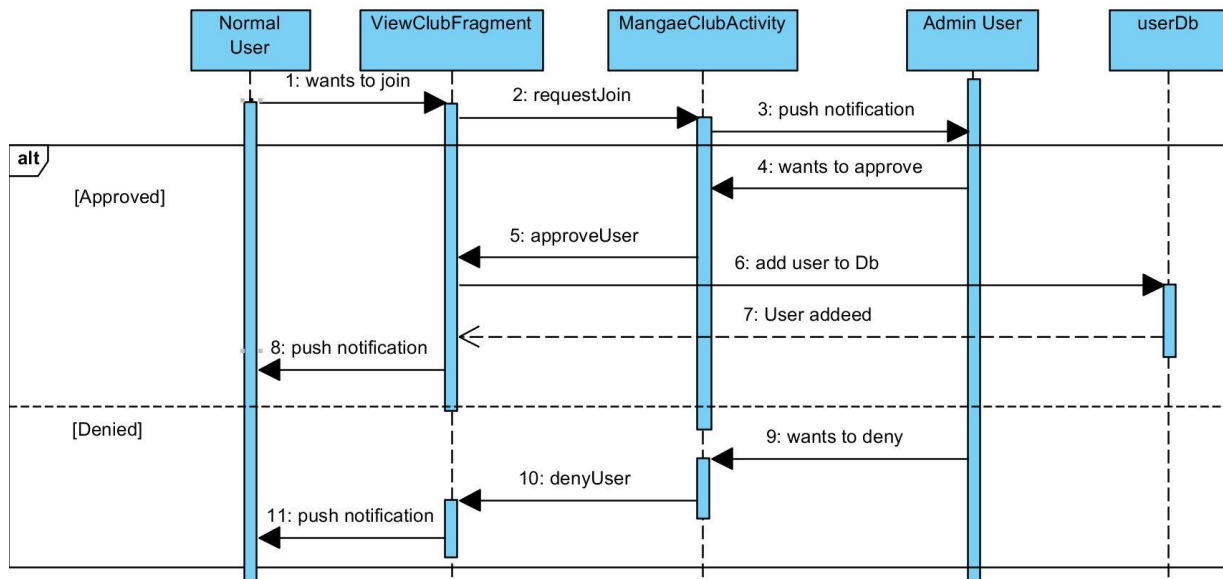
7.2 User Login

When a user first opens the app or website, the app will check to see if they are logged in or not. If not, then the app will connect to the server, the user will then enter their username and password, and if they are both correct and exist in the database, then the login will be successful and they will be directed to the main screen.



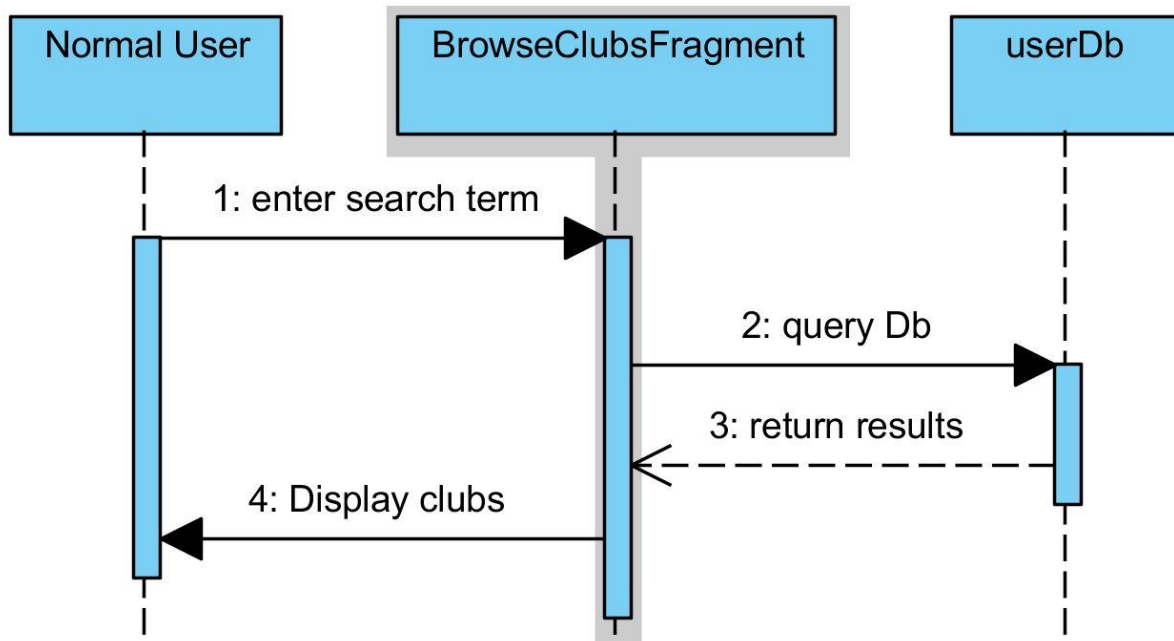
7.3 Join Club

As previously stated there are two types of groups in the app open and closed. Open groups are more trivial to join, the user just pushes the request to join button and is added to the group, so the diagram shows the process for joining a closed group. The user first navigates to the club or organization that they are interested in and then pushes the request to join button. This will send a push notification to an admin of the group and the admin can either approve the request or deny it. If the request is denied the user will be notified that their request was denied. If their request was approved then they will be added to the group in the database and then will be notified that they have been added to the group.



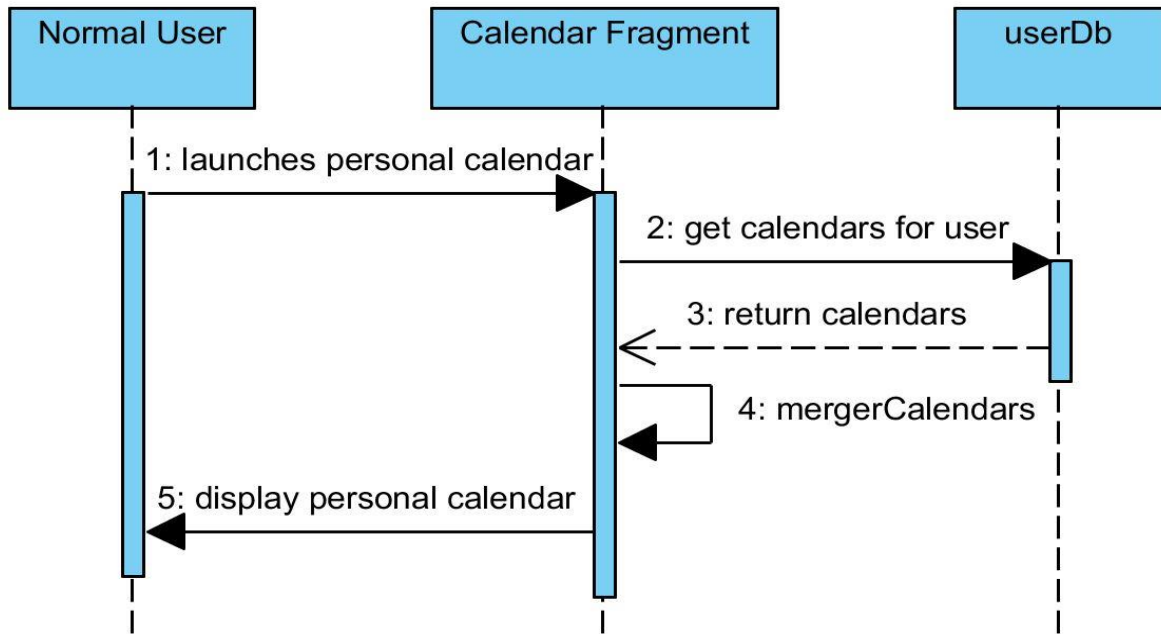
7.4 Search Club

To begin the user will launch the browse clubs fragment and will enter in their search term. The fragment will send the search term to the database as a query. The database will return the results of the query back to the fragment for it to display to the user.



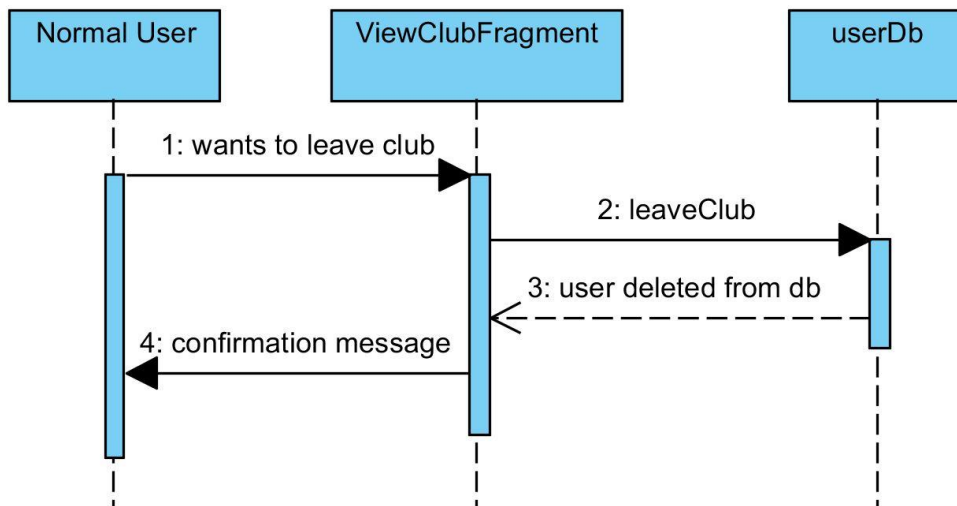
7.5 Look at Calendar

First the user will launch their calendar fragment. The calendar fragment will pull the individual calendars of every group that the user is in from the database. These calendars will then be merged together into one calendar so that the user only needs to look in one place to get the relevant information about events for all of the groups that they are a part of. After they are merged the fragment will display the user's calendar.



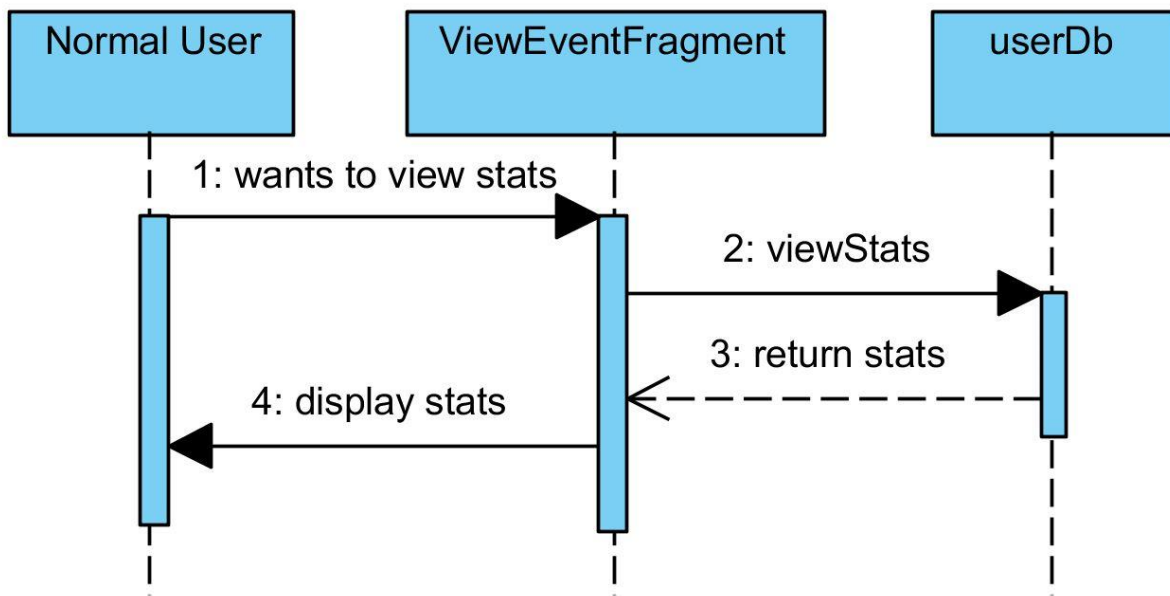
7.6 Leave Club

If a user decides that they wish to leave a group, all that they need to do is navigate to the group's information page and then push the leave group button. This will send a message to the database to delete that user from the group's roster. Once they have been removed they will receive a confirmation message saying that they have successfully left the group. The process for removing a user is similar. The admin of the group will need to go to the group management page and select remove member and then select the member to be removed and then the rest of the process is the same.



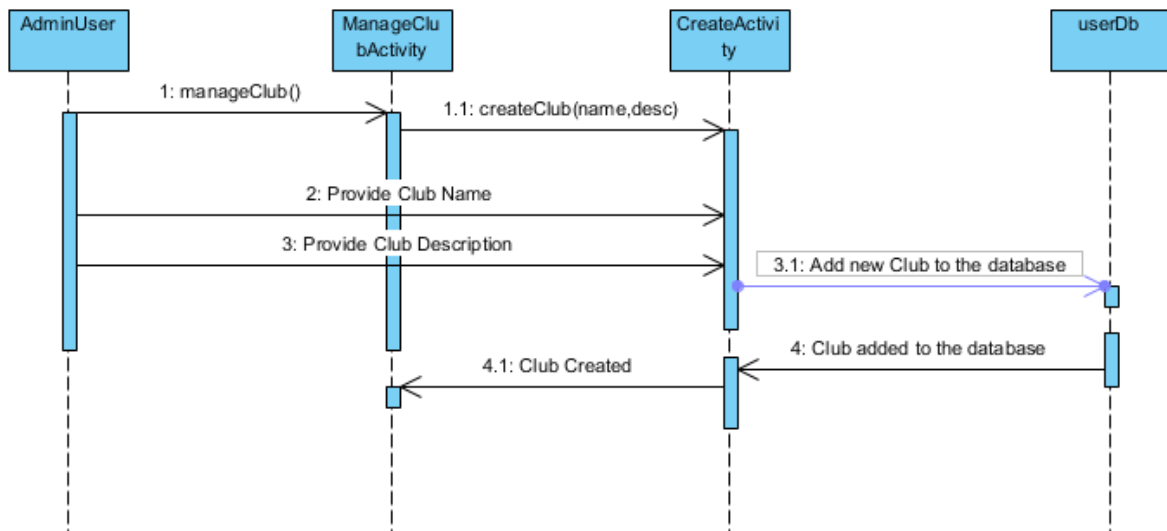
7.7 View Event Statistics

To view the statistics for a particular event a user must first navigate to the event they wish to view stats for. Then they push the view stats button, this prompts the view event fragment to pull the stats from the database. The stats for the event are then displayed for the user to view. The referee functionality of the app is a lower priority so the scope of the stats that are available will be dependent on the amount of time left to devote to the referee functionality.



7.8 Create Club

If a user wants to create a new club, then they must go to the manage club section, and then select create new club. After that they will be prompted to enter a name and a description for that club. The club will then be added to the database and then a message will be passed back saying that the club has been successfully created and the user can go about managing that club as normal.



7.9 Create Event

If an administrator user wants to add a new event for their club, first they must enter the manage club screen for their club. Then they must push the button to add a new event, which will then ask for a name and a description for that event. The event will then be added to the database, and also added to the calendar for the group. After all of that, a message will be passed back saying that the event has been successfully created.

