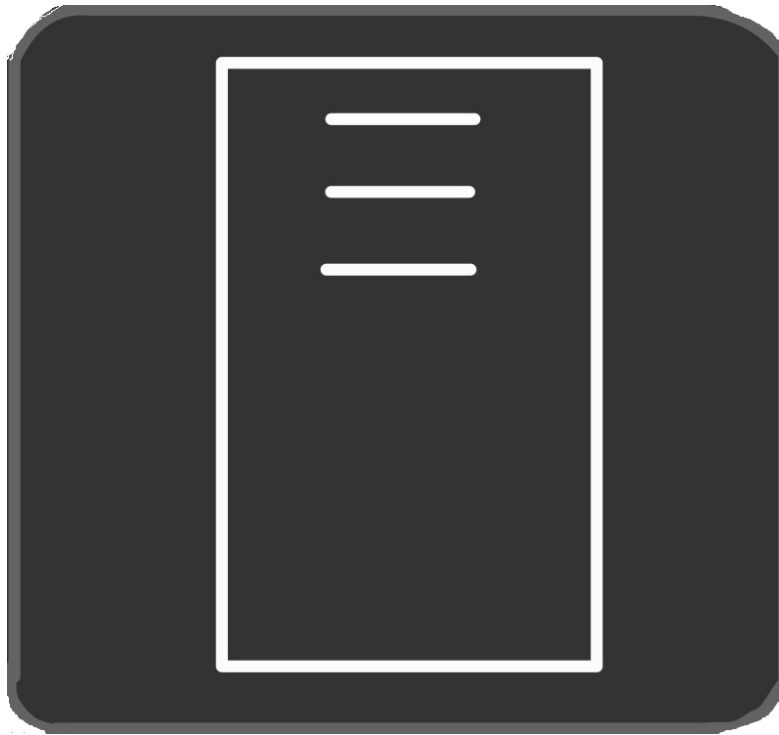


Design Document



The Locker Room

Jordan Ramkalawan (CSE) jramkalwan2018@my.fit.edu

Saeed Binsalam (CSE) sbinsalman2019@my.fit.edu

Matthew Rampersad (ECE) mrampersad2019@my.fit.edu

Faculty Advisor:

Dr. Eraldo Ribiero (eriberio@cs.fit.edu)

Table of Contents

	Page Number
1. Overview	
1.1 Purpose	2
1.2 Scope	2
1.3 Intended Audience	2
2. System Architecture Diagrams	
2.1 Overall structure for The Locker Room System with its main components	3
2.2 Overall Web Application System Architecture	3
2.3 System Architecture diagram pertaining to profile implementation	4
2.4 System Architecture diagram pertaining to friends implementation	5
2.5 System Architecture Diagram pertaining to Searching for Sweats	6
2.6 Class Diagram containing information relevant for Profiles, Friends and Groups component implementation	7
3. Modules	8
4. Entity Relationship Diagram	10
5. Graphic User Interface Diagrams	11

1. Overview

1.1 Scope

This document provides the design details behind The Locker Room. The Locker Room is a web application which enables soccer players to better connect to one another. Users will be given the platform to invite familiar and new users to their unofficial games, also referred to as sweats throughout this project, and to request to join sweats of both new and familiar users. Moreover, users will be able to efficiently plan sweats by making important information such as venue, date, time and attendance readily available, thus removing the dependency on platforms such as Facebook and whatsapp to organize sweats. Additional features include the ability to track personal statistics and create groups with friends.

1.2 Purpose

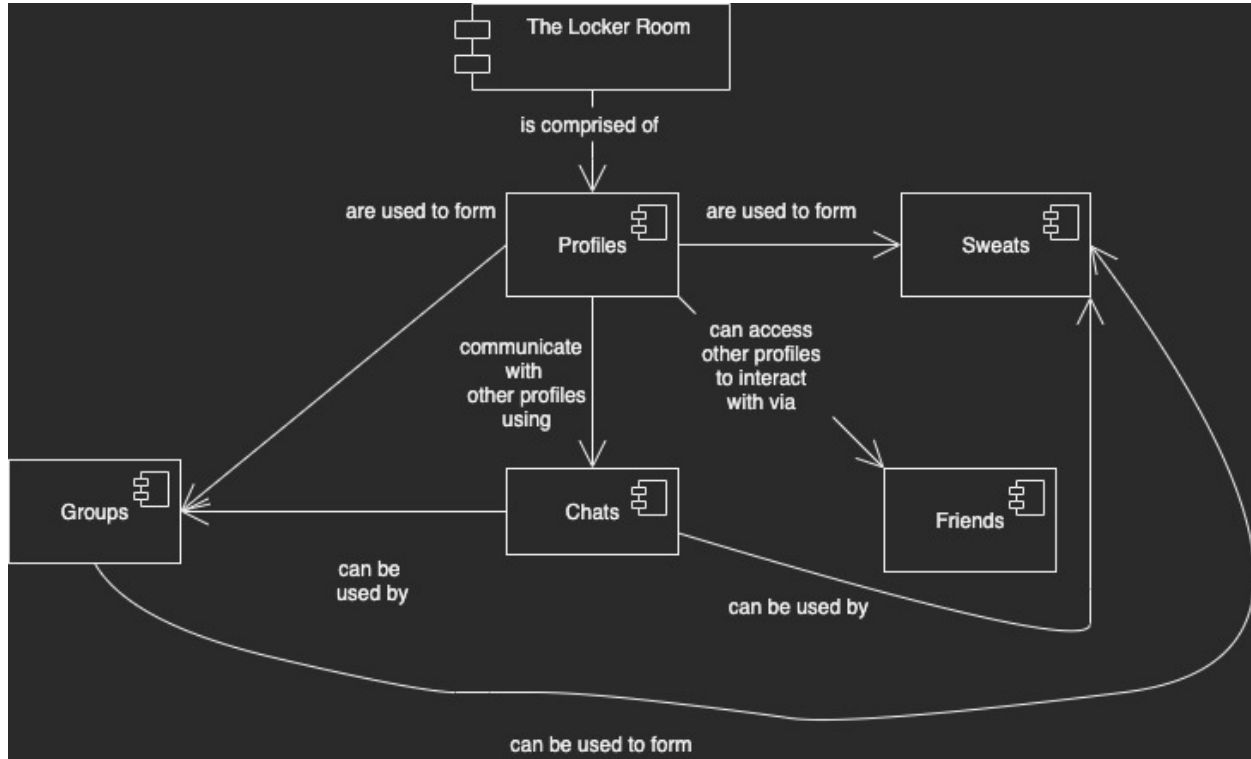
The purpose of this design document is to illustrate the overall design of The Locker Room with its many components. Additionally, this document will highlight the implementation of the profiles and friends components which are foundational to the overall system's performance.

1.3 Intended Audience

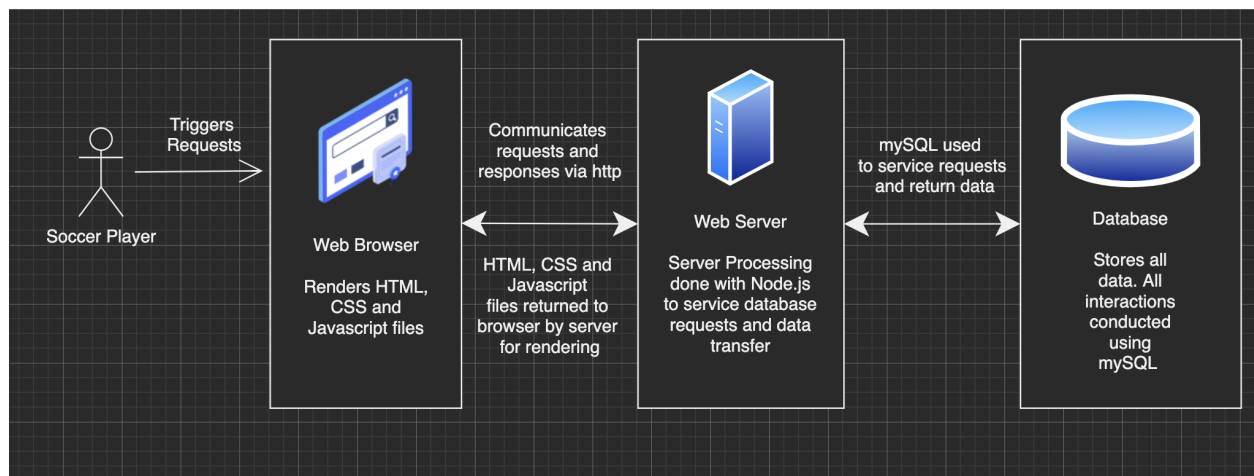
The Locker Room is designed for soccer players who play unofficial games frequently with friends, and who are also open to meeting new people. By using the designed system, searching for new people to play with, as well as organising sweats will be easier than ever before. Although the system is restricted to usage by soccer players currently, there is hope that the services provided by the system can be extended to athletes of other sports such as Basketball and Cricket.

2. System Architecture Diagram

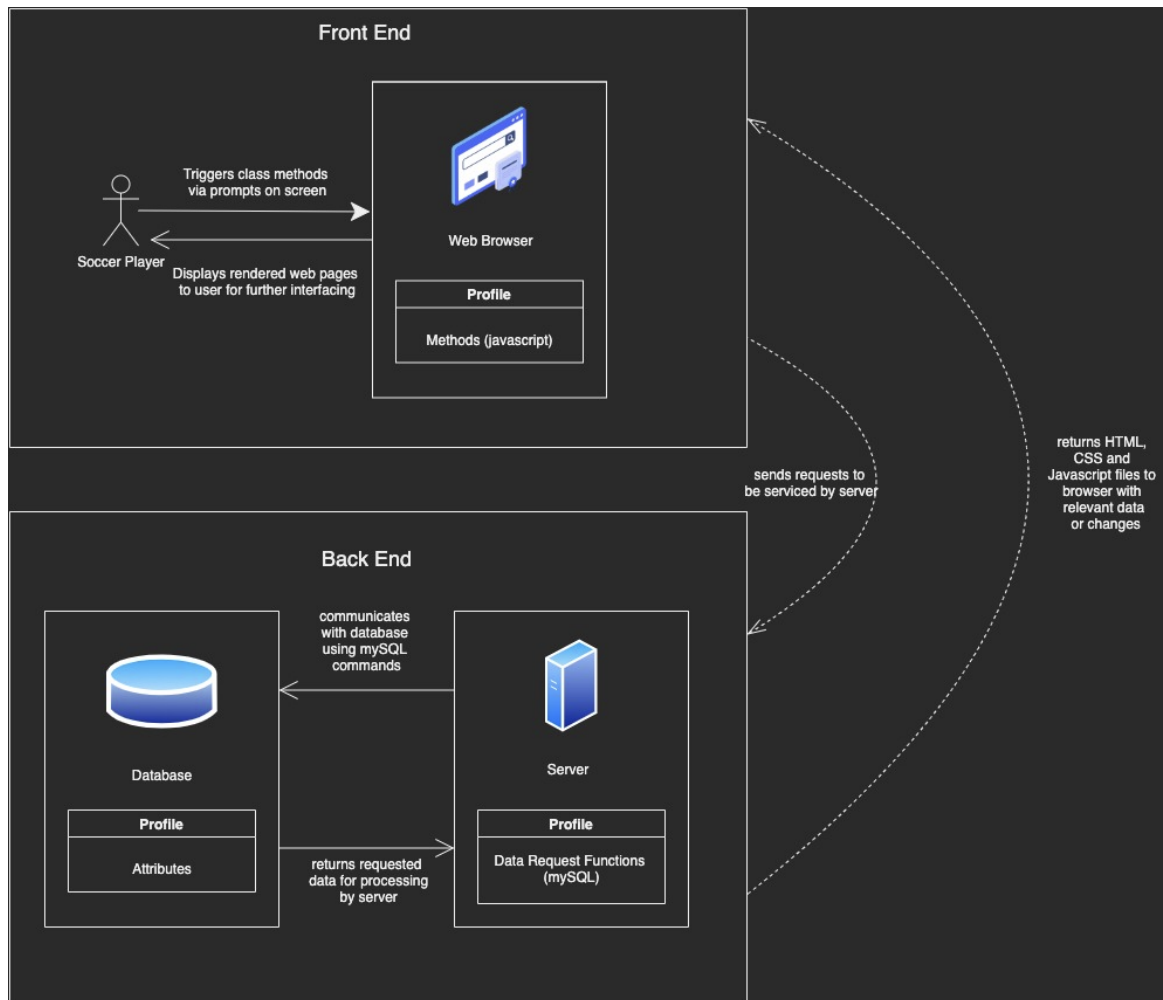
2.1- Overall structure for The Locker Room System with its main components



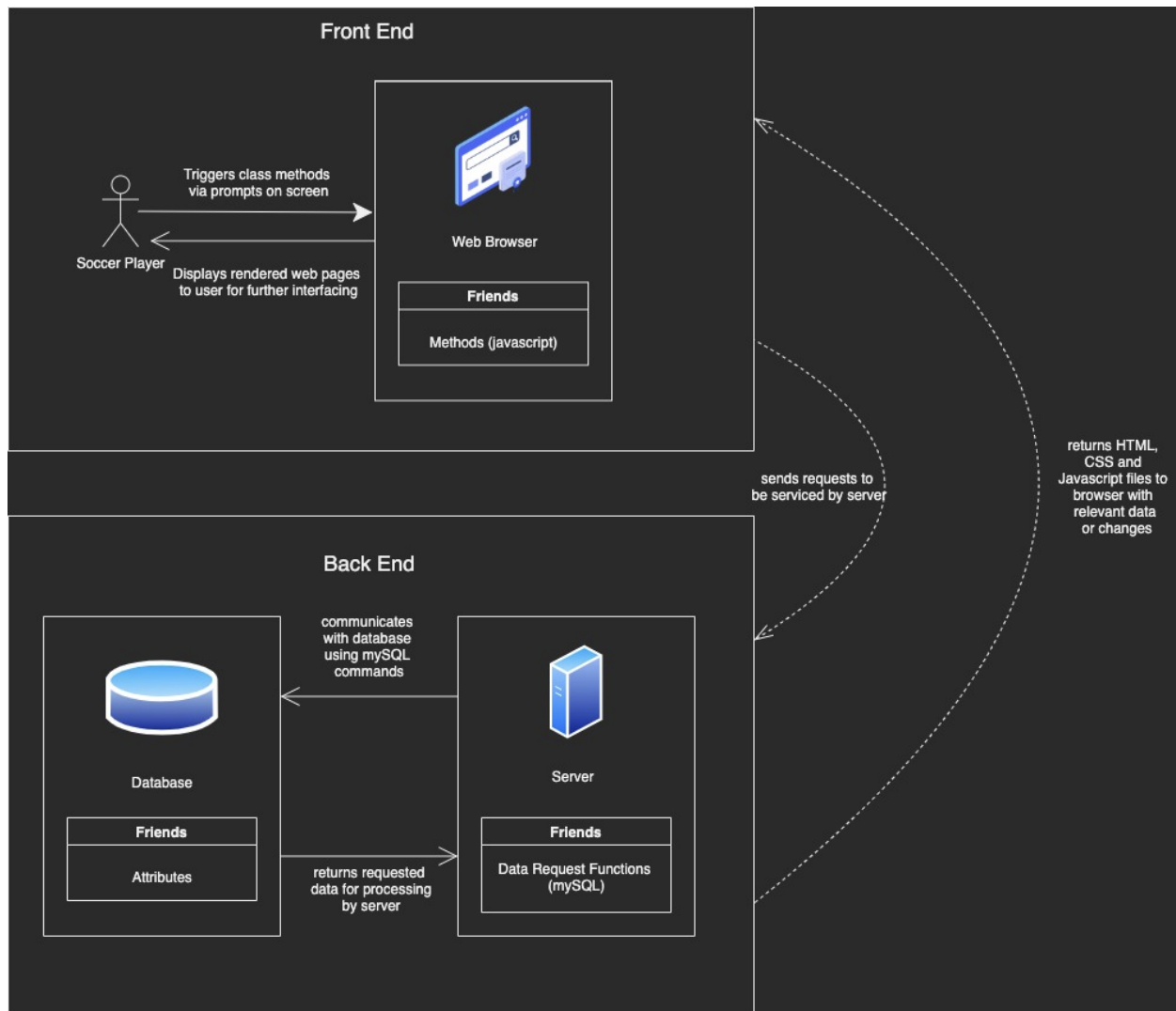
2.2 - Overall Web Application System Architecture



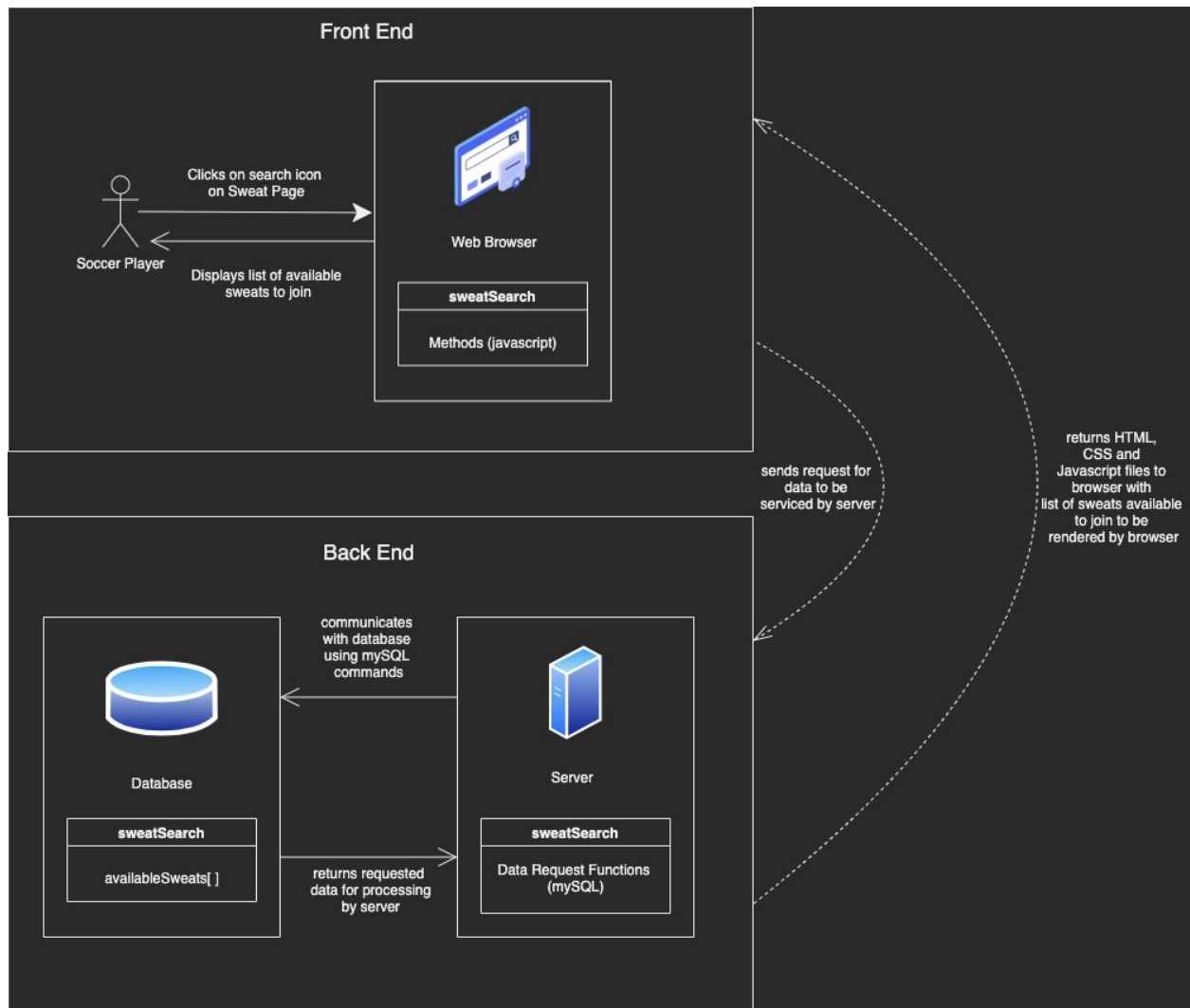
2.3 - System Architecture diagram pertaining to profile implementation.



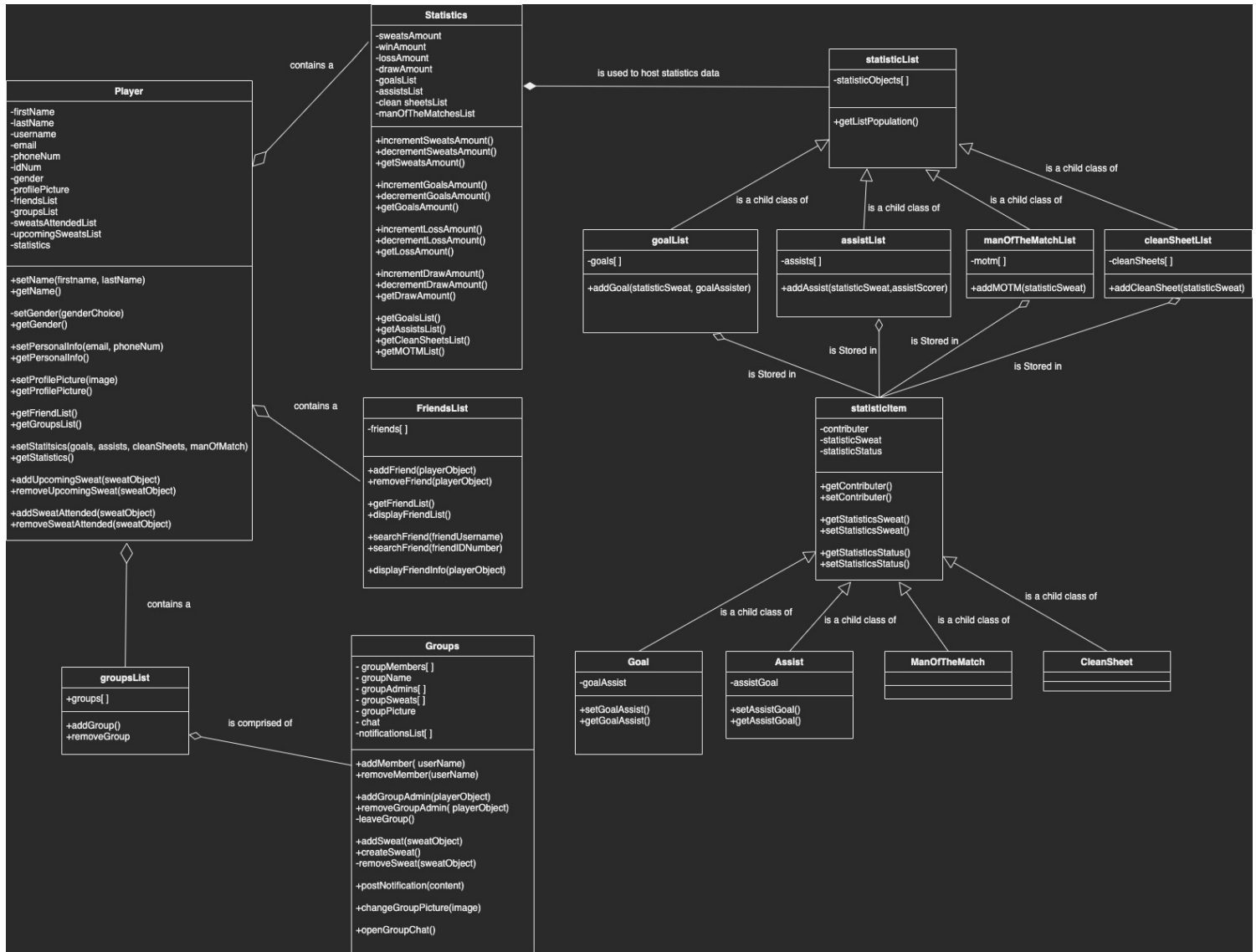
2.4 - System Architecture diagram pertaining to friends implementation.



2.5 - System Architecture Diagram pertaining to Searching for Sweats



2.6 - Class Diagram containing information relevant for Profiles, Friends and Groups component implementation



3. Modules

As seen in the classes diagram in the previous section, there will be many classes used in the development of The Locker Room. The classes used in the implementation of the Profiles and Friends components will be discussed below.

Class: player

The main functionality of the player class is to change, store and retrieve information for each user. Its attributes include personal information such as name, email and phone number, lists containing data on friends, groups, sweats and their statistics. Some of these attributes themselves will be or contain instances of other classes that will be mentioned later on in this section. The methods of the class are mainly setter and getter functions for the class' attributes.

Class: groupsList

This class will be used to store the list of groups which the user is a member of. As such its only attributes will be a list of group objects. Its methods will add and remove groups from this list.

Class: Groups

This class will be the building block for the groups component. Its attributes will be used to store key information such as the group name, group display picture, group's chat, list of notifications, list of group members (list of profile objects) and list of sweats the group played in (list of sweat objects). The methods will be used to interact with these attributes by adding and removing elements from the lists, as well as to create a group sweat.

Class: FriendsList

Similar to groupsList, FriendsList's attribute will be a list of player objects which will be the friends of this instance of the player class. Its methods will be used to add, remove and search for friends, as well as to display the info of a friend.

Class: Statistics

The statistics class will be used to access information on the player's statistics as the name suggests. Lists of goals scored, assists, clean sheets and man of the matches, as well as the amount of games played, wins, draws and losses will make up the class' attributes. The methods will be used to increment tallies and retrieve lists

Class: Statistics List

This class is a parent class for each type of statistic. It sets up a list for each statistic to store each instance of a goal, assist, man of the match and clean sheet instance. The method for this parent class will be used to obtain the total amount of objects in each list.

Class: goalList, assistList, manOfTheMatchList, cleanSheetList

Since these children classes will inherit the list attribute of the parent StatisticList class, these classes have no further attributes but do however contain methods to add to these lists. Each method of adding objects to these lists will differ from statistic to statistic.

Class: StatisticItem

This class is used to store add and retrieve information about each statistic. It serves as a parent class for the statistics goal, assist, clean sheet and man of the matches.

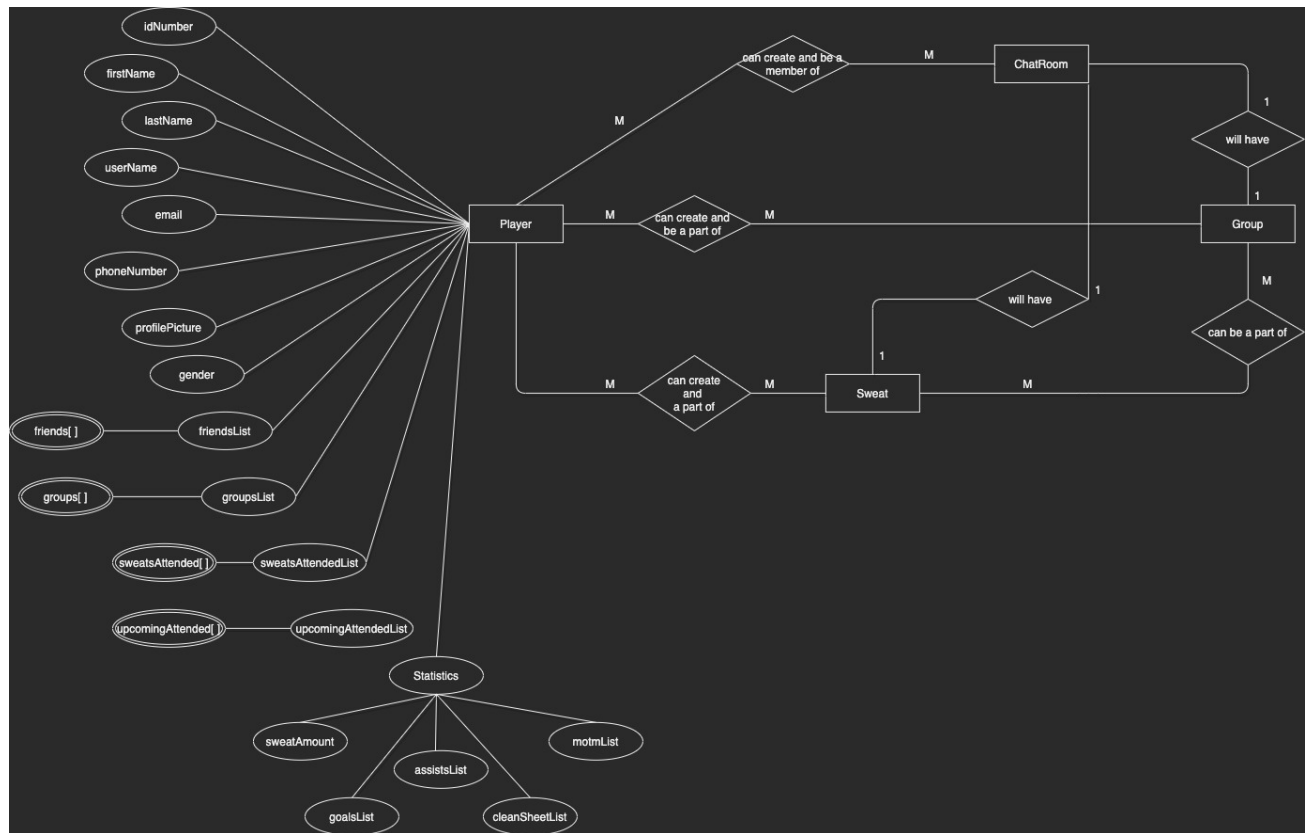
Attributes include the contributor, which is the player who the stat belongs to, the sweat which the statistic belongs to, and whether or not the statistic has received approval from the sweat survey. The methods are comprised of setter and getter functions for each of the attributes.

Class: Goal, Assist, Man of the Match, Clean Sheets

These classes represent the statistic itself and will make up the statistic list for its type.

The Man of the Match and Clean Sheet classes do not possess any unique attributes or methods, but the goal and assist classes do. The corresponding assister to the goal is the only attribute of the goal class, and the scorer of the assist is the only attribute of the assist class. They both contain setter and getter functions for their unique attributes.

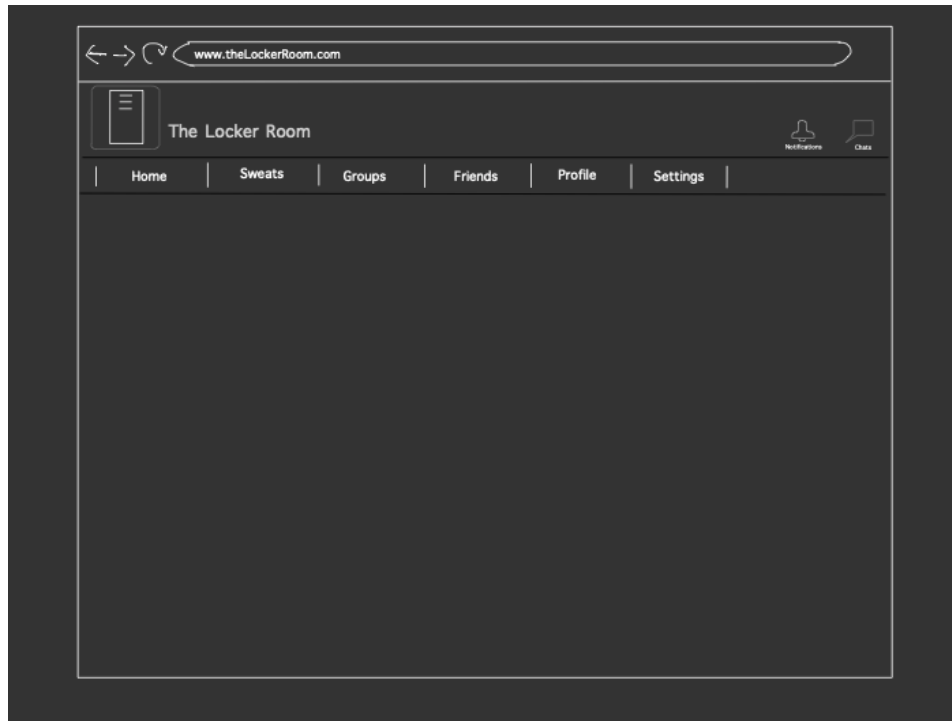
4. Entity Relationship Diagram



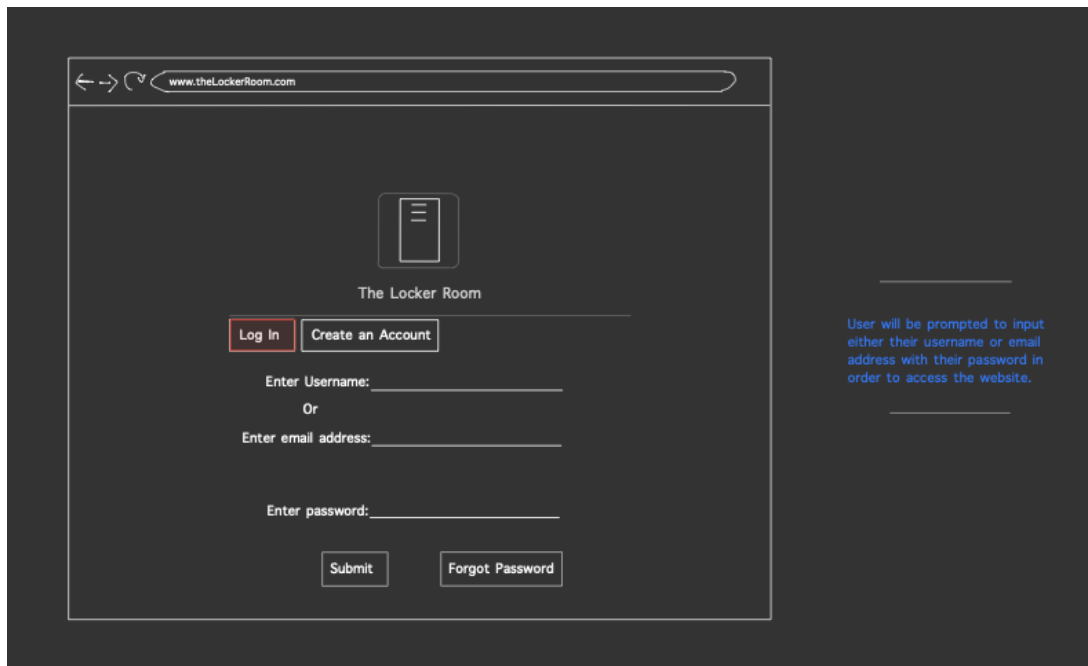
5. GUI Designs

Below are the sketches for the following pages:

1. Web Page Header and Navigation Bar



2. User Login Page



3. Create a New Account Page

The mockup shows a web browser window with the address bar displaying 'www.theLockerRoom.com'. The page features a logo for 'The Locker Room' and two buttons: 'Log In' and 'Create an Account' (highlighted with a red border). Below these buttons, a text prompt reads 'Please fill out the following:'. This is followed by five input fields: 'First Name:', 'Last Name:', 'Email Address:', 'Phone Number:', and 'Choose a Unique Username:'. A 'Create Account' button is positioned at the bottom of the form.

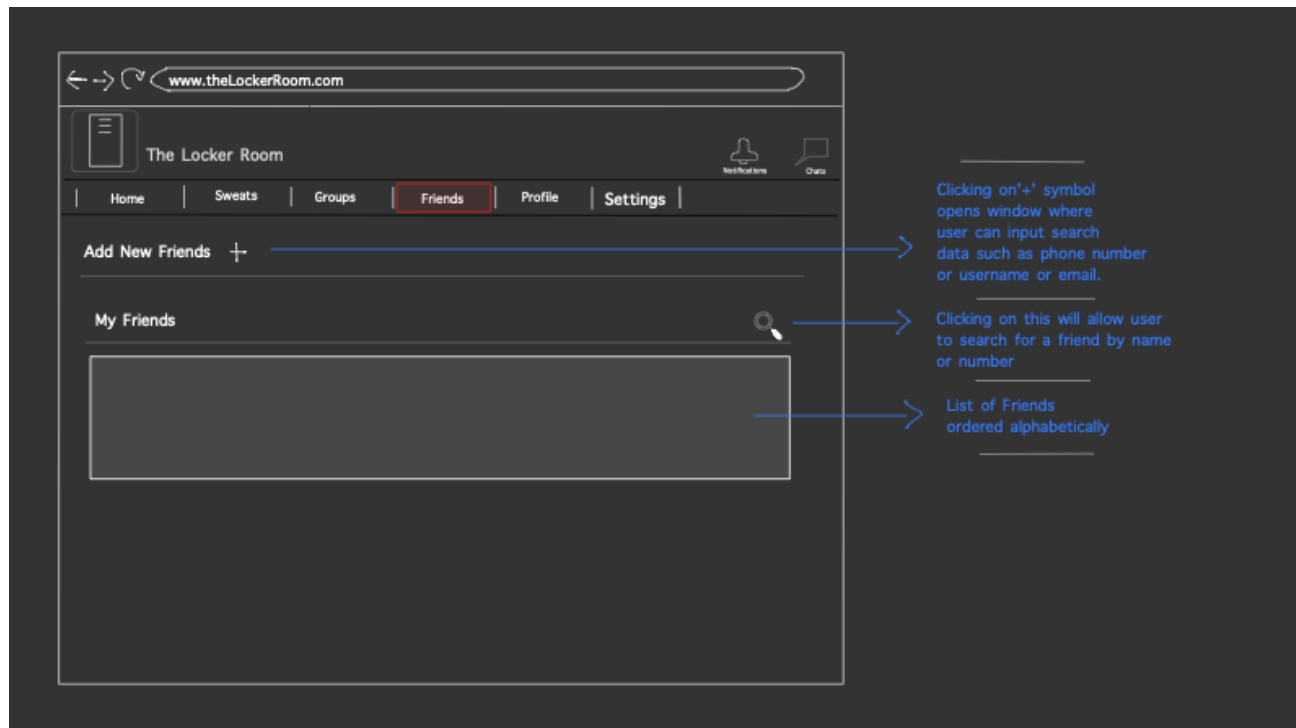
User will be asked to enter specified information. Information such as email, phone number and username will be compared to those already in use in database. If these already exist in database, user will be asked for different input

4. Home-Page

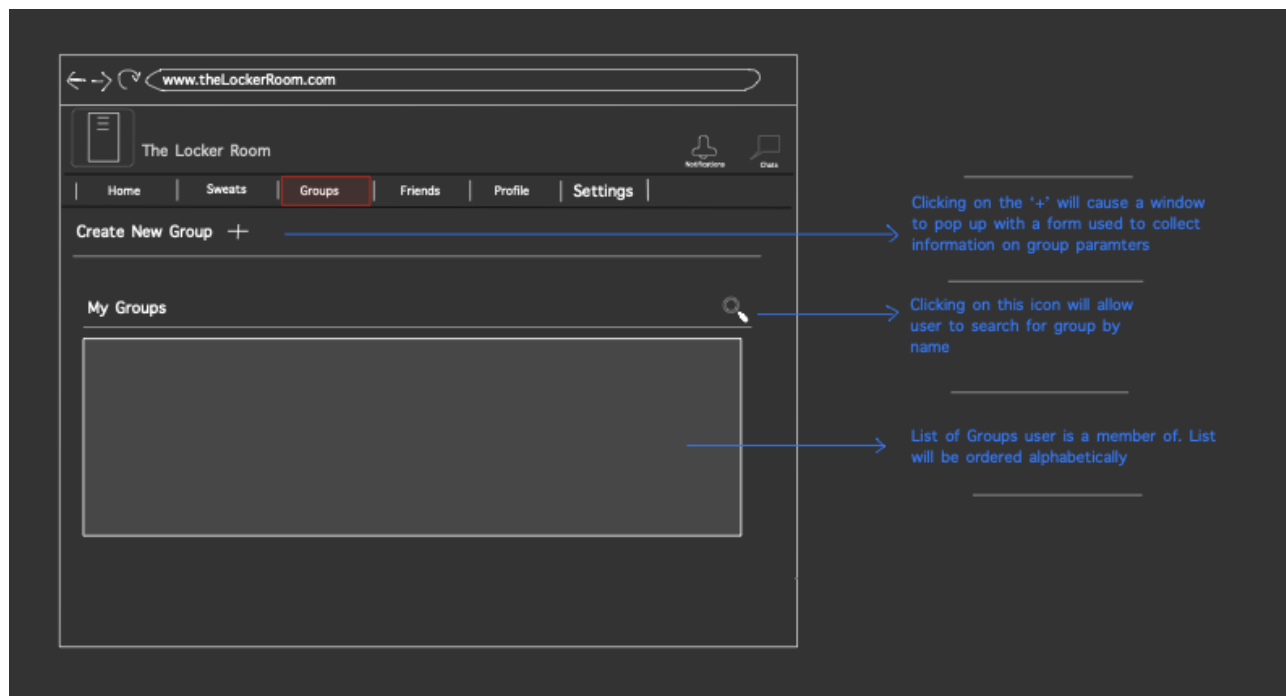
The mockup displays the 'Home' page of 'The Locker Room' application. The browser address bar shows 'www.theLockerRoom.com'. The page has a navigation bar with a hamburger menu icon, the app name 'The Locker Room', and icons for 'Notifications' and 'Chat'. Below the navigation bar is a horizontal menu with tabs: 'Home' (highlighted with a red border), 'Sweats', 'Groups', 'Friends', 'Profile', and 'Settings'. The main content area is divided into four sections, each with a title and a placeholder box: 'Upcoming Sweats', 'Pending Invites', 'Pending Requests', and 'Recent Sweats'. Blue lines connect the placeholder boxes for 'Upcoming Sweats', 'Pending Invites', 'Pending Requests', and 'Recent Sweats' to a handwritten note on the right that reads 'Lists of Relevant Data'.

Lists of Relevant Data

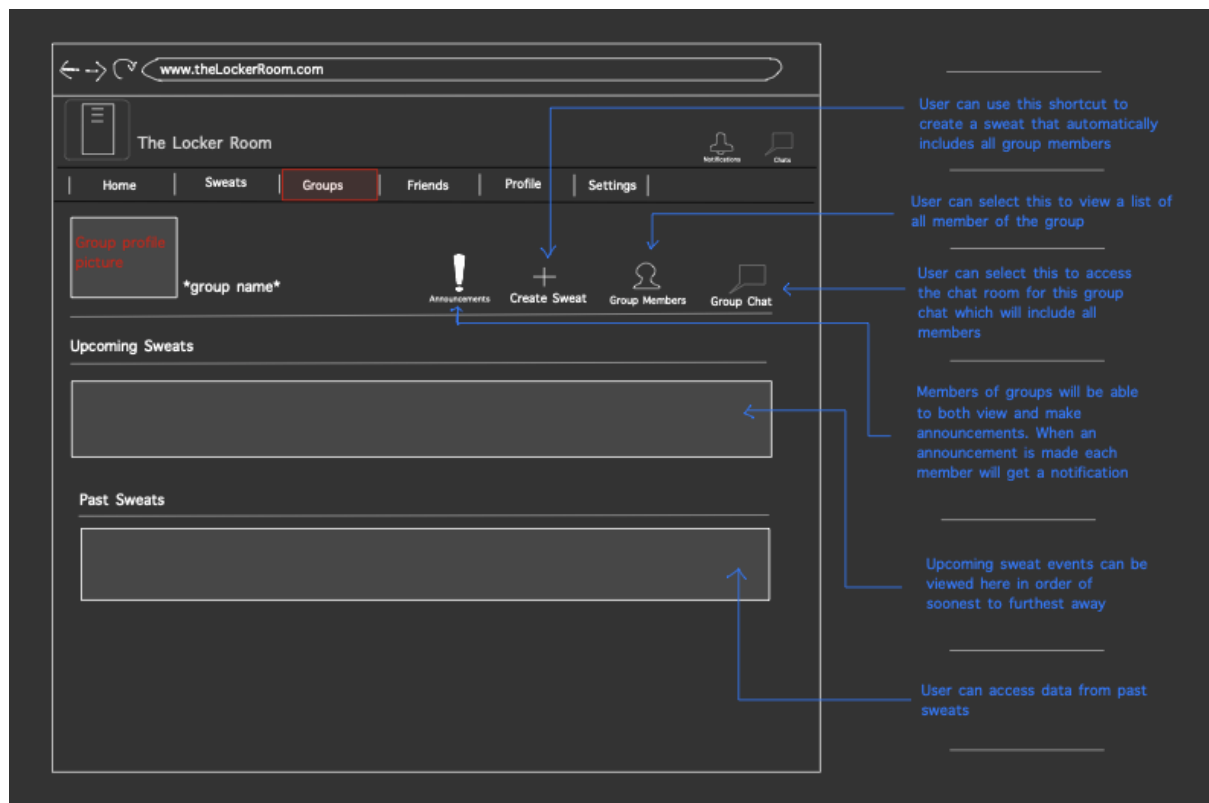
5. Friends Page



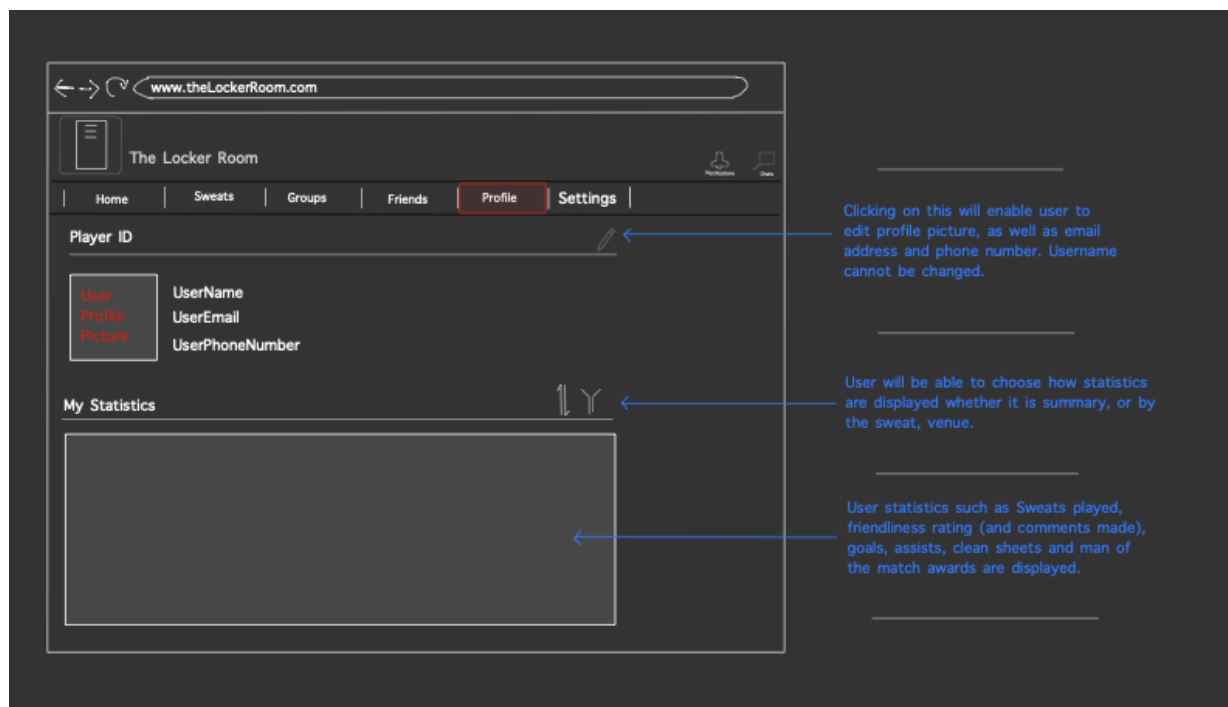
6. Group Hub Page



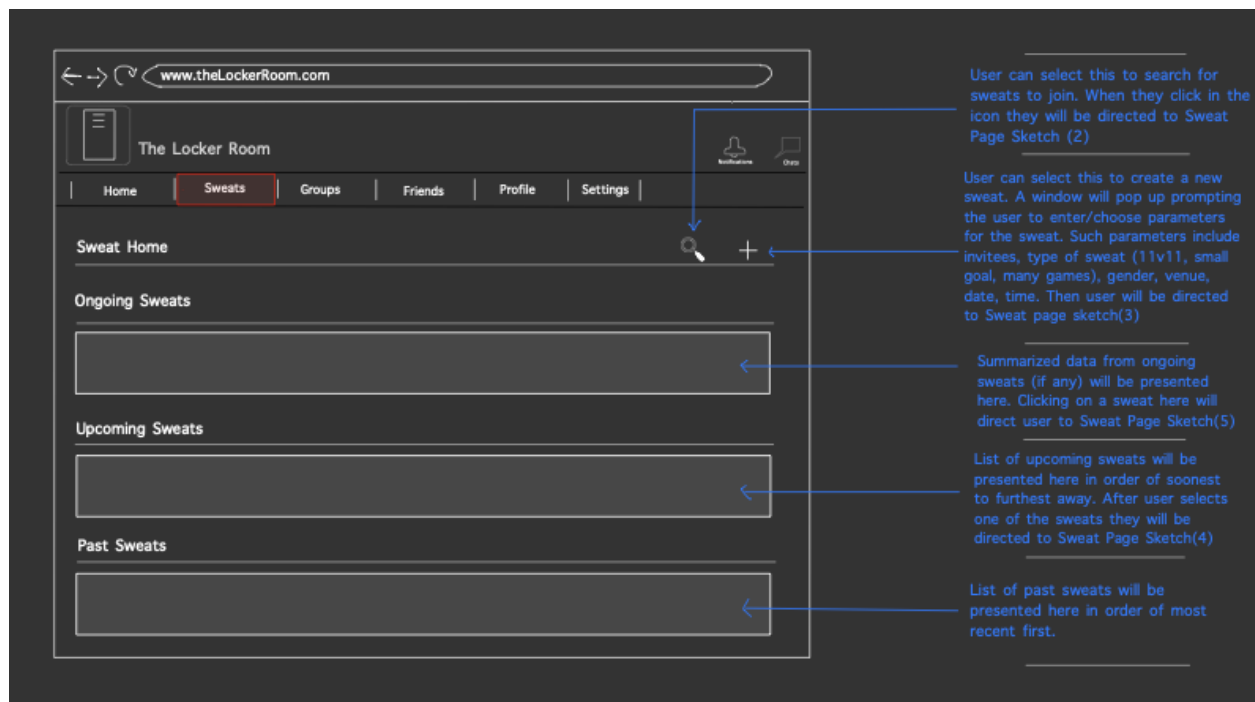
7. Group Home Page (After user has selected one of their groups)



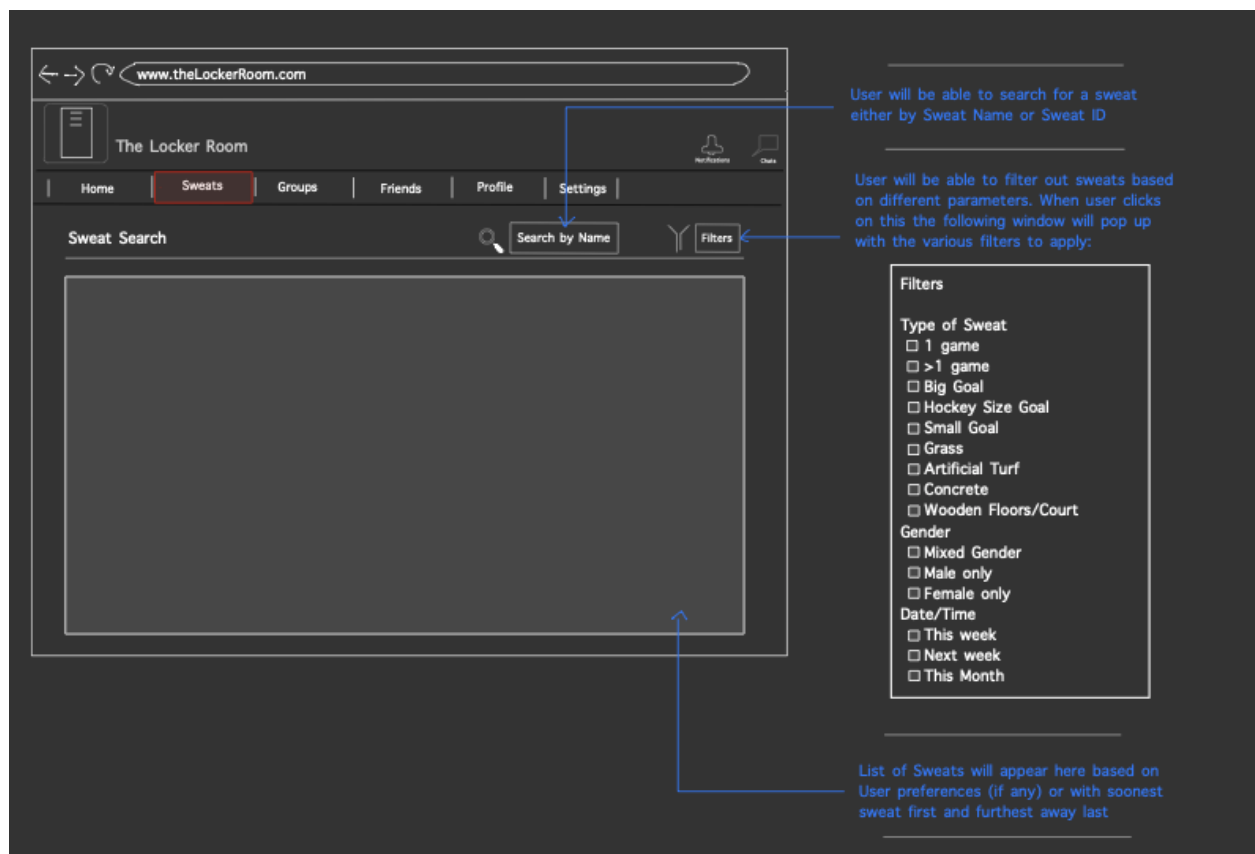
8. Profile Page



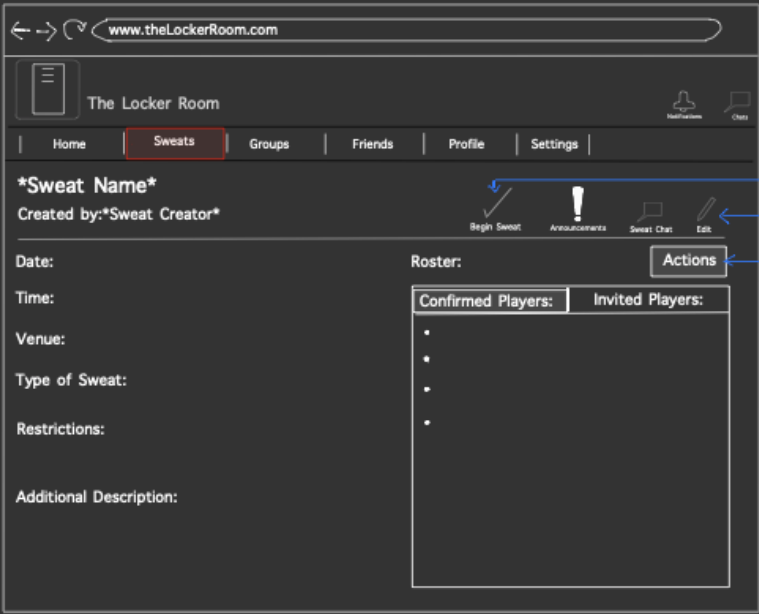
9. Sweat Page 1: Sweat Home



10. Sweat Page 2: Searching for a Sweat



11. Sweat Page 3: View of Pre Sweat Page as Admin/Creator of Page



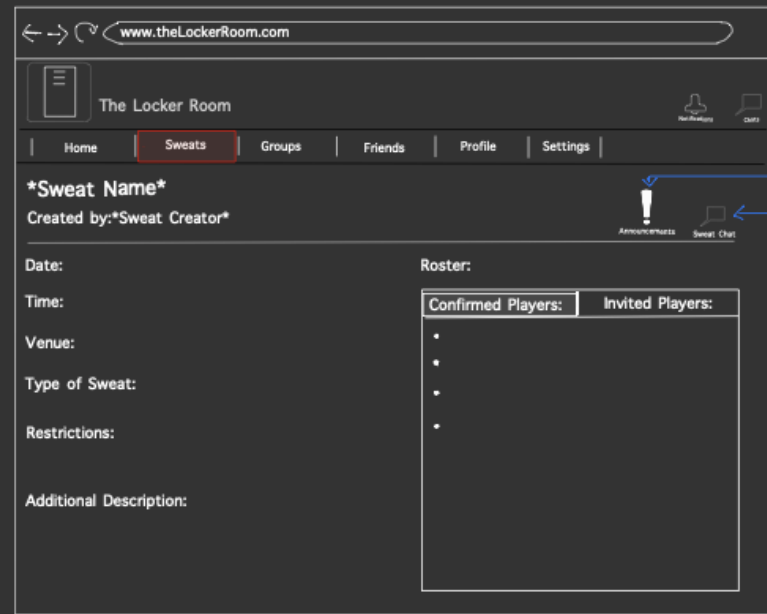
The screenshot shows the 'Pre Sweat Page' for an admin/creator. The page is titled '*Sweat Name*' and 'Created by: *Sweat Creator*'. It features a navigation bar with 'Home', 'Sweats', 'Groups', 'Friends', 'Profile', and 'Settings'. The 'Sweats' tab is selected. The page layout includes a form for creating a sweat with fields for Date, Time, Venue, Type of Sweat, Restrictions, and Additional Description. To the right of the form is a 'Roster' section with 'Confirmed Players' and 'Invited Players' lists. Above the roster is an 'Actions' button. The 'Actions' button has a dropdown menu with options: 'Begin Sweat', 'Announcements', 'Sweat Chat', and 'Edit'. The 'Begin Sweat' option is highlighted with a blue arrow. The 'Announcements' option is also highlighted with a blue arrow. The 'Sweat Chat' option is highlighted with a blue arrow. The 'Edit' option is highlighted with a blue arrow. The 'Actions' button is highlighted with a blue arrow.

Admins can select this to begin the sweat live session or to begin inputting events such as Goals, assists, substitutions. When this is selected, user will be directed to Sweat Page(5)

Admins can select this to edit data such as sweat name, Date, Time, Venue, Type of Sweat, Restrictions, additional description and privacy of sweat (private or public)

Admins can select this to alter the roster or players involved in the sweat. They will be able to invite and remove players, as well as change the status of the sweat (closed or open to requests). Additional functions include: sorting confirmed players into teams and making other participants admins

12. Sweat Page 4: View of Pre Sweat Page as Normal, non-admin user



The screenshot shows the 'Pre Sweat Page' for a normal, non-admin user. The page is titled '*Sweat Name*' and 'Created by: *Sweat Creator*'. It features a navigation bar with 'Home', 'Sweats', 'Groups', 'Friends', 'Profile', and 'Settings'. The 'Sweats' tab is selected. The page layout includes a form for creating a sweat with fields for Date, Time, Venue, Type of Sweat, Restrictions, and Additional Description. To the right of the form is a 'Roster' section with 'Confirmed Players' and 'Invited Players' lists. Above the roster is an 'Announcements' button and a 'Sweat Chat' button. The 'Announcements' button is highlighted with a blue arrow. The 'Sweat Chat' button is highlighted with a blue arrow.

User can access the sweat announcements to both view and make announcements

User can access the sweat group chat via this shortcut.

13. Sweat Page 5: View of Real - Time Sweat Event

For sweats that contain more than one game, admin can create a new game before or after ending current game. If admin is entering data after the sweat procedure will remain the same

If admins are entering data in real time during the sweat, they can use the stopwatch to keep track of the time elapsed

Admins can add/remove events such as goals, assists, substitutions, yellow and red cards.

Logged data will be displayed here

14. Sweat Page 6: View of Post Sweat Page and Survey

Users can access the sweat Survey here. The survey will contain a mixture of question types such as multiple choice and ranking. The survey is used to validate the logged events of each of the games by asking users to confirm or deny whether or not an event has occurred. If a user does not think a particular event has occurred, they will be asked to provide an alternative event which will then be subject to a vote bconfirmation or denial by participants.

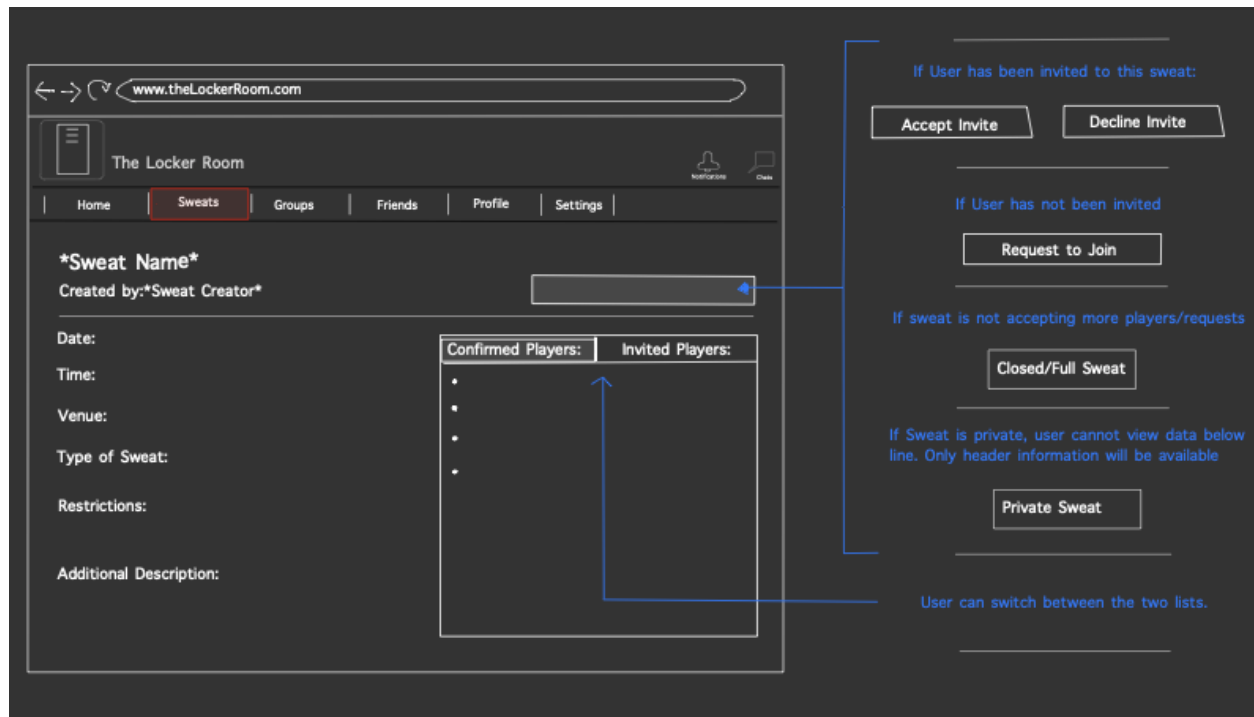
Shows how many participants have completed the survey. If at least 75% of participants do not complete the survey, the events of the sweat will be nullified and events will not be added to players' profiles.

Shows the time left to complete the survey.

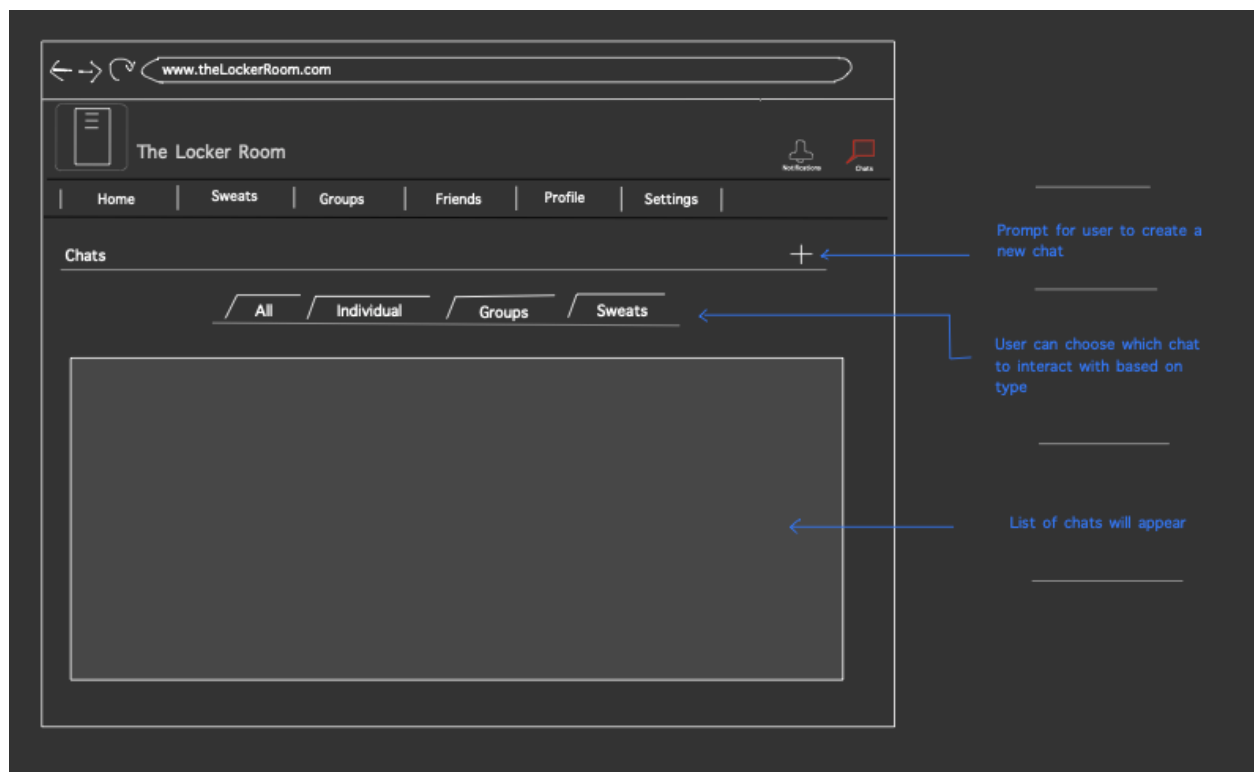
Shows summarized events which are subject to the survey.

Shows top scorer, top assist provider, most clean sheets, and man of the sweat, which will be voted for by all participants in the survey

15. Sweat Summary as viewed by user interested in joining



16. Chats Page



17. Notifications Page

