Sweat Rivals

Group 11

Caden Lalley, George Viveros-Zavaleta, Harrison Whetzel, Michael Daramola, Alexander Griffin

Project Overview

Most people when working out will use some kind of fitness app that will track calories consumed as well calories burned. Fitness apps are a great way to keep track of your fitness progress as well as compete with friends through those numbers. While most fitness apps have friends and such, none are focused on that competitive side. Our application will fill that void by focusing on competition between friends using point based challenges. These challenges can range from weight and calorie loss to actual exercises. By completing these challenges you can compete with your friends as well as groups that can be made. We want to design an application that will help the average person get more excited about exercising as well as helping better their health.

Architectural Overview

We are developing a web application that follows the MVC architecture. We found that this architecture suited the needs of our application and the natural design pattern of Express.js. Within the MVC design pattern, we chose to add a middleware module that allows for quick development and implementation of middlewares as we see fit. For our view we used javascript and ejs.

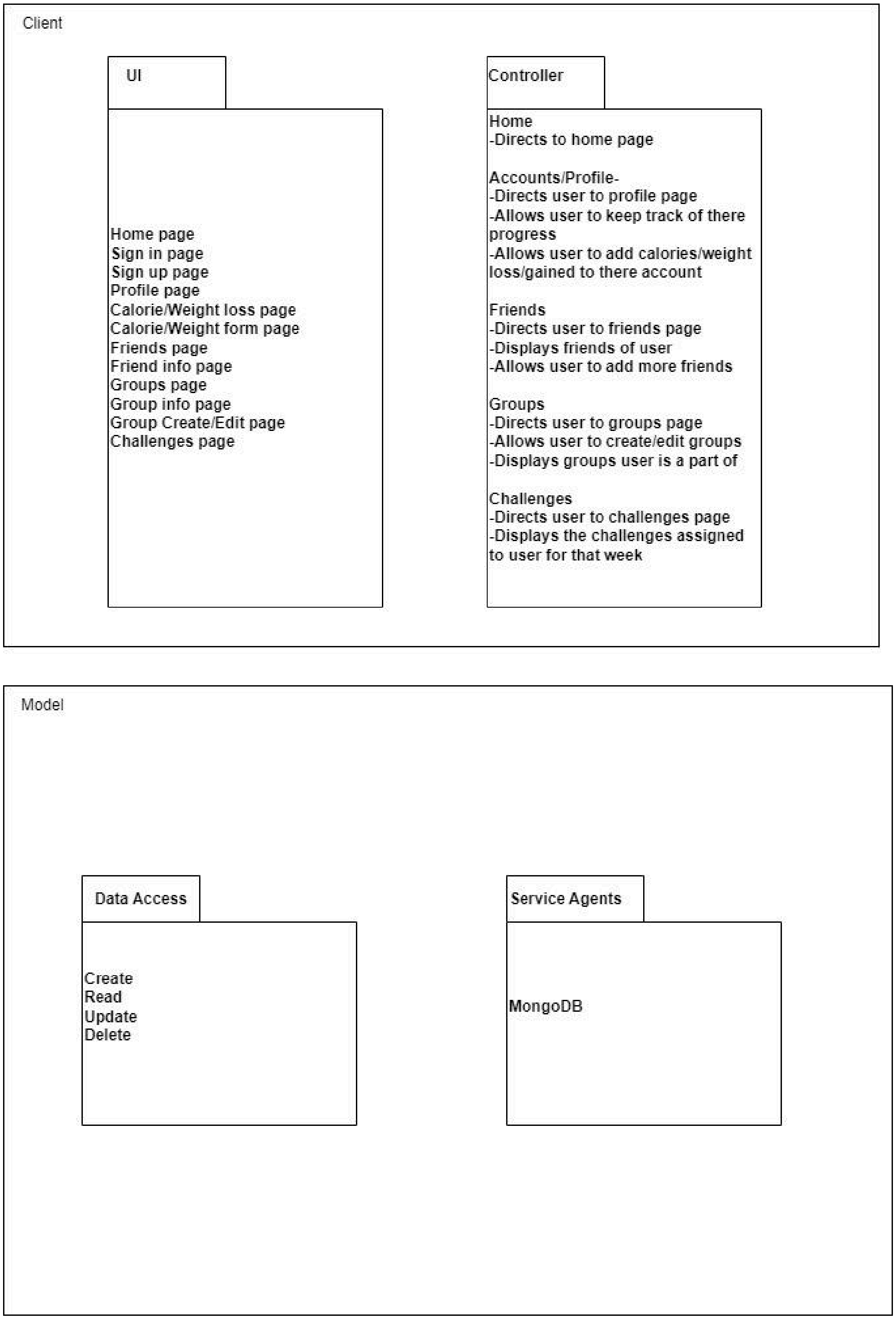
Subsystem Architecture

It has two layers, view and server

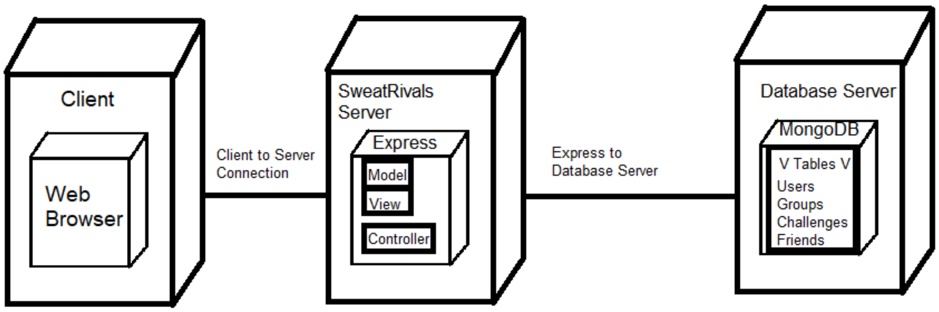
- Client
  - Our client is mainly composed of multiple web pages, each having a different purpose
- Server
  - The server consists of four modules:
    - Model

- Controller
- Routes
- Middlewares

These modules are used to dynamically send views to the client with data that is stored in our persistent DB. The model, controller and middleware modules all interact with the requests made by the client to dynamically display different pieces of data.


Client
UI
Home page
Sign in page
Sign up page
Profile page
Calorie/Weight loss page
Calorie/Weight form page
Friends page
Friend info page
Groups page
Group info page
Group Create/Edit page
Challenges page
Controller
Home
-Directs to home page
Accounts/Profile-
-Directs user to profile page
-Allows user to keep track of there progress
-Allows user to add calories/weight loss/gained to there account
Friends
-Directs user to friends page
-Displays friends of user
-Allows user to add more friends
Groups
-Directs user to groups page
-Allows user to create/edit groups
-Displays groups user is a part of
Challenges
-Directs user to challenges page
-Displays the challenges assigned to user for that week
Model
Data Access
Create
Read
Update
Delete
Service Agents
MongoDB

Deployment Architecture



The client uses the HTTPS on their browser to connect to the Undesired server over the internet. Express is used to host the Undesired server as well as the models, views, and controllers. All of this is then connected to the database server using MongoDB.

Global Control Flow

SweatRivals is user-driven and focuses on friendly fitness competition, meaning the user's actions and choices determine their app experience. Before signing in, users can view the marketing homepage highlighting key features: tracking health metrics, competing with friends, and earning points through challenges. Users create an account providing basic information like email and password. Once the account is created they land on the homepage which will allow them to view recent activity, current fitness goals, friend updates, and leaderboard rankings - from here they can track calorie intake, log workouts, update weight, and view point totals. Users can search for friends, send/accept requests, view each other's progress and stats creating a social accountability loop, with the friends feed showcasing recent accomplishments. A central feature is weekly rotating challenges which incentivizes users to go the extra mile through friendly competition, with completion earning points contributing to group leaderboard rankings. The app allows users to set personal fitness goals like losing a set amount of weight or running a marathon. When each goal is reached users will earn points giving them a sense of accomplishment as points can be seen by users that view another user's profile.

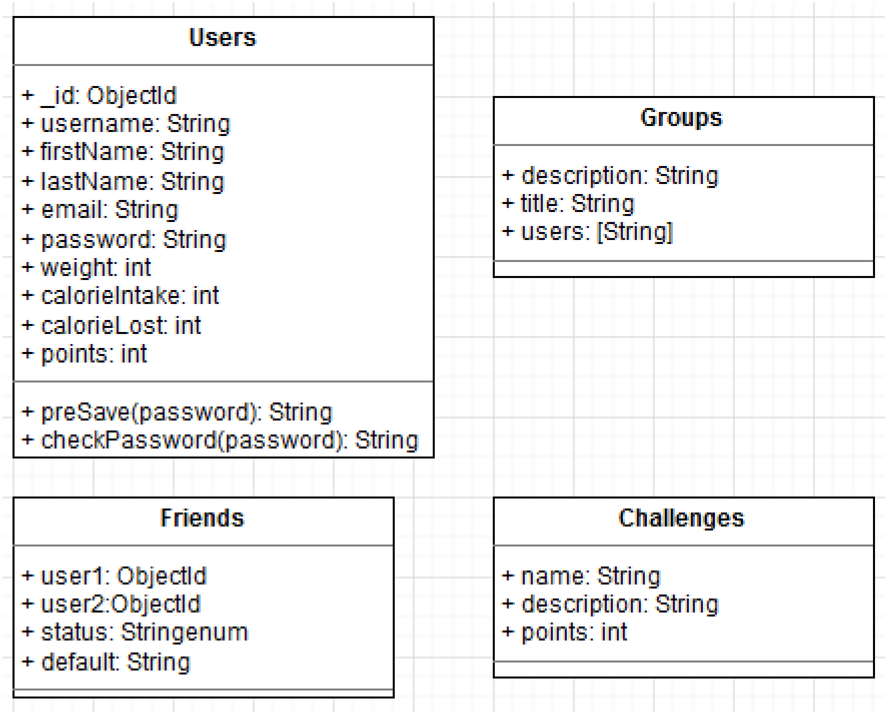
Persistent Data Storage

We will be storing all our information, user accounts, challenges, friends, and groups in a mongoDB database. All of that information will usually be added to the database through the use of forms except for challenges which we will manually make in the database. We’ve made four tables. The first table which is for the user accounts has username, first name, last name, email, password, friends, weight, calorie intake, calorie loss, and total points. The second table that's for challenges has the name of the challenge, its description, amount of points it's worth, and what you need to complete the challenge. The third table that's for friends has user1, user2, and status. And finally the fourth table that is for groups has title, description, and array of usernames

Detailed System Design

For our model we used the Node.js and express framework, as well as ejs, javascript, and CSS. We used mongoDB as our database and used mongoose to populate it. Our web page views were created with ejs, and were formatted with a mix of bootstrap and css. Any kind of information needed from the database is displayed by getting that information in the controller, sending it through the route and using javascript in the ejs.

Static View



Users

+ _id: ObjectId
+ username: String
+ firstName: String
+ lastName: String
+ email: String
+ password: String
+ weight: int
+ calorieIntake: int
+ calorieLost: int
+ points: int

+ preSave(password): String
+ checkPassword(password): String

Groups

+ description: String
+ title: String
+ users: [String]

Friends

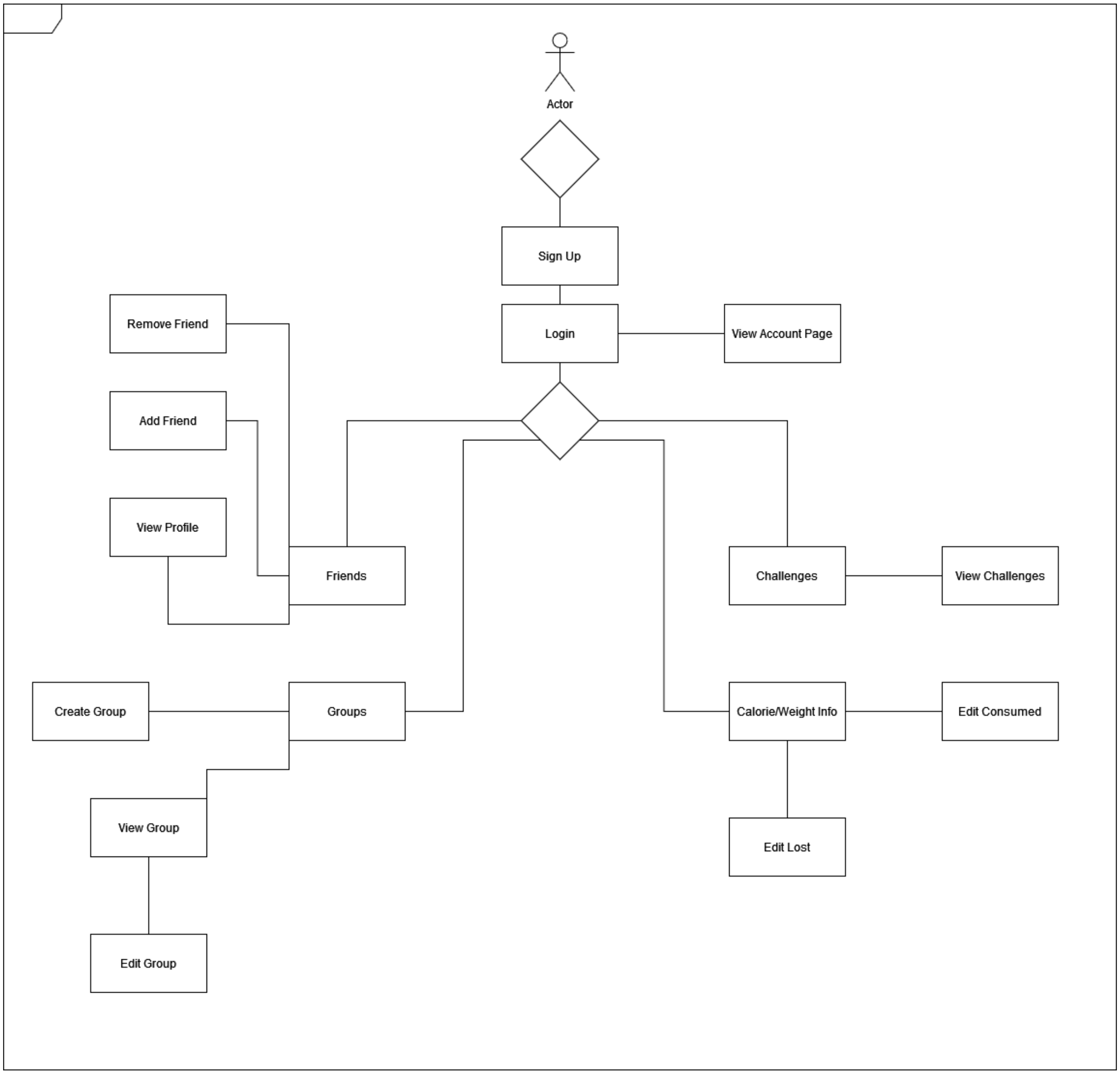
+ user1: ObjectId
+ user2:ObjectId
+ status: Stringenum
+ default: String

Challenges

+ name: String
+ description: String
+ points: int

Our User table holds the users information: username, first name, last name, email, password, their weight, their calorieIntake and calorieLost, as well as their total points earned. It also has a friends array that saves all the friends that the user has. Each value in the friends array has what's in the Friends table. The table holds the id of user1 and user2, the status of the friendship and the default status. The groups table has the title or name of the group, the description, and an array of the different users. The Challenges table has the name of the challenges, the description, as well as the amount of points it's worth. It also has what you need to complete it but we aren't sure if we need multiple different value storages(caloriesLost, caloriesIntake, weight loss) for the different types of challenges. So we did not include that yet.

Dynamic View



Signing up gives you the ability to login. Logging in allows you to view your account page. Additionally, you are able to view friends, challenges, groups, or calories weight/info. Navigating to the friends tab allows you to view the profiles of yourself and your friends, add friends or remove friends. The groups tab allows you to create groups and view groups. In the view group screen, you’re able to edit groups that you have the ability to. Under challenges, the only manual option available to the user is to view challenges and their progress on said challenges. In calorie/weight info, you’re able to edit your consumed and lost calories.