

**Team Information:**

Project Title: Swole Goals

Group number: Burgundy

Members (Name | Email | GitHub):

Michael Lawrence | [mdlawrence@utexas.edu](mailto:mdlawrence@utexas.edu) | [prilak](#)

Nicole Finks | [finksnicole@gmail.com](mailto:finksnicole@gmail.com) | [nfinks](#)

Kaibo Cai | [kaibocai@utexas.edu](mailto:kaibocai@utexas.edu) | [KaiboCai](#)

Nick Pattie | [nick.pattie@utexas.edu](mailto:nick.pattie@utexas.edu) | [NickPattie](#)

Vivian Nguyen | [vivian.m.nguyen@utexas.edu](mailto:vivian.m.nguyen@utexas.edu) | [vivianmnguyen](#)

Rohan Koripalli | [rkoripalli@utexas.edu](mailto:rkoripalli@utexas.edu) | [rkoripalli](#)

**URLs:** Github: <https://github.com/vivianmnguyen/SwoleGoals>

Google Docs: <https://drive.google.com/drive/u/1/folders/0AlbWej0ljZcQUk9PVA>

**Vision:**

Swole Goals is a competitive fitness web app that allows users to compete with their friends in different fitness challenges. The goal for the application is to get people to become more physically fit by working out consistently. We will facilitate this goal by having our user workout and compete with their friends. When the user sets up the app for the first time, they will create a group, and a set of exercises that they would like the group to compete in. Each exercise will measure performance by either time, repetitions, and weight. After completing setup, a Challenge Map is generated based on the number of exercises our group of users will be completing each day, with each tile representing one exercise. As the user completes exercises they will move forward on the map with the goal being to complete the map before your friends. The user will also be able to see their progress on exercises, friends progress, and current rankings on a Leaderboards page.

Points will be rewarded based on whether the user got to the end of the map, improvement on each exercise, and overall performance. At the end of the month final rankings will be released, the user will have the option to change what exercises they would like to do for the next month, and the Challenge Map will reset. Swole Goals fills the market for friends who want to exercise and enjoy competing with each other. The application solves the problem of being motivated to workout, because competing with friends on a daily basis provides that motivation.

We recognize that there are many fitness apps out there such as InKin, FitBit, and Strava. We've found that these apps are focused mainly on calories burned, steps walked, and distance travelled, but don't have the ability for users compete on things like push ups, weight lifted, or mile pace. They also don't focus on the group competition on the level that our app hopes to achieve with our Challenge Map and ranking system. None of the apps currently on the market allow for a comprehensive, gamified experience like Swole Goals.

Phase	Overview	Details
1	Server and Database Initialization, Begin Scraping Exercises	<ul style="list-style-type: none"> <li>• Create a plain text page for game board, user profile, and exercise list.</li> <li>• Add about information from Github to the web page</li> <li>• Provide 4 user stories and add them to the issue boards</li> <li>• Host the server on Google App Engine</li> <li>• Get all exercises from Bodybuilding.com</li> <li>• Learn and use BootStrap for frontend</li> <li>• Create unit tests for data source scraping</li> <li>• Learn HTML, SQL, JavaScript, and Angular</li> <li>• Learn Python and Selenium for scraping</li> </ul> <p>Deliverables: Begin MySQL database initialization with exercises</p>
2	Account Creation, Retrieve Formatted Data from Data Sources, Challenge Selection	<ul style="list-style-type: none"> <li>• Create User Profile with database support</li> <li>• Create Account Creation Page with database support</li> <li>• Adding friends functionality</li> <li>• Devise game points algorithm</li> <li>• Devise exercise rankings</li> <li>• Create exercise selection and challenge creation menu</li> <li>• Finish exercise scraping with strengthlevel.com</li> <li>• Create Exercise Filters</li> <li>• Learn and use jQuery for frontend</li> </ul> <p>Deliverables: Write point-system algorithms to compare user's progress. UI for user workout selection &amp; user profile</p>
3	Leaderboard, Game Map, Calculate Game Statistics and User Rankings, and improved UI elements	<ul style="list-style-type: none"> <li>• Complete unit tests of RESTful API using Postman</li> <li>• Use Selenium to create functional tests</li> <li>• Create game board page</li> <li>• Create current exercise page</li> <li>• Create leaderboard to display rankings</li> <li>• Create leaderboard filters</li> <li>• Application is fully functional at the end of this phase</li> <li>• Provide 4 more user stories</li> <li>• Create unit tests for database with node-unit</li> </ul> <p>Deliverables: Complete frontend game map and leaderboards, Complete web application functionality</p>
4	Refactoring & Applying Design Pattern	<ul style="list-style-type: none"> <li>• Apply at least 2 refactorings &amp; at least 2 design patterns</li> <li>• Complete additional testing on the new design patterns.</li> </ul> <p>Deliverables: Final refactored code</p>

## Data Sources, Scraping, and Database:

We will scrape exercise information such as muscle groups, difficulty, equipment, and images from [bodybuilding.com/exercises/](http://bodybuilding.com/exercises/). From there, we will then scrape [strengthlevel.com](http://strengthlevel.com) by entering in various weights and ages to obtain percentiles for all the available exercises then store them into the database. All scraped data will be gathered with Python and Selenium. We will also store user information such as age, weight, and gender, and exercise performance. All this will be stored in a mySQL database, where we will have four main tables for storing our data: Exercises, User Information, User Performance, and Game Statistics.

## Requirements:

### User creates account:

As a new user I want to create an account so that I can utilize the web app.

### User adds/removes exercise to group challenge:

As a user I want to add/remove an exercise from the exercise database to my group challenge.

### User views leaderboards:

As a user I want to view the leaderboard for my group so I can compare myself to my friends.

### User logs in / out:

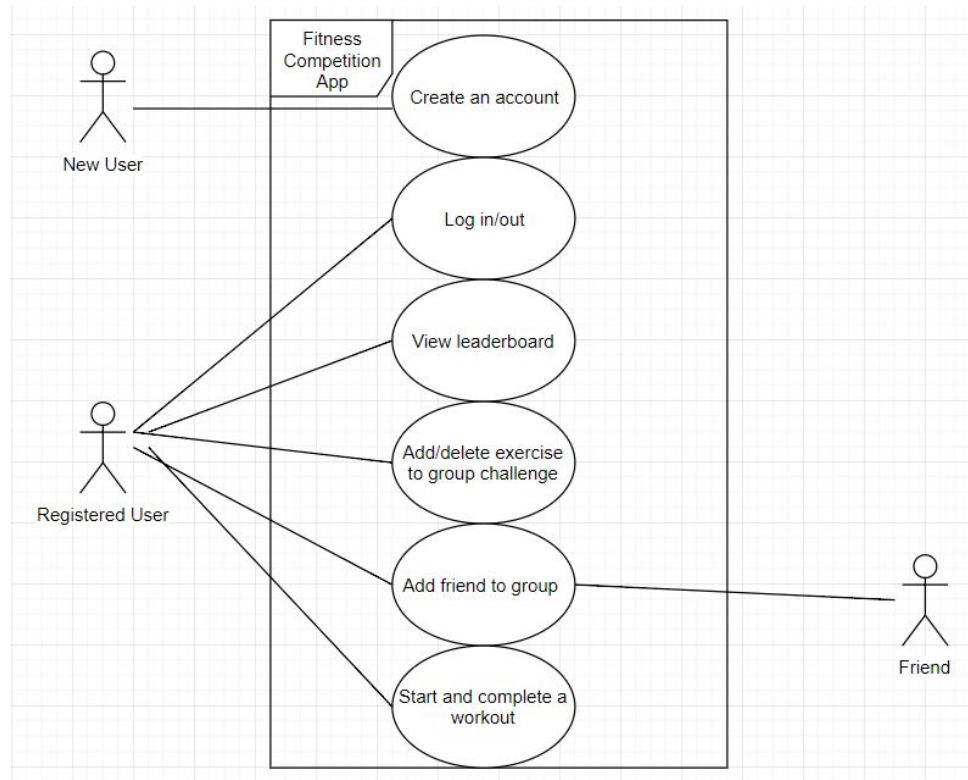
As an existing user I want to log in/out my account so I can open/close the web app.

### User adds friend to group:

As a group creator I want to add my friend to a group so I can include him/her in the workout.

### User starts and ends a workout:

As a user I want to start a workout and mark it complete once I have finished it.



### **Formal Use Cases:**

Project: Swole Goals

Use Case: User adds exercise to group challenge

Stakeholders and Interests:

User: wants to add an exercise to his/her group challenge

Primary Actor: User

Goal: User adds a specific exercise to his/her group challenge.

Success end condition: Exercise successfully added.

Failure end condition: Challenge is not successfully added.

Precondition: User is logged into Swole Goals

Trigger: User indicates desire to add an exercise to his/her group challenge.

1. User indicates desire to add an exercise the challenge.
2. User searches for the exercise type
3. User indicates that the conditions are set correctly and the exercise is ready to be added
4. Swole Goals adds the exercise information to the user's group challenge.
5. Swole Goals informs the user of the successful upload.

Extensions:

1a. The exercise search feature is not available.

1a1. Swole Goals informs the user that the feature is not currently available.

2a. The user does not enter specific necessary information for the exercise (workout type, specific data for the workout)

2a1. Swole Goals notifies the user that there is missing information that must be filled.

4a. Database search is slow

4a1. Swole Goals notifies the user of the slowdown, gives option to cancel the search.

4b. Database connection failure.

4b1. User is notified of the failure and is prompted to try again from step 2.

Goal: User wants to view group leaderboard

Primary Actor: User

Precondition: User is logged in

Success end condition: User is able to view leaderboard

Failure end condition: User cannot view leaderboard

Trigger: User navigates to Challenges page

1. User clicks on challenges page
2. User selects which leaderboard they want to view
3. User clicks on leaderboard to view detailed information and progress

Extensions:

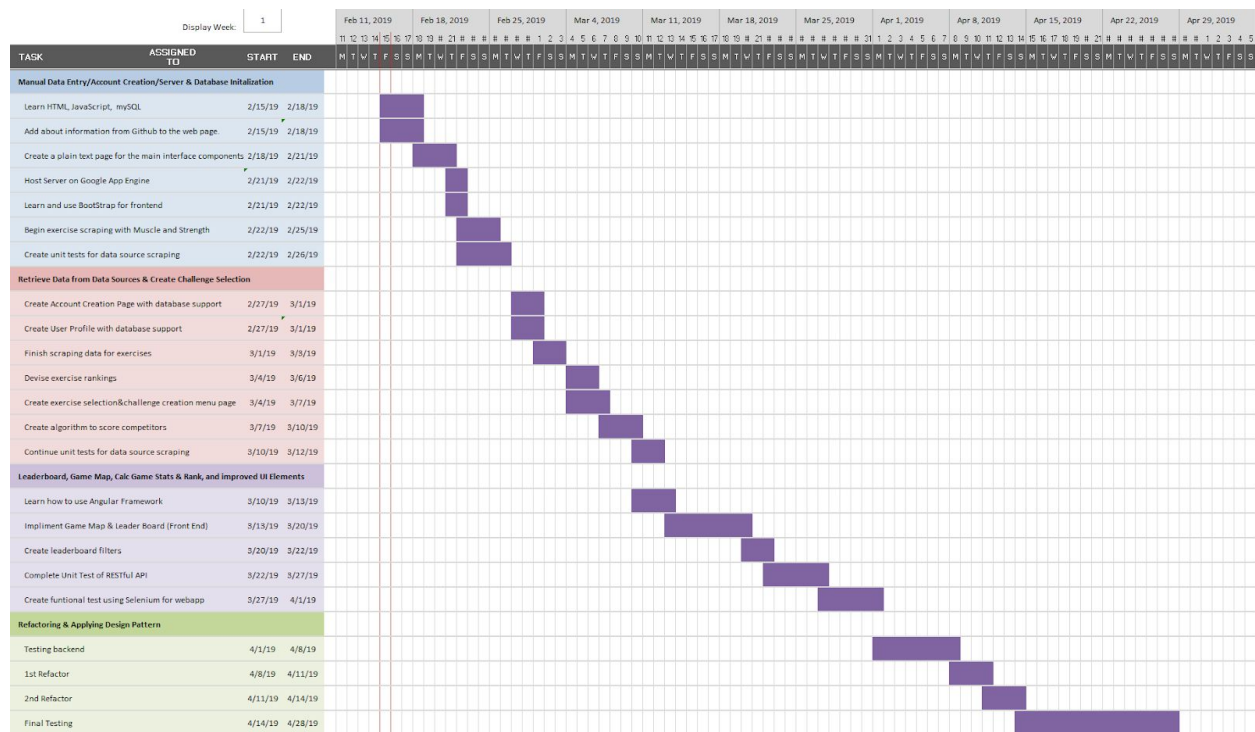
1a. Challenges page is slow to load.

1a1. User is informed of slow load time.

2a. Leaderboard is slow to load.

2a1. User is informed of slow load time.

## Planning and Scheduling:



Link to full size: [https://1drv.ms/x/s!AmjcbJSTgrn76QxaDq6yWQR\\_hPpE](https://1drv.ms/x/s!AmjcbJSTgrn76QxaDq6yWQR_hPpE)

## Tools, software, frameworks:

We will use Balsamiq to design the User Interface. We will use Angular, Bootstrap and jQuery for the frontend, and Node.js and MySQL for the backend. We will use text editors like Sublime Text and Visual Studio Code to write the code, and collaborate using Git with a Github repository. We will use Google Chrome and Mozilla Firefox for frontend debugging, and MySQL workbench to setup the database. For testing our JavaScript code, we will use Mocha, a JavaScript test framework. We will complete unit tests of RESTful API using Postman. We will also use Selenium to create functional tests for our web application.

## Feasibility:

For the leaderboards to be accurate, we need a lot of users to normalize the results. It is also crucial that users consistently and accurately input their workouts. If we are unable to successfully use the APIs and web-scrape the websites we have mentioned above, we will have difficulty obtaining information necessary to generate accurate rankings for users. If we are unable to pull the data from our database, we will be unable to create the leadership board/game board. We might not have the time to develop the gameboard.

## Interface:

