Physio: An Elite Gym Experience

**Table of Contents**

**1.1 Overview**

**1.2 Goals**

**1.3 Scope**

**1.4 Definitions**

**1.5 Assumptions**

**2.0 General Design**

**2.1 Environment**

**2.2 User Characteristics**

**2.3 Constraints**

**3.0 Diagrams**

**3.1 Class Diagram**

**3.2 UML Diagram**

**3.3 Test Cases**

**Prepared by:**

Brett Parker, Christian McDermott, Dante DiClemente

**1 Introduction**

**1.1 Overview**

Our hope is that with this project is to make a place where people who are adamant about going to the gym. We hope to streamline the process for finding a gym partner as well as finding a personal trainer no matter where they are. We are doing this in a tinder-esk fashion where we can pair people based off of likenesses as well as physical capabilities, and free time.

**1.2 Goals and Objectives**

The goal of this project is to optimize the experience of anyone who goes to the gym. This is done by allowing people to not only find gym partners that are similar in fitness levels, but also similar in personalities and abilities. In addition to these features, there are areas for documentation of routines to share workouts for people that are matched with you to try and use. With the capability of finding trainers streamlined, it has never been easier to find a coach for people of any abilities. There are descriptions and background checks that will allow people who apply to be posted as trainers to become verified.

**1.3 Scope**

Our final product will allow users to swipe on possible gym partners, similar to Tinder. This will give people the ability to find people anywhere and will take away the awkwardness of sending people a message over other social media platforms. From the final, we hope to allow trainers to become more easily accessible as well as help people track their fitness and share their progress with people that they are matched with. We are hoping to implement companion iOS and Android applications to our web app.

**1.4 Definitions**

iOS is iphone servers

MongoDB is the backend environment

Angular.io is the front end environment

**1.5 Document Conventions**

The incomplete parts will be marked TBD. We will be referring to the web app as Physio™.

**1.6 Assumptions**

It is assumed that all users are using the application under their real name and no alias. For this program to work, the user must have a connection to the internet and be using any mainstream browsers (Chrome, Firefox, Safari, etc).

**2.0 General Design Constraints**

**2.1 Product Environment**

We will be using MongoDB, Express, Node.js, and angular.io to develop our web app. We will use these programs in conjunction with each other to put together a fine-tuned application. For the time being, the servers will be hosted locally and only allow local users while we are in testing. There will be a complete graphical user interface provided by angular.io.

**2.2 User Characteristics**

The users will be common gym goers, personal trainers, and administrators (admins). The admins will only be the developers of Physio™ and they will have access to every part of the front end and back end code as well as the management of all accounts. The common user will have access to posts on their wall, basic account management, access to make new matches, and the ability to request a trainer. The trainers will have all permissions inherited from the common user with a few added bonuses after they're verified. They will be able to make posts (with given permission from the account owner) of progress of a certain user. This will allow people to show verifiable progress and will show commitment for other users to acknowledge.

**2.3 Mandated Constraints**

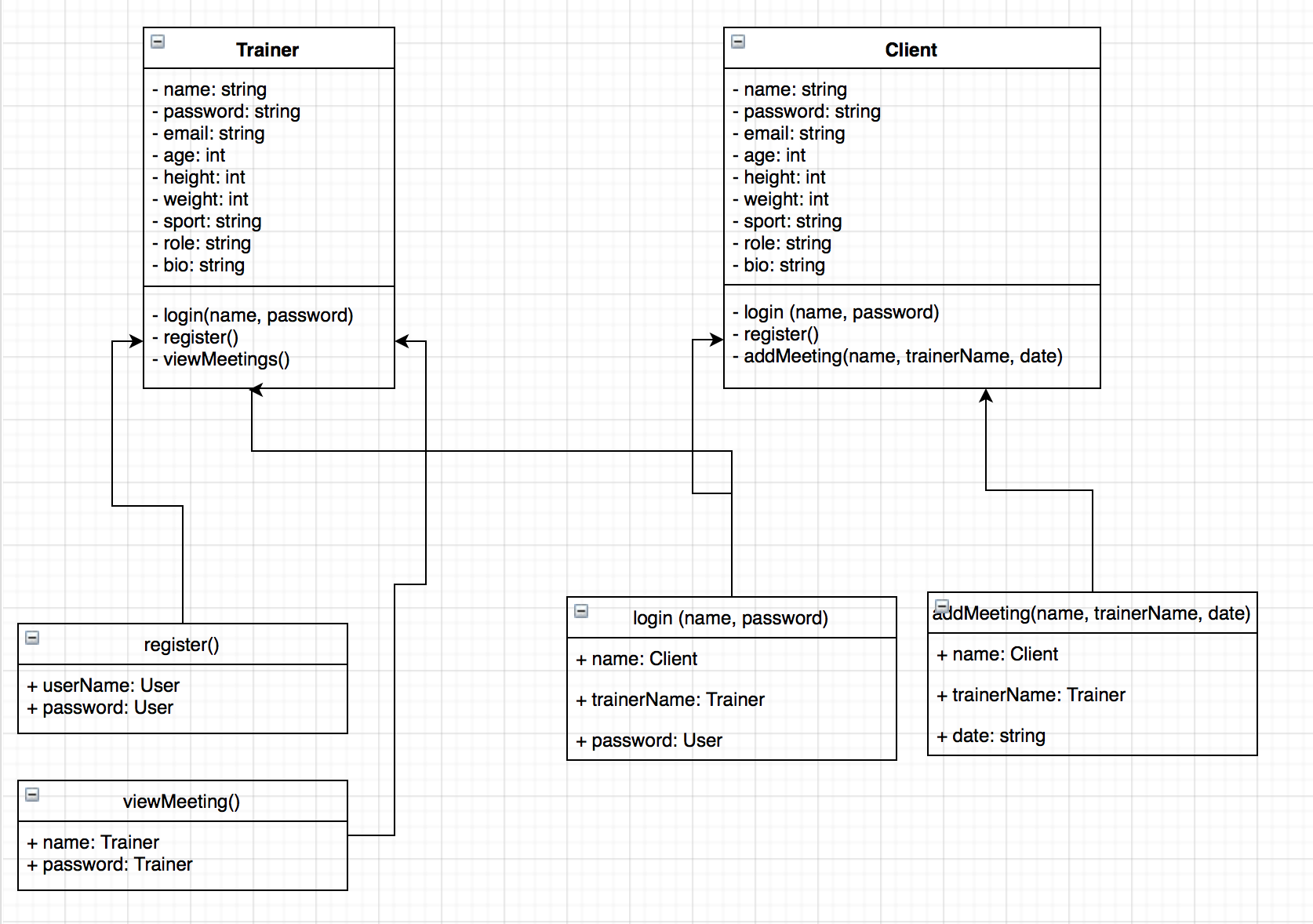
Time was definitely an issue for us. We needed to make sure that we had the front and backend code working well. With more time we would be able to make a better, more in depth application.

**2.4 Possible Evolution**

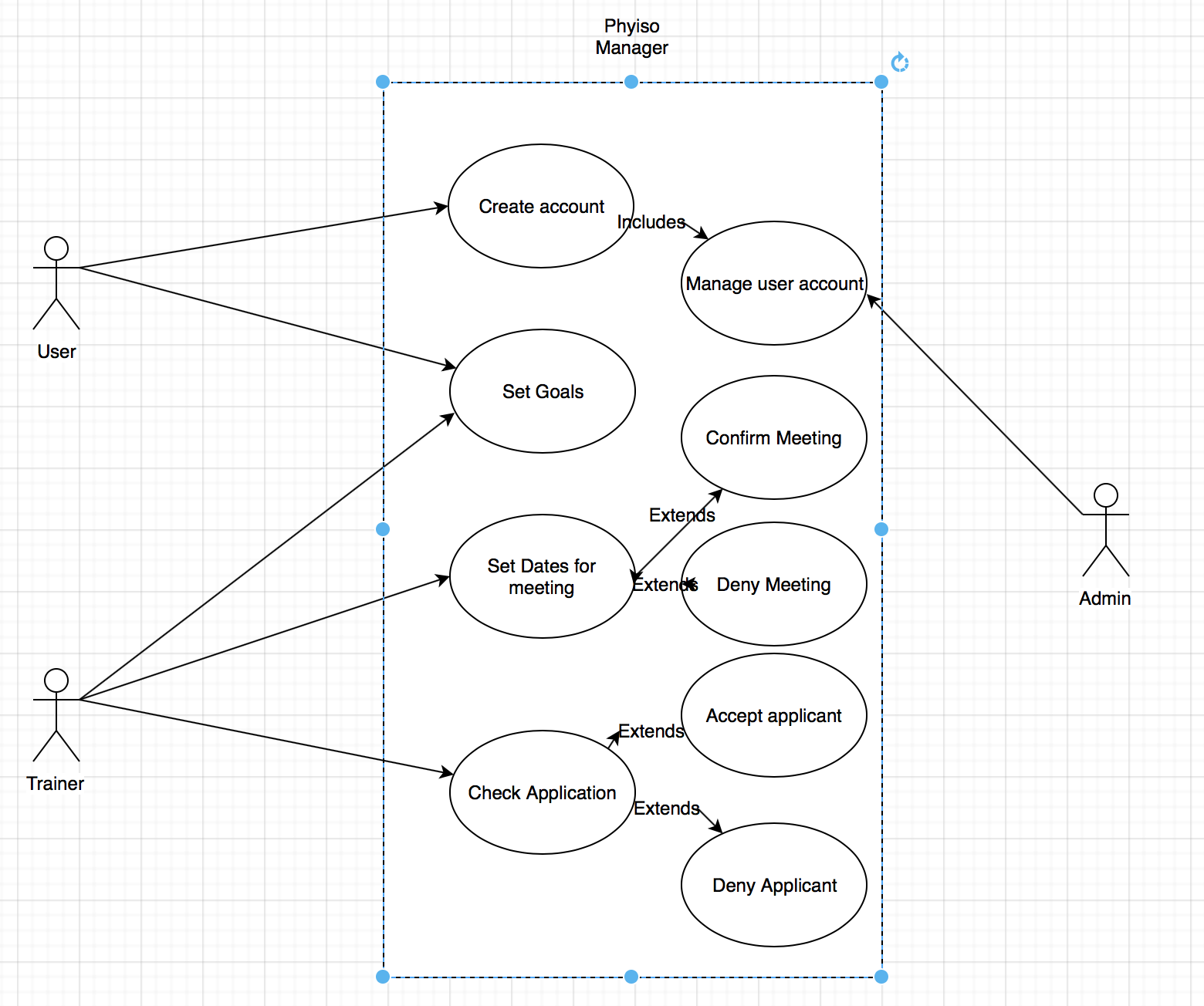
With any product, we hope to have room for our product to evolve. With this being said, we have a few things that we hope to be able to include down the road. One feature would be a real time map that has nike+ or fitbit capabilities. Having the application be available to link to a product like this would be enormous considering the amount of time that it would save users when entering information in.

**3.0 Diagrams**

**3.1 Class Diagram**

****

**3.2 UML Diagram**

****

**3.3 Test Cases**

|  |  |
| --- | --- |
| Test ID | 1.0 |
| Title | Correct Login |
| Feature | Log into Physio Application |
| Objective | Type correct username and password to access the web app |
| Setup | Any laptop that can access internet |
| Test Data | Username: Dan, Password: hello |
| Test Actions | Launch web app -> Type in information |
| Expected Results | User should be able to log in and access web app |

|  |  |
| --- | --- |
| Test ID | 1.1 |
| Title | Incorrect Login |
| Feature | Denied access from Physio App |
| Objective | Inform that username or password is incorrect |
| Setup | Any laptop that can access internet |
| Test Data | Username: wrong, Password: password |
| Test Actions | Launch web app -> Type in information |
| Expected Results | User should see an error message and not be able to log in |

|  |  |
| --- | --- |
| Test ID | 1.2 |
| Title | Registration |
| Feature | Register as a Physio user |
| Objective | Fill out information |
| Setup | Any laptop that can access internet |
| Test Data | User Information |
| Test Actions | Launch web app -> Click Register -> Enter Information |
| Expected Results | User should be able to enter credentials in login screen |

|  |  |
| --- | --- |
| Test ID | 1.3 |
| Title | Preview trainers |
| Feature | User can check information about specific trainers |
| Objective | Find trainer details |
| Setup | Any laptop that can access internet that has access to Physio website |
| Test Data | Trainer information |
| Test Actions | Launch web app -> Go to trainer tab -> click any trainer |
| Expected Results | Trainer information should be displayed |

|  |  |
| --- | --- |
| Test ID | 1.4 |
| Title | Set a meeting |
| Feature | User can set a meeting with a trainer |
| Objective | Fill out meeting details |
| Setup | Any laptop that can access internet |
| Test Data | Meeting Information |
| Test Actions | Launch web app -> Login -> Navigate to Personal Trainers Tab -> Click on a personal trainer -> Click Schedule a Meeting -> Fill out Information |
| Expected Results | User should be able to see the meeting details that were just entered |