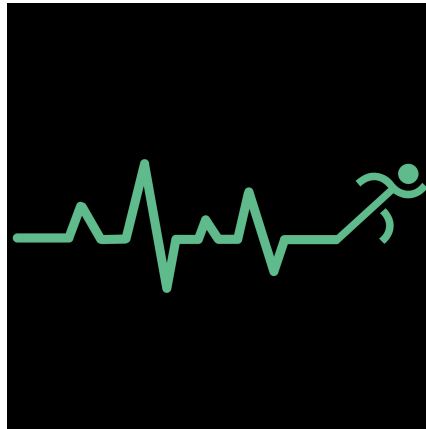


Version 1

Team Agone

Software Design Document

February 4, 2022



Team Members:

Rylee Mitchell , Kaila Iglesias , Samantha Rodriguez ,
Jonathan White Velasco

Sponsor:

Adam Stepanovic

Mentor:

Volodymyr Saruta

Table of Contents

Introduction	3
Implementation Overview	5
Architectural Overview	8
Module and Interface Descriptions	11
4.1 Back-end Database Module	11
4.2 Frontend Display Module	12
Implementation Plan	15
Conclusion	16

Introduction

A study from Yale found that 50% of regular runners get injured every year, primarily from overuse. That is one in every two people who run regularly that will suffer from an injury just this year. Insane! For an activity that is so common among people of every age around the world, it is quite a shame that this statistic is so high.

For almost any athlete or any individual that participates in sports, whether it be competitively or as a hobby, injuries are among the worst culprits for impeding progress. While some preventative measures can be taken, such as using proper form, stretching, and taking time off from activity, it is virtually impossible to prevent injury let alone predict when it will happen. As a result, injury is something that can only be responded to after the injury has already occurred.

However, PWR Labs has a better solution - to predict and prevent injury rather than treating injury after the fact. Reading through various statistics on athlete performance and seeking out the minor indicative details of change in an athlete's body is a process that can be time consuming, but incredibly revealing for coaches. These details can influence the development of dynamic training regimens for each athlete in an effort to protect their bodies, thereby reducing injuries. PWR Lab's vision to help facilitate this process, is to take the biometric data that can be tracked from wearable devices (such as Apple Watches and Fitbits) and make it easily accessible to be viewed and analyzed by coaches.

Being able to predict and prevent running injuries is extremely valuable. However, this can go far beyond only runners; this wellness portal could be used in several different professional fields. For example, physical therapists play an integral role in helping athletes and others recover from varying types and severities of injury. Consider a therapist that is trying to determine whether their suggested exercises and activities have been successful in treating an individual. Our wellness portal, used alongside devices that track the necessary biometric information related to that individual's injury, could provide the physical therapist with visual information that would allow them to more easily see whether progress is being made. They could even fine tune their treatment plan and make adjustments that could rapidly improve the healing process, allowing people to recover from injuries faster.

This is why our wellness platform is so vital to not only runners, but to anyone that could benefit from visualizing data relating to and produced by the human body. The key user level requirement is customizability as each coach has a unique vision for how they want

this portal to function. This necessary factor drives many other user level requirements such as flexible dashboards and personalized data visualization. We must also make sure that this portal along with it's connected database are both secure and connect to the existing PWR Labs API. This wellness platform is expected to perform efficiently for each individualized lay out and make the entire data collection and analysis process simple.

Now, we will establish the software design specifications and suggest a general blueprint for development. This way, we can dive deeper into a careful analysis of the architectural design and implementation plan for building such a platform. At this stage of development, we are in the process of coding and developing the key components of the functionality of our portal as well as outlining the framework to make this platform as customizable as possible.

Implementation Overview

We will begin by creating a web application. This web application will be flexible enough to meet different coaches' needs. Each sports coach looks at different statistics to measure the success of their athletes. There are three main categories of permissions for the website: the coach, the staff, and the athletes. There will be subcategories for the staff, such as physical therapists to help with injured athletes and therapists to help with athletes that are having mental health issues. For each of these different categories, they will have different dashboards. The dashboards are what the permissions will be tied to. For example, the physical therapist can only see the athletes on the injured dashboard.

In order to improve the current system our client has we will:

- Have the coach create their own environment and dashboards in the beginning of the process.
- Automatically populate the database as the athlete enters data through the web application.
- Allow for new information to be added such as injuries and mental health.
- Stretch goal: Provide data visualization to not only make it visually appealing for the coach, but to also facilitate the analysis process.
- Include a messaging system so a coach and athlete may converse about workout plans.
- Substantially lower the amount of work our client has to put into each environment for different coaches.

Our web application will integrate with the client's API to collect the data from athletes' wearable devices. The collected data populates the database and is shown on the appropriate user's dashboard. A coach can select different ways to view the data, all of which will be presented in a simple to interpret format. A functioning web portal that refines these processes will also allow for the client to provide more coaches with these services. By providing a better environment for the coach to analyze data, rather than spend time collecting it, this solution will also ease the workload of the coach. We will still need to implement our own database to store dashboard data so users may be able to access an athlete's historic data. A case diagram of how the system should work is displayed below.

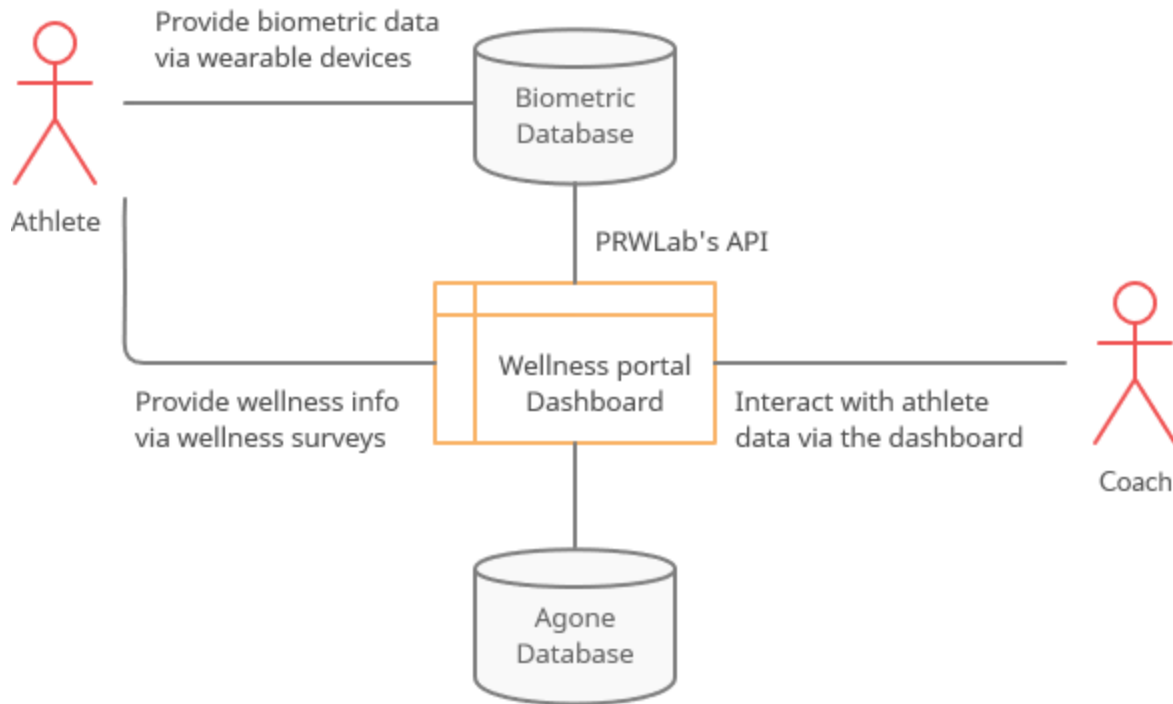


Figure 1: Interaction between users and the system.

The breakdown of the chosen technologies:

A. Back-End Databases

a. PWRLab API

- i. This API will provide us with the biometric data that will populate the dashboards.

b. AWS

- i. AWS will be used to host our site and connect to our site database

c. PostgreSQL

- i. We will be using a mixture of PostgreSQL and AWS to store data from the site and communicate with it/pull it when necessary to the user.

B. Front-End Display

a. React

- i. Our base framework will allow us to create an in-depth, flexible and customizable web portal.

b. html/css

- i. Necessary for creating and styling our website.

c. Apex charts

- i. If we choose to implement data visualization this is the software we will use. Not only being the cheapest option (free) it will provide visually appealing graphs that are easy to implement.

d. Firebase

- i. We will implement Firebase authentication to protect user data.

- ii. Create user accounts to display different views depending on the credentials.

These technologies will be the basis for our system. Each will be necessary for the overall architecture of the system and provide some form of functionality that will ease the implementation process.

Architectural Overview

Figure 2 below displays an overview of the system's architecture. The Front-End and Back-End will make up our two-part system. They will need to interact with each other by pushing and pulling data. The back-end consists of our hosting platform AWS, our team build database that will store athlete biometric data, wellness information and a coaches view settings pertaining to a team, and PWRLab's API. The Front-End will be a react built web application that will display a multitude of features including: Displaying API data, multiple views based on which user is signed in, a Coach/Athlete message system, a request for athlete wellness information, Flexible and customizable dashboards, secure accounts, and the ability to request historic student data. Below we will delve deeper into what each component does and how it interacts with other components of the system.

A. Front-End

a. Displaying API data

- i. The Dashboard will display populated athlete biometric data to a table located inside the athletes tab. This table will also contain wellness information sent by the athlete. Data will be requested through the api and distilled into the database to save for later use and record keeping.

b. Message System

- i. Coaches will be able to comment on workouts, and respond to wellness information provided by the athletes. As well as comment or respond to any other information that may be requested.

c. Wellness Information Request

- i. There will be a request for information button that will email all team members requesting them to complete a wellness information survey in their dashboard. The athletes will then log in and fill out the survey and that information will be uploaded to the Coach dashboard to be viewed or commented on.

d. Flexibility and Customizability of the Dashboard

- i. The dashboard is fully customizable for the coaches when they initially join. The dashboard will provide them with a template that will import the student data and allow for them to add staff, display Graphics(stretch goal), view all staff views, move around student data, and hide/show student data.

e. User Accounts

- i. Users will be able to create accounts. These accounts will have permissions based on if you're a coach, staff or athlete along with different views. Coaches will be able to invite athletes along with staff and customize staff views as well as their own.

B. Front/Back-End Interaction

a. Distilling Data

- i. The front end will distill student data to the database. The backend receives the data and saves it for later use.

b. Pulling Data

- i. The database will pull data when requested by the front end. The front end will be able to access the data of previous students to populate it into their dashboards

C. Back-End

a. API

- i. POWERLabs API will directly connect with our backend sending data from POWERLabs database and transmit that information on request to our dashboards through AWS.

b. Hosting Service

- i. Our website and database will be created and hosted by AWS. This will allow easy connectivity and communication between our website and database.

c. Data/History/Settings Storing

- i. The database will store athlete biometric information into an individual athlete history section along with any provided wellness information. Views created by the coaches will also be stored and saved after creation.

d. Database management

- i. PostgreSQL will be used to manage data inside the database

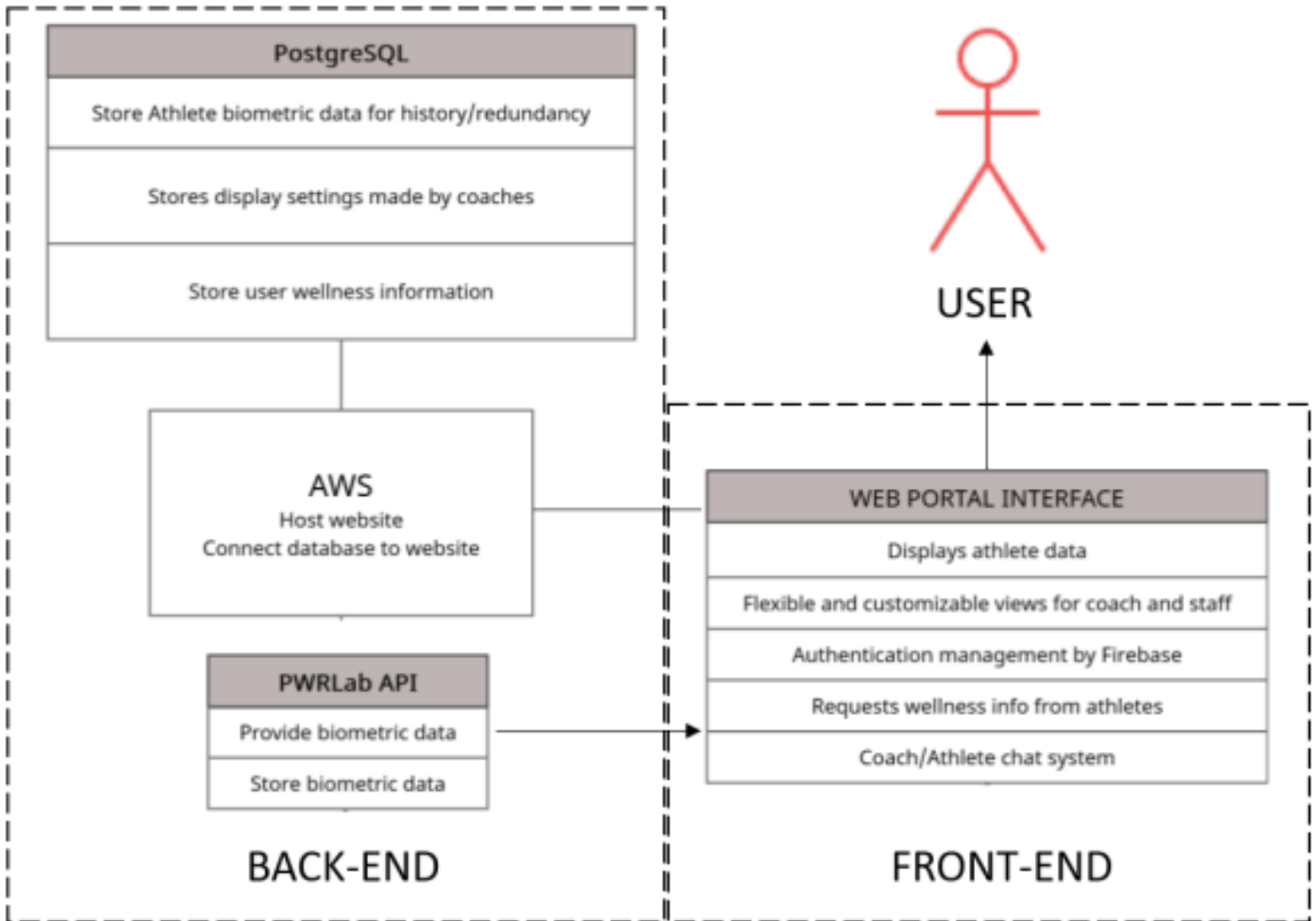


Figure 2: Architectural layout of the system and how parts of the system interact with each other.

Module and Interface Descriptions

As indicated in Figure 2, our system is divided into two main modules: the frontend and the backend. In the following sections we will be diving deeper into the tools and architecture used in each module and explaining their functionality.

4.1 Back-end Database Module

The backend module will hold a database in order for users to interact with. Organization of the backend will be essential to keep the users in order. In Figure 3 is a database layout of the organization of the backend. There will be three main users in the system: the students, the coach, and the staff. They will all be connected through a team database. The coach and staff will be interacting with the data and the students will be entering the data. The students are going to be entering two different types of data. The first type is their activity data that is either collected from their watch or manually. The second type is their personal data that will be entered manually.

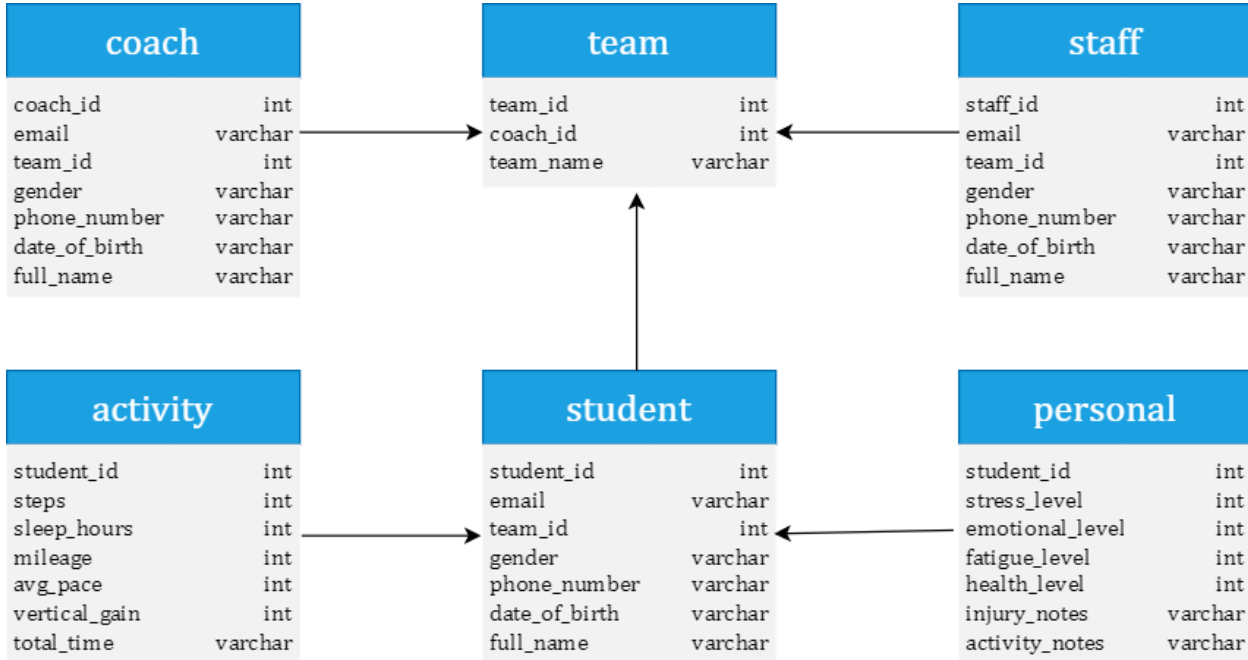


Figure 3. Database layout in PostgreSQL. NOTE: the “activity” and “personal” layouts will differ for each coach using the application. This is an example based on information our client provided.

A. Coach

The coach will be the main user of the system, since they are initiating the application. When they start the application, they will be the first to enter the information into the database. Coaches can have athletes in different teams which is why they are connected to the team database. The team database connects all the

users to each other. The coach will interact with the data the athletes enter. Which will be discussed later in the front end of how they will be interacting with it.

B. Staff

The coach can add staff to their team to assist them. This can be an assistant coach, physical therapist, or a counselor. These users won't be entering data but they will be interacting with it. They are kept separate from the coach to have separate permissions. The staff may not be able to look at all the athletes in the team. For example, the physical therapist doesn't need to look at fully healthy athletes. As seen with all the users, they will be given a randomized ID number. This will assist in ordering which user gets specific permissions.

C. Student

The athletes, who are students in this case, will be the users entering the data. This data will be the one the coach and staff will analyze to better understand the athletes needs. The athlete will be entering two different types of data. The activity data and personal data need to be kept separate since they fulfill different needs. The activity data is collected from their sports watch or collected manually. The data collected will differ depending on what the coach wants to see. In Figure 3, this is an example of what a coach may want to collect. The personal data is collected manually. It will be put in a questionnaire form to see the mental state of the athlete.

Above is how the information will be kept and be used. In the next section, it will be discussed how all the information will be displayed in an organized manner.

4.2 Frontend Display Module

The frontend display module communicates with the backend and effectively displays data to the user interface using several tools. The major components of the frontend module include the chosen React framework, HTML, CSS, and JS for coding the application, Firebase for authentication services, and finally, Apex charts for visualization if data graphs are implemented. All of these components work to provide a flexible and highly customizable health and wellness web portal, targeted at making communication between coaches and their team far easier.

The front-end display module, also the public interface, is based around each user of the application. There are several different user classifications in our system. Each user classification represents a sub component of the frontend display, these sub components include: coach, athlete, and staff. The information displayed in the user interface will differ based on the type of user. The following section provides descriptions of each frontend module, or user view, and a UML class diagram depicting the key functionalities of each view.

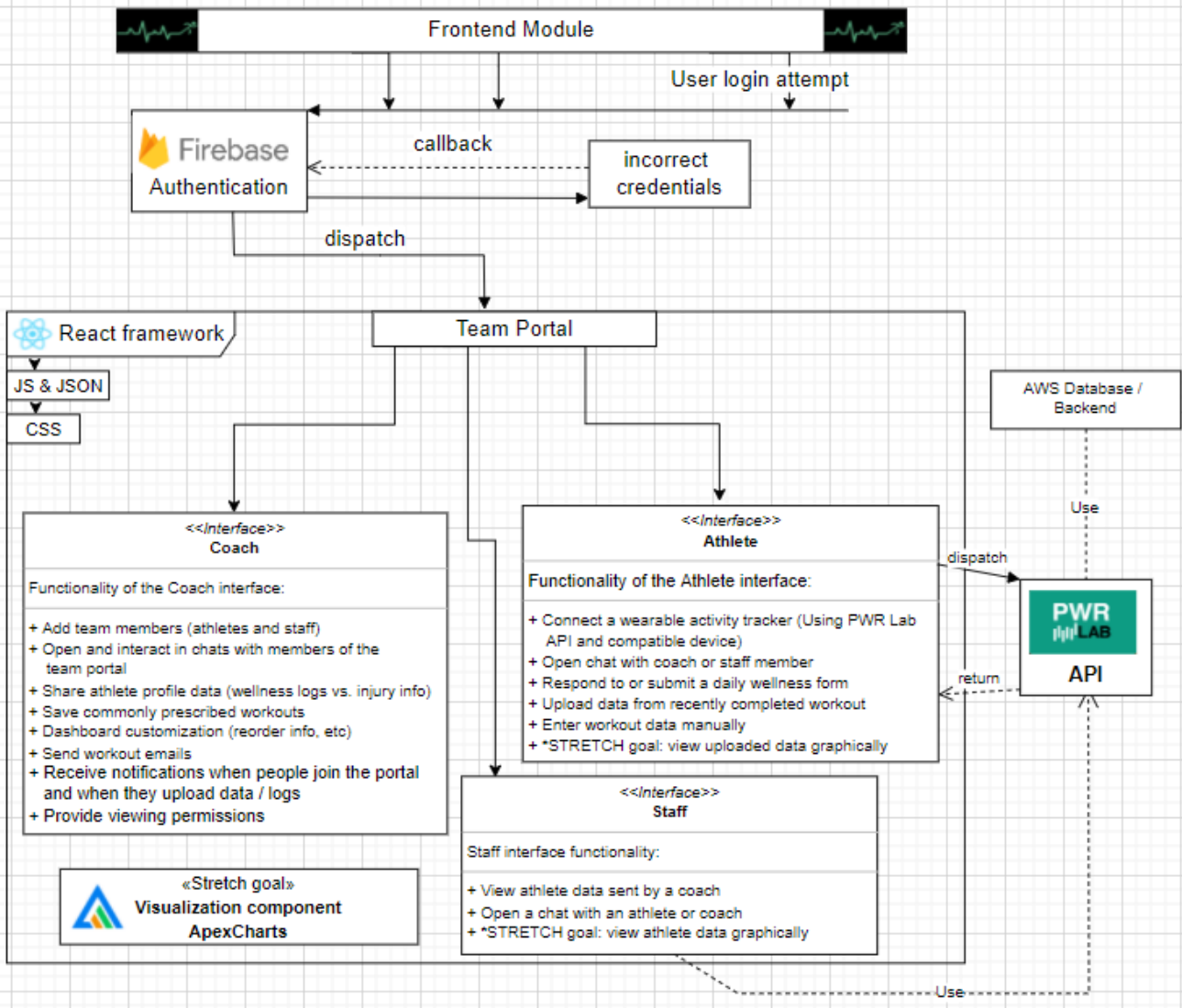


Figure 4. This diagram depicts the frontend components and how they interact with PWR Lab's API and the database. NOTE: Each <<Interface>> block represents a module or view of the frontend.

A. Coach Interface:

The coach's interface is the first to be created when setting up a new team's personalized health and wellness portal. This means that the coach will be the first to sign up for an account and will subsequently become responsible for onboarding other users into the system (onBoardTeamMember function). Thus, the coach is one of the most important user interfaces in this web portal, as customization occurs here and will allow coaches to arrange their dashboard view to optimize viewing of athlete metrics and wellness. Coaches will also have the option of customizing their dashboard by uploading a team logo. Coaches will be responsible for onboarding athletes and staff members and designating the appropriate classification (i.e. athlete, physical therapist, etc.) to the new user. In terms of functionality, the coach will be able to open chats for communication with staff and athletes, save workouts

that are often prescribed or are important, share athlete profiles with staff members and limit the amount of information that the staff member is seeing, and send out various templated emails that can be customized to include different information. One of the key features of the coach's interface is that it will have the highest level of access in regard to user information and designation of athlete viewing permissions to support staff members.

B. Athlete Interface:

The athlete interface serves as the second most integral part of the portal. Information provided here will determine what is displayed to the coach, support staff, and of course, the athlete. Since information entered by the athlete is typically related to health, wellness, and exercise, it is critical that the information collected is accurate and easy to distinguish across athletes. Athlete data should also be stored for a period of time, as discussed with the client, training data should be available for the past 7 days at a minimum. The athlete module interface in the frontend will need to be integrated seamlessly with the data collection API developed by PWR Lab and with the backend so that information may be stored in the database. Furthermore, athletes and coaches will be in consistent communication regarding daily fitness and wellness logs, injuries, and other related information. For this reason, students should be able to open up a chat with the coach or other support staff when necessary. As for uploading data, athletes will have the choice to either enter exercise data manually or enroll a compatible wearable device with the PWR Lab API connection URL provided by the coach. If implemented, students will also be able to toggle between viewing their data qualitatively and with graph-like visualizations.

C. Staff Interface:

The final interface that is relevant to this portal, is the staff interface. This interface will be similar in layout to the head coach's interface as it will contain the same template view of athlete information that is setup by the coach. However, some fields will be removed or grayed out depending on the type of user that is interacting with the system (i.e physical therapist should only see injury and exercise data, etc.). As mentioned, all staff members of the wellness portal will have a reason to communicate with an athlete at some point or should have the ability to open a new chat with a specified athlete. Lastly, the staff interface will also have the option to toggle between a qualitative or graph view of the athlete exercise data.

Implementation Plan

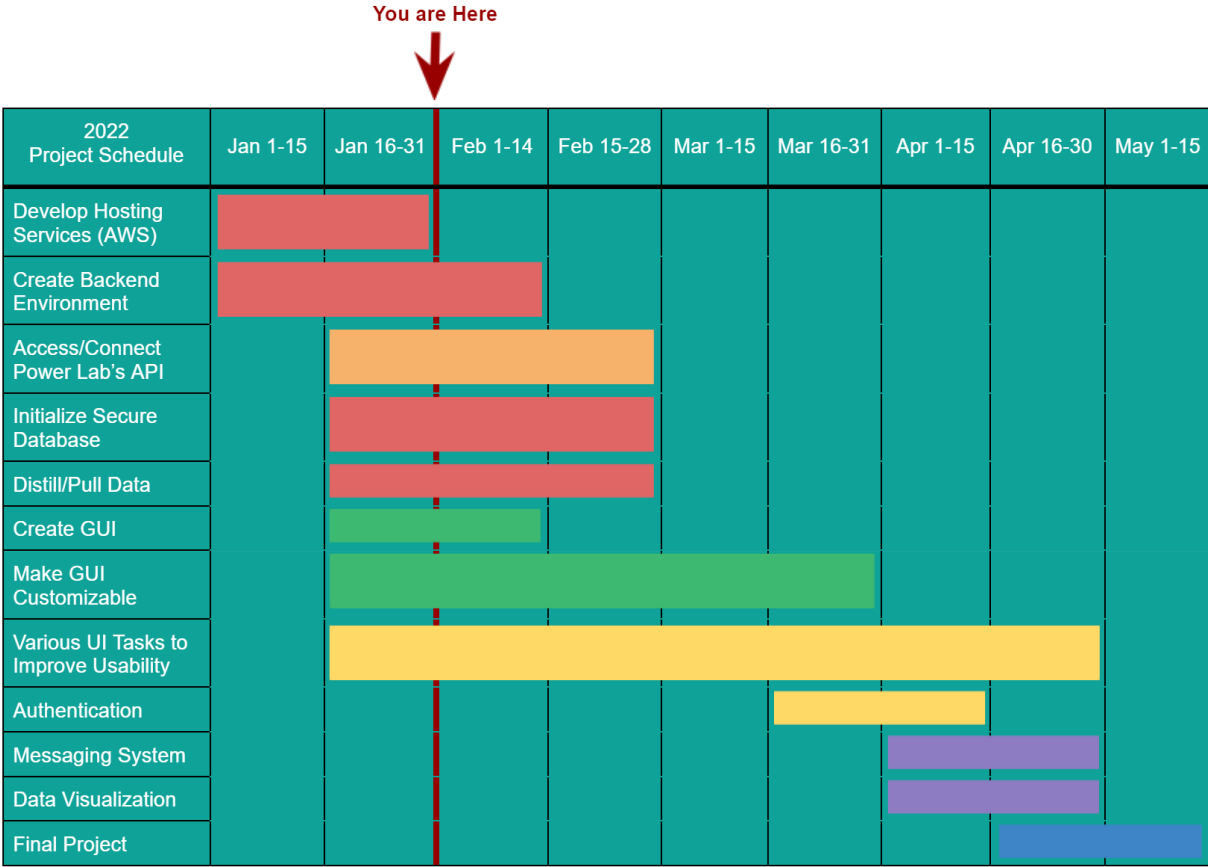


Figure 5: Gantt Chart of Anticipated Schedule

Above is a Gantt chart that depicts our current implementation plan and the due dates we have anticipated for each design component. The items in red are spearheaded by Sam and Jonathan while the items in green and yellow are designated for Rylee and Kaila. In short, Jonathan and Sam will be focusing more on the backend development whereas Kaila and Rylee will prioritize the front end. The orange task is one that our entire team will be working together to complete, as are all of the tasks- but we based development on our individual strengths. To begin, we must develop a hosting service as well as create our entire backend environment including connecting to POWERLabs API, initializing our secure database, and enabling the feature which allows us to pull and distill the incoming data. This is the first priority as without these design features, we cannot effectively continue with our project. This can be partnered with our creation of a GUI, which combined form the backbone of our wellness portal. Throughout the entire development phase, we will be implementing various tasks in an effort to perfect our GUI. We meet with our client every two weeks (on each thin black line) to show progress and inquire about any potential changes we may need to make. One of the final steps will be to create an

authentication system to encompass our platform. A live messaging system as well as data visualization capabilities are stretch goals and will be attempted if time permits. The final project will be delivered by the end of April/early May.

Conclusion

This document is meant to provide a complete description of the software design schema of our health and wellness portal. In this document, we have provided graphical examples including: an architectural diagram and frontend and backend diagrams. The architectural overview will provide a high-level understanding of the framework and design for the portal. The frontend and the backend modules provide more details of the software and services we used to create them, as well as UML diagrams depicting the expected functionality in each.

Overall, this team-oriented health and wellness portal was created to provide a way for coaches and their staff to have a direct impact on the wellbeing of their athletes. Rather than simply observing and reacting to changes in physical and mental health, this portal, along with PWR Lab's API for collecting biometric data, will expedite a coach's ability to develop dynamic training plans. By giving athletes the ability to upload training data, respond to daily wellness forms, and report recent injuries or injury progress, coaches and staff will have more insight into the details of an athlete's wellbeing. They will also be better equipped to make decisions regarding athlete exercise regimens. Furthermore, centralizing the location of all of this information will make for a much more efficient approach. Compared to sorting through numerous sheets or documents dedicated to individual athletes, this portal will contain data customization and communication tools to facilitate the communication between athletes and staff. Allowing coaches to share specific pieces of athlete data with staff members, such as a recent injury with the physical therapist, will also be a primary functionality. This capability will greatly contribute to quick and easy collaboration on decisions that are intended to maximize athlete health and wellness. While the main goal in developing this software is to ease the current data analysis process and communication efforts between coaches and their athletes, this portal has the potential to positively impact athlete performance. By creating an environment where analysis of athlete health and wellness are a priority, athletes can receive optimal advice from coaches who have more time for one-on-one interactions.