CU Fueling Station Web Portal

CSCI 3308 Project Proposal

Group No. 014-1: Team 0x1

Team 0x1 Members
Tyler McCormick, Ella Herniak, Jake Derouin, John Aldrete, Michael Metz, Ryan Chaknova

Table of Contents

1.1 - Application Description

Vision Statement: To better connect the Colorado Performance Nutrition department with its athletes through technology in innovative and impactful ways.

The CU Fueling Station Web Portal provides an easy to use interface for CU athletic trainers and nutrition professionals to communicate with athletes and ensure they are fueling to be able to perform their best. Working with the already launched iOS/Android applications, the site will allow staff of the CU athletic department to easily view data collected from the mobile application relating to attendance of the Fueling Station.

CU athletes will be able to check in fast with the iOS/Android app, and that data will be stored into a database which can be accessed by the web portal, so athletic staff can see who exactly has checked in on what day. Additionally, nutrition staff will be able to make graphs and to find patterns in athlete fueling station attendance. Nutrition staff can also easily send out messages and notifications to athletes of certain sports and give nutrition information based on certain circumstances. For example, a nutritionist may be able to advise athletes to eat more of a certain food group for an upcoming game or training session.

Faculty will login with their CU Identikey (pending OIT approval), and be given a sleek UI that will give them information they may be looking for right away. Easy to use UI will let them easily navigate to the page they're looking for, whether they’re looking to export data to a certain file format, or just check on who has checked into the fueling station dining hall.

1.2 - Team Organization

1.2.1 - Version Control

We will be using GitHub as our version control system.

Link to Github Repository

1.2.2 - Development Methodology

Our team will be following the Agile methodology for software development. We'll be using the Jira website to organize our sprints, epics, and user stories

Link to JIRA Board

1.2.3 - Communication Plan

Our team's main mode of communicating is through Discord. In the server we created we can send messages to the entire group or to direct members to discuss the logistics of the project. We can also create meetings and talk amongst each other using voice channels.

1.2.4 - Meeting Plan

Weekly group meetings will be held remotely using Discord.

Times:
Sundays 1pm-2pm
Wednesdays 6pm-7pm

TA Meeting: Thursdays 10:15-10:30am; in-person ECCR 114

1.3 - Project Architecture

1.3.1 - Front End

The front end of the CU Fueling Station currently consists of 3 mobile apps that have already been deployed to production to allow athletes to check in. In addition, a Web Portal will be programmed in HTML, CSS, and JavaScript to provide interactive elements and access to staff functions without the use of an iOS device. If decided upon, we may implement another CSS library such as Bootstrap or TailwindCSS to make creating style classes easier. To allow our front-end layer to communicate with the back-end server, we will be using NodeJS. Node will also let us implement any other features we may need, including potentially login/authentication. As of now, the plan is to get permission from the CU OIT to use Identikey for authentication since the application will be used only by CU faculty, however if not we will have a backup plan in place.

1.3.1 - Back End

Currently, the plan is for the web portal to pull data from an existing Firebase database as well as a PostgreSQL database. The CU Fueling Station mobile application uses Firebase to store its data for quick and accurate check in records as well as get data on the food menu and time changes. Because Firebase has caps on large data transactions, we will be implementing a bridge to archive check in data daily by moving it from Firebase to PostgresSQL.

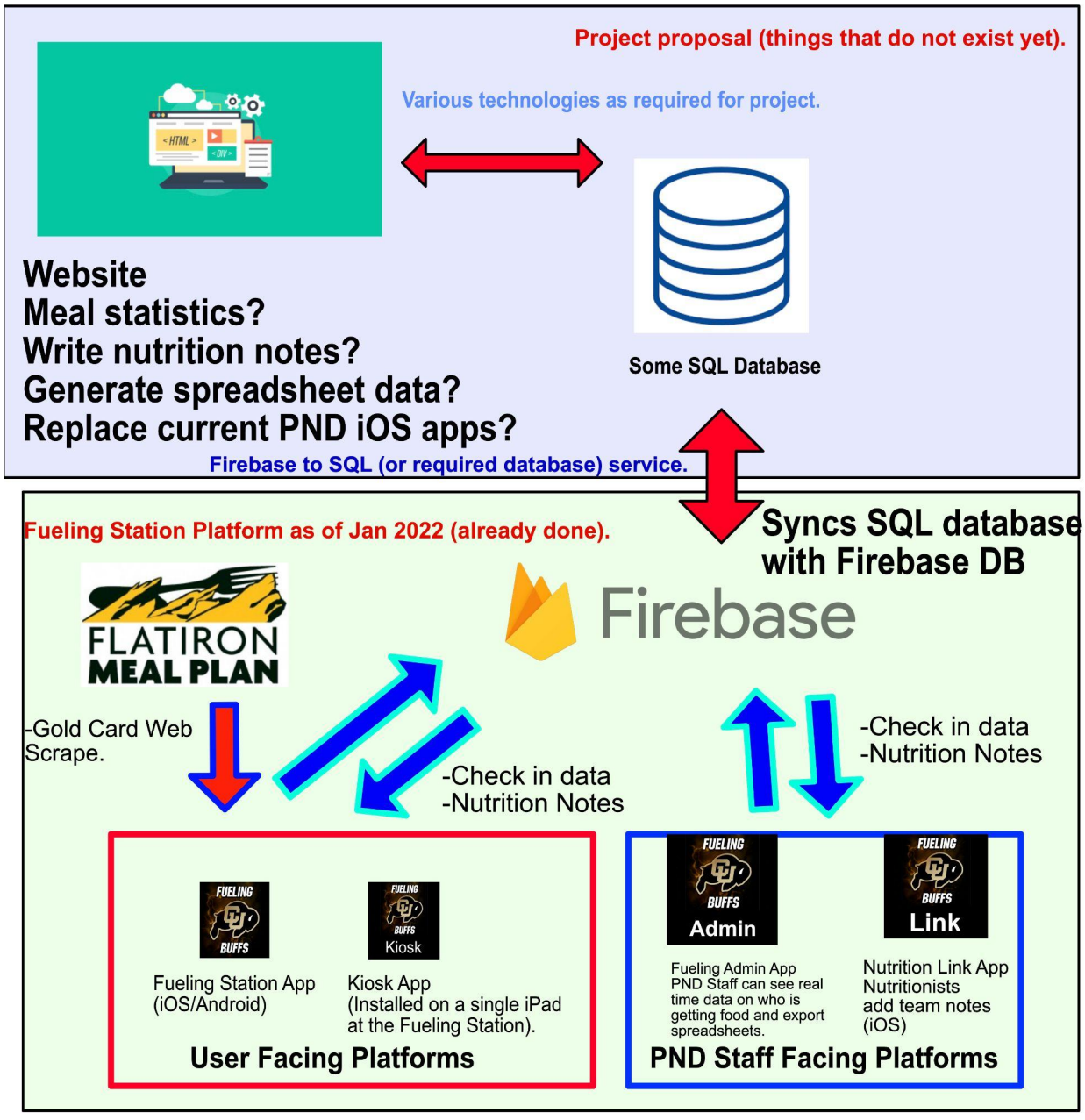
The web portal will need to be able to access and modify this data from both databases to allow nutrition staff to view archival check in data and conduct data experiments as well as write nutrition notes to communicate with athletes.

The bridge between the Firebase database and the PostgreSQL database will be written in Swift as a macOS app with the possibility of moving the bridge into the same program as the rest of the backend in the future.

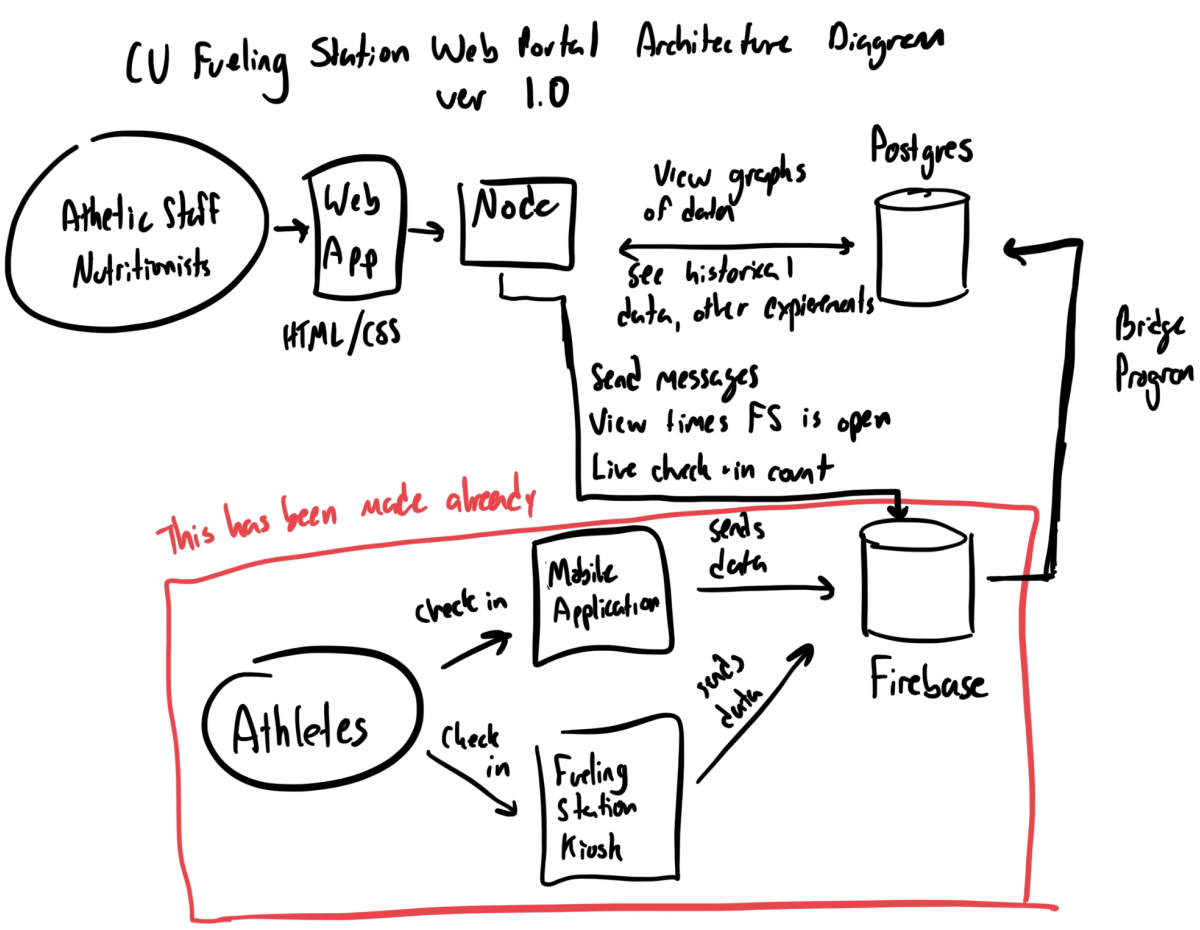
Link to architecture diagram

1.4 - Diagrams and Pictures

1.4.1 - Initial Architecture Design proposed by Jake Derouin



1.4.2 - Architecture Diagram 1.0



1.4.3 - Use case diagram