Matthew Gleich
Goffstown, NH 03045

Project Proposal

Goffstown Sports App

5th July 2019

OVERVIEW

The Goffstown sports app would simply display scores and player information from live home game. Using a Raspberry Pi (small computer, about the size of a credit card) that is hooked up to the scoreboard, we can upload the current score shown on the scoreboard to the Google Server. Then, when you open up the app the app loads the data from secure Google servers. It then takes that data and displays it to you a panel like a layout (similar to Instagram's layout).

GOALS

1. Create a project in which different groups can get involved to make. A few examples:
    a. People interested in art could help with the design.
    b. People interested in tech could help write the code for the app.
2. Make an App

SPECIFICATIONS

Getting Scores

At this time, I haven't had a chance to go to GHS, and test their scoreboards. The goal is to use the Raspberry Pi to get the current scores being displayed on the scoreboard and upload them to Google Servers. In order to test this, I would need specifications on the model of all the scoreboard controllers and be able to do some testing with it.

Google Database

Using a free Google service called Firebase, we are able to upload data to Google server. We are planning on having a password that people enter when they download the app (password would be the same for everyone and displayed on posters around the school). So, people who enter the password correctly are able to read the data and only the Raspberry Pi's are able to write data to

the Google Server. We do not collect user identifiable data. All data collections will be shown in official terms of use. One of the many great features of using the Google Cloud is that we can update the scores in the app without the user having to click a refresh button or something like that. It's all real-time.

Building the App

We are going to be building the app using a programming language called Swift. Swift is a programming language for building IOS (iPhone), watchOS (Apple Watch), MacOS (Mac Computer), and tvOS (Apple TV) apps. All of the code will be open source on Github.