

Kern Khanna
Cody Sugarman
Connor Hegarty
January 19, 2017

CS194W Project Proposal

Description of the Product: This section should explain what your product is from the user's perspective. This is not a technical overview of your project, but rather a description of the features and functionality. (You will give a technical overview of your project in a different section.)

This product is designed to be a mobile iOS app that allows one to follow their favorite NFL team. On first use of the app, the user will be prompted to enable geolocation services and create an account via either a Facebook OAuth or simple email/password combination through Firebase. Once an account is created, the user will be able to begin to tailor their main feed by selecting their favorite team. The app will notify the user of any recent scores and relevant statistics on the official team page. While the user will have the option of selecting any team, the app will also make team recommendations based on the user's location. For instance, a user in San Francisco would be suggested to follow the 49ers while a user in Boston might be suggested to follow the Patriots.

In addition, users will be able to post pictures regarding their viewing experiences and engage in chats on game threads. It is designed to make the act of watching sports more collaborative and social. Ideally, many users would be able to "live" the gameday experience by seeing pictures and comments about the game as it is taking place.

Need for the Product: Discuss why your project will be useful.

There is a definite need for simplistic software that allows fans to engage in the sports viewing experience. Currently, there are many sites like ESPN and NFL.com that store statistics about sports team. However, the purpose of these sites is to educate fans about their favorite teams rather than to make sports watching more social. There is not a robust mobile application that allows individuals to post pictures about the stadium experience on a feed while engaging with friends at the same time! Instead, internet game day threads on sites like Reddit are too saturated with advertising and not welcoming to the average fan. To be specific, these sites involve high amounts of cyber bullying and are not only focused to friends unlike this app. In addition, these websites are also too broad and may not invite our targeted audience who is only focused on sports.

Potential Audience: Highlight the potential audience for your product. How large is the audience? What is their level of technical sophistication?

The potential audience for this project will be fans who enjoy the NFL (if successful with NFL, we will expand to other sports). While this may seem like a niche market, roughly 50% of all Americans love the NFL, which causes the national market size to be around 150 million people. In addition, we hope that the social element of this app will invite people who normally do not watch the NFL to use the product so that they converse with their friends. This could increase the market size even more. If the application is designed well, we will first market it on college campuses. Assuming this process goes well, we will then try to realistically capture up to 0.8 to 1% of the total market share after Year 1. These numbers are all optimistic projections because we feel that this application has promise.

Since the application is designed to be easy to use, users do not need to be technically sophisticated. If the user is able to use apps like Snapchat and Facebook, using this application should be straightforward. In conclusion, this app has great potential because it is an easy to use application that can provide value to a great segment of the American population.

Discussion of Competing Products: Describe any competing products which currently exist in the marketplace. Tell the reader why your project will be different or better than existing products.

While there is no shortage of sports app for iOS, the majority provide functionality prioritizing score reporting, news, and live sports streaming. The most popular competitor is currently ESPN, which also has functionality to provide a personalized sports news experience for all the franchises and athletes the user cares about. Cable subscribers can also use the WatchESPN app to stream live games and watch on-demand videos. The most direct competitors to our application are the NFL app and GameOn. These both provide custom feeds for sports fan, with GameOn providing communication between fans of both the same and rival teams. Other competitors in this space would include WhatsApp and Facebook. While they are not designed around a centralized sports feed, they are currently the most common means of online communication for sports fans.

Where our app aims to stand out is in providing a centralized feed to be shared amongst friends of the user. Instead of connecting all fans of the sport, our goal is to create a small community amongst friends (via Facebook and geolocation) where fans and rivals can share photos, comments, and recent news.

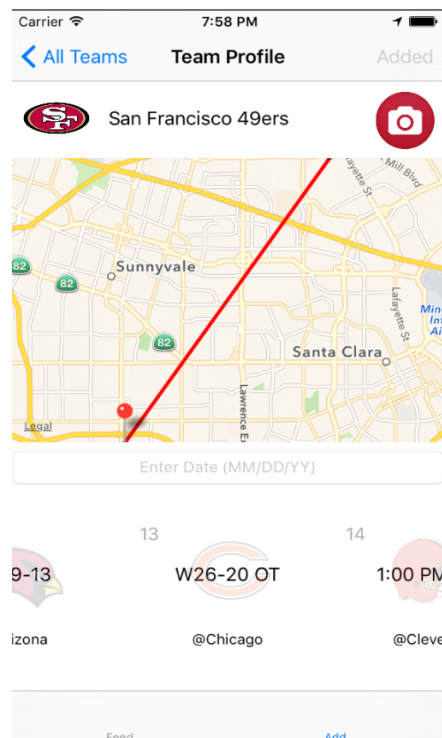
High-level Technical Design: How are you going to build your project? Give an overview of the technical architecture of your project. Your description should name the specific programming language(s), libraries, frameworks, and other technologies needed for the critical components of your design.

We are designing an iOS application in Swift using an MVC framework to implement this product. To complete this project, we need to first fetch the relevant sports metrics, create game-day threads regarding this data, and then render the information in an appropriate manner.

We are planning to get our data from the ESPN NFL page. Specifically, we will initialize a URL session in the background to pull the relevant data (team logo, schedule, location) from the <http://www.espn.com/nfl> link. To help us pull the HTML and JSON data from ESPN, we plan to use the TFHpple Parser library (a common Objective C and Swift library) to gather the relevant metrics. In order to make sure that the app works offline, we will cache some of the data when the user pulls information on the team for the first time.

Another major goal of this project is to enable game-day threads. This will most likely be done using the backend server Firebase (JSON database). We will store user accounts and the chats on a firebase database. Once users are authenticated, we will be able to add privatization to their account and store their favorite teams for example. To implement the chats, we are going to use the provided JSQMessagesViewController (a strong messaging UI) with CocoaPods that allows instant messaging.

Using table-views and scroll-views will be essential to getting a good UI. We will most likely use a table view to store the number of teams in the NFL, and then a map view with a scroll view to show the basic schedule. To help visualize our intended goal, we have included potential mock designs below.



In the left diagram, the user will see a map that will ideally give directions to the stadium and a camera button to add pictures about the stadium. They should also be able to see the result of the game closest to the date provided in the input field. Clicking on each individual game will bring up a chat thread, where users can talk to each other and converse. This will involve synchronizing the UIMapKit framework and the directions API. Since directions can be costly, clicking on the map will redirect the user to the third party Google Maps or Apple Maps. Meanwhile, the right diagram provides the user with the option to pick his or her favorite NFL team. The recommended team is the team geographically closest to the user, and it will be calculated also using the directions API.

Resource Requirements: What do you need to build your project? Are there any unusual requirements or particular challenges?

We should be able to build our project with a Macbook, XCode, and developer tools such as Firebase. For testing purposes, we should only need an updated version of an iPhone. At this juncture we see no need for unusual requirement or particular challenges.

Potential Approaches: Are there different ways to deal with the problem area your product addresses? Why did you choose your particular approach?

While there are different ways to deal with the problem area our product addresses (as done by apps like ESPN and GameOn), none of these applications enable users to interact with their friends on one centralized platform. We believe our approach will encourage users to interact more frequently as it will ideally replace any group texts or GroupMe platforms they would previously use to discuss sporting events. We also project higher user retention due to the nature of the user base.

Assessment of Risks: What are the potential risks involved in your project? What can be done to reduce the risks?

As we will need to use developer tools and services our group members have no previous experience with there is a risk that the steep learning curve will slow the progress of our app development. We will mitigate this risk by assigning each group member specific development tasks so they can learn the technologies needed to achieve these goals and teach the rest of the group. As our app will not require any personal information from the user (outside of login credentials and geolocation) there should be very mild security risks for this project.

Another potential risk is that a competing company might try to enter this sports market because it sees the same opportunity that our group members see. With only three people on the team, we would be hard-pressed to match the effort of a hardworking startup.

Next Steps: What do you need to do next in order to build your product? Feel free to add additional items as appropriate for your project.

The first step we will take in building our project is determining how we want the UI to look and function for the user. We have started to use programs like Sketch to make mock designs in addition to pen and paper. Once we have determined how we want the application to work, we will start building out screens in XCode through Storyboard. After we have a draft completed, we will run user tests to get feedback and improve the UI.