Pitchslapp Final Report

COS 333 Spring 2016 - Prof. Moretti
May 10, 2016

Caroline Congdon - ccongdon@princeton.edu
Andrew Hunt - amhunt@princeton.edu
Zachary Stecker - zstecker@princeton.edu
Team Leader: Andrew Hunt

Introduction

This report outlines the exciting, educational, frustrating, and fun process that we undertook to build Pitchslapp. Throughout the semester we ran into hurdles both small and large, but were ultimately able to overcome almost all of them and we ended up with a product that several students have personally expressed excitement about using in the future. Our reflections on the semester are detailed in this report, starting with the technical milestones and ending with the changes we would have made given a second opportunity. Overall, we felt this was a worthwhile project and we are proud of everything we've accomplished this semester - we hope our enthusiasm is apparent in our final product as well as in this report.

Technical Milestones

For the most part, we were able to stick to the list of milestones we set out at the beginning of the project. Some of these included setup and login, the capability to add and edit songs, setlist functionality, pitch playing, search, sheet music integration, and group management. Our development process for each feature usually consisted of adding it first in iOS, then modifying the plan as needed for web. Throughout the semester, we were also forced to consider which features were best suited to each respective platform, and thus which were to take priority in our development process. For example, we decided that the most important web features were group management and bulk data entry, because both of these features were more convenient to do on a computer than they are on a phone. Some of the mobile-specific features were playing pitches and search. We assumed that the need to play a starting pitch on a desktop computer would be almost non-existent, as very few singers would be holding a laptop during rehearsal or a performance. Also, the need to quickly access a specific song via search was probably a rehearsal activity, and thus would be done primarily on mobile.

Of course, some of the technical milestones were dependent on those preceding. For example, there was no way to work on setlists before we could add songs, and group management required login and setup to be working perfectly. Luckily, our original timeline accounted for most of these dependencies and proved to be a good guideline for our development process. We did have a few instances where we misestimated the amount of time and work a certain feature would take. Fortunately for us, our overestimates largely correlated with our underestimates, allowing us to stay on track overall even if we fell slightly behind for a week.

Design

We started working on Pitchslapp with a strong vision of our final product in mind. This vision is what made our design process successful and robust over the course of the semester. Because two of our team members are part of Princeton a cappella groups, it was easy to envision the core features of our product in use by groups at Princeton. Additionally, designing Pitchslapp with real groups in mind allowed us to prioritize what we felt were the most important and crucial features for an MVP.

We spent a significant amount of time at the beginning of our process designing the structure of our database to align with the core functionality of the app and website. This early time investment proved to be invaluable over and over again as we were able to add new features without reworking the entire data structure. The robust data structure design also helped us to ensure that the mobile app and the website synced together seamlessly and worked in tandem from the same database.

One design issue we encountered was related to our implementation of the web application. We did not make a strong enough design plan for either the framework or the UI, and thus were left without much guidance as we developed the web interface throughout the semester. Better planning in this case would've been beneficial.

Testing

During the development process, most of our testing involved simple checks of functionality. Testing on this scale was important because it allowed us to move at a fast pace while assuring that the implementation of each feature was viable. Most of these tests consisted of creating or manipulating test data, which was centered around our test user "Phil." Phil was part of the a cappella group called "Phil's group" at a school called Phil's Academy. Phil's group had a wide ranging repertoire of about 20 songs at any given point. These songs were accompanied by the appropriate metadata and were combined into several setlists. We used Phil to test both app and web functionalities as they were added, and to ensure that the two platforms functioned together appropriately (live update, overwriting, etc.).

One test in particular allowed us to catch and correct a major flaw. When we first developed the sheet music integration functionality, we used Dropbox on both mobile and web. Multiple tests of this feature made us realize that on mobile, the Dropbox link to the sheet music expired after 4 hours. Because we caught this issue early, we were

able to switch from link to preview mode within the Dropbox extension and eliminate the expiration problem.

After all of our core functionality was in place, we extended our testing to encompass corner cases. Some of the cases we examined were related to overwrites, group management, login validity, and incompatible Dropbox files.

The final stage of our testing involved letting a real Princeton a cappella group try out our app. The Footnotes, of which Zach is a member, got to use Pitchslapp during a few rehearsals. This simple test was designed to make sure that the app didn't crash under normal use, and to ensure that Pitchslapp's usability was similar to our estimates. The Footnotes' Music Director used the website to add sheet music to the database, and then used the mobile app to play pitches and look up sheet music during rehearsal. Some members also used Pitchslapp to put together a setlist for an upcoming gig. We were happy to see that this first iteration of user testing not only avoided errors and crashes but also got people excited about using the Pitchslapp in the future!

Lessons

One of the biggest lessons we learned was to *plan.* We certainly could have spent more time finding appropriate integrations and plugins to use, particularly on Web. There are two ways we learned that we should have done more research in the beginning. First, we began by jumping into frameworks that sounded cool. We completed a React tutorial before even thinking about why we wanted to use React. Spending the time up front to choose an appropriate framework is the best way to not waste time later on in the process. Second, for some of our features, we didn't spend enough time looking for existing frameworks so we assumed they wouldn't work or didn't exist. For instance, Andrew tried to create his own accordion class to display extendable song info. However, this took a long time to create and didn't end up looking as nice as the basic JQuery accordion. We also thought about trying to create our own file management system for the PDFs of songs, but this would have been extremely difficult. We would have had to set up a separate, more open-ended cloud storage system, fetch files from said system, and figure out how uploading and downloading would work with its associated edge cases. The best solution was using the Dropbox integration to store the files. Many people already have Dropbox accounts and many a cappella groups already store their music PDFs in a Dropbox account, so this was the easiest solution for both our developers and users.

Another lesson was to *always pull at the beginning of your working session and push at the end.* We had so many merge conflicts at the beginning and git issues that were huge wastes of time. In a couple of cases, we even had to revert to a previous commit, thus losing the work we had done in the most recent commit.

Could've, Should've, Would've

Although we think the project went very well, we certainly have a few regrets and lessons learned. Most of these regrets have to do with not enough planning before we actually started producing the applications.

The biggest regret was not planning the website more. Because it was always meant to be supplementary to the app, we didn't spend as much time on it as we did on the app. Although we learned Facebook's ReactJS framework in the beginning semester, when we thought about the features that we needed and what we wanted the website to look like, we decided that it wasn't necessary. This seemed like a good decision in the beginning, but as we tried to make the website more visually appealing and responsive to users at the end of the semester, it was too late to go back to ReactJS. If we had drawn out the actual interfaces of the web application prior to beginning to write the code, we might have seen that plain JS/HTML/CSS wasn't the best choice for making our website ultra user-friendly. Other options such as Bootstrap should have been considered as well.

Asking members of a cappella groups at Princeton more about what the most important features would be for them would have been a great way to choose which features to implement. We especially regret not including one feature that wouldn't have been extremely difficult and that many users say they wish it had. At Princeton, there is a governing body of the a cappella group called AcaPrez. A cappella groups associated with AcaPrez cannot have arrangements of the same song. One of the groups in AcaPrez, the Wildcats, are responsible for managing song permissions. If a group wants to claim a song to arrange, they have to email the Wildcats to see if it's already claimed by another group and, if not, claim it themselves. This process could have been automated in Pitchslapp such that no group would have to manage this and communicate back and forth. There were many features that we would have liked to implement, but we knew even in the beginning that we wouldn't be able to implement them all. We asked people in a cappella groups whether they thought they would use an app that acted as an a cappella repertoire management system and got many positive responses. However, we could and should have been more detailed and methodical in

our “market research.” Letting people rank possible features by their usefulness would have been a great way to prune down our idea to the best and most useful features. I think the AcaPrez song calling feature would have certainly ranked higher than some of the features that we did implement. However, this still remains a great opportunity for future expansion.

There also should have been more communication between the mobile and web app development team members. For a while in the middle of the semester, the two subsets had worked making different assumptions about the (shared) database. This wasn’t a huge issue, but planning out the database together and making some sort of shared document containing the data structure plan could have saved some time.

This paper represents our own work in accordance with University regulations.
Andrew Hunt, Caroline Congdon, Zachary Stecker