Yelp Project

Randyll Bearer, Raj Patel, and Zachary Barlotta

Yelp PoC Dean Thompson and Liaison Steve Blass

12/8/2017

**Our Assigned Project and Initial Research**

For our Capstone project, we chose to work on the project proposed by the local Yelp team. Taken directly from the project proposal, our project for this semester was to proceed with “building a voice UI against [Yelp’s] public API.” The project proposal proposal itself was rather open-ended, as the Yelp team primarily wanted to see what direction we would take it in and how exactly we would decide to tackle the issue. This left us with a lot of options, which isn’t always a good thing as we were left with no clear direction. Thus, the first steps we took were to figure out which technologies we would be using throughout the entire semester. We also had to figure out which tools we would be using to create such a voice UI as well as designing basic conversation flows. So, we used the first sprint to conduct our research into these technologies and tools so we would be prepared to tackle the project later on.

Some of the technologies and tools we looked into included Amazon’s Alexa, Apple’s Siri, and Google’s Google Home. Since none of us had a device that uses Siri, we decided to eliminate Siri early on in our research. When we looked into Amazon’s Alexa, we found out that their simulation does not support voice input. Thus, we decided to eliminate Amazon’s Alexa from our list of possible technologies to complete this project. Thus, after looking into install base and ease of programming, we decided we would fulfill the project proposal by designing for the Google home device and use its accompanying tools (a la firebase/nodejs).

**Our Plan**

We would program our fulfillment in Javascript using node.js as its client library allow for easy communication with JSON objects which the Google Home uses to communicate. Google Home operates on Actions, which are programmable skills that can be run on any device with Google Assistant. These skills can be programmed in Javascript using the node.js framework. We would use DialogFlow, Google’s native natural language processing tool, to handle voice inputs, as well as handle conversation branching. Using JS, we can program the specific functions to match our user stories.Each function typically works by drawing input from the user’s voice command. Before the input command is processed, the program retrieves the supported list of categories Yelp utilizes using a series of REST operations on the Yelp Fusion API. Now, depending on if the input contains a certain key phrase, which Dialogflow defines as an “entity”, the program will take this phrase, and add it as a search parameter to a JSON object. The program will then prompt Google Assistant to ask the user for their location. Once the permission is given, the location coordinates of the user is added to the JSON object as a required search parameter. Finally a HTTP GET call is sent using this JSON object as the body. The response is returned as a JSON object, containing a wide variety of information for several businesses that match the search parameters as closely as possible. We would initially focus on planning four major features of functionality:

- DirectSearch(): This function would take in a business’ name as an entity, and search the nearby area for that particular business. If the business was not found, it would return the closest match.

- vagueSearch(): This function would take in a particular category of food, i.e. “Italian”, “Mexican”, “Burgers”, as an entity, and use that to search for nearby businesses that Yelp categorizes under that category.

- randomSearch(): This function would simply take the user’s coordinates as an input, does the search using an umbrella term “restaurants” as a category. The search returns a random collection of businesses that are nearby to the user.

- UtilitySearch(): This function would take in a business’ name as a search parameter, as well as a certain “utility” entity, i.e. wait time, parking, delivery options, etc. The appropriate utility information for the particular business would be returned.

We would use Google’s Firebase service to deploy our fulfillment and functions to a cloud server so that the devices could actually run our code. Firebase also proved to be a useful tool in debugging our errors. By the time of our final sprint, we would have documentation to provide the Yelp team with in the case that they ever wanted to expand upon what we had worked on.

**Collaboration with Yelp team**

For our first meeting with Yelp, we planned to discuss our plan for the project as well as the technologies we decided to use to accomplish our plan. At our first meeting with the Yelp team, we presented this plan to all the members of the team, not just Dean. During the meeting, we provided our reasoning for why we chose the platform we did, as well as explained the features we were intending to accomplish. The team also provided their assent as they found our reasoning reasonable Thus, it was agreed that we would meet in person every two weeks (the length of our sprints), and that we would be assigned a liaison within the team, namely Steve Blass. Throughout development, Steve was able to aid us in any roadblocks or questions related to overall design by providing his professional insight and advice through a dedicated Slack channel. We did not consult him for the actual low-level implementation, as we thought it was best to not bother him with such details.

Collaboration with the Yelp team went smoothly throughout the semester. Since they allowed us so much freedom to tackle the problems as we saw fit, there was no real confusion over requirements for the project. Rather, they acted as a representative of a prospective customer/user, suggesting functionality and providing advice on how to better smooth our conversation flows. Their suggestions helped us improve on what we had done to the point of the meetings as well as help us develop our project in the future. All things considered, collaboration with the Yelp team was a very positive experience for all of us, and we can use our experiences while collaborating with them in the future.

**What We Accomplished**

As we come upon the close of the semester, with our last meeting with the Yelp team just around the corner next week, it is now time for us to show and tell what we have been working on. Our final product has taken the form of a Google Assistant simulation which emulates how our software would be executed on an actual Google Home device. We have not published our Action on the Google Play Store yet for multiple reasons, namely we wanted to leave it for the Yelp team to double check the quality of our software and handle all legal matters.

Our simulation has implementations for three of the four major search features we outlined earlier in this paper, unfortunately UtilitySearch() had to be cut due to deadline restraints. We knew going into this project that we most likely would not have been able to implement everything we had planned, and as a wise Software Engineering professor once taught us “Software Engineering is the study of trade-offs”, so we do not necessarily view the exclusion of this feature as a failure per se. We cut UtilitySearch() specifically because the stakeholders at Yelp thought it the least important of the features as Yelp is primarily used as a restaurant search/review service.

Our final software includes the ability to search for general categories, specific businesses, and provide suggested restaurants. The conversation flow is more on the basic side (no ability yet to ask the app to return to an arbitrary state), but we do have support for the app to be able to repeat specific strings of information as no user would be able to memorize everything if listed all at once without text support.

**Our Development Process**

As required by the course, our primary methodology was Agile/Scrum with sprints of two week periods. Role and title of Scrum Master rotated between the three of us on a sprint-by-sprint basis, and responsibilities of the Scrum Master were primarily communication with the Yelp team who we remained in contact with through both emails and a Slack channel.

Tasks for each sprint were decided upon at each meeting we had with the Yelp team, as we wanted to hear any of the suggestions they may have had as well as update them on our progress and provide to them our reasoning for what we thought should be completed next. Once the meetings had concluded, it was then the role of the new Scrum Master to divy up the tasks as he saw fit amongst the group.

Due to the nature of Dialogflow development, we all had to work with the same Dialogflow intents as each other simultaneously (i.e. no version control for intents). This could make things difficult to program and test when someone else was changing the intent structure at the same time you were adding the fullfilment. Very quickly we realized that we had to add the responsibility to divide intent development between us to the job of the Scrum Master.

By the end of every sprint, in accordance with the Agile/Scrum ideology, we guaranteed that we would have some form of a working product up on our Github Repository and a working (if not complete) simulation.

**Noteworthy Difficulties**

During the second sprint of our project, Google had officially rebranded their development suite for building conversational interfaces. It was previously known as “API.ai” to “Dialogflow”. For a brief period, several libraries had not been updated, and it caused our functions to not run properly. However this was only a temporary inconvenience, as we were able to complete the goals for our sprint in time.

Another unexpected issue we ran into was having functions in a remote function hosting environment communicate with each other. In our program, Google Assistant requires the user’s permission before retrieving and using their location to perform the actions. This requires a separate function that is specifically designated to retrieving the user’s location coordinates. Once these coordinates are received, those coordinates must be passed into the function that will perform on of the primary actions. The documentation was descriptive how to get the location, but does not entirely say how we can actually use it with our functions. When we had initially implemented the permission procedure, our program would retrieve the coordinates (once permission was given) and the program would then return an empty text-to-speech response, completely disregarding the functions. The fix ended up being a little more complex than using a global variable, as it involved restructuring the primary method logic to include wrap methods.

**Our Reflections on the Capstone Course**

Randy: I definitely learned a lot about not only how to program in javascript/deploy cloud functions/manage language parsing and conversation flows, but also how to become more comfortable with talking to stakeholders. The very first meeting with Yelp we had prepared to talk only with Dean, yet we were quickly ushered into a roundtable discussion with the entire team. Fortunately I understood our plan well enough to manage my way through it without embarrassing myself, but after that initial shock of being dropped immediately into things subsequent meetings felt a lot more comfortable. So as for what I consider the best thing about this Capstone course, it was without a doubt all the experience I received communicating with stakeholders.

As for what I would have to consider the worst part of the capstone course, it would be the constant sprint deadlines in conjunction with other coursework deadlines, especially around midterms and final weeks. For only taking a twelve-credit semester I felt incredibly overwhelmed for weeks at a time where I would be working on something or other non-stop. If I had taken this course during an eighteen-credit semester I would have been seriously burned out/unable to complete everything. I understand that this was partially due to me setting high expectations for myself and taking on perhaps more work for a sprint / setting too high of a scope for this project, but as a soon-to-be professional who wants to set high standards for myself I couldn’t see myself doing much less. If I were to suggest any changes to the course it would be to update the course description to more accurately reflect the amount of time a student may work on their Capstone project per week from ten hours to fifteen or twenty.

2. Zach: Overall, the Capstone experience has been a positive one for me. I never had real experience working with stakeholders before, so I did not know what to expect heading into our first meeting. When we had our first meeting, I was not expecting them to bring the entire team into the meeting. Looking back on that meeting, I’m glad they brought everyone in because that one meeting will help me in the future when it comes to round table discussions with a medium to large sized team.

As for the coding portion of the project, I knew a little bit of JavaScript before doing this project. Learning more about JavaScript was a positive experience for me because I learned more about a language I knew little about while working on a project I enjoyed working on. Another thing I enjoyed learning for this project was DialogFlow. I have never dealt with anything like DialogFlow before, so learning how it works was a challenge for me. However, once I learned how it worked, I really enjoyed working with it. Overall, the implementation portion of this project was fun for me, and I really enjoyed learning new languages and frameworks.

The hardest thing for me to get used to with this project was meeting the deadlines for each sprint. Sometimes, I would have other project deadlines coincide with the end of a sprint. This made it hard for me to get everything I wanted to get done for that specific sprint. Even though I may have had trouble keeping up with my work for deadlines, I feel like getting that out of the way now will help me in the real world because I know my threshold for how much work I can handle per sprint. Overall, there are always things I can improve on, and setting more realistic goals for my sprints is one of those things.

All things considered, I had a lot of fun working on this project with the Yelp team. The project itself was a lot of fun to work on, and I am glad I selected this project to work on. Learning different frameworks and languages was very useful to me, and I am glad I took the time to learn them when I had the chance to. All in all, I will take the experiences I gained while working on this project into the future with me and beyond.

3. Raj: I appreciated the opportunity for work with an Industry, and well-known one at that. I learned what it’s like to actually work with stakeholders in an Agile Scrum development process. This was a profound experience to me because I have been used to getting a project and having it completely working by either the end of the month or the end of the semester with little no “checkups” with my professor. Having to report to a stakeholder every two weeks was a new, eye-opening experience that let me understand what it means to have deadlines that affected not just my own work but the needs of a stakeholder. What I appreciated the most, however, was the free reign we had over developing the project. Sure we had a few guidelines that were to be met by the stakeholder, but our implementation decisions as well as our tech stack were left completely to our own imagination. There was no correct way of developing this project, and that provide me the opportunity to expand on my skills in JavaScript and explore frameworks which I ended up loving to code in towards the end of the project.