

Intelligent Search

Sidhant Chitkara, Alexander Meyer, Apoorva Parmar,
Yash Pujara, James Shao, Michael Vieck

What went well?

This sprint went much more smoothly compared to Sprint 1. Last sprint, we were pressed for time to finish many features and had an intense last day. This sprint, while we still worked to the last day, we were not as pressed to complete user stories. Overall, we managed to complete our user stories in a timely manner.

While the CityBus web-service had to be written in Visual Basic, a language that both Alex and I were unfamiliar with, we did not have much trouble creating our own calls. We ended up not having to spend as much time working on the CityBus web-service as planned. The live and static data scripts ended up working well on the first few tries, allowing us to help the front-end team get started with their user stories without a huge delay. Finally, communicating from backend to front-end ran a lot smoother this time. Most of the necessary backend calls were written before they were needed by the front-end. Because of this head start, we were able to better support smaller details that the front-end team needed. Since, we started integration early this time, it allowed us to easily change the calls to better meet what they needed allowing us on the back-end to update and adjust as needed.

We found an amazing framework called NgMap, which allowed google maps objects to be created simply by creating a new block in our html code. By abstracting the Google API calls to this framework, we were able to speed up production significantly. It allowed us to focus on our existing knowledge (AngularJS), while still incorporating the map objects we needed to create the bus page.

What did not go well?

The large amount of information returned by CityBus' live API calls means that it takes a long time to load the page. It might be more feasible to ask for specific pieces of information and modify the call, or speed up the call by means of making the query more efficient.

A major blocker this sprint was plotting bus routes on the front-end. Google Maps API allows for automatic path drawing that connects different points through their fastest routes. Purdue's buses don't always take the fastest route between stops (to service different people) and as a result, automatic path drawing was not an option for us. Upon inspecting code of other bus apps, we found that they hand-draw the routes, which eventually also helps with

animating the buses along the hand-drawn line. With the manual route mapping, we only serviced one full bus route this sprint, with the plans to add further buses in sprint 3.

Our initial plan was to create a slider for the calories functionality. The user could potentially create a high and low bound, to display food that satisfies these ranges, but the HTML libraries we are using made the slider hard to implement. Right now, we support single bound searches, but no modification after the user has searched. We hope to finish the slider and allow dynamic range modification in sprint 3.

While it is important to have functionality as a main priority, CSS is also an issue with some aspects of our app, and it would definitely help the user with readability and usability if our search results and bus app were cleaner. We pushed CSS again to sprint 3 in order to maximize functionality built into the app, but at this point it will become more of a necessity when we present at the end of sprint 3.

On the backend, we did not realize how much work would have to be done for setting up routing. The shortest path algorithm had many small nuances that we had not anticipated. This led to these user stories getting delayed until the next sprint. Additionally, we discovered that due to the size of the static data database, certain calls were taking a bit too long to execute. We ended up having to work on improving database calls to help keep performance up.

Finally, we realized that when the server restarts, we did not have any functionality in place to restart running scripts. This could end up being something that comes up in the future and will be something to consider moving forwards.

What can we improve on?

For the next sprint, there are three things that we will need to work on. The first is finding a way to split work more evenly among team members. This is especially true for sprint 3 because the majority of the team will be working on the same map page. We will have to have multiple group sessions and good distribution of features for members to work on. Additionally, because so many people will be working on the same page, we will have to improve our github push frequency and ability to resolve merge conflicts. I anticipate that this will be an issue during this sprint because previously, most members worked on separate pages and features. Hopefully, good communication between team members, along with knowing what everyone else is working on, will help alleviate this concern before it becomes a real issue.

Another tentative issue that will have to be looked at is the possibility of opening the server so that it can be accessed from any network outside of PAL. If this is possible, we will have to refactor our backend codebase so that we access passwords and other important data through a properties file, instead of through hard-code as it is implemented now.

