

Semester: Fall 2018

Progress made this semester:

From the application prototype created in the Spring of 2018, the homescreen featured a map display along with three buttons, a camera button, a send button and a severity button.

According to the demands of the stakeholders, which is the city of West Lafayette, the Fall 2018 team has decided to get rid of the severity option as well as give the application a new user interface.

We plan that the application has a search location button, much like the Uber app, in case the user wished to input the location manually. In either case, the location would be sent to the database using the Place Picker API that we use in our app. We are trying to structure the application such that a user can take a picture of the issue, that takes the user to a different page, where they can view the picture and select the type of issue (pothole, traffic light, trip hazard or other), add a short description (which is optional) and finally hit submit.

This is shown in order in the three images below.

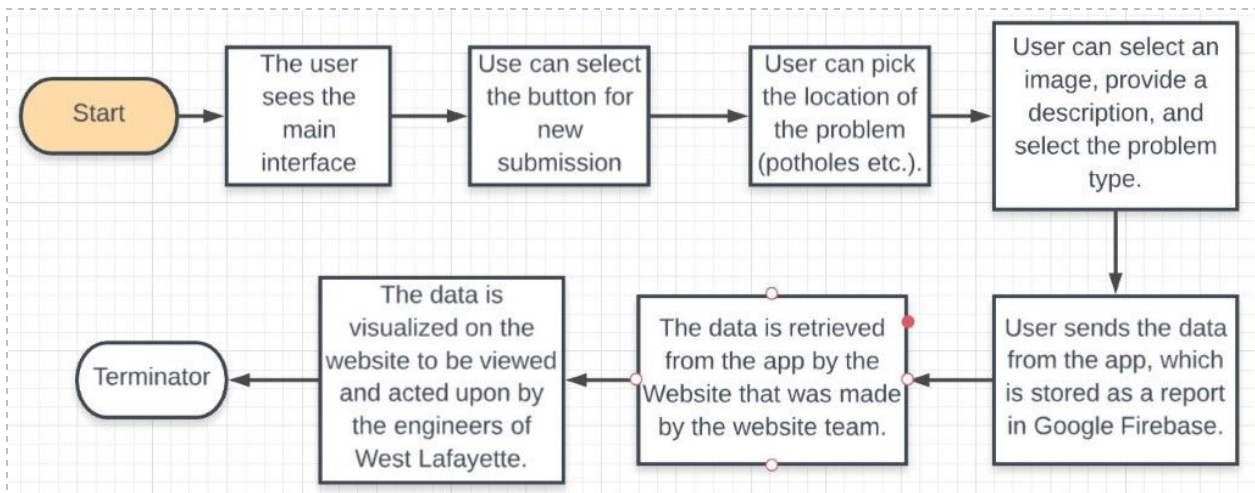
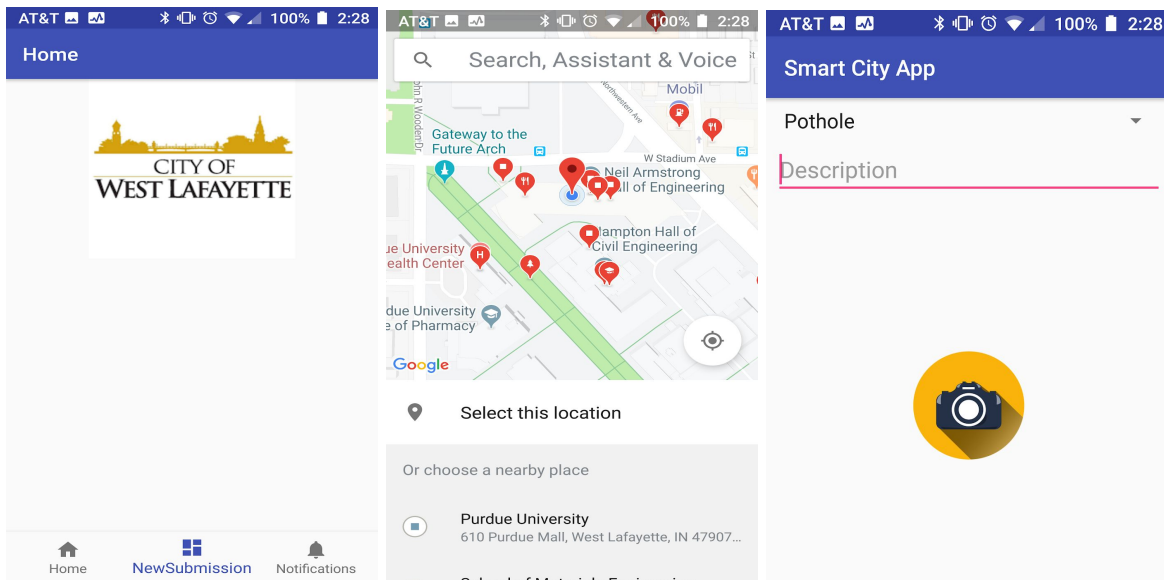
The application from last semester crashed when the user tried to take a picture as the team earlier did not have time to link google's server, Firebase with the app. This semester's technical sub team has managed to fix that. This required a significant amount of time learning android studio and getting familiarized with the code from last semester. The team has finally found a way to take a picture, linked Firebase to the application (where the data collected is going to be stored) and are also currently working on implementing the options menu where the user can select the type of issue. It is a challenging task for the team as the members are new to using android studio and to app development in general. The design for the app has been selected after researching other applications and picking out the best features from the applications as per the demands of the stakeholders.

The team has also planned to switch the background color of the application to blue, which is the color of the website that the visualization team is creating. This would enhance user experience in using the application for reporting any issue.

The application team has made good progress with documenting and developing the app. The team has been divided into about 3 sub teams where each person is assigned a task based on their role and strengths. The progress for the app till date is as follows:

1. The map is dynamic and working everywhere and can search location using search bar or voice
2. The search bar is linked to the google place picker API
3. When you select a button/option (such as traffic light, pothole etc.), a function is called that would help categorize the data based on the respective issue on the database.
4. The picture taken successfully uploads to Firebase with the image as an encoded string and other parameters such as type of issue, a status and a description.
5. The manual is completed and the goals for future development have been set.

Current application interface:

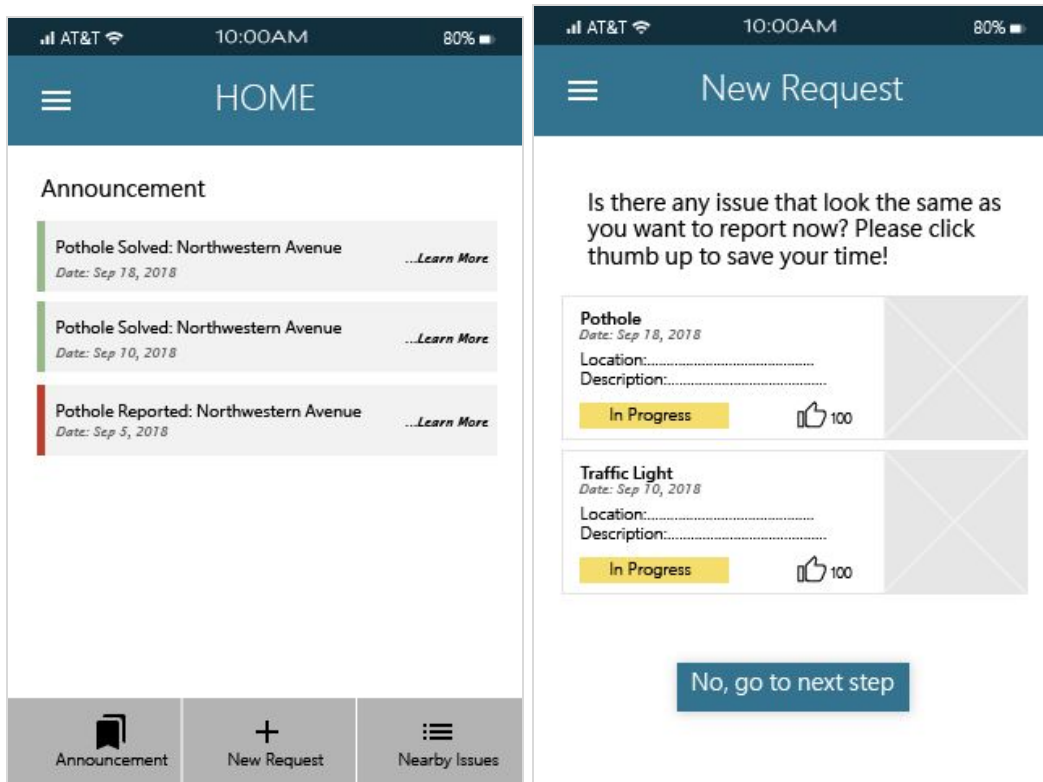


Above: Current software development life cycle for the app

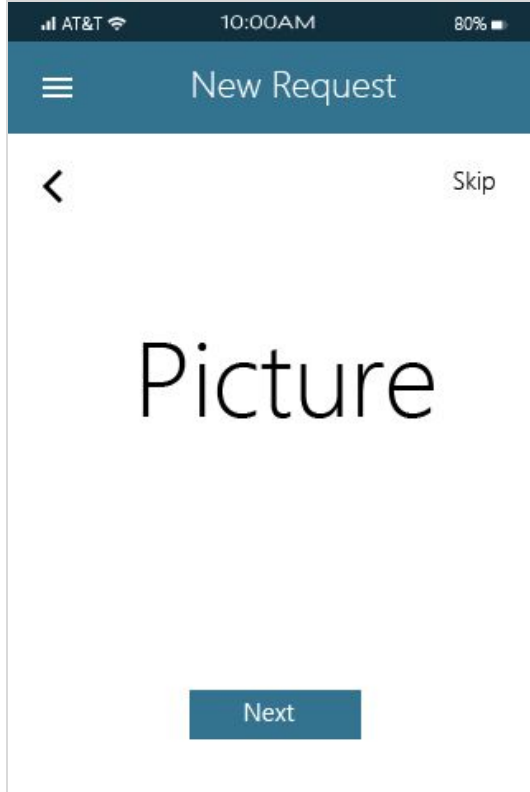
Future goals:

- 1* Design solution for the issue of repeated reports: The addition of a like button feature was a possibility
2. * Establish reporting database that sends info from the website to the application
3. * Implement new graphic user interface based on specifications given by the project partner and based on advice from design reviewers
4. * Work on adding notifications to the application to alert citizens when a new problem has occurred or when a problem has been resolved
5. * Implement tutorial links to the application based on specifications given by the project partner: Bike safety videos or roundabout tutorial videos.
6. User choice of mode of transportation - walking/biking/car
7. Let users see a nearby issue and get a notification when it is resolved.
8. Add a border that separates Purdue's campus with the rest of West Lafayette much like the "Bird" app that a lot of Purdue students use!

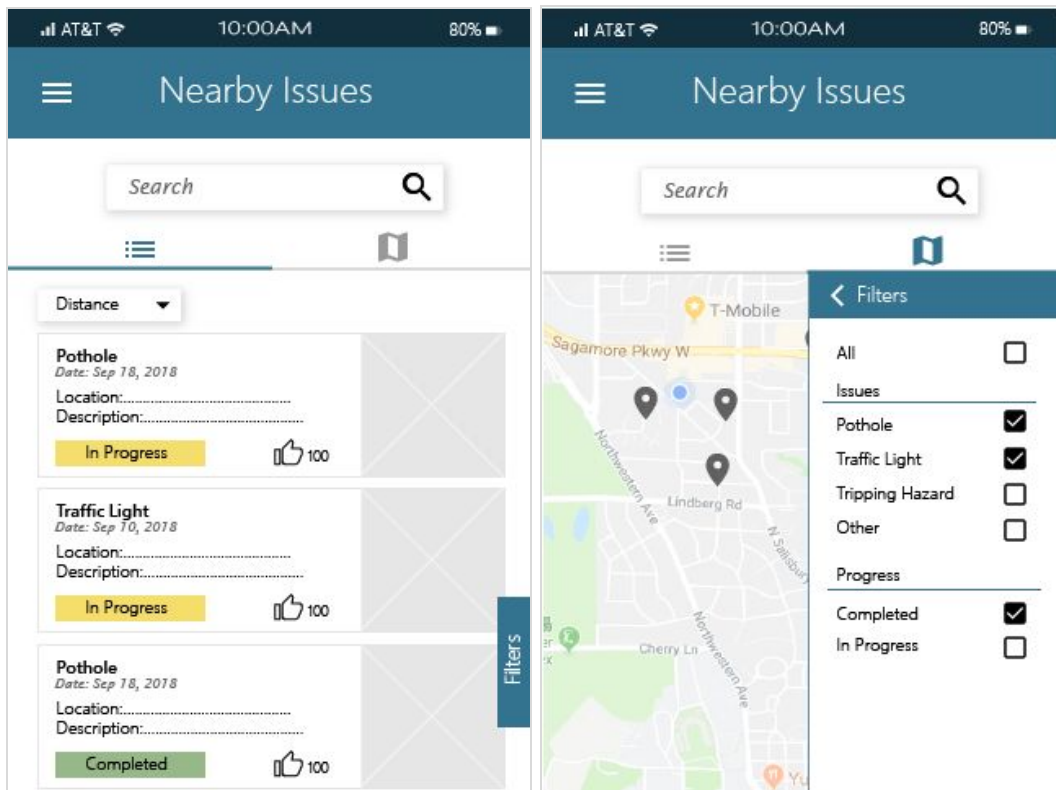
Our vision of the app in the future:



On the homepage, users will see announcements from government. There are two categories, problems reported and solved, which are labeled for green and red color.



Offer drop-down menu for users to choose issue, including Pothole, Traffic Light, Tripping Hazard, and Other.



Provide drop down menu, including Distance, Progress (In progress/ completed), Date, Issues (Name ascending). Users can change orders to easily find what they want to see.

Progress Made from this semester:

1. All of the code has been fully transferred from Amazon Web Services (AWS) to Google Firebase.
2. The Google Maps API key has been updated and should now be functional.
3. The website now has a table that is linked to a database, which can be updated when a new report has been sent to the database and when the user refreshes the page.
4. There is an initial version of the filtering algorithm that would be able to hide rows of a table that do not fit under a certain criteria (i.e. only show the rows with “submitted” reports.).
5. There is an initial version of the code to overlay markers onto the embedded Google Maps to show the location of the issue.

Project Parameters:

In the midterm design review, the City of West Lafayette representatives wants a web page that lists the main issue, which is potholes, but they also want an issue dropdown menu that is flexible (i.e. it will contain more issues such as traffic lights, etc.).

There is also a pertinent need for the City of West Lafayette Department of Engineers to be able to read and visually see the data. In response to this, we would be implementing both a reverse-geolocation API and an algorithm to display the markers on the embedded Google Maps.

In-depth explanation of progress:

At the very beginning of the semester, the website only had some dropdown menus and a static map to start (refer Figure 1)

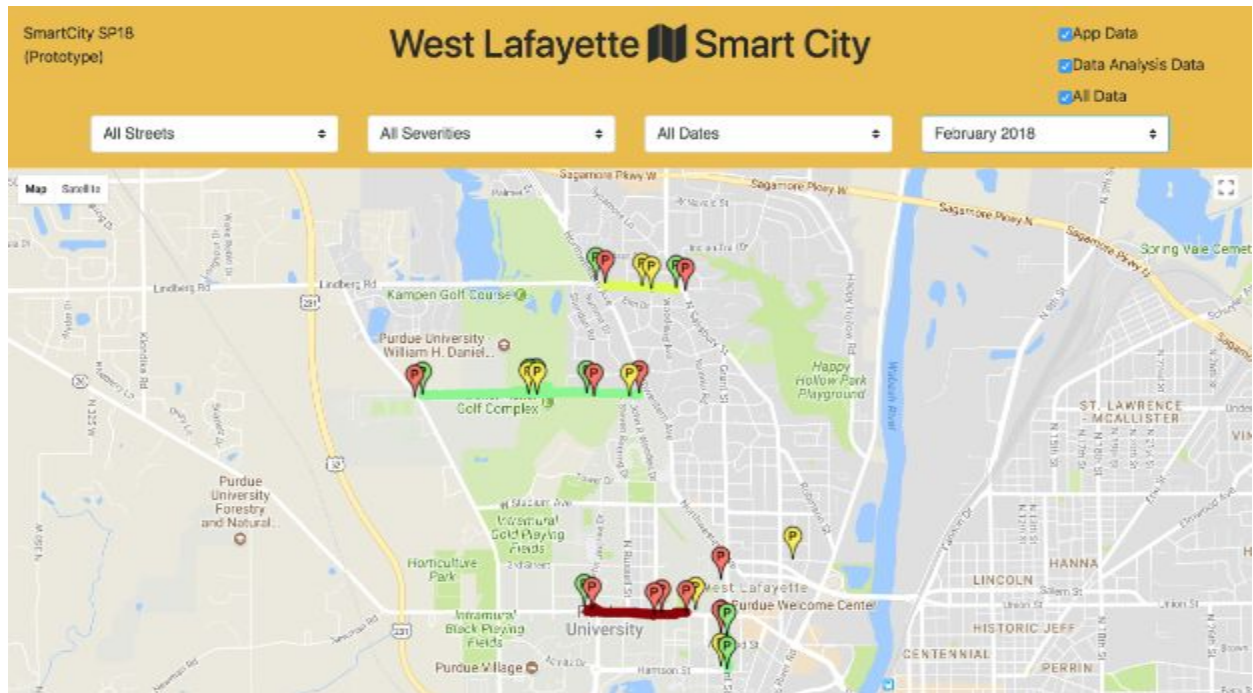


Figure 1: Website Prototype from end of Spring 2018 semester.

The map and the map markers that you see are static images. Our team did not find any functionality with dropdown menus or checkboxes at the top of the website.

From this point-on, it is notable that none of us had any major experience with either Javascript or HTML. Our team then set out to scour through the code as much as possible so that we could understand each part's functionality. Afterwards, we experimented with some of the new features that we would implement, like creating a table at the bottom of the page. However, when it came time to create a table that would connect to the database, we had a few setbacks.

First, the table could not be fully coded in HTML because Javascript is more capable at retrieving data. Javascript also can create HTML elements (like creating a whole table in Javascript that is equivalent to using the "table" tag in HTML).

Second, the data for the image was entirely in base64. For some reason, there were some minor difficulties in converting from base64 to displaying an actual image, but after some trial and error, it was working. (Refer to the last column of the Reports table in Figure 2.)

Third, as we knew the design reviewers would be attentive to detail, the original "table" that displayed the data was originally just a series of tables created for each row of data. What this meant was that each cell in each row would have a different width and the whole table would look misaligned. After tinkering with the code for a while, the way this table now works is that it consists of 2 main tables. The first table would contain the dropdown filter menus and the second table would have the rest of the data. (Refer to top row of the Reports table in Figure 2.)

While this was going on, we started implementing overlaying dynamic, which means taking the data from the database rather than hard coding them in. But, there was a major flaw with this: the map would briefly show up for a split second and then disappear again. We suspect that there is something with the way we implemented our code when we had to connect

the current map to our code. In Figure 2, there is a sizable gray space where the map is supposed to be.

Purdue EPICS **West Lafayette Smart City** App Data Data Analysis Data All Data

Reports

Filter

Status	From Date:	Issue	Location1	Location2	Image	Description
submitted	2018.11.05.11.11.21	Pothole	40.43102250000001	-86.91506640624999		newer
submitted	2018.11.07.11.13.01	Pothole	40.43094249999999	-86.91513671874999		newTest
submitted	2018.11.07.11.14.02	Tripping Hazard	40.430947499999995	-86.91510546875		new Submission 11:15 pm
submitted	2018.11.14.11.27.28	Pothole	40.43094249999999	-86.91517578125	no image	
submitted	2018.11.14.11.32.10	Pothole	40.43095250000002	-86.91515234375001	no image	hello
submitted	2018.11.14.11.33.17	Pothole	40.43095250000002	-86.91515234375001	no image	
submitted	2018.11.14.11.55.21	Pothole	40.43095250000002	-86.91517578125	no image	
submitted	2018.11.14.14.28.36	Pothole	40.431037499999995	-86.91509765625	no image	

Figure 2: Picture of the website as of the writing of this design document (11/30/2018)

Goals for the end of the semester:

1. Get the filtering algorithm functional for the table to sort by status,date,and issue.
2. Get the reverse-geolocation code working so that longitude and latitude (location1 and location2 in the table in Figure 2) would not need to be displayed.
3. Get the markers overlayed onto the map using the given longitude and latitude coordinates.
4. Resize the first row (with the dropdown menus) so that the widths align with the rest of the table.