Design Document

Team: Smart City

Date: 04/03/2019

1 Table of Contents

2 Revision History

Date	Author	Revisions Made
3/4/19	Owen Price Financial Officer	● All categories were updated to fit Spring 2019 and Archive
2/12/19	Miguel Kulisic: Team Leader	● Website Progress ● App Progress
9/28/18	Jason Cao: Fall 2018 Website Development Team	● Website Progression
02/16/17	Eric Jin Wook Choi: SP17 Project Manager	● Design Document Creation ● Project Charter ● Semester Information
04/14/17	Eric Jin Wook Choi	● Current Design
9/27/2017	Wesley Sawyer, Grant Hilbert: FA17 Hardware Team	● Fall 2017 End-Semester update
10/6/17	Eric Jin Wook Choi: FA17 Project Manager	● Fall 2017 End-Semester update
11/29/2017	Eric Jin Wook Choi	● Fall 2017 End-Semester Update
11/30/2017	Hardi Sura	● Fall 2017 End-Semester Update
02/16/2018	Erika Lai Ting Lin: SP18 Project Manager	● Project Motivation ● Addition of Links ● Outcomes/Deliverables ● Semester Information
04/05/2018	Erika Lai Ting Lin	● Reorganization of Design Document
04/19/2018	Erika Lai Ting Lin	● Final Design Review Comments Reflections
4/28/2018	Erika Lai Ting Lin Romita Biswas Kalpan Jasani Kartik Mittal Ayuub Jose	● End of Semester Updates Contour ● Tracking

3 Design Status

Phase 1: Project Identification	**Status:** *Completed* **Semester:** *Spring 2017*
Phase 2: Specification Development	**Status:** *Completed* **Semester:** *Spring 2017*
Phase 3: Conceptual Design	**Status:** *Completed* **Semester:** *Spring 2017*
Phase 4: Detailed Design	**Status:** *In Progress* **Semester:** *Spring 2019*
Phase 5: Delivery	**Status:** *To be done** **Semester:**
Phase 6: Service / Maintenance	**Status:** *To be done** **Semester:**

4 Project Charter

4.1 Description of the Community Partner

The project partner for the Purdue Engineering Projects In Community Service (EPICS) Smart Cities Team is the City of West Lafayette Department of Engineering ("The City"), whose primary responsibilities are:

1.) To evaluate engineering solutions.
2.) To provide recommendations to the Mayor, other city departments, boards, and commissions regarding private and public works throughout the city[1].

The City has requested EPICS to develop an integrated system that will detect, observe, analyze, and suggest fixture recommendations for "potholes", "traffic hazards", and "tripping hazards" within City limits. Potholes, are road defects where road materials erode and form holes due to erosion from weather and vehicle travel rates, etc.[2][3][4].

Potholes may oftentimes result in harm to the community ranging from pedestrians to vehicle operators by causing vehicle damages as well as putting lives at risk. "Out of 33,000 traffic fatalities per year, one-third involve poor road conditions" and United States (U.S.) motorists spend approximately "$67 billion a year in extra for repairs and operating costs due to the poor conditions of roads." To combat increasing frequency and severity of potholes, the City has requested EPICS to tackle the

[1] http://www.westlafayette.in.gov/engineering/
[2] https://www.fhwa.dot.gov/pavement/pub_details.cfm?id=139
[3] https://www.pothole.info/the-facts/
[4] http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7556304

problem through Smart City design. Tripping hazards could be a variety of things such as cracks and holes in the sidewalk, fallen tree limbs, or even large enough litter. Traffic hazards would include broken glass on the road, anything blocking traffic, or roadkill potentially. Car accidents should be immediately called to the police and/or fire department as those require immediate fixing.

The City has expressed concerns regarding ongoing City efforts to provide pothole fixtures within City limits; visual inspection is the primary source of detecting potholes through Pavement Surface Evaluation and Rating (PASER).[5] Marcus Smith, assistant city engineer and Smart City's primary contact with The City, provided general comments regarding The City's needs and user/stakeholder identification.[6][7] The City is seeking to improve its processes with pothole detection and location tracking. The currently implemented PASER and fixture schedules through the Department of Street & Sanitation require manual labor and do not always produce accurate results. The project partner's overall mission is to increase the efficiency of the road damage identification system within the city of West Lafayette. Smart City's end-deliverable is to develop a system that satisfies the City's needs and will be delivered to the Department of Engineering for implementation.

As a result of this design, the Department of Engineering will be able to recommend appropriate adjustments to the Mayor's office and other city departments in hopes of making the pothole detection process more effective through a cost efficient and reliable design. Ultimately, the project partner will be able to effectually detect potholes, traffic hazards, and tripping hazards, thus being able to effectively repair road and sidewalk damage, making roads and the city safer for the West Lafayette community.

4.2 Stakeholders

The stakeholders of this project include those who will be affected other than the project partner and those who have interest in the project's success. As this project focuses on a design specifically for the City, the primary user is the West Lafayette Department of Engineering. However, this pothole detection system will impact others as well:

- Primary Users
    1. The City of West Lafayette Department of Engineering ■ 3-5 city engineers with respective sub-department interns.
        - Provision: A platform that will help identify/located potholes such that the city is able to communicate a fixture schedule to the Department of Streets & Sanitation (secondary users).
    - ○ West Lafayette Community/General Public
        - ■ Provision: An application allowing the community to report road conditions throughout the city, i.e. pothole detection by public reachability.
- Secondary Users
    1. Other departments and employee, e.g. Streets & Sanitation, Internal, etc.

[5] http://www.in.gov/indot/2469.htm

[6] https://sharepoint.ecn.purdue.edu/epics/teams/smartcity/_layouts/15/WopiFrame.aspx?sourcedoc=/epics/teams/smartcity/Semester%20Documentation/Spring%202018/Project%20Partner%20Information/PP%20Questions%20and%2 [7] Answers.docx&action=default

- Primary Stakeholders:
    1. The City of West Lafayette Department of Engineering
    - Purdue University EPICS
    - West Lafayette Community and General Public

4.3 Project Objectives

4.3.1 Project Operations & Logistics

The Smart City EPICS Team is split into two sub-teams: Website and Application Development - both teams need to work collaboratively to achieve the overall goal. Each sub-team consists of 3-4 members directed by a Design Lead. The motivation for this project is the lack of efficient methods of pothole, traffic hazard, and tripping hazard detection in the city of West Lafayette, thus resulting in increased risk for the community. The Smart City Team will develop a website for city use to view pothole, traffic hazard, and tripping hazard location while also producing a smart-phone application accessible to the general public such that they are able to directly report potholes and issues throughout the city. The mission of the project partner "to develop an application for residents to submit issues to the city."[7] Smart City's design aims to satisfy the mission of the project partner while also achieving individual team goals. Each team has developed an independent design process according to their need-finding and project deliverable goals and specifications. In the overall scope, individual design processes merge towards the end-deliverable, but team dynamic, management, and procedure should retain individuality. The "EPICS Smart City Project Design Document" is compiled to illustrate the progress and design process across the whole team. The Design Document will encompass *all* of the work done in previous semesters and current semester, displaying the entire design process that has been iterated.

4.3.2 Project Motivation

Potholes, traffic hazards, and tripping hazards present a high risk for the community. The current approach the City of West Lafayette utilizes, described below, is not integrated and requires much human effort/interaction. Pothole repairs are unable to be completed all at once:

1. Vehicle and personal damage arising from poor road conditions
2. Need for identifying pothole, traffic hazard, and tripping hazard location: GPS coordinates; streets with most potholes
3. Make repair recommendations
    a. Engineering department recommends to Mayor's office
    b. Fixtures completed by Street & Sanitation department
    c. Repair crews are sent to the most critical sections to patch and address any and all damage

[7] https://sharepoint.ecn.purdue.edu/epics/teams/smartcity/_layouts/15/WopiFrame.aspx?sourcedoc=/epics/teams/smartcity/Semester%20Documentation/Spring%202018/Project%20Partner%20Information/PP%20Questions%20and%20Answers.docx&action=default

The City currently does "patchwork" fixture for potholes, i.e. filling potholes until a certain level is reached where the pothole can no longer be ignored. The City also does patchwork for an entire street until the total threshold of potholes on a given street cannot be neglected. We hope that our system can locate and archive pothole fixtures so that The City can have an accurate assessment of the pothole threshold.

4.3.3 Project Specification

Based on initial need-finding and background identification of users/stakeholders, Smart City has determined the following requirements and limitations for the project:

- **Objectives or goals:** The design will be cost-effective yet efficient, that the City is able to utilize with city employees with the goal of locating and quantifying potholes, traffic hazards, and tripping hazards. The system's results will be accessible to the city via a website, and an application will be accessible to the public to map potholes as well as report road issues to the city.
- **Constraints:** The design must be cost efficient yet produce an effective analysis as potholes throughout the city are frequently monitored. The design must also be easily accessible and adaptable, and the system cannot be drilled into city property for attachment.
- **Function**: A website and application will be developed to visualize the location of potholes, traffic hazards, tripping hazards, and for the general public to report additional road issues.
- **Implementations or means:** The design will be implemented by giving city employees logins and passwords and by having the citizens download the functioning app.

Smart City acknowledges that primary users have a mission of an easily operable system, thus concluding that the end-deliverable should depend on minimal technical specifications and should be easily usable for the appropriate audience.

4.3.4 Proposed Solution

Smart City proposes a solution for a website and app to be developed. Smart City's end-deliverable should be able to provide visualization and digital documentation of the location of potholes, traffic hazards, and tripping hazards and be able to communicate with appropriate technology and users/stakeholders for completeness.

The Website Team will be creating a website that will allow the City to receive feedback from the public regarding city issues. The Application Development Team will be creating an app compatible for both Android and iOS for the public to submit reports on potholes and other potential hazards.

4.4 Outcomes/Deliverables

The Smart City Team then takes the analyzed data and displays the location of potholes, traffic hazards, and tripping hazards on a map. The Application will allow the citizens to send reports, but also it will display to the public the progress of their requests.

4.4.1 Smart City Team

The Smart City team aims to develop a system to make identifying street issues easier with the use of crowdsourcing. The solution we have designed has two parts. The first part of the solution is an Android/IOS application that will allow the citizens of West Lafayette to quickly and efficiently report various types of issues they encounter in the streets instantaneously. The second part of

our solution is a website. This website will only be accessible to the city engineers and it will allow them visualize all the reports submitted by the citizens.

The information gathered by the App and displayed by the Website will allow city engineers to not only know where the issues are, but also view pictures of the problem and a small description send by the user. Thanks to have this information easily available, city engineers will be able to very quickly assess the severity of a certain issue and prioritize which issues should be fixed first. After fixing an issue, the city engineers can use the website to notify user of the app that action has been taken towards a certain report.
As a result of implementing this system, we hope that the city will be able to easily find and analyze issues as they arise on the streets and fix them before they either grow bigger of cause an accident. For Spring 2019, we hope to deliver a prototype of the system that has the basic functionality for testing.

4.4.2 Website Team

In Spring 2019, the Website Team has focused on improving the functionality and user interface of the Website. This team is working on the best ways for the city engineers to visualize the information provided by the App. In addition to this, the website needs to be able to send notifications to the server to notify user that action has been taken towards fixing a certain issue.

With this in mind, the team will deliver a finished prototype of the Website. This prototype will be able to read reports send by the user, display them on both a map and a table, and filter them based on a few parameters. This prototype will also be capable of editing reports on the server. Editing the reports on the website will act as notification for the App. This prototype will cover all the essential features the website needs to have in order to operate as intended.

In future semesters, we aim to add extra functionality in order to make the city engineers job a lot easier. Some of these features include, but are not limited to, creating a login page, allowing the user to create a priority queue and data fusion to avoid duplicate reports.

4.4.1 Application Team

In Spring 2019, The Application team has focused on both improving the functionality of the Android App and creating an IOS App. Development of the IOS application was started this semester in order to reach a bigger user base once the app launches. This team is working on making sending reports as easy and quick as possible. The main idea is that sending a report should not be a hassle.

By the end of Spring 2019, the Application Team will deliver two prototypes. One for each one of the applications. Both the IOS and Android applications will be capable of compile a full report that includes a picture, description, type and location of an issue. After compiling the report, the applications will be able to send it to our server for the website to see it.

In future semesters, the Application Team will add more features in order to add more functionality to the app and make it more user friendly. Some of these features include, but are not limited to, visualizing notifications from the Website, read from the server in order to show the user nearby reports to him, allow user to “acknowledge” an issue reported by another user and filtering issues reported by other users.

5 Semester Documentation – Spring 2019

5.1 Team Members

Name	*Role*
Mohammad Jahanshahi	●EPICS Advisor ○ Advises EPICS Syllabus learning objectives ○ Technical and professional guidance
Margaret Phillips	●EPICS Advisor ○ Advises EPICS Syllabus learning objectives ○ Professional guidance
Seyedali Ghahari	●Teaching Assistant ○ Academic logistics and operations for EPICS section ○ Guidance in area of expertise - Civil Engineering ○ Assisted team in moving forward and finding resources
Miguel Kulisic	●Project Manager ○ Senior ○ Manages the Website and Application Teams ○ Helps the Website Team with Technical Issues
Ishaan Ahuja	●Website Design Lead ○ Senior ○ Manages the progress of the website team
Alex Goebel	●Project Partner Liaison ○ Sophomore ○ App Team ○ Main Contributor to iOS App Development

Connor Gonzalez-Fox	●Project Archivist ○ Freshman ○ App Team ○ Main Contributor to Android App Progress
Nikhil Iyer	●App Design Lead ○ Sophomore ○ Manages the Progress of the App Team ○ iOS/Android App Contribution
Sreekara Yachamaneni	●Webmaster ○ Sophomore ○ Website Team ○ Assisted with the Development of a Working Website Firebase
Owen Price	Financial Officer ○ Freshman ○ Website Team ○ Assisted with the Development of a Working Website Firebase

5.2 Current Status and Location on Overall Project Timeline

The *Project Specification, Specification Development, and Conceptual Design phases have been completed.*

The Detailed Design phase is currently in progress.

We began the Spring 2019 Semester by familiarizing the current website team with the previous semester work. Two out of the three members in the app team were returning members so this process was not necessary for this team. Our main goal for this semester is to deliver a full working prototype. That means establish back and forth communication between the app and website. Currently the app is capable of sending messages to the website and the website can read those messages.

At the middle of the semester, we are currently working on polishing the read capabilities of the website so that it can display the data received in the manner we envision. In a similar fashion, the app team is both finishing the sending reports feature in the app and started development on the iOS version of the app.

5.3 Goals for the Semester

As previously mentioned, this semester we want to achieve back and forth communication between the app and the website. This will allow the project to be tested in the real world and will give us feedback on things we'll need to improve. This first working prototype will only be available with an Android application. Since development for the iOS app only started this semester it will not be ready until a few more semesters of work. However, we do expect the iOS app to be able to send reports by the end of the semester.

5.4 Semester Timeline

Smart City SharePoint>Semester Documentation>Fall 2018>Gantt Charts:

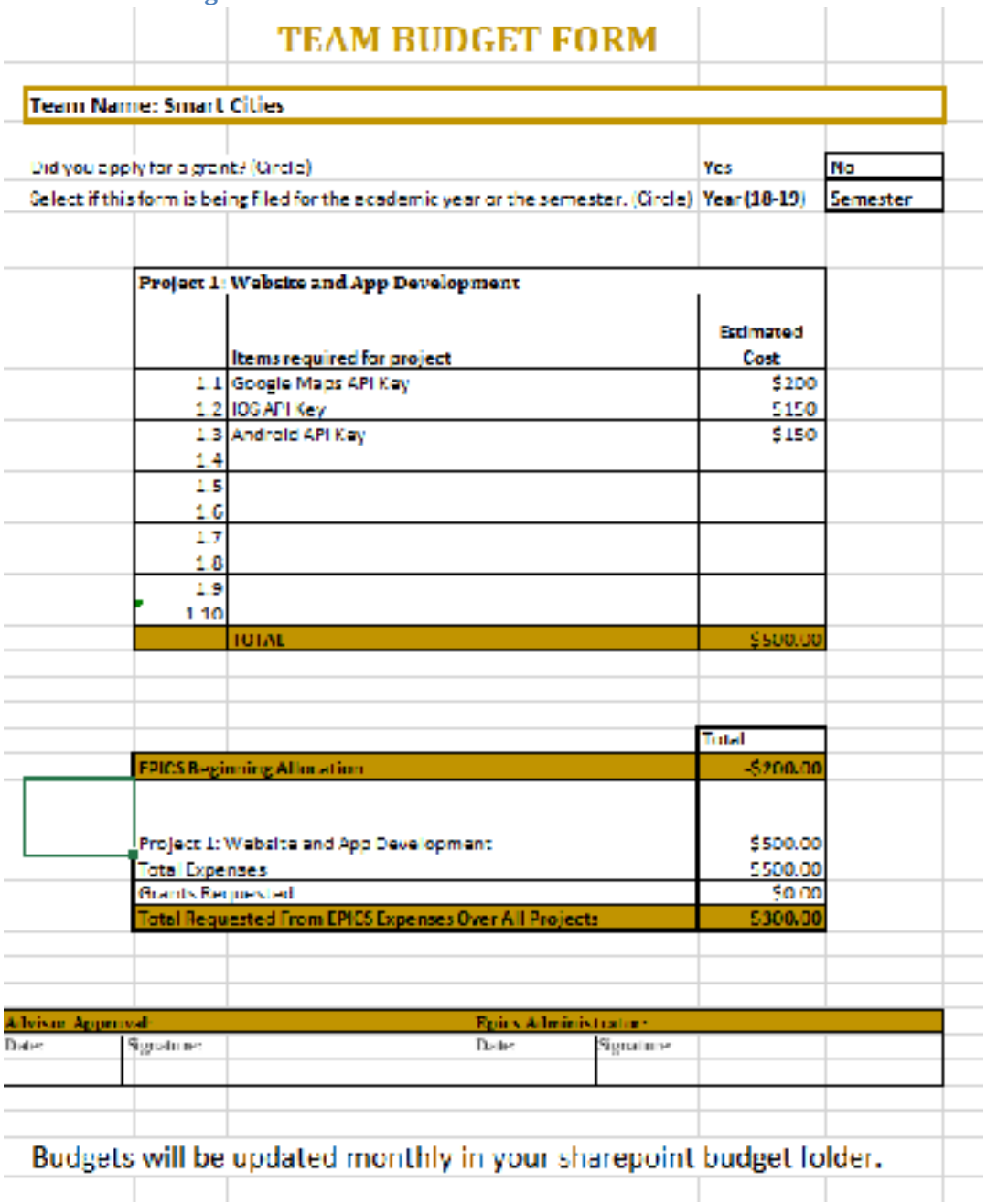
Week	*Design Process Milestones*	*Status/Notes*
1 — 01/11	Introduction to EPICs	• EPICS information/outcomes overview
2 — 01/18	Project Scope Review	• Smart City information/outcomes overview • Project roles assignment
3 — 01/25	Detailed Design	• Understand where previous semester left off • Brainstorming and timeline generation
4 — 02/01	Detailed Design	• Individual team cost-breakdown analysis • Team Website Update • Meeting with project community partner
5 — 02/08	Detailed Design	• Gantt Chart Update to Advisors and TA • Budget Analysis to EPICS Office
6 — 02/15	Detailed Design	• Mock Mid-Semester Design Review

7 — 02/22	Mid-Semester Design Review	• Design Document preparation and slides for MidSemester Design Review/Evaluation • Individual team testing/elaboration of available technology
8 — 03/01	Detailed Design	• Re-evaluate project based on Mid-semester review feedback • Revisit *Project Identification*, *Specification Development*, and *Conceptual Design*
9 — 03/08	Detailed Design	• Further design implementation
10 — 03/15	SPRING BREAK	
11 — 03/22	Detailed Design	• Further design implementation
12 — 03/29	Detailed Design	• Further design implementation
13 — 04/05	Detailed Design	• Final Design Review Presentation Slides • Design Document preparation for Final-Semester Design Review/Evaluation
14 — 04/12	Detailed Design	• Delivery checklist, if delivering • Design Document preparation for Final-Semester Design Review/Evaluation • 2-page Transition Documents for each team • Mock Final Design Review
15 — 04/19	Final Design Review	• Slides and Presentation preparation for Final-Semester Design Review/Evaluation • 2-page Transition Documents for each team

16 — 04/26	Final Write-ups	• Finalize 2-page Transition Document • Update Design Document with feedback
17 - 05/03	FINALS WEEK	**Design Process to be done** (see **Detailed Design**): • *Functional Prototype* (needs finishing) • *Functional Evaluation* • *Revised Prototyping* • *Field Testing* • *Drawing Conclusions & Recommendations*

5.5 Semester Budget

5.5.1 Proposed Semester Budget



TEAM BUDGET FORM

Team Name: Smart Cities

Did you apply for a grant? (Circle)	Yes	No
Select if this form is being Filed for the academic year or the semester. (Circle)	Year (18-19)	Semester

Project 1: Website and App Development		
	Items required for project	Estimated Cost
1.1	Google Maps API Key	$200
1.2	IOS API Key	$150
1.3	Android API Key	$150
1.4		
1.5		
1.6		
1.7		
1.8		
1.9		
1.10		
	TOTAL	$500.00

	Total
EPICS Beginning Allocation	-$200.00
Project 1: Website and App Development	$500.00
Total Expenses	$500.00
Grants Requested	$0.00
Total Requested From EPICS Expenses Over All Projects	$300.00

Advisor Approval:		EPICS Administrator:	
Date:	Signature:	Date:	Signature:

Budgets will be updated monthly in your sharepoint budget folder.

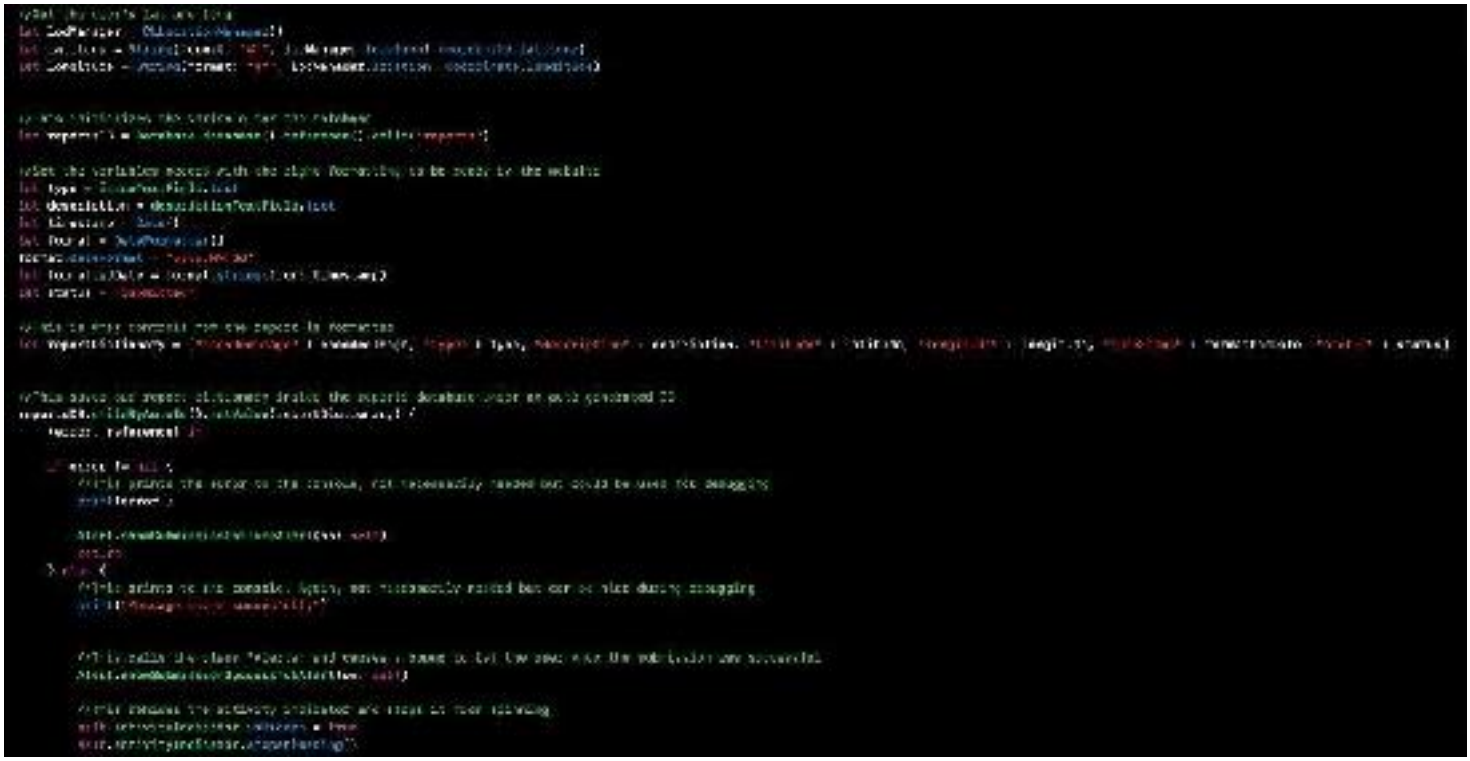
5.5.2 End of Semester Spending

We ended the Spring 2019 Semester with no expenses to the team as the API keys were not needed to be bought without a finished product.

6. Current Design

6.1 Website and Application Development

6.1.1 Application Development



The Application Team had an interesting and difficult task ahead of itself as up to now the team had only worked on the Android App and not the iOS version. This is important as part of the request from the city was for the app to be available on both iOS and Android devices. The App Team split into two groups, one to focus on the Android Development and the other to focus on the iOS development. The Android Application had functionality by the end of the Fall 2018 semester, but it had a few quirks and did not look entirely the best. Because the previous team was able to get us a completed app, the team was able to reformat the code into fragments to allow more optionality. Doing this would make it so in the future when the app would need additional functionality for new ideas and uses, the ability to update would be easier for future programmers. The app was also updated in an aesthetically pleasing way to provide a better experience for the user (Pictures of this can be found below). The iOS Application was built from the ground up this semester, so our main hope was to be able to at least get the app slightly ready, but not necessarily ready to submit requests to the Firebase. Our programmers did not have much experience with Swift (the coding language/software for iOS devices), and due to this it was a slow start as a lot of learning had to be done. After this learning the app took off with help of the knowledge of layout of the Android App. The team managed to complete an iOS Application that not only functions, but it can submit reports to the Firebase. This was a big accomplishment, and the code for the iOS devices sending to the Firebase can be found above.

6.1.2 Website Development

The Website Team had a less grueling task for the semester, but while the work was not a large amount, it was hard work. The Fall 2018 Semester produced a website that was able to receive the submissions from the Firebase, but it was unable to send back to the Firebase for the application to receive updates on the different reports or submissions. The Website also had functionality but could be cleaned up a little bit more. The team decided to first tackle the formatting issues on the appearance of the website, as solving that issue could lead to a nicer look and easy implementation at the end of the Semester to send updated reports back to the Firebase. The team was able to fix the formatting errors (pictures of the new website look can be found below) and the website could take reports and update their status with a drop down. This meant that if a report was completed we were able to send it back to the users of the application, so they would know that their issue was resolved.

6.1.3 Existing Solutions

There are a few mobile apps that are available in the market which detects potholes, traffic hazards, and tripping hazards. For instance, Street Bump[31], an application which residents of Boston can use to collect road condition data while they drive. It can detect bumps on the streets and provide the city with real-time information to fix short-term problems and plan long-term investments. However, this type of applications is city-specific and is not suitable for the city of West Lafayette. Other solution would be what the city of West Lafayette currently does which is manually assessing damages on the roads. In order to benchmark our potential solution, we could use the similar applications available in other cities (Street Bump, Get It Done San Diego Official). Based on the initial benchmarking with the project partner, we've identified some needs and functionalities for the smartphone application:

User Need	*Specification*
City must be able to easily see frequency of pothole reports	• Application must have an admin login option, with a password requirement (backend) • Must have a map to show frequency of reports to admin users • Must have a map to visually plot reports, as well as an address description of the general area, instead of exact coordinates • Map must merge submissions in the same location as hardware data
City must be able to determine the severity of the potholes	• Ability to take picture of damage • Users must be able to upload pictures of road damage • Users must be able to write a description of the damage • Ability to send the location of damage

Application must be accessible and appealing to the widest range of users possible	• Application should be programmed for iOS • Application should be programmed for Android • Option to make submissions anonymously • Application should require less than 1 minute to submit a report of damage

We discussed many strengths and weaknesses that would allow for convenience and speed for user-application interaction. We wanted to identify functions that would provide more accurate and more useful information to the community partner. After an initial meeting with our community partner, we assigned roles to create a conceptual design that would "provide the best user ability and practicality to all of our stakeholders" (Marcus, assistant city manager). Various conceptual designs that our team created:

6.1.2 Progression: Spring 2019

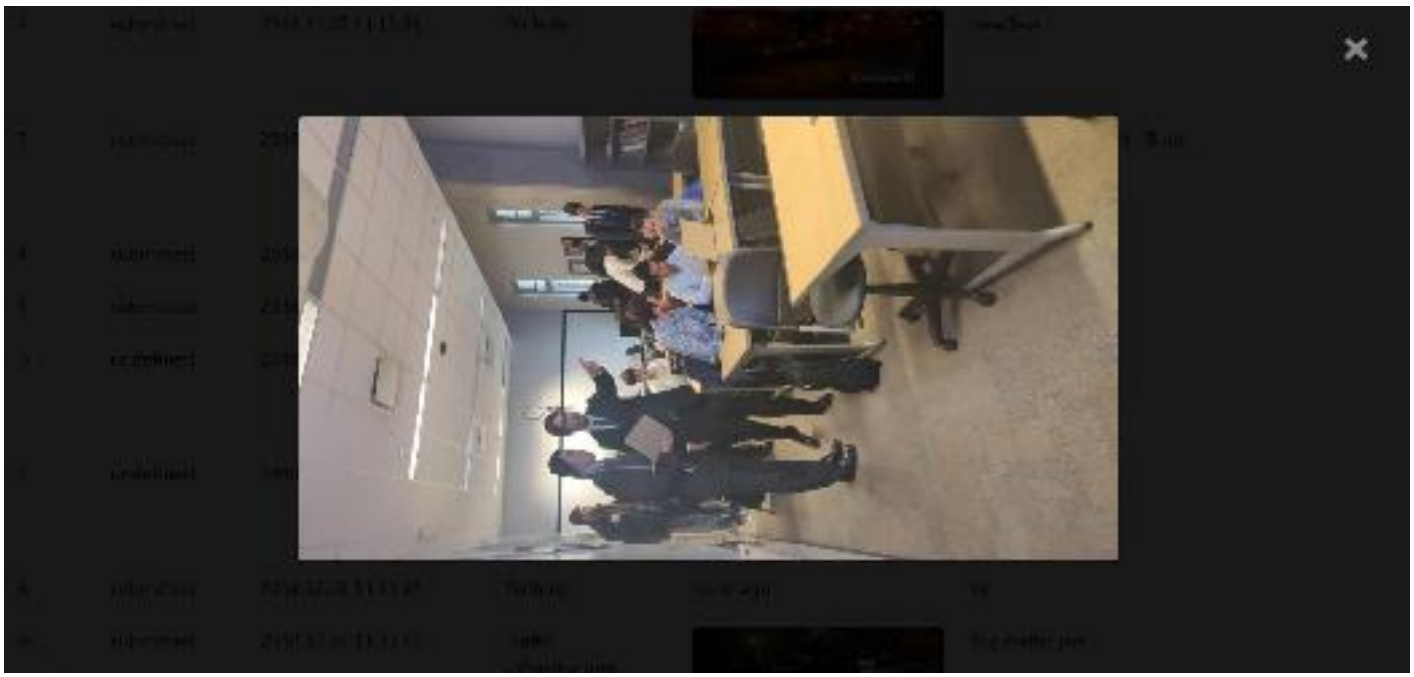
The Spring 2019 website team has been working hard on making the visualization of the data better. We are also focusing on giving the website a more professional look.

As you can see, filters have been moved to the top of the page once again. We made the decision to do this because the main focus of the website should be markers on the map.

Reports

Index	Status	Time Stamp	Type	Image	Description
1	submitted	2018.11.05.11:11:21	Pothole		never
2	submitted	2018.11.07.11:13:01	Pothole		newTest
3	submitted	2018.11.07.11:14:02	Tripping Hazard		new Submission 11:15 pm
4	submitted	2018.11.14.11:27:26	Pothole	no image	
5	submitted	2018.11.14.11:55:21	Pothole	no image	
6	undefined	2018.12.03.10:33:38	Pothole		new

The appearance of the table has also been changed to make it more pleasant to look at. The columns displaying both latitude and longitude have been removed because those mean very little to a human. They will be replaced with street addresses once reverse-geolocation is implemented. Also, an index has been added to the table, this index matches the index of the markers on the map.



Finally, in order to make the images in the table usable, we have made them clickable. Once clicked they expand so the user can see the image at its full size.

6.1.3 Project Specifications (Spring 2019)

Website

- ❏ Website for the city to review this information
- ❏ Pictures can be used to identify potholes (deep learning)
- ❏ Place for city to be able to view reported pictures
    - ❏ Attached with markers in website
    - ❏ Streets with many potholes are painted based on their severity
- ❏ Read data from a server
- ❏ Usage of simple GET requests
- ❏ Use information gathered by both the users of the app and the POLES team
- ❏ Adjust data points to correct for curvature in streets
- ❏ Use of express, bootstrap and jquery to make web development easier
- ❏ Online programming environment for the website that will later be transferred into a server (cloudnine)
- ❏ Website will only be accessible through the city's network
- ❏ Local password might be instated (if the city needs one)
- ❏ Easy to use UI

- ❏ Visually appealing styling (simple styling)
- ❏ Filtering of the potholes based on Severity, street and date (month)
- ❏ Programmed to work on all screen sizes including mobile **API**
- ❏ Map API integrated into app
    - ❏ Map can move around
    - ❏ Move point in center of map (e.g. Uber)
- ❏ Address bar able to locate positions
    - ❏ Street name, city name, state, zip code
- ❏ Location Services
    - ❏ Current location
    - ❏ Location of the address typed
- ❏ Camera API integrated into app
- ❏ Usage of simple POST requests
- ❏ Developed to run on both new and old android version
    - ❏ Developed UI

User Experience

- ❏ Scrollbar for severity
    - ❏ Use of three severity levels:
        - ❏ Mild
        - ❏ Moderate
        - ❏ Severe
- ❏ One screen layout, easy to use even for first time users
    - ❏ Map functionality easy to interact with
- ❏ Visually appealing interface
- ❏ User safety accounted for
    - ❏ Alert/notification system

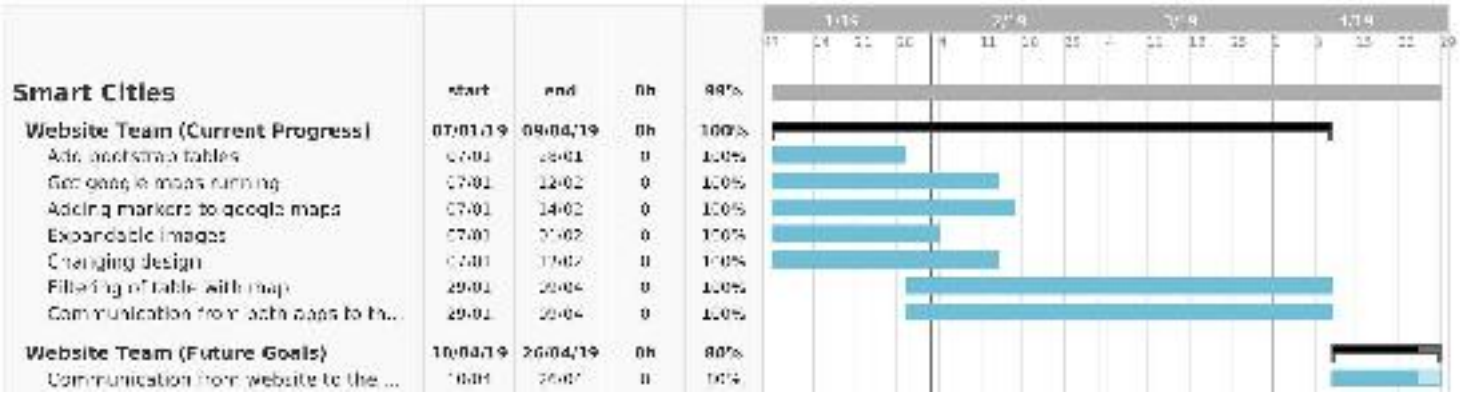
Data Collection

- ❏ Setting up an appropriate server
    - ❏ Enough space to store data
    - ❏ Compatible with both app and website
- ❏ Efficient data package
    - ❏ Stores picture separately from text data
    - ❏ Packages data to be sent to server for storage

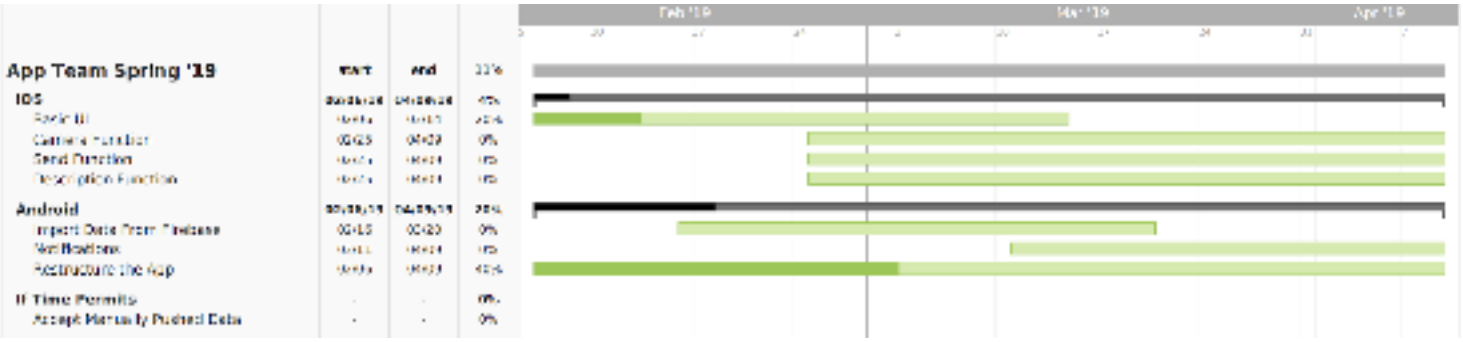
- ❏ Take user input into data package
- ❏ Send data to server

6.1.4 Gantt Charts

Website:



App:



6.1.5 End-of-Semester Summary

For Spring 2019, we continued what we had from previous semester and made much progress in improvements as well as additions. This semester, we particularly focused on having proper documentation for our team so that the future semester don't have a problems understanding what has previously been completed. Moreover, for the app, we included a Real Time Database server so that we have the information collected from the user of the reported pothole. This was a major accomplishment as any app needs a server to collect data. We also fixed the user interface of the app so that it appears robust. We also did some error handling in the app such that the user cannot send the information unless they have specified the severity and taken a photo so that inaccurate information is sent. Additionally, a major feature we added is to use the GPS of the device to get user location so that the user starts the app with the map pointing to their current location. This makes it easy for the user to start off the app with their own location.

For future semesters, we hope to do rigorous testing of the app and deliver it to general users to get feedback and suggestions for improvements regarding, but not exclusively, usability. Moreover, as this app is current only compatible with Android devices, futures semester should have a goal of delivering an iOS version as well. Additional features should also be implemented to report general issues apart from potholes as per the project partners' need.

7. Most Recent Archive

7.1 Website and Application Development

Figure: Conceptual designs of Smart City Cities App

List of questions and criteria for project partner meeting, assumed needs to be met:

Question	**Response**
What data do you want to be able to collect? ● Size of pothole (SML)? ● Description box? ● Do you need a picture? ● What needs to be in the picture?	● Picture would be nice ● But worried about asking for a picture ● Maybe can be option, but not necessary because of safety ● App would be more for street department ● Would be developed for immediate identification ● Better to have a map than location services because of location of person ● Like point on the area ● Maybe both??
How do you want to collect location data? ● Is GPS at location of phone enough? ● Specific address?	● Location of pothole ● Coordinates
What kind of user data will be useful to you? ● Name is enough? ● Anonymous is allowed? ● Maybe a Facebook/Gmail sign up?	● Maybe jerks are putting random points… ● Not really needed to sign in, can be anonymous
Is the app meant to be long term? ● Does it only need to focus on potholes, or cracks and other things as well? ● maybe can be used for broken street lamps, broken stop signs etc.	● only potholes ● Maybe leaves? ● But not most important
Do you want other people to see all reports made on a map? ● Avoiding repetitive reports ● Confidential? ● admin privileges to report when a report has been resolved	● Want to see the same pothole reported more than once to show importance ● Don't show other reports because will compromise showing frequency ● Admin privileges would be good
iOS and Android?	● Whatever is easiest, look at statistics ● Preferably both ● Want to get as many people as possible

Spring 2018 continued where Fall 2017 left off with the same specifications from previous semesters. To be thorough, we asked the project partner once again regarding goals and specifications desired within the application. The following are the questions that were reiterated:

Question	**Answer**
Specifically, who will be utilizing the app? Will the app be available to the public? Or only for city use? Is there a specific design that is preferred for the application?	● No, you can design what you think is best. ● Simple, clean, and professional design so that residents who are using the app aren't confused and can quickly figure out how to use it.
What specific features are you looking for in the app?	● Giving residents the ability to report issues (potholes, drainage issues, broken sidewalks, missing or damaged street signs, traffic concerns, and other issues) ● Giving us enough information to find the issue and fix it quickly such as description, location, and an image (optional). ● Currently residents can either call or email us, or submit a complaint via our website (https://www.westlafayette.in.gov/egov/apps/action/center.egov?view=form;page=1;id=126) ● Giving residents another option for reporting complaints will be beneficial and may reduce the number of calls coming into our office. That would be the base desire of the app. Another use for the app could be to include a notification or information sharing system so that we could send an alert to people when a road is closed, when we are doing construction, if roads are icy, etc. Right now we use Nixle, which texts and emails people with that information, but residents might like another option. ● Have the option to send an update once an issue has been addressed. Duke Energy uses a system to report street light issues. Once the light has been fixed, an alert is sent.

Specifically, who will be utilizing the app? Will the app be available to the public or only for city use?	• The general public to report concerns to us. • Back-end website would only be available to city employees.

After collecting an initial project partner meeting, the team decided to draw a sketch to decide what we want our app to look like. We considered affordability, user needs, and all stakeholders as main functionality needs. We decided to design a simplistic front page to keep space for future additions. For affordability, we wanted to release the app to one app store only. This will help us test out the application first, and resolve any issues, before releasing it on another app store as well. With user needs, the app has to be easy to use, free of charge and it will process quick submissions. Finally, while brainstorming, we also had to make sure we took all the stakeholders into account.

After analyzing each conceptual design, the team has decided to stick with the following design (simplistic, modern, and practical):

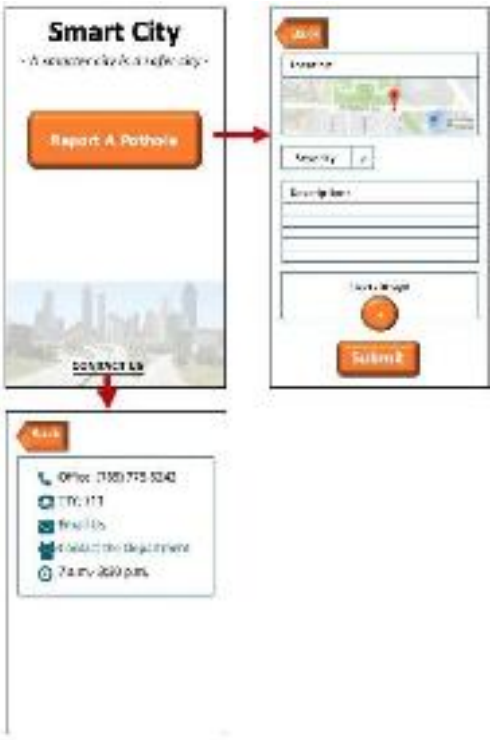


Figure: low-resolution prototype

From this concept, the team learned that the submit button functions as a data sender to the Purdue data storage server at EPICS. Also, we learned that the severity button should be a dropdown and not a pop up – simplifying the application functions with little possible error. We also learned that certain images are copyrighted and cannot be used in the platform (we took our own pictures). Overall, the team was excited to see that the functional prototype produced little errors while using the application.

**Note*: The chosen design opens up to a very simplistic and straightforward screen, where the name of the application is highlighted, with an option to report a pothole or contact the city (leading the

user into different pages). The opening screen includes report and contact to increase user-friendliness and allow users to complete their objective in an efficient manner. The "Report A Pothole" button is very eye-catching and takes users to a second page. Users will be able to confirm the location of the pothole on a map, select the severity of the pothole from a preselected list of severities, describe the pothole with a 2-3 description, or add a picture of the pothole. Options to describe or add a picture of a pothole are not required, but users will need to put fill in one of them. After finishing up their report, users are able to submit the pothole report to the city. If the user accidentally got to pothole report page, instead of contact the city page, there will be a "Back" button conveniently located on the top left corner for users to click. The "Contact Us" button is located on the bottom of the opening page that takes users to a page with information regarding contacting the city with city business hours. Finally if the user wishes to go back to the main page from the contact page, there is another "Back" button located in the same location as the other to reduce room for confusion.

Due to our project scope, we do not have a typical manufacturing or assembling process. Our end-deliver is a smart-device application and is not tangible – digital propitiatory property does not require physical manufacturing.

Github is an online programming repository for open source collaboration. Downloading Github and the programming compiler (translates human language to machine language) was long and difficult.[33] Implementing Github and compiler installation will be documented in our transition document. Also the code behind our process was "assembled" by our team. While this code will stand alone and not be changed following delivery of the product, it is still important to document what each part of this code does. The practicality of documenting commands and functions in a user manual type setting is nonexistent. Instead the code which we will deliver will be commented thoroughly. Not only will this be beneficial upon the delivery of the product, but also in the transition between semesters. This documentation will be important in the increase of efficiency between semesters.

Here are bill-of-materials that may be necessary to understanding/implementing future progress:

Table: Bill-of-materials for future progress

The team has already faced several setbacks resulting in delays. One example of this is the difficulty we experienced while downloading the selected compiler. Spring 2017 App was not able to complete the application code prior to the end of this semester. Some of our time constraints include spring break and individual time allocated for studying and other class work. Our EPICS document deadlines and individual time allocated for studying and exams. There are no external time constraints imposed by our project partner. While they have indicated that a timely delivery of this phone application is favorable there is no required publication date.

Item	**Made/ Bought**	**Vendor**	**Quantity**
Homebrew	Downloaded	https://brew.sh/ Design Do	ument mart 1
Node	Downloaded	React Native Ci	y 1- Fall 2018
Watchman	Downloaded	React Native	1
React Native	Downloaded	Terminal; instructions: https://facebook.github.io/reactnative/docs/gettingstarted.html	1
Xcode	Downloaded	App Store	1
Android Studio	Downloaded	Google Play	1
Github	Downloaded	https://github.com	1
Sublime Text	Downloaded	https://sublimetext.com/2	1
Notepad ++	Downloaded	https://notepad-plus-plus.org/download/v7.3.3.html	1

**Note:* Submit functionality currently shows errors and we suggest looking through where objects are being called and changed within the React Native environment.

[33] Contact Kartik at mittal38@purdue.edu for access to Github

**Note*: Our detailed design did not implement data saving (server, locally, writing data to .txt). Generally, App needs to implement the back-end analytics.

For Fall 2017, strides were made in making an actual functioning app. The code was scrapped from the previous semester, as it allowed for us to have more freedom in creating what we have envisioned. One of our early prototypes for the application ended up looking too bulky, so we decided against using it:

One of our problems with the option above was that it took too many screens to get to reporting. We have opted for a single screen experience, going for a look similar to that of Uber:

This gave Fall 2017 the functionality Spring 2017 semester was looking for (ability to take pictures, to report severity, and to use GPS), while also incorporating a sleek, efficient overall layout. The design has a permanent marker in the middle that would allow for the user to simply place where the pothole was that they saw, and can convert it to an address later for the city using the Google Maps API. All the needed features are on the one page, and next to the address bar on the top of the image there is an 'options' button that will give contact information for the city. Fall 2017 intended to add a 'severity' scrollbar, but due to time constraints, were unable to execute. Thus, this has become a goal for the Spring 2018 semester - a 'severity' scrollbar that has only 'mild', 'moderate', or 'severe' to simplify and categorize the potholes for people deciding on the severity of the pothole. These options are much more concrete than a number scale from one to ten. Each pothole will still have a ranking for the city to determine which pothole requires more attention than others.

Code was developed for what this will look like, and we currently have a layout prototype with some functionality. To make things easier for the city, we have created a website for the reported data to get sent to so that it can be seen without a problem for city administrators, but cannot be viewed by the general public. Below is a view of our current app, with its current functionality, and our near fully operational website:

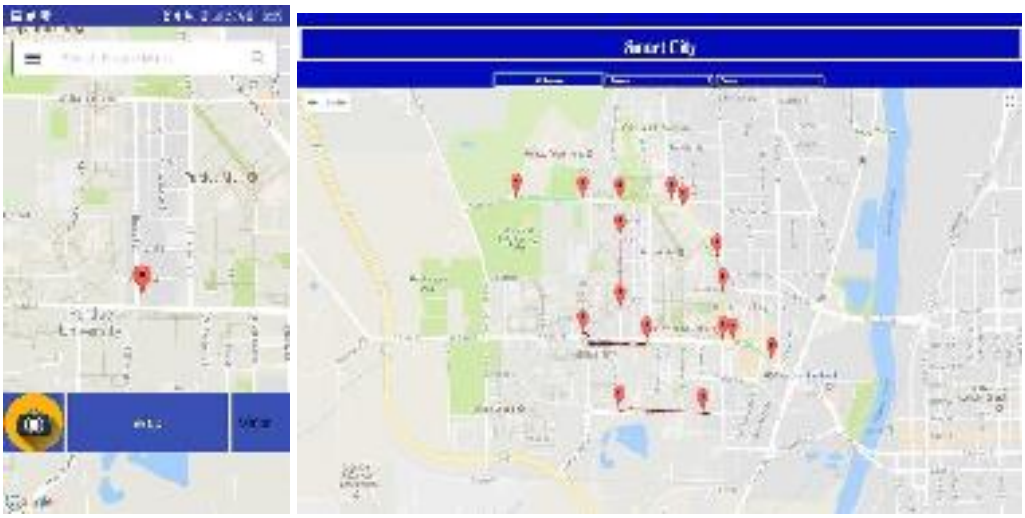


Figure: A view of the app and website

Figure: Enhanced version of the app (Spring 2018)

We are continuing to add functionality, but for now we have attained a scrolling map with a working camera option. Soon we will link our app and website to get reported points to show up on the site. Already, the website is able to calculate severity of roads and can filter through multiple options. Spring 2018 is also making it a priority to consistently document code to ensure ease of transition between semesters as a result of personal experience. For this reason we used GitHub, which is a web-based hosting service for version control and storing the changes made by user is the code. It also gives us the option to revert back to any old code so if we hit a dead end, we can go back to a point where we want. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features

The prototype of the app does have the functionalities that we intended to include within it; however, the layout does not look like the way that it is supposed to be. The bottom bar does not stick to the bottom and more stays in the middle, which distracts the view of the map. Thus, Spring 2018 App Team fixed this problem after learning the code process of Android Studio and the result is shown as the figure above. Spring 2018 is also trying to come up with the ways to improve the user interface to increase the accessibility of the app for the users.

The main objective for Spring 2018 semester was to have a backend server for the app where the information of pothole like coordinates, severity, image encoded in Base64 format (this is an encryption format which convert an image to string using inbuilt Android libraries so that it is easy to store on the server and decrypt back to an image to display it on the website) and time stamp as in what time the data was sent to the server. The team brainstormed on what server to choose and made a decision matrix included below to make it more clear which one to use. The 3 different option we narrowed down for server were - Google's Firebase, Amazon AWS Mobile Hub and Microsoft's Azure.

What is Firebase?	What is AWS Mobile Hub?	What is Microsoft Azure?
Firebase is a cloud service designed to power real-time, collaborative applications. Simply add the Firebase library to user application to gain access to a shared data structure; any changes user makes to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.	AWS Mobile Hub is the fastest way to build mobile apps powered by AWS. It lets users easily add and configure features for their apps, including user authentication, data storage, backend logic, push notifications, content delivery, and analytics. After users build their apps, AWS Mobile Hub gives them easy access to testing on real devices, as well as analytics dashboards to track usage of their apps – all from a single, integrated console.	Azure is an open and flexible cloud platform that enables user to quickly build, deploy and manage applications across a global network of Microsoft-managed datacenters. User can build applications using any language, tool or framework. In addition user can integrate his or her own public cloud applications with existing IT environment.

After thinking it through, we made a decision matrix to choose the most optimal option.

Factors: (0-5 scale)	Cost	Quality	Reliability	Accessibility	Total
Google's Firebase	4	5	5	5	**19**
Amazon AWS	3	4	5	3	**15**
Microsoft's Azure	5	5	5	3	**18**

Benchmark

	5	4	3	2	1
Cost	< $25	$25~$34	$35~$44	$45~$54	≥ $55
Quality	≥ 100k simultaneous connections	100k~1k simultaneous connections	1k~500 simultaneous connections	500~100 simultaneous connections	< 100 simultaneous connections

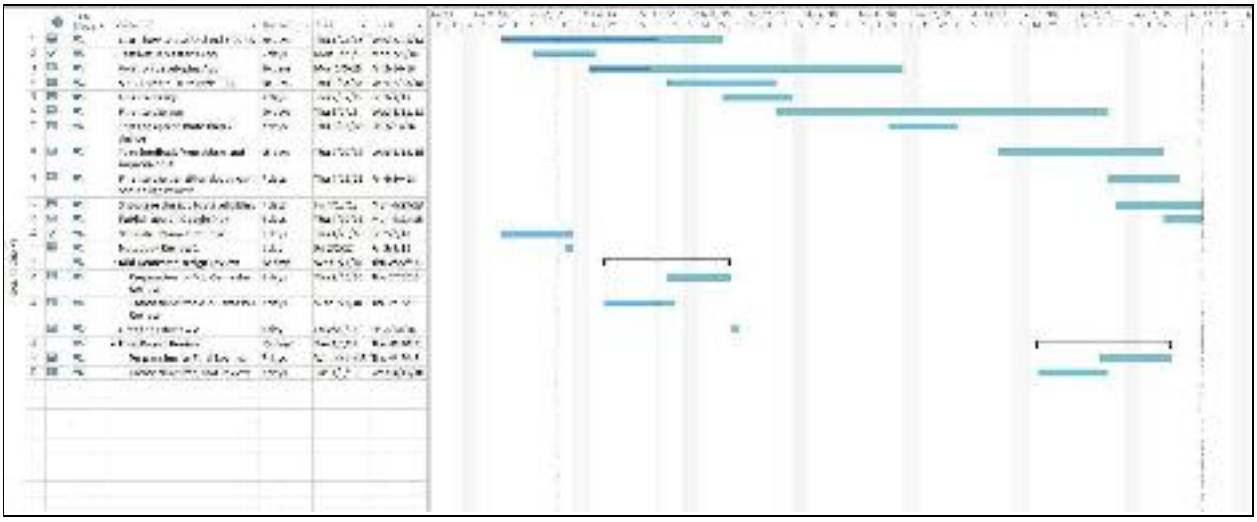
Reliability	Excellent security	High security	Average security	Low security	No security
Accessibility	Fully compatible with Android	Almost compatible with Android	Partially compatible with Android	Barely compatible with Android	Not compatible with Android

Figure: Data stored on Google's Firebase server

The image above shows how actually the data is stored in the server. The information is stored in JSON file (JavaScript Object Notation file) on the server which is easy to read and understand by humans as well as software.

Overall, Spring 2018 is focusing on implementing the expected features as well as additional ones that have been requested by the project partner.

Overall Timeline



7.2 Website Progression: Fall 2018

The team from the previous semester had done great work. One of the issues that was odd was that the code on the EPICS Sharepoint from Spring 2018 seemed to be structured for Amazon Web Services (AWS) Cloud9. It took around 2 to 3 weeks to make sure that every team member could deploy the website locally on their machine.

After the initial setup was completed, we identified some areas on the website that could use some improvement such as implementing colored buttons that will separate the in-progress reports and the new reports or the use of a basic login system. The colored buttons would help improve report organization and the basic login system would allow for a basic layer of security. With all of these improvements in mind, we designed our concept design of the website to make it as efficient and as readable as possible for the City of West Lafayette's Engineering Department.

Hosting

For Fall 2018, we managed to host the website (https://epics-smart-city-website.firebaseapp.com/) on Google Firebase's hosting service. One problem that arose when trying to host the website was that the Google Maps was not loading properly when deploying locally. This semester, we originally got the map to load, but the API key was expired. (The API key is a unique string of letters and numbers generated by Google as a password and proof of use of their services.) Setting up a database is still proving to be tricky as Firebase a couple of files that needs to be changed or moved from their current locations to meet our needs. Another problem that came up last semester was coding the file in either JavaScript or HTML and whether Firebase would be compatible. After team meeting, we decided to go forward with using HTML as the primary language. This was useful because HTML has a few built-in features, like authentication.

It was much easier to rearrange website elements as we saw fit in the near future. For example, it was much easier to move the category filters towards the bottom of the website, if necessary.

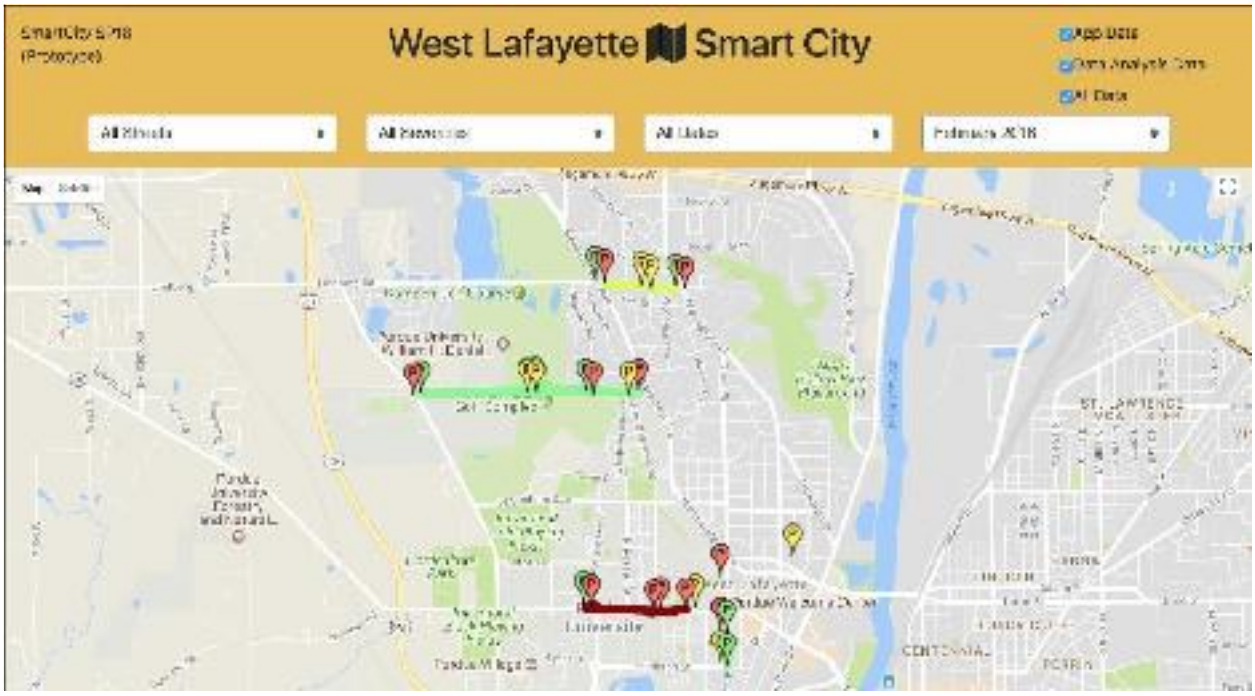
Connecting to a Server

For now, our Android application has been successfully integrated with Firebase Database server. More information can be seen in the application section.

During this semester (Fall 2018), there is a team member working on setting up the database so that it would easily connect with the Smart City website. After this has been tested and operational, we will configure the settings from the Firebase console and the necessary Firebase files.

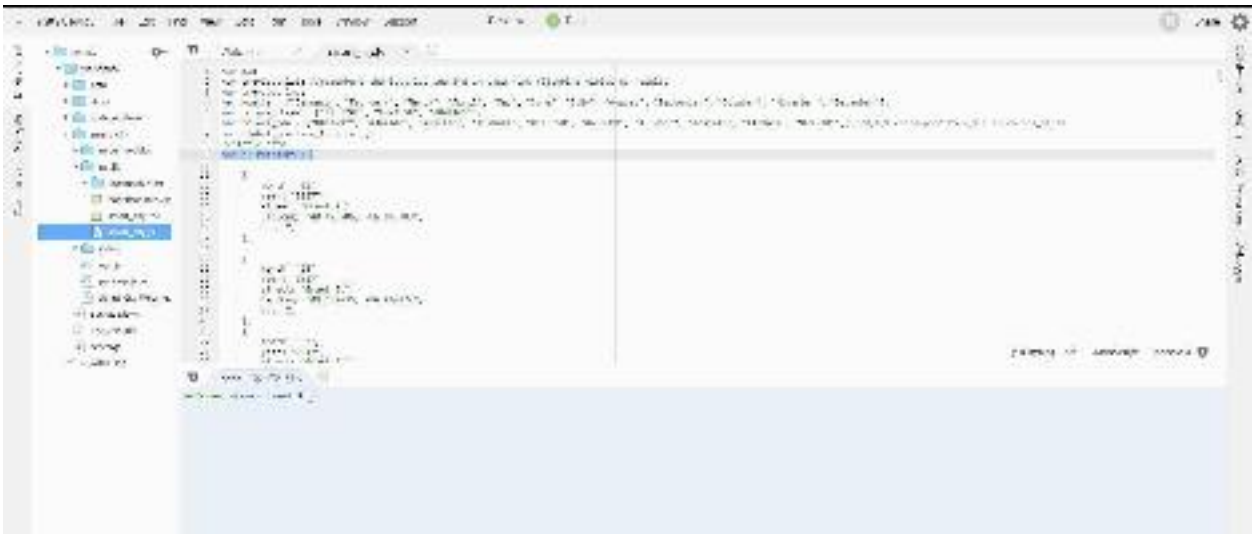
Data (Markers on Map)

The Fall 2018 team learned that all the data in the map (below) are hardcoded. This is the only option available as no server is established yet. The server is needed so that the data from Kinect and application can be stored in one place. The plan is to make the website read the data from the server and automatically update it on the location instead of manually entering the code. The current focus of the website is to display data from both app and Kinect as pothole reporting website is already available on the official City of West Lafayette's website.



The code for the markers can be found at workspace > smart city > public > smart_city.js . **var allPotholes** is the code where all the data for "App data" filter while **var allPotholes_DA** is where the data for "Data Analysis" filter.

Also attached is the location of the marker, and the line of the code (assuming the code is untouched):



7.2.1 Final Design Review Comments/Reflection

- All decision matrices need to be weighted and ranges specified.
- Include “usability;” ease for a user to work with a server service.
- How do you differentiate between levels of severity from different users?
- Implement feature so that the app cannot be used while driving.
- Will there be any type of prioritization when scheduling repairs? Will traffic data be incorporated?
    1. Future implementation of re-routing feature
- Address why the app is only being designed for Android at the moment in the Design Review.
- How were coding and interfacing decisions made, and does this meet the needs of majority of users?
    1. Reiterate the needs of the project partner.
    - Include decision matrices.
- Data flow diagram was over simplified and confusing.
- Address security and data ownership on server selection.
- Size constraints and/or optimization in app before sending through app?
- Consider how the website will be monitored and updated when potholes are fixed.
- What happens if two people mark the same pothole on the app? Does the severity level update with the most recent data received?

Appendix A: Past Semester Archive

A.1: Team Members

A.1.1: Fall 2017

Name	*Role*
Mohammad Jahanshahi	●EPICS Advisor o Advises EPICS Syllabus learning objectives o Guidance in sensor technology/analysis o Initial project innovator

Margaret Phillips	●EPICS Advisor o Advises EPICS Syllabus learning objectives o Guidance in academic research and group cohesion/leadership
Dahjung Chung	●Teaching Assistant o Academic logistics and operations for EPICS section o Guidance in sensor technology/analysis
Eric Jin Wook Choi	●Project Manager – responsible for overall operation and effectiveness of team and provides planning, direction, and guidance ●POLES (Data Analysis) o Detection and quantification of potholes o GUI (frontend) development

Mohammad Kobeissi	●App Design Lead o Oversees App design o Responsible for facilitating project through components of design process o Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget
Fajar Ausri	●POLES Co-Design Lead o Oversees POLES design o Responsible for facilitating project through components of design process o Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget ●POLES (Data Analysis) o Ensuring Kinect data collection (data management, error analysis) o Quantification of potholes o Google Maps API implementation to GUI (front-end)
Grant Hilbert	●POLES Co-Design Lead o Oversees POLES design o Responsible for facilitating project through components of design process o Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget ●POLES (ultrasonic) – ensuring ultrasonic data collection (data management, programming)
Gytis Kriauciunas	●Financial Officer – develop/manage project's budget ●POLES (GPS) – ensuring GPS data collection (data management, programming)
Yvette Chowdry	●Project Archivist – ensuring quality of project documentation and documentation practices ●App – GitHub implementation

Tiyani Hu	●Project Partner Liaison o Communication between teams and the community project partner o Inform on regular basis of progress of the project and relevant team documentation for partner observation/comment ●POLES (Kinect) – ensuring Kinect data collection (data management, programming, error analysis)
Weili Wang	●Webmaster – update/maintain project's website ●POLES (GPS) – ensuring GPS data collection (data management, programming)
Lexie Plocher	●POLES – ensuring inter-team communication, group dynamic, and team milestones
Nicholas Briggs	●App – pothole reporting functionality
Nicholas Idso	●App – GUI (front-end) development

A.1.2: Spring 2017

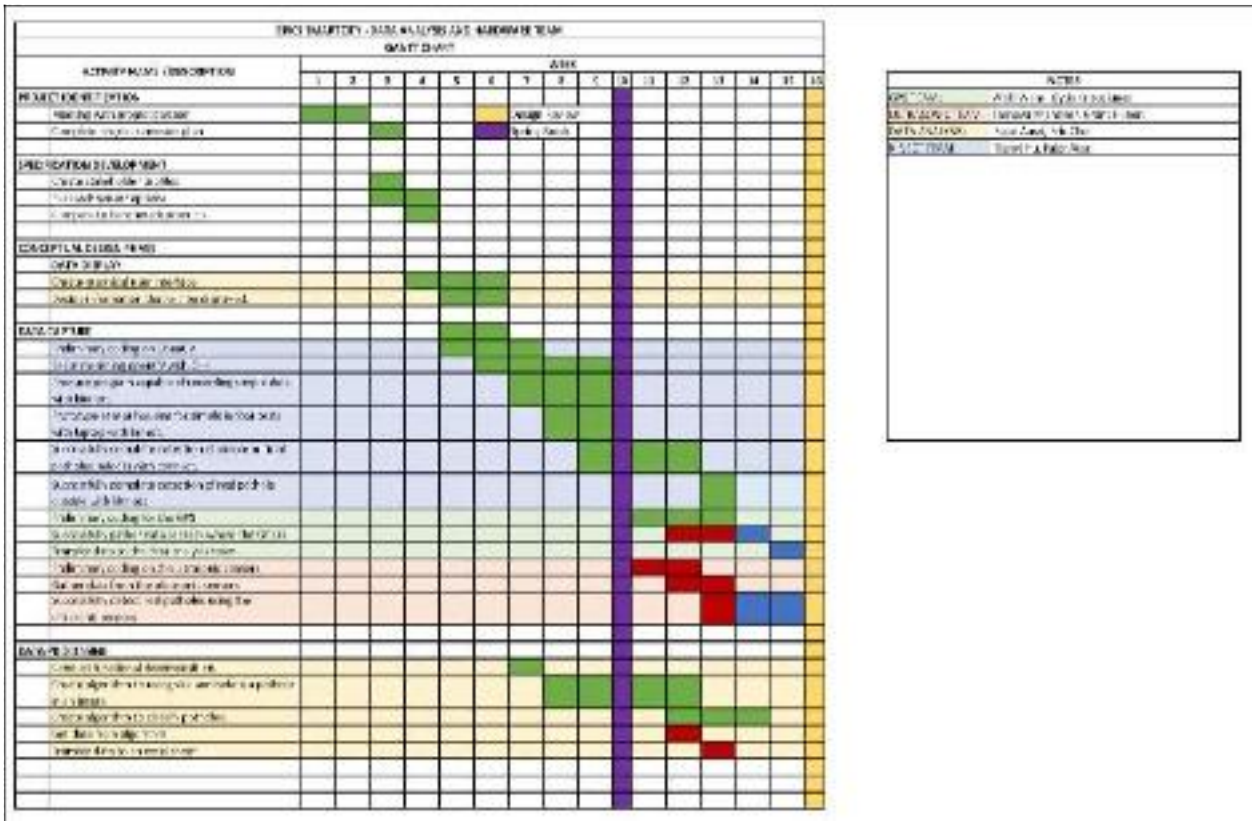
Name	*Role*
Mohammad Jahanshahi	●EPICS Advisor ○ Advises EPICS Syllabus learning objectives ○ Guidance in sensor technology/analysis ○ Initial project innovator

Margaret Phillips	●EPICS Advisor ○ Advises EPICS Syllabus learning objectives ○ Guidance in academic research and group cohesion/leadership

Dahjung Chung	●Teaching Assistant ○ Academic logistics and operations for EPICS section ○ Guidance in sensor technology/analysis
Eric Jin Wook Choi	●Project Manager – responsible for overall operation and effectiveness of team and provides planning, direction, and guidance ●POLES (Data Analysis) ○ Detection and quantification of potholes ○ GUI (front-end) development
Mohammad Kobeissi	●App Design Lead ○ Oversees App design ○ Responsible for facilitating project through components of design process ○ Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget
Fajar Ausri	●POLES Co-Design Lead ○ Oversees POLES design ○ Responsible for facilitating project through components of design process ○ Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget ●POLES (Data Analysis) ○ Ensuring Kinect data collection (data management, error analysis) ○ Quantification of potholes ○ Google Maps API implementation to GUI (front-end)
Grant Hilbert	●POLES Co-Design Lead ○ Oversees POLES design ○ Responsible for facilitating project through components of design process ○ Responsible for project planning, execution, risk assessment to deliver a quality team end-deliverable on time/budget ●POLES (ultrasonic) – ensuring ultrasonic data collection (data management, programming)

Gytis Kriauciunas	●Financial Officer – develop/manage project's budget ●POLES (GPS) – ensuring GPS data collection (data management, programming)
Yvette Chowdry	●Project Archivist – ensuring quality of project documentation and documentation practices ●App – GitHub implementation
Tiyani Hu	●Project Partner Liaison ○ Communication between teams and the community project partner ○ Inform on regular basis of progress of the project and relevant team documentation for partner observation/comment ●POLES (Kinect) – ensuring Kinect data collection (data management, programming, error analysis)
Weili Wang	●Webmaster – update/maintain project's website ●POLES (GPS) – ensuring GPS data collection (data management, programming)
Lexie Plocher	●POLES – ensuring inter-team communication, group dynamic, and team milestones
Nicholas Briggs	●App – pothole reporting functionality

A.2: Fall 2017 Timelines



A.3.2: App Team

	Week															
Phases	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Project Identification Phase																
Description of the Community																
Stakeholders																
Social Context																
User Needs																

Specification																
Development Phase																
Benchmarking/ IP																
Specifications																
Conceptual Design Phase																
Brainstorm																
Low Resolution Prototyping																
Concept Convergence																
Proof-of-Concept Prototyping																
Proposed Solution																
Detailed Design Phase																
Design Process B.O.M.s																
Manufacturing/ Assembly Process																
Risk Analysis																
Verification																
Validation																
Delivery Phase																
User Manual																
Waiver Release & Hold Harmless																

Customer Satisfaction Questionnaire																
Delivery Checklist																
Approvals																
Service/Maintenance P hase **Retirement or** **Redesign**																

Appendix B: Overall Project Design

B.1 Project Identification

Description: Each document is should include a *Project Identification* outlining and describing the specific problem that the *team* will be addressing and some preliminary description of the overall function of the end-product (which preliminary needs will be fulfilled, describe what the solution/product will do, and how the solution/product will solve the preliminary problem/need). Descriptions should be based on some educated support (secondary research, gauging small sample of the population, peer-reviewed sources/citations, etc.). Explain why the team's problem is worth addressing (socioeconomic or geopolitical impact).

Provide a description of Smart City's community project partner and some of the needs (portray how *your team* views and identifies the partner and their needs). Identify primary/secondary users and stakeholders (description and needs). Project identification of users/stakeholders and needs should be preliminary description based on initial meetings with the project partner and initial need-finding of a small sample of users. Further elaboration should be done in **Overall Project Design** > *Functional Evaluation*.

**Note*: Goal is to identify a specific, compelling need to be addressed

Phase 1: Project Identification	**Status**	**Evidence can be found:**
Conduct needs assessment (if need not already defined)	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.1 Project Charter** ● **4.4.1 POLES Outcomes/Deliverables** ● **4.4.2 Smart City App Outcomes/Deliverables**
Identify stakeholders (customer, users, person maintaining project, etc.)	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.2 Stakeholders**
Understand the Social Context	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.4.1 POLES Outcomes/Deliverables** ● **4.4.2 Smart City App Outcomes/Deliverables**

Define basic stakeholder requirements (objectives or goals of projects and constraints)	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.1 Project Charter**
Determine time constraints of the project	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.4.1 POLES Outcomes/Deliverables** ● **4.4.2 Smart City App Outcomes/Deliverables**

[36] Smart City SharePoint>Spring2017>Smart City Appendix

B.2 Specification Development

Description: Prepare a background review based on initial project partner engagement and conceptual design brainstorming relevant to the users and stakeholders. Discuss requirements and limitations of a prototype based on users/stakeholders, technology, and environmental issues (identify and define the design issues associated with the prototype). Reference any peer reviewed documents to support the team's decisions (refer to Purdue Libraries). Discuss the performance requirements/limitations of the prototype (consider the level of needed functionality of a product and how the product will perform various functions to satisfy needs and activities). Refer to the **Semester Team Information** discussion of cost-analysis breakdown and consider the effects of cost into the performance of a prototype/end-product. Provide some discussion to the possible inter-project limitations that may influence the initial prototype and field testing (waiting on another team to finish a progress, communicating/establishing testing standards, differing user/stakeholder testing targets, etc.).

**Note*: Goal is to understand "what" is needed by understanding the context, stakeholders, requirements of the project, and why current solutions don't meet need, and to develop measurable criteria in which design concepts can be evaluated.

Phase 2: Specification Development	**Status:**	**Evidence can be found:**
Understand and describe context (current situation and environment)	**Status:** *Completed* **Semester:** *Spring 2017*	• **4.4.1 POLES Outcomes/Deliverables** • **4.4.2 Smart City App Outcomes/Deliverables**
Create stakeholder profiles	**Status:** *Completed* **Semester:** *Spring 2017*	• **4.2 Stakeholders**
Create mock-ups and simple prototypes: quick, low-cost, multiple cycles incorporating feedback	**Status:** *Completed* **Semester:** *Spring 2017*	• **6 Current Design**
Develop a task analysis and define how users will interact with project (user scenarios)	**Status:** *To be done* **Semester:**	
Identify other solutions to similar needs and identify benchmark products (prior art)	**Status:** *Completed* **Semester:** *Spring 2017*	• **6 Current Design**
Define customer requirements in more detail; get project partner approval	**Status:** *Completed* **Semester:** *Spring 2017*	• **6 Current Design**
Develop specifications document	**Status:** *Completed* **Semester:** *Spring 2017*	• **6 Current Design**

Establish evaluation criteria	**Status:** *To be done* **Semester:**	

B.3 Conceptual Design

Description: Create a system map that outlines the connections and functions of every operation and activity of an ideal, initial prototype. The system map should clearly show that each function satisfies some activity or need of the users/stakeholders (system map forces designers to archive need satisfaction and includes the users/stakeholders to the design table). An example of an optimal system map can be presented in lab.

Develop a simple prototype that can be built in a few minutes and utilize the initial prototype as a basis model to further improve upon. Do not create a fully-somewhat functional prototype. Tap into the engineering imagination and imitate the functionality and end-goal standards of an end-product to the initial prototype. Document your initial prototype with pictures and a description of functions based on a system map. Return to the community project partner and discover if the team's *Project Specification* is appropriate and satisfies the users' and stakeholders' needs. Investigate the user's interaction with the initial prototype and note the smallest interactions details. Question and gauge individual interactions as they may provide insight into a "problem" that the team may have overlooked or unpredicted. An initial functional evaluation with the user allows the design team to get a sense of how someone might interact with your initial prototype and may reveal suggestions into the *Functional Prototype*.

**Note*: Goal is to expand the design space to include as many solutions as possible. Evaluate different approaches and selecting "best" one to move forward. Exploring "how".

B.4 Detailed design

Phase 3: Conceptual Design	Status:	Evidence can be found:
Complete functional decomposition	**Status:** *Completed* **Semester:** *Spring 2017*	● **6 Current Design** Document Smar City - Spring 201
Brainstorm several possible solutions	**Status:** *Completed* **Semester:** *Spring 2017*	● **6 Current Design**
Prior Artifacts Research	**Status:** *Completed* **Semester:** *Spring 2017*	● **4.3 Project Objectives**
Create prototypes of multiple concepts, get feedback from users, refine specifications	**Status:** *Completed* **Semester:** *Spring 2017*	● **6 Current Design**
Evaluate feasibility of potential solutions (proof-of-concept prototypes)	**Status:** *Completed* **Semester:** *Spring 2018*	
Choose "best" solution	**Status:** *Completed* **Semester:** *Spring 2017*	● **6 Current Design**

Description: Develop a physical model for functional evaluation. Refer to the system map and initial functional evaluation with the project partner and develop a physical functioning prototype. Document with pictures and descriptions as appropriate. In order to test the interactive and usability functionality of the team's prototype, develop a prototype that integrates some level of activity that the users will be able to learn the objective of the team's end-product.

Ideally recruit 7 users for an acceptable functional evaluation. One suggestion of testing may be to allow testers interact with the prototype through self-exploration (let the testers play with it). The main objective of functional evaluation is to obtain an initial assessment of the "goodness" of the need-finding, system map, and design process. Sitting down to gauge a small sample of testers will allow engineers to uncover

unforeseen problems not identified within the scope of functional decomposition. For a functional evaluation to yield effective results, try to remove any team's bias in asking questions and imagine being in the shoes of an "everyday" user. For documentation, write up a description of how the team conducted field trials (what did you ask the participants to do? how did you recruit testers? how many testers? etc.), some secondary research on demographic information, and data/conclusions from field trials.

After an "effective" *Functional Evaluation*, return to the drawing table and implement changes based on data/conclusions. Redesign the *Functional Prototype* (simplifying functions, aesthetic adjustments, new activities, etc.) and incorporate the redesign into a new functional prototype. A *Revised Prototype* should be ideal product that an engineer would like to propose to his/her superiors for further research/investment. A revised prototype should be able to communicate the appearance (what the team had in mind for a product; shape, size, weight, color, etc.) for the redesign.

Field testing differs from functional evaluation. Field testing determines if the executable functions of a prototype yields acceptable results (does the prototype work to produce data?). For documentation, write up a description of how the team conducted field trials (what did you ask the participants to do? how did you recruit testers? how many testers? etc.), some secondary research on demographic information, and data/conclusions from field trials. *Please recruit different testers*.
Go back to project prospectus and project specification and access how well your redesigned solution compares to the team's original goals for the project. Identify and document what improvements have been made (include some reasoning why improvements had to be made/ why "problem" was not identified) and suggest further improvements for future EPICS redesign (suggestions should be optimizations goals to streamline the existing design framework established in this team's design document). Describe what your team views as the "best" solution for the users/stakeholders. Clearly state what final improvements are needed and why they are essential to the success of the users/stakeholders based on the team's findings in functional evaluation and redesign.

**Note*: Goal is to design working prototype which meets functional specifications.

Phase 4: Detailed Design	**Status:**	**Evidence can be found:**
Bottom-Up Development of component designs	**Status:** *Completed* **Semester:** *Spring 2017*	● **6 Current Design**
Develop Design Specification for components	**Status:** *In-Progress* **Semester:** *Spring 2018*	● **6 Current Design**
Design/analysis/evaluation of project, sub-modules and/or components (freeze interfaces)	**Status:** *In-Progress* **Semester:** *Spring 2018*	● **6 Current Design**

Design for Failure Mode Analysis (DFMEA)	**Status:** *To be done* **Semester:**	
Prototyping of project, submodules and/or components	**Status:** *To be done* **Semester:**	
Field test prototype/usability testing	**Status:** *To be done* **Semester:**	

B.5 Delivery

Description:

**Note*: Goal is to refine detailed design so as to produce a product that is ready to be delivered! In addition, the goal is to develop user manuals and training materials.

Phase 5: Delivery	**Status:**	**Evidence can be found:**
Complete deliverable version of project including Bill of Materials	**Status:** *To be done* **Semester:**	
Complete usability and reliability testing	**Status:** *To be done* **Semester:**	
Complete user manuals/training material	**Status:** *To be done* **Semester:**	
Complete delivery review	**Status:** *To be done* **Semester:**	
Project Partner, Advisor, and EPICS Admin Approval	**Status:** *To be done* **Semester:**	

B.6 Service / Maintenance

Name of Servicer/Maintenances	Date	Service/Maintenance Done Notes