March 26, 2021

Dr. Craig Scratchley and Dr. Shervin Jannesar
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

Re: ENSC 405W/440 Design Specification for Cylindrotech's Plot-Hole

Dear Dr. Scratchley and Dr. Jannesar,

The design specifications of Cylindrotech's Plot-Hole are enclosed. The goal of Plot-Hole is to utilize untapped pothole detection methods and simplify the pothole reporting process. Plot-Hole's streamlined capabilities involving reporting pertinent information to the city are coordinated with a website that is backed by a database. The website displays potholes on a synchronized map, where users can add, edit, and remove potholes.

The design document provides details of feature requirements of Plot-Hole in its proof-of-concept, prototype, and final product stages. The document showcases the design choices made for the product while justifications and other considerations are detailed. The appendices contain notable information: design options, user interface considerations, and plans for testing.

Cylindrotech continues to be comprised of a team of five engineers: Jonathan Wong and Yuxin (River) Zhang in Electronics Engineering; Evan Lam and Bifei (Elvira) Huang in Computer Engineering; and Samantha Betts in Engineering Physics. The team's small size and diverse skillset has continued to provide every member a direct, vital role in influencing the Plot-Hole project.

Thank you for acknowledging Cylindrotech's design specification. For further communication and questions, please send an email to Cylindrotech at lamevanl@sfu.ca.

Sincerely,

Evan Lam
Chief Communications Officer
Cylindrotech

Design Specification
Plot-Hole

Team Members:

Evan Lam
Bifei (Elvira) Huang
Jonathan Wong
Yuxin (River) Zhang
Samantha Betts

Contact Person:

Evan Lam — lamevanl@sfu.ca

Submitted To:
Dr. Craig Scratchley, ENSC 405W
Dr. Shervin Jannesar, ENSC 405W
Dr. Andrew Rawicz, ENSC 440
School of Engineering Science

Simon Fraser University

Revision 1.0

Revision Date: 2021-03-26

Abstract

The document covers the design specification of the product, Plot-Hole, that Cylindrotech is building. Plot-Hole utilizes image-based deep learning and ultrasonic sensors to detect and determine the size of potholes. The system uses GPS data to display location and geolocational time on an online database-synchronized mapping system. The design document provides a survey of the system through an introduction that includes scope, expected challenges, and background. The system design overview further details subcomponents of the system through text and block diagrams. The design specifications through different development stages are then outlined in terms of hardware, electrical, and software subsystems. The conclusion of the document is presented and followed with appendices. Appendix A details and shows justification for separate design options in each product subsystem. Appendix B details user interface and appearance considerations in depth. The final section of the document, Appendix C, details a comprehensive test plan.

Revision History

Revision	Revision Description	Date
1.0	First revision of document finished	2021-03-26

Table of Contents

List of Figures

List of Tables

Glossary

AI - Artificial Intelligence. In computer science, artificial intelligence describes a machine that learns (Machine/Deep Learning) and exhibits problem-solving.

API – Application Programming Interface. A software intermediary that allows two applications to talk to each other.

CMOS – Complementary Metal–Oxide–Semiconductor. Common technology for image sensors.

COCO - Common Objects in Context. Object detection dataset.

CSA - Canadian Standards Association. A Canadian national standards organization that develops standards for Canada across numerous industries.

CSV – A file format where data can be stored in a spreadsheet-like format.

Deep Learning - The use of mathematical models to process input. The models are "trained" with hidden layers to analyse datasets.

Edge Device - A device that does real-time data processing at the source.

FOV - Field of View. In terms of vision, FOV describes the extent of the observable world that is seen at any given moment by a camera lens, an organism's eyes, or by an observing device such as binoculars.

FPN – Feature Pyramid Network. A solution for building recognition systems in deep learning networks.

FPS - Frames per second (Frame rate). The rate at which images are processed. Videos commonly play in 30 FPS or 60 FPS.

GPS - Global Positioning System. A navigation system that uses satellites to provide devices with geolocation (latitude and longitude) and time.

GPX – A file format that GPS data can be returned in. Can be converted to TXT easily.

HQ – High Quality. Typically refers to a relatively high resolution (pixel density).

IEC - International Electrotechnical Commission. An international standards organization that publishes electronic and technological standards.

IEEE - Institute of Electrical and Electronics Engineers. An engineering association that seeks to advance engineering worldwide. The IEEE-SA (Standards Association) is an operating unit within IEEE that develops and publishes standards relating to engineering.

ISO - International Organization for Standardization. An international standard-setting body composed of representatives from various national standards organizations.

MAP – Mean Average Precision. A metric for measuring the overall accuracy of a deep learning model.

MobileNetV2 – The second version of the neural network model MobileNet.

RGB Camera – A CMOS sensor camera that can acquire coloured images (Red, Green, Blue).

SSD – Single-Shot Detector. A deep learning model that detects an object a single glance.

TPU – Tensor Processing Unit. An AI acceleration application specific integrated circuit.

TXT – A file format that stores text. It is a universal standard that is basic and used by a wide variety of programs.

UART – Universal Asynchronous Receiver-Transmitter. Device used for serial communication.

USB - Universal Serial Bus. An industry standard interface between computers and peripherals that allows for data transfer, charging, networking, and other functions. The standard is maintained and continually updated by the USB Implementers Forum.

1 Introduction

Plot-Hole is Cylindrotech's state of the art pothole detection system for use on city roads. The product consists of two systems: the vehicular attachment, and the web application. Both systems work interconnectedly to give informative data about citywide potholes, allowing for categorization and highlighting to the user which potholes require the most attention. The vehicle system attains dimension, depth, location, and timestamp information regarding a pothole. Whereas the website provides an interactive map for visualization of the pothole locations. The website also allows the user to manually add and remove potholes.

1.1 Background

In 2016 the Canadian Automobile Association reports that on average Canadians spend around $1.4 billion on pothole associated car damage [1]. These damages include tire punctures, bent rims, or even rendering the car undrivable by destroying the suspension [2].

Currently cities rely on local citizens to report potholes. This method is inefficient because people are not incentivized to report these potholes unless they are personally affected. If a pothole is not reported when it is small, it can grow and lead to more serious problems in the future. With an automated pothole detection system, potholes of all sizes can be logged without reliance on input from local city drivers.

1.2 Intended Use

Plot-Hole is designed to be used on a street sweeper since they move at a slow enough pace for sufficient data collection, as well as travel the majority of the main city roads. The main operator of the product will be the driver of the street sweeper and data will be collected during the commute. Once the user has felt sufficient data is collected, they can unplug the external drive attached to the vehicle system and upload the file to the website for viewing of the collected pothole information.

1.3 Scope

The purpose of this document is to supply the reader with an in-depth knowledge of the technical aspects of Plot-Hole. Detailed explanations of design choices will accompany diagrams to communicate the roles of each subsystem within the main product.

1.4 Expected Challenges

Challenges arise during the development for the vehicle system of this project. We do not have access to a street sweeper therefore the enclosure and ultrasonic array will be mounted to a normal vehicle for our proof-of-concept and prototype models. Regarding mounting the system on top of the street sweeper. The user may have difficulty reaching the battery and the storage device. For the final product, the group may develop a product that is used inside the vehicle that harnesses the vehicle's power through the internal auxiliary power outlet.

The different stages in the development process are shown in Table 1.1 below:

1.5 Design Classification

The format of the Design ID will be as follows:
Des {Section number} {Design number} - {Development stage}

Table 1.1 Stage Classification

Development Stage	Design ID letter
Proof of Concept	C
Prototype	P
Final Product	F

2 System Design Overview

Plot-Hole's vehicular attachment is also broken down into two subsystems. First, there is a RGB camera that feeds images to a trained deep learning system to detect potholes up to a range of 13.3m. Once a pothole is classified, an on-board GPS sensor will mark its location and trigger the ultrasonic system to begin its work. This setup along with the rest of the electronic components including the power supply will be mounted in an enclosure on the roof of the street sweeper.

The second subsystem is the aforementioned ultrasonic array. Made of up 10 ultrasonic sensors, this horizontal array is tasked with detecting the size of each pothole. Dimensional measurements of the width and depth of the pothole can be obtained with greater accuracy than an image-based approach. This system is located at the front of the vehicle and will be mounted at a lower height on the front bumper of the street sweeper. Based on its angled geometry, this array has a maximum horizontal detection distance of 227 cm. This system will have a wired connection to the roof enclosure of the deep learning system. Figure 2.1 presents an overview of the hardware system design. The core of the system will be powered by a Raspberry Pi, with sensors, a camera, and a deep learning accelerator connected.

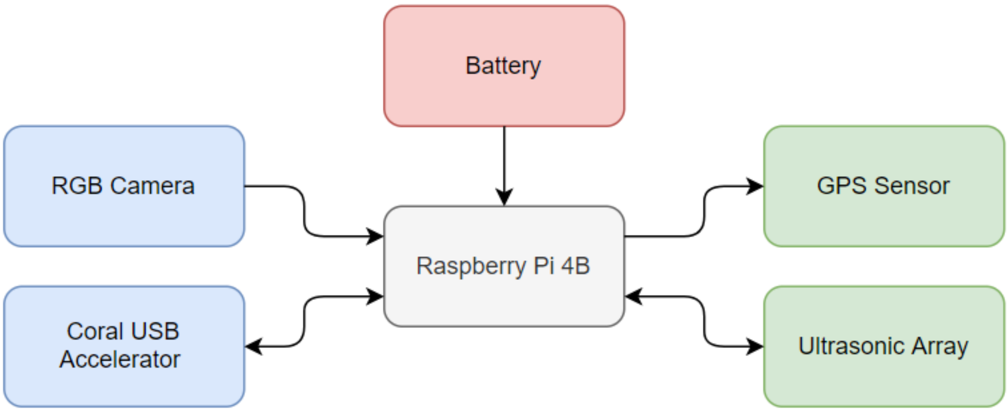


Figure 2.1 System Design

Plot-Hole's web application is focused on being an easy-to-use tool for reporting potholes and integrating into the local municipality's reporting processes. By importing the collected data by Plot-Hole's automatic vehicular attachment system, the web application can display the location, size, and date of all potholes. It also uses a Google Maps API for a superior and familiar user experience. Figure 2.2 on the next page illustrates a context diagram that shows data flow between different systems.

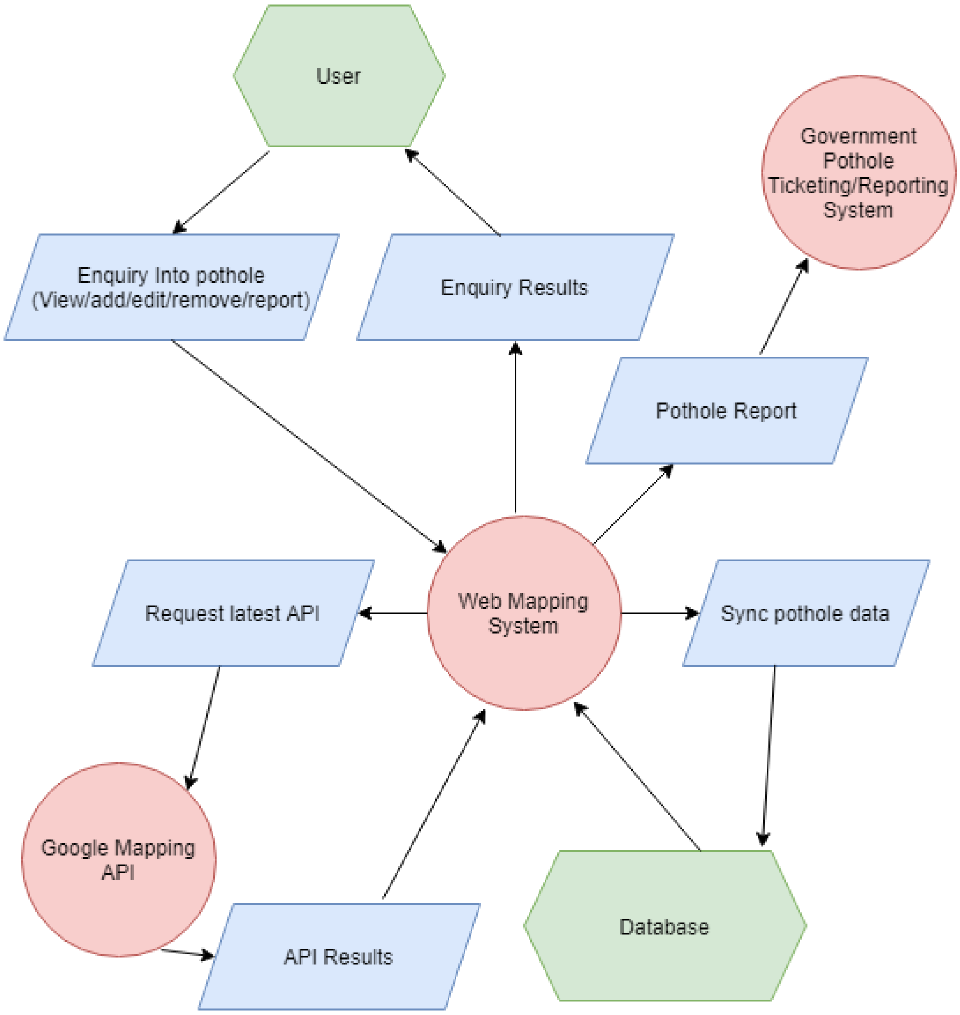


Figure 2.2 Web Mapping System: Context Diagram

In Figure 2.2, the circles represent different systems. The central diagram is defined as being the product's web mapping system. The Google Maps API is a system that must be used to acquire the latest map data (road changes, location name changes, etc.). The top-right system is the currently existing municipal pothole reporting system.

Hexagons represent data endpoint systems. The user and the database supply information to the web mapping system, keeping the information up to date. The database syncs data between different users and different endpoints. Parallelograms represent the information that is sent between endpoints and systems. Data may need to flow from one system to another before arriving at its intended location. For example, if the user clicks on a pothole to view it, the request enters the web mapping system. From the web mapping system, the data is synced with the database and the web mapping system displays the most up-to-date data back to the user.

3 Hardware Design Specification

3.1 Ultrasonic System

The ultrasonic system is made up of a horizontal array of ultrasonic sensors and a Raspberry Pi. The ultrasonic array is primarily horizontal for the most accurate readings in the front of the vehicle, however it also curves around the sides of the vehicle for a larger FOV.

The array is mounted with a downward angle of 60 degrees. The front bumper of a street sweeper is approximately one-fifth of the total height, which is approximately 2.35m [3]. Therefore, calculating the maximum range of the ultrasonic array,

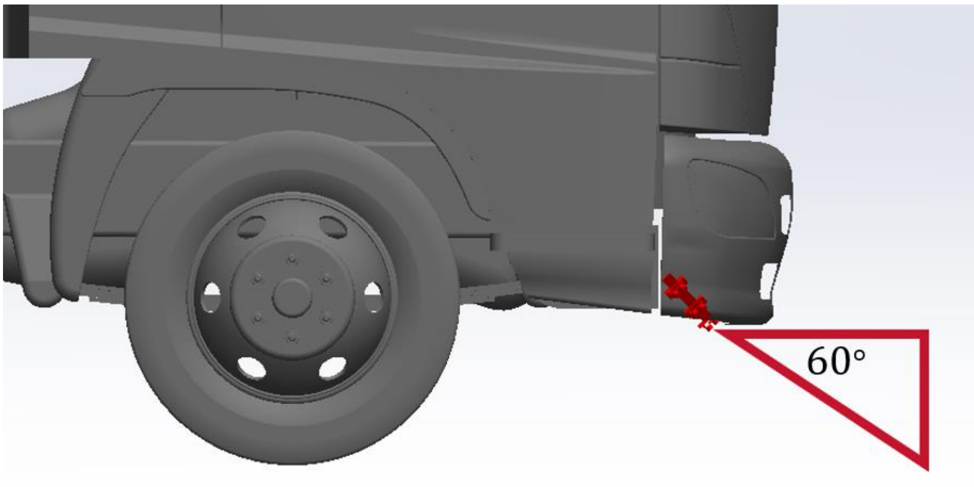


Figure 3.1 Ultrasonic Array Range [4]

Using the geometry in Figure 3.1,

$$Ultrasonic\ Array\ Range = 0.2 \times 2.35m/\tan(60°) = 0.271m = 27.1cm \quad \textit{(Equation 3.1)}$$

The ultrasonic array is also comprised of a mainly horizontal section, with smaller angled sides. The width of the street sweeper is 167cm, therefore the horizontal section of the array will be 175cm wide [3]. Angled bends of 30 degrees on both sides will have a shorter section of 30cm.

The number of sensors is determined by the pothole size classification according to VanConnect. The three sizes listed are: Small (30cm), Medium (30-60cm), Large (60+ cm) [5]. Therefore, to detect a minimum resolution of 30cm, the distance between each sensor should be 15cm.

$$Horizontal\ Distance = 175cm + 2 \times 30cm \times cos(30°) = 226.96cm \quad \textit{(Equation 3.2)}$$
$$Number\ of\ Sensors = 226.96cm/15cm = 16 \quad \textit{(Equation 3.3)}$$

With a horizontal distance of approximately 227cm, 16 sensors are needed. Figure 3.2 on the next page represents the block diagram associated with the ultrasound system. When the deep learning system classifies a pothole, the ultrasonic system will become active and attempt to determine the depth and width of the pothole. If successful, it will save a copy of the pothole's data.

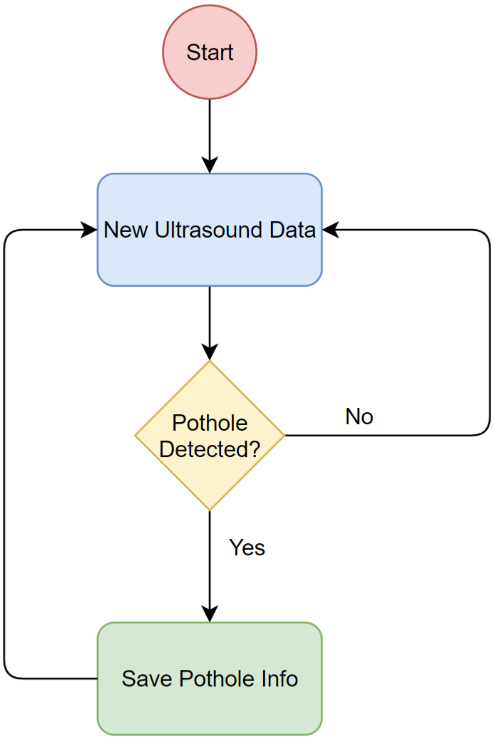


Figure 3.2 Ultrasonic Data Acquisition Algorithm

3.1.1 Ultrasonic Sensor

The ultrasonic sensor chosen in Figure 3.3 is the RCWL-1601 capable of operating with 3-5.5V [6]. It can operate within 0 to 70°C, and measure distances between 2cm to 450cm [6]. This sensor is also RoHS3 compliant [6].

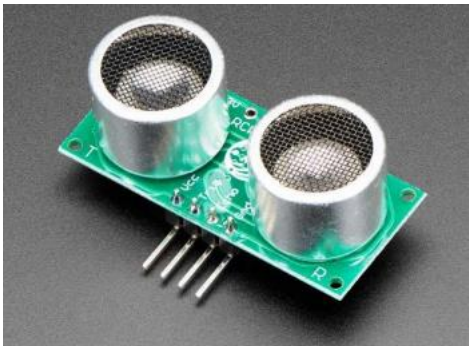


Figure 3.3 RCWL-1601 Ultrasonic Sensor [6]

Table 3.1 Design Specifications for Ultrasonic Sensors

Design ID	Description	Corresponding Requirement
Des 3.1.1.1 C	The ultrasonic sensor will be powered by 3.3V.	Req 3.2.1 C
Des 3.1.1.2 C	The ultrasonic sensor will be work in the ultrasonic range of 40 kHz.	N/A
Des 3.1.1.3 C	The ultrasonic sensor will be connected to the Raspberry Pi by Trigger and Echo.	Req 3.2.2 C
Des 3.1.1.4 C	The ultrasonic sensor will transmit and receive its own signal.	Req 3.2.2 C

3.1.2 Ultrasonic Array

Table 3.2 Design Specifications for Ultrasonic Array

Design ID	Description	Corresponding Requirement
Des 3.1.2.1 C	The ultrasonic array will distinguish potholes between 30cm, 30-60cm, and greater than 60cm.	Req 3.2.3 P
Des 3.1.2.2 C	The ultrasonic array will cover a horizontal distance of 227cm.	N/A
Des 3.1.2.3 C	The ultrasonic array will have a downward angle of 60 degrees.	N/A
Des 3.1.2.4 C	The ultrasonic array will consist of 16 sensors.	Req 3.2.3 P
Des 3.1.2.5 C	The ultrasonic array will be activated upon detection of a pothole by the deep learning system.	N/A

3.2 Enclosures

The enclosures are divided into two parts: the ultrasonic array enclosure and the roof-mounted camera and electronics enclosure. The connection between these two enclosures will be covered in Section 4.1 Wiring.

3.2.1 Ultrasonic System Enclosure

The ultrasonic system enclosure includes individual mounting solutions for each sensor as well as the enclosure for the entire array. The combination of these two will provide a secure platform for the ultrasonic sensors, attachment points for mounting to a street sweeper, and protection of the PCB and sensitive components.

Table 3.3 Design Specifications for Ultrasonic System Enclosure

Design ID	Description	Corresponding Requirement
Des 3.2.1.1 C	The ultrasonic system enclosure will secure all ultrasonic sensors.	Req 3.4.3 F
Des 3.2.1.2 P	The ultrasonic system enclosure will house the wires to each ultrasonic sensor.	Req 3.4.1 F
Des 3.2.1.3 P	The ultrasonic system enclosure will consist of modular parts.	Req 3.4.3 F
Des 3.2.1.4 F	The ultrasonic system enclosure will securely mount to a street sweeper.	Req 3.4.3 F

The ultrasonic system enclosure will feature modular components so that faulty sensors can be replaced independently.

3.2.2 Camera and Electronics Enclosure

The camera and electronics enclosure will provide a stable mounting solution for the microcontrollers, RGB camera, power source, storage, and GPS sensor. It will be mounted at the top of the street sweeper.

Table 3.4 Design Specifications for Camera and Electronics Enclosure

Design ID	Description	Corresponding Requirement
Des 3.2.2.1 C	The camera and electronics enclosure will provide housing for the microcontrollers, RGB camera, power source, storage, and GPS sensor.	Req 3.4.1 F
Des 3.2.2.2 F	The camera and electronics enclosure will include a rooftop mounting option for the street sweeper.	Req 3.4.3 F
Des 3.2.2.3 P	The camera and electronics enclosure will protect the sensitive electronic components.	Req 3.4.1 F

3.3 RGB Camera

The camera will provide the visual data necessary to run the deep learning pothole object detection algorithm. To satisfy requirement specifications Req 3.1.1 and Req 3.1.3, a camera with good resolution and high FPS is needed. Furthermore, the camera must be able to operate in outside weather conditions. The processing power should also be considered, high resolution means more processing power is needed and will lead to slower inference time for the deep learning algorithm. The highest resolution is not necessarily the best as during training. The neural network images are resized to around 300x300, and down sampling may be required for the incoming video stream of the camera to run inference. Due to these reasons, the Raspberry Pi HQ camera, shown in Figure 3.4, was chosen out of the design options listed in Appendix A.1.2. Its modifiable resolution and frame rate can be experimented with accordingly to get the best results for a slow-moving vehicle. Also, its ability to fit a multitude of lenses with different parameters can improve the focal view of the road.

Figure 3.4 Raspberry Pi HQ Camera [7]

Using the geometry in Figure 3.5 and calculating the maximum range of the camera with a 10-degree downward angle towards the road, [4]

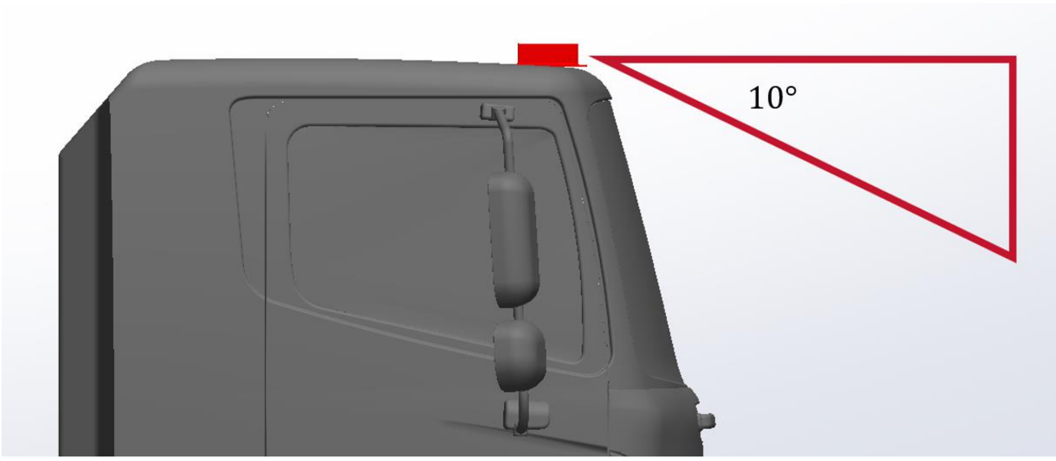


Figure 3.5 RGB Camera Range [4]

$$RGB\ Camera\ Range = 2.35m/\tan(10°) = 13.3m$$ *(Equation 3.4)*

Therefore, the camera and the deep learning system will be able to see potholes approximately 13.3m ahead.

Table 3.5 Design Specifications for RGB Camera

Design ID	Description	Corresponding Requirement(s)
Des 3.3.1 C	The camera will have a frame rate of 30fps.	Req 3.1.1 F
Des 3.3.2 C	The camera will have a resolution of 1080p.	Req 3.1.3 P
Des 3.3.3 P	The camera will be placed in a secure enclosure.	Req 3.1.2 C Req 3.1.4 P Req 3.1.5 P

3.4 GPS Sensor

The pothole detection system consists of one GPS sensor. This sensor is used to make sure the system is able to locate the detected potholes.

The sensor chosen is the NEO-7M. It has an accuracy of 2.5m, and an update rate of 5Hz. It can be operated between temperatures of -40°C and 85°C. The antenna mode is active and it is compatible with the Raspberry Pi using the UART pins.

Figure 3.6 NEO-7M GPS sensor [8]

The block diagram in Figure 3.7 on the next page represents the working algorithm of the GPS sensor.

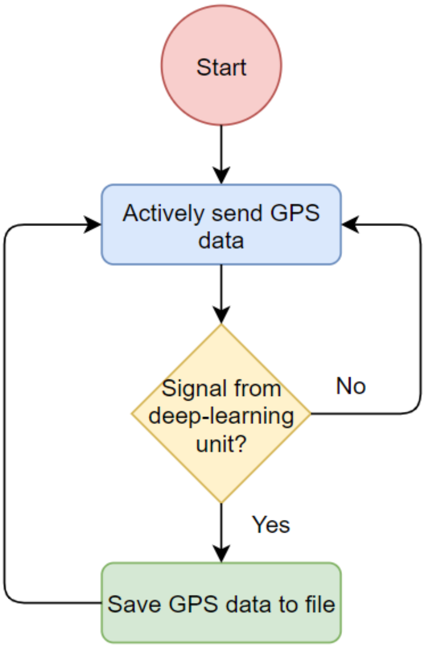


Figure 3.7 GPS Data Acquisition Algorithm

Table 3.6 Design Specifications for GPS Sensor

Design ID	Description	Corresponding Requirement
Des 3.4.1 C	This GPS sensor will actively return location and time data under a 1Hz frequency.	N/A
Des 3.4.2 C	This GPS sensor will be communicating with a Raspberry Pi.	N/A
Des 3.4.3 C	This GPS sensor will be powered by 3.3V.	N/A
Des 3.4.4 P	This GPS sensor will be secured inside the enclosure.	N/A
Des 3.4.5 C	The GPS sensor will send longitude and latitude data to Raspberry Pi through the TX pin.	Req 3.3.2 P

3.5 Neural Network Processing Unit

The pothole detection algorithm uses deep learning for object detection. This technique relies on a proper neural processing unit to implement the necessary arithmetic and logic for the object detection algorithm. To cover Req 5.6 and Req 5.7, the processing unit must allow for a fast inference time and support a neural network that will provide high accuracy results. In addition, it must be compatible communicating with our choice of microcontroller, the Raspberry Pi, without consuming high amounts of power. Due to these constraints the Google Coral USB accelerator was chosen as seen in Figure 3.8, because it has the fastest inference time and the lowest power consumption of all the design options listed in Appendix A.1.4. Its speed is due to the Edge TPU, which is designed specifically for convolutions. Additionally, the USB accelerator has easy connectivity to the Raspberry Pi. [9]

Figure 3.8 Coral USB Accelerator [9]

Table 3.7 Design Specifications for Neural Network Processing Unit

Design ID	Description	Corresponding Requirement
Des 3.4.1 C	The USB accelerator will be connected to the Raspberry Pi via USB 3.0 to maximize inference performance time [10].	Req 5.6 C
Des 3.4.2 C	The connected USB cable will supply at least 500mA at 5V [10].	Req 4.1.1 P
Des 3.4.3 C	The USB accelerator will be operated at the reduced clock frequency in order to operate in an ambient temperature of up to 35°C [10].	Req 3.5.3 P

4 Electrical Design Specification

4.1 Wiring

Wiring serves an important role in connecting all components to ensure they communicate well with each other. The connections of the ultrasonic sensor array and the GPS sensor to Raspberry Pi is as shown below in the schematic. The RGB camera will be connected to the Pi through the ribbon cable.

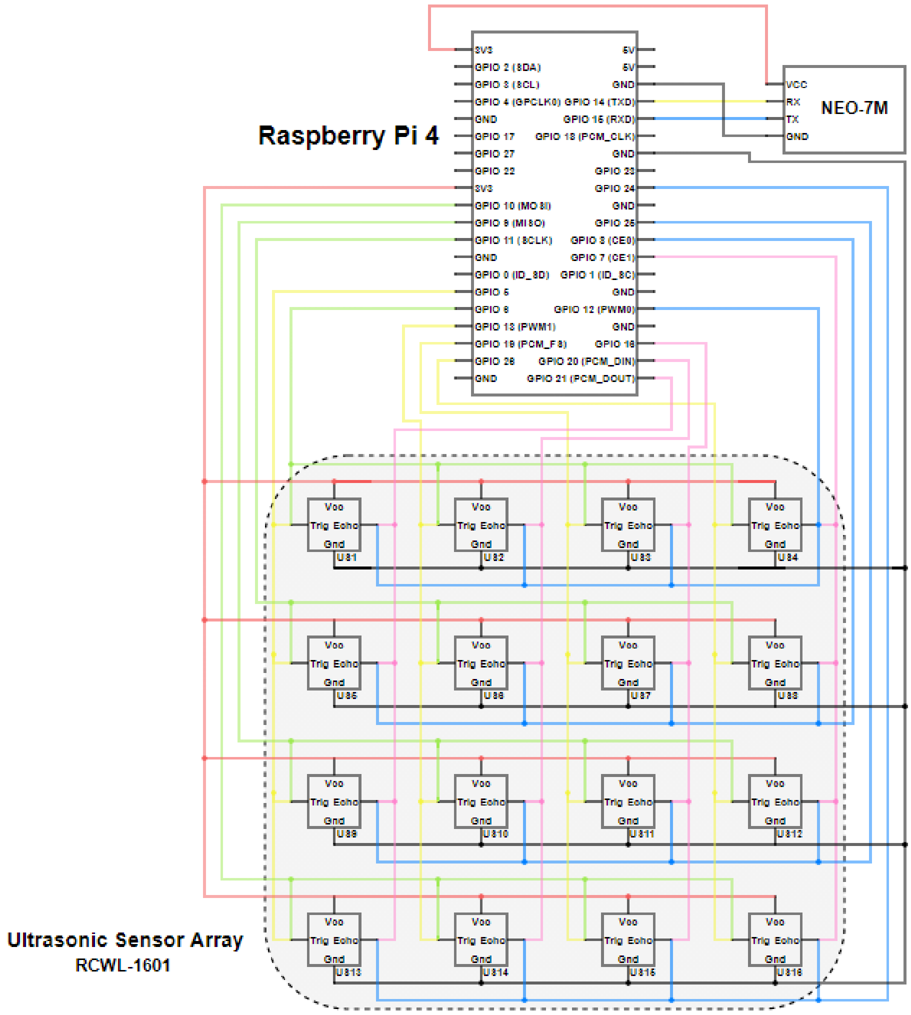


Figure 4.1 Schematic of Raspberry Pi GPIO Connection

Table 4.1 Design Specifications for Wiring

Design ID	Description	Corresponding Requirement
Des 4.1.1 C	Wiring will be organized in an orderly manner.	Req 4.1.1 P
Des 4.1.2 P	Wiring between the controlling unit and ultrasonic sensor array will be kept inside the enclosure or fixed onto a surface.	Req 4.1.1 P

4.2 Power Supply

The Raspberry Pi will be the source of power for the ultrasonic sensor array and the GPS sensor, both operates under 3.3V. It also provides power for the Coral USB accelerator, which requires at least 500mA at 5V.

The Raspberry Pi requires a good quality USB-C power supply, delivering 5V at 3A since the downstream USB devices consume more than 450mA [11]. The system is required to be able to operate for at least 8 hours, the power capacity in milliamp hour required is 24000mAh. The calculation is as shown below:

$$Charge\ (in\ mAh) = 3000mA \times 8h = 24000mAh \quad \textit{(Equation 4.1)}$$

Figure 4.2 Charmast 26800mAh Portable Charger [12]

Lithium-polymer and Lithium-ion batteries can both be affected by problems of over-charging, over-discharging, overheating, etc. While the Lithium-ion batteries can be explosive in failing conditions, although very rare, Li-Po batteries are relatively safer with a chance of deformation. Power banks are advised not to be charged below 0°C or discharged over temperature of 40°C. The total power consumption was calculated above in Equation 4.1. The power bank chosen has a capacity of 26800mAh, which is enough to cover the total power consumption.

Table 4.2 Design Specifications for Power Supply

Design ID	Description	Corresponding Requirement
Des 4.2.1 C	A power bank will be used to provide power for the whole system.	Req 4.2.2 F Req 4.2.4 P
Des 4.2.2 C	The USB-C output will be used for powering the system.	Req 4.2.1 C
Des 4.2.3 C	The 3.3V pin on Raspberry Pi will be used to power the sensors.	Req 4.2.1 C
Des 4.2.4 C	The USB 3.0 port on Raspberry Pi will be used to power Coral USB accelerator.	Req 4.2.1 C
Des 4.2.5 C	The camera will be connected and powered via the designated Pi camera pinout.	Req 4.2.1 C

5 Software Design Specification

5.1 Web application

The system's user interface state chart is shown below in Figure 5.1. The red circle shows the start point. States are represented with rounded rectangles. The statements enclosed in square brackets are guard statements; when the action inside the brackets occurs, the chart moves from one state to another.

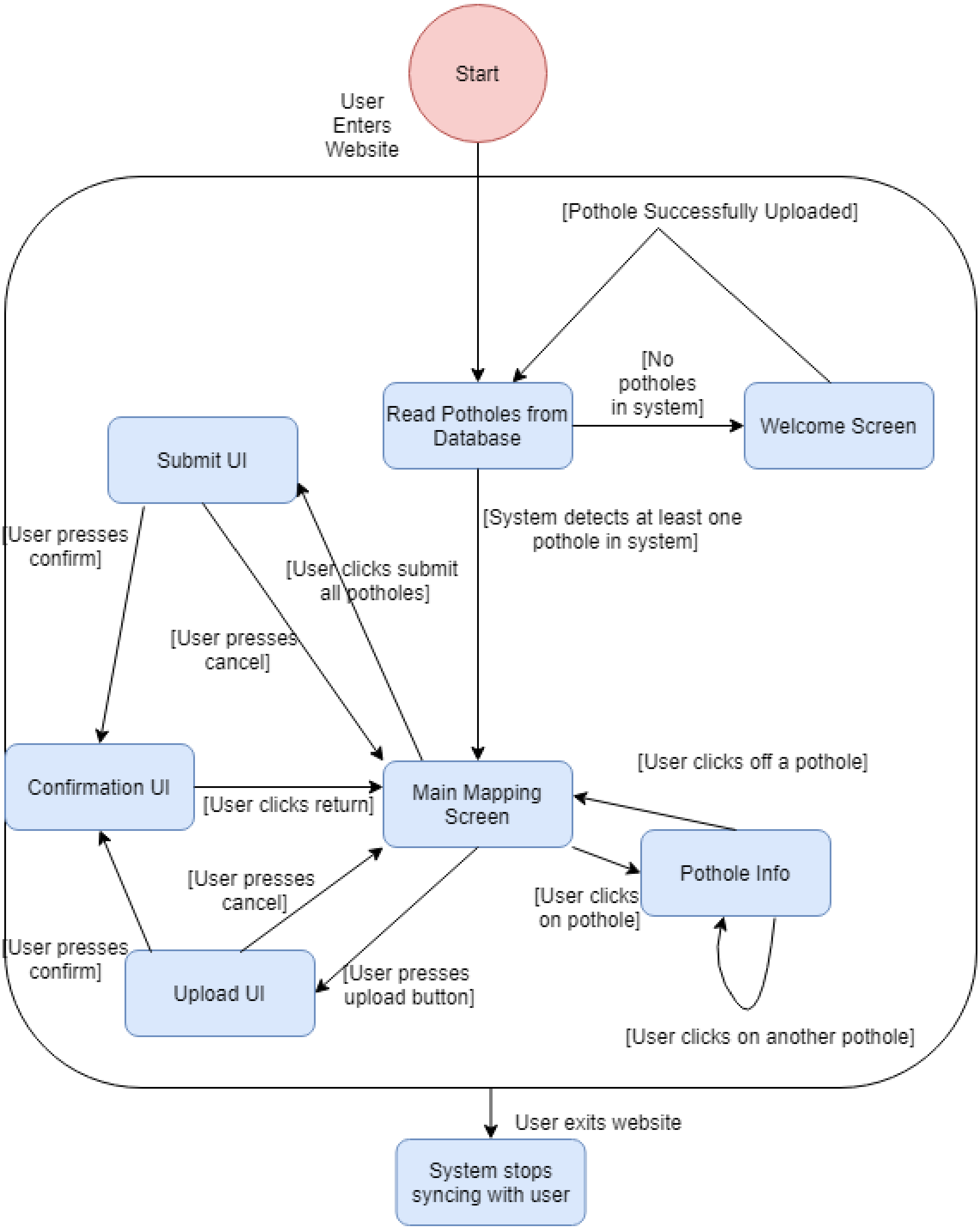


Figure 5.1 Web Application: User Interfaces State Chart

When considering other options (further justification in Appendix A.2.1). The Google Maps API is the most chosen web API for mapping [13]. Its documentation is more extensive than other mapping APIs. The Place API allows place information to be retrieved, which differentiates Google's API from open-source mapping APIs. Information on businesses, parks, and other locations is kept up to date by users and Google's large information databases. The Google Maps API was chosen for its ubiquitous usage on the world wide web.

The design choices are based on requirements for a mapping and ticketing system. Table 5.1 illustrates the design choices made by Cylindrotech.

Table 5.1 Design Specifications for Mapping

Design ID	Description	Corresponding Requirement
5.1.1 C	Potholes are represented on a Google map with pothole-stylized markers.	Req 5.2 C
5.1.2 C	Clicking the Upload Text File button to allow users to upload a text file.	Req 5.5 C Req 5.13 P
5.1.3 P	Clicking the Upload Manually button jumps to a page where the user can enter information of potholes into text fields.	Req 5.5 C
5.1.4 P	Hovering on the marker will show preview information of pothole: Location, time, image (if an image exists), and uploader (if applicable).	Req 5.3 C Req 5.5 C
5.1.5 C	Clicking on a pothole marker, detailed information will be shown in a sidebar menu.	Req 5.5 C
5.1.6 C	Clicking the remove button causes the marker and pothole information to be removed.	Req 5.5 C
5.1.7 C	Clicking the edit button in pothole information allows the user to enter data into text fields.	Req 5.5 C
5.1.8 P	Click Submit and the information of the pothole will be sent to the programmed local municipality's website. The user is met with a success screen or failure screen..	Req 5.5 C Req 5.13 P
5.1.9 P	Clicking "Submit All" sends the information of all potholes to the programmed local municipality's website. The user is met with a success screen or failure screen.	Req 5.5 C Req 5.13 P
5.1.10 P	Clicking "Filter" allows users to sort potholes by a specific range of time.	Req 5.5 C Req 5.11 P
5.1.11 C	The web software can be updated or reverted to a previous version by pressing the Version button.	Req 5.4 C Req 5.9 C Req 5.14 F

5.1.12 C	The map uses Satellite, Terrain, and standard road views.	Req 5.5 C
5.1.13 C	The map is 2D (North Up) and can swap to 3D isometric when the user has sufficiently zoomed in.	Req 5.5 C
5.1.14 P	When the webpage first launches, the map zooms out so that all pothole markers are visible.	Req 5.5 C
5.1.15 P	When the map is not yet ready to display to the user, a loading icon is shown to the user.	Req 5.5 C

5.2 Parsing data

Firebase was chosen as the database application of choice. As a Google product, it features simple integration with the Google Maps API. The pricing is competitive with other database software solutions. Google's uptime on their database as well as the easy-to-scale services make it a solid fit for an expanding database. Traffic management features differentiate it from standard database solutions. Relational data is a nice-to-have but mapping systems do not require relational databases. Appendix A.2.2 shows other design options for the database which were considered.

The following table represents data parsing design choices based on previous specified requirements.

Table 5.2 Design Specifications for Parsing Data

Design ID	Description	Corresponding Requirement
5.2.1 C	The GPS coordinates data will be converted to a specific location on the map.	Req 5.3 C Req 5.2 C
5.2.2 P	The pothole data will be stored on the Firebase database.	Req 5.1 C

5.3 Pothole Detection Algorithm

The pothole detection algorithm involves a deep learning object detection model. Out of the choices listed in Appendix A.3, MobileNet V2 was chosen. MobileNet V2 is preferred because it provides the fastest inference time with only a minor decrease in accuracy. For the product's purposes it was decided that an accuracy of 80% is sufficient (based on Req 5.7). The high inferencing time is desired because the algorithm must process the data incoming from the RGB fast enough while in a moving vehicle.

Table 5.3 Design Specifications for Object Detection Algorithm

Design ID	Description	Corresponding Requirement
Des 5.3.1 C	All deep learning processing should be done within the hardware.	N/A
Des 5.3.2 C	The object detection code will communicate with the GPS and ultrasonic sensors.	N/A

Figure 5.2 demonstrates how the object detection algorithm will communicate will other subsystems of the project.

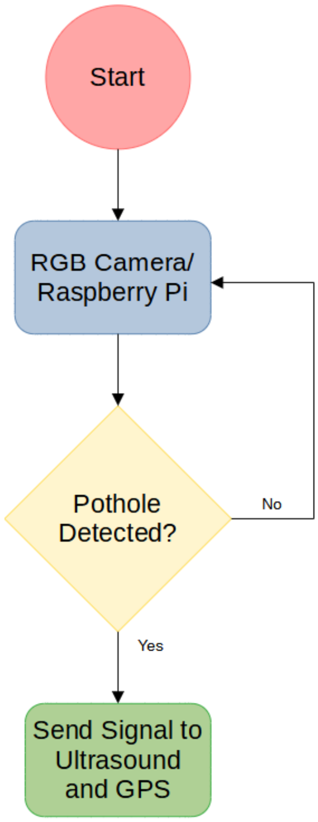


Figure 5.2 Object Detection Algorithm

6 Conclusion

The design choices of Cylindrotech's Plot-Hole were made following the requirement specifications and they are outlined in this document. This document provides detailed information on both hardware and software components of Plot-Hole in its proof-of-concept stage.

There have not been any major changes in our decided functionalities since the requirement specifications. The required hardware and software components remain unchanged. Modified processing unit hardware associated with the deep-learning component is specified. Due to the accessibility issue of a street sweeper, the proof-of-concept prototype will be installed and tested on a regular car.

A test plan is provided in Appendix C. These questionnaires will help with evaluating the performance of each component to make sure the requirements are met. Adjustments may be made to the design throughout the testing phase, with more knowledge gained about the actual performance of the hardware.

Appendix B: User Interface and Appearance Details provides the users an overview of the user interfaces, as this product requires both hardware interaction and software operation from the users. The user can learn about the operation of the whole system with minimal effort.
This document serves as a reference of the development of the Plot-Hole product. It will be kept active as any possible adjustments made in the design of future versions of the product will be recorded.

References

[1] D. Hanno, “How Potholes Damage Your Vehicle,” CTV News, 14 March 2019. [Online]. Available: https://www.ctvnews.ca/autos/how-potholes-damage-your-vehicle-1.4336050. [Accessed 25 March 2021].

[2] “Pothole Season – Prevent Vehicle Damage from Potholes,” CARFAX, [Online]. Available: https://www.carfax.ca/resource-centre/articles/pothole-season-prevent-vehicle-damage-from-potholes. [Accessed 25 March 2021].

[3] Isuzu Truck Canada, “N-SERIES DIESEL,” 2020. [Online]. Available: https://www.isuzutruck.ca/en/app/site/pdf?file=n_brochure.pdf. [Accessed 25 March 2021].

[4] D. Petty, “Hino 500 FD 1126 Air,” Grabcad, 26 December 2020. [Online]. Available: https://grabcad.com/library/hino-500-fd-1126-air-1. [Accessed 19 March 2021].

[5] City of Vancouver, “Use our app to report potholes,” [Online]. Available: https://vancouver.ca/streets-transportation/vanconnect-potholes.aspx. [Accessed 20 March 2021].

[6] Adafruit, “Ultrasonic Distance Sensor - 3V or 5V - HC-SR04 compatible - RCWL-1601,” [Online]. Available: https://www.adafruit.com/product/4007. [Accessed 22 March 2021].

[7] Raspberry Pi Foundation, “Raspberry Pi High Quality Camera,” [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-high-quality-camera/. [Accessed 26 March 2021].

[8] Ublox, “NEO-7M Datasheet,” [Online]. Available: https://www.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf. [Accessed 22 March 2020].

[9] “USB Accelerator,” coral.ai, [Online]. Available: https://coral.ai/products/accelerator/. [Accessed 23 March 2021].

[10] “USB Accelerator Datasheet,” coral.ai, [Online]. Available: https://coral.ai/docs/accelerator/datasheet/. [Accessed 23 March 2021].

[11] “Raspberry Pi 4 Model B Datasheet,” June 2019. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf. [Accessed 22 March 2021].

[12] “CHARMAST 26800MAH ULTRA SLIM PORTABLE CHARGER,” 2019. [Online]. Available: https://www.charmast.com/products/charmast-usb-c-power-bank-26800mah-portable-charger-usb-c-slimline-type-c-battery-pack-with-3-inputs-5v-1-5a-2a-2-6a-4-outputs-5v-2-4a-3a-compatible-with-macbook-galaxy-s8-pixel-iphone-ipad. [Accessed 25 March 2021].

[13] “APIs,” ProgrammableWeb, [Online]. Available: https://www.programmableweb.com/apis. [Accessed 20 March 2020].

[14] "Raspberry Pi Camera Module," raspberrypi.org, [Online]. Available: https://www.raspberrypi.org/documentation/hardware/camera/. [Accessed 23 May 2020].

[15] "Firefly DL," flir.ca, [Online]. Available: https://www.flir.ca/products/firefly-dl/. [Accessed 23 March 2020].

[16] Garmin, "Garmin 66W Dashcam with 2" LCD Screen & Wi-Fi," [Online]. Available: https://buy.garmin.com/en-CA/CA/p/661787. [Accessed 26 March 2021].

[17] Canakit, "Raspberry Pi HQ Camera Kit with Telescopic and Wide Angle Lens," [Online]. Available: https://www.canakit.com/raspberry-pi-hq-camera-kit.html. [Accessed 26 March 2021].

[18] Beitian, "BN220 Datasheet," October 2015. [Online]. Available: https://files.banggood.com/2016/11/BN-220%20GPS+Antenna%20datasheet.pdf. [Accessed 22 March 2020].

[19] A. Allan, "The Big Benchmarking Roundup - Getting started with machine learning and edge computing," hackster.io, 2019. [Online]. Available: https://www.hackster.io/news/the-big-benchmarking-roundup-a561fbfe8719. [Accessed 23 March 2021].

[20] "Anker PowerCore 26800," anker.com, [Online]. Available: https://www.anker.com/products/variant/powercore-26800/A1277011. [Accessed 25 March 2020].

[21] Amazon Web Services, "About AWS," Amazon, [Online]. Available: https://aws.amazon.com/about-aws/. [Accessed 26 March 2021].

[22] F. Lardinois, "AWS launches Amazon Location, a new mapping service for developers," TechCrunch, 16 December 2020. [Online]. Available: https://techcrunch.com/2020/12/16/aws-launches-amazon-location-a-new-mapping-service-for-developers/. [Accessed 25 March 2021].

[23] D. A. Norman, The Design of Everyday Things, Revised & Expanded Edition, New York: Basic Books, 2013.

[24] "ISO 8601 Date and Time Format," ISO, 25 February 2019. [Online]. Available: https://www.iso.org/iso-8601-date-and-time-format.html. [Accessed 20 March 2021].

[25] "ISO/IEC 11581-6:1999 Information technology — User system interfaces and symbols — Icon symbols and functions — Part 6: Action icons," ISO, [Online]. Available: https://www.iso.org/standard/24263.html?browse=tc. [Accessed 20 March 2021].

[26] "ISO/IEC 11581-10:2010 Information technology — User interface icons — Part 10: Framework and general guidance," ISO, [Online]. Available: https://www.iso.org/standard/46445.html?browse=tc. [Accessed 20 March 2021].

[27] "ISO/IEC 29136:2012 Information technology — User interfaces — Accessibility of personal computer hardware," ISO, May 2015. [Online]. Available: https://www.iso.org/standard/45159.html. [Accessed 20 March 2021].

[28] "ISO 9241-110:2020 Ergonomics of human-system interaction — Part 110: Interaction principles," ISO, May 2020. [Online]. Available: https://www.iso.org/standard/75258.html. [Accessed 20 March 2021].

[29] "ISO 9241-161:2016 Ergonomics of human-system interaction — Part 161: Guidance on visual user-interface elements," ISO, February 2016. [Online]. Available: https://www.iso.org/standard/60476.html. [Accessed 20 March 2021].

[30] "ISO 19128:2005 Geographic information — Web map server interface," ISO, December 2005. [Online]. Available: https://www.iso.org/standard/32546.html. [Accessed 20 March 2021].

[31] "IEC 61508-1:2010," IEC Webstore, 30 April 2010. [Online]. Available: https://webstore.iec.ch/publication/5515. [Accessed 20 March 2021].

[32] "ISO 13849-1:2015 Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design," ISO, December 2015. [Online]. Available: https://www.iso.org/standard/69883.html. [Accessed 20 March 2021].

[33] "CSA C22.1:21 Canadian Electrical Code, Part I (25th Edition), Safety Standard for Electrical Installations," CSA Group, 2021. [Online]. Available: https://www.csagroup.org/store/product/CSA%20C22.1%3A21/?gclid=Cj0KCQjwutaCBhDfARIsAJHWnHuZlR1bZkWldTWijurjhEriSDvuoZ5E8n7Xjjo6mIq5btOD7_9-YpoaArrxEALw_wcB. [Accessed 20 March 2021].

[34] "CSA C22.2 No. 62133-2:20 Secondary cells and batteries containing alkaline or other non-acid electrolytes — Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications — Part 2: Lithium syste," SCC, 01 January 2020. [Online]. Available: https://www.scc.ca/en/standardsdb/standards/30175. [Accessed 20 March 2021].

[35] M. P. -. B. Design3d, "Raspberry pi HQ camera with telephoto lens," Grabcad, 9 November 2020. [Online]. Available: https://grabcad.com/library/raspberry-pi-hq-camera-with-telephoto-lens-1. [Accessed 20 March 2021].

[36] "AMP CPC, Circular Power Connectors, Housing, Operating Voltage 600 VDC, Wire-to-Wire, 14 Position, Wire & Cable, Receptacle, Reverse Gender, Socket," TE Connectivity, [Online]. Available: https://www.te.com/global-en/product-206043-1.html. [Accessed 21 March 2021].

[37] "E-Switch RBW2ABLKGILFF1," Digi-Key, [Online]. Available: https://www.digikey.ca/en/products/detail/e-switch/RBW2ABLKGILFF1/2639098. [Accessed 21 March 2021].

Appendix A: Design Options

A.1 Hardware Design Option

A.1.1 Ultrasonic sensors

1. Adafruit 3.3V Ultrasonic Sensor RCWL-1601 [ref]
   Tech Specifications:
   - Input Voltage: 3V-5.5V
   - Measuring Range: 2cm-450cm
2. Sparkfun Ultrasonic Distance Sensor HC-SR04 [ref]
   Tech Specifications:
   - Input Voltage: 5V
   - Measuring Range: 2cm-400cm

A.1.2 RGB Camera

1. Raspberry Pi Camera Module V2 [14]
   Tech Specifications:
   - Resolution: 1080p at 30fps, 720p at 60fps and 640 480p at 60-90fps
   - Horizontal FOV: 62.2°
   - Vertical FOV: 48.8°
   - Focal length: 3.04mm
2. Firefly DL Camera [15]
   Tech Specifications:
   - Resolution: 1440x1080 at 60fps
   - Can deploy trained neural network directly on camera's Intel Movidus 2 VPU and can make decisions at real time on camera.
   - Supports TensorFlow and Caffe frameworks
3. Garmin 66W Dashcam with 2'' LCD Screen and Wi-Fi [16]
   Tech Specifications:
   - Resolution: 1440p
   - Frame Rate: 60fps
   - Wi-Fi
   - USB connectivity
   - Powered Via 12V battery (30min lifetime)
4. Raspberry Pi HQ Camera with 6mm Wide Angle Lens and 16mm Telephoto Lens [17]
   Tech Specifications:
   - Resolution: 1080p at 30fps, 720p at 60fps and 640 480p at 60-90fps
   - 6mm Lens – FOV: 63°, Focal length: 6mm
   - 12 mm Lens – FOV: adjustable, Focal length: 12mm

A.1.3 GPS Sensor

1. NEO-7M/NEO-6M [8]
   Tech Specifications:
   - Antenna mode: Active
   - Update rate: maximum 5Hz
   - Data format: NMEA

- Accuracy: 2.5m autonomous, 2.0m with SBAS
- Compatibility with Raspberry Pi: yes

2. BN220 [18]
   Tech Specifications:
   - Antenna mode: Passive
   - Update rate: maximum 10Hz
   - Data format: NMEA
   - Accuracy: 2.0m with SBAS
   - Compatibility with Raspberry Pi: Software installation required

A.1.4 Neural Network Processing Unit

1. Raspberry Pi 4B
   - Inferencing Time (using MobileNetV2): 122.6ms
   - Idle Current Consumption: 620mA
   - Peak Current Consumption: 1430mA
   - *Current Consumption is a rough estimate based on Pi3 numbers [19]
2. Raspberry Pi 4B + Movidius Neural Compute Stick
   - Inferencing Time (using MobileNetV2): 80.4ms (USB3) or 116.7ms (USB2)
   - Idle Current Consumption: 500mA
   - Peak Current Consumption: 860mA
   - *Current Consumption is a rough estimate based on Pi3 numbers [19]
3. Raspberry Pi 4B + Intel Neural Compute Stick 2
   - Inferencing Time (using MobileNetV2): 176.4ms (USB3) or 202.9ms (USB2)
   - Idle Current Consumption: 480mA
   - Peak Current Consumption: 910mA
   - *Current Consumption is a rough estimate based on Pi3 numbers [19]
4. Raspberry Pi 4B + USB Coral Accelerator
   - Inferencing Time (using MobileNetV2): 18.2ms (USB3) or 102.3ms (USB2)
   - Idle Current Consumption: 470mA
   - Peak Current Consumption: 880mA
   - *Current Consumption is a rough estimate based on Pi3 numbers [19]

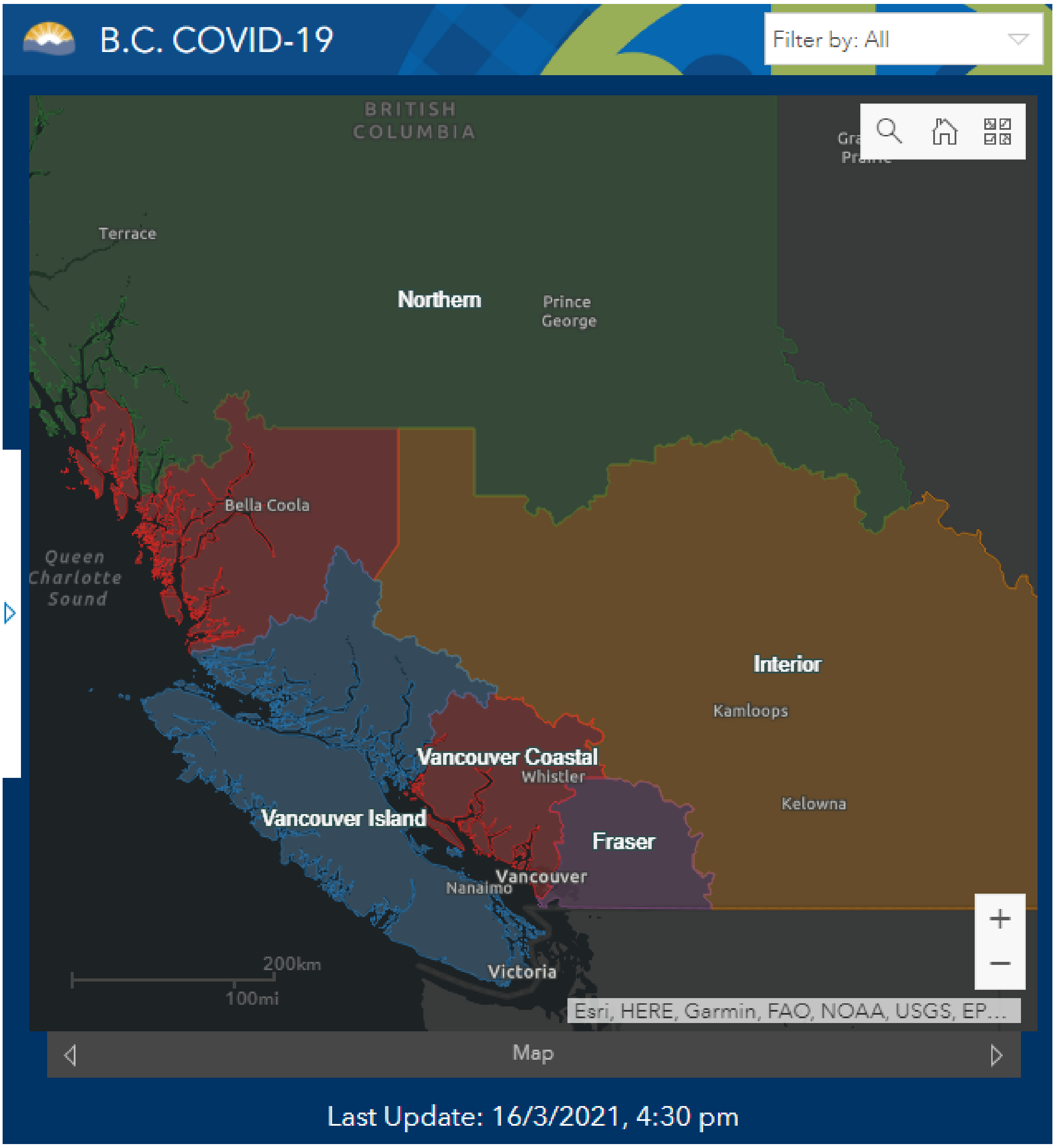
A.1.5 Power Supply

2. Charmast 26800 mAh Ultra Slim Portable Charger [12]
   - Battery type: Li-polymer
   - Compatibility with Raspberry Pi: 5V-3A mode available
   - Size: 7.75 * 3.7 * 0.55 inches
   - Capacity: 26800mAh
3. Anker PowerCore 26800 [20]
   - Battery type: Li-polymer
   - Compatibility with Raspberry Pi: 5V-3A maximum per port
   - Size: 7.1 * 3.2 * 0.9 in
   - Capacity: 26800mAh

A.2 Software Design Options

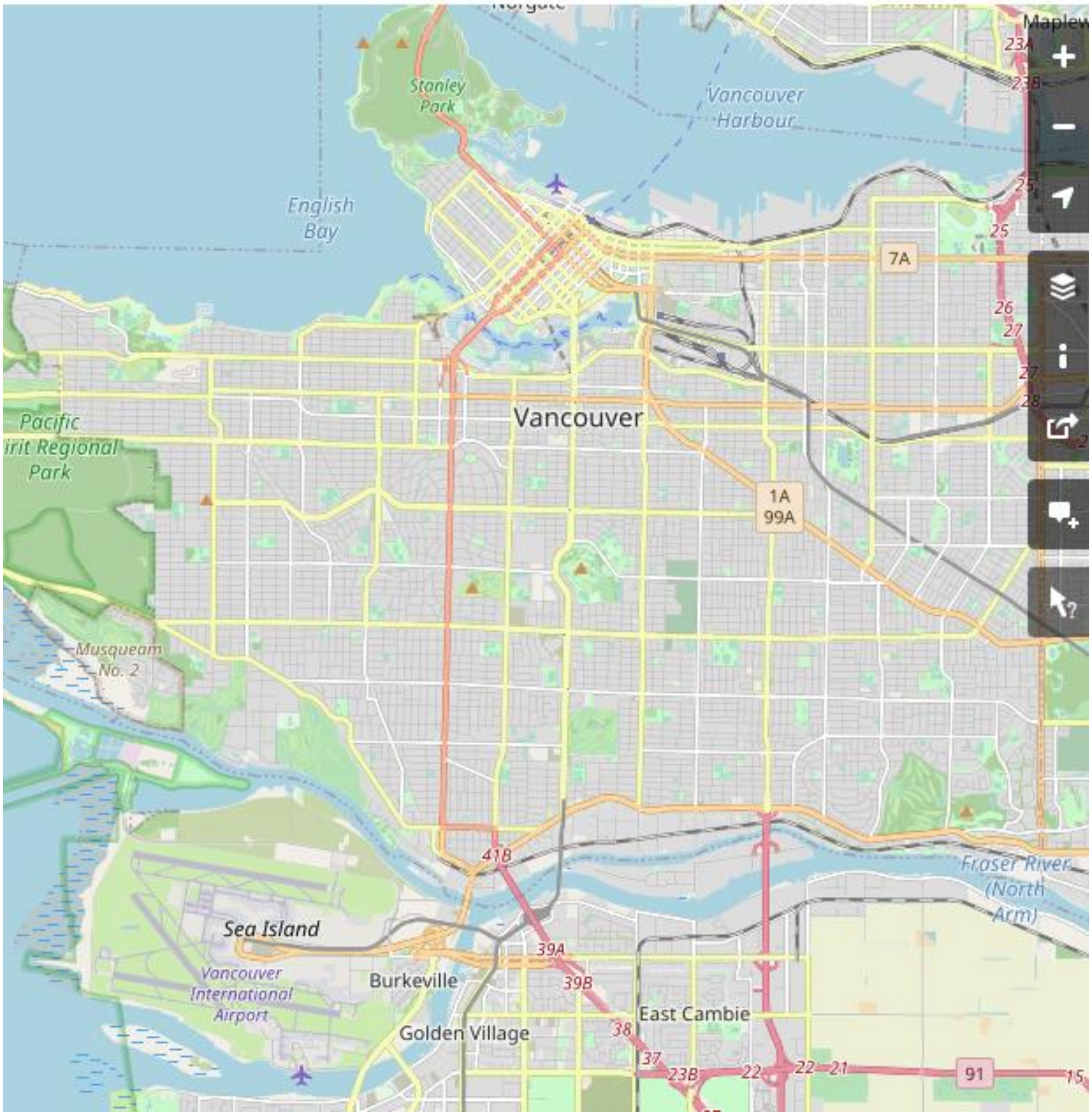
A.2.1 Mapping API Options

ArcGIS is a geographic information system that can be used as a mapping tool. The tool uses desktop platforms for development but can output its data into a web format. Figure A1A-1 shows the British Columbia COVID-19 dashboard, which uses ArcGIS to display disease data.



Appendix Figure A-1 Office B.C. Covid-19 Dashboard Built upon ArcGIS

ArcGIS is enterprise-grade software, which can be adapted to small-scale projects. Government websites frequently employ ArcGIS to display data to the public, however the graphical interface is less functional than using an API such as Google Maps. ArcGIS allows the use of Python to create custom functionality, however the functionality is limited to what the user adds. There are few shortcuts when uploading data in a CSV, TXT, or GPX (GPS data format) file. As such, importing and exporting data can become cumbersome when they are a focus of the program.



Appendix Figure A-2 OpenStreetMap Website Default UI

OpenStreetMap is a crowd-sourced collaborative mapping tool. For government usage, the use of collaborative data is of dubious benefit, since levels of government desire stable data sources. If used as a mapping API, the functionality is very barebones. Many functions, such as returning information about a place, are not available with OpenStreetMap, since it was not built with a place API. Third-party plugins such as Mapbox would allow OpenStreetMap to attain place information (reviews, geographic information, etc.), however the complexity of the project would increase, and subcomponent dependencies would increase as well.

Google Maps API is the most chosen web API for mapping [13]. Its documentation is more extensive than other mapping APIs. The Place API allows place information to be retrieved, which differentiates Google's API from open-source mapping APIs. Information on businesses, parks, and other locations is kept up to date by users and Google's large information databases. The Google Maps API was chosen for its ubiquitous usage on the world wide web.

A.2.2 Database Options

Firebase was chosen as the database application of choice. As a Google product, it features simple integration with the Google Maps API. The pricing is competitive with other database software solutions. Google's uptime on their database as well as the easy-to-scale services make it a solid fit for an expanding database. The system is basic but exceeds the needs of the product. Traffic management features differentiate it from standard database solutions. Relational data is a nice-to-have but mapping systems do not require relational databases.

Amazon Web Services (AWS) is a cloud service that offers database storage, including relational database (RDS) [21]. The service is scalable, and AWS is one of the most popular solutions for cloud computing. The infrastructure is global, and the deployment is secure. The services are extensive and are excessive for the mapping solution at hand. The system only needs to associate GPS data, time data, and other miscellaneous data. For the associated mapping API, AWS's mapping system was released in late December of 2020 [22]. It is a relatively inexpensive solution but is lacking in features and documentation.

A.3 Pothole Object Detection Algorithm Options

There is a multitude of available pretrained networks that can be finetuned with a custom dataset to detect a specific object. However, due to the design choice for the neural processing unit choice outlined in section 3.5, a neural network that is compatible with the Edge TPU (I.e the USB coral accelerator) must be used.

A model with a good accuracy as well as fast enough inferencing time for use in a moving vehicle must be chosen. Therefore, out of all the possible models the following two were the most considered.

1. SSD MobileNetV1 FPN
    - When run with the COCO dataset this model had a mean accuracy precision (mAP) value of 32 and an inferencing speed of 56 ms.
2. MobileNetV2
    - When run with the COCO dataset this model had a mAP of 24.3 and a speed of 6.9ms with the edge TPU

Appendix B: User Interface and Appearance Details:

B.1 Introduction

The motivation behind creating Plot-Hole was to simplify and improve the pothole detection and reporting process. When designing a product to address a problem, one must always keep usability and accessibility in mind. While that innovative solution might fulfil its intended purpose, it can also create other issues that could potentially render the product useless. Cylindrotech's Plot-Hole was designed with automation in mind, minimizing user input. Once powered on, it will not require human operation until the user decides that sufficient data has been collected. From detection to submission, the user interfaces were designed to be easy and intuitive.

B.1.1 Purpose

The purpose of the appendix is to review the human interaction elements of Plot-Hole and provide a straightforward experience to the user. Aspects of the product including sections for the hardware and software subsystems will be analysed and discussed.

B.1.2 Scope

The appendix will cover user analysis, technical analysis, engineering standards, analytical and empirical usability testing, and graphical representations.

B.2 User Analysis

The following section outlines the presupposed knowledge a user should have before being able to utilize the product. Plot-Hole consists of two main components: the vehicle system and the webpage.
The target audience for Plot-hole is city workers who are involved in road repair or management. Since the product is designed for use in street sweepers, those operating the street sweepers should be able to use this product.

B.2.1 Vehicle System Analysis

The main user of the vehicle system will be the driver of the street sweeper. The vehicle system has an enclosure which will be mounted on the roof of the street sweeper. A separate ultrasonic sensor array will be mounted to the bottom front bumper. The user will only need to know how to press a button to turn on the product whenever it is required for use. The communication between the ultrasound and all other subcomponents within the system will be done by the system; the user needs no knowledge of the inner workings to successfully operate the product. Data collection is done by writing the size, location, and timestamp about a detected pothole to a text file which will be stored on an external hard drive. Once data collection is completed the user can then upload the results to the web application.

B.2.2 Website Analysis

Cylindrotech's webpage was designed so that anyone with prior knowledge of using a standard website interface can interact with it. For each task, descriptive text accompanies interactable components of the webpage to communicate with the user. All aspects of the user interface are displayed upfront to the user so that there is no ambiguity.

B.3 Technical Analysis

For a product to have a good design, it must be designed from a user's focal point. Engineering must satisfy the company's needs but also the user's needs. In this product, from Don Norman's The Design of Everyday Things, the principles of effective design are followed [23]. The section details how the project adheres to the seven fundamental principles of design (discoverability, feedback, conceptual models, affordances, signifiers, mappings, and constraints). The product's user interface and appearance comprise of two parts: hardware and software.

B.3.1 Discoverability

Discoverability emphasises the need for user interfaces to be intuitive. Labels should be kept to a minimum, as users create mental maps of the environments that they face. Switches, menus, and buttons should be simple. If the interfaces require labelling, then every interface should be designed the same way, so users are only required to learn once. In terms of learning, standards should be followed to allow the user's previous knowledge to be reused.
Hardware:

- The on/off switch will be implemented with only two states: on and off.

- The hardware on/off switch will use universal colours and language (I for on, O for off, lit for on, unlit for off). The universal standards must be adopted for on/off discoverability.
- If a light-switch-styled on/off switch is used, then the depressed side will point towards the on/off state. This is synonymous with light switches in houses and in electrical devices, such as lamps.

Software:

- All software buttons will behave in the same way.
- Buttons in the software are labelled clearly with their functions.
- A help button is present in the software to aid with discoverability.

B.3.2 Feedback

Once a user has interacted with the UI, feedback should be noticeable. Auditory and visual feedback are the major ways in which users understand that the product is responding to stimulus. The feedback must stay consistent – if one button plays a sound, then all buttons should play sounds. If a user is accustomed to one button playing a sound, the user may glean incorrect information should a clicked button have the absence of sound.

Hardware:

- The on/off switch will make an auditory sound when pressed.
- A sound will not play if the button is pressed, and the system has not changed state.
- A sound will play in any on/off state change. This comes from the physical on/off switch mechanism.
- The system will have a light that is on when the system is turned on.
- External storage will have visual and physical cues when successfully plugged in.

Software:

- Buttons will have an unpressed and a pressed mode, to give the user feedback on whether their click was successful.
- Buttons will have effective debounce mechanisms – if a button is pressed multiple times, then the button will respond that number of times.
- Clicking on a pothole marker shows a visual indication that the user has selected it.
- When a file is uploading, an uploading loading message is shown.

B.3.3 Conceptual Model

Conceptual models have concepts that are like discoverability. Users create mappings of what they perceive in the mind. For example, geographers understand people to create “mental maps” of the areas that they live in. The mental maps are not accurate drawings but rather represent how the user sees the world around them. For hardware and software, users should gain an intuitive understanding of how the system should respond to stimulus. The system should feel natural for users to learn and use. Although some cultures may have widely different mappings, the way in which the system interacts should accommodate all cultures (e.g., reading right-to-left vs. left-to-right, differing colour connotations).

Hardware:

- A switch in the “off position” and the absence of lights indicate that the system is off.

Software:

- Holding on a button will not cause any actions until the button has been released. A common behaviour across both physical and virtual buttons is used.
- Time formats will be written in ISO 8601 format [24]. The format is YYYY-MM-DD, and then time. Different cultures have different ways of writing year, month, date, and time. To alleviate this, a universal time format is used.
- The Google Maps implementation functions similarly to already existing popular mapping solutions. Clicking on an icon displays more information, the map can be scrolled, and the map can be zoomed.

B.3.4 Affordances and Signifiers

Affordances and signifiers indicate to users how to use a product. Affordances are very similar to signifiers in this project and differentiating the two would be superfluous. For example, effective doors have push and pull handles. Signifiers ensure discoverability and show proper feedback to users. Visible stimuli guide users. Conversely, anti-affordances guide users on what not to do.

Hardware:

- The user will be discouraged from disconnecting internal wires by an anti-affordance. The wiring will be embedded deeply into the enclosure such that tools would be needed to change internal connections.
- Physical switches offer a clear affordance and signifier to the user. The on/off switch will not be a touch screen but rather a physical button.
- Plugging in external storage has a clear location on where to plug in the drive. The drive will also be easily removable to give the user a clear affordance on where to disconnect.

Software:

- Affordances refer to possible actions, so the Google Maps API will be used to show buttons for zooming, showing different terrain views, and panning. Although the buttons are not all necessary, they are a visual signifier to show the user of what the map is capable of, without a manual.
- Only relevant buttons are visible to the user to avoid confusion.
- Buttons are shown with a consistent colour and font to differentiate from other text.
- The default map zoom shows all pothole markers. It is not necessary for the user to zoom out to see every pothole that is mapped. The affordance is that the user is afforded the complete visibility of the mapping application as soon as possible.
- Documentation will be shown to the user if they request it via a question mark button.

B.3.5 Mappings

Don Norman describes natural mapping as perceivable cues [1]. The knowledge of mappings is taken for granted and is ubiquitous. Actions should mimic the desired outcome.

Hardware:

- Plugging in an external drive is an essential part of the project. The natural mapping is the cable and connection that represents the flow of data. I/O has become a standard, therefore the natural mapping is obvious. A cable will not plug in if it is the wrong connector.

Software:

- Uploading files is on the same page as the map. The user should glean that the uploaded information is displayed on the map, without having to navigate to another page. The ease of access is a natural mapping of input to output (inputting data and the map outputs the result).
- Users that manually type in pothole data should see that their text appears on the screen. Since typing requires visual feedback, the site adheres to the natural mapping of a keyboard to a text field.

B.3.6 Constraints

Constraints help users understand the way to proceed in both hardware and software. Norman's book, The Design of Everyday Things, mentions four classes of constraints: physical, cultural, semantic, and logical [23].
Physical constraints are defined as limitations in the physical world. Possible actions are tangible and visible to the naked eye. Cultural constraints refer to guidelines and manners. Cultural norms vary across the world and violation often results in negative feedback from other people. Hardware and software must account for cultural constraints in the markets that they deal with. Software cultural constraints are conventions that users take for granted. Semantic constraints are described as constraints that rely upon situational meanings. Vehicles have standard interfaces for driving and displaying information to other drivers. An example of a semantic constraint would be the brake lights on the rear of the vehicle – red indicates braking while an absence of red indicates the opposite. Semantic constraints quickly evolve as technology evolves. The hardware and software portions must consider how the product's environment may change in coming decades.
Logical constraints follow from other constraints. After the user has taken other constraints into account, logical constraints are what follow. Puzzles typically become easier to solve as more pieces are played. The user interfaces of the product can exploit logical constraints to cause users to enhance their natural mappings.

Hardware:

- Cables are chosen such that they cannot be plugged in with an incorrect orientation. USB-C is chosen as much as possible to ensure that users and maintenance workers can avoid having to contend with cable connections that are not bidirectional. The physical constraint of cable connectors is well-understood by users.

- The semantic constraint of the system is that it is mounted facing the direction of travel. The product is designed to not hinder the visibility of the front-facing driver. As self-driving cars become more common, vehicle standards (and semantic constraints) may change, so the system could be updated to match new conventions.
- The hardware product only has two user-interactive parts: the on/off switch and the swappable external drive. The logical constraint from only having two parts is that once the user realizes that the parts are in, then there is no question that the product is ready for use.

Software:

- The Google Maps API is used as it transcends cultural constraints. Google is a pioneer in the field of internet information systems. As such, cultural conventions were learned on Google's interfaces. Dragging a map around and dragging scroll bars moves the window in the specified direction.
- The interface will remain the same across all web platforms. The convention should not change with platform. Cultural convention is important in software – users expect congruent behaviour.
- The software can be upgraded to match new semantic constraints. Old semantic constraints can become problematic if they cannot be removed. The way in which pothole maps relate to the physical world may change – maps could contain more information, data acquisition methods could evolve, and some information types could become outdated. Updating software can alleviate the aforementioned issues.
- A filter for time is included in the pothole map. This feature uses a logical constraint of time to allow users to make sure that no new potholes have been discovered.

B.4 Engineering Standards

The Plot-hole user interface must comply with the industrial standards and regulations. The following Engineering Standards in Table B.4.1 and Safety Standards in Table B.4.2 will be considered during the design of the UI.

Appendix Table B-1 Engineering Standards

Standard ID	Description of Engineering Standard
ISO/IEC 11581-6:1999, 10:2010	Information Technology – User system interfaces and symbols – Icon symbols and functions - Part 6: Action icons [25] - Part 10: Framework and general guidance [26]
ISO/IEC 29136:2012	Information technology – User interfaces – Accessibility of personal computer hardware [27]
ISO 9241-110:2020, 161:2016	Ergonomics of human - system interaction - Part 110: Interaction principles [28] - Part 161: Guidance on visual user-interface elements [29]
ISO 8601	Universal date and time format [24]
ISO 19128:2005	Geographic information – Web map server interface [30]

Appendix Table B-2 Safety Standards

Standard ID	Description of Engineering Standard
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements [31]
ISO 13849-1:2015	Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design [32]
CSA C22.1	Canadian Electrical Code, Part I, Safety standards for electrical installations [33]
CSA C22.2 No. 62133-2:20	Secondary cells and batteries containing alkaline or other non-acid electrolytes - Safety requirements for portable sealed secondary cells, and for batteries made from them, for use in portable applications – Part 2: Lithium systems [34]

B.5 Analytical Usability Testing

Analytical usability testing focuses on testing the user interface from an engineer's point of view. Performance, overlooked features, and issue documentation are the main facets of analytical testing. General usability of software can be done through testing throughout development. Designing a product requires evaluation, and testing leads to engineers developing a sense of what is needed for further building. In designing the project, the general usability tests below detail a list of tests that Cylindrotech's engineers will engage in. Other tests that may occur on the fly will be documented if they happen.

B.5.1 Hardware

- The on/off switch makes a sound when pressed.
- The on/off switch is tactile, and it is clear whether the system is on or off.
- The external storage is securely connected.
- The mounting mechanism is intuitive and robust.
- The recharging port on the power source is clearly labelled.
- The battery lasts long enough for 8 hours.
- The battery charges to full overnight.
- The system does not generate excessive heat.
- The enclosure will protect its components from water damage.

B.5.2 Software

- The map loads with a suitable default zoom level for viewing all the potholes.
- Potholes are represented with the appropriate marker.
- Many potholes in the same area should have a special marker to indicate a cluster – the marker contains the number of potholes in the area.
- The map can be dragged, zoomed, and customized via specific buttons.
- Error messages are descriptive.
- Uploading a valid GPS data file has the potholes appear on the map in a reasonable time (<1 minute for multiple potholes).
- Clicking on a marker for a pothole shows pertinent information about the pothole.
- Clicking on a marker for a pothole cluster shows a list of potholes in the specified area.
- The buttons are intuitively labelled and performs their expected action
- If an image fails to load, descriptive alternative text is used.
- The help page is easily accessible and is descriptive.

B.6 Empirical Usability Testing

Testing is essential to the success of the product. Designing a product also requires building and evaluation stages. Empirical usability testing evaluates the product's user interface and appearance. Information from a user perspective may fill in the information that engineers overlook. Software can be tested via the world wide web without the need for having users in the physical presence. Questionnaires are one of the main methods of acquiring valuable feedback. Questionnaires require little effort from the engineers to collect large amounts of data from many people. Although collecting user information is useful, anonymity can be provided to users upon request. The questions are left open since open questions are more meaningful than closed questions.

Questionnaire 1:
Hardware Situation: Turning on the machine.
Questions for users:

1. Is the on/off button in an intuitive place?
2. Is the on/off button tactile? Did you realize when the button press occurred?
3. Is the button easy to press? Are there any other features that are nice to have?
4. Ignoring the on/off switch, is it apparent when the system is on?
5. What would you like to add or change about the on/off system?

Questionnaire 2:
Hardware Situation: Replacing the storage device.
Questions for users:

1. Is the storage device attachment location in a good place?
2. After attaching a storage device, does it feel secure?
3. If you were using the product for the first time, would you be aware that a storage device is required for it to run?
4. Do you prefer a data cable or direct attachment (flash drive/external solid-state drive) approach?

Questionnaire 3:
Software Situation: Uploading a pothole manually or uploading a text file to the website.
Questions for users:

1. Is the uploading text file button in a good place?
2. After uploading the text file, would you be aware that the file is uploaded successfully?
3. Did the file upload in a fast time?
4. If the file did not upload instantly, were you aware that the file was uploading?
5. Did the potholes update successfully and quickly after the text file was uploaded?
6. Is it clear what format the text fields require?
7. Is it intuitive when submitting a pothole manually?
8. Is it clear what the difference is between uploading a pothole via a text file versus the manual method?

Questionnaire 4:

Software Situation: Submitting a pothole to the city of Vancouver (both individual potholes and all potholes at once)
Questions for users:

1. Is the submit button easy to find?
2. If you encountered an error message, was it descriptive?
3. If a button was clicked by mistake, were there difficulties in undoing the operation?
4. Keeping in mind that the system is built to report potholes to the government, are there more text fields that you would like?
5. Is the user interface performant and elegant?

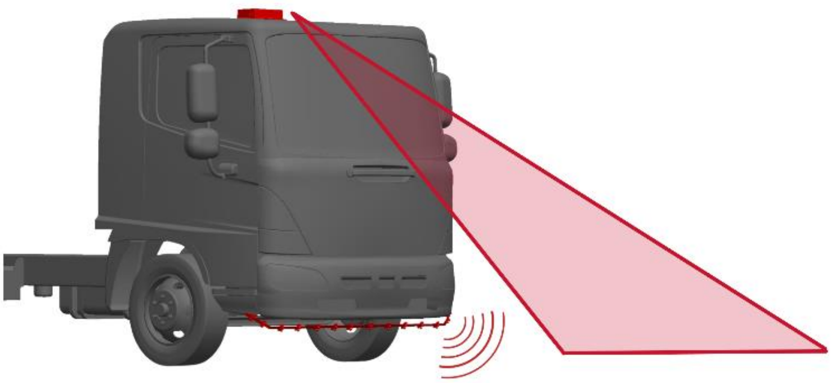
Questionnaire 5:
Software Situation: Using the map. Modifying and removing a pothole.
Questions for users:

1. Are the pothole locations easily identifiable on the map?
2. Is it easy to modify the information on a pothole?
3. Is it a simple process when removing a pothole?
4. Is navigating the map intuitive and optimized?
5. What are your device specifications and operating system? Did the map work fluidly on your device?
6. Did you prefer a 2D or 3D view? Were the default map settings desirable?

B.7 Graphical Presentation

B.7.1 Hardware

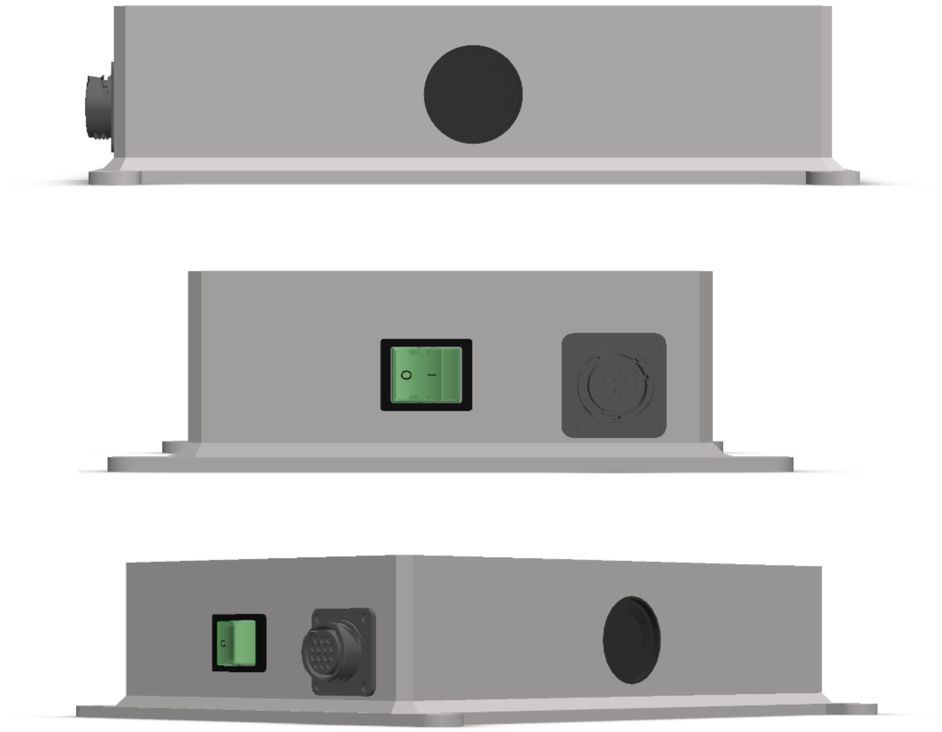
The hardware for Plot-Hole consists of two main components: the electronics and camera enclosure and the ultrasound array. Both systems will be pre-installed, and the user will not have to adjust the mounts for these units.
The interactive elements of the hardware include the power button, an opening for recharging the battery, and removing the storage media for uploading to the web application.
An overview of the deep learning camera and ultrasonic array working together is shown in Figure B.7.1.1:



Appendix Figure B-1 Overview of the Hardware Components

[4]

The enclosure as shown in Figure B.7.1.2 houses all major electronic components such as the microcontroller, battery, neural network processing unit and storage media. There is an intuitive and tactile power button on the outside of the enclosure facing the same side as the passenger's door. The hatch for creating an accessible opening for the battery and storage has not been designed yet and is currently in the design stage.

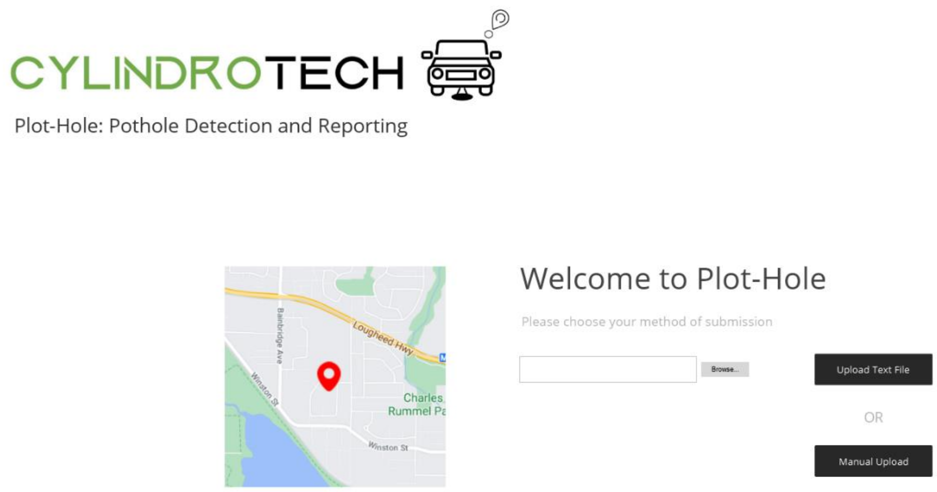


Appendix Figure B-2 Different Views of the Proposed Enclosure

[35]

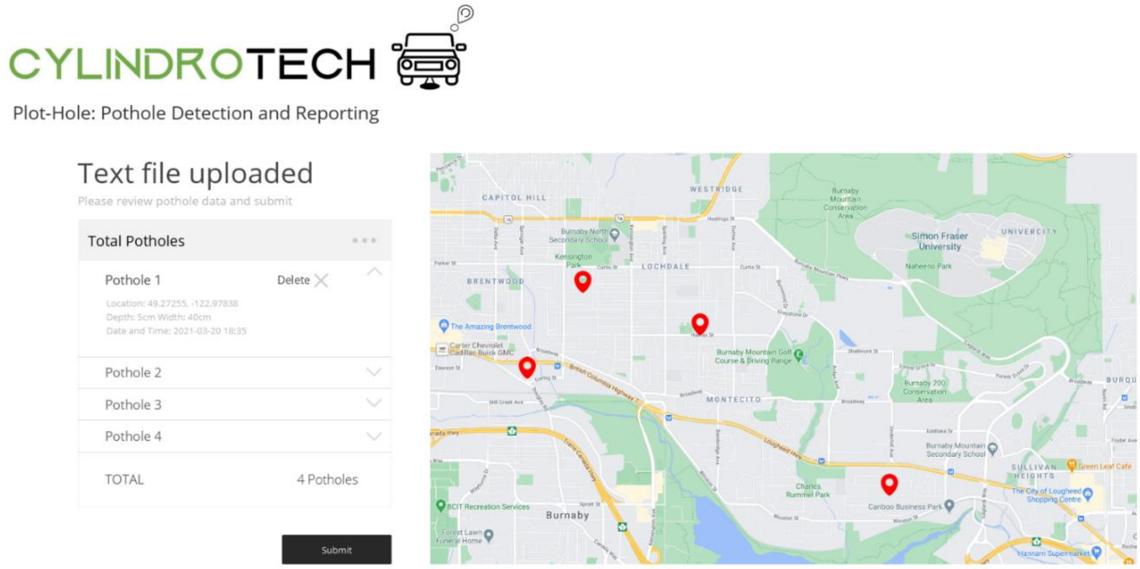
B.7.2 Software

The majority of the user interface will be based on the web application. This requires human interaction for submitting the data collected by the automated pothole detection system. Figure B.7.2.1 shows the home page of the web application. When users open the web page for the first time, they will be asked to submit potholes through a text file or manual entry.

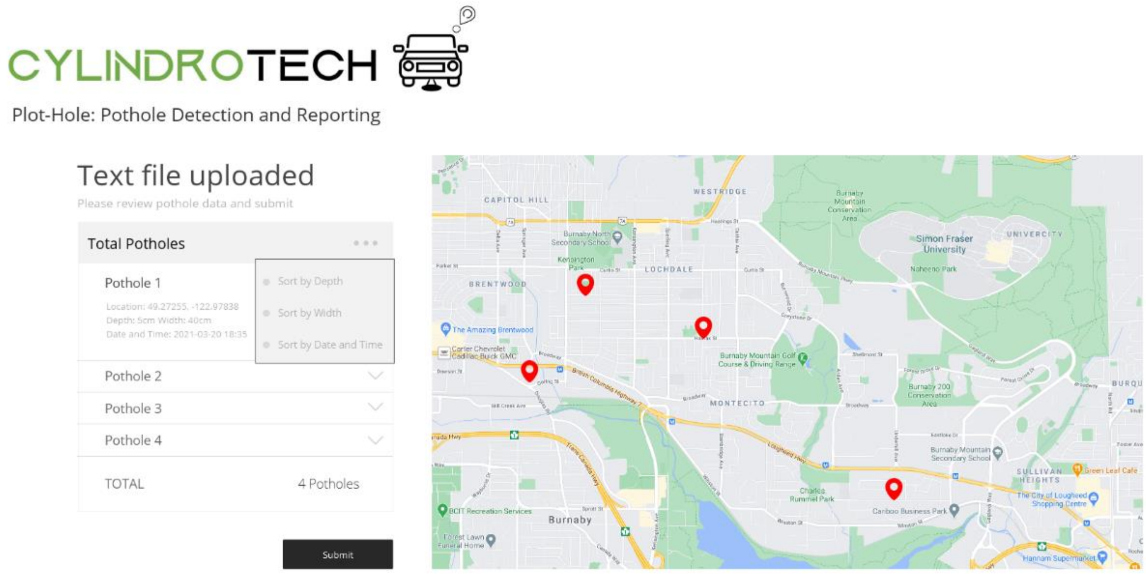


Appendix Figure B-3 Home Page

If the text file method is chosen, the user is already prompted to attach a text file on the home page. After "Upload Text File" is clicked, the web application will automatically import all pothole data and display useful information such as pothole depth, width, and date and time as shown in Figure B.7.2.2. A Google map integration with the location plots of the potholes is also provided for an intuitive interface and visualization. A delete button is available to the user for each pothole. Figure B.7.2.3 highlights the web application's ability to sort potholes based on depth, width, or date and time.

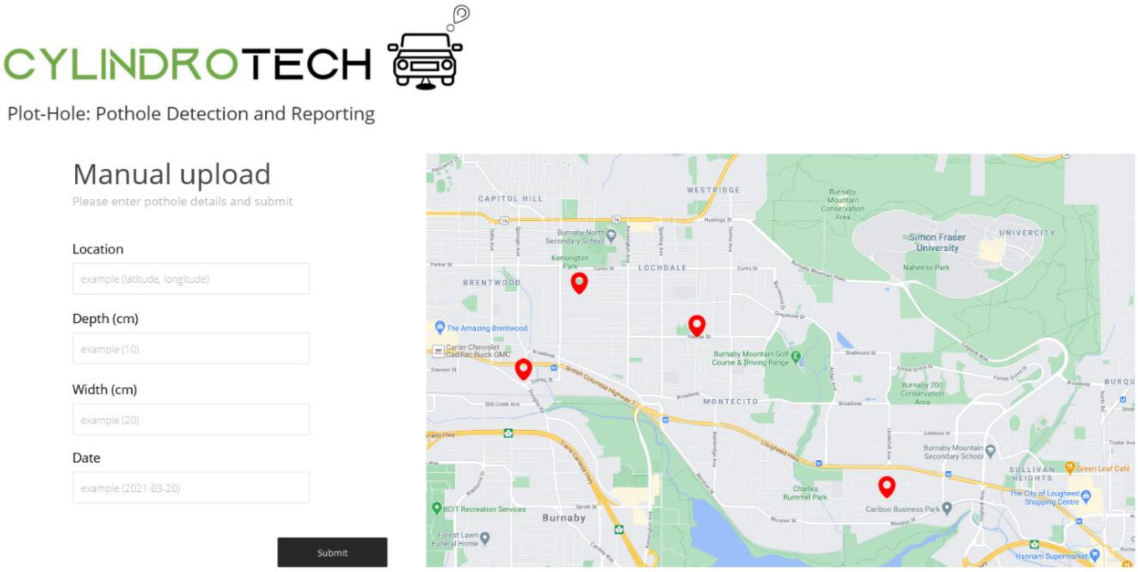


Appendix Figure B-4 Text File Uploaded

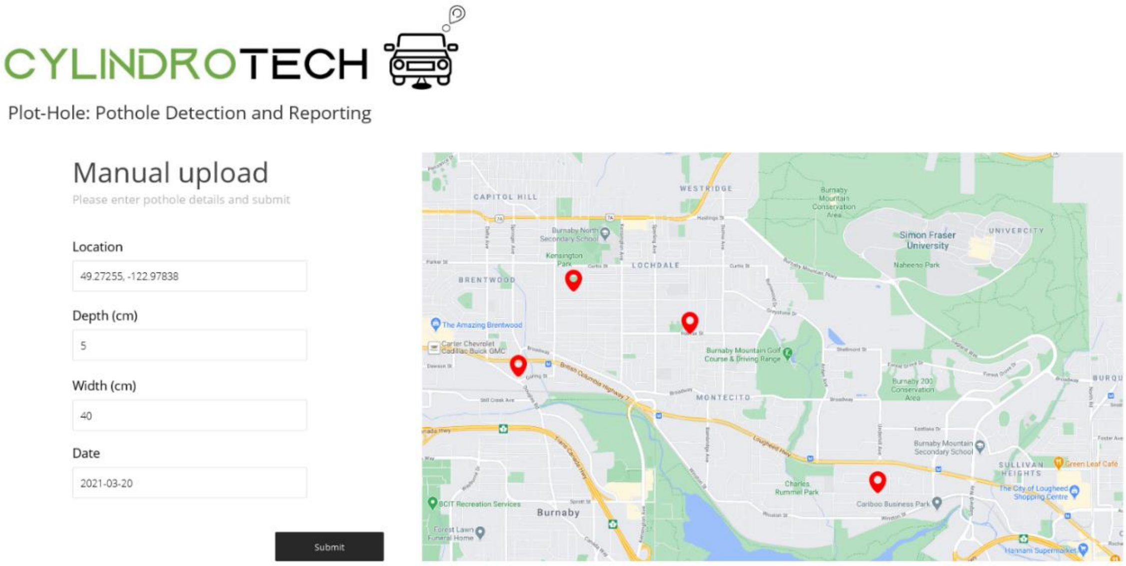


Appendix Figure B-5 Options for Sorting Potholes

If the manual entry method is chosen, the users will be asked to fill the information of the pothole such as location, depth, width, and date. Figure B.7.2.3 and Figure B.7.2.4 show the web application before and after the user has filled out the necessary information. Note that the correct format of these information fields is provided to the user.

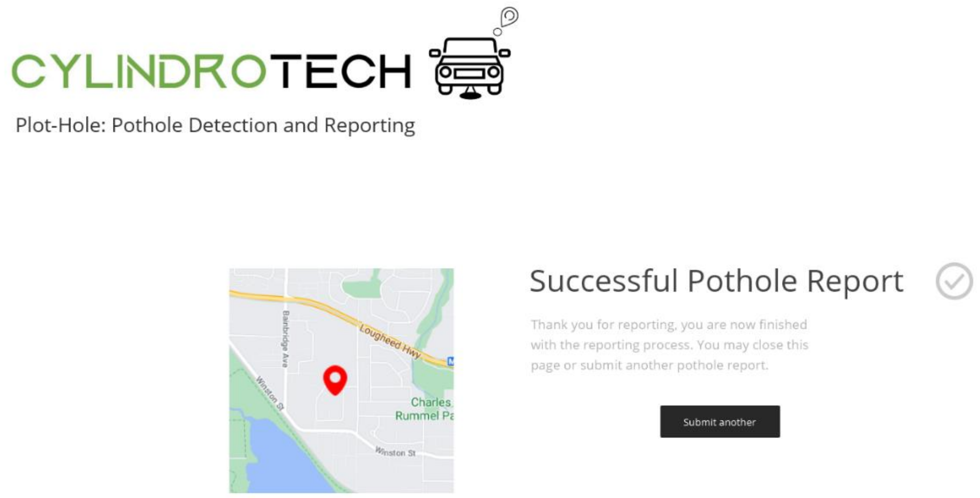


Appendix Figure B-6 Manual Upload Page with Empty Text Fields

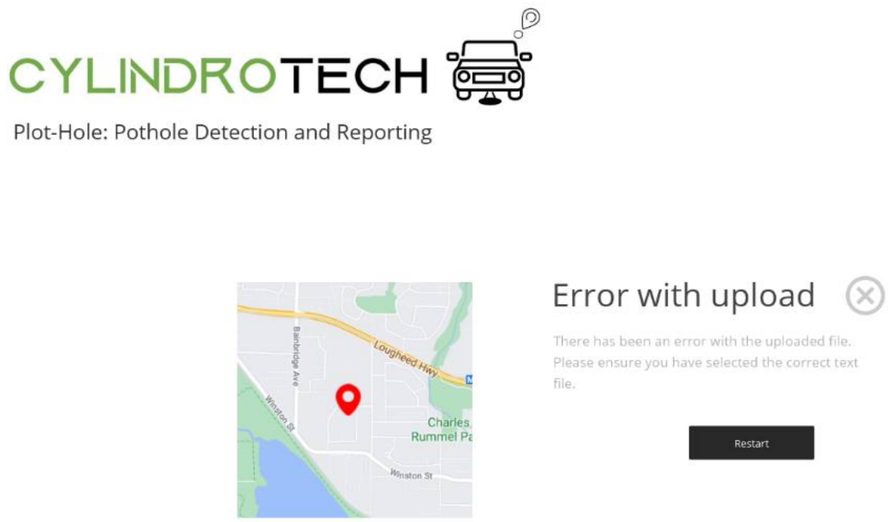


Appendix Figure B-7 Manual Upload Page with Filled in Text Fields

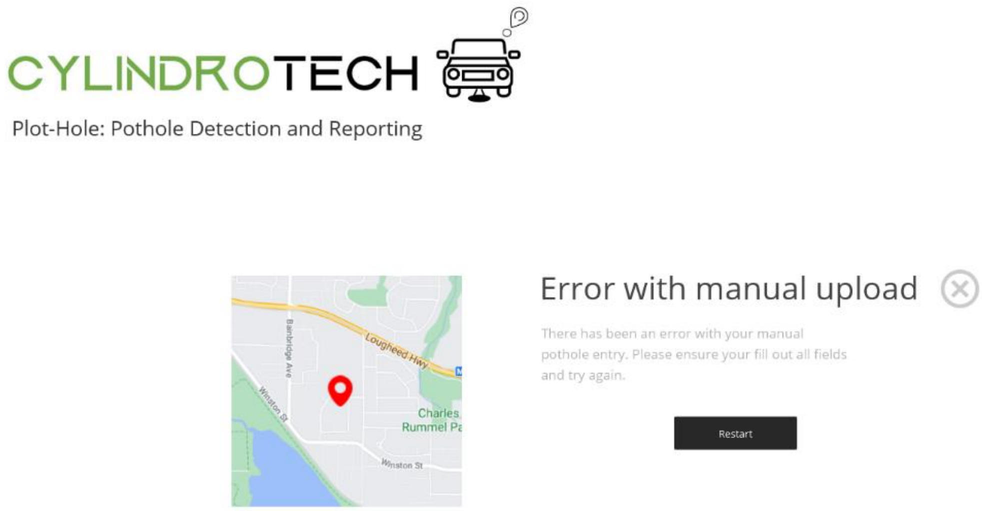
In both the upload text file and manual entry methods, a single submit button is made available to the user to avoid confusion. Upon a successful submission, the user will be greeted with a confirmation page shown in Figure B.7.2.5. If an error occurs, the user will be notified of the issued and be offered to restart the reporting process in Figures B.7.2.6 and B.7.2.7.



Appendix Figure B-8 Successful Confirmation of Pothole Report



Appendix Figure B-9 Error with Upload



Appendix Figure B-10 Error with Manual Upload

B.8 Conclusion

To ensure widespread adoption, equipment should require minimal technical knowledge to operate. This appendix explains the design ideas and usage methods of the product. To ensure that users can better understand and use Cylindrotech's Plot-Hole detection system, the UI design of Plot-Hole follows the "Seven Elements of Interaction" outlined in Don Norman's book "The Design of Everyday Things" [1]. In addition, the Plot-Hole team will undertake usability testing with users, the feedback will help develop the product's usability. For our proof-of-concept and appearance prototypes, the team will keep progressing both the hardware and software designs to deliver a user-friendly system.

Appendix C: Test Plan

The test plan is designed to verify and ensures the Proof-of-Concept prototype of the Plot-Hole meets the design specification and requirements in terms of: Test coverage, Test Methods, and Test Responsibilities. The different stages in the development process are abbreviated as follows:

C – Proof-of-concept/Alpha stage
P - Engineering Prototype/Beta stage
F - Final Product/Production Stage

Test Sheet		
Test Date:	Test Identifier:	
Hardware Test Plan		
1. Ultrasonic System	Pass/Fail:	Comments:
The ultrasonic sensor transmits and receives its own signal.		
The ultrasonic array distinguishes potholes between 30cm, 60cm, and greater than 60cm.		
The ultrasonic array covers a horizontal distance of 227cm.		
The ultrasonic array has a downward angle of 60 degrees.		
The ultrasonic array activated upon detection of a pothole by the deep learning system.		
2. Enclosures		
The ultrasonic system enclosure secures all ultrasonic sensors.		
The ultrasonic system enclosure securely mounts to a street sweeper.		
The camera and electronics enclosure protect the sensitive electronic components.		
The camera and electronics enclosure include a rooftop mounting option for the street sweeper.		
3. RGB Camera		
The camera has a frame rate of at least 30fps.		
The camera has a resolution of 1080p.		
4. GPS Sensor		
The GPS sensor actively returns location and time/date under a 1Hz frequency.		
The GPS sensor is able to communicate with a Raspberry Pi.		

The GPS sensor sends longitude and latitude data to Raspberry Pi through the TX pin.		
5. Neural Network Processing Unit		
The USB accelerator has maximum inference performance time when connected to the Raspberry Pi via USB 3.0.		
The connected USB cable supplies at least 500mA at 5V.		
The USB accelerator can be operated in an ambient temperature of up to 35°C at reduced clock frequency.		
Electrical Test Plan		
1. Power Supply	Pass/Fail:	Comments:
The power bank lasts for at least 8 hours when the system is on.		
Software Test Plan		
1. Web application	Pass/Fail:	Comments:
The web application can launch successfully.		
The web application has an interactable user interface.		
A map should be generated based on the GPS data and pins will be placed where potholes are located.		
The web application can send information of potholes to the city of Vancouver.		
Users are able to plug in an external storage device and upload relevant information to the web application.		
2. Parsing data		
The GPS coordinates data can be converted to specific location on the map.		
The pothole data will be synced on Firebase.		
3. Pothole Detection Algorithm		

The pothole detection algorithm can communicate with the GPS and ultrasonic sensors.		