Software Requirements Specification (SRS)

for

Sidewalk Slope Monitoring System

Version 2.0

Prepared by:
Aquil Alam, Alejandro Chanocua, Omar Eclicerio, Ernesto Garcia, Francisco Gastelum, Henry Gonzales, Gui He, Perla Ramirez, Rishi Shah, Daniel Zeng

Department of Public Works and Bureau of Engineering, LA City

CSULA Senior Design 2020-2021 / LA City Bureau of Engineering

Table of Contents

Table of Contents 2
Revision History 3
1. Introduction 4
1.1. Purpose 4
1.2. Intended Audience and Reading Suggestions 4
1.3. Product Scope 4
1.4. Definitions, Acronyms, and Abbreviations 4
1.5. References 5
2. Overall Description 6
2.1. System Analysis 6
2.2. Product Perspective 6
2.3. Product Functions 6
2.4. User Classes and Characteristics 7
2.4.1 Operator 7
2.4.2 Administrator 7
2.5. Operating Environment 8
2.6. Design and Implementation Constraints 8
2.7. Assumptions and Dependencies 9
2.8. Apportioning of Requirements 10
3. External Interface Requirements 11
3.1. User Interfaces 11
3.1.1 Web Application 11
3.2. Hardware Interfaces 11
3.3. Software Interfaces 12
3.4. Communications Interfaces 12
4. Requirements Specification 13
4.1. Functional Requirements 13
4.1.1 User Control 13
4.1.2 Data Transfer Control 13
4.1.3 Traversal 13
4.2. External Interface Requirements 13
4.3. Logical Database Requirements 14
4.4. Design Constraints 15
5. Other Nonfunctional Requirements 16
5.1. Performance Requirements 16
5.2. Safety Requirements 16
5.3. Security Requirements 16
5.4. Software Quality Attributes 16
5.5. Business Rules 16
6. Legal and Ethical Considerations.… 17
Appendix A: Glossary 18
Appendix B: Analysis Models 19
Appendix C: To Be Determined List 20

Revision History

Name	Date	Reason for Changes	Version
Aquil Alam, Henry Gonzales	5/11/22	Modified Initial Draft for project.	2.0
Perla Ramirez, Ernesto Garcia Alejandro Chanocua,	5/11/22	Modified Task 2 Perspective to draft	2.0
Daniel Zeng, Omar Eclicerio	5/11/22	Modified Task 4 Perspective	2.0
Francisco Gastelum, Gui He, Rishi Shah	5/11/22	Modified Task 1 Perspective	2.0

1. Introduction

The Sidewalk Slope Monitoring System project is an effort to develop the necessary databases, user interfaces, and automation scripts to aid the City of Los Angeles, Bureau of Engineering (BOE) in maintaining over 11,000 miles of sidewalk. Based on their prioritization and scoring system, BOE can assign a numerical score to each sidewalk segment to determine which segments require immediate attention or repair for their Sidewalk Repair Program.

The system developed by our team is designed to help BOE in their data collection by building a robot user interface to control the robot BOE uses during their field analysis, a web application to display and map image data, and a database to hold the data collected and extracted by the system.

1.1 Purpose

The document will define all software and hardware requirements for the rover. It will also cover all the modules and their purposes. The modules and their functionalities will be described in this document.

1.2 Intended Audience and Reading Suggestions

The intended audience of this document is for the developers of this project, City of Los Angeles Bureau of Engineering and members of the sidewalk repair team. Reading this document will give a better understanding of how to operate the rover and its data collection.

1.3 Product Scope

The rover will provide a simple approach for the field user(s) when assessing sidewalks. The rover will collect longitudinal and latitudinal slope data as well as capture images of the sidewalk.

The data will be transferred to an application that will give a visual representation of the findings. This will allow the user to collect a list of sidewalks that needs repairing.

A library of scripts will be developed to create files and layers of data that will be used alongside the Bureau of Engineering's mapping application, NavigateLA. The scripts will obtain data from the database and automatically generate mapping files. These mapping files will be hosted on NavigateLA as a layer of sidewalk data for the Bureau of Engineering to view.

1.4 Definitions, Acronyms, and Abbreviations

For the purposes of this project, the following terms are defined as follows

Rover	Device with wheels that will display the GUI software.
IMU	Inertial Measurement Unit
DB	Database
NavLA	Navigate Los Angeles (Site)
BOE	Bureau of Engineering
GUI	Graphical User Interface
API	Application Programming Interface
SQL	Structured Query Language
CSV	Comma-separated values

1.5 References

Python Documentation - https://docs.python.org/3/

Leo Rover Documentation - https://www.leorover.tech/the-rover

This document was used to comprehend and understand the Leo rover software and hardware.

Javascript Documentation - https://developer.mozilla.org/en-US/docs/Web/JavaScript

Azure Documentation https://docs.microsoft.com/en-us/azure/azure-sql/

Microsoft documentation used as reference when learning Azure SQL database.

ArcGIS API Documentation: https://developers.arcgis.com/python/guide/install-and-set-up/

ArcGIS API documentation used to explore how other users utilize ArcGIS to create projections of our GPS data, learn about possible use cases, and apply best practices to our automation scripts.

React Documentation: https://reactjs.org/docs/getting-started.html

Node.js Documentation: https://nodejs.org/docs/latest-v17.x/api/

ExpressJS Documentation: https://expressjs.com/en/5x/api.html/

MS-SQL Documentation: https://github.com/tediousjs/node-mssql

React and Node.js documentation were used to set-up a testing environment. The documentation was also used to understand how to get started developing the frontend and backend, as React uses Node.js and javascript for the backend.

2. Overall Description

The rover will provide a faster approach for the user(s) when they go out to assess sidewalks. As the user controls the rover down a block, it will collect data about the longitudinal and latitudinal slope as well as capture images of the sidewalk.

This data will later be saved in the Azure SQL database where it can be accessed on our website which will have a visual representation of our findings. Collecting the data will allow the user to run analytics and determine the severity of each sidewalk--providing a list of which sidewalks have priority when it comes to fixing them.

2.1 System Analysis

The system is designed to solve the manual collection of data by providing field workers with a semi-automated rover to assist in collecting sidewalk data. The goal is to provide field workers with simple yet effective designs for the user interface that will assist in collecting and exporting sidewalk data.

Furthermore, the software will provide functionality to process the images of sidewalks rendered by the rover camera. This functionality will allow users to distinguish the horizontal and vertical (X and Y) displacement of the sidewalk.

2.2 Product Perspective

The Leo Rover will serve to replace manual methods of collecting sidewalk assessment information. It will reduce the level of training the user will need in order to gather data from the sidewalk. The rover's ability to take images and associate them to a specific location will help the user(s) to correlate a specific location to a visual representation.

2.3 Product Functions

Raspberry PI 3 and Rover

- Initialize GPS
- Display rover controls
- Output data collected results to CSV file
- Collect photographs using a GoPro Fusion
- Rover will traverse sidewalk

Graph/Front-end

- Read data from CSV file and produce graph
- Display pictures taken and associate them to specific readings on graph

Database/Backend

- Store unprocessed and processed data from rover or GoPro
- Associates GoPro images with sidewalk and rover data
- Associates rover data with sidewalk data
- Provide a central point of data retrieval for all relevant rover, GoPro, and sidewalk data
- Processes CSV files from GoPro or rover and inserts them into the database.
- Converts different spatial references
- Organizes data by unique GPS point for sidewalk severity categorization

Web application and Mapping files

- A web application will display the collected images, one at a time, with its related EXIF data and slope data.
- In addition, Task 1 will develop scripts to create mapping files that can be hosted on BOE's mapping application, NavigateLA.

Rover UI

- A web application that displays the controls for the robot.
- Initiates the rover to start collecting data of the sidewalk segment.

Field Testing

- Testing the rover on the sidewalk to make sure it is reading data
- Testing the digital level to see if the slope data is being read accurately

2.4 User Classes and Characteristics

2.4.1 Operator: refers to any user who will physically operate the rover and will collect the data.

2.4.2 Administrator: refers to any user who will have direct access to the system to modify or change functions of the system. As well as access, input, remove, and my changes to the database.

2.5 Operating Environment

The Raspberry Pi 3 operates in the open-source ecosystem that runs on Linux, and its main supported operating system, Raspbian is open source and runs a suite of open-source software. The application that will be responsible for visualizing the data will most likely be on a web application. For testing purposes, we will be running this application locally on our computers (MacBook laptop/ iPad and Windows laptop /tablet). The azure database can be reached using Windows, Mac, and Linux.

2.6 Design and Implementation Constraints

Possible constraints:

- Accuracy of rover controls
- Accurately calculating the cross slope of the sidewalk
- Analyzing various types of surfaces
- Supplying enough power for all components
- Rover should be operated by someone with minimal training.
- Measuring the vertical displacement
- Accuracy of correlating data from Rover and NavLA (Mainly GPS Coordinates)
- Limited to 1 terabyte of accessible database storage

  - Web application

    - Web application can use a local React server for development and testing purposes.
    - Web application must have access to the database containing sidewalk data.
    - Web application must use a database when displaying sidewalk data and cannot store or use data locally.
    - Web application must have access to the Azure storage containing GoPro images/EXIF data.
    - Web application must use a BOE web server for production.
    - Web application must be accessible to BOE, either using Internet or Intranet access.
    - Web application must be accessible to BOE regardless of their preferred browser.
    - BOE must provision a web server for web application production use.
    - BOE must maintain the web server hosting the web application.
    - Web application backend will be done using React/Node.js.

        - Web application frontend will be done using HTML/CSS and Bootstrap. Will also use React to combine the frontend and backend.
    - Mapping files
        - Mapping files must be generated by scripts using a 3rd party application like ArcGIS.
        - Mapping files must be importable in NavigateLA, if and only if, mapping files will not be hosted on a database managed by BOE.
        - Mapping files must maintain all design features, such as color schemes, annotations, polygons, lines, shapes, when hosted in NavigateLA.
        - Mapping files must be hosted on NavigateLA when a substantial amount of mapping files is available for viewing.
        - A request for a database to host the mapping files can only be placed after discussing and receiving approval from the advisor and BOE.
    - Database/API server
        - Web host is required.
        - A table containing section information is required to associate recordings with a larger entity.
        - NodeJS must be installed on server instances to run the API.
        - IPs from ISP are manually approved by BOE to access the database.
    - Rover UI
        - Phone/Tablet/PC to be used to connect with the Rover.
        - Phone/Tablet/PC must have wifi connectivity.
    - Field Testing
        - Inaccurate data being read
        - Hardware issues that could malfunction

2.7 Assumptions and Dependencies

- Assumptions:
    - Cracks and holes on the sidewalk shall be minimal.
    - Battery 4 hrs of nominal driving or 8hrs of video streaming.
    - Hardware is reliable.
    - Users will protect hardware from damage.
    - System is waterproof.
    - Operator will clear the sidewalk before measurement.
    - System will be used during the day to take optimal photos.

- Sidewalk will always be captured as the center of the image
- Rendered images require user-input before processing measurements
- User will manually input location data fields after every use.
- User will use GoPro Provided app to render 360 Degree Images
- User will manually input data into Azure DB
- Users will encode JSON object based on URI standards
- Users will only upload CSV to the CSV endpoint
- Data will have outliers for the algorithm to clean

- Web application and Mapping files
  - Web application
    - Web application will be used to view specific or individual sidewalk segment images and slope/GPS data.
    - Web application will be used to package data for NavigateLA visualization.
  - Mapping files
    - Mapping files will be viewed by importing them to NavigateLA, if and only if, mapping files are not hosted on NavigateLA.
    - Mapping files will contain data for many sidewalk segments.
    - Mapping files will be viewed by selecting a layer hosted on NavigateLA, if and only if, mapping files are kept on a database managed by BOE.
    - Mapping files will contain its damage categorization based on the Severity Index from the Damage Severity Matrix, provided by the BOE report.

2.8 Apportioning of Requirements

Regarding the data storage side of things, the azure blob storage will be implemented in the future, once the BOE allocates the necessary space for it. The automation of data entries and image rendering will be delayed until we have the capability to pull and read from the azure blob storage.

3. External Interface Requirements

3.1 User Interfaces

The system will use a web application that receives the slope data and photos recorded by the rover. The web application will have a graphical interface to visualize the system's Time Stamps (in seconds) and slope data will be the labels for the X and Y axis. The web application will also have a page where the rover's collected data can be packaged and viewed on the NavigateLA webmap.

3.1.1 Web application

The web interface will receive image data from Leo Rover's camera to be transferred to the website. BOE uses an Azure database for their backend, which we will integrate into our web application.

Our web application will include a page to display our data. The data to be displayed includes from the rover, longitude and longitude slope data, Global Positioning System (GPS) data as well as the image name and date when what image was taken. The images that will be displayed come from the GoPro fusion camera on the rover, and we will also be displaying the GoPro metadata which includes, longitude and latitude data. The web application will have the ability to iterate through the image with a previous button and next button. The user will be able to pan the image and zoom in and out of the image. The web application will also have a page where the Rover's collected data can be packaged so that it can be imported and viewed on the NavigateLA webmap. The data is received and displayed using the date the rover field tests were conducted. Each longitude/latitude record is assigned its appropriate Slope x&y, Date, Time, SRID or Spatial Reference ID, Speed of rover, & the Severity Index. To maintain an uncluttered page we utilized a dropdown bar from which the data can be toggled on/off the page.

The web application will have additional pages which will describe the purpose of our project, the overall description, the environment where our project was worked on, and the algorithms used in this project.

3.2 Hardware Interfaces

The web interface will receive image data from Leo Rover's camera to be transferred to the website. The Leo recovery is equipped with a fisheye lens with a 170-degree view. The images the rover captures will transfer to our database side.

Power input to power the sensor board and Raspberry Pi 70 / 3000mA current consumption, depends on external modules standard 5.5/2.1 mm DC plug (center-positive)

I/O ports 3.3V/5V tolerant GPIOs series resistance is 330Ω, 2.4 GHz Wi-Fi modem, 12V battery

3.3 Software Interfaces

Our customer, the city of Los Angeles, uses an Azure database for their backend, which we will integrate into our web application. Our web application will include a page to display our data. The data to be displayed includes images received from the rover, longitude and longitude slope data, Global Positioning System (GPS) data as well as the image name and date when what image was taken. The web application will have the ability to iterate through the image with a previous button and next button. The user will be able to pan the image and zoom in and out of the image. The web application will have additional pages which will describe the purpose of our project, the overall description, the environment where our project was worked on and the algorithms used in this project.

The software products we will be using include:

- React
- HTML/CSS
- Azure Storage Blob
- Microsoft SQL Server
- ArcGIS
- Rover UI
- Python
- Javascript
- JSX
- GoPro Fusion Studio

3.4 Communications Interfaces

One of our user interfaces is a web application which will use HTTP to communicate between our web application pages.

4. Requirements Specification

4.1 Functional Requirements

4.1.1. User Control (UC):

- The system shall initialize GPS
- The system shall capture a photo using the attached camera GoPro module.
- The system shall display the data captured in the CSV file.
- The system shall control desired speed.
- The system shall move forward, backward, left, and right.

4.1.2. Data Transfer Control (DTC)

- The system shall store the data in a MicroSD card.

4.1.3. Traversal

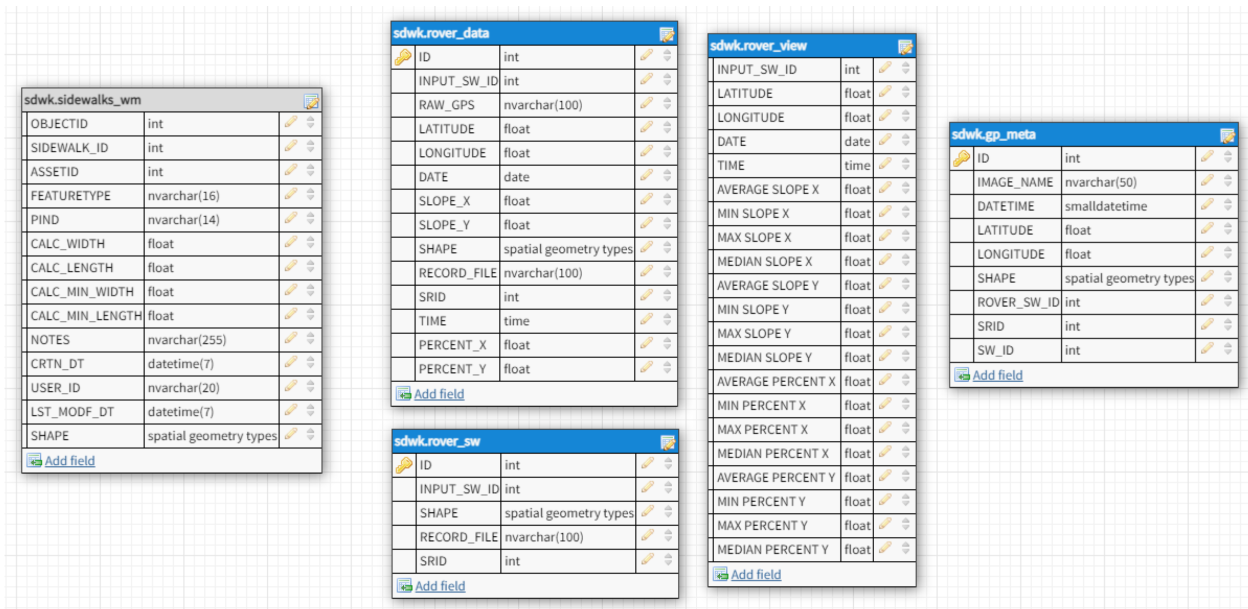
- The rover max linear speed ca. 0.4 m/s
- Estimated maximum obstacle size: 70 mm

4.2 External Interface Requirements

Raspberry Pi 3:
The Raspberry Pi 3 will operate as the main module of the system. The Raspberry Pi will receive the data from the digital level, and Camera to perform the calculation and/or store this data into the Raspberry Pi's MicroSD card. The system's data will be transferred manually from the MicroSD card to a database.

The digital level, accelerometer data (imu/accel topic) represents linear acceleration along the sensor's axes. This module will capture the raw data of the sidewalk's measurements. The data obtained is the values of the three-axis acceleration sensor(m/s^2) and three-axis gyroscope(°/s). Using this data, the Raspberry Pi 3 performs the slope calculations, and the data will be saved into the microSD 00+card as a CSV file.

4.3 Logical Database Requirements



Data which is currently being stored in the Azure DB is coming from three sources. NavigateLA offers map data which provides asset and pin IDs which help us ID property lines. The majority of the data comes from the rover which provides the data it analyzes from the sidewalks it scans. Lastly, the GoPro not only provides images but a complete table of EXIF Data which we extract using a python code.

The rover data is our main source of information. It mainly consists of numeric data which the rover creates by calculating the sidewalk's slope. This data includes GPS coordinates, slope percentages, accelerometer reads, date, and a sidewalk ID used to associate the data to a certain sidewalk. This will be the information that will be constantly displayed and used to calculate whether a sidewalk is in need of repair. This data will also be used to piece together the rover sidewalk shape and GoPro metadata tables. The rover view includes processed data grouped by unique GPS point from the rover data table. This data will be used to assist in sidewalk severity categorization. The data in these tables can be accessed by using the provided endpoints. At the moment, all data will be retained in one table unless it reaches its row entry limit, after which subsequent duplicate tables will be created to store additional data. All data added to the Azure DB is currently being inputted manually after each rover session.

4.4 **Design Constraints**

- The Rover processing equipment is dependent on battery use.
- Maximum linear speed: ca. 0.4m/s
- Maximum angular speed: ca 60deg/s
- Wheel diameter: 130 mm
- Tire material: rubber with foam insert (non - pneumatic)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The camera shall be able to take photos on at least a one second interval
- The digital level shall be able to collect data on at least a one second interval
- The estimated maximum obstacle size is 70 mm while running circa 4 hrs of nominal driving.
- Connection range: Up to 100m (with live video stream)
- Azure database is limited to three types of users: Admin (Full Access), Regular user (Read/Write), guest (Read Only)
- The system shall be charged on a standard outlet
- The system shall run on a rechargeable battery
- The system shall be calibrated before use.

5.2 Safety Requirements

There is no concern for possible loss, damage, or harm that could result from using the system.

5.3 Security Requirements

The system shall only be developed and accessed by CSULA and BoE teams. The system shall only take pictures of the sidewalks.

5.4 Software Quality Attributes

Python scripts in the system shall follow the PEP 8 standard.

5.5 Business Rules

Only staff assigned by the City of Los Angeles Bureau of Engineering shall operate and access the robot, web interface, and database system.

6. Legal and Ethical Considerations

Our project is focused on redesigning the robot, web interface, and database system that will be used by the City of Los Angeles Bureau of Engineering. Our redesigned robot, web interface and database system will assist in collecting and displaying sidewalk slope monitoring data in order to help the City of Los Angeles to get their sidewalks to reach Americans with Disabilities Act Compliance standards. To fulfill the new design of the project we had a couple of items we had to take into consideration such as privacy and public safety concerns.

Since our project is sponsored by the City of Los Angeles Bureau of Engineering, which is a part of a government agency we must take into consideration the privacy of their data. This means that any data that we will be collecting and storing using the robot, web interface and database systems we have designed must be kept secure and private. Some of the information such as the GPS coordinates and photos taken by the rover is a privacy concern. Some examples of GPS coordinate data and photos that could be a privacy threat are those of private or residential locations and if not protected could lead to personal data being stolen or disclosed. We also had to take into consideration the privacy and security of our database system in order to keep all data safe. Our solution to combat this issue is that we are currently working with the BOE team to research the best database security structure.

Furthermore, since our developed software for the City of Los Angeles focuses on fixing sidewalks that do not meet ADA compliance, we must take into consideration the legal and ethical implications of public safety when it comes to the rover. We have to take necessary precautions to ensure that our rover user interface is efficiently designed so that rover movements will be precise. We took this into consideration since a malfunction of the rover can cause physical harm to the field worker operating the rover or to any pedestrian that is on the street. To combat this issue our user interface has ensured that all buttons are big enough so that there are no discrepancies in controlling the rover and we will conduct extensive testing once the rover has been completely built.

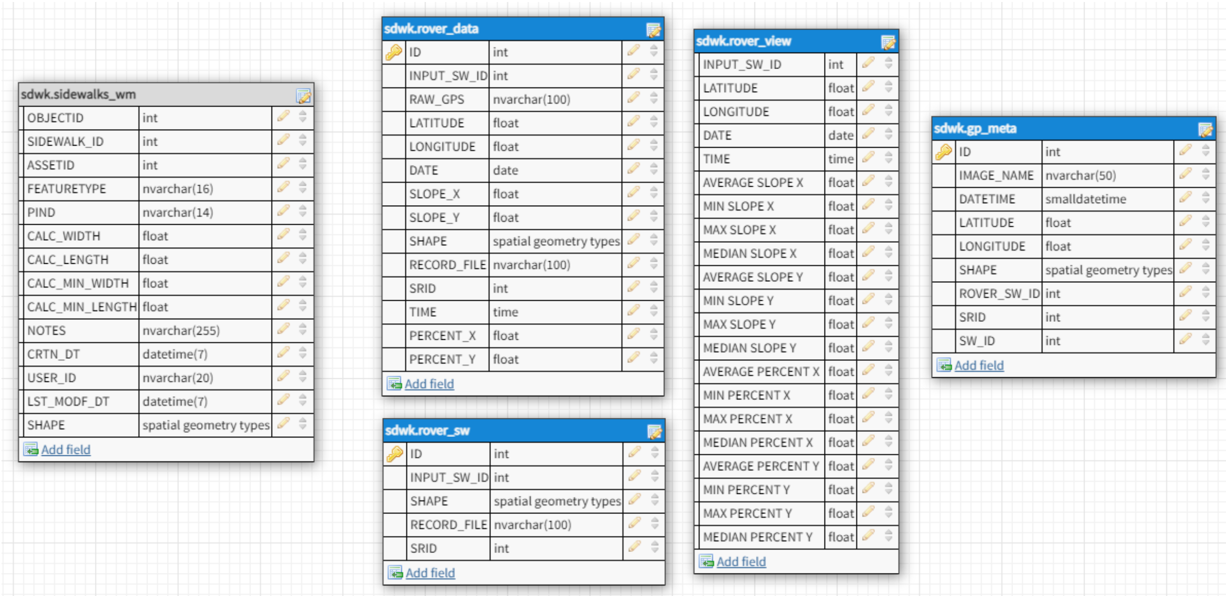
In conclusion, when developing our software, we took into consideration the privacy and public safety concerns that surround our project. We took into account that privacy is crucial for the City of Los Angeles Bureau of Engineering and we are currently collaborating with BOE to choose the most secure platform. Also, since data collection will be done by field workers who will be navigating the rover, we made sure to make our interfaces as simple to navigate as to prevent any accidents from causing bodily harm to any pedestrians or field workers. Our software will be compliant with all legal and ethical standards to ensure the success of the project.

Appendix A: Glossary

BOE	City of Los Angeles, Bureau of Engineering
SDD	Software Design Document
SRS	Software Requirements Specification
Rover	Device with wheels that will display the GUI software.
IMU	Inertial Measurement Unit
DB	Database
NavLA	NavigateLA (Site)
EXIF	Exchangeable Image File
IDE	Integrated Development Environment
GUI	Graphical User Interface

Appendix B: Analysis Models

Azure SQL database Data Schema



Appendix C: To Be Determined List

Not Applicable.