Software Requirements Specification

for

A.I for Smart Cities: Pedestrian and Bicycle Safety

Version 1.0, approved

Prepared by Marcio Arakaki, Alejandro Bernal, Haley Gray, Freddy Gutierrez, Matthew Ramos

City of Los Angeles and LADOT
September 3, 2019
# Table of Contents

Table of Contents..................................................................................................................pg 2  
Revision History....................................................................................................................pg 2  
1. Introduction..................................................................................................................pg 4  
   1.1. Purpose....................................................................................................................pg 4  
   1.2. Intended Audience and Reading Suggestions.........................................................pg 4  
   1.3. Product Scope........................................................................................................pg 4  
   1.4. Definitions, Acronyms, and Abbreviations ...........................................................pg 4  
   1.5. References...............................................................................................................pg 6  
2. Overall Description..................................................................................................... pg 7  
   2.1. Product Perspective................................................................................................pg 7  
   2.2. Product Functions...................................................................................................pg 7  
   2.3. User Classes and Characteristics............................................................................pg 9  
   2.4. Operating Environment...........................................................................................pg 9  
   2.5. Design and Implementation Constraints.................................................................pg 10  
   2.6. User Documentation..............................................................................................pg 10  
   2.7. Assumptions and Dependencies.............................................................................pg 10  
   2.8. Apportioning of Requirements..............................................................................pg 11  
3. External Interface Requirements................................................................................pg 12  
   3.1. User Interfaces........................................................................................................pg 12  
   3.2. Hardware Interfaces................................................................................................pg 13  
   3.3. Software Interfaces................................................................................................pg 14  
   3.4. Communications Interfaces....................................................................................pg 14  
4. Requirements Specification.........................................................................................pg 16  
   4.1. Functional Requirements.........................................................................................pg 16  
   4.2. External Interface Requirements............................................................................pg 18  
   4.3. Logical Database Requirements.............................................................................pg 19  
   4.4. Design Constraints...............................................................................................pg 19  
5. Other Nonfunctional Requirements...........................................................................pg 21  
   5.1. Performance Requirements....................................................................................pg 21  
   5.2. Safety Requirements..............................................................................................pg 21  
   5.3. Security Requirements............................................................................................pg 21  
   5.4. Software Quality Attributes....................................................................................pg 22  
   5.5. Business Rules........................................................................................................pg 22  
Appendix A: Glossary..........................................................................................................pg 23  
Appendix B: To Be Determined List...................................................................................pg 24  

# Revision History
<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haley Gray, Freddy Gutierrez, Marcio Arakaki, Alejandro Bernal, Matthew Ramos</td>
<td>10/01/19</td>
<td>Revised Sections 1, 2, 3, 4, 5</td>
<td>1.0</td>
</tr>
<tr>
<td>Haley Gray</td>
<td>10/10/19</td>
<td>Revised Sections 2, 4</td>
<td>1.0</td>
</tr>
<tr>
<td>Freddy Gutierrez, Alejandro Bernal</td>
<td>10/13/19</td>
<td>Revised Sections 1, 2, 3, 4</td>
<td>1.0</td>
</tr>
<tr>
<td>Haley Gray</td>
<td>11/15/19</td>
<td>Revised Sections 1, 2, 3, 4</td>
<td>1.0</td>
</tr>
<tr>
<td>Haley Gray, Freddy Gutierrez, Marcio Arakaki, Alejandro Bernal, Matthew Ramos</td>
<td>12/06/19</td>
<td>Revised all sections for final draft.</td>
<td>1.0</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Purpose

The purpose of this document is to explain in detail the functions the following applications will perform. This document will cover all aspects of the software for each project.

I. Metro Bike Share Real Time (MBSRT)
   - The current version of the MBSRT map is 1.0.0. Being that it is the first iteration, there are no current revisions or release numbers.

II. Bicycle Accident Visualization
    - The current version of the Bicycle Accident Visualization Application is 1.0.0. Being that it is the first iteration, there are no current revisions or release numbers.

III. Metro Bike Share Historical Data Visualization
    - The current version of the Metro Bike Share Historical Visualization Application is 1.0.0. There are no current revisions or release numbers.

1.2 Intended Audience and Reading Suggestions

The main audience of the software requirements specifications document are developers, project managers, government workers, and testers. The SRS is organized by sections and includes divisions for each application pertaining to this SRS. Due to several applications included in the SRS, it is suggested to read through the Product Scope first to understand the purpose of each application. For developers, some of the main points to focus on are the software requirements for each application including the programming languages and APIs. Project managers can focus on overall project functionalities, including the interface, requirement specifications, and external interface requirements. Testers may review the overall description for understanding application functions and operating system environment.

1.3 Product Scope

I. Metro Bike Share Real Time
The software making use of Metro’s Bike Share Real time data displays a map that will be referred to as Metro Bike Share Real Time (MBSRT). This map will fetch geojson data from the Metro’s Bike Share web app, and display a marker, representing a station, using Google’s Map API. From there the user will be able to interact with numerous built in features, such as filtering, directions, heatmaps, etc. Currently the purpose of our software is to help users around the greater Los Angeles area to obtain a bike, and cycle around the city using the safest way possible. Ideally contributing to our ultimate goal of realizing Vision Zero.

II. Bicycle Accident Visualization

The Bicycle Accident Visualization Application maps historical data pertaining to bicycle accidents across Los Angeles from 2012 to 2018 that include at least one injured bicyclist. Additionally, the application maps Los Angeles designated bikeways. The application offers several features for viewing bicycle accidents including the ability to filter by accident year, view accidents within a radius of mouse click, and retrieve counts of bicycle accidents for an area. The goal of the application is visualize accident data and expose patterns that will help to make Vision Zero a success.

III. Metro Bike Share Historical Data Visualization

This software is dealing with Metro Bike Share Historical data displayed on a map. With this data we will manipulate the data so that we can get specific data so that we can see patterns of Metro Bike Share stations popular days and peak times. The purpose of this software is to help Metro decide where more bikes need to be placed or where new stations may need to be added. This contributes to our goal of Vision zero with our hopes more people are encouraged to ride bikes around the city instead of cars, ultimately reducing casualties.

1.4 Definitions, Acronyms, and Abbreviations

- **AISC**: A.I. for Smart Cities: Pedestrian and Bicycle Safety
- **API**: Application programing interface
- **DFD**: Data Flow Diagram
- **SRS**: Software Requirements Specifications
- **SDD**: Software Design Document
- **MBSRT**: Metro Bike Share Real Time
## 1.5 References

The following table contains the references that are most often referred to for this application.

<table>
<thead>
<tr>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcGIS</td>
<td>All references to ArcGIS services. <a href="https://doc.arcgis.com/en/">https://doc.arcgis.com/en/</a></td>
</tr>
<tr>
<td>Metro Bike Data</td>
<td>Anonymized Metro Bike Share trip data for data collection <a href="https://bikeshare.metro.net/about/data/">https://bikeshare.metro.net/about/data/</a></td>
</tr>
<tr>
<td>Google Maps API</td>
<td>The Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries. <a href="https://developers.google.com/maps/documentation/javascript/tutorial">https://developers.google.com/maps/documentation/javascript/tutorial</a></td>
</tr>
<tr>
<td>Jupyter Notebook</td>
<td>Organize and manipulate data.</td>
</tr>
</tbody>
</table>
2. Overall Description

The overall product shall satisfy the Product Perspective, Functions, User Classes, and Operating Environment. The application shall provide Documentation and inform users of any Constraints, Assumptions, and Dependencies.

2.1 Product Perspective

I. Metro Bike Share Real Time

MBSRT relies on the real time data set provided on the Metro Bike Share website. This data is provided in a geojson format. MBSRT is also reliant on a filtered version of the “Los Angeles Collisions 2012 Through 2018,” data set from the Los Angeles Geohub website. All views and functionalities of our map are built using Javascript, HTML and CSS. Our map also relies on Maps Javascript API for certain features. The Metro Bike Share website contains a map that displays bike share stations around the city, similar to ours. However, our map is superior when it comes to user interactions by allowing the user to do much more with it.

II. Bicycle Accident Visualization

The product is a visualization of bicycle accident data available on Los Angeles Geohub using the ArcGIS Online Javascript API. The application uses the Feature Layer from the dataset, “Los Angeles Collisions 2012 Through 2018,” filtering for bicycle accidents from 2012 - 2018 that include at least one injury, and displays the location of accidents on a map of Los Angeles with visualization for accident severity. While a map of bicycle accidents is available on Los Angeles Geohub, the map is not concise requiring many levels of zoom to reach Los Angeles along with non-intuitive filtering by the user to reach a view of bicycle accidents with injuries. Furthermore, it is not possible to see distinctions between accident severity from just a map view nor are designated bikeways represented.

III. Metro Bike Share Historical Data Visualization

This product will help visualize Metro Bike Share Historical data. We used ArcGis to create our own API for this data. This application uses feature layers from the transactions dataset of Metro Bike Share Historical data allowing the user to filter through days of the week to find out a stations data history. All views and functionalities of our map are built using Javascript, HTML and CSS. Our map is the first of its kind for Metro Bike Share Historical Data.

2.2 Product Functions

The following list contains the major functionalities of each application:

I. Metro Bike Share Real Time
● Display a marker representative of each station in the Metro Bike Share geojson dataset.

● Ask for the user’s permission to get their location. If agreed to a marker is dropped at their location.

● Create a clickable list item for each station. When clicked, the map zooms in to the corresponding marker on the map and animates it.

● Allow the user to filter stations by station cities.

● Toggle a heatmap representative of each bike accident in the geohub data set.

● On the click of a station, display a polyline and directions of how to get from the user’s location to the clicked station.

● Allow the user to draw his/her path when toggling the draw polyline button.

● Reset the map to its original format when clicking the reset map button.

II. Bicycle Accident Visualization

● The application shall display bicycle accidents with injuries from 2012 to 2018.

● The application shall visualize accident severity using color and circle size.

● The application shall displays designated bikeways using different colors.

● The application shall allow the user to get location of accidents within radius of a mouse click.

● The application shall provide statistical counts of bicycle accidents from predefined areas.

III. Metro Bike Share Historical Data Visualization

● The application shall display data regarding metro bike share bikes.

● The application shall display information about each bike share station.

● The application shall allow the user to filter data based on the days of the week for each station.

● The application shall visualize peak hours and days of each station.

● The application shall provide statistical counts of metro bike share trips
2.3 User Classes and Characteristics

The different classes and their characteristics that will use each application are:

I. Metro Bike Share Real Time
   A. General Public: This user will be able to interact with all current features available on the map, but will not be able to change any features.
   B. Developer: This user will be able to add more features to the map.

II. Bicycle Accident Visualization
   A. General Public: The users shall be able to view bicycle accident visualizations and use the UI filters for viewing bicycle accidents.
   B. Developer: The users shall be able view bicycle accident visualizations, use the UI filters for viewing bicycle accidents, and modify the application source code.

III. Metro Bike Share Historical Data Visualization
   A. Developer: Update records and modify application
   B. User: Able to view and use all the features in the application.

2.4 Operating Environment

For both end-users and developers, the applications shall work on Windows, Mac, and Linux operating systems. The application shall run in a web browser including Chrome, FireFox, Safari, and Internet Explorer and requires an internet connection.

I. Metro Bike Share Real Time
   - Because MBSRT is reliant on Maps Javascript API this API must not be deprecated. Similar to Maps Javascript API, MBSRT must also have access to the Metro Bike Share real time geojson data set.

II. Bicycle Accident Visualization
Since the application uses a Feature Layer Service hosted on Los Angeles Geohub, the application must be able to connect with Los Angeles Geohub. Additionally, the application uses the ArcGIS Online Javascript API 4.13 requiring the API to be accessible by the application.

III. Metro Bike Share Historical Data Visualization
   ● Since this application uses a Feature Layer Service hosted by ArcGIS Online. The application also require the API to accessible by the application.

2.5 Design and Implementation Constraints

The following various operations and tasks may affect the product’s timetable:

I. Metro Bike Share Real Time
   ● Designing our own safest path algorithm.

II. Bicycle Accident Visualization
   ● Point Clustering in not supported in the ArcGIS Online Javascript API 4.x and later.
   ● Learning the ArcGIS Online Javascript API 4.13.

III. Metro Bike Share Historical Data Visualization
   ● Learning the ArcGIS Online Javascript API 4.13.

2.6 User Documentation

Tutorials shall be provided to show step-by-step how to utilize each application.

2.7 Assumptions and Dependencies
The following factors are assumptions that could affect the requirements:

I. **Metro Bike Share Real Time**  
   ● User accepts location permission when prompted.  
   ● Metro Bike Share keeps their real time data public, and in the same format.  
   ● Developers are provided with a Google Maps API key that has access to required APIs.

II. **Bicycle Accident Visualization**  
   ● Data sets remain on Los Angeles Geohub and in the same format.

III. **Metro Bike Share Historical Data Visualization**  
   ● Metro Bike Share keeps their historical data public and in the same format.

2.8 **Apportioning of Requirements**

In the case that the project is delayed, some requirements may be transferred to the next version of the application.
3. External Interface Requirements

The external interface requirements detail the User Interfaces, Hardware Interfaces, Software Interfaces, and Communication Interfaces of the application.

3.1 User Interfaces

The list below contains the requirements for the user interfaces of each application:

I. Metro Bike Share Real Time

- Metro Bike Share real time data shall be used to create station markers on the map, as well as the station items list.
- Hovering over a station marker shall display an info window with information about the corresponding station.
- Hovering off a station marker shall close its info window.
- User Location button shall prompt the user for their location when clicked.
- Reset Map button shall be used to reset the map to its original state.
- Toggle Heatmap button shall be used to toggle heatmap either on or off.
- Draw Polyline button shall be toggled on or off to allow the user to draw his/her path by clicking anywhere on the map.
- Station list item shall be highlighted, and its corresponding maker should be animated on the click of a list item.
- Filter city dropdown shall display only markers with the corresponding city on the map.
- Clicking on a station marker shall display directions and a polyline of how to get from the user’s location to the clicked station.
- Default Google Map’s features shall work as expected.

II. Bicycle Accident Visualization

- Bicycle Accident Feature Layer shall use esriGeometryPoint to render bicycle accidents on a map.
- Polygon Map Layer shall use esriGeometryPolygon for rendering polygon areas on the map.
- Bikeway Map Layer shall use esriGeometryPolyline objects to display designated bikeways on the map.
- Layer list shall allow the user to turn layers on and off for better clarity.
- Clicking a bicycle accident marker shall provide details about the accident.
- Clicking a polygon area shall provide counts of bicycle accidents in that area.
- A pop out widget shall allow the user to filter the display of each bikeway.
- A pop out widget shall allow the user to filter the display of bicycle accident year.
- The application shall use the CSS “Dark Theme” provided by ArcGIS Online.

III. Metro Bike Share Historical Data Visualization

- Metro Bike Share data shall be used to create station markers on the map.
- Clicking on a marker shall display information about the station such as average trips throughout the week.
- The application has a map legend at the bottom right which the user can scroll through.
- A drop down menu shall let the user filter the marker by day of the week.
- Choosing to filter by day shall change the marker to display information regarding hours.
- Choosing to filter by day shall change the map legend to show by hours instead of days.

3.2 Hardware Interfaces

This applications do not have any hardware interface requirements.
3.3 Software Interfaces

Listed are the requirements for software interfaces of the applications:

I. Metro Bike Share Real Time
   ● On the load of the page an API call will be made to the Metro Bike Share real
time data set
   ● Maps Javascript API will be used to visualize data, and for certain map features

II. Bicycle Accident Visualization
   ● ArcGIS Online Javascript API 4.13 shall be used to visualize bicycle accidents
and designated bikeways.
   ● The Feature Layer from Los Angeles Geohub shall be used for bicycle accident
data.
   ● The Map Layer from Los Angeles Geohub shall be used for designated bikeways.
   ● The Map Layer from Los Angeles Geohub shall be used for the polygon layer on
the map.

III. Metro Bike Share Historical Data Visualization
   ● ArcGIS Online JavaScript API shall be used to visualize metro bike share stations
trips.
   ● The feature layer created from ArcGIS Online services shall be used for metro
bike historical data.
   ● The Map Layer from ArcGIS Online services shall be used for polygon layer on
the map.

3.4 Communications Interfaces

Listed are the requirements for the communication interface:

I. Metro Bike Share Real Time
   ● The application shall receive data in the form of Geojson via HTTP Request.
II. Bicycle Accident Visualization

- The application shall receive data using HTTPS requests in a web browser.

III. Metro Bike Share Historical Data Visualization

- The application shall receive data using HTTPS requests in a web browser.
4. Requirements Specification

Requirements for each application is provided below:

4.1 Functional Requirements

I. Metro Bike Share Real Time Module

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1</td>
<td>This system shall fetch a Geojson object from the Metro Bike Share real time data set, and display a marker on the map for each station.</td>
</tr>
<tr>
<td>4.1.2</td>
<td>This system shall display an info window on the hover of a station.</td>
</tr>
<tr>
<td>4.1.3</td>
<td>This system shall remove an info window when hovering off a station.</td>
</tr>
<tr>
<td>4.1.4</td>
<td>The system shall prompt the user for his/her location when the “User Location” button is clicked.</td>
</tr>
<tr>
<td>4.1.5</td>
<td>The system shall reset the map to its original state when the “Reset Map” button is clicked.</td>
</tr>
<tr>
<td>4.1.6</td>
<td>The system shall toggle a heatmap representative of each accident in the Geohub data set when the “Toggle Heatmap” button is clicked.</td>
</tr>
<tr>
<td>4.1.7</td>
<td>The system shall draw a polyline on the map when the “Draw Polyline” button is toggled on, and the map is clicked.</td>
</tr>
<tr>
<td>4.1.8</td>
<td>The system shall highlight a station item and move the map to the corresponding marker on the map when the station item is clicked.</td>
</tr>
<tr>
<td>4.1.9</td>
<td>The system shall display only markers on the map which have the same value as the “Filter By City” dropdown.</td>
</tr>
</tbody>
</table>

II. Bicycle Accident Visualization Module
| 4.1.1 | The system shall use the Feature Layer Service provided by Los Angeles Geohub and the dataset, Los_Angeles_Collisions_2012through2018, to retrieve bicycle accident data. |
| 4.1.2 | The system shall use the accident data to display markers on a map of Los Angeles. |
| 4.1.3 | Accident severity shall be visualized using colors from yellow and increasing to red to indicate increasing severity. |
| 4.1.4 | Accident severity shall be visualized using increasing marker size for the number of injuries. |
| 4.1.5 | Accident information including year and time shall be displayed in a popup. |
| 4.1.6 | The system shall use the Map Layer Service provided by Los Angeles Geohub and the dataset, Bikeways (Existing), to retrieve bikeway data. |
| 4.1.7 | The system shall use bikeway data to display bikeways on a map of Los Angeles. |
| 4.1.8 | The system shall use different colors to distinguish types of bikeways. |
| 4.1.9 | The system shall use the Map Layer Service provided by Los Angeles Geohub and the dataset, 2000 Census Tracts, to retrieve polygons. |
| 4.2.0 | The system shall use the polygon data to display the polygons on the map. |
| 4.2.1 | The system shall provide counts of bicycle accidents within the area of the polygons in a popup. |
| 4.2.2 | The system should have a function to display accidents within a radius of a mouse click. |
| 4.2.3 | The system should allow the user to change the radius for displaying accidents. |
| 4.2.4 | The system should allow for filtering of bicycle accidents displayed on the map by year. |
| 4.2.5 | The system should allow for filtering of bikeways displayed on the map by type. |
III. Metro Bike Share Historical Data Visualization

<table>
<thead>
<tr>
<th>4.1.1</th>
<th>The system shall use the Feature Layer Service provided by ArcGIS Online services and the dataset, Metro Bike Share Historical data, to retrieve trip history data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2</td>
<td>The system shall use the station location to display markers on a map of Los Angeles.</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Metro bike share stations peak hours shall be visualized using colors and opacity features.</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Metro bike share stations popularity shall be visualized by the size of the polygon marker.</td>
</tr>
<tr>
<td>4.1.5</td>
<td>The system shall use the polygon data to display the polygons on the map.</td>
</tr>
<tr>
<td>4.1.6</td>
<td>The system should have a function to display a pop-up window of trip statistics.</td>
</tr>
<tr>
<td>4.1.7</td>
<td>The system should allow the user to filter through the days of the week for more visualization of each station.</td>
</tr>
</tbody>
</table>

4.2 External Interface Requirements

I. Metro Bike Share Real Time
   - This interface will display the stations in the Metro Bike Share real time data set, where he/she will be able to interact with the map. More information about the interface is further detailed in Section 3.

II. Bicycle Accident Visualization
   - This interface will display bicycle accidents with injuries data set, where the user shall be able to interact with the map. More information about the interface is further detailed in Section 3.
III. **Metro Bike Share Historical Data Visualization**

   - This interface will display metro bike share stations trip history data set, where the user shall be able to interact with the map. More information about the interface is further detailed in [Section 3](#).

### 4.3 Logical Database Requirements

None of the applications require a logical database.

### 4.4 Design Constraints

#### I. Metro Bike Share Real Time

   - **Standard Limitation**
     - Limited experience with javascript
     - Limited experience with css
     - No prior experience using Maps Javascript API
     - No prior experience using ArcGIS
     - Limited experience using Git

   - **Hardware Limitation**
     - Application must run on browsers that support implemented features
     - System must have access to the internet

#### II. Bicycle Accident Visualization

   - **Standard Limitation**
     - No prior experience using ArcGIS Online Javascript API
     - Limited experience using Git

   - **Hardware Limitation**
     - Application must run on browsers that supports ArcGIS requirements
     - System must have access to the internet

#### III. Metro Bike Share Historical Data Visualization

   - **Standard Limitation**
- No prior experience using ArcGIS Online Javascript API
- Limited experience using Git

**Hardware Limitation**
- Application must run on browsers that supports ArcGIS requirements
- System must have access to the internet
5. Other Nonfunctional Requirements

In other nonfunctional requirements, the application shall detail the performance, safety, and security requirements. The application shall include the software quality and business rules.

5.1 Performance Requirements

Listed are the performance requirements:

I. Metro Bike Share Real Time
   - The machine must be able to process real time data.

II. Bicycle Accident Visualization
   - No other nonfunctional requirements have been identified.

III. Metro Bike Share Historical Data Visualization
    - No other nonfunctional requirements have been identified.

5.2 Safety Requirements

No safety requirements were identified for these applications.

5.3 Security Requirements

The following are the safety requirements for each application:

I. Metro Bike Share Real Time
   - User must agree to allow Google to access their location in order to display their location and calculate their route.

II. Bicycle Accident Visualization
• No security requirements at the moment.

III. Metro Bike Share Historical Data Visualization
• No security requirements at the moment.

5.4 Software Quality Attributes
Software quality attributes of the applications:
• Adaptability: Currently for desktop only. No mobile devices.
• Availability: It is accessible through a website application.
• Correctness: Only display the data provided by GeoHub and Metro Bike Share.
• Maintainability: Must update data every week or so.
• Usability: Simple User Interface for all users.

5.5 Business Rules
The following are business rules about the product:
• All data on metro bikes and accidents are open to the public.
• All features are accessible to all users.
• Permission is needed to access user’s current location.
# Appendix A: Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application program interface</td>
</tr>
<tr>
<td>ArcGis</td>
<td>Esri’s all-in-one solution to work with geographic information.</td>
</tr>
<tr>
<td>AISC</td>
<td>A.I. for Smart Cities: Pedestrian and Bicycle Safety</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet is a style sheet that is used to describe the</td>
</tr>
<tr>
<td></td>
<td>presentation of a markup language.</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values. File format that is used to store tabular data</td>
</tr>
<tr>
<td></td>
<td>such as spreadsheets or databases.</td>
</tr>
<tr>
<td>DFD</td>
<td>Data Flow Diagram</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language is the standard markup language for creating</td>
</tr>
<tr>
<td></td>
<td>web pages.</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol is an application protocol for distributed,</td>
</tr>
<tr>
<td></td>
<td>collaborative, hypermedia information systems</td>
</tr>
<tr>
<td>Javascript</td>
<td>A programming language that is heavily used for web applications</td>
</tr>
<tr>
<td>LADOT</td>
<td>Los Angeles Department of Transportation</td>
</tr>
<tr>
<td>MBSRT</td>
<td>Metro Bike Share Real Time</td>
</tr>
<tr>
<td>Machine learning</td>
<td>Predictive mathematical model used for predictions</td>
</tr>
<tr>
<td>Operating System</td>
<td>The software that allows any computer to communicate, modify, and</td>
</tr>
<tr>
<td></td>
<td>terminate any hardware and software communications based on end-users</td>
</tr>
<tr>
<td>Python</td>
<td>A general-purpose programming language that can also be used to program</td>
</tr>
<tr>
<td></td>
<td>web application and data analytics application.</td>
</tr>
<tr>
<td>Runtime</td>
<td>The time when an application is executed.</td>
</tr>
<tr>
<td>SDD</td>
<td>Software Design Document</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specifications</td>
</tr>
</tbody>
</table>
Appendix B: To Be Determined List

The following requirements and conditions are to be determined:

- Using machine learning to begin predicting patterns.