Software Design Document (SDD)

Team Name: Group 2
Jacob Vasquez, Dominick Daito Jr. , Jolen Reid, Francis Agyemang

Date: December 5, 2024

Contents

Version Description

Version Number	Description	Date Added
1.0	Initial draft	December 11, 2024

1 Introduction

1.1 Purpose of the Document

This document is to explain in detail the functions that the application will perform. The document will inform readers as to what the application will do. The purpose of this product is to visualize pedestrian and bicycle data to find and identify problem areas and the safest navigation routes.

1.2 Intended Audience

The main audience of the software requirements specifications document are developers, project managers, and testers. The SRS contains information about each project such as what the project is, what each of its UI elements should do, and what dependencies each project may have. It is suggested that you first look at the table of contents for any topics you may be looking for, if not then quickly skim the document to get a better understanding of the projects. If you are a developer or project manager it is suggested that you look into section 4 of the SRS so that you may check if project requirements are being met. If you are a tester it is suggested that you look into section 3 so that you have a better understanding of how the user interfaces should work.

1.3 Overview of the System

I. Metro Bike Share Real Time (Web and Android App) The focus of this project is to aid in the realization of Vision 0 is the greater Los Angeles area. This is done by modeling real time Metro Bike Share stations within the city, in conjunction with bike accidents around the city. Currently the 4 idea is to allow the user to draw the "safest" path by avoiding areas of the city where major accidents have occured. II. Bicycle Accident Visualization III. Metro Bike Share Historical Data Visualization (Web and Android App) The purpose of this project is to help visualize the Metro Bike Share data. The stations are shown as feature layers on the map. Clicking the station displays information such as number of trips and busiest day. Station icons vary in size and color depending on what information we want to show.

2 System Architecture

2.1 Workflow of the System

Level 0: Data Engines Overview
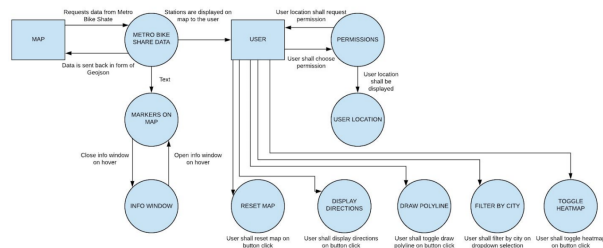
2.2 Breakdown of Components

- Use of a product(programming language, database, library, etc)

  – Java
  – JavaScript
  – HTML CSS
  – Python
  – Pandas Library
  – SkLearn
  – Google Maps API
  – Android Studio
  – Maps SDK for Android
  – Maps SDK for Android Utility Library
  – Directions API
  – Firebase Database
  – ArcGIS
  – JSON/GeoJSON
  – ArcGIS Runtime SDK

- Reuse of existing software components to implement various parts or features of the system

  – This software is the first version, no reuse of existing software components

- Plans for extending or enhancing the software

  – Collect data in real-time
  – Create a database to hold all data
  – Use Python libraries to manipulate the data

- User Interface paradigms (or system input and output models)

  – Physical mouse required to interact with the application
  – Computer is required to use the application

- Hardware and/or software interface paradigms

- User's interface will be updated using JavaScript

- Error detection and recovery

  - The error can be checked in the console log in order to see if the data loaded

- External databases and/ or data storage management and persistence

  - Made use of historical data using Geohub
  - Made use of real time data from Metro Bike Share website
  - Firebase

- Management of other resources

  - No other external resources used

3 User Interface

3.1 How to Use the System

From the user interface, the user can:

- Zoom in and out to change map visibility

- Hovering over the marker displays an info window about the station

- User Location button to prompt the user for their location

- Reset map to its original state using the reset button

- Toggle heat map on and off

- Draw a Polyline to allow the user to draw his/her path by clicking anywhere on the map. This can be toggled on/off

- Click on the station to be highlighted and its corresponding marker will be animated

- Filter by city with the drop-down menu

- Click on a station marker to display directions and a polyline of how to get from the user's location to the clicked station

- Use Google Map's default features

3.2 Database Design and Explanation

No databases are used in our projects for now.

3.3 Screenshots (Optional)

4 Glossary

Acronym	Definition
UI	User Interface
API	Application Programming Interface
DB	Database
ArcGis	Esris all in one solution to work with geographic information
AISC	A.I. for Smart Cities: Pedestrian and Bicycle Safety
CSS	Cascading Style Sheet is a style sheet that is used to describe the presentation of a markup language.
CSV	Comma Separated Values. File format that is used to store tabular data such as spreadsheets or databases.
DFD	Data Flow Diagram.
Firebase	BaaS for cloud storage and authentication. HTML
BaaS for cloud storage and authentication	
HTTP	Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems.
Javascript	A programming language that is heavily used for web scripts.
LADOT	Los Angeles Department of Transportation.
MBSRT	Metro Bike Share Real Time
Machine learning	Predictive mathematical models are used for predictions.
Operating System	The software allows any computer to communicate, modify, and terminate hardware and software communications based on end-users decisions.
Python	A general-purpose programming language that can also be used to program web applications and data analytics applications.
Runtime	The time when an application is executed.
SDD	Software Design Document.
SRS	Software Requirements Specifications.
SDK	Software Development Kit.

5 References

- ArcGIS All references to ArcGIS services. https://doc.arcgis.com/en/

- GeoHub Data collection for the city of Los Angeles http://geohub.lacity.org/

- Metro Bike Data Anonymized Metro Bike Share trip data for data collection https://bikeshare.met

- 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. https://developers.google.com/maps/documentation/javascript/tutorial

- Android Studio Used to develop MBSRT android version. https://developer.android.com/docs

- Maps SDK for Android. Adds functionality to elements within the map https://developers.google.c sdk/intro

- Google's Directions API Used to retrieve a JSON object containing directions information between points https://developers.google.com/maps/documentation/directions/start

- Maps SDK for Android Utility Library Used to decode the directions polyline from the Directions JSON object. Used to add heatmap layer over map https://developers.google.com/r sdk/utility

- Firebase Used to authenticate and store user data. https://firebase.google.com/

- Jupyter Notebook Organize and manipulate data. https://jupyter.org/