Social Distance Assistant

Software Design Specification

Group 5 - 5-31-2020 - v2.1

Irfan Filipovic (IF), Shane Folden (SF), Man Him Fung (MF), Mason Jones (MJ), Siqi Wang (SW)

Table of Contents 1

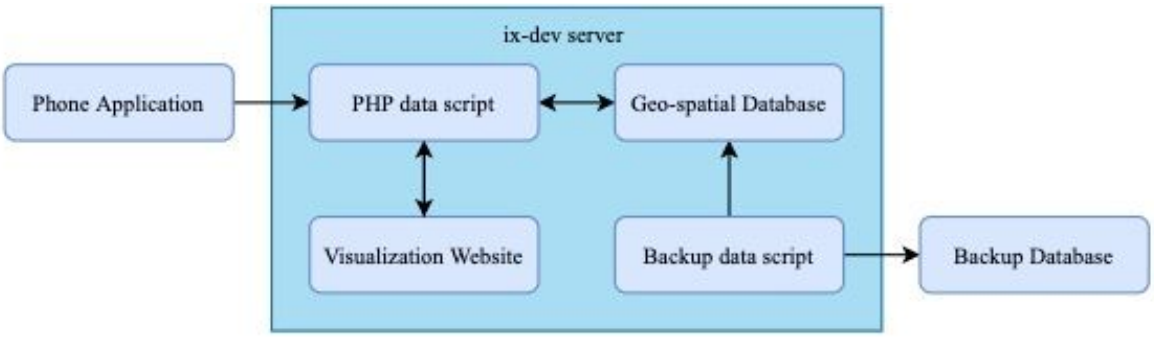
1. SDS Revision History

Date	Author	Description
		Update page numbers
6-1-2020	Group	v3.0. - Finalized all revisions
5-31-2020	IF	v2.1 - Added diagram descriptions and titles.
5-29-2020	IF	v2.0 - Updated dynamic diagrams, fixed section 2 & 3
5-15-2020	Group	v1.3 - Added diagrams, revised, and finalized document
5-14-2020	Group	v1.2 - Initial Project 2 SDS information added
5-11-2020	Group	v1.1 - Created simple Project 2 SDS skeleton

2. System Overview

A user will be granted access to a website which will display other user locations with additional information, and an application which will assist in tracking location with options to test the connections and transmission of data. These two user accessed components will be connected via the ix-dev server using php scripts to store data in a MySQL database. In order to prevent loss of much data a backup of the database will be made using a script. The application will post data to the database, which will then be used to present the information to users on the website.

3. Software Architecture



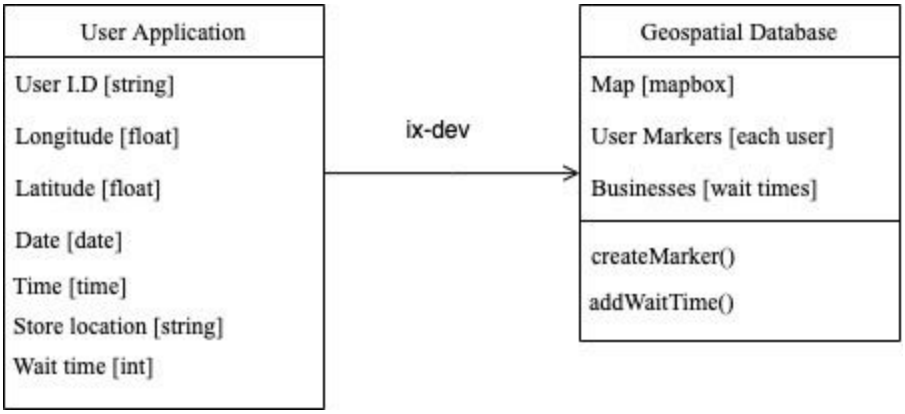
<Figure 1> Diagram displaying component interaction.

- **Geo-spatial Database:**
  - The geo-spatial database holds all the location data and is connected to each component via the ix-dev server
  - The database will store data retrieved by the application, and provide this data for the visualization website
- **Backup Database:**
  - We will be regularly backing up our geo-spatial database file to a new .sql file
  - Data will be backed up to our private GitHub repository
- **ix-dev Host Server:**
  - Hosts the database, various scripts, and the "public_html" folder where our website components are stored
  - The ix-dev server will host interaction between the geo-spatial database, phone application, and visualization website. It will also push the backup database files to GitHub using a script
- **Phone Application:**
  - Our phone application will work on Apple devices.
  - It will use the iOS Standard Location Service to obtain location data.
  - It will send the data to the geo-spatial database via the website's .php backend page
- **Visualization Website and .php Backend:**
  - Data will be fetched from the geo-spatial database to be presented visually
  - Website will interact with mapbox to display user location data

4. Software Modules

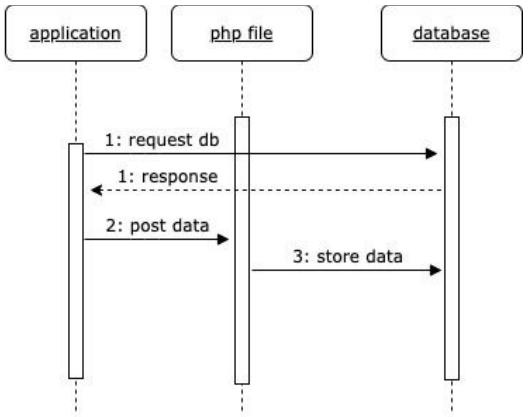
4.1. Geo-spatial Database

A. Host the database which is used for central data storage
B. The database interfaces with the ix-dev server and the backup server



<Figure 2> Transfer of data from User application to Database hosted by ix-dev server

C. The location database receives information for a user's unique device id, and then stores that information in the database. On a user input to enter wait time that information is passed and stored under the store indicated by the user



<Figure 3> Diagram displaying interaction sequence between the application and database

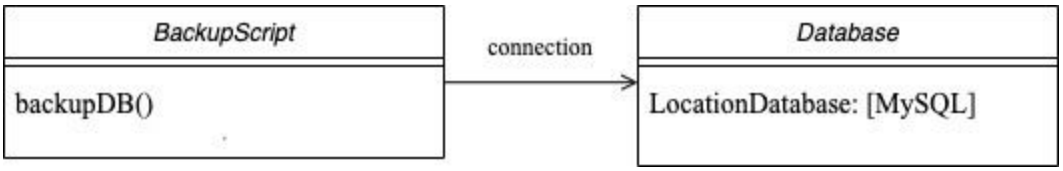
D. Upon launch the user application will attempt to connect to the location database. If there is an error on connection the user will be notified, otherwise the user's data will be sent to the php file which will store it within the database

E. This is used to store and connect to other components after we obtain user geo-location. This is also where data will be fetched from in order to create the visual website

4.2. Backup Database

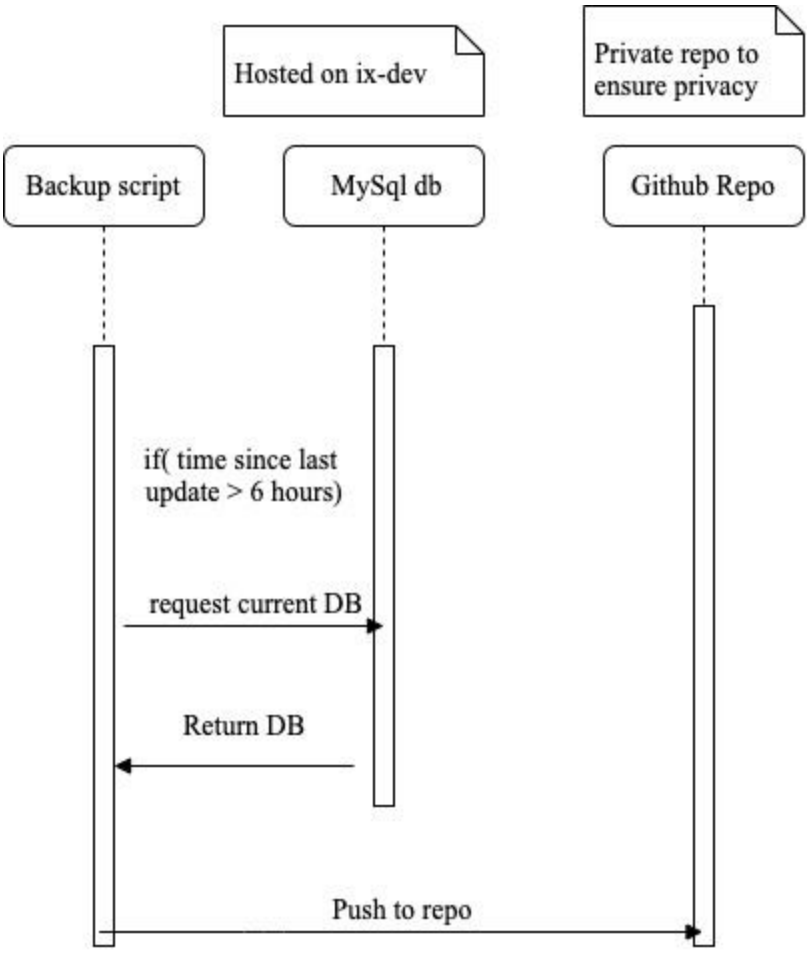
A. This module's basic function is to be a backup in case the primary MySQL database goes down

B. By running a script on the ix-dev server, the database file will be copied and uploaded to a private GitHub repository



<Figure 4> The backup script's only function is to create a copy and store it on Github..

C. The diagram denotes that on a backup call the backup script must interact with the database. The location database is then copied and posted to the private Github repository



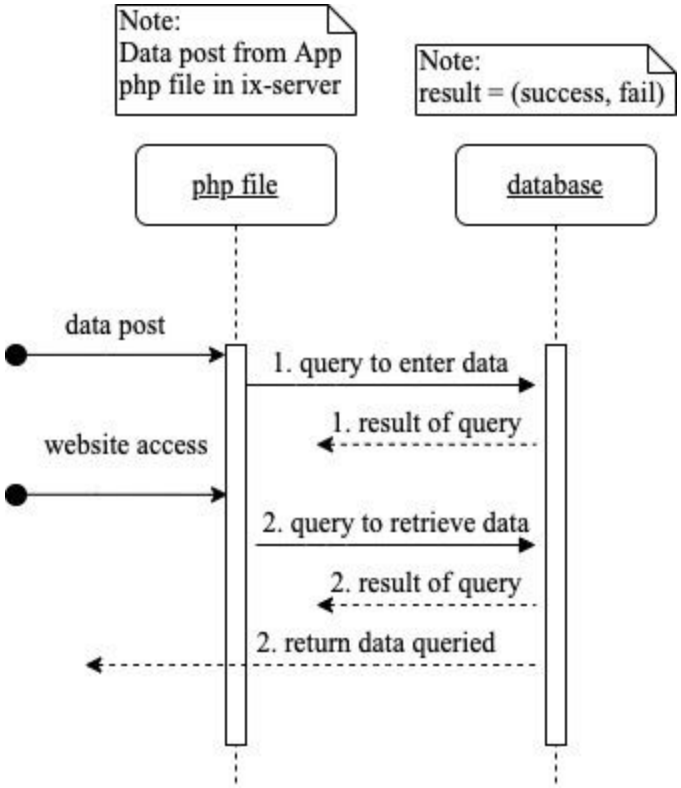
<Figure 5> Sequence of events to create a backup.

D. Every six hours the backup script requests a copy of the current location database. The copy is returned and posted to a private Github repository

E. In order to ensure we never lose more than 12 hours of data we are creating a backup of the database in case the primary database fails

4.3. ix-dev Host Server

A. This module's primary functions are to host the primary database and run the database backup script
B. The ix-dev server is the center of all application interaction. It processes phone application geo input and adds it to the mySQL database. It handles PHP requests from the website to display geolocation data. It interacts with GitHub via a script to upload a backup copy of the database every 12 hours
C. No static diagram was created, as data passes through described in other diagrams



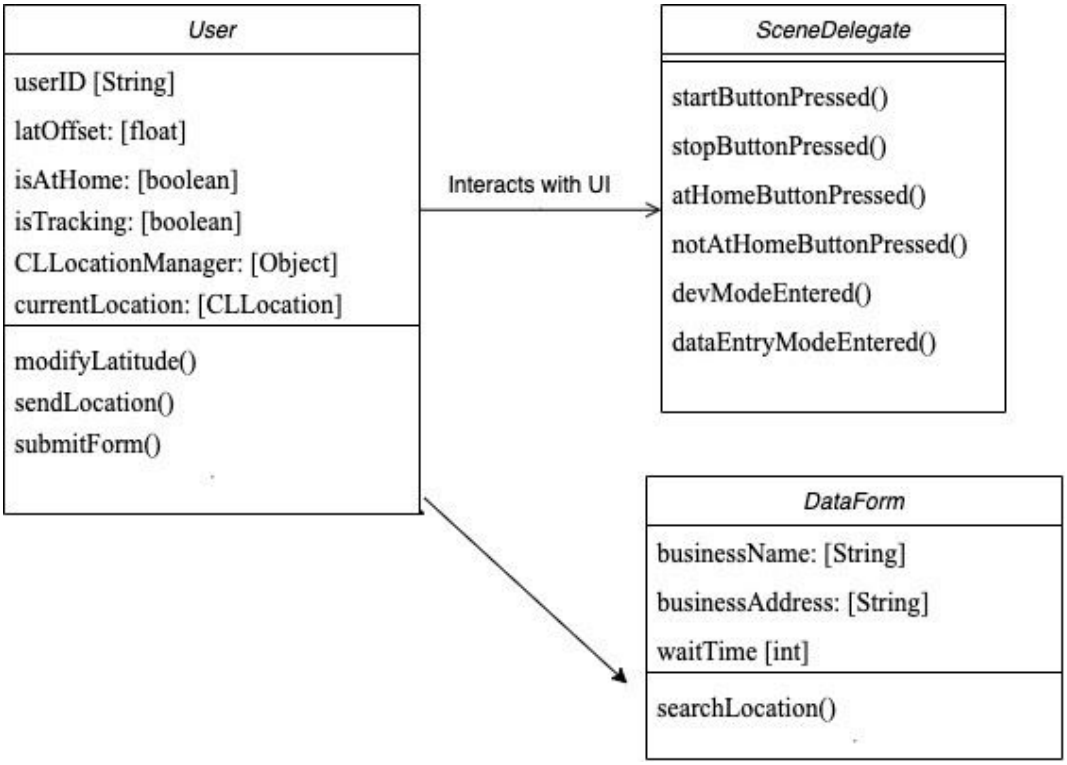
<Figure 6> The ix-dev server hosts interactions between the other components.

D. A data post from the application goes through the ix-dev server to the php file which queries, enters, data into the database. Upon the website being accessed the php file will issue a query which retrieves all currently active users, and will then display the information as markers
E. The ix-dev server was chosen because we can use it for free as UO students. We are also able to host free web pages using /public_html

4.4. Phone Application

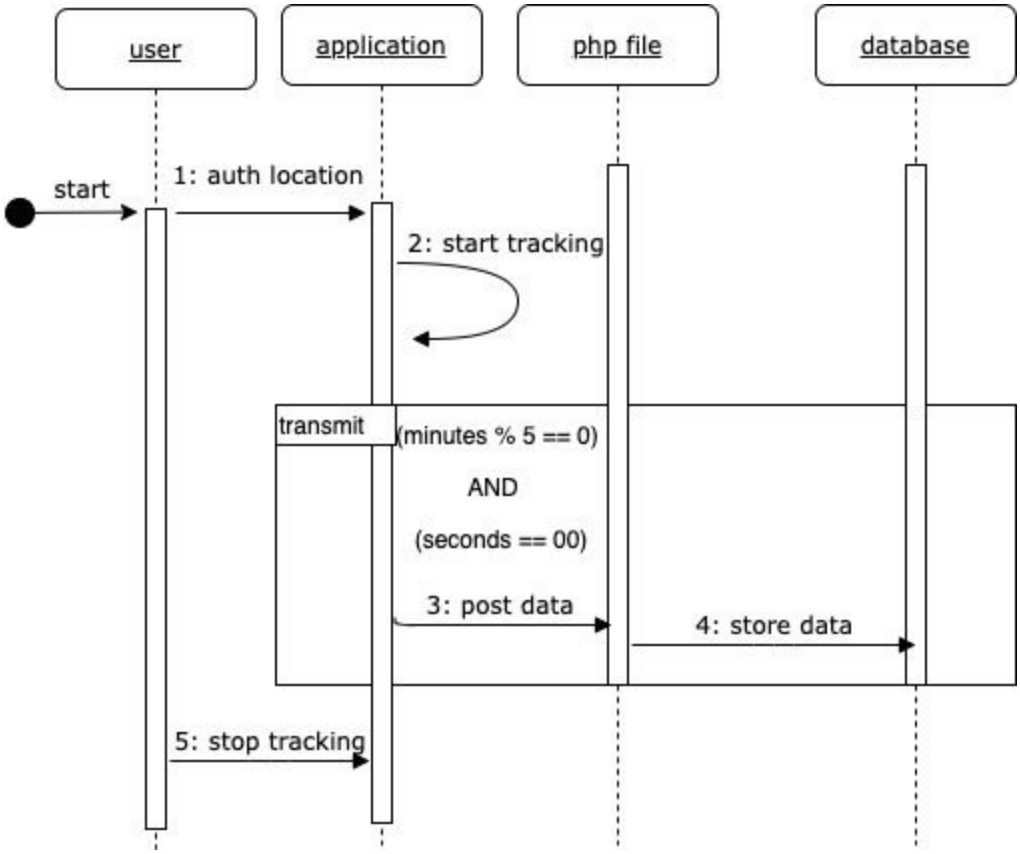
A. Provide a basic user interface and obtain user geo-spatial data every five minutes and upload it to ix-dev server. Allows the user to designate time they are and are not at home to obfuscate the location of users' homes

B. Interfaces with ix-dev server, to upload geo-spatial data for eventual storage in the database



C.

<Figure 7> User device tracks data and interactions the dataform is for wait time implementation.

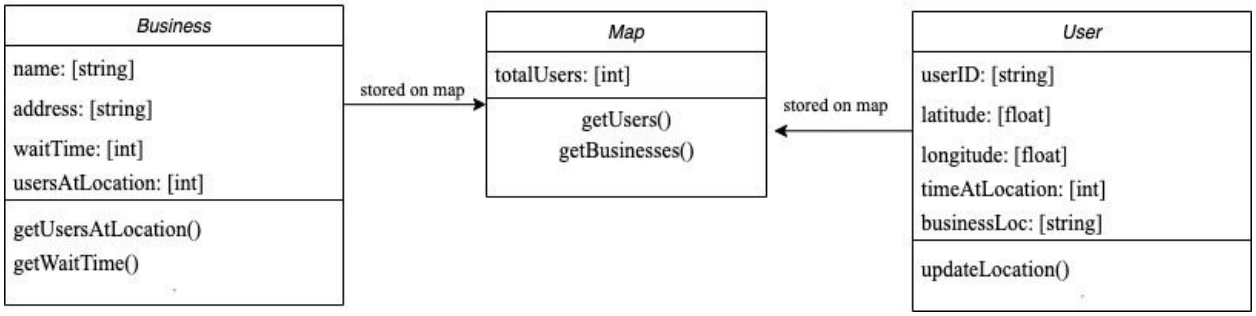


<Figure 8> Displays how the application posts data every 5 minutes.

D. Upon starting the application a user must accept tracking services. Every five minutes on the hour a user's location is sent to be entered into the database. If the user indicates to stop tracking location the loop will be halted
E. The project requires collecting geospatial data from the user for 7 days. The phone is the only device that people take with them when they go places Additionally, it is not possible to get geo-spacial data in the background with a browser. Thus we chose to have the primary UI and data collection done via a phone application

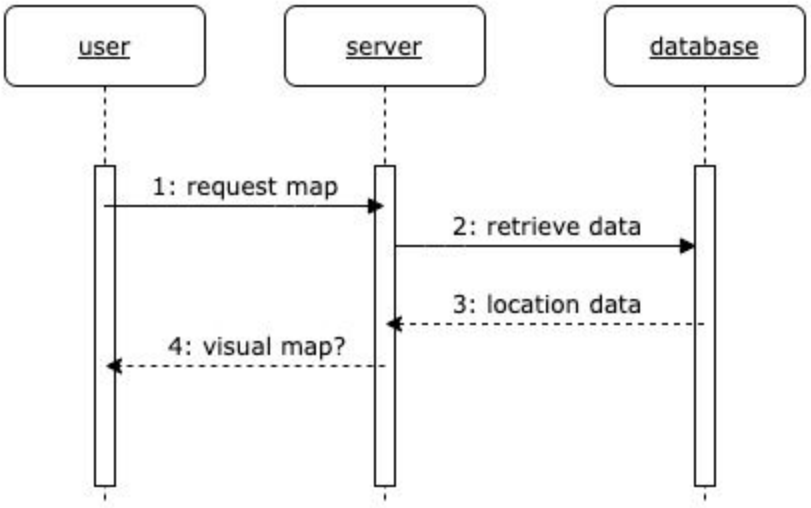
4.5. Visualization Website

A. This is a website to display the data stored in the database
B. It will interface with the ix-dev server to fetch geo-spatial data from the database for display



<Figure 9> Diagram showing marker classes for the website map.

C. Each user location marker has data shown while each business marker is represented by Business. Once a marker is made for each active user and business data in the database they are entered into the map. The map will have a total count of active users.



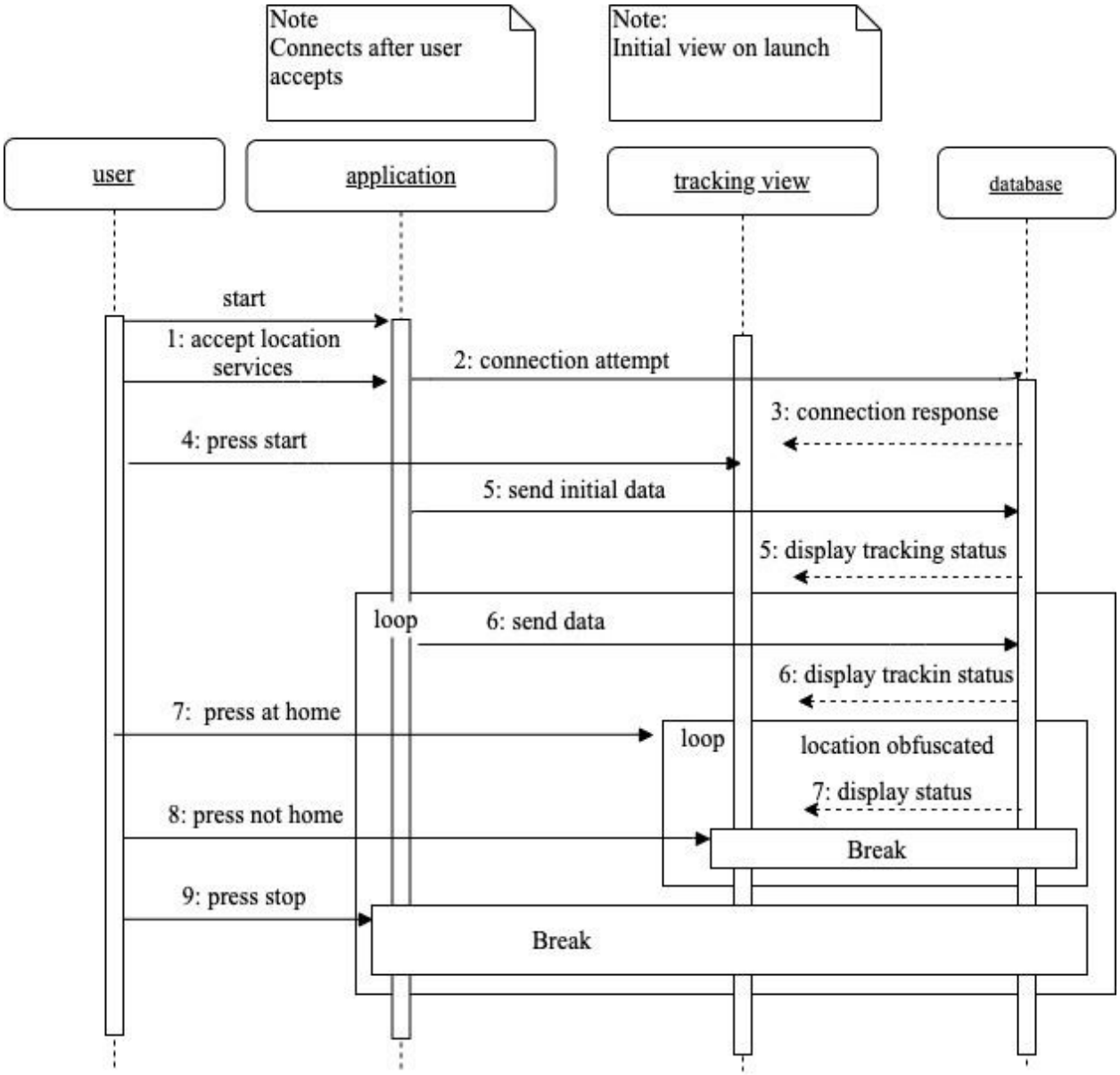
<Figure 10> This sequence of events takes place to display markers on the map.

D. Upon a user visiting the map website, a query to retrieve current location data is issued by the php file on the server. The query returns the location and information which are then entered into map markers which will then be visible to the user.

E. This is used to create a way of displaying data that is easily understood, as opposed to viewing raw latitude and longitude values

5. Dynamic Models of Operational Scenarios (Use Cases)

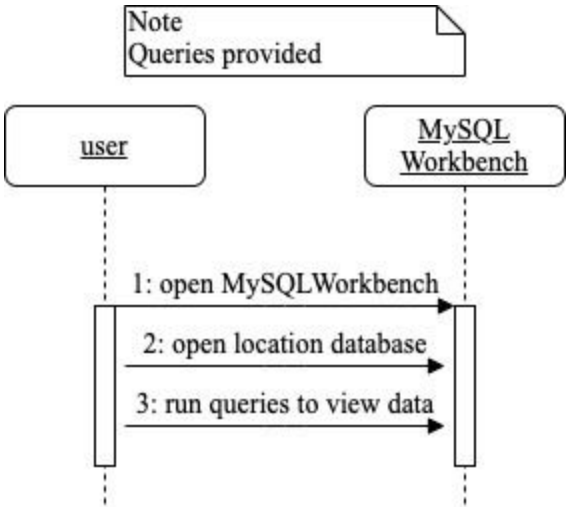
Use Case: Passively storing location data



<Figure 11> Sequence of events that take place when a user wants to start tracking location.

To start tracking location a user will start the application, upon start the user is prompted to accept location services. During this the application attempts a connection to the database, and the response is shown to the user. After a user has enabled location services they will have the option to toggle the start switch, once toggled the initial location of the user is sent. An image will be displayed while location is tracked, and the location will be sent on the scheduled interval. If the user toggles the at home switch the location is altered to maintain privacy. If the at home switch is switched off the location values are no longer changed when storing. If the tracking switch is toggled all application processes are halted, until started again.

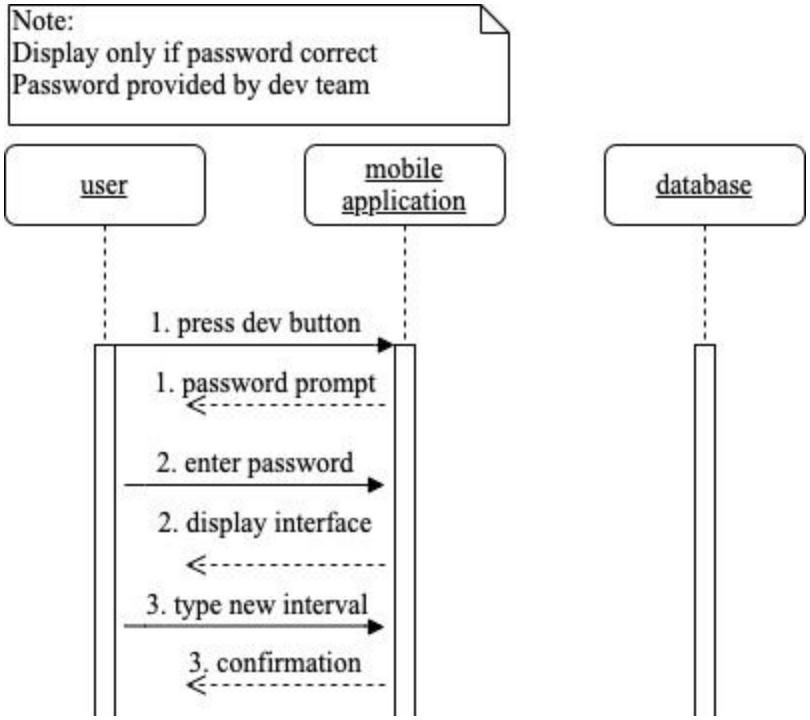
Use Case: View raw location data in the database



<Figure 12> Sequence of events to view raw data.

The user may access the raw location data by accessing the database on MySQLWorkbench.

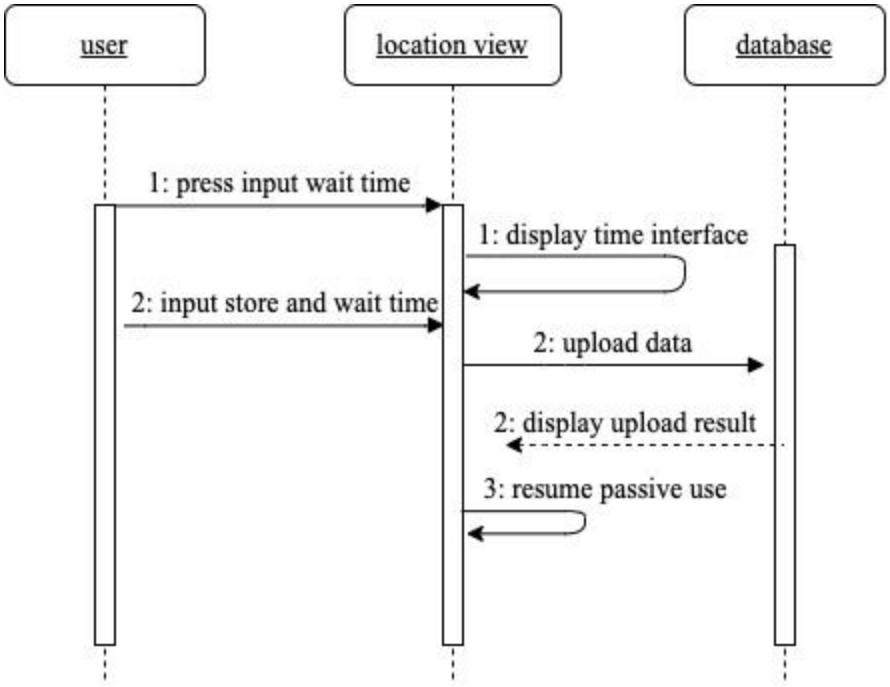
Use Case: Dev mode



<Figure 13> Diagram of events that take place in the dev mode view.

For an administrator or developer to test the application they must access the developer view. The user may press the developer button which will prompt the user for a password, upon entering the correct password provided the application will load a new view. On the new view a user will have the option to enter values to create a data packet and test the transmission. They may also type a new interval into a textbox which will change the frequency of transmission.

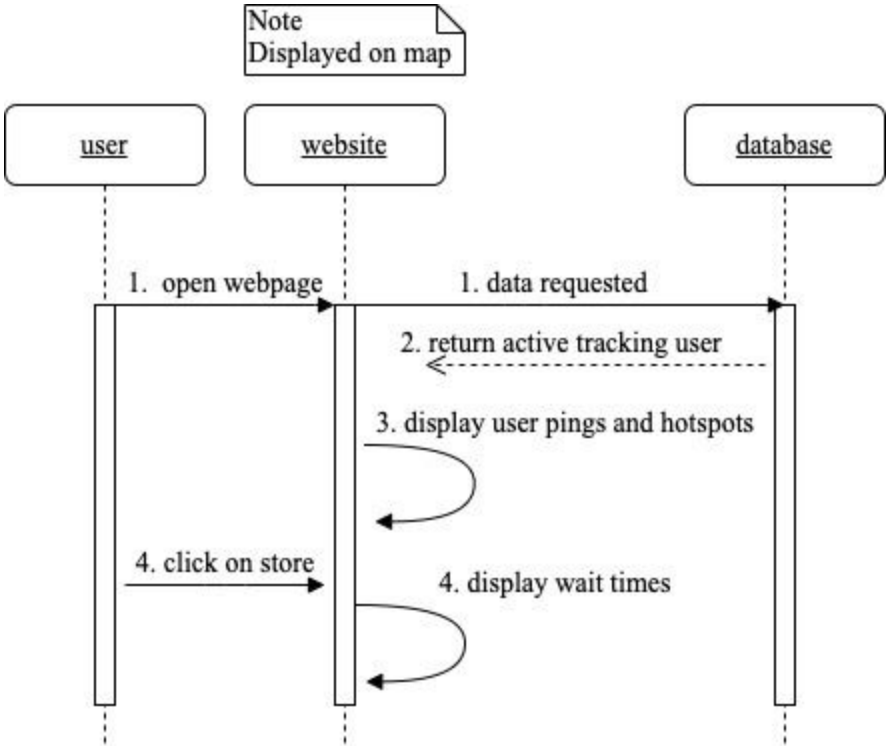
Use Case: Entering Wait Times of Businesses



<Figure 14> Diagram of a user entering information while waiting, temporary until automated.

If a user is standing in line at a store, or waiting for food at a restaurant they have the option to enter line time data. This is conducted by pressing a button to input the information, and a new view being displayed. On this view the user will be able to choose which store they are located at, and then enter the relevant waiting information. The user will then submit the data which will be transmitted to the database, then returning to the location tracking view.

Use Case: View location data on webpage



<Figure 15> A use case diagram detailing how information is displayed to a user upon accessing the site.

Providing information to the users of the application is achieved through the website. If a user directs to the website in their browser, a query is run to retrieve all current user's location and stored information regarding businesses. Each user location is then made into a marker on the map which when clicked will display the information stored.

6. References

Hornof, Anthony. "Notes on Reading Van Vliet (2008) " Apr. 16, 2020
https://classes.cs.uoregon.edu/20S/cis422/Handouts/Lecture_Notes.pdf

Hans van Vliet. 2008. Software Engineering: Principles and Practice (3rd. ed.). Wiley Publishing.

Rosson, M. B., & Carroll, J. M. (2002). Usability Engineering. San Francisco: Morgan Kaufmann.

7. Acknowledgements

https://app.diagrams.net/

This web page was used to create the diagrams in the document.