

Project 1

Software Design Specification

Group 5 - 4-23-2020 - v1.03

Team Members (and initials): Irfan Filipovic (IF), Shane Folden (SF), Man Him Fung (MF),
Mason Jones (MJ), Siqi Wang (SW)

Table of Contents

| | |
|---|-------------|
| 1. SDS Revision History | 1 |
| 2. System Overview | 2 |
| 3. Software Architecture | 2, 3 |
| 4. Software Modules | 3 |
| 4.1. Geo-spatial database | 3 |
| 4.2. Backup database | 3 |
| 4.3. ix-dev server | 4 |
| 4.4. Phone Application | 4 |
| 4.5. Visualization Website | 4 |
| 5. Dynamic Models of Operational Scenarios (Use Cases) | 4 |
| 6. References | 5 |
| 7. Acknowledgements | 5 |

1. SDS Revision History

NOTE:

RED TEXT = DEFAULT TEXT

BLUE TEXT = TBD

GREEN TEXT = PROJECT 2

YELLOW HIGHLIGHTER = NEED REVISION

| Date | Author | Description |
|-----------|----------|--|
| 4-15-2020 | MF/SF/MJ | Developed the SDS skeleton @ noon meeting. |

| | | |
|-----------|-------|---|
| 4-15-2020 | group | Expanded on the SDS skeleton, finalized v1. |
| 4-23-2020 | mj | Revising SDS to reflect progress. |

2. System Overview

Components:

- Database + ix-dev server
 - Database communicates with the website's hidden .php page
 - Outputs tab-delimited text files of the location data using PHP
 - Database is written in MySQL
 - Server will utilize script(s) to backup the database
- Phone application
 - Application communicates with website's .php page, sends user geo-spatial data to the database every 5 minutes
 - Communicates with permissions on users phone to obtain geo-spatial data
 - Shifts location data to assist in anonymizing locations
 - Will only be available on iPhones
 - Will be written in Swift
- Backup database
 - Communicates with primary database/ix-dev server to create a backup every 12 hours
 - Will use a script to copy the database to ensure data retention
 - Backup files will be stored in our private GitHub repository
- Website + .php backend
 - The backend .php page communicates with the iPhone app to pull user location data
 - The frontend .html page communicates with the database to display the gathered geo-spatial data
 - Displays data in a human readable format
 - The goal is to have the geo-spatial data represented on a map
 - Written using HTML, php [and other languages](#)

3. Software Architecture

- **Geo-spatial Database:**
 - We will write our database using MySQL.
 - The geo-spatial database holds all the location data and is connected to each component via the ix-dev server.

- **Backup Database:**
 - We will be regularly backing up our geo-spatial database file. ([.sql/.mwb](#))
 - 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.
 - [We will be using a possible variety of languages on our host server.](#)
 - 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.
- **Phone Application:**
 - We will write our phone application in Swift to 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.
[EXPLAIN HOW THE APP SENDS DATA TO OUR SERVER IN V2](#)
- **Visualization Website and .php Backend:**
 - We will write our website using HTML. We will most likely need to use [JavaScript, PHP, or another language](#) to visually represent our location data.
 - Data will be fetched from the *Geo-spatial Database* to be presented visually.
 - The php backend file will compare the previous location to see if the user shipped location.

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.
- C. [A static model.](#)
- D. [A dynamic model.](#)
- 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.

- C. [A static model.](#)
- D. [A dynamic model.](#)
- 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. [A static model.](#)
- D. [A dynamic model.](#)
- 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. [A static model.](#)
- D. [A dynamic model.](#)
- 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-spatial data in the background with a browser. Because of this, 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.
- C. [A static model.](#)
- D. [A dynamic model.](#)
- 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)

Passive Use

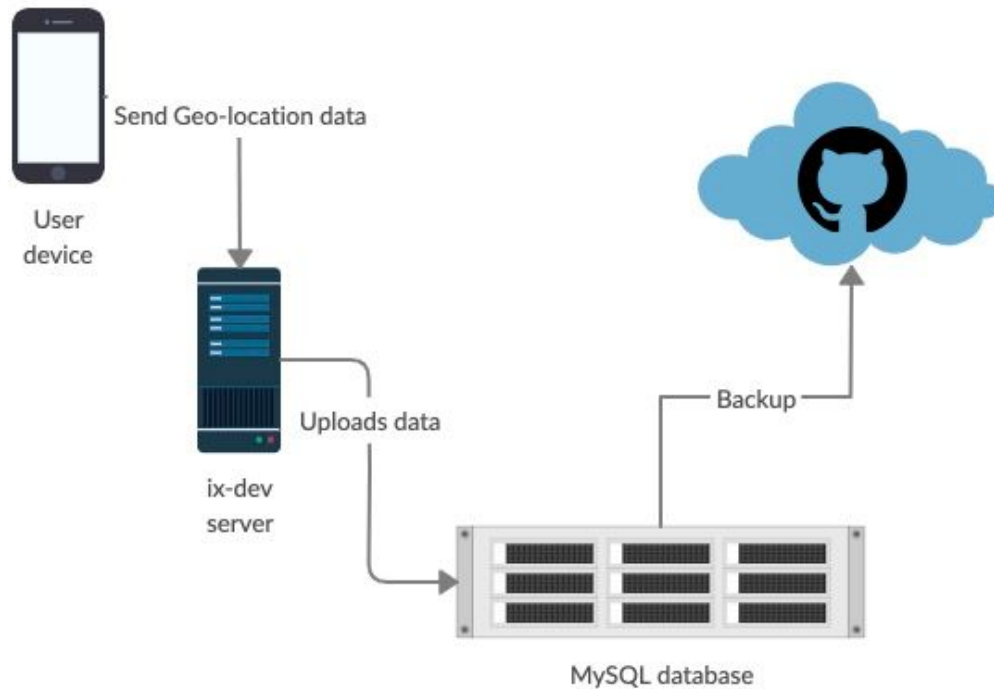


Figure 1: Diagram displaying passive use case of tracking location data.

General Use (Access data)

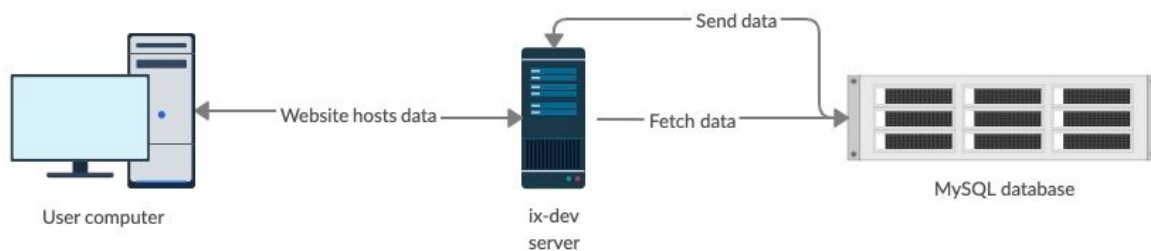


Figure 2: Diagram displaying accessing data via the website.

6. References

https://developer.apple.com/documentation/corelocation/getting_the_user_s_location/using_the_standard_location_service

<https://app.creately.com/diagram/>

https://www.cs.utah.edu/~jamesj/ayb2005/docs/SDS_v2.htm

<https://classes.cs.uoregon.edu/20S/cis422/P1/WWeek.pdf>

7. Acknowledgements

