Project 1
Software Requirements Specification

Group 5 - 4-15-2020 - 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. SRS Revision History

NOTES:

RED TEXT = DEFAULT TEXT

YELLOW HIGHLIGHTER = REVISION NEEDED

GREEN TEXT = PROJECT 2 RELATED

BLUE TEXT = TBD

Date	Author	Description

4-10-2020	group	Gutted default text, revised document to be the skeleton for Sunday's meeting.
4-12-2020	group	Expanded on SRS skeleton. Established color code, fully marked up SRS v1.
4-15-2020	mj	Cleaned up for presentation.

2. The Concept of Operations (ConOps)

The goal of this application is to collect a user's location data, store it in a database, and display a collective of all users' data in a meaningfully visual way. This database will produce tab-delimited text files of the location data. The database will be backed up periodically to ensure minimal data loss.

2.1. Current System or Situation

A new coronavirus, Covid-19, was first noticed to be spreading in November of 2019. By March of 2020, everyday life in the United States, along with the rest of the world, had been dramatically changed. Stay-at-home and social distancing orders led to a significant decrease in social interaction in efforts to minimize human contact. Some scientists estimate that one person infected with the new coronavirus will infect another 3.3 people. Despite these orders many individuals need to access essential services such as grocery stores, which results in long lines and conglomerations of people.

Currently, the best application to assist social distancing measures is Trailcheck. Trailcheck is very similar to our application, but is limited to hiking trails and requires users to manually enter their location data. Google Maps also functions similarly to our application, but the only relevant reports deal with car-related road traffic and peak foot-traffic in stores.
EXPAND IN V2

2.2. Justification for a New System

Trailcheck: This application does not have a large user base, likely because it requires users to actively enter their location data. It focuses solely on hiking trails whereas our application will deal with many popular destinations. These destinations will include grocery stores, pharmacies, hiking trails, public parks, and more.

Google Maps: While it has many useful functions, it's only *situationally relevant* functions are displaying car-traffic by road and peak average foot traffic in stores by time. It does not consider the maximum allowed number of people in a store, line lengths, or average idle/wait times.

Traffic is never considered for non-roads, such as hiking trails or popular stretches of sidewalk (such as one that stretches through the University of Oregon campus).
EXPAND IN V2

2.3. Operational Features of the Proposed System

First, our team will track and store location data with the plan of assisting communities in keeping peak foot traffic low and human interaction to a minimum. A database for the location data may be designed with the function of storing raw data and outputting formatted text files. A website may be created in order to visualize the data on a map, and a system will be implemented to backup our database.

While this won't directly fulfill user needs, this will lay the framework for Project 2. Project 2 will greatly enhance the practicality and usability of the location data.

2.4. User Classes

- *Development Team*
    - *Creates, updates, and maintains the application*
- *Everyday Users*
    - *Visit the grocery store, pharmacy, or other essential businesses*
    - *Pick up food from restaurants*
    - *Go hiking/general outdoor activity*
    - *Fill up at gas stations*
        - *Users may be running the app on the background of their phones to constantly transmit location data*
        - *Users will look at the app to see when the best times are to do the above activities*

2.5. Modes of Operation

- Developer Mode (Unconfirmed)
    - Available to Development team
    - Provide operations to manage the database and collection services
    - Provide ability to change function, representation, or format of data
    - View raw data
- General Use (Interactive)
    - Available to Development team, and Everyday Users

    - Track location data, and upload to database
    - View visual anonymous traffic data
    - Find best times to go to the store, etc.
    - See average restaurant wait times
- Passive Use (Running in background)
    - Available to Development team, and Everyday Users
    - Track location data, and upload to database

2.6. Operational Scenarios (Also Known as “Use Cases”)

Use Case: Passively storing location data

Brief description: This use case describes how a user’s location data will be tracked and stored.

Actors: Any user.

Preconditions:

1.The user has access to the internet.
2.The user has location services enabled on their mobile device.

Steps to Complete the Task:

1.The user runs the application. It can be open or in the background.
2.The user leaves their device powered on.
3.The user keeps their device with them at all times.

Postconditions:

The user has allowed the application to run uninterrupted, has maintained a powered mobile device, and has their device with them at all times.

Use Case: BACKEND View location data

Brief description: This use case describes how a developer views information.

Actors: Development Team Member

Preconditions:

1. The member has access to the internet and is able to connect to ix-dev.
2. The member is on a computer, not a phone, tablet, or another device.

Steps to Complete the Task:

1. The member connects to ix-dev.
2. The member connects to the database.
3. *??? Update once known*

Postconditions:

The member has connected to ix-dev, established a connection to the database, and is able to modify/view data.

Use Case: FRONTEND View location data

Brief description: This use case describes how a user views information.

Actors: Any user

Preconditions:

1. The user has access to the internet.
2. The user is on a computer, a phone, a tablet, or another device.
3. The user has downloaded our application.

Steps to Complete the Task:

1. The user opens the application.
2. ??? The user views data
3. ??? Update once known

Postconditions:

The user has the application installed and operating, and is able to view location data.

3. Specific Requirements

3.1. External Interfaces (Inputs and Outputs)

MUST HAVE:

User Data Collection

1. User Data Collection
2. This is the data that will be stored and used for our application. Every function is based on this data.
3. **Input -** Location data from user cell phones **| Output -** Location data is formatted and stored in our database.
4. User I.D. \tDate\tTime\tLatitude\tLongitude\tTime at Location\n
5. N/A
6. Tab-delimited text

Output Formatted Text Files

1. Output Formatted Text Files
2. This is how raw data can be viewed.
3. **Input -** Location data from database **| Output -** The text file is stored in the project repository.
4. Formatted text from database, output to a ‘.txt’ file.
5. N/A
6. Text file organizing tab-delimited data by User I.D.

SHOULD HAVE:

Backup System

1. Backup Database
2. Creates and stores a copy of all location data. This is to ensure there is a minimal loss of data if the primary database goes down.
3. **Input** - mySQL database file | **Output** - mySQL database file
4. mySQL database file
5. N/A
6. mySQL database → .mwb?

Visual Representation of Data

1. Visual Representation of Data
2. This enable user to view the data easily
3. **Input -** mySQL database file **| Output -** visual represent data, e.g. chart, numbers
4. mySQL database file
5. N/A
6. Map

COULD HAVE:

User Verification

1. User login
2. This is how a user will receive a user-id, and verify access to the id.
3. **Input -** Username and password **| Output -** Token verifying access
4. Input retrieved from user
5. N/A
6. Token string

WON'T HAVE:

Have 1 'won't have' by the end of the project.

3.2. Functions

- **User Data Collection**
  - 1. Validity checks on the inputs.
  - 2. Sequence of operations in processing inputs.
  - 3. Responses to abnormal situations, including error handling and recovery.
  - 4. Relationship of outputs to inputs, including
    - (a) input/output sequences
    - (b) formulas for input to output conversion

- **Output Formatted Text Files**
    - 1. Validity checks on the inputs.
    - 2. Sequence of operations in processing inputs.
    - 3. Responses to abnormal situations, including error handling and recovery.
    - 4. Relationship of outputs to inputs, including
        - (a) input/output sequences
        - (b) formulas for input to output conversion
- **Backup System**
    - 1. Validity checks on the inputs.
    - 2. Sequence of operations in processing inputs.
    - 3. Responses to abnormal situations, including error handling and recovery.
    - 4. Relationship of outputs to inputs, including
        - (a) input/output sequences
        - (b) formulas for input to output conversion
- **Visual Representation of Data**
    - 1. Validity checks on the inputs.
    - 2. Sequence of operations in processing inputs.
    - 3. Responses to abnormal situations, including error handling and recovery.
    - 4. Relationship of outputs to inputs, including
        - (a) input/output sequences
        - (b) formulas for input to output conversion
- **User Verification (if time permits)**

3.3. Usability Requirements

The system will be designed to require low interaction from the user. While operating in the background, the system will need to record the user's geo-location data to the database in order to be effective. During the 5 minute interval between recording data, the system will halt most of its actions to reduce resource utilization.
These requirements ensure user interaction will be minimal while still recording and maintaining location data, yielding the desired outcome.
EXPAND V2

3.4. Performance Requirements

Location data will be fetched and stored every 5 minutes.
The ix-dev server will perform a scheduled backup of the database every 12 hours.
EXPAND V2

3.5. Software System Attributes

Reliability:
The system tracks and records location data from many users. To maintain valid data it is imperative that the system is reliable. To maintain reliability the system will incorporate a backup database and the location data will be stored immediately upon being fetched.

Portability:
To operate effectively on user devices such as phones, it is important that the app can be executed on multiple environments. The system will be designed to operate on mobile devices, but accessing data will be available via a computer as well.

Usability:
The usability of the app will be straightforward, as once the user authorizes location tracking, location will be tracked in the background without user input.

Maintainability:
To preserve portability and maintain a functioning system it is important it be highly maintainable. Organizing functions and components of the system with effective naming conventions and maintaining documentation of system alterations will provide a high level of maintainability.

Safety:
Since the system tracks and saves user location and time, it is important that the data only can be accessed by the development team. Users will have no permission to access or edit the data. Users can only view the visual representation of data. Also the location data should not be recognized.

4. References

https://classes.cs.uoregon.edu/20S/cis422/P1/WWeek.pdf
http://www.trailcheck.info/

Book:
van Vliet, Hans. (2008). *Software Engineering: Principles and Practice*, 3nd edition, John Wiley & Sons.

5. Acknowledgements

All sources used in the creation of the document should be listed here. This template builds slightly on a similar document produced by Stuart Faulk in 2017, and heavily on the publications cited within the document, such as IEEE Std 1362-1998 and ISO/IEC/IEEE Intl Std 29148:2018.

Possible Minor Future Revision

The ConOps portion of this template was written based on IEEE Std 1362-1998 before realizing that this standard has been subsumed by IEC/IEEE Intl Std 29148:2018. A future exercise could, in short, determine if the references to the former could be replaced with references to the latter; that is, if the ConOps description in each of the two documents is sufficiently identical.