Quarantine Overseen
Software Requirements Specification

Y. Gao (yg), J. Johnson (jj), M. El Jamal (mj), N. Rudolph (nr), T. Wong (tw) - 5-31-2020 - v2.0

Table of Contents

1. SRS Revision History

Date	Author	Description
5-11-2020	yg, mj, jj	Created the initial document outline
5-15-2020	yg, mj, jj	Added seconds and flushed out document
5-31-2020	yg, mj, jj	Revision for Final Project

2. The Concept of Operations (ConOps)

2.1. Current System or Situation

The year 2020 started with a series of unfortunate events, where notably the coronavirus (COVID-19) pandemic ravaged the globe. The coronavirus' nature of being contagious, even during long incubation periods, has made it spread throughout communities easily. Inevitably, this led to countries across the globe requiring quarantine procedures and "social distancing" measures to be taken in place. This social distancing is something that many, if not all, of us are not used to. Systems are in place by businesses to promote this social distancing, but it is hard to maintain it in everyday life.

2.2. Justification for a New System

This system is intended to be a COVID-19 contact tracing system. As of the time of writing this, there is no streamlined, real-time system in place to detect if someone has come into contact with someone who either has the virus or is at high-risk of having the virus that respects individuals privacy and anonymity. This system can help to warn a user if they come into contact with someone who is infected, and give suggestions such as self-quarantine for a set incubation period.

2.3. Operational Features of the Proposed System

2.3.1. Data Collection and Use

The system aims to collect geo-spatial and temporal data from users and compare that with other users to see if they have come within a set distance of an individual who has flagged themselves as *Infected* or with an individual who is marked by the system as *High Risk.*

2.3.2. Volunteering Information

To be marked as *Infected* a user would have to volunteer the information to the system; however, to be marked as *High Risk* would be processed automatically by the system.

2.4. User Classes

2.4.1. System Administrators, Programmers and Instructors

This class of users will be able to deploy the system and view database information on a UNIX environment. The system will require this user class to possess basic knowledge of UNIX environments,web-server management and MySQL commands. Shell scripts will be provided to aid this class of users with these processes.

2.4.2. Students. General Population

This class of users will be able to give their location - through the acceptance of html tracking - and view if they have come into the vicinity of another user that could put them at *High Risk* of being infected. This class of user will need to have access to internet services and have a device that can provide GPS location. Also, they will need basic internet browser skills.

2.5. Modes of Operation

2.5.1. Developer Mode

- Developers have partial access to the system
    - Requirements: Access to server that system is on
    - Have ability to operate MySQL server and manage database
    - Monitor Data in the server

2.5.2. General User Mode

- Users only have the access to view the Web Page
    - Users can view their user ID, location, infection risk status and mark themselves as *infected*

2.6. Operational Scenarios

2.6.1. Provide Location

Brief description: This use case describes how a user provides location information
Actors*:* A general user
Preconditions*:*

1. System is active
2. Users have access to internet and web-portal

Steps to Complete the Task:

1. Open web-portal
2. Allow GPS tracking through web browser prompt

Postconditions*:*

1. The user can see geographical location as Longitude and Latitude in the Middle of web page | refresh every ten-seconds automatically
2. The User also can see their unique identification
3. The user can see infection status:
    a. High Risk of Infection | Low Risk of Infection | Infected
    b. Default Status is Low Risk of Infection

2.6.2. Check if at *High Risk*

Brief description: This use case describes how a user would check if they are at *High Risk* of contracting the COVID-19 virus
Actors*:* A general user
Preconditions*:*

1. System is active
2. Users have access to Internet and web-portal
3. Users have provided location for the system to track

Steps to Complete the Task:

1. Open web-portal

Postconditions*:*

1. The user can see geographical location | user ID
2. The user can see if they are at *High Risk* of infection
    a. System will automatically check all other users around to evaluate the risk level of infection and give the Result on top of the webpage

2.6.3. Screen Sleep Enable/Disable

Brief description: This use case describes how a user would check if they are at *High Risk* of contracting the COVID-19 virus
Actors*:* A general user
Preconditions*:*

1. System is active
2. Users have access to Internet and web-portal
3. Users have provided location for the system to track

Steps to Complete the Task:

1. Open web-portal
2. Click the Button “Screen sleep is enabled” | “Screen sleep is diabled” to enable or disable screen sleep

Postconditions*:*

1. Screen sleep disabled will turn the web page to a light blue background, enable will turn back to normal User default background
2. The user can see geographical location | user ID
3. The user can see if they are at *High Risk* of infection

2.6.4. Check Infected People

Brief description: This use case describes how a user would check if they want to see all other infected Users ID with the COVID-19 virus
Actors*:* A general user
Preconditions*:*

1. System is active
2. Users have access to web-portal
3. Users have provided location for the system to track

Steps to Complete the Task:

1. Open web-portal

2. Click button “Infected” on the top left It will direct user to the Infected page

***Postconditions**:*

1. The user can see all Infected User IDs in the middle of web page

2.6.5. Mark self as *Infected/Not Infected*

Brief description: This use case describes how a user report if they are *Infected/Not Infected* with the COVID-19 virus

***Actors**:* A general user

***Preconditions**:*

1. System is active
2. Users have access to web-portal
3. Users have provided location for the system to track

Steps to Complete the Task:

1. Open web-portal
2. If User is in the Home Page, Click button “Infected” on the top left It will direct user to the Infected page else go to next step
3. Click button “Report Yourself as Infected”| “Report Yourself as not Infected” in the middle of the web page depending on what the user wants to report as.

***Postconditions**:*

1. The web alert message will pop out indicate user report self in as infected/not infected
2. If user already report self infected/not infected, the web alert message will pop out indicate user already reported in to the system
3. The user can see all Infected User IDs in the middle of web page

2.6.6. Set Up System

Brief description: This use case describes how a user would set-up the system on a device

***Actors**:* A programmer, teacher or administrator

***Preconditions**:*

1. User has access to set up files
2. User has access to a UNIX system
3. UNIX system has internet access

Steps to Complete the Task:

1. Run set-up script
2. Fill in information as prompted (such as credential set-up)
3. Include info in programmer documentation for manual changes if errors

***Postconditions**:*

1. System is active

2.6.7. Check Database Information

Brief description: This use case describes how a user would access database information
Actors*:* A programmer, teacher or administrator
Preconditions*:*

1. System is active
2. User has access to internet

Steps to Complete the Task:

1. Head to database view URLs
2. Fill in credentials as prompted (if added by programmer)

Postconditions*:*

1. Database information is displayed for user

2.6.8. Take Down System

Brief description: This use case describes how a user would terminate the system
Actors*:* A programmer, teacher or administrator
Preconditions*:*

1. System is active
2. User has access to system host location
3. User has credentials to access active system

Steps to Complete the Task:

1. Stop the MySQL server
2. Remove files from public_html

Postconditions*:*

1. System is taken down

3. Specific Requirements

3.1. External Interfaces (Inputs and Outputs)

3.1.1. Absolutely Required

Functional Requirements

- User Data Collection
  - All Data Below are necessary to keep our System Work Correctly
  - Inputs
    - User ID (String) | Generate By User's IP address
    - Date (mm/dd/yyyy) | Current Date
    - Current Time (00:00:00 to 23:59:59) | Current Time
    - Latitude (x.xxxxxx) | User Current Latitude
    - Longitude (x.xxxxxx) | User Current Longitude
    - Infection Status (boolean) | Default Status is Low Risk of Infection
  - OutPuts

- User ID (String) | Generate By User's IP address
- Date (mm/dd/yyyy) | Current Date
- Current Time (00:00:00 to 23:59:59) | Current Time
- Latitude (x.xxxxxx) | User Current Latitude
- Longitude (x.xxxxxx) | User Current Longitude
- Infection Status (boolean) | Status will Change By User Activity | Get Too Close To Infections | Or High Risk Users

- Backup System
    - Backup MySQL DataBase
    - Prevent Permanent Data Losses | Server Went Down
    - Take Data Sets from MySQL DataBase and Reformat it to CSV file and can be download to Localhost

Non Functional Requirements

- **Hardware**
    - A mobile phone can have internet access with a Web Browser Installed
- **Software performance**
    - The System should be functional to track user and evaluate user risk level of infection for user
    - The System should give accurate user location up to 0.000001 in both latitude and longitude | refreshing every 10 seconds
- **Output files**
    - The system has no output files, just ability to view data

3.1.2. Not Absolutely Required

- User Interface
    - A PHP Web Page Visual User status location and Risk of Infection
    - A Homepage for Current User Status
    - A Infected Page:
        - Can report self as infected/not infected
        - Show all other User IDs who are infected
        - High Risk of Infection User after 20 days will change to low risk status automatically (can change in hard code if needed)

3.2. Functions

3.2.1. User Data Collection

- **Inputs and Validity**
    - Every User Has A Unique ID, Location Will Be Stored as Latitude and Longitude
    - Four Types of User Tables:

            - Infected | Contact (between all users) | High Risk of Infection | General Tracking (all users)
            - User can report in Infection Case
        - Invalid inputs will display error messages
    - **Inputs Process**
        - Ten-Seconds Interval For User Location
        - Gets distance between users in infected table and tracked user
            - Warn user of infection risk
            - Moves user to high risk table if contact is made
        - Insert all users datasets to the MySQL database
    - **Error Handling and Recovery**
        - Any critical error will cause the webpage to refresh, fixing functionality and displaying an error message
        - Minor errors will display an error message
    - **Inputs and Outputs**
        - Inputs:
            - User Location (Need Permission by User | Gather every 10 seconds)
            - User Unique ID (Auto Generated By The System by IP address)
            - Self-reported infected users
        - Outputs:
            - Data will be sent to MySQL database

3.2.2. Backup System

- **Inputs and Validity**
    - Connection status of MySQL status
- **Inputs Process**
    - Data is backed up from MySQL server by shell scripts every 6 hours
- **Error Handling and Recovery**
    - Data can be stored in Local to prevent permanent loss All Data
- **Inputs and Outputs**
    - Input: DB_HOST|DB_CHARSET|DB_NAME|DB_USER,|DB_PASSWORD
    - Output
        - Backup in .sql format every 6 hours

3.2.3. User Interface

- **Inputs and Validity**
    - User Location | Status | Unique ID
- **Inputs Process**
    - Get User Location (Latitude and Longitude) by permission
    - Get IP address to make a Unique ID for that IP
    - Default User Statue is Low Risk of Infection
    - User can report self in as infected
- **Error Handling and Recovery**
    - Website crashed will auto refresh the webpage
- **Inputs and Outputs**
    - Input: User Location (Latitude and Longitude) | IP address
    - Output:
        - User Current Location | ID | Status will be shown in User Home webpage (Data will refresh every 10 seconds)
        - Screen sleep can be disabled or enabled by click of sleep toggle button

- Infected page will be shown with infected user IDs and infected self report button

3.3. Usability Requirements

1. Track users' data at ten-seconds intervals
2. Automatic backup every 6 hours
3. Calculate distances between users at ten-seconds intervals
4. Web-portal is light and portable
5. Web-portal requires no downloads or installs
6. Web-portal works with most hardware interfaces that have internet access
7. System can record data from multiple operating systems
8. Deployment requires no installs (if the host has MySQL) of third-party services, software and/or APIs

3.4. Performance Requirements

1. Location data will be properly collected handled and display in User view updating every ten seconds
2. The system will backup the MySQL Database every 6 hours to local host in .sql format
3. The .sql output file contain all of the recorded Data
4. Website should be able to run continuously without crashes

3.5. Software System Attributes

3.5.1. Construction

1. The system uses minimal interfaces/subsystems to avoid overcomplication creating issues with the system.

3.5.2. Deployment

1. The system uses scripts to aid the programmer in the deployment of the system
2. Systems requires no third-party software/APIs to keep deployment simple

3.5.3. Testability

1. Automated interactions allow testing of the program to be completed in fashions (such as contact testing done between two devices)

3.5.4. Usability

1. All functionalities of the web page are straight forward
    a. System will auto generate a ID for the user
    b. Only on first visit will a user be prompted for location permissions
    c. Disable screen sleep functionality by click of a single button in the Home page
    d. Report self in as Infected or Not Infected in the Infected Page

4. References

IEEE Std 1362-1998 (R2007). (2007). IEEE Guide for Information Technology–System Definition–Concept of Operations (ConOps) Document. https://ieeexplore.ieee.org/document/ 761853

Faulk, Stuart. (2013). Understanding Software Requirements. https://projects.cecs.pdx.edu/attachments/download/904/Faulk_SoftwareRequirements_v4.pdf

Hornof, Anthony. (2019). Software Requirements Specification [Template]. https://classes.cs.uoregon.edu/20S/cis422/Handouts/Template-SRS.pdf