Software Design Specification

for

*Contact Tracer Network*

Prepared by Natch Sadindum – ns174

SoftFlux

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

2. Additional Requirements 2

2.1 Features List 2

2.2 Extra Specifications 3

3. Software Architecture 3

3.1 Overview 3

4. Detailed Design 5

4.1 Database 5

4.2 Database UI 6

4.3 Tablet UI 8

4.4 Data Correction Subunit 10

4.5 Tablet Disconnection Notification 11

5. User Interface Design 11

5.1 Tablet UI 12

5.2 Staff UI 14

6. Nonfunctional Requirements 16

6.1 Language Requirements 16

6.2 Query Speed 16

7. Conclusion 16

7.1 Closing Notes 16

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Natch Sadindum | 24/4/2020 | Initial Version | 1.0 |

NOTE – I unfortunately didn’t commit enough time to this assignment so it is incomplete. I have done as much as I can in the time I had.

# Introduction

## Purpose

This document is a software design specification document for the Contract Tracer Network project. It will provide specific information on program architecture and UI design for the Contract Tracer Network project, as well as specific information on all components of this project.

## Product Scope

This document is to be used by all departments of SoftFlux related to the development of the Contract Tracer Network project, as a design definition for the implementation of the Contract Tracer Network project.

# Requirements

## Features List

**User/Role management** - System admins can manage users and add/remove them to roles. Roles are Logged-In user, NZPRU Admin, NZPRU Installer and NZPRU personnel.

**Tablet management**

- Tablets can be added/deleted in the system; they will be identified by mac address.

**Venue management**

- Venues can be added/deleted in the system

**Tablet/Venue registration management**

- One or more tablets can be (un)registered with a specific venue. A tablet can be associated with at most one venue but one venue might have more than one tablets.

**User & Tablet authentication**

- Remote users and tablets must be able to log in/out to the system.

**Log record inputting, transmission and storage**

- A UI on the tablet will enable users to insert their record log on arrival, the time of exit and submit it to the NZPRU servers where it will be stored.

**Periodic extraction of normalized data from record log into relational database**

- At a configurable period of time, the system will scan the record log, extract information, perform automatic corrections and insert the data to the relational db.

**Relational database querying and suspected infected visitors’ notification**

- The system will calculate a list of visitors that might have been infected. NZPRU personnel will be providing a list of users as infected and a desired contact depth. For contact depth 1, the system will list all visitors in contact with the designated infected population; for contact depth 2, the system will list all depth 1 visitors but also, all the visitors in contact with them, and so on. NZPRU personnel will be able to automatically notify via email all listed visitors in the end.

**Tablet-not-responding notifications**

The system will monitor the liveness of the tablets and will issue an automated email alert to all installers if a tablet does not respond after a certain amount of time.

## IBM Knowledge CenterExtra Specifications

**Transport Layer Security (TLS)**

Due to this application taking private data from large numbers of users and sending them to an offsite database instead of locally, we must use secured channels for moving this data, requiring encryption to prevent the data from being compromised. TLS is a tool we will use to achieve this. It is somewhat slow, but the response times required for transporting data do not need to be particularly fast, nor is the volume of data per unit time particularly high.

In our case, we will use TLS to achieve secure data transfer through the network between our tablets and database.

# Software Architecture

## Overview and Discussion

The <name> is intended to be scalable, as it is intended to be used at many locations for the purpose of contact tracing and tracking. We intend to design a simple, scalable system, where any future changes would only require adding new data types to the system components.

The application will be written in Java due to its high compatibility with a large range of devices. As a result, all other components of this application will be Java based, and the tablets used will be those capable of running Java applications.

Due to the nature of MySQL workshop (discussed later), plugins and scripts will be written in Python to enhance the usability of the GUI.

This application is divided into two main parts:

The database, which contains all data logged by this application, such as user data, venues and tablets, as well as all record logs.

The user client, which provides a means for both normal users and staff members to use the software for its intended purposes.

The user client is further divided into a staff client (accessible offsite on separate machines) and the public client, which is located on the tablets. The staff client will provide a linkage to the database and allows a large variety of operations on the database in a secure manner, as well as other management capability listed in the features.

The public client is simpler, only allowing the user to log in and out, while sending a record log of user data and a record log of their data and login/logout time + venue to the database.

The system is intended to be run on a dedicated server. This server will run both staff client functions in parallel with constant background functions such as data backup and checking and other periodic automatic functions, such as tablet disconnect checking.

Our database software of choice will be MySQL. It is a popular open-source database software and will be located on the server computer, storing its contents offline. It is simple to install and deploy and has sufficient capability on a single server for nationwide use in New Zealand. Should server performance become sluggish, it is relatively simple to scale out our MySQL server to additional servers if required.

MySQL also has a range of useful tools that we could use, such as the capability for cloud deployment (not useful in this application due to privacy requirements), a large range of available graphical user interfaces for efficient and easy control over database elements and account access management built into a convenient GUI.

Furthermore, MySQL directly supports the use of TLS (transport layer security) for secure transmission of data between the user client, the database, and the GUI. It is intended for tablets to only open communication with the server and database when logs are uploaded, tablets are being checked for connection issues periodically and for remote access and control.

Another useful aspect we can use is the ability to create custom plugins and scripts for the GUI using Python. We can make use of this feature to automatically correct misspelled log records

Overall, MySQL and its closely related tools provides all the functionality we require.

**Application Structure Diagram**

Tablets

Staff Computer Access

Features

User/Role Management

Tablet Management

Tablet – Venue Management

Venue Management

Staff Client

Record Log Auto-upload

Login/Logout Functionality

Public Client

User Client

Network

Tablet Disconnect Notification

Server-side functions

Server

Log – all record logs

User Data

Venue Data

Tablet Data

Offsite Database

# Detailed Design

## Database

|  |  |
| --- | --- |
| Purpose | Contains all data relating to the network |
| Processes | RemoveDuplicateUser:  Inbuilt function allows staff to check and remove duplicate users, created by logging into multiple venues. |
| Dependencies | An admin must set up the database initially so that all data is stored appropriately. |
| Resources | The database will be using the MySQL software as a base and using the MySQL workshop GUI for ease of access |
| Data | Tablet  *linkedVenue*  *macAddress*  *alert*  User  *firstName*  *lastName*  *mobileNum*  *emailAddress*  *homeAddress*  *status* (0 for normal, 1 for infected)  *log* (contains many log records associated with the user)  Venue  *venueName*  *contactNum*  *contactName*  *address*  *email*  Log  *firstName*  *lastName*  *mobileNum*  *emailAddress*  *homeAddress*  *loginTime*  *logoutTime* |
| Notes |  |

## Database UI

|  |  |
| --- | --- |
| Purpose | Allows authorised users access to the database to make queries or modify data.  This UI should allow for user/role management, tablet management, venue management, tablet/venue linkage, user authentication |
| Processes | Log-in Screen:  Inbuilt into MySQL workshop, a staff user may log in using the MySQL login screen using a predefined username and password.  User Role management:  Inbuilt into MySQL workshop, only administrator users may create and delete users, as well as changing their permissions, allowing/denying access to any functions within the database.  Admins may create and delete new staff users and set their username and password  Tablet Management:  An admin and an installer user are able to modify the database properties of a stored tablet object.  Venue Management:  Administrators, installers and personnel are able to use the GUI to insert new venue entries into the database, and are able to modify them. Additionally they may also link tablets and venues together.  Contact |
| Interfaces | Modification of any database material is done here, via the MySQL workshop GUI. |
| Dependencies | An initial administrator must create all other users and set their privileges before other users can use this system. |
| Resources | The database will be using the MySQL software as a base and using the MySQL workshop GUI for user control, database management and queries. |
| Data | Uses the same data found in the database |
| Notes | Note the entity relationship diagram below for a clearer view on the organisation of the data. |

**Database Entity-Relationship Diagram**

Has a

Contains many

Log

Log Record

takes data from

uses

User

Note that a venue may have multiple linked tablets

Venue

has

Tablet

## Tablet UI

|  |  |
| --- | --- |
| Purpose | Allows a user to sign in and out using a simple interface, while recording the users details and creating a log of their stay in a venue. |
| Processes | A general user experience diagram can be found in section 3.  Log-in Screen:  Pressing login button brings user to screen containing fields for the 5 data fields listed in the data section. User must fill in these fields and press *sign in*, which will create a log record using the 5 data fields and the current date and time. This will also return the user to an acknowledgement page and then back to the initial screen after a short period of time.  The fields for phone number and email must be validated via regular expression code to make sure they are valid, and messages can successfully be sent.  Log record is stored across all tablets with the same venue. All tablets must be synchonised so that a logged in user can be shown on all same-venue tablets.  Log Out Function  Pressing Logout button brings up a screen displaying a list of names of people currently signed into this venue. User may select their name and press logout. The current date/time is recorded and added to the log record associated with the now logged out user. The log is sent to the database and stored.  All same-venue tablets must be synchronised again.  A button on the bottom right of the initial screen opens accessibility/language options. A user can select different languages to switch the language used in the UI. A text to speech option is available but only in English language setting.  The pressing of each control should be accompanied by a sound effect as feedback for the user. |
| Interfaces | The tablet will make full use of touchscreen functionality for interaction with the user. |
| Dependencies | Java language localiser function allows all strings on the UI to be switched to different languages based on language localiser files stored locally on each tablet.  Java synthesiser allows for text to speech functionality using pre-set strings as a key for translation. Requires extra sound files/voices for different languages, will only use English language text to speech. |
| Resources |  |
| Data | *firstName*  *lastName*  *mobileNum*  *emailAddress*  *homeAddress*  *loginTime*  *logoutTime* |
| Notes | The reason for not providing text-to-speech in other languages is that we do not foresee the requirement for both text-to-speech and another language occurring in regularity. It is also likely a person with absolutely no English capability is will probably be accompanied by another who is English capable. |

## Data Correction Subunit

|  |  |
| --- | --- |
| Purpose | Extracts data from log records, makes corrections based on already stored user information and inserts data into the database. |
| Processes | Extract 7 pieces of data listed in data section from record log objects sent from the tablet. A record log is sent to the database every time a user logs in or out.  Check database for user data matching that of the log. If no user exists with any matching data, make no corrections and insert into the database.  If matching user data is found, check for differences in each of the fields.  Critical unique data is mobile phone number, as it is unique between users.  Normalise and correct data if mobile number matches an existing user, then insert to database underneath the log belonging to the user. |
| Dependencies | Supervised by personnel. Dependent on existing data in the database being correct or existing. Without data, a user and log will be added. If the first instance of a user and log has errors, then the user will permanently have an error in name or other fields. |
| Resources | Requires secure connection between the database and the tablet sending the data. Will use TLS to provide a secure connection |
| Interactions | Takes data from the tablet sent by secure connection and interacts with the database, inserting extracted data after corrections if needed. |
| Data | *firstName*  *lastName*  *mobileNum*  *emailAddress*  *homeAddress*  *loginTime*  *logoutTime*  *venue* |

## Tablet Disconnection Notification

|  |  |
| --- | --- |
| Purpose | Checks each tablet if they are connected to the system. |
| Processes | Check:  Every 30 minutes, the server will need to “ping” each tablet, which should send a response back if it is connected to the network.  If no response received from a tablet, the server needs to try up to 3 times total to contact the tablet. If all contact attempts fail, then the tablet should send a request to the database to change its alert status to true.  A tablet which has a true alert status will create a notification sent to the mobile phone and email of all installers, alerting them to the connection failure.  Installers or administrators may then manually reset the status of the target tablet once connection has been restored, or the system will check again in 30 minutes, automatically changing the status to normal if required. |
| Dependencies | Network based application that runs in the background on the server. |
| Resources | Can use TLS for secure connection, but because no valuable data is being sent, it is possible to use faster or simpler forms of connection instead. |
| Interactions | Interacts with the tablet to request its status, also interacts with the database directly as it can change the status of the tablet. Also interacts with staff users (installers) to alert them. |
| Data | Modifies the status property of a tablet in the database. Can be reset by any staff user. |
| Notes | It is expected that venues that have the tablets installed should have a stable internet connection present, so the cycle period of 15 minutes per tablet is reasonable. It is expected that most disconnect notification |

# User Interface Design and Discussion

## Tablet UI

The user interface design on the tablet side must fulfil several goals:

- Must be easy to use

- Must be streamlined and convenient

- Must be useable in different languages and have a degree of disability access.

We expect a large variety of people to use the UI, and thus a large variety of skills in device use. The UI needs to allow the user to sign in with personal details and sign out with minimal difficulty.

To log in, the user simply needs to press a large button on the initial screen, which will take them to a menu screen. 5 fields are open for the user to fill in, with clear instructions on the top of the screen.

A help button should be situated on the screen to help answer the users’ questions. Pressing the help button should bring up frequently asked questions and answers.

For example, children may not have an email or mobile phone number. This means they cannot be entered into the system. The answer is that a user can sign in for others as a “bubble”, meaning that their user represents other close family members in addition to themselves.

If the user does not fill out all the fields, an error message will show, asking the user to fill out the fields.

The email and phone number fields will be validated in code to make sure messages can reach those locations. If all the fields are filled appropriately, then the screen will switch to an acknowledgement screen with a message showing that they have successfully signed in, also reminding them to sign out before they leave the premises.

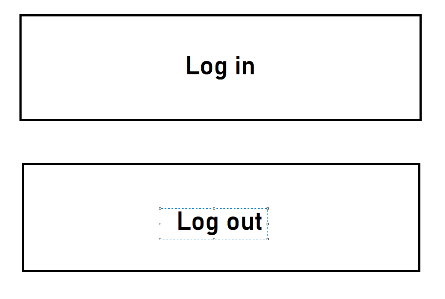
Feedback is very important to the user, especially with such a simple UI design. Basic sound effects should be played quietly every time the user presses a control/button, which should make it clear to the user that the controls are working.

When the user is to leave the premises, they need to sign out of the system. The initial menu contains a large log out button which will bring them to a scroll menu containing a list of names of those currently signed into the venue. Selecting a name highlights it and then displays a mobile phone number only when selected. This also enables another sign out button, when pressed, will prompt the user with a confirmation message, in case they chose the wrong name. Once confirmed, the screen will change to a farewell message to confirm that they are signed out.

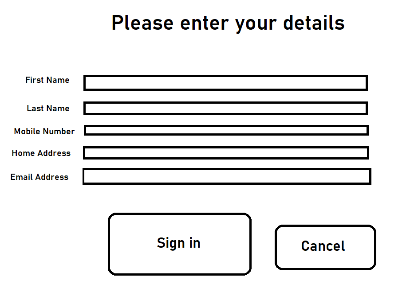
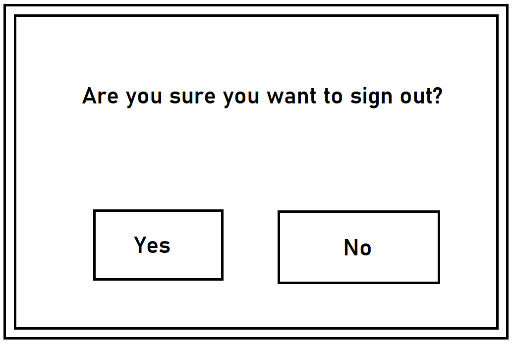
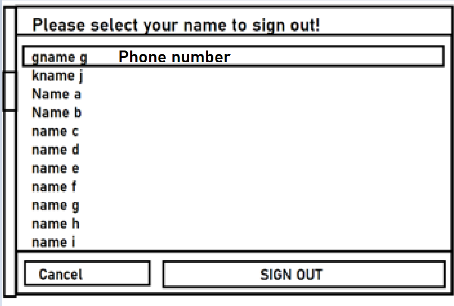
Potential problems:

On login, people may fill the fields with valid but useless data. An extension of this is that people may “spam” the login with junk entries. This can be fixed later with database operations, removing obviously broken logs.

There is no security regarding the logout feature, so someone could logout all the other users on site. This could be mitigated if we used different system where logout is controlled by scanning their phone or other object. We believe that the risks of system abuse are low considering the circumstances in which this application is being used, so we are content with keeping the login/logout system as is.

**User Experience Diagram – Normal Users - Tablet**

Logout



If all fields filled in

Cancel

No

Yes

Sign out

If fields not properly filled

Throw message and highlight improperly filled fields

Sign in

Login

## Staff UI

The UI used by staff members will follow along the same lines of design style, being simple to use and streamlined, but does not have to be designed around being “easy to use”, rather around the functionality it is to provide.

The UI will use the MySQL workshop as a base, as it provides all needed functionality and can be further modified using custom plugins and scripts.

**Staff UI Diagram**

# Non-Functional Requirements

## Language Requirements

As noted in previous sections, the tablet UI must be translatable into different languages for those unfamiliar with English. Additionally, those with impairments such as blindness should be accounted for.

As noted in the detailed design section, two solutions have been added:

Text to speech capability – Using the java.synthesiser and javax.speech , we are able to convert strings into spoken speech for sight impaired users. Volume can be adjusted using the tablet volume controls to suit the situation.

Language/Localisation capability – Via the tablet UI, options are made available to translate the contents of the UI, consisting of words and short sentences into different languages. The languages available should be English and Maori (being the official languages of New Zealand) as well as French, Spanish, German and Chinese (which are the next most common languages). To avoid the situation where a tablet is stuck in a different language, preventing other users from using it, the tablet should switch back to English mode once an operation is completed (successful sign in/out or cancel, returning to the main screen).

We considered adding an expanded text mode, where all controls and text is expanded in size which would be useful for somewhat sight-impaired users. However the tablet UI is designed to make controls as large and obvious as possible for simplicity and visibility, so this option is not required.

It is considered that during times of operation, it is unlikely a completely blind person is unlikely to be entering the premises without a companion. It is also unlikely that those who cannot understand the 6 listed languages above would be without a companion during the time this system is in operation, so it is believed that these measures for increased usability are enough for general use.

## Query Speed

# Conclusion

## Closing Notes

The Contact Tracer Network is a system intended to assist in contact tracing for COVID-19 cases in New Zealand, by installing a network of tablets throughout venues, which log visitors to those stores. The data will be stored securely in an offsite database using secure connections, to prevent exposure of sensitive data. Upon finding that a user is infected, NZPRU staff can check the stored log, sending email alerts to those in contact with the infected user. The system can go a step further, also alerting those in contact with close contacts of the infected. Finally the system can