Software Requirements Specification

COMP30830
Software Engineering

COVID-19 Track & Trace Smartphone Application

Prepared by
Shivam Bhatia
20200642

Table Of Contents

1.0 Introduction

1.1 Purpose

This document describes in detail software requirements specifications (SRS) for a '*COVID-19 Track And Trace*' smartphone application. It is addressed to all the stakeholders, such as any member of a software producer company or individual, and the National Health Service (NHS). A clear overview of the architecture of the software along with the design and specifications of the application is provided.

This is version 1.0 of the software requirements specification.

1.2 Scope

'CoviSafe' is a software to identify and notify all users who come in contact with a carrier of Covid-19 novel Coronavirus (nCOV). It is a smartphone application to help public health services contain the spread of a virus during a pandemic. This is possible due to the contact tracing feature of the application.

Since the ownership of internet-connected mobile devices has rapidly increased, they are a perfect tool to get real-time interaction data of people. This application will enable users to respond to health emergencies or any rules and regulations shared by the local authorities in real-time. It will aid in complying with crowd control practices such as lockdowns and social distancing.

The application is free-to-use software that can be downloaded from all major app stores such as Google Play Store or Apple App Store.

1.3 Definitions, Acronyms, and Abbreviations

The following table explains the key terms and abbreviations used in the document:

Term	Definition
User	Any person who has signed up on the application and has an active account. He/She can interact with the application.
NHS	National Health Service

HSE	Health Service Executive
nCOV	Coronaviruses (CoV) are a large family of viruses that cause illnesses ranging from the common cold to more severe diseases. A novel coronavirus (nCoV) is a new strain that has not been previously identified in humans.
Host	User infected with nCOV
Cryptographic Keys	Encrypted Unique Identifiers

1.4 Overview

The rest of this SRS analyzes the detailed requirements.

The document is organized as follows:

Section 1 provides an overview of the software. The purpose of this document is mentioned in this section.

Section 2 addresses the product specification. It mentions the perspective of this product and its functions.

Section 3 is a technical description of the functionality of this software. It specifies the functional and non-functional requirements.

2.0 Overall Description

2.1 Product Perspective

This software is a stand-alone smartphone application to serve as a contact-tracing platform. It will aid in tracking and tracing the contacts of the user exposed to a Covid-19 patient. The ultimate goal is to help health authorities break the chain of coronavirus infections by complementing manual tracing.

There is no system built in this application to store personal user information. The software does not even require permission to access location services.

It is extremely crucial to let people know about the below-listed facts:

- No location tracking
- No identity sharing

Smartphones will only communicate with other smartphones [Fig. 1]. No third party can access the system. This is a decentralized system. There is no need for a database.

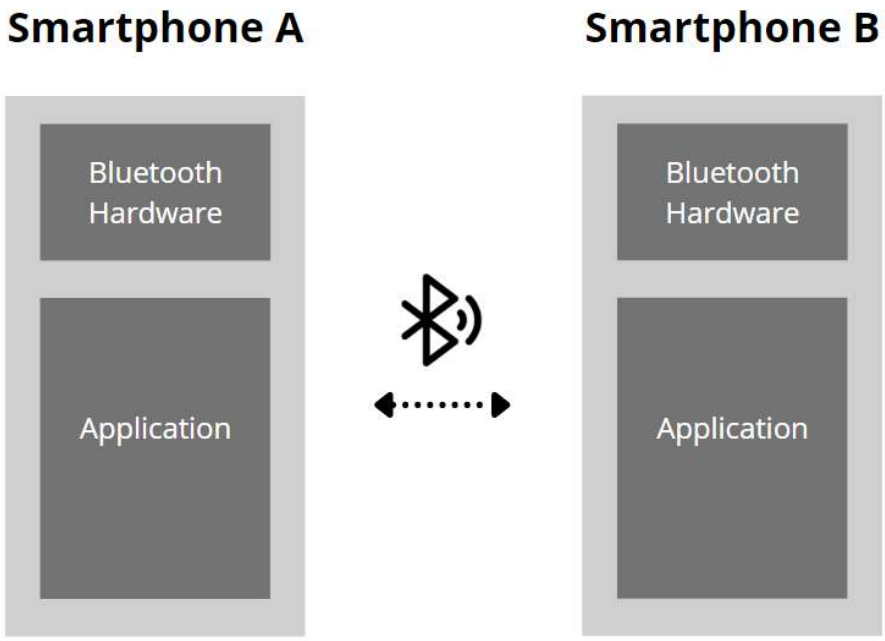


Fig. 1 System Architecture

A contact tracing system can only be successful if more people use it. This can be made possible by making them trust a public health technology that protects individual privacy.

2.2 Product Functions

The application should be very easy to set up. Once the initial setup has been completed the application should remain dormant for an indefinite interval of time. There should be no intervention needed from the user unless he/she receives an exposure notification due to close proximity to a host.

The application will never request the personal information of the users. The only permission it needs is to access Bluetooth API on the device. The application should prompt the user to keep the device Bluetooth turned on at all times.

On the '*Terms Of Service*' page, the user should be able to read how the application works in a concise and easy-to-understand language. The specified statements should be included: *The software will generate random unique identifiers on the device and offline at fixed intervals of time. These cryptographic keys will be unique to each device. Each key contains information about the time and date when it was generated. When two smartphones with this application pass in each other's proximity [Fig. 2], they share all the generated keys from the past 14 days with another smartphone. This functionality will maintain a log of multiple keys on a user's device for every user he or she has been in close contact with for the past 14 days.*

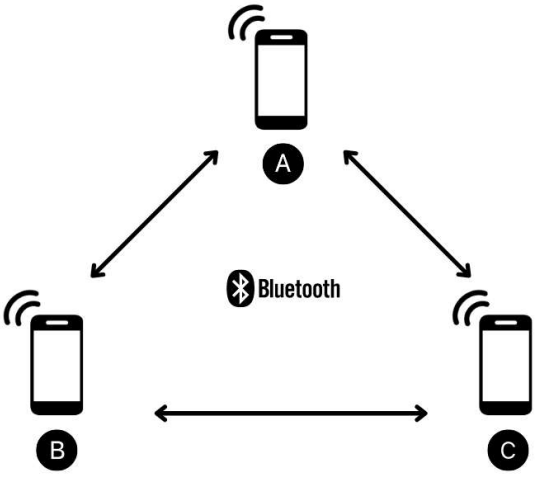


Fig. 2 Smartphones in close proximity and sharing keys

If one day he tests positive for Coronavirus, he would become a host. As soon as he updates his status on the application, all his unique keys will be blacklisted and broadcasted across the network. Every device with matching keys will be issued exposure notifications and advised to get tested.

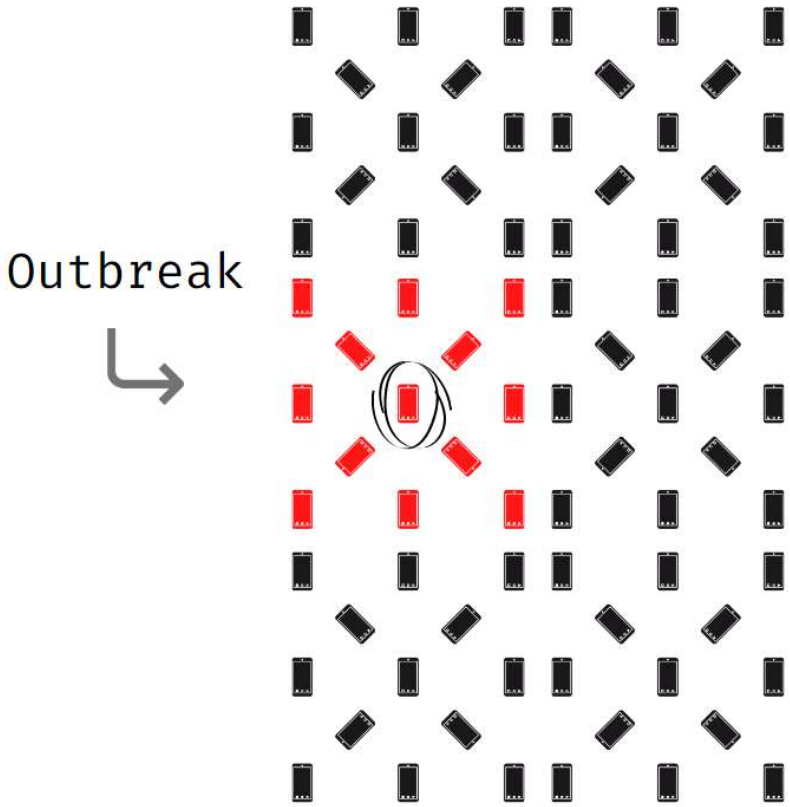


Fig. 3 Smartphones in close proximity

These keys only contain information about the date and time when they were generated. They contain no personal information of the user. Hence, there is no way to track the user. The user stays anonymous.

There are three features of the app:

- Exposure Notification
- Quarantine location broadcast
- Route planner

2.2.1 Exposure Notification System

On close encounters, if the keys of an anonymous host user match one of the keys saved locally on the device the application will alert the user with notifications. There will be a high priority alert pinned at the top of the tray. The user should also receive an automated call generated locally with a pre-recorded message telling him to check the status of the application.

2.2.2 Quarantine location broadcast

The whereabouts of the users are not tracked through GPS. The host, i.e. any user who tested positive for Coronavirus, should voluntarily update his quarantining location with a degree of precision of his choice. Since this is the location where he will be staying for an extended period of days, it can be marked on the map. When multiple users start pinning their quarantine locations a heatmap will be generated to detect hotspots.

This feature will again keep the users anonymous.

The users will be able to view these heatmaps and plan their outings accordingly. The heatmaps will be precise. The color should be in sharp contrast to the map to make it easy to view the varying density of the positive cases.

2.2.3 Route planner

This feature will enable users to plan their route from one location to another. The user should be able to choose a point. The application should suggest the shortest route accordingly.

2.3 User Classes And Characteristics

This application is intended to be used by anyone with a smartphone. These smartphones should only run on either IOS or Android.

The type of users can be differentiated in two ways:

- Frequency of toggling Bluetooth on or off
- Interaction with Route planning feature

2.4 Constraints

The application should be able to run with full functionality even on low-end smartphones with at least Bluetooth 4.0. The operating system should be either IOS or Android. It should be compatible with the previous 3 versions of the mentioned systems.

The language of the software should only be English.

Any personal information through which the user can be tracked should never be collected, stored, and processed. All data should be erased after 14 days.

At all points during the development, the below-mentioned checklist should be referred to maintain accessibility:

- Ensure that users have access to the operating system accessibility tools, without affecting application functionality
- Ensure compatibility with assistive technologies
- Adhere to all user-selected system settings for input and output
- Adhere to the standard keyboard access methods
- Ensure that all information can be perceived by users with restricted vision
- Ensure that all information can be perceived by users with restricted hearing
- Do not cause the screen to flash at a frequency of above 2 Hertz
- Use the simplest language possible for instructions, prompts, and outputs and, where possible, supplement it with pictorial information or spoken language
- Provide descriptions and instructions for all accessibility features
- Allow sufficient response time to accommodate the slowest users

2.5 User Documentation

There should be 2 ways the user can take assistance to learn to use the application:

- showcase app's features and functionality on its store listing page
    - One 30-second video giving a walkthrough of the application
    - 10 screenshots with graphic assets on User Interface

2.6 Assumptions And Dependencies

It is assumed that the developers have implemented the internationally accepted and standard Bluetooth API.

Currently, IOS and Android operating systems are dominating the market with more than 99% market share. It is assumed that the application is compatible with most of these smartphones.

3.0 External Interface Requirements

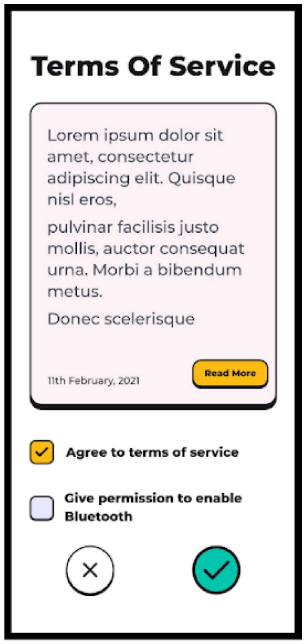
This section gives a detailed description of the system and its features. All functional and quality requirements along with inputs into and outputs from the system are also mentioned.

3.1 User Interfaces

The first screen that the user sees should be a loading screen with a logo and the name of the application (Screen 1). After loading the resources and waiting for at least 1.5 seconds, the

Screen 1

Screen 2

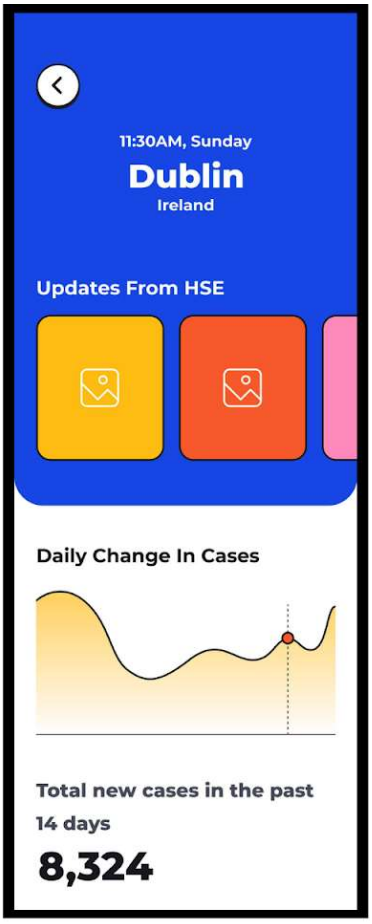


Screen 3

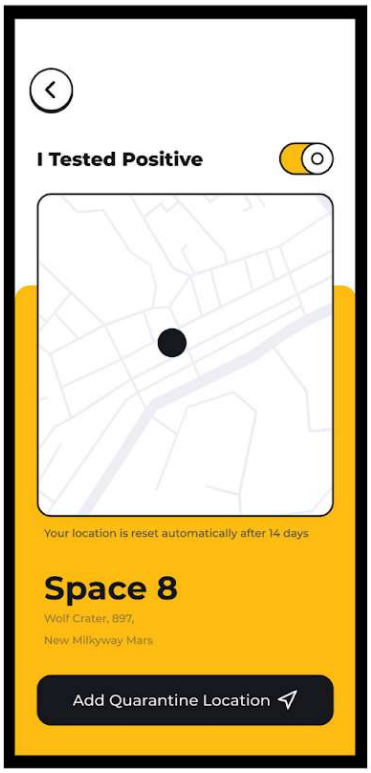
user should be greeted with a second screen as shown (Screen 2). It asks the user if he wants to continue. The next screen should display the terms of service of the software (Screen 3). It should ask permission to access Bluetooth hardware on the device. The user should only be able to proceed if he agrees with the terms and conditions of use. There should be an 'exit' button in case the user decides to quit.

The landing screen is where the user should always be sent when the application is started (Screen 4). It should feature a full-screen map in the background. There should be an app bar at the bottom and a search bar at the top. Hotspots should be visible on the heatmap as bubbles. The app bar should have three buttons: updates page, add test result page, and settings page. The settings page (screen 5) should have all the settings: toggle Bluetooth, toggle automated call notification, mute app sound, toggle battery saver feature, and an invitation form.

Updates page (Screen 6) should show the user's city (detected when the user logs quarantine location), updates from Public Health Department, and Covid-19 statistics.



Screen 6



Screen 7

When the '*Plus*' button is tapped, the user should land on a page where he can update his quarantine location (Screen 7). There should be a toggle button for a positive result that activates the map. On clicking, a search bar should pop up for him to look for the place on the map.

3.2 Hardware Interfaces

Since neither the mobile application nor the web portal has any designated hardware, it does not have any direct hardware interfaces. Both the Bluetooth API and Network connections are managed by the respective operating systems.

3.3 Functional Requirements

This section discusses the requirements for all the necessary actions of the application. Since there is only one use case for a user there is only one user class.

ID: FR1

TITLE: Download mobile application
DESC: The application should be free to download manually from Google Play Services and Apple App Store. All application updates should be provided through the same store.
RAT: In order for a user to register on the mobile application.
DEP: None

ID: FR2

TITLE: release updates automatically for the mobile application
DESC: The application should be updated automatically from Google Play Services and Apple App Store.
RAT: In order for a user to update the mobile application.
DEP: FR1

ID: FR3

TITLE: user should be able to agree to the terms and conditions and proceed or deny and quit
DESC: The application should only start if the user has agreed to the terms of service
RAT: In order for a user to agree to the terms of service of the mobile application.
DEP: none

ID: FR4

TITLE: Home Screen

DESC: The home screen is the default and main screen of the application. The user should be able to navigate to any page from here. It should have a search bar to find a safe route through the hotspots, an app bar with 3 tabs: updates page, settings page, and a button to add Covid-19 test results.
RAT: In order for a user to navigate through the application

ID: FR5

TITLE: Planning Safe Route
DESC: The user should be able to find a safe route between two points on the map. The software should respond with a path with its total distance that avoids all the hotspots. The hotspots should be visible on the map.
RAT: In order for a user to navigate safely to a location

ID: FR6

TITLE: Updates page
DESC: The Updates page should brief the user with the following information: His local time and city (if the location has been updated by him with the test result), real-time briefings from HSE as cards, a graph showing the daily change in the new Covid-19 cases and the total number of new Covid-19 cases in the last 14 days.
RAT: In order for a user to view Covid-19 statistics and updates from HSE

ID: FR7

TITLE: Toggle Bluetooth
DESC: The user should be able to turn Bluetooth on or off at his will. This setting should be available as a checkbox on the Settings page.
RAT: In order for a user to control Bluetooth on his device.

ID: FR8

TITLE: Disable or Enable automated call notification
DESC: The user should be able to turn off or on the automated call notification
RAT: In order for a user to control call notification.

ID: FR10

TITLE: Mute notification sounds
DESC: The user should be able to mute the sounds of all the notifications from the application on the Settings page. This feature should be available as a checkbox.
RAT: In order for a user to control application notifications with sound.

ID: FR11

TITLE: Battery Saver
DESC: The user should be able to control the Battery Saver feature which turns off the Bluetooth if the device battery is below 15%. This feature should be available as a checkbox.
RAT: In order for a user to save device battery by switching off Bluetooth.

ID: FR12

TITLE: Invitation Form
DESC: The user should be able to send an invite to his contacts to invite them to the application. It should ask for their name and email address. There should be a submit button that should send an automated email to the specified email with a link to download the application from the stores.
RAT: In order for a user to invite others to the platform.

ID: FR13

TITLE: update Quarantine location
DESC: The user should be able to add his quarantine location to the application. His and others' quarantine locations should be shown on the map as a bubble heatmap. This action should be triggered through a button available on the home screen app bar.
RAT: In order for a user to update his quarantine location on the map.

3.4 Design Constraints

3.4.1 Hard Drive Space

ID: DC 1
TAG: HardDriveSpace
GIST: Hard drive space
SCALE: The application's need for hard drive space.
METER: MB
MUST: No more than 80 MB
PLAN: No more than 50 MB

3.4.2 Memory Space Usage

ID: MSU 1
TAG: ApplicationMemoryUsage
GIST: Application memory usage in the operating system
SCALE: Observations done from the performance log during testing
METER: MB
MUST: No more than 150 MB
PLAN: No more than 100 MB

3.4.3 Software System Constraints

ID: SSC 1
TITLE: Internet Connection
DESC: The application should be connected to the Internet.
RAT: In order for the application to communicate with the database