0. Table of contents

1. Introduction

1.1 Overview

The aim of our project was to create a COVID-19 contact tracing app that features a BLE Social distancing feature that can detect phones closeby and how long they were in close contact.

Our primary focus was the contact tracing aspect that's being used worldwide to combat the spread. Our application provides users with live data on the current state of COVID-19 worldwide, giving users access to case numbers, deaths, tests and those who have recovered.

The application also informs users of the many symptoms of COVID-19, how to protect yourself, how and when to Isolate and how we as a people can reduce the spread.

Should a user feel interested in discovering the latest news on COVID related sources they may opt to use the news feature which displays the top headlines within a selected country, which can be changed via the settings option.

While the social distancing feature of our application involves using Bluetooth Low Energy to detect nearby phones, and warn a user that they have been in close contact with another person over a period of 15 minutes in a 24 hour cycle.

This document will describe the application and its features in more detail and also bring to light the challenges we faced in creating our application, and furthermore how we overcame these said challenges.

1.2 Glossary

BLE - Bluetooth Low Energy

Retrofit - Rest client library for android that makes it easy to consume/parson JSON

XML - Extensible Markup Language

JSON - JavaScript Object Notation, Data interchange format

Google firebase - Firebase is a mobile and web application development platform

Android studio - Android Studio is the official integrated development environment for Google's Android operating system

Newsapi.org - A website that gathers worldwide news and allows developers to create News API's to display current news, old news or topical news based on what the developer has requested through his application

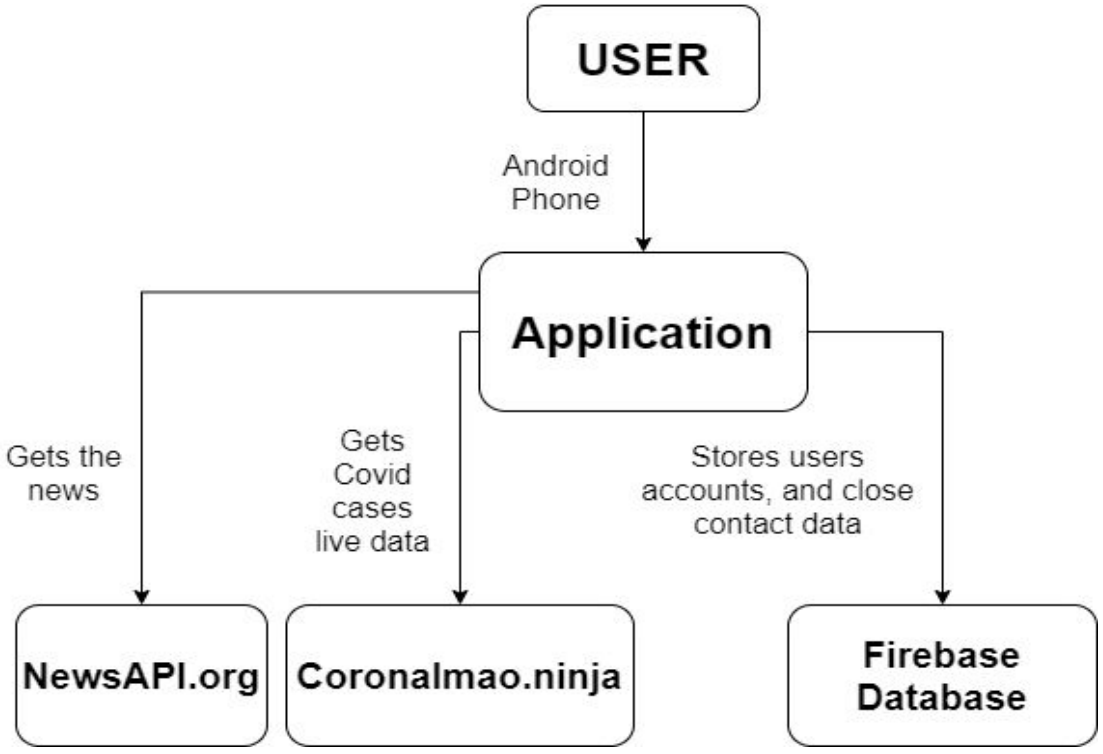
Coronalmao.ninja - A website that displays JSON data of live COVID-19 case data

Volley - Volley is an HTTP library that makes networking very easy and fast, for Android apps. It manages the processing and caching of network requests and it saves developers valuable time from writing the same network call/cache code again and again.

2. System Architecture

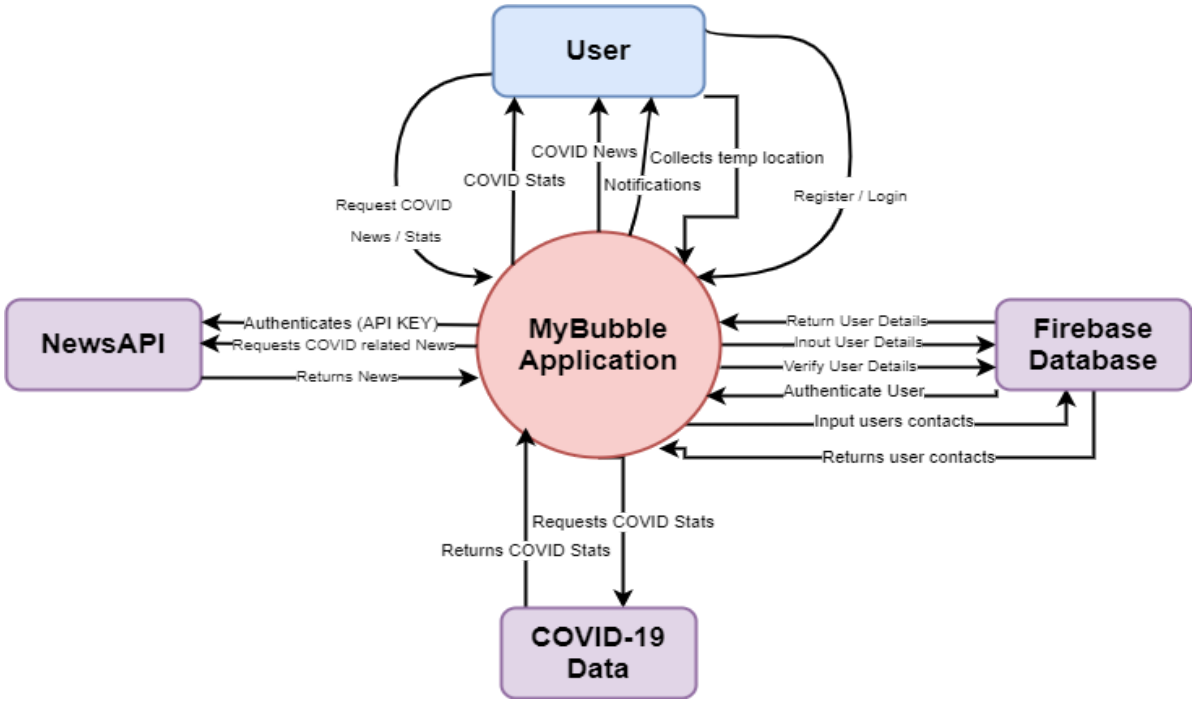
The system architecture hasn't changed much from the functional specifications apart from a more clear approach to our architecture as we have now built the application, which involves the use of third party API's (NewsAPI.org, Coronalmao.ninja and Firebase.google.com)

Figure 1 - System Architecture



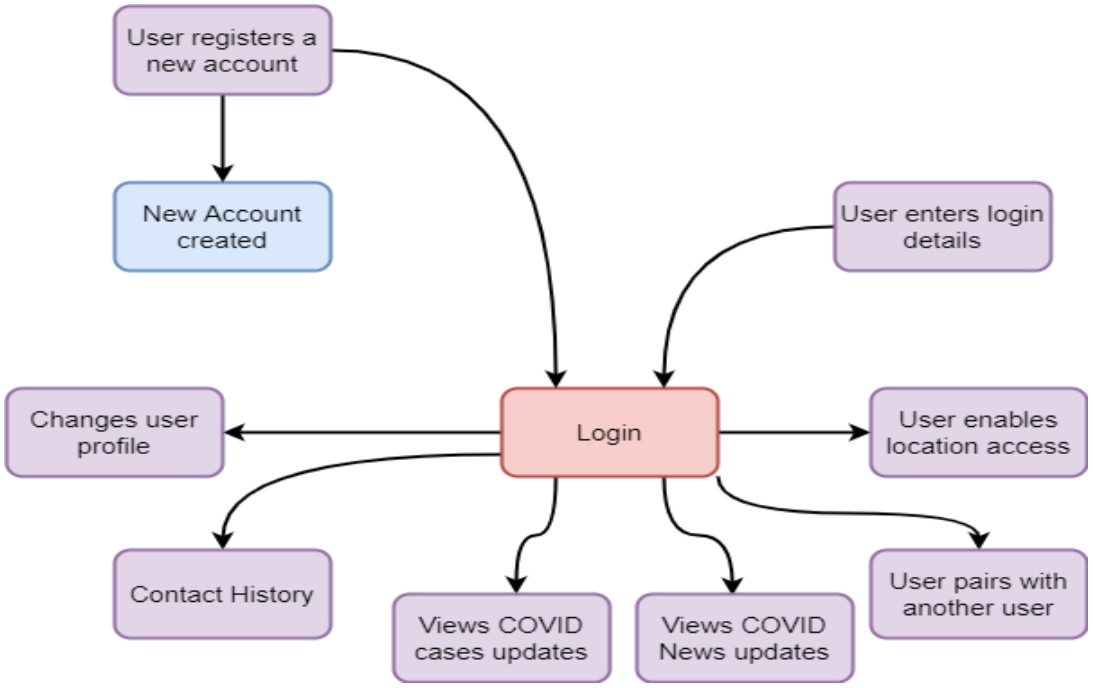
3. High-Level Design

Figure 1 - Context Diagram



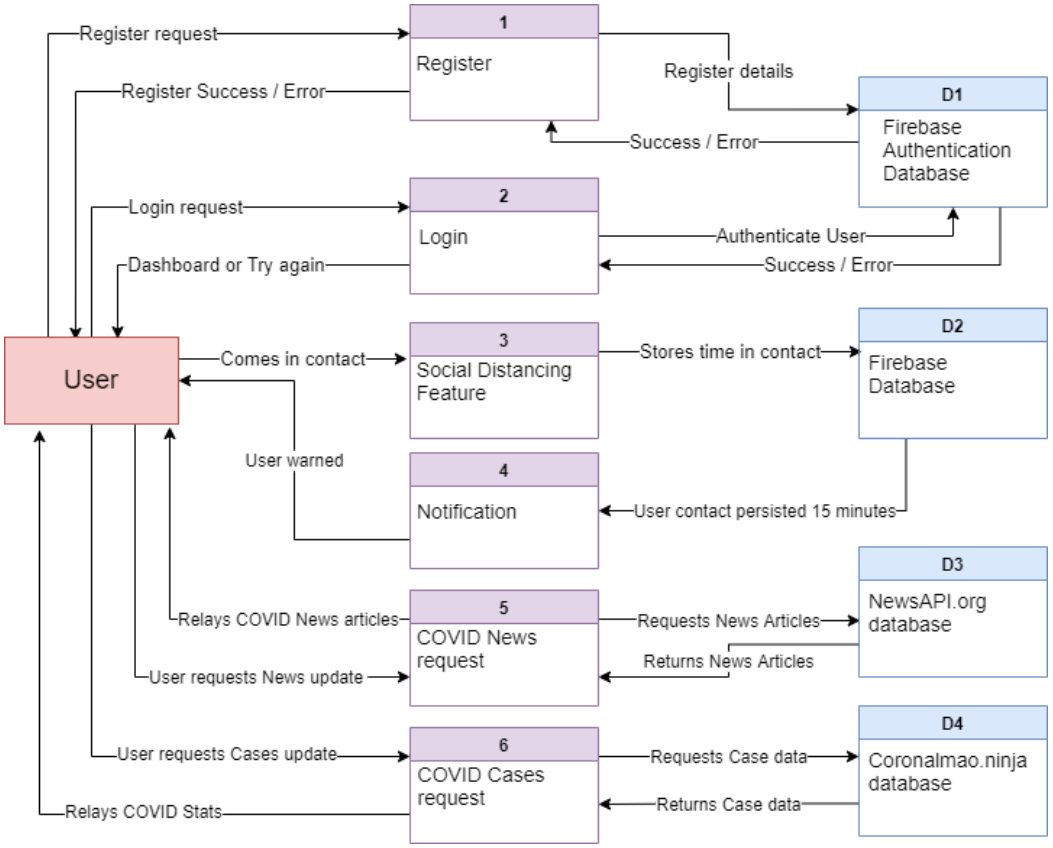
Depicts the Context aspect of the application where the application makes contact with each external part of the application. Making requests, returning those requests, verifying details and giving the user the data they have asked for.

Figure 2 - Activity Diagram



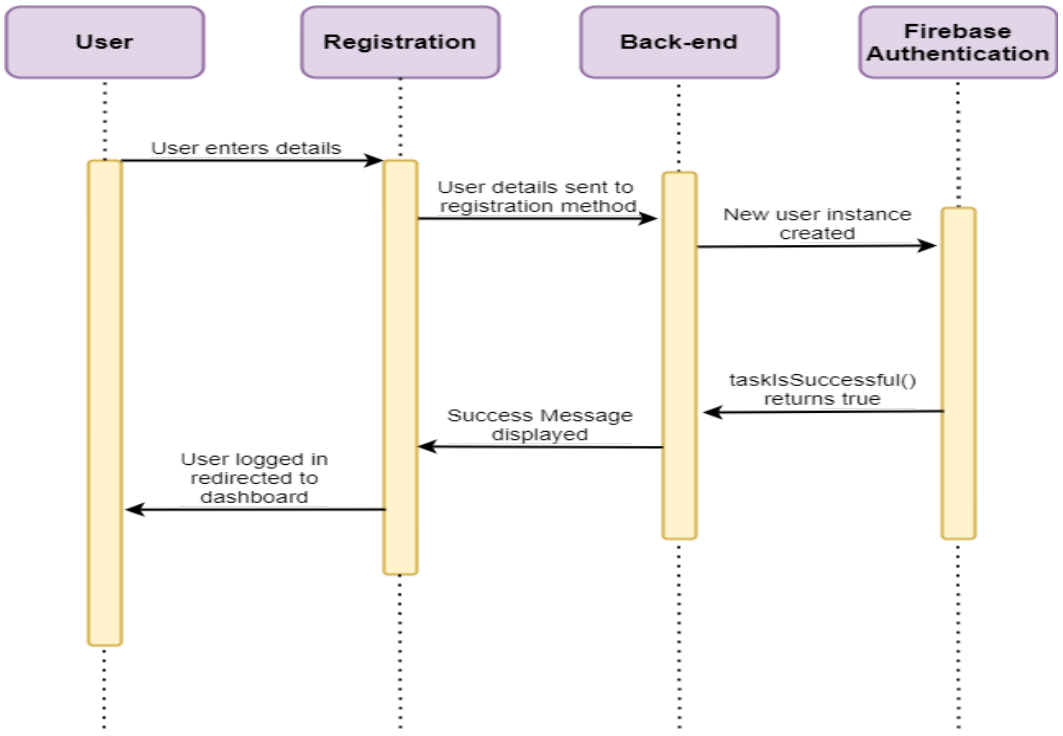
This is the App activity, from Login, how the user will make their way through the application from the login page. A user will set up an account, enter login details and then use one of the many features of the application such as adding a friend to the social bubble or viewing live COVID Cases.

Figure 3 - Data flow diagram



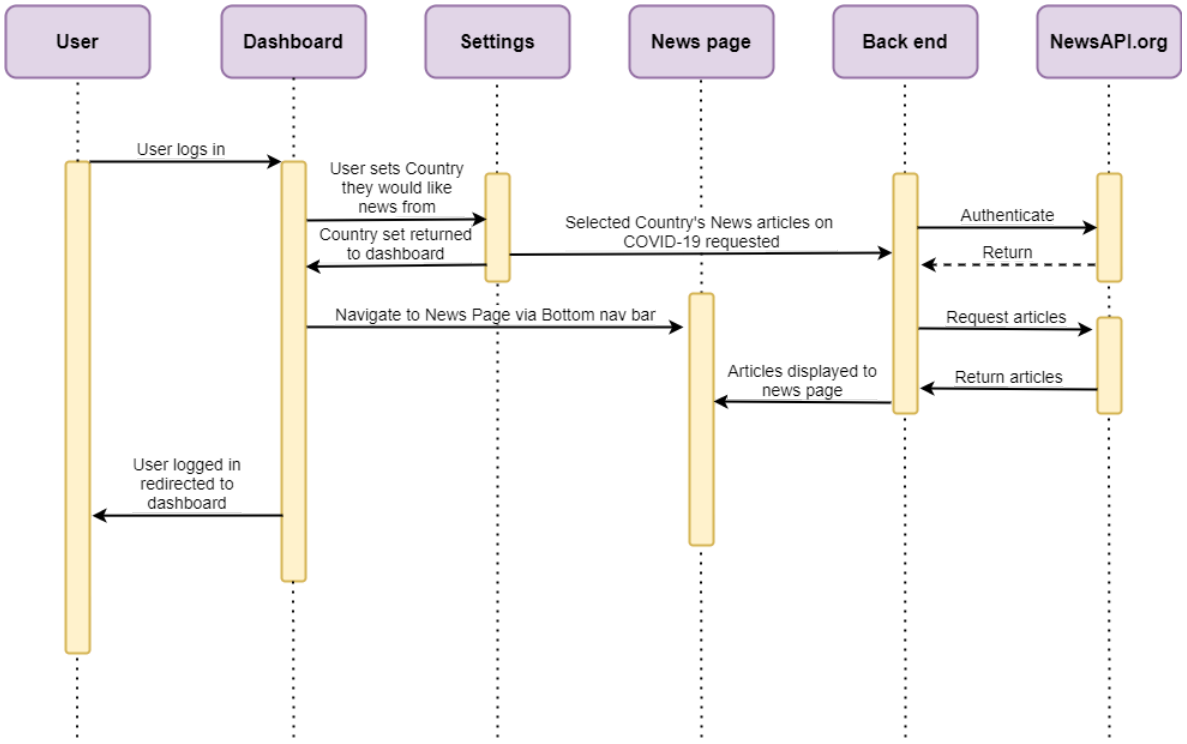
This diagram depicts the flow of data throughout the app, a user will send requests for data through various methods, which will then grab that data, and either store or return new data for the user to see and view.

Figure 4.1 - Registration Sequence Diagram



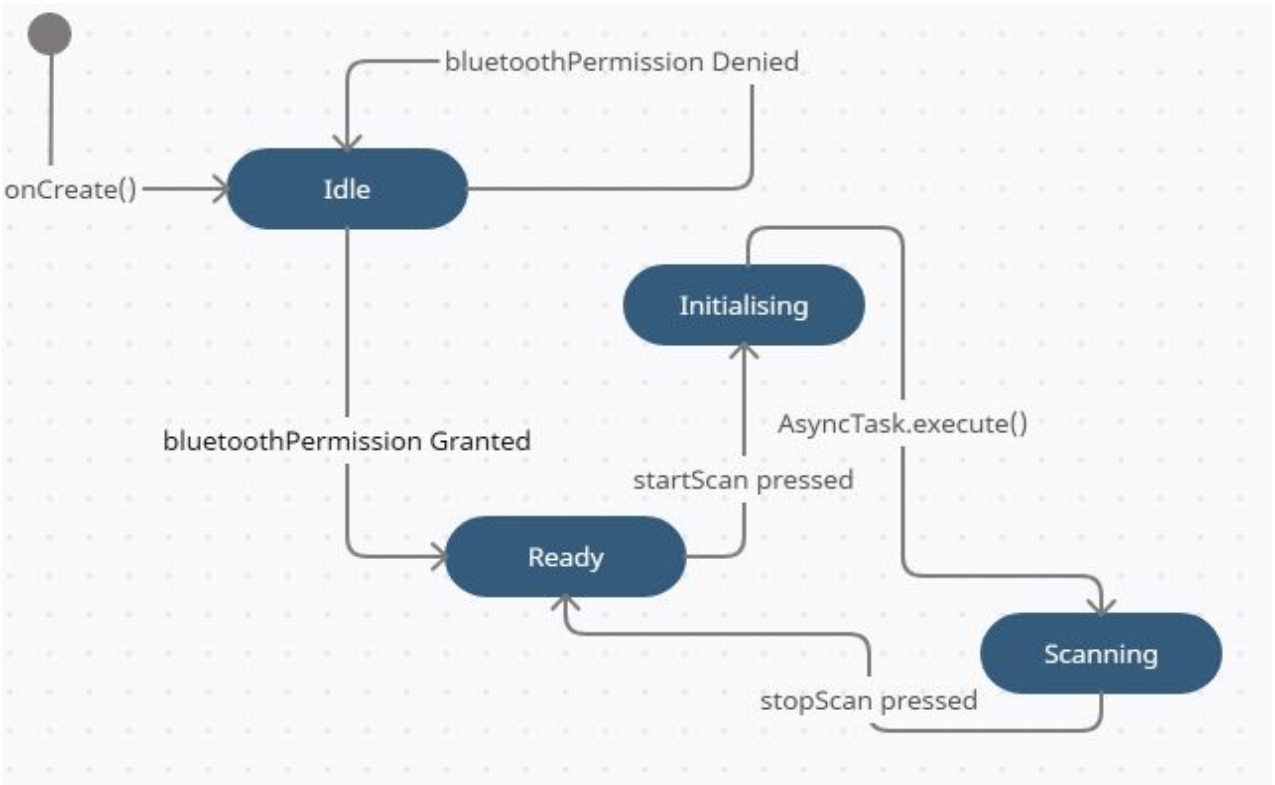
The following diagram depicts the sequence process of registering a new account, working between what the user sees, to the backend and then to the database and then all the way back to the user.

Figure 4.2 - News API Sequence Diagram



The following Sequence diagram depicts the sequential process of requesting news data, where a user will select a country for the news to come from, this will then be sent to the backend where it will process a URL to request the data from the NewsAPI.org website. This request will then be processed and return back the relevant articles that the user may want to see.

Figure 5 - State Diagram of Social Distance Breach Scanner



This diagram depicts the State diagram process of our Social Distance feature, which will notify a user if they are in breach of social distancing conduct by scanning for nearby devices, location them and then notifying if within their bubble.

The formula we used to calculate the distance between two devices is as follows;

*Distance = 10 ^ ((Measured Power — RSSI)/(10 * N))*

Where the measured power is a constant which varies from device to device, RSSI or Received Signal Strength Indicator is a measure of the power of the signal sent back to the BLE Scanner and N is a constant environmental factor.
Both Measured Power and Environmental Factors were estimations based on examples. It would be very difficult to estimate measured power for every device so we used a constant across all devices. This also allowed for more consistent testing of the scanner.
Environmental factors would be almost impossible to calculate with a mobile device so again we opted to go for a fixed constant across the board.

4. Problems and Resolution

4.1:

Problem: Understanding the android development environment.

Resolution: Began by following numerous introductions to Android Studio via youtube and a course on Udemy, followed by implementing some basic Applications to perform simple tasks. Once we completed and went through a lot of tutorial content we became familiar with Android studio and also learned how to read and fix common errors thrown when first beginning with the IDE, the best practice is learning from a difficult bug and stack overflow if you're lucky.

4.2:

*Problem***:** Figuring out how to build the correct requests to interact with the News API. Initially we were not adding the correct parameters to our authorization request and struggled to understand why we were not getting a bearer token in response to our requests.

*Solution***:** Read all of the relevant API documentations in the News Api docs, watched several youtube videos on building a News REST API, and went through many stack overflow questions which led us to use retrofit and multiple built in libraries to figure out how to handle requests and responses. Figured out which required parameters were missing from the initial request (It required creating a variable of the API Key at run time that would generate within the BuildConfig) and resolved the issue.

4.3:

Problem: Becoming familiar with Firebase

Solution: Firebase is highly specialised to use in Android applications and seems to be the industry standard. Familiarising ourselves with Firebase Firestore's structure and implementation within Android Studio was a challenge that we overcame through online resources such as StackOverflow, Medium and javatpoint. The absence of a full-text search also proved an issue but we found ways to get around this.

4.4:

Problem: Implementing a NavBar for activities, to coincide with the rest of the application

Solution: A big obstacle involved in developing the app was coming up with a functional NavBar that would display for both fragments and Activities, it was decided in the end to opt towards creating a fragment of the activity, which would allow for proper navigation rather than displaying a back button on an activity page which would return the user to

the dashboard. However, some libraries did not display proper documentation on implementing them in a fragment and that also raised an issue.

4.5:

Problem: Using GPS for Detecting nearby devices was not accurate enough

Solution: Having worked on a GPS feature for social distancing it turned out to be quite inaccurate and not providing substantial data that we required and hence we opted for a more accurate implementation of using Bluetooth to detect nearby devices and warn a user if it's within its social bubble.

5. Development Cycle

- In the development process of our android application we used Android studio, this is a free and reliable platform for android app development and it is used by the vast majority of developers in the industry. For our server and database we used Google Firebase, Firebase comes with a great amount of tools such as a database, storage, authentication and analytics. It is also free and stable as it is provided by google. The firebase services that we used authentication to allow users to create accounts and login to our app and a database to store user temporary data.
- Our final implementation of the project is very similar to the original idea set out in the functional specification at the beginning of this development cycle. All the main features that we set out to do such as social distancing feature, covid case tracker, news tracker and a clean UI have been successfully implemented and are working as intended.
- Some features however had to be left out of the final app due to us running out of time. For example we intended to allow users to make a health check in which would allow them to notify others if they tested positive for COVID-19, we also opted to get rid of the GPS function of the social distancing feature as we felt that for the time it took to create a fully functional application using GPS it would not give the same accuracy or results as using Bluetooth.

- If given more time and if we continue work on this application in our spare time these features would be implemented and the app will be exactly as we set out in the functional specification, if not even more improved with a more fluid UI, and a more competent Social distancing feature.

6. Installation Guide

6.1 Requirements

You will need the following to install and use our application:

- Android Phone - With a device capable of running android 5.0 and above.
- Latest version of Android Studio
- Internet Connection
- Java 8+

6.2 How to install

- Download our GitLab repo at https://gitlab.computing.dcu.ie/joycem32/2021-ca326-jmichael-projectmanager
- Save the repo to a folder of your choice
- Make sure developer mode is enabled on your Android device and USB Debugging is enabled, to enable developer mode you must tap the Build Number of your phone in your “About Phone” section of your phone settings at least 7 times.
- Open the project in Android studio and plug in your Android phone to the computer/laptop via a USB cable.
- Click the run button, which looks like a green play button in the top right corner of the Android studio project and select your android phone as the target device, if you connect your phone it should default to your physical device, otherwise you also have the option of running the app within one of the many emulators that android studio offers.
- The app will automatically install and you can then run the app, if it doesn’t open automatically.

Obviously, should this application go to market it could be installed via the Google play store, or even quite possibly through a website via an APK.

7. References

- https://firebase.google.com/docs/android/setup
- https://developer.android.com/studio
- https://square.github.io/retrofit/
- https://github.com/google/volley

- https://corona.lmao.ninja/
- https://newsapi.org/
- Creating a Coronavirus tracker app: https://www.youtube.com/watch?v=P60i4P4E_e0
- Creating a NewsAPI: https://www.youtube.com/watch?v=MNaP12Q2ZT8
- Learning Firebase: https://www.youtube.com/watch?v=i-gZAYBMuBs