**Creation and Distribution of an Android Application:**

**COVID Self-Report**

**By**

Russell Smith,

Dhanush Karthikeyan,

Sydney Cady

College of Engineering

Department of Electrical and Computer Engineering

Advisor: Tim Lin

Spring Semester 2021

May 13, 2020

Table of Contents

[Abstract 3](#_Toc71806677)

[Introduction 3](#_Toc71806678)

[Design 3](#_Toc71806679)

[Project 3](#_Toc71806680)

[Android Application Platform 4](#_Toc71806681)

[First Semester Progress 5](#_Toc71806682)

[Standards and Constraints 8](#_Toc71806683)

[Android Studio Maker 8](#_Toc71806684)

[Agile 8](#_Toc71806685)

[Methodology 9](#_Toc71806686)

[Transition of Virtual Format 9](#_Toc71806687)

[Agile Development 9](#_Toc71806688)

[Development 10](#_Toc71806689)

[Code (Functions) 15](#_Toc71806690)

[Testing 16](#_Toc71806691)

[Deployment 16](#_Toc71806692)

[APK 16](#_Toc71806693)

[Results 17](#_Toc71806694)

[Future Considerations 19](#_Toc71806695)

[Conclusion 20](#_Toc71806696)

[References 21](#_Toc71806697)

# Abstract

The main objective for this project is to develop an app that helps users self-report COVID symptoms, outgoings and notify contacts of illness. One of our verticals was to service those impacted COVID and create an application that would augment contact tracing, while still facilitating in-person meetups. The team is designing an app for android in Java using Android App Studio Maker to accomplish this program. The app will hold contact information, personal health issues and preferences, and allow for daily check-ins. We hope that this app makes COVID a little easier for those who tend to lose track of days and are cautious. This app is meant for users to ensure they do not lose track of days and any symptoms that may appear. This can be very useful for those who are in high-risk thus meaning they should remain diligent and on top of reporting any symptoms to roommates, family members or people within their close circle. We wanted to build something with great intention of having an impact on people and people’s lives and we hope that this app has accomplished this and can eventually do more for communities or individuals.

# Introduction

When creating our senior project, we have the goal to build something impactful and useful in today’s world. One of our verticals was to service those impacted COVID and create an application that would augment contact tracing, while still facilitating in-person meetups.

Initially, we had a couple ideas as to what we could do for our project. After considering how reasonable and reliable some of the ideas were to implement, we settled on doing a COVID self-reporting app in where users could record their outgoings, where they were comfortable meeting with friends and sharing this type of data in a safe and secure manner.

We decided on the COVID self-report because it is something that had impacted everyone, and everyone could agree with the sentiment that the days were becoming blurred. This app would allow for a running log so users could be aware of where they went out or who they out with and when symptoms become present. Users would also be able to share their profiles with each other to determine if they are comfortable meeting up and the type of activity, they are comfortable doing together, based on what they see in the other user’s profile. We believe that this will be very helpful during these times, as the world is slowly starting to open back up.

# Design

## Project

Our senior project consists of us creating an application where Users would also be able to share their profiles with each other to determine if they are comfortable meeting up and the type of activity, they are comfortable doing together, based on what they see in the other user’s profile.

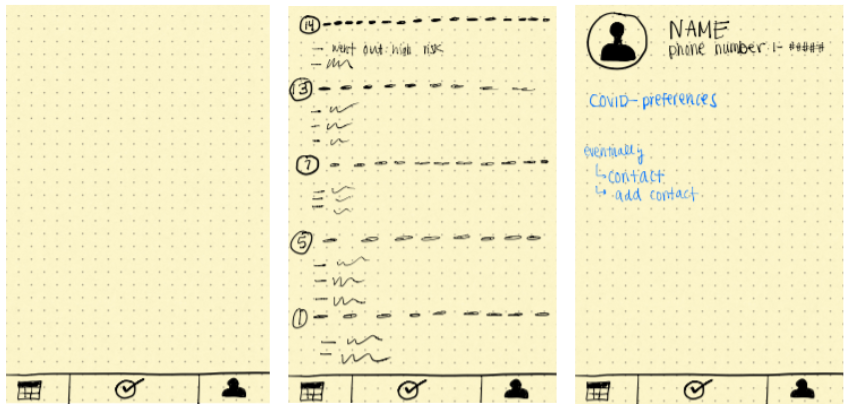
Since circumstances have changed because of a global pandemic we have had to adjust and make changes to the way our project will progress. We had to switch to a completely virtual implementation of our project for which entailed software development and collaboration.

## Android Application Platform

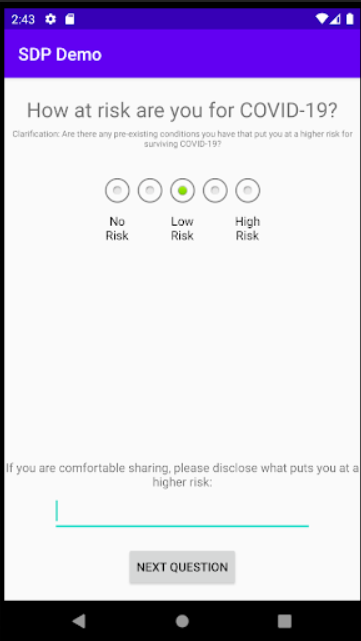
As defined per the Android documentation, Android applications are software which can be used on any Android operating system. They are either pre-installed with Android OS or can be installed. Applications are created for a specific purpose, such as accessing Google’s Gmail client via a dedicated application. Android applications are officially available from the Google Play Store, which is basically Google’s online store for various software. However, some platforms such as the Google Play Store require payment to host on their platform. Thus, it is often beneficial to “sideload” which is where Android allows you to download an APK file (Android app file extension) and install it on your phone. It will ask for confirmation whether you want to install an app from unverified source first. After this minor notification, it is possible to install any app. This is the approach we will be using for our senior project.

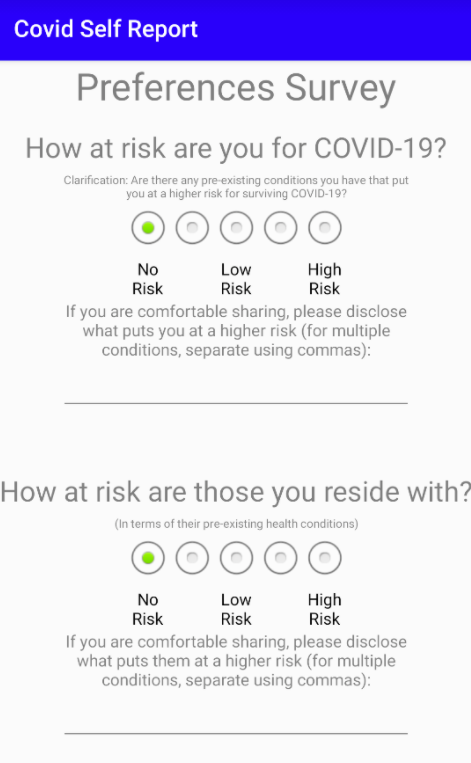
## First Semester Progress

The first semester was spent planning our project. We were able to get our Gantt chart done, thus planning our rest of the project. In these times we were able to come up with an approach as to how to complete the project. We settled on a commonly used methodology which was using Agile. Or at least, taking some concepts from Agile to better split up our work. Through understanding agile we knew that our first semester would mainly be planning. First and foremost, after some research, we learned that it would be best to create drawings or mockups of what the app would look like. Below you will find photos of some of our first drawings.

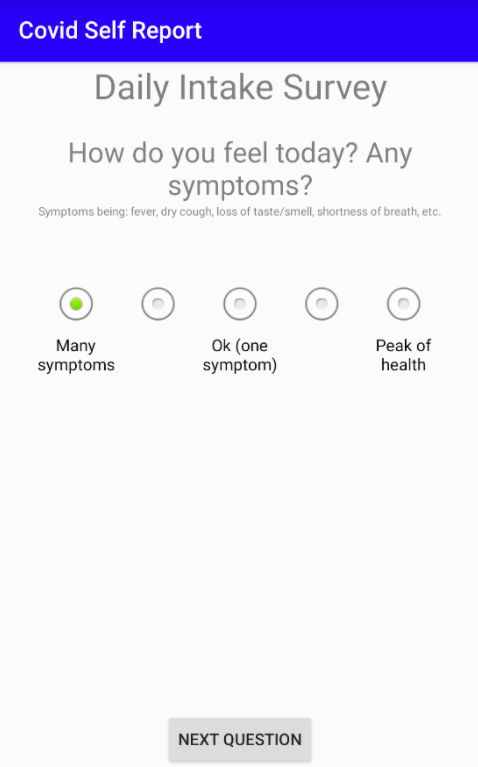


After being able to have a good idea of what our project would be and how it would work, we were able to effectively split up the work and determine our next steps through Agile methods and utilizing our skills. After this step we started to work on the app itself. By the end of the first semester, we were able to get the surveys created. Below will be photos of some of the survey questions as it is shown on the app.





Even though our first semester was mainly in planning and not much app work gone done. Due to all this prep work that went into the first semester we were able to fully complete our project by the end of the second semester.





Overall, our first semester was spent planning and getting some app basics down to better prepare ourselves for the rest of the app work that would be the bulk of the work. This methodology mainly comes from Agile as a lot of planning work goes into Agile which allows the workers to work more efficiently get most of the grunt work done quickly, which is something we demonstrated throughout this project.

# Standards and Constraints

## Android Studio Maker

Our standards for our first method include the Cisco Catalyst 2960-X Series, raspberry pi machines to act as clients, ethernet cables, and a computer monitor. We used VirtualBox, which is a cross-platform virtualization application, and installed Ubuntu, our guest machine. We also installed OpenDaylight as our controller. The OpenDaylight (ODL) platform provided a flexible common platform that involves network resource optimization, automated service delivery, and network visibility and control. Within OpenDaylight, we had to use Yang UI, which is a user interface application where one can navigate among all Yang models in the controller. Although we were able to successfully download all the required software applications, our topology did not show up. We were able to see the nodes and the packets being sent, but the flows and topology were not clear. There were constraints for this method, starting with the Cisco switch. The first 6 ports were only available for Openflow and if we attached the ethernet cables to any other ports, then it would not work. Our second constraint was that we were only able to use one computer monitor due to the availability of resources. Because we were only able to use one, we could not see our clients and the OpenDaylight controller windows at the same time, so we had to keep switching back and forth.

## Agile

The Covid Self Report app was modeled off of Agile development. Agile methods focus on the rapid development and delivery of software, often in weekly increment deliveries. Agile development itself also focuses on interleaving the program specification, design, and implementation, and testing steps of software development. This means that a detailed plan is not used for development. The project is planned as development progresses, not before it starts. The focus is to move quickly, which often times cannot be done using a plan-driven approach. Because of the fast pace of agile development, documentation is also minimal. There is not focus on writing up separate documents on the software, and the code is not guaranteed to be thoroughly commented. The focus is on writing working code instead. Agile methods were chosen for this project because it is very widely used and still growing in the software development industry. This way, project team members can gain exposure to the agile methods that they will likely have to learn in the industry, giving them a head start. The app was planned to be developed in weekly “sprints”, taking after the sprint practice in Scrum, which is a version of agile development. Sprints come from the focus of frequent version delivery. Each sprint focuses on the development of a specific feature, and upon the end of the sprint, when the feature is finished, a new version is delivered (pushed to the GitHub in the case of this app). Toward the end of each sprint, before finalizing the feature being worked on, refactoring occurred, where code was cleaned up and duplicate methods and classes were removed. This made the project more maintainable, even without a heavily detailed plan.

# Methodology

## Transition of Virtual Format

Due to the transition to virtual format, we notified us that in person classes were not going to occur anymore because of the COVID-19 pandemic. Consequently, we had to adjust our project. Our group had to transition to a totally virtual project with little to no physical elements since working with the physical switches was not a viable option.

Luckily, we were fortunate to a start our project during quarantine. Thus, we pivoted our project from conception to better align with virtual means. We chose a software project and found a very proficient professor who specializes in programming. This mentorship has been especially invaluable as we continue our development and deployment of our application.

## Agile Development

In the wake of COVID and busy senior year schedules, Agile was not used conventionally, as that is mainly used in the workplace where members meet in person daily to go over objectives. Instead, as many aspects of Agile as possible were used, and they were applied to the project to get teammates: one, familiar with Agile, and two, to get work done efficiently.

Of course, project planning, implementation, and testing were all interleaved. To integrate this into our workflow, a weekly check-in meeting was adopted, rather than a daily check-in. General planning was mainly done through the Gantt chart, though this was a rough overview of the project and did not contain specific details. Working on specific tasks was done under the Agile model, where different goals are prioritized for different sprints. For example, for the first semester, or the first sprint, the goal was to complete the planning and get a little bit of the app work done which was accomplished as discussed previously.

Through the application of Agile on a weekly basis, meetings and discussions were held weekly on what the team had previously been working on and what was to be planned for the next week/sprint. During the second semester, weekly sprints were used, where one feature of the app was developed in a week. These features (completed over the course of a week) included developing the daily intake survey functionality, developing the bottom navigation UI, developing the “intake” tab, developing the calendar tab, developing the profile tab, developing sharing functionality, and a whole host of other features. The goal was always to at least get something done and continue to work forward instead of forward and backward that is used by the waterfall method. So, the team was constantly looking forward, even whilst in the middle of something the team would be looking at what section of the app needed to get done next or what was the next senior project objective.

# Development

Below, we can see different code excerpts and their corresponding analysis. The logic breakdown can be visualized soon after in the next section.

|  |
| --- |
| **MainActivity fields and its onCreate() method:** |

|  |
| --- |
| **MainScreen Activity fields and onCreate() method:** |

|  |
| --- |
| **Calendar Fragment key methods:** |

|  |
| --- |
|  |

|  |
| --- |
| **Intake Fragment key methods:** |

## Code (Functions)

In the Android application ecosystem, an activity is a single independent component of an app that generally represents a task that the user can perform. Each activity has a method called onCreate(), which performs the initial tasks for activity creation. OnCreate() is called immediately upon activity creation. Each android app also has an entry point activity, usually referred to as the “main activity.” For Covid Self Report, the entry point activity is called MainActivity.

MainActivity functions as the decider as to what part of the app to launch next. In onCreate(), it first calls the required Android methods (including its super class method), and then calls the updateDirectory() method, which gets rid of old daily intake survey files that are no longer needed. OnCreate() then checks to see if there is a non-empty JSON file for the user’s profile. If it does not exist, it knows to call the ProfileCreator activity, so the user can create a profile. If it does exist, it then verifies that the user has filled out the initial preferences survey, by calling checkForPreferenceSurvey() (this directs the user to the MainScreen if the survey has been filled out already). Then, onCreate() verifies that the user’s daily notifications are set. The point of this activity is to verify that there is a user profile and a perference survey filled out before it allows the user to continue to the main app, which is contained in the MainScreen activity. The full activity and its methods are displayed on GitHub.

MainScreen is the primary user interface activity for the app. Covid Self Report is a bottom navigation app, meaning that there is a bar at the bottom of the app that is used to navigate bettween the different components of the app (calendar, intake, and profile). This functionality is implemented in MainActivity, in the onCreate() method. It uses different Android provided calsses to implement bottom navigation, such as AppBarConfiguration, NavController, NavigationUI, and BottomNavigationView. After initializing the bottom navigation UI, onCreate() calls generateIntakeArray(), which gathers the intake survey files from the last 14 days and initializes them as Survey objects.

Calendar is a Fragment, representing one of the three bottom navigation tabs. In Android, a Fragment is a reusable portion of an app’s UI. It is displayed within an activity. In this case, the Calendar Fragment is displayed by the MainScreen activity when the user has selected the calendar option in the bottom navigation bar. This fragment displays the user’s intake survey results (their daily activity) from the last 14 days, including the current day. Most tasks are done in onViewCreated(), which is called once the Calendar fragment has been initialized. This method gets the current date and the intake survey objects (from MainScreen). It then initializes the title UI components (the headers for each day) and the info UI components (the bodies for each day). Then, the method loops through each title and info components for each day and fills them out with the appropriate information. The onResume() method is called as soon as the user navigates back to the Calendar fragment. It makes sure that if the user just filled out a new daily intake survey, that the calendar gets updated with that survey response. The generateIntakeInfoString() method takes information from the specified intake survey and formats it for being displayed on the calendar.

Intake is a Fragment, representing one of the three bottom navigation tabs. This is the tab that is selected by default. In onViewCreated(), all of the UI components (the buttons, the picture, the text) are initialized. The buttons are initialized and given functionality when they are clicked (implemented by the setOnClickListener() method). The pictures are initialized. In onResume(), the fragment decides the proper buttons and images to display. If the intake was already taken today, it displays a large checkmark photo to indicate that there is no more work to be done. It also shows the “view responses” button and hides the “take today’s survey” button. If the intake was not taken today, it displays a large exclamation point to get the user’s attention. It also shows the “take today’s survey” button and hides the “view responses” button. OnResume() is called every time the user navigates back to the intake tab and is also called upon the Intake fragment’s initialization.

## Testing

After the development of each section of the app (the preference survey, main UI, intake survey, calendar, profile, sharing, notifications, etc.), that specific feature underwent lots of testing. This was mostly UI-related testing, meaning all of the buttons were tested and their actions were closely monitored. With each testing, the team ensured that the survey and activity information were still properly saved into their active Survey object and their long-term JSON file. Testing occurred under a variety of conditions, including a newly created profile, and older profile, etc. Any errors were then fixed.

# Deployment

## APK

The Covid Self Report app is deployed through an APK (Android Package) file. APKs are used to distribute and install mobile apps and software on the Android platform.

Below are the steps to deploying an app to Android through an APK file:

1. Download the APK file to the Android device, or copy it from an SD card.
2. Open the settings app on the device.
3. Open the applications menu in settings
4. Make sure the box next to “Unknown Services” is checked
5. Navigate to the APK file’s location in storage.
6. Go to the APK file to open it.
7. Tap the “install” button to install the application.

## Results

Through the course of this project, from conception to deployment, we see that we were able to successfully incorporate all the use cases we initially desired into our app. We were able to deploy the following features, as visualized in the following workflows.

**Graphical user interface, application

Description automatically generatedA picture containing application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

Intake tab selected after the intake survey is completed.

Question 2 of the intake survey (result of the Take Survey button)

Intake tab selected.

**Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

Share menu after clicking the “Share Profile” button

Calendar tab selected (scrolled all the way up)

Profile tab Selected

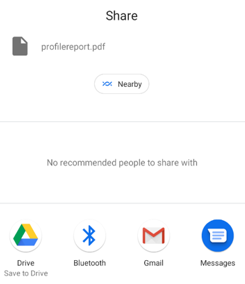
**Sharing Mechanism**

Sharing the profile is an essential function of the app. The world is slowly opening from the pandemic, and the primary reason we landed on the idea of the Covid Self Report app is so people who are considering meeting up in person can share their profile information with others to see if it is safe to meet up. This simplifies communication and ensures people can access what they are comfortable with.

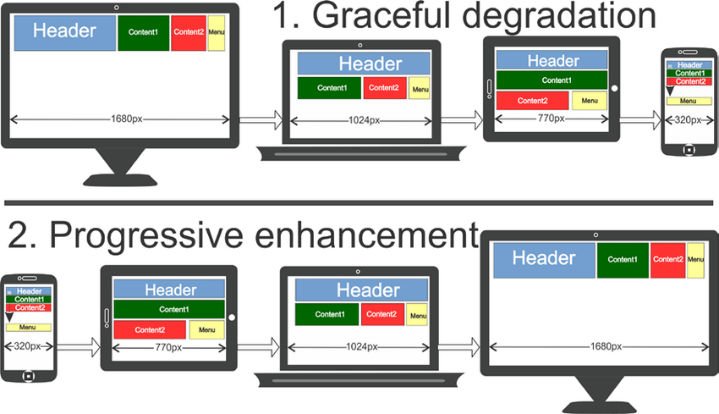
When users share their profiles, they have two options of what to share, as seen in the middle screenshot. One option is to only share their preferences, which is the initial survey they filled out when they downloaded the app, showing their long-term preferences and what they are comfortable with. Option number two is to share both their preferences and their intake surveys, which shows their activity from the previous 14 days. Once the user selects what they want to share, a PDF report of their profile is generated, tailored to their selected sharing option. As seen in the right-hand screenshot, user has a variety of options as to how to share their profile, such as via text message, Google Drive, or email. Our app provides a flexible way for users to share what they want, and how they want.

## Future Considerations

Considering the status of our project and the user feedback, we would like to work further on the development of a sharing mechanism. Currently, we generate a pdf from the applications cached memory, which is then shared to other users through various outlets. As seen in the figure below, we currently support Google Drive, Bluetooth, Gmail, and Messaging capabilities to share our profile pdf with others.



However, such a concept is not scalable due the complexity in which the profiles will grow, as well the process of generation and deleting the pdf files. To reduce this redundancy, we intend on migrating to a SQLite integration. Through the integration of this built-in database, we can parse through user inputs and copy this information to clipboard. We can also parse more efficiently and store the inputs of more user in a more efficient manner, making use of relational databases to navigate user inputs.

Finally, we would like to continue our development using the “Progressive enhancement” app development and deployment scheme. Through this process, we can gradually scale our android application approach into web development. Currently, our android application can only be deployed onto android devices. By creating a web application that can be accessed by any device, we remove the any device-specific restrictions previously placed by our development. They can also be accessed by more devices, not just phones and tablets. It would also reduce our reliance on Android Studio Maker for development and allow us to make use of other open-source software, such as Visual Studio Code.

# Conclusion

To conclude, this project was aimed at learning the software development processes, specifically using the agile development model. Agile was successfully used to develop a fully functioning software application.

Of the people who have used and tested the app, they were all satisfied. These include us, the developers, and our family and friends who installed the app on Android devices. No issues were encountered, and the application was easy to use and fully functional. This application provides the activity information and contract tracing functionality that that it set out to provide, while also allowing for a solid user experience. We were able to address and succeed in our vertical to service those impacted COVID and create an application that would augment contact tracing, while still facilitating in-person meetups.

# References

“Activity  :   Android Developers.” *Android Developers*, Google, developer.android.com/reference/android/app/Activity.

“Fragments  :   Android Developers.” *Android Developers*, Google, developer.android.com/guide/fragments.

Jablonski, Kyle. “Android Notifications Tutorial: Getting Started.” *Raywenderlich.com*, Raywenderlich.com, 22 Apr. 2019, www.raywenderlich.com/1214490-android-notifications-tutorial-getting-started.

Khorakhun, Chonlatee. *Using quantified-self for future remote health monitoring*. Diss.

University of St Andrews, 2015.

Murphy, Mark L. *The busy coder's guide to Android development*. United States:

CommonsWare, 2008., 2008.

Pendse, Prachi, and Jalisa Greene. "A wellness android application with social networking

capabilities." *Proceedings of the 51st ACM Southeast Conference*. 2013.

Shekhar, Amit. “Learning Android Development In 2020 - A Practical Guide.” *Medium*, MindOrks, 2 Jan. 2021, medium.com/mindorks/learning-android-development-in-2019-a-practical-guide-ddc71e008696.

Vogel, Lars. “Android Application and Activity Lifecycle - Tutorial.” *Vogella.com*, Vogella.com, 7 Dec. 2017, www.vogella.com/tutorials/AndroidLifeCycle/article.html.