Detailed Design:
The At-Home Covid Test with Smart Capabilities

Spring 2022

Group-8

Presented by: Saul-Leal Garcia Justin Phuong Mark Martin
Damandeep Singh Dean Quach Can Chai Jared Mallari

Table of Contents

1. Document Overview

We are pleased to present our progress on our covid testing device. This document includes how we used our design models to create our device, user stories to understand our product and its functionality better, and various diagrams with vital information in a visual format. For our hardware side of the project, we went more into detail on the more crucial requirements for these steps. We decided to take a waterfall approach for the hardware side of our project to produce accurate and reliable hardware. The agile approach is adopted to develop our software system, mainly the application and website, to deliver a user-friendly interface for our customers. A requirements traceability matrix provides the relationship between design requirements and testing. In the end, different testing approaches are discussed to test the whole system.

2. Requirements Overview

2.1 Detailed Requirements (Waterfall)

We employ the waterfall methodology to process our hardware design of the covid testing device. One of our more critical requirements will be for our device to accurately detect covid on those who use our device. Our device must be similar in use to other tests currently on the market in which our test must be simple for our users to administer. As a competitor product, our test must produce accurate results within 10 minutes as one of our selling points is that it can be used in airports before flights. Another selling point of our test is that it must be able to tell users the variant of covid that they have if they have a positive covid result.

Our device's ability to connect to users' smartphones sets it apart from other covid tests that are currently out on the market. Compatibility among smartphones is crucial, so the device must have the Bluetooth 5.0 technology so that most smartphones can connect to this device. Our device will have the ability to send positive or negative test results to users' smartphones through an app downloaded through the Apple store or the Play store. Our device will not rely on wifi to be able to send data to a user's smartphone, as this will defeat the purpose of using our device on the go. The device should have three 5mm LED lights namely red for positive results, green for negative results and blue for Bluetooth status.

Our device will use built-in batteries as a power source to the device. Special measures for this step as using too many batteries will increase the price and the environmental impact. Our system design is eco-friendly with minimal wastage with the testing swabs made out of cotton, and minimal plastic usage is enforced for the handling of the swabs. Despite the type of material, our primary focus is to deliver a robust hardware device capable of taking covid tests and sending the results to an app/website.

We identified requirements that are mandatory for the hardware component of our system. Here is a list of detailed requirements for our covid test hardware component:

Req ID	Requirement Description
1	The device should have built-in six silver oxide batteries (1.55 volts) as a power source to the device
2	The device will have on and off buttons to enable and disable the device
3	The device should have a bluetooth range of upto 800 ft
4	The device should turn off after 5 minutes of inactivity
5	The red LED light should turn on with positive test result
6	The blue LED light should flash unlimited times until a device is paired.
7	The green LED light should turn on with negative test result
8	The device should connect to only one end-user personal device at a time through Bluetooth
9	The device should take no more than 10 minutes to process covid results
10	The device should send a matching 5 digit pin to ensure the connection is established between the user's choice of personal device and the covid device

Table 1. Detailed Requirements of Covid Device

2.2 User Stories (Agile)

Our second priority is to create reliable and easy-to-use software that users can navigate through quickly, and we describe the interactions users will have with our software through a few complete user stories. We took an agile approach for the software side as this would allow us to adapt to any change that might occur during our software development.

2.2.1 User Story 1

Bob is a new user and have recently completed a covid test from the hardware component.
Bob can go into google play store or apple store and download our application. Bob should also choose whether this app can access mobile device data or not when he enters the app for the first time. After installation is complete, our app will display options such as if they are a new user or an existing user. If Bob is a first-time user, he should press on the “New User Registration” button, otherwise the “Existing User” button. Bob will be in our U.I. when registration is complete, which will display options such as connecting the device via Bluetooth, sending test results to authorized health providers, government and non-government institutions, editing profile, and history of current/previous covid tests. Once Bob connects to the covid device via Bluetooth, a 5-digit pin will appear on the U.I to confirm that the correct device is requesting to connect with the user. Once confirmed, Bob will have access to the analytics, heatmap, and news tabs. The covid result data will transfer to the app, and under the analytics tab, a personalized covid results report will be generated, which includes: name, address, social security number, phone number, home address, test result, and date/time. Under the same analytics tab, Bob may send the data to their requested institutions using either institution’s name or use the unique institution code and plan to send their data within 24 hours of the report generation.

2.2.2 User Story 2

Knowing recent covid outbreaks in the neighborhood and outside the radius of the local area is difficult to visualize and know the severity of these outbreaks. Using the heatmap feature available on the app/website, the user may get to know about the severity of recent outbreaks based on their zip code. Under the heatmap tab/page, the user may look to input their zip code (if they have not already) and click the “Submit” button to generate a live heatmap of covid cases based on the zip code provided by the user. The user may use the “radius” slider to increase the area coverage of the heatmap showing the covid cases around the area with maximum of 100 miles.

2.2.3 User Story 3

Sending covid results to institutions can be challenging at times when covid tests results are immediately required. The user may visit the analytics tab/page and check their results sent from the hardware component and look up the institution to which the user wants to send their personalized results. The user may search “institution codes” or search the institution by their name. Once the institution is selected, the user is prompted to enter SSN and review their information. The user may select the “send” or “cancel” option upon their personal preferences.

2.2.4 User Story 4

Receiving localized outbreak-related news and alerts is helpful for the user to prioritize their outings. The user may visit the news tab/page and check their local news regarding any government and CDC related actions/alerts and updates.

3. Design Overview

3.1 Design Specifications

The design of the at-home covid test device is split into two subsystems. The first subsystem is the covid test hardware component and the second subsystem is the application/website software component for users to interact with their respective data interested parties. Here is an abstract level use case diagram that describes the interactions between the user and different actors:

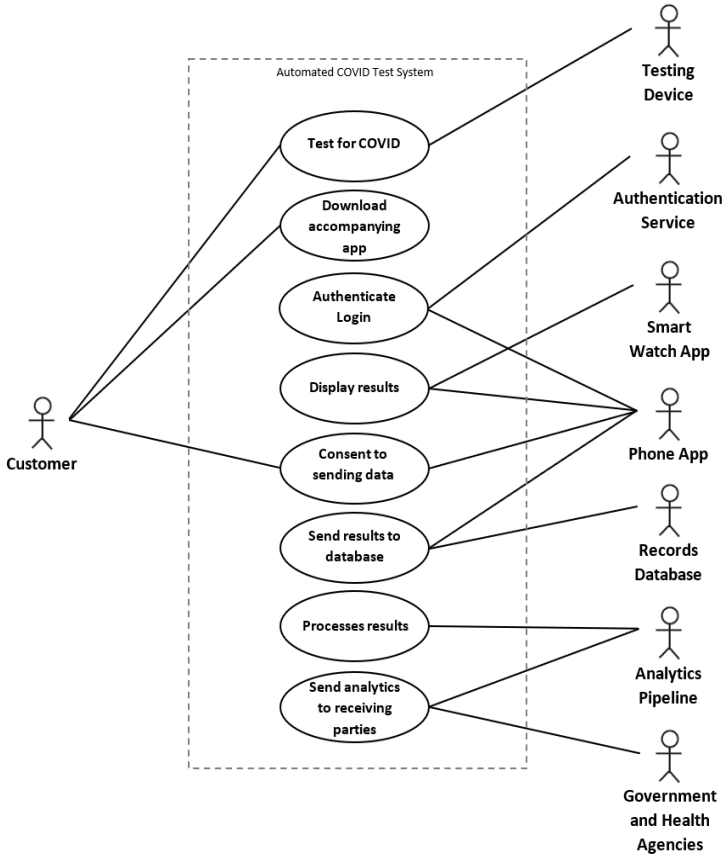

Figure 1. Use Case Diagram of Whole System

The following sequence diagram explains the detailed interaction and procedure of how the whole system will work in a stepwise manner with the integrated system (subsystem I and subsystem II):

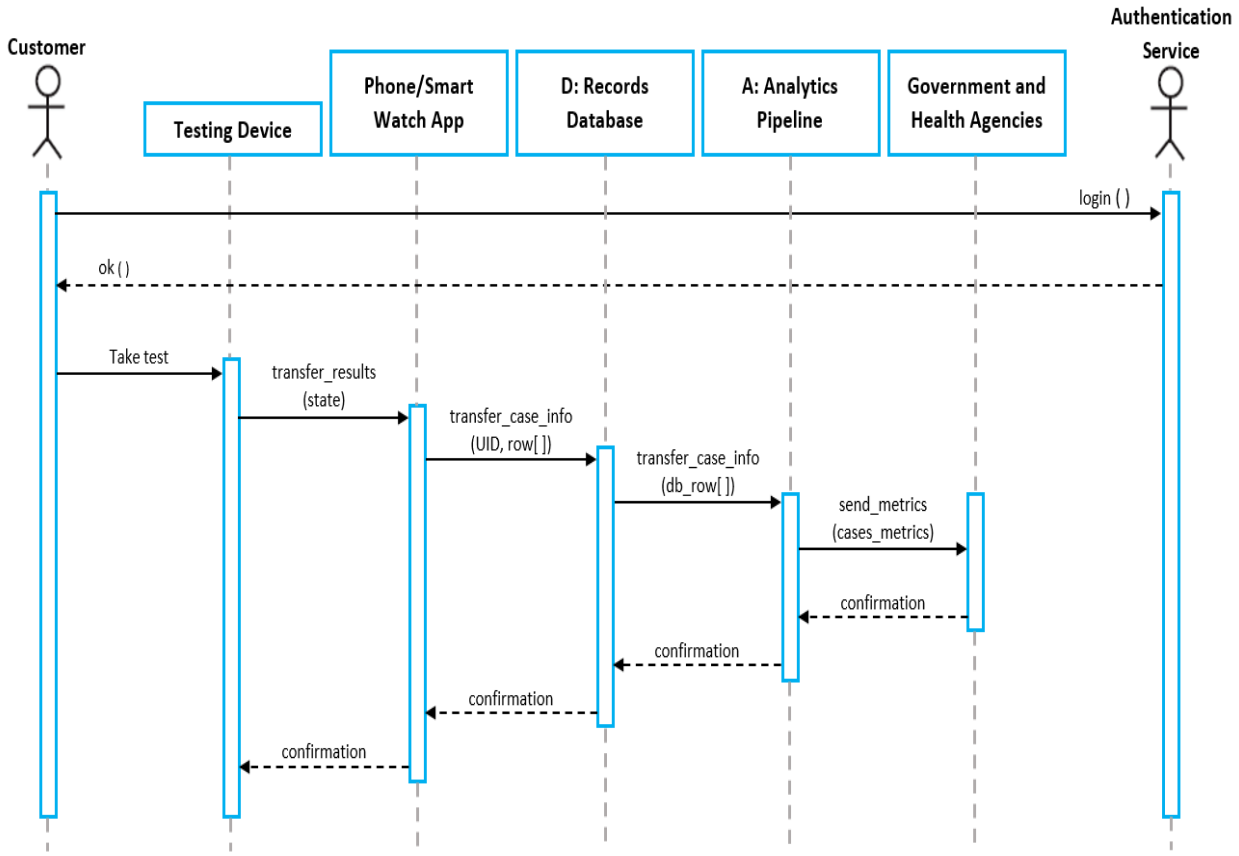

Figure 2. Sequence Diagram of Whole System

3.2 Website/App Design

Our project prototype relies on the creation of an easy-to-use user interface for our customers. The main functionality of the application/website is to provide the user with capabilities to personalize their covid results along with the privileges to interact with institutions requiring immediate results of covid tests.

3.2.1 User Interface

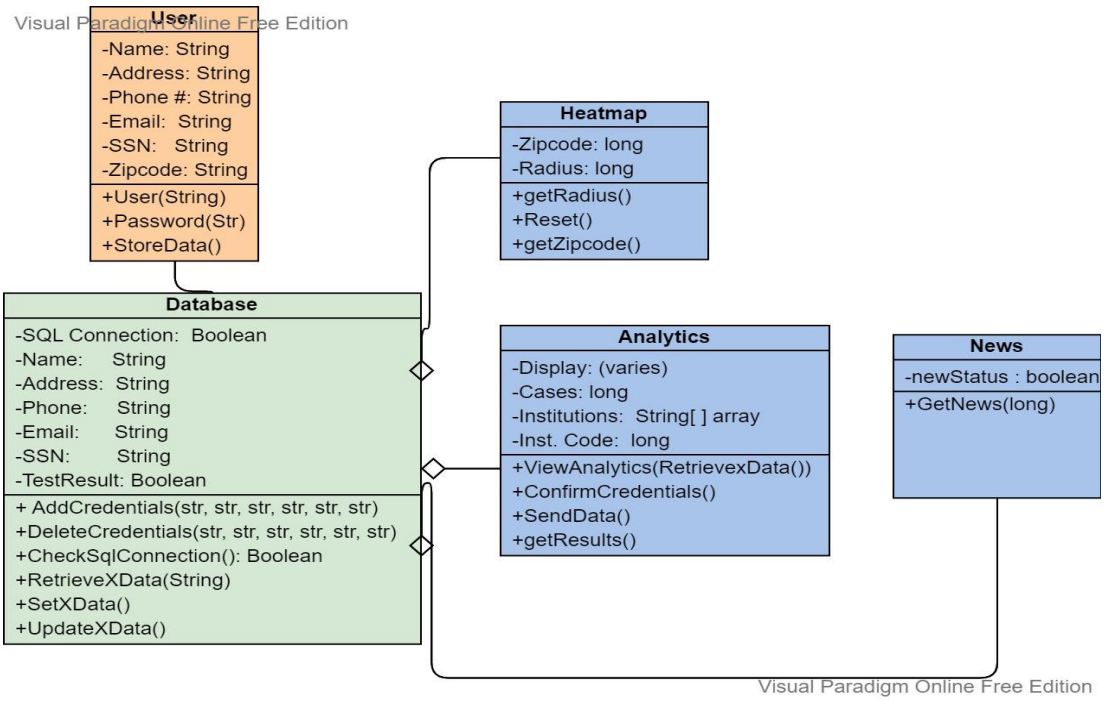

Figure 3. Class diagram of the app/website

The app and the website will consist of the four classes: user, heatmap, analytics, and news. The user interface of the app/website is similar in layout and functionality for all users. A relational database is used to store and retrieve user data to execute requested operations.

User Class

The purpose of the user class is to provide the registration page for the new users and allow them to add their personal information such as name, home address, phone number, email, social security number and zipCode (optional). The user class will communicate with the relational database SQL and execute CRUD operations to validate the data into the system.

Private variables: name, address, phone number, SSN, and ZipCode (all string data type)

Public methods:

User(string): Automate a username creator or allow user create their own
Password(string): Automate a password generator or allow user to create one
StoreData(): communicates with database to store the user input into a table

Heatmap Class

The purpose of this class is to display a heat map according to the area's zipcode. The heatmap class will use some current or previous datasets present online through the Johns Hopkins covid resource center website. We use the streamlit library of the python programming language and generate a heatmap as per the data. The map will allow users to know the high risk and low risk areas around their neighborhood. The user can also increase the radius of their area using the radius feature. An additional feature of reset is provided in this class to allow the user to reset the heatmap.

Private variables: radius: long and zipcode: long (in case it is not provided)

Public methods:

getRadius(): communicates with the database and gets radius input by the user.
Reset(): resets the current heatmap (a void function)
getZipCode(): communicates with the database and gets zipcode input by the user.

Analytics Class

The purpose of the analytics class is to allow users to access, view and send their sensitive data such as the covid test results. This class will communicate with our database to retrieve the test results that are provided through the bluetooth capability of the at home covid test component of the whole system. The class will also have a list of institutions (government/non-government) that the user may want to send their data to along with a unique institution code to ensure the data is sent to the selected entity.

Private variables: Display, Cases, Institutions, codes (varies, long, string[] array, long)

Public methods:

ViewAnalytics(RetrieveData()): Displays the covid results along with the personal information

ConfirmCredentials(): A void method that will require user to input their username/password again for security purposes.

getResult(): This function will display the test results to the user (void function)

SendData(): We will create a hashmap of where keys are institution codes and values are institution names for a faster lookup functionality. This function will also carry the ability to send data to the institutions through an email (we assume institution emails exist in our database) while invoking the getResult() and RetrieveData() methods for a report generation.

News Class

The news class aims to display covid results related news according to their zip code and recent CDC and government updates on the mask and covid surges. The news class will notify the user regarding covid surges in the neighborhood.

private variable: zipcode

Public method: getNews(zipcode) → calls the zipcode function from the database and the google news API to display local news updates and outbreaks.

Database

The database plays a critical role in development of both hardware and software components of this system. We are planning to use a relational database to store all the necessary variables and features mentioned in class diagram of the app/website. The schema for the database will have the same variables listed in Figure 3 and mostly all the variables will be private and abstract to add a security layer. The database will have many functions for the data:

- CRUD operation queries for basic create/read/update/delete the structured table with variables of the multiple class diagram
- SQL boolean connection output to ensure its communicating with the app/website

The combination of the four classes and the database work interchangeably to provide an easy-to-use interface for the customers interested in sending their covid data to the interested parties. The user may have a privilege or choice whether they want to send their data to the interested or wish to keep the data on our database for future references. Another requirement for our database is to authenticate the user details if the user already exists in the database. We will have a SQL query that may search for the social security number in the database and look up the matching name input by the user at the time of registration. If the existing user's data does not match, the user will not proceed to the next step of accessing the dashboard.

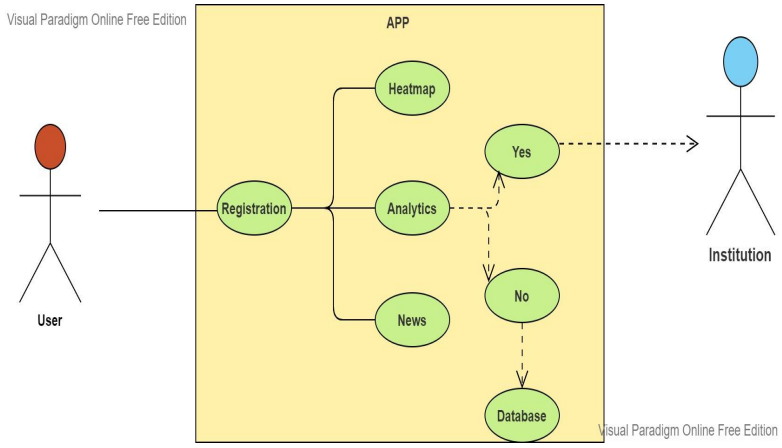

Figure 4. Use Case Diagram of App/website

The user will have simple steps over the app/website component. First, the user must register with their personal information to allow the successful registration and transfer of data over Bluetooth. Second, the user can select one of the three options: heatmap, analytics, or news. Third, the user can choose to send their covid test results to the institutions or avoid sending their covid tests results; however, the results are stored in our database for quality purposes.

3.2.2 Technology Stack:

One of the primary goals for our project is to develop a web application and an app to promote personalized interaction between users and the data interested institutions. Both website and app will have similar functionality and layout as both software components will have the same features/variables and methods to attain a standard level.

The whole website will operate on the HTML5 and CSS3 technologies. In addition, many front-end and frameworks are used to develop a vivid and easy-to-use user interface for our customers.

Here is a list of technologies we are going to use to develop our website:

- Frontend technologies: HTML5, CSS3, Animation.CSS, and Bootstrap
- Backend technologies: Javascript/EJS, React.js, Node.js, and AWS Relational Database Environment (SQL)
- Heatmap development: Streamlit library available on python 3.x to generate a heatmap
- News: Google API

Here is the technology we are going to use to develop an IOS app:

- IOS app development platform: xCode using the Swift programming language

Note: We will expand our app support to android devices because we notice apple users are dominant in the U.S.

3.3 Hardware: The Covid Device

The hardware device will have the Bluetooth 5.0 transmitter (Tx) to send covid results over to the device. The device will have on/off buttons to turn on and off the device. The device will have three 5mm LED lights: red light (positive result), green light (negative result) and blue light (Bluetooth status). The device will have a place to insert tested cotton swabs for testing purposes. The detailed requirements are also mentioned in the 2.1 section. Since, waterfall

methodology is used to design this covid hardware design we will have the chances to accommodate changes earlier into our design as new covid variants emerge.

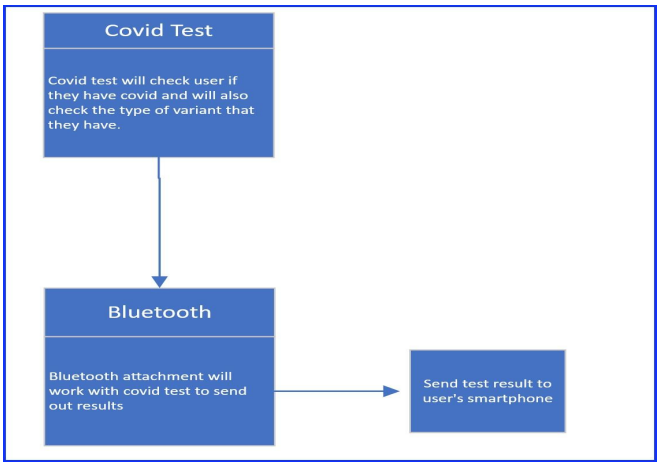

Figure 5. Class diagram of the covid device

The covid test device is a physical component, and the communication between the covid device and the software is critical to ensure results are communicable to the right person. We have taken measures to ensure that only one device simultaneously communicates to the covid device. We have enabled a 5-digit pin confirmation method on both the user's device and covid device to ensure the suitable device is connected and no other Bluetooth device is connected to the device and receiving any results. The device will have an embedded timer not exposed to the user to allow the device to reset the device after 5 minutes of inactivity. The device may have a BlueTooth connection range in the covid device to the personal device of 800ft.

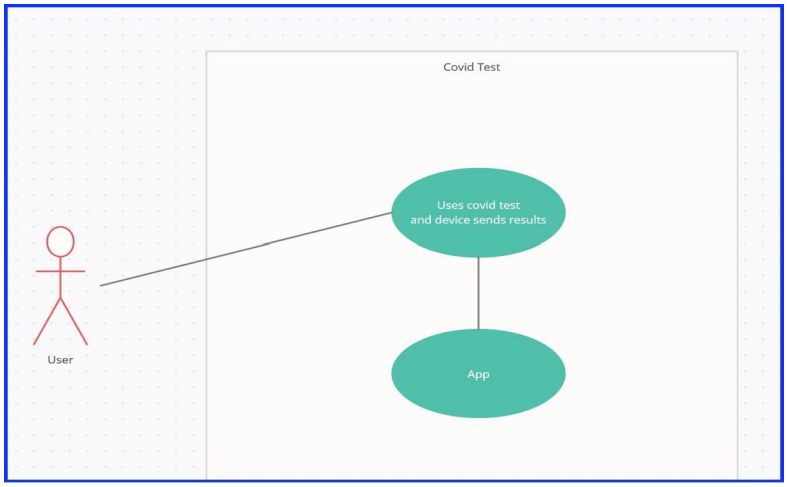

Figure 6: Use Case Diagram of the Covid Device

For the covid device subcomponent, the interaction between the user of the covid test device and the app or website will allow the user to render their results to interested parties. There is only one actor to interact with the hardware covid device to get their results.

4. Requirements Traceability Matrix

We will use forward traceability to map each requirement with test cases and ensure each component is tested thoroughly. It allows the developers to recognize and test for essential components and decide on the root causes of the failing test cases.

Req ID	Req Desc.	Team	Test ID #	Sub Test ID #	Test Strategy	Status	Pass/Fail
1	Covid Test	Doctors	001	TC#001A	Random testing		
2	Schema	Database	002	TC#002A	Cross validation	✓	
3	Bluetooth	SDET (Wireless)	003	TC#003A, TC#003B, TC#003C	On/Off, range, signal strength, connectivity etc.		
4	App	SDE	004	TC#004A TC#004B TC#004C	Functional, working modules, different devices	✓	
5	Heatmap	Data Scientist	005	TC#005A TC#005B TC#005C	Dataset, Radius slider and accuracy		
6	Website	Web Developer	006	TC#006A TC#006B	Different browsers: font, text, color, layout, performance		
7	Database	Database	007	TC#007A TC#007B	CRUD operations, Authenticattion		
8	On/Off (hardware)	Hardware	008	TC#008A TC#008B	Turn on/off and inactivity of 5 mins	✓	
9	Voltage (hardware)	Hardware	009	TC#009A	Voltage meter (~1.55V)	✓	
10	LED (hardware)	Hardware	0010	TC#0010A	Test +, - and bluetooth status	✓	

Note: Some hardware requirements are condensed into one as they fall into the same category for testing purposes

5. Test Plan

The testing of our whole system will revolve around the four testing methods to ensure our system is error-free when our customer is ready to take their covid test. Overall, we may test for various functionalities with some of the high-level testing requirements and procedures, which includes:

- Exploratory testing will be will be executed once build is ready
- Test team will be provided access via VPN
- Test case design will be performed by a QA team
- Defects will be tracked through Atlassian, Jira and given to test team
- Performance testing is considered in later cycles of testing (Q2, Q3)
- Testing will be focused on meeting the business requirements, quality, and cost efficiency
- Testing will be a repeatable, quantifiable, and measurable function of activity
- There will be an entrance and exit criteria
- Testing will be defined but have room for flexibility depending on changing requirements

5.1 Unit testing

We will perform unit testing on both the software and hardware components of the system. This is our first stage of testing. For our software component, we will use the following functions to test the integrity of our app and the website:

- func setUpWithError(): checks for valid characters or inputs from username, password, address, social security, email, zip. Exits upon invalidation returns false
- func validateResults(): checks for valid results based on known COVID test criteria
- func checkUploadTarget(): checks for valid targets to upload results. Based off of known health networks and facilities
- func checkDownloadData(): checks for valid downloaded data from database and ensures complete information with no null values
- func isRepeatedPassword(): checks if the password is repeated correctly upon creation

For our hardware component, which is the covid device, we will test for the multiple units or sub-components of the system using the following functions:

- Manually check ON/OFF buttons to ensure the device shows the expected behavior.
- Check Bluetooth range of the covid device:

    - Checking the Recieved Signal Strength Indicator (RSSI) value of -40 to 55 and no less than -70 to ensure a strong connection between the covid device and the personal device.
    - The RSSI value is available under bluetooth option of many smart devices.
- Check voltage of batteries should have ~1.55 volts/battery using the electronic volt meter.

5.2 Interface Testing

The interface testing will mainly focus on the bluetooth and database connectivity of hardware and software side.

- Testing between the web application and mobile application interfaces will be conducted
- Testing between the database and web application interfaces will be conducted
- Testing between the database and the mobile application will be conducted
- Testing will be done to ensure strong network connectivity between database and application environments
- Interface between devices and COVID test kit will be thoroughly tested for connectivity, response time, and accuracy
- Bluetooth connectivity between the device and the app to result with higher RSSI value of -40 to 55.

5.3 System Testing

Testing the combination of subsystem I and subsystem II requires integrating the two system and executing a full scale test on the whole system and expect desired results.

- Testing of usability of the interface for users will be done
- Testing under load stress with thousands up to hundreds of thousands of users will be tested for capacity limit testing
- Testing for network connectivity between interfaces will be conducted
- Crash testing will be conducted to ensure software is fully functional after any hardware disfunction or software crashes

- We will be using the Eggplant software to seamlessly integrate CI/CD pipeline and compress testing cycles to allow for quicker releases

5.4 Release Testing

Our last stage of testing to identigy any bugs and errors in the software and hardware component and make sure the product is ready to reach our end users.

- A suite of release testing programs will be running before release to ensure that high priority bugs are non existent in the program

- Specific release tests include working notifications to the user incase their data upload went wrong, invalid information on user login or registration, and inability to download data from database

- Testing will also be done before release for proper encryption and decryption of user information

- Stress testing will be incorporated to monitor performance and ensure we can account for 1000 downloads/transactions per second

6. Schedule Update

Tasks	Assigned To	Start Date	End Date	Status Summary
Planning				
Develop Schema	Mark	02/23/22	04/06/22	
Build Cloud Database	Daman	02/25/22	03/31/22	
Manufacture Prototype	Saul	02/14/22	03/03/22	
Software Development				
Develop Landing Page	Chai	05/04/22	05/28/22	
Develop Login and Terms of Agreement	Chai	05/20/22	06/16/22	
Develop Registration	Dean	04/29/22	05/11/22	
Develop Profile	Saul	05/02/22	05/27/22	
Testing				
Unit Testing	Jared	06/01/22	06/08/22	
Interface Testing	Justin	06/07/22	06/15/22	
System Testing	Jared	06/14/22	06/23/22	
Release Testing	Justin	06/22/22	06/29/22	
Marketing Campaign				
Begin TV Marketing	Daman	07/06/22	07/28/22	
Begin Internet Marketing	Chai	07/13/22	07/30/22	
Final Product Rollout				
Begin Finalizing Packaging	Mark	11/02/22	11/04/22	
Finalize Manufacturing to Global Market	Saul	11/09/22	12/02/22	
Create Business and System Plans	Mark	12/01/22	12/30/22	

Figure 7. Schedule update of the project

We are on track to implement our hardware and software components of this project. So far, we have completed 5 tasks related to the initial stages of the projects as mentioned in the Figure 7 of this document. In coming weeks, our team will transition from starting implementations of the website or the app to following a rigorous schedule of testing which will last 1-2 months of time to ensure a robust system is created for our end users and customers.

7. Conclusion and Next Step

In conclusion, our project will help make COVID tests much more convenient, allowing us to test and share our results with ease. When going through the testing process, we ideally try and put the software through various situations that may “break” the system, further making adjustments to ensure a functional U.I. We plan to release this project for the public while still evolving the system to meet new requirements.