8-2021

APPLICATION OF REAL-TIME DETECTION SYSTEM IN EPIDEMIC PREVENTION

Haoyu Wang

APPLICATION OF REAL-TIME DETECTION SYSTEM IN EPIDEMIC PREVENTION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information System and Technology

by

Haoyu Wang

August 2021

APPLICATION OF REAL-TIME DETECTION SYSTEM IN EPIDEMIC PREVENTION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Haoyu Wang

August 2021

Approved by:

Conrad Shayo, Committee Member, Chair

Jay Varzandeh, Committee Member,

Chair, Information & Decision Sciences Department

ABSTRACT

The number of confirmed cases of the covid-19 virus have skyrocketed since November 2019. Now researchers are beginning to realize that due to the continuous change in the mode of virus transmission, it is too slow to rely on outdated news from the government. This project focused on solving the following question: How can we use a cost effective real-time regional diagnosis to decelerate the speed of virus spreading and save more lives? This project developed a real-time monitoring system model that had four main modules: (1) the data receiving, processing and transmission module, (2) the real-time analysis module, (3) the remote-control module, and (4) the mobile terminal module. The real-time monitoring system collected confirmed cases in specific areas (e.g., Fontana area) and rendered them through a mobile terminal module. Through the data receiving, processing and transmission module and the real-time analysis module, people in the region are reminded not to crowd too densely, resulting in accelerated virus transmission. The combination of the remote-control module and mobile platform module dynamically obtained the diagnosis position, without setting a fixed sampling point. The main conclusion is that compared with manual sampling, the cost of a case monitoring system was more effective, and the monitoring results were more accurate. Future studies should explore more effective prevention of the virus. For example, the development of facial recognition systems can remind users to wear masks when unlocking mobile phones.

ACKNOWLEDGEMENTS

I'd like to convey my sincerely gratefulness and appreciation to my supervisor Dr Conrad Shayo, who supported me in completing and revamping my project. Thanks to his help, which led me to finish the project on time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1. Overview

In this chapter, the author identified, summarized the issues addressed by the project, including the issues that the system should figure out, the solution of the problem, the scope of the problem being solved, and the questions that will be answered.

1.2. Problem Statement

On December 31, 2019, some unexplained pneumonia cases were found in some hospitals in Wuhan, Hubei Province. On January 7, 2020, Chinese officials confirmed a new type of coronavirus as the cause and named it "COVID-19" (World Health Organization, n.d.). Having a fever, cough, dyspnea, fatigue, along with malaise are its main symptoms. Most people found that the disease is slight. (Pediatr, 2020). Multiple new researches along with systematic reviews have been published, and there is evidence to show its potential for transmission via several routes including respiratory droplets and close-contact, vertical, conjunctival and fomite transmission. (Public Health Ontario, 2021). Moderate to low fever were found in critically ill patients, in some cases even no significant fever were found. Low-grade fever, slight fatigue with no pneumonia were found in patients who were slightly affected.

On Monday (April 12, 2021), the United States arrived at an astonishing data. In the pandemic that lasted for nearly one year, the number of known coronavirus-related deaths exceeded 500,000 (The New York Times, 2021). The spreading speed of the COVID-19 virus in the United States is much faster than expected. The virus has taken thousands of lives in the world. The virus can be easily spread among human beings. Due to the different stages and severity of the epidemic, the details of open country vary from place to place. Therefore, we still need to pay attention to the spread of the virus.

In the early days of a pandemic or open country, people tend to ignore the importance of virus prevention. If too many people are infected with the virus, it will speed up the spread of the virus. The following problems are many Americans face during the pandemic:

Lack of Detection System: One of the problems is that residents lack virus detection software and don't know if there are people with confirmed cases within the six feet recommended distance. Please refer to the comparison table in chapter 1.3. Such alert applications will help people to protect themselves and avoid unnecessary inhalation of the Corona-19 virus.

Lack of Unified Vaccine Management: Although each health center or vaccination site has its website. There is a lack of a unified management system. If users want to get vaccinated, they need to open navigation to find the nearest site. If the appointment is not successful, they need to find a new site again. If there are too many people queuing for vaccination, users may need to waste half a day. An Online queuing system can improve this problem.

1.3. Proposed Solution

The purpose of the proposed system is to provide accurate information of confirmed cases for people. Many applications don't monitor the number of confirmed cases in a particular area. For example, the application “COVID-19!” and “COVID-19 Tracker”, confirmed data is available for only one state. and it has been shown to lack accuracy within six feet. Moreover, it doesn't provide an alarm when there's a confirmed case close to 6 feet (My COVID-19 Tracker, n.d.). Some applications such as Apple Map only provide vaccine appointments, but there is no online queuing system. This is inspired by the restaurant online system. It can also be used in vaccination queuing systems. This may include users who cannot tell if the area is too crowded. The system will also help users find the nearest doctor or vaccination site in any emergency. Below are the proposed solutions that will help to reduce the speed of virus transmission:

Monitoring Area Density: Nowadays, almost everyone has a smartphone. The system can collect signals from smart phones in a certain area to judge whether the area is too crowded. Users can enter a zip code or use GPS to see the population density of the area. The system will automatically give a warning whether the area is a high-risk area by collecting the number of confirmed cases in the area and whether the confirmed cases are still in the area.

Monitoring Confirmed Cases: The system can automatically obtain the number of confirmed cases from the information released by the nearby hospitals and local governments on the Internet. The hospital can upload more accurate diagnosis case information to the system, but pay special attention to anonymity, not to disclose the patient's name or any other personal identifying information (PII). Users can also upload the diagnosis information to the system according to their own situation. However, the authenticity of the information uploaded by users' needs to be validated. So, the information uploaded by users is listed separately for reference only.

Monitoring Vaccination: With the advent of vaccines, many CVS or pharmacies have vaccination sites. After covid-19 vaccination, a record will be sent to the vaccinator. Users can upload records to the system (please be anonymous). The system will automatically count the number of people who have been vaccinated and record relevant information.

Find the Nearest Vaccination Site: The system helps users to find the vaccination sites that accept the reservation request. This can be done by entering the postcode of the user's residence to display the vaccine outlets in the area. In the future, automatic navigation can be realized by connecting with apple maps or Google maps. This will help users to find outlets more quickly and conveniently.

	COVID - 19!	COVID-19 Tracker	COVID alert	Apple Map
Find nearest vaccination site				✓
Monitor vaccination		✓		✓
Monitor Area Density	✓		✓	

Table 1. System Comparison

1.4. Project Scope

The project is an open-source application for people who are willing to make suggestions to improve the application. The application allows users to input their credentials and upload their information anonymously through authentication. Users can make an appointment for a choice vaccine and find open locations with shorter vaccination lines. To effectively get more people vaccinated, prevent the spread of the virus, and protect life.

CHAPTER TWO

REQUIREMENTS AND SPECIFICATIONS

2.1. Chapter Overview

This chapter addresses: (1) the three types of end users: Admin, Application users, Health Center Staff, (2) describes the requirements/ specifications of the Real-Time COVID-19 Detection system in detail, and (3) displays a list of system specifications. These requirements include functional and nonfunctional requirements that the system must provide.

2.2. End Users

Three forms of users were shown as following:

Administrators: They ‘ll manage maintenance, updates, adding along with deleting users, finding users, increasing detailed, confirmed information, analyzing feedback. Make suggestions to customers by analyzing data.

Application users: They are users who can access all the functions of the system, including finding vaccination sites nearby, real-time population density, etc. Users can choose to upload their own data or not.

Health Center: They are the main users that provide scientifically certified data. They can provide the system with the number of confirmed cases and the number of people who have been vaccinated.

2.3. System Requirement Specification

The appendix lists and cites a list of all functional and nonfunctional requirements. The table contains a set of structured information which reflects the real-time inspection system's requirements.

2.4. Actors and Use Cases

Actors	Semantics
Real-Time Detection System Admin	The administrator of the system. Maintain, update, add and delete users, search users.
Application Users	The main users of the system. All functions of the system can be accessed. And can upload personal data.
Health Center Staff	The main source of confirmed case data. Data can be uploaded to the system for users' reference.

Table 2. Actors

CreatAccount	Create an account for everyone who wants to use the systen Including users and Health Center Staff.
UserLogin	Users log in the system to find information.
FindVaccineSite	Find the nearest vaccine site by entering the zip code or usir

	GPS.
AppointmentFor Vaccination	Make an appointment at the nearest vaccination site. Time a name required.
ManageAppointiment	Cancel or change the vaccination time.
GenerateReport	The system gives reasonable suggestions based on big data analysis. Red is high risk. Green is normal.
UploadUserInfo	You can choose to upload your own data. You need to distin between organizational users and individual users.
UpdateConfirmationCa	Update the confirmed cases everyday if possible.
OnlineQueuing	Can line up online. No need to line up offline
UserLogout	Users quit system

Table 3. Use Cases

2.5. Modules

The modular design method divided the system into smaller parts that were called modules, which is easier for any stakeholders to use the system. This application has three important modules, the sub modules are as follows:

2.5.1 Real-Time Detection System Admin Login:

Add Vaccination sites: The system allows administrators to add details of vaccination sites to the system, which can provide information for users. The administrator can edit the risk degree according to the corresponding information.

View Users: Can view and edit users' profile when users request. All registered users can be viewed. Personalized reports can be generated when users need.

View Feedback: Can view feedback from users.

2.5.2 Users Login:

Register: Users should register and provide basic information before using the system.

My Details: Users could add personal information for further use to the system. Username, frequent activity area, diagnosis and vaccination were included.

Check Nearest Vaccine Sites: Users can query vaccination sites according to GPS or input zip code. And you can check which sites reserve vaccination.

Appointment for vaccination: Choose the time of vaccination according to the time.

Modify appointment: Modify the appointment. Including modification time and place.

Cancel reservation: Cancel the corresponding appointment.

Check current region status: Test the number of confirmed cases, vaccination, and population density in the current area.

Check Number of confirmed cases: Detection of confirmed cases in the area. It is divided into personal upload data and organization upload data.

Check Vaccinations: Monitoring the number of vaccinations in the area.

Check population density: Monitor how many active users in the area are receiving signals from smart phones. If the density is too high, the red high-risk area is displayed.

2.5.3 Health Center Login:

Upload Confirmed Data: Upload the number of confirmed cases to the system. The system automatically marks the data and uploads it to the organization certified by the government. It is authoritative.

CHAPTER THREE

SYSTEM DESIGN AND ARCHITECTURE

3.1. Chapter Overview

Discussion about the design and analysis phase of the system were shown in this chapter, including use case diagram along with ER diagram.

3.2. System Architecture & Design

The Real-Time detection system is on account of a three-tier architecture. Every functional module was separated to 3 layers: the first one is presentation layer, and then business logic layer along with data access layer. Interfaces are used for access between the layers, the whole classes of the object model are used as a carrier of data transmission. The whole classes of object models correspond to distinctive tables of the database, and the attribute of entity class is the same as the field name of database table to you.

This three-tier architecture can divide the specific development work of technicians and developers in the process of software development, pay attention to the analysis, design, and development of core business systems, improve the quality and efficiency of information system development, and provide great convenience for the future update and maintenance of information systems.

3.3. User Interface

Real-Time Detection System is divided into three main components: Real-Time Detection System administrator, application user and health center staff. A login interface page was included.

3.4. Communication Interface

The Real-Time detection system uses a mobile phone, SQL Server and managing system to control the users' connection with the servers.

3.5. Use Case Diagram

Use case diagram is an interaction between user and system, which have shown the connection between distinctive use cases involved by user and user. There are three types of end users: Real-Time Detection System Admin, Application users and Health Center Staff User. The section will explore individual use case diagrams for 3 forms of users.

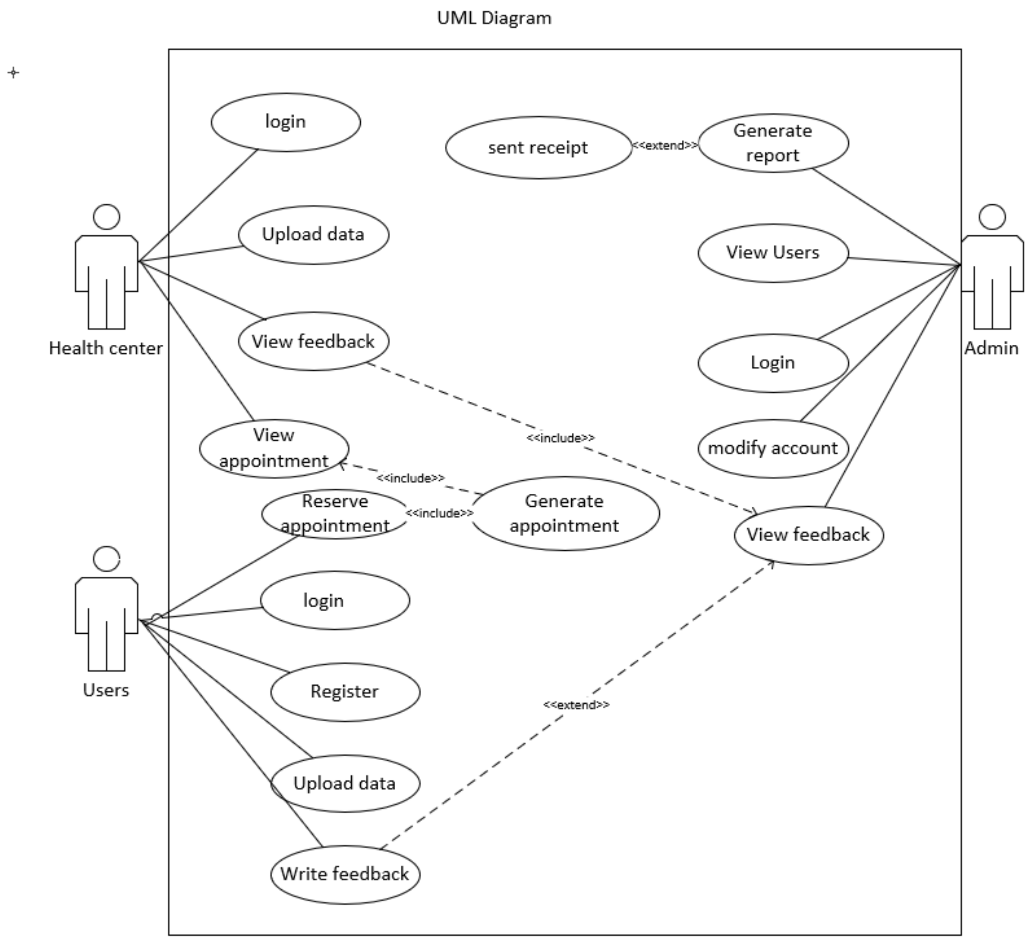

Figure 1. UML Diagram

3.5.1. Use Case Diagram for Real-Time Detection System Admin

A detailed use case diagram for Administrator is shown below.

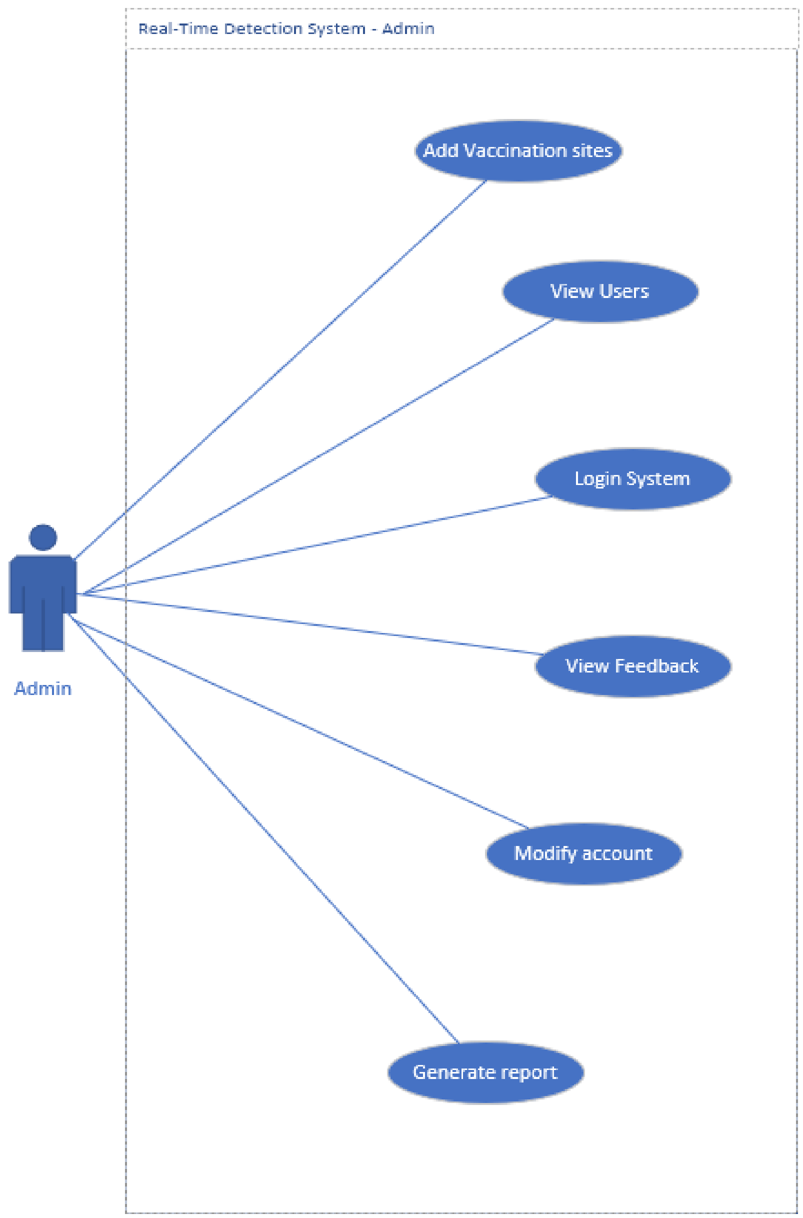

Figure 2. Real-Time Detection System - System Admin

3.5.2. Use Case Diagram for Application Users

A detailed use case diagram for Application Users is shown below.

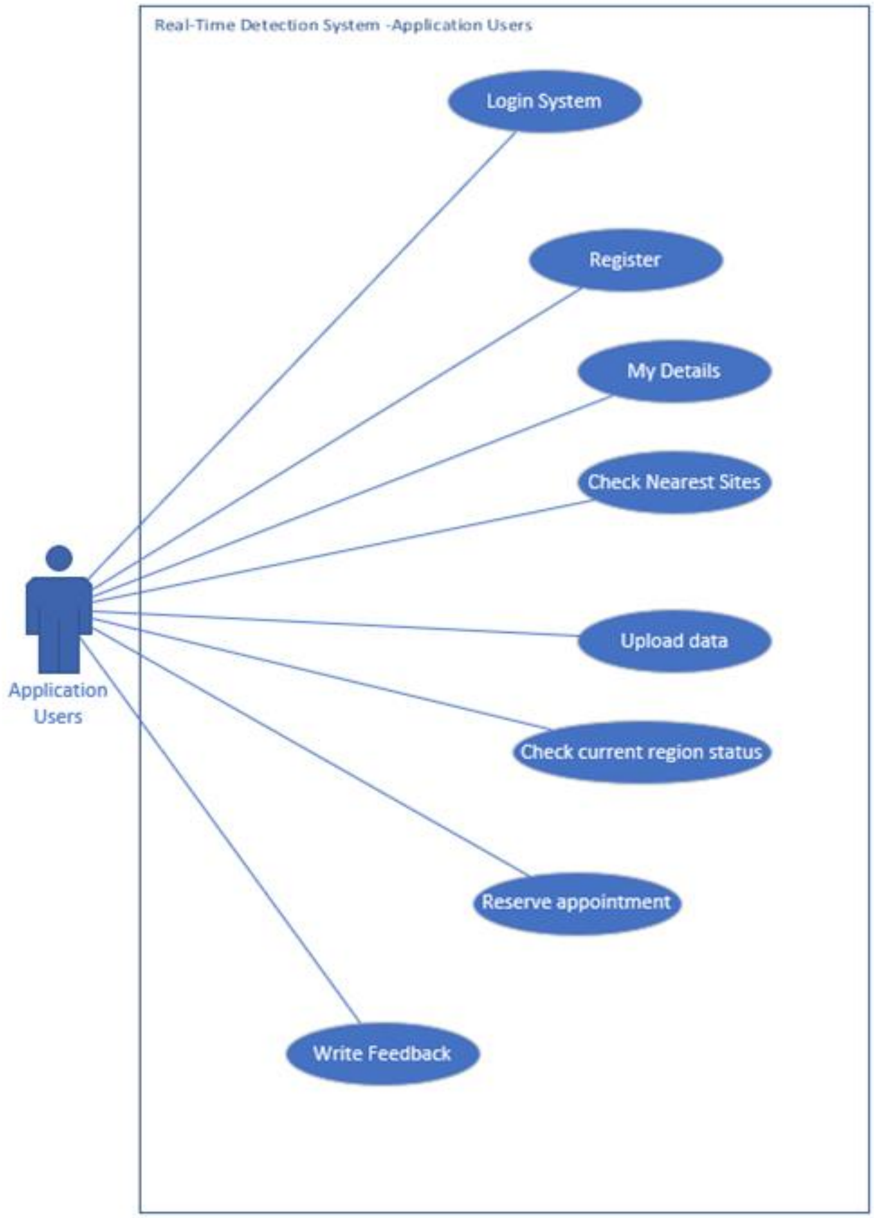

Figure 3. Real-Time Detection System -Application Users

3.5.3. <u>Use Case Diagram for Health Center Staff</u>

A detailed use case diagram for the Health Center staff is shown below.

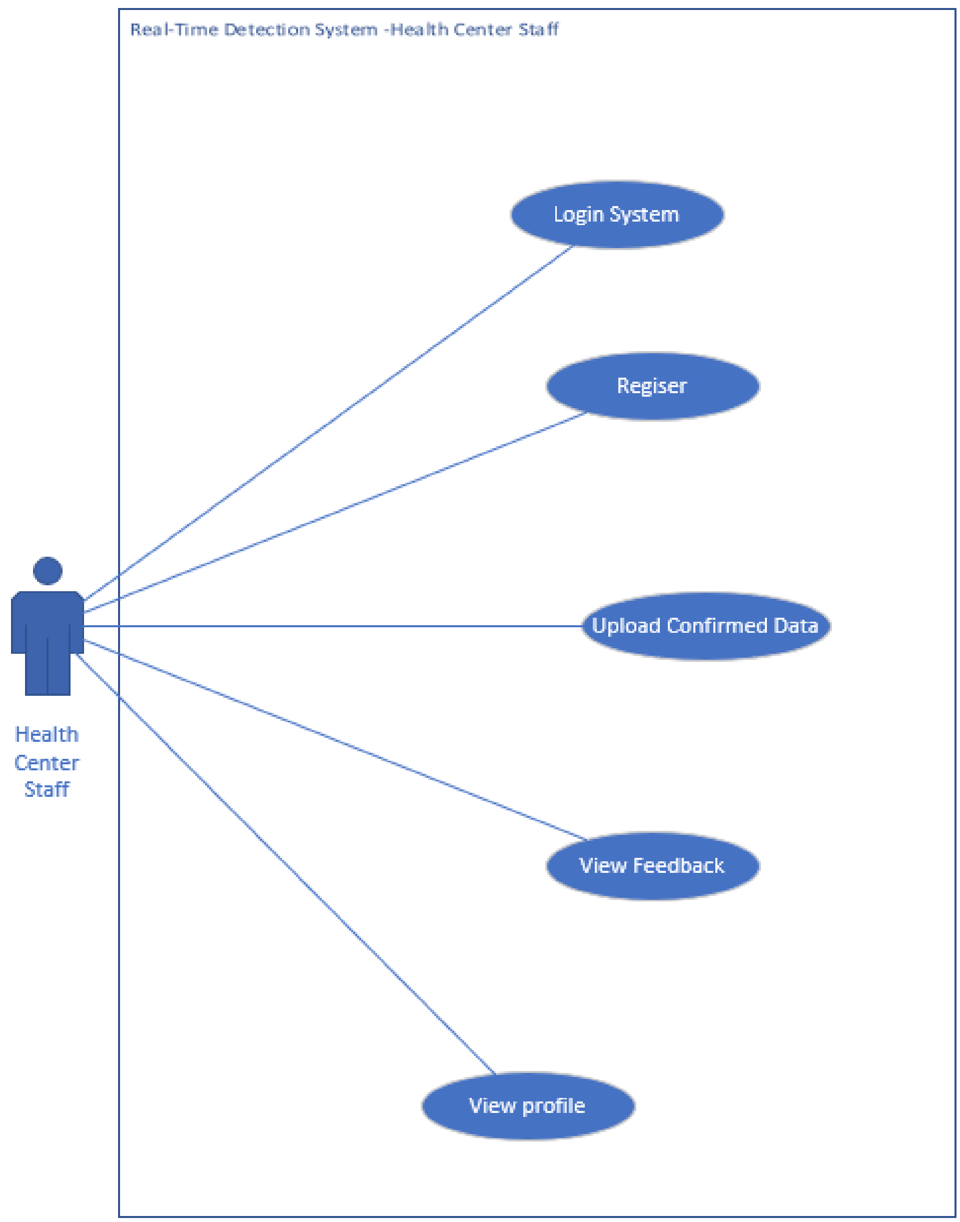

Figure 4. Real-Time Detection System -Health Center Staff

3.5.4. System Class Diagram

The structure of the system is described by displaying the classes, attributes, operations (or methods) and the relationships between objects.

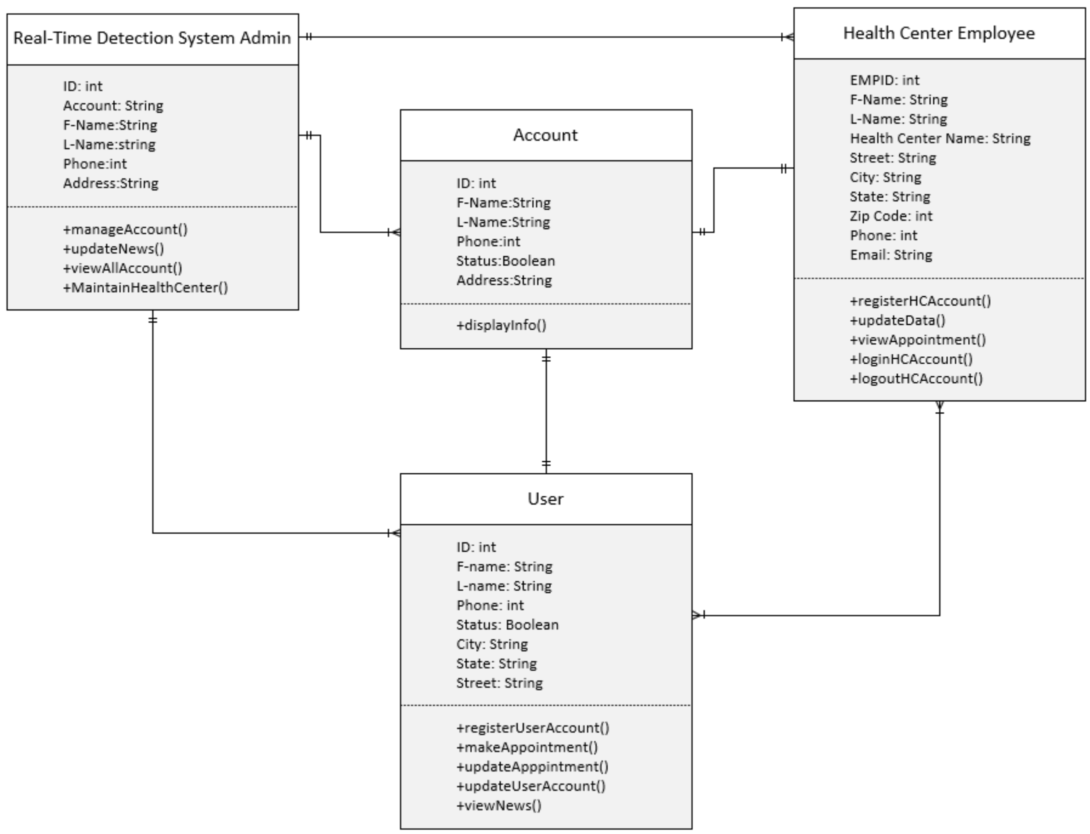

Figure 5. System Class

3.6. E-R Diagram

All the diagrams shown the relationship between entities and the relationship of every entity in the Real-Time Detection System.

1. User Account
2. Admin
3. Geography
4. Health Center
5. user
6. Vaccine

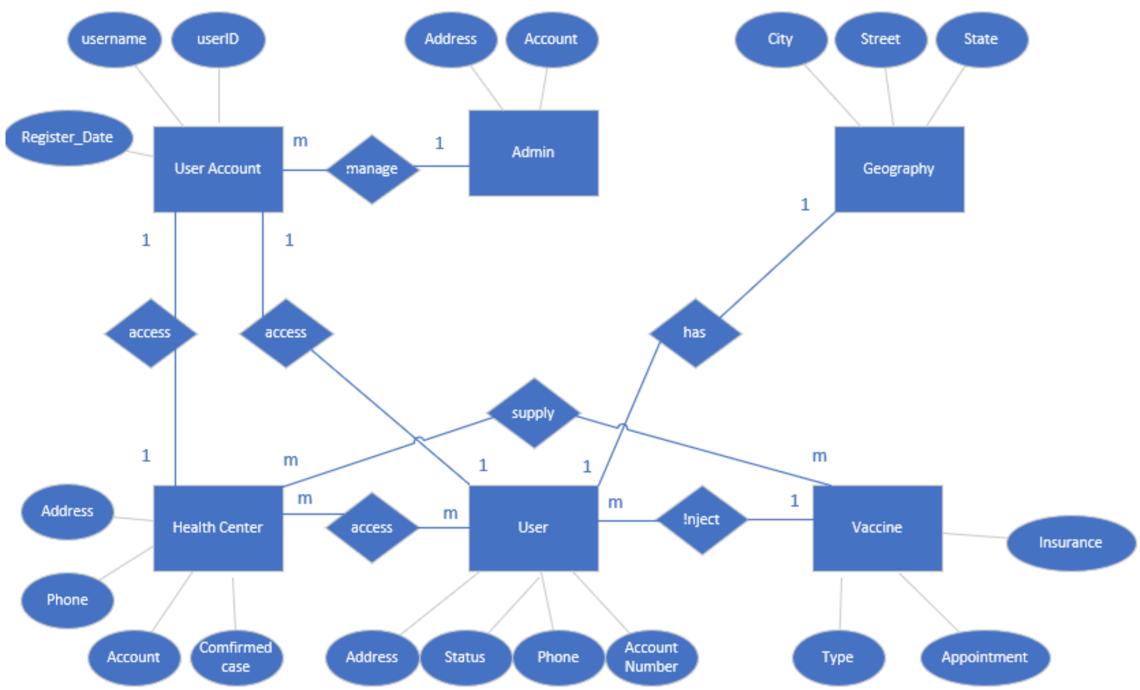

Figure 6. E-R Diagram

CHAPTER FOUR

IMPLEMENTATION

4.1 Chapter Overview

The Real-Time Detection System is a mobile application using Appmakr for development. Appmakr is a fashionable DIY mobile application development tool, which currently supports IOS, Android, along with other systems. Appmakr allows users to create applications in IOS along with Android with no programming. Additionally, appmakr can connect websites to applications, support HTML5, push notifications along with advertising support.

4.2 Account Interface Design

The following is the specific implementation of the system and some of the main user interfaces. It includes the home page, user login, appointment page, etc.

4.2.1 Home Page

These pages make the site for general and Health Center Staff users. This page is a brief introduction to this application. This includes mission, vision, and values.

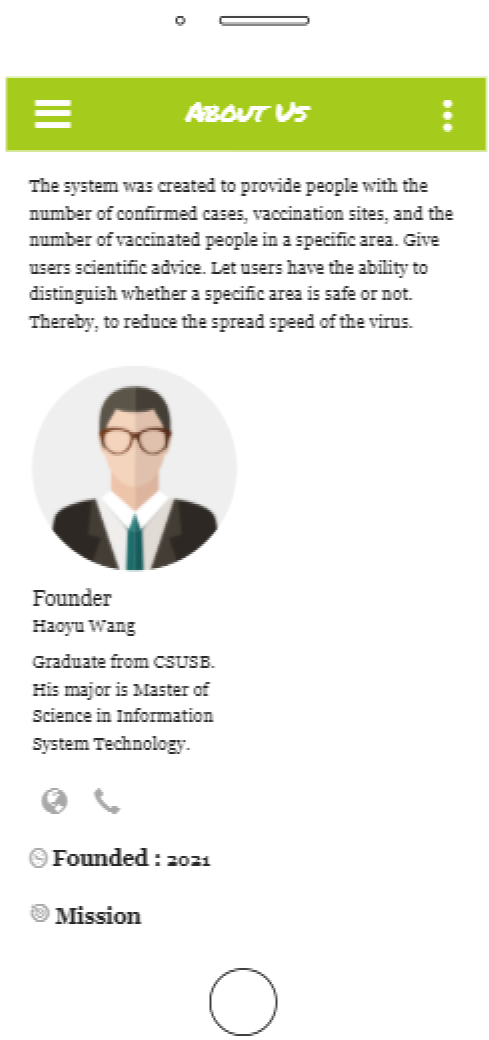

Figure 7. Home page

4.2.2 User Login

The sign in page there is for users to fill in their credentials in the system. We ask users to enter their phone / email address and password. Only after the user enters the username and password can the content be viewed.

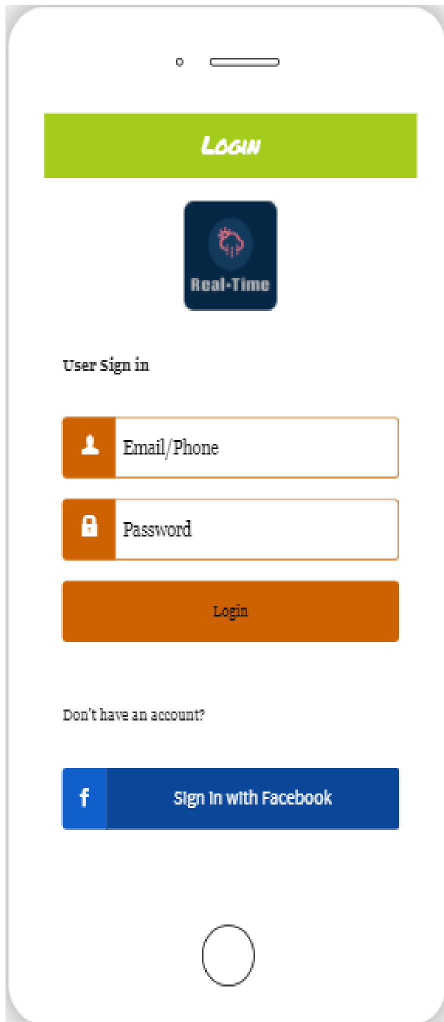

Figure 8. User Login

4.2.3 User Sign Up Page

The page is for users who hasn't created an account in the system. You need to create a new account. The system requires the user to enter the full name email, phone number, enter password twice and a nickname.

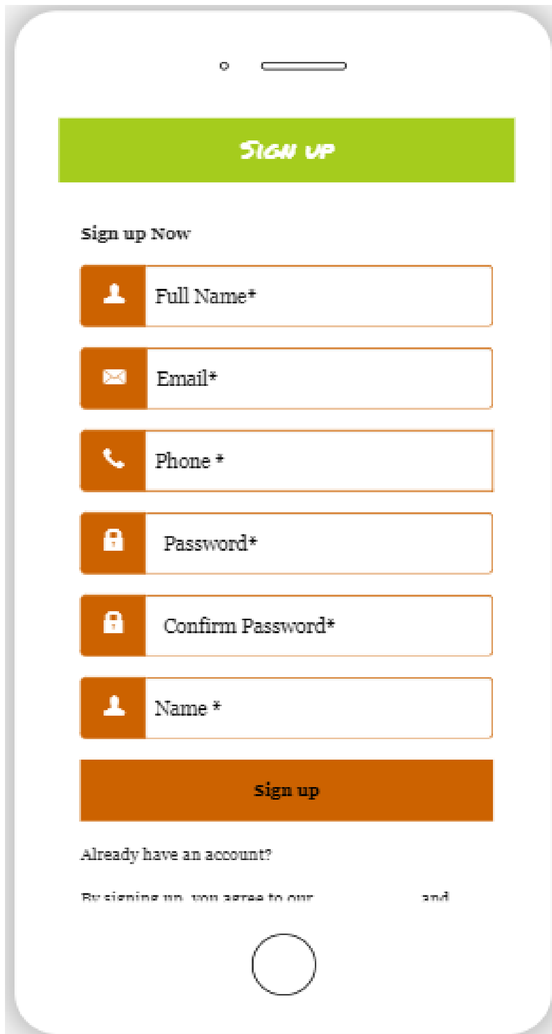

Figure 9. Sign Up

4.2.4 Health Center Page

This page can be used to query nearby clinics. All clinic information is entered manually and maintained in real time. All information comes from the clinic's official website or Google maps. In case of any change, it is maintained manually by the background. Click the site where you want to vaccinate, and an appointment interface will appear.

Figure 10. Health Center

4.2.5 Appointment Page

This interface provides users with the opportunity to make an appointment for the vaccine. The system requires the user to enter the name, email, mobile phone number, appointment time and remarks. When the appointment is successful, a confirmation letter will be sent to the mailbox automatically.

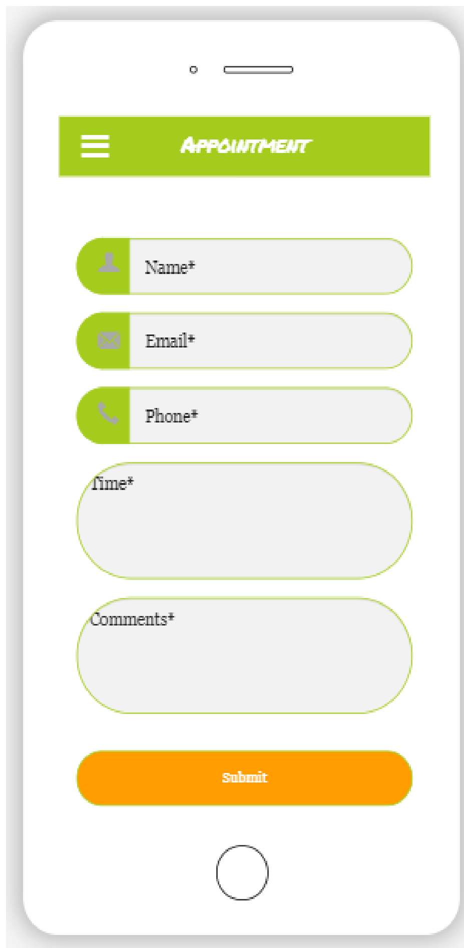

Figure 11. Appointment

4.2.6 Online Queueing Page

Once the user's reservation is successful. Users will automatically enter the queue of the day. The system automatically sorts according to the appointment time. Users can query how many people are ahead on this page to decide when to leave for the clinic. It can save time.

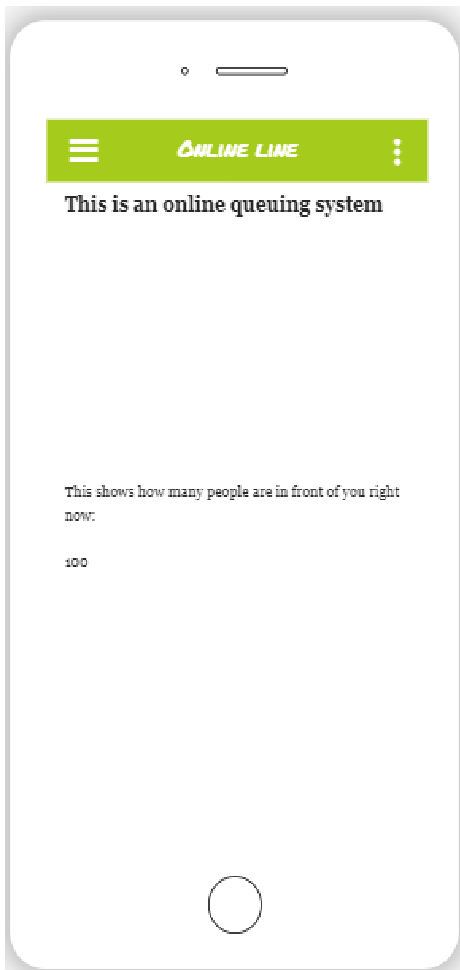

Figure 12. Online Queueing

4.2.7 My Profile Page

This page allows users to view their own user information. It includes name, phone number, email, whether it is a confirmed case and whether it has been vaccinated. Once the user is marked as a confirmed case. It will add one to the number of confirmed cases in the area. But the system does not record the names of confirmed cases.

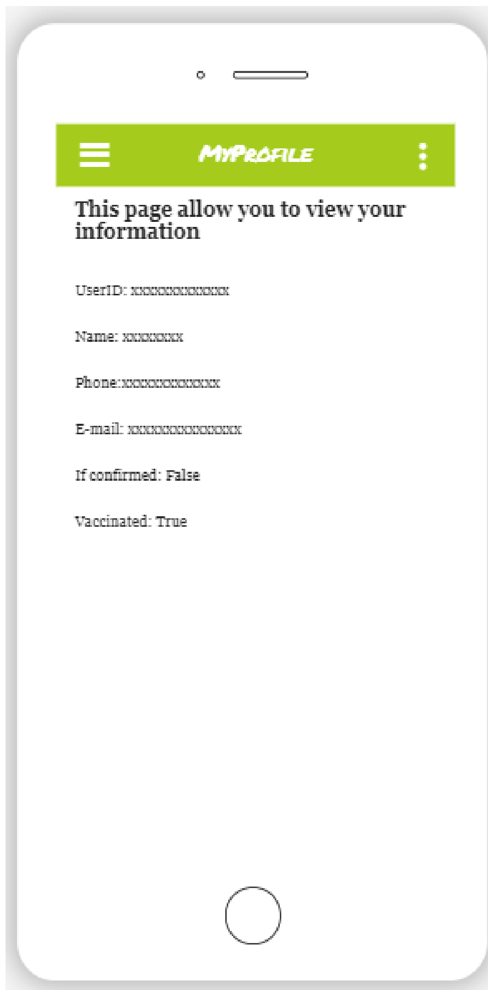

Figure 13. My Profile

4.2.8 Health Center Login

This page is dedicated for clinic login. Different from individual users. The clinic is the organization user, and the clinic user needs to provide the user ID and password for login.

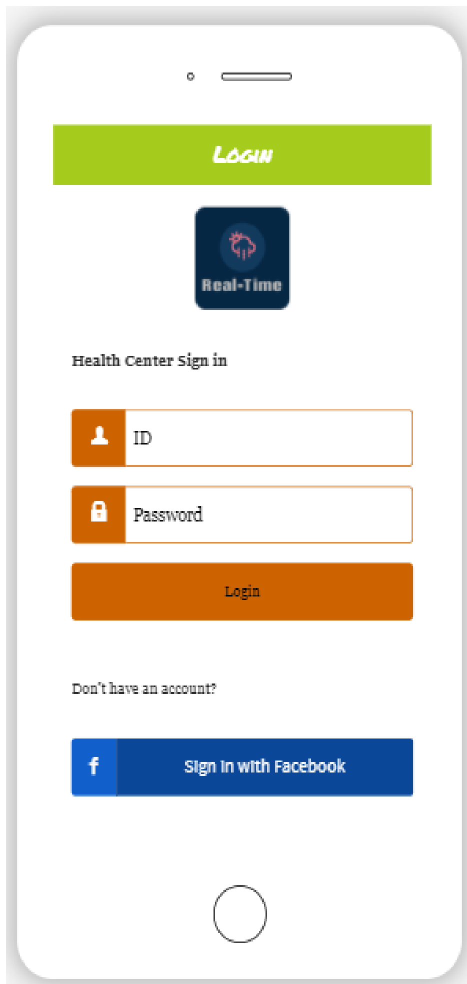

Figure 14. Health Center

4.2.9 Data Upload

This page allows organization users to upload the number of confirmed cases today. Generally, the numbers uploaded by users are marked as data with high reliability. In addition to the number of confirmed cases today, the contents of the column can be changed. Other information can be static for convenience.

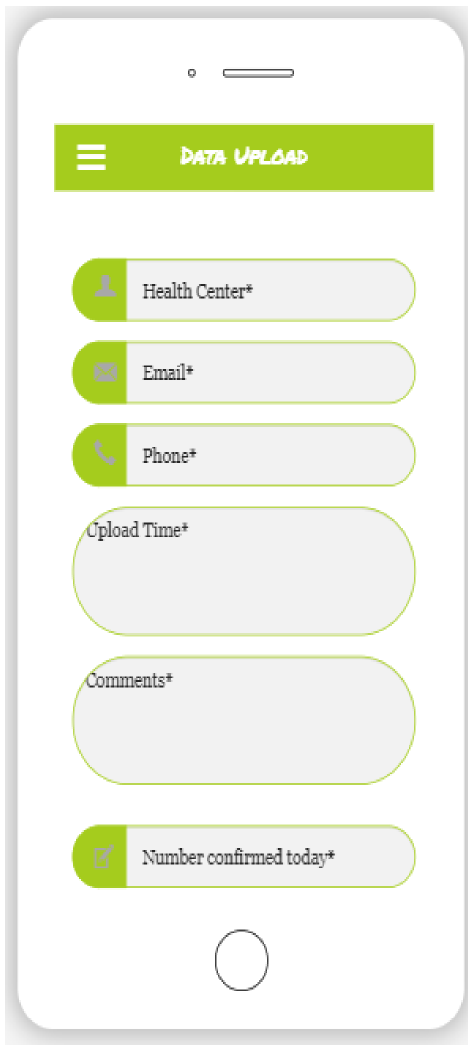

Figure 15. Data Upload

4.2.10 Real-Time Detection System Administrator Background Management

The management uses SQL server to connect with the background. It is used to manage and maintain all data of users. This includes the user's name, email, phone number, password and nickname. The system will also automatically connect with the database, once users forget their registration information, they can get information from the database.

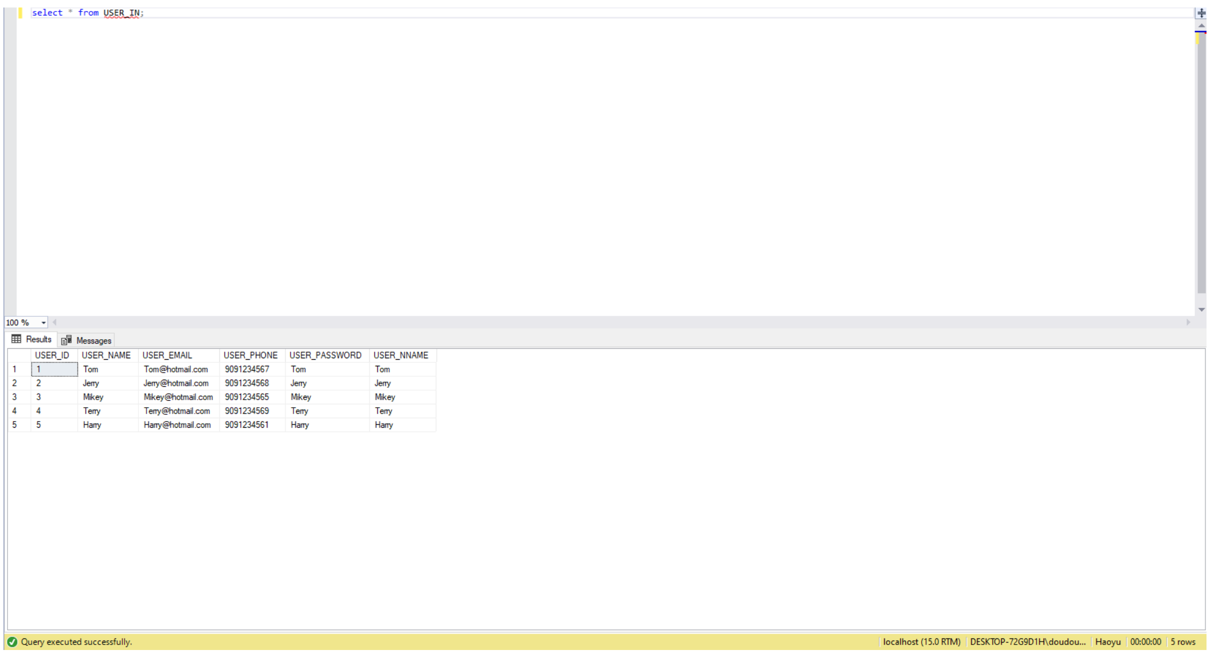

Figure 16. Background Management

4.2.11 Interfaces and Components Architecture Design

Because the system mainly runs on the mobile phone, the navigation buttons in the upper left corner are used to interact with each other. As shown in the figure below, you can select the function you need through the navigation button.

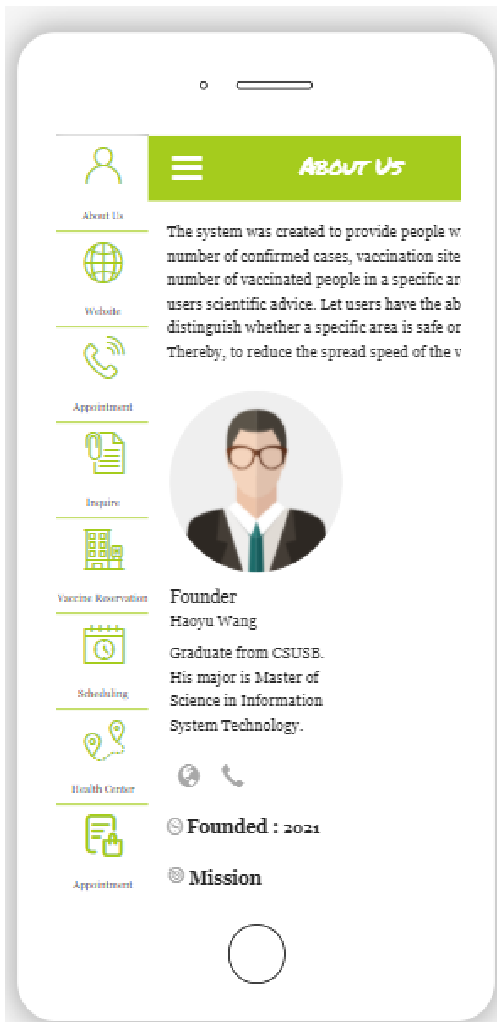

Figure 17. Interfaces and Components

4.2.12 Deployment Implementation Diagram

The deployment implementation mainly includes three modules: application server, client and database server. Their interaction is shown in the figure below:

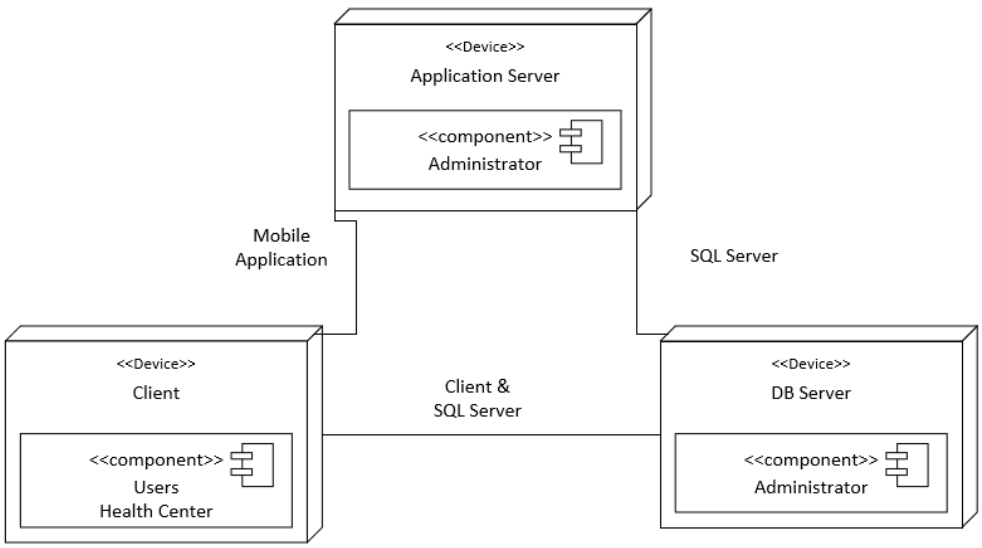

Figure 18. Deployment

4.3 Technical Feasibility

In this chapter, the system will certify whether the ideal system is technically acceptable. That is to say, the technologies needed to improve the system are available at any time.

Whether an organization has the technologies needed to implement a project, and how to acquire them could be tested in technical feasibility determines. The system is feasible for the following reasons:

The software system used in the system exists. And the operation is simple (Appmakr, SQL server).

The system is reusable. This system can be run whenever an unforeseen epidemic comes. Just add epidemic characteristics to the system.

The system has strong expansibility. In the future, more functions about diseases can be added. For example, disease knowledge, detection of whether there are virus carriers around.

4.4 Economic Feasibility

This project is economically feasible because only a small amount of investment is involved in generating the product. For example, it costs only $20 a month for an application to generate a web page. Most of the resources used such as the Microsoft SQL Server are open source products. It took me about three months to build the system. But it may take more time to include more features. The company running this system can get huge economic benefits from this system. Because everyone pays attention to COVID-19 pandemic news. Users of the system only need to pay $10 a month membership fee to get the service of the system.

4.5 Operational Feasibility

This step focuses on low input and high output. It's in the form of labor costs, a small amount of money and time. In the case of a real time detection system, there are fewer staff, and the main labor cost comes from the later system maintenance. Although this project takes time to build, it is the first of its kind and everyone needs it. It has easy to use and portable features that provide high operational feasibility.

CHAPTER FIVE

PROTOTYPE SYSTEM TESTING AND RESULTS

5.1 Summary

This project invited people who understand (White box testing) a project, and people who don't understand (Black box testing) a project, to test the prototype. This included: testing the string input of users and administrators in the application, such as name, email, details, password, etc. and complete the independent test and Integration test. The test results met the requirements of the deployment.

Users can find out if there are too many confirmed cases in the area by using the prototype. All users in a sample survey reported will not visit an area with many confirmed cases. Since the prototype system allowed users to make an appointment for vaccination, the objective of having a system that helped reduce virus transmission was achieved.

5.2 Blackbox Testing

Testing the system not knowing the internal workings of the system is included in black box testing (Imperva, n.d.). The worker will provide input, then observe the results produced. So, it possible to determine how the system responds This can monitor the expected operation time, availability, and reliability of the system.

5.3 Whitebox Testing

Also known as glass test. The internal structure along with the data structure used, the internal design, the code structure, the work of the software were what the white box testing technology analyzes (GeeksforGeeks,2019). The box is visible according to white box, that is, knowing what's inside and how it works in the box.

5.4 Graybox Testing

A kind of test between what we've mentioned above is gray box testing. It's mainly applied in the integration testing phase, which pays more attention to the input, output along with the internal correctness of the program. It's not as detailed and complete as white box testing, however, it focused more on the internal logic than black box testing. It always judges the internal running state of the enterprise through some representative phenomena along with events. It combines the two testing approaches to overcome their deficits (Patricia,2021).

5.5 Satisfaction Survey

As shown below, we can see the system almost meets clients' expectations.

Distributioı	UserLangu	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Distribut	User Lar	Real-Tir	Real-Tir	Real-Tir	I will cor	Real-Tir	Real-Tir	Real-Tir	Real-Tir	Real-Tir	Real-Tir
anonymous	EN	Strongly agree	agree	Extremely useful	Yes	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Somewhat agree	Strongly agree
anonymous	EN	agree	agree	Slightly useful	Yes	Somewhat agree	Somewhat agree	Somewhat agree	Strongly agree	Somewhat agree	Somewhat agree
anonymous	EN	agree	Strongly agree	Very useful	Yes	Strongly agree	Strongly agree	Strongly agree	Somewhat agree	Strongly agree	Strongly agree
anonymous	EN	agree	agree	Very useful	Yes	Somewhat agree	Somewhat agree	Somewhat agree	Somewhat agree	Somewhat agree	Somewhat agree
anonymous	EN	agree	agree	Very useful	Yes	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree
anonymous	EN	agree	agree	Extremely useful	Yes	Somewhat agree	Strongly agree	Strongly agree	Somewhat agree	Strongly agree	Strongly agree
anonymous	EN	Strongly agree	Strongly agree	Extremely useful	Yes	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree
anonymous	EN	Strongly agree	Strongly agree	Extremely useful	Yes	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree
anonymous	EN	Strongly agree	Strongly agree	Extremely useful	Yes	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree

Figure 19. Satisfaction Surevey1

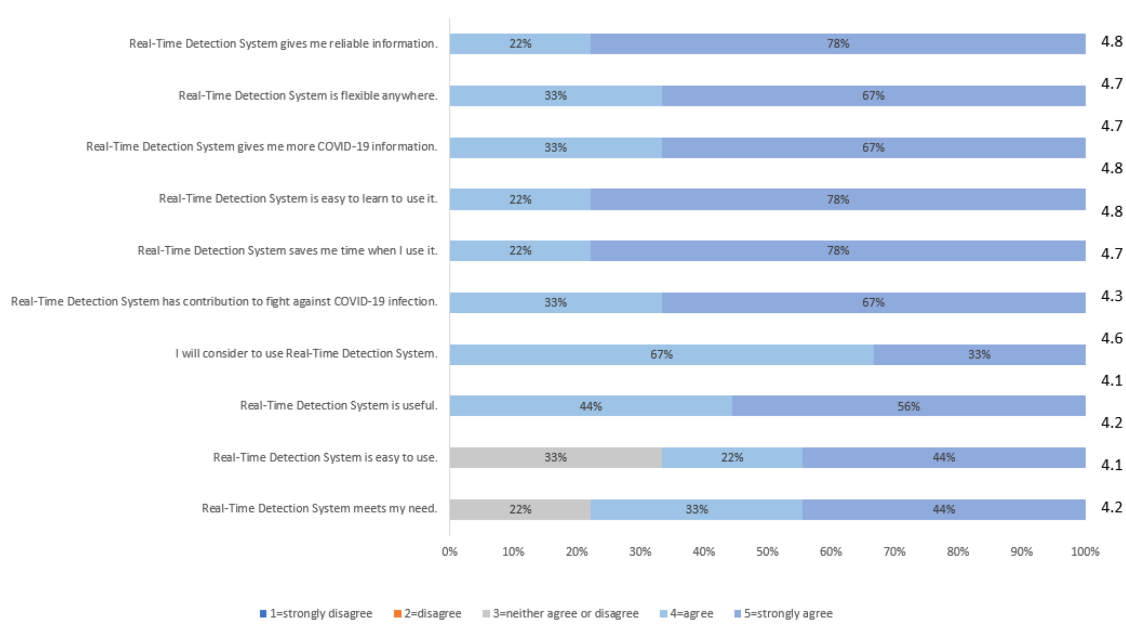

Figure 20. Satisfaction Surevey2

CHAPTER SIX

CONCLUSION

6.1 Summary

This project is our joint effort of every course under the MSIST program in the past two years. The reason why we chose this project is that thousands of lives are at risk because of the covid-19 pandemic. More than 10000 confirmed cases spread rapidly every day. Therefore, it is obviously necessary to design a system to solve this problem. Real-Time detection system is the first such system. It’s a blueprint with strong features, which could be improved to play its functions at any time.

To complete this project, I have done a lot of tests and collected the majority of data. Vital functions are essential to achieve the deployment requirements. Through these functions, we can effectively slow down the spread of the virus.

6.2 Conclusion

The development of the Real-time detection system takes into account users from different areas and different aspects. Its main focus is to make users benefit from it and indirectly slow down the spread of the virus. The system is a mobile app that can be accessed from anywhere, anytime.

This friendly interface consists of three modules: administrator, general user and organization user. Each user uses its own user interface independently. It's easy to develop. You only need to use Appmakr. User don't need these tools, because it's just a front-end prototype, and the application will be sufficient to interact with the user in the system and obtain information.

The facial mask recognition function can be added in the future development. Once a variety of viruses start to spread, you can start the application in an emergency. The app can be connected to the background, whenever the user needs to unlock the phone for face recognition. Once it is found that the user is not wearing a mask, the user can be reminded in time. If we can further develop the function of forcing users to wear masks. For example, if you don't wear a mask, you can't unlock your mobile phone, or you keep making noises to remind users. After the virus epidemic, once entering normal life. The system can be used as a common application to query health status. The reuse rate of the system is greatly improved.

APPENDIX A

MISSION，VISION, VALUES, SWOT

MISSION: The mission of Real-Time Detection System is to collect data uploaded by users. It includes individual users and organizational users. We will analyze these data and use the analysis results to give users reasonable suggestions. Help users effectively avoid high-risk areas. And provide users with available vaccination sites.

VISION: Because of the Covid-19 pandemic. With more than 500000 deaths in the United States, the most effective way to protect ourselves is to reduce contact with confirmed cases and stay away from crowded areas. Through the reasonable suggestions given by the Real-Time Detection system, users can contact virus carriers less. Return to normal everyday life as soon as possible. The next time a new virus appears, the system can still be used.

VALUES: Initiative -Constantly improve and update the system and strive to let every user get help from our system.

Communication -Keep in touch with each user and read each user's feedback carefully.

Service -Try to provide 7x 24-hour online service. Help every user who needs help.

Privacy- Protect the privacy of every user according to the law and ethics.

Quality & Safety - Maintain and update the system at any time. Guarantee to provide high quality service and user interface.

SWOT:

Strengths	Weaknesses
1.Powerful data management methods can quickly provide different solutions for different problems of different users. 2.Low cost of data acquisition. 3.Low system maintenance cost.	1.The reliability of data uploaded by individual users needs to be further confirmed. 2.At the end of the epidemic, some functions of the software cannot be reused.
Opportunities	**Threats**
1.Advanced technologies and methods can quickly respond to epidemic prevention and control strategies. 2.Once the epidemic broke out, the system could be put into use quickly. 3.Provide affordable open source and data mining tools, and recruit business intelligence partners / Coordinators and employees to jointly maintain the source code. 4.The same system has less competition pressure.	1.Some institutions and individuals may be reluctant to upload data for fear of privacy issues.

APPENDIX B

REQUIREMENTS

ID	Details	Type	Priority
R1	The Real-Time Detection System is supposed to register new account for individual user	Functional User Account	Must Have
R2	The Real-Time Detection System is supposed to permit users to log on.	Functional User Account	Must Have
R3	The Real-Time Detection System is supposed to permit users to log off.	Functional User Account	Must Have
R4	The Real-Time Detection System is supposed to register new accounts for organizational users.	Functional User Account	Want to Have
R5	The Real-Time System is supposed to require the account password to fulfill these complexity requirements: at least 10 characters, including at least one uppercase letter, lowercase letter, number, along with special character.	Functional User Account	Must Have

R6	The Real-Time Detection System is supposed to permit users to reset their password.	Functional User Account	Must Have
R7	The Real-Time Detection System is supposed to permit editing.	Functional User Account	Must Have
R8	The Real-Time Detection System is supposed to permit the deletion of an account.	Functional User Account	Should-Have
R9	The Real-Time Detection System is supposed to permit users to view to their accounts	Functional User Account	Must Have

R10	The Real-Time Detection System is supposed to permit users to update profiles.	Functional Performance	Must Have
R11	The Real-Time Detection System is supposed to authenticate users who use 256-bit encryption.	Non-Functional Security	Should Have

R12	The Real-Time Detection System is supposed to be enterable via HTTPS only.	Non-Functional Security	Must Have
R13	The Real-Time Detection System is supposed to offer a security audit trail.	Non-Functional Security	Should-Have
R14	The Real-Time Detection System is supposed to stay operational all day long, 365 days every year.	Non-Functional Availability	Should-Have
R15	The Real-Time Detection System is supposed to be acceptable in any devices.	Non-Functional Availability	Should-Have
R16	The Real-Time Detection System is supposed to charge monthly	Functional Payment	Must have

R17	The Real-Time Detection System shall send out receipts within 5 days of processing the payment.	Functional Payment	Must Have

R18	The Real-Time Detection System shall accept Visa and Mastercard.	Non-Functional Payment	Should Have
R19	The Real-Time Detection System shall allow users to reserve vaccine sites.	Functional Performance	Must Have
R20	The Real-Time Detection System shall use cloud computing.	Non-Functional Infrastructure	Should Have
R21	The Real-Time Detection System shall request data from Health Centers and store them in our own databases.	Functional Database Entry API	Must Have
R22	The Real-Time Detection System shall allow the input of original data information. -Country -Region -Virus Type	Functional performance	Must Have

R23	The Real-Time Detection System shall allow updates to original data information.	Functional Database	Must Have
R24	The Real-Time Detection System shall allow the input of potential new clients.	Functional Database	Should Have
R25	The Real-Time Detection System shall allow the updates via Government Websites.	Functional Database	Must Have
R26	The Real-Time Detection System shall provide a method to search through raw data.	Functional Database	Should Have
R27	The Real-Time Detection System shall provide search results within 3 ticks.	Non-Functional Performance	Must Have

R28	The Real-Time Detection System is supposed to provide a method to filter data by any attribute. · Country · Region · Virus Type	Functional Performance	Must Have
R29	The Real-Time Detection System shall provide a simple interface.	Non-Functional Performance	Must Have
R30	The Real-Time Detection System is supposed to prepare a responsive interface.	Non-Functional Performance	Must Have
R31	The Real-Time Detection System is supposed to show analytics on the original data.	Functional Performance	Must Have
R32	The Real-Time Detection System provides the total number of confirmed cases in a geographical area.	Functional Analytics	Must Have

R33	The Real-Time Detection System provides the total number of death cases in a geographical area.	Functional Analytics	Must Have
R34	The Real-Time Detection System provides the total number of vaccinations in a geographical area.	Functional Analytics	Must Have
R35	The Real-Time Detection System is supposed to prepare the mean and median production value within a geographical area.	Non-Functional Analytics	Must Have
R36	The Real-Time Detection System is supposed to prepare a reasonable suggestion in a geographical area.	Non-Functional Analytics	Must Have
R37	The Real-Time Detection System is supposed to produce monthly reports which was predefined.	Non-Functional Analytics	Should Have
R38	The Real-Time Detection System is supposed to permit us to generate personalized reports.	Non-Functional Analytics	Should Have

R39	The Real-Time Detection System shall protect the privacy of individual users. Including name, address and health status. All information provided should be anonymous.	Functional Database	Must Have
R40	The system automatically classifies the information uploaded by individual users. Providing relevant records will be considered as real data.	Functional Database	Must Have

APPENDIX C

USE CASE SPECIFICATIONS

Use Case: CreateAccount
Use Case ID: 1
Description: Create an account for everyone who wants to use the system. Including users and Health Center Staff.
Preconditions: 1.The User turns on the system and is ready to register. 2.The system runs normally. 3.The user doesn’t have an account.
Main Flow: 1. Whenever the Users picked “Don’t have an Account?” the case started. 2. Their username along with password were asked by the system. 3.The User enters their username along with password. 4. The system stores the user’s passwords along with user's name. 5.Users have access to the system.
Secondary scenarios: UserLogin FindVaccineSite ManageAppointime
Postconditions: 1.The user has a unique account. 2.The user can process his/her own account.
Primary Actors: User, Health Center
Secondary Actors: None

Use Case: UserLogin
Use Case ID: 2
Description: Users log in the system to find information.
Preconditions: 1.An account which was possessed by the user. 2.The system runs normally. 3.The account status is normal.
Main Flow: 1. User cases started when the User picks “Log in”. 2. Username and password were asked by the system. 3.The Users enter their username along with password. 4. The system checks the user’s passwords along with user's name. 5.The User gains access to the system.
Secondary scenarios: FindVaccineSite ManageAppointime UploadUserInfo
Postconditions: 1.The user can process his/her own account. 2.The user can process the whole function of the system.
Primary Actors: User, health center staff
Secondary Actors: None

Use Case: FindVaccineSite

Use Case ID: 3

Description:

Find the nearest vaccine site by entering the zip code or using GPS.

Preconditions:
1.The user has an account.
2.The system runs normally.
3.The account status is normal.

Main Flow:

1. System started whenever the Users select “Find Vaccine Site”.
2. Users can find the nearest health center by GPS.
3. System shows the list of health centers.
4. Users can pick any one from the nearest health center.
5. The system accesses the database and records the users information.

Secondary scenarios:
AppointmentForVaccination

Postconditions:
1.The user can manage his/her vaccine status.
2.Users can make an appointment for vaccination.

Primary Actors:

User

Secondary Actors:

None

Use Case: AppointmentForVaccination
Use Case ID: 4
Description: Make an appointment at the nearest vaccination site. Time and name required.
Preconditions: 1.The user has an account. 2.The system runs normally. 3.The account status is normal. 4.The user finds the nearest health center.
Main Flow: 1. System started when the Users select “Make an appointment”. 2. The system requests the user to import E-mail , name, phone number, appointment time and comments. 3. The system will record the changes. 4.The system will ask users to confirm the appointment.
Secondary scenarios: ManageAppointime GenerateReport
Postconditions: 1.The user can manage his/her vaccine status. 2.Users can cancel an appointment for vaccination. 3.Users can change the date of the appointment.
Primary Actors: User
Secondary Actors: None

Use Case: ManageAppointime
Use Case ID: 5
Brief Description: Cancel or change the vaccination time.
Preconditions: 1.An account was needed for the user. 2.The system runs normally. 3.The account status is normal. 4.The user has an appointment in his/her account. The appointment is available.
Main Flow: 1. System starts whenever the Users select “Update an appointment”. 2. The system requests the User to import email , name, phone number, appointment time and comments. 3. The system will ask users to confirm the changes. 4. The system will record changes in the database.
Secondary scenarios: None
Postconditions: 1.The user can manage his/her vaccine status. 2.Users can cancel an appointment for vaccination. 3.Users can change the date of the appointment.
Primary Actors: User
Secondary Actors: Health Center Staff

Use Case: GenerateReport
Use Case ID: 6
Preconditions: 1.The user has an account. 2.The system runs normally. 3.The account status is normal. 4.The user clicks the Report icon to generate a report.
Brief Description: The system gives reasonable suggestions based on big data analysis. Red is high risk. Green is normal.
Main Flow: 1. When the Users select "Report" then the system starts. 2. The system asks the database whether the demand has specific data. 3. If there is data in the database 3.1 The system asks the users to enter a zip code or specific city. Then the system will interact with the database to return the number of confirmed cases. 4. If there is no data in the database 4.1 Return a NULL value. 5. If there are many confirmed cases in the area, return red. ELSE If the number of confirmed cases in the area is small, return green.
Secondary scenarios: None
Postconditions: 1.The user can decide whether to go to the region or not.
Primary Actors: User
Secondary Actors: Health Center

Use Case: UploadUserInfo
Use Case ID: 7
Brief Description: Users could choose to upload their own data. Need to distinguish between organization users and individual users.
Preconditions: 1.The user has an account. 2.The system runs normally. 3.The account status is normal. 4.The user clicks the MyProfile button and modifies his/her information.
Main Flow: 1. System begins whenever the Users select "My profile". 2. System shows the ID along with the user's Name. These two attributes can't be modified. 3. The users can modify their phone number, e-mail. 4. Users can upload their own vaccination or not. 5. The system asks the user to confirm any changes.
Secondary scenarios: None
Postconditions: 1.The user can modify his/her profile at any time and anywhere. 2.The user can check his/her new profile.
Primary Actors: User
Secondary Actors: Health Center Staff

Use Case: UpdateConfirmationCases
Use Case ID: 8
Preconditions: 1.The user has an account. 2.The system runs normally. 3.The account status is normal. 4.The Health Center Staff clicks the DATA UPLOAD button.
Brief Description: Update the confirmed cases everyday if possible.
Main Flow: 1. System begins whenever the Health Center staff selected “Data upload”. 2. The Health Center staff enter the Health Center name. 3. The system will automatically interact with the database and return the current confirmed cases quantity to the user. 4. The Health Center staff input the confirmed cases of the day. 5. The system will interact with the database. Add confirmed cases to the original total. 6. The system asks the Health Center staff to confirm the changes. 7. The Health Center staff confirm the changes.
Secondary scenarios: None
Postconditions: 1.The user can modify his/her status at anytime and anywhere. 2.The user can check his/her new status.
Primary Actors: Health Center Staff
Secondary Actors: None

Use Case: OnlineQueuing
Use Case ID: 9
Preconditions: 1.The user has an account. 2.The system runs normally. 3.The account status is normal. 4.The user clicks the online queuing button.
Brief Description: Can line up online. No need to line up offline
Main Flow: 1. The system begins whenever the User pointed “OnlineQueuing”. 2. The user chooses to queue online and input zip code or city. 3. The system will interact with the database and return nearest Health Centers or Vaccination sites to the user. 4. The system returns how many people are waiting to the user. 5. The user chooses any one from the list of Health Centers or Vaccination sites. 6. The system will automatically add one to the waiting number in the database. 7. The system returns how many people are waiting before the user.
Secondary scenarios: None
Postconditions: 1.The user can check how many people are before him/her at any time and anywhere.
Primary Actors: User

Secondary Actors: Health Center

Use Case: UserLogout
Use Case ID: 10
Brief Description: Users quit system
Preconditions: 1.The users have accounts. 2.The system runs normally. 3.The account status is normal. 4.The user clicks the log out button.
Main Flow: 1. The system begins whenever the User pointed "Log out". 2. All user data in the database were saved. 3. Users were log out by system.
Secondary scenarios: None
Postconditions: 1. The users log out the system. 2. The system will automatically record the status when the user logs out.
Primary Actors: User
Secondary Actors: Health Center Staff

REFERENCES

GeeksforGeeks, (2019, Apr 30). Software Engineering | White box Testing https://www.geeksforgeeks.org/software-engineering-white-box-testing/

Imperva,(n.d.). What is Black Box Testing. https://www.imperva.com/learn/application-security/black-box-testing/

Indian J Pediatr,(2020, March 13). A Review of Coronavirus Disease-2019 (COVID-19). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7090728/

My COVID-19 TRACKER,(2021,April 10). My COVID-19 TRACKER. https://play.google.com/store/apps/details?id=edu.slu.mycovid19tracker&hl=en_US&gl=US

Patricia Johnson,(2021, February 4)Gray Box Testing Guide. https://www.whitesourcesoftware.com/resources/blog/gray-box-testing

/

Public Health Ontario, (2021, June 30). COVID-19 Routes of Transmission – What We Know So Far. https://www.publichealthontario.ca/-/media/documents/ncov/covid-wwksf/2020/12/routes-transmission-covid-19.pdf?la=en

The New York Times, (2021, April 12). Covid-19: U.S. Surpasses 500,000 Covid-19 Deaths, a Monumental Loss. https://www.nytimes.com/live/2021/02/22/world/covid-19-coronavirus

World Health Organization, (n.d.). Coronavirus disease(COVID-19) pandemic. https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/novel-coronavirus-2019-ncov