Software Design Description

Applicability of mobile contact tracing in fighting pandemic (COVID-19)

Version: 1.1

Prepared by

Group: oneByte

SCSJ4423 – Real-time Software Engineering

School of Computing

Faculty of Engineering

Universiti Teknologi Malaysia

Submitted To

Dr.Dayang Norhayati Abang Jawawi

Date of Submission:

10th May,2021

Revision Page

a. Overview

This System Design Description of our proposed system includes the introduction consisting purpose, scope and etc, system software process model description, system software requirements like functional, non-functional, constraints and etc, system software architecture design such as system architecture, component diagram and so on, detailed description of components, data design, user interface and requirement matrix.

In the part of introduction, the purpose to create this system design description, scope of proposed system, definitions, acronyms and abbreviations will be described. The references which were used to be referred will also be included as well as the overview of this system design description. Additionally, the real-time profile and modeling tools will be included followed by an innovation solution for IOT and data analytics.

It is followed by the process model we used to develop the proposed system. In the system software requirements, the functional requirements and non-functional requirements will be described together with constraints and interface. The part of system software architectural design will show the architecture design diagrams including system architecture, component diagram, class diagram and concurrency design.

The detailed description of components will show the package diagram with detailed descriptions. And the data design will show the Entity Relationship Diagram and data dictionary. This SDD will be ended with a user interface design and requirements matrix.

b. Target Audience

Our target audience for this document are Software Architect, Software Engineer, Developer, and System Administrator.

c. Project Team Members

1) Muhammad Mohsin Shaikh

2) Muhammad Naif Bin Norian

3) Nurul Ismat Tanni

4) Syed Muhammad Yamin Gharbi

5) Lee Kar Hung

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	1) Muhammad Mohsin Shaikh 2) Muhammad Naif Bin Norian 3) Nurul Ismat Tanni 4) Syed Muhammad Yamin Gharbi 5) Lee Kar Hung	● Introduction ● System Software Process Model ● System Software Requirements ● System Software Architecture Design	18 April, 2021
1.1	1) Muhammad Mohsin Shaikh 2) Muhammad Naif Bin Norian 3) Nurul Ismat Tanni 4) Syed Muhammad Yamin Gharbi 5) Lee Kar Hung	● Introduction (refinement) ● System Software Process Model(refinement) ● System Software Requirements(refinement) ● System Software Architecture Design(refinement) ● Detail description of components ● Data Design ● User Interface Design ● Requirement Matrix	10 May 2021

Note:

This document is adapted from ***IEEE SA - 830-1998 -*** *IEEE Recommended Practice for Software Requirements Specifications* and ***IEEE SA - 1016-1998 -*** *IEEE Recommended Practice for Software Design Descriptions*.

Table of Contents

Revision Page 2

Table of Contents 4

1 Introduction 6

1.1 Purpose 6

1.2 Scope 6

1.3 Definitions, Acronyms and Abbreviations 7

1.4 References 7

1.5 Overview 8

1.6 Real-time profile and modeling tools 8

1.7 Innovation solution for IOT and Data analytics 9

2 System Software Process Model 10

2.1 Introduction 10

2.2 Scrum 11

3. System Software Requirements 13

3.1 Functional 13

3.2 Non-Functional 14

3.2.1 Performance 14

3.2.2 Security 14

3.3 Constraints 14

3.4 Interface 14

4 System Software Architectural Design 15

4.2 Software Architecture – Component Diagram 17

4.3 Class Diagram 17

4.4 Concurrency Design 19

5 Detailed Description of Components 20

5.1 Complete Package Diagram 20

5.2 Detailed Descriptions 21

This section will describe the class diagram, package diagram and sequence diagram of the system. 21

5.2.1 Module <Distance measuring among users> 21

5.2.1.1 P001: Package <Distance measuring via bluetooth> 21

5.2.1.2 Class Diagram 22

5.2.1.3 Sequence Diagrams 23

6 Data Design 28

6.1 Data Description - ERD 28

6.2 Data Dictionary 29

7 User Interface Design 31

7.1 Overview of User Interface 31

7.2 User Interface 31

7.2.1 Scanner Interface 31

7.2.1.1 Main Scanning Page 32

7.2.1.2 Setting Page 33

7.2.1.3 Data Analytic Page 34

7.2.2 Health Authorities Interface 35

7.2.2.1 Dashboard Page 35

8 Requirements Matrix 36

1 Introduction

In this world, now the sensor based or the IoT based applications are highly demanding where billions of people have already started relying on it. So, to make the hectic work easier, it has become significant for the current situation of pandemic. So, in such a time of huge pandemic due to the world wide spreaded virus COVID-19 and in the need of the self detecting device for the infected patients, we are proposing this amazing idea of the bluetooth tracing system where the individual have to just keep open the bluetooth and the device itself will notify the organization. Which in turn is really helpful for particular organizations to get this device and start using it.

1.1 Purpose

This Software Design Description includes and describes the system software requirements, system software architecture, detailed description of component, data design, user design interface and requirement matrix of Contact Tracing System. Our targeted audience of our proposed system includes developer team, stakeholder and project manager.

1.2 Scope

Our proposed system is called the Contact Tracing System. It is expected to operate normally as proposed which meets all the functional and non-functional requirements in this system. The goals of our proposed project is listed as following:

1. Identifying the Contact: From the already confirmed positive cases, identifying those that the patient had contact with (according to the transmission modalities of the pathogen).

2. Listing of Contacts: Keep a record of possible contacts of the infected patients and inform those individuals.

3. Contact Follow-Up: A necessary follow-up of the patients that are believed to have come in contact with the infected individuals and those who are positive

The objectives of proposed system is as following:

- Can shorten the time to trace the close contact with the infected users.
- Can help the health authorities to gather all the important information and data which can lead to reduction of spreading of COVID-19.
- Can increase the awareness of users of surrounding if potential users exist

1.3 Definitions, Acronyms and Abbreviations

Acronyms / Abbreviations	**Definitions**
IoT	Internet of Things
COVID-19	Coronavirus disease
SDD	Software Design Description
Beacon	Small Radio Transmitter

1.4 References

I. U.C.L.P.H.D.S.R.G. (2020, November 1). *Automated and partly automated contact tracing: a systematic review to inform the control of COVID-19*. PubMed Central (PMC). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7438082/

II. Karwa, R. (2019). Realtime Indoor Location-Based Passenger Tracking System using Bluetooth Beacon for Airport Authority. International Journal for Research in Applied Science and Engineering Technology, 7(4), 3617–3626. https://doi.org/10.22214/ijraset.2019.4607

III. Lin, Y. W., & Lin, C. Y. (2018). An Interactive Real-Time Locating System Based on Bluetooth Low-Energy Beacon Network †. *Sensors*, *18*(5), 1637. https://doi.org/10.3390/s18051637

1.5 Overview

Our proposed system is a mobile application in which the user must register before using the system and enter the information into the system in which the user will get a broadcast id that is linked with the system database. Scanner used in the system will keep scanning all the broadcast ids that are in the range area of the scanner every 30 seconds. The history list of broadcast ids of users that detected as in close social distance will be recorded for 15 days. If one of the users is verified as infected, the status of the user in the system will need to be changed after notifying the health authorities with the history list of broadcast ids that used to be in close social distance with the infected user. In addition, the infected person will go to do a swab test and the status of the user in the system will be changed to potential infected user until they are proved that negative infected result. At the same time, when the broadcast id is detected by the scanner in the range area, the scanner will notify the health authorities too.

1.6 Real-time profile and modeling tools

The suitable Real-Time Modelling Profile that we choose for our selected project which is Contact Tracing System are UML for Schedulability, Performance and Time (UML-SPT). We choose UML-SPT profile because this profile focuses on schedulability, performance and time whereby it essential elements in our project. In the angle for schedulability, our project emphasizes scheduling tasks which operate regularly by scanning signals that come in range using bluetooth every 30 seconds. For performance, every detected broadcast for infected users will immediately upload to the server and be sent to health authorities in less than one minute. For the tool, we decided to choose Rhapsody tool for UML designing for Contact Tracing System. Basically, Rhapsody tools are commonly used for UML-SPT profile. With the ability to support code generation and animation of diagrams, this tool is suitable for us to build diagrams for use case diagrams, sequence diagrams, statechart and others.

1.7 Innovation solution for IOT and Data analytics

As the issue of Covid-19 pandemic is arising over the world, everyone has to be highly conscious of avoiding every chance that will expose them to any potential infection person surrounding them. In this era of Internet of Things(IoT) and data analytics, we can utilize both technology and be a solution to fight back this pandemic COVID-19. Internet of Things technology is important for this application as this application will rely on the internet to link to each other and gather all the resources that are in the same network. Besides that, using IoT technology, we can exchange all the required information between end-users, administrators and server in which it saves time and cost. In terms of data analytics, the history list that includes the broadcast id of the user, location, length of time that a group of users in close social distance will be sent to the server and be analyzed.

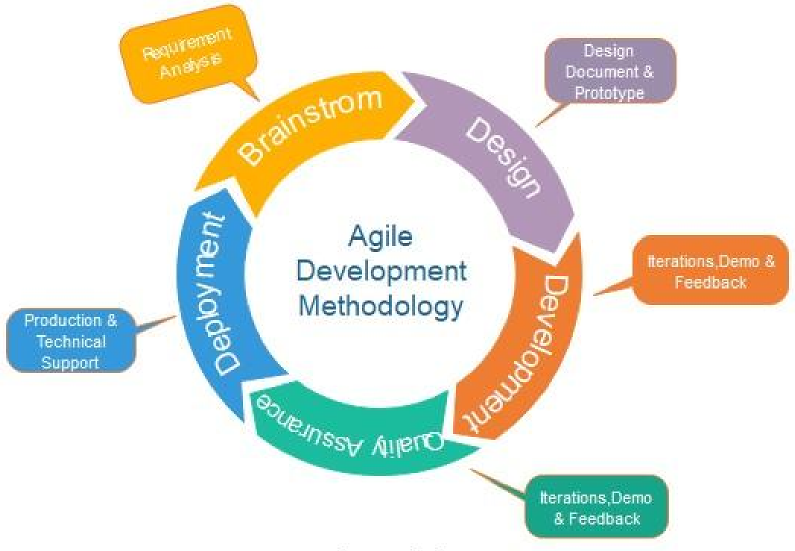
2 System Software Process Model

2.1 Introduction

In this section we selected the software process model is Agile Process Model. That we implement in our system Applicability of mobile contact tracing. Agile process software development approach based on iterative development. Agile processes divide tasks into smaller or more pre-planning iterations. At the starting of the planning process, the project scope and specifications are defined. Plans are well specified in advance with respect to the number of iterations, time, and complexity of each iteration.

Agile model's phases flow:

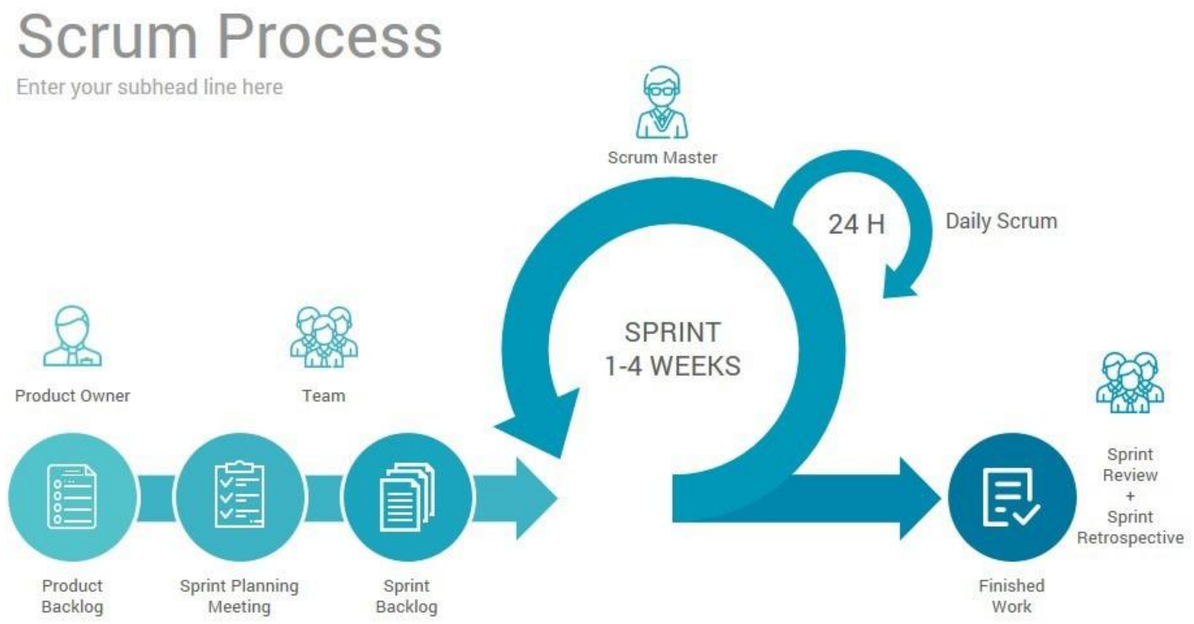
i) Planning
ii) Requirements Analysis
iii) Design
iv) Coding
v) Unit Testing and
vi) Acceptance Testing.


Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Iterations,Demo & Feedback
Development
Iterations,Demo & Feedback
Quality Assurance
Deployment
Production & Technical Support
Agile Development Methodology

Agile method is used so the timeline of the project is defined. This approach will allow the schedule and time required for an advance sprint. This technique further reduces the time required for training and review when tests and tests take place at any sprint. The errors are also not directed to the end product. This further reduces the problems of defect fixation.

2.2 Scrum

For specific software development in agile methodology, our project development will apply scrum project management to ensure our project development can be built smoothly. Apart from project manager/project leader, we will have a scrum master to supervise the development process using scrum method. The tasks will be divided into four sprints where every member will take part in each sprint and finish the tasks in the time given. At the end of each sprint, we will conduct one short meeting with our team members to review what has been done, what will be done until the next meeting and obstacles that occur during the sprint.

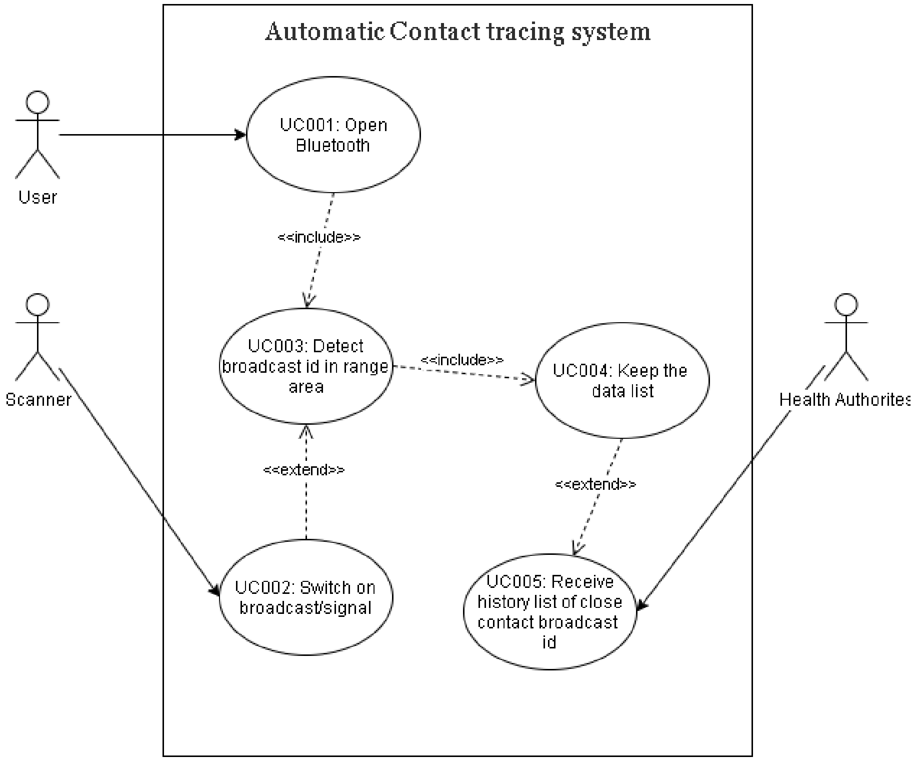


To ease our work, we will use one of the best scrum tools we have experienced with. The tool is called Jira software where this tool can be used to track our project developments and management.

Jira

3. System Software Requirements

3.1 Functional


Automatic Contact tracing system
User
Scanner
Health Authorites
UC001: Open Bluetooth
<<include>>
UC003: Detect broadcast id in range area
<<include>>
UC004: Keep the data list
<<extend>>
<<extend>>
UC002: Switch on broadcast/signal
UC005: Receive history list of close contact broadcast id

1) Users are needed to enable the bluetooth from phones.
2) The scanner will switch on the broadcast scanner to scan the signals in range.
3) The broadcast id locator will scan the available bluetooth signal as broadcast id.
4) The broadcast id locator app will automatically save the broadcast ids in list and upload to server.
5) Health authorities can receive and view the updated broadcast ids list in the dashboard from the server.

3.2 Non-Functional

3.2.1 Performance

1) Automated scan function will scan the broadcast ID with interval time of every 30 seconds.
2) The list of scanned broadcast IDs will be sent to the database server automatically.
3) Database dashboard will refresh every one minute to ensure the data is updated.

3.2.2 Security

1) Each user has access limitations according to each role in this application.
2) The broadcast ID will be confidential and only health authorities can access all the data.

3.3 Constraints

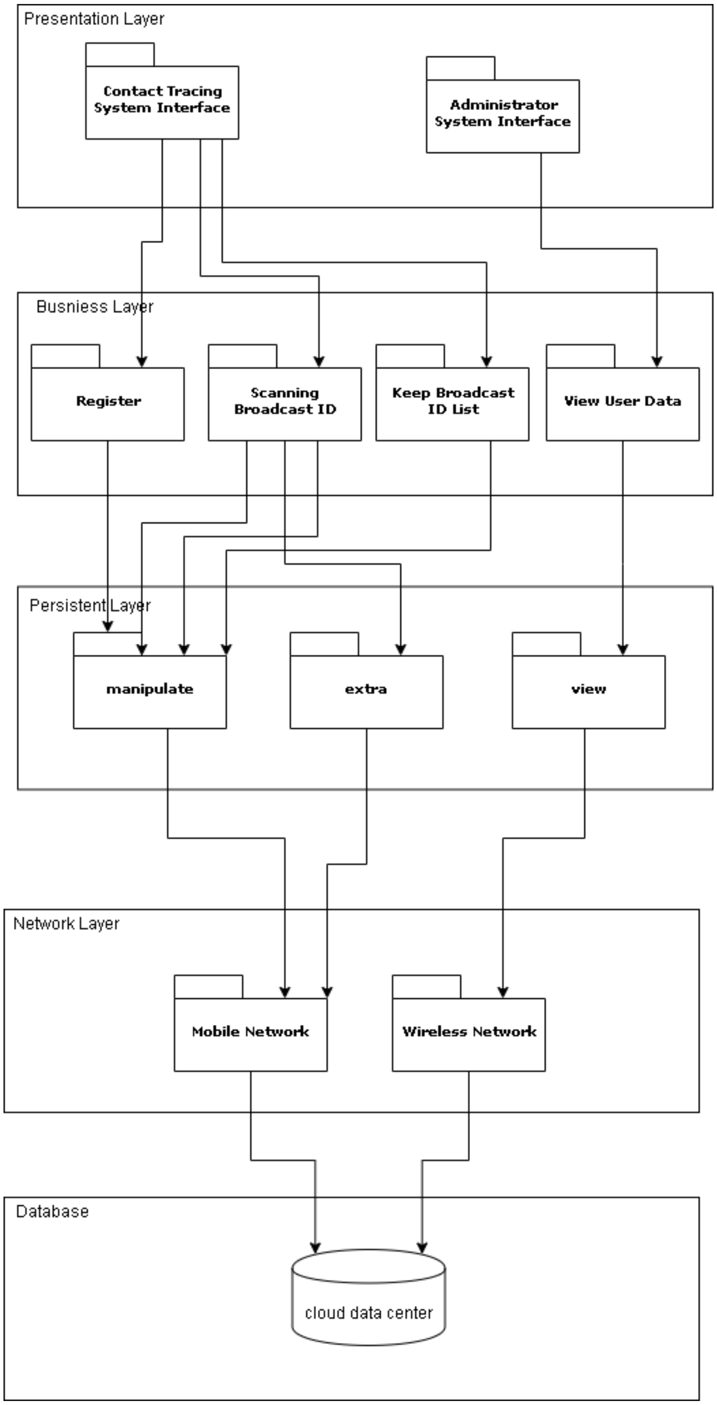
1) Scanner users can only scan the broadcast ID and the broadcast ID will automatically be saved in the database.
2) Scanner users cannot access the scanned broadcast ID details such as phone number and address.
3) General users cannot use the scanner function.
4) Health authorities only can access the broadcast ID details that have been scanned.

3.4 Interface

1) The application will be able to display the user surrounding to check whether it has a potential infected user.
2) The application setup will show the installation path and ask the user for permission to access bluetooth.
3) The system will ask the symptoms of users when registering.

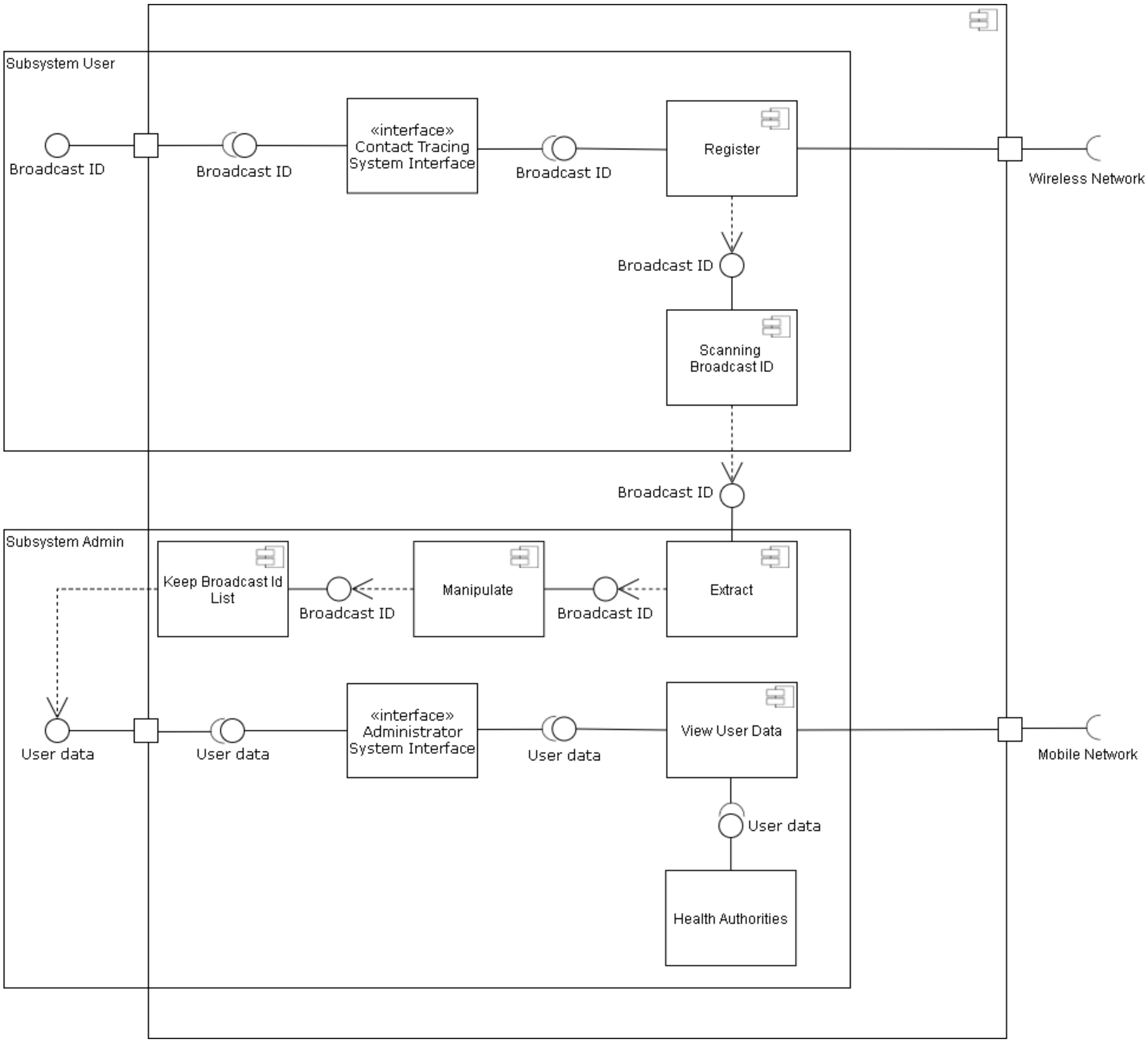
4 System Software Architectural Design

4.1 System Architecture



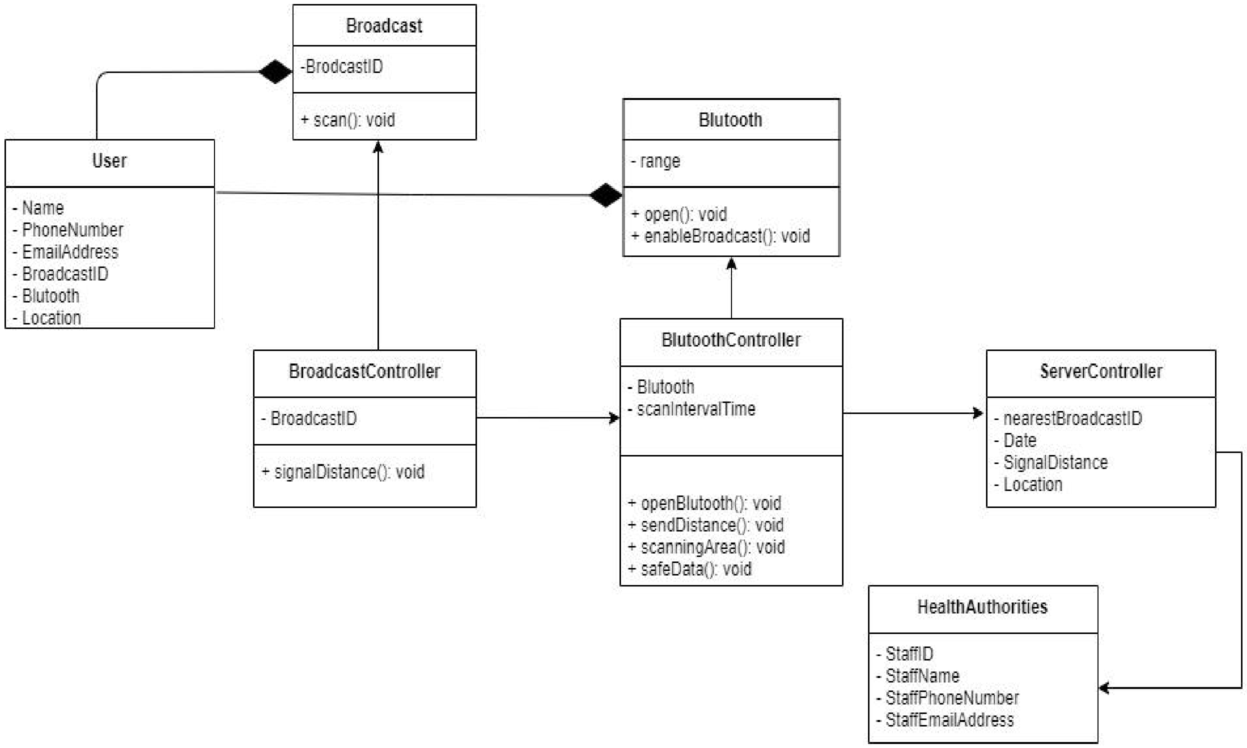
We applied layered architecture as our software architecture style because this pattern is the most well-known software architecture pattern among developers. Besides, this style can provide an easy way for developers to understand and write well-organized code. From the diagram above, it shows that all the codes are splitted into layers which are presentation layer, business layer, persistence layer and database layer where each layer has its own tasks connected in sequence. In the presentation layer, we have two main interfaces which are contact tracing system interface and administrator system interface where the users and administrators interact with the system. In addition, in the business layer, our proposed system will focus on four primary business activities including register, scanning broadcast ID, keep broadcast id list and viewing user data that is sent to the server for administrators. Furthermore, in the persistence layer, all the data being collected in the server will be extracted, manipulated or viewed (access) according to different situations. Next, in the network layer, our system will use the mobile network and wireless network (Bluetooth) to scan the broadcast ID and transfer the data to the server. Lastly, we only have one cloud data center acting as the database in our proposed system in which for storing and fetching purposes.

4.2 Software Architecture – Component Diagram



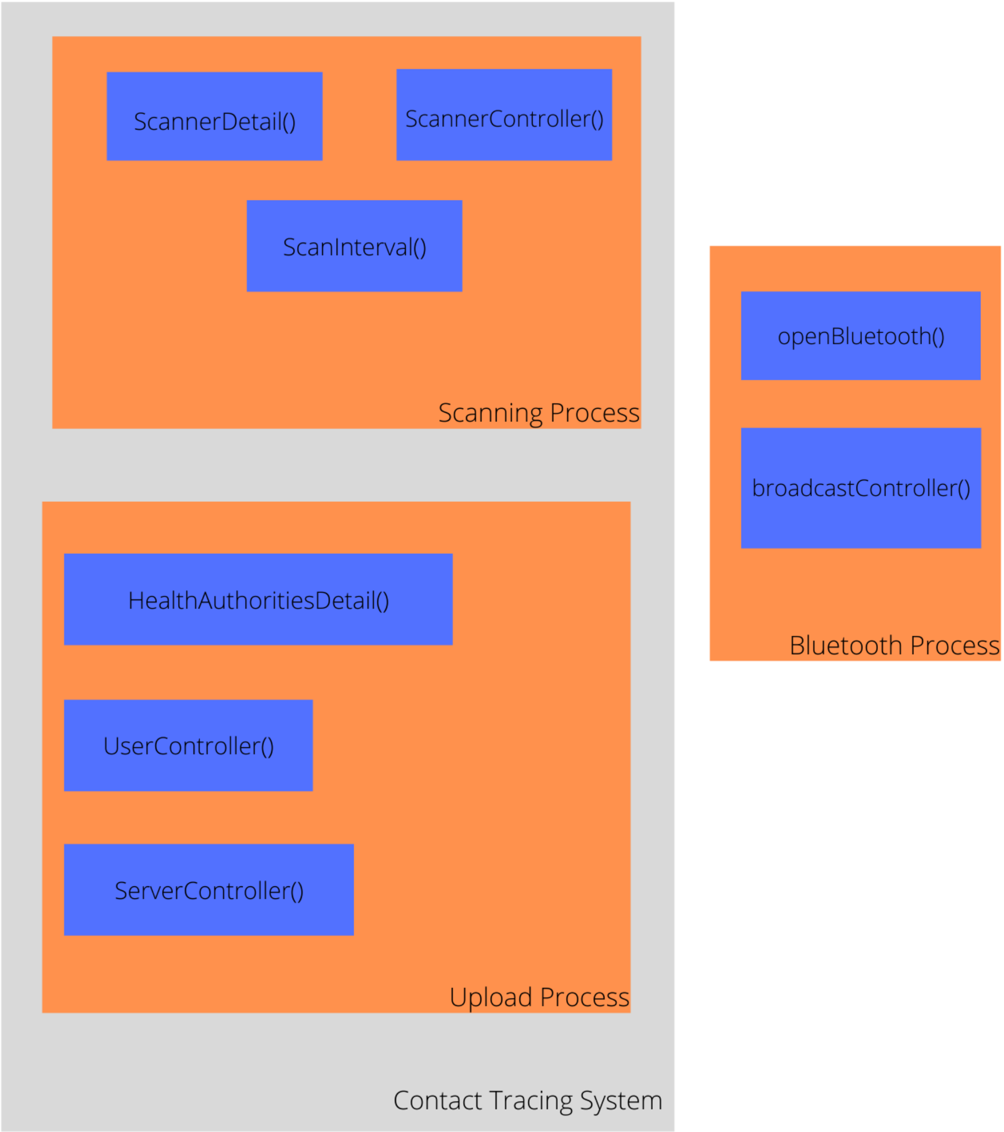
The flow of the component models above, starts from the upper left diagram. Before using this system, the user has to register and get an unique broadcast id. Firstly, the system will scan and detect the user broadcast Id which then will be identified whether the user is categorized as infected user or not. Then the system will start to scan the surrounding broadcast id detected in the range for each 30 seconds. If the broadcast id detected has been identified as an infected user, it will send the data to the server below than one minutes. After that, it will then be extracted and manipulated from the server. After automatically keeping the broadcast id list, it will fetch the user data from the database based on the broadcast id. Next the user data will be transferred to the administrator system interface which can be viewed by the Health Authorities within one minute.

4.3 Class Diagram



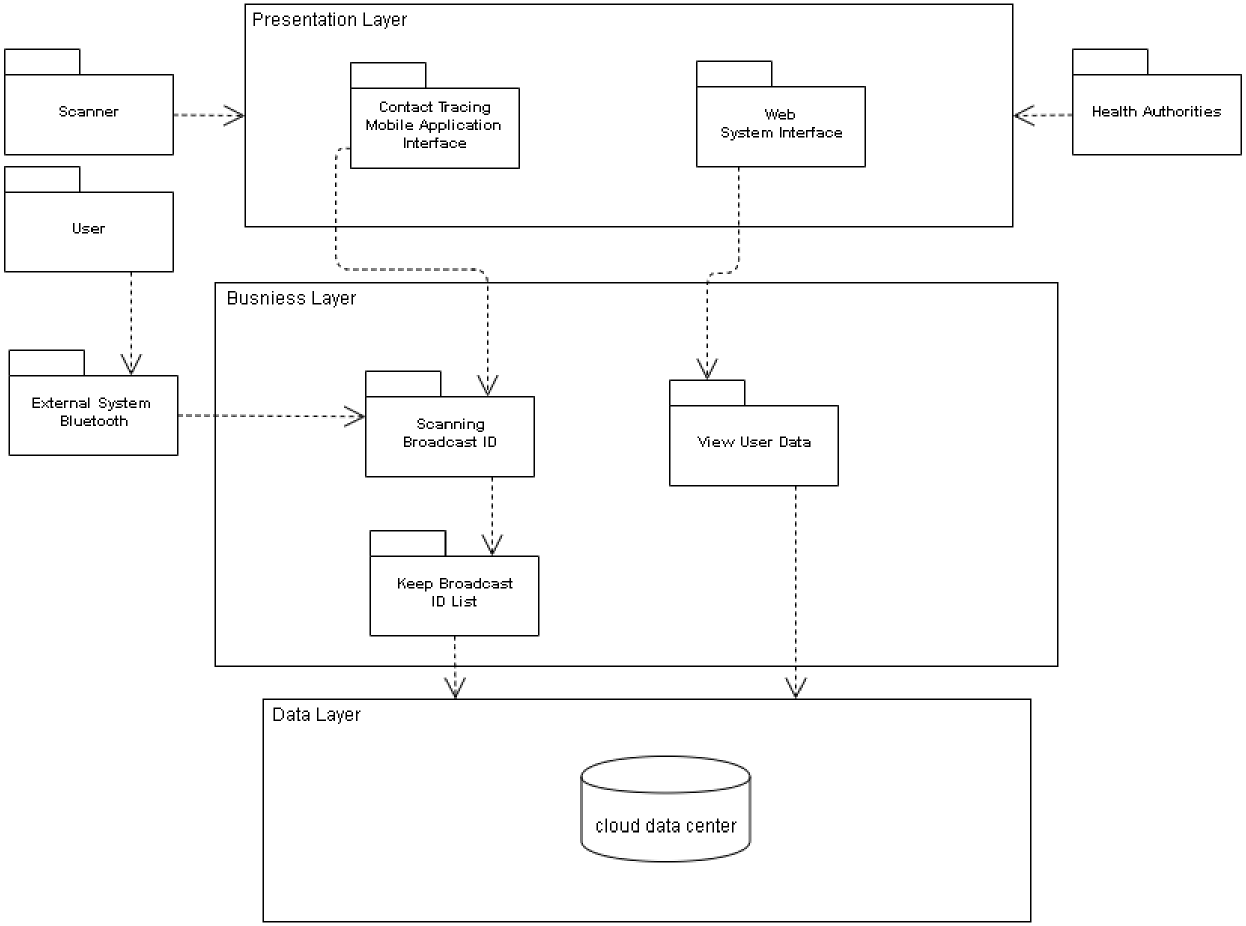
4.4 Concurrency Design

In our program, we will have multiple threads run in a multiple process in a single sistem. We will have scanning process and upload process in our application and we have external process, bluetooth process.



5 Detailed Description of Components

5.1 Complete Package Diagram



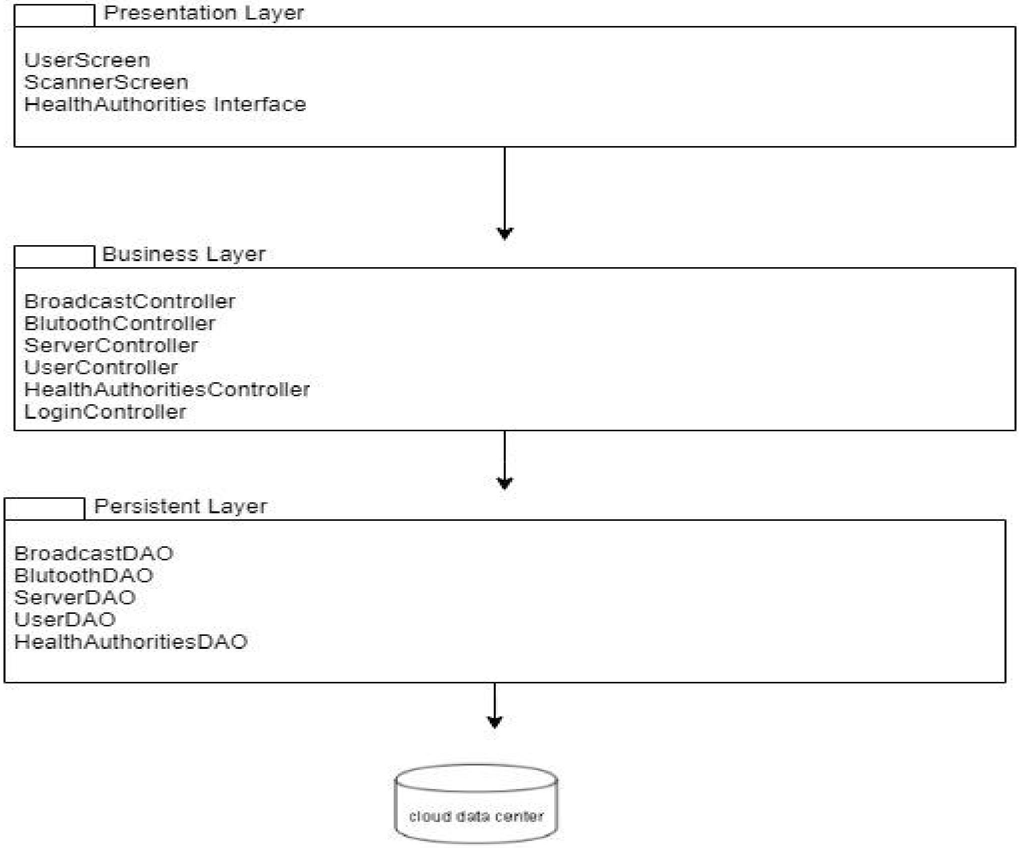
5.2 Detailed Descriptions

This section will describe the class diagram, package diagram and sequence diagram of the system.

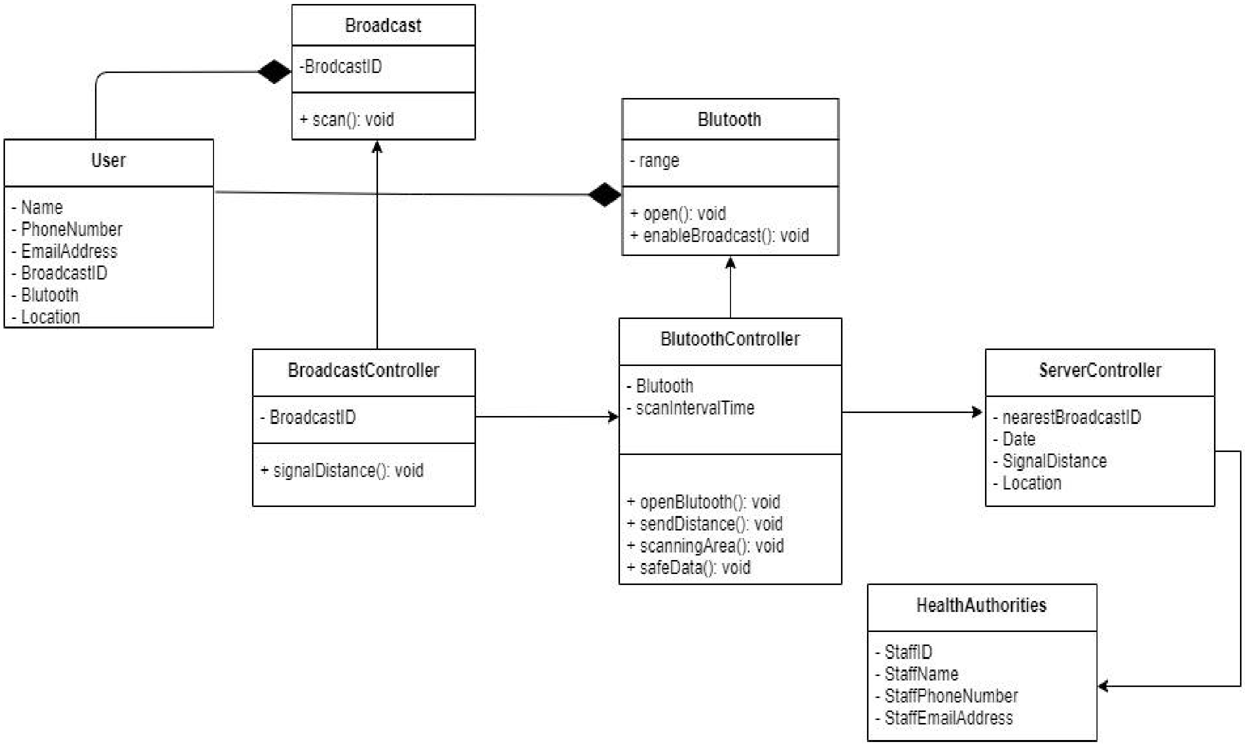
5.2.1 Module <Distance measuring among users>

It is a distance measuring module which would be used to measure the distance among the users, to determine how far they were from each other, so by doing this we would be able to detect that how many users have been come closer to each other during the covid pandemic, which would help to identify the covid patients.

5.2.1.1 P001: Package <Distance measuring via bluetooth>

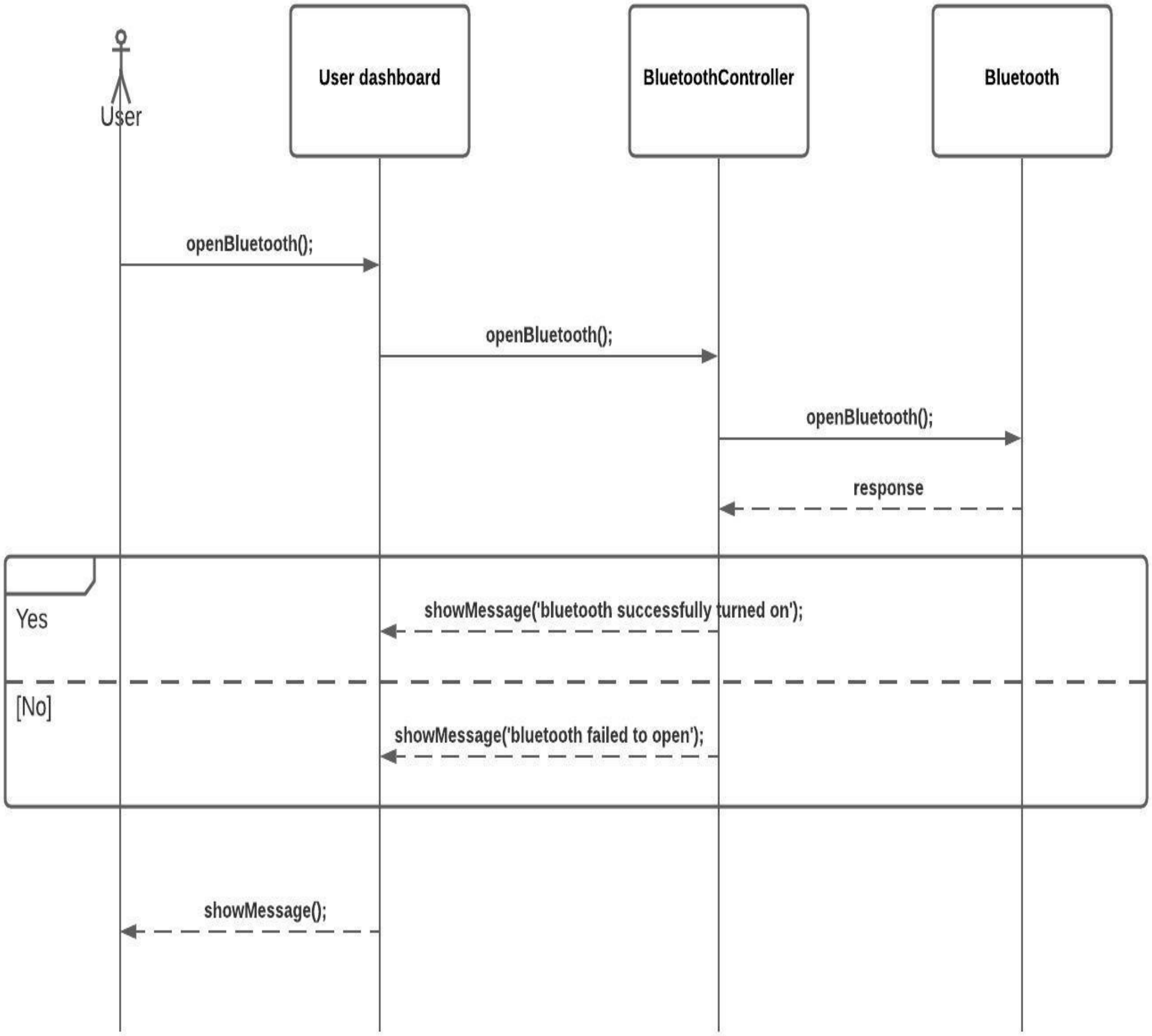


5.2.1.2 Class Diagram

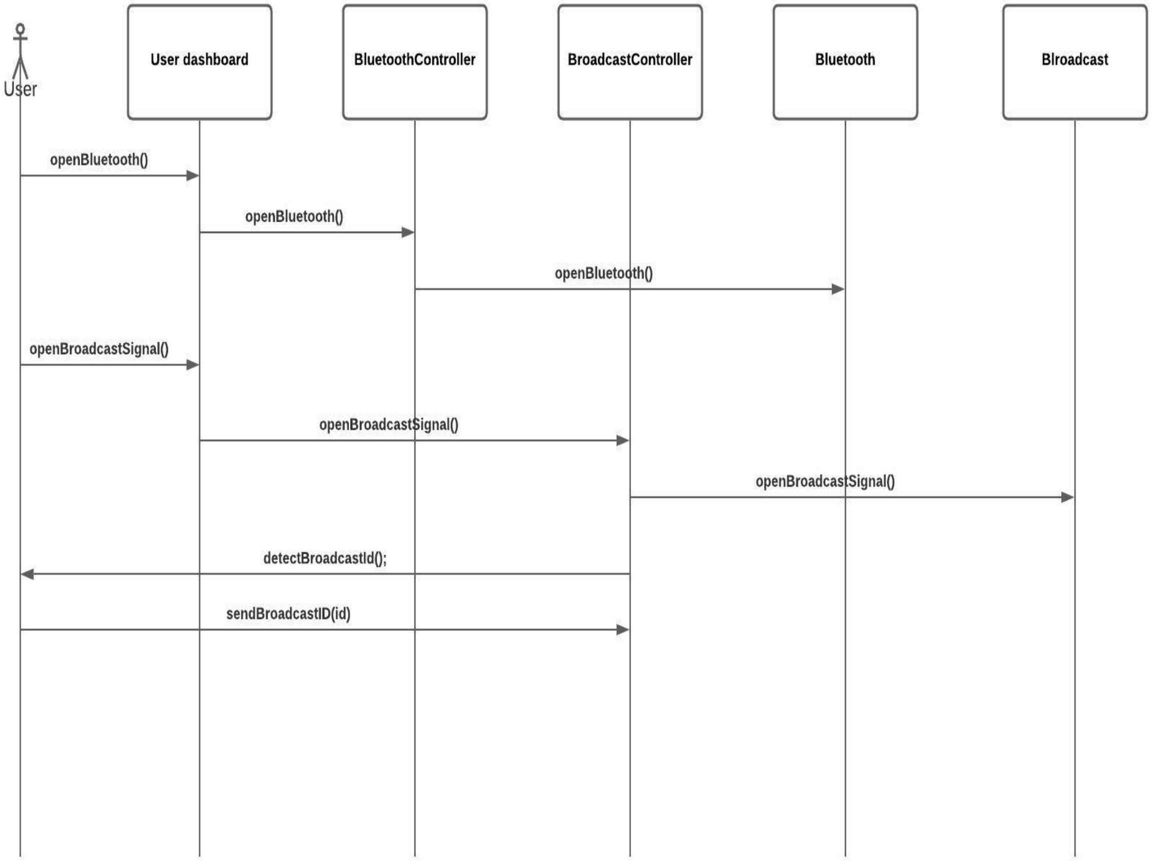


5.2.1.3 Sequence Diagrams

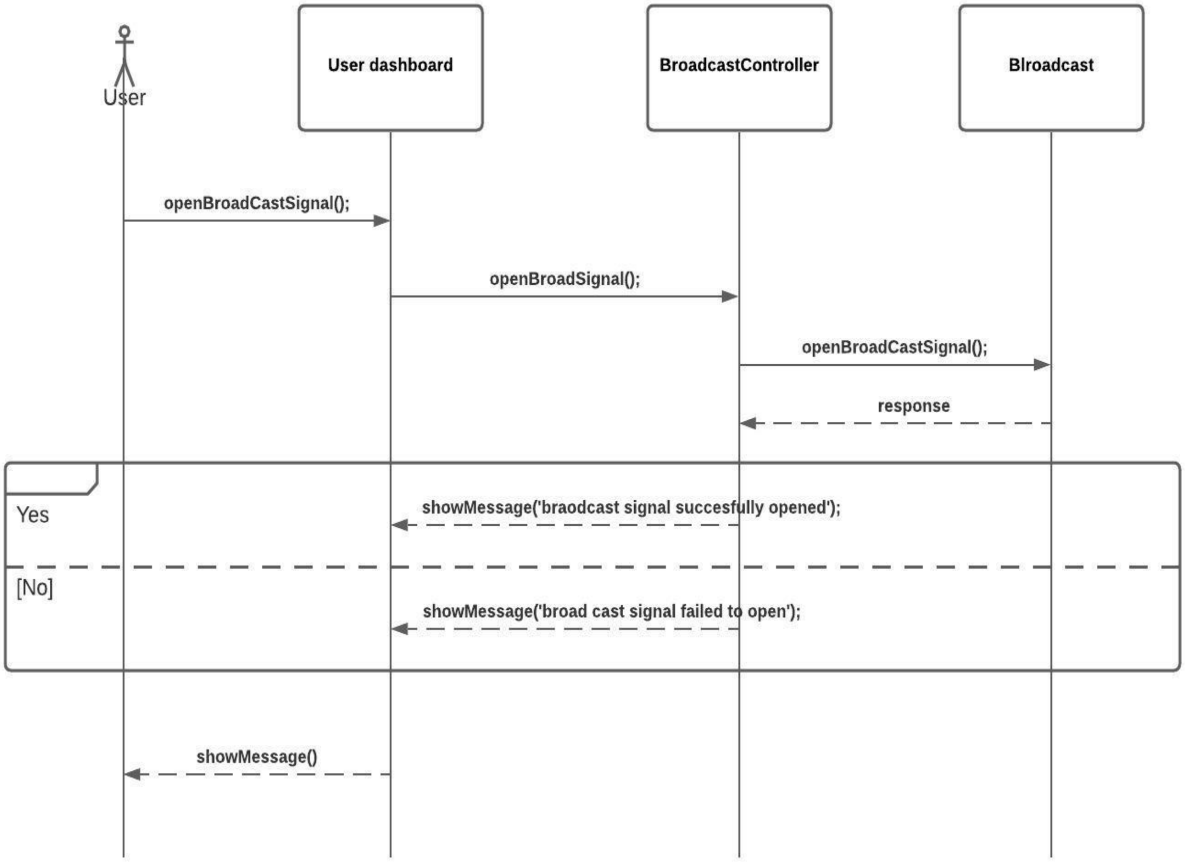
a) SD001: Sequence diagram for Open Bluetooth



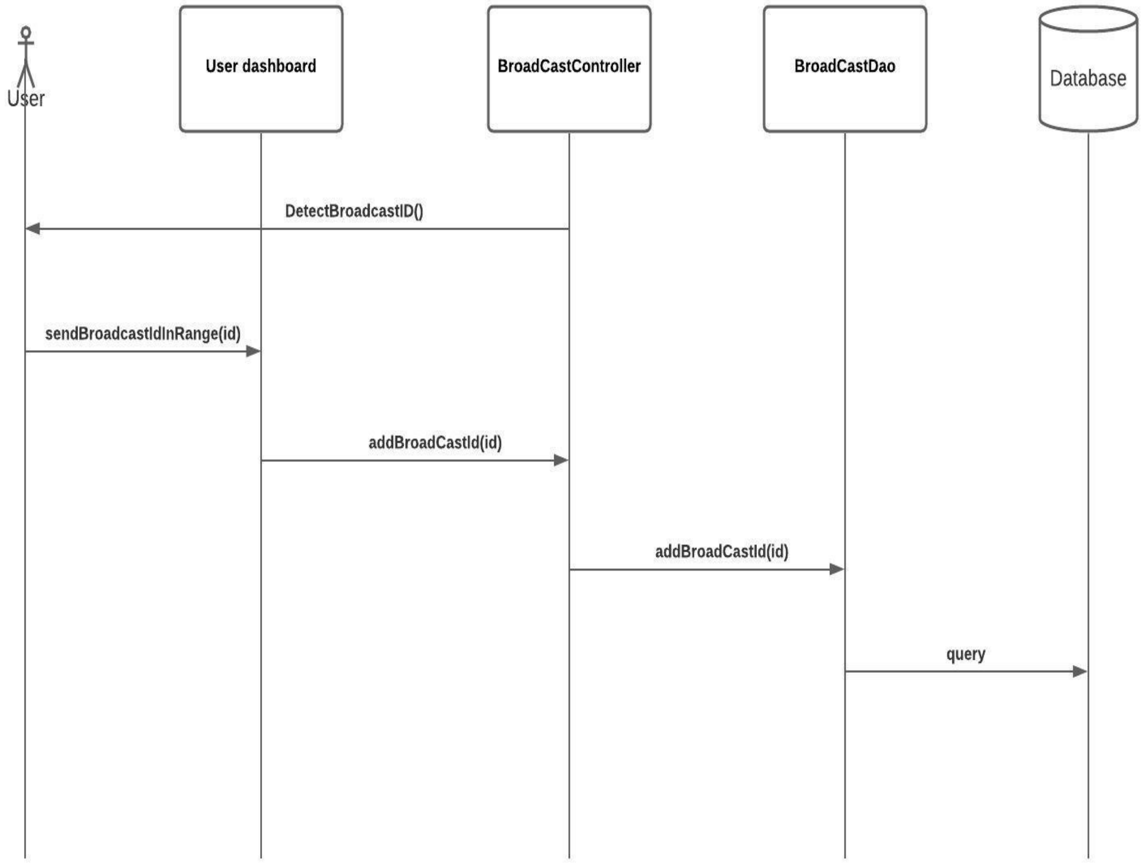
b) SD002: Sequence diagram for Detect Broadcast ID in range Area



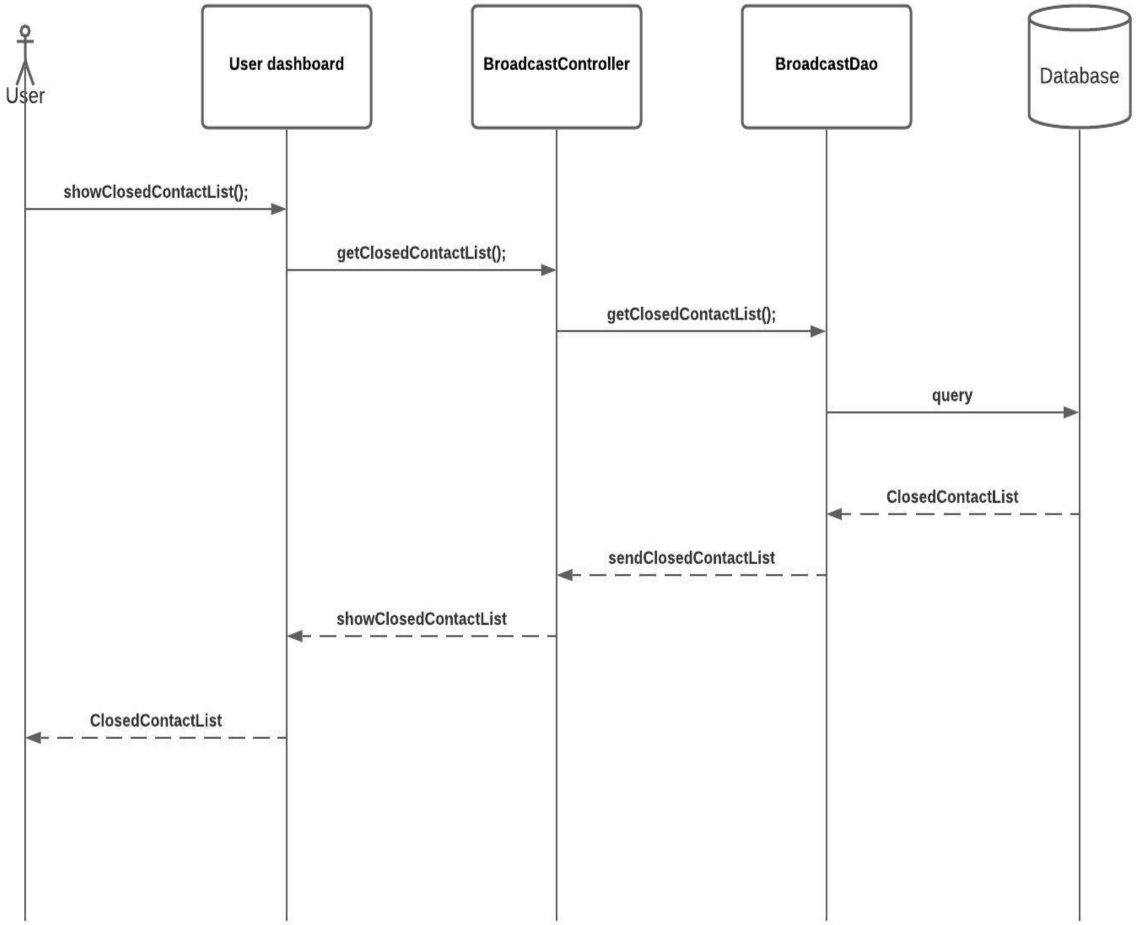
c) SD003: Sequence diagram for Open Broadcast Signal



d) SD004: Sequence diagram for Add data into database

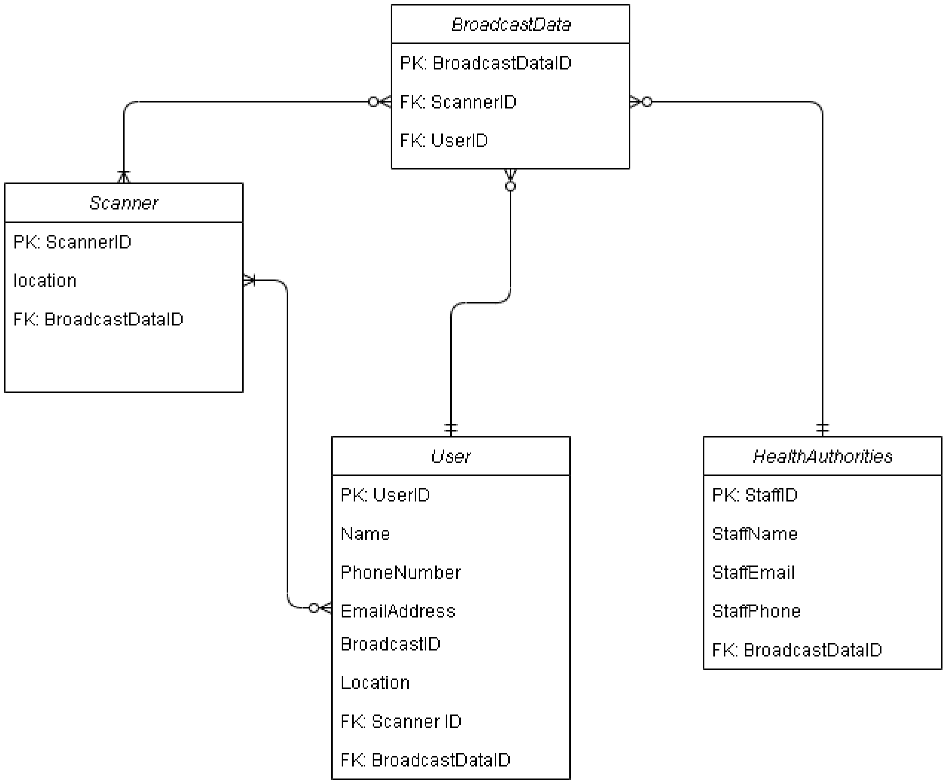


e) SD005: Sequence diagram for Get closed contact broadcast IDs



6 Data Design

6.1 Data Description - ERD



6.2 Data Dictionary

Attributes	Data Types	Size	Description
user_name	Varchar	60	Used for register account
PhoneNumber	INT	20	Used for register account
Emailaddress	Varchar	30	Used for register account
BroadcastID	INT	15	register and get an unique broadcast id
Location	Varchar	40	Address

Attributes	Data Types	Size	Description
staffID	INT	10	staffID for staff to log in to the account
staffName	Varchar	60	staffName for staff to log in to the account
staffEmail	Varchar	30	staffEmail for staff to log in to the account
staffPhone	INT	15	staffPhone for staff to log in to the account

Attributes	Data Types	Size	Description
scannerID	INT	15	According to scannerID will notify the health authorities.
location	Varchar	40	Address

Attributes	Data Types	Size	Description
BroadcastDataID	INT	15	Broadcast data id detected in the range for each 30 seconds
signalDistance	INT	10	scanner will switch on the broadcast scanner to scan the signals in range

7 User Interface Design

7.1 Overview of User Interface

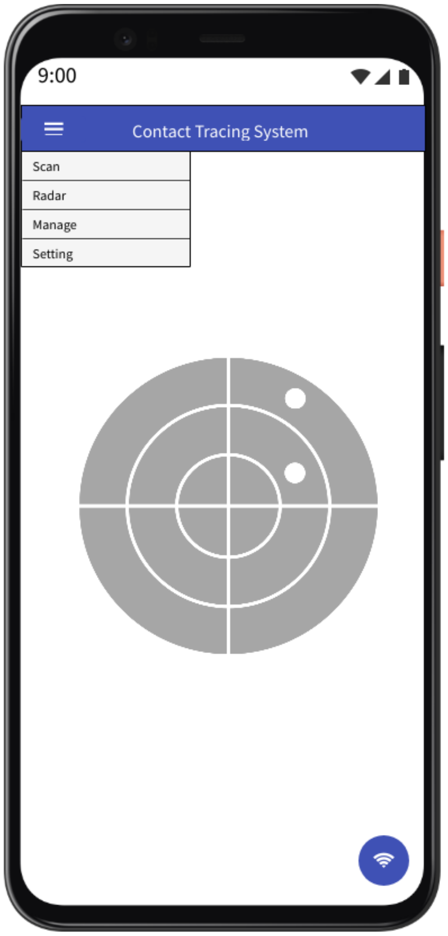
Mobile contact tracing systems will have three types of users where the first one is the general user, scanner user and last one is health authorities. From the general user perspective, they will need to have a mobile device which already installed the simulator app which will broadcast the bluetooth signals.

For our main user interface in our proposed system, we have a main scanning page, data analytic graphs page,, setting page in mobile application and dashboard that shows data analytic graphs in browser.

7.2 User Interface

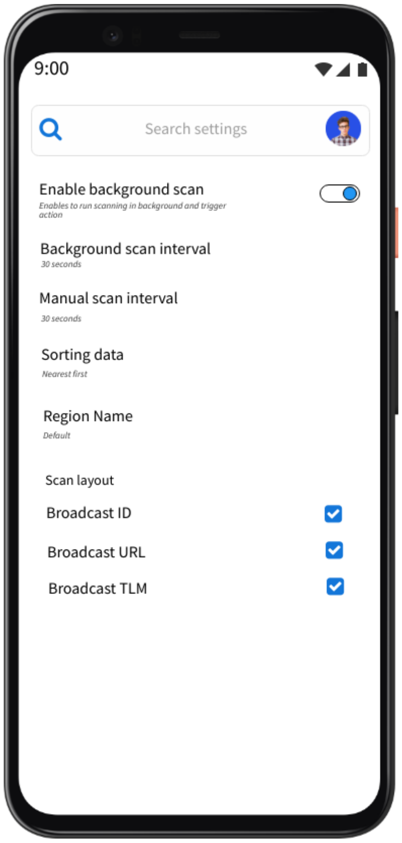
7.2.1 Scanner Interface

7.2.1.1 Main Scanning Page



This is the main page of scanner users. In this page it has the main function which is a manual scanner. The button at the right bottom is used for the user to click to manually start to scan the region for all available beacon bluetooth signals. On top of the page, it has the menu which contains the function scan, radar, manage and setting.

7.2.1.2 Setting Page



Enable background scan ●
Enables to run scanning in background and trigger action

Background scan interval
30 seconds

Manual scan interval
30 seconds

Sorting data
Nearest first

Region Name
Default

Scan layout

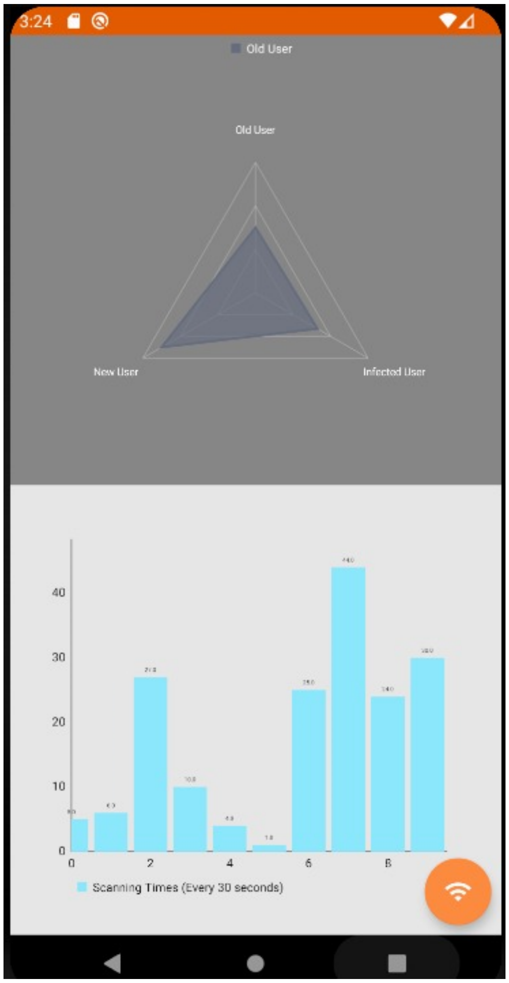
Broadcast ID ☑

Broadcast URL ☑

Broadcast TLM ☑

This is the setting page for our proposed system in which the important settings here include the time interval for the background scanning, region name and the scan layout where to toggle what bluetooth signals should be shown in the scanning page. Users also can set timing for automatic or background scan interval and manual scan interval in this setting page.

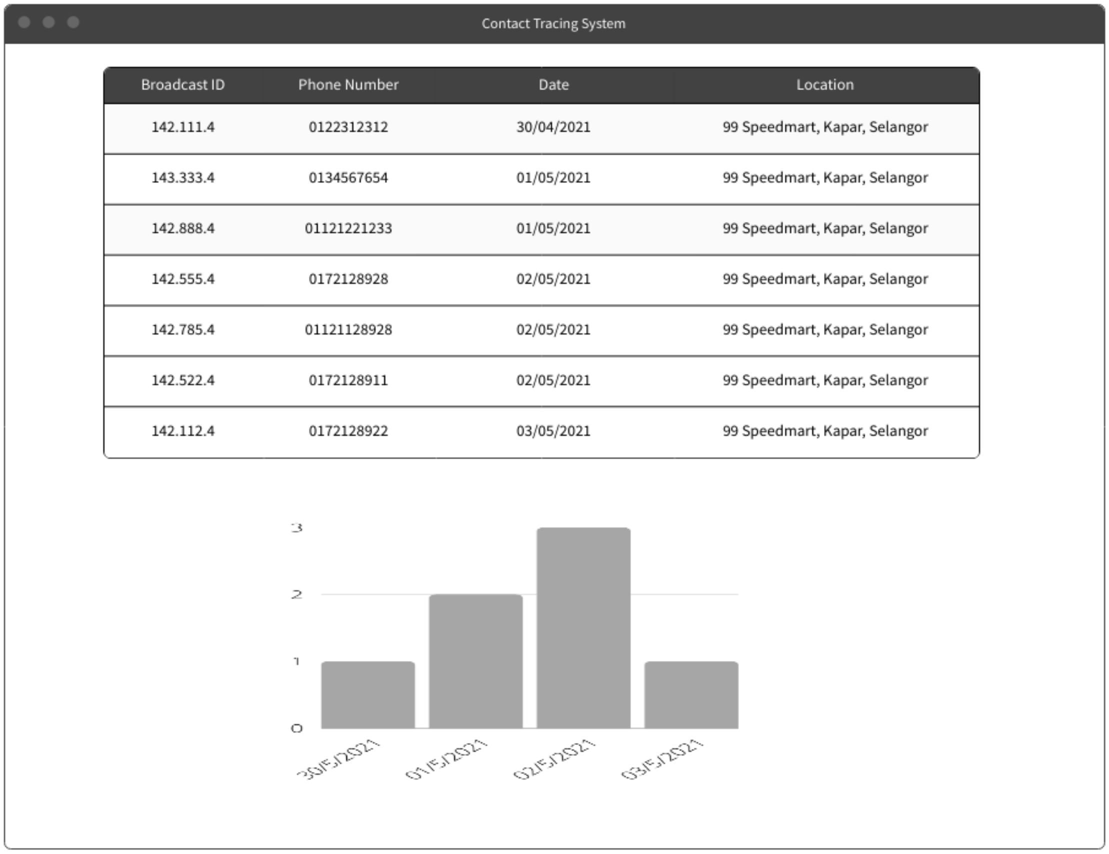
7.2.1.3 Data Analytic Page



This page is to show the data analytics of the total broadcast id scanned in real time. For the graph showing in the upper part, it is to show the ratio among the types of users detected in the region. Then, the graph in lower part is to show the number of all users in each scanning every 30 seconds.

7.2.2 Health Authorities Interface

7.2.2.1 Dashboard Page



Contact Tracing System

Broadcast ID	Phone Number	Date	Location
142.111.4	0122312312	30/04/2021	99 Speedmart, Kapar, Selangor
143.333.4	0134567654	01/05/2021	99 Speedmart, Kapar, Selangor
142.888.4	01121221233	01/05/2021	99 Speedmart, Kapar, Selangor
142.555.4	0172128928	02/05/2021	99 Speedmart, Kapar, Selangor
142.785.4	01121128928	02/05/2021	99 Speedmart, Kapar, Selangor
142.522.4	0172128911	02/05/2021	99 Speedmart, Kapar, Selangor
142.112.4	0172128922	03/05/2021	99 Speedmart, Kapar, Selangor

This is the main page of health authorities where this page contains the data list of users who are detected and stored in the database. This is to show the history record which is useful when health authorities want to check the close contact user list with a certain identified broadcast id or infected user. From the data in table, it will auto generate a bar graph of the history record based on the date.

8 Requirements Matrix

The use cases and respective packages are as below.

	P001
Module 1, UC001	X
Module 1, UC002	X
Module 1, UC003	X
Module 2, UC004	X
Module 2, UC005	X

The packages and respective sequence diagrams for the scenarios are as below.

	SD001	SD002	SD003	SD004	SD005
P001	X	X	X	X	X

The sequence diagrams and respective classes are as below.

	SD001	SD002	SD003	SD004	SD005
User	x				x
Broadcast		x	x		x
BroadcastController		x	x		x
Bluetooth	x	x			
BluetoothController	x	x			
ServerController				x	
Health Authorities				x	