

**HS 8001 – PROFESSIONAL READINESS FOR
INNOVATION,EMPLOYABILITY AND ENTREPRENEURSHIP**

CONTAINMENT ZONE ALERTING APPLICATION

A PROJECT REPORT

Submitted by

TEAM ID:PNT2022TMID50125

BALAMURUGAN P

BHARATHRAJ S

EBANESH S

ESAKKIMUTHU K

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

JAYARAJ ANNAPACKIAM C.S.I COLLEGE OF

ENGINEERING NAZARETH.

ANNA UNIVERSITY::CHENNAI 600 025

NOV/DEC 2022

ABSTRACT

The World Health Organization has declared the outbreak of the novel coronavirus, Covid-19 as pandemic across the world. With its alarming surge of affected cases throughout the world, lockdown, and awareness (social distancing, use of masks etc.) among people are found to be the only means for restricting the community transmission. In a densely populated country like India, it is very difficult to prevent the community transmission even during lockdown without social awareness and precautionary measures taken by the people. Recently, several containment zones had been identified throughout the country and divided into red, orange and green zones, respectively. The red zones indicate the infection hotspots, orange zones denote some infection and green zones indicate an area with no infection. This paper mainly focuses on development of an Android application which can inform people of the Covid-19 containment zones and prevent trespassing into these zones. This Android application updates the locations of the areas in a Google map which are identified to be the containment zones. The application also notifies the users if they have entered a containment zone and uploads the user's IMEI number to the online database. To achieve all these functionalities, many tools, and APIs from Google like Firebase and Defencing API are used in this application. Therefore, this application can be used as a tool for creating further social awareness about the arising need of precautionary measures to be taken by the people of India.

TABLE OF CONTENTS:

CHAPTER	CONTENTS	PAGE NO
1	1.INTRODUCTION	5
	1.1.Project Overview	
	1.2. Purpose	
2	2.LITERATURE SURVEY	7
	2.1.Existing problem	
	2.2.References	
	2.3.Problem Statement Definition	
3	3.IDEATION & PROPOSED SOLUTION	11
	3.1 Empathy Map Canvas	
	3.2 Ideation & Brainstorming	
	3.3 Proposed Solution	
	3.4 Problem Solution Fit	
4	4. REQUIREMENT ANALYSIS	13
	4.1 Functional requirement	
	4.2 Non-Functional requirements	
5	5. PROJECT DESIGN	15
	5.1 Data Flow Diagrams	
	5.2 Solution & Technical Architecture	
	5.3 User Stories	
6	6. PROJECT PLANNING & SCHEDULING	17
	6.1 Sprint Planning & Estimation	
	6.2 Sprint Delivery Schedule	
	6.3 Reports from JIRA	

7	7. CODING & SOLUTIONING	18
	7.1 Feature 1	
	7.2 Feature 2	
	7.3 Database Schema	
8	8. TESTING	20
	8.1 Test Cases	
	8.2 User Acceptance Testing	
9	9. RESULTS	21
	9.1 Performance Metrics	
10	10. ADVANTAGES & DISADVANTAGES	22
11	11. CONCLUSION	23
12	12. FUTURE SCOPE	24
13	13. APPENDIX	25

CHAPTER 1

INTRODUCTION

1.1. Project Overview

Currently there are several research works undergoing in the country to prevent Covid-19 cases from rising. Previously our country was importing medical kits like PPE (Personal Protection Kits), mask from outside, but now it has been successful in developing these kits. Along with taking initiatives to fight this disease, our country has also taken steps to make people aware of the disease. The news and media have a great part in creating this awareness by informing the public about the preventive measures that can keep them away from infection. Awareness among the people to carry out all the preventive measures can immensely help to reduce spread of the virus. The country has created containment zones throughout the cities wherever Covid-19 cases have been reported to prevent further spread of the virus. These containment zones have been kept isolated from the outside public to ensure no contamination occurs outside.

After more than 2 months of the lockdown, the government has relaxed some of the lockdown rules and has permitted reopening of government offices, bus and other road transportation facilities and shopping markets. People can move inside the city for work and other purposes. But the containment zones are still being kept isolated, and new containment zones are being formed wherever Covid-19 cases have been reported. These zones are highly contagious as droplets with virus coughed out from an unscreened asymptomatic patient can travel up to 8 m (Bahl et al. 2020). Though these containment zones are guarded by policemen, still there remains a chance that people might unknowingly step into them. In this situation where people can move in the city, these containment zones pose a risk of infection to these city dwellers. Therefore, informing people about the location of the containment zones can help them bypass and avoid these zones and thereby reduce the chance of community transmission. In this paper, we focus on developing a mobile based application to provide information regarding the Covid-19 containment zones in West Bengal. The application further tracks the user's location and provides notification alert if the user has entered a containment zone.

The application also provides daily Covid-19 case statistics to the users to keep them updated. The application is developed on Android SDK and uses Firebase Cloud Firestore to store the location data. Android's geofencing client is used to create geofences around the containment zones and notification manager is used to provide notifications. The application also uses RESTful web services to show the

Covid-19 cases in West Bengal. We have tested our application with different users in different locations across West Bengal and it works efficiently and is able to attain our target.

1.2. Purpose

The Android application shows the location of the containment zones to the users. It also notifies the user when he or she trespasses the boundary of a containment zone or stays in the containment zones

The application uses Fire store which is a flexible and scalable database for mobile, web and server developments from Firebase and Google cloud platform (Cloud Fire store 2020). In Cloud Fire store, the mobile application supports server less app architecture where the application connects to the Cloud Fire store database directly without any intermediate servers in between (Cloud Fire store SDKs and client libraries 2020). The application receives data from the database using Web. The Cloud Fire store does not always fetch data from the database unless the data has been changed, it gathers previous data from the cache memory which also enables its offline functioning.

The Cloud Fire store features a NoSQL, document-oriented database (Cloud Firestore Data model 2020) and the data is stored in a JavaScript Object Notation (JSON) format. The location data are stored in documents, which are organized into collections. All the containment zones are stored in a collection in which each containment zone is represented as an individual document. Each document has four fields namely “Lat”, “Long”, “locationName” and “radius” for storing latitudes, longitudes, location names and radius respectively. Figure 5 shows the documentoriented Cloud Firestore database with data of few containment zones. The “radius” field in each document is used to indicate the radius of the containment zone. In the development stage, the billing plan for the database used here is the frebase spark plan which is the free plan provided by Firebase. This plan has few limitations like 50,000 and 20,000 reads and writes per day. These usage limits can be resolved by choosing a different paid plan from Firebase according to requirement.

CHAPTER 2

LITERATURE SURVEY

2.1.Existing problem

[1]Currently there are several research works undergoing in the country to prevent Covid-19 cases from rising. Previously our country was importing medical kits like PPE (Personal Protection Kits), mask from outside, but now it has been successful in developing these kits. Along with taking initiatives to fight this disease, our country has also taken steps to make people aware of the disease. The news and media have a great part in creating this awareness by informing the public about the preventive measures that can keep them away from infection. Awareness among the people to carry out all the preventive measures can immensely help to reduce spread of the virus. The country has created containment zones throughout the cities wherever Covid-19 cases have been reported to prevent further spread of the virus. These containment zones have been kept isolated from the outside public to ensure no contamination occurs outside.

[2]According to (Wollersheim, 2020) during the COVID-19 crisis the field of Data Science is in center. Most of the community is interested, watching and looking forward the statistical analysis and epidemiology graphs and sharing the same in social media on a large scale. The expectation from Data Science is very high. Data Science is emerging field consist of number of applicable and useful tools, techniques and functions, using which taking the fact-based decisions and planning can be possible, which is very essential in current situation.

The cluster containment strategy for Zika virus outbreak (Singh et al., 2019) was found effective in Rajasthan, India. Singh et al (2019) in their paper explained that how surveillance strategies used to control the disease from spreading beyond containment zones of 3 km radius. The article gives emphasis on creating to containments to prevent the outburst of disease, however it does not explain about how to make these zones quickly and accurately. In their paper (Maier & Brockmann, 2020) explained about the effective containment to control specifically COVID-19 cases in China. The model which they explained in their paper captures both quarantine of symptomatic infected individuals and other population isolation practices. The focus of the research is on contagion process and general effects as well as significance of the containment. Their research work implies and supports the need to define the containment zones accurately.

As stated in old article of Teena (2020), the Government of India had given a broad guidelines to classify the containment zones in three types as Green-Zone (if there are no confirmed cases or no report of cases since last 21 days), Orange-Zone (where zonal retractions can be relaxed based on situation) and RedZone (containment zone where strict lock-down can be imposed). As per Teena (2020), Government asked district administration to demarcate the containment areas with red and orange zones around connection with the Coronavirus outbreak boundary of containment zones as colony, mohalla, ward and police station area etc. Which support the need to micro-level defining and updating the containment zones.

[3]The World Health Organization has declared the outbreak of the novel coronavirus, Covid-19 as pandemic across the world. With its alarming surge of affected cases throughout the world, lockdown, and awareness (social distancing, use of masks etc.) among people are found to be the only means for restricting the community transmission. In a densely populated country like India, it is very difficult to prevent the community transmission even during lockdown without social awareness and precautionary measures taken by the people. Recently, several containment zones had been identified throughout the country and divided into red, orange and green zones, respectively. The red zones indicate the infection hotspots, orange zones denote some infection and green zones indicate an area with no infection.

This paper mainly focuses on development of an Android application which can inform people of the Covid-19 containment zones and prevent trespassing into these zones. This Android application updates the locations of the areas in a Google map which are identified to be the containment zones. The application also notifies the users if they have entered a containment zone and uploads the user's IMEI number to the online database. To achieve all these functionalities, many tools, and APIs from Google like Firebase and Geofencing API are used in this application. Therefore, this application can be used as a tool for creating further social awareness about the arising need of precautionary measures to be taken by the people of India.

[4]Since its outbreak in December 2019, the number of COVID-19 cases has been rapidly increasing across the world. As of 21st August 2020, there were more than 226 million of confirmed cases worldwide, including more than 7.9 million deaths.¹ The most affected countries were the USA (55 million cases), Brazil (35 million cases), India (29 million cases), the UK (3 million cases), Saudi Arabia (3 million cases), and Italy (2.5 million cases).¹ However, the recovery rates differ significantly across the countries due to various factors such as health interventions, effective planning and methods for managing the outbreak.² The

COVID-19 outbreak has severely impacted various industries, being the healthcare industry one of the most affected. It faces a severe burden with the allocation of resources, delivering services and containing the spread of the COVID-19 virus. ³⁻⁵ In order to minimize the impact of COVID-19 on the healthcare industry, improve the delivery of healthcare services, and facilitate the process of returning to normal life, countries are developing various strategies.

Adopting health interventions using innovative technologies such as mobile health applications integrated with Bluetooth, global positioning system (GPS), artificial intelligence (AI), and machine learning (ML) techniques can significantly improve the delivery of healthcare services remotely while following preventive measures such as social distancing and home quarantine.

2.2References

- [1] M. Ranajoy, "Department of Instrumentation and electronics Engineering," Jadaypur University, Kolkatta, 2010.
- [2] D. chinchorkar, "Research square," Jhonny Steaphan, Andra pradhesh, 2020.
- [3] M. V. Rane, "Application for Covid 19 real time counter," jesu denison industries, India, 2022.
- [4] A. t, "Health Information Management and Technology Department," Iman Abdulrahuman Bin Fasil University, Saudi Arabia, 2020.

2.3.Problem Statement Definition

Problem Statement 1:

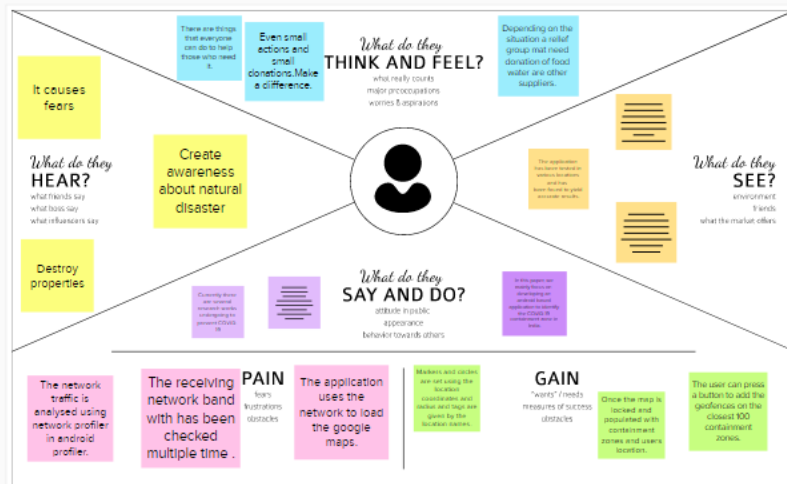


CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

CONTAINMENT ZONE ALERTING APPLICATION



Reference link:

<https://app.mural.co/invitation/mural/singlemugeshgmailcom7089/1664166908862?sender=u4f790a49a432e7a021871338&key=38b5d5c5-ca45-439c-ac74-6e0804852299>

3.2 Ideation & Brainstorming

This application is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individual's location. Key benefits of the application are monitoring people's activity and alerting them of their safety movements.

3.3 Proposed Solution

The Android application shows the location of the containment zones to the users. It also notifies the user when he or she trespasses the boundary of a containment zone or stays in the containment zones.

3.4 Problem Solution Fit

Defining CS, fit into CC	1. CUSTOMER SEGMENT(S) The user/customer who belongs to the business man	6. CUSTOMER CONSTRAINTS There is a foundation of using this application because the user/customer who is having knowledge of this application can work on it easily	5. AVAILABLE SOLUTIONS We can use google maps and GPS to show which area is least cases and more cases and other instructions, to alert the people in the zone and the public knowledge
	2. JOBS-TO-BE-DONE / PROBLEMS It is easy to analyze the issues and risks in containment zones. It is a best way to assist the peoples easily to identify the disaster region and prevented from danger. Extension and recognition of risk zones using cloud computing are very efficient in providing information about containment zones at its earliest.	9. PROBLEM ROOT CAUSE Generally, we cannot identify the number of cases on area in the particular location. Whether it is a red zone or normal zone or any instruction to survive on the particular area.	7. BEHAVIOUR Easy to use Can be able to respond quickly Able to provide precise decision based on the disease Analysis Reinforcement of others' need
Focus on J&P / BE into BE	3. TRIGGERS Movement in containment zones will be monitored to ensure that nobody leaves or visit, except for medical emergencies	10. YOUR SOLUTION The application is built which uses this model. The application updates you to stay up to date regarding the number of cases, both locally and nationally. The accurate numbers can help you assess your risk further	8. CHANNELS OF BEHAVIOUR 8.1 ONLINE Customers can be able to identify their zone is disaster or not. Using google map and GPS in the map to know the cases in the containment zone. 8.2 OFFLINE Store the data and information is being stored
	4. EMOTIONS: BEFORE / AFTER Before – The user/customer who never have used before makes them nervous After – As the user knows how to use this application then they will become comfortable and friendly in environment		
Idea fit's strong TR & EM			

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub- Task)
FR-1	User Registration	Registration through Gmail. Registration through mobile number.
FR-2	User Confirmation	Confirmation via Email. Confirmation via OTP.
FR-3	Authentication	It checking the confirmation of the password.
FR-4	Business rule	For subscriber's we give first 3 day's free trail. For un subscriber's the user needs to watch some advertisement for knowing the zone alert for first 3 day's.

4.2 Non-Functional requirements

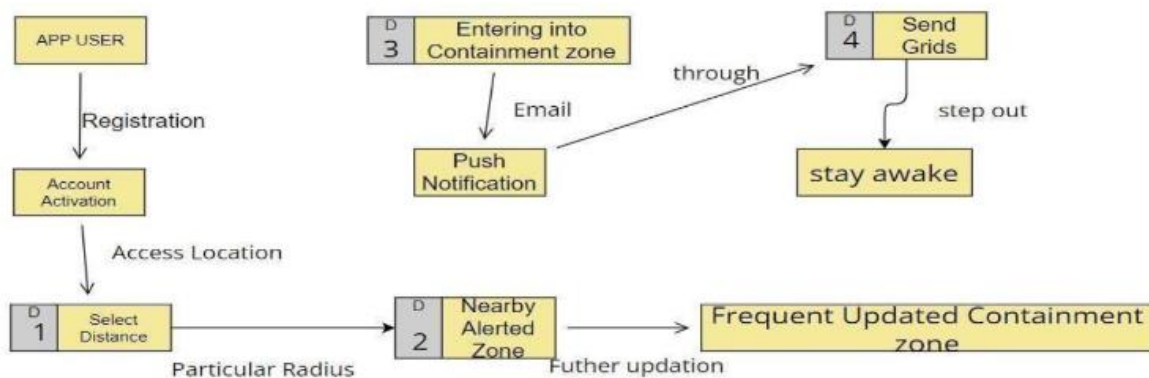
NR No.	Non-Functional Requirement	Description
NFR-1	Usability	Providing recommendation link by using customer preference .
NFR-2	Security	The software team will issue some strong security co for the user's.
NFR-3	Reliability	The database update process must rollback all related updates when any update fails.
NFR-4	Performance	The loading speed of the server is quick and fast.

CHAPTER 5

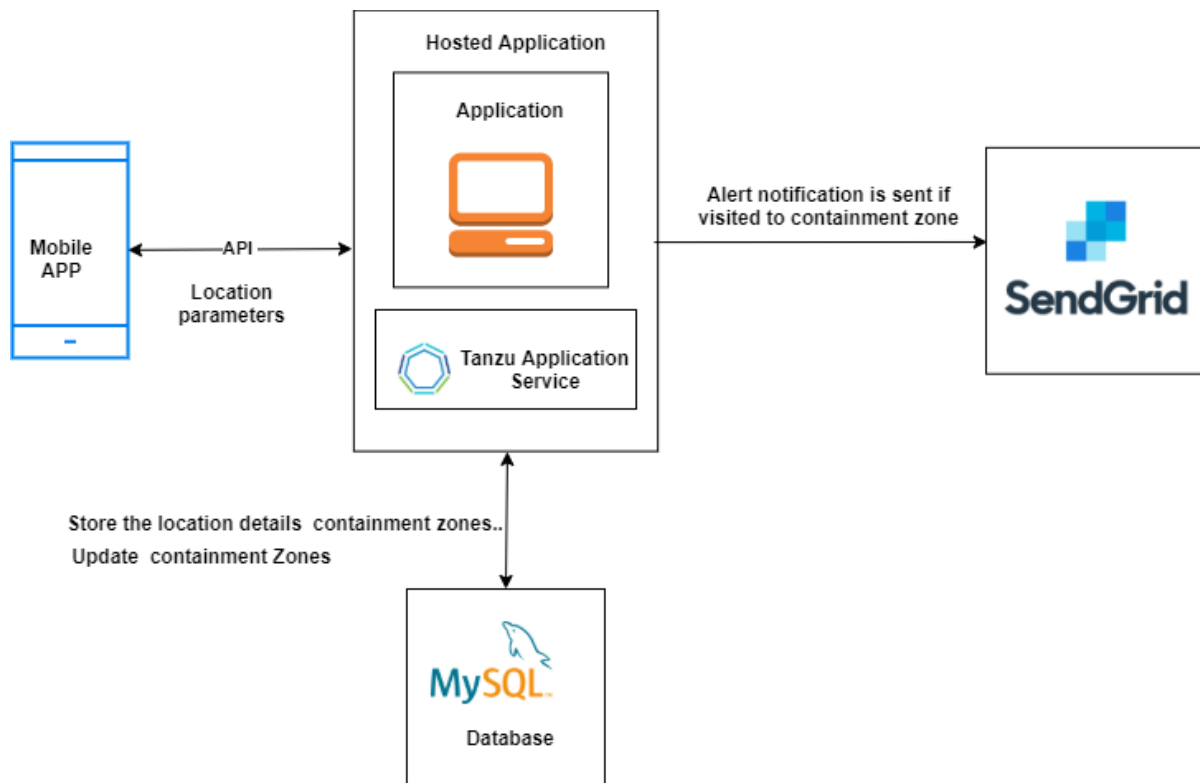
PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Eple)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN - 6	As a User , Can I manually plot the alerted zone for my convenience only.	It can be viewed in the user dashboard	Low	Sprint - 2
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	User account activities can be viewed in dashboard.	High	Sprint - 2
			Confirmation code has been send through the registered mail id ,phone number or any other acnts.			
	Location Access	USN - 2	As a User , I can viewed into the page , if there is any condition to access the location	Location can be turned through Control center	High	Sprint - 2
	Contaminated Zones	USN - 3	Is it accurately show off the alerted zone If I entered into the zone the messages are properly received through email.	Alerted messages are send by sendgrids through the registered mail id	High	Sprint - 3
Administrator	Frequent Updates	USN - 4	Admin are necessary to updates the recent containment through their portals and these seen throught the app.	It can be accessed by Geo fencing.	Medium	Sprint - 4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprints are the backbone of any good Agile development team. And the better prepared you are before a sprint, the more likely you are to hit your goals. Sprint planning helps to refocus attention, minimize surprises, and (hopefully) guarantee better code gets shipped.

But maybe more than that, sprint planning aligns the development team with the product owner. Like any relationship, the one between you and your team requires communication and clarity. And taking the time to sit down and make sure that your expectations are understood and *can* be done by your team is key to keeping everyone motivated and productive.

6.2 Sprint Delivery Schedule

The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life.

On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful.

If you're looking to schedule your next sprint, you've come to the right place. Keep reading to learn everything you need to know about sprint scheduling, including some tips to drive the best results.

6.3 Reports from JIRA

Jira applications can generate **email notifications** for various events that happen throughout the lifecycle of an issue, including custom events. Notifications are defined within a notification scheme (see below), which associates particular events with particular email recipients. The notification scheme is then assigned to a particular project.

CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1

1. Sign-up verification by phone number. User registration contains name, age, home address, work address (optional).
2. User locations are tracked by explicit user choice and stored in local storage only. During tracking user is
3. User uploads locally stored tracked locations and home address after testing positive for COVID-19. These locations are stored in cloud storage. Note that, no other user information except the locations are accessible by other users. Nevertheless, in case of false information administrative measures can be taken to track down the culprit (not implemented yet).

7.2 Feature 2

1. The App queries the database storage of infected locations by locally stored locations and notifies the user immediately if a match is found. This background process runs once every hour.
2. Users can view and post queries or relief requests through a global news feed, similar to other social media platforms.
3. Relief request posts are anonymous and provide only a post author contact number, which was used for verification. Other users willing to help can contact him through the app using the phone number.

7.3 Database Schema

You can configure IBM® Data Server Manager (DSM) to generate alerts when specific performance issues arise on your connected Db2® database.

You can select, and in some instances configure, these alerts, from the monitoring profile of a connected database. Areas that you can apply performance alerts to with DSM fall into two categories:

- excessive lock wait times
- poorly performing statements

You activate the use of alerts by going to **Settings > Monitoring Profiles**, selecting a profile for editing (or creating a new profile), and then selecting the **Alert Monitoring** option.

By expanding the **Performance** list on the *Alert Monitoring* page, you can access all of the available alerts.

If applicable, you can configure the properties of specific alerts by rolling over the associated cell under the *Action* column and clicking the **Configure** link that appears. A pop-up box will open, allowing you to change the default values to those that better suit your requirements.

CHAPTER 8

TESTING

8.1 Test Cases

Infrastructure-as-a-Service (IaaS) delivers fundamental compute, network and storage resources to consumers on-demand, over the Internet and on a pay-as-you-go basis. Using cloud infrastructure on a pay-per-use scheme enables companies to save on the costs of acquiring, managing and maintaining their own IT infrastructure. Plus, the cloud is easily accessible. Most major cloud service providers — including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure — offer IaaS with their cloud computing services.

Platform-as-a-Service (PaaS) provides customers a complete cloud platform — hardware, software and infrastructure — for developing, running and managing applications without the cost, complexity and inflexibility of building and maintaining that platform on-premises. Organizations may turn to PaaS for the same reasons they look to IaaS; they want to increase the speed of development on a ready-to-use platform and deploy applications with a predictable and cost-effective pricing model.

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

CHAPTER 9

RESULTS

9.1 Performance Metrics

Tests have been carried out in various containment zones across West Bengal for the validation of the

Android application. The identified containment zones chosen for the testing of the application were visited one by one. various containment zones identified for conducting the test, the date, time of entry, time of receiving the notification alerts upon entering. From it is highlighted that the application sends notification alerts within 5–8 seconds on entering.

CHAPTER 10

ADVANTAGES & DISADVANTAGES

(1) User Privacy Protection

Location tracking is enabled by the user and is informed to the user via a fixed notification. Before user tests positive for COVID-19 and uploads all his/her locations, the locations are stored in the device's local storage, none but the user has access to it. Once user tests positive for COVID-19 and uploads his/her locations, the identity of the user is preserved and not accessible to any other user. However, administrative access is enabled for tracking down false claims (not implemented yet) for taking legal actions.

(2) Efficient Access to potential Huge Server Data Storage

Tracked location data of COVID-19 positive patients will evidently get very large, as the number of affected people is rising each day. Moreover, in many areas people are still reluctant or don't have the luxury to maintain social distancing. To somewhat make the query process of a possible huge data storage a hashing algorithm is implemented. A particular tracked location is converted into its corresponding square block/s of area 20 meters x 20 meters along with an hourly time frame.

The block generation is similar to hashing function by providing a key that is the particular index for a query, with the additional benefit that the block also defines a radius of presence for any particular location. A block is defined by its bottom left and top right diagonal coordinates.

(3) Anonymous Relief Posts:

Through the app's global news feed, relief requests can be posted without directly sharing personal or family information of a user. A contact button is attached to relief posts through which any other user can call and contact the relief request post's author and reach out for help. This feature especially targets the middle-class families that are suffering greatly in silence and cannot seek help publicly. A user is allowed to make only one relief post every seven days, this is a measure taken to stop misuse of the feature.

CHAPTER 11

CONCLUSION

To achieve the golden balance between lock-down as well as keeping the industries / business running together it is recommended that the Data Science techniques such as K-means can be adapted to define the micro-level demarcation of containment zones and manage them effectively. The clusters formed based on COVID-19 patient's locational data using Data Science techniques (specifically K-means) will be agile, unbiased, accurate, visible, economic and easy to apply.

CHAPTER 12

FUTURE SCOPE

The application provides an efficient way of showing the identified Covid-19 containment zones to the users in a Google map. With the alarming increase of Covid-19 affected cases throughout the world, this developed application can be employed as a tool for creating further social awareness among the people. This application further tracks the user's location and checks whether it is present in the list of identified containment zones. It sends separate notification alerts to the user on entering. The developed android application further extracts the IMEI Number of the trespasser in the containment zones which can be useful to the local police to track and identify people who are frequently trespassing the containment zones. Thereby this application identifies the containment zones and highlights the need for taking further precautionary measures for combating Covid-19.

CHAPTER 13

APPENDIX

Source code

APP.PY

```
import adminverification
from flask import Flask, render_template, request, redirect, session
app = Flask(__name__)
app.secret_key = 'venki2002'
@app.route('/')

def home_page():

    if session.get('conzo_login'):

        return render_template('home_page.html', conzo_Mail=session.get('conzo_Mail'))

    else:

        return redirect('/admin')

@app.route('/admin', methods=['POST', 'GET'])

def admin_login():

    if request.method == 'GET':

        return render_template("adminlogin.html")

    elif request.method == 'POST':

        username = request.form.get('mail')
```

```
res = adminverification.adminloginverification(username, request.form.get('password'))
```

```
if res:
```

```
    session['conzo_login'] = True
```

```
    session['conzo_Mail'] = username
```

```
    return redirect('/')
```

```
else:
```

```
    return render_template('adminlogin.html', data=res)
```

```
@app.get('/logout')
```

```
def admin_logout():
```

```
    session.pop('conzo_login', None)
```

```
    session.pop('conzo_Mail', None)
```

```
    return redirect('/')
```

```
@app.route('/admin/registration', methods=['POST', 'GET'])
```

```
def admin_register():
```

```
    if request.method == 'GET':
```

```
        return render_template('adminRegistration.html')
```

```
elif request.method == 'POST':
```

```
    res = adminverification.adminRegister(request.form.get('mail'), request.form.get('password'),  
request.form.get('reqid'))
```

```
    if res:
```

```
        return redirect('/')
```

```
    else:
```

```
        return render_template('adminRegistration.html', data=res)
```

```
if __name__ == '__main__':
```

```
app.run(debug=True)
```

```
register.html
```

```
<!DOCTYPE html>
```

```
<html lang="en"
```

```
xmlns="http://www.w3.org/1999/html"><head>
```

```
<meta charset="UTF-8">
```

```
<title>Registration</title>
```

```
<link rel="stylesheet" href="../static/adminreg.css">
```

```
<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"  
integrity="sha384-
```

```
AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p" crossorigin="anonymous"/>
```

</head>

<body>

<form id="forms" method="post">

<h1>Admin Registration</h1>

<div class="user">

<i class="fas fa-user"></i>

<input type="email" placeholder="Mail ID" name="mail"
id="mail"><small></small>

</div>

<div class="password">

<i class="fas fa-lock"></i>

<input type="password" placeholder="Password" name="password" id="password">

<small></small>

</div>

<div class="password">

<i class="fas fa-lock"></i>

<input type="text" placeholder="HOSPITAL ID" name="reqid" id="RequestID">

<small></small>

</div>

<div class="button">

<button class="button">REGISTER</button>

<button>LOGIN</button>

</div>

<h4 class="text-danger" id="pass-warning" style="color: white"></h4></form>

<script src="../static/js/adminreg.js"></script>

</body>

</html>

Home.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8"/>

```
<title>ConZon</title>
```

```
<link rel="stylesheet" href="../static/home_page.css"/>
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/5.15.4/css/all.min.css"/><link rel="icon" href="../static/picture/conzo-logo.png"  
type="image/icon type">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<!-- navbar -->
```

```
<nav>
```

```
<ul>
```

```
<li><a href="#" class="logo">
```

```
<imgsrc="../static/picture/admin%20logo.png"  
alt="adminlogo"><span class="nav-item"  
id="userName"></span>
```

```
</a></li>
```

```
<li><a onclick="show('dashboard')">
```

```
<i class="fas fa-menorah"></i>
```

```
<span class="nav-item">Dashboard</span>
```

```
</a></li>
```

```
<li><a onclick="show('update')">
```

```
<i class="fas fa-database"></i>
```

```
<span class="nav-item">Update</span>
```

```
</a></li>
```

```
<li><a href="/logout" class="logout">
```

```
<i class="fas fa-sign-out-alt"></i>
```

```
<span class="nav-item">Log out</span>
```

```
</a></li>
```

```
</ul>
```

```
</nav>
```

```
<!-- dashboard -->
```

```
<section class="main" id="dashboard">
```

```
<div class="main-top">
```

```
<h1>Current Status</h1>
```

```
</div>
```

```
<div class="users">
```

```
<div class="card">
```

```
<h4>Current patient status</h4>
```

```
<div class="status"><h1></h1></div>
```

```
</div>
```

```
<div class="card">
```

<h4>Current Zone status</h4>

<div class="status" id="sta2"><h1></h1></div>

</div>

</div>

<section class="patent">

<div class="patent-list">

<h1>Recently Added Containment Zone</h1>

<table class="table" id="Current_details">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>City</th>

<th>Latitude</th>

<th>Longitude</th>

<th>Address</th>

</tr>

</thead>

<tbody id="show_details">

</tbody>

</table>

</div>

</section>

</section>

<!-- update detail -->

<section class="main" id="update">

<div class="main-top">

<h1>Updating Data</h1>

</div>

<!--Inserting Data to Db-->

<form action="/display_data_add" method="post"
class="admin_form"><section class="patent">

<h1>Insert Data To Add Containment Zone </h1>

<table class="table" id="admin_table">

<thead>

<tr>

<th>S.No</th>

<th>Name</th>

<th>City</th>

<th>Latitude</th>

<th>Longitude</th>

<th>Mail_id</th>

<th>Button</th>

</tr>

</thead>

<tbody id="add_input">

</tbody>

</table>

</section>

<button type="button" onclick="assign_name('#add_input','#admin_table');onAddWebsite();" id="add_button">ADD

ROW

</button>

<button type="submit"
class="submit_btn">Submit</button></form>

<!--Deleting Data from Db-->

<form action="/delete_data" method="post"
class="admin_form_delete"><section class="patent">

<h1>Delete Data From Containment Zone </h1>

<table class="table" id="admin_table_delete">

<thead>

<tr>

<th>S.No</th>

<th>Name</th>

<th>City</th>

```

        <th>Latitude</th>

        <th>Longitude</th>

        <th>Mail_id</th>

        <th>Button</th>

    </tr>

</thead>

<tbody id="delete_input">

</tbody>

</table>

</section>

<button type="button" onclick="assign_name('#delete_input','#admin_table_delete');onAddWebsite()"
    id="add_button">ADD ROW

</button>

<button type="submit"
class="submit_btn">Submit</button></form>

</section>

</div>

</body>

<script>
document.getElementById("userName").innerHTML = '{{ConZon_user}}';
document.querySelector('.status h1').innerHTML = '{{ count_data }}';
document.querySelector('#sta2 h1').innerHTML = '{{ count_data }}';

</script>

```

```
<script src="../../static/js/home_page.js">
```

```
</script>
```

```
</html>
```

Mail.py

```
from flask_mail import Mail, Message
```

```
from app import ConZon
```

```
app = ConZon
```

```
mails = Mail(app)
```

```
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
```

```
app.config['MAIL_PORT'] = 465
```

```
app.config['MAIL_USERNAME'] = '20euit511@skcet.ac.in'
```

```
app.config['MAIL_PASSWORD'] = 'venki2002'
```

```
app.config['MAIL_USE_TLS'] = False
```

```
app.config['MAIL_USE_SSL'] = True
```

```
mails = Mail(app)
```

```
def index(message, bmessage, remail):
```

```
msg = Message(message,  
  
sender='20euit511@skcet.ac.in',  
  
recipients=[remai]  
  
)  
  
msg.body = bmessage  
  
mails.send(msg)  
  
return 'sent'
```

GithubLink :

<https://github.com/IBM-EPBL/IBM-Project-44126-1660722379>

Video Demo Link :

<https://youtu.be/2HLb-17fhL0>