CONTAINMENT ZONE ALERTING APPLICATION

PROJECT BASED LEARNING

Submitted by

TEAM ID: PNT2022TMID11463

ARAVINTH P	**(910619104006)**
BALAMANIKANDAN R	**(910619104010)**
JAGATHGURU G	**(910619104030)**
JANARTHANAN	**(910619104031)**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

K.L.N. COLLEGE OF ENGINEERING, POTTAPALAYAM

(An Autonomous Institution, Affiliated to Anna University, Chennai)

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certificated that this project report **“Containment Zone Alerting Application”** is the bonafide work of **“ARAVINTH P (910619104006)”, “BALAMANIKANDAN R (910619104010)”, “JAGATHGURU G (910619104030)”** and **“JANARTHANAN K (910619104031)”** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MIRUNA JOE AMALI	Mr.D.Prabhu
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
Computer science and engineering,	Computer science and engineering,
K.L.N. College of Engineering,	K.L.N. College of Engineering,
Pottapalayam,	Pottapalayam,
Sivagangai-630 612.	Sivangangai-630-612.

Submitted for the project viva-voce conducted on __________

EVALUATOR	MENTOR

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.

The 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 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.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
API	Application Programming Interface
DB	DataBase

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Since the outbreak of Covid-19 pandemic, lots of preventive measures have been automated by the technical Industry. One of the important measures is to isolate Covid prone zones from normal zones. For this, we seek technical support to create an alert system which makes the user aware of the containment zones by facilitation the integration of the geolocation of the containment areas with a mobile app through cloud so that real time alerting could be made possible. For the updation and storage purposes for the respective zones we use cloud support to make the application complete.

1.2 Purpose

The purpose of the application is to monitor the locations of user continuously and provide alert while trespassing a containment zone. Furthermore, the application is intended to provide information about Covid-containment zones in a particular area. The prime objective of the project is to build an application that provides information about the containment zones of a particular region in order to make the people aware of containment zones in real time.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

Without widespread public awareness and proactive measures taken by the populace, it is very challenging to stop the community transmission even during a lockdown in a densely populated nation like India. There were several containment zones spread out around the nation, and they were classified into red, orange, and green zones, accordingly. The red zones represent infection hotspots, the orange zones represent some infection, and the green zones represent an infection-free area. It is not a really easy task to inform every citizen of the country of every Covid Containment Zone. Even if it is done, the statistics and the zone details would change every now and then. Also, people cannot remember all the containment zones announced in the news or radio. A solution has to be brought not only to keep the people informed of Containment Zone, but also to assist them in staying away from the zones. When a person travels to a place, it is highly unlikely that he or she knows if it is a containment zone or not. There arises a confusion if the person should enter the zone or not.

2.2 References

[1] A Survey of COVID-19 Contact Tracing Apps, July 2020.

[2] COVID-19 LIFEGUARD: A Compact Wearable-IOT(W-IOT) System for Health Safety and Protection of outgoers in the Post-Lockdown World, September 2022.

[3] J. Li and X. Guo. “COVID-19 contact-tracing apps: A survey on the global deployment and challenges,” 2022, arxiv:2005.03599.

[4] T. Kuchler, D. Russel, and J. Stroebel, “JUE insight: The geographic spread of COVID-19 correlates with the structure of social networks as measured by Facebook,” J. Urban Econ., vol. 127, Jan 2022, Art. No. 103314

[5] COVIDTRAK: A Vision on social Intelligence-Empowered COVID-19 Contact Tracing.

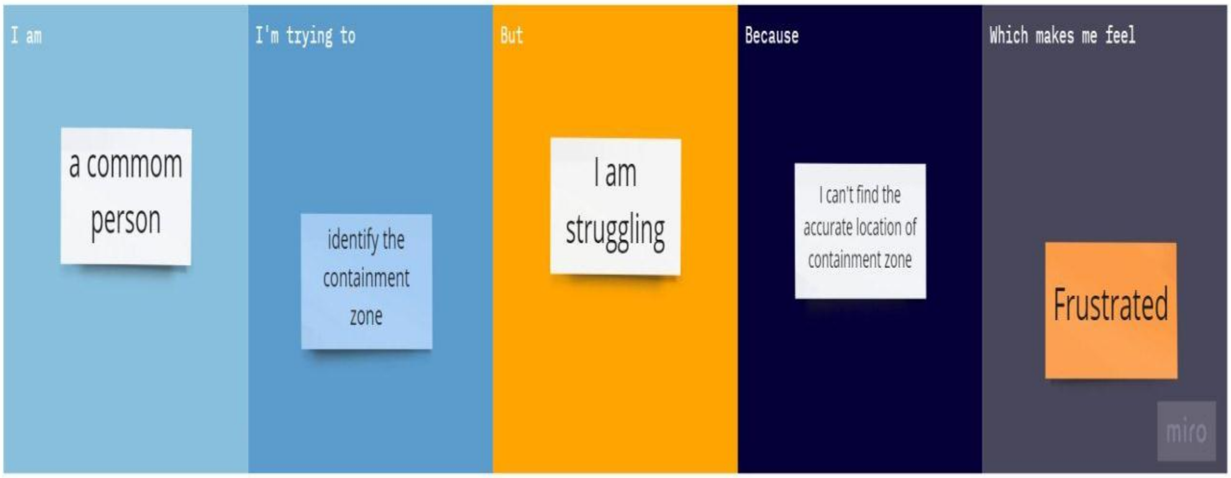
[6] Development of an android application for viewing Covid-19 Containment Zones and monitoring violators who are trespassing into it using Firebase and Geofencing.

[7]Aarogya Setu[2020] htttps://play.google.com/store/apps/details?id=nic.goi.aarogyasetu&hl=en

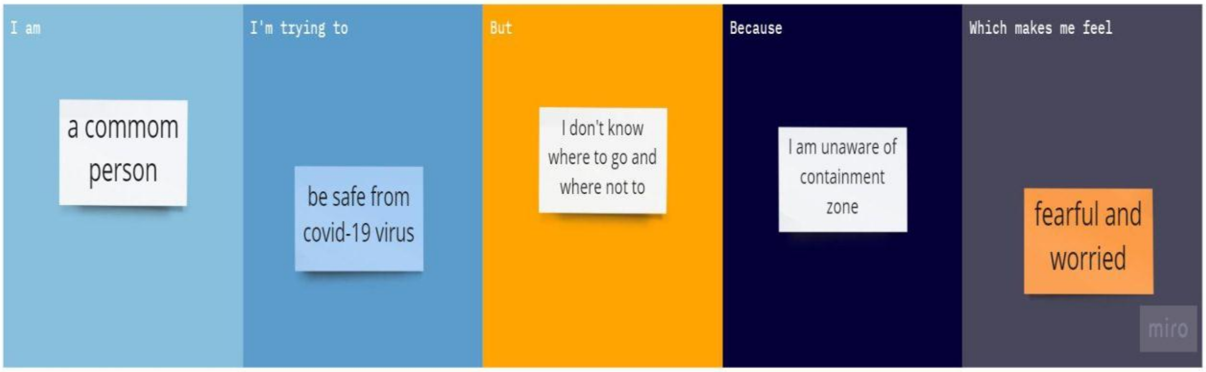
[8] Bahl P, Doolan C, de Silva C, Chugthai AA, Bourouiba L, MacIN-tyre cr[2020] Airborne or droplet precautions for health workers treating Coronavirus disease 2019? J Infect dis. https://doi.org/10.1093/infdis/jiaa189.

2.3 Problem Statement Definition

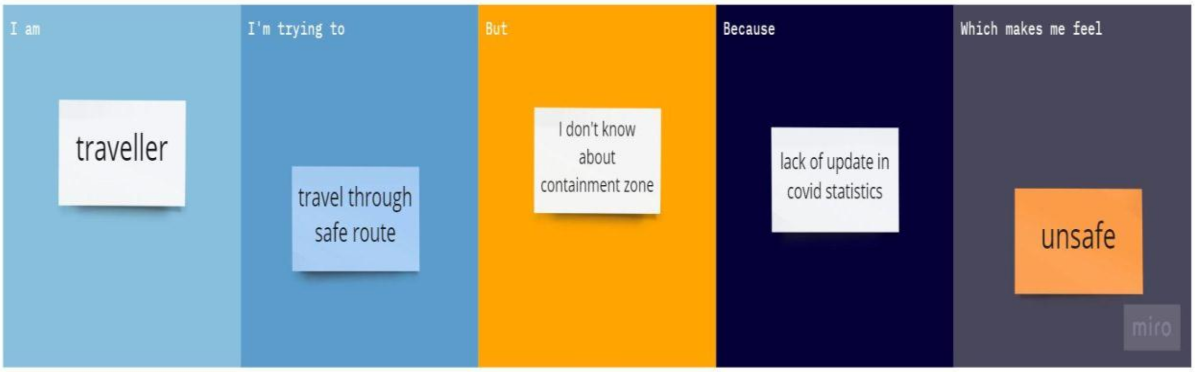
People travel to different places unaware of the fact that it is a COVID-19 containment zone and hence don't take necessary precautions. As a result, the people getting inside a containment zone are at a higher risk of getting affected by the disease.



I am	I'm trying to	But	Because	Which makes me feel
a commom person	identify the containment zone	I am struggling	I can't find the accurate location of containment zone	Frustrated



I am	I'm trying to	But	Because	Which makes me feel
a commom person	be safe from covid-19 virus	I don't know where to go and where not to	I am unaware of containment zone	fearful and worried



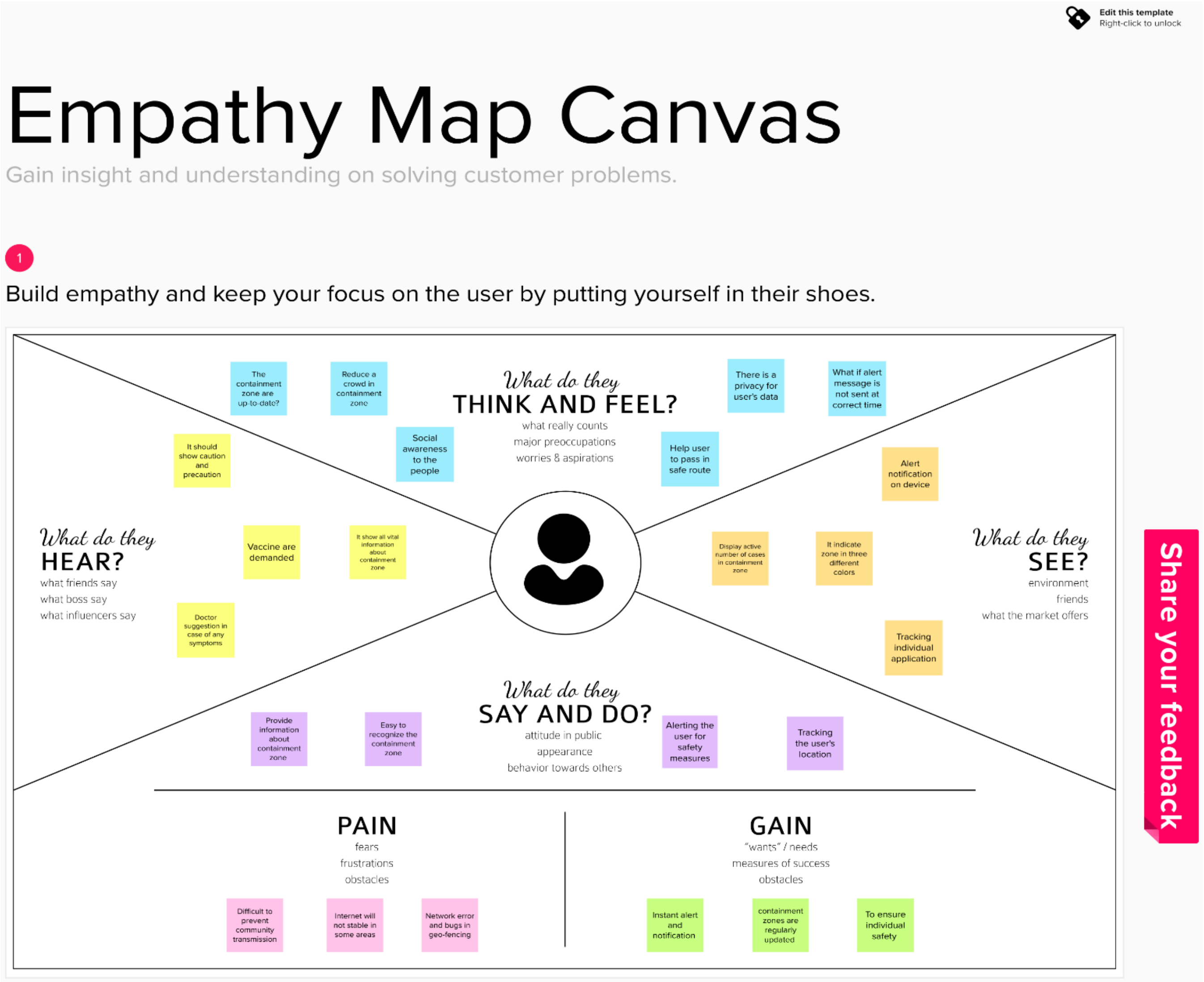
I am	I'm trying to	But	Because	Which makes me feel
traveller	travel through safe route	I don't know about containment zone	lack of update in covid statistics	unsafe

2.1 Problem Statement

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- The containment zone are up-to-date?
- Reduce a crowd in containment zone
- Social awareness to the people
- There is a privacy for user's data
- What if alert message is not sent at correct time
- Help user to pass in safe route

What do they
HEAR?
what friends say
what boss say
what influencers say

- It should show caution and precaution
- Vaccine are demanded
- It show all vital information about containment zone
- Doctor suggestion in case of any symptoms

What do they
SEE?
environment
friends
what the market offers

- Alert notification on device
- Display active number of cases in containment zone
- It indicate zone in three different colors
- Tracking individual application

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

- Provide information about containment zone
- Easy to recognize the containment zone
- Alerting the user for safety measures
- Tracking the user's location

PAIN
fears
frustrations
obstacles

- Difficult to prevent community transmission
- Internet will not stable in some areas
- Network error and bugs in geo-fencing

GAIN
"wants" / needs
measures of success
obstacles

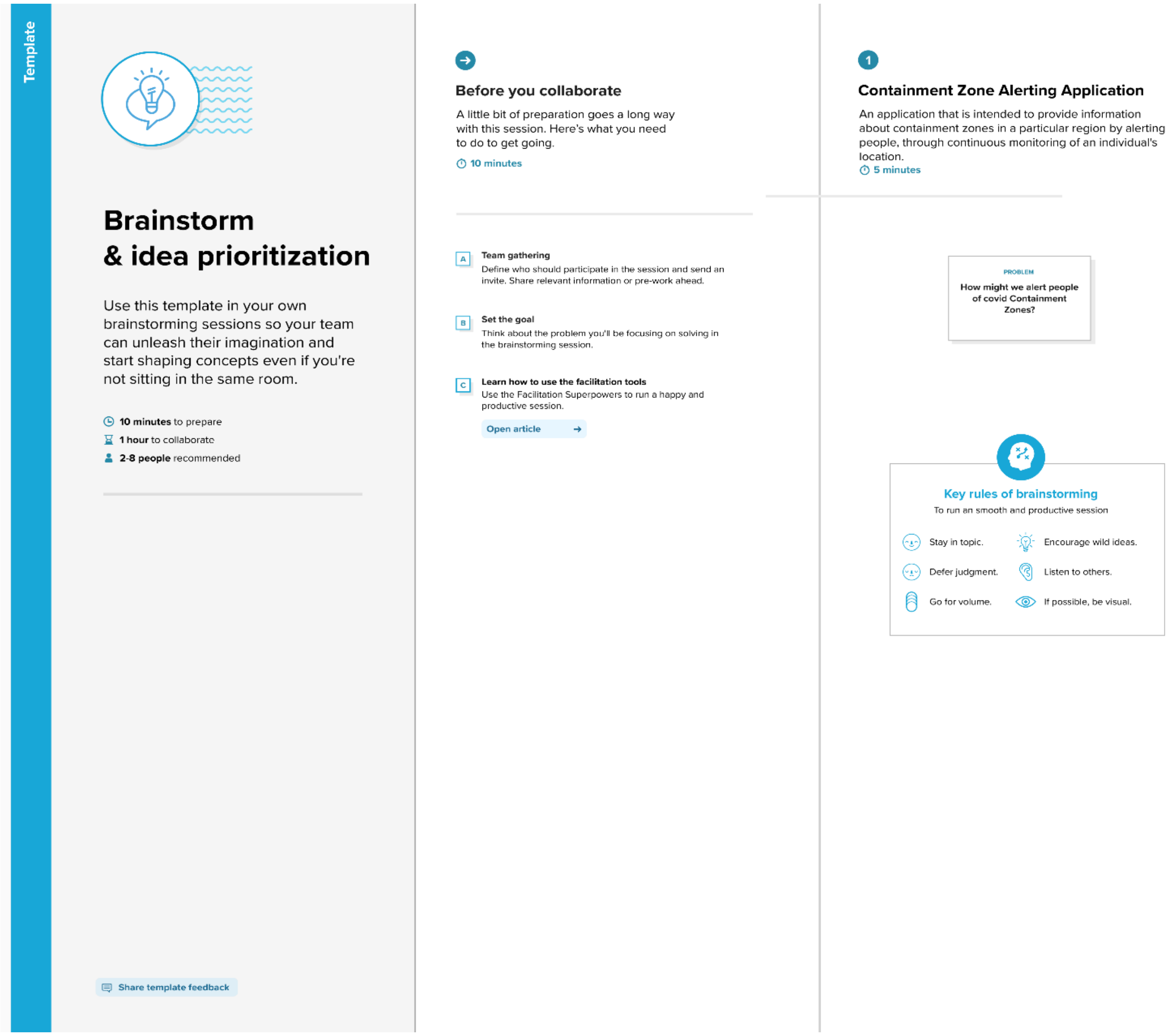
- Instant alert and notification
- containment zones are regularly updated
- To ensure individual safety

Share your feedback

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

Step 1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Containment Zone Alerting Application

An application that is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individual's location.

5 minutes

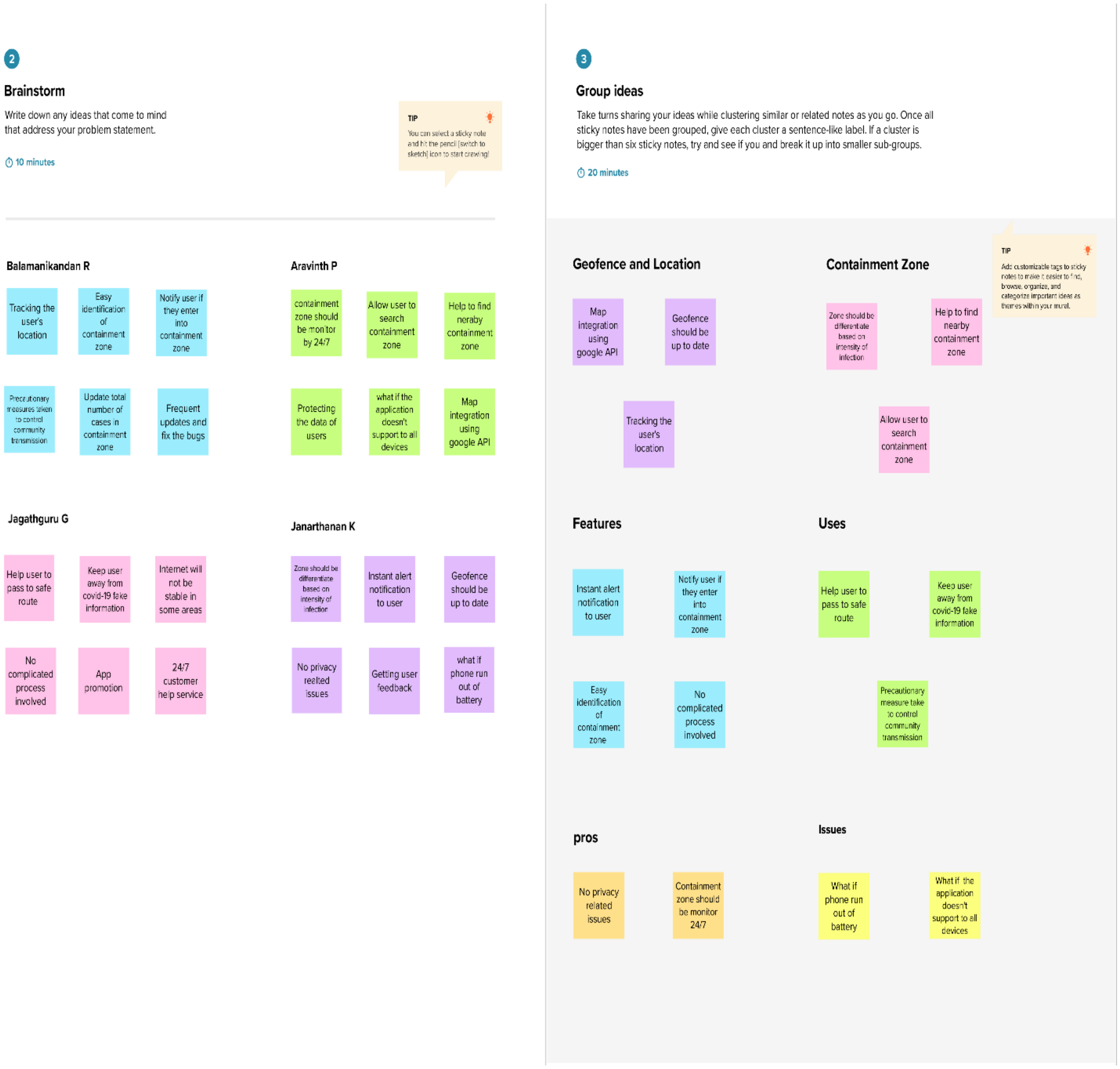
PROBLEM
How might we alert people of covid Containment Zones?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

3.2 Team Gathering, Collaboration and Select the Problem Statement

Step 2: Brainstrom, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Balamanikandan R

Tracking the user's location

Easy identification of containment zone

Notify user if they enter into containment zone

Precautionary measures taken to control community transmission

Update total number of cases in containment zone

Frequent updates and fix the bugs

Aravinth P

containment zone should be monitor by 24/7

Allow user to search containment zone

Help to find neraby containment zone

Protecting the data of users

what if the application doesn't support to all devices

Map integration using google API

Jagathguru G

Help user to pass to safe route

Keep user away from covid-19 fake information

Internet will not be stable in some areas

No complicated process involved

App promotion

24/7 customer help service

Janarthanan K

Zone should be differentiate based on intensity of infection

Instant alert notification to user

Geofence should be up to date

No privacy realted issues

Getting user feedback

what if phone run out of battery

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Geofence and Location

Map integration using google API

Geofence should be up to date

Tracking the user's location

Containment Zone

Zone should be differentiate based on intensity of infection

Help to find nearby containment zone

Allow user to search containment zone

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Features

Instant alert notification to user

Notify user if they enter into containment zone

Easy identification of containment zone

No complicated process involved

Uses

Help user to pass to safe route

Keep user away from covid-19 fake information

Precautionary measure take to control community transmission

pros

No privacy related issues

Containment zone should be monitor 24/7

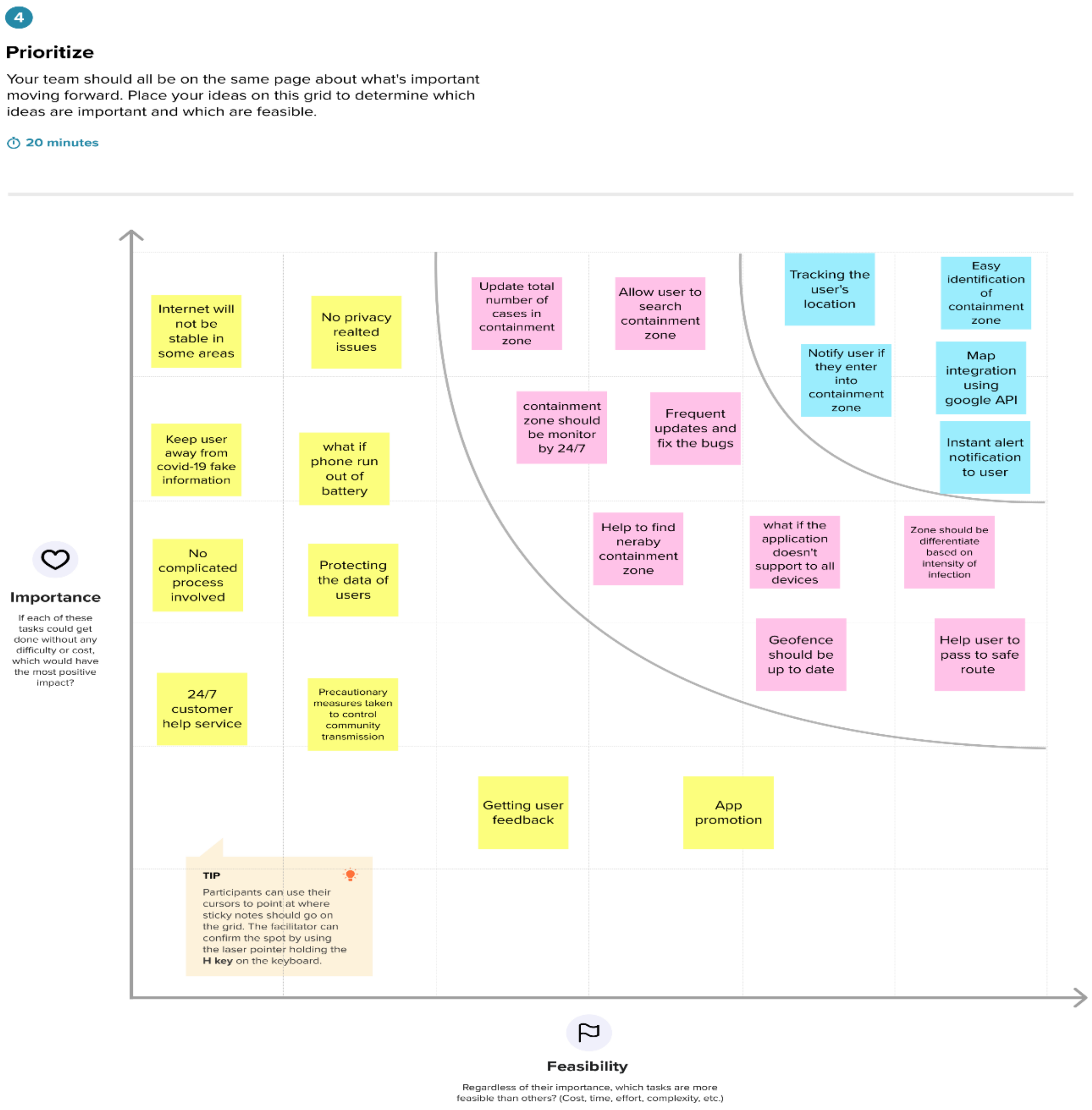
Issues

What if phone run out of battery

What if the application doesn't support to all devices

3.3 Brainstrom, Idea Listing and Grouping

Step 3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Internet will not be stable in some areas

No privacy realted issues

Update total number of cases in containment zone

Allow user to search containment zone

Tracking the user's location

Easy identification of containment zone

Notify user if they enter into containment zone

Map integration using google API

Keep user away from covid-19 fake information

what if phone run out of battery

containment zone should be monitor by 24/7

Frequent updates and fix the bugs

Instant alert notification to user

No complicated process involved

Protecting the data of users

Help to find neraby containment zone

what if the application doesn't support to all devices

Zone should be differentiate based on intensity of infection

Geofence should be up to date

Help user to pass to safe route

24/7 customer help service

Precautionary measures taken to control community transmission

Getting user feedback

App promotion

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

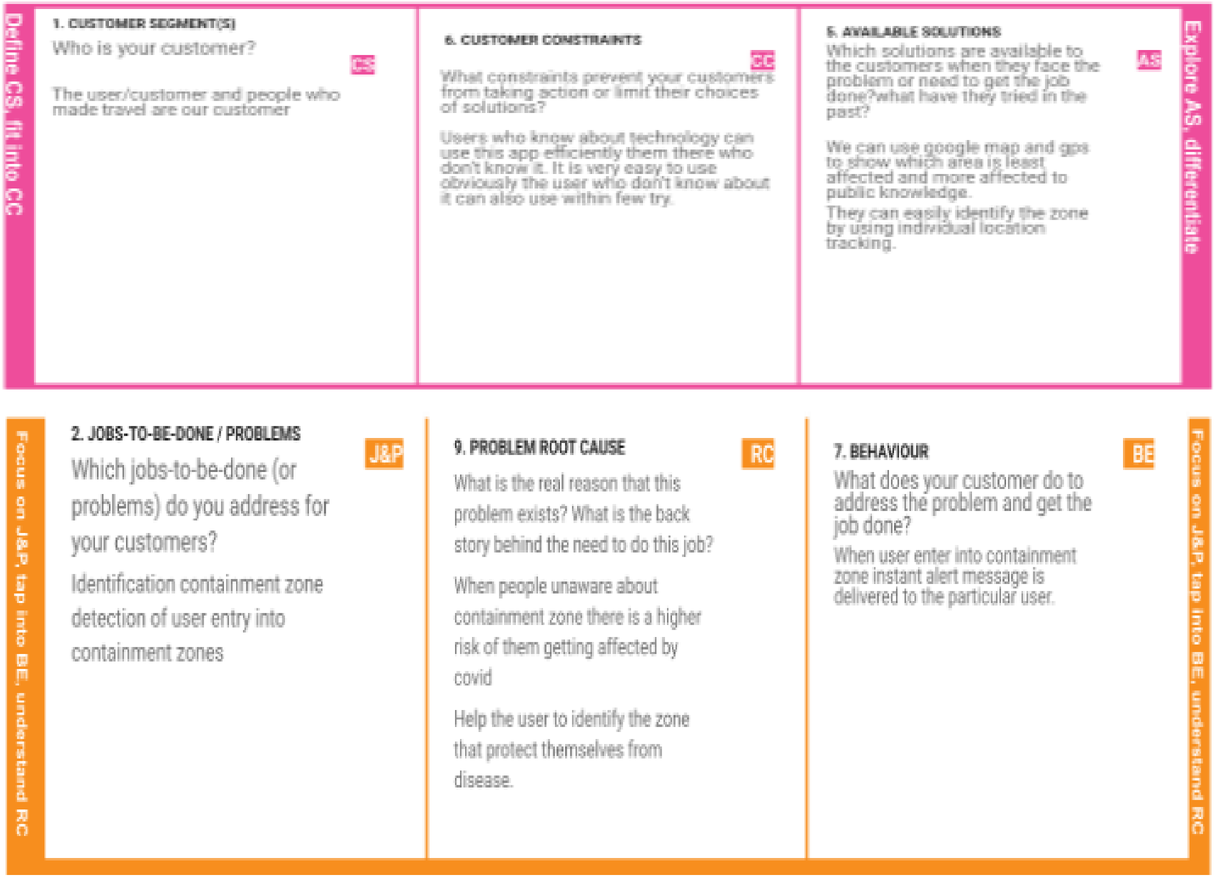
3.4 Idea Prioritization

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	The project aims at building an application that provides information about the containment zones of a particular region by continuously monitoring an individual's location. Location of the individual must be stored in the Database. Alerts are sent using the notification service.
3.	Novelty / Uniqueness	The application updates the location of the areas in google map which are identified to be containment zone. The application also notify the users if they entered into the containment zone and upload the details of the user in the database.
4.	Social Impact / Customer Satisfaction	The application will alert the people on containment zones. It protects the user’s data and updates the total number of cases.
5.	Business Model (Revenue Model)	Tie up with people for normal and premium plans. Introducing Premium plan that monitors user health by connecting app with their smart band.
6.	Scalability of the Solution	Though the application is intended for Covid-19 as of now, it can be extended and made use of prevention and containment of any disease in the future.

		The application is also extended so that users can't trespass into restricted area.

3.4 Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

Who is your customer?

CS

The user/customer and people who made travel are our customer

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions?

Users who know about technology can use this app efficiently them there who don't know it. It is very easy to use obviously the user who don't know about it can also use within few try.

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done?what have they tried in the past?

AS

We can use google map and gps to show which area is least affected and more affected to public knowledge.

They can easily identify the zone by using individual location tracking.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers?

Identification containment zone detection of user entry into containment zones

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

When people unaware about containment zone there is a higher risk of them getting affected by covid

Help the user to identify the zone that protect themselves from disease.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?

When user enter into containment zone instant alert message is delivered to the particular user.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

What triggers customers to act?

whenever the user gets into containment zone the proposed solution will alert to the user by sending message

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Before:users might be unaware about the containment zone and worried about getting out of the house.

After:no need for the user to panic since the application would alert user if they enter into containment zone

10. YOUR SOLUTION SL

The application will be created with the live location of the user

with that we can notify if they enter into the containment zone. Display of covid statistics and precaution measure

CHANNELS of BEHAVIOUR CH

1. **ONLINE**

customer can access the updated containment zone and the user can also see the covid statistics

8.2OFFLINE

User information are stored locally .

Fig 3.5 Problem Solution fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Mobile Number Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Permission	The user needs to provide location access, in order to trace his/her location.
FR-4	Containment zones are shown	Containment zones are marked and trespassers were indicated by geofencing.
FR-5	Tracking the location	Trace the trespassers by using Google map API
FR-6	Alert message via notification	By tracking their location, a message will be sent if they enter the containment zone.

4.2 Non-Functional requirement

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	People who travel can use this application to track the containment zone and pass through a safe route.
NFR-2	**Security**	Using blockchain technology for location and data encryption to protect user's data from getting into wrong hands.
NFR-3	**Reliability**	Fake news will be avoided and proper guidance is given in the application. The user can trust the result and navigate safely.
NFR-4	**Performance**	The geofencing is updated daily and shows the day-to-day updates of containment zones.
NFR-5	**Availability**	The application uses the network to load the google maps to retrieve containment zones. It is available for a good range of network bandwidth.
NFR-6	**Scalability**	This application can be accessed from any place and monitoring user's movements in pandemic zones and alerts before they are affected.

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams

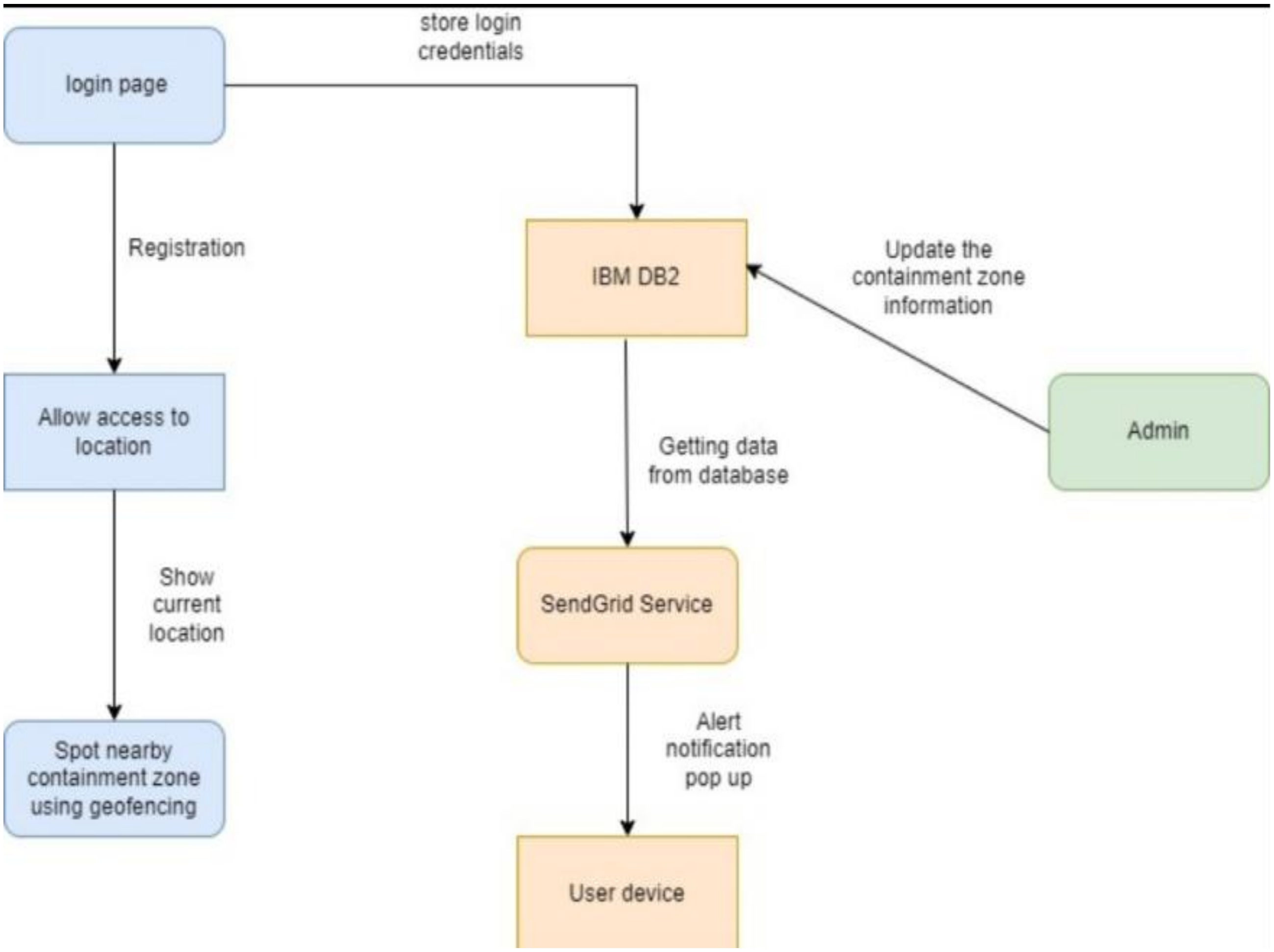

login page
store login credentials
Registration
IBM DB2
Update the containment zone information
Allow access to location
Admin
Getting data from database
Show current location
SendGrid Service
Spot nearby containment zone using geofencing
Alert notification pop up
User device

Fig 5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

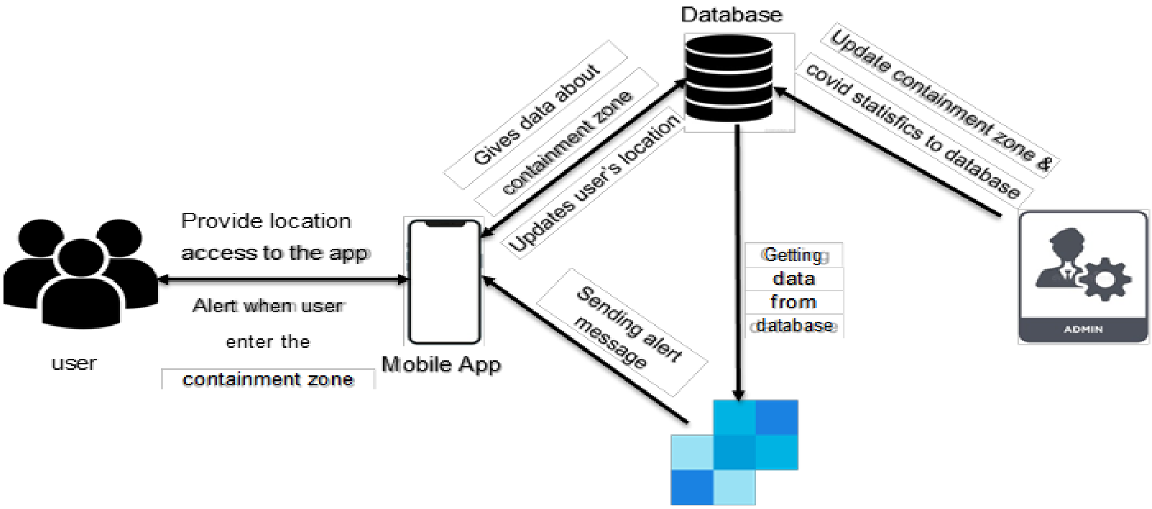

Database
Gives data about containment zone
Updates user's location
Update containment zone & covid statisfics to database
Provide location access to the app
Alert when user enter the containment zone
user
Mobile App
Getting data from database
Sending alert message
ADMIN

Fig 5.2 Solution Architecture

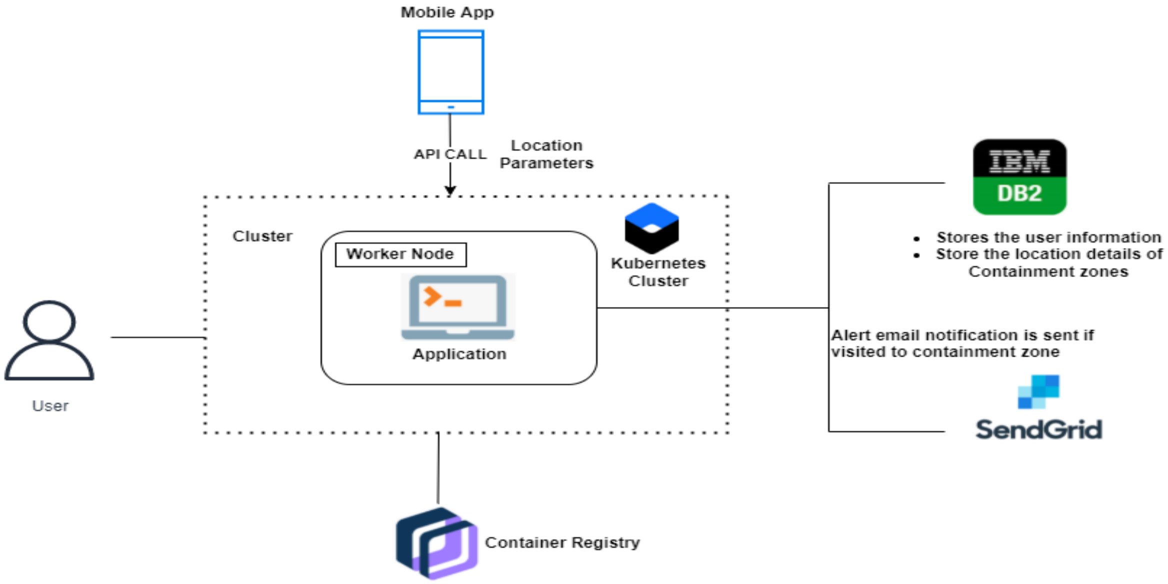
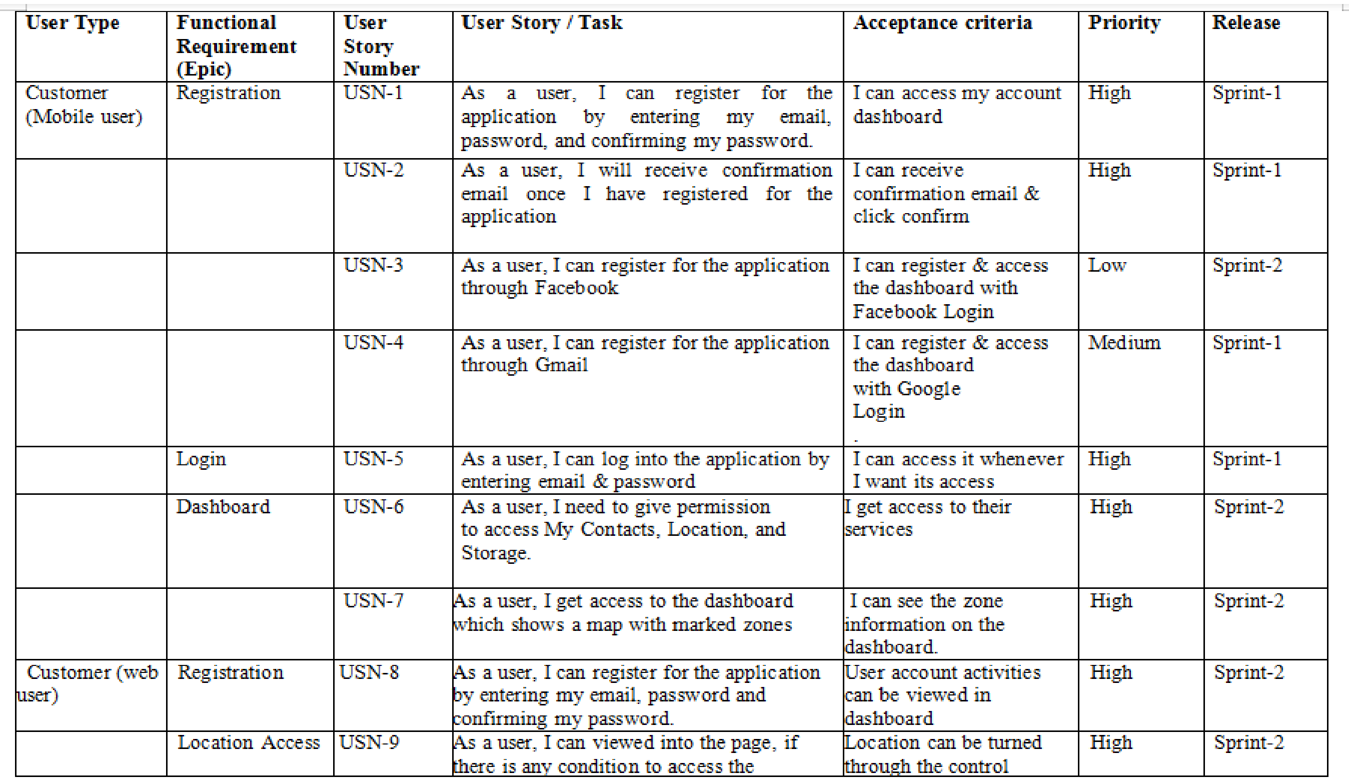
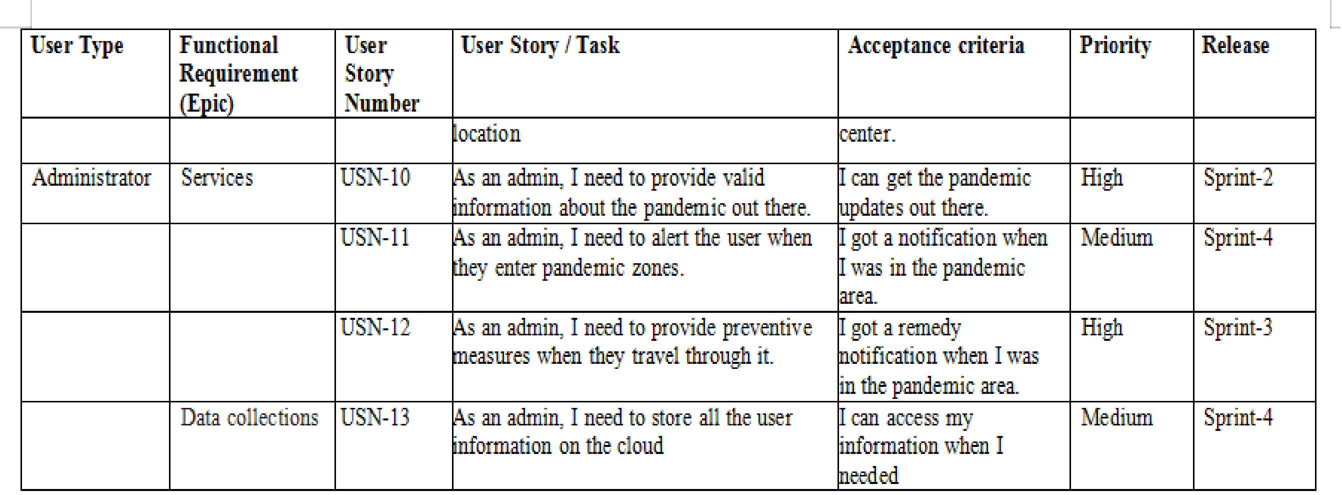

Mobile App
API CALL
Location Parameters
Cluster
Worker Node
Application
Kubernetes Cluster
User
IBM DB2
Stores the user information
Store the location details of Containment zones
Alert email notification is sent if visited to containment zone
SendGrid
Container Registry

Fig 5.3 Technical Architecture

5.3 User Stories



User Type	Functional Requirement (Epic)	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	I can register & access the dashboard with Google Login .	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access it whenever I want its access	High	Sprint-1
	Dashboard	USN-6	As a user, I need to give permission to access My Contacts, Location, and Storage.	I get access to their services	High	Sprint-2
		USN-7	As a user, I get access to the dashboard which shows a map with marked zones	I can see the zone information on the dashboard.	High	Sprint-2
Customer (web user)	Registration	USN-8	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
	Location Access	USN-9	As a user, I can viewed into the page, if there is any condition to access the	Location can be turned through the control	High	Sprint-2

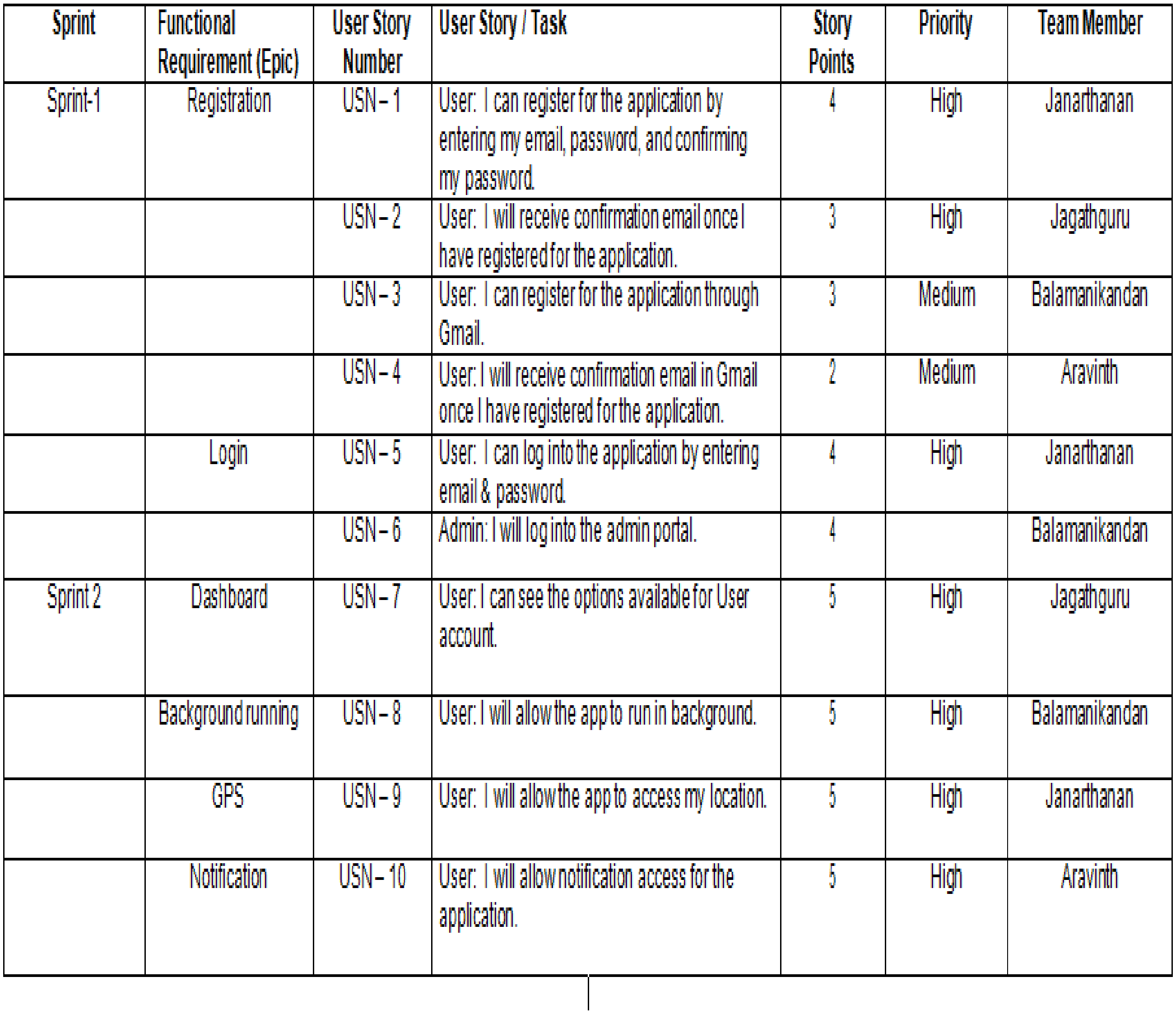


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			location	center.		
Administrator	Services	USN-10	As an admin, I need to provide valid information about the pandemic out there.	I can get the pandemic updates out there.	High	Sprint-2
		USN-11	As an admin, I need to alert the user when they enter pandemic zones.	I got a notification when I was in the pandemic area.	Medium	Sprint-4
		USN-12	As an admin, I need to provide preventive measures when they travel through it.	I got a remedy notification when I was in the pandemic area.	High	Sprint-3
	Data collections	USN-13	As an admin, I need to store all the user information on the cloud	I can access my information when I needed	Medium	Sprint-4

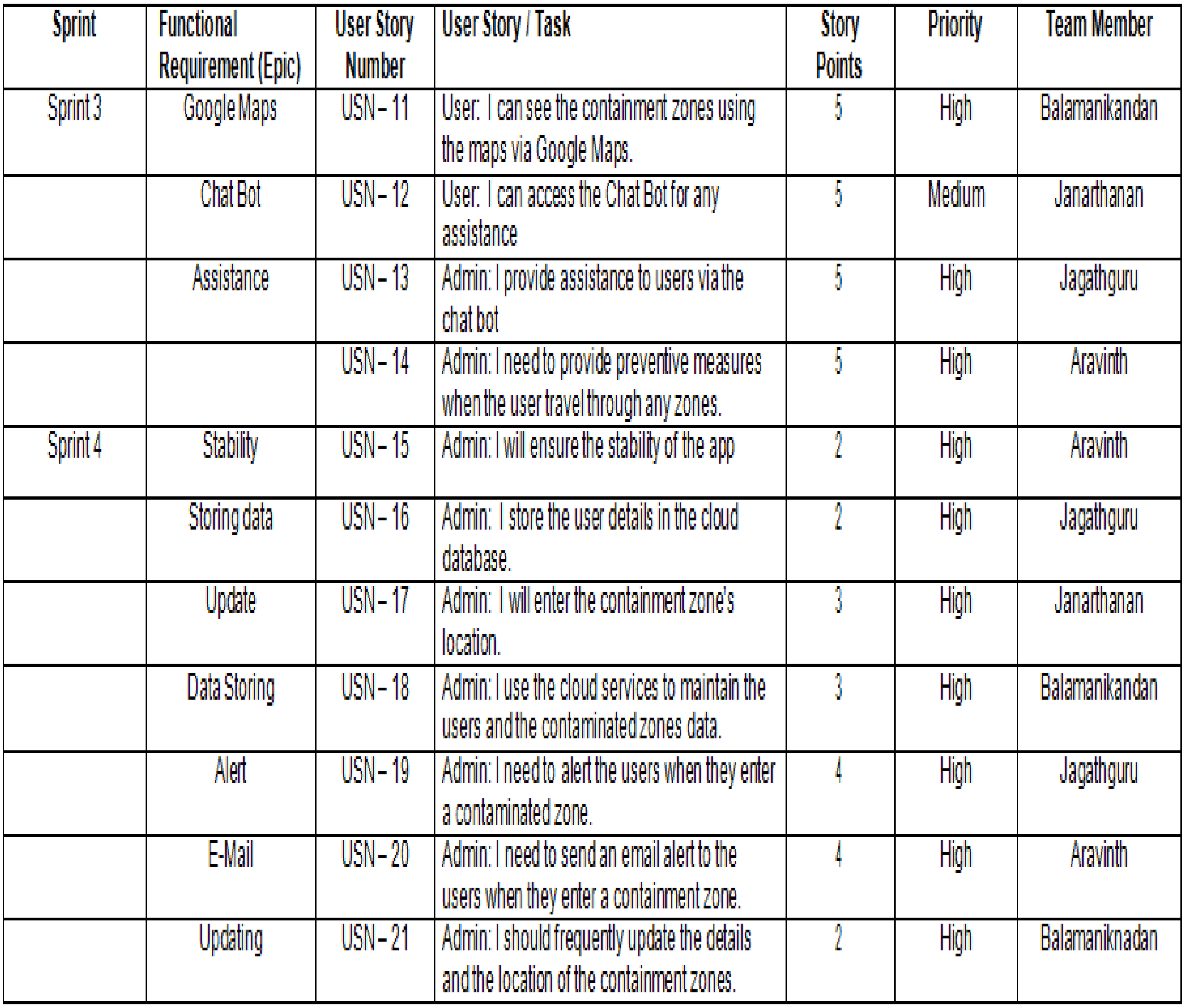
CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Registration	USN – 1	User: I can register for the application by entering my email, password, and confirming my password.	4	High	Janarthanan
		USN – 2	User: I will receive confirmation email once I have registered for the application.	3	High	Jagathguru
		USN – 3	User: I can register for the application through Gmail.	3	Medium	Balamanikandan
		USN – 4	User: I will receive confirmation email in Gmail once I have registered for the application.	2	Medium	Aravinth
	Login	USN – 5	User: I can log into the application by entering email & password.	4	High	Janarthanan
		USN – 6	Admin: I will log into the admin portal.	4		Balamanikandan
Sprint 2	Dashboard	USN – 7	User: I can see the options available for User account.	5	High	Jagathguru
	Background running	USN – 8	User: I will allow the app to run in background.	5	High	Balamanikandan
	GPS	USN – 9	User: I will allow the app to access my location.	5	High	Janarthanan
	Notification	USN – 10	User: I will allow notification access for the application.	5	High	Aravinth



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint 3	Google Maps	USN – 11	User: I can see the containment zones using the maps via Google Maps.	5	High	Balamanikandan
	Chat Bot	USN – 12	User: I can access the Chat Bot for any assistance	5	Medium	Janarthanan
	Assistance	USN – 13	Admin: I provide assistance to users via the chat bot	5	High	Jagathguru
		USN – 14	Admin: I need to provide preventive measures when the user travel through any zones.	5	High	Aravinth
Sprint 4	Stability	USN – 15	Admin: I will ensure the stability of the app	2	High	Aravinth
	Storing data	USN – 16	Admin: I store the user details in the cloud database.	2	High	Jagathguru
	Update	USN – 17	Admin: I will enter the containment zone's location.	3	High	Janarthanan
	Data Storing	USN – 18	Admin: I use the cloud services to maintain the users and the contaminated zones data.	3	High	Balamanikandan
	Alert	USN – 19	Admin: I need to alert the users when they enter a contaminated zone.	4	High	Jagathguru
	E-Mail	USN – 20	Admin: I need to send an email alert to the users when they enter a containment zone.	4	High	Aravinth
	Updating	USN – 21	Admin: I should frequently update the details and the location of the containment zones.	2	High	Balamaniknadan

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 OCT 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

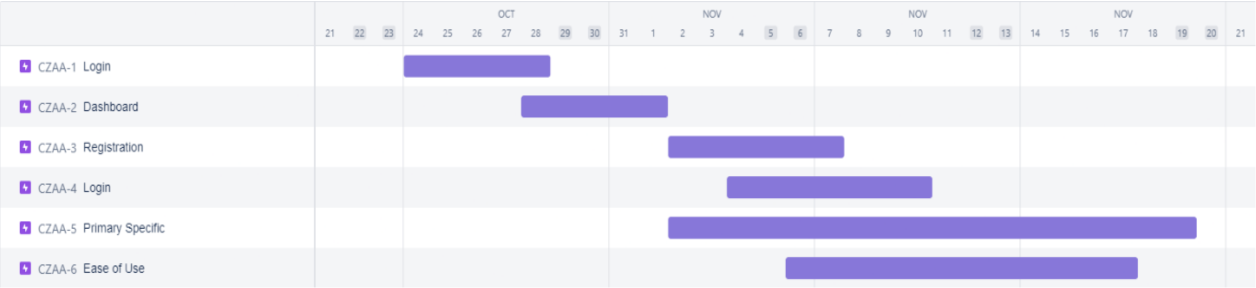
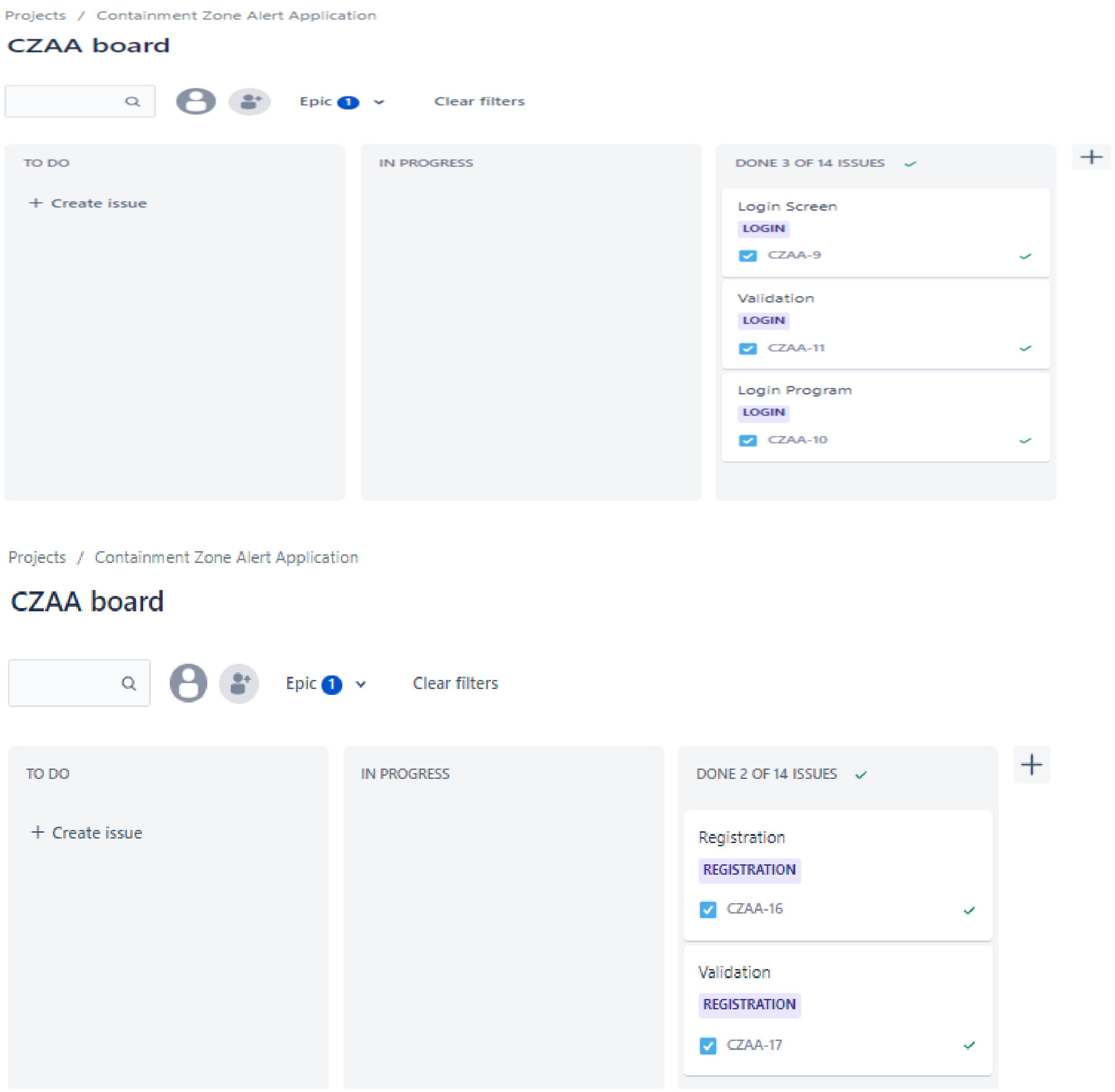


Fig 6.1 JIRA ROADMAP



Projects / Containment Zone Alert Application

CZAA board

Epic 1 Clear filters

TO DO

+ Create issue

IN PROGRESS

DONE 3 OF 14 ISSUES ✓

Login Screen
LOGIN
☑ CZAA-9 ✓

Validation
LOGIN
☑ CZAA-11 ✓

Login Program
LOGIN
☑ CZAA-10 ✓

Projects / Containment Zone Alert Application

CZAA board

Epic 1 Clear filters

TO DO

+ Create issue

IN PROGRESS

DONE 2 OF 14 ISSUES ✓

Registration
REGISTRATION
☑ CZAA-16 ✓

Validation
REGISTRATION
☑ CZAA-17 ✓

Fig 6.2 Snip from JIRA Board

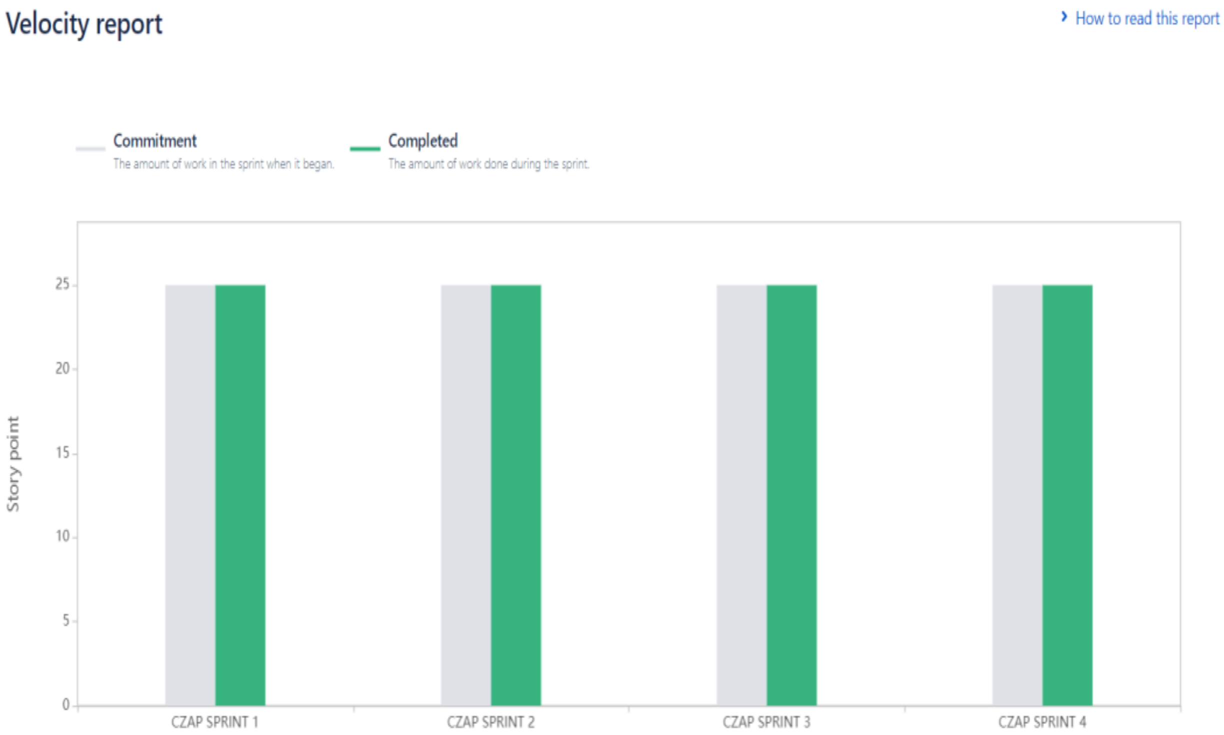


Fig 6.3 Velocity Report

CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1

7.1 LOGIN:

```
@app.route('/', methods=['POST', 'GET'])
def login():
    msg = ''
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM admincreds WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_both(stmt)
        accounts = account
        if (account):
            if (password == accounts['PASSWORD']):
                msg = 'Logged in successfully !'
                session['id'] = 1
                session['email'] = email
                session['password'] = password
                return render_template('home.html', msg=msg)
            else:
                msg = 'Wrong Credentials'
                return redirect(url_for("login"))
    return render_template('index.html', msg=msg)
```

7.2 REGISTER:

```
@app.route('/reg', methods=['POST', 'GET'])
def reg():
    msg = ''
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        sql = "SELECT * FROM admincreds WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not password or not email:
            msg = 'Please fill the Details !'
        else:
            insert_sql = "INSERT INTO admincreds VALUES (?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, email)
            ibm_db.bind_param(prep_stmt, 2, password)
            ibm_db.execute(prep_stmt)
            msg = 'Account created successfully '
            return render_template('index.html', msg=msg)
    return render_template('register.html', msg=msg)
```

7.3 HOME PAGE:

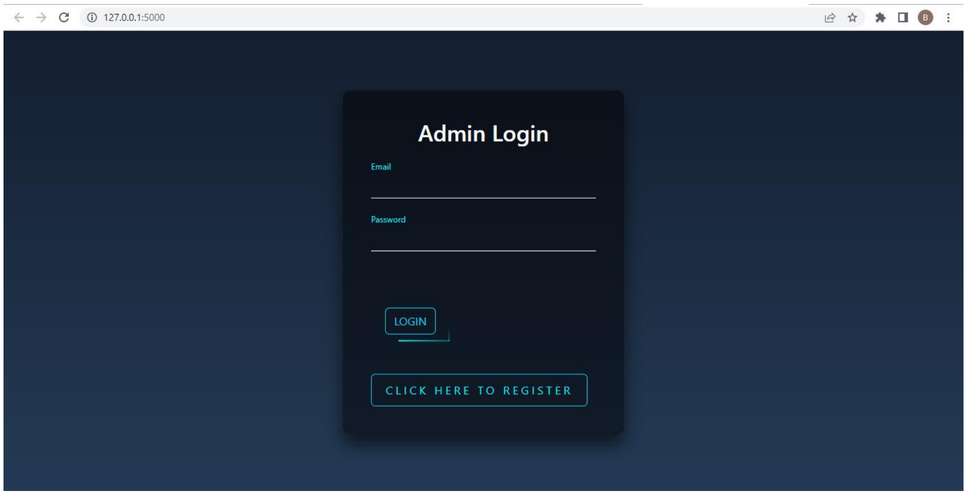
```
@app.route("/home", methods=["POST", "GET"])
def home():
    if (session['id'] == None):
        return redirect(url_for('login'))
    if (request.method == "POST"):
        lat = request.form["lat"]
        lon = request.form["lon"]
        vis = 0
        if (lat == "" or lon == ""):
            return render_template('home.html', email=session['email'], id=session['id'],
success=0)
        location_cursor = mysql.connection.cursor()
        location_cursor.execute(
            'INSERT INTO location(location_lat,location_long,location_visited)
VALUES(%s,%s,%s)', (
                lat, lon, vis
            )
        )
        mysql.connection.commit()
        location_cursor.close()
        return render_template('home.html', email=session['email'], id=session['id'],
success=True)
    return render_template('home.html', email=session['email'], id=session['id'])
```

7.4 Containment zone List Page:

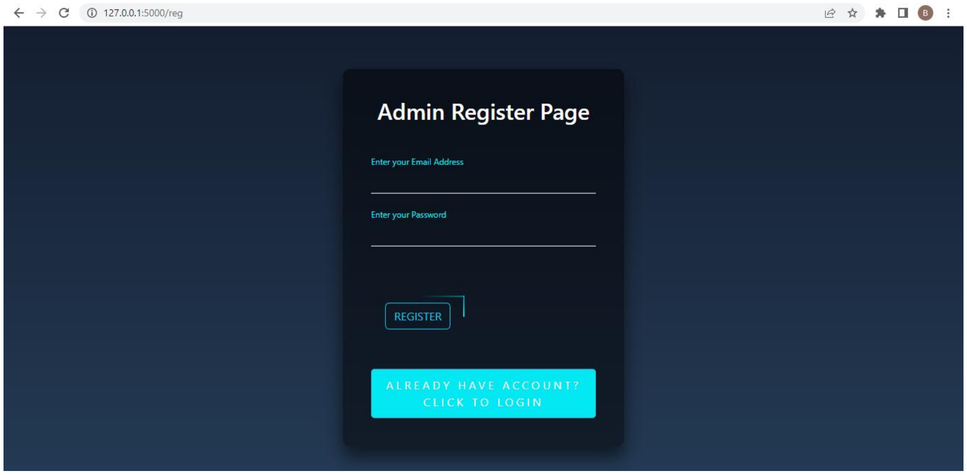
```
@app.route("/data")
def data():
    if (session['id'] == None):
        return redirect(url_for('login'))
    location_cursor = mysql.connection.cursor()
    # check whether user already exists
    user_result = location_cursor.execute(
        "SELECT * FROM location"
    )
    if (user_result == 0):
        return render_template("data.html", responses=0)
    else:
        res = location_cursor.fetchall()
        print(res)
        return render_template("data.html", responses=res)
# ------------Redirect to Home Page-------------
@app.route("/gotohome")
def gotohome():
    return redirect(url_for('home'))
@app.route("/gotoreg")
def gotoreg():
    return redirect(url_for('reg'))
```

7.5 LOGOUT:

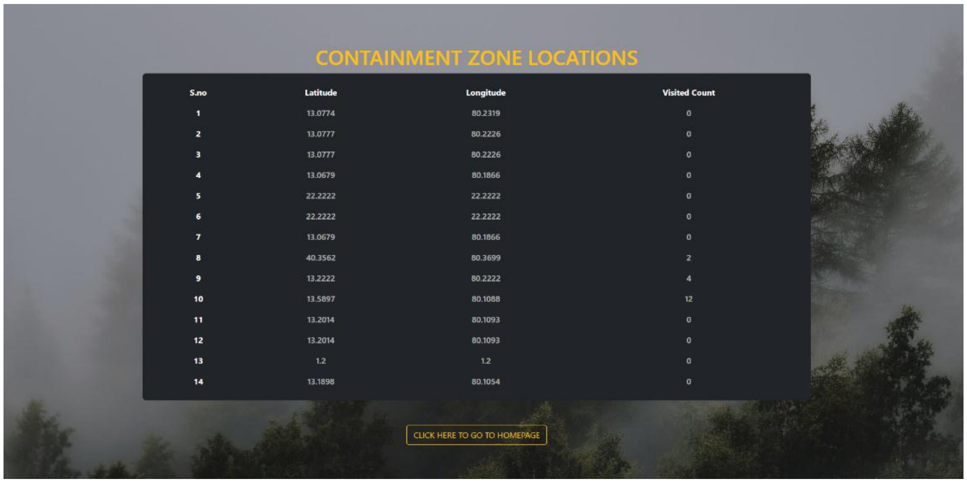
```
@app.route("/logout")
def logout():
    session['id'] = None
    session['name'] = None
    session['email'] = None
    return redirect(url_for('login'))
```



7.1 Login Page



7.2 Register Page



CONTAINMENT ZONE LOCATIONS

S.no	Latitude	Longitude	Visited Count
1	13.0774	80.2319	0
2	13.0777	80.2226	0
3	13.0777	80.2226	0
4	13.0679	80.1866	0
5	22.2222	22.2222	0
6	22.2222	22.2222	0
7	13.0679	80.1866	0
8	40.3562	80.3699	2
9	13.2222	80.2222	4
10	13.5897	80.1088	12
11	13.2014	80.1093	0
12	13.2014	80.1093	0
13	1.2	1.2	0
14	13.1898	80.1054	0

7.3 Containment Zones

7.2 Feature 2

7.1 Android sign_up

```
@app.route("/android_sign_up", methods=['GET', 'POST'])
def android_sign_up():
    if (request.method == "POST"):
        name = request.json['name']
        email = request.json['email']
        password = request.json['password']
        signup_cursor = mysql.connection.cursor()
        user_result = signup_cursor.execute(
            "SELECT * FROM users WHERE email=%s", [email]
        )
        if (user_result < 0):
            signup_cursor.close()
            return {'status': 'failuree'}
        else:
            signup_cursor.execute(
                'INSERT INTO users(user_id,name,email,password)
VALUES(%s,%s,%s,%s)', (
                    "2", name, email, password,
                )
            )
            mysql.connection.commit()
            id_result = signup_cursor.execute(
                'SELECT user_id FROM users WHERE email = %s', [email]
            )
            if (id_result > 0):
                id = signup_cursor.fetchone()
                return {"id": 1}
            signup_cursor.close()
    return {"status": "failure"}
```

7.2 Android Coordinates

```
@app.route("/post_user_location_data", methods=['GET', 'POST'])
def post_user_location_data():
    if (request.method == "POST"):
        id = request.json['id']
        lat = request.json['lat']
        lon = request.json['long']
        ts = request.json['timestamp']
        user_location_cursor = mysql.connection.cursor()
        user_location_cursor.execute(
            'INSERT INTO usersloc(latitude,longitude,user_id,timestamp)
VALUES(%s,%s,%s,%s)', (
                lat, lon, id, ts
            )
        )
        mysql.connection.commit()
        return {"response": "success"}
```

7.3 Android Warning

```
@app.route("/send_trigger", methods=["POST"])
def send_trigger():
    if (request.method == "POST"):
        email = request.json['email']
        location_id = request.json['id']
        location_cursor = mysql.connection.cursor()
        user_result = location_cursor.execute(
            "SELECT location_visited FROM location WHERE location_id=%s", [
                location_id]
        )
        if (user_result == 0):
            return {"response": "failure"}
        else:
            res = location_cursor.fetchone()
            print(res[0])
            visited = res[0]
            visited = visited+1
            location_cursor.execute(
                "UPDATE location SET location_visited = %s WHERE location_id=%s",
                (visited, location_id)
            )
            mysql.connection.commit()
        send_mail(email)
        return {"response": "success"}
```

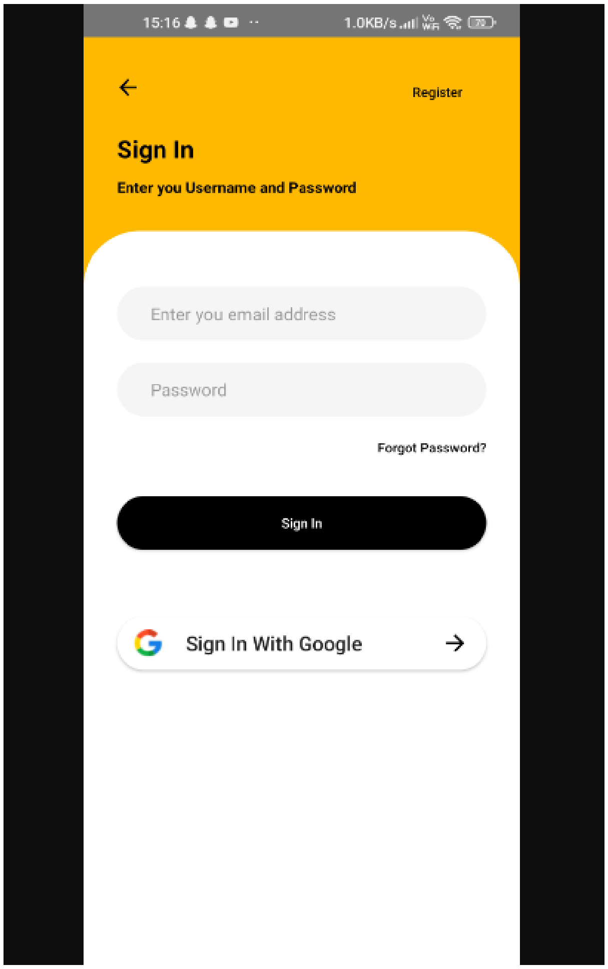
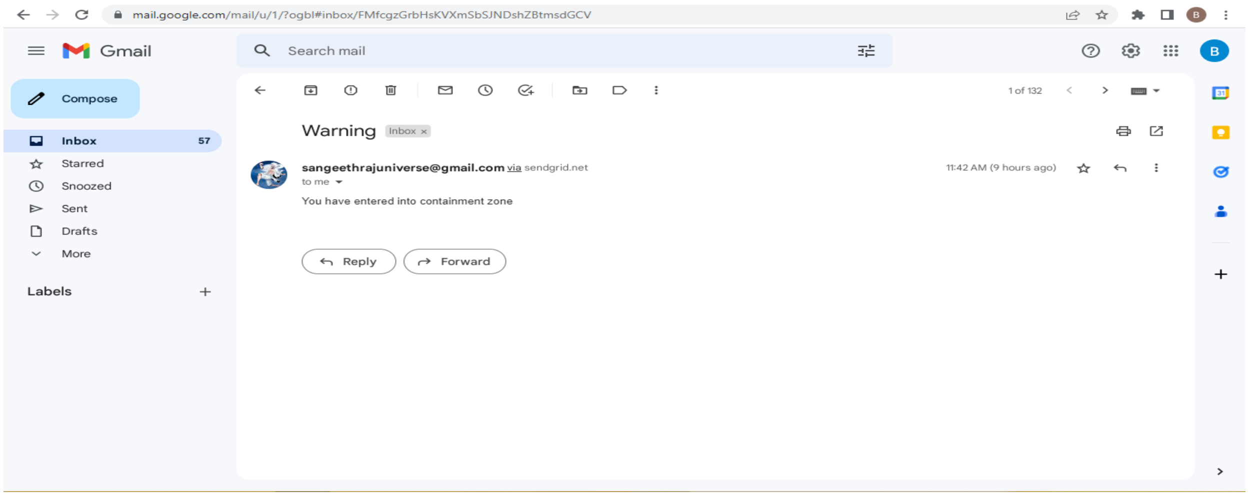


Fig 7.4 Android sign in



Warning Inbox ×

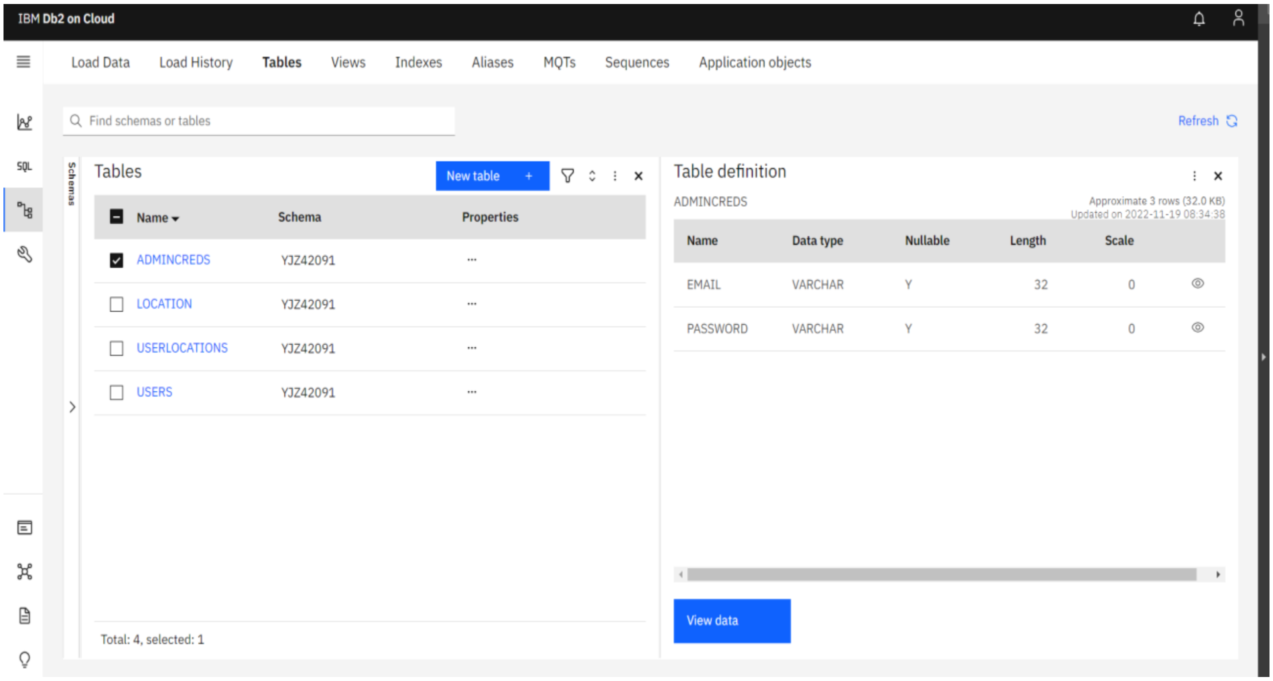
sangeethrajuniverse@gmail.com via sendgrid.net
to me

11:42 AM (9 hours ago)

You have entered into containment zone

Fig 7.5 Alert Notification

7.3 Database Schema



Tables

☐ Name	Schema	Properties
☑ ADMINCREDS	YJZ42091	···
☐ LOCATION	YJZ42091	···
☐ USERLOCATIONS	YJZ42091	···
☐ USERS	YJZ42091	···

Total: 4, selected: 1

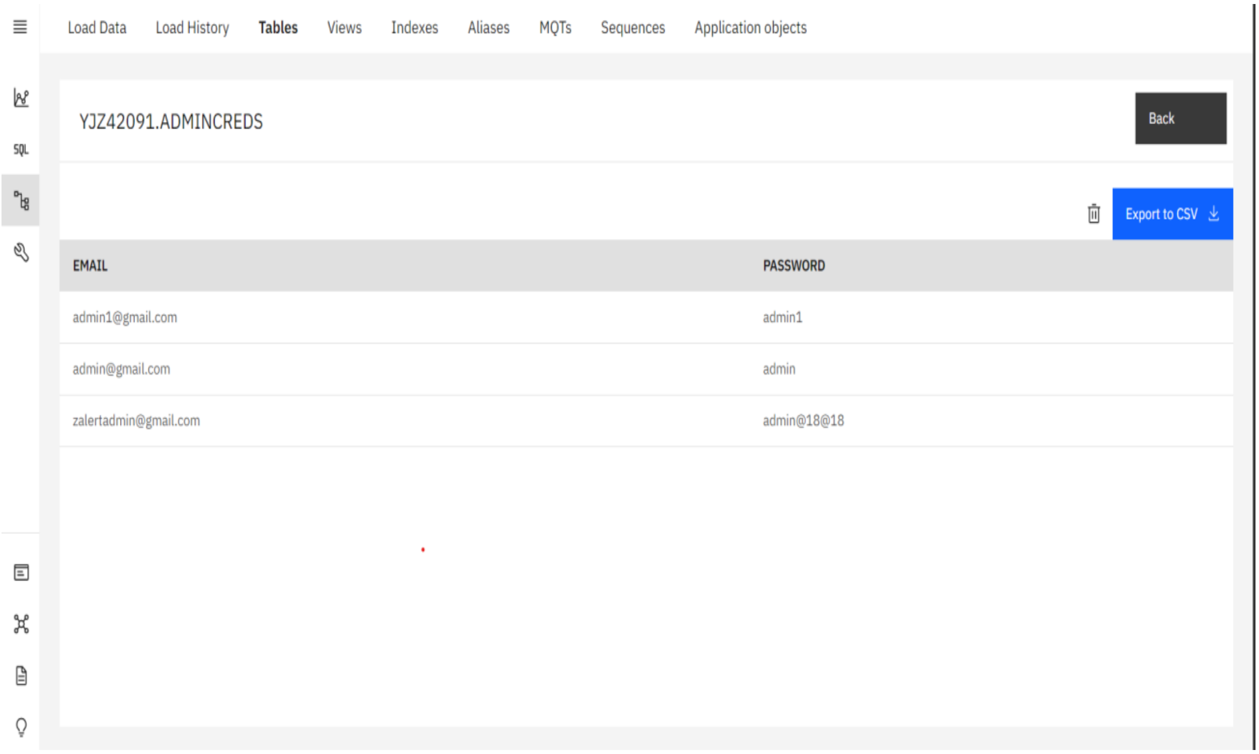
Table definition

ADMINCREDS

Approximate 3 rows (32.0 KB)
Updated on 2022-11-19 08:34:38

Name	Data type	Nullable	Length	Scale
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

Fig 7.6 IBM Database Table



YJZ42091.ADMINCREDS

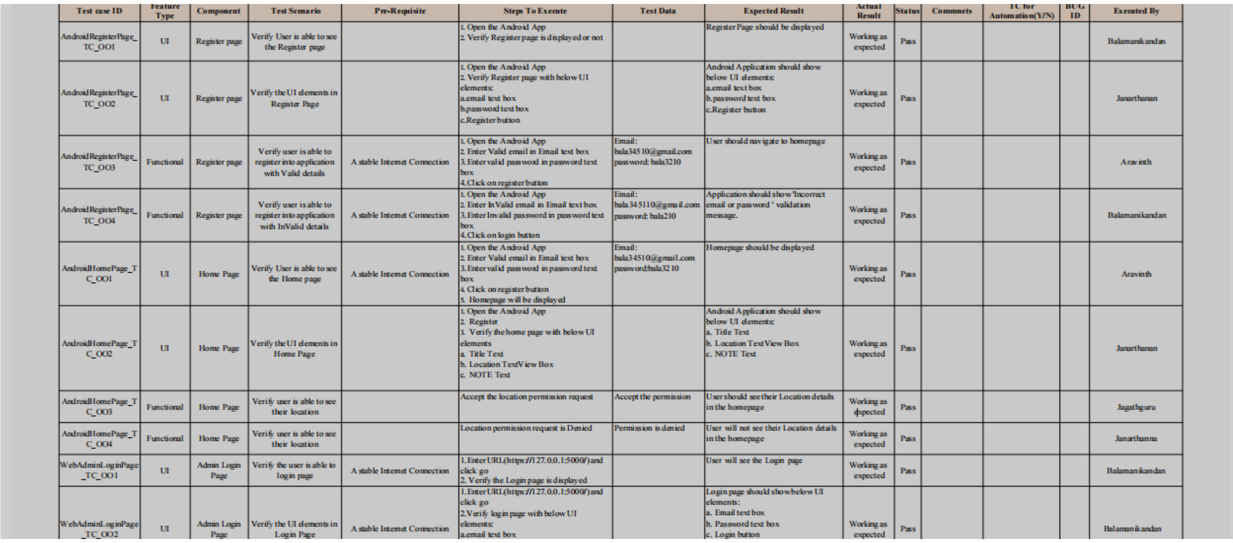
EMAIL	PASSWORD
admin1@gmail.com	admin1
admin@gmail.com	admin
zalertadmin@gmail.com	admin@18@18

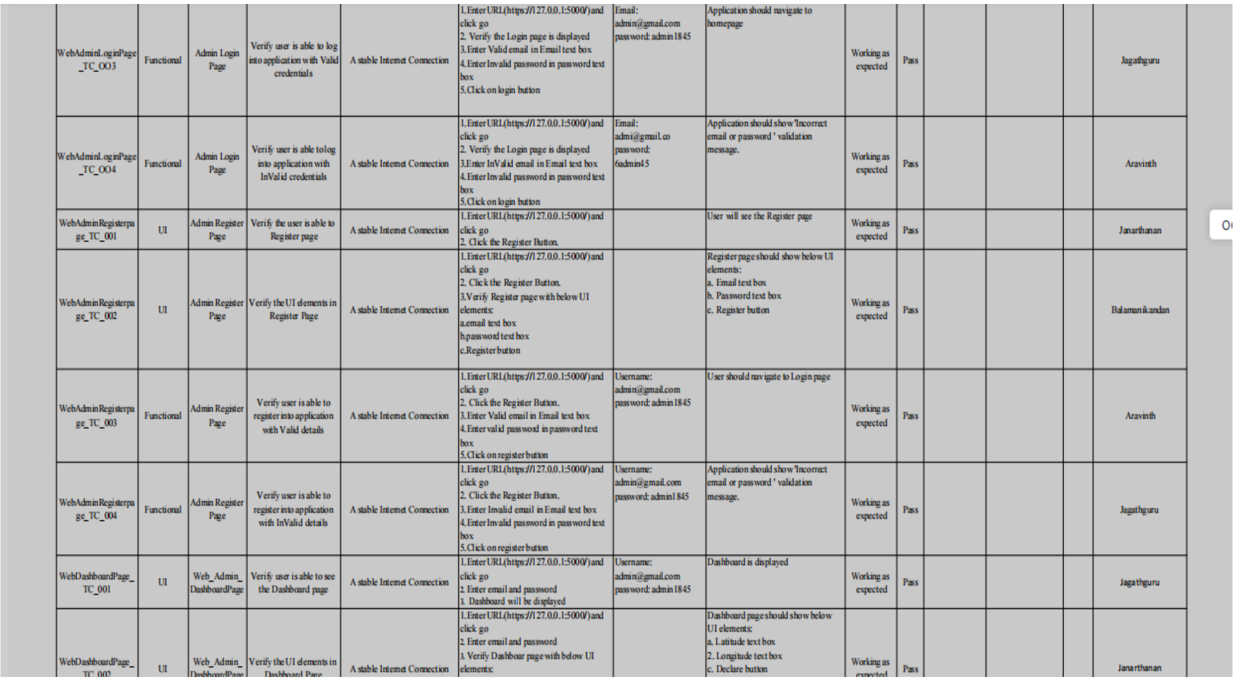
Fig 7.7 Data Storage

CHAPTER 8

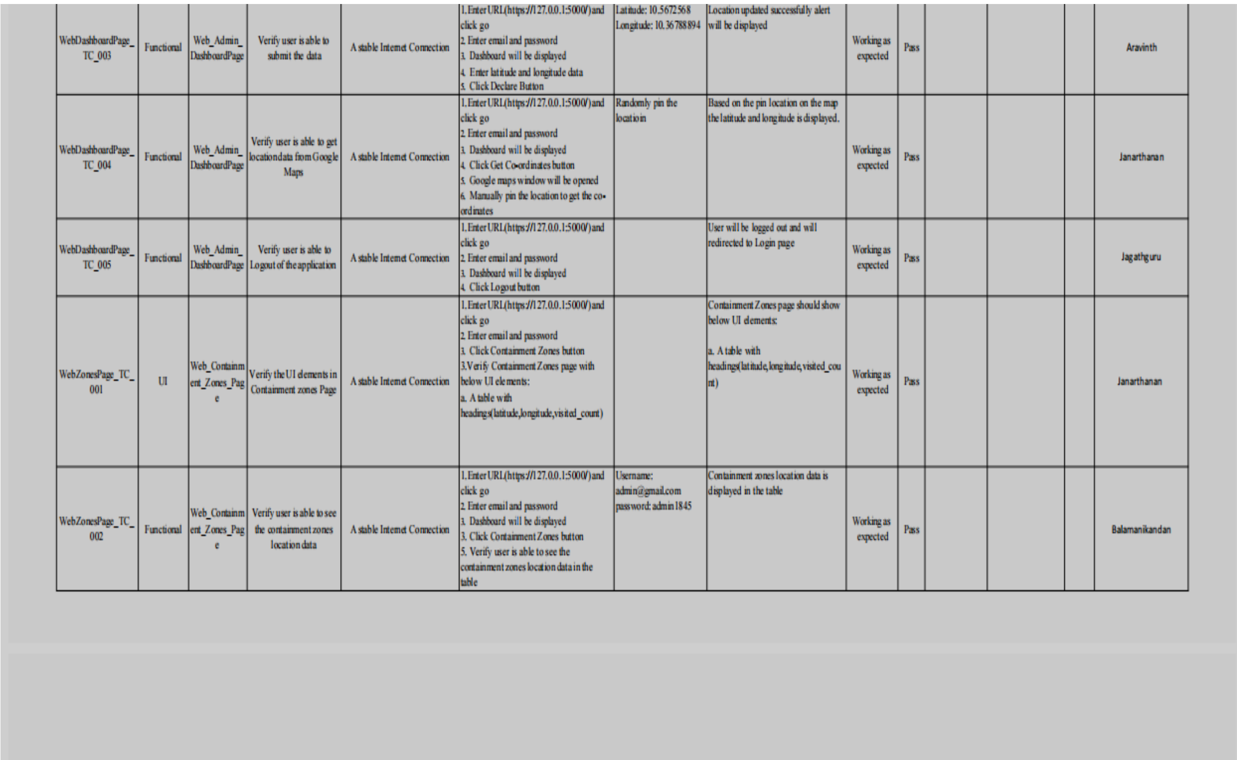
TESTING

8.1 Test cases





Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
AndroidRegisterPage_ TC_OO1	UI	Register page	Verify User is able to see the Register page		1. Open the Android App 2. Verify Register page is displayed or not		Register Page should be displayed	Working as expected	Pass				Balamanikandan
AndroidRegisterPage_ TC_OO2	UI	Register page	Verify the UI elements in Register Page		1. Open the Android App 2. Verify Register page with below UI elements: a.email text box b.password text box c.Register button		Android Application should show below UI elements: a.email text box b.password text box c.Register button	Working as expected	Pass				Janarthanan
AndroidRegisterPage_ TC_OO3	Functional	Register page	Verify user is able to register into application with Valid details	A stable Internet Connection	1. Open the Android App 2. Enter Valid email in Email text box 3. Enter valid password in password text box 4. Click on register button	Email: bala345 10@gmail.com password: bala3210	User should navigate to homepage	Working as expected	Pass				Aravinth
AndroidRegisterPage_ TC_OO4	Functional	Register page	Verify user is able to register into application with InValid details	A stable Internet Connection	1. Open the Android App 2. Enter InValid email in Email text box 3. Enter Invalid password in password text box 4. Click on login button	Email: bala 345110@gmail.com password: bala210	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Balamanikandan
AndroidHomePage_T C_OO1	UI	Home Page	Verify User is able to see the Home page	A stable Internet Connection	1. Open the Android App 2. Enter Valid email in Email text box 3. Enter valid password in password text box 4. Click on register button 5. Homepage will be displayed	Email: bala34510@gmail.com password:bala32 10	Homepage should be displayed	Working as expected	Pass				Aravinth
AndroidHomePage_T C_OO2	UI	Home Page	Verify the UI elements in Home Page		1. Open the Android App 2. Register 3. Verify the home page with below UI elements a. Title Text b. Location TextView Box c. NOTE Text		Android Application should show below UI elements: a. Title Text b. Location TextView Box c. NOTE Text	Working as expected	Pass				Janarthanan
AndroidHomePage_T C_OO3	Functional	Home Page	Verify user is able to see their location		Accept the location permission request	Accept the permission	User should see their Location details in the homepage	Working as expected	Pass				Jagathguru
AndroidHomePage_T C_OO4	Functional	Home Page	Verify user is able to see their location		Location permission request is Denied	Permission is denied	User will not see their Location details in the homepage	Working as expected	Pass				Janarthanna
WebAdminLoginPage _TC_OO1	UI	Admin Login Page	Verify the user is able to login page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Verify the Login page is displayed		User will see the Login page	Working as expected	Pass				Balamanikandan
WebAdminLoginPage _TC_OO2	UI	Admin Login Page	Verify the UI elements in Login Page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2.Verify login page with below UI elements: a.email text box		Login page should showbelow UI elements: a. Email text box b. Password text box c. Login button	Working as expected	Pass				Balamanikandan
WebAdminLoginPage _TC_OO3	Functional	Admin Login Page	Verify user is able to log into application with Valid credentials	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Verify the Login page is displayed 3. Enter Valid email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	Email: admin@gmail.com password: admin1845	Application should navigate to homepage	Working as expected	Pass				Jagathguru
WebAdminLoginPage _TC_OO4	Functional	Admin Login Page	Verify user is able tolog into application with InValid credentials	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Verify the Login page is displayed 3.Enter InValid email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	Email: admi@gmail.co password: 6admin45	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Aravinth
WebAdminRegisterpa ge_TC_001	UI	Admin Register Page	Verify the user is able to Register page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Click the Register Button.		User will see the Register page	Working as expected	Pass				Janarthanan
WebAdminRegisterpa ge_TC_002	UI	Admin Register Page	Verify the UI elements in Register Page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Click the Register Button. 3.Verify Register page with below UI elements: a.email text box b.password text box c.Register button		Register page should show below UI elements: a. Email text box b. Password text box c. Register button	Working as expected	Pass				Balamanikandan
WebAdminRegisterpa ge_TC_003	Functional	Admin Register Page	Verify user is able to register into application with Valid details	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Click the Register Button. 3. Enter Valid email in Email text box 4. Enter valid password in password text box 5. Click on register button	Username: admin@gmail.com password: admin1845	User should navigate to Login page	Working as expected	Pass				Aravinth
WebAdminRegisterpa ge_TC_004	Functional	Admin Register Page	Verify user is able to register into application with InValid details	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Click the Register Button. 3. Enter Invalid email in Email text box 4. Enter Invalid password in password text box 5. Click on register button	Username: admin@gmail.com password: admin1 845	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Jagathguru
WebDashboardPage_ TC_001	UI	Web_Admin_ DashboardPage	Verify user is able to see the Dashboard page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Enter email and password 3. Dashboard will be displayed	Username: admin@gmail.com password: admin1845	Dashboard is displayed	Working as expected	Pass				Jagathguru
WebDashboardPage_ TC_002	UI	Web_Admin_ DashboardPage	Verify the UI elements in Dashboard Page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/)and click go 2. Enter email and password 3. Verify Dashboar page with below UI elements:		Dashboard page should show below UI elements: a. Latitude text box 2. Longitude text box c. Declare button	Working as expected	Pass				Janarthanan



WebDashboardPage_TC_003	Functional	Web_Admin_DashboardPage	Verify user is able to submit the data	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/) and click go 2. Enter email and password 3. Dashboard will be displayed 4. Enter latitude and longitude data 5. Click Declare Button	Latitude: 10.5672568 Longitude: 10.36788894	Location updated successfully alert will be displayed	Working as expected	Pass				Aravinth
WebDashboardPage_TC_004	Functional	Web_Admin_DashboardPage	Verify user is able to get locationdata from Google Maps	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/) and click go 2. Enter email and password 3. Dashboard will be displayed 4. Click Get Co-ordinates button 5. Google maps window will be opened 6. Manually pin the location to get the co-ordinates	Randomly pin the locatioin	Based on the pin location on the map the latitude and longitude is displayed.	Working as expected	Pass				Janarthanan
WebDashboardPage_TC_005	Functional	Web_Admin_DashboardPage	Verify user is able to Logout of the application	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/) and click go 2. Enter email and password 3. Dashboard will be displayed 4. Click Logout button		User will be logged out and will redirected to Login page	Working as expected	Pass				Jagathguru
WebZonesPage_TC_001	UI	Web_Containment_Zones_Page	Verify the UI elements in Containment zones Page	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/) and click go 2. Enter email and password 3. Click Containment Zones button 3.Verify Containment Zones page with below UI elements: a. A table with headings(latitude,longitude,visited_count)		Containment Zones page should show below UI elements: a. A table with headings(latitude,longitude,visited_count)	Working as expected	Pass				Janarthanan
WebZonesPage_TC_002	Functional	Web_Containment_Zones_Page	Verify user is able to see the containment zones location data	A stable Internet Connection	1. Enter URL(https://127.0.0.1:5000/) and click go 2. Enter email and password 3. Dashboard will be displayed 3. Click Containment Zones button 5. Verify user is able to see the containment zones location data in the table	Username: admin@gmail.com password: admin1845	Containment zones location data is displayed in the table	Working as expected	Pass				Balamanikandan

FIG 8.1 Test Case Report

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Containment Zone Alerting Application project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	6	9	10	32
Duplicate	1	0	3	2	6
External	1	2	9	5	17

Fixed	10	7	13	12	42
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	20	15	34	29	98

3. Test Case Analysis

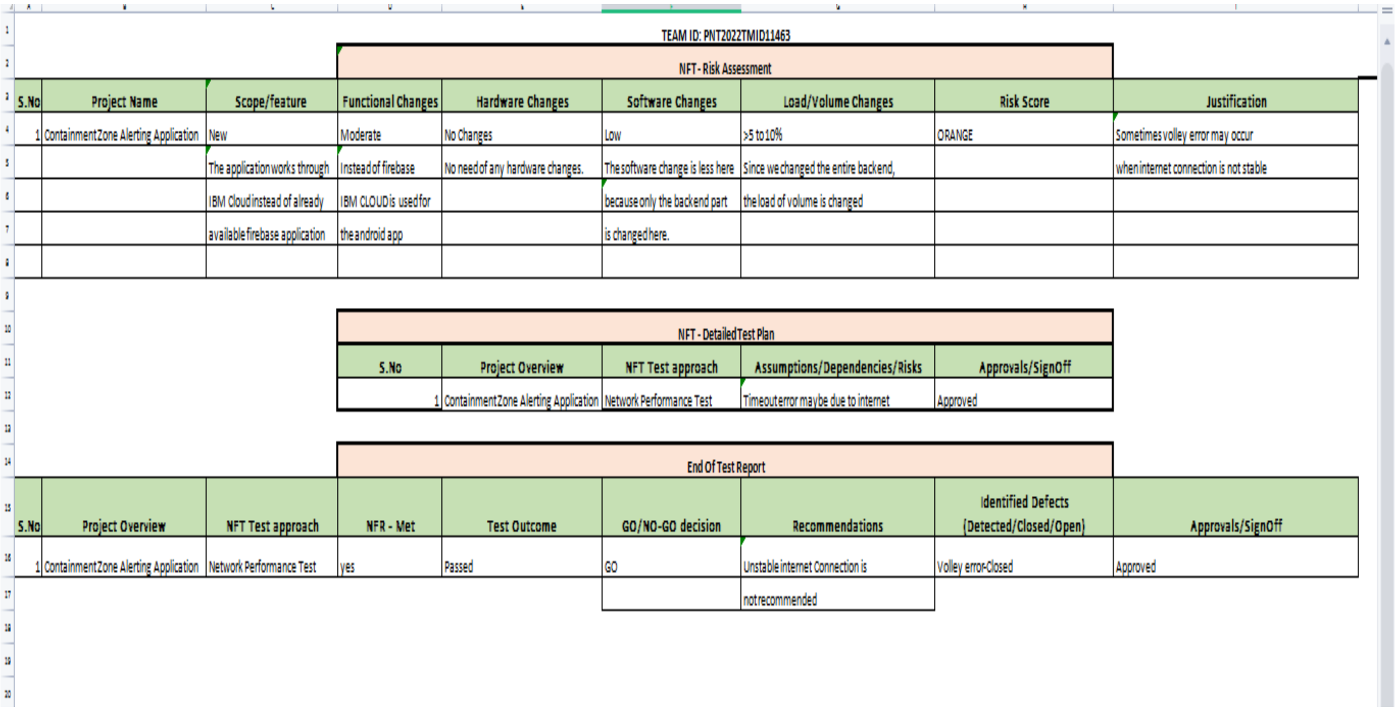
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	45	0	0	45
Security	2	0	0	2
Outsource Shipping	5	0	0	5
Exception Reporting	14	0	0	14
Final Report Output	8	0	0	8
Version Control	3	0	0	3

CHAPTER 9

RESULTS

9.1 Performance Metrics



TEAM ID: PNT2022TMID11463

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	ContainmentZone Alerting Application	New	Moderate	No Changes	Low	>5 to 10%	ORANGE	Sometimes volley error may occur
		The application works through	Instead of firebase	No need of any hardware changes.	The software change is less here	Since we changed the entire backend,		when internet connection is not stable
		IBM Cloud instead of already	IBM CLOUD is used for		because only the backend part	the load of volume is changed		
		available firebase application	the android app		is changed here.			

NFT - DetailedTest Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	ContainmentZone Alerting Application	Network Performance Test	Timeout error maybe due to internet	Approved

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	ContainmentZone Alerting Application	Network Performance Test	yes	Passed	GO	Unstable internet Connection is	Volley error-Closed	Approved
						not recommended		

Fig 9.1 Performance Metrics

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

• The application aims at warning the users before entering a Covid containment zone, thereby providing a chance for precautionary measures.

• The application solves user confusions on information about containment zones

• The user need not worry if to enter a zone or not due to the lack of information

Covid Containment Zones.

• The app monitors the user's location at real time

• User privacy is maintained. Very little information on the user is collected.

• Easy administration of the users and zones through a separate portal

DISADVANTAGES

• The application is useful for the user only if the user carries his/her mobile phone

while traveling.

• The application requires the user to have access to Internet.

• The location services in the user's smartphone should be turned on for the application to monitor the real time location of the user and alert, in case if the user enters a containment zone.

CHAPTER 11

CONCLUSION

Users of the application can quickly view the designated Covid-19 containment zones on a Google map. With the worrisome rise in Covid-19-affected cases around the globe, this developed application might be used as a tool to raise more public awareness.

This application keeps track of the user's location and determines if it appears on the list of designated containment zones. If a user enters or is within a Covid Containment Zone, an immediate alert notification is given to the user.

The application delineates the containment areas and emphasizes the necessity for additional precautions to be taken in the fight against COVID-19. The application has been tested in various locations and has been found to yield accurate results.

CHAPTER 12

FUTURE SCOPE

The application has a broad scope of improvements in the future. The location details of the places visited could be analyzed with the help of Machine Learning to discover interesting patterns.

If the application is teamed up with a government authority, the authenticity of the application will spike. Furthermore, future improvements to the application could include measures to send out an emergency signal in case if GPS signal or connectivity to the Internet is lost and to improve more precise tracking of user location.

CHAPTER 13

APPENDIX

TEAM ID: PNT2022TMID11463

Source Code

The source code of the application could have been made available in the GitHub repository IBM-EBPL/ IBM-Project-43641-1660718565