Project Report

Team ID	**PNT2022TMID14431**
Project Name	**CONTAINMENT ZONE ALERTING APPLICATION**

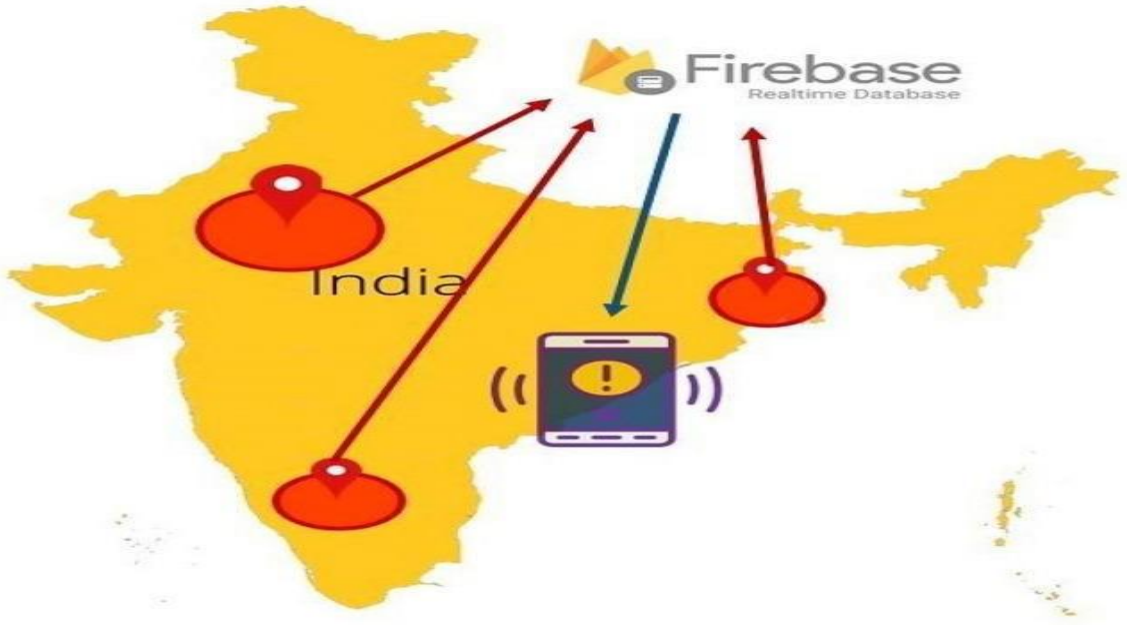
1. INTRODUCTION

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.

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



2. LITERATURE SURVEY:

2.1 Existing problem:

People doesn't have proper knowledge about containment zones since they do change daily and hard to keep updated and if they are not updated properly, they will lead to wide spread of disease.

2.2 References:

1. https://ieeexplore.ieee.org/document/9711880
2. https://ieeexplore.ieee.org/document/9432254
3. https://ieeexplore.ieee.org/document/9356316
4. https://ieeexplore.ieee.org/document/9388625
5. https://ieeexplore.ieee.org/document/9609407
6. https://ieeexplore.ieee.org/document/9356316

2.3.Problem Statement Definition:

Problem Statement (PS)	I am (USER)	I'm trying to	But	Because	Which makes me feel
PS-1	Is it belonging all the age groups	Share my status to Everyone	Is it any compulsion to do that	It makes me embarrass e d	As an influence person by spreadingan irrelevant content to my nativeareas
PS-2	If a mobile app user whether we give all personal details what the app expecting from us.	The Given details should be verified by an admin side.	There are many apps cannot securethe data in a proper manner	The updatingof an app cannot be prolonged.	Unsecured to givemy personal data's.
PS - 3	Whether the updating of containment zone is plotted in a really affected area.	There is any way to share the alerted zone to others	It has an any constraint to share the location within the particular distance.	I am trying to reach beyond theareas.	Niche intimate toour others.
PS - 4	The alert messages arereceived by where only entered into the alerted zone	To stay awake from the alerted zone	Usage of mobile phone are taken out frequently to check out the messages	Just to checkout whetherI entered in tothe containmen tzone or not.	Phone Holder for thefrequent usage.
PS - 5	There is any feature to connect the Emergency Service.	When there is a helpless situation appeared.	The connections to helpline response as much earlier	On that hectic moment it was an onlyhope.	Pulled down into theBlinded mind

PNT2022TMID14431

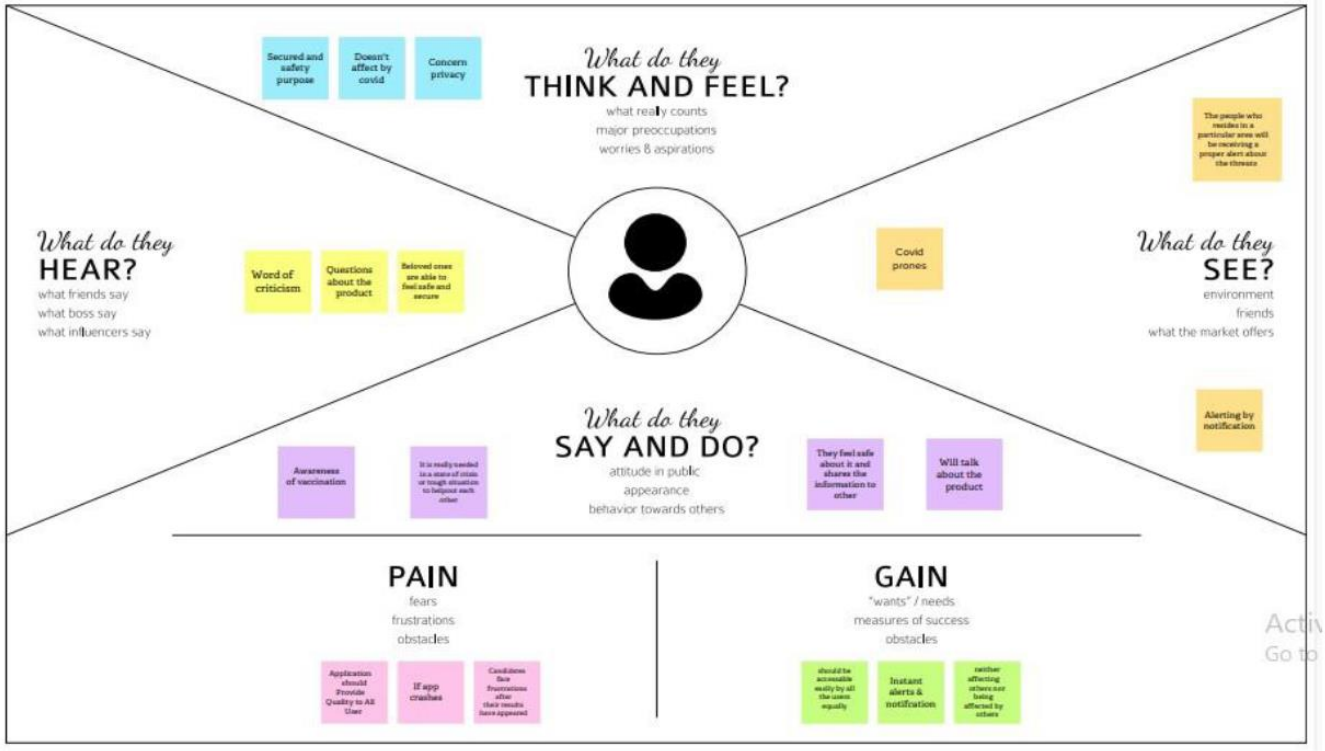
Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	The users of the Android application can seewhere the containment zones are located. Additionally, it alerts the user if they remaininside containment zones or cross their boundaries.
2.	Idea / Solution description	To contain infectious disease outbreaks, contacttracing and case isolation are utilized, as they were with the coronavirus epidemic in 2019. (COVID-19). The features of the disease and theresponse from public health organizations will determine if this technique will be successful in achieving control. Long intervals between the onset of symptoms and isolation, a dearth of cases discovered through contact tracing, and rising transmission prior to symptoms all reducethe likelihood of control.
3.	Novelty / Uniqueness	Due to its crucial role in the creation of new vaccinations, scientific novelty is significant throughout the pandemic. In order to broaden teams' search operations for a larger scope of resources needed to handle the global challenge, parachuting collaboration and international collaboration are key channels. A pandemic call for the preservation of a cooperative scientific community that goes beyond established networks and nationalism, according to findings. Evidence from COVID-19 suggests that pandemics spur scientific innovation. The search for new vaccinations andtreatments during the epidemic is driven by scientific novelty. The expansion of teams' search efforts for a greater impact requires international and first-time collaboration, whichare crucial channels.
4.	Social Impact / Customer Satisfaction	The 2019 Coronavirus 1 disease (Covid-19 2) pandemic 3 has grown to be the highest total demand for immediate assistance in the twenty-first century. The World Health Organization (WHO) identified Covid-19 2 to be a Public Health immediately needed of international business place on January 30, 2020. (PHEIC). TheCovid-19 2 pandemic 3 have greater size, range, degree of readiness to move limitations, travel stops, and edge close to stone edging than thosein preceding PHEICs during the past hundred years

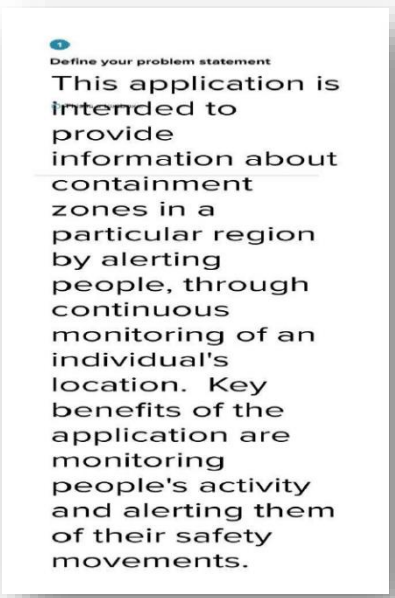
5.	Business Model (Revenue Model)	Scale-copy refers to many experiments, thinkingthrough how to create value for clients and other interested companies on the back end, and capturing value through the organization's mechanisms for converting value into money. The company's sector is the first mathematical factor that has influence over this. Higher degrees of inclusivity have typically been accompanied by an improvement in the ability to bounce back. The likelihood that someone receiving assistance or commodities will continue their relationship with the same organization providing goods increases with theoffering's size and distance down.
6.	Scalability of the Solution	The Covid-19 pandemic's confluence with the accompanying lockdown and many businesses' need to operate entirely remotely.In February, our Upscale programme for mid- stage, scaling businesses had just begun. Whatwas it like for businesses that were in the process of rapidly scaling? ODDBOX doubled their predicted growth prediction and saw growth of 600% year over year. Farewill had to adjust to entirely remote service at the same time that demand increased.

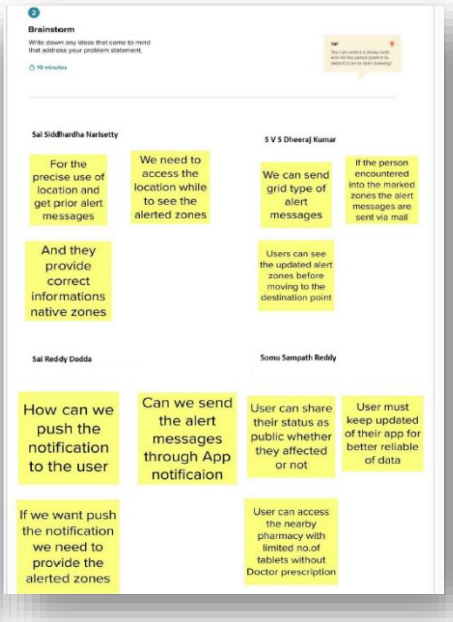
3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges

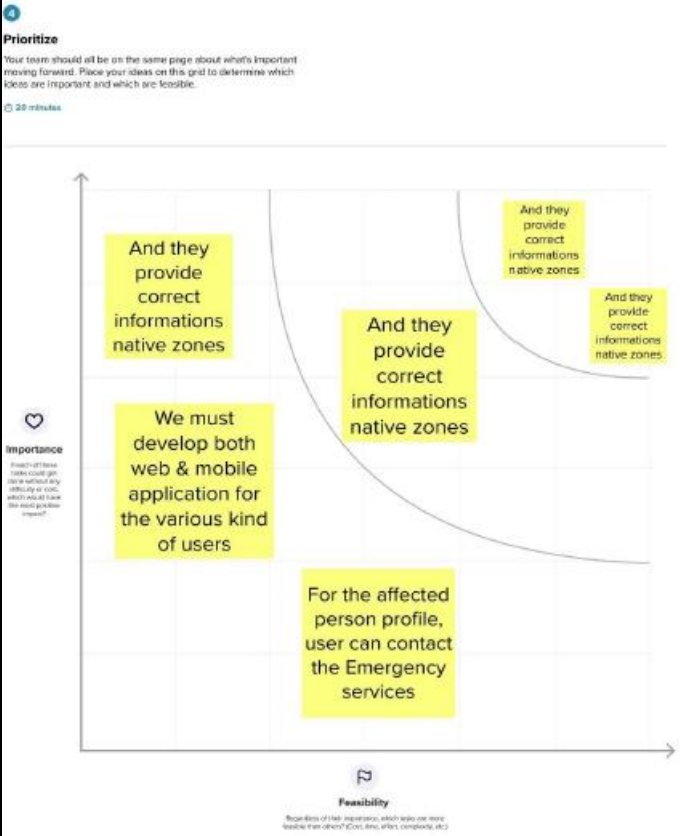


3.3 Ideation & Brainstorming



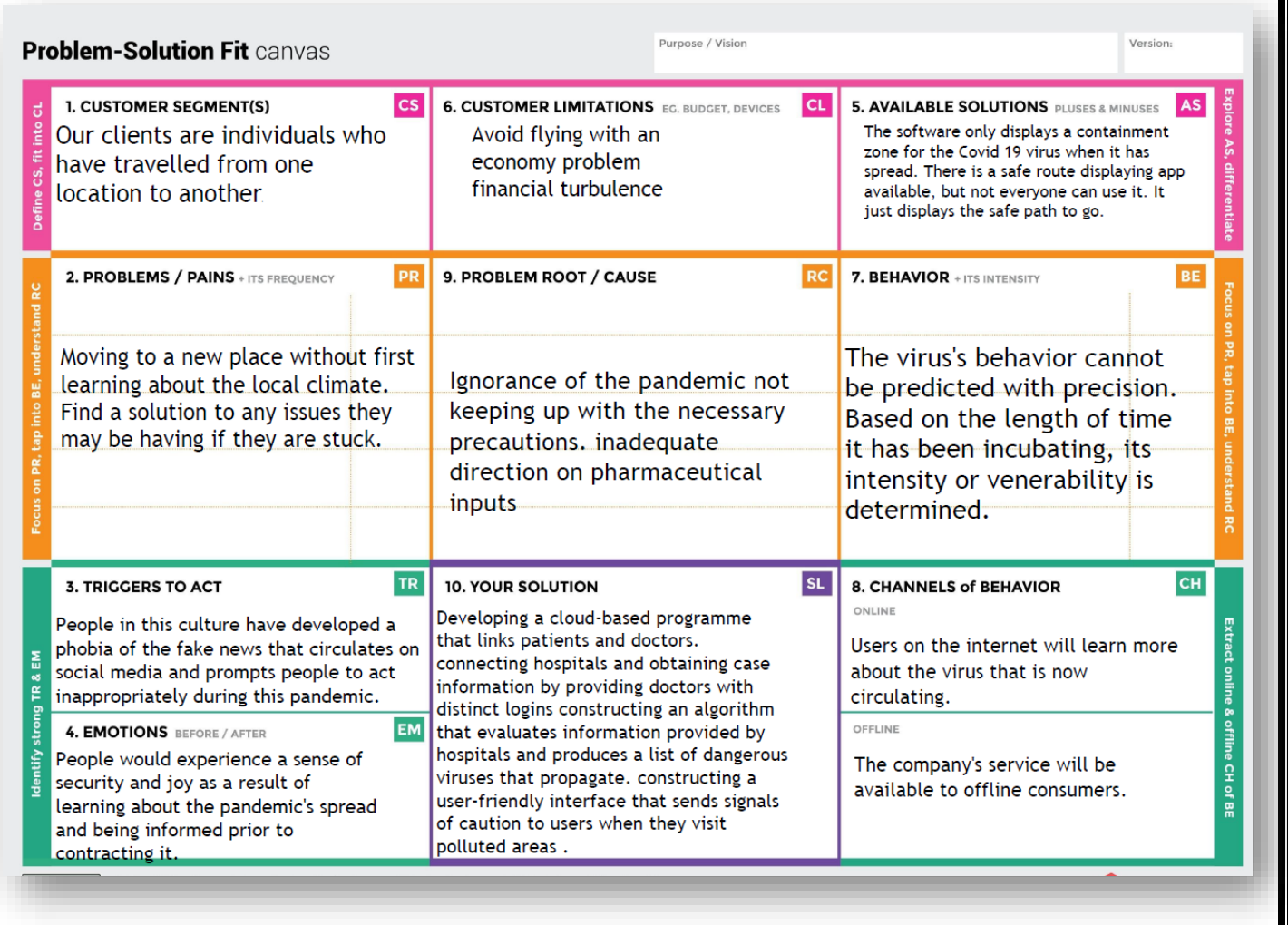


PNT2022TMID14431



PNT2022TMID14431

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile NumberRegistration 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 send if they enter the containment zone.

4.1 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

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 anyplace and monitoring users movements in pandemic zones and alerts before they are affected.

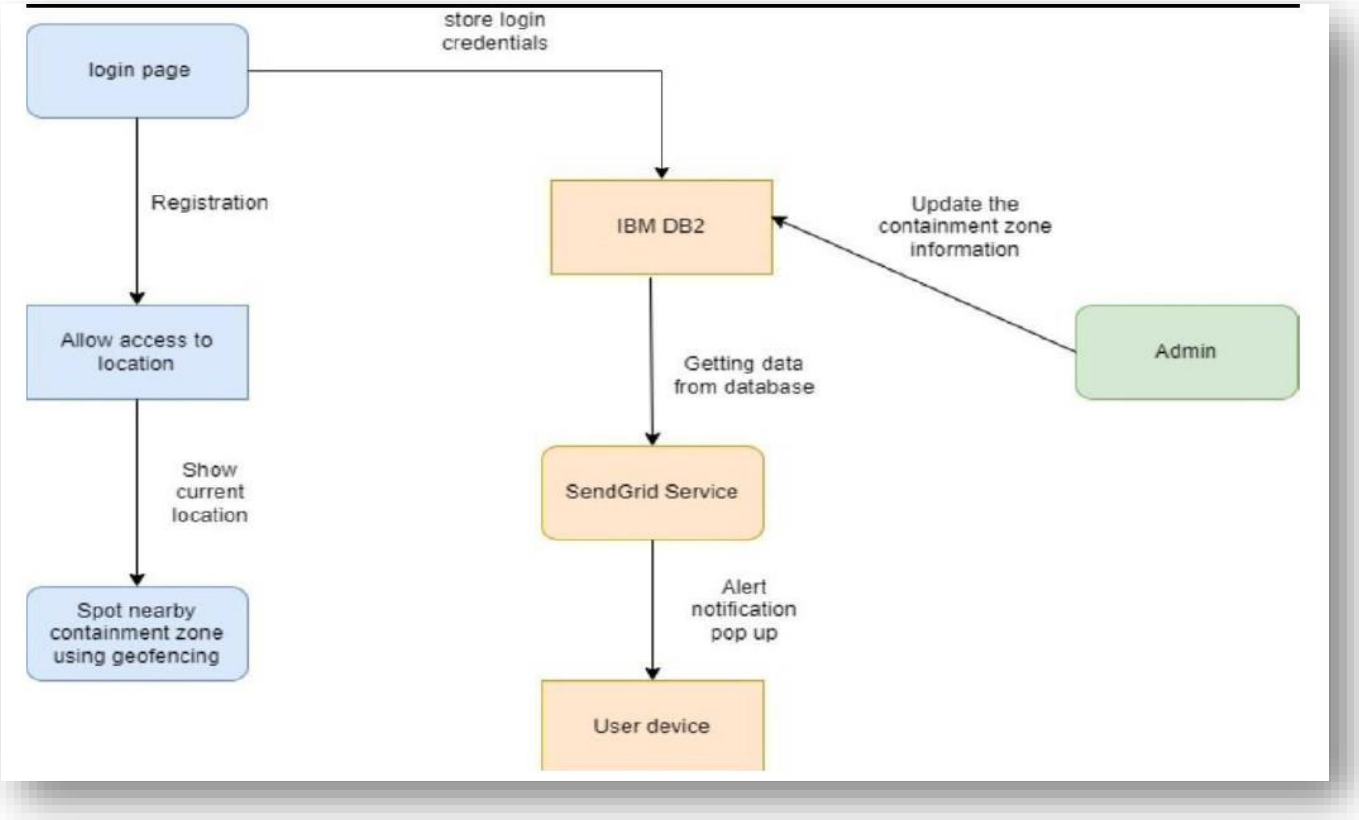
5.PROJECT DESIGN

Data Flow Diagrams

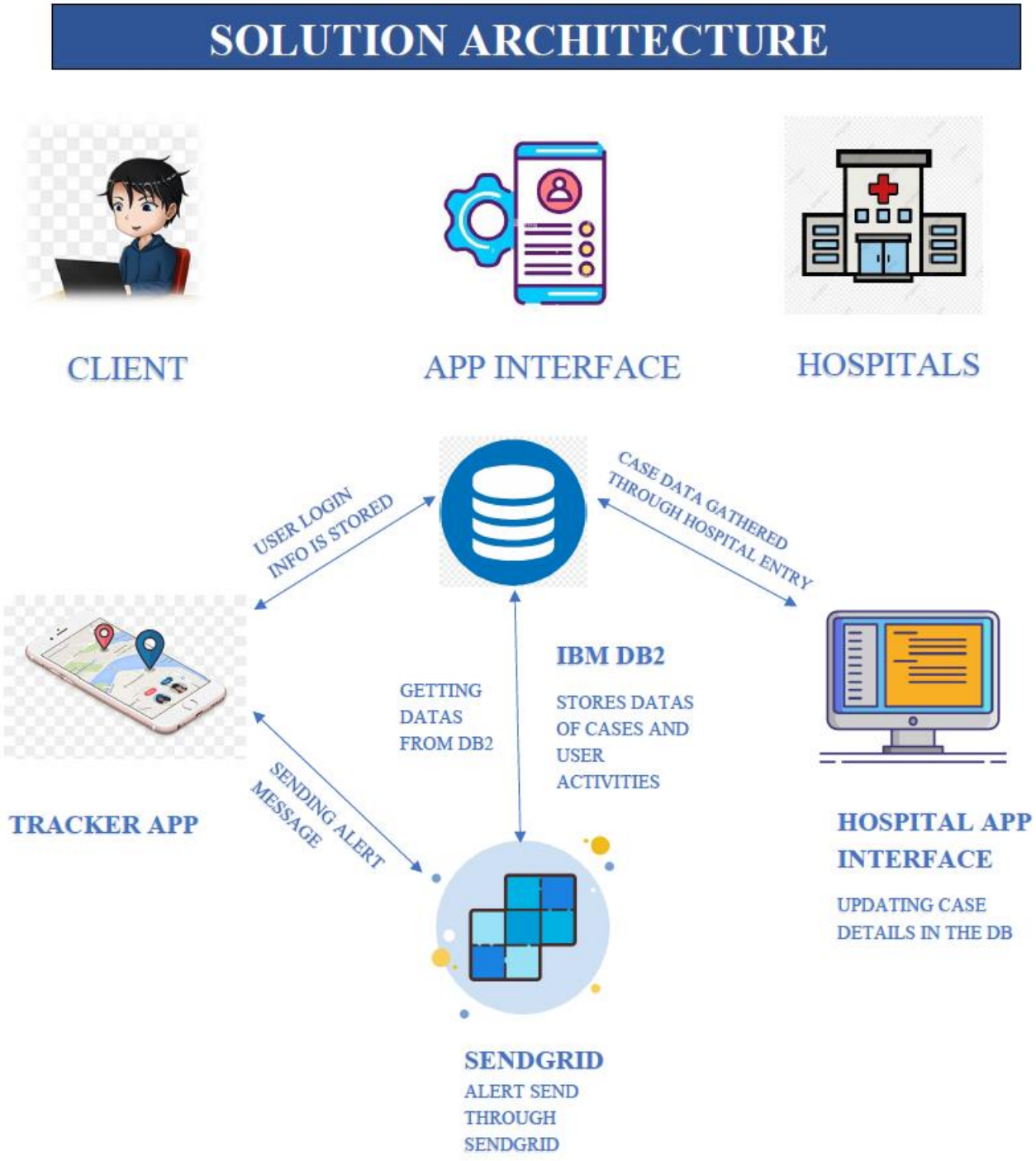
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

4.2 Data flow diagram:



5.2.SOLUTION ARCHITECURE:

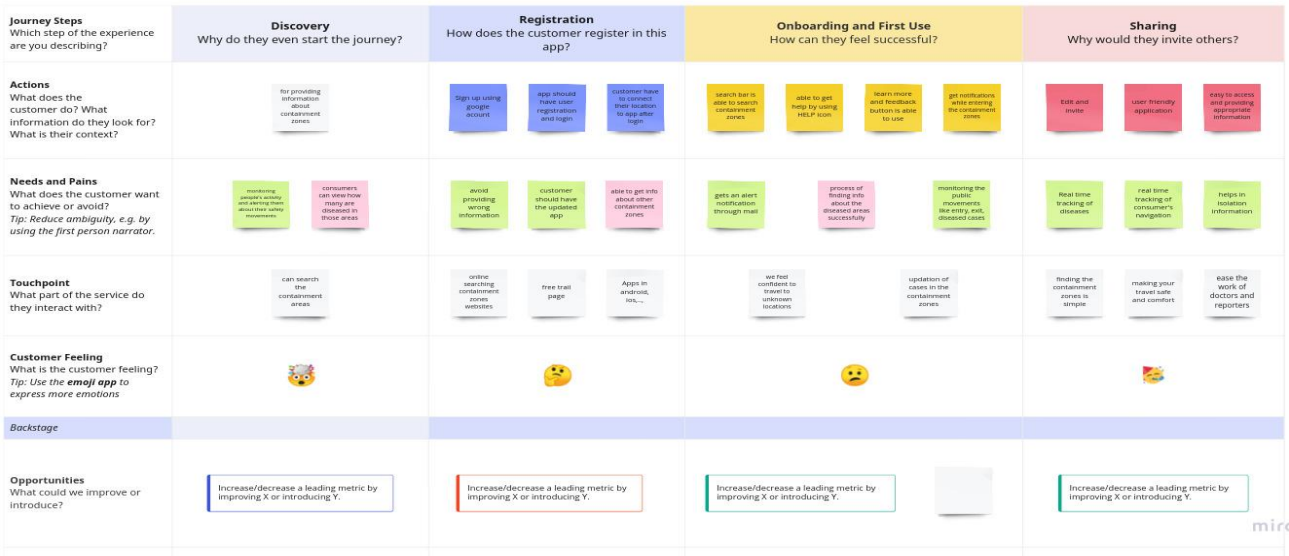


PNT2022TMID14431

TECHNICAL ARCHITECTURE:

Table-1: Components & Technologies: S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, Java, XML, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python-Flask
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Cloud Object Storage
8.	External API-1	Purpose of External API used in the application	IBM CLOUD API, Google Maps API
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes,

User Stories:



PNT2022TMID14431

6.PROJECT PLANNING & SCHEDULING:

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	User: I can register for the application byentering my email, password and verifying password.	3	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-2	User: I will receive a confirmationemail once I have registered for the application.	2	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-3	User: I can register for the applicationthrough Gmail.	5	Medium	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-4	Management: I need to register my hospitals on the site.	2	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-5	User: I can log into the application byentering my email & password	3	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Login	USN-6	Management: I need to login into my dashboard with my given hospital id and password.	5	Medium	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Dashboard	USN-7	User: I need to give permission to accessmy Contacts, Location, and Storage	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-8	User: I get access to the dashboard which shows a map with containment zones	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-9	Management: I need to enter the case information of the	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath,

			patient that visits our hospital.			Vijay
	Services	USN-10	Admin: I need to provide valid information about the pandemic out there.	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
Sprint-3	Dashboard	USN-11	Management: I need to store all the patient information on the cloud.	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Services	USN-12	Admin: I need to provide medical advice through a chatbot.	5	Medium	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-13	Admin: I need to provide medical recommendations by collaborating with top hospitals.	5	Low	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-14	Admin: I need to provide preventive measures when they travel through it.	5	High	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Registration	USN-15	User: I can register for the applicationthrough Facebook.	2	Low	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
Sprint-4		USN-16	User: I can register for the applicationthrough Twitter.	2	Low	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Services	USN-17	Admin: I need to alert the user whenthey enter pandemic zones.	3	Medium	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
		USN-18	Admin: I need to provide special services for premium users by giving services like monitoring health bytheirsmart bands.	3	Low	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay
	Data Collection	USN-19	Admin: I need to store all the user information on the cloud	5	Medium	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay

PNT2022TMID14431

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
		USN-20	Admin: I need to collect the recentlist of diseases in the world.	5	Low	Siddhardha, Dheeraj,Sai Reddy,Somu Sampath, Vijay

6.2. Sprint Delivery Schedule

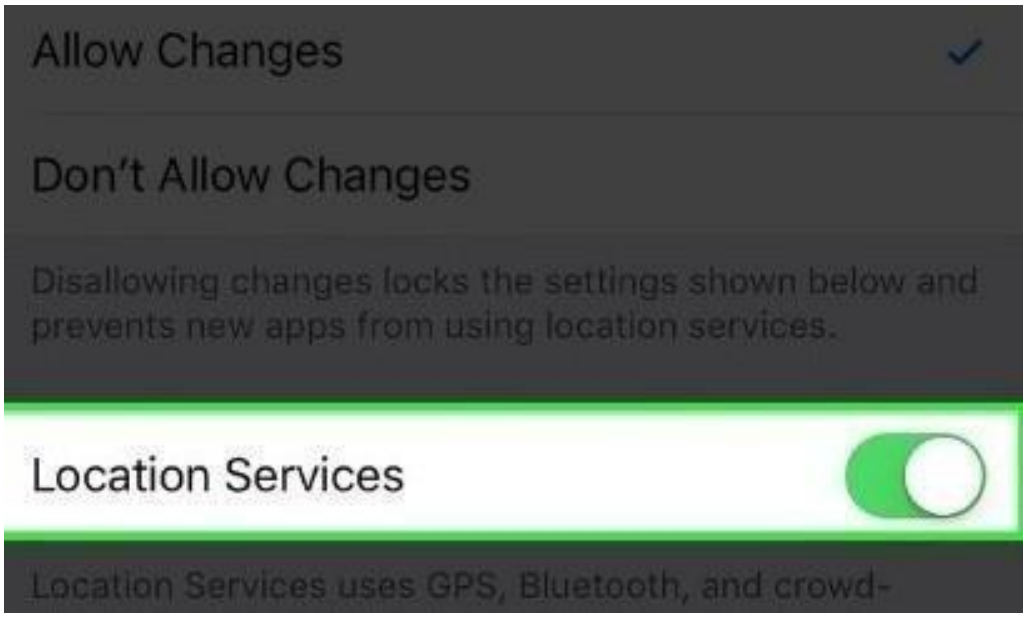
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

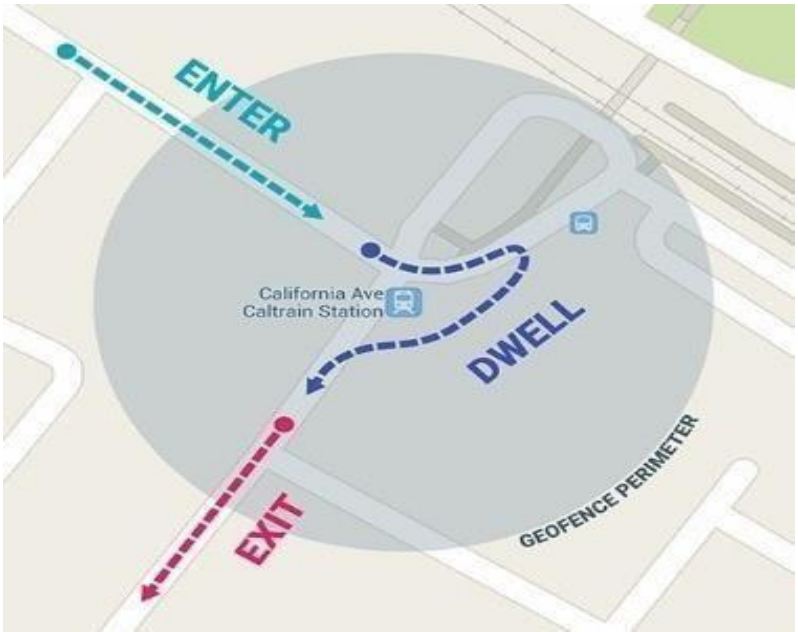
7.CODING & SOLUTIONING



GEOFENCE IN ANDROID APP :


5556:Nexus_5_API_23
9:37
Geofencing
Lat: 52.0
Long: -1.0


Remind me there
Park
Teatro Colseo
Pizza Restaurant
Shopping Mall
Buenos Aires
Choose the radius
600 meters
CONTINUE


ENTER
California Ave
Caltrain Station
DWELL
EXIT
GEOFENCE PERIMETER

8 TESTING

8.1 Test Cases

2

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Scroll down 3.Verify login/Signup popup displayed or not	http://169.51.204.215:30100/	Login/Signup popup should display	Working as expected	PASS	Successfull			ATCHAYA JENITHA KOWSALYA
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	1. Enter URL and click go 2.Click on Signup button for User 3. Verify login/Signup popup with below UI elements: a.Id text box b.password text box c.Login button d.New customer? Create account link e.Lost password? Recovery password link	http://169.51.204.215:30100/	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Lost password? Recovery password link	Working as expected	PASS	Successful			PRIYADHARSHINI KOWSALYA PRIYADHARSHINI
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL(https://shopzone.com/) and click go 2.Click on My Account dropdown button 3. Enter Valid ID in ID text box 4. Enter valid password in password text box 5. Click on login button	ID: 3342 password: Testing123	User should navigate to user account homepage	Working as expected	PASS	Successful			ATCHAYA PRIYADHARSHINI

Test Case (SPRINT 01) 3

Test Case (SPRINT 01) 3

LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30100/) and click go 2.Click on My Account dropdown button 3.Enter inValid ID in ID text box 4.Enter valid password in password text box 5.Click on login button	ID: 3342 password: Testing123	Application should show 'Incorrect email or password' validation message.	Working as expected	PASS	Successful			JENITHA
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30100/) and click go 2.Click on My Account dropdown button 3. Enter Valid ID in ID text box 4. Enter invalid password in password text box 5.Click on login button	ID: 3342 password: Testing123678687878/88/8	Application should show 'Incorrect email or password' validation message.	Working as expected	PASS	Successful			PRIYADHARSHINI
LoginPage_TC_OO6	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(http://169.51.204.215:30100/) and click go 2.Click on My Account dropdown button 3.Enter inValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 3342 password: Testing123	Application should show 'Incorrect email or password' validation message.	Working as expected	PASS	Successful			ATCHAYA

Test Case (SPRINT 01) 4

LoginPage_TC_007	Functional	Login page	Verify User is able to log into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter invalid password in password text box 5.Click on login button	ID: 5121 Password: Testing123	Application should show 'correct email or password' validation message.	Working as expected	PASS	Successful			ATCHAYA PRIYADHARSHINI
LoginPage_TC_008	Functional	Login page for ADMIN	Verify User is able to log into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106) and click go 2.Click on My Account dropdown button 3.Enter Valid ID in ID text box 4.Enter valid password in password text box 5.Click on login button	ID: 1111 password: 5678	Application should show 'correct email or password' validation message.	Working as expected	PASS	Successful			PRIYADHARSHINI KOWSALYA
LoginPage_TC_009	UI	ADMIN PAGE	Verify all the Customer database is visible	1.Enter URL(http://169.51.204.215:30106) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box. 5.Click on login button	http://169.51.204.215:30106	Customer database is visible	Working as expected	PASS	Successful			JENITHA

Test Case (SPRINT 01) 5

LoginPage_TC_010	Functional	USER REGISTER	Verify 'Id sent to customer email address	1.Enter URL(http://169.51.204.215:30106) and click go 1.Register the account by giving credentials 2. Click on button Submit	http://169.51.204.215:30106	Email sent successfully	Working as expected	PASS	Successful			ATCHAYA
LoginPage_TC_011	Functional	AGENT REGISTER	Verify AGENT is able to login into application with Valid Credentials	1.Enter URL(http://169.51.204.215:30106) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 5342 password: Testing123	ID sent successfully	Application should show 'correct email or password' validation message.	PASS	Successful			PRIYADHARSHINI
LoginPage_TC_012	Functional	Login page for ADMIN	Verify User is able to log into application with InValid Credentials	1.Enter URL(http://169.51.204.215:30106) and click go 2.Click on My Account dropdown button 3.Enter InValid ID in ID text box 4.Enter Invalid password in password text box 5.Click on login button	ID: 1111 password: 5678	Application should show 'Incorrect ID or password validation message.	Working as expected	PASS	Successful			JENITHA
LoginPage_TC_013	UI	Home page for Agent	Verify user is able to see the agent home page when user finish on submitting Credentials	1. Enter URL (http://169.51.204.215:30106) and click go 2. To the Agent Login page and submit Your Credentials	ID: 1111 password: 5678	AGENT Home Page popup should display	Working as expected	PASS	Successful			PRIYADHARSHINI

Test Case (SPRINT 01) 6

LoginPage_TC_014	UI	Home page for USER	Verify user is able to see the User home page when user finish on submitting Credentials	1. Enter URL(http://169.51.204.215:30106) and click go 2. To the User Login page and submit Your Credentials	http://169.51.204.215:30106/	USER Home Page popup should display	Working as expected	PASS	Successful		ATCHAYA
LoginPage_TC_015	UI	Home page for ADMIN	Verify user is able to see the ADMIN home page when user finish on submitting Credentials	1. Enter URL(http://169.51.204.215:30106) and click go 2. To the User Login page and submit Your Credentials	http://169.51.204.215:30106/	ADMIN Home Page popup should display	Working as expected	PASS	Successful		PRIYADHARSHINI
LoginPage_TC_016	Functional	AGENT PAGE	On delete Button the user Credentials will be selected	1. Enter URL(http://169.51.204.215:30106) and click go 2. To the Admin Page and select an User Credentials	http://169.51.204.215:30106/	ADMIN Home Page popup should display	Working as expected	PASS	Successful		PRIYADHARSHINI KOWSALYA

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] 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	10	3	1	2	17
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2

External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	13	12	25	78

3. Test Case Analysis

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

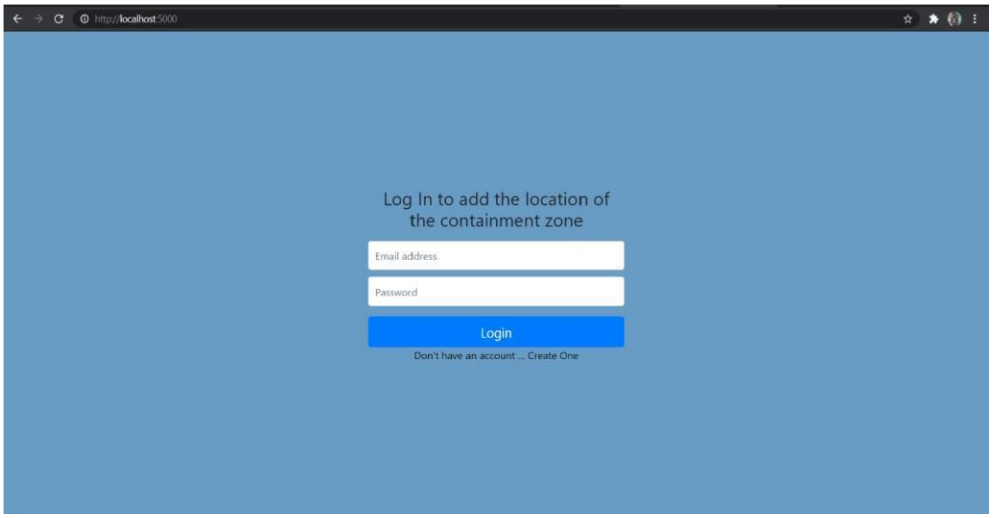
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	2	0	0	2

9. RESULTS:

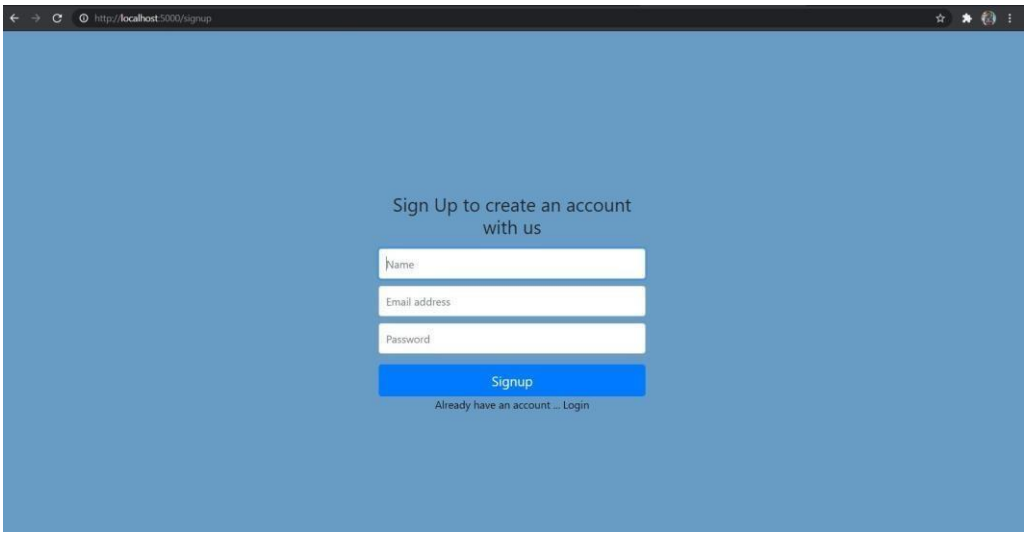
9.1 Performance Testing:

Admin App:

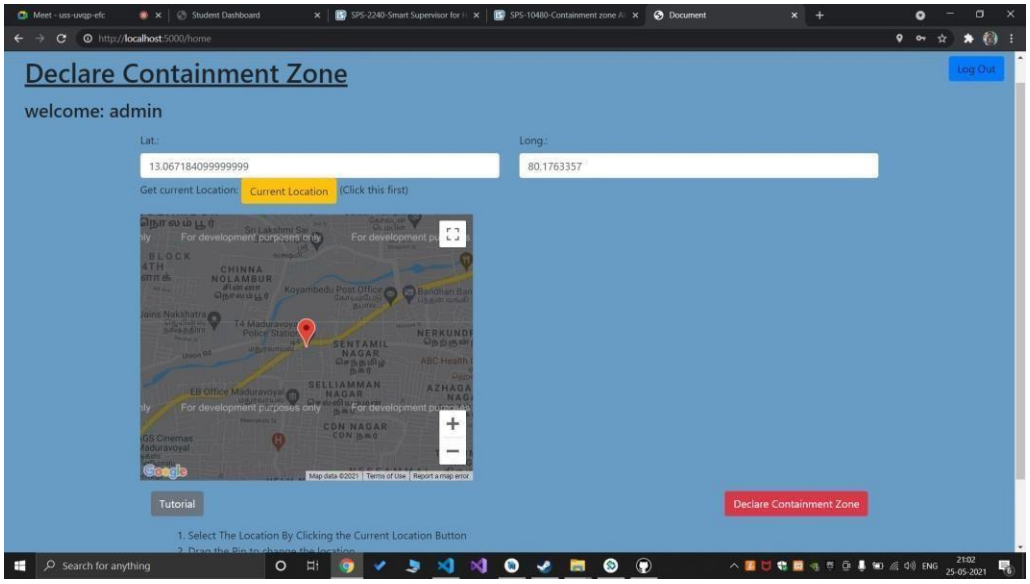
Login Page:



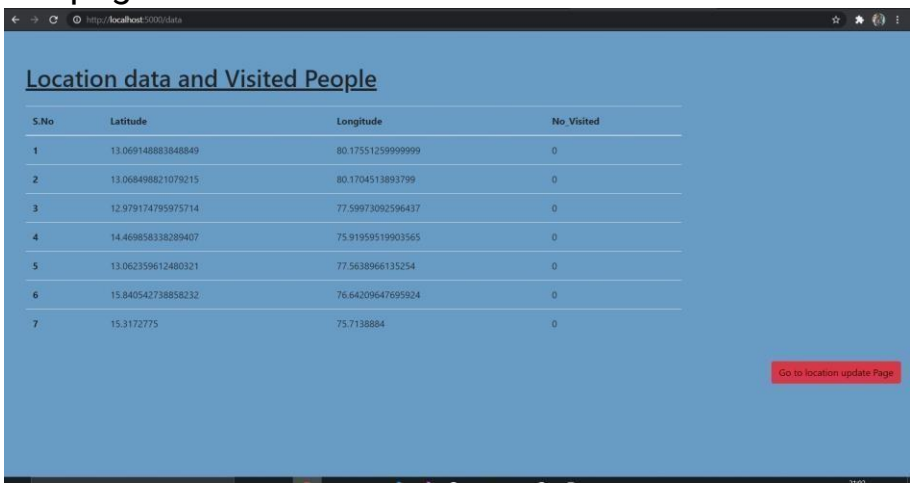
Register page:



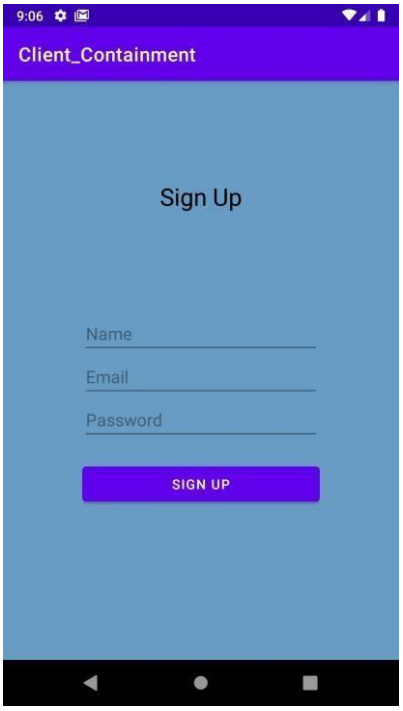
Home page:



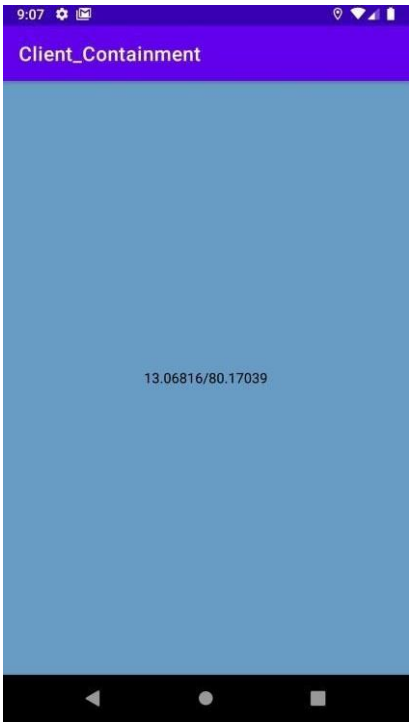
Location data page:



Client Application:
Register screen:



Current Location:



An Email will be sent to the registered mail id if the location is within 100 meters of the locations present in the admin app.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- People can be alerted before entering containment zone.
- Further spread of virus can be reduced considerably.

DISADVANTAGES:

- Accuracy of applicationdepends on the number of data given to the application.
- Application's accuracy is directly proportional to the number of data given to the application
- about the infected patients.

11. CONCLUSION

This application is intended to provide information about containment zones in a particular

region by alerting people, through continuous monitoring of an individuals location. Key benefits of

the application are monitoring peoples activity and alerting them to their safety movements.

12. FUTURE SCOPE

Although we tried to cover almost all of the aspects during our developmental phase, however we were forced to leave some aspects because of lack of time as well as monetary and other reasons. Just like in the field of software development where there are always some shortcomings and room for improvement our application can be enhanced further:-

1) The application can include various government organization to help act faster.

2) The dataset obtained from the application can be used for predictive analysis to determine prone areas and include special method for tackling the problem in those areas.

3) Emergency signal in case of network failure and internet connection loss.

4) Tackling victim's movements.

5) Improved Google positioning system's precision.

6) The client part of application can be integrated in a single intelligent device.

For analysis purpose, we could use machine learning (ML) algorithms as well as data mining applications. There is a sub branch of machine learning known as time series analysis (TSA), which could be used to predict and analyze the data obtained through this application. Time series analysis is used to predict crop production as well as sales in different quarter.

13 APPENDIX

Source Code

```
# Project : CONTAINMENT ZONE ALERTING APPLICATION
# Team ID : PNT2022TMID14431

APP.PY
from logging import error from flask import *
from jinja2.utils import select_autoescape import bcrypt
from flask_mysqldb import MySQL
```

```
import json
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
# initialization
app = Flask(_name_)
 # config
app.secret_key =
"\x19Ts\xbe\xe7\x8c_\r\x12Q\x14\x13>q\xb7'WTH0\x9f\xe4\xec\xb1"
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'zone2'
mysql = MySQL(app)
 # functions
 def send_mail(email):
    print(email)
    message = Mail(from_email='varundutia.h@gmail.com',
to_emails=email,
subject='caution',
  plain_text_content='Please Stay Safe',
html_content='<h2>You are entering into a containment Zone</h2>')
     try:
        sg = SendGridAPIClient(
'SG.7BJDtQDlS8unH0r5_TufVQ.Ykpcz19QcqgcNwYZC3a0mNRPhGksG117YURqOTa
2HL')        response = sg.send(message)
print(response.status.code)
print(response.body)
print(response.headers)
except Exception as e:
print(e)
def create_bcrypt_hash(password):     # convert the string to bytes
    password_bytes = password.encode()
    # generate a salt
```

```
    salt = bcrypt.gensalt(14)     # calculate a hash as bytes
    password_hash_bytes = bcrypt.hashpw(password_bytes, salt)
    # decode bytes to a string
    password_hash_str = password_hash_bytes.decode()     return password_hash_str




def verify_password(password, hash_from_database):
    password_bytes = password.encode()     hash_bytes = hash_from_database.encode()


    # this will automatically retrieve the salt from the hash,
    # then combine it with the password (parameter 1) # and then hash that, and compare it to the user's hash does_match = bcrypt.checkpw(password_bytes, hash_bytes)


    return does_match


# Api's


@app.route("/", methods=["GET", "POST"]) def login():     if(request.method == "POST"):


#         get the data from the form                                  password = request.form['password']                                  email = request.form['email']
```

```
    # initialize the cursor
    signup_cursor = mysql.connection.cursor()


    # check whether user already exists        user_result = signup_cursor.execute(
      "SELECT * FROM USERS WHERE user_email=%s", [email]
    )


    if(user_result > 0):
        data = signup_cursor.fetchone()  data_password = data[3]  if(verify_password(password, data_password)):
            signup_cursor.close()  session['id'] = data[0]  session['name'] = data[1]  session['email'] = data[2]  return redirect(url_for("home"))  else:
            return render_template('login.html', error=1)  else:
        return render_template('login.html', error=2)  return render_template('login.html', error=3)


@app.route("/signup", methods=["POST", "GET"])
```

```
def verify_password(password, hash_from_database):
password_bytes = password.encode()
hash_bytes = hash_from_database.encode()
# this will automatically retrieve the salt from the hash,
# then combine it with the password (parameter 1)
# and then hash that, and compare it to the user's hash
does_match = bcrypt.checkpw(password_bytes, hash_bytes)
return does_match
# Api's
@app.route("/", methods=["GET", "POST"])
def login():
if(request.method == "POST"):
# get the data from the form
password = request.form['password']
email = request.form['email']
# initialize the cursor
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS WHERE user_email=%s", [email]
)
if(user_result > 0):
data = signup_cursor.fetchone()
data_password = data[3]
if(verify_password(password, data_password)):
signup_cursor.close()
session['id'] = data[0]
session['name'] = data[1]
session['email'] = data[2]
return redirect(url_for("home"))
else:
return render_template('login.html', error=1)
else:
return render_template('login.html', error=2)
return render_template('login.html', error=3)
@app.route("/signup", methods=["POST", "GET"])

def create_bcrypt_hash(password):
# convert the string to bytes
password_bytes = password.encode()
# generate a salt
salt = bcrypt.gensalt(14)
# calculate a hash as bytes
password_hash_bytes = bcrypt.hashpw(password_bytes, salt)
# decode bytes to a string
password_hash_str = password_hash_bytes.decode()
return password_hash_str
```

PNT2022TMID14431

```
def verify_password(password, hash_from_database):
password_bytes = password.encode()
hash_bytes = hash_from_database.encode()
# this will automatically retrieve the salt from the hash,
```

```
# then combine it with the password (parameter 1)
# and then hash that, and compare it to the user's hash
does_match = bcrypt.checkpw(password_bytes, hash_bytes)
return does_match
# Api's
@app.route("/", methods=["GET", "POST"])
def login():
if(request.method == "POST"):
# get the data from the form
password = request.form['password']
email = request.form['email']
# initialize the cursor
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS WHERE user_email=%s", [email]
)
if(user_result > 0):
data = signup_cursor.fetchone()
data_password = data[3]
if(verify_password(password, data_password)):
signup_cursor.close()
session['id'] = data[0]
session['name'] = data[1]
session['email'] = data[2]
return redirect(url_for("home"))
else:
return render_template('login.html', error=1)
else:
return render_template('login.html', error=2)
return render_template('login.html', error=3)
@app.route("/signup", methods=["POST", "GET"])
def signup():
if(request.method == "POST"):

    # get the data from the form
name = request.form['name']
email = request.form['email']
    password = request.form['password']

    # hash the password
    pw_hash = create_bcrypt_hash(password)

    # initialize the cursor
    signup_cursor = mysql.connection.cursor()

    # check whether user already exists
user_result = signup_cursor.execute(
        "SELECT * FROM USERS WHERE user_email=%s", [email]
    )
    if(user_result > 0):
        signup_cursor.close()
```

```
        return render_template('signup.html', error=True)
else:
        # execute the query
signup_cursor.execute(
          'INSERT        INTO        USERS(user_name,user_email,user_password,user_type)
VALUES(%s,%s,%s,%s)', (
             name, email, str(pw_hash), "2"
          )
        )

        mysql.connection.commit()
signup_cursor.close()
        return redirect(url_for('login'))

   return render_template('signup.html', error=False)

@app.route("/home", methods=["POST", "GET"])
def home():
if(session['id'] == None):
      return redirect(url_for('login'))
def upload():
if(request.method == "POST"):
# get the data from the form
name = request.json['name']
email = request.json['email']
password = request.json['password']
# hash the password
pw_hash = create_bcrypt_hash(password)
# initialize the cursor
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS WHERE user_email=%s", [email]
)
if(user_result > 0):
signup_cursor.close()
return {'status': 'failure'}
else:
# execute the query
signup_cursor.execute(
'INSERT INTO USERS(user_name,user_email,user_password,user_type)
VALUES(%s,%s,%s,%s)', (
name, email, str(pw_hash), "1"
)
)
mysql.connection.commit()
id_result = signup_cursor.execute(
'SELECT user_id FROM USERS WHERE user_email = %s', [email]
)
if(id_result > 0):
id = signup_cursor.fetchone()
return {"id": id[0]}
```

```
signup_cursor.close()
return {"status": "failure"}
@app.route("/get_all_users")
def getusers():
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS"

   if(request.method == "POST"):
      # get data
      lat = request.form["lat"]
      lon = request.form["lon"]
vis = 0
  if(lat == "" or lon == ""):
return render_template('home.html', name=session['name'], email=session['email'],
id=session['id'], success=0)
# create a location cursor
location_cursor = mysql.connection.cursor()
# Execute the query
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', name=session['name'], email=session['email'],
id=session['id'], success=True)
return render_template('home.html', name=session['name'], email=session['email'],
id=session['id'])

@app.route("/logout")
def logout():
# remove the username from the session if it is there
session['id'] = None
session['name'] = None
session['email'] =  None
return redirect(url_for('login'))
@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)
@app.route("/android_sign_up", methods=["POST"])
def upload():
if(request.method == "POST"):
# get the data from the form
name = request.json['name']
email = request.json['email']
password = request.json['password']
# hash the password
pw_hash = create_bcrypt_hash(password)
# initialize the cursor
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS WHERE user_email=%s", [email]
)
if(user_result > 0):
signup_cursor.close()
return {'status': 'failure'}
else:
# execute the query
signup_cursor.execute(
'INSERT INTO USERS(user_name,user_email,user_password,user_type)
VALUES(%s,%s,%s,%s)', (
name, email, str(pw_hash), "1"
)
)
mysql.connection.commit()
id_result = signup_cursor.execute(
'SELECT user_id FROM USERS WHERE user_email = %s', [email]
)
if(id_result > 0):
id = signup_cursor.fetchone()
return {"id": id[0]}
signup_cursor.close()
return {"status": "failure"}
@app.route("/get_all_users")
def getusers():
signup_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = signup_cursor.execute(
"SELECT * FROM USERS"
PNT2022TMID14431
)
if(user_result > 0):
rv = signup_cursor.fetchall()
row_headers = [x[0] for x in signup_cursor.description]
json_data = []
for result in rv:
json_data.append(dict(zip(row_headers, result)))
return json.dumps(json_data)
```

```
@app.route("/post_user_location_data", methods=["POST"])
def post_user_location():
if(request.method == "POST"):
# get the data from the form
lat = request.json['lat']
lon = request.json['long']
id = request.json['id']
ts = request.json['timestamp']
# initialize the cursor
user_location_cursor = mysql.connection.cursor()
# execute the query
user_location_cursor.execute(
'INSERT INTO USER_LOCATION(location_lat,location_long,user_id,timestamp)
VALUES(%s,%s,%s,%s)', (
lat, lon, id, ts
)
)
mysql.connection.commit()
return {"response": "success"}
@app.route("/location_data")
def location_data():
location_cursor = mysql.connection.cursor()
# check whether user already exists
user_result = location_cursor.execute(
"SELECT * FROM LOCATION"
)
if(user_result != 0):
res = location_cursor.fetchall()
print(res)
row_headers = [x[0] for x in location_cursor.description]
json_data = []
PNT2022TMID14431
for result in res:
json_data.append(dict(zip(row_headers, result)))
return json.dumps(json_data)
else:
return {"response": "failure"}
@app.route("/send_trigger", methods=["POST"])
def send_trigger():
if(request.method == "POST"):
# get the data from the form
email = request.json['email']
location_id = request.json['id']
location_cursor = mysql.connection.cursor()
# check whether user already exists
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"}
# main
if_name_== "_main_":
app.run(host='0.0.0.0', port=5000)
```

DATA.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Zones</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous" />
<style>
body {
padding-top: 30px;
padding-bottom: 30px;
background-color: #699cc5;
}
a {
color: black;
}
</style>
</head>
<body>
<div class="m-4 container">
<h1><u>Location data and Visited People</u></h1>
</div>
<div class="m-4 container">
<table class="table">
<thead>
<tr>
<th scope="col">S.No</th>
<th scope="col">Latitude</th>
<th scope="col">Longitude</th>
<th scope="col">No_Visited</th>
```

```
</tr>
</thead>
<tbody>
{%- for row in responses %}
<tr>
<th scope="row">{{loop.index}}</th>
<td>{{row[1]}}</td>
<td>{{row[2]}}</td>
<td>{{row[3]}}</td>
</tr>
{%- endfor %}
</tbody>
</table>
</div>
<div class="m-3 float-right">


<button type="button" class="btn btn-danger"><a href={{url_for("home")}}>Go to location
update Page</a></button>
   </div>

</body>

</html>
```

HOME.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
<link     rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
                                              integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous" />
   <style>        body {
          padding-top: 30px;
padding-bottom: 30px;
          background-color: #699cc5;
       }

       a {
          color: black;
       }
   </style> </head>

<body>
```

```
    {% if success == True %}
    <script>
        alert("Location Uploaded Successfully");
    </script>
    {% elif success == 0 %}
    <script>
        alert("Enter Proper Location data");
    </script>
    {% endif %}
    <div class="m-3 float-right">
        <button type="button" class="btn btn-primary"><a href={{url_for("logout")}}>Log
Out</a></button>
    </div>
    <div class="container m-3">
        <h1><u>Declare Containment Zone</u></h1>
    </div>
    <div class="container m-3">
        <h3>welcome: {{name}}</h3>
    </div>
    <form method="POST" action="/home">
        <div class="container">
            <div class="form-group row">
                <div class="col-sm-6">
                    <label class="control-label">Lat.:</label>
                    <input type="text" class="form-control" id="lat" name="lat" />
                </div>
                <div class="col-sm-6">
                    <label>Long.:</label>
                    <input type="text" class="form-control" id="lon" name="lon" />
                </div>
                <div class="col-sm-6">
                    <label>Get current Location:</label>
                    <button type="button" class="btn btn-warning" onclick="getLocation()">Current
Location</button>
                    <label>(Click this first)</label>
                </div>
            </div>

            <!-- map -->
            <div id="map_disp" style="height: 400px;width: 500px;"></div>
            <div class="m-3 float-right">
                <button type="submit" class="btn btn-danger">Declare Containment Zone</button>
            </div>
            <div class="m-3">
            <button onclick="toggleTips()" type="button" class="btn
btnsecondary">Tutorial</button>
                <div id="tips" class="m-3">
                    <ol>
                        <li>Select The Location By Clicking the Current Location Button</li>
                        <li>Drag the Pin to change the location</li>
                        <li>Click on Declare Containment Zone to save the location to the database </li>
                    </ol>
```

```
            </div>
          </div>
          <div class="m-3 float-right">
            <button type="button" class="btn btn-warning"><a href="{{url_for('data')}}">Click
Here To View The
                    Containment Zones and Number of


                    people visited</a>
</button>
          </div>
        </div>

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js"
                                                                integrity="sha384-
+YQ4JLhjyBLPDQt//I+STsc9iw4uQqACwlvpslubQzn4u2UU2UFM80nGisd026JF"          cros
sorigin="anonymous">
</script>
        <script src="https://code.jquery.com/jquery-2.2.4.min.js">
</script>                                   <script
src="https://maps.google.com/maps/api/js?sensor=false&amp;libraries=places"></script>
        <script
                                                    src="https://rawgit.com/Logicify/jquery-
locationpickerplugin/master/dist/locationpicker.jquery.js"></script>

        <script>
          function getLocation()
{
if (navigator.geolocation)
{
                navigator.geolocation.getCurrentPosition(showPosition);
            } else {
                alert("No location");
            }

          }
          function showPosition(position)
{             $('#map_disp').locationpicker({
location:
{
                    latitude: position.coords.latitude,
longitude: position.coords.longitude
                },
                radius: 0,
inputBinding:
{
                    latitudeInput: $('#lat'),
                    longitudeInput: $('#lon'),
                },
                enableAutocomplete: true,
                onchanged: function (currentLocation, radius, isMarkerDropped)
{
```

```
                    // Uncomment line below to show alert on each Location Changed event
                    // alert("Location changed. New location (" + currentLocation.latitude + ", " +
currentLocation.longitude + ")");
                }
            });
        }
        function toggleTips() {
            var x = document.getElementById("tips");
if (x.style.display === "none") {
x.style.display = "block";
            } else {
                x.style.display = "none";
} }
</script>
</body>
</html>
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-16429-1659614398

Video Demo Link:

https://youtu.be/YlllqJ_1F84