Project Report

1. INTRODUCTION

1.1 Project Overview

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

1.2 Purpose

Provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individual's location. This Android application updates the locations of the areas in a Google map which are identified to be the containment zones. The application also notifies the users if they have entered a containment zone and uploads the user's info to the online database. To achieve all these functionalities, many tools and APIs from Google like Firebase and Geofence are used in this app. 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.

2. LITERATURE SURVEY

1. Social Distance Alert System to Control Virus Spread using UWB RTLS incorporateEnvironments

The author proposed a method to develop a real-time location system (RTLS) based on ultra-wideband (UWB) wireless technology that gives the most accurate locations of approximately 10cm using methods like trilateration and TDOA (Time Difference of Arrival).
Coordinates of the location can be obtained by installing RTLS in predefined areas which areused to calculate the distance between Mobile UWB Devices (MUD's). An alert triggered by asystem to maintaindistance if distance between the employees is less than the prescribed social distance can keep the work premises safe and control the spread of coronavirus. This study can be a great solution to control the spread of virus in corporate working environments which are mostly confined in size and indoor in nature.

2. A Detection, Tracking and Alerting System for Covid-19 using Geo-Fencing and MachineLearning

The author proposed a complete Covid-19 Detection, Tracking and Alerting Mobile Application Kit which helps people to defend against Covid-19 spread. This is a first of its kind application that uses Geofencing and Machine learning together to combat the spread of

Coronavirus. This app is a threefold app. The first fold is a Detection System for a user to undergo a Symptomatic Quiz based on a Risk Assessment ML Model to detect the presence of Covid in the user's body. The second fold is an efficient Tracking system that uses Geofencing technology to keep track of all the people who come into contact with the user. And the third fold is an Alerting system that sends the alertmessage to all those people who came into contact with the user if the user is tested as Corona positive. Thus, by using the technology, Geofencing allows to perform contact tracing of potential patients and alerts the possible network of people who might be infected by coronavirus.

3. **Android Application based Smart Bus Transportation System for Pandemic Situations**

Smart Bus Transportation System was introduced which guides the passengers in bookingthe bus tickets using the Android Application and it also helps the passengers to keep an updateon bus location based on their request. This system also sends alert message few minutes in advance to the passengers before the bus reaches thepassengers boarding point. This system also sends the precautionary instruction priory the passengers that have to be followed while traveling in the bus. In order to provide additional safety to the passengers the temperature of the passengers is monitored and intimated to the bus in change before they are permitted into the bus.

4. **Social Distancing Inspection To Mitigate COVID-19 Using K-Nearest Neighbor**

In this paper, a model is recommended where the total number of people presenting theframe is detected using the YOLO object detection algorithm, and distance between each individual is measured Using K-Nearest Neighbor. If the distance between any two individualsis less than 6 feet or 2 meters then a red bounding box pops around them indicating that they are violating the rule of social distancing. This model is implemented on Raspberry Pi with a buzzer system for alert.

5. **Social Distancing and Face Mask Monitoring System Using Deep Learning Based onCOVID-19 Directive Measures**

The author proposed a system consisting of data processing, data augmentation, image classification using mobilenetv2 and object detection. The modules are developed using TensorFlow and open-cv python programming to detect faces with masks. If a person wears a mask they will be in a safe zone and the system shows a green box where if the person doesn't wear a mask, then it will be shown in a red box and with the message of alert as well. Social distancing detection will detect that two or more person in a single frame are walking with maintaining social distancing with at least 2 meters of range with each other using the Euclidean distance method, it will work in a Reliable manner with accurate results during this current situation which will easily help to track the person and collect fine if they violate any government directive guidelines so our system, will prevent the spread of the disease. Every Automation process reduces manual inspection to inspect the people which can be used in public places to control the spread of the virus and this prototype could be used in

many places like park, hospital,airports, temples, railway station etc. to control this pandemic situation

6. **Application of Face Recognition in Tracing COVID-19 Fever Patients and CloseContacts**

The author developed a face recognition system to detect patients with fever symptomsand to trace close contacts. A real-time alert is sent to the account manager on a web ormobile app to enable further actions to quarantine the patients and close contacts. The RGB camera is used to detect a face and locate the forehead. The thermal image of the face is used to measure the temperature of the

skin in the forehead. A black body is optional to improve the temperature measurement accuracy. After a patient is confirmed, his identification can be recognized using face recognition. By face recognition clustering, all face images of this person in the past given period of time (e.g., 14 days) can be retrieved. Furthermore, close contacts of this patient can also be retrieved from saved frame images or the camera ID and time stamp. The work [2] proposed a similar idea of using face recognition to trace fever patients and close contacts butdid not give an algorithm on how to trace them. These retrieved results are displayed in an account console, and a notification is sent to the personnel (account manager) on duty in real time, and safety action can be taken to quarantine the persons, achieving the goals of stopping the virus spreading.

References

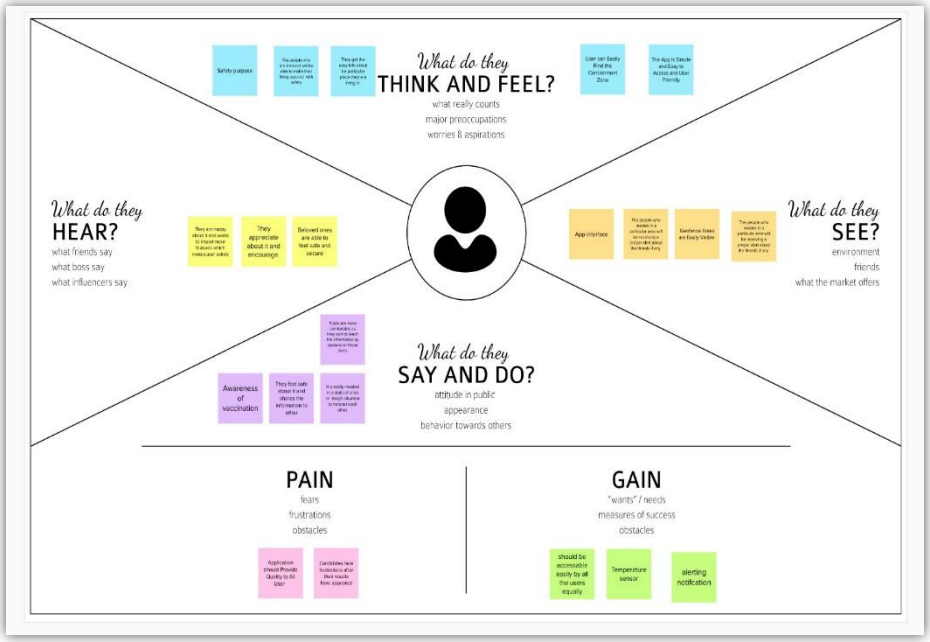
1. https://ieeexplore.ieee.org/document/9711880
2. https://ieeexplore.ieee.org/document/9432254
3. https://ieeexplore.ieee.org/document/9356316
4. https://ieeexplore.ieee.org/document/9388625
5. https://ieeexplore.ieee.org/document/9609407
6. https://ieeexplore.ieee.org/document/9356316

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map

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.

Containment Zone Alerting Application:

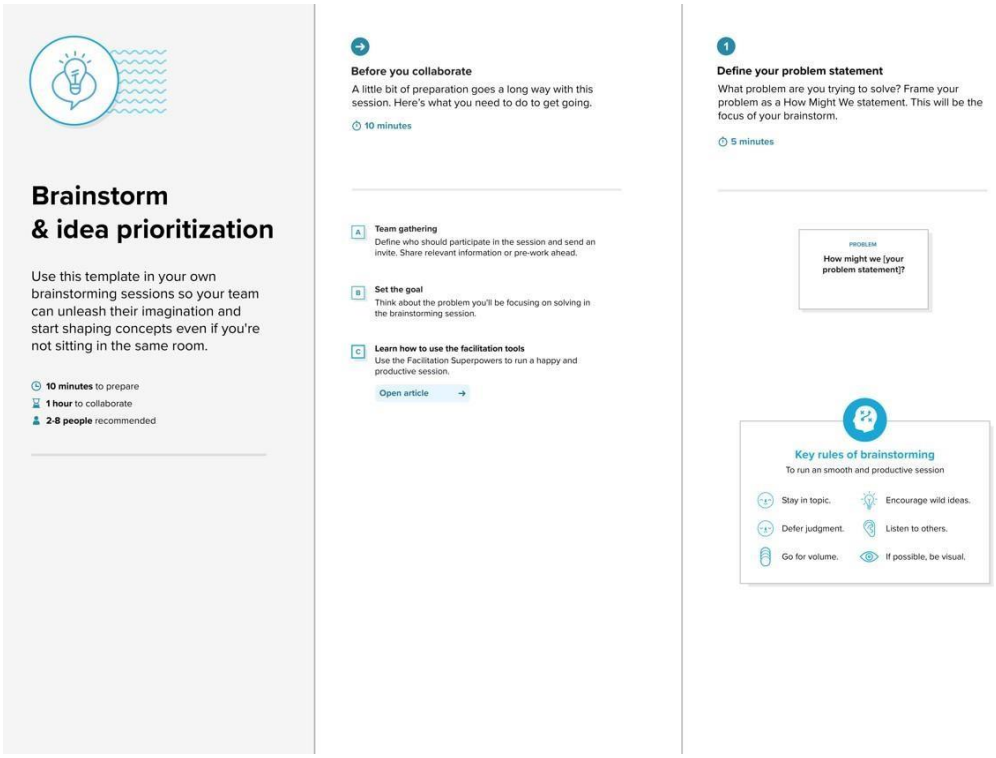


3.2 Ideation & Brainstorming

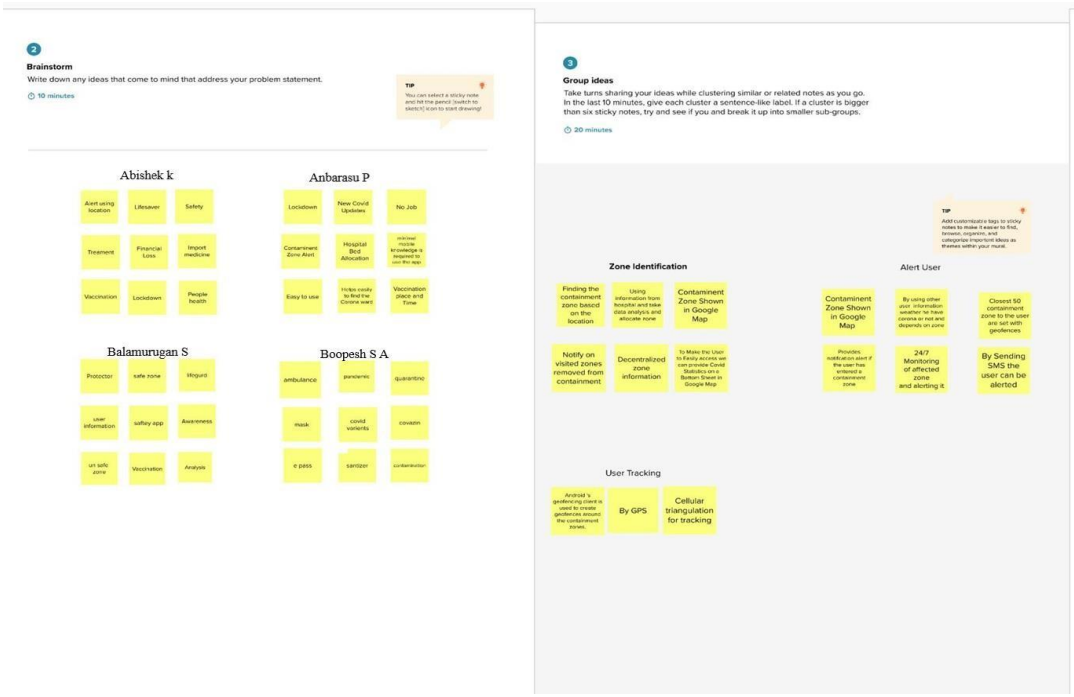
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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.

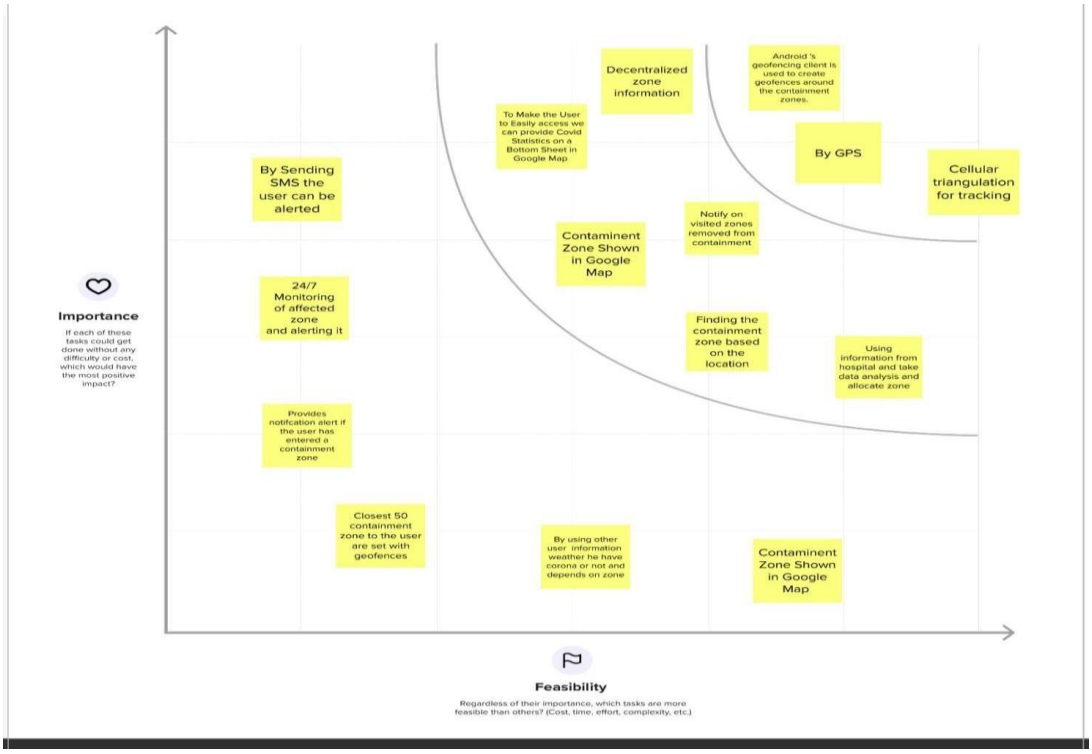
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Provide information about the covid containment zones and alerting the users by notifying the them while they were near the affected area.
2.	Idea / Solution description	To create an application to alert the particular user in that specific region from entering a Containment Zone by sending notification. To provide accurate results and alerting the users with in time before entering the zone.
3.	Novelty / Uniqueness	• Informing people about the Covid containment zones and prevent from entering into the zones. • Notify the users if they have entered a containment zone and warn them
4.	Social Impact / Customer Satisfaction	Lives can be saved by alerting the users from entering the Containment zone. Stops the disease from spreading to the people by sending notification to the users.
5.	Business Model (Revenue Model)	The program can be handled by the government to prevent their citizens from approaching the containment zones.
6.	Scalability of the Solution	Easy user interaction with the application. Collected data is stored in database for future use.

3.4 Proposed Solution Fit

Project Title: Containment Zone Alerting Application **Project Design Phase-I – Solution Fit Template** **Team ID:** PNT2022TMID02859

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Any customers who is in urgent to travel find this application useful which alert and notify them the contaminated zone area location near them before entering the zone.

6. CUSTOMER CONSTRAINTS CC

Very easy to use the application for the users who don't know about it with making few tries.

5. AVAILABLE SOLUTION AS

- Notification messages send to the individual
- Identifying the zones by tracking individual's location.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- To identify the locations of the containment zones
- Detect and notify when users enter any containment zones.

9. PROBLEM ROOT CAUSE RC

- Unaware of the contaminated zone and entering.
- Possibility of getting infected when entering the containment zone.

7. BEHAVIOUR BE

- Help taking precautionary measures when they enter the zone by accident.
- Geofence the containment zone areas near the user's location.
- Shows the current cases in the area.

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

Identify strong TR & EM

3. TRIGGERS TR

Alert the users when they enter the containment zone and gives warning not enter the zone.

4. EMOTIONS: BEFORE / AFTER EM

Before: The user doesn't know or will be unaware of the containment zone and the situation.

After: The user will get alerted when they approach the containment zone and make them safe.

10. YOUR SOLUTION SL

- The application will notify the users if they are about to enter the containment zones.
- Record information about the affected people, recovered people and number of death cases which helps the users to know about the current situation

8. CHANNEL OF BEHAVIOUR CH

Customers can identify the containment zone areas before entering and thus stop from spreading. Stores the collected data in database for future use.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

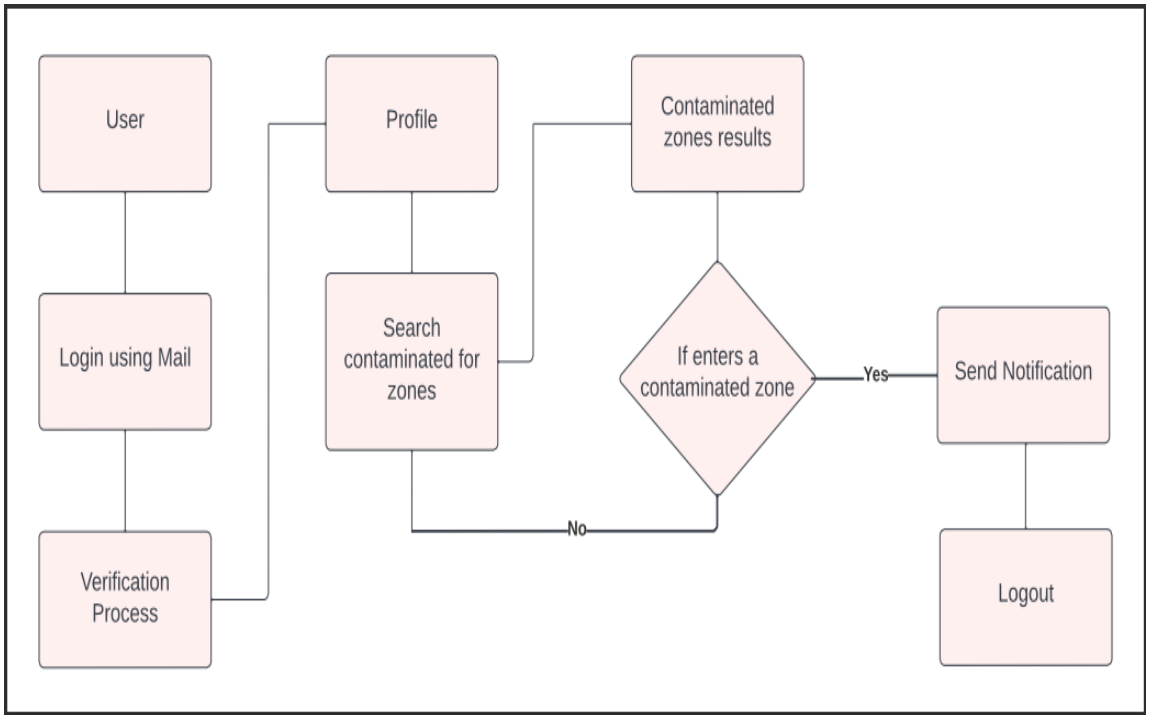
FR NO.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR – 1	User registration	Registration through Form Registration through Gmail
FR – 2	User confirmation	Confirmation via Mail Confirmation via OTP
FR – 3	Alert Message via Notification	Location tracking and notification services
FR – 4	Show infected zones	Geofencing functionality

4.2 Non-Functional Requirements

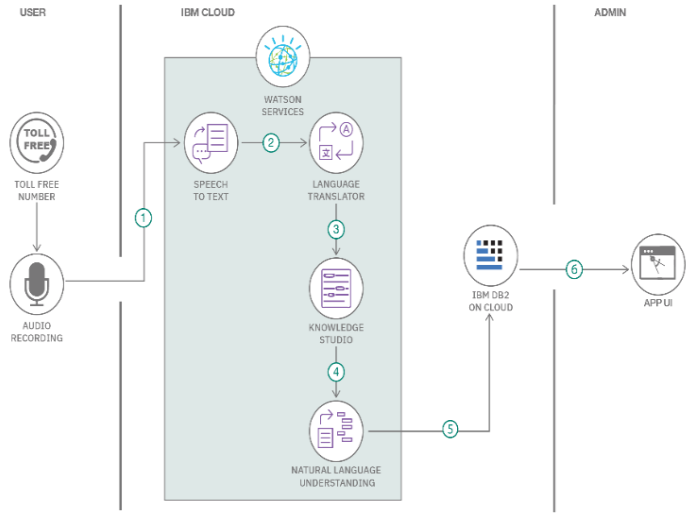
FR NO.	Non-Functional Requirement	Description
FR – 1	**Usability**	Easy interaction with Graphical User Interface
FR – 2	**Security**	The data which is collected will be stored in the database securely
FR – 3	**Reliability**	User can trust the accuracy of the service provided
FR – 4	**Performance**	Fast and maximum accuracy provided to the users using real time location
FR – 5	**Availability**	Available if the user allows GPS tracking and with the good network range
FR – 6	**Scalability**	User can interact with both the website and mobile app

5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution & 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 Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can access my dashboard after signing in.	I can access my account / dashboard	High	Sprint-2
Customer Care Executive	Geofencing	USN-6	As a user, I can see containment zones from the maps by tracking my current location.		High	Sprint-2
Administrator	DBMS	USN-7	As a administrator, I can keep the applications updated with containment zone details and regular covid related news.	I can perform various modifications in the applications according to user feedback.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information byreferring the technical papers, research publication.	4 SEPTEMBER 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	11 SEPTEMBER 2022
Ideation	Prioritizing the top 3 ideas based on feasibility and Importance	18 SEPTEMBER 2022
Proposed Solution	Solution include novelty, feasibility, business model,social impact and scalability of solution	25 SEPTEMBER 2022
Problem Solution Fit	Solution fit document	2 OCTOBER 2022
Solution Architecture	Solution Architecture	8 OCTOBER 2022
Customer Journey	To Understand User Interactio nsand experiences with application	9 OCTOBER 2022

Functional Requirement	Prepare functional Requirement	16 OCTOBER 2022
Data flow Diagrams	Data flow diagram	16 OCTOBER 2022
Technology Architecture	Technology Architecturediagram	23 OCTOBER 2022
Milestone & sprint deliveryplan	Activity what we done&further plans	29 OCTOBER 2022
Project Development Deliveryof sprint 1,2,3 & 4	Develop and submit the developed code by testing it	29 OCTOBER 2022 – 19NOVEMBERS 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	**Registration** (web and android)	USN-1	USER: I can register for the application by entering myemail and password	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A
		USN-2	USER: I will receive a confirmation email once I haveregistered for the application	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A
	Login (web and android)	USN-3	USER: I can log into the application by entering my credentials	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Dashboard	USN-4	USER: need to give permission to access my location	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A
		USN-5	As a user, I can log into the application by entering email & password	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 3	**Service**	USN 6	ADMIN: I need to update the containment zones.	5	High	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A Boopesh S A
		USN 7	ADMIN: I need to differentiate the containment zonesbased on the intensity of infection.	3	Medium	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A

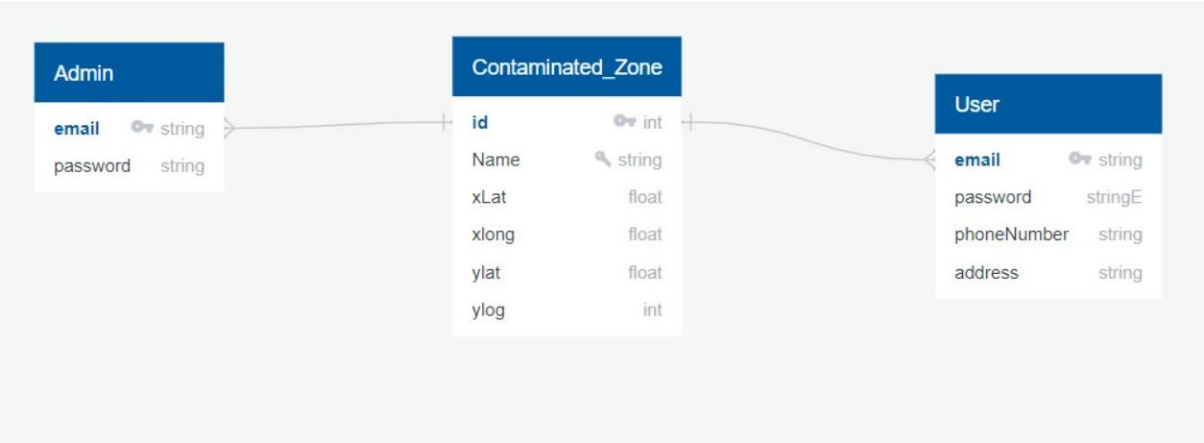
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 4	Service	USN 8	ADMIN: I need to alert the user when they enter thecontainment zone through the notification	3	Medium	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A
	Data collection	USN 9	ADMIN: I need to store user details on the cloud	3	Medium	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A
		USN 10	ADMIN: I need to collect details about covid -19 cases from verified sources	3	Medium	Abishek K, Anbarasu P, Balamurugan s, Boopesh S A

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Features

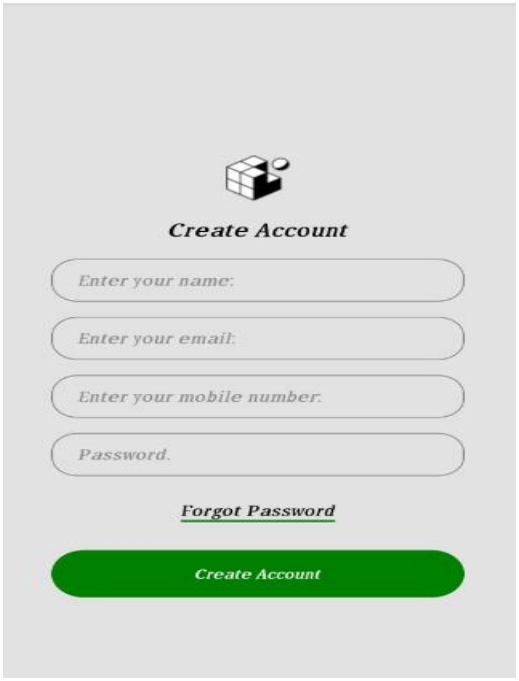
The users get alerted from entering the contaminated zone by geofencing the location and sending it as notification.

7.2 Database Schema

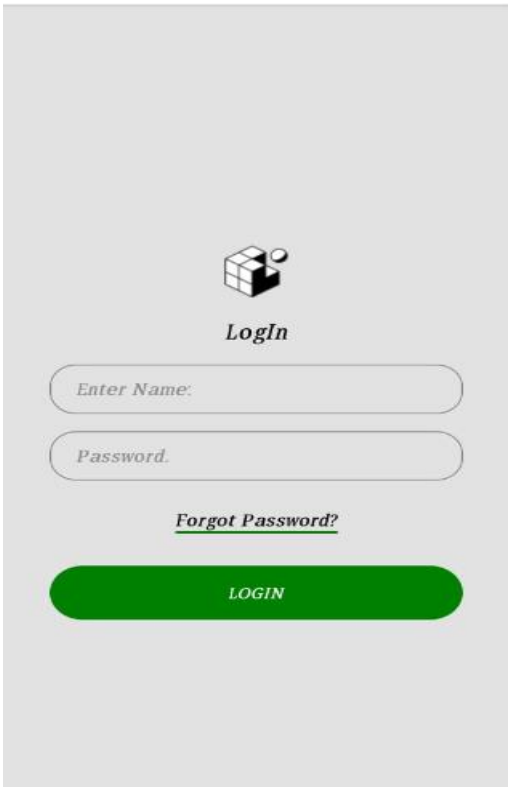


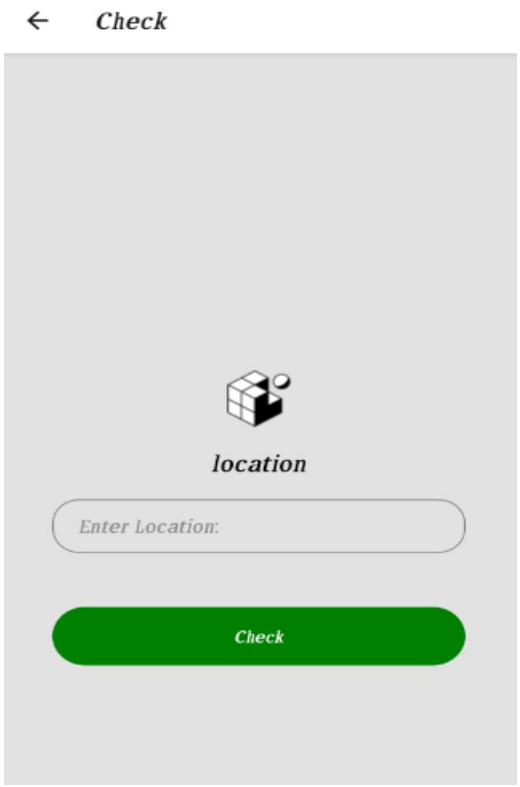
8. TESTING

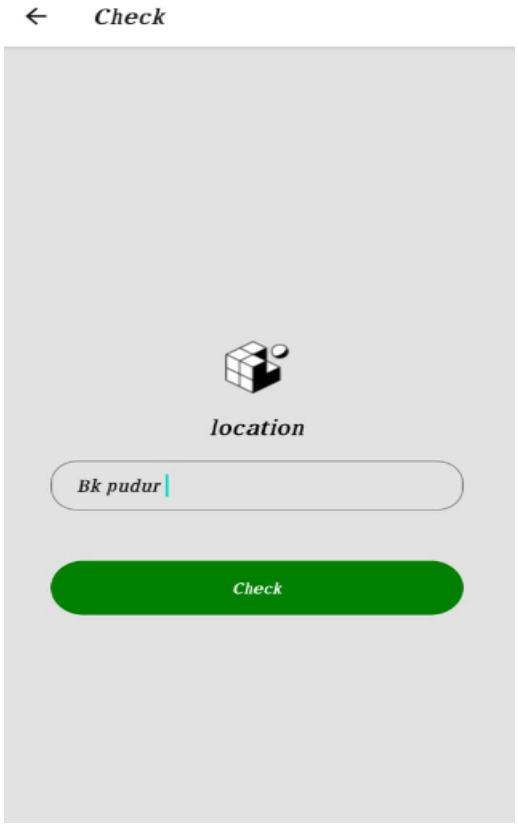
login



← *One*




Check
location
Enter Location:
Check


Check
location
Bk pudur
Check

9. RESULTS

This app service monitors the location and provide information about the contaminated zones near a particular user and send notification to the user. It displays the contaminated zone area by geofencing the particular location.

10. ADVANTAGES & DISADVANTAGES

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.

11. CONCLUSION

We proposed a framework for identifying the contaminated zone areas and store it in database for future use. Then using the database, information is provided to the user about contaminated zone areas and alerting them by sending notification and geofencing the location.

12. FUTURE SCOPE

The application provides an efficient way of showing the identified COVID-19 containment zones to the users in a Google map. With the alarming increase of COVID-19 affected cases throughout the world, this developed application can be employed as a tool for creating further social awareness among the people. This application further tracks the user's location and checks whether it is present in the list of identified containment zones. It sends separate notification alerts to the user on entering and exiting the containment areas. The developed android application further extracts the IMEI Number of the trespasser in the containment zones which can be useful to the local police to track and identify people who are frequently trespassing the containment zones. Thereby this application identifies the containment zones and highlights the need for taking further precautionary measures for combating COVID-19. The application has been tested in various locations and has been found to yield accurate results. The application can be further used for many purposes like maritime and forest safety to prevent users from entering restricted areas.

13. APPENDIX

App.js:

```
import * as React from 'react';
import { Button, View, Text } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import Login from "./Login";
```

```
import One from "./One";
import Check from "./check"

const Stack = createNativeStackNavigator();

function App() {
  return (
      <NavigationContainer>
      <Stack.Navigator initialRouteName="login">
        <Stack.Screen name="login" component={Login} />
        <Stack.Screen name="One" component={One} />
        <Stack.Screen name="Check" component={Check} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

One.js

```
import { StatusBar } from 'expo-status-bar';
import {
  StyleSheet,
  Text,
  TextInput,
  View,
  Image,
  TouchableOpacity
} from 'react-native';

export default function One({ navigation }) {
  return (
    <View style={styles.container}>
    <Image style={styles.image} source={require("./assets/favicon.png")} />

      <Text style={styles.font}>LogIn</Text>
      <TextInput style={styles.name} placeholder='Enter Name:'></TextInput>
      <TextInput style={styles.name} placeholder="Password."
secureTextEntry={true}/>
        <TouchableOpacity>
          <Text style={styles.forgot_button}>Forgot Password?</Text>
        </TouchableOpacity>

        <TouchableOpacity style={styles.loginBtn}
onPress={()=>navigation.navigate("Check")}>
          <Text style={styles.loginText}>LOGIN</Text>
```

```
        </TouchableOpacity>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  font:{
    margin:18,
    fontSize:20,
  },
  container: {
    flex: 1,
    backgroundColor: '#e2e2e2',
    alignItems: 'center',
    justifyContent: 'center',
  },
  name:{
    fontSize:16,
    marginBottom:16,
    padding:8,
    paddingLeft:20,
    borderWidth:1,
    borderColor:"gray",
    width:"80%",
    borderRadius:20,
  },

  forgot_button:{
    marginTop:10,
    fontSize:16,
    borderBottomWidth:2,
    borderColor:"green",
  },
  loginBtn:{
    width: "80%",
    borderRadius: 25,
    height: 50,
    alignItems: "center",
    justifyContent: "center",
    marginTop: 30,
    backgroundColor: "green",
  },
  loginText:{
    color:"white",
  },
});
```

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, TouchableOpacity, Image, TextInput } from
'react-native';



export default function Login({navigation}) {
  return (
      <View style={styles.container}>
    <Image style={styles.image} source={require("./assets/favicon.png")} />

      <Text style={styles.font}>Create Account</Text>
      <TextInput style={styles.name} placeholder='Enter your
name:'></TextInput>
      <TextInput style={styles.name} placeholder='Enter your
email:'></TextInput>
      <TextInput style={styles.name} placeholder='Enter your mobile
number:'></TextInput>
      <TextInput style={styles.name} placeholder="Password."
secureTextEntry={true}/>
        <TouchableOpacity>
          <Text style={styles.forgot_button}>Forgot Password</Text>
        </TouchableOpacity>

        <TouchableOpacity style={styles.loginBtn} onPress={
          ()=>navigation.navigate("One")
        }>
          <Text style={styles.loginText}>Create Account</Text>
        </TouchableOpacity>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  font:{
    margin:18,
    fontSize:20,
  },
  container: {
    flex: 1,
    backgroundColor: '#e2e2e2',
```

```
    alignItems: 'center',
    justifyContent: 'center',
  },
  name:{
    fontSize:16,
    marginBottom:16,
    padding:8,
    paddingLeft:20,
    borderWidth:1,
    borderColor:"gray",
    width:"80%",
    borderRadius:20,
  },

  forgot_button:{
    marginTop:10,
    fontSize:16,
    borderBottomWidth:2,
    borderColor:"green",
  },
  loginBtn:{
    width: "80%",
    borderRadius: 25,
    height: 50,
    alignItems: "center",
    justifyContent: "center",
    marginTop: 30,
    backgroundColor: "green",
  },
  loginText:{
    color:"white",
  },
});

import { StatusBar } from 'expo-status-bar';
import {
  StyleSheet,
  Text,
  TextInput,
  View,
  Image,
  TouchableOpacity
} from 'react-native';

export default function Check({ navigation }) {
  return (
    <View style={styles.container}>
    <Image style={styles.image} source={require("./assets/favicon.png")} />
```

```
      <Text style={styles.font}>location</Text>
      <TextInput style={styles.name} placeholder='Enter
Location:'></TextInput>

        <TouchableOpacity style={styles.loginBtn}>
          <Text style={styles.loginText}>Check</Text>
        </TouchableOpacity>
      <StatusBar style="auto" />
    </View>
  );
}
const styles = StyleSheet.create({
    font:{
      margin:18,
      fontSize:20,
    },
    container: {
      flex: 1,
      backgroundColor: '#e2e2e2',
      alignItems: 'center',
      justifyContent: 'center',
    },
    name:{
      fontSize:16,
      marginBottom:16,
      padding:8,
      paddingLeft:20,
      borderWidth:1,
      borderColor:"gray",
      width:"80%",
      borderRadius:20,
    },

    forgot_button:{
      marginTop:10,
      fontSize:16,
      borderBottomWidth:2,
      borderColor:"green",
    },
    loginBtn:{
      width: "80%",
      borderRadius: 25,
      height: 50,
      alignItems: "center",
      justifyContent: "center",
      marginTop: 30,
      backgroundColor: "green",
```

```
  },
  loginText:{
    color:"white",
  },
});
```