Project Report

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule

c. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)
8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing
9. **RESULTS**
    a. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1. INTRODUCTION

a. Project Overview

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 using mobile app. Key benefits of the application are monitoring people's activity and alerting them of their safety movements. The officials can monitor the user’s activity from their admin panel.

b. Purpose

Admin Panel – Website

User App – Mobile App

Admin App (portal):

1. They can login to the site and update the containment zones locations in the portal.
2. Based on the location a Geofence will be created within a 100 meters radius.
3. They can be able to see how many people are visiting that zone.

User App (Mobile App):

1. The app will have a user registration and login.
2. After the user logged into the app it will track the user's location and update the database with the current location.
3. If the user is visiting the containment zone he/she will get an alert notification.

2. LITERATURE SURVEY

a. Existing problem

1. The offcials can monitor the user’s activity from their Admin panel.
2. Location data of individuals will be stored in the database (IBM DB2) User privacy may be affected by tracking their location
3. No proper alerts received by users in the existing applications
4. Hence they can easily enter into the containment zone Zones are added by admin only.
5. It may cause human error Network unstability lead to huge cause

b. References

1. Coronavirus disease 2019 (COVID-19): situation report – 78. World Health Organization. 2020. Apr 07, [2020-07-01]. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200407-sitrep-78-covid19.pdf.

2. Mahmood S, Hasan K, Colder Carras M, Labrique A. Global preparedness against COVID-19: we must leverage the power of digital health. JMIR Public Health Surveill. 2020 Apr 16;6(2):e18980. doi: 10.2196/18980. https://publichealth.jmir.org/2020/2/e18980/ [PMC free article] [PubMed] [CrossRef] [Google Scholar]

3. Triantafyllidis A, Kondylakis H, Votis K, Tzovaras D, Maglaveras N, Rahimi K. Features, outcomes, and challenges in mobile health interventions for patients living with chronic diseases: A review of systematic reviews. Int J Med Inform. 2019 Dec;132:103984. doi: 10.1016/j.ijmedinf.2019.103984. [PubMed] [CrossRef] [Google Scholar]

4. Kouroubali A, Koumakis L, Kondylakis H, Katehakis DG. An integrated approach towards developing quality mobile health apps for cancer. In: Moumtzoglou A, editor. Mobile Health Applications for Quality Healthcare Delivery. Hershey, PA: IGI Global; 2019. pp. 46–71. [Google Scholar]

c. Problem Statement Definition

1. Containment zones will be identified throughout the country and decided into red, orange & green zones.

    a. **Red Zone** – Infection Hotspots

    b. **Orange Zone** – Some Infection Zone

    c. **Green Zone** – No Infection Zone

2. App captures the user's IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.

1. **IDEATION & PROPOSED SOLUTION**

    a. 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.

Reference: https://www.mural.co/templates/empathy-map-canvas

b. 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.

Reference: https://www.mural.co/templates/empathy-map-canvas\

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

Step-2: Brainstorm, Idea Listing and Grouping

c. 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	Our application is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individuals location using mobile app.The monitoring process will based on the containment zones which is added by the Admin.App captures the user's IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.The users can prevent their health from pandemic diseases.
4.	Social Impact / Customer Satisfaction	Key benefits of the application are monitoring people's activity and alerting them of their safety movements. Taking into account the areas affected by the specific diseases (Like Covid-19) across the country. Provides information about the containment of a particular region by continuously monitoring an individual's location.
5.	Business Model (Revenue Model)	Social media is the best platform to develop this application. This application will increase the confidence among the people. It is great to use, amazing convenience and also have subscription once user hit certain services.

6.	Scalability of the Solution	Containment zones will be identified throughout the country and decided into red, orange & green zones. ● Red Zone – Infection Hotspots ● Orange Zone – Some Infection Zone ● Green Zone – No Infection Zone App captures the user's IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.

d. Problem Solution fit

1. **CUSTOMER SEGMENT(S)**

1. In Admin Panel they can login to the site and update the containment zone locations
2. Based on the location a geofence will be created within a 100 m radius.
3. In User App(Mobile App) will have a user registration and login.
4. After the user logged into the app it will track the user's location and if the user is visiting the containment zone he/she will get an alert notification

2. **JOBS-TO-BE-DONE / PROBLEMS**

1. Tracking the users who are all visiting the containment zones.
2. The Identified containment zones chosen for the testing of the application were users visited one by one
3. Provides information about the containment of a particular region by continuously monitoring an individual's location.

3. **TRIGGERS**

1. The monitoring process will based on the containment zones which is added by the Admin.

2. The users can prevent their health from pandemic diseases.

4. **EMOTIONS: BEFORE / AFTER**

1. Taking into account the areas affected by the specific diseases (Like Covid-19) across the country.
2. App captures the user's IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.

5. **AVAILABLE SOLUTIONS**

1. Containment zones will be identified throughout the country and decided into red, orange & green zones.
    - **Red Zone** – Infection Hotspots
    - **Orange Zone** – Some Infection Zone
    - **Green Zone** – No Infection Zone
    - No proper alerts received by users in the existing applications

6. **CUSTOMER CONSTRAINTS**

1. Containment zones will be identified throughout the country and decided into red, orange & green zones.
    - **Red Zone** – Infection Hotspots
    - **Orange Zone** – Some Infection Zone
    - **Green Zone** – No Infection Zone
2. Taking into account the areas affected by the specific diseases (Like Covid-19) across the

country.

3. App captures the user’s IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.

7. **BEHAVIOUR**

1. In Admin Panel they can login to the site and update the containment zone locations
2. Based on the location a geofence will be created within a 100 m radius.
3. In User App(Mobile App) will have a user registration and login.
4. After the user logged into the app it will track the user’s location and if the user is visiting the containment zone he/she will get an alert notification

1. **CHANNELS of BEHAVIOUR**

    a. **ONLINE**

We are aiming the all type of online users

8.2 OFFLINE

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 using mobile app.

9. PROBLEM ROOT CAUSE

1. Our project aims at building an application that provides information about the containment zones of a particular region by continuously
2. monitoring an individual location.
3. The officials can monitor the user’s activity from their Admin panel.
4. Location data of individuals will be stored in the database (IBM DB2)
5. Alerts will be send to the user using the Notification service and also through the Email service

1. **YOUR SOLUTION**

1. Containment zones will be identified throughout the country

    a. **Red Zone** – Infection Hotspots

    b. **Orange Zone** – Some Infection Zone

    c. **Green Zone** – No Infection Zone

2. App captures the user's IMEI. The police can keep an eye on the people who are frequently violating the lockdown rules.

2. REQUIREMENT ANALYSIS

a. Functional requirement

IBM Cloud,HTML,Javascript,IBM Cloud Object Storage,Python-Flask,Kubernetes,Docker,IBM DB2,IBM Container Registry

b. Non-Functional requirements

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. PROJECT DESIGN

a. Data Flow Diagrams

5.2 Solution & Technical Architecture

Admin Panel – Website

User App – Mobile App

Admin App (portal):

4. They can login to the site and update the containment zones locations in the portal.
5. Based on the location a Geofence will be created within a 100 meters radius.
6. They can be able to see how many people are visiting that zone.

User App (Mobile App):

4. The app will have a user registration and login.
5. After the user logged into the app it will track the user's location and update the database with the current location.
6. If the user is visiting the containment zone he/she will get an alert notification.

Example - Solution Architecture Diagram:

a. 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	High	Sprint-1
		USN-2	As a user, I can register for the application through Gmail		Medium	Sprint-2
	Login	USN-3	As a user, I can log into the application by entering email & password		High	Sprint-3
Customer						

(Web user)						
Customer Care Executive						
Administrator	Login	USN-4	As a admin, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a admin, I can track the user details		High	Sprint-2

4. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Deepak Kumar R,Jaya Karthik Kumar R.D
Sprint-2		USN-2	As a user, I can register for the application through Gmail	10	Medium	Gayathri Devi R S
Sprint-3	Login	USN-3	As a user, I can log into the application by entering email & password	20	High	Jaya Priya Dharshini K
Sprint-1	Login (Admin)	USN-4	As a admin, I can log into the application by entering email & password	10	High	Deepak Kumar R

Sprint-2	Dashboard (Admin)	USN-5	As a admin, I can track the user details	10	High	Deepak Kumar R

a. 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	10	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	10	
Sprint-3	20	6 Days	07 Nov 2022	16 Nov 2022	10	

b. Reports from JIRA

Burnup report

Velocity report

Cumulative flow diagram

5. CODING & SOLUTIONING

a. Feature 1

Admin Panel – Website

Admin App (portal):

7. They can login to the site and update the containment zones locations in the portal.

8. Based on the location a Geofence will be created within a 100 meters radius.
9. They can be able to see how many people are visiting that zone.

b. Feature 2

User App – Mobile App

User App (Mobile App):

7. The app will have a user registration and login.
8. After the user logged into the app it will track the user's location and update the database with the current location.
9. If the user is visiting the containment zone he/she will get an alert notification.

c. Database Schema

Users

1. email
2. password
3. imei

Admins

1. email
2. password

Zones

1. area
2. latlon
3. type

6. TESTING

a. Test Cases

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	1	0	50
Security	2	0	0	2
OutsourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
Final Report Output	4	0	0	4
VersionControl	2	0	0	2

b. User Acceptance Testing

1. Defect Analysis

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1

Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

2. Test Case Analysis

This reportshows the numberof test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	1	0	50
Security	2	0	0	2
OutsourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
Final Report Output	4	0	0	4
VersionControl	2	0	0	2

7. **RESULTS**

a. Performance Metrics

1. In Admin Panel they can login to the site and update the containment zone locations
2. Based on the location a geofence will be created within a 100 m radius.
3. In User App(Mobile App) will have a user registration and login.
4. After the user logged into the app it will track the user’s location and if the user is visiting the containment zone he/she will get an alert notification

1. ADVANTAGES & DISADVANTAGES

Advantages

1. Mobile apps are considered to be a valuable tool for citizens, health professionals, and decision makers
2. Critical challenges imposed by the pandemic, such as reducing the burden on hospitals, providing access to credible information
3. Critical challenges imposed by the pandemic can avoided
4. Reducing the burden on hospitals, providing access
5. Provides information about the containment of a particular region The users can prevent their health from pandemic diseases

Disadvantages

1. The offcials can monitor the user's activity from their Admin panel.
2. Location data of individuals will be stored in the database (IBM DB2) User privacy may be affected by tracking their location
3. No proper alerts received by users in the existing applications
4. Hence they can easily enter into the containment zone Zones are added by admin only.
5. It may cause human error Network unstability lead to huge cause

2. CONCLUSION

Mobile apps are considered to be a valuable tool for citizens, health professionals, and decision makers in facing critical challenges imposed by the pandemic, such as reducing the burden on hospitals, providing access to credible information, tracking the symptoms and mental health of individuals, and discovering new predictors

Mobile apps have been implemented for training, information sharing, risk assessment, self-

management of symptoms, contact tracing, home monitoring, and decision making, rapidly offering effective and usable tools for managing the COVID-19 pandemic.

3. FUTURE SCOPE

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

4. APPENDIX

Source Code

app.py (Used in admin panel)

```
from flask import Flask, render_template, request, redirect, url_for
import ibm_db
app = Flask(__name__, template_folder='templates', static_folder='static')

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate:DigiCertGlobalRootCA;UID=gyj23042;PWD=a5CFymQzrGXGeOnC;", "", "")

# sql1 = "INSERT INTO zones VALUES ('Cbe', '11.11586160918341, 77.18835750740392', 'Green', '0')"
# stmt = ibm_db.exec_immediate(conn, sql1)
# sql = "SELECT * FROM zones"
# stmt = ibm_db.exec_immediate(conn, sql)
# dict = ibm_db.fetch_assoc(stmt)
```

```
# print(dict)

@app.route('/')
def html():
    return render_template('login.html')

@app.route('/logout')
def logout():
    return render_template('login.html')

@app.route('/seezones')
def seezones():
    msg = ''
    sql = "SELECT * FROM zones"
    results = []
    stmt = ibm_db.exec_immediate(conn, sql)
    row_dict = ibm_db.fetch_both(stmt)
    cols = ibm_db.num_fields(stmt)

    while ( row_dict ):
        results.append(row_dict)

        row_dict = ibm_db.fetch_both(stmt)
    print(results)
    if(stmt):
        msg = 'Zones getted successfully !'
        return render_template('/zoneslist.html', zones = results )
    else:
        msg = 'Unable to get zones !'
        return render_template('/addzone.html', msg = msg)

@app.route('/login',methods =['POST'])
def login():
```

```
    msg = ''
    if request.method == 'POST' :
        email = request.form.get('email')
        password = request.form.get('password')
        sql = "SELECT * FROM admins WHERE email =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            msg = 'Logged in successfully !'
            return render_template('/addzone.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
            return render_template('/login.html', msg = msg)

@app.route('/addzone', methods =['POST'])
def addzone():
    msg = ''
    if request.method == 'POST' :
        area = request.form.get('area')
        latlon = request.form.get('latlon')
        type = request.form.get('type')
        insert_sql = "INSERT INTO zones VALUES (?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, area)
        ibm_db.bind_param(prep_stmt, 2, latlon)
        ibm_db.bind_param(prep_stmt, 3, type)
        ibm_db.bind_param(prep_stmt, 4, 0)
        ibm_db.execute(prep_stmt)
        msg = 'You have successfully registered !'
```

```
    return render_template('/addzone.html', msg = msg)

if __name__ == '__main__':
    app.run(host='0.0.0.0',debug=True)
```

api.py (Used in mobile app)

```
from flask import Flask, render_template, request, redirect, url_for, jsonify
import ibm_db
app = Flask(__name__)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate:DigiCertGlobalRootCA;UID=gyj23042;PWD=a5CFymQzrGXGeOnC;", "", "")

# sql1 = "INSERT INTO zones VALUES ('Cbe', '11.11586160918341, 77.18835750740392', 'Green', '0')"
# stmt = ibm_db.exec_immediate(conn, sql1)
# sql = "SELECT * FROM zones"
# stmt = ibm_db.exec_immediate(conn, sql)
# dict = ibm_db.fetch_assoc(stmt)
# print(dict)

@app.route('/register',methods =['POST'])
def register():
    email = request.form.get("email")
    password = request.form.get("password")
    imei = request.form.get("imei")
    sql = "SELECT * FROM users 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 !'
```

```
        return msg
    else:
        sql = "INSERT INTO users (email, password, imei) VALUES (?,?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, email)
        ibm_db.bind_param(prep_stmt, 2, password)
        ibm_db.bind_param(prep_stmt, 3, imei)
        ibm_db.execute(prep_stmt)
        msg = 'Registered successfully !'
        return msg


@app.route('/login',methods =['POST'])
def login():
    email = request.form.get("email")
    password = request.form.get("password")
    sql = "SELECT * FROM users WHERE email =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        msg = 'Logged in successfully !'
        return msg
    else:
        msg = 'Incorrect username / password !'
        return msg

@app.route('/getzones',methods =['GET'])
def seezones():
    msg = ''
    sql = "SELECT latlon FROM zones"
```

```
    results = []
    stmt = ibm_db.exec_immediate(conn, sql)
    row_dict = ibm_db.fetch_both(stmt)
    cols = ibm_db.num_fields(stmt)

    while ( row_dict ):
        results.append(row_dict)

        row_dict = ibm_db.fetch_both(stmt)
    print(results)
    if(stmt):
        return str(results)
    else:
        msg = 'Unable to get zones !'
        return msg

if __name__ == '__main__':
    app.run(host='0.0.0.0',debug=True)
```

GitHub & Project Demo Link

Github Links :

Mobile App : https://github.com/DkDev2001/ContainmentZone.git

Web Site : https://github.com/711119104020/Containment-Zone-Admin.git

Video Link :

https://drive.google.com/file/d/1CdgGy7ieGqeVUTb1IwmIHXy-cwWFyYDQ/view?usp=share_link