

Project Report

Team ID	PNT2022TMID25917
Project Name	CONTAINMENT ZONEALERTING APPLICATION

1.INTRODUCTION

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 to their safety movements.

1.1 Project Overview

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.

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

1.2 Purpose

The main purpose of this application is to alert the people who are in need for any medical emergency and alert them through notification. 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.

2. LITERATURE SURVEY

2.1 Existing problem

1. 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.
2. In a thickly populated nation like India, it is hard to forecast community transmission of COVID-19. Hence, a number of containment zones had been recognized all over the country separated into red, orange, and green zones, individually. People are restricted to move into these containment zones.
3. To help understand whether you've been exposed to someone who reports having COVID-19, you can turn on Exposure Notifications. If you change your mind, you can turn it off. To use the system, you need to either download an official app from your region's government public health authority.
4. Use the Exposure Notification framework to inform people of potential exposure to COVID-19, the disease caused by the SARS-CoV-2 virus. You can build a notification system that employs random, rotating keys and identifiers to convey positive diagnoses in addition to data such as associated symptoms, proximity and duration.
5. COVID-19 tracking tools or contact-tracing apps are getting developed at a rapid pace by different governments in their respective countries. This study explores one such tool called Aarogya Setu, developed by the Government

of India. It is a mobile application developed under the Health Ministry, as a part of the E-Governance initiative, to track and sensitize the citizens of India in a joint battle against COVID-19 spread. The study aims to understand various useful features of this tool and to present different concepts of data science applied within the application along with its importance in managing the ongoing pandemic.

6. The Seha mobile application is one of the most innovative mobile health applications being used in Saudi Arabia. The application was developed for providing e-consultations through audio and video modes for users in the comfort of their own homes. The application employs AI technologies, enabling users to receive safe medical information, and enhancing the user's experience during the consultation process. It also features a health assessment tool, where the users need to answer certain questions. Based on the responses, a health score is developed based on a disease or condition and the relevant feedback and precautions are prescribed.
7. 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.
8. The article gives emphasis on creating to containments to prevent the outburst of disease, however it does not explain about how to make these zones quickly and accurately. In their paper (Maier & Brockmann, 2020) explained about the effective containment to control specifically COVID-19 cases in China. The model which they explained in their paper captures both quarantine of symptomatic infected individuals and other population isolation practices. The focus of the research is on contagion process and general effects as well as significance of the containment.

2.2 References

Paper 1:

Title:Development of An Android Application for Viewing Covid-19 Containment Zones and Monitoring Violators Who are Trespassing into It Using Firebase and Geofencing.

Author: RanajoyMallik, AmlanProtimHazarika, SudarshanaGhoshDastidar, Dilip Sing, and RajibBandyopadhyay , Department of Instrumentation and Electronics Engineering, Jadavpur University, Salt Lake Campus, Kolkata, 700 098 India.

Paper 2:

Title: Moblie Geo-Fencing Triggers for Alerting Entries into COVID-19 Containment Zones using IoT.

Author: M. V. RamanaRao (Osmania University, India), ThondepuAdilakshmi (Vasavi College of Engineering, India), M. GokulVenkatesh (Sidhartha Medical College, India) and Jothikumar R (Department of Computer Science and Engineering, Shadan College of Engineering and Technology, India).

Paper 3:

Title: Use the COVID-19 Exposure Notifications System on your Android phone.

Author: Dilip Sing and Rajib B and Yopadhyay.

Paper 4:

Title: Implement a COVID-19 exposure notification system that protects user privacy.

Author: JaraneeMonchan,Department of Agro-Industrial Technology, Kasetsart University, Bangkok, Thailand.

Paper 5:

Title: Analysis of COVID-19 Tracking tool in India.

Author: Rajan Gupta, Deen Dayal Upadhyaya College, University of Delhi, India. Manan Bedi,Prashi Goyal, Srishti Wadhera and Vaishnavi Verma, Shaheed

Sukhdev College of Business Studies, University of Delhi, India.

Paper 6:

Title: Development of Seha app.

Author: Alanzi T

Paper 7:

Title: Awareness about the Virus.

Author: Ranajoy Mallik, Department of Instrumentation and Electronics Engineering, Jadavpur University, Salt Lake Campus, Kolkata, 700 098 India.

Paper 8:

Title: Defining Covid 19 containment zones using Kmeans dynamically.

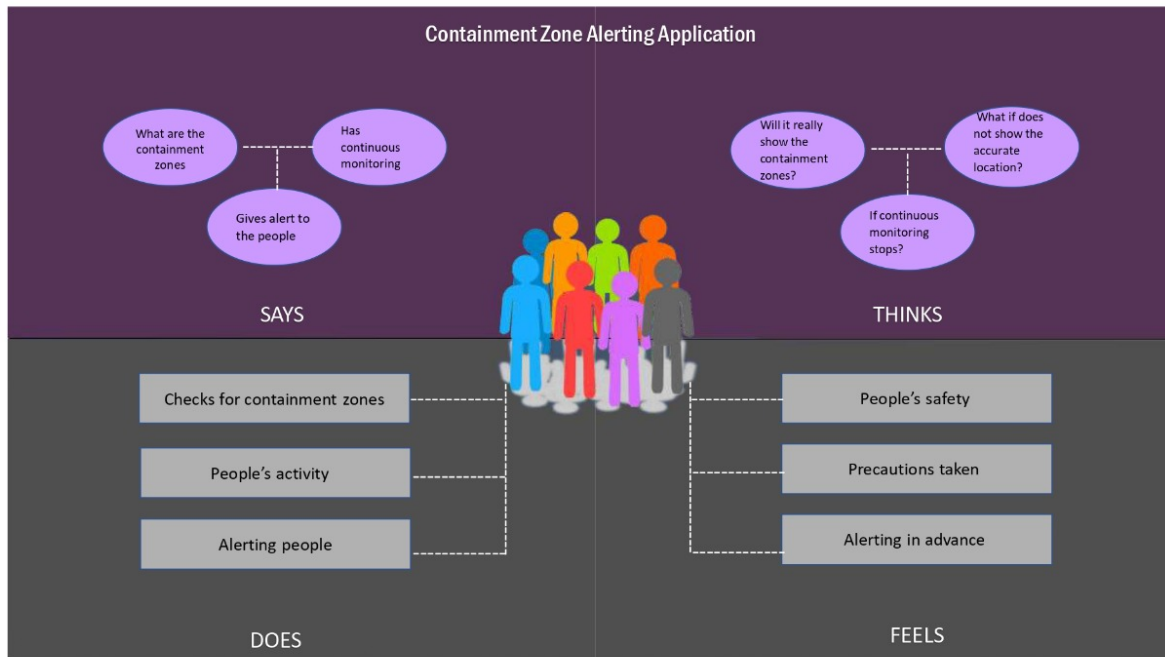
Author: Satish Chinchorkar, September 25th, 2020, Research square.

2.3 Problem Statement Definition

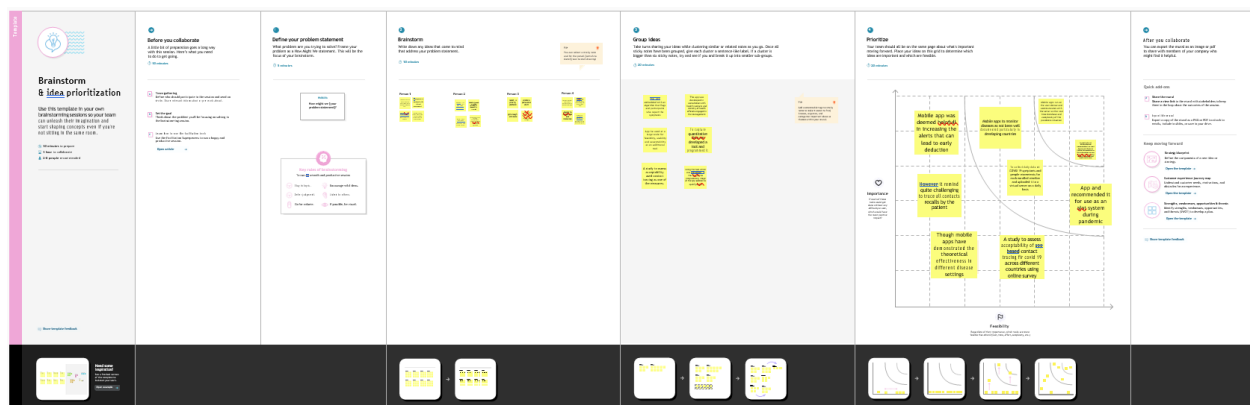
An issue that has to be resolved or a situation that can be improved is summarised in a problem statement. The existing state and the desired state of a process are identified in the problem statement. By continuously tracking a user's location, this application aims to warn users of containment zones in a certain area and provide information about them. The application's main advantages include tracking users' activities and warning them of potential dangers. The application's main advantages include tracking users movements and warning them of safety precautions.

3. Ideation & Proposed Solution

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Development of an android application for viewing the covid containment zones and also alerting the users not to enter the affected area using cloud and geofencing by sending notification.
2.	Idea / Solution description	To create an easy-to-use android application to alert the user when they enter a Containment Zone. To provide accurate results and alerting at the exact time when they enter the zone. This is done with the help of integration of Google Maps.
3.	Novelty / Uniqueness	<ol style="list-style-type: none"> 1. Development of an Android application is necessary which can inform people of the Covid19 containment zones and prevent trespassing into these zones. 2. Android application updates the locations of the areas in a Google map which are identified to be the containment zones. 3. The application also notifies the users if they have entered a containment zone and upload the details of individual in online database.
4.	Social Impact / Customer Satisfaction	The application saves people's life from restricting them from entering the Containment zone which saves them from catching the disease. Also shows precautionary measures when they entered the zones
5.	Business Model (Revenue Model)	<p>Can tie up with people with normal and premium charges.</p> <p>The data that is derived can be used in Government sectors.</p> <p>Can tie up the Government and get profit through that.</p>

6.	Scalability of the Solution	The application will be useful for all people from saving their life's from catching the disease by alerting them when they accidentally entered the containment zone.
----	-----------------------------	--

4. Requirement Analysis

4.1 Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	User can register through Email id or current phone number.
FR-2	User Confirmation	Confirmation can be done by verification code through Mail or OTP.
FR-3	Track the location	Trace the trespassers by using Google map API.
FR-4	Affected areas are shown	Containment zones were marked and trespassers are indicated by geofencing.
FR-5	Alert notification	By tracking their location using GPS system, notification or message will be sent if the user enters the Containment zone.

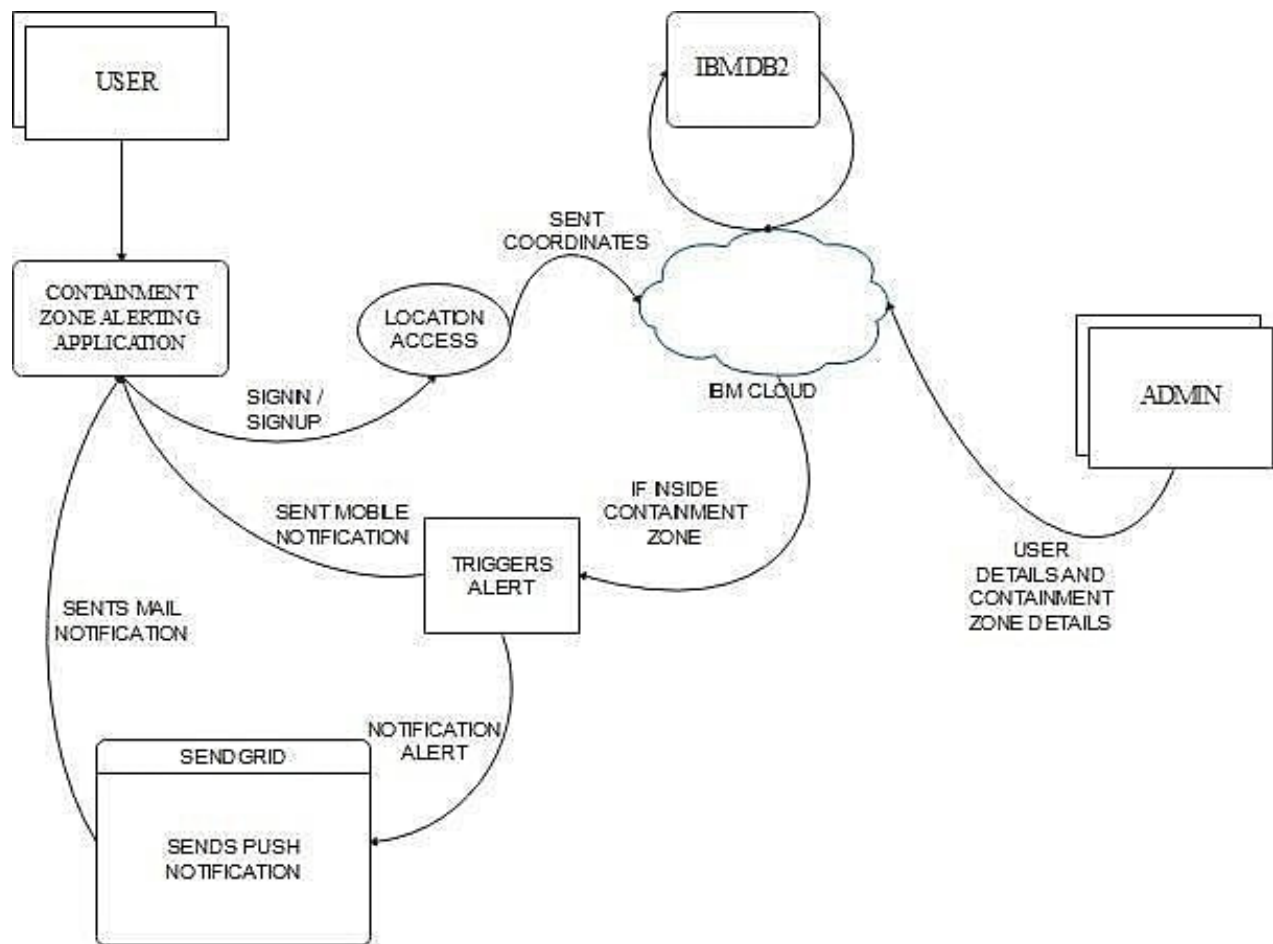
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User interface is very effective to use when compared with other.
NFR-2	Security	Data from the user will be secured properly.
NFR-3	Reliability	User can trust this application and travel safely.
NFR-4	Performance	Most appropriate results can be achieved due to using the Geofencing and GPS.
NFR-5	Availability	The application uses the network to load the google Maps to retrieve containment zones. It is available for good range of network bandwidth.
NFR-6	Scalability	This application can be accessed from any place and information about the zones are up to date.

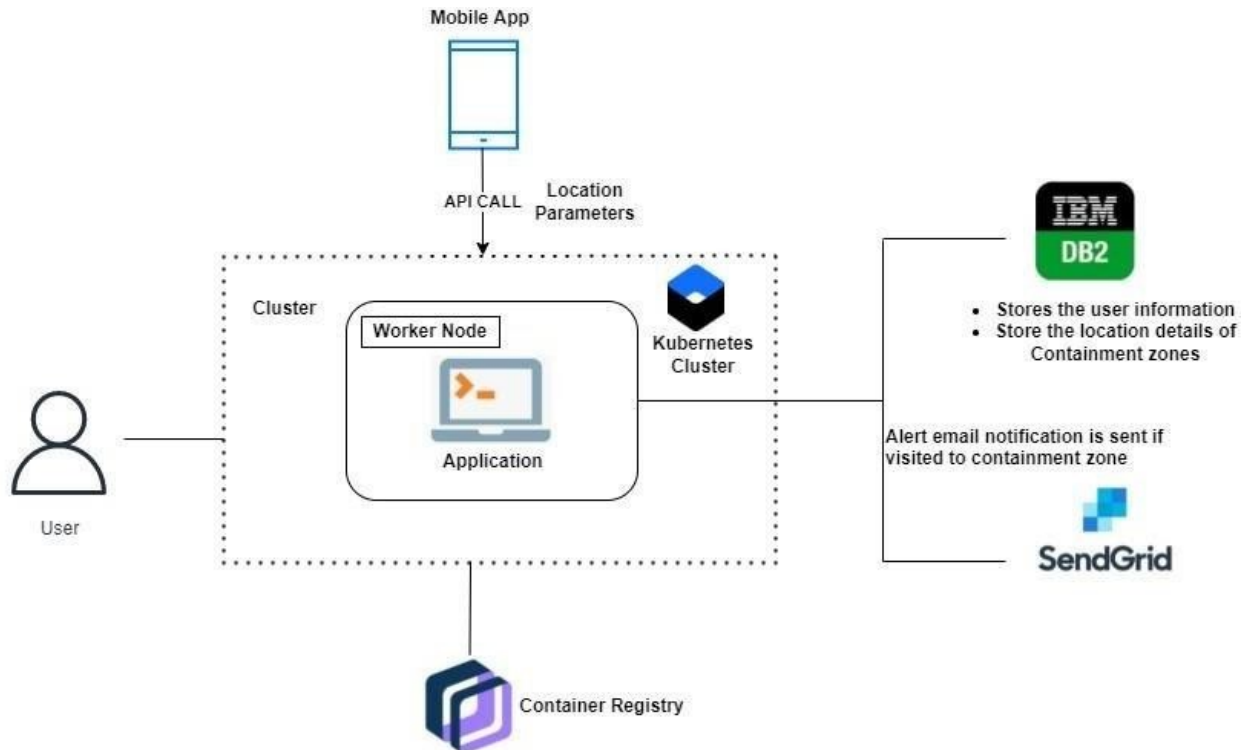
5. Project Design

5.1 Data Flow Diagram

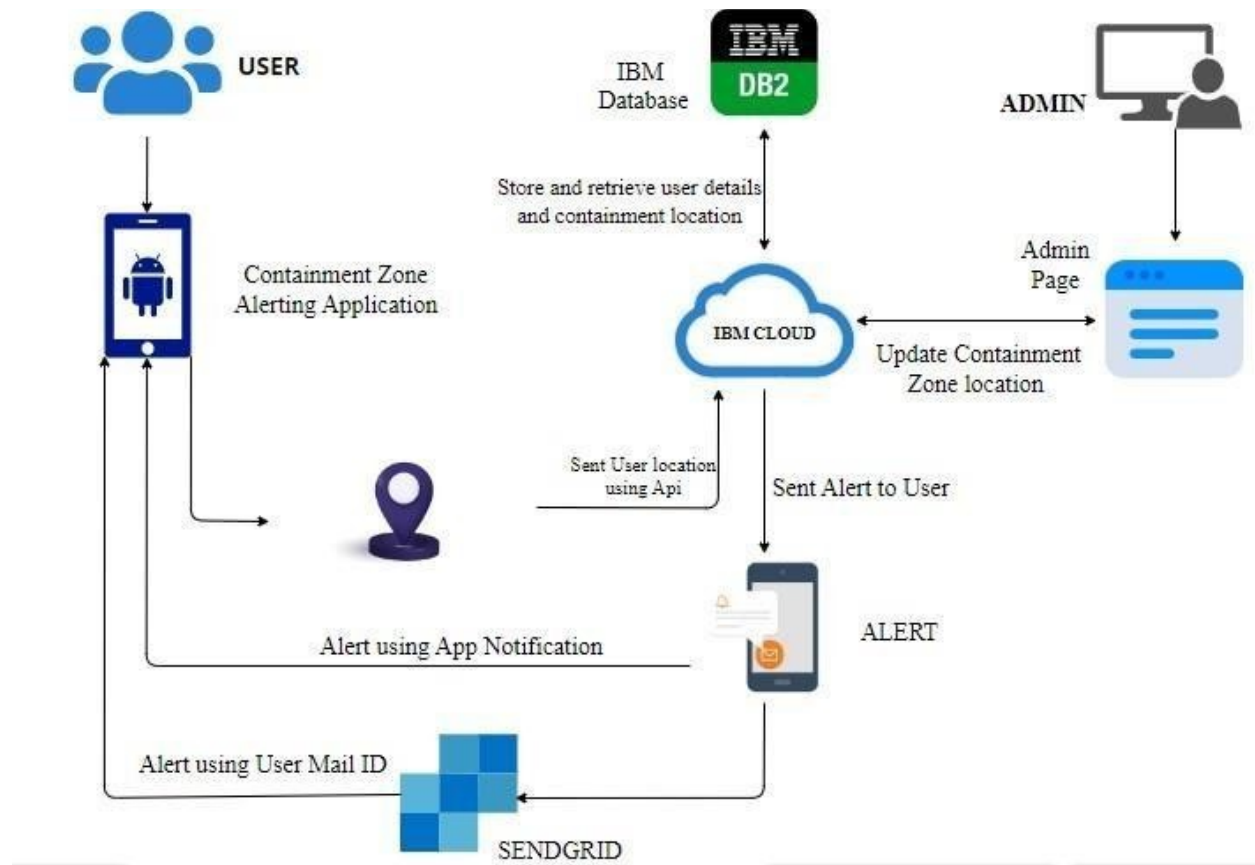


5.2 Solution & Technical Architecture

Solution Architecture:



Technical Architecture:



6. 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 (web and android)	USN-1	<p>USER:</p> <p>I can register for the application by entering my email and password</p>	3	High	<p>Sruthi B</p> <p>Gayathri M</p> <p>Harini S</p> <p>Sujitha S</p>
		USN-2	<p>USER:</p> <p>I will receive a confirmation email once I have registered for the application</p>	2	High	<p>Sruthi B</p> <p>Gayathri M</p> <p>Harini S</p> <p>Sujitha S</p>
	Login (web and android)	USN-3	<p>USER:</p> <p>I can log into the application by entering my</p> <p>4</p>	3	High	<p>Sruthi B</p> <p>Gayathri M</p> <p>Harini S</p> <p>Sujitha S</p>

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	Sruthi B Gayathri M Harini S Sujitha S
		USN-5	As a user, I can log into the application by entering email & password	5	High	Sruthi B Gayathri M Harini S Sujitha S

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	Sruthi B Gayathri M Harini S Sujitha S
		USN 7	ADMIN: I need to differentiate the containment zones based on the intensity of infection.	3	Medium	Sruthi B Gayathri M Harini S Sujitha S

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 the containment zone through the notification	5	Medium	Sruthi B Gayathri MHarini S Sujitha S
	Data collection	USN 9	ADMIN: I need to store user details on the cloud	5	Medium	Sruthi B Gayathri MHarini S Sujitha S
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
		USN 10	ADMIN: I need to collect details about covid - 19 cases from verified sources	5	Medium	Sruthi B Gayathri MHarini S Sujitha S

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	7 Days	25 Oct 2022	31 Oct 2022	20	31 Oct 2022
----------	----	--------	-------------	-------------	----	-------------

Sprint-2	20	6 Days	01 Nov 2022	06 Nov 2022	20	06 Nov 2022
Sprint-3	20	5 Days	07 Nov 2022	11 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	12 Nov 2022	17 Nov 2022	20	17 Nov 2022

6.2 Sprint Delivery Schedule

Velocity:

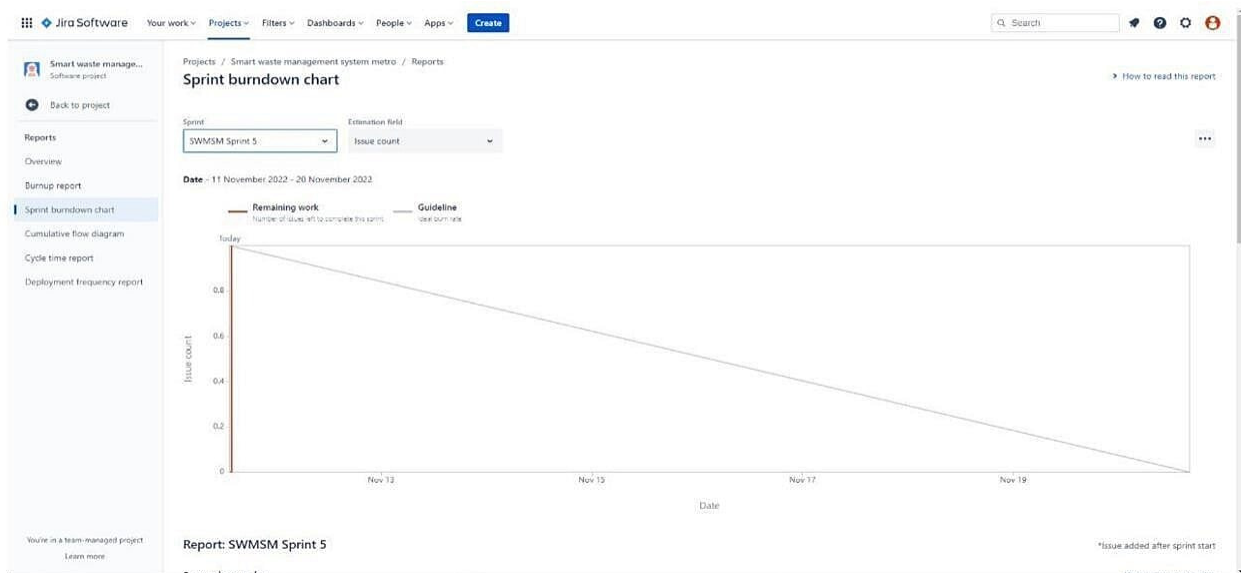
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).

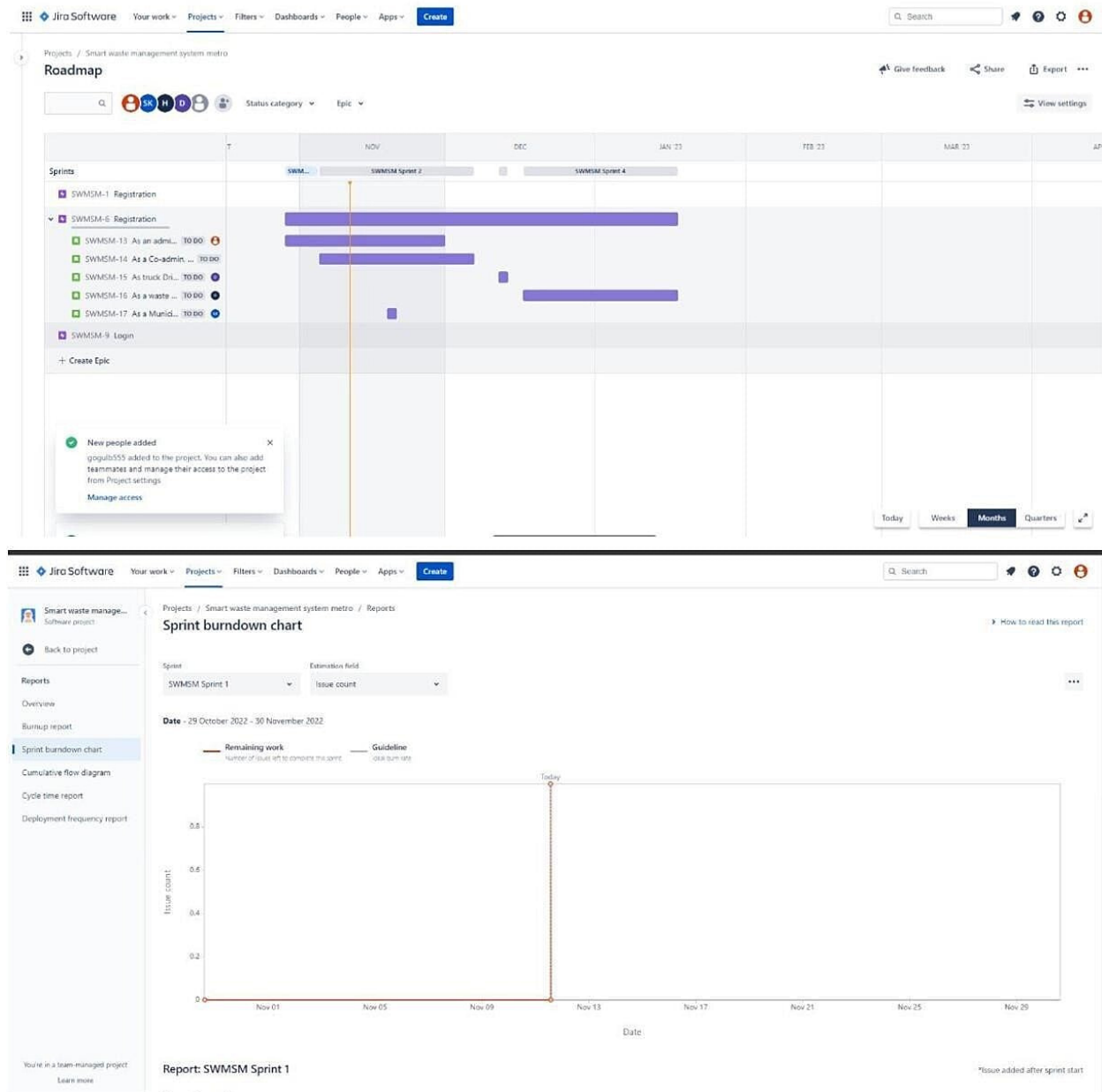
Let's calculate the team's average velocity (AV) per iterationunit (story points per day)

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

6.3 Reports from JIRA

BURNDOWN CHART





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

Location permission



Android Auto

LOCATION ACCESS FOR THIS APP

- ☒ Allow only while using the app
- ☐ Ask every time
- ☐ Don't allow

Use precise location

When precise location is off, apps can access your approximate location



[See all Android Auto permissions](#)

Please enable Location Services

You currently have all Location Services for this device or application disabled. Please enable them in Settings.

[CLOSE](#) [CONNECTION SETTINGS](#)

Allow Changes



Don't Allow Changes

Disallowing changes locks the settings shown below and prevents new apps from using location services.

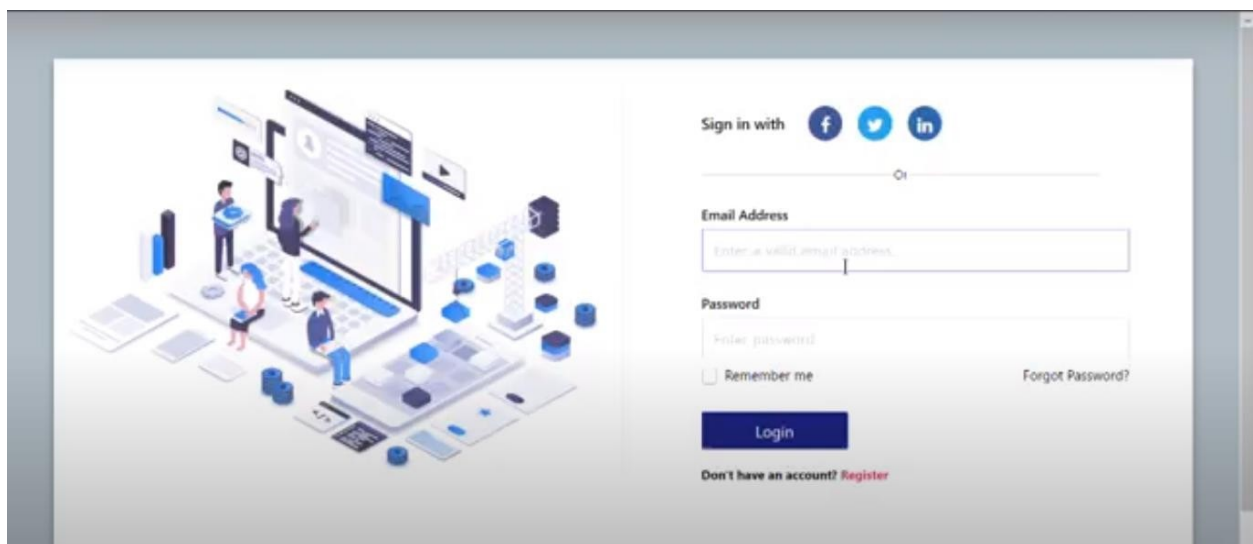
Location Services



Location Services uses GPS, Bluetooth, and crowd-



8. RESULTS:



Stay Alert!!!

Prevent Yourself from entering into Containment zone. Enable your location to get notified about the covid zones available nearby.

ENABLE LOCATION

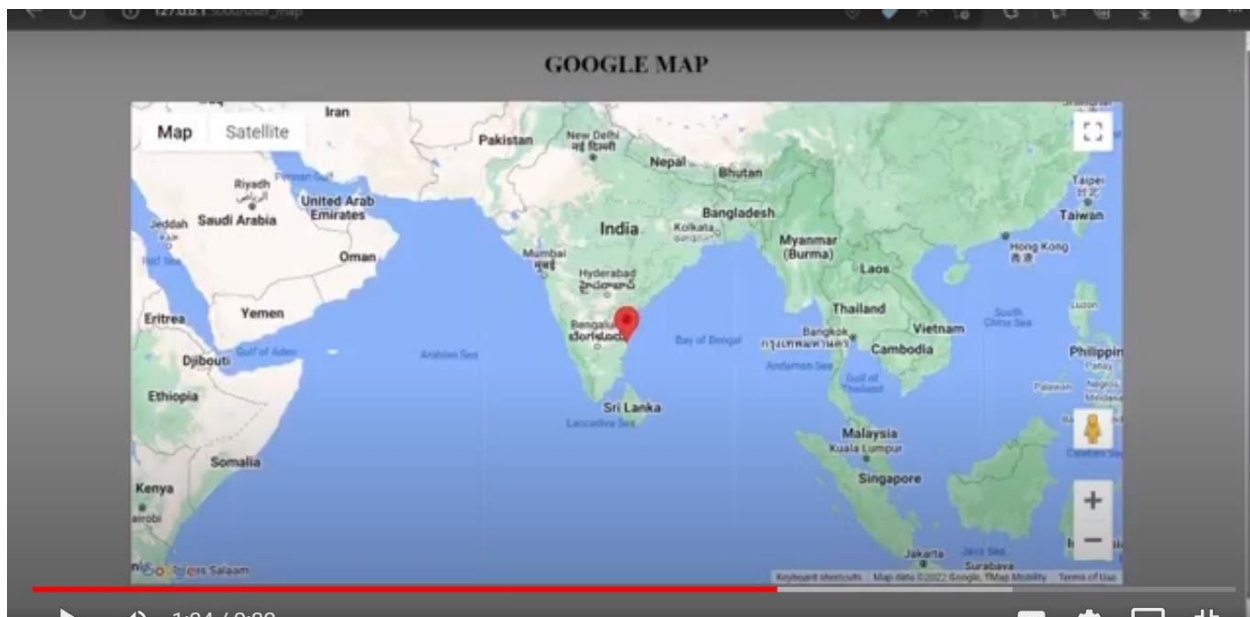


Containment zone map

Containment activities are undertaken in a confined area to prevent infection from getting established and prevent its spread outside the area.

ZONE MAP





CONTAINMENT ZONES

S.NO LOCATION ZONE

1	Chennai	Red
2	Coimbatore	Red
3	Trichy	Green
4	Vellore	Yellow
5	Salem	Yellow



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

DISADVANTAGES:

- Accuracy of application depends 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 individual's location. Key benefits of the application are monitoring people's 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. Timeseries analysis is used to predict crop production as well as sales in different quarters.

12. APPENDIX

Source Code:

MenuActivity.java

```
package com.example.covid_19alertapp.activities;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.lifecycle.Observer;
import androidx.work.Constraints;
import androidx.work.PeriodicWorkRequest;
import androidx.work.WorkInfo;
import androidx.work.WorkManager;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

import com.example.covid_19alertapp.R;
import com.example.covid_19alertapp.extras.Constants;
import com.example.covid_19alertapp.extras.LogTags;
import com.example.covid_19alertapp.services.BackgroundWorker;
import com.example.covid_19alertapp.sharedPreferences.MiscSharedPreferences;

import java.util.List;
import java.util.concurrent.TimeUnit;

public class MenuActivity extends AppCompatActivity {
    /*
    starter activity to test and get the permissions + all time running start worker
    overwrite or edit this later, keeping the permission codes
```



```
*/
```

```
Button home_btn;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    home_btn = findViewById(R.id.home_button_menu);
```

```
    home_btn.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(View v) {
```

```
            finish();
```

```
        }
```

```
    });
```

```
    // start background worker for always
```

```
    startWorker();
```

```
}
```

```
private void startWorker() {
```

```
    if(!MiscSharedPreferences.getBgWorkerStatus(this)){
```

```
        Constraints constraints = new Constraints.Builder()
```

```
            .setRequiresBatteryNotLow(true)
```

```
            .setRequiresCharging(false)
```

```
            .build();
```

```
        PeriodicWorkRequest promptNotificationWork =
```

```
            new PeriodicWorkRequest.Builder(BackgroundWorker.class, 30, TimeUnit.MINUTES)
```

```
                .setConstraints(constraints)
```

```
                .addTag(Constants.background_WorkerTag)
```

```
                .build();
```

```
        WorkManager.getInstance(getApplicationContext()).getWorkInfoByIdLiveData(promptNotificationWork.getId())
```

```
            .observe(this, new Observer<WorkInfo>() {
```

```
                @Override
```

```
                public void onChanged(@Nullable WorkInfo workInfo) {
```

```

        if (workInfo != null && workInfo.getState() == WorkInfo.State.ENQUEUED) {
            Log.d(LogTags.Worker_TAG, "onChanged: worker is enqueued");

            // set shared preference true
            MiscSharedPreferences.setBgWorkerStatus(MenuActivity.this, true);
        }

        if (workInfo != null && workInfo.getState() == WorkInfo.State.CANCELLED) {
            Log.d(LogTags.Worker_TAG, "onChanged: worker was stopped. why?");

            // set shared preference false
            MiscSharedPreferences.setBgWorkerStatus(MenuActivity.this, false);
        }
    }
});

WorkManager.getInstance(getApplicationContext())
    .enqueue(promptNotificationWork);

}

}

public void uploadClick(View view) {

    if(!MiscSharedPreferences.getUploadStatus(this)) {

        Intent intent = new Intent(this, UploadLocationsActivity.class);
        startActivity(intent);
    }
    else{

        // show dialog and prevent

        AlertDialog.Builder builder = new AlertDialog.Builder(this);

        builder.setMessage(getText(R.string.cant_upload_twice_message))
            .setCancelable(false)
            .setPositiveButton(getText(R.string.permissions_dialogbox_positive), new
DialogInterface.OnClickListener() {

```

```

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    })
    .setNegativeButton("Override", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();

            // TODO: remove this
            Intent intent = new Intent(MenuActivity.this, UploadLocationsActivity.class);
            startActivity(intent);
        }
    });

    AlertDialog alertDialog = builder.create();
    alertDialog.show();

}
}

public void startNewsFeed(View view)
{
    startActivity(new Intent(getApplicationContext(), NewsFeedActivity.class));
}

public void openSettingsClick(View view) {

    Intent intent = new Intent(this, TrackerSettingsActivity.class);
    startActivity(intent);

}

public void showMatchedLocationsClick(View view) {

    Intent intent = new Intent(getApplicationContext(), ShowMatchedLocationsActivity.class);
    startActivity(intent);

}

```

```

    public void startMyLocationsMap(View view) {

        startActivity( new Intent(this, MyLocationsMapsActivity.class) );

    }
}

```

LocationmapActivity.java

```

package com.example.covid_19alertapp.activities;

import androidx.fragment.app.FragmentActivity;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

import com.example.covid_19alertapp.R;
import com.example.covid_19alertapp.models.MapMarkerLocation;
import com.example.covid_19alertapp.roomdatabase.VisitedLocations;
import com.example.covid_19alertapp.roomdatabase.VisitedLocationsDao;
import com.example.covid_19alertapp.roomdatabase.VisitedLocationsDatabase;
import com.example.covid_19alertapp.sharedPreferences.UserInfoSharedPreferences;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

import java.util.ArrayList;
import java.util.List;

public class MyLocationsMapsActivity extends FragmentActivity implements
    OnMapReadyCallback, GoogleMap.OnMarkerClickListener {

    private GoogleMap mMap;

```

```

// model
private List<MapMarkerLocation> locations = new ArrayList<>();
private int listPosition;

// local database
private VisitedLocationsDatabase roomDatabase;
private VisitedLocationsDao visitedLocationsDao;
private List<VisitedLocations> visitedLocationsList = new ArrayList<>();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    roomDatabase = VisitedLocationsDatabase.getDatabase(this);
    visitedLocationsDao = roomDatabase.visitedLocationsDao();

    setContentView(R.layout.activity_my_locations_maps);
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
    .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // move camera to home
    moveCameraToHome();

    // fetch locations from local db and plot on map
    fetchNShowLocationMarkers();

    mMap.setOnMarkerClickListener(this);
}

private void moveCameraToHome() {

    // home = latitude,longitude

```

```

String[] home = UserInfoSharedPreferences.getHomeLatLng(this).split(",");

LatLng homeLatLng = new LatLng( Double.valueOf(home[0]), Double.valueOf(home[1]));
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(homeLatLng, 16.5f);

}

private void fetchNShowLocationMarkers() {

    roomDatabase.databaseWriteExecutor.execute(new Runnable() {
        @Override
        public void run() {

            visitedLocationsList = visitedLocationsDao.fetchAll();

            listPosition = 0;

            for (VisitedLocations visitedLocation: visitedLocationsList) {

                // pk = latLon_dateTime
                String[] splitLLDT = visitedLocation.splitPrimaryKey();

                final MapMarkerLocation location = new MapMarkerLocation(splitLLDT[0],
splitLLDT[1]);
                locations.add(location);

                // plot marker on map
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {

                        String markerTitle = locations.get(listPosition).getMeaningfulDateTime();
                        LatLng markerLatLng = new LatLng(locations.get(listPosition).getLatitude(),
locations.get(listPosition).getLongitude());

                        // show the marker
                        Marker myLocationMarker =
                            mMap.addMarker( new
MarkerOptions().position(markerLatLng).title(markerTitle));

```

```

        // tag = primary key of local db
        myLocationMarker.setTag(
            locations.get(listPosition).getRawLatLon() +
            "_" +
            locations.get(listPosition).getRawDateTime()
        );

        listPosition++;

    }
});

}

}

}

@Override
public boolean onMarkerClick(final Marker marker) {

    // present delete location option to user

    // marker tag has local db PK
    final String tag = (String) marker.getTag();

    AlertDialog Dialog = new AlertDialog.Builder(this)
        .setMessage(marker.getTitle())
        .setPositiveButton("Delete", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                // delete location from local db
                roomDatabase.databaseWriteExecutor.execute(new Runnable() {
                    @Override
                    public void run() {

```

```

        visitedLocationsDao.deleteByPrimaryKey(tag);

    }
});

marker.remove();
Toast.makeText(MyLocationsMapsActivity.this, "location removed",
Toast.LENGTH_LONG)
    .show();

dialog.dismiss();

    }
})
.setNegativeButton("dismiss", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
})
.show();

return false;
}
}

```

Github Link:

<https://github.com/IBM-EPBL/IBM-Project-3462-1658564334>

Demo video Link:

<https://www.youtube.com/embed/iu-twG725jMS>

