

Summer Project On

Codet

By

**Mayank Mali (2021510033)
Ankita Selokar (2021510062)**

Under the guidance of
Internal Supervisor

Prof. Nikhita Mangaonkar



Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Mayank Mali (2021510033)
Ankita Selokar (2021510062)

Have satisfactorily carried out work on the project
entitled

“Codet”

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Nikhita Mangaonkar

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Mayank Mali (2021510033)
Ankita Selokar (2021510062)

Have successfully completed the Project report on

“Codet”,

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF DEPARTMENT

PRINCIPAL

Contents

Abstract	i
Objectives	i
List Of Figures	ii
List Of Tables	ii
1 Introduction	1
1.1 Problem Definition	1
1.2 Objectives and Scope	1
1.2.1 Objectives	1
1.2.2 Scope	1
1.3 System Requirements	2
2 Software Requirement Specification (SRS) and Design	3
2.1 Purpose	3
2.2 Definition	3
2.3 Overall Description	3
2.3.1 Product Functions	3
2.3.2 User Characteristics	4
2.4 System Feature	4
2.4.1 Functional Requirement	4
2.4.2 Non-Functional Requirement	4
3 Project Analysis and Design	5
3.1 Methodologies Adapted	5
3.2 Modules	6
3.2.1 Activity diagram	6
3.2.2 Deployment Diagram	7
3.2.3 Architectural Design	7
3.2.4 Gantt Chart	8
3.2.5 Use-Case	9
4 Project Implementation and Testing	12
4.1 Home page	12
4.2 Menu	13
4.3 Vaccination center	14
4.4 Precautions	15
4.5 Symptoms	16
4.6 News updates	17
4.7 Testing Lab	18
4.8 Code 1	19
4.9 Code 2	20
4.10 Code 3	21

4.11	Code 4	22
4.12	Code 5	23
5	Test Cases	24
6	Limitations	24
7	Future Enhancements	24
8	System maintainance	25
9	Bibliography	25
9.1	Web References	25

Abstract

When the COVID-19 outbreak was reported to first originate from Wuhan, China, it has been declared as a Public Health Emergency of International Concern (PHEIC) on 30 January 2020 by WHO, and it has spread over 228 Countries and Territories. As the disease spreads around the globe, it has evolved into a world-wide pandemic, endangering the state of global public health and becoming a serious threat to the global community. To combat and prevent the spread of the disease, all individuals should be well-informed of the rapidly changing state of COVID-19. This needs to keep track of change in number of cases and recoveries, and to keep user update with the surrounding. In the endeavor of accomplishing this objective, the idea of building 'CODET' which is COVID-19 real-time analytical tracking application with the latest status of the disease and relevant analytical insights. The real-time tracker is designed to cater to the general audience without advanced statistical aptitude. It aims to communicate insights through various straightforward and concise data visualizations that are supported by sound statistical foundations and reliable data sources. In addition to it, this application shows availability of Vaccination, aware about covid symptoms and provide precautions, locate the nearby covid testing labs.

Objectives

The Android based Application "Codet" is used

- To fetch the data and keep track on Covid-19 cases.
- To get the availability of vaccine and nearby vaccine centers.
- To spread awareness about symptoms of Covid-19.
- To provide precautions from Covid-19.
- To get the location of nearby vaccine centers.

List of Figures

3.1.1Diagrammatic Representation of Iterative Model	5
3.2.1Activity Diagram	6
3.2.2Deployment Diagram	7
3.2.3Architectural Diagram	7
3.2.4Gantt Chart	8
3.2.5Use-Case Diagram	9
4.1.1 Home page	12
4.2.1 Menu	13
4.3.1Vaccination centers	14
4.4.1Precautions	15
4.5.1Symptoms	16
4.6.1News updates	17
4.7.1Testing Lab	18

List of Tables

1.5.1 Hardware Requirements on Server Side	2
1.5.2 Hardware Requirements on Client Side	2
1.5.3 Software Requirements on Server Side	2
1.5.3 Software Requirements on Client Side	2
4.2.1 Use Case Table - Vaccination center	10
4.2.2 Use Case Table - Vaccination center	10
4.2.3 Use Case Table - Precautions	10
4.2.4 Use Case Table - Symptoms	11
4.2.5 Use Case Table - News update	11
6.1 Test Case - Login and Register	24

1 Introduction

1.1 Problem Definition

To After the outbreak of Covid-19 in 2020, it was not possible to keep update of all the information at once. User have to refer different sites and application for each functionality. If a user wants to search for availability of vaccine and testing center, he needs to refer two different platforms to use these functionalities. There is no such platform where covid-19 tracking, availability of vaccine centers, testing centers, symptoms and precautions all are available at one place.

1.2 Objectives and Scope

1.2.1 Objectives

The Android based application "Codet" is

- To fetch the data and keep track on Covid-19 cases.
- To get the availability of vaccine and nearby vaccine centers.
- To spread awareness about symptoms of Covid-19.
- To provide precautions from Covid-19.
- To get the location of nearby vaccine centers.

1.2.2 Scope

By using this application user can get update of Covid-19 global statistics of total confirmed case, active cases, recovered patients and total deaths till the date.

Precautions to take and the covid-19 symptoms are provided to spread awareness and to avoid the rumors.

It has functionality which provides the name and address of nearby vaccination center. Based on user location it suggests the nearby covid testing centers.

1.3 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Android Device with Touch Screen minimum 5" inch Display
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Frontend	XML
Tool Used	Git for collabarative working, Android Studio

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Android/IOS Smartphone
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of our project is to develop an UI application that can help user to keep track on Covid-19 cases and get updates on Covid-19 from all around the world.

This application helps users to book Covid-19 vaccination according to their location and shows the Covid testing lab nearby.

2.2 Definition

To build a Covid-19 Application so the user can have all the essential features to keep update of Covid-19.

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. Vaccination center: This will display the nearest vaccine center depend on user pincode and the date.
2. Testing lab: This will locate all the nearby testing lab according to the users location.
3. Symptoms: This will contain information regarding the Covid-19 symptoms.
4. Precautions: This will contain information regarding the Covid-19 Precautions.
5. New Update: User can view latest and verified news.

2.3.2 User Characteristics

- User have to enter the pincode to view the availability of vaccination in nearby vaccination center

2.4 System Feature

2.4.1 Functional Requirement

1. Vaccination center: The application need user's area pin code to fetch the data and show the availability of vaccination center near users locations.
2. Testing lab: This application need access to users current location to provide the location of nearby testing labs. It uses the Google map.
3. Symptoms and Precautions: Researched based symptoms of covid-19 and Doctor suggested precautions to avoid covid-19 are provided to the user.
4. New Update: The functionality with latest news from the reliable resources, which will bring the user ease to keep update with the covid-19 news around the world.

2.4.2 Non-Functional Requirement

1. Usability : Usability is the degree to which a software can be used by specified users to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.sThe application can be easily used by the user and the admin.
2. Flexibility : Flexibility is the effort required to modify an operational program. The software can be change according to the need.
3. Maintainability : This application is easy to maintain.
4. Portability : The effort required to transfer the program from one hardware and/or software system environment to another. This application is portable as it use same basic software.
5. Timeliness : The application performs all the operations in less amount of time.

3 Project Analysis and Design

3.1 Methodologies Adapted

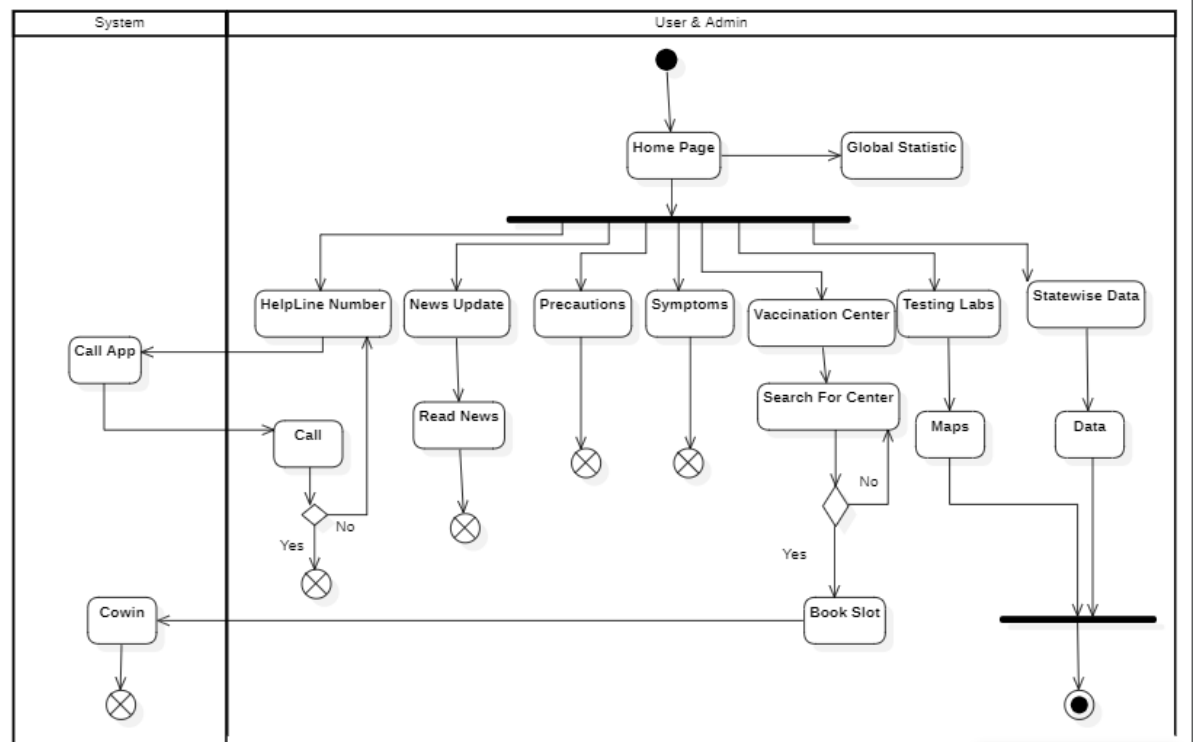
The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration. The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process. While creating this application in the first iteration we focused on creating the simple modules like static pages and other later iterations, precautions and symptoms. We focused on creating the complex modules such as data fetching, displaying data from API in second iteration. And then as modules were completed, they were evaluated and again if any glitches were there planning was done to solve those and the process went on.



3.1.1: Diagrammatic Representation of Iterative Model

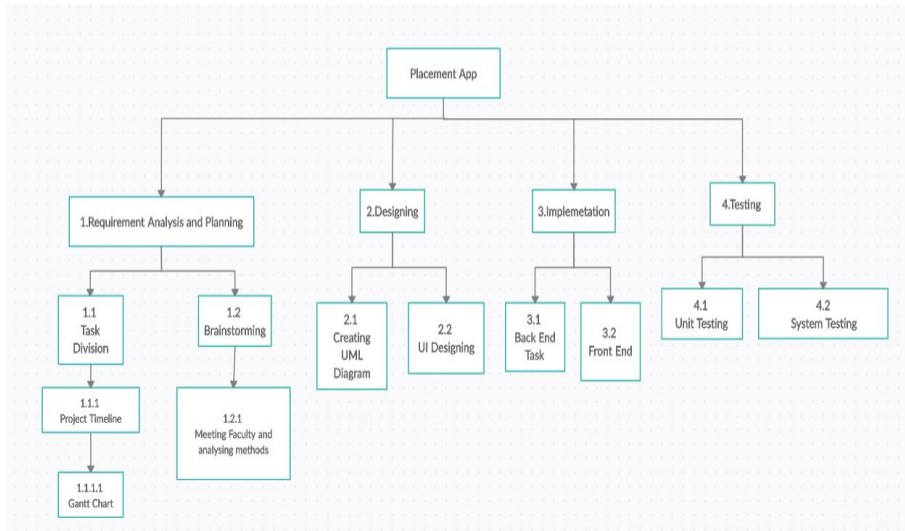
3.2 Modules

3.2.1 Activity diagram



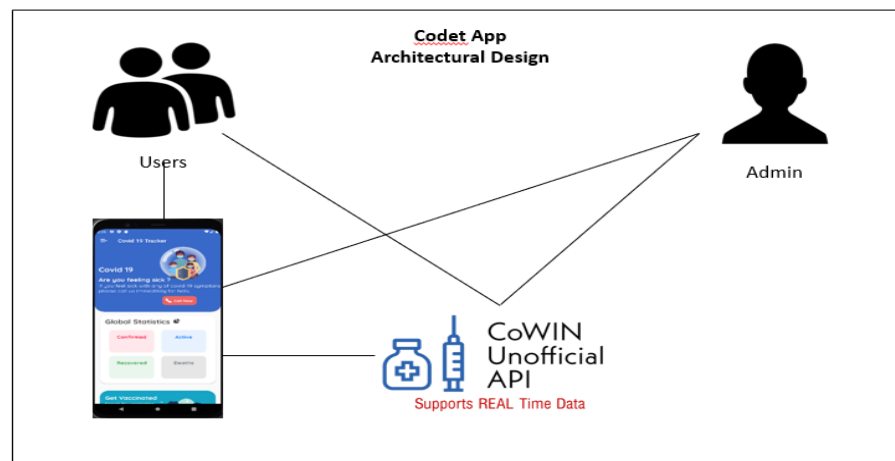
3.2.1: Activity Diagram

3.2.2 Deployment Diagram



3.2.2: Deployment Diagram

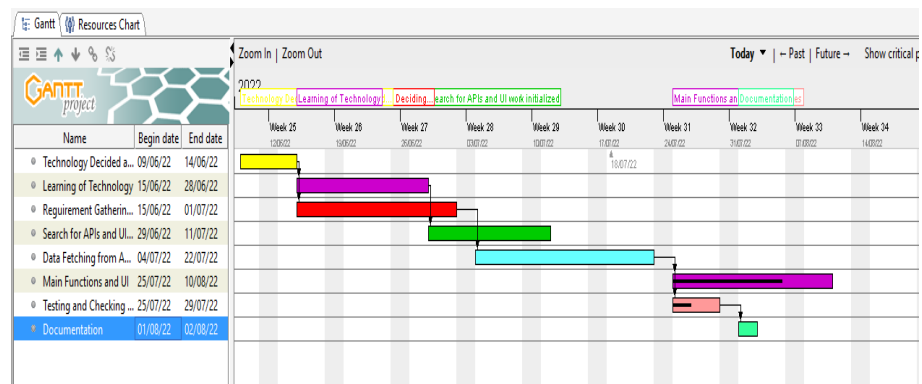
3.2.3 Architectural Design



3.2.3: Architectural Diagram

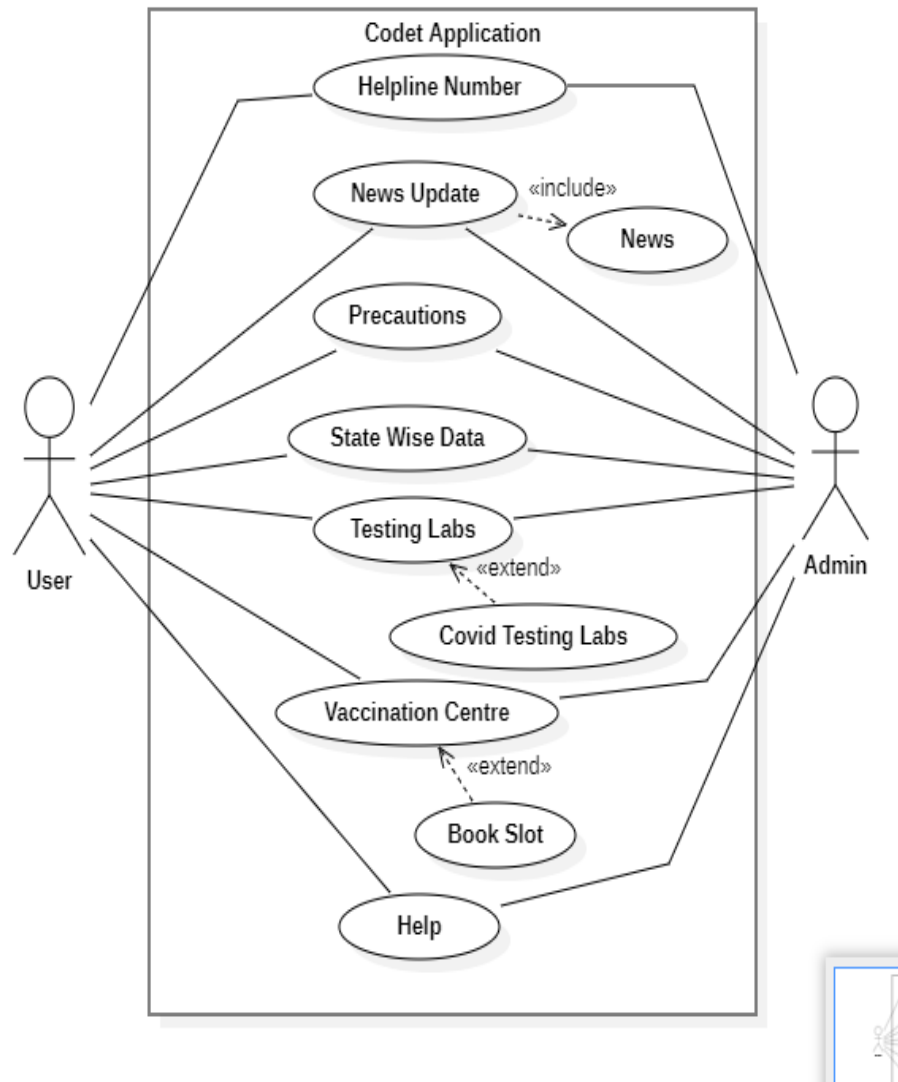
Codet

3.2.4 Gantt Chart



3.2.4: Gantt Chart

3.2.5 Use-Case



3.2.5: Use-Case Diagram

Use Cases:

1. Vaccination center
2. Test Lab
3. Precautions
4. Symptoms
5. News update

Table 4.2.1: Use Case Table - Vaccination center

Use Case ID	1
Use Case Name	Vaccination center
Actor	User
Pre-Condition	User must enter the pincode and select the date.
Post-Condition	Can view the vaccination center.

Table 4.2.2: Use Case Table - Vaccination center

Use Case ID	2
Use Case Name	Testing Lab
Actor	User
Pre-Condition	User must have their location on.
Post-Condition	Can view all the nearby Testing labs.

Table 4.2.3: Use Case Table - Precautions

Use Case ID	3
Use Case Name	Precautions
Actor	User
Pre-Condition	User must have good network connectivity.
Post-Condition	User can view the Precautions

Table 4.2.4: Use Case Table - Symptoms

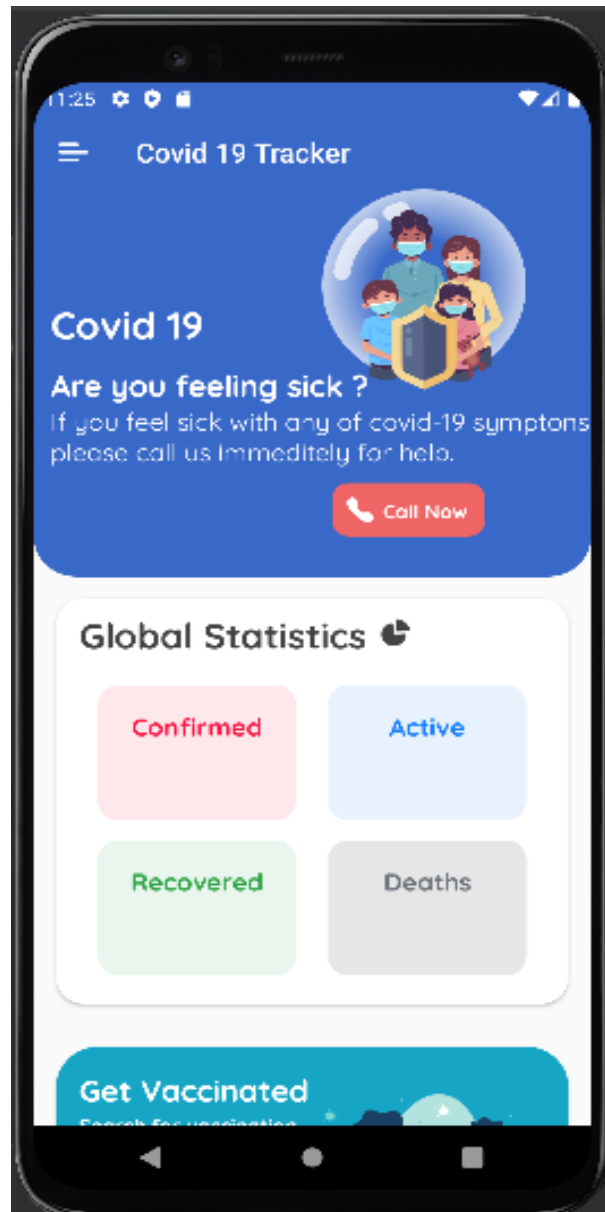
Use Case ID	4
Use Case Name	Symptoms
Actor	user
Pre-Condition	User must have good network connectivity.
Post-Condition	User can view the Symptoms.

Table 4.2.5: Use Case Table - News update

Use Case ID	5
Use Case Name	News update
Actor	User
Pre-Condition	User must have good network connectivity.
Post-Condition	User can view the News update.

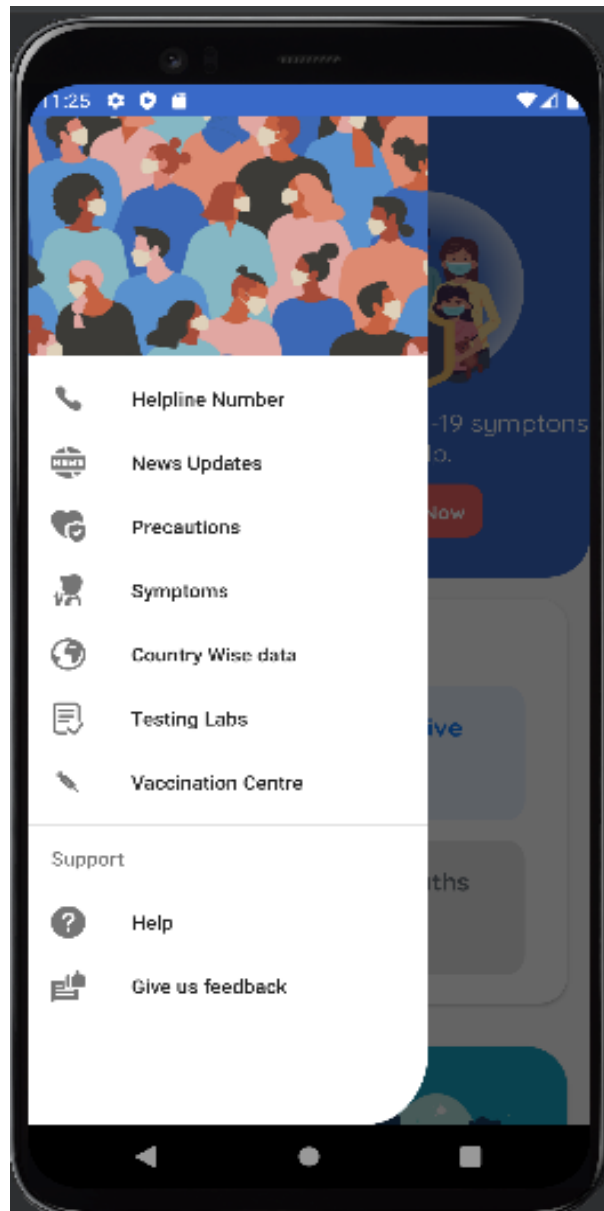
4 Project Implementation and Testing

4.1 Home page



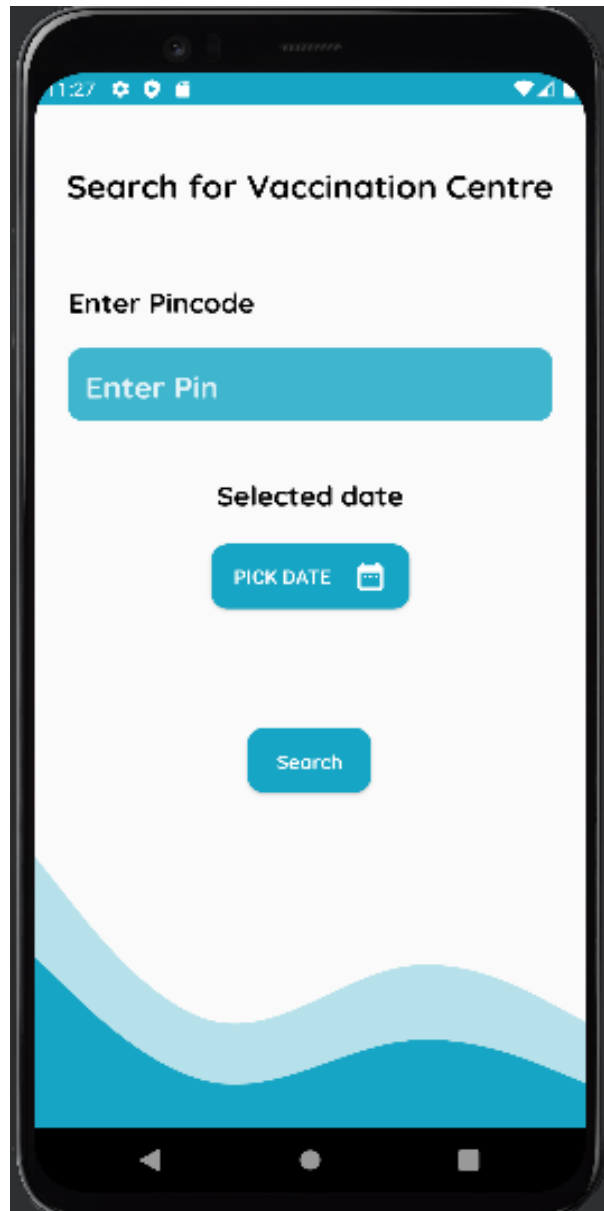
4.1.1: Home page

4.2 Menu



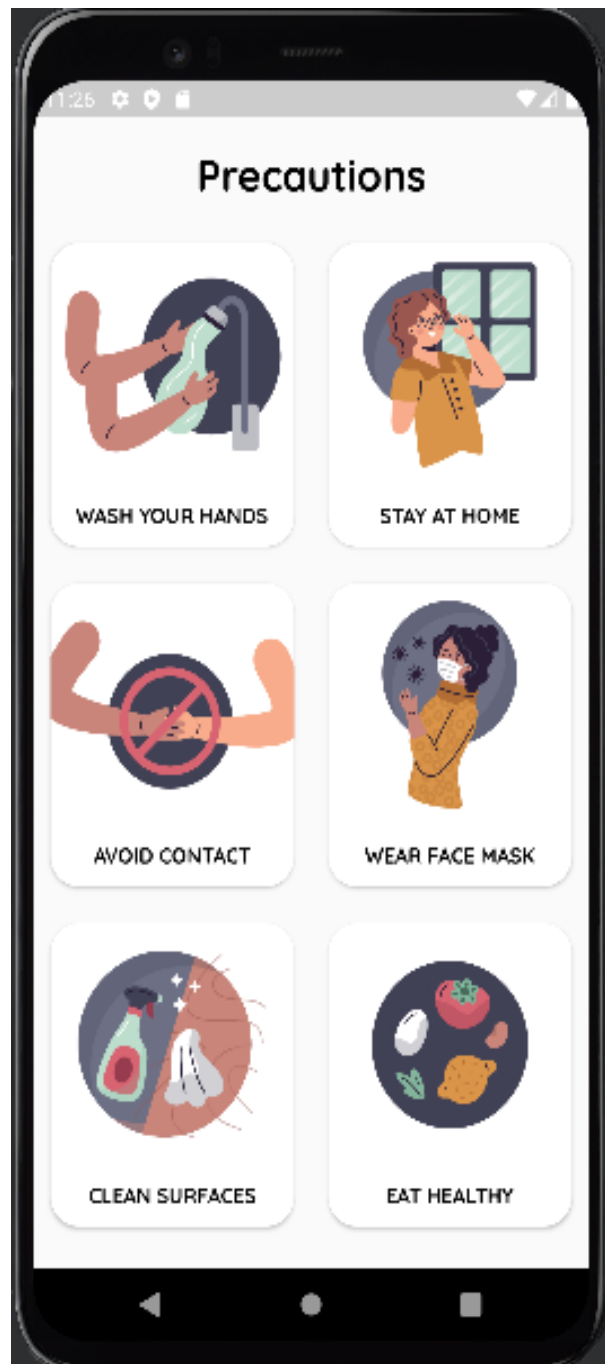
4.2.1: Menu

4.3 Vaccination center



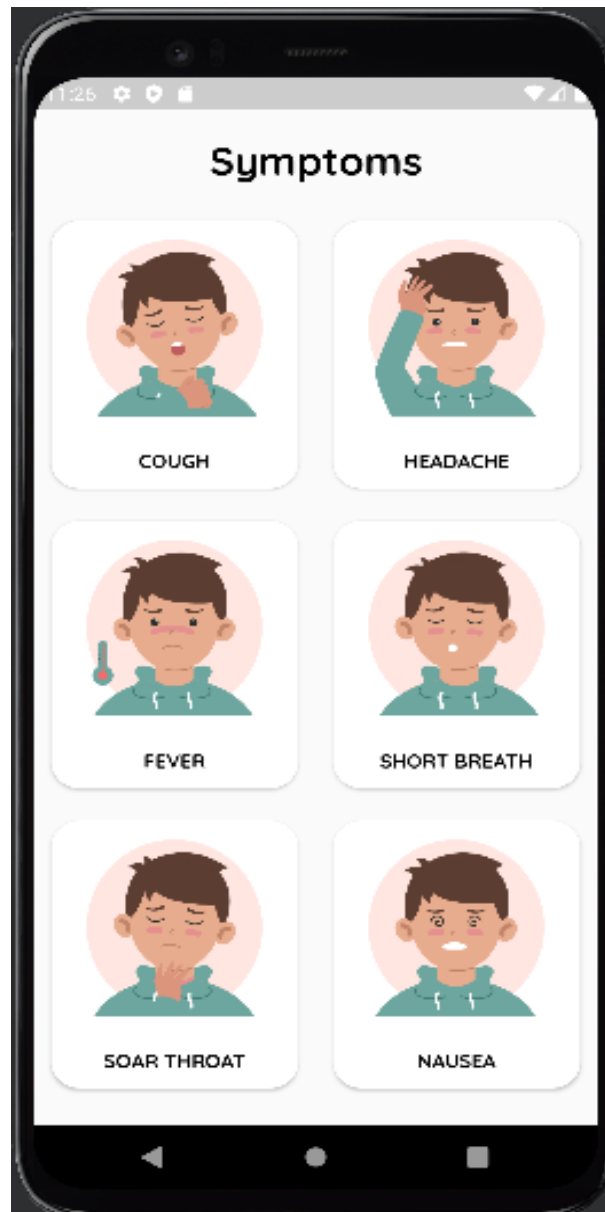
4.3.1: Vaccination centers

4.4 Precautions



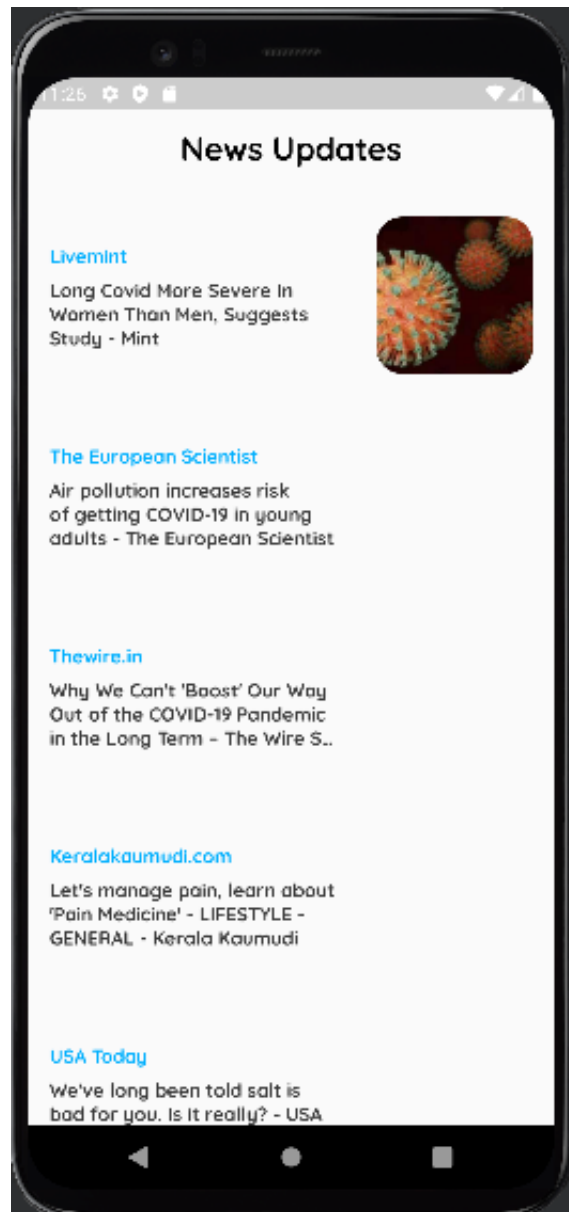
4.4.1: Precautions

4.5 Symptoms



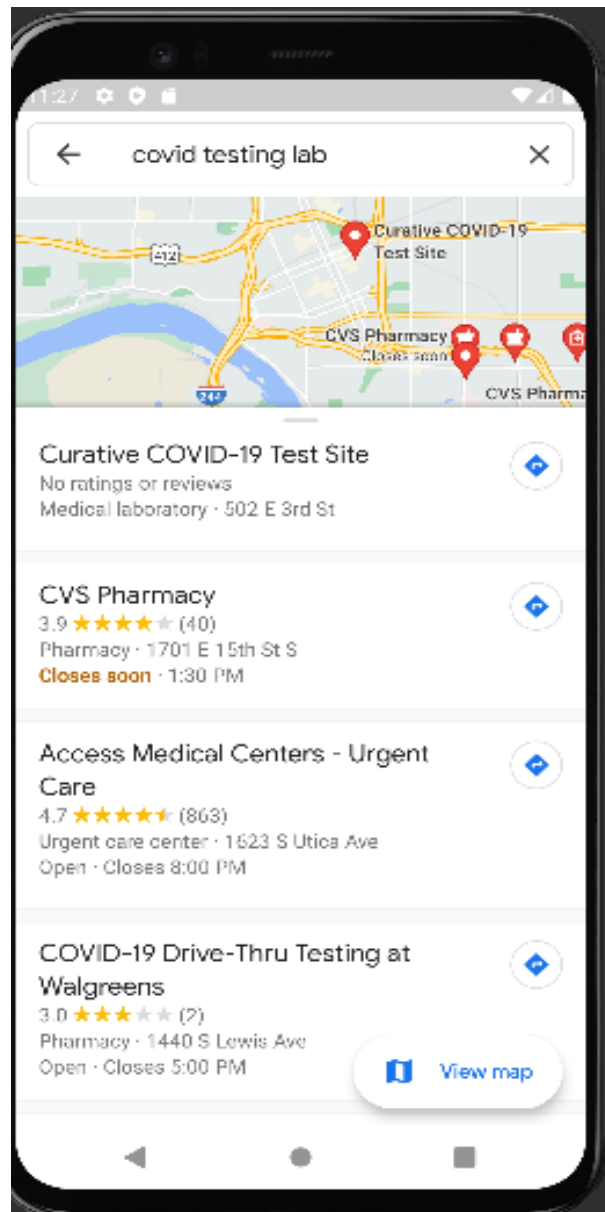
4.5.1: Symptoms

4.6 News updates



4.6.1: News updates

4.7 Testing Lab



4.7.1: Testing Lab

4.8 Code 1

```
package com.example.covid19tracker;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.widget.Toolbar;
import androidx.browser.customtabs.CustomTabsIntent;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
```

4.9 Code 2

```
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {

    Toolbar toolbar;
    Button call_btn;
    Button news_update_btn, precautions_btn, symptoms_btn, countryWise_data_btn, test_labs, search_vaccine_btn;
    TextView total, active, recovered, deaths;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        toolbar = findViewById(R.id.toolbar);
        call_btn = findViewById(R.id.call_button);
        news_update_btn = findViewById(R.id.news_update_btn);
        precautions_btn = findViewById(R.id.precautions_btn);
        symptoms_btn = findViewById(R.id.symptoms_btn);
        countryWise_data_btn = findViewById(R.id.countryWise_data_btn);
        test_labs = findViewById(R.id.testLabs_btn);
        search_vaccine_btn = findViewById(R.id.search_vaccine_btn);

        total = findViewById(R.id.tv_total);
        active = findViewById(R.id.tv_active);
        recovered = findViewById(R.id.tv_recovered);
        deaths = findViewById(R.id.tv_death);

        setSupportActionBar(toolbar);

        if (isNetworkAvailable()) {
```

4.10 Code 3

```
search_vaccine_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(MainActivity.this, VaccinationSearchActivity.class));
    }
});

news_update_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, NewsActivity.class);
        startActivity(intent);
    }
});

precautions_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, FragmentActivity.class);
        intent.putExtra("frag", "precautions");
        startActivity(intent);
    }
});

symptoms_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, FragmentActivity.class);
        intent.putExtra("frag", "symptoms");
        startActivity(intent);
    }
});
```

4.11 Code 4

```
test_labs.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Uri gmmIntentUri = Uri.parse("geo:0,0?q=covid testing lab");
        Intent intent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
        intent.setPackage("com.google.android.apps.maps");
        startActivity(intent);
    }
});

NavigationView navigationView = findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
}
//No Internet Activity is started if the network is not available
else {

    Intent intent = new Intent(this, NoInternet.class);
    startActivity(intent);
    finish();
}

}

@Override
public void onBackPressed() {
    DrawerLayout drawerLayout = findViewById(R.id.drawer_layout);
    if (drawerLayout.isDrawerOpen(GravityCompat.START)) {
        drawerLayout.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}
```

4.12 Code 5

```
String recovered_json = jsonObject.getString("recovered");
long recoveries = Long.parseLong(recovered_json);
recovered.setText(NumberFormat.getNumberInstance(Locale.US).format(recoveries));

//
//      total.setText(jsonObject.getString("cases"));
//      deaths.setText(jsonObject.getString("deaths"));
//      active.setText(jsonObject.getString("active"));
//      recovered.setText(jsonObject.getString("recovered"));

    } catch (JSONException e) {
        e.printStackTrace();
    }
}, error -> Toast.makeText(MainActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show());

}

/**
 * To check the Network Availability on device
 *
 * @return true or false based on network connectivity
 */
private boolean isNetworkAvailable() {
    ConnectivityManager connectivityManager
        = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetworkInfo = connectivityManager.getActiveNetworkInfo();
    return activeNetworkInfo != null && activeNetworkInfo.isConnected();
}
```

5 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	Helpline Number	User Clicks on Helpline number	Call the number	Called successfully	Pass
2	Helpline Number	User Clicks on Helpline number	Call the number	Call failed	Fail
3	Search for vaccine	User enters right pin and date	Display available slots	Slots displayed successfully.	Pass
4	Search for vaccine	User enters wrong pin and date	Enter correct pin and date	Slots display unsuccessful.	Fail
4	State wise data	User enters correct state	Displays correct stats	Stats displayed successfully	Pass

6 Limitations

- Requires Internet connection.
- No Support for iOS and MacOS Application.
- Fetching the API Data takes more time.

7 Future Enhancements

- Graphs can be made out of the data fetched from the API.
- Reports can be maintain by keeping track on users health and the data can be stored in database.
- Registration and login functionality can be added and connected to the database to keep track on individual's activity. The data can be stored using the database.
- Important notifications can be sent through mails to the Users.

8 System maintainance

- System Maintenance is needed when a new version of Codet is released.
- System Maintenance will be carried out at least quarterly to check whether everything is working fine.
- Entire System Maintenance work is also carried out when the system fails to work properly.
- Application will require maintenance when there is a major release or change of functionality if some new functionality is added.
- Maintenance is also required when this application fails to run in a certain environment, then proper debug and update will be released so as to increase compatibility.
- Symptoms and precautions needs to be update with the change in the research with respective to the new covid variants.

9 Bibliography

9.1 Web References

- [1.] <https://www.cowin.gov.in/>
- [2.] <https://developer.android.com/docs>
- [3.] <https://www.java.com/en/>
- [4.] <https://www.google.com/maps>
- [5.] <https://www.draw.io/>
- [6.] https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm