

MINI PROJECT REPORT

ON

# MEDICATION REMINDER

Submitted by

***Gangavarapu Hemanth***

***Inakota Jeshwanth***

***Bhoopathi Sai Sagar***

***in partial fulfillment for the award of the degree of***

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

**PRATHYUSHA ENGINEERING COLLEGE**

**THIRUVALLUR-602 025**

**ANNA UNIVERSITY :: CHENNAI 600 025 ANNA UNIVERSITY :: CHENNAI 600 025**

###### BONAFIDE CERTIFICATE

Certified that this project report “**MEDICATION REMINDER**” is the bonafide work of the “**GANGAVARAPU HEMANTH (111420104027) , INAKOTA JESHWANTH (111420104035), BHOOPATHI SAISAGAR (111420104010)** ” who carried out the project work under my supervision.

|  |  |
| --- | --- |
| SIGNATURE | SIGNATURE |
| **Dr. Thamba Meshach W,** | **Ms.SRI PRIYA N,** |
| PROFESSOR, | ASSISTANT PROFESSOR, |
| HEAD OF THE DEPARTMENT, | SUPERVISOR, |
| Department of CSE, | Department of CSE , |
| Prathyusha Engineering college, | Prathyusha Engineering College |
| Thiruvallur- 602 025. | Thiruvallur- 602 025. |

**Place:** Thiruvallur

**Date:**

Submitted for the Project Viva-Voce held on…………..…. at Prathyusha Engineering College, Thiruvallur- 602 025.

**INTERNALEXAMINER EXTERNALEXAMINER**

#### ABSTRACT

This is an Android-based application that incorporates an automated alarm ringing mechanism. Patients do not need to remember when to take their medications because they may set an alarm for such times. The alert may be set for a variety of medications and timings, as well as for the date, time, and medication description. They can search doctor disease wise. The patients will get the contact details of doctors as per their availability. They will receive a notification through internal system message, ideally chosen by the patients. Yet, in our work, we have made an effort to design a system that is affordable, time-saving, and improves medication adherence. Several of these medical reminder systems have been built where new hardware is necessary.

**KEYWORDS**

* Automatic Alarm
* Reminder System
* Notification System
* Medicine Scheduler
* Finding Nearest Hospitals
* Emergency Calls
* Symptom Checker

TABLE OF CONTENTS

**Chapter Contents Page No**

Abstract 3

Keywords 4

1. Introduction 6
2. System Analysis 10
   1. Existing System 11
   2. Proposed System 12
3. System Requirements 13
   1. Hardware Requirements 14
   2. Software Requirements 14

3.2.1 Software Description 15

1. System Implementation 22
   1. Module Description 23
2. Result 29
3. Conclusion And Future Enhancement 31
   1. Conclusion 31
   2. Future Enhancement 32
4. Appendix 34
   1. Code 35
   2. Screenshots 50

**CHAPTER 1**

INTRODUCTION

##### 1 INTRODUCTION

All people fall within the group of patients, including teachers, students, businesspeople, housewives, and children. We all have demanding schedules as well. Life nowadays is stressful and loaded with duties. People are therefore susceptible to a variety of ailments, therefore it is our responsibility to maintain our own health and fitness. If the patient stays at home, someone may help care for him or her, but if the patient travels or lives in another city or state away from home, it is difficult for family members to contact and constantly remind them to take their medication as prescribed. We completely rely on technologies, especially smartphones, in our ever growing and technologically reliant lives. Through navigation they can search doctors and hospitals and contact details so that they can easily get proper treatment on time. The patients can search doctor disease wise and area wise which will provide easy searching facility along with doctor’s contact information, visiting place and availability time.

Nowadays, every person owns a smartphone. This gives us the chance to utilise technology more effectively so that it can benefit us. Also, it has a significant impact on our everyday lives and aids in a variety of ways with our fitness. The striking issue is that patients frequently forget to take the right medications in the right amounts at the right times. Medication adherence, which is the amount to which a patient follows a doctor's prescription and takes the appropriate drug at the appropriate time, has recently become a significant problem because several research have found that non-adherence can seriously harm the patient, increasing the expense of healthcare. Medication nonadherence is a frequent, expensive, and complex issue that depletes healthcare resources and leads to subpar treatment results.

So, in order to help patients maintain their health and fitness, we are launching an Android application whose goal is to remind patients of their dose timings using Alarm Ringing system.

This application targets those who neglect to take their medications on schedule. Together with the sections for the date, time, and drug description, it allows users to create an alarm, enabling them to set alarms for several medications at various intervals. After setting an alarm, the notification system will send a notice. The user has the option to turn on or off the notification as needed. It will be transmitted as the user-selected message.

Reminders for taking medications aid in reducing drug mistakes and incorrect doses. Eclipse is used to design the application. It can promote knowledge of health care and be useful in the defence sector and in emergency situations (accidents). It is a simple-to-use tool that has a fantastic user experience and may save your life, money, and time.

# CHAPTER 2

SYSTEM ANALYSIS

###### 2 System Analysis

## 2.1 Existing System

Only available for iPhone users, MedsLog is a very complex app compared to others. To properly understand the software's functionalities, users must spend a lot more time with it. The system's "consumed by" box, which asks users to enter their username in the appropriate field, is the system's main flaw. The system still displays "no people." In contrast the proposed system is very much user friendly because it is made for the people of all ages. So one can utilise the time in using the system rather than wasting the time in understanding the software. The users' profiles are simple to manage. Because of issues with the reminder system and a service that backs up user data, MotioPHR Health Record Manager, which is available for $10 for the full version on Android and iPhone and $2 for a Lite version on iPhone, receives lower ratings from users. Medsy is also an application which tries to provide medicine remaindering system but it is loaded with less features.

Disadvantage

* + - Inaccuracy or unreliability of reminder systems, which can result in missed or incorrect doses
    - Lack of flexibility in medication schedules, as some reminder systems may not be able to accommodate changes or adjustments to medication regimens.

## 2.2 Proposed System

The suggested system, which is based on the Android operating system, will alert users when it's time to take their medications and will sound an alarm automatically if they forget. Google and the Open Handset Alliance collaborated to create Android, a Linux-based operating system primarily intended for touch-screen mobile devices like smartphones and tablet computers. Android was created from the bottom up to provide developers the tools they need to make engaging mobile applications that make the most of a device's capabilities. The system solely uses the Android operating system due of its substantial market dominance.

# CHAPTER 3

SYSTEM REQUIREMENTS

**3 System Requirements**

The system requirements involve hardware and software requirements.

## 3.1 Hardware Requirements

* + - Android mobile device with version 4.4 or above.
    - Minimum of 2GB RAM.
    - Minimum of 16GB internal storage.

## 3.2 Software Requirements

**Languages Used :**

* Android Studio IDE.
* Firebase Console.
* Java programming language.
* Google Maps API

### 3.2.1 Software Description

Software Description is a technical specification of requirements of software products. This specifies the environment for development, operation and maintenance of the product.

## Android studio IDE

Android Studio is an Integrated Development Environment (IDE) used for developing Android applications. It provides a comprehensive set of tools for building, debugging, and testing Android apps. Android Studio is the primary development tool used in the development of the Attendance Android app.

Android Studio provides a user-friendly interface that simplifies the development process. It has a rich set of features, including a code editor, a layout editor, and a visual designer, which enable developers to design and develop applications quickly and efficiently. The code editor in Android Studio provides features such as syntax highlighting, code completion, and error highlighting, which enable developers to write high-quality code with ease. The layout editor allows developers to design the app's user interface visually, and the visual designer allows developers to preview the app's design in real-time. Android Studio integrates with various tools and libraries, such as Firebase, which simplifies the development of apps that use cloud services. The Attendance Android app uses Firebase as the database and authentication provider, and Android Studio makes it easy to integrate these services into the app. Overall, Android Studio is a powerful and versatile tool that has played a critical role in the development of the Attendance Android app. Its comprehensive set of features and integration with various tools and libraries have helped in simplifying the development process, resulting in a stable, efficient, and user-friendly app.

**Java programming language**

Java is a widely-used programming language for developing mobile applications, and it is the language used in the development of the Attendance Android app. Java is known for its versatility, stability, and reliability, and it is one of the most popular programming languages for Android development. Java's object-oriented approach enables developers to build complex applications by breaking them down into smaller, reusable modules. This approach promotes code reusability, making it easier to maintain and update the codebase. In the Attendance Android app, Java is used to implement the logic for various features, such as authentication, database operations, and UI design. The app leverages various Java libraries and frameworks, such as Firebase Authentication, Firebase Firestore, and Android Studio, to simplify the development process and improve the app's performance. Overall, Java is a powerful and flexible programming language that is well-suited for mobile application development, and its use in the development of the Attendance Android app has resulted in a stable, reliable, and efficient application.

**Advantages:**

* Platform independent
* Object-Oriented Programming
* Rich Set of Libraries
* Garbage Collection
* Security

**Firebase console**

Firebase Console is a web-based platform provided by Google to manage Firebase projects. Firebase is a mobile and web application development platform that provides a set of tools and services for building and managing mobile and web applications. The Attendance Android app uses Firebase as the backend database for storing user details, attendance records, and notifications.

Firebase Console provides a graphical user interface for managing Firebase projects. It allows developers to create and manage Firebase projects, add Firebase services, and configure project settings. Firebase Console also provides a real-time database for storing and retrieving data in real-time.

The Attendance Android app uses Firebase Authentication for user authentication, Firebase Firestore for storing user details, attendance records, and notifications, and Firebase Cloud Messaging for sending notifications to users. Firebase Console is used to manage these Firebase services.

In Firebase Console, the developer can manage the database, add, edit or delete records, view database statistics, and configure database rules. The developer can also manage authentication, create new users, and view user details.

Overall, Firebase Console provides a user-friendly interface for managing Firebase services, making it easier for developers to build and manage mobile and web applications. The Attendance Android app leverages Firebase Console to manage Firebase services, allowing the developer to easily add, configure, and manage Firebase services without writing any server-side code.

**Advantages:**

* + - * **Easy to use:** Firebase Console provides a user-friendly interface for managing Firebase projects. It allows developers to easily configure and manage Firebase services without writing any server-side code.
      * **Real-time database:** Firebase Console provides a real-time database for storing and retrieving data in real-time. This allows developers to build real-time applications with ease.
      * **Scalable:** Firebase Console is built on Google Cloud Platform, which provides scalable and reliable infrastructure for building and managing mobile and web applications.
      * **Cross-platform:** Firebase Console supports multiple platforms, including Android, iOS, and web. This allows developers to build applications for multiple platforms using a single backend.
      * **Analytics:** Firebase Console provides analytics features that allow developers to track user engagement, retention, and conversion rates.
      * **Notifications:** Firebase Console provides Firebase Cloud Messaging (FCM), which allows developers to send push notifications to users across multiple platforms.

**Google Maps API :**

The google maps places API is used in our app to search for nearby hospitals and doctors. This API provides a simple way to search for places of interest based on specific keywords, such as “hospital” or “doctor”, as well as filter results based on various criteria, such as distance and rating. The API returns a list of places matching search criteria, along with details such as location, opening hours, and user reviews.

Our app also uses the google maps geocoding API to convert addresses into geographic coordinates that can be used to display location data on a map. This API provides a way to retrieve accurate and up-to-date information about specific locations, such as the latitude and longitude coordinatyes, as well as the type of location and other details.

Overall, the google maps API provides a powerful set of tools that enable developers to build location-aware applications that are highly useful and engaging for users.

# CHAPTER 4

# SYSTEM IMPLEMENTATION

**4 SYSTEM IMPLEMENTATION**

LIST OF MODULES

* Medication alarm
* Finding nearest hospitals by using GPS
* Finding nearest doctors by using GPS
* Symptom checker

**4.1 MODULE DESCRIPTION**

**1) Medication alarm**

A medication alarm is a device that helps people remember to take their medication on time. It is a simple yet effective tool that can be used by individuals who need to take medication at specific times of the day or night.

There are different types of medication alarms available in the market. Some are standalone devices that can be programmed to sound an alarm at a specific time of the day, while others are smartphone apps that can be downloaded and used on mobile devices. Some medication alarms can also be integrated into a larger healthcare system, such as a hospital or clinic.

However, it is important to note that medication alarms should not replace the advice of a healthcare professional. They are intended to supplement medication management and support, but ultimately, it is up to the individual to take their medication as prescribed by their doctor.

**2) Finding nearest hospitals by using GPS**

Finding the nearest hospitals using GPS (Global Positioning System) is a convenient and easy way to locate medical facilities in your area. GPS is a satellite-based navigation system that uses signals from multiple satellites to determine the location of a device or user.

To find the nearest hospitals using GPS, you will need a device that has GPS capabilities, such as a smartphone or a GPS device. Many mapping applications, such as Google Maps, Apple Maps, and Waze, offer this functionality.

Here are the steps to find the nearest hospitals using GPS:

Open the mapping application on your device.

Tap on the search bar and type in "hospitals" or "medical centers."

The mapping application will use your GPS location to show you the hospitals in your immediate area.

You can also use the "directions" feature to get turn-by-turn directions to the hospital of your choice.

In addition to finding the nearest hospitals, GPS technology can also be used to provide real-time traffic updates and estimated arrival times. This can be especially helpful when trying to reach a hospital quickly in an emergency situation.

It is important to note that while GPS technology can be a useful tool in locating medical facilities, it is not a substitute for emergency medical services. In an emergency, it is always best to call 911 or your local emergency number for immediate assistance.

**3) Finding nearest doctors by using GPS**

Finding the nearest doctors using GPS (Global Positioning System) is a convenient way to locate medical professionals in your area. GPS is a satellite-based navigation system that uses signals from multiple satellites to determine the location of a device or user.

To find the nearest doctors using GPS, you will need a device that has GPS capabilities, such as a smartphone or a GPS device. There are many medical applications available that offer this functionality, such as Zocdoc, Healthgrades, and WebMD.

Here are the steps to find the nearest doctors using GPS:

1. Open the medical application on your device.

2. Tap on the search bar and type in the type of doctor you are looking for, such as a family doctor, dentist, or specialist.

3. The medical application will use your GPS location to show you the doctors in your immediate area.

4. You can filter the results by location, rating, availability, and insurance coverage to find a doctor that meets your needs.

5. You can also use the "directions" feature to get turn-by-turn directions to the doctor's office.

In addition to finding the nearest doctors, GPS technology can also be used to provide real-time traffic updates and estimated arrival times. This can be especially helpful when trying to reach a doctor's appointment on time.It is important to note that while GPS technology can be a useful tool in locating medical professionals, it is not a substitute for medical advice or emergency medical services. In an emergency, it is always best to call 911 or your local emergency number for immediate assistance.

**4) Symptom checker**

A symptom checker is a tool that allows individuals to input their symptoms and receive a list of potential health conditions that could be causing those symptoms. It is typically an online service or a mobile application that uses a database of symptoms and medical conditions to provide users with an initial assessment of their health status.

Symptom checkers typically ask users a series of questions about their symptoms, such as when they started, how severe they are, and whether they are accompanied by other symptoms. Based on this information, the symptom checker will generate a list of potential health conditions that could be causing the symptoms.

Some symptom checkers also provide additional information about each condition, such as common causes, risk factors, and treatment options. This can be helpful for individuals who want to learn more about their symptoms and possible health conditions.

It is important to note that symptom checkers are not a substitute for professional medical advice or a medical diagnosis. They are designed to provide users with general information and guidance, and are not intended to replace the expertise of a healthcare professional. Individuals who are experiencing severe symptoms or have concerns about their health should always seek medical advice from a qualified healthcare provider.

**CHAPTER 5**

RESULTS

5 RESULTS

The app offers trustworthy reminders, a lovely user interface, pleasant interactions with the user, and many additional features that assist medication adherence. This demonstrated how people of all ages may benefit from the overall functionality offered. Individuals who are older are more prone to forget when to take their medications. the location based (Area wise) searching of the doctor was a good choice to the population ageing 58.The users will receive a schedule of when to take their medications, the beginning and finishing dates of their medications, message notifications, and an automatic alert system and navigation system.The technology also sends warnings when the phone's battery is low. It was determined that everyone in the population benefited from the automated alarm ringing function. As a result, the system as a whole performed admirably in our study and actually encourages medication compliance.

**CHAPTER 6**

CONCLUSION AND FUTURE ENHANCEMENTS

## 6.1 CONCLUSION

Across various platforms, a number of medication reminder systems have been created. Several of these systems call for specialised hardware to remind patients when to take their medications. It costs more money and takes more time and effort to buy new gear. Hence, an effort has been made in the provided work to establish a system that is affordable, accessible. Medication nonadherence decreases a treatment's efficacy and costs the health care system money. The patients will receive information on the timing of medication intake, the beginning and end dates of their medications, message notifications, and an automatic alert system.

6.2 FUTURE ENHANCEMENTS

We intend to concentrate on enhancing the system's overall performance. The patients will get the schedule of medicine in-take time with medicine description, starting and ending date of medicine, notification through message, automatic alarm ringing system and navigation system. We can extract the text in the medical prescription by the scanning the prescription by mobile camera.

**CHAPTER 7**

APPENDIX

# 7 APPENDIX

# 7.1 CODE

**Dashboard Front End**

package com.example.myapplication;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.pm.ApplicationInfo;

import android.net.Uri;

import android.os.Bundle;

import android.view.MenuItem;

import android.view.View;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBarDrawerToggle;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.cardview.widget.CardView;

import androidx.core.content.FileProvider;

import androidx.core.view.GravityCompat;

import androidx.drawerlayout.widget.DrawerLayout;

import com.google.android.gms.auth.api.signin.GoogleSignIn;

import com.google.android.gms.auth.api.signin.GoogleSignInClient;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.material.navigation.NavigationView;

import com.squareup.picasso.BuildConfig;

import java.io.File;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.nio.file.Files;

public class MainActivity extends AppCompatActivity {

Toolbar t;

TextView userNaame;

DrawerLayout drawer;

CardView remainder,symptom,sos,hospital,doctor;

GoogleSignInOptions gso;

NavigationView nav;

GoogleSignInClient gsc;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

drawer = findViewById(R.id.draw\_activity);

t=(Toolbar) findViewById(R.id.toolbar);

userNaame=findViewById(R.id.usermessage);

SharedPreferences sharedPreferences=getSharedPreferences("userInfo", Context.MODE\_PRIVATE);

String nameUser = sharedPreferences.getString("name", "");

userNaame.setText("Welcome " + nameUser);

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer, t, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);

drawer.addDrawerListener(toggle);

toggle.syncState();

nav = findViewById(R.id.nav\_view);

gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN).requestEmail().build();

gsc = GoogleSignIn.getClient(this, gso);

nav.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

switch (menuItem.getItemId()) {

case R.id.profile:

Intent in = new Intent(getApplicationContext(), profile.class);

startActivity(in);

break;

case R.id.share:

shareApplication();

break;

case R.id.alarmView:

startActivity(new Intent(MainActivity.this, AlarmView.class));

break;

case R.id.contact\_us:

startActivity(new Intent(MainActivity.this, contact\_us.class));

break;

case R.id.about\_us:

startActivity(new Intent(MainActivity.this, about\_us.class));

break;

case R.id.logout:

logout();

break;

}

drawer.closeDrawer(GravityCompat.START);

return true;

}

});

remainder = findViewById(R.id.reminder);

symptom = findViewById(R.id.symptom);

hospital = findViewById(R.id.hospital);

doctor = findViewById(R.id.listofdoctors);

sos = findViewById(R.id.sos);

remainder.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(MainActivity.this, alert\_medicine.class));

}

});

symptom.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(MainActivity.this, symptom\_checker.class));

}

});

hospital.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(MainActivity.this, nearest\_hospital.class));

}

});

doctor.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(MainActivity.this, nearest\_doctor.class));

}

});

sos.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(MainActivity.this, call.class));

}

});

}

private void logout() {

SharedPreferences sharedPreferences = getSharedPreferences("userInfo", Context.MODE\_PRIVATE);

sharedPreferences.edit().clear().apply();

// Start the login activity

Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);

startActivity(loginIntent);

// Close the main activity

finish();

}

private void shareApplication() {

ApplicationInfo app = getApplicationContext().getApplicationInfo();

String filePath = app.sourceDir;

String appLabel = getString(R.string.app\_name);

Intent intent = new Intent(Intent.ACTION\_SEND);

// MIME of .apk is "application/vnd.android.package-archive".

// but Bluetooth does not accept this. Let's use "\*/\*" instead.

intent.setType("\*/\*");

// Append file and send Intent

File originalApk;

originalApk = new File(filePath);

try {

//Make new directory in new location=

File tempFile = new File(getExternalCacheDir() + "/ExtractedApk");

//If directory doesn't exists create new

if (!tempFile.isDirectory())

if (!tempFile.mkdirs())

return;

//Get application's name and convert to lowercase

tempFile = new File(tempFile.getPath() + "/" +appLabel.replace(" ","").toLowerCase() + ".apk");

//If file doesn't exists create new

if (!tempFile.exists()) {

if (!tempFile.createNewFile()) {

return;

}

}

//Copy file to new location

InputStream in = null;

if (android.os.Build.VERSION.SDK\_INT >= android.os.Build.VERSION\_CODES.O) {

in = Files.newInputStream(originalApk.toPath());

}

OutputStream out = null;

if (android.os.Build.VERSION.SDK\_INT >= android.os.Build.VERSION\_CODES.O) {

out = Files.newOutputStream(tempFile.toPath());

}

byte[] buf = new byte[1024];

int len;

while ((len = in.read(buf)) > 0) {

out.write(buf, 0, len);

}

in.close();

out.close();

System.out.println("File copied.");

//Open share dialog

// intent.putExtra(Intent.EXTRA\_STREAM, Uri.fromFile(tempFile));

Uri photoURI = FileProvider.getUriForFile(this, BuildConfig.APPLICATION\_ID + ".provider", tempFile);

// intent.putExtra(Intent.EXTRA\_STREAM, Uri.fromFile(tempFile));

intent.putExtra(Intent.EXTRA\_STREAM, photoURI);

startActivity(Intent.createChooser(intent, "Share app via"));

} catch (IOException e) {

e.printStackTrace();

}

}

@Override

public void onBackPressed() {

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

alertDialogBuilder.setTitle("Exit Application?");

alertDialogBuilder

.setMessage("Click yes to exit!")

.setCancelable(false)

.setPositiveButton("Yes",

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

moveTaskToBack(true);

android.os.Process.killProcess(android.os.Process.myPid());

System.exit(1);

}

})

.setNegativeButton("No", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}

});

AlertDialog alertDialog = alertDialogBuilder.create();

alertDialog.show();

}

}

**Dashboard Back End**

<?xml version="1.0" encoding="utf-8"?>

<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/draw\_activity"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fitsSystemWindows="true"

android:theme="@style/Theme.AppCompat"

tools:context=".MainActivity"

tools:openDrawer="start"

android:background="#0b1f52">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<androidx.appcompat.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:elevation="5dp"

app:popupTheme="@style/Theme.AppCompat.NoActionBar"

tools:ignore="MissingConstraints">

</androidx.appcompat.widget.Toolbar>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/usermessage"

android:layout\_marginTop="8dp"

android:text="Hii Welcome"

android:textAlignment="center"

android:textSize="28sp"

android:textColor="@color/white"/>

</LinearLayout>

</RelativeLayout>

<GridLayout

android:layout\_weight="3"

android:id="@+id/Frame"

android:layout\_width="match\_parent"

android:columnCount="2"

android:rowCount="3"

android:background="@drawable/bg"

android:layout\_height="wrap\_content"

android:padding="15dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.0">

<androidx.cardview.widget.CardView

app:cardBackgroundColor="@color/white"

android:layout\_columnWeight="1"

android:id="@+id/reminder"

android:layout\_rowWeight="1"

app:cardCornerRadius="15dp"

android:layout\_margin="10dp"

android:elevation="4dp"

android:layout\_width="0dp"

android:outlineProvider="background"

android:layout\_height="0dp"

>

<LinearLayout

android:orientation="vertical"

android:weightSum="3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:layout\_margin="25dp"

android:src="@drawable/ic\_baseline\_add\_alert\_24"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="15dp"

android:textColor="@color/teal\_700"

android:text="Alert Reminder"

android:textSize="20dp"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

app:cardBackgroundColor="@color/white"

android:id="@+id/symptom"

android:layout\_columnWeight="1"

android:layout\_rowWeight="1"

app:cardCornerRadius="15dp"

android:layout\_margin="10dp"

android:elevation="4dp"

android:layout\_width="0dp"

android:layout\_height="0dp"

>

<LinearLayout

android:orientation="vertical"

android:weightSum="3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:layout\_margin="25dp"

android:src="@drawable/symptom\_check"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="15dp"

android:textColor="@color/teal\_700"

android:text="Symptom checker"

android:textSize="19dp"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

app:cardBackgroundColor="@color/white"

android:layout\_columnWeight="1"

android:id="@+id/hospital"

android:layout\_rowWeight="1"

app:cardCornerRadius="15dp"

android:layout\_margin="10dp"

android:elevation="4dp"

android:layout\_width="0dp"

android:layout\_height="0dp"

>

<LinearLayout

android:orientation="vertical"

android:weightSum="3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:layout\_margin="25dp"

android:src="@drawable/hospitalsearch"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="5dp"

android:textColor="@color/teal\_700"

android:text="Hospital Finder"

android:textSize="20dp"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/listofdoctors"

app:cardBackgroundColor="@color/white"

android:layout\_columnWeight="1"

android:layout\_rowWeight="1"

app:cardCornerRadius="15dp"

android:layout\_margin="10dp"

android:elevation="4dp"

android:layout\_width="0dp"

android:layout\_height="0dp"

>

<LinearLayout

android:orientation="vertical"

android:weightSum="3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="50dp"

android:layout\_height="55dp"

android:layout\_weight="2"

android:layout\_margin="25dp"

android:src="@drawable/doctors"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="15dp"

android:textColor="@color/teal\_700"

android:text="Doctor Finder"

android:textSize="20dp"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/sos"

app:cardBackgroundColor="@color/white"

android:layout\_columnWeight="1"

android:layout\_rowWeight="1"

app:cardCornerRadius="15dp"

android:layout\_margin="10dp"

android:elevation="4dp"

android:layout\_width="0dp"

android:layout\_height="0dp"

>

<LinearLayout

android:orientation="vertical"

android:weightSum="3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<ImageView

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_weight="2"

android:layout\_margin="25dp"

android:src="@drawable/ic\_baseline\_call\_24"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="5dp"

android:textColor="@color/teal\_700"

android:text="Emergency Call"

android:textSize="20dp"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

</GridLayout>

</LinearLayout>

<com.google.android.material.navigation.NavigationView

android:id="@+id/nav\_view"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

app:headerLayout="@layout/headerfile"

app:menu="@menu/sidemenu"

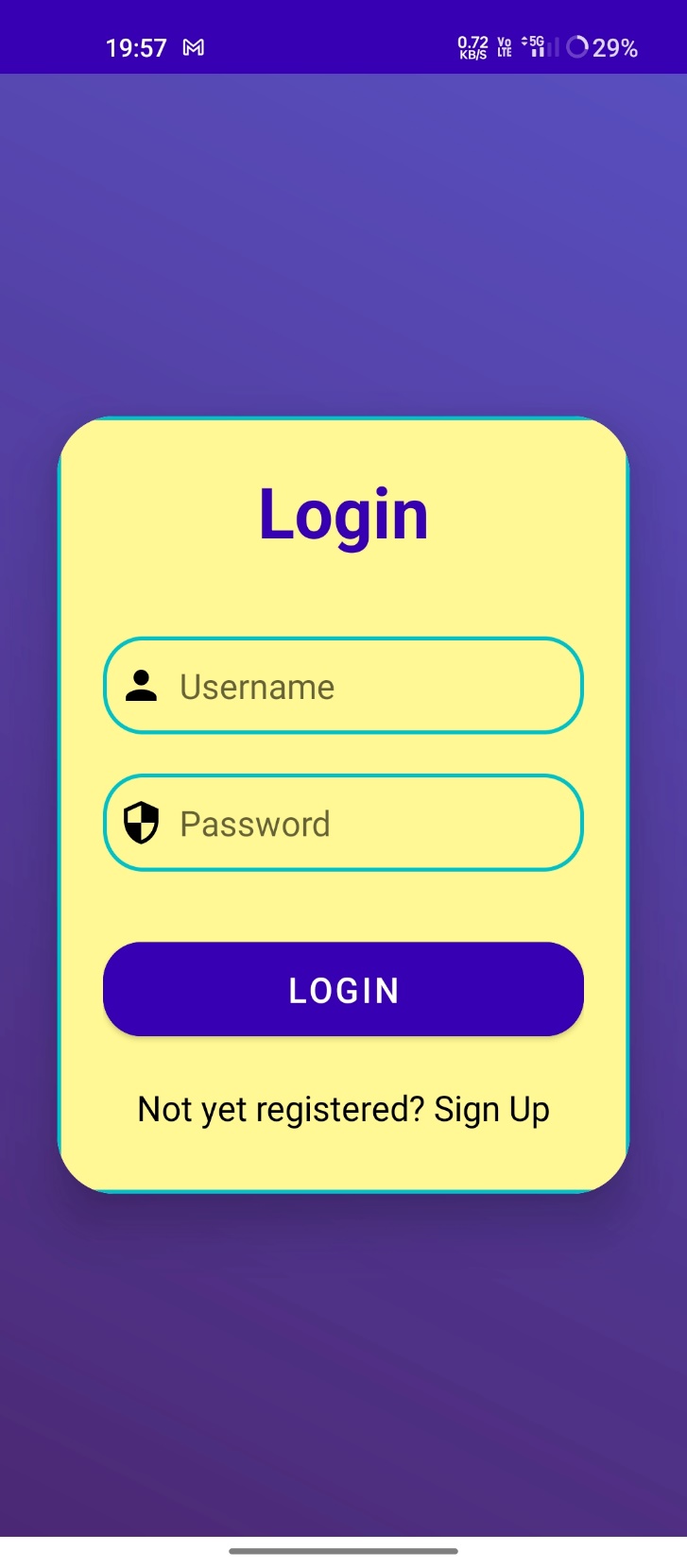
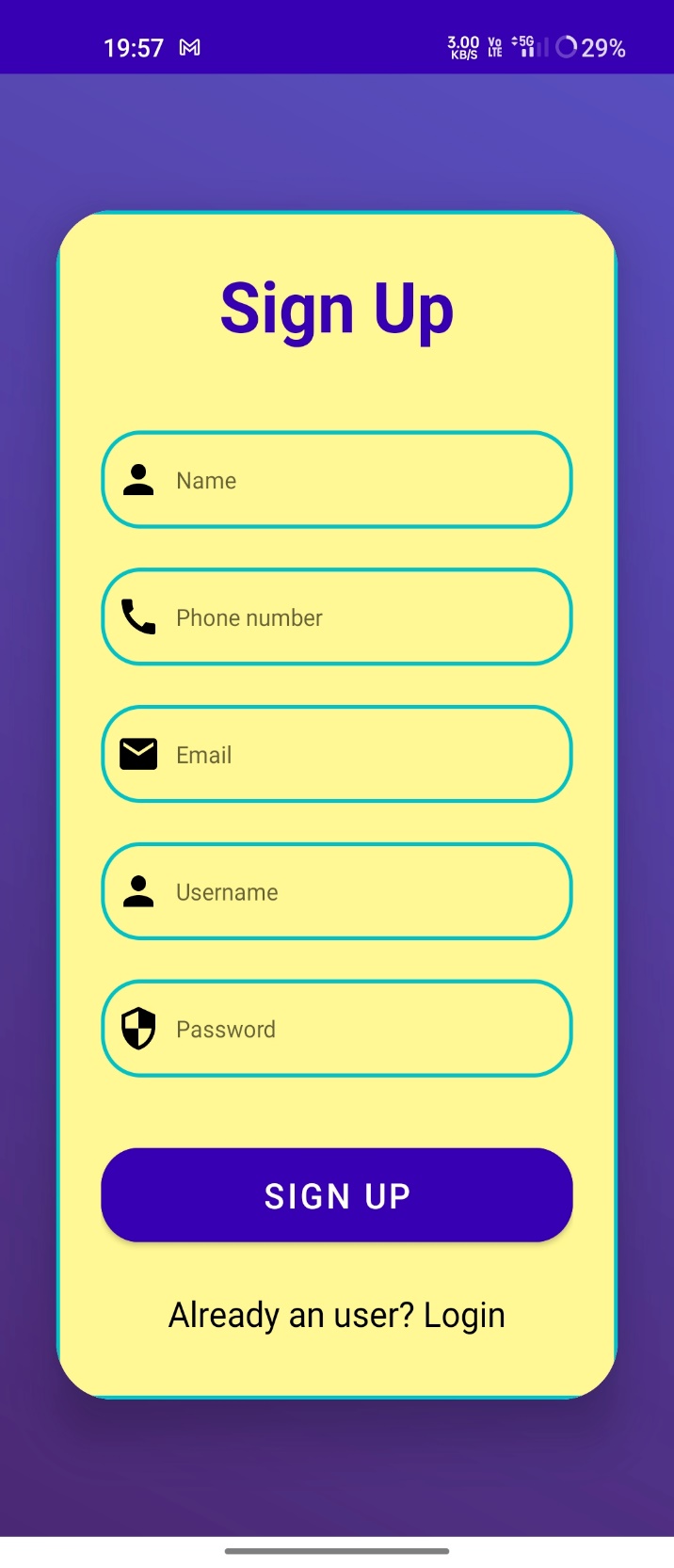
tools:ignore="MissingConstraints" />

</androidx.drawerlayout.widget.DrawerLayout>

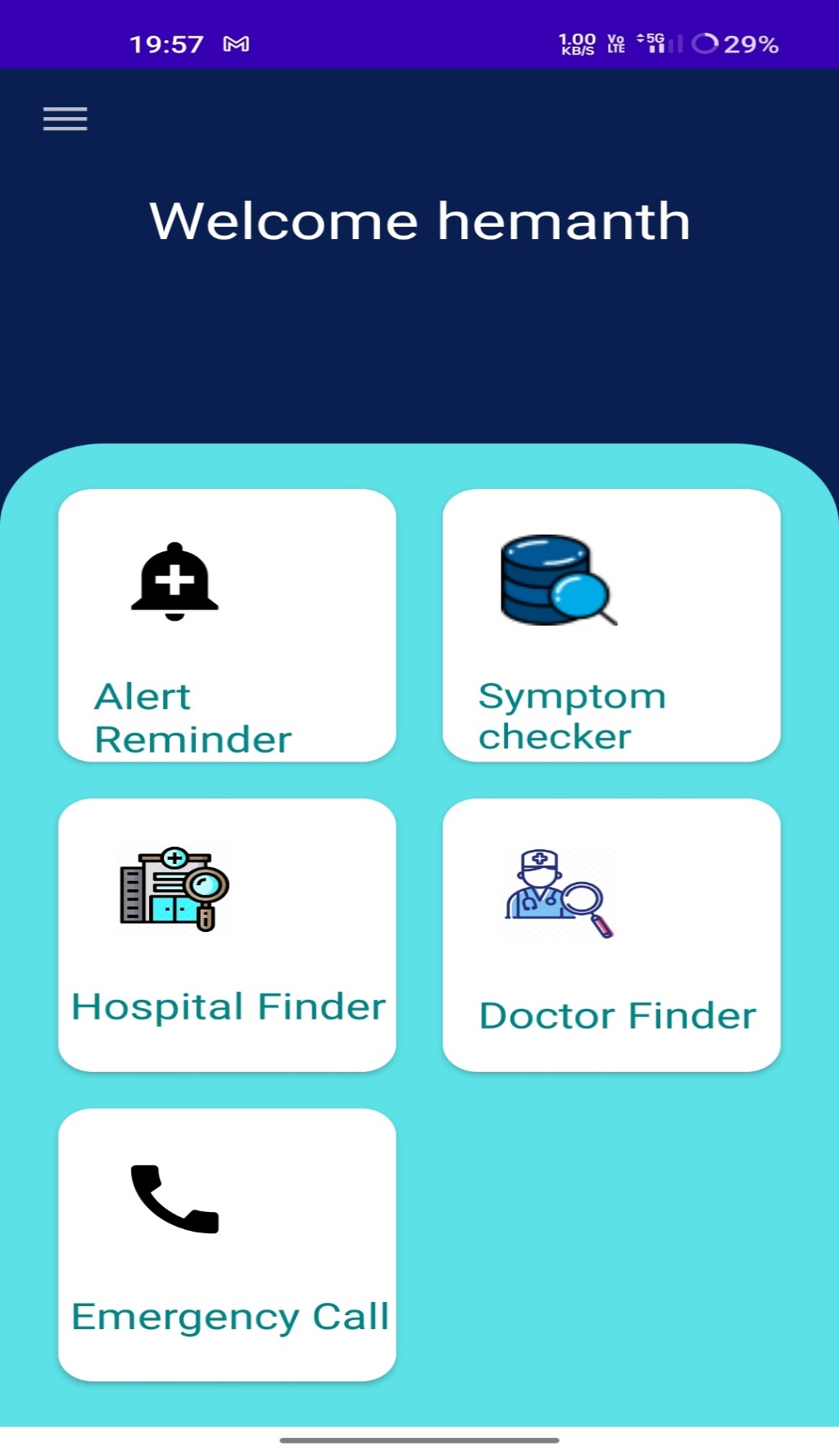
**THE REMINING CODE IS IN GITHUB**

GitHub link :

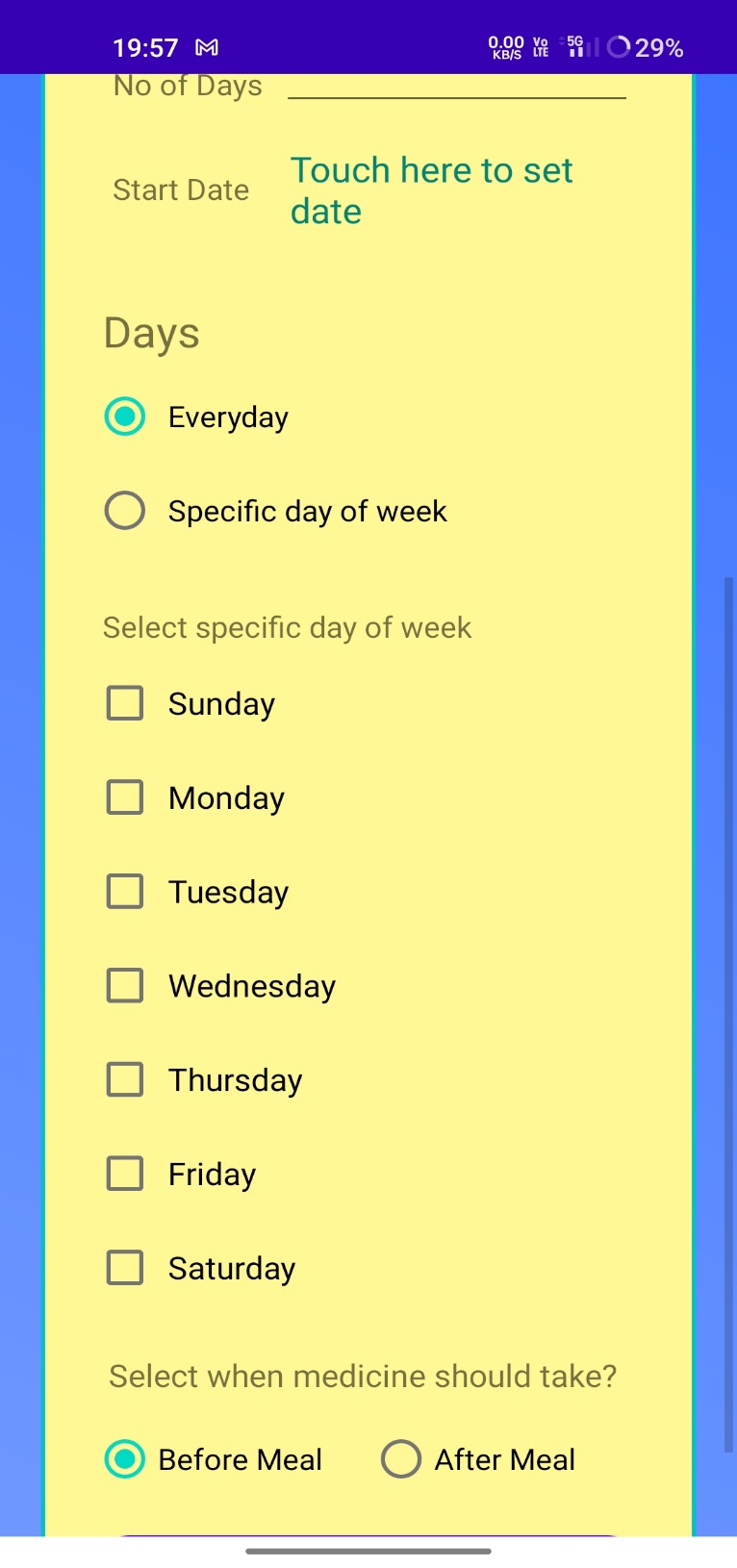
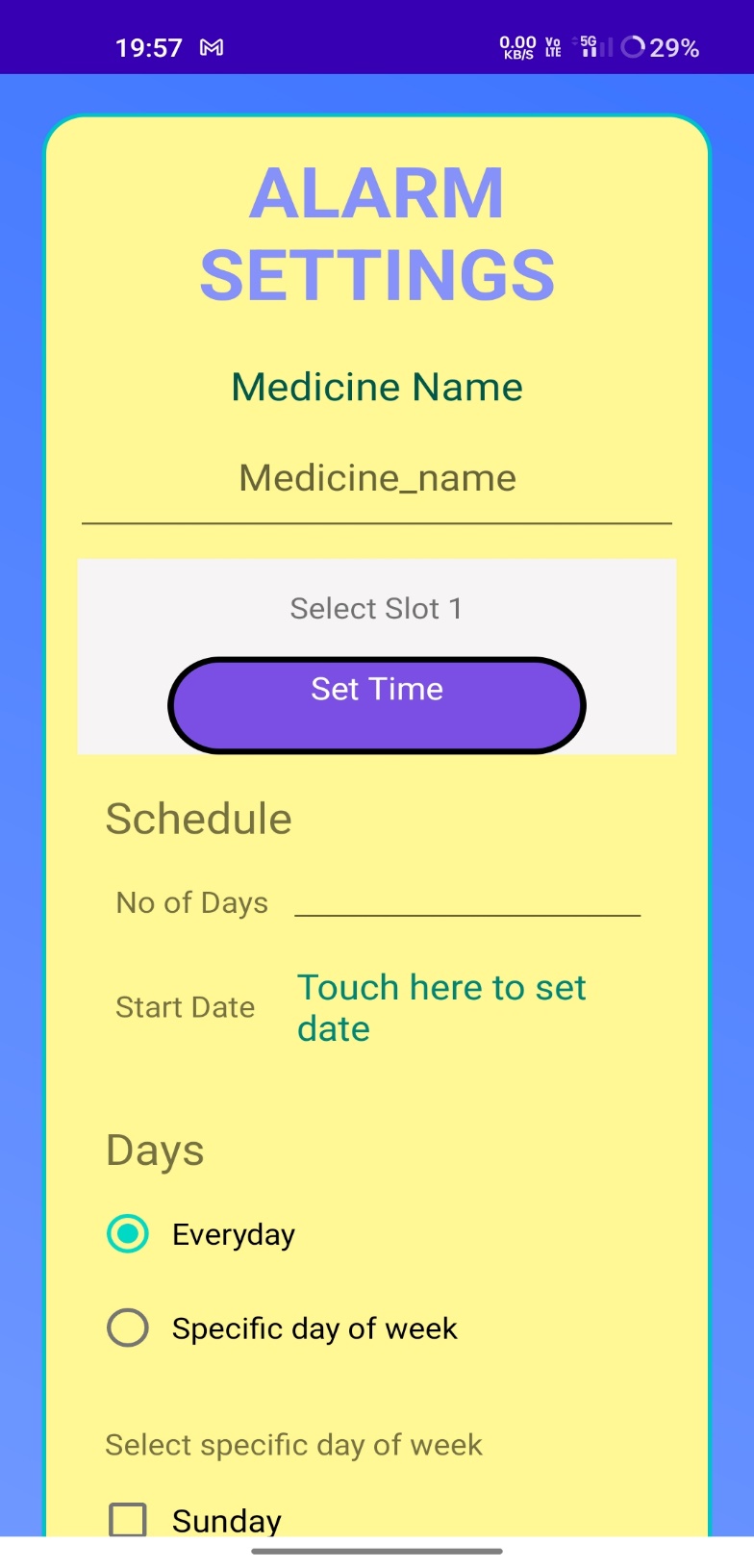
## 7.2 SCREENSHOTS



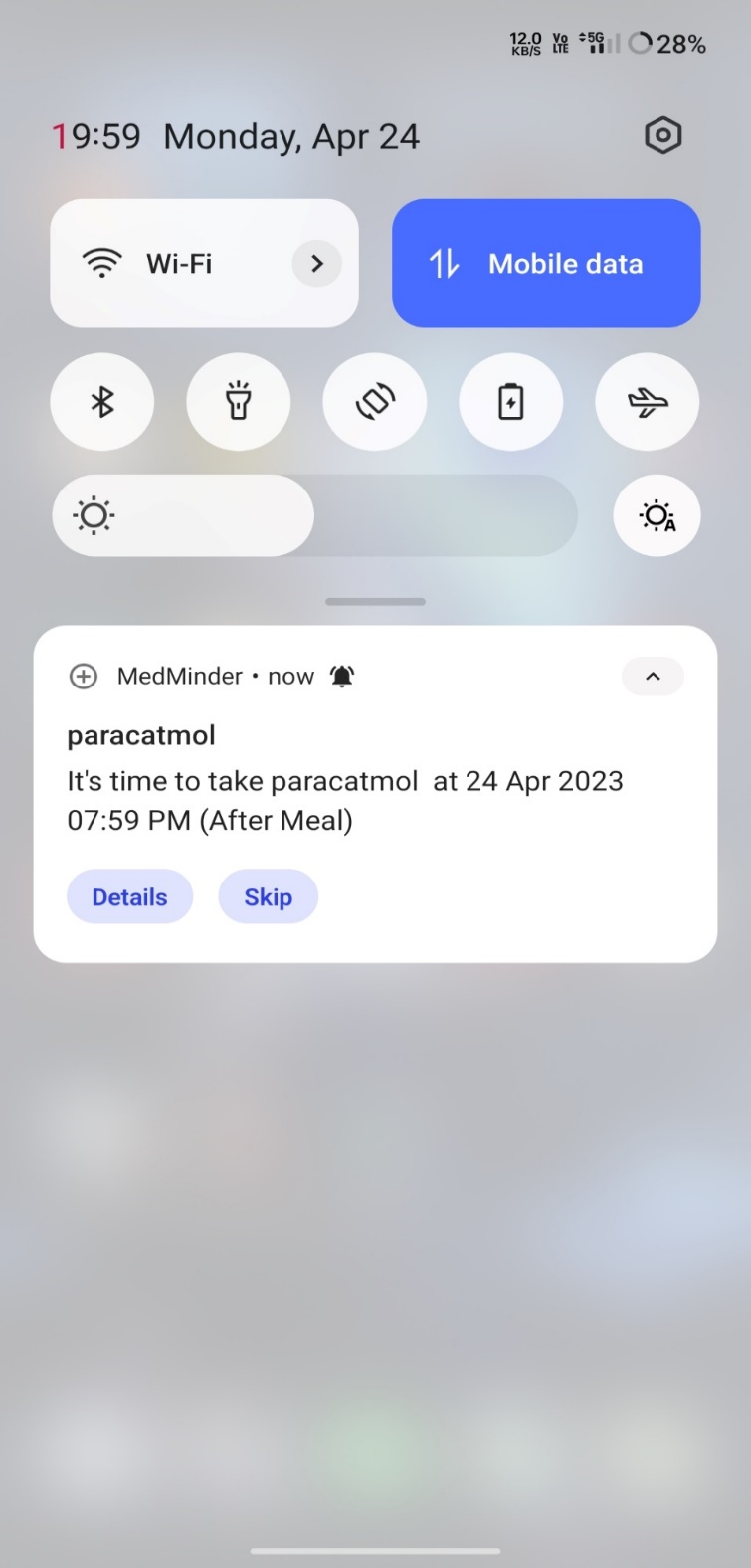
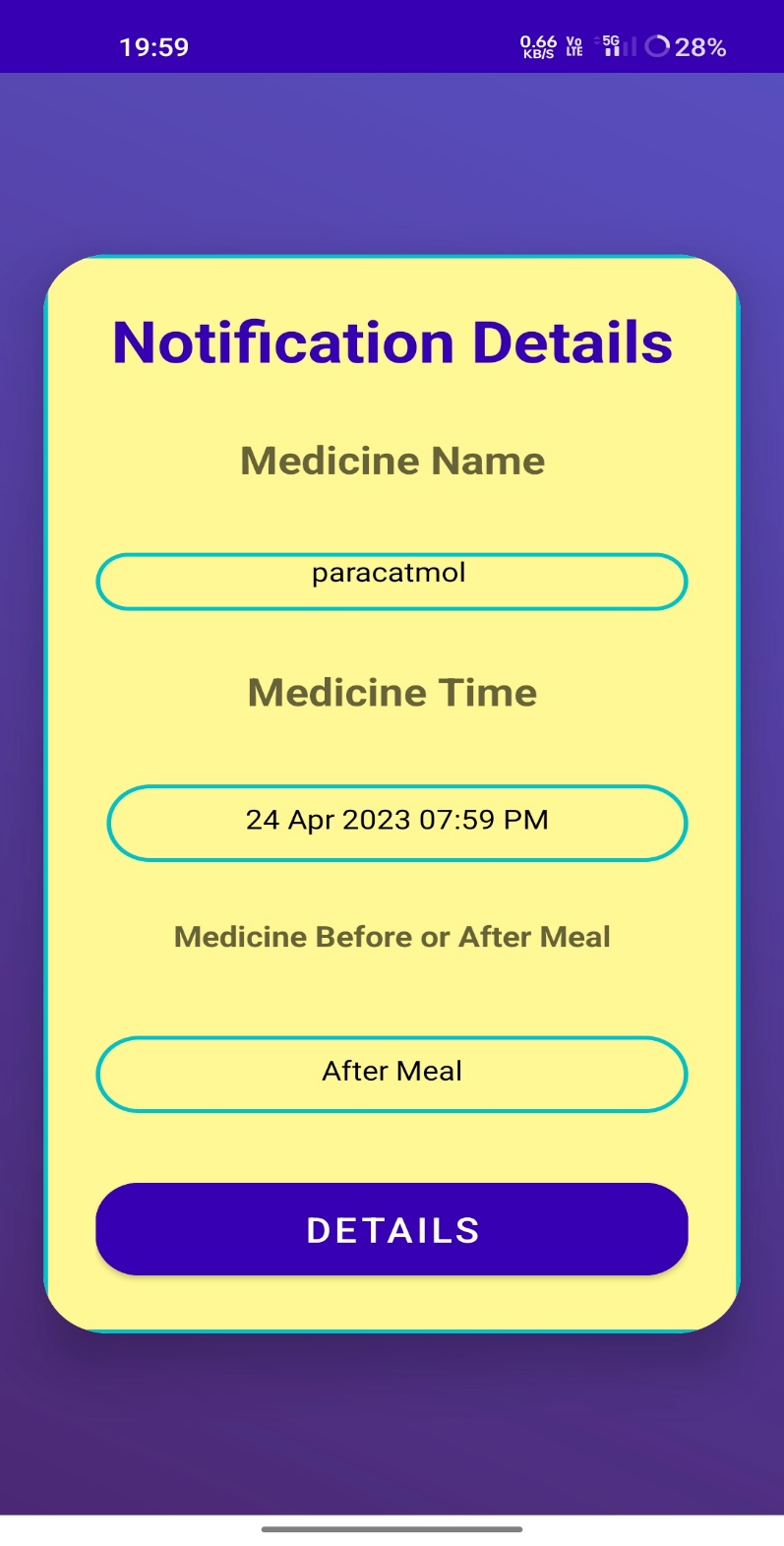
***Application login page***



***Application dashboard page***



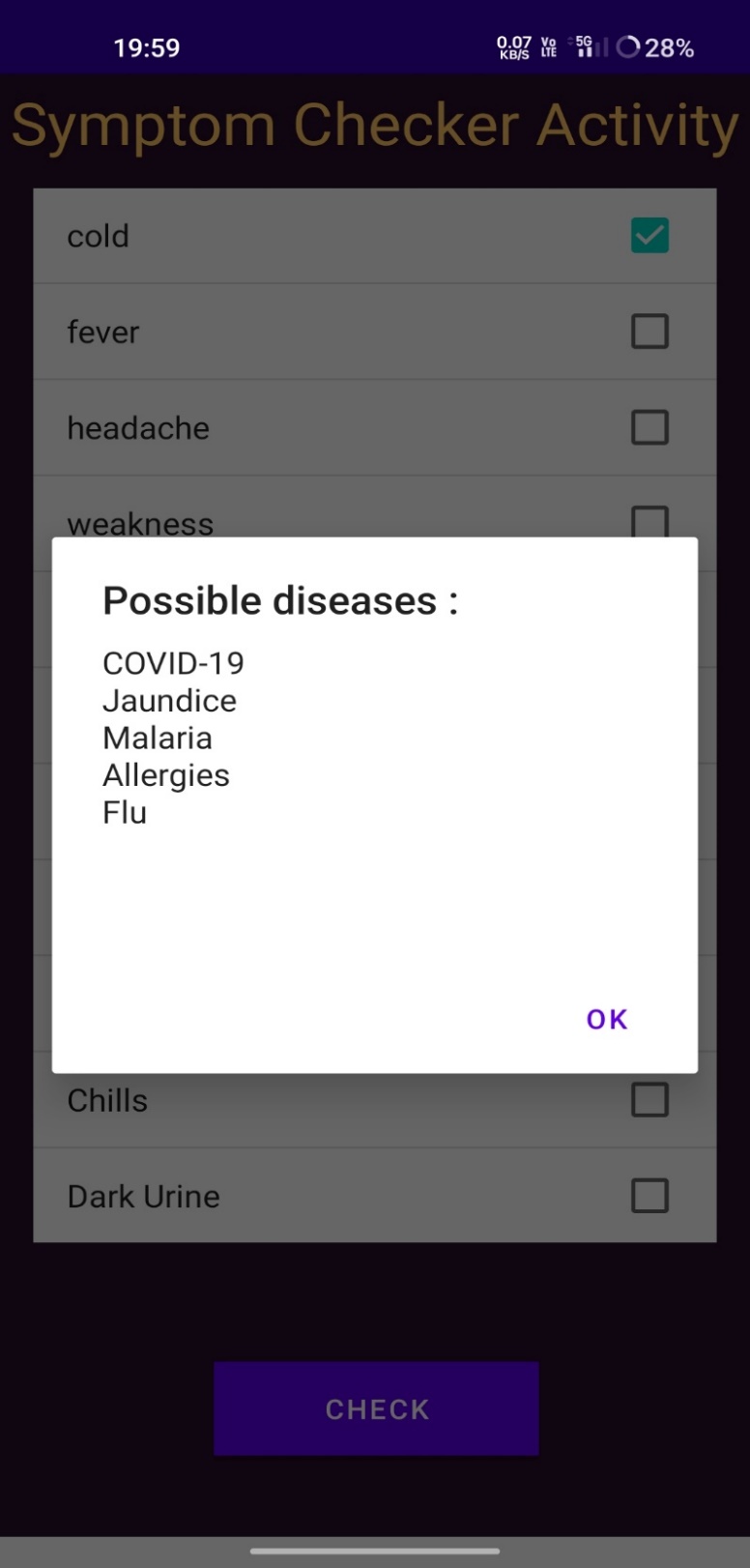
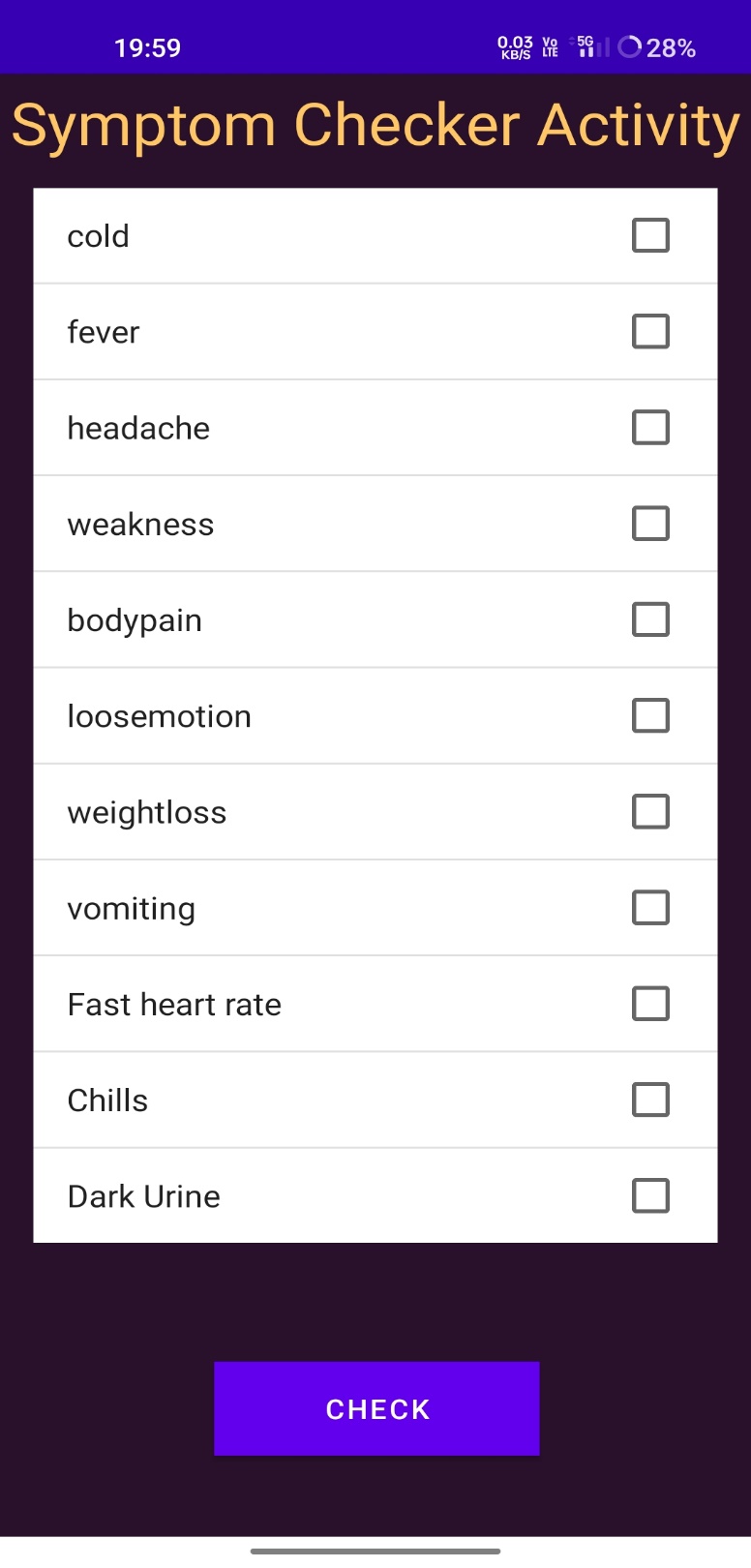
***Alarm setting page***



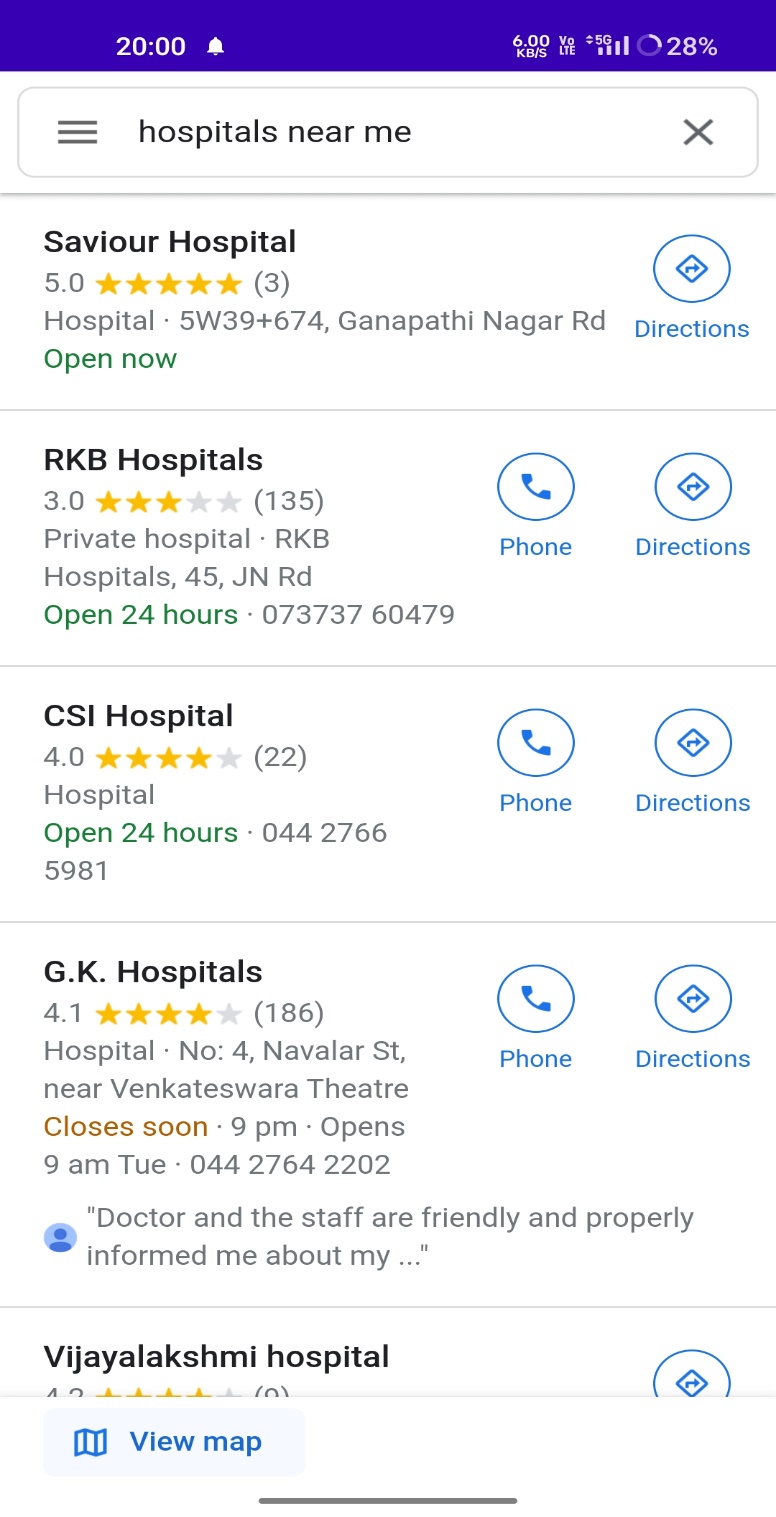
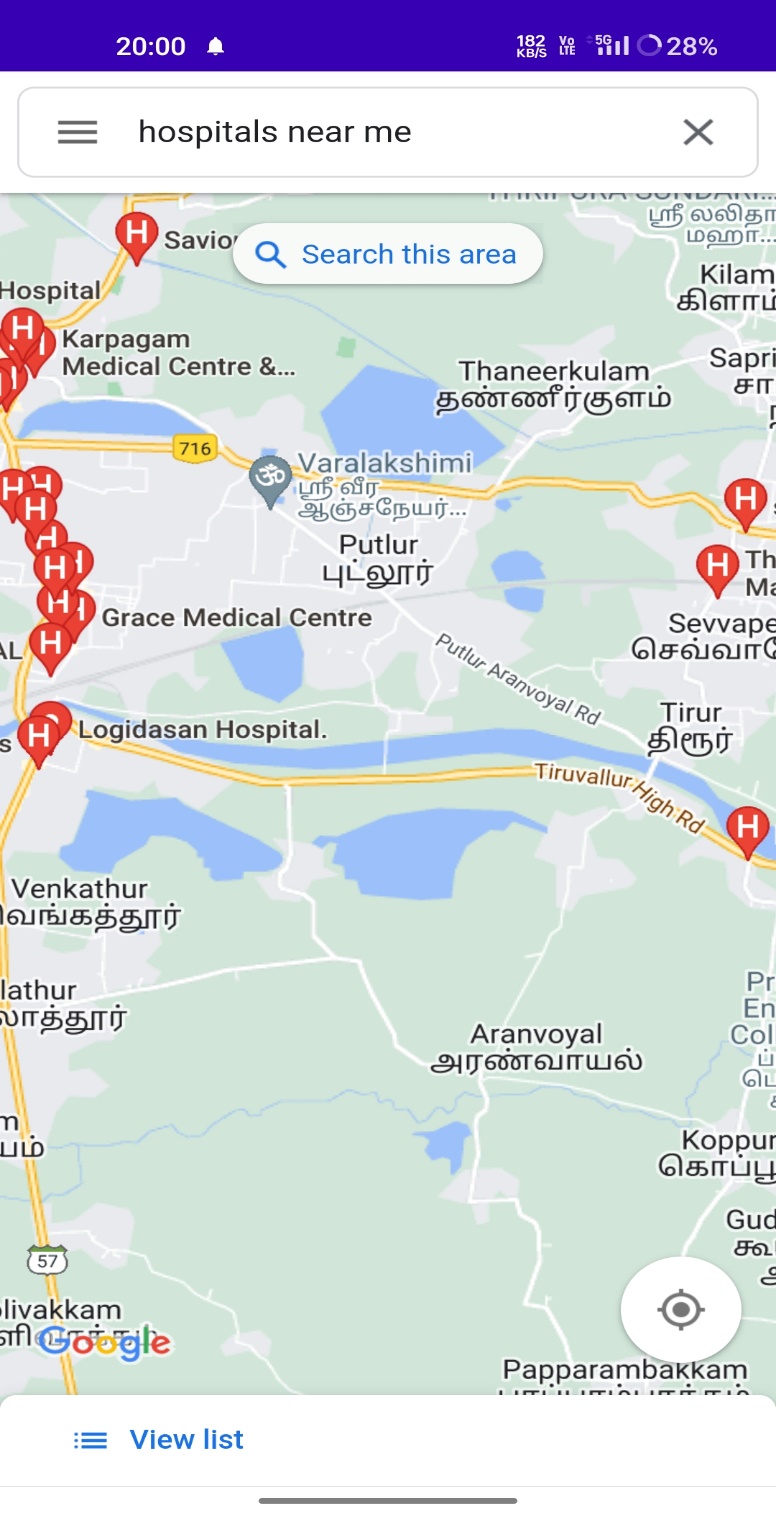
***Notification details page***



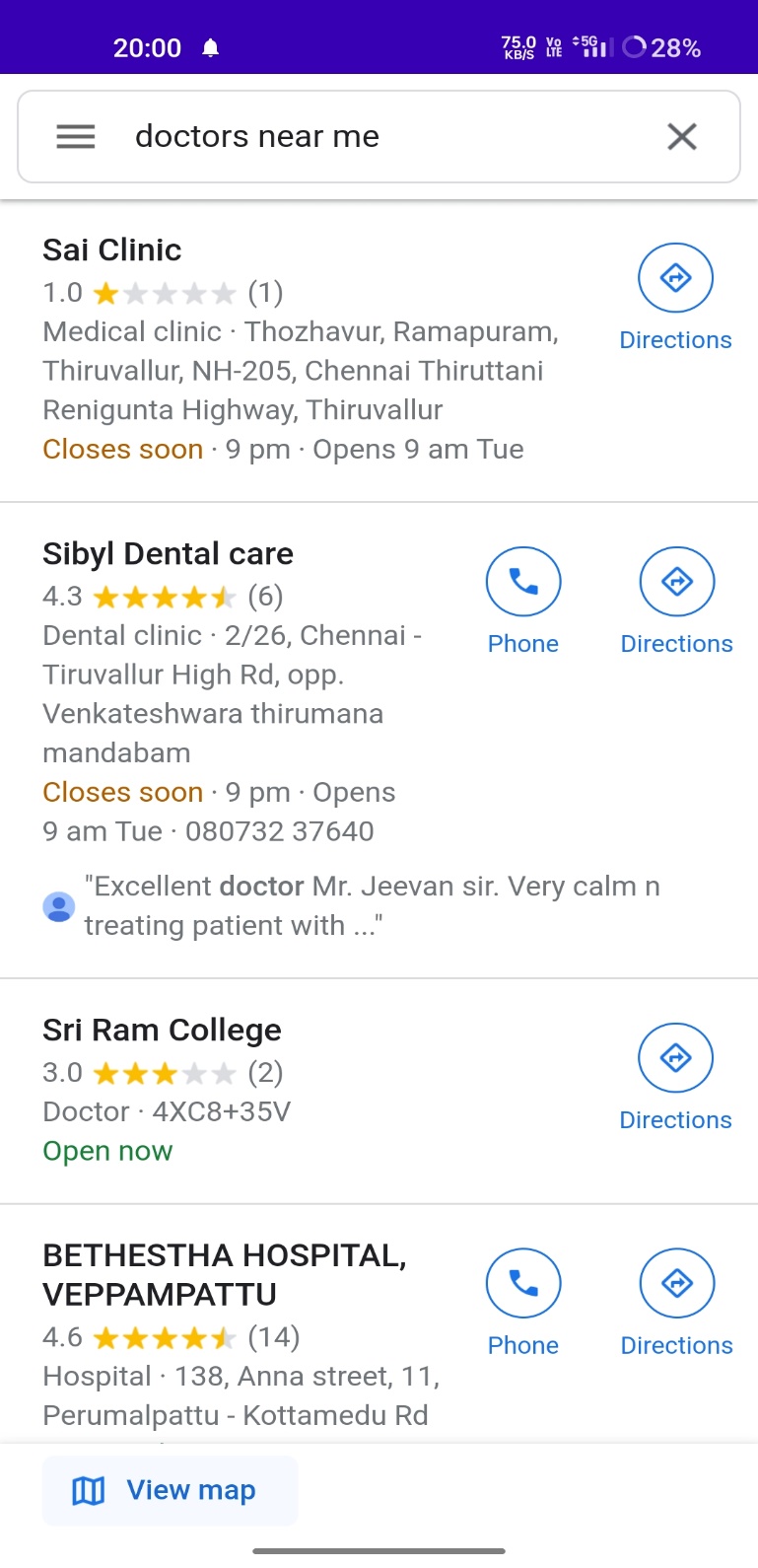
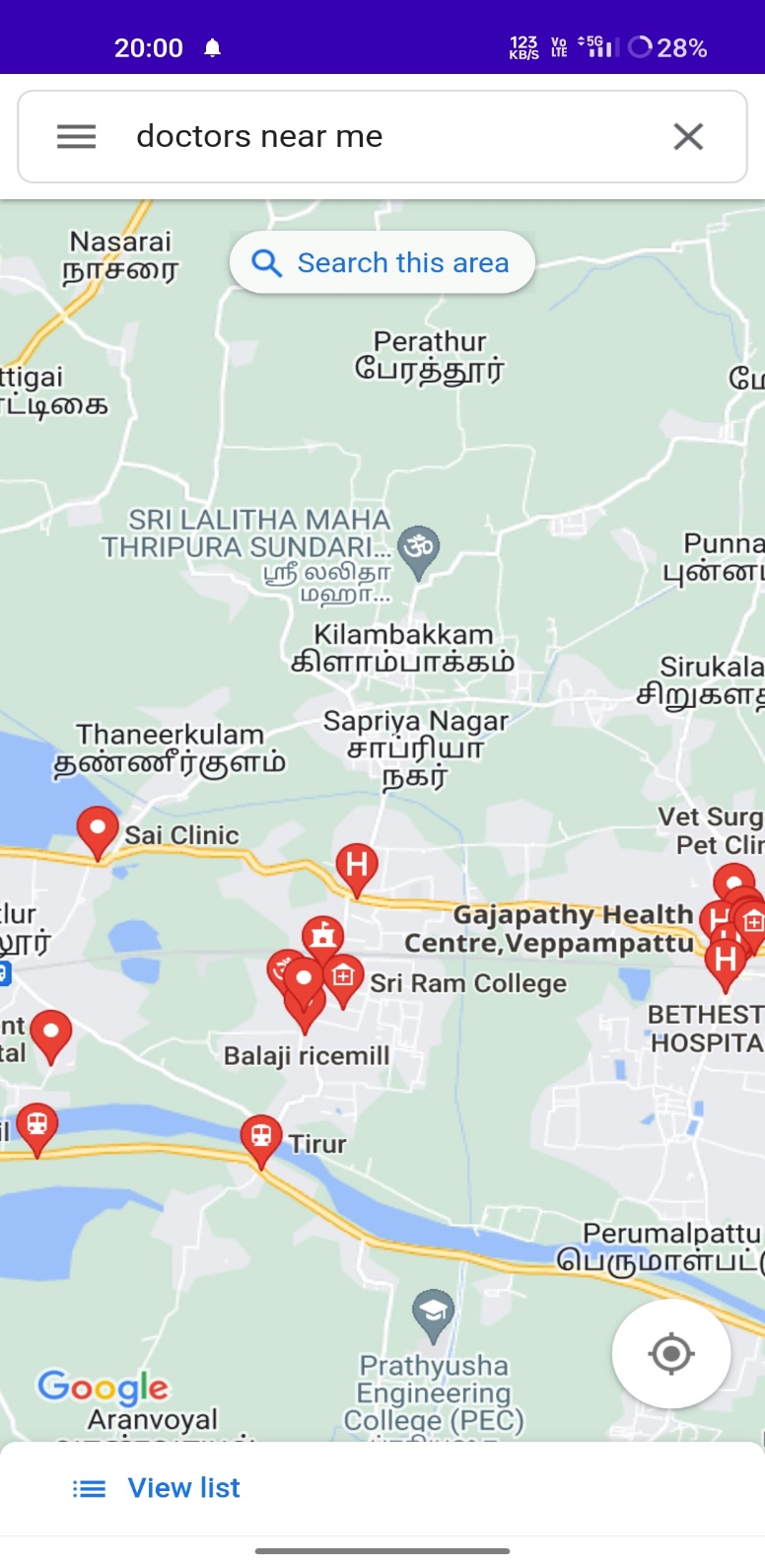
***Alarm details page***



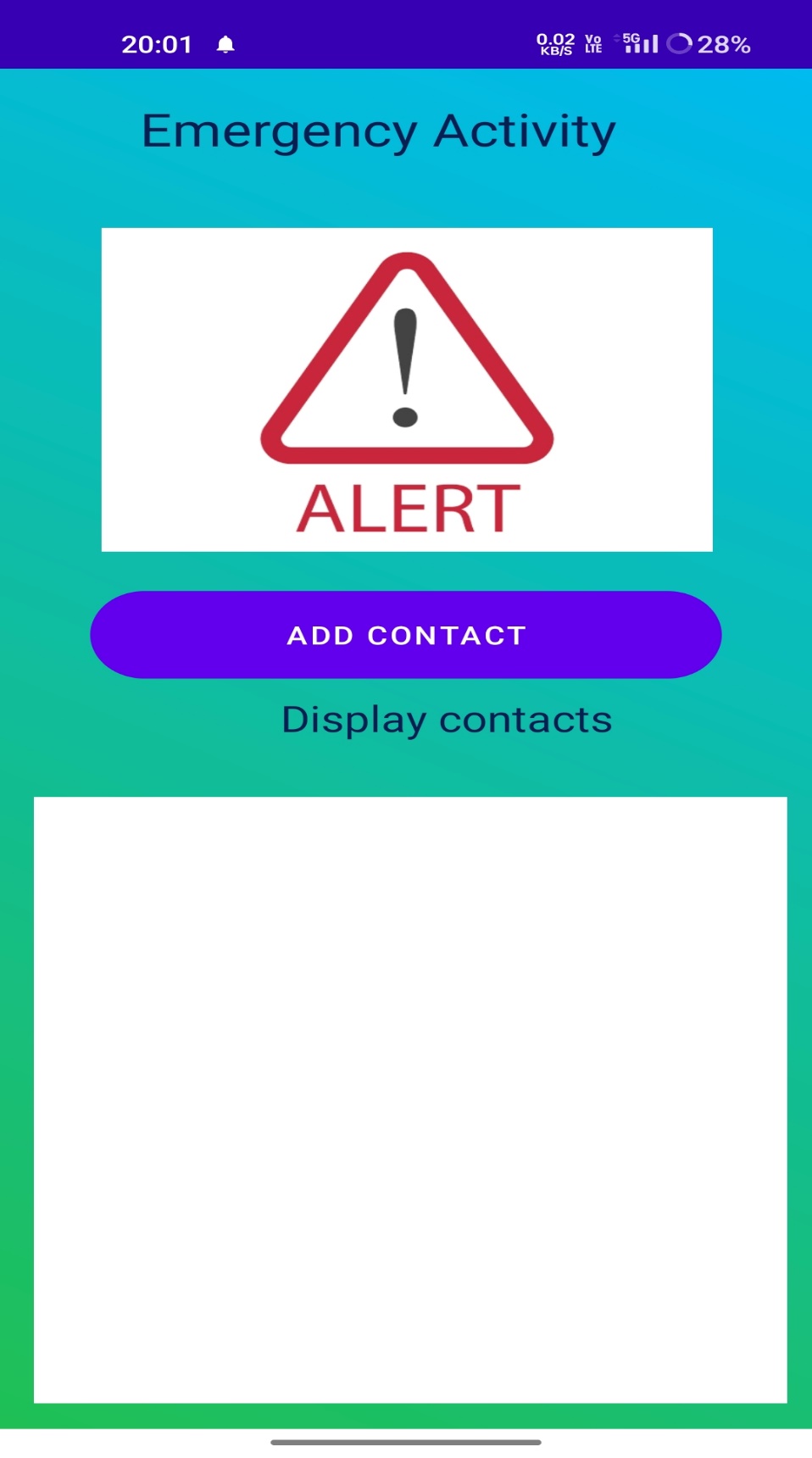
***Symptom checker page***



***Nearby Hospitals***



***Nearby Doctors page***



***Emergency calls page***