VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

"MEDAID"

Submitted by

Sunidhi S Patwardhan	4SF19IS109
Swathi S Nayak	4SF19IS115

In partial fulfillment of the requirements for the VI semester

MOBILE APPLICATION DEVELOPMENT

of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Under the Guidance of

Mr. B. Kranthi Kumar

Assistant Professor, Department of ISE

at

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

2021 - 22

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

CERTIFICATE

This is to certify that the **Mini Project** entitled **"MedAid"** has been carried out by **Sunidhi S Patwardhan (4SF19IS109)** and **Swathi S Nayak (4SF19IS115)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the VI semester **Mobile Application Development (18CSMP68)** of **Bachelor of Engineering in Information Science & Engineering** of Visvesvaraya Technological University, Belagavi during the year 2021 - 22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

Mr. B. Kranthi Kumar Assistant Professor Dept. of ISE, SCEM	**Dr. Shamanth Rai** Professor & HOD Dept. of ISE, SCEM

External Practical Examination:

Examiner's Name Signature with Date

1. ....

2. ....

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"MedAid"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mr. B. Kranthi Kumar** as the part of the VI semester **Mobile Application Development (18CSMP68)** of **Bachelor of Engineering** in **Information Science & Engineering**. This report has not been submitted to this or any other University.

Sunidhi S Patwardhan (4SF19IS109)
Swathi S Nayak (4SF19IS115)
SCEM, Mangaluru

Abstract

This project aims at maintaining the health status of the elderly people. The process of keeping track of medicine intake can sometimes be tedious for elderly people. There may be a lack of awareness or knowledge associated with their health issues. To overcome this,keeping track of medications is made simpler by this surveillance application for elderly people. It aims to remind patients to take their pills in a timely manner so they don't miss them. This provides efficiency and flexibility in tracking the medications. A major target customer would be an NGO that serves the elderly.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **“MedAid”**. We have completed it as a part of the VI semester **Mobile Application Development (18CSMP68)** of **Bachelor of Engineering** in **Information Science & Engineering** of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Mr. B. Kranthi Kumar**, Assistant Professor, Department of Information Science & Engineering for innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We express our sincere gratitude to **Dr. Shamanth Rai**, Professor &Head, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering & Management who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Sunidhi S Patwardhan
4SF19IS109
VI Sem, B.E., ISE
SCEM, Mangaluru

Swathi S Nayak
4SF19IS115
VI Sem, B.E., ISE
SCEM, Mangaluru

Table of Contents

List of Figures

Chapter 1

Introduction

There is a significant problem with patients failing to take their prescribed medicines at the appropriate times. Medical adherence refers to the degree or extent to which a patient takes the right medication at the right time. A number of studies reported that non-adherence may adversely affect the health of the patient, increasing medical costs, poor treatment outcomes and consumption of healthcare resources.

With our developing and technology dependent lives, we rely on gadgets, such as smart phones. Today, everyone owns one. We can use technology by using it effectively. In order to remind patients of their dosage timings, we are developing an android application that uses notification system to remind them.

1.1 Purpose

This project aims to develop an application which is user-friendly and reminds patients to take their medicines. The reminder of the intake of the particular medication depends on the their respective dosage time. Once the reminder is set, it becomes easy and efficient enough to keep track of the medicines. This application majorly focuses on helping the elderly people who tend to forget to take the pills on time.

1.2 Scope

Since medication non-adherence is a serious issue pertaining among patients, this application helps to fill this delay in intake of medicines thereby reducing the resulting medical costs and efficient use of healthcare resources.

1.3 Overview

This application is designed and developed which essentially focuses on the medicine intake of the elderly people who tend to forget to take their pills on time. Most of there is a risk that they can forget or there are elderly people who go out for work and forget to take the pills. The application is a way to reduce this risk as it is always important to keep track of the health status. It is handled by a single user called the admin who enters the patient details and also sets the reminder according to the time mentioned in the details filled. It has the feature of storing the patient details with the medicine they are supposed to take. There are two databases in which one keeps track of who has registered and the other one has the required patient details. Overall, this project gives an overview of how the medicine intake can be tracked with ease and felxibility.

Chapter 2

Requirements Specification

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz
- RAM : 8GB
- Hard Disk : 500GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Programming Language :Java and XML
- IDE :Android Studio
- Database: SQLite

Chapter 3

System Design

3.1 Architecture Diagram

The architecture diagram of the application is as shown in the below figure:

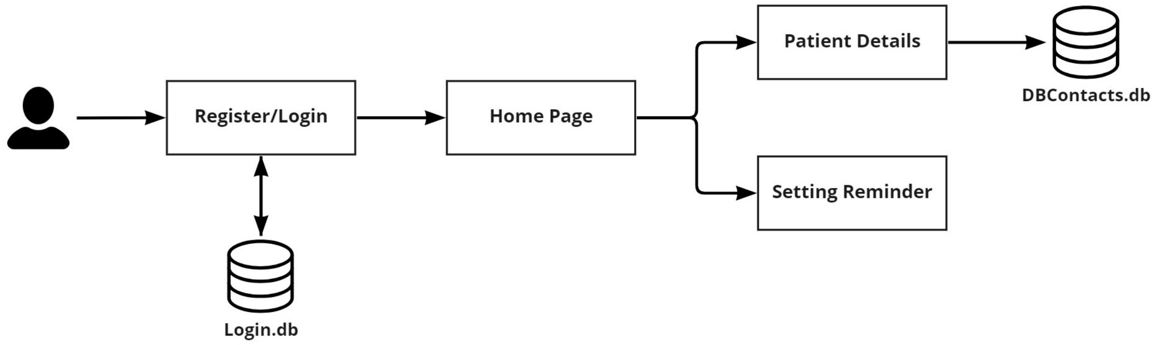


Figure 3.1: System Architecture Diagram

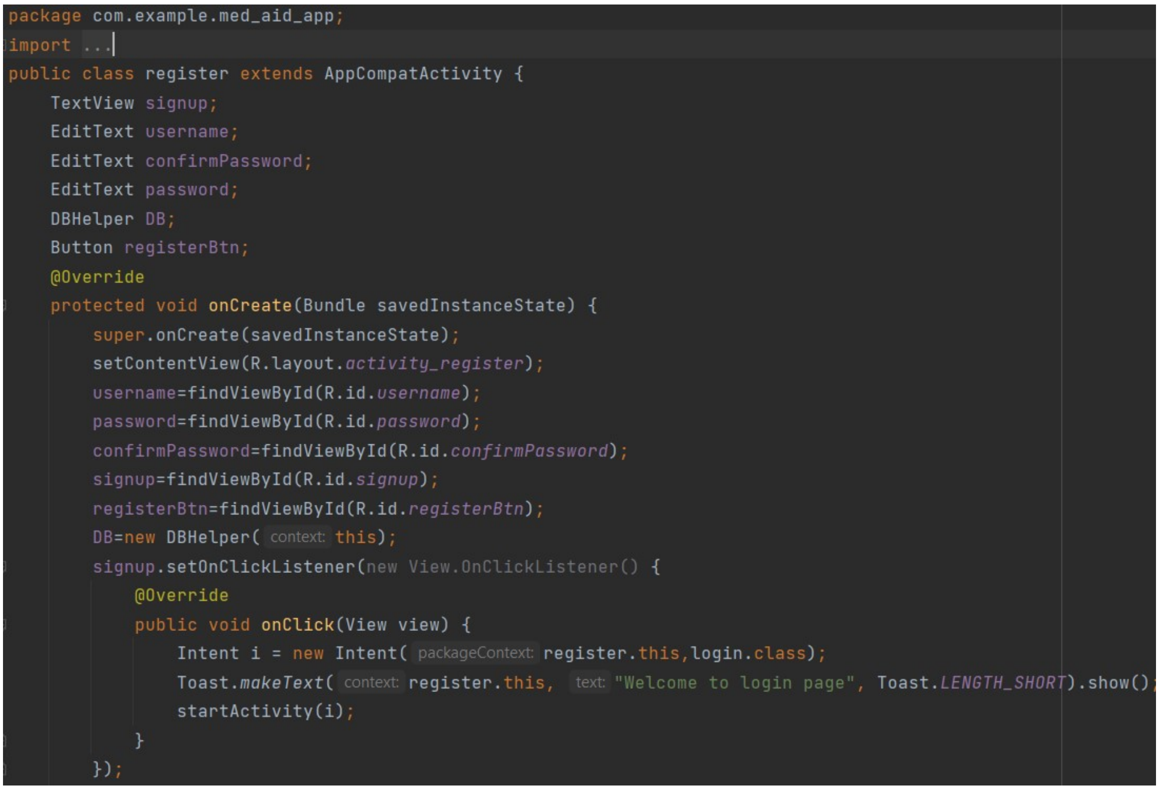
The new user registers in order to use the application. If the user is already existing, there is a provision of logging in. The user logs into the application and stores patient details. These patient details are stored in the database called Login.db which helps in the data retrieval. The user can then set reminder according the pill intakes of patients. Once the reminder is set, a notification pops up regarding the intake of medications.

Chapter 4

Implementation

Code Registration

In figure 4.1, this piece of code helps the user to enter the details in order to access and use the application. It consists of username, password and confirm password with a register button.

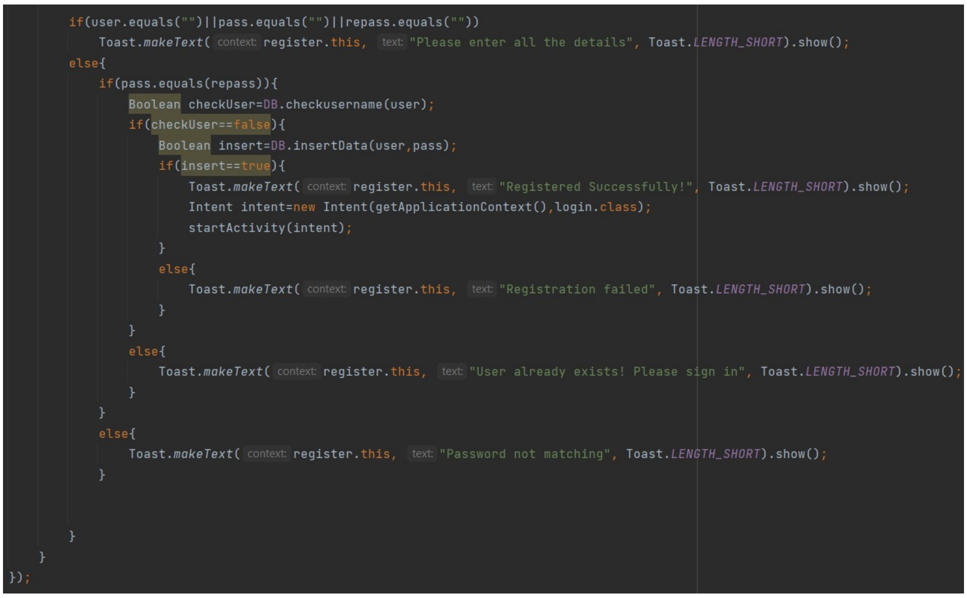


```
package com.example.med_aid_app;
import ...
public class register extends AppCompatActivity {
    TextView signup;
    EditText username;
    EditText confirmPassword;
    EditText password;
    DBHelper DB;
    Button registerBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        username=findViewById(R.id.username);
        password=findViewById(R.id.password);
        confirmPassword=findViewById(R.id.confirmPassword);
        signup=findViewById(R.id.signup);
        registerBtn=findViewById(R.id.registerBtn);
        DB=new DBHelper( context: this);
        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: register.this,login.class);
                Toast.makeText( context: register.this,  text: "Welcome to login page", Toast.LENGTH_SHORT).show();
                startActivity(i);
            }
        });
```

Figure 4.1: Snippet of user registration

Code for validity of registration

In figure 4.2, the piece of code checks two important things. Firstly, it checks if the username already exists. Secondly, it registers the particular user's details into the database.

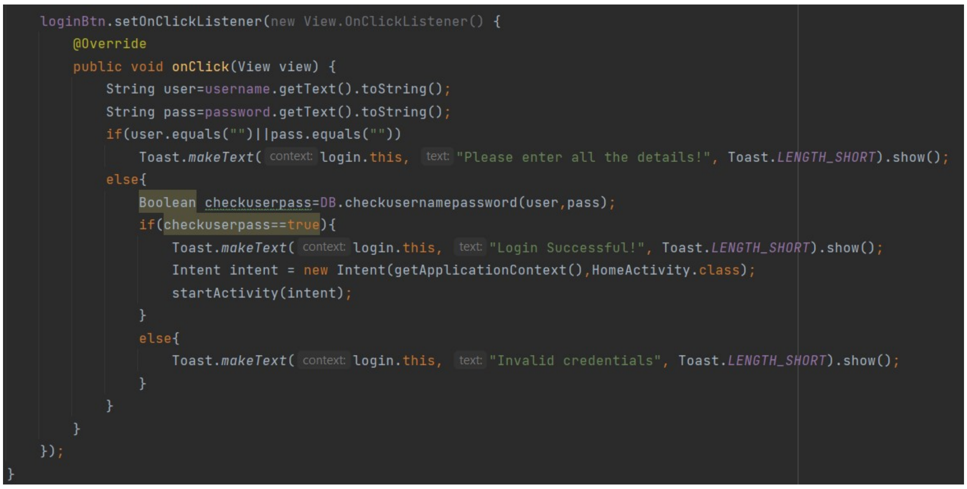


```
        if(user.equals("")||pass.equals("")||repass.equals(""))
            Toast.makeText( context: register.this,  text: "Please enter all the details", Toast.LENGTH_SHORT).show();
        else{
            if(pass.equals(repass)){
                Boolean checkUser=DB.checkusername(user);
                if(checkUser==false){
                    Boolean insert=DB.insertData(user,pass);
                    if(insert==true){
                        Toast.makeText( context: register.this,  text: "Registered Successfully!", Toast.LENGTH_SHORT).show();
                        Intent intent=new Intent(getApplicationContext(),login.class);
                        startActivity(intent);
                    }
                    else{
                        Toast.makeText( context: register.this,  text: "Registration failed", Toast.LENGTH_SHORT).show();
                    }
                }
                else{
                    Toast.makeText( context: register.this,  text: "User already exists! Please sign in", Toast.LENGTH_SHORT).show();
                }
            }
            else{
                Toast.makeText( context: register.this,  text: "Password not matching", Toast.LENGTH_SHORT).show();
            }


        }
    }
});
```

Figure 4.2: Snippet of validating the registration

Code for login

In figure 4.3, the code asks for the user to enter the username and password in order to login to the application.

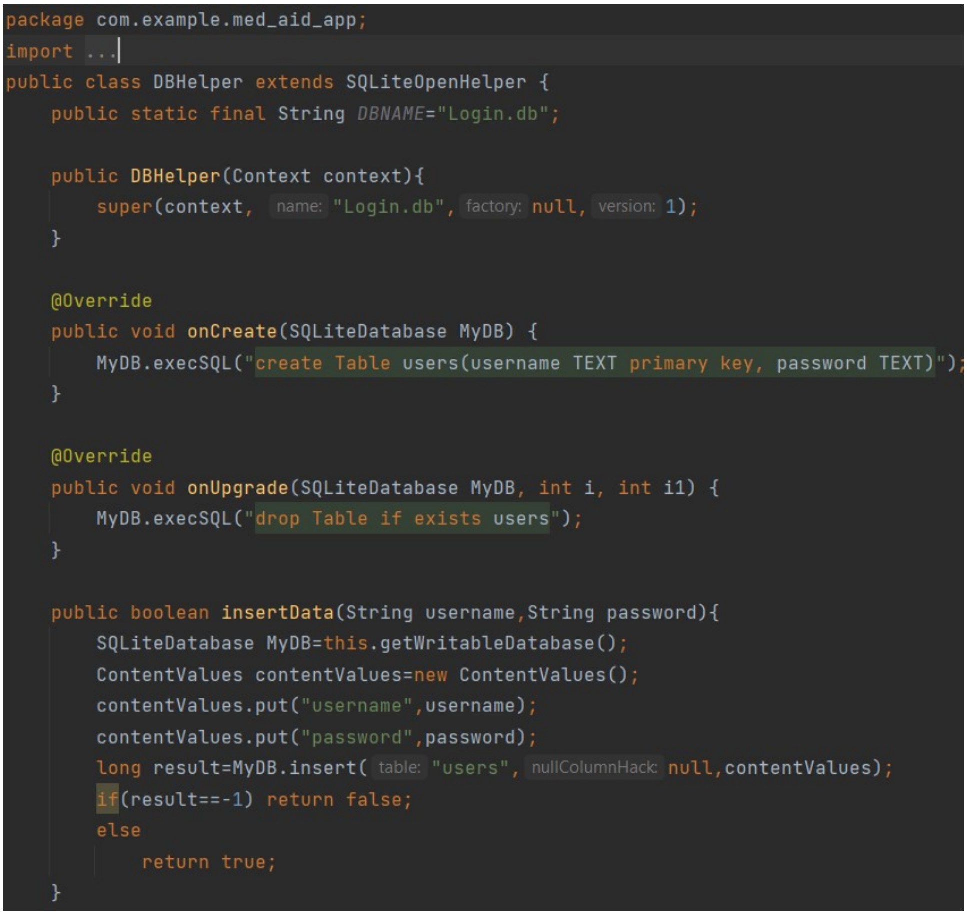


```
    loginBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String user=username.getText().toString();
            String pass=password.getText().toString();
            if(user.equals("")||pass.equals(""))
                Toast.makeText( context: login.this,  text: "Please enter all the details!", Toast.LENGTH_SHORT).show();
            else{
                Boolean checkuserpass=DB.checkusernamepassword(user,pass);
                if(checkuserpass==true){
                    Toast.makeText( context: login.this,  text: "Login Successful!", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(getApplicationContext(),HomeActivity.class);
                    startActivity(intent);
                }
                else{
                    Toast.makeText( context: login.this,  text: "Invalid credentials", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
}
```

Figure 4.3: Snippet of login

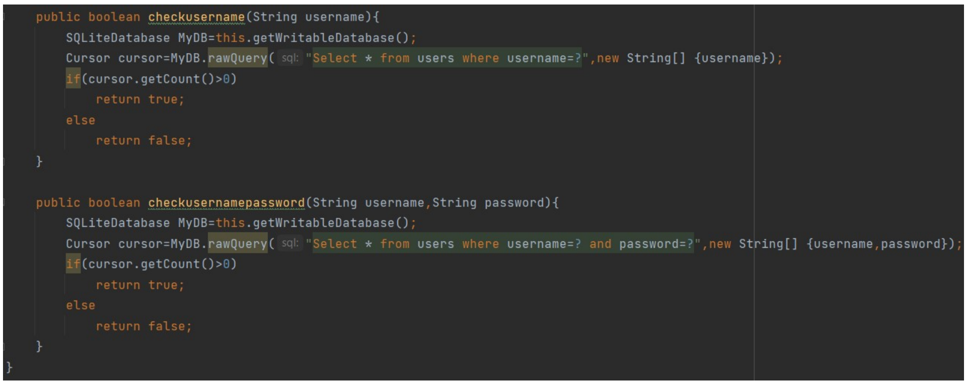
Code for DBHelper Database

In figure 4.4, the code helps to store the registration details of the user at the back-end named as Login.db.



```
package com.example.med_aid_app;
import ...
public class DBHelper extends SQLiteOpenHelper {
    public static final String DBNAME="Login.db";

    public DBHelper(Context context){
        super(context,  name: "Login.db",  factory: null,  version: 1);
    }

    @Override
    public void onCreate(SQLiteDatabase MyDB) {
        MyDB.execSQL("create Table users(username TEXT primary key, password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {
        MyDB.execSQL("drop Table if exists users");
    }

    public boolean insertData(String username,String password){
        SQLiteDatabase MyDB=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("username",username);
        contentValues.put("password",password);
        long result=MyDB.insert( table: "users",  nullColumnHack: null,contentValues);
        if(result==-1) return false;
        else
            return true;
    }
```

Figure 4.4: Snippet of back-end database for login - 1



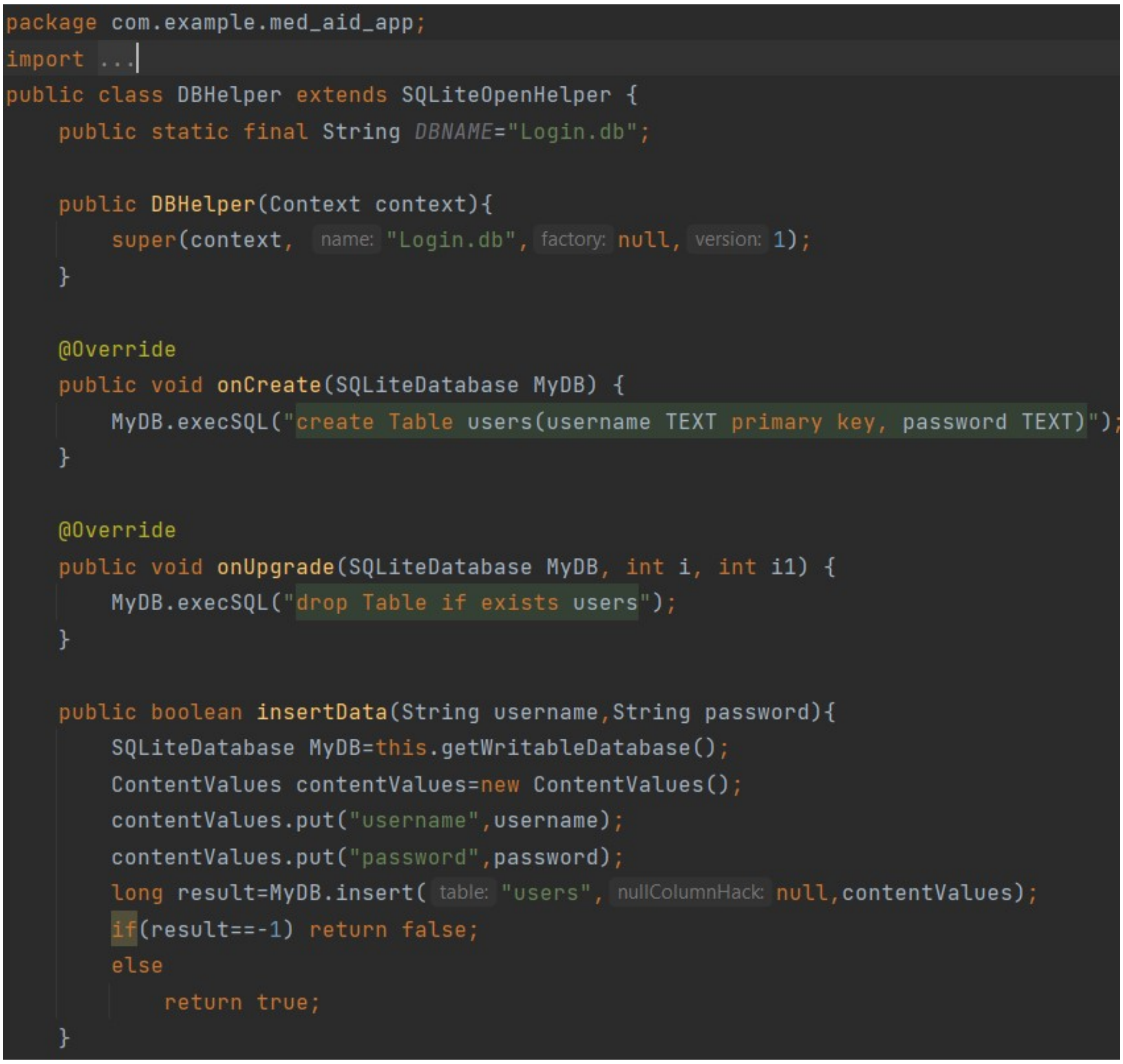
```
    public boolean checkusername(String username){
        SQLiteDatabase MyDB=this.getWritableDatabase();
        Cursor cursor=MyDB.rawQuery( sql: "Select * from users where username=?",new String[] {username});
        if(cursor.getCount()>0)
            return true;
        else
            return false;
    }

    public boolean checkusernamepassword(String username,String password){
        SQLiteDatabase MyDB=this.getWritableDatabase();
        Cursor cursor=MyDB.rawQuery( sql: "Select * from users where username=? and password=?",new String[] {username,password});
        if(cursor.getCount()>0)
            return true;
        else
            return false;
    }
}
```

Figure 4.5: Snippet of back-end database for login - 2

The username and password's validity are checked with the help of this database as shown in the below figure 4.5.

Patient Details Code

In the above figure 4.6, the code helps to store the patient details at the back-end with the help of database named DBContacts.db.

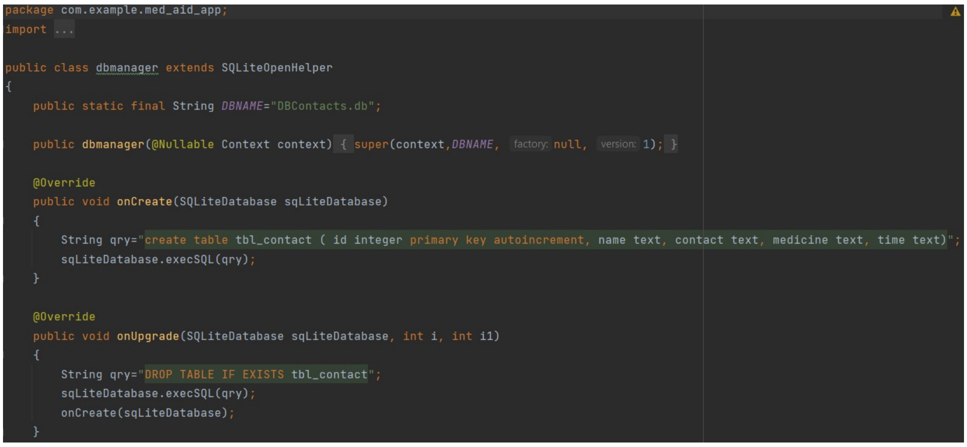


```
package com.example.med_aid_app;
import ...
public class DBHelper extends SQLiteOpenHelper {
    public static final String DBNAME="Login.db";

    public DBHelper(Context context){
        super(context, name: "Login.db", factory: null, version: 1);
    }

    @Override
    public void onCreate(SQLiteDatabase MyDB) {
        MyDB.execSQL("create Table users(username TEXT primary key, password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {
        MyDB.execSQL("drop Table if exists users");
    }

    public boolean insertData(String username,String password){
        SQLiteDatabase MyDB=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("username",username);
        contentValues.put("password",password);
        long result=MyDB.insert( table: "users", nullColumnHack: null,contentValues);
        if(result==-1) return false;
        else
            return true;
    }
```

Figure 4.6: Snippet of patient details

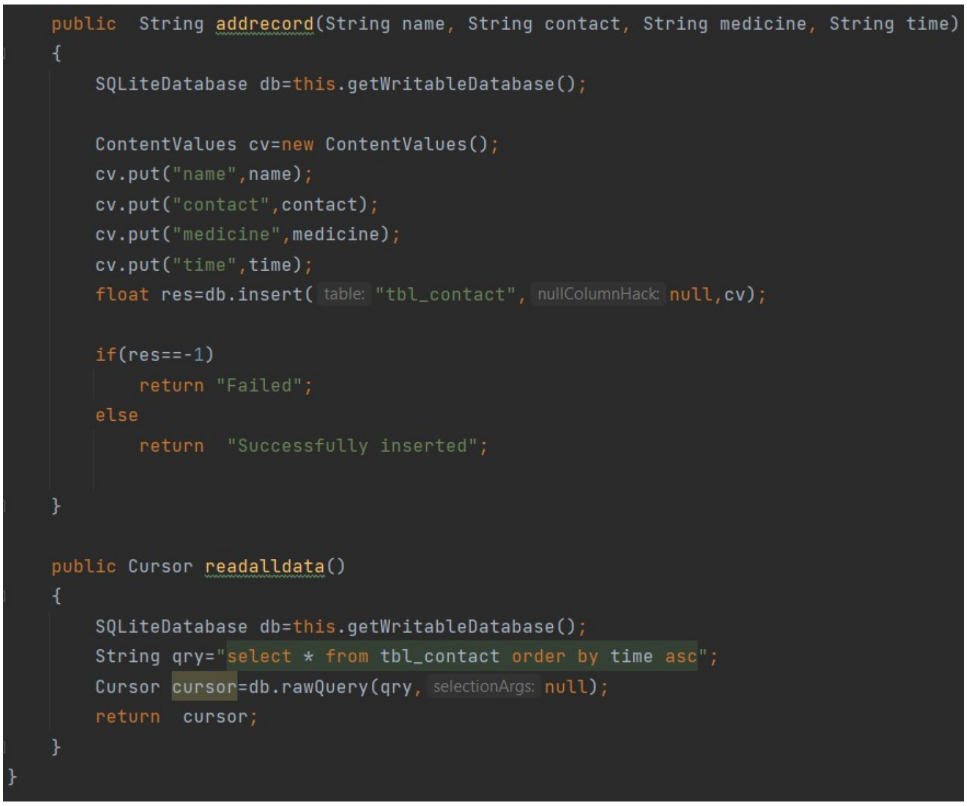
Code for dbmanager

In figure 4.7, the code creates table in order to store the details of the patients and drops the table if it is already exists in the database.



```
package com.example.med_aid_app;
import ...

public class dbmanager extends SQLiteOpenHelper
{
    public static final String DBNAME="DBContacts.db";

    public dbmanager(@Nullable Context context) { super(context,DBNAME, factory: null, version: 1); }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase)
    {
        String qry="create table tbl_contact ( id integer primary key autoincrement, name text, contact text, medicine text, time text)";
        sqLiteDatabase.execSQL(qry);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1)
    {
        String qry="DROP TABLE IF EXISTS tbl_contact";
        sqLiteDatabase.execSQL(qry);
        onCreate(sqLiteDatabase);
    }
```

Figure 4.7: Snippet of back-end database DBContacts.db - 1



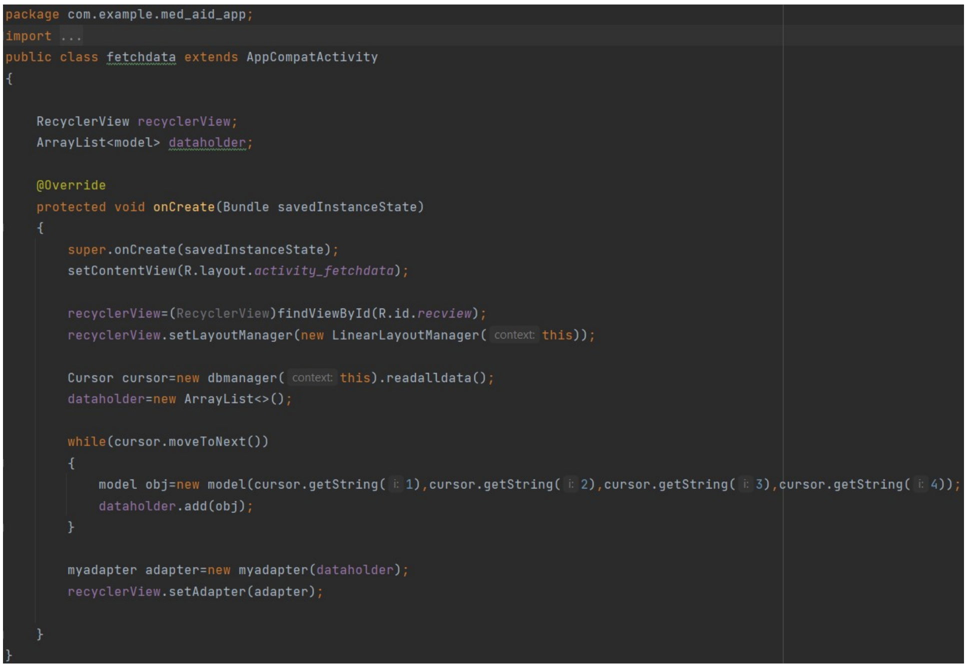
```
    public  String addrecord(String name, String contact, String medicine, String time)
    {
        SQLiteDatabase db=this.getWritableDatabase();

        ContentValues cv=new ContentValues();
        cv.put("name",name);
        cv.put("contact",contact);
        cv.put("medicine",medicine);
        cv.put("time",time);
        float res=db.insert( table: "tbl_contact", nullColumnHack: null,cv);

        if(res==-1)
            return "Failed";
        else
            return  "Successfully inserted";

    }

    public Cursor readalldata()
    {
        SQLiteDatabase db=this.getWritableDatabase();
        String qry="select * from tbl_contact order by time asc";
        Cursor cursor=db.rawQuery(qry, selectionArgs: null);
        return  cursor;
    }
}
```

Figure 4.8: Snippet of back-end database DBContacts.db - 2

In the above figure 4.8, the code adds a particular patient record into the database. With this the readalldata() helps in displaying the data according to the specified condition.

Code for fetching the data from the database

In figure 4.9, the code asks for the user to enter the username and password in order to login to the application.

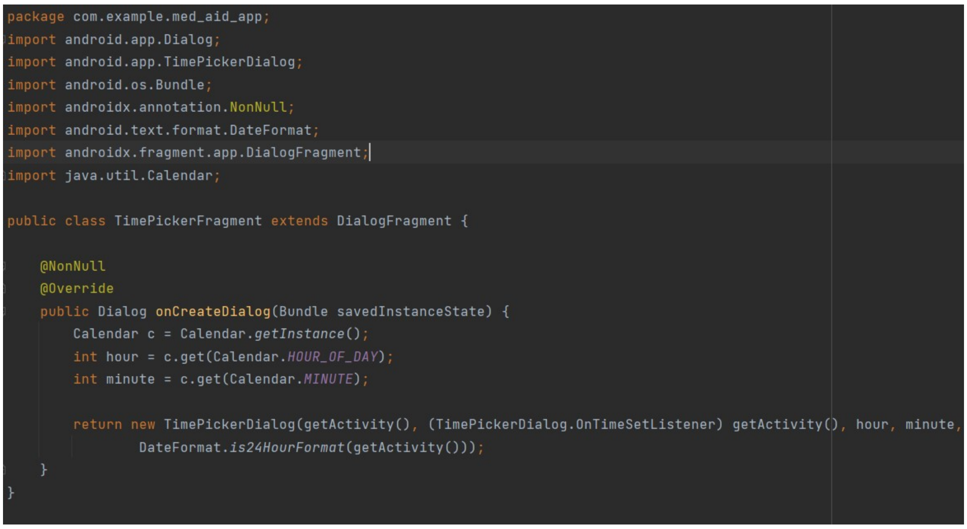


```
package com.example.med_aid_app;
import ...
public class fetchdata extends AppCompatActivity
{

    RecyclerView recyclerView;
    ArrayList<model> dataholder;

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

        recyclerView=(RecyclerView)findViewById(R.id.recview);
        recyclerView.setLayoutManager(new LinearLayoutManager( context: this));

        Cursor cursor=new dbmanager( context: this).readalldata();
        dataholder=new ArrayList<>();

        while(cursor.moveToNext())
        {
            model obj=new model(cursor.getString( i: 1),cursor.getString( i: 2),cursor.getString( i: 3),cursor.getString( i: 4));
            dataholder.add(obj);
        }

        myadapter adapter=new myadapter(dataholder);
        recyclerView.setAdapter(adapter);

    }
}
```

Figure 4.9: Snippet of fetching the data from the database

Code for time picker

In figure 4.10, the code allows to select the time of day in either 24 hour or AM/PM mode. The time consists of hours, minutes and clock format.

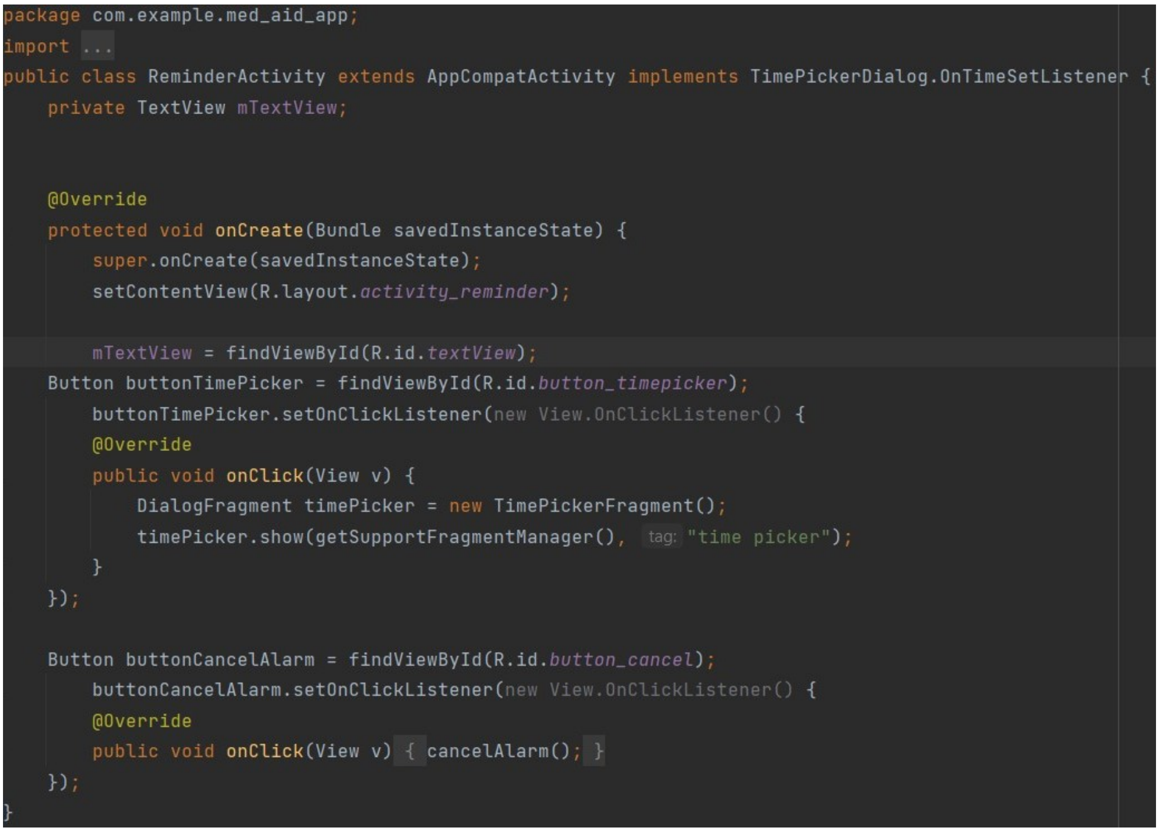


```
package com.example.med_aid_app;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import androidx.annotation.NonNull;
import android.text.format.DateFormat;
import androidx.fragment.app.DialogFragment;
import java.util.Calendar;

public class TimePickerFragment extends DialogFragment {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);

        return new TimePickerDialog(getActivity(), (TimePickerDialog.OnTimeSetListener) getActivity(), hour, minute,
                DateFormat.is24HourFormat(getActivity()));
    }
}
```

Figure 4.10: Snippet of time picker

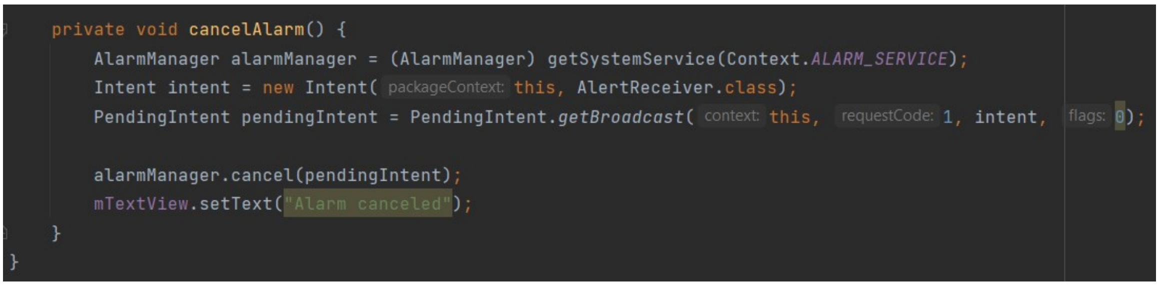
Code for setting the reminder

In figure 4.11, the code uses two buttons, one for opening a time picker for setting the alarm and the other for cancelling the alarm.



```
package com.example.med_aid_app;
import ...
public class ReminderActivity extends AppCompatActivity implements TimePickerDialog.OnTimeSetListener {
    private TextView mTextView;


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

        mTextView = findViewById(R.id.textView);
    Button buttonTimePicker = findViewById(R.id.button_timepicker);
        buttonTimePicker.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            DialogFragment timePicker = new TimePickerFragment();
            timePicker.show(getSupportFragmentManager(),  tag: "time picker");
        }
    });

    Button buttonCancelAlarm = findViewById(R.id.button_cancel);
        buttonCancelAlarm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { cancelAlarm(); }
    });
}
```

Figure 4.11: Snippet of setting the reminder for pill intake - 1



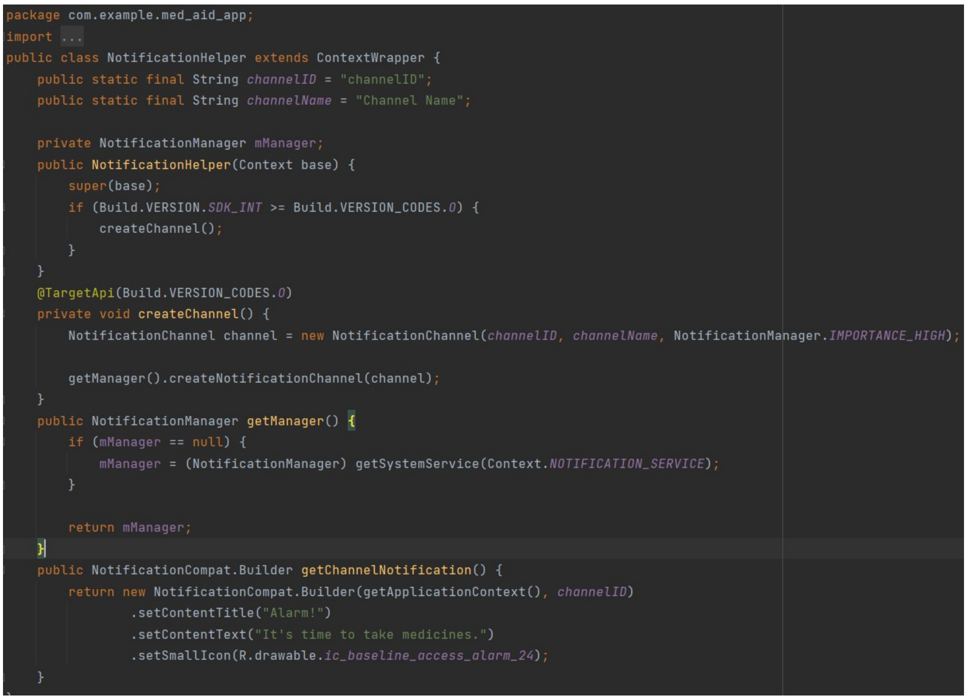
```
    private void cancelAlarm() {
        AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
        Intent intent = new Intent( packageContext: this, AlertReceiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast( context: this,  requestCode: 1, intent,  flags: 0);

        alarmManager.cancel(pendingIntent);
        mTextView.setText("Alarm canceled");
    }
}
```

Figure 4.12: Snippet of setting the reminder for pill intake - 2

The above figure 4.12 is used to cancel an alarm if not necessary. The alarm can again be set with the help of time picker.

Code for getting the notification

In figure 4.13, the code is used to send a notification about reminding the medicine intake.



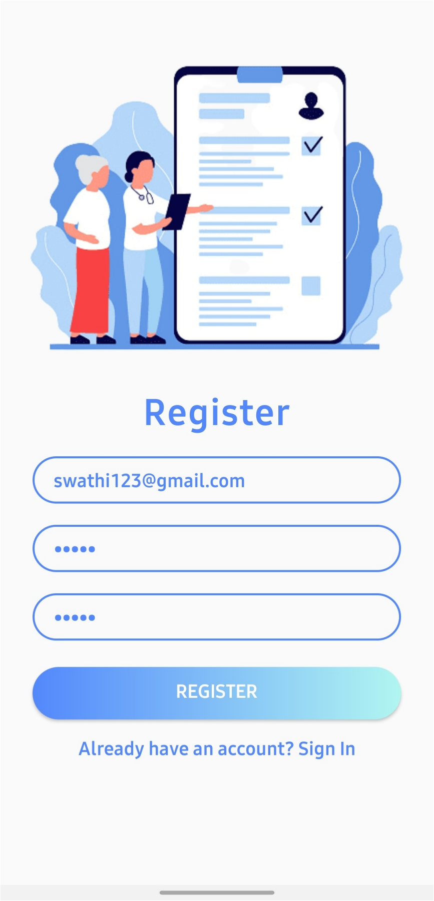
```java
package com.example.med_aid_app;
import ...
public class NotificationHelper extends ContextWrapper {
    public static final String channelID = "channelID";
    public static final String channelName = "Channel Name";

    private NotificationManager mManager;
    public NotificationHelper(Context base) {
        super(base);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            createChannel();
        }
    }
    @TargetApi(Build.VERSION_CODES.O)
    private void createChannel() {
        NotificationChannel channel = new NotificationChannel(channelID, channelName, NotificationManager.IMPORTANCE_HIGH);

        getManager().createNotificationChannel(channel);
    }
    public NotificationManager getManager() {
        if (mManager == null) {
            mManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        }

        return mManager;
    }
    public NotificationCompat.Builder getChannelNotification() {
        return new NotificationCompat.Builder(getApplicationContext(), channelID)
                .setContentTitle("Alarm!")
                .setContentText("It's time to take medicines.")
                .setSmallIcon(R.drawable.ic_baseline_access_alarm_24);
    }
```

Figure 4.13: Snippet of receiving the notification

Chapter 5

Results and Discussion

Below figure 5.1 is the registration page of the application.



Register

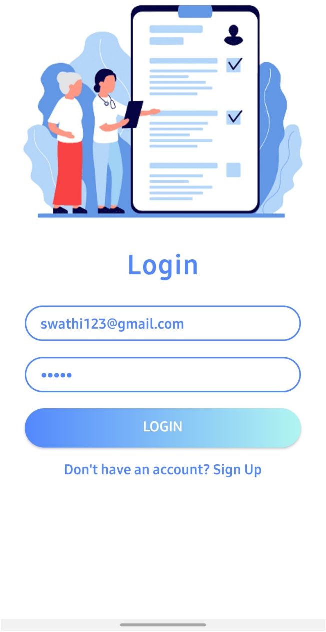
swathi123@gmail.com

•••••

•••••

Figure 5.1: Registration page

Below figure 5.2 is the login page of the application.



Login

swathi123@gmail.com

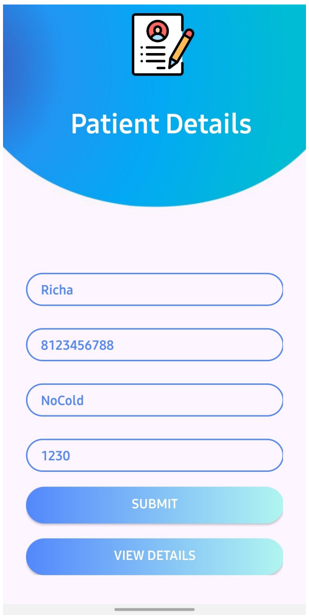
•••••

LOGIN

Don't have an account? Sign Up

Figure 5.2: Login page

Below figure 5.3 is used to enter the patient details.



Patient Details

Richa

8123456788

NoCold

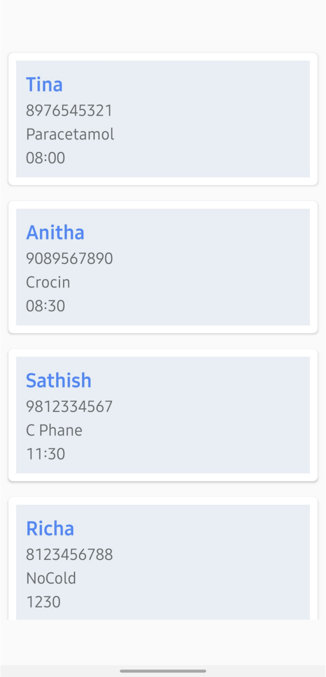
1230

SUBMIT

VIEW DETAILS

Figure 5.3: Form for patient details

Below figure 5.4 shows the patient details.



Tina
8976545321
Paracetamol
08:00

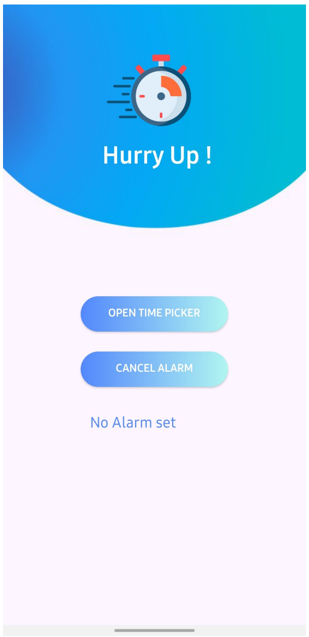
Anitha
9089567890
Crocin
08:30

Sathish
9812334567
C Phane
11:30

Richa
8123456788
NoCold
1230

Figure 5.4: Patient details

Below figure 5.5 shows option to set or cancel the alarm.



No Alarm set

Figure 5.5: Setting or cancelling the alarm

Below figure 5.6 shows the alarm setting.

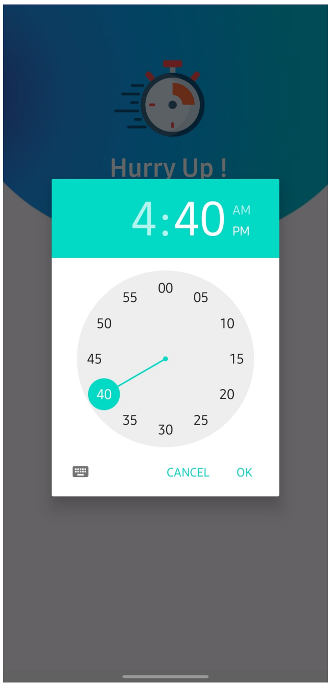
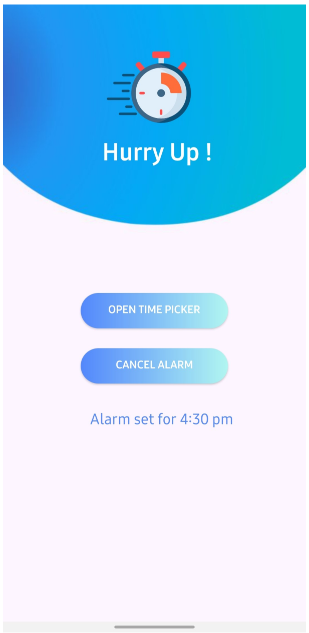


Figure 5.6: Alarm Setting

Below figure 5.7 shows the alarm set.



Alarm set for 4:30 pm

Figure 5.7: Alarm Set

Chapter 6

Conclusion and Future work

This project is developed in favour of tracking medicines for elderly people so that they can take care of their health without compromise. The system facilitates the timely intake of medicines by the patients by maintaining accurate patient information. Furthermore, the application provides the functionality of setting reminders for pill intake in addition to storing the medicines that must be taken by patients. In this one-user application, the administrator keeps track of the details of the patients and sets reminders accordingly. As a result, it facilitates the insertion of details, viewing of patient details, and setting of reminders. The patient information is stored in one database, and the administration information is stored in the other database. The implementation of storing medicine stocks and letting users know if they have consumed the pills is the next milestone to be reached.

References

[1] Google Developer Training, ”Android Developer Fundamentals Course-Concept Reference”, Google Developer Training Team, 2017. https://www.gitbook.com/book/googledeveloper-training/android-developer-fundamentals-course-concepts/details.

[2] Erik Hellman, “Android Programming-Pushing the limits”, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197.

[3] Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st SPD Publishers, 2015. ISBN-13: 978-9352131341.

[4] Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Berd Ranch Guide”, 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054.