# **COS30017 – Software Development for Mobile Devices**

## **Formative Assignment 5**

### **Task 1**

Screenshot of application

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Java Codes**

MainActivity

package com.example.task1;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cardview.widget.CardView;  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
import androidx.fragment.app.FragmentActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.app.AlarmManager;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.android.material.navigation.NavigationView;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
  
public class MainActivity extends AppCompatActivity implements DeleteDialog.OnFragmentInteractionListener{  
 //Database  
 public static final String *DATABASE\_NAME*="taskDB";  
 DatabaseManager databaseManager;  
  
 //RecyclerView  
 RecyclerView recyclerView;  
 RecyclerViewAdapter recyclerViewAdapter;  
 ArrayList<Task> taskArrayList;  
 ArrayList<Task> overdueArrayList;  
 Date currentDate = Calendar.*getInstance*().getTime();  
  
 //Nav Menu  
 BottomNavigationView bottomNavigationView;  
  
 //Alarm  
 static AlarmManager *alarmManager*;  
 static PendingIntent *alarmIntent*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 //Recycler  
 initRecycler();  
 //SQL  
 loadTable();  
 //Floating Button  
 initFloatingButton();  
 //Navigation  
 initNav();  
 //Alarm  
  
 }  
  
  
 private void initNav() {  
 bottomNavigationView = findViewById(R.id.*main\_bottom\_nav*);  
 bottomNavigationView.getMenu().getItem(1).setChecked(true);  
 bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 //make swtich case link to other activityes  
 switch (menuItem.getItemId()){  
 case R.id.*completed\_view*:  
 startActivity(new Intent(getApplicationContext(),CompletedActivity.class));  
 finish();  
 break;  
 case R.id.*overdue\_view*:  
 startActivity(new Intent(getApplicationContext(),OverdueActivity.class));  
 finish();  
 break;  
 }  
 return false;  
 }  
 });  
 }  
  
 private void initRecycler(){  
 recyclerView=findViewById(R.id.*recyclerView*);  
 taskArrayList=new ArrayList<>();  
 overdueArrayList=new ArrayList<>();  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
 }  
  
 private void loadTable(){  
 databaseManager = new DatabaseManager(this);  
 Cursor cursor = databaseManager.loadTasks();  
 if(cursor.moveToFirst()){  
 do{  
 sortDate(cursor);  
  
 }while(cursor.moveToNext());  
 for(int i=0;i<overdueArrayList.size();i++){  
 taskArrayList.add(overdueArrayList.get(i));  
 }  
 recyclerViewAdapter= new RecyclerViewAdapter(taskArrayList,this);  
 recyclerView.setAdapter(recyclerViewAdapter);  
 }  
 }  
 private void initFloatingButton(){  
 FloatingActionButton fb = findViewById(R.id.*fb*);  
 fb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this,AddTask.class));  
 finish();  
 }  
 });  
 }  
  
 //Support functions  
 private void insertMain(Cursor cursor){  
 taskArrayList.add(new Task(  
 cursor.getInt(0),  
 cursor.getString(1),  
 cursor.getString(2),  
 cursor.getString(3),  
 cursor.getString(4),  
 cursor.getString(5)  
  
 ));  
 }  
 private void insertOverdue(Cursor cursor){  
 overdueArrayList.add(new Task(  
 cursor.getInt(0),  
 cursor.getString(1),  
 cursor.getString(2),  
 cursor.getString(3),  
 cursor.getString(4),  
 cursor.getString(5)  
  
 ));  
 }  
 private void sortDate(Cursor cursor){  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 try {  
 Date selectedDate = sdf.parse(cursor.getString(2));  
  
 long curdate=currentDate.getTime();  
 long seldate=selectedDate.getTime();  
 float result = (float)(seldate-curdate)/(24\*60\*60\*1000)+1;  
  
  
 if(result<0){  
 if(cursor.getString(5).equals("0")){  
 insertOverdue(cursor);  
 }  
  
 }else{  
 if(cursor.getString(5).equals("0")){  
 insertMain(cursor);  
 }  
 }  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void onFragmentInteraction(int position) {  
 taskArrayList.remove(position);  
 recyclerViewAdapter.notifyItemRemoved(position);  
 recyclerViewAdapter.notifyItemRangeChanged(position,taskArrayList.size());  
 }  
  
 @Override  
 public void onBackPressed() {  
 finish();  
 }  
  
}

AddTask

package com.example.task1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.AlarmManager;  
import android.app.DatePickerDialog;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.provider.CalendarContract;  
import android.util.Log;  
import android.view.View;  
import android.widget.CheckBox;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
import java.sql.Time;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Locale;  
import java.util.Map;  
  
public class AddTask extends AppCompatActivity implements DatePickerDialog.OnDateSetListener{  
 //Edit Text Fields  
 private EditText editText,editDetails;  
 private String title,details;  
  
 //Date Selection  
 private TextView calendarText;  
 private ImageView calendar;  
 private Date selectedDate;  
  
 //Buttons  
 private CheckBox priorityButton;  
 private String priority="0";  
  
 //Submission  
 private ImageButton submit;  
 private Boolean isValid=false;  
 private DatabaseManager databaseManager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_task*);  
 databaseManager = new DatabaseManager(this);  
 initMain();  
 }  
 private void initMain(){  
 initEditableFields();  
 initCalendar();  
 initButtons();  
 }  
 private void initEditableFields(){  
 editText=findViewById(R.id.*editText*);  
 editDetails=findViewById(R.id.*multiline*);  
 }  
 private void initCalendar(){  
 calendarText=findViewById(R.id.*editDate*);  
 calendar=findViewById(R.id.*imageView*);  
 calendar.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 DatePickerDialog();  
 }  
 });  
 }  
 private void initButtons(){  
 priorityButton=findViewById(R.id.*important*);  
  
 submit=findViewById(R.id.*submit*);  
 submit.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 editText.clearFocus();  
 Validation();  
 if(isValid){  
 //SQLite  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 databaseManager.addTask(title,sdf.format(selectedDate),details,priority,"0");  
 initAlarm();  
 startActivity(new Intent(v.getContext(),MainActivity.class));  
 finish();  
 }  
  
 }  
 });  
 }  
 private void Validation(){  
 //extract data  
 title=editText.getText().toString().trim();  
 details=editDetails.getText().toString().trim();  
 if(title.matches("") || selectedDate==null){  
 Toast.*makeText*(getApplicationContext()  
 , "Fill in the required fields!"  
 ,Toast.*LENGTH\_SHORT*).show();  
 isValid=false;  
 }else{  
 isValid=true;  
 }  
 if(priorityButton.isChecked()){  
 priority="1";  
 }  
 }  
 private void DatePickerDialog(){  
 DatePickerDialog datePickerDialog=new DatePickerDialog(this,this,  
 Calendar.*getInstance*().get(Calendar.*YEAR*),  
 Calendar.*getInstance*().get(Calendar.*MONTH*),  
 Calendar.*getInstance*().get(Calendar.*DAY\_OF\_MONTH*));  
 datePickerDialog.show();  
 }  
  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 String temp=dayOfMonth+" "+(month+1)+" "+year;  
 SimpleDateFormat sdf=new SimpleDateFormat("d M yyyy");  
 try {  
 selectedDate=sdf.parse(temp);  
 sdf=new SimpleDateFormat("d-MMM-yy");  
 sdf.format(selectedDate);  
 calendarText.setText(sdf.format(selectedDate));  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void initAlarm() {  
 AlarmManager alarmManager= (AlarmManager) getSystemService(Context.*ALARM\_SERVICE*);  
 Intent intent = new Intent(this,AlarmReceiver.class);  
 PendingIntent alarmIntent = PendingIntent.*getBroadcast*(this,0,intent,PendingIntent.*FLAG\_UPDATE\_CURRENT*);  
  
 if(alarmManager!=null){  
 alarmManager.cancel(alarmIntent);  
 }  
 //Set time  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.setTimeInMillis(System.*currentTimeMillis*());  
 calendar.set(Calendar.*HOUR\_OF\_DAY*,8);  
 calendar.set(Calendar.*MINUTE*,0);  
 calendar.set(Calendar.*SECOND*,0);  
 Log.*d*("message",String.*valueOf*(calendar.getTime()));  
 alarmManager.setInexactRepeating(AlarmManager.*RTC\_WAKEUP*,calendar.getTimeInMillis(),AlarmManager.*INTERVAL\_DAY*,alarmIntent);  
  
  
 }  
  
 @Override  
 public void onBackPressed() {  
 startActivity(new Intent(this,MainActivity.class));  
 finish();  
 }  
}

EditTask

package com.example.task1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.CheckBox;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
  
public class EditTask extends AppCompatActivity implements DatePickerDialog.OnDateSetListener {  
  
 //Edit Text Fields  
 private EditText editText,editDetails;  
 private String title,details;  
  
 //Date Selection  
 private TextView calendarText;  
 private ImageView calendar;  
 private Date selectedDate;  
  
 //Buttons  
 private CheckBox priorityButton;  
 private String priority="0";  
  
 //Submission  
 private ImageButton submit;  
 private Boolean isValid=false;  
 private DatabaseManager databaseManager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_edit\_task*);  
 final Task task = getIntent().getExtras().getParcelable("edit\_task");  
 databaseManager=new DatabaseManager(this);  
 initMain(task);  
 }  
 private void initMain(Task task){  
 initEditableFields(task);  
 initCalendar(task);  
 initButtons(task);  
 }  
 private void initEditableFields(Task task){  
 editText=findViewById(R.id.*edit\_editText*);  
 editDetails=findViewById(R.id.*edit\_multiline*);  
 editText.setText(task.getTitle());  
 editDetails.setText(task.getDetails());  
 }  
  
 private void initCalendar(Task task){  
 calendarText=findViewById(R.id.*edit\_editDate*);  
 //date not yet  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 try {  
 selectedDate=sdf.parse(task.getDue\_date());  
 sdf= new SimpleDateFormat("d-MMM-yy");  
 calendarText.setText(sdf.format(selectedDate));  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 calendar=findViewById(R.id.*edit\_imageView*);  
 calendar.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 DatePickerDialog();  
 }  
 });  
 }  
 private void initButtons(final Task task){  
 priorityButton=findViewById(R.id.*edit\_important*);  
 if(task.getPriority().equals("1")){  
 priority=task.getPriority();  
 priorityButton.setChecked(true);  
 }  
 submit = findViewById(R.id.*edit\_submit*);  
 submit.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 editText.clearFocus();  
 Validation();  
 if(isValid){  
 //SQLite  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 databaseManager.updateTask(title,sdf.format(selectedDate),details,priority,String.*valueOf*(task.getId()));  
 startActivity(new Intent(v.getContext(),MainActivity.class));  
 finish();  
 }  
 }  
 });  
 }  
 private void Validation(){  
 title=editText.getText().toString().trim();  
 details=editDetails.getText().toString().trim();  
 if(title.matches("") || selectedDate==null){  
 Toast.*makeText*(getApplicationContext()  
 , "Fill in the required fields!"  
 ,Toast.*LENGTH\_SHORT*).show();  
 isValid=false;  
 }else{  
 isValid=true;  
 }  
 if(priorityButton.isChecked()){  
 priority="1";  
 }else{  
 priority="0";  
 }  
 }  
 private void DatePickerDialog(){  
 DatePickerDialog datePickerDialog=new DatePickerDialog(this,this,  
 Calendar.*getInstance*().get(Calendar.*YEAR*),  
 Calendar.*getInstance*().get(Calendar.*MONTH*),  
 Calendar.*getInstance*().get(Calendar.*DAY\_OF\_MONTH*));  
 datePickerDialog.show();  
 }  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 String temp=dayOfMonth+" "+(month+1)+" "+year;  
 SimpleDateFormat sdf=new SimpleDateFormat("d M yyyy");  
 try {  
 selectedDate=sdf.parse(temp);  
 sdf=new SimpleDateFormat("d-MMM-yy");  
 sdf.format(selectedDate);  
 calendarText.setText(sdf.format(selectedDate));  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
 @Override  
 public void onBackPressed() {  
 startActivity(new Intent(this,MainActivity.class));  
 finish();  
 }  
  
  
}

DeleteDialog

package com.example.task1;  
  
import android.app.AlertDialog;  
import android.app.Dialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatDialogFragment;  
  
import java.util.ArrayList;  
  
public class DeleteDialog extends AppCompatDialogFragment{  
  
 DatabaseManager databaseManager;  
 private int id;  
 private int position;  
 private Context context;  
  
 private OnFragmentInteractionListener mListener;  
 public DeleteDialog(Context context,int id,int position) {  
 this.context=context;  
 this.id = id;  
 this.position=position;  
 }  
  
 @Override  
 public Dialog onCreateDialog(Bundle savedInstanceState) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
 builder.setTitle("Deleting Task")  
 .setMessage("Do you confirm deletion?")  
 .setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 databaseManager=new DatabaseManager(context);  
 databaseManager.deleteTask(String.*valueOf*(id));  
 //this is where the dynamics begin!  
 onButtonPressed(position);  
  
 }  
 })  
 .setNegativeButton("No", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 }  
 });  
 return builder.create();  
 }  
 //Required to interact with MainActivity  
 public void onButtonPressed(int position) {  
 if (mListener != null) {  
 mListener.onFragmentInteraction(position);  
 }  
 }  
  
 @Override  
 public void onAttach(Context context) {  
 super.onAttach(context);  
 if (context instanceof OnFragmentInteractionListener) {  
 mListener = (OnFragmentInteractionListener) context;  
 } else {  
 throw new RuntimeException(context.toString()  
 + " must implement OnFragmentInteractionListener");  
 }  
 }  
  
 @Override  
 public void onDetach() {  
 super.onDetach();  
 mListener = null;  
 }  
 public interface OnFragmentInteractionListener {  
 void onFragmentInteraction(int position);  
 }  
  
  
  
}

CompletedActivity

package com.example.task1;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.view.MenuItem;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
import java.util.ArrayList;  
  
public class CompletedActivity extends AppCompatActivity {  
 RecyclerView recyclerView;  
 StatusRecyclerViewAdapter statusRecyclerViewAdapter;  
 ArrayList<Task> taskArrayList;  
 DatabaseManager databaseManager;  
 BottomNavigationView bottomNavigationView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_completed*);  
 initMain();  
 }  
 private void initMain(){  
 initDatabase();  
 initRecyclerView();  
 initNav();  
 }  
  
 private void initNav() {  
 bottomNavigationView = findViewById(R.id.*completed\_bottom\_nav*);  
 bottomNavigationView.getMenu().getItem(0).setChecked(true);  
 bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 switch (menuItem.getItemId()){  
 case R.id.*main\_view*:  
 startActivity(new Intent(getApplicationContext(),MainActivity.class));  
 finish();  
 break;  
 case R.id.*overdue\_view*:  
 startActivity(new Intent(getApplicationContext(),OverdueActivity.class));  
 finish();  
 break;  
 }  
 return false;  
 }  
 });  
 }  
  
 private void initRecyclerView() {  
 recyclerView=findViewById(R.id.*completed\_recyclerView*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
 statusRecyclerViewAdapter=new StatusRecyclerViewAdapter(taskArrayList,this);  
 recyclerView.setAdapter(statusRecyclerViewAdapter);  
 }  
  
 private void initDatabase(){  
 databaseManager = new DatabaseManager(this);  
 Cursor cursor = databaseManager.loadTasks();  
 taskArrayList=new ArrayList<>();  
 if(cursor.moveToFirst()){  
 do {  
 if(cursor.getString(5).equals("1")){  
 taskArrayList.add(new Task(  
 cursor.getInt(0),  
 cursor.getString(1),  
 cursor.getString(2),  
 cursor.getString(3),  
 cursor.getString(4),  
 cursor.getString(5)  
  
 ));  
 }  
  
 }while(cursor.moveToNext());  
 }  
 }  
}

OverdueActivity

package com.example.task1;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MenuItem;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
  
public class OverdueActivity extends AppCompatActivity {  
 RecyclerView recyclerView;  
 StatusRecyclerViewAdapter statusRecyclerViewAdapter;  
 ArrayList<Task> taskArrayList;  
 DatabaseManager databaseManager;  
 BottomNavigationView bottomNavigationView;  
 Date currentDate= Calendar.*getInstance*().getTime();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_overdue*);  
 initMain();  
 }  
 private void initMain(){  
 initDatabase();  
 initRecyclerView();  
 initNav();  
 }  
  
 private void initNav() {  
 bottomNavigationView = findViewById(R.id.*overdue\_bottom\_nav*);  
 bottomNavigationView.getMenu().getItem(2).setChecked(true);  
 bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 switch (menuItem.getItemId()){  
 case R.id.*main\_view*:  
 startActivity(new Intent(getApplicationContext(),MainActivity.class));  
 finish();  
 break;  
 case R.id.*completed\_view*:  
 startActivity(new Intent(getApplicationContext(),CompletedActivity.class));  
 finish();  
 break;  
 }  
 return false;  
 }  
 });  
 }  
  
 private void initRecyclerView() {  
 recyclerView=findViewById(R.id.*overdue\_recyclerView*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
 statusRecyclerViewAdapter=new StatusRecyclerViewAdapter(taskArrayList,this);  
 recyclerView.setAdapter(statusRecyclerViewAdapter);  
 }  
  
 private void initDatabase(){  
 databaseManager = new DatabaseManager(this);  
 Cursor cursor = databaseManager.loadTasks();  
 taskArrayList=new ArrayList<>();  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 if(cursor.moveToFirst()){  
 do {  
 try {  
 Date selectedDate = sdf.parse(cursor.getString(2));  
 long curdate=currentDate.getTime();  
 long seldate=selectedDate.getTime();  
 float result = (float)(seldate-curdate)/(24\*60\*60\*1000)+1;  
 if(result<0){  
 taskArrayList.add(new Task(  
 cursor.getInt(0),  
 cursor.getString(1),  
 cursor.getString(2),  
 cursor.getString(3),  
 cursor.getString(4),  
 cursor.getString(5)  
  
 ));  
 }  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }while(cursor.moveToNext());  
 }  
 }  
}

App(Channel)

package com.example.task1;  
  
import android.app.Application;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.os.Build;  
  
public class App extends Application {  
 public static final String *CHANNEL\_1\_ID* = "channel1";  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 createNotificationChannels();  
 }  
  
 private void createNotificationChannels() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel1 = new NotificationChannel(  
 *CHANNEL\_1\_ID*,  
 "Channel 1",  
 NotificationManager.*IMPORTANCE\_HIGH* );  
 channel1.setDescription("This is Channel 1");  
  
  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 manager.createNotificationChannel(channel1);  
 }  
 }  
}

AlarmReceiver

package com.example.task1;  
  
import android.app.Notification;  
import android.app.NotificationManager;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.provider.ContactsContract;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
  
public class AlarmReceiver extends BroadcastReceiver {  
 //Required variables to check if it is one day before due\_date, if true, notification  
 Date currentDate;  
 DatabaseManager databaseManager;  
 NotificationManagerCompat notificationManager;  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 Log.*d*("message","Accessed Alarm Receiver");  
 notificationManager = NotificationManagerCompat.*from*(context);  
 currentDate= Calendar.*getInstance*().getTime();  
 databaseManager = new DatabaseManager(context);  
 Cursor cursor = databaseManager.loadTasks();  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 if(cursor.moveToFirst()){  
 do{  
 try {  
 Date due\_date = sdf.parse(cursor.getString(2));  
 long curDate=currentDate.getTime();  
 long selDate=due\_date.getTime();  
 int result=(int)((selDate-curDate)/(24\*60\*60\*1000))+1;  
 if(result==1 && cursor.getString(5).equals("0")){  
 Log.*d*("message",cursor.getString(1));  
 Notification notification = new NotificationCompat.Builder(context, App.*CHANNEL\_1\_ID*)  
 .setSmallIcon(R.drawable.*home*)  
 .setContentTitle("Due Soon!")  
 .setContentText(cursor.getString(1))  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setCategory(NotificationCompat.*CATEGORY\_MESSAGE*)  
 .build();  
  
 notificationManager.notify(cursor.getInt(0), notification);  
 }  
  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }while(cursor.moveToNext());  
  
 }  
 }  
  
}

DatabaseManager

package com.example.task1;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.PriorityQueue;  
  
public class DatabaseManager extends SQLiteOpenHelper {  
  
 private static final String *DATABASE\_NAME*="taskDB";  
 private static final int *DATABASE\_VERSION* =1;  
 private static final String *TABLE\_NAME* = "tasks";  
 private static final String *COLUMN\_ID* = "id";  
 private static final String *COLUMN\_TITLE* = "title";  
 private static final String *COLUMN\_DUE\_DATE* = "due\_date";  
 private static final String *COLUMN\_DETAILS* = "details";  
 private static final String *COLUMN\_PRIORITY* = "priority";  
 private static final String *COLUMN\_COMPLETED* = "completed";  
  
  
 public DatabaseManager(Context context){  
 super(context,*DATABASE\_NAME*,null,*DATABASE\_VERSION*);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String sql = "CREATE TABLE IF NOT EXISTS "+*TABLE\_NAME*+" (\n" +  
 " "+*COLUMN\_ID*+" integer NOT NULL PRIMARY KEY AUTOINCREMENT,\n" +  
 " "+*COLUMN\_TITLE*+" varchar(250) NOT NULL,\n" +  
 " "+*COLUMN\_DUE\_DATE*+" DATETIME NOT NULL,\n" +  
 " "+*COLUMN\_DETAILS*+" varchar(250) ,\n" +  
 " "+*COLUMN\_PRIORITY*+" varchar(1) NOT NULL,\n" +  
 " "+*COLUMN\_COMPLETED*+" varchar(1) NOT NULL\n" +  
 ");";  
 db.execSQL(sql);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 }  
  
 boolean addTask(String title,String due\_date,String details,String priority,String completed){  
 SQLiteDatabase sqLiteDatabase = getWritableDatabase();  
  
 ContentValues cv = new ContentValues();  
  
 cv.put(*COLUMN\_TITLE*,title);  
 cv.put(*COLUMN\_DUE\_DATE*,due\_date);  
 cv.put(*COLUMN\_DETAILS*,details);  
 cv.put(*COLUMN\_PRIORITY*,priority);  
 cv.put(*COLUMN\_COMPLETED*,completed);  
 sqLiteDatabase.insert(*TABLE\_NAME*,null,cv);  
 return true;  
 }  
  
 Cursor loadTasks(){  
 SQLiteDatabase sqLiteDatabase = getReadableDatabase();  
 return sqLiteDatabase.rawQuery("SELECT \* FROM "+*TABLE\_NAME*+" ORDER BY "+*COLUMN\_DUE\_DATE*+" ASC",null);  
 }  
  
 boolean updateTask(String title,String due\_date,String details, String priority,String id){  
 SQLiteDatabase sqLiteDatabase = getWritableDatabase();  
  
 ContentValues cv = new ContentValues();  
  
 cv.put(*COLUMN\_TITLE*,title);  
 cv.put(*COLUMN\_DUE\_DATE*,due\_date);  
 cv.put(*COLUMN\_DETAILS*,details);  
 cv.put(*COLUMN\_PRIORITY*,priority);  
 sqLiteDatabase.update(*TABLE\_NAME*,cv,*COLUMN\_ID*+"=?",new String[]{id});  
 return true;  
 }  
 boolean updateTask(String id){  
 SQLiteDatabase sqLiteDatabase = getWritableDatabase();  
  
 ContentValues cv = new ContentValues();  
  
 cv.put(*COLUMN\_COMPLETED*,"1");  
 sqLiteDatabase.update(*TABLE\_NAME*,cv,*COLUMN\_ID*+"=?",new String[]{id});  
 return true;  
 }  
  
 boolean deleteTask(String id){  
 SQLiteDatabase sqLiteDatabase = getWritableDatabase();  
 sqLiteDatabase.delete(*TABLE\_NAME*,*COLUMN\_ID*+"=?",new String[]{id});  
 return true;  
 }  
}

RecyclerViewAdapter

package com.example.task1;  
  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.database.sqlite.SQLiteDatabase;  
import android.graphics.Color;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.cardview.widget.CardView;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentActivity;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.Locale;  
  
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder>{  
 SQLiteDatabase sqLiteDatabase;  
 //DatabaseManager databaseManager;  
 //Required variables  
 ArrayList<Task> taskArrayList;  
 Context context;  
  
 //Date  
 Date currentDate = Calendar.*getInstance*().getTime();  
 Date selectedDate;  
  
  
 public RecyclerViewAdapter(ArrayList<Task> taskArrayList, Context context) {  
 this.taskArrayList = taskArrayList;  
 this.context = context;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*task\_layout*,parent,false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, final int position) {  
 final Task task =taskArrayList.get(position);  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 SimpleDateFormat sdf\_day= new SimpleDateFormat("d");  
 SimpleDateFormat sdf\_month = new SimpleDateFormat("MMM");  
 try {  
 String temp = task.getDue\_date();  
 selectedDate = sdf.parse(temp);  
 //Log.d("after parse date",String.valueOf(selectedDate));  
 holder.day.setText(sdf\_day.format(selectedDate));  
 holder.month.setText(sdf\_month.format(selectedDate));  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 //Status icon and duration initialization  
 statusCheck(holder);  
 holder.task\_title.setText(task.getTitle());  
 if(task.getPriority().equals("1")){  
 int resID=context.getResources().getColor(R.color.*colorImportant*);  
 holder.cardView.setBackgroundColor(resID);  
 }  
 holder.cardView.setOnLongClickListener(new View.OnLongClickListener(){  
 @Override  
 public boolean onLongClick(View v) {  
 //Dialog  
 DeleteDialog deleteDialog = new DeleteDialog(context,task.getId(),position);  
 deleteDialog.show(((FragmentActivity)context).getSupportFragmentManager()  
 ,"Delete Dialog Fragment");  
 return false;  
 }  
 });  
 holder.cardView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 //to edit task page,send as parcelable  
 Intent intent = new Intent(context,EditTask.class);  
 intent.putExtra("edit\_task",taskArrayList.get(position));  
 context.startActivity(intent);  
 ((Activity)context).finish();  
 }  
 });  
 holder.pending\_switch.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 //Toast.makeText(context,"I am completed",Toast.LENGTH\_SHORT).show();  
 //databaseManager = new DatabaseManager(context);  
 sqLiteDatabase=context.openOrCreateDatabase(MainActivity.*DATABASE\_NAME*,Context.*MODE\_PRIVATE*,null);  
 String sql = "UPDATE tasks SET completed='1' WHERE id="+taskArrayList.get(position).getId();  
 sqLiteDatabase.execSQL(sql);  
 //Log.d("completed?",taskArrayList.get(position).getCompleted());  
 taskArrayList.remove(position);  
 notifyItemRemoved(position);  
 notifyItemRangeChanged(position,taskArrayList.size());  
 }  
 });  
 }  
  
 private void statusCheck(ViewHolder holder){  
 long curdate=currentDate.getTime();  
 long seldate=selectedDate.getTime();  
 float result = (float)(seldate-curdate)/(24\*60\*60\*1000)+1;  
 String test = String.*format*("%.2f",result);  
 String[] split = test.split("\\.");  
 int days = Integer.*parseInt*(split[0]);  
 split[1]="0."+split[1];  
 float hours = Float.parseFloat(split[1]);  
 hours = hours\*24;  
 int hr= (int)hours+1;  
 if(result<0){  
 int resID= context.getResources().getIdentifier("clipboard\_red1"  
 ,"drawable"  
 ,context.getPackageName());  
 holder.status\_icon.setImageResource(resID);  
 holder.duration.setText("0 hours");  
 holder.day.setTextColor(Color.RED);  
 holder.month.setTextColor(Color.RED);  
 holder.task\_title.setTextColor(Color.RED);  
  
 }else if(days==0){  
 int resID= context.getResources().getIdentifier("clipboard\_orange1"  
 ,"drawable"  
 ,context.getPackageName());  
 holder.status\_icon.setImageResource(resID);  
 holder.duration.setText(days+" days and "+hr+" hours");  
  
 }else{  
 int resID= context.getResources().getIdentifier("clipboard\_blue1"  
 ,"drawable"  
 ,context.getPackageName());  
 holder.status\_icon.setImageResource(resID);  
 holder.duration.setText(days+" days and "+hr+" hours");  
 }  
 }  
 @Override  
 public int getItemCount() {  
 return taskArrayList.size();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder{  
 //CardView  
 private CardView cardView;  
 //TextView  
 private TextView day,month,task\_title,duration;  
 //ImageView  
 private ImageView status\_icon;  
 //Switch  
 private Switch pending\_switch;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 cardView=itemView.findViewById(R.id.cardView);  
 day=itemView.findViewById(R.id.day);  
 month=itemView.findViewById(R.id.month);  
 task\_title=itemView.findViewById(R.id.task\_title);  
 duration=itemView.findViewById(R.id.duration);  
 status\_icon=itemView.findViewById(R.id.status\_icon);  
 pending\_switch=itemView.findViewById(R.id.pending\_switch);  
 }  
 }  
  
}

StatusRecyclerViewAdapter

package com.example.task1;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Switch;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
  
public class StatusRecyclerViewAdapter extends RecyclerView.Adapter<StatusRecyclerViewAdapter.ViewHolder>{  
 //Required variables  
 ArrayList<Task> taskArrayList;  
 Context context;  
  
 //Date  
 Date selectedDate;  
  
 public StatusRecyclerViewAdapter(ArrayList<Task> taskArrayList, Context context) {  
 this.taskArrayList = taskArrayList;  
 this.context = context;  
 }  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*completed\_task\_layout*,parent,false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 final Task task = taskArrayList.get(position);  
 holder.title.setText(task.getTitle());  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");  
 SimpleDateFormat sdf\_format=new SimpleDateFormat("d-MMM-yy");  
 try {  
 String temp = task.getDue\_date();  
 selectedDate = sdf.parse(temp);  
 Date currentDate = Calendar.*getInstance*().getTime();  
 long curdate=currentDate.getTime();  
 long seldate=selectedDate.getTime();  
 float result = (float)(seldate-curdate)/(24\*60\*60\*1000)+1;  
 holder.date.setText(sdf\_format.format(selectedDate));  
 if(result<0){  
 holder.aSwitch.setText("Not Completed");  
 }  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 if(task.getCompleted().equals("0")){  
 holder.aSwitch.setChecked(true);  
 }  
  
 }  
  
 @Override  
 public int getItemCount() {  
 return taskArrayList.size();  
 }  
  
 public class ViewHolder extends RecyclerView.ViewHolder{  
 private TextView title,date;  
 private Switch aSwitch;  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 title=itemView.findViewById(R.id.*title\_completed*);  
 date = itemView.findViewById(R.id.*datetitle\_completed*);  
 aSwitch=itemView.findViewById(R.id.*switch\_completed*);  
 }  
 }  
}

Task

package com.example.task1;  
  
import android.os.Parcel;  
import android.os.Parcelable;  
  
public class Task implements Parcelable {  
 private int id;  
 private String title,due\_date,details,priority,completed;  
  
 public Task(int id, String title, String due\_date, String details, String priority, String completed) {  
 this.id = id;  
 this.title = title;  
 this.due\_date = due\_date;  
 this.details = details;  
 this.priority = priority;  
 this.completed = completed;  
 }  
  
 protected Task(Parcel in) {  
 id = in.readInt();  
 title = in.readString();  
 due\_date = in.readString();  
 details = in.readString();  
 priority = in.readString();  
 completed = in.readString();  
 }  
  
 public static final Creator<Task> *CREATOR* = new Creator<Task>() {  
 @Override  
 public Task createFromParcel(Parcel in) {  
 return new Task(in);  
 }  
  
 @Override  
 public Task[] newArray(int size) {  
 return new Task[size];  
 }  
 };  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public String getDue\_date() {  
 return due\_date;  
 }  
  
 public void setDue\_date(String due\_date) {  
 this.due\_date = due\_date;  
 }  
  
 public String getDetails() {  
 return details;  
 }  
  
 public void setDetails(String details) {  
 this.details = details;  
 }  
  
 public String getPriority() {  
 return priority;  
 }  
  
 public void setPriority(String priority) {  
 this.priority = priority;  
 }  
  
 public String getCompleted() {  
 return completed;  
 }  
  
 public void setCompleted(String completed) {  
 this.completed = completed;  
 }  
  
 @Override  
 public int describeContents() {  
 return 0;  
 }  
  
 @Override  
 public void writeToParcel(Parcel dest, int flags) {  
 dest.writeInt(id);  
 dest.writeString(title);  
 dest.writeString(due\_date);  
 dest.writeString(details);  
 dest.writeString(priority);  
 dest.writeString(completed);  
 }  
}

### **Task 2**

Screenshot of application

A screenshot of a cell phone

Description automatically generated

Java Codes for MainActivity

package com.example.task2;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.ContextCompat;  
  
import android.Manifest;  
import android.app.Service;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.location.LocationListener;  
import android.location.LocationManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity{  
 public static int *count*=2;  
 Button start,stop;  
 EditText input;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 input = findViewById(R.id.*editText\_interval*);  
 if(!initPermissions()){  
 initButtons();  
 }  
 }  
  
 private boolean initPermissions() {  
 if(Build.VERSION.*SDK\_INT*>=23 && ContextCompat.*checkSelfPermission*(this,Manifest.permission.*ACCESS\_FINE\_LOCATION*) !=PackageManager.*PERMISSION\_GRANTED* && ContextCompat.*checkSelfPermission*(this,Manifest.permission.*ACCESS\_COARSE\_LOCATION*) !=PackageManager.*PERMISSION\_GRANTED*){  
 requestPermissions(new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*,Manifest.permission.*ACCESS\_COARSE\_LOCATION*},100);  
 return true;  
 }  
 return false;  
 }  
  
 private void initButtons(){  
 start=findViewById(R.id.*button\_start*);  
 stop = findViewById(R.id.*button\_stop*);  
  
 start.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //intent pass the variable to set how long in minutes to service class  
 if(input.getText().toString().matches("")){  
 Toast.*makeText*(getApplicationContext(),"Please input a duration!",Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Intent i = new Intent(getApplicationContext(),MyService.class);  
 i.putExtra("minute",input.getText().toString());  
 startService(i);  
 start.setEnabled(false);  
 stop.setEnabled(true);  
 }  
  
 }  
 });  
 stop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 stopService(new Intent(getApplicationContext(),MyService.class));  
 start.setEnabled(true);  
 stop.setEnabled(false);  
  
 }  
 });  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if(requestCode==100){  
 if(grantResults[0]==PackageManager.*PERMISSION\_GRANTED* && grantResults[1]==PackageManager.*PERMISSION\_GRANTED*){  
 initButtons();  
 }else{  
 initPermissions();  
 }  
 }  
 }  
}

Java Codes for MyService

package com.example.task2;  
  
import android.Manifest;  
import android.app.IntentService;  
import android.app.Service;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.location.LocationListener;  
import android.location.LocationManager;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.IBinder;  
import android.provider.Settings;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
  
import com.android.volley.AuthFailureError;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
  
import java.net.URL;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
public class MyService extends Service {  
 private Date currentTime;  
 private LocationManager locationManager;  
 private LocationListener locationListener;  
 private String input;  
 private ArrayList<String> stringArrayList;  
 private String url = "http://10.0.2.2/assignment5/insert.php";  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Log.*d*("message","onStartCommand is called");  
 input = (String)intent.getExtras().get("minute");  
 Log.*d*("message",input);  
 stringArrayList= new ArrayList<>();  
 locationListener = new LocationListener() {  
 @Override  
 public void onLocationChanged(Location location) {  
 //Do write to database here  
 if(MainActivity.*count*!=0){  
 currentTime = Calendar.*getInstance*().getTime();  
 final SimpleDateFormat sdf = new SimpleDateFormat("YYYY-MM-d HH:mm:ss");  
 Log.*d*("message",sdf.format(currentTime)+" "+"Long: "+location.getLongitude()+" Lat: "+location.getLatitude());  
 //stringArrayList.add(sdf.format(currentTime)+" "+"Long: "+location.getLongitude()+" Lat: "+location.getLatitude());  
 insertDB(sdf.format(currentTime)+" "+"Long: "+location.getLongitude()+" Lat: "+location.getLatitude());  
 MainActivity.*count*--;  
 }else{  
 Log.*d*("message","Nothing anymore :D");  
  
 }  
 }  
  
 @Override  
 public void onStatusChanged(String s, int i, Bundle bundle) {  
  
 }  
  
 @Override  
 public void onProviderEnabled(String s) {  
  
 }  
  
 @Override  
 public void onProviderDisabled(String s) {  
 Intent i = new Intent(Settings.*ACTION\_LOCATION\_SOURCE\_SETTINGS*);  
 i.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(i);  
 }  
 };  
 locationManager = (LocationManager) getApplicationContext().getSystemService(Context.*LOCATION\_SERVICE*);  
 if (checkSelfPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && checkSelfPermission(Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 return super.onStartCommand(intent, flags, startId);  
 }  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*, 1000\*60\*Integer.*parseInt*(input), 0, locationListener);  
  
 return super.onStartCommand(intent, flags, startId);  
 }  
 private void insertDB(final String string){  
 Log.*d*("inside insertDB",string);  
 StringRequest stringRequest = new StringRequest(Request.Method.*POST*, url  
 , new Response.Listener<String>() {  
 @Override  
 public void onResponse(String response) {  
  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 Toast.*makeText*(getApplicationContext(),error.getMessage(),Toast.*LENGTH\_SHORT*).show();  
 }  
 }){  
 @Override  
 protected Map<String, String> getParams() throws AuthFailureError {  
 Map<String,String> params = new HashMap<>();  
 params.put("location",string);  
 return params;  
 }  
 };  
  
 RequestQueue requestQueue = Volley.*newRequestQueue*(this);  
 requestQueue.add(stringRequest);  
 }  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onCreate() {  
 Log.*d*("message","Service started");  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 Log.*d*("message","Service stopped");  
 if(locationManager!=null){  
 stringArrayList.clear();  
 locationManager.removeUpdates(locationListener);  
 MainActivity.*count*=2;  
 }  
 }  
}