1/19/2022

Mobile Application Development

Lab-9 Manual

Safdar Hussain, BSCS-VII-A

CMS: 023-18-0059

**Activity 1: Create a notification shown like attached picture**

**MainActivity.java**

package com.example.myapplication;  
  
import androidx.annotation.RequiresApi;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.NotificationCompat;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.RemoteViews;  
  
public class MainActivity extends AppCompatActivity {  
 CustomNotification custom\_notification;  
 public static final String *CHANNEL\_ID*="ForegroundServiceChannel";  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 custom\_notification = createNotification();  
  
  
 }  
 public void ShowNotification(View view){  
 custom\_notification.getNotificationManager().notify(1234,custom\_notification.getNotification());  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public CustomNotification createNotification(){  
  
 *//Creating notification Manager* NotificationManager notificationManager = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);  
 *//Creating notification channel* NotificationChannel notificationChannel = new NotificationChannel(*CHANNEL\_ID*,"custom\_notification",NotificationManager.*IMPORTANCE\_DEFAULT*);  
 *//create channel* notificationManager.createNotificationChannel(notificationChannel);  
 *//Creating notification custom layout holder* RemoteViews collapsedView = new RemoteViews(getPackageName(),R.layout.*collapse\_notification*);  
 RemoteViews expandedView = new RemoteViews(getPackageName(),R.layout.*expand\_notification*);  
 *//setting text view and other view values.  
// collapsedView.setTextViewText(R.id.text\_view\_collapsed\_1, "Hello Zeeshan!");  
// expandedView.setImageViewResource(R.id.image\_view\_expanded, R.drawable.ds);  
 //Creating notification builder* NotificationCompat.Builder builder = new NotificationCompat.Builder(this,*CHANNEL\_ID*)  
 .setContentTitle("Customized notification")  
 .setSmallIcon(R.drawable.*ic\_launcher\_foreground*)  
 .setCustomContentView(collapsedView)  
 .setCustomBigContentView(expandedView);  
 *//Creating notification* Notification notification = builder.build();  
 CustomNotification custom\_notification = new CustomNotification(notification,notificationManager);  
 return custom\_notification;  
 }  
}

**CustomNotification.java**

package com.example.myapplication;  
  
import android.app.Notification;  
import android.app.NotificationManager;  
  
public class CustomNotification {  
 Notification notification;  
 NotificationManager notificationManager;  
  
 public CustomNotification(Notification notification, NotificationManager notificationManager) {  
 this.notification = notification;  
 this.notificationManager = notificationManager;  
 }  
  
 public Notification getNotification() {  
 return notification;  
 }  
  
 public void setNotification(Notification notification) {  
 this.notification = notification;  
 }  
  
 public NotificationManager getNotificationManager() {  
 return notificationManager;  
 }  
  
 public void setNotificationManager(NotificationManager notificationManager) {  
 this.notificationManager = notificationManager;  
 }  
}

**Activity 2: Create an SMS reader app.**

Which receives all the incoming messages and shown them on screen in either Toast or A label along with the sender’s information.

MainActivity.java

package com.example.messagelistener;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import android.Manifest;  
import android.content.IntentFilter;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.ListView;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
 List<MessageDataHolder> messageDataHolderList;  
 CustomAdapter customAdapter;  
 ListView listView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 listView = findViewById(R.id.list\_of\_messages);  
 checkForSmsReceivePermissions();  
 messageDataHolderList = new ArrayList<>();  
 customAdapter = new CustomAdapter(this,R.layout.list\_item,messageDataHolderList);  
  
 MyReceiver BR\_smsreceiver = null;  
 BR\_smsreceiver = new MyReceiver();  
 BR\_smsreceiver.setMainActivityHandler(this);  
 IntentFilter fltr\_smsreceived = new IntentFilter("android.provider.Telephony.SMS\_RECEIVED");  
 registerReceiver(BR\_smsreceiver,fltr\_smsreceived);  
 }  
  
  
 void checkForSmsReceivePermissions(){  
 *// Check if App already has permissions for receiving SMS* if(ContextCompat.checkSelfPermission(getBaseContext(), "android.permission.RECEIVE\_SMS") == PackageManager.PERMISSION\_GRANTED) {  
 *// App has permissions to listen incoming SMS messages* Log.d("adnan", "checkForSmsReceivePermissions: Allowed");  
 } else {  
 *// App don't have permissions to listen incoming SMS messages* Log.d("Aslam", "checkForSmsReceivePermissions: Denied");  
  
 *// Request permissions from user* ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.RECEIVE\_SMS}, 43391);  
 }  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if(requestCode == 43391){  
 if(grantResults.length>0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){  
 Log.d("Aslam", "Sms Receive Permissions granted");  
 } else {  
 Log.d("Aslam", "Sms Receive Permissions denied");  
 }  
 }  
 }  
  
  
 public void addNewMessage(String from, String msg) {  
 messageDataHolderList.add(new MessageDataHolder(from,msg));  
 listView.setAdapter(customAdapter);  
 Log.d("message",from+msg);  
 }  
}

MyReceiver.java

package com.example.messagelistener;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.provider.Telephony;  
import android.telephony.SmsMessage;  
import android.util.Log;  
import android.widget.Toast;  
  
public class MyReceiver extends BroadcastReceiver {  
  
 MainActivity main = null;  
 void setMainActivityHandler(MainActivity main){  
 this.main=main;  
 Log.d("message","updated main");  
 }  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
  
 if(intent.getAction().equals("android.provider.Telephony.SMS\_RECEIVED")){  
 Bundle bundle = intent.getExtras(); *//---get the SMS message passed in---* SmsMessage[] msgs = null;  
 String msg\_from = null;  
 String msgBody = null;  
 if (bundle != null){  
 *//---retrieve the SMS message received---* try{  
 Object[] pdus = (Object[]) bundle.get("pdus");  
 msgs = new SmsMessage[pdus.length];  
 for(int i=0; i<msgs.length; i++){  
 msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);  
 msg\_from = msgs[i].getOriginatingAddress();  
 msgBody = msgs[i].getMessageBody();  
  
 }  
  
 main.addNewMessage(msg\_from,msgBody);  
  
  
 }catch(Exception e){  
 Log.d("Exception caught",e.getMessage());  
 }  
 }  
 }  
 }  
}

CustomAdapter.java

package com.example.messagelistener;  
  
import android.app.Activity;  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import org.w3c.dom.Text;  
  
import java.util.List;  
  
public class CustomAdapter extends ArrayAdapter<MessageDataHolder> {  
 int resource;  
 Context context;  
 public CustomAdapter(@NonNull Context context, int resource, @NonNull List<MessageDataHolder> objects) {  
 super(context, resource, objects);  
 this.resource=resource;  
 this.context=context;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater layoutInflater = LayoutInflater.from(context);  
 convertView = layoutInflater.inflate(resource,null);  
  
 MessageDataHolder messageDataHolder = getItem(position);  
 TextView txt\_from = convertView.findViewById(R.id.txt\_from);  
 TextView txt\_msg = convertView.findViewById(R.id.txt\_msg);  
 txt\_msg.setText(messageDataHolder.getMessage());  
 txt\_from.setText(messageDataHolder.getFrom());  
  
 return convertView;  
 }  
}

MessageData.java

package com.example.messagelistener;  
  
public class MessageDataHolder {  
 String from,message;  
  
 public MessageDataHolder(String from, String message) {  
 this.from = from;  
 this.message = message;  
 }  
  
 public String getFrom() {  
 return from;  
 }  
  
 public void setFrom(String from) {  
 this.from = from;  
 }  
  
 public String getMessage() {  
 return message;  
 }  
  
 public void setMessage(String message) {  
 this.message = message;  
 }  
}

**Activity 3: Create a user dictionary and make it available so that other applications can read and**

**Update the words in the dictionary.**

**MainActivity.java**

package com.example.android.dictionary;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.RequiresApi;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
  
 ArrayList<String> listItems = new ArrayList<String>();  
 ArrayAdapter<String> adapter;  
 ListView list;  
  
  
 @RequiresApi(api = Build.VERSION\_CODES.*JELLY\_BEAN*)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 setTitle("Dictionary");  
  
 list = (ListView) findViewById(R.id.*listView*);  
  
 adapter = new ArrayAdapter<>(this,  
 R.layout.*support\_simple\_spinner\_dropdown\_item*,listItems);  
  
 Cursor cursor = getContentResolver().query(Uri.*parse*("content://com.example.android.dictionary.WordProvider/words"),  
 null,  
 null,  
 null,  
 null,  
 null);  
  
 listItems.removeAll(listItems);  
 if (cursor.getCount() == 0) {  
 *// show message that no record found* Toast.*makeText*(getBaseContext(), "No Words Found", Toast.*LENGTH\_LONG*).show();  
 return;  
 }  
  
 while (cursor.moveToNext()) {  
 listItems.add(cursor.getInt(0)+" - "+cursor.getString(1));  
 }  
  
 adapter.notifyDataSetChanged();  
  
  
 }  
  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.*menu*,menu);  
  
 return super.onCreateOptionsMenu(menu);  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
  
 if (item.getItemId() == R.id.*add\_word*)  
 {  
  
 }  
  
 else if (item.getItemId() == R.id.*search*)  
 {  
  
 }  
  
  
 return super.onOptionsItemSelected(item);  
 }  
}

WordProvider.java

package com.example.android.dictionary;  
  
import android.content.ContentProvider;  
import android.content.ContentUris;  
import android.content.ContentValues;  
import android.content.Context;  
import android.content.UriMatcher;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteException;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.database.sqlite.SQLiteQueryBuilder;  
import android.net.Uri;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import java.util.HashMap;  
  
public class WordProvider extends ContentProvider {  
  
 static final String *PROVIDER\_NAME* = "com.example.android.dictionary.WordProvider";  
 static final String *URL* = "content://"+ *PROVIDER\_NAME* + "/words";  
 static final Uri *CONTENT\_URI* = Uri.*parse*(*URL*);  
  
  
 static final String *id* = "ID";  
 static final String *word* = "WORD";  
 static final String *meaning* = "MEANING";  
 static final int *uriCode* = 1;  
 static final UriMatcher *uriMatcher*;  
 private static HashMap<String, String> *values*;  
  
 static {  
 *uriMatcher* = new UriMatcher(UriMatcher.*NO\_MATCH*);  
 *uriMatcher*.addURI(*PROVIDER\_NAME*, "words", *uriCode*);  
 *uriMatcher*.addURI(*PROVIDER\_NAME*, "words/\*", *uriCode*);  
 }  
  
 @Override  
 public boolean onCreate() {  
  
 Context context = getContext();  
 DatabaseHelper dbHelper = new DatabaseHelper(context);  
  
 db = dbHelper.getWritableDatabase();  
  
 if (db != null)  
 return true;  
  
 return false;  
 }  
  
 @Nullable  
 @Override  
 public Cursor query(@NonNull Uri uri, @Nullable String[] projection, @Nullable String selection, @Nullable String[] selectionArgs, @Nullable String sortOrder) {  
  
 SQLiteQueryBuilder qb = new SQLiteQueryBuilder();  
 qb.setTables(*TABLE\_NAME*);  
  
 switch (*uriMatcher*.match(uri))  
 {  
 case *uriCode*:  
 qb.setProjectionMap(*values*);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI "+ uri);  
 }  
  
 if (sortOrder == null || sortOrder == "")  
 sortOrder = *id*;  
  
 Cursor c = qb.query(db,projection,selection,selectionArgs,null,null,sortOrder);  
 c.setNotificationUri(getContext().getContentResolver(),uri);  
  
 return c;  
 }  
  
 @Nullable  
 @Override  
 public String getType(@NonNull Uri uri) {  
  
 switch (*uriMatcher*.match(uri)) {  
 case *uriCode*:  
 return "vnd.android.cursor.dir/words";  
 default:  
 throw new IllegalArgumentException("Unsupported URI: " + uri);  
 }  
 }  
  
 @Nullable  
 @Override  
 public Uri insert(@NonNull Uri uri, @Nullable ContentValues values) {  
  
 long rowID = db.insert(*TABLE\_NAME*,null,values);  
  
 if(rowID > 0)  
 {  
 Uri \_uri = ContentUris.*withAppendedId*(*CONTENT\_URI*,rowID);  
 getContext().getContentResolver().notifyChange(\_uri,null);  
 return \_uri;  
 }  
  
 throw new SQLiteException("Failed to insert into "+uri);  
 }  
  
 @Override  
 public int delete(@NonNull Uri uri, @Nullable String selection, @Nullable String[] selectionArgs) {  
  
 int count = 0;  
  
 switch (*uriMatcher*.match(uri))  
 {  
 case *uriCode*:  
 count = db.delete(*TABLE\_NAME*,selection,selectionArgs);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI "+uri);  
 }  
  
 getContext().getContentResolver().notifyChange(uri,null);  
 return count;  
 }  
  
 @Override  
 public int update(@NonNull Uri uri, @Nullable ContentValues values, @Nullable String selection, @Nullable String[] selectionArgs) {  
  
 int count = 0;  
  
 switch (*uriMatcher*.match(uri))  
 {  
 case *uriCode*:  
 count = db.update(*TABLE\_NAME*,values,selection,selectionArgs);  
 break;  
 default:  
 throw new IllegalArgumentException("Unknown URI "+uri);  
 }  
  
 getContext().getContentResolver().notifyChange(uri,null);  
 return count;  
 }  
  
  
 private SQLiteDatabase db;  
 public static final String *DATABASE\_NAME* = "Dictionary.db";  
 public static final String *TABLE\_NAME* = "WordsTable";  
 static final int *DATABASE\_VERSION* = 1;  
 static final String *CREATE\_DB\_TABLE* = " CREATE TABLE " + *TABLE\_NAME* + " (ID INTEGER PRIMARY KEY AUTOINCREMENT, "  
 + " WORD VARCHAR NOT NULL, MEANING VARCHAR NOT NULL);";  
  
 private static class DatabaseHelper extends SQLiteOpenHelper{  
  
 DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(*CREATE\_DB\_TABLE*);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS "+*CREATE\_DB\_TABLE*);  
 onCreate(db);  
 }  
 }  
  
  
  
}

