

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

Задание 2 – Умные часы

Выполнил: Швец Александр Евгеньевич

Группа: ПР-31

Преподаватель: Мирошниченко Г.В.

2023

## **Описание задачи**

1.Cоздать функциональное приложение по макету, используя фрагменты и неявные интенты.  
2.Загрузить проект в репрозиторий с именем zd4\_FI.  
3. Сделать отчет.  
4. Ссылку на репрозиторий выгрузить в гугл-класс.

**Листинг**

*MainActivity*

package com.example.watchappvariant7  
  
import android.content.Context  
import android.content.Intent  
import android.content.SharedPreferences  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var loginEditText: EditText  
 private lateinit var passwordEditText: EditText  
 private lateinit var sharedPreferences: SharedPreferences  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
  
 loginEditText = findViewById(R.id.loginEditText)  
 passwordEditText = findViewById(R.id.passwordEditText)  
  
 sharedPreferences = getSharedPreferences("user\_credentials", Context.MODE\_PRIVATE)  
  
 val loginButton: Button = findViewById(R.id.loginButton)  
 loginButton.setOnClickListener **{** val login = loginEditText.text.toString()  
 val password = passwordEditText.text.toString()  
 val savedLogin = sharedPreferences.getString("login",null)  
 val savedPassword = sharedPreferences.getString("password",null)  
 if(savedLogin==null) {  
  
 // Сохраняем логин и пароль в SharedPreferences  
 saveCredentials(login, password)  
 }  
 else{  
 if(savedLogin!=login||savedPassword!=password){  
 Toast.makeText(this,"Неверный логин или пароль",Toast.LENGTH\_SHORT).show()  
 return@setOnClickListener  
 }  
 }  
  
 // Переходим на второй экран  
 val intent = Intent(this, SecondActivity::class.java)  
 startActivity(intent)  
 **}** }  
  
 private fun saveCredentials(login: String, password: String) {  
 val editor = sharedPreferences.edit()  
 editor.putString("login", login)  
 editor.putString("password", password)  
 editor.apply()  
 }  
}

*SecondActivity*

package com.example.watchappvariant7  
  
import android.content.Intent  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import androidx.appcompat.app.AppCompatActivity  
  
class SecondActivity : AppCompatActivity() {  
  
 private lateinit var artistEditText: EditText  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_second*)  
  
 artistEditText = findViewById(R.id.*artistEditText*)  
  
 val nextButton: Button = findViewById(R.id.*nextButton*)  
 val historyButton:Button = findViewById(R.id.*historyButton*)  
 nextButton.setOnClickListener **{** val artistName = artistEditText.*text*.toString()  
  
 // Переходим на третий экран и передаем название исполнителя  
 val intent = Intent(this, ThirdActivity::class.*java*)  
 intent.putExtra("artistName", artistName)  
 startActivity(intent)  
 **}** historyButton.setOnClickListener**{** startActivity(Intent(this,SearchHistoryActivity::class.*java*))  
 **}** }  
}

*CrimeTypeConverters*

package com.example.crimese.database  
  
import androidx.room.TypeConverter  
import java.util.\*  
  
class CrimeTypeConverters {  
 @TypeConverter  
 fun fromDate(date: Date?): Long? {  
 return date?.*time* }  
 @TypeConverter  
 fun toDate(millisSinceEpoch: Long?): Date?{  
 return millisSinceEpoch?.*let***{** Date(**it**)  
 **}** }  
 @TypeConverter  
 fun toUUID(uuid: String?): UUID?{  
 return UUID.fromString(uuid)  
 }  
 @TypeConverter  
 fun fromUUID(uuid: UUID?):String?{  
 return uuid?.toString()  
 }  
}

*ThirdActivity*

package com.example.watchappvariant7  
  
import android.os.Bundle  
import android.util.Log  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.*lifecycleScope*import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.android.volley.Request  
import com.android.volley.Response  
import com.android.volley.toolbox.JsonObjectRequest  
import com.android.volley.toolbox.Volley  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
import org.json.JSONArray  
import org.json.JSONObject  
  
class ThirdActivity : AppCompatActivity() {  
  
 private lateinit var artistNameTextView: TextView  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var adapter: AlbumAdapter  
 private lateinit var database: AppDatabase  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_third*)  
  
 artistNameTextView = findViewById(R.id.*artistNameTextView*)  
 recyclerView = findViewById(R.id.*recyclerView*)  
 database = AppDatabase.getDatabase(this)  
  
 val layoutManager = LinearLayoutManager(this)  
 recyclerView.*layoutManager* = layoutManager  
  
 // Получаем название исполнителя из предыдущего экрана  
 val artistName = *intent*.getStringExtra("artistName")  
  
 // Отображаем название исполнителя  
 artistNameTextView.*text* = "Альбомы исполнителя: $artistName"  
  
 // Выполняем запрос к Discogs API для поиска альбомов по имени исполнителя  
 if (artistName != null) {  
 searchAlbums(artistName)  
 }  
 }  
  
 private fun searchAlbums(artistName: String) {  
 val token = "FJnLVZOhTMlTbiTXzbVQqzbEXRIClbKKnDfmyxOr"  
 val baseUrl = "https://api.discogs.com/database/search"  
 val apiKeyParam = "token=$token"  
 val artistNameParam = "artist=$artistName"  
 val typeParam = "type=release"  
  
 val url = "$baseUrl?$apiKeyParam&$artistNameParam&$typeParam"  
  
 val request = JsonObjectRequest(Request.Method.*GET*, url, null,  
 **{** response **->** // Обработка успешного ответа  
 val albums = parseAlbums(response)  
  
 // Инициализируем адаптер и устанавливаем его для RecyclerView  
 adapter = AlbumAdapter(albums)  
 recyclerView.*adapter* = adapter  
  
 // Запускаем корутину для вставки альбомов в базу данных  
 *lifecycleScope*.*launch* **{** insertAlbumsIntoDatabase(albums)  
 **}  
 }**,  
 **{** error **->** // Обработка ошибок  
 error.printStackTrace()  
 **}**)  
  
 // Добавляем запрос в очередь Volley  
 Volley.newRequestQueue(this).add(request)  
 }  
  
 private suspend fun insertAlbumsIntoDatabase(albums: List<Album>) {  
 // Используем withContext для выполнения операции в фоновом потоке  
 withContext(Dispatchers.IO) **{** // Запускаем корутину для вставки каждого альбома в базу данных  
 for (album in albums) {  
 database.albumDao().insert(album)  
 }  
 **}** }  
  
  
 private fun parseAlbums(response: JSONObject): List<Album> {  
 val albumsList = *mutableListOf*<Album>()  
 val resultsArray: JSONArray = response.optJSONArray("results") ?: JSONArray()  
  
 for (i in 0 *until* resultsArray.length()) {  
 val resultObject: JSONObject = resultsArray.optJSONObject(i) ?: JSONObject()  
 val title = resultObject.optString("title", "")  
  
 val imageUrl = resultObject.optString("cover\_image")  
  
 // Устанавливаем заглушечный URL, если URL изображения не найден  
 val placeholderImageUrl = "https://developers.elementor.com/docs/assets/img/elementor-placeholder-image.png"  
 val finalImageUrl = imageUrl ?: placeholderImageUrl  
  
 albumsList.add(Album(title, finalImageUrl))  
  
 // Отладочная информация  
 Log.d("AlbumInfo", "Title: $title, Image URL: $finalImageUrl")  
 }  
  
 return albumsList  
 }  
  
}

*SearchHistoryActivity*

package com.example.watchappvariant7  
  
import android.app.Activity  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import androidx.lifecycle.*lifecycleScope*import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
  
class SearchHistoryActivity : AppCompatActivity() {  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var adapter: AlbumHistoryAdapter  
 private lateinit var database: AppDatabase  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_search\_history*)  
 recyclerView = findViewById(R.id.*historyRecyclerView*)  
 database = AppDatabase.getDatabase(this)  
  
 val layoutManager = LinearLayoutManager(this)  
 recyclerView.*layoutManager* = layoutManager  
 loadAlbumsFromDatabase()  
 }  
  
 private fun loadAlbumsFromDatabase() {  
 *lifecycleScope*.*launch* **{**   
 val albums = withContext(Dispatchers.IO) **{** database.albumDao().getAllAlbums()  
 **}**  
 adapter = AlbumHistoryAdapter(albums)  
 recyclerView.*adapter* = adapter  
 adapter.setOnAlbumHistoryItemClickListener(object :  
 AlbumHistoryAdapter.OnAlbumHistoryItemClickListener {  
 override fun onHistoryItemClick(album: Album) {  
   
 val intent = Intent(this@SearchHistoryActivity, EditAlbumActivity::class.*java*)  
 intent.putExtra("album", album)  
 startActivityForResult(intent, EDIT\_ALBUM\_REQUEST\_CODE)  
 }  
  
 override fun onHistoryItemDeleteClick(album: Album) {  
   
 *lifecycleScope*.*launch*(Dispatchers.IO) **{** database.albumDao().delete(album)  
  
  
 withContext(Dispatchers.Main) **{** val updatedAlbums = withContext(Dispatchers.IO) **{** database.albumDao().getAllAlbums()  
 **}** adapter.setAlbums(updatedAlbums)  
 adapter.notifyDataSetChanged()  
 **}  
 }** }  
 })  
 **}** }  
  
  
 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 super.onActivityResult(requestCode, resultCode, data)  
 if (requestCode == EDIT\_ALBUM\_REQUEST\_CODE && resultCode == Activity.*RESULT\_OK*) {  
 val updatedAlbum = data?.getParcelableExtra<Album>("updatedAlbum")  
 val oldAlbum = data?.getParcelableExtra<Album>("oldAlbum")  
 updatedAlbum?.*let* **{** *lifecycleScope*.*launch*(Dispatchers.IO) **{** // Update the album in the database  
 if (oldAlbum != null) {  
 database.albumDao().updateAlbum(oldAlbum.title,updatedAlbum.title,updatedAlbum.imageUrl)  
 }  
  
 loadAlbumsFromDatabase()  
 **}  
 }** }  
 }  
  
 companion object {  
 private const val EDIT\_ALBUM\_REQUEST\_CODE = 1  
 }  
}

*EditAlbumActivity*

package com.example.watchappvariant7  
  
import android.app.Activity  
import android.content.Intent  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import androidx.appcompat.app.AppCompatActivity  
import com.example.watchappvariant7.Album  
  
class EditAlbumActivity : AppCompatActivity() {  
 private lateinit var titleEditText:EditText  
 private lateinit var imageUrlEditText:EditText  
 private lateinit var saveButton: Button  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_edit\_album*)  
 titleEditText = findViewById(R.id.*titleEditText*)  
 imageUrlEditText=findViewById(R.id.*imageUrlEditText*)  
 saveButton = findViewById(R.id.*saveButton*)  
  
 val album: Album? = *intent*.getParcelableExtra("album")  
  
 // Check if album is not null  
 if (album != null) {  
 // Display album details in your UI components (EditText, ImageView, etc.)  
 titleEditText.setText(album.title)  
 imageUrlEditText.setText(album.imageUrl)  
  
 // Implement your save/update logic here when the user clicks the save button  
 saveButton.setOnClickListener **{** // Update the album with the new values  
 val updatedAlbum = Album(  
 titleEditText.*text*.toString(),  
 imageUrlEditText.*text*.toString()  
 )  
  
 // Pass the updated album back to the calling activity  
 val resultIntent = Intent()  
 resultIntent.putExtra("updatedAlbum", updatedAlbum)  
 resultIntent.putExtra("oldAlbum",album)  
 setResult(Activity.*RESULT\_OK*, resultIntent)  
 finish()  
 **}** } else {  
 // Handle the case where the album is null (optional)  
 }  
 }  
}

*Album*

package com.example.watchappvariant7  
  
import android.os.Parcel  
import android.os.Parcelable  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
  
@Entity(tableName = "albums")  
data class Album(  
 @PrimaryKey  
 val title: String,  
 val imageUrl: String  
): Parcelable {  
 constructor(parcel: Parcel) : this(  
 parcel.readString() ?: "",  
 parcel.readString() ?: ""  
 )  
 override fun writeToParcel(parcel: Parcel, flags: Int) {  
 parcel.writeString(title)  
 parcel.writeString(imageUrl)  
 }  
  
 override fun describeContents(): Int {  
 return 0  
 }  
  
 companion object CREATOR : Parcelable.Creator<Album> {  
 override fun createFromParcel(parcel: Parcel): Album {  
 return Album(parcel)  
 }  
  
 override fun newArray(size: Int): Array<Album?> {  
 return *arrayOfNulls*(size)  
 }  
 }  
}

*AlbumDao*

package com.example.watchappvariant7  
  
import androidx.room.Dao  
import androidx.room.Delete  
import androidx.room.Insert  
import androidx.room.OnConflictStrategy  
import androidx.room.Query  
import androidx.room.Update  
  
@Dao  
interface AlbumDao {  
 @Insert(onConflict = OnConflictStrategy.IGNORE)  
 suspend fun insert(album: Album): Long  
  
 @Query("SELECT \* FROM albums")  
 suspend fun getAllAlbums(): List<Album>  
  
 @Query("UPDATE albums SET title = :title, imageUrl = :imageUrl WHERE title = :oldTitle")  
 suspend fun updateAlbum(oldTitle: String, title: String, imageUrl: String)  
 @Delete  
 suspend fun delete(album:Album)  
}

*AlbumHistoryAdapter*

package com.example.watchappvariant7  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Button  
import android.widget.ImageButton  
import android.widget.ImageView  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
import com.example.watchappvariant7.R  
import com.squareup.picasso.Picasso  
import kotlinx.coroutines.Dispatchers  
import androidx.lifecycle.*lifecycleScope*class AlbumHistoryAdapter(private var albums: List<Album>) :  
 RecyclerView.Adapter<AlbumHistoryAdapter.ViewHolder>() {  
  
 private lateinit var onAlbumHistoryItemClickListener: OnAlbumHistoryItemClickListener  
  
 interface OnAlbumHistoryItemClickListener {  
 fun onHistoryItemClick(album: Album)  
 fun onHistoryItemDeleteClick(album: Album)  
 }  
  
 fun setOnAlbumHistoryItemClickListener(listener: OnAlbumHistoryItemClickListener) {  
 onAlbumHistoryItemClickListener = listener  
 }  
  
 fun setAlbums(albums: List<Album>) {  
 this.albums = albums  
 notifyDataSetChanged()  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(parent.*context*)  
 .inflate(R.layout.*item\_history\_album*, parent, false)  
 return ViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 val album = albums[position]  
 holder.bind(album)  
 }  
  
 override fun getItemCount(): Int {  
 return albums.size  
 }  
  
 inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 private val titleHistoryTextView: TextView = itemView.findViewById(R.id.*albumHistoryTitleTextView*)  
 private val imageHistoryImageView: ImageView = itemView.findViewById(R.id.*albumHistoryImageView*)  
 private val deleteButton: ImageButton = itemView.findViewById(R.id.*deleteButton*)  
  
 init {  
 deleteButton.setOnClickListener **{** val position = *adapterPosition* if (position != RecyclerView.*NO\_POSITION*) {  
 val albumToDelete = albums[position]  
 onAlbumHistoryItemClickListener.onHistoryItemDeleteClick(albumToDelete)  
 }  
 **}** itemView.setOnClickListener **{** val position = *adapterPosition* if (position != RecyclerView.*NO\_POSITION*) {  
 val clickedAlbum = albums[position]  
 onAlbumHistoryItemClickListener.onHistoryItemClick(clickedAlbum)  
 }  
 **}** }  
  
 fun bind(album: Album) {  
 titleHistoryTextView.*text* = album.title  
 // Используйте библиотеку Picasso (или другую подобную) для загрузки изображений  
 Picasso.get().load(album.imageUrl).into(imageHistoryImageView)  
 }  
 }  
}

*AppDatabase*

package com.example.watchappvariant7  
  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
import android.content.Context  
  
@Database(entities = [Album::class], version = 1, exportSchema = false)  
abstract class AppDatabase : RoomDatabase() {  
 abstract fun albumDao(): AlbumDao  
  
 companion object {  
 @Volatile  
 private var INSTANCE: AppDatabase? = null  
  
 fun getDatabase(context: Context): AppDatabase {  
 return INSTANCE ?: *synchronized*(this) **{** val instance = Room.databaseBuilder(  
 context.*applicationContext*,  
 AppDatabase::class.*java*,  
 "app\_database"  
 ).build()  
 INSTANCE = instance  
 instance  
 **}** }  
 }  
}

*activity\_main.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="#173337"  
 tools:context=".MainActivity">  
  
 <EditText  
 android:id="@+id/loginEditText"  
 android:layout\_width="120dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="16sp"  
 android:hint="Login"  
 android:layout\_marginBottom="16dp"  
 android:layout\_gravity="center"/>  
  
 <EditText  
 android:id="@+id/passwordEditText"  
 android:layout\_width="120dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="16sp"  
 android:hint="Password"  
 android:layout\_marginBottom="16dp"  
 android:layout\_gravity="center"/>  
  
 <Button  
 android:id="@+id/loginButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textColor="#231F1F"  
 android:background="@drawable/button\_design"  
 android:layout\_gravity="center"/>  
</LinearLayout>

*activity\_second.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="#173337"  
 tools:context=".SecondActivity">  
  
 <EditText  
 android:id="@+id/artistEditText"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:textSize="16dp"  
 android:hint="Введите артиста"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="16dp"/>  
  
 <Button  
 android:id="@+id/nextButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:text="Next"  
 android:textSize="10sp"  
 android:textColor="#231F1F"  
 android:layout\_marginBottom="5dp"  
 android:background="@drawable/button\_design"  
 android:layout\_gravity="center"/>  
 <Button  
 android:id="@+id/historyButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:textSize="10sp"  
 android:layout\_gravity="center"  
 android:textColor="#231F1F"  
 android:background="@drawable/button\_design"  
 android:text="Посмотреть историю поиска"/>  
</LinearLayout>

*activity\_third.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="#173337"  
 tools:context=".ThirdActivity">  
  
 <TextView  
 android:id="@+id/artistNameTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="text"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="16dp"/>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:layout\_marginBottom="16dp" />  
  
</LinearLayout>

*activity\_search\_history.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="#173337"  
 tools:context=".SearchHistoryActivity">  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/historyRecyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:layout\_marginBottom="16dp" />  
  
</LinearLayout>

*activity\_edit\_albom.xml*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp"  
 tools:context=".EditAlbumActivity"  
 android:orientation="vertical"  
 android:background="#173337">  
  
 <EditText  
 android:id="@+id/titleEditText"  
 android:layout\_marginTop="10dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:hint="Album Title"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="16dp"/>  
  
 <EditText  
 android:id="@+id/imageUrlEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:hint="Image URL"  
 android:layout\_gravity="center"  
 android:layout\_below="@id/titleEditText"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="16dp"/>  
  
 <Button  
 android:id="@+id/saveButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Save"  
 android:textColor="#231F1F"  
 android:background="@drawable/button\_design"  
 android:layout\_gravity="center" />  
  
</LinearLayout>

*item\_album*

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@color/\_173337"  
 android:padding="16dp">  
  
 <ImageView  
 android:id="@+id/albumImageView"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:layout\_gravity="center"  
 android:src="@drawable/placeholder\_image"  
 android:scaleType="centerCrop" />  
  
  
 <TextView  
 android:id="@+id/albumTitleTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:layout\_gravity="center"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/artistNameTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:textSize="16sp" />  
  
  
</LinearLayout>

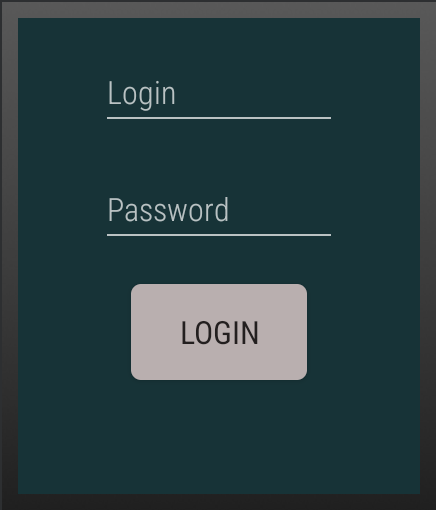
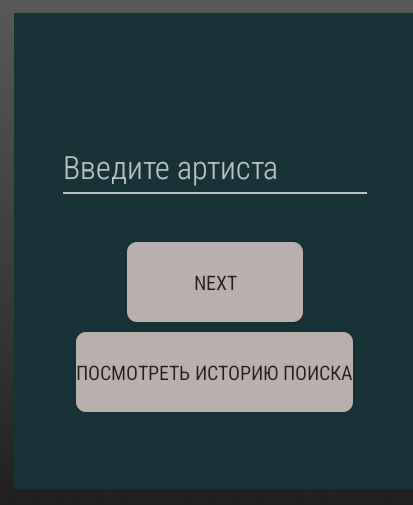
*item\_album\_history.xml*

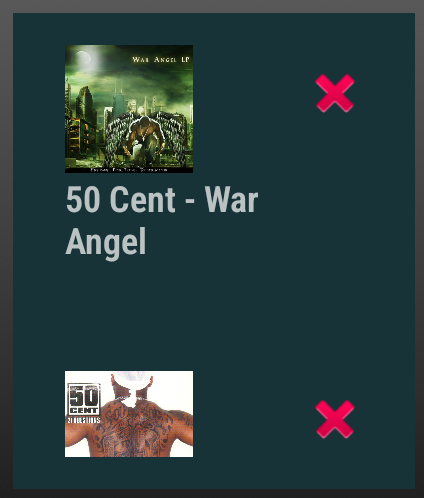
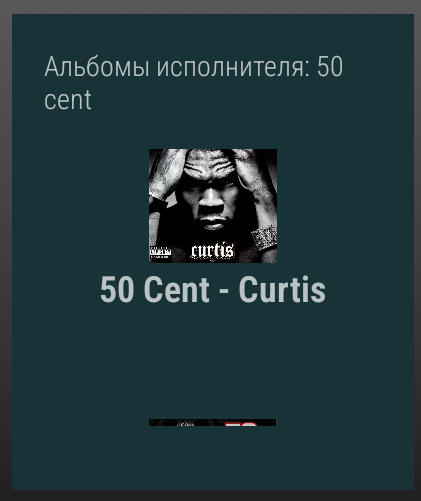
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:background="#173337"  
 android:padding="16dp">  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:orientation="vertical"  
 android:layout\_marginStart="10dp">  
  
 <ImageView  
 android:id="@+id/albumHistoryImageView"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 android:src="@drawable/placeholder\_image"  
 android:scaleType="centerCrop" />  
  
  
 <TextView  
 android:id="@+id/albumHistoryTitleTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/artistHistoryNameTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="16sp" />  
  
 </LinearLayout>  
  
 <ImageButton  
 android:id="@+id/deleteButton"  
 android:layout\_width="48dp"  
 android:layout\_height="48dp"  
 android:background="?android:selectableItemBackground"  
 android:src="@android:drawable/ic\_delete"  
 android:contentDescription="Delete Album" />  
  
</LinearLayout>

## **Используемые библиотеки**

*implementation*("androidx.core:core-ktx:1.12.0")  
*implementation*("com.google.android.gms:play-services-wearable:18.1.0")  
*implementation*("androidx.percentlayout:percentlayout:1.0.0")  
*implementation*("androidx.legacy:legacy-support-v4:1.0.0")  
*implementation*("androidx.recyclerview:recyclerview:1.3.2")  
*implementation*(platform("androidx.compose:compose-bom:2023.03.00"))  
*implementation*("androidx.compose.ui:ui")  
*implementation*("androidx.compose.ui:ui-tooling-preview")  
*implementation*("androidx.wear.compose:compose-material:1.2.1")  
*implementation*("androidx.wear.compose:compose-foundation:1.2.1")  
*implementation*("androidx.lifecycle:lifecycle-runtime-ktx:2.6.2")  
*implementation*("androidx.activity:activity-compose:1.8.1")  
*implementation*("androidx.appcompat:appcompat:1.6.1")  
*implementation*("com.google.android.material:material:1.10.0")  
*implementation*("androidx.constraintlayout:constraintlayout:2.1.4")  
*androidTestImplementation*(platform("androidx.compose:compose-bom:2023.03.00"))  
*androidTestImplementation*("androidx.compose.ui:ui-test-junit4")  
*debugImplementation*("androidx.compose.ui:ui-tooling")  
*debugImplementation*("androidx.compose.ui:ui-test-manifest")  
//volley  
*implementation*("com.android.volley:volley:1.2.1")  
//picasso  
*implementation*("com.squareup.picasso:picasso:2.71828")  
//room  
*implementation*("androidx.room:room-runtime:2.6.1")  
*kapt*("androidx.room:room-compiler:2.6.1")  
*implementation*("androidx.room:room-ktx:2.6.1")  
*implementation*("androidx.lifecycle:lifecycle-viewmodel:2.6.1")  
*implementation*("androidx.lifecycle:lifecycle-livedata:2.6.1")

## **Экраны**



## **Используемые инструменты**

kotlin, xml, android studio

## **Вывод:**

Я создал функциональное приложение для часов, используя room database и volley