

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.01

Выполнила:Оберюхтина Елена

Группа: ПР-31

Преподаватель: Мирошниченко Г.В.

2023

**Задание:**Разработать мобильное приложение – умные часы, информация в приложении

должна сохраняться в SharedPreferences, в формате JSON. Приложение должно

содержать 4 экрана:

- приветственный экран с авторизацией пользователя, проверкой

ввода на пустоту и сохранением информации о пользователе через механизм

SharedPreferences.

- содержать форму ввода данных

- экран, содержащий меню со значками

- список из меню с использованием библиотеки ROOM,

возможностью добавления, удаления введенных данных, с сохранением в базе

данных (ROOM или SQL).

- стилизация приложения в соответствии с вариантом

- изменить название, тему и иконку приложения (должны

соответствовать названию)

- предусмотреть портретную ориентацию

Приложение «Правильное питание» Spoonacular API ( https://spoonacular.com )

**Входные данные:**

editText:пользователь вводит ID рецепта

**Выходные данные:**

resultTextView:Вывод информации о названии, ингридиентах и времени приготовления рецепта

**Листинг программы:**

Dao:

package com.example.zd2\_2up  
  
  
import androidx.room.\*  
import androidx.room.Dao  
import kotlinx.coroutines.flow.Flow  
  
@Dao  
interface Dao {  
 //запись  
 @Insert  
 fun insertItem(item: Item)  
  
 //получение всей бд  
 @Query("SELECT \* FROM items")  
 fun getAllItem(): Flow<List<Item>>  
 //удаление  
 @Delete  
 suspend fun deleteItem(item: Item)  
 //обновление  
 @Update  
 suspend fun updateItem(item: Item)  
}

Item:

package com.example.zd2\_2up  
  
import androidx.room.ColumnInfo  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
  
  
@Entity(tableName = "items")  
  
//коллоны  
data class Item(  
  
 //ид  
 @PrimaryKey(autoGenerate = true)  
 var id: Int? = null,  
 @ColumnInfo(name = "title")  
 var title : String,  
 @ColumnInfo(name = "ingredients")  
 var ingredients:String,  
 @ColumnInfo(name = "readyInMinutes")  
 var readyInMinutes:String  
  
 )

ItemAdapter:

package com.example.zd2\_2up  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.BaseAdapter  
import android.widget.ImageButton  
import android.widget.TextView  
  
class ItemAdapter (  
 context: Context,  
 private val itemList: List<Item>,  
 private val onDeleteClick: (Item) -> Unit,  
 private val onEditClick: (Item) -> Unit  
) : BaseAdapter() {  
 private val inflater: LayoutInflater = LayoutInflater.from(context)  
  
 override fun getCount(): Int {  
 return itemList.size  
 }  
  
 override fun getItem(position: Int): Any {  
 return itemList[position]  
 }  
  
 override fun getItemId(position: Int): Long {  
 return position.toLong()  
 }  
  
 override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {  
 val view: View  
 val holder: ViewHolder  
  
 if (convertView == null) {  
 view = inflater.inflate(R.layout.*activity\_list\_of\_recipes*, parent, false)  
 holder = ViewHolder(  
 view.findViewById(R.id.*titleTextView*),  
 view.findViewById(R.id.*ingridentText*),  
 view.findViewById(R.id.*readyTextView*),  
 view.findViewById(R.id.*editButton*),  
 view.findViewById(R.id.*deleteButton*)  
 )  
 view.*tag* = holder  
 } else {  
 view = convertView  
 holder = view.*tag* as ViewHolder  
 }  
  
 val item = itemList[position]  
 holder.titleTextView.*text* = item.title  
 holder.descriptionTextView.*text* = item.readyInMinutes  
 holder.ingridentTextView.*text*=item.ingredients  
  
  
 // Обработка нажатий на кнопку "Редактировать"  
 holder.editButton.setOnClickListener **{** onEditClick(item)  
 **}** // Обработка нажатий на кнопку "Удалить"  
 holder.deleteButton.setOnClickListener **{** onDeleteClick(item)  
 **}** return view  
 }  
  
 private class ViewHolder(  
 val titleTextView: TextView,  
 val descriptionTextView: TextView,  
 val ingridentTextView:TextView,  
 val editButton: ImageButton,  
 val deleteButton: ImageButton,  
  
 )  
}

ListOfRecipes:

package com.example.zd2\_2up  
  
import android.app.Activity  
import android.os.Bundle  
import com.example.zd2\_2up.databinding.ActivityListOfRecipesBinding  
  
class ListOfRecipes : Activity() {  
  
 private lateinit var binding: ActivityListOfRecipesBinding  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityListOfRecipesBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 }  
}

MainActivity:

package com.example.zd2\_2up  
  
import android.app.Activity  
import android.content.Intent  
import android.content.SharedPreferences  
import android.os.Bundle  
import android.view.View  
import android.widget.EditText  
import android.widget.Toast  
import com.example.zd2\_2up.databinding.ActivityMainBinding  
  
class MainActivity : Activity() {  
  
 private lateinit var binding: ActivityMainBinding  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityMainBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
 val sharedPreferences = getSharedPreferences("LoginAndPassword", *MODE\_PRIVATE*)  
 val editor: SharedPreferences.Editor = sharedPreferences.edit()  
  
 editor.putString("username", "ects")  
 editor.putString("password", "ects2023")  
 editor.apply()  
 }  
 fun perehod(view: View) {  
  
 val usernameEditText: EditText = findViewById(R.id.*email*)  
 val passwordEditText: EditText = findViewById(R.id.*pass*)  
  
 val log: String = usernameEditText.*text*.toString()  
 val pas: String = passwordEditText.*text*.toString()  
  
  
 if (usernameEditText.*text*.toString().*isEmpty*() || passwordEditText.*text*.toString()  
 .*isEmpty*()  
 ) {  
  
 Toast.makeText(this@MainActivity, "Введите логин и пароль", Toast.*LENGTH\_SHORT*).show()  
 } else {  
  
 val sharedPreferences = getSharedPreferences("LoginAndPassword", *MODE\_PRIVATE*)  
  
 val savedUsername = sharedPreferences.getString("username", "")  
 val savedPassword = sharedPreferences.getString("password", "")  
  
  
 if (log == savedUsername && pas == savedPassword) {  
 val intent = Intent(this@MainActivity, RecipesActivity::class.*java*)  
 startActivity(intent)  
 }  
 else  
 {  
 Toast.makeText(this@MainActivity, "Неверный логин или пароль", Toast.*LENGTH\_SHORT*).show()  
  
 }  
  
 }  
 }  
}

MainDb:

package com.example.zd2\_2up  
  
import android.content.Context  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
  
@Database(entities = [Item::class], version = 1)  
abstract class MainDb : RoomDatabase() {  
  
 abstract fun getDao(): Dao  
  
 companion object{  
 fun getDb(context: Context): MainDb{  
 return Room.databaseBuilder(  
 context.*applicationContext*, MainDb::class.*java*, "test.db"  
 ).build()  
 }  
 }  
}

RecipesActivity:

package com.example.zd2\_2up  
  
import android.app.Activity  
import android.content.Intent  
import android.os.Bundle  
import android.util.Log  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import android.widget.Toast  
import com.android.volley.Request  
import com.android.volley.RequestQueue  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import com.example.zd2\_2up.databinding.ActivityRecipesBinding  
  
import okhttp3.Response  
import okhttp3.\*  
import org.json.JSONException  
import org.json.JSONObject  
  
class RecipesActivity : Activity() {  
  
 private lateinit var binding: ActivityRecipesBinding  
 private lateinit var editText: EditText  
 private lateinit var button: Button  
 private lateinit var perhod:Button  
 private lateinit var resultTextView: TextView  
 //private val BASE\_URL = "https://api.spoonacular.com/recipes/"  
 private val API\_KEY = "12939f360bff4fef8800f479ecd5b7aa"  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityRecipesBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
 editText = findViewById(R.id.*editText*)  
 button=findViewById(R.id.*button*)  
 resultTextView= findViewById(R.id.*resultTextView*)  
 perhod=findViewById(R.id.*butperehod*)  
 button.setOnClickListener **{** val recipeId = editText.*text*.toString()  
 getResualt(recipeId)  
 **}** perhod.setOnClickListener **{** val intent = Intent(this, ScreenStatistics::class.*java*)  
 startActivity(intent) **}** }  
 /\*private fun fetchRecipeInfo(recipeId: String) {  
 val url = "$BASE\_URL$recipeId/information?apiKey=$API\_KEY"  
 FetchRecipeTask(resultTextView).execute(url)  
 }\*/  
 private fun getResualt(recipeId:String){  
 if (recipeId.*isNotEmpty*()) {  
 val url = "https://api.spoonacular.com/recipes/$recipeId/information?apiKey=$API\_KEY"  
  
 val queue = Volley.newRequestQueue(this)  
 val stringRequest = StringRequest(  
 Request.Method.*GET*, url,  
 **{** response **->** try {  
 val jsonObject = JSONObject(response)  
 val title = jsonObject.getString("title")  
 val readyInMinutes = jsonObject.getInt("readyInMinutes")  
 val extendedIngredients = jsonObject.getJSONArray("extendedIngredients")  
 val ingredients = StringBuilder()  
 for (i in 0 *until* extendedIngredients.length()) {  
 val ingredientObject = extendedIngredients.getJSONObject(i)  
 val ingredient = ingredientObject.getString("original")  
 ingredients.append(ingredient).append("\n")  
 }  
 Log.d("MyLog", "Title: $title")  
 Log.d("MyLog", "Ingredients: $ingredients")  
 Log.d("MyLog", "Ready in minutes: $readyInMinutes")  
  
 val resultText = "Recipe: $title\n\nIngredients:\n$ingredients\n\nReady in $readyInMinutes minutes"  
 resultTextView.*text* = resultText  
 //запись в бд  
 val db = MainDb.getDb(this)  
 val item = Item(  
 null,  
 title,  
 readyInMinutes.toString(),  
 ingredients.toString(),  
 )  
 Thread **{** db.getDao().insertItem(item)  
 **}**.start()  
  
 } catch (e: JSONException) {  
 Log.e("MyLog", "JSON Parsing error: ${e.message}")  
 }  
 **}**,  
 **{** error **->** Log.d("MyLog", "Volley error: ${error.message}")  
 **}**)  
  
 queue.add(stringRequest)  
 }  
 else{ Toast.makeText(this@RecipesActivity, "Заполните поле", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
  
 }

ScreenStatistics:

package com.example.zd2\_2up  
  
import android.app.Activity  
import android.app.AlertDialog  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.LinearLayout  
import android.widget.ListView  
import androidx.appcompat.app.AppCompatActivity  
import com.example.zd2\_2up.databinding.ActivityScreenStatisticsBinding  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import androidx.lifecycle.asLiveData  
import kotlinx.coroutines.withContext  
  
class ScreenStatistics : AppCompatActivity() {  
  
 private lateinit var binding: ActivityScreenStatisticsBinding  
 private val dataList = *mutableListOf*<Item>()  
 private lateinit var buttonclear: Button  
 private lateinit var buttonadd: Button  
 override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState)  
  
 binding = ActivityScreenStatisticsBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
 printInfo()  
 buttonclear=findViewById(R.id.*butclear*)  
 buttonadd=findViewById(R.id.*butadd*)  
 buttonclear.setOnClickListener **{** val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** // Очистить все таблицы в базе данных  
 db.clearAllTables()  
 withContext(Dispatchers.Main) **{  
 }  
 }** printInfo() **}** buttonadd.setOnClickListener **{** val builder = AlertDialog.Builder(this)  
 builder.setTitle("Введите данные")  
  
 // Создайте контейнер LinearLayout для размещения двух EditText  
 val layout = LinearLayout(this)  
 layout.*orientation* = LinearLayout.*VERTICAL* // Создайте два EditText для ввода данных  
 val EditTitle = EditText(this)  
 val EditIngredient = EditText(this)  
 val EditReady = EditText(this)  
  
 // Установите хинты для EditText  
 EditTitle.*hint* = "Введите название рецепта"  
 EditIngredient.*hint* = "Введите ингридиенты"  
 EditReady.*hint* = "Введите время приготовления"  
  
 // Добавьте EditText к контейнеру  
 layout.addView(EditTitle)  
 layout.addView(EditIngredient)  
 layout.addView(EditReady)  
  
 builder.setView(layout)  
  
 // Установите кнопку "OK" для сохранения данных  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** val title = EditTitle.*text*.toString()  
 val ingredient = EditIngredient.*text*.toString()  
 val ready=EditReady.*text*.toString()  
 // Здесь можно обработать введенные данные (value1 и value2)  
 //запись в бд  
 if(title.length == 0 || ingredient.length == 0||ready.length==0){  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Не сохранено")  
 .setMessage("Необходимо заполнить все поля")  
 .setPositiveButton("ОК") **{** dialog, id **->** dialog.cancel()  
 **}** builder.create()  
 }  
 else{  
 //сама запись  
 val db = MainDb.getDb(this)  
 val item = Item(null, title,ingredient,ready)  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
 dialog.dismiss()  
 printInfo()  
 }  
  
 **}** // Установите кнопку "Отмена" для закрытия диалога  
 builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 **}** }  
 private fun printInfo() {  
 val textList = findViewById<ListView>(R.id.*listView*)  
 dataList.clear()  
 val db = MainDb.getDb(this)  
 db.getDao().getAllItem().*asLiveData*().observe(this) **{** list **->** val updatedDataList = *mutableListOf*<Item>()  
 list.*forEach* **{** item **->** val newItem = Item(id = item.id, title = item.title, readyInMinutes = item.readyInMinutes, ingredients = item.ingredients)  
 updatedDataList.add(newItem)  
 **}** // После завершения цикла, обновите dataList  
 dataList.clear()  
 dataList.addAll(updatedDataList)  
 val itemAdapter = ItemAdapter(this, dataList,  
 onDeleteClick = **{** itemToDelete **->** // Здесь выполняйте удаление элемента из базы данных и обновление списка  
 val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 db.getDao().deleteItem(itemToDelete)  
  
 printInfo()  
 } catch (e: Exception) {  
 }  
 **}** printInfo()  
 printInfo()  
 **}**,  
 onEditClick = **{** position **->** val db = MainDb.getDb(this)  
 var bdNewText: Array<String> = *arrayOf*("id", "title", "readyInMinutes", "ingredients")  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 bdNewText= *arrayOf*(position.id.*toString*(), position.title, position.ingredients, position.readyInMinutes)  
 } catch (e: Exception) {  
 }  
 **}** //изменения через AlertDiolog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Измените данные")  
 // Создайте контейнер LinearLayout для размещения EditText  
 val layout = LinearLayout(this)  
 layout.*orientation* = LinearLayout.*VERTICAL* val EditTitle = EditText(this)  
 val EditIngredients = EditText(this)  
 val EditReadyInMinutes = EditText(this)  
 EditTitle.*hint* = "Введите название рецепта"  
 EditReadyInMinutes.*hint* = "Введите время приготовления"  
 EditIngredients.*hint* = "Введите ингридиенты"  
 EditTitle.setText(bdNewText[1])  
 EditReadyInMinutes.setText(bdNewText[3])  
 EditIngredients.setText(bdNewText[2])  
 layout.addView(EditTitle)  
 layout.addView(EditReadyInMinutes)  
 layout.addView(EditIngredients)  
 builder.setView(layout)  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** var title = EditTitle.*text*.toString()  
 var readyInMinutes = EditReadyInMinutes.*text*.toString()  
 var ingredients = EditIngredients.*text*.toString()  
  
 //запись в бд  
 //проверка  
 if(title.*isEmpty*()){  
 title= bdNewText[1];  
 }  
 if(readyInMinutes.*isEmpty*()){  
 readyInMinutes = bdNewText[3];  
 }  
 if(ingredients.*isEmpty*()){  
 ingredients = bdNewText[2];  
 }  
 val item = Item(bdNewText[0].*toInt*(), title, ingredients,readyInMinutes)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** db.getDao().updateItem(item)  
 **}** printInfo()  
 **}** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 printInfo()  
  
 **}**)  
 textList.*adapter* = itemAdapter  
 itemAdapter.notifyDataSetChanged()  
  
 **}** }  
}

activity\_list\_of\_recipes:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.wear.widget.BoxInsetLayout 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:padding="@dimen/box\_inset\_layout\_padding"  
 tools:context=".ListOfRecipes"  
 tools:deviceIds="wear">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/inner\_frame\_layout\_padding"  
 app:layout\_boxedEdges="all"  
 >  
  
 <LinearLayout  
 android:layout\_width="130dp"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/titleTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:paddingStart="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="4dp" />  
 <TextView  
 android:textColor="@color/black"  
 android:text="Ingredient:"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/ingridentText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:textColor="@color/black"  
 android:text="Ready in:"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/readyTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="12sp"  
 />  
  
 </LinearLayout>  
  
  
 </LinearLayout>  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" android:orientation="vertical">  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/editButton"  
 android:src="@drawable/edit"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
  
 >  
  
 </ImageButton>  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/deleteButton"  
 android:src="@drawable/delet"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
  
 >  
  
 </ImageButton>  
</LinearLayout>  
  
  
 </LinearLayout>  
</androidx.wear.widget.BoxInsetLayout>

activity\_main:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.wear.widget.BoxInsetLayout 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:padding="@dimen/box\_inset\_layout\_padding"  
 tools:context=".MainActivity"  
 tools:deviceIds="wear"  
 android:background="@color/font">  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/inner\_frame\_layout\_padding"  
 app:layout\_boxedEdges="all">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/Reprice"  
 android:textSize="15dp"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginTop="10dp"  
 android:textStyle="bold"  
 android:textColor="@color/black"/>  
 <EditText  
 android:textColor="@color/black"  
 android:inputType="textEmailAddress"  
 android:id="@+id/email"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"  
 android:hint="@string/login"  
 android:textColorHint="@color/black"/>  
  
 <EditText  
 android:textColor="@color/black"  
 android:inputType="textPassword"  
 android:id="@+id/pass"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"  
 android:hint="@string/pass"  
 android:textColorHint="@color/black"/>  
  
 <Button  
 android:onClick="perehod"  
 android:id="@+id/singin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="140dp"  
 android:text="@string/entrance"  
 android:textColor="@color/black"  
 android:textSize="12dp"  
 android:textStyle="bold"  
 android:textAllCaps="false"  
 android:background="@color/but"/>  
  
 </FrameLayout>  
</androidx.wear.widget.BoxInsetLayout>

activity\_recipes:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.wear.widget.BoxInsetLayout 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:padding="@dimen/box\_inset\_layout\_padding"  
 tools:context=".RecipesActivity"  
 tools:deviceIds="wear"  
 android:background="@color/font">  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/inner\_frame\_layout\_padding"  
 app:layout\_boxedEdges="all">  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <EditText  
 android:id="@+id/editText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Введи число и получи рецепт"  
 android:textSize="10dp"  
 android:inputType="number"  
 android:textColor="@color/black"  
 android:textColorHint="@color/black"  
 />  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="35dp"  
 android:text="@string/Reprice"  
 android:textSize="12dp"  
 android:textColor="@color/black"  
 android:background="@color/but"/>  
 <Button  
 android:id="@+id/butperehod"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="35dp"  
 android:text="@string/list"  
 android:textSize="12dp"  
 android:textColor="@color/black"  
 android:background="@color/but"/>  
  
</LinearLayout>  
  
<ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_boxedEdges="all">  
 <TextView  
 android:id="@+id/resultTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="10sp"  
 android:textColor="@color/black"/>  
</ScrollView>  
  
</LinearLayout>  
  
  
 </FrameLayout>  
</androidx.wear.widget.BoxInsetLayout>

activity\_screen\_statistics:

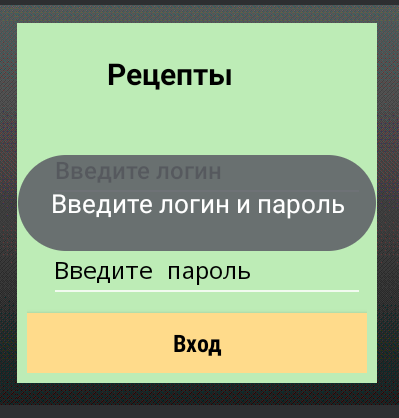
<?xml version="1.0" encoding="utf-8"?>  
<androidx.wear.widget.BoxInsetLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/box\_inset\_layout\_padding"  
 tools:context=".ScreenStatistics"  
 tools:deviceIds="wear"  
 android:background="@color/font">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="@dimen/inner\_frame\_layout\_padding"  
 app:layout\_boxedEdges="all"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/histori"  
 android:textColor="@color/black"/>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/butclear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="35dp"  
 android:text="@string/clear"  
 android:textSize="12dp"  
 android:textColor="@color/black"  
 android:background="@color/but"/>  
 <Button  
 android:id="@+id/butadd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="35dp"  
 android:text="@string/add"  
 android:textSize="12dp"  
 android:background="@color/but"  
 android:textColor="@color/black"/>  
  
 </LinearLayout>  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1" />  
 </LinearLayout>  
</androidx.wear.widget.BoxInsetLayout>

Bild.grable:

dependencies **{** implementation 'androidx.core:core-ktx:1.7.0'  
 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 'androidx.wear:wear:1.3.0'  
 implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
 implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
 implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.5.2'  
 implementation 'com.google.code.gson:gson:2.9.0'  
 implementation 'com.android.volley:volley:1.2.1'  
 implementation 'androidx.room:room-runtime:2.4.0'  
 kapt 'androidx.room:room-compiler:2.4.3'  
 implementation 'androidx.room:room-ktx:2.4.3'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
 implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'  
 implementation 'androidx.appcompat:appcompat:1.3.0'  
**}**

**Проверки:**

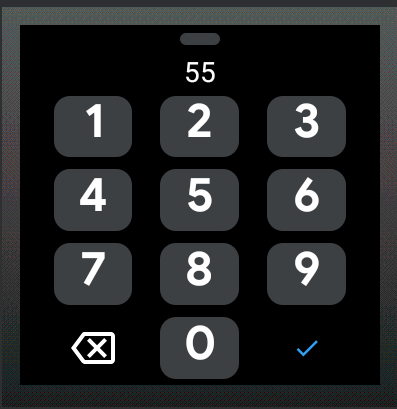
Проверка на пустоту:



Проверка на правильность логина и пароля:

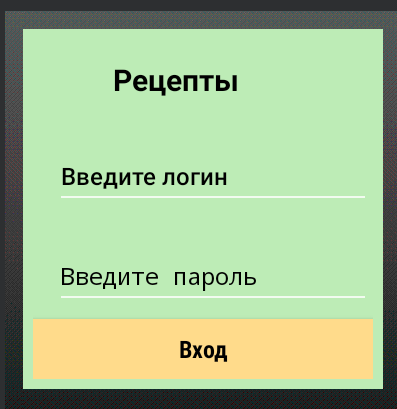


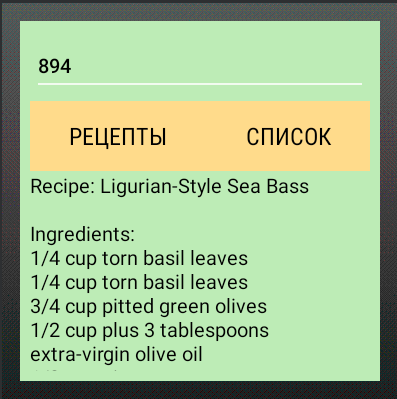
Пользователь может водить только числа:

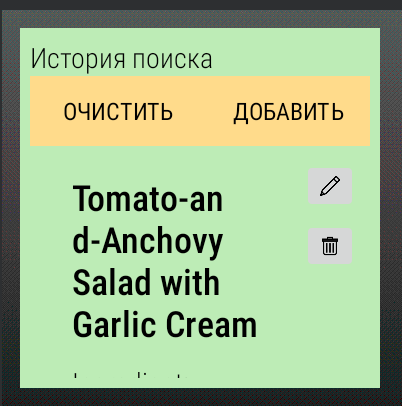


Если будут пустые поля при редактировании или добавлении и нформации то информация не запишеться

**Экраны:**







**Вывод:**Закрепила знания с базой данных Room