

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

Задание 2-2. Приложение - умные часы

Выполнил: Абдулазиомов Руслан

Группа: ПР-31

Преподаватель: Мирошниченко Г.В.

2023

## **Описание задачи**

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 «Трекер шагов (для бега и фитнеса)»

**Листинг**

*ChatListScreen*

package com.example.smartwatch  
  
import android.app.Activity  
import android.os.Bundle  
import android.view.View  
import android.widget.ListView  
import androidx.lifecycle.asLiveData  
import com.example.smartwatch.databinding.ActivityChatListScreenBinding  
import android.app.AlertDialog  
import android.app.Application  
import android.content.Context  
import android.content.DialogInterface  
import android.util.Log  
  
import android.widget.\*  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.Lifecycle  
import androidx.lifecycle.LifecycleEventObserver  
import androidx.lifecycle.LifecycleOwner  
import androidx.lifecycle.MutableLiveData  
import androidx.lifecycle.Observer  
import androidx.lifecycle.asLiveData  
import kotlinx.coroutines.\*  
import kotlinx.coroutines.flow.collect  
import android.text.InputType  
  
  
  
  
class ChatListScreen : AppCompatActivity() {  
  
 private lateinit var binding: ActivityChatListScreenBinding  
 var users = ArrayList<String>()  
 private val dataList = mutableListOf<Item>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityChatListScreenBinding.inflate(layoutInflater)  
 setContentView(binding.root)  
  
 printInfo()  
  
  
 }  
 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 text = "id ${item.id}, type ${item.type} number ${item.number} fact ${item.fact} \n"  
 val newItem = Item(id = item.id, type = item.type, number = item.number, fact = item.fact)  
 updatedDataList.add(newItem)  
 Log.d("MyLog", "Coordinates: $text")  
 **}** dataList.clear()  
 dataList.addAll(updatedDataList)  
  
 val itemAdapter = ItemAdapter(this, dataList,  
 onDeleteClick = **{** itemToDelete **->** val db = MainDb.getDb(this)  
 CoroutineScope(Dispatchers.IO).launch **{** try {  
 db.getDao().deleteItem(itemToDelete)  
 // После удаления, обновите список данных  
 withContext(Dispatchers.Main) **{** printInfo()  
 **}** } catch (e: Exception) {  
 // Обработка ошибки удаления, если произошла  
 }  
 **}  
 }**,  
 onEditClick = **{** position **->** val db = MainDb.getDb(this)  
 var bdNewText: Array<String> = arrayOf("id", "type", "number", "fact")  
 CoroutineScope(Dispatchers.IO).launch **{** try {  
 bdNewText = arrayOf(position.id.toString(), position.type, position.number, position.fact)  
 } catch (e: Exception) {  
 // Обработка ошибки удаления, если произошла  
 }  
 **}** val builder = AlertDialog.Builder(this)  
 builder.setTitle("Измените данные")  
  
 val layout = LinearLayout(this)  
 layout.orientation = LinearLayout.VERTICAL  
  
 val EditKm = EditText(this)  
 val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
  
 EditKm.hint = "Введите расстояние"  
 EditNumber.hint = "Введите шаги"  
 EditFakt.hint = "Введите каллории"  
  
 EditKm.setText(bdNewText[1])  
 EditNumber.setText(bdNewText[2])  
 EditFakt.setText(bdNewText[3])  
  
 EditKm.inputType = InputType.TYPE\_NUMBER\_FLAG\_DECIMAL  
  
  
 layout.addView(EditKm)  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
  
 builder.setView(layout)  
  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** var number = EditNumber.text.toString()  
 var fakt = EditFakt.text.toString()  
  
 var km = EditKm.text.toString()  
  
 if (number.isEmpty() || !cheakNumber(number)) {  
 number = bdNewText[2];  
 }  
 if (fakt.isEmpty() || !cheakNumber(fakt)) {  
 fakt = bdNewText[3];  
 }  
 if (km.isEmpty() || !cheakNumber(km)) {  
 km = bdNewText[1];  
 }  
  
 val item = Item(bdNewText[0].toInt(), km, number, fakt)  
 CoroutineScope(Dispatchers.IO).launch **{** db.getDao().updateItem(item)  
 // После обновления, обновите список данных  
 withContext(Dispatchers.Main) **{** printInfo()  
 **}  
 }  
 }** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 **}**)  
  
 textList.adapter = itemAdapter  
 itemAdapter.notifyDataSetChanged()  
 **}** }  
  
  
 fun ButCleaAll(view: View) {  
 val db = MainDb.getDb(this)  
 CoroutineScope(Dispatchers.IO).launch **{** // Очистите все таблицы в базе данных  
 db.clearAllTables()  
  
 // После удаления данных, выполните операции обновления UI на главном потоке (если необходимо)  
 withContext(Dispatchers.Main) **{** // Обновление UI, если необходимо  
 **}  
 }** printInfo()  
 }  
  
  
  
fun AddNewButton(view: View) {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Введите данные")  
  
 // Создайте контейнер LinearLayout для размещения двух EditText  
 val layout = LinearLayout(this)  
 layout.orientation = LinearLayout.VERTICAL  
  
 val EditKm = EditText(this)  
 val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
  
 EditKm.hint = "Введите расстояние"  
 EditNumber.hint = "Введите шаги"  
 EditFakt.hint = "Введите каллории"  
  
 // Добавьте EditText к контейнеру  
 layout.addView(EditKm)  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
  
 builder.setView(layout)  
  
 // Установите кнопку "OK" для сохранения данных  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** val number = EditNumber.text.toString()  
 val fakt = EditFakt.text.toString()  
 var km = EditKm.text.toString()  
  
 // Здесь можно обработать введенные данные (value1 и value2)  
 //запись в бд  
 if(number.length == 0 || fakt.length == 0 || km.length == 0 || !cheakNumber(km)|| !cheakNumber(fakt)|| !cheakNumber(number)){  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Не сохранено")  
 .setMessage("Необходимо заполнить все поля и использовать только неотрицательные числа")  
 .setPositiveButton("ОК") **{** dialog, id **->** dialog.cancel()  
 **}** builder.create()  
 }  
 else{  
 //сама запись  
 val db = MainDb.getDb(this)  
 val item = Item(null, km, number, fakt)  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
 dialog.dismiss()  
 printInfo()  
 }  
  
 **}** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
  
 }  
  
 //проверка числа  
 fun cheakNumber(input: String): Boolean {  
 return try {  
 val number = input.toDouble()  
 number >= 0  
 } catch (e: NumberFormatException) {  
 false  
 }  
 }  
  
}

*Dao*

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

*HomeLocation*

package com.example.smartwatch  
  
import android.app.Activity  
import android.content.SharedPreferences  
import android.os.Bundle  
import android.util.Log  
import android.view.View  
import android.widget.EditText  
import android.widget.ImageView  
import com.android.volley.Request  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import com.example.smartwatch.databinding.ActivityHomeLocationBinding  
import com.example.smartwatch.databinding.ActivityMoviesScreenBinding  
import com.google.android.gms.maps.model.LatLng  
import com.google.maps.android.SphericalUtil  
import com.squareup.picasso.Picasso  
import org.json.JSONException  
import org.json.JSONObject  
  
class HomeLocation : Activity() {  
 private val apiKeySity = "c72b98a5-ea26-4adf-85de-db5c0cb8c5df"  
  
  
 private lateinit var img: ImageView  
 private lateinit var edittext: EditText  
 private lateinit var binding: ActivityHomeLocationBinding  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityHomeLocationBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
 img = findViewById(R.id.*imgKarta*)  
 edittext = findViewById(R.id.*findSity*)  
  
 val sharedPreferences = getSharedPreferences("loc", *MODE\_PRIVATE*)  
 val editor: SharedPreferences.Editor = sharedPreferences.edit()  
  
 editor.putString("a", "60.597474")  
 editor.putString("b", "56.838011")  
 editor.apply()  
  
 }  
  
 fun SearhButton(view: View) {  
  
 // zapros()  
  
 zapros(edittext.*text*.toString())  
  
  
 }  
  
  
 private fun zapros(sity: String){  
 Log.d("MyLog", "zapros() method called")  
  
 // Проверка наличия разрешения на использование интернета  
  
  
 val url = "https://geocode-maps.yandex.ru/1.x/?apikey=23486e71-2360-45e1-8e8f-9c3c12a63d7f&geocode=$sity&format=json"  
  
 val queue = Volley.newRequestQueue(this)  
  
 val stringRequest = StringRequest(  
 Request.Method.*GET*, url,  
 **{** response **->** // Обработка успешного ответа  
 try {  
 val obj = JSONObject(response)  
 val featureMember = obj.getJSONObject("response")  
 .getJSONObject("GeoObjectCollection")  
 .getJSONArray("featureMember")  
  
 if (featureMember.length() > 0) {  
 val firstObject = featureMember.getJSONObject(0)  
 val point = firstObject.getJSONObject("GeoObject")  
 .getJSONObject("Point")  
 .getString("pos")  
 val tochki = point.toString().*split*(" ")  
  
 searchKarta(tochki)  
  
 Log.d("MyLog", "Coordinates: $point")  
 } else {  
 Log.d("MyLog", "No features found in the response")  
 }  
 } catch (e: JSONException) {  
 Log.d("MyLog", "JSON parsing error: ${e.message}")  
 }  
 **}**,  
 **{** error **->** // Обработка ошибки  
 val statusCode = error.networkResponse.statusCode  
 Log.d("MyLog", "Volley error status code: $statusCode")  
  
  
 **}**)  
  
 // Добавление запроса в очередь  
 queue.add(stringRequest)  
 }  
  
 fun searchKarta(array: List<String>){  
 val imageUrl = "https://static-maps.yandex.ru/v1?ll=${array[0]},${array[1]}&size=450,450&z=13&pt=${array[0]},${array[1]},pmwtm1~${array[0]},${array[1]},pmwtm99&apikey=f9ce7b23-8786-44b7-8308-864c74bf640a"  
  
 Picasso.get().load(imageUrl).into(img)  
  
 val sharedPreferences = getSharedPreferences("loc", *MODE\_PRIVATE*)  
 val editor: SharedPreferences.Editor = sharedPreferences.edit()  
 editor.putString("a", array[0])  
 editor.putString("b", array[1])  
 editor.apply()  
 }  
  
  
  
}

*Item*

package com.example.smartwatch  
  
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,  
  
 //тип  
 @ColumnInfo(name = "type")  
 var type: String,  
  
 //число  
 @ColumnInfo(name = "number")  
 var number: String,  
  
 //факт  
 @ColumnInfo(name = "fact")  
 var fact: String  
)

*ItemAdapter*

package com.example.smartwatch  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.BaseAdapter  
import android.widget.Button  
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\_print\_statistics*, parent, false)  
 holder = ViewHolder(  
 view.findViewById(R.id.*titleTextView*),  
 view.findViewById(R.id.*descriptionTextView*),  
 view.findViewById(R.id.*editButton*),  
 view.findViewById(R.id.*deleteButton*),  
 view.findViewById(R.id.*kilometrov*),  
 )  
 view.*tag* = holder  
 } else {  
 view = convertView  
 holder = view.*tag* as ViewHolder  
 }  
  
 val item = itemList[position]  
 holder.titleTextView.*text* = item.number // Отображаем "number" в заголовке  
 holder.descriptionTextView.*text* = item.fact  
 holder.km.*text* = item.type  
  
 // 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 editButton: ImageButton,  
 val deleteButton: ImageButton,  
 val km: TextView  
 )  
}

*Main\_Screen*

package com.example.smartwatch  
  
import android.app.Activity  
import android.content.Intent  
import android.os.Bundle  
import android.widget.ImageButton  
import com.example.smartwatch.databinding.ActivityMainScreen2Binding  
  
class Main\_Screen : Activity() {  
  
 private lateinit var binding: ActivityMainScreen2Binding  
  
 private lateinit var speak:ImageButton  
 private lateinit var speak2:ImageButton  
 private lateinit var speak3:ImageButton  
 private lateinit var loveButton:ImageButton  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityMainScreen2Binding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 speak=findViewById(R.id.*discussionsImageView*)  
 speak2=findViewById(R.id.*setsImageView*)  
 speak3 = findViewById(R.id.*discussionsImageView1*)  
  
  
 speak.setOnClickListener**{** val intent=Intent(this,MoviesScreen::class.*java*)  
 startActivity(intent)  
 **}** speak2.setOnClickListener **{** val intent=Intent(this, ChatListScreen::class.*java*)  
 startActivity(intent)  
 **}** speak3.setOnClickListener**{** val intent=Intent(this,HomeLocation::class.*java*)  
 startActivity(intent)  
 **}** }  
}

*MainActivity*

package com.example.smartwatch  
  
import android.app.Activity  
import android.content.Context  
import android.content.Intent  
import android.content.SharedPreferences  
import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
import com.example.smartwatch.databinding.ActivityMainBinding  
  
class MainActivity : Activity() {  
  
 private lateinit var binding: ActivityMainBinding  
 private lateinit var button: Button  
 private lateinit var sharedPreferences: SharedPreferences  
 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", "11111")  
 editor.putString("password", "11111")  
 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, Main\_Screen::class.*java*)  
 startActivity(intent)  
 } else {  
 Toast.makeText(this@MainActivity, "Неверный логин или пароль", Toast.*LENGTH\_SHORT*).show()  
  
 }  
  
 }  
  
 }  
}

*MainDb*

package com.example.smartwatch  
  
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()  
 }  
 }  
}

*Movie*

package com.example.smartwatch  
  
data class Movie(  
 val Title: String,  
 val Poster: String  
)

*MoviesScreen*

package com.example.smartwatch  
  
  
import android.app.Activity  
import android.os.Bundle  
import android.util.Log  
import android.view.View  
import android.widget.EditText  
import android.widget.ImageView  
import com.android.volley.Request  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import com.example.smartwatch.databinding.ActivityMoviesScreenBinding  
import com.google.android.gms.maps.model.LatLng  
import com.google.maps.android.SphericalUtil  
import com.squareup.picasso.Picasso  
import org.json.JSONException  
import org.json.JSONObject  
  
class MoviesScreen : Activity() {  
  
 private val apiKeySity = "c72b98a5-ea26-4adf-85de-db5c0cb8c5df"  
 private lateinit var binding: ActivityMoviesScreenBinding  
  
 private lateinit var img: ImageView  
 private lateinit var edittext: EditText  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityMoviesScreenBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 img = findViewById(R.id.*imgKarta*)  
 edittext = findViewById(R.id.*findSity*)  
  
 }  
  
 fun SearhButton(view: View) {  
  
 // zapros()  
  
 zapros(edittext.*text*.toString())  
  
  
 }  
  
  
 private fun zapros(sity: String){  
 Log.d("MyLog", "zapros() method called")  
  
 // Проверка наличия разрешения на использование интернета  
  
  
 val url = "https://geocode-maps.yandex.ru/1.x/?apikey=23486e71-2360-45e1-8e8f-9c3c12a63d7f&geocode=$sity&format=json"  
  
 val queue = Volley.newRequestQueue(this)  
  
 val stringRequest = StringRequest(  
 Request.Method.*GET*, url,  
 **{** response **->** // Обработка успешного ответа  
 try {  
 val obj = JSONObject(response)  
 val featureMember = obj.getJSONObject("response")  
 .getJSONObject("GeoObjectCollection")  
 .getJSONArray("featureMember")  
  
 if (featureMember.length() > 0) {  
 val firstObject = featureMember.getJSONObject(0)  
 val point = firstObject.getJSONObject("GeoObject")  
 .getJSONObject("Point")  
 .getString("pos")  
 val tochki = point.toString().*split*(" ")  
 //txt.text = "${tochki[0]}UU${tochki[1]} "  
 searchKarta(tochki)  
  
 Log.d("MyLog", "Coordinates: $point")  
 } else {  
 Log.d("MyLog", "No features found in the response")  
 }  
 } catch (e: JSONException) {  
 Log.d("MyLog", "JSON parsing error: ${e.message}")  
 }  
 **}**,  
 **{** error **->** // Обработка ошибки  
 val statusCode = error.networkResponse.statusCode  
 Log.d("MyLog", "Volley error status code: $statusCode")  
  
  
 **}**)  
  
 // Добавление запроса в очередь  
 queue.add(stringRequest)  
 }  
  
 fun searchKarta(array: List<String>){  
 val imageUrl = "https://static-maps.yandex.ru/v1?ll=${array[0]},${array[1]}&size=450,450&z=13&pt=${array[0]},${array[1]},pmwtm1~${array[0]},${array[1]},pmwtm99&apikey=f9ce7b23-8786-44b7-8308-864c74bf640a"  
  
 Picasso.get().load(imageUrl).into(img)  
  
  
 val sharedPreferences = getSharedPreferences("loc", *MODE\_PRIVATE*)  
  
 val a= sharedPreferences.getString("a", "")  
 val b = sharedPreferences.getString("b", "")  
 val point1: LatLng  
 if(a.*toString*().*isEmpty*() || b.*toString*().*isEmpty*()){  
 point1 = LatLng(60.597474, 56.838011)  
 }  
 else{  
 point1 = LatLng(a.*toString*().*toDouble*(), b.*toString*().*toDouble*())  
 }  
  
 val point2 = LatLng(array[0].*toDouble*(), array[1].*toDouble*())  
  
 val distance = SphericalUtil.computeDistanceBetween(point1, point2)  
 Log.d("MyLog", "Расстояние $distance метров")  
  
 val shag = Math.round(distance \* 0.60 + distance)  
 val kallroia = Math.round(distance \* 0.035)  
 val formattedDistance = String.*format*("%.1f", distance / 1000)  
 val kilometr = formattedDistance.*toDouble*()  
 //запись в бд  
 Log.d("MyLog", "Шагов $distance ")  
  
 val db = MainDb.getDb(this)  
 val item = Item(null, kilometr.toString(), shag.toString(), kallroia.toString())  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
 Log.d("MyLog", "Ну вроде сохранилось")  
 }  
  
  
  
}

*PrintStatistics*

package com.example.smartwatch  
  
import android.app.Activity  
import android.os.Bundle  
import androidx.appcompat.app.AppCompatActivity  
import com.example.smartwatch.databinding.ActivityPrintStatisticsBinding  
  
class PrintStatistics : AppCompatActivity() {  
  
 private lateinit var binding: ActivityPrintStatisticsBinding  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityPrintStatisticsBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 }  
}

*activity\_home\_location.xaml*

<?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=".HomeLocation"  
 tools:deviceIds="wear"  
 android:background="@color/lightBlue">  
  
 <LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <ImageView  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:id="@+id/imgKarta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"></ImageView>  
 <EditText  
 android:textColor="@color/gray"  
  
 android:id="@+id/findSity"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"  
 android:hint="Введите город"/>  
  
  
 <Button  
 android:onClick="SearhButton"  
 android:id="@+id/searchBut"  
 android:text="@string/searcttetx"  
 android:background="@color/blue"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
 </LinearLayout>  
</androidx.wear.widget.BoxInsetLayout>

*activity\_chat\_lust\_screen*

<?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=".ChatListScreen"  
 tools:deviceIds="wear"  
 android:background="@color/lightBlue">  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button  
 android:background="@color/blue"  
 android:id="@+id/clearButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="очистить"  
 android:textColor="@color/white"  
 android:layout\_margin="4dp"  
 android:onClick="ButCleaAll"/>  
  
  
 <Button  
 android:textColor="@color/white"  
 android:background="@color/blue"  
 android:id="@+id/AddButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="добавить"  
 android:layout\_margin="4dp"  
 android:onClick="AddNewButton"/>  
  
  
 </LinearLayout>  
 <ListView  
 android:layout\_marginTop="55dp"  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1" />  
</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/lightBlue">  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_boxedEdges="all">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Вход"  
 android:textSize="25dp"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="10dp"  
 android:textStyle="bold"  
 android:textColor="@color/white"/>  
  
 <EditText  
 android:textColor="@color/gray"  
 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="email"/>  
 <EditText  
 android:textColor="@color/gray"  
 android:inputType="textPassword"  
 android:id="@+id/pass"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="90dp"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"  
 android:hint="пароль"/>  
  
 <Button  
 android:onClick="perehod"  
 android:id="@+id/singin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="140dp"  
 android:text="Войти"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:textAllCaps="false"  
 android:background="@color/blue"/>  
 </FrameLayout>  
</androidx.wear.widget.BoxInsetLayout>

*activity\_main\_screen\_xml*

<?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=".Main\_Screen"  
 tools:deviceIds="wear"  
 android:background="@color/lightBlue">  
  
 <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:padding="9dp"  
 android:orientation="vertical">  
  
 <ImageButton  
 android:id="@+id/discussionsImageView1"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_marginStart="5dp"  
 android:background="@drawable/homeicon"/>  
  
 <TextView  
 android:layout\_marginStart="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
  
 android:text="Дом"  
 android:textSize="10.5sp"  
 android:textColor="@color/white"  
 tools:ignore="SmallSp" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="80dp"  
 android:padding="8dp"  
 android:orientation="vertical">  
  
 <ImageButton  
 android:id="@+id/discussionsImageView"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_marginStart="5dp"  
 android:background="@drawable/icon2"/>  
  
 <TextView  
 android:layout\_marginStart="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:text="Карта"  
 android:textSize="10.5sp"  
 android:textColor="@color/white"  
 tools:ignore="SmallSp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="90dp"  
 android:layout\_marginLeft="15dp"  
 android:orientation="vertical">  
  
 <ImageButton  
 android:id="@+id/setsImageView"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_marginEnd="5dp"  
 android:background="@drawable/icon3"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:text="Статистика"  
 android:textSize="10.5sp"  
 android:layout\_marginStart="1dp"  
 android:textColor="@color/white"  
 tools:ignore="SmallSp" />  
  
 </LinearLayout>  
  
  
 </FrameLayout>  
</androidx.wear.widget.BoxInsetLayout>

*activity\_movies\_screen*

<?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=".MoviesScreen"  
 android:background="@color/lightBlue"  
 tools:deviceIds="wear">  
  
 <LinearLayout  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <ImageView  
 android:adjustViewBounds="true"  
 android:scaleType="centerCrop"  
 android:id="@+id/imgKarta"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"></ImageView>  
 <EditText  
 android:textColor="@color/gray"  
  
 android:id="@+id/findSity"  
 android:layout\_width="160dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"  
 android:hint="Введите город"/>  
  
  
 <Button  
 android:onClick="SearhButton"  
 android:id="@+id/searchBut"  
 android:text="@string/searcttetx"  
 android:background="@color/blue"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
 </LinearLayout>  
  
</androidx.wear.widget.BoxInsetLayout>

*activity\_print\_statistick.xaml*

<?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:padding="@dimen/box\_inset\_layout\_padding"  
 tools:context=".PrintStatistics"  
 tools:deviceIds="wear"  
android:background="@color/lightBlue"  
 >  
  
 <LinearLayout  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:textColor="@color/white"  
 android:text="Расстояние в км"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
  
 android:textColor="@color/white"  
 android:id="@+id/kilometrov"  
 android:layout\_width="105dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:textColor="@color/white"  
 android:text="Шаги"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:id="@+id/titleTextView"  
 android:layout\_width="120dp"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="4dp"  
 android:textColor="@color/white"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/editButton"  
 android:background="@color/lightBlue"  
 android:src="@drawable/redact"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"></ImageButton>  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:textColor="@color/white"  
 android:text="Калории"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
  
 android:textColor="@color/white"  
 android:id="@+id/descriptionTextView"  
 android:layout\_width="95dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/deleteButton"  
 android:background="@color/lightBlue"  
 android:src="@drawable/mus"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp">  
  
 </ImageButton>  
  
 </LinearLayout>  
  
  
 </LinearLayout>  
  
</LinearLayout>

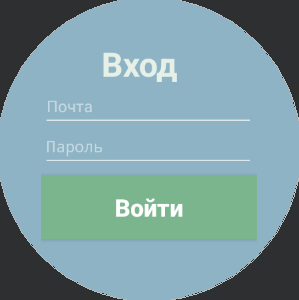
## **Используемые библиотеки**

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 'androidx.appcompat:appcompat:1.6.1'  
implementation 'com.google.android.material:material:1.4.0'  
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
  
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
  
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'  
  
implementation 'com.squareup.picasso:picasso:2.71828'  
  
implementation 'com.android.volley:volley:1.2.1'  
implementation 'com.google.android.gms:play-services-maps:18.2.0'  
implementation 'com.google.maps.android:android-maps-utils:0.5'  
  
implementation 'androidx.room:room-ktx:2.4.3'  
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'  
kapt 'androidx.room:room-compiler:2.4.3'  
implementation 'androidx.appcompat:appcompat:1.3.0'

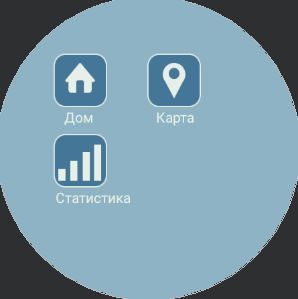
## **Тестовые случаи**

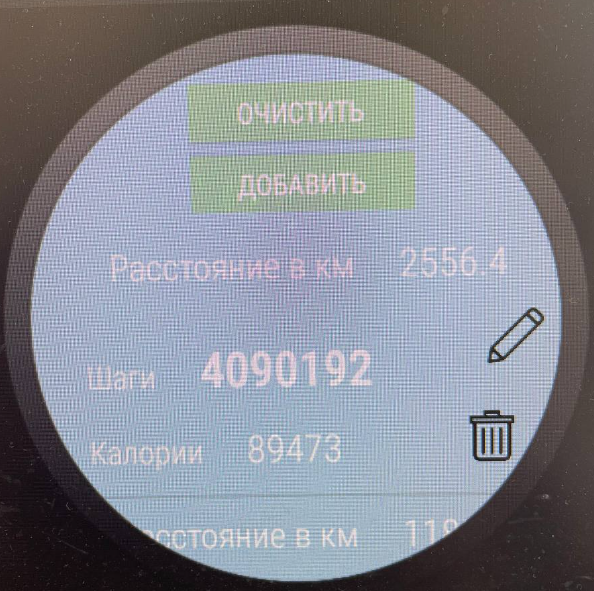
Если пользователь попытается изменить данные, и введет отрицательное число и букву, информация не запишется. так же есть проверки на правильность и пустоту пароля

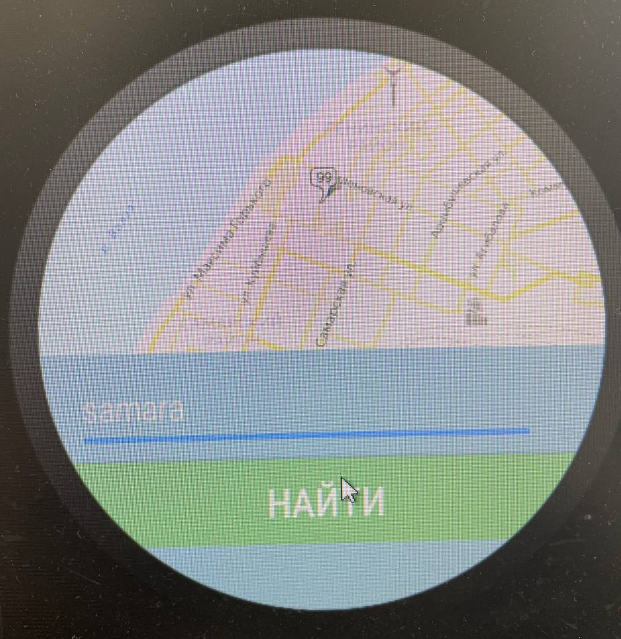
## **Экраны**











## **Используемые инструменты**

kotlin, xml, android studio

## **Вывод:**

Я разработал мобильное приложение – умные часы, используя запросы api,базу данных room и sharedpreferens