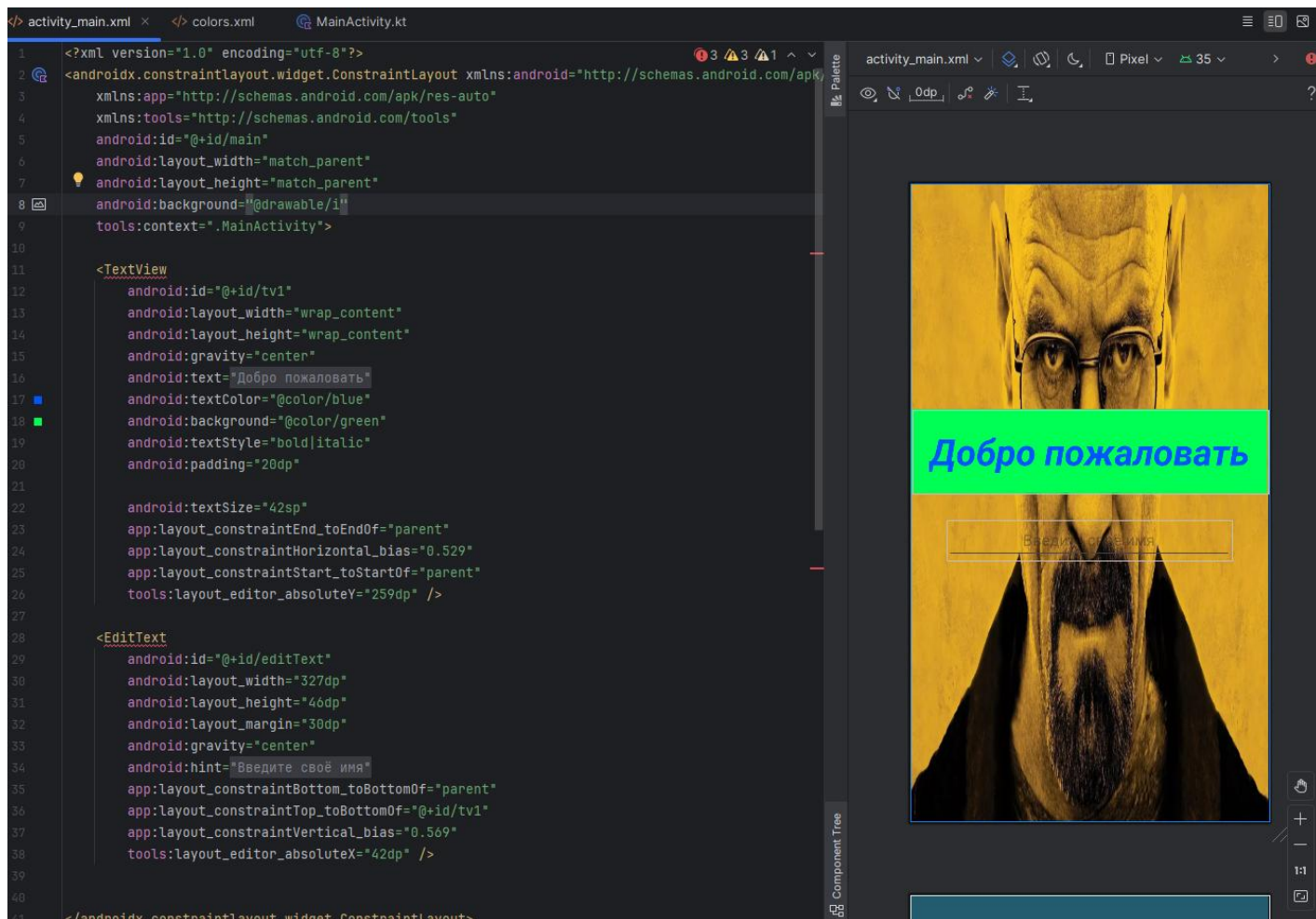


ПРАКТИЧЕСКАЯ РАБОТА №1

Создал текстовое поле, покрасил текст, добавил ему фон

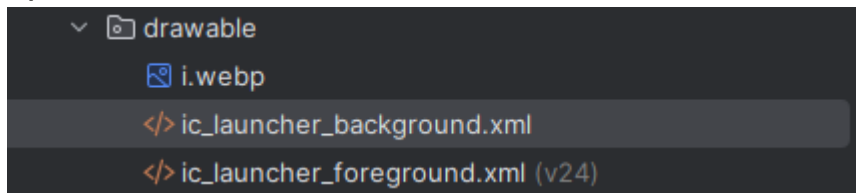
Создал поле ввода, привязал к текстовому полю, добавил подпись

Поменял фон

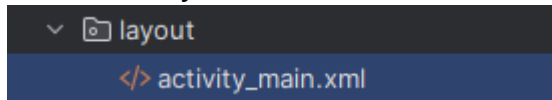


1.1

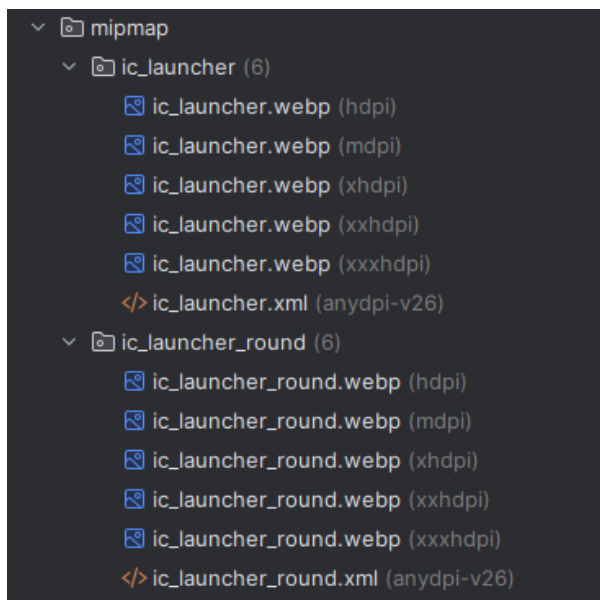
В папке `drawable` находятся все графические объекты, например, фон приложения:



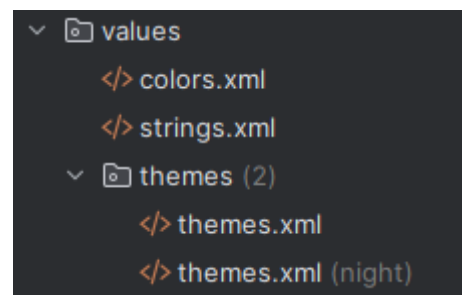
В папке `layout` находится главный файл редактора :



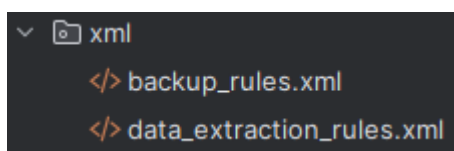
В папке `mipmap` находится ещё две папки — это иконки в лаунчере разных размеров:



В папке `values` находятся файлы с постоянными переменными, а также папка `themes` с файлами настроек тем приложения:

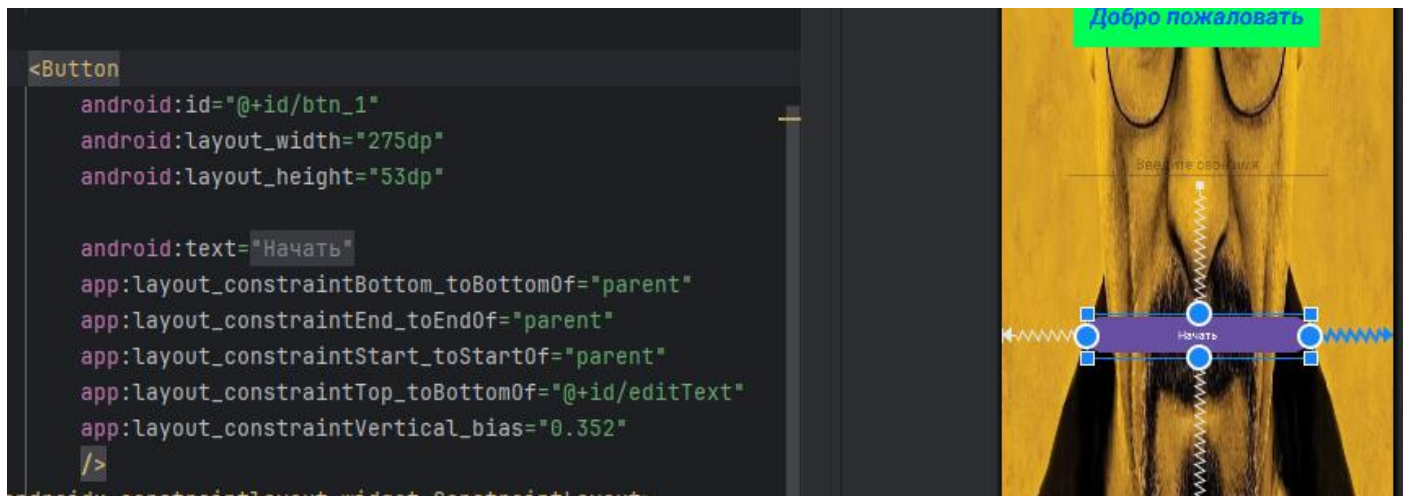


В папке `xml` находятся правила бэкапа и экспорта данных:

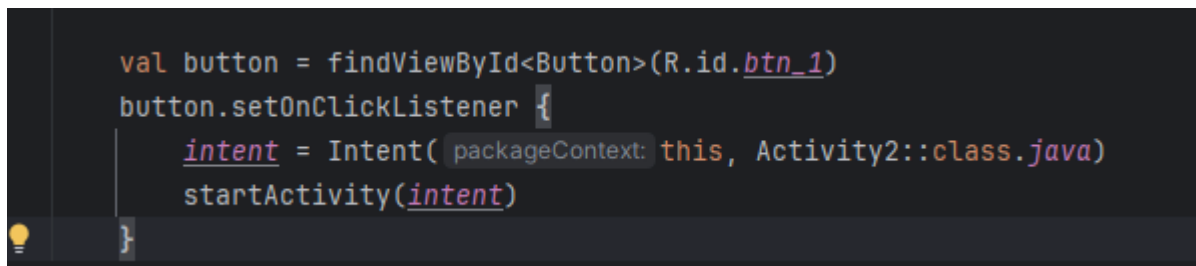


2

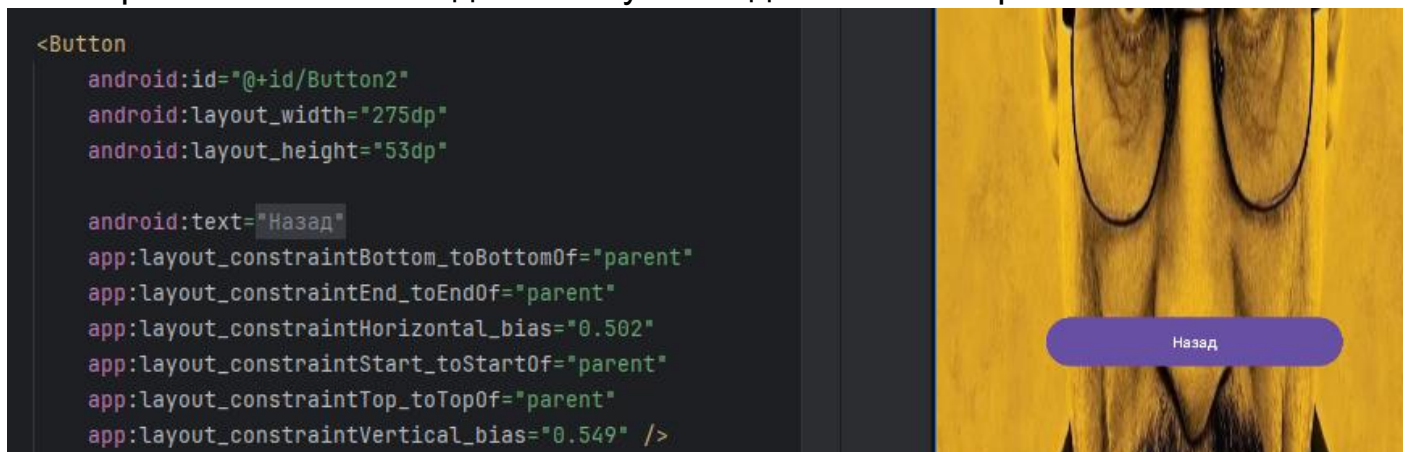
Добавил кнопку:



Задал ей функцию перемещения на вторую активность:



Во второй активности создал кнопку "Назад" и поменял фон:



3

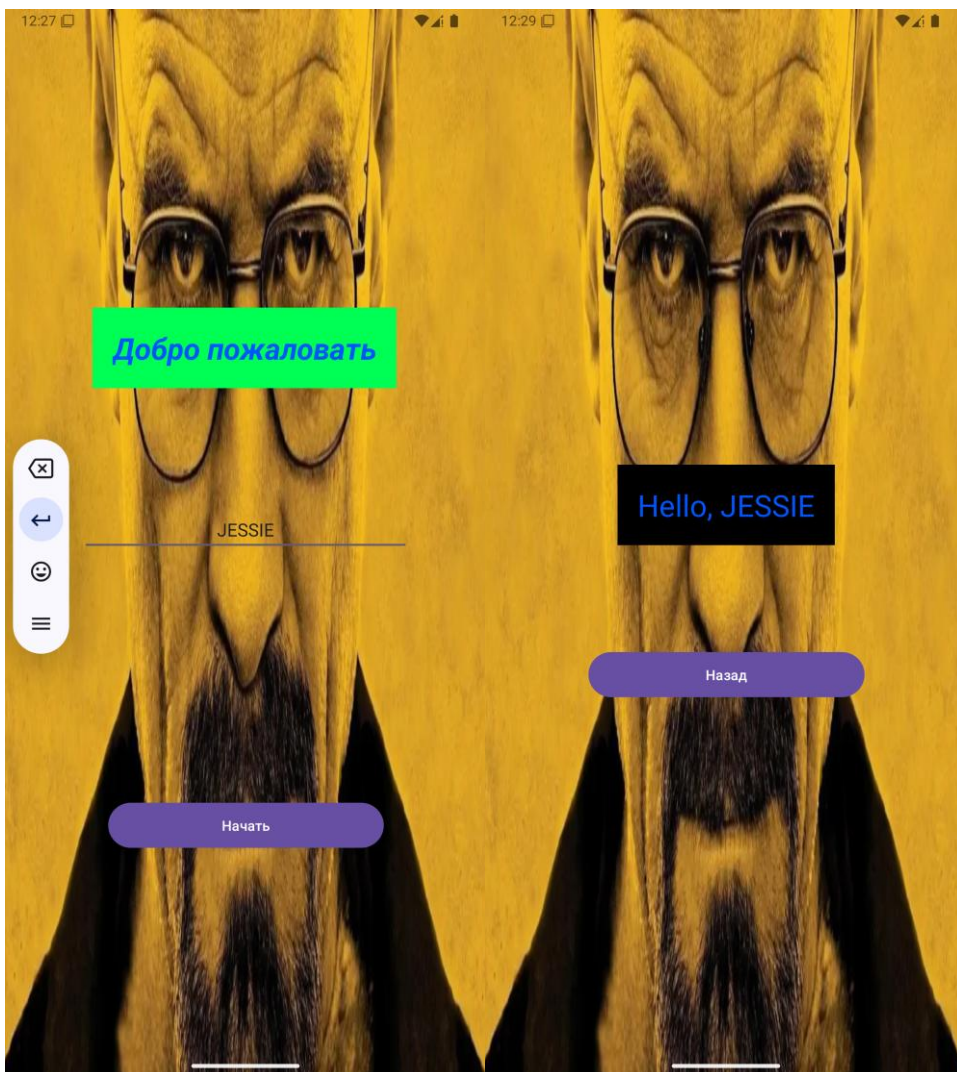
Сделал сохранение введённого текста и вывод его на второй активности:

MainActivity:

```
val editText = findViewById<EditText>(R.id.editText)
val button = findViewById<Button>(R.id.btn_1)
button.setOnClickListener {
    val enteredText = editText.text.toString().trim()
    intent = Intent( packageContext: this, Activity2::class.java)
    intent.putExtra( name: "text", enteredText)
    startActivity(intent)
```

Activity2:

```
val text : TextView = findViewById(R.id.tv)
val message = intent.getStringExtra( name: "text")
text.text = message
```



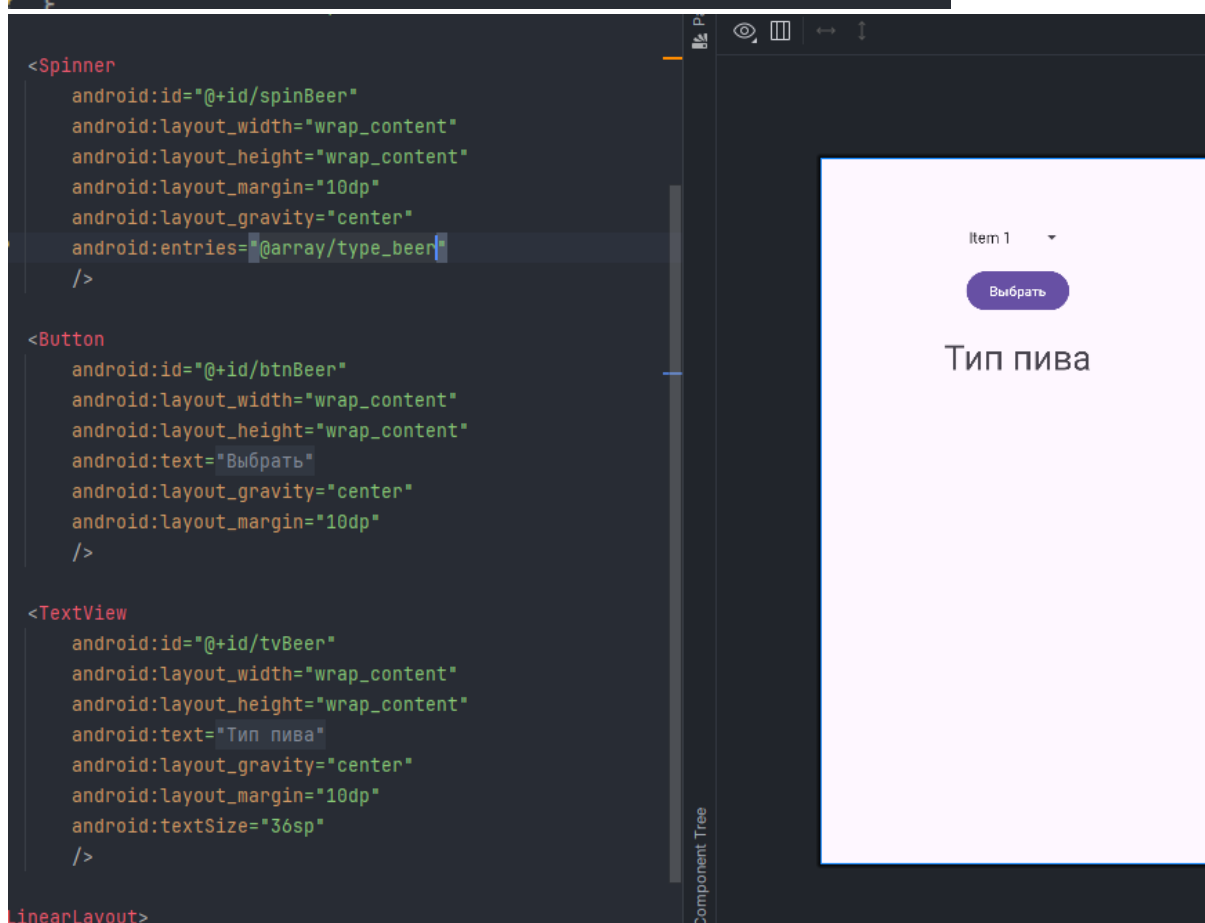
4.

Создал новый проект, по заданию оформил, написал код кнопки:

```
        insets
    }

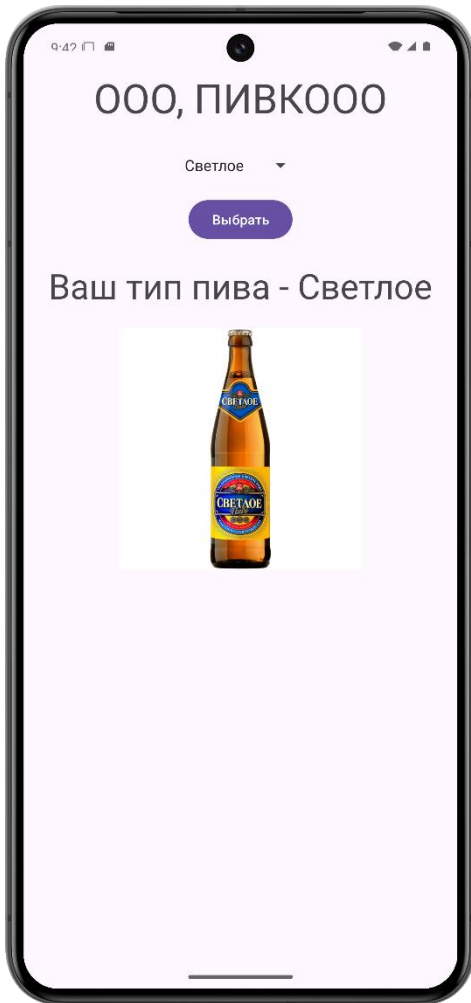
    val btnBeer:Button=findViewById(R.id.btnBeer)
    val tvBeer:TextView=findViewById(R.id.tvBeer)
    val spinnerBeer:Spinner=findViewById(R.id.spinBeer)

    btnBeer.setOnClickListener{
        val type = spinnerBeer.selectedItem
        tvBeer.text=("Ваш тип пива $type")
    }
}
```



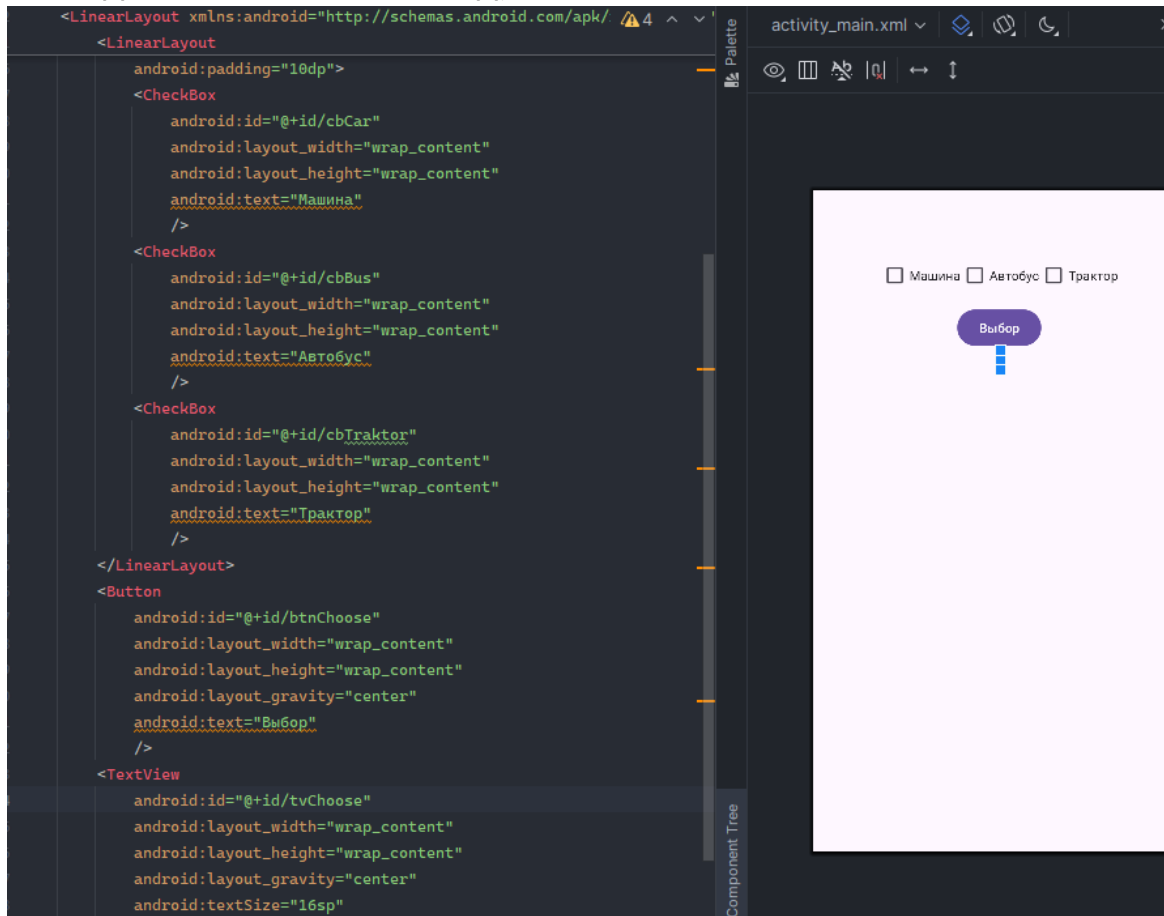
(да, я поменял тему андроид студио)

Добавил некоторый функционал, теперь приложение отображает картинку с выбранным пивом:



5

По заданию элементы и код, как обычно:

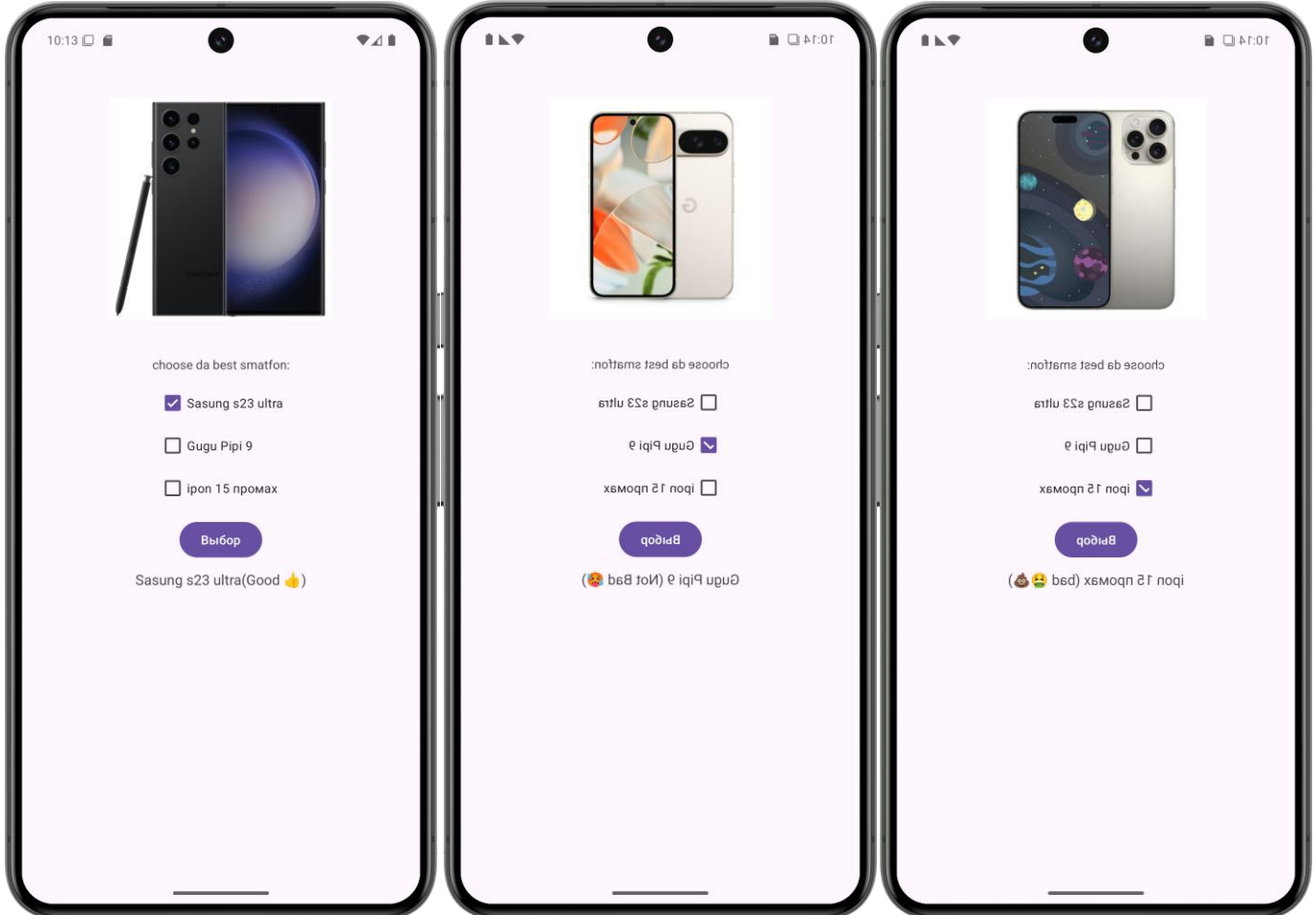


```
class MainActivity : AppCompatActivity() {  
    private lateinit var binding : ActivityMainBinding  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        enableEdgeToEdge()  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {  
            v, insets ->  
                val systemBars =  
                    insets.getInsets(WindowInsetsCompat.Type.systemBars())  
                v.setPadding(systemBars.left, systemBars.top, systemBars.right,  
                    systemBars.bottom)  
                insets  
            }  
        }  
        val btn : Button = findViewById(R.id.btnChoose)  
        val tvChos : TextView = findViewById(R.id.tvChoose)  
        btn.setOnClickListener {  
            val result = StringBuilder()  
            if (binding.cbCar.isChecked){  
                tvChos.text = binding.cbCar.text  
            }else if (binding.cbBus.isChecked){  
                tvChos.text = binding.cbBus.text  
            }else if (binding.cbTraktor.isChecked){  
                tvChos.text = binding.cbTraktor.text  
            }  
        }  
    }  
}
```

В build.gradle.kts включил функцию viewBinding

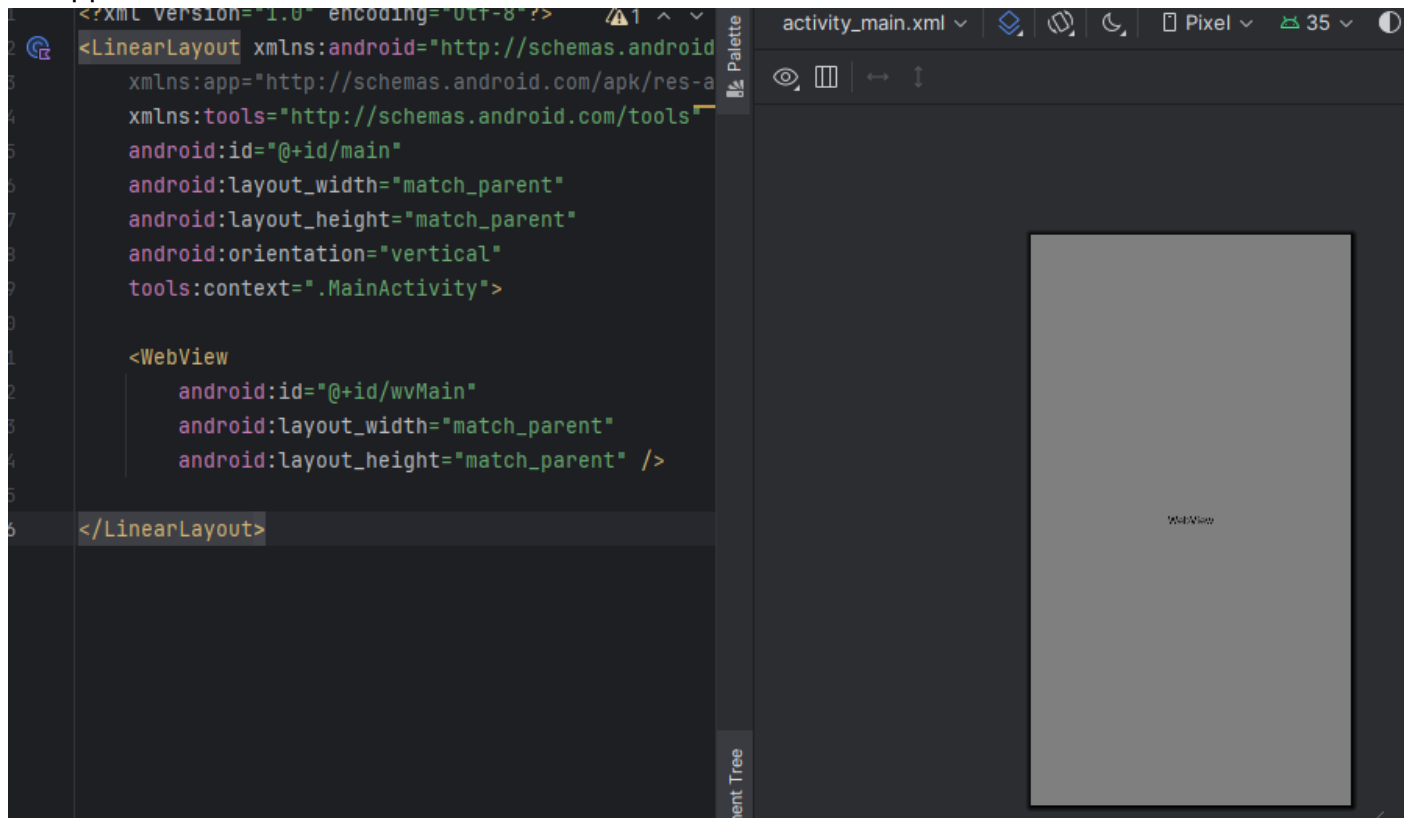
Добавил также строчку, чтоб отображалась надпись трактора, т.к. её не было
(да, ещё и шрифт поменял, ужс)

Поменял функционал, строчки и добавил приколы с картинками:



5.2

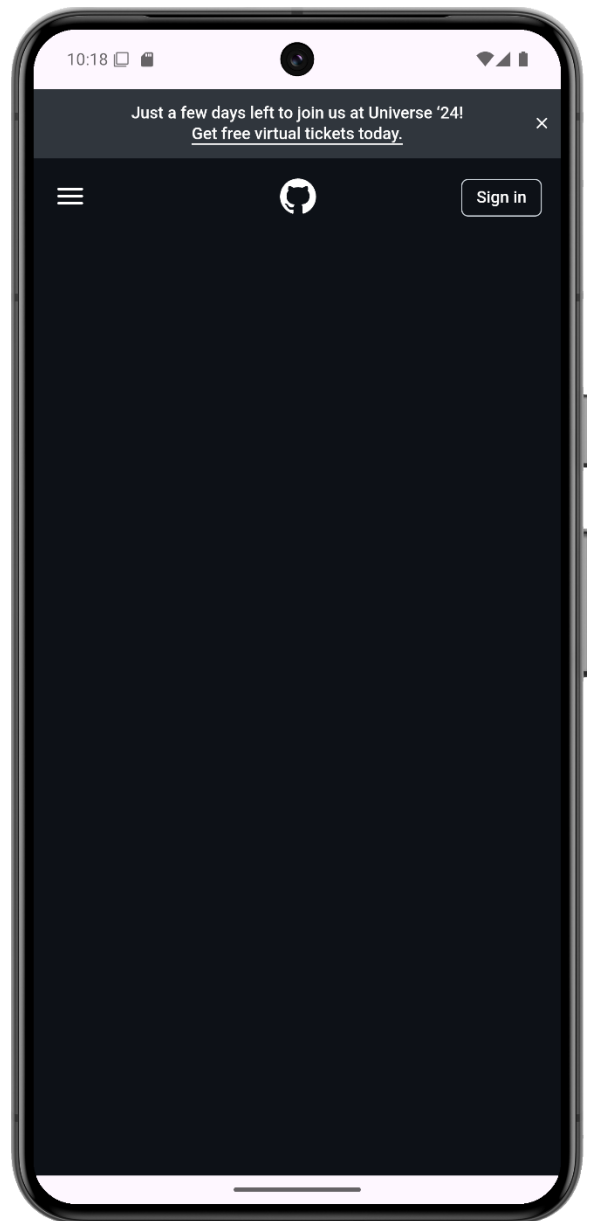
Создал WebView:



```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        binding=ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
        binding.wvMain.loadUrl(url: "https://github.com")
    }
}
```

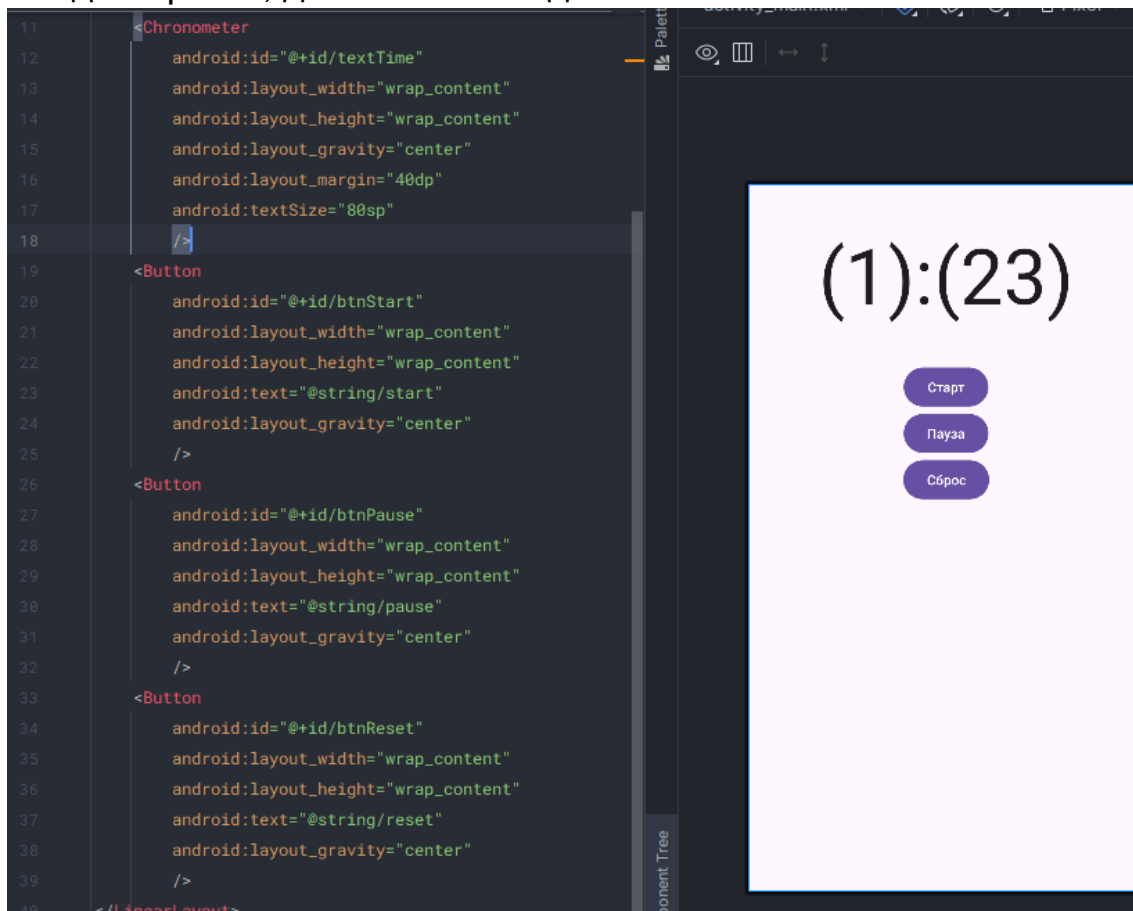
В манифесте также прописал разрешение на доступ к интернету

Работает, что ещё добавить, не знаю:

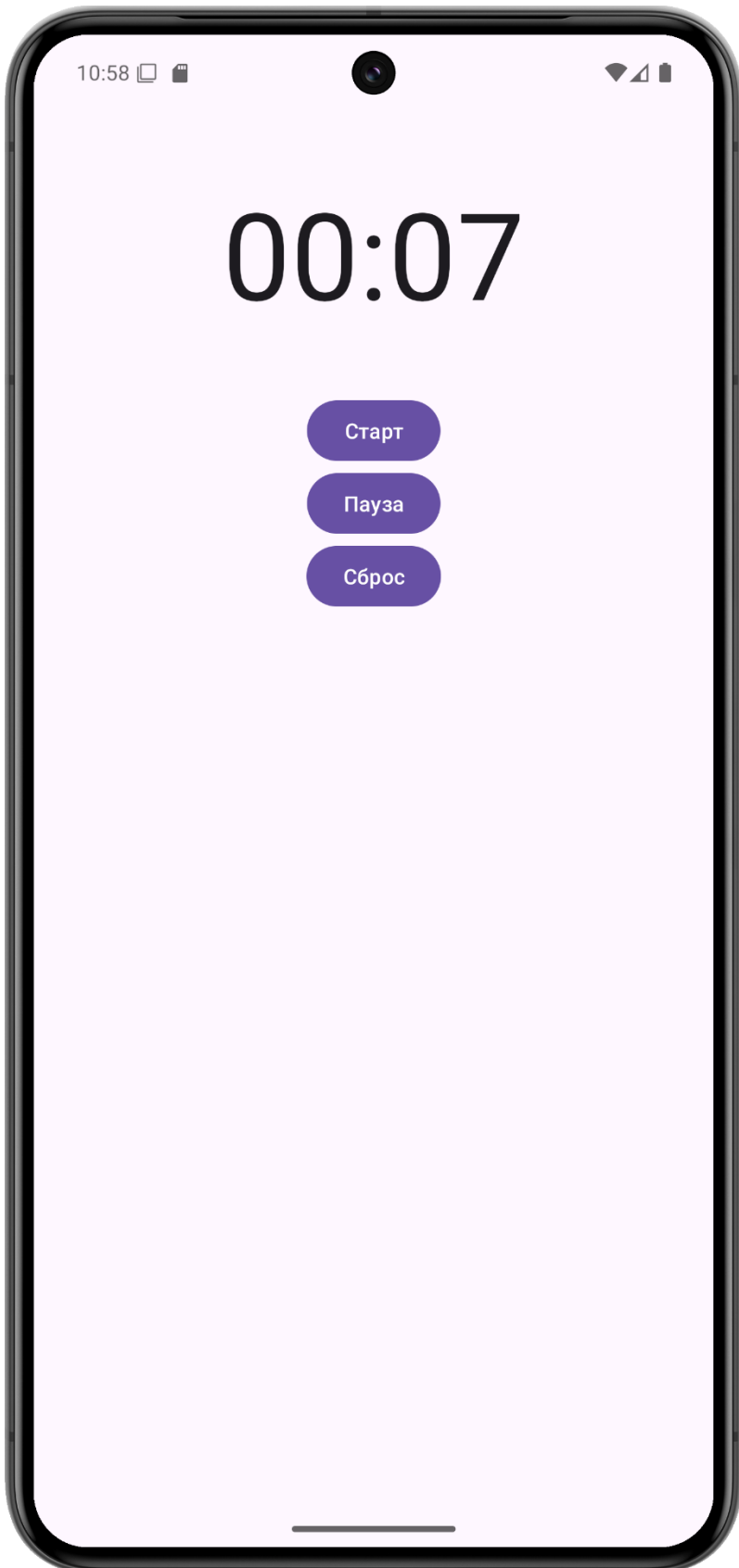


6.

Создал проект, добавил всё код написал:



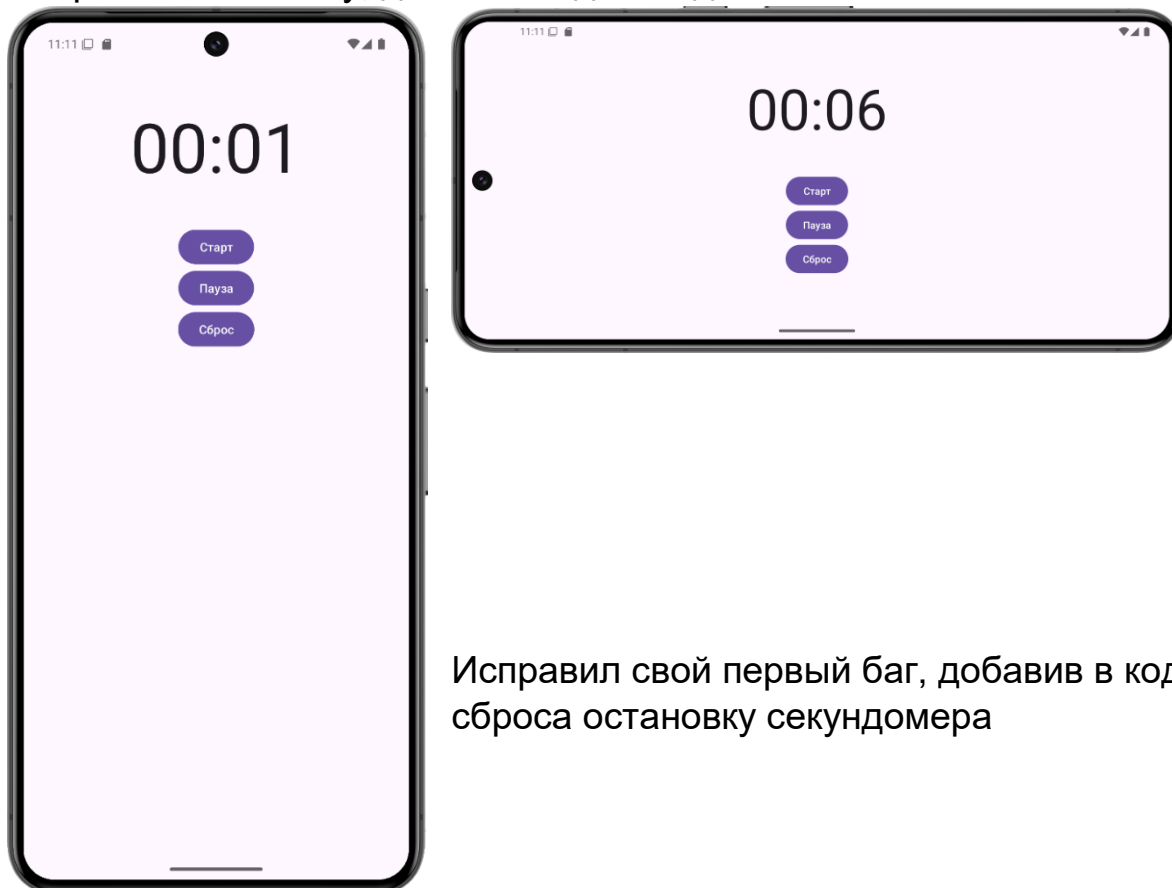
Тест:



Сразу заметил баг, если запустить, далее сбросить секундомер, то время идёт и пауза не нажимается

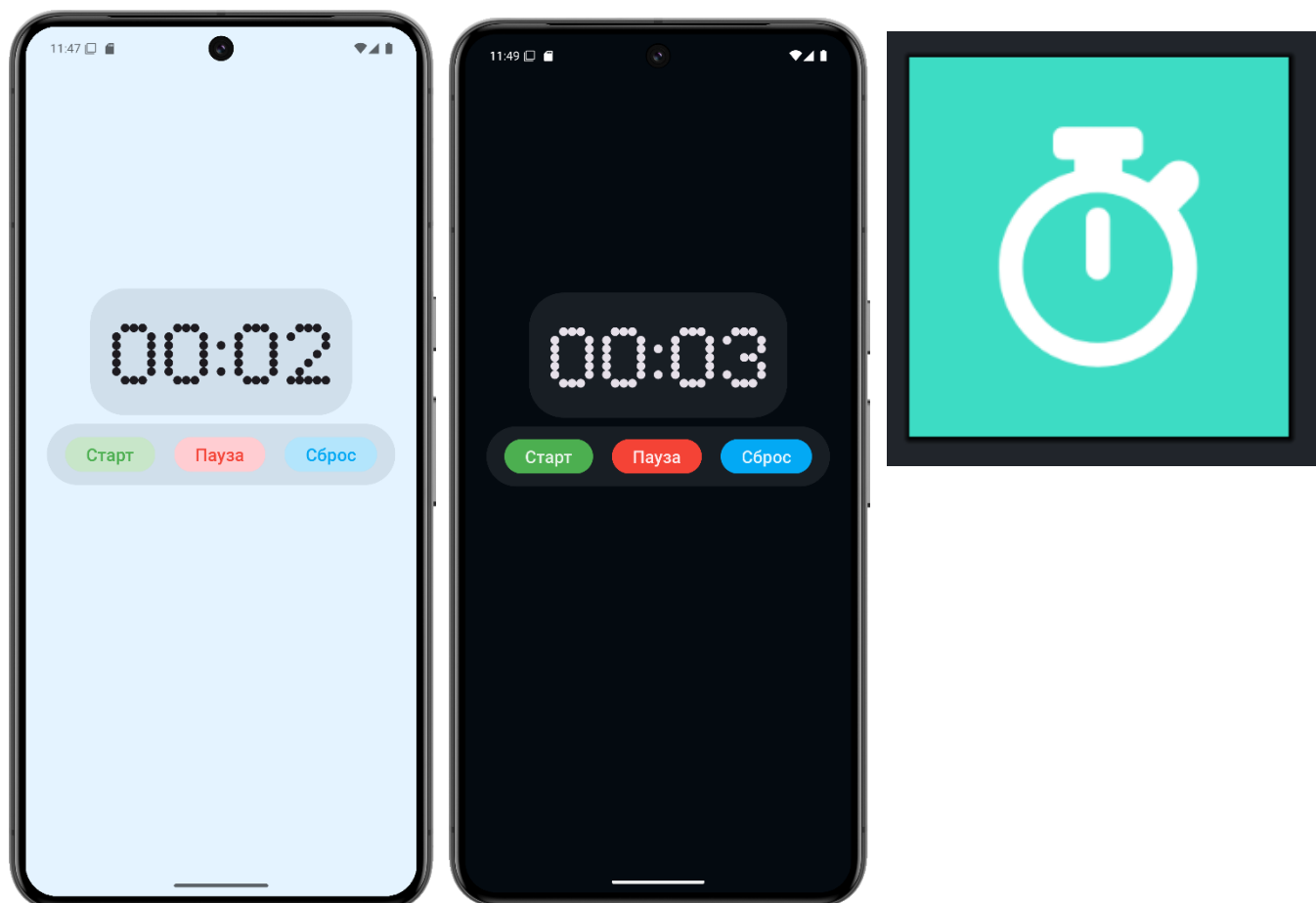
Ну и при повороте время и вправду сбрасывается

Исправил остановку, добавив код из задания:



Исправил свой первый баг, добавив в код кнопки сброса остановку секундомера

Поменял дизайн, добавил свой шрифт и нормальную поддержку тёмной темы и сменил иконку:



7.

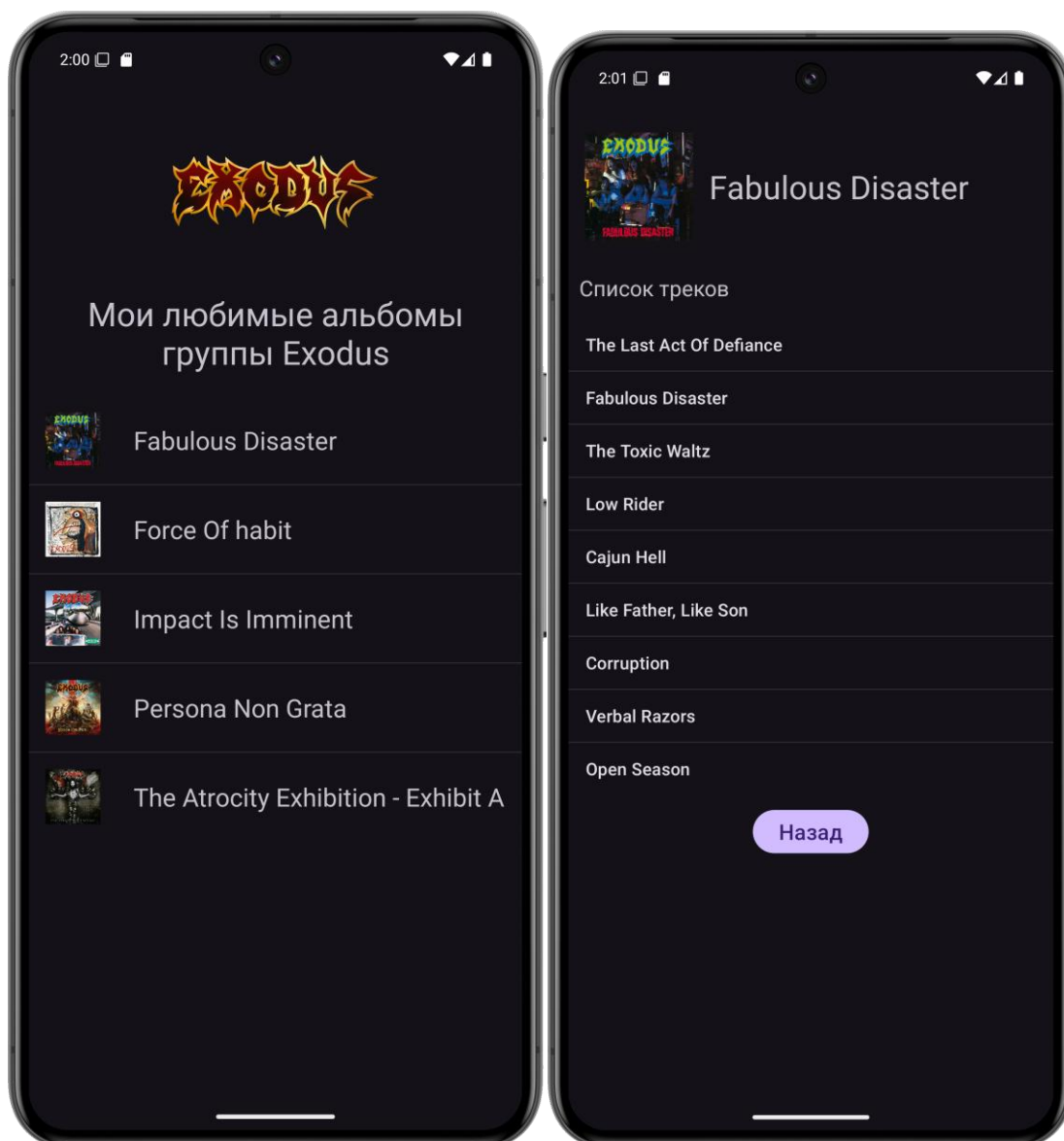
Создал проект, в нём ListView, код:

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        enableEdgeToEdge()  
        setContentView(R.layout.activity_main)  
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
            insets  
        }  
  
        val listView:ListView=findViewById(R.id.lvMain)  
        val arrayAdapter:ArrayAdapter<*>  
        val names = arrayOf("Oleg","Max","Gleb","Walter","Vladimir")  
        arrayAdapter= ArrayAdapter<context: this,android.R.layout.simple_list_item_1,names)  
        listView.adapter=arrayAdapter  
        listView.setOnItemClickListener { parent, view, position, id ->  
            val selection=names[position]  
            Toast.makeText( context: this, text: "Вы нажали на элементы ListView", Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```

Тест:



Поменял немного суть задания, сделал список любимых альбомов одной группы:



Создал проект, написал код, как всегда крч:

```
private fun getUserdata(){
    for (i in imageid.indices){
        val data = DataClass(imageid[i],textId[i],textBonkId[i])
        newList.add(data)
    }
    newRecyclerView.adapter=MyAdapter(newArrayList)
}
}

data class DataClass(var titleImage:Int,var heading:String, var bonktxt:String){

class MyAdapter(private val newList:ArrayList<DataClass>):RecyclerView.Adapter<MyAdapter.MyViewHolder>(){

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyAdapter.MyViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item,parent, attachToRoot: false)
        return MyViewHolder(itemView)
    }

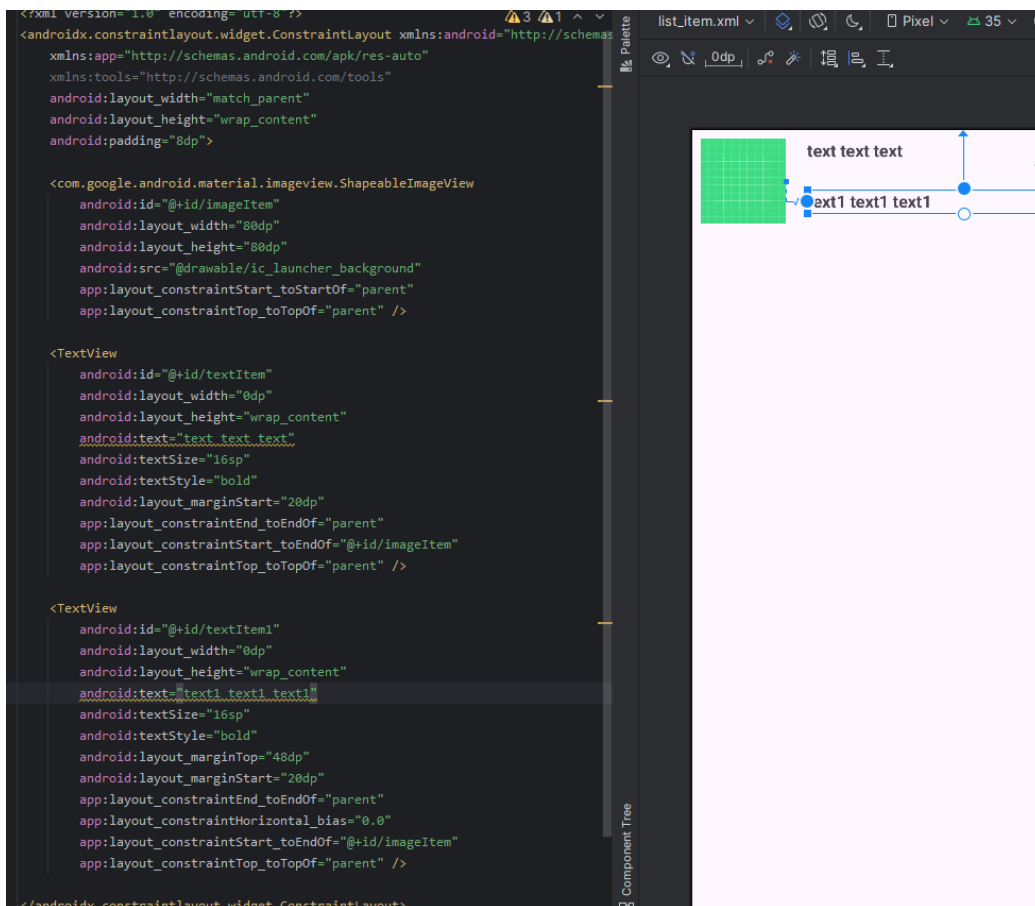
    override fun getItemCount(): Int {
        return newList.size
    }

    override fun onBindViewHolder(holder: MyAdapter.MyViewHolder, position: Int) {
        val currentItem = newList[position]
        holder.imageView.setImageResource(currentItem.titleImage)
        holder.txtHead.text = currentItem.heading
        holder.txtBonk.text = currentItem.bonktxt

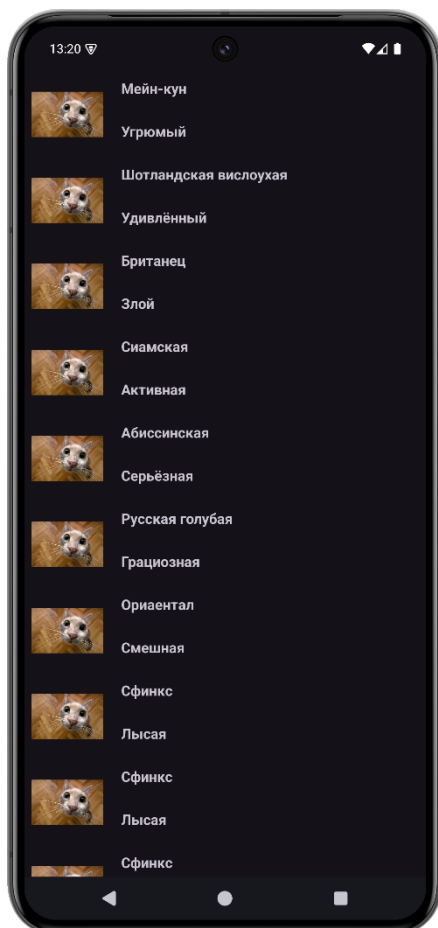
        holder.itemView.setOnClickListener{
            if (position==0)
                Toast.makeText(holder.itemView.context, "1 элемент", Toast.LENGTH_SHORT).show()
        }
    }
}

class MyViewHolder(itemView: View):RecyclerView.ViewHolder(itemView){
    val imageView : ShapeableImageView = itemView.findViewById(R.id.imageItem)
    val txtHead:TextView=itemView.findViewById(R.id.textItem)
    val txtBonk:TextView=itemView.findViewById(R.id.textItem1)
}
}
```

Активити листа:

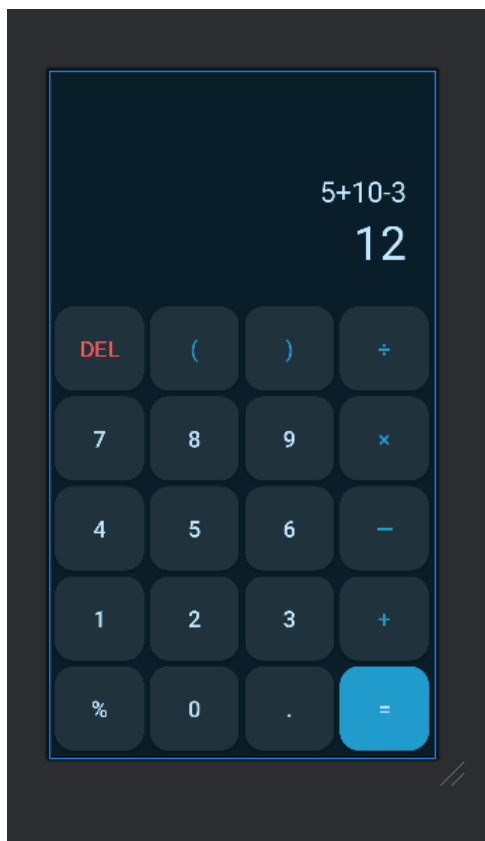


Само приложение:

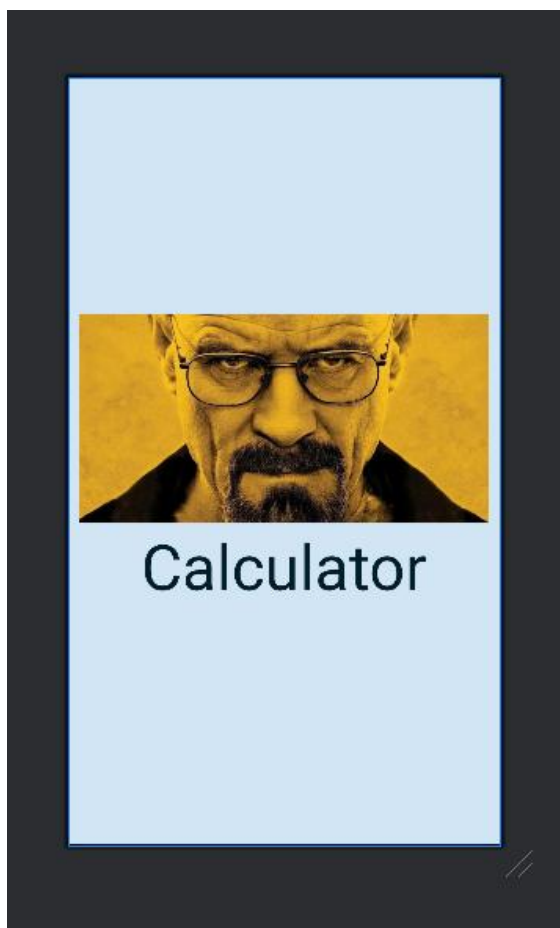


9.

Создал макет, дизайн под себя:



Макет splash_screen:



Код MainActivity:

```
private lateinit var binding: ActivityMainBinding
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
        val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
        insets
    }
    binding.buttonClear.setOnClickListener{
        binding.input.text = " "
        binding.output.text = " "
    }
    binding.buttonBracketLeft.setOnClickListener {
        addToInputText( value: "(")
    }
    binding.buttonBracketRight.setOnClickListener {
        addToInputText( value: ")")
    }
    binding.button0.setOnClickListener {
        addToInputText( value: "0")
    }
    binding.button1.setOnClickListener {
        addToInputText( value: "1")
    }
    binding.button2.setOnClickListener {
        addToInputText( value: "2")
    }
    binding.button3.setOnClickListener {
        addToInputText( value: "3")
    }
    binding.button4.setOnClickListener {
        addToInputText( value: "4")
    }
    binding.button5.setOnClickListener {
        addToInputText( value: "5")
    }
    binding.button6.setOnClickListener {
        addToInputText( value: "6")
    }
}
```

```

    }
    binding.buttonEquals.setOnClickListener {
        showResult()
    }
    binding.buttonPercent.setOnClickListener {
        addToInputText( value: "%")
    }
}

private fun addToInputText(value: String) {
    binding.input.append(value)
}

private fun getInputExpression(): String {
    return binding.input.text.toString()
}

private fun showResult() {
    try {
        val expression = getInputExpression().replace( oldValue: "%", newValue: "/100")
        val result = ExpressionBuilder(expression).build().evaluate()
        binding.output.text =
            DecimalFormat( pattern: "0.#####").format(result).toString()
        binding.output.setTextColor(
            ContextCompat.getColor( context: this,
                R.color.txtSec))
    } catch (e: Exception) {
        binding.output.text = "Ошибка"
        binding.output.setTextColor(
            ContextCompat.getColor(
                context: this,
                R.color.acc_red
            )
        )
    }
}
}

```


SplashScreenActivity:

```
class SplashScreenActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        enableEdgeToEdge()  
        setContentView(R.layout.activity_splash_screen)  
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
            insets  
        }  
        android.os.Handler(Looper.getMainLooper()).postDelayed({  
            val intent = Intent( packageContext: this,MainActivity::class.java)  
            startActivity(intent)  
        }, delayMillis: 2000)  
    }  
}
```

12.

Создал макеты:

Почта

Пароль

Подтвердите пароль

РЕГИСТРАЦИЯ

Уже есть аккаунт!

Почта

Пароль

ВХОД

Нет аккаунта? Нужна регистрация

Код MainActivity:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {

        btnSignUp.setOnClickListener {
            singUpUser()
        }
        tvRedirectLogin.setOnClickListener{
            intent = Intent( packageContext: this, LoginActivity::class.java)
            startActivity(intent)
        }
    }

    private fun singUpUser() {
        val email = etEmail.text.toString()
        val pass = etPass.text.toString()
        val confPass = etConfPass.text.toString()
        if (email.isBlank() || pass.isBlank() || confPass.isBlank()) {
            Toast.makeText( context: this, text: "Почта и пароль не могут быть пустыми", Toast.LENGTH_SHORT).show()
            return
        }
        if (pass != confPass) {
            Toast.makeText( context: this, text: "Пароль и пароль подтверждения не совпадают", Toast.LENGTH_SHORT).show()
            return
        }

        auth.createUserWithEmailAndPassword(email,pass).addOnCompleteListener(this){
            if (it.isSuccessful){
                Toast.makeText( context: this, text: "Регистрация успешна",Toast.LENGTH_SHORT).show()
                intent = Intent( packageContext: this, MainActivity::class.java)
                startActivity(intent)
            }else {
                Toast.makeText( context: this, text: "Ошибка регистрации",Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

LoginActivity:

```

class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        setContentView(R.layout.activity_login)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {
            v, insets ->
            val systemBars =
                insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
                systemBars.bottom)
            insets
        }

        tvRedirectSignUp = findViewById(R.id.tvRedirectSignUp)
        btnLogin = findViewById(R.id.btnLogin)
        etEmail = findViewById(R.id.etEmailAddress)
        etPass = findViewById(R.id.etPassword)

        auth = FirebaseAuth.getInstance()

        btnLogin.setOnClickListener {
            login()
        }

        tvRedirectSignUp.setOnClickListener {
            val intent = Intent( packageContext: this, MainActivity::class.java)
            startActivity(intent)
        }
    }

    private fun login() {
        val email = etEmail.text.toString()
        val pass = etPass.text.toString()

        auth.signInWithEmailAndPassword(email, pass).addOnCompleteListener(this){
            if (it.isSuccessful) {
                Toast.makeText( context: this, text: "Вход успешен!", Toast.LENGTH_SHORT).show()
                intent = Intent( packageContext: this, HomeActivity::class.java)
                startActivity(intent)
            } else
                Toast.makeText( context: this, text: "Ошибка входа ", Toast.LENGTH_SHORT).show()
        }
    }
}

```

В build.gradle добавил следующее:

```

plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    alias(libs.plugins.google.gms.google.services)
}

```



```
dependencies {  
    implementation(libs.firebase.bom)  
    implementation(libs.firebase.auth)  
    implementation(libs.androidx.core.ktx)  
    implementation(libs.androidx.appcompat)  
    implementation(libs.material)  
    implementation(libs.androidx.activity)  
    implementation(libs.androidx.constraintlayout)  
    implementation(libs.firebase.auth.ktx)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.androidx.junit)  
    androidTestImplementation(libs.androidx.espresso.core)  
}
```