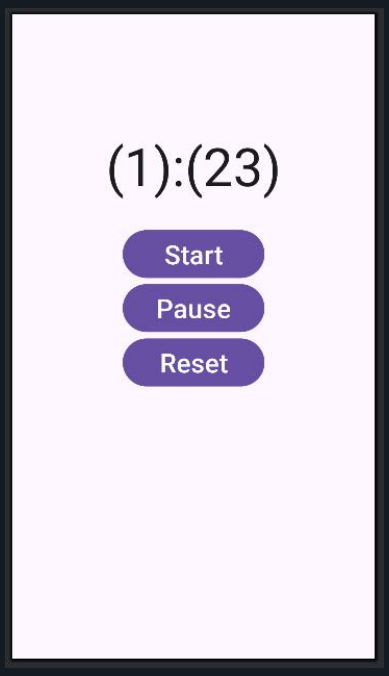
# 6. Практическая работа (Приложение секундомер)

- Создал макет приложения




Хронометр

```
<Chronometer
    android:id="@+id/chronometer1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="60sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.202" />
```

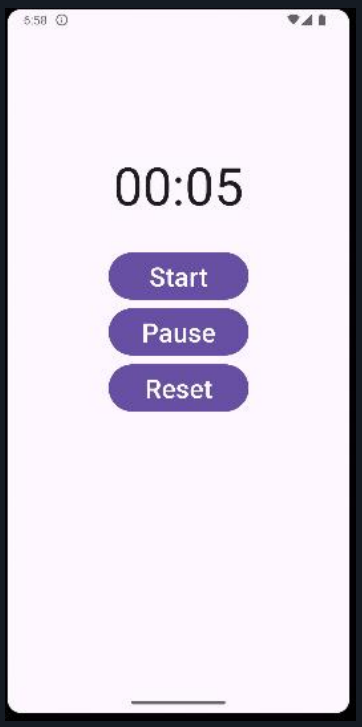
Кнопка

```
<Button
    android:id="@+id/btnStart"
    android:layout_width="161dp"
    android:layout_height="62dp"
    android:text="@string/start"
    android:textSize="30sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.36" />
```

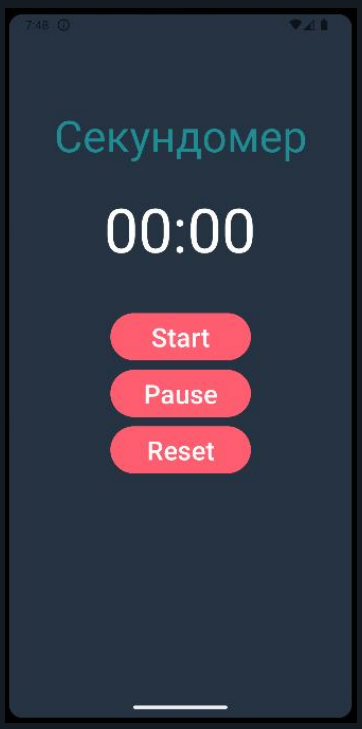
- Написал код для MainActivity
  - Теперь приложение работает.
    Start начинает и продолжает отсчёт
    Pause останавливает таймер
    Reset возвращает значение секундомера к 00:00

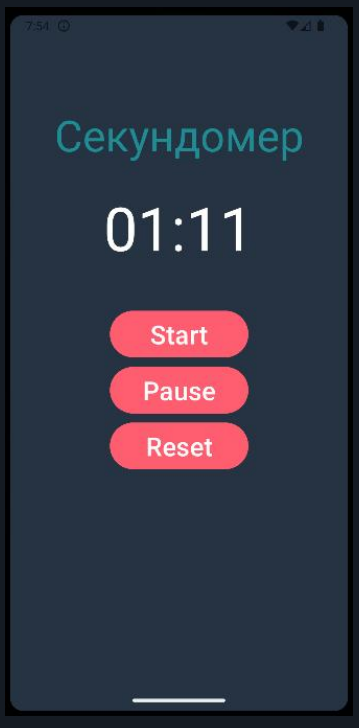
```
class MainActivity : AppCompatActivity() {
    lateinit var chronometr : Chronometer
    var running : Boolean = false
    var offset : Long = 0

    val OFFSET_KEY = "offset"
    val RUNNING_KEY = "running"
    val BASE_KEY = "base_key"

    override fun onSaveInstanceState(savedInstanceState: Bundle){
        savedInstanceState.putLong("offset", offset)
        savedInstanceState.putBoolean("offset", running)
        savedInstanceState.putLong("offset", chronometr.base)
        super.onSaveInstanceState(savedInstanceState)
    }

    fun setBaseTime(){
        chronometr.base =SystemClock.elapsedRealtime() - offset
    }

    @SuppressLint("MissingInflatedId")
    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
    }

    chronometr = findViewById(R.id.chronometer1)
    val btnStart = findViewById<Button>(R.id.btnStart)
    val btnPause = findViewById<Button>(R.id.btnPause)
    val btnReset = findViewById<Button>(R.id.btnReset)

    if(savedInstanceState != null){
        offset = savedInstanceState.getLong(OFFSET_KEY)
        running = savedInstanceState.getBoolean(RUNNING_KEY)
        if (running){
            chronometr.base = savedInstanceState.getLong(BASE_KEY)
            chronometr.start()
        } else setBaseTime()
    }

    fun saveOffset(){
        offset = SystemClock.elapsedRealtime() - chronometr.base
    }

    btnStart.setOnClickListener {
        if (!running){
            setBaseTime()
            chronometr.start()
            running = true
        }
    }

    btnPause.setOnClickListener {
            if (running){
                saveOffset()
                chronometr.stop()
                running = false
            }
    }

    btnReset.setOnClickListener {
        offset = 0
        setBaseTime()
        running = false
    }
}
```

▪ Работа приложения




▪ Конечный дизайн




```
<color name="off_black">#253343</color>
<color name="fiery_rose">#ff5d70</color>
<color name="lochinvar">#218c94</color>
```