Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

Факультет: информационных технологий и управления

Кафедра: систем автоматизированного проектирования и управления

Направление подготовки:09.03.03 Прикладная информатика в химии

Форма обучения: очная

Уровень подготовки: бакалавр

Учебная дисциплина: разработка программного обеспечения для мобильных систем

Группа: 405

**ОТЧЕТ**

**О ЛАБОРАТОРНОЙ РАБОТЕ№4**

ТЕМА: Клиент-серверная архитектура мобильного приложения для систем массового обслуживания и сложных вычислений.

Вариант № 10

Преподаватели Д.Н. Петров

К.А. Захаренко

Исполнители М. А. Шахов

К. И. Емельянова

Санкт-Петербург

2023

**Задание:**

Задание на лабораторную работу представлено в таблице 1.

Таблица 1 – Задание на лабораторную работу

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Входные данные | Результат |
| 10 | Вычисление времени нагрева жидкости, с от 24 °С до температуры кипения | Нагреваемая жидкость (5 шт. на выбор), объем нагреваемой жидкости, мощность нагревателя | Теплоемкость и температура кипения выбранной жидкости, время нагрева, с. до температуры кипения |

**Этапы выполнения**

1) Изучение функций и принципов взаимодействия элементов клиент-серверной архитектуры (прикладные протоколы, виды посылок, форматы данных). Основы клиент-серверного взаимодействия мобильного приложения и веб-сервера. Сервисно-ориентированный подход. Основы программирования на скриптовом серверном языке Php.

2) Разработка мобильного приложения:

- создание и настройка нового проекта приложения;

- разметка макета графического интерфейса главной активности (в соответствии с вариантом);

- создание управляющей логики для отправки HTTP-запроса на веб-сервер, получения и отображения ответа;

- создание веб-сервиса в виде файла \*.php для приема и обработки поступающих запросов и выдачи результата решения вычислительной задачи (в соответствии с вариантом);

- тестирование мобильного приложения.

**Описание выполнения лабораторной работы**

**1) Изучение функций и принципов взаимодействия элементов клиент-серверной архитектуры**

Предприятия в сфере массового обслуживания населения для нужд контроля, мониторинга, предоставления полезных инструментов и сервисов клиентам используют многоуровневые клиент-серверные приложения, в которых клиентский уровень (клиент) не выполняет сложные вычисления и обработки данных, а только посылает запрос на серверный уровень (сегмент). Сервер, в свою очередь, понимая по принятым данным из запроса, что клиент желает получить и перенаправляет клиента в нужный сервис. Вызванный сервис может использовать распределенный или кластерный серверный сегмент, чаще всего состоящий из большого количества связанных в одну сеть ЭВМ, расположенный нередко в разных странах и на разных континентах. Актуальным трендом в сфере IT и развития систем массового обслуживания (электронных сервисов) является применение микросервисной архитектуры на основе концепции REST API (Representational State Transfer Application Programming Interface). Используется протокол прикладного уровня передачи данных HTTP, исключительно защищенный, т.е. HTTPS, обеспечивающий целостность и конфиденциальность данных при их передаче между веб-сервисом и устройством пользователя. Как архитектурно и программно устроен серверный сегмент пользователь не знает, для него доступно только приложение-клиент и точка входа (API Gateway) для вызова доступных сервисов. В лабораторной работе точкой входа является веб-ссылка на файл index.php, принимающий, обрабатывающий HTTP-запросы (GET/POST/PUT), формирующий и отправляющий ответ на эти запросы (в виде HTML-кода или в формате JSON или XML). Здесь следует упомянуть MVC (Model-View-Controller) как наиболее популярный способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения. Выходит, что Controller принимает запрос, верифицирует его, защищает систему от возможных атак, блокирует попытки клиента нарушить работу сервиса, перенаправляет правильный запрос к сервисной функции (это уже Model), которая результат возвращает в Controller для его последующего перенаправления во View с целью рендеринга ответа. Controller выдает ответ клиенту на его запрос. Работа контроллера в этом смысле очень важная и ответственная.

**2) Разработка мобильного приложения**

**2.1 Создание и настройка нового проекта приложения**

В среде разработки мобильных приложений Android Studio Giraffe 17.0.6 для Android был создан проект lab4. На рисунках 1-4 представлен процесс создания проекта.

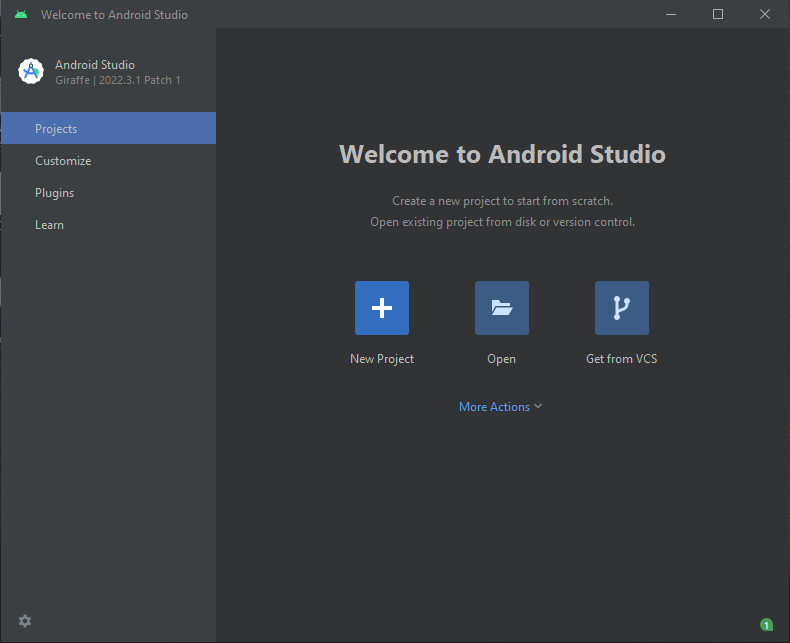


Рисунок 1 – Создание нового проекта

Создать новый проект также возможно из рабочего интерфейса по команде File->New->New Project (рисунок 2)

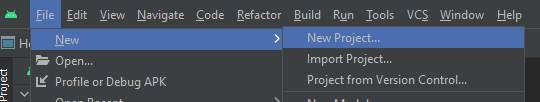


Рисунок 2 – Создание нового проекта

На появившейся форме выбран тип целевого мобильного устройства (Phone and Tablet) и шаблон приложения (Empty Activity до версии Android Studio Giraffe, а начиная с Giraffe, Empty Views Activity, т.к. требуется поддержка языка Java) (рисунок 3).

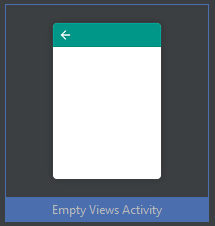
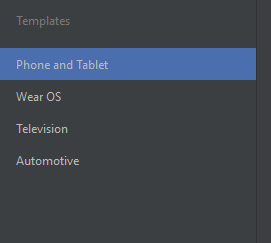


Рисунок 3 – Выбор типа целевого устройства и шаблона приложения

Далее указано наименование приложения, пакета, выбрана директория для сохранения файлов проекта и выбрана минимальную версию SDK, от которой зависит поддержка приложения операционными системами (рисунок 4).

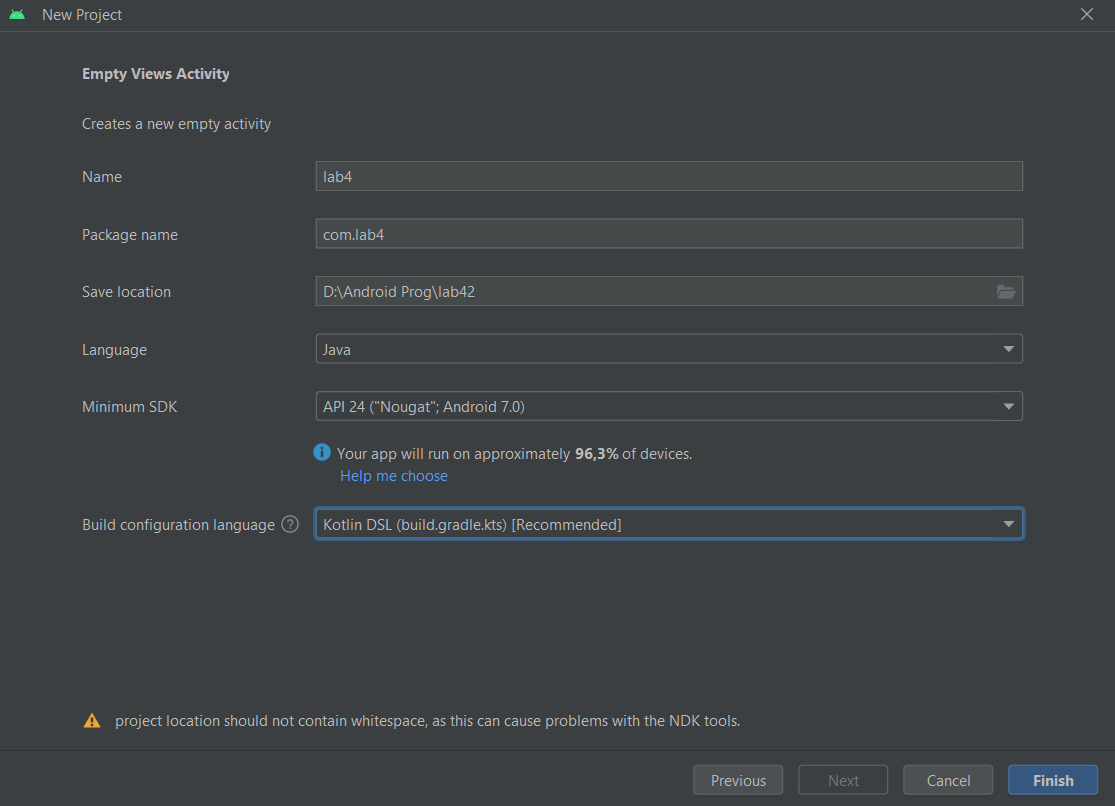


Рисунок 4 – Настройка нового приложения

Минимальная версия SDK – 24, минимальная версия ОС Android – 7.

При модификации файла манифеста приложения было включено разрешение на доступ к сети Интернет:

<uses-permission android:name="android.permission.INTERNET" />

**2.2 Разметка макета графического интерфейса главной активности**

**Структура проекта**

В таблице 2 представлена структура мобильного приложения.

Таблица 2 – Структура проекта мобильного приложения

|  |  |  |
| --- | --- | --- |
| Каталог | Файлы | Описание |
| manifests | AndroidManifest.xml | Основная информация о программе |
| java/com.lab4 | MainActivity.java | Управляющий код приложения |
| res/drawable | info.xml | Логотип кнопки с информацией о проекте |
| foreground.xml | Логотип приложения |
| res/layout | activity\_main.xml | Главная активность приложения |
| res/values | colors.xml | Используемые цвета в приложении |
| strings.xml | Строковые константы приложения |
| res/values/themes | themes.xml | Файл светлой темы приложения |
| themes.xml (night) | Файл темной темы приложения |
| res/xml | backup\_rules.xml | Настройка резервной копии приложения |
| data\_extraction\_rules.xml | Настройка извлечения данных |

При разработке приложения для запуска и отладки было использовано физическое устройство – мобильный телефон Xiaomi Redmi Note 9 Pro. Минимальная версия OC Android – 7, целевая – 13. Минимальная версия SDK – 24, целевая – 31.

Листинг файла манифеста проекта (файл AndroidManifest.xml):

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.INTERNET" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@drawable/icon"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>

Листинг файла строковых констант strings.xml:

<resources>  
 <string name="URL">https://worldlab.technolog.edu.ru/stud/group10/index2.php?</string>  
 <string name="app\_name">Лабораторная работа №4</string>  
 <string name="dialog\_title">Ошибка</string>  
 <string name="dialog\_message">Неверный формат данных</string>  
 <string name="water">Вода</string>  
 <string name="ethyl">Этиловый спирт</string>  
 <string name="oil">Нефть</string>  
 <string name="glycerol">Глицерин</string>  
 <string name="heatCapacity">Теплоемкость</string>  
 <string name="boilingTemperature">Температура кипения</string>  
 <string name="density">Плотность</string>  
 <string name="heatingTime">Время нагрева</string>  
 <string name="result">Результат</string>  
 <string name="benzene">Бензол</string>  
 <string name="Error\_Title">Ошибка</string>  
 <string name="Error\_Empty">Введите значения</string>  
 <string name="Error\_Empty\_Volume">Введите значение объема</string>  
 <string name="Error\_Empty\_Wattage">Введите значение мощности нагревательного элемента</string>  
 <string name="Label\_Volume">Объем жидкости, л</string>  
 <string name="Label\_HeaterWattage">Мощность\nнагревательного элемента, Вт</string>  
 <string name="Label\_ComputeTimeToHeat">Расчет времени нагрева жидкости</string>  
 <string name="Label\_ChooseLiquid">Выберите жидкость</string>  
 <string name="Button\_Calculate">Рассчитать</string>  
 <string name="TextInput\_Volume">Л</string>  
 <string name="TextInput\_PowerWattage">Вт</string>  
 <string name="Unit\_J\_K\_KG">Дж/(К\*кг)</string>  
 <string name="Unit\_0C">°C</string>  
 <string name="Unit\_S">с</string>  
 <string name="Result">Результат:</string>  
 <string-array name="liquids\_array">  
 <item>вода</item>  
 <item>этил</item>  
 <item>нефть</item>  
 <item>глицерин</item>  
 <item>бензин</item>  
 </string-array>  
  
</resources>

Листинг файла цветовых констант colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="textColorPrimary">#000000</color> <!-- Цвет для светлой темы -->  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="purple">#9334e6</color>  
 <color name="creamy">#F7E7CE</color>  
 <color name="grey">#1a1a1a</color>  
</resources>

Листинг файла цветовых констант colors.xml (night):

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="textColorPrimary">#FFFFFF</color> <!-- Цвет для темной темы -->  
</resources>

Используемые изображения:

Логотипы приложения и оформления кнопок размещены в res/drawable (таблица 3)

Таблица 3 – Логотипы для оформления кнопок в приложении

|  |  |
| --- | --- |
| Код разметки | Графический вид |
| info.xml | |
| <vector android:height="24dp" android:tint="#FFFFFF"  android:viewportHeight="24" android:viewportWidth="24"  android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  <path android:fillColor="@android:color/white" android:pathData="M12,2C6.48,2 2,6.48 2,12s4.48,10 10,10 10,-4.48 10,-10S17.52,2 12,2zM13,17h-2v-6h2v6zM13,9h-2L11,7h2v2z"/></vector> |  |
| foreground.xml | |
| <vector android:height="48dp" android:tint="@color/purple" android:viewportHeight="24" android:viewportWidth="24" android:width="48dp" xmlns:android="http://schemas.android.com/apk/res/android"> <path  android:fillColor="@color/purple"  android:pathData="M15,13L15,5c0,-1.66 -1.34,-3 -3,-3S9,3.34 9,5v8c-1.21,0.91 -2,2.37 -2,4 0,2.76 2.24,5 5,5s5,-2.24 5,-5c0,-1.63 -0.79,-3.09 -2,-4zM11,5c0,-0.55 0.45,-1 1,-1s1,0.45 1,1h-1v1h1v2h-1v1h1v2h-2L11,5z"/> </vector> |  |

Темы оформления:

Листинг файла светлой темы (файл themes.xml):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <!-- Base application theme. -->  
 <style name="Base.Theme.MyApplication" parent="Theme.Material3.DayNight.NoActionBar">  
 <!-- Customize your light theme here. -->  
 </style>  
 <style name="Theme.MyApplication" parent="Base.Theme.MyApplication" />  
</resources>

Листинг файла темной темы (themes.xml(night)):

<resources xmlns:tools="http://schemas.android.com/tools">  
 <!-- Base application theme. -->  
 <style name="Base.Theme.MyApplication" parent="Theme.Material3.DayNight.NoActionBar">  
 <!-- Customize your dark theme here. -->  
 <item name="android:background">@color/grey</item>  
 <item name="android:textColor">@color/textColorPrimary</item>  
 </style>  
</resources>

**Разметка макетов графических интерфейсов**

На рисунке 5 представлена разработанная структура макета активности приложения в режиме дизайна.

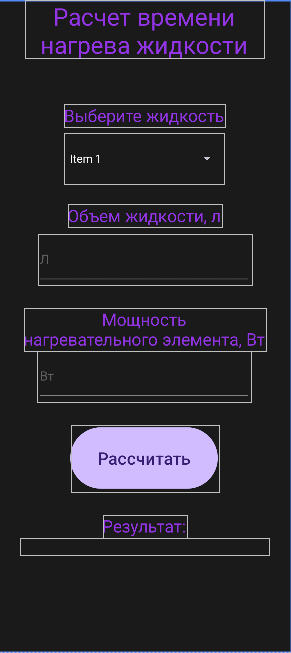


Рисунок 5 – Структура макета activity\_main

На макет графического интерфейса были добавлены следующие графические компоненты:

* scrollView – вертикальная полоса прокрутки (scrollView);
* ConstraintLayout – представляет контейнер, который позволяет создавать гибкие и масштабируемые визуальные интерфейсы (ConstraintLayout);
* textView – предназначен для отображения текста без возможности редактирования его пользователем (textViewHeader, textViewLqdTtl, textViewVTtl, textViewPWTtl, textViewRsltTtl, textViewRslt);
* TextInput – предназначен для ввода текста пользователем (vTextInput, pWTextInput);
* Spinner – выпадающий список, позволяющий выбрать одно значение (spinnerLqdLst);
* button – кнопка (calcBtn).

Листинг файла разметки макета экрана главной активности (файл activity\_main.xml):

<?xml version="1.0" encoding="utf-8"?>  
  
<ScrollView 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:fillViewport="true"  
 tools:context=".MainActivity">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 tools:context=".MainActivity"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:layout\_editor\_absoluteY="-116dp">  
  
  
 <TextView  
 android:id="@+id/textViewHeader"  
 android:layout\_width="335dp"  
 android:layout\_height="82dp"  
 android:text="@string/Label\_ComputeTimeToHeat"  
 android:textAlignment="center"  
 android:textColor="@color/purple"  
 android:textSize="34sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/textViewLqdTtl"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.04"  
 tools:ignore="TextSizeCheck" />  
  
 <TextView  
 android:id="@+id/textViewLqdTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="64dp"  
 android:text="@string/Label\_ChooseLiquid"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/spinnerLqdLst"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewHeader" />  
  
 <Spinner  
 android:id="@+id/spinnerLqdLst"  
 android:layout\_width="225dp"  
 android:layout\_height="72dp"  
 android:layout\_marginTop="8dp"  
 android:entries="@android:array/emailAddressTypes"  
 android:textColor="@color/creamy"  
 app:layout\_constraintBottom\_toTopOf="@+id/textViewVTtl"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewLqdTtl" />  
  
 <TextView  
 android:id="@+id/textViewVTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:text="@string/Label\_Volume"  
 android:textAlignment="center"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/vTextInput"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/spinnerLqdLst" />  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/vTextInput"  
 android:layout\_width="300dp"  
 android:layout\_height="72dp"  
 android:layout\_marginTop="9dp"  
 android:hint="@string/TextInput\_Volume"  
 android:inputType="number|numberDecimal|numberSigned"  
 android:maxLines="1"  
 android:minHeight="48dp"  
 android:textColorHint="#616161"  
 app:layout\_constraintBottom\_toTopOf="@+id/textViewPWTtl"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewVTtl" />  
  
 <TextView  
 android:id="@+id/textViewPWTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="@string/Label\_HeaterWattage"  
 android:textAlignment="center"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/pWTextInput"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/vTextInput" />  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/pWTextInput"  
 android:layout\_width="300dp"  
 android:layout\_height="72dp"  
 android:layout\_marginStart="16dp"  
 android:hint="@string/TextInput\_PowerWattage"  
 android:inputType="number|numberDecimal|numberSigned"  
 android:maxLines="1"  
 android:minHeight="48dp"  
 android:textColorHint="#616161"  
 app:layout\_constraintBottom\_toTopOf="@+id/calcBtn"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.41"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewPWTtl" />  
  
 <Button  
 android:id="@+id/calcBtn"  
 android:layout\_width="208dp"  
 android:layout\_height="94dp"  
 android:layout\_marginTop="32dp"  
 android:text="@string/Button\_Calculate"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/textViewRsltTtl"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/pWTextInput" />  
  
 <TextView  
 android:id="@+id/textViewRsltTtl"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="@string/Result"  
 android:textAlignment="textStart"  
 android:textColor="@color/purple"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/textViewRslt"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/calcBtn" />  
  
 <TextView  
 android:id="@+id/textViewRslt"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="viewStart"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewRsltTtl" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</ScrollView>

**2.3 Создание управляющей логики**

Листинг кода MainActivity.java:

package com.example.myapplication;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import org.json.JSONObject;  
  
import java.io.BufferedReader;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.net.URLEncoder;  
import java.text.DecimalFormat;  
import java.util.Locale;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Spinner spinnerLiquid;  
 private EditText editTextVolume, editTextWattage;  
 private TextView textViewResult;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 spinnerLiquid = findViewById(R.id.*spinnerLqdLst*);  
 editTextVolume = findViewById(R.id.*vTextInput*);  
 editTextWattage = findViewById(R.id.*pWTextInput*);  
 Button buttonCalculate = findViewById(R.id.*calcBtn*);  
 textViewResult = findViewById(R.id.*textViewRslt*);  
  
 // Настройка Spinner  
 ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*liquids\_array*, android.R.layout.*simple\_spinner\_item*);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spinnerLiquid.setAdapter(adapter);  
  
 buttonCalculate.setOnClickListener(v -> {  
 String liquid = spinnerLiquid.getSelectedItem().toString();  
 double volume = parseDoubleOrNull(editTextVolume.getText().toString());  
 double wattage = parseDoubleOrNull(editTextWattage.getText().toString());  
 if (volume == 0 && wattage == 0) {  
 runOnUiThread(() -> textViewResult.setText(R.string.*Error\_Empty*));  
 }  
 else if(volume == 0) {  
 runOnUiThread(() -> textViewResult.setText(R.string.*Error\_Empty\_Volume*));  
 } else if (wattage == 0) {  
 textViewResult.setText(R.string.*Error\_Empty\_Wattage*);  
 }  
 else {  
 calculateBoilingTime(liquid, volume, wattage);  
 }  
 });  
 }  
  
 private void calculateBoilingTime(String liquid, double volume, double heaterWattage) {  
 new Thread(() -> {  
 try {  
 String query = String.*format*(Locale.*getDefault*(), "liquid=%s&volume=%f&heaterWattage=%f",  
 URLEncoder.*encode*(liquid, "UTF-8"), volume, heaterWattage);  
 URL url = new URL(getString(R.string.*URL*) + query);  
 HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
 connection.setRequestMethod("GET");  
  
 InputStream stream = connection.getInputStream();  
 BufferedReader reader = new BufferedReader(new InputStreamReader(stream));  
 StringBuilder response = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 response.append(line);  
 }  
  
 JSONObject jsonResponse = new JSONObject(response.toString());  
 double timeToBoil = jsonResponse.getDouble("timeToBoil");  
 double heatCapacity = jsonResponse.getDouble("heatCapacity");  
 double boilingTemperature = jsonResponse.getDouble("boilingTemperature");  
  
 String resultMsg = getString(R.string.*heatCapacity*) + ", " + getString(R.string.*Unit\_J\_K\_KG*) + ": " +  
 new DecimalFormat("0.00").format(heatCapacity) + "\n" +  
 getString(R.string.*boilingTemperature*) + ", " + getString(R.string.*Unit\_0C*) + ": " +  
 new DecimalFormat("0.00").format(boilingTemperature) + "\n" +  
 getString(R.string.*heatingTime*) + ", " + getString(R.string.*Unit\_S*) + ": " +  
 new DecimalFormat("0.00").format(timeToBoil);  
  
 runOnUiThread(() -> textViewResult.setText(resultMsg));  
 } catch (Exception e) {  
 e.printStackTrace();  
 runOnUiThread(() -> textViewResult.setText(R.string.*Error\_Empty\_Wattage*));  
 }  
 }).start();  
 }  
  
 public double parseDoubleOrNull(String value) {  
 try {  
 return Double.*parseDouble*(value);  
 } catch (NumberFormatException e) {  
 return 0;  
 }  
 }  
}

**2.4 Создание веб-сервиса**

В ходе выполнения лабораторной работы был использован веб-сервис https://worldlab.technolog.edu.ru/.

В выделенный каталог веб-сервера был размещен файл index2.php, содержащий код, часть которого отвечает за прием и верификацию запроса, часть кода выполняет главные вычислительные операции, и код, «упаковывающий» результат вычислений в нужный выходной формат с отправкой результата клиенту.

Листинг кода index.php:

<?php

header('Content-Type: application/json');

// Принимаем параметры из запроса

$liquidName = $\_GET['liquid'];

$volume = $\_GET['volume'];

$heaterWattage = $\_GET['heaterWattage'];

// Определяем свойства жидкостей

$liquids = [

"вода" => ["heatCapacity" => 4186, "boilingTemperature" => 100, "density" => 1],

"этил" => ["heatCapacity" => 2470, "boilingTemperature" => 78.37, "density" => 0.789],

"нефть" => ["heatCapacity" => 2100, "boilingTemperature" => 140, "density" => 0.850],

"глицерин" => ["heatCapacity" => 2430, "boilingTemperature" => 290, "density" => 1.252],

"бензин" => ["heatCapacity" => 1050, "boilingTemperature" => 80.1, "density" => 0.876]

];

// Выбираем жидкость

if(array\_key\_exists($liquidName, $liquids)) {

$liquid = $liquids[$liquidName];

$startTemp = 24; // Начальная температура в °C

// Вычисляем время до кипения

$timeToBoil = ($liquid["heatCapacity"] \* $volume \* $liquid["density"] \* ($liquid["boilingTemperature"] - $startTemp)) / ($heaterWattage); // В секундах

// Формируем и отправляем ответ

$response = [

"timeToBoil" => $timeToBoil,

"heatCapacity" => $liquid["heatCapacity"],

"boilingTemperature" => $liquid["boilingTemperature"]

];

echo json\_encode($response);

} else {

// Жидкость не найдена

http\_response\_code(404);

echo json\_encode(["error" => "Liquid not found"]);

}

?>

**2.5 Тестирование мобильного приложения**

Тестирование выполнялось на физическом устройстве (мобильный телефон Xiaomi Redmi Note 9 Pro).

Результат работы мобильного приложения представлен на рисунках 6-13.

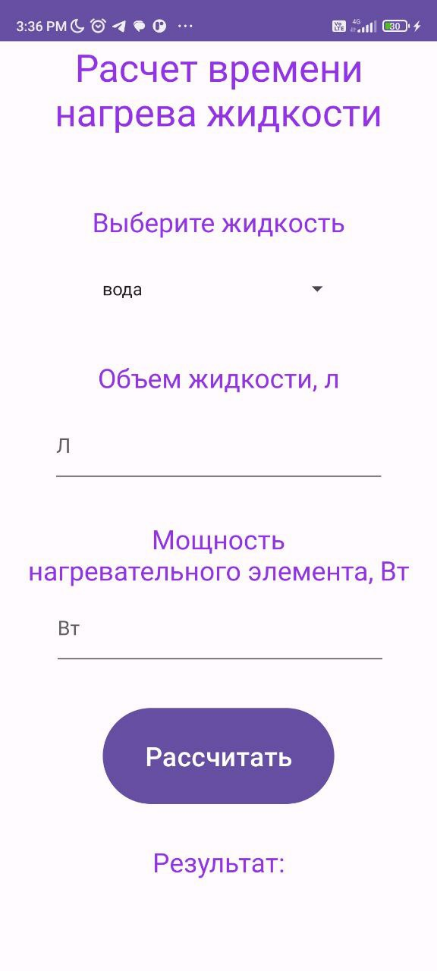


Рисунок 6 – Результат работы главной активности мобильного приложения на светлой теме, портретная ориентация

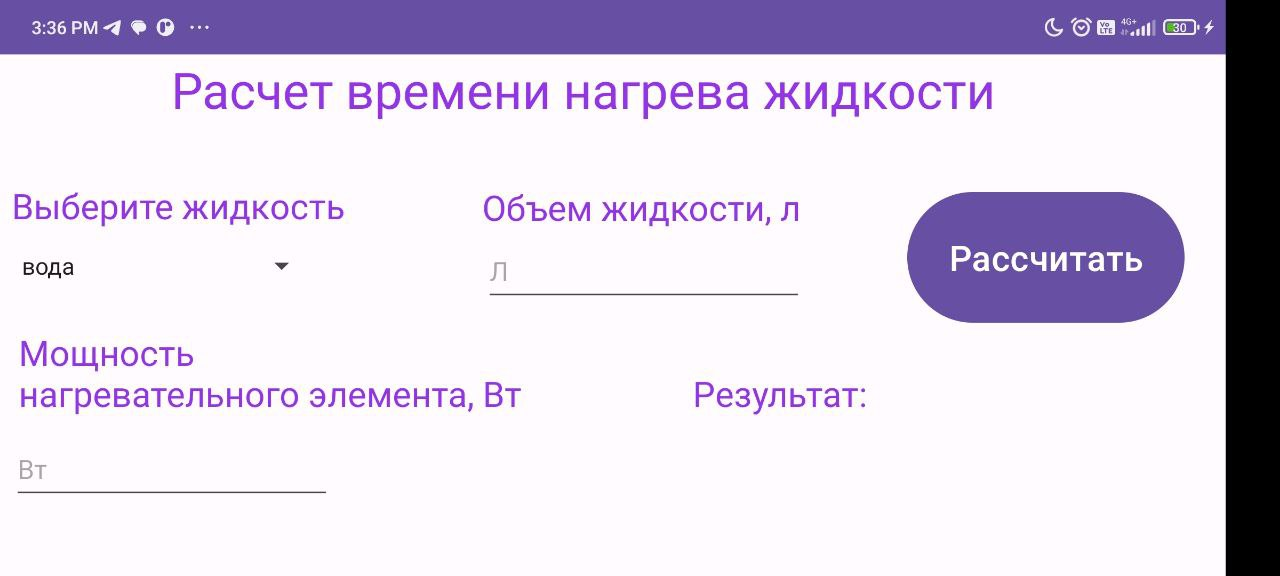


Рисунок 7 – Результат работы главной активности мобильного приложения на светлой теме, альбомная ориентация

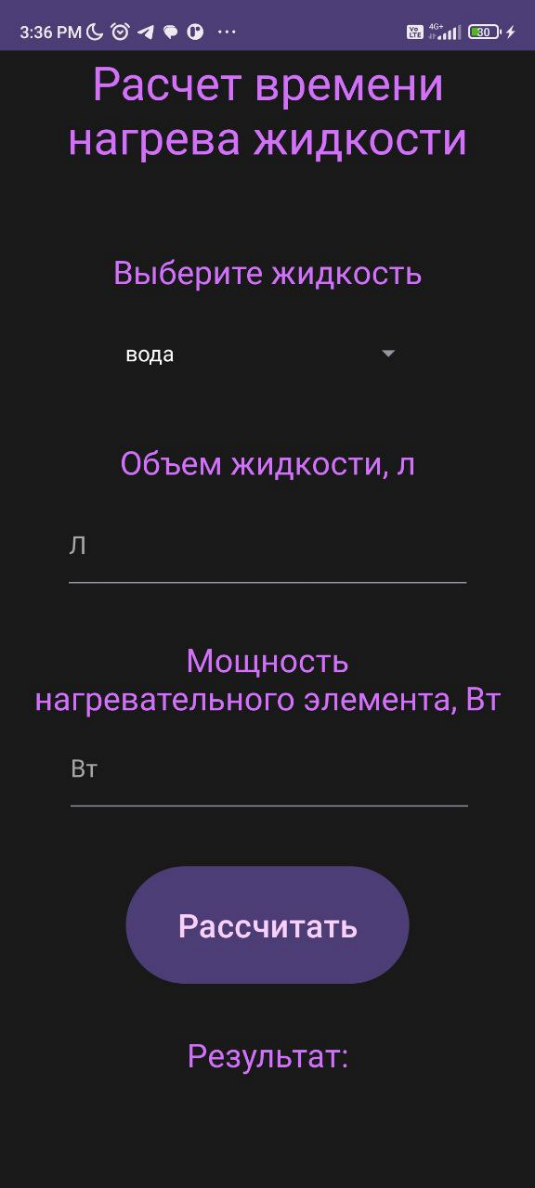


Рисунок 8 – Результат работы главной активности мобильного приложения на темной теме, портретная ориентация

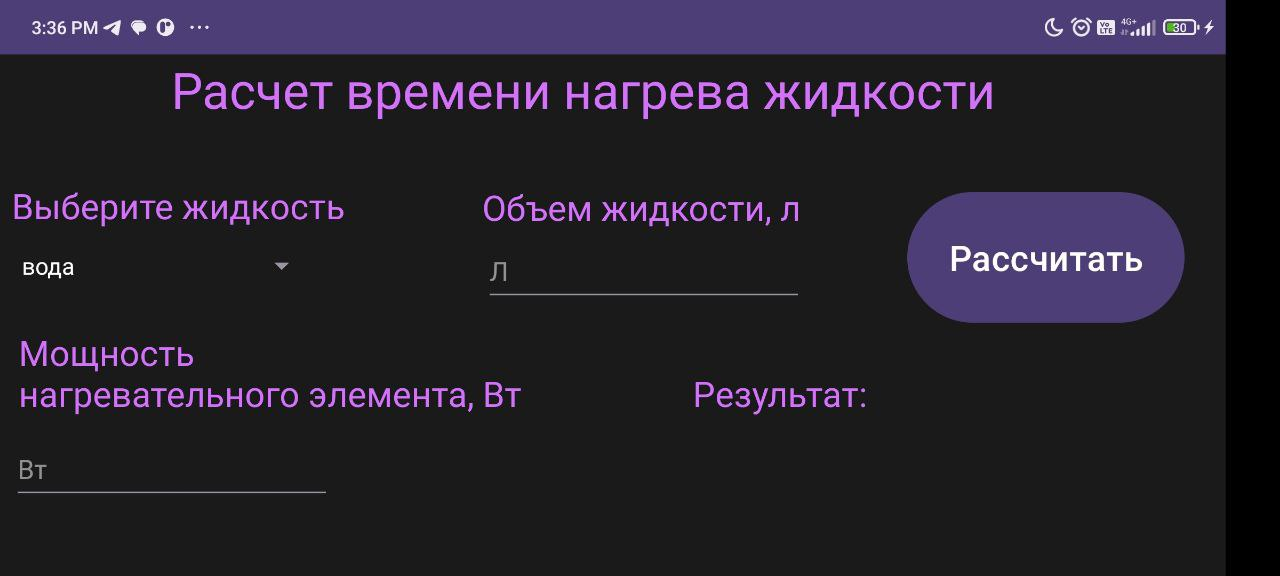


Рисунок 9 – Результат работы главной активности мобильного приложения на темной теме, альбомная ориентация

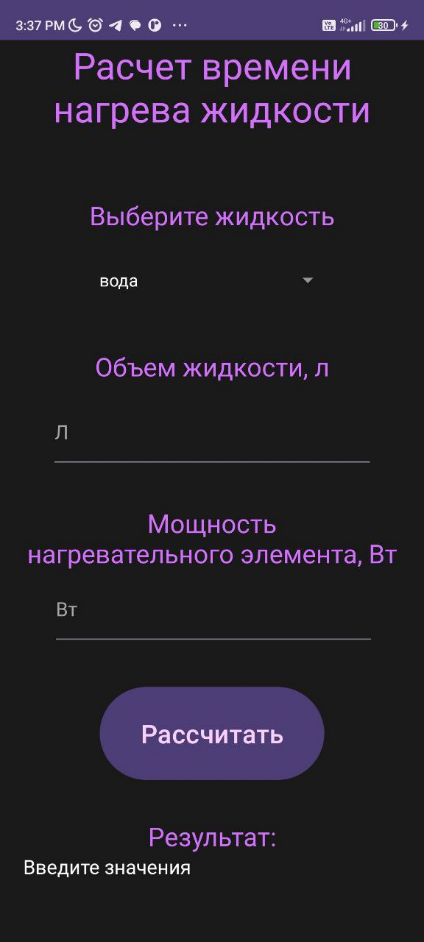


Рисунок 10 – Результат работы главной активности мобильного приложения с ошибкой ввода

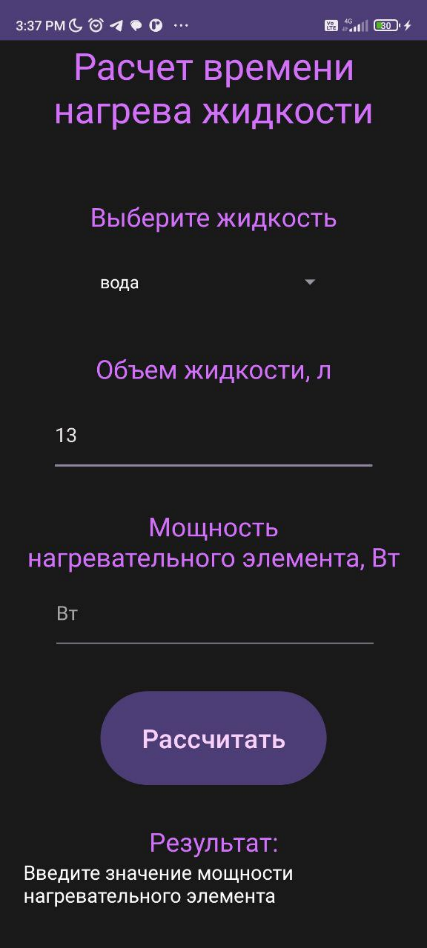


Рисунок 11 – Результат работы главной активности мобильного приложения с ошибкой ввода

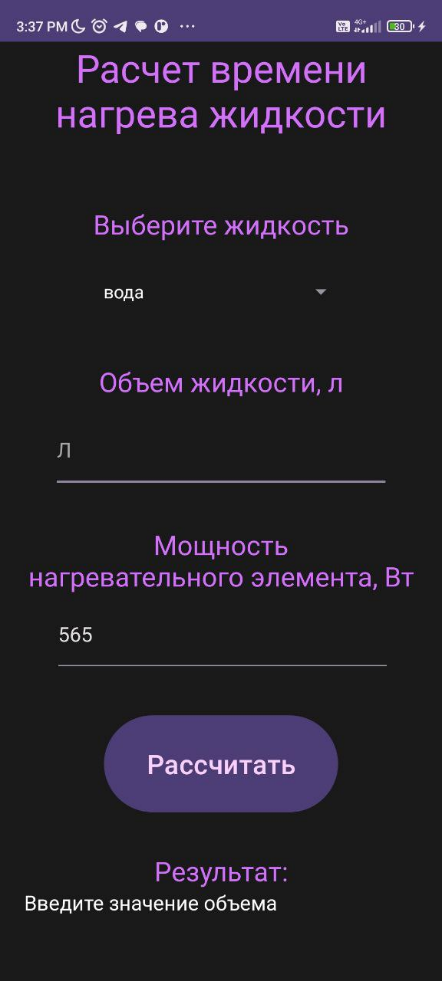


Рисунок 12 – Результат работы главной активности мобильного приложения с ошибкой ввода

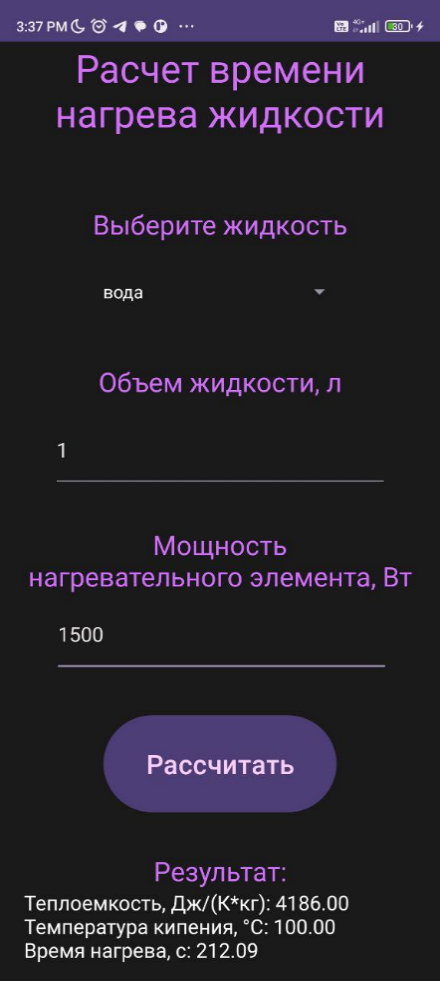


Рисунок 13 – Результат работы главной активности мобильного приложения

**Выводы**

В ходе выполнения лабораторной работы было разработано мобильное приложения для Android с клиент-серверной архитектурой для систем массового обслуживания. Для достижения поставленной цели были выполнены следующие задачи:

1) Изучены функции и принципы взаимодействия элементов клиент-серверной архитектуры (прикладные протоколы, виды посылок, форматы данных). Основы клиент-серверного взаимодействия мобильного приложения и веб-сервера. Сервисно-ориентированный подход. Основы программирования на скриптовом серверном языке Php.

2) Разработано мобильное приложение для Android в среде Android Studio Giraffe 2022.3.1:

- создан и настроен новый проект приложения;

- указано наименование приложения и выбраны версии SDK: минимальная – 24, целевая – 31, а также версии ОС Android: минимальная – 7, целевая – 13;

- выбран язык программирования – Java.

- создана разметка макета графического интерфейса;

- создана управляющая логика для отправки HTTP-запроса на веб-сервер, получения и отображения ответа;

- создан веб-сервис в виде файла index2.php для приема и обработки поступающих запросов и выдачи результата решения вычислительной задачи (расчета константы скорости реакции);

- проведено тестирование мобильного приложения. Тестирование показало работоспособность мобильного приложения в портретной и альбомной ориентации, а также с ветлой и темной теме.

Характеристики аппаратного обеспечения разработчика мобильного приложения приведены в таблице 4.

Таблица 4 – Характеристики аппаратного обеспечения

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Ноутбук |
| Тип и тактовая частота (ν) процессора | Intel Core i7, ν = 2,71 ГГц |
| Состав и технические характеристики периферийных устройств ЭВМ | Мобильный телефон Xiaomi Redmi Note 9 Pro;  Кабель USB - TYPE-C |
| Операционная система ЭВМ | Windows 10 |
| Версия Android Studio | Android Studio Giraffe 2022.3.1 |