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№5**

ТЕМА: Гибридные технологии разработки нетиповых клиент-серверных мобильных систем.

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 принципов слияния технологий разработки нетиповых мобильных приложений, обоснование применения веб-технологий к разработке мобильных приложений. Описание назначения и функционала виджета WebView. Преимущества и недостатки использования WebView в мобильном приложении. Изучение основ JavaScript, технологии AJAX, обзор JS-библиотек, поддерживающих асинхронные HTTP-запросы. Изучение каскадных стилей CSS для оформления веб-страницы и динамики поведения ее DOM-объектов как реакций на события (смена расширения экрана, действия пользователя).

2) Разработка мобильного приложения:

- создание и настройка нового проекта приложения;

- разметка макета графического интерфейса главной активности;

- настройка виджета WebView, создание кода настройки и загрузки веб-страницы;

- создание и оформление веб-страницы;

- создание клиентского управляющего JS-кода;

- создание серверного скрипта;

- тестирование мобильного приложения.

**Описание выполнения лабораторной работы**

**1) Изучение принципов слияния технологий разработки нетиповых мобильных приложений, обоснование применения веб-технологий к разработке мобильных приложений**

Существует два варианта технической реализации приложений для мобильных устройств: мобильный веб-сайт и мобильное приложение. Мобильным веб-сайтом считается сайт, адаптирующийся для просмотра и функционирования на мобильном устройстве. Сайт может включать интерактивные компоненты с использованием JavaScript, HTML5, новых API браузеров. Мобильное приложение – это приложение, разработанное под конкретную мобильную платформу (iOS, Android, Windows Phone). Обычно мобильное приложение разрабатывается на языке высокого уровня (например, Java) и компилируется в нативный код ОС, дающий максимальную производительность.

Существует еще третий вариант – гибридное мобильное приложение, включающее виджет браузера (например, веб-компонент WebView) и другие требуемые виджеты. В этом случае часть мобильного приложения используется для навигации и интергации с ОС, а веб-компонент – для показа веб-контента. Обычные пользователи не могут зачастую отличить такой вариант от нативного мобильного приложения.

В таблице 2 приведены основные преимущества и недостатки использования нативных мобильных приложений (НМП), мобильных веб-сайтов (МВС) и гибридных мобильных приложений (ГМП).

Таблица 2 – Преимущества и недостатки вариантов технической реализации мобильных приложений

|  |  |  |  |
| --- | --- | --- | --- |
|  | НМП | МВС | ГМП |
| Преиму-щества | 1) Тесная связь с операционной системой и аппаратным обеспечением мобильного устройства.  2) Высокое быстродействие. | 1) Безграничные функциональные возможности при использовании современных технологий JS, HTML5, CSS3.  2) Снижение затрат на проект с привлечением веб-разработчиков.  3) Кроссплатформенность «из коробки». | 1) Связь с операционной системой и аппаратным обеспечением мобильного устройства.  2) Высокое быстродействие.  3) Возможность использования технологий JS, HTML5, CSS3.  4) При выходе новой версии МП, не затрагивающей нативный код, не требуется загрузка обновления на мобильные устройства.  5) Относительно невысокие затраты на разработку проекта.  6) Возможность «встраивания» контента на другие веб-ресурсы в виде отдельных компонентов. |
| Недоста-тки | 1) Требуются высокооплачиваемые профессиональные разработчики ПО для мобильных устройств.  2) При выходе новой версии МП необходима загрузка обновления на устройства клиентов.  3) Требуется разработка (иногда «с нуля») и тестирование МП под каждую мобильную ОС.  4) Возникновение «особых» условий лицензирования и распространения при использовании некоторых виджетов. | 1) Нет связи с операционной системой и аппаратным обеспечением мобильного устройства.  2) Невысокое быстродействие интерактивного сайта на устаревших мобильных веб-браузерах и устройствах.  3) Трудности в адаптации МВС к работе на различных мобильных веб-браузверах. | 1) Требуется портирование МП под каждую мобильную ОС.  2) Снижение быстродействия при высокой нагрузке на виджет веб-браузера для сложного веб-контента. |

Как видно из таблицы 2, при использовании гибридных технологий разработки мобильных приложений, зачастую получаем высокий технико-экономический эффект. Однако, для принятия решения об использовании технологии разработки МП, со стороны предприятия-разработчика требуется анализ нескольких критериев, например, финансовое состояние компании и объем резерва средств под проект, опыт и квалификация программистов, требования к мобильному приложению (функциональные, аппаратные и программные), целевая аудитория и т.д.

**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 lab5. На рисунках 1-4 представлен процесс создания проекта.

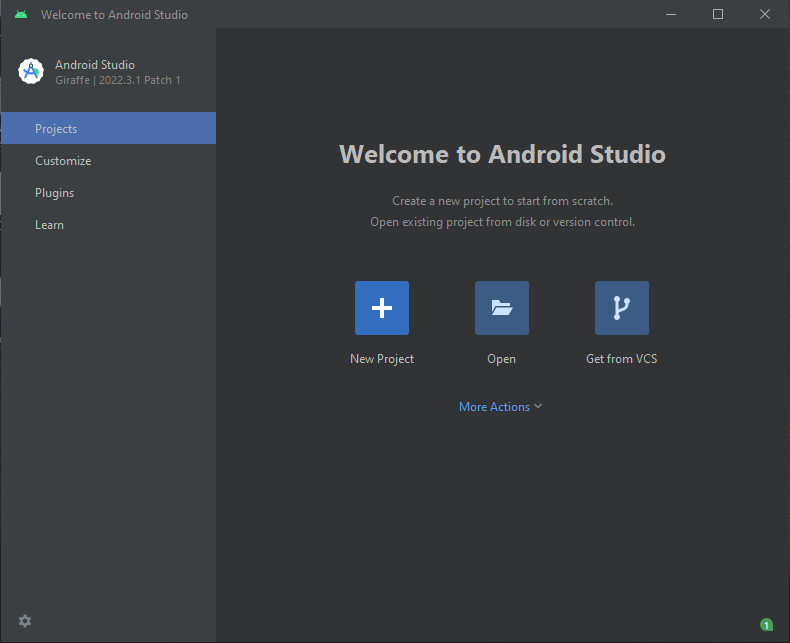


Рисунок 1 – Создание нового проекта

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

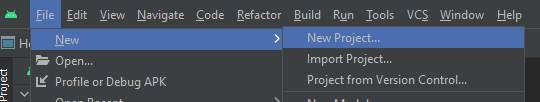


Рисунок 2 – Создание нового проекта

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).

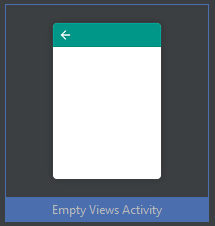
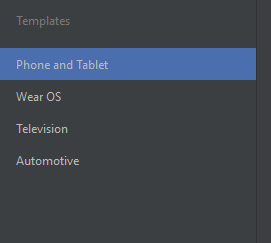


Рисунок 3 – Выбор типа целевого устройства и шаблона приложения

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).

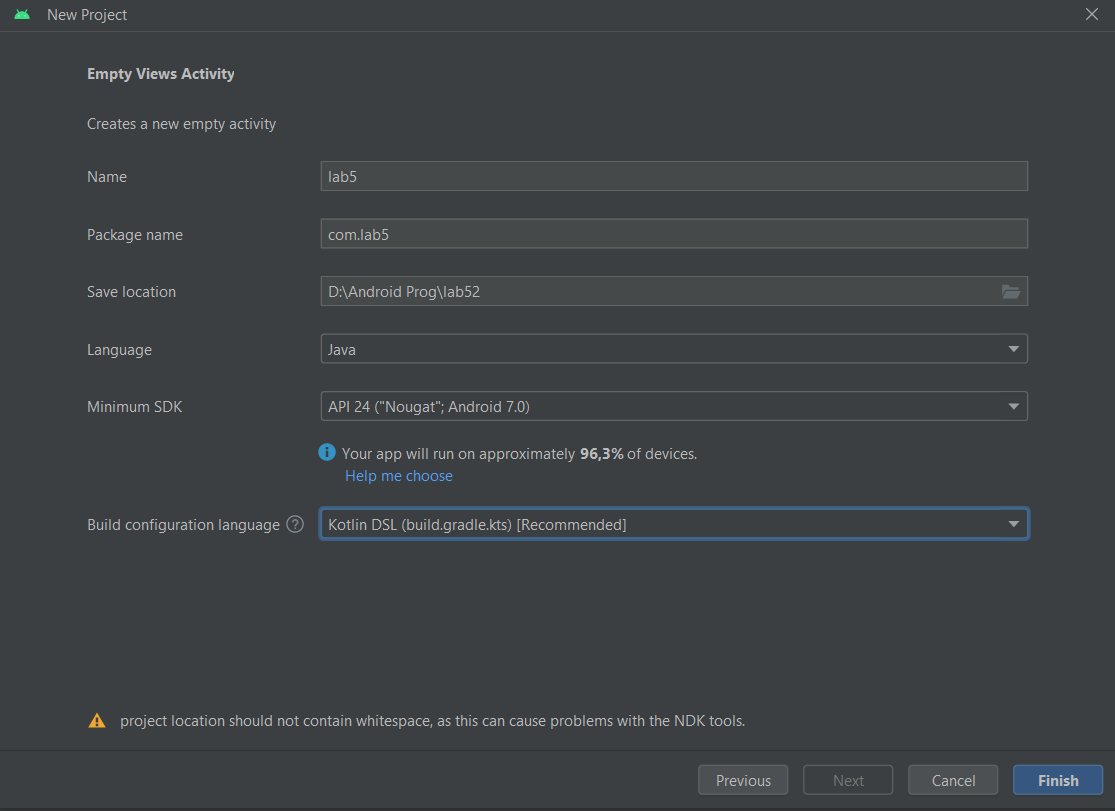


Рисунок 4 – Настройка нового приложения

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 3 представлена структура мобильного приложения.

Таблица 3 – Структура проекта мобильного приложения

|  |  |  |
| --- | --- | --- |
| Каталог | Файлы | Описание |
| manifests | AndroidManifest.xml | Основная информация о программе |
| java/com.lab5 | MainActivity.java | Управляющий код встраивания веб-страницы |
| res/drawable | baseline\_info.xml | Логотип кнопки с информацией о проекте |
| foreground.xml | Логотип приложения |
| res/layout | activity\_main.xml | Главная активность приложения с веб-страницей |
| 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" />  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
  
 <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.QuadEq2"  
 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>  
  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 </application>  
</manifest>

Листинг файла цветовых констант colors.xml:

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="purple\_500">#7737D3</color>  
 <color name="purple\_700">#7737D1</color>  
 <color name="purple\_200">#7737D3</color>  
 <color name="purple\_701">#7737D3</color>  
 <color name="purple">#7737D3</color>  
</resources>

Используемые изображения:

Логотипы приложения и оформления кнопок размещены в res/drawable (таблица 4)

Таблица 4 – Логотипы для оформления кнопок в приложении

|  |  |
| --- | --- |
| Код разметки | Графический вид |
| baseline\_info.xml | |
| <vector android:height="30dp" android:tint="#FFFFFF"  android:viewportHeight="24" android:viewportWidth="24"  android:width="30dp" 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> |  |
| icon.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="Theme.QuadEq2" parent="Theme.MaterialComponents.DayNight.NoActionBar">  
 <!-- Primary brand color. -->  
 <item name="colorPrimary">@color/purple\_500</item>  
 <item name="colorPrimaryVariant">@color/purple\_700</item>  
 <item name="colorOnPrimary">@color/white</item>  
 <!-- Secondary brand color. -->  
 <item name="colorSecondary">@color/purple\_200</item>  
 <item name="colorSecondaryVariant">@color/purple\_701</item>  
 <item name="colorOnSecondary">@color/black</item>  
 <!-- Status bar color. -->  
 <item name="android:statusBarColor">?attr/colorPrimaryVariant</item>  
 <!-- Customize your theme here. -->  
 </style>  
</resources>

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

<resources xmlns:tools="http://schemas.android.com/tools">  
 <!-- Base application theme. -->  
 <style name="Base.Theme.Lab5" parent="Theme.Material3.DayNight.NoActionBar">  
 <!-- Customize your dark theme here. -->  
 <!-- <item name="colorPrimary">@color/my\_dark\_primary</item> -->  
 </style>  
</resources>

**Разметка макетов графических интерфейсов**

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

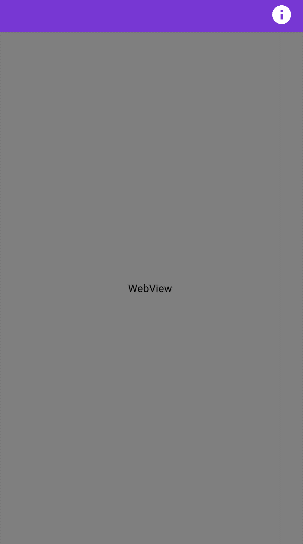


Рисунок 5 – Структура макета activity\_main

На макет графического интерфейса были добавлены следующие графические компоненты:

* ConstraintLayout – выравнивает все дочерние объекты в одном направлении – вертикально или горизонтально (ConstraintLayout);
* textView – предназначен для отображения текста без возможности редактирования его пользователем (textView);
* imageButton – предназначен для отображения изображений (imageButtonAbout);
* WebView – позволяет отображать веб-контент как часть макета активности (MainWV).

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

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="0dp"  
 android:layout\_height="50dp"  
 android:background="@color/purple\_500"  
 android:textColor="@color/white"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageButton  
 android:id="@+id/imageButtonAbout"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:layout\_marginEnd="15dp"  
 android:background="@drawable/baseline\_info"  
 android:onClick="showDialog"  
 app:layout\_constraintBottom\_toBottomOf="@+id/textView"  
 app:layout\_constraintEnd\_toEndOf="@id/textView"  
 app:layout\_constraintTop\_toTopOf="@+id/textView"  
 tools:ignore="SpeakableTextPresentCheck,TouchTargetSizeCheck" />  
  
 <WebView  
 android:id="@+id/MainWV"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="48dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"  
 tools:layout\_editor\_absoluteX="0dp" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Создание управляющего кода**

Листинг кода MainActivity.java:

package com.lab5;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
  
import android.view.View;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;  
  
  
public class MainActivity extends AppCompatActivity {  
 public void showDialog(View info) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
 builder.setTitle(getString(R.string.*info*))  
 .setMessage(R.string.*authors*)  
 .setPositiveButton(R.string.*ok*, null)  
 .show();  
 }  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 WebView webView = findViewById(R.id.*MainWV*);  
 webView.getSettings().setJavaScriptEnabled(true);  
 webView.setWebViewClient(new WebViewClient());  
 webView.loadUrl(getString(R.string.*API*));  
 }  
}

**2.3 Настройка виджета WebView, создание кода загрузки веб-страницы**

Основой графического интерфейса был выбран виджет WebView для последующей загрузки в него веб-страницы. Также была добавлена верхняя панель инструментов с кнопкой «Информация».

Класс WebView позволяет создать объект с настройками в соответствии со спецификой загружаемого в него HTML-кода, например:

- webSettings.setJavaScriptEnabled(true); //разрешаем JavaScript

Загрузка веб-страницы осуществляется при использовании класса WebViewClient – встроенного веб-браузера и виджета WebView как средства (контейнера) для просмотра загруженной веб-страницы.

Загрузка и визуализация веб-страницы выполняется созданием встроенного веб-браузера (webViewClient) и его указанием для визуализации веб-страницы в виджете myWebView:

setContentView(R.layout.*activity\_main*);

WebView webView = findViewById(R.id.*MainWV*);  
webView.getSettings().setJavaScriptEnabled(true);  
webView.setWebViewClient(new WebViewClient());  
webView.loadUrl(getString(R.string.*API*));

**2.4 Создание и оформление веб-страницы**

Веб-страница версталась в редакторе VisualStudio Code в виде документа HTML.

Была установлена кодировка веб-контента по умолчанию:

<meta charset="utf-8">

К веб-странице был подключен файл с каскадными стилями, прикладная специальная JS-библиотека JQuery, а также JS-код:

<link rel="stylesheet" href="styles.css">

<script src="jquery.min.js" type="text/javascript"></script>

<script src="script.js"></script>

Также была создана веб-форма с полями для ввода входных данных. Отдельно создан блок «DIV» для отображения результата запроса.

Листинг кода index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Расчет времени до кипения</title>

<link rel="stylesheet" href="styles.css">

<script src="jquery.min.js" type="text/javascript"></script>

<script src="script.js"></script>

</head>

<body>

<h2>Расчет времени до кипения</h2>

<form id="boilingForm">

<label for="liquidName">Выберите жидкость:</label>

<select id="liquidName">

<option value="вода">Вода</option>

<option value="этиловый спирт">Этиловый спирт</option>

<option value="нефть">Нефть</option>

<option value="глицерин">Глицерин</option>

<option value="бензол">Бензол</option>

</select><br><br>

<label for="volume">Объем (л):</label>

<input type="number" id="volume" step="0.01" required><br><br>

<label for="heaterWattage">Мощность нагревателя (Вт):</label>

<input type="number" id="heaterWattage" step="0.01" required><br><br>

<button type="submit" >Рассчитать</button>

</form>

<div id="result"></div>

</body>

</html>

**2.5 Создание управляющего JS-кода**

Для отправки GET/POST запроса с входными данными на веб-сервер и приема ответа с результатом вычислений был создан JS-скрипт в виде отдельного файла myscript1.js с «прикреплением» его к веб-странице.

Листинг кода script.js:

$(document).ready(function() {

$('#boilingForm').on('submit', function(e) {

e.preventDefault();

var liquidName = $('#liquidName').val();

var volume = $('#volume').val();

var heaterWattage = $('#heaterWattage').val();

$.ajax({

url: 'calculate.php',

type: 'GET',

dataType: 'json',

data: { liquidName, volume, heaterWattage },

success: function(response) {

console.log(response);

if(response.error == false)

{

$('#result').html('Время до кипения: ' + response.timeToBoil + ' секунд<br>' +

'Теплоемкость: ' + response.heatCapacity + ' Дж/(кг·К)<br>' +

'Температура кипения: ' + response.boilingTemperature + ' °C');

} else {

$('#result').html('Ошибка: ' + response.error);

}},

error: function(xhr, status, error) {

console.log(data);

$('#result').html('Ошибка: ' + error);

}

});

});

});

**2.6 Создание серверного скрипта**

Листинг кода calculate.php:

<?php

header('Content-Type: application/json; charset=utf-8');

// Принимаем параметры из запроса

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

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

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

// Определяем свойства жидкостей

$liquids = [

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

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

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

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

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

];

if($volume == 0 && $heaterWattage == 0) {

echo json\_encode(["error" => "Введите значения больше нуля"],JSON\_UNESCAPED\_UNICODE);

}

else if($volume == 0) {

echo json\_encode(["error" => "Значение объема должно быть больше нуля"],JSON\_UNESCAPED\_UNICODE);

}

else if($heaterWattage == 0) {

echo json\_encode(["error" => "Значение мощности должно быть больше нуля"],JSON\_UNESCAPED\_UNICODE);

}

else {

// 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 = [

"error" => false,

"timeToBoil" => round($timeToBoil, 2),

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

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

];

echo json\_encode($response,JSON\_UNESCAPED\_UNICODE);

} else {

// Жидкость не найдена

echo json\_encode(["error" => "Жидкость не найдена"],JSON\_UNESCAPED\_UNICODE);

}

}

?>

**2.7 Тестирование мобильного приложения**

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

Результат работы мобильного приложения представлен на рисунках 6-12.

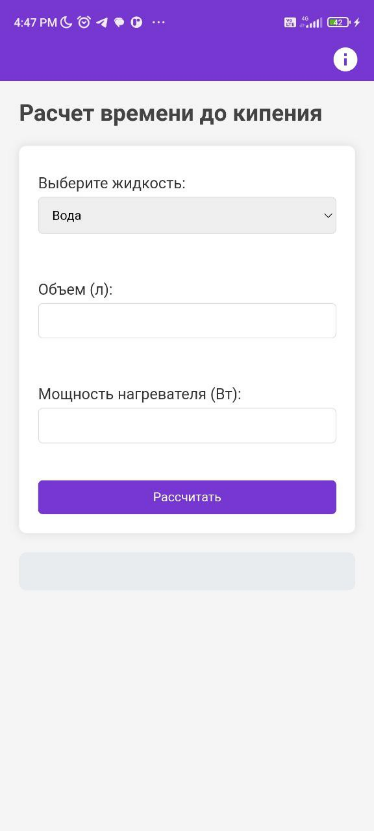


Рисунок 6 – Результат работы главной активности мобильного приложения на светлой теме, портретная ориентация

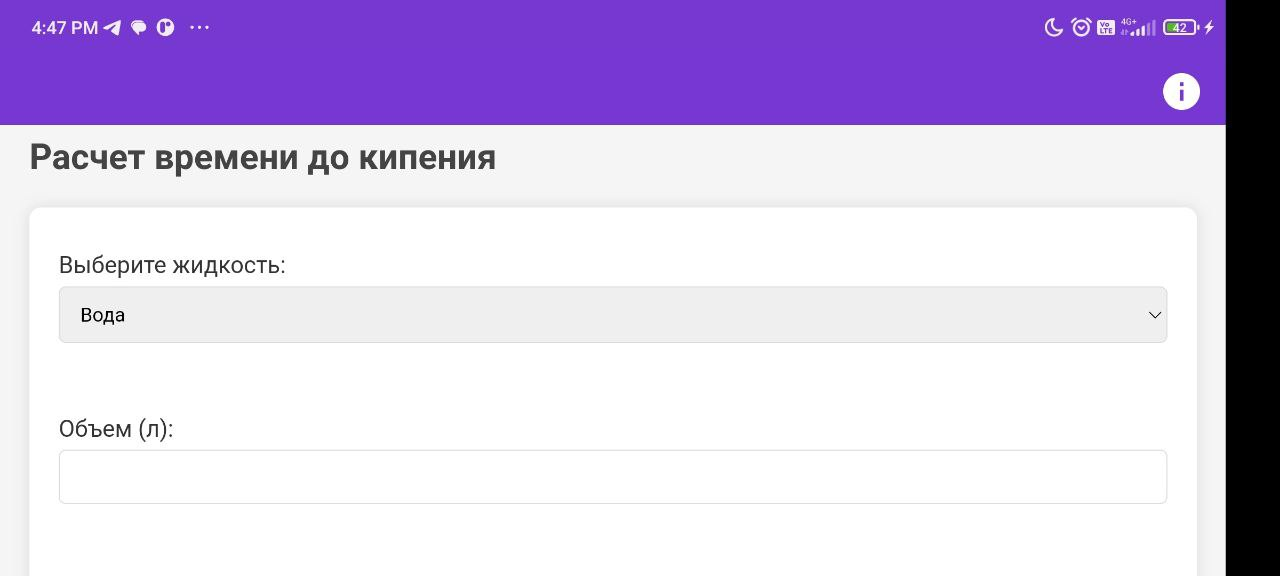


Рисунок 7 – Результат работы главной активности мобильного приложения на светлой теме, альбомная ориентация

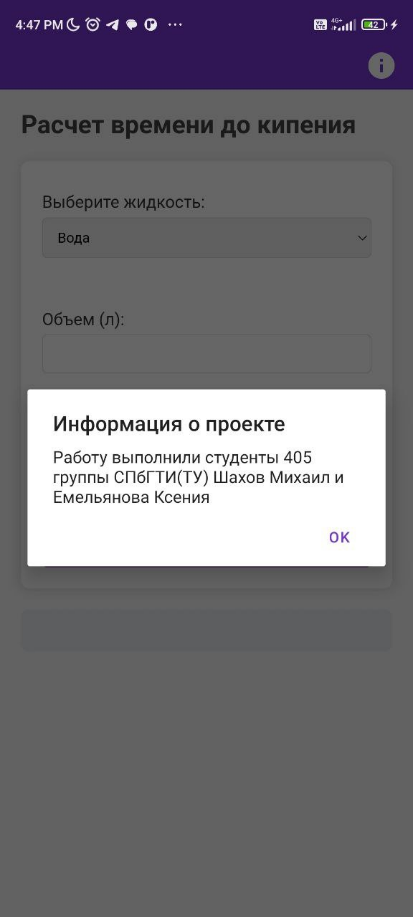


Рисунок 8 – Диалоговое окно мобильного приложения на светлой теме, портретная ориентация

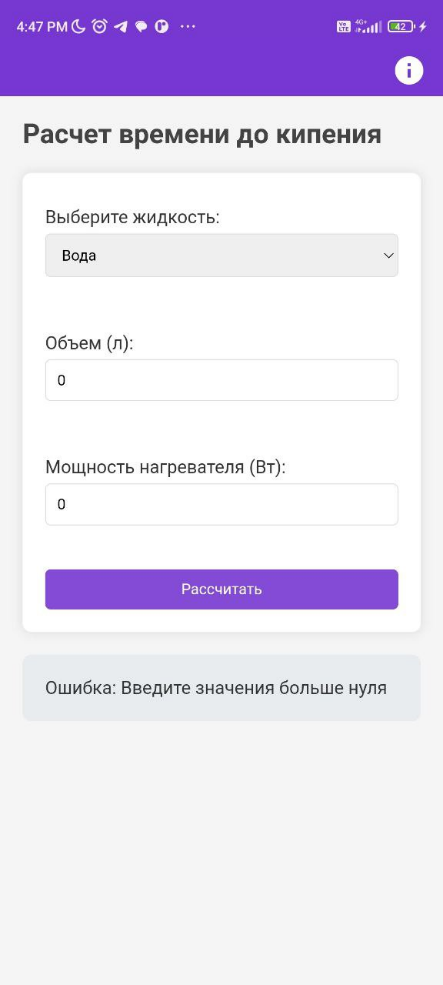


Рисунок 9 – Результат работы главной активности мобильного приложения с ошибкой ввода

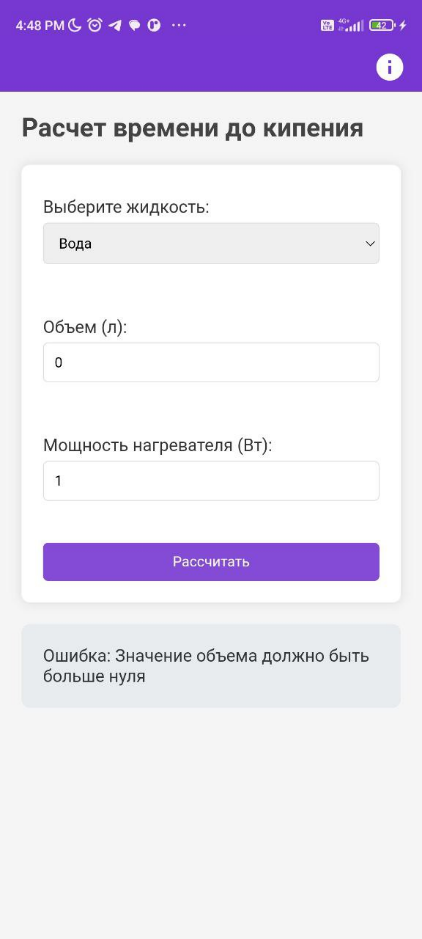


Рисунок 10 – Результат работы главной активности мобильного приложения с ошибкой ввода

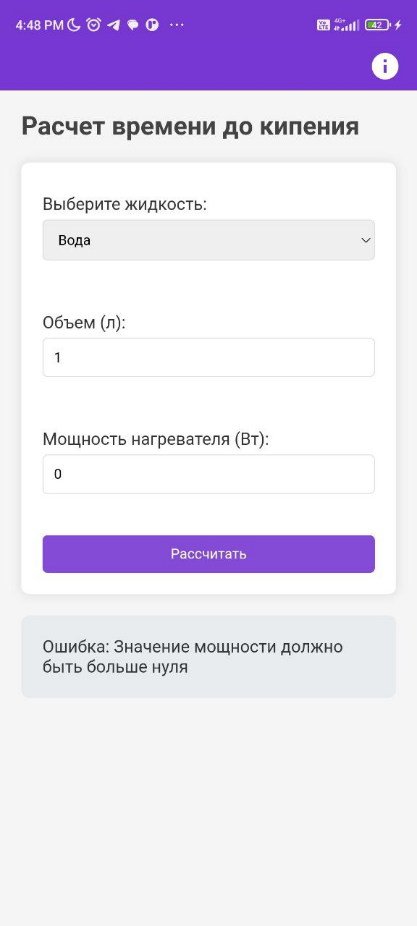


Рисунок 11 – Результат работы главной активности мобильного приложения с ошибкой ввода

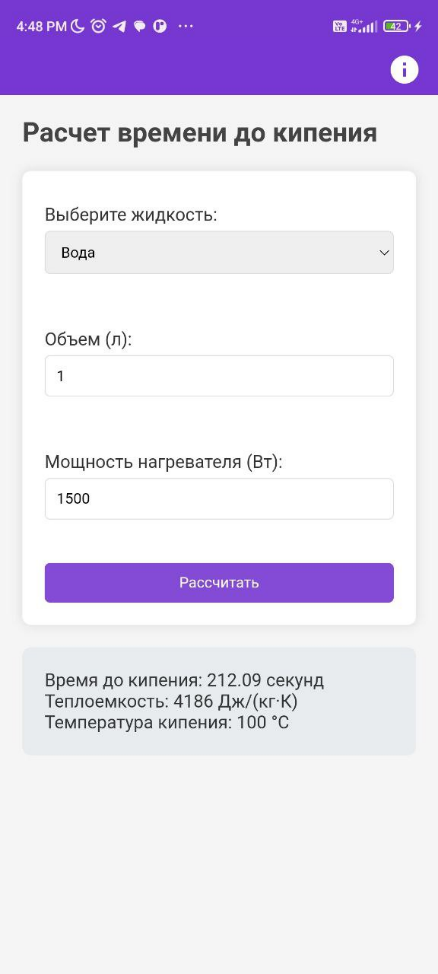


Рисунок 12 – Результат работы главной активности мобильного приложения

**Выводы**

В ходе выполнения лабораторной работы было разработано мобильное приложения для Android для загрузки веб-страницы. Для достижения поставленной цели были выполнены следующие задачи:

1) Изучены принципы слияния технологий разработки нетиповых мобильных приложений, обоснованы применения веб-технологий к разработке мобильных приложений. Описаны назначения и функционала виджета WebView. Преимущества и недостатки использования WebView в мобильном приложении. Изучены основы JavaScript, технологии AJAX, обзор JS-библиотек, поддерживающих асинхронные HTTP-запросы. Изучены каскадные стили CSS для оформления веб-страницы и динамики поведения ее DOM-объектов как реакций на события (смена расширения экрана, действия пользователя).

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 приложения;

- настройка виджета WebView, создание кода настройки и загрузки веб-страницы;

- создана и оформлена веб-страница;

- создан клиентский управляющий JS-кода;

- создан серверный скрипт;

- 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 5.

Таблица 5 – 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 |