TestPing

Software Documentation

Author: matiwa

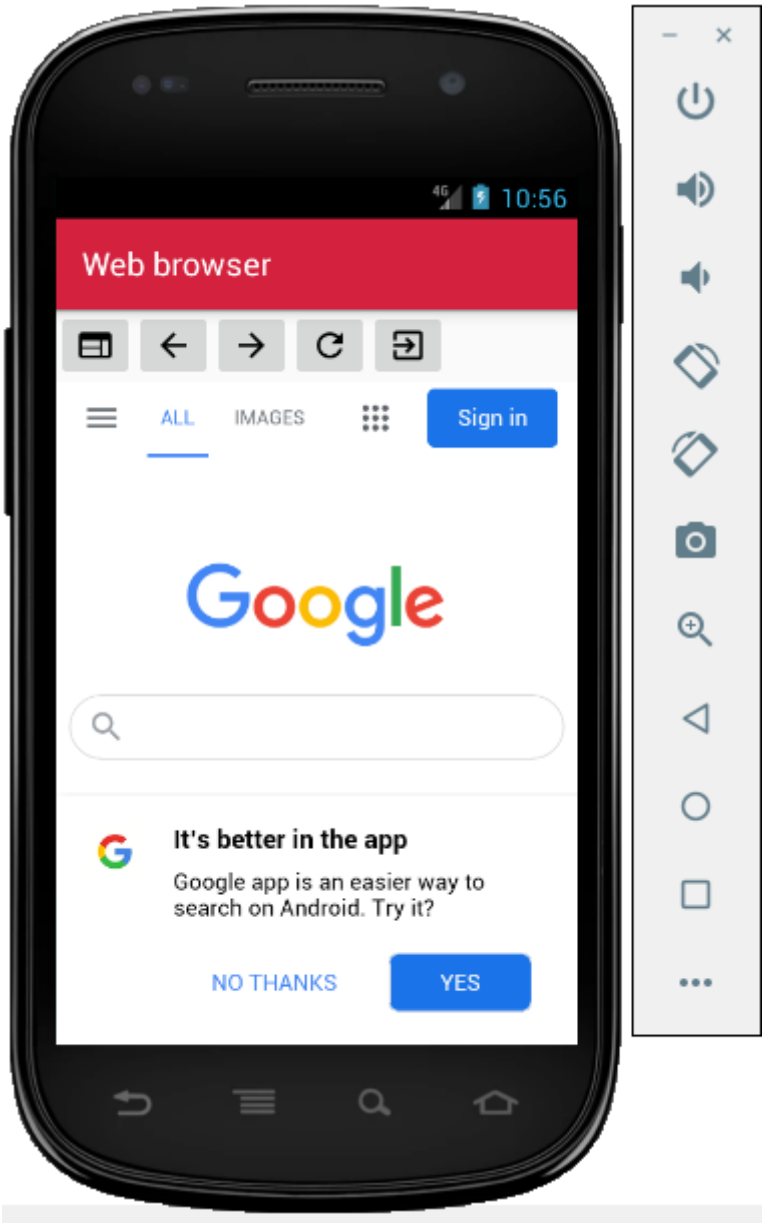
Table of contents

Table of contents……2
Introduction……...3
Describing of the application's operation……...3
What is needed for use?.....5
Algorithm used……5
Interface description……5
Source code description……..6
List of drawings……12
List of listings……12
Bibliography……..12

Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application serves as a web browser.

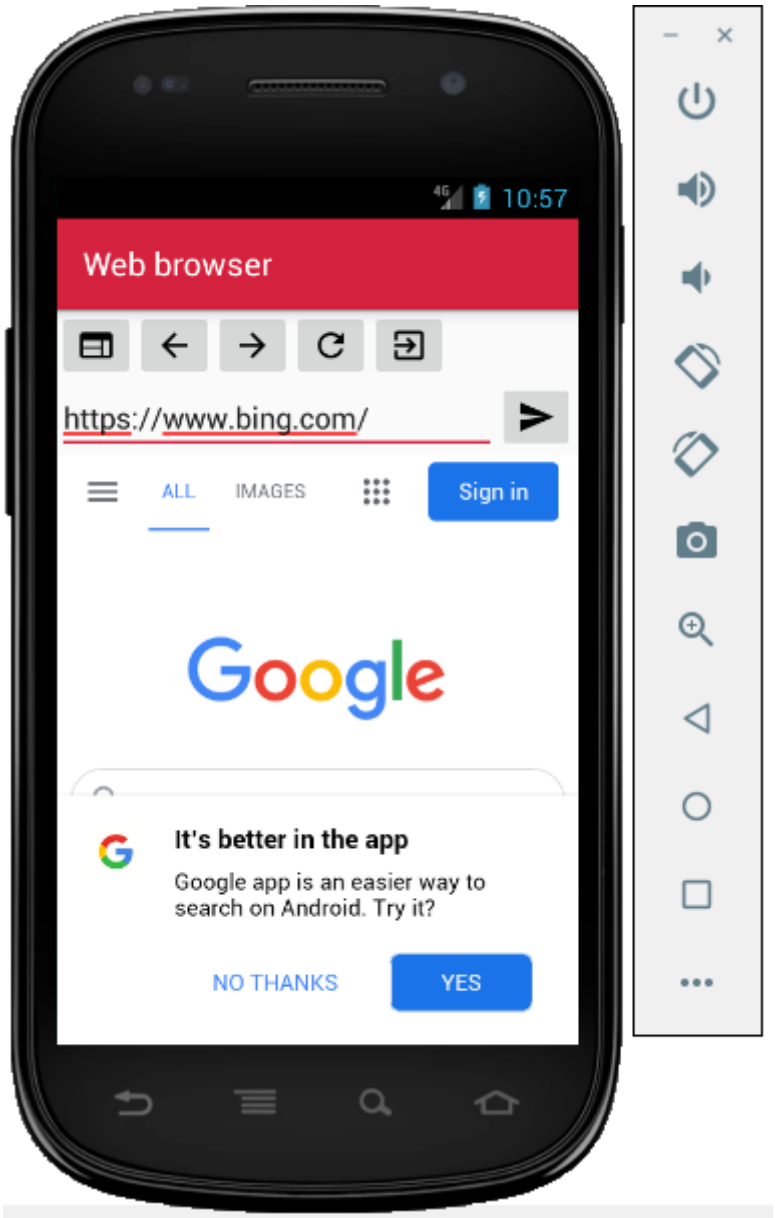
Describing of the application’s operation



Drawing 1: The beginning of the application's operation [own study]

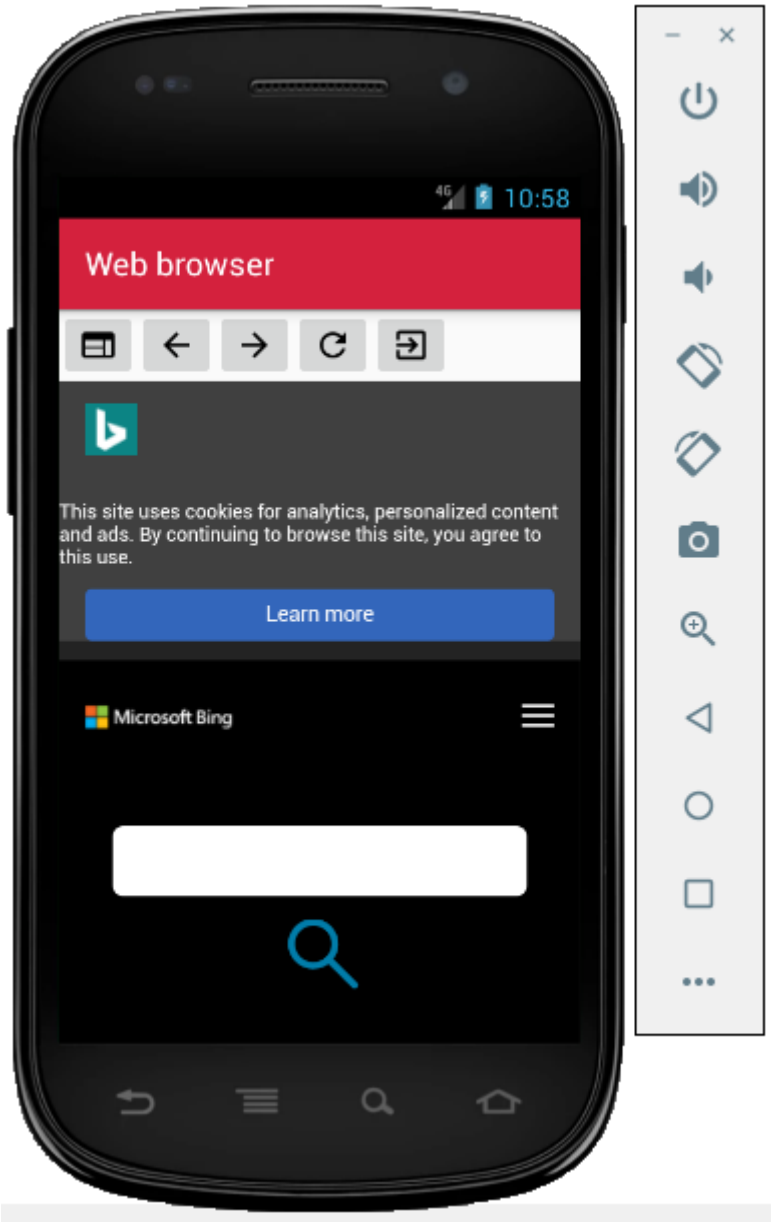
When opening the application, the google search engine loads. If the user wants to go to another page, he clicks the button to edit the url, enters the address (along with https: // or http: // !!!) and clicks the button to go to the page. It can also go back to the previous page or

exit the application. There is some precaution that the user may not enter a prefix to the url. Nevertheless, it is worth remembering to enter it.



Drawing 2: Entering an url address [own study]

You can also refresh the page that the user is currently viewing by clicking the button.



Drawing 3: Displaying the results [own study]

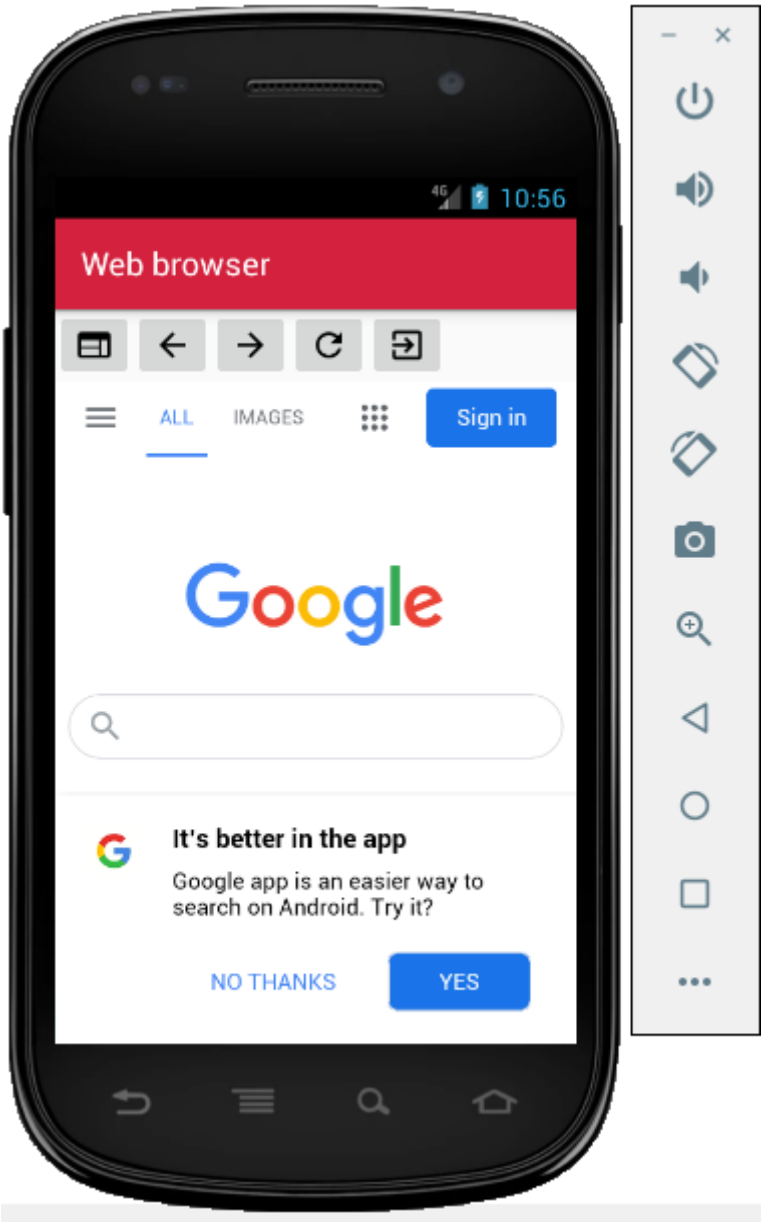
What is needed for use?

The application requires installation on the Android operating system with a minimum API level Android 4.0 (IceCreamSandwich), i.e. on all mobile devices with this system.

Algorithm used

Web browser - a computer program used to download and display websites provided by web servers, as well as play multimedia files, often using add-ons called plug-ins. [1]

Interface description



Drawing 4: Graphical interface [own study]

The interface has the basic components available in the Android Studio development environment: WebView, ImageButton and EditText.

Source code description

The project was made in the Java programming language, in the Android Studio programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
package com.example.webbrowser;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ImageButton;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private ImageButton
imgbtnsend,imgbtnweb,imgbtnback,imgbtnforward,imgbtnreload,imgbtnexi
t;
    private WebView mywebview;
    private EditText eurl;
    private LinearLayout panel;
    private String url="https://www.google.pl/";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initializecomponents();
    }

    private void initializecomponents(){
        eurl=findViewById(R.id.eurl);
        eurl.setText(url);
        imgbtnsend=findViewById(R.id.imgbtnsend);
        imgbtnweb=findViewById(R.id.imgbtnweb);
        imgbtnback=findViewById(R.id.imgbtnback);
        imgbtnforward=findViewById(R.id.imgbtnforward);
        imgbtnreload=findViewById(R.id.imgbtnreload);
        imgbtnexit=findViewById(R.id.imgbtnexit);
        panel=findViewById(R.id.panel);
        panel.setVisibility(View.GONE);
        mywebview = findViewById(R.id.webView);
        mywebview.setWebViewClient(new WebViewClient());
        WebSettings ws=mywebview.getSettings();
        ws.setJavaScriptEnabled(true);
        mywebview.getSettings().setAppCacheEnabled(true);
        mywebview.setVerticalScrollBarEnabled(false);
        mywebview.setHorizontalScrollBarEnabled(false);
        mywebview.loadUrl(url);
        imgbtnsend.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                loadwebpage();
            }
        });
        imgbtnweb.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                panel.setVisibility(View.VISIBLE);
            }
        });
        imgbtnback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mywebview.canGoBack();
                onBackPressed();
            }
        });
        imgbtnforward.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                mywebview.canGoForward();
                onForwardPressed();
            }
        });
        imgbtnreload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onReloadPressed();
            }
        });
        imgbtnexit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }

    @Override
    public void onBackPressed() {
        if(mywebview.canGoBack()){
            mywebview.goBack();
        }else {
            finish();
            //super.onBackPressed();
        }
    }
```

```
    public void onForwardPressed() {
        if(mywebview.canGoForward()){
            mywebview.goForward();
        }else {
            //super.onForwardPressed();
        }
    }

    public void onReloadPressed() {
        mywebview.reload();
    }

    public void loadwebpage(){
        try{
            url=eurl.getText().toString();
            //if(url.substring(0,6)!="http://" &&
url.substring(0,7)!="https://") url="https://"+url;
            mywebview.loadUrl(url);
            eurl.setText(url);
        }catch(Exception e){
            Toast.makeText(MainActivity.this,"Wprowadź poprawny
adres url!",Toast.LENGTH_LONG).show();
        }finally {
            eurl.setText(null);
            panel.setVisibility(View.GONE);
        }
    }
}
```

Listing 1: The source code of MainActivity.java file [own study]

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <ImageButton
            android:id="@+id/imgbtnweb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_web_black_24dp" />

        <ImageButton
            android:id="@+id/imgbtnback"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_arrow_back_black_24dp" />

        <ImageButton
            android:id="@+id/imgbtnforward"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_arrow_forward_black_24dp"
            android:text="Forward" />

        <ImageButton
            android:id="@+id/imgbtnreload"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_refresh_black_24dp"
            android:text="Reload" />

        <ImageButton
            android:id="@+id/imgbtnexit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_exit_to_app_black_24dp"
            android:text="Exit" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/panel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
```

```
            <EditText
                android:id="@+id/eurl"
                android:layout_width="270dp"
                android:layout_height="wrap_content" />

            <ImageButton
                android:id="@+id/imgbtnsend"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:src="@drawable/ic_send_black_24dp"
                android:text="Send" />

        </LinearLayout>

        <WebView
            android:id="@+id/webView"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </LinearLayout>

</RelativeLayout>
```

Listing 2: The source code of activity_main.xml file [own study]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.webbrowser">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Listing 3: The source code of AndroidMainfest.xml file [own study]

The following code snippet from the AndroidManifest.xml file enables you to connect to the global internet network. For this reason, its content has been included in the documentation.

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

Listing 4: The source code snippet from the AndroidManifest.xml file enables you to connect to the global internet network [own study]

List of drawings

Drawing 1: The beginning of the application's operation [own study]……...3

Drawing 2: Entering an url address [own study]……4

Drawing 3: Displaying the results [own study]……..5

Drawing 4: Graphical interface [own study]……..…….7

List of listings

Listing 1: The source code of MainActivity.java file [own study]…….7
Listing 2: The source code of activity_main.xml file [own study]…….9
Listing 3: The source code of AndroidMainfest.xml file [own study]……11
Listing 4: The source code snippet from the AndroidManifest.xml file enables you to connect to the global internet network.[own study]..…….12

Bibliography

[1] https://pl.wikipedia.org/wiki/Przegl%C4%85darka_internetowa