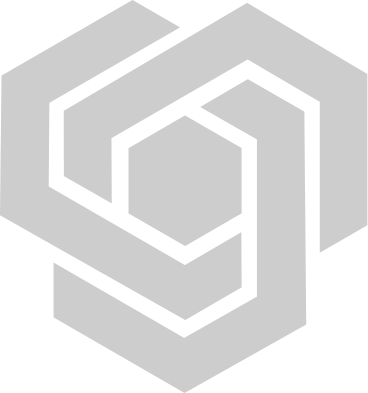
**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ СОФИЯ**

****

**Курсов Проект**

**по**

**Програмиране на мобилни устройства**

Тема: **Приложение за показване на времето**

Изготвена от:Стефан Филиповски и Андрей Зафиров

(Фак. №:123220004) (Фак. №:123220013)

Факултет: ФКСТ

Специалност: КСИ

Модул: ПС

Група: 44

**Съдържание**

1. Увод..........................................................................................3
2. Анализ на съществуващи разработки....................................4
3. Проектиране.............................................................................5
4. Софтуерна реализация.............................................................8
5. Потребителско ръководство....................................................15
6. Заключение................................................................................20
7. Използвана литература.............................................................20

**Увод**

Приложенията за времето на днешните мобилни телефони са едни от най-ползваните приложения на телефоните.Приложенията за времето са построени за потребителя да има лесен и удобен достъп до прогнозата за времето. По едно време хората имали възможност да видят прогнозата за времето само на телевизията, но днес всеки който ползва Смарт телефон има приложение за времето.

Много от хората не вярват на приложенията, но дали някога са си задали въпрос от къде се взимат тия данни за времето?

Националната метеорологична служба прекарват часове в изливане на данни от компютърни модели, сателити, радари и наблюдения на повърхността.

Прогнозите не са винаги 100% верни, но дали прогнозите на телевизията са 100% верни ?

Това е природа така че не може да предвидим на 100% какво ще бъде времето.

**Анализ на съществуващи разработки**

Това приложение за времето е построено за устройства който работят с операционната система Android.Приложението има много конкуренти който могат да бъдат намерени в Google Play магазина намиращ се на всички Android устройства.Известни приложения за времето на Android са: WeatherBug,AccuWeather,Google Weather и други.

Чрез това приложение имате достъп до времето във всеки един момент. Имате възможност да видите времето във всеки един град.

Положителни страни:

* Достъп до времето през цялото време
* Достъп до времето във всички места по желание
* Удобно приложение за използване

Отрицателни страни:

* Прогнозата не е 100% верна
* Показва прогнозата за текущия ден
* Не показва прогнозата за една седмица

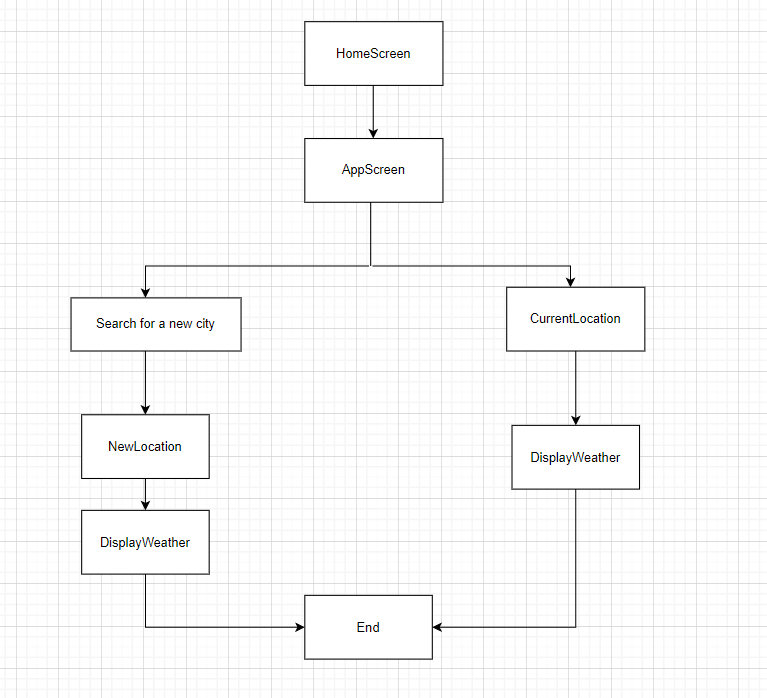
**Проектиране**

Приложението е разработено чрез “Android Studio”, програма която се ползва за разработка на Android приложения.Програмата е построена да поддържа устройства, работещи с Android 4.4 и нагоре. Самото това ни казва че всички днешни телефони, ползващи оперативната система Android, ще могат да го изпозлват. Имаме напълно функционално приложение което взима локацията на телефона и показва времето в тази област.Сщо така може да виждаме прогнозата в място по желание като напишем името на мястото и приложението ще ни покаже прогнозата за това място. Това приложение може да се използва от всеки потребител който ползва телефон с операционна система Android.

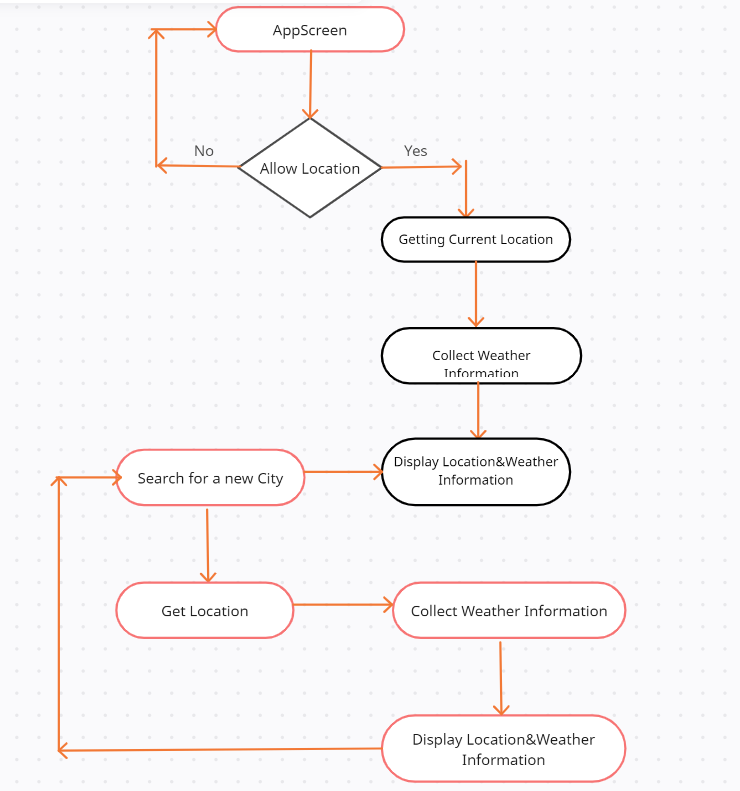
Приложението няма възрастна граница за изпозлване което означава че всички хора от най-малка възраст до най-възрастните ще могат да я използват.

Данните, които ще използва потребителят ще са предоставени от Метеорологичните станции по света и ще ни предоставят нужните данни за най-точна прогноза на времето.

Блокова схема:



Алгоритъм:



**Софтуерна реализация**

1.Клас „cityFinder”:

public class cityFinder extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_city\_finder*);  
 final EditText editText=findViewById(R.id.*searchCity*);  
 ImageView backButton=findViewById(R.id.*backButton*);  
  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 editText.setOnEditorActionListener(new TextView.OnEditorActionListener() {  
 @Override  
 public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
 String newCity= editText.getText().toString();  
 Intent intent=new Intent(cityFinder.this,MainActivity.class);  
 intent.putExtra("City",newCity);  
 startActivity(intent);  
  
  
  
 return false;  
 }  
 });

Този код създава Activity (приложение) за търсене на градове. Има поле за въвеждане на име на град (EditText) и бутон за връщане назад (ImageView).  
При натискане на бутона за връщане назад се извиква методът finish() на Activity, който я затваря и се връща към предишната Activity.  
При въвеждане на име на град в полето (EditText), се извиква методът setOnEditorActionListener() на EditText, който изчаква потребителя да натисне даден клавиш на клавиатурата. Ако това се случи, се извлича текстът в полето и се създава нов Intent (намерение) за преминаване към MainActivity.

2. Клас „MainActivity”:

public class MainActivity extends AppCompatActivity {  
  
  
 final String APP\_ID = "dab3af44de7d24ae7ff86549334e45bd";  
 final String WEATHER\_URL = "https://api.openweathermap.org/data/2.5/weather";

Класът MainActivity е основният клас на приложението и наследява класа AppCompatActivity от библиотеката за дизайн на Android.

Константите APP\_ID и WEATHER\_URL съхраняват API ключа и URL адреса за извличане на данни за времето от уеб услугата OpenWeatherMap.

OpenWeatherMap е доставчик на метеорологични данни, който предоставя безплатен API за използване от разработчици на софтуер. Приложението използва тези данни за да предоставя информация за времето в различни градове на света.

protected void onResume() {  
 super.onResume();  
 Intent mIntent=getIntent();  
 String city= mIntent.getStringExtra("City");  
 if(city!=null)  
 {  
 getWeatherForNewCity(city);  
 }  
 else  
 {  
 getWeatherForCurrentLocation();  
 }  
  
  
}  
  
  
private void getWeatherForNewCity(String city)  
{  
 RequestParams params=new RequestParams();  
 params.put("q",city);  
 params.put("appid",APP\_ID);  
 letsdoSomeNetworking(params);  
  
}  
  
  
  
  
private void getWeatherForCurrentLocation() {  
  
 mLocationManager = (LocationManager) getSystemService(Context.*LOCATION\_SERVICE*);  
 mLocationListner = new LocationListener() {  
 @Override  
 public void onLocationChanged(Location location) {  
  
 String Latitude = String.*valueOf*(location.getLatitude());  
 String Longitude = String.*valueOf*(location.getLongitude());  
  
 RequestParams params =new RequestParams();  
 params.put("lat" ,Latitude);  
 params.put("lon",Longitude);  
 params.put("appid",APP\_ID);  
 letsdoSomeNetworking(params);  
  
  
  
  
 }

Кодът има две метода за получаване на данни за времето: getWeatherForNewCity() и getWeatherForCurrentLocation(). Методът getWeatherForNewCity(String city) получава информация за времето за определен град. Методът getWeatherForCurrentLocation() се използва за получаване на информация за времето на текущата локация на устройството.

private void letsdoSomeNetworking(RequestParams params)  
{  
 AsyncHttpClient client = new AsyncHttpClient();  
 client.get(WEATHER\_URL,params,new JsonHttpResponseHandler()  
 {  
 @Override  
 public void onSuccess(int statusCode, Header[] headers, JSONObject response) {  
  
 Toast.*makeText*(MainActivity.this,"Data Get Success",Toast.*LENGTH\_SHORT*).show();  
  
 weatherData weatherD=weatherData.*fromJson*(response);  
 updateUI(weatherD);  
  
 }  
  
  
 @Override  
 public void onFailure(int statusCode, Header[] headers, Throwable throwable, JSONObject errorResponse) {  
 }  
 });  
  
  
  
}  
  
private void updateUI(weatherData weather){  
  
  
 Temperature.setText(weather.getmTemperature());  
 NameofCity.setText(weather.getMcity());  
 weatherState.setText(weather.getmWeatherType());  
 int resourceID=getResources().getIdentifier(weather.getMicon(),"drawable",getPackageName());  
 mweatherIcon.setImageResource(resourceID);  
  
  
}

Тук се използва библиотеката AsyncHttpClient, за да направи заявка към API за времето (WEATHER\_URL), използвайки параметрите, подадени чрез обекта RequestParams. Функцията updateUI се използва за обновяване на потребителския интерфейс с новата информация за времето, получена от заявката. Тя приема обект от клас weatherData, който съдържа информацията за времето. В тази функция се извличат различните свойства на weatherData като температура, град, тип на времето и икона на времето и се обновяват в съответните полета в потребителския интерфейс.

3. Клас „weatherData”

public class weatherData {  
  
 private String mTemperature,micon,mcity,mWeatherType;  
 private int mCondition;  
  
 public static weatherData fromJson(JSONObject jsonObject)  
 {  
  
 try  
 {  
 weatherData weatherD=new weatherData();  
 weatherD.mcity=jsonObject.getString("name");  
 weatherD.mCondition=jsonObject.getJSONArray("weather").getJSONObject(0).getInt("id");  
 weatherD.mWeatherType=jsonObject.getJSONArray("weather").getJSONObject(0).getString("main");  
 weatherD.micon=*updateWeatherIcon*(weatherD.mCondition);  
 double tempResult=jsonObject.getJSONObject("main").getDouble("temp")-273.15;  
 int roundedValue=(int)Math.*rint*(tempResult);  
 weatherD.mTemperature=Integer.*toString*(roundedValue);  
 return weatherD;  
 }  
  
  
 catch (JSONException e) {  
 e.printStackTrace();  
 return null;  
 }  
  
  
 }

Класът weatherData се използва за обработка на информацията за времето, получена от заявката към API-то. Класът съдържа променливи за температура, икона, град и вид на времето, както и метод за извличане на тези данни от JSON обекта, получен от API-то.

4. „activity\_main.xml ”

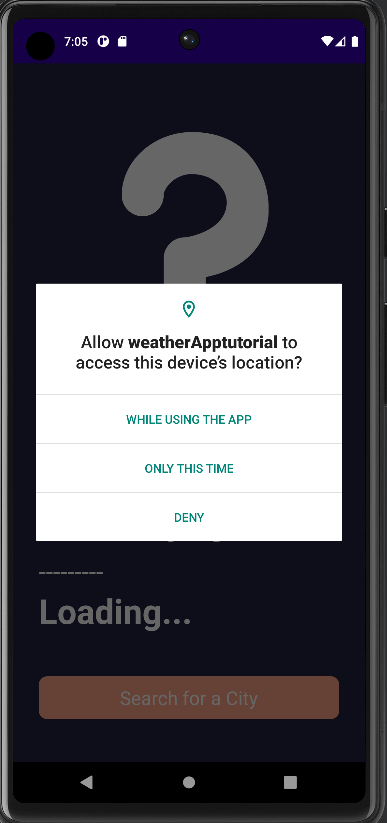
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#242343"  
 tools:context=".MainActivity">  
  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="270dp"  
 android:id="@+id/weatherIcon"  
 android:src="@drawable/finding"  
 android:layout\_marginTop="80dp">  
  
 </ImageView>  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="50dp"  
 android:orientation="vertical"  
 android:layout\_above="@id/cityFinder">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/temperature"  
 android:paddingStart="170dp"  
 android:textSize="50sp"  
 android:textStyle="bold"  
 android:textColor="#ffffff"  
 android:text="0\*C">  
  
 </TextView>  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/weatherCondition"  
 android:textSize="30sp"  
 android:textColor="#ffffff"  
 android:text="---------"  
 android:paddingStart="30sp">  
  
 </TextView>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/cityName"  
 android:textStyle="bold"  
 android:textSize="40sp"  
 android:text="Loading..."  
 android:paddingStart="30sp"  
 android:textColor="#ffffff">  
  
 </TextView>  
  
 </LinearLayout>  
  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_centerInParent="true"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginRight="30dp"  
 android:layout\_alignParentBottom="true"  
 android:id="@+id/cityFinder"  
 android:background="@drawable/buttondesign"  
 android:layout\_marginBottom="50dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Search for a City"  
 android:textColor="#ffffff"  
 android:layout\_centerInParent="true"  
 android:textSize="22sp" />  
  
 </RelativeLayout>  
  
</RelativeLayout>

Това е кодът на XML файла за дизайна на главния екран на приложението. В него има основен RelativeLayout, в който са разположени ImageView, показващ графиката на времето, и LinearLayout, в който са разположени TextView, показващи информация за температурата, състоянието на времето и името на града. В долната част на екрана има втори RelativeLayout, който е стилизиран като бутон и съдържа TextView с текст "Search for a City", който ще отвори activity за търсене на град при кликване върху него. Освен това има и някои настройки за размери, цветове и маргини на различните елементи в дизайна.

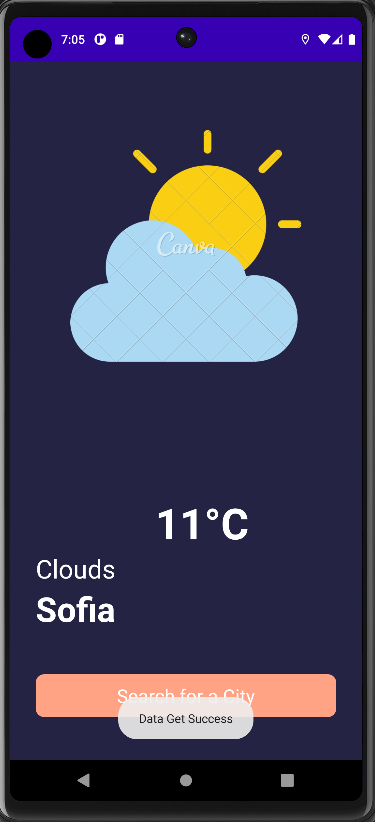
**Потребителско ръководство (Резултати)**

На снимките е показано как работи приложението:

Фиг 1. Приложението изисква од потребителя разрешение за користене на локация.



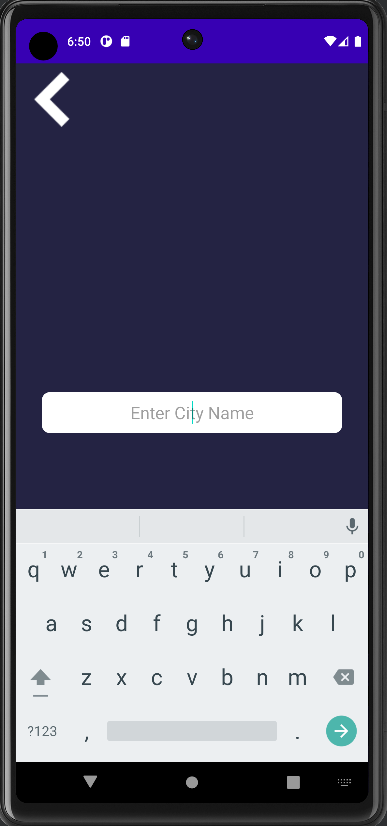
Фиг 2. След получаване на разрешение, потребителя вижда времето в момента.



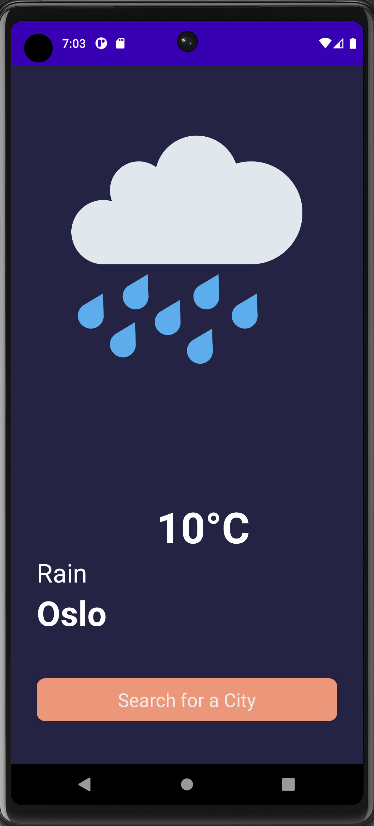
Фиг 3. Когато потребителя не е позволил приложението да има достъп до локация



Фиг 4. Въвеждане на град



Фиг 5. След въвеждането на желаното място, връща информация.



**Заключение**

Приложението за прогноза на времето „Weather App” работи без проблеми. Чрез това приложение потребителя може да провери прогнозата за времето на локацията кадето се намира, също така може да провери времето на всяко едно място на света чрез въвеждане на името на мястото.Приложението е направено така че да може да го използват хора, притежаващи телефони с операционна система Android.

**Използвана литература**

1. <https://www.geeksforgeeks.org/how-to-build-a-weather-app-in-android/>
2. <https://www.youtube.com/watch?v=FtmI0qqQsl8&ab_channel=TechProjects>
3. <https://www.youtube.com/watch?v=VHgM_MQBQPg&ab_channel=TechProjects>
4. <https://code.tutsplus.com/tutorials/create-a-weather-app-on-android--cms-21587>
5. Упражнения по ПМУ
6. https://app.diagrams.net/#