

Софийски университет „Св. Климент Охридски“



Курсова работа по "Мобилни приложения"

Времето (Weather v1.0)

ANDROID

Изготвил: Юсуф Маджаров

Ръководител: Георги Байчев

Специалност: Компютърни науки

Курс: 3-ти

Факултетен номер: 80815



Приложението е свързано със информация за времето за избран град. Предоставя информация за

- текущата температура
- влажност на въздуха
- атмосферно налягане
- името на града и държавата
- точната дата на последното обновяване на данните.

Приложението динамично сменя фоновото изображение изхождайки от това какво е времето в дадения град. Има две възможности за избиране на града:

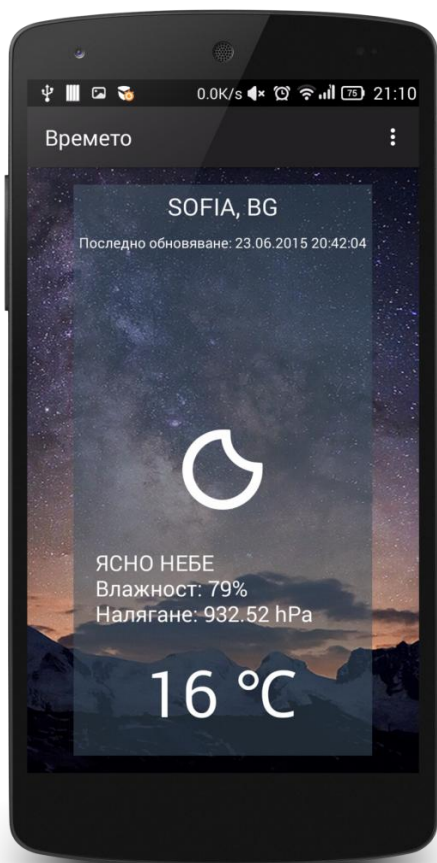
- ръчно въвеждане името на града
- използване на текущото местоположение

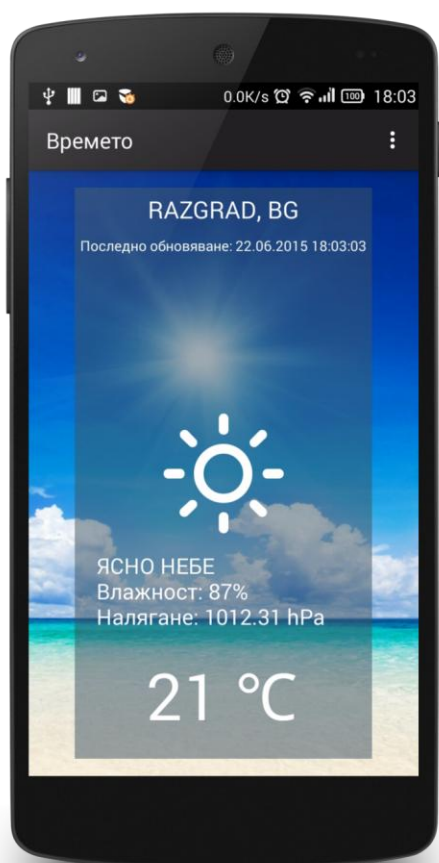
Приложението подлежи на добавяне на още функционалности, като няколко дневна прогноза, вероятност за валеж и др; възможност на потребителя да добавя картинки за фоново изображение и др.

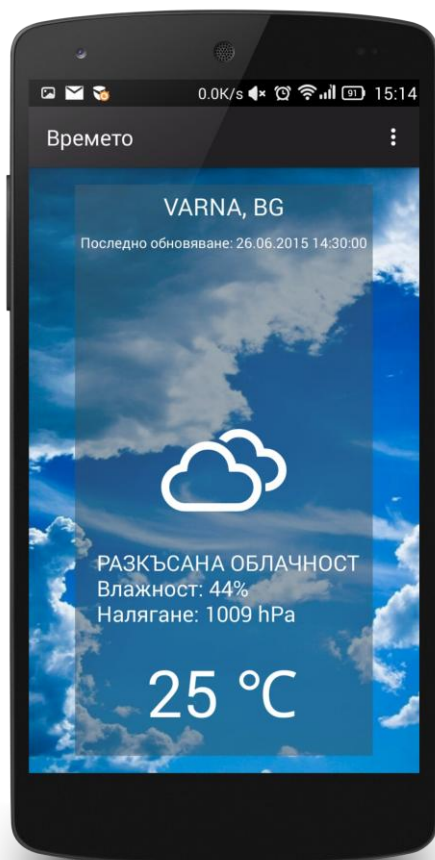
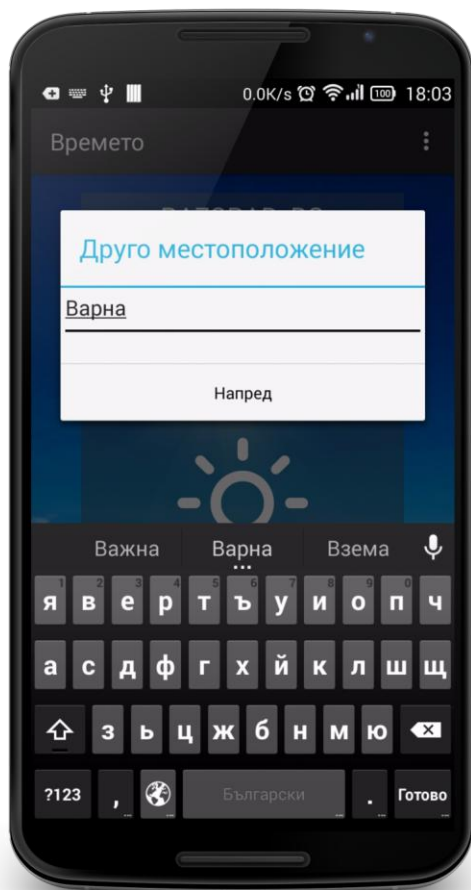
За предоставяне на необходимата информацията за времето за съответния град е използван API предоставян от OpenWeatherMap. Сайта предоставя информация за над 200,000 града по света, като получената информация е достъпна в JSON формат.



Използвани са няколко класа за подаване на заявка към openweathermap.org, обработката на получения отговор и съответното визуализиране на екрана. Изгледи на екраните:









Приложението използва основно едно Activity - **WeatherActivity.java** и фрагмент **WeatherFragment.java**, в който се парсва получената информация, след което се визуализира. В **WeatherActivity.java** се стартира google api client-а чрез, който се взимат текущите координати. За целта се използва класът **FetchAddressIntentService.java**. Стартира се intent service, чрез който се предава информация между **WeatherActivity.java** и **FetchAddressIntentService.java**.

По важните методи в **FetchAddressIntentService.java** са:

```
@Override  
protected void onHandleIntent(Intent intent)
```

 - тук се получават текущите координати на устройството, като се използва **geocoder.getFromLocation()**, за намиране на името на града по тях, ако **getFromLocation()** не върне резултат се използва директно Google Map Api, чрез:

```
//-----GOOGLE MAP START-----  
  
private String fetchCityNameUsingGoogleMap(Location location)
```

 в случай когато **getFromLocation()** не върне нищо, се използва директно Google Map Api

```
private void deliverResultToReceiver(int resultCode, String message)
```



Във фрагмента **WeatherFragment.java** по-важните методи са:

```
@Override
public View onCreateView(LayoutInflater inflater,
                          ViewGroup container,
                          Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_weather, container, false);
    cityField = (TextView)rootView.findViewById(R.id.city_field);
    updatedField = (TextView)rootView.findViewById(R.id.updated_field);
    detailsField = (TextView)rootView.findViewById(R.id.details_field);
    currentTemperatureField = (TextView)rootView.findViewById(R.id.current_temperature_field);
    weatherIcon = (TextView)rootView.findViewById(R.id.weather_icon);

    weatherIcon.setTypeface(weatherFont);
    return rootView;
}

private void updateWeatherData(final String city){
    new Thread(){
        public void run(){
            final JSONObject json = RemoteFetch.getJSON(getActivity(), city);
            if(json == null){
                handler.post(new Runnable(){
                    public void run(){
                        Toast.makeText(getActivity(),
                                    getActivity().getString(R.string.place_not_found),
                                    Toast.LENGTH_LONG).show();
                    }
                });
            } else {
                handler.post(new Runnable(){
                    public void run(){
                        renderWeather(json);
                    }
                });
            }
        }
    }.start();
}
```



```
private void renderWeather(JSONObject json){

private void setWeatherIcon(int actualId, long sunrise, long sunset){

private void setWeatherBackground(int actualId, long sunrise, long sunset){

public void changeCity(String city){
public static String transliterate2(String message){
```

Класът RemoteFetch.java се грижи за подаване на заявка

```
public class RemoteFetch {

    private static final String OPEN_WEATHER_MAP_API_CITY_NAME =
        "http://api.openweathermap.org/data/2.5/weather?q=%s&units=metric&lang=bg";

    private static final String OPEN_WEATHER_MAP_API_COORDINATES =
        "api.openweathermap.org/data/2.5/weather?lat=%d&lon=%d";
    // If use coordinates, not city name
    public static JSONObject getJSONC(Context context, double latitude, double longitude){...}
    //
    public static JSONObject getJSON(Context context, String city){
        try {
            URL url = new URL(String.format(OPEN_WEATHER_MAP_API_CITY_NAME, city));
            HttpURLConnection connection =
                (HttpURLConnection)url.openConnection();

            connection.setRequestProperty("x-api-key",
                "b4caa3e8267dda2068deacc56f211a9e");

            BufferedReader reader = new BufferedReader(
                new InputStreamReader(connection.getInputStream()));

            StringBuffer json = new StringBuffer(1024);
            String tmp="";
            while ((tmp=reader.readLine()) != null)
                json.append(tmp).append("\n");
            reader.close();

            JSONObject data = new JSONObject(json.toString());

            // This value will be 404 if the request was not
            // successful
            if (data.getInt("cod") != 200){
                return null;
            }

            return data;
        } catch (Exception e){
            return null;
        }
    }
}
```