P3 Dokumentation

Olle Olsson
Jimmy Åkesson
Linus Forsberg

API'er som används i applikationen

Sign in with Google

OpenWeatherMapAPI

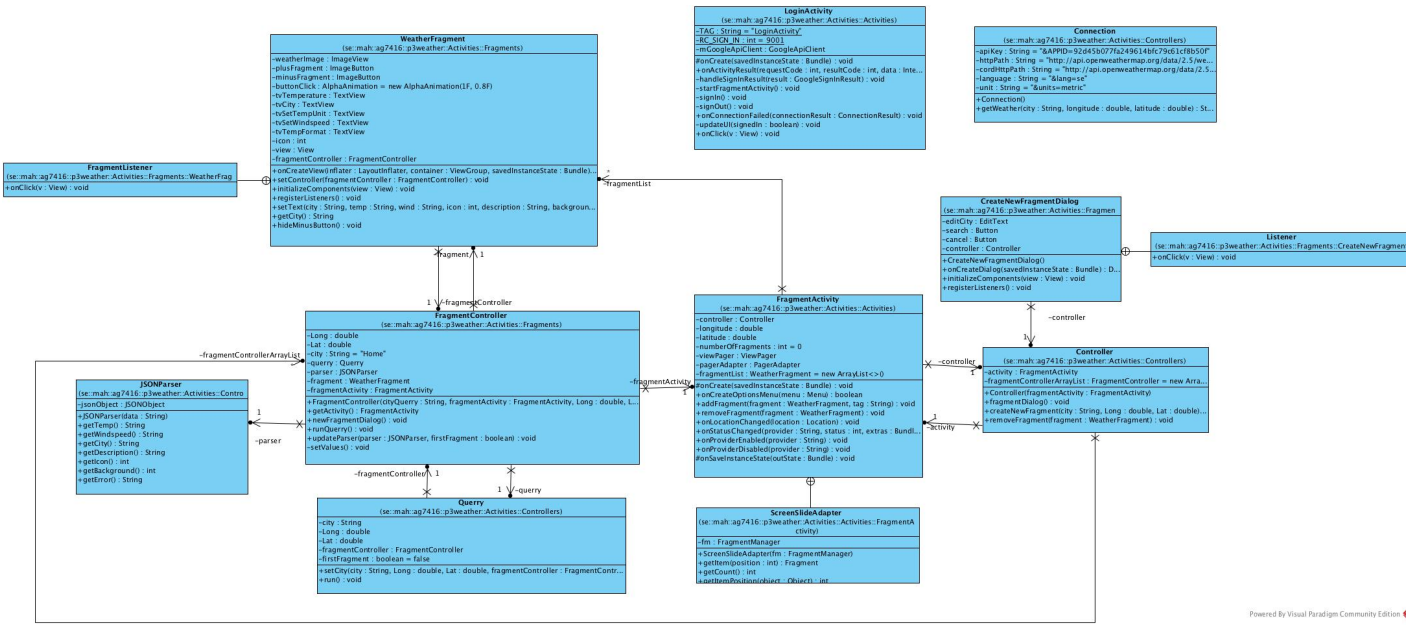
Google Sign-in

Googles egna API för inloggning. API'et har använts för att låta användaren logga in med befintligt Google-konto för att kunna nyttja applikationen. Nyckeln som gett gruppen tillgång till att nyttja API'et är endast för utveckling och inte en publik nyckel.

OpenWeatherMap (openweathermap.org)

Ett API som hämtar väderinformation och visar vädret för användaren. Användaren har möjlighet att lägga till så många städer som möjligt och kan även ta bort dessa om fel stad läggs till eller om användaren inte längre vill visa just den staden i applikationen. Applikationens grafiska design är interaktiv förhållande till vilket väder som ligger över staden som användaren valt. Olika bakgrundsbilder visas i applikationen beroende på väderleken och där finns ytterligare ett förtydligande med en ikon centrerad på skränen som också ändras efter väderleken.

Klassdiagram



Källkod

Connection.java
Skriven av: Jimmy Åkesson

```
package se.mah.ag7416.p3weather.Activities.Controllers;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class Connection {

    private String apiKey = "&APPID=92d45b077fa249614bfc79c61cf8b50f";
    private String httpPath = "http://api.openweathermap.org/data/2.5/weather?q=";
    private String cordHttpPath = "http://api.openweathermap.org/data/2.5/weather?";
    private String language = "&lang=se";
    private String unit = "&units=metric";

    public Connection() {

    }

    public String getWeather(String city, double longitude, double latitude) {
        HttpURLConnection connection;
        InputStream inputStream;
        if (!city.equals("")) {

            try {
                if (!city.equals("Home")) {
                    connection = (HttpURLConnection) (new URL(httpPath + city + ",se" + apiKey + unit)).openConnection();
                } else {
                    connection = (HttpURLConnection) (new URL(cordHttpPath + "lat=" + latitude +
                            "&lon=" + longitude + apiKey + unit)).openConnection();
                }
                connection.setRequestMethod("GET");
                connection.setDoInput(true);
                connection.setDoOutput(true);
                connection.connect();

                StringBuilder buffer = new StringBuilder();
                inputStream = connection.getInputStream();
                BufferedReader bufferedReader = new BufferedReader(new InputStreamReader
                        (inputStream));
                String line;
                while ((line = bufferedReader.readLine()) != null) {
                    buffer.append(line).append("\r\n");
                }
                inputStream.close();
                connection.disconnect();
                return buffer.toString();

            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        } else {
            return null;
        }

    }
}
```

Controller.java
Skriven av: Jimmy Åkesson och Olle Olsson

```
package se.mah.ag7416.p3weather.Activities.Controllers;

import java.util.ArrayList;

import se.mah.ag7416.p3weather.Activities.Activities.FragmentActivity;
import se.mah.ag7416.p3weather.Activities.Fragments.CreateNewFragmentDialog;
import se.mah.ag7416.p3weather.Activities.Fragments.FragmentController;
import se.mah.ag7416.p3weather.Activities.Fragments.WeatherFragment;

public class Controller {


    private FragmentActivity activity;
    private ArrayList<FragmentController> fragmentControllerArrayList= new ArrayList<>();

    public Controller(FragmentActivity fragmentActivity) {
        this.activity = fragmentActivity;
    }

    public void fragmentDialog() {
        CreateNewFragmentDialog dialog = new CreateNewFragmentDialog();
        dialog.show(activity.getFragmentManager(), "");
        dialog.setController(this);
    }

    public void createNewFragment(String city, double Long, double Lat){
        FragmentController fragmentController = new FragmentController(city,activity,Long,Lat);
        fragmentControllerArrayList.add(fragmentController);

    }

    public void removeFragment(WeatherFragment fragment){
        activity.removeFragment(fragment);

    }
}
```

CreateNewFragmentDialog.java
Skriven av: Olle Olsson

```
package se.mah.ag7416.p3weather.Activities.Fragments;

import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DialogFragment;
import android.content.DialogInterface;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import se.mah.ag7416.p3weather.Activities.Controllers.Controller;
import se.mah.ag7416.p3weather.R;


public class CreateNewFragmentDialog extends DialogFragment {

    private EditText editCity;
    private Button search;
    private Button cancel;
    private Controller controller;

    public CreateNewFragmentDialog() {

    }

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        LayoutInflater inflater = getActivity().getLayoutInflater();
        View view = inflater.inflate(R.layout.newdialogfragment, null);
        builder.setView(view);

        initializeComponents(view);
        registerListeners();
        return builder.create();

    }

    public void initializeComponents(View view) {
        editCity = (EditText) view.findViewById(R.id.editCity);
        search = (Button) view.findViewById(R.id.search);
        cancel = (Button) view.findViewById(R.id.cancel);
    }

    public void registerListeners() {
      Listener listener = new Listener();
        search.setOnClickListener(listener);
        cancel.setOnClickListener(listener);
    }

    private class Listener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.search:
                    controller.createNewFragment(editCity.getText().toString(),0,0);
                    Log.d("CreateNewFragmentDialog", "onClick: " + editCity.getText().toString());
                    dismiss();
                    break;
                case R.id.cancel:
                    dismiss();

            }
        }
    }

    public void setController(Controller controller){
        this.controller=controller;
    }
}
```

FragmentActivity.java
Skriven av: Olle Olsson och Jimmy Åkesson

```
package se.mah.ag7416.p3weather.Activities.Activities;

import android.Manifest;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentStatePagerAdapter;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import com.google.android.gms.awareness.state.Weather;
import java.util.ArrayList;
import java.util.HashMap;
import se.mah.ag7416.p3weather.Activities.Controllers.Controller;
import se.mah.ag7416.p3weather.Activities.Fragments.WeatherFragment;
import se.mah.ag7416.p3weather.R;

public class FragmentActivity extends AppCompatActivity implements LocationListener {

    private Controller controller;
    private double longitude, latitude;
    private int numberOfFragments = 0;
    private ViewPager viewPager;
    private PagerAdapter pagerAdapter;
    private ArrayList<WeatherFragment> fragmentList = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fragment);
        getSupportActionBar().hide();
        LocationManager lm = (LocationManager) this.getSystemService(LOCATION_SERVICE);
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        lm.requestSingleUpdate(LocationManager.GPS_PROVIDER, this, null);
        Location loc = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        latitude = loc.getLatitude();
        longitude = loc.getLongitude();
        viewPager = (ViewPager) findViewById(R.id.pager);
        viewPager.setOffscreenPageLimit(100);
        pagerAdapter = new ScreenSlideAdapter(getSupportFragmentManager());
        viewPager.setAdapter(pagerAdapter);
        controller = new Controller(this);
        controller.createNewFragment("Home", longitude, latitude);

        SharedPreferences preferences = getSharedPreferences("save", MODE_PRIVATE);
        if(preferences.contains("numberOfFragments")) {
            int number = preferences.getInt("numberOfFragments", 0);
            Log.d("FragmentActivity", "onCreate: nbrFrag "+number);
            for (int x = 1; x <= number-1; x++) {
                String city = preferences.getString("city" + x, "");
                Log.d("FragmentActivity", "onCreate: city: "+city);
                controller.createNewFragment(city, 0, 0);
            }
        }
    }
```

```
public Controller getController() {
    return controller;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    return true;
}

public void addFragment(WeatherFragment fragment, String tag) {
    numberOfFragments++;
    fragmentList.add(fragment);
    Log.d("FragmentActivity", "addFragment: "+tag);
    pagerAdapter.notifyDataSetChanged();
    viewPager.setCurrentItem(numberOfFragments);

}

public void removeFragment(WeatherFragment fragment) {
    numberOfFragments--;
    int index = fragmentList.indexOf(fragment);
    fragmentList.remove(index);
    Log.d("FragmentActivity", "removeFragment: "+fragment.getCity());
    pagerAdapter.notifyDataSetChanged();
    pagerAdapter.destroyItem(viewPager,index,fragment);

}

@Override
public void onLocationChanged(Location location) {
    latitude = location.getLatitude();
    longitude = location.getLongitude();

    //TODO Updatera hemfragmentet
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {

}

private class ScreenSlideAdapter extends FragmentStatePagerAdapter {
    private FragmentManager fm;


    public ScreenSlideAdapter(FragmentManager fm) {
        super(fm);
        this.fm = fm;
    }

    @Override
    public Fragment getItem(int position) {
        return fragmentList.get(position);
    }

    @Override
    public int getCount() {
        return fragmentList.size();
    }
```

```
        @Override
        public int getItemPosition(Object object) {
            if (fragmentList.contains((WeatherFragment) object)) {
                return fragmentList.indexOf((WeatherFragment) object);
            } else {
                return POSITION_NONE;
            }

        }
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);

        SharedPreferences preferences = getSharedPreferences("save", MODE_PRIVATE);
        SharedPreferences.Editor editor = preferences.edit();
        editor.clear();
        editor.commit();
        if (numberOfFragments > 0) {
            editor.putInt("numberOfFragments", numberOfFragments);
            for (int x = 0; x <= fragmentList.size() - 1; x++) {
                editor.putString("city" + x, fragmentList.get(x).getCity());
            }
            editor.apply();
        }
    }
}
```

FragmentController.java
Skriven av: Jimmy Åkesson, och Linus Forsberg

```
package se.mah.ag7416.p3weather.Activities.Fragments;

import android.util.Log;
import se.mah.ag7416.p3weather.Activities.Activities.FragmentActivity;
import se.mah.ag7416.p3weather.Activities.Controllers.JSONParser;
import se.mah.ag7416.p3weather.Activities.Controllers.Querry;


public class FragmentController {

    private double Long, Lat;
    private String city = "Home";
    private Querry querry;
    private JSONParser parser;
    private WeatherFragment fragment;
    private FragmentActivity fragmentActivity;

    public FragmentController(String cityQuerry, FragmentActivity fragmentActivity, dou
                              double Lat) {
        this.fragmentActivity = fragmentActivity;
        this.city = cityQuerry;
        this.Long = Long;
        this.Lat = Lat;
        fragment = new WeatherFragment();
        fragment.setController(this);
        runQuerry();
        fragmentActivity.addFragment(fragment, city);
    }

    public WeatherFragment getFragment() {
        return fragment;
    }

    public FragmentActivity getActivity() {
        return fragmentActivity;
    }

    public void newFragmentDialog() {
        fragmentActivity.getController().fragmentDialog();
    }

    public void runQuerry() {

        querry = new Querry();
        querry.setCity(city, Long, Lat, this);
        querry.start();
    }

    public void updateParser(JSONParser parser, final boolean firstFragment) {
        this.parser = parser;

        fragmentActivity.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                setValues();
                if(firstFragment) fragment.hideMinusButton();
            }
        });

    }

    private void setValues() {

        fragment.setText(parser.getCity(), parser.getTemp(), parser.getWindspeed(),
                parser.getIcon(), parser.getDescription(), parser.getBackground());
        Log.d("FragmentController ", "setValues: " + parser.getIcon());

    }


}
```

JSONParser.java
Skriven av: Jimmy Åkesson och Linus Forsberg

```
package se.mah.ag7416.p3weather.Activities.Controllers;

import android.util.Log;
import java.text.DecimalFormat;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import se.mah.ag7416.p3weather.R;

public class JSONParser {

    private JSONObject jsonObject;

    public JSONParser(String data) {
        Log.d("JSONParser", "JSONParser: " + data);
        if (data != null) {
            try {
                jsonObject = new JSONObject(data);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }

    public String getTemp() {
        if (jsonObject != null) {
            try {
                JSONObject main = jsonObject.getJSONObject("main");
                String stringTemp = main.getString("temp");
                DecimalFormat df = new DecimalFormat("##.#");
                double roundedTemp = Double.parseDouble(stringTemp);
                return String.valueOf(df.format(roundedTemp));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return null;
    }

    public String getWindspeed() {
        if (jsonObject != null) {
            try {
                JSONObject wind = jsonObject.getJSONObject("wind");
                return wind.getString("speed");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return null;
    }

    public String getCity() {
        if (jsonObject != null) {
            try {
                return jsonObject.getString("name");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
```

```
    public String getDescription() {
        if (jsonObject != null) {
            try {
                JSONArray array = jsonObject.getJSONArray("weather");
                JSONObject description = array.getJSONObject(0);
                return description.getString("description");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return null;

    }

    public int getIcon() {
        if (jsonObject != null) {
            try {
                JSONArray weather = jsonObject.getJSONArray("weather");
                JSONObject w = weather.getJSONObject(0);
                String icon = w.getString("icon");
                if (icon.equals("01d") || icon.equals("01n")) {
                    return R.drawable.sunny;
                } else if (icon.equals("02d") || icon.equals("02n")) {
                    return R.drawable.sunnycloudy;
                } else if (icon.equals("03d") || icon.equals("03n") || icon.equals("04d") || icon
                        .equals("04n")) {
                    return R.drawable.cloudy;
                } else if (icon.equals("09d") || icon.equals("09n") || icon.equals("11d") || icon
                        .equals("11n")) {
                    return R.drawable.rainy;
                } else if (icon.equals("10d") || icon.equals("10n")) {
                    return R.drawable.sunnyrainy;
                } else if (icon.equals("13d") || icon.equals("13n")) {
                    return R.drawable.snowy;
                }else if(icon.equals("50d")|| icon.equals("50n")){
                    return R.drawable.misty;
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return 0;
    }

    public int getBackground() {
        if (jsonObject != null) {
            try {
                JSONArray weather = jsonObject.getJSONArray("weather");
                JSONObject w = weather.getJSONObject(0);
                String icon = w.getString("icon");
                if (icon.equals("01d") || icon.equals("01n")) {
                    return R.drawable.background_sunny;
                } else if (icon.equals("02d") || icon.equals("02n")) {
                    return R.drawable.background_sunnycloudy;
                } else if (icon.equals("03d") || icon.equals("03n") || icon.equals("04d") || icon
                        .equals("04n") || icon.equals("50d")|| icon.equals("50n")) {
                    return R.drawable.background_clouds;
                } else if (icon.equals("09d") || icon.equals("09n") || icon.equals("11d") || icon
                        .equals("11n")) {
                    return R.drawable.background_rainy;
                } else if (icon.equals("10d") || icon.equals("10n")) {
                    return R.drawable.background_sunnyrainy;
                } else if (icon.equals("13d") || icon.equals("13n")) {
                    return R.drawable.background_snowy;
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return 0;
    }

    public String getError() {
        if (jsonObject != null) {
            try {
                return jsonObject.getString("error");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
}
```

LoginActivity.java
Skriven av: Linus Forsberg och Olle Olsson.

```
    private void handleSignInResult(GoogleSignInResult result) {
        Log.d(TAG, "handleSignInResult:" + result.isSuccess());
        if (result.isSuccess()) {
            GoogleSignInAccount acct = result.getSignInAccount();
            updateUI(true);
            startFragmentActivity();
        } else {
            updateUI(false);
        }
    }

    private void startFragmentActivity() {
        Intent intent = new Intent(this, FragmentActivity.class);
        startActivity(intent);
    }

    private void signIn() {
        Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

    private void signOut() {
        Auth.GoogleSignInApi.signOut(mGoogleApiClient).setResultCallback(
                new ResultCallback<Status>() {
                    @Override
                    public void onResult(Status status) {
                        updateUI(false);
                    }
                });
    }

    @Override
    public void onConnectionFailed(ConnectionResult connectionResult) {
        Log.d(TAG, "onConnectionFailed:" + connectionResult);
    }

    private void updateUI(boolean signedIn) {
        if (signedIn) {
            findViewById(R.id.sign_in_button).setVisibility(View.GONE);
        } else {
            Log.d(TAG, "updateUI: Signed in failed");
            findViewById(R.id.sign_in_button).setVisibility(View.VISIBLE);
        }
    }

    @Override
    public void onClick(View v) {
        signOut(); // Signing out automatic sign-in (for display purposes)
        signIn();
    }
}
```

```
package se.mah.ag7416.p3weather.Activities.Activities;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.TextView;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.OptionalPendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
import se.mah.ag7416.p3weather.R;


public class LoginActivity extends AppCompatActivity implements
        GoogleApiClient.OnConnectionFailedListener,
        View.OnClickListener {

    private static final String TAG = "LoginActivity";
    private static final int RC_SIGN_IN = 9001;
    private GoogleApiClient mGoogleApiClient;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        findViewById(R.id.sign_in_button).setOnClickListener(this);
        getSupportActionBar().hide();

        GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions
                .DEFAULT_SIGN_IN)
                .requestEmail()
                .build();

        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .enableAutoManage(this, this)
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        SignInButton signInButton = (SignInButton) findViewById(R.id.sign_in_button);
        signInButton.setSize(SignInButton.SIZE_STANDARD);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == RC_SIGN_IN) {
            GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
            Log.d(TAG, "onActivityResult: " + "CommonStatusCode: " + result.getStatus()
                    .getStatusCode());
            handleSignInResult(result);
        }
    }
```

Querry.java
Skriven av: Jimmy Åkesson

```
package se.mah.ag7416.p3weather.Activities.Controllers;

import android.util.Log;
import se.mah.ag7416.p3weather.Activities.Fragments.FragmentController;


public class Querry extends Thread {

    private String city;
    private double Long, Lat;
    private FragmentController fragmentController;
    private boolean firstFragment = false;

    public void setCity(String city, double Long, double Lat, FragmentControlle
            fragmentController) {
        this.city = city;
        this.fragmentController = fragmentController;
        this.Long = Long;
        this.Lat = Lat;
        if (city.equals("Home")) firstFragment = true;
    }


    @Override
    public void run() {
        boolean running = true;
        int counter =0;

        while (running) {
            counter++;
            String data = "";
            Log.d("Querry", "run: data= "+data);
            Log.d("Querry", "run: antal "+counter);
            data = new Connection().getWeather(city, Long, Lat);
            Log.d("Querry", "run: data after="+data);
            if (!data.equals("")) {
                JSONParser question = new JSONParser(data);
                fragmentController.updateParser(question, firstFragment);
                running = false;
            }
        }
        interrupt();
    }
}
```

WeatherFragment.java

Skriven av: Olle Olsson, Jimmy Åkesson och Linus Forsberg

```
package se.mah.ag7416.p3weather.Activities.Fragments;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AlphaAnimation;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import se.mah.ag7416.p3weather.R;


public class WeatherFragment extends Fragment {

    private ImageView weatherImage;
    private ImageButton plusFragment;
    private ImageButton minusFragment;
    private AlphaAnimation buttonClick = new AlphaAnimation(1F, 0.8F);
    private TextView tvTemperature;
    private TextView tvCity;
    private TextView tvSetTempUnit;
    private TextView tvSetWindspeed;
    private TextView tvTempFormat;
    private int icon;
    private View view;
    private FragmentController fragmentController;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        view = inflater.inflate(R.layout.fragment_weather, container, false);
        initializeComponents(view);
        registerListeners();
        return view;
    }


    public void setController(FragmentController fragmentController) {
        this.fragmentController = fragmentController;
    }

    public void initializeComponents(View view) {
        plusFragment = (ImageButton) view.findViewById(R.id.plusFragment);
        minusFragment = (ImageButton) view.findViewById(R.id.minusFragment);
        weatherImage = (ImageView) view.findViewById(R.id.weatherImage);
        tvTemperature = (TextView) view.findViewById(R.id.tvTemperature);
        tvCity = (TextView) view.findViewById(R.id.tvCity);
        tvSetTempUnit = (TextView) view.findViewById(R.id.tvSetTempUnit);
        tvSetWindspeed = (TextView) view.findViewById(R.id.tvSetWindspeed);
        tvTempFormat = (TextView) view.findViewById(R.id.tvTempFormat);
    }

    public void registerListeners() {
        FragmentListener listener = new FragmentListener();
        plusFragment.setOnClickListener(listener);
        minusFragment.setOnClickListener(listener);
    }
```

```
    public void setText(String city, String temp, String wind, int icon, String description, int background) {
        tvCity.setText(city);
        tvTemperature.setText(temp);
        tvSetWindspeed.setText(wind + " m/s");
        weatherImage.setImageResource(icon);
        tvSetTempUnit.setText(description);
        getView().setBackgroundResource(background);
    }

    public String getCity(){
        return tvCity.getText().toString();
    }

    public void hideMinusButton(){
        minusFragment.setVisibility(View.INVISIBLE);
    }

    private class FragmentListener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.plusFragment:
                    plusFragment.startAnimation(buttonClick);
                    fragmentController.newFragmentDialog();
                    break;
                case R.id.minusFragment:
                    minusFragment.startAnimation(buttonClick);
                    fragmentController.getActivity().getController().removeFragment
                            (fragmentController.getFragment());
                    break;
            }
        }
    }
}
```

Projektmedlemmars insatser

Olle Olsson
Olle har varit ansvarig för utformning av GUI samt varit skrivit delar av OpenWeather och Google sign-in implementationerna. Han har även sammanställt dokumentation.

Jimmy Åkesson
Jimmy har haft den mest drivande rollen under arbetets gång och varit den som skrivit mest kod. Jimmy har haft ansvaret för att få OpenWeatherMap- api:et att fungera samt skrivit uppkoppling mot server, controllers och mycket annat.

Linus Forsberg
Linus ansvarade för att få Google-inloggningen att fungera och samarbetade med Jimmy i arbetet med JSON-hanteringen. Linus har även hjälpt till med arbetet av GUI, sammanställt dokumentation samt klassdiagram.