KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

T120B169 App Development for Smart Mobile Systems

Individual task

[App] Find a Hitchhiker (App for hitchhikers people)

Did:
Andrius Malakauskas, IFF-6/15 gr.

Instructor:
prof. Rytis Maskeliūnas

KAUNAS, 2018

CONTENT

Content....2
Description....3
moqup....4
1. Login function....8
2. Registration function....11
3. Map function....15
4. mENIU....16
5. Locations....18
6. Directions....20
Literature list....23

DESCRIPTION

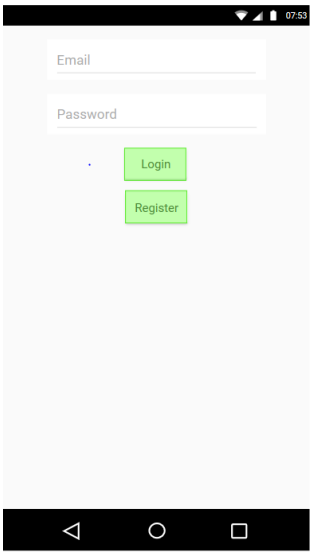
Find a HitchHiker is app for Hitchhikers who wants find destinations easily. You just need to login and plan your routes. In this app you can:

- save destinations
- Find destination
- App automatically finds fastest route
- Find your location

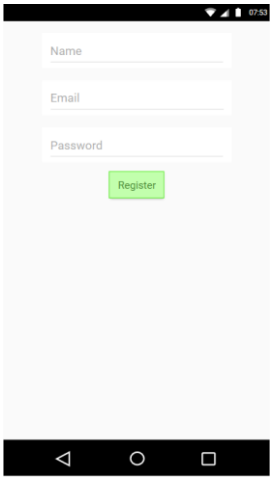
Moqup:
https://app.moqups.com/andrius.malakauskas.darbas@gmail.com/qVGN8Bf5Tw/view

MOQUP

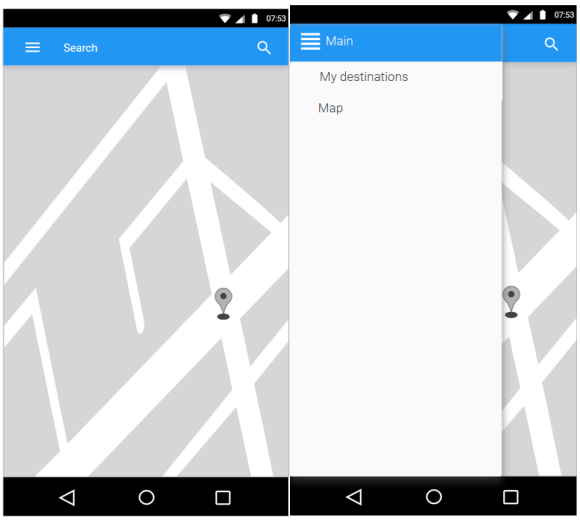
This is a login frame where you can login to app or register(to login you need to enter registered email address and password):



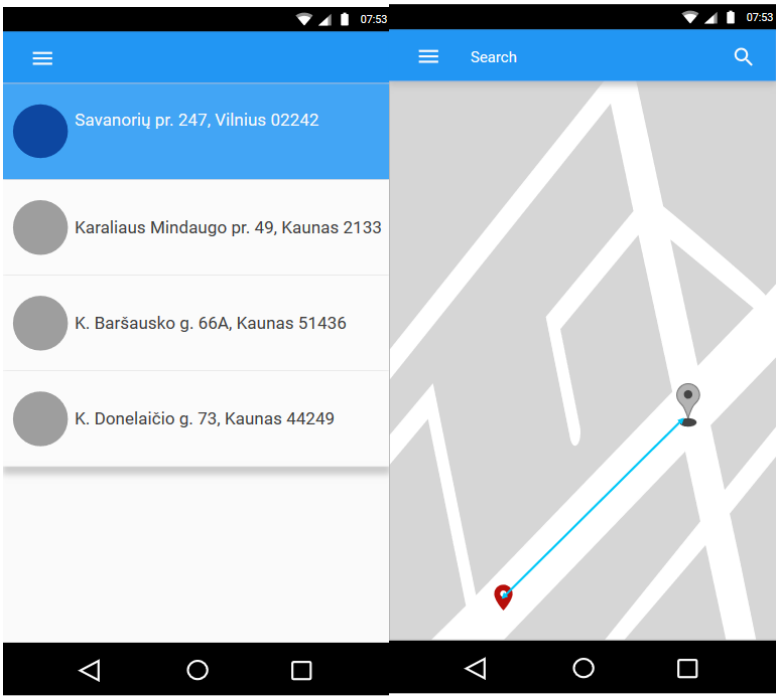
In the registration frame you can register to our system (by entering name, email and password) all data automatically will be saved in local database, so you dont have to register every time:



The main frame in app is map with search bar (to search a location) and meniu bar where are two frames. In „My destinations“ window you can save destinations that after you close app it will be saved in local database. Map window is for searching and showing routes. :

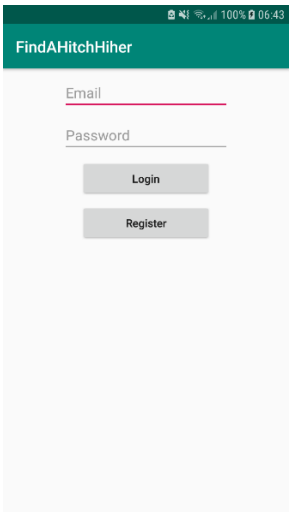


When you enter your destination or select from your saved destinations you get screen with directions from your location to your chosen destination:



1. LOGIN FUNCTION

When you open the app, first you have to login (if you are registered) to enter map. If you are not registered, press „Register“ button and opens registration form and then try again to login.



img. 1 Login screen

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".Login_Activity">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/login_email"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="Email"
        android:inputType="textPersonName|textCapWords"
        android:singleLine="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginLeft="8dp"/>

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/login_password"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:singleLine="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/login_email"
        android:layout_marginLeft="8dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button_login"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
```

```
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:text="Login"
    android:inputType="textPersonName|textCapWords"
    android:singleLine="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login_password"
    android:layout_marginLeft="8dp"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button_registration"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:text="Register"
    android:inputType="textPersonName|textCapWords"
    android:singleLine="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_login"
    android:layout_marginLeft="8dp"/>

</android.support.constraint.ConstraintLayout>
```

Img. 2 activity_login.xml

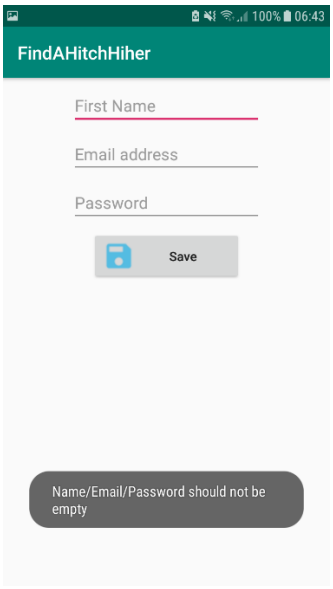
```
package com.example.admin.findahitchhiher;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.util.List;

public class Login_Activity extends AppCompatActivity {

  private Button login;
  private Button register;
  private EditText email;
  private EditText password;
  private Context context = this;
  private AppDatabase mDb;


  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login_);
    mDb = AppActivity.getDatabase();

    login = (Button) findViewById(R.id.button_login);
    register = (Button) findViewById(R.id.button_registration);

    email = (EditText) findViewById(R.id.login_email);
    password = (EditText) findViewById(R.id.login_password);

    register.setOnClickListener(startLoginActivity);
    login.setOnClickListener(openMapsActivity);
  }
```

```
    View.OnClickListener startLoginActivity = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(context, MainActivity.class);
            context.startActivity(intent);
        }
    };
    View.OnClickListener openMapsActivity = new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String passwordText = password.getText().toString();
            String emailText = email.getText().toString();


            List<Person> personList = mDb.personDAO().getAllPersons();
            for (Person person : personList){
                System.out.println(person.getName() + "|||" + person.getSurname() + "|||" + person.getPassword());
                if(passwordText.equals(person.getPassword()) && emailText.equals(person.getSurname())){
                    Intent intent = new Intent(context, MapsActivity.class);
                    context.startActivity(intent);
                }
            }

        }
    };
}
```

Img. 3 Login_activity.java

2. REGISTRATION FUNCTION

This functio allows you to register in our system. In the areas you have to fill your name, email and password. If registration is successful automatically opens login screen. (data automatically updates in database)



img. 4 Registration screen

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/edittext_name"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="First Name"
        android:inputType="textPersonName|textCapWords"
        android:singleLine="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginLeft="8dp"/>

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/edittext_surname"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="Email address"
        android:inputType="textPersonName|textCapWords"
        android:singleLine="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edittext_name"
        android:layout_marginLeft="8dp"/>

    <EditText
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/edittext_password"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:singleLine="true"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edittext_surname"
    android:layout_marginLeft="8dp"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:text="Save"
    android:inputType="textPersonName|textCapWords"
    android:singleLine="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edittext_password"
    android:layout_marginLeft="8dp"/>

  <ScrollView
    android:id="@+id/scrollView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:padding="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button">

    <TextView
      android:layout_width="match_parent"
      android:id="@+id/txt_list"
      android:layout_height="wrap_content" />

  </ScrollView>



</android.support.constraint.ConstraintLayout>
```

Img. 5 registration layout

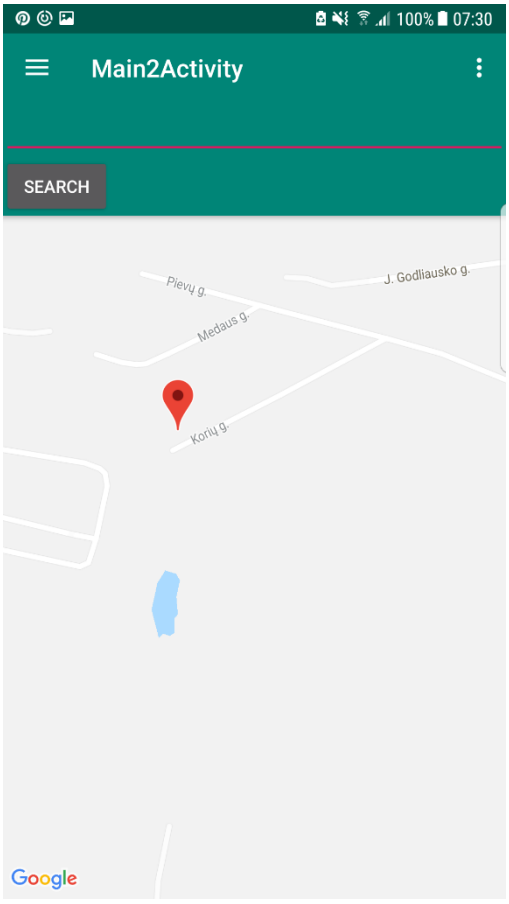
```
package com.example.admin.findahitchhiher;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

public class MainActivity extends AppCompatActivity {

    private AppDatabase mDb;
    private TextView txt_list;
    private Button button;
    private EditText etFirstName;
    private EditText etLastName;
    private EditText etPassword;
    private Context context = this;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mDb = AppActivity.getDatabase();
        txt_list = (TextView) findViewById(R.id.txt_list);

        etFirstName = (EditText) findViewById(R.id.edittext_name);
        etLastName = (EditText) findViewById(R.id.edittext_surname);
        etPassword = (EditText) findViewById(R.id.edittext_password);

        button = (Button) findViewById(R.id.button);
        button.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_save, 0, 0,0);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = etFirstName.getText().toString().trim();
                String surname = etLastName.getText().toString().trim();
                String password = etPassword.getText().toString().trim();

                if(TextUtils.isEmpty(name) || TextUtils.isEmpty(surname) ||
TextUtils.isEmpty(password)){
                    Toast.makeText(getApplicationContext(), "Name/Email/Password should
not be empty", Toast.LENGTH_SHORT).show();
                }else{
                    Person person = new Person();
                    person.setName(name);
                    person.setSurname(surname);
                    person.setPassword(password);
                    mDb.personDAO().insert(person);
                    Toast.makeText(getApplicationContext(), "Registration successful",
Toast.LENGTH_SHORT).show();
                    etFirstName.setText("");
                    etLastName.setText("");
                    etPassword.setText("");
                    etFirstName.requestFocus();
                    getPersonList();
                    Intent intent = new Intent(context, Login_Activity.class);
                    context.startActivity(intent);

                }
```

```
            }
        });

    }

    private void getPersonList() {
        txt_list.setText("");
        List<Person> personList = mDb.personDAO().getAllPersons();
        for (Person person : personList){
            txt_list.append(person.getName() + "|||" + person.getSurname() + "|||" +
person.getPassword());
            txt_list.append("\n");
        }
    }
}
```

Img. 6 registration activity(MainActivity.java)

3. MAP FUNCTION

After login you see google map with your location and marker in Kaunas. In top is search bar where you can search for destinations also app automatically zoom in the screen to the location you want:



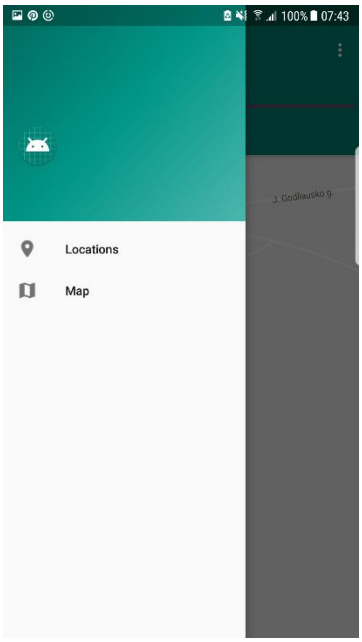
img. 7 Map screen

Im using Geocoder so you could enter a name of the street or other location and it automatically generates latitudes and longitudes.

```
View.OnClickListener searchh = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String location = searchField.getText().toString();
        List<Address> addressList = null;

        if (location != null || !location.equals("")) {

            Geocoder geocoder = new Geocoder(context);
            try {
                addressList = geocoder.getFromLocationName(location, 1);

            } catch (IOException e) {
                e.printStackTrace();
            }

            Address address = addressList.get(0);

            LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());
            mMap.addMarker(new MarkerOptions().position(latLng).title("Marker"));
            mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 18), 5000, null);
        }
    }
};
```

4. MENIU

In meniu are two options: you can see, search and explore map or go to your saved destinations in the future there will be user profile photo, email and name also log out button:



img. 8 meniu.

To create meniu I used android studio template that consist of two pieces: drawer:

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/locations"
            android:icon="@drawable/baseline_place_black_18dp"
            android:title="Locations" />
        <item
            android:id="@+id/map"
            android:icon="@drawable/baseline_map_black_18dp"
            android:title="Map" />
    </group>


</menu>
```

And header where fits profile photo and user information:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/ic_launcher_round" />
```

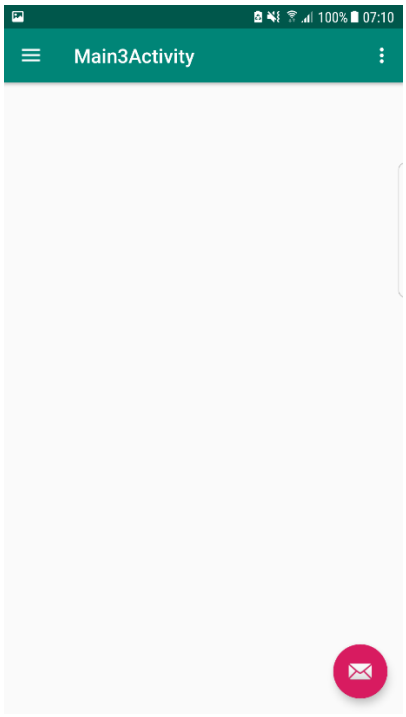
```
    <TextView
        android:id="@+id/contacts_name_surname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/contacts_email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        />


</LinearLayout>
```

And functionality:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main2, menu);

    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.locations) {
        Intent intent = new Intent(context, Main3Activity.class);
        context.startActivity(intent);
    } else if (id == R.id.map) {

    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```

5. LOCATIONS

In this page will be all your saved locations:



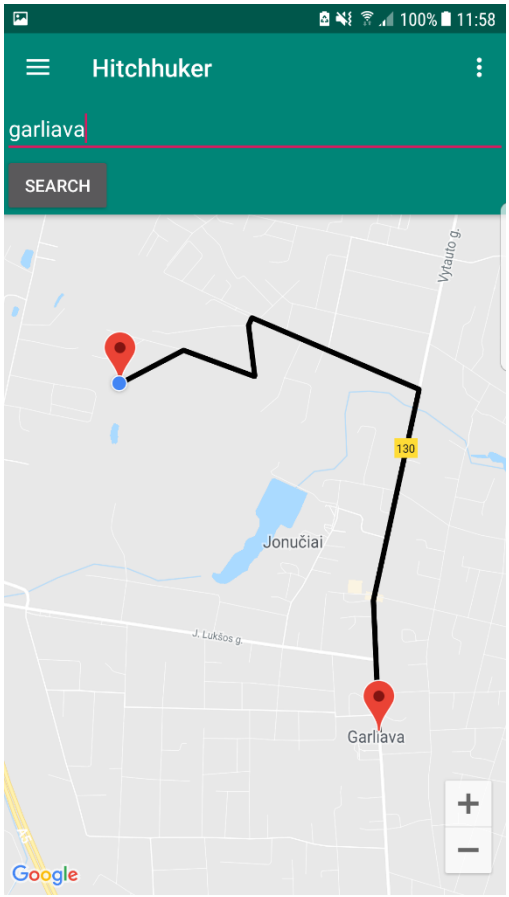
img. 9 locations page.

```
package com.example.admin.findahitchhiher;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class Main3Activity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {


    Context context = this;
    TextView contacts_name;
    TextView contacts_email;
    ListView locations_listt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        Intent intent = getIntent();
        Person current_person = (Person) intent.getSerializableExtra("CurrentPerson");
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);


        //View headerView = navigationView.getHeaderView(0);
      //  contacts_email = (TextView) headerView.findViewById(R.id.contacts_email);
       // contacts_email.setText(current_person.getSurname());
      //  contacts_name = (TextView) headerView.findViewById(R.id.contacts_name_surname);
       // contacts_name.setText(current_person.getName());



    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main2, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.locations) {

        } else if (id == R.id.map) {
            Intent intent = new Intent(context, Main2Activity.class);
            context.startActivity(intent);
        }

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}
```

6. DIRECTIONS

Here you enter destination where you want to go and app automatically show tou fastes way to get there from your current location.



This method changes place latitudes and longitudes to address, that other methods could get more accurate directions:

```
private void finder(LatLng latLng) throws IOException {

    List<Address> addressList;
    Geocoder geocoder = new Geocoder(this, Locale.getDefault());
    addressList = geocoder.getFromLocation(curr.latitude, curr.longitude, 1);
    List<Address> addesss;
    addesss = geocoder.getFromLocation(latLng.latitude, latLng.longitude, 1);

    String pra = addressList.get(0).getAddressLine(0);
    String pab = addesss.get(0).getAddressLine(0);
    Log.e("labas", pra);
    Log.e("labas", pab);

    DirectionsResult results = getDirectionsDetails(pra, pab, TravelMode.WALKING);
    //DirectionsResult results = getDirectionsDetails("K. Baršausko g. 66A, Kaunas 51436",
"Karaliaus Mindaugo pr. 49, Kaunas 44333", TravelMode.WALKING);


    //DirectionsResult results = null;
    if (results != null) {
        addPolyline(results, mMap);

        positionCamera(results.routes[overview], mMap);
        addMarkersToMap(results, mMap);
    } else {
```

```
        Toast.makeText(Main2Activity.this, "Nenuskaitomi duomenys",
Toast.LENGTH_LONG).show();
    }

}
```

This method gets directions information from google directions api

```
private DirectionsResult getDirectionsDetails(String origin, String destination,
TravelMode mode) {
    DateTime now = new DateTime();
    try {
        return DirectionsApi.newRequest(getGeoContext())
                .mode(mode)
                .origin(origin)
                .destination(destination)
                .departureTime(now)
                .await();
    } catch (InterruptedException e) {
        e.printStackTrace();
        return null;
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    } catch (com.google.maps.errors.ApiException e) {
        e.printStackTrace();
        return null;
    }
}
```

this method creates nice looking map:

```
private void setupGoogleMapScreenSettings(GoogleMap mMap) {
    mMap.setBuildingsEnabled(true);
    mMap.setIndoorEnabled(true);
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
!= PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    mMap.setMyLocationEnabled(true);
    //mMap.setTrafficEnabled(true);
    UiSettings mUiSettings = mMap.getUiSettings();
    mUiSettings.setZoomControlsEnabled(true);
    mUiSettings.setCompassEnabled(true);
    mUiSettings.setMyLocationButtonEnabled(true);
    mUiSettings.setScrollGesturesEnabled(true);
    mUiSettings.setZoomGesturesEnabled(true);
    mUiSettings.setTiltGesturesEnabled(true);
    mUiSettings.setRotateGesturesEnabled(true);
}
```

this method adds markers to your current and destination locations

```
private void addMarkersToMap(DirectionsResult results, GoogleMap mMap) {
    mMap.addMarker(new MarkerOptions().position(new
LatLng(results.routes[overview].legs[overview].startLocation.lat,results.routes[overview].
legs[overview].startLocation.lng)).title(results.routes[overview].legs[overview].startAddr
```

```
ess));
    mMap.addMarker(new MarkerOptions().position(new
LatLng(results.routes[overview].legs[overview].endLocation.lat,results.routes[overview].le
gs[overview].endLocation.lng)).title(results.routes[overview].legs[overview].startAddress)
.snippet(getEndLocationTitle(results)));
}
```

This used to change camera position:

```
private void positionCamera(DirectionsRoute route, GoogleMap mMap) {
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new
LatLng(route.legs[overview].startLocation.lat, route.legs[overview].startLocation.lng),
12));

}
```

and this for route paint:

```
private void addPolyline(DirectionsResult results, GoogleMap mMap) {
    List<LatLng> decodedPath =
PolyUtil.decode(results.routes[overview].overviewPolyline.getEncodedPath());
    mMap.addPolyline(new PolylineOptions().addAll(decodedPath));
}
```

LITERATURE LIST

https://moodle.ktu.edu/course/view.php?id=229

https://developers.google.com/maps/documentation