



Mobile App Development 1

Study diary

Jingjing Yang

Contents

1	Week exercises	4
1.1	Android Studio setup.....	4
1.2	My first application	4
1.2.1	Test on emulator.....	4
1.2.2	activity_main.xml	4
1.2.3	MainActivity.java.....	5
1.3	Findings	6
2	Week excercises.....	7
2.1	Written questions and answers	7
2.1.1	Activity	7
2.1.2	Xml and Java.....	7
2.1.3	Localization	7
2.2	BMI App	7
2.2.1	Test on emulator.....	8
2.2.2	activity_main.xml	8
2.2.3	MainActivity.java.....	9
2.3	Use of resource files and localizing your app	10
2.4	Creating GUI layouts with Constraint layout.....	10
2.4.1	Test on emulator.....	11
2.4.2	activity_main.xml	11
2.4.3	MainActivity.java.....	12
3	Week exercises	13
3.1	App GUI with 2 screens	13
3.1.1	Test on emulator.....	13
3.1.2	activity_main.xml	13
3.1.3	MainActivity.java.....	15
3.1.4	activity_forecast.xml	16
3.1.5	ForecastActivity.java.....	17
3.2	Written questions and answers	17
3.2.1	Activity lifecycle	17
3.2.2	Debugging practice.....	18
3.2.3	Intent	21
3.2.4	Bundle	22
4	Week exercises	23
4.1	Written questions and answers	23
4.1.1	Fetching HTTP data	23

4.1.2 Parsing JSON.....	24
4.2 Fetching HTTP data and parsing JSON – practical work	24
4.2.1 Test on emulator.....	25
4.2.2 activity_main.xml	26
4.2.3 MainActivity.java.....	28
5 Week exercises	30
5.1 Written questions and answers	30
5.1.1 Activity lifecycle states.....	30
5.1.2 Activity lifecycle callbacks.....	30
5.1.3 Activity data Bundle	31
5.2 Code to store your activity data.....	31
5.2.1 Test on emulator.....	32
5.2.2 activity_main.xml	33
5.2.3 MainActivity.java.....	34
5.3 Common implicit intents	37
5.3.1 Special permissions.....	37
5.3.2 Test on emulator.....	38
5.3.3 activity_main.xml	39
5.3.4 MainActivity.java.....	40
6 Final project	42
6.1 Description	42
6.2 Test on emulator	43
6.3 activity_main.xml.....	45
6.4 MainActivity.java	47
6.5 activity_todo_list.xml	50
6.6 todoListActivity.java.....	51
6.7 strings.xml.....	53
Sources used with exercises.....	54

1 Week exercises

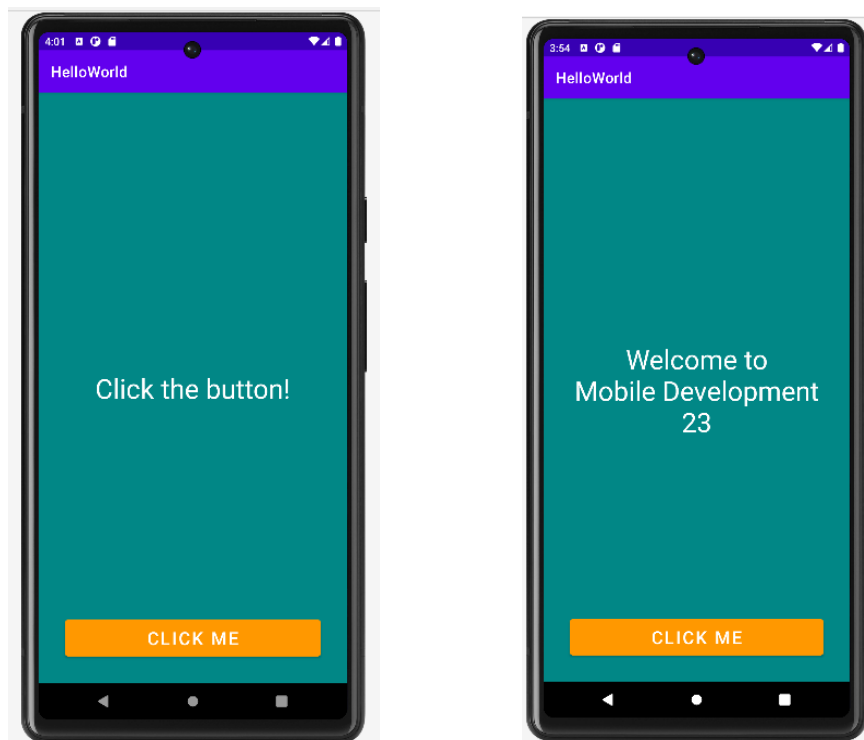
1.1 Android Studio setup

I installed and configured Android Studio

1.2 My first application

There is a button "CLICK ME" and text "Click the button!" on the screen. When clicking the button, the text changes to "Welcome to Mobile Development" plus the times that button has been pressed.

1.2.1 Test on emulator



1.2.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="@color/design_default_color_secondary_variant"
    tools:context=".MainActivity">
```

```

<TextView
    android:id="@+id/tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click the button!"
    android:textAlignment="center"
    android:textColor="@color/white"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/bt"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:layout_marginStart="35dp"
    android:layout_marginEnd="35dp"
    android:layout_marginBottom="30dp"
    android:backgroundTint="#FF9800"
    android:onClick="changeText"
    android:text="Click me"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

1.2.3 MainActivity.java

```

package com.example.helloworld;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private int counter = 0;

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

```

```
public void changeText(View view) {  
    TextView tv = findViewById(R.id.tv);  
    tv.setText("Welcome to\nMobile Development\n" + (counter++));  
}
```

1.3 Findings

Use the attribute “backgroundTint” to set the background color of a Button.

Use `findViewById(R.id.tv)` to get access to a specific component.

2 Week exercises

2.1 Written questions and answers

2.1.1 Activity

What is an activity and what is the role of XML and Java files when implementing an GUI?

An activity provides the window in which the app draws its UI. Generally, one activity implements one screen in an app.

XML files declare layouts (look of the app) and Java files provide logic (what things the app does).

2.1.2 Xml and Java

What advantages you see in defining user interfaces in UI resource files (.xml) instead of coding the UI (e.g. with Java)?

XML makes it easier to understand complex layouts and style UI elements/blocks.

XML can be reused in different widgets and previewed.

XML is easier to change independently and support. If logic doesn't change, Java is not edited. Just XML.

2.1.3 Localization

In Android studio, how localization and language variants are created for an Android project?

By creating locale directories and resource files and using the resources in the Android project.

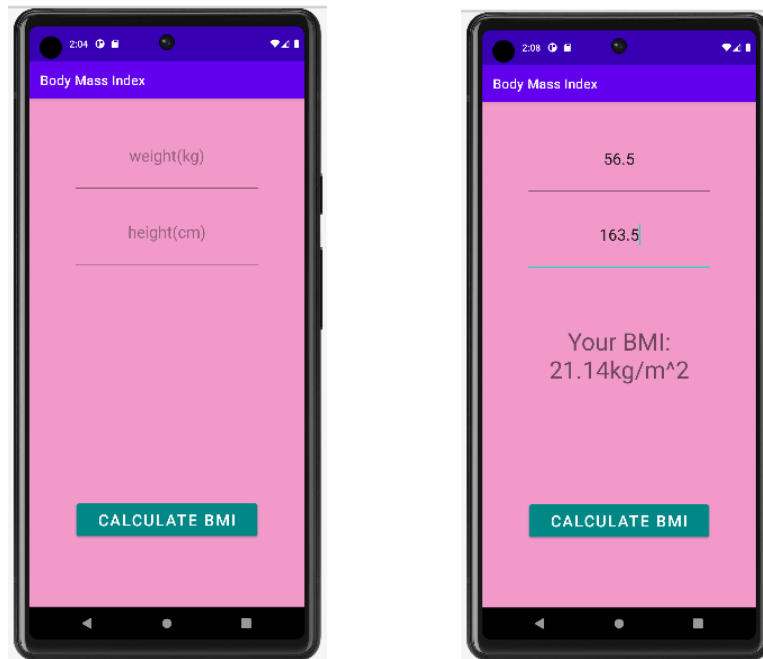
Open the strings.xml file located under the res/values folder. Type the text you want to add multilanguage support, like "Hello World". Do not manually enter the text part, instead enter the text you wrote in `strings.xml`. Like `android:text="@string/helloworld"` for example.

Open the Design part in activity.xml file. Click the `Edit Translations`. Click the Add Locale. Select the language you want to add. Write the equivalent of your text in the TextView in the language of your choice.

2.2 BMI App

The user inputs his/her height and weight and the app calculates and shows the body mass index, when the user clicks a button on the screen.

2.2.1 Test on emulator



2.2.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="@color/pink"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/weightEt"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="65dp"
        android:layout_marginTop="28dp"
        android:layout_marginEnd="65dp"
        android:hint="@string/weight_kg"
        android:inputType="numberDecimal"
        android:selectAllOnFocus="false"
        android:textAlignment="center"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/heightEt"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/heightEt"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="65dp"
        android:layout_marginEnd="65dp"
        android:layout_marginBottom="503dp"
        android:hint="@string/height_cm"
        android:inputType="numberDecimal"
        android:selectAllOnFocus="true"
        android:textAlignment="center"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/calculateBmi"
        android:layout_width="150dp"
        android:layout_height="40dp"
        android:layout_margin="20dp"
        android:text="CALCULATE BMI"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



```

        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/weightEt" />

<TextView
    android:id="@+id/tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/bt"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:layout_marginStart="70dp"
    android:layout_marginEnd="70dp"
    android:layout_marginBottom="100dp"
    android:backgroundTint="@color/teal_700"
    android:onClick="getBMI"
    android:text="@string/calculate_bmi"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2.2.3 MainActivity.java

```

package com.example.helloworld;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.EditText;

import com.example.clayout.R;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void getBMI(View view) {
        EditText editTextWeight = findViewById(R.id.weightEt);
        String weightAsString = editTextWeight.getText().toString();
        float weightInKgs = Float.valueOf(weightAsString);
        EditText editTextHeight = findViewById(R.id.heightEt);
        String heightAsString = editTextHeight.getText().toString();
        float heightInCms = Float.valueOf(heightAsString);
        // BMI = kg/m2
        float bmi = weightInKgs / ((heightInCms / 100f) * (heightInCms / 100f));
    }
}

```

```

String result = String.format("%.2f", bmi);
TextView tv = findViewById(R.id.tv);
tv.setText(getString(R.string.Your_BMI) + ":\n" + result +
             "kg/m^2");
    }
}

```

2.3 Use of resource files and localizing your app

- Remove all hard coded strings from your resource files (.xml) and extract the strings where they should be (i.e. strings.xml).

```

<resources>
    <string name="app_name">Body Mass Index</string>
    <string name="weight_kg">weight (kg)</string>
    <string name="height_cm">height (cm)</string>
    <string name="calculate_bmi">Calculate BMI</string>
    <string name="Your_BMI">Your BMI</string>
</resources>

```

- Implement localization and language support for the BMI application you developed last week.

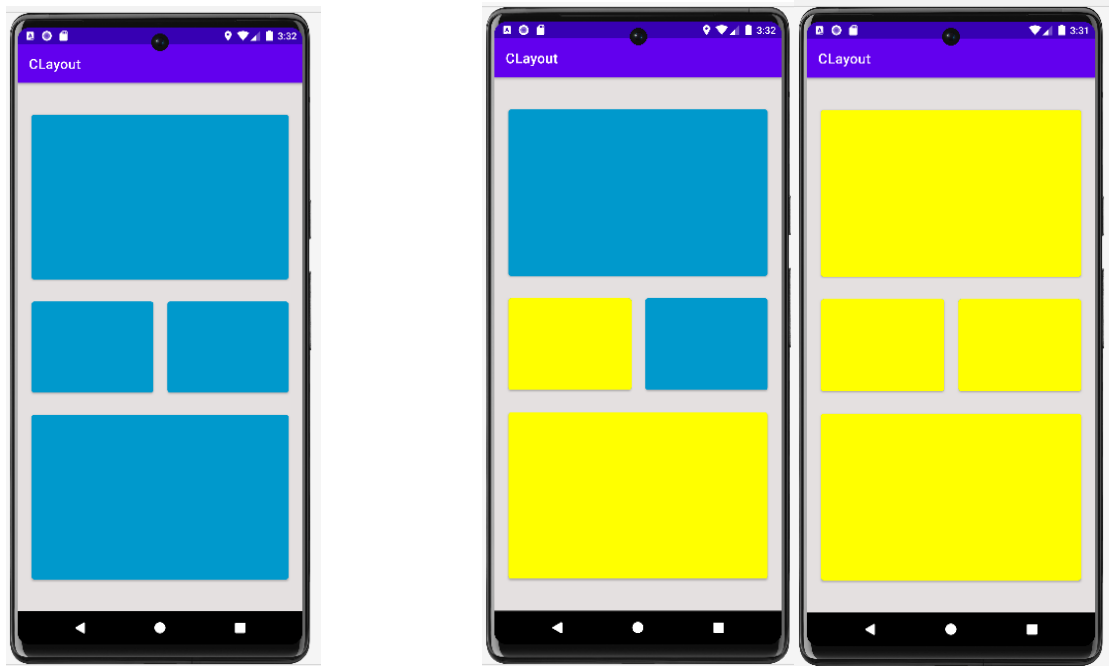
I add support for Chinese, Finnish and Japanese.

Key	Resource Folder	Untranslatable	Default Value	Chinese (zh) i...	Finnish (fi)	Japanese (ja)
app_name	app/src/main/res	<input type="checkbox"/>	Body Mass Index	身体质量指数	Painoindeksi	ボディ・マス・インデックス
weight_kg	app/src/main/res	<input type="checkbox"/>	weight(kg)	体重 (千克)	Paino (kg)	重量(kg)
height_cm	app/src/main/res	<input type="checkbox"/>	height(cm)	身高 (厘米)	korkeus (cm)	高さ(cm)
calculate_bmi	app/src/main/res	<input type="checkbox"/>	Calculate BMI	计算BMI	Laske BMI	BMIを計算する
Your_BMI	app/src/main/res	<input type="checkbox"/>	Your BMI	您的BMI	Sinun BMI	あなたのBMI

2.4 Creating GUI layouts with Constraint layout

- Create the following layout by using Android Designer. The elements below (blue) can be e.g. buttons.
- Optional extra: Implement the elements so that when clicked, they turn their color to Yellow.

2.4.1 Test on emulator



2.4.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    android:background="#E4E0E0"
    tools:context=".MainActivity">

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.5" />

    <Button
        android:id="@+id/button1"
        android:layout_width="0dp"
        android:layout_height="250dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="20dp"
        android:backgroundTint="@android:color/holo_blue_dark"
        android:onClick="changeColor"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
```

```

        android:layout_marginEnd="10dp"
        android:layout_marginBottom="20dp"
        android:backgroundTint="@android:color/holo_blue_dark"
        android:onClick="changeColor"
        app:layout_constraintBottom_toTopOf="@+id/button4"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button1" />

<Button
    android:id="@+id/button3"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="20dp"
    android:backgroundTint="@android:color/holo_blue_dark"
    android:onClick="changeColor"
    app:layout_constraintBottom_toTopOf="@+id/button4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/guideline"
    app:layout_constraintTop_toBottomOf="@+id/button1" />

<Button
    android:id="@+id/button4"
    android:layout_width="0dp"
    android:layout_height="250dp"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="40dp"
    android:backgroundTint="@android:color/holo_blue_dark"
    android:onClick="changeColor"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2.4.3 MainActivity.java

```

package com.example.clayout;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void changeColor(View view) {
        view.setBackgroundColor(Color.parseColor("#ffff00"));
    }
}

```

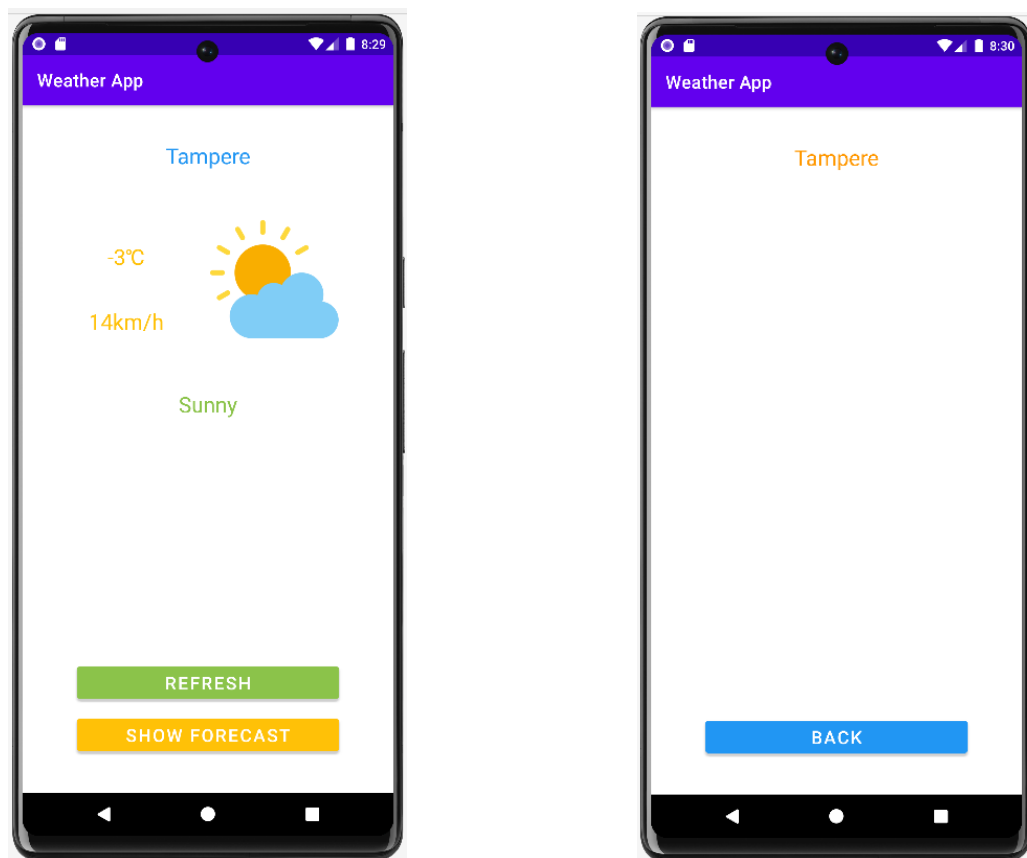
3 Week exercises

3.1 App GUI with 2 screens

1st screen shows weather conditions like temperature, wind speed and overall weather; 2nd screen for weather forecast; The 1st screen contains a button, which upon clicking, opens the 2nd screen with an intent.

Some data is sent to the 2nd screen via the Intent from the 1st screen and is shown in some TextView in the 2nd screen.

3.1.1 Test on emulator



3.1.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
```

```

        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="50dp"
        android:text="@string/tampere"
        android:textAlignment="center"
        android:textColor="#2196F3"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

```

<TextView
    android:id="@+id/textView2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="20dp"
    android:text="-3\u2103"
    android:textAlignment="center"
    android:textColor="#FFC107"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id/imageView2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

```

```

<TextView
    android:id="@+id/textView3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="20dp"
    android:text="14km/h"
    android:textAlignment="center"
    android:textColor="#FFC107"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id/imageView2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

```

```

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="200dp"
    android:layout_marginTop="50dp"
    android:layout_marginEnd="50dp"
    android:layout_marginBottom="350dp"
    android:scaleType="fitCenter"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    android:src="@drawable/sunny" />

```

```

<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="50dp"
    android:text="@string/sunny"

```

```

        android:textAlignment="center"
        android:textColor="#8BC34A"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />

<Button
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="60dp"
    android:layout_marginEnd="60dp"
    android:layout_marginBottom="10dp"
    android:backgroundTint="#8BC34A"
    android:onClick="refreshData"
    android:text="@string/refresh"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/button2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="60dp"
    android:layout_marginEnd="60dp"
    android:layout_marginBottom="40dp"
    android:backgroundTint="#FFC107"
    android:onClick="openForecastActivity"
    android:text="@string/show_forecast"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.1.3 MainActivity.java

```

package com.example.weatherapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); //Activity is being created
        setContentView(R.layout.activity_main);
    }

    // Activity lifecycle methods
    protected void onStart() {
        super.onStart();
        Log.d("Weather_APP", "MainActivity:onStartCalled"); // Activity
        is about to become visible
    }
}

```

```

        protected void onResume() {
            super.onResume();
            Log.d("Weather_APP", "MainActivity:onResumeCalled");// Activity
            ity has become visible
        }

        protected void onPause() {
            super.onPause();
            Log.d("Weather_APP", "MainActivity:onPauseCalled");// Activity
            is taking focus
        }

        protected void onStop() {
            super.onStop();
            Log.d("Weather_APP", "MainActivity:onStopCalled");// Activity
            is no longer visible
        }

        protected void onDestroy() {
            super.onDestroy();
            Log.d("Weather_APP", "MainActivity:onDestroyCalled");// Activity
            ity is about to be destroyed
        }

        public void openForecastActivity(View view) {
            Intent openForecast = new Intent(this, ForecastActiv-
            ity.class);
            openForecast.putExtra("LOCATION", "Tampere");
            openForecast.putExtra("HOW_MANY_DAYS", 5);
            startActivity(openForecast);
        }

        public void refreshData(View view) {
        }
    }
}

```

3.1.4 activity_forecast.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:an-
droid="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=".ForecastActivity">

    <TextView
        android:id="@+id/forestHeadertextView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="50dp"
        android:text="TextView"
        android:textAlignment="center"
        android:textColor="#FF9800"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```



```

<Button
    android:id="@+id/button3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="60dp"
    android:layout_marginEnd="60dp"
    android:layout_marginBottom="40dp"
    android:backgroundTint="#2196F3"
    android:onClick="openMainActivity"
    android:text="Back"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.1.5 ForecastActivity.java

```

package com.example.weatherapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class ForecastActivity extends AppCompatActivity {

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

        Intent intent = getIntent();
        String location = intent.getStringExtra("LOCATION");
        String somethingNotThere = intent.getStringExtra("NOT_THERE");
        int howManyDays = intent.getIntExtra("HOW_MANY_DAYS", 7);

        TextView forecastHeader = findViewById(R.id.forecastHeader-
textView);
        if(location != null){
            forecastHeader.setText(location);
        }
        else{
            forecastHeader.setText(R.string.LOCATION_NOT_KNOWN);
        }
    }

    public void openMainActivity(View view) {
        Intent openMain = new Intent(this, MainActivity.class);
        startActivity(openMain);
    }
}

```

3.2 Written questions and answers

3.2.1 Activity lifecycle

Explain what is an Activity and also explain the activity lifecycle.

An Activity is an application component that provides a screen with which users can interact in order to do something.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, and `onDestroy()`. The system invokes each of these callbacks as an activity enters a new state.

3.2.2 Debugging practice

Practical debugging with an activity: Override and implement all activity lifecycle methods in your activity and debug your application by setting breakpoints to the methods. In which order the methods will be called, when:

• A new incoming call occurs and then ends.

`onPause()`>>`onStop()`

>>`onStart()`>>`onResume()`

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); //Activity is being created
        setContentView(R.layout.activity_main);
        Log.d( tag: "Weather_APP", msg: "MainActivity:onCreateCalled");
    }

    // Activity lifecycle methods
    protected void onStart() {
        super.onStart();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onStartCalled");
    }

    protected void onResume() {
        super.onResume();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onResumeCalled");
    }

    protected void onPause() {
        super.onPause();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onPauseCalled");
    }

    protected void onStop() {
        super.onStop();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onStopCalled");
    }

    protected void onDestroy() {
        super.onDestroy();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onDestroyCalled");
    }
}

```

Logcat x +

16 Pro API 27 (emulator-5554) Android 10, API 27 package:mine

```

----- beginning of main
----- beginning of system
23-03-05 20:05:35.122 4849-4849 Weather_APP com.example.weatherapp
23-03-05 20:05:35.462 4849-4889 EGL_emulation com.example.weatherapp
23-03-05 20:05:36.231 4849-4849 Weather_APP com.example.weatherapp
D MainActivity:onPauseCalled
D eglMakeCurrent: 0xe8469980: ver 2 0 (t
D MainActivity:onStopCalled

```

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); //Activity is being created
        setContentView(R.layout.activity_main);
        Log.d( tag: "Weather_APP", msg: "MainActivity:onCreateCalled");
    }

    // Activity lifecycle methods
    protected void onStart() {
        super.onStart();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onStartCalled");
    }

    protected void onResume() {
        super.onResume();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onResumeCalled");
    }

    protected void onPause() {
        super.onPause();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onPauseCalled");
    }

    protected void onStop() {
        super.onStop();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onStopCalled");
    }

    protected void onDestroy() {
        super.onDestroy();
        Log.d( tag: "Weather_APP", msg: "MainActivity:onDestroyCalled");
    }
}

```

Logcat x +

I 6 Pro API 27 (emulator-5554) Android 0, API 27 package:mine

```

----- beginning of main
----- beginning of system
23-03-05 20:05:35.122 4849-4849 Weather_APP com.example.weatherapp D MainActivity:onPauseCalled
23-03-05 20:05:35.462 4849-4889 EGL_emulation com.example.weatherapp D eglMakeCurrent: 0xe8469980: ver 2 0 (t
23-03-05 20:05:36.231 4849-4849 Weather_APP com.example.weatherapp D MainActivity:onStopCalled
23-03-05 20:06:12.326 4849-4849 Weather_APP com.example.weatherapp D MainActivity:onStartCalled
23-03-05 20:06:12.326 4849-4849 Weather_APP com.example.weatherapp D MainActivity:onResumeCalled
23-03-05 20:06:12.380 4849-4889 EGL_emulation com.example.weatherapp D eglMakeCurrent: 0xe8469980: ver 2 0 (t

```

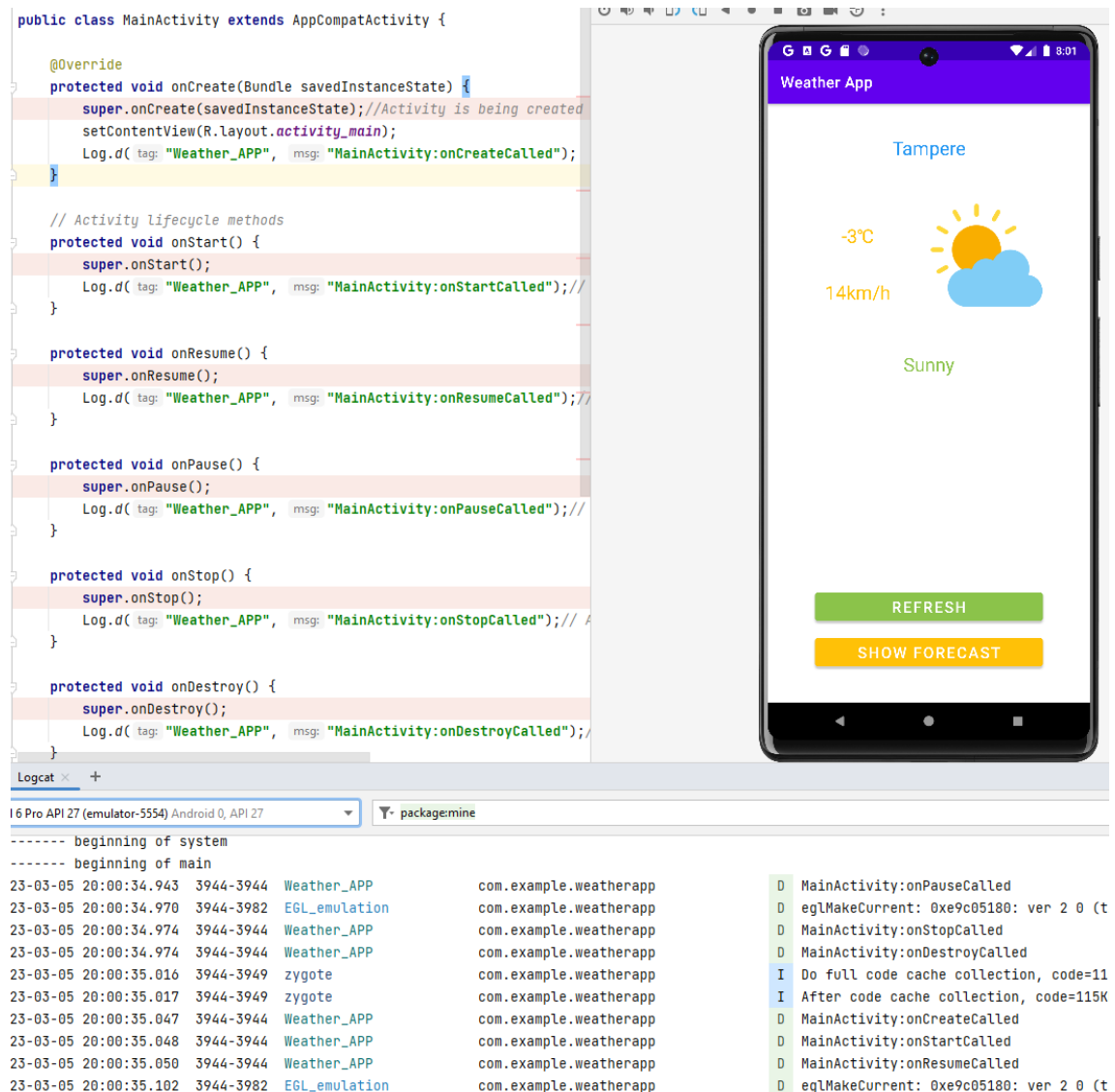
• The display orientation is changed from portrait to landscape and vice versa.

Both are: `onPause()`>>`onStop()`>>`onDestroy()`

>>onCreate()>>onStart()>>onResume()

The screenshot displays the Android Studio environment. On the left, the `MainActivity.java` file is open, showing the lifecycle methods: `onCreate`, `onStart`, `onResume`, `onPause`, `onStop`, and `onDestroy`. Each method is annotated with a log statement using the tag `"Weather_APP"` and the message `"MainActivity:methodNameCalled"`. The `onCreate` method sets the content view to `R.layout.activity_main`. The `onStart` and `onResume` methods are currently selected. In the center, a virtual device (tablet) displays the "Weather App" interface. The app shows the location "Tampere", a temperature of "-3°C", a wind speed of "14km/h", and two buttons: "REFRESH" (green) and "SHOW FORECAST" (orange). On the right, the Logcat window shows the sequence of log messages corresponding to the lifecycle methods being executed. The messages are as follows:

Time	Process	Package	Message
123-03-05 20:02:17.502	Weather_APP	com.example.weatherapp	D MainActivity: onPauseCalled
123-03-05 20:02:17.504	EGL_emulation	com.example.weatherapp	D eglMakeCurrent: 0xe9c05180: ver 2 0 (tinfo 0xe9c032e0)
123-03-05 20:02:17.512	Weather_APP	com.example.weatherapp	D MainActivity: onStopCalled
123-03-05 20:02:17.513	Weather_APP	com.example.weatherapp	D MainActivity: onDestroyCalled
123-03-05 20:02:17.599	Weather_APP	com.example.weatherapp	D MainActivity: onCreateCalled
123-03-05 20:02:17.600	Weather_APP	com.example.weatherapp	D MainActivity: onStartCalled
123-03-05 20:02:17.602	Weather_APP	com.example.weatherapp	D MainActivity: onResumeCalled



3.2.3 Intent

What is an Intent in Android?

All Android activities are started or activated with an intent. Intents are message objects that make a request to the Android runtime to start an activity or other app component in some app.

How are intents used to switch between application screens? How do you pass data between the activities with intents? Explain the procedure e.g. with the 1st exercise (2 screens).

In the first (sending) activity: Create the Intent object>>Put data or extras into that intent>>Start the new activity with `startActivity()`.

In the second (receiving) activity: Get the intent object the activity was started with.>>Retrieve the data or extras from the Intent object.

What kind of data you can pass through the intent.

We can pass data as a key value pair or Bundle data object through the intent. The data to operate on, such as a person record in the contacts database, expressed as a Uri.

3.2.4 Bundle

Below is an activity onCreate method getting savedInstanceState Bundle. Explain what it is and where it is used?

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

Android platform allows the users to save any instance state. Instance state can be saved in the Bundle. Bundle is passed as argument to the onSaveInstanceState method.

If you save the state of the application in a bundle, it can be passed back to onCreate if the activity needs to be recreated (e.g., orientation change) so that you don't lose this prior information. If no data was supplied, savedInstanceState is null.

4 Week exercises

4.1 Written questions and answers

4.1.1 Fetching HTTP data

Explain options and procedure when you need to get REST data from a HTTP server. What are the options for that in Android?

In Android, tools such as Volley and Retrofit allow you to connect to APIs seamlessly.

Before getting started with Volley, one needs to import Volley and add permissions in the Android Project.

Open build.gradle(Module: app) and add the following dependency:

```
dependencies {
    //...
    implementation 'com.android.volley:volley:1.0.0'
}
```

In AndroidManifest.xml add the internet permission:

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

How is fetching done with Android Volley Library?

Volley has two main classes:

Request Queue: It is the interest one uses for dispatching requests to the network. One can make a request queue on demand if required, but typically it is created early on, at startup time, and keep it around and use it as a Singleton.

Request: All the necessary information for making web API call is stored in it. It is the base for creating network requests(GET, POST).

1) // Instantiate the RequestQueue.

```
RequestQueue queue = Volley.newRequestQueue(this);
String url = "https://www.google.com";
```

2) // Request a string response from the provided URL.

```
StringRequest stringRequest = new StringRequest(Request.Method.GET,
url,
    response -> {
        parseJsonAndUpdateUI(response);
    }, error -> {
    });
```

3) // Add the request to the RequestQueue.

```
queue.add(stringRequest);
```

4.1.2 Parsing JSON

How is JSON parsed in Android/Java. Explain the API for that.

Android provides four different classes to manipulate JSON data. These classes are `JSONArray`, `JSONObject`, `JSONStringer` and `JSONTokenizer`.

The first step is to identify the fields in the JSON data in which you need.

For parsing a JSON object, we will create an object of class `JSONObject` and specify a string containing JSON data to it. Its syntax is –

```
JSONObject objectName = new JSONObject(in);
```

The last step is to parse the JSON. A JSON file consists of different object with different key/value pair. So `JSONObject` has a separate function for parsing each of the component of JSON file. Its syntax is –

```
name = USER.getString("name");
```

The method `getString(String name)` returns the string value of the specified key.

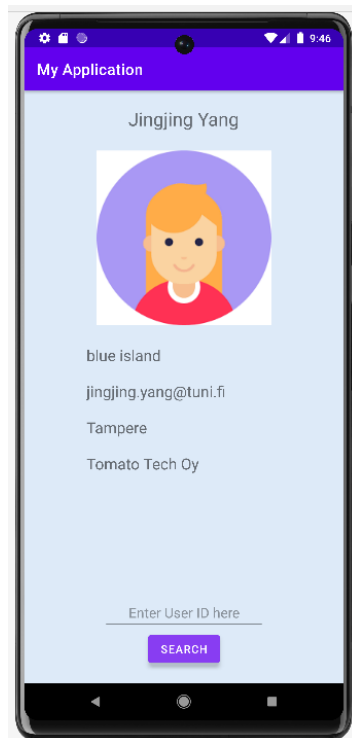
There are other methods provided by this class –

1. `get(String name)` returns the value but in the form of Object type.
2. `getBoolean(String name)` returns the boolean value specified by the key.
3. `getDouble(String name)` returns the double value specified by the key.
4. `getInt(String name)` returns the integer value specified by the key.
5. `getLong(String name)` returns the long value specified by the key.
6. `length()` returns the number of name/value mappings in this object.
7. `names()` returns an array containing the string names in this object.

4.2 Fetching HTTP data and parsing JSON – practical work

This is a simple Android GUI about user information. It fetches some HTTP/JSON data (name, username, email, city, phone) from server and parse the data in GUI elements for the Activity. The REST API is from <https://jsonplaceholder.typicode.com/>

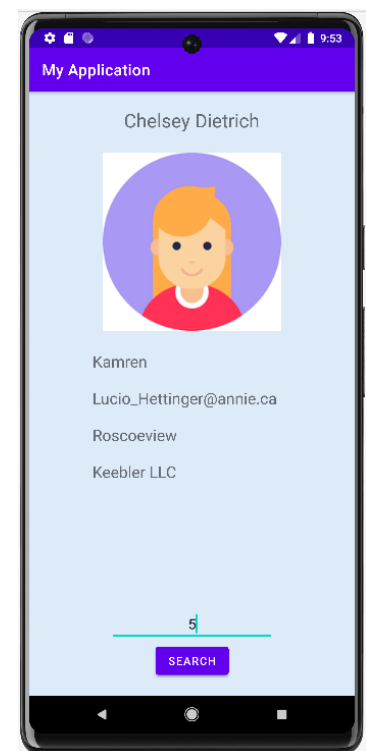
4.2.1 Test on emulator



```

{
  "id": 5,
  "name": "Chelsey Dietrich",
  "username": "Kamren",
  "email": "Lucio_Hettinger@annie.ca",
  "address": {
    "street": "Skiles Walks",
    "suite": "Suite 351",
    "city": "Roscoeview",
    "zipcode": "33263",
    "geo": {
      "lat": "-31.8129",
      "lng": "62.5342"
    }
  },
  "phone": "(254)954-1289",
  "website": "demarco.info",
  "company": {
    "name": "Keebler LLC",
    "catchPhrase": "User-centric fault-tolerant solution",
    "bs": "revolutionize end-to-end systems"
  }
},

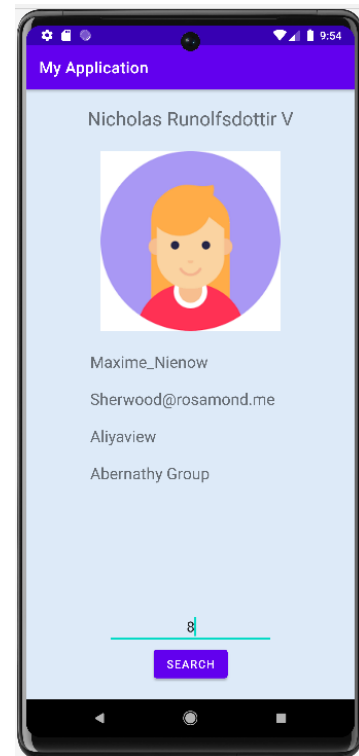
```



```

{
  "id": 8,
  "name": "Nicholas Runolfsson V",
  "username": "Maxime_Nienow",
  "email": "Sherwood@rosamond.me",
  "address": {
    "street": "Ellsworth Summit",
    "suite": "Suite 729",
    "city": "Aliyaview",
    "zipcode": "45169",
    "geo": {
      "lat": "-14.3990",
      "lng": "-120.7677"
    }
  },
  "phone": "586.493.6943 x140",
  "website": "jacynthe.com",
  "company": {
    "name": "Abernathy Group",
    "catchPhrase": "Implemented secondary concept",
    "bs": "e-enable extensible e-tailers"
  }
},

```



4.2.2 activity_main.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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"
    android:background="#DDEAF8"
    android:textAlignment="center"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/nameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:text="Jingjing Yang"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nameTextView"
        app:srcCompat="@drawable/user_female" />

```

```

<TextView
    android:id="@+id/usernameTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="80dp"
    android:layout_marginTop="25dp"
    android:text="blue island"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

<TextView
    android:id="@+id/emailTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="jingjing.yang@tuni.fi"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="@+id/usernameTextView"
    app:layout_constraintTop_toBottomOf="@+id/usernameTextView" />

<TextView
    android:id="@+id/cityTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Tampere"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="@+id/emailTextView"
    app:layout_constraintTop_toBottomOf="@+id/emailTextView" />

<TextView
    android:id="@+id/companyTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Tomato Tech Oy"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="@+id/cityTextView"
    app:layout_constraintTop_toBottomOf="@+id/cityTextView" />

<EditText
    android:id="@+id/UserIDeditTextNumber"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="@string/enter_user_id_here"
    android:inputType="number"
    android:textAlignment="center"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:onClick="userInfoUpdate"
    android:text="@string/search"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

4.2.3 MainActivity.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity {

    private RequestQueue queue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Instantiate the RequestQueue.
        queue = Volley.newRequestQueue(this);
    }

    public void userInfoUpdate(View view) {
        String url = "https://jsonplaceholder.typicode.com/users";

        // Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
            response -> {
                Log.d("User_App", response);
                parseJsonAndUpadateUI(response);
            }, error -> {
                Log.d("User_App", error.toString());
            });

        queue.add(stringRequest);
    }

    private void parseJsonAndUpadateUI(String response) {
        EditText editTextUserID = findViewById(R.id.UserIDeditTextNumber);
        String idAsString = editTextUserID.getText().toString();
        int userid = Integer.valueOf(idAsString);

        try {
            JSONArray userResponse = new JSONArray(response);
            JSONObject USER = userResponse.getJSONObject(userid - 1);

            String name, username, email, city, company;
```

```
name = USER.getString("name");
TextView nameTv = findViewById(R.id.nameTextView);
nameTv.setText(name);

username = USER.getString("username");
TextView usernameTv = findViewById(R.id.usernameTextView);
usernameTv.setText(username);

email = USER.getString("email");
TextView emailTv = findViewById(R.id.emailTextView);
emailTv.setText(email);

city = USER.getJSONObject("address").getString("city");
TextView cityTv = findViewById(R.id.cityTextView);
cityTv.setText(city);

company = USER.getJSONObject("company").getString("name");
TextView companyTv = findViewById(R.id.companyTextView);
companyTv.setText(company);

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

}

}
```

5 Week exercises

5.1 Written questions and answers

5.1.1 Activity lifecycle states

Explain Android Activity lifecycle states.

An Activity does not have the control over managing its own state. It just goes through state transitions either due to user interaction or due to system-generated events. There are four states of an Activity in Android:

Active: An Activity is running - created, started or resumed. It is visible to the user and the user is able to interact with it.

Paused: User can still see the Activity in the background such as behind a transparent window or a dialog box i.e it is partially visible.

Stopped: When a new Activity is started on top of the current one or when a user hits the Home key, the activity is brought to Stopped state. The Activity goes to the background and the Home Screen of the device is made visible. Our app doesn't close. It just gets minimized.

Destroyed: When we finally destroy the App, the Activity is out of the memory and it is invisible to the user.

Give an example of each state and when it is entered.

Active: When a user clicks on the App icon, the Main Activity gets started and it creates the App's User Interface using the layout XMLs.

Paused: When any dialog box appears on the screen, like when you press exit on some apps, it shows a box confirming whether you want to exit or not. At that point of time, we are not able to interact with the App's UI until we deal with that dialog box/popup.

Stopped: When we press the Home button while using the app.

Destroyed: When we completely close an App or Android Runtime decides to reclaim the memory allocated to an Activity i.e in the paused or stopped state.

5.1.2 Activity lifecycle callbacks

Explain also activity lifecycle callbacks.

An Android activity goes through six major lifecycle stages or callbacks. These are: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, and `onDestroy()`. The system invokes each of these callbacks as an activity enters a new state.

5.1.3 Activity data Bundle

What is meant by Activity data Bundle.

Android Bundles are generally used for passing data from one activity to another. Bundles are used with intent and values are sent and retrieved in the same fashion, as it is done in the case of Intent.

How do you save activity state (member data of your activity)?

Basically here concept of key-value pair is used where the data that one wants to pass is the value of the map, which can be later retrieved by using the key. It depends on the user what type of values the user wants to pass, but bundles can hold all types of values (int, String, boolean, char) and pass them to the new activity.

What happens with activity lifecycle when you change the orientation of the device from portrait to landscape when your activity is visible and active?

Whenever we change the orientation of the screen i.e from portrait to landscape or vice-versa, lifecycle methods start from the start i.e from onCreate() method. This is because the complete spacing and other visual appearance gets changed and adjusted.

First the Activity will disappear for a millisecond when the onPause(), onStop, and onDestroy() methods are called. After a few milliseconds, the activity will be restarted when the onCreate(), onStart() and onResume() methods are called.

5.2 Code to store your activity data

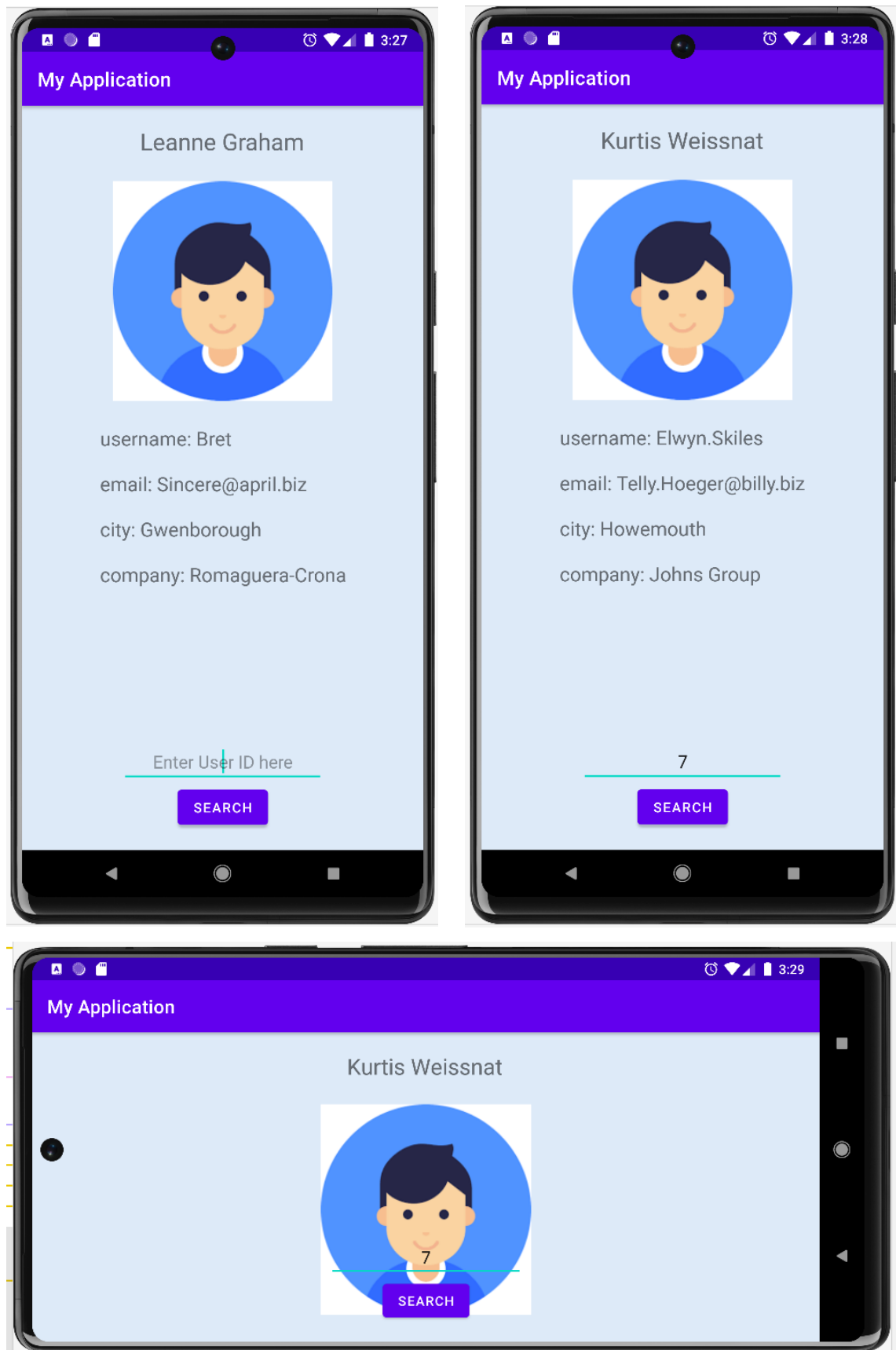
Implement saving/restoring local activity data whenever the device is turned and activity is destroyed and recreated.

Take some specific data e.g. data which you have fetched from the internet..

Implement the following callbacks to prevent your activity to lose data.

- onSaveInstanceState()
- onRestoreInstanceState()

5.2.1 Test on emulator



5.2.2 activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:an-
droid="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="#DDEAF8"
    android:textAlignment="center"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/nameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:text=""
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nameTextView"
        app:srcCompat="@drawable/user_male" />

    <TextView
        android:id="@+id/usernameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="80dp"
        android:layout_marginTop="25dp"
        android:text=""
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <TextView
        android:id="@+id/emailTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text=""
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="@+id/usernameTextView"
        app:layout_constraintTop_toBottomOf="@+id/usernameTextView" />

    <TextView
        android:id="@+id/cityTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text=""

```

```

        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="@+id/emailTextView"
        app:layout_constraintTop_toBottomOf="@+id/emailTextView" />

<TextView
    android:id="@+id/companyTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text=""
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="@+id/cityTextView"
    app:layout_constraintTop_toBottomOf="@+id/cityTextView" />

<EditText
    android:id="@+id/UserIDeditTextNumber"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="@string/enter_user_id_here"
    android:inputType="number"
    android:textAlignment="center"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:onClick="userInfoUpdate"
    android:text="@string/search"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5.2.3 MainActivity.java

```

package com.example.myapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity {

```

```

private RequestQueue queue;
private String name = "Leanne Graham",
        username = "Bret",
        email = "Sincere@april.biz",
        city = "Gwenborough",
        company = "Romaguera-Crona";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Instantiate the RequestQueue.
    queue = Volley.newRequestQueue(this);

    TextView nameTv = findViewById(R.id.nameTextView);
    nameTv.setText(name);

    TextView usernameTv = findViewById(R.id.usernameTextView);
    usernameTv.setText("username: " + username);

    TextView emailTv = findViewById(R.id.emailTextView);
    emailTv.setText("email: " + email);

    TextView cityTv = findViewById(R.id.cityTextView);
    cityTv.setText("city: " + city);

    TextView companyTv = findViewById(R.id.companyTextView);
    companyTv.setText("company: " + company);
}

@Override
protected void onDestroy() {
    super.onDestroy();
}

@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putString("NAME", name);
    outState.putString("USER_NAME", username);
    outState.putString("EMAIL", email);
    outState.putString("CITY", city);
    outState.putString("COMPANY", company);
}

@Override
protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    name = savedInstanceState.getString("NAME");
    username = savedInstanceState.getString("USER_NAME");
    email = savedInstanceState.getString("EMAIL");
    city = savedInstanceState.getString("CITY");
    company = savedInstanceState.getString("COMPANY");

    TextView nameTv = findViewById(R.id.nameTextView);
    nameTv.setText(name);

    TextView usernameTv = findViewById(R.id.usernameTextView);
    usernameTv.setText("username: " + username);

    TextView emailTv = findViewById(R.id.emailTextView);
    emailTv.setText("email: " + email);
}

```

```

        TextView cityTv = findViewById(R.id.cityTextView);
        cityTv.setText("city: " + city);

        TextView companyTv = findViewById(R.id.companyTextView);
        companyTv.setText("company: " + company);
    }

    public void userInfoUpdate(View view) {
        String url = "https://jsonplaceholder.typicode.com/users";

        // Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url, response -> {
            Log.d("User_App", response);
            parseJsonAndUpdateUI(response);
        }, error -> {
            Log.d("User_App", error.toString());
        });

        queue.add(stringRequest);
    }

    private void parseJsonAndUpdateUI(String response) {
        EditText editTextUserID = findViewById(R.id.UserIDeditTextNumber);
        String idAsString = editTextUserID.getText().toString();
        int userid;
        if (idAsString.equalsIgnoreCase("")) {
            userid = 1;
        } else {
            userid = Integer.valueOf(idAsString);
        }

        try {
            JSONArray userResponse = new JSONArray(response);
            JSONObject USER = userResponse.getJSONObject(userid - 1);

            name = USER.getString("name");
            username = USER.getString("username");
            email = USER.getString("email");
            city = USER.getJSONObject("address").getString("city");
            company = USER.getJSONObject("company").getString("name");

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

        TextView nameTv = findViewById(R.id.nameTextView);
        nameTv.setText(name);

        TextView usernameTv = findViewById(R.id.usernameTextView);
        usernameTv.setText("username: " + username);

        TextView emailTv = findViewById(R.id.emailTextView);
        emailTv.setText("email: " + email);

        TextView cityTv = findViewById(R.id.cityTextView);
        cityTv.setText("city: " + city);

        TextView companyTv = findViewById(R.id.companyTextView);
        companyTv.setText("company: " + company);
    }

```

```
}
```

5.3 Common implicit intents

Explain terms implicit and explicit intents.

Implicit Intent doesn't specify the component. In such a case, intent provides information on available components provided by the system that is to be invoked. **Explicit Intent** specifies the component. In such a case, intent provides the external class to be invoked.

To your application, implement the following:

- 1) Display a web page
- 2) Set up an alarm or timer
- 3) Open maps on a specific location
- 4) Optional extra- Make a phone call

5.3.1 Special permissions

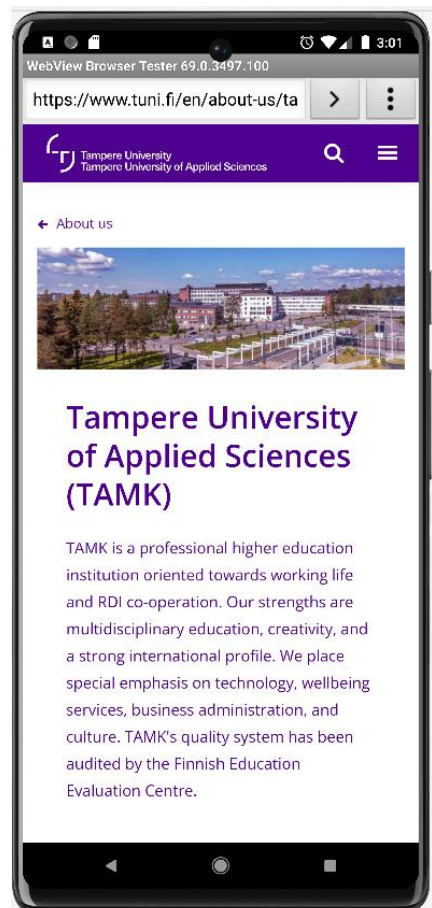
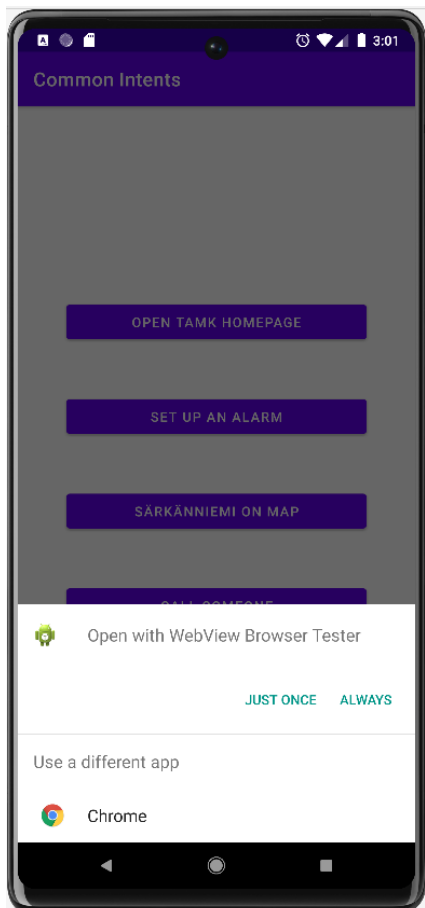
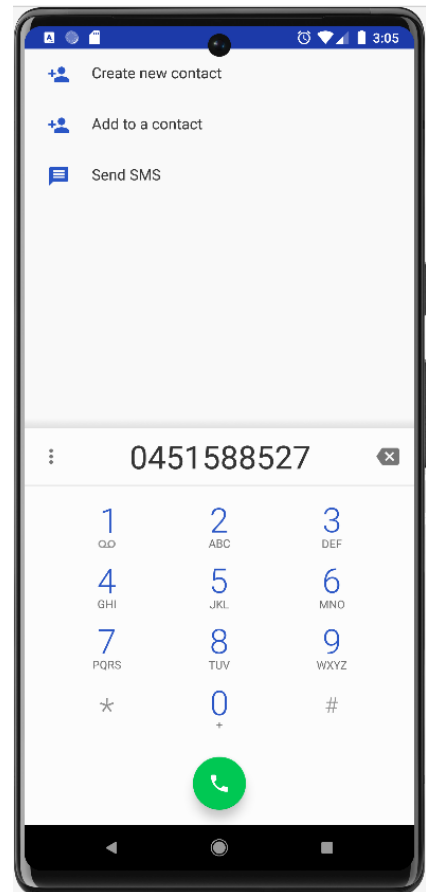
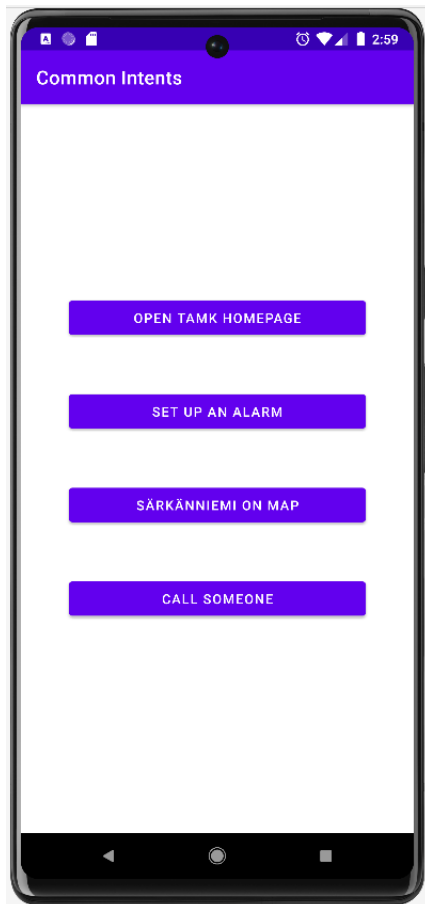
In order to invoke the ACTION_SET_ALARM intent, your app must have the SET_ALARM permission:

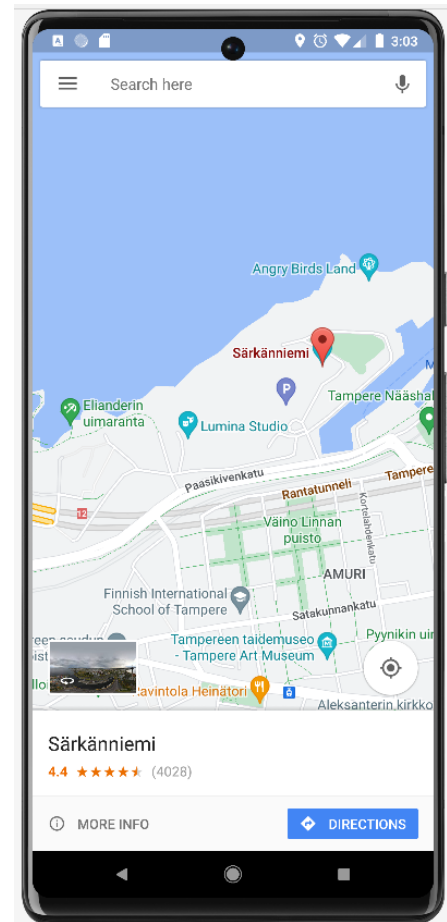
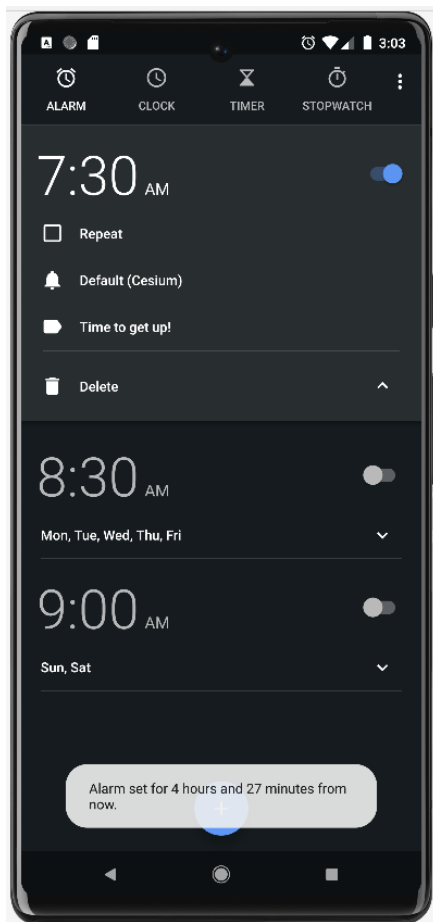
```
<uses-permission android:name="com.android.alarm.permission.SET_ALARM" />
```

The ACTION_CALL action requires that you add the CALL_PHONE permission to your manifest file:

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

5.3.2 Test on emulator





5.3.3 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <Button
        android:id="@+id/button1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="200dp"
        android:layout_marginEnd="50dp"
        android:onClick="openWebPage"
        android:text="Open TAMK homepage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="50dp"
        android:onClick="openWebPage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

        android:onClick="setAlarm"
        android:text="Set up an alarm"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button1" />

<Button
    android:id="@+id/button3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="50dp"
    android:layout_marginEnd="50dp"
    android:onClick="showLocationOnMap"
    android:text="Särkänniemi on map"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.553"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button2" />

<Button
    android:id="@+id/button4"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="50dp"
    android:layout_marginEnd="50dp"
    android:onClick="makeCall"
    android:text="Call someone"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button3" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5.3.4 MainActivity.java

```

package com.example.commonintents;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.AlarmClock;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void openWebPage(View view) {
        Uri webpage = Uri.parse("https://www.tuni.fi/en/about-us/tamk");
        Intent intent = new Intent(Intent.ACTION_VIEW, webpage);
        try {

```



```

        startActivity(intent);
    } catch (ActivityNotFoundException e) {
    }
}

public void setAlarm(View view) {
    Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM)
        .putExtra(AlarmClock.EXTRA_MESSAGE, "Time to get up!")
        .putExtra(AlarmClock.EXTRA_HOUR, 7)
        .putExtra(AlarmClock.EXTRA_MINUTES, 30);
    try {
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
    }
}

public void showLocationOnMap(View view) {
    Uri geoLocation = Uri.parse("geo:61.503164654,23.739663708");
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(geoLocation);
    try {
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
    }
}

public void makeCall(View view) {
    String phoneNumber = "(0)451588527";
    Intent intent = new Intent(Intent.ACTION_DIAL);
    intent.setData(Uri.parse("tel:" + phoneNumber));
    try {
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
    }
}
}

```

6 Final project

Write a Restful Android App, which utilizes some Rest API. You can freely select the API from any open api in WWW or use your own backend (if you have one). **Other requirements:**

- 1) The app should have at least 2 screens (Activities) and navigation between them.
- 2) The activity data should not be lost during Activity lifecycle (e.g. when rotating from landscape to portrait or navigating out and back in).
- 3) All strings should be localized for at least 2 languages and there should not be any hard-coded strings.
- 4) Optionally, you can use some common implicit intent to browse out of your application to utilize some device capabilities.
- 5) The final project must be uploaded to Github or similar and documented in the study diary. The deadline for the submission is Friday 3th March at 23.59.

6.1 Description

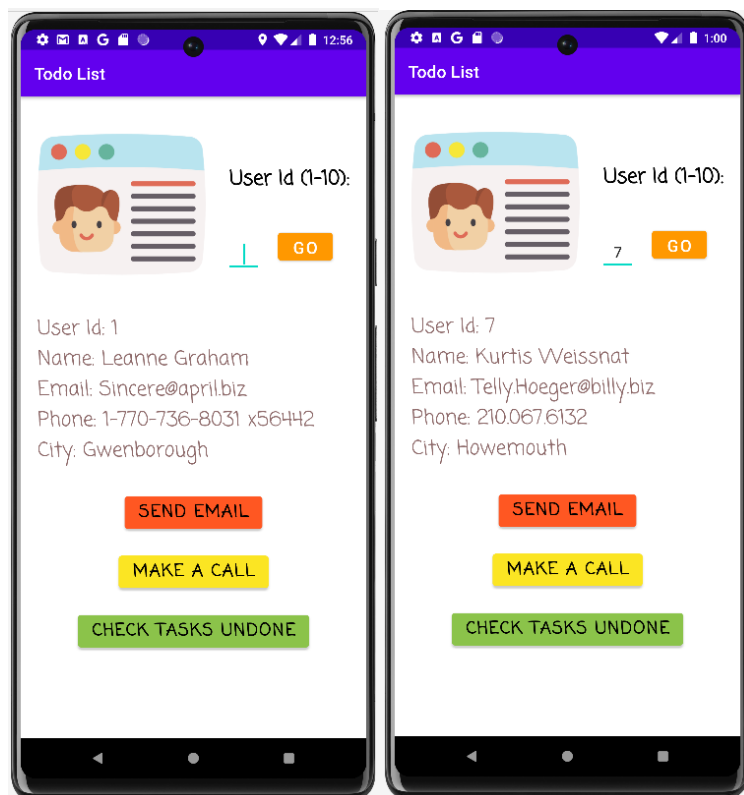
- 1) This Todo List App has two Activities (**MainActivity** and **todoListActivity**).
- 2) Navigation between the 2 screens: go to **todoListActivity** by clicking Button-**CHECK TASKS UNDONE**, go back to **MainActivity** by clicking Button-**BACK**.
- 3) **MainActivity** displays some data (id, name, email, phone, city of a specific user) fetched from <https://jsonplaceholder.typicode.com/users> , **todoListActivity** displays some data (todos that the user has not completed yet) fetched from <https://jsonplaceholder.typicode.com/todos>.
- 4) The activity data is not lost during Activity lifecycle, like when rotating the device or back from email/call apps/**todoListActivity**.
- 5) All strings are localized for English, Chinese and Finnish.
- 6) There are no hard-coded strings.
- 7) After entering an id (number format, 1~10) in the given field, App users can click Button-**GO** and obtain the information of that user (initial user 1).
- 8) If a invalid user id (outside 1-10) is given, a toast-**User Not Found...** will pop.
- 9) The App allows to send email to or dial the given user by clicking corresponding button-**SEND EMAIL**, **MAKE A CALL**.

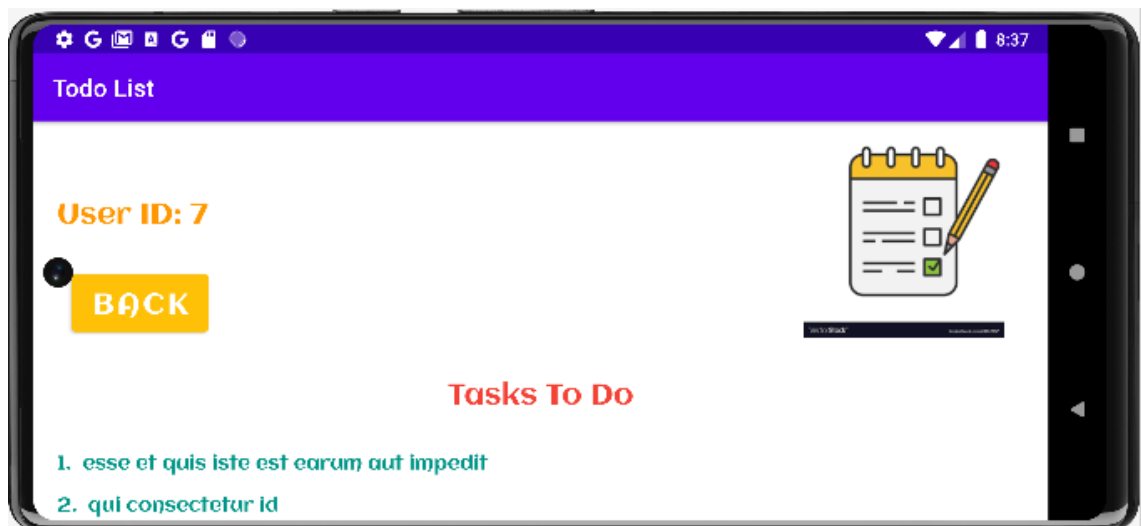
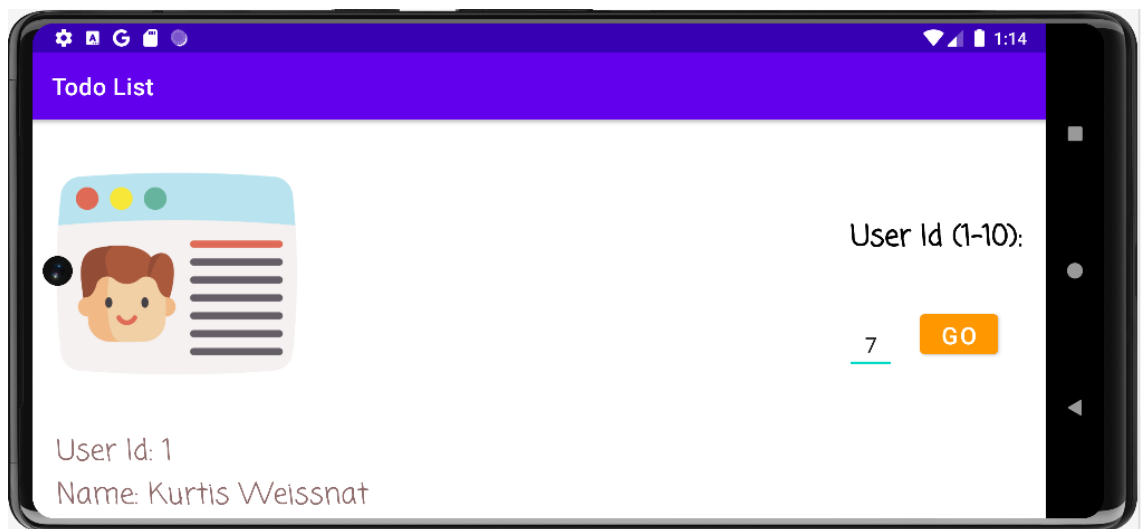
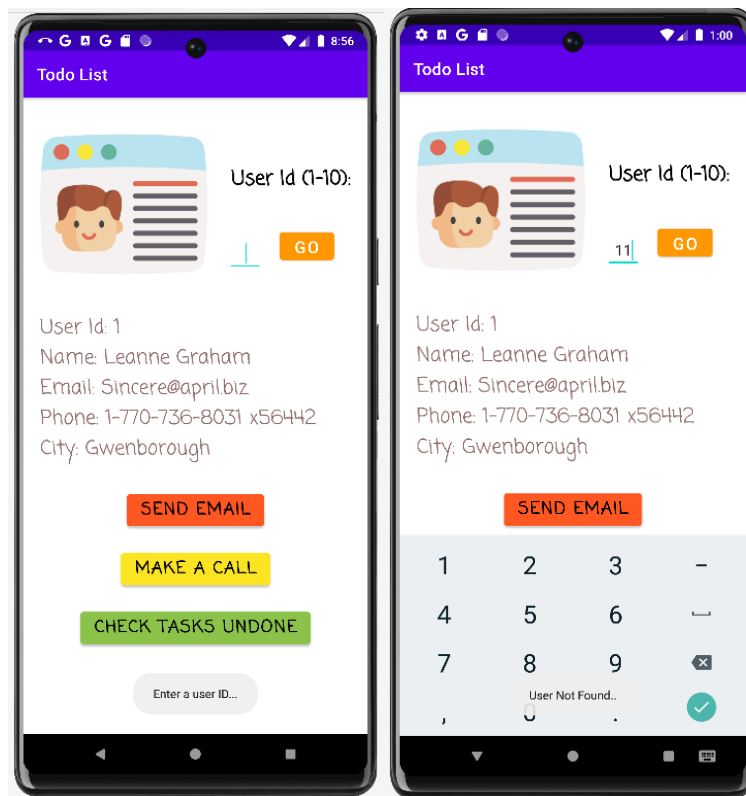
10) The final project is uploaded to Github:

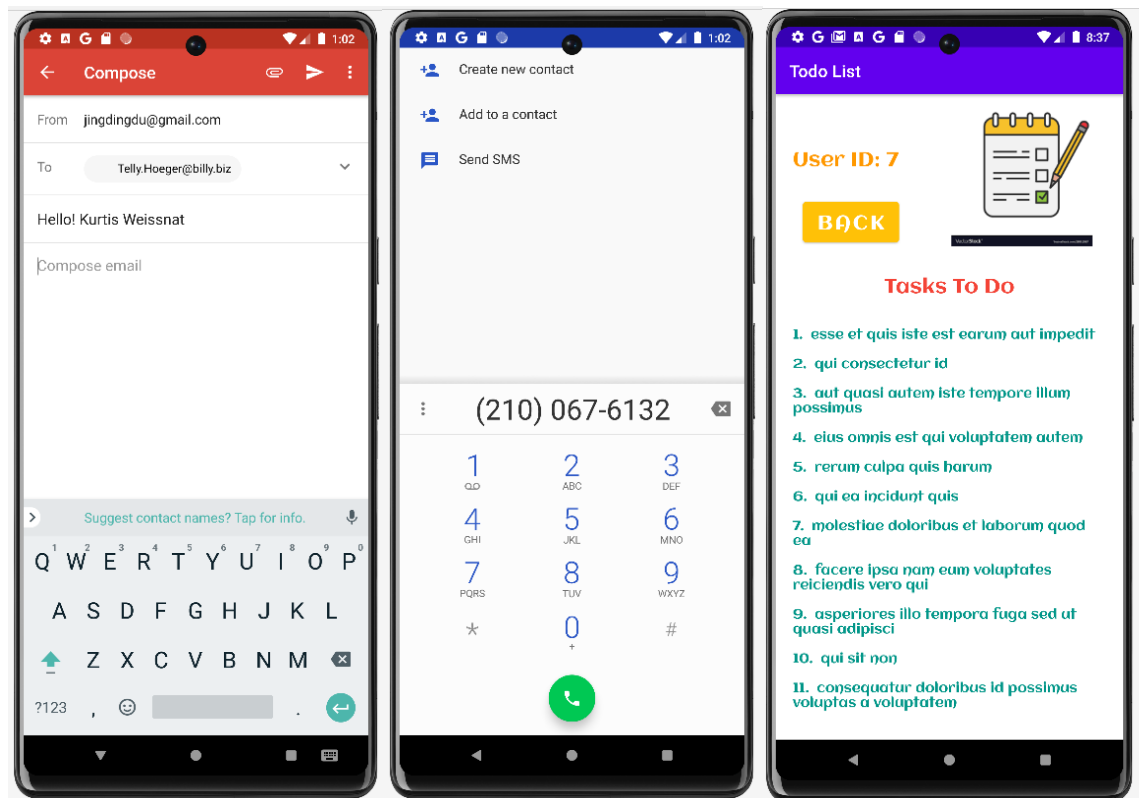
https://github.com/jingjingyang0803/Android_develop_final_project

Jingjing Yang fix empty entering ...		22 minutes ago	🕒 5
📁 .idea	Initial commit-todolist app as final project	last week	
📁 app	fix empty entering	22 minutes ago	
📁 gradle/wrapper	Initial commit-todolist app as final project	last week	
📄 .gitignore	Initial commit-todolist app as final project	last week	
📄 README.md	Create README.md	last week	
📄 build.gradle	Initial commit-todolist app as final project	last week	
📄 gradle.properties	Initial commit-todolist app as final project	last week	
📄 gradlew	Initial commit-todolist app as final project	last week	
📄 gradlew.bat	Initial commit-todolist app as final project	last week	
📄 settings.gradle	Initial commit-todolist app as final project	last week	

6.2 Test on emulator







Key	Resource Folder	Untranslatable	Default Value	Chinese (zh) in China (CN)	Finnish (fi)
app_name	app/src/main/res	<input type="checkbox"/>	Todo List	任务清单	Tehtävälista
hint_message	app/src/main/res	<input type="checkbox"/>	User Id (1-10):	用户ID (1~10):	Käyttäjätunnus (1-10):
send_email	app/src/main/res	<input type="checkbox"/>	Send Email	发送邮件	Lähetää sähköpostia
make_a_call	app/src/main/res	<input type="checkbox"/>	Make a Call	拨打电话	Soita
go	app/src/main/res	<input type="checkbox"/>	Go	确认	Mennä
user_id_display	app/src/main/res	<input type="checkbox"/>	User Id:	用户ID	Käyttäjätunnus:
back	app/src/main/res	<input type="checkbox"/>	Back	返回	Takaisin
check_tasks_undone	app/src/main/res	<input type="checkbox"/>	Check tasks undone	查看待完成任务	Tarkista peruutetut tehtävät
tasks_to_do	app/src/main/res	<input type="checkbox"/>	Tasks To Do	待办事项	Tehtäviä

6.3 activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="28dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/user_info" />

```

```

<TextView
    android:id="@+id/hintTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="20dp"
    android:fontFamily="casual"
    android:text="@string/hint_message"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/userIdEditView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:ems="2"
    android:inputType="number"
    android:textAlignment="center"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintEnd_toStartOf="@+id/userInfoButton"
    app:layout_constraintTop_toBottomOf="@+id/hintTextView" />

<Button
    android:id="@+id/userInfoButton"
    android:layout_width="65dp"
    android:layout_height="45dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="40dp"
    android:backgroundTint="#FF9800"
    android:onClick="userInfoUpdate"
    android:text="@string/go"
    android:textAlignment="center"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="@+id/userIdEditView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/hintTextView" />

<TextView
    android:id="@+id/userInfoTextView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="10dp"
    android:fontFamily="casual"
    android:textAlignment="viewStart"
    android:textColor="#8E6A6A"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

<Button
    android:id="@+id/emailButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:backgroundTint="#FF5722"

```

```

        android:fontFamily="casual"
        android:onClick="sendEmail"
        android:text="@string/send_email"
        android:textColor="#000000"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/userInfoTextView" />

<Button
    android:id="@+id/callButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:backgroundTint="#FBE523"
    android:fontFamily="casual"
    android:onClick="makeCall"
    android:text="@string/make_a_call"
    android:textColor="#000000"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/emailButton" />

<Button
    android:id="@+id/todoListButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:backgroundTint="#8BC34A"
    android:fontFamily="casual"
    android:onClick="openTodoListActivity"
    android:text="@string/check_tasks_undone"
    android:textColor="#000000"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/callButton" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

6.4 MainActivity.java

```

package com.example.todolist;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;

```

```

import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity {

    private RequestQueue queue;
    private int userid = 1;
    private String name = "Leanne Graham",
        email = "Sincere@april.biz",
        phone = "1-770-736-8031 x56442",
        city = "Gwenborough";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Instantiate the RequestQueue.
        queue = Volley.newRequestQueue(this);

        Intent intent = getIntent();
        userid = intent.getIntExtra("USER_ID", 1);

        TextView information = findViewById(R.id.userInfoTextView);
        information.setText("User Id: " + userid +
            "\nName: " + name +
            "\nEmail: " + email +
            "\nPhone: " + phone +
            "\nCity: " + city);
    }

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putString("NAME", name);
        outState.putString("EMAIL", email);
        outState.putString("PHONE", phone);
        outState.putString("CITY", city);
    }

    @Override
    protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        name = savedInstanceState.getString("NAME");
        email = savedInstanceState.getString("EMAIL");
        phone = savedInstanceState.getString("PHONE");
        city = savedInstanceState.getString("CITY");

        TextView information = findViewById(R.id.userInfoTextView);
        information.setText("User Id: " + userid +
            "\nName: " + name +
            "\nEmail: " + email +
            "\nPhone: " + phone +
            "\nCity: " + city);
    }

    public void userInfoUpdate(View view) {
        String url = "https://jsonplaceholder.typicode.com/users";
    }
}

```



```

        // Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url, response -> {
            Log.d("User_App", response);
            parseJsonAndUpdateUI(response);
        }, error -> {
            Log.d("User_App", error.toString());
        });

        queue.add(stringRequest);
    }

    private void parseJsonAndUpdateUI(String response) {
        EditText editTextUserID = findViewById(R.id.userIdEditView);
        String idAsString = editTextUserID.getText().toString();
        int userid_onsearch = 1;
        if (idAsString.equalsIgnoreCase("")) {
            Toast.makeText(MainActivity.this, "Enter a user ID...",
                Toast.LENGTH_SHORT).show();
        } else {
            userid_onsearch = Integer.valueOf(idAsString);

            if (userid_onsearch > 0 && userid_onsearch < 11) {
                userid = userid_onsearch;
                try {
                    JSONArray userResponse = new JSONArray(response);
                    JSONObject USER = userResponse.getJSONObject(userid - 1);

                    name = USER.getString("name");
                    email = USER.getString("email");
                    phone = USER.getString("phone");
                    city = USER.getJSONObject("address").getString("city");

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

                TextView information = findViewById(R.id.userInfo-
                    TextView);

                information.setText("User Id: " + userid +
                    "\nName: " + name +
                    "\nEmail: " + email +
                    "\nPhone: " + phone +
                    "\nCity: " + city);

            } else {
                Toast.makeText(MainActivity.this, "User Not Found..",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }

    public void makeCall(View view) {
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:" + phone));
        try {
            startActivity(intent);
        } catch (ActivityNotFoundException e) {
        }
    }
}

```

```

public void sendEmail(View view) {
    Intent intent = new Intent(Intent.ACTION_SENDTO);
    intent.setData(Uri.parse("mailto: ")); // only email apps
    should handle this
    intent.putExtra(Intent.EXTRA_EMAIL, new String[]{email});
    intent.putExtra(Intent.EXTRA_SUBJECT, "Hello! " + name);
    try {
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
    }
}

public void openToDoListActivity(View view) {
    Intent openToDo = new Intent(this, todoListActivity.class);

    openToDo.putExtra("USER_ID", userid);
    startActivity(openToDo);
}
}

```

6.5 activity_todo_list.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".todoListActivity">

    <TextView
        android:id="@+id/idTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="60dp"
        android:fontFamily="@font/aclonica"
        android:text="@string/user_id_display"
        android:textAlignment="center"
        android:textColor="#FF9800"
        android:textSize="24sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="180dp"
        android:layout_marginEnd="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/todo" />

    <TextView
        android:id="@+id/listTextView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="20dp"
        android:fontFamily="@font/aclonica"
        android:textAlignment="viewStart"

```

```

        android:textColor="#009688"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/headerTextView" />

<Button
    android:id="@+id/todoListButton"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="30dp"
    android:backgroundTint="#FFC107"
    android:fontFamily="@font/aclonica"
    android:onClick="openMainActivity"
    android:text="@string/back"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/idTextView" />

<TextView
    android:id="@+id/headerTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:fontFamily="@font/aclonica"
    android:text="@string/tasks_to_do"
    android:textColor="#F44336"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

6.6 todoListActivity.java

```

package com.example.todolist;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.LinkedList;
import java.util.Queue;

```

```

public class todoListActivity extends AppCompatActivity {

    private RequestQueue queue;
    private int userid;
    private Queue<String> todolist = new LinkedList<>();

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

        Intent intent = getIntent();
        userid = intent.getIntExtra("USER_ID", 1);

        TextView idTv = findViewById(R.id.idTextView);
        idTv.setText("User ID: " + userid);

        // Instantiate the RequestQueue.
        queue = Volley.newRequestQueue(this);

        String url = "https://jsonplaceholder.typicode.com/todos";

        // Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
            response -> {
                Log.d("User_App", response);
                parseJsonAndUpdateUI(response);
            }, error -> {
                Log.d("User_App", error.toString());
            });

        queue.add(stringRequest);
    }

    private void parseJsonAndUpdateUI(String response) {
        try {
            JSONArray todoResponse = new JSONArray(response);
            for (int i = 0; i < todoResponse.length(); i++) {
                JSONObject TODO = todoResponse.getJSONObject(i);
                if (TODO.getInt("userId") == userid && TODO.getBoolean("completed") == false) {
                    todolist.add(TODO.getString("title"));
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        String result = "";
        int counter = 1;
        TextView todolistTextView = findViewById(R.id.listTextView);
        for (String todo : todolist) {
            result += counter + ". " + todo + "\n\n";
            counter++;
        }
        todolistTextView.setText(result);
    }

    public void openMainActivity(View view) {
        Intent openMain = new Intent(this, MainActivity.class);
    }
}

```

```
        openMain.putExtra("USER_ID", userid);
        startActivity(openMain);
    }
}
```

6.7 strings.xml

```
<resources>
    <string name="app_name">Todo List</string>
    <string name="hint_message">User Id (1-10):</string>
    <string name="send_email">Send Email</string>
    <string name="make_a_call">Make a Call</string>
    <string name="go">Go</string>
    <string name="user_id_display">User Id:</string>
    <string name="back">Back</string>
    <string name="check_tasks_undone">Check tasks undone</string>
    <string name="tasks_to_do">Tasks To Do</string>
</resources>
```

Sources used with exercises

<https://stackoverflow.com/questions/18070008/android-button-background-color>

<https://developer.android.com/guide/components/activities/intro-activities>

<https://stackoverflow.com/questions/36957636/what-is-the-relationship-between-xml-and-java-in-android>

<https://developer.android.com/guide/topics/resources/localization>

<https://www.quora.com/Why-does-Android-use-XML-to-define-user-UI-and-not-just-Java-code>

<https://developer.android.com/develop/ui/views/layout/constraint-layout>

https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/en/Unit%201/21_c_understanding_activities_and_intents.html

<https://medium.com/android-news/passing-data-between-activities-using-intent-in-android-85cb097f3016>

<https://developer.android.com/guide/components/activities/activity-lifecycle>

<https://google.github.io/volley/simple.html>

<https://www.section.io/engineering-education/how-to-consume-data-from-an-api-in-android/>

<https://stackoverflow.com/questions/10810418/whats-oncreatebundle-savedInstanceState>

https://www.tutorialspoint.com/android/android_json_parser.htm

<https://www.geeksforgeeks.org/volley-library-in-android/>

<https://www.studytonight.com/android/activity-in-android>

<https://www.section.io/engineering-education/understanding-and-implementing-the-android-lifecycle/>

<https://www.geeksforgeeks.org/bundle-in-android-with-example/>

<https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/>

<https://developer.android.com/guide/components/intents-common>

<https://latitude.to/map/fi/finland/cities/tampere>

<https://dev.to/vtsen/how-to-upload-android-studio-project-to-github-4d2>