kaunas university of technology

ktu

1922

FUNDAMENTALS OF APP DEVELOPMENT (T120B169)

Made by:

group IFF-8/3 student

Kristijonas Karalaitis

Accepted by:

prof. Rytis Maskeliūnas

KAUNAS, 2020

Table of contents

Description of an application

This is an application for travel enthusiasts that whenever the journey is chosen it will generate items to pack for your chosen journey. The application will have an external database therefore the user will have an ability to choose from a variety of items. When a journey is chosen and items are generated, the user wil be redirected to a backpack fragment which will show the list and will let the user to edit the item list or create a notification to remind about the activity the user will be doing.

Implementation

The application is created on Android Studio which provides flexibility by letting a developer to change things programically. To implement social networking and fetching user data from facebook, a facebook API is connected. When the application starts it will let the user to connect to facebook or continue to the application without it. Instead of a new activity for every application‘s feature, the application will use fragments and a menu to navigate in between.

Screenshots of main functions

1. MainActivity

When a user opens an application the following view is displayed. A user is greeted by a message and if he/she is logged in on facebook – the name will be displayed.

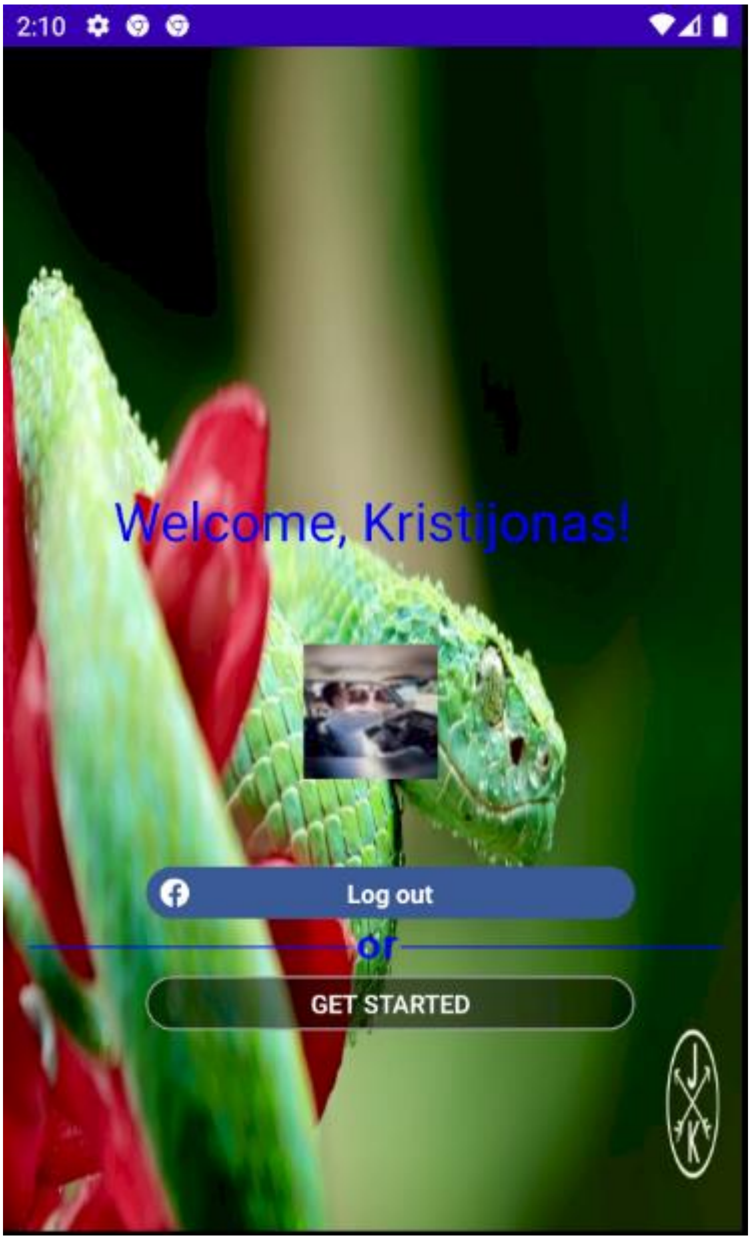
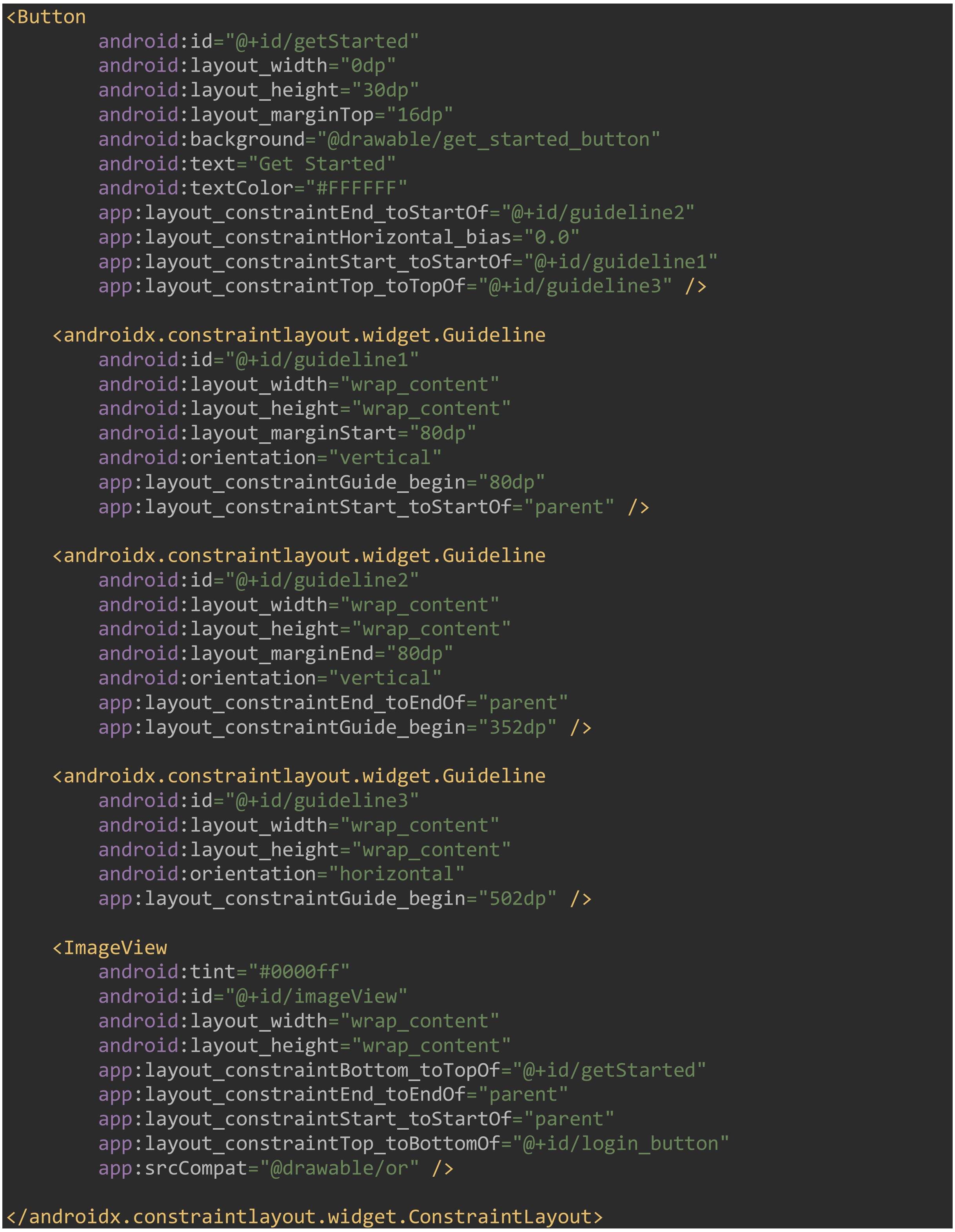


Figure 1 Main screen

Xml layout:

```
<?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">


    <VideoView
        android:id="@+id/video2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome!"
        android:textColor="#0000ff"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.393" />

    <ImageView
        android:id="@+id/profile"
        android:layout_width="75dp"
        android:layout_height="75dp"
        app:layout_constraintBottom_toTopOf="@+id/login_button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <com.facebook.login.widget.LoginButton
        android:id="@+id/login_button"
        android:layout_width="0dp"
        android:layout_height="30dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="16dp"
        android:background="@drawable/fb_button"
        android:textColor="#FFFFFF"
        app:layout_constraintBottom_toTopOf="@+id/guideline3"
        app:layout_constraintEnd_toStartOf="@+id/guideline2"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/guideline1" />
```



```xml
<Button
        android:id="@+id/getStarted"
        android:layout_width="0dp"
        android:layout_height="30dp"
        android:layout_marginTop="16dp"
        android:background="@drawable/get_started_button"
        android:text="Get Started"
        android:textColor="#FFFFFF"
        app:layout_constraintEnd_toStartOf="@+id/guideline2"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/guideline1"
        app:layout_constraintTop_toTopOf="@+id/guideline3" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="80dp"
        android:orientation="vertical"
        app:layout_constraintGuide_begin="80dp"
        app:layout_constraintStart_toStartOf="parent" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="80dp"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintGuide_begin="352dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="502dp" />

    <ImageView
        android:tint="#0000ff"
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/getStarted"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/login_button"
        app:srcCompat="@drawable/or" />

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

2. SecondPage

Once a user continues to an application, a SecondPageActivity is shown with a journey generator fragment. Here a user can choose between a variety of journeys in the database and once clicked on the „Generate items" button, the item list fragment will be shown.

When button is pressed, application pulls items' data from backend that will generate a random list of items

When pressed on the upper left corner or slided from left to the middle, a menu is displayed. Therefore a user can navigate between fragments.

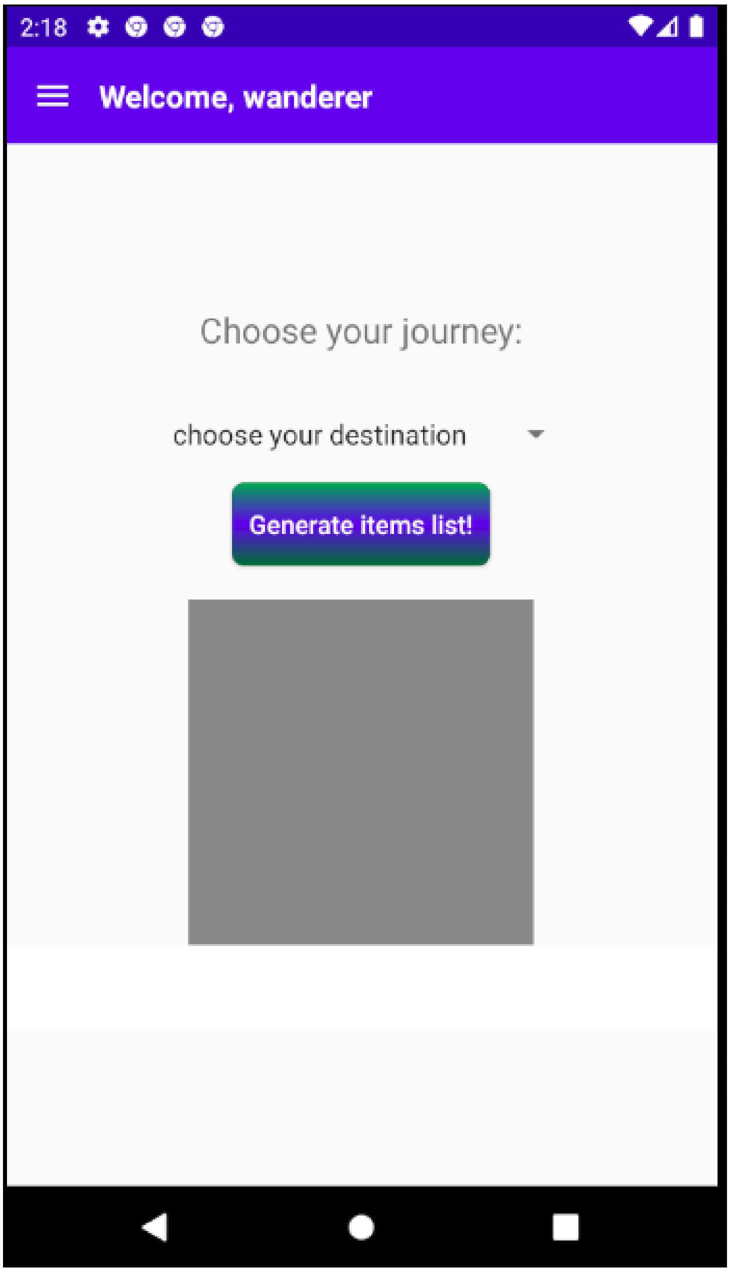


Figure 2 Generator fragment

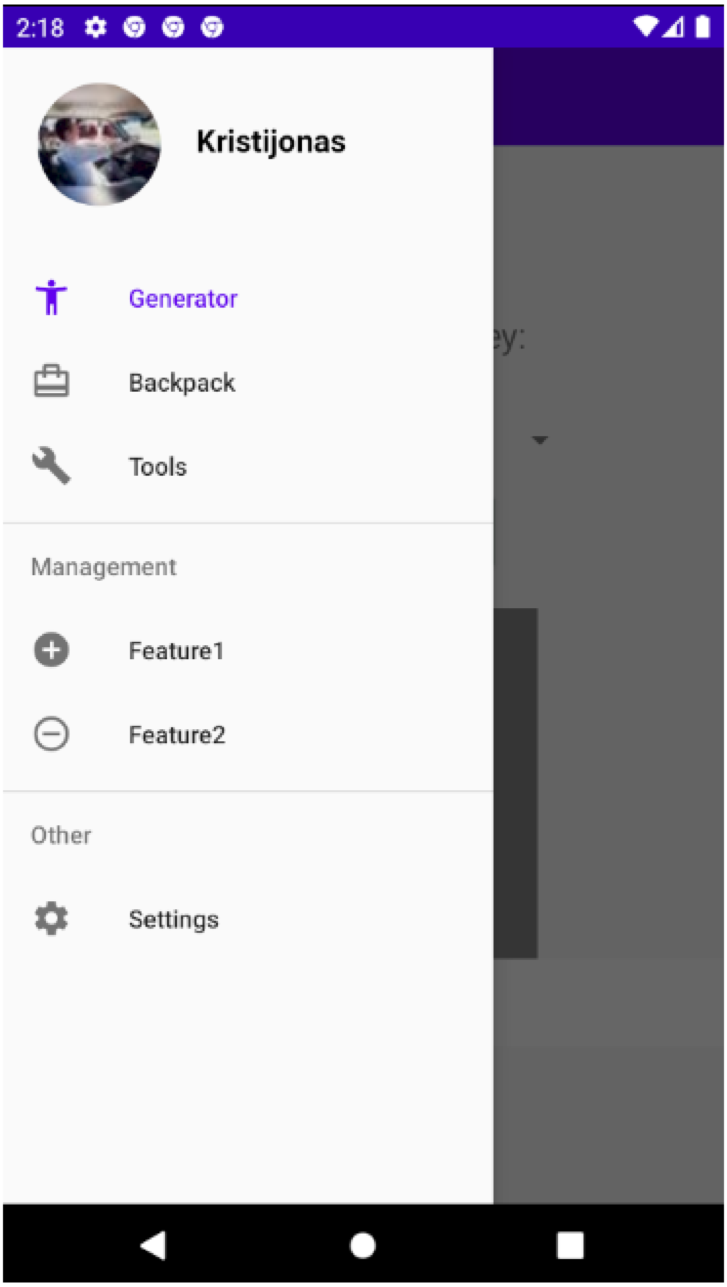


Figure 3 Navigation menu

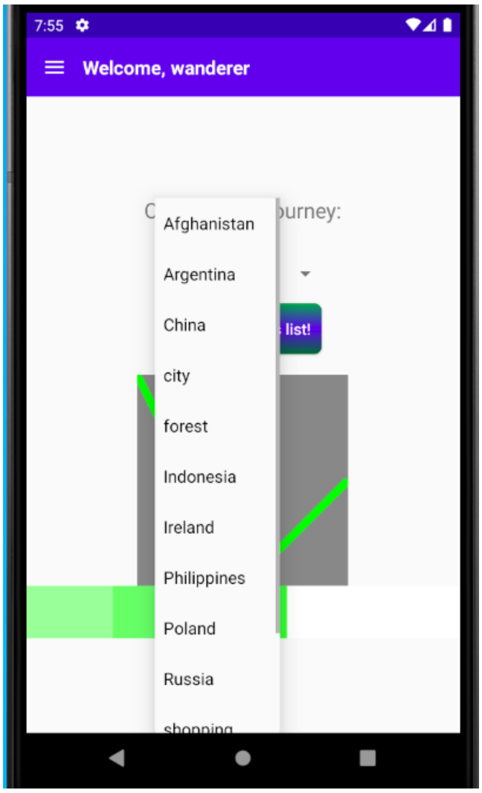


Figure 4 When pressed on spinnner, application pulls journeys' data from Golang backend that pulls data from a database. During pulling process a yellow triangle is shown and progress bar is loading. At the end a green tick is shown

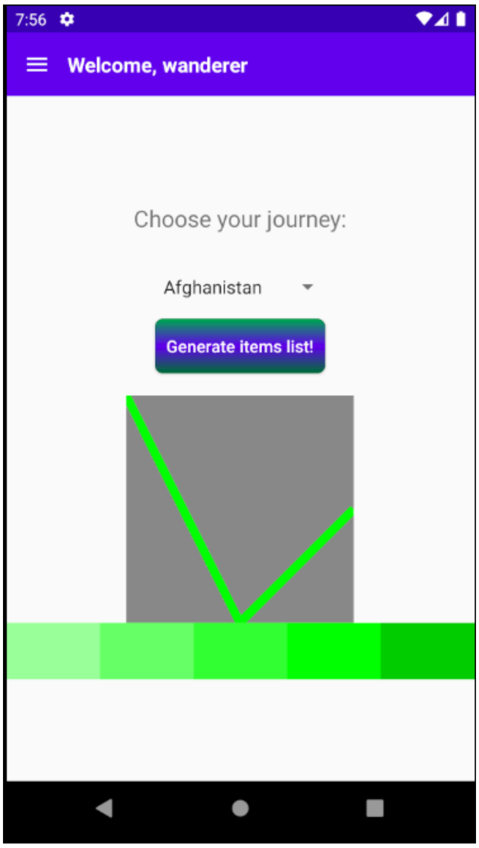
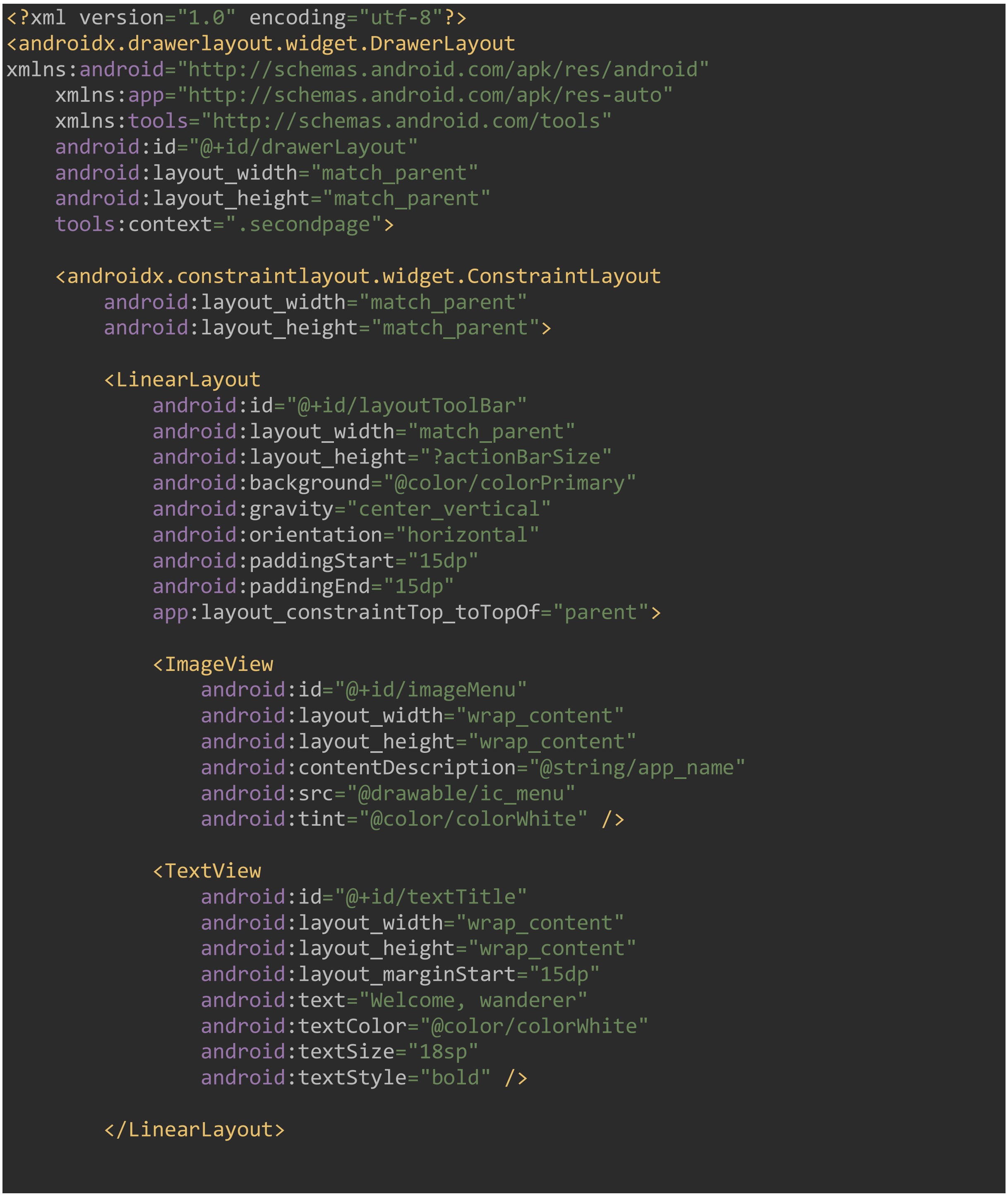


Figure 5 Journey is chosen so items can be generated.

Xml layout:

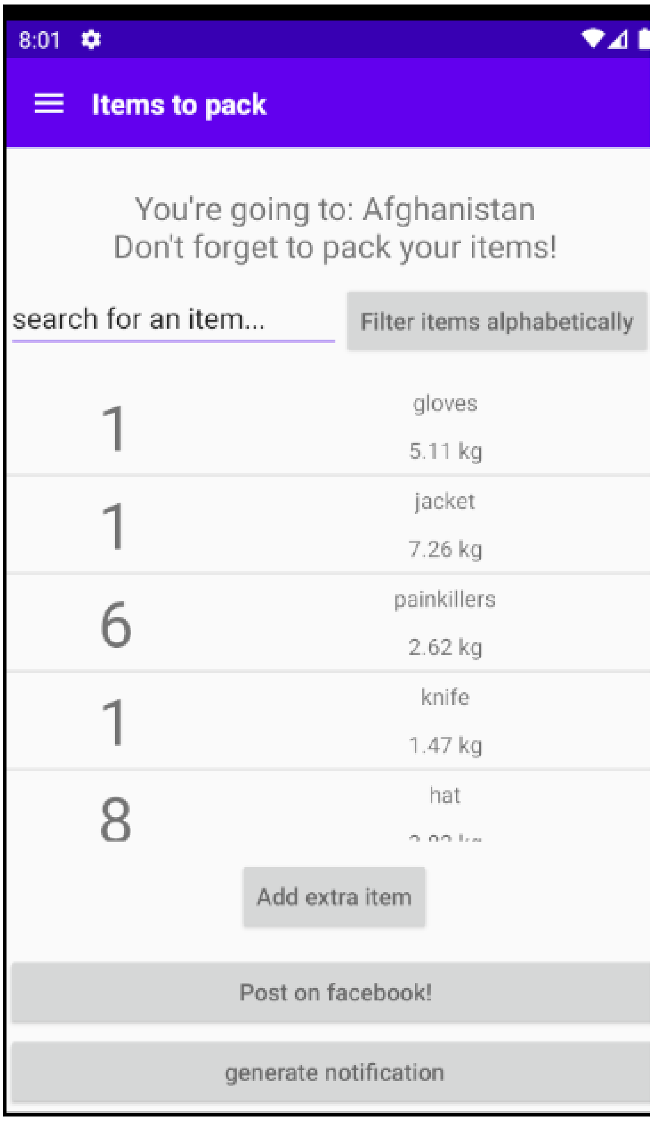


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".secondpage">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:id="@+id/layoutToolBar"
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize"
            android:background="@color/colorPrimary"
            android:gravity="center_vertical"
            android:orientation="horizontal"
            android:paddingStart="15dp"
            android:paddingEnd="15dp"
            app:layout_constraintTop_toTopOf="parent">

            <ImageView
                android:id="@+id/imageMenu"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:contentDescription="@string/app_name"
                android:src="@drawable/ic_menu"
                android:tint="@color/colorWhite" />

            <TextView
                android:id="@+id/textTitle"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="15dp"
                android:text="Welcome, wanderer"
                android:textColor="@color/colorWhite"
                android:textSize="18sp"
                android:textStyle="bold" />

        </LinearLayout>
        <fragment
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_constraintTop_toBottomOf="@id/layoutToolBar"
            app:layout_constraintBottom_toTopOf="parent"
            android:id="@+id/navHostFragment"
            android:name="androidx.navigation.fragment.NavHostFragment"
            app:defaultNavHost="true"
            app:navGraph="@navigation/main"/>

    </androidx.constraintlayout.widget.ConstraintLayout>

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/navigationView"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        app:headerLayout="@layout/layout_navigation_header"
        app:menu="@menu/navigation_menu"
        android:layout_gravity="start"/>

</androidx.drawerlayout.widget.DrawerLayout>
```

3. Backpack fragment

Once the item list is generated the user is redirected to a backpack fragment. Here the item list will be displayed and can be sorted by alphabet, count or weight. A user also can search for a specific item. In this fragment items can be edited, removed or added a new one.

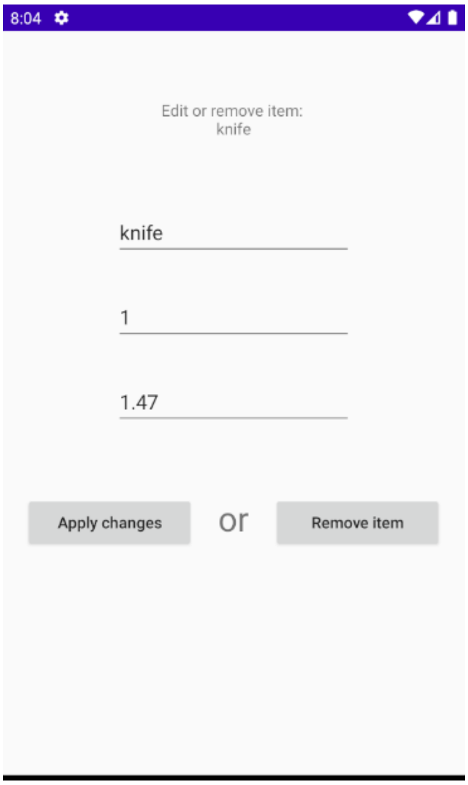
If needed, a user can generate a notifiaction just by a button click.



You're going to: Afghanistan
Don't forget to pack your items!

1	gloves 5.11 kg
1	jacket 7.26 kg
6	painkillers 2.62 kg
1	knife 1.47 kg
8	hat

Figure 6 Backpack (item list) fragment



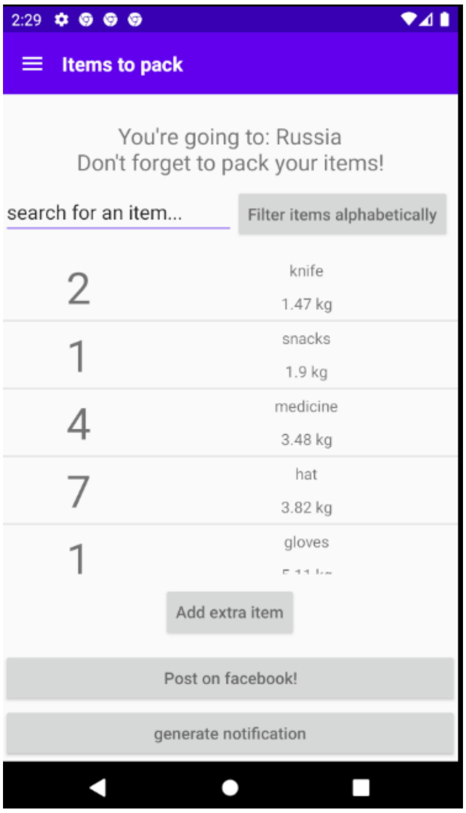
Edit or remove item:
knife

knife

1

1.47

Figure 7 When pressed on the listview, Item can be edited/removed or added a new one



You're going to: Russia
Don't forget to pack your items!

2 knife 1.47 kg

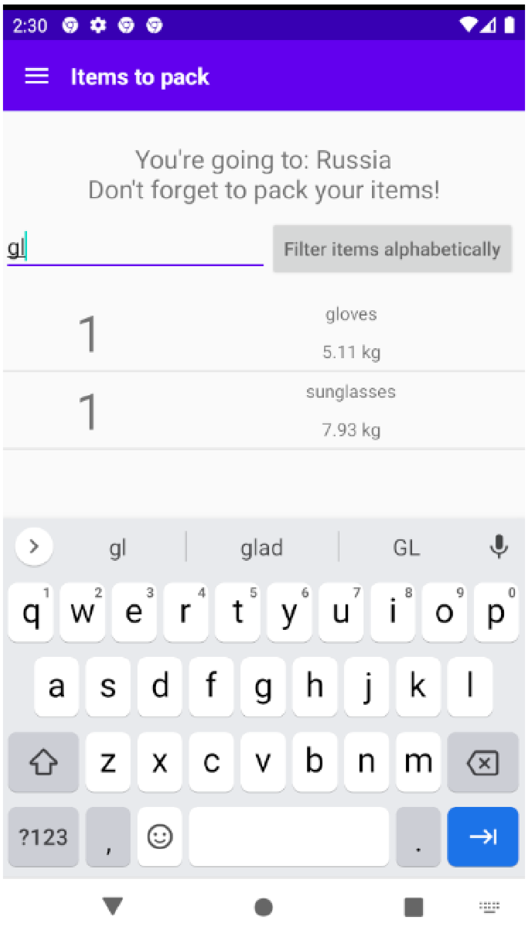
1 snacks 1.9 kg

4 medicine 3.48 kg

7 hat 3.82 kg

1 gloves

Figure 8 Items sorted by weight

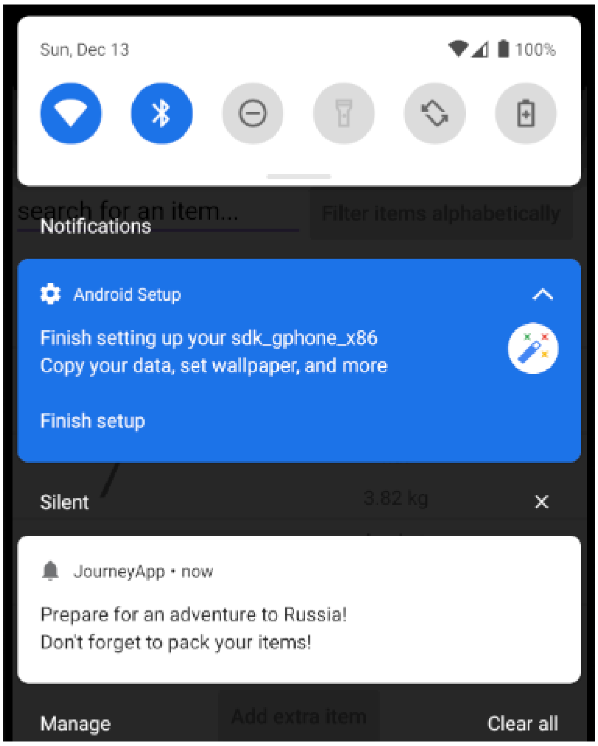


You're going to: Russia
Don't forget to pack your items!

gl | Filter items alphabetically

1	gloves 5.11 kg
1	sunglasses 7.93 kg

Figure 9 Item sorting by text input



Android Setup

Finish setting up your sdk_gphone_x86
Copy your data, set wallpaper, and more

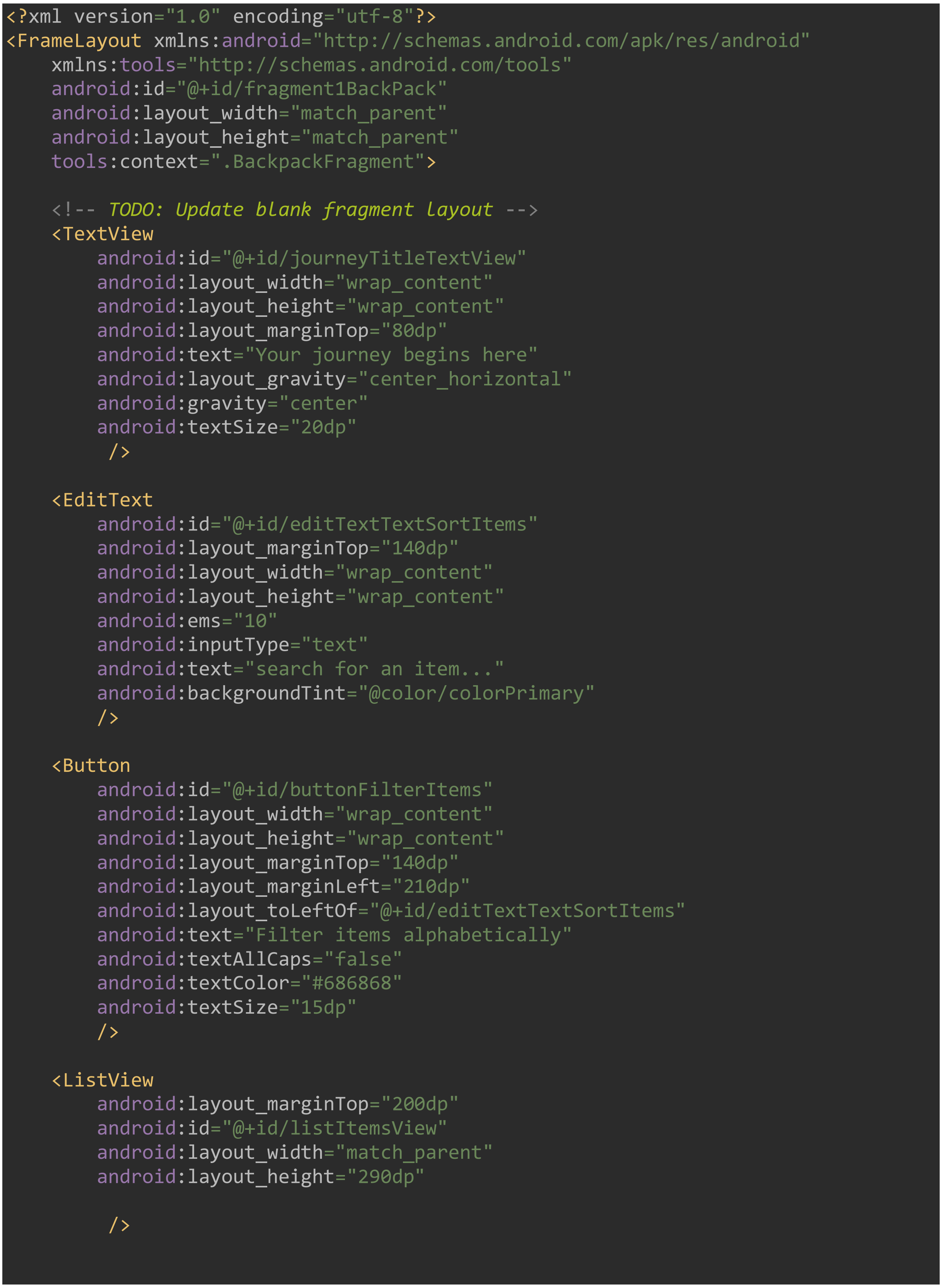
Finish setup

JourneyApp • now

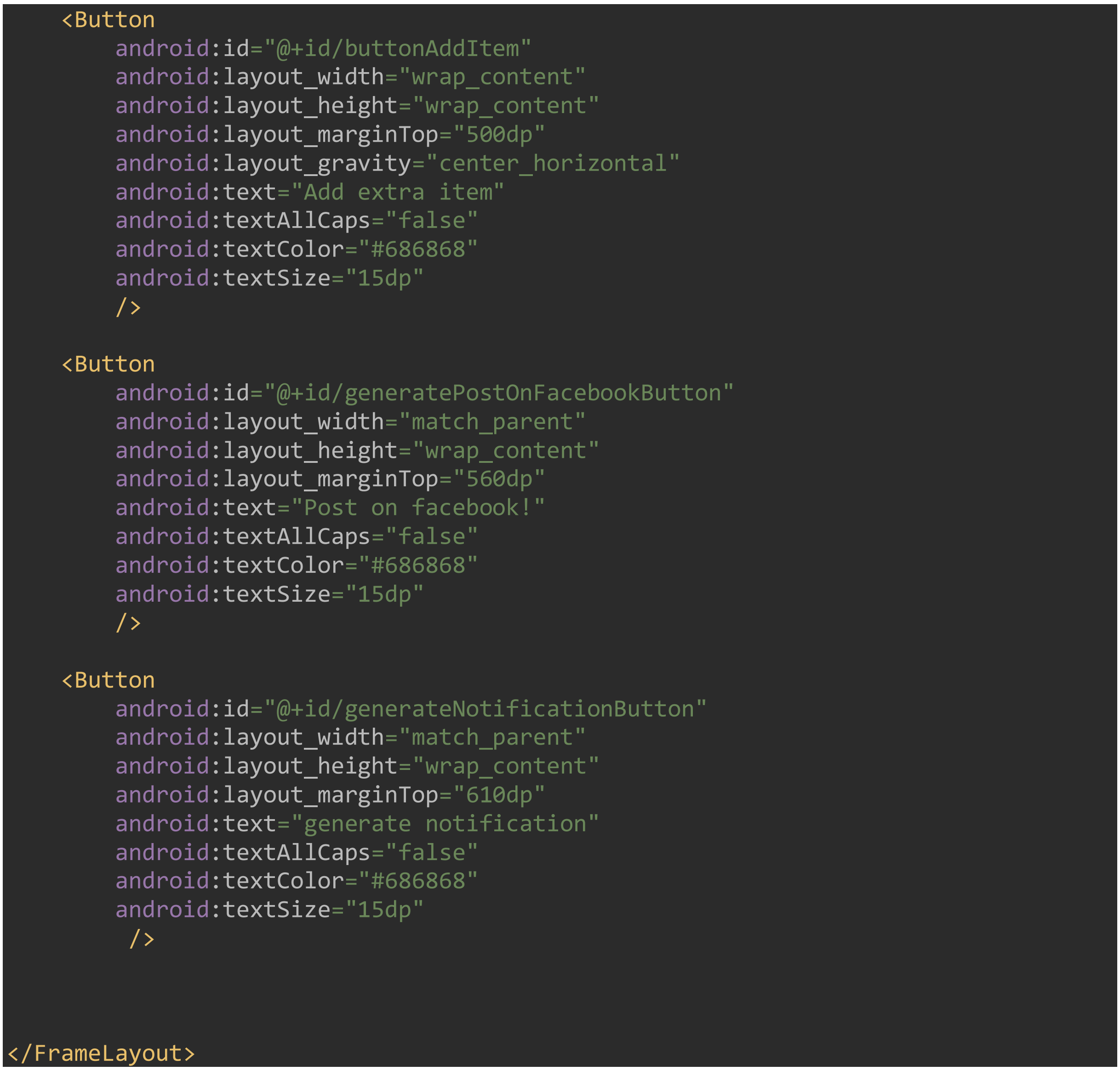
Prepare for an adventure to Russia!
Don't forget to pack your items!

Figure 10 Notification generated

Xml layout:



```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment1BackPack"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BackpackFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/journeyTitleTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:text="Your journey begins here"
        android:layout_gravity="center_horizontal"
        android:gravity="center"
        android:textSize="20dp"
         />

    <EditText
        android:id="@+id/editTextTextSortItems"
        android:layout_marginTop="140dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text"
        android:text="search for an item..."
        android:backgroundTint="@color/colorPrimary"
        />

    <Button
        android:id="@+id/buttonFilterItems"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="140dp"
        android:layout_marginLeft="210dp"
        android:layout_toLeftOf="@+id/editTextTextSortItems"
        android:text="Filter items alphabetically"
        android:textAllCaps="false"
        android:textColor="#686868"
        android:textSize="15dp"
        />

    <ListView
        android:layout_marginTop="200dp"
        android:id="@+id/listItemsView"
        android:layout_width="match_parent"
        android:layout_height="290dp"

         />
```



```
    <Button
        android:id="@+id/buttonAddItem"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="500dp"
        android:layout_gravity="center_horizontal"
        android:text="Add extra item"
        android:textAllCaps="false"
        android:textColor="#686868"
        android:textSize="15dp"
        />

    <Button
        android:id="@+id/generatePostOnFacebookButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="560dp"
        android:text="Post on facebook!"
        android:textAllCaps="false"
        android:textColor="#686868"
        android:textSize="15dp"
        />

    <Button
        android:id="@+id/generateNotificationButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="610dp"
        android:text="generate notification"
        android:textAllCaps="false"
        android:textColor="#686868"
        android:textSize="15dp"
         />



</FrameLayout>
```

4. Tools fragment

On a journey a variety of tools might be helpful. Therefore a menu has an additional fragment „tools“ where a user can try them. All of those were tested and the results were uploaded to youtube.

- In this video we compare network operator‘s and GPS coordinates.Those are displayed by numbers in a TextView box and on google maps: https://youtu.be/QEN-tLK5CUI
- In this video we test the rest of the implemented sensors: https://youtu.be/Q_AwSVbasHI
  - Daylight sensor
  - Magentic field (compass)
  - Accelerometer (also adjusting screen brightness by phone‘s position and informing a user about phone‘s position by coordinates‘ readings)
  - Camera
  - Flashlight
  - A user can also send a SOS signal with a flashlight, when a compass is pointed to the North

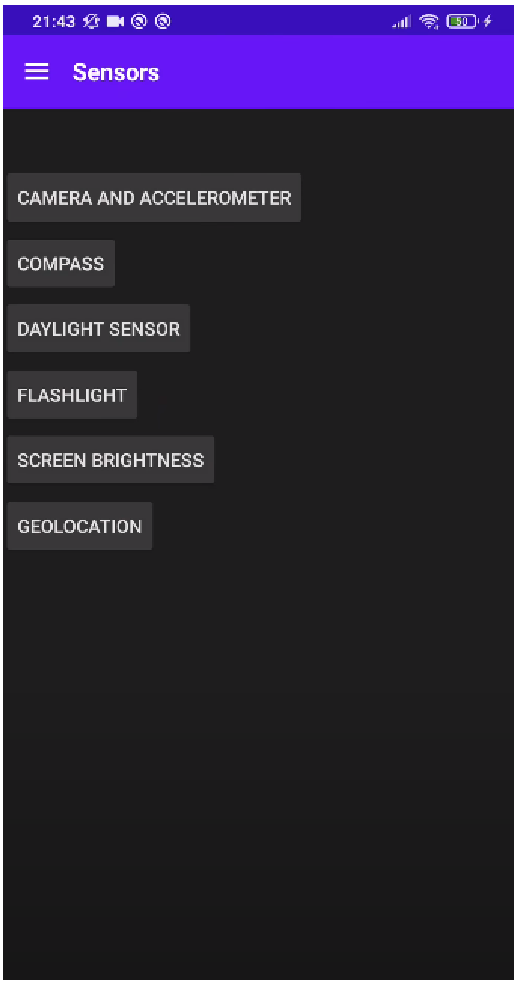


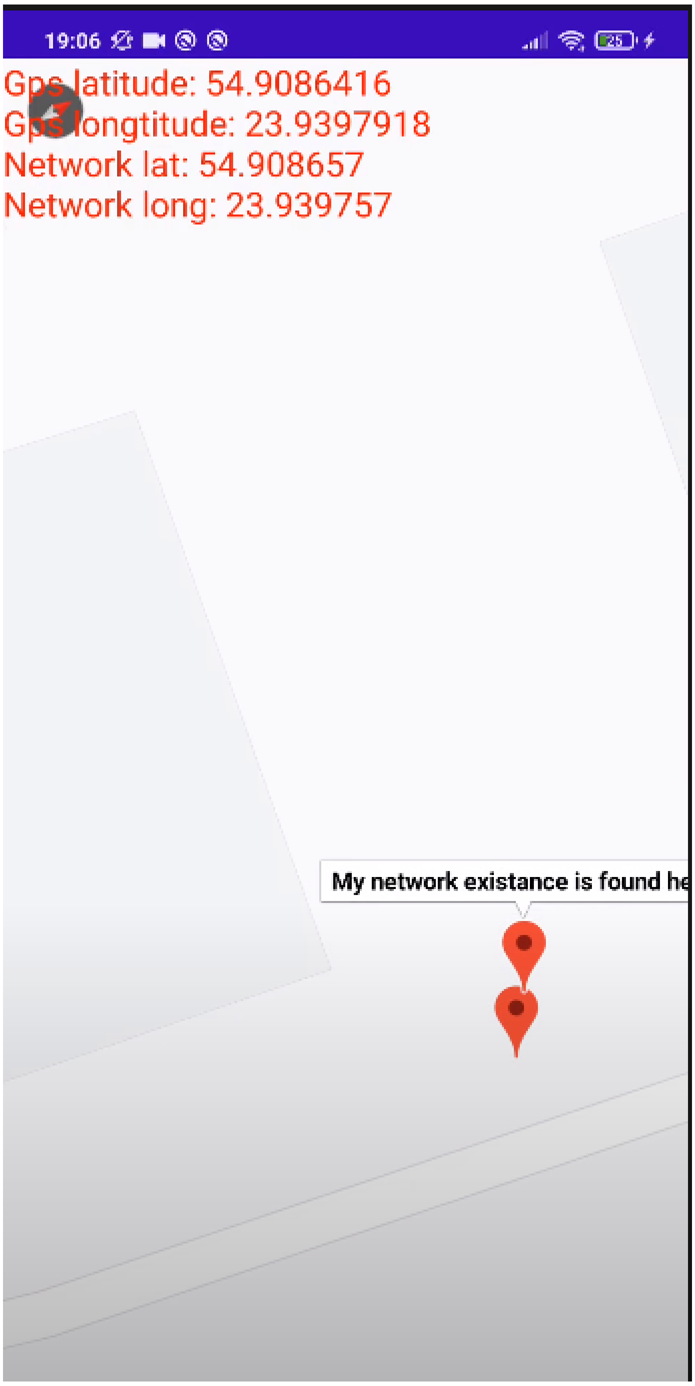
Figure 11 Sensors fragment

Xml layout:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RemindersFragment">

    <!-- TODO: Update blank fragment layout -->
    <Button
        android:id="@+id/sensorsButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="Camera and accelerometer"/>

    <Button
        android:id="@+id/compassButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="150dp"
        android:text="Compass"/>

    <Button
        android:id="@+id/daylightSensorButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="200dp"
        android:text="Daylight sensor"/>

    <Button
        android:id="@+id/flashLightSensorButton"
        android:layout_marginTop="250dp"
        android:text="Flashlight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/screenBrightnessSensorButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Screen brightness"
        android:layout_marginTop="300dp"/>

    <Button
        android:text="Geolocation"
        android:layout_marginTop="350dp"
        android:id="@+id/geolocationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>


</FrameLayout>
```

5. Geolocation activity

If a user grants permissions, an application is getting coordinates from network operator ar GPS sensor. Those are displayed by text and in google maps.



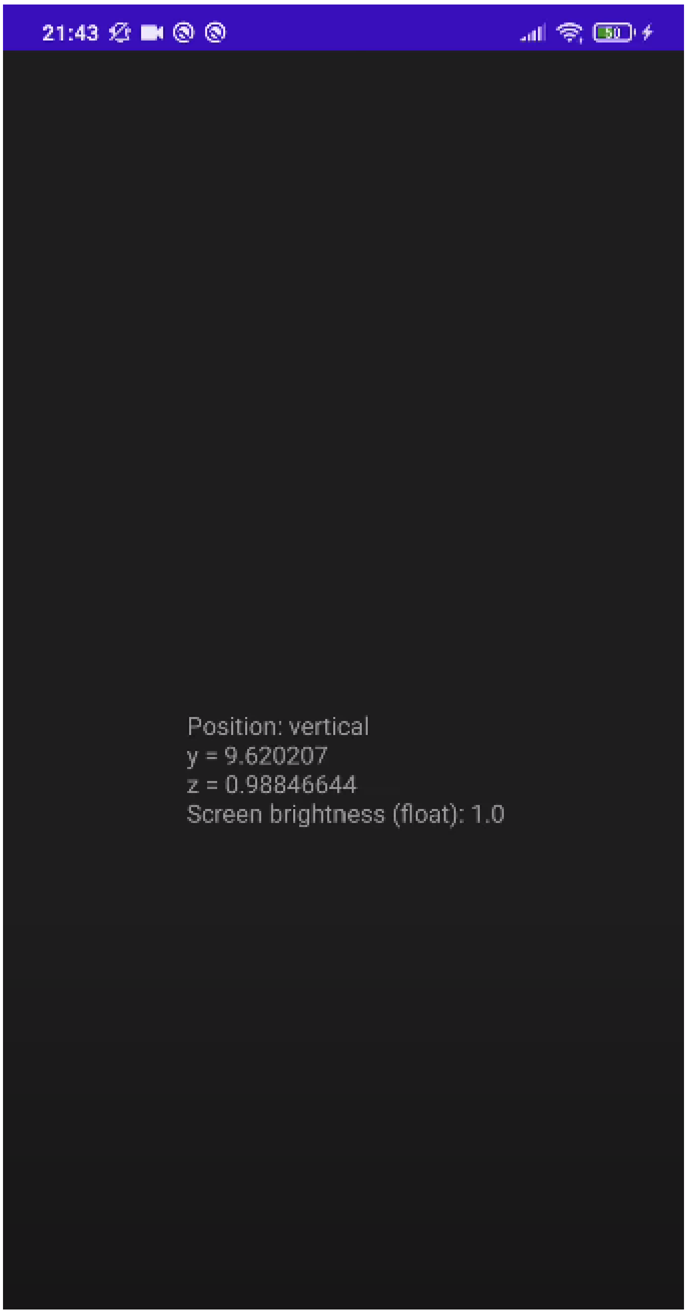
Gps latitude: 54.9086416
Gps longtitude: 23.9397918
Network lat: 54.908657
Network long: 23.939757

My network existance is found he

Figure 12 Coordinates displayed on google maps

6. Screen birghtness activity

Accelerometer detects phone‘s position by x/y/z coordinates and adjusts max screen brightness if a phone is positioned vertically and min brightness if phone is lying on the back.



Position: vertical
y = 9.620207
z = 0.98846644
Screen brightness (float): 1.0

Figure 13 Adjusting screen brightness by phone‘s orientation

7. Flashlight activity

A user is able to turn on or off phone‘s flashlight

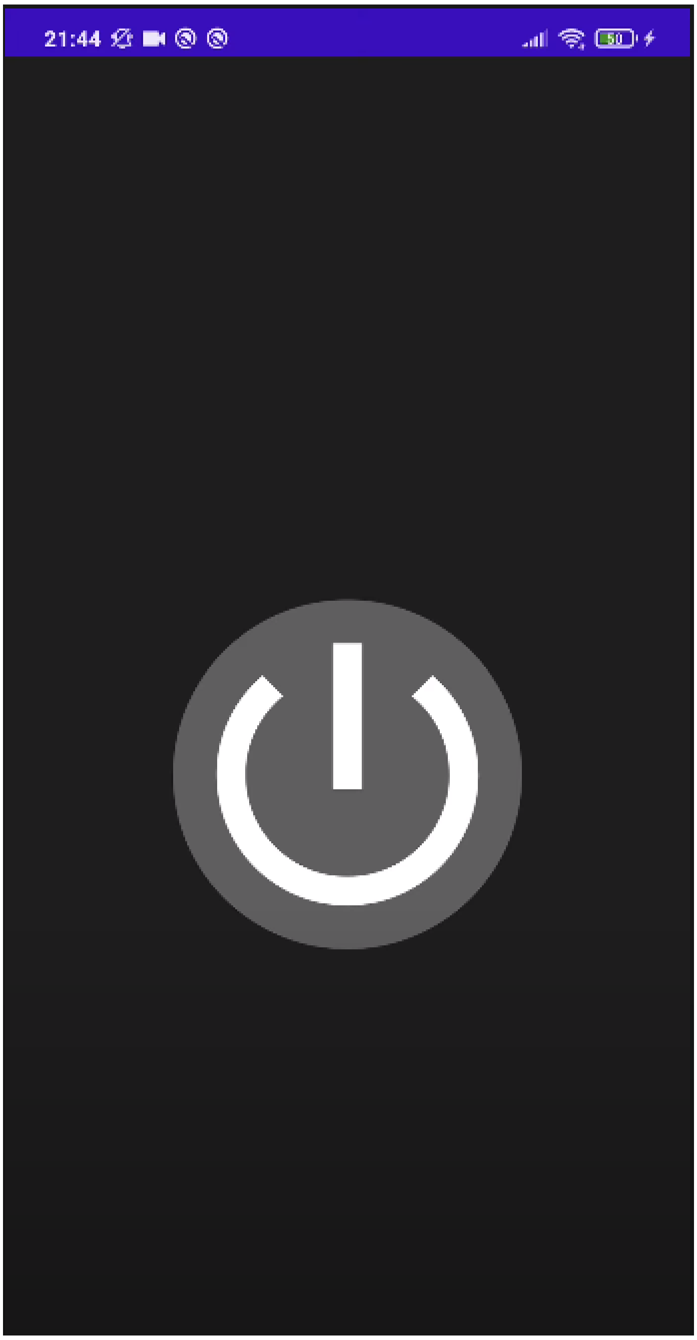
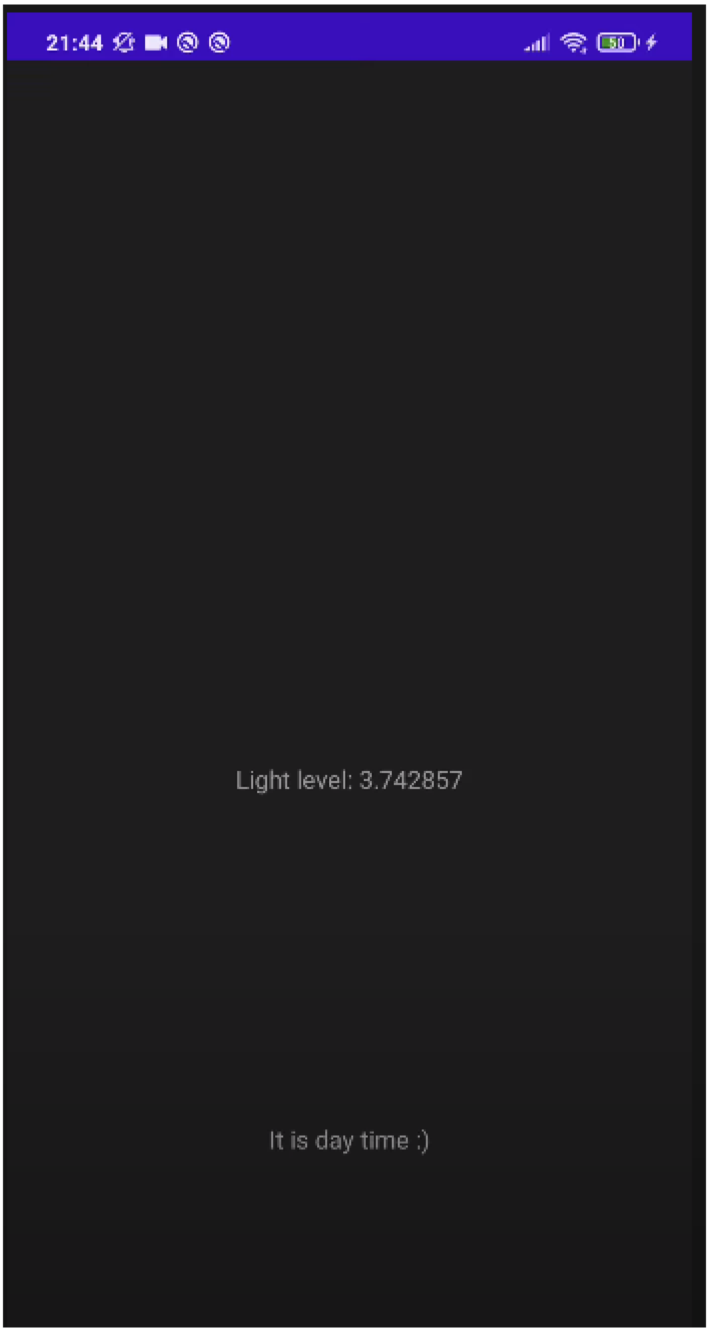


Figure 14 Flashlight activity

8. Daylight sensor activity

A user can check whether it is a night time or day time with a daylight sensor. In this activity the light level is displayed and if it is less than 1 then a message is displayed that it is night time.



Light level: 3.742857

It is day time :)

Figure 15 Daylight sensor

9. Compass activity

In this activity we use .png compass picture and it is being rotated according to magnetic field sensor‘s azimith.

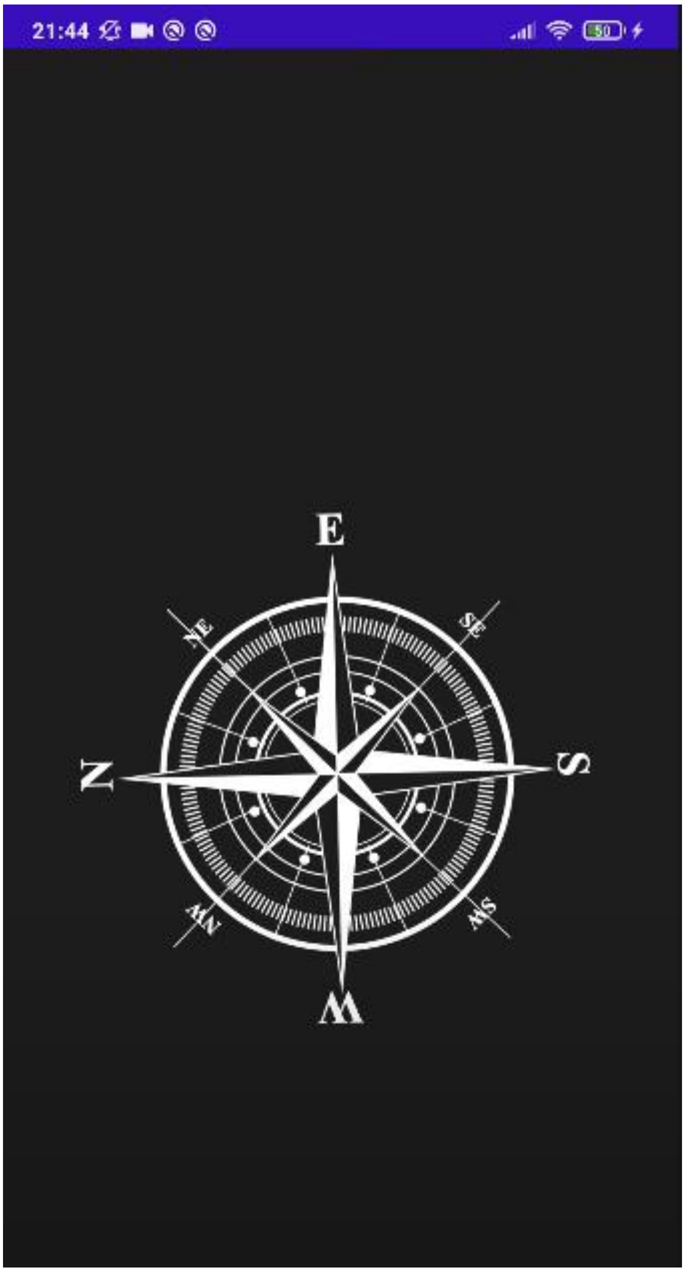
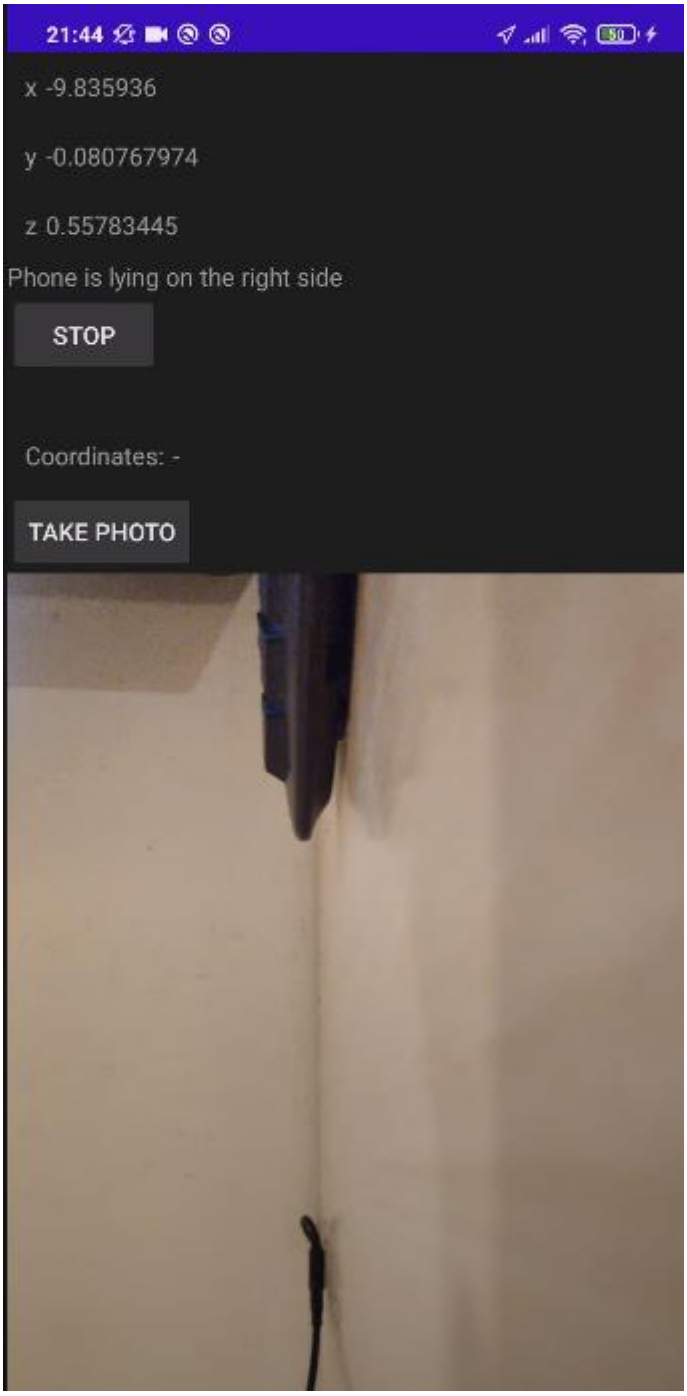


Figure 16 Compass

10. Camera activity

In this activity a user can see reading from accelerometer sensor and will see a message about current phone‘s position. Also if a permision is granted a user can use the device‘s camera to take pictures and save it on device‘s storage.



21:44

x -9.835936

y -0.080767974

z 0.55783445

Phone is lying on the right side

Coordinates: -

Figure 17 Camera

Descriptions of main functions

1) Main activity

Main activity checks if a user is connected on facebook and if so, fetches the data and applies it on a welcoming text and profile picture.

```
AccessTokenTracker tracker = new AccessTokenTracker() {
    @Override
    protected void onCurrentAccessTokenChanged(AccessToken oldAccessToken,
AccessToken currentAccessToken) {
        if (currentAccessToken==null) {
            //jei useris logoutines
        } else {
            loadUserData(currentAccessToken);
        }
    }
};

void loadUserData(AccessToken token) {
    GraphRequest request = GraphRequest.newMeRequest(token, new
GraphRequest.GraphJSONObjectCallback() {
        String firstName;
        @Override
        public void onCompleted(JSONObject object, GraphResponse response) {
            try {
                firstName = object.getString("first_name");
                welcome.setText("Welcome, " + firstName + "!");
            } catch (Exception e) {
                firstName = e.getMessage();
                welcome.setText(firstName);
            }
        }
    });

    Bundle parameters = new Bundle();
    parameters.putString("fields", "first_name");
    request.setParameters(parameters);
    request.executeAsync();

    String imageUrl =
"https://graph.facebook.com/"+token.getUserId()+"/picture?return_ssl_resources=1";
    Picasso.get().load(imageUrl).into(profile);
}

void checkLoginStatus() {
    if (AccessToken.getCurrentAccessToken() != null) {
        loadUserData(AccessToken.getCurrentAccessToken());
    }
}
```

2) SecondPage

Second page activity firstly check for an intent, whether or not it was opened from the notification.

If so, it redirects user to the backpack fragment.

```
Intent intent = getIntent();
String directions = intent.getStringExtra(Constants.navigateToBackpack.key);
if (directions!=null) {
    NavController navController = Navigation.findNavController(
                                                this, R.id.navHostFragment);
    navController.navigate(R.id.backpackFragment);
}
```

Else, it will open main fragment – items generator

```
NavController navController = Navigation.findNavController(this,
R.id.navHostFragment);
NavigationUI.setupWithNavController(navigationView, navController);

final TextView textTitle = findViewById(R.id.textTitle);
navController.addOnDestinationChangedListener(new
NavController.OnDestinationChangedListener() {
    @Override
    public void onDestinationChanged(@NonNull NavController controller, @NonNull
NavDestination destination, @Nullable Bundle arguments) {
        textTitle.setText(destination.getLabel());
    }
});
```

Also this activity will fetch facebook data and if user is logged in, will change menu’s header

```
if (AccessToken.getCurrentAccessToken() != null) {
    //jei neatsijunge nuo fb
    AccessToken token = AccessToken.getCurrentAccessToken();
    loadUserData(token);
}
```

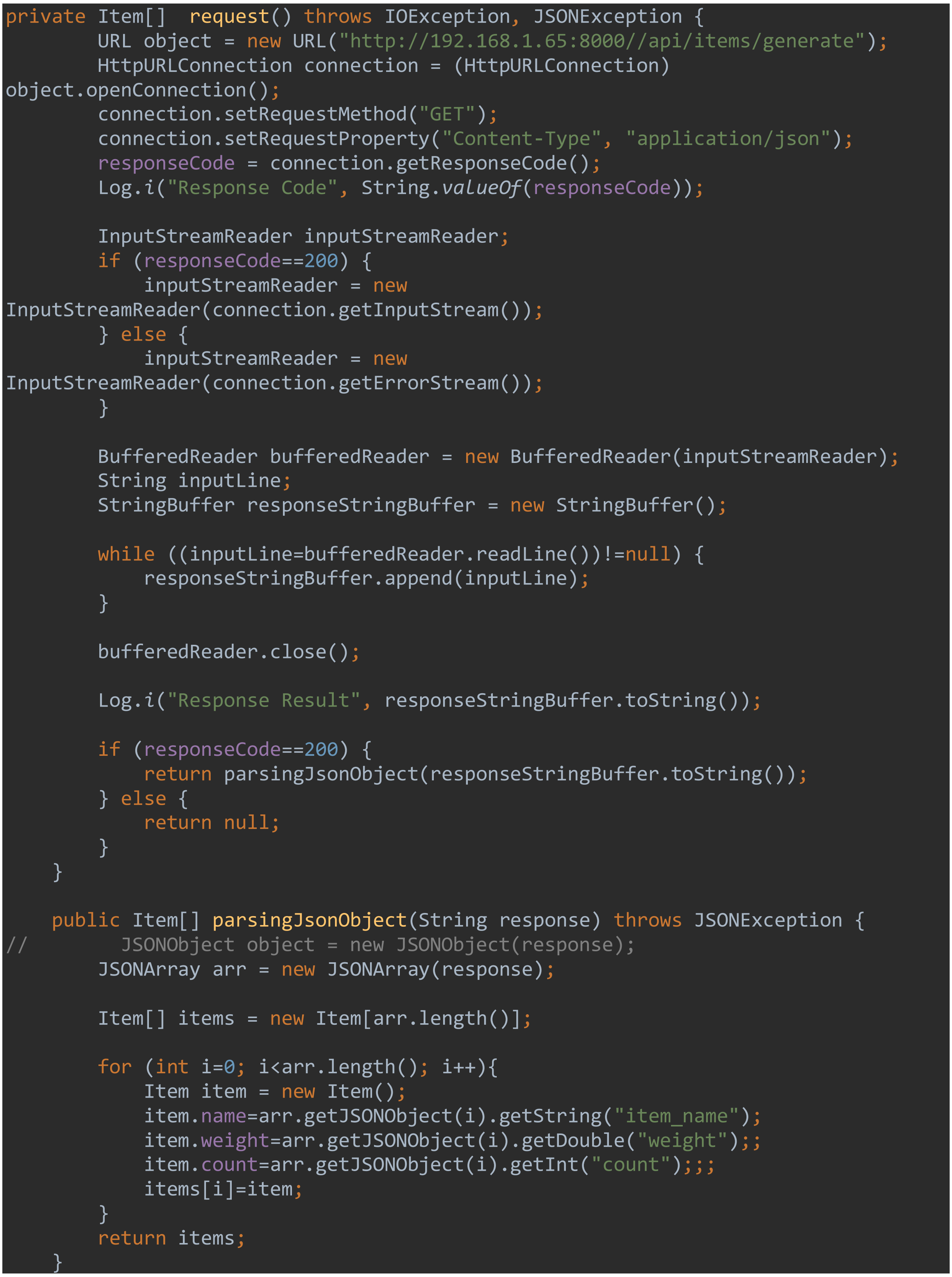
3) GeneratorFragment

Generator fragment will fetch data from GoLang backend (that gets data from a database) and will let the user choose a journey from a list. If fetch was successful the green tick icon will be shown, together with a progress bar

```
spinner.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        setProgressBarBelow();

        //draw in progress
        //------------------------------------------------------------------------
----------
        indicator = view.findViewById(R.id.generated_graphic);
        setIndicatorStatus(IndicatingView.INPROGRESS);

        whiteProgressIndicator = view.findViewById(R.id.progress_bar_white);
        final int w = whiteProgressIndicator.getWidth();

        final Handler handler = new Handler();
        t = new Thread(new Runnable() {
            @Override
            public void run() {
                for (int i=w-10; i>0; i-=10) {
                    final int x=i;
                    handler.post(new Runnable() {
                        @Override
                        public void run() {
                            drawInProgress(x);
                        }
                    });
                    try {
                        Thread.sleep(10);
                    } catch (InterruptedException e)  {

                    }
                }
                drawInProgress(0);
            }
        });
        t.start();
        //------------------------------------------------------------------------
----------

        sendRequest();
        return false;
    }
});
```

```java
void drawInProgress(final int w) {
    getActivity().runOnUiThread(new Runnable() {
        @Override
        public void run() {
            whiteProgressIndicator.setTransWidth(w);
            whiteProgressIndicator.setState(ProgressBarIndicatingView.LOADING);
            whiteProgressIndicator.invalidate();
        }
    });
}

void drawSuccess() {
    indicator = view.findViewById(R.id.generated_graphic);
    setIndicatorStatus(IndicatingView.SUCCESS);
}

public void setIndicatorStatus(final int status) {
    getActivity().runOnUiThread(new Runnable() {
        @Override
        public void run() {
            indicator.setState(status);
            indicator.invalidate();
        }
    });
}

private void sendRequest() {
    RequestOperator db = new RequestOperator();
    db.setListener(this);
    db.start();
}

void sendItemsRequest() {
    GenerateItemsRqOperator ro = new GenerateItemsRqOperator();
    ro.setListener(this);
    ro.start();
}

@Override
public void success(Item[] items) {
    this.items = items;

    LinkedList<Item> generatedItems = new LinkedList<>();
    for (int i=0; i<items.length; i++) {
        generatedItems.add(items[i]);
    }

    Item.SaveGeneratedItemsToSM(generatedItems, context);
}
```

Then the same fragment will fetch items' data and will generate items list accordingly



```java
private Item[]  request() throws IOException, JSONException {
        URL object = new URL("http://192.168.1.65:8000//api/items/generate");
        HttpURLConnection connection = (HttpURLConnection)
object.openConnection();
        connection.setRequestMethod("GET");
        connection.setRequestProperty("Content-Type", "application/json");
        responseCode = connection.getResponseCode();
        Log.i("Response Code", String.valueOf(responseCode));

        InputStreamReader inputStreamReader;
        if (responseCode==200) {
            inputStreamReader = new
InputStreamReader(connection.getInputStream());
        } else {
            inputStreamReader = new
InputStreamReader(connection.getErrorStream());
        }

        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        String inputLine;
        StringBuffer responseStringBuffer = new StringBuffer();

        while ((inputLine=bufferedReader.readLine())!=null) {
            responseStringBuffer.append(inputLine);
        }

        bufferedReader.close();

        Log.i("Response Result", responseStringBuffer.toString());

        if (responseCode==200) {
            return parsingJsonObject(responseStringBuffer.toString());
        } else {
            return null;
        }
    }

    public Item[] parsingJsonObject(String response) throws JSONException {
//        JSONObject object = new JSONObject(response);
        JSONArray arr = new JSONArray(response);

        Item[] items = new Item[arr.length()];

        for (int i=0; i<arr.length(); i++){
            Item item = new Item();
            item.name=arr.getJSONObject(i).getString("item_name");
            item.weight=arr.getJSONObject(i).getDouble("weight");;
            item.count=arr.getJSONObject(i).getInt("count");;;
            items[i]=item;
        }
        return items;
    }
```

4) BackpackFragment

Backpack Fragment will check if a user has generated a journey

```
final String selectedJourney = GetSelectedJourney(context);
if (selectedJourney=="NULL")
{
    HandleWhenJourneyIsNotSelected();
    return view;
}
```

If so, it will fetch data from shared memory and will reflect data with listView

```
void createListAndTextView(String selectedJourney, String[] items) {
    TextView topText = view.findViewById(R.id.journeyTitleTextView);
    String topMessage = "You're going to: " + selectedJourney + "\n" +
            "Don't forget to pack your items:";
    topText.setText(topMessage);

    ListView listView = view.findViewById(R.id.listItemsView);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(context,
android.R.layout.simple_spinner_dropdown_item, items);
    listView.setAdapter(adapter);
}
```

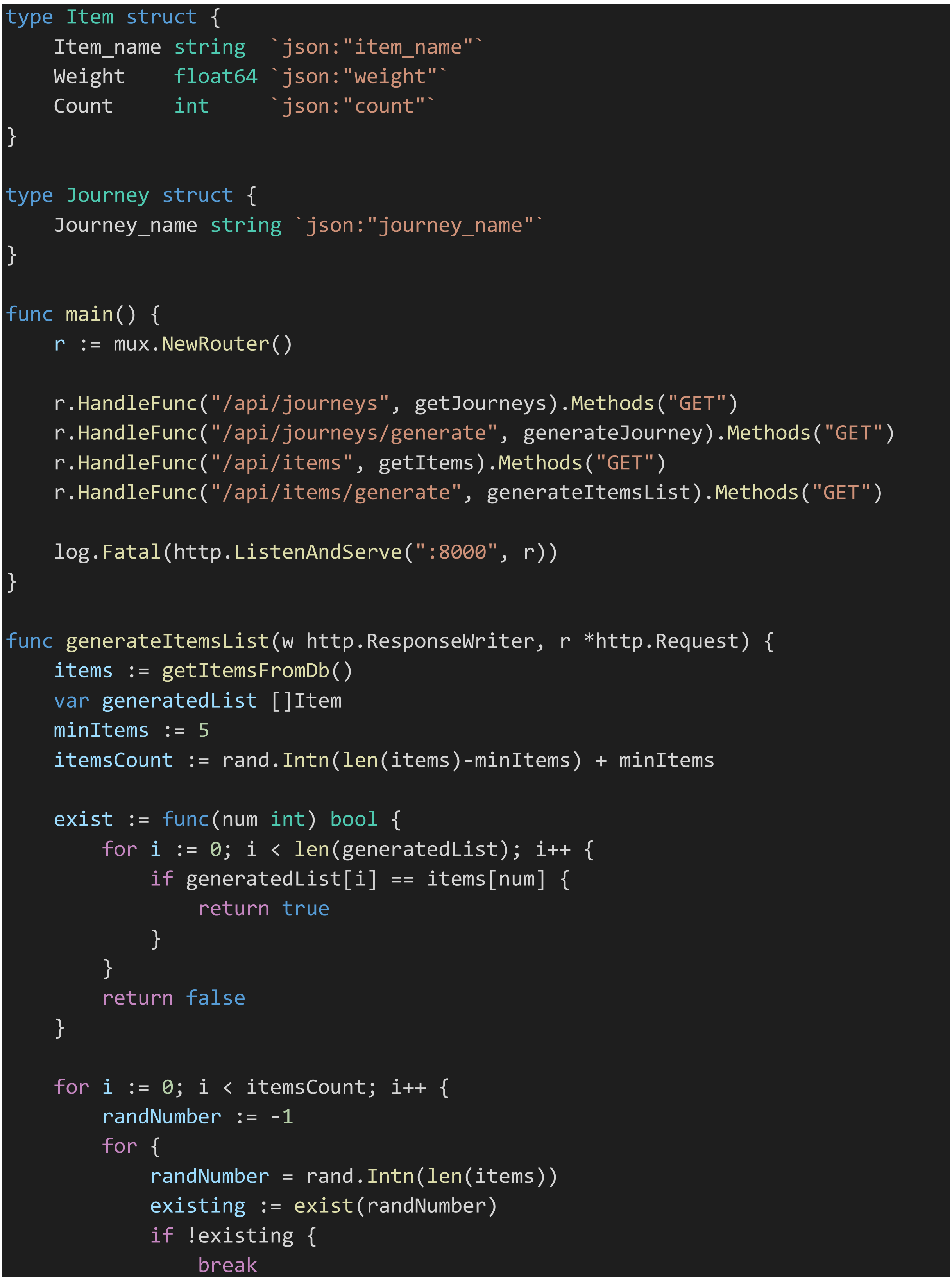
Also user is able to sort the list alphabetically or by letter

```
    LinkedList<String> items = GetGeneratedItems(context);
    @Override
    public void onClick(View v) {
        Collections.sort(items);
        createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
    }
```

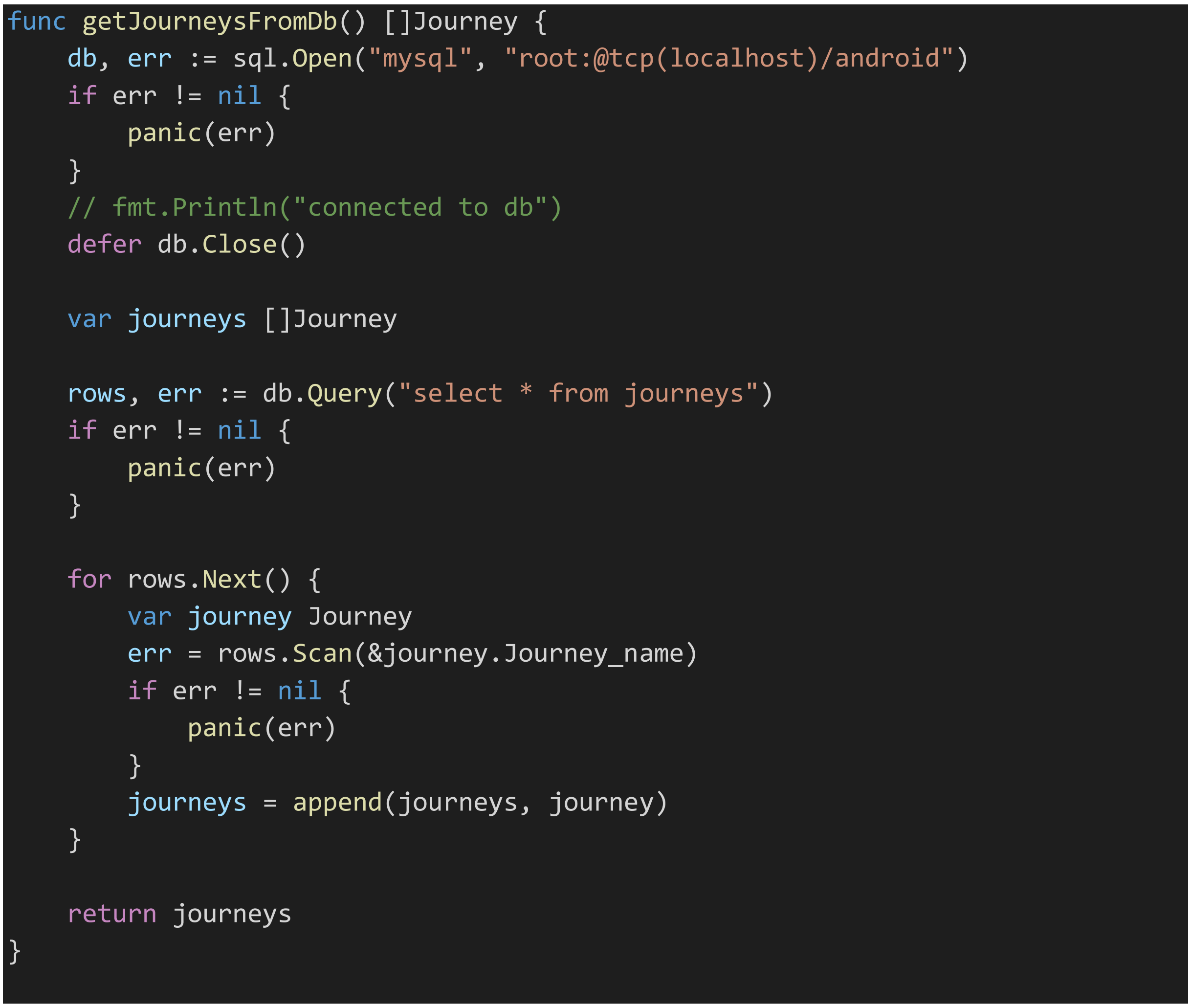
```
@Override
public void onTextChanged(CharSequence s, int start, int before, int count) {
    LinkedList<String> items = GetGeneratedItems(context);
    String filter = editText.getText().toString();

    if ("".equals(filter)) {
        createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
    } else {
        LinkedList<String> sortedItems = new LinkedList<String>();
        for (String item : items) {
            if (item.contains(filter)) {
                sortedItems.add(item);
            }
        }
        createListAndTextView(selectedJourney, sortedItems.toArray(new
String[sortedItems.size()]));
    }
}
```

5) Golang backend

To pull data from a database a REST api is needed. Api was implemented with Golang so the android application can get data via http requests.



```go
type Item struct {
    Item_name string  `json:"item_name"`
    Weight    float64 `json:"weight"`
    Count     int     `json:"count"`
}

type Journey struct {
    Journey_name string `json:"journey_name"`
}

func main() {
    r := mux.NewRouter()

    r.HandleFunc("/api/journeys", getJourneys).Methods("GET")
    r.HandleFunc("/api/journeys/generate", generateJourney).Methods("GET")
    r.HandleFunc("/api/items", getItems).Methods("GET")
    r.HandleFunc("/api/items/generate", generateItemsList).Methods("GET")

    log.Fatal(http.ListenAndServe(":8000", r))
}

func generateItemsList(w http.ResponseWriter, r *http.Request) {
    items := getItemsFromDb()
    var generatedList []Item
    minItems := 5
    itemsCount := rand.Intn(len(items)-minItems) + minItems

    exist := func(num int) bool {
        for i := 0; i < len(generatedList); i++ {
            if generatedList[i] == items[num] {
                return true
            }
        }
        return false
    }

    for i := 0; i < itemsCount; i++ {
        randNumber := -1
        for {
            randNumber = rand.Intn(len(items))
            existing := exist(randNumber)
            if !existing {
                break
            }
        }
        if items[randNumber].Count > 1 {
            items[randNumber].Count = rand.Intn(items[randNumber].Count) + 1
        }
        generatedList = append(generatedList, items[randNumber])
    }

    w.Header().Set("Content-Type", "application/json")
    json.NewEncoder(w).Encode(generatedList)
}

func getItems(w http.ResponseWriter, r *http.Request) {
    items := getItemsFromDb()
    w.Header().Set("Content-Type", "application/json")
    json.NewEncoder(w).Encode(items)
}

func getJourneys(w http.ResponseWriter, r *http.Request) {
    journeys := getJourneysFromDb()
    w.Header().Set("Content-Type", "application/json")
    json.NewEncoder(w).Encode(journeys)
}

func getItemsFromDb() []Item {
    db, err := sql.Open("mysql", "root:@tcp(localhost)/android")
    defer db.Close()

    var items []Item

    rows, err := db.Query("select * from items")
    if err != nil {
        panic(err)
    }

    for rows.Next() {
        var item Item
        err = rows.Scan(&item.Item_name, &item.Weight, &item.Count)
        if err != nil {
            panic(err)
        }
        items = append(items, item)
    }

    return items
}
```



```go
func getJourneysFromDb() []Journey {
    db, err := sql.Open("mysql", "root:@tcp(localhost)/android")
    if err != nil {
        panic(err)
    }
    // fmt.Println("connected to db")
    defer db.Close()

    var journeys []Journey

    rows, err := db.Query("select * from journeys")
    if err != nil {
        panic(err)
    }

    for rows.Next() {
        var journey Journey
        err = rows.Scan(&journey.Journey_name)
        if err != nil {
            panic(err)
        }
        journeys = append(journeys, journey)
    }

    return journeys
}
```

6) Geolocation activity

In this activity the application asks user for GPS and network permissions and if they are granted, data from sensors is fetched

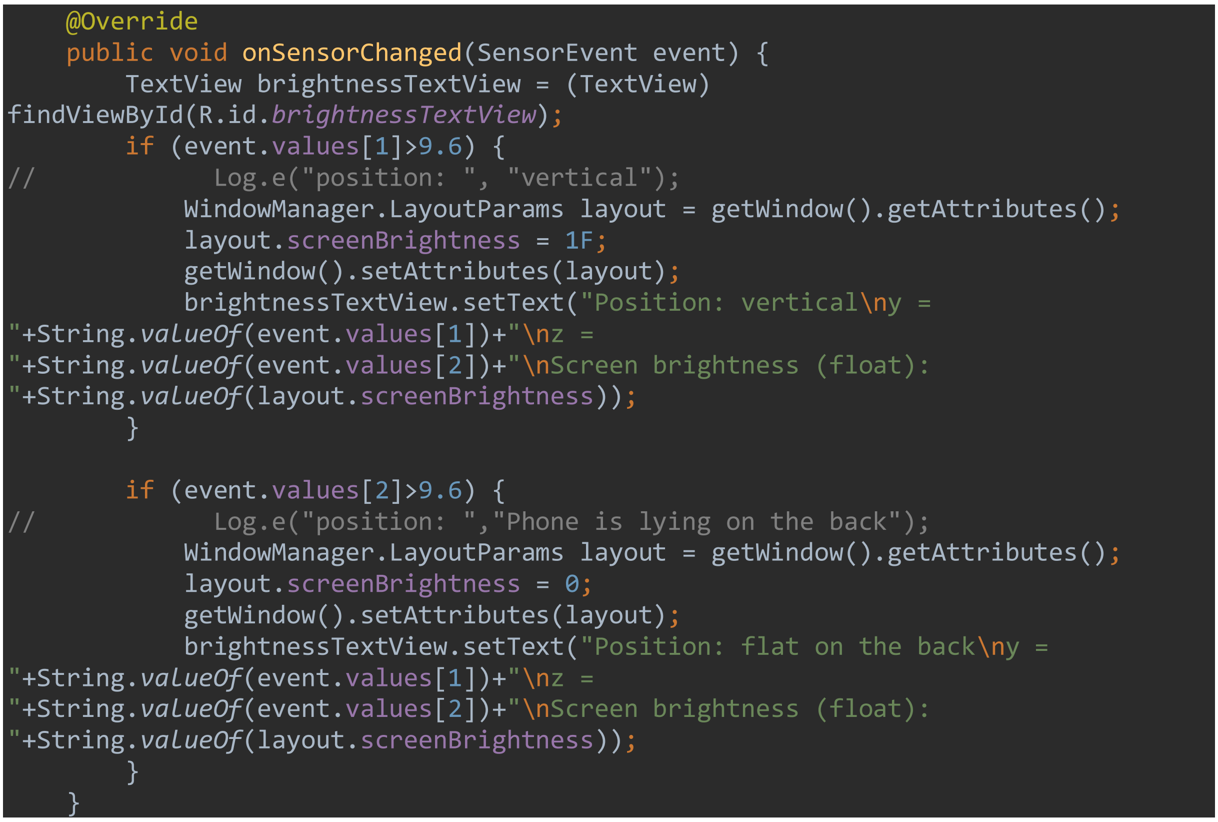
```
fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);

if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
    return;
}
fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this, new
OnSuccessListener<Location>() {
    @Override
    public void onSuccess(Location location) {
        if (location != null) {
            String gpsLatitude = Double.toString(location.getLatitude());
            String gpsLongtitude = Double.toString(location.getLongitude());

            String networkLat = "0";
            String networkLong = "0";

            if
(locationManagerNetwork.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
                if (ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED
&&

ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                    location =
locationManagerNetwork.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    networkLat = Double.toString(location.getLatitude());
                    networkLong = Double.toString(location.getLongitude());
                }
            }

            geolocationTextView.setText("Gps latitude: " + gpsLatitude + "\nGps
longtitude: " + gpsLongtitude+"\nNetwork lat: "+networkLat+"\nNetwork long:
"+networkLong);
            geolocationTextView.setTextSize(20);
            geolocationTextView.setTextColor(Color.RED);
        }
    }
});
```

If fetch was successful the coordinates are displayed on google maps

```
@Override
public void onMapReady(GoogleMap googleMap) {
    LatLng latLng = new LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions().position(latLng).title("My
Gps existance is found here");
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 5));
    googleMap.addMarker((markerOptions));
    //---

    latLng = new LatLng(networkLocation.getLatitude(),
networkLocation.getLongitude());
    markerOptions = new MarkerOptions().position(latLng).title("My network
existance is found here");
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 5));
    googleMap.addMarker((markerOptions));
}
```

7) Screen brightness activity

In this activity an application monitors phone’s orientation. Once a phone is at 90 degrees (positioned vertically) screen brightness is adjusted to max level. And once a phone is put on his back, screen brightness is reduced to min level.



```
    @Override
    public void onSensorChanged(SensorEvent event) {
        TextView brightnessTextView = (TextView)
findViewById(R.id.brightnessTextView);
        if (event.values[1]>9.6) {
//            Log.e("position: ", "vertical");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 1F;
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: vertical\ny =
"+String.valueOf(event.values[1])+"\nz =
"+String.valueOf(event.values[2])+"\nScreen brightness (float):
"+String.valueOf(layout.screenBrightness));
        }

        if (event.values[2]>9.6) {
//            Log.e("position: ","Phone is lying on the back");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 0;
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: flat on the back\ny =
"+String.valueOf(event.values[1])+"\nz =
"+String.valueOf(event.values[2])+"\nScreen brightness (float):
"+String.valueOf(layout.screenBrightness));
        }
    }
```

8) Flashlight activity

In this activity a click on imageButton is being listened. Once an action is made, a torch’ light state is changed.

```
void runFlashlight() {
    switchOff = findViewById(R.id.switchOff);
    switchOn = findViewById(R.id.switch_on);

    switchOff.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switchOff.setVisibility(View.GONE);
            switchOn.setVisibility(View.VISIBLE);

            CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
            try {
                String cameraId= cameraManager.getCameraIdList()[0];
                cameraManager.setTorchMode(cameraId, true);
            } catch (CameraAccessException e) {
                e.printStackTrace();
            }
        }
    });

    switchOn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            switchOff.setVisibility(View.VISIBLE);
            switchOn.setVisibility(View.GONE);

            CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
            try {
                String cameraId= cameraManager.getCameraIdList()[0];
                cameraManager.setTorchMode(cameraId, false);
            } catch (CameraAccessException e) {
                e.printStackTrace();
            }
        }
    });
}
```

9) Daylight sensor activity

In this activity the application uses daylight sensor to determine the light level. It is displayed in a TextView together with a message whether it is daytime or night time

```
    @Override
    public void onSensorChanged(SensorEvent event) {
        TextView brightnessTextView = (TextView)
findViewById(R.id.brightnessTextView);
        if (event.values[1]>9.6) {
//            Log.e("position: ", "vertical");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 1F;
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: vertical\ny =
"+String.valueOf(event.values[1])+"\nz =
"+String.valueOf(event.values[2])+"\nScreen brightness (float):
"+String.valueOf(layout.screenBrightness));
        }

        if (event.values[2]>9.6) {
//            Log.e("position: ","Phone is lying on the back");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 0;
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: flat on the back\ny =
"+String.valueOf(event.values[1])+"\nz =
"+String.valueOf(event.values[2])+"\nScreen brightness (float):
"+String.valueOf(layout.screenBrightness));
        }
```

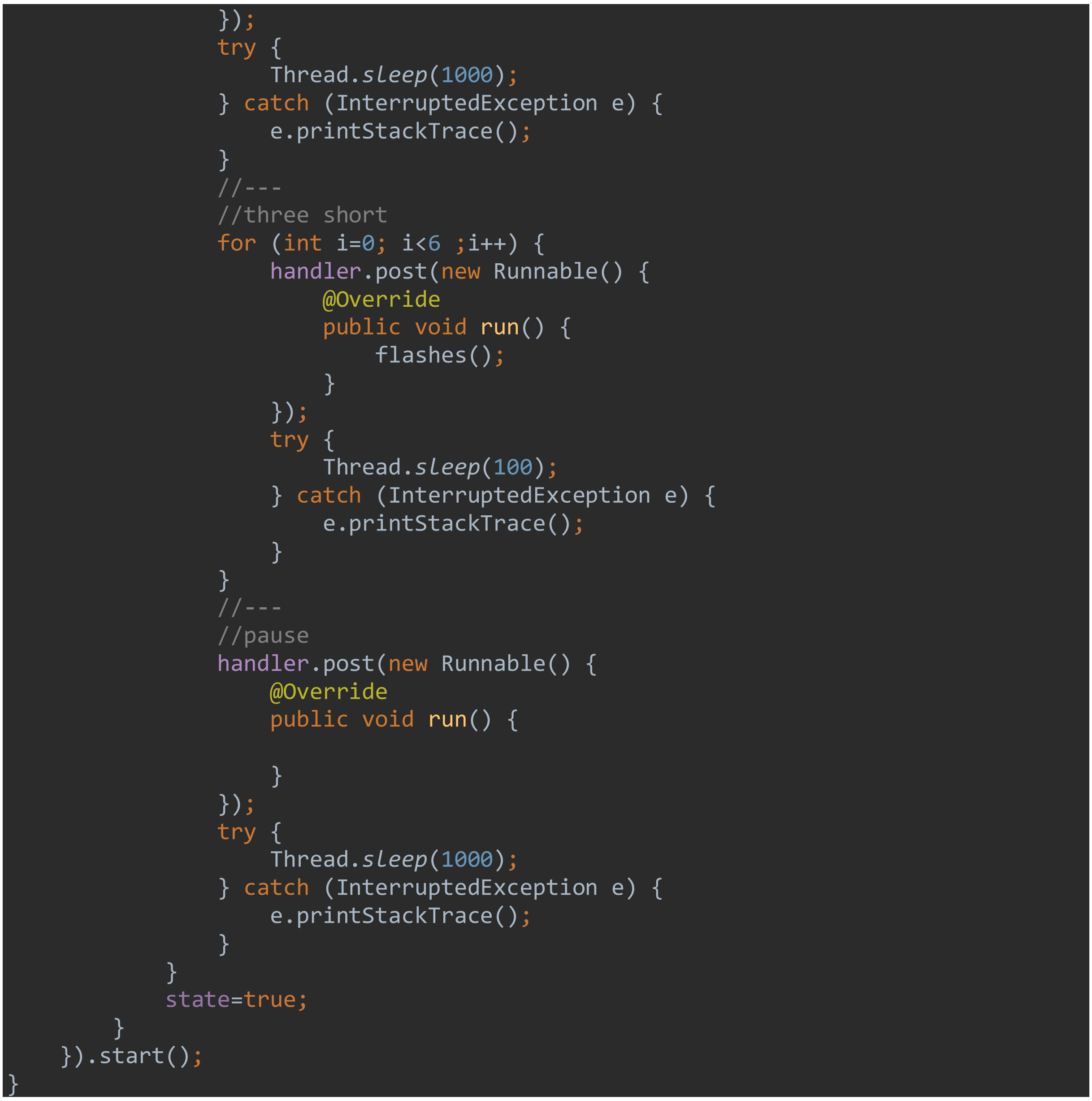
10) Compass activity

In this activity the application listens on magnetic field sensor. Once data readings are changed, a compass image is rotated accordingly.

```
@Override
public void onSensorChanged(SensorEvent event) {
    final float alpha = 0.97f;
    synchronized (this) {
        if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            mGeomagnetic[0]=alpha*mGeomagnetic[0]+(1-alpha)*event.values[0];
            mGeomagnetic[1]=alpha*mGeomagnetic[1]+(1-alpha)*event.values[1];
            mGeomagnetic[2]=alpha*mGeomagnetic[2]+(1-alpha)*event.values[2];
        }

        float R[] = new float[9];
        float I[] = new float[9];
        boolean success = SensorManager.getRotationMatrix(R,I, mGravity,
mGeomagnetic);
        if (success) {
            float orientation[] = new float[3];
            SensorManager.getOrientation(R,orientation);
            azimuth = (float) Math.toDegrees(orientation[0]);
            azimuth = (azimuth+360)%360;


            // animating
            Animation animation = new RotateAnimation(-correctAzimuth, -azimuth,
Animation.RELATIVE_TO_SELF, 0.5f,
                    Animation.RELATIVE_TO_SELF,0.5f);
            correctAzimuth = azimuth;

            animation.setDuration(500);
            animation.setRepeatCount(0);
            animation.setFillAfter(true);

            imageView.startAnimation(animation);
        }
    }
}
```

If a user is in danger, he can point compass to the North and the SOS signal will be displayed on a device's flashlight.

```
if (state) {
    state=false;
    final Handler handler = new Handler();
    new Thread(new Runnable() {
        @Override
        public void run() {
            if (azimuth<10 || azimuth>350) {
                //three short
                for (int i=0; i<6 ;i++) {
                    handler.post(new Runnable() {
                        @Override
                        public void run() {
                            flashes();
                        }
                    });
                    try {
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                //---
                //pause
                handler.post(new Runnable() {
                    @Override
                    public void run() {

                    }
                });
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                //---
                //three long
                for (int i=0; i<6 ;i++) {
                    handler.post(new Runnable() {
                        @Override
                        public void run() {
                            flashes();
                        }
                    });
                    try {
                        Thread.sleep(500);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                //---
                //pause
                handler.post(new Runnable() {
                    @Override
                    public void run() {

                    }
```

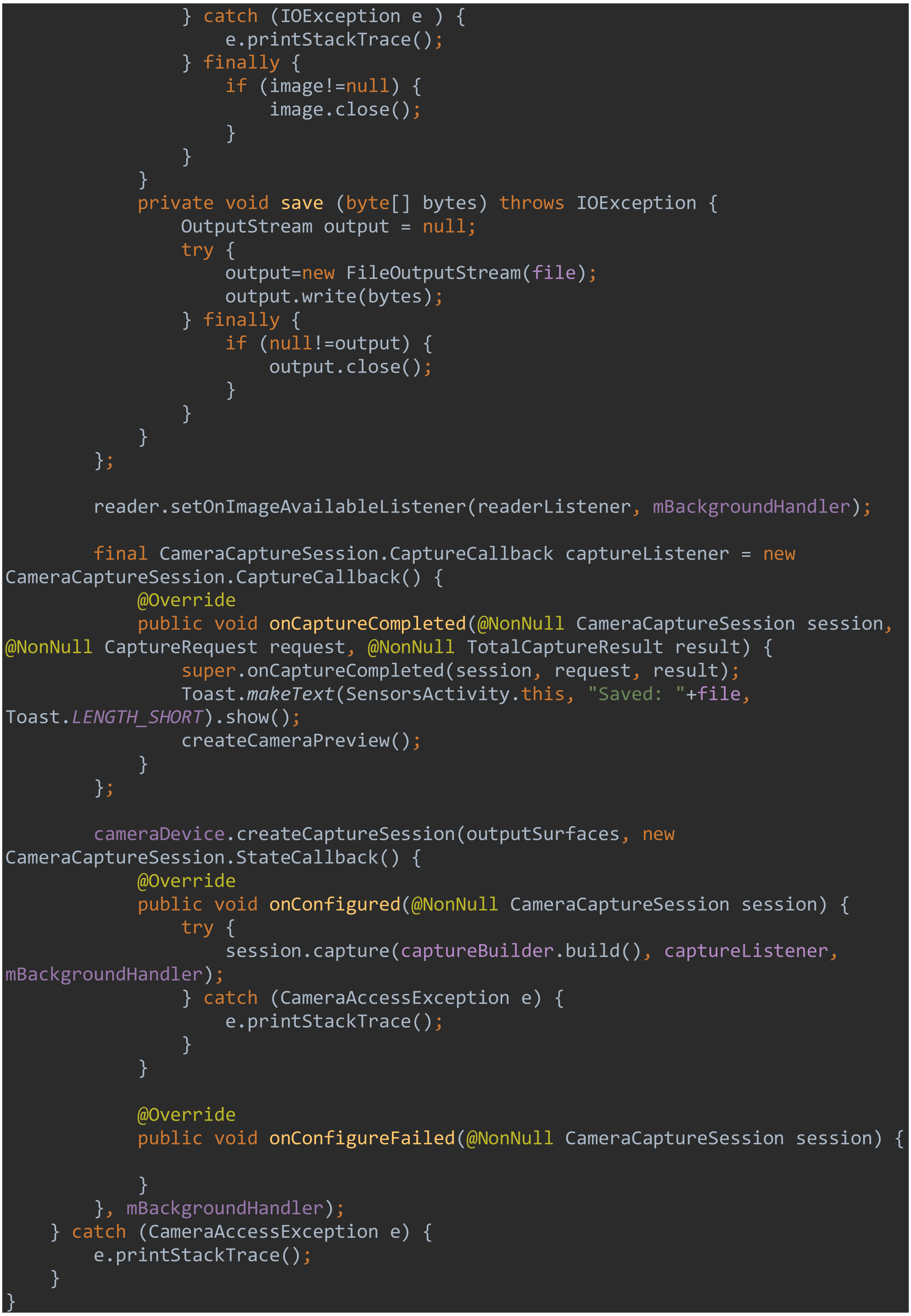


```
                });
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                //---
                //three short
                for (int i=0; i<6 ;i++) {
                    handler.post(new Runnable() {
                        @Override
                        public void run() {
                            flashes();
                        }
                    });
                    try {
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                //---
                //pause
                handler.post(new Runnable() {
                    @Override
                    public void run() {

                    }
                });
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            state=true;
        }
    }).start();
}
```

11) Camera activity

In this activity a user is able to take pictures (for example of items he is taking on a journey). Once a button is clicked, the image is saved on device’s local storage.

```
private void takePicture() {
    if (cameraDevice==null) {
        Log.e(TAG, "CameraDevice is null");
        return;
    }

    CameraManager manager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);

    try {
        CameraCharacteristics characteristics =
manager.getCameraCharacteristics(cameraDevice.getId());
        Size[] jpegSizes = null;
        if (characteristics!=null) {
            jpegSizes =
characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP).getOutp
utSizes(ImageFormat.JPEG);
        }
        int width = 640;
        int height = 480;
        if (jpegSizes!=null && 0<jpegSizes.length) {
            width=jpegSizes[0].getWidth();
            height=jpegSizes[0].getHeight();
        }
        ImageReader reader = ImageReader.newInstance(width, height,
ImageFormat.JPEG, 1);
        List<Surface> outputSurfaces = new ArrayList<Surface>(2);
        outputSurfaces.add(reader.getSurface());
        outputSurfaces.add(new Surface(textureView.getSurfaceTexture()));
        final CaptureRequest.Builder captureBuilder =
cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_STILL_CAPTURE);
        captureBuilder.addTarget(reader.getSurface());
        captureBuilder.set(CaptureRequest.CONTROL_MODE,
CameraMetadata.CONTROL_MODE_AUTO);
        int rotation = getWindowManager().getDefaultDisplay().getRotation();
        captureBuilder.set(CaptureRequest.JPEG_ORIENTATION,
ORIENTATIONS.get(rotation));
        final File file = new
File(Environment.getExternalStorageDirectory()+"/pic.jpg");

        ImageReader.OnImageAvailableListener readerListener = new
ImageReader.OnImageAvailableListener() {
            @Override
            public void onImageAvailable(ImageReader reader) {
                Image image = null;
                try {
                    image = reader.acquireLatestImage();
                    ByteBuffer buffer =image.getPlanes()[0].getBuffer();
                    byte[] bytes = new byte[buffer.capacity()];
                    buffer.get(bytes);
                    save(bytes);
                } catch (FileNotFoundException e ) {
                    e.printStackTrace();
```



```
                } catch (IOException e ) {
                    e.printStackTrace();
                } finally {
                    if (image!=null) {
                        image.close();
                    }
                }
            }
            private void save (byte[] bytes) throws IOException {
                OutputStream output = null;
                try {
                    output=new FileOutputStream(file);
                    output.write(bytes);
                } finally {
                    if (null!=output) {
                        output.close();
                    }
                }
            }
        };

        reader.setOnImageAvailableListener(readerListener, mBackgroundHandler);

        final CameraCaptureSession.CaptureCallback captureListener = new
CameraCaptureSession.CaptureCallback() {
            @Override
            public void onCaptureCompleted(@NonNull CameraCaptureSession session,
@NonNull CaptureRequest request, @NonNull TotalCaptureResult result) {
                super.onCaptureCompleted(session, request, result);
                Toast.makeText(SensorsActivity.this, "Saved: "+file,
Toast.LENGTH_SHORT).show();
                createCameraPreview();
            }
        };

        cameraDevice.createCaptureSession(outputSurfaces, new
CameraCaptureSession.StateCallback() {
            @Override
            public void onConfigured(@NonNull CameraCaptureSession session) {
                try {
                    session.capture(captureBuilder.build(), captureListener,
mBackgroundHandler);
                } catch (CameraAccessException e) {
                    e.printStackTrace();
                }
            }

            @Override
            public void onConfigureFailed(@NonNull CameraCaptureSession session) {

            }
        }, mBackgroundHandler);
    } catch (CameraAccessException e) {
        e.printStackTrace();
    }
}
```

Literature

https://developers.facebook.com/docs/android/getting-started/

https://youtu.be/6SrKOBV_hx8

https://youtu.be/urQp7KsQhW8

https://youtu.be/LXb3EKWsInQ

Lab1, lab2 and lab3 learning material

Defence

Lab1

To defend this project I was given three tasks regarding my items list in a backpack:

- Add a new item with a count on it
- Implement ability to remove item from a list
- Edit item from the list

To do this tasks I firstly created two EditText items (for an item and a count) and two buttons (to add/edit item and remove):

```
<EditText
    android:id="@+id/editTextEditItem"
    android:layout_marginTop="500dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="8"
    android:inputType="text"
    android:text="enter item"
    android:backgroundTint="@color/colorPrimary"
    android:focusable="true"    />

<EditText
    android:id="@+id/editTextTextEditCount"
    android:layout_marginTop="500dp"
    android:layout_marginLeft="180dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="4"
    android:inputType="text"
    android:text="count"
    android:backgroundTint="@color/colorPrimary"
    android:focusable="true"    />

<Button
    android:id="@+id/buttonEditItem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="500dp"
    android:layout_marginLeft="280dp"
    android:text="Add item"
    android:textAllCaps="false"
    android:textColor="#686868"
    android:textSize="15dp"    />

<Button
    android:id="@+id/buttonRemoveItem"
    android:layout_width="30dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="500dp"
    android:layout_marginLeft="378dp"
    android:text="-"
    android:textAllCaps="false"
    android:textColor="#686868"
    android:textSize="15dp"    />
```

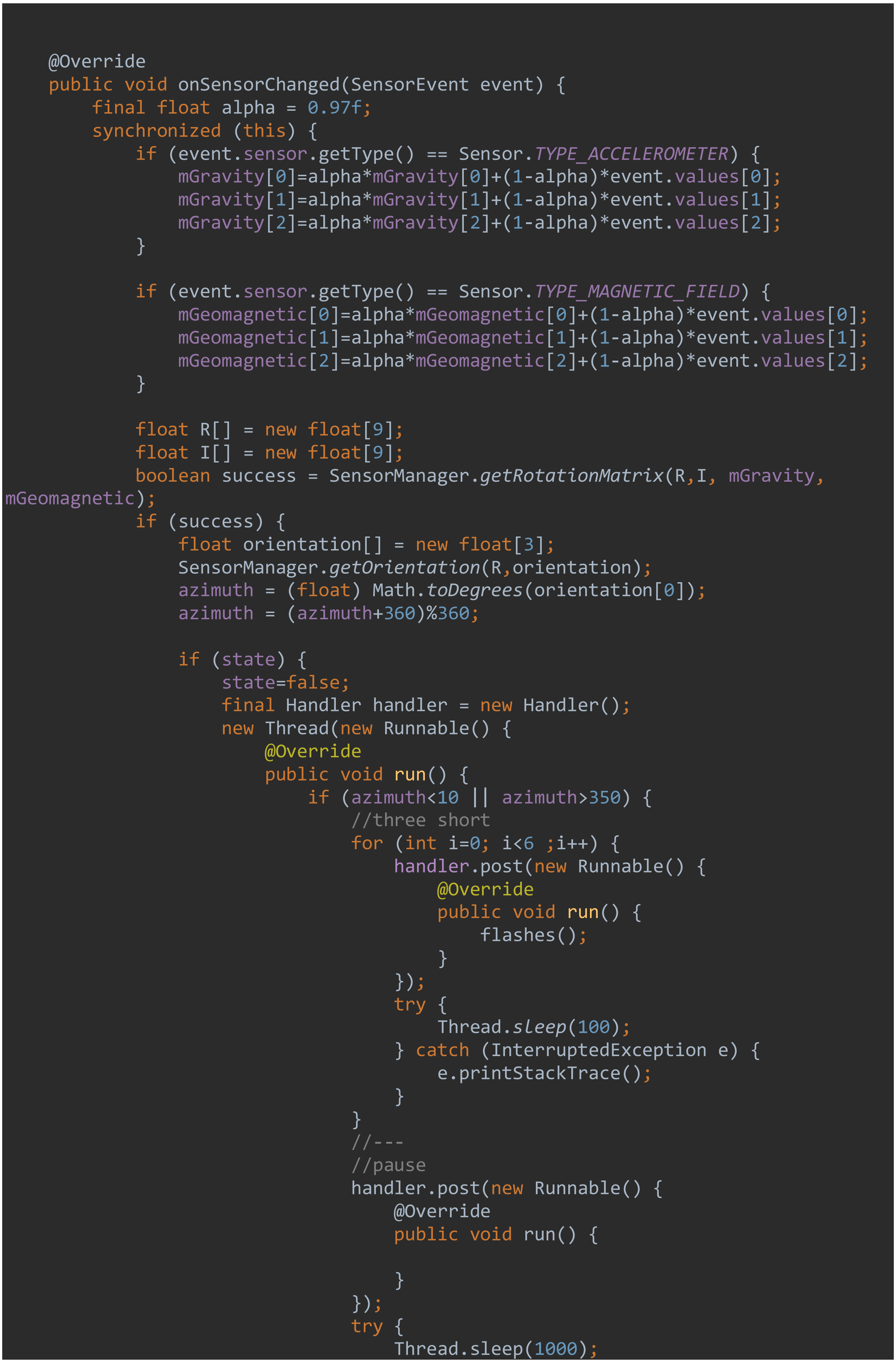
To achieve the functionality the code fragment was inserted:

```
final Button buttonRemoveItem = view.findViewById(R.id.buttonRemoveItem);
final Button buttonAddOrEditItem = view.findViewById(R.id.buttonEditItem);
final EditText editItem = view.findViewById(R.id.editTextEditItem);
final EditText editCount = view.findViewById(R.id.editTextTextEditCount);

buttonRemoveItem.setVisibility(View.GONE);

buttonRemoveItem.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        items = GetGeneratedItems(context);
        items.remove(itemSelected);
        SaveItemsToSharedMemory(items, context);
        createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
        //when remove done
        buttonRemoveItem.setVisibility(View.GONE);
        edit=false;
        editItem.setText("");
        editCount.setText("0");
        buttonAddOrEditItem.setText("Add item");
    }
});

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        itemSelected = parent.getItemAtPosition(position).toString();
        edit=true;
        buttonAddOrEditItem.setText("Apply");
        editItem.setText(itemSelected);
        editCount.setText("0");
        buttonRemoveItem.setVisibility(View.VISIBLE);
    }
});

buttonAddOrEditItem.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (edit) {
            String newValue = editItem.getText().toString();
            String newCount = editCount.getText().toString();
            items = GetGeneratedItems(context);
            items.remove(itemSelected);
            items.add(newValue+"; count: "+newCount);
            SaveItemsToSharedMemory(items, context);
            createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
        } else {
            String addedItem = editItem.getText().toString();
            String addedCount = editCount.getText().toString();
            items = GetGeneratedItems(context);
            items.add(addedItem+"; count: "+addedCount);
            SaveItemsToSharedMemory(items, context);
            createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
```

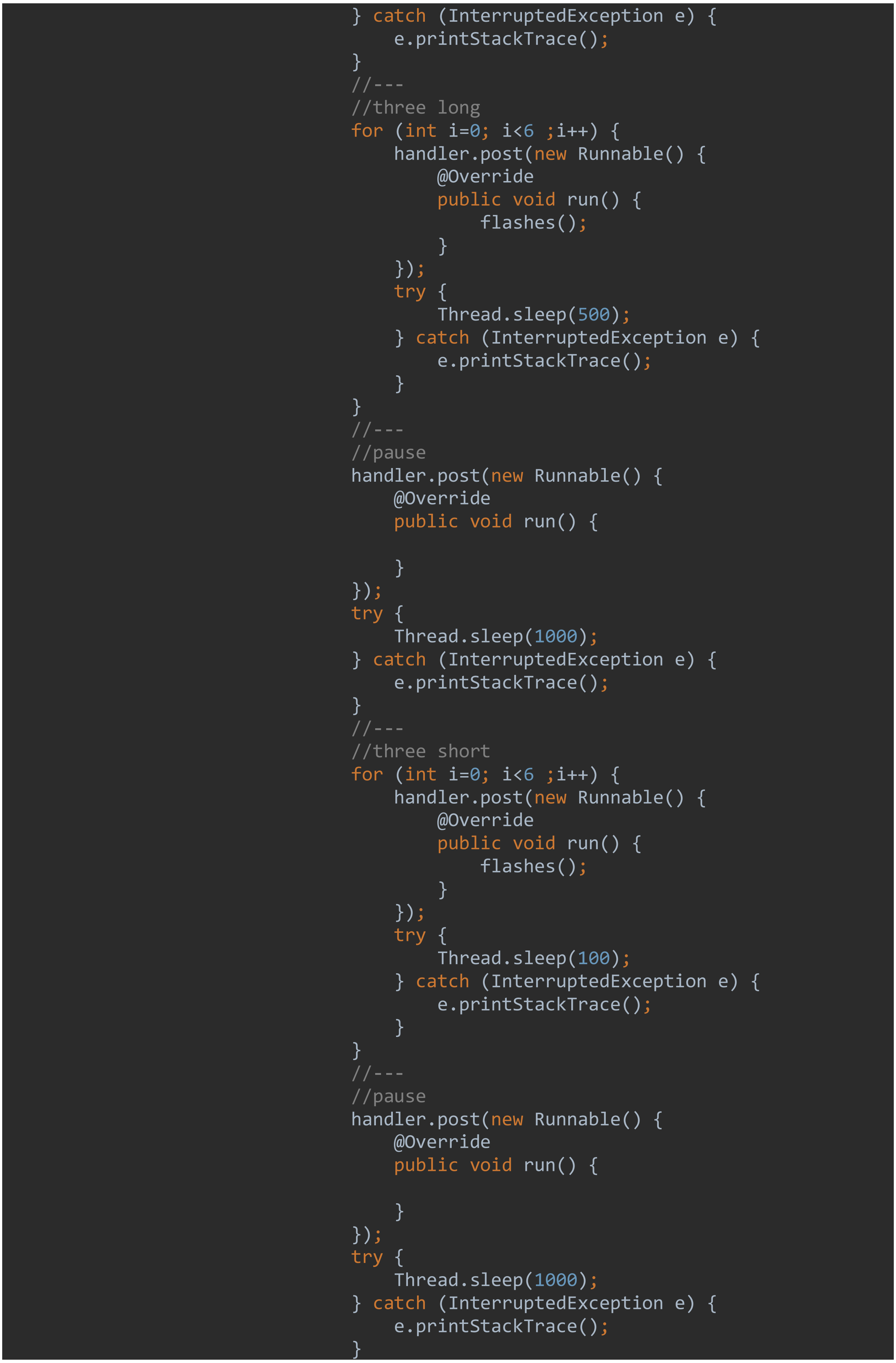
```
        }

        //uzbaigus
        buttonAddOrEditItem.setText("Add item");
        edit=false;
    }
});

editItem.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!edit) {
            editItem.setText("");
        }
    }
});

editCount.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!edit) {
            editCount.setText("");
        }
    }
});
```

Compass activity

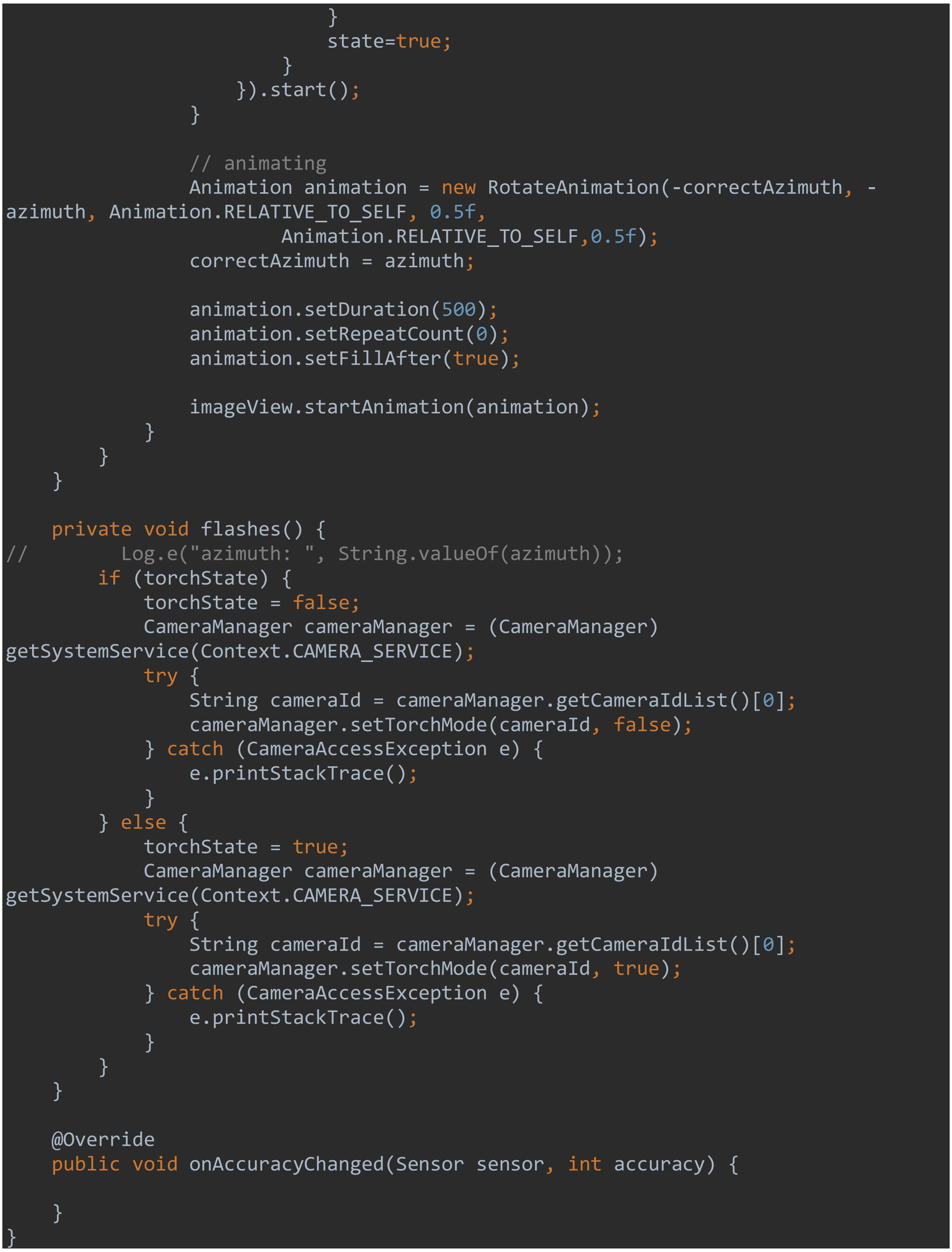
```
package com.example.journeyapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraManager;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.animation.Animation;
import android.view.animation.RotateAnimation;
import android.widget.ImageView;

public class CompassActivity extends AppCompatActivity implements
SensorEventListener {
    private boolean state=true;
    private boolean torchState=false;

    private ImageView imageView;
    private float[] mGravity = new float[3];
    private float[] mGeomagnetic  = new float[3];
    private float azimuth = 0f;
    private float correctAzimuth = 0f;
    private SensorManager sensorManager;

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

        imageView = (ImageView) findViewById(R.id.compass);
        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    }

    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this,
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD),
                SensorManager.SENSOR_DELAY_GAME);
        sensorManager.registerListener(this,
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
                SensorManager.SENSOR_DELAY_GAME);
    }

    @Override
    protected void onPause() {
        super.onPause();

        sensorManager.unregisterListener(this);

    }
```



```
    @Override
    public void onSensorChanged(SensorEvent event) {
        final float alpha = 0.97f;
        synchronized (this) {
            if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
                mGravity[0]=alpha*mGravity[0]+(1-alpha)*event.values[0];
                mGravity[1]=alpha*mGravity[1]+(1-alpha)*event.values[1];
                mGravity[2]=alpha*mGravity[2]+(1-alpha)*event.values[2];
            }

            if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
                mGeomagnetic[0]=alpha*mGeomagnetic[0]+(1-alpha)*event.values[0];
                mGeomagnetic[1]=alpha*mGeomagnetic[1]+(1-alpha)*event.values[1];
                mGeomagnetic[2]=alpha*mGeomagnetic[2]+(1-alpha)*event.values[2];
            }

            float R[] = new float[9];
            float I[] = new float[9];
            boolean success = SensorManager.getRotationMatrix(R,I, mGravity,
mGeomagnetic);
            if (success) {
                float orientation[] = new float[3];
                SensorManager.getOrientation(R,orientation);
                azimuth = (float) Math.toDegrees(orientation[0]);
                azimuth = (azimuth+360)%360;

                if (state) {
                    state=false;
                    final Handler handler = new Handler();
                    new Thread(new Runnable() {
                        @Override
                        public void run() {
                            if (azimuth<10 || azimuth>350) {
                                //three short
                                for (int i=0; i<6 ;i++) {
                                    handler.post(new Runnable() {
                                        @Override
                                        public void run() {
                                            flashes();
                                        }
                                    });
                                    try {
                                        Thread.sleep(100);
                                    } catch (InterruptedException e) {
                                        e.printStackTrace();
                                    }
                                }
                                //---
                                //pause
                                handler.post(new Runnable() {
                                    @Override
                                    public void run() {

                                    }
                                });
                                try {
                                    Thread.sleep(1000);
```



```
} catch (InterruptedException e) {
    e.printStackTrace();
}
//---
//three long
for (int i=0; i<6 ;i++) {
    handler.post(new Runnable() {
        @Override
        public void run() {
            flashes();
        }
    });
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
//---
//pause
handler.post(new Runnable() {
    @Override
    public void run() {

    }
});
try {
    Thread.sleep(1000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
//---
//three short
for (int i=0; i<6 ;i++) {
    handler.post(new Runnable() {
        @Override
        public void run() {
            flashes();
        }
    });
    try {
        Thread.sleep(100);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
//---
//pause
handler.post(new Runnable() {
    @Override
    public void run() {

    }
});
try {
    Thread.sleep(1000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
```



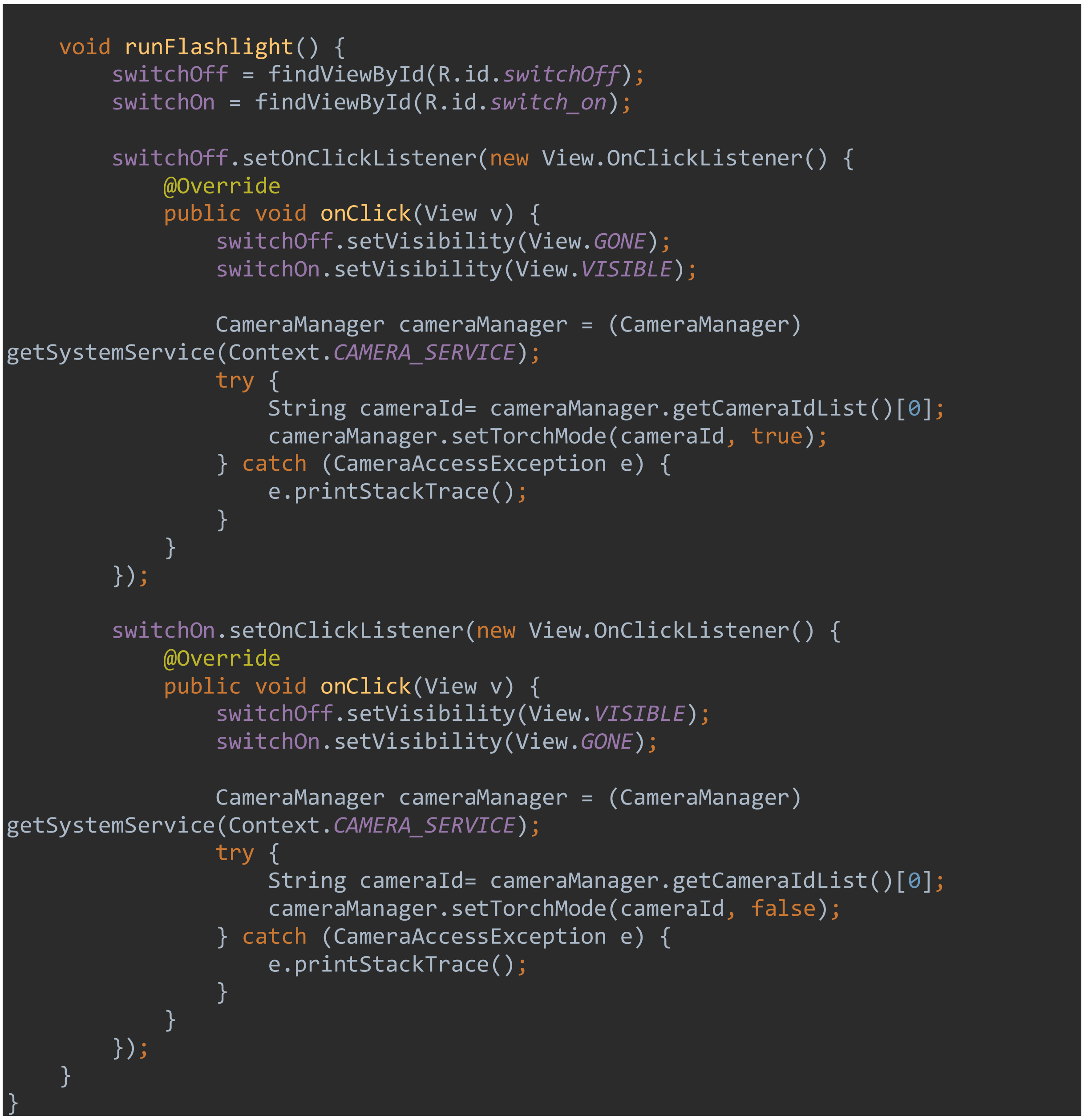
```
                            }
                            state=true;
                        }
                    }).start();
                }

                // animating
                Animation animation = new RotateAnimation(-correctAzimuth, -
azimuth, Animation.RELATIVE_TO_SELF, 0.5f,
                        Animation.RELATIVE_TO_SELF,0.5f);
                correctAzimuth = azimuth;

                animation.setDuration(500);
                animation.setRepeatCount(0);
                animation.setFillAfter(true);

                imageView.startAnimation(animation);
            }
        }
    }

    private void flashes() {
//          Log.e("azimuth: ", String.valueOf(azimuth));
        if (torchState) {
            torchState = false;
            CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
            try {
                String cameraId = cameraManager.getCameraIdList()[0];
                cameraManager.setTorchMode(cameraId, false);
            } catch (CameraAccessException e) {
                e.printStackTrace();
            }
        } else {
            torchState = true;
            CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
            try {
                String cameraId = cameraManager.getCameraIdList()[0];
                cameraManager.setTorchMode(cameraId, true);
            } catch (CameraAccessException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
}
```

Daylight activity

```
package com.example.journeyapp;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Service;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextView;

import org.w3c.dom.Text;

import java.security.Provider;

public class DaylightSensorActivity extends AppCompatActivity implements
SensorEventListener {

    TextView textView;
    TextView dayOrNightText;

    SensorManager sensorManager;
    Sensor sensor;

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

        textView = findViewById(R.id.daylightTextView);
        dayOrNightText = findViewById(R.id.dayOrNightTextView);

        sensorManager = (SensorManager) getSystemService(Service.SENSOR_SERVICE);
        sensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this, sensor,
SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType()==Sensor.TYPE_LIGHT) {
            textView.setText("Light level: "+event.values[0]);
            if (event.values[0]>1) {
                dayOrNightText.setText("It is day time :)");
            } else {
                dayOrNightText.setText("It is night time. Be careful!");
            }
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
}
```

Flashlight activity

```
package com.example.journeyapp;

import androidx.appcompat.app.AppCompatActivity;


import android.Manifest;
import android.content.Context;
import android.graphics.Camera;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraManager;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;

import com.karumi.dexter.Dexter;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.single.PermissionListener;

import dalvik.system.DexClassLoader;

public class FlashlightActivity extends AppCompatActivity {

    ImageButton switchOff, switchOn;


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


Dexter.withContext(this).withPermission(Manifest.permission.CAMERA).withListener(n
ew PermissionListener() {
            @Override
            public void onPermissionGranted(PermissionGrantedResponse
permissionGrantedResponse) {
                runFlashlight();
            }

            @Override
            public void onPermissionDenied(PermissionDeniedResponse
permissionDeniedResponse) {
                Toast.makeText(FlashlightActivity.this, "Camera permission is
required", Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onPermissionRationaleShouldBeShown(PermissionRequest
permissionRequest, PermissionToken permissionToken) {

            }
        }).check();
    }
```



```
    void runFlashlight() {
        switchOff = findViewById(R.id.switchOff);
        switchOn = findViewById(R.id.switch_on);

        switchOff.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switchOff.setVisibility(View.GONE);
                switchOn.setVisibility(View.VISIBLE);

                CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
                try {
                    String cameraId= cameraManager.getCameraIdList()[0];
                    cameraManager.setTorchMode(cameraId, true);
                } catch (CameraAccessException e) {
                    e.printStackTrace();
                }
            }
        });

        switchOn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switchOff.setVisibility(View.VISIBLE);
                switchOn.setVisibility(View.GONE);

                CameraManager cameraManager = (CameraManager)
getSystemService(Context.CAMERA_SERVICE);
                try {
                    String cameraId= cameraManager.getCameraIdList()[0];
                    cameraManager.setTorchMode(cameraId, false);
                } catch (CameraAccessException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Geolocation activity

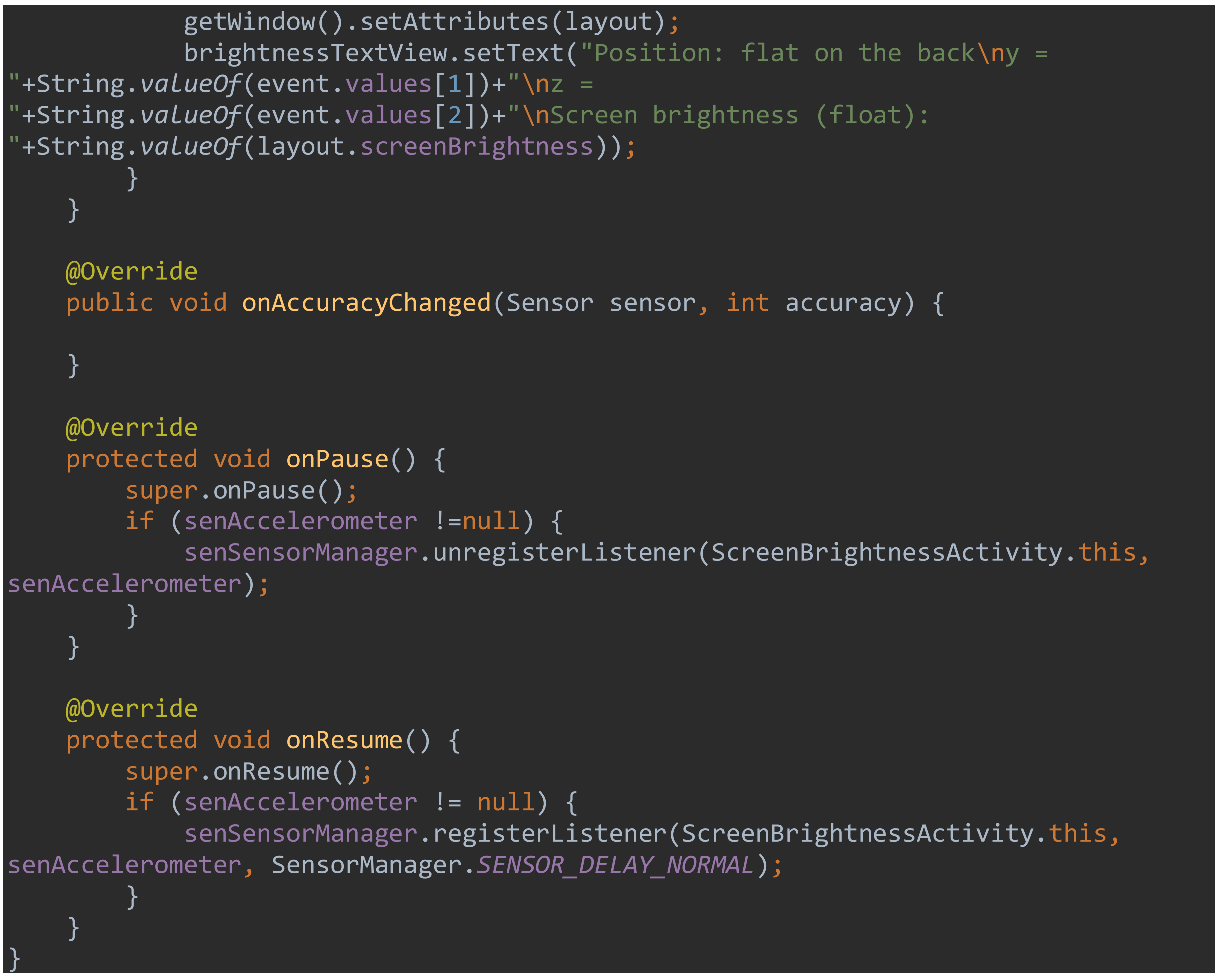
```
package com.example.journeyapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;
import androidx.fragment.app.FragmentFactory;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.content.res.ColorStateList;
import android.graphics.Color;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

public class GeolocationActivity extends FragmentActivity implements
OnMapReadyCallback {

    Location currentLocation;
    Location networkLocation;
    LocationManager locationManagerNetwork;
    SupportMapFragment supportMapFragment;
    private static final int REQUEST_CODE = 101;

    private TextView geolocationTextView;
    private String latitude = "0.0";
    private String longtitude = "0.0";

    private FusedLocationProviderClient fusedLocationProviderClient;

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

        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION},
                PackageManager.PERMISSION_GRANTED);

        geolocationTextView = (TextView) findViewById(R.id.geolocationTextView);
        locationManagerNetwork = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
```

```
        fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
            return;
        }
        fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this,
new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                if (location != null) {
                    String gpsLatitude = Double.toString(location.getLatitude());
                    String gpsLongtitude =
Double.toString(location.getLongitude());

                    String networkLat = "0";
                    String networkLong = "0";

                    if
(locationManagerNetwork.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
                        if
(ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED
&&

ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                            location =
locationManagerNetwork.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                            networkLat = Double.toString(location.getLatitude());
                            networkLong =
Double.toString(location.getLongitude());
                        }
                    }

                    geolocationTextView.setText("Gps latitude: " + gpsLatitude +
"\nGps longtitude: " + gpsLongtitude+"\nNetwork lat: "+networkLat+"\nNetwork long:
"+networkLong);
                    geolocationTextView.setTextSize(20);
                    geolocationTextView.setTextColor(Color.RED);
                }
            }
        });

        fetchLastLocation();
    }

    private void fetchLastLocation() {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
            ActivityCompat.requestPermissions(this, new String[]
                    {Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_CODE);
            return;
        }
        Task<Location> task = fusedLocationProviderClient.getLastLocation();
        task.addOnSuccessListener(new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                if (location!=null) {
                    currentLocation = location;
                    supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.google_map);
                    supportMapFragment.getMapAsync(GeolocationActivity.this);

                    if
(locationManagerNetwork.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
                        if
(ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED
&&

ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                            networkLocation =
locationManagerNetwork.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                        }
                    }
                }
            }
        });
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        LatLng latLng = new LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude());
        MarkerOptions markerOptions = new
MarkerOptions().position(latLng).title("My Gps existance is found here");
        googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 5));
        googleMap.addMarker((markerOptions));
        //---

        latLng = new LatLng(networkLocation.getLatitude(),
networkLocation.getLongitude());
        markerOptions = new MarkerOptions().position(latLng).title("My network
existance is found here");
        googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 5));
        googleMap.addMarker((markerOptions));
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
            case REQUEST_CODE:
                if (grantResults.length>0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
```

```
                    fetchLastLocation();
                }
                break;
        }
    }
}
```

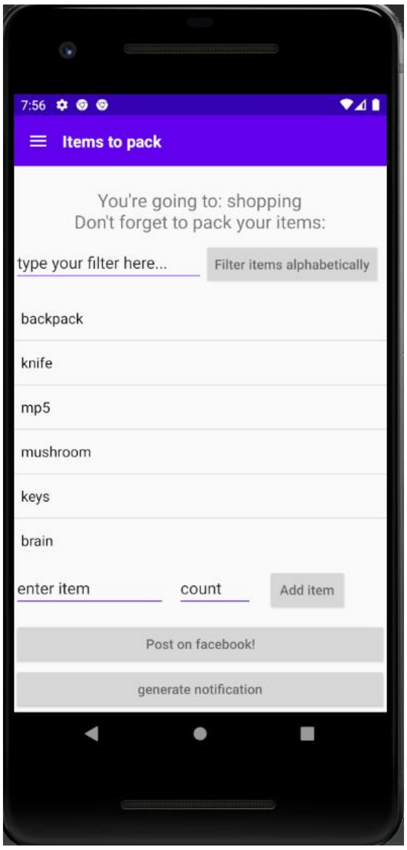
Screen brightness activity

```
package com.example.journeyapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.media.audiofx.Equalizer;
import android.os.Bundle;
import android.os.PowerManager;
import android.provider.Settings;
import android.util.Log;
import android.view.WindowManager;
import android.widget.TextView;

public class ScreenBrightnessActivity extends AppCompatActivity implements
SensorEventListener {

    private SensorManager senSensorManager;
    private Sensor senAccelerometer;

    int maxBrightness =0;

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

        senSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        senAccelerometer =
senSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        senSensorManager.registerListener(ScreenBrightnessActivity.this,
senAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        TextView brightnessTextView = (TextView)
findViewById(R.id.brightnessTextView);
        if (event.values[1]>9.6) {
//            Log.e("position: ", "vertical");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 1F;
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: vertical\ny =
"+String.valueOf(event.values[1])+"\nz =
"+String.valueOf(event.values[2])+"\nScreen brightness (float):
"+String.valueOf(layout.screenBrightness));
        }

        if (event.values[2]>9.6) {
//            Log.e("position: ","Phone is lying on the back");
            WindowManager.LayoutParams layout = getWindow().getAttributes();
            layout.screenBrightness = 0;
```



```
            getWindow().setAttributes(layout);
            brightnessTextView.setText("Position: flat on the back\ny = 
"+String.valueOf(event.values[1])+"\nz = 
"+String.valueOf(event.values[2])+"\nScreen brightness (float): 
"+String.valueOf(layout.screenBrightness));
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }

    @Override
    protected void onPause() {
        super.onPause();
        if (senAccelerometer !=null) {
            senSensorManager.unregisterListener(ScreenBrightnessActivity.this, 
senAccelerometer);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (senAccelerometer != null) {
            senSensorManager.registerListener(ScreenBrightnessActivity.this, 
senAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
        }
    }
}
```

The result:

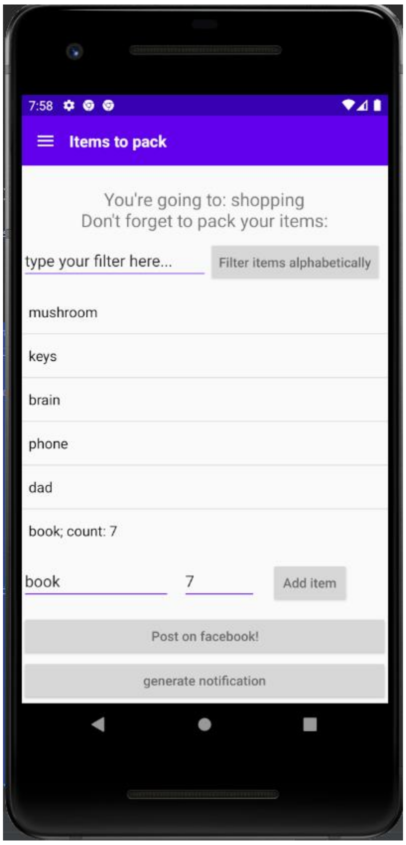


You're going to: shopping
Don't forget to pack your items:

type your filter here...

backpack
knife
mp5
mushroom
keys
brain

enter item
count



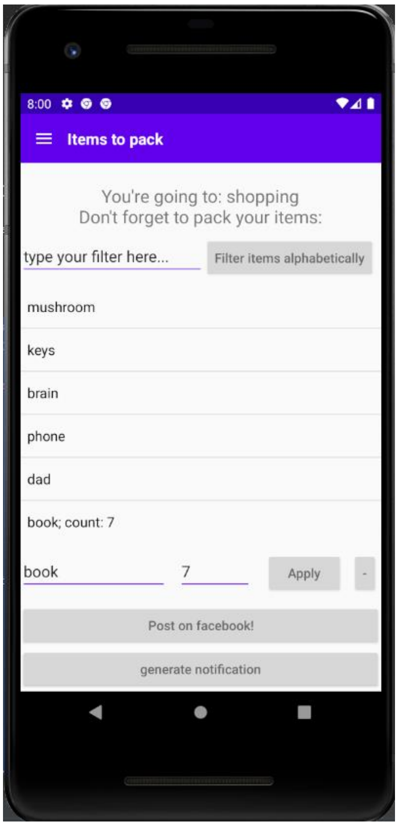
You're going to: shopping
Don't forget to pack your items:

type your filter here...

mushroom
keys
brain
phone
dad
book; count: 7

book
7

When item is clicked on the list, the user is able to edit or remove it



You're going to: shopping
Don't forget to pack your items:

type your filter here...

mushroom

keys

brain

phone

dad

book; count: 7

book

7

Evaluation: 10

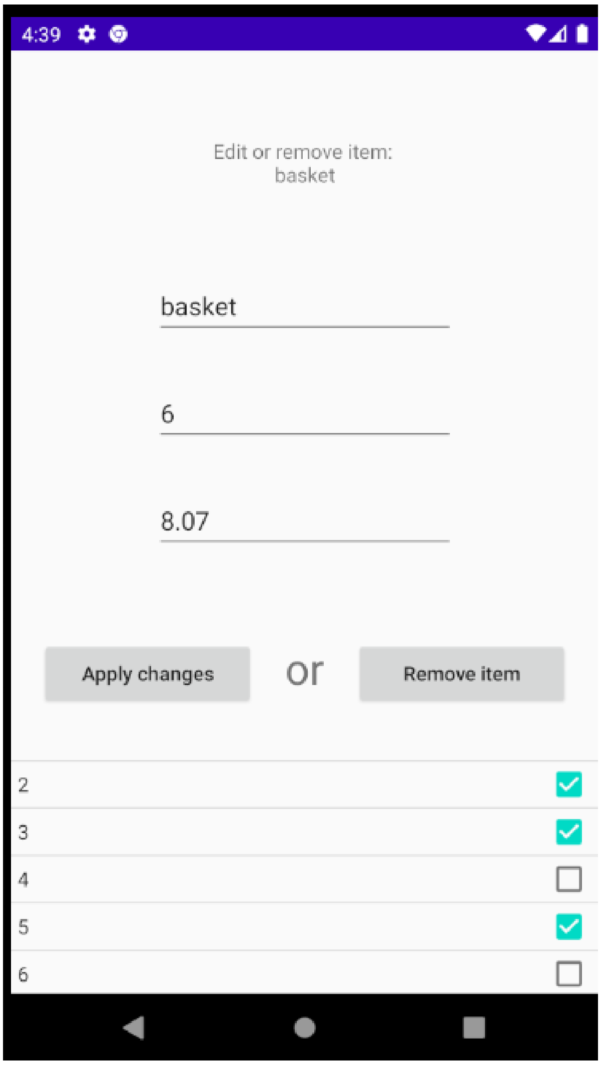
Lab2

During the defence I was given a task to make a checkbox list when pressing on backpack‘s item.

To achieve this I created a layout of CheckedTextView

```xml
<?xml version="1.0" encoding="utf-8"?>
    <CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/txt_lan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:checkMark="?android:attr/listChoiceIndicatorMultiple"
        android:padding="5dp"/>
```

That was used to generate my list‘s rows.



Edit or remove item:
basket

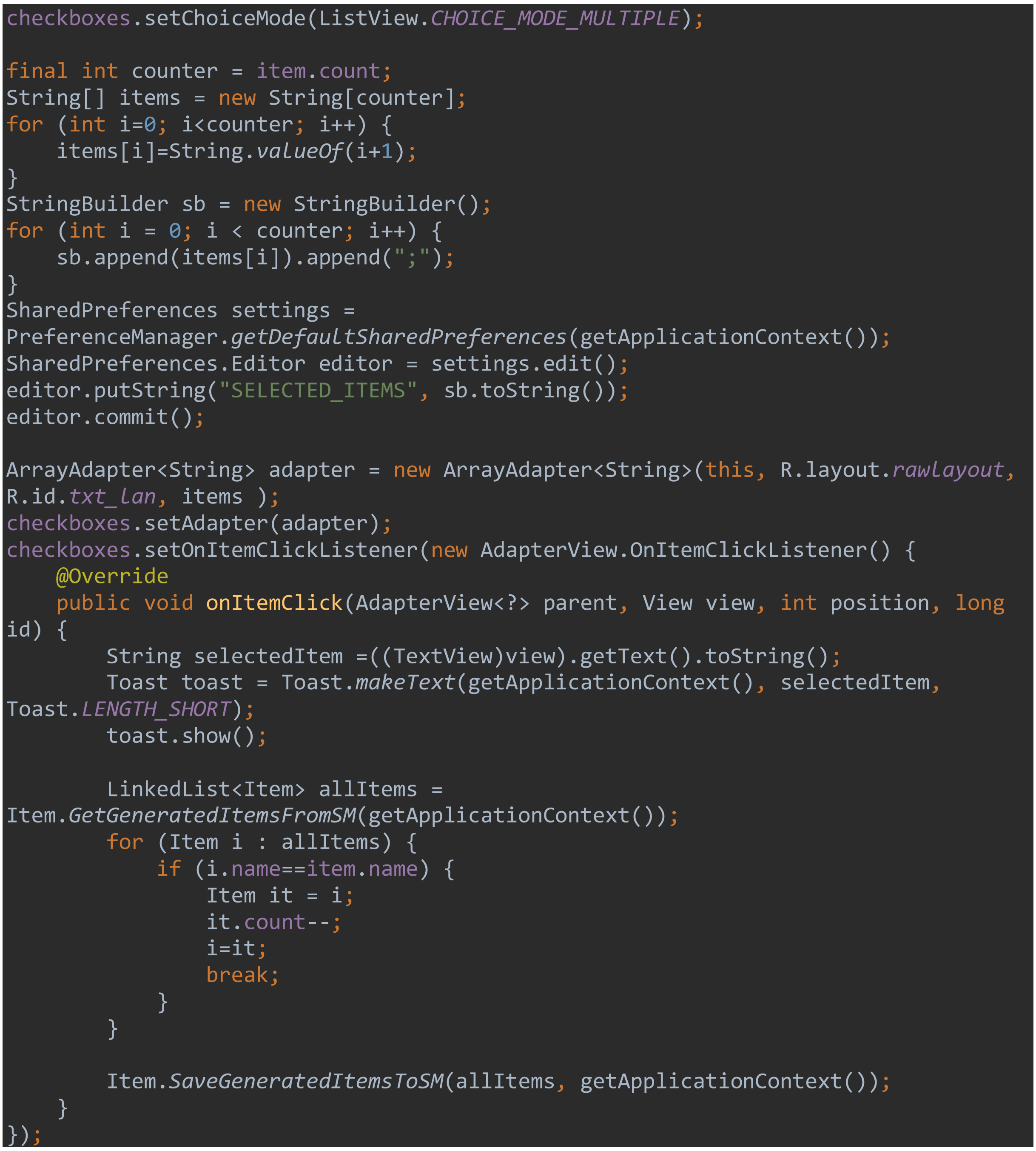
basket

6

8.07

2 ☑
3 ☑
4 ☐
5 ☑
6 ☐

Figure 18 CheckBox list



```
checkboxes.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);

final int counter = item.count;
String[] items = new String[counter];
for (int i=0; i<counter; i++) {
    items[i]=String.valueOf(i+1);
}
StringBuilder sb = new StringBuilder();
for (int i = 0; i < counter; i++) {
    sb.append(items[i]).append(";");
}
SharedPreferences settings =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
SharedPreferences.Editor editor = settings.edit();
editor.putString("SELECTED_ITEMS", sb.toString());
editor.commit();

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.rawlayout,
R.id.txt_lan, items );
checkboxes.setAdapter(adapter);
checkboxes.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        String selectedItem =((TextView)view).getText().toString();
        Toast toast = Toast.makeText(getApplicationContext(), selectedItem,
Toast.LENGTH_SHORT);
        toast.show();

        LinkedList<Item> allItems =
Item.GetGeneratedItemsFromSM(getApplicationContext());
        for (Item i : allItems) {
            if (i.name==item.name) {
                Item it = i;
                it.count--;
                i=it;
                break;
            }
        }

        Item.SaveGeneratedItemsToSM(allItems, getApplicationContext());
    }
});
```

Evaluation: 5

Lab3

Defence task: When a picture is taken with a camera, set its image on one of your activity backgrounds. When a new pic is taken, the background has to update. To get max points, you have to reset the background back to default after 5 seconds. There should be an indicator which informs when the background will be reset.

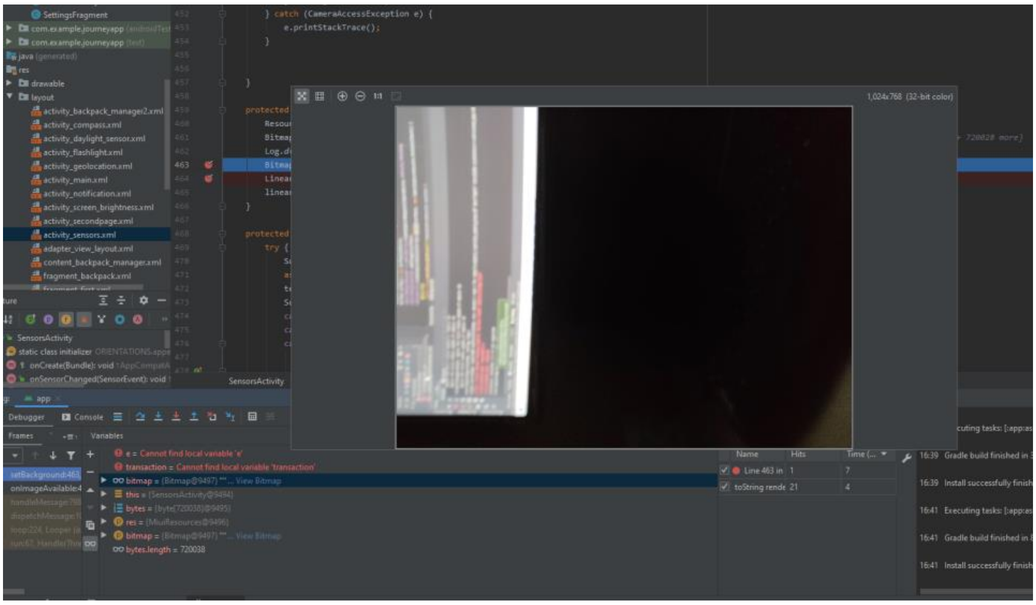
To achieve this task the following code was implemented:

```
ImageReader.OnImageAvailableListener readerListener = new
ImageReader.OnImageAvailableListener() {
    @Override
    public void onImageAvailable(ImageReader reader) {
        Image image = null;
        try {
            image = reader.acquireLatestImage();
            ByteBuffer buffer = image.getPlanes()[0].getBuffer();
            byte[] bytes = new byte[buffer.capacity()];
            buffer.get(bytes);

            setBackground(bytes);
```

We pass bytes captured by the camera to setBackground method:

```
protected void setBackground(byte[] bytes) {
    Resources res = getResources();
    Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
    Log.d("DEBUG", "hello "+bitmap+" hello");
    BitmapDrawable bd = new BitmapDrawable(res, bitmap);
    LinearLayout linearLayout = (LinearLayout) findViewById(R.id.layoutas);
    linearLayout.setBackground(bd);
}
```

This method coverts bytes to bitmap and bitmap to drawable object. The latter is set to the background image.



```
} catch (CameraAccessException e) {
    e.printStackTrace();
}
```

```
Variables
e = Cannot find local variable 'e'
transaction = Cannot find local variable 'transaction'
bitmap = {Bitmap@9497} "" ... View Bitmap
this = {SensorsActivity@9494}
bytes = {byte[720038]@9495}
res = {MiuiResources@9496}
bitmap = {Bitmap@9497} "" ... View Bitmap
bytes.length = 720038
```

	Name	Hits	Time (...
☑	Line 463 in	1	7
☑	toString rend	21	4

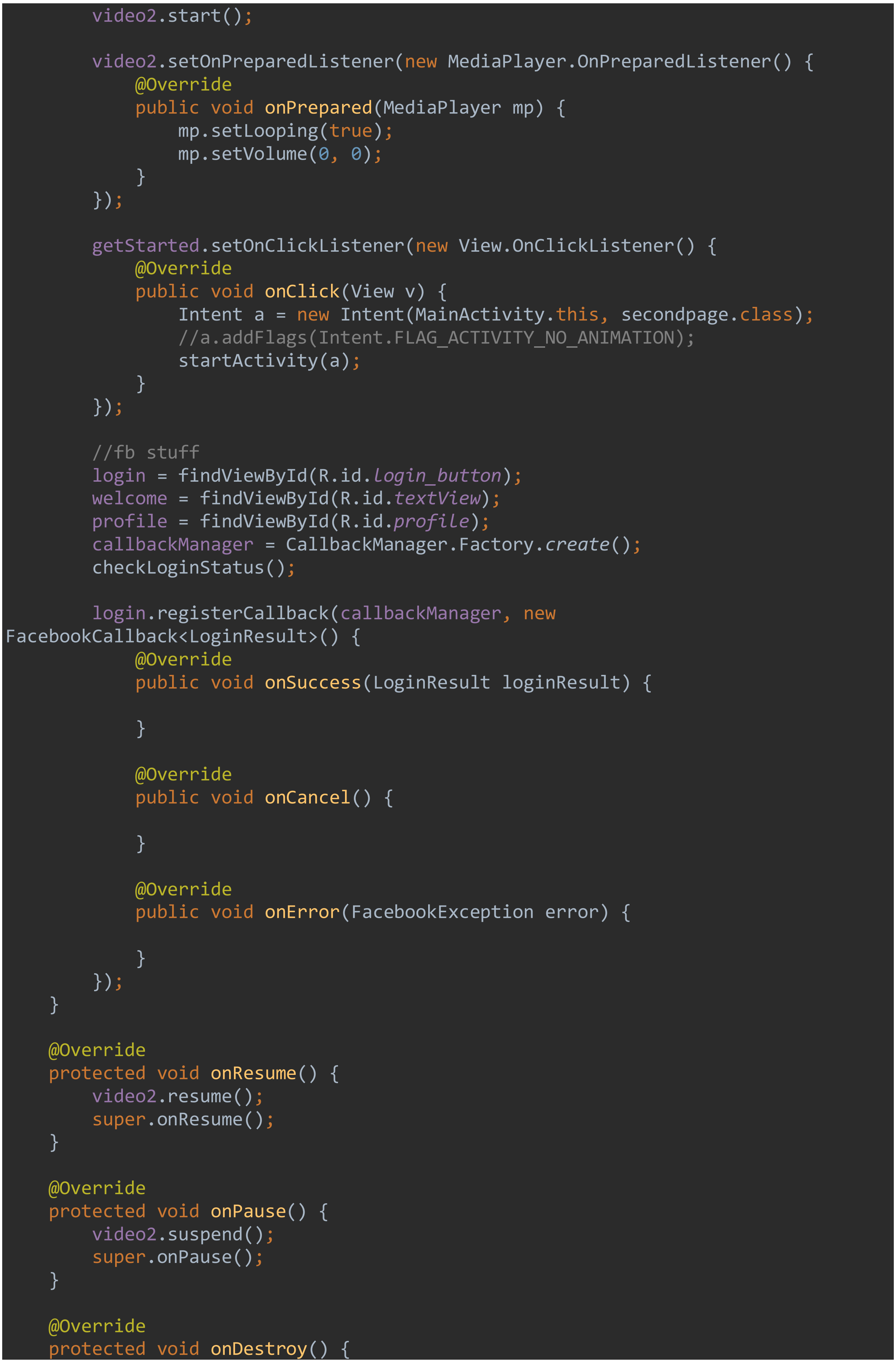
```
cuting tasks: [:app:as
16:39  Gradle build finished in
16:39  Install successfully finish
16:41  Executing tasks: [:app:as
16:41  Gradle build finished in
16:41  Install successfully finish
```

Total mark: 7

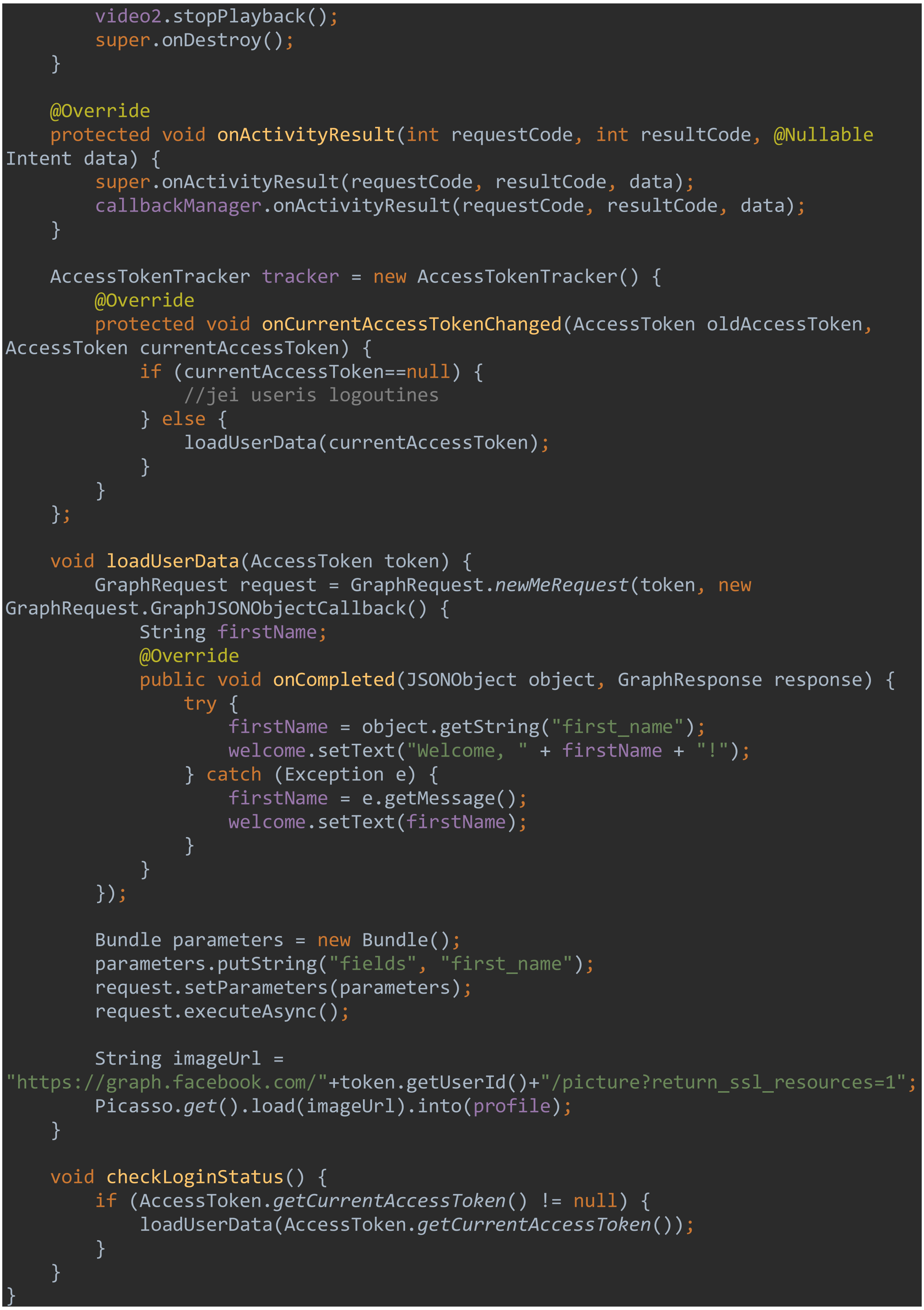
Source code

MainActivity

```
package com.example.journeyapp;

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

import android.content.Intent;
import android.media.FaceDetector;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.telecom.Call;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.VideoView;

import com.facebook.AccessToken;
import com.facebook.AccessTokenTracker;
import com.facebook.CallbackManager;
import com.facebook.FacebookCallback;
import com.facebook.FacebookException;
import com.facebook.GraphRequest;
import com.facebook.GraphResponse;
import com.facebook.login.LoginResult;
import com.facebook.login.widget.LoginButton;
import com.squareup.picasso.Picasso;

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

public class MainActivity extends AppCompatActivity {

    Button getStarted;
    VideoView video2;

    TextView welcome;
    ImageView profile;
    LoginButton login;
    CallbackManager callbackManager;

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

        getStarted=(Button)findViewById(R.id.getStarted);
        video2 = (VideoView)findViewById(R.id.video2);

        String path = "android.resource://com.example.journeyapp/"+R.raw.nature;
        Uri u = Uri.parse(path);
        video2.setVideoURI(u);
```



```
        video2.start();

        video2.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
            @Override
            public void onPrepared(MediaPlayer mp) {
                mp.setLooping(true);
                mp.setVolume(0, 0);
            }
        });

        getStarted.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent a = new Intent(MainActivity.this, secondpage.class);
                //a.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                startActivity(a);
            }
        });

        //fb stuff
        login = findViewById(R.id.login_button);
        welcome = findViewById(R.id.textView);
        profile = findViewById(R.id.profile);
        callbackManager = CallbackManager.Factory.create();
        checkLoginStatus();

        login.registerCallback(callbackManager, new
FacebookCallback<LoginResult>() {
            @Override
            public void onSuccess(LoginResult loginResult) {

            }

            @Override
            public void onCancel() {

            }

            @Override
            public void onError(FacebookException error) {

            }
        });
    }

    @Override
    protected void onResume() {
        video2.resume();
        super.onResume();
    }

    @Override
    protected void onPause() {
        video2.suspend();
        super.onPause();
    }

    @Override
    protected void onDestroy() {
```



```
        video2.stopPlayback();
        super.onDestroy();
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        callbackManager.onActivityResult(requestCode, resultCode, data);
    }

    AccessTokenTracker tracker = new AccessTokenTracker() {
        @Override
        protected void onCurrentAccessTokenChanged(AccessToken oldAccessToken,
AccessToken currentAccessToken) {
            if (currentAccessToken==null) {
                //jei useris logoutines
            } else {
                loadUserData(currentAccessToken);
            }
        }
    };

    void loadUserData(AccessToken token) {
        GraphRequest request = GraphRequest.newMeRequest(token, new
GraphRequest.GraphJSONObjectCallback() {
            String firstName;
            @Override
            public void onCompleted(JSONObject object, GraphResponse response) {
                try {
                    firstName = object.getString("first_name");
                    welcome.setText("Welcome, " + firstName + "!");
                } catch (Exception e) {
                    firstName = e.getMessage();
                    welcome.setText(firstName);
                }
            }
        });

        Bundle parameters = new Bundle();
        parameters.putString("fields", "first_name");
        request.setParameters(parameters);
        request.executeAsync();

        String imageUrl =
"https://graph.facebook.com/"+token.getUserId()+"/picture?return_ssl_resources=1";
        Picasso.get().load(imageUrl).into(profile);
    }

    void checkLoginStatus() {
        if (AccessToken.getCurrentAccessToken() != null) {
            loadUserData(AccessToken.getCurrentAccessToken());
        }
    }
}
```

secondpage

```
package com.example.journeyapp;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.navigation.NavController;
import androidx.navigation.NavDestination;
import androidx.navigation.Navigation;
import androidx.navigation.ui.NavigationUI;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.facebook.AccessToken;
import com.facebook.AccessTokenTracker;
import com.facebook.GraphRequest;
import com.facebook.GraphResponse;
import com.google.android.material.navigation.NavigationView;
import com.makeramen.roundedimageview.RoundedImageView;
import com.squareup.picasso.Picasso;
import org.json.JSONObject;

public class secondpage extends AppCompatActivity {

    RoundedImageView menuProfileImage;
    TextView menuProfileName;

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

        Intent intent = getIntent();
        String directions =
intent.getStringExtra(Constants.navigateToBackpack.key);
        if (directions!=null) {
            NavController navController = Navigation.findNavController(this,
R.id.navHostFragment);
            navController.navigate(R.id.backpackFragment);
        }

        final DrawerLayout drawerLayout = findViewById(R.id.drawerLayout);
        findViewById(R.id.imageMenu).setOnClickListener(new View.OnClickListener()
{
```

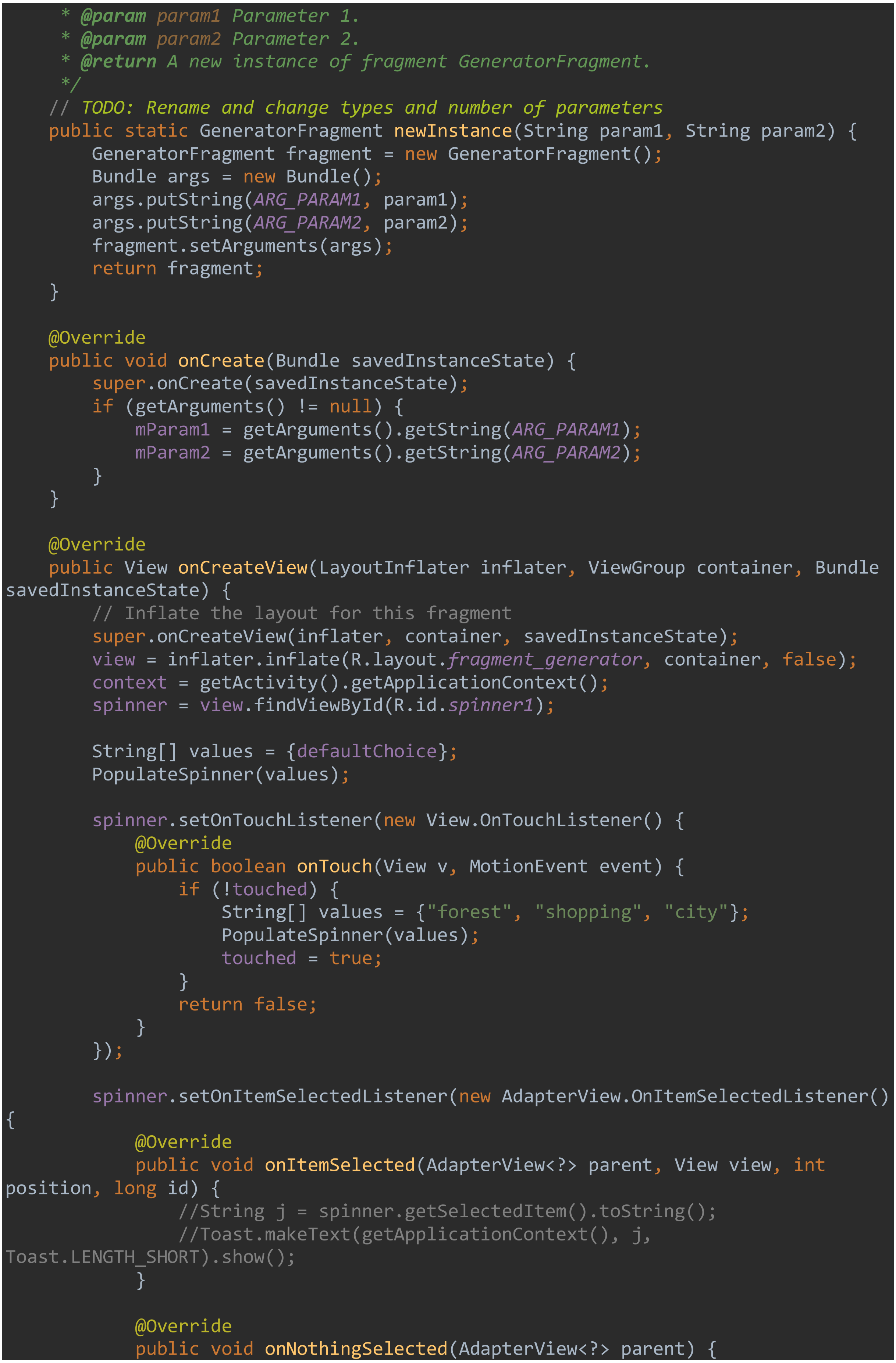
```
            @Override
            public void onClick(View view) {
                drawerLayout.openDrawer(GravityCompat.START);
            }
        });

        NavigationView navigationView = findViewById(R.id.navigationView);
        //navigationView.setItemIconTintList(null);

        NavController navController = Navigation.findNavController(this,
R.id.navHostFragment);
        NavigationUI.setupWithNavController(navigationView, navController);
        final TextView textTitle = findViewById(R.id.textTitle);
        navController.addOnDestinationChangedListener(new
NavController.OnDestinationChangedListener() {
            @Override
            public void onDestinationChanged(@NonNull NavController controller,
@NonNull NavDestination destination, @Nullable Bundle arguments) {
                textTitle.setText(destination.getLabel());
            }
        });

        if (AccessToken.getCurrentAccessToken() != null) {
            //jei neatsijunge nuo fb
            AccessToken token = AccessToken.getCurrentAccessToken();
            loadUserData(token);
        }
    }

    void loadUserData(AccessToken token) {
        NavigationView navView = findViewById(R.id.navigationView);
        View headerLayout = navView.getHeaderView(0);
        menuProfileName = headerLayout.findViewById(R.id.header_fb_name);
        menuProfileImage = headerLayout.findViewById(R.id.imageProfile);

        GraphRequest request = GraphRequest.newMeRequest(token, new
GraphRequest.GraphJSONObjectCallback() {
            String firstName;
            @Override
            public void onCompleted(JSONObject object, GraphResponse response) {
                try {
                    firstName = object.getString("first_name");
                    menuProfileName.setText(firstName);
                } catch (Exception e) {
                    firstName = e.getMessage();
                    menuProfileName.setText(firstName);
                }
            }
        });
        Bundle parameters = new Bundle();
        parameters.putString("fields", "first_name");
        request.setParameters(parameters);
        request.executeAsync();

        String imageUrl =
"https://graph.facebook.com/"+token.getUserId()+"/picture?return_ssl_resources=1";
        Picasso.get().load(imageUrl).into(menuProfileImage);
    }
}
```

ReminderFragment

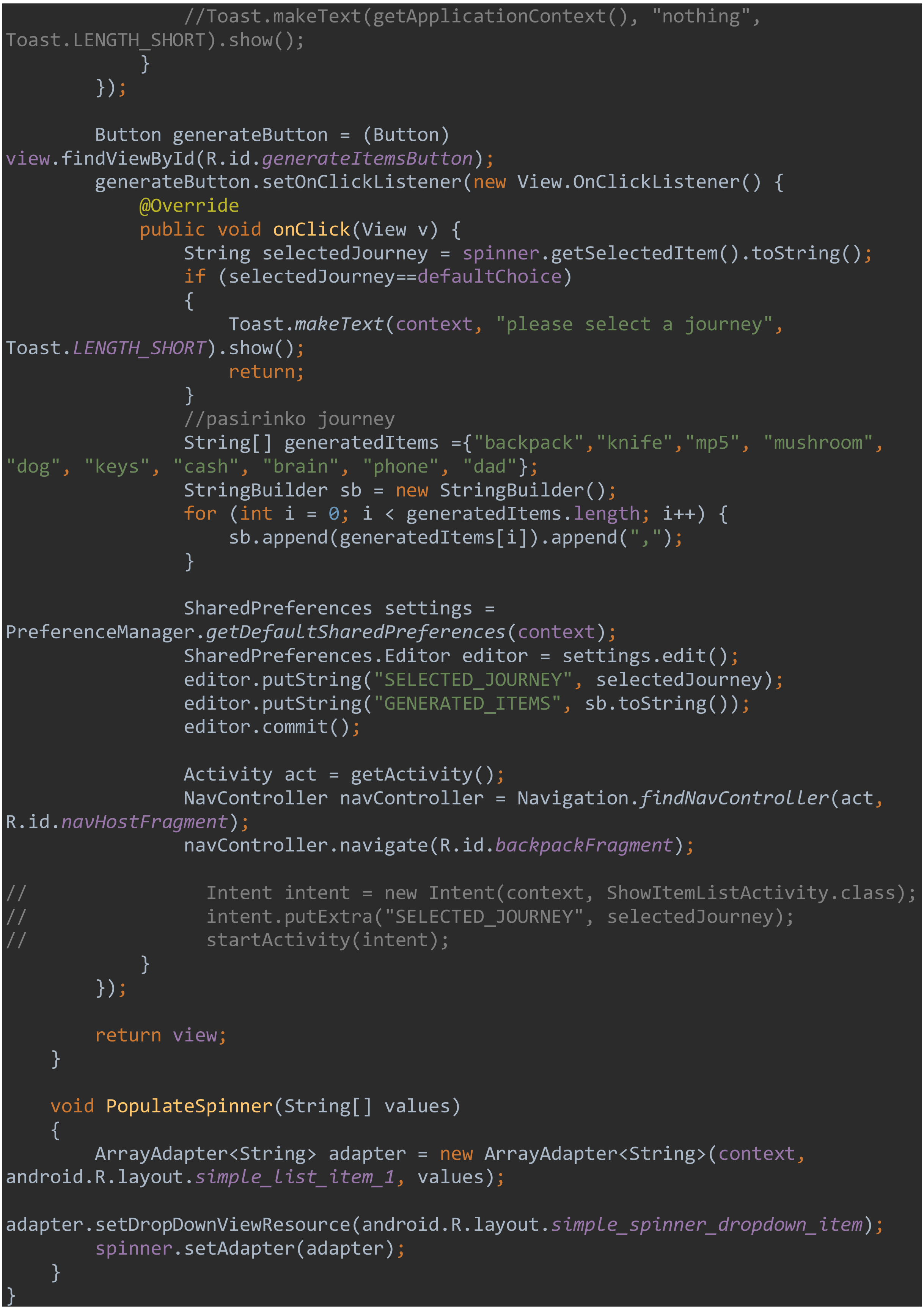
```java
package com.example.journeyapp;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link RemindersFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class RemindersFragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    public RemindersFragment() {
        // Required empty public constructor
    }

    public static RemindersFragment newInstance(String param1, String param2) {
        RemindersFragment fragment = new RemindersFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_reminders, container, false);
    }
}
```

GeneratorFragment

```
package com.example.journeyapp;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;

import android.preference.PreferenceManager;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.material.internal.NavigationMenu;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link GeneratorFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class GeneratorFragment extends Fragment {

    View view;
    Context context;
    Spinner spinner;
    boolean touched=false;
    String defaultChoice="----------";

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    public GeneratorFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
```



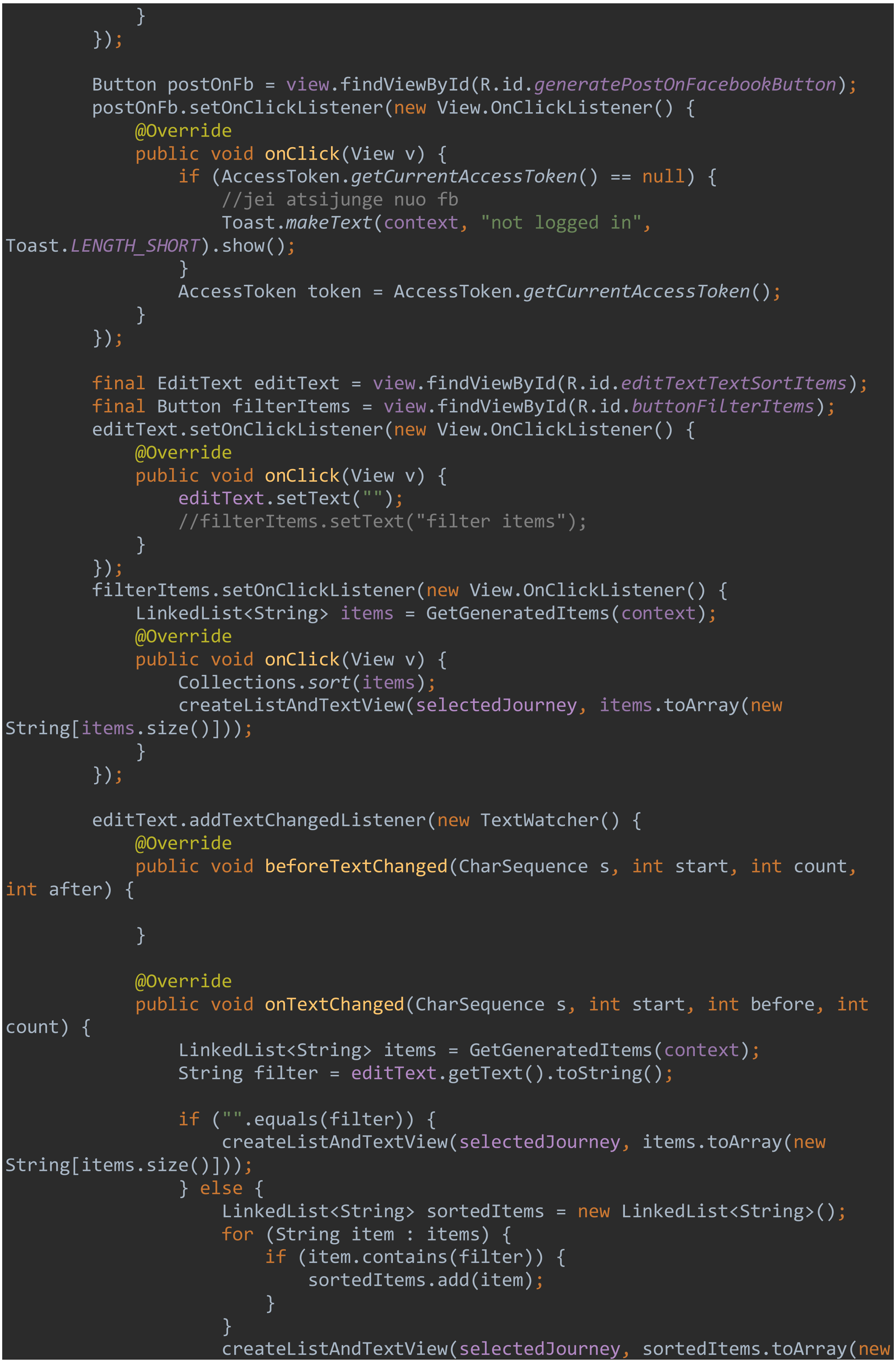
```
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment GeneratorFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static GeneratorFragment newInstance(String param1, String param2) {
        GeneratorFragment fragment = new GeneratorFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        // Inflate the layout for this fragment
        super.onCreateView(inflater, container, savedInstanceState);
        view = inflater.inflate(R.layout.fragment_generator, container, false);
        context = getActivity().getApplicationContext();
        spinner = view.findViewById(R.id.spinner1);

        String[] values = {defaultChoice};
        PopulateSpinner(values);

        spinner.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                if (!touched) {
                    String[] values = {"forest", "shopping", "city"};
                    PopulateSpinner(values);
                    touched = true;
                }
                return false;
            }
        });

        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
{
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {
                //String j = spinner.getSelectedItem().toString();
                //Toast.makeText(getApplicationContext(), j,
Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
```



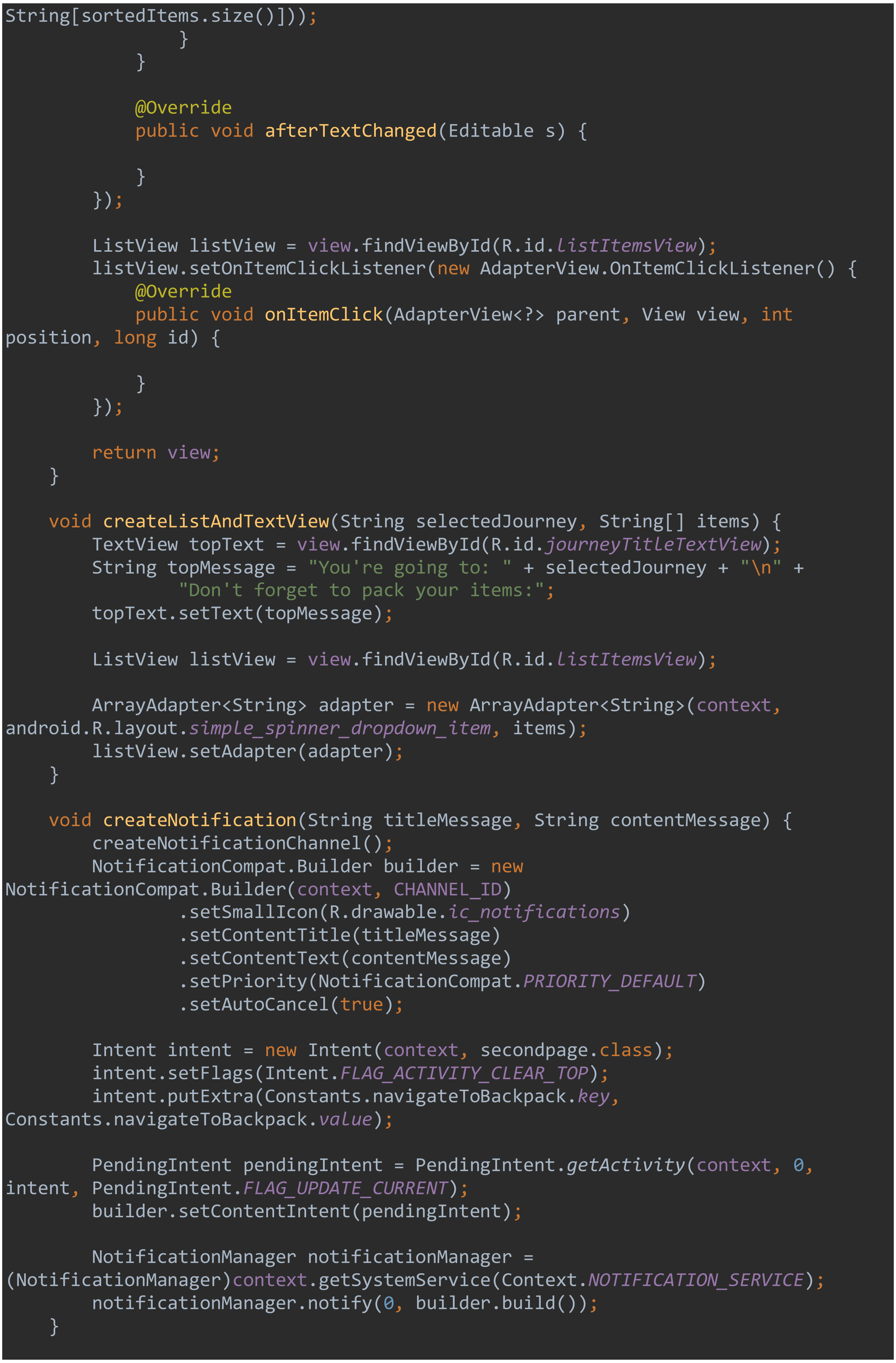
```
                //Toast.makeText(getApplicationContext(), "nothing",
Toast.LENGTH_SHORT).show();
            }
        });

        Button generateButton = (Button)
view.findViewById(R.id.generateItemsButton);
        generateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String selectedJourney = spinner.getSelectedItem().toString();
                if (selectedJourney==defaultChoice)
                {
                    Toast.makeText(context, "please select a journey",
Toast.LENGTH_SHORT).show();
                    return;
                }
                //pasirinko journey
                String[] generatedItems ={"backpack","knife","mp5", "mushroom",
"dog", "keys", "cash", "brain", "phone", "dad"};
                StringBuilder sb = new StringBuilder();
                for (int i = 0; i < generatedItems.length; i++) {
                    sb.append(generatedItems[i]).append(",");
                }

                SharedPreferences settings =
PreferenceManager.getDefaultSharedPreferences(context);
                SharedPreferences.Editor editor = settings.edit();
                editor.putString("SELECTED_JOURNEY", selectedJourney);
                editor.putString("GENERATED_ITEMS", sb.toString());
                editor.commit();

                Activity act = getActivity();
                NavController navController = Navigation.findNavController(act,
R.id.navHostFragment);
                navController.navigate(R.id.backpackFragment);

//                Intent intent = new Intent(context, ShowItemListActivity.class);
//                intent.putExtra("SELECTED_JOURNEY", selectedJourney);
//                startActivity(intent);
            }
        });

        return view;
    }

    void PopulateSpinner(String[] values)
    {
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(context,
android.R.layout.simple_list_item_1, values);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);
    }
}
```

BackpackFragment

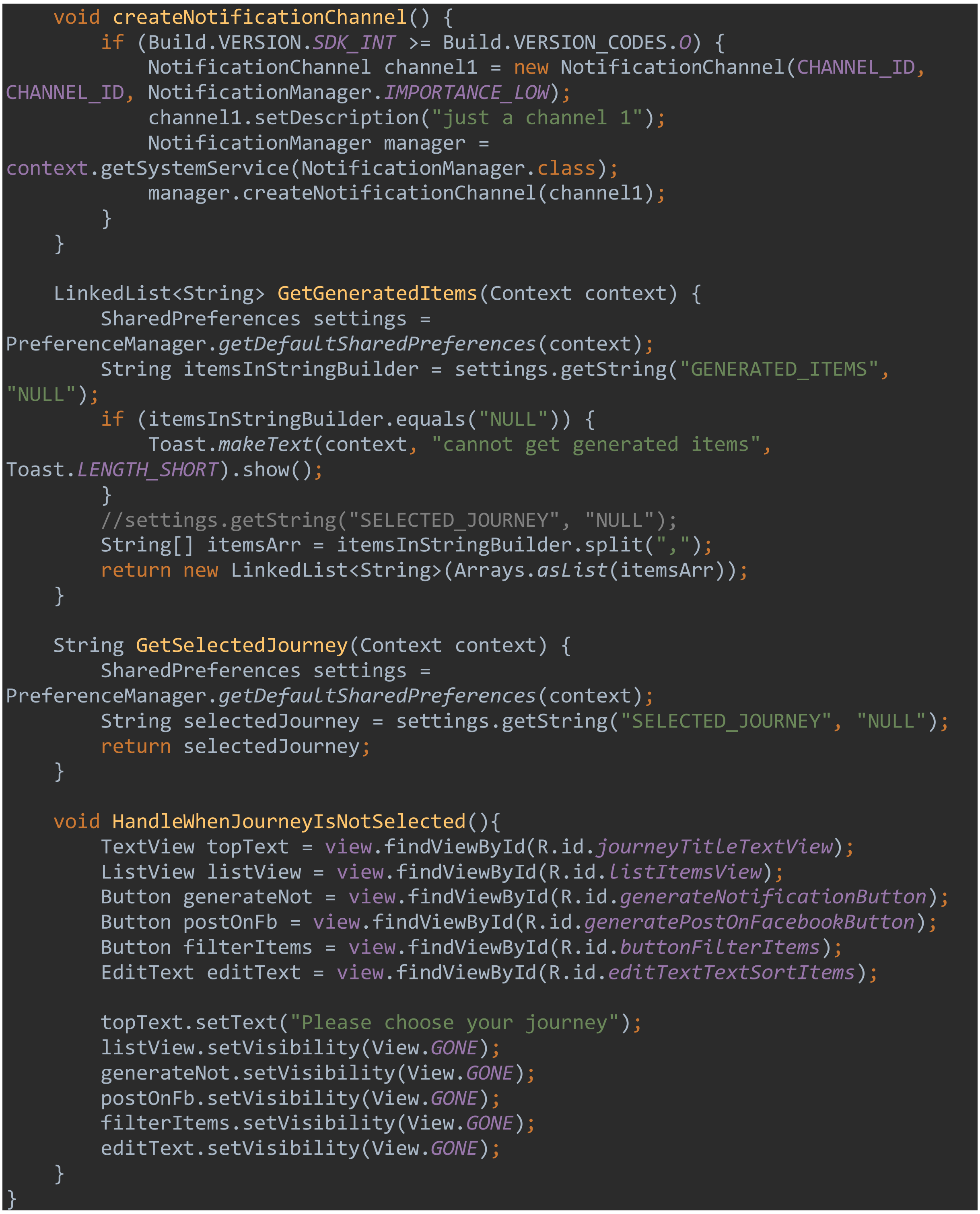
```
package com.example.journeyapp;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Build;
import android.os.Bundle;

import androidx.core.app.NotificationCompat;
import androidx.fragment.app.Fragment;

import android.preference.PreferenceManager;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.facebook.AccessToken;

import java.util.Arrays;
import java.util.Collections;
import java.util.LinkedList;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link BackpackFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class BackpackFragment extends Fragment {
    final String CHANNEL_ID = "channel1";
    View view;
    Context context;
    //String[] generatedItems ={"backpack","knife","mp5", "mushroom", "dog",
"keys", "cash", "brain", "phone", "dad"};
    //LinkedList<String> items = new
LinkedList<String>(Arrays.asList(generatedItems));

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
    public BackpackFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment BackpackFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static BackpackFragment newInstance(String param1, String param2) {
        BackpackFragment fragment = new BackpackFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        super.onCreateView(inflater, container, savedInstanceState);
        view = inflater.inflate(R.layout.fragment_backpack, container, false);
        context = getActivity().getApplicationContext();

        final String selectedJourney = GetSelectedJourney(context);
        if (selectedJourney=="NULL")
        {
            HandleWhenJourneyIsNotSelected();
            return view;
        }

        LinkedList<String> items = GetGeneratedItems(context);
        createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));

        Button generateNot = view.findViewById(R.id.generateNotificationButton);
        generateNot.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(context, "Notification generated",
Toast.LENGTH_SHORT).show();
                String titleMessage = "Prepare for an adventure to " +
selectedJourney + "!";
                String contentMessage = "Don't forget to pack your items!";
                createNotification(titleMessage, contentMessage);
```



```
            }
        });

        Button postOnFb = view.findViewById(R.id.generatePostOnFacebookButton);
        postOnFb.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (AccessToken.getCurrentAccessToken() == null) {
                    //jei atsijunge nuo fb
                    Toast.makeText(context, "not logged in",
Toast.LENGTH_SHORT).show();
                }
                AccessToken token = AccessToken.getCurrentAccessToken();
            }
        });

        final EditText editText = view.findViewById(R.id.editTextTextSortItems);
        final Button filterItems = view.findViewById(R.id.buttonFilterItems);
        editText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                editText.setText("");
                //filterItems.setText("filter items");
            }
        });
        filterItems.setOnClickListener(new View.OnClickListener() {
            LinkedList<String> items = GetGeneratedItems(context);
            @Override
            public void onClick(View v) {
                Collections.sort(items);
                createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
            }
        });

        editText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count,
int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int
count) {
                LinkedList<String> items = GetGeneratedItems(context);
                String filter = editText.getText().toString();

                if ("".equals(filter)) {
                    createListAndTextView(selectedJourney, items.toArray(new
String[items.size()]));
                } else {
                    LinkedList<String> sortedItems = new LinkedList<String>();
                    for (String item : items) {
                        if (item.contains(filter)) {
                            sortedItems.add(item);
                        }
                    }
                    createListAndTextView(selectedJourney, sortedItems.toArray(new
```



```
String[sortedItems.size()]));
                }
            }

            @Override
            public void afterTextChanged(Editable s) {

            }
        });

        ListView listView = view.findViewById(R.id.listItemsView);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

            }
        });

        return view;
    }

    void createListAndTextView(String selectedJourney, String[] items) {
        TextView topText = view.findViewById(R.id.journeyTitleTextView);
        String topMessage = "You're going to: " + selectedJourney + "\n" +
                "Don't forget to pack your items:";
        topText.setText(topMessage);

        ListView listView = view.findViewById(R.id.listItemsView);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(context,
android.R.layout.simple_spinner_dropdown_item, items);
        listView.setAdapter(adapter);
    }

    void createNotification(String titleMessage, String contentMessage) {
        createNotificationChannel();
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_notifications)
                .setContentTitle(titleMessage)
                .setContentText(contentMessage)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setAutoCancel(true);

        Intent intent = new Intent(context, secondpage.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.putExtra(Constants.navigateToBackpack.key,
Constants.navigateToBackpack.value);

        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
intent, PendingIntent.FLAG_UPDATE_CURRENT);
        builder.setContentIntent(pendingIntent);

        NotificationManager notificationManager =
(NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify(0, builder.build());
    }
```



```
    void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel1 = new NotificationChannel(CHANNEL_ID,
CHANNEL_ID, NotificationManager.IMPORTANCE_LOW);
            channel1.setDescription("just a channel 1");
            NotificationManager manager =
context.getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel1);
        }
    }

    LinkedList<String> GetGeneratedItems(Context context) {
        SharedPreferences settings =
PreferenceManager.getDefaultSharedPreferences(context);
        String itemsInStringBuilder = settings.getString("GENERATED_ITEMS",
"NULL");
        if (itemsInStringBuilder.equals("NULL")) {
            Toast.makeText(context, "cannot get generated items",
Toast.LENGTH_SHORT).show();
        }
        //settings.getString("SELECTED_JOURNEY", "NULL");
        String[] itemsArr = itemsInStringBuilder.split(",");
        return new LinkedList<String>(Arrays.asList(itemsArr));
    }

    String GetSelectedJourney(Context context) {
        SharedPreferences settings =
PreferenceManager.getDefaultSharedPreferences(context);
        String selectedJourney = settings.getString("SELECTED_JOURNEY", "NULL");
        return selectedJourney;
    }

    void HandleWhenJourneyIsNotSelected(){
        TextView topText = view.findViewById(R.id.journeyTitleTextView);
        ListView listView = view.findViewById(R.id.listItemsView);
        Button generateNot = view.findViewById(R.id.generateNotificationButton);
        Button postOnFb = view.findViewById(R.id.generatePostOnFacebookButton);
        Button filterItems = view.findViewById(R.id.buttonFilterItems);
        EditText editText = view.findViewById(R.id.editTextTextSortItems);

        topText.setText("Please choose your journey");
        listView.setVisibility(View.GONE);
        generateNot.setVisibility(View.GONE);
        postOnFb.setVisibility(View.GONE);
        filterItems.setVisibility(View.GONE);
        editText.setVisibility(View.GONE);
    }
}
```