NAME: SAKSHI VINAYAK KITTURKAR
CSU ID: 2860273
HOMEWORK 1

In the UsingIntent app, add a UI control on the screen of the second activity so that you can go back to the first activity (i.e., the main activity). In addition, on the main activity, display an iteration count on the number of times the main activity is displayed.

= UI control on the first screen takes you to the second screen and second screen takes you back to the first one and tell you how many iterations were made and a proper count.

In layout: Activity_One.java the code sets up an Android activity with UI components, handles a button click to start another activity, uses a ViewModel to manage and observe data changes persistently, and increments data when the activity resumes. This design pattern helps maintain a clean separation of concerns in Android app development.

Setting the content view to layout and initializing ‘TextView’ and ‘FloatingActionButton’

```
TextView textView = findViewById(R.id.textView);
FloatingActionButton fab = findViewById(R.id.floatingActionButton);
```

Button Click Handling the click event of the floating action button to navigate.

```
fab.setOnClickListener(view -> {
 Intent intent = new Intent(this, Activity_Two.class);
 startActivity(intent);
 });
```

ViewModel Initialization and LiveData Observation

```
 fab.setOnClickListener(view -> {
Intent intent = new Intent(this, Activity_Two.class);
startActivity(intent);
});
```

Incrementing data when the activity resumes, ensuring data consistency.

```
fab.setOnClickListener(view -> {
Intent intent = new Intent(this, Activity_Two.class);
startActivity(intent);
});
```

In layout: Acitvity_two.java this code sets up a simple Android activity with a button, and when that button is clicked, it finishes the current activity, facilitating navigation back to the previous activity in the application.

For UI Setup setting the content view to the layout initializing the floating action button.

```
setContentView(R.layout.activity_two);
FloatingActionButton fab = findViewById(R.id.floatingActionButton2);
```

Button click handling the click event of the floating action button to finish the current activity and go back to the first activity.

```
fab.setOnClickListener(view -> {
    finish(); // close current activity to go back to the first activity
});
```

In layout: backup_rules.xml it serves as a configuration file for Android Auto Backup features which is used to automatically back up app data and settings. The comments within the file provide guidance on how to customize the backup rules.

```
<full-backup-content>
    <!--
        <include domain="sharedpref" path="."/>
        <exclude domain="sharedpref" path="device.xml"/>
    -->
</full-backup-content>
```

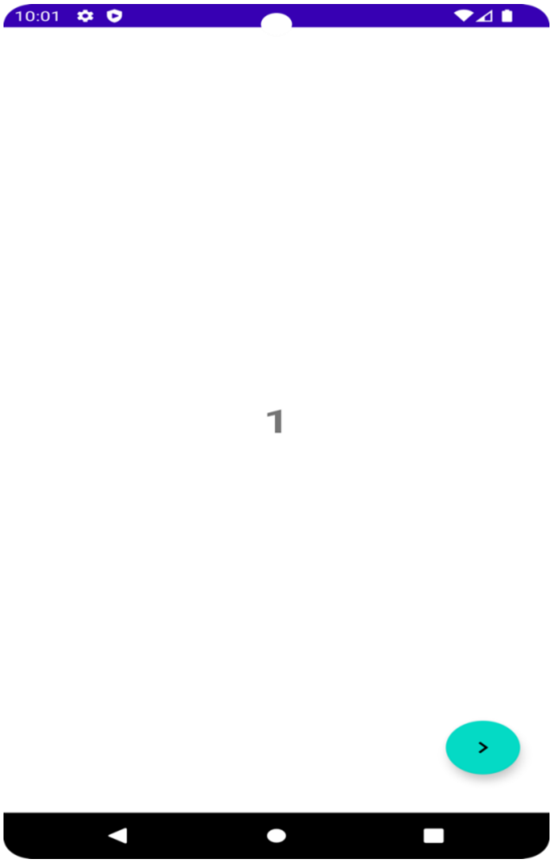
In layout: data_extraction_rules.xml this file serves as a template for specifying data extraction rules related to Android’s backup and restore mechanisms. It provides instruction and placeholders for developers to uncomment and customize the rules based on their app’s requirements.

```
<data-extraction-rules>
```

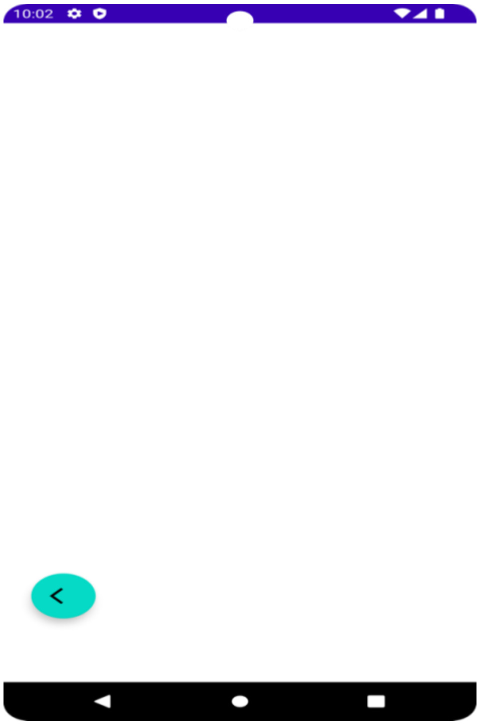
```
    <cloud-backup>
        <!-- TODO: Use <include> and <exclude> to control what is backed up.
            <include .../>
            <exclude .../>
        -->
    </cloud-backup>
    <!--
    <device-transfer>
        <include .../>
        <exclude .../>
    </device-transfer>
    -->
</data-extraction-rules>
```

I have used two arrow buttons to show first screen and second screen and just the number count when we come back to first screen I have not put any text message like count but just implemented it.

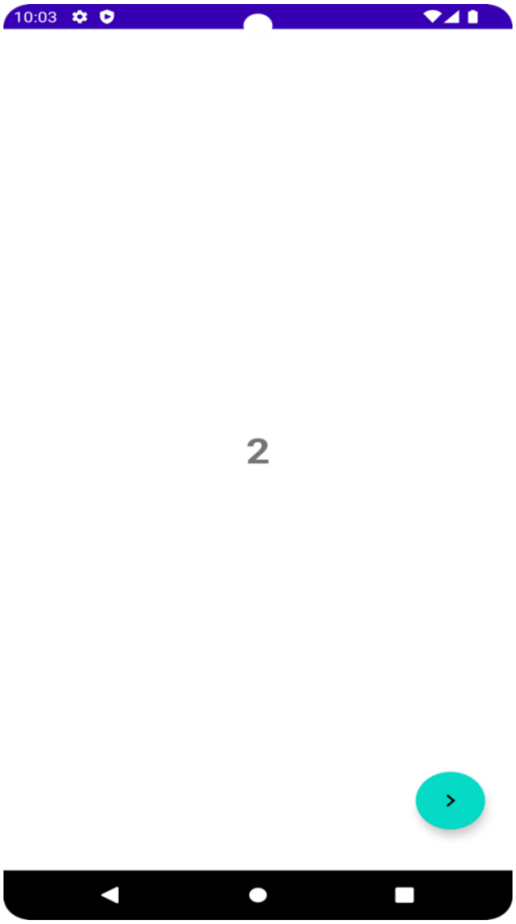
First screen



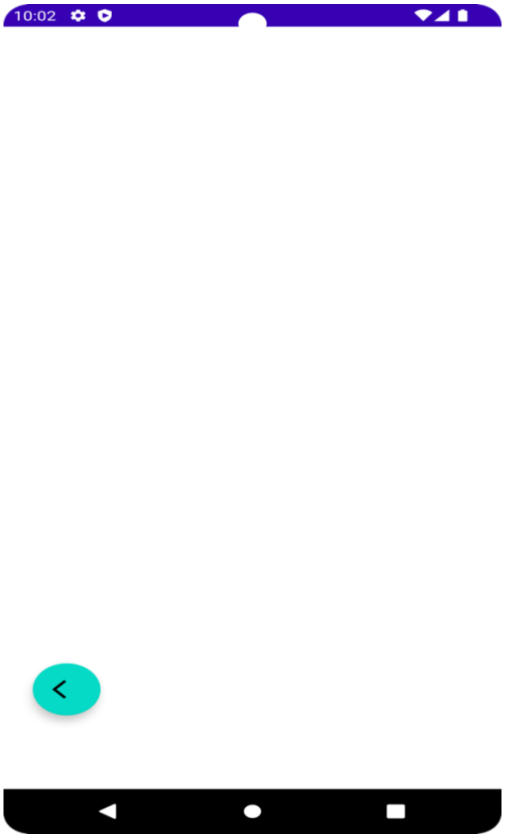
Second screen



Back to first scrren with a count



Again back to second screen



Back to screen with an updated count