Mobile Development

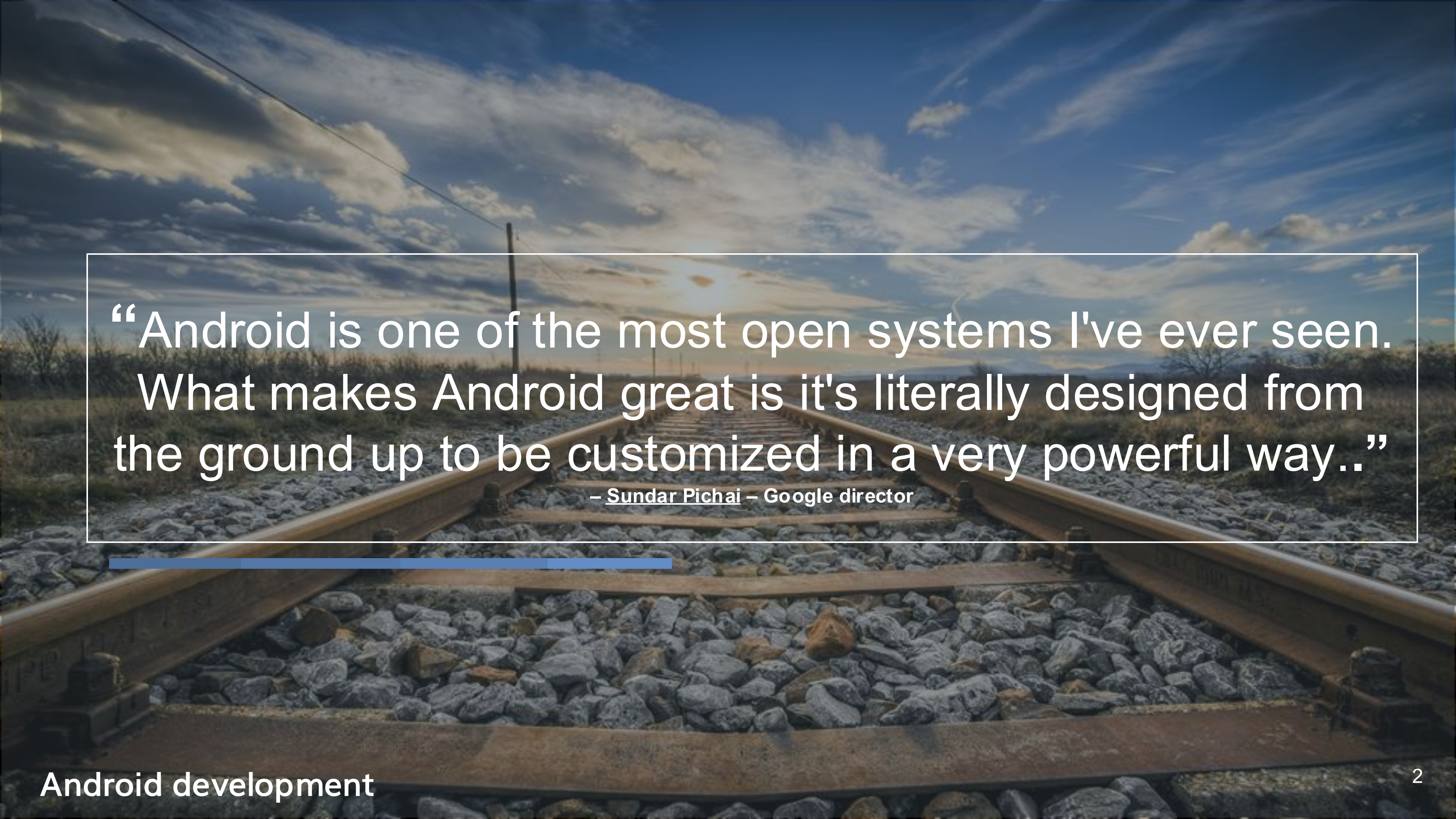
(Lab 4)

Instructor: Tran Vinh Khiem

March 1st, 2022

S3T

Smart Software System Team



> “Android is one of the most open systems I've ever seen. What makes Android great is it's literally designed from the ground up to be customized in a very powerful way..”

– Sundar Pichai – Google director

Learning Objectives

- Provide you with realistic, hands-on experience developing Android applications.
- Create a portfolio of apps that you can show your friends, discuss in interviews, and borrow for other applications.

Storage

Shared preferences

most commonly used for storing app preferences, as key-value pairs

Database

for storing structured data in a private database

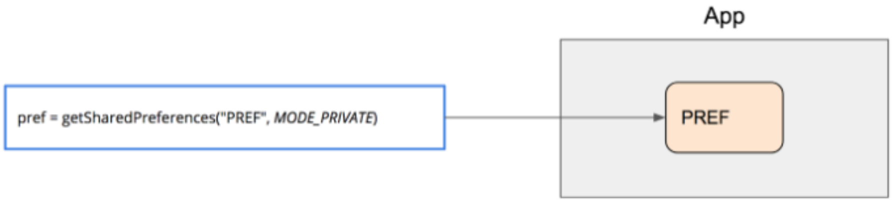
File storage

for storing all kinds of media/documents to the file system

Shared Prereferences

```
val pref = getSharedPreferences("PREF", MODE_PRIVATE)
```

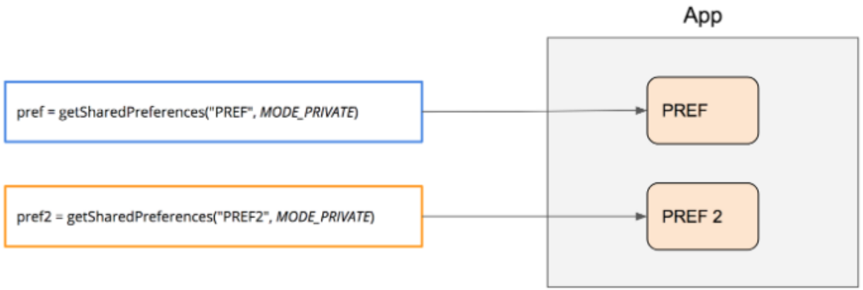
```
val pref2 = getSharedPreferences("PREF2", MODE_PRIVATE)
```



App

pref = getSharedPreferences("PREF", *MODE_PRIVATE*)

PREF



App

pref = getSharedPreferences("PREF", *MODE_PRIVATE*)

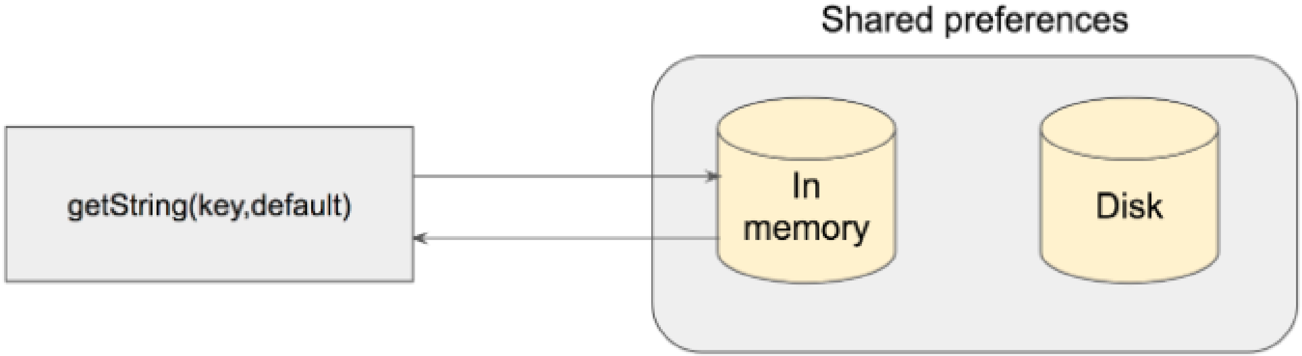
pref2 = getSharedPreferences("PREF2", *MODE_PRIVATE*)

PREF

PREF 2

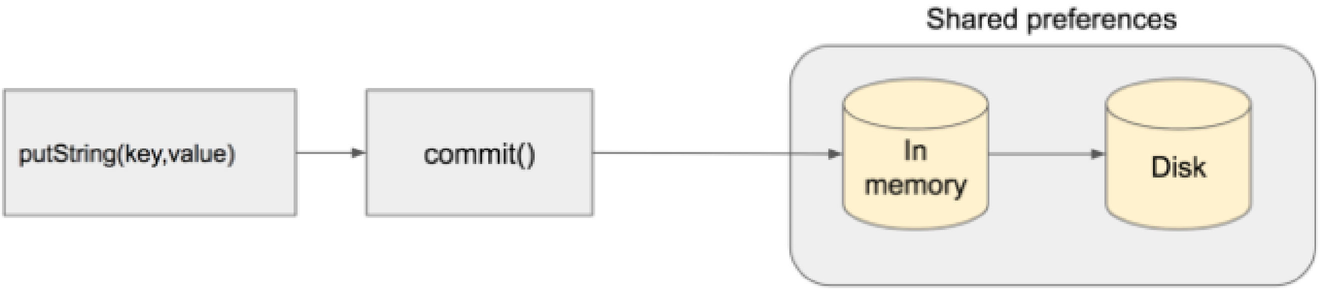
Shared Preferences – Get data

```
val value = pref.getString(key, null)
```



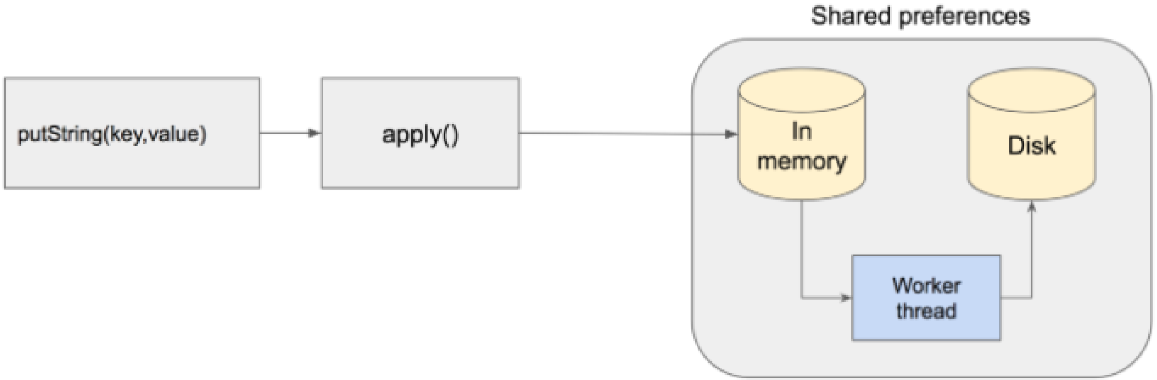
Shared Preferences – Save data via commit

```
val result = pref.edit().putString(key, "value").commit()
```

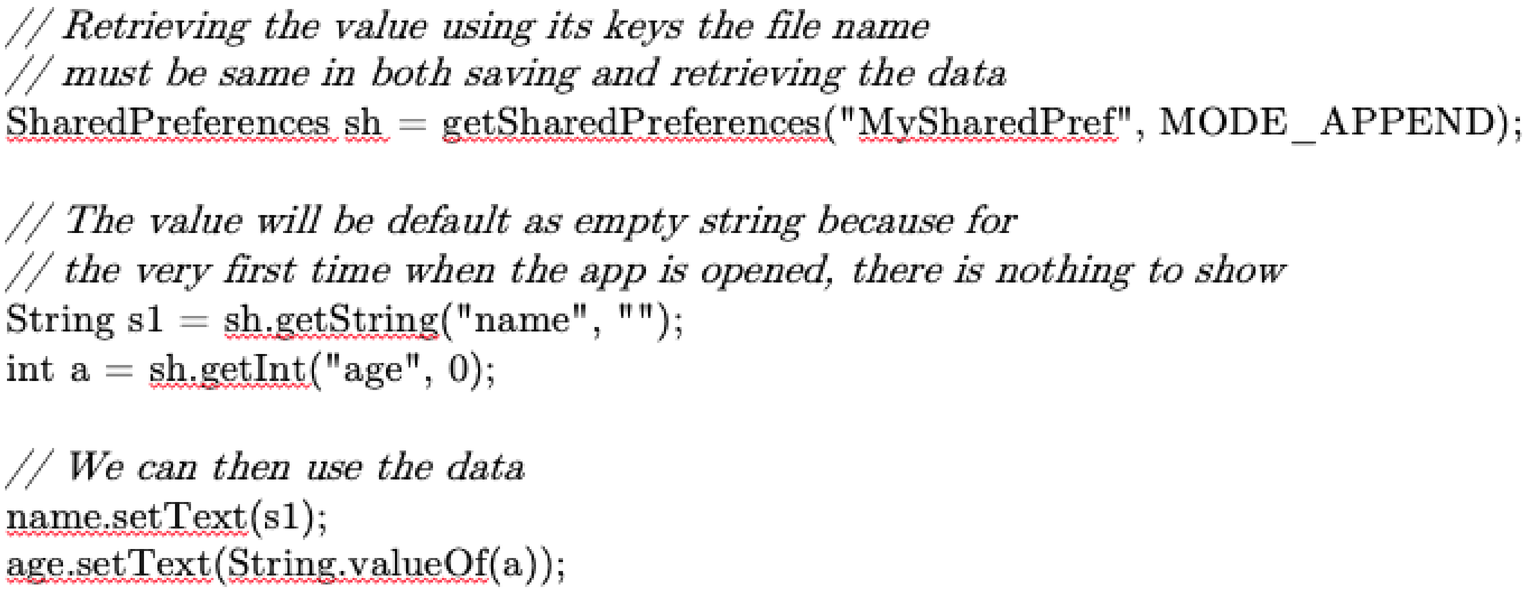


Shared Preferences – Save data via apply

```
pref.edit().putString(key, "value").apply()
```

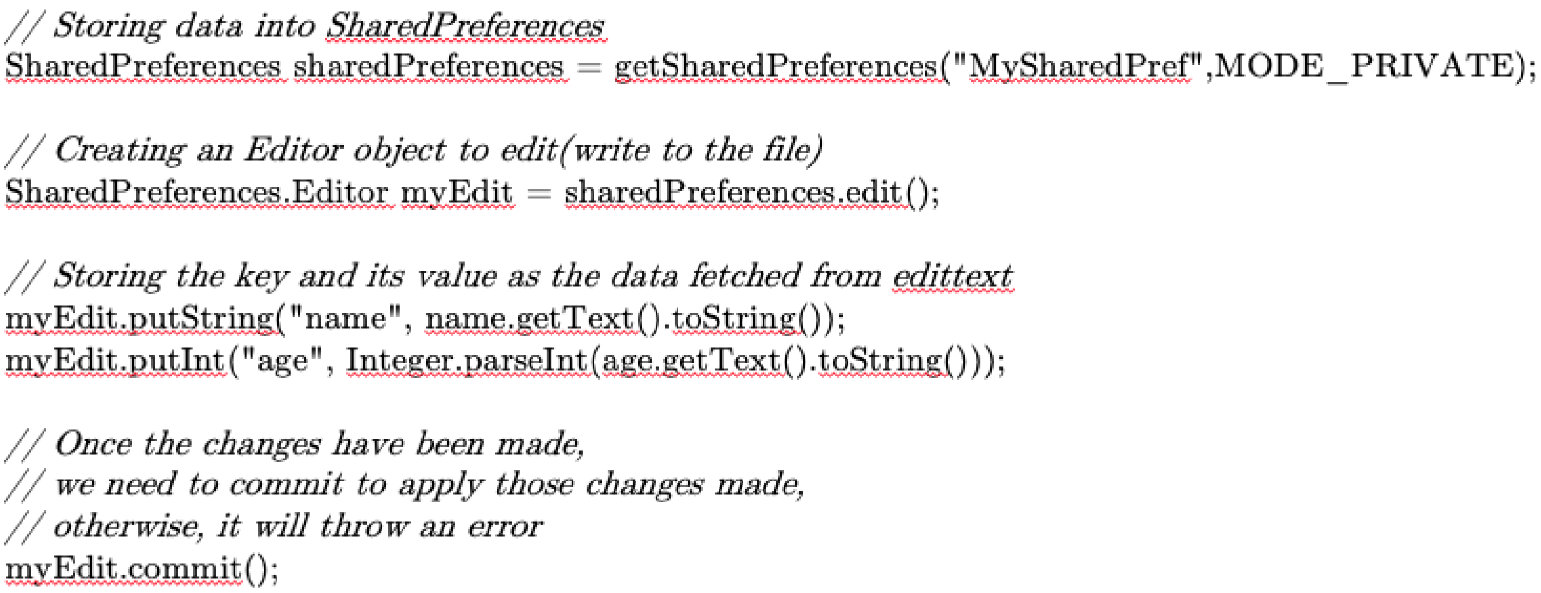


Shared Preferences – Read data



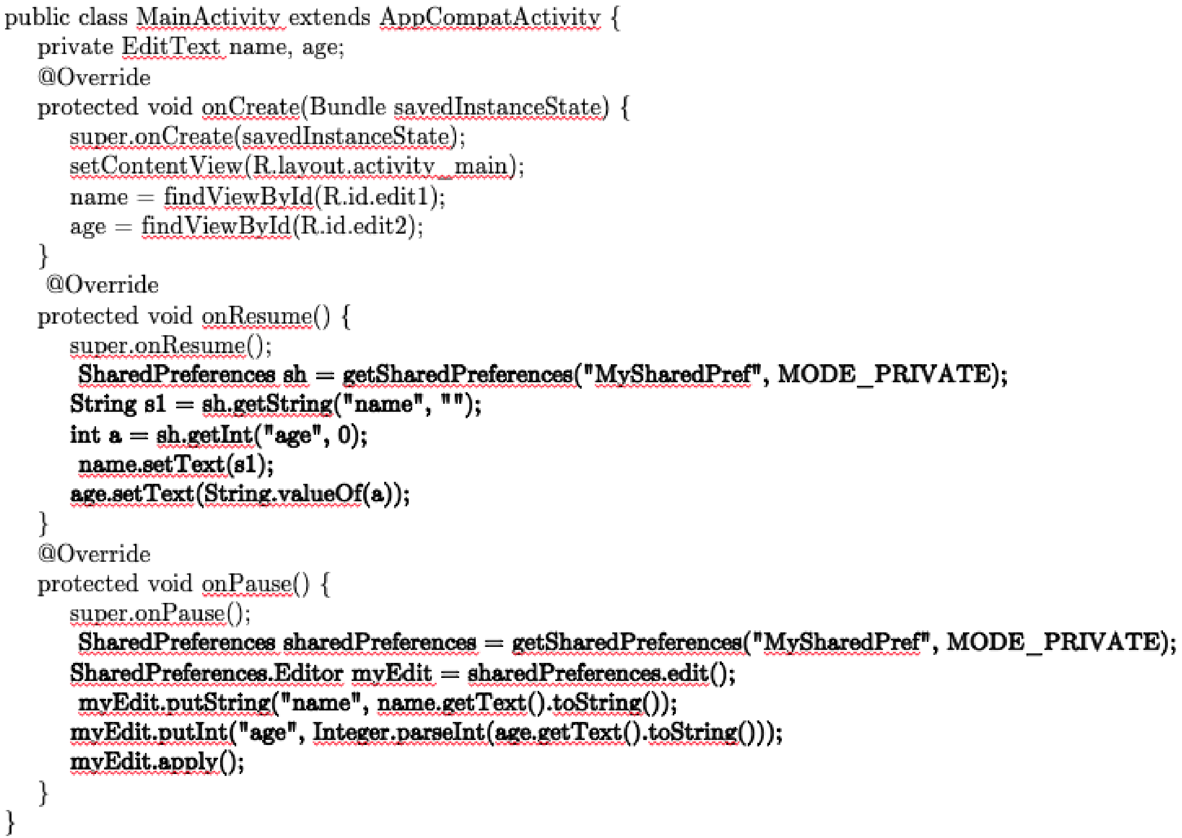
```
// Retrieving the value using its keys the file name
// must be same in both saving and retrieving the data
SharedPreferences sh = getSharedPreferences("MySharedPref", MODE_APPEND);

// The value will be default as empty string because for
// the very first time when the app is opened, there is nothing to show
String s1 = sh.getString("name", "");
int a = sh.getInt("age", 0);

// We can then use the data
name.setText(s1);
age.setText(String.valueOf(a));
```

Shared Preferences – Write data



```
// Storing data into SharedPreferences
SharedPreferences sharedPreferences = getSharedPreferences("MySharedPref",MODE_PRIVATE);

// Creating an Editor object to edit(write to the file)
SharedPreferences.Editor myEdit = sharedPreferences.edit();

// Storing the key and its value as the data fetched from edittext
myEdit.putString("name", name.getText().toString());
myEdit.putInt("age", Integer.parseInt(age.getText().toString()));

// Once the changes have been made,
// we need to commit to apply those changes made,
// otherwise, it will throw an error
myEdit.commit();
```

Shared Preferences - Example



```
public class MainActivity extends AppCompatActivity {
    private EditText name, age;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name = findViewById(R.id.edit1);
        age = findViewById(R.id.edit2);
    }
     @Override
    protected void onResume() {
        super.onResume();
         SharedPreferences sh = getSharedPreferences("MySharedPref", MODE_PRIVATE);
        String s1 = sh.getString("name", "");
        int a = sh.getInt("age", 0);
         name.setText(s1);
        age.setText(String.valueOf(a));
    }
    @Override
    protected void onPause() {
        super.onPause();
         SharedPreferences sharedPreferences = getSharedPreferences("MySharedPref", MODE_PRIVATE);
        SharedPreferences.Editor myEdit = sharedPreferences.edit();
         myEdit.putString("name", name.getText().toString());
        myEdit.putInt("age", Integer.parseInt(age.getText().toString()));
        myEdit.apply();
    }
}
```

Internal and external storage

Internal

Phone memory

External

SD or Micro SD Card

Internal storage – Write and read data

```
FileOutputStream fOut = openFileOutput("file name here",MODE_WORLD_READABLE);

String str = "data";
fOut.write(str.getBytes()); fOut.close();
```

```
FileInputStream fin = openFileInput(file);
```

```
int c; String temp="";
while( (c = fin.read()) != -1){
         temp = temp + Character.toString((char)c);
}
fin.close();
```

Internal storage

```
public void onClick(View v) {
       String myData = "";
       switch (v.getId()) {
           case R.id.btnSave:
               try {
                     FileOutputStream fos = new FileOutputStream(myInternalFile);
                     fos.write(myInputText.getText().toString().getBytes());
                   fos.close();
               } catch (IOException e) {
                   e.printStackTrace();
               }
               myInputText.setText("");
               responseText
                       .setText(”Write data successfully");
               break;
           case R.id.btnDisplay:
               try {
                     FileInputStream fis = new FileInputStream(myInternalFile);
                   DataInputStream in = new DataInputStream(fis);
                   BufferedReader br = new BufferedReader(
                           new InputStreamReader(in));
                   String strLine;
                     while ((strLine = br.readLine()) != null) {
                       myData = myData + strLine;
                   }
                   in.close();
               } catch (IOException e) {
                   e.printStackTrace();
               }
               myInputText.setText(myData);
               responseText
                       .setText(”Get data sucessfully");
               break;
       }
   }
```

External storage - Permission

```
<manifest ... >
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application ... >
        <activity android:name=".MainActivity" ... >
          ...
        </activity>
    </application>
</manifest>
```

External storage

```
private int EXTERNAL_STORAGE_PERMISSION_CODE = 23;
    EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
         editText = (EditText) findViewById(R.id.editText_data);
            File folder =           Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
        File file = new File(folder, ”UIT.txt");
        writeTextData(file, editTextData);
    }
    private void writeTextData(File file, String data) {
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream(file);
            fileOutputStream.write(data.getBytes());
            Toast.makeText(this, "Done" + file.getAbsolutePath(), Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileOutputStream != null) {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
```

Database

- **Oracle**
- Microsoft
  - **SQL Server** (powerful)
  - **Access** (simple)

- **PostgreSQL**
  - powerful/complex free open-source database system
- **SQLite**
  - transportable, lightweight free open-source database system
- **MySQL**
  - simple free open-source database system
  - many servers run "LAMP" (Linux, Apache, MySQL, and PHP)
  - Wikipedia is run on PHP and MySQL

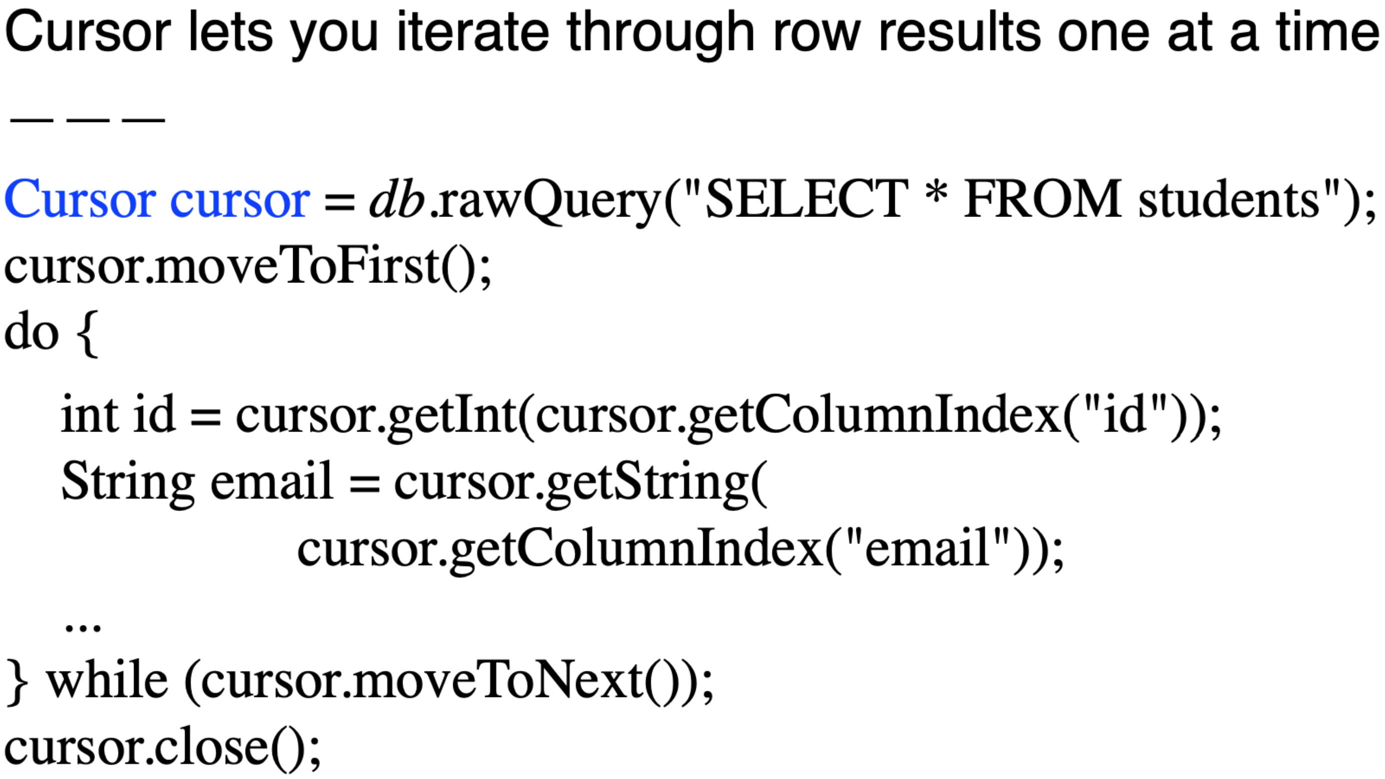
Android SQLite Database

```
SQLiteDatabase db = openOrCreateDatabase(
        "name", MODE_PRIVATE, null);
db.execSQL("SQL query");
```

```
ContentValues cvalues = new ContentValues();
cvalues.put("columnName1", value1);
cvalues.put("columnName2", value2);
...
db.insert("tableName", null, cvalues);
```

```
db.execSQL("INSERT INTO tableName ("
    + columnName1 + ", " + columnName2
    + ") VALUES (" + value1 + ", " + value2 + ")");
```

Android SQLite Database – Cursor



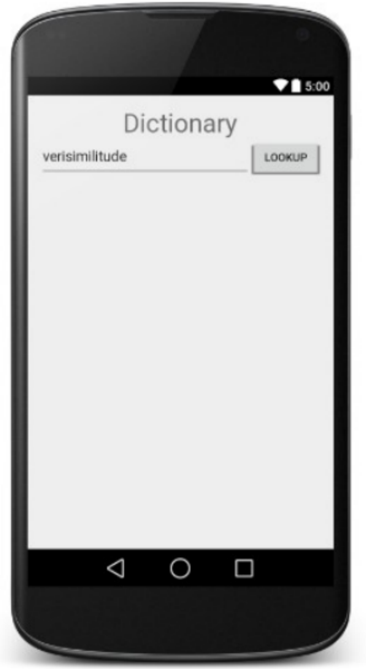
Cursor lets you iterate through row results one at a time

— — —

```
Cursor cursor = db.rawQuery("SELECT * FROM students");
cursor.moveToFirst();
do {
    int id = cursor.getInt(cursor.getColumnIndex("id"));
    String email = cursor.getString(
                    cursor.getColumnIndex("email"));
    ...
} while (cursor.moveToNext());
cursor.close();
```

Dictionary app exercise

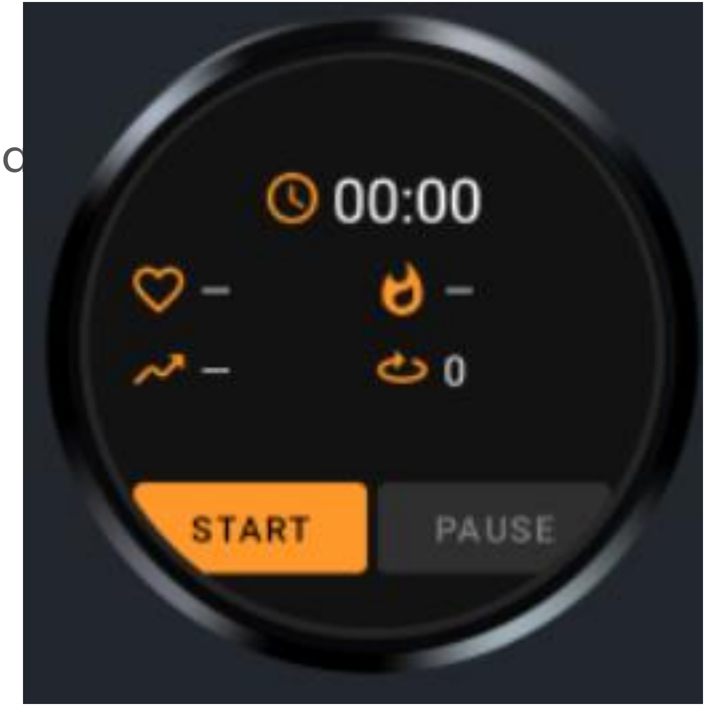
- Write an app that lets the user look up words in a dictionary.
  - The dictionary should be created as a SQLite **database**.
  - When the user types in a word, if that exact word exists in the dictionary, show its definition.
  - If the exact word does not exist in the dictionary, list all words of which the user's text is a substring.



Or use your assignment's app if it's already used SQLite database.

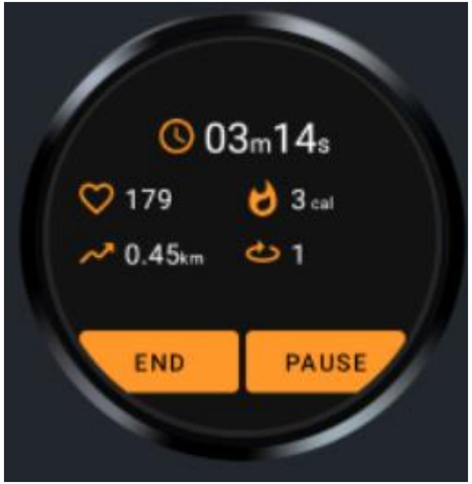
Exercise

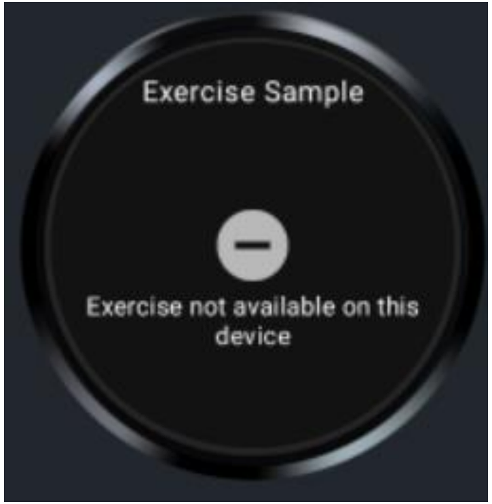
- Build your own Health measure applica
- Example of this app is on the picture.



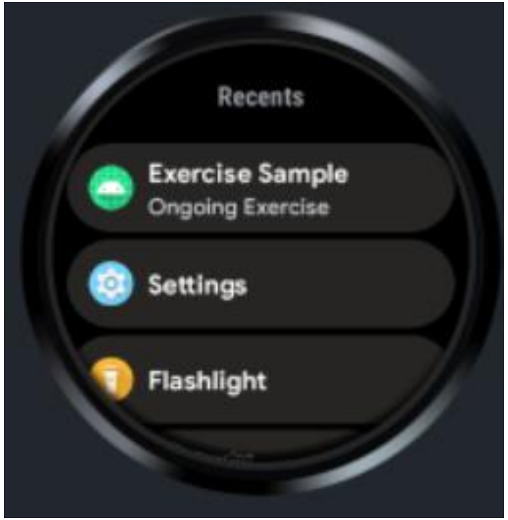
Exercise

Hint: https://developer.android.com/training/wearables/health-services



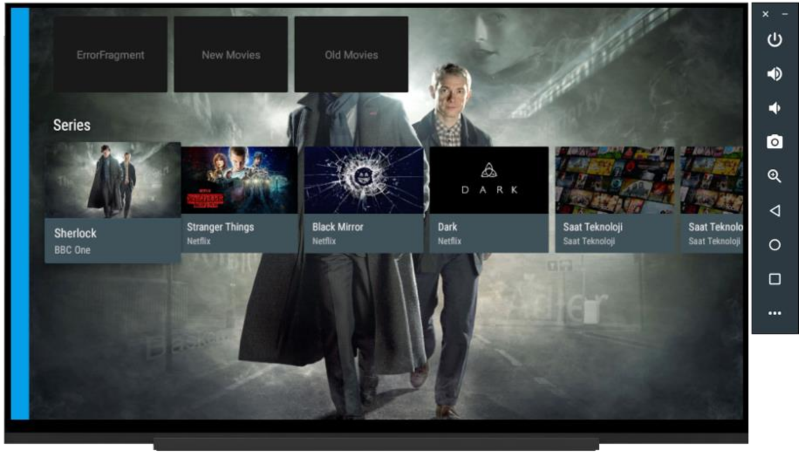

Exercise Sample

Exercise not available on this device



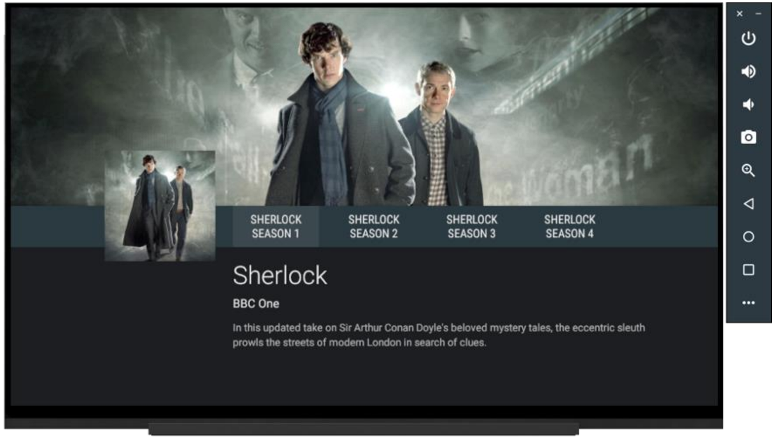
Homework

- Build your own TV Android application with your favorite movie.
- Example of this app is on the picture.
- You can use this API to load film: https://developers.themoviedb.org/3



Homework

https://developer.android.com/training/tv/start/start

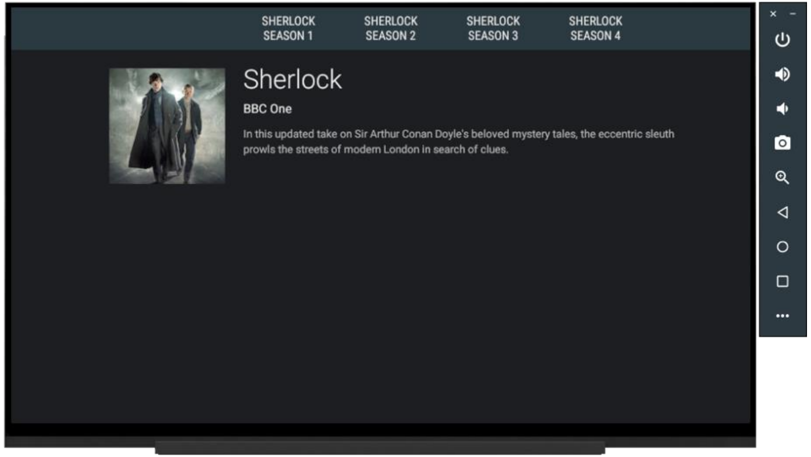


SHERLOCK SEASON 1 SHERLOCK SEASON 2 SHERLOCK SEASON 3 SHERLOCK SEASON 4

Sherlock

BBC One

In this updated take on Sir Arthur Conan Doyle's beloved mystery tales, the eccentric sleuth prowls the streets of modern London in search of clues.



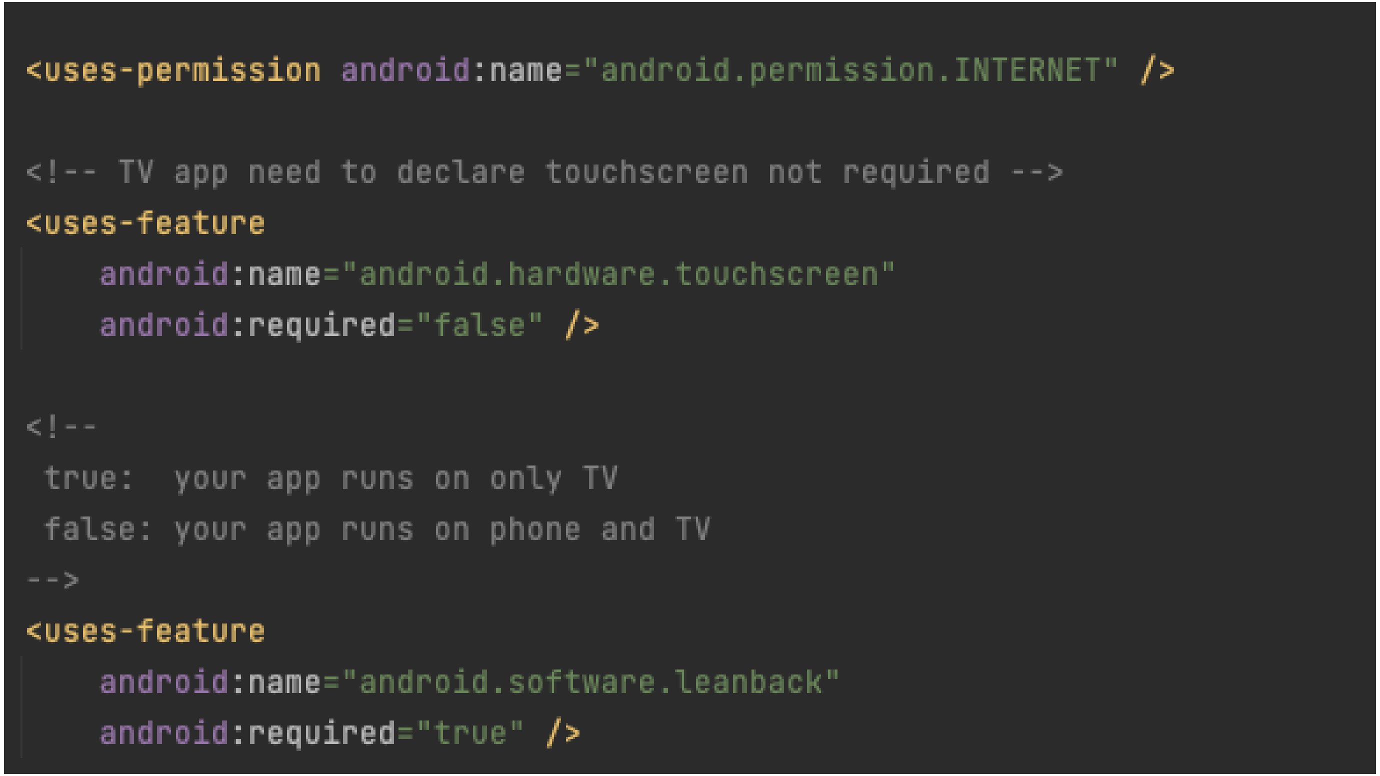
SHERLOCK SEASON 1 SHERLOCK SEASON 2 SHERLOCK SEASON 3 SHERLOCK SEASON 4

Sherlock

BBC One

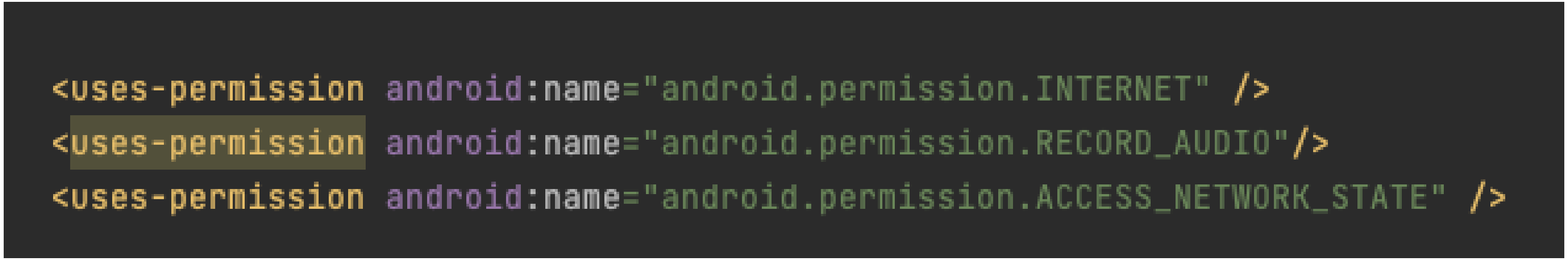
In this updated take on Sir Arthur Conan Doyle's beloved mystery tales, the eccentric sleuth prowls the streets of modern London in search of clues.

HW – Hint – Android Mainfest

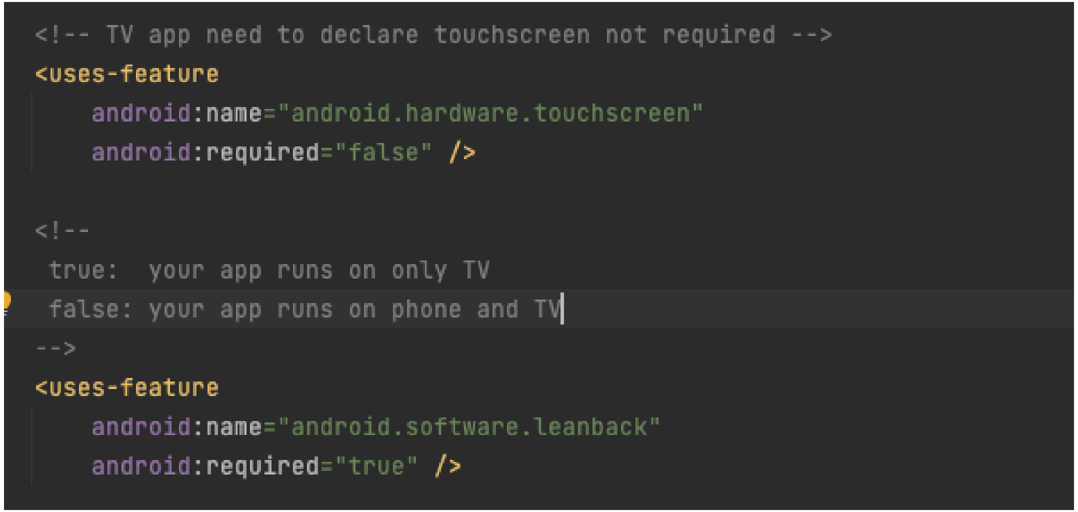


```
<uses-permission android:name="android.permission.INTERNET" />

<!-- TV app need to declare touchscreen not required -->
<uses-feature
    android:name="android.hardware.touchscreen"
    android:required="false" />

<!--
 true:  your app runs on only TV
 false: your app runs on phone and TV
-->
<uses-feature
    android:name="android.software.leanback"
    android:required="true" />
```

HW – Hint – Android Manifest

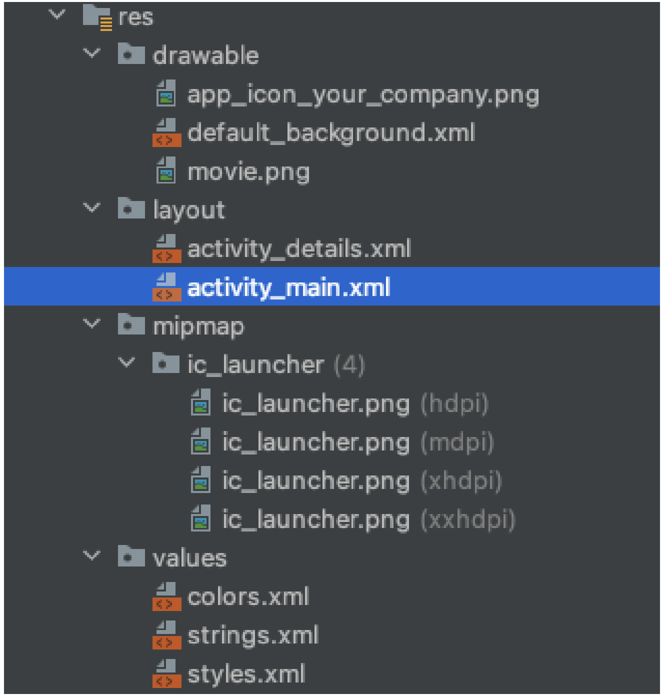


```xml
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

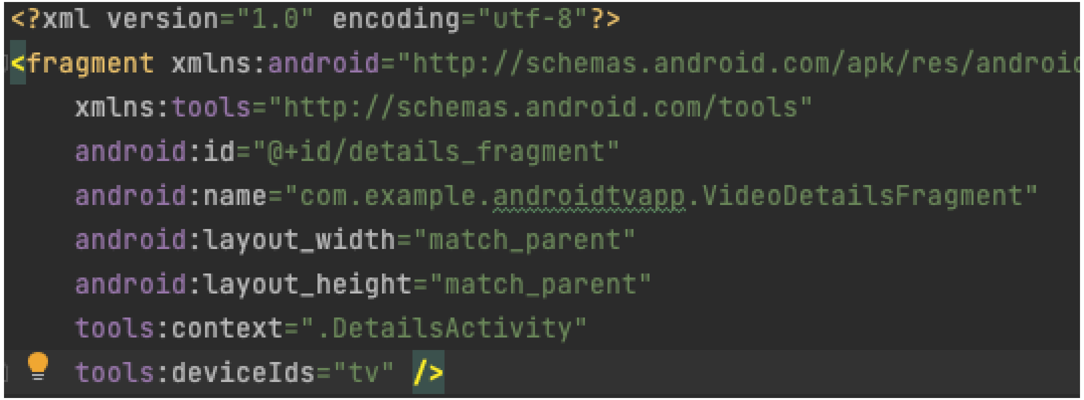


```xml
<!-- TV app need to declare touchscreen not required -->
<uses-feature
    android:name="android.hardware.touchscreen"
    android:required="false" />

<!--
 true:  your app runs on only TV
 false: your app runs on phone and TV
-->
<uses-feature
    android:name="android.software.leanback"
    android:required="true" />
```

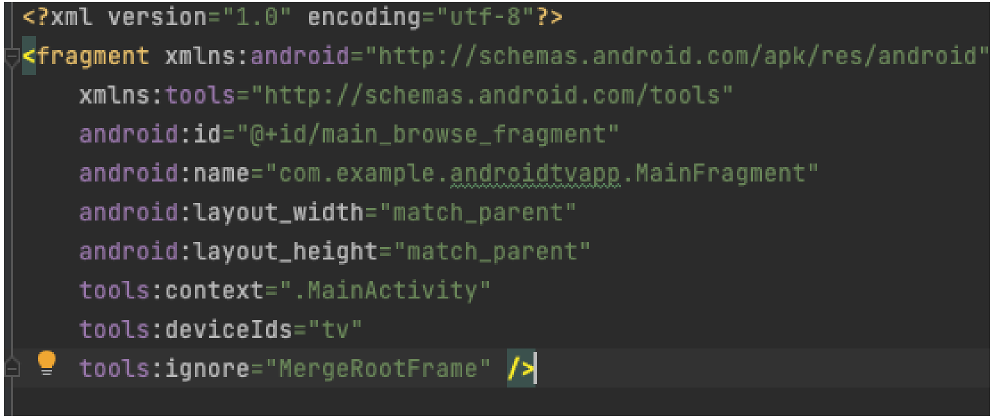
HW – Hint – Layout



```
res
  drawable
    app_icon_your_company.png
    default_background.xml
    movie.png
  layout
    activity_details.xml
    activity_main.xml
  mipmap
    ic_launcher (4)
      ic_launcher.png (hdpi)
      ic_launcher.png (mdpi)
      ic_launcher.png (xhdpi)
      ic_launcher.png (xxhdpi)
  values
    colors.xml
    strings.xml
    styles.xml
```

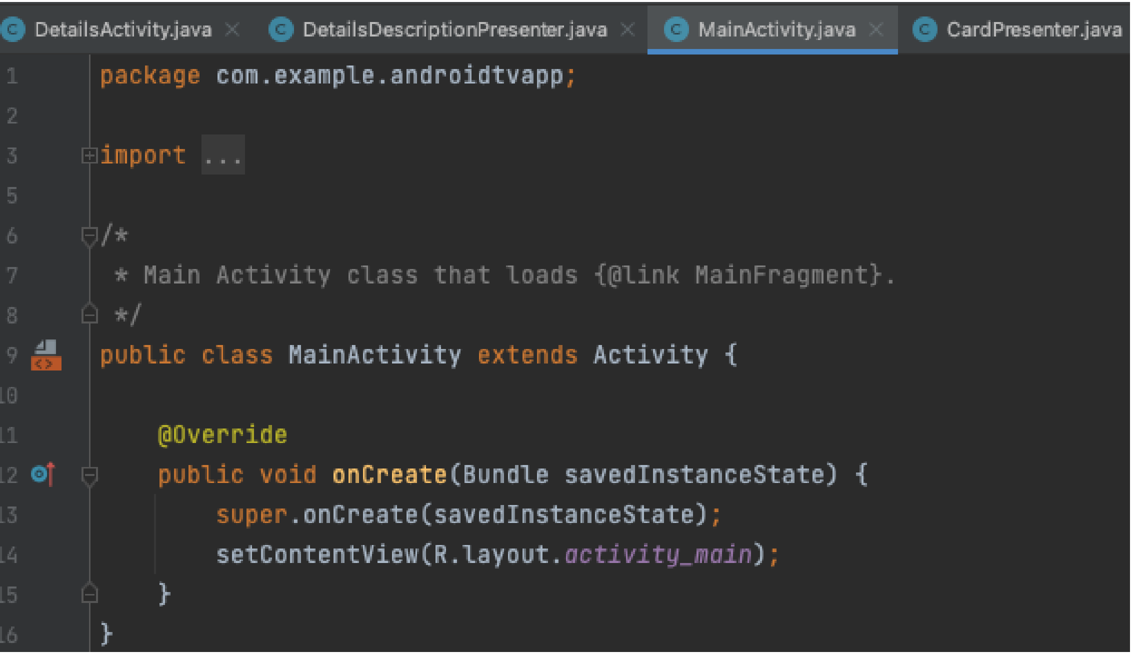


```xml
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/details_fragment"
    android:name="com.example.androidtvapp.VideoDetailsFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DetailsActivity"
    tools:deviceIds="tv" />
```

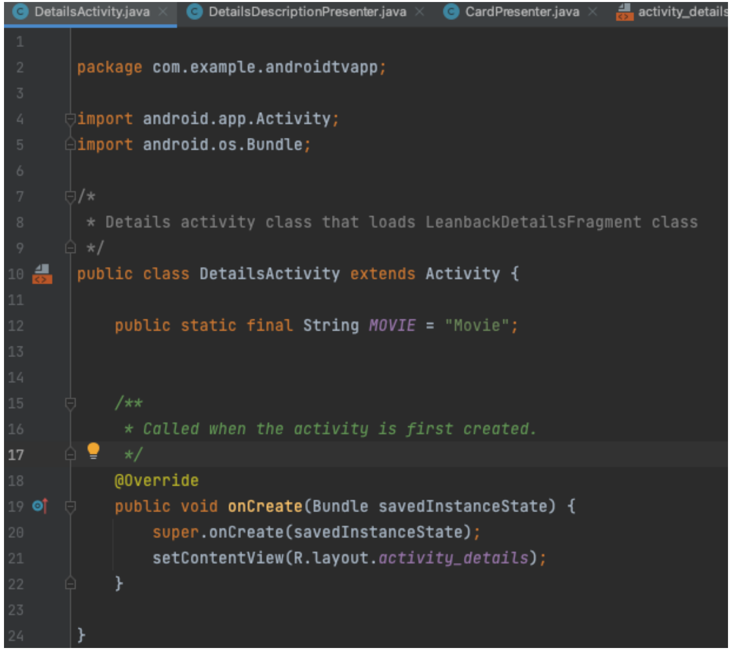


```xml
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_browse_fragment"
    android:name="com.example.androidtvapp.MainFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:deviceIds="tv"
    tools:ignore="MergeRootFrame" />
```

HW – Hint

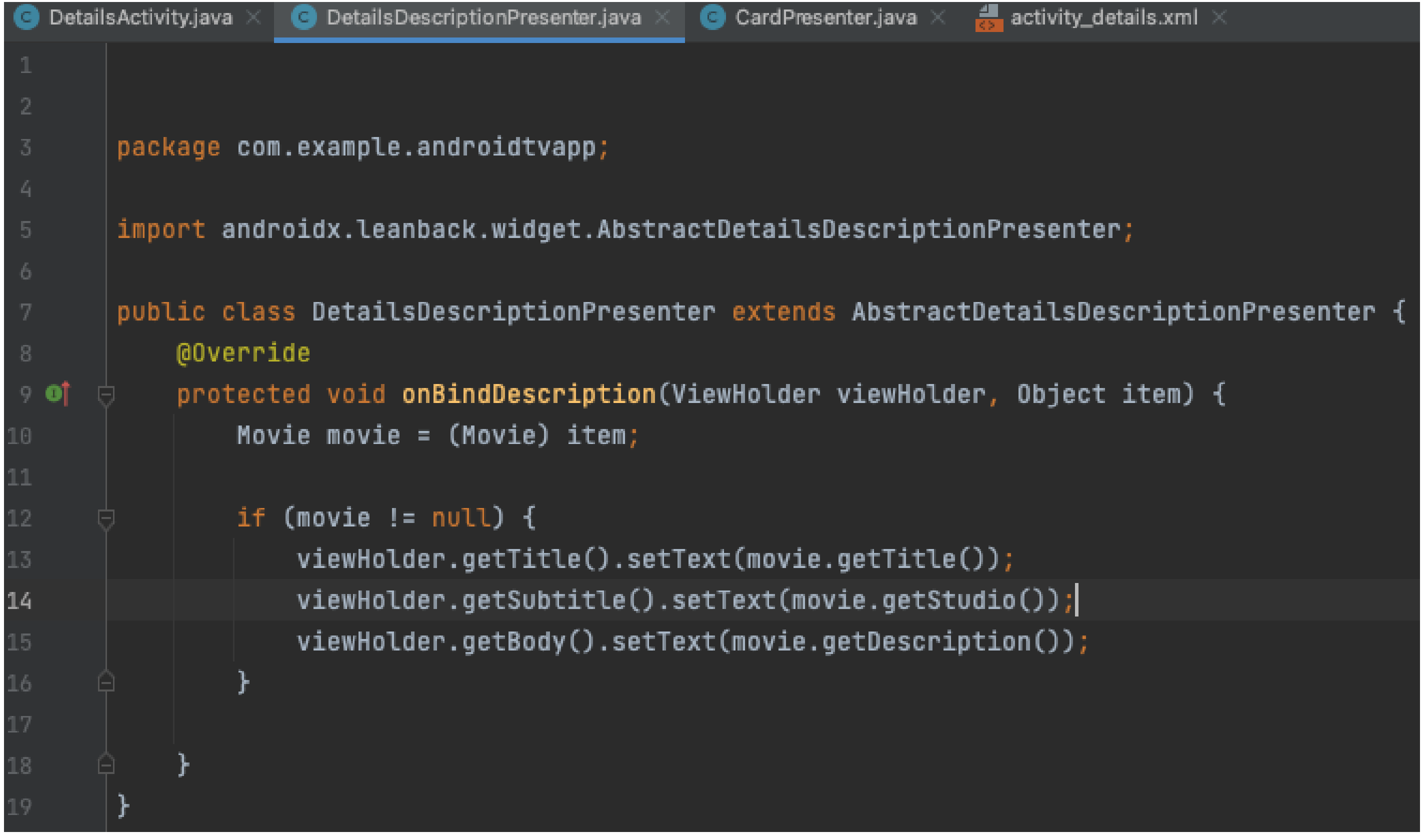


```java
package com.example.androidtvapp;

import ...

/*
 * Main Activity class that loads {@link MainFragment}.
 */
public class MainActivity extends Activity {

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



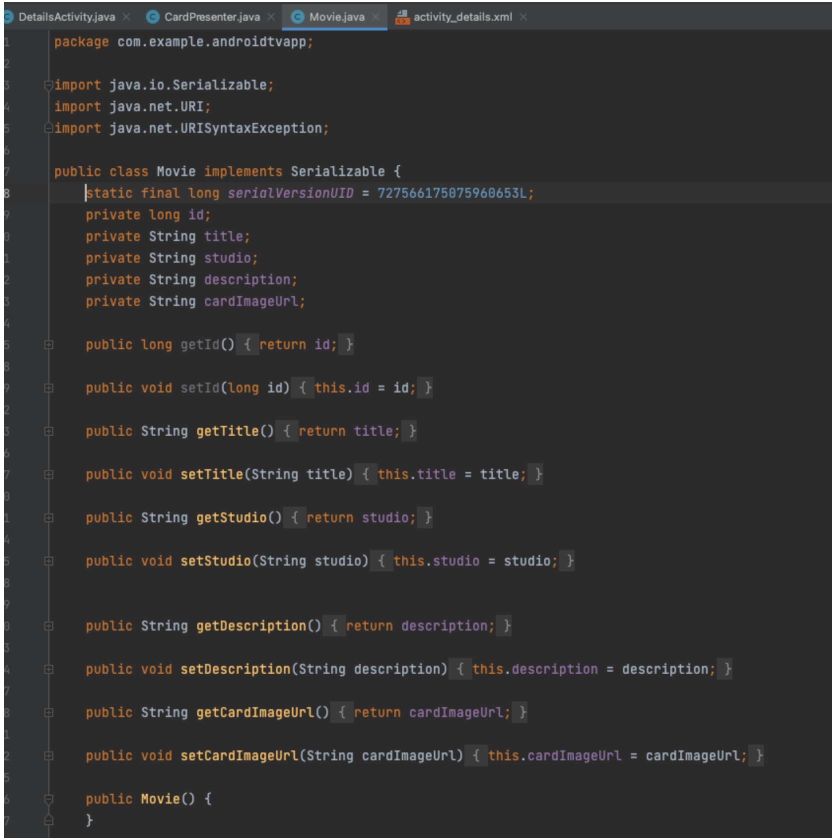
```java

package com.example.androidtvapp;

import android.app.Activity;
import android.os.Bundle;

/*
 * Details activity class that loads LeanbackDetailsFragment class
 */
public class DetailsActivity extends Activity {

    public static final String MOVIE = "Movie";


    /**
     * Called when the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);
    }

}
```

HW – Hint

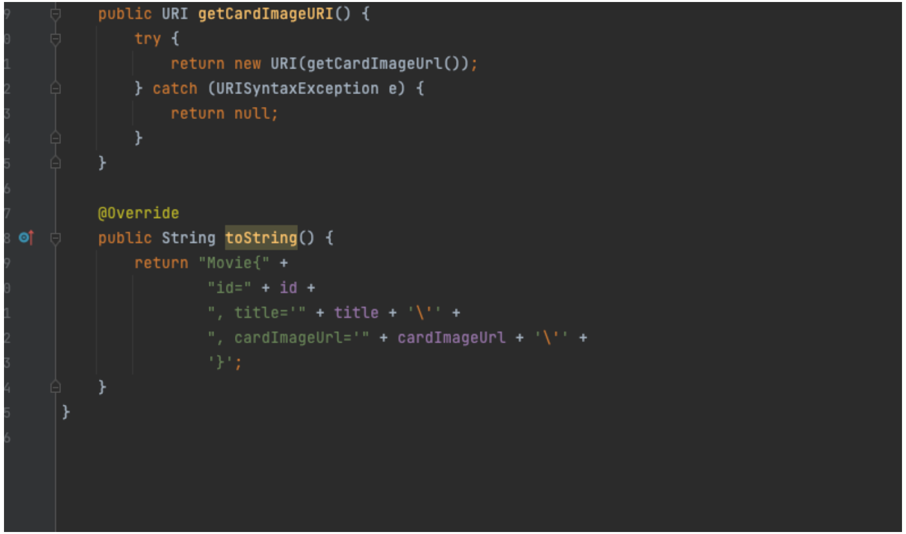


```java


package com.example.androidtvapp;

import androidx.leanback.widget.AbstractDetailsDescriptionPresenter;

public class DetailsDescriptionPresenter extends AbstractDetailsDescriptionPresenter {
    @Override
    protected void onBindDescription(ViewHolder viewHolder, Object item) {
        Movie movie = (Movie) item;

        if (movie != null) {
            viewHolder.getTitle().setText(movie.getTitle());
            viewHolder.getSubtitle().setText(movie.getStudio());
            viewHolder.getBody().setText(movie.getDescription());
        }

    }
}
```

HW – Movie

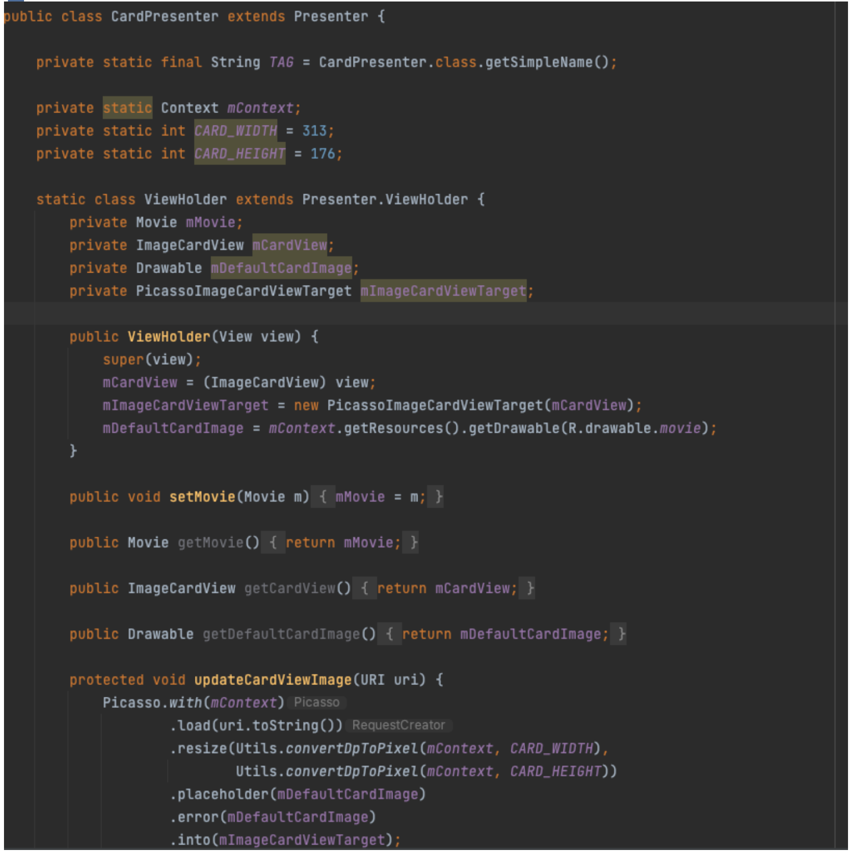


```java
package com.example.androidtvapp;

import java.io.Serializable;
import java.net.URI;
import java.net.URISyntaxException;

public class Movie implements Serializable {
    static final long serialVersionUID = 727566175075960653L;
    private long id;
    private String title;
    private String studio;
    private String description;
    private String cardImageUrl;

    public long getId() { return id; }

    public void setId(long id) { this.id = id; }

    public String getTitle() { return title; }

    public void setTitle(String title) { this.title = title; }

    public String getStudio() { return studio; }

    public void setStudio(String studio) { this.studio = studio; }


    public String getDescription() { return description; }

    public void setDescription(String description) { this.description = description; }

    public String getCardImageUrl() { return cardImageUrl; }

    public void setCardImageUrl(String cardImageUrl) { this.cardImageUrl = cardImageUrl; }

    public Movie() {
    }
```

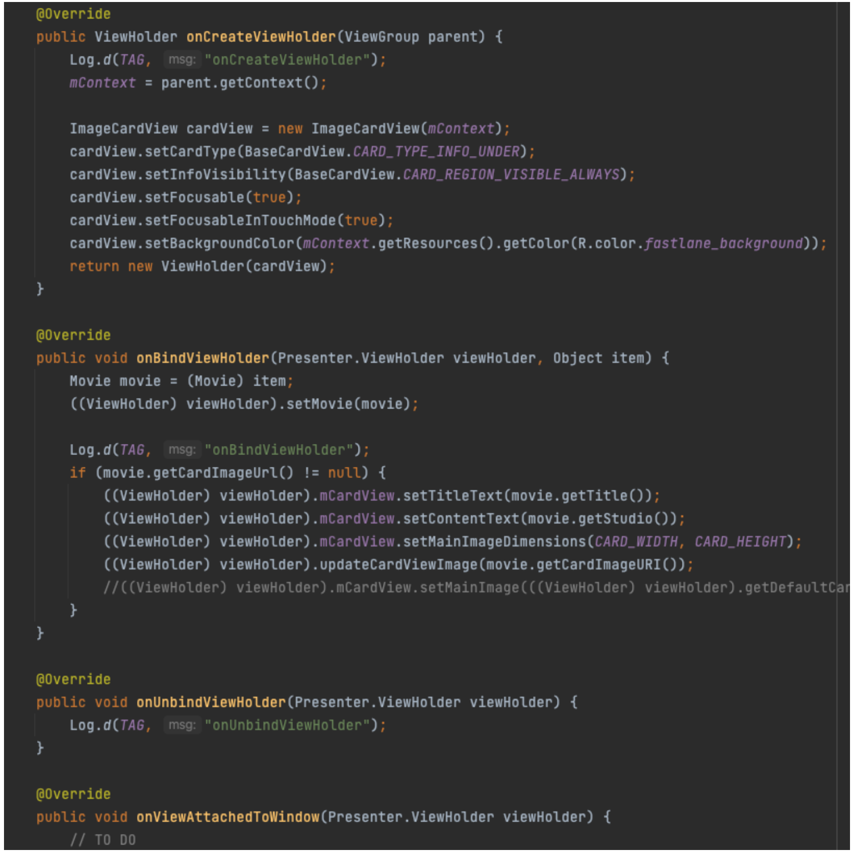


```java
    public URI getCardImageURI() {
        try {
            return new URI(getCardImageUrl());
        } catch (URISyntaxException e) {
            return null;
        }
    }

    @Override
    public String toString() {
        return "Movie{" +
                "id=" + id +
                ", title='" + title + '\'' +
                ", cardImageUrl='" + cardImageUrl + '\'' +
                '}';
    }
}
```

HW – Hint - CardPresenter

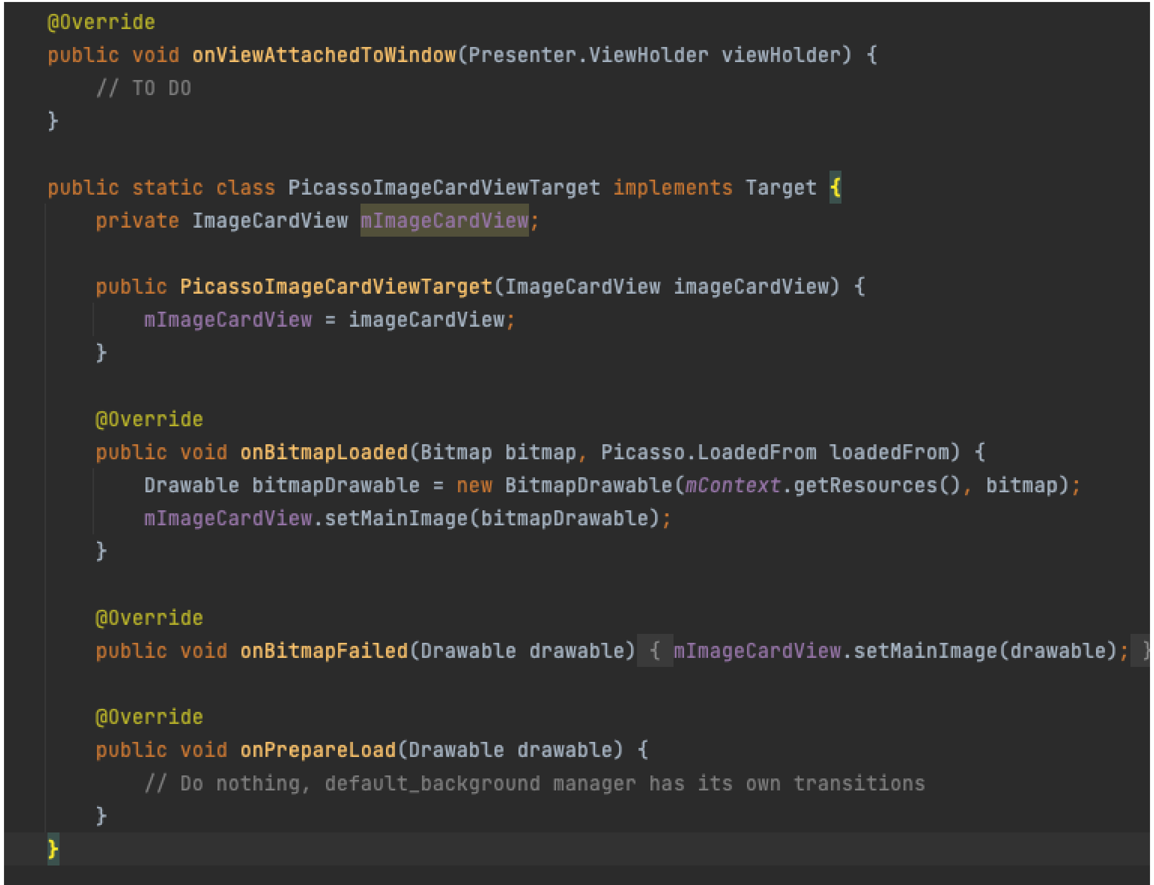


```java
public class CardPresenter extends Presenter {

    private static final String TAG = CardPresenter.class.getSimpleName();

    private static Context mContext;
    private static int CARD_WIDTH = 313;
    private static int CARD_HEIGHT = 176;

    static class ViewHolder extends Presenter.ViewHolder {
        private Movie mMovie;
        private ImageCardView mCardView;
        private Drawable mDefaultCardImage;
        private PicassoImageCardViewTarget mImageCardViewTarget;

        public ViewHolder(View view) {
            super(view);
            mCardView = (ImageCardView) view;
            mImageCardViewTarget = new PicassoImageCardViewTarget(mCardView);
            mDefaultCardImage = mContext.getResources().getDrawable(R.drawable.movie);
        }

        public void setMovie(Movie m) { mMovie = m; }

        public Movie getMovie() { return mMovie; }

        public ImageCardView getCardView() { return mCardView; }

        public Drawable getDefaultCardImage() { return mDefaultCardImage; }

        protected void updateCardViewImage(URI uri) {
            Picasso.with(mContext) Picasso
                    .load(uri.toString()) RequestCreator
                    .resize(Utils.convertDpToPixel(mContext, CARD_WIDTH),
                            Utils.convertDpToPixel(mContext, CARD_HEIGHT))
                    .placeholder(mDefaultCardImage)
                    .error(mDefaultCardImage)
                    .into(mImageCardViewTarget);
```



```java
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent) {
    Log.d(TAG, msg: "onCreateViewHolder");
    mContext = parent.getContext();

    ImageCardView cardView = new ImageCardView(mContext);
    cardView.setCardType(BaseCardView.CARD_TYPE_INFO_UNDER);
    cardView.setInfoVisibility(BaseCardView.CARD_REGION_VISIBLE_ALWAYS);
    cardView.setFocusable(true);
    cardView.setFocusableInTouchMode(true);
    cardView.setBackgroundColor(mContext.getResources().getColor(R.color.fastlane_background));
    return new ViewHolder(cardView);
}

@Override
public void onBindViewHolder(Presenter.ViewHolder viewHolder, Object item) {
    Movie movie = (Movie) item;
    ((ViewHolder) viewHolder).setMovie(movie);

    Log.d(TAG, msg: "onBindViewHolder");
    if (movie.getCardImageUrl() != null) {
        ((ViewHolder) viewHolder).mCardView.setTitleText(movie.getTitle());
        ((ViewHolder) viewHolder).mCardView.setContentText(movie.getStudio());
        ((ViewHolder) viewHolder).mCardView.setMainImageDimensions(CARD_WIDTH, CARD_HEIGHT);
        ((ViewHolder) viewHolder).updateCardViewImage(movie.getCardImageURI());
        //((ViewHolder) viewHolder).mCardView.setMainImage(((ViewHolder) viewHolder).getDefaultCar
    }
}

@Override
public void onUnbindViewHolder(Presenter.ViewHolder viewHolder) {
    Log.d(TAG, msg: "onUnbindViewHolder");
}

@Override
public void onViewAttachedToWindow(Presenter.ViewHolder viewHolder) {
    // TO DO
```

HW – Hint



```java
@Override
public void onViewAttachedToWindow(Presenter.ViewHolder viewHolder) {
    // TO DO
}

public static class PicassoImageCardViewTarget implements Target {
    private ImageCardView mImageCardView;

    public PicassoImageCardViewTarget(ImageCardView imageCardView) {
        mImageCardView = imageCardView;
    }

    @Override
    public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom loadedFrom) {
        Drawable bitmapDrawable = new BitmapDrawable(mContext.getResources(), bitmap);
        mImageCardView.setMainImage(bitmapDrawable);
    }

    @Override
    public void onBitmapFailed(Drawable drawable) { mImageCardView.setMainImage(drawable); }

    @Override
    public void onPrepareLoad(Drawable drawable) {
        // Do nothing, default_background manager has its own transitions
    }
}
```

References

- https://web.stanford.edu/
- https://developer.android.com/
- https://introcs.cs.princeton.edu/
- https://www.cs.utexas.edu/
- https://www.codepath.com/
- https://geeksforgeeks.com/
- proandroidev

Q & A

Thank you for listening

“Coming together is a beginning;
Keeping together is progress;
Working together is success.”
- HENRY FORD