CAT 3

Alejandro Pérez Bueno

Nov 28, 2023

Table of Contents

Chapter 1

Create the local database

1.1 CREATE SQL

The code to be added is the following:

```
db.execSQL(SQL_CREATE_user)
db.execSQL(SQL_CREATE_seminar)
db.execSQL(SQL_CREATE_item)
db.execSQL(SQL_CREATE_request)
db.execSQL(SQL_CREATE_user_seminar)
```

Note

See `DbHelper.kt` for more details

1.2 command_list

```
for (command in command_list) {
    db.execSQL(command)
}
```

Note

See `DbHelper.kt` for more details

Chapter 2

Local filesystem

Here are the comments I added to the code:

```
// Create the /media directory in the app's internal storage
// Create the /media/seminar directory in the app's internal storage
// Move three logos to the /media/seminar directory we just created
// Create the /media/item directory in the app's internal storage
// Move three dog images to the /media/item directory we just created
// Move three medicine images to the /media/item directory we just created
// Move three AI images to the /media/item directory we just created
// Create the /media/request directory in the app's internal storage
```

Note

See `DbHelper.kt` for more details

Chapter 3

Implement the class DataSourceLocal

3.1 `str_sql`

> Note
>
> See `DataSourceLocal.kt` for more details

3.2 `selectSeminarsUser`

Here is the code to be added to the code:

```
val cursor = db.rawQuery(str_sql, null)
while (cursor.moveToNext()) {
    userSeminarList.add(Seminary(cursor.getInt(cursor.getColumnIndex("sem_id")),
        cursor.getString(cursor.getColumnIndex("sem_name")),
        cursor.getInt(cursor.getColumnIndex("sem_duration")),
        context.filesDir.path + "/media/seminar/" +
                cursor.getInt(cursor.getColumnIndex("sem_id")) + ".jpg"))
cursor.close()
```

The code launches the query specified in `str_sql` and then iterates over every result to create a seminary from each result.

> Note
>
> See `DataSourceLocal.kt` for more details

3.3 selectItems part 1

Here are the lines to be added to the code:

```
while (cursor.moveToNext()) {
    seminarItemList.add(Item(ItemType.BASIC,
        cursor.getInt(cursor.getColumnIndex("item_id")),
        cursor.getString(cursor.getColumnIndex("item_title")),
        cursor.getString(cursor.getColumnIndex("item_description"))))
}
cursor.close()
```

Note

See `DataSourceLocal.kt` for more details

3.4 selectItems part 2

Here are the lines to be added to the code:

```
val cursor = db.rawQuery(str_sql, null)
var UserRequestList = mutableListOf<UserRequest>()
while (cursor.moveToNext()) {
    UserRequestList.add(UserRequest(cursor.getLong(cursor.getColumnIndex("request_id")),
        cursor.getString(cursor.getColumnIndex("request_description"))))
}
cursor.close()
```

Note

See `DataSourceLocal.kt` for more details

Chapter 4

Use the class DataSourceLocal

Here is how the code looks:

```
val Default = DataSourceType.LocalStorage
```

Note

See `DataSourceLocal.kt` for more details

Chapter 5

Read files from the filesystem

5.1 `SeminarsAdapter`

Here are the lines to be added to the code:

```
val imagePath = item.image_path
val bitmap = BitmapFactory.decodeFile(imagePath)
itemImageView.setImageBitmap(bitmap)
```

In this code, we use the `decodeFile()` to load the bitmap from the filesystem using `imagePath` value as a parameter. We then assign the loaded bitmap to `itemImageView` using `setImageBitmap()`.

> ℹ Note
>
> See `SeminarsAdapter.kt` for more details

5.2 `ItemsAdapter`

Here are the lines to be added to the code:

```
val imagePath = item.image_path
val bitmap = BitmapFactory.decodeFile(imagePath)
itemImageView?.setImageBitmap(bitmap)
```

> ℹ Note
>
> See `ItemsAdapter.kt` for more details

5.3 DetailFragment

Here are the lines to be added to the code:

```
val imagePath = item?.image_path
val bitmap = BitmapFactory.decodeFile(imagePath)
v2.setImageBitmap(bitmap)
```

Note

See ItemsAdapter.kt for more details

Chapter 6

Create an activity AddRequest

6.1 Activity Creation

Note

See Figure 8.1 to verify the parameters used in the creation of this activity

6.2 Activity Design

Here is how the layout file looks:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/gray">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp"
        android:background="@drawable/purple_border">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:gravity="center_vertical"
            android:padding="8dp"
            android:text="Title" />

        <EditText
            android:id="@+id/editTitle"
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:background="@color/white"
            android:gravity="center_vertical"
            android:padding="8dp" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="40dp"
            android:gravity="center_vertical"
            android:padding="8dp"
            android:text="Description" />

        <EditText
            android:id="@+id/editDescription"
            android:layout_width="match_parent"
            android:layout_height="460dp"
            android:background="@color/white"
            android:gravity="top|left"
            android:inputType="textMultiLine"
            android:padding="8dp" />

        <Button
            android:id="@+id/btnSelectImage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:background="#673AB7"
            android:gravity="center_horizontal"
            android:text="SELECT IMAGE"
            android:textColor="@android:color/white" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/white"
            android:gravity="center_vertical"
            android:padding="8dp"
            android:text="Image" />

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

Note

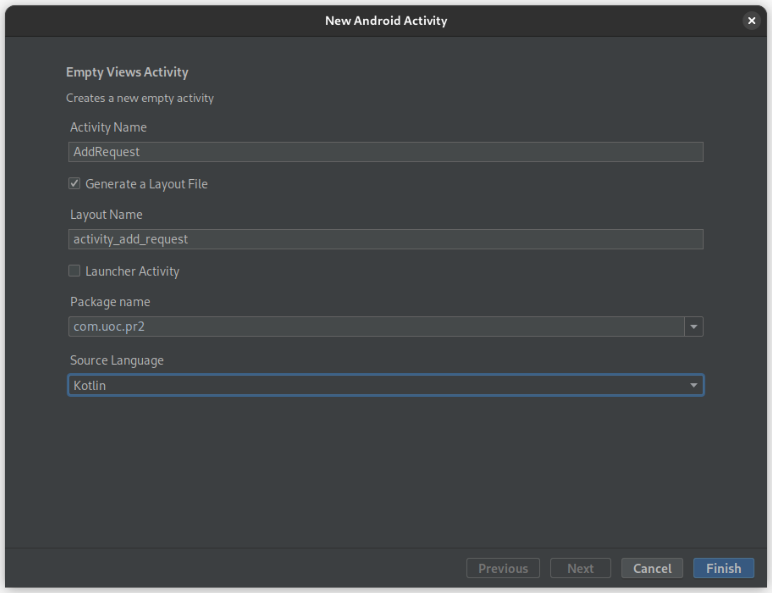
See `activity_add_request.xml` for more details
See Figure 8.2 to see the result

Chapter 7

Testing

Chapter 8

Annexes



Empty Views Activity

Creates a new empty activity

Activity Name: AddRequest

[✓] Generate a Layout File

Layout Name: activity_add_request

[ ] Launcher Activity

Package name: com.uoc.pr2

Source Language: Kotlin

Figure 8.1: Creation of `AddRequest` empty views activity

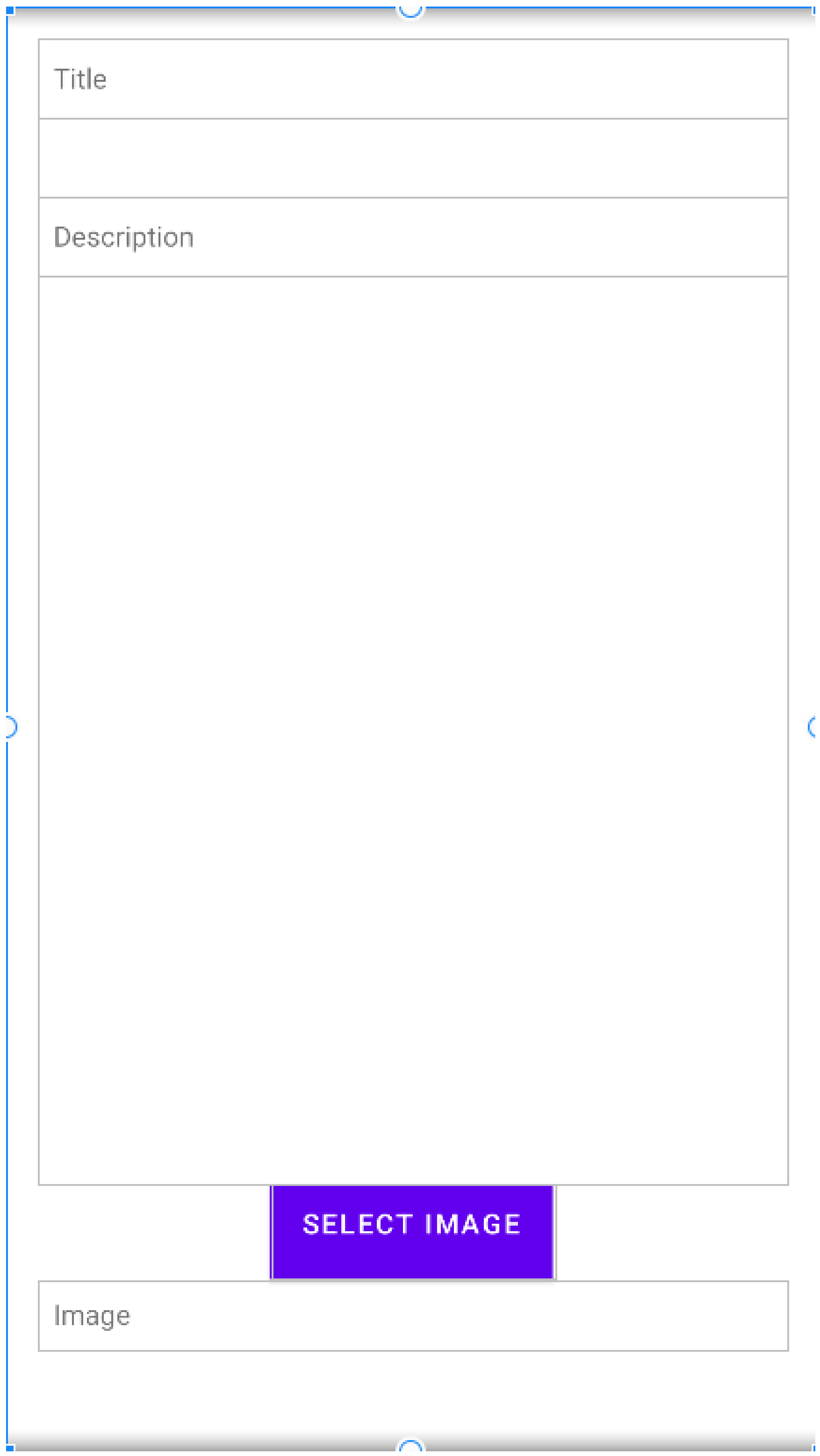


Figure 8.2: Creating the `AddReqeust` layout