ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

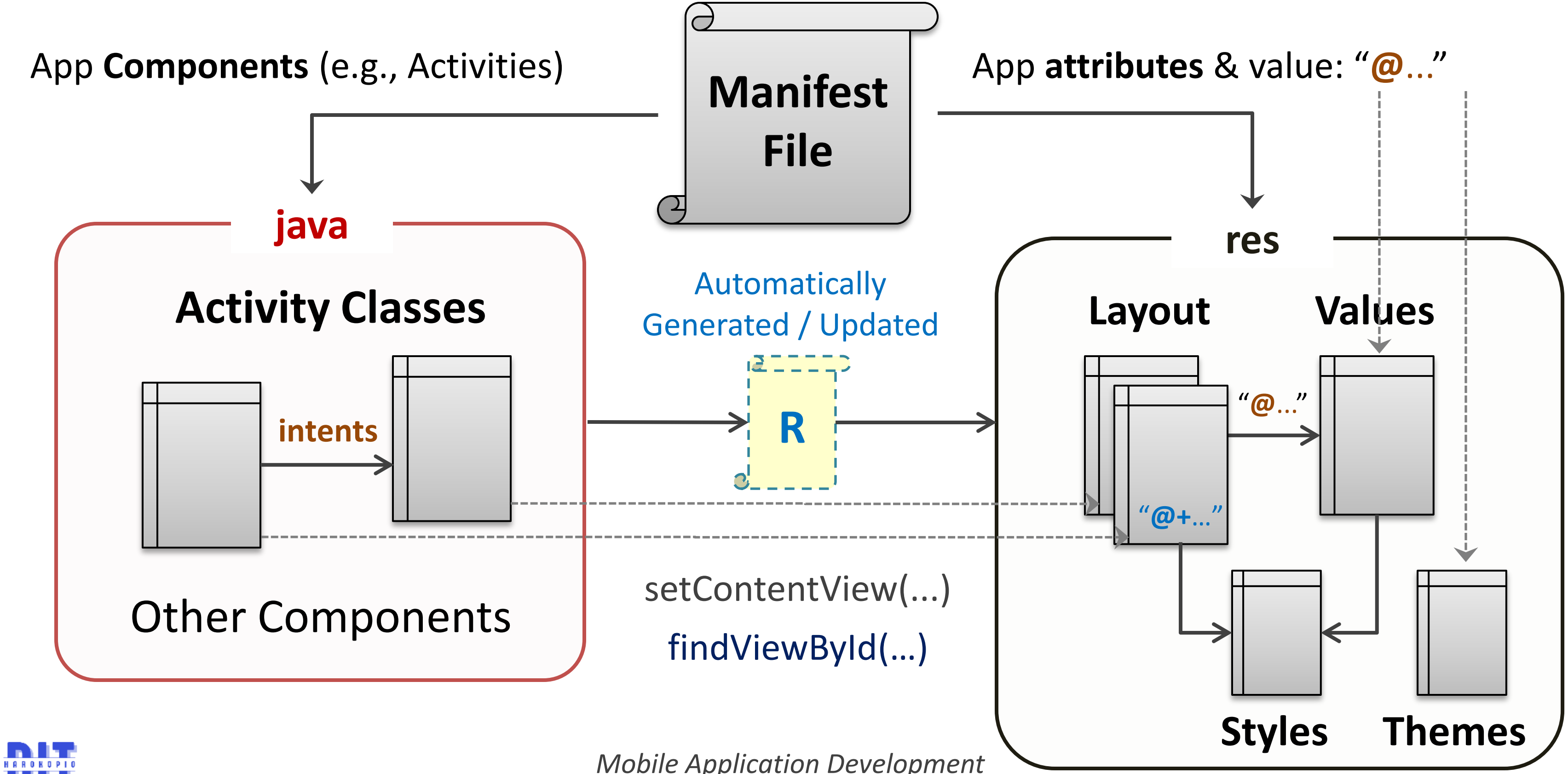
Τμήμα Πληροφορικής
και Τηλεματικής

Mobile Application Development

Lab 2

Dr. Efthymios Chondrogiannis

Android Development – Main Entities



Data and File Storage

Android uses a file system that is like disk-based file systems

- **App-specific storage**: Store **files** that are meant for your **app's use only**, either in dedicated directories within an internal storage volume or different dedicated directories within external storage.
- **Shared storage**: Store **files** that your app intends to **share with other apps**, including media, documents, and other files.
- **Preferences**: Store **private**, primitive data in **key-value** pairs.
- **Databases**: Store **structured data** in a **private** database using the **Room** persistence library.

Data and file storage overview: https://developer.android.com/training/data-storage

Lab 2 Tasks

- App Development using Android Studio
- **Task 1**: App Text Files and Shared Preferences
- **Task 2**: Threads, Shared Files and Room (SQLite)

App Specific Files – Read / Write

- **Read File** – Input Stream
  - `getResources().openRawResource(R.raw.file_name);`
  - `BufferedReader & InputStreamReader`

- **Write File** – Output Stream
  - `openFileOutput("new_file_name.txt", MODE_PRIVATE);`
  - `FileOutputStream`

Shared Preferences – Read / Write

```
SharedPreferences sp = getSharedPreferences(“DEF_SP”, MODE_PRIVATE);
```

- Write to Shared Preferences

```
SharedPreferences.Editor editor = sp.edit();
// ...
editor.putInt( param_key , param_value);
// ...
editor.apply(); // Data Stored to Disk asynchronously
```

MODE_PRIVATE: The created file can only be accessed by the calling application

- Read from Shared Preferences

```
int default_value = getResources().getInteger(R.string.param_key);
int param = sp.getInt( param_key, default_value );
```

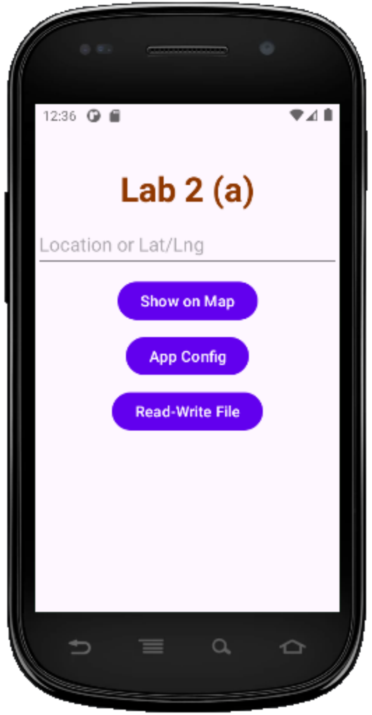
Task 1

- Create and Android Application that provides the following functionality:

  a) Show the Location of the Point of Interest provided in the **Google Maps**

  b) Read data from a **Text File** and Write new data provided through UI in a new one

  c) Save default location provided through a UI using **Shared Preferences**



Shared Files

- For user data that can or should **be accessible to other apps** and saved even if the user uninstalls your app.
  - An app invokes an **intent** that contains a **storage-related action**. This action corresponds to a specific use case that the framework makes available.
  - The user sees a **system picker**, allowing them to browse a documents provider and **choose a location or document** where the storage-related action takes place.
  - The **app gains read and write access** to a **URI** that represents the user's chosen location or document. Using this URI, the app can perform operations on the chosen location.

```
getContentResolver().openOutputStream(uri, mode) // e.g., mode = "w“
```

Access documents and other files from shared storage: https://developer.android.com/training/data-storage/shared/documents-files

ROOM components

- There are three major components in Room:

  ○ **Entities**

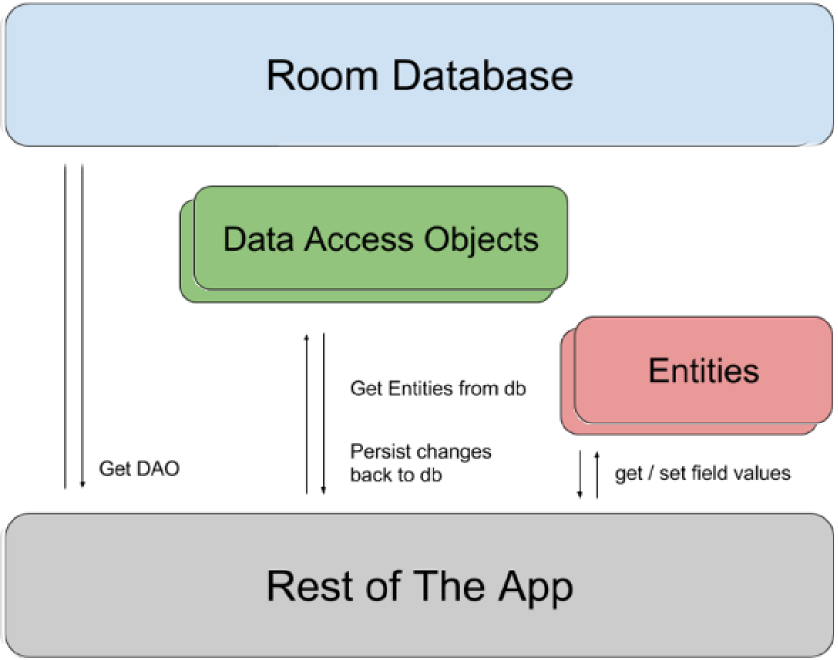
  Represent **tables** in your app's DB

  ○ **Data Access Objects (DAOs)**

  Provide methods that your app can use to **query**, **update**, **insert**, and **delete** data

  ○ **Database**

  Serves as **the main access point** for the underlying connection to your DB



Room: https://developer.android.com/training/data-storage/room

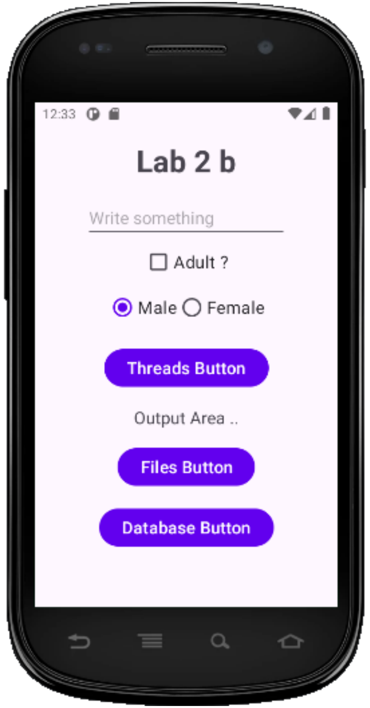
Task 2

- Create and Android Application that provides the following functionality:

  a) Run a simple task in a **new Thread** that accordingly updates the **UI**

  b) Store data provided in a **Text File** in **Downloads**

  c) Store data provided through the UI in **SQLite DB** using Room



Conclusion

- Android App Development using Android Studio
- **Task 1**: App Text Files and Shared Preference
- **Task 2**: Shared Text Files and SQLite DB using Room

Thank You!