Content Provider: Data Sharing across App

Course: Mobile Application Development

Unit : III – Building Blocks of Mobile Apps - II

Class / Semester: II MCA / III Semester

SESSION OBJECTIVES

- Understand the characteristics of User App interface
- Know about elements constitutes user interface
- Differentiate about programming and non-programming elements
- Utlize the Android features on user interface design

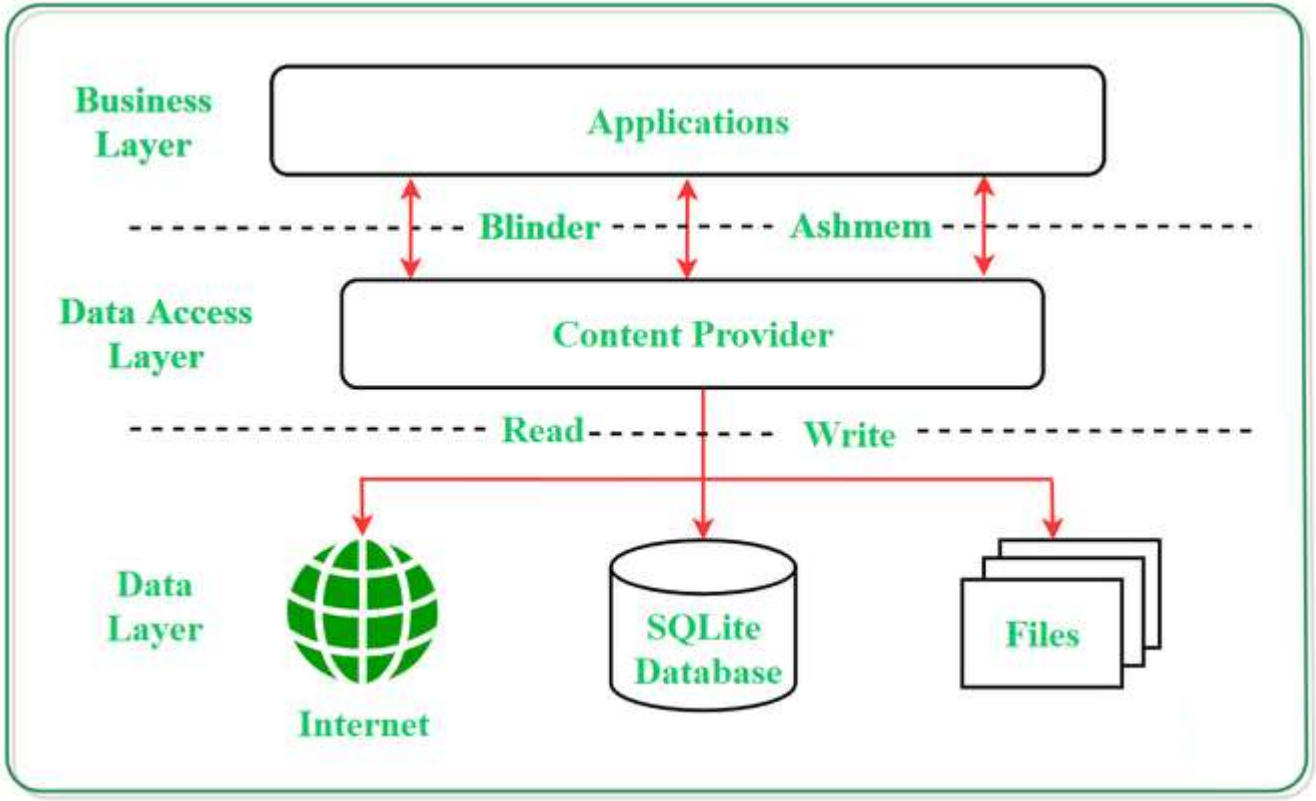
USE CASE PROBLEM

Association of one gated community decided to introduce system for visitor's recording to the Apartment houses

- Security at the front gate is the end user
- Mobile notification sent to Resident/Host for approval
- Resident may accept/reject the visitor
- Security falls for manual checking if no response is received
- Pre-authorized guest provision may be given (Expected visitor)

CONTENT PROVIDER

- It acts like a central repository in which data of the applications are stored, and it facilitates other applications to securely access and modifies that data
- Users can manage to store the application data like images, audio, videos, and personal contact information by storing them in **SQLite Database**, in files, or even on a network
- In order to share the data, content providers have certain permissions (grant or restrict the rights) to other applications to interfere with the data
- It hides the implementation details of the data from other apps to provide an abstract and secure way of sharing data across apps
- We can carry out CRUD operations on data of other apps as a black box
- Data of in-built apps are made accessible using in-built content provider
- other apps can access our app's data using customer-built content provider


Business Layer
Applications
Blinder
Ashmem
Data Access Layer
Content Provider
Read
Write
Data Layer
Internet
SQLite Database
Files

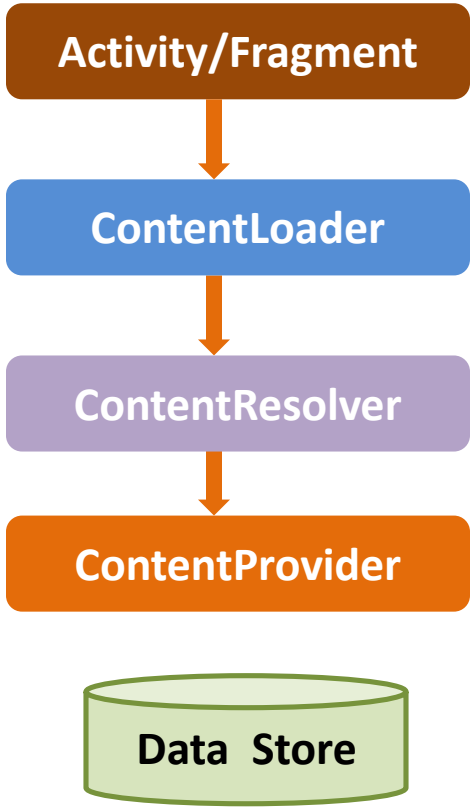
CONTENT-URI

- Content URI is the key concept used to access the data from a content provider, URI is used as a query string
- **Structure of a Content URI:** *content://authority/optionalPath/optionalID*
  - **content:// –** Mandatory part, represents that the given URI is a Content URI.
  - **authority –** Signifies the name of the content provider like contacts, browser, etc. This part must be unique for every content provider.
  - **optionalPath –** Specifies the type of data provided by the content provider. Content providers to support different types of data
  - **optionalID –** It is a numeric value that is used when there is a need to access a particular record
- If an ID is mentioned in a URI then it is an id-based URI otherwise a directory-based URI

- Four fundamental operations are possible (called CRUD operations)

Methods
Create: Operation to create data in a content provider
Read: Used to fetch data from a content provider
Update: To modify existing data
Delete: To remove existing data from the storage.

- UI components of android applications like Activity and Fragments use an object **CursorLoader** to send query requests to **ContentResolver**
- The ContentResolver object sends requests (like create, read, update, and delete) to the **ContentProvider** as a client
- After receiving a request, ContentProvider process it and returns the desired result



CREATING CONTENT PROVIDER

- Create a class in the same directory where the that MainActivity file resides and this class must extend the ContentProvider base class
- To access the content, define a content provider URI address
- Create a database to store the application data
- Implement the six abstract methods of ContentProvider class
- Register the content provider in AndroidManifest.xml file using <provider> tag

REFERENCES

- Anubhav Pradhan, Anil V Deshpande, "Composing Mobile Apps using Android", Wiley Edition, 2014
- https://www.tutorialspoint.com/android/android_application_components.htm
- https://www.javatpoint.com/android-core-building-blocks

http://yuliana.lecturer.pens.ac.id/Android/Download/ppt/