Mobile Application Development

(Storages)

Instructor: Thanh Binh Nguyen

February 1st, 2020

S3Lab
Smart Software System Laboratory

“The future of mobile is the future of online. It is how people access online content now.”

– David Murphy, Founder and Editor of Mobile Marketing Daily

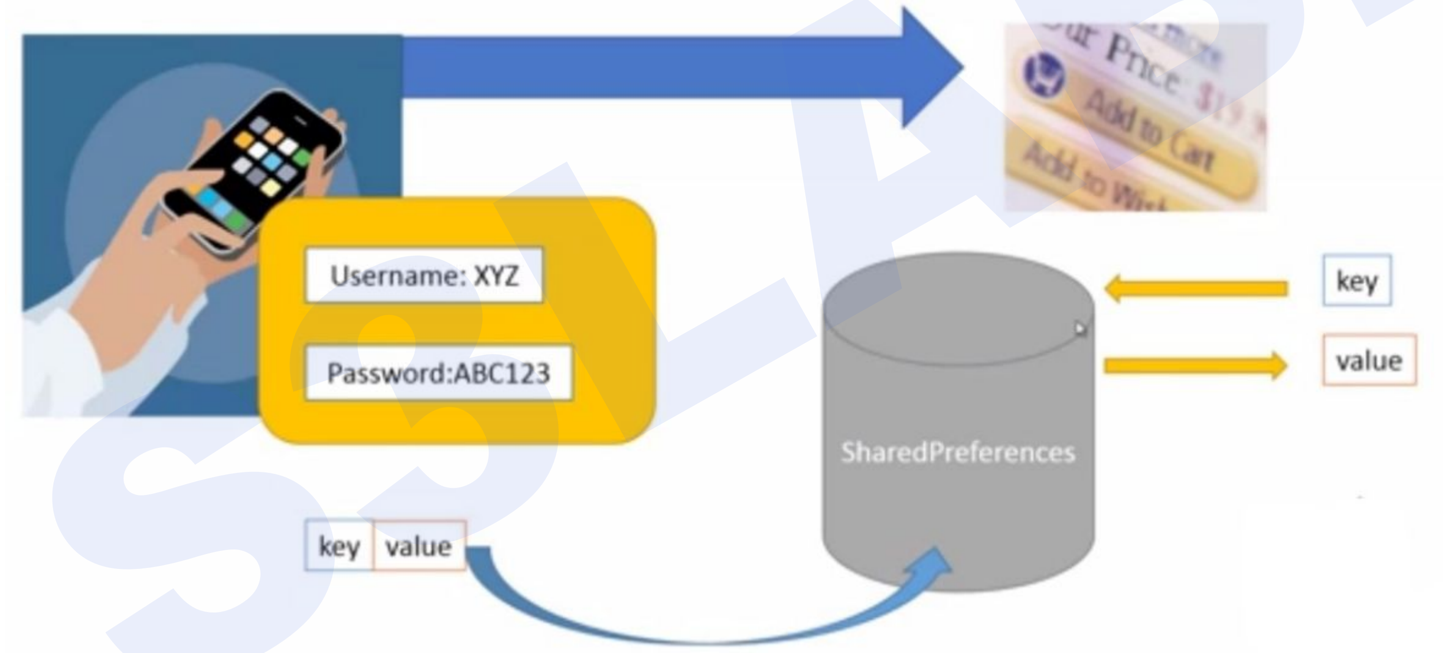
Data Storages

Methods

- **Shared Preferences**: Store private primitive data in key-value pairs.
- **Internal Storage**: Store private data on the device memory.
- **External Storage**: Store public data on the shared external storage.
- **SQLite Databases**: Store structured data in a private database.
- **Network Connection**: Store data on the web with your own network server.

Data Storages

Shared Preferences



Data Storages

Shared Preferences

- Useful for storing and retrieving primitive data in **Key** - **Value** pairs.
- Lightweight usage, such as saving **application settings**.
- Typical usage of **SharedPreferences** is for saving application such as username and password, auto login flag, remember-user flag etc.

Data Storages

Shared Preferences

- The shared preferences information is stored in an XML file on the device
    - Typically in/data/data//shared_prefs
- SharedPreferences can be associated with the entire application, or to a specific activity.
- Use the **getSharedPrefernces()** method to get access to the preferences

Data Storages

Shared Preferences

- Example:

  SharedPrefernces prefs = this.getSharedPrefernces(“myPrefs”, MODE_PRIVATE)

  boolean silent = prefs.getBoolean("silentMode", false);

- If the preferences XML file exist, it is opened, otherwise it is created.
- To Control access permission to the file:
  - **MODE_PRIVATE**: private only to the application
  - **MODE_WORLD_READABLE**: all application can read XML file
  - **MODE_WORLD_WRITABLE**: all application can write XML file

Data Storages

Shared Preferences

- To add Shared preferences, first an editor object is needed

```
Editor prefsEditor = prefs.edit();
```

- Then,use the put() method to add the key-value pairs

```
prefsEditor.putString(“username”,”D-Link”);
prefsEditor.putString(“password”,”vlsi#1@2”);
prefsEditor.putint(“times-login”,1);
prefsEditor.commit();
```

Data Storages

Shared Preferences

- To retrieve shared preferences data:

  String username = prefsEditor.getString(“username”.” ”);

  String password = prefsEditor.getString(“password”.” ”);

- If you are using **SharedPrefernces** for specific activity, then use **getPreferences()** method
  - No need to specify the name of the preferences file

Data Storages

Internal Storage

- Android can save files directly to the device internal storage.
- These files are private to the application and will be removed if you uninstall the application.
- We can create a file using **openFileOutput()** with parameter as **file name** and the **operating mode**.
- Generally not recommended to use files.

Data Storages

Internal Storage

- Similarly, we can open the file using **openFileInput()** passing the parameter as the **filename with extension**.
- File are use to store large amount of data
- Use I/O interfaces provided by java.io.* libraries to read/write files.
- Only local files can be accessed.

Data Storages

Internal Storage - file operations

- Use **context.openFileInput(string name)** to open a private input file stream related to a program.
- **Throw FileNotFoundException** when file does not exist.
- Syntax:

  FileInputStream in = this.openFileInput(“xyz.txt”)

  . . . . .

  In.close() //Close input stream

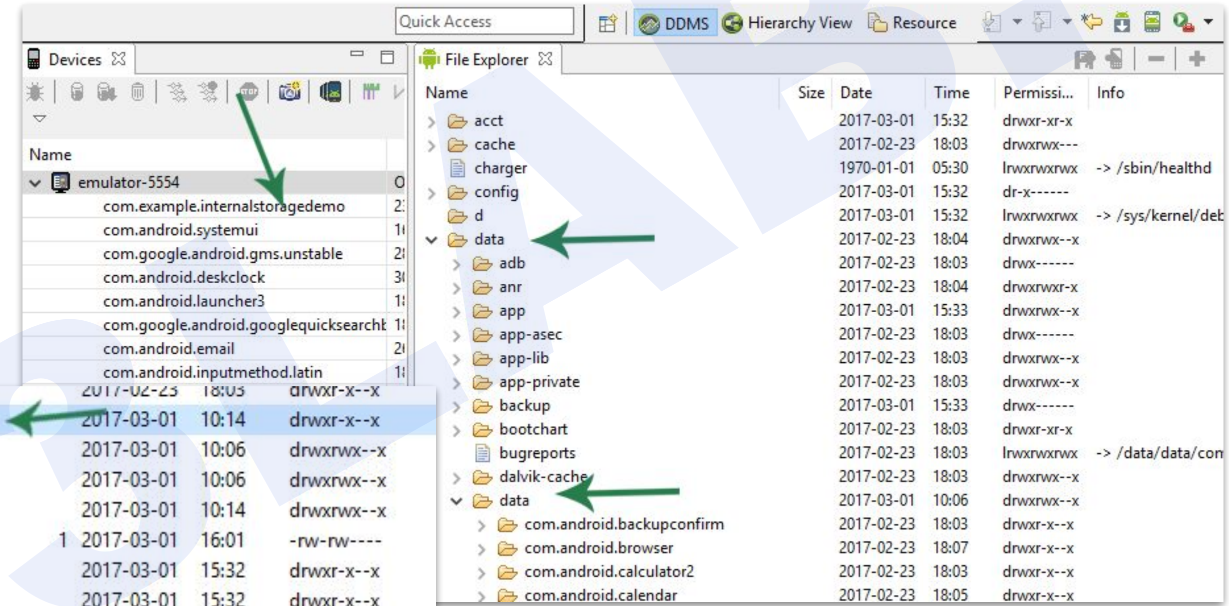
Data Storages

Internal Storage - file operations

- Use **context.openFileOutput(string name,int mode)** to open a private output file stream related to a program.
- The file will be created if it does not exist.
- Output stream can be opened in append mode, which means new data will be appended to end of the file.
    - *String mystring="Hello World"*
    - *FileOutputStream outfile = this.openFileOutput("myfile.txt",MODE_APPEND);*
    - *outfile.write(mystring.getBytes());*
    - *outfile.close();*

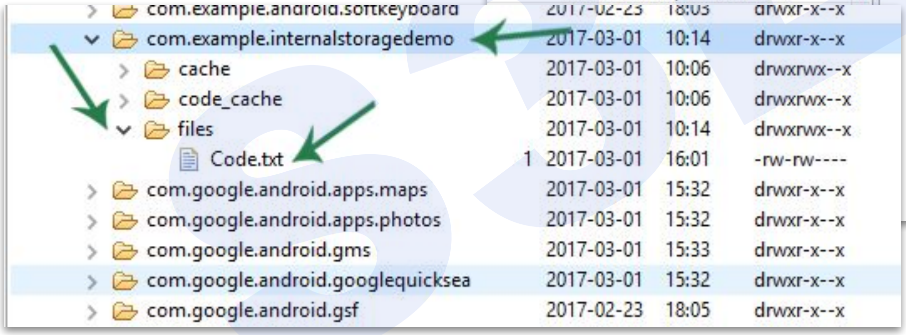
Data Storages

Internal Storage



Name	
emulator-5554	O
com.example.internalstoragedemo	2
com.android.systemui	1
com.google.android.gms.unstable	2
com.android.deskclock	3
com.android.launcher3	1
com.google.android.googlequicksearchb	1
com.android.email	2
com.android.inputmethod.latin	1

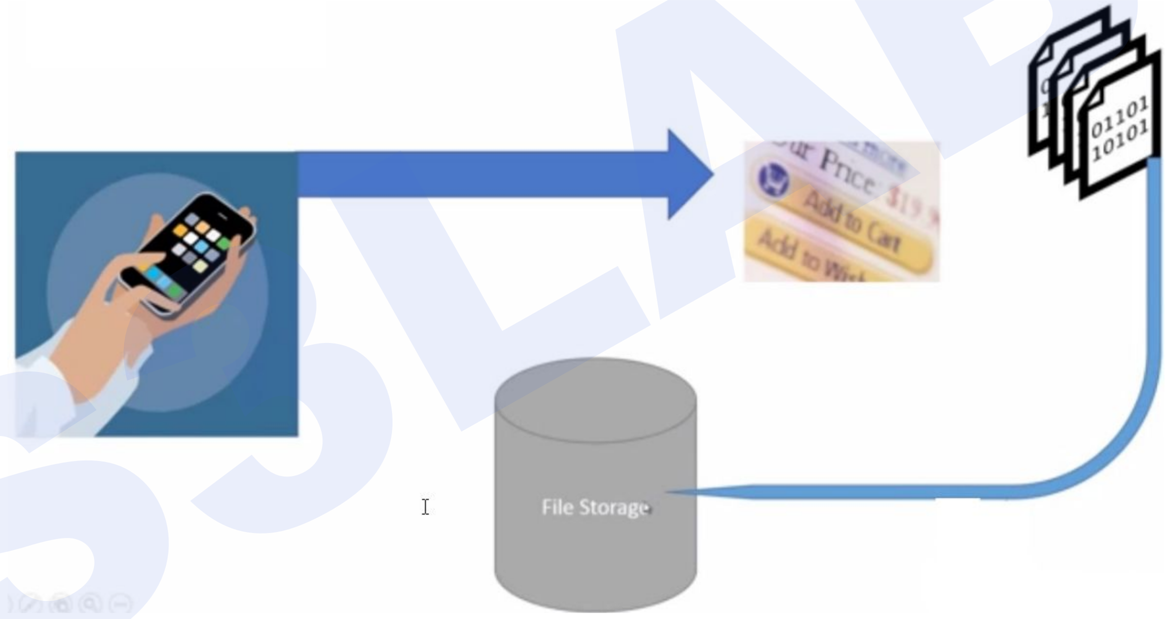
Name	Size	Date	Time	Permissi...	Info
acct		2017-03-01	15:32	drwxr-xr-x	
cache		2017-02-23	18:03	drwxrwx---	
charger		1970-01-01	05:30	lrwxrwxrwx	-> /sbin/healthd
config		2017-03-01	15:32	dr-x------	
d		2017-03-01	15:32	lrwxrwxrwx	-> /sys/kernel/deb
data		2017-02-23	18:04	drwxrwx--x	
adb		2017-02-23	18:03	drwx------	
anr		2017-02-23	18:04	drwxrwxr-x	
app		2017-03-01	15:33	drwxrwx--x	
app-asec		2017-02-23	18:03	drwx------	
app-lib		2017-02-23	18:03	drwxrwx--x	
app-private		2017-02-23	18:03	drwxrwx--x	
backup		2017-03-01	15:33	drwx------	
bootchart		2017-02-23	18:03	drwxr-xr-x	
bugreports		2017-02-23	18:03	lrwxrwxrwx	-> /data/data/com
dalvik-cache		2017-02-23	18:03	drwxrwx--x	
data		2017-03-01	10:06	drwxrwx--x	
com.android.backupconfirm		2017-02-23	18:03	drwxr-x--x	
com.android.browser		2017-02-23	18:07	drwxr-x--x	
com.android.calculator2		2017-02-23	18:03	drwxr-x--x	
com.android.calendar		2017-02-23	18:05	drwxr-x--x	



Name	Size	Date	Time	Permissions
com.example.android.softkeyboard		2017-02-23	18:03	drwxr-x--x
com.example.internalstoragedemo		2017-03-01	10:14	drwxr-x--x
cache		2017-03-01	10:06	drwxrwx--x
code_cache		2017-03-01	10:06	drwxrwx--x
files		2017-03-01	10:14	drwxrwx--x
Code.txt	1	2017-03-01	16:01	-rw-rw----
com.google.android.apps.maps		2017-03-01	15:32	drwxr-x--x
com.google.android.apps.photos		2017-03-01	15:32	drwxr-x--x
com.google.android.gms		2017-03-01	15:33	drwxr-x--x
com.google.android.googlequicksea		2017-03-01	15:32	drwxr-x--x
com.google.android.gsf		2017-02-23	18:05	drwxr-x--x

Data Storages

External Storage



Data Storages

External Storage

- Every Android-compatible device supports a shared “external storage” that you can use to save files
  - Removable storage media (such as an SD card)
  - Internal (non-removable) storage
- File saved to the external storage are **world readable** and can be modified by the user when they enable USB mass storage to transfer files on computer.

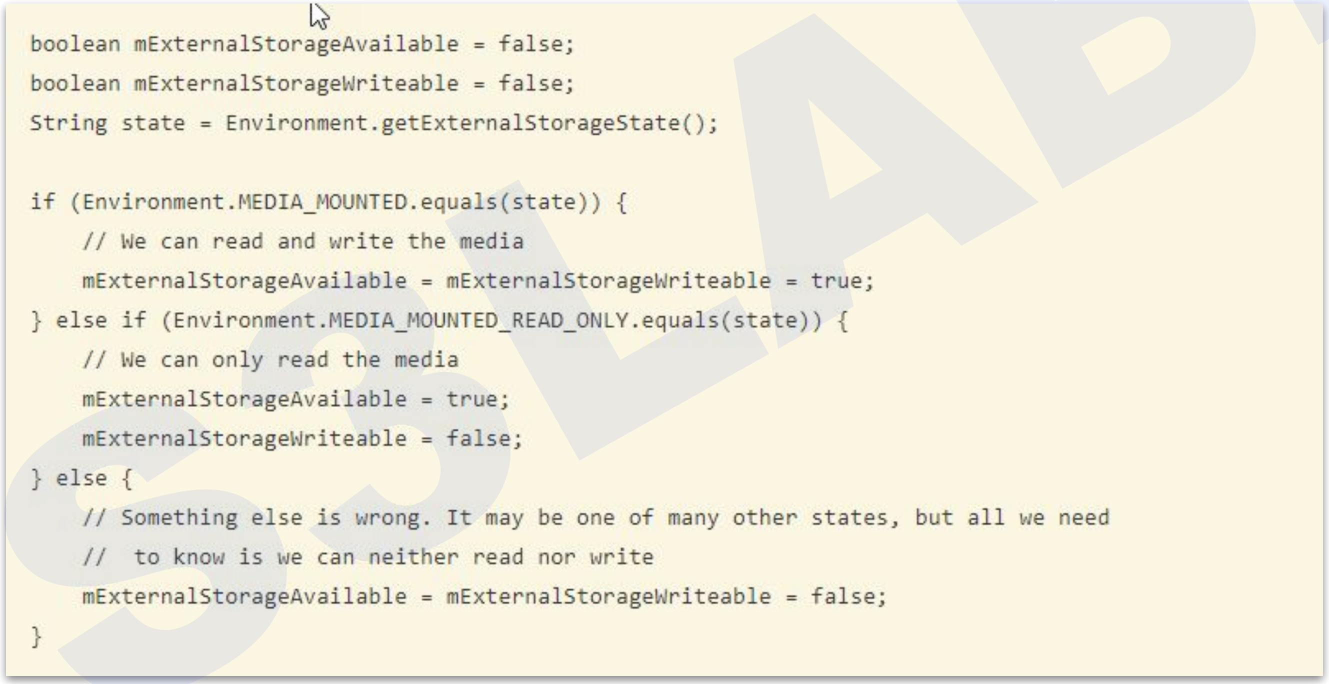
Data Storages

External Storage

- Must check whether external storage is available first by calling **getExternalStorageState()**
  - Also check whether it allows read/write before reading/writing on it
- **getExternalFilesDir()** takes a parameter such as DIRECTORY_MUSIC,DIRECTORY_RINGTONE etc,to open specific type of subdirectories.
- For **public** shared directories,use **getExternalStoragePublicDirectory()**

Data Storages

External Storage



```
boolean mExternalStorageAvailable = false;
boolean mExternalStorageWriteable = false;
String state = Environment.getExternalStorageState();

if (Environment.MEDIA_MOUNTED.equals(state)) {
    // We can read and write the media
    mExternalStorageAvailable = mExternalStorageWriteable = true;
} else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
    // We can only read the media
    mExternalStorageAvailable = true;
    mExternalStorageWriteable = false;
} else {
    // Something else is wrong. It may be one of many other states, but all we need
    //  to know is we can neither read nor write
    mExternalStorageAvailable = mExternalStorageWriteable = false;
}
```

Data Storages

External Storage

- For cache files, use **getExternalCacheDir()**
- All these are applicable for API level 8 or above
- For API level 7 or below ,use the method;
    - **getExternalStorageDirectory()**
    - Private files stored in //Android/data/<package name>/files/
    - Cache files stored in //Android/data/<package name>/cache/

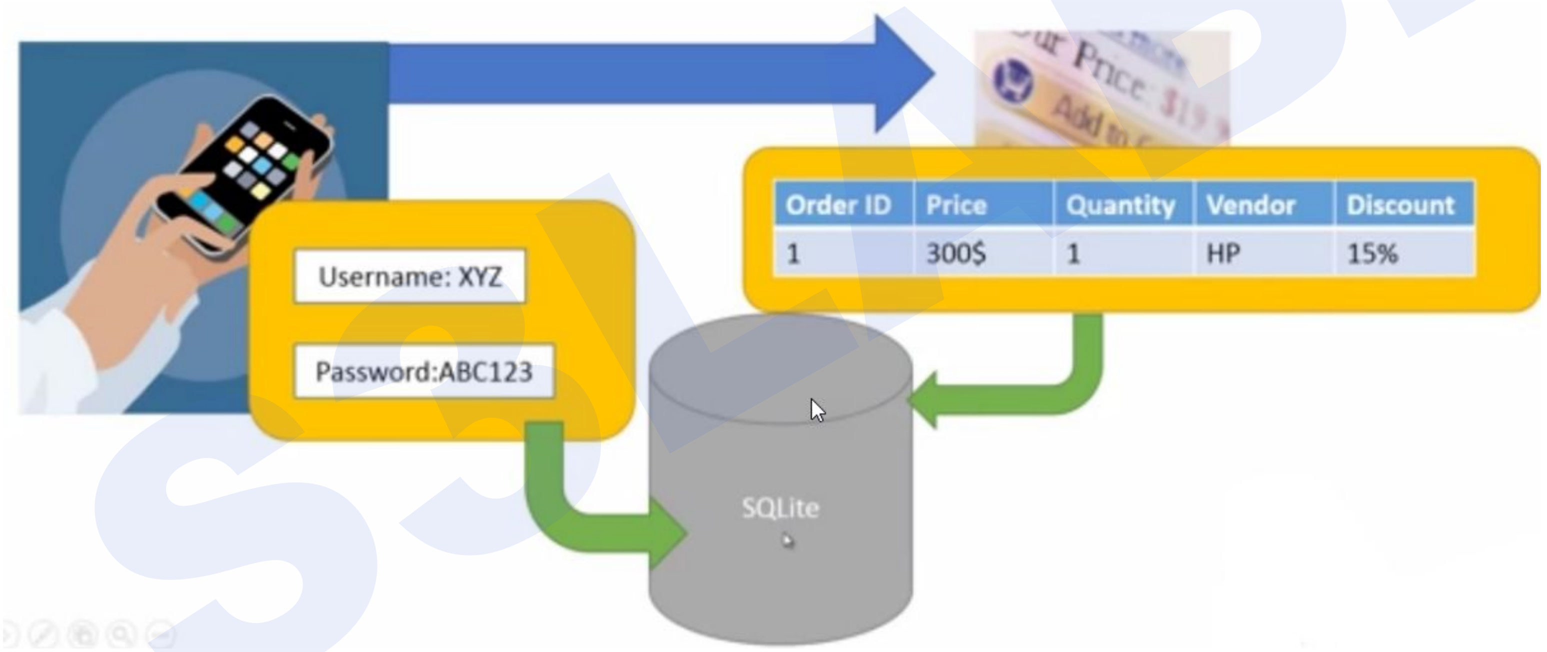
Data Storages

External Storage

- For Hiding your files from the Media Scanner
  - Add an empty file named **.nomedia** in your external files directory
- Saving files that should be shared
  - If you want to save files that are not specific to your application and that should not be deleted when your application is uninstalled, save them to one of the public directories on the external storage. These directories lay at the root of the external storage, such as **Music**/, **Pictures**/, **Ringtones**/, and others**.**

Data Storages

SQLite Database



Order ID	Price	Quantity	Vendor	Discount
1	300$	1	HP	15%

Data Storages

SQLite Database

- **android.database.sqlite** Contains the SQLite database management classes that an application would use to manage its own private database.
- **android.database.sqlite.SQLiteDatabase** Contains the methods for: creating, opening, closing, inserting, updating, deleting and querying an SQLite database.

Data Storages

SQLite Database - android.database.sqlite

- **SQLiteCloseable** - An object created from a SQLiteDatabase that can be closed.
- **SQLiteCursor** - A Cursor implementation that exposes results from a query on a SQLiteDatabase.
- **SQLiteDatabase** - Exposes methods to manage a SQLite database.
- **SQLiteOpenHelper** - A helper class to manage database creation and version management.
- **SQLiteProgram** - A base class for compiled SQLite programs.
- **SQLiteQuery** - A SQLite program that represents a query that reads the resulting rows into a CursorWindow.
- **SQLiteQueryBuilder** - a convenience class that helps build SQL queries to be sent to SQLiteDatabase objects.
- **SQLiteStatement** - A pre-compiled statement against a SQLiteDatabase that can be reused.

Data Storages

SQLite Database - OpenOrCreateDatabase

- This method will open an existing database or create one in the application data area
  - import android.database.sqlite.SQLiteDatabase;
  - SQLiteDatabase myDatabase;

    myDatabase = openOrCreateDatabase ("my_sqlite_database.db" , SQLiteDatabase.CREATE_IF_NECESSARY , null);

Data Storages

SQLite Database - Creating Tables

- Create a static string containing the SQLite CREATE statement, use the execSQL( ) method to execute it.
    - String createAuthor = "CREAT TABLE authors (id INTEGER PRIMARY KEY AUTOINCREMENT, fname TEXT, lname TEXT);
    - myDatabase.execSQL(createAuthor);

Data Storages

SQLite Database - insert( ), delete( )

- long insert(String table, String nullColumnHack, ContentValues values)
  - import android.content.ContentValues;
  - ContentValues values = new ContentValues( );
  - values.put("firstname" , "J.K.");
  - values.put("lastname" , "Rowling");
  - long newAuthorID = myDatabase.insert("tbl_authors" , "" , values);
- int delete(String table, String whereClause, String[] whereArgs)

  -
    ```
    public void deleteBook(Integer bookId) {
            myDatabase.delete("tbl_books" , "id=?" , new String[ ] { bookId.toString( ) } ) ;
    }
    ```

Data Storages

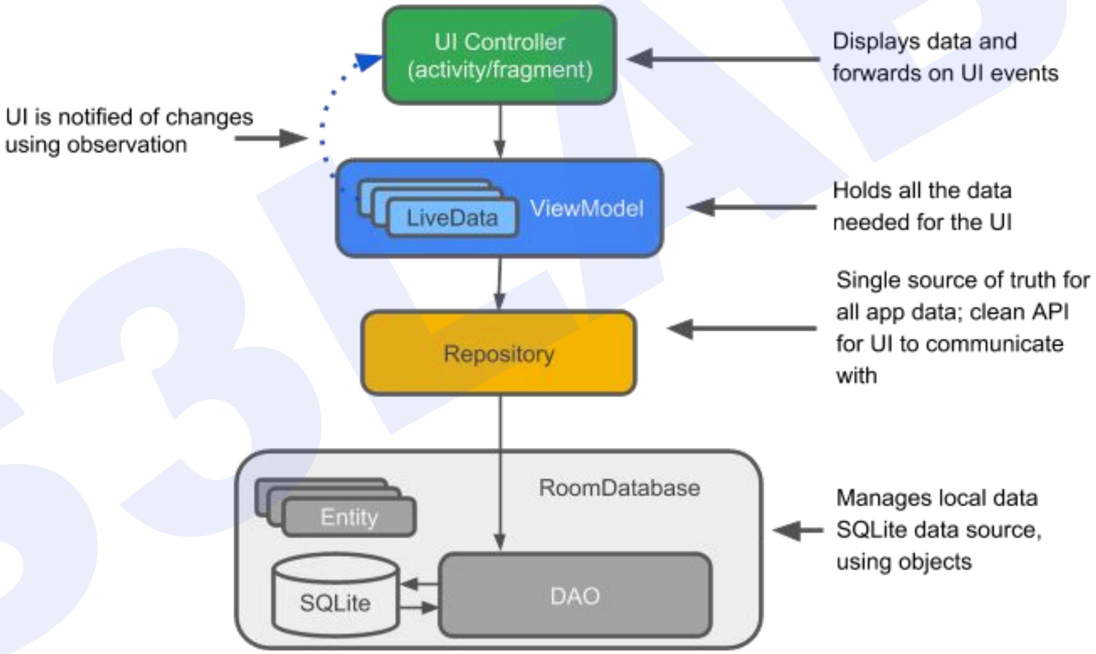
SQLite Database - Update()

- int update(String table, ContentValues values, String whereClause, String[ ] whereArgs)

  -
    ```
    public void updateBookTitle(Integer bookId, String newTitle) {
            ContentValues values = new ContentValues();
            values.put("title" , newTitle);
            myDatabase.update("tbl_books" , values , "id=?" , new String[ ] {bookId.toString() } );
    }
    ```

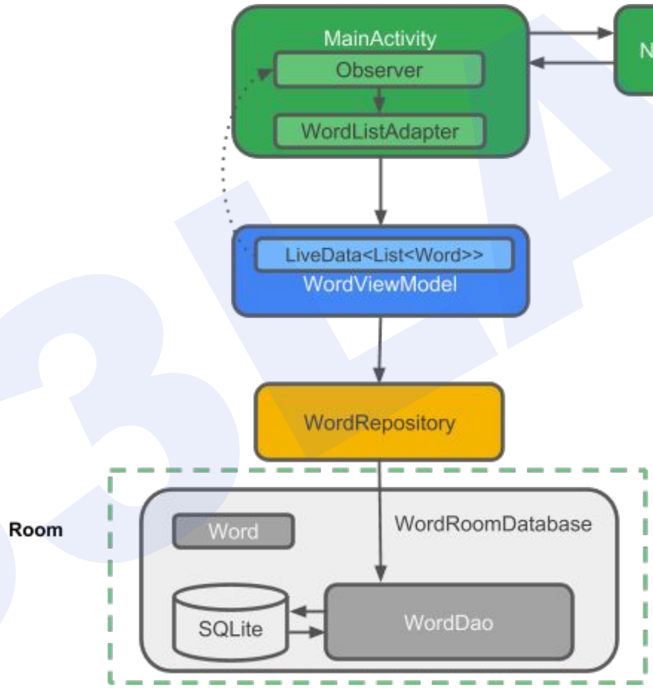
Data Storages

*SQLite Database - With Architecture **Component Room**, **LiveData** and **ViewModel***


UI Controller
(activity/fragment)
Displays data and forwards on UI events
UI is notified of changes using observation
LiveData
ViewModel
Holds all the data needed for the UI
Repository
Single source of truth for all app data; clean API for UI to communicate with
RoomDatabase
Entity
SQLite
DAO
Manages local data SQLite data source, using objects

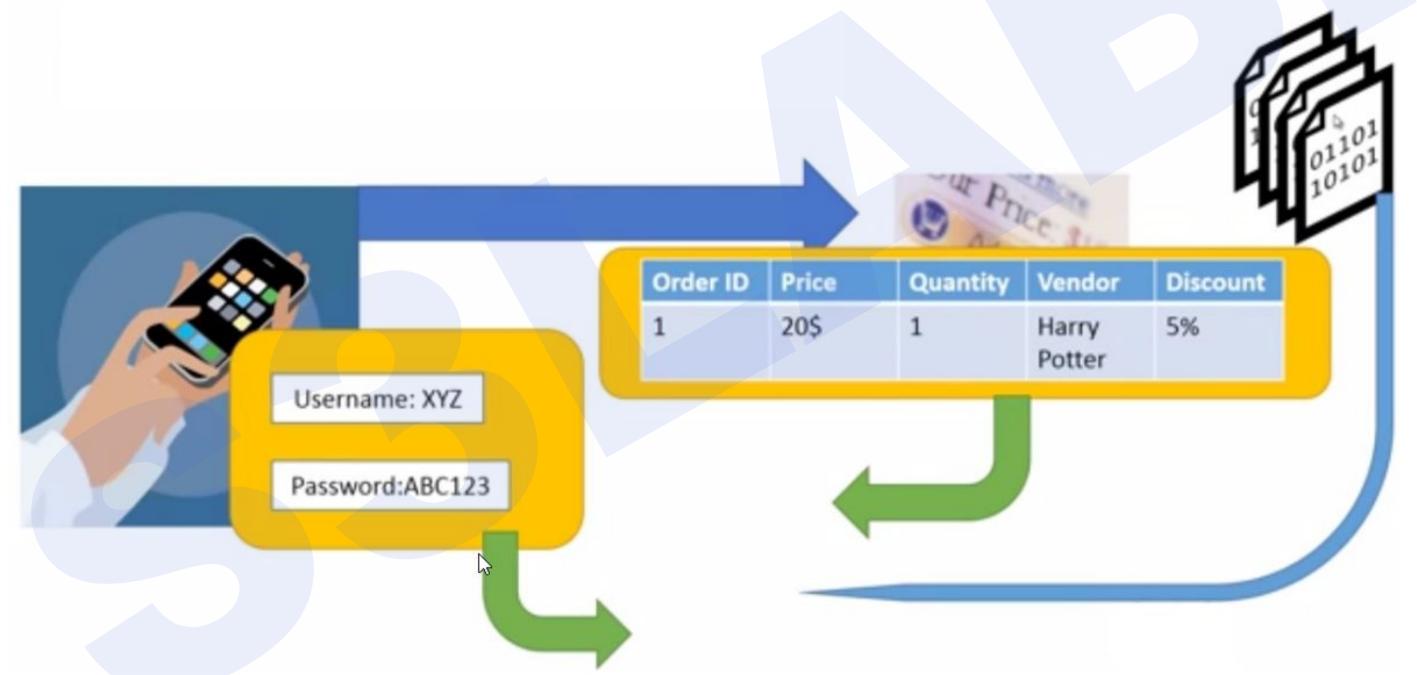
Data Storages

*SQLite Database - With Architecture **Component Room**, **LiveData** and **ViewModel***



Data Storages

Content Provider

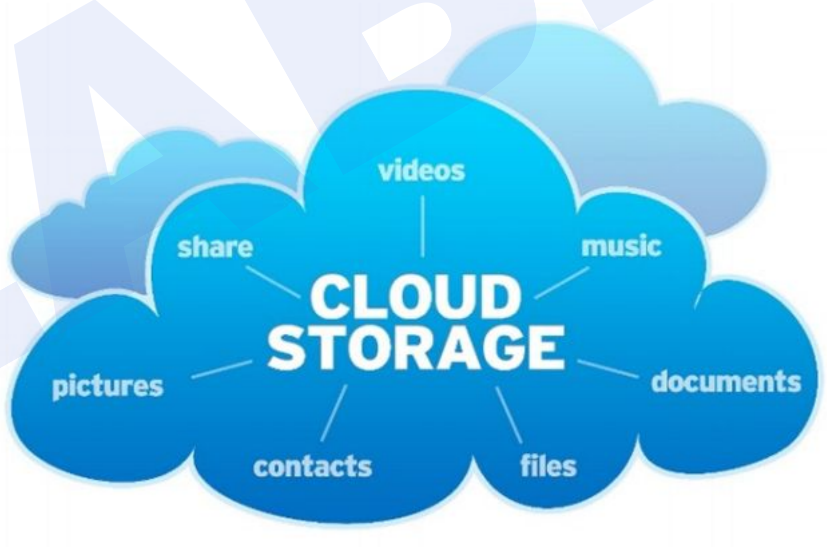


Order ID	Price	Quantity	Vendor	Discount
1	20$	1	Harry Potter	5%

Data Storages

Cloud Storage

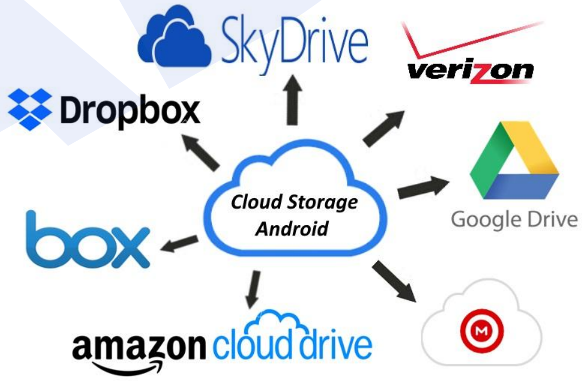
- Online file storage centres or cloud storage providers allow you to safely upload your files to the Internet.



Data Storages

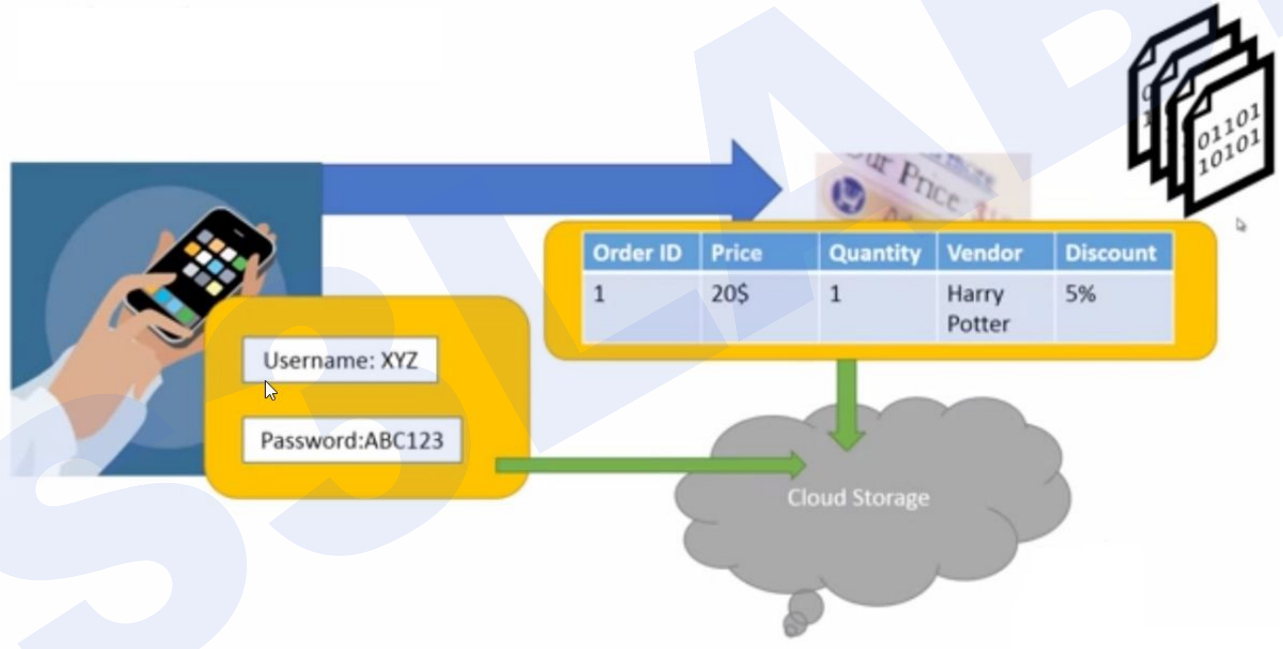
Cloud Storage

- There are various providers of cloud storage:
    - Apple iCloud (Gives 5GB of free storage)
    - Dropbox (Gives 2GB of free storage)
    - Google Drive (Gives 15GB of free storage)
    - Amazon Cloud Drive (Gives 5GB of free storage)
    - Microsoft SkyDrive(Gives 7GB of free storage)



Data Storages

Cloud Storage



Order ID	Price	Quantity	Vendor	Discount
1	20$	1	Harry Potter	5%

Homeworks

- Update one of your homework by using above storage methods
- References
  - https://codelabs.developers.google.com/codelabs/android-training-shared-preferences
  - https://codelabs.developers.google.com/codelabs/android-training-adding-settings-to-app
  - https://developer.android.com/training/data-storage/room
  - https://codelabs.developers.google.com/codelabs/android-training-livedata-viewmodel
  - https://codelabs.developers.google.com/codelabs/android-training-room-delete-data

Q & A

Thank you for listening

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