

## **UNIT VI: Working with Database and Publishing the Apps**

6.1 Introduction to SQLite, SQLiteOpenHelper and SQLiteDatabase

6.2 Creating, opening and closing database,

6.3 Working with cursors, Insert, Update, Delete, Building and executing queries

6.4 Preparing for publishing the App, Publishing to the Play Store.

## 6.1 Introduction to SQLite

SQLite is a opensource SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC,ODBC e.t.c

### Database - Package

The main package is android.database.sqlite that contains the classes to manage your own databases

### Database - Creation

In order to create a database you just need to call this method openOrCreateDatabase with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object

```
SQLiteDatabase mydatabase = openOrCreateDatabase("your database name",MODE_PRIVATE,null);
```

### Database - Insertion

we can create table or insert data into table using execSQL method defined in SQLiteDatabase class.

```
mydatabase.execSQL("CREATE TABLE IF NOT EXISTS user(Username VARCHAR,Password VARCHAR);");  
mydatabase.execSQL("INSERT INTO user VALUES('admin','admin');");
```

### Database - Fetching

We can retrieve anything from database using an object of the Cursor class. We will call a method of this class called.rawQuery and it will return a resultset with the cursor pointing to the table. We can move the cursor forward and retrieve the data.

```
Cursor resultSet = mydatabase.rawQuery("Select * from user",null);  
resultSet.moveToFirst();  
String username = resultSet.getString(0);  
String password = resultSet.getString(1);
```

There are other functions available in the Cursor class that allows us to effectively retrieve the data.

### **1 getColumnCount()**

This method return the total number of columns of the table.

### **2 getColumnIndex(String columnName)**

This method returns the index number of a column by specifying the name of the column

### **3 getColumnName(int columnIndex)**

This method returns the name of the column by specifying the index of the column

### **4 getColumnNames()**

This method returns the array of all the column names of the table.

### **5 getCount()**

This method returns the total number of rows in the cursor

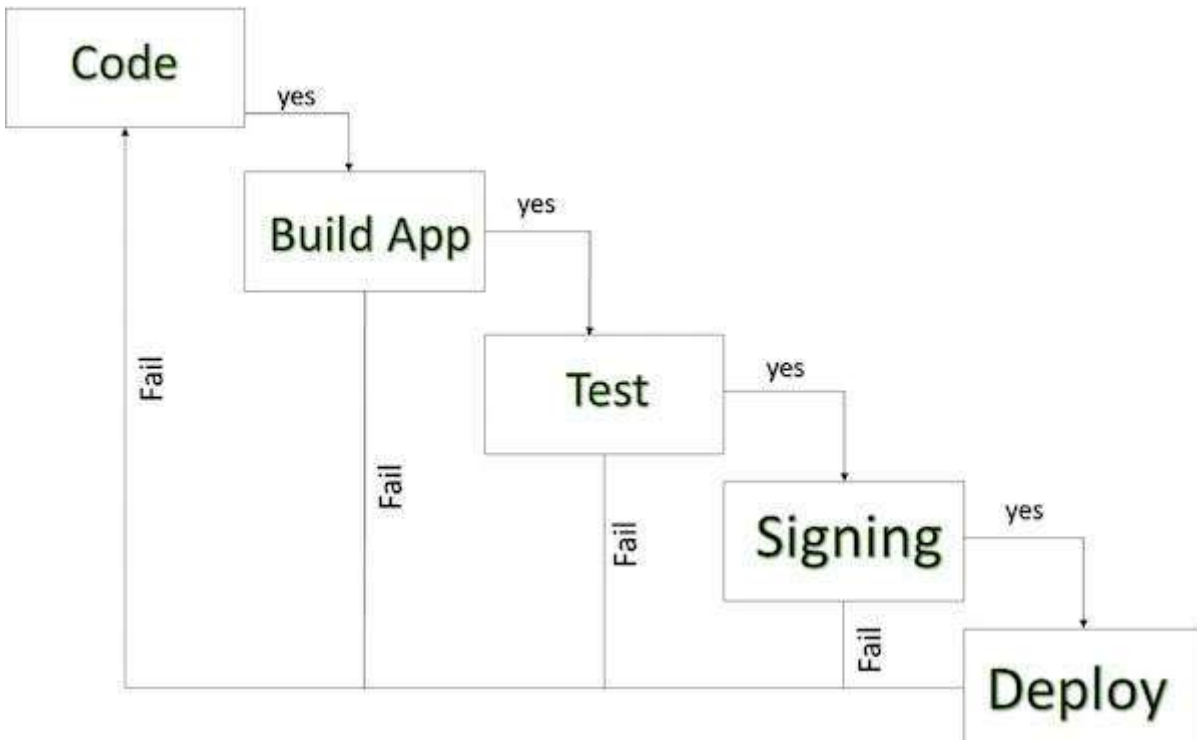
### **6 getPosition()**

This method returns the current position of the cursor in the table

## 6.4 Preparing for publishing the App

Android application publishing is a process that makes your Android applications available to users.

publishing is the last phase of the Android application development process.



Once you developed and fully tested your Android Application, you can start selling or distributing free using Google Play

You can also release your applications by sending them directly to users or by letting users download them from your own website.

**Regression Testing** Before you publish your application, you need to make sure that its meeting the basic quality expectations for all Android apps, on all of the devices that you are targeting. So perform all the required testing on different devices including phone and tablets.

**Application Rating** When you will publish your application at Google Play, you will have to specify a content rating for your app, which informs Google Play users of its maturity level. Currently available ratings are (a) Everyone (b) Low maturity (c) Medium maturity (d) High maturity.

**Application Size** Currently, the maximum size for an APK published on Google Play is 50 MB. If your app exceeds that size, or if you want to offer a secondary

download, you can use APK Expansion Files, which Google Play will host for free on its server infrastructure and automatically handle the download to devices.

**SDK and Screen Compatibility** It is important to make sure that your app is designed to run properly on the Android platform versions and device screen sizes that you want to target.

**Application Pricing** Deciding whether your app will be free or paid is important because, on Google Play, free app's must remain free. If you want to sell your application then you will have to specify its price in different currencies.

**Build and Upload release-ready APK** The release-ready APK is what you will upload to the Developer Console and distribute to users.

### **Publishing to the Play Store.**

The most important step is to register with Google Play using Google Play Marketplace. You can use your existing google ID if you have any otherwise you can create a new Google ID and then register with the marketplace.

You can use Continue to payment button to proceed to make a payment of \$25 as a registration fee and finally to complete your account detail.

Once you are a registered user at Google Play, you can upload release-ready APK

**Thank You**