Technical Specification for Creating Apps in Android

Kauser Hameed, Manal Elobaid

Department of Computer Science, CCSIT, King Faisal University, Hofuf, KSA
manalobaid@yahoo.com

Abstract: As it has been observed, people are adopting latest technologies to access their favorite apps to enhance their knowledge in their busy and spare time. The aim of this research work is to develop a guidance framework that can help programmers to develop Quran Apps using Android technologies. The paper aspire to specify all the significant steps in initiating a “Quran app" development process, starting from setting up a device with Android SDK, Android Tools and other technical software and hardware necessities to finally launching it in Play Store for public access. This paper will narrate the configuring process involved in Integrated Development Environment with Android SDK Tools required for coding using Eclipse, and developing the App’s backend with the SQLite. Various APIs will be explored to create user interfaces, activities, and services based on the UML diagrams and Use Cases. Likewise, converting the ER diagrams (logical database) into physical database using SQLite format and further connecting the UI with these databases will also be discussed. In the meantime, Sighting authentic databases in various formats and clarifies sources will also be enlightened. Moreover, the paper will cover the process of hosting database in an App Server along with other app resources. Testing condition that can be carried out in parallel at different incremental development stages in android device and in emulator will also be focused. Finally the paper will wrap up with the best practices used to launch the App in Google play store to make it available to the world and by other different supported means.

[Kauser Hameed, Manal Elobaid. **Technical Specification for Creating Apps in Android.** *Rep Opinion* 2014;6(7):1-6]. (ISSN: 1553-9873). http://www.sciencepub.net/report. 1

Keywords: Mobile Applications, Android, Quran Applications, UML diagrams, Quran Database

Introduction:

This paper aims to illustrate the technical procedure involved in developing “Al-Zikr mobile app for android devices”. The purpose of this paper is to help any individual interested in creating such application with basic programming knowledge. As the idea being to proliferate the ease of developing Digital Holy Quran (using authentic sources) to run on latest smartphone technologies the *Android* platform has been selected. As stated android is winning the global smartphone war being fought everywhere, smartphone sales grew 38 % last quarter to reach 217 million units worldwide, and over 700 million units for the entire year, according to a new report from Strategy Analytics of those 700 million-plus smartphones, 68.4% smartphones ran Android as the operating system, while only 19.4 % ran iOS, Apple’s mobile operating system. The firm’s executive director Neil Mawston said in a statement. “Android is clearly the undisputed volume leader of the smartphone industry at the present time”[1]. Experts predict that there could be more smartphones on the planet than humans by the end of this year, and by 2016 there could be 10 billion smartphones that is 1.4 mobile devices per capita [2]. However, techniques presented in this paper could serve generally as the basis to support other mobile operating systems and different devices.

People are adopting latest technologies to access their favorite app to enhance their knowledge in their busy and spare time. In certain situation, there is a need of defining a prevailing procedure to devise the logic and help mobile programmers in developing Quran Apps that could cater to the growing needs of such.

1. Setting the device ready for android development:

1.1 **Minimum Software and Hardware Requirements:** The necessary things required to initiate Android apps development and testing are as follows:

1.1.1 Hardware Requirement:

i. Computer device with latest configuration for development

ii. Android compatible smart phone for testing

Software Requirement:

i. Java Runtime Environment

ii. Android Development Tool Kit

Java Runtime Environment can be installed for free from its official website[3]

The Android Development Tool Kit (ADT) provided by Google in the Android developers website comprises of all the necessary tools including the Integrated Development Environment (IDE) to create, test and debug apps that can run on any android compatible devices. ADT Bundle consist of the following, that can be downloaded from android official developer website [4].

i. Eclipse + ADT plugin

ii. Android SDK Tools

iii. Android Platform-tools

iv. The latest Android platform

v. The latest Android system image for the emulator

A new IDE called Android Studio, based on IntelliJ IDEA, is also available as early access preview that can be used instead of eclipse IDE[5].

2. Getting Started with App Development :

The three major stages that comprise all the steps required to initiate app development and app delivery in the Google play store, are listed below:

Design Phase

i. Development Phase

ii. Distribute

2.1 **Design Phase:** Designing the User Interface is the crucial thing in defining the scope of the app in terms of its features, functionality and user friendly interactions with its audience. A Quran app should be designed with attractive and interactive screens describing its major and minor features such as its recitation and repeat options in recitation, translation and changing translation options, explanation and root words. The Design phase can include the following techniques.

2.1.1 **Designing the UML:** All the possible app interactions must be keenly observed and illustrated in the form of use cases that can later be converted into app screen, such as launching the app and selecting a particular verse for listening the recitation or viewing its translation and explanation or sharing it etc. Similarly many use cases should be designed to describe each and every App/User interaction. The figure 1.0 depicts the basic functionality of Quran app.

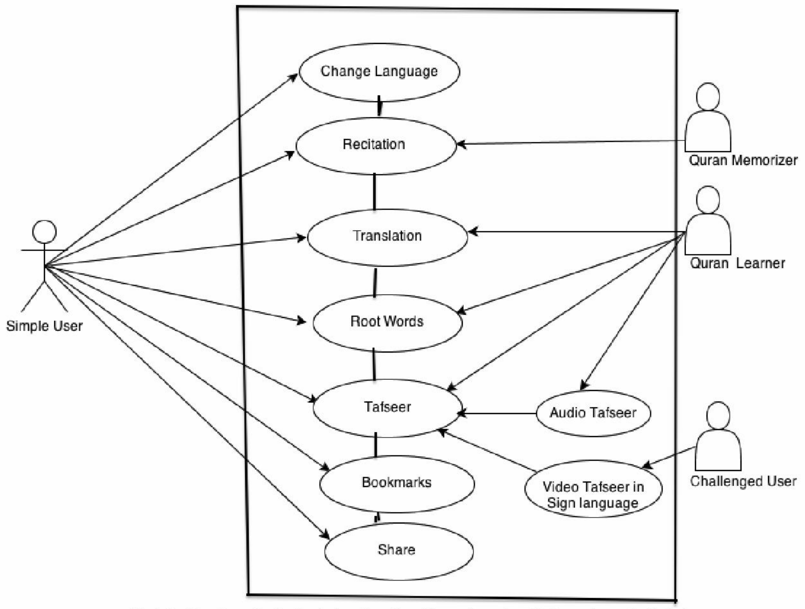

Fig 1.0 : Use Case for the basic functionality of Quran Learning Software for android devices

2.1.2 **Designing User Interface in Photoshop:** A mobile app designer can then convert all the use cases in the form of screens or app features and resize them to suite the display of various mobile screen sizes. The design must be simple, smart and powerful tailored to suit different Android devices and versions. The design guidelines[6] provided by Google developer website provides quick insight into the Design section that every android app designer must look at. A good android app must support its UI to accommodate different android device sizes, versions and languages to reach large number of audience, the below illustration shows few sample screens extracted from the UML listed in Fig 1.0, care must be taken to resize image resources in basic four screen sizes and to support different android versions and locales.

2.2 **Development Phase:** Once the miniature model of the app is designed in UML and converted into draft screens as listed in figure 1.1 in Photoshop, an android programmer can start creating the actual screens using android APIs. The development phase will include the following steps:

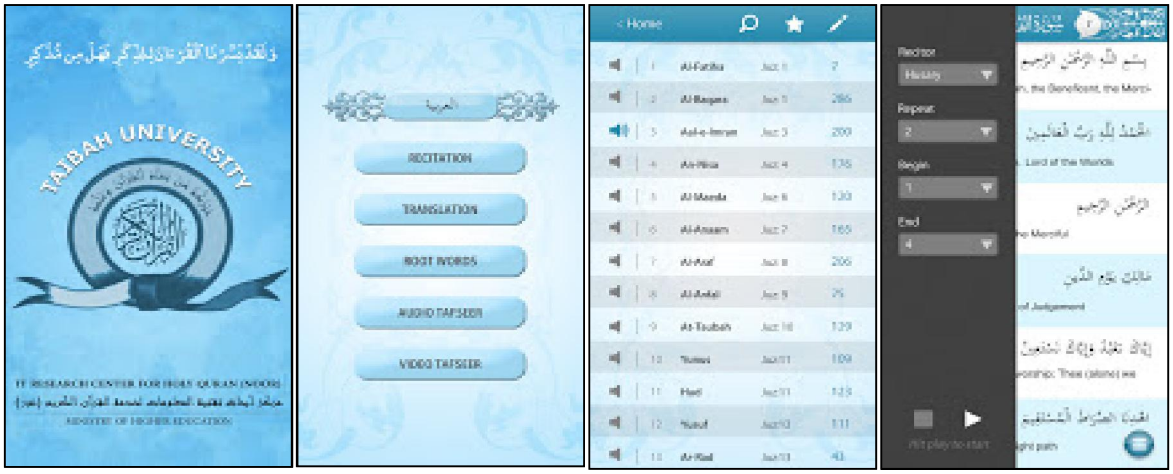

Figure 1.1 Sample Screens designed in Photoshop

2.2.1 **Creating User Interface in Android and Exploring Various APIs:** Android provides access to a wide range of useful libraries and tools that can be used to build rich applications[7] .Keeping each draft screen in view the programmer must create the layout with its UI components either in XML or in the code itself, Listing 1.2 illustrates a sample code to create the home screens with the major App features in XML:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".TranslationListView_lang" >
  <ImageView
    android:id="@+id/imageView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:fitsSystemWindows="true"
    android:scaleType="fitXY"
    android:src="@drawable/background" />
  <ToggleButton
    android:id="@+id/toggleButton1"
    android:layout_width="250sp"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="30dp"
    android:background="@drawable/btntoggle_selector"
    android:textOff="عـــــربي "
    android:textOn="English " />
  <Button
    android:id="@+id/recitation"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="200sp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/toggleButton1"
    android:layout_centerHorizontal="true"
    android:background="@drawable/btn_feature"
    android:text="@string/home_recitation" />
 // Similar Button code for translation, root_words, video_tafseer, audio_tafseer shououd be added here
 </RelativeLayout>
```

Listing 1.2: XML code for creating Home screen with Six buttons to show major features of the app.

The package *android.widget* contains all the classes and methods to create and manipulate any UI widget (UI component in android).

2.2.2 **Creating App databases in SQLite format using SQLite professional and storing it :** Post designing the UI the Quran database schema must be created with all the required tables, as the mobile apps uses SQLite database, a tool called as SQLite professional can be very useful, the App database will contain many tables depending on the features employed in the app. A sample table structure is shown in fig1.3 that can be used to store Quran chapters and verses, like wise other tables must also be created:

```
CREATE TABLE "Verse_Text" (
Index integer(4) NOT NULL,
SurahID integer(3) NOT NULL DEFAULT('0'),
VerseID integer(3) NOT NULL DEFAULT('0'),
VerseText varchar2(1000) NOT NULL,
PRIMARY KEY(Index)
);
```

```
CREATE TABLE Reciter (
ID integer NOT NULL,
ReciterName  varchar2(40),
PRIMARY KEY(ID)
);
```

Various online resources are available that provide Quran database in many formats, however care must be taken to use the data from accurate sources only.

2.2.3 Connecting User Interface with Database using SQLite API: Once the database structure is ready then the programmer should link the User interface with backend using packages provided by Android system such as:

1. android.database.Cursor;
2. android.database.SQLException;
3. android.database.sqlite.SQLiteDatabase;
4. android.database.sqlite.SQLiteException;

Listing 1.3: Shows a sample code to link the database with in the app

```
public class DataBaseHelper extends SQLiteOpenHelper
{
private static String TAG = "DataBaseHelper"; // Tag just for the LogCat window
//destination path (location) of our database on device
private static String DB_PATH = "";
private static String DB_NAME ="MyQuran.sqlite";// Database name
private SQLiteDatabase mDataBase;
private final Context mContext;
public static String table = "Surah";
public DataBaseHelper(Context context)
{
   super(context, DB_NAME, null, 7);//  Database Version
   DB_PATH = "/data/data/" + context.getPackageName() + "/databases/";
   this.mContext = context;
}
public void createDataBase() throws IOException
{
   //If database not exists copy it from the assets
   boolean mDataBaseExist = checkDataBase();
   if(!mDataBaseExist)
   {
      this.getReadableDatabase();
      this.close();
      try
      {
         //Copy the database from assests
         copyDataBase();
         Log.e(TAG, "createDatabase database created");
      }
      catch (IOException mIOException)
      {
            throw new Error("ErrorCopyingDataBase");
        }
    }
}
    //Check that the database exists here: /data/data/your package/databases/ the Name
    private boolean checkDataBase()
    {
        File dbFile = new File(DB_PATH + DB_NAME);
        //Log.v("dbFile", dbFile + "   "+ dbFile.exists());
        return dbFile.exists();
    }
    //Copy the database from assets
    private void copyDataBase() throws IOException
    {
        InputStream mInput = mContext.getAssets().open(DB_NAME);
        String outFileName = DB_PATH + DB_NAME;
        OutputStream mOutput = new FileOutputStream(outFileName);
        byte[] mBuffer = new byte[1024];
        int mLength;
        while ((mLength = mInput.read(mBuffer))>0)
        {
            mOutput.write(mBuffer, 0, mLength);
        }
        mOutput.flush();
        mOutput.close();
        mInput.close();
    }
    //Open the database, so we can query it
    public boolean openDataBase() throws SQLException
    {
        String mPath = DB_PATH + DB_NAME;
        //Log.v("mPath", mPath);
        mDataBase = SQLiteDatabase.openDatabase(mPath, null, SQLiteDatabase.CREATE_IF_NECESSARY);
        //mDataBase = SQLiteDatabase.openDatabase(mPath, null,
SQLiteDatabase.NO_LOCALIZED_COLLATORS);
        return mDataBase != null;
    }
    @Override
    public synchronized void close()
    {
        if(mDataBase != null)
            mDataBase.close();
        super.close();
    }
          @Override
          public void onCreate(SQLiteDatabase arg0) {
              // TODO Auto-generated method stub
          }
          @Override
          public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
              // TODO Auto-generated method stub
          }
}
```

Listing 1.3: Sample code to link the database with in the app.

After linking the database, all its table can be accessed from any screen or user interface component of the app using simple SQLite Queries[8]. Database can be stored in one of the following ways:

i. Creating the database in the code by extending SQLiteOpenHelper

ii. Storing the preloaded database in asset folder and then copying it in the application internal memory

iii. Storing the database file (MyQuran.Sqlite) in the webserver an then copying it in the application

2.2.4. Testing

Testing/ debugging is carried out simultaneously during the development phase, emulator provided in the form of Android Virtual Devices (AVD) can be used to test the functioning of the app on the computer itself, however testing on at least one physical device is recommended (one mobile and one tablet)

3. Launching the App

Once the app is tested in various environments (i.e., different AVDs & Physical devices) the app can be distributed in one of the three ways:

i. By sending the .apk to the audience as the email attachment, for which they need to open the attachment in the android compatible device and they should install.

ii. By uploading the downloading app in website.

iii. By launching the app in the Google play.

3.1 launching the app in the Google play:

In order to launch the app in Google play, one must have a developer account in Google account. The developer website provide the checklist [9] that explains all the necessary steps to upload an app in Google play.

4. Conclusion:

The procedures discussed above can help a programmer to create a Quran App in android using the best practices to cater to the growing needs of mobile users. However deferent methods and ways are available to reach the same point, depending in the programmer background and the available Tools he prefer.

7/11/2014

5. Acknowledgements:

NRC Project (No.: **NRC1-123**). Authors are grateful to IT Research Center for the Holy Quran (Noor), Taibah University, for financial support to carry out this work.

Refrences

1. http://venturebeat.com/2013/01/28/android-captured-almost-70-global-smartphone-market-share-in-2012-apple-just-under-20
2. http://mashable.com/2012/02/14/more-smartphones-than- humans
3. http://www.oracle.com/technetwork/java/javase/downloads/jre6downloads-1902815.html
4. http://developer.android.com/sdk/index.html#download
5. http://developer.android.com/sdk/installing/studio.html
6. http://developer.android.com/design/index.html
7. http://developer.android.com/about/versions/android-2.3.4.html
8. h.Schneidawind, J: "Big Blue unveiling", *USA Today*, November 23, 1992, p. 2B
9. Kevin McLaughlin (December 17, 2009). "BlackBerry Users Call For RIM To Rethink Service". CRN.com.
10. http://www.crn.com/news/client-devices/222002587/blackberry-users-call-for-rim-to-rethink-service.htm. Retrieved 2011-12-15
11. iPhone 4S Pre-Orders Top One Million in First 24 Hours". Apple. http://www.apple.com/pr/library/2011/10/10iPhone-4S-Pre-Orders-Top-One-Million-in-First-24-Hours.html. Retrieved 10 October 2011.
12. Anderson, Ash. "iPhone 4S Sells 1 Million in Under 24 Hours". *KeyNoodle*. http://www.keynoodle.com/iphone-4s-sells-1-million-in-under-24-hours/. Retrieved 2011-12-15.