

# A Mobile App Development Framework for Database Connectivity and High Level Services

Vivek Sthanam, Rama Krishna Raju Rudraraju, Venkatadri Raparla

Dept. of Electrical Engineering, University of Alabama at Birmingham

Email: {sthanamv, rkr2016, vr1221} @uab.edu

**Abstract-** Frameworks have been successfully implemented to simplify, in part, the software development process. There are many frameworks developed for android platforms, normally targeted towards specific tasks such as login, database connectivity etc. Many mobile apps in use today make use of databases to implement user-level services within the app. For apps requiring limited amount of data to be managed, the data is typically managed on the device itself where the app resides. However, for apps requiring access to large databases, the database management system is often installed remotely rather than on the device, and accessed via the network. Implementing apps that require interaction with remote databases is tedious, error prone, and often requires using a specific protocol related to the database management system.

## 1. INTRODUCTION

The stipulation of mobile app development in the present rising market arouse multitudinous pristine frameworks. These frameworks favoring the development by reducing loads of repetitive code and providing support for building of something by expanding the structure into usefulness. There are two discrete types of frameworks that will normally serve for accomplishment of the task. The first is theoretical framework, is a study based on an existing theory or theories. The second is conceptual framework, on the other hand, is something you can develop yourself based on this theory. Numerous frameworks provided pavement for developing related apps in a flexible mode. While providing frameworks for app development it is much important of providing feasibility with working databases. As workability with database is crucial in most of apps that are delivering today, the sight of proposing framework is for database connectivity and intended services through it.

In this research, we introduce a novel approach to implement a framework for mobile app

development based on a two-layer approach. At the lower layer, we define a database abstraction that allow developers to abstractly connect to any remote database using the concept of a virtual machine. At the higher layer, we define a set of useful services that transparently use the database to implement these services.

The lower layer of the framework which encapsulates the database access and interaction can be used independently from the higher layer to satisfy most database needs of an app, including the basic create, read, update, and delete operations. The abstract interface of this layer reduces the amount of coding that developers can work on implementing database connectivity within their apps. In general, the implementation of database connectivity takes considerable amount of time to code and requires knowledge of the underlying database being used.

The primary motivation for frameworks varies but for dispensing precise features. Even though every framework is having their in-built database to work on, the interacting pattern is different. Many existing frameworks such as Basic4Android, Framework 7, Fat Free, and Play framework provide a plug-in for managing JDBC connection pools [1] to access data from external database, so interacting with different databases follow different methods. The proposed framework aims to provide an android package in which the connection pattern remains same to interact with any sort of databases through JDBC connectivity.

## 2. A SURVEY OF DIFFERENT FRAMEWORKS

A framework is a real or conceptual structure intended to serve as a support or guide for building of something that expands the structure into something useful. Frameworks are categorized into different types even in computer terminology based on the purpose. Some of them are enterprise

architecture framework, software framework, web application framework, framework oriented design, etc. Most often a framework is a layered architecture in which it indicates what kind of programs can be built and how would they interrelate. Generally, frameworks contain some distinguishing features like default behavior, extensibility and non-modifiable framework code. The main aim of frameworks is to increase the productivity by reducing efforts which eventually saves a lot of time for developers to resolve other issues.

There are few selective frameworks provided contribution in performing database connectivity. Basic4Android is one those selective frameworks which holds providence to connect with different database apart from SQLITE [2]. Connectivity can be achieved by creating the helper classes for the respective databases. But the vicinity of this framework is limited because with this cardinal level of app development can be performed. Framework7 provides the connectivity with SQL SERVER [3] by default. It is an entity framework suitably for developing the apps using C# [4] i.e. on Visual Studio. Making a connection with other databases hardly possible using this framework. Spring for android is a part of services offered by Spring Framework that is solely for android mobile app development. Although possessing SQLITE as in built one it needs external driver to connect with other databases. Play and Fat free frameworks are more flexible to work on compared to the above frameworks.

Play framework [5] as a part of services providing plugin for managing JDBC connection pools. So, it makes a flexible way for interacting with any database easily. Fat free framework [6] seems to be same as play framework coming to the functionality. But in general, its handling is so easy compared to play framework. As its name, it handles very light code and so very easy to manage. Even all these frameworks offering connection with different databases through JDBC or in any other way flexibility of connection with databases comes into the matter over here.

The proposed framework aims to fulfill few things that lack in the frameworks mentioned above. Even though few frameworks provide portability to connect with different databases, the methodology made them complicated. For making the process simply, an android package which provides

accessibility to connect any database can be implemented with this framework. Generally, the connection to database can be done in two ways in which the first way is through REST services and the second way through JDBC. Within the android package there will be helper classes and their implementation methods of different databases which is extended by DB Helper class. This will make the developers to perform their jobs easy.

### 3. PROPOSED SYSTEM AND ARCHITECTURE

The proposed system is offering an Android Framework which provides workability in two layers shown in fig 3-1. The bottom layer is primarily for database connectivity, which indeed offers services in the top layer. The bottom layer consists of JDBC, Android SDK [7], and different databases such as MYSQL, SQLSERVER, ORACLE, and SQLITE. The in-built database for Android Studio is SQLITE. But the main thing in working with the apps for developers is flexibility, scalability, and reusability. If you come across flexibility different developers are intended to work with different databases. So, we need connection drivers to handle different databases with Android Studio. JDBC acts as a connectivity driver for databases such as MYSQL, SQLSERVER, and ORACLE to collaborate with Android Studio to perform the required task.

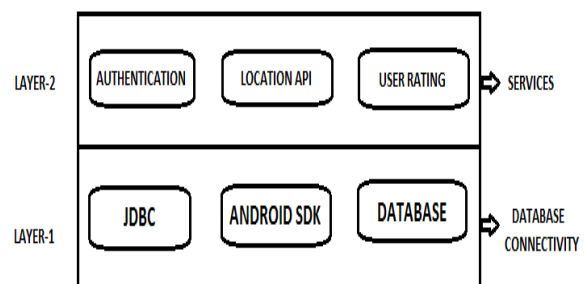


Figure 3-1 Architecture for the proposed mobile framework

The top layer is for providing services like third party authentication [8] or login with Google/Facebook, location API [9], and user rating. These services are indulging in most of the apps that are being developed nowadays. This is due to the acceleration of business as well as fantasy app development these days.

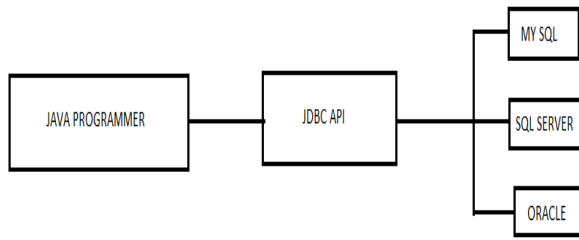


Figure 3-2 Connectivity between web container and different databases in server domain

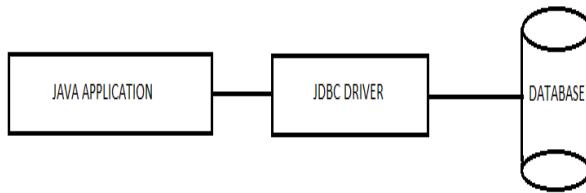


Figure 3-3 JDBC driver between JDBC API and database server

Provided services within this layer such as login, rating and location use the connectivity to the database for dealing with the data obtained within the process. New services can further extend this service layer per the specific needs of related apps. The combined layers of this framework offer developers a useful tool for including database operations and advanced services within their apps. The developer who wants to use MYSQL as a database for developing an app need to establish a connection between Android Studio and MYSQL. Also for establishing connections between other databases and Android Studio, it is needed to perform a connection with JDBC. Instead of making an establishment of connection every-time intending to make a package which offers connection establishment with any prescribed database with a single line of code.

Proposed framework is for android mobile apps. The basic purpose of this framework deals with database connectivity and high level services offered by using this. Not all the developers familiar dealing with all the databases like SQL SERVER, MYSQL [10], ORACLE [11], and SQLITE etc. as shown in the fig 3-2. And every Integrated Development Environment (IDE) for mobile apps is having different in-built database to work with, example SQLITE is in-built database to deal with

for Android Studio. Instead of learning new in built database for pertained IDE we can work on our flexible database by using the proposed framework. To work with external databases, we need JDBC connectivity especially here. The working of JDBC is substantially explained in the fig 3-3 with drivers on JDBC. After the connection with database we can offer high level services like Google/ Facebook authentication, logging, and user rating which are extensively needed in most of the mobile apps.

#### 4. EVALUATION AND FUTURE WORK

The frequent challenges faced by mobile frameworks in furnishing mobile app development are databases, animation, gestures, performance etc. If we look at the problems in detail, then there is a possibility of sorting out few solutions. Some of the android mobile apps confronting performance issues in providing web view compared to others. Even though few functionalities related to web view working smooth, they lack in some important things which are handy. Most of the supported animations from JavaScript and CSS even we are having new scripting languages with substantial attributes. This makes android apps in lag position analogize to disparate mobile apps.

Now coming to our problem statement, most of the mobile frameworks arousing difficulty in dealing with different databases. Generally, every mobile framework has in built database to deal with which in turn they will have predefined methods to query the data. But the limitation is not in providing the connectivity with different database, it is due to complexity in dealing with it. If the developer wants to use different database instead of in built one, each time the developer should make connectivity to get services from it.

The proposed framework neutralizes some of the problems faced in performing database connectivity. Connectivity will be done by using JDBC through server which helps in providing interactive communication between Java and database. The methodology needed in performing this must be constructed in a well scripted manner. The foremost thing we needed is android package which consists of DB Helper class as a parent class. All other databases which need to be used as a part of android package such as MYSQL, SQL SERVER, ORACLE and SQLITE must make sub classes for main parent class DB Helper. The parent

class must extend all these sub classes such that those databases can take advantage from parent class along with their own methods. So, android package consists of all the classes and methods with properties of above mentioned databases extended by super class. But to make use of this package we need some connection handling unit to control the things around.

Demanding services will satisfy the needs for some of the commercial apps. Services such as authentication, location API, and user rating will satisfy those needs and provide accountability to improve the working capability. Furthermore, we can extend the implementation of the framework by extending the features. It can be possible by making an open framework which indeed helpful in developing new features on the top.

## REFERENCE

- [1] Apache, Play 2.5.x documentation. (2015). [Online]. Available: <https://www.playframework.com/documentation/2.5.x/JavaDatabase>
- [2] Ming Xu, Xinchun Yin, Jing Rong, "Researchment and realization based on Android database application technology", *ISCCCA Intl. Sym. Comp. Comm. Con.*, 2013
- [3] S. Sivasubramanian, S. Sivasankaran, S. Thiru, "A proposed android based mobile application to monitor works at remote sites", *IJSR*, vol. 3, no. 2, Feb. 2014
- [4] J. Albabari, B. Albabari, *C# 5.0 in a nutshell*, Reilly Media, 2012
- [5] Apache, Play 2.5.x documentation. (2015). [Online]. Available: <https://www.playframework.com/documentation/2.4.3/ScalaDatabase>
- [6] F3, (2015). [Online]. Available: <https://fatfreeframework.com/databases>
- [7] Google, (2016), [Online]. Available: <https://developer.android.com/guide/index.html>
- [8] Zhang Longjun, Zou Tao, "A Trusted Third Party Based Secure Authentication Scheme of E-Commerce", *IEEE*, Dec 2008
- [9] Yunus Ozen, Oguzhan Ozdemir, Necla Bandirmali, "Android based energy aware real-time location tracking system", *IEEE*, July 2015
- [10] Oracle Corporation, (2016). [Online]. Available: <http://dev.mysql.com/doc/>
- [11] Oracle Corporation, (2016). [Online]. Available: <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/database-093888.html>