Local or External Databases in Android Programming. A Practical Comparative Study

Dorin BIBICU*, Luminita MORARU**, Simona MOLDOVANU***

Abstract

Currently, databases are used in the programming process of software applications. In a computer machine, a database associated to a software application can be localized either at the local or at external level. This paper aims to present, analyze and discuss various methods used to access and operate both the local and external databases from Android programming code and also to provide concrete examples in this respect. An Android software application that simulate the virtual class book has been developed. An external database is used for data storage. The advantages and disadvantages of using both database types are presented.

JEL Classification

D83, L86

Keywords:

Android, App, Database, Class book software

1. Introduction

The mobile applications or apps are software developed to run on mobile devices: smartphones, tablets or mini-PCs. In the last years, the popularity of mobile apps has grown massively [1]. This development has been facilitated by the easy handling of the mobile devices. Today, various mobile operating systems (OS) are designed such as, Android, iOS and Windows. According to [2], the smartphones that run Android OS are dominant in the world smartphones market, accounting for 87 percent of sales. Android is an open-source license system provided by Google. Apps development requires a combination between Java and XML (Extensible Markup Language) techniques. The widely used integrated development environments (IDE) for Android programming are: Android Studio, Eclipse and C# with a Xamarin framework.

A database is defined as organized collections of data (numbers, characters, dates, objects linking and embedding objects). Data is organized taking into account the relationships between them, are updated and saved on various physical supports. In some situations, the database can be localized at local level, in other at external level. The main operations for manipulating databases are as follow: create, insert, update and delete. The commonly DBMS (data base management systems) are: Oracle, MySql or Microsoft Access software. In this paper we present two methods devoted to access local or external databases using an original Android app code, called “ClassBook” and adapted to the educational domain. The rest of the paper is organized as follows: in Section 2, the software materials and the theoretical methods are described. The results and discussions are presented in Section 3. Finally, we conclude on the advantages and disadvantages of using either a local or external database.

2. Materials and Methods

2.1 Materials

The ClassBook app was developed using Android Studio that is the official IDE for app development [3-4]. Also, a Java development kit was requested for this application [5]. Firstly, it is compulsory to configure the Android Studio IDE by downloading and installing the convenient Android Protocol Interface (API), choosing and configuring the Android Virtual Device (ADP) emulator and installing the HAxm routine. We used the Android Studio facilities as the available parts: the declaratory eXtensible Markup Language (XML) and the procedural Java.

2.2 App design

The “ClassBook” app design is showed in figure 1: (a) the login form can be done as a teacher or as a student; (b) the teacher login form and begin to run the students’ scoring process from their mobile devices; (c) the student list associated to a logged teacher; (d) the marks of the student. In the associated Java code there are sql commands executed to the apps’ associated database.
2.3 Database design

The database associated to the “ClassBook” app consist from six tables (figure 2) denoted: students, class, teachers, allocations, marks and absences.

![Database Design](image)

Fig. 2 “ClassBook” app database design consist of following workspaces and list pages: (a) students table, (b) class table, (c) teachers table, (d) allocations table, (e) marks table, (f) absences table.

2.4 Local databases

A local database or internal database (IDB) is stored in the owner computer. Usually, local databases are used to store data in a local laboratory information management system (LIMS) [6] for “in house” sequences search. An easier way to manage a local database in an Android app is to instantiate the SQLiteDatabase class [7]. SQLite is an open-source library developed to manage the relational databases, saved as a single file on the disk [8]. The software app accesses the SQLite library using a header called android.database.sqlite.SQLiteDatabase. At an external level, in order to access a local database, an user should have an access agreement. The Java code proposed to create, insert and interrogate the students table is presented bellow.

```java
//Open database
db=openOrCreateDatabase("students", Context.MODE_PRIVATE, null);
//Create table
db.execSQL("CREATE TABLE IF NOT EXISTS students(id INTEGER, name VARCHAR,class VARCHAR,user VARCHAR, password VARCHAR);");
//Insert data
db.execSQL("INSERT INTO students VALUES(1,'Maria','9','Maria_u','Maria_p');");
//Query data
Cursor c=db.rawQuery("SELECT * FROM students", null);
if(c.getCount()==0)
{
    Toast.makeText(getApplicationContext(),"Error, ndata!",Toast.LENGTH_LONG).show();
    return;
}
StringBuffer buffer=new StringBuffer();
while(c.moveToNext())
{
    buffer.append("Id: "+c.getString(0)+"\n");
    buffer.append("Name: "+c.getString(1)+"\n");
    buffer.append("Class: "+c.getString(2)+"\n");
    buffer.append("Password: "+c.getString(3)+"\n");
    buffer.append("User: "+c.getString(4)+"\n");
}
```

2.5 External databases

The external databases (EDB) are stored in an external server. The communication between app client application and the external database is performed using the Internet, as figure 3 shows. The software app accesses the external database (as an example it can be MySQL), using a few web services scripts written in PHP [9]. In Internet space are many servers able to offer web hosting services, such as: hostinger [10], ipage [11], web.com [12] etc. Almost all web hosting servers offer free, trial or paid services.

Our proposed app directly communicates with an external database using a PHP file, that is additionally stored in the hosting server. The PHP file contains connection with the external database and another instructions on how do the database management. In the next rows we present code used in students_interogation.php PHP file, called for students’ table interrogation.

```php
<?php
define('HOST','mysql.hostinger.in');
define('USER','u259593258_d');
define('PASS','ddddd1');
define('DATABASE_NAME','u259593257_cat');
$con=mysqli_connect(HOST,USER,PASS,DATABASE_NAME);
$sql="select * from students";
$r = mysqli_query($con,$sql);
$name='';
$class='';
$user='';
$password='';
while($row = mysqli_fetch_array($r))
{
    $name.= $name['name'].';';
    $class .= $class['class'].';';
    $user .= $user['user'].';';
    $password .= $password['password'].';';
}
$result='';
$result.= $name."\$class."\$user."\$password.";
echo $result;
mysqli_close($con);
?>
```

The Java source code used in the ClassBook app for interrogation of ‘students table’ is presented below. To be able to do this, we create instantiations of HttClient, HttpPost, HttpResponse and HttpEntity classes.
HttpClient httpClient = new DefaultHttpClient();
HttpPost httpPost = new HttpPost( "http://dorin.16mb.com/classbok/students_interogation.php");  
HttpResponse response = httpClient.execute(httpPost);
HttpEntity entity = response.getEntity();
String responseStr = EntityUtils.toString(entity);  

2.5 Statistical analysis

The statistical t-test and its statistical significance level p-value is a proper method to validate any discernible differences between two feature classes, in our case we refer to local and external databases. The p-value is achieved assuming that the sample values can be approximated by a normal distribution. The probability that a random variable $X$ is less than $x$ is given by the relationship [1], where $\mu$ is the mean and $\sigma$ is the standard deviation:

$$P(X < x) = \int_{-\infty}^{x} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dt$$  \hspace{1cm} (1)

A p-value less than 0.05 indicates a significant difference between the two data sets [13-14].

3. Results and Discussion

Using either a local or an external databases in Android app has pros and cons. In table 1, some advantages and disadvantages of local and external databases were identified.

| Table 1. Advantages and disadvantages of using local and external databases |
|-----------------------------|-----------------------------|
| Local database              | External database           |
| Advantages                  | Disadvantages               |
| - less code being written for; programming; | - more memory consumption; |
| - greater security;         | - cannot be accessed on-line; |
| - no costs;                 | - if a system fails the integrity of the database can be lost; |
| - faster data access;       | - saving memory;            |
|                             | - accessed from Internet;   |
|                             | - if a system fails the integrity of the database is maintained; |
| Advantages                  | Disadvantages               |
| - more code being written for programming; | - less security; |
| - services must be paid;    | - slow data access;         |

As data in table 1 indicates, the advantages counted for local databases are disadvantages for external databases. Mutual affirmation is also valid. However, for any online apps it is mandatory to use an external databases. This allows to the application to access data in different networks or from the internet space.

Practically we evaluated a local database and an external database in a comparative approach based on the database speed access. Sixty experimental tests were made by ten teachers and fifty students in order to rate and compare the ClassBook app implemented using a local database (ClassBook_IDB) and the ClassBook app implemented using an external database (ClassBook_EDB). Each subject used his personal mobile device to execute a predefined set of operations as follows: app login & load the student marks or app login & load the students listed & load the student marks & insert any particular student mark. The elapsed time operation is recorded in seconds for both ClassBook apps. The experimental data is shown in figure 4. The mean and standard deviation values calculated for both databases are displayed in table 2. They reveal the higher speed access of the EDB against IDB. Also the 0.05 p-value certifies the significant difference between the run-time recorded in the each experimental dataset.

![Fig. 4 Elapsed time for ClassBook_IDB vs. ClassBook_EDB](image-url)
<table>
<thead>
<tr>
<th>databases</th>
<th>mean ± st. dev.</th>
<th>t-test p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassBook_IDB</td>
<td>12.95 ± 5.76</td>
<td>0.029</td>
</tr>
<tr>
<td>ClassBook_EDB</td>
<td>10.81 ± 4.70</td>
<td>0.029</td>
</tr>
</tbody>
</table>

4. Conclusions

A method and two code examples devoted to use a local and an external databases in Android applications were presented. An app code design to create and manage databases in Android applications was presented as an original app called “ClassBook”. A comparison between the advantages and disadvantages of using local and external databases in app programming was performed. In the case of online apps, the necessity of external databases usage is obvious. When the run-time speed was addressed, our experimental results clearly highlighted the advantage of using external databases.

At the end, we have to say that the implementation of the ClassBook app is not limited only to the features presented in this paper. Further research will be considered to improve the design and to examine the possibility of adding new modules in the ClassBook app.

References