eLearning Mobile App for Android and Ios "English Grammar Learn&Test"

Anca-Georgiana FODOR, Bogdan Vasile COVACI
LABsterzz.com
anca@labsterzz.com, bogdan@labsterzz.com

This article is aiming to present the architecture and few elements from the developing cycle of "English Grammar Learn &Test" app. This is an e-learning tool for people who want to improve their English Grammar and Vocabulary. The app was approved by Google Play and Apple Store and it is available for free on both platforms as following:
Android: https://play.google.com/store/apps/details?id=com.labsterzz.english_tests
iOS: https://itunes.apple.com/us/app/english-grammar-learn-test/id1126468980
The app already reached350.000 users, it is rated at 4.43out of maximum 5.0 in Google Play Store. Since mid-June 2016, we launched the app also in the Apple Store iOS devices.
***Keywords**: e-learning, m-Learning,English Grammar Learn & Test, Android Mobile Apps, IOS Mobile Apps, SQLite, Eclipse, XCode, Java, Swift.*

1 Introduction

The scope of launching mobile e-learning App for Android and iOS systems "*English Grammar Learn & Test*" was to help users across the world improving their English skills enhancing their vocabulary and practicing grammar.
According to [3], student reactions to grammar-focused lessons seem to be typically one of three kinds. Some students find grammar very appealing, some find it intrinsically boring, and some find it useful but really hard to comprehend. In this context, making grammar appealing for students might intrigue their curiosity and turn them back to the English learning table.
Bearing in mind this goal, we developed a simple, enjoyable and useful app for Android and iOS devices - smartphones and tablets –a kind of pocket book edition.
The concept is designed to create a daily 5 minutes education window wherever the student is: in the school-bus, waiting for the tube, in the park, etc., since the smartphone is their pocket-ready educational tool.
As content, "*English Grammar Test & Learning"* app offers 120 short grammar and vocabulary lessons, more than 1000 grammar questions, 20 tests for practice (each containing 30 random questions), Evolution Table and the wrong answers list after each test.
The app is free, ready to be used and available at the below addresses:

- Android: https://play.google.com/store/apps/details?id=com.labsterzz.english_tests
- iOS: https://itunes.apple.com/us/app/english-grammar-learn-test/id1126468980

2. E-learning world

Nowadays, during the Information Era, people are traveling a lot, are trying to optimize their time, space and life.
In the last years, we crossed over from reading paper based books in nice classical libraries to electronic learning and reading in the bus, tube or plain. We talk about big amount of information which is flying around such as Big Data which comes to the forefront in e-Learning soon.
The trend in commercial companies and education institutions [1] for example is moving very fast to e-learning, as this approach is opening a lot of opportunities for staff and students development with a proper balance between expenses (less travel costs, less time) and resources. This

will engage more learners to online education.

We hear more and more about *m-Learning* which follows the boost of mobile device development. People prefer small computing gadgets over relatively bulky laptops mainly for their portability. For this reason, m-Learning started to be a trend and we expect to rapidly evolve during next years.

As expectation [11], new apps with more expanded features in online education will flood the market in the near future. As effect, they will give access to e-Learning to much more people than before and make e-Learning more widespread as people will be able to make use of it on the go. Especially because the time seems to compress and people are traveling more and more and their natural tendency is to use the time in their benefit.

In addition, Social Networks shouldn't be neglected. This is an opportunity of live communication between people. Social Networks can become a strong amplifier as they are a means of direct communication with users and potential users.

According to [4], one of the best examples of this trend is Twitter. This social media network has been ranked # 1 in the list of Top 100 Tools for Learning for 7 years in a row since 2009.

This trend will lead to the increase of education level of the people, due to m-Learning accessibility. There are millions of apps already in Google Play and Apple App Store ready to be downloaded and used. It's true that a part of them are games, however even *gamification* which is a trend as well can be used for education. Nothing stimulates learners better than challenges, points, badges and leader boards. Number of educational apps developed as games is rapidly growing and is projected to double its quantity in the near future.

We expect that 2017 will show us a great rise of *gamification* and this will give a foundation for its implementation in augmented learning.

As a short conclusion, e-Learning, the actual trend in education which is growing, gives time a new scale enlightens travel, saves costs and gives the chance for better education to much more people.

In a specific case, m-Learning is becoming much stronger and it is a clear trend for near future because is coming with an additional value: portability and availability via mobile apps ready to be used on your smartphone or tablet.

3. Application architecture

"*English Grammar Learn & Test*" is three-tier architecture for both versions, Android and iOS, as it is presented in Figure 1.

Despite the similar architecture, the two versions of the app are developed with different tools, each of them with its specificities of development environment, types of scalability, database manipulation and programming language used.

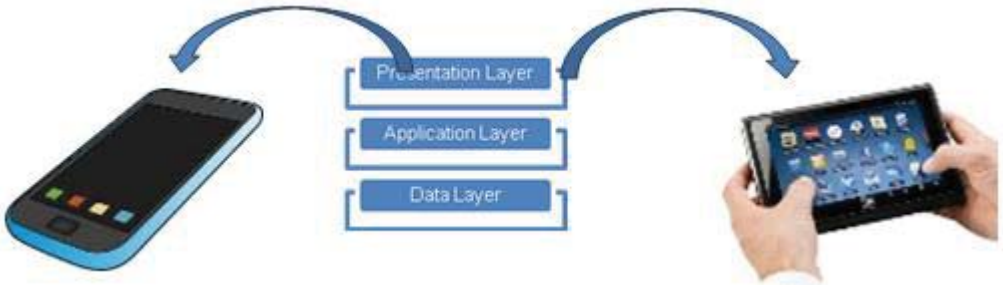

Fig. 1"*English Grammar Learn &Test*" app architecture

4. Data Layer

Data Layer was built using SQLite which is a relational database management system contained in a C programming library, according to [8].

In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program, as part of the application program–Eclipse [5] for Android operating system and Xcode [6] for iOS operating system – on our specific case.

Due to the server-less design, SQLite applications require less configuration than client-server databases. SQLite [8] is called zero-configuration because it does not require service management. This is a clear advantage in database usage as SQLite engine has no standalone processes which the application program communicates with. Instead, the SQLite library is linked n and thus becomes an integrated part of the app program.

For "*English Grammar Learn&Test*" app's data layer, the most important Table is QUESTIONS_LIST - which stores the questions for tests, together with answers options and the correct answer as well. This is created using the sequence of instructions (*Android version* sample): we will start by creating a database called "questionsDB" and set a version number as in the example below. The reason for assigning a version number for the database is to make SQLite aware when we want to update the content of the database.

```
public class E_1_DbHelper extends
SQLiteOpenHelper {

public static final String DB_NAME =
"questionsDB";
public static final int DB_VERSION =
1;
```

Afterwards we will declare the creation of QUESTIONS_LIST table. As it can be seen we use simple SQL language for the CREATE statement.

```
public static final String TABLE =
"QUESTIONS_LIST";
public static final String C_ID =
BaseColumns._ID;
public static final String C_Q = "Q";
public static final String C_R1= "R1";
public static final String C_R2= "R2";
public static final String C_R3= "R3";
public static final String C_R4= "R4";
public static final String C_RC= "RC";
public static final String C_CATEG=
"Q_CATEG";
public static final String C_SHOW=
"Q_SHOWN";
public static final String
CREATE_TABLE = "CREATE TABLE " + TABLE
+ " ("
+ C_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT, "
+ C_Q+ " VARCHAR(255), "
+ C_R1 + " VARCHAR, "
+ C_R2 + " VARCHAR, "
+ C_R3 + " VARCHAR, "
+ C_R4 + " VARCHAR, "
+ C_RC + " VARCHAR, "
+ C_CATEG + " VARCHAR, "
+ C_SHOW + " INTEGER);";
```

Then the database is populated with questions (one sample below):

```
db.execSQL("INSERT INTO " +TABLE+"
(q,r1,r2,r3,r4,rc,q_categ)
values ('In ten years' time, all
students _____ their own computers in
school.','will have','had
','have','have been','will
have','Future');");
```

Two functions are called when accessing a database: onCreate and onUpgrade:

- onCreate runs only once if it does not detects a database installed on the device associated with the app (or if there is already a table in the targeted Schema with the same name) and will execute all the SQL statements included in its boundary. (E.g. create table, insert into etc.).
- onUpgrade will run every time when the database version is changed into a higher value (e.g. new value 2 instead of 1).

As there will be cases when you need to update your database with new content it is a good practice to put all DROP statements inside onUpgrade func and add your new content inside the onCreate func.

In addition to QUESTIONS_LIST table, the app uses few more tables to manage the tests done, results obtained, wrong answers list to, Evolution Table were we store the users progress, lessons attended, etc.

For *iOS version*, we used the SQL Lite Database package [6], already built in

Xcode platform, working in a similar manner at conceptual level, with some platform specificities such additional steps required to be done prior (e.g. database needs to be created in a console mode – presented in the Fig. 2).

Fig. 2. SQLite for IOS console

5. Application Layer

Application layer is dependent on the operating system platform. "*English Grammar Test & Learning*" was built on Eclipse platform for *Android* version by using the already available SQLite Database package, called from the main module of the application. As main elements used for *Android version* of the application there are packages to access SQLite Database:

```
package com.xxx.english_tests;
import
android.database.sqlite.SQLiteDatabase
;
import
android.database.sqlite.SQLiteOpenHelp
er;
```

To open and close the connection to database there are used bellow functions:

```
public void open()
{
       this.getWritableDatabase();
       // return this;
}

public void close()
{
       this.close();
}
```

Main code elements of the *IOS Version* of the app is presented below:

```
import UIKit
import SQLite
import CoreData

public class dbHelper:
UIViewController {

// opens the database
// we will use the returning
value  (db)  in our sql statements.

class func openDatabase() ->
COpaquePointer
{
 var db: COpaquePointer = nil

 let dbLocation
=  AppDelegate.copyBundledSQLiteDB()

if sqlite3_open(dbLocation, &db) ==
SQLITE_OK {
      return db
        }
else {
      print("Unable to open database.
Verify path")
      return db
        }
    }

// example of a SQL count statement
// read that we need to transform the
outcome to a Int32

class func
queryCountFromQuestiondDB()->Int32{
   let db = openDatabase()
```

```
let queryStatementString = "SELECT
count(*) FROM questionsDB;"
   var queryStatement: COpaquePointer
= nil

   sqlite3_prepare_v2(db,queryStatemen
tString, -1, &queryStatement, nil)
   sqlite3_step(queryStatement)

// declare a constant (let) to link it
to the result // 0 is the column
index. In swift we count from 0
   let countResult =
sqlite3_column_int(queryStatement, 0)

// finalize the sql statement
   sqlite3_finalize(queryStatement)

   return countResult
    }


//classic "select * from …" statement

class func queryList30questions()-
>NSArray{
    let db = openDatabase()
    let queryStatementString = "SELECT
* FROM questionsDB;"
    var queryStatement: COpaquePointer
= nil

//declare anempty array to throw the
data in
    var queryResult: [String] = [""]

    sqlite3_prepare_v2(db,
queryStatementString, -1,
&queryStatement, nil)

//sqlite will fetch one row at a time
so we need to introduce a while clause
// ==SQLITE_ROW … will make sqlite
read records until it reaches an empty
row
    while
(sqlite3_step(queryStatement) ==
SQLITE_ROW) {

// as in the example above create
constants to link them to the query
result
     let Q =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
1)))!
     let R1 =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
2)))!
     let R2 =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
3)))!
     let R3 =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
4)))!
     let R4 =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
5)))!
     let RC =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
6)))!
     let QCATEG =
String.fromCString(UnsafePointer<CChar
>(sqlite3_column_text(queryStatement,
7)))!

// append in the earlier created array
the query result.
     queryResult.insert(Q+"|"+R1+"|"+R
2+"|"+R3+"|"+R4+"|"+RC+"|"+QCATEG,
atIndex: 0)
        }
    sqlite3_finalize(queryStatement)
    sqlite3_close(queryStatement)
    sqlite3_close(db)

    return queryResult
    }
}
```

On iOS, SQLite does not require assigning a version number to the database.
When opening a database on iOS, SQLite will check if the database exists on the target device and if not it will copy the database created above.
This can be a problem if we need to update our database with new content.
In our case in order to overcome this we have split the database in two with one database containing the QUESTION_LIST table and one containing all user saved data (e.g. lessons attended, test scores, medals).
In case we will update the QUESTION_LIST with a new set of question we will create a new database containing the new QUESTION_LIST table.
Because we cannot send to the users a new version of our database without updating the app itself on the AppStore and in this way changing the version number of the app we will make use of this new app version to check what version the user is currently using inside the `openDatabase()` function explained above.
All English lessons are designed in a static manner, as no interactivity is required from the app users. The only database activity in

this area is related to status of each lesson, read or not-read by the user and the Summary presented per lessons category.
We also used functions to manage the tests done by local user and unlock the next test depending on accomplishments, evolution table management, wrong answer list, etc.

6. Presentation Layer

The app interface for users is the same for both Android and iOS versions. Below, in Figure 3, there are few samples of the app interfaces:

Fig.3 *English Grammar Learn & Test* app presentation layer samples

The most important part of the Presentation Layer, in this specific case of Mobile app is to insure the scalability for the entire range of devices (smartphones and tablets). As a solution for this part, we treated separately smartphones and their standards.

Basically you will need to ensure that your app will look the same no matter what device the user is using.

For this we used the built-in simulators provided by Xcode for iOS and Eclipse for Android.

If for iOS this can be an easy task as there are no more than 6-7 devices with different height and width for Android it can be very time consuming since today there are more than 10,000 Android devices on the market with tens of combinations of height and width.

One tip from us is to go to [9] and structure the devices with the biggest market share on groups of same aspect ratio and scale the screen to accommodate them.

7. Conclusions

"*English Grammar Learn & Test*" is a **simple**, user - **friendly** and appreciated app.

English Grammar Test & Learning

Google Statistics today:

- Downloads: 350.000
- No of last month sessions: 170.000
- No of Ratings: 6.000
- Rating: 4.4 (maximum is 5 and competition average is 4.1)

According to [4] in 2017 we expect to see e-learning becoming even more individual, well-timed, and technologically advanced.

In this respect *m-Learning* is earning more and more space into our educational life. The apps offer is growing fast in Google Play and Apple App Store, users are becoming selective and they are expressing their needs while choosing the most appropriate apps for them.

According to our user's feedback on Google Play platform, the app comes handy for Grammar practice, for Vocabulary development, easy to use and intuitive.

User's feedback is a very important aspect we look after as it shows us their needs and desires so we treat it very seriously.

The app is built on three-tier architecture and it is available for both Android and IOS platforms with the same presentation Layer, scalable for all types and standards of mobile devices. Differences between versions are derived from different

platforms, different programming language, database usage and calls from source code.

Scalability is as important as the user's feedback because the user has to see the app proper scaled on his own smartphone as well and in the same manner as on a tablet device, otherwise any app, no matter how life changing will be can be underestimated since the beginning.

From the content point of view, "*English Grammar Learn & Test"* app offers 120 short grammar and vocabulary lessons, more than 1000 grammar questions, 20 tests to practice (each containing 30 random questions), an Evolution Table and wrong answers after each test. The content is extremely important, to keep the user interested in making him use your app on a daily basis.

Apart from the app development cycle we will talk about entering the mobile apps market. [10]There is no other better reward for the developer than knowing his own app is part of the daily activities of users across the world. To achieve this, we tracked our app parameters on a daily basis: downloads per day, user retention rate, number of ratings and reviews received per day, average time spent by users within our app, number of sessions in a specific time period, etc. and compared them with our peers (were available). The commercial part requests a lot of time, discipline, energy and commitment.

We are also constantly monitoring the feedback our users are sending and we make sure they get a response within 1 working day.

According to [10], a marketing study focused on Mobile Apps, there are a number of interesting conclusions regarding user's behaviour:

- The average app user has 36 apps installed on his or her smartphone.
- most installed apps are not used often – 26% those apps are used daily, while 1 in 4 apps are never used;
- App discovery can occurs out of the app store–52 % of users are aware of apps from friends, family, and colleagues and 24% discover an app through it's company website, then having a website in place before you launch your app is a good idea;
- 2 out of 3 users consider the average ratings of an app as an important factor when deciding to download and the same number consider the app description an important factor. Your app description should tell the user what to expect when downloading your app as clear and simple as possible.
- 3 out of 4 users expect apps to be free but the willingness to pay is $2.17
- Making users life easier and always having new content are two of the top attributes associated with highly used apps. A good practice is to launch a new update once 3-4 weeks.

Developing, launching and keeping alive a mobile app is a balance between technology, marketing, dynamism and effort to understand and adapt to users' needs [3].

According to statistics [7] and based on our experience, the hard work only starts after the app is launched.

Acknowledgment

This paper presents the architecture and few elements from the developing cycle of "*English Grammar Learn & Test*" app, an educational tool for users across the world who want to improve their English Grammar and Vocabulary. This app was approved by Google Play and Apple Store and is available to download for free on both platforms.

References:

[1]. Velicanu Anda, Lungu Ion, Diaconita Vlad, Codrin-Florentin Nisioiu - *Cloud E-learning*, Volume of 9th International Conference"E-learning and software for education", 2013

[2]. Catalin Ionut Silvestru, Constantin Marian Matei, Codrin Nisioiu, Dragos Stefan Silvestru - *The E-Learning Systems and Platforms Role and*

Involvement in New Economy, Volume of International Conference " Knowledge Management – Projects, Systems and Technologies, 2008
[3]. http://www.onestopenglish.com/methodology/methodology/grammar-vocabulary-and-skills/grammar-and-vocabulary-seven-ways-to-help-students-enjoy-grammar/146459.article
[4]. Masolova Elena – "7 e-learning trends to keep an eye on it in 2016", on-line article https://trainingmag.com/7-e-learning-trends-keep-eye-2016
[5]. https://eclipse.org/downloads/
[6]. http://xcode-mac.en.softonic.com/mac
[7]. https://en.wikipedia.org/wiki/SQLite
[8]. http://sqllite.com/
[9]. https://design.google.com/devices/
[10]. Mobile Application Marketing Insights https://think.storage.googleapis.com/docs/mobile-app-marketing-insights.pdf
[11]. http://www.ispringsolutions.com/blog/top-e-learning-trends-in-2016-follow-or-ignore/

Anca-Georgiana FODOR graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies in 1998. She got the title of doctor in economy in the specialty economic informatics in 2008. At present she is working for Banca Comerciala Romana.

Bogdan Vasile COVACI graduated from the Faculty of Economics of the Dimitrie Cantemir University in 2008. At present he is attending Master courses at the Academy of Economic Studies (Databases – Support for Business) and he is working for Banca Comerciala Romana.
Together with Anca-Georgiana FODOR, they started to develop Mobile Apps as LABsterzz in 2014.