Design and Development of an Android Accounting Application Using Web Services and Quality of Experience for Mobile Computing

Ustijana Rechkoska
"St. Paul the Apostle" University
Faculty for Computer Science and Engineering
Ohrid, Republic of Macedonia
ustijana@gmail.com

Danco Davcev
"Sts Cyril and Methodius" University
Faculty for Computer Science and Engineering
Skopje, Republic of Macedonia
dancodavcev@gmail.com

Darjan Djamtovski
"St. Paul the Apostle" University
Faculty for Communication Network Security
Ohrid, Republic of Macedonia
dzamtomk@gmail.com

Carlo Ciulla
"St. Paul the Apostle" University
Faculty for ISVMA
Ohrid, Republic of Macedonia
cxc2728@njit.edu

Jordan Sikoski
State University of Tetovo
Tetovo, Republic of Macedonia
jordans@t-home.mk

Abstract—In this work, we propose an easier and innovative way of calculating the salary of institutions' employees, through their smartphones running the *Android OS*. The application is developed in *Android Ice Cream Sandwich* - the most popular version of *Android*. The specifications and features of the smartphones had achieved the highest level of quality, allowing performance of tasks concerning mobile computing. The application proposed in this work allows calculating the salaries of the employees of the institution - Universities staff, proposing an original, efficient way of documentation managing at institutions. Our approach improves the *Quality of Experience (QoE*), providing the users all necessary resources and performances, which refer to a user-friendly mobile application, using *Web Services*, *QoE* evaluated referring the quality of the graphical user interface, efficiency and time saving, enabling *Cloud connected experience* possibility.

Keywords-Android application development; mobile computing; salary calculation; Web Services; Quality of Experience (QoE)

I. INTRODUCTION

Android, as a *Linux*-based operating system designed for touchscreen devices like smartphones and tablet computers, is an open source and its code is released under the Apache License which allows the software to be modified and distributed by device manufacturers, wireless carriers, as well as individual users [8][9].

The history of *Android* raises through some versions, from 1.0 to 4.4.2, so far. In this work, we have chosen *Android 4.0* because it provides new graphical interface; there is a security concept improvement and it involves the cloud environment connection. One of the main features of *Android 4.0* is the new *connected Cloud experience*, *Android Beam*, *refined User Interface (UI)*, security for applications, content and enhancement for enterprise *Virtual Private Network (VPN)* client *Application Programming Interface (API)*. The design of this application is developed consistently with the new *refined UI* which allows easier usability and easier transition of the salary calculation. This application is connected to the *Web Server,* which allows dynamic download of the necessary information for calculation of the salary. The features of this application are described in the third section of this work.

In the next section, the performed tests and results are presented, continuing at the fifth section of this work with *Quality of Experience (QoE)* evaluation concerning mobile computing.

Our approach for managing the documentation of an institution provides saving time, energy and money. It provides comparison to previously developed *Android* applications and gives the reasons why this application is more suitable for usage. This work provides the features of the application and how they can be used. The results of the tests done to the application are presented. The conclusion and our concept for working on this type of application are given. The *Cloud connected experience* allows users to synchronize: photos, e-mails, applications, and contacts [2]. *Android 4.0* provides easier implementation of the applications to manage the authentication and the *secure session* [2]. One of the important security features is that *Android 4.0* allows encrypted storage and remote data deletion [2]. This feature is the most helpful when the device is lost. The last feature of *Android 4.0* is the enhancement for enterprise (*VPN client API*). This feature allows us to construct the application to configure the addresses and routing rules, process outgoing and incoming packets and establish secure tunnels to remote server. This feature allows the application to be configured with *centric networks*. The main point of the implementation of this application is the usage of *Web Services*, which is going to provide better performance and possibility for calculation of the salaries for institutions having thousands of employees. By using the Cloud connection possibility, the application is going to result in saving energy, time and money of the institutions and numerous users stated in the conclusion and future work concepts of the paper.

II. RELATED WORK

Nowadays, the *Android* market is full of applications giving the users greater choice, but not all of them provide excellent quality to the user. There are many applications that have great marketing characteristics, but poor performances in terms of usability and efficiency. On the Android market several applications for calculating the salary can be found:

- Salary Bot by CAB Designs [10]
- Paycheck by Green App Developer [11]
- Quick Wage by CWE Software LLC [12]

Testing and analyzing these applications, *Salary Bot* [10] has the best performances among three of them, which allows the results to be broken yearly, monthly, weekly, daily, and hourly data; so, the design is much better than the other two applications. The common feature of these applications is the calculation of the gross salary with the additional work of the user. The *Paycheck* application [11] has simple and old design which is not consistent with the updates of Android and it provides approximate calculation of the salary. *Quick Wage* [12] allows the user to calculate the salary according to the wager. It also has old design which is not consistent with the Android updates and provides yearly, monthly, weekly, daily, and hourly salary calculation, providing salary calculation for several years. The disadvantage of *Quick Wage* is that it has an approximate calculation of salary.

The applications which are already developed allow the user to calculate the salary according to the user’s gross earnings, but the gross earnings must also be calculated. The application that we decided to develop allows Web integration which gives the user easier way to calculate the salary. The user does not need to have previous knowledge on how to calculate the salary. This application is mainly developed for calculating the salary of the University staff, but it can be easily adjusted to other institutions as well. It is user-friendly, web integrated mobile application and its structure is presented in detail in the next section.

III. ANDROID APPLICATION

UniSal Android application is created to help the University staff mostly to calculate the salary in a user-friendly and efficient manner. The necessary information for calculating the salary is downloaded from the *Web Server* and parsed into usable information within the application (Figure 1). The information for calculating the salary is stored into *MySQL* database [14], which is stored into the *Web Server*. For the future development this application is going to give the *Cloud experience* to the user, so the application will be used by thousands of employees.

The application is calculating the gross salary based on monthly data for each employee in the institution, as well as on the basis of the prescribed rates and monthly average salary in our country.

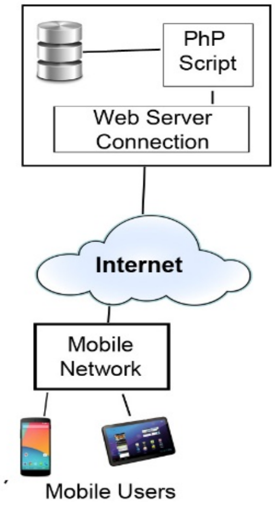


Figure 1. System architecture for mobile computing using Web Services.

The variables for calculating the Gross Salary *(GS)* are given as follows: Net Salary *(NS)*, Personal Income Tax *(PIT)*, and Contributions *(C)*. The equation for calculating gross salary is [6]:

$$GS = NS + PIT + C \qquad (1)$$

The category of salary is determined according to the law, the collective agreement and the employment contract. In order to calculate the salary, it is necessary to submit the data for its calculation to the salary referent in charge of data entry on the part of the authorized departments and managers in the firm. For calculating the salary the institution also has to calculate the personal income tax and the contributions. To calculate the amount of personal income tax and salary contributions starting the 1st of January *2009*, the rate of the personal income tax in the taxation of salaries is *10%* regardless of the amount of the employee’s salary. Thereby the basis for the calculation of the personal income is the basis for the calculation of contributions (gross salary), less the total amount of the contributions and the personal exemption determined in monthly amount [6]. The variables for calculating the Basis for Personal Income Tax (BPIT) are: Gross Salary (GS), Contributions (C) and Personal Exemption (PE).

$$BPIT = GS – C – PE \qquad (2)$$

Contributions are calculated according to the rates determined into the Law regulations of the country. Table I describes the rates for the contributions [4][5]. The total sum of contributions for *2013* is described as (TSC).

Calculating contributions for the salary are presented in Table 1, as given below:

TABLE I. CONTRIBUTIONS FOR CALCULATING THE SALARY

Contributions	*Period*
	From the 1st of January
Mandatory PDI	18%
Mandatory health insurance	7.3%
Mandatory insurance in case of unemployment	1.2%
Additional contribution for compulsory health insurance in case of injury at work and occupational disease	0.5%

$$GS = NS + ((NS – PE) \times 11.111111) / 0.73 \qquad (3)$$

$$*0.73 = (100 – TSC) / 100$$

Equation (3) represents the way of calculating the GS of the employer [7].

A. Structure of the Application UniSal

This Android application *UniSal* is connected with remote database (*MySQL*) [14] and all required information for salary calculation is stored into the database. The information retrieved from the database is *JSON* [15] code, which is parsed when it is received into the application. Before it is parsed, the *JSON* code is received as a *JSON* Array and then each Object is accessed separately. All the values that are received from the database are stored into the Android device and are used for offline calculation of the salary. The *MySQL* database is saved into *Apache 2.0 Web Server* with *PhP 5* [16] *installation*. The *PhP scripts* are very important for this application, because the Android application does not need to have authentication information with the *MySQL* database. This provides *secure communication* between the *Android* application and *MySQL* database because all the information for authentication with the *MySQL* database is stored into the *PhP* scripts which can be accessed only from the server. When the Android application makes a *Http Post* request to the server, the server executes a specific script for returning the *JSON* code to the Android application.

The Main screen of the application is consisting of *ViewPager* that holds the fragments which are managed with *Section Adapter* and *Action Bar* that holds the tabs i.e. the style of the *Action Bar* is set to “Tab Style”. On the main screen, two tabs are displayed and they represent the calculation of the Gross and the Net Salary. The user interface is user-friendly and the navigation is done with swiping right-to-left or reverse.

The specific salary is calculated from the mandatory *PDI*, health insurance, contribution for employment, professional sick leave, total contributions, total contributions and charges. Another important section is the *Settings Panel* of the application. The settings panel allows the user to change the values of the contributions according to the changes done according to the Law regulations.

Once the application is launched, it checks the contributions on the remote server. If the *Android* device is connected on the *Internet*, the values for the contributions are downloaded automatically and stored into the device for offline calculation of the salary. The values are downloaded by sending a *Http Post* request to the server. The remote server holds *PhP* script for sending the values from *MySQL* database to the Android device. The script authenticates with the *MySQL* library and queries the values. Once the values are queried, they are parsed into *JSON* code and the code is sent to the *Android* device. When the *Android* device receives the *JSON* code it parses it and the values are saved into *xml* file for offline calculation of the salary. The Settings Panel is also available and allows the user to change the contribution values manually.

Due to improper changes into the Settings Panel, the application is not going to calculate the proper value of the salary for the employer. For this case scenario, the application generates error log holding the error values for the contributions. The error log is displayed to the user when he/she wants to calculate the salary. When the user presses the “Calculate” button the error log is displayed. The error log holds the exact contribution value which has error i.e. a value which is not reasonable for calculating the salary of the employee.

B. University staff salary calculation

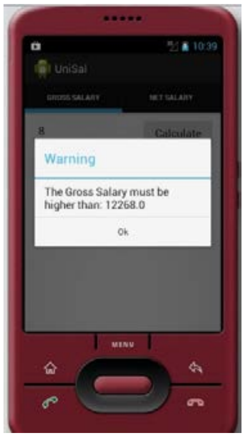
If the user wishes to calculate the salary based on points, he/she has to open the options menu on the Android device and tap on “Salary on points”. The application is going to open new activity containing the proper information for calculating the salary. The salary on points contains the following *Android* components: *Spinner*, *Button*, and *Checkbox*. The *Spinner* is used to hold the academic status; the *Checkbox* is used to determine whether the employee has been on a sick leave; and the *Button* is used to calculate the salary. The information into the *Spinner* is populated from *Remote Web Server*. When the information is downloaded into the device, it is stored into local database for offline calculation of the salary.

The information from the *Web Server* is retrieved by sending a *Http Post* request to the *Web Server*. Into the web server a *PhP* script is stored for retrieving the information to the *Android* device. The *PhP* script authenticates with the database on the server and queries the information that has to be sent to the Android device. The queried information is parsed into JSON code and sent to the Android device. On the Android hand side the information received is parsed into JSON Array. The information from the JSON Array is processed and it is saved into the local *SQLite* database for offline calculation of the salary. When the user presses the "Calculate" button, the salary is calculated according to the information downloaded from the remote server. If the "Sick leave" option is marked and the "Calculate" button is pressed, the calculated salary amount is going to equal *70 %* of the full amount. This is determined by the Law regulations in the country.

IV. TESTING THE APPLICATION AND RESULTS

The application testing is performed in a couple of steps: testing the Gross and Net salary, testing the Settings Panel and testing of University staff salary.

A. *Testing the gross and the net salary*



Warning

The Gross Salary must be higher than: 12268.0

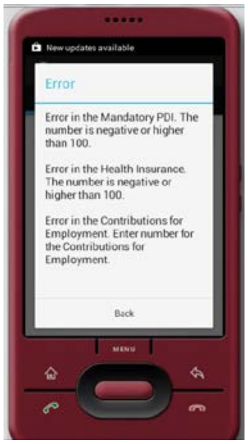
Figure 2. Gross Salary test result.

For the gross and the net salary, we have applied the test scenarios - how the application responds when there is no active Internet connection, how the application responds when *0* is entered as a value and the last test is how the application responds when the negative button is returned as a value. For the first test scenario we have entered gross salary which is below the minimum gross salary determined by the Law regulations. Figure 2 gives the result when the application is notifying the user that he/she needs to enter salary greater than the minimum salary - the result is the same when the user enters *0*. Of course, the user can enter only positive numbers. In the next test, we have left the field for the salary empty and tried to calculate the salary. When the "Calculate" button was pressed, the application responded with a message that the user has to enter salary. If the user is trying to download the data from the Web Server when there is no Internet connection, the user is going to receive notification that there is problem with the Internet connection. The same test scenarios were applied for the net salary. The difference between these two types of salaries is the minimum salary determined according to the Law regulations. The response of the application when the salary amount is below the minimum salary amount of the country is also presented in Figure 2.

B. *Testing the Settings Panel*

This test is used to determine whether the user has made any mistakes inserting the values for the contributions when there is no internet connection. The information for the contributions which is downloaded from the *Web Server* is updated according to the Law regulations and the application is updating accordingly. When the user is unable to connect to the Internet and there is a need for calculating the salary urgently, the user can enter the values for the contributions manually. When the application is performed by the user, the latest values for the salary contributions are already inserted.

This case scenario is displaying the result when the user has made a mistake entering the contribution values. Figure 3 presents the result when the user has made a mistake entering the values manually. This type of error dialog is only generated if the user has entered a value below zero, a value above one hundred and instead of number has entered an alphabetic character. If the value is below zero, it means that the government should pay the user, which is not real; if the value is above one hundred that means that the whole salary and more should go for the government, which is also not real. When a character is entered as a value, the error dialog is also generated, because a positive number is required as a value..



Error

Error in the Mandatory PDI. The number is negative or higher than 100.

Error in the Health Insurance. The number is negative or higher than 100.

Error in the Contributions for Employment. Enter number for the Contributions for Employment.

Figure 3. Settings Panel test result.

C. *Testing the University staff salary*

This test is performed to determine whether the values are successfully downloaded and correctly displayed into the application. When the user enters the application section for calculating the University staff salary, the values are automatically downloaded if there is available internet connection. If there is no Internet connection, the user is notified that he needs to connect to the internet in order to calculate the salary. If the "Calculate" button is pressed when the values are not downloaded the user is going to receive notification that the value cannot be zero. If no value is returned the application is considering that the value is zero. If the application is connected to the Internet the values are downloaded, parsed and added into the spinner. Now, the user can calculate the salary according to the downloaded values. If the checkbox "Sick leave" is marked, *70 %* of the salary is calculated according to the Law regulations.

V. QUALITY OF EXPERIENCE (QOE) OF MOBILE COMPUTING

Quality of Service (QoS) refers to the technical operational aspects of a service, such as time to support services, capacity, and transport. *Quality of Experience (QoE)* measures the difference between what users expected and what they actually received. Using the *QoE* is beneficial to estimate the perception of the user about the quality of a particular service and it depends on the customer's satisfaction in terms of usability, accessibility, retaining ability and integrity of using specific service [13]. *QoE* means overall acceptability of an application or service, as perceived subjectively by the end-user and represents multidimensional subjective concept that is not easy to evaluate (Figure 4). In our work, we have used *QoE* evaluation in order to measure the quality of the mobile application usage.

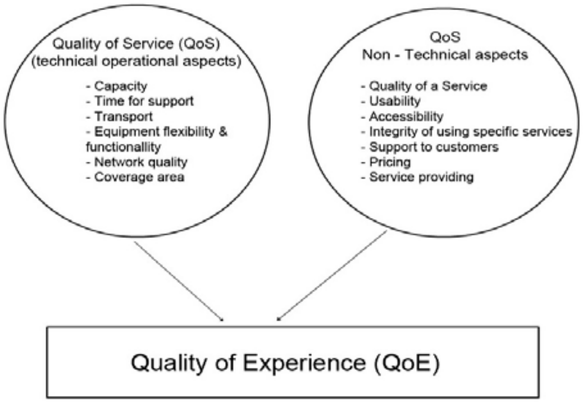


Figure 4. Relation between QoS & QoE.

The results presented in the following chart were obtained according the Quality of Experience evaluation performed with the University staff. The application is tested and evaluated by different scenarios. The survey questions were answered by a group of representatives of the University staff that participated in the *Android* application implementation locally (*UniSal*) and using Web Services (*UniSal Web*) as well. Analyzing the answers from the Mobile application implementation using *Web Services* has provided with the summary given in histogram presented in Figure 5.

The first concept evaluation was the quality of the *Graphical User Interface* of the *Android* application. By the observation the following things can be concluded: *76 %* of the University staff participants concluded that the graphical design of the Web application was *Android* consistent and easy to use; they were supportive about the design and suggest the idea for the colors to be according to the University colors.

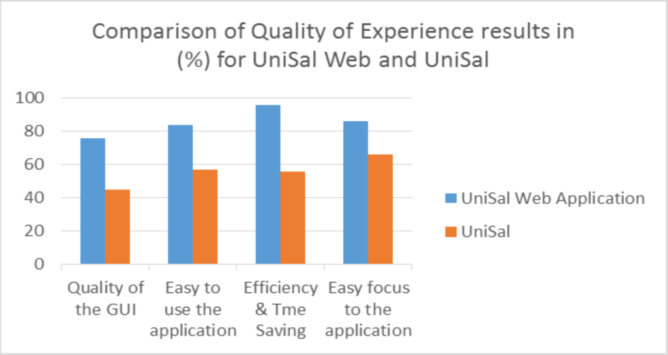


Figure 5. Comparison of QoE results in %.

The second evaluation concept was Usability of the Android application. As presented in Figure 5, *84 %* of the University staff concluded that the application is very efficient and usable because it is *Web* based application and all necessary information for calculating the salary is downloaded automatically.

The third evaluation scenario was concerned with the Efficiency and Time Saving. *96 %* of the University staff concluded that by using this application time is saved when calculating the salary, because all the information is downloaded from the Internet, so energy, time, and money are saved. It is an efficient mobile application, while 56% were declaring the efficiency of *UniSal Android application* which is not *Web* oriented.

86 % of the University staff concluded that it is easy to focus to the Web-oriented Android application. The other 66% were focusing on the *UniSal*, local server supported application.

Most of the users were satisfied with the Web-oriented mobile application, its usability, efficiency and availability referring to the *Web Services* users and of course, *Cloud connection possibility* providing to the application.

VI. CONCLUSION

The *Android UniSal Web application* is divided into two sections. These are calculating the Gross and the Net Salary for the University staff. The application has the functionality that allows the user to update the contributions if they have been changed according to the new Law regulations. The user can change the following contributions: *PDI, health insurance, contributions, personal tax and personal income*. If the user makes mistake inserting the contributions, the application generates error log so that the user can fix the contributions. The application allows the user of to check whether the employee had sick leave in order to calculate the salary. To ensure that the application is working according to the inserted parameters the application has passed couple of tests scenarios - tests for *University staff* gross and net salary calculation, tests for the Setting Panel and the menu of contributions, entering unreal numbers in order to check whether the application will accept the inputs or it will generate error log to the user.

The *mobile web application* presented in this work proposed an original, efficient way of managing the documentation in an institution. Our approach improves the *Quality of Experience (QoE)*; provides the users all necessary resources and performances, which refers to a user-friendly mobile application; web services oriented, evaluated by *QoE*, referring the quality of the *graphical user interface*, *usability*, *efficiency and time saving*.

By using the *mobile cloud computing connection possibility* of this *Android* application, as a future work, more advantages would be provided, such as *flexibility, portability and scalability* that where obstacles of the mobile devices so far. By observing the *QoE* survey, the advantages from using this application have significantly increased users' attention. Possibility of *mobile computing* technology has provided improvement in the process of *Android Web application* development in the direction of increasing the quality of services.

The application presented in this work is developed by using the graphical components provided from the Android API. The key graphical components for this application are the Action Bar and the Fragment implementation. The user does not need to have a special background for salary calculation i.e. the salary is calculated simply by clicking one button. All the necessary information required for calculating the salary is downloaded automatically from the Web Server. In time, the application would require hardware with higher performance, and other APIs to work faster and to process the information. By using the cloud connection, this would not be a problem since the Cloud provider allows using different plans Virtual Machines and APIs for the application. As part of our future work, we are developing an application with Cloud connection. The “Brain Application” which is handling and processing the request is Java EE application and it will be uploaded into the Cloud. By using this model and infrastructure it allows us to enlarge our work in different technologies and creating more complex and more useful options to the user, such as iOS integration, Windows Mobile integration, creating suitable desktop application. Some of the additional options are the following: Employee Management (adding, removing and updating employees), Salary Management (creating, removing and updating the salaries), Salary Calculation for certain employee.

We believe that these Web-oriented configurations of mobile applications will be especially beneficial for universities and their academic and administrative staff, by promoting the advantages from using such an application in terms of saving energy, time, and money.

VII. ADVANTAGES AND FUTURE WORK

The *Android UniSal Web application* could be improved in terms of using *Android 4.0,* which has new features that are still upgrading and new features are coming, such as *NFC* system, which allows the smartphones’ users to unlock cars, pay for parking bills, pay the food into a restaurant and the most common usage is transferring images by just touching the smartphone. With this functionality, this application in the future can be used as a check-in card into some institutions or to calculate the salary by just touching the phone of the employer and the phone of the employee. By using the cloud services in the future, and the proper security issues that *Android 4.0* delivers, the information of the employees can be stored on *cloud database* and the employer can analyze the work of his employees from his phone while he is on his way to work. An interesting *Security* concept for this application given by the unlocking the app with face recognition unlock system that is integrated into *Android 4.0* and higher. The employer who is using the app can set a security lock from his face and he will be the only person who can load the application, calculate the salary or analyze the work of the employee and much more. This mobile application will be easily considered as an adjustable content for *Content Centric Network (CCN)* and *Information Centric Networks (ICN)* approach, aiming to achieve efficient and reliable distribution of the content by providing general platform for communication services which refers to a part of our future work.

REFERENCES

[1] D. Morril, “Announcing the Android 1.0 SDK,” Release 1, Android Developers Blog, 2008, http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html, [Retrieved: May, 2014]

[2] Ice Cream Sandwich (n.d.) In Developers, http://developer.android.com/about/versions/android-4.0-highlights.html, [Retrieved: May, 2014]

[3] P. Alto, “Google’s Android becomes the world’s leading smart phone platform,” Canalys, 2011, http://www.canalys.com/newsroom/google%E2%80%99s- android-becomes-world%E2%80%99s-leading-smart-phone-platform, [Retrieved: May, 2014]

[4] Health Insurance Fund, “Instructions for the manner of calculating contributions for compulsory Health Insurance, according to the amendments of the Health Insurance Law,” 2007.

[5] Ministry of Finance, “Instructions for the implementation of the concept for calculation and payment of the gross salary of the employees in the institutions - budget beneficiaries,” 2009.

[6] Public Revenue Office of the Republic of Macedonia, “Gross salary,” 2013.

[7] Public Revenue Office of the Republic of Macedonia, “Calculation of salary,” 2013.

[8] G. Allen, “Beginning Android 4 Apress,” 2012, http://www.apress.com/9781430239840, [Retrieved: May, 2014]

[9] E. Burnette, “Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers),” Pragmatic Bookshelf: Third Edition, Released: August 4, 2010.

[10] Salary Bot (n.d.) https://play.google.com/store/apps/details?id=salaryCalculator.salaryCalc, [Retrieved: May, 2014]

[11] PayCheck (n.d.) https://play.google.com/store/apps/details?id=salary.calculat, [Retrieved: May, 2014]

[12] Quick Wage (n.d.) https://play.google.com/store/apps/details?id=com.cwesoftware.salary, [Retrieved: May, 2014]

[13] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. M. Sheppard, and Z. Yang, “Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework,” MM '09 Proceedings of the 17th ACM international conference on Multimedia, October, 2009, pp. 481-490, DOI: 10.1145/1631272.1631338.

[14] MySQL Official Site (n.d.) Retrieved from http://www.mysql.com/, [Retrieved: May, 2014]

[15] JSON Official Site (n.d.) Retrieved from http://json.org/, [Retrieved: May, 2014]

[16] PhP Official Site (n.d.) Retrieved from http://www.php.net/, [Retrieved: May, 2014]