An Andriod Smart Phone Web Application for Individual Notification of Exams Dates

Amal Y. Al-Fadhly¹, Oussama H. Hamid²*

¹²Faculty of Computer Studies, Arab Open University
²²P.O. Box 830 Al-Ardia, 92400 Kuwait

Abstract: One practical aspect of ‘open learning’ is that much of the communication between students and their tutors occurs online. Consequently, students contribute actively to their personal education. From this point of view, we aspired to develop a mobile application that facilitates and enhances the receiving of course-related messages by university students over smart phones. Our system comprises three components: (a) Google web portal project, (b) the mobile client application and (c) the web server. Though the application runs currently only on Android mobile phone operating system. Yet, its individuality-based character in sending messages makes it a welcome innovative gesture from students to their peers in the university space.

Keywords: Individuality-Based Systems, Open learning, Android Application, Google Cloud Messaging, E-Organizer.

Introduction

The latest generation of advancements in digital technology enhanced the rapid growth of ‘individuality-based’ software applications. From computerized diagnostic systems in medicine, over tracking people’s purchase habits, to educational applications, individuality-based systems rely on personal experiences and characteristic discrepancies to improve the effectiveness of the system. Particularly in an ‘open learning’ environment, considering individuality is crucial for the overall success of the educational process [1]. This is due to the observation that students stem from different educational and professional backgrounds and get enrolled at the university in different age stages. As a result, various implications emerge, whether in respect to the delivery of the learning material or to the organization of students’ assessments.

One practical aspect of open learning is that most of the communication between students and their tutors occurs online. The Arab Open University (AOU) is an academic institution that adopts a trade-off between a traditional university education and a complete open learning paradigm. From this point, we aspired to develop a mobile application that facilitates and enhances the process of receiving university messages in such an attractive and simple way over smart phones.

An academic institution that relies on ‘open learning’ as an educational paradigm, deploy the integration of digital technologies and online-based learning within classical face-to-face classrooms [2, 3, 4]. Affectedly, students contribute actively to their personal education, as a consequence of personal commitment and accountability, which in turn facilitates innovative improvement of the overall implementation of the educational system. For example, in terms of the dates of exams, the department of examination affairs at AOU posts on the Learning Management System (LMS), which is a platform for the administration, documentation and tracking of electronically provided learning material [5, 6], one schedule that contains exams’ dates for all courses. Students are then informed per e-mail about the posting of this schedule on the LMS.

However, it is the students’ responsibility to check the concrete dates of their exams on the LMS. All this is done online and there is no need for a tutor to inform students about these dates in person. This could be problematic, if for some reason the schedule was updated. Though, the department of the examination affairs timely informs the students about possible updates, the process of checking the updates personally could be troublesome for some students.

To help better manage this problematic issue, we propose a solution that complements the LMS and at the same time directly targets each student individually. Our solution implements an application, termed as ‘E-Organizer’ that notifies every student about the dates of their exams. Students can install the E-Organizer on their Android devices (e.g. mobile phones). Accessing the database that contains students’ course-related information is achieved via Google Cloud Messaging (GCM) [7,8]. This is a free service that allows the transfer of data between the database and the E-Organizer. The GCM service registers each mobile client by assigning it a unique registration code and sends it back to the mobile client application, i.e., the E-Organizer. The mobile client application sends the registration code to the web server (Fig. 1).
When the exam time is due, the LMS administrator sends a message associated with the student registration code to Google along with the student’s exams schedule and other relevant information. Subsequently, the GCM service broadcasts this message by sending it to the student individually. The message will be displayed in the mobile notification area. This procedure can be generalized to include cut-off dates of other assessments, such as the Tutor Marking Assessment (TMA) and other course-related events.

**Figure 1:** The interaction between the system’s components (schematic)

---

### Description of the System

Our devised system is used mainly by students, the department of examination affairs at any university with an ‘open learning’ educational paradigm, and the corresponding staff of the university (e.g. the LMS administrator). Accordingly, the system is divided into technically separated components that better meet the different levels of its clients’ perspectives.

The following subsections provide detailed description of the three components:(a) Google web portal project, (b) the mobile client application (application) and (c) the web server parts.

#### A. Google Web Portal Project

Creating a Google Application Programming Interface (API) project gives developers ability to own a unique project ID, server and browser API Keys [9]. By this Google API, developers can have a set of different services that they may use in order to meet the requirements of their project. Some of the services are charged and others are free. Enabling the free GCM can be done through Google API project. GCM is free service provided by Google. The main role of the GCM service in our system is to send push notification from the GCM server to every student’s device (i.e. E-Organizer). And this is done on the basis of a unique code that has been assigned to each student during the registration into the web server (registration code). Moreover, the GCM server handles queuing and storing messages along the way to delivering them to the registered Android devices.

#### B. Mobile Client Application

Students can install the E-Organizer on their Android devices. First, the application displays a registration form, which should be filled out by a student. Currently, the form requires only the student’s university-related ID number and her/his courses in the present semester. Once registered (by pressing the corresponding button), the application connects to Google free portal web project and registers for receiving broadcasted GCM. Subsequently, the E-Organizer receives a unique registration code, marking the individual identification of this student by the system. As a result, the registration code is combined with a record that contains the student’s name, student’s ID number, her/his registered courses in the current semester, are sent to the web server to be stored in an existing database for the purpose of later announcement.

Once done, an acknowledgment message returns from the web server back to the student’s E-Organizer. At this point, individual exams’ information will be forwarded to the GCM, as soon as the schedules are created by the web administrator. Once the web server forwards this information to the GCM, the GCM broadcasts the corresponding messages to all registered students to be displayed in the notification area of the E-Organizer(Fig.2). The application permits students to set a reminder for each message they have received. Independently, the E-Organizer will display a reminder message to the student even if it has been sent into the background. Of course, students may always delete received messages by clicking on the delete icon (button) in the application (Fig. 3).

For implementing connotation between the mobile phone and the server, students’ information was submitted from the mobile phone to the server via aRepresentational State Transfer (REST) API [10, 11]. This API architecture is simpler in the implementation and use than its alternatives, such asSimple Object Access protocol (SOAP) and Web Services Description Language (WSDL)-based web services [12, 13].

Note, for the Mobile Client Application to function, it is necessary the mobile phone has a working internet connection in order to register at Google web portal and receive the corresponding registration code. Also, the E-Organizer must be running on the student’s mobile phone in order to receive messages.
Figure 2: The message is appearing in notification area via Galaxy Note3

Figure 3: The use-case of the student (user of E-Organizer)
**Web Server**

All Students’ information (this regards only registered users of the E-Organizer) would be received by the web server along with their own Google registration codes. The web server stores, then, this information into a database that can be accessed only by the web server. Subsequently, the web server links the individual student’s ID along with other relevant information in the database in order to generate a full record about this student. When the examination time is due, the web server generates separate exams’ schedules to all registered students, taking into account individual configurations of courses’ varieties between students. This is guaranteed by appropriate search algorithms within the database. It is worth mentioning that other issues such as TMAs cut-off dates and course-related events could also be handled in similar way.

For the REST API of the Mobile Client Application to be ultimately utilized, a Hypertext Transfer Protocol (HTTP) request, containing the student’s relevant information which is sent from the mobile phone, is used [14].

Finally, broadcasting of individual exams’ notifications occurs immediately after a correct matching of students’ IDs and corresponding information in the database.

Importantly, for the web server to function as intended, it must be published over the Internet and have a valid URL. Moreover, the database must contain all registered students’ applicable information (Fig. 4).

![Figure 4: The use-case of the web server](image-url)
The Proposed System Architecture

1. Mobile Application send request to Google Cloud Messaging (GCM) for a unique Registration-ID, after installed the application into Android device.
2. Google Cloud Messaging (GCM) registers mobile and return with return each mobile with its unique Registration-ID.
3. Mobile application send Registration-ID and Student's information to web server in order to identify it.
4. Web Server Store the Registration-ID and information into its database.
5. Web Server gets Registration-ID from database and send request to GCM along with the message, requesting GCM to push notification to each mobile application.
6. Google Cloud Messaging (GCM) send message to all mobile application.
Figure 6: The data flow diagram of the system
Advantages and Disadvantages of the System

The main advantages and disadvantages of the system are as follow: 1) it is attractive and has simple design, 2) it is understandable with user-friendly interface, 3) it’s flexible: department and subjects can be changed as necessary and 4) it has the ability to set a reminder and delete messages easily.

The disadvantages may be summarized as follows: 1) it requires installation and networking, 2) the application still doesn’t support multi languages, application’s theme and/or font’s type or size, and most importantly, 3) unlike the learning management system (LMS), no direct connection between students and tutors is possible.

Conclusions and Future Work

As described above system the application is built using Java language for mobile side as programming language. Eclipse is an integrated development environment (IDE), Eclipse were used to designing the application interface such as screens and menus, and also used it for programming the application logic to control different screens of the application [15]. Here are the interface of the application clearly shows the application’s screens such as Registration form screen, list of messages application already revised and finally the reminder screen. On the Web Server side, Eclipse were also used to program, and using Apache Tomcat as web server. Please see below for these results. (Fig 7,8).

The long-term goals of the system is to publish web server on real server such as a university’s server, allowing web server to send other types of messages or events such sending an image as invitation card for some events. Developing the application into MAC version is another important goal to spread the application more broadly among students.

Figure 7: E-Organizer as appears on Galaxy Note3
Figure 8: An example of LMS’s Admin page of sending exam's messages

Authors’ Biographies

Amal Y. Al-Fadhly received a Dual Bachelor degree of Science, Information Technology and Computing (ITC) from the Arab Open University (Kuwait, 2015) and the Open University (OU) with first class Honors Programme – (United Kingdom, 2015). Her research interests comprise mainly the design and the programming of Android mobile web applications.

Oussama H. Hamid received in 2011 a Ph.D. degree in Natural Sciences from the University of Magdeburg, Germany. The title of his doctoral thesis was: “On the Role of Temporal Context in Human Reinforcement Learning”. Before, he was awarded a Master's degree in Computational Visualistics (Germany, 2002) and Bachelor degrees in Computer Science (Germany, 2000) and Mathematics (Germany, 1998). In the years between 2003 and 2015, Dr. Hamid has been investigating both theoretical and practical aspects of Human-Machine-Interaction, Neural Network Functions, Cognitive Systems and Big Data Analysis. Dr. Hamid has published his research findings in several European and international refereed journals and conferences. Currently, Dr. Hamid is an Assistant Professor at the Faculty of Computer Studies at the Arab Open University in Kuwait.

References


[7]. Penghui Li; Yan Chen; Taoying Li; Renyuan Wang; Junxiong Sun, "Implementation of Cloud Messaging System Based on GCM Service," Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on , vol., no., pp.1509,1512, 21-23 June 2013.


[10]. Richardson, Leonard; Sam Ruby (2007), RESTful web service, O'Reilly Media,ISBN 978-0-596-52926-0, retrieved 18 January 2011, The main topic of this book is the web service architectures which can be considered RESTful: those which get a good score when judged on the criteria set forth in Roy Fielding's dissertation.


[13]. "Web Services Definition Language (WSDL) Version 2.0 Part 1: Core Language".
