

ON ANNOUNCEMENT FOR UNIVERSITY WHITEBOARD USING MOBILE APPLICATION

Yaser A. Jasim¹, Mustafa O. Alsaigh², Tawfeeq M. Flaih³, Mustafa Ghanem Saeed⁴

Accounting Department, College of Administrative and Financial Science, Cihan University-Erbil, Erbil1

Business Administration Department Cihan University, Slimani2

Software Engineering Department University of Mosul3

Computer science Department College of Science Cihan University, Slimani4

yaser.jasim@cihanuniversity.edu.iq¹, mustafa.alsaigh@sulicihan.edu.krd²,

tawfeeq.m.flaih@uomosul.edu.iq³, mustafa.swe@gmail.com⁴

Abstract

The university students who use traditional announcement boards nowadays are exposed to various problems. The most severe problem is that the instructors may want to publish an announcement, but they may not exist in university, or maybe they are after a work time. So, they cannot or can inform to the announcement board moment by moment. Moreover, the students should be near the announcement board all the time. The scope of this research is to build a unique mobile application that acts as a digital university announcement board. Where this application mainly consists of two sides; the server-side (PHP with MySQL) and client-side Android Applications. Through using the proposed application, the university staff can send announcements to students directly, and the announcement will be delivered to student devices in real-time with push notification. Instructors will be able to announce one student or a group of students or all the students from different departments and stages. The suggested application was built and tested in the android operating system and the real mobile device.

Keywords: *Mobile Application, Android Volley, Real-time notification, Firebase.*

I. INTRODUCTION

For more than a century, mobile computing has attracted the attention of the research community and has also reached the business sector and mainstream customers via smartphones and PDAs. Mobile computing enhancements in hardware such as (better processing power, more fully wireless network bandwidth) also enhance mobile device capabilities [1]. There are similarities in several facets between Mobile Applications and other Software Applications such as [2]:

1. Security: protection of different devices (smartphones, tablets, laptops, other portable devices).
2. Performance: load uptime of applications in devices.
3. Storage limitation: the fixed size of memory.

There are some variations concerning these similarities, such as communication with other applications. Mobile applications are multi-sources, which means that one mobile may have a significant amount of applications, and communication between them is possible. Sensor handling: contemporary mobile devices such as (smartphones) have an accelerometer that will be impacted by the device's motion. There is also a touch screen on mobile devices. It has got a multi-job process, a virtual keyboard, a microphone, linked to the internet with GPS scope as well as back and front camera.

Furthermore, Flexibility of the code for different O.S versions: unlike other software's mobile applications are code flexible. This situation means that it works on almost any mobile device even if the operating system of those phones is different [2]. Android is a platform used to create applications in Mobile application developing. Where the application is also a Linux-based open-source Operating System used for mobile devices such as smartphones, TVs, smartwatches, and

tablets. The suggested app uses the Android platform because users of mobiles widely use it, and it is open-source [1][2].

Announcement boards are used by all universities and most organizations and businesses and exposed to multiple issues. The most severe issues are like informing learners in real-time, so the student should always be close to the advertising board. As well as sometimes teachers may want to post an announcement after a job, or they may not be able to post an announcement from college. For all those reasons the suggested mobile application was built to computerize the traditional way of whiteboard announcements to make the student receive an announcement from anywhere. Since he has a mobile application and an internet connection, then the instructor can announce the students anytime or anywhere (he/she) wants, besides, to Reduce time and paper works. Recently, technology has a very significant and vital role in improving the quality of services provided by the business [13]. According to, the enormous quantity of stored data in databases, repositories, and files, it is increasingly significant and necessary, to develop robust approaches to analyze and exponent of such data and to extract motivated knowledge that could provide the decision-making [14].

2. LITERATURE REVIEWS

Android is a software founded in Palo Alto of California in the year 2003, and it is a Linux based operating system [5][6]. It is designed primarily for touch screens mobile devices such as smartphones and tablet computers. It is one of the most commonly used operating systems for mobile devices these days. In the last (15) years, this operating system has been developed massively, since the invention of black and white phones through recent smartphones or mini computers. The android operating system is so powerful, and it supports a large number of applications in Smart-phones. The based hardware that supports android software is on the ARM architecture platform, which is a family of architectures for computer processors that employ reduced instruction set computing (RISC) [3]. Android is an open-source operating system that means that it is free and anyone can use it. Also, android contains millions of applications that are available at low cost or free in the market, and for these reasons, android is very popular [5]. The android development backups with the full java programming language, other packages that are API and JSE are not supported.

A. Android Technology

Application components are the essential keys to the Android application. These components follow the loosely-coupled architecture by the application manifest file *AndroidManifest.xml* that depicts each component of the application and how they interact; there are four main components in android [7]:

1. Activities: It is a platform that appears to the user which handles the user interaction to the smartphone screen, and holds the UI components.
2. Services: They control a background process associated with any application.
3. Broadcast Receivers: They manage communication among the Android operating system and the applications.
4. Content Providers: They control the management issues of data and databases.

The suggested application uses some Android Technology: Servers, *AsyncTask*, permissions, Volley, Shared-Preferences, Firebase Cloud Messaging, and Firebase-Messaging-Service. These techniques are described in the following paragraphs representing the client-side of the suggested application.

B. Android Service

Android service is a component of an application that has the ability to perform long-running operations in the background, and it does not supply a user interface. In general, there are three various kinds of services in Android: Foreground, Background, and Bound [7].

However, in practical implementations, there are two primary states in android service, which are started and bound. Moreover, starting a service is when an application component, such as an activity, begins by calling the function *startService()*. Once it has started the task, then a service process can run in the background, even if they started component is terminated. However, service is bounded if an application component binds to it by calling the function *bindService()*.

A bound service provides a client-server interface that gives allowance for components to interact with the service, send requests, get results, and even do so across processes with inter-process communication (IPC) [8].

C. AsyncTask

Android defines *AsyncTask* as "a class that supply expansion to the *Object* class for allowing short processes to run in the background asynchronously" with "doing background" and "onPostExecute," *Async* can run tasks asynchronously on new threads [2].

Asynchronous tasks use [2]:

1. Params, sending the parameters to an executing task.
2. Progress, progress units are published through the computation of background.
3. The result, the process of resulting in the background computation.

Using *AsyncTask* is also for short-running tasks such as connection to the network.

D. Android permissions

Android contains a system that offers permission and predefined permissions for specific tasks. Where any application can request permission. The cause of permission is to guard the privacy of an Android user. Android applications must ask permission to approach sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet) [5]. Depending on the feature, the system might give permission automatically or might induce the user to accept the request. An application must announce the permissions it requires by containing `<uses-permission>` tags in the app manifest [6]. For instance, an application that wants to use the internet must have this line in the manifest [5][6][7]. There are two kinds of permissions (Normal, Dangerous); Normal permissions are those who are deemed harmless for the user's privacy or the operation of other applications. Where granting Normal permissions is done automatically to the application [6].

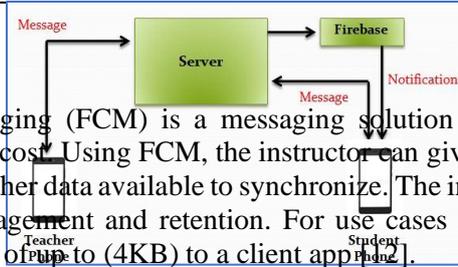
E. Android Volley

Volley is a library of HTTP that constructs the network for Android applications in a more comfortable and faster manner. To add Volley to an App should add the dependency to the app's build. Gradle file.

F. Shared-Preferences

Context.getSharedPreferences (String, int) returned interface for accessing and modifying preference data. There is a single instance of this class for any particular set of preferences that all clients share. Changes to preferences must pass through an Editor object in order to guarantee that preferential values stay in a coherent state and control when committed to storage. Returned objects are from the different methods of obtaining that the application must treat as immutable. Since all operations go through in-memory first, this is guaranteed to return the latest value [10].

G. Firebase Cloud Messaging



Firestore Cloud Messaging (FCM) is a messaging solution cross-platform that lets you reliably deliver messages at no cost. Using FCM, the instructor can give a notification to the client's application by a new email or other data available to synchronize. The instructor can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to (4KB) to a client application.

FirebaseMessagingService is an object class that extends *EnhancedIntentServiceBase* object class to receive messages from Firestore Cloud Messaging. Extending this object class is required to be able to handle downstream messages. It also provides the functionality to display notifications automatically and has ways that are invoked in giving the attitude of upstream messages. Override base class methods to handle any events required by the application. FCM Token or much widely known as a registration Token is an ID created by the GCM servers employed for connections to the client application that allows it to receive messages [11]. The fastest way is to store it in *sharedPrefs* and add this logic to the *onCreate* method in his Main Activity or class, which is extending *Application* [13].

H. Server-Side Technology

The server is a computer code written by a programmer or a device that supplies functionality for other codes or devices, called "clients." This architecture is known as the client-server model, and a single overall distributed computation is multiple processes or devices. Servers can offer multiple functionalities, usually known as "services," such as sharing data or resources with multiple users or performing the computation for a user. A single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device [4].

The suggested application uses some Server Technology: MySQL Database [3], PHP, and JSON. For the increase suggested application security, all PHP APIs used *mysqli_real_escape_string()*. While this function Escapes special characters in a string that used in an SQL statement which a very effected way to avoid SQL injection [6].

3. METHODOLOGY RESEARCH

This section illustrates the general structure of the research, which includes modeling, designing, and implementation of the Student Whiteboard Announcement Using Mobile Application. Moreover, it demonstrates how to connect an Android mobile application with a remote server and exchange data between them.

A. Whiteboard Application Requirements

The basic idea is that the instructor sends an announcement to the server, and then the student gets an announcement from the server and also notification from firestore, as shown in figure (1).

Figure (1) Sending an announcement to a student through the server

From the instructor perspective, Firstly, the instructor must login to the system then they have four options to use as following:

1. The first option is sending a message to broadcast an announcement; here, the instructor must choose the department and stage of the student and then select the target student. The target student can be one or more.
2. Outbox option; the instructor can observe the sent messages announcements.
3. Change password is the third option; the instructor (or student) can change (his/her) password.
4. The final option is to logout the system, as explained in figure (2) Use case diagram for Whiteboard Application.

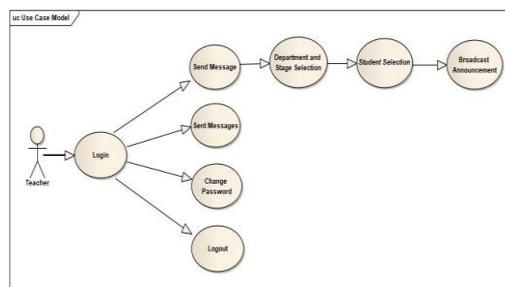


Figure (2) Use case Model for Whiteboard Instructors

From the student perspective, at the beginning the student must login to the system, After that they can use three main option, first option is inbox; to view the incoming announcements, change password is the second option where the student can change (his/her) password, final option is logout the system, as explained in figure (3) Use case diagram.

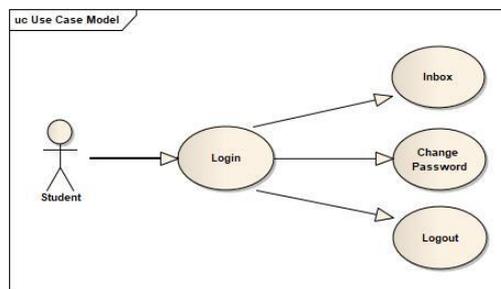


Figure (3) Use case Model for Whiteboard Students

According to the server, it contains MySQL database and PHP API. Therefore, the server is connected to the firebase server and the user's mobile, as explained in figure (4) Use case diagram.

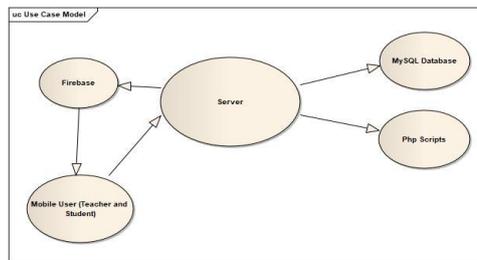


Figure (4) Use case Model of Whiteboard Server

B. Whiteboard Server Side

The server contains a MySQL database and PHP API. While, PHP is used to (connect firebase to the server, connect mobile users to the server, process the data), as explained in figure (5). Besides that, MySQL is used to (create database and tables, Store information and data). The database consists of four tables which are (login, lecturer, student, message), and the relationship between login and both student table and lecturer table is a (one-to-one) relationship while the relationship from student and lecturer table to message table is (one-to-many), as explained in figure (6).

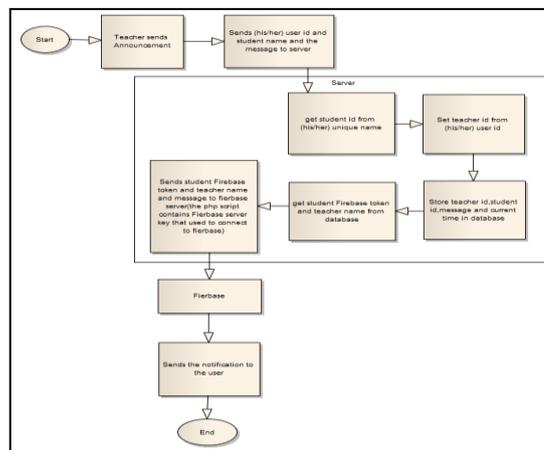


Figure (5) Processing Data inside the Server Using PHP

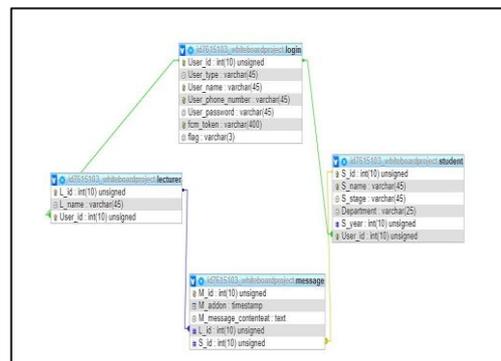


Figure (3-6) Database Implementation

C. Whiteboard Client Side

The Whiteboard client-side in general has two types of user which is instructor and student:

1. Instructor:
 - a) asynchronous task class: used to send an HTTP request to the server.
 - b) Login activity: check if the user exists in the database or not, as explained in figure (7).

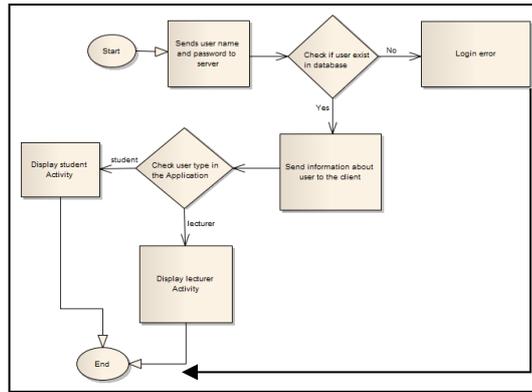


Figure (7) Login Activity

- c) Shared preference class: to store user information into the application that the user will login to until they will logout, as explained in figure (8).

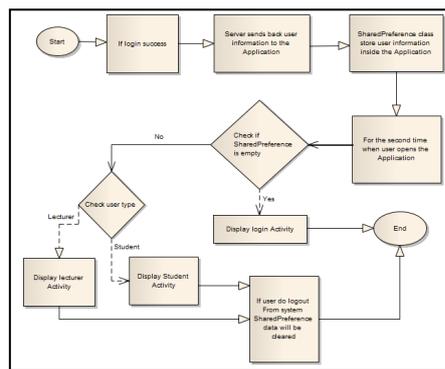


Figure (8) SharedPreference Class

- d) Home activity: here, the instructor can give an announcement to students, view the sent announcements, change his/her password, logout from the system.
 e) Choose department and stage activity: in this procedure, the instructor must select the student's department and stage, as explained in figure (9).

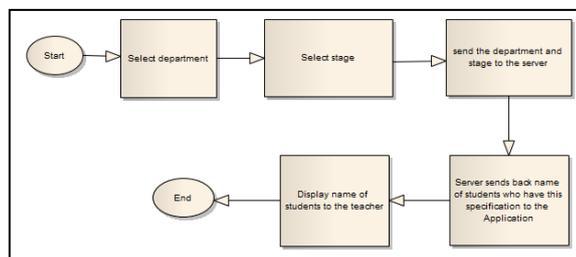


Figure (9) Choose department and stage activity

- f) Send announcement activity: in this activity, the instructor will type his/her announcement and select the target student, which can be one student or more, as explained in figure (10).

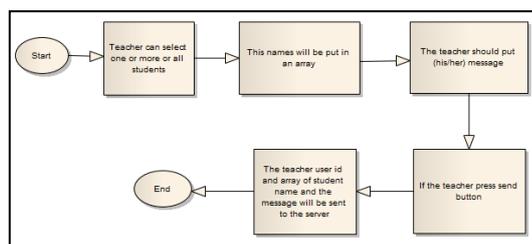


Figure (10) Send announcement activity

g) Sent announcement activity: according to this process, the instructors can view the announcements which are sent by themselves to students, as explained in figure (11).

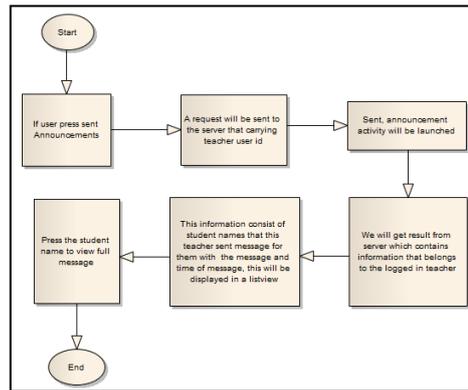


Figure (11) Sent announcement activity

h) Change password activity: here, this procedure requires to insert the old password then a new password by the user in order to change the user’s password, as explained in figure (12).

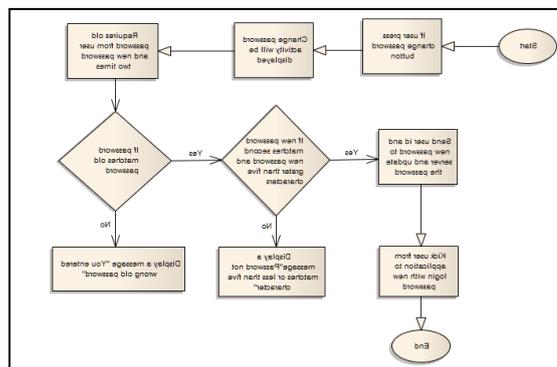


Figure (12) Change password activity.

2. Student:

1. asynchronous task class: used to send an HTTP request to the server.
2. Login activity: to check if the user exists in the database or not, as explained in figure (13).

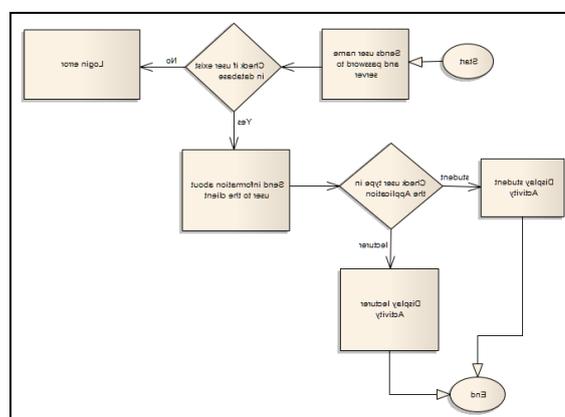


Figure (13) Login Activity

3. Shared preferences class: to store user information into the application that users will be logged in until they logout, as explained in figure (14).

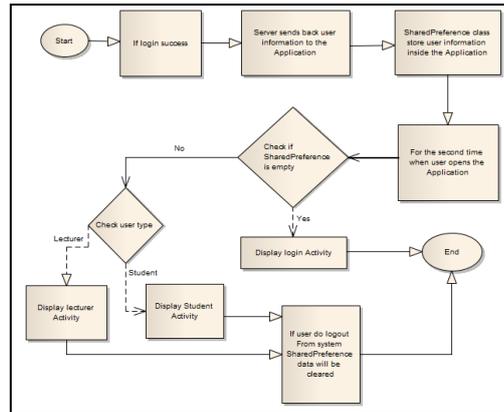


Figure (14) Sharedpreference Class

4. Home activity: from here, the student can view incoming announcements, change his/her password, logout from the system.
5. Inbox activity: in this activity, the student can view the announcements which were sent via the instructors, as explained in figure (15).

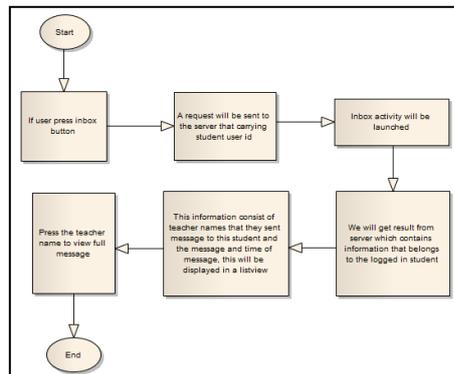


Figure (15) Inbox activity

6. Change password activity: this procedure requires inserting the old password then a new password by the user in order to change password, as explained in figure (16).

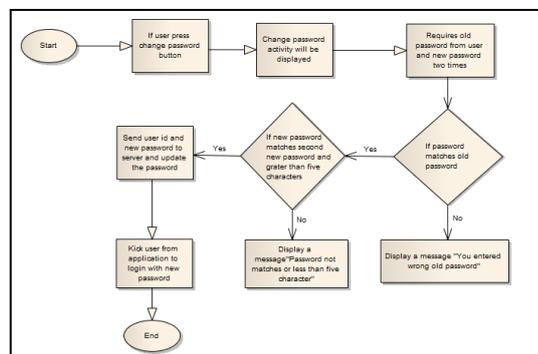


Figure (16) Change password activity.

7. Firebase messaging service: this service is responsible for handling the data passed by firebase to the application to initiate a notification and display it to students when they receive a new announcement from instructors, as explained in figure (17) and figure (18).

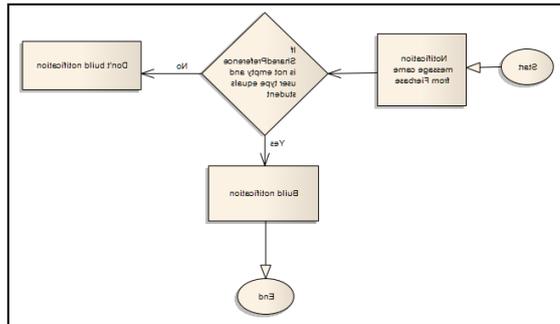


Figure (17) Firebase messaging service

```

public class MyFirebaseMessagingService extends FirebaseMessagingService {
    boolean flag=false;
    @Override
    public void onMessageReceived(@RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);
    }
    if (SharedPreferences.getInstance(this).isLoggedIn() && SharedPreferences.getInstance(this).getUserType().matches("student"))
    {flag=true}
    Intent intent=new Intent(getApplicationContext(), MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent pendingIntent=PendingIntent.getActivity(this, requestCode, intent, PendingIntent.FLAG_ONE_SHOT);

    if (flag=true) {
        if (remoteMessage.getNotification() != null) {
            String title = remoteMessage.getNotification().getTitle();
            String body = remoteMessage.getNotification().getBody();
            Uri alarmSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
            NotificationCompat.Builder mBuilder =
                new NotificationCompat.Builder(getApplicationContext(), MainActivity.CHANNEL_ID)
                    .setSmallIcon(R.mipmap.ic_launcher)
                    .setContentTitle(title)
                    .setContentText(body)
                    .setSound(alarmSound)
                    .setLights(Color.BLUE, 500, 500)
                    .setContentIntent(pendingIntent)
                    .setAutoCancel(true)
                    .setPriority(NotificationCompat.PRIORITY_HIGH);
            Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
            v.vibrate(1000);
            NotificationManagerCompat notificationMgr = NotificationManagerCompat.from(this);
            notificationMgr.notify(1, mBuilder.build());
        }
    }
}
    
```

Figure (18) example of Firebase messaging service java code

D. Whiteboard Firebase

Besides, firebase will receive the name of the instructor, the announcement, and the student device unique token which is sent to it via the server. After that, it will send instructor name and the announcement to the target phone known by its token, and then in the student phone, the firebase messaging service will achieve its primary function and display the instructor name and their announcement to let the student know that there is a new announcement, as explained in figure (19) and figure (20).

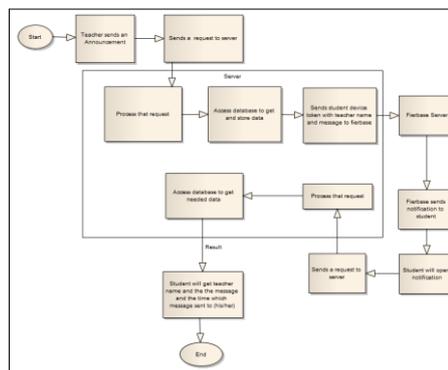


Figure (19) Sending notification through firebase to the student.

```

<code>
</code>

```

Figure (20-A) example of Sending notification through firebase to student PHP code

```

<code>
</code>

```

Figure (20-B) example of Sending notification through firebase to student PHP code

```

<code>
</code>

```

Figure (20-C) example of Sending notification through firebase to student PHP code

IV. Whiteboard implementation and testing

The Whiteboard has two different types of GUI for both instructor and student which contains several sections such as (inbox, send a message and sent a message, change password) as shown in figure (21), and each part is responsible for a particular job.

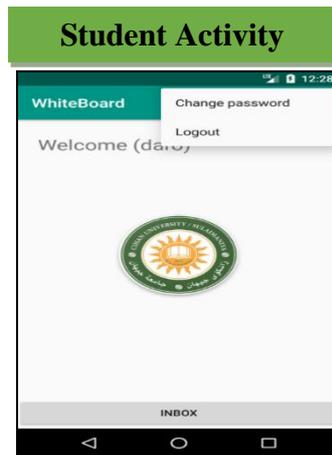


Figure (21-A) Whiteboard user interface

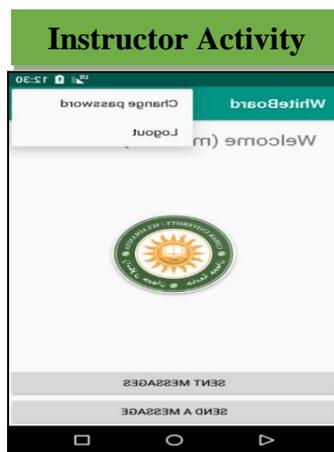


Figure (21-B) Whiteboard user interface

When a user downloads the Application, they have to login to the system; first, the users must get their username and password from the administrator, as shown in figure (22).

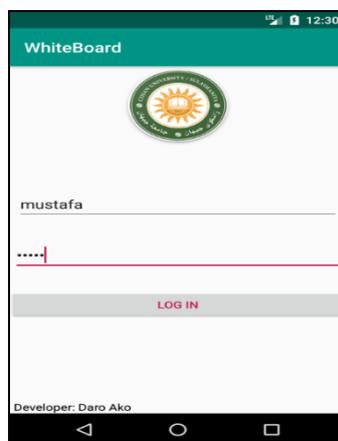


Figure (22) User Login

Now the user can use the features and options of the application after logging in, for example, the instructor can send an announcement to students as shown in figure (23) after selecting the department and stage. Therefore, the instructor will put his/her message and select the target student and send the announcement, as shown in figure (24).

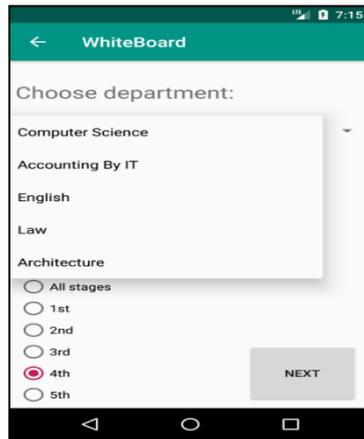


Figure (23) Choosing department and stage.

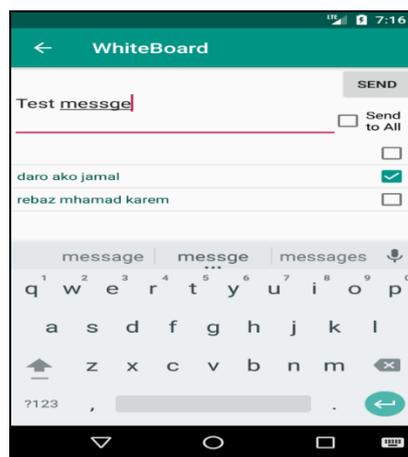


Figure (24) Choosing student and typing message

Also, the sent announcement was made by the instructor he can view all the announcements that had been sent as shown in figure (25), and by enabling one of the items in the list he/she can view the full message as shown in figure (26).

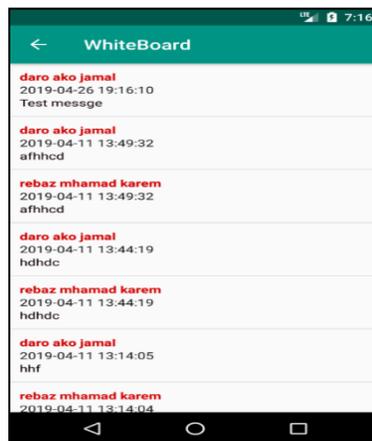


Figure (25) Sent announcements.

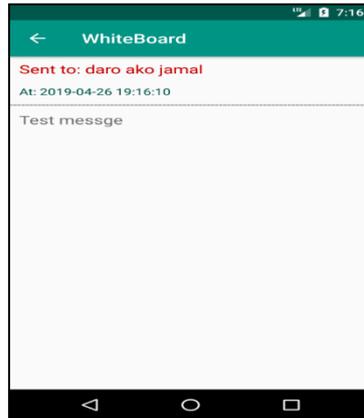


Figure (26) Information about sent announcement

After completing the previous procedures, the student checks his/her inbox to be able of seeing the income announcement from instructors as shown in figure (27), and by activating one of the items in the list he/she can view the full message as shown in figure (28).



Figure (27) Incoming announcements

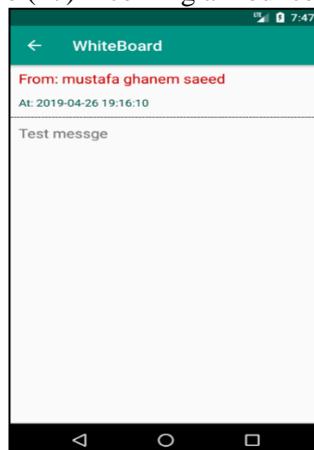


Figure (28) Information about incoming announcement

Moreover, there is also the Change password option in which both students and the instructor can change their password from here, as shown in figure (29).

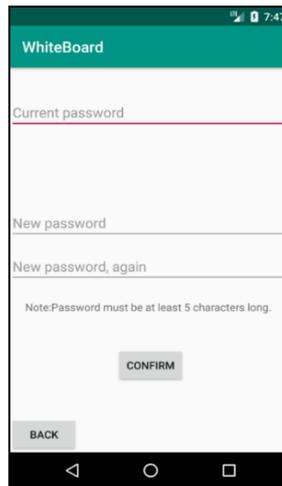


Figure (29) Changing password

4. CONCLUSIONS

Overall, the experimental findings depict that this android application which is connected to MySQL database, PHP (hypertext preprocessor) and FCM (firebase cloud messaging) can be used to deliver a message with a push notification from source to destination in real-time. Despite having any security issues. Where the only problem faced is concerned with firebase, the firebase generates only one token for each device that means if two users employ the same device then the first user receives a message with notification but the second user will receive the only message without push notifications. The cause of that is where the token is registered for the first user only. This problem will be solved by uninstalling the application and reinstalling it again; now when the second user does login to the system, the firebase registers the token of the second user. Therefore, the second user will receive a message with push notifications.

REFERENCES

- [1] Holzer, A., & Ondrus, J., (2011), "Mobile application market: A developer's perspective", *Telematics and informatics*, 28(1), pp. 22-31.
- [2] Wasserman, A. I., (2010), "Software engineering issues for mobile application development," In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pp. 397-400, ACM.
- [3] Joshi Vaibhav Vijay and Balbhim Bansode, (2015), "Arm Processor Architecture", *International Journal of Science, Engineering and Technology Research (IJSTR)*, Vol. 4, Issue 10.
- [4] Connolly, R., (2015), "Fundamentals of web development," Pearson Education.
- [5] Burton, M., & Felker, D., (2015), "Android App Development for Dummies," John Wiley & Sons.
- [6] Hello, A., (2015), "Introducing Google's Mobile Development Platform," Ed Burnette.
- [7] Rogers, R., Lombardo, J., Mednieks, Z., & Meike, B., (2009), "Android application development: Programming with the Google SDK," O'Reilly Media, Inc.
- [8] Website article, Android – Services, https://www.tutorialspoint.com/android/android_services.htm, 18 February 2019 – 08:30 PM.
- [9] Slobodan Aleksandrov and Saša Vulović, (2018), "Creating an Android Weather Forecast Application in the Android Studio", 7th International Scientific Conference Technics and Informatics in Education, Serbia.

- [10] Tegawende F. Bissyande', Mike Papadakis, Siegfried Rasthofer, Alexandre Bartel, Damien Oceau, Jacques Klein and Li Li, (2017), "Static Analysis of Android Apps: A Systematic Literature Review", *Information and Software Technology*
- [11] Sangwon Hyun, Junsung Cho, Geumhwan Cho and Hyoungshick Kim, (2018), "Design and Analysis of Push Notification-Based Malware on Android", *Security and Communication Networks*, Hindawi.
- [12] Nilay Ganatra and Rachana Patel, (2016), "Google Cloud Messaging (Gcm): A Lightweight Communication Mechanism Between Client and Server on Android Platform", *International Journal of Information Sciences and Techniques (IJIST)*, Vol. 6, No. 2.
- [13] Yaser A. Jasim, (2016), "Data Modelling for Sw-Commerce Concept," *Journal of Global Business and Social Entrepreneurship (GBSE)*, Vol. 3, no. 5, page 187–194.
- [14] Yaser A. Jasim, Mustafa G. Saeed, (2018), "Developing a Software for Diagnosing Heart Disease via Data Mining Techniques," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal Regular Issue*, Vol. 7, No. 3, pp. 99-114.