5.3.3 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) [24] is a platform messaging service that can deliver messages securely, free of charge.

Cross-platform notifications and updates for Android, iOS and web applications, which are actually available for free, are supported by Firebase Cloud Messaging, which has formerly been known as the FCM, Google Cloud Messaging (GCM).

We will alert the consumer device with FCM of the availability of new email or other data for sync. We will submit app re-engagement and maintenance update notifications. A notification will pass payloads of up to 4 KB to a client device for use cases like instant messaging.

The following figure [figure 12] demonstrate the working principle of Firebase Cloud Messaging.

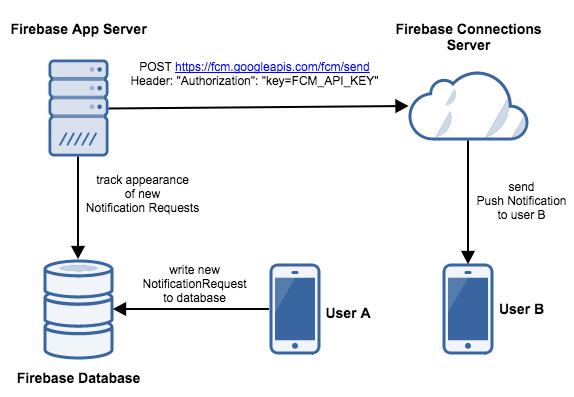


Figure 12: FCM

[https://blog.techmagic.co/firebase-cloud-messaging-for-push-notifications/]

A broad range of messaging and functionality are enabled for Firebase Cloud Communications (FCM). Message types with FCM can send two types of messages to clients:

* Messages for notification, sometimes called ' display messages, ' which are automatically dealt by FCM SDK.
* Messages with data are handled by the client’s app.

Notification messages provide a number of user-visible keys those are predefined. By contrast, the data messages include the custom key value pairs, which is defined by the user. Additional data payload may be included in notification messages. For both types of messages, the maximum payload is 4 KB unless the Firebase console sends messages, for which the 1024 characters limit has been enforced.

5.3.4 Server Side Scripting

The server side scripts are written in php language and uploaded to the server. The following files are included in the parent directory.

* DoctorMessage.php
* PatientMessage.php
* gettingmessage.php
* gettingnews.php
* newspostfromapp.php
* push\_notification.php
* push\_notification\_to\_doctor.php
* push\_notification\_to\_patient.php
* register.php[zahed.ml/fcm/register.php]

**Data Uploading and Storing:**  There is a button in the home activity in the patient’s Android Application (PatientApp) called “*upload the patient health status*”. If we click on the button then it will go to a new activity where we see three text fields 1. Enter the patient name 2. Enter the temperature (Centigrade) 3. Enter the Heat Rate (bpm). After filling the text fields we click on the button called “submit information” to upload the data to the server. The url [*http://www.zahed.ml/api/newspostfromapp.php*] link is used to in the Android Application program. A request is sent with data to the server in that link. The *newspostfromapp.php* file is stored in the server which catch the data that are sent by the Andriod Application. This php file is responsible to store the data in the server database. There is a table in the database named mytab. The columns of the table are title, Temperature, HeartRate, time. This four parameters are sent by the PatientApp. The above php file store the data in the mytab table.

**Notification Sending to Doctor:**  The *push\_notification.php* file is included in the *newspostfromapp.php*. The *push\_notification.php* is called when the temperature is in abnormal state. There is a “if condition” in the *newspostfromapp.php* which check the uploaded the is normal or not. If not, then the push\_notification.php file is called. Then the push\_notification.php send a request to the Firebase Cloud Messaging (FCM) to send a notification to the doctor’s application (DoctorApp) with patients abnormal data which is taken from the mytab table. To send a request to FCM we need a FCM server key and the server need a FCM token key of the doctor’s application which is unique. We get the FCM server key and FCM token key by creating project in https://console.firebase.google.com/ and adding the doctor application to the project. When the push\_notification.php send a request to the FCM server, then the FCM server send a notification to the DoctorApp using the FCM token of the DoctorApp which is generated by the FCM. When DoctorApp get the notification with message from the FCM server, There is text fields to show the server notification message in the DoctorApp.

**Messaging from Doctor to Patient:** After getting the notification from the FCM server, If the doctor has any query or suggestion, then the doctor can send a message to the patient. To complete this task, there is an option “*send message to the patient*” in the home activity of the DoctorApp. After clicking that option, it leads to a new activity where doctor can write his/her query/message and clicks on the “*submit information”* button to send POST request to the server, then the server receives the request and stores the message in the database and forward the message to FCM server, the FCM server sends the query/message to the PatientApp in the JSON format. To perform the above task weed two php files. *DoctorMessage.php* catches the message sent by the doctor and stores the message in table named Message. The columns of the Message table are name, message, date. This php file includes an another php file called *push\_notification\_to\_patient.php*. This *push\_notification\_to\_patient.php* file sends a request with the doctor’s message to FCM server to send a notification message to the PatientApp. FCM server key and FCM token key for the PatientApp is provided in the request.

**Previous Message Between Doctor and Patient:** In both the PatientApp and the DoctorApp has an option “show previous messages” to see the previous messages between the doctor and patient. If he/she clicks on the option, then an activity shows all messages. To do this we need a php file named *gettingmessage.php*. When we clicked on that option then both PatientApp and DoctorApp send a GET request the server to get the messages. The gettingmessage.php receives the request and retrieves all the messages from the Messages table of the database and sends that messages to the DoctorApp and the PatientApp in JSON format. After getting messages from the server in JSON format, the processes the JSON format data and shows in an activity.

**Previous Health Status:** The doctor and the patient can see the previous health status of the patient. We shows the patient’s health status in the website *zahed.ml* that are uploaded by the PatientApp. Actually the sensor parameters of the patient is uploaded by the PatientApp and is shown in that website. The doctor can monitor the website to see the previous health status of the patient. There is an option “show previous health status” in the DoctorApp. If the doctor click on the option, the DoctorApp send a GET request a to the server to get the previous health status. The DoctorApp request is received by the *gettingnews.php*. This file retrieves the data from the mytab table and sends to the DoctorApp in JSON format. The DoctorApp processes the JSON format data and shows in an activity.

**Messaging from Patient to Doctor:** There is an option “send message to the doctor” in the home activity of the PatientApp. If the patient wants to send a message to the doctor, then he/she need to click on that option. It leads to a new activity in which we can write the message for the doctor. After writing the message when the he/she click on the button “*send the message to doctor*”, PatientApp sends a POST request to the server with the message. The PatientMessage.php receives the request and stores the message in the Messages table of the database. It also sends a request to the FCM server to send message notification to the DoctorApp. The FCM server receives the request and forward the message the DoctorApp.

There is an another php file called register.php to handle the devices registration token generated by the FCM server. We have an another table in the database to store the FCM token key which is unique for every single device. This is done by the register.php file.