

CHAPTER 1

CHAPTER 1

INTRODUCTION

Notifce helps user access online notices on their phone. The notice board has always been the place where staff/students gather to get their latest release of corporate news. Notifce brings the notice board to a virtual location where staff/students can not only read notices, but immediately react and respond to them - from their own desks! With this electronic notice and announcement system, notification alerts may be sent out notifying staff and students that a new notice has been posted, where staff may know if it concerns him directly.

The features that Notifce app should have:

- An electronic dashboard board for disseminating information out to staff and students.
- Notices can be posted, with response obtained instantly.
- Staff or student can be notified of new postings via notification alert.

The interface of this application is straightforward and takes us roughly a minute to get started. Adding notes to board is easy, just open the app, register and enter the text. Users can view the post on the spot by having a notification alert in android phone. Here registration is must for all the users having this application in order they want to have notification and staying stunned. This application can also act as a market place and let us advertise, for example, a room to rent, a car for sale, a job opportunity, a house for sale, an upcoming event, an announcement, a service, and so on.

1.1 Literature Survey

1.1.1 The Role of android communication application

”Notifce” is a mobile application an Internet based Mobile application. A mobile application is a computer program designed to run on smart phones, tablet computers and other mobile devices[1]. Apps are usually available through application distribution platforms, which began appearing in 2008 and are typically operated by the owner of the mobile operating system, such as the Apple App Store, Google Play, Windows Phone Store, and BlackBerry App World[2,3].

Mobile apps were originally offered for general productivity and information retrieval, including email, calendar, contacts, and stock market and weather information. However, public demand and the availability of developer tools drove rapid expansion into other categories, such as mobile games, factory automation, GPS and location-based services, banking, order-tracking, ticket purchases and recently mobile medical apps[4]. The explosion in number and variety of apps made discovery a challenge, which in turn led to the creation of a wide range of review, recommendation, and other sources including blogs, magazines, and dedicated online app-discovery services.

Mobile application development is the process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones[5]. These applications can be preinstalled on phones during manufacturing, downloaded by customers from various mobile software distribution platforms, or delivered as web applications using server-side or client-side processing (e.g. JavaScript) to provide an "application-like" experience within a Web browser[6].

Application software developers also have to consider a lengthy array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms[7,8]. Mobile app development has been steadily growing, both in terms of revenues and jobs created. The popularity of mobile apps has continued to rise, as their usage has become increasingly prevalent across mobile phone users.

1.2 Limitation of the current work

Currently our college has manual system of putting notices on notice board, which is out dated now. As expected, the current generation has no time to stand in rush in order to read the notices on notice board.

- Notice can get out of order in traditional notice board system. If someone accidentally puts some data in the wrong place, it can lead to lost data.
- Automated notice management systems allow users to quickly check whether information exists somewhere in the system, which helps avoid problems like redundant data.

- Automated system is less complex than manual system of handling notices, which can make it easier for untrained people to access and manipulate data. Anyone having the basic knowledge of mobiles can work on the automated system.
- There will be an unavailability for future use, since notice might get misplaced during manual notices management. So notice won't be preserved properly for future use.
- Manual notices stack are vulnerable to damage, destruction and theft in ways that digital databases are not. A company may back up its digital data both on site and at offsite locations, ensuring its security if the office building suffered a fire or similar disaster.
- Manual notices do not allow users to easily edit data or information. Manual notices often cannot be edited directly, forcing users to make new copies.

1.3 Problem definitions

To develop a mobile application that will help us receiving the notices from the college, anywhere, anytime. Earlier there was problem that notices were pasted on notice board regarding some placement or examinations. Also, if there is holiday on the next day, nobody will be able to know the updates. The more easy way this is, just type in message sitting wherever and post by pressing a button. It will notify all the staff and students.

1.4 Objectives

The main objective here is to develop and design an automated notice application to overcome the manual system of giving notice to students and staff members. of The online Notifice application is user-friendly android application. The objective of the application is its simplicity of design and ease of implementation that shows and helps to collect most of the information about events going on in college premises. The interface will be very user-friendly.

As discussed earlier that manual maintenance of a notices is a tedious job. So to enhance the ease of working, we go for this package.

- Giving the facility to convey messages to all students anytime and anywhere.

- Making students updated about all the events and activities going on in the college.
- The student will not require standing in the crowd to see the notice. There will be no issue of fighting in order to see the notice first. Everyone is first to see that notice inside their own mobile phone anywhere and anytime.
- The least but most important it saves time.
- Utilizing less man power. As there are many persons involved in circulating the message. With this application, only one person is required to post the notice. Rest of the man power is saved in the entire process.

The main objective is to propose this system in our college to overcome listed limitations of manual notice work. The following are some of the advantages that can be expected through this application.

- **To eliminate wastage of time and energy:** Notifce will be able to save lot of paper and time. It directs both teacher and pupils' energy and attention to one thing at a time by placing proper persons at their proper places at the proper time. Everything will be instantaneous.
- **To ensure due attention of student to each and every notice:** *Notifce* ensures that everyone has kind attention to every notice and updates going on in college. There will be a buzz at each and every notice to drive the attention of student to check it once. In this way, students will be well informed about their college activities.
- **To bring system into college life:** It would be dire need of all colleges as it's easy and shortcut method to inform all the students. In the absence of proper notification system will make it very difficult to inform students at right time.
- **Free Service:** It gives free service to notify all the students. There will be no cost of sending notification to all. Just have the good system implemented in college and that too free of cost.
- **Prevent Crowd in College:** As we are aware, there is always a crowd at notice board. As notice board is one, and people to see notice is more. With this application there will be no more crowds. Everyone will be well informed even at their homes. So they are free to do there other work.

- **Anytime Anywhere Service:** With this application, notices will be delivered anytime and at any place. There is no restriction of time to send a notice.

1.5 Methodology

The development cycle is divided into following two sections

- The look and feel of the application (ANDRIOD STUDIO)
- The back end (USING XAMPP SERVER AND Firebase Cloud Messaging)

1.6 Hardware and software tools used

1.6.1 Software

- Operating System : Windows 10/8/7
- IDE : Android Studio

1.6.2 Hardware

- Processor : Intel i3 Core
- Hard Disk : 4GB
- RAM : 8GB

CHAPTER 2

CHAPTER 2

BASIC THEORY

2.1 Feasibility Study

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. The objective for this phase is not to solve the problem but to acquire a sense of scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined.

Mobile Application Development Systems are capital investments because resources are being spent currently in order to achieve benefits to be received over a period of time following completion. There should be a careful assessment of each project before it is begun in terms of economic justification, technical feasibility, operational impact and adherence to the master development plan. We started the project by listing the possible queries that the user might want to be satisfied. And on these lines we took the project further.

The three main points, kept in mind at the time of project, are:

- Possible (To build it with the given technology and resources)
- Affordable (given the time and cost constraints of the organization)
- Acceptable (for use by the eventual users of the system)

The three major areas to be considered while determining the feasibility of a project are:

- I. **Technical Feasibility:** The technical issue usually raised during the feasibility stage of the investigation includes the following:
 - Does the necessary technology exist to do what is suggested?
 - Does the proposed equipment have the technical capacity to hold the data required to use the new system?

- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of Secure Infrastructure Implementation System. The current system developed is technically feasible. It is a web based user interface. Thus it provides an easy access to the users. The databases purpose is to create, establish and maintain a work- flow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

II. **Operational Feasibility:** Under this category of service we conduct a study to analysis and deter-mine whether user needs can be fulfilled by using a proposed solution. The result of our operational feasibility Study will clearly outline that the solution proposed for user business is operationally workable and conveniently solves user problems under consideration after the proposal is implemented. We would precisely describe how the system will interact with the systems and persons around. Our feasibility report would provide results of interest to all stakeholders. It will do as per the needs of the business requirements.

III. **Timeline Feasibility:** It is important to understand that a need must be fulfilled when it has to be. Some otherwise feasible and highly desirable projects can become non-feasible due to very restrictive timeline constraints. This fact makes it imperative that milestones are clearly linked to the timeline and projects are well conceived with safe unforeseen margins. We make sure that we strictly follow what has been stated above.

2.2 Software Requirement Specification Document

2.2.1 Data Requirements

Data requirement is meant to be the data that will be used in our application. Data required in this project is all notices that need to be conveyed to the user. This application also requires the username and passwords of persons in order to register them and sending notification about updates. So the two main requirements are: Notice Details and User Details.

2.2.2 Functional Requirements

In order to make this application functional, we require the following:

- ***Download mobile application:***

A user should be able to download the mobile an application through either an application store or similar service on the mobile phone. The application should be free to download.

- ***User registration:***

Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number.

- ***User Login:***

Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

- ***Posting Notices:***

The admin of this application should be able to post the notices. He should be able to add a picture within notices. That picture can be taken either from gallery or by using the camera of the mobile phone.

- ***Notification Alert:***

All the registered users should be able to have a ping or notification on their mobile phone whenever a new notice is posted.

2.2.3 Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

- **Prominent search feature:**

The search feature should be prominent and easy to find for the user.

- **Usage of the Notice Information:**

The notice link should be prominent and it should be evident that it is a usable link. Selecting the notice link should only take one click.

- **Response Time:**

The response time should not be more than 5 seconds if user has a proper internet connection.

2.2.4 System Dependability

Following are the requirements that an application require from the device/mobile on which it is installed.

- **Internet Permission:**

Application developed, require full internet permissions of mobile so that it can fetch notices from the server. At the same time, it should be able to receive buzz or notification tone whenever new notice is posted by admin.

- **External SD Card Writable Permissions:**

This application would be requiring read write access to SD card. It is required in order to download the notices attachment and save in SD card of mobile phone.

- **System Tools:**

This application require various system tools to be used. For example, it requires Camera of mobile in order to click the image and post in into notice. It also require system tool that prevents it from sleeping.

- **Hardware Control:**

It uses vibrator of mobile phone whenever any notification arrives.

- **Account Info:**

It also fetches user Google account information in order to get the user registered with Firebase Cloud Messaging.

2.2.5 Maintainability Requirements

Following are the maintainability requirement of e-Notice mobile application:

- ***Application extendibility:*** The application should be easy to extend. The code should be written in a way that it favours implementation of new functions. It is requires in order for future functions to be implemented easily to the application.
- ***Application testability:*** Test environments should be built for the application to allow testing of the applications different functions.

2.2.6 Look and Feel Requirements

Regarding look and feel, our client is straight forward. They believe in simplicity. So these are their requirements:

- ***Simple and Light:***
The user interface should be simple and lightly coloured. It should give relaxing effect on looking at its GUI. No bright colors should be used while designing the UI of this application.
- ***Easy to Use***
The application should be easy to use. If any user is doing something wrong, he/she should be informed correctly about what is going wrong behind the scenes. There should be proper instructions for the user to use this application.
- ***Soft Sound Notification:***
The sound for notification should be very soft. It should not disturb the person with a loud note. Everything should be sober in this application.

2.3 Validations

Any application is useless without validation. There should be a way to validate the user input first before sending the user request to the server. Following are the validations implemented in proposed system:

- ***User Password Validation:***
The application should check the user and password fields before sending any request to the server. It should check whether the fields are filled or not. if fields

are not filled up, user should be instructed to fill up the fields before moving further. in this way, there will be less traffic on the server.

- ***Validations During Registration:***

There are a lot of validations that needs to be implemented in the application.

They are as follow:

1. ***First and Last Name of User:***

The first and last name of user should be not null. Also first letter of first and last name should be in uppercase.

2. ***Username:***

The username can contain only alphabets, digits, underscore and hyphen. It should be at least characters long and maximum of 15 characters.

3. ***Password:***

The password must contains one digit from 0-9 and length of password must be at least 6 characters and maximum of 20.

4. ***Email:***

The application must validate and email address entered by the user before sending request to the server.

5. ***Mobile Number:***

The mobile number should be of only ten digits. No more, no less than that.

CHAPTER 3

CHAPTER 3

TOOL DESCRIPTION

3.1 Introduction to Languages, IDE's, Tools and Technologies

3.1.1 Java

Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications.

Some of the Java's important core features are:

- It's easy to learn and understand.
- It's designed to be platform-independent and secure, using virtual machines.
- It's object-oriented.

Android relies heavily on these Java fundamentals. The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else that we may wish) as well as special Android libraries that will help us develop awesome Android applications. With many programming languages, user need to use a compiler to reduce their code down into machine language that the device can understand. While this is well and good, different devices use different machine languages. This means that user might need to compile their applications for each different device or machine language in other words; our code isn't very portable. This is not the case with Java.

The Java compilers convert user code from human readable Java source files to something called Byte code in the Java world. These are interpreted by a Java Virtual Machine, which operates much like a physical CPU might operate on machine code, to actually execute the compiled code.

Although it might seem like this is inefficient, much effort has been put into making this process very fast and efficient. These efforts have paid off in that Java performance is generally second only to C/C++ in common language performance comparisons.

Android applications run in a special virtual machine called the Dalvik VM. While the details of this VM are unimportant to the average developer, it can be helpful to think of the Dalvik VM as a bubble in which user Android application runs, allowing user to not have to worry about whether the device is a Motorola Droid, an HTC Evo, or the latest toaster running.

3.1.2 Why is Java Secure?

Let's take this bubble idea a bit further. Because Java applications run within the bubble that is a virtual machine, they are isolated from the underlying device hardware. Therefore, a virtual machine can encapsulate, contain, and manage code execution in a safe manner compared to languages that operate in machine code directly. The Android platform takes things a step further. Each Android application runs on the (Linux- based) operating system using a different user account and in its own instance of the Dalvik VM. Android applications are closely monitored by the operating system and shut down if they don't play nice (e.g. use too much processing power, become unresponsive, waste resources, etc.).

Therefore, it's important to develop applications that are stable and responsive. Applications can communicate with one another using well- defined protocol.

3.2 Android Development Tools

3.2.1 Android SDK

The Android Software Development Kit (Android SDK) contains the necessary tools to create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language.

3.2.2 Android Debug Bridge

The Android SDK contains the Android debug bridge, which is a tool that allows user to connect to a virtual or real android device, for the purpose of managing the device or debugging user application.

3.2.3 Android Developer Tools and Android Studio

Google provides two integrated development environments (IDEs) to develop new applications. The Android Developer Tools are based on the Eclipse IDE. ADT is a set of components (plugins), which extend the Eclipse IDE with Android development capabilities. Google also supports an IDE called Android Studio for creating Android applications. This IDE is based on the IntelliJ IDE.

Both IDEs contain all required functionality to create, compile, debug and deploy Android applications. They also allow the developer to create and start virtual Android devices for testing. Both tools provide specialized editors for Android specific files. Most of Android's configuration files are based on XML. In this case these editors allow us to switch between the XML representation of the file and a structured user interface for entering the data.

3.2.4 Dalvik Virtual Machine

The Android system uses a special virtual machine, i.e., the Dalvik Virtual Machine (Dalvik) to run Java based applications. Dalvik uses a custom Bytecode format which is different from Java Bytecode. Therefore user cannot run Java class files on Android directly; they need to be converted into the Dalvik Bytecode format.

3.2.5 Android RunTime

With Android 4.4, Google introduced the Android RunTime as optional runtime for Android 4.4. It is expected that versions after 4.4 will use ART as default runtime. ART uses Ahead of Time compilation. During the deployment process of an application on an Android device, the application code is translated into machine code. This results in approx. 30% larger compile code, but allows faster execution from the beginning of the application.

3.3 Security and Permission Concept in Android

3.3.1 Security

The Android system installs every Android application with a unique user and group ID. Each application file is private to this generated user, e.g., other applications cannot access these files. In addition each Android application is started in its own process.

Therefore, by means of the underlying Linux kernel, every Android application is isolated from other running applications. If data should be shared, the application must do this explicitly via an Android component which handles the sharing of the data, e.g., via a service or a content provider.

3.3.2 Permission

Android contains a permission system and predefines permissions for certain tasks. Every application can request required permissions and also define new permissions. For example, an application may declare that it requires access to the Internet. Permissions have different levels. Some permissions are automatically granted by the Android system, some are automatically rejected. In most cases the requested permissions are presented to the user before installing the application. The user needs to decide if these permissions shall be given to the application.

If the user denies a required permission, the related application cannot be installed. The check of the permission is only performed during installation, permissions cannot be denied or granted after the installation. An Android application declares the required permissions in its `AndroidManifest.xml` configuration file. It can also define additional permissions which it can use to restrict access to certain components.

3.4 Coding Standards of Language Used

Coding Standards of Java are used in the whole project. This standard includes the following:

- No wildcard imports.
- Overloads appear sequentially.

- Braces are used even when the body is empty or contains a single statement.
- Two space indentation.
- Column limit can be 80 or 100 characters.
- No C-style array declarations.
- Default statements required in switch statements.
- Modifiers appear in the order recommended by the Java Language Specification.
- Constants use CONSTANT CASE. Note that every constant is a static final field, but not all static final fields are constants.
- Class name should start with uppercase letter.
- Function name should start with lowercase letter.

3.5 Test Plan and Test Activities

3.5.1 Test Plan

A test plan can be defined as a document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. In software testing, a test plan gives detailed testing information regarding an upcoming testing effort, including

- Scope of testing
- Schedule
- Test Deliverables
- Release Criteria
- Risks and Contingencies

It is also be described as a detail of how the testing will proceed, who will do the testing, what will be tested, in how much time the test will take place, and to what quality level the test will be performed. The process of defining a test project ensures that it can be properly measured and controlled. The test planning process generates a high level test plan document that identifies the software items to be tested, the degree of tester independence, the test environment, the test case design and test measurement techniques to be used, and the rationale for their choice.

A testing plan is a methodological and systematic approach to testing a system such as a machine or software. It can be effective in finding errors and flaws in a system. In order to find relevant results, the plan typically contains experiments with a range of operations and values, including an understanding of what the eventual workflow will be.

Test plan is a document which includes, introduction, assumptions, list of test cases, list of features to be tested, approach, deliverables, resources, risks and scheduling. A test plan is a systematic approach to testing a system such as a machine or software. The plan typically contains a detailed understanding of what the eventual workflow will be. A record of the test planning process detailing the degree of tester independence, the test environment, the test case design techniques and test measurement techniques to be used, and the rationale for their choice.

3.5.2 Test Activities

Various Testing Activities are as follow:

1. **Black box testing** Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.
2. **White box testing** This testing is based on knowledge of the internal logic of an applications code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.
3. **Unit testing** Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.
4. **Incremental integration testing** Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. It's done by programmers or by testers.
5. **Integration testing** Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

6. **Functional testing** This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.
7. **System testing** Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.
8. **End-to-end testing** Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.
9. **Acceptance testing** Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer does this testing to determine whether to accept application.
10. **Usability testing** User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

CHAPTER 4

CHAPTER 4

IMPLEMENTATION

4.1 Hardware Design and Implementation

4.1.1 GCM Messaging

Google Cloud Messaging for Android is a service that allows user to send data from server to users' Android-powered device, and also to receive messages from devices on the same connection.

The GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target device. GCM is completely free no matter how big user messaging needs are, and there are no quotas.

4.1.2 GCM Working

1. Android device sends SENDER ID to GCM Server for registration.
2. After successful registration, GCM sends Registration ID to Android device.
3. After getting registration id, Android device sends registration id to Web Server.
4. Store registration id in our database at the server.
5. Whenever Push Notification needed, get registration ids from server, and send the request to GCM with registration id and message.
6. After push notification request, GCM sends Push Notifications to Android device.

Fig 4.1 shows working of Google Cloud Messaging.

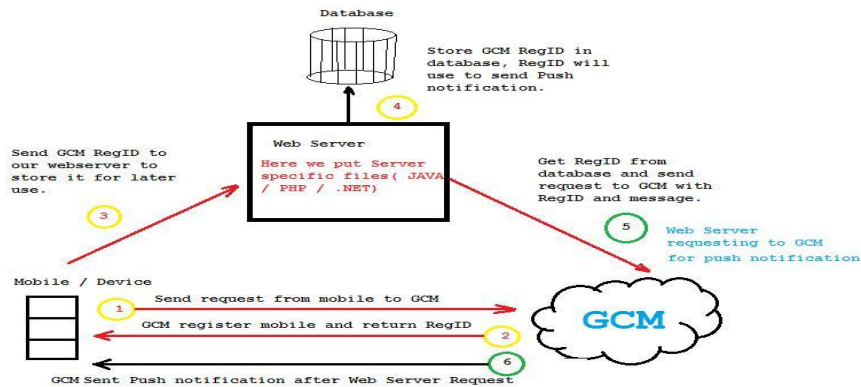


Figure 4.1: Google Cloud Messaging Working

Following are the steps to use GCM in android project:

1. Register with Google Cloud Messaging from GOOGLE API Console and get Sender ID and API key for Google Cloud Messaging.
2. Set Emulator for Google Cloud Messaging helper library.
3. Create Android project to register with Google Cloud Messaging (GCM).
4. Create server side code to save Google Cloud Messaging registration id in our database and send push notification to device.

4.1.3 HTTP Connections to Web Server

Steps to have HTTP connection to Web Server are as follow:

1. Declare Internet permissions in the manifest by adding the following line to AndroidManifest.xml.
2. Create HttpClient and HttpPost objects to execute the POST request.
3. Set POST data. This is done by creating and setting a list of NameValuePairs as user Http-Post's entity. Be sure to catch the UnsupportedOperationException thrown by HttpPost.setEntity().
4. Execute the POST request. This returns an HttpResponse object, whose data can be extracted and parsed. Be sure to catch the ClientProtocolException and IOException thrown.

4.1.4 JSON Parsing

JSON is an independent data exchange format and is the best alternative for XML. Steps of JSON Parsing are:

1. For parsing a JSON object, we will create an object of class JSONObject and specify a string containing JSON data to it.
2. A JSON file consist of different object with different key/value pair e.t.c. So JSONObject has a separate function for parsing each of the component of JSON file.

4.1.5 Broadcast Receiver

Broadcast receiver is a component that responds to system conditions such as low battery or the screen being turned off. User can use broadcast receivers to initiate a response from a running application, such as if a picture has been taken.

4.1.6 Services

Service is a process that runs in background to perform long term operations or work for remote processes. Services don't provide a User Interface.

4.1.7 Content Provider

It manages persistent data on the device or external sources such as web or cloud or any other system application has access to. Android devices has on-board SQLite database management system to provide organized persistent data storage.

4.2 Software Algorithm

This project requires using the services of Firebase Cloud Messaging (FCM), which is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which currently can be used at no cost.

4.2.1 Integrating Firebase Cloud Messaging in Android Project

4.2.1.2 Creating a new Android Project

- With Android Studio 2.2 it is really easy to integrate firebase in user project.
- First create a new Android Studio Project with an Empty Activity.
- Once user project is loaded click on firebase from the tools menu.

Fig 4.2 shows the snapshot to select firebase

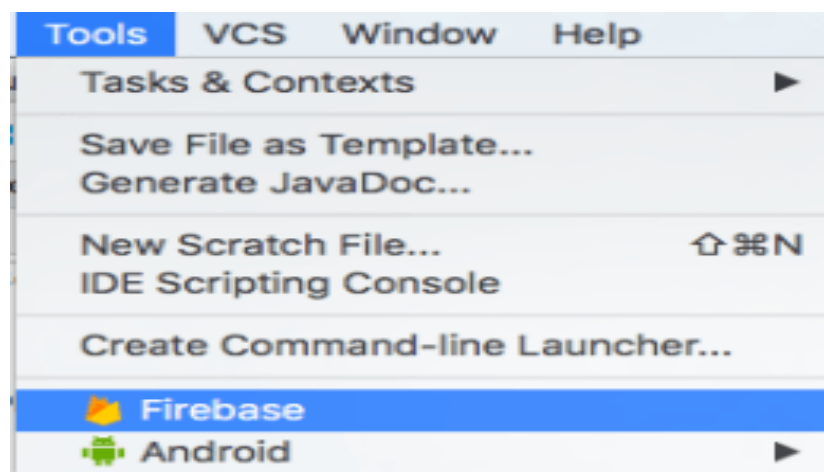


Fig 4.2 Selecting firebase

- After clicking on Firebase an assistant will open showing all the available firebase features Fig 4.3.

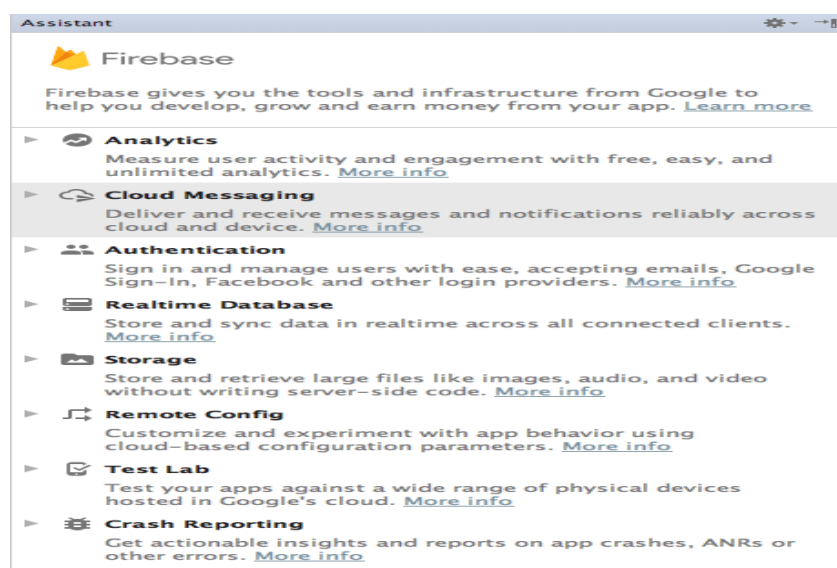


Fig 4.3 Firebase features

- As in Fig 4.4 we will see a link saying **Set Up Firebase Cloud Messaging**. Click on it.

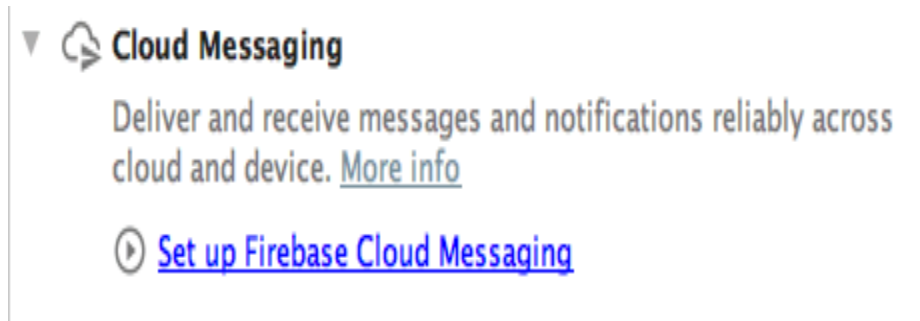


Fig 4.4 Setting up Firebase Cloud Messaging

- Now we will see all the steps required to **Set up Firebase Cloud Messaging** in the project Fig 4.5.

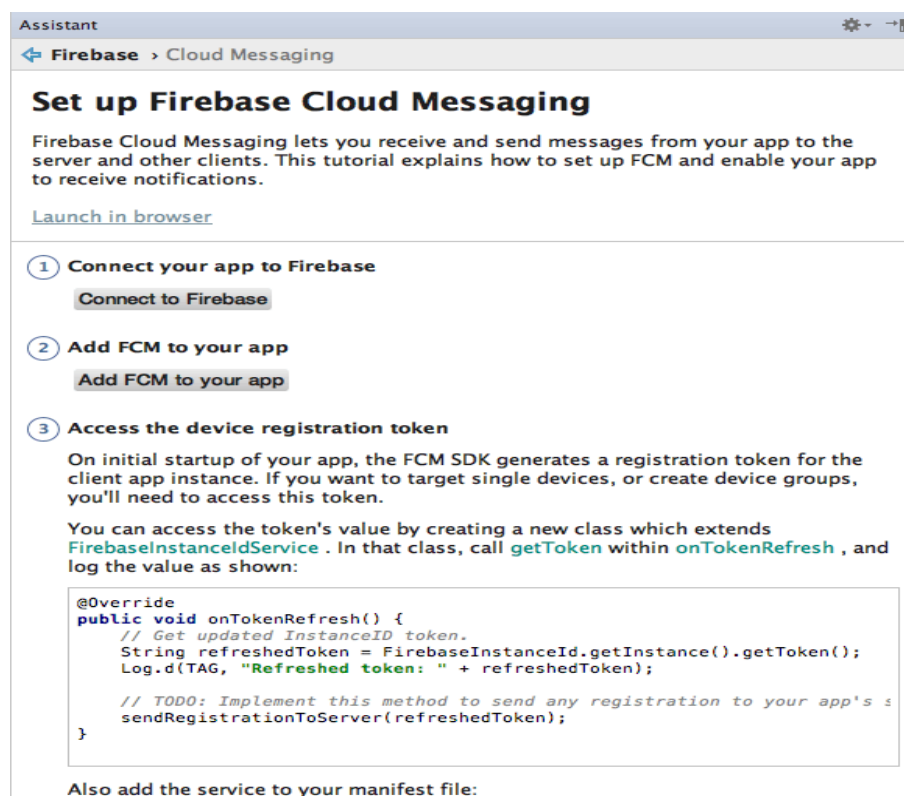


Fig 4.5 Cloud Messaging dialog box

- Simply we have to click on the **Connect to Firebase** button. It will start a connect dialog.
- Here we can create a new Project on Firebase, as in Fig 4.6 we are creating a project named **FcmSimplifiedCoding** (See the below screenshot). Or you can also select an existing firebase project.

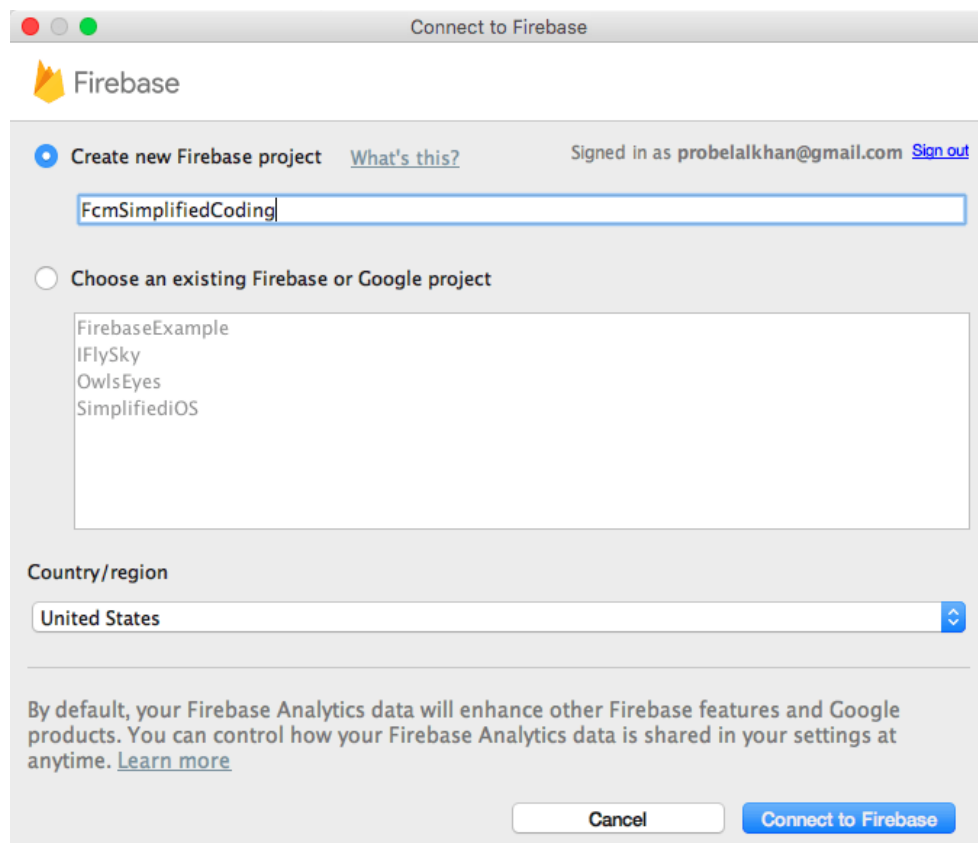


Fig 4.6 Connect Your App to Firebase

- Now we have to simply click on **Connect to Firebase**. And wait for a few seconds. And we will see the following message in the assistant Fig 4.7.

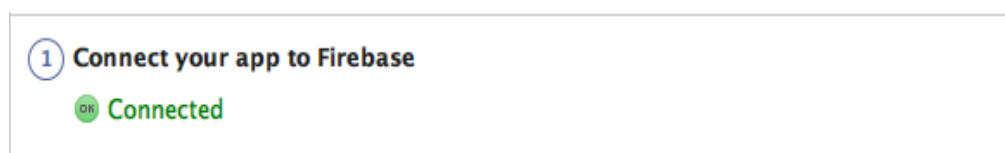


Fig 4.7 Connect status dialog

- Again we click on the button **Add FCM to your app** and we will see a dialog box Fig 4.8.

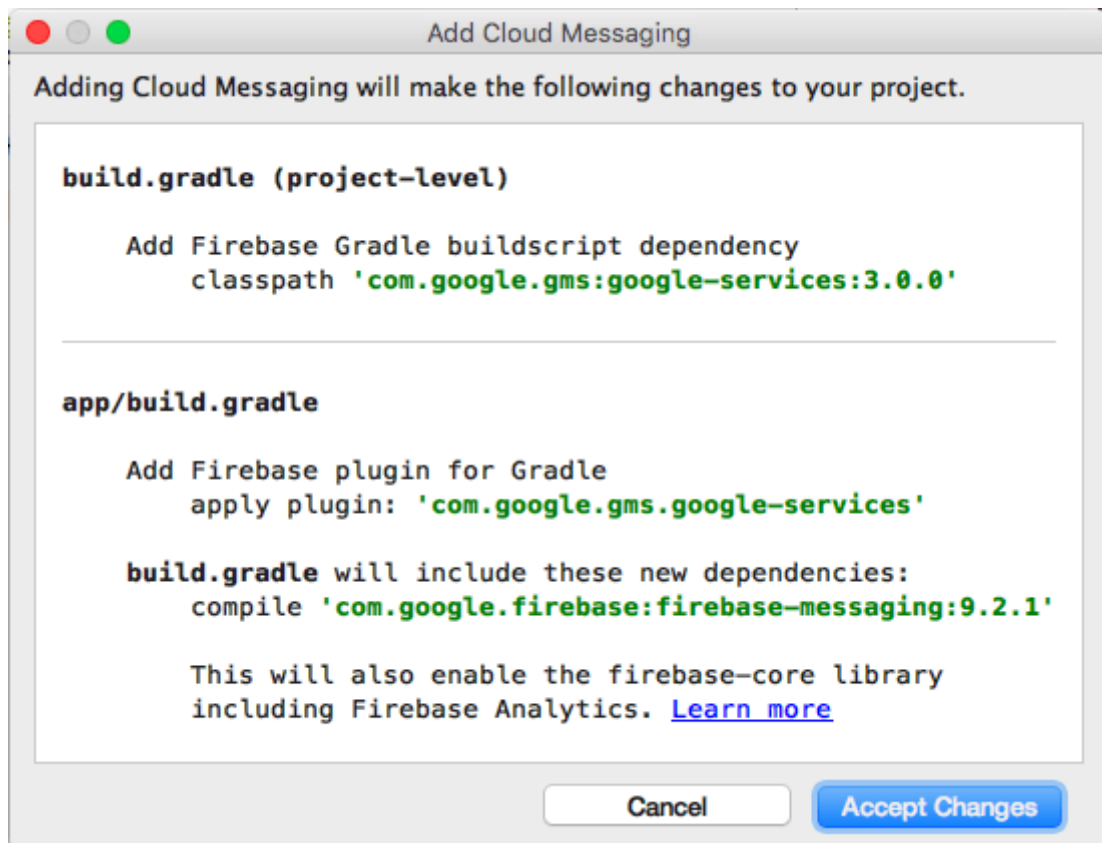


Fig 4.8 Adding FCM to user App

- We have click on **Accept Changes**. And firebase is setup in our project now.

CHAPTER 5

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Snapshot of the System

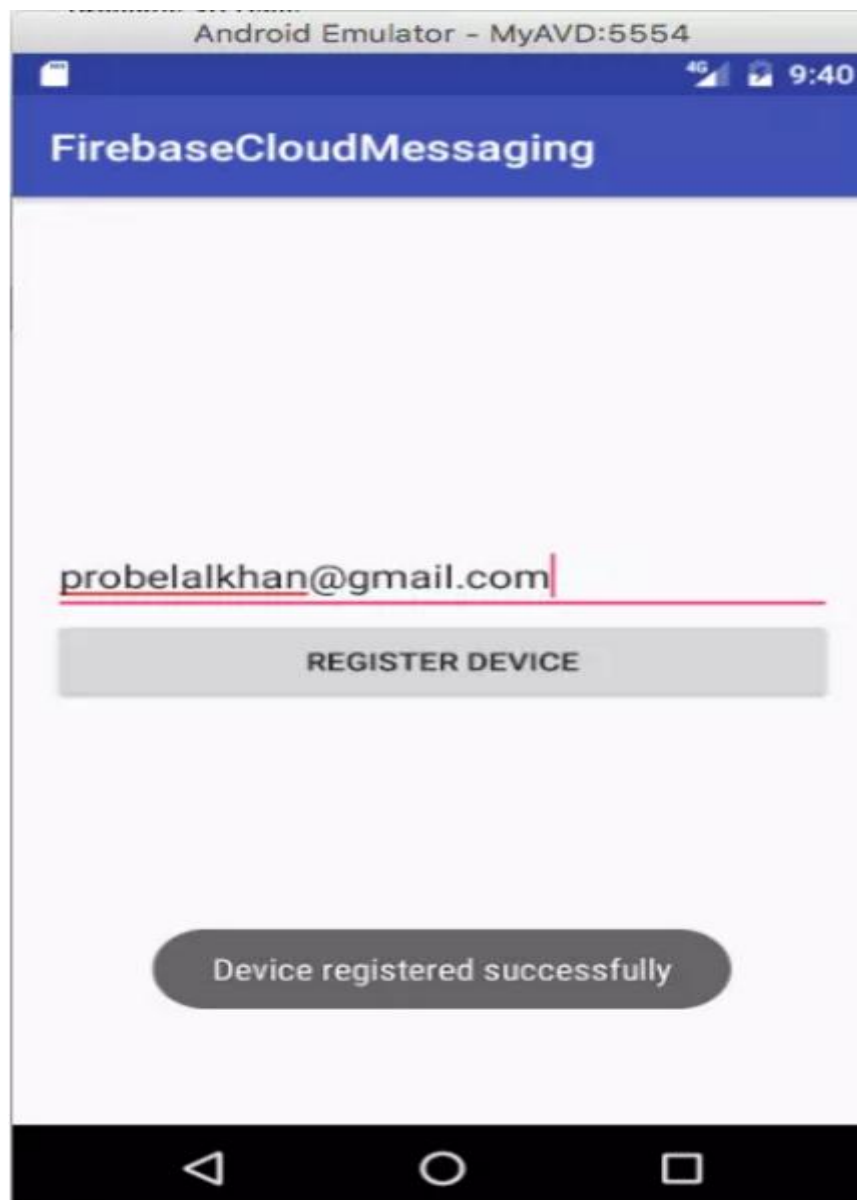


Fig 5.1 Adding FCM to user App

- As we can see in fig 5.1, we are getting the success message for device registration. Now to check the MySQL database to ensure that token is stored, the snapshot is in Fig 5.2.

+ Options

id

email

token

1

probelalkhan@gmail.com

fsySFlyklrA:APA91bHefvUgPC2_CN7qvDNboVvBZMxdJayLXt...

Check all

With selected:

Show all

Number of rows: 25

Filter rows: Search this table

Fig 5.2 MySQL database verification

- Now again we can test this script using POSTMAN.
- First we tried sending a notification without image. So we have to put only **title, message and email** in parameters Fig 5.3.

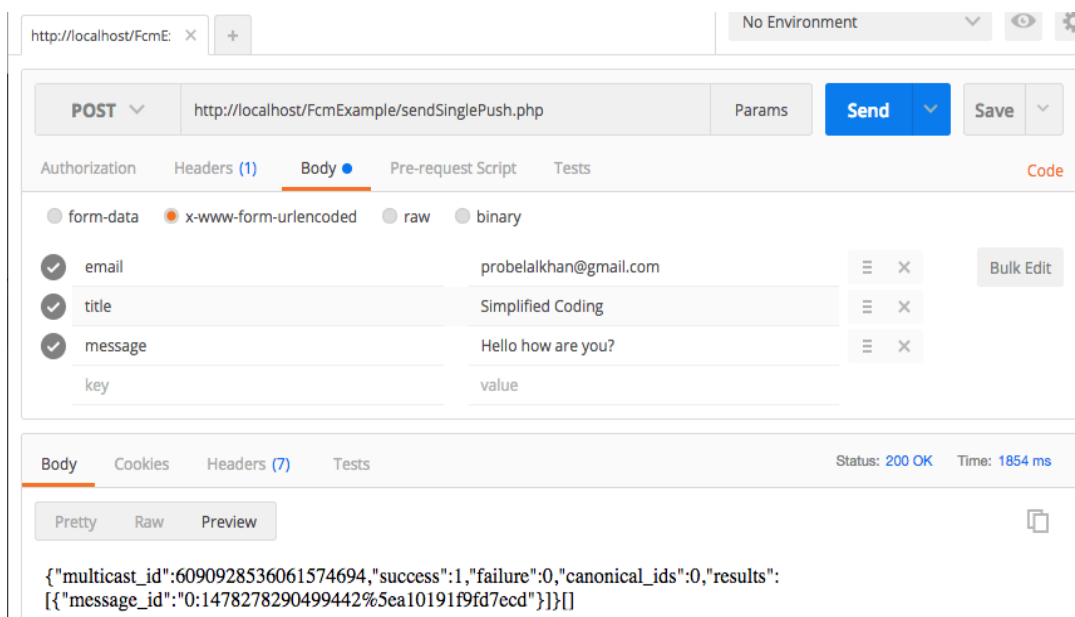


Fig 5.3 Text notification

- If we are getting a success then you should see the notification in our device as well Fig 5.4.

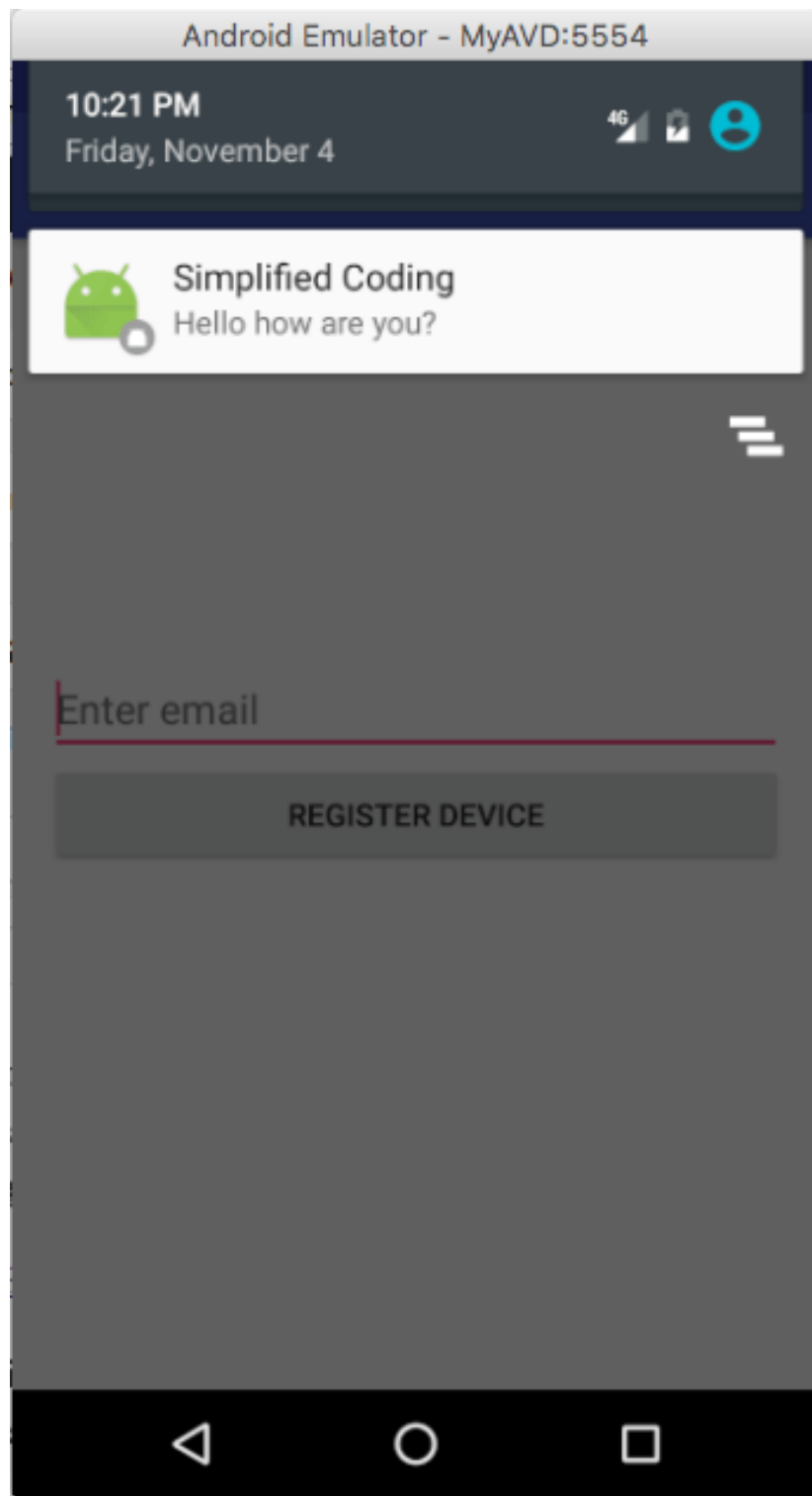


Fig 5.4 Text notification in app

- Now, we also tried notification with an image. To send an image along with the message we just need to put one more parameter in the request named image and it will contain the url of the image, Fig 5.5.

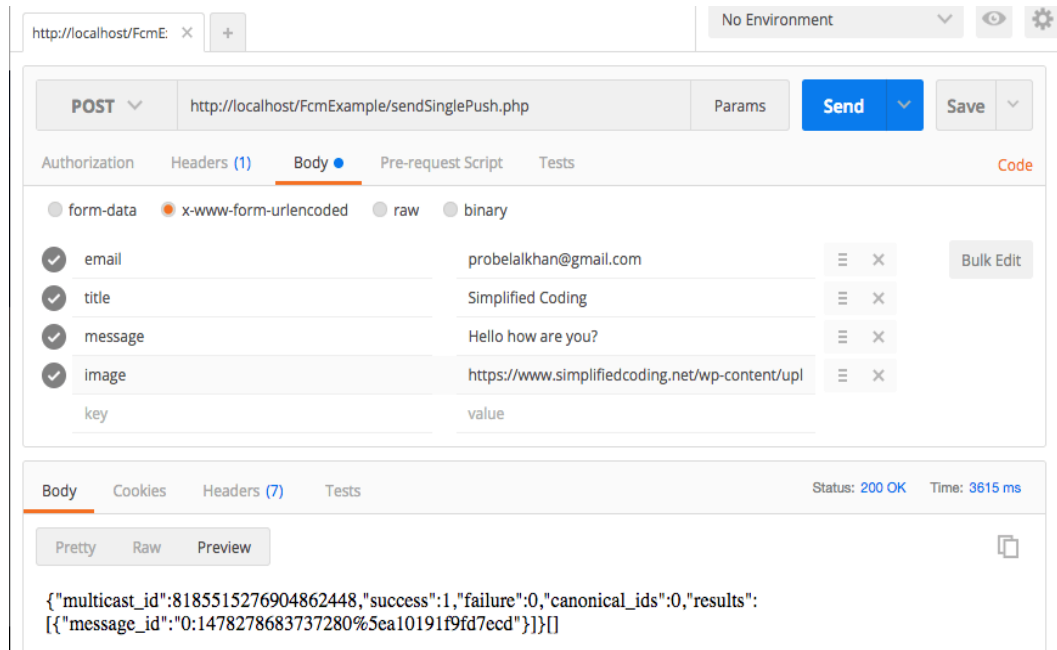


Fig 5.5 Notification with image

- At this time, we should see a notification as below in the device Fig 5.6.

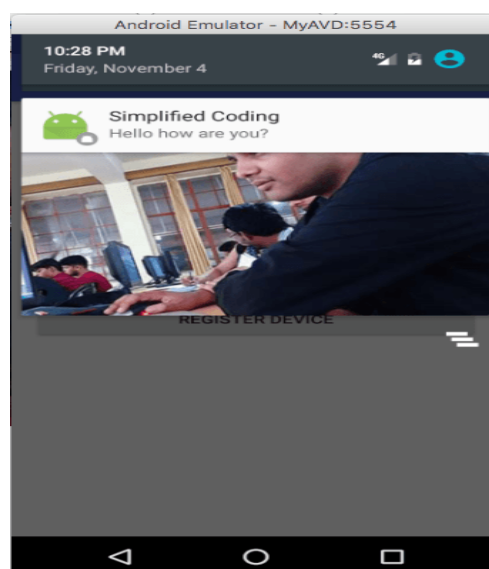


Fig 5.6 Notification with image in app

- **Sending to one**
 - If we run the application, now we will see the emails in the spinner Fig 5.7a and 5.7b.

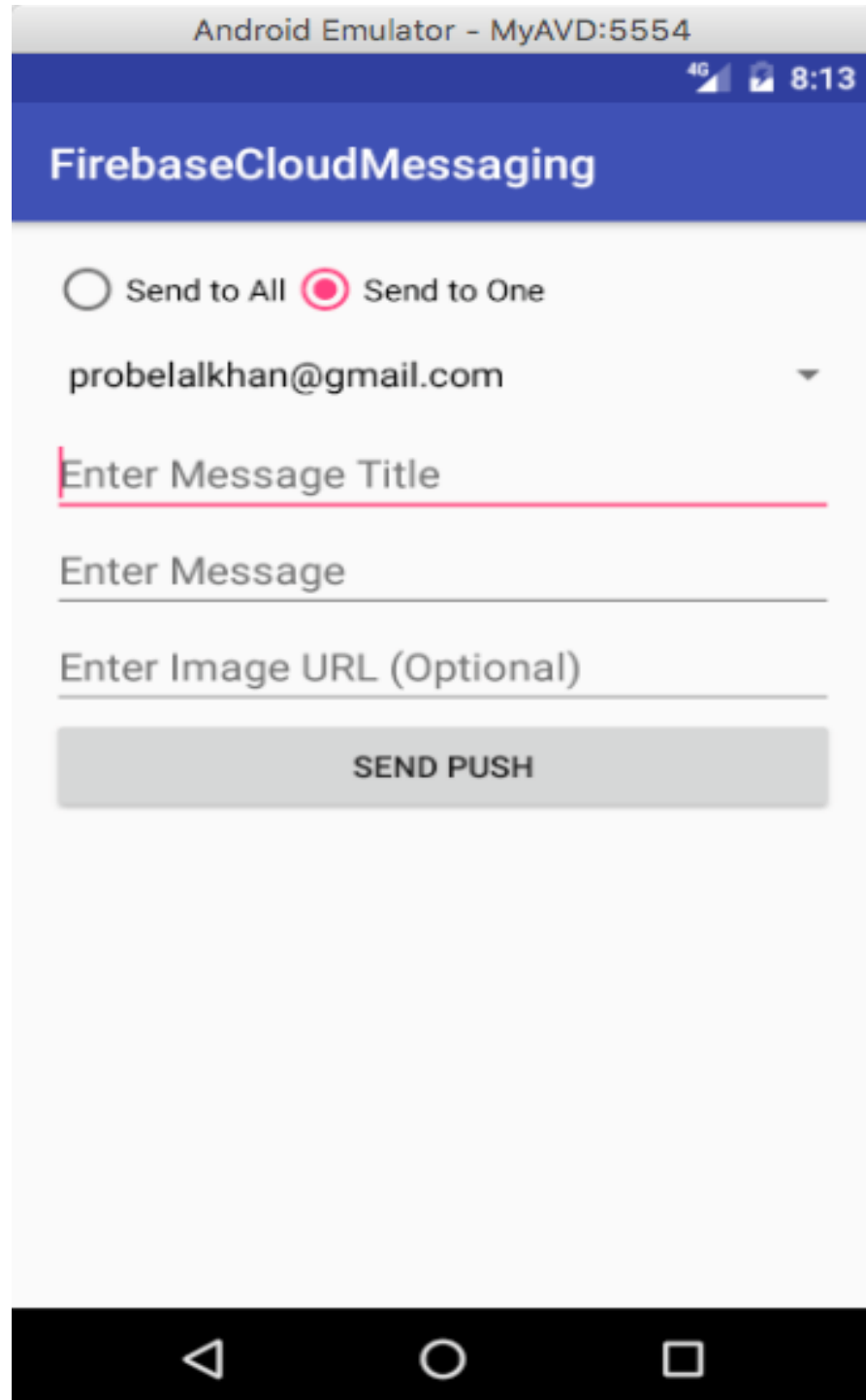


Fig 5.7a sending to one user

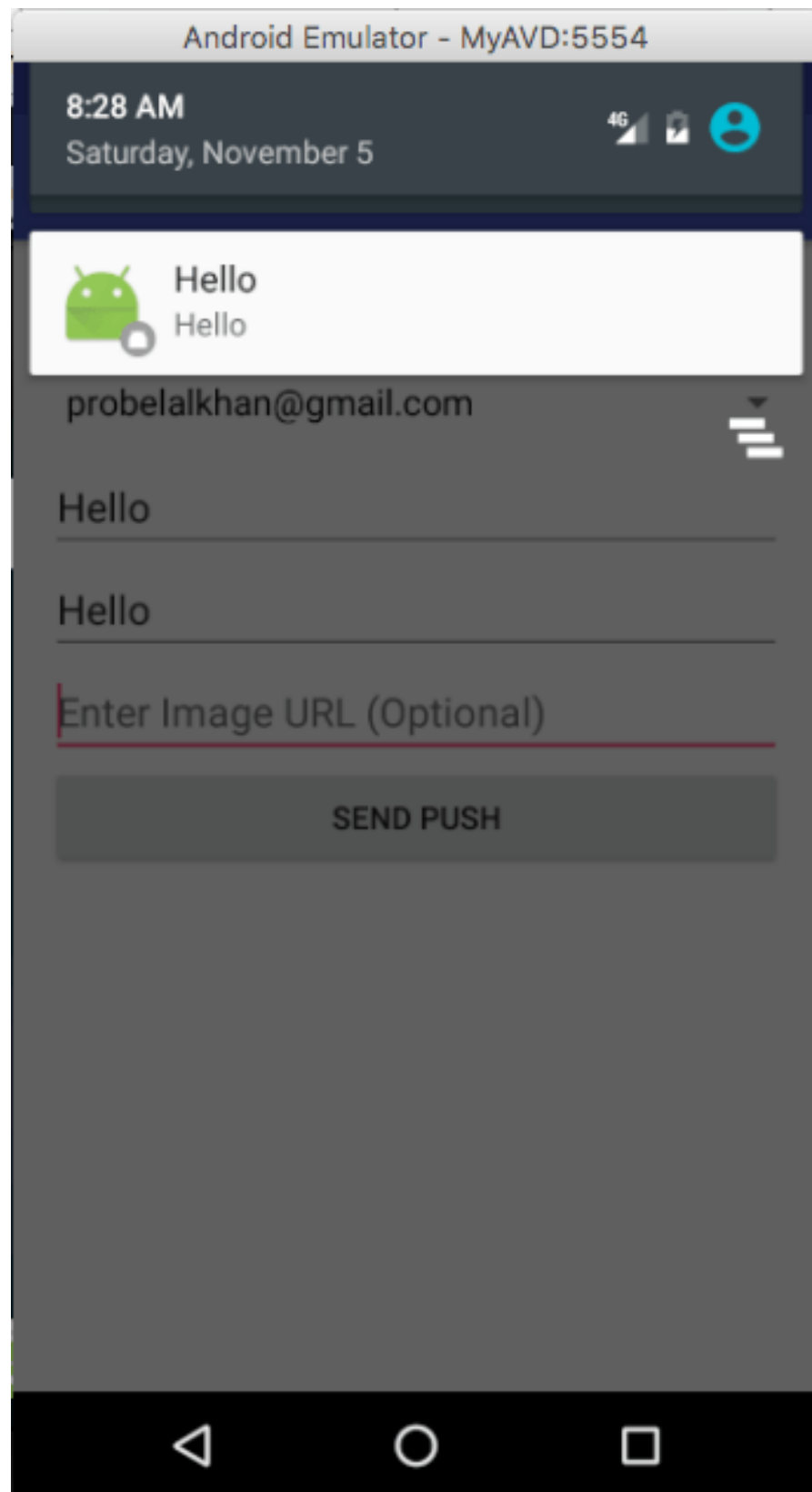


Fig 5.7b sending to one user

- **Sending to Multiple Device**
 - We just need multiple emulators now to test the application Fig 5.8.

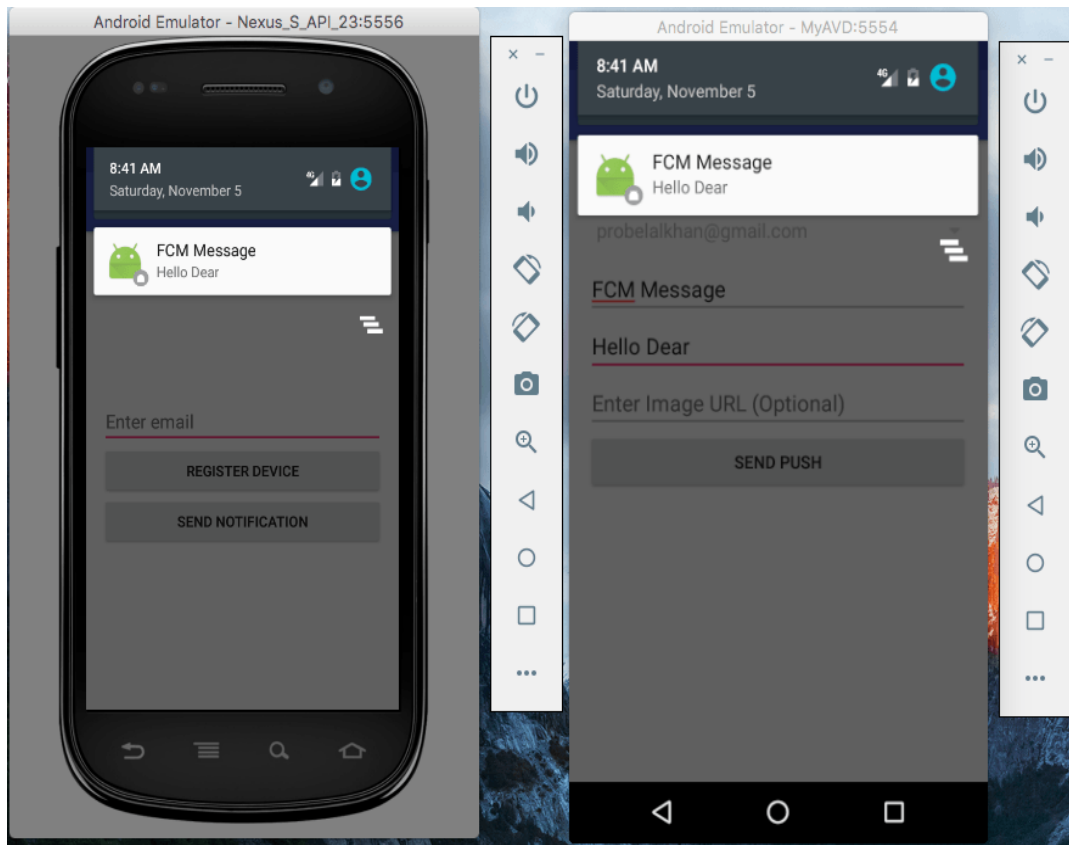


Fig 5.8 sending to multi user

CONCLUSIONS AND FUTURE SCOPE

Notifce is a app that helps user in real-time to access online the notices on their phone. Notifce brings the notice board to a virtual location where staff/students can read notices anywhere anytime in their smartphone with internet. They immediately react and respond - from their own desks! With this electronic notice and announcement system, notification alerts may be sent out notifying staff and students that a new notice has been posted, where staff may know if it concerns him directly.

The features that we would like to enhance in our Notifce app:

- We include the Industries tied up with colleges to share their requirements on projects, where the interested and eligible student can take part.
- We include the colleges associated with industries that are tied up with our college to share their upcoming knowledge based events, where in the student gets the information on the platform of their interest.