

# **TRIBHUVAN UNIVERSITY INSTITUTE OF ENGINEERING**

**Kathmandu Engineering College**  
Department of Computer Engineering



Major Project Report  
On

**NOTIFY-ME**

[Code No: CT755]

By:

Dishant Maharjan (70026)

Kushal Budhathoki (70037)

Nidesh Gautam (70041)

Bijen Maharjan (70044)

Kathmandu, Nepal

2074

**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING**

**Kathmandu Engineering College**  
Department of Computer Engineering

**NOTIFY-ME**  
[Code No: CT755]

PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF  
COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE BACHELOR OF ENGINEERING



By:

Dishant Maharjan (70026)

Kushal Budhathoki (70037)

Nidesh Gautam (70041)

Bijen Maharjan (70044)

Kathmandu, Nepal

2074

**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING**

**Kathmandu Engineering College**  
Department of Computer Engineering

**CERTIFICATE**

The undersigned certify that they have read and recommended to the Department of Computer Engineering, a major project work entitled "Notify-Me" submitted by Dishant Maharjan (70026), Kushal Budhathoki (70037), Nidesh Gautam (70041) and Bijen Maharjan (70044) in partial fulfillment of the requirements for the degree of Bachelor of Engineering.

---

Supervisor

Er. Nabin Neupane

Department of Computer Engineering  
Kathmandu Engineering College

---

External Examiner

Bibha Sthapit

Department of Electronics and Computer  
Institute of Engineering, Central Campus  
Pulchowk

---

Project Coordinator

Er. Reshma Maharjan

Department of Computer Engineering  
Kathmandu Engineering College

---

Head of Department

Er. Sudeep Shakya

Department of Computer Engineering  
Kathmandu Engineering College

## **COPYRIGHT**

The authors have agreed that the Library, Department of Computer Engineering, Kathmandu Engineering College may make this report freely available for inspection. Moreover, the authors have agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the authors of this project and to the Department of Computer Engineering, Kathmandu Engineering College in any use of the material of this report. Copying or publication or the other use of this report for financial gain without approval of the Department of computer Engineering, Kathmandu Engineering College and authors' written permission is strictly prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to

Head

Department of Compute Engineering

Kathmandu Engineering College

Kalimati, Kathmandu

## ACKNOWLEDGEMENT

We must express our deep gratitude for the opportunity given to us to work on this project. We are also deeply grateful to all for their support and encouragement in the report, without whom this proposal would not have been possible.

First of all, we have no words that can adequately carry our sincere gratitude to Department of Computer Engineering, Kathmandu Engineering College, Kalimati for providing us all the opportunity to conduct this project.

We are very grateful to our supervisor **Er. Nabin Neupane** for his guidance, we are also thankful to our Project coordinator **Er. Reshma Maharjan** and the Head of Department of Computer Engineering, **Er. Sudeep Shakya** for their valuable suggestions and encouragement for this project.

Lastly we would also like to acknowledge and thank all the faculty members for the kind support and guidance as well as their inspiration to move forward for the project. We can't leave to thank our dear friends and family as they have provided us with their valuable comments and suggestions for the completion.

Dishant Maharjan

Kushal Budhathoki

Nidesh Gautam

Bijen Maharjan

## **ABSTRACT**

Notify-Me, the notification system is a combination of software and hardware that provides a means of delivering a message to a set of recipients. The system is designed to take over the classical method of notice board system by providing in-hand notices along with push notification. Push notification is the delivery of information from a software application to a computing device without a specific request from the client. This information system comprises of webpage for admin and android application for recipients, both interacting with the server. The admin can send words and images with provided webpage. The server sends request to Firebase Cloud Messaging for push service while storing notices in database. The users with the application logged on receives the notifications through API tokens requested during the register of application device. The application verifies the login information of the user to send the valid notices of the affiliated organization.

## Table of Content

Copyright.....	i
Acknowledgement.....	ii
Abstract.....	iii
List of Figures.....	vi
List of Abbreviation.....	vii
 CHAPTER ONE: INTRODUCTION.....	 1
1.1 Introduction.....	2
1.2 Background Theory.....	2
1.3 Problem statement.....	3
1.4 Objective.....	4
1.5 Scope and its Application.....	4
 CHAPTER TWO: LITERATURE REVIEW.....	 5
2.1 Existing System and Applications.....	6
2.2 Android Concepts.....	7
2.2.1 Android Life Cycle.....	7
2.2.2 Activity.....	10
2.2.3 Intent.....	10
2.2.4 Adapter.....	10
2.3 REST API.....	11
 CHAPTER THREE: METHODOLOGY.....	 12
3.1 Data Collection.....	13
3.2 Requirement Analysis.....	13
3.2.1 Major Functional Requirements.....	13
3.2 External Interface Requirements.....	14
3.3 Other Non-functional Requirements.....	16
3.3 Architecture.....	17
3.3.1 System Block Diagram.....	17
3.3.2 Working of System.....	18
3.4 Software Process model.....	19
4.3.1Software Increments.....	20
3.5 Flow Chart.....	22
3.6 Entity Relationship Diagram.....	23
 CHAPTER FOUR: CODING AND IMPLEMENTATION.....	 24
4.1 IDE Used.....	25
4.1.1 Android Studio.....	25
4.1.2Notepad++.....	25

4.2 Designing.....	25
4.2.1 Android Asset Studio.....	25
4.2.2 Pixlr.....	25
4.2.3 HTML.....	25
4.2.4 CSS.....	26
4.2.5 XML.....	26
4.3 Programming and Scripting Language Used.....	26
4.3.1 PHP.....	26
4.3.2 Java.....	26
4.4 Database Design and Implementation.....	27
5.4.1 MySQL.....	27
4.5 Framework Used.....	27
4.5.1 Bootstrap.....	27
4.6 Server.....	27
4.6.1 Apache.....	27
4.7 REST Implementation.....	28
4.7.1 REST Request.....	28
4.7.2 REST Response.....	28
CHAPTER FIVE: EPILOGUE.....	29
5.1 Results.....	30
5.1.1 Screenshots of webpage.....	30
5.1.2 Screenshots of Android Application.....	33
5.2 Discussion.....	38
5.3 Future Enhancement.....	38
5.4 Limitations of the system.....	38
5.5 Conclusion.....	38
REFERENCES.....	39



## List of Figure

<b>Figure 2.1</b> Android Life Cycle.....	9
<b>Figure 3.1</b> Use Case Diagram.....	15
<b>Figure 3.2</b> System Block Diagram.....	17
<b>Figure 3.3</b> Incremental Model Block Diagram.....	19
<b>Figure 3.4.a</b> Web Design for Admin.....	20
<b>Figure 3.4.b</b> Design Android App for Client.....	21
<b>Figure 3.4.c</b> Implement Push Service.....	21
<b>Figure 3.5</b> Flow Chart.....	22
<b>Figure 3.6</b> Entity Relationship Diagram.....	23
<b>Figure 5.1.a</b> Login Page for Admin.....	30
<b>Figure 5.1.b</b> Welcome Page for Admin.....	31
<b>Figure 5.1.c</b> Notice Page for Admin.....	31
<b>Figure 5.1.d</b> Delete notice in Notice Page.....	32
<b>Figure 5.1.e</b> Notice List in MySQL database.....	32
<b>Figure 5.2.a</b> Splash Screen.....	33
<b>Figure 5.2.b</b> Register Screen.....	33
<b>Figure 5.2.c</b> Login Screen.....	34
<b>Figure 5.2.d</b> Notice Screen showing menu.....	34
<b>Figure 5.2.e</b> Notice list Screen.....	35
<b>Figure 5.2.f</b> Pinned post Screen.....	35
<b>Figure 5.2.g</b> Image Notice.....	36
<b>Figure 5.2.h</b> Text Notice.....	36
<b>Figure 5.2.i</b> Push Notification.....	37

## **List of Abbreviation**

API	: Application Program Interface
DB	: Database
GCM	: Google Cloud Messaging
HTML	: Hyper Text Markup Language
HTTP	: Hypertext Transfer Protocol
IEEE	: Institute of Electrical and Electronics Engineers
iOS	: iPhone Operating System
IS	: Information System
JDK	: Java Development Kit
JSON	: JavaScript Object Notation
JVM	: Java Virtual Machine
LCD	: Liquid Crystal Display
REST	: Representational State transfer
SDK	: Software Development Kit
SOAP	: Simple Object Access Protocol
UI	: User Interface
WSDL	: Web Service Definition Language
XML	: Extensible Markup Language

# **CHAPTER ONE**

## **INTRODUCTION**

## 1.1 Introduction

Notify-Me, by name states is a notification system that is a part of information system. An information system (IS) is an organized system for the collection, organization, storage and communication of information. Notification system is a combination of software and hardware that provides a means of delivering a message to a set of recipients. In information technology, the system is designed to take over the classical method of notice board system by providing in-hand notices along with push notification. The system is comprised of webpage and android application. Webpage for sending and controlling the input of notices while android application for recipients, both working on same network and server. We can use this application in various fields for quick message pop-ups. This application focuses on the notifications as for school and colleges, the students require information like class schedule changes, exam seat allocations and timings, etc. Simply by installing this app and enabling the push notification service, user can get notified without attempting any request for notifications. There can be many added features to improve and beautify the UI according to end-user requirements. This application provides list of notices on android application that could have cost paperwork on notice board along with notification to keep the end-user up to date about the notice.

## 1.2 Background

The designed app uses push notification for sending pop-up notifications to the android phones. Push notification, also called server push notification, is the delivery of information from a software application to a computing device without a specific request from the client.

Unlike pull notifications, in which the client must request information from a server, push notifications originate from a server. Typically, the end user must opt-in to receive alerts; opt-in usually takes place during the install process and end users are provided with a way to manage alerts if they change their minds later.

An important advantage of push notifications in mobile computing is that the technology doesn't require specific applications on a mobile device to be open for a message to be

received. This allows a smartphone to receive and display social media or text message alerts even when the device's screen is locked and the social media application that is pushing the notification is closed.

Different devices and services rely on different methods to deliver push notifications. Apple developers, for example, can use the Apple Push Notification Service's Developers application programming interface (APIs) to have their apps deliver push notifications to iOS devices. Another approach is to use mobile backend as a service, cloud services to provide push notification functionality for a mobile app.

### **1.3 Problem Statement**

Over the past decade, the world has become increasingly hyper-connected. We live in an environment where the Internet and its associated services are accessible and immediate, where people and businesses can communicate with each other instantly, and where machines are equally interconnected with each other. This hyper-connectivity is deeply redefining relationships between individuals, consumers and enterprises, and citizens and governments; it is introducing new opportunities but also new challenges and risks in terms of individual rights and privacy, security, cybercrime, the flow of personal data, and access to information. Information technology plays vital part in smooth flow of this system.

Notice board is intended to provide information to staff and surrounding peoples. The board itself does not notify the published information to the particular person. The person has to look for notice board for any information or changes from the administration.

1. Notification on notice board are inefficient way of notifying end-users.
2. People are not up to date about notices from these classical methods.
3. Not secure since the board is available for admin and users at same time.
4. No record of notices for user or viewers.

## 1.4 Objective

The main objectives of the project:

- Develop the application for admin and users to send and receive notices with push notification to keep users updated about published notices.

## 1.5 Scope and its application

- No need of extra hardware since application installed on android phones.
- Useful for accessing notice and information on android phones.
- Student does not need to visit notice board every time.
- Removes the paperwork of notice board system.
- Efficient and time saving.
- Timely and immediate information on notices.
- Auto update of notices on device.

The application does not need any extra hardware equipment. The application is installed in android phones of students itself. It is beneficial as it gives information with low data usage. Due to these reasons, it is quite cheaper and can be accommodated by any organizations. One can quickly access this application to get notifications and materials.

Some of its other application area are:

- Schools and colleges

It is feasible for quick information sharing between admins, teachers and students. It covers large area unlike the notice board that is rigid at a bulletin board. It can be used to share notes and information and also discuss queries among the participants.

- Event Organizers and Convention Center Management

For multiple employees working on the same event, this application can maintain the flow of work by quick supply of information. It can work as memo for the staffs, a simple feature can be used to efficiently run the events.

- Hotels

Management of hotel can be smoothly carried out by information sharing and guests can take advantage of this application for quick queries.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

## **2.1 Existing system and applications**

Push technology, or server push, is a style of Internet-based communication where the request for a given transaction is initiated by the publisher or central server. It is contrasted with pull/get, where the request for the transmission of information is initiated by the receiver or client. Push services are often based on information preferences expressed in advance. This is called a publish/subscribe model. A client "subscribes" to various information "channels" provided by a server; whenever new content is available on one of those channels, the server pushes that information out to the client. [1]

Previously we worked on Arduino with Ethernet shield to develop a message sending device that transfers any short message from server to a client. The designed html page acted as server while LCD display connected to Arduino acted as client.

This concept is reworked in Notify-Me to give proper user interface and application available in android phones. It seems easily viable for students with necessity of information. It is comparatively cheap because it is available in android phones without extra hardware and have multiple purposes.

### **Web push**

The Web push proposal of the Internet Engineering Task Force is a simple protocol using HTTP version 2 to deliver real-time events, such as incoming calls or messages, which can be delivered (or "pushed") in a timely fashion. The protocol consolidates all real-time events into a single session which ensures more efficient use of network and radio resources. A single service consolidates all events, distributing those events to applications as they arrive. This requires just one session, avoiding duplicated overhead costs. [1]

### **HTTP server push**

HTTP server push (also known as HTTP streaming) is a mechanism for sending unsolicited (asynchronous) data from a web server to a web browser. HTTP server push can be achieved through any of several mechanisms.

As a part of HTML5 the Web Socket API allows a web server and client to communicate over a full-duplex TCP connection.



Generally the web server does not terminate a connection after response data has been served to a client. The web server leaves the connection open so that if an event occurs (for example, a change in internal data which needs to be reported to one or multiple clients), it can be sent out immediately; otherwise, the event would have to be queued until the client's next request is received. Most web servers offer this functionality via CGI (e.g., Non-Parsed Headers scripts on Apache HTTP Server). The underlying mechanism for this approach is chunked transfer encoding.

Another mechanism is related to a special MIME type called multipart/x-mixed-replace, which was introduced by Netscape in 1995. Web browsers interpret this as a document changing whenever the server feels like pushing a new version to the client. It is still supported by Firefox, Opera, and Safari today, ignored by Internet Explorer. It can be applied to HTML documents, and also for streaming images in webcam applications. [2][3]

### **Google classroom**

Google Classroom is a blended learning platform for schools that aims to simplify creating, distributing and grading assignments in a paperless way. It was introduced as a feature of Google Apps for Education following its public release on August 12, 2014. Google Classroom apps are available for iOS and Android devices. With the app, teachers can create classrooms, post to classroom feeds, communicate with students, and view assignments. [4]

## **2.2 Android Concepts**

Familiarity with few concepts in android is required for understanding many of the later portions of the document. Some of the key concepts are as follows:

### **2.2.1 Android Life Cycle**

Activities in the system are managed as an activity stack. When a new activity is started, it is always placed on the top of the stack and becomes the running activity, the previous activity always remains below it in the stack and will not come to the foreground again until the new activity exits.

An activity has essentially four states.

- If an activity is in the foreground of the screen (at the top of the stack), it is active or running.
- If an activity has lost focus but is still visible (that is, a new non-full – sized or transparent activity has focus on top of our activity), it is paused. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager) but can be killed in extreme low memory situations.
- If an activity is completely obscured by another activity, it is stopped. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.
- If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.[8]

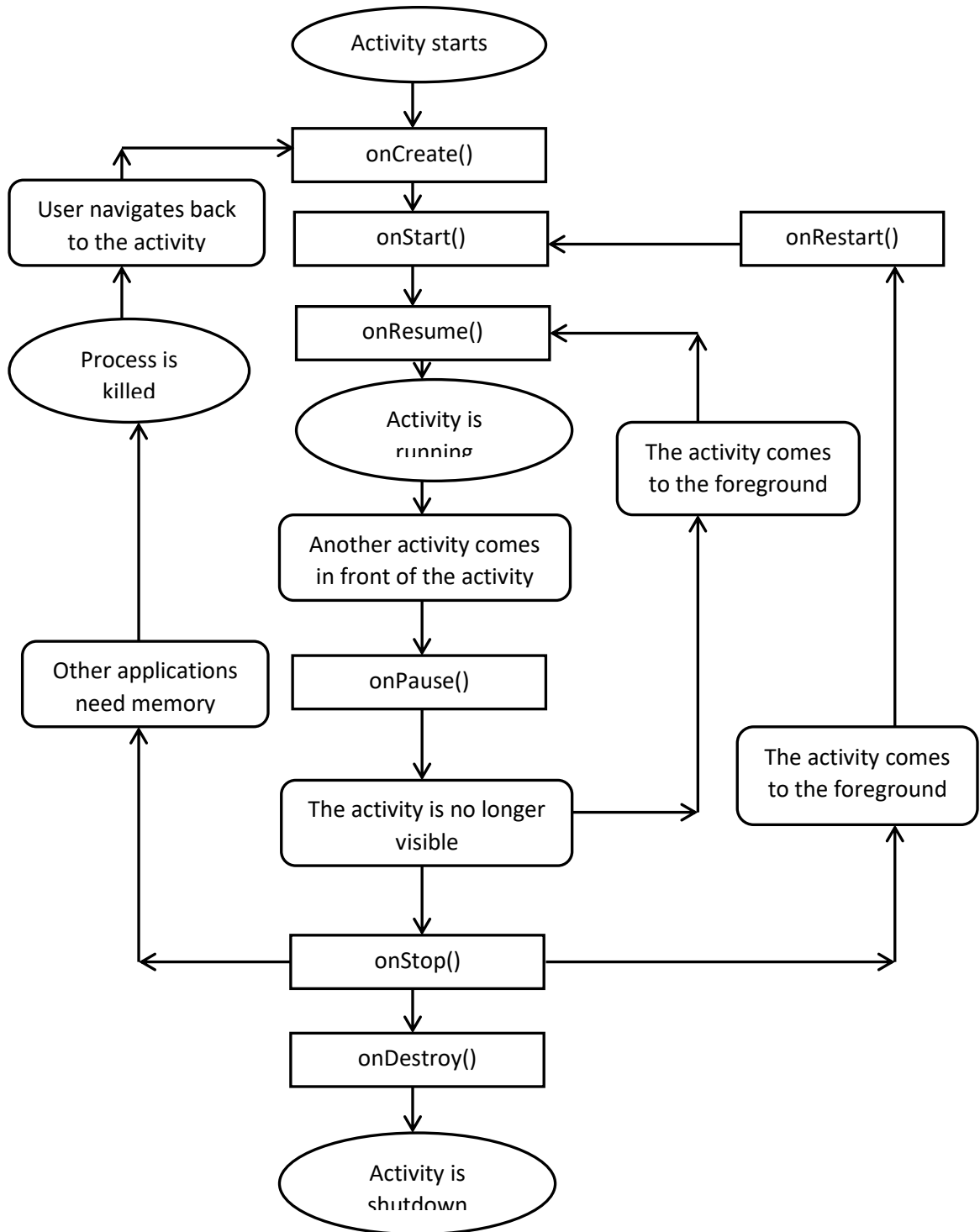


Fig. 2.1 Android Life Cycle

### 2.2.2 Activity

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with `setContentView(View)`. While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with `windowIsFloating` set) or embedded inside of another activity (using `ActivityGroup`). There are two methods almost all subclasses of Activity will implement:

- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` with a layout resource defining your UI, and using `findViewById(int)` to retrieve the widgets in that UI that you need to interact with programmatically.
- `onPause()` is where you deal with the user leaving your activity. Most importantly, any changes made by the user should at this point be committed (usually to the `ContentProvider` holding the data). [9]

### 2.2.3 Intent

An intent is an abstract description of an operation to be performed. It can be used with `startActivity` to launch an Activity, `broadcastIntent` to send it to any interested `BroadcastReceiver` components, and `startService(Intent)` or `bindService(Intent, ServiceConnection, int)` to communicate with a background Service.

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.[9]

### 2.2.4 Adapter

An Adapter object acts as a bridge between an `AdapterView` and the underlying data for that view. The Adapter provides access to the data items. The Adapter is also responsible for making a View for each item in the data set.[10]

## 2.3 REST API

Representational state transfer (REST) or RESTful web services is a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. Other forms of Web service exist, which expose their own arbitrary sets of operations such as WSDL and SOAP.

REST uses XML and JSON formats for data interchange. REST is often used in mobile applications, social networking websites, mashup tools, etc. The REST style emphasizes that interactions between clients and services is enhanced by having a limited number of operations.

## **CHAPTER THREE**

### **METHODOLOGY**

### **3.1 Data Collection**

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypothesis, and evaluate outcomes. [5]

The primary data source will be generated by the webpage provided to the admin. The server gets all notices from the admin webpage by which a chunk of data is created for each linked end-user viewing notices. The data can be text and images with the sender and timestamp on each notice.

The secondary data in the system are the tokens generated during the register of the end-user devices. These tokens are used to request push notification when the application is inactive.

### **3.2 Requirement Analysis**

#### **3.2.1 Major Functional Requirement**

##### **User Requirements**

- The system shall have form to sign up to register to an account of server.
- The system shall require user to have a valid email id.
- The system shall be capable of verifying the user's email by sending registration verification email.
- The system shall have a sign in form via the application for user to login to his/her account using valid username and password.
- The system shall provide a list of subjective notices sorted by date.
- The user shall be able to scroll through the notices and information.

##### **Administrator Requirements**

- The system shall provide a username and password for the administrator.
- The system shall provide a login page for administrator access administrative panel.
- The system shall allow administrator to send notices to database.

### **3.2.2 External Interface Requirement**

#### Hardware Interface

- i. Android phone

This is fairly obvious, the android application requires a phone running android as the operating system. The minimum required version is Android 4.4.0 KitKat as the SDK used in the application are supported from this version.

- ii. PC

A computer running a windows or a Linux as operating system with any web browser will be required to run the web application going to be developed for the system. The web application is exclusive to only administrative users.

#### Software Interface

- i. Web browser

Any web browser (Chrome, Firefox) will be required to run the web application with administrative privileges.



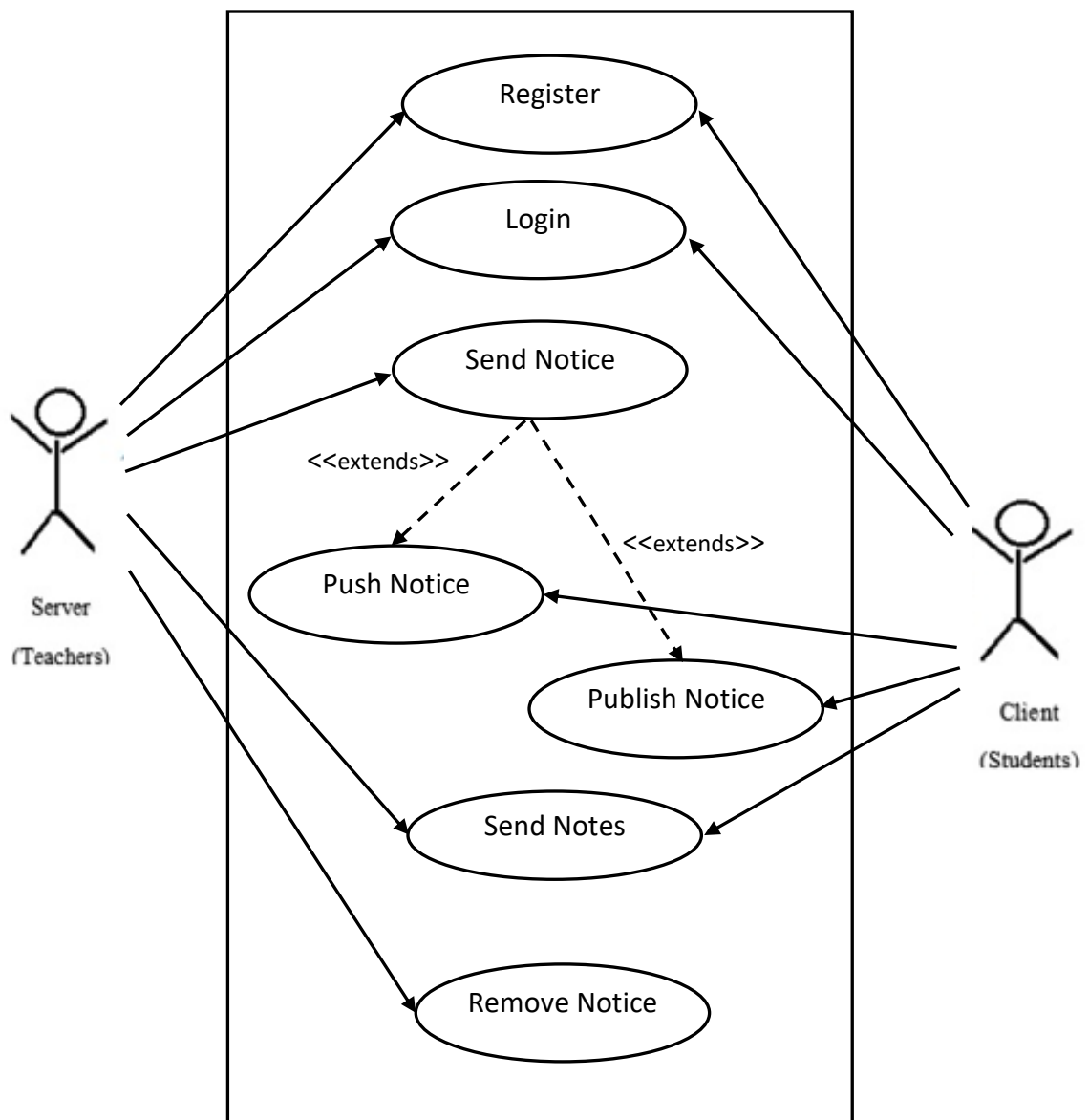
User Interface

Fig. 3.1 Use case diagram of Notify-Me

### Communication Interface

#### i. Internet

A reliable internet connection will be required on the user front to connect to the system.

#### ii. Administrative user

The administrative user will access the database hosted by the server through the web application.

#### iii. User

The user interface to access the system will be android application. The user must have internet connection on phone to be able receive services.

### **3.2.3 Other Non-functional Requirement**

#### Product Requirements

##### i. Usability Requirements

- The system shall employ familiar user interface design or application for customer ease of use.

##### ii. Performance Requirement

- The system shall have minimal content load time.
- The system shall have fast database update.

##### iii. Dependability Requirements

- The system shall have database design compliant ACID properties.
- The system shall be fault tolerant in carrying out transaction functions.

##### iv. Security Requirements

- The system shall be robust to security risks through various security measures like assigning different passwords to admin and user which shall be encrypted.

### Organizational Requirements

#### i. Operational Requirements

- The system shall deliver list of notices.
- The system shall be able to send push service for added notices.
- The system shall be capable of classifying user and destined users accordingly via database update.

#### ii. Development Requirement

- The system development shall employ incremental process model.

#### iii. Environmental Requirement

- The system shall support Windows and Linux as operating system with capable of running Chrome and Firefox web browsers.
- The system shall have minimum of requirement of Android 4.4 KitKat for android application.

## 3.3 Architecture

### 3.3.1 System Block Diagram

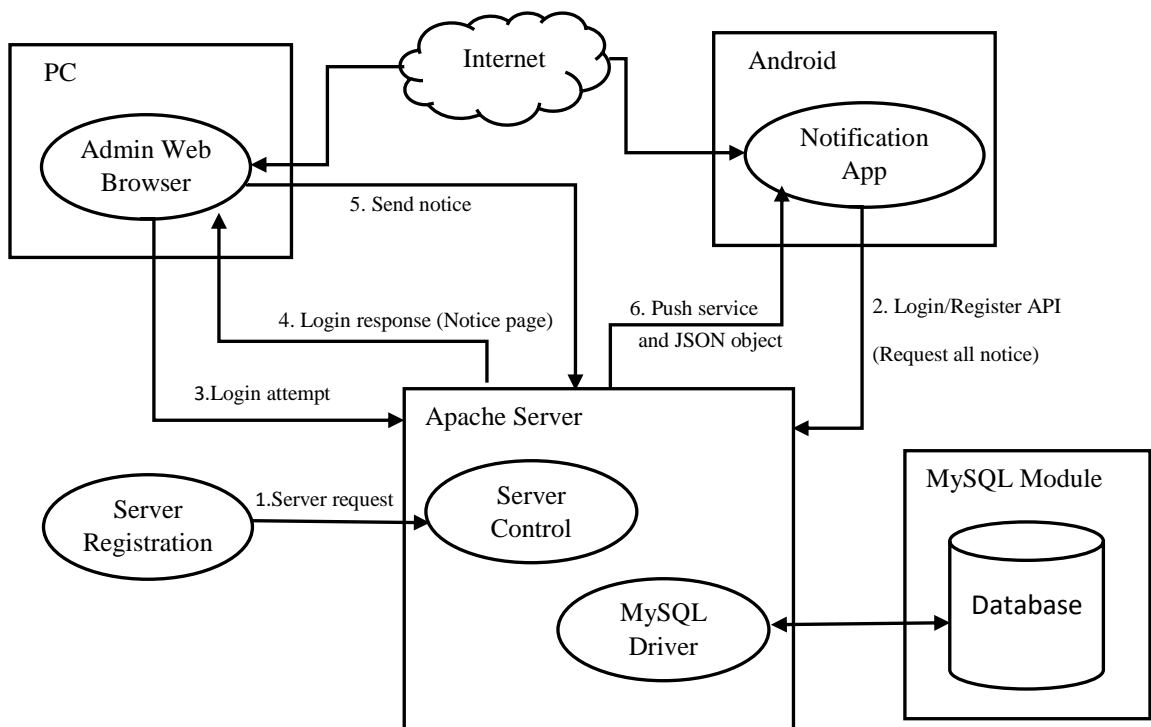


Fig. 3.2 Block diagram of system

### 3.3.2 Working of System

1. Admin registers a server for the organization.
2. The registered organization stores previous notices on MySQL database.
3. Client logs in to organization.
4. Client requests registration through API.
5. Client receives all the respective notices from server.
6. Admin attempts login request.
7. Server responds to notice page for services.
8. Admin sends notice that stores in database.
9. Clients receives notification through push service.

#### GCM Mechanism of Push Service

Google Cloud Messaging also now called as Firebase Cloud Messaging, functions using server APIs and SDKs, both maintained by Google. The GCM has the ability to send push notifications, deep-linking commands, and application data. Larger messages can be sent with up to 4 KB of payload data. Upon allowing the application permission to receive and display notifications, the client application sends a registration API request to the Google Cloud Messaging interface to begin the registration process. The GCM Service receives and acknowledges the request and responds by giving the device a GCM Registration ID, a unique identifier that the developer later uses to send a notification to the individual device. The identifier is stored onto the device, and is typically sent to the developer's application server to be stored.

The GCM Registration ID is a randomly-generated identifier that does not contain any personal or device information that could allow a developer to discover the personal identity of the user. When the developer wishes to send a notification event to a device, the process begins with an API POST request being sent to the GCM Authentication Service. The POST request includes the GCM Registration ID, priority, optional values and links, and the information that is to be displayed on the device upon its arrival. Upon successful verification of the GCM Registration ID and other credentials, an authentication token is

returned. Both identifiers are then sent to the GCM Service to be enquired and delivered to the device.

### 3.4 Software Process Model

The software process model used is the **Incremental Model**. The first increment is often a core product where the basic requirements are addressed and the supplementary features are added in the next increments. The core product is used and evaluated by the client. Once the core product is evaluated by the client there is plan development for the next increment. Thus, in every increment the needs of the client are kept in mind and more features and functions are added and the core product is updated.

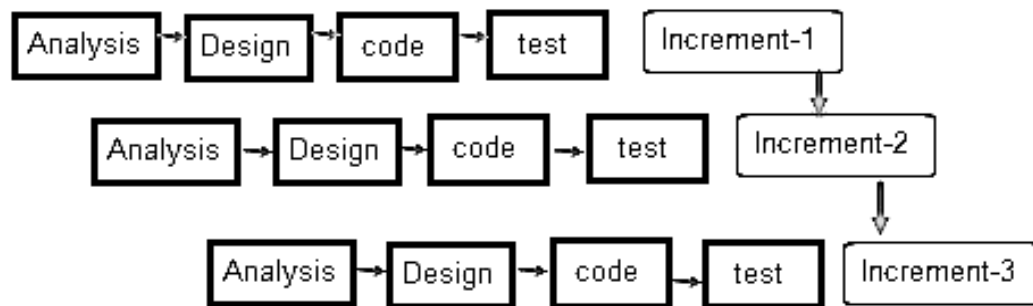


Fig. 3.3 Incremental Model block diagram

The phases of Incremental model are as follow:

#### (a) Analysis and requirements

In this phase, the requirements for the software are established through discussion with project members and are then documented.

#### (b) Designing

In this phase, the requirements are converted into design to be used in the next phase. The design includes various elements such as database structures, software architecture and procedures. The design is first passes through the quality test and then documented

#### (c) Coding

In this phase, the actual coding of the software is done. The design of the previous phase is converted into the code.

### (d) Testing

In this phase, the output generated is checked to ensure that it matches the requirements. The programs developed in the previous phases are checked for the logical and syntax errors.

### (e) Maintenance and Support

The software developed needs to maintenance and support. This refers to the changes as well as new requirements in the software after delivery.

The above phases are repeated for every step-up in the version, features, requirements, etc.

## 3.4.1 Software Increments

### Increment-1: Web Design for Admin

- Analyze server requirements and database.
- Design web page for admin section.
- Code to implement and link webpage to server.
- Perform tests to confirm the success of webpage launch.

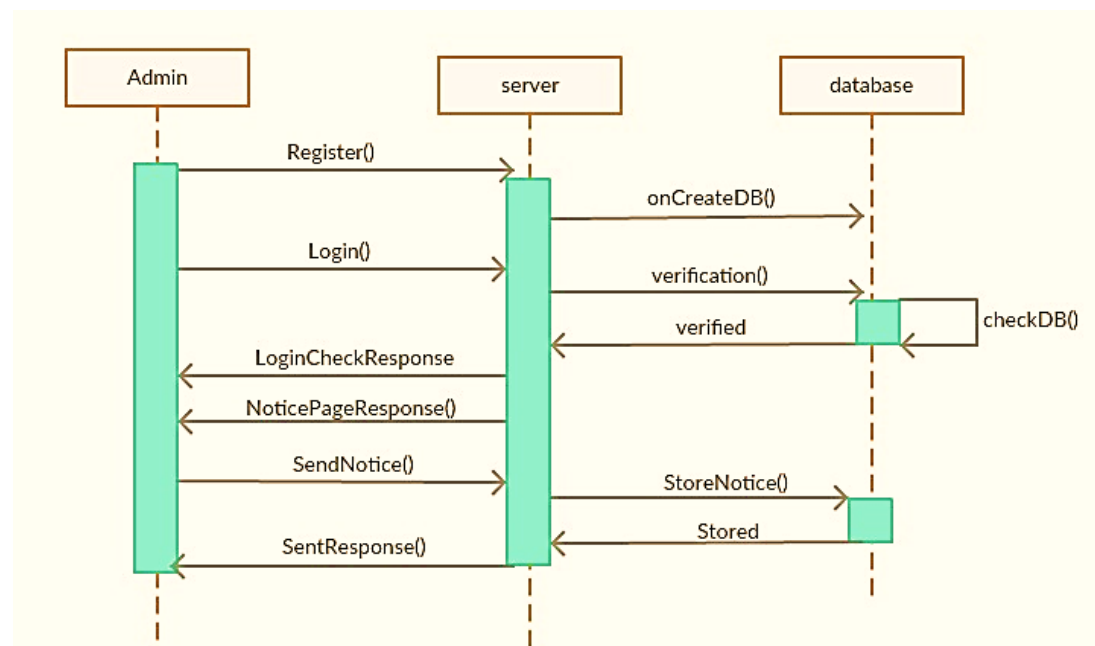


Fig. 3.4(a) Sequence diagram of webpage

**Increment-2 : Android App Design for Client**

- Analyze android platform, servers and database.
- Design android for client section.
- Code to implement and link android to server.
- Perform tests to confirm the success of android launch.

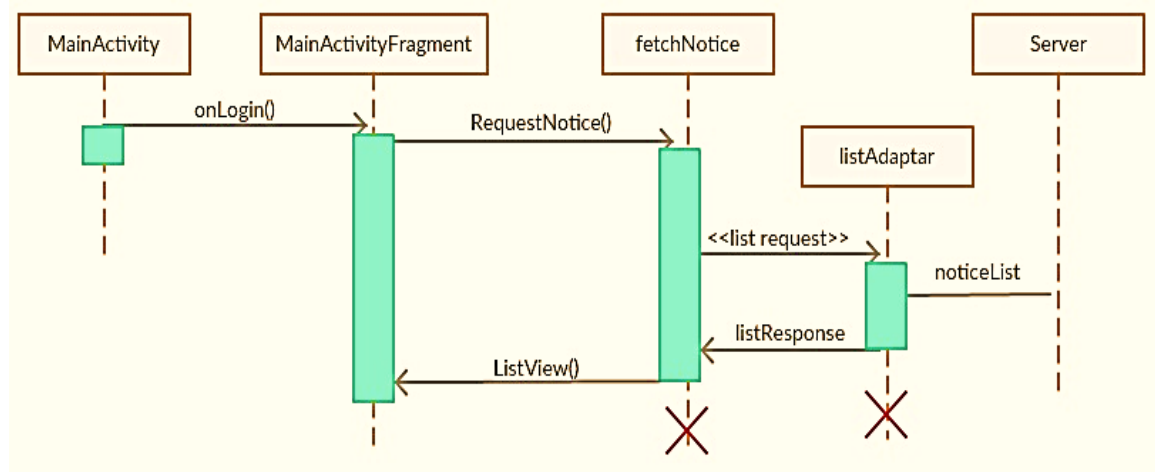


Fig. 3.4(b) Sequence diagram of android listview

**Increment-3 : Implement Push Service**

- Analyze android platform, firebase and push services.
- Design push service mechanism.
- Code to implement and perform android push service.
- Perform tests to confirm the success of push service.

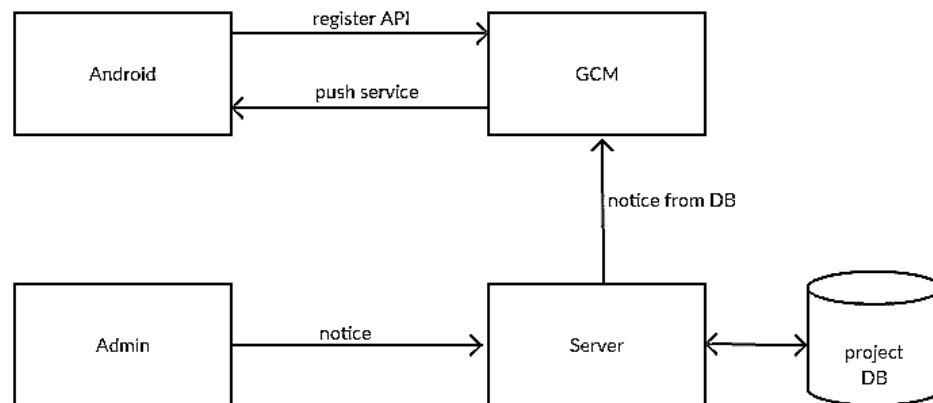


Fig. 3.4.(c) Block diagram of push service

### 3.5 Flow Chart

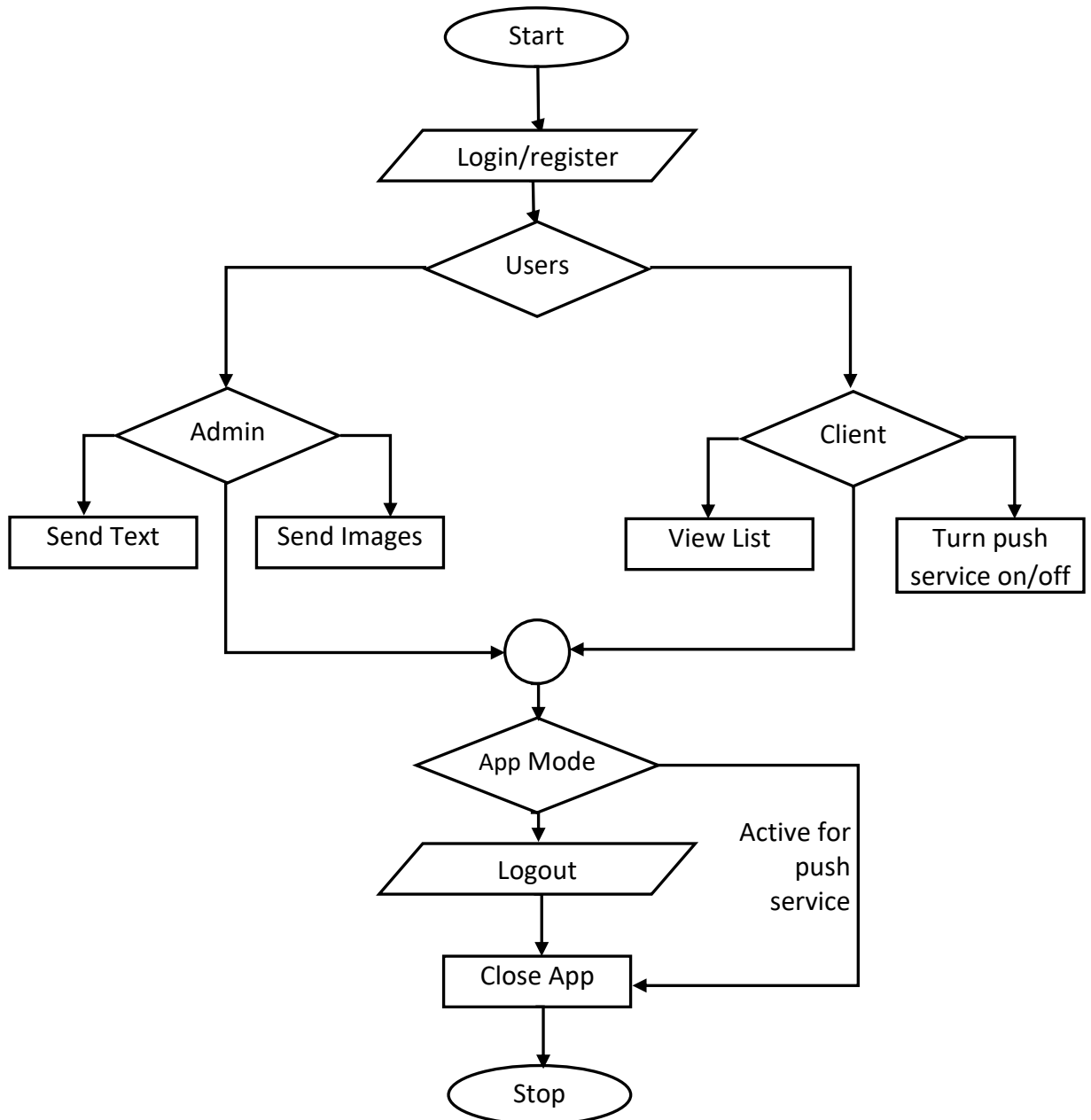


Fig. 3.5 Flow chart of the system



### 3.6 Entity Relationship Diagram

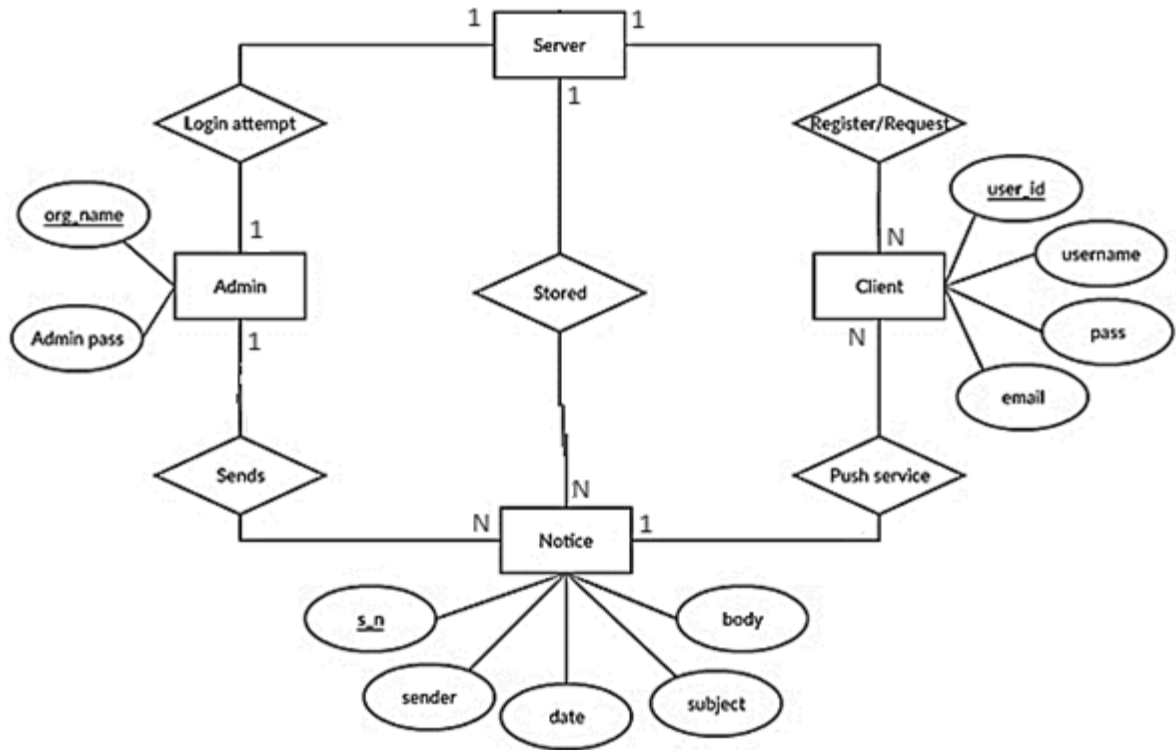


Fig. 3.6 ER-diagram of the system

Database design was done as shown in the ER-diagram. This ER-diagram works for a single organization with admin and client. For multiple organization, database shares unique and primary key 'organization name' throughout the admin and client. This unique key makes sure that user only gets valid notices according to the organization. A sequence of operations takes place with the database while sending and receiving notices along the diagram.

- Admin attempts login with organization name and admin password
- Client registers and login with organization name and user password.
- Admin sends notices that gets stored in database and sent to users.
- Client receive push notice through cloud while publish list of notices from server.

## **CHAPTER FOUR**

### **CODING AND IMPLEMENTATION**

## **4.1 IDE Used**

### **4.1.1 Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built based on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

### **4.1.2 Notepad++**

Notepad++ is a text editor and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window. The project's name comes from the C increment operator. Notepad++ is distributed as free software. At first the project was hosted on SourceForge.net, from where it has been downloaded over 28 million times, and twice won the SourceForge Community Choice Award for Best Developer Tool.

## **4.2 Designing**

### **4.2.1 Android Asset Studio**

Android Asset Studio, and contains a variety of tools you can use to generate icons for your Android app. A collection of tools to easily generate assets such as launcher, notification and action bar icons for your Android app.

### **4.2.2 Pixlr**

Pixlr is a cloud-based set of image tools and utilities, including a number of photo editors, a screen grabber browser extension, and a photo sharing service. The suite was intended for non-professionals, however the apps range from simple to advanced photo editing. It can be used on PCs, and on smartphones or tablets using a mobile app.

### **4.2.3 HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML

documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

#### **4.2.4 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

#### **4.2.5 XML**

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. Android parses the XML using Java to create views according to what is defined in the XML resources.

### **4.3 Programming and Scripting Languages Used**

#### **4.3.1 PHP**

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. It has been used for development of admin side webpage backend.

#### **4.3.2 Java**

Java is a general-purpose computer programming language that is concurrent class-based, object-oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled

to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

## **4.4 Database Design and Implementation**

### **4.4.1 MySQL**

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing and is an important component of an open source enterprise stack called LAMP. LAMP is a Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language.

## **4.5 Frameworks Used**

### **4.5.1 Bootstrap**

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

## **4.6 Server**

### **4.6.1 Apache**

Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Most commonly used on a Unix-like system (usually Linux), the

program is available for Microsoft Windows as well. Version 2.0 improved support for non-Unix, e.g. Windows and OS/2.

## 4.7 REST Implementation

### 4.7.1 REST Request

POST is a request method supported by the HTTP protocol used by the World Wide Web. By design, the POST request method requests that a web server accept the data enclosed in the body of the request message, most likely for storing it. Some of the POST request along with parameters are:

localhost/notifyme/login	->	org_name, org_pass
localhost/notifyme/sendMultiplepush	->	id, token
localhost/notifyme/notice	->	date, subject, detail, etc.

### 4.7.2 REST Response

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

Example of JSON object:

```
myObj= {
  "name":"John",
  "age":30,
  "cars": [
    { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
    { "name":"BMW", "models":[ "320", "X3", "X5" ] },
    { "name":"Fiat", "models":[ "500", "Panda" ] }
  ]
}
```

## **CHAPTER FIVE**

### **EPILOGUE**

## 5.1 Results

The screenshots below show the system UI for admin and clients. The webpage developed using HTML, PHP having MySQL database hosted in Apache server have features required by admin. The screenshots below also show the android application developed for client. The application the features to go through the list of notices and view the text and images of notice sent by admin. The client eventually receives push notification from the admin on the request sent by admin. The figures below show the detail of the result accomplished on the project.

### 5.1.1 Screenshots of webpage

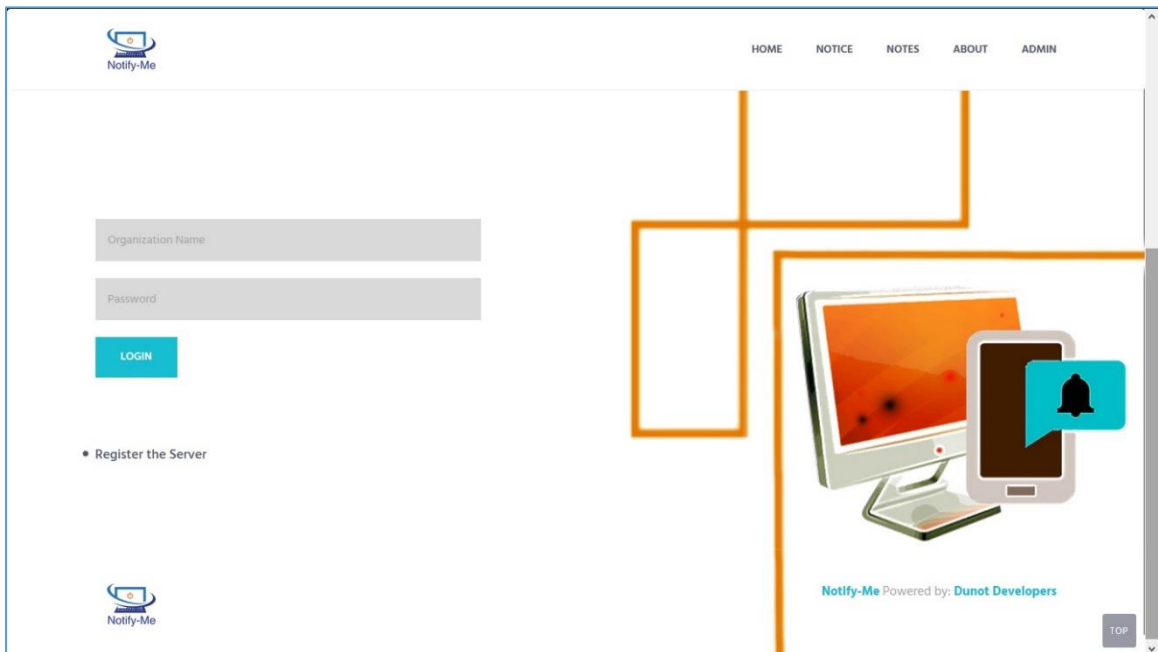


Fig. 5.1(a) Login page for Admin



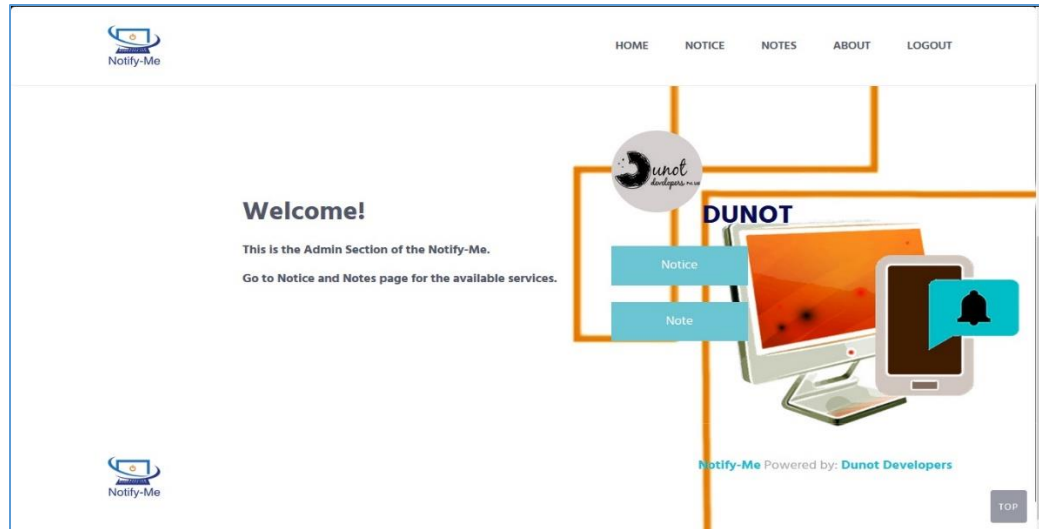


Fig. 5.1(b) Welcome page for Admin

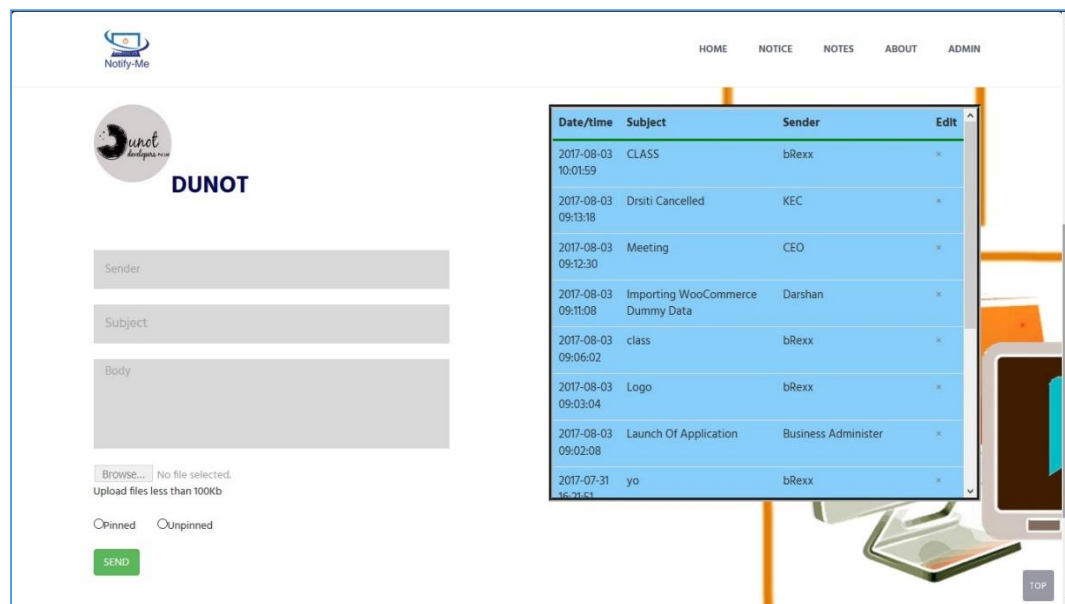


Fig. 5.1(c) Notice page for Admin

The notice page contains the list of previously sent notices by different admins. The list is categorized by name of sender, title of notice and ordered by the date of submission. The admin can send text and images using this page. By clustering algorithm, this page gets only the affiliated notice list that is managed by the admin.

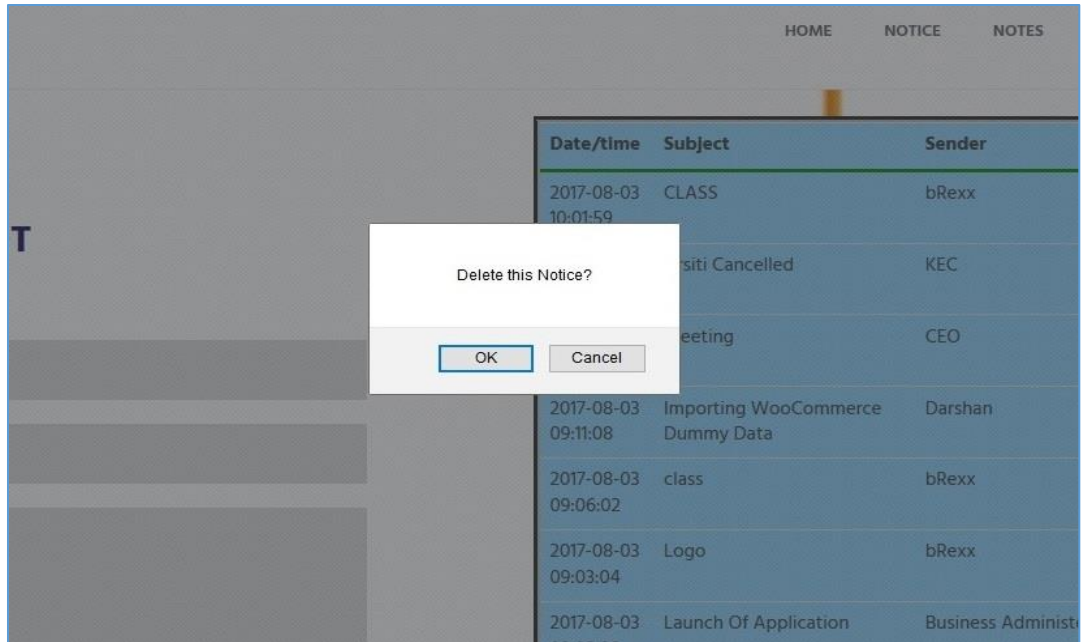


Fig. 5.1(d) Delete notice in Notice page

		id	date	subject	sender	body	pin	attachment	org_name
<input type="checkbox"/>	Edit Copy Delete	47	2017-07-30 14:35:16	Parking To all the students of new building	Administration	The parking area in front of the basketball court ...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	46	2017-07-30 14:34:24	National Interaction Competition	Administration	ACEM is organizing national interaction competitio...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	45	2017-07-30 14:33:55	Final Report Submission	Administration	The final report of the major project are to be su...	Yes		dunot
<input type="checkbox"/>	Edit Copy Delete	44	2017-07-30 14:33:38	Revision class	Administration	To all the faculties-Revision classes are going to...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	43	2017-07-30 14:33:18	Annual Magazine v2.0	Administration	All the students whose articles are selected for a...	Yes		dunot
<input type="checkbox"/>	Edit Copy Delete	42	2017-07-30 14:32:53	Annual Magazine	Administration	All the students who want to publish their article...	Yes		dunot
<input type="checkbox"/>	Edit Copy Delete	41	2017-07-30 14:32:29	Fee deadline	Administration	All the students are requested to clear their dues...	Yes		dunot
<input type="checkbox"/>	Edit Copy Delete	40	2017-07-30 14:30:47	Interaction class	Administration	All the students of BCT are requested to be presen...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	39	2017-07-30 14:30:06	Annual Calendar available	Administration	The Class Representative of all faculties are requ...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	38	2017-07-30 14:29:11	Admit card for Back exams	Administration	All the students appearing for back exams can coll...	No		dunot
<input type="checkbox"/>	Edit Copy Delete	37	2017-07-29 20:59:56	Notice	bRexx	New Localhost at work	No	d6f53ac5ad223c55e6e1	asd
<input type="checkbox"/>	Edit Copy Delete	36	2017-07-27 23:57:45	Launch Of Application	Business Administer	At common law, notice is the fundamental principle...	No		asd
<input type="checkbox"/>	Edit Copy Delete	35	2017-07-27 23:57:31	Launch Of Application	bRexx	At common law, notice is the fundamental principle...	No		asd
<input type="checkbox"/>	Edit Copy Delete	34	2017-07-27 23:57:07	Launch Of Application	Business Administer	At common law, notice is the fundamental principle...	No		asd
<input type="checkbox"/>	Edit Copy Delete	33	2017-07-27 23:55:07	Launch Of Application	brexx	At common law, notice is the fundamental principle...	No		asd
<input type="checkbox"/>	Edit Copy Delete	32	2017-07-27 23:54:42	Launch Of Application	bRexx	At common law, notice is the fundamental principle...	No		asd
<input type="checkbox"/>	Edit Copy Delete	31	2017-07-27 14:16:21	Importing WooCommerce	brexx	asdaqfww	No	18921204_1457428460	asd

Fig. 5.1(e) List of notices in MySQL database

By delete function in notice page, the list of notice can be managed by the admin. One can go through the list and check on the notice sent and remove unnecessary notices. The database gets updated as the changes in the webpage takes place.

### 5.1.2 Screenshots of Android Application

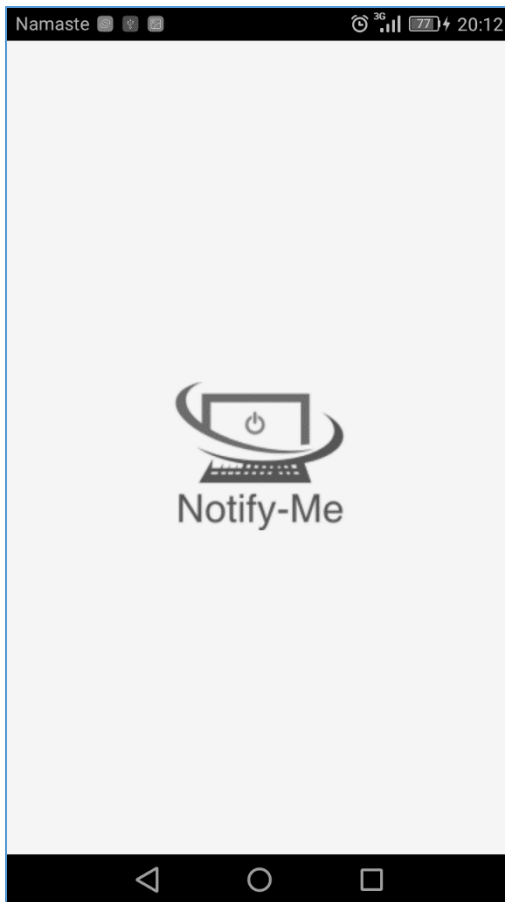


Fig. 5.2(a) Splash Screen

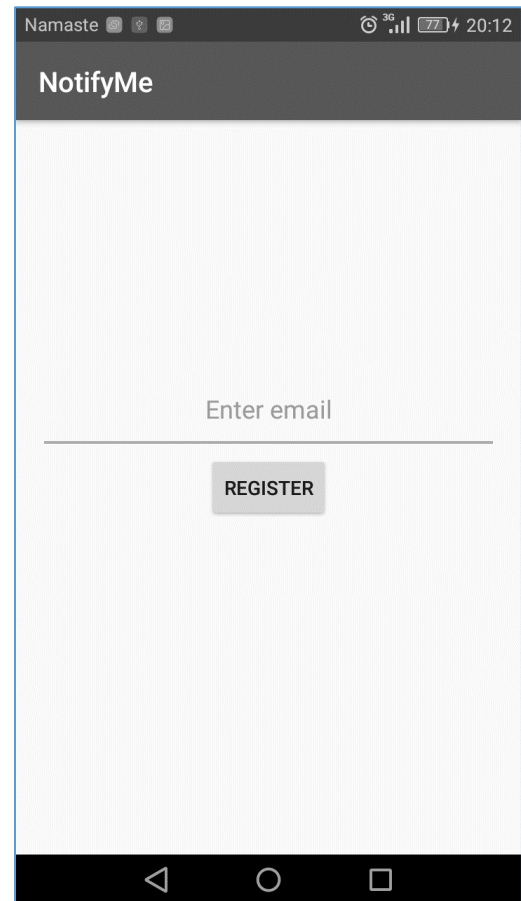


Fig. 5.2(b) Register Screen

Splash Screen in the android application holds the function of checking whether the device is previously registered. The registered device saves the token generated by API for push service. If the device is not registered, the splash screen redirects the screen to register screen where the user can apply for device registration and new token is generated. After the process, the splash screen directs the screen to Login Activity that holds the login screen for end-user. Developer can change the way register works by asking number of details (like name, email, faculty, id, etc) of the end-user before registration.

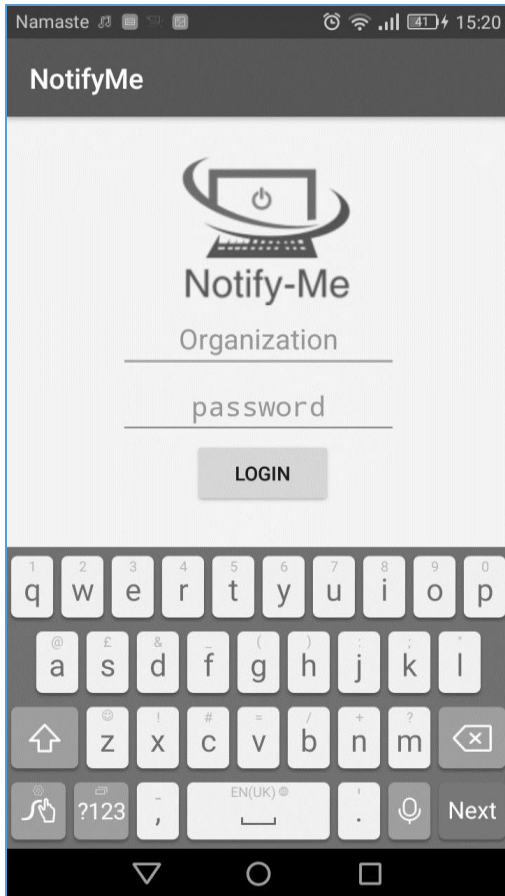


Fig.5.2(c) Login Screen

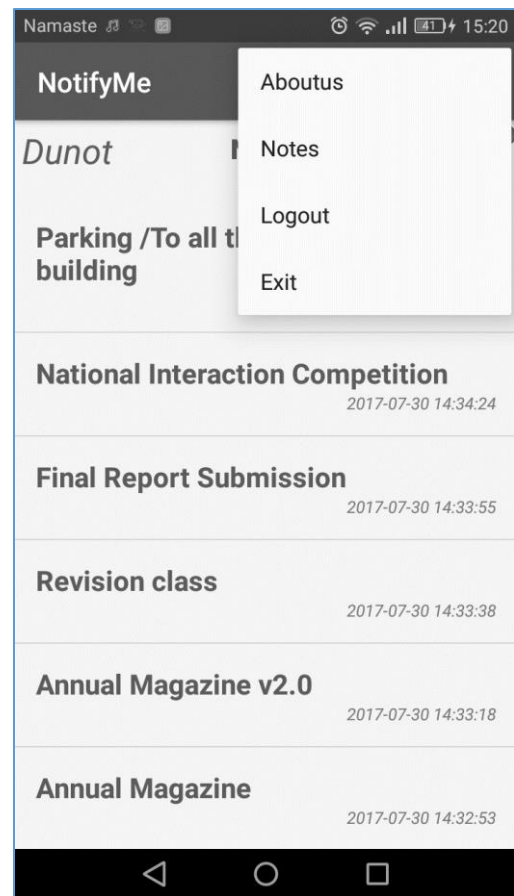


Fig.5.2 (d) Notice Screen showing menu

The figure above shows the login and logout feature which manipulates how the device receives the notification while inactive. The push service works only when the device is logged in even though the application is not running. The end-user can login then exit device without logging out. The menus in the Notice screen or home screen contains the logout button as well as exit button.

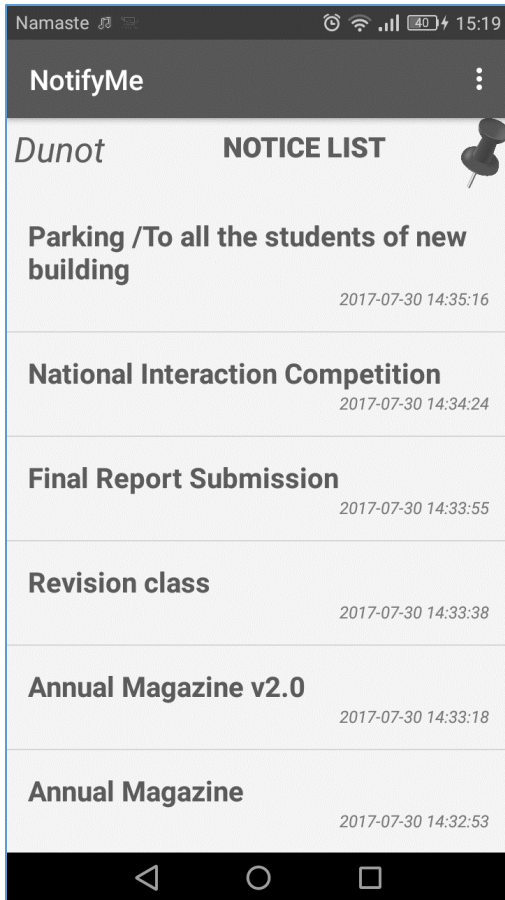


Fig.5.2 (e) Notice list Screen

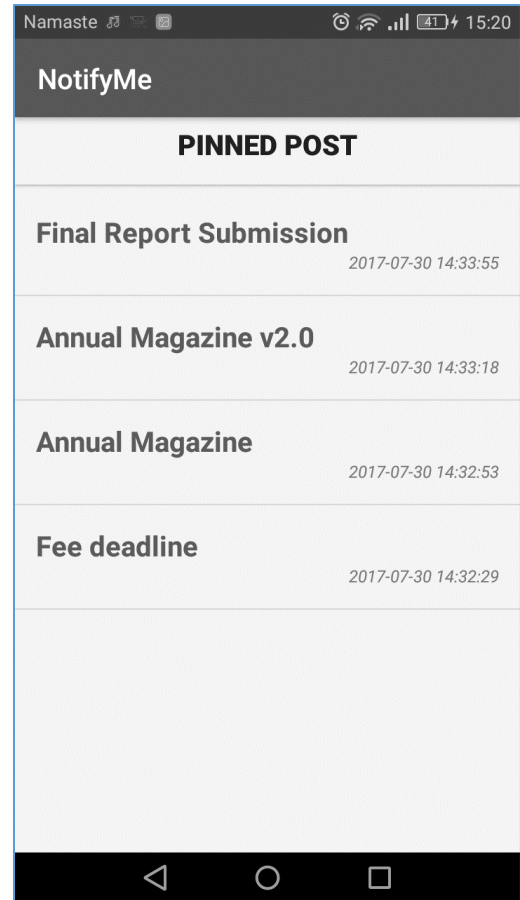


Fig.5.2 (f) Pinned post Screen

The figure above shows the screen of android application containing the scroll view of list of notices. The notices are received from JSON object from the server and by clustering algorithm affiliated list of notice are displayed in the screen. The admin holds the key to put the notices in main list or pinned list. The pinned list is supposed to make the user access the importance notices quickly. Some notices can be important for longer time than usual, such notices are to be pinned in another list for easier and faster access.

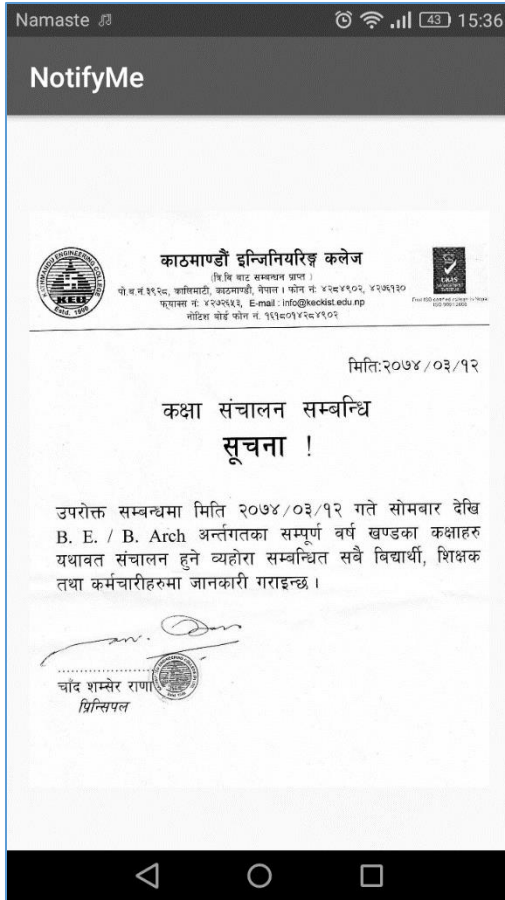


Fig.5.2 (g) Image Notice

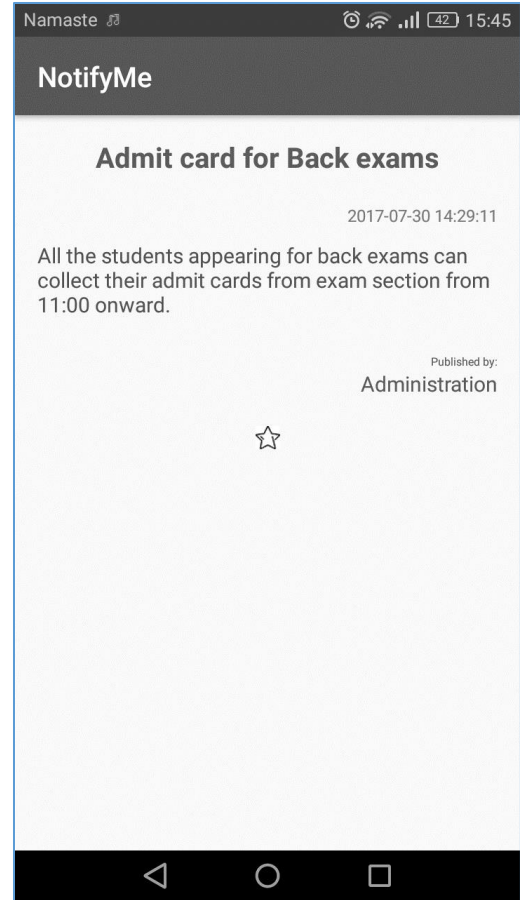


Fig.5.2 (h) Text Notice

The figure above shows the screen that displays the detail view of notices selected by user on list. The list view contains ‘onclicklistener’ that directs to detail screen with views for image and text. The detail view takes array position from array adapter of the list and displays the selected array value. The image is displayed using Picasso that transfers quick images from the URL which is why the image size of the notice should be comparatively small (~100 kB).



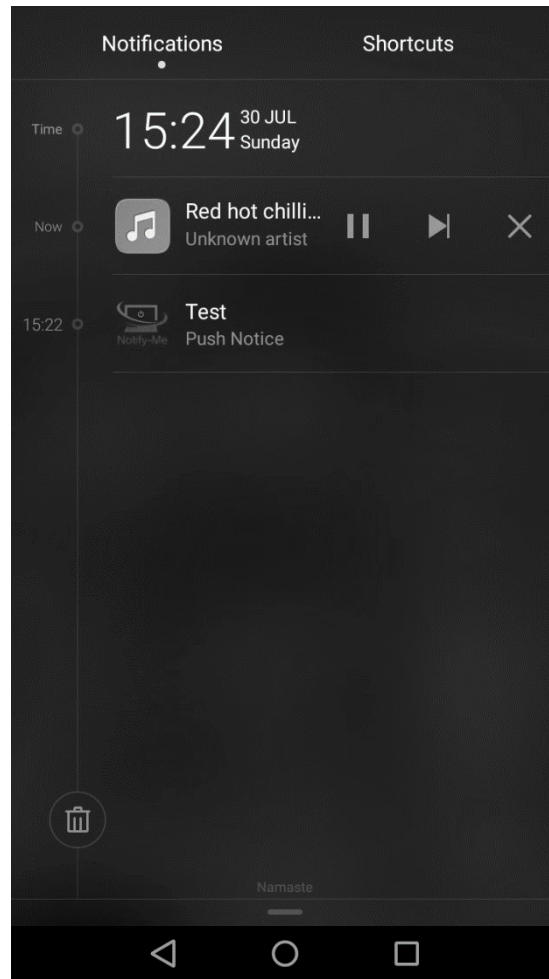


Fig.5.2 (i) Push notification

The above figure shows the psuh notificaiton on the notification bar. The end-user recieves push notification along with sound notificaiton from the admin request to GCM.

### **5.3 Discussion**

In this project we not only applied programming knowledge but also of software product development method and strategy as well as communication with clients and marketing techniques. Before doing this project, we were unknown to many topics related to this field which was a challenging task to tackle the real-world problem.

After facing many problems during this project, some got solved in no time while others bugged for many days. The admin webpage that runs on Apache server was built using HTML and PHP to access and manipulate MySQL database. The android application developed on android SDK has features to go through list of notices and pinned posts with push service.

### **5.4 Future Enhancement**

After successfully exhibiting the mentioned features in the system and android app, further enhancement can be achieved. With incremental model, the system can be upgraded timely with added features and interface. The system can be updated to give higher quality images and features such as zoom in and out can be put into effect. Instead of MySQL i.e. Relational Database Management System, NoSQL technology can be used to better accommodate the scale of number of notices and organization that uses the server. Enhancing user experience and making app more interactive.

### **5.5 Limitations of the system**

1. JSON parsing takes longer for large data array.
2. Internet connection required every time to view list.
3. Corrupt file when downloading notes i.e. pdf, doc files via local server.

### **5.6 Conclusion**

Ergo, the objective of this project was successfully achieved. The notification system was built for Android operating system. Timely and proper implementation of Notify-Me will definitely ease the organizations in sharing notices with update to users.



## REFERENCES

- [1] M. Thomson, E. Damaggio and B. Raymor (October 22, 2016). "Generic Event Delivery Using HTTP Push". Internet Draft. Internet Engineering Task Force. Retrieved October 28, 2016.
- [2] CGI Programming on the World Wide Web O'Reilly book explaining how to use Netscape server-push
- [3] Server-Push Documents (HTML & XHTML: The Definitive Guide) O'Reilly book explaining server-push
- [4] Cloud Messaging | Google Developers (2017)  
<https://developers.google.com/cloud-messaging/>
- [5] Push notification introductions (2017)  
<http://searchmobilecomputing.techtarget.com/definition/push-notification>
- [6] Notifications | Android Developers (2017)  
<https://developer.android.com/guide/topics/ui/notifiers/notifications.html>
- [7] Working of FCM/GCM (2017)  
<https://firebase.google.com/docs/cloud-messaging/>
- [8] Activity Life Cycle | Android Developers (2017)  
<https://developer.android.com/guide/components/activities/activity-lifecycle.html>
- [9] Intent | Android Developers (2017)  
<https://developer.android.com/reference/android/content/Intent.html>
- [10] Adapter | Android Developers (2017)  
<https://developer.android.com/reference/android/widget/Adapter.html>
- [11] Android JSON Parse Images and Texts (2017)  
<http://www.androidbegin.com/tutorial/android-json-parse-images-and-texts-tutorial/>