



FARHAN ULLAH KHAN

01-134132-046

MUHAMMAD AWAIS

01-134132-109

# Notes Oriented Tractable Application (NOTA)

**CS16-69**

Bachelor of Science in Computer Science

Supervisor: Dr. MUHAMMAD MUZAMMAL

Department of Computer Science  
Bahria University, Islamabad

# Certificate

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS (CS).

---

Dr. Faisal Bashir  
Head of Department

---

Dr. Muhammad Muzammal  
Supervisor

---

Internal Examiner

---

External Examiner

## **Acknowledgements**

In the name of Allah, the Most Gracious and the Most Merciful. Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this project. We would like to dedicate this work to our parents, whose love, encouragement, motivation and prayers made us achieve this success. To our teachers who taught us to the very best. To our friends for always being there for us in the need and most importantly to our supervisor Dr. Muhammad Muzammal who helped us throughout our project.

## **Abstract**

With the advancement in the field of computer science, the size and weight of the computer reduced and its efficiency has increased. It has influenced our life in the form of smart phones. A smart phone now days not only performs the basic function of a phone (to serve the purpose of communication) but also comes with many other features.

NOTA is an android application which will help users to store notes. There are applications that allow users to store text and reminders (Notes, Calendar) but in this application, there are some addition, such as it features allows user to store multimedia. In existing notes and calendar application, user can only save text and reminders. We have combined these application, and provide user to do both things in a single application. Some application does provide user with a facility to set a reminder but they do not provide a workspace to do that task, therefore user visits different applications to complete the task.

This application not only stores simple text notes but have many other options, user can attach multimedia (pictures, video, audio) with his notes, other than that user can also attach text files (word, power point) with it. User can set a reminder with his note or even can set a reminder to do a specific task, user can even edit and delete their notes. User can also share his notes with other users by creating a group. The interface of this application is simple and user friendly, when the user login there is a dashboard which will be displaying all the notes or tasks created by that user. NOTA also uses cloud services to store user data, which can be accessed from any android device.

# Table of Contents

<b>1. Introduction .....</b>	<b>10</b>
1.1. Overview .....	10
1.2. Project Statement .....	11
1.3. Problem Description .....	11
1.4. Project Objective .....	11
1.4.1. Academic Objectives .....	11
1.4.2. Product Objectives .....	11
1.5. Project Scope .....	12
1.6. Outline .....	12
<b>2. Literature Review .....</b>	<b>14</b>
2.1. Notes Application .....	14
2.1.1. Color Notes .....	14
2.1.2. Omni Notes .....	15
2.1.3. Apple Notes.....	17
2.1.4. NOTA.....	18
<b>3. Requirement Specification .....</b>	<b>20</b>
3.1. Application Overview.....	20
3.2. Basic Functionality .....	20
3.3. Functional Requirements .....	20
3.3.1. Cloud Integration .....	20
3.3.2. Notes .....	20
3.3.3. Groups.....	21
3.3.4. Push Notification.....	21
3.3.5. Login .....	21
3.3.6. Multimedia and Reminder.....	21
3.4. Non-Functional Requirements.....	22
3.4.1. Design Principle .....	22
3.4.2. Security .....	22
3.4.3. Reliability.....	22
3.4.4. Portability .....	22
3.4.5. Timing.....	23
3.4.6. Extensibility .....	23
3.5. Performance Requirements.....	23
3.6. User Interface .....	23
3.7. Use cases.....	24
3.7.1. Main Use Case .....	24
3.7.2. Use Case 1.....	25
3.7.3. Use Case 2.....	26

3.7.4.	Use Case 3.....	27
3.7.5.	Use Case 4.....	28
3.7.6.	Use case 5.....	29
3.7.7.	Use Case 6.....	30
<b>4.</b>	<b>System Design .....</b>	<b>32</b>
4.1.	System Architecture .....	32
4.2.	Deployment Diagram .....	33
4.3.	Database Design .....	34
4.4.	System Sequence Diagram .....	35
4.5.	Process Interaction Models.....	36
<b>5.</b>	<b>Methodology .....</b>	<b>41</b>
5.1.	Tools and Technologies.....	41
5.1.1.	Android Studio .....	41
5.1.2.	Firebase .....	41
5.1.3.	Butter Knife.....	41
5.1.4.	Glide.....	41
5.2.	Languages Used.....	42
5.3.	Modules Description.....	42
5.3.1.	Login .....	42
5.3.2.	Creating a note .....	43
5.3.3.	Cloud Database .....	44
5.3.4.	Group.....	45
5.3.5.	Dashboard .....	45
5.3.6.	Navigation troll .....	46
<b>6.</b>	<b>System Testing and Evaluation .....</b>	<b>48</b>
6.1.	Software Testing Techniques .....	48
6.2.	Function Testing .....	48
6.3.	Performance Testing.....	49
6.4.	Acceptance Testing.....	50
6.5.	Test Cases .....	51
6.5.1.	Test Case 1. Installation of Application .....	51
6.5.2.	Test Case 2. Register and Login the User .....	52
6.5.3.	Test Case 3. Creating a New Note .....	53
6.5.4.	Test Case 4. Options in a Note.....	54
6.5.5.	Test Case 5. Creating a Group .....	55
6.5.6.	Test Case 6. Share the notes.....	56
6.5.7.	Test Case 7. Logout from the application .....	57
<b>7.</b>	<b>Conclusion .....</b>	<b>59</b>
	<b>References .....</b>	<b>61</b>

## List of Figures

Figure 2.2.1: Color Note Application .....	14
Figure 2.1.1: Omni Notes.....	16
Figure 2.2.3: Apple NotesS.....	17
Figure 2.2.4: NOTA.....	18
Figure 3.7.1: Main Use Case.....	24
Figure 3.7.2: Use Case 1 .....	25
Figure 3.7.3: Use Case 2 .....	26
Figure 3.7.4: Use Case 3 .....	27
Figure 3.7.5: Use Case 4.....	28
Figure 3.7.6: Use Case 5 .....	29
Figure 3.7.7: Use Case 6.....	30
Figure 4.1: System Architecture .....	32
Figure 4.2: Deployment Diagram .....	33
Figure 4.3: Database Design.....	34
Figure 4.4: Sequence Diagram.....	35
Figure 4.5 (a): Process Model 1 .....	36
Figure 4.5 (b): Process Model 2.....	37
Figure 4.5 (c): Process Model 3.....	38
Figure 4.5 (d): Process Model 4.....	39
Figure 5.3 Modular diagram .....	42
Figure 5.4.3: Cloud Database.....	44
Figure 5.4.4: Group Synchronization.....	45
Figure 5.4.5: Dashboard.....	46
Figure 5.4.6: Navigation Troll .....	46

## List of Tables

Table 2.2.1: Pros and Cons of Color Note .....	15
Table 2.1.2: Pros and Cons of Color Notes .....	16
Table 2.1.3: Pros and Cons of Apple Notes.....	18
Table 2.2.4: Pros and Cons of NOTA.....	18
Table 3.7.2: Use Case 1 .....	25
Table 3.7.3: Use Case 2 .....	26
Table 3.7.4: Use Case 3 .....	27
Table 3.7.5: Use Case 4 .....	28
Figure 3.7.6: Use Case 5.....	29
Figure 3.7.7: Use Case 6.....	30
Table 4.5 a: Process Model 1 .....	36
Table 4.5 b: Process Model 2.....	37
Table 4.5 c: Process Model 3.....	38
Table 4.5 d: Process Model 4.....	39
Table 6.6.1: Test Case 1.....	51
Table 6.6.2: Test Case 2.....	52
Table 6.6.3: Test Case 3.....	53
Table 6.6.4: Test Case 4.....	54
Table 6.6.5: Test Case 5.....	55
Table 6.6.6: Test Case 6.....	56
Table 6.6.7: Test Case 7.....	57



# Chapter 1

# Introduction

---

# 1. Introduction

In this chapter, we will give introduction of our project that will include the facts and figures that why this application is being made, on what platform we are working, problem description, what is the objective of our project, scope or defining boundaries of the project.

## 1.1. Overview

In this age of information technology computers are playing their role in every field of life and their involvement is increasing day by day. They have proven to be faster, efficient and reliable than humans in every possible way. That is why in today's world people greatly rely on computer generated information. All the processes that were done manually are now being automated.

With the advancement in the field of computer science, the size and weight of the computer reduced and its efficiency has increased. It has influenced our life in the form of smart phones. A smart phone now days not only performs the basic function of a phone (to serve the purpose of communication) but also comes with many other features. Therefore, the idea is to make an application on android platform because 66% people in the world are android users, and plus it is cheaper than iPhone [8].

The application designed (NOTA), is an android application that facilitates user to store notes in the form of text as well as multimedia. There are applications that allow users to store text and reminders (Notes, Calendar) but in this application, there are addition such as it features allows user to store multimedia. In existing notes and calendar application, user can only save text and reminders. We have combined these application, and provide user to do both things in a single application. These application does provide user with a facility to set a reminder but they do not provide a workspace to do that task, therefore user must visit different applications to complete the task. For example, if there is a student who sets a reminder for his due assignment, NOTA allows the user to view his assignment with in the application. It also allows the user to edit that file so that the user does not has to visit some other application to complete his task. Other than this, the application also has functionalities like to save, view and edit multimedia.

## **1.2. Project Statement**

NOTA is a digital notebook for capturing and organizing everything across your devices. Jot down your ideas, keep track of classroom and meeting notes or make a to do list.

## **1.3. Problem Description**

There are many applications which are used to add notes and reminders but these applications cannot help user to complete or do their tasks. NOTA can add reminders, text files and multimedia as well. There is a dashboard displaying user tasks. On clicking the tasks, application gives related task information, where user finds its saved files and if the task needs some other application to complete the user task, NOTA will take user to related application on a single click. User can also add its playlist for some event. User can add its own gallery. All the multimedia files and notes will be saved using cloud services.

## **1.4. Project Objective**

The following are the project objectives.

### **1.4.1. Academic Objectives**

To learn how to apply the concepts learned from the course work, to solve the practical problems and to explore how to research and learn new tools and technologies that are expected of a fresh graduate in the software industry.

### **1.4.2. Product Objectives**

The objective of this project is to develop multimedia notes is your digital notebook for capturing and organizing everything across your devices. Jot down your ideas, keep track of classroom and meeting notes, or make a to do list.

## **1.5. Project Scope**

NOTA will store common multimedia format. It will also be able to edit those multimedia files. This application is providing user with a workspace to complete different tasks but application is not concerned with the completion of the task.

User creates an identification (id) and password by which he will create before using this application.

NOTA will provide a user with a facility to create groups and add users in it, with the help of user's email id the group admin will add the users in a group. The user can share his notes in that group. A user can create more than one group. And when a user sends some note in that group a push notification is generated.

## **1.6. Outline**

The rest of the report is organized as following.

Chapter 2: The literature review of the project.

Chapter 3: Requirements of the project.

Chapter 4: Design of system.

Chapter 5: In this chapter, the methodology of project is discussed.

Chapter 6: In this chapter, test cases are designed.

Chapter 7: In the last there is a conclusion.

Chapter 2

Literature Review

---

## 2. Literature Review

In this chapter, developed applications that store notes other than NOTA are discussed.

### 2.1. Notes Application

An overview of some of the existing notes applications.

#### 2.1.1. Color Notes

Color Notes is a note taking application that allows you to make text notes and lists. The name of the application defines its feature, it has the capability to transforming the background color of notes to help you stay organized. Some other features include calendar, backup support to both internal and cloud storage, some task reminder features, and more [1].

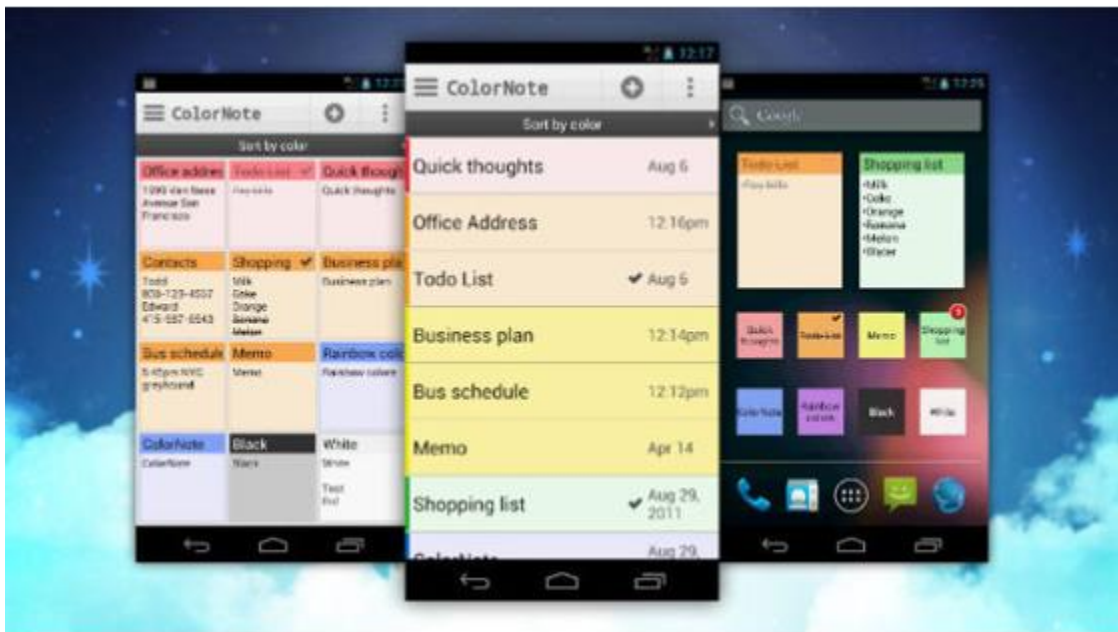


Figure 2.2.1: Color Note Application

The following are the pros and cons of this application.

Pros	Cons
<ol style="list-style-type: none"> <li>1. Color Note is a simple, color-coded note taking app that also uses sticky note style home screen widgets to give you quick access to your note from your home screen.</li> <li>2. This application allows you to create text notes and lists.</li> <li>3. Its namesake feature is the ability to change the background color of notes to help you stay organized.</li> <li>4. This is one of the good application if you are taking simple notes.</li> </ol>	<ol style="list-style-type: none"> <li>1. This application only save notes in the form of text whereas NOTA has option to add multimedia and text files in it.</li> <li>2. NOTA automatically sets notes and reminders priority as the due date arrives.</li> <li>3. NOTA is user friendly and have many more features as compared to this application as you can share your notes.</li> <li>4. You can create groups and share and send your notes in NOTA which this application does not provide.</li> </ol>

Table 2.2.1: Pros and Cons of Color Note

### 2.1.2. Omni Notes

Omni Notes is another very simple note taking application with a good design interface. This one uses a card layout, that is both easy to scroll through and easy to keep track of the notes. It also can combine, sort and search through your notes for better organization and discovery. And above all that, it has clock, widgets and a sketch note mode where you can draw if you want to. It's a free app that has enough features to be economical [2].



Figure 2.1.1: Omni Notes

The following are the pros and cons of this application.

Pros	Cons
<ol style="list-style-type: none"> <li>1. This application uses a vertical card layout that is both easy to scroll through and easy to keep track of.</li> <li>2. This application can merge, sort, and search through your notes for better organization and discovery</li> <li>3. This application has a sketch-note mode where you can draw and doodle if you want to.</li> <li>4. This is one of the good application if you are taking simple notes.</li> </ol>	<ol style="list-style-type: none"> <li>1. This application only save notes in the form of text whereas NOTA has option to add multimedia and text files in it.</li> <li>2. NOTA automatically sets notes and reminders priority as the due date arrives.</li> <li>3. Our application is user friendly and have many more features as compared to this application as you can share your notes.</li> <li>4. You can create groups and share and send your notes in NOTA which this application does not provide.</li> </ol>

Table 2.1.2: Pros and Cons of Color Notes



### 2.1.3. Apple Notes

With Notes in iOS 9, you can create checklists, change formatting, add attachments, insert photos or videos and sketch. To delete a note, tap or swipe left over the note in the Notes list. User can take a new photo or video, or tap photo library to add existing photos and videos to your note. User can look for the note he created last week or month. User can search for a specific note from the top of the Notes list. Scroll to the top of the Notes list to reveal the search field, then enter what you're looking for. Then you can use that password or your fingerprint to unlock your protected notes [3].

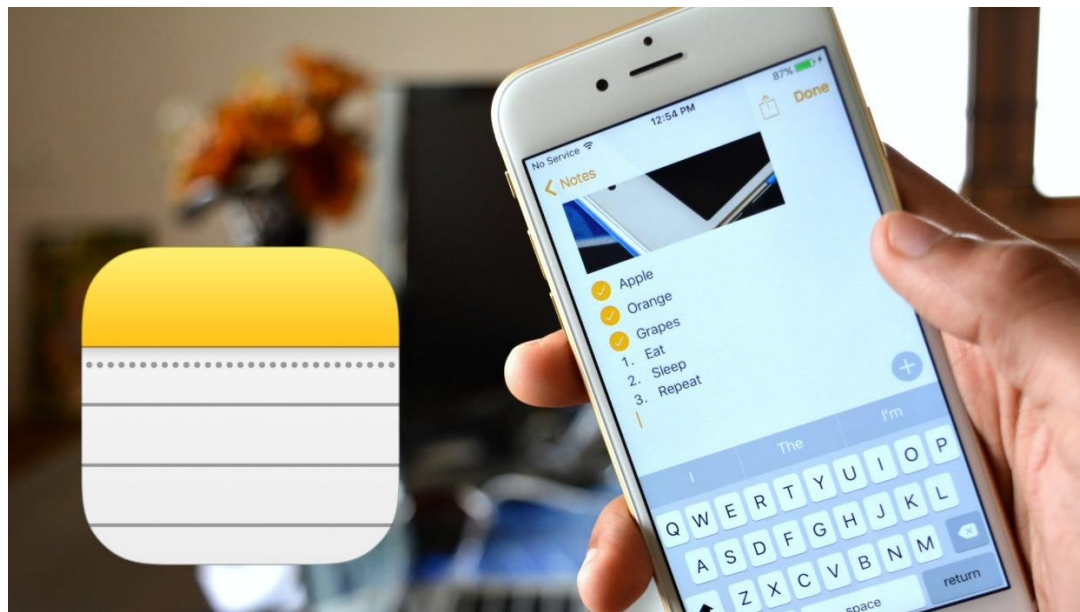


Figure 2.2.3: Apple Notes

The following are the pros and cons of this application.

Pros	Cons
<ol style="list-style-type: none"><li>1. Notes in iOS 9, you can create checklists, change formatting, add attachments, insert photos or videos, and sketch.</li><li>2. This application also has the ability to merge, sort, and search through your notes for better organization and discovery</li></ol>	<ol style="list-style-type: none"><li>1. This application only available in iOS but it not there for android user, as NOTA is android based.</li><li>2. NOTA automatically sets notes and reminds as the due date arrives.</li><li>3. NOTA is user friendly and have many more features as compared to this</li></ol>

<ol style="list-style-type: none"> <li>3. Deleted notes can also be recovered. To recover a deleted note, some simple steps must be taken.</li> <li>4. This is one of the good application if you are taking simple notes.</li> </ol>	<p>application as you can share your notes.</p> <ol style="list-style-type: none"> <li>4. You can create groups and share and send your notes in NOTA which this application does not provide.</li> </ol>
---	---

Table 2.1.3: Pros and Cons of Apple Notes

**2.1.4. NOTA**

This application is an android application that facilitates user to store notes in the form of text as well as multimedia. This application will store common multimedia format. It will also be able to edit those multimedia files. Users can access their data and can view their notes from different devices as the data will be stored on cloud.

Figure 2.2.4: NOTA

<b>Pros</b>	<b>Cons</b>
<ol style="list-style-type: none"> <li>1. In this application user, can create notes.</li> <li>2. User can set reminders for a specific note or task.</li> <li>3. User can attach multimedia and text files with notes.</li> <li>4. User can share notes by creating a group.</li> <li>5. The NOTA data is saved on cloud, so can be accessed on multiple devices.</li> <li>6. The interface of NOTA is user friendly.</li> </ol>	<ol style="list-style-type: none"> <li>1. The deleted notes cannot be recovered.</li> <li>2. NOTA does not have sketch mode.</li> <li>3. The notes created are not displayed in different colors.</li> </ol>

Table 2.2.4: Pros and Cons of NOTA

# Chapter 3

## Requirements Specifications

---

## **3. Requirement Specification**

The requirements of our project are discussed in this chapter.

### **3.1. Application Overview**

NOTA is expected to maximize customer satisfaction and maximize marketing by enabling the buyer to do much more than just writing down notes. This application is an android application that facilitates user to store notes in the form of text as well as multimedia. There are some applications that allows user to store text and reminder (Notes, Calendar) but in NOTA, there are some addition such as it will allow user to store multimedia. NOTA is your digital notebook for capturing and organizing everything across your devices. Jot down your ideas, keep track of classroom and meeting notes, clip from the web, or make a to do list.

### **3.2. Basic Functionality**

As soon as user launches this application a dashboard appears, so that on the dashboard, there would be different options from which user will chose his desire option by clicking that option. The user will navigate within the application by pressing different options.

### **3.3. Functional Requirements**

#### **3.3.1. Cloud Integration**

All the data of user is accessible from any devise, only user must login to his account. This is possible by providing the user with cloud services. With Cloud Infrastructure as a service, our application receives a private cloud that is quite like the classic understanding of clouds but is used only for our application purposes.

#### **3.3.2. Notes**

User will be able to create simple text notes with. Those text notes will be displayed to the user on the main dashboard. That dashboard will contain all the notes that are created by the user. User can view all those notes that he has created.

### **3.3.3. Groups**

A user can create multiple groups and can add different users in it. In that group, they can share their notes and files. In a group, you can place note which everyone (the users which are in that group) can see. With the help of user's email id, the group admin will add users in a group. This group will be created by the user before using this application. Only those notes and files which are added by a user will be visible to all those users which are added in a group, if a user does not want that other users should see their notes they just simply don't have to add them in to group. There will be multiple dashboards, for different groups.

All that is shared in a group will be visible to all the users in that group. They can see that note and but cannot edit that note.

### **3.3.4. Push Notification**

Enable push notifications on the device, the application will generate a notification whenever the time of the reminder arrives. The time of the reminder will be set by the user, when the phone clock reaches that time the notification will be generated and will alert the user of his task.

### **3.3.5. Login**

There will be a login id and password for every user by which users will be differentiated. The id and password will be created by the user before the user start using this application. This id and password will help user to use this application from any device. The user id will also be used when adding members in a group. With the help of this user id the user can be found by other users and can be added in to the group.

### **3.3.6. Multimedia and Reminder**

This application will save multimedia files and reminders. The multimedia will include images, videos and audio files. These files are attachable with notes. The user will select his multimedia from his devise. A single note can contain multiple multimedia files.

The user can also set reminder in that note or task. The user must select time and date at which the NOTA will remind the user for that note or task.

## **3.4. Non-Functional Requirements**

The following are the nonfunctional requirements of the application.

### **3.4.1. Design Principle**

A mobile app that doesn't include design principles is destined to fail when pitted against the many consumer apps that have made their way into the workplace.

A mobile application should take advantage of the full screen, sensibly positioning elements in a natural hierarchy that reflects the relationships between them. Consider how users will direct those elements to carry out a specific task. Now for example, smartphone users trust heavily on their thumbs to navigate a User Interface and most those users are right handed, so we will consider placing the most important features within easy reach.

When making decisions about fonts, colors, styles, titles, buttons and icons we will keep readability and usability in mind. Fonts should be visible on different screen resolutions. We will pick contrasting colors that clearly delineate one element from the next.

Above all, we will keep the User Interface clean and simple, without a lot of mess.

### **3.4.2. Security**

The application should be secure, no other user should be allowed to access another user data, until it is allowed or shared by the user. As the application, may require any sort of confidential information.

### **3.4.3. Reliability**

The application should operate reliably. Some other content should not be displayed against some other notes.

### **3.4.4. Portability**

The nature of the application is itself portable as it is being developed for hand held android devices.

#### **3.4.5. Timing**

The application should response in seconds.

#### **3.4.6. Extensibility**

Incase updates of the application are available; the user should be able to update the application.

### **3.5. Performance Requirements**

The smart phone should have the minimum system requirements for the application. The processor, the memory and the android version should be compatible with the ones the application is designed for.

### **3.6. User Interface**

In this section the graphical user interface is discussed along with the tools used to develop it.

#### **a. GUI**

The application has a relatively simple graphical user interface. There are menus and different pages. When the application is launched, there will first be a login page, which will take the user to a dashboard, where user notes will be displayed, from there user can open the menu and can navigate through the application.

#### **b. Tools**

The graphical user interface was developed in “android studio”.

### 3.7. Use cases

#### 3.7.1. Main Use Case

The use case diagram show how the user can interact with the system.

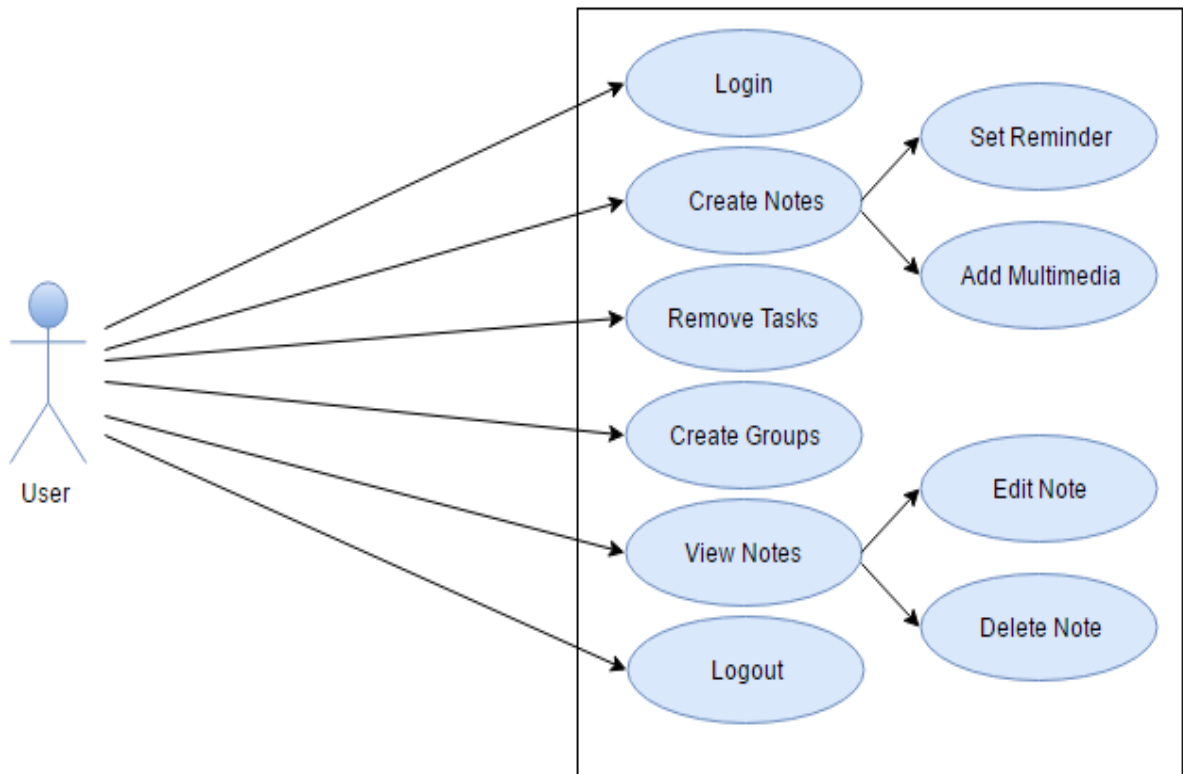


Figure 3.7.1: Main Use Case



### 3.7.2. Use Case 1



Figure 3.7.2: Use Case 1

<b>Use Case Id</b>	UC 01.
<b>Title</b>	Login.
<b>Description</b>	Asks the user to enter their username and password.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	The user has to open the application
<b>Steps</b>	<ol style="list-style-type: none"><li>1. User will open the application.</li><li>2. After an application is opened there will be a login page displayed.</li><li>3. After entering a valid login id and password user will enter the application.</li></ol>
<b>Post Condition</b>	User can use the application.

Table 3.7.2: Use Case 1

### 3.7.3. Use Case 2

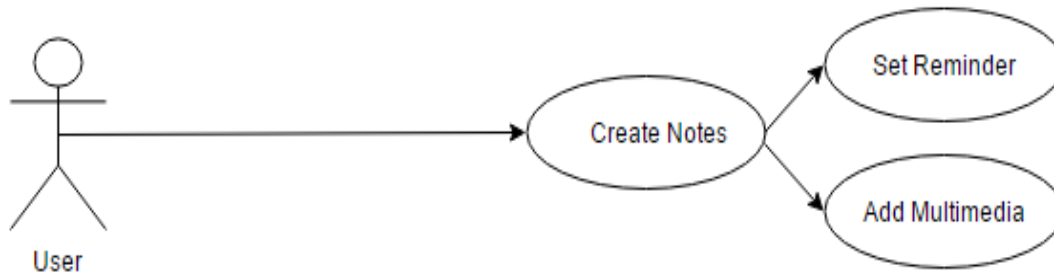


Figure 3.7.3: Use Case 2

<b>Use Case Id</b>	UC 02.
<b>Title</b>	Create Note.
<b>Description</b>	User creates a new task with the help of text, multimedia, text files and reminders.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	The login is done and user has chosen to create a new task or note.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Login to the application.</li> <li>2. There will be a create option on the main page.</li> <li>3. The user will click on create new note.</li> <li>4. User can attach multimedia, text files and reminder in it.</li> <li>5. After making the note user will click on save button.</li> </ol>
<b>Post Condition</b>	Task is saved and can be viewed and edited for further use.

Table 3.7.3: Use Case 2

### 3.7.4. Use Case 3

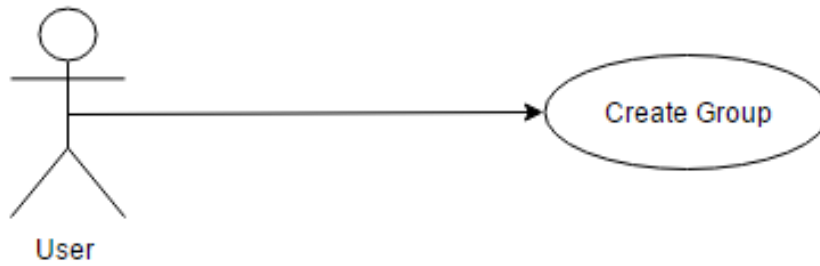


Figure 3.7.4: Use Case 3

<b>Use Case Id</b>	UC 03.
<b>Title</b>	Create Groups.
<b>Description</b>	To make a group in which there will be multiple users and one of the user will be admin. That user can share notes in that group and can manage that group.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	User has to login to the application to create a group.
<b>Steps</b>	<ol style="list-style-type: none"><li>1. The user has opened the application.</li><li>2. He has clicked on the create button.</li><li>3. Options will appear, the user will click on create group.</li><li>4. The user is going to insert other user's id in order to add them in to a group.</li><li>5. After clicking on save the group will be created.</li></ol>
<b>Post Condition</b>	A group will be created in which user can share and send notes to specific group of users.

Table 3.7.4: Use Case 3

### 3.7.5. Use Case 4



Figure 3.7.5: Use Case 4

<b>Use Case Id</b>	UC 04.
<b>Title</b>	Share Note.
<b>Description</b>	If a user wants a share a note that he has created. User can do it in a group.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	There must be note made by the user.
<b>Steps</b>	<ol style="list-style-type: none"><li>1. The user opens the application.</li><li>2. The user selects the note he wants to share.</li><li>3. The user selects the group he wants to share it with.</li><li>4. The note is successfully shared.</li></ol>
<b>Post Condition</b>	That note will be available to users to whom the note is shared.

Table 3.7.5: Use Case 4

### 3.7.6. Use case 5

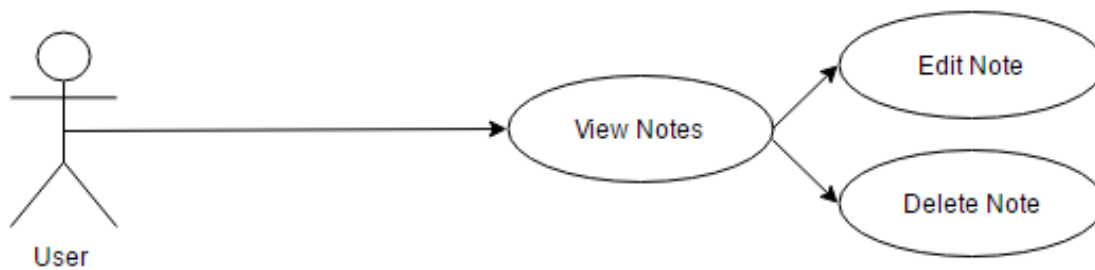


Figure 3.7.6: Use Case 5

<b>Use Case Id</b>	UC 05.
<b>Title</b>	View Notes.
<b>Description</b>	A user view the notes that he has created. The user can edit those notes and can delete those notes.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	A note should be created before the user can view it.
<b>Steps</b>	<ol style="list-style-type: none"><li>1. The user opens the application.</li><li>2. The user clicks on a specific note he wants to view.</li><li>3. User has the option to delete and edit that note.</li></ol>
<b>Post Condition</b>	User views the notes that he has created.

Figure 3.7.6: Use Case 5

### 3.7.7. Use Case 6

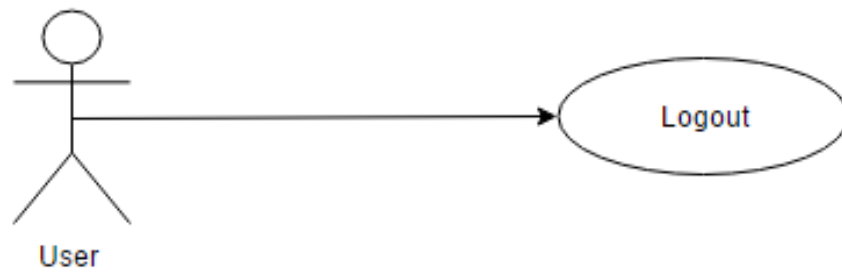


Figure 3.7.7: Use Case 6

<b>Use Case Id</b>	UC 06.
<b>Title</b>	Logout.
<b>Description</b>	User can logout from the application.
<b>Primary Actor</b>	User.
<b>Pre-Condition</b>	User must login to the application.
<b>Steps</b>	<ol style="list-style-type: none"><li>1. User opens the application.</li><li>2. There will open the menu.</li><li>3. User selects the logout option.</li></ol>
<b>Post Condition</b>	The user will be logged out of the application.

Figure 3.7.7: Use Case 6

# **Chapter 4**

# **System Design**

---

## 4. System Design

In this chapter the design and architecture of the application intended to be developed is discussed. This chapter also presents an overview of how different processes will carry out and interact with each other to achieve the result.

### 4.1. System Architecture

The architectural view of our system is defined in figure 4.1.

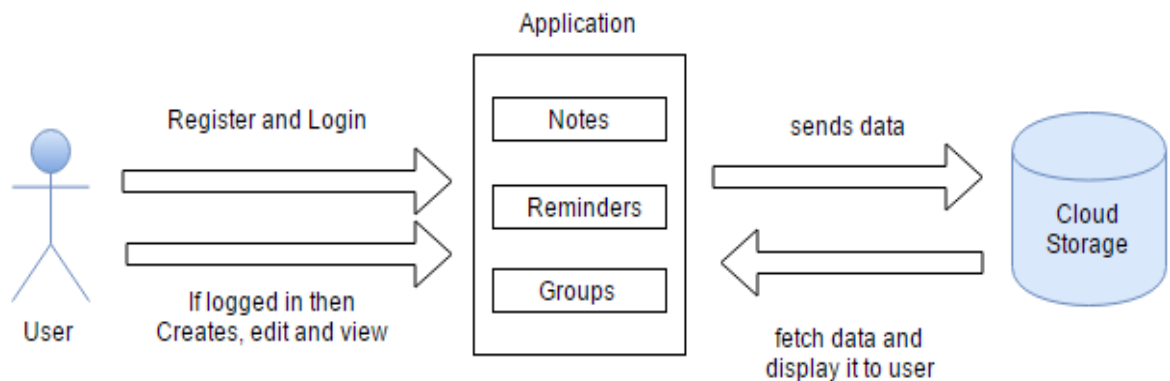


Figure 4.1: System Architecture

The user will simply interact with the application. The user will first create an account for the application then user will be able to create notes and view those notes. Those notes will be sent to the database (cloud storage).

The application will fetch data from the data base and will display all the notes to the user on the dashboard, from where user can view his created notes.

User can also create group and add different users in it. Which will allow user to share their notes.



## 4.2. Deployment Diagram

Figure 4.2 shows how the application will be deployed on hardware on device.

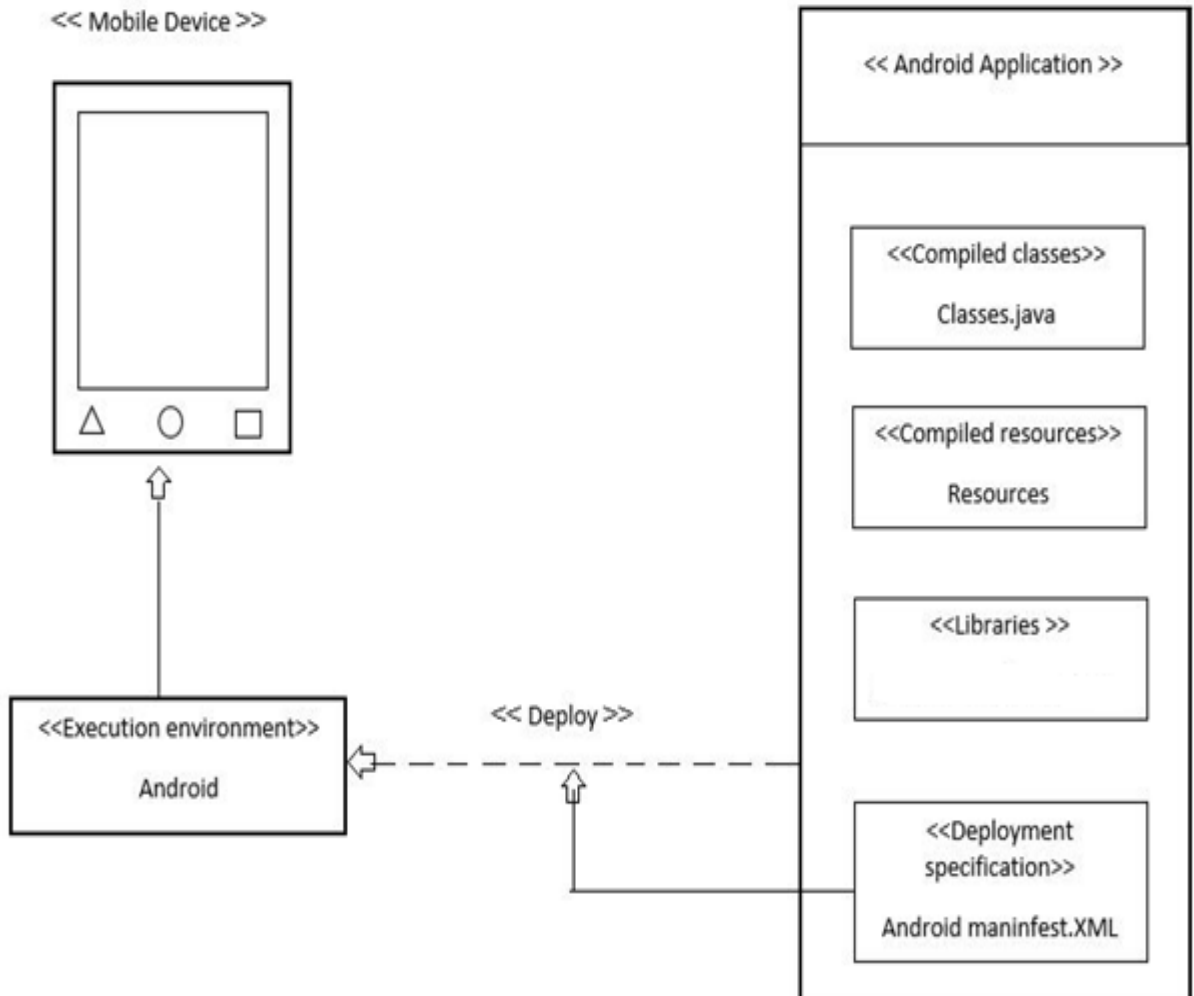


Figure 4.2: Deployment Diagram

### 4.3. Database Design

The database designed for NOTA is by using cloud services. The database is hosted by firebase, which is the service provided by google. There are currently 4 tables in the database. Attributes along with their tables are shown in figure 4.3.

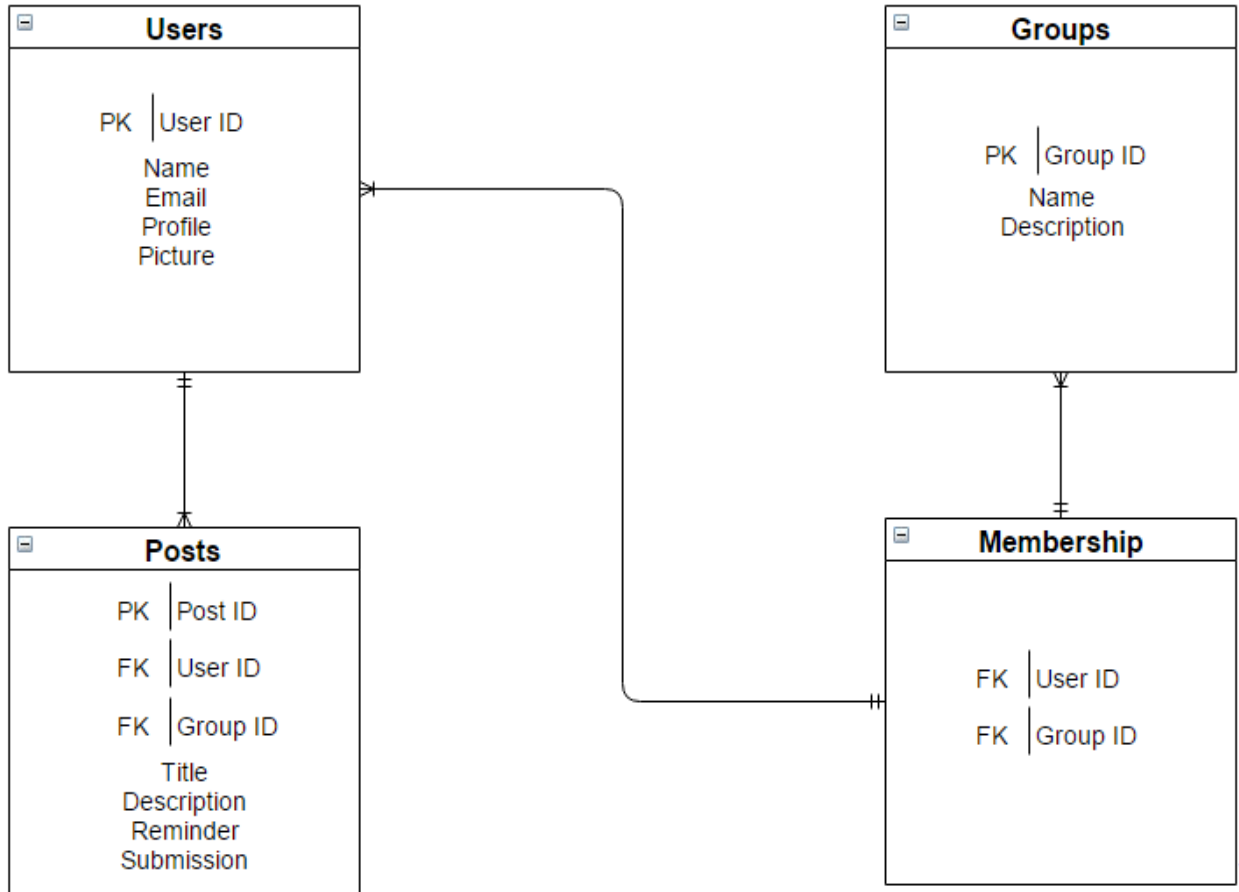


Figure 4.3: Database Design

## 4.4. System Sequence Diagram

The Figure 4.4 shows how the process interact with each other and in what order.

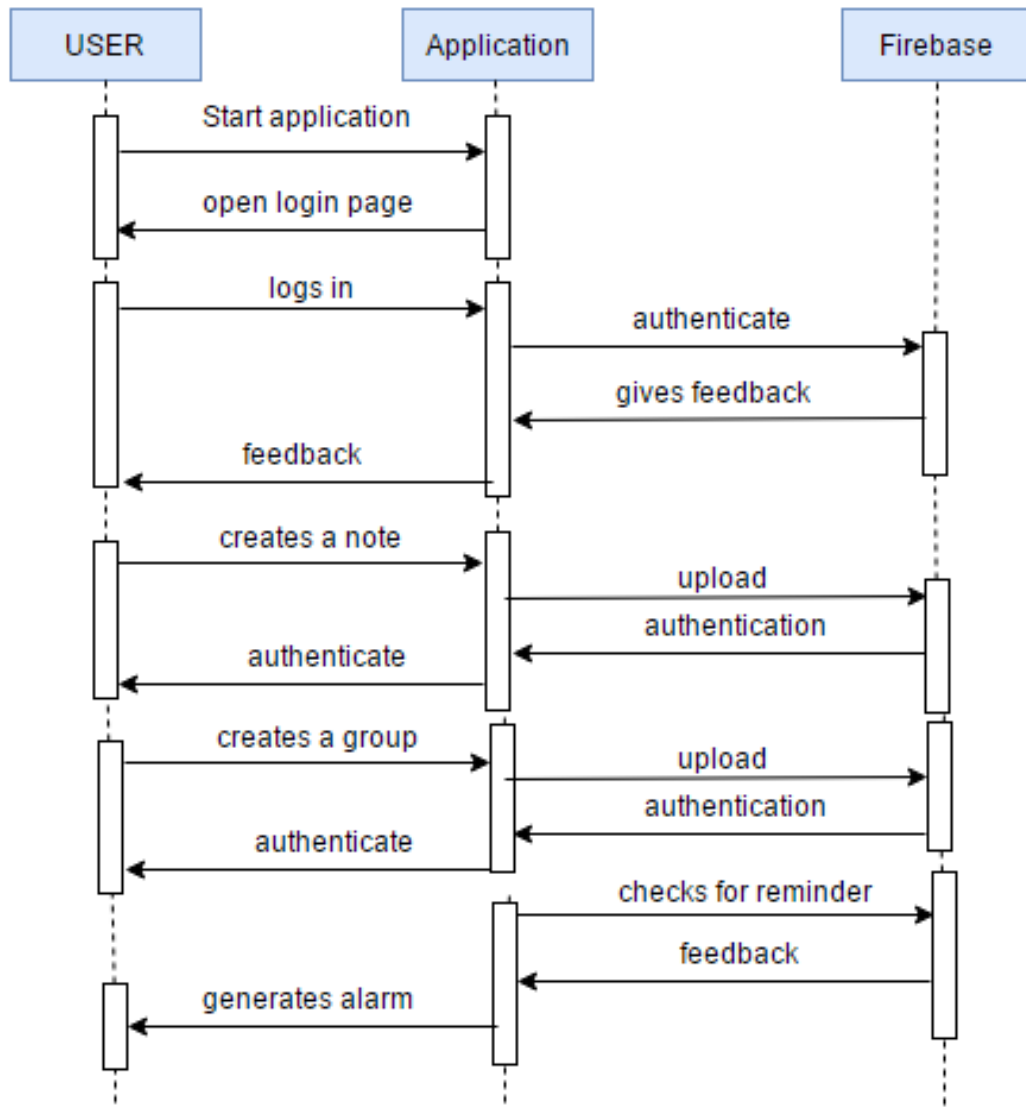


Figure 4.4: Sequence Diagram

## 4.5. Process Interaction Models

How the processes will interact with each other are defined.

### a. Process Model 1

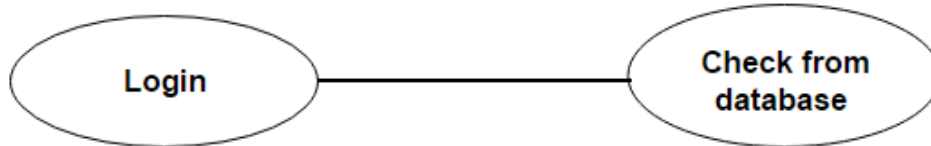


Figure 4.5 (a): Process Model 1

<b>Process Id</b>	P 1
<b>Title</b>	Login Process
<b>Description</b>	Compares the username and password from the database.
<b>Pre-Condition</b>	To open the application.
<b>Post Condition</b>	The login is confirmed and application is opened.

Table 4.5 a: Process Model 1

## b. Process Model 2

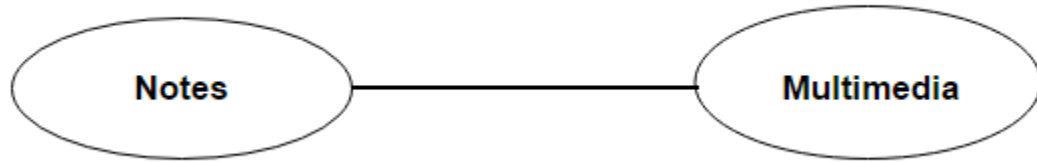


Figure 4.5 (b): Process Model 2

<b>Process Id</b>	P 2
<b>Title</b>	Adding Multimedia to Notes
<b>Description</b>	Other than plain text multimedia can also be attached to the notes.
<b>Pre-Condition</b>	The user creates or edit a note.
<b>Post Condition</b>	A note is created.

Table 4.5 b: Process Model 2

**c. Process Model 3**

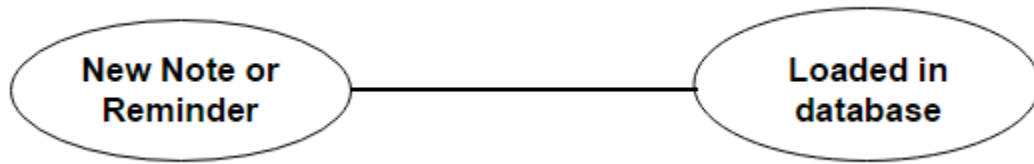


Figure 4.5 (c): Process Model 3

<b>Process Id</b>	P 3
<b>Title</b>	New Note or Reminder
<b>Description</b>	The new note that will be created is loaded in to the database.
<b>Pre-Condition</b>	The user must be logged in to the application.
<b>Post Condition</b>	The note is successfully created.

Table 4.5 c: Process Model 3

#### d. Process Model 4

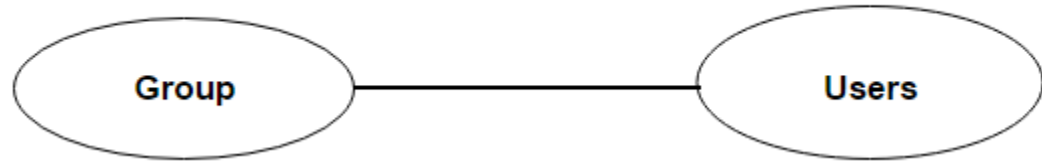


Figure 4.5 (d): Process Model 4

<b>Process Id</b>	P 4
<b>Title</b>	Groups
<b>Description</b>	A user creates a group and will be admin of that group. He can add users, using their email id.
<b>Pre-Condition</b>	The user should be logged in to the application
<b>Post Condition</b>	The group will be crated and notes can be shared in it.

Table 4.5 d: Process Model 4

# Chapter 5

# Methodology

---



## 5. Methodology

This chapter provides an overview of the tools and technologies, the libraries and other software development techniques used to develop the application.

### 5.1. Tools and Technologies

The following are the tools and technologies which have been used for the development of this application.

#### 5.1.1. Android Studio

Android Studio is the integrated development environment (IDE) for the android platform. Android studio is precisely designed for android development. The official language used for android development is Java. Some of the prodigious parts of android are written in Java and its APIs are designed in such a way that they can be called from Java [4].

#### 5.1.2. Firebase

Firebase is a cloud backend, and one of the significant choices for backend as a service. It enables you to get synchronized data up and running for multiuser applications. This is significant to use firebase, because nearly every mobile application these days requires confirmation and real time data updates [5].

#### 5.1.3. Butter Knife

Android Butter Knife is an open source view injection library for Android created by Jake Wharton. Main features this injection library can provide [6].

- Binding views and resources
- Event Listeners
- List adapters, Recycler view implementation

#### 5.1.4. Glide

Glide is an image loader library for Android developer.

## 5.2. Languages Used

Java language been used for development of this application. Java is a widely-used object oriented language, which supports a wide range of libraries. It is also the mother language to android operating system.

## 5.3. Modules Description

In this section the methodology and the flow of the application development process is explained.

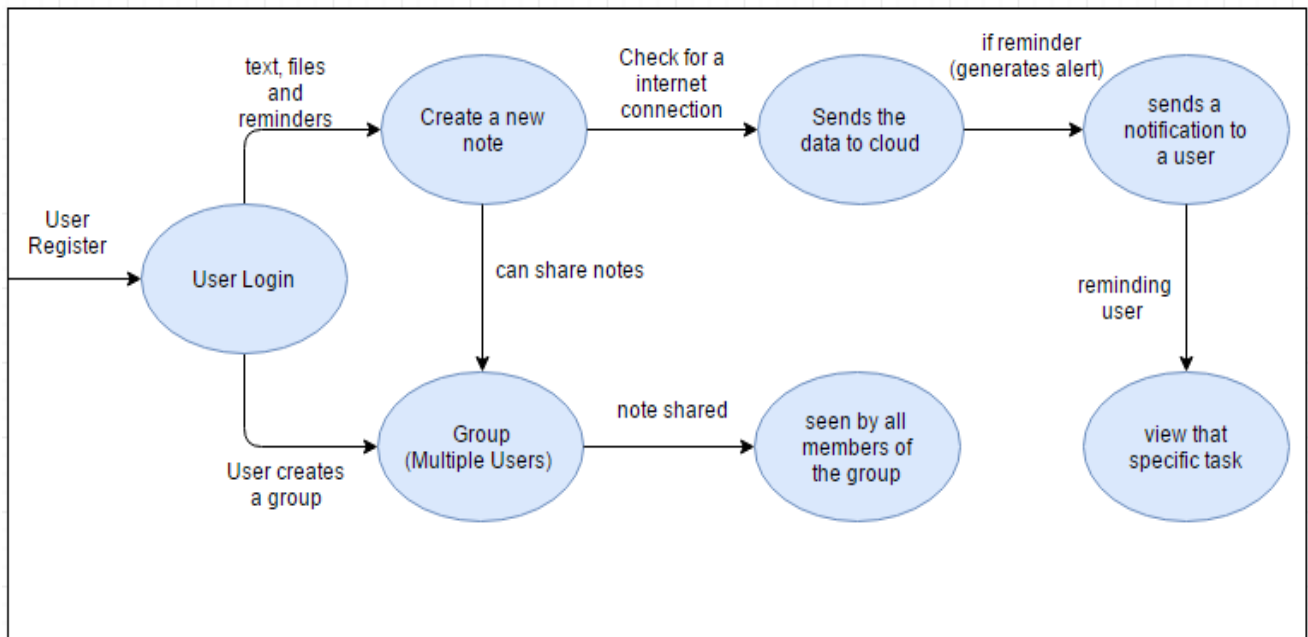


Figure 5.3 Modular diagram

### 5.3.1. Login

After the application is opened a login page appears. The user has two options after opening the application.

**a. Register**

If the user registers himself, the application will need some data from the user to register him in to the application. Like email id and password, and there is a constrain on the password that it must be larger than 6 characters.

When the user registers himself, his data is saved in to the database.

**b. Login**

The user must provide with authentic email and password to enter in to the application. When the user enters the email id and password the application compares the entered data with the user details saved in the database. If the data entered by the user matches the data in the database the user is granted access to the application, otherwise a message appears that email or password is invalid.

**5.3.2. Creating a note**

Creating a new note is one of the essential part of this application. So, the activity of creating a task or note was build. A note contains different options for a user. He can either only save a text note or add multimedia, files and reminder with that note. A note must have a description and title of the note.

**a. Reminder**

On clicking on the reminder option, it will display a user an alert from which a user will select a date and time for their reminder.

If the reminder is set, a class will be called which will provide access to system alarm services. When that date and time arrives, broadcast receiver will generate a notification of that task. The constrain in this part off the application is that a user can only select the date and time of the reminder after the date and time of the reminder that is being set.

**b. Files**

User can add one or more files in to a note. A file can be of any type. Word file, pdf file, text file and many other formats of file.

**c. Multimedia**

Multimedia can also be attached. Videos, pictures and audio files. After pressing on an attachment, the application will open the file manager of the device, from where a user can select his desirable multimedia file to be attached to the note.

**d. Save**

On clicking the save button, the application is going to check that if the description or the task title of the note is empty or not, if it is empty the application is going to generate a message for the user.

Otherwise the note or task of the user will be uploaded in to the database.

**5.3.3. Cloud Database**

Firebase is used to create a database of this application. Firebase is tightly integrated with google cloud platform. The firebase SDK directly stores data into google cloud storage buckets. The advantage of using cloud is that it can synchronize data and that data can be accessed on different devices simultaneously as shown in figure 5.4.3, if the data is uploaded from a personal computer then the data will be accessed by different devices, which are connected to that cloud database.

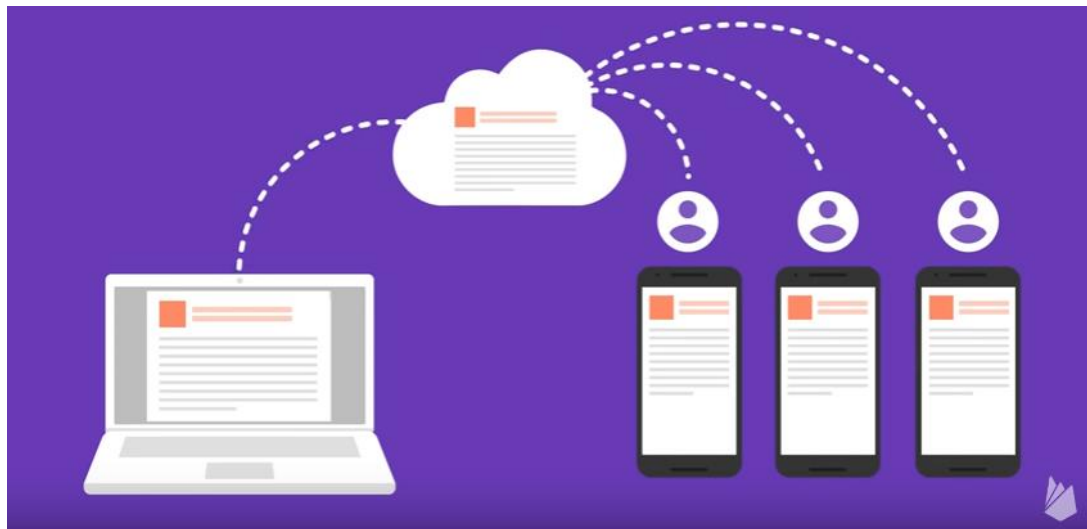


Figure 5.4.3: Cloud Database

Same is the case with NOTA, if a user uploads a data or note from one android device then that data or note will be accessed by the user if he login to the NOTA from different android device.

#### 5.3.4. Group

User can also create a group, in order to add users in the group, the group admin must know the email id of the users. The user can share data or notes in to that group which is visible to all the users which are added in to that group as shown in figure 5.4.4.

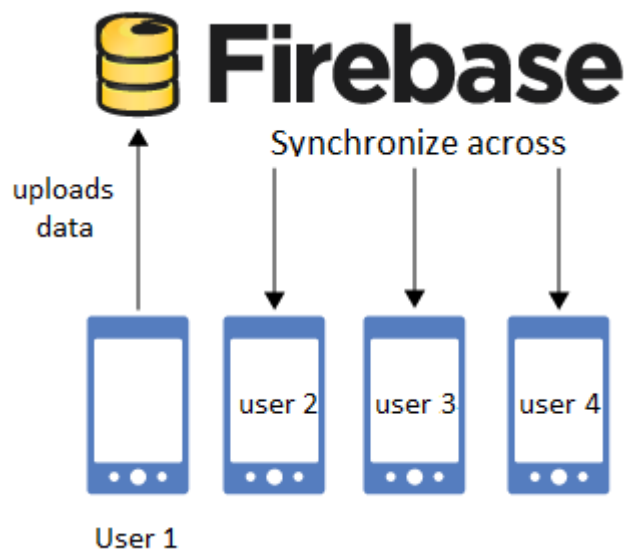


Figure 5.4.4: Group Synchronization

#### 5.3.5. Dashboard

The main dashboard of the application displays all the tasks that are made by the user. The dashboard of the application is divided in to fragments. Each fragment displays a new task or note as shown in Figure 5.4.5.



Figure 5.4.5: Dashboard

### 5.3.6. Navigation troll

Navigation troll helps the user to navigate through the application, it consists of a navigation head and the navigation body. From navigation head the user can change his display picture and his profile name, to upload the image glide library is used. The body of the navigation troll contains other options like settings and groups as shown Figure 5.4.6.

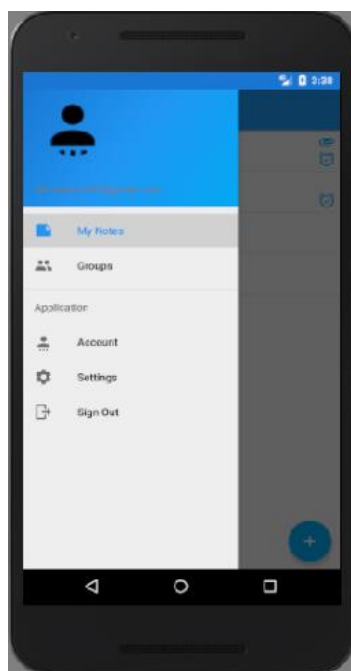


Figure 5.4.6: Navigation Troll

Chapter 6

# System Testing & Evaluation

---

## 6. System Testing and Evaluation

In this chapter, various software testing techniques which have been used to evaluate the application have been described followed by the test results, ending with a conclusion about the final product.

### 6.1. Software Testing Techniques

Software testing is a process of running a program with the intent of finding errors in the software. It is also defined as testing is the process of evaluating a system, or its components with the intent to find whether it satisfies the specified requirements or not [7]. In other words, it can also be defined as the process of checking and evaluating whether a system is doing what it has been programmed to do. The following are some of the software testing techniques which have been used to test the application.

- a) Function Testing
- b) Performance Testing
- c) Acceptance Testing

### 6.2. Function Testing

In function testing, rather than testing the complete system, the focus is to test the components and functions within the system and their interactions with each other. This can also be defined as feeding inputs to the functions to get outputs. Function testing has the following types.

#### a. Unit Testing

Test the programs and files individually to check if they are working properly. All the resources, the models and the android activity files have been tested and have appeared to work fine.

#### b. Integration Testing

Testing the programs, files and the process of the system to verify if they have been integrated with each other properly.



### **c. System Testing**

Testing the behavior of the complete system. A good way to do system testing to verify the system against the nonfunctional requirements. The system appears to fulfill all the required nonfunctional requirements.

### **d. White Box Testing**

Testing the system with the knowledge of system's internal structure and working methodologies. The system produced fair results.

### **e. Black Box Testing**

Testing the system without the knowledge of system's internal structure and working methodologies. For black box testing the application was deployed on a phone and given to another user for testing. After testing the system by using it the user seemed to be quite satisfied by the system's performance.

## **6.3. Performance Testing**

Software performance testing can be defined as the process of testing the performance of a system. This process involves checking how reliable, effective and accurate the system is. It can determine whether the system is achieving the performance criteria. In general terms, we check all the aspects regarding the performance of the system. It has the following types.

### **a. Configuration Testing**

Testing the system against various software and hardware configuration with the exact and similar system requirements. The application was tested for configuration testing on android versions 5.1 (lollipop), 6.1 (marshmallow). Acceptable results were produced.

## **b. Security Testing**

Testing the security requirements of the system. The system requires the security as there are different user of this application and each user has his own private data, which must be secured.

## **6.4. Acceptance Testing**

Acceptance testing can be termed as testing the system to verify if it is ready for implementation or use. Acceptance testing reveals errors and omissions in the system requirements. There are three type of acceptance testing but for this project only one has been checked which is the following.

### **a. Installation Testing**

Testing that whether the system can be installed on user devices or not. The focus of such testing is on two things which are as follows.

- a) To verify the installation of the complete system on device
- b) To verify the functional and nonfunctional requirements of the system after installation on device.

The application was tested for installation testing on android versions 5.1 (lollipop) and 6.1(marshmallow). Acceptable results were produced.

## 6.5. Test Cases

### 6.5.1. Test Case 1. Installation of Application

**Special Instructions:** The application is to be installed on a device with android operating system. Ideally with android version greater then android version (4.0.3) (ice-cream sandwich).

Test Case Id	01
Description	The application is to be installed on an android device.
Application for	Android based mobile phones
Requirements	The device should have and android operating system greater then android version (4.0.3) (ice cream sandwich).
Steps to be taken	Run the application apk form the respective location on device.
Expected Result	The complete application should be installed on the device.
Actual Result	The complete application is installed on the device.
Status	Success
Remarks	N/A

Table 6.6.1: Test Case 1

### 6.5.2. Test Case 2. Register and Login the User

**Special Instructions:** The user should be register after providing the email and password. The user must be able to login providing the valid email and password.

Test Case Id	02
Use Case Id	01
Description	The user must register and would be able to login after providing valid email and password.
Application for	Android based mobile phones
Requirements	<ul style="list-style-type: none"><li>• User must register.</li><li>• The user must be logged in after authentication.</li></ul>
Steps to be taken	<ul style="list-style-type: none"><li>• Save the user data in the database when registering.</li><li>• Authenticate the user when logging in to the application.</li></ul>
Expected Result	The user should successfully registered and should be logged in the application.
Actual Result	The user is successfully registered and have logged in the application.
Status	Passed
Remarks	There will be a message displayed when the user fails to login the system.

Table 6.6.2: Test Case 2

### 6.5.3. Test Case 3. Creating a New Note

Test Case Id	03
Use Case Id	02
Description	After pressing the new note icon. The application would take the user to the next page where he can create a new note.
Application for	Android based mobile phones.
Requirements	The icon of the new note should be pressed.
Steps to be taken	After logging in to the application, press the new note icon.
Expected Result	New page should be displayed where the user can create new notes.
Actual Result	A new page is displayed after pressing the new note icon.
Status	Passed
Remarks	N/A

Table 6.6.3: Test Case 3

#### 6.5.4. Test Case 4. Options in a Note

Test Case Id	04
Use Case Id	02
Description	Checking whether a note have all the options like attachments and reminder.
Application for	Android based mobile phones.
Requirements	A note contains an image, video, text and user can also set a reminder.
Steps to be taken	Set on the reminder option and press on the attachment icon to attach something.
Expected Result	A note should be able to attach files and images and user can also set a reminder on that note.
Actual Result	A note is able to attach files and images and user can also set a reminder on that note.
Status	Passed
Remarks	N/A

Table 6.6.4: Test Case 4

### 6.5.5. Test Case 5. Creating a Group

**Special Instructions:** Email id of the users must be known to add them in a group, then the users will be added in the group.

Test Case Id	05
Use Case Id	03
Description	Testing if the group is successfully created or not.
Application for	Android based mobile phones
Requirements	If the user is valid, he must be added in a group. They should receive what is send by another user.
Steps to be taken	Click on the group icon in the menu. Select the create menu option in that.
Expected Result	The group should be created properly.
Actual Result	The group is being created properly.
Status	Passed
Remarks	N/A

Table 6.6.5: Test Case 5

### 6.5.6. Test Case 6. Share the notes

Test Case Id	06
Use Case Id	04
Description	Testing if a note can be shared through a group to the users in that group.
Application for	Android based mobile phones
Requirements	If the user is added in a group, he must receive the shared note.  The note should be created and must be selected before sending it.
Steps to be taken	Click on the group icon in the menu.  Select the note that is to be shared.
Expected Result	The note is shared, and is received by the users in a group.
Actual Result	The note shared is received by the users to whom it was shared.
Status	Passed
Remarks	N/A

Table 6.6.6: Test Case 6



### 6.5.7. Test Case 7. Logout from the application

Test Case Id	07
Use Case Id	06
Description	Logout from the application.
Application for	Android based mobile phones
Requirements	The user logouts from the application.
Steps to be taken	Open the menu from the application. Select the logout option.
Expected Result	User logout from the application and return to login page.
Actual Result	User logouts from the application.
Status	Passed
Remarks	N/A

Table 6.6.7: Test Case 7

# Chapter 7

# Conclusion

---

## 7. Conclusion

The developed application is an android based application (NOTA) which will help users to store notes. This application not only stores simple text notes but have many other options, user can attach multimedia (pictures, video, audio) with his notes, other than that user can also attach text files (word, power point) with it. User can set a reminder with his note or even can set a reminder to do a specific task, user can even edit and delete their notes. When the reminder goes off, it will generate a notification which will remind user to do that specific task. User can also share his notes with other users by creating a group. The interface of this application is simple and user friendly, when the user login there is a dashboard which will be displaying all the notes or tasks created by that user. User can also change their username and profile picture from the menu. User just must register themselves to use this application. Cloud services are used to store the data of the users. Firebase is used for NOTA, firebase is the service provided by Google. The user can access their data from anywhere, they just have to login to the application. NOTA helps users to organize and remember their tasks. The users take notes in many ways and other then that they can share them with other users. NOTA even provide users with privacy, the data of the users is secure and is not shared with any other user, unless the user themselves does not share data. Users can also view their notes in an offline mode. NOTA provides user with much more then storing simple text notes.

# References

---

## References

[1] Color Notes. 2008.

<https://www.colornote.com/feature-overview/>

[2] Omni Notes. January 28, 2014.

<https://www.xda-developers.com/omni-notes-the-open-source-note-app/>

[3] Notes of an iPhone. April 10, 2017. Apple.

<https://support.apple.com/en-us/HT205773>

[4] Matthew David. About Android Studio. September 2015.

<http://searchsoftwarequality.techtarget.com/feature/Learn-more-about-the-Android-Studio-IDE-from-Google>

[5] Firebase Realtime Database. 17 April 2017. Google.

<https://firebase.google.com/docs/database/>

[6] Valdio Veliu. Android Butter Knife. July 2016.

<https://www.sitepoint.com/tidying-code-with-android-butterknife/>

[7] Tutorials Point. "Software Testing Tutorial." *Tutorials point*. N.p., n.d. Web.

[http://www.tutorialspoint.com/software\\_testing/](http://www.tutorialspoint.com/software_testing/).

[8] Roy Sandberg and Mark Rollins. Apps development. 2013. Cited on p. 1.