ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ
VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A MINI PROJECT REPORT ON

"COLLEGE CONNECT"

Submitted to Visvesvaraya Technological University in partial fulfillment of the requirement for the award of degree of

Bachelor of Engineering

in

Computer Science and Engineering.

Submitted by:

Mr. Navaneeth N Kamath	4JN18CS054
Mr. Nagaraj Tirumaleshwar Hegde	4JN18CS049
Mr. Madhusudhana B. M	4JN18CS043

Under the Guidance of:

Dr. Chetan K. R Ph.D.
Associate Prof., CSE

Department of Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga - 577 204

August 2021

National Education Society ®

Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Mini-project Report entitled

COLLEGE CONNECT

Submitted by:

Mr. Navaneeth N Kamath	4JN18CS054
Mr. Nagaraj Tirumaleshwar Hegde	4JN18CS049
Mr. Madhusudhana B. M	4JN18CS043

Students of 6th semester B.E under the supervision and guidance towards the partial fulfillment of the requirement for award of degree of the Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2020 – 2021.

Signature of Guide and HOD:

Dr. Chetan K R Ph.D.,	**Dr. Poornima K M** Ph.D.,
Associate Prof., CSE	Professor & Head, CSE

Name of Examiners	Signature with Date
1.	
2.	

ABSTRACT

College Connect is a project which aims in developing a mobile application to ease the communication between students and faculties of a particular educational institution. This project has all the basic functionalities which are generally needed for online communication like sending messages, images, and documents. The use of this application to keep conversations secured and confidential within the faculties and students themselves. It reduces the dependency on other internet-based messengers and the institute can have its own messaging application. The main aim is to provide a separate platform for students and faculty so that they can communicate with each other either in one-to-one messaging or group messaging to share the information and other things related to academics only.

ACKNOWLEDGEMENT

On presenting the project report on "**College Connect**", we feel great to express our humble feelings of thanks to all those who have helped us directly or indirectly in the successful completion of the project work.

We would like to thank our respected guides **Dr. Chetan K R, Associate Professor, Dept. of CS & E,** and **Dr. Jalesh Kumar, Professor, Dept. of CS & E,** who helped us a lot in completing this project, for their continuous encouragement and guidance throughout the project work.

We would like to thank **Dr. Poornima K M,** Professor and Head, Dept. of CSE, JNNCE, Shivamogga and **Dr. Manjunatha P**, Principal, JNNCE, Shivamogga for all their support and encouragement.

We are grateful to **Department of Computer Science and Engineering** and our institution **Jawaharlal Nehru New College of Engineering** and for imparting us the knowledge with which We can do our best.

Finally, we also would like to thank the whole teaching and non-teaching staff of Computer Science and Engineering Dept.

Thanking you all,

NAVANEETH N KAMATH	**4JN18CS054**
NAGARAJ TIRUMALESHWAR HEGDE	**4JN18CS049**
MADHUSUDHANA B M	**4JN18CS043**

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1. Overview

College Connect is an android software designed to assist in the communication between college faculties and students of a particular educational institution, and the need of such application is to maintain security and privacy of chats and other information which is shared with the student by the college.

This College Connect application is built using Kotlin as a programming language with Cloud Firestore as the database management system which is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud.

1.2. Applications

- This application can be used in almost all the educational institutions where there is a need for separate communication the medium between faculty and the student.
- This application can be used to communicate in between the staff to maintain the confidentiality of the information.
- This application can be used for group communication between teaching faculties and the student or in between staff members themselves.
- This application can be used to share images and documents along with text messaging.
- This application can also become the medium for collecting personal and academic data from new students joined to the institution.

1.3. Problem Statement

For the need of a communication medium either in between institutional faculties or in-between faculty and students to share information, images, and documents generally depends on different internet-based messengers. In those cases, there will be a high risk of confidentiality and security of data. And different messengers will have different functionality. If faculties and students need some different functionality or features which is not there in their current using messenger then they all need to switch to other applications which are time-consuming, waste of other resources, and generally, there will be a high risk of loss of data.

1.4. Objectives

- This College Connect application is designed to ease the means of communicating, sending images, documents and circulars to the faculties and students of a particular institute.
- The main feature of this application is an easy one-to-one communication between faculties or between faculty and student in a secured and confidential manner.
- This application can be adopted in any educational institution where there is a need of communication.
- This application can be used to get academic and personal information from students who joined the institution.
- This application can also be used to make group communication in between faculties, or in between faculties and students.

1.5 Overview of Android

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.

Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application. Pictorial representation of android architecture with several main components and their sub components

1.5.1 Applications:

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third-party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only. It runs within the Android run time with the help of the classes and services provided by the application framework.

1.5.2 Application framework:

Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the

Applications creation. It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

1.5.3 Application runtime:

Android Runtime environment is one of the most important part of Android. It contains components like core libraries and the Dalvik virtual machine (DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries. Like Java Virtual Machine (JVM), Dalvik Virtual Machine (DVM) is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

1.5.4 Platform libraries:

The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

- Media library provides support to play and record an audio and video formats.
- Surface manager responsible for managing access to the display subsystem.
- SGL and OpenGL both cross-language, cross-platform application program interface (API) is used for 2D and 3D computer graphics.
- SQLite provides database support and Free Type provides font support.
- Web-Kit This open-source web browser engine provides all the functionality to display web content and to simplify page loading.
- SSL (Secure Sockets Layer) is security technology to establish an encrypted link between a web server and a web browser.

1.5.5 Linux Kernel:

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime. The Linux Kernel will provide an abstraction layer

between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.

The features of Linux kernel are:

- **Security:** The Linux kernel handles the security between the application and the system.
- **Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.
- **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
- **Network Stack:** It effectively handles the network communication.
- **Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build.

1.6. Overview of Kotlin

Kotlin is an open-source, statically-typed programming language that supports both object-oriented and functional programming. Kotlin provides similar syntax and concepts from other languages, including C#, Java, and Scala, among many others. Kotlin does not aim to be unique—instead, it draws inspiration from decades of language development. It exists in variants that target the JVM (Kotlin/JVM), JavaScript (Kotlin/JS), and native code (Kotlin/Native).

1.6.1 Kotlin language features

1. Efficiency: The Kotlin programming language is not only easy to learn, it is also effective in the way that it codes your application. Developers won't need to spend a lot of time grasping this language. The syntax is intuitive and lean. One of the most attractive Kotlin features for developers is that it reduces code length. Compared to Java, you can say the same things with less lines. In fact, the coding is approximately 20% less than Java. With this reduced length, you have better chances at improving quality and operations. Kotlin basically makes coding efficient and testing near perfect.

2. Extension Functions: As the name "extension function" suggests, this allows you to add extra features to your existing component, which makes it easy for you to add simple yet impressive features. This is perfect for calculations as well as for appending strings. In simple terms, extension functions allow you to extend functionality without changing the major part

of the code. The question is, where can you use it? Well, It works perfect in calculations as well as in appending strings.

Here's how you can define extension functions in Kotlin.

Suppose you need to cut down the first and the last character of the string. In Java, you need to write the same boilerplate code again and again, but with the use of Kotlin, you can create an extension of the String class.

Here is an example:

```
fun  String.removeFirstLastChar(): String = this.substring(1, this.length – 1)
```

This snippet will cut down the first and last character of String.

How to use it:

```
val  myString = "Hello World"

val result = myString.removeFirstLastChar()
```

We can use this function to extend any class and create an extension for it.In short, with extension function, you can extend any class. Yes, any class. Even if you don't have access to the code.

3. Massive Interoperability: The two languages co-exist, which makes developers' lives easier and more productive. You can easily compile your single project in both languages with the help of the interoperability function, which allows you to switch the programming language without switching the codes.Kotlin is a language that is 100% interoperable. If you want to access a Kotlin method from a Java class or vice versa, you can do it without any extra parameters.

Here is a simple example: Suppose you have a BaseActivity created in Kotlin, which contains the function printLog that simply prints the message.

```
fun printLog(msg : String){ //Kotlin Function

print(msg)

}
```

What if your MainActivity is in Java? If you want to print the message, then just write the message and pass the argument (String message).

```
private void main(){ //Accessing Kotlin Function from Java

printLog(msg:“Print this message.”);

}
```

This will automatically print the message in the logs.

4. Reducing Crashes at Runtime:

Nullable	var name: String? = null
Non-null	var name: String = “abc”

In Kotlin, you can easily identify the Null Pointer Exception while compiling your project. This ensures every variable works as a non-null, thus reducing the number of crashes at run-time. If you want to hold onto a null value, just append a question mark as the end of the variable type, as shown above.

5. Smart Cast Function

```
val  n:Any = “Hello”

when (n) {

        is String -> n.length

        is Int -> n.inc()

}
```

This is the ultimate function from a developer’s perspective. It effectively reduces the application’s speed, thus improving its performance. The smart cast function is known to identify the type of function. It will perform all operations that have been coded for a particular type, thus improving code efficiency. Let’s say you have identified a string. With the smart cast function, you can copy, count the length, and perform other functions on the string type.

6. Safe and Reliable: As mentioned earlier, one of the strongest Kotlin features is that it is relatively safe. Avoiding errors such as “Null Point Exception” is easy with Null Safe function.

This function even allows you to avoid use for nullable types. You can automatically remove mistakes and simplify your code for debugging and operations.

7. Low Cost of Adoption: A major reason business favour Kotlin over other programming languages is the no or low cost of adoption associated with it. It is open source, so you don't need to invest money on license. Secondly, relative to other programming languages, Kotlin is pretty easy to learn for developers.

1.6.2 Advantages of Kotlin over Java and its role in App development

Here are 5 reasons why developers like Kotlin over Java and think that it will overtake the later:

1. Shorter program for the same task: Kotlin is a statically-typed language which is very easy to read and write. It has a much simpler and shorter code than Java's code for the same problem. As this makes the language more human-readable, it becomes easy to debug. Kotlin's code is much smaller and streamlines the programming process, in comparison to Java. This is partly because of Kotlin's slick IDE.

2. Easy Code: Kotlin programs do not need semicolons in their program. This makes the programs easy to read and understand. They also have smart casts and string templates. Java is not a succinct language. Such a language code increases the chances of bugs. The code being in a concise language means fewer chances of both runtime and compile time errors. Kotlin gives a simple way to use mutable and immutable declarations for different data structures.

3. Java Compatibility: Kotlin can easily exchange and use information from Java in a number of ways. This is one of the most powerful advantages of Kotlin. Java and Kotlin code can co-exist in the same project. Kotlin plays well with the Java programming language. Moreover, a number of Java libraries can be used in Kotlin projects, making it even more compatible. Not just the libraries but plenty of frameworks from Java are compatible with Kotlin, including some advanced frameworks.

4. Eliminating Null References: One of the biggest advantages of Kotlin over Java is the null references. This null reference, referred by Sir Tony Hoare, a British computer scientist, as The Billion Dollar Mistake. Accessing a member of a null reference result in a null reference exception. This is one of the major drawbacks of Java in which it is called a

NullPointerException or NPE. Kotlin's type system is aimed to eliminate NullPointerException from the code.

5. Solution to Some of Java's Flaws: Plain old Java has a number of flaws. It inhibits the very famous problem of null pointer. Kotlin attempted to solve these hurdles created by Java. It has adopted things from a number of languages like C# and mainly from Scala, to overcome difficulties of Java. It contains instances from the language of Pascal and is considered as very influential in the development of Kotlin. Elements like parameter lists and variable declarations with the data type following a variable could also be in Kotlin.

Chapter 2

REQUIREMENTS AND DESIGN

This College Connect application includes various functional and non-functional requirements.

2.1 Functional Requirements

Functional Requirements includes various functionalities which are intended for developing this application like,

- This application should allow students to send a request for registration by giving all the information which are there in the registration form.
- After approval of the student request, the application should allow the student to login into the application.
- After logging in students should be allowed to change their password, change their profile picture, see their profile, send a message to their respective department faculty if they are intended to.
- Similarly, Admin should be able to review student requests, add new faculty, add a new admin, create a group, etc.
- Faculty members should be able to log in if their record exists in the database and they should be allowed to send messages to a student or they should be allowed to send messages to other faculties of the institution.

2.2 Non-Functional Requirements

Non-Functional Requirements include the functionalities which are needed for scalability, security, usability, and other factors. Some of the non-functional requirements are,

- The application will have to handle invalid login credentials.
- This application can block irregular login activities if anybody tries to.
- This application can take backup of data regularly.
- This application can keep a track of already existing connections, and if a new connection is requested during the existing connection then the application may warn the user with multiple connections.

2.3 Design

This section includes the wireframes/blueprints of the application like login page, change password page of student, faculty, and Admin, etc.

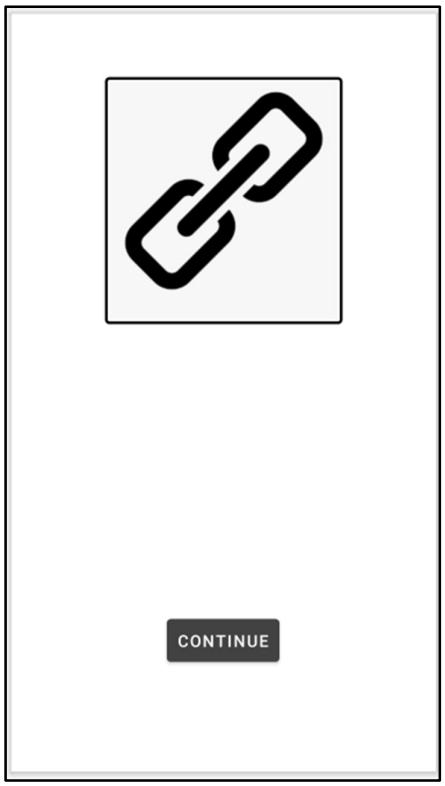

Fig: 2.1 Splash Screen of the Application

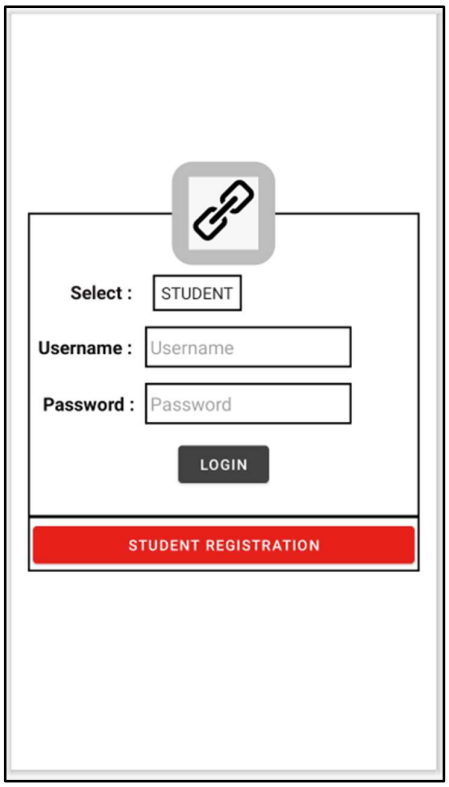

Fig: 2.2 Login Page

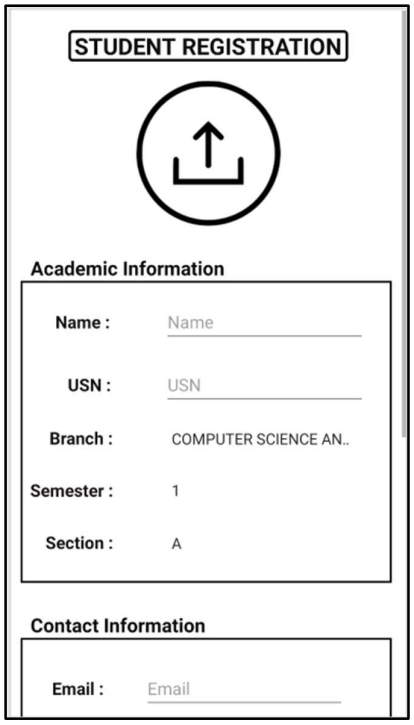

Fig: 2.3 Student Registration Page

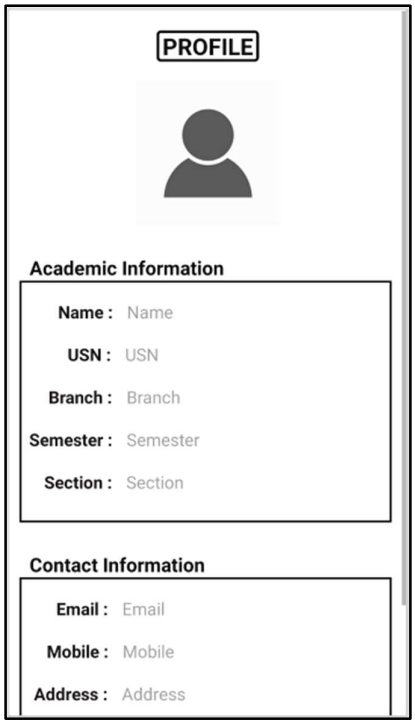

Fig: 2.4 Student Profile Page

Fig: 2.5 Student Change Password Screen

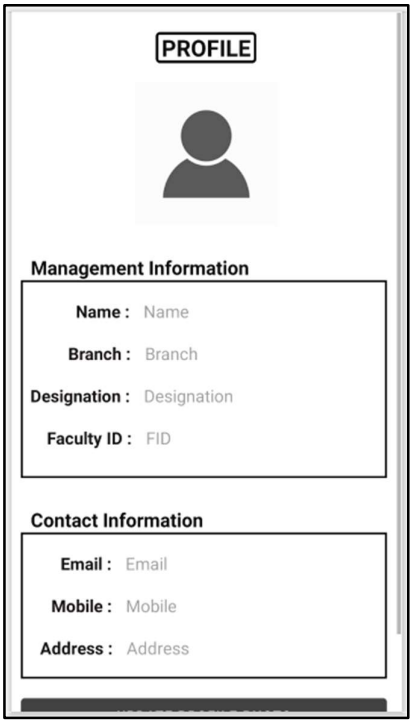

Fig: 2.6 Faculty Profile Page

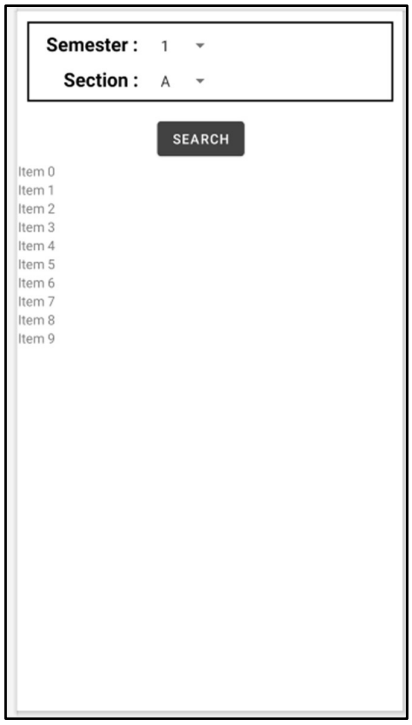

Fig: 2.7 Faculty Find Student Page

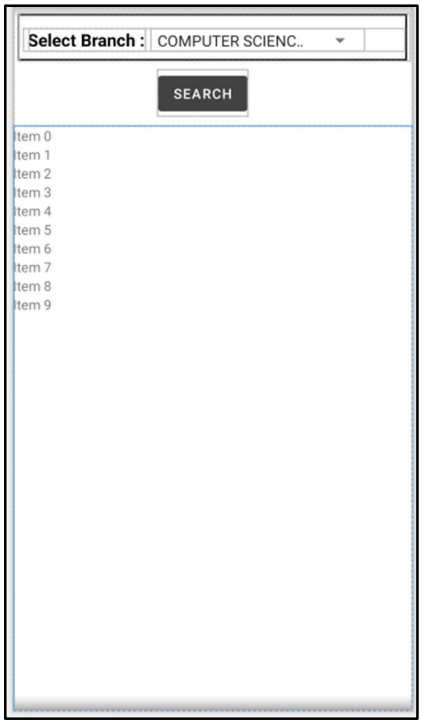

Fig: 2.8 Faculty Find Another Faculty Page

Fig: 2.9 Faculty Change Password Page

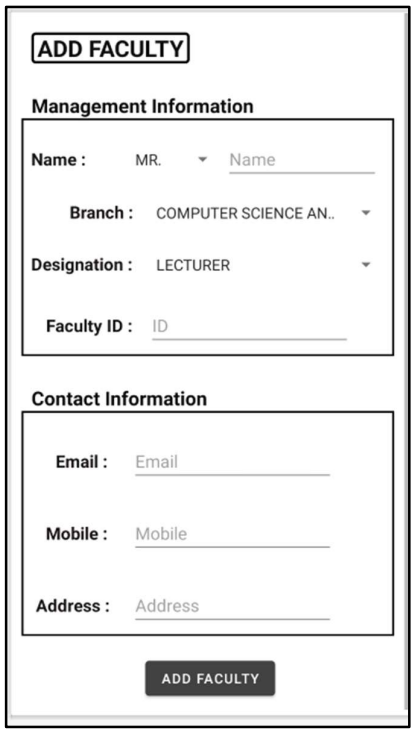

Fig: 2.10 Admin Add Faculty Page

Fig: 2.11 Admin Add Another Admin Page

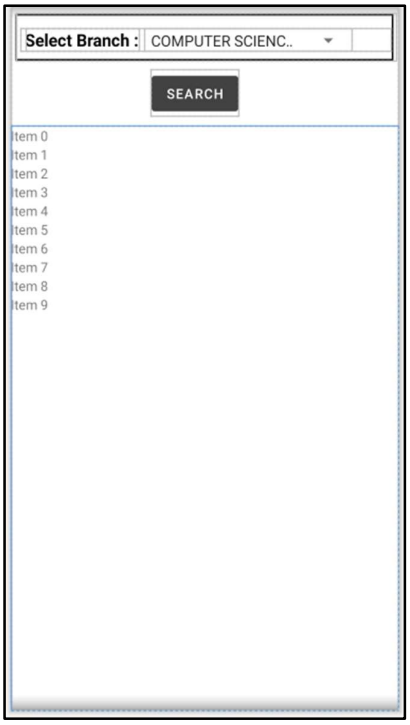

Fig: 2.12 Admin Find Faculty Page

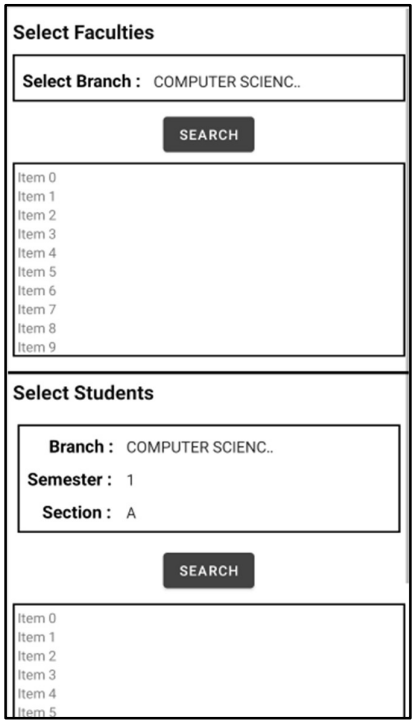

Fig: 2.13 Admin Create Group Page

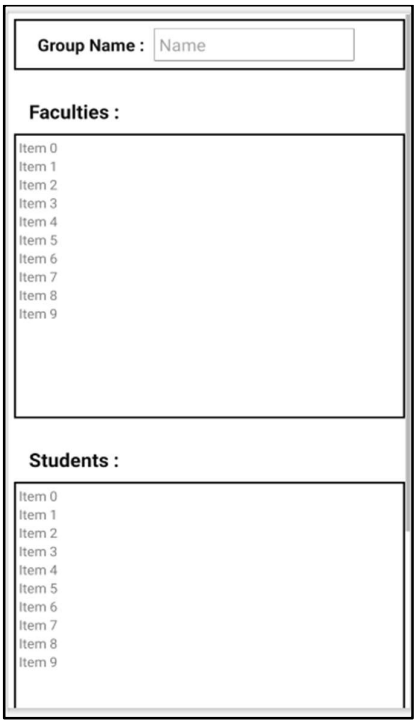

Fig: 2.14 Admin Finalize Group Page

2.4 Firestore:

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through Realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

2.4.1 Overview of Firebase technologies

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

Firebase offers a number of services, including:

- **Analytics** – Google Analytics for Firebase offers free, unlimited reporting on as many as 500 separates events. Analytics presents data about user behaviour in iOS and Android apps, enabling better decision-making about improving performance and app marketing.
- **Authentication** – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- **Cloud messaging** – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android and the web at no cost.
- **Realtime database** – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.
- **Crashlytics** – Firebase Crashlytics is a real-time crash reporter that helps developers track, prioritize and fix stability issues that reduce the quality of their apps. With crashlytics, developers spend less time organizing and troubleshooting crashes and more time building features for their apps.

- **Performance** – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.
- **Test lab** – Firebase Test Lab is a cloud-based app-testing infrastructure. With one operation, developers can test their iOS or Android apps across a variety of devices and device configurations. They can see the results, including videos, screenshots and logs, in the Firebase console.

2.4.2 Firestore structure:

The structure of data in Cloud Firestore, we have a few different options:

- Documents
- Multiple collections
- Subcollections within documents

Consider the advantages of each option as they relate to your use case. A few example structures for hierarchical data are outlined in this guide.

Nested data in the document

You can nest complex objects like arrays or maps within documents.

- **Advantages:** If you have simple, fixed lists of data that you want to keep within your documents, this is easy to set up and streamlines your data structure.
- **Limitations:** This isn't as scalable as other options, especially if your data expands over time. With larger or growing lists, the document also grows, which can lead to slower document retrieval times.

SUBCOLLECTIONS:

You can create collections within documents when you have data that might expand over time.

- **Advantages:** As your lists grow, the size of the parent document doesn't change. You also get full query capabilities on subcollections, and you can issue collection group queries across subcollections.

- **Limitations:** You can't easily delete subcollections.

ROOT-LEVEL COLLECTIONS:

Create collections at the root level of your database to organize disparate data sets.

- **Advantages:** Root-level collections are good for many-to-many relationships and provide powerful querying within each collection.
- **Limitations:** Getting data that is naturally hierarchical might become increasingly complex as your database grows.

2.5 Implementation

In this section, the working process or how the activity works and gives results is explained using flowcharts. Flowcharts of different activities of the College Connect application are attached.

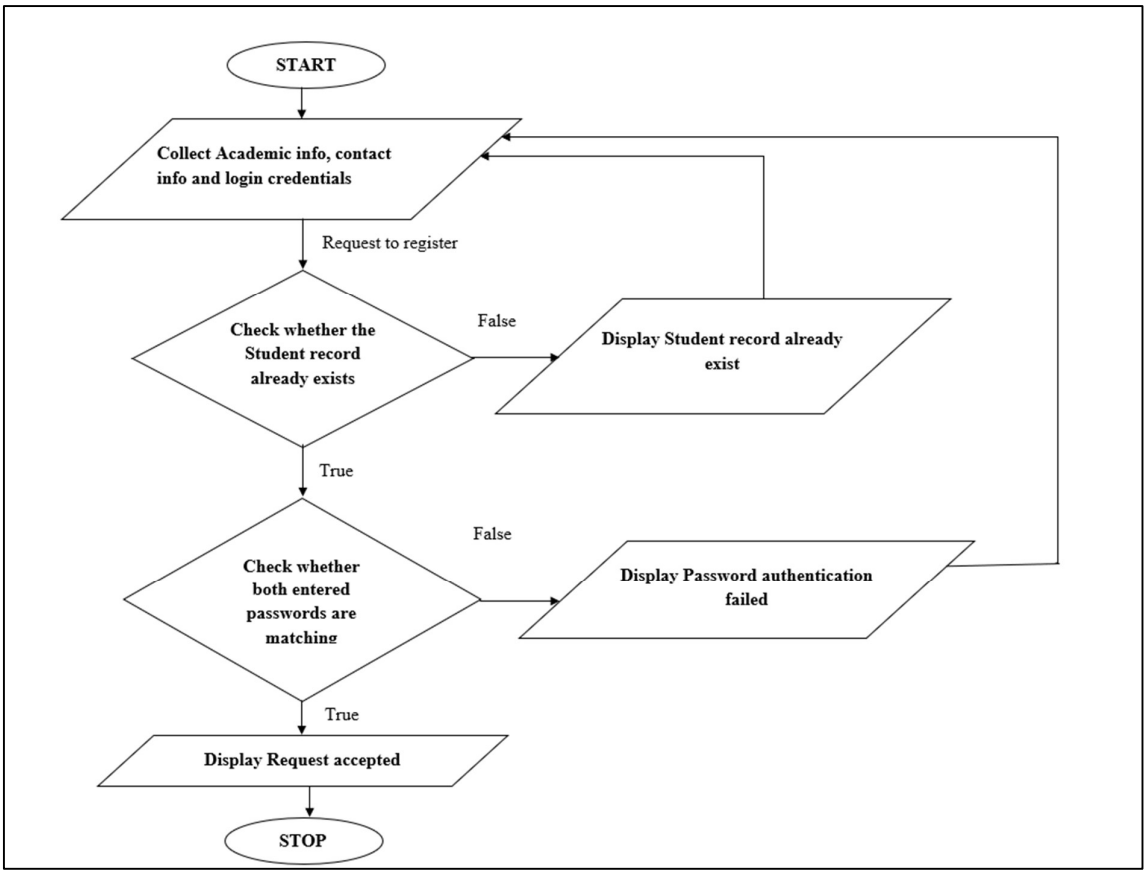

Fig: 2.15 Student Request Activity Flowchart

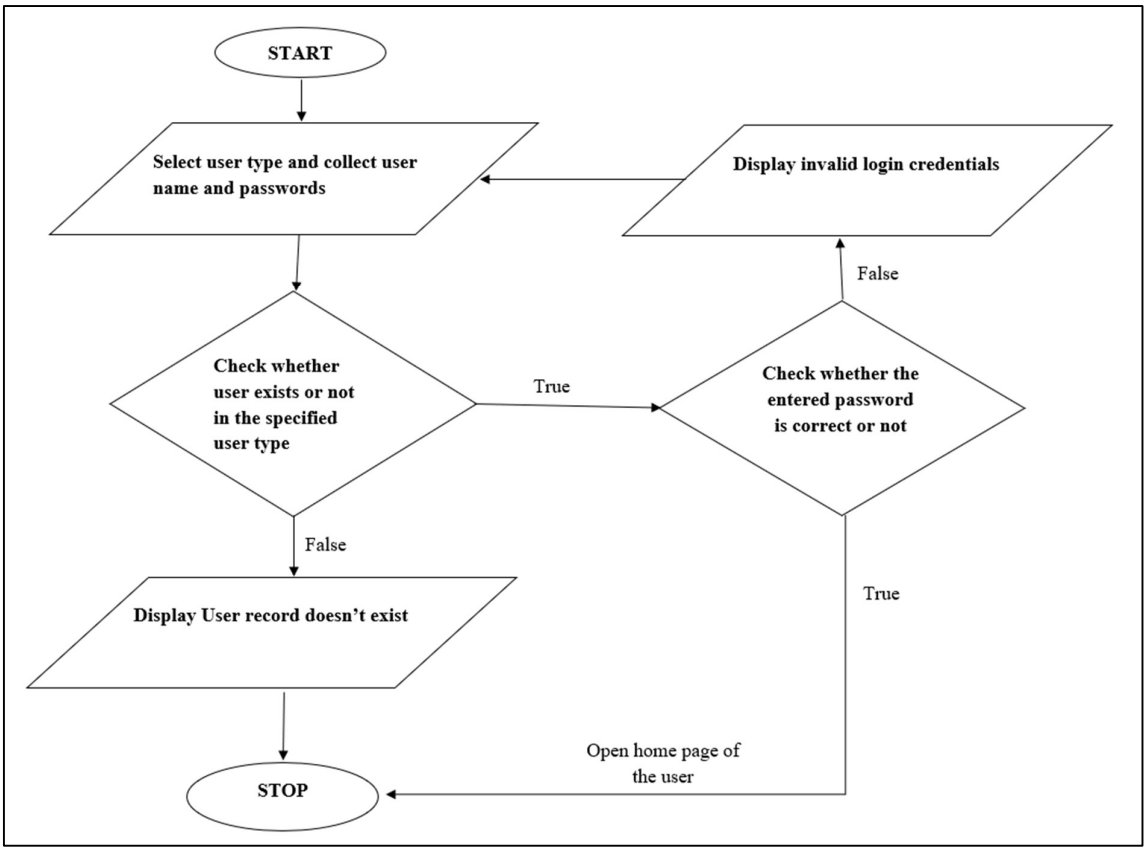

Fig: 2.16 Login Activity Flowchart

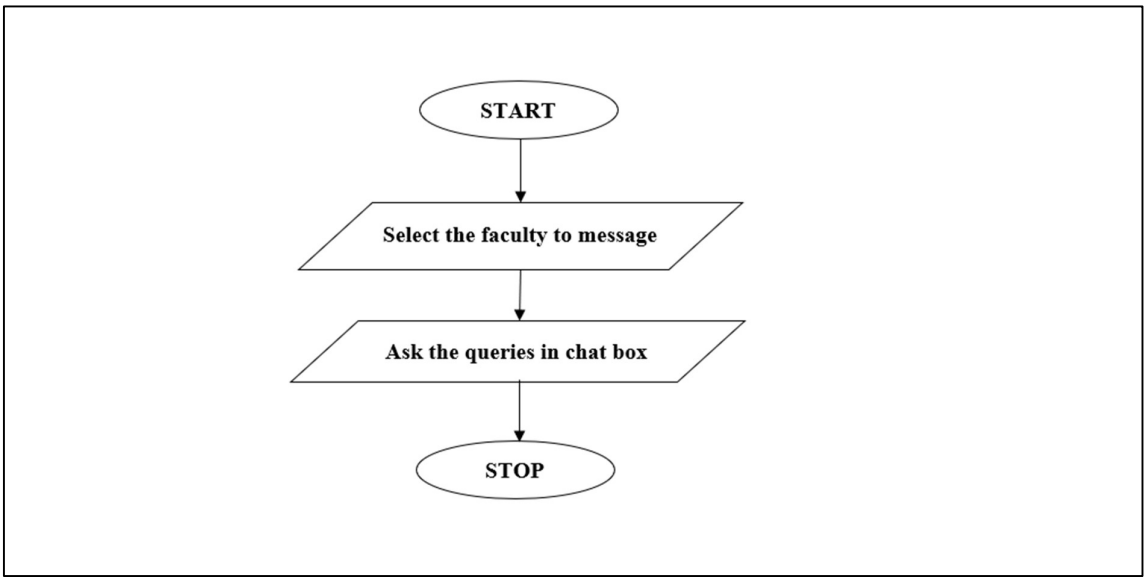

Fig: 2.17 Student Faculty Chat Activity Flowchart

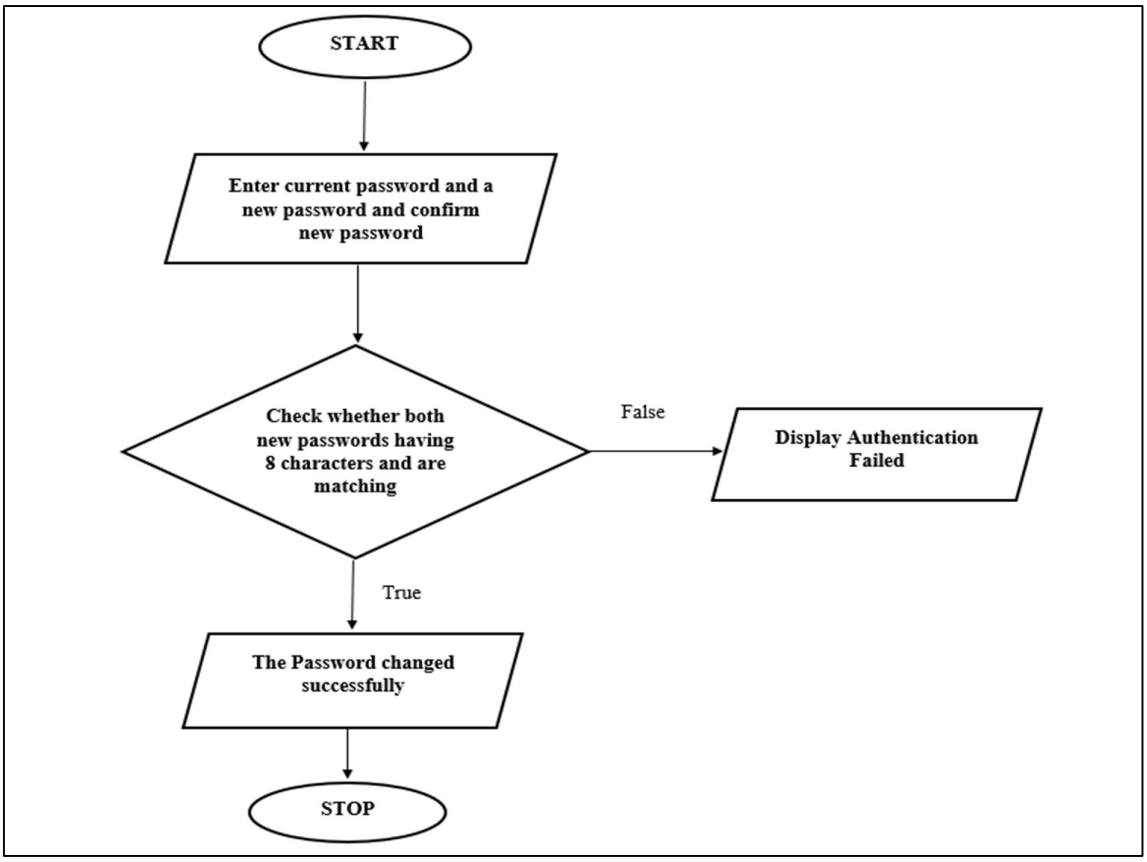

Fig: 2.18 Student Change Password Activity Flowchart

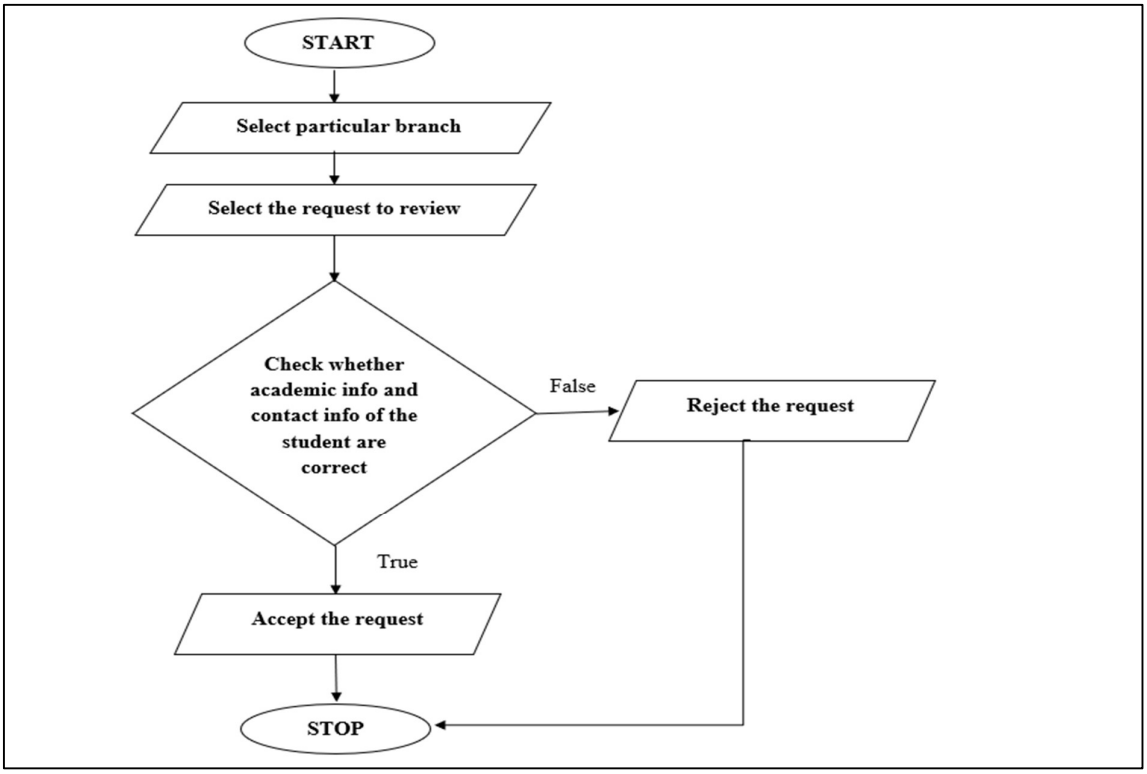

Fig: 2.19 Admin Student Request Review Flowchart

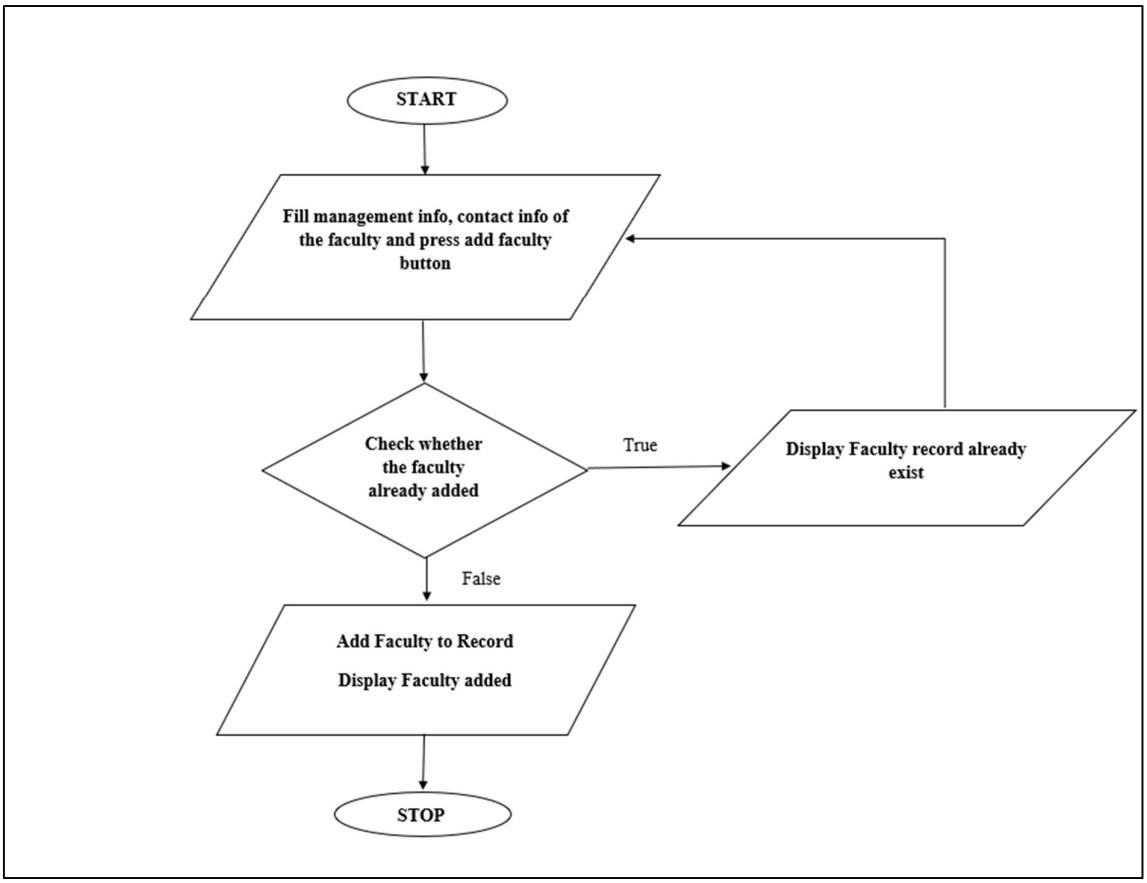

Fig: 2.20 Admin Add Faculty Activity Flowchart

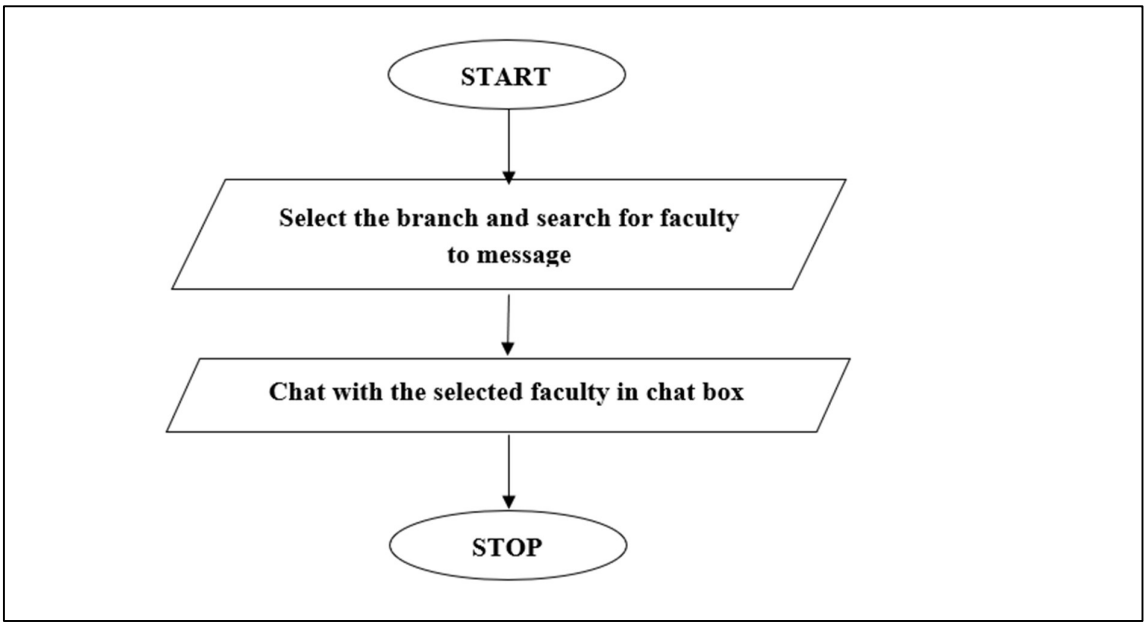

Fig: 2.21 Admin Faculty Chat Activity Flowchart

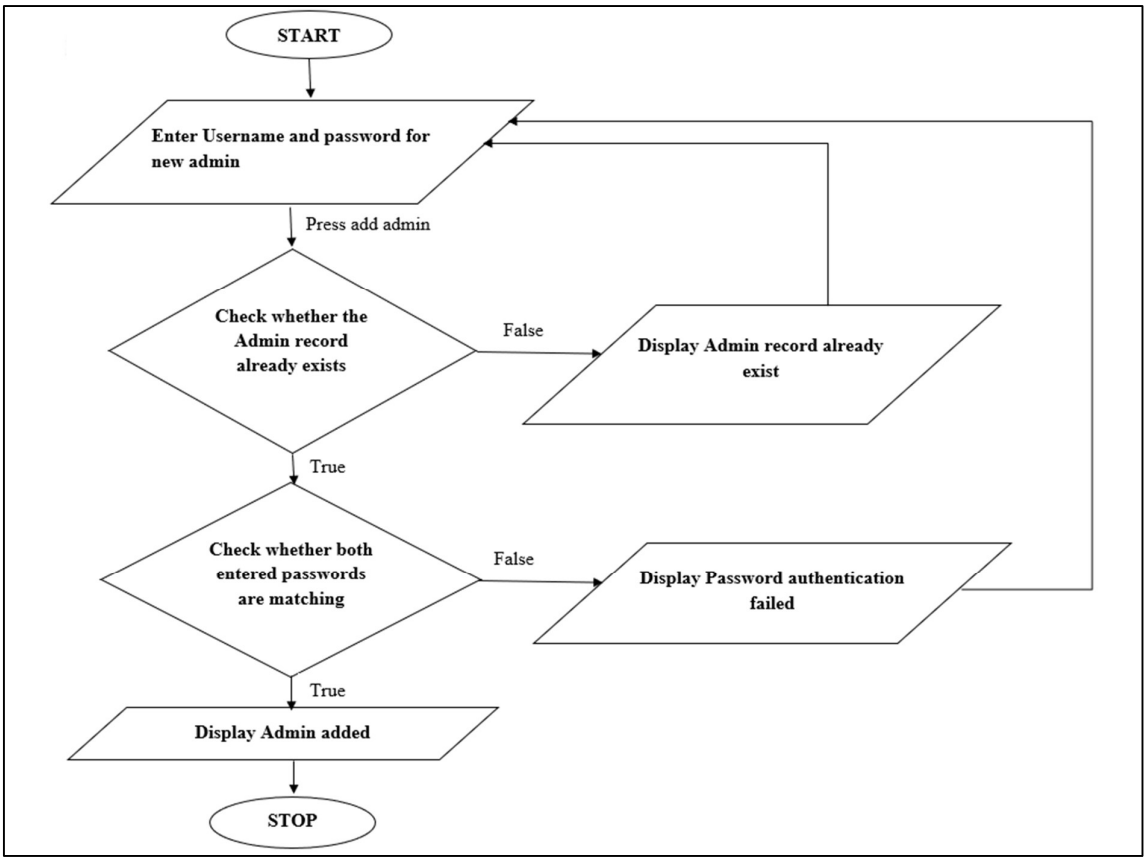

Fig: 2.22 Admin Add Admin Activity Flowchart

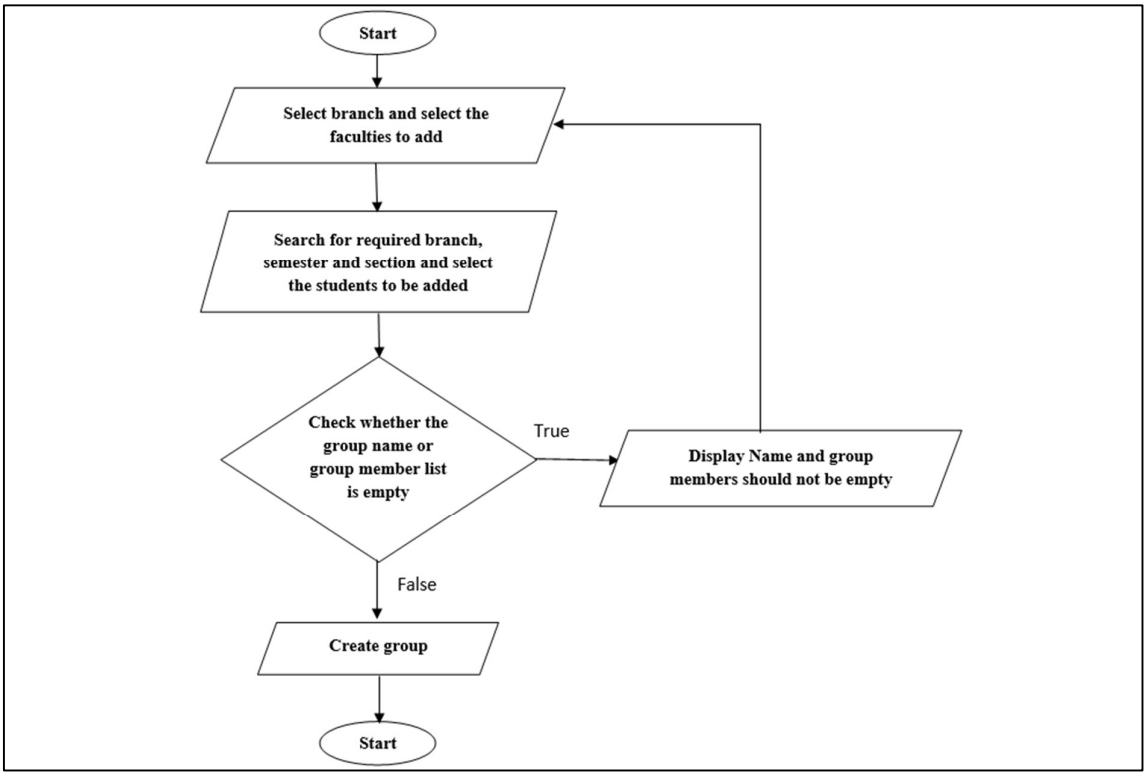

Fig: 2.23 Admin Create Group Activity Flowchart

2.6 Android APIs used

APIs used	Explanation
findViewById<View>(id)	Finds a view that was identified by the android:id XML attribute that was processed in onCreate.
FirebaseStorage.getInstance()	Firebase Firestore provides the **getInstance()** method, which opens a socket (only one, at any time of execution of the app) and instanciates the Firestore client.
FirebaseAuth.getInstance()	The entry point of the Firebase Authentication SDK. First, obtain an instance of this class by calling **getInstance()**
FirestoreRecyclerAdapter<T?, RecyclerView.ViewHolder?>(users)	**RecyclerView** adapter that listens to a **FirestoreArray** and displays its data in real time.
Glide.with(fragment).load(url) .into(imageView);	Glide supports fetching, decoding, and displaying video stills, images, and animated GIFs. Glide includes a flexible api that allows developers to plug in to almost any network stack

val adapter = GroupieAdapter() recyclerView.setAdapter(adapter)	Groupie abstracts away the complexity of multiple item view types. Each Item declares a view layout id, and gets a call back to bind the inflated layout. That's all you need; you can add your new item directly to a **GroupieAdapter** and call it a day.
Picasso.get().load(url).resize(50, 50) .centerCrop().into(imageView)	**Picasso** helps in setting image to the image view just using the url of the image given to it.

Chapter 3

RESULTS

3.1 Student

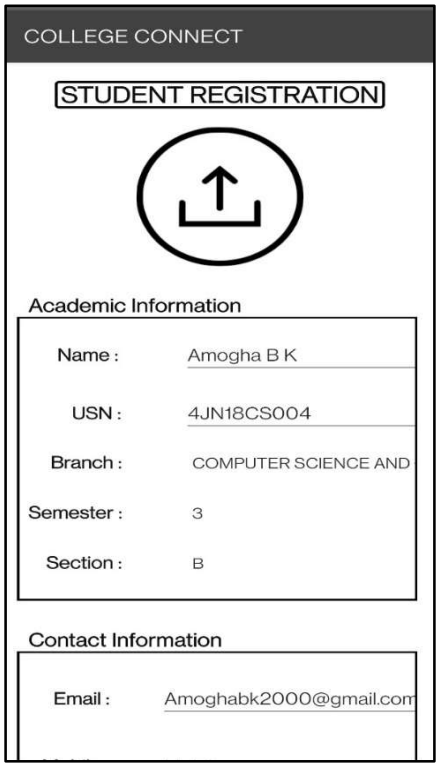

Fig 3.1 Student Registration Request Page

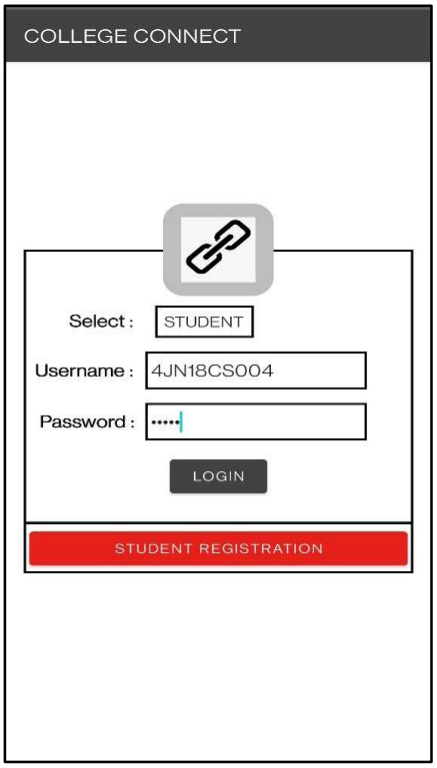

Fig 3.2 Student Login Activity

- Each student who is using this application for the first time has to send a request for registration to the admin by filling his academic and contact information in the registration form.
- After approval of the request, the student can log into his account by giving the login credentials on the login page. On the login page, login credentials are checked if the student is not registered or if the student's request is not approved in both cases student will get a message that the STUDENT RECORD DOESN'T EXISTS. Else student will be successfully logged into his account application will display the respective home page of the student.

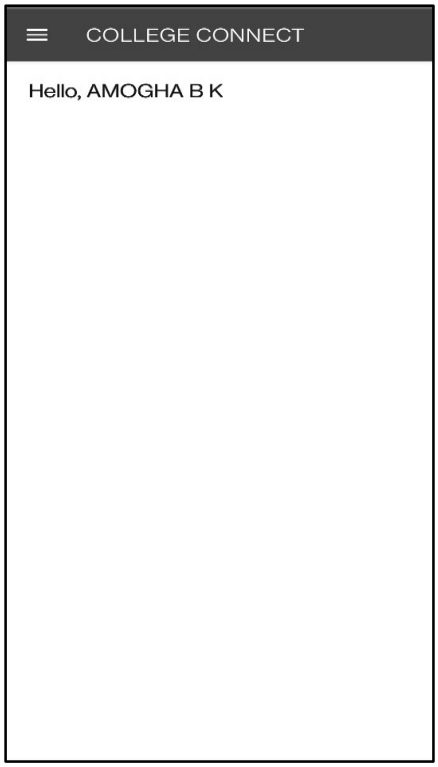

Fig 3.3 Student Homepage

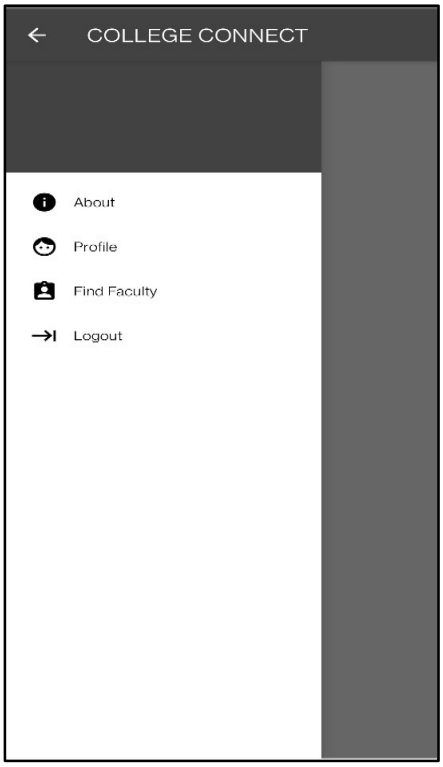

Fig 3.4 Student Navigation Drawer

- After successful login, the application will show the respective home page of the student. Where a student can his recent messages and recent group messages. But for the first time since there would not be any conversions started so the application will show only greeting messages.
- On clicking the button which is there in the top left corner of the home page, students can see other options like profile, find faculty, logout.

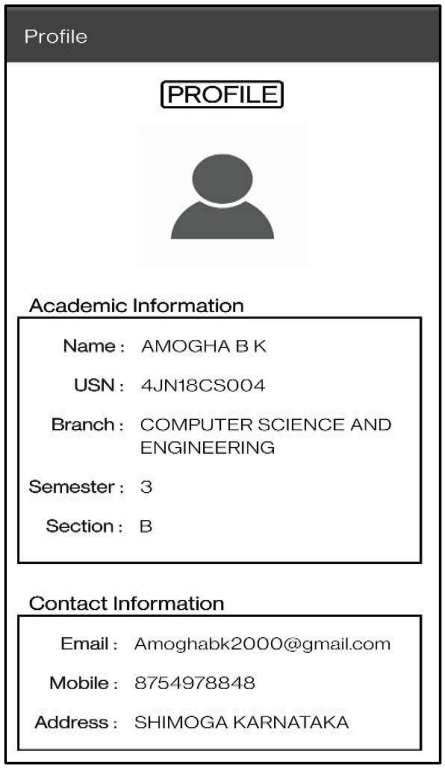

Fig 3.5 Student Profile Page

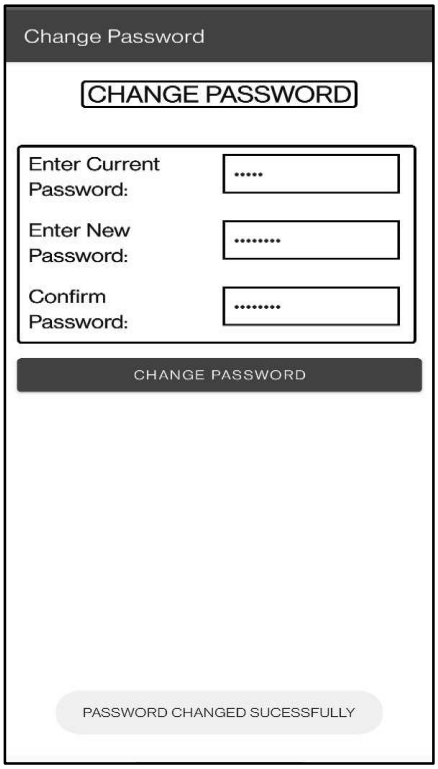

Fig 3.6 Student Change Password Page

- On clicking Profile in the homepage options, the student can view their academic and contact details. Also, they can change their profile picture by clicking their present profile picture the application will prompt the mobile gallery where the student can choose his new profile picture. After choosing the profile picture student have to click on the change profile picture button to upload the picture to the database.
- On the profile page, students can choose the change password button to change their account password. On the change password page student have to enter their current password and new password and have to click on the change password button.

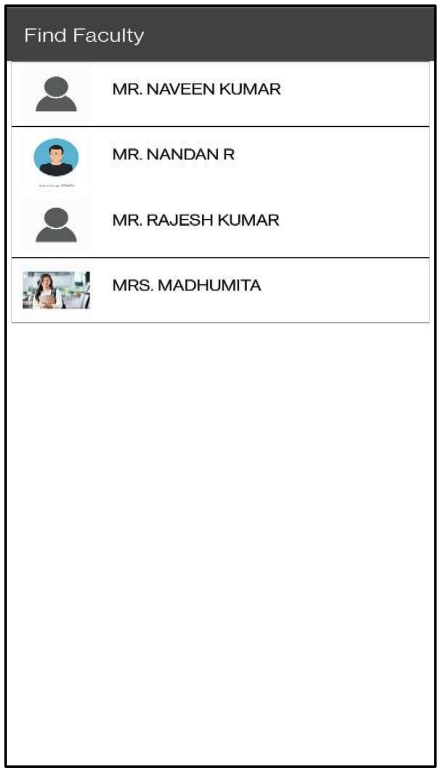

Fig 3.7 Student Find Faculty Page

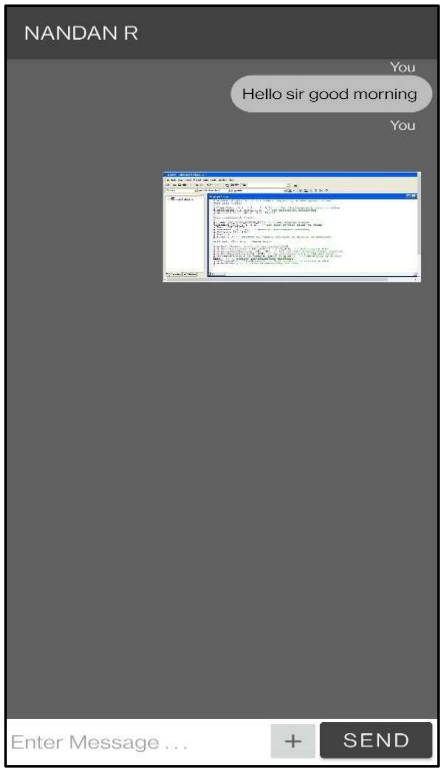

Fig 3.8 Student Faculty one-to-one chat Page

- On choosing the find faculty option in the homepage options students can view their department faculty names, on clicking the particular faculty name they can initiate one-to-one chat with their respective faculty.
- In the chat screen, all the previous conversions will be preloaded and there will be an edit text box on which student can write their message and send to their respective faculty by clicking on the SEND button. And the student can send images or documents by clicking on the plus button.

3.2 Faculty

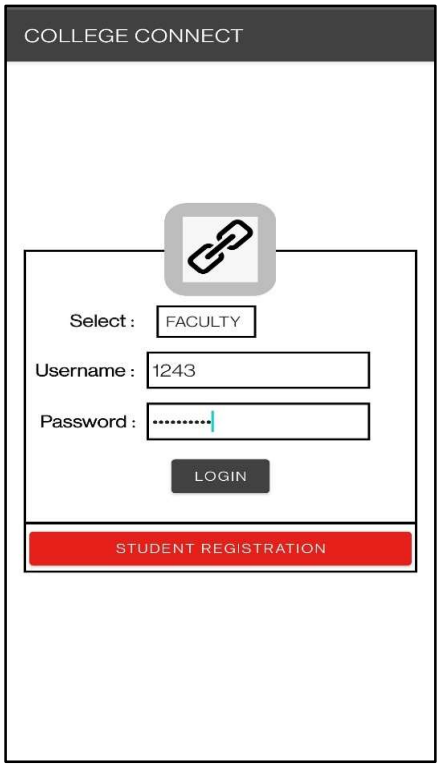

Fig 3.9 Faculty Login Page

Fig 3.10 Faculty Home Page

- Registered faculties can log in to their respective account by selecting user type as faculty and by giving their username and password. If login credentials are correct then they can view their homepage, if the password is wrong then the INVALID LOGIN CREDENTIAL message will be displayed. If the faculty record doesn't exist according to the given credentials then FACULTY RECORD DOESN'T EXIST message will be displayed.
- On the faculty home page, they can see their previous chat conversations(if any) else the homepage will be empty and only a greeting message will be displayed.

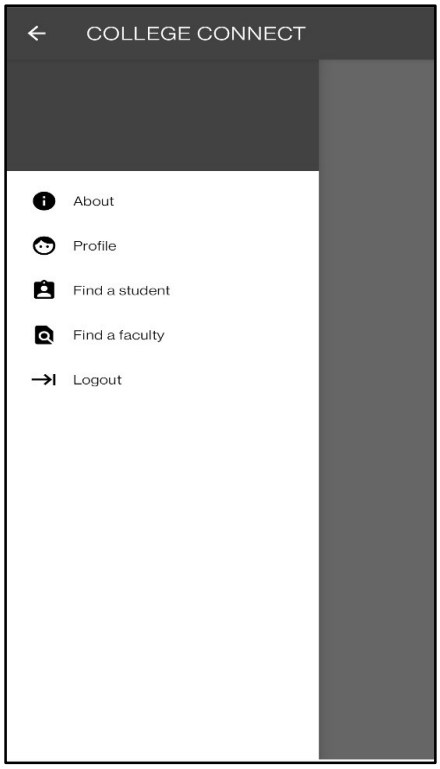

Fig 3.11 Faculty Navigation Page

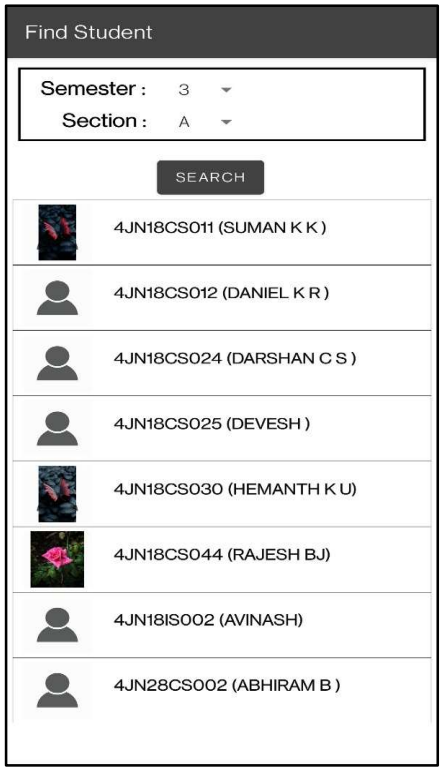

Fig 3.12 Faculty Find Student Page

- To see more options faculty have to click on the button which will be there in the top left corner which will contain options like profile, find student, find faculty logout.
- On clicking the Find a Student option, faculty will get a page where the faculty have to choose the semester and section of student which is intended to see by the faculty. Respective semester and section student list will be displayed. Where faculty can choose the student and can start a new conversation with the student.

Fig 3.13 Faculty one-to-one chat with Student Page

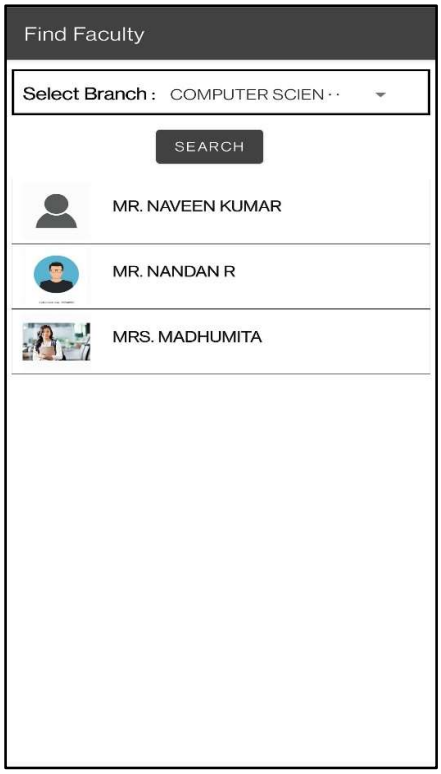

Fig 3.14 Faculty Find Faculty Page

- On the chat page faculty can send a message by typing the message in the edit text box and have to click on SEND button to send that message to the respective student similar to the text message faculty can send an image or document to the respective student.
- Similar to find a student, faculty can also start new conversations with the other faculties of various branches of the same institution. On clicking the faculty name a new conversation thread will be initiated where similar to sending the message to a student, faculty can send a message, send an image or document to another faculty.

3.3 Admin

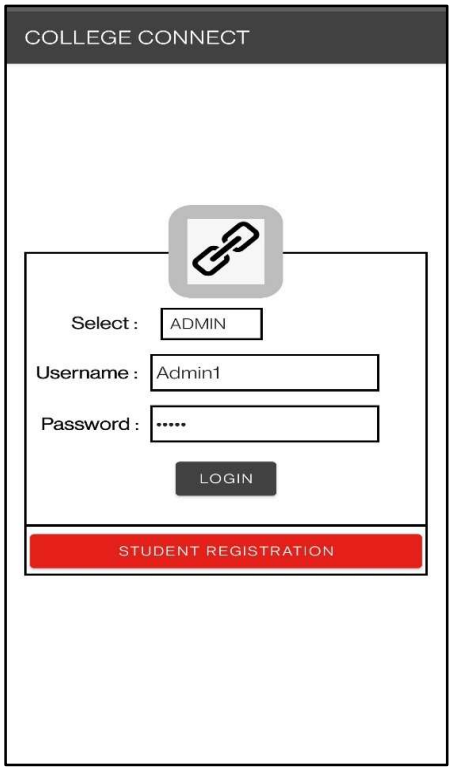

Fig 3.15 Admin Login Page

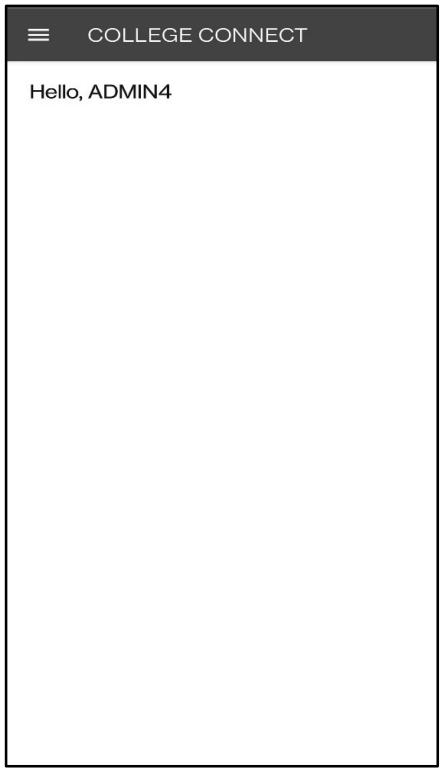

Fig 3.16 Admin Home Page

- In this College Connect application, there will be a special kind of user called ADMIN who maintains and manages the application. Admin also logs in similarly as students or faculty log in but by selecting user type as Admin.
- After successful authentication of the admin, he will be directed to the respective home page where his previous conversation threads will be displayed (if any).

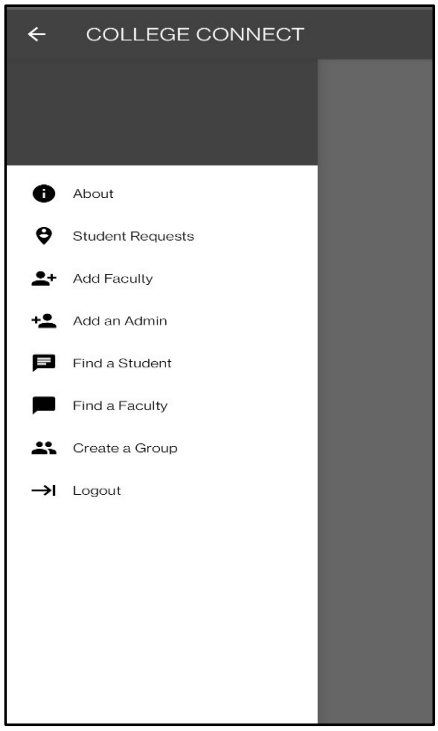

Fig 3.17 Admin Navigation Page

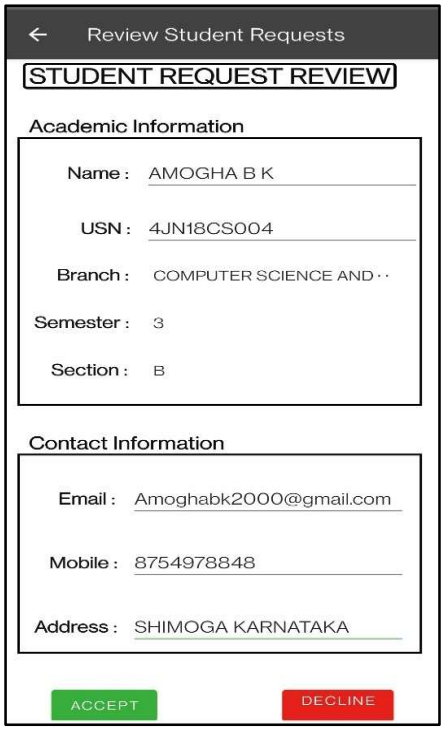

Fig 3.18 Admin Student Request Review Page

- Similar to other users, the admin also gets different options like Student Request Review, Add Faculty, Add Another Admin, Find a Student, Find a Faculty, Create a Group, and Logout.
- On clicking Student Requests, Admin can review the student request and can accept the request if all information uploaded is correct, or he can make changes in the uploaded data if needed and can accept, or he can reject the request.

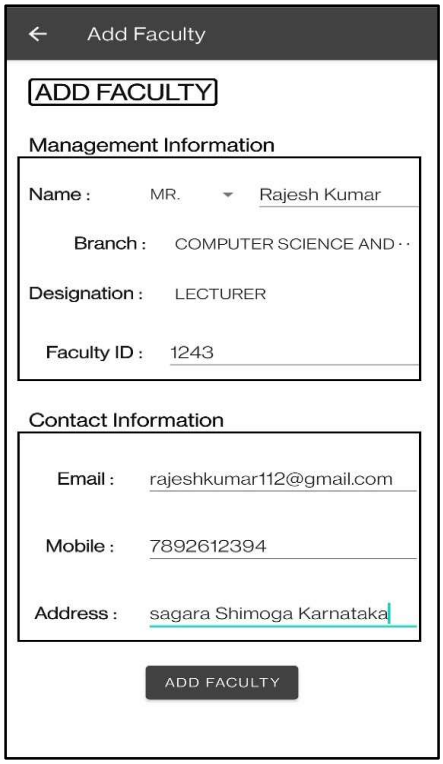

Fig 3.19 Admin Add Faculty Page

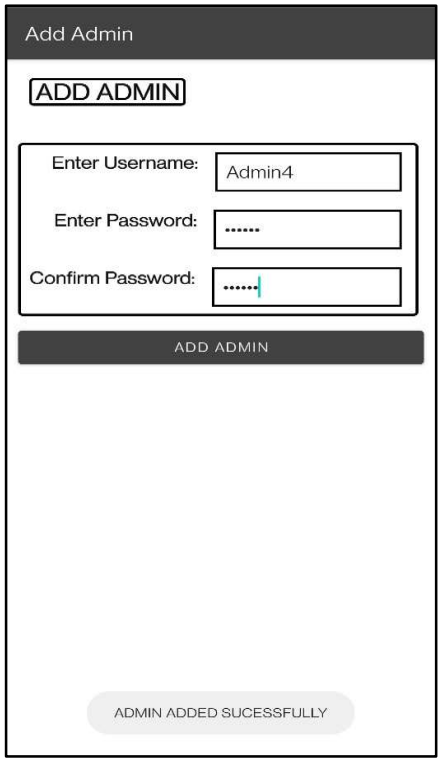

Fig 3.20 Admin Add Another Admin Page

- On clicking Add Faculty option from the admin navigation page, the admin can upload new faculty's information and can create his account. By default, the new faculty's account username will be his faculty ID and the password will be his mobile number, which can be changed any time by the faculty by logging in to his account and also can upload his profile picture.
- On clicking Add an Admin option, the current admin can add a new Admin member by entering a username and password. The username will be checked if already exists then the ADMIN RECORD EXIST message will be displayed. Else new record will be added and ADMIN ADDED SUCCESSFULLY message will be displayed.

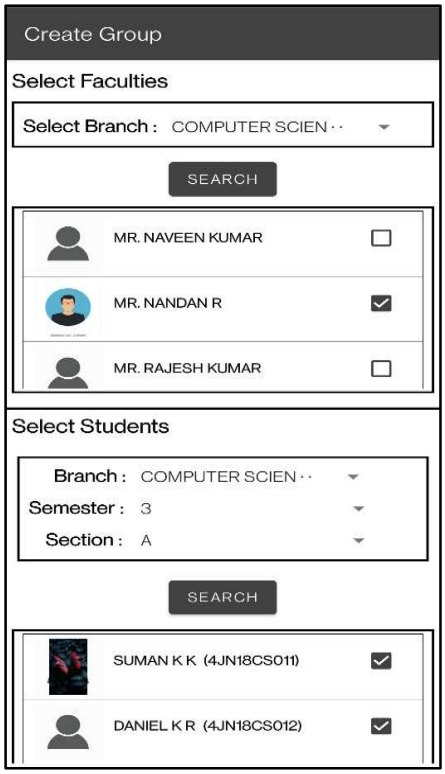

Fig 3.21 Admin Create Group Page

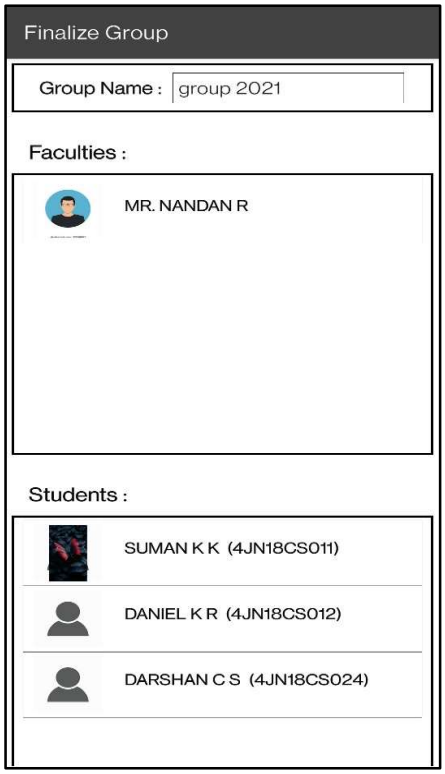

Fig 3.22 Admin Finalize Group Page

- Admin can create new groups of faculties and students or only of faculties by choosing the create a group option in the admin homepage option. In Admin Create Group Page admin have to select faculties by filter using branch and by checking the checkbox which is near to the faculty name. Similarly, Admin can choose students by checking the checkbox which is near the student name. After choosing faculties and students, Admin has to click on the continue next button.
- Next Admin will be navigated to Finalize Group Page where he can see all selected faculties and students list and he has to give a name to the group. If the group name already exists a message will be displayed to give a new name and the admin can click on CREATE button then a group will be created. If any mistakes are there in selecting the user admin can click on the DELETE button then the current group created will be deleted.

3.4 Firestore Contents

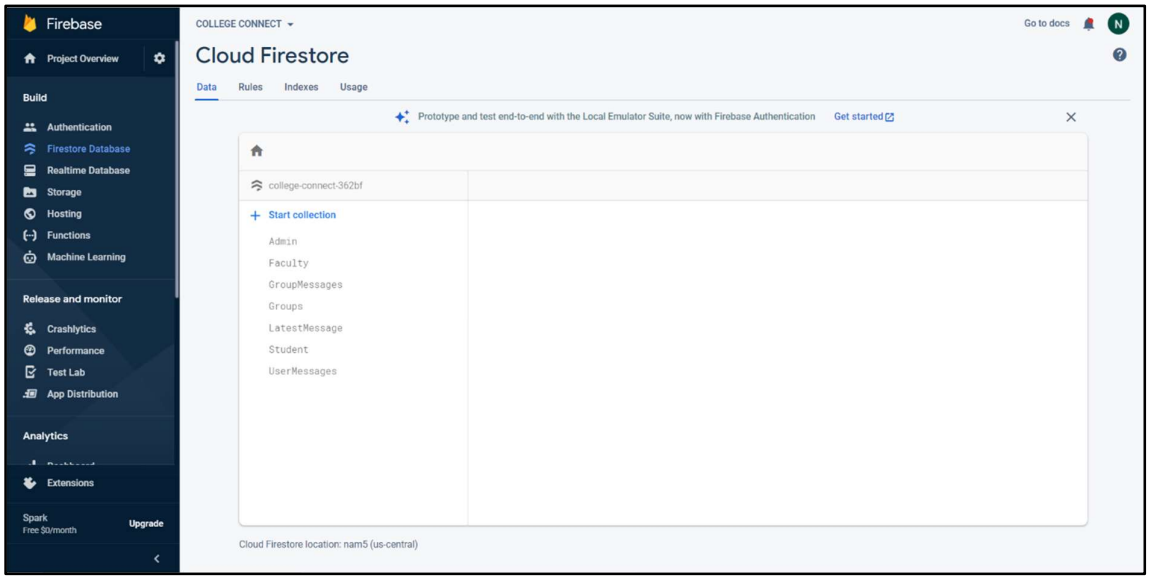

Fig 3.23 Firestore Collections

In this application, we have used Google Firestore as a database where we are maintaining different collections namely Admin, Faculty, Group Messages, Groups, Latest Messages, Student, User Messages.

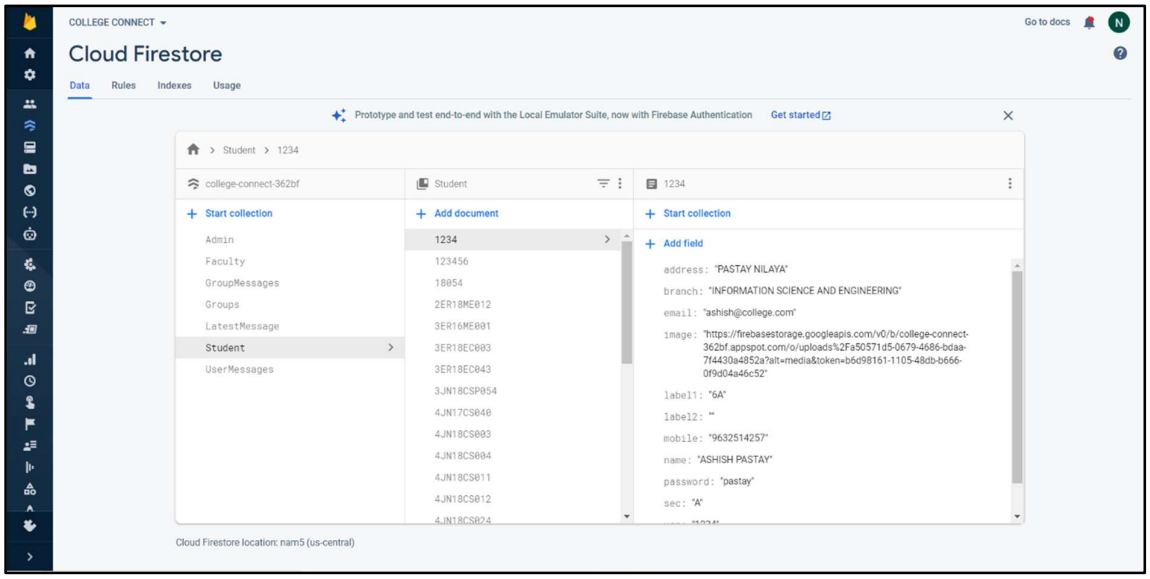

Fig 3.24 Student Collection

In student collection, we will store academic and personal details of students like name, email, branch, address, section, USN, semester, mobile no, and some special data like image which will contain Firebase Storage link of student profile photo, some labels which are needed to store information regarding groups in which particular student is assigned by the admin and password which is required for student login.

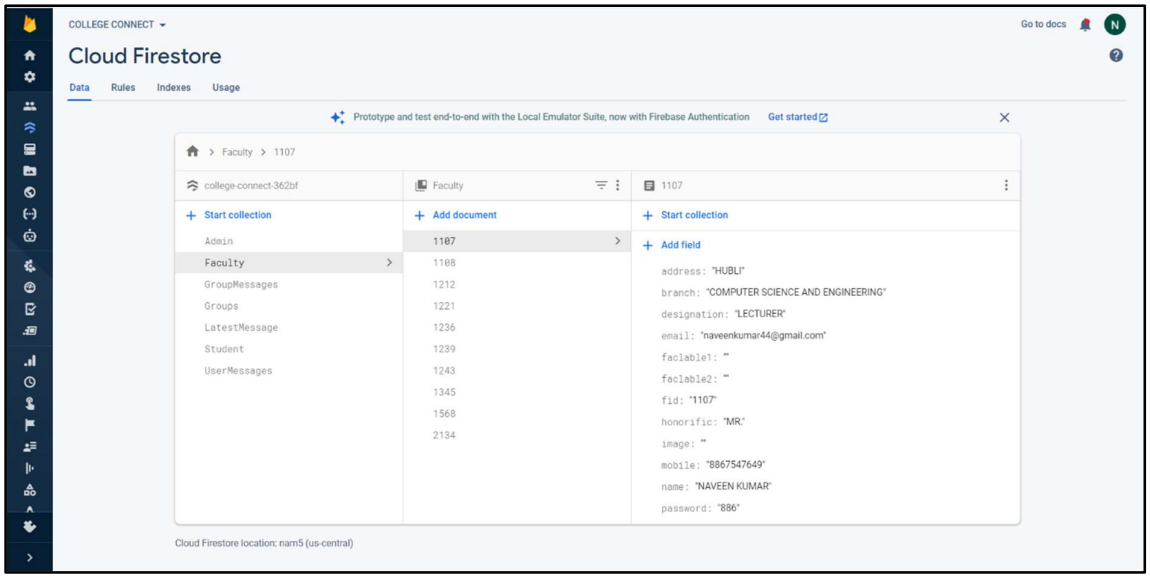

Fig 3.25 Faculty Collection

In faculty collection, we will store details of faculties like name, email, branch, address, designation, FID, mobile no, and some special data like image which will contain Firebase Storage link of faculty profile photo, some labels which are needed to store information regarding groups in which particular faculty is assigned by the admin and password which is required for faculty login.

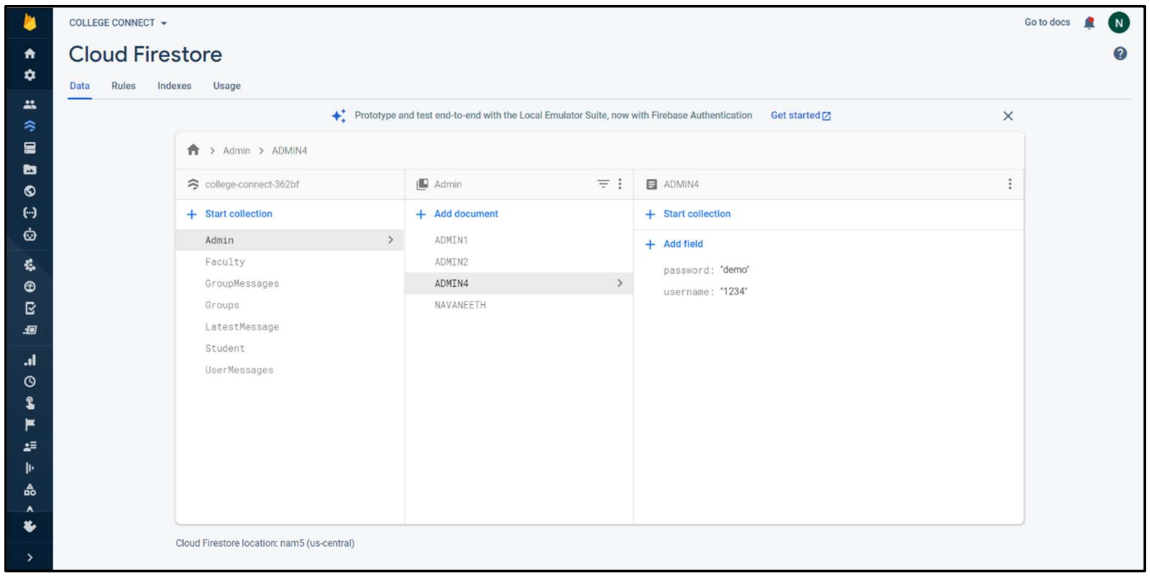

Fig 3.26 Admin Collection

In Admin collection, we will store admin login credentials like username and password which is needed for admin login in the application.

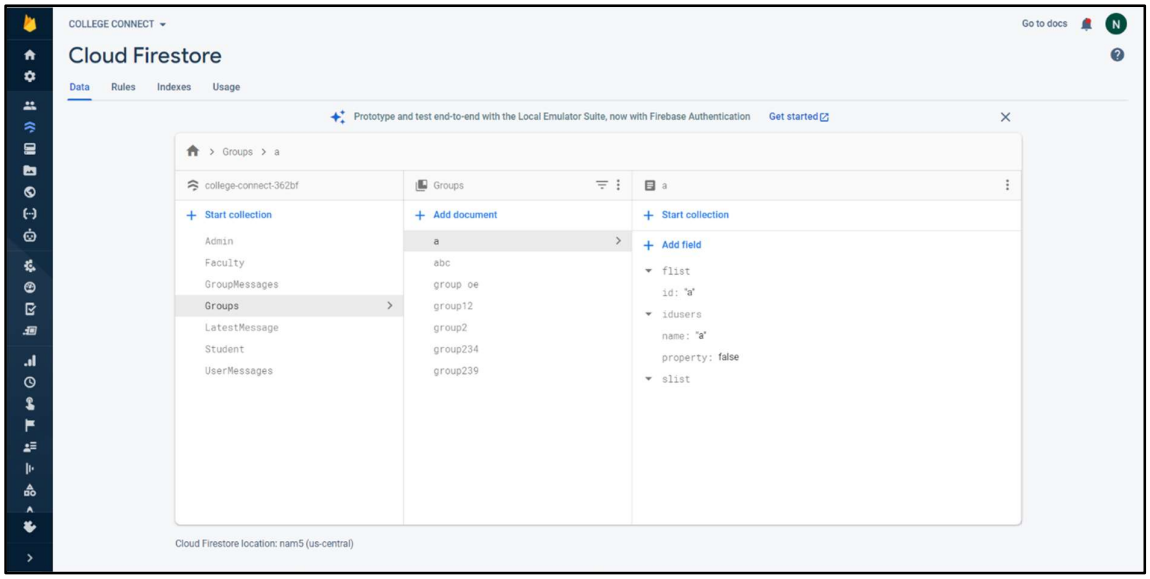

Fig 3.27 Groups Collection

In Groups collection we will store chat group information like faculty id list, student id list, group name, and one special data called property which will store whether in the particular group student can chat or not.

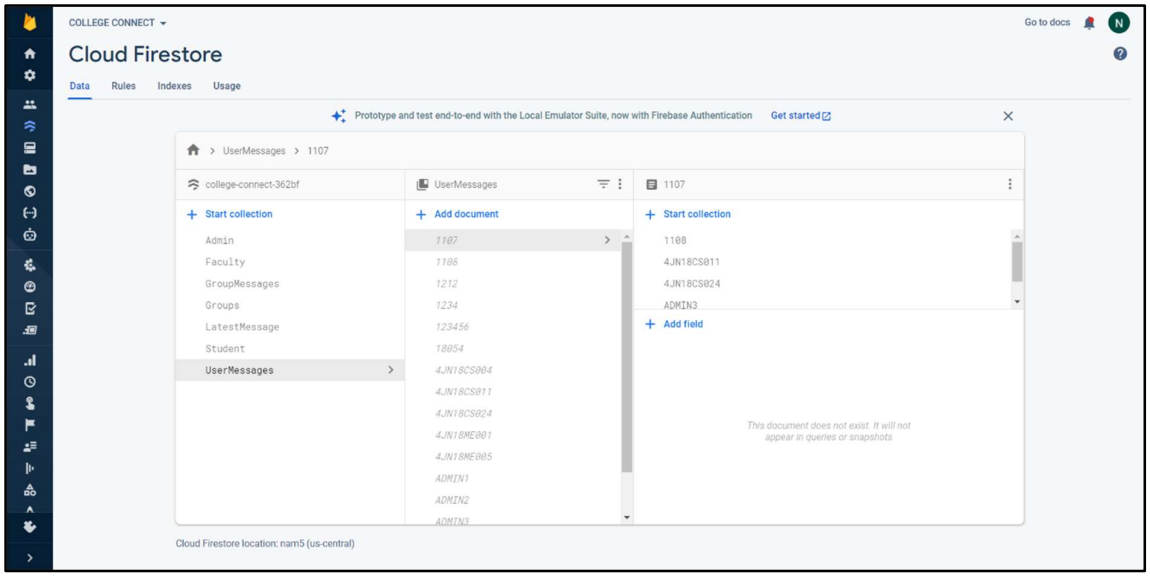

Fig 3.28 User Messages Collection

In User Messages Collection we will store one-to-one chat message information like from ID, to ID, timestamp, to name, message type, and message. Which will be helpful in saving and keeping a track of one-to-one chat message treads.

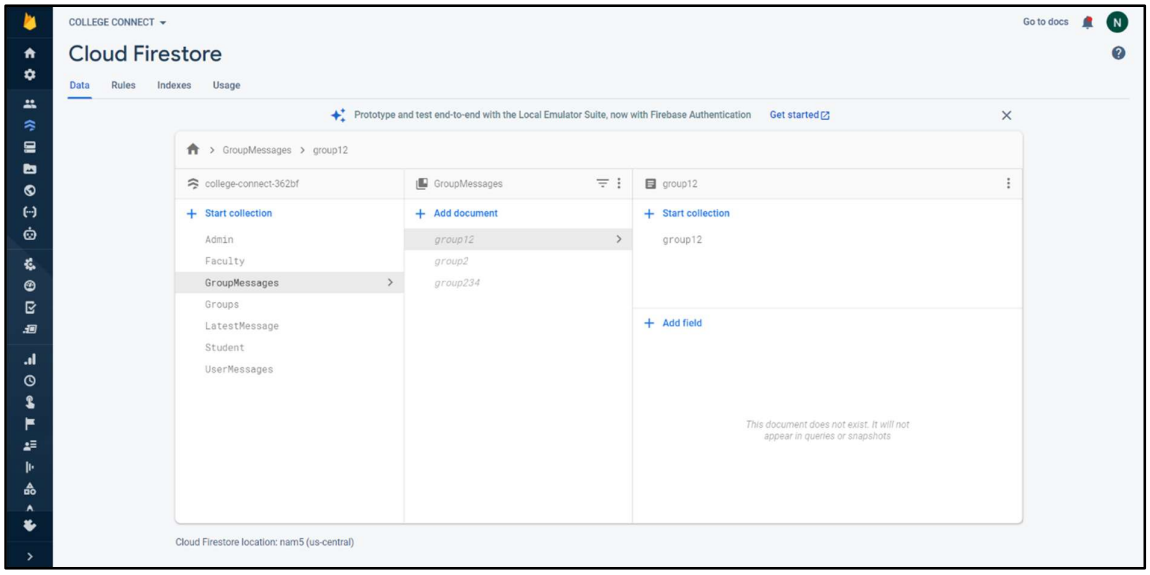

Fig 3.29 Group Messages Collection

In Group Messages Collection we will store group message information like from ID, group name, timestamp, message type, and message. Which will be helpful in saving and keeping a track of group chat treads.

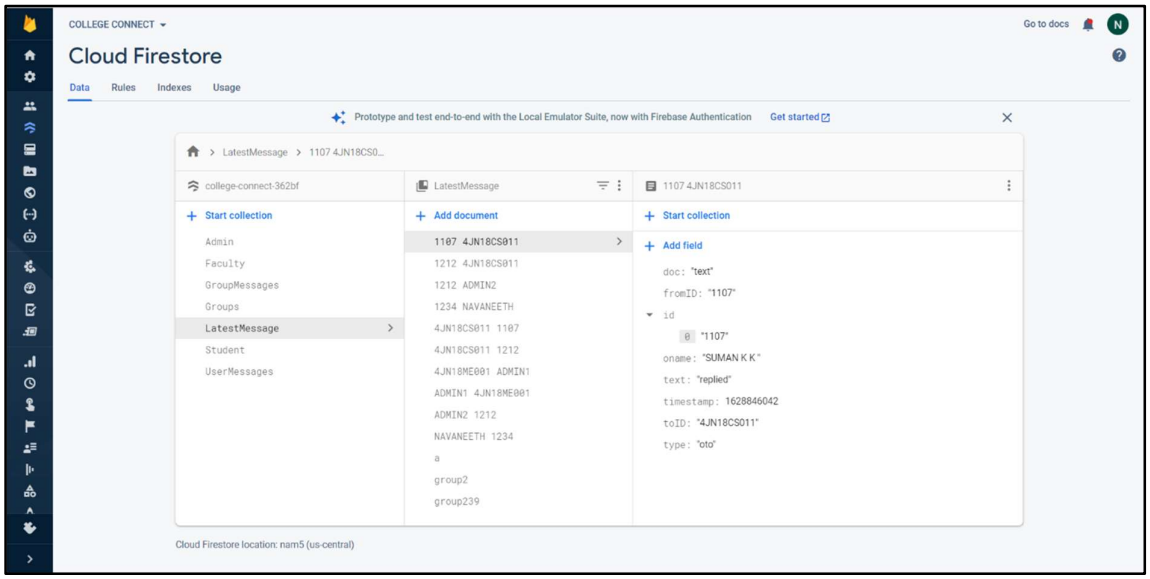

Fig 3.30 Latest Messages Collection

In Latest Messages Collection we will store each one to one and group's latest messages which will be updated when a new message is sent. This is useful in displaying the homepage of students, faculty, and admin where the user can see his last message conversations and can navigate to that message by clicking on it.

Chapter 4

CONCLUSION

College Connect is the application that provides the facility to store the information about the personal and academic details of students. It will help in one-to-one communication between faculties, or in between faculty and student. Also, College Connect helps in creating groups for group communication between faculties or in between faculties and students. Since most of the educational institutes use other 3rdparty messengers to communicate but it may lead to security and confidentiality issues. It's our small try to that using Android Application Development skills we can build any type of project to the work of people.

4.1 Future Scope

In a nutshell it can be summarized that the future scope of the project circles around the maintaining information regarding:

- One can give more advanced software for Educational Institutes for communication.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.
- Further development can be done in the interface of the application.
- Further development can be done in the security checks and encryption in the transfer of data from database to application can be implemented to increase the security of the application.

REFERENCES

[1]. Erik Hellman, **"Android Programming – Pushing the Limits"**, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197

[2]. Dawn Griffiths and David Griffiths, **"Head First Android Development"**, 1st Edition, O'Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341

[3]. Bill Phillips, Chris Stewart and Kristin Marsicano, **"Android Programming: The Big Nerd Ranch Guide"**, 3rd.Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

[4]. https://firebase.google.com/docs/reference/android/com/google/firebase/firestore/FirebaseFirestore

[5]. https://square.github.io/picasso/#introduction