Software Design Specifications
for
Student Help Desk

Version 1.0 approved

Prepared by:
Group id – CS61

Mansi Bhatt	1913016	btbtc19105_mansi@banasthali.in
Riyanshi Verma	1913075	btbtc19047_riyanshi@banasthali.in
Vanshika	1913162	btbtc19046_vanshika@banasthali.in

Project Mentor:
Ms. Sakshi Indolia
Assistant Professor
Department of Computer Science

Banasthali Vidyapith

Date: 23/01/2022

Table of Contents

1. Introduction

1.1 Purpose of this document

This Software Design Specification (SDS) is made with the purpose of outlining the software architect and design of the StudentHelpDesk (SHD) in detail. The document will provide the developers an insight to the solutions of the problems faced by the placement cell and students of various colleges. Moreover this facilitates communication and understanding by providing several views of SHD.

1.2 Scope of the development project

The SDS would demonstrate how the design will accomplish the functional and non-functional requirements captured in the SRS. It will provide a framework to the programmers through describing the high level components and architecture, subsystems, interfaces, database design etc. It is achieved through the use of design patterns, design patterns, sequence diagrams, relational models and user interfaces.
The purpose of this StudentHelpDesk System is to manage the inconvenience that the placement cell faces while collecting and organizing the data of the students of various courses and provide them to the companies for the placement or internship drive. So, the system would store all the academic details, personal details and other related documents (i.e., Resume or marksheets) of the students who wish to sit for the placement/internship drive. This would also store the data of the companies who come to offer jobs to the students.

The Details of the students along with the resume will be there for all the admins to view and analyze. It will also have a notification section to keep the students updated about the company i.e., the details of the company, the eligibility criteria, date of exams, etc. All the details regarding the placement activities and procedures for the company (i.e., the selection procedure) would be provided by the admin/company to all the students by the notification section only. There may or may not be direct contact of the company with the student, depending upon company's choice. The system would be based on **Firebase database**. Firebase is database, which is real time and allows us to store data in the form of collections and documents. The whole database is divided into *'Collections'*, which contains '*Documents*', which in turn contain '*Fields*' and '*Data*' along with *'Sub Collections'*.
We hope to provide a comfortable user experience for both the interfaces i.e., the admin and the user.

1.3 Definitions, acronyms, and abbreviations

S.NO.	TERM	DESCRIPTION
1.	admin	Administrator
2.	API	Application Program Interface
3.	app	Application
4.	FAQ	Frequently Asked Questions
5.	GUI	Graphical User Interface
6.	JSON	JavaScript Object Notation
7.	NA	Not Applicable
8.	NoSQL	not Sequential Query Language
9.	PDF	Portable Document Format
10.	SDS	Software Design Specification
11.	SHD	StudentHelpDesk – Project Title

12.	SRS	Software Requirement Specification
13.	TPO	Training and Placement Officers

1.4 References

- Pressman, Roger S. Software Engineering: A Practitioner's Approach. New York: McGraw-Hill, 1987.
- Authors, K.K. Aggarwal, Yogesh Singh, Software Engineering. New Age International (P) Limited, 2008.
- http://cse.anits.edu.in/projects/projects1920C15.pdf

1.5 Overview of document

This document describes the architectural design of the StudentHelpDesk. It consists of high level components and their interactions, suitable architectural patterns, physical arrangement of components and design decisions applied to the whole SHD.
As we go down the SDS, we have component and detail design which includes design patterns, sequence diagrams, class diagrams, database design in detail and user interface design with screenshots of the interface.

The next section - **System architecture description,** introduces the various components and subsystems of the app SHD. The high level components and their interactions, suitable architectural patterns, physical arrangement of components and design decisions applied to the whole system. This section makes it clear the interrelationships and dependencies among the various components. Structure charts are used for this.

The third portion of the SDS is on **Component and detailed design,** which includes design pattern, sequence diagrams and database design in detail and user interface design with screen shots of the interface.

The following forth section of this document is **Reuse and relationships to other products.** This section includes the following subsections -- how reuse is playing a role in your product design, how reuse is playing a role in the product implementation etc.

The fifth part of this SDS is **Design decisions and tradeoffs,** which will help the readers to understand the design used for the project better. This part also includes all the trade-off ideas.

The heading that follows will consist works as a rough documentation, so the program of one developer can be understood easily. The main aim of this division is making the code construction phase easier for the programmer. Hence making it very vital. This section is of **Pseudocode for various components.**

The succeeding bit included supplement material and additional information on the topic explored in the contents of the text. Hence concluding the document.

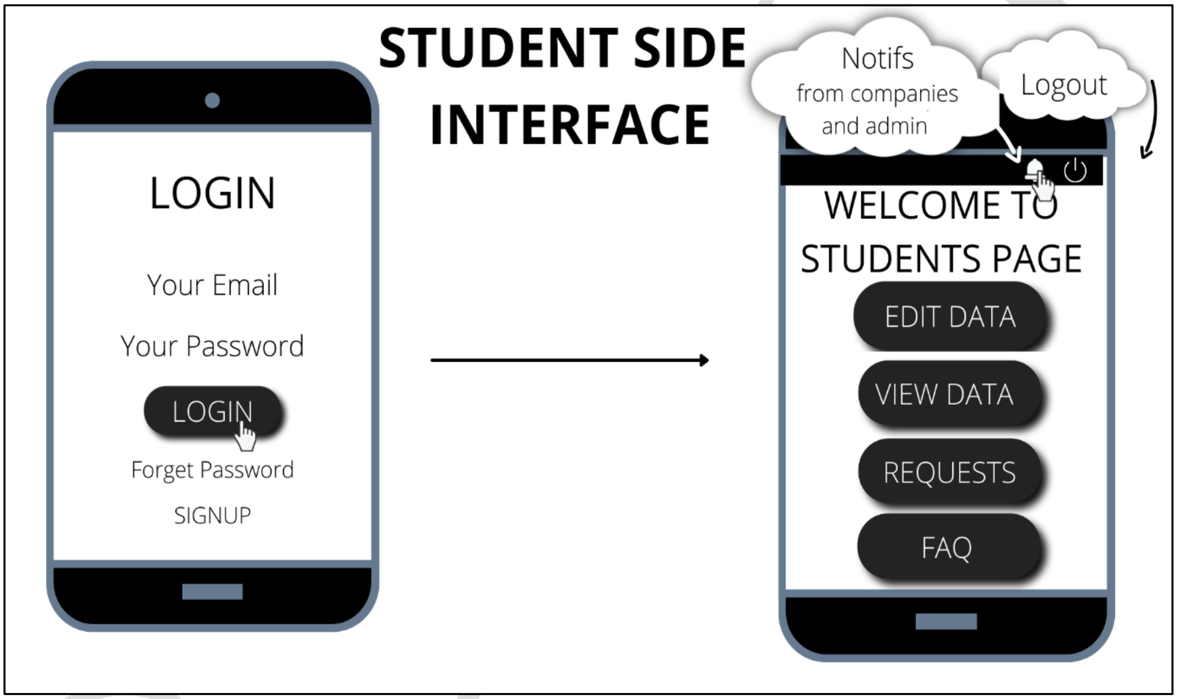
2. System architecture description

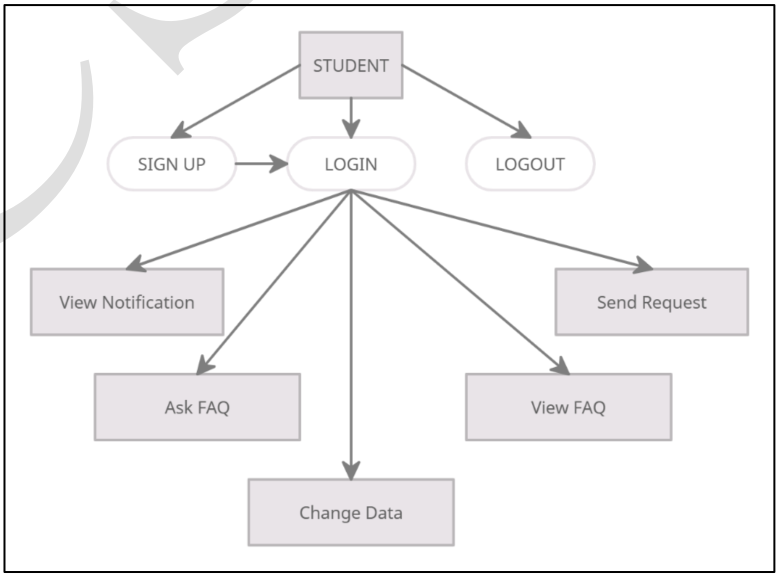
2.1 Overview of modules / components

The various components of the system includes:

1. Student

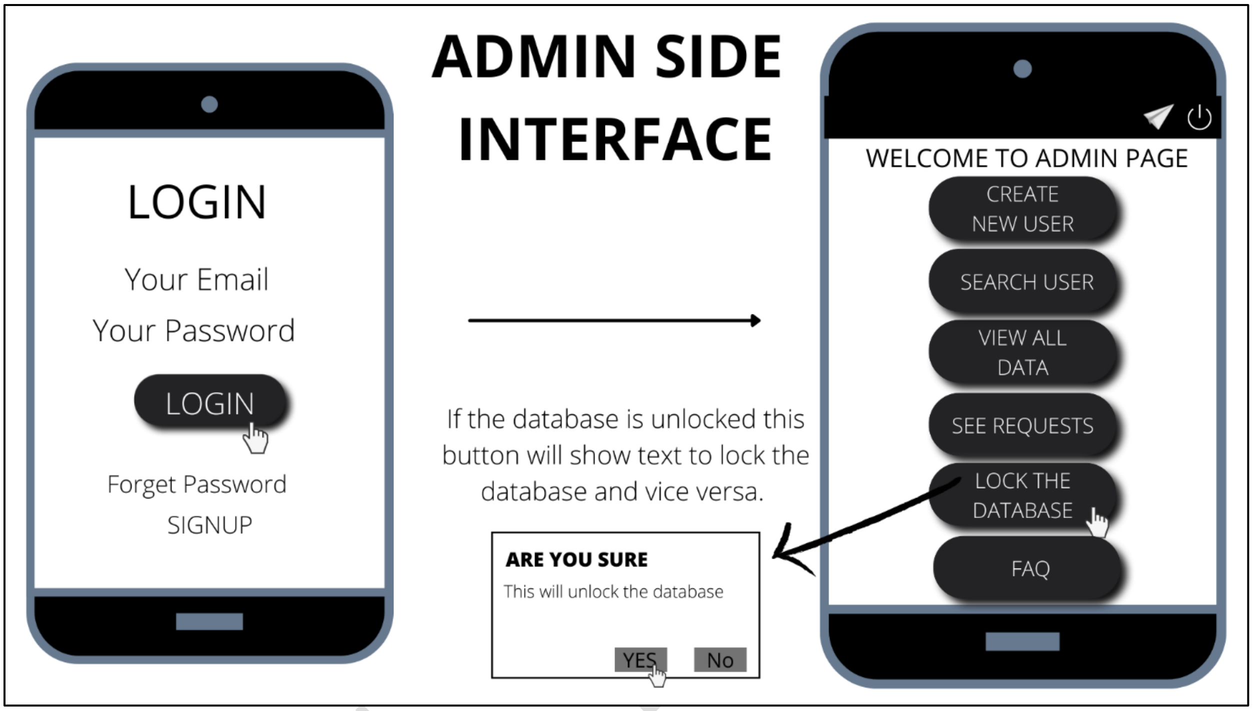
Student is one of the major component of this app. A new student needs to *sign up* by giving their complete details i.e., Personal details, Academic details and other related documents as mentioned by their college admin while they register the college in the app. Students can *update their profile information in real time*, only when database is unlocked. Also, only changeable fields would be allowed to be changed directly by students, whereas non-changeable fields requires sending a *change request to admin* (e.g. – Name, college id, etc.). They *will receive notification* from the companies and admin as and when the need comes.

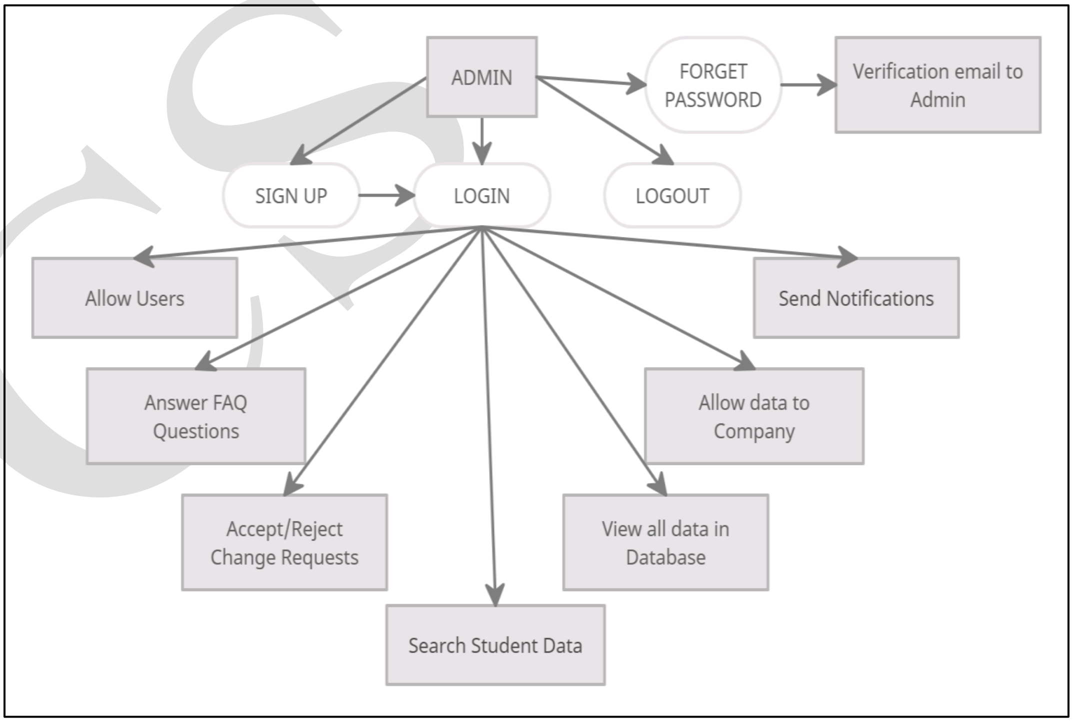




2. Admin

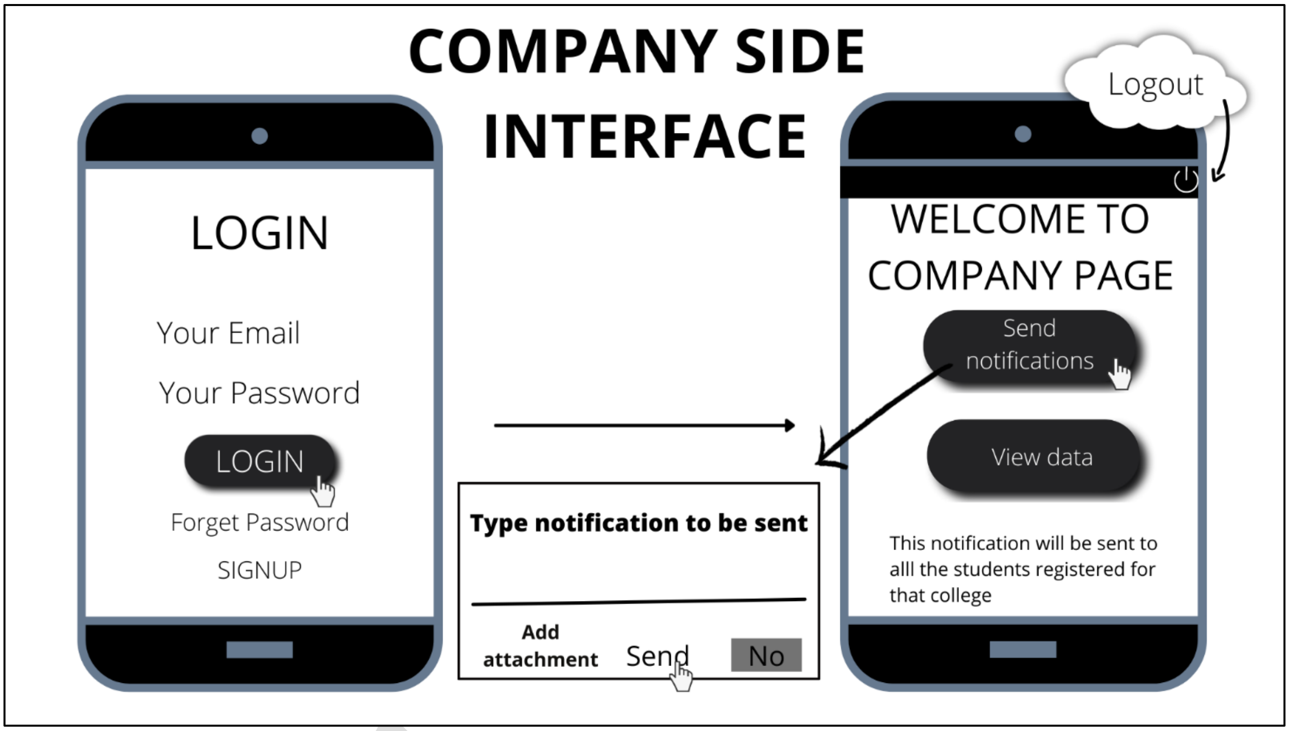
Admin is another major component of the app. It has *supreme power* of the application SHD. Admin *provides approval to Student and the corporate registration*, thereafter, they can follow up for signup or login procedure. Admin holds the responsibility for *maintaining and updating the whole system* and has to authority to *notify the students* regarding any changes in the procedure or selection. He has the power to *lock the database* and notify students for the same, before sending the data to required company.

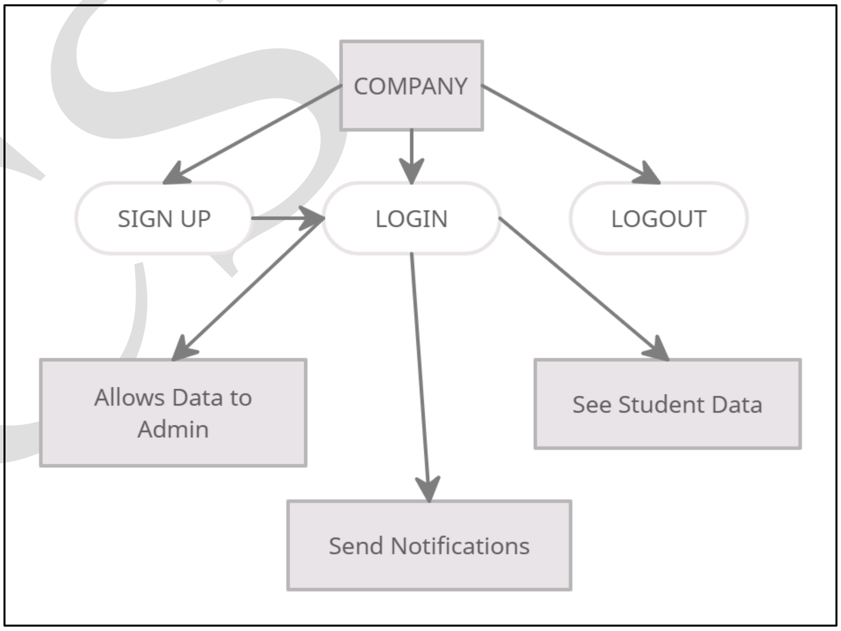




3. Company

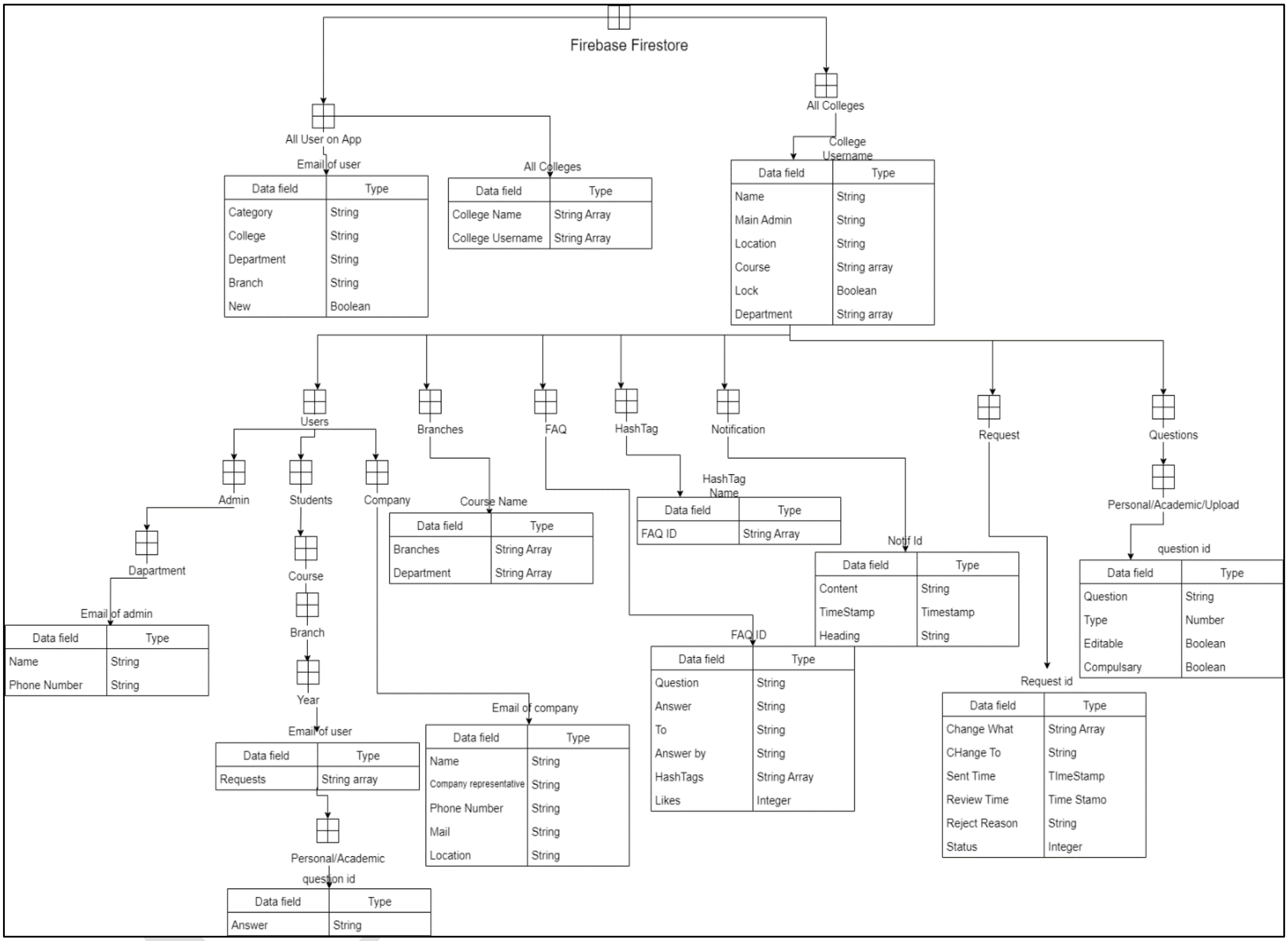
Company is a minor component of the app. The Company *initially has to register* using the details that admin provides them. The Company can *notify the Admin or the TPO* as well as *send direct notifications to students* regarding placement/internship after signup. It *can view shortlisted students' details* (based on eligibility criteria) as provided by TPO. Each company will have Company Representative for guiding the procedure of drive in campus registered.





4. Firebase database- The database used for this app is Firebase Firestore. It is a real-time database that stores information in the form of collections and documents. A collection contains various documents and a document contains collections as well as data fields. Firebase is a NoSQL database i.e. It doesn't require tables to store data but it stores data in a JSON-like document. The following is the proposed database's diagram.

Database Diagram for Firebase Firestore



Data field	Type
Category	String
College	String
Department	String
Branch	String
New	Boolean

Data field	Type
College Name	String Array
College Username	String Array

Data field	Type
Name	String
Main Admin	String
Location	String
Course	String array
Lock	Boolean
Department	String array

Data field	Type
FAQ ID	String Array

Data field	Type
Branches	String Array
Department	String Array

Data field	Type
Content	String
TimeStamp	Timestamp
Heading	String

Data field	Type
Question	String
Type	Number
Editable	Boolean
Compulsary	Boolean

Data field	Type
Name	String
Phone Number	String

Data field	Type
Question	String
Answer	String
To	String
Answer by	String
HashTags	String Array
Likes	Integer

Data field	Type
Change What	String Array
CHange To	String
Sent Time	TImeStamp
Review Time	Time Stamo
Reject Reason	String
Status	Integer

Data field	Type
Name	String
Company representative	String
Phone Number	String
Mail	String
Location	String

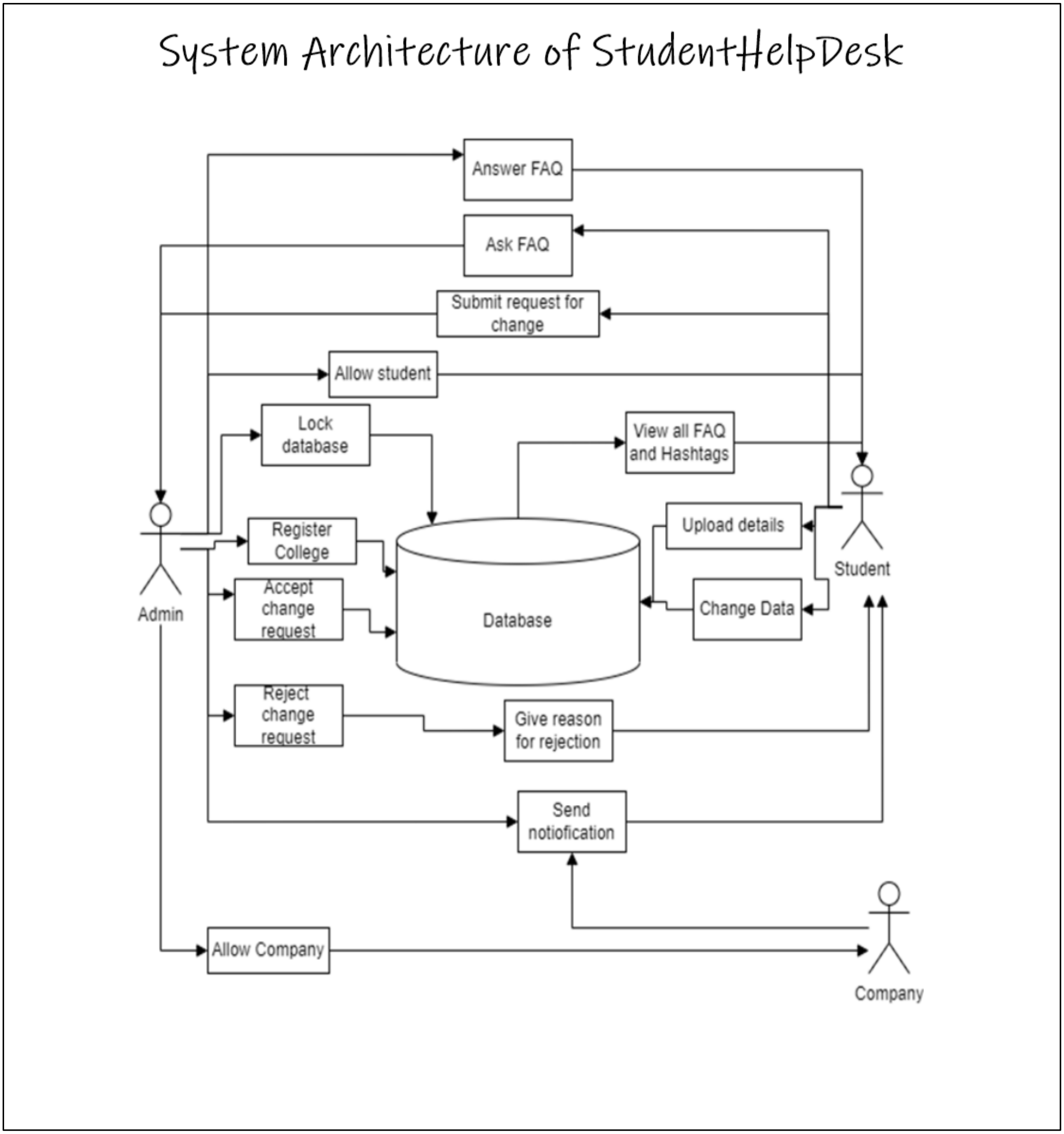
Data field	Type
Requests	String array

Data field	Type
Answer	String

In the above diagram, we can see collections and documents. The fields that each document contains are shown in the table along with it's type. We will also be using other Firebase functions like authentication for authenticating the user i.e. allowing user to login and thus get access to database components. It also has feature of Firebase storage which we will be using for storing multimedia files like resume and photos.

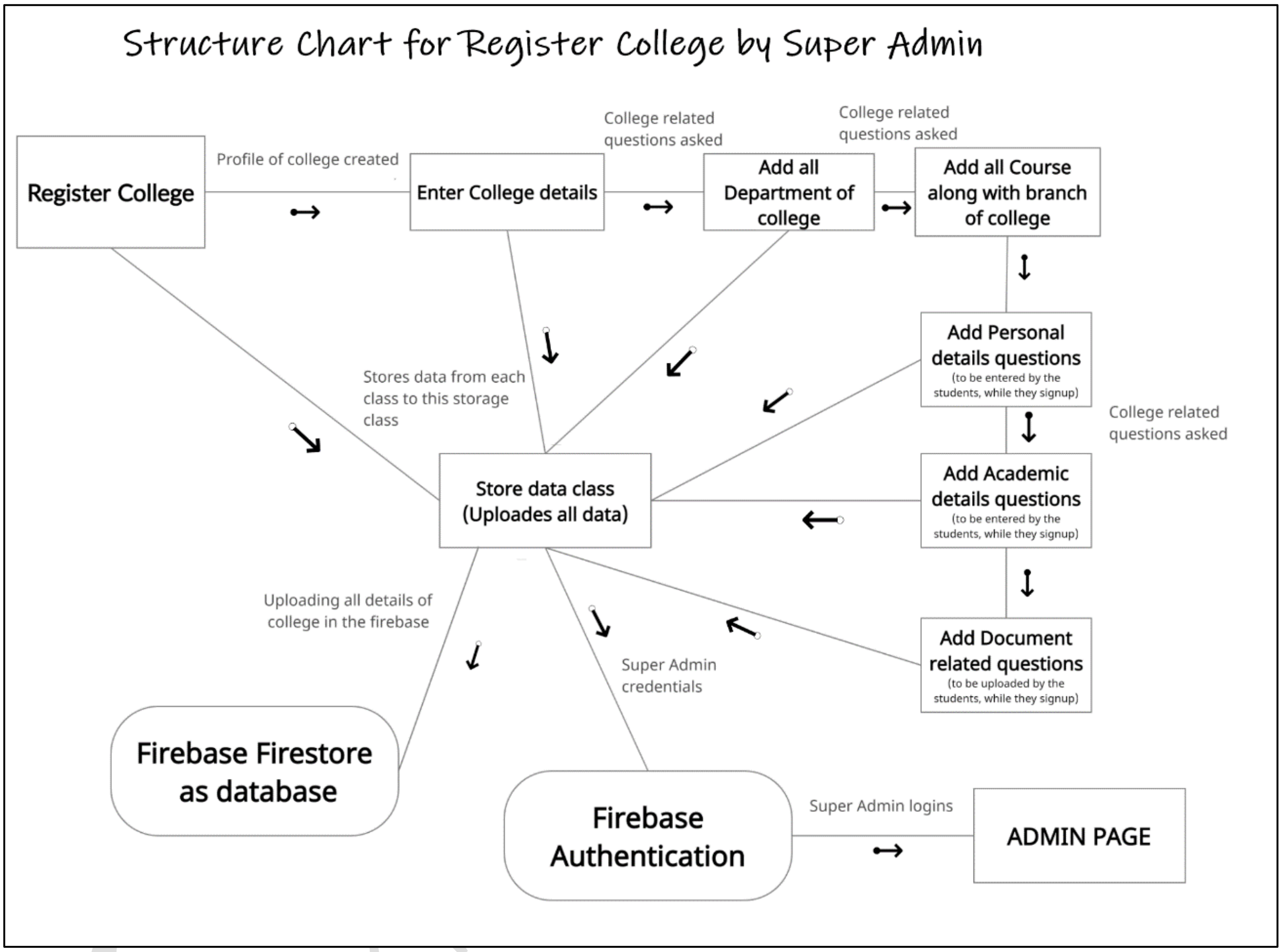
2.2 Structure and relationships

There are a lot of components joined together internally to provide various functionality in an application. The main functionality is that an admin can register their college for managing various types of details of its students and the companies that visit their campus, and can have an effective communication with them. Below is the system architecture of the app displaying the various different functionalities.



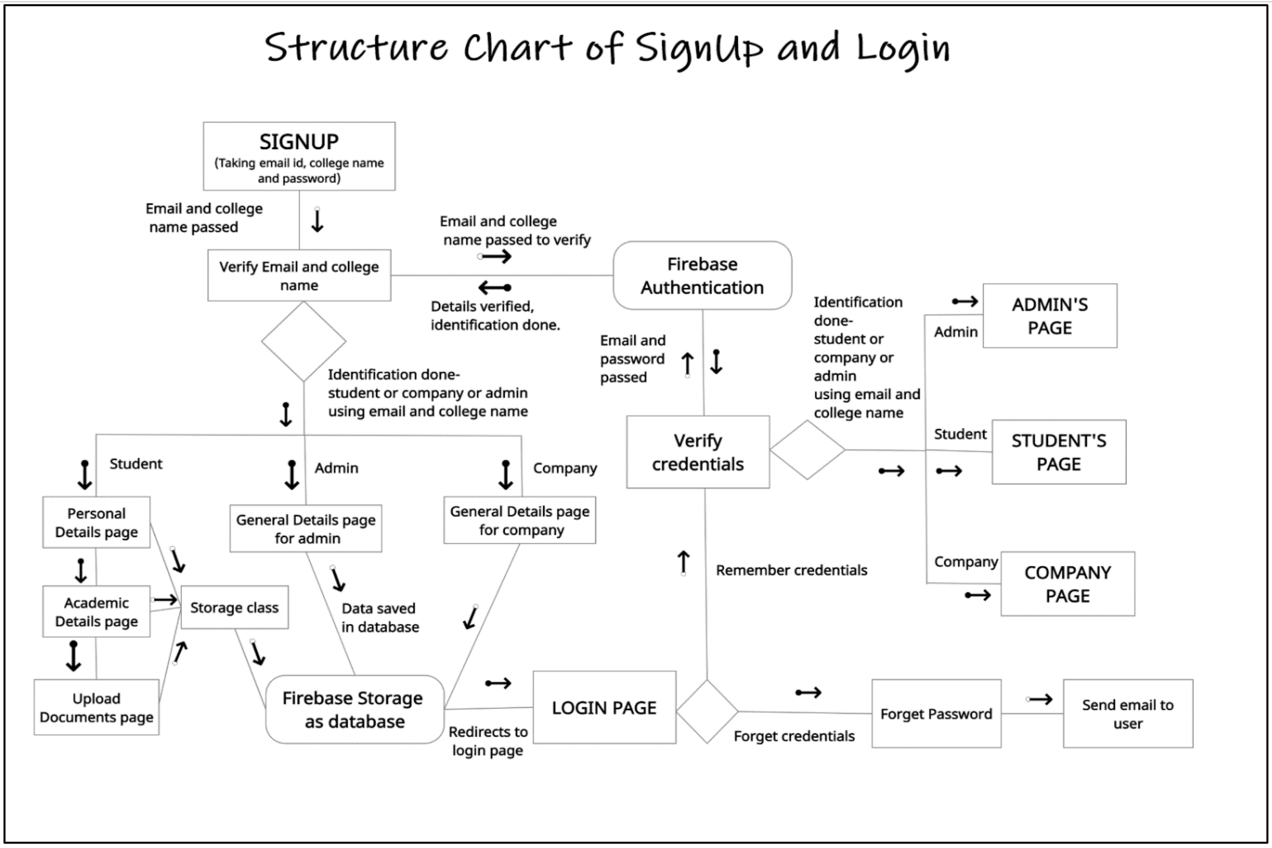
The following structure charts represents the transfer of data and flow between the various components of the app mentioned above. They represent the interrelationships and dependencies of various components with the modules of the app.

1. **College Register by Super admin** - At very initial stage, the super admins have to register their college on the app and provide various details about their college, and add questions/ detail entries they want from their students. Here, the following structure chart show the dependencies and flow how this section will work.

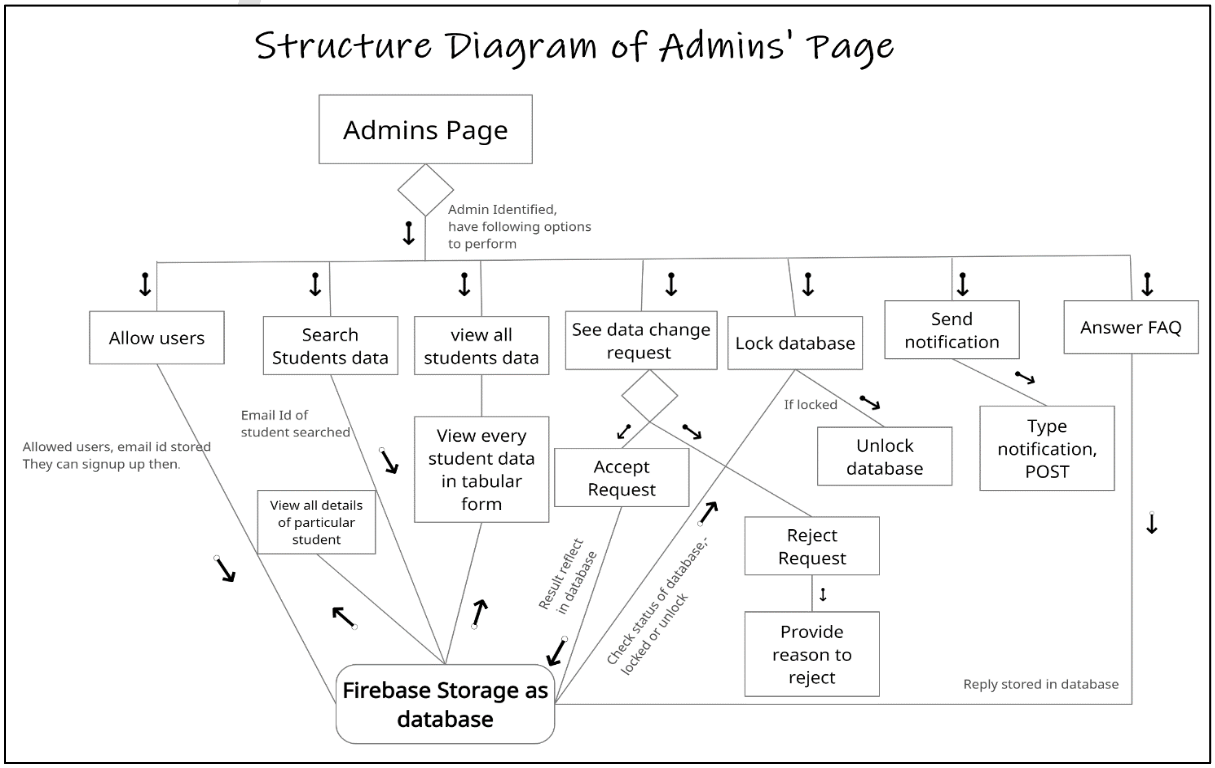

Structure Chart for Register College by Super Admin
Register College
Profile of college created
Enter College details
College related questions asked
Add all Department of college
College related questions asked
Add all Course along with branch of college
Add Personal details questions
(to be entered by the students, while they signup)
College related questions asked
Add Academic details questions
(to be entered by the students, while they signup)
Add Document related questions
(to be uploaded by the students, while they signup)
Stores data from each class to this storage class
Store data class (Uploades all data)
Uploading all details of college in the firebase
Super Admin credentials
Firebase Firestore as database
Firebase Authentication
Super Admin logins
ADMIN PAGE

Here, all the data from the various activities will get stored in the storage class, after which at last all the data is sent to firebase firestore to save. This super admin is then directly logged in to the admin page module.

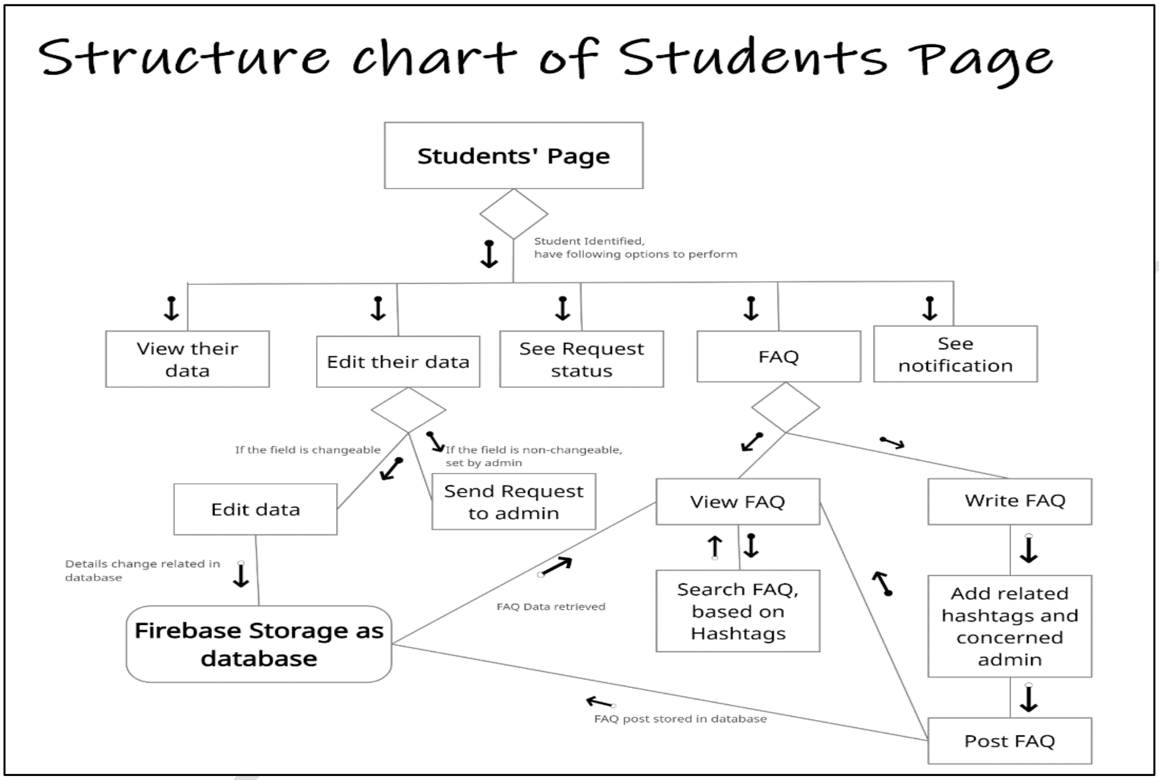
2. **Signup and Login** – After any particular college is registered, the member of the college- its other admin, students or the company (for that particular college) may signup. Every user has to have an allowed email id by any of the admin. Through that only they will be able to sign up (for consideration, our college email ids). After they enter the email ids and college name, authentication is done by firebase as to whether it is a student or admin or college, after which it directs it to its associated detailed signup pages. After successful signup, data gets stored in the firebase firestore and the users can login thereafter to their respective pages. Below is the detailed structure chart.


Structure Chart of SignUp and Login
SIGNUP
(Taking email id, college name and password)
Email and college name passed
Verify Email and college name
Email and college name passed to verify
Details verified, identification done.
Firebase Authentication
Identification done- student or company or admin using email and college name
Email and password passed
Identification done- student or company or admin using email and college name
Admin
ADMIN'S PAGE
Student
STUDENT'S PAGE
Company
COMPANY PAGE
Verify credentials
Student
Admin
Company
Personal Details page
General Details page for admin
General Details page for company
Academic Details page
Storage class
Data saved in database
Remember credentials
Upload Documents page
Firebase Storage as database
Redirects to login page
LOGIN PAGE
Forget credentials
Forget Password
Send email to user

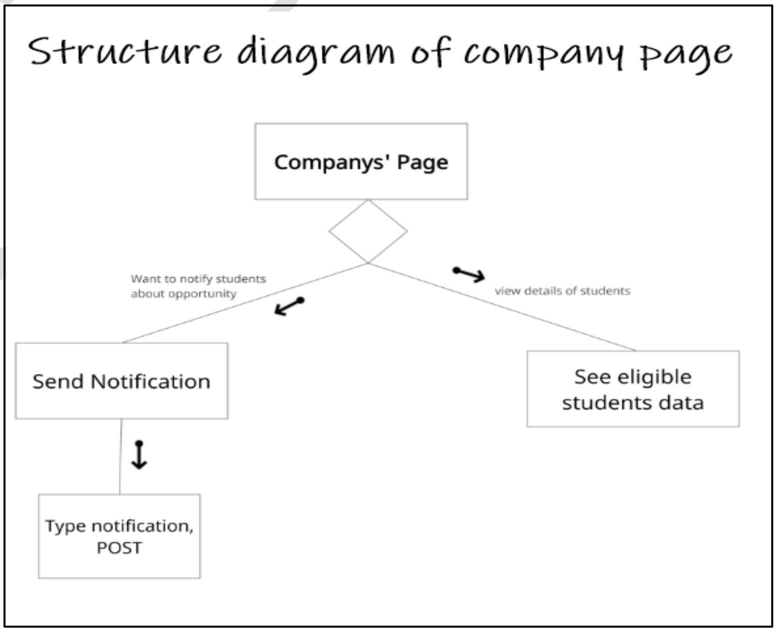
3. **Admin's Page -** The major component of the app is admin, whose has the following relations with the modules of the app. He can allow new user from college, search student, view all data, accept/ reject requests from students, lock/unlock database, answer FAQ and send notifications. Admin is supreme power and all changes done by him reflects directly in the database.


Structure Diagram of Admins' Page
Admins Page
Admin Identified, have following options to perform
Allow users
Search Students data
view all students data
See data change request
Lock database
Send notification
Answer FAQ
Allowed users, email id stored They can signup up then.
Email Id of student searched
View all details of particular student
View every student data in tabular form
Accept Request
If locked
Unlock database
Type notification, POST
Reject Request
Provide reason to reject
Result reflect in database
Check status of database,- locked or unlock
Reply stored in database
Firebase Storage as database

4. **Students page**- Another major component of the app is Student, for whose details basically this app works. He has the module relations to perform various activities like editing their own details, which changeable one and send request to admin for changing the non-changeable ones. The changed data will directly reflect in the database and will have no concern from the admin. Students can as well post and view FAQ and receive notifications from the company and the admin. Following is the structure chart to represent them.

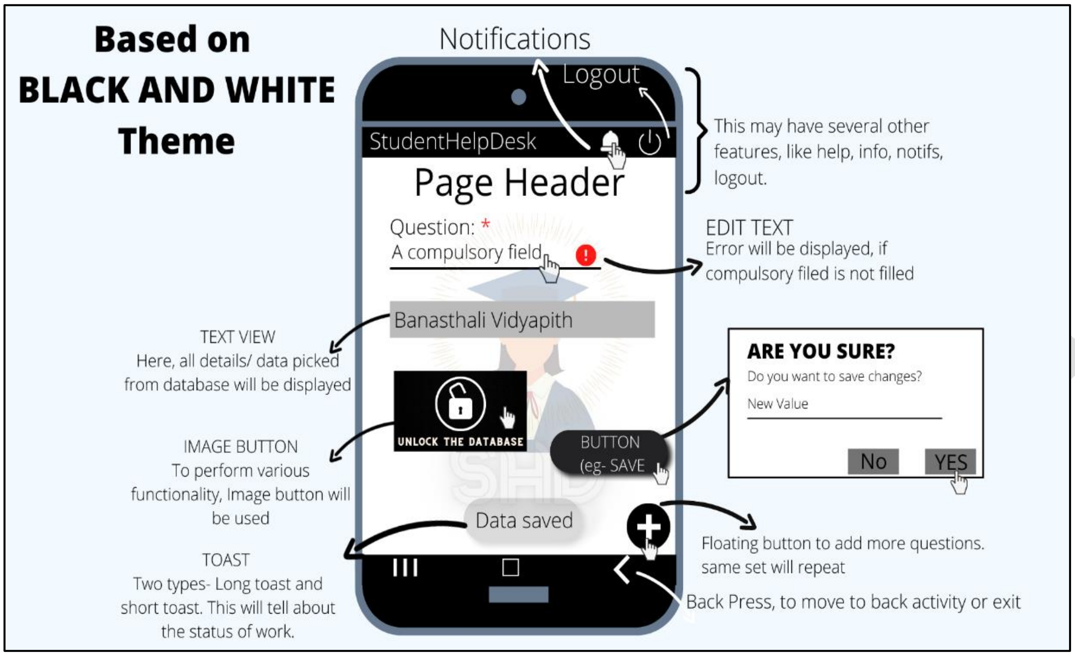

Structure chart of Students Page
Students' Page
Student Identified, have following options to perform
View their data
Edit their data
See Request status
FAQ
See notification
If the field is changeable
If the field is non-changeable, set by admin
Edit data
Send Request to admin
View FAQ
Write FAQ
Details change related in database
Search FAQ, based on Hashtags
Add related hashtags and concerned admin
FAQ Data retrieved
Firebase Storage as database
FAQ post stored in database
Post FAQ

5. **Company page**- Company is a minor component and has only a few functionality, which is of sending notifications to the students/admin and view data of eligible student. Following is the structure chart for it.


Structure diagram of company page
Companys' Page
Want to notify students about opportunity
view details of students
Send Notification
See eligible students data
Type notification, POST

2.3 User interface issues

Following image below includes sample screen images, GUI standards or product family style guides that are to be followed in to be app, screen layout constraints, standard buttons and functions (e.g., SAVE) that will appear on every screen, keyboard shortcuts, error message display standards, etc.



Based on BLACK AND WHITE Theme

This may have several other features, like help, info, notifs, logout.

EDIT TEXT
Error will be displayed, if compulsory filed is not filled

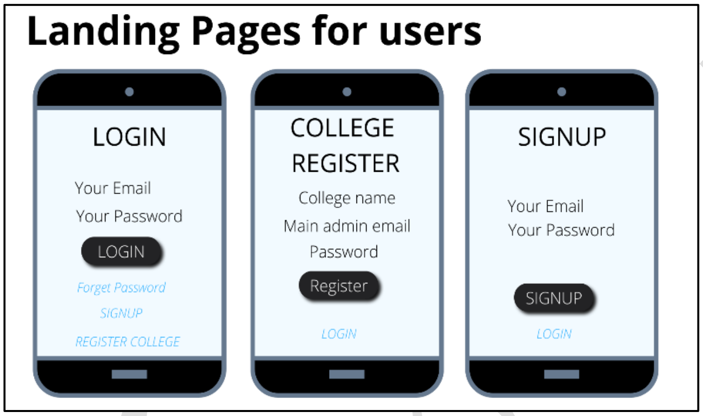
TEXT VIEW
Here, all details/ data picked from database will be displayed

IMAGE BUTTON
To perform various functionality, Image button will be used

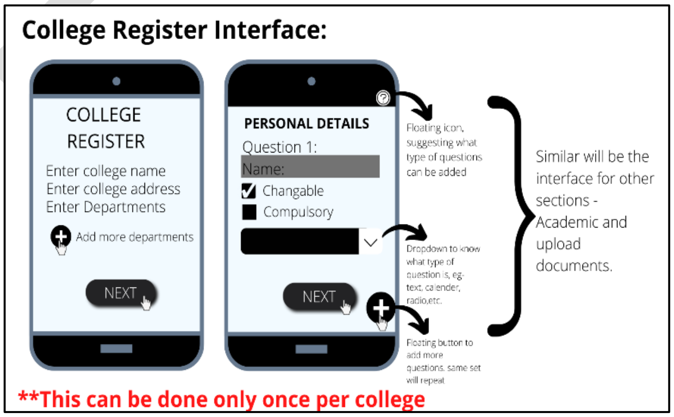
TOAST
Two types- Long toast and short toast. This will tell about the status of work.

Floating button to add more questions. same set will repeat

Back Press, to move to back activity or exit



Landing Pages for users



College Register Interface:

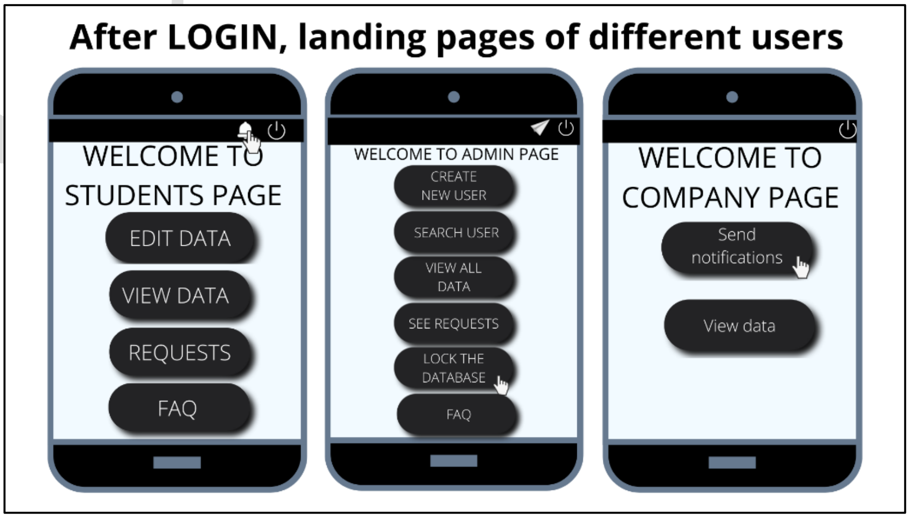
Floating icon, suggesting what type of questions can be added

Dropdown to know what type of question is, eg- text, calender, radio,etc.

Floating button to add more questions. same set will repeat

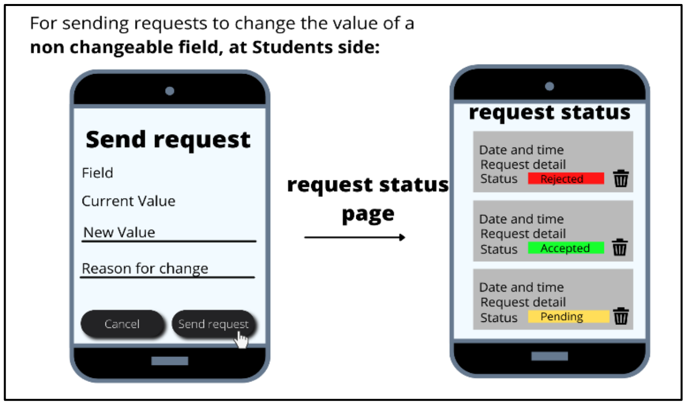
Similar will be the interface for other sections - Academic and upload documents.

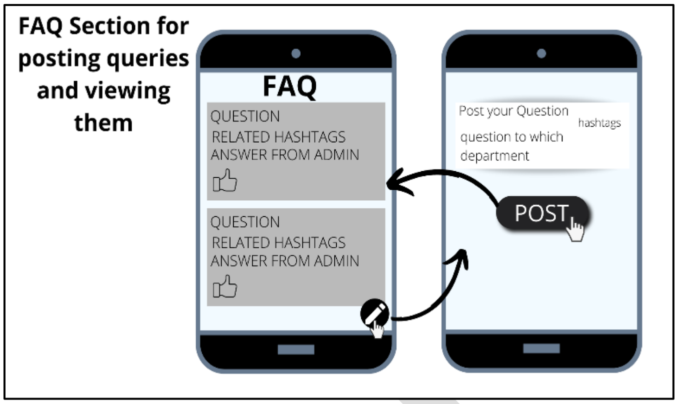
**This can be done only once per college



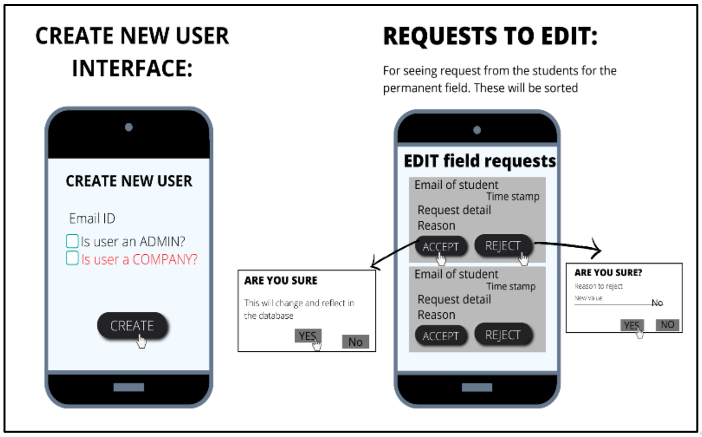
After LOGIN, landing pages of different users

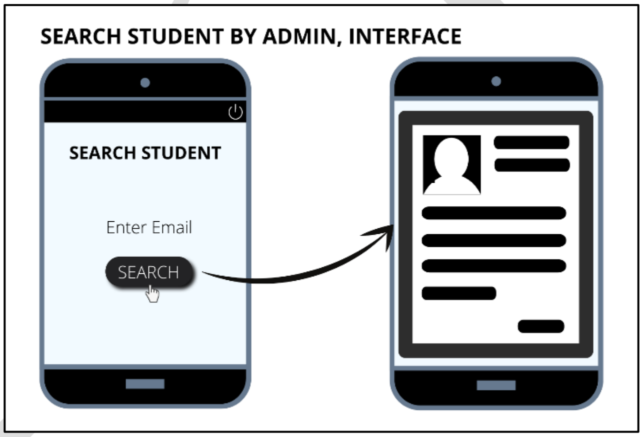
Student side, User Interface

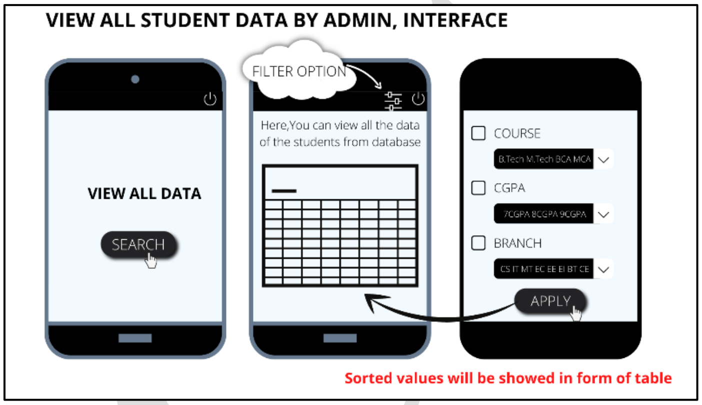


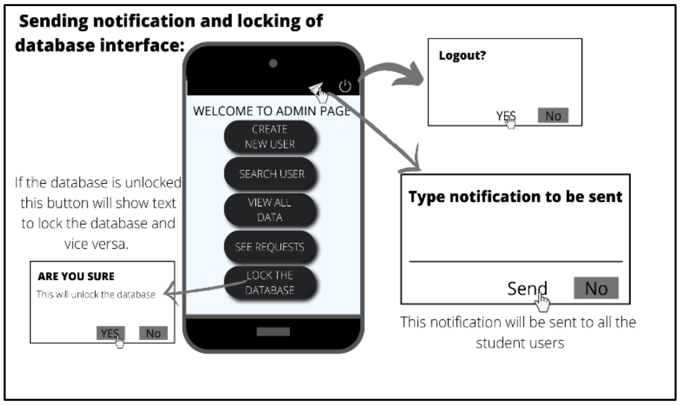


Admin side, User Interface

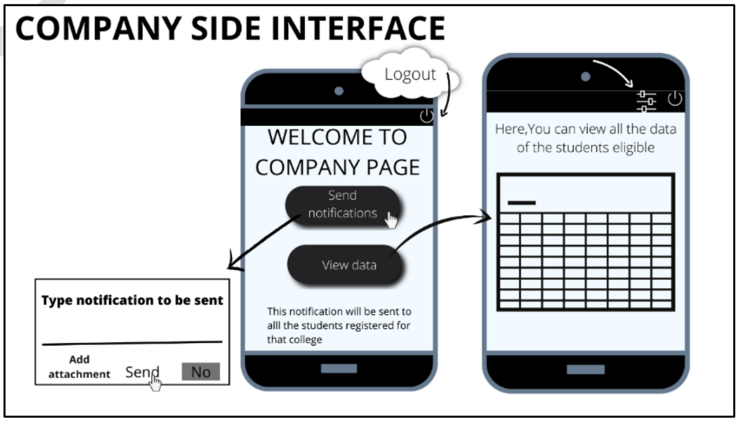





VIEW ALL STUDENT DATA BY ADMIN, INTERFACE
VIEW ALL DATA
SEARCH
FILTER OPTION
Here,You can view all the data of the students from database
□ COURSE
B.Tech M.Tech BCA MCA
□ CGPA
7CGPA 8CGPA 9CGPA
□ BRANCH
CS IT MT EC EE EI BT CE
APPLY
Sorted values will be showed in form of table

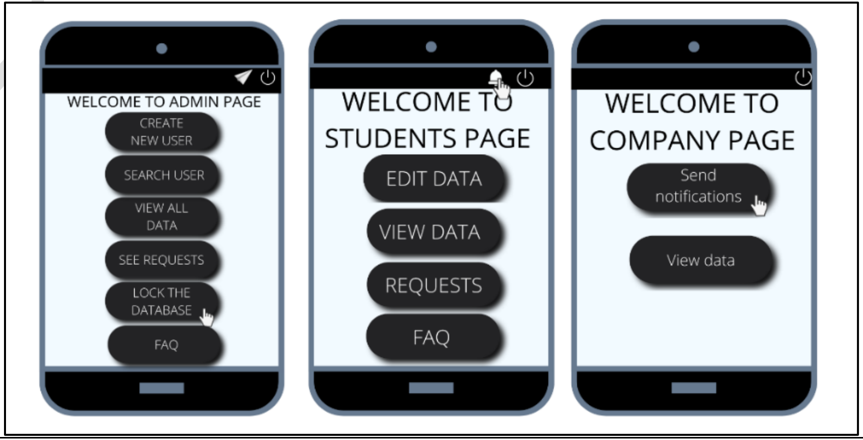

Sending notification and locking of database interface:
WELCOME TO ADMIN PAGE
CREATE NEW USER
SEARCH USER
VIEW ALL DATA
SEE REQUESTS
LOCK THE DATABASE
If the database is unlocked this button will show text to lock the database and vice versa.
ARE YOU SURE
This will unlock the database
YES
No
Logout?
YES
No
Type notification to be sent
Send
No
This notification will be sent to all the student users

Company side, User Interface


COMPANY SIDE INTERFACE
Logout
WELCOME TO COMPANY PAGE
Send notifications
View data
This notification will be sent to alll the students registered for that college
Type notification to be sent
Add attachment
Send
No
Here,You can view all the data of the students eligible

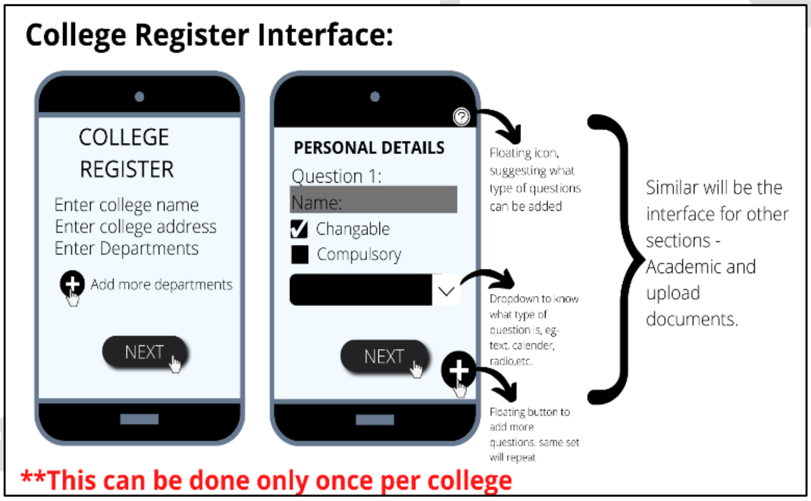
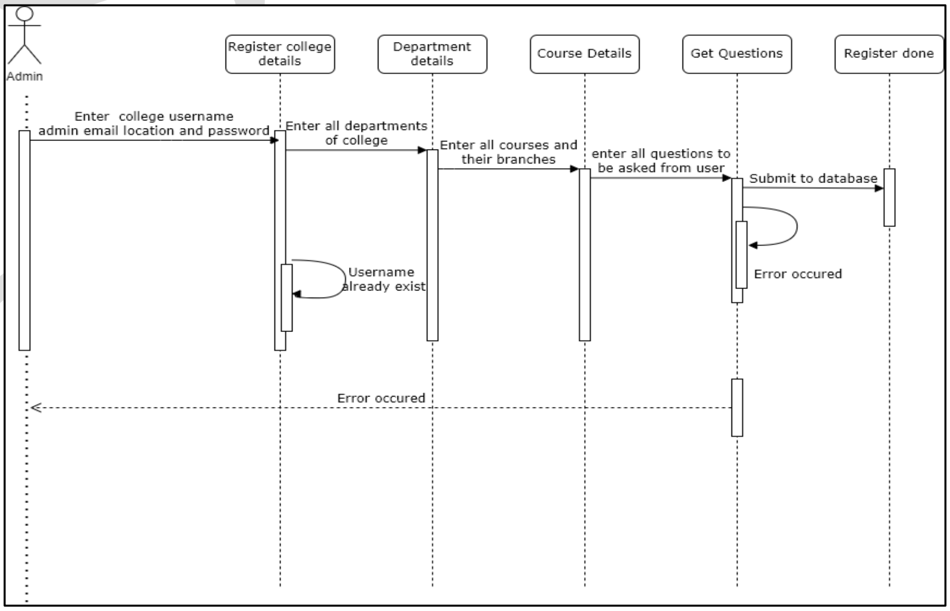
3. Detailed description of components

3.1 Component template description

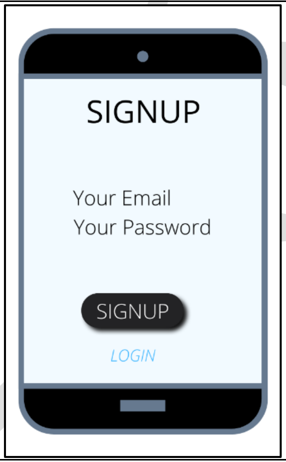
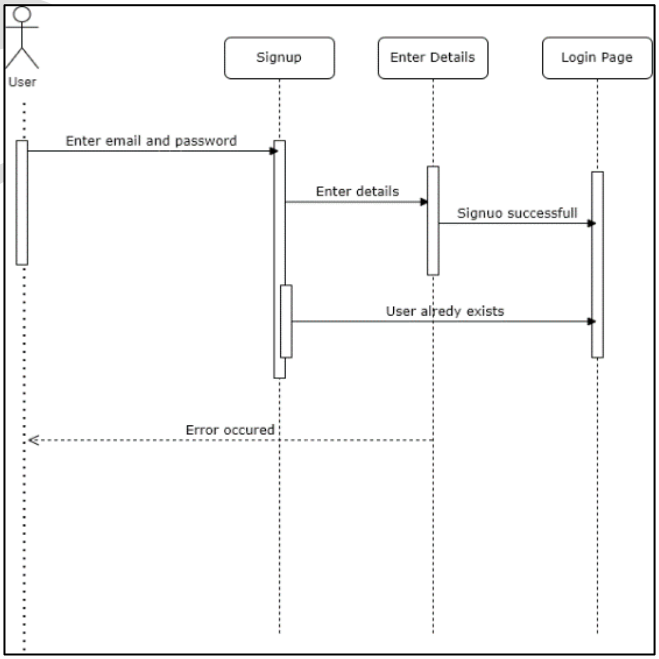
Identification	ADMIN	STUDENT	COMPANY
Type	Components	Components	Components
Purpose	They are the supreme controller of the app for their college. They have the authority to change, update and authorize other app components and modules.	They are the target users of the app. It is their data that is managed by the app.	They are the guests users of the app.
Function	• Register college • Login • Signup • Allow users • Answer FAQ questions • Accept/Reject change requests from students • Search student data • View all data from database • All data to company • Send notification • Logout	• Login • Signup • Send request • Change data • Ask FAQ • View FAQ • Logout	• Login • Signup • Send Notification • View Data • Logout
Subordinates	Modules for admin (Using android) which includes most features and connect to backend application via API	Modules for Student (Using android) which includes most features and connect to backend application via API	Modules for Company (Using android) which includes most features and connect to backend application via API
Dependencies	As soon as the college is registered, the super admin is created. Other admins can signup and login. They can create other users, view students data, accept/reject requests from students, answer FAQs, and allow data to company.	As soon as the admin creates student user, they can login, signup (feed their information), view their data, see notifications send by admin or company. They can also ask admin FAQs and request them to change their non-changeable data.	As soon as the admin creates company user, they can login, signup, view student data(allowed by admin) and send notification to students.
Interfaces	 WELCOME TO ADMIN PAGE CREATE NEW USER SEARCH USER VIEW ALL DATA SEE REQUESTS LOCK THE DATABASE FAQ WELCOME TO STUDENTS PAGE EDIT DATA VIEW DATA REQUESTS FAQ WELCOME TO COMPANY PAGE Send notifications View data		
Resources	Firebase firestore, firebase storage, firebase authentication, Android Studio		

Processing	**ADMIN** **STEP 1.** Open SHD **STEP 2.** If user exists go to step 8 otherwise go to step 3 **STEP 3.** If college exists go to step 6 otherwise go to step 4 **STEP 4.** Register college. **STEP 5.** Go to step 8 **STEP 6.** Sign up as admin **STEP 7.** Go to step 8 **STEP 8.** Login **STEP 9.** If user wants to view data go to step 10 otherwise step 11 **STEP 10.** View all data of various students in tabular format **STEP 11.** If user wants to send notification go to step 12 otherwise step 13 **STEP 12.** Enter notification details and send **STEP 13.** If user wants to see request go to step 14 otherwise step 18 **STEP 14.** If user wants to accept any available request go to step 15 else step 18 **STEP 15.** Accept request and go to step 14 **STEP 16.** If user wants to reject any request got to step 17 otherwise step 18 **STEP 17.** Give reason for rejection of the request and go to step 14 **STEP 18.** If user want to answer FAQ go to step 19 otherwise step 20 **STEP 19.** Give answer and go to step 20 **STEP 20.** If user want to change state of database lock go to step 21 otherwise step 24 **STEP 21.** If database is locked go to step 22 otherwise step 23 **STEP 22.** Unlock database and go to step 24 **STEP 23.** Lock database and go to step 24 **STEP 24**. If user want to all other users go to step 24 otherwise step 25 **STEP 25.** Enter user's basic info and Create and go to step 25 **STEP 26.** If User want to logout go to step 26 otherwise step 9 **STEP 27.** If user want to exit app go to step 28 otherwise step 2 **STEP 28.** Exit	**STUDENT** **STEP 1.** Open SHD **STEP 2.** If user exists go to step 3 otherwise go to step 4 **STEP 3.** Login and go to step 5 **STEP 4.** Sign Up and go to step 3 **STEP 5.** if user want to view data go to step 5 otherwise step 7 **STEP 6.** View data and go to step 7 **STEP 7.** if user wants to send request to change data go to step 8 otherwise step 9 **STEP 8.** Send Request and go to step 9 **STEP 9.** if user wants to ask FAQ go to step 10 otherwise step 11 **STEP 10.** Ask FAQ , add department and hashtags and go to step 11 **STEP 11.** if user wants to view FAQ go to step 12 otherwise step 13 **STEP 12.** View FAQ and go to step 13 **STEP 13.** if user wants to see notification go to step 14 otherwise step 15 **STEP 14**. See Notification and go to step 15 **STEP 15.** if user wants to logout go to step16 otherwise step 5 **STEP 16.** if user wants to exit app go to step 17 otherwise step 2 **STEP 17.** Exit	**COMPANY** **STEP 1.** Open SHD **STEP 2.** If user exists go to step 5 otherwise go to step 3 **STEP 3.** Sign up as company and go to step 5 **STEP 4:** Enter company details **STEP 5.** Login **STEP 6.** If user wants to view data go to step 7 otherwise step 9 **STEP 7.** If company has data access go to step 8 otherwise step 9 **STEP 8.** See data and go to step 9 **STEP 9.** If user wants to send notification go to step 10 otherwise step 12 **STEP 10.** Enter notification details and send **STEP 11**. If User want to logout go to step 12 otherwise step 12 **STEP 12.** If user want to exit app go to step 13 otherwise step 6 **STEP 13.** Exit
Data	Some of the data that will be stored are—email id, name, department, phone number	Some of the data that will be stored are—email id, personal details, academic details, some documents, requests of edit data sent to admin and FAQs.	Some of the data that will be stored are—email id, name, department, phone number, company location.

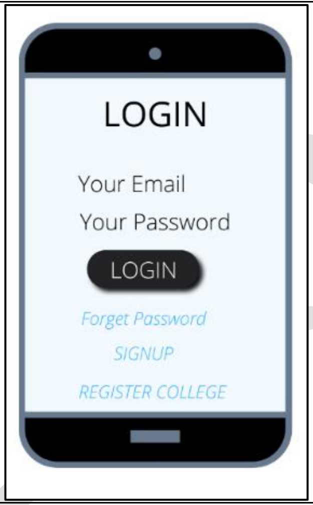
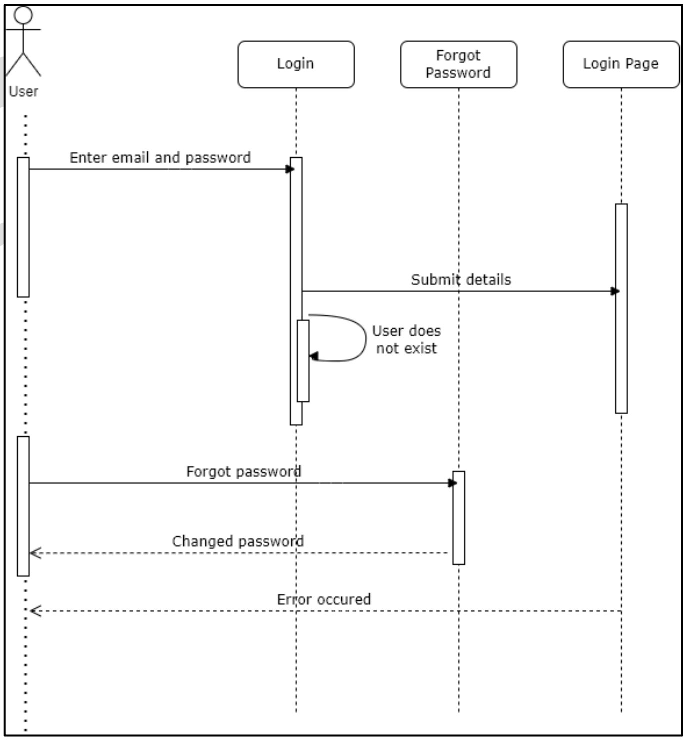
3.2 Register College

Identification	Register college
Type	Module
Purpose	This is a module which is executed to register a new college into database and get all information about the college and superadmin.
Function	• Get College's location, email, and other details • Get department names of college • Get courses and branches in it • Get all questions to be asked from students • Get super admin's details
Subordinates	NA
Dependencies	It depends on nothing because it is the first step to be done by any college.
Interfaces	 College Register Interface: COLLEGE REGISTER Enter college name Enter college address Enter Departments Add more departments NEXT PERSONAL DETAILS Question 1: Name: ☑ Changable ■ Compulsory NEXT Floating icon, suggesting what type of questions can be added Dropdown to know what type of question is, eg- text, calender, radio,etc. Floating button to add more questions. same set will repeat Similar will be the interface for other sections - Academic and upload documents. **This can be done only once per college
Resources	Firebase firestore, firebase authentication, Android Studio
Processing	 Admin Register college details Department details Course Details Get Questions Register done Enter college username admin email location and password Enter all departments of college Enter all courses and their branches enter all questions to be asked from user Submit to database Username already exist Error occured Error occured
Data	College Name, College Location, Admin Details, Branches, Courses, Departments, Questions to be asked.

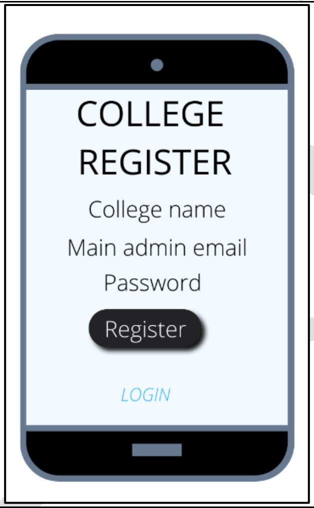
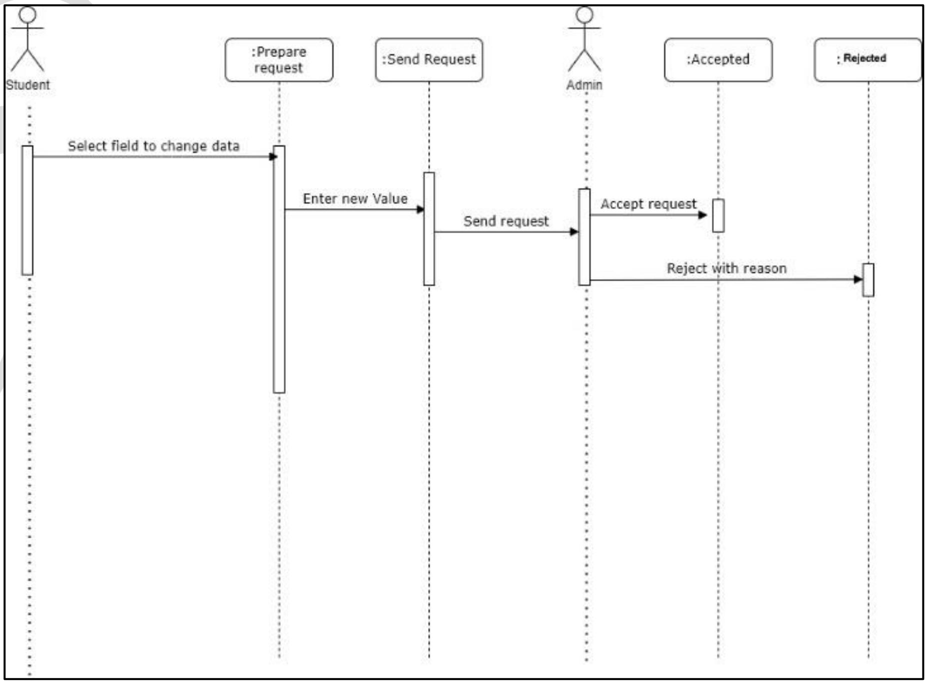
3.3 Signup

Identification	Signup
Type	Module
Purpose	This is module which authorize the user to use the app and provide their detail to database.
Function	• Get username and password • Get personal details • Get academic details • Get uploads
Subordinates	NA
Dependencies	Only if college has been registered and admin has allowed the user, this step will complete successfully.
Interfaces	
Resources	Firebase Authorization , Firebase FireStore, Firestore Storage
Processing	 User Signup Enter Details Login Page Enter email and password Enter details Signuo successfull User alredy exists Error occured
Data	Email, Password, College, Branch, Year, Personal Details, Academic Details, Uploads

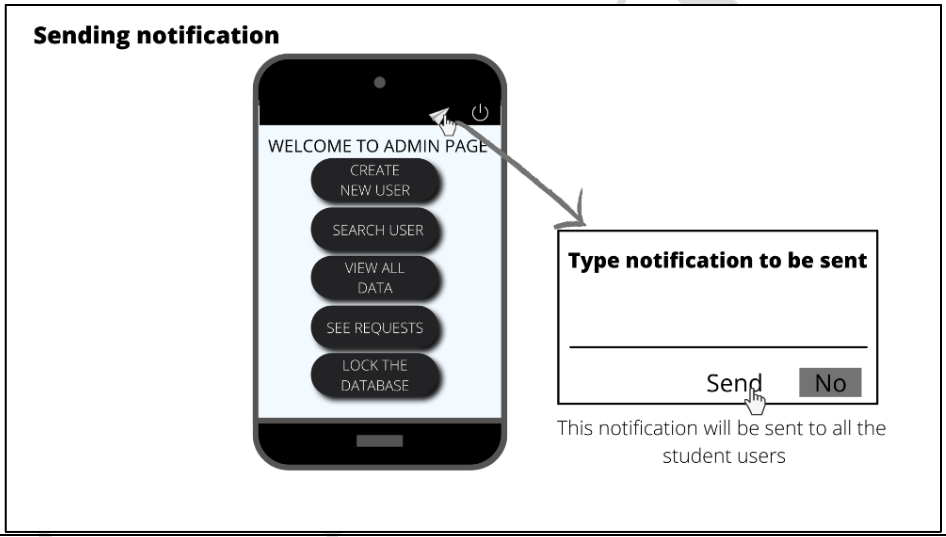
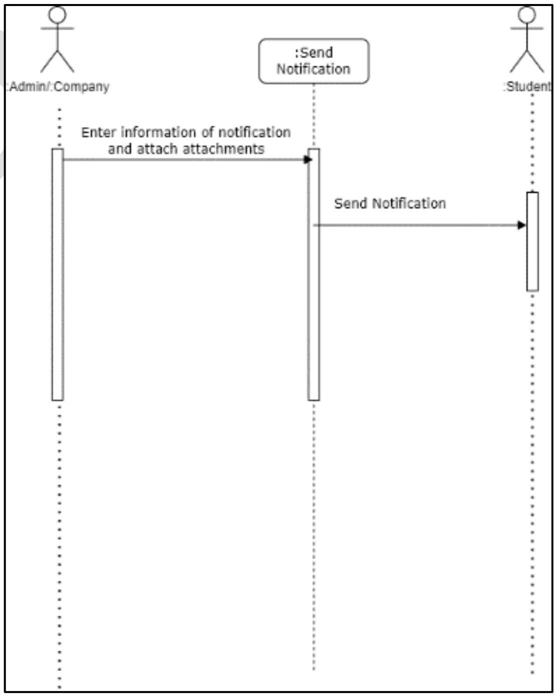
3.4 Login

Identification	Login
Type	Module
Purpose	This is the module which allow the user to enter the app.
Function	• Get Email and Password • Check if credentials are correct • Login
Subordinates	NA
Dependencies	Login will only work if the user has already signed up and have entered correct email and password
Interfaces	 LOGIN Your Email Your Password LOGIN Forget Password SIGNUP REGISTER COLLEGE
Resources	Firebase Authorization
Processing	 User Login Forgot Password Login Page Enter email and password Submit details User does not exist Forgot password Changed password Error occured
Data	Email, Password

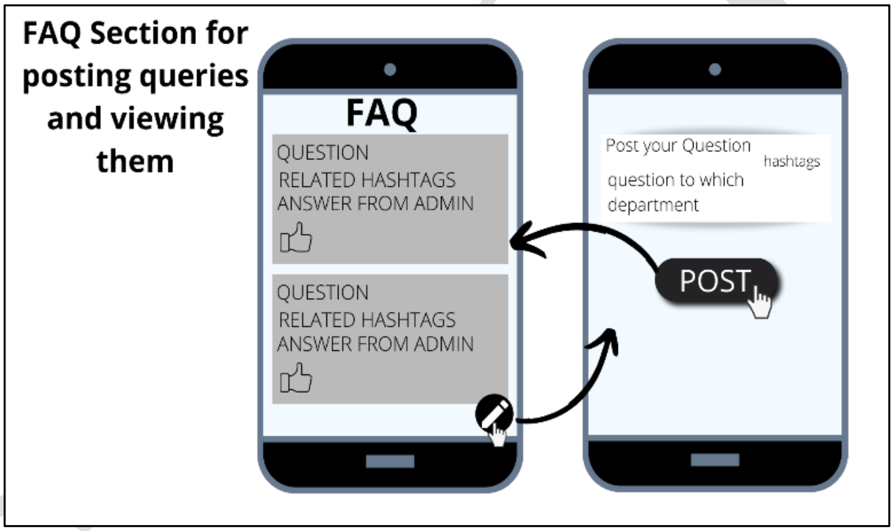
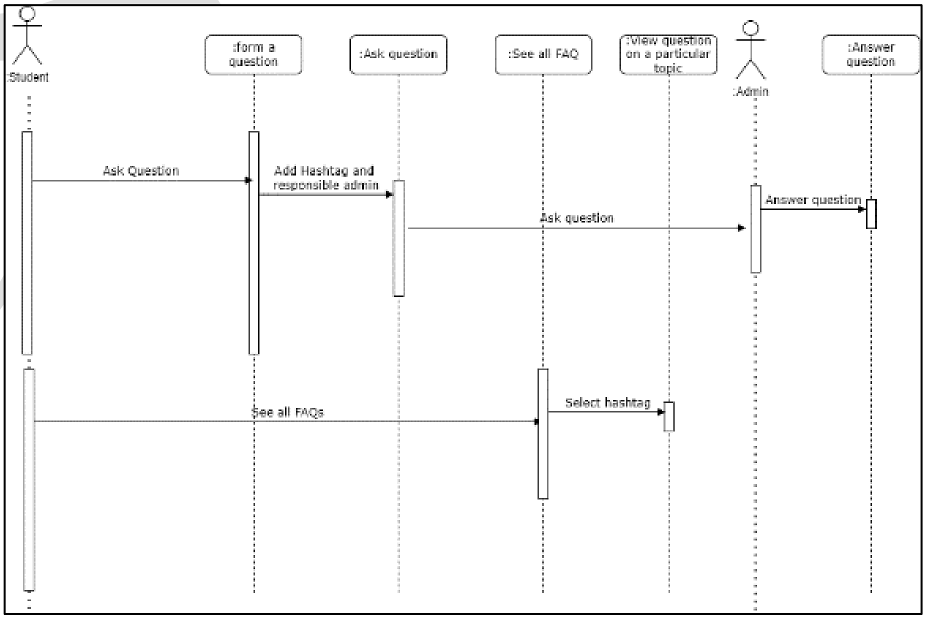
3.5 Request to change data

Identification	Request to change data
Type	Module
Purpose	This module gives student the power to request admin to change the data which is otherwise set as non-changeable by admins.
Function	• Get details to change • Send request
Subordinates	NA
Dependencies	This module will only work if the data field is set non-changeable by admin and if database is unlocked
Interfaces	 COLLEGE REGISTER College name Main admin email Password Register LOGIN
Resources	Firebase Firestore
Processing	 Student :Prepare request :Send Request Admin :Accepted : Rejected Select field to change data Enter new Value Send request Accept request Reject with reason
Data	Questions to change, New Value, TimeStamp

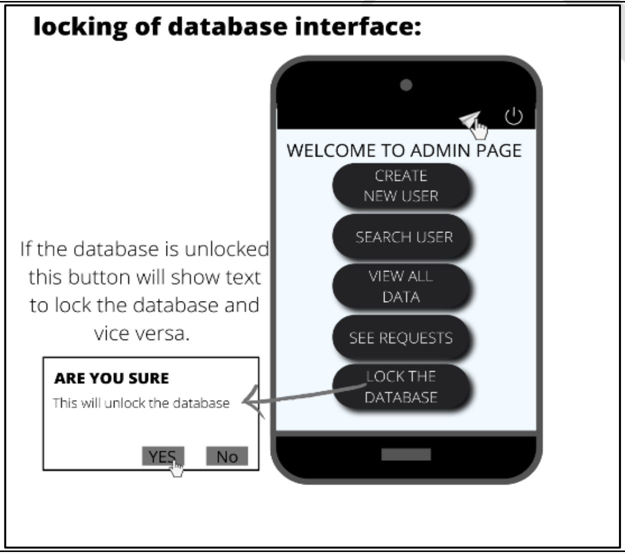
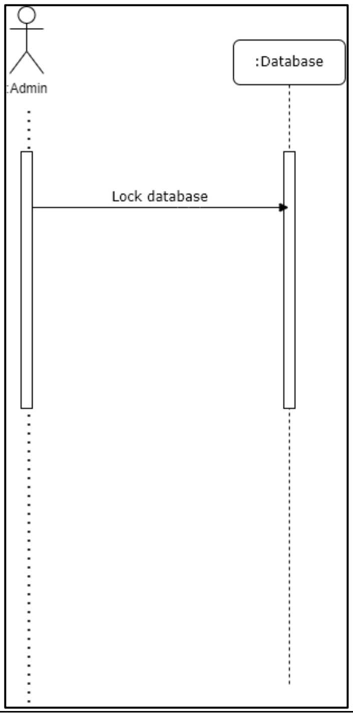
3.6 Notifications

Identification	Notification
Type	Module
Purpose	This module can be executed by both company and admin, to send notification about any updates that student should know.
Function	• Send notification
Subordinates	NA
Dependencies	NA
Interfaces	 Sending notification WELCOME TO ADMIN PAGE CREATE NEW USER SEARCH USER VIEW ALL DATA SEE REQUESTS LOCK THE DATABASE Type notification to be sent Send No This notification will be sent to all the student users
Resources	Firebase firestore
Processing	 :Admin/:Company :Send Notification :Student Enter information of notification and attach attachments Send Notification
Data	Notification Header, Notification detail, Attachment, Sent By, Sent Time

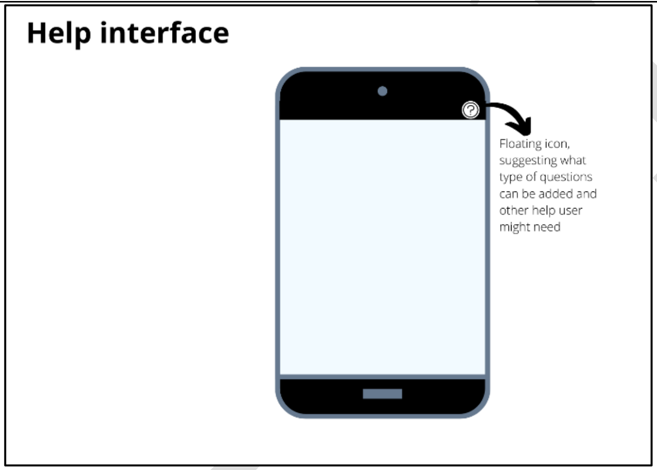
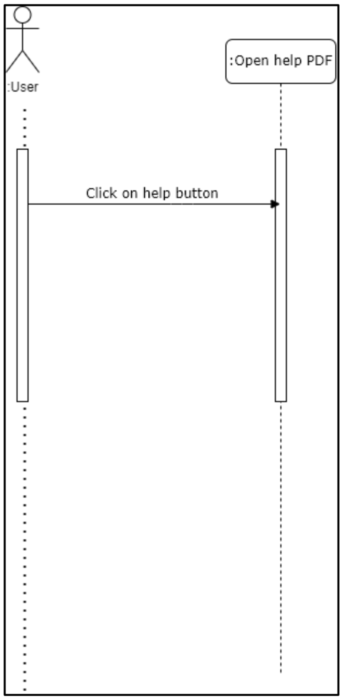
3.7 FAQ

Identification	FAQ
Type	Module
Purpose	This module gives user the ability to ask their doubts directly from admins of their desired documents.
Function	• See all FAQ · • See Hashtags • Ask Questions
Subordinates	NA
Dependencies	NA
Interfaces	 FAQ Section for posting queries and viewing them FAQ QUESTION RELATED HASHTAGS ANSWER FROM ADMIN QUESTION RELATED HASHTAGS ANSWER FROM ADMIN Post your Question hashtags question to which department POST
Resources	Firebase Firestore
Processing	 :Student :form a question :Ask question :See all FAQ :View question on a particular topic :Admin :Answer question Ask Question Add Hashtag and responsible admin Ask question Answer question See all FAQs Select hashtag
Data	Question details, Hastags to add, Department to add

3.8 Lock Database

Identification	Lock database
Type	module
Purpose	Sometimes the college will have to send data to company, and they would not want students to change data at that time. It is at such moments when the module will be used. Admin will lock the database and thus no student would be able to make changes in their data.
Function	• Check current database status • Change status
Subordinates	NA
Dependencies	Admin can only lock the database of his department only.
Interfaces	 **locking of database interface:** If the database is unlocked this button will show text to lock the database and vice versa. WELCOME TO ADMIN PAGE CREATE NEW USER SEARCH USER VIEW ALL DATA SEE REQUESTS LOCK THE DATABASE ARE YOU SURE This will unlock the database YES No
Resources	Firestore firestore
Processing	
Data	Database status

3.9 Help

Identification	Help
Type	Module
Purpose	While using the app user might not understand the interface or features, thus to help them understand it this module is used.
Function	Redirect to help document Show help toast
Subordinates	NA
Dependencies	NA
Interfaces	 Help interface Floating icon, suggesting what type of questions can be added and other help user might need
Resources	PDF viewer, Android API
Processing	
Data	NA

4.0 Reuse and relationships to other products

The environment during placements and internship interviews is one of the most hectic environments that a student face in his college life. This is mainly because of the fact that the current existing methods used by college are obsolete in terms of data collection and correction. Students have to mail the placement cell every time they have to change their data and placement cell have to release a form every time they need data. Although currently online forms are released by the college to get data , these forms often leave no scope of correction in real time and in case any mistake is made , student have to go back to the old method of mailing the placement team. But, SHD brings both student and admin of the colleges to one single platform, releasing the extra burden from both the sides. It also not only allows the students to upload their data but also admin can see it all the time. Admin can also set some data fields as non-editable thus making students to take his permission before changing the data. The editable fields can be changed anytime, given the database is unlocked.

SHD also provides the following advantages:

- Admin need not to send data to company separately. They can allow the desired company to see data from their database provided company has registered.
- Admin can send important notifications to student directly. These notifications will be sent to user in real time thus helping them to not miss any update. Send notification feature is also available with the companies, helping them to communicate with students directly eliminating the need of middleman.
- Admin can lock the database anytime necessary. They can do it to ensure consistency in the database before sending data to company.
- Students can ask their doubts to various department of the colleges directly using the FAQs. They can directly tag the related department and hashtags. The admins from related department can answer the questions. These questions will be categorised with hashtags, so all questions related to a particular hashtags will be available for student to refer.

5.0 Design decisions and tradeoffs

StudentHelpDesk(SHD) uses technologies and provides an efficient way for students to send their data to the placement cell, Admins to receive the data from the students and provide it to the companies required for placement/internship purpose. SHD helps to solve the problems faced by the placement cell by collecting data from the students and provide it to companies for placements. It is necessary to have a mobile phone and a good internet connection to access the app which is easily available to everyone nowadays.

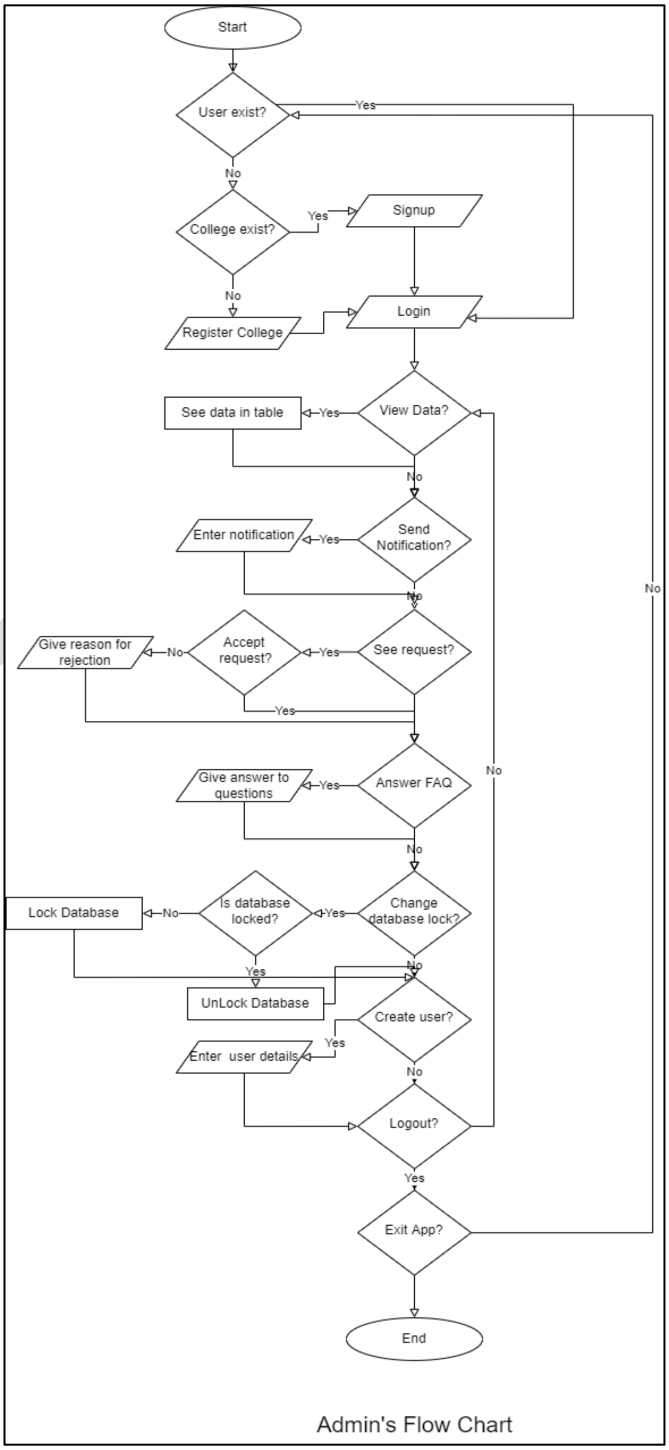
This app uses the firebase database and stores the data dynamically. The firebase database allows us to use the real time database which helps our app to be dynamic and more realistic. It is time effective as well as cost effective. Since our app works only with verified users and admins therefore it offers security of data of students as well as companies.

6.0 Pseudocode for components

6.1 Pseudocode for Admin

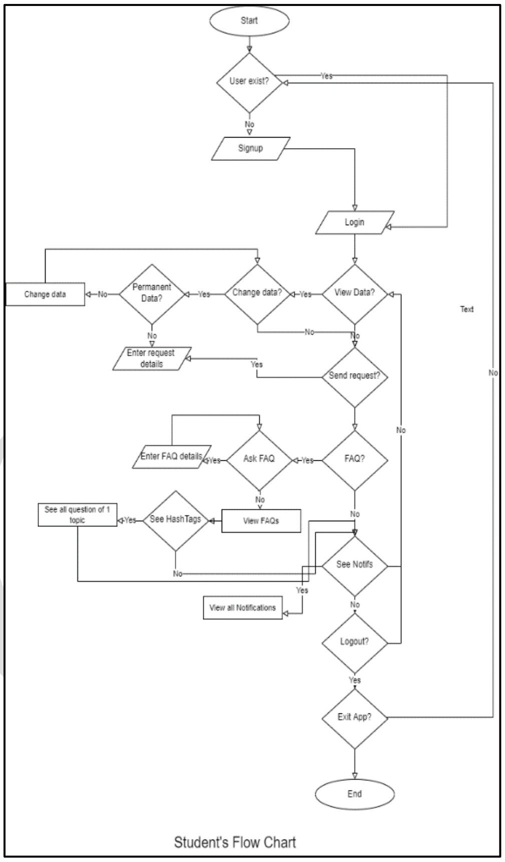
Admin

```
STEP 1. Open SHD
STEP 2. If user exists go to step 8 otherwise go to step 3
STEP 3. If college exists go to step 6 otherwise go to step 4
STEP 4. Register college.
STEP 5. Go to step 8
STEP 6. Sign up as admin
STEP 7. Go to step 8
STEP 8. Login
STEP 9. If user wants to view data go to step 10 otherwise step 11
STEP 10. View all data of various students in tabular format
STEP 11. If user wants to send notification go to step 12 otherwise step 13
STEP 12. Enter notification details and send
STEP 13. If user wants to see request go to step 14 otherwise step 18
STEP 14. If user wants to accept any available request go to step 15 else step 18
STEP 15. Accept request and go to step 14
STEP 16. If user wants to reject any request got to step 17 otherwise step 18
STEP 17. Give reason for rejection of the request and go to step 14
STEP 18. If user want to answer FAQ go to step 19 otherwise step 20
STEP 19. Give answer and go to step 20
STEP 20. If user want to change state of database lock go to step 21 otherwise step 24
STEP 21. If database is locked go to step 22 otherwise step 23
STEP 22. Unlock database and go to step 24
STEP 23. Lock database and go to step 24
STEP 24. If user want to all other users go to step 24 otherwise step 25
STEP 25. Enter user's basic info and Create and go to step 25
STEP 26. If User want to logout go to step 26 otherwise step 9
STEP 27. If user want to exit app go to step 28 otherwise step 2
STEP 28. Exit
```



Admin's Flow Chart

6.2 Pseudocode for Student

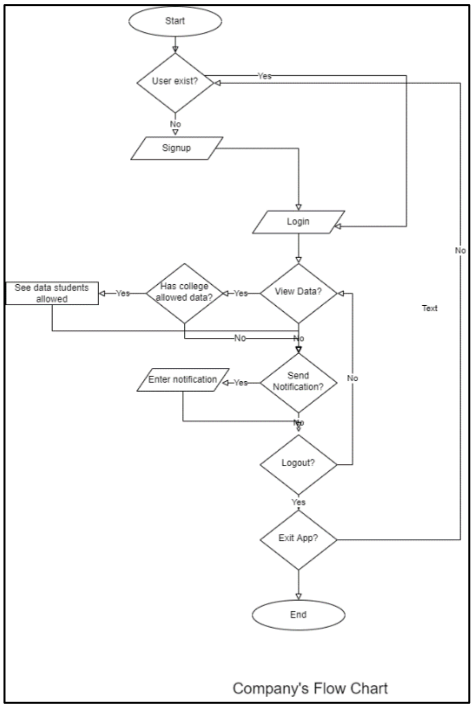


Student's Flow Chart

Student

```
STEP 1. Open SHD
STEP 2. If user exists go to step 3 otherwise go to step 4
STEP 3. Login and go to step 5
STEP 4. Sign Up and go to step 3
STEP 5. if user want to view data go to step  5 otherwise step 7
STEP 6. View data and go to step 7
STEP 7. if user wants to send request to change data go to step 8 otherwise step 9
STEP 8. Send Request and go to step 9
STEP 9. if user wants to ask FAQ go to step 10  otherwise  step 11
STEP 10. Ask FAQ , add department and hashtags and go to step 11
STEP 11. if user wants to view FAQ go to step 12 otherwise step 13
STEP 12. View FAQ and go to step 13
STEP 13. if user wants to see notification go to step 14 otherwise step 15
STEP 14. See Notification and go to step 15
STEP 15. if user wants to logout go to step16 otherwise step 5
STEP 16. if user wants to exit app go to step 17 otherwise step 2
STEP 17. Exit
```

6.3 Pseudocode for Company



Company's Flow Chart

Company

```
STEP 1. Open SHD
STEP 2. If user exists go to step 5 otherwise go to step 3
STEP 3. Sign up as company and go to step 5
STEP 4: Enter company details
STEP 5. Login
STEP 6. If user wants to view data go to step 7 otherwise step 9
STEP 7. If company has data access go to step 8 otherwise step 9
STEP 8. See data and go to step 9
STEP 9. If user wants to send notification go to step 10 otherwise step 12
STEP 10. Enter notification details and send
STEP 11. If User want to logout go to step 12 otherwise step 12
STEP 12. If user want to exit app go to step 13 otherwise step 6
STEP 13. Exit
```

7.0 Appendices (if any)

Not Applicable