

COLLEGE CONNECT

(College Apps Bundle)



**Cluster Innovation Centre
University of Delhi**

**AMIT | ASHUTOSH | SACHIN
SHUBHAM | TUSHAR**

AUG-NOV 2021

Project submitted for the paper of:
Software Project Management

CERTIFICATE OF ORIGINALITY

The work embodied in this report entitled “**College Connect (College Apps Bundle)**” has been carried out by **Amit, Ashutosh, Sachin, Shubham** and **Tushar** for the paper of “**Software Project Management**”. We declare that the work and language included in this project report is free from any kind of plagiarism.

The work submitted is original and has not been submitted earlier to any institute or university for the award of any degree or diploma.

Amit
(11807)

Ashutosh
(11811)

Sachin
(11831)

Shubham
(11838)

Tushar
(11840)

ACKNOWLEDGEMENT

With a deep sense of gratitude, we express our dearest indebtedness to **Prof. Anjani Kumar Verma** for his support throughout the duration of our project. We would like to thank him for giving us the opportunity to do this wonderful project. His learned advice and constant encouragement has helped us complete this project. It is a privilege for us to be his students. Without his encouragement and guidance this project would not have materialized.

We are also thankful to our friends and family who helped us with their valuable suggestions and guidance that have been helpful in various phrases of the completion of the project.

ABSTRACT

The objective of the project was to apply the theoretical as well as practical knowledge of Software Engineering to build android applications. The main purpose is to make mobile apps by first analyzing various requirements using software specifications and constructing a software specifications document and design document.

The prime objective is to build College Connect which is a set of four android applications for students, teachers, admins and canteen vendors built on android framework and hosted on Firebase which offers backend as a service.

TABLE OF CONTENTS

1. Introduction & Objective

- 1.1. Problem Statement
- 1.2. Problem Structure
- 1.3. Features & Goals
- 1.4. Intended Audience

2. System Analysis

- 2.1. System Requirements
- 2.2. Other Non-functional Requirements
- 2.3. Project Timeline
- 2.4. Survey

3. Design Document

- 3.1. Context Diagram (Data Flow Diagram Level-0)
- 3.2. Data Flow Diagram Level-1
- 3.3. Data Flow Diagram Level-2
- 3.4. Data Flow Diagram Level-3
- 3.5. State Transition Diagram
- 3.6. Architectural Diagram
- 3.7. Entity Relationship Diagram

4. Functional & Design Requirements

5. Appendix

- 5.1. Test Document**
- 5.2. Screenshots**

References

Abbreviations and terms

Android	An open source platform designed for mobile devices
OS	Operating system
SDK	Software development kit
JRE	Java Runtime Environment
IDE	Integrated development environment
GUI	Graphical user interface
Android Emulator	Container for running Android OS along with applications
XML	Extensible Markup Language
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
RAM	Random Access Memory
IT	Information Technology
GHz	GigaHertz
API	Application Programming Interface
ICT	Information and Communication Technology
UI	User Interface
URL	Universal Resource Locator
HD	Hard Disk
CPU	Central Processing Unit

1. INTRODUCTION & OBJECTIVE

The objective of this project is to build four android applications with various features, collectively called College Connect. Every app will have different features and will be interlinked with each other to provide the required functionality.

1.1. Problem Statement

This project tries to tackle various problems present in the institutes and offers a single integrated platform based on android called College Connect. We have tried to solve the following problems:

- Students, especially the newcomers, were unable to contact teachers. Therefore, we have introduced a contact feature where they can contact teachers via external apps or via our built-in chat system with ease.
- In case a teacher is unable to attend a lecture, students were not notified about the same, especially in physical classes in real time. So, whenever a teacher is not able to take class, they can cancel that class from their app and the same will be notified to students.
- There is no interface where students can view their real time attendance for various subjects.
- There is no common discussion forum for the whole institute
- No single platform for students where they can view their timetables, syllabus, events, assignment deadlines, tests dates etc. in real time. (Integrated in single app)
- For teachers, the canteen transactions were not stored and they had to call the canteen to order food.
- The colleges have to relay the information about various events and announcements on the website or over the email. And checking these regularly can be a bit of hassle and inconvenient too.
- At last but not least, no integrated bundle of applications are available for practical usage.

1.2. Problem Structure

The plan is to design four android apps for:

- *Students / Alumni*
 - *Teachers*
 - *Canteen*
 - *Admins*
-
- *College Connect Student App*: In this app, students can view their timetables, events, tests & assignment deadlines in real time. Moreover, they can contact teachers either via external apps like whatsapp, phone or can chat with them using a built-in chat system. Students can join rooms where they can see all the subject related items posted by the teacher. We will also be incorporating a common discussion forum in this app.
 - *College Connect Teachers App*: In this app, teachers can view their timetables for the day or edit them in real time. E.g. They can cancel the classes for the day if they are absent and it will be notified to students. Moreover, they can create rooms for a subject and can share all the materials related to it in that room. Teachers can contact students or other faculty members either using external apps or using our built-in chat system. Teachers can send documents and can create various deadlines or test schedules via this app. Atlast, but not least, teachers can order food from the canteen via this app and all the transactions / orders will be stored on the server which means the teacher can view how much balance is outstanding to the canteen along with tons of other features.
 - *College Connect Canteen App*: In this app, canteen vendors can view / edit / change availability of various food items available in the canteen. We will implement an order management system in this app and canteen vendors can easily view and can edit orders. We will also develop a built-in chat system for canteen vendors and teachers.
 - *College Connect Admin App*: This is the backbone app of College Connect applications and all the users (Students, teachers, canteen vendors, other admin users) can be created / modified from this app. Admins can contact any person in College

with our built in chat system and have no restrictions. They can add course structures, timetables and can create various events. They can send custom notifications and can send documents to various other users. We will also develop an admin dashboard using React and Django.

1.3. Features & Goals

- Authorization & Authentication (login screen): All the apps will have an authorization & authentication interface where users can login. All the users having a college email ID can sign up directly or we can provide them login credentials individually.
- Home Screen: We are planning to implement different home screens for the different apps. For students, it will show a time-table, due assignments, or college notices/events etc. For teachers, it will show their timetable, assignments given to the students with their deadlines etc. For administrators, it will show functionality to send notices to the students, teachers etc. It may come handy to send the details of events/workshops related to the college. Also, admins will be able to modify/add/delete the profile data.
- Subjects Information: We can implement this thing for the students and for teachers as well where they can find the subjects syllabus, credit score, breakdown of marks etc. for various courses (available on college website).
- Attendance Feature: We will implement an attendance feature in the app where the teacher will mark attendance from their app for a particular day by just selecting students who are present from the dropdown menu and then students can view their attendance from the student app. Moreover, teachers can view the total attendance of all the students and can export into PDF format.. We will also implement various tools like filtering and showing a list of students with a particular attendance.
- Chat Functionality: We will implement a chat functionality where students can contact teachers and vice-versa through the app. If some student wants some

help/guidance, he/she will be able to contact the teacher directly.

- *Document Viewing Feature:* Teachers can send assignments, documents and can post announcements from their app and students can see these from the main app. Students can comment on these announcements and can download these from the app. It will be like a mini Google Classroom.
- *Events Functionality:* Admins can create events from their app and teachers and students can both see these on the home page of their apps. Teachers can't create events. Moreover, the admin can attach any document or notes or link for the event from their app.
- *Course Structure:* We will be adding a course structure tab for the students where students can see their course structure and can see the credits of the paper for that particular semester.
- *Contact Feature:* We will also add a contacts feature where a student can view the contact info of the teachers (their phone numbers, emails etc.) and can contact them via college app. (chat functionality).
- *Discussion Forums:* We are also planning to create a common discussion forum for the students and teachers. Teachers can create groups and can add students. Moreover, a student-alumni discussion forum can be created where students will be able to interact with the alumni.
- *Canteen Management:* Teachers can order food from the canteen from their app. They will be shown a list of food items available in the canteen which can be customized using the canteen app. Upon ordering, the canteen person will be notified in the app about the order.

1.4. Intended Audience

The intended audience is the students/alumni, teachers, canteen staff and institute management belonging to various educational sectors, who will be testing and using the College Connect apps. Also, the document is to be utilized by the software project management professor to evaluate the software's design and features.

2. SYSTEM ANALYSIS

Analysis can be defined as breaking up any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful way. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

It is the most creative and challenging phase of the system life cycle. The output of this phase is a description of the recommended alternative solution. The steps involved during system analysis process are:

- Understanding application
- Planning
- Scheduling
- Forming theory background
- Developing the solution
- Performing the test analysis
- Recommending alternative solutions
- Launching the proposed solution

System analysis can include looking at end-user implementation of a software package or product and involves gathering requirements for the system. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system. The project should address a real world interface design and be implementable. Feasibility Study is a major process in System Analysis. It helps in determining whether the project will yield a desired output with realistic and economic use of available resources.

2.1. Requirement Analysis

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

2.1.1. Functional Requirements:

In a development process, functional requirement provides the developer with a blueprint of how the application and its component will behave or function. The functional requirements describe what the application system should do. The functional requirements of the application are as follows:

So, the following are the identified functional requirements for College Connect:

General:

G1	All the data shall be hosted on Firebase for data processing and storage.
G2	A surface app page shall provide a user with all user system functionality.
G3	An app shall provide a User with all user/admin system functionality (according to access control)
G4	A display shall provide all the users with their respective functionalities according to their privileges.
G5	Notifications shall be sent to users in various event triggers

Students:

S1	Students shall be able to login via their email and password given by college office
S2	Students shall be able to view their timetables, events and various deadlines.
S3	Students shall be able to contact teachers via our built-in chat system
S4	Students can see their info and can request changes if corrections are required

S5	Students can join rooms
S6	Students can view materials posted by teachers in that rooms
S7	Students can view and can post in common discussion forums
S8	Students can view notifications in various events
S9	Students can view documents and notifications send by college office

Teachers:

T1	Teachers shall be able to login via their email and password given by the college office.
T2	Teachers shall be able to view food items of canteen and their availability
T3	Teachers shall be able to order food and add to cart
T4	Teachers shall be able to remove orders from cart
T5	Teachers shall be able to cancel the order
T6	Teachers receives notification for order accepted and once order is delivered
T7	Teachers can request account statements from vendors
T8	Teachers can view their previous orders
T9	Teachers can view their wallet balances
T10	Teachers can request addition / deletion of individual credit / debit items if something is not matching.
T11	Teachers can view their timetables
T12	Teachers can modify their timetables and can cancel classes
T13	Teachers can create rooms
T14	Teachers can add students in that rooms and can send link to students for joining

	them
T15	Teachers can upload different materials in that rooms
T16	Teachers can assign assignment deadlines
T17	Teachers can create test dates, presentation dates etc.
T18	Teachers can chat with students and teachers via our built-in chat system
T19	Teachers can view account details
T20	Teachers can request changes in their account by admin
T21	Teachers can send custom notifications to students

Admins:

A1	Admin shall able to Manage students, teachers and canteen vendors
A2	Admin shall be able to CRUD different users
A3	Admin shall be able to create courses
A5	Admin shall be able to create course structures and syllabus
A6	Admin shall be able to create batches of different courses
A7	Admin shall be able to assign course structures to batches
A8	Admin shall be able to create students in different batches
A9	Admin shall be able to create teachers account
A10	Admins shall be able to create timetables
A11	Admins shall be able to assign different subjects in timetables to different teachers
A12	Admins shall be able to create events
A13	Admins shall be able to manage events

A14	Admin shall be able to send notifications to different users
A15	Admin shall be able to create custom notifications
A16	Admin shall be able to send documents
A17	Admins shall be able to create common discussion forums
A18	Admin shall be able to chat with different users via our built-in chat system

Canteen Vendors:

C1	Vendors shall able to CRUD food items
C2	Vendors shall be able to receive orders from teachers
C3	Vendors shall be able to view the orders which has been ordered by teachers
C4	Vendors shall be able accept or cancel order depends upon the order received and availability of order
C5	Vendors shall able able to receive notifications once order delivered
C6	Vendors can view different teacher's account transactions
C7	Vendors can add more orders to teachers account in case the order is not done via apps
C8	Vendors can settle payments with teachers
C9	Vendors can print account transactions and can send it to teachers

2.1.2. Non-functional Requirements:

Non-functional requirements are not concerned with the functions of the system. Instead, they look at the criteria to which the application is expected to conform to. Non-functional requirements can include things like response time and reliability. Some of the Non-functional requirement for the applications are:

- College Connect should be compatible with android devices with SDK greater than 25.
- All the components of the application should be fully loaded within reliable time without downgrading performance.
- Should be user friendly and content should be readable by all types of users.
- Should take minimal time, effort, resources or cost to create the android application.
- Should provide the correct information about all the modules.
- Should consider the Response times
- Android applications should strictly follow Material Design guidelines
- The data should get retrieved from the server even in all types of network connectivity
- All the communications between server and app should have a secure endpoint

2.1.3. Usability Requirements:

The android applications can be accessed by users on their android devices, given they install the desired apks. We will be pushing this app to the playstore where the different users can download their respective apps.

2.1.4. Efficiency Requirements:

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

2.1.5. Accuracy:

The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.

2.1.6. Safety Requirements:

- As the app is connected to the internet occasionally, an antivirus should be installed on the system for its safety purpose.
- Since, app does not require any body movement to work properly, there is no physical safety requirement.

2.1.7. Performance Requirements:

- The program must be able to run concurrently by multiple users.
- Transmission of data to the database shall occur spontaneously.
- Queries upon the database shall be performed in less than 5 seconds.
- Reading and writing data shall occur instantly with the required HTTP protocols

2.1.8. Maintainability and Portability Requirements:

Changes (new parts in addition, password changes, and database changes) must be verified once per day at least.

2.1.9. Security Requirement:

- Re-authentication will be required once every week.
- The security and privacy policies will be open source and disclosed at the time of installation.

2.1.10. Software Quality Attributes:

- Availability: The system shall be available to all the users through information systems.
- Reliability: Overall reliability of the system and roll information shall be achieved through the process of database manipulation.
- Reusability: The app shall be able to be reused for each new semester.
- Robustness: If no network connection can be established to sync the data then the user will be allowed to do that later. If a network connection is lost during syncing, the app will allow the user to continue roll syncing at a later time.
- Updatability: The system shall allow for addition or deletion of classes, subjects, assignments and notes while incorporating new semesters.

- Usability: Usability of the system shall be achieved through help page, FAQs and an introductory training guide for all students upon installation of the program.

2.1.11. Feasibility Study:

After studying and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

- *Economic Feasibility*: This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor. All hardware and software cost has to be borne by us.
- *Technical Feasibility*: This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system and checked if everything was possible using different types of frontend and backend platforms.
- *Operational Feasibility*: No doubt the proposed system is fully GUI based and is very user friendly and all inputs to be taken are all self-explanatory even to a layman.

2.2. Technical Requirements

2.2.1. Hardware Requirements:

The minimum requirements for PC are:

- Operating System: Windows 10
- Application system: Android Studio, Android SDK
- Development environment: Java, XML, Android SDK
- IDE: Android Studio 4.1.1
- Frameworks: Android framework , Firebase (Authentication Methods, Realtime Database, Cloud Storage, Cloud Functions, Cloud Messaging)

The minimum requirements for Android device are:

- Android version: Nougat (7.1)
- API level: 25
- RAM: 2 GB recommended
- Processor: Quad Core 1.2GHz
- Internet Connectivity
- Google Play Services

2.2.2. Software Requirements:

- Application system: Android Studio, Android SDK
- Language: Java, XML, Android Framework

2.3. Input Design

Input design is part of overall system design that requires special attention. For designing input data, the data entered should be easy and free from errors. The input forms and dialog boxes in applications are designed using the controls available. Validation is made for each and every data that is entered. Help information is provided for the users.

Input design is the process of converting the user originated inputs to a computer based format. The collection of input data is considered to be the most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information.

The application gives an interface for registering and validating users and the data entered by users are converted to suitable models and uploaded to firebase for further use

2.4. Output Design

Output design of the applications generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The output is designed in such a way that it is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. The application presents the information in an interactive, user friendly manner. Moreover, the user can further print Invoices and can interact with the app.

3. PROJECT SCHEDULE

3.1. Project Phases

It is very important to follow up phases as the pre-requirement for every next phase is the completion of the previous one. Each project was divided broadly into 4 phases

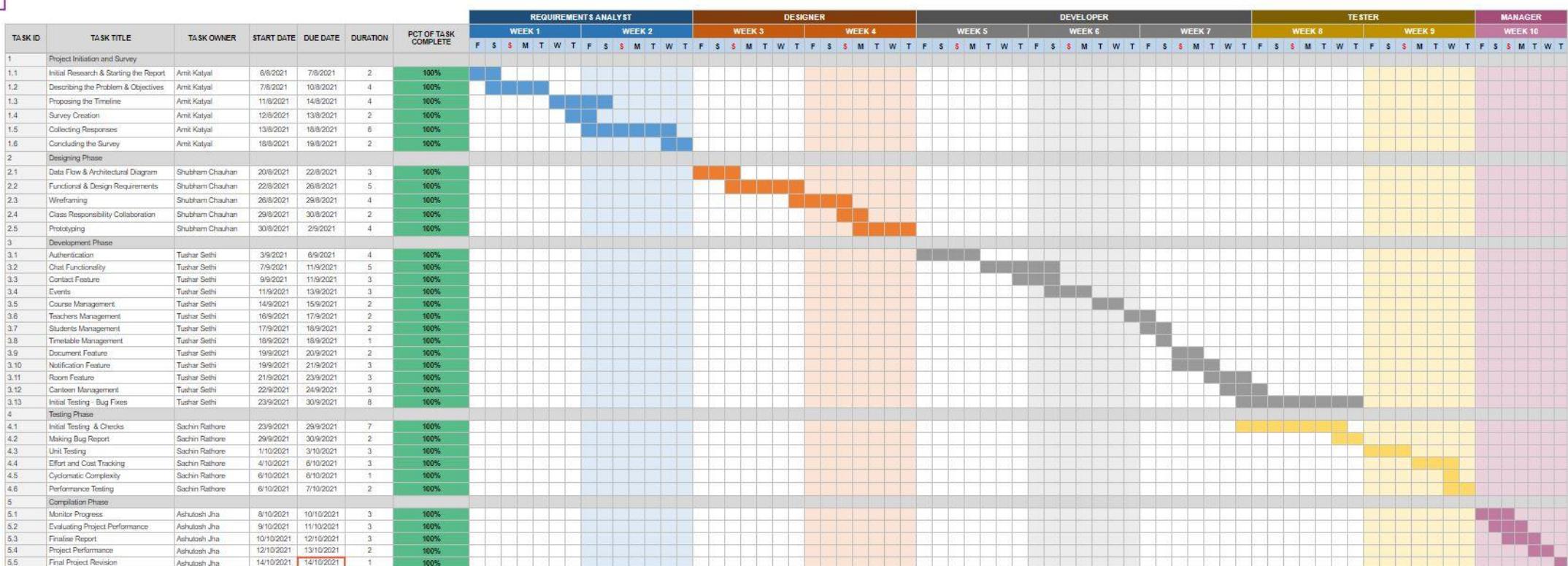
- Designing Layouts
- Implementing the classes and functions
- Hosting on Firebase and setting required methods
- Integrating APIs

3.2. Project Schedule

3.2.1. *Gantt Chart:*

Link to the document:

https://docs.google.com/spreadsheets/d/1oI-d5xD0lpBduqfWLqnBHqIIINqMAj9KQzP_n462X3c/edit?usp=sharing



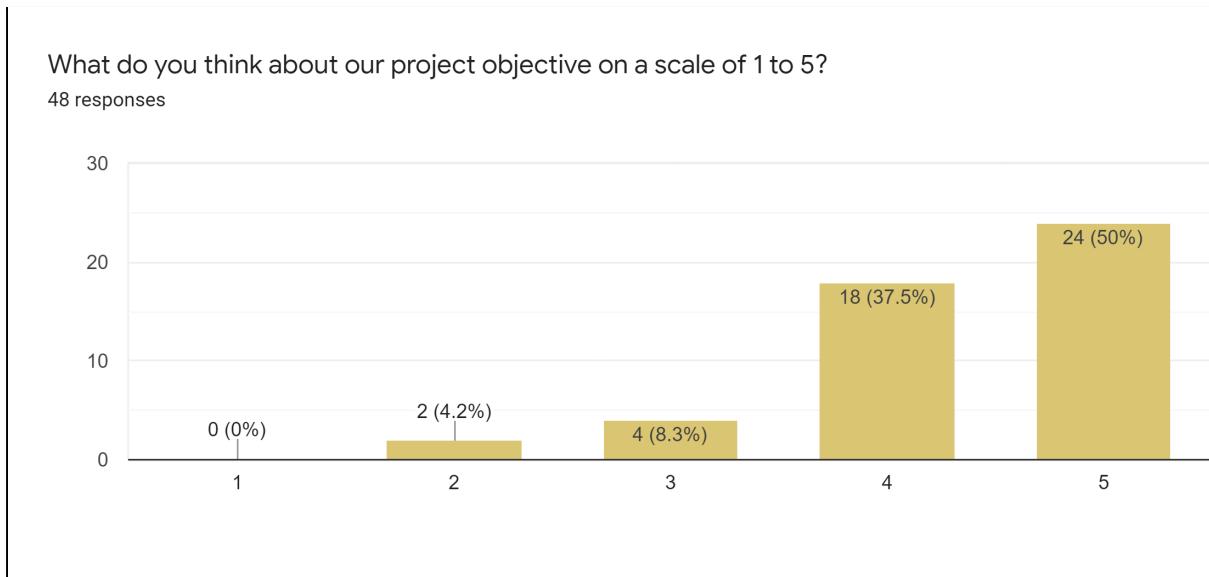
3.2.2. Program Evaluation and Review Technique:

Tasks	Time Allocation (person days)	Planned Start	Planned End	Optimistic Time Estimates (person days)	Most Likely Time Estimates (person days)	Pessimistic Time Estimates (person days)	Estimated Time $Te = (T_0 + 4T_m + T_p)/6$
Requirement Analysis and Project Planning	12	06/08/2021	18/08/2021	5	10	13	10
Setting up the Environment	10	20/8/2021	2/9/2021	3	5	14	6
Software Construction	28	3/9/2021	30/9/2021	9	19	30	19
Unit Testing	12	23/9/2021	7/10/2021	5	10	14	10
System Testing	5	8/10/2021	14/10/2021	2	4	8	4

4. METHODOLOGY

A survey was created using Google Forms having 10 suitable questions. This survey was then circulated to a set of Intended Audience and responses were recorded. The aim of this survey was to get a wide perspective from the audience and to get a much better scenario for the practical usage.

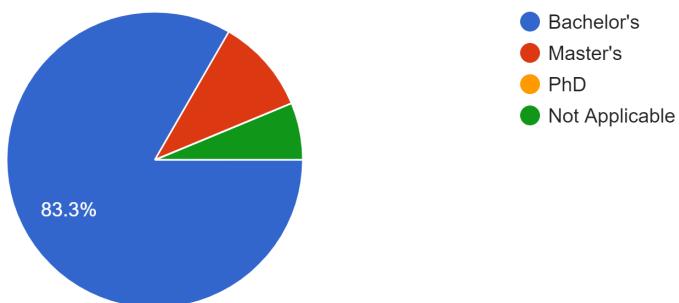
Following were the questions purposed and their respective responses recorded:



The aim of this question was to get an idea about the scope of the project. Most responses recorded were as expected.

Which college degree are you currently pursuing?

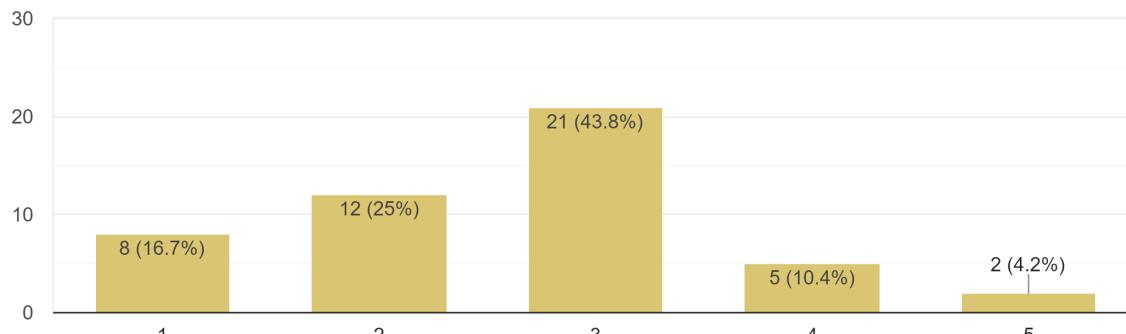
48 responses



The responses of this question give an overall idea about the qualification of the survey participants.

On a scale of 1 to 5, how satisfied are you with your college website to check events and announcements on regular basis?

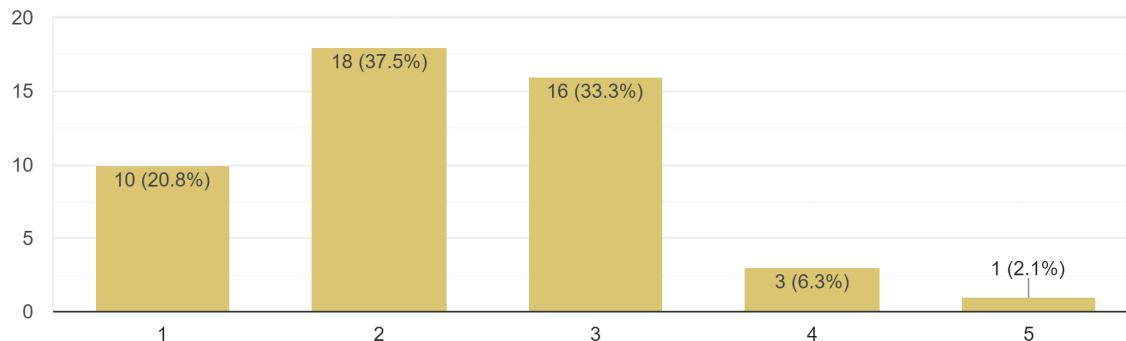
48 responses



The responses of this question conclude that most people find the college websites less interactive than their expectations. Hence, this is a good opportunity for us to include this feature in our apps which will be much more interactive than the website.

On a scale of 1 to 5, how satisfied are you with your traditional college attendance management system?

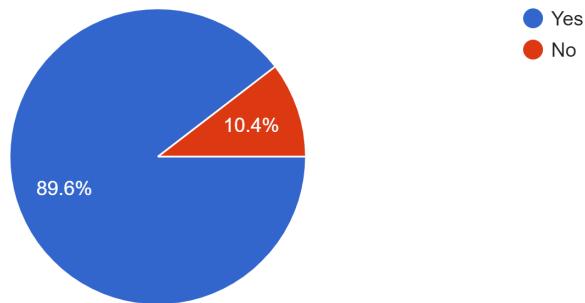
48 responses



These responses conclude that most people are not satisfied with the traditional attendance management system used in their colleges.

Would you like to have attendance feature in the app where students/teachers will be able to view/manage the attendance in real time?

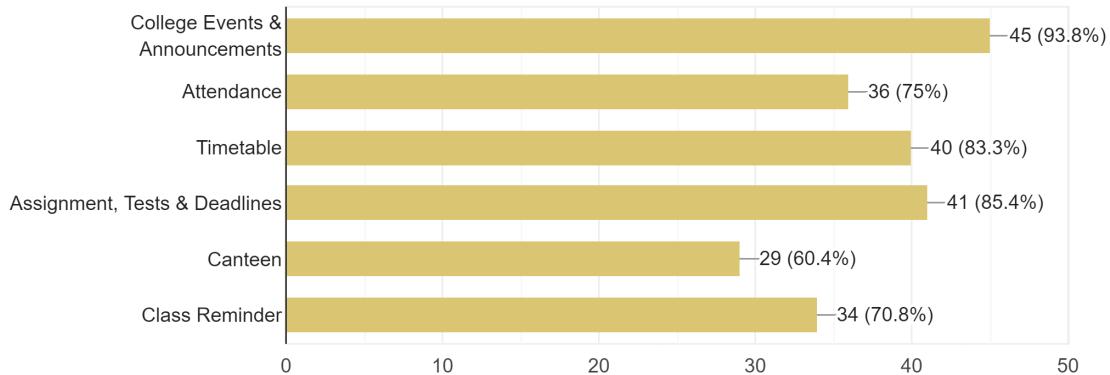
48 responses



The responses of the previous question and this one conclude that the recipients of this survey are looking forward to the attendance feature of the College Connect.

Which features would you prefer the most in a college app?

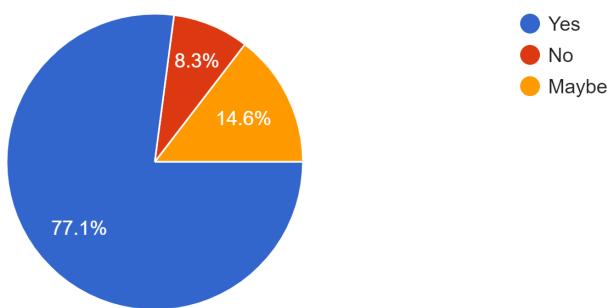
48 responses



These responses give an order to the proposed features and help us decide the priority of the feature to be implemented.

Would you like to use a chat feature in college app to chat with your teachers?

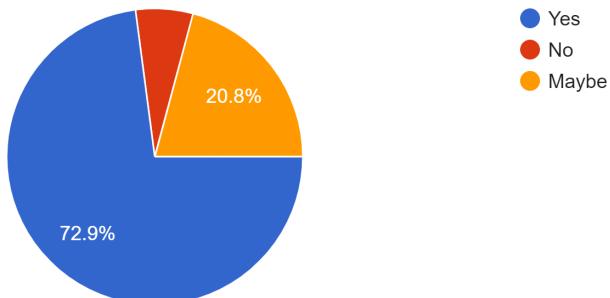
48 responses



The responses here were relatively mixed but rather on the positive side for the chat feature in the app as intended.

Would you like to have a dedicated assignment submission feature in our app instead of other alternatives out there?

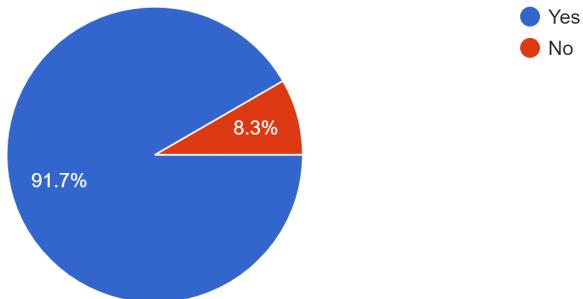
48 responses



The responses here were relatively mixed too due the competition like Google Classroom. But we think this feature integrated within a single app along with other features would be a great idea. This can further be improved in future.

Would you prefer placing food order online from College Canteen?

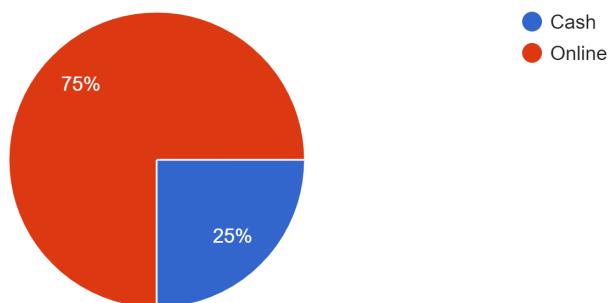
48 responses



The responses suggest that most people prefer the online system to order in the canteen, which is no surprise in this growing digital world.

What payment mode do you prefer to pay at college canteen?

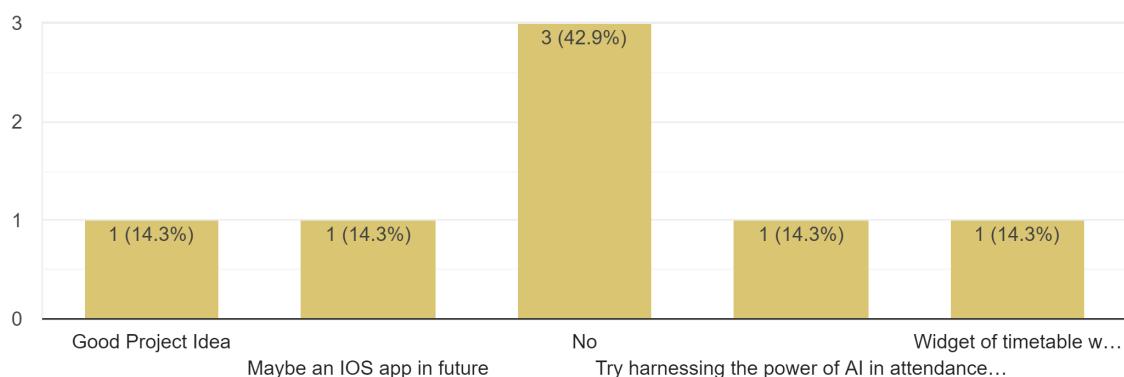
48 responses



Also, the responses here conclude that 3/4th of the participants prefer online payment for the orders in the canteen instead of cash.

Any features / suggestions you would like...

7 responses



This was a bonus to the survey apart from the 10 questions to get suggestions from the participants. The responses here were few but they also give other insights into the project.

5. IMPLEMENTATION

5.1. Implementation Tools

The various implementation tools are listed below:

- Operating System - Windows 10
- Development environment - Java, XML, Android SDK
- IDE - Android Studio 4.1.1

5.2. Tools Used

The front is an abstraction, simplifying the underlying component by providing a user-friendly interface. There are several tools available that can be used to develop the front end of android application

5.2.1. Android framework:

Android is one of the Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with the goal “accelerate innovation in mobile”. As such Android has taken over a field of mobile innovation. It is definitely a free and open platform that differs hardware from software that runs on it. It results in many more devices running the same application. Also it gives the possibility of a friendlier environment for developers and consumers. Android is a complete software package for a mobile device. Since the beginning the android team offers the developing kit (tools and frameworks) for creating mobile applications as quickly and easily as possible. In some cases you do not especially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufacturers it is a ready and free solution for their devices. Except specific drivers, the android community provides everything else to create their devices.

5.2.2. *Firebase:*

Firebase is Back-end as a service owned by Google which provides server-less back-end to the app developers. It makes app developers concentrate on the front-end by managing the back-end itself for developers. You can have many features and options like analytics, notifications to apps using FCM, cloud functions, authentication, real-time databases, storage. For testing purposes we can use testlab and crash reporting .

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of November 2020, the Firebase platform has 18 products, which are used by 1.5 million apps. We used it for hosting purposes. The features we used in our app are

- Authentication Methods
- Realtime Database
- Cloud Storage
- Cloud Functions
- Cloud Messaging

5.3. System Implementation

System implementation is the important stage of a project when the theoretical design is tuned into practical system. The main stages in the system implementation are Planning, Training, System testing and Changeover planning

5.4. System Maintenance

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system. The main task may be to adapt existing systems in a changing environment. Backup for the entire database files are taken and stored in cloud systems so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintenance of the system.

6. DEVELOPMENT

6.1. Designing Layouts

Layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and ViewGroup objects. The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView. The ViewGroup objects are usually called "layouts" and can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout.

You can declare a layout in two ways:

- Declare UI elements in XML: Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.
- Instantiate layout elements at runtime: App can create View and ViewGroup objects (and manipulate their properties) programmatically.

Declaring your UI in XML allows you to separate the presentation of your app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations. So, the main elements used in our project were:

- Parent Views: Linear View, Relative View and Constraint view for holding widgets and child elements
- Child Elements:
 - Buttons: Whenever user clicks it, then it performs a specific action
 - EditText: To take user Input
 - CardView: Show information inside cards that have a consistent look across the platform
 - GridLayout: a Layout manager that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles, and one component is placed in each rectangle. Used along with a card view.

- TextView
 - ImageView etc.
- *Sub-child elements:* Combination of the above child elements

6.2. Implementing the classes and functions

We used various classes and methods in our project which is beyond the scope of this report.

The main methods and classes used were:-

- setOnClickListener: It helps us to link a listener with certain attributes. setOnClickListener is a method in Android basically used with buttons, image buttons etc.
- addOnSuccessListener
- addOnFailureListener
- Toast: For displaying user a message
- Intent: For taking user to other activities
- DatabaseReference: for referring to the firebase database according to the document ID etc.

6.3. Hosting on Firebase and setting required methods

Firebase gives us functionality like analytics, databases, messaging and crash reporting so we can move quickly and focus on the users. Firebase is built on Google infrastructure and scales automatically, for even the largest apps. The steps involved were:

- We need to set up a Firebase Account and create a new project.
- Then provide the necessary details and provide the app's SHA-1 fingerprint.
- Add the firebase configuration files and plugins.
- Add firebase SDK to the app
- In the console under sign-in method, enable Email / Password authentication and implement the required methods according to the Firebase Docs.
- Then, in the database section, select the Realtime database and the region we wish to store our data. Then make a collection and accordingly document. Each document has a specific ID. In our case, it is the user's UID (Unique Identification Number). We use DatabaseReference to refer to documents and their subchilds.

7. TESTING

Testing is a method of assessing the functionality of a program. Testing is a set of processes aimed at investigating, evaluating and ascertaining the completeness and quality of a project. Testing refers to the process of implementing all or part of the system with the intent of finding errors. It is performed in order to find the bugs or errors in the system and minimize it. In general, testing is finding out how well something works .Testing is more than just debugging.

7.1. Types of Testing:

- *White Box Testing*: It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software. Knowledge of implementation is required. This type of testing of software is started after a detailed design document. It is generally applicable to the lower levels of software testing. Data domains along with inner or internal boundaries can be better tested. We have done path, loop and conditional testing under this category.
- *Black Box Testing*: It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. No knowledge of implementation is needed. This testing can be initiated on the basis of requirement specifications document. Can be done by trial and error ways and methods. We have done functional, non-functional and regression testing under this category.
- *Alpha Testing*: Alpha testing is a type of testing that is done on an application towards the end of a development process when the product is almost in a usable state. This type of testing does not involve functional testing on the application. Instead, it is a user testing on the application in order to understand the user behavior and experience on the application. Normally this test is performed by test engineers, employees and sometimes friends / family members with the aim of trying to emulate around 80% of the customers.

- *Beta Testing*: Beta testing is one of the final steps in your software development lifecycle (SDLC) before a product goes live. Also referred to as user testing or customer validation, beta testing aims to ensure that end users are satisfied with a software product before you make it generally available (GA). While beta tests want to catch any software bugs and errors that have snuck through the testing process, it is more about understanding and improving the product's full end user experience before it becomes GA. That means thoroughly investigating the experience flow and understanding any pain points that will hinder enjoyment of the experience for your end user.
- *Unit Testing*: Each division class of every page or layout is tested in the android device. Inspecting XML, and modifying style and layout in real-time.
- *Integration testing*: This type of white box testing involves combining individual units or components of the application's source code and testing them as a group. The purpose is to expose errors in the interactions of the different interfaces with one another. It takes place after unit testing.
- *System Testing*: After completing the overall application design and development it is tested for error. We have also validated output errors with detailed debugging and have taken necessary actions.
- *Performance Testing*: Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

7.2. Test Cases

Link to the document:

<https://docs.google.com/document/d/1zd04SUdOCmomWKamEfbogose52S3BHwaJDfNU6CuXN4/edit?usp=sharing>

8. SYSTEM FEATURES

8.1. User Authentication

8.1.1. Description and Priority:

We have implemented a login screen where the person needs to provide their email and password generated by the office. In our backend, there is a separate node for each type of user which can login through this interface. As soon as the email and password is supplied to the EditText, it is sent to our backend services and an OAuth token from a federated identity provider is generated and then these credentials are passed to the Firebase Authentication SDK which are then verified and a return response is given back telling if this email exists in our database or not. If not, then the user is notified about the same. If the email exists and the password matches, the user is diverted to the main screen (home) where they can have their necessary actions. All these backend services are implemented using firebase.

The Backend services used are:

- Account Management
- Account Linking

Methods used are:

- emailBuilder for creating instance
- getInstance();
- createSignInIntentBuilder etc,

The limit for handling multiple logins is 200,000 and after that it is done through first come first basis. Simultaneous responses sent from a single database is 100,000/second. These are the backend services and depends on the no. of servers

8.1.2. Stimulus / Response Sequences:

Following are the steps:

- User clicks on the app
- Login screen will be followed by the splash screen.
- Users will be given edit text views for filling out their email and password.
- After successful sign in the user will be directed to the home page.

8.1.3. Functional Requirements:

Cellular Service provider is required for connecting the phone to the internet for the user login process.

In the event of an authentication failure, the Firebase SDK will throw one of the following types of exception:

- [FirebaseAuthInvalidUserException](#): This exception indicates that account does not exist or has been disabled in the Firebase console. The precise reason for the exception can be identified by accessing the error code.
- [FirebaseAuthInvalidCredentialsException](#): This exception signifies that the user has provided the invalid login credentials.
- [FirebaseAuthUserCollisionException](#): Thrown during account creation, this exception indicates a problem with the email address entered by the user.
- [FirebaseAuthRecentLoginRequiredException](#): The user has attempted to perform a security sensitive operation but too much time has elapsed since signing in to the app. When this exception is detected the user will need to be re-authenticated.

All these exceptions will be handled by showing an alert dialog to the user and the user needs to reauthenticate our application.

8.2. Subject Information

8.2.1. Description and Priority:

Admins can create course structures under which they need to create subjects and allocate these course structures to different batches. In batches, there are different students and whenever a student logs in, they can find their course's subject's syllabus, credit score, breakdown of marks etc. Teachers can also view this data.

8.2.2. Stimulus / Response Sequences:

Following are the steps:

- In Navigation View Of Students app, there will be a menu item named “My Course Structure”
- After clicking on that menu item, students can view their course structures as well as other course structures

8.2.3. Functional Requirements:

Cellular Service provider for connecting the phone to the internet to retrieve the data from Firebase Realtime Database.

8.3. Timetable Fetching

8.3.1. Description and Priority:

Admins can create timetables and assign different subjects to different structures. This data is managed by SubjectsModel and TimetableModel. Holidays are managed by HolidaysModel. As soon as the admin has generated or created all the data, it is uploaded to Firebase. Students and teachers can then view their respective timetables from their app

8.3.2. Stimulus / Response Sequences:

Following are the steps

- Admin can create timetables by clicking on the “Add Timetable” card from their home menu which will open up a timetable fragment.
- All the days in the week will be listed on that fragment
- Admin needs to assign this timetable to a batch.

- When an admin clicks on a day, then they will be given the list of subjects from that batch's course structure.
- Admin needs to drop-down them to the respective days or click on the add button
- Students and teachers can view their respective timetables on the home screen.

8.3.3. *Functional Requirements:*

Cellular Service provider for connecting the phone to the internet to store the data on Firebase.

8.4. Chat Functionality

8.4.1. *Description and Priority:*

We have implemented a built-in chat functionality where students can contact teachers and vice-versa through the app. If some student wants some help/guidance, he/she will be able to contact the teacher directly. Moreover, if a student / teacher doesn't want to use this functionality they can use other external apps which we have linked in College Connect like whatsapp, linkedin etc. Admins have got all the privileges and can contact any user of the institute. No student to student communication is allowed.

Note: These chats are not encrypted and if the admin opens up the firebase console, they can view these chats.

8.4.2. *Stimulus / Response Sequences:*

- In navigation view of different apps, there is a menu item named “Chat”
- After clicking on that menu item, the user will be directed to Chat activity.
- They will have a list of the persons they can contact
- After clicking on the desired person, they can chat with them.

8.4.3. *Functional Requirements:*

Cellular Service provider for connecting the phone to the internet and to store the data on Firebase realtime database.

8.5. Rooms

8.5.1. Description and Priority:

Teachers can create rooms and can add students in those rooms. Teachers can then send assignments, documents and can post announcements from their app and students can see these from the main app. Moreover, teachers can also create test dates, project dates, assignment deadlines, report deadlines etc. Students can comment on these announcements and can download the necessary files from the app. It is like a mini Google Classroom. All the events will be notified to students and teachers whichever applicable through FCM.

8.5.2. Stimulus / Response Sequences:

Following are the steps:

- Teachers can create rooms by choosing a room's menu item in their navigation view.
- They will be presented with a add room option and a list of their rooms
- After clicking on “Add room”, they will be asked a name and followed by a list of students to be added.
- After clicking on “Done”, they can post their announcements on that room
- The same will be notified to students of that room.

8.5.3. Functional Requirements:

Cellular Service provider for connecting the phone to the internet to store the data on Firestore in case of google login.

8.6. Events Functionality

8.6.1. Description and Priority:

Admins can create events from their app and teachers and students can both see these on the home page of their apps. Teachers can't create events. Moreover, the admin can attach any document or notes or link for the event from their app.

8.6.2. *Stimulus / Response Sequences:*

- The user needs to click on the ADD NOTES button corresponding to each subject in order to add notes.
- This will open a dialog box where they can type-in or add images, pdfs etc.

8.6.3. *Functional Requirements:*

Cellular Service provider for connecting the phone to the internet to store the data on Firestore in case of google login.

8.7. Course Structure

8.7.1. *Description and Priority:*

We have added a course structure tab for the students where students can see their course structure and can see the credits of the paper for that particular semester. Admins can create course structures and can assign these course structures to different batches.

8.7.2. *Stimulus / Response Sequences:*

- The student needs to click on the COURSE STRUCTURE menu item in the navigation view.
- This will open a course structure fragment where the student can view their course structures.
- Admins can create course structures from the Add Course Structure card on home activity.
- Then they need to add subjects in that fragment and upon clicking Done, it will be uploaded to Firebase.

8.7.3. *Functional Requirements:*

Cellular Service provider for connecting the phone to the internet and to store the data on Firebase realtime database.

8.8. Canteen Management

8.8.1. Description and Priority:

Teachers can order food from the canteen from their app. They will be shown a list of food items available in the canteen which can be customized using the canteen app. Upon ordering, the canteen vendors will be notified in the app about the order. Teachers can track their wallet transactions and can track their current orders.

8.8.2. Stimulus / Response Sequences:

- The teacher needs to click on the ORDER FOOD menu item in the navigation view.
- This will open a list of food items of canteen with their availability
- They can then add items to cart
- In the cart, teachers can select in which room they want their order.
- Upon PLACE ORDER, order is sent to canteen vendor
- Upon successful event triggers, teachers are notified about their order status.

8.8.3. Functional Requirements:

Cellular Service provider for connecting the phone to the internet and to store the data on Firebase realtime database.

9. CONCLUSION

This project for the subject of Software Project Management was a great opportunity for us to discover new fields and ways of working. Learning Android was truly enthralling and opened new vistas for us. Indeed, it forced us to understand app development deeply, something that always interested us. We started the project with clear objectives, and our main goal at the beginning was to build just a College Connect apps bundle with a mobile interface. Therefore, we were very glad to be able to finally produce a complete and working application. We learned a lot and the project definitely broadened our horizons and we are thankful for that. The artistic atmosphere in which we worked clearly motivated us to try new things and gave us the freedom that was needed for such a project.

The application was able to connect to the server and retrieve information from the server. It can be installed and can function on several devices at the same time. The prototype is working on an android platform and made on the base of Java framework. It uses Firebase to store and to receive information. The information is stored on Firebase and can be accessed any time.

In conclusion, the College Connect app performs its functions perfectly but some things still need to be polished including the UI/UX & animations. There is a lot of scope for the improvement of algorithms as well as the fluidity of apps. The app will also be published on the playstore and a revenue model can also be implemented including ads or paid features.

Important Links:

- Survey: <https://forms.gle/AHRbmuUnSDqzqrxd8>
- Gantt Chart Excel Sheet:
https://docs.google.com/spreadsheets/d/1oI-d5xD0lpBduqfWLqnBHqIIINqMAj9KQzP_n462X3c/edit?usp=sharing
- Test Cases Document:
<https://docs.google.com/document/d/1zd04SUdOCmomWKamEfbogose52S3BHwaJDfNU6CuXN4/edit?usp=sharing>
- Screens:
https://docs.google.com/document/d/1Tf944LFMzxRE8iC_R1McOX6jsKDvUVyb-1NIKYDiDkw/edit?usp=sharing
- Presentation:
https://docs.google.com/presentation/d/185fwpG_qfD1qBulpB3gzZu7Yt4EBCHJloPHZN4hbGTc/edit?usp=sharing

References:

[1] How to write a good SRS for your Project

From <https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/>

[2] How To Write Software Design Documents

From <https://blog.tara.ai/software-design-documents/>

[3] Important Features You Should Include During Android App Development

From <https://www.mindinventory.com/blog/features-include-during-android-app-development/>

[4] Software Development Process: How to Pick The Process That's Right For You

From <https://plan.io/blog/software-development-process/>

[5] Software Development Process: From Idea to Product

From <https://www.udacity.com/course/software-development-process--ud805>

[6] How to turn an app idea into a product: Creating a feature set

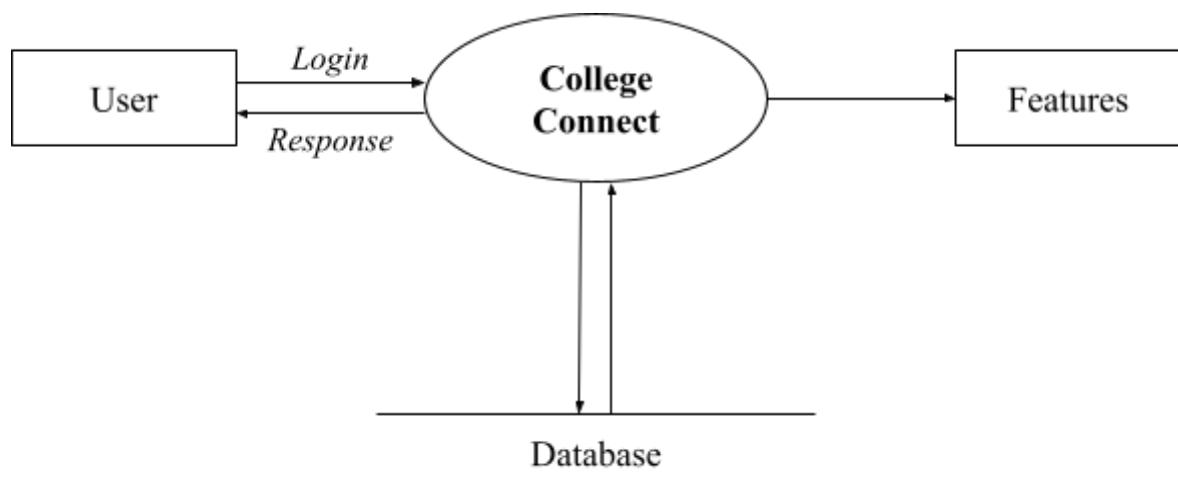
From <https://www.peerbits.com/blog/turn-an-app-idea-into-a-product.html>

[7] Android Developers Documentation

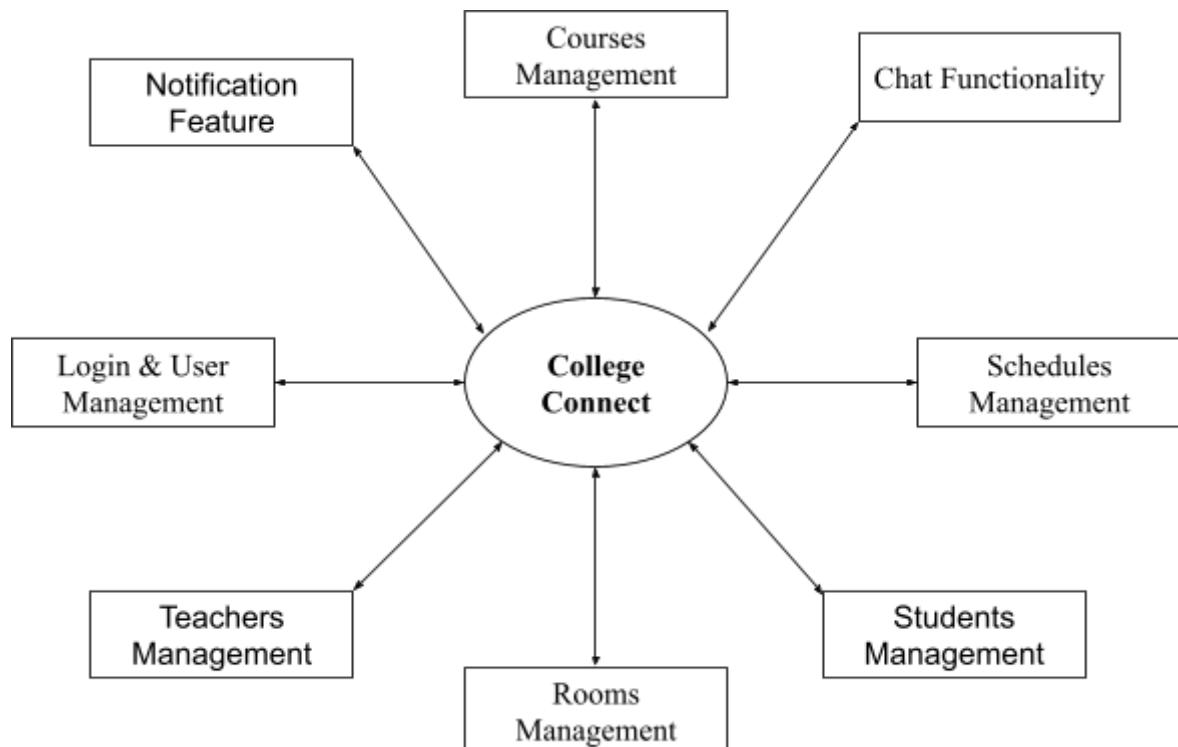
From <https://developer.android.com>

Appendix A: DESIGN DOCUMENT

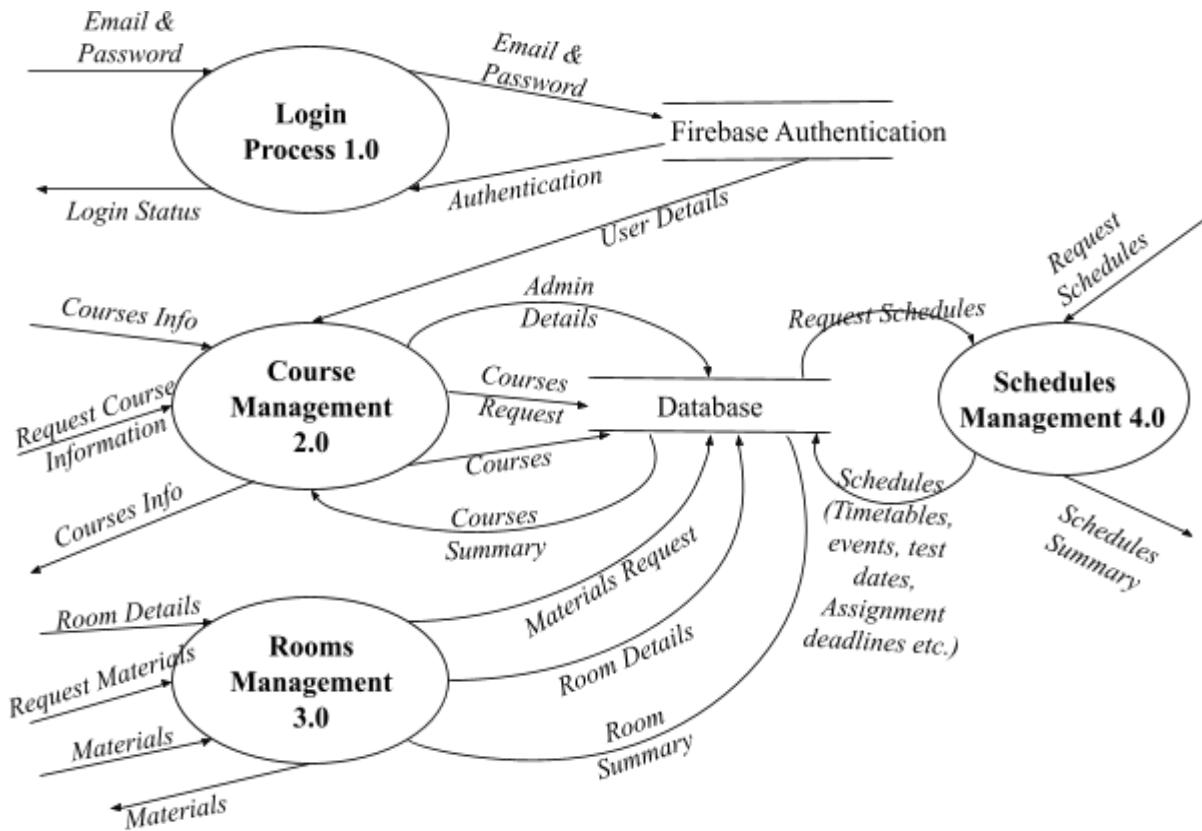
1. Context Diagram (*Data Flow Diagram Level-0*)



2. Data Flow Diagram Level-1

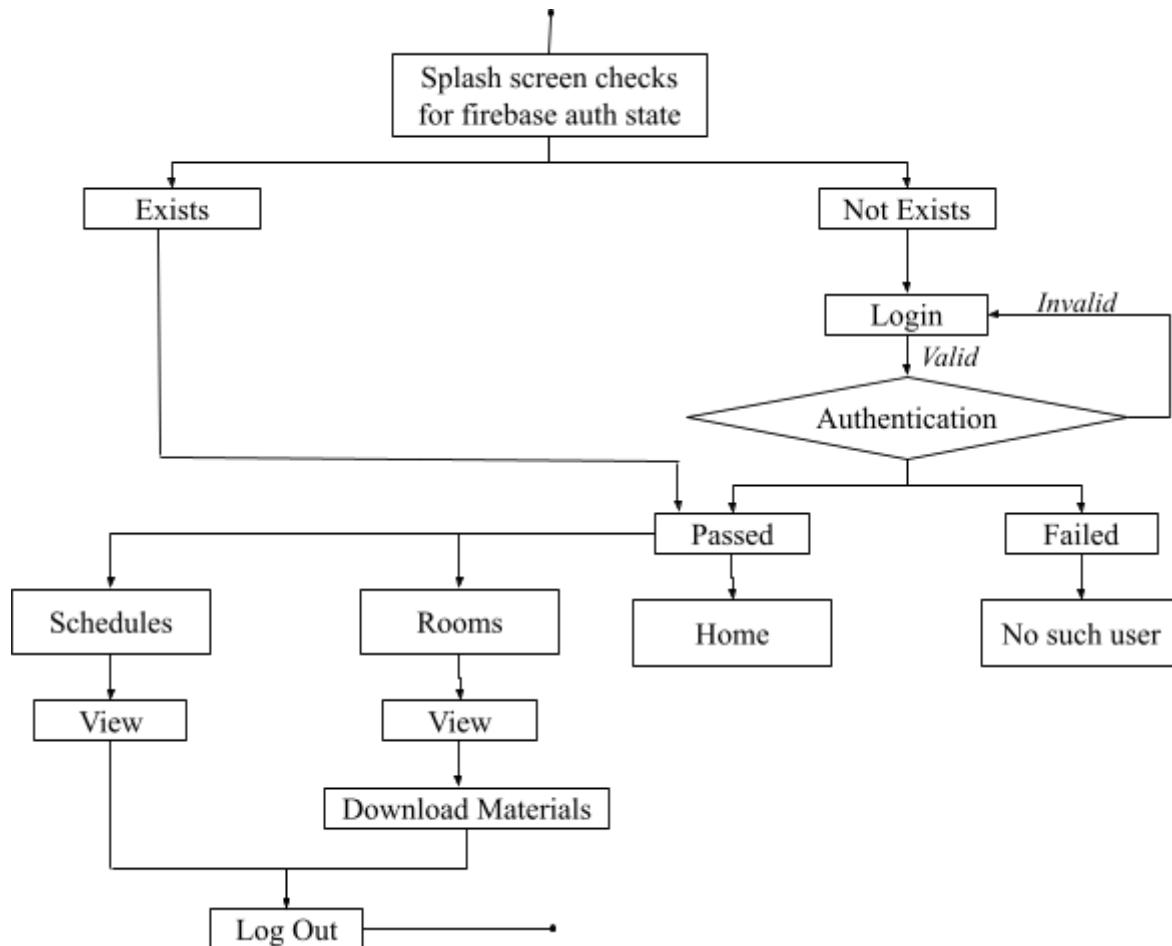


3. Data Flow Diagram Level-3

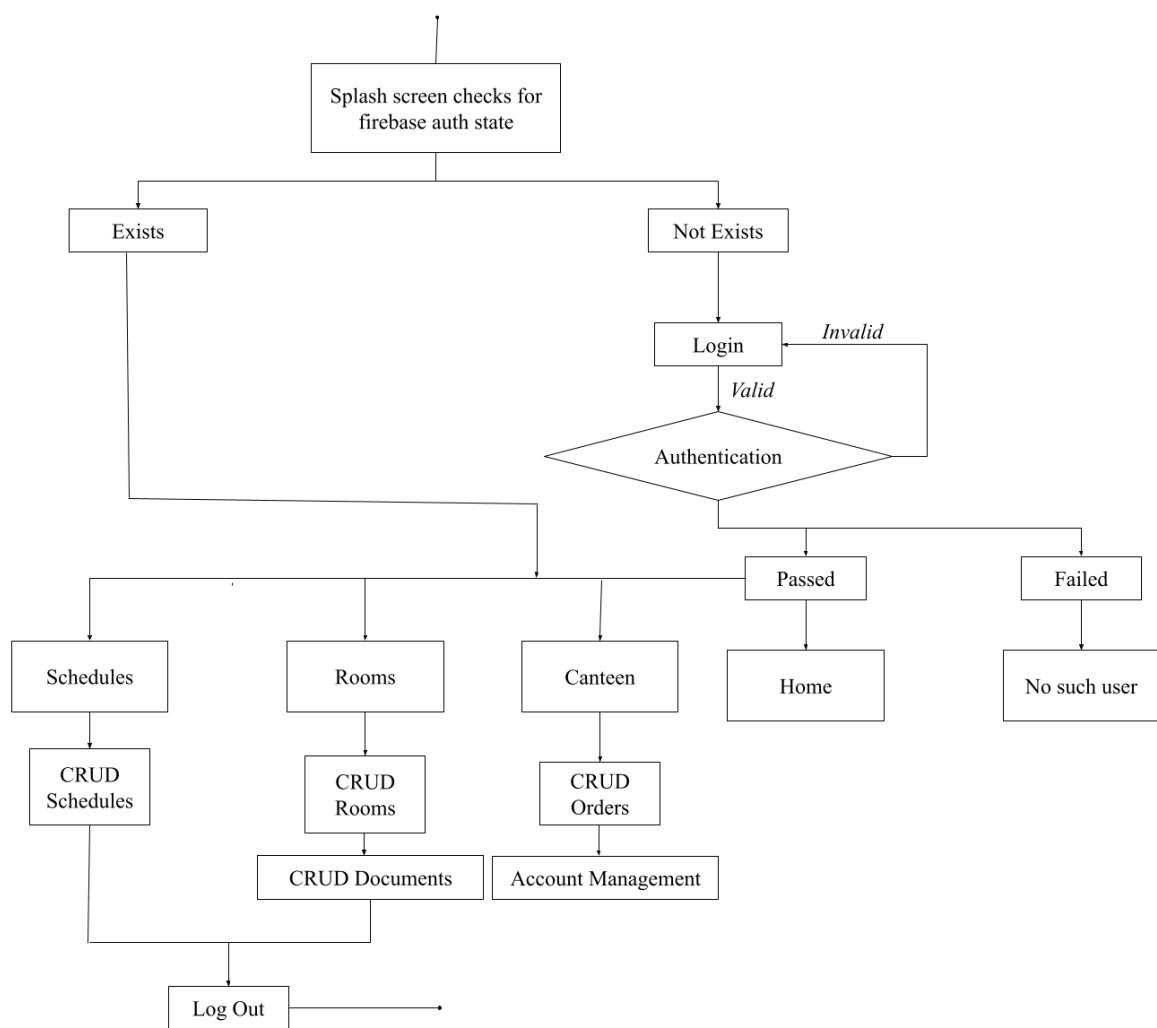


4. State Transition Diagram

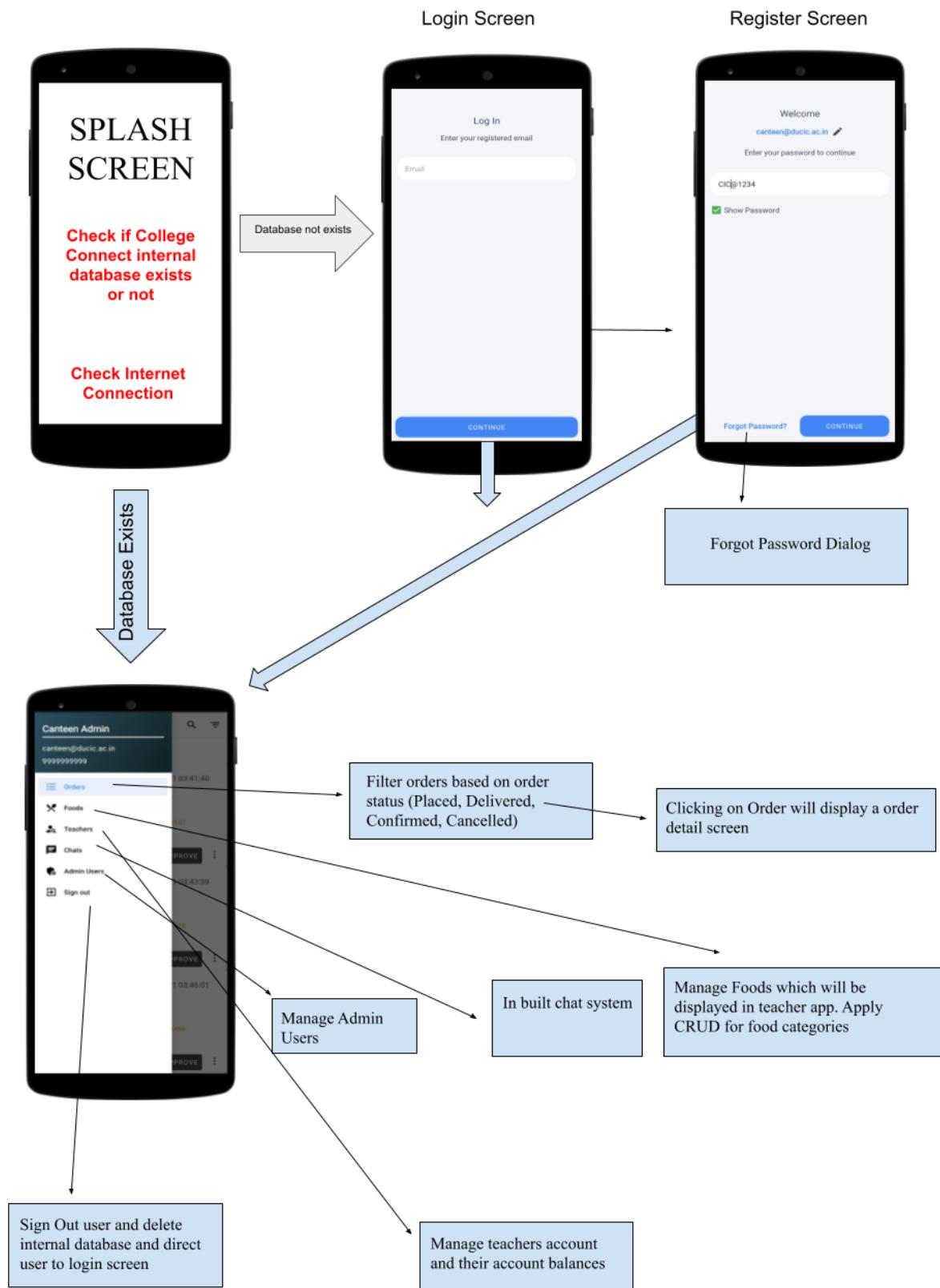
For Students App:



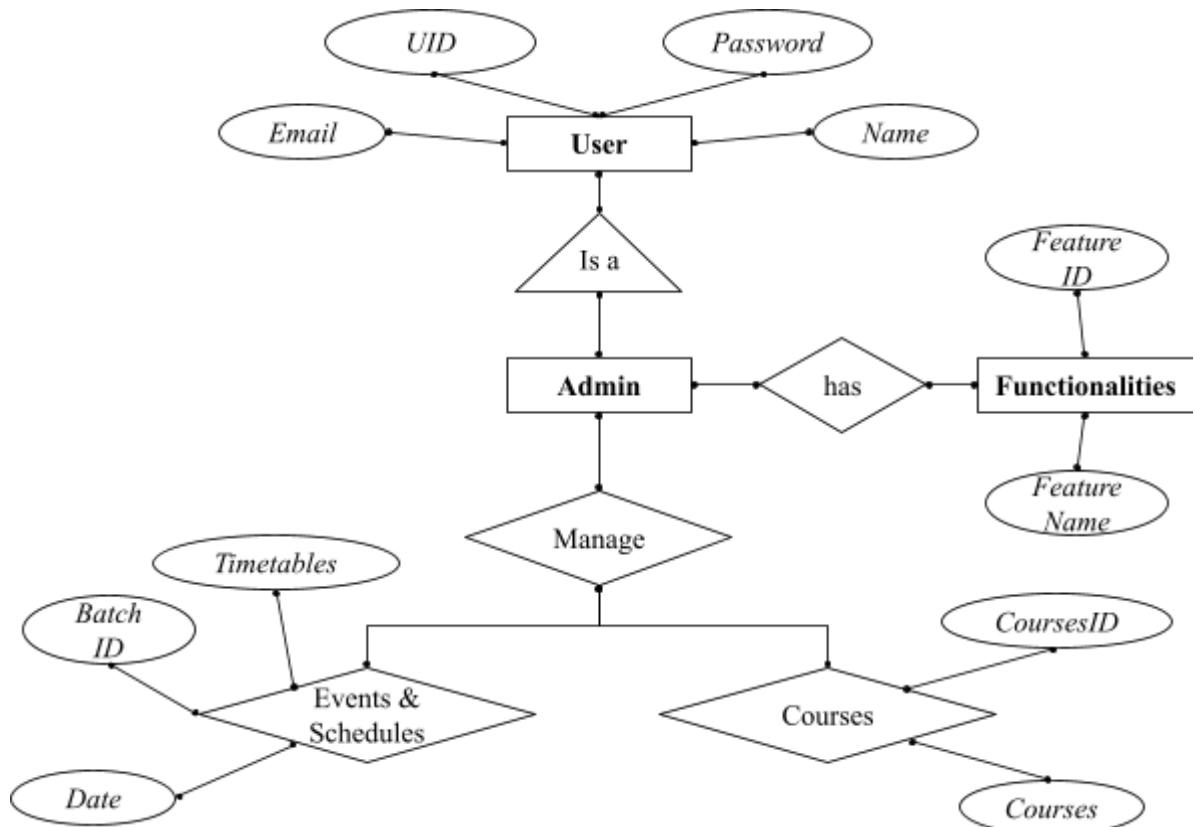
For Teachers App:



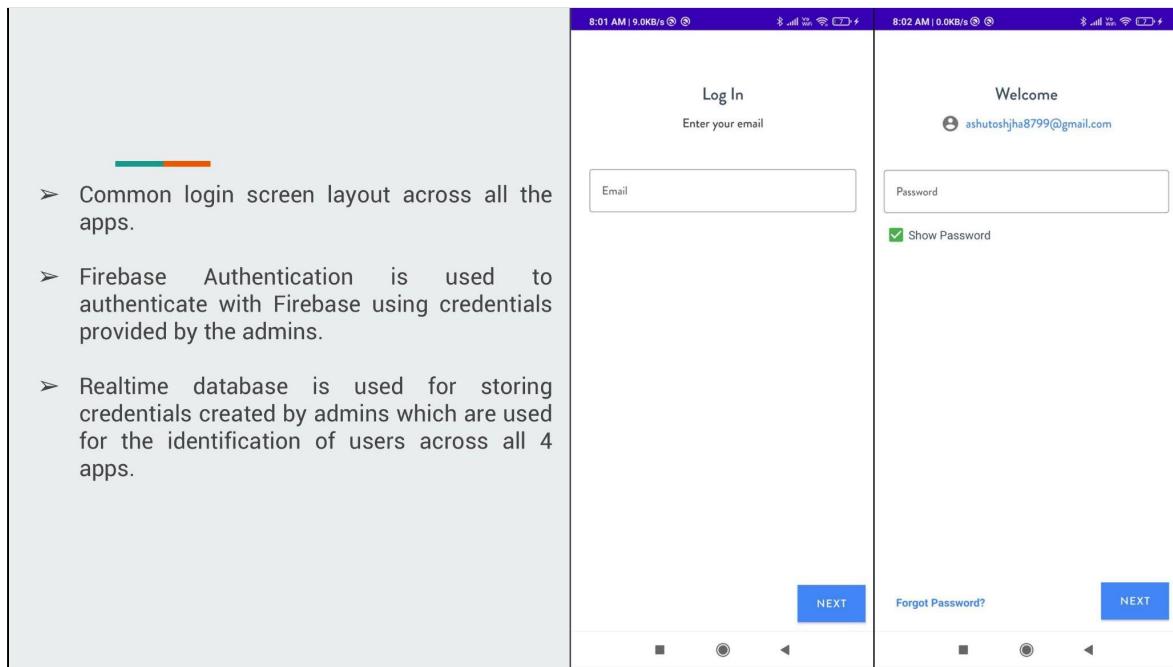
5. Architectural Diagram



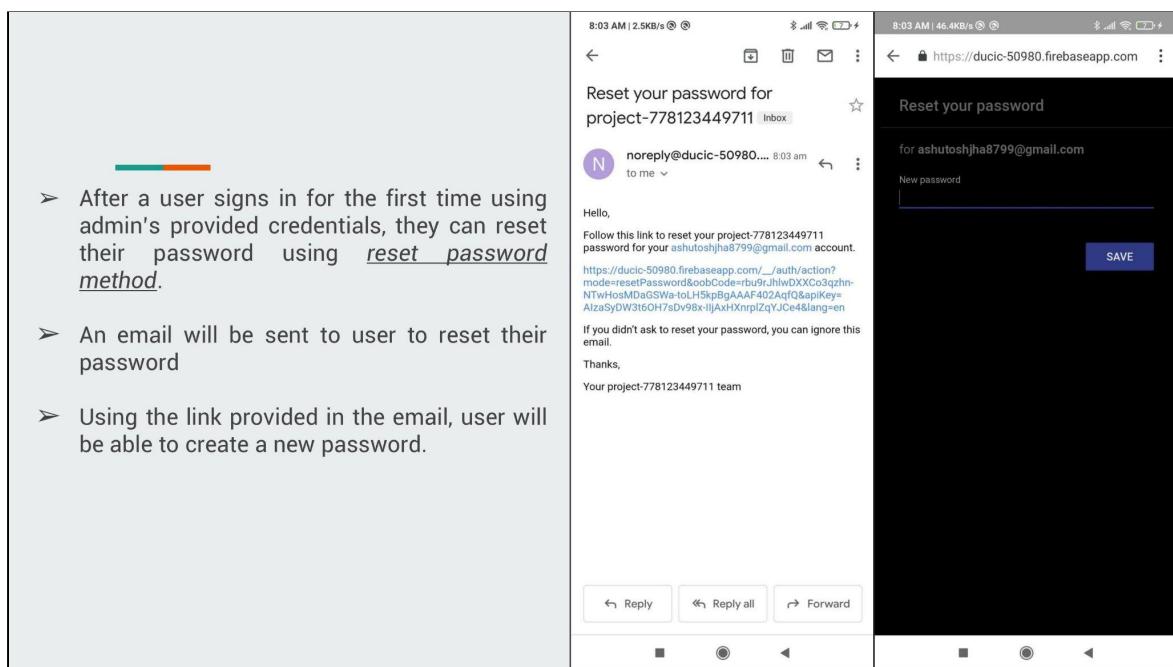
6. Entity Relationship Diagram



Appendix B: Screenshots



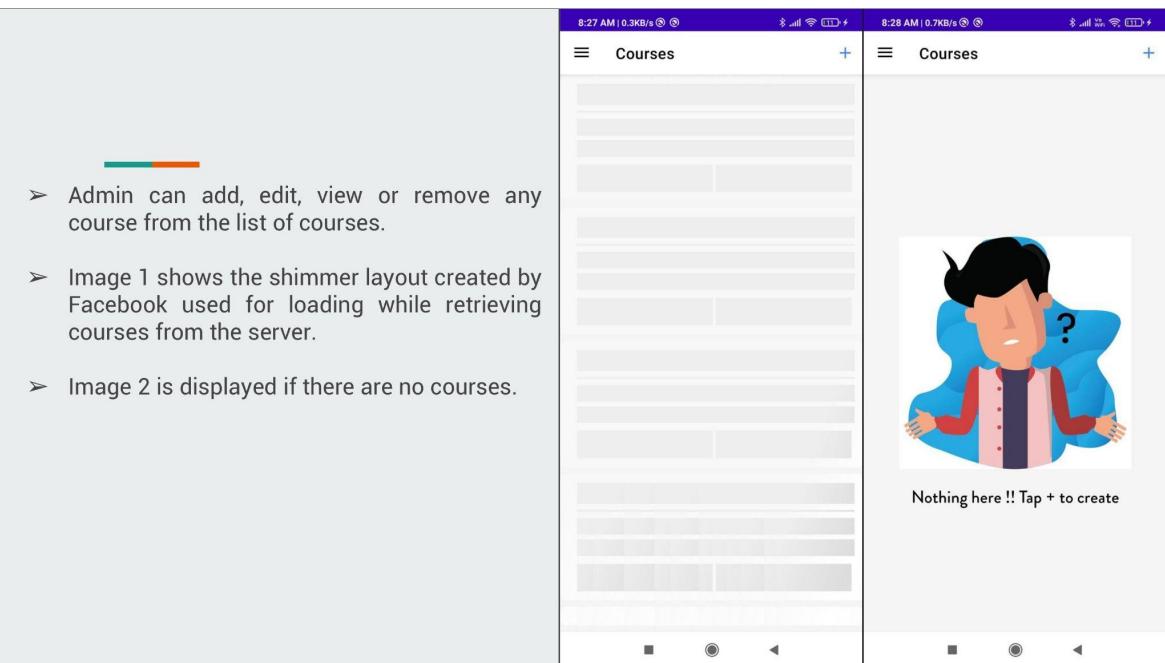
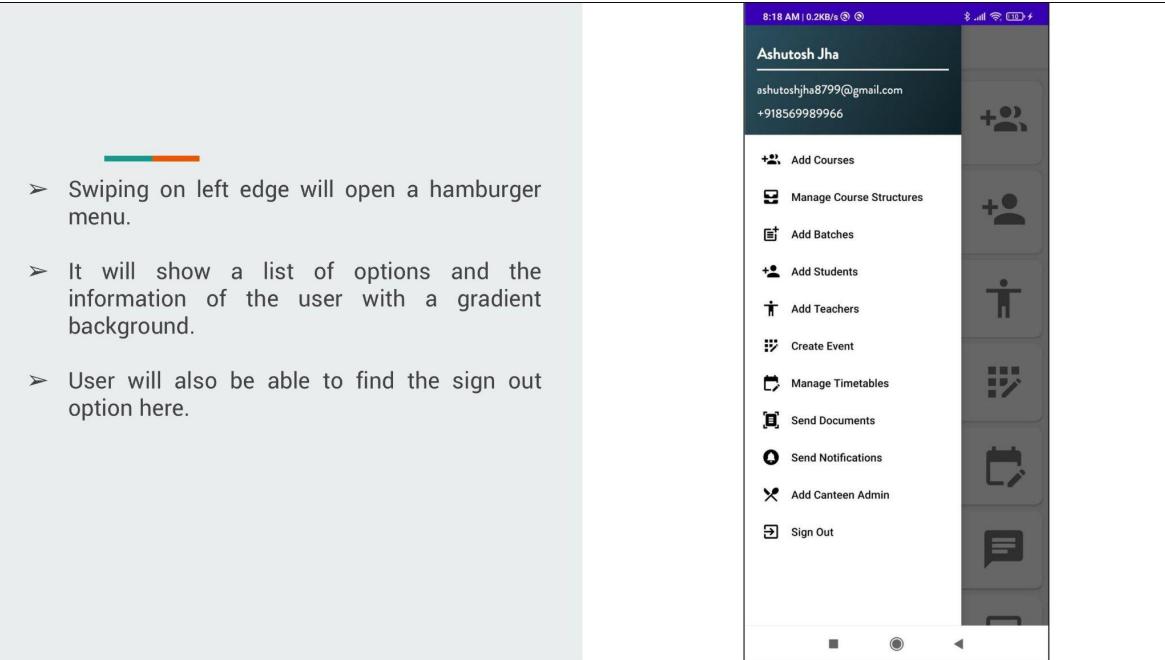
- Common login screen layout across all the apps.
- Firebase Authentication is used to authenticate with Firebase using credentials provided by the admins.
- Realtime database is used for storing credentials created by admins which are used for the identification of users across all 4 apps.

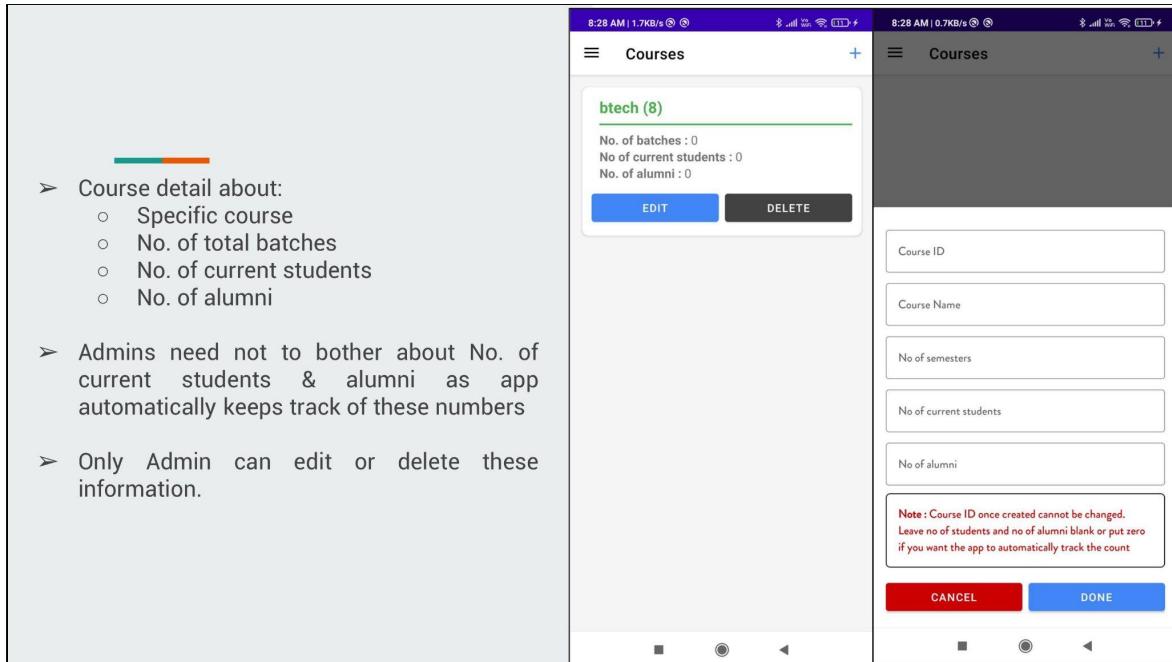


- After a user signs in for the first time using admin's provided credentials, they can reset their password using reset_password method.
- An email will be sent to user to reset their password
- Using the link provided in the email, user will be able to create a new password.

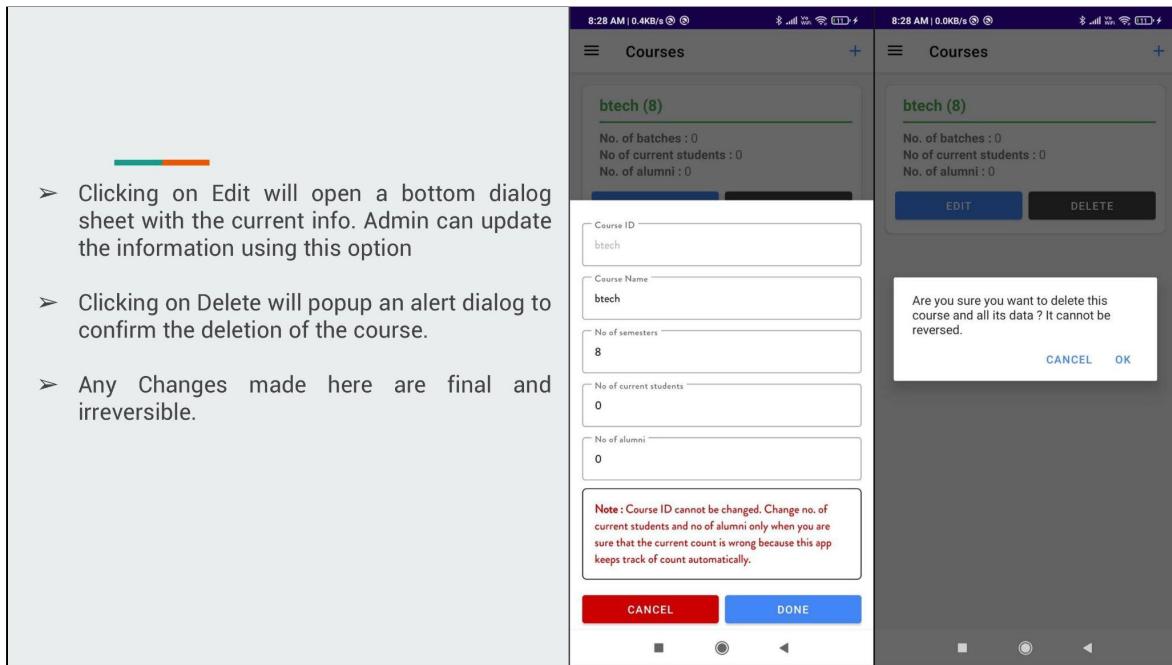
- Cards providing different options to admin.
- Admin can add, edit or remove information.
- Admin will be responsible for creating any event happening in CIC.
- Admins can also send notifications and documents to teachers/students.
- Moreover, admins can manage timetables of different batches and teachers.

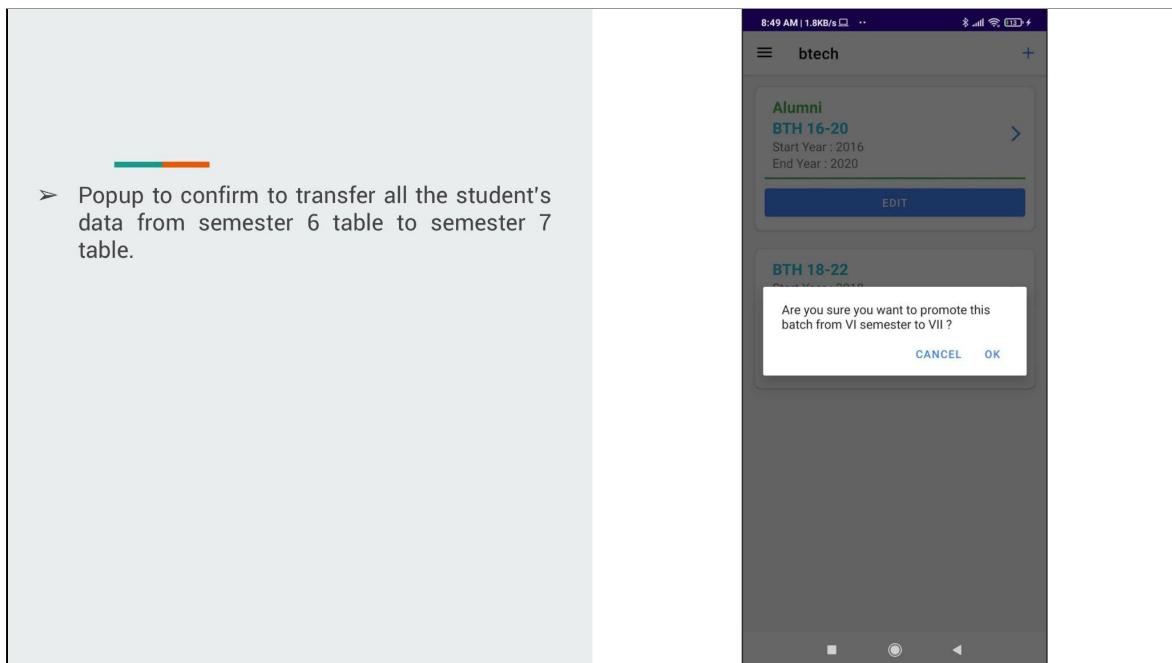
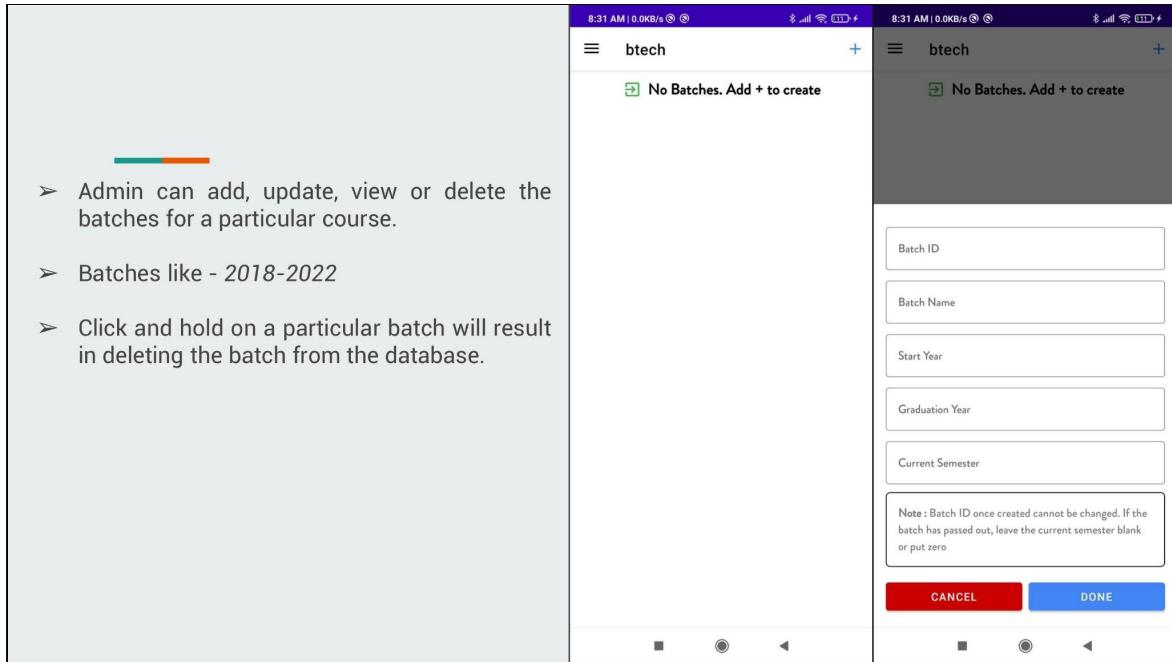
Left Screen (Home)	Right Screen (Home)
Courses	View and manage events
Students	Timetables
Teachers	Chats
Events	Courses Structure
Timetables	Documents
Chats	Notifications
Courses Structure	Canteen





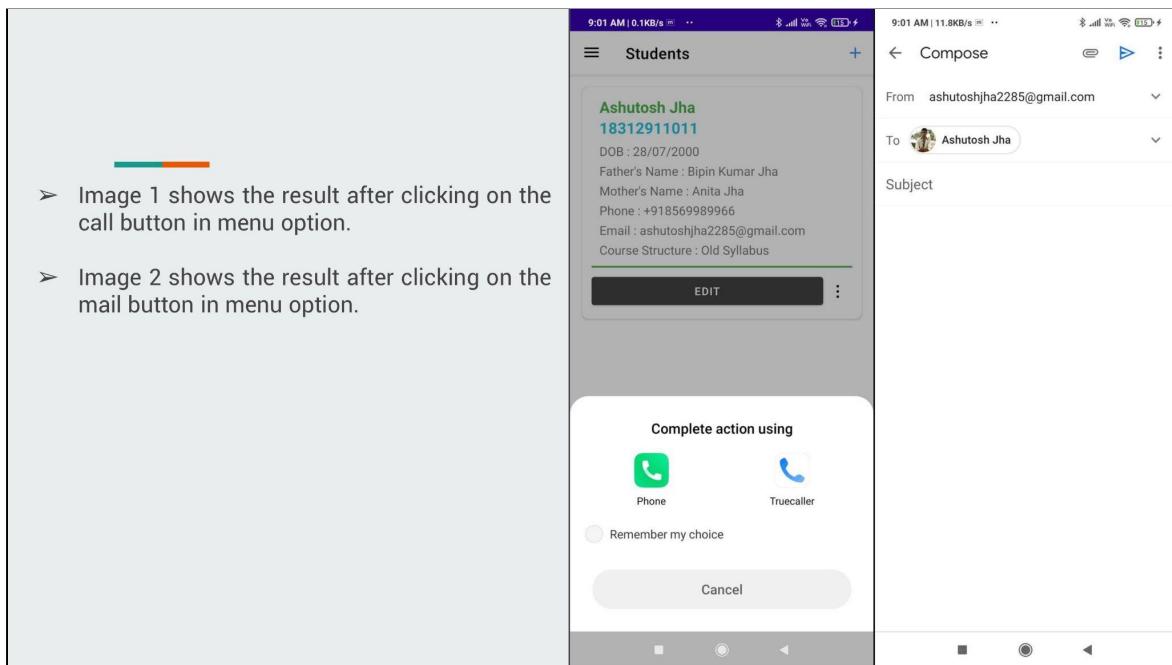
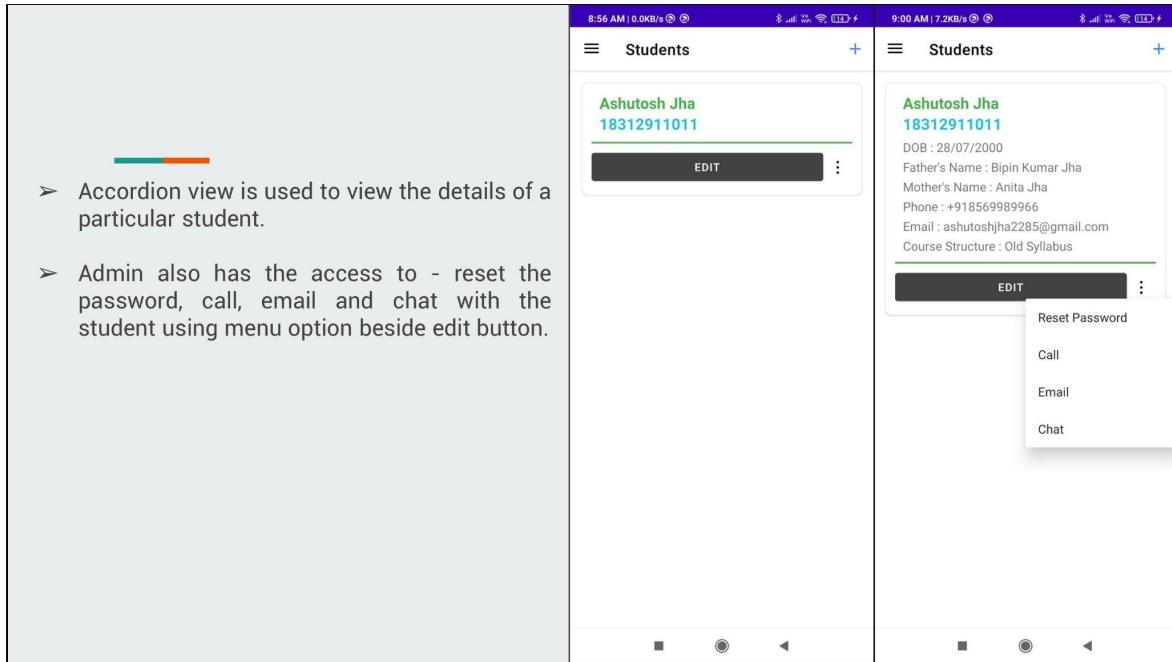
- Course detail about:
 - Specific course
 - No. of total batches
 - No. of current students
 - No. of alumni
- Admins need not to bother about No. of current students & alumni as app automatically keeps track of these numbers
- Only Admin can edit or delete these information.

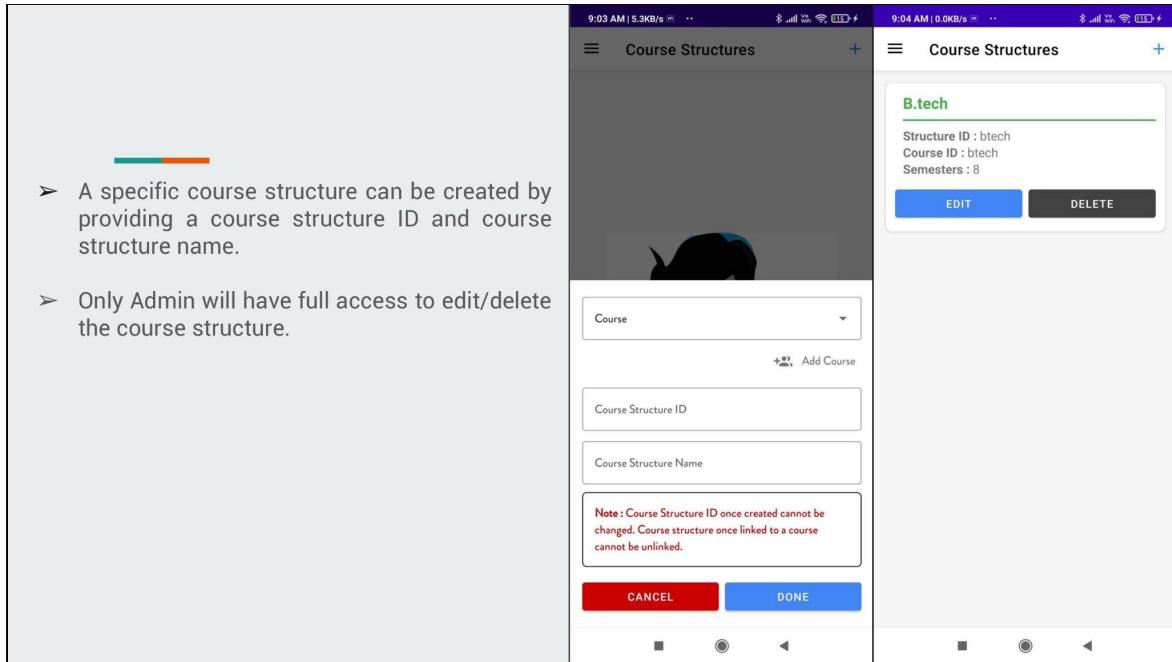




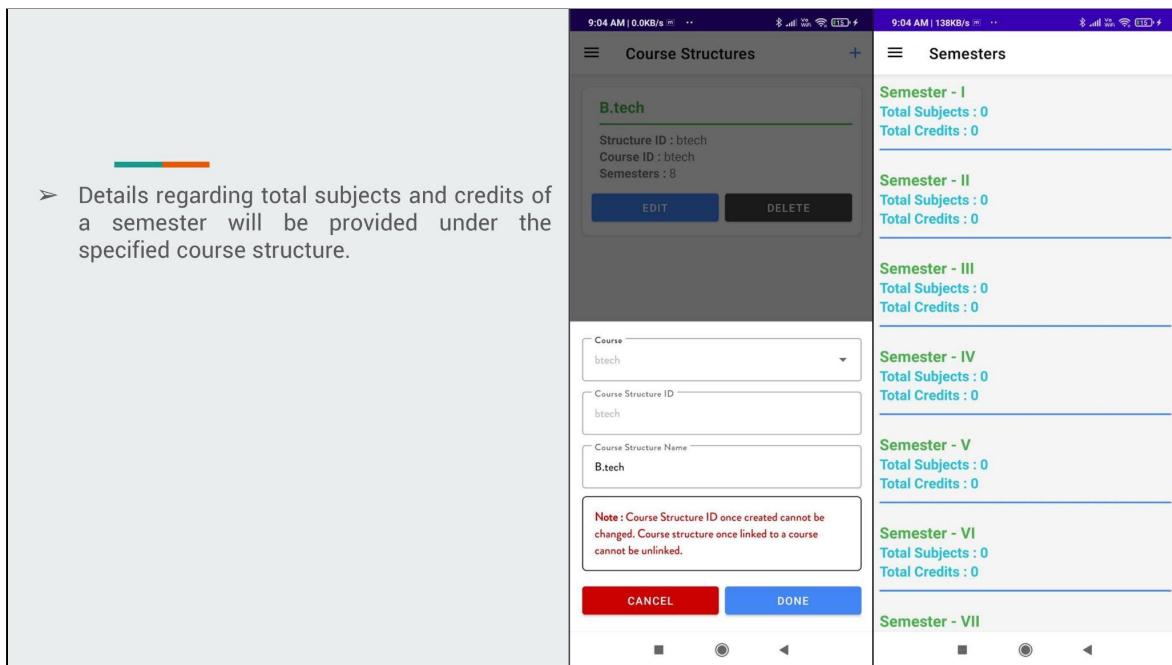
➤ View and edit panel for all the batches.

➤ Admin can add, edit, view or delete any student to a particular batch of a particular course.

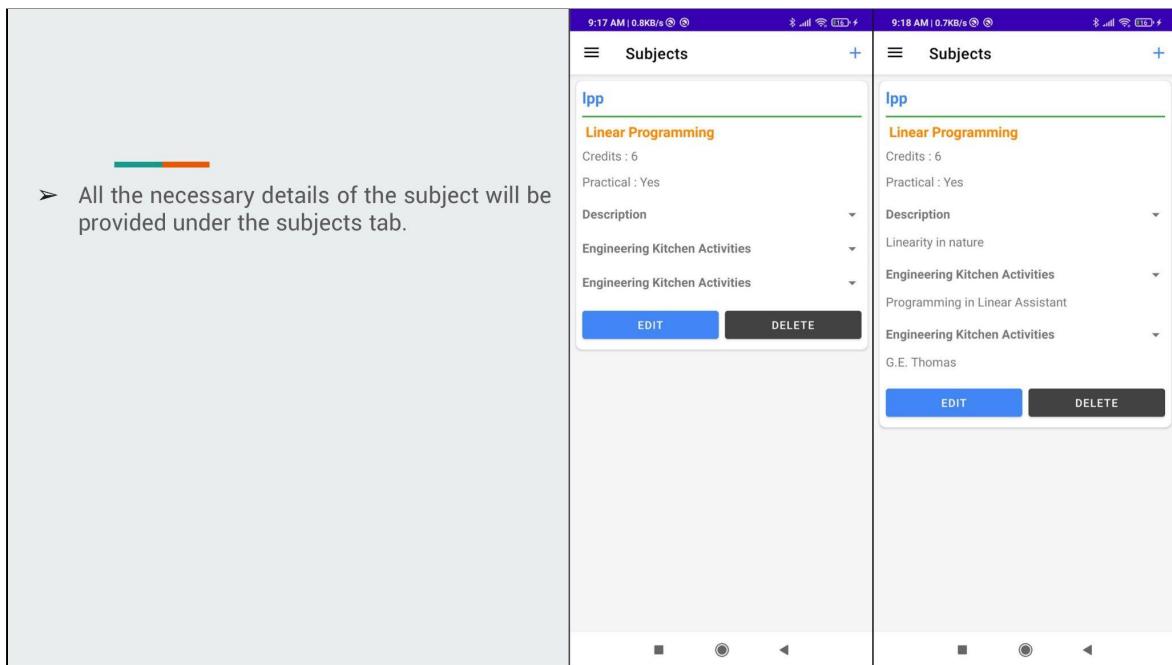
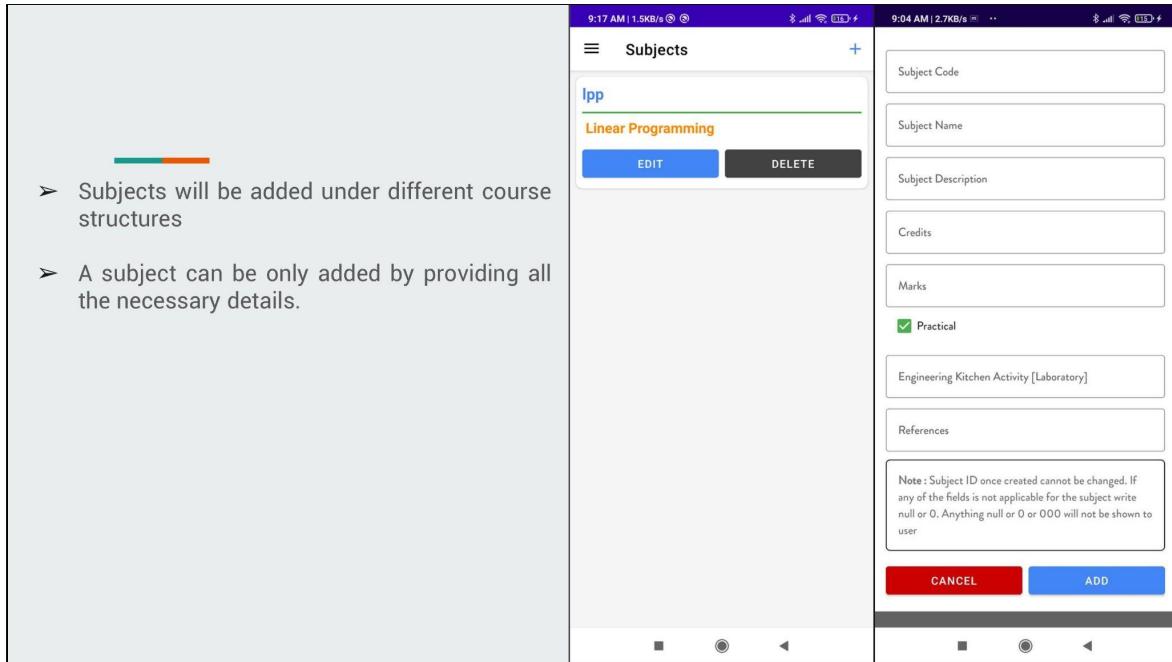




- A specific course structure can be created by providing a course structure ID and course structure name.
- Only Admin will have full access to edit/delete the course structure.



- Details regarding total subjects and credits of a semester will be provided under the specified course structure.



25% 9:04:00

September

SUN 26 Nothing planned. Tap to Create.

MON 27 HRM (10:00 -11:00 am)

TUE 28 Sonam (10:00 -11:00 am)

October 2021

3 - 9 OCT
10 - 16 OCT
17 - 23 OCT
24 - 30 OCT
31 OCT - 6 NOV



6:15:48

MON 1 Nothing planned. Tap to Create.

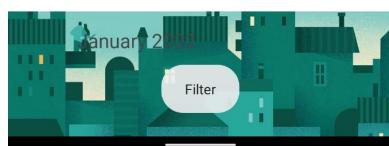
7 - 13 NOV
14 - 20 NOV

MON 15 Nothing planned for today

21 - 27 NOV
28 NOV - 4 DEC

December 2021

5 - 11 DEC
12 - 18 DEC
19 - 25 DEC
26 DEC - 1 JAN



38% 6:16:07

November

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Filter

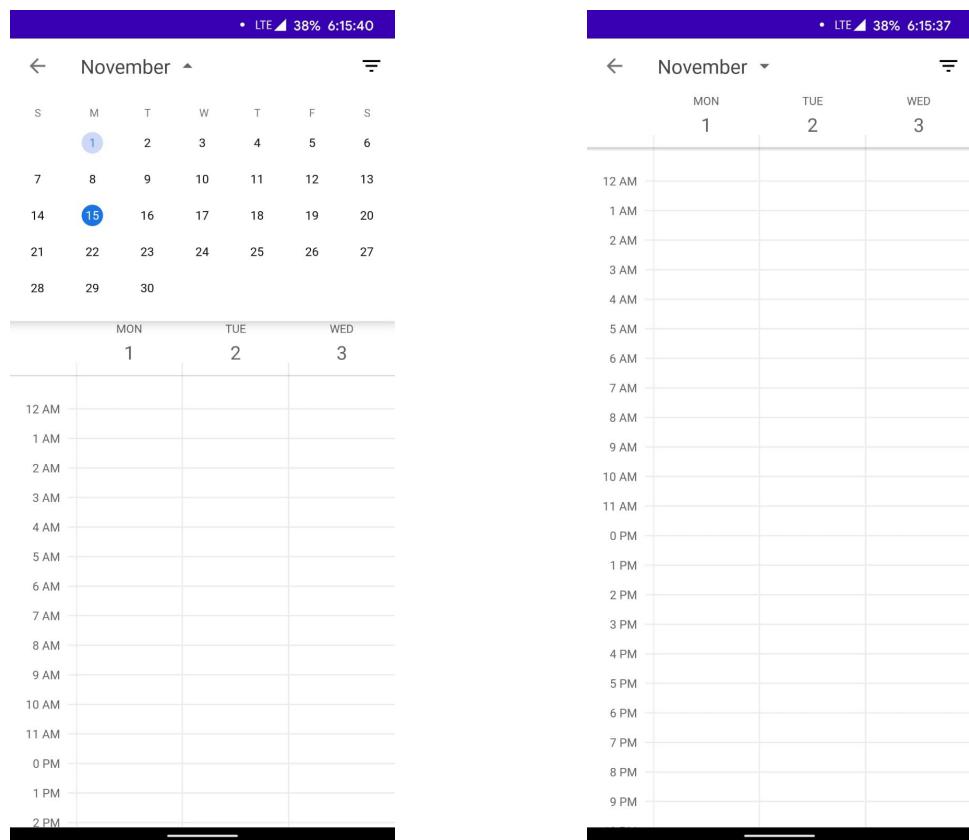
6:15:43

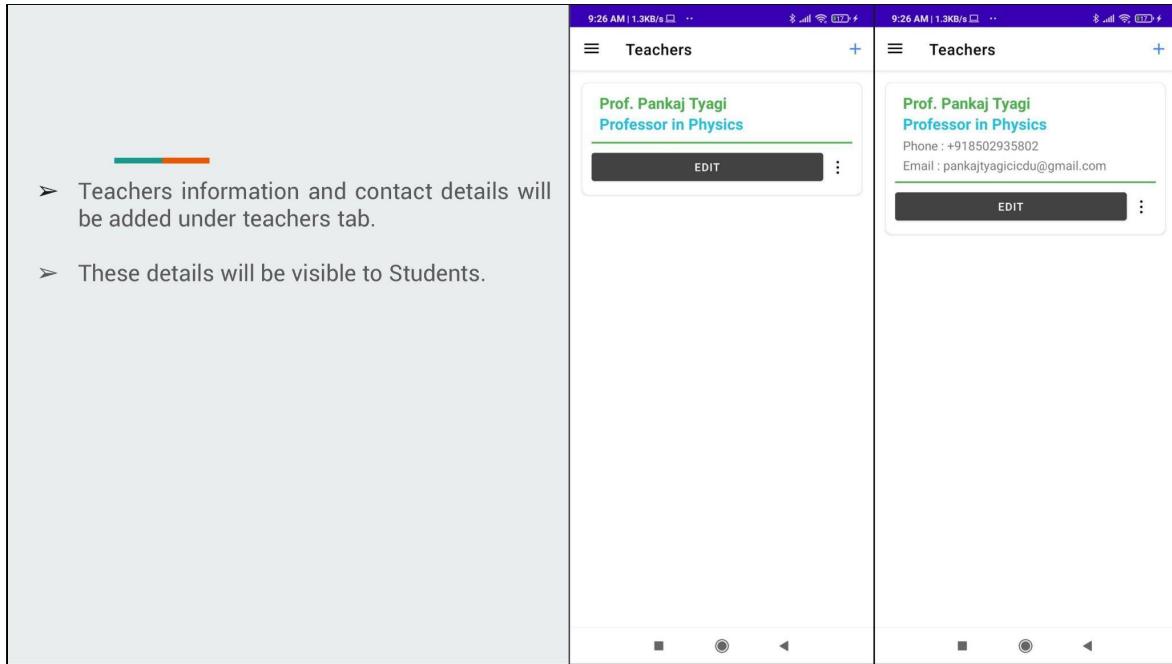
November

S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

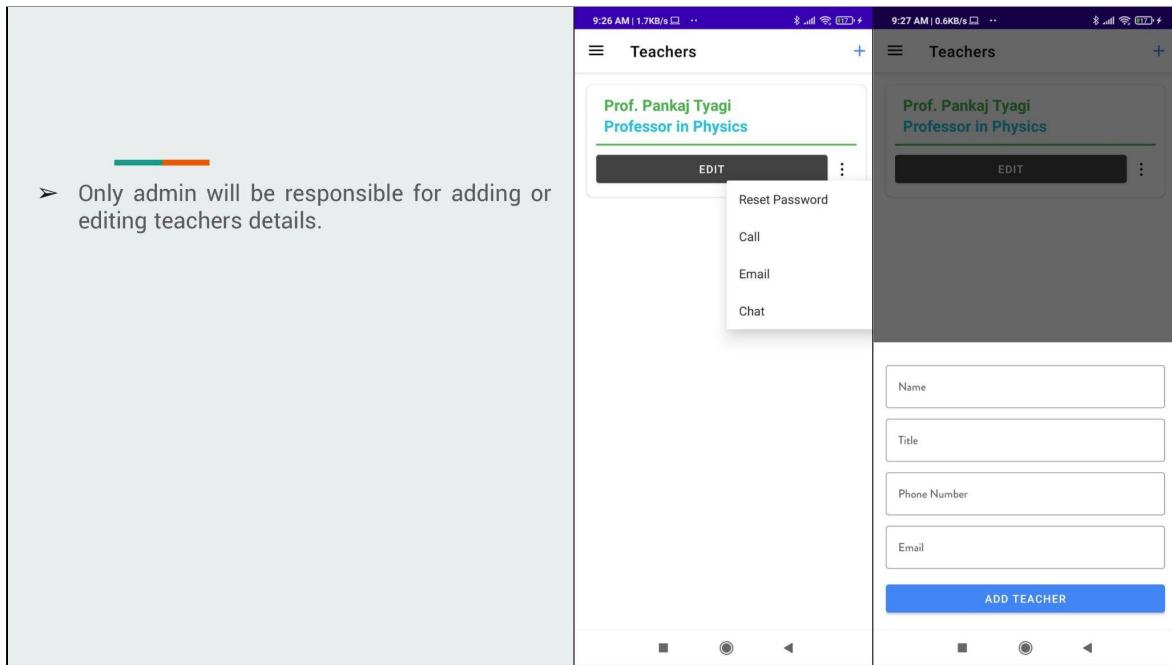
MON	TUE	WED
1	2	3
12 AM		
1 AM		
2 AM		

- Schedule
- Day
- 3 Days
- Week
- Month

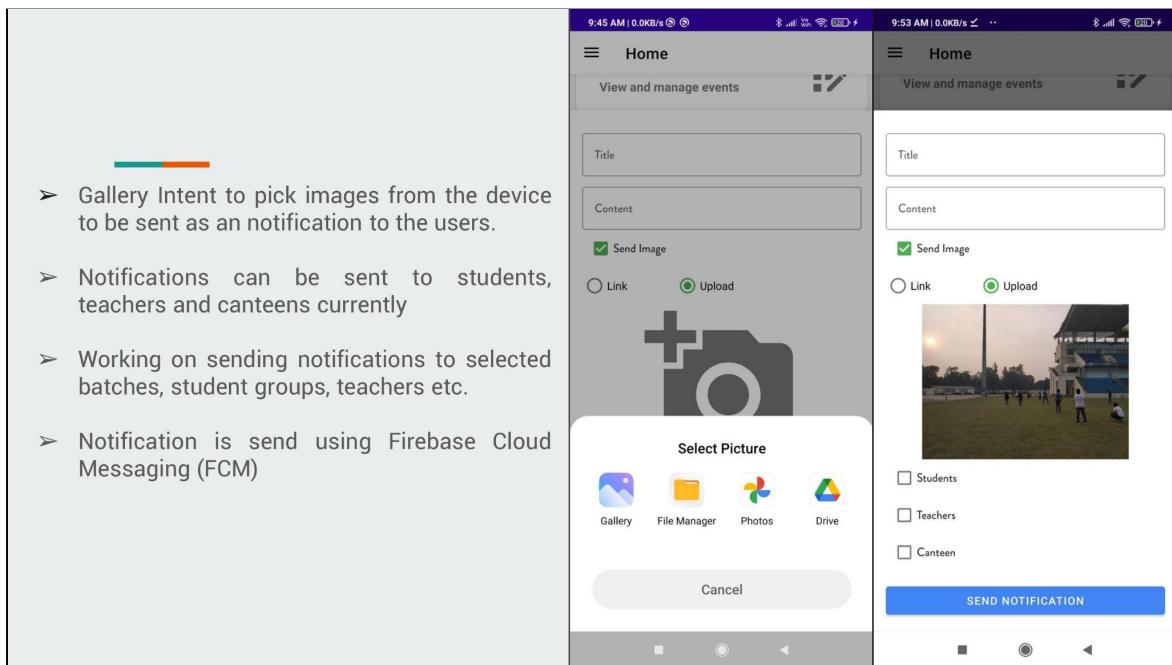
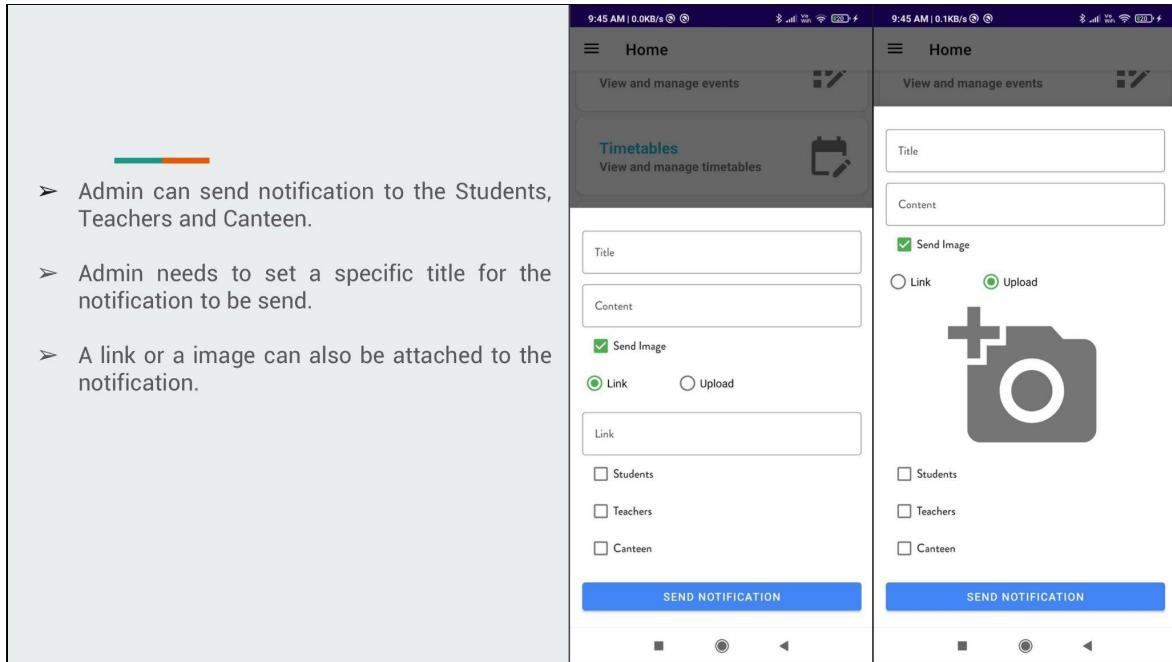


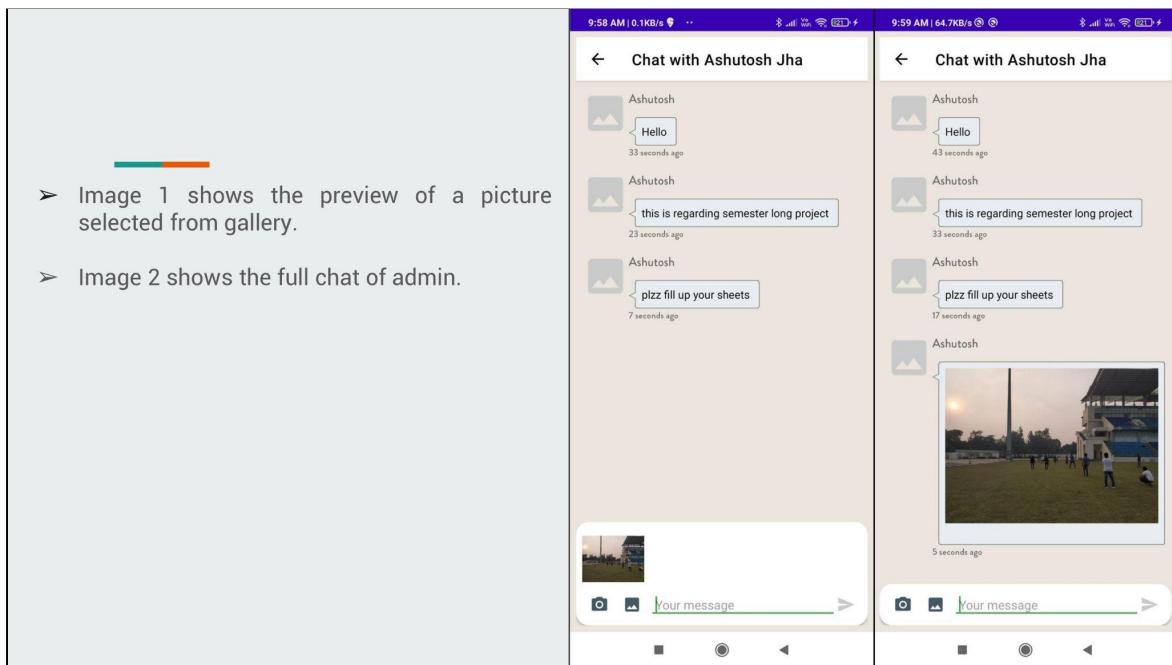
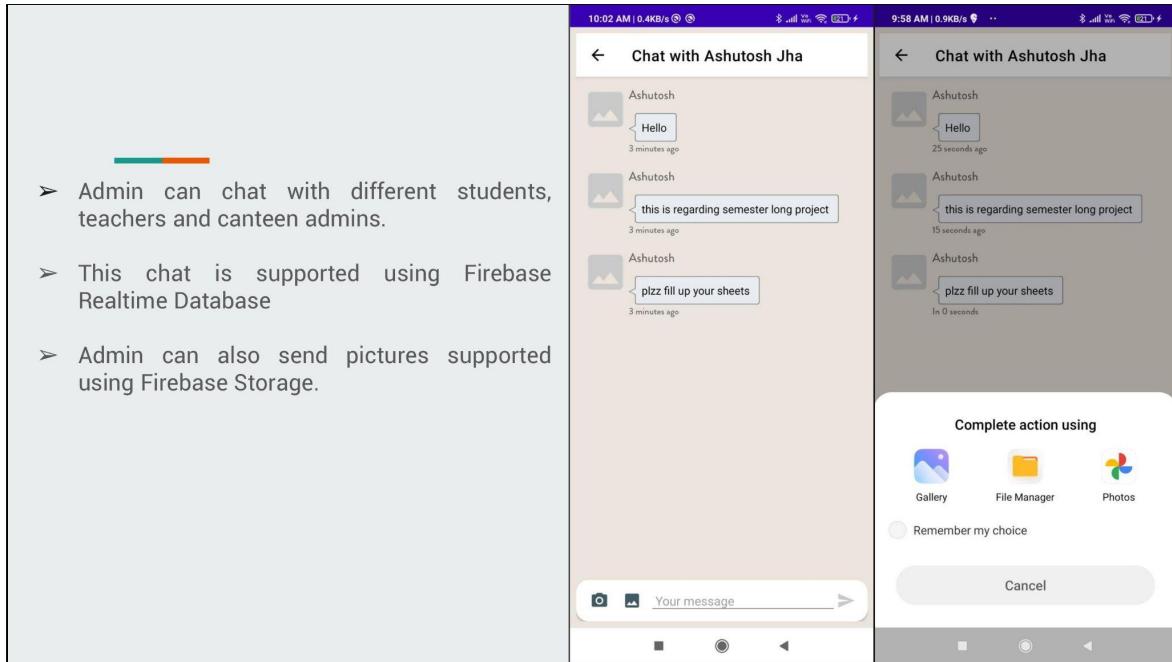


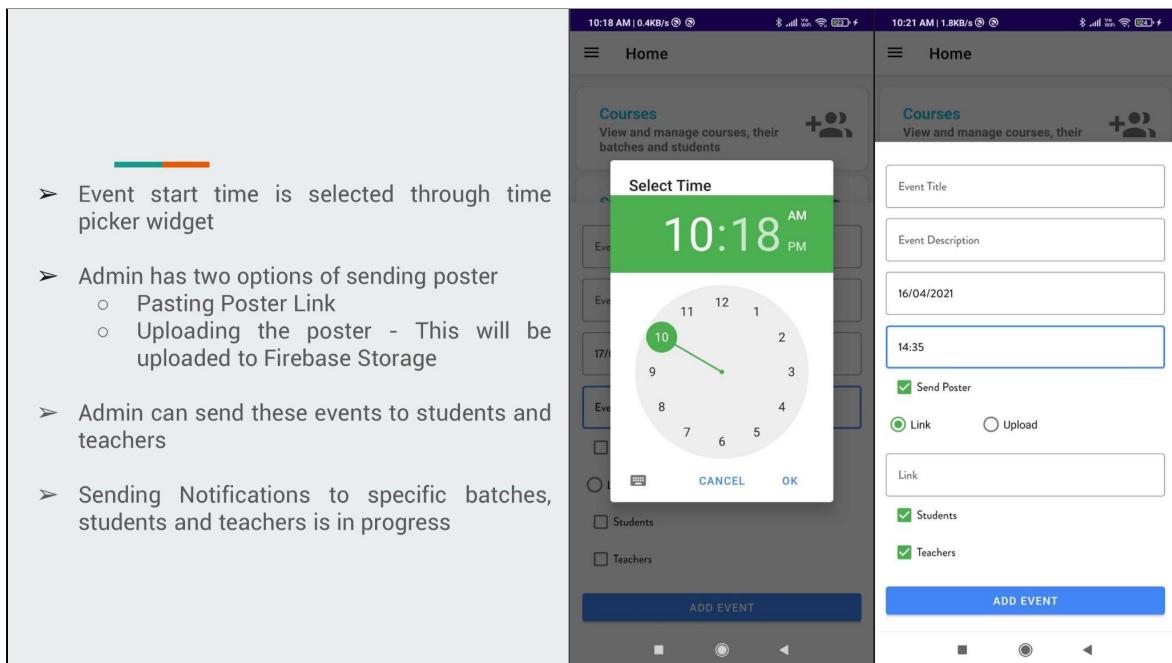
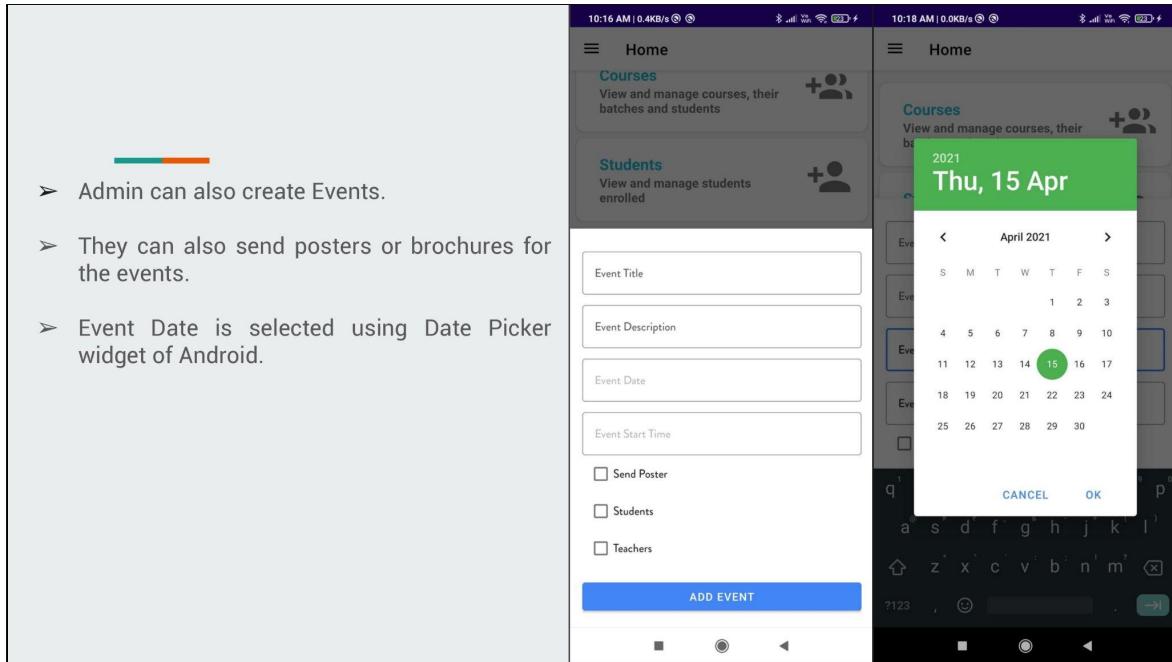
- Teachers information and contact details will be added under teachers tab.
- These details will be visible to Students.

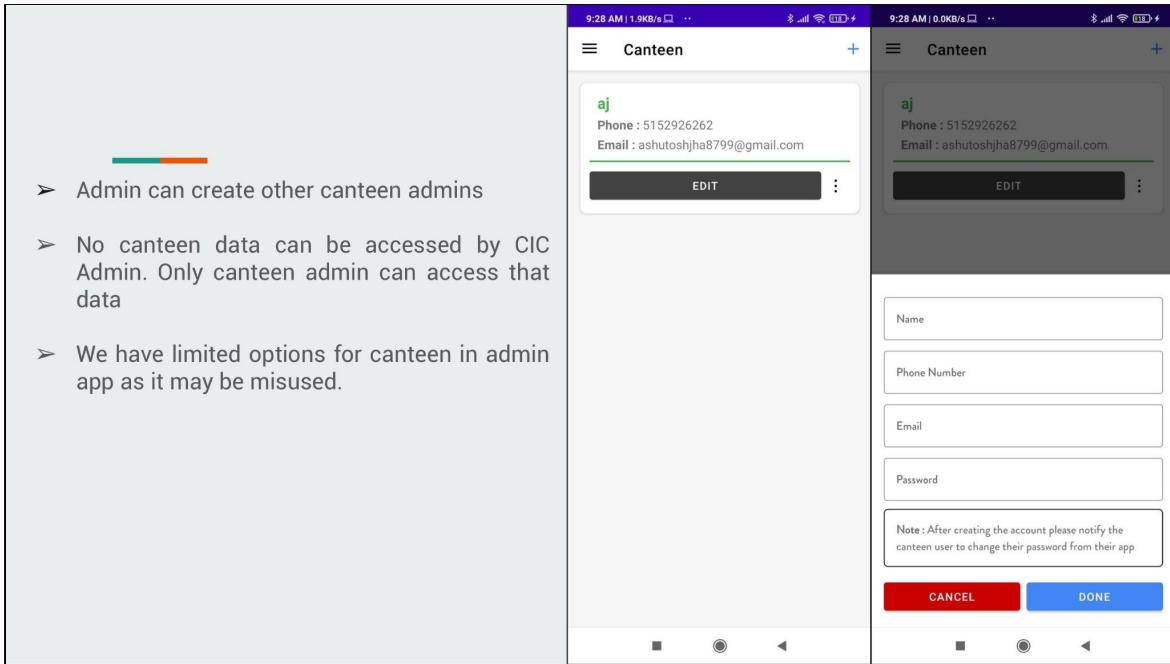


- Only admin will be responsible for adding or editing teachers details.

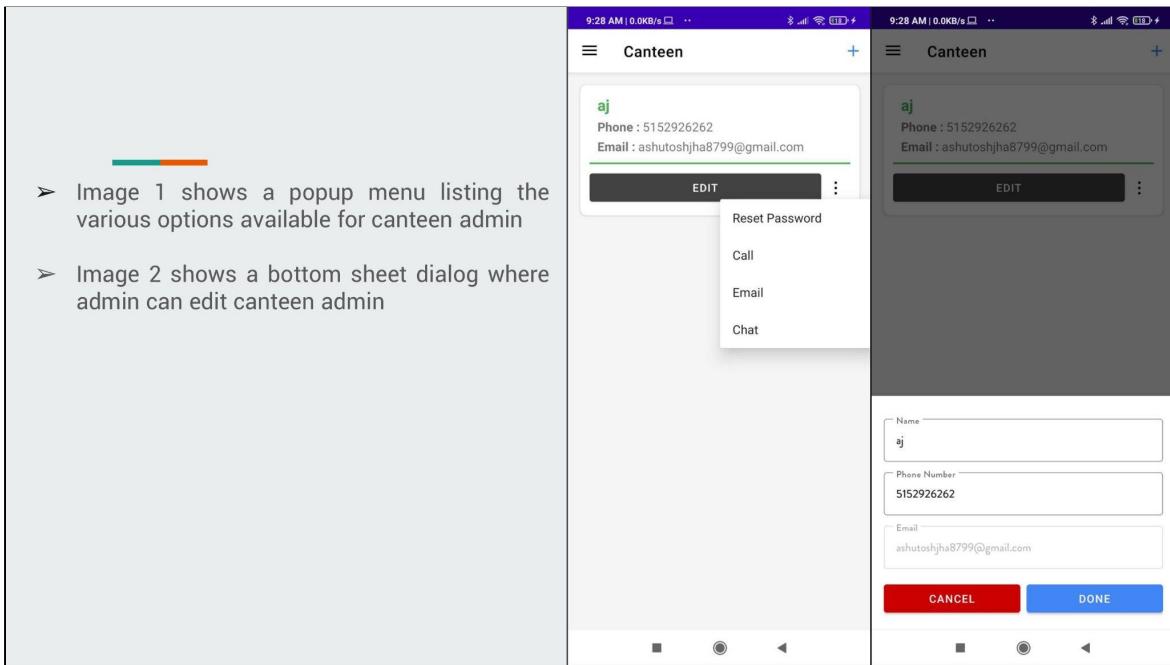


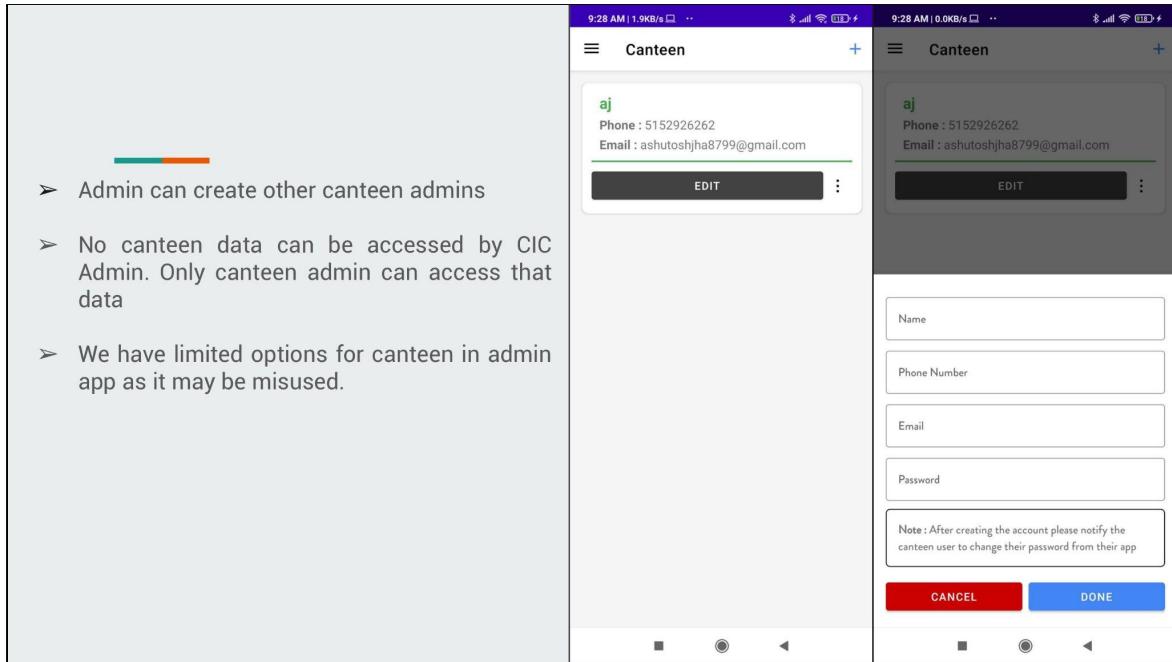




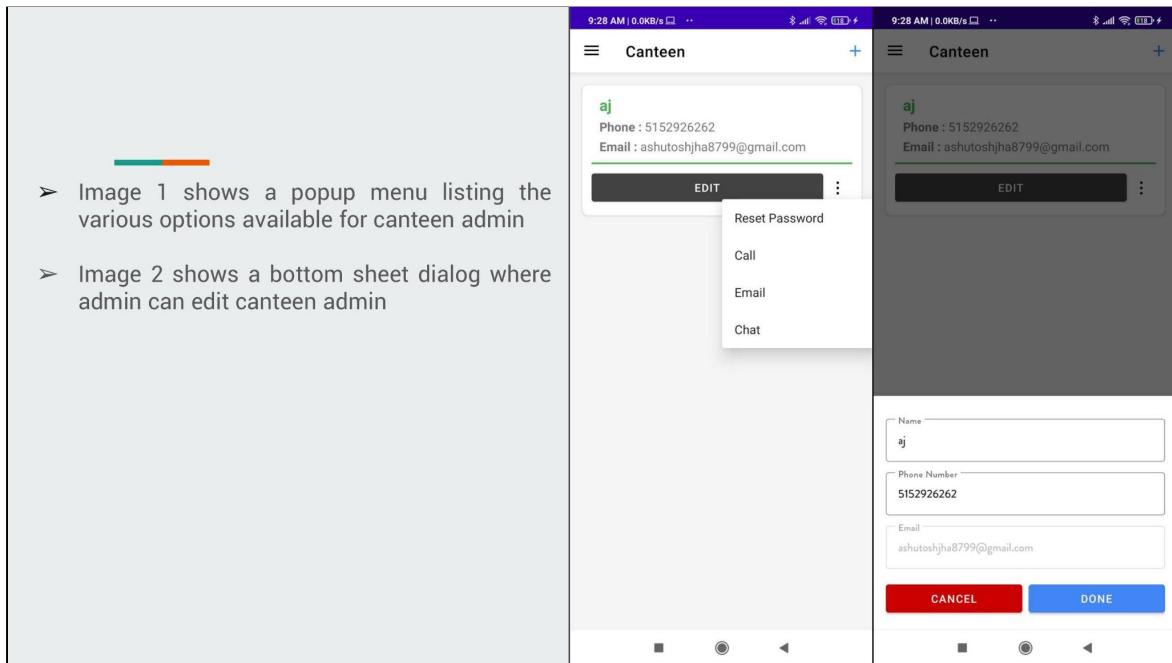


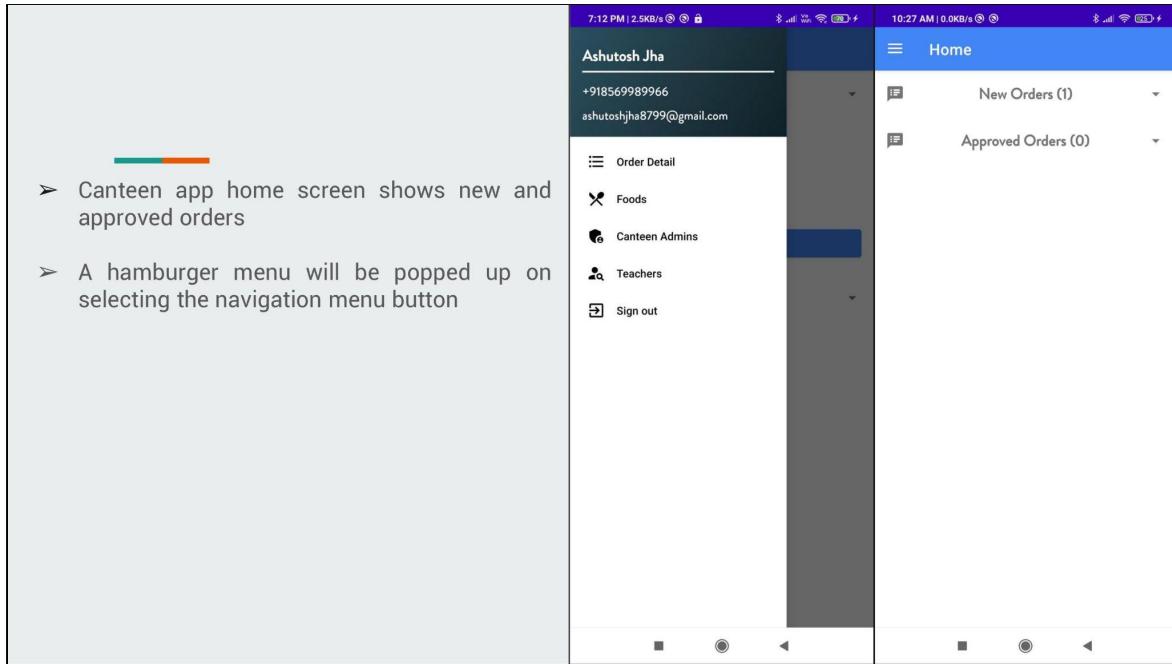
- Admin can create other canteen admins
- No canteen data can be accessed by CIC Admin. Only canteen admin can access that data
- We have limited options for canteen in admin app as it may be misused.



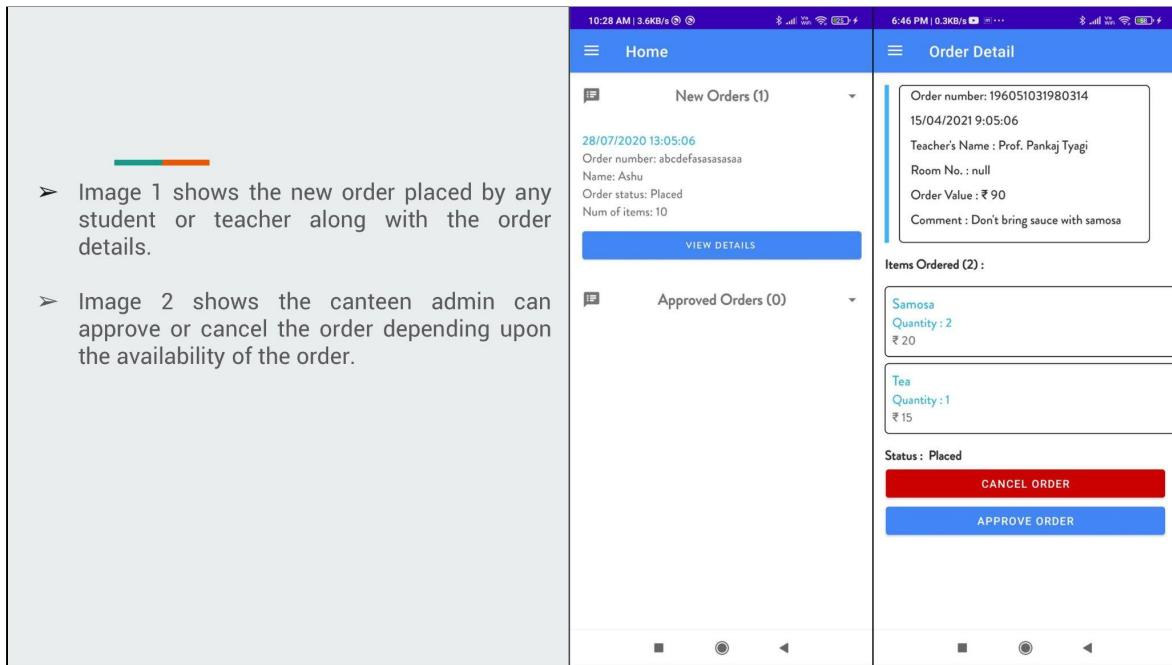


- Admin can create other canteen admins
- No canteen data can be accessed by CIC Admin. Only canteen admin can access that data
- We have limited options for canteen in admin app as it may be misused.

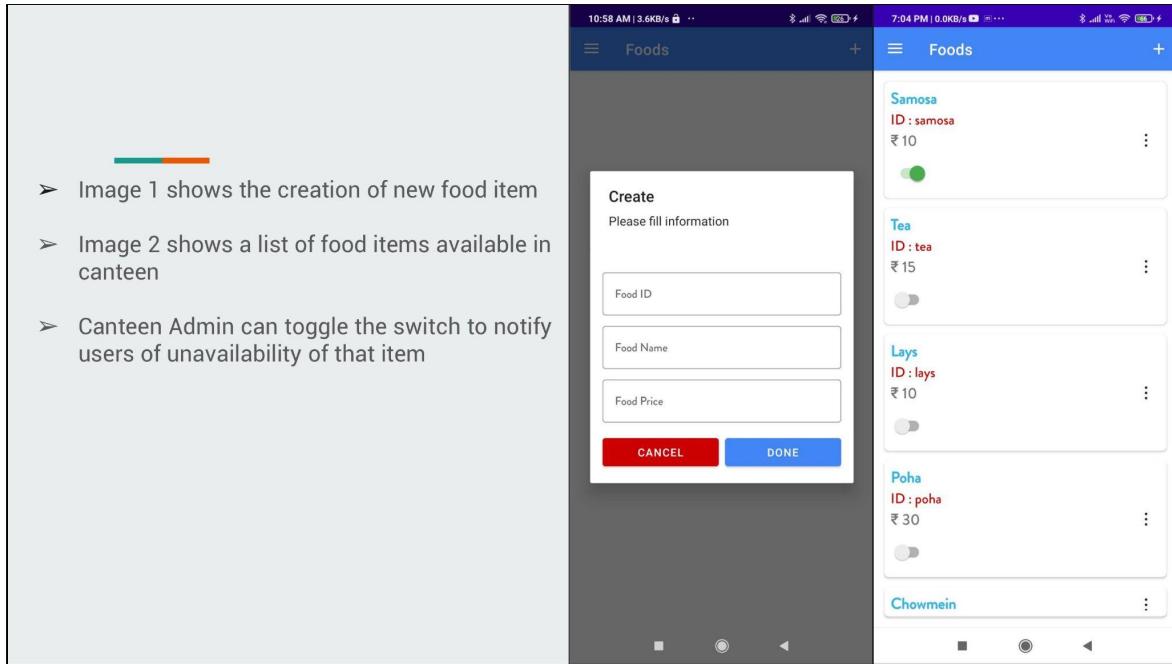




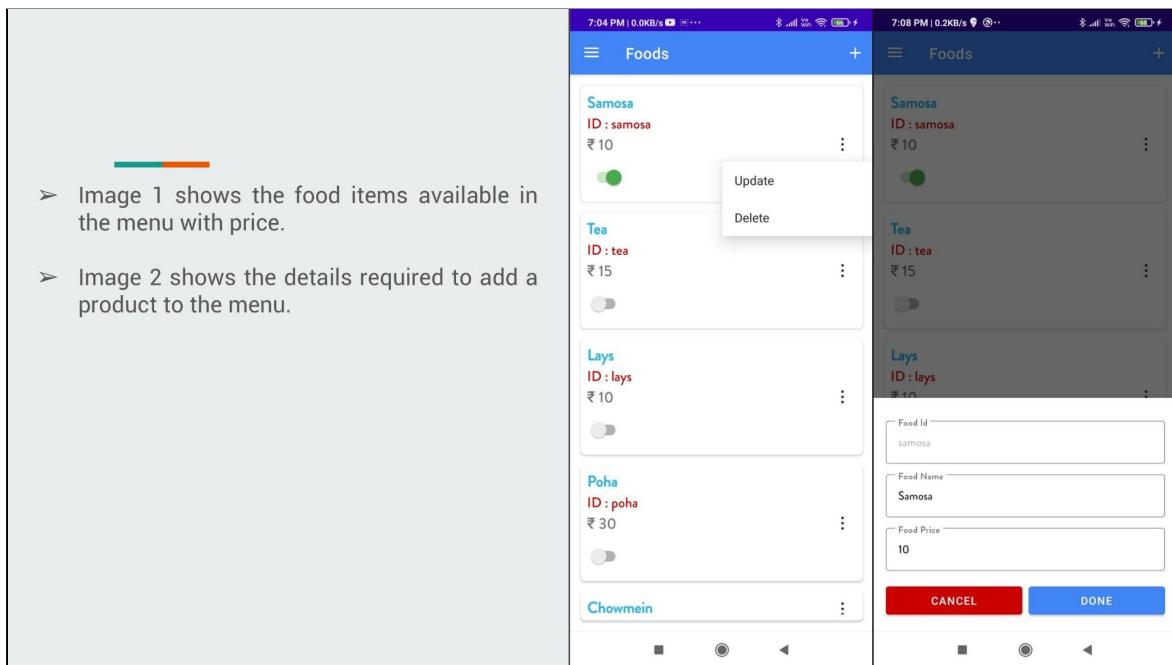
- Canteen app home screen shows new and approved orders
- A hamburger menu will be popped up on selecting the navigation menu button



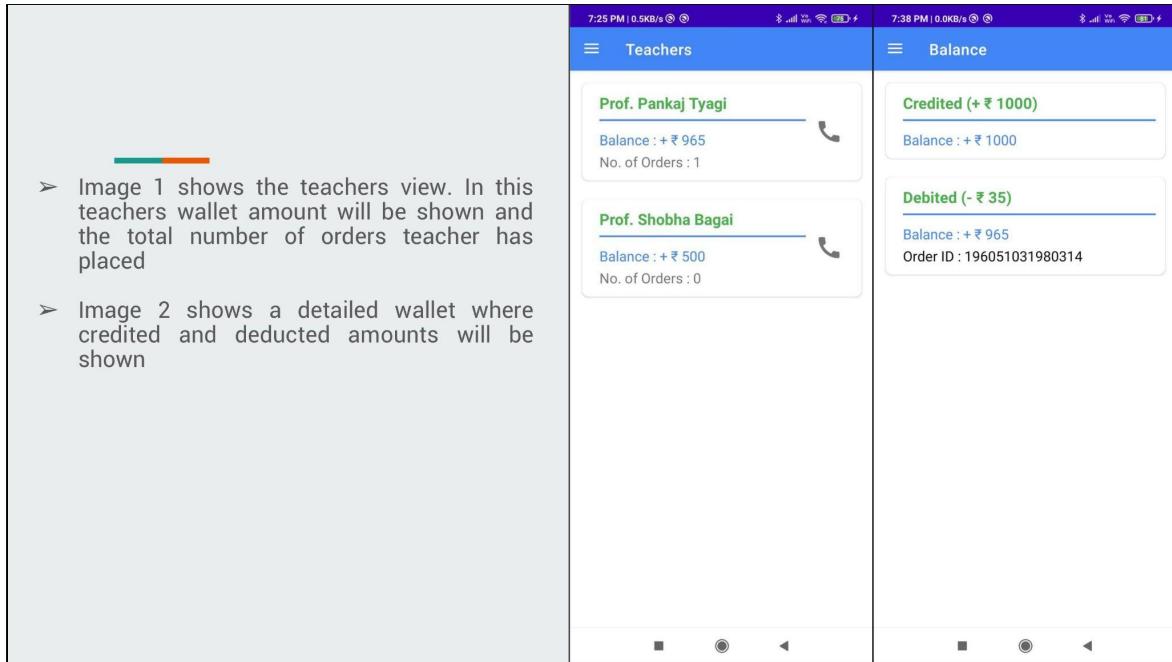
- Image 1 shows the new order placed by any student or teacher along with the order details.
- Image 2 shows the canteen admin can approve or cancel the order depending upon the availability of the order.



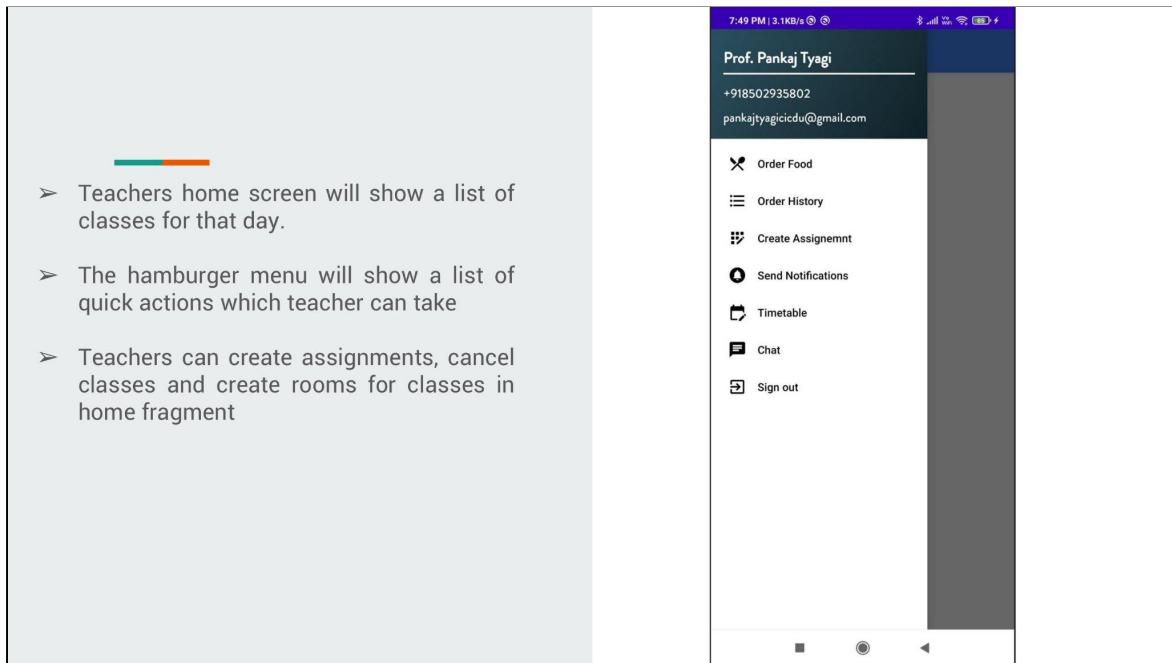
- Image 1 shows the creation of new food item
- Image 2 shows a list of food items available in canteen
- Canteen Admin can toggle the switch to notify users of unavailability of that item



- Image 1 shows the food items available in the menu with price.
- Image 2 shows the details required to add a product to the menu.



- Image 1 shows the teachers view. In this teachers wallet amount will be shown and the total number of orders teacher has placed
- Image 2 shows a detailed wallet where credited and deducted amounts will be shown



- Teachers home screen will show a list of classes for that day.
- The hamburger menu will show a list of quick actions which teacher can take
- Teachers can create assignments, cancel classes and create rooms for classes in home fragment

➤ This is where teachers can order food from canteen. Food Items which are not available will be shown to teacher

➤ Image 2 shows the cart menu for the teacher where teacher can edit their food items

➤ Comments can also be added with each order

➤ Once an order is created, the order amount is deducted from the teacher's wallet

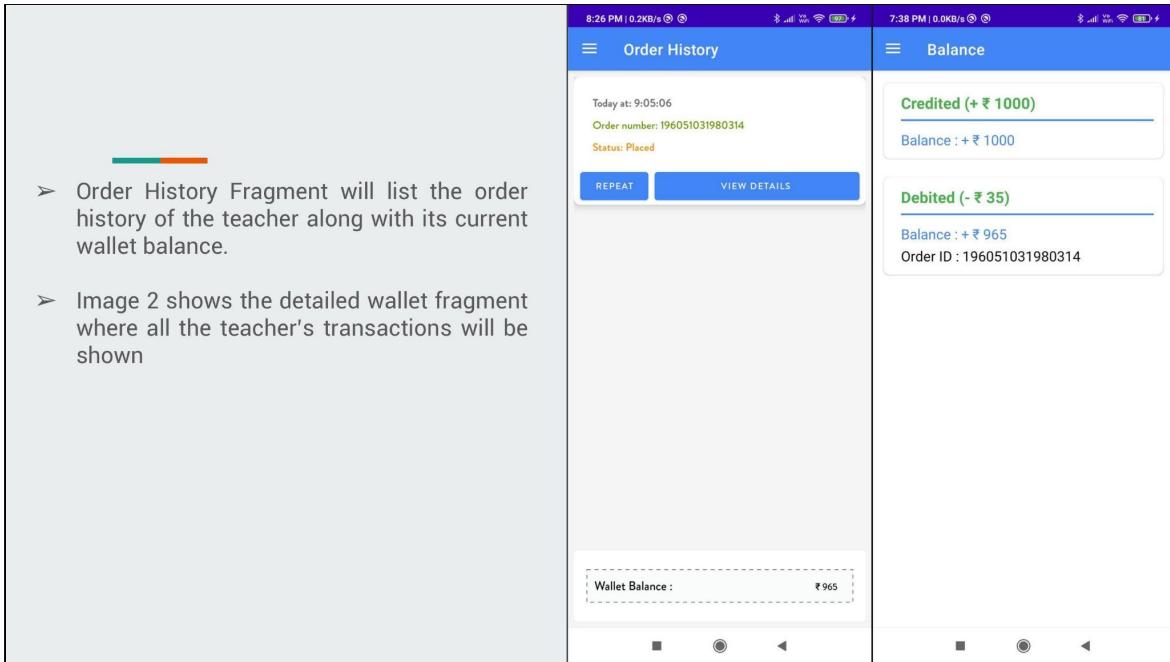
Food Item	Price	Status
Samosa	₹10	Available
Tea	₹15	Available
Lays	₹10	Not Available
Poha	₹30	Not Available
Chowmein	₹50	Not Available

Item	Quantity	Price
Samosa	1	₹20
Tea	1	₹15

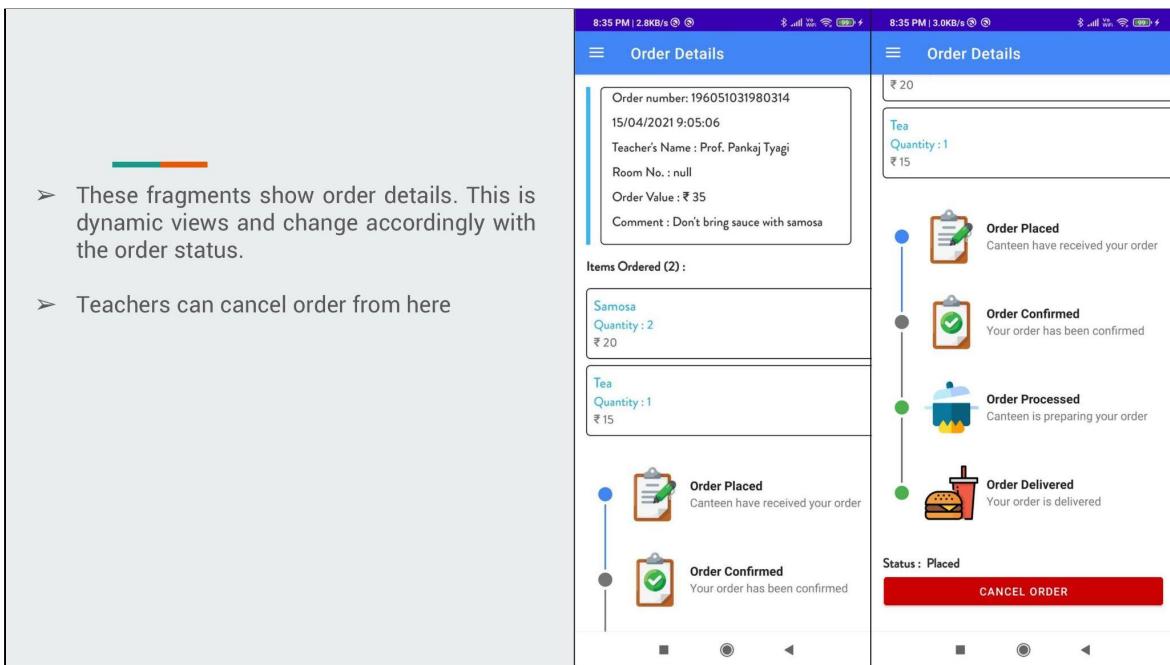
Total Amount : ₹35

Any special requests

PLACE ORDER



- Order History Fragment will list the order history of the teacher along with its current wallet balance.
- Image 2 shows the detailed wallet fragment where all the teacher's transactions will be shown



- These fragments show order details. This is dynamic views and change accordingly with the order status.
- Teachers can cancel order from here

Appendix C: MEMO



Ashutosh Jha <ashutoshjha2285@gmail.com>

Document the findings of the survey

2 messages

Ashutosh Jha <ashutoshjha2285@gmail.com>
To: Amit Katyal <amitkatyal097@gmail.com>

14 August 2021

Dear Amit Katyal

Please structure the questions properly in your survey and try to expand the reach of your study. Also, document the findings of your survey using different mathematical models.

Regards,
Ashutosh Jha

Amit Katyal <amitkatyal097@gmail.com>
To: Ashutosh Jha <ashutoshjha2285@gmail.com>

14 August 2021

Dear Ashutosh Jha

I have noted down the changes requested by you and I am working on it as of now. It is expected to be fulfilled within 2 days. Additionally, I am adding a few bonus questions. I expect that they will help us understand the survey results in a better manner.

Regarding the reach of the survey, I have already circulated the survey among some other groups of our target audience and the findings are expected to be collected and compiled by this week. The mathematical models and the charts required for the study will take additional 2 days.

Please expect the work to be completed within our schedule.

Thanks & Regards
Amit



Ashutosh Jha <ashutoshjha2285@gmail.com>

Update the Data Flow Diagrams

2 messages

Ashutosh Jha <ashutoshjha2285@gmail.com>
To: Shubham Chauhan <shubhamchauhan0899@gmail.com>

28 August 2021

Dear Shubham

As per the guidelines given by professor Anjani Kumar Verma, our data flow diagram needs some update as per the discussion. There are no relations between entities and the processes are also wrong. However, the level 3 diagram is correct but you need to update the level 1 and level 2 diagrams. Moreover, add some more necessary input fields in the blueprint as specified by the requirement analyst in the meeting.

Report back to me in 1 week.

Regards,
Ashutosh Jha

Shubham Chauhan <shubhamchauhan0899@gmail.com>
To: Ashutosh Jha <ashutoshjha2285@gmail.com>

28 August 2021

Dear Ashutosh Jha

Sure, I will update the data flow diagram as per the guidelines we discussed in the previous meeting and will coordinate with the requirement analyst to update the blueprint. Will get back to you within a few days with the updated data.

Thank You



Ashutosh Jha <ashutoshjha2285@gmail.com>

Integrate the four applications

2 messages

Ashutosh Jha <ashutoshjha2285@gmail.com>
To: TUSHAR SETHI <sethitushar2000@gmail.com>

11 September 2021

Dear Tushar Sethi

We have developed four applications for College Connect but the findings of the survey and the feasibility study indicates and concludes that all the applications should be combined. Therefore, I request you to please compile all these applications and their data and manage the flow of the app accordingly. So, in short, there will only be one apk file at the end of your work.

Regards,
Ashutosh Jha

TUSHAR SETHI <sethitushar2000@gmail.com>
To: Ashutosh Jha <ashutoshjha2285@gmail.com>

11 September 2021

Dear Ashutosh Jha

I have already started working on combining all the apps to a single one. I will be able to complete the task and send you the final apk within a week.

Regards
Tushar Sethi



Ashutosh Jha <ashutoshjha2285@gmail.com>

Update the documentation and do Stress Testing

2 messages

Ashutosh Jha <ashutoshjha2285@gmail.com>
To: sachinrathore028@gmail.com

30 September 2021

Dear Sachin Rathore
You have done a nice work in developing the unit and instrumental test. However, the documentation of these tests is blurry. You have not specified any details regarding the alpha test, beta test, black-box test, white-box test, etc. So, please categorize your test results according to the specifications explained to you. Moreover, I request you to please do a stress test and give me the correct info regarding the number of users and queries the app can handle per minute.

Regards,
Ashutosh Jha

Sachin <sachinrathore028@gmail.com>
To: Ashutosh Jha <ashutoshjha2285@gmail.com>

30 September 2021

Dear Ashutosh Jha

I have been working on the documentation of the tests and there are a few remaining tests to be documented.
I will be sending you the final test report as soon as the tests are completed within a week.

Thank You