Software Engineering project

Result Management System

JAN 2021 - April 2021

Purpose

Created BY:

1. DWARKA PRASAD BAIRWA - 1801061
2. AMAR KUMAR - 1801016
3. DEEPAK KUMAR YADAV - 1801048

The purpose of developing this “Result Management System” is to automate student result management. This application will greatly simplify and speed up the result preparation and management process.The application will manage the information about various students enrolled in this course in different years, the marks obtained by the various students in various subjects in different semesters.

Scope

The application will manage the information about various students enrolled in this course in different years, the subjects offered during different semesters, and the various students' marks in multiple subjects in different semesters.The application will greatly simplify and speed up the result preparation and management process.

Project Introduction

Existing System:

The manual method of students’ academic result processing was found to be tedious, especially when carried out for a large number of students, this makes the entire process time-consuming and error prone.

A existing system has following disadvantages:

- In the manual system students need to bear a lot of time and cost.
- Compiling the result in the manual system is a tedious process.
- Lack of transparency in the present system.
- In the existing system, students have no proper idea of marks in different assessments for their self-evaluation. Learning from mistakes is the best way for the growth of a student.
- Existing System creates more pressure on the examination board of the college.

Proposed System:

The system presents a single platform that will be used to manage the processing of all examination records within the institution. Admin gives the privileges to the Teacher to

update/add the student's result of the respective course.

- ***Student Registration***: Admin can register a student.
- ***Faculty Registration***: Admin can register a faculty.
- ***Result***: Students can get the Updated result of all the exams (quiz,viva,mid semester and end semester) via this Application.
- ***Authentication***: Admin will send secure login credentials to all students & faculty using a python script.
- ***Marks of all internal assessments***: Students are able to view their marks of quiz, viva, projects, mid-sem,end-sem,lab assignments etc. for self evaluation.
- ***Administration Access***: The administration has all the privileges to add/update any details about students & faculty.

Advantage of System

Student :

- Time Saving: Students will be able to see the result whenever they want .
- Easy to Use: The app interface is designed such that it'll be easy to use.
- All in one service: Students are able to view all previous semester results as well as current semester results.

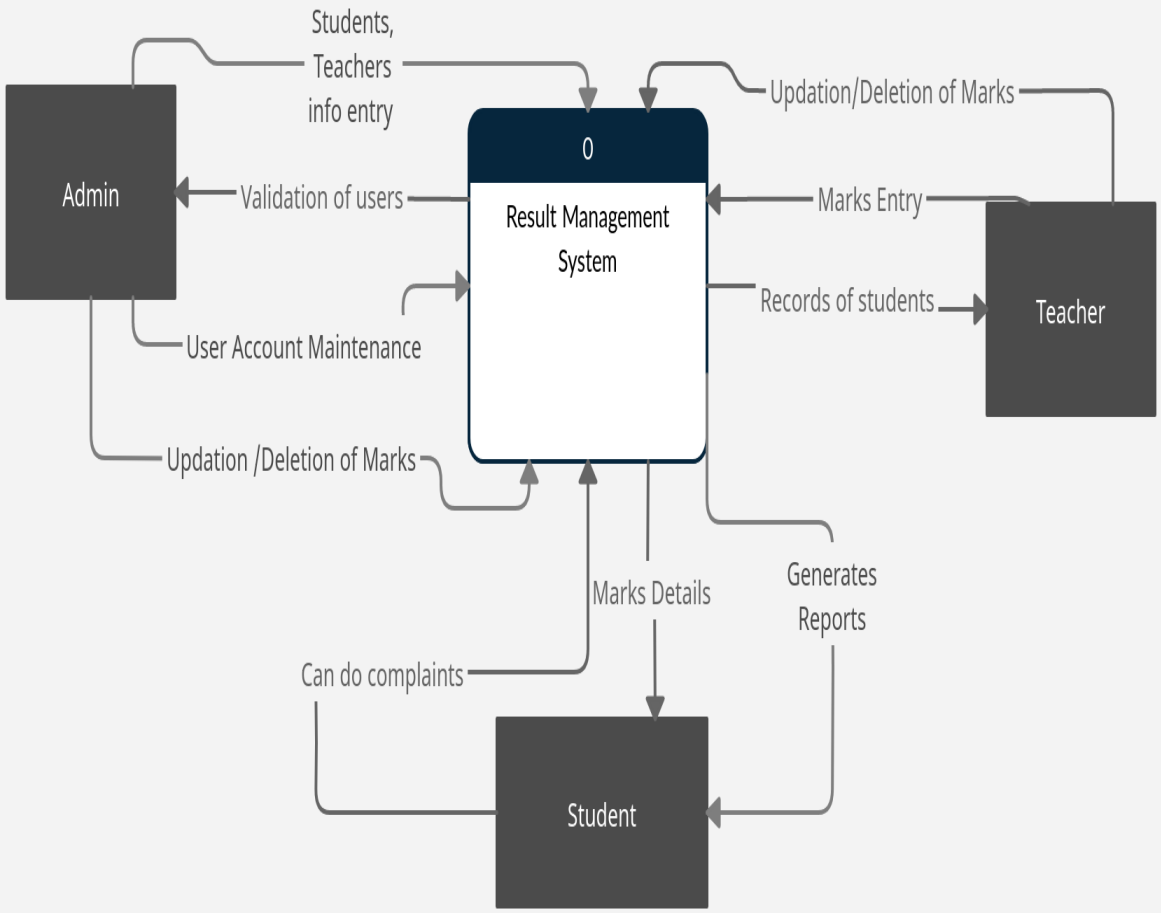
Faculty :

- Faculty have to just upload the marks/grade and all the calculation of SPI & CPI will be done at the backend.
- View the list of students assigned to them(Automatically show the students list with same semester)

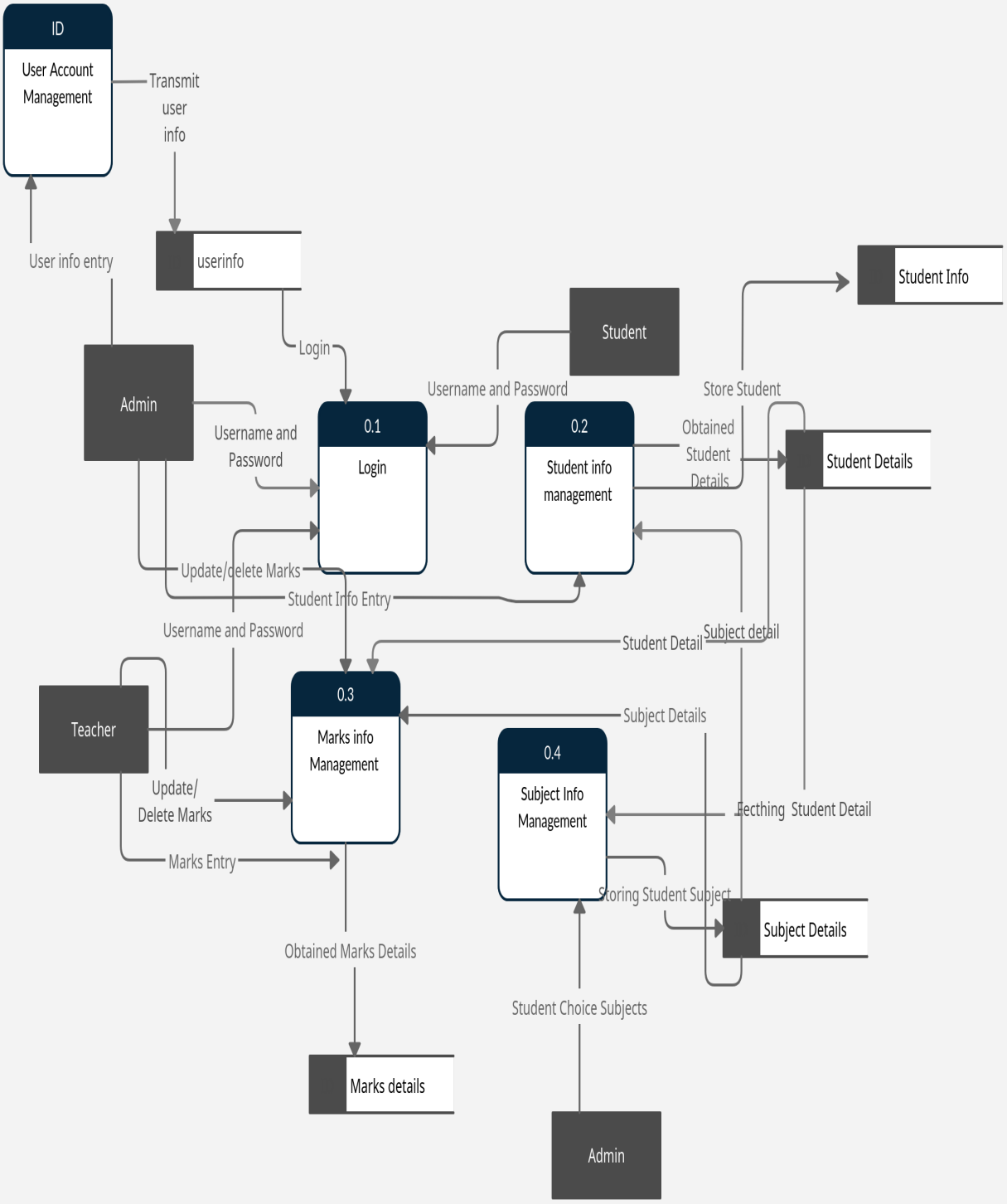
Overview of System

Data Flow Diagram

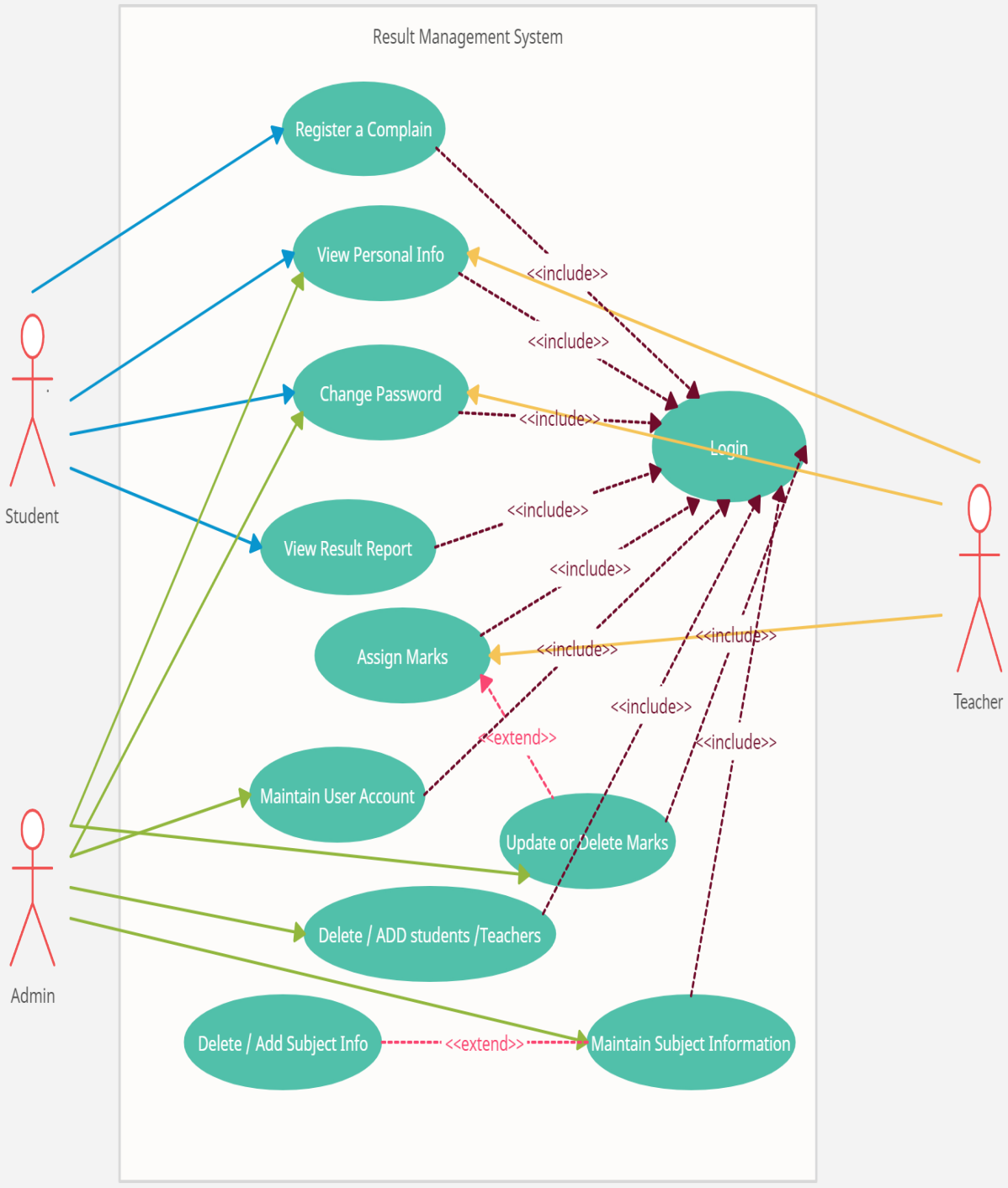
⇨ Context Diagram (Level - 0 DFD) :



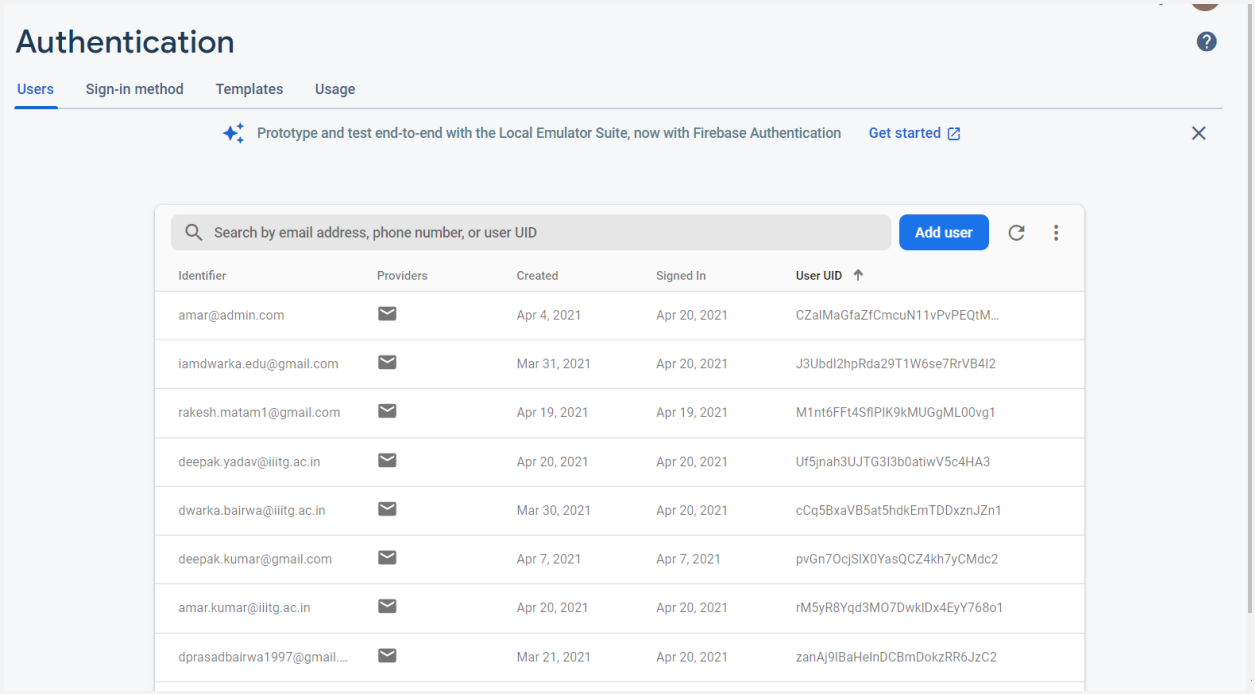
⇨ *Overview Diagram* (*Level -1 DFD*) :

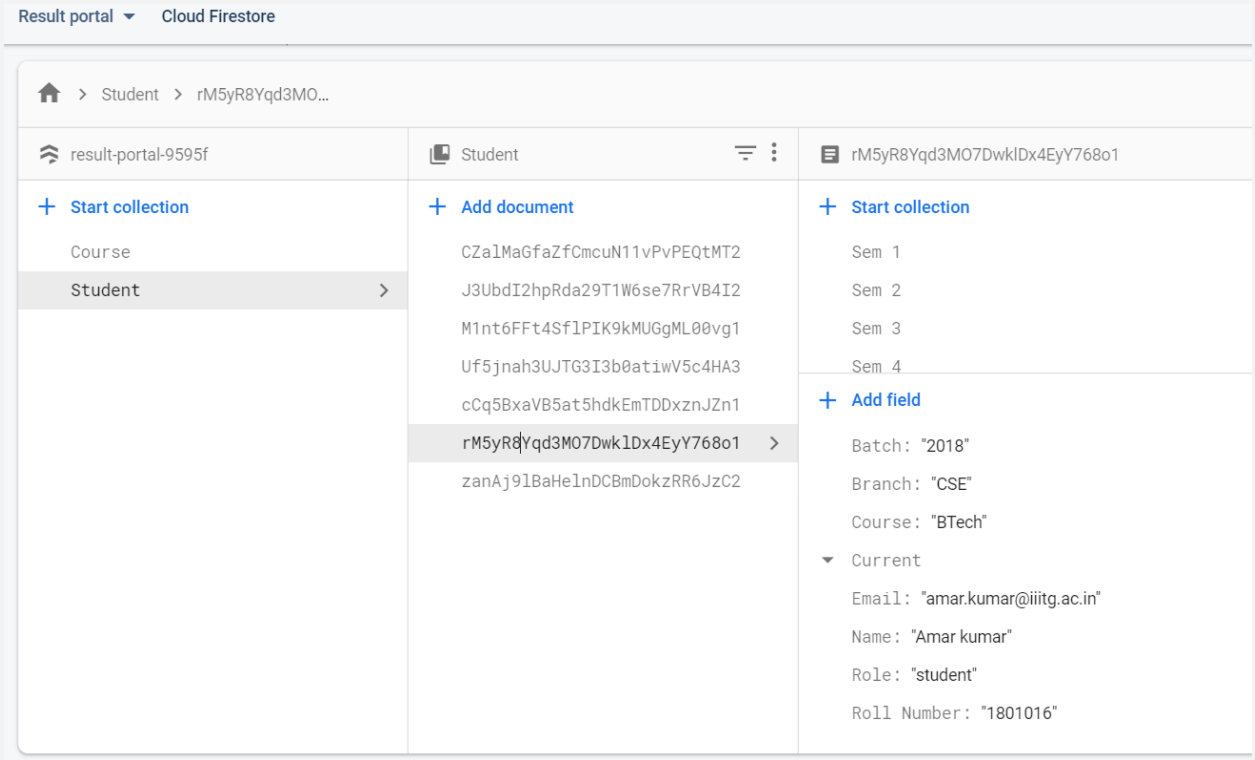


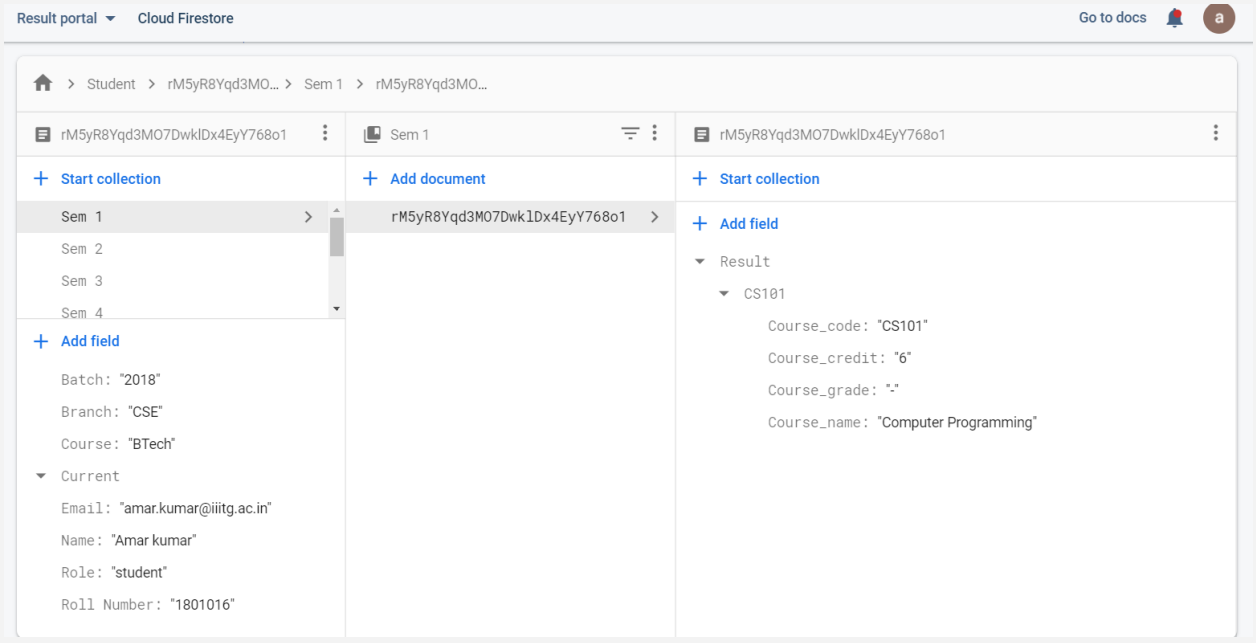
★ *Use Case Diagram* ★

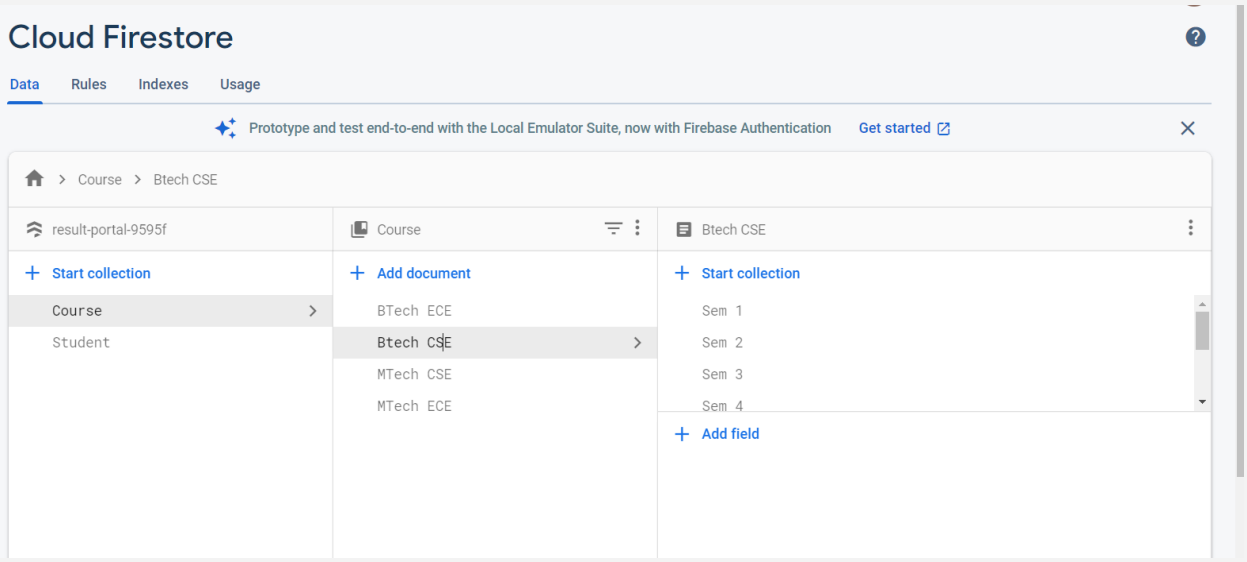


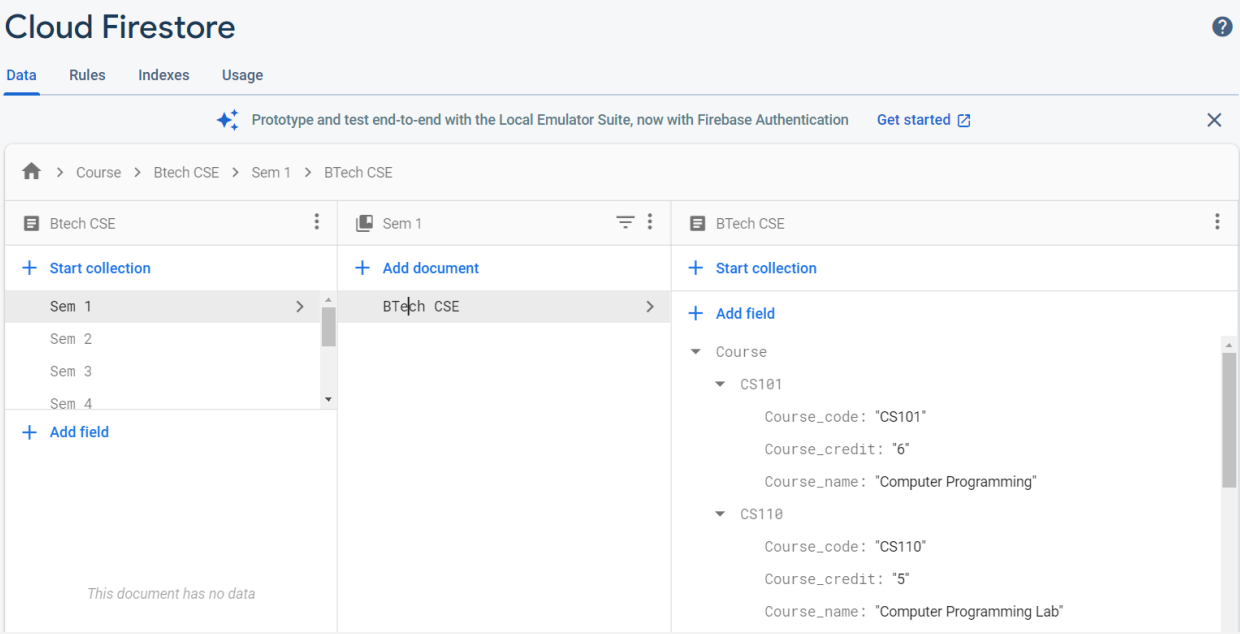
★ *Databases Structure*★


Authentication
Users
Sign-in method
Templates
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
amar@admin.com
Apr 4, 2021
Apr 20, 2021
CZalMaGfaZfCmcuN11vPvPEQtM...
iamdwarka.edu@gmail.com
Mar 31, 2021
Apr 20, 2021
J3UbdI2hpRda29T1W6se7RrVB4I2
rakesh.matam1@gmail.com
Apr 19, 2021
Apr 19, 2021
M1nt6FFt4SflPIK9kMUGgML00vg1
deepak.yadav@iiitg.ac.in
Apr 20, 2021
Apr 20, 2021
Uf5jnah3UJTG3I3b0atiwV5c4HA3
dwarka.bairwa@iiitg.ac.in
Mar 30, 2021
Apr 20, 2021
cCq5BxaVB5at5hdkEmTDDxznJZn1
deepak.kumar@gmail.com
Apr 7, 2021
Apr 7, 2021
pvGn7OcjSIX0YasQCZ4kh7yCMdc2
amar.kumar@iiitg.ac.in
Apr 20, 2021
Apr 20, 2021
rM5yR8Yqd3MO7DwklDx4EyY768o1
dprasadbairwa1997@gmail....
Mar 21, 2021
Apr 20, 2021
zanAj9lBaHelnDCBmDokzRR6JzC2


Result portal
Cloud Firestore
Student
rM5yR8Yqd3MO...
result-portal-9595f
Start collection
Course
Student
Student
Add document
CZalMaGfaZfCmcuN11vPvPEQtMT2
J3UbdI2hpRda29T1W6se7RrVB4I2
M1nt6FFt4SflPIK9kMUGgML00vg1
Uf5jnah3UJTG3I3b0atiwV5c4HA3
cCq5BxaVB5at5hdkEmTDDxznJZn1
rM5yR8Yqd3MO7DwklDx4EyY768o1
zanAj9lBaHelnDCBmDokzRR6JzC2
rM5yR8Yqd3MO7DwklDx4EyY768o1
Start collection
Sem 1
Sem 2
Sem 3
Sem 4
Add field
Batch: "2018"
Branch: "CSE"
Course: "BTech"
Current
Email: "amar.kumar@iiitg.ac.in"
Name: "Amar kumar"
Role: "student"
Roll Number: "1801016"


Result portal
Cloud Firestore
Go to docs
Student > rM5yR8Yqd3MO... > Sem 1 > rM5yR8Yqd3MO...
rM5yR8Yqd3MO7DwklDx4EyY768o1
Sem 1
rM5yR8Yqd3MO7DwklDx4EyY768o1
Start collection
Add document
Start collection
Sem 1
Sem 2
Sem 3
Sem 4
rM5yR8Yqd3MO7DwklDx4EyY768o1
Add field
Result
CS101
Course_code: "CS101"
Course_credit: "6"
Course_grade: "-"
Course_name: "Computer Programming"
Add field
Batch: "2018"
Branch: "CSE"
Course: "BTech"
Current
Email: "amar.kumar@iiitg.ac.in"
Name: "Amar kumar"
Role: "student"
Roll Number: "1801016"


Cloud Firestore
Data
Rules
Indexes
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
Course > Btech CSE
result-portal-9595f
Course
Btech CSE
Start collection
Add document
Start collection
Course
Student
BTech ECE
Btech CSE
MTech CSE
MTech ECE
Sem 1
Sem 2
Sem 3
Sem 4
Add field

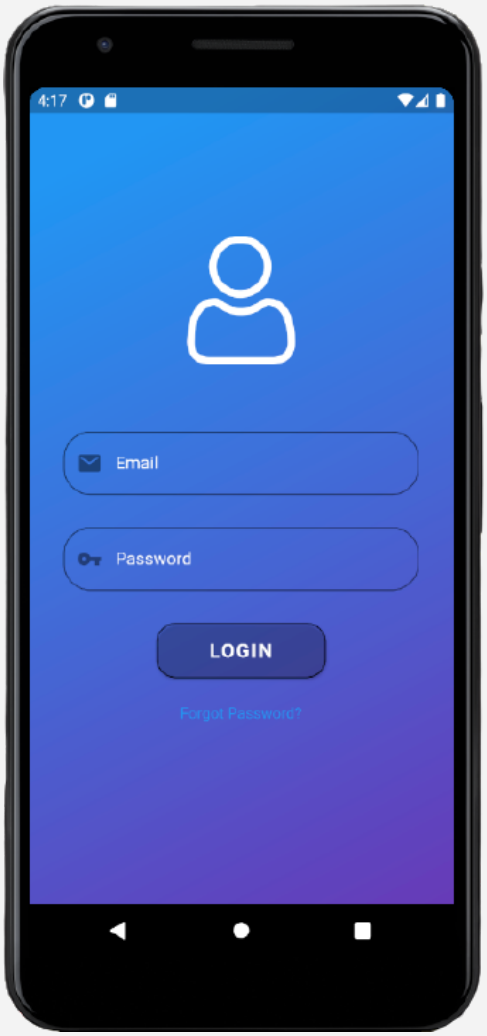

Cloud Firestore
Data
Rules
Indexes
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
Course > Btech CSE > Sem 1 > BTech CSE
Btech CSE
Sem 1
BTech CSE
Start collection
Add document
Start collection
Sem 1
Sem 2
Sem 3
Sem 4
BTech CSE
Add field
Course
CS101
Course_code: "CS101"
Course_credit: "6"
Course_name: "Computer Programming"
CS110
Course_code: "CS110"
Course_credit: "5"
Course_name: "Computer Programming Lab"
Add field
This document has no data

Requirements

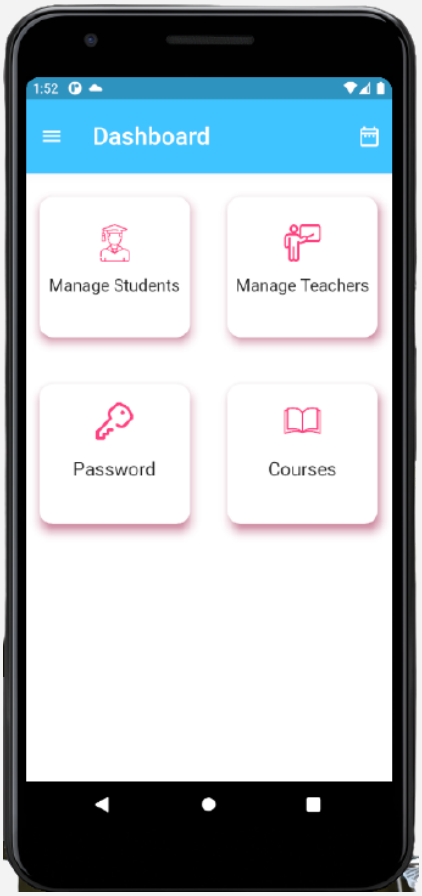
Functional Requirements:

➢ *Login Page for Admin/Faculty/Student :*

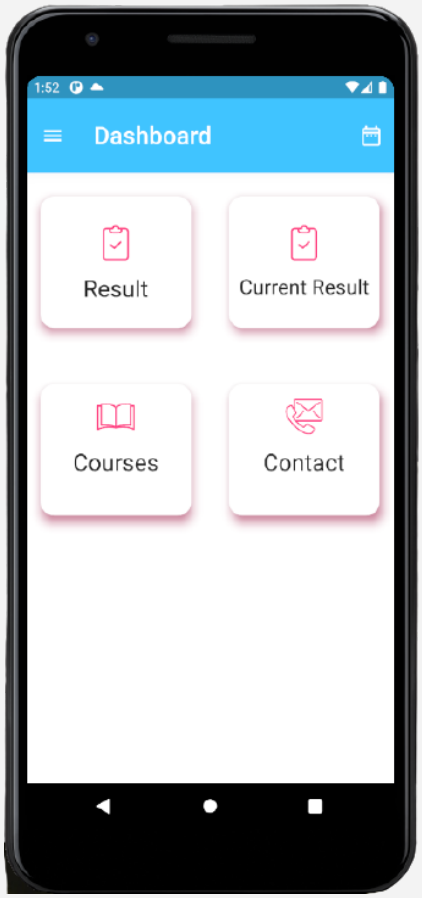
In our System there are three types of users : Admin , Student and Faculty. So, we have a Single login page for all users and based on their role in the database they will be redirected towards their respective Dashboard.



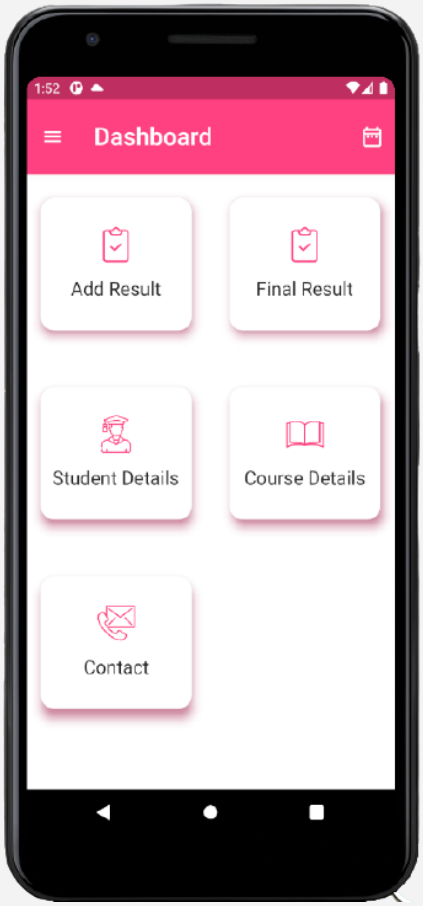
1.Admin Dashboard



2.Student Dashboard

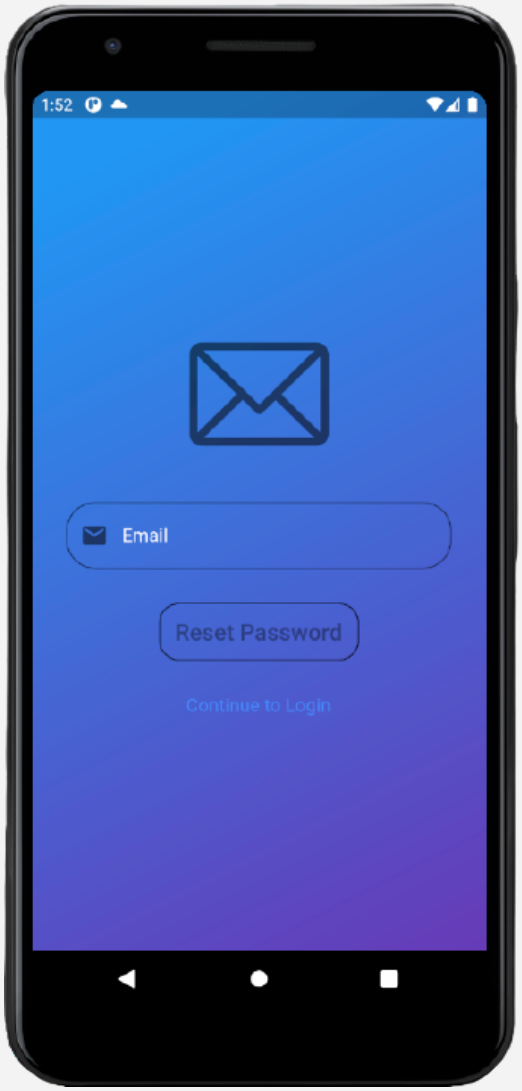


3.Faculty Dashboard



❏ *Additional Feature on Login Page:-*

If a user forgets his/her password then my clicking on Forgot Password user will get a reset password link on his/her registered email id.



➢ *Admin can add accounts of Faculty and students or delete accounts of students or teachers.*

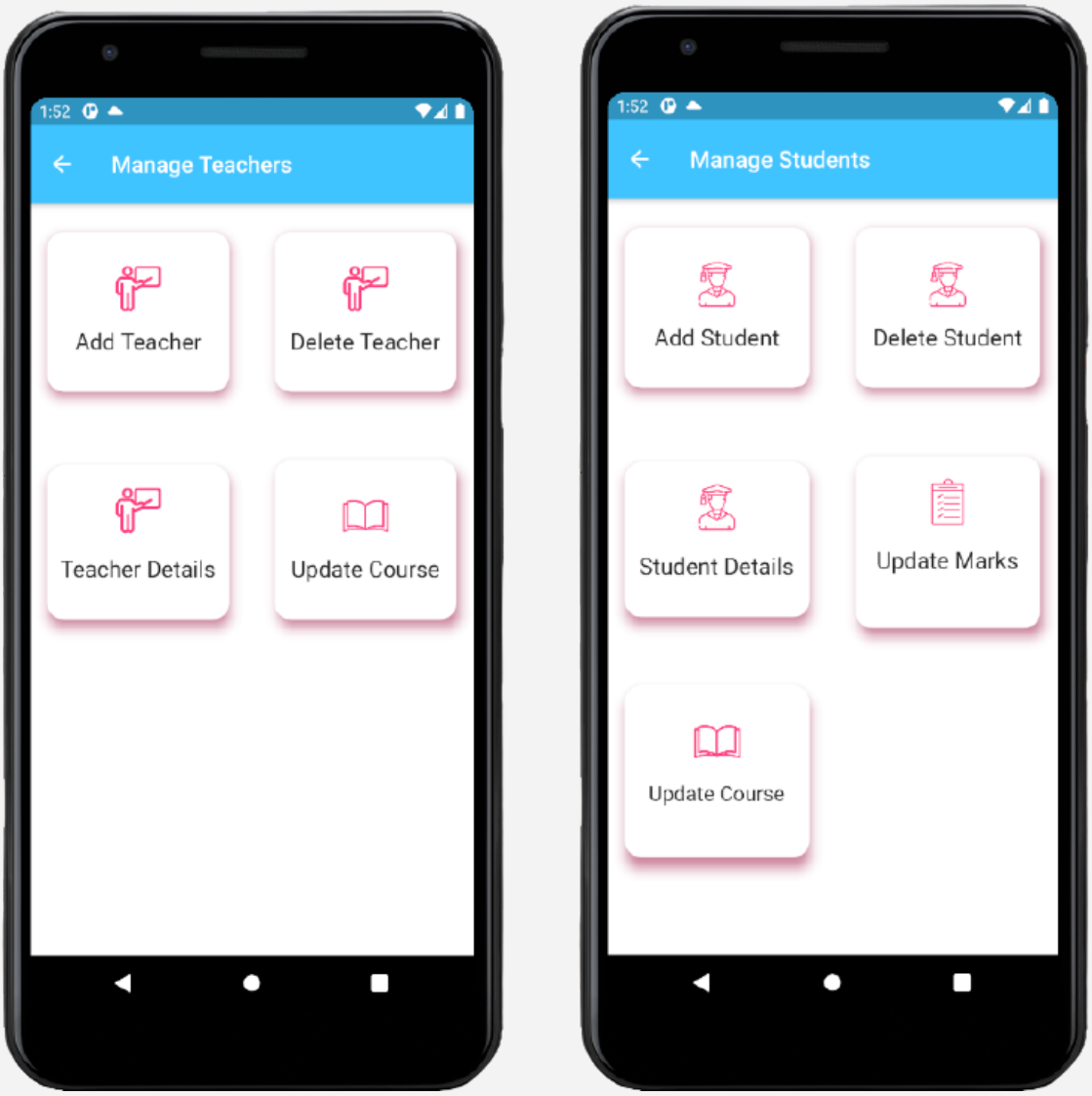
❏ Manage Student UI:-

- *Add Student* : Admin can add a single student by using a form or add multiple students by using excel sheet of predefined format provided in app.
- *Delete Student :* Just by entering a roll no. of a student, admin can delete the account of that student or upload an excel sheet to delete the account of multiple students.

❏ Manage Teacher UI:-

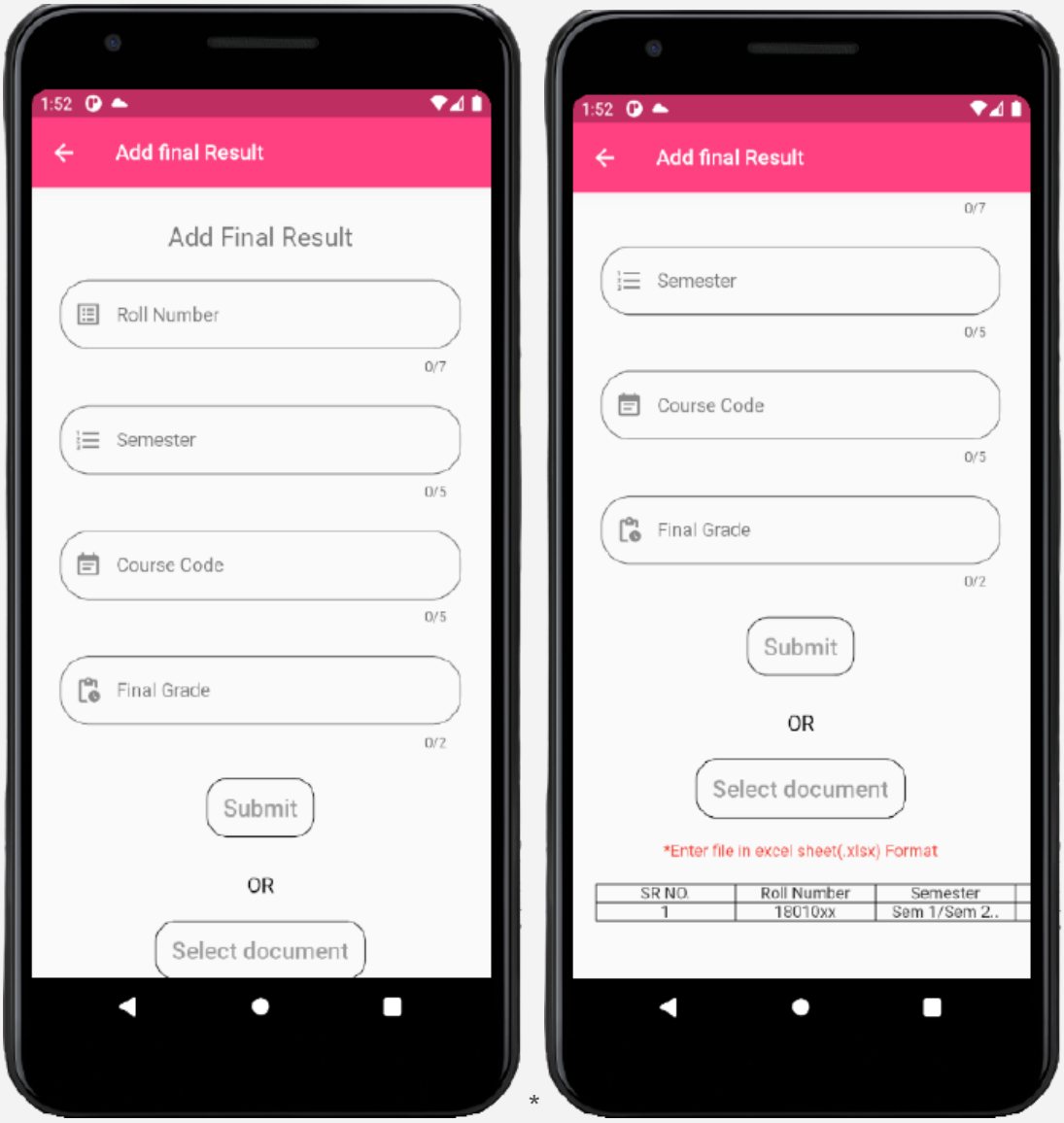
- *Add Teacher* : Admin can add a single faculty by using a form or add multiple faculties by using an excel sheet of predefined format provided in the app.

- *Delete Teacher* :Just by entering the Teacher ID of a teacher , the admin can delete the account of that teacher or upload an excel sheet to delete the account of multiple Teachers.



➢ *The Admin and Faculty will also have permission to update the marks of the Student.*

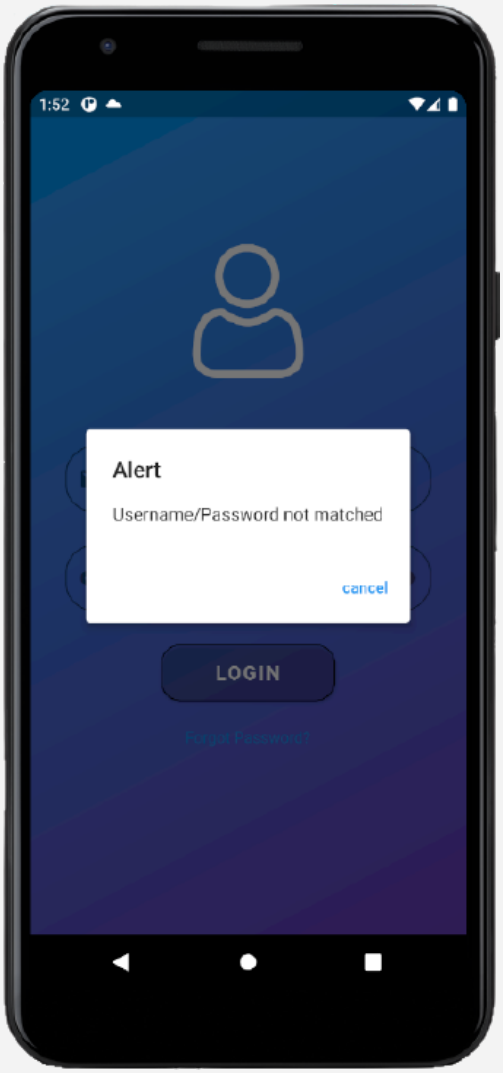
Admin and faculty can update the final marks of students by using Roll No.,Semester No., Course Code & Grade obtained by Student using a form provided in the app or by using excel sheet of predefined format.



*

➢ *Students/Faculty must have a valid User ID and password to login .*

Students & Faculty can only access their respective dashboard by using login credentials provided by Admin.If any one tries to login with wrong login credentials the app will pop up an alert message,"UserName/Password not matched".

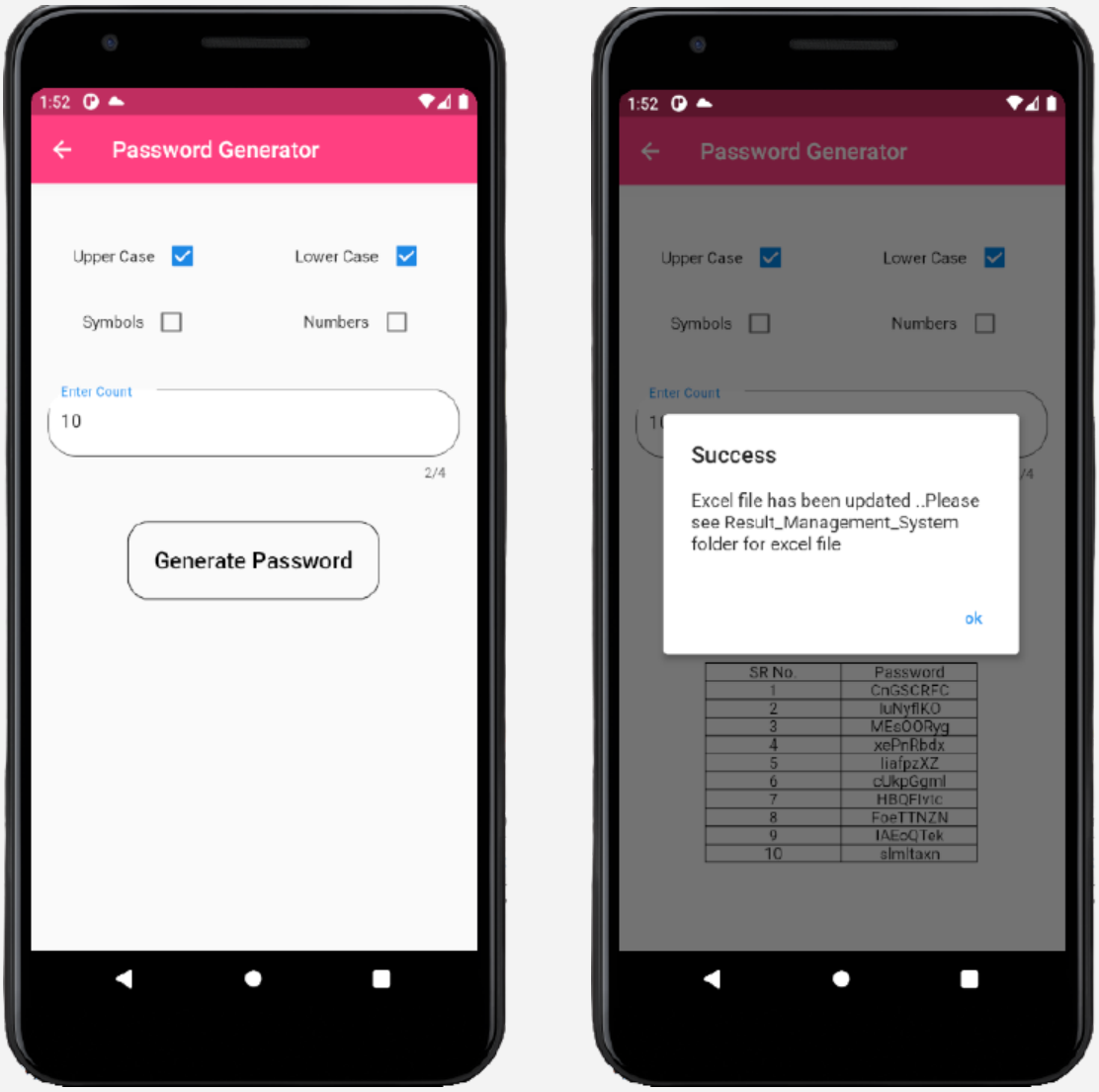


➢ *User Id and Password of student / faculty will be sent by Admin.*

Login Credentials for students and faculty will be sent by admin so the only authorised users can access the app features.

➢ *User id will be the roll no/email-id of student / faculty and password will be generated by using Inbuilt password generator:-*

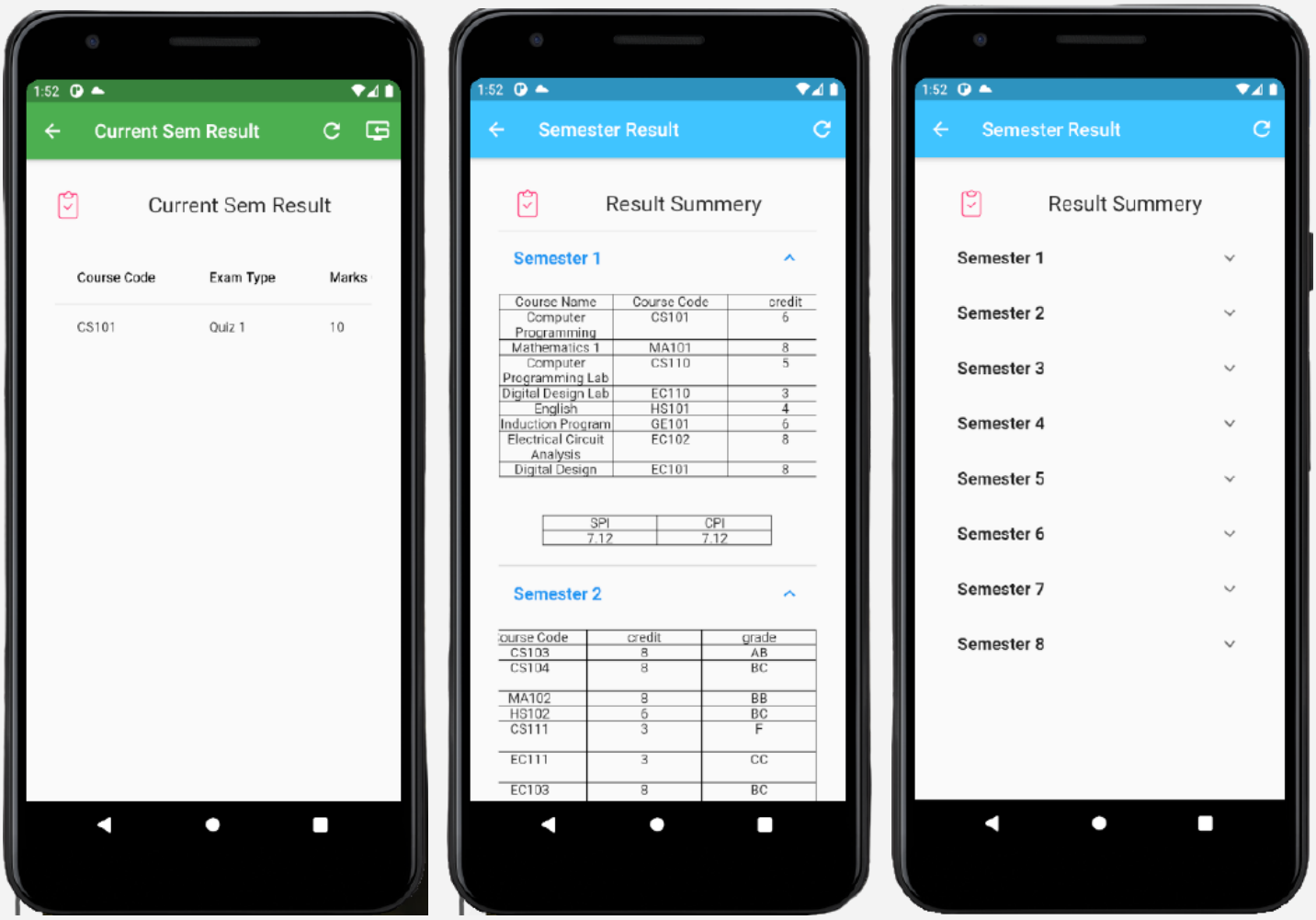
We have integrated a password generator that uses a mix of UpperCase, LowerCase, Symbols, and Numbers so that we can generate random passwords of different difficulty levels having 8 characters and it'll be really hard for anyone to guess. Later on, students can reset their passwords. After Clicking on "Generate Password" Button an excel sheet will be saved to the device. The excel sheet contains SR. No. and password in one row. Later on, Admin can use this excel sheet to create the account of students/faculty just by uploading this excel sheet to his/her dashboard's add student/faculty section.



➢ *Students will be allowed to see their result report of the ongoing semester as well as the result of previous semesters . They cannot modify or delete any information.*

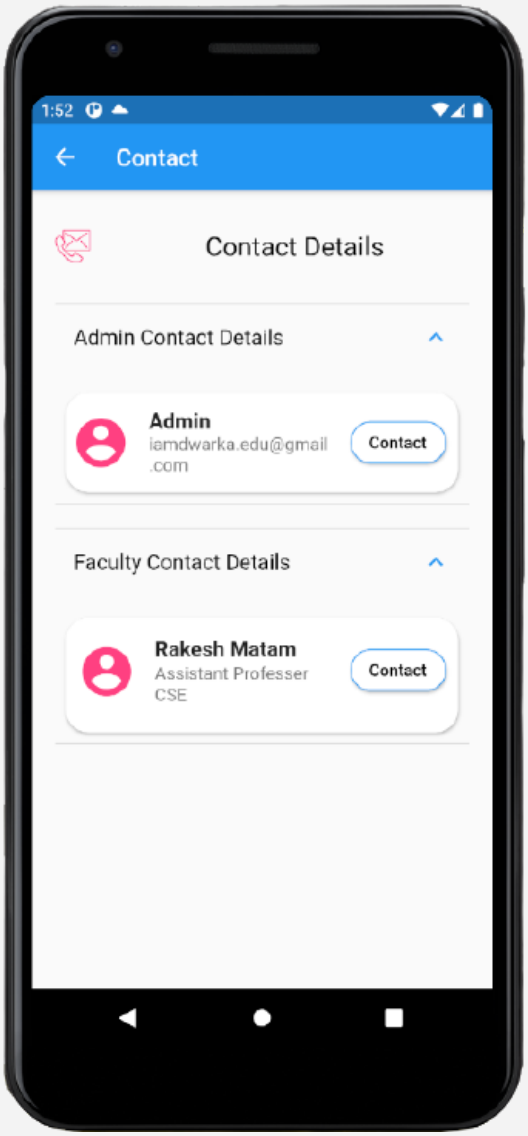
Students can see semester-wise marks in tabular form. As well as marks of quizzes, tests, mid sem, end sem, etc. of the ongoing semester that will be updated by respective course faculty.

We formulated an algorithm to automatically calculate the CPI using SPI and Course credit of all the previous semester.



➢ *Students can send an email to the administration and Faculty for their result related query.*

If students have any problem with their marks or any other query then students can send their query directly to the respective course faculty's email id or to the admin's email id by just tapping on contact button it will redirect to the default mailing app of student's device.



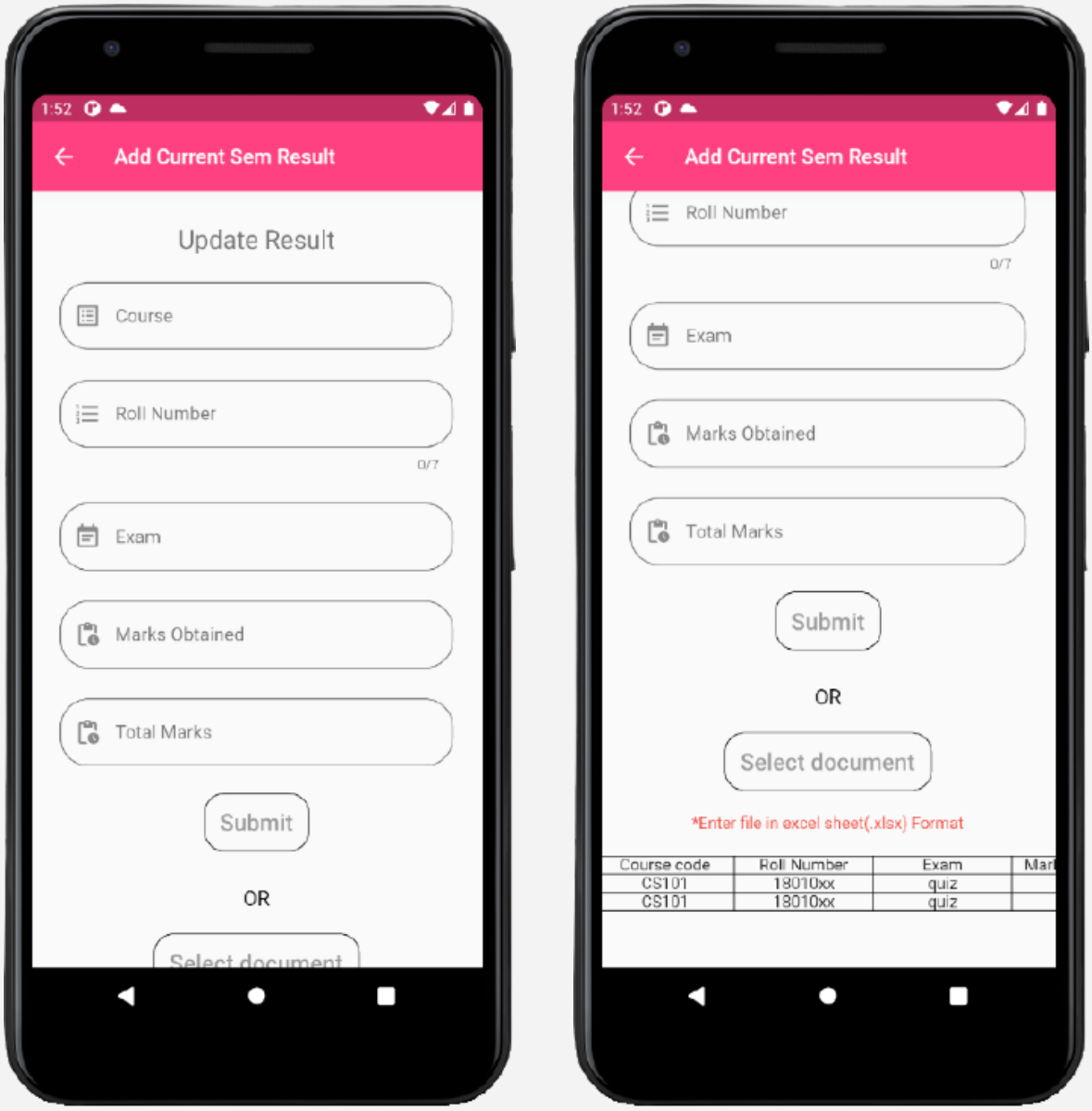
➢ *Faculty can give the final grade to the student via excel sheet which will be reflected in the student profile.*

Faculty can upload final grades of students by using Roll No.,Semester,Course Code & Final Grade

*For reference see screenshot on Page No. 12

➢ *Faculty can update marks of quiz/midsem/viva/endsem via excel sheet which will be reflected in the student profile.*

Apart from viewing the final result of the whole semester at the end, students can see the marks of quizzes, tests, weekly lab marks, mid-sem, end-sem etc. which will be uploaded by respective course faculty by using excel sheets.Students just by clicking on the current sem result can use this feature.



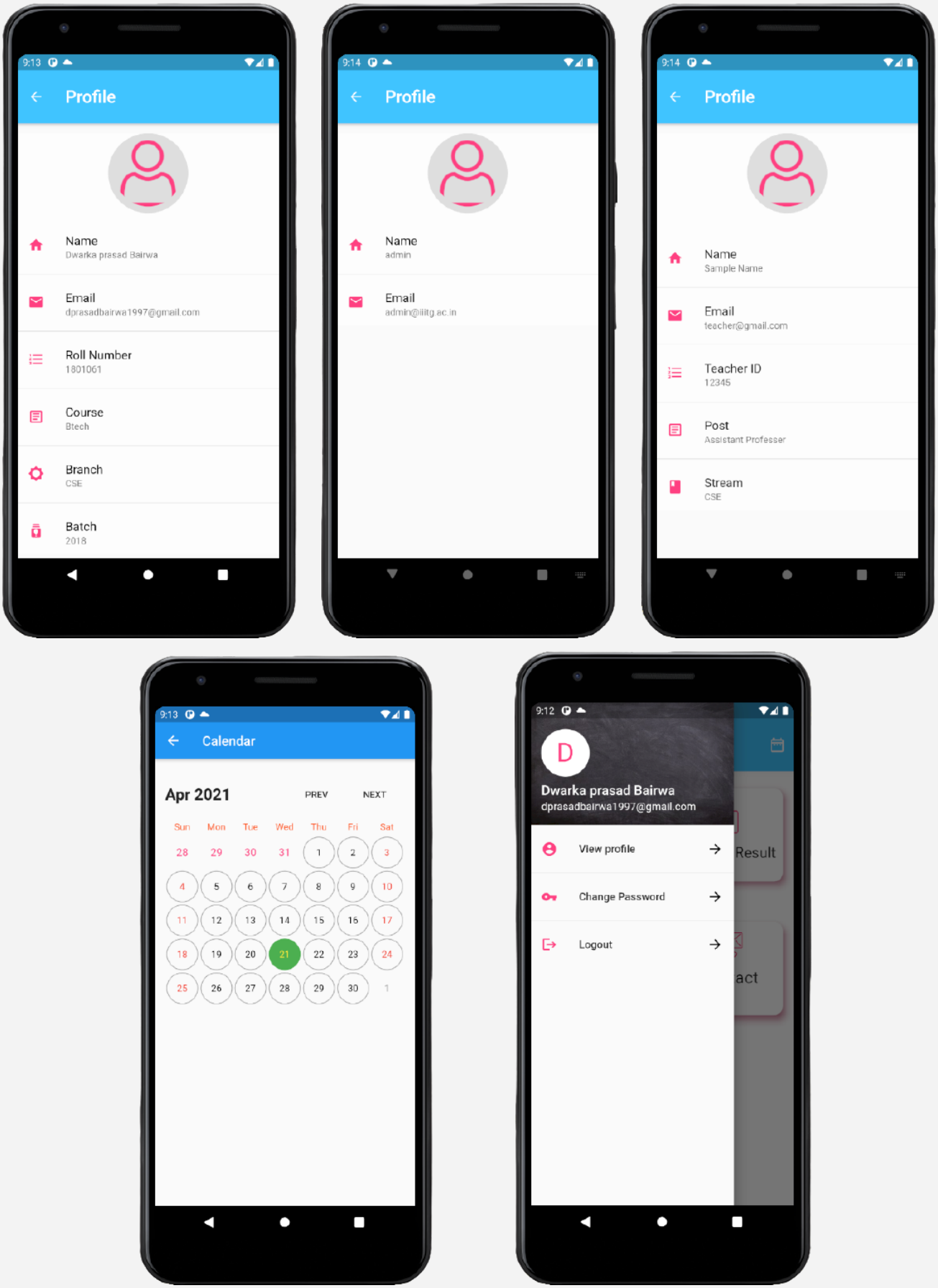
Extra Functional Requirements:

- *Profile page , Calendar and Change password Screen For All Users* :

  Added a side drawer that consists of 3 features:-View Profile, Reset Password & Logout.

  - By tapping on “View profile” students can view their Name,Roll No.,Email ID,Course,Branch and teacher can view their name,profession.

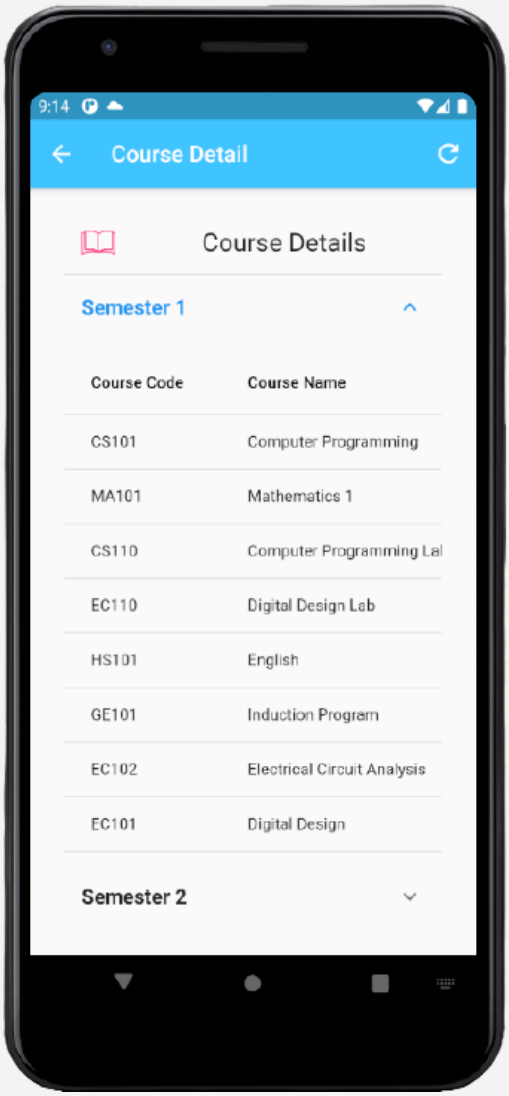
  - Reset Password will send a reset password link to the user's registered mail by using firebase inbuilt feature.


Profile
Name
Dwarka prasad Bairwa
Email
dprasadbairwa1997@gmail.com
Roll Number
1801061
Course
Btech
Branch
CSE
Batch
2018
Profile
Name
admin
Email
admin@iiitg.ac.in
Profile
Name
Sample Name
Email
teacher@gmail.com
Teacher ID
12345
Post
Assistant Professer
Stream
CSE
Calendar
Apr 2021
PREV
NEXT
Sun Mon Tue Wed Thu Fri Sat
28 29 30 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
D
Dwarka prasad Bairwa
dprasadbairwa1997@gmail.com
View profile
Change Password
Logout
Result
act

➢ *Course Detail Screen For Student and Teacher :*

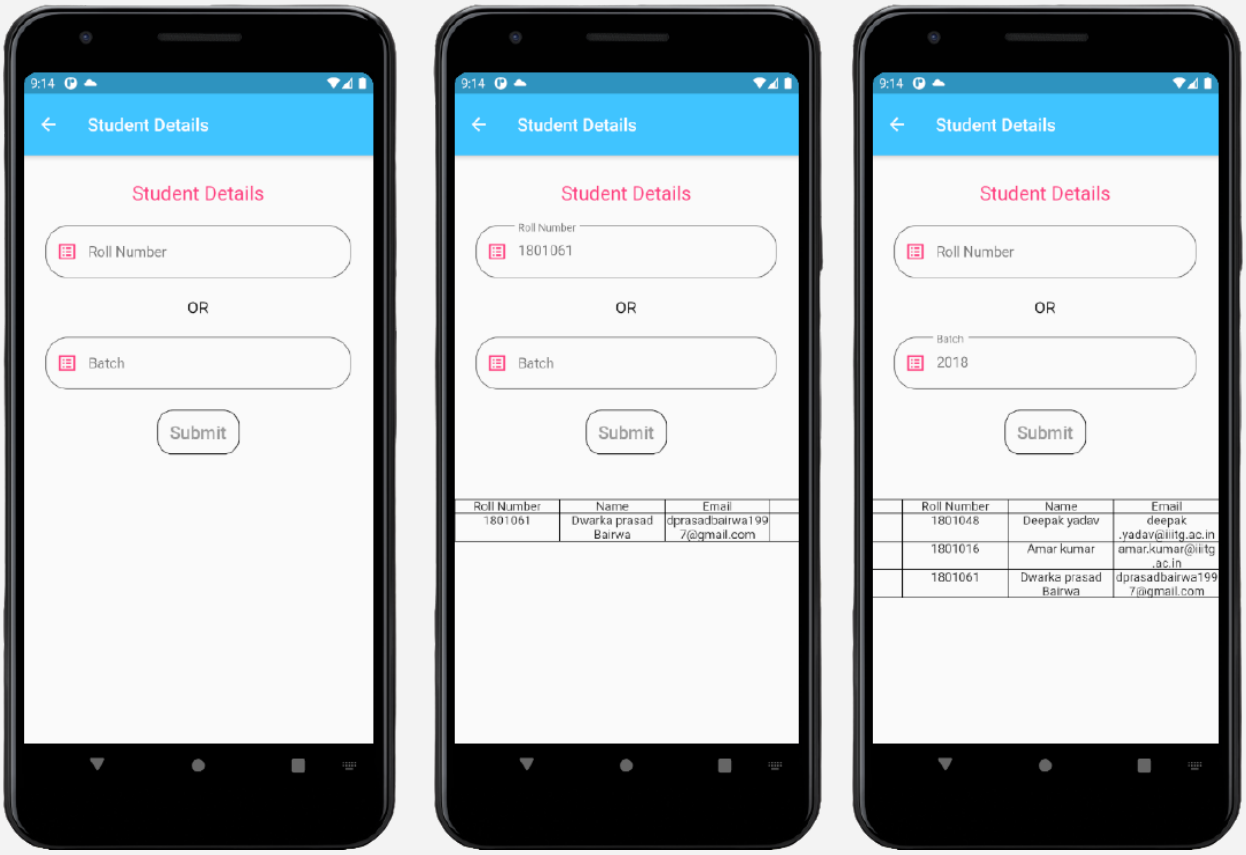
➔ Students are only able to view the semester-wise courses for which they have registered by admin. Suppose that a semester has two elective courses then a student can not view all the elective courses in their dashboard rather than they will be able to view the elective course they have chosen.

➔ Teachers can view the semester-wise courses that have been assigned to them by admin.



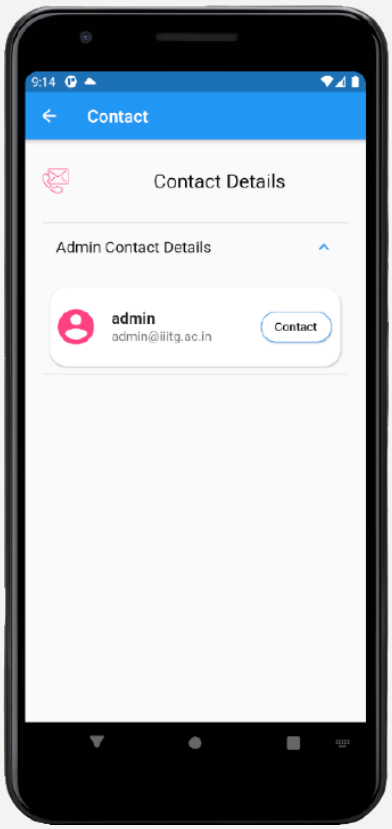
➢ *Student Detail Screen For Teacher and Admin :*

➔ By using Roll no. teacher or admin can view the details of any specific student. App also has a feature to view the details of all students just by entering their batch name.



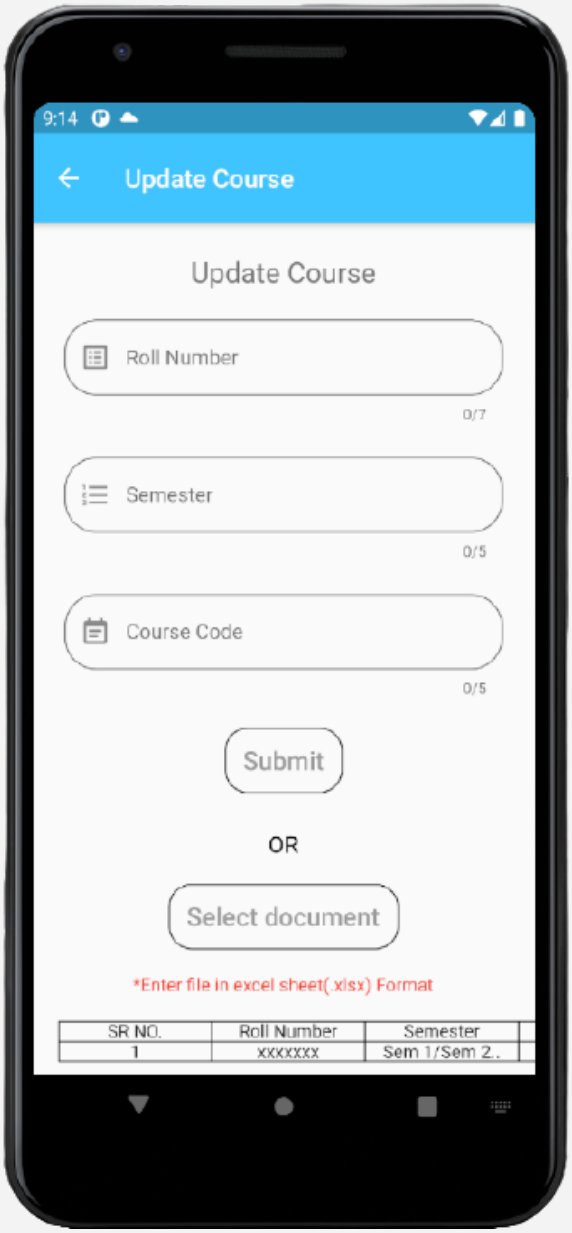
➢ *Contact Screen For Teacher* :

➔ Teachers can directly contact the admin by clicking on the contact button,it'll redirect the teacher to their device's default mailing app.



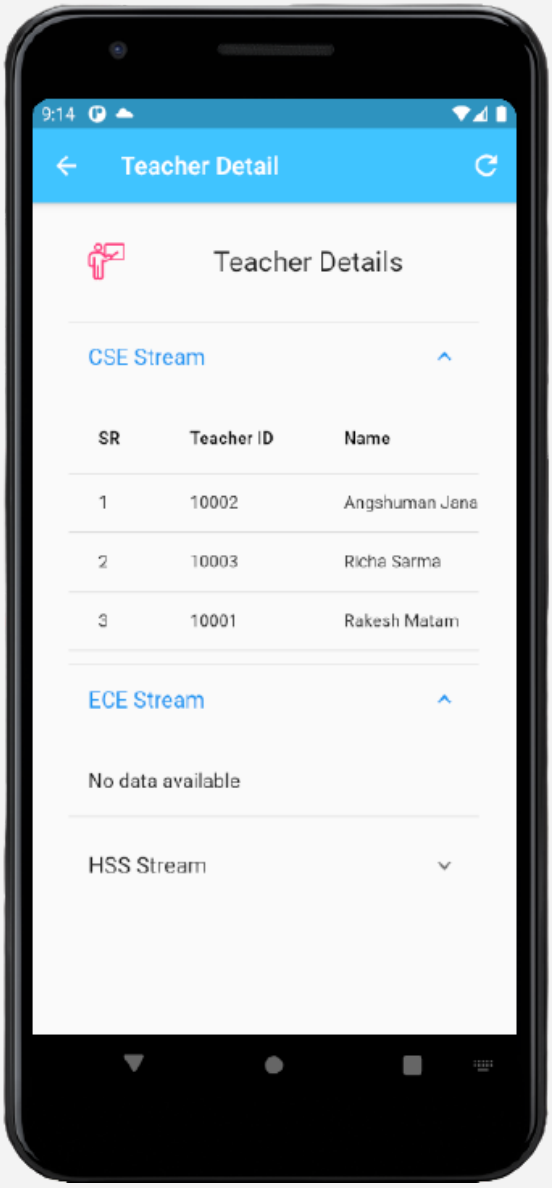
➢ *Update Course Screen For Admin* :

➔ Only the admin has permission to add any course to the student dashboard by using a form orby uploading an excel sheet of predefined format.



➢ *Teacher Details Screen For Admin* :

Admin can see the details of registered teachers in the respective stream.

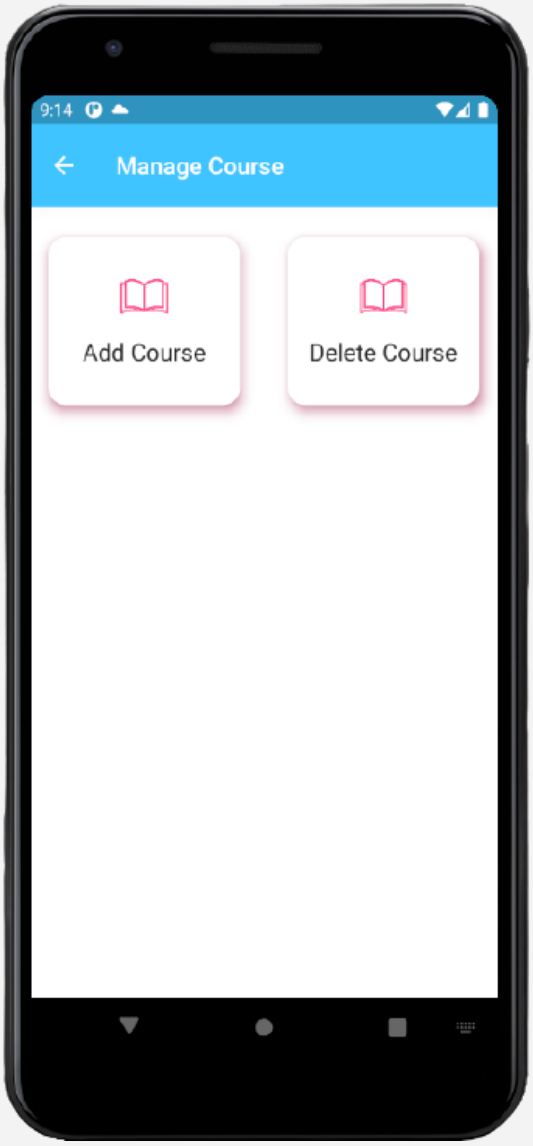


➢ *Manage Courses Screen For Admin* :

Admin can add the course and update the course.

- *Add Course* : Admin can add a single Course by using a form.

- *Delete Course* : Admin can delete a course by using a form.



Non - Functional Requirements:

➢ *Security :*

We developed a secured database for the System Which has the following features .

- ❖ Password is stored in Encrypted form.
- ❖ Users have different access rights so that they can access only those information for which they are assigned (Update, Delete etc.).

```
Role: "admin"
```

```
Role: "teacher"
```

```
Role: "student"
```

➢ *Safety Requirements :*

Cloud Firestore managed export and import service to recover from accidental deletion of data and to export data for offline processing. We can export all documents or just specific collections. Likewise, We can import all data from an export or only specific collections. Data exported from one Cloud Firestore database can be imported into another Cloud Firestore database .

➢ *Reliability :*

As we are using firebase for our backend services.Firebase security is enforced by server-side rules, that *you* author, and govern read or write access to given paths in your Firebase data tree.

➢ *Mobility :*

Our System can be Used from anywhere just by entering Email and Password.

➢ *UI Design / Functionality :*

UI Design is User friendly and easy to understand . Whenever it is needed a specific format we have given format to understand.

Implementation

➢ *Waterfall Model* :

To implement the System we followed Waterfall Model steps :

1. In the first Phase , We gathered all possible requirements of the system to be developed and finalized the tech Stack and created an SRS document.

2. In the Second Phase , We created a Data Flow Diagram and Use Case Diagram(Prepared System Design).

3. In the Next Phase (Implementation Phase) , We started implementing the app in small units. Each unit was developed and tested for its functionality simultaneously. (Manual testing).

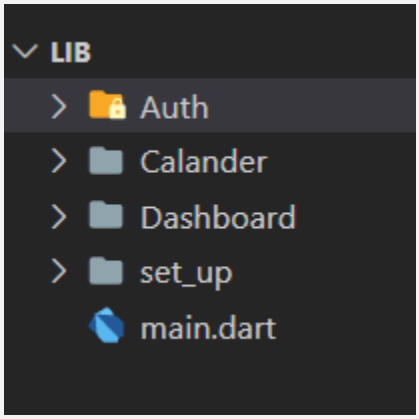
4. In the Next Phase (Testing Phase) , All the Developed units were integrated and tested manually and some part of System UI (because form is not tested as ROBO test unable to work properly with flutter form fields) is tested by ROBO test in Firebase Test Lab.

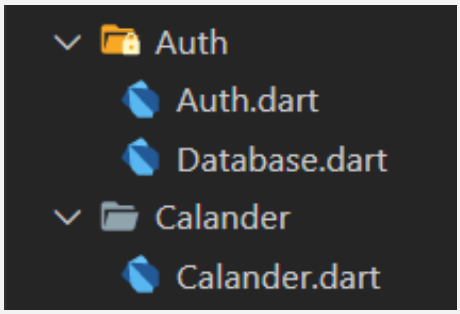
5. In the Next Phase Deployment , we created an apk file.

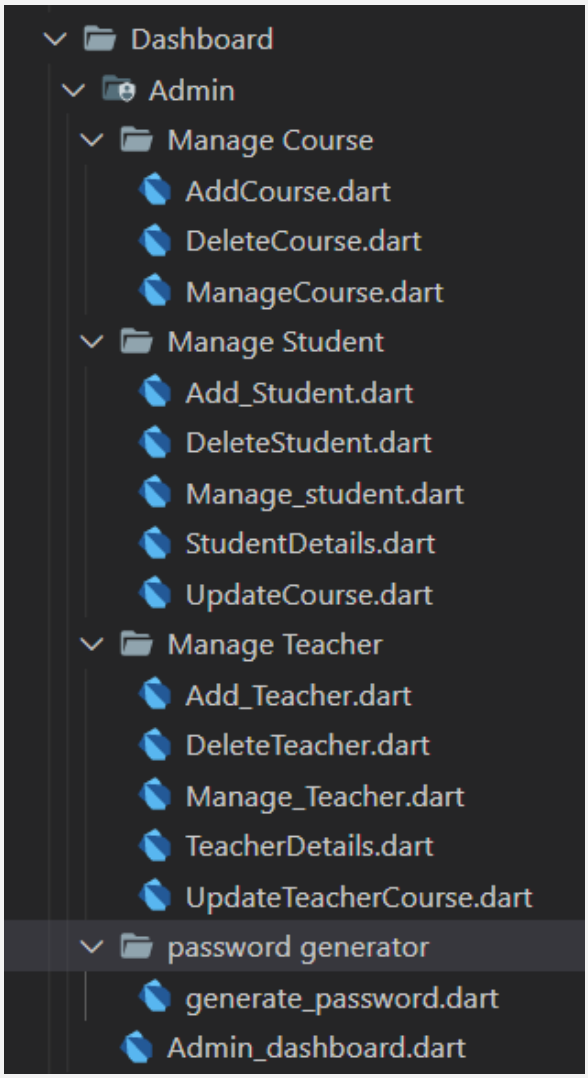
➢ *Technology Stack* :

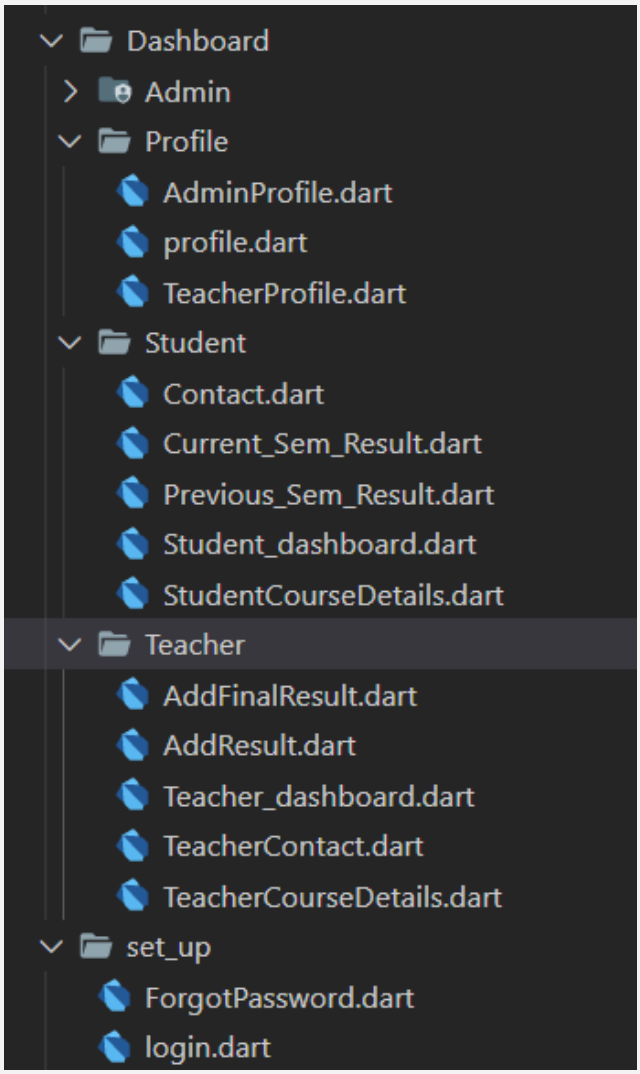
1. Database Server : **Firebase**
2. Client : **Android/i-OS**
3. Development Tools : **Android Studio , VS Code**
4. Programming Language : **Dart**
5. SDK : **Flutter**

➢ *Project File Structure* :









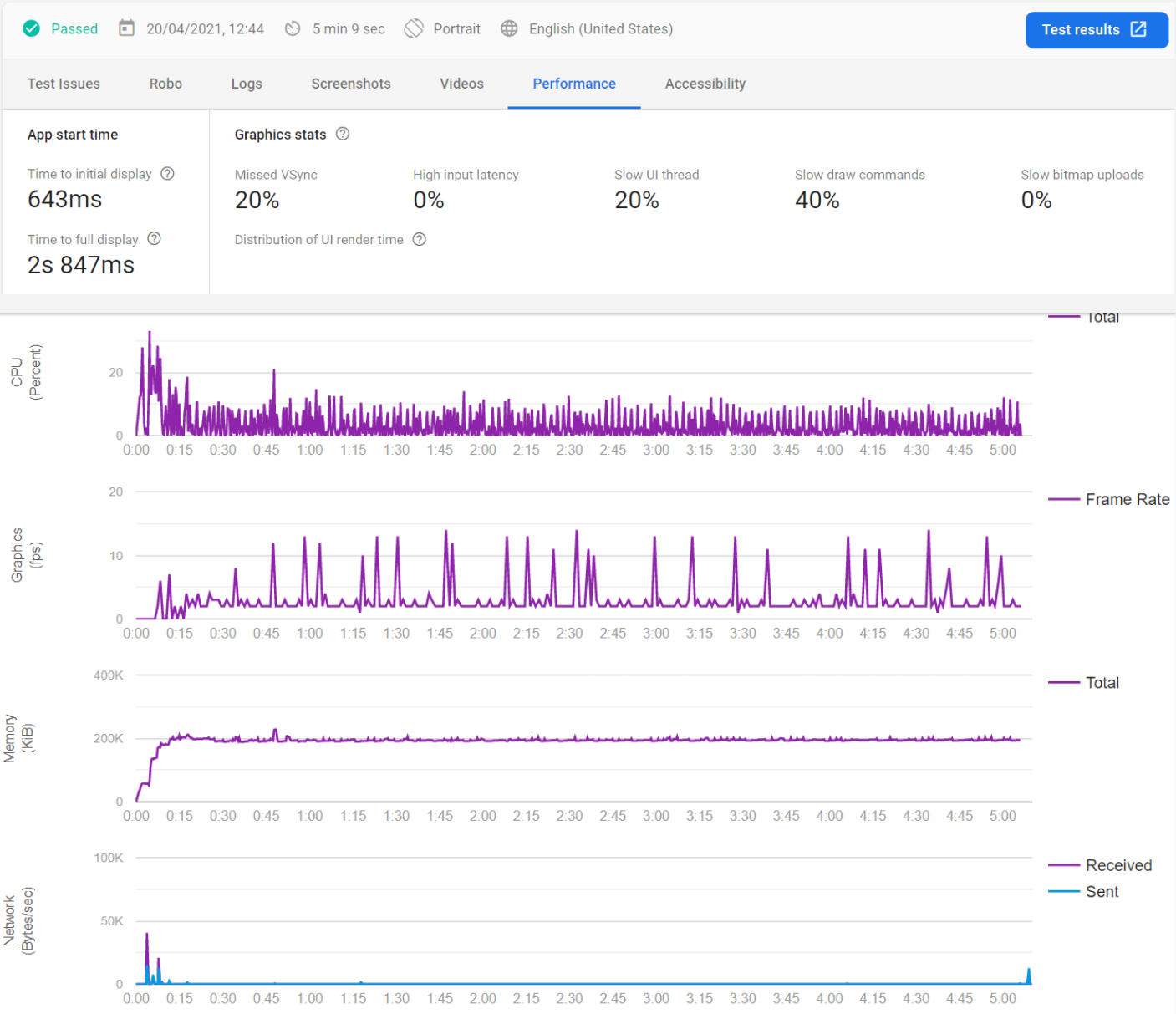
➢ *Application Requirements* :

```
environment:
  sdk: ">=2.7.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^0.4.0+9
  firebase_analytics: ^5.0.2
  firebase_auth: ^0.14.0+5
  cloud_firestore: ^0.12.9+5
  flutter_svg: ^0.19.1
  excel: ^1.1.5
  cupertino_icons: ^1.0.0
  firebase_database: ^3.1.6
  package_info: ^0.4.3
  device_info: ^0.4.2+8
  file_picker: ^1.12.0
  flutter_document_picker: ^3.0.0
  documents_picker: ^1.0.0
  random_password_generator: ^0.1.0
  url_launcher: ^5.7.10
  flutter_calendar_carousel: ^1.5.2
  splashscreen: ^1.3.5
  path_provider: ^1.6.14
  connectivity: ^2.0.0
  shared_preferences:
  syncfusion_flutter_pdf: ^18.2.59-beta.1
  open_file: ^3.0.1
  permission_handler: ^5.0.0

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_launcher_icons: ^0.8.0
  test: any
```

Testing

➢ *ROBO Tests* :

Robo test is a testing tool that is integrated with Firebase Test Lab. Robo test analyzes the structure of your app's UI and then explores it methodically, automatically simulating user activities. We have used ROBO tests to check System and Output is Given below :



- *Manual Testing* :

  We have implemented all the mentioned functionality in our app using different types of stateful, stateless widgets and functions. We have manually tested all the functionality of the app and below We are providing a drive link for your reference of manual testing which consists of all the required excel sheets and video of a fully functional app.

  Click Here : Manual Testing

Limitations

Although we have put our best efforts to make this software flexible , easy to operate,we can not rule limitations out even. Though the software presents a broad range of options to its users , some intricate options could not be covered on it.

- The application is only for Android & i-OS and will not work for other mobile OS like blackberry,Symbian etc.
- We have not implemented a Change Password page in which you can enter your password and confirm it . We are sending a Reset link to Email - Id .
- When an admin tries to delete account of any admin or student then he/she can only be able to delete data of user from firestore but not from authentication, Admin have to do this manually.

Future Enhancements

- In future, We can add some academic features which are useful to the student like the class timetable, holiday list,fees management etc.
- We can make a web version of the current application.

Contributions & Resources

- *Dwarka Prasad Bairwa (1801061)* :
  - Code Implementation (Admin and Teacher) & Database (Function for retrieve and Send Data)
  - Data Flow Diagram & Use Case Diagram
  - SRS implementation (Functional Requirements).
  - Bug Fixing

- *Amar Kumar (1801016)* :
  - SRS Documentation
  - Database Architect and Design
  - Code Implementation(Role based Authentication,Add Admin Functionality)
  - Project Architect
  - Manual testing

- *Deepak kumar Yadav (1801048)* :
  - Done Robo Testing
  - SRS Documentation & Data Flow diagram
  - Integrated Flutter with firebase.
  - Code Implementation (Student UI integrated with Firebase).

Resources :

- Manual Testing
- Github