Authentication Mobile Apps Development

Indian Institute of Information Technology, Allahabad

Project Mentor:

Prof. O.P. Vyas (Dean Technology & Development),
Dr. S. Venkatesan (Department of Information Technology)

Technical Assistant - **Abhinaba Basu** (Automation Project)

Teaching Assistant(TA) - **Bagesh Kumar** (Senior Research Scholar)

Students Name		Tools & Language
Akash Biswas	(IIT2020001)	Java, Js, MySQL
Rohan Singhvi	(IIB2020004)	JS, Java
Shubham Kumar	(IIT2020007)	Frontend (Android)
Aashish Agarwal	(IIT2020009)	Java, MySQL, Android
Rahul Mahato	(IIT2020022)	Python, JS, Java

Table of Contents

1. Problem Statement

1.1 Problem Definition:

Existing process of the getting information for the student is done by manually inserting in the "Aviral" website. All the wanted information need to be provided by the administration. If the students want to know the results of the particular subject so there should wait for long time because work done manually. Sometimes the existing system have time consuming to fetch the data and in proper manner. The existing system is not fully secure and can be attacked by unauthorized user. The code which was written in the existing system is not written in a structured manner leading to writing of more code to implement a function.

1.2 Proposed System

We have successfully proposed the improved version "Aviral" for replacing the mentioned problems faced today. This application is flexible and can easily access by the admin or faculty in their mobile phones. Once we login to the system then it is easy and compatible to admin or faculty to fill the result and information about the students. The data can be saved to the system and then the data can be stored to database in safe and proper manner. First, we will be converting the existing website to an app using ionic Framework. Then we will change the UI design of the existing system which is very outdated. Also, in this project the code will be in a structured manner such that the code can be reusable i.e., instead of writing the whole code for implementing a new function and increasing the length of the code we will reuse the same code for implementing a new functionality.

2. Introduction

Technology in today's world has reached to extent that it can be used to do various task in day-to-day life easily with less effort and time. World today has realized importance of education in one's life which has led to revolution in field of education. Universities, colleges, schools today have loads of task to be completed in given timeline. In today's scenario colleges still needs to calculate the student result manually which takes a lot of time and effort by the faculties working on it.

A web-based school management system will reduce the manual work by deploying centralized software incorporated with various loosely coupled services which interact with each other to address above mentioned issues and improves the communication between management and the student/guardian through notifications via email, SMS and push messages. The system takes file of student results obtained by universities from faculty as an input. The result of analysis will then be displayed in according to the details of student. As it is a server-side enterprise application it is designed to support desktop browsers, mobile browsers and native mobile applications.

The Aviral is developed to provide efficient, inexpensive, well grounded, well planned and comfortable usage of the application for college staff or administrators. The existing system having very time taking process. This application can Help the staff to Perform various operation on the result like insert, update, delete and view result very easily. In addition to this project gives a provision of time taken for getting the information will be reduces. This application stores the result and meaningful information of the student in database very safely.

when the admin wants to view the result or information about the student then it will easily retrieve the information from the database. This application having the unique login module which is only for the trusted administrator or faculty to perform the operations on the result.

The current system was developed using Python (Django), React Native and RDBMS and was hosted locally using Apache web server. The data used were obtained from the Department of Information Technology and Electrical and communication technology, Indian Institute of Information Technology Allahabad.

Functional decomposition of the system and its key modules are provided to explain the major functionalities proffered by the system. Also, use case diagram is presented to show the different categories of the system users and the various functionalities associated the different system user.

3. Literature Review

Title: Development of Students Result Management System: A case study of University of Uyo.

Introduction:

In this paper, a sample student RMS design is presented. In the system, the department of Electrical/Electronic and Computer Engineering University of Uyo is used as a case study. A detailed functional decomposition of the system is presented 27 along with use case diagram.

Methodology:

In this paper, the development of student's result management system (RMS) is presented. The software is meant to address the challenges facing student's records management in the University of Uyo.

The RMS is developed using PHP, MYSQL, HTML, CSS, JAVASCRIPT and was hosted locally with Apache web server.

Also, a Participatory Incremental Process Model (PIP Model) is used as the software development methodology.

Functional decomposition of the system and its key modules are provided to explain the major functionalities proffered by the system.

Result:

The software is meant to address the challenges facing students' records management in the University of Uyo.

As compared to existing methods of managing the academic institutions, this project which yields centralized software makes the work administration and management easier and provides detailed information about the topic of user's interest in just one mouse click.

Dataset Used:

The department of Electrical/Electronic and Computer Engineering University of Uyo is used as a case study.

Title: Student result management system.

Introduction

Student Result Management System is a web-based application that mainly focuses on providing the results to the student and the faculty. The student checks their respective results using their university registered recognition id's along with their grades and percentage of that particular semester.

The student accessing their results through college site is more convenient and the faculty can easily analyze the pass and failure of a particular subject.

The system is divided into three modules- Student, Faculty and Administrator. The student using his roll number can view his results and the faculty using the joining year and the subject name and view the analysis of pass and failure count in the selected subject. The administrator uploads the results file to the database by converting the file to SQL format(.SQL) from the PDF format (.pdf

Methodology

The system can be developed using web technologies HTML, CSS, PhP and using the database MySQL.

The front end can consist of user registration with the respective university registered number and the password by the user.

The student can view his results in the tabular format with the respective aggregate and percentage of that semester. The data based on the roll number of the student all the data can be retrieved back to the table and displayed as results.

The PHP can also be used for visualization of data. We use fusion charts for the dynamic visualization

Primarily the data can be collected from the college administration. This data includes university registered number of every student currently collected is then classified and tabulated into useful and understandable manner.

HTML is used to develop different pages like user registration, login page and the page for providing results.

PHP is used for connecting to the database and perform operations on it through queries.

Result

System testing, or end-to-end testing, tests a completely integrated system to verify that it meets its requirements.

For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion of entries, then logoff.

Unit Testing
Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

these types of tests are usually written by developers as they work on code, to ensure that the specific function is working as expected.

System testing is undertaken by independent testers who haven't played a role in developing the program.
User Input Validation Testing

The User input must be validated to confirm to expected values. In the end the result met it requirements as expected.

Dataset Used

The department of computer science and engineering,

Andhra Loyola Institute of Engineering and Technology Jawaharlal Nehru Technological University, kakinada

4. Research Gap

The disadvantages of our current model are that the memory taken for loading is 40mb. Our proposed model is to get it down to kb. It can sometimes take more time than expected. Sometimes we have to log out first to login into the system. Extensive modules and features make it difficult for a user to utilize the application. When traffic on website is high then it is prone to performance issues. Risk of data mismanagement.

5. Proposed Methodology

In the proposed methodology there are 5 stages which are gathering requirements, design, development and implementations, testing, and maintenance.

a. Gathering Requirements

Before taking up any projects, the requirements must be collected and verified for the feasibility. The project can continue if the requirements are feasible. In this phase all the requirements necessary to develop and implement the project are collected and explained below. In this project, final product will be a mobile based application, the requirements are categorized into four categories such as, Student Management Service, Course Management Service, Administration Management Service, an Employee Management Service.

b. Student Management Service

In this service, the student will be able to view their attendance, progress report, result, view notifications.

c. Course Management Service

In this service, the admin will be able to add, modify and delete the courses. Admin will also be able to add, update and delete the subjects of the particular course. The teacher and students can only view the courses added by the admin.

d. Administration Management Service

In this service the admin will have access to all the resources. The admin can broadcast the notifications via email, SMS and push messages. The admin can also add, update and delete the student, guardian and employee details.

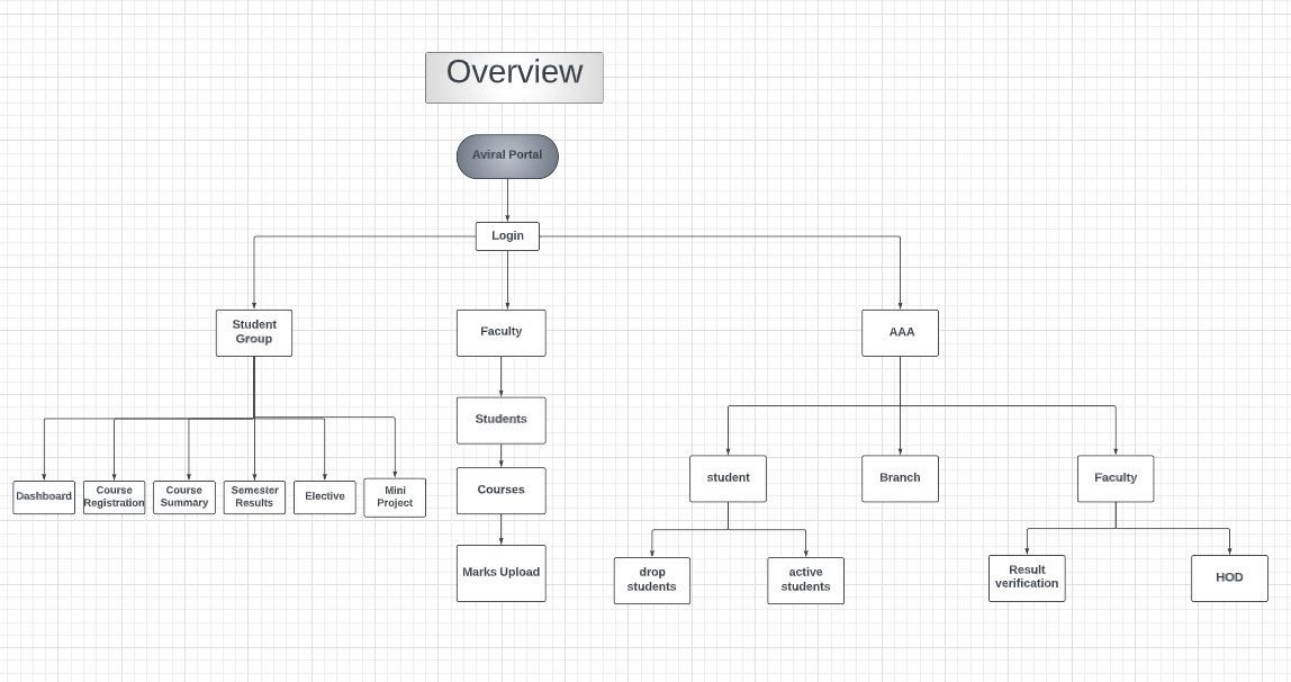
e. Faculty Management Service

In this service, the faculty will be able to view the details of the students under their mentorship. The teacher will be able to report the student activities and evaluate students' performance. Teacher will be able to upload the assignments and verify the assignments.

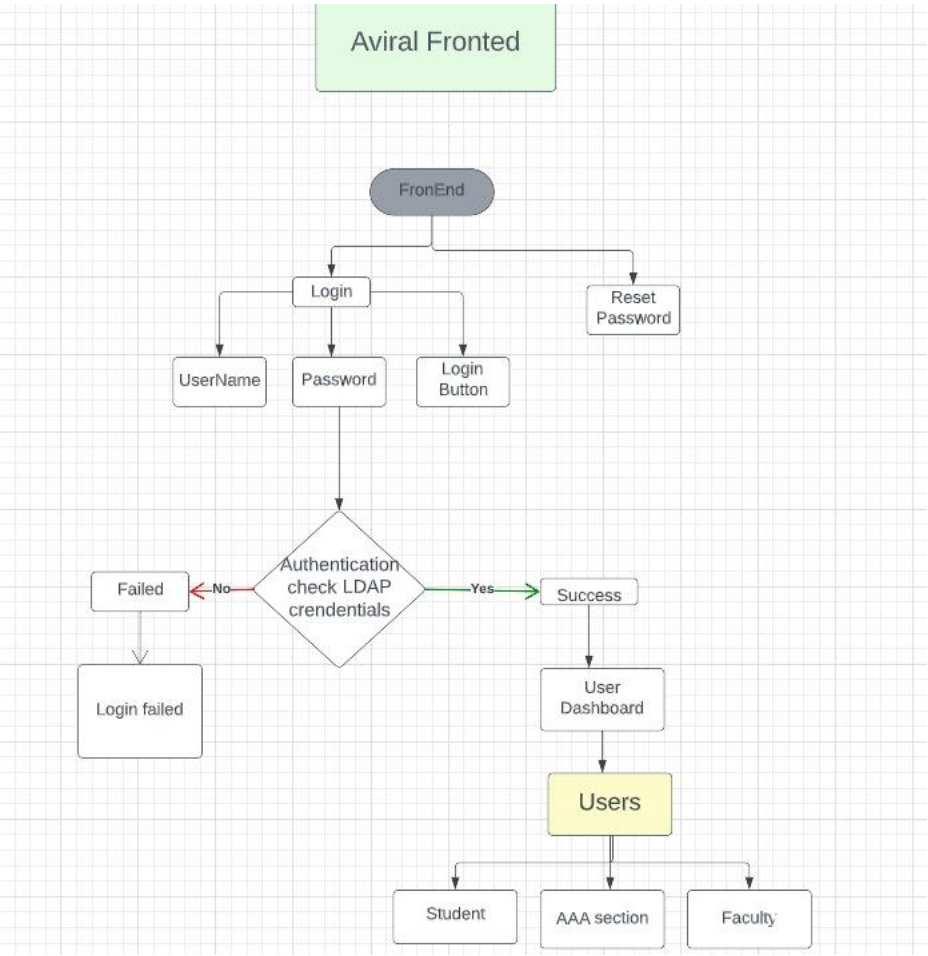
Functional Decomposition of current Aviral System

The functional decomposition of the result and transcript management system is presented and the modules are further decomposed into its respective sub-modules as shown in the subsequent figures. The system can be accessed by both students and faculty. The user must register on the RMS platform and then login with the user username and password. The user access privileges depend on the role the user plays in the system. The system administrator i.e., AAA role has the highest access privileges. Also, the students can access the system as ‘student’. The student must register on the Aviral platform and then login with the student username and password. In the system, the student will have student privileges which will enable the student to register for course, drop registered course if required, view the result pertaining to the student alone, change the student’s settings like the photo, password and student’s contact address a, email and phone number.

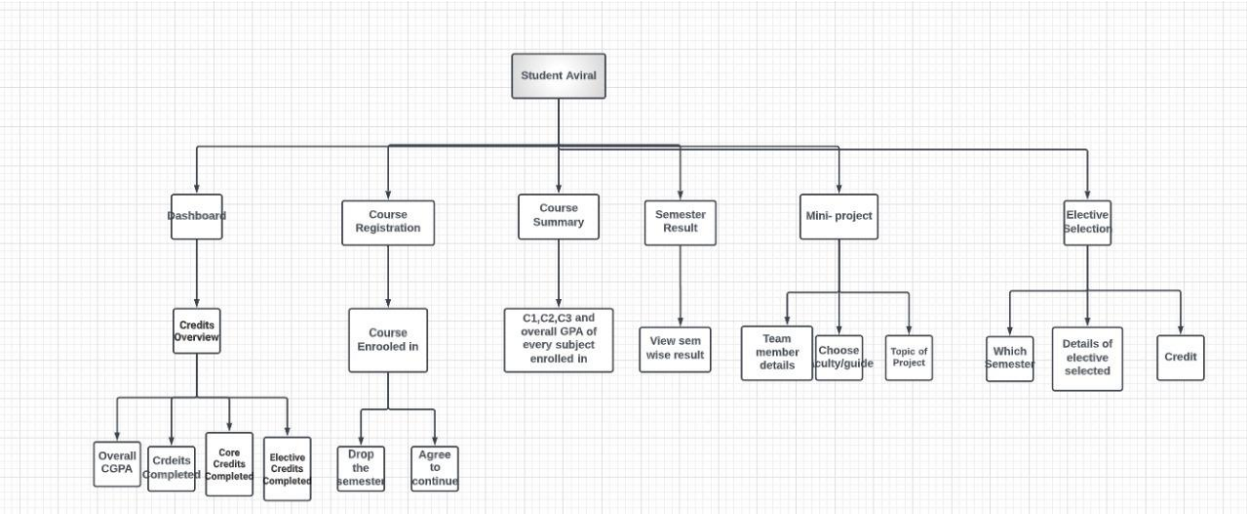
Overview:



"Aviral" Frontend:



Student Frontend:



Hardware Requirements

- Standard PC
- Internet connection with good speed.
- Pentium IV 1.7Ghz dos or better processor.
- 1 GB ram or more
- At least 10 Gb Hard Disk Space.

Software Requirements

- Visual Studio Code
- ReactJS
- Web browser supporting HTML 5 and JavaScript
- Firebase

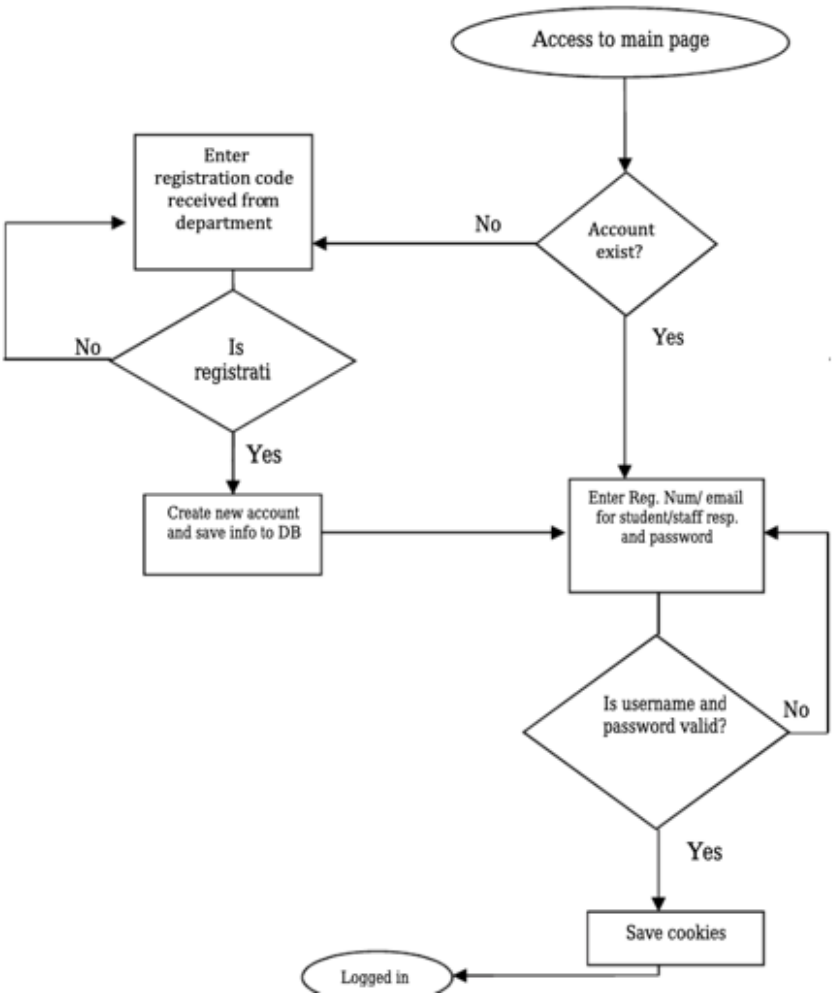
Registration and Login Module

Whether a student or staff, the system policy requires that every user should login each time the user want to access the system functionalities. In this system as per the username and password a new interface will be shown to each user respective to their role in college.

First the user will enter the username and password to the frontend. Then related component gets opened. Then it will search for the access permission which is stored in browser local storage and can be manipulated easily.

Then the username and password are checked from the backend which is then authenticated using LDAP/Database. If the authentication is found valid then it will generate token for the session which will store the user related data like logout time (time+30min). later it will return the token along with the user information, user group to the frontend.

The frontend store token, user group and user information to the browser local storage for the next API calls.



Our objective:

Our main motive is to change the current UI design of the “Aviral” which is outdated to a new modern UI, the modern design will be more user friendly. The proposed system will also contain code which is reusable such that more code need not be written again and again for implementing new feature.

Our second motive is to create the same “Aviral” application as a mobile based application for the faculty and the students. The “Aviral” will be created using React JS framework along with Django as the backend. Then using the ionic framework, we will convert the desktop app to a mobile app.

6.Dataset Description

All the data are taken from the previous “Aviral” database. The data consist of data about details of students from Information Technology, Electrical and communication Technology of BTech and MTech and even of Faculty members of Indian Institute of Information Technology, Allahabad.

7. Evaluation Metric

Android app development process is divided into several Phases

7.1 Tools needed for project

- **Language - Python (Django), React Native**
- **Tools - Visual Studio Code**
- **Database - RDBMS**

7.2 Understanding the Aviral Portal

A. DEAN.AAA Section (Admission assistant and award section)

1. Dashboard

a. **Course registration overview**

i. **Course Registration:** AAA can see the courses registered by the students.

ii. **Control panel:** Consist of details of the students like Batch id, Program name, Total, registered, Active

b. **Component Submission overview**

I. **Component Submission:** Professor can upload the Marks of the students for C1, C2 and C3.

II. **Control panel –**
Consist of details of the students like course id, department, Active courses.

There is even option for:

- **Stop publish-** Control the already published marks.
- **Open Publish**- will publish the uploaded marks.

2. BTech Branch Change: AAA can change the Branch of a student if they meet the requirements.

a. **B.tech IT:** Consist of details of the students like student id, student name,GPA

b. **B.tech ECE:** Consist of details of the students like student id, student name,GPA

c. **B.tech BI:** Consist of details of the students like student id, student name,GPA

3. Projects:

a. **Major projects -** Eligible BTech students and MTech students can apply for it. Select faculty under which project can be done. Details of selected group members of team is shown. All the information related to the project is then sent to the HOD by the AAA.

b. **Mini Projects: Record-** Final year BTech students, pre final year BTech and MTech students can apply for it**.** Select faculty under which project can be done. Details of select group members of team is shown.

4.Theses: Applicable for PHD students.

5. Summer internships: 8th Sem students have option to do major project or 6 - month internship (opt for blank semester) Internship is evaluated at the end of 6 months.

6.New course request: HOD can request for a new course to be added in the curriculum. Hod can raise query to upload a new course and after the approval of competent authority it can be uploaded. It is made optional.

7. Statistics: Shows the details of students like credits completed, semester wise result, drop courses and overall cgpa.

a. **Drop-list:** Shows the details of students of different departments like IT,ECE,As,MGMT who have dropped a course and the reason behind it.

b. **Make-Up List:** Shows the details of students of different departments like IT,ECE,As,MGMT who are giving make-up test.

c. **Unregistered Students:** If students have not completed the course registration completely then the AAA will contact and get it done manually.

d. **Add on statistics:** Shows the details of course id, course name, opted 1st Preference, opted 2nd Preference. If there is a tie then higher cgpa students are given more preference.

e. **Elective statistics:** Shows the details of course id, course name, opted 1st Preference, opted 2nd Preference. If there is a tie then higher cgpa students are given more preference. Student can give 4 preferences and will be allotted accordingly

10. Preference lists

a. **Addon student list:** Choose addon course, choose preference, submit

b. **Elective student list:** Choose addon course, choose preference, submit

11. Result Menu

a. **Complete result –** A batch wise result is generated which can be downloaded the user.

b. **Grade card –** AAA can also generate a grade card which consist of batch name, format, type (core, elective and addon), session time.

c. **Generate transcript –** AAA can also generate transcript using batch name which will contain all the details of the student present in that batch.

12. M.tech Menu

a. **M.Tech SPLS preference:** Seat declaration of MTech it, MTech ECE is done by the HOD where specialization allotment is done based on student preference.

b. **M.Tech SPLS Allotment:** The AAA can confirm allotment and also be clear the allotment in case of any problem.

c. **Component statistics:** Shows the marks submission statistics done by faculty. Also shows the last date and time for submission.

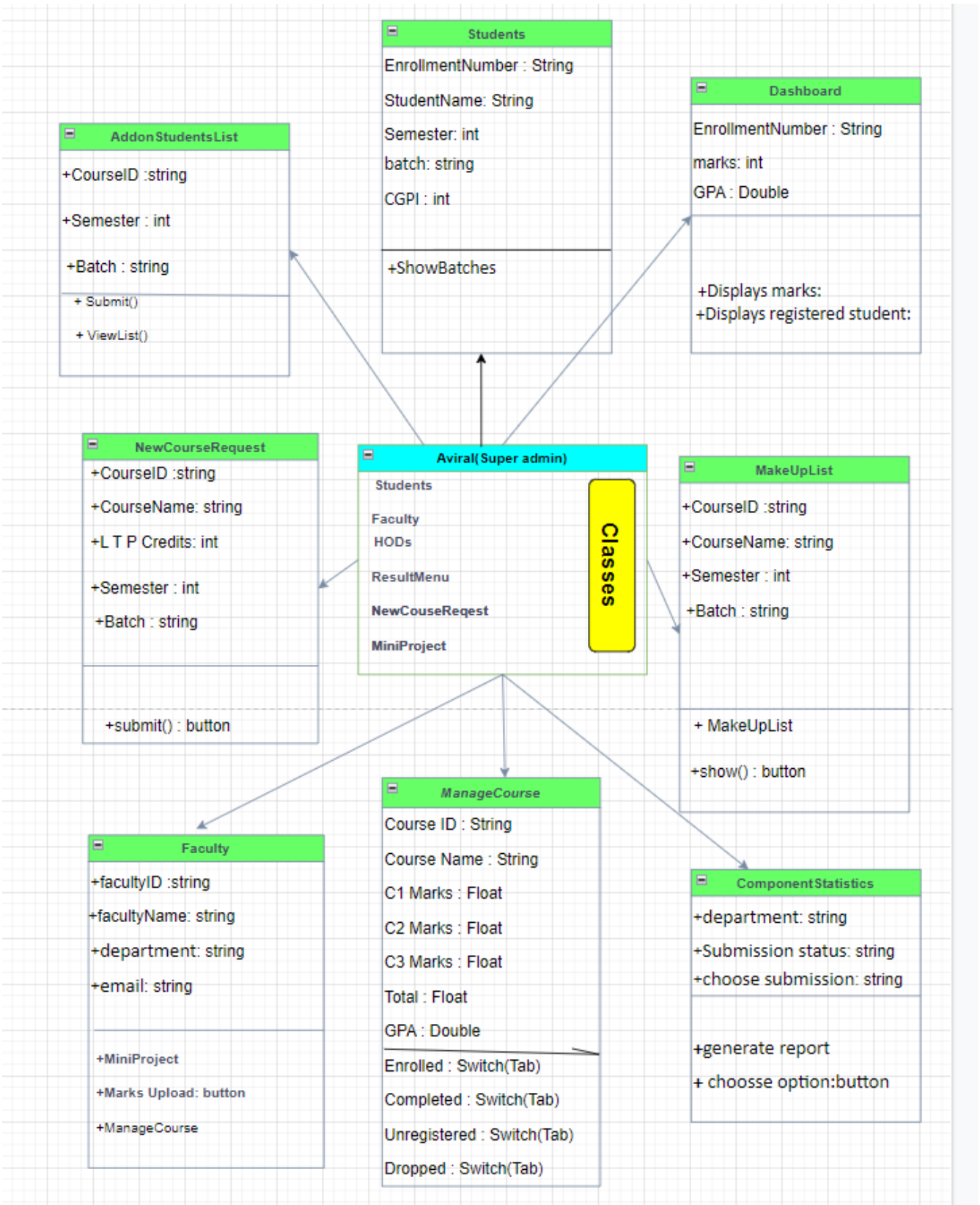
13. Details

a. **Courses:** Available courses for BTech, MTech, Instructor for courses

b. **Faculty:** Can check the details of the faculty members.

c. **Students:** Can check the details of the students.

Class Diagram:


Students
EnrollmentNumber : String
StudentName: String
Semester: int
batch: string
CGPI : int
+ShowBatches
Dashboard
EnrollmentNumber : String
marks: int
GPA : Double
+Displays marks:
+Displays registered student:
AddonStudentsList
+CourseID :string
+Semester : int
+Batch : string
+ Submit()
+ ViewList()
NewCourseRequest
+CourseID :string
+CourseName: string
+L T P Credits: int
+Semester : int
+Batch : string
+submit() : button
Aviral(Super admin)
Students
Faculty
HODs
ResultMenu
NewCouseReqest
MiniProject
Classes
MakeUpList
+CourseID :string
+CourseName: string
+Semester : int
+Batch : string
+ MakeUpList
+show() : button
ManageCourse
Course ID : String
Course Name : String
C1 Marks : Float
C2 Marks : Float
C3 Marks : Float
Total : Float
GPA : Double
Enrolled : Switch(Tab)
Completed : Switch(Tab)
Unregistered : Switch(Tab)
Dropped : Switch(Tab)
Faculty
+facultyID :string
+facultyName: string
+department: string
+email: string
+MiniProject
+Marks Upload: button
+ManageCourse
ComponentStatistics
+department: string
+Submission status: string
+choose submission: string
+generate report
+ choosse option:button

B. Dean AAA Section

1. Dashboard:

a. **Course session description –** Shows the description about courses.

b. **Faculty department description – Shows the** faculty allotted to each course

c. **Students program distribution – Shows the** distribution of students to each course

2. Courses: Displays all the available courses for each semester across B. Tech, M. tech, PHD.

3.Faculty:

a. **Department:** Displays all the Departments like IT, Add-on, 1st year uf First year, AS-Applied science, MBA-management studies, ALL-ALL, EC electronics and communication.

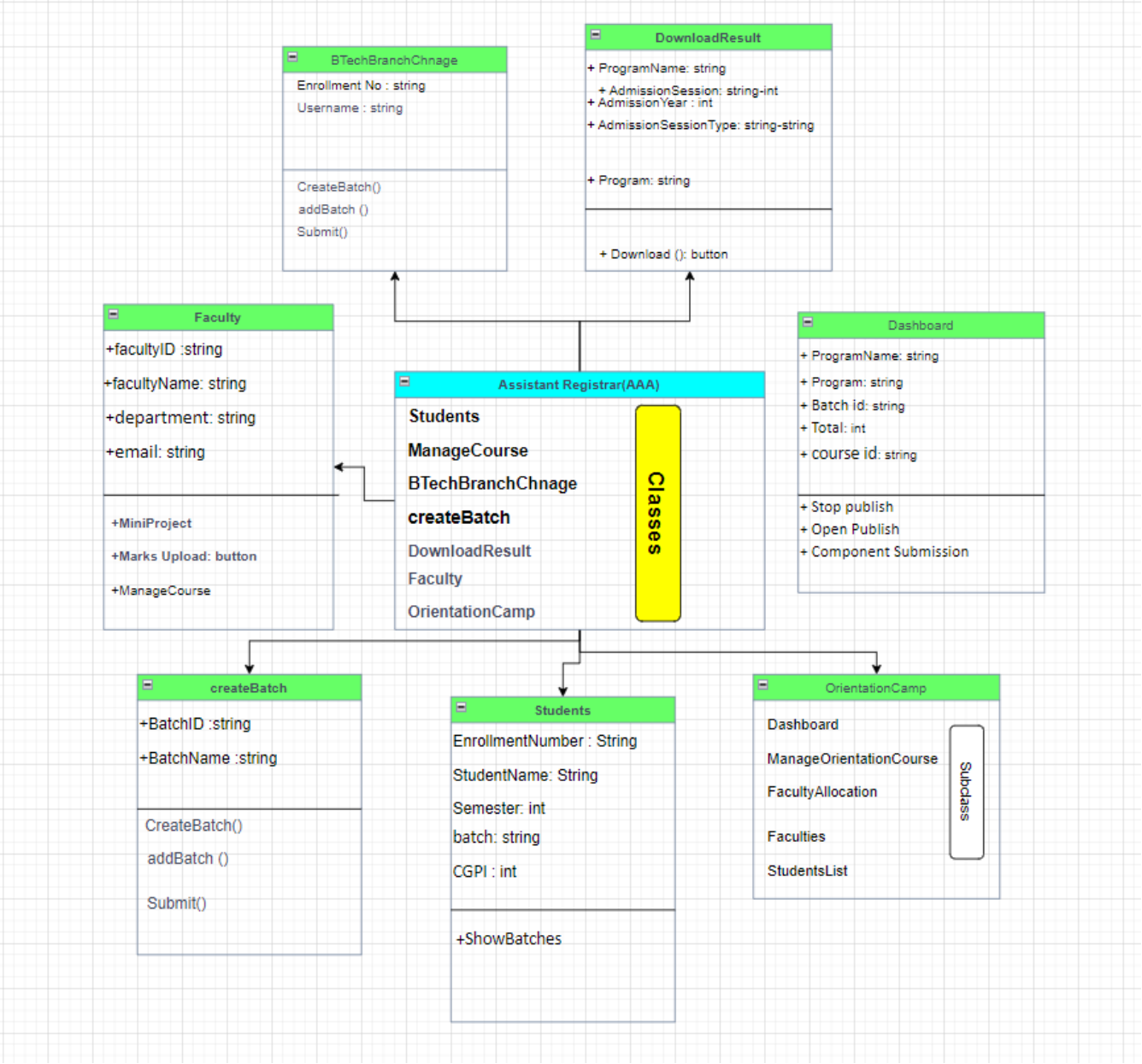
b. **Shows all the details of faculties like** faculty id, Name, Department, email, phone

4.Students:

a. **Batches:** Displays all the details of the batches.

b. Shows all the details of students like student id, Name, batch, completed credits, CGPI

Class Diagram:



C. HOD.As interface

1.Dashboard - Shows the course type description (Hard core & Elective)

2.Faculty allocation - Allotment of faculty (based on their specialization) to various departments is done by the HOD. We can Change the coordinator of the subject as well as add or delete from the system.

3. Download results - The HOD can download the result of students which consists of program, admission year, admission session type, admission session, download, result csv

4. Course:

a. **Core courses-** Can edit the course like add or import new course.

b. **Elective course-** Can edit the course like add or import new elective

5. Course curriculum

a. **Core courses:** Shows the details of the core courses

b. **Elective courses:** Shows the details of the elective courses

6. Elective menu

a. **Elective preference:** Show the details of choose elective course, choose preference, submit by the student

b. **Elective statistics:** Show the statistics of students and chosen elective course.

7. Drop and makeup

a. **Drop list –** Shows the details of students who didn't clear makeup.

b. **Makeup list -** Students who failed to clear c1 + c2 or c3 cut off.

c. **Component Statistics-** Choose component, choose department (it, etc), choose submission (submit, freeze), submission status (completed, not completed), generate report.

8. Details

a. **Courses:**

b. **Faculties:** department choose option, sl no, faculty id, Name, department, email, phone

c. **Students:**

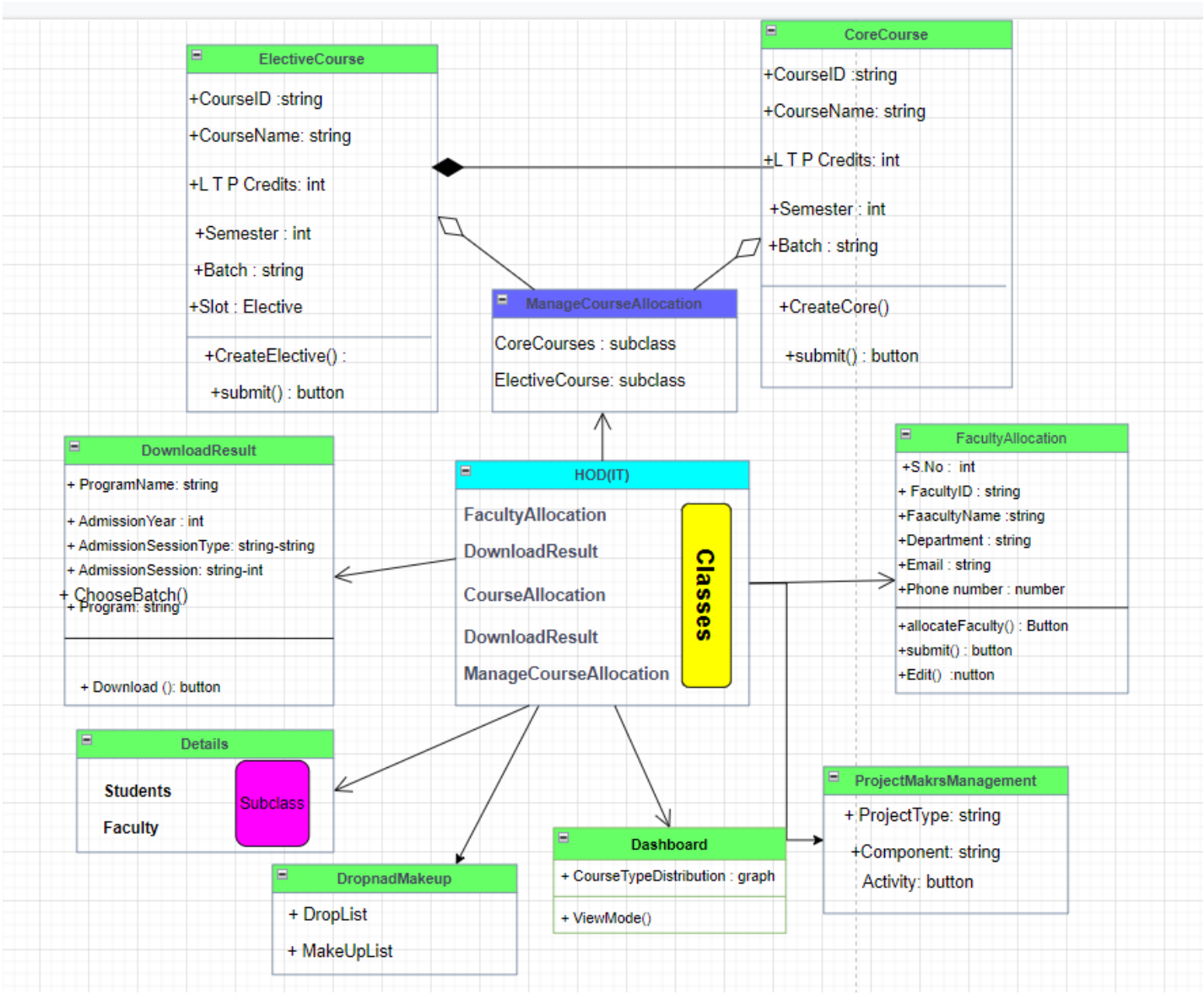
i. **Batches:** Displays all the details of the batches.
ii. Shows all the details of students like student id, Name, batch, completed credits, CGPI

9. Project marks management: Multiple faculties can submit project marks of the students. The HOD can See and upload as well as scale the marks.

10. Elective baskets – A basket is created by the HOD where 1 basket contains how many courses a student can opt in a year. The students then choose a course from a particular basket.

11. Elective selection- Show details of students with the courses selected them from the given basket.

Class Diagram:



D. faculty

1. Faculty name and Interest: Whenever a faculty logs in the "Aviral" his/her name will be shown in the top left along with their interest.

2. Dashboard-

a. **Current semester courses-** The faculty can upload the marks of students for C1, C2 & C3

b. **Course overview -** Details of components c1, c2 and c2 is shown in the dashboard along with Graph distribution of completed courses.

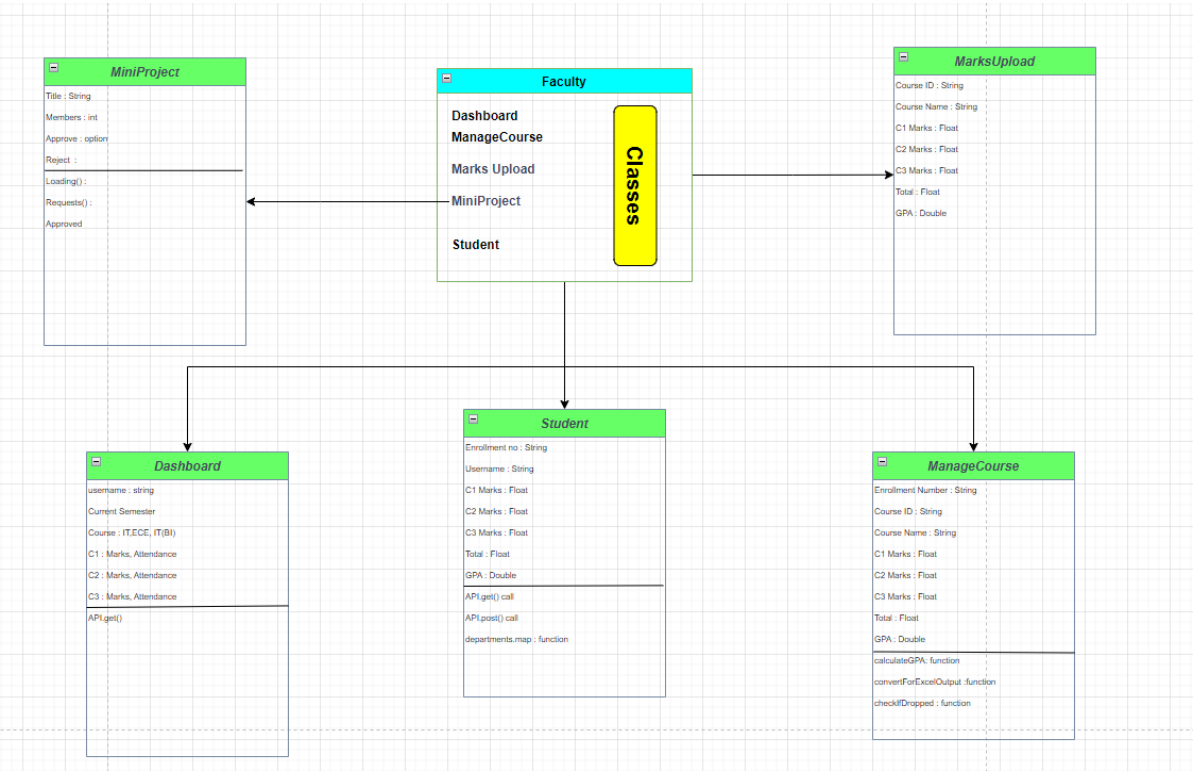
3. Active courses - Shows the List of courses which are currently running this semester

4. Projects – The faculty can either accept or reject the project and give the marks to the allotted students. All the other details like attendance etc. can also be uploaded and seen by the faculty. The faculty can only upload the marks when Projects marks opened by HOD.

5. Mini- projects – The faculty can either accept or reject the project and give the marks to the allotted students. All the other details like attendance etc. can also be uploaded and seen by the faculty. The faculty can only upload the marks when Projects marks opened by HOD.

6. Orientation courses- The faculty can give the orientation grade as well as range the same.

Class Diagram:



E. AD (addon) Admin

1. Dashboard - Shows the course distribution pie chart.

2.Active courses - Displays the details of which basket, batch and how many seats are allocated per subject.

3. All courses – Displays the details of all the courses like course id, course name, L, P, T, Total seat, batch and basket

4. Faculty allocation - Allocation is done by AD admin. Shows details like course id, course name, Faculty, section and also gives option of Delete.

5.Faculties - Shows the details of the faculties and their allotted courses and semester.

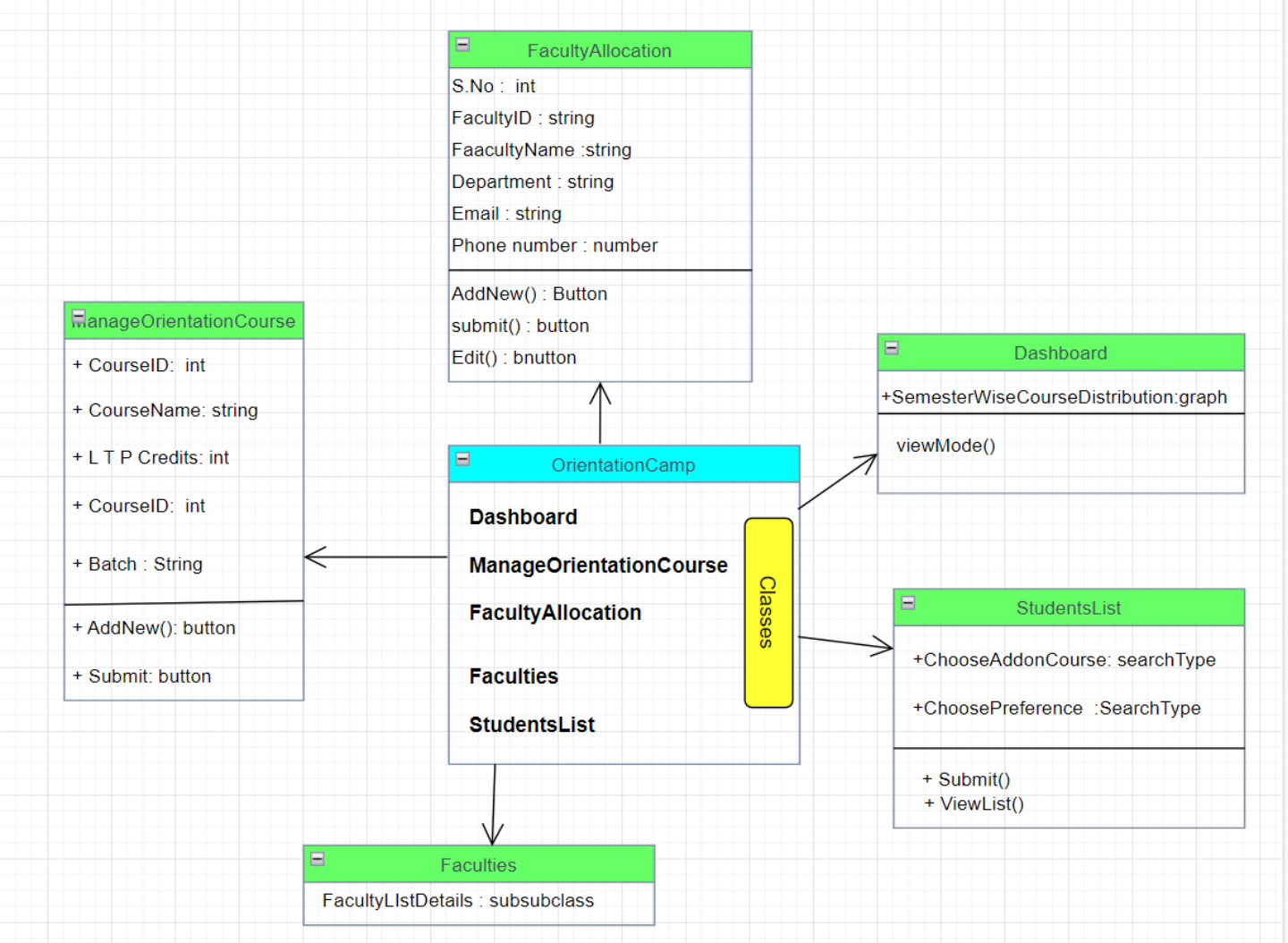
6.Addon Statistics - Shows the details of courses like course id, course name, opted 1st preference, opted 2nd preference

7.Student list - Shows details of which subject like Hindi, French etc. and how many had that subject as the 1st preference and 2nd preference.

F. OrentationCamp_Admin

1. **Dashboard –** Displays details of batch-wise course distribution
2. **Orientation course –** Consist of course id, Course name, L, T, P, Total
3. **Faculty Allocation -** Allocate faculty to particular course.
4. **Faculties-** Faculty ID, Faculty Name, Department, Email, Phone, Edit
5. **Students -** Select course - choose addon course-choose preference -submit- list of students enrolled.

Class diagram:



G. Aviral Administrator

1.Dashboard

a. **Overview-** Displays the details of c1, c2, c3, course registration

b. **Program wise registered students –** Displays the details of registered student program wise

2. Database admin

3. New course requests- Admin can request for a new course to be added in the curriculum. HOD can raise a query to upload a new course and after the approval of competent authority it can be uploaded. It is made optional.

4. Make-up list- Shows the details of students of different departments like IT, ECE, As, MGMT who are giving make-up test.

5. Import students - Can import details of students using batch name, choose file, upload

6. Import faculties - Can import details of faculties using choose file, upload.

7. Unregistered students - If students have not completed the course registration completely then the AAA will contact and get it done manually.

8. Addon statistics - Shows the details of course id, course name, opted 1st Preference, opted 2nd Preference. If there is a tie then higher CGPA students are given more preference.

9. Addon students - Shows the details of addon courses, choose preference of the students which is then submitted by admin.

10.Registered students - choose program

11. Component statistics statistics - choose component, choose department, choose submission type, Submission status - completed, not completed

12. Choose component, choose department (it, etc), choose submission (submit, freeze), submission status (completed, not completed), generate report.

13.Courses: Available courses for BTech, MTech, Instructor for courses

14. faculties: department choose option, sl no, faculty id, Name, department, email, phone

15 Students:

a. **Batches –** Shows the details of all the batches since the aviral was made

b. **Details –** Shows the details of students like student id, Name, batch, completed credits, CGPI as per the batch selected

H. students

A. Student Section

The name of the student is displayed along with its current semester. It also shows the BTech/MTech period of the student.

1. Dashboard

a) **Credits overview**
   i. Current C.G.P.A
   ii. Credits Completed
   iii. Core Credits Completed
   iv. Elective Credits Completed

b) **Program Progress**
   i. SGPI per Semester: In this section Graph of SGPI per Semester has been shown
   ii. Current Semester Credits Distribution

2. Course Registration - The student can either register for the upcoming semester or select blank semester

3. Mini-Project Registration - Here the students can fill the details of the mini-project like group members name, guide and topic name. Then the guide can either accept or reject the project.

4. Course Summary

i. Enrolled: Displays the details of subjects and its marks of the current semester**.**

ii. Completed: A list view all courses which has been completed

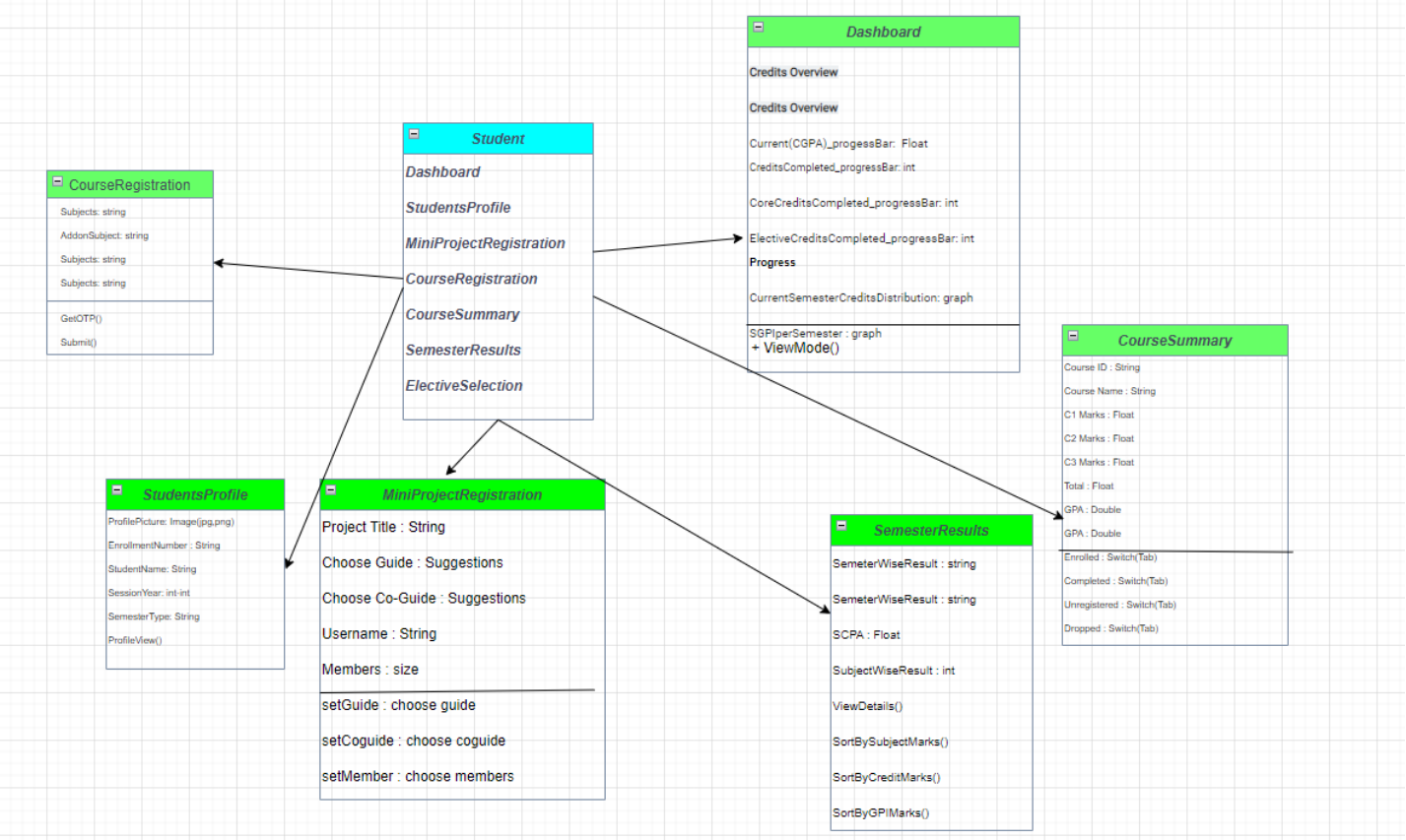
iii. Unregistered: Shows the details of subjects that are unregistered.

Iv. Dropped: Shows the details of subjects that are dropped.

5. Semester Results - In this page results of all semester are displayed in different cards view. Also, we can further see the details of specific semester like its subjects and marks of c1, c2, c3.

6. Elective selection - Shows the details of the electives chosen by the student.

Class diagram:



8. Other Nonfunctional Requirements

8.1 Performance Requirements

8.2 Hardware Requirements

- Standard PC
- Internet connection

8.3 Software Requirements

- Visual Studio Code

- React JS
- Web browser supporting HTML 5 and JavaScript
- Firebase

8.4 Software Quality Attributes

8.5 Design Constraint

8.6 Other Requirements

9. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List

10. Outcome

We will be creating a brand new mobile based application using modern UI/UX design for the higher authority. The main outcome of this will be the ease of use for the faculty and the admin. It will have a clean look with more features for fast and hassle-free management about details of

student. The code written will be in a structured format hence it can. Be reused again and again for adding new feature instead of writing new code and increasing the length of the code.

11. Future Scope

Previously, data used to be inserted manually to analyze result. In this project, we use RDBS as the database for extraction of data and presenting it the user interface. The future scope is that data can be fetched, parsed in other formats like doc, excel. Visualization can be provided to represent data in graphical format.

There is a future scope of this facility that many more features can be added such as group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfils each user need in the best way possible. We can even add the facility to have “Aviral” mobile app for Dean, HOD and Director.

12. Reference

- Available: https://en.wikipedia.org/wiki/Student_information_s ystem [2] [online] Available: https://en.wikipedia.org/wiki/Systems_analysis
- Chew Li Sa,Dayang Hanani bt. Abang Ibrahim,Emmy Dahliana Hossain,Mohammad bin Hossin, Student performance analysis system (SPAS),2015.
- Ishwank Singh,A Sai Sabitha,Abhay Bansal,Student performance analysis using clustering algorithm,2016.
- Sérgio Gonçalves ,Davide Carneiro, Javier Alfonso, Florentino Fdez-Riverola, Paulo Novais, Analysis of student's context in e-Learning, 2015.

13. Panel Comments/Suggestion