College Management System

1.0 Problem statement

CMIS may be a software system developed for a university or a college for maintaining information associated to varied daily activities within the college. The CMIS is utilized to keep details of various departments in the university or college, varied courses offered by the school or college, student admission details, faculty details, day-to-day attendance details, their salary details, end exam marks details, internal marks details; laboratory infrastructure details and then on.

By building these information, the CMIS software system creating reports that are helpful for office staff, faculty, principal, HOD’s, and for management within the administration.

By using the details, we create report details and send to the parents about the student’s progress and status. So this is often complete and comprehensive software solution that gives number of solutions and services for managing data related to day-after-day activities of the company or the organization.

Scope of the System

The scope of the system is explained through its modules as follows

- User Registration and Login - The existing users can login to the application directly and the new users should click the **New User? Create Account** link to register themselves.

  When the user clicks on the **New User? Create Account** link, it should re-direct to the registration page, where the user needs to fill in some of the basic attributes/fields mentioned below

  - Email Address
  - Password
  - Confirm Password

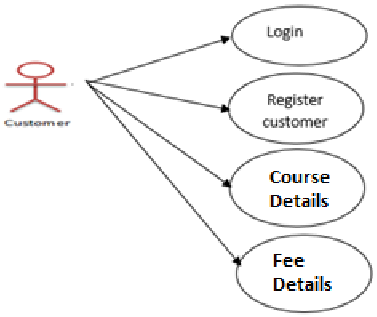
  If the email ID given by the user already exists in the database, the following message is displayed: “User is already registered. Please login to the application.”

- Student Marks Details – A user should be able to search for marks and display the marks.

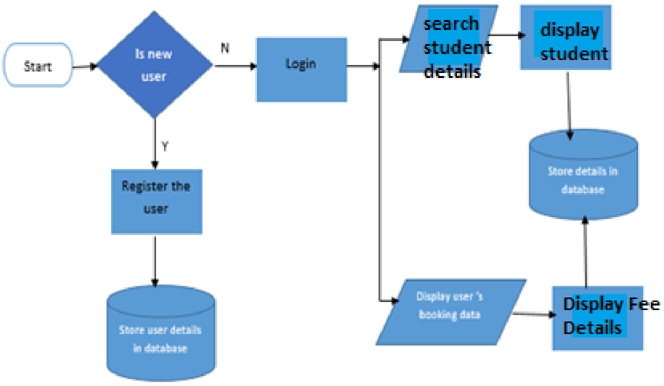
Course Details : This module should be able to display all the courses and its particulars

Fees Details : This module aims at displaying the fee paid, balance due by the students.

2.0 Use Case Diagram



Flow Diagram



3.0 Project Development Guidelines

The project to be developed based on the below design considerations

Backend Development	● Use Rest APIs (Springboot/ASP.Net Core WebAPI to develop the services ● Use Java/C# latest features ● Use ORM with database ● Use Swagger to invoke APIs ● Use JWT for authentication in SpringBoot/WebApi. A Token must be generated using JWT. Tokens must expire after a definite time interval, and authorization must be handled accordingly based on token expiry ● Implement Logging ● Implement API Versioning ● Implement security to allow/disallow CRUD operations ● Message input/output format should be in JSON (Read the values from the property/input files, wherever applicable). Input/output format can be designed as per the discretion of the participant. ● Any error message or exception should be logged and should be user-readable (not technical) ● Database connections and web service URLs should be configurable ● Implement Unit Test Project for testing the API ● Follow Coding Standards
Frontend Development	● Use Angular/React to develop the UI ● Implement Forms, databinding, validations ● Implement Routing and navigations ● Use JavaScript to enhance functionalities ● Implement External and Custom JavaScript files ● Implement Typescript for Functions, Operators. ● Any error message or exception should be logged and should be user-readable (and not technical) ● Follow coding standards ● Follow Standard project structure

4.0 Good to have implementation features

- Generate a SonarQube report and fix the required vulnerability
- Use the Moq framework as applicable
- Create a Docker image for the frontend and backend of the application
- Implement design patterns
- Deploy the docker image in AWS EC2 or Azure VM
- Build the application using the AWS/Azure CI/CD pipeline. Trigger a CI/CD pipeline when code is checked-in to GIT. The check-in process should trigger unit tests with mocked dependencies
- Use AWS RDS or Azure SQL DB to store the data