DATA FOUNDATION SYSTEMS PROJECT

STUDENT PROJECT REVIEW SYSTEM

TEAM PIXEL

PRATHAM MISHRA , ANUSHKA AGRAWAL , KHUSHI WADHWA

Content

INTRODUCTION

We develop a system that facilitates the assessment and evaluation of various projects hosted on GitHub. These projects traverse various phases, such as requirement gathering, design, coding, testing, deployment, and presentation.

01 UNDERSTANDING PROJECT REQUIREMENTS AND EXPECTED OUTPUT

02 DESIGNING A COMPREHENSIVE AND WELL-ORGANIZED BACKEND FOR THE PROJECT

03 DESIGING A USER-CENTERED AND VISUALLLY APPEALING FRONTEND

04 INTEGRATING ANG TESTING THE FRONT END AND BACKEND

APPROACH

BACKEND - SCHEMA

Users Table: stores the information of all the users including the students, professors.

- user_id: a unique id (like roll no given to every user that acts as a identifier
- name: the name of the user
- email: email id of the user which is also unique
- user_type: the type of user, whether student or professor

Sem_courses Table: stores information about courses offered in the semester. Each row represents a single course with the following details:

- **course_code:** unique code identifying the course (e.g., CS101)
- **course_name:** name of the course (e.g., Introduction to Programming)
- **professor_email:** email of the professor teaching the course

BACKEND - SCHEMA

Course_ Table: each course will have a independent table storing the information of all the students pursuing the respective course.

- user_id: id of the student pursuing the course
- role: role of the user (eg: student, professor, assistant prof)
- name: name of the user
- email: email of the user
- team_id: id of the team the student is part of. (if the user is not a student, this cell will be NULL)
- project_id: the id of the project assign to the team the student is part
- task1 - task5: the grades obtained in the tasks by each students (this allows individual grading)
- ta_userid: userid of the TA who will mentor the student
- final_grade: the final grade obtained by the student
- github_link: the github link of the repository for the project

BACKEND - SCHEMA

Course1_comments Table: corresponding to each course, we will have a comments table that contains the following deatils:

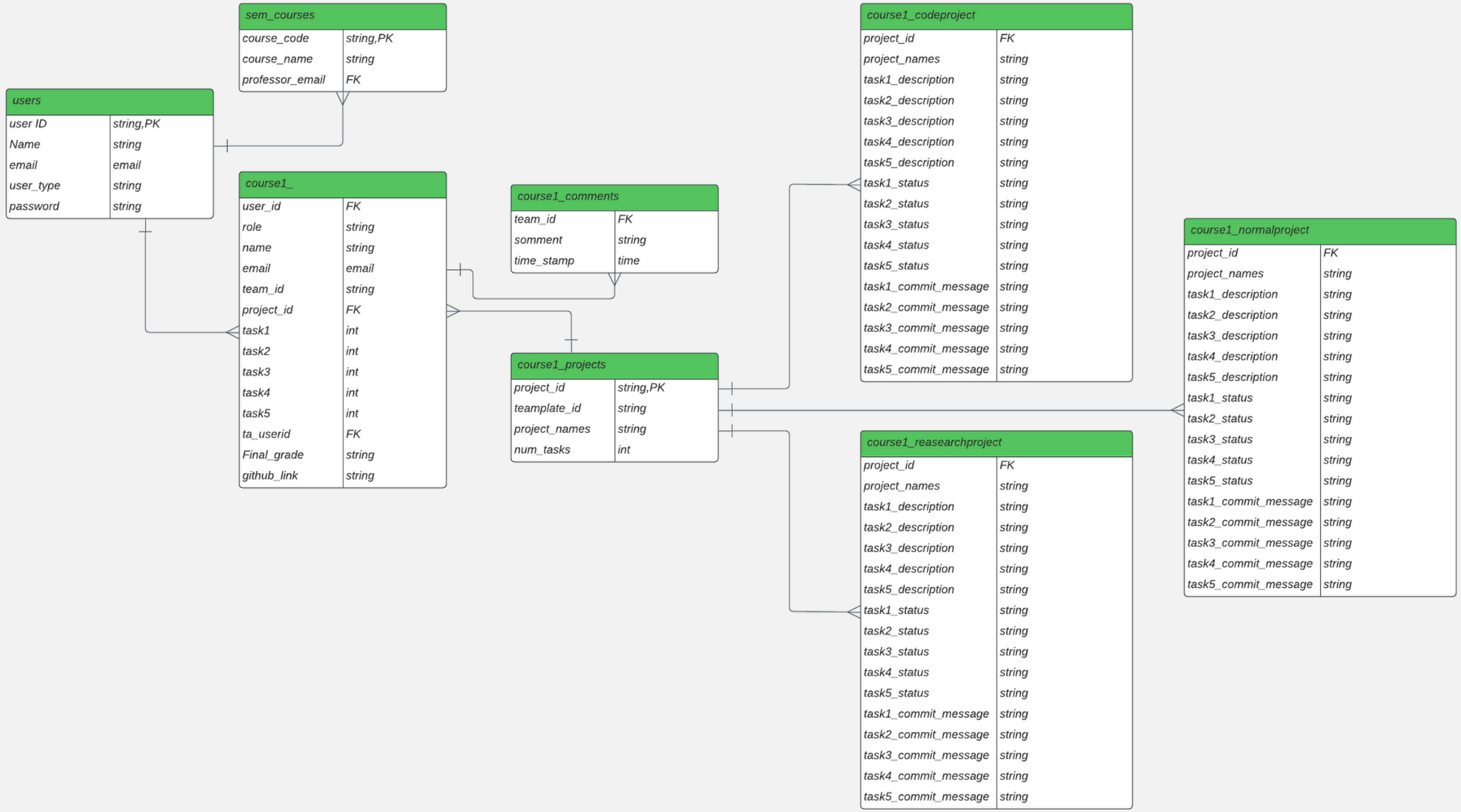
- team_id: id of the team for which the comment corresponds to
- comment: the comment/feedback from the prof or TA to the team
- time_stamp: the time at which the comment was given
-

Course1_projects Table: corresponding to each course, we will have a projects table that contains different projects offered in the course.

- project_id: unique id given to each project
- project_name: name of the project
- template_id: the id of the template for the project
- num_tasks: no of tasks the project is going to have

Course1__template1project Table: there is a default table for the exisitng template that provides the discription of the tasks for each project, status of each task and the commit-message for each task that will be used to track the completion of the task.

BACKEND - SCHEMA



FRONTEND

LOGIN/REGISTER

Student Dashboard

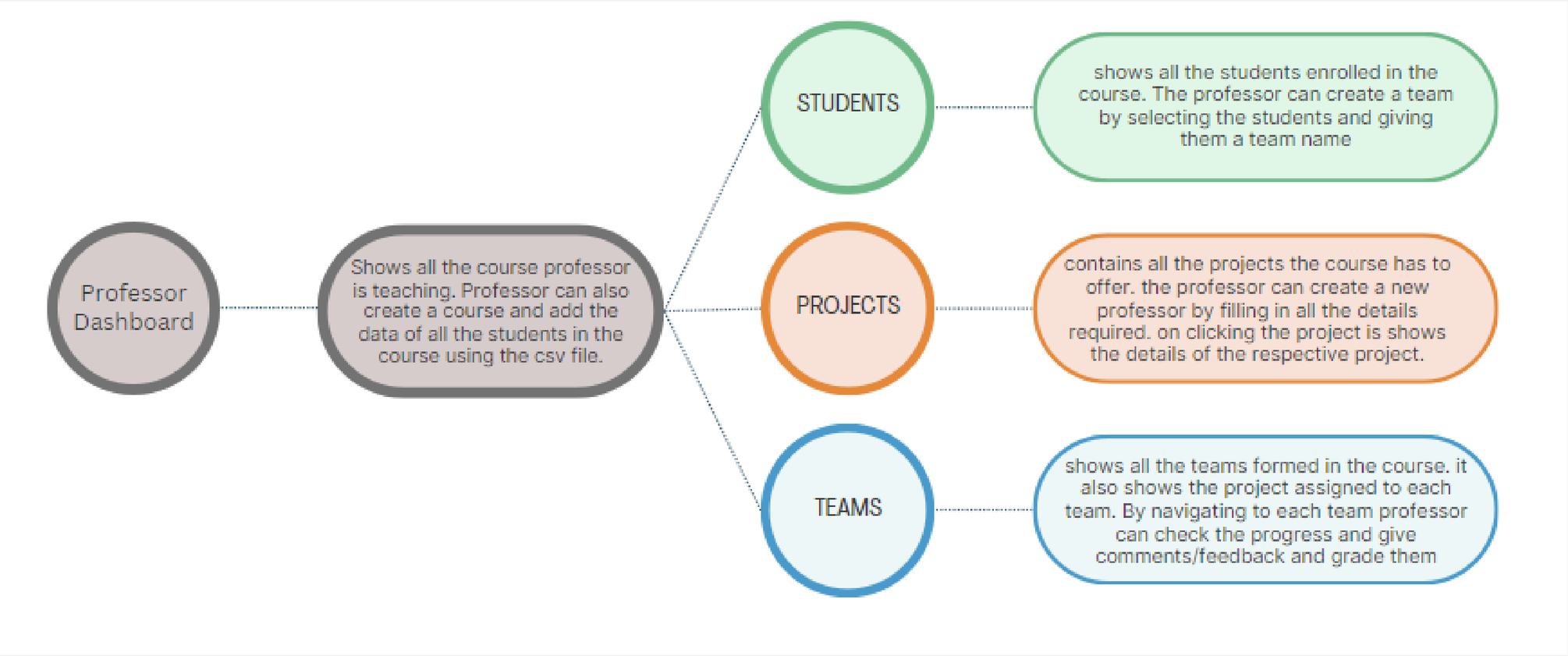
On logging in, the student will see a dashboard. This will display all the courses the student has be enrolled in. The student can go to the particular course which will then display the project he is working on. The project will display all the team members working alongside him and the progress they have made so far. It will also display the reviews and remarks the professor/TA has provided.
if the student is TAing any course , he can access that courses' dashboardthrough the TA tab

Professor dashboard

When professors log in, they will see a dashboard. Professors can see all the courses they are teaching and can also add new course.They also have the option of adding student data to a course. They click on the course and they can see all the projects the course offers and also also the students that are enrolled in the course. The prof can create teams and assign them projects. And for each team, the prof can grade them and add comments to communicate their feedback.

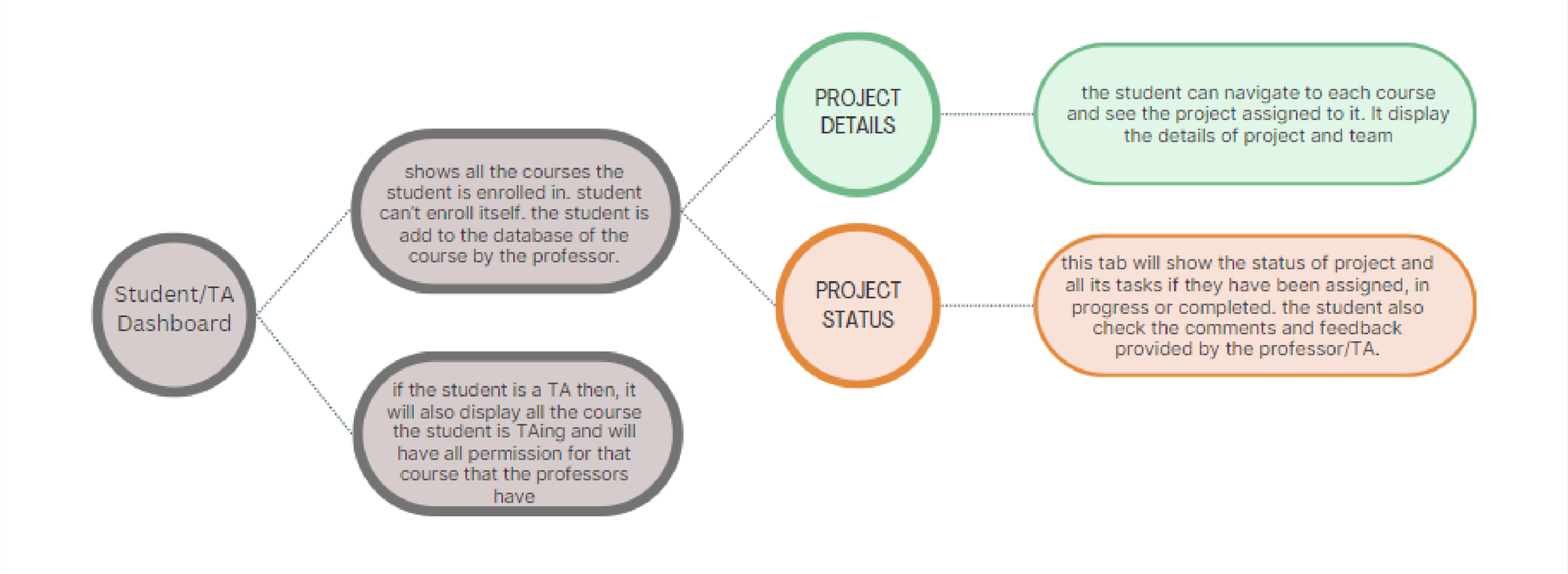
FLOW CHART

Professor Dashboard



FLOW CHART

Student/TA Dashboard



USE CASES

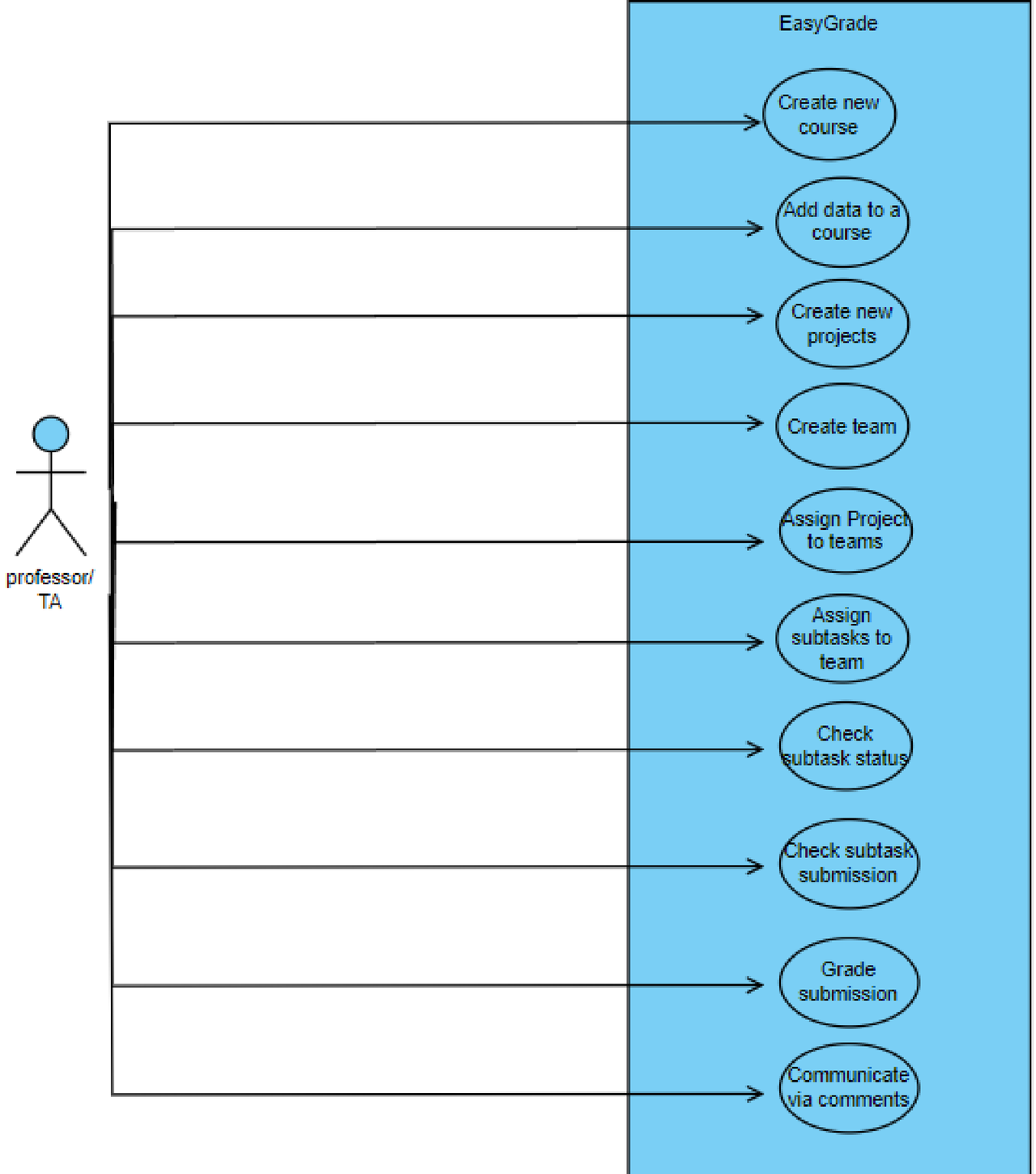
Professor/TA End

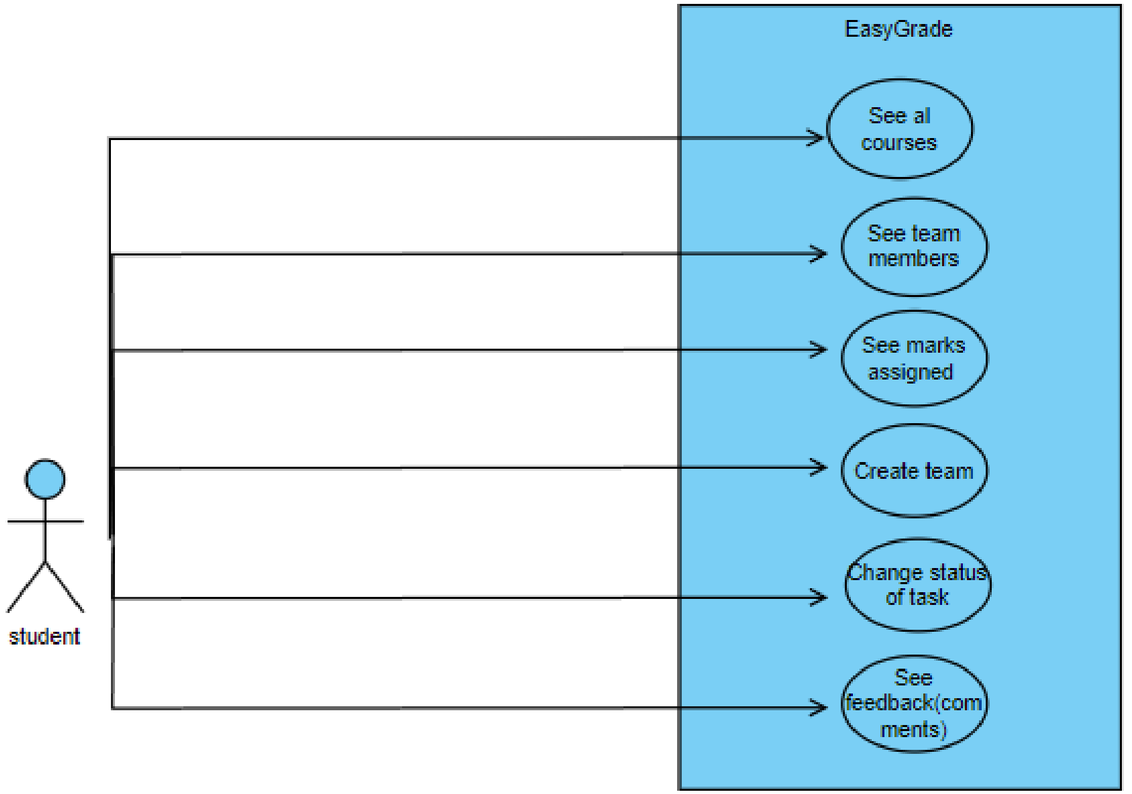
- **User Management:**
  - Ability to add students, teaching assistants (TAs), and fellow professors to the system.
  - Users are identified by their email addresses and assigned roles.
- **Team Creation:**
  - Capability to create teams of students for collaborative projects.
- **Project Assignment:**
  - Professors can assign custom projects to the created teams.
  - Each project is defined from scratch, including task descriptions, subtasks, and other relevant details.
- **Project Flexibility:**
  - Projects can be assigned to more than one team.
  - However, a specific team can only be assigned one project.
- **GitHub Integration:**
  - Monitoring of GitHub submissions for each project.
  - Real-time tracking of the status of individual subtasks.
- **Grading Functionality:**
  - Professors can grade subtasks and the overall project.
  - Option to grade the entire group uniformly or grade each student within a team separately.
- **Communication with students via Comments**
  - The professor and tas can share feedback, ideas or suggestions using comments
- **Final Grade Determination:**
  - Similar flexibility in determining the final grade for the project.

Student End

Key Features:

- **Course Overview:**
  - Provides a consolidated view of all enrolled courses.
- **Team Management:**
  - Enables access to a detailed list of team members within each course, along with their contact details
- **Transparent Grading:**
  - Offers transparent and detailed insights into marks assigned for every subtask.
  - Enables students to track their progress and identify areas for improvement.
  - the professors can share ideas / feedback via comments
- **Dedicated TA Access:**
  - Features a dedicated "TA" section for students serving as Teaching Assistants.
  - Provides streamlined access to TA-specific functionalities and resources.
  - Enables direct navigation to the TA/Professor dashboard for specific courses.


EasyGrade
Create new course
Add data to a course
Create new projects
Create team
Assign Project to teams
Assign subtasks to team
Check subtask status
Check subtask submission
Grade submission
Communicate via comments
professor/ TA


EasyGrade
See al courses
See team members
See marks assigned
Create team
Change status of task
See feedback(comments)
student

TIMELINE

ist week of November

Completed student side of backend apis , as well as the corresponding frontend

Mid November

Completed the professor side of backend , and tested all the Api calls involved

By end of november

Created the frontend for the professor dashboard, and integrated it with the backend

1st week of december

Added the github functionality , and tested the final product

CHALLENGES FACES

- **Unfamiliarity with SQL for Backend:**
  One of our initial challenges was our limited experience in using SQL as the backend database. We needed to acquire a solid understanding of SQL to effectively manage the database.

- **Complex Database Schema Creation:**
  Designing the comprehensive database schema proved to be a challenging task. Defining the structure of the entire database, including its various tables and relationships, required careful planning and organization.

- **Inadequate Senior Codebase:**
  We encountered difficulties with the existing codebase left by our seniors. The codebase was not performing optimally, which prompted us to undertake a complete rewrite from scratch.

- **GitHub API Integration**
  Integrating the GitHub API presented another hurdle as it was a new and intricate technology for us. Navigating the GitHub API's intricacies and effectively using it in our project required a significant learning curve.

- **Change from Nod.js to Fastapi**
  We had to change our beckend from node to Fastapi , which caused it to have to do some parts twice. We were also unfamiliar wth fastAPi , and had to learn it .

THANK YOU!