Problem Statement: University Course Management System

Overview:

The goal of this project is to design and implement a comprehensive University Course Management System. The system will keep records of students, lecturers, advisors, courses, transcripts and many other elements of a typical University system. The system will allow students to search for courses, check their availability, and select courses. Additionally, the system will track students' transcripts. Multiple sections can be opened for each course. Each section represents a different part of the same course given at different time. Students can enroll in any section of the course, providing them with more flexibility and aiding in the better management of courses.

Project Features:

- A tool for students to search for and select courses
- A system that tracks students' transcripts
- A tool for advisors to manage students and courses.
- Allow different sections to be selected for each course.

Purpose

Making Course Registration a Breeze for Students and Advisors

Embarking on this Java project, I envision crafting a Course Registration System (CRS) that's not just a bunch of code but a helpful tool for students and advisors in our department. Think of it as a user- friendly space where students can easily sign up for courses and advisors can guide them through the process. As we start, we'll have two main users - students and advisors, but we're thinking ahead to include more roles like department head, admin, and student affairs down the line.

Goals we're aiming for:

Easy Sign-in:

Logging in should be a snap. Students and advisors will use their usernames and passwords to get into the system.

Classes Doing the Heavy Lifting:

Behind the scenes, we'll have different classes doing different jobs - Courses, Sections, Staff, Lecturers, Advisors, Students, Grades, Registrations, Transcripts, and a big boss - CourseRegistrationSystem. It's like having a cast of characters each doing their part.

Smooth Registration:

Registering for courses will follow the rules we have in our department. The system will chat with users, ask for the right details, and make sure everything's on track.

User-Friendly Vibes:

Students and advisors won't need a tech manual. The system will talk to them in a way that's easy to understand, with pop-ups and messages guiding them through each step.

Growing with Us:

As things change or we want to add more features, our CRS will be ready. We're building it to handle new roles and cool stuff we might want to throw in later.

Rules, but not the Boring Kind:

Registration will follow our department's rules, but we're making it interesting. The system will check for prerequisites and other rules without being a buzzkill.

Locking it Down:

Nobody wants their stuff out in the open. The system will make sure that only the right people get in and see the important info.

Teamwork and Tweaks:

We are not doing this alone. It's a team effort. We'll be tweaking and improving things based on feedback from the classroom. It's a work in progress, and that's the fun part.

So, in a nutshell, we're building a friendly, rule-following, and ever-evolving Course Registration System that's here to make life easier for students and advisors alike. Let's make registering for courses as smooth as possible!

GLOSSARY

Term	Definiton
User	User can be a student or a staff(lecturer or advisor). Each user has his/her own username and password and logins the system.
Student	Students are identified by their IDs. A student has his/her own course list, transcript, grades and these properties are viewable to themselves. Each student is assigned to an advisor.
Staff	A staff can be lecturer or/and advisor. Unlike the students, a staff has permission to access the system and can make changes in the system.
Lecturer	Staff member who assigned in courses to lecture students. Lecturers have titles such as professor,asisstant etc.
Advisor	Staff member whom students can consult. Advisor takes care of students course selection and approve or reject the selections. Every advisor a lecturer but every lecturer doesn't have to be an advisor.
Transcript	Official document that details a student's academic performance. It typically includes information such as courses taken, grades received, and cumulative grade point average (GPA).
Grade	Grade is the score that students receive from courses. There is also letter grade to determine whether the student pass or fail the course.
Course	A series of lectures. Courses have quotas, credits, types (must/elective).Each course may be divided into multiple sections (CourseSection).
Course Section	Each course section is given by one lecturer. Sections of a course should be at different classrooms and different times of a week to avoid overlapping.

Functional and Non - Functional Requirments

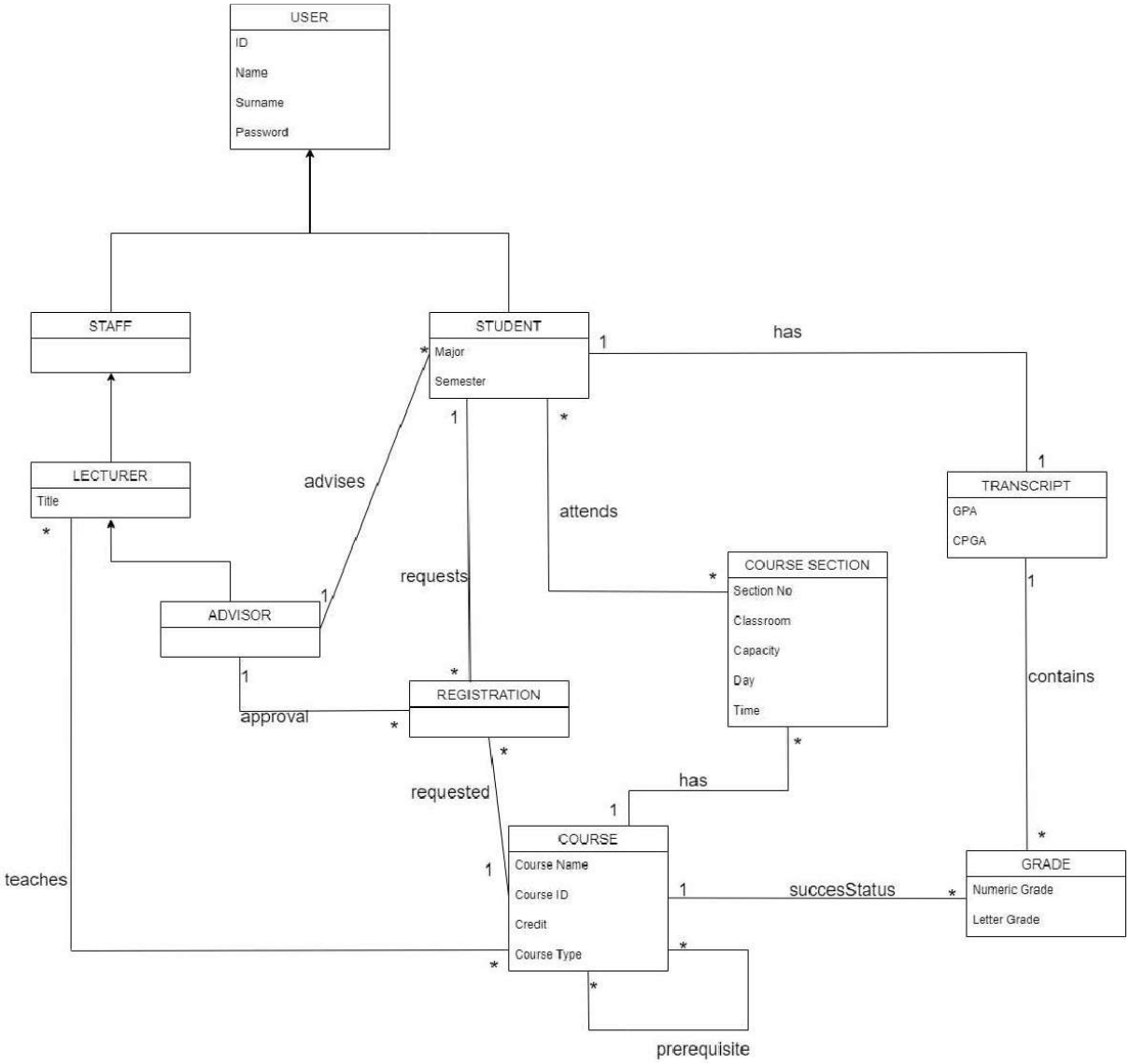
Functional Requirments

#	Requirement	Priority (Must / Want)	Description
1	Student Login	Must	Students can log in to the system with their school numbers and passwords.
2	View Information	Must	Students can view and change their personal information. (e-mail, phone number etc.)
3	View Transcript	Must	Students can look at their transcripts for topics such as work, internships, etc.
4	View Grade	Must	Students can look their grades for grade point average, letter grade.
5	Prerequisite tree	Must	According to the prerequisite tree, students check which course they can or cannot take.
6	Course Selection and Confirmation	Must	Students can choose courses. Advisor has the authority to approve course selections.

Non - Functional Requirments

#	Requirement	Priority (Must / Want)	Description
1	Performance	Must	The system must provide a responsive and efficient experience for students when accessing their grades and transcripts.
2	Usability	Must	The interface must be intuitive and user-friendly, allowing students to easily navigate through the system. Clear and concise design elements should enhance the overall usability of the platform.
3	Reliability	Must	The system should provide low error rates and high continuity.
4	Scalability	Must	The system should be scalable to support increases in the number of users.

Domain Model



Use Cases

Use Case: Enroll in courses
Actors: Student, System
Goal: To select the courses to send the approval of advisor in order to enroll in courses
Preconditions: Student is logged into the system
Steps:
Actor Actions
1- start course registration
3- select the courses
4- send selections for advisors approval
System Responses
2- display the courses student can be enrolled
5- display the confirmation that the registration has been sent for advisors approval

Use Case: Advisor accepting or rejecting course selection.
Actors: Advisor, System.
Goal: To accepting or rejecting the courses which students sends for course request to advisor.
Preconditions: Advisor is logged into the system, student send course request to advisor.
Steps:
Actor Actions

1- Select "Course Requests" option
3- Select student who to be looked at
5- Select accept or reject button for specific course

System Responses

2- Display all students who send course requests
4- Display course requests from selected student
6- Display course is successfully accepted/rejected on screen

System Sequence Diagrams