CSE-3063

OBJECT-ORIENTED SOFTWARE DESIGN

ITERATION - 1

GROUP 13

DEVELOPERS:

- 150120027 Mert Muslu
- 150120051 Erkut Dönmez
- 150121539 Gülsüm Ece Günay
- 150119777 Ayşegül Bihter Banuşoğlu
- 150119729 Kağan Boyacıoğlu
- 150119767 Furkan Bozkurt
- 150119019 Ahmet Bilal Karabulut

VISION

In this project, we create a simulation of the Student Registration System. Policies and guidelines are inspired by the Marmara University Computer Engineering Department.

PROBLEM STATEMENT

The aim of this project is to help students to add courses to their schedules. After the student login the system successfully, it allows students to authenticate, and enroll in courses. Courses are taken according to the class of the student.

The primary objectives of the Course Registration System are:

- To authenticate students and provide them with the ability to enroll in courses.
- To maintain information about courses, lecturers, and students.

FUNCTIONAL REQUIREMENTS

Authenticated students should be able to:

I. Retrieve course data from the courses.json file.
II. List courses based on the academic year they belong to.
III. Associate each student with a specific academic year.
IV. Display a list of courses according to the academic year of the student.
V. Enable course enrollment for each student, listing available courses based on their academic year.
VI. Add selected courses to the student's course list.
VII. Store enrolled courses for each student in an ArrayList.
VIII. Generate a JSON file with a .json extension for each student, including their enrolled courses.

The system must provide a mechanism for students to:

I. Enter a username and password for authentication.
II. Successfully authenticate against stored student information.

NON-FUNCTIONAL REQUIREMENTS

I. The system is implemented using the Java programming language.
II. Diagrams for the project were created using Draw.io.
III. We write our documents using Google Docs.
IV. No Graphical User Interface was used in this project.

USE CASE

Use Case: Course Registration System

Actors:

- Student

Main Flow:

Student Registration:

a. The student enters the Course Registration System.
b. The system prompts the student to enter their username and password.
c. Student provides valid credentials.

Authentication and Course Selection:

d. The system authenticates the student.
e. The authenticated student chooses courses from the available list based on their academic year.

f. The system validates course eligibility and updates the available course list.

Enrollment Submission:

g. Student submits the selected courses for approval.
h. The system compiles the selected courses and generates JSON files for each student.

Course Enrollment Confirmation:

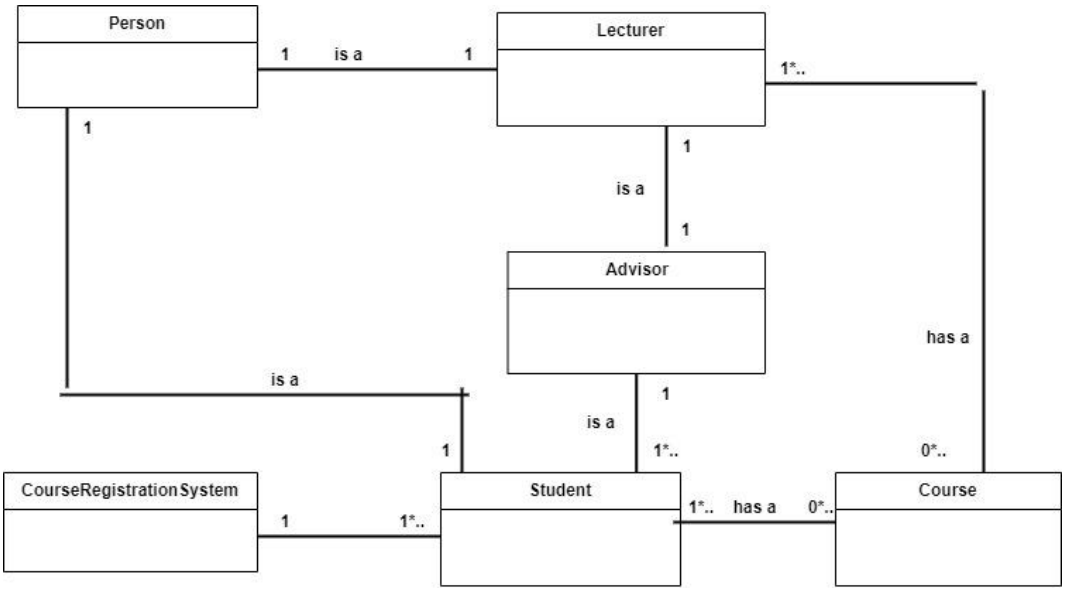
i. The system confirms the enrollment of submitted courses.
j. Student receives confirmation of successfully registered courses.

GLOSSARY

- **Advisor**: A person responsible for guiding students on enrolling in courses.
- **Course**: Lessons that students need to pass for graduation.
- **Credit**: Recognition for completing a course at school or university.
- **Course:** Courses that students must take.
- **Lecturer:** A person that gives the courses, educator.
- **Course Registration:** The action or process of registering or being registered for courses.
- **GPA:** Abbreviation for "Grade Point Average," representing the average value of grades earned in courses over time.
- **Java**: Class-based, object-oriented programming language used for system development.
- **JSON File:** A file format for storing and transferring data in JSON (JavaScript Object Notation) format.
- **Person:** Superclass of student and advisor with name, surname, email, and phone number.

- **Prerequisite:** Course or requirement a student must have taken before enrolling in a specific course or program.
- **Schedule**: A class holding the day and course time of the courses taken by the student.
- **Student**: The main character of the course registration system under the Person class.
- **Student ID:** Unique ID assigned to students.
- **Login**: A class responsible for handling the authentication process.

DOMAIN MODEL



SYSTEM SEQUENCE DIAGRAM