CSE-3063

OBJECT-ORIENTED SOFTWARE DESIGN

ITERATION - 2

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. System aims to help students to register and reach their stored academic informations. Moreover, it creates smooth environment for advisors for registration. Policies and guidelines are inspired by the Marmara University Computer Engineering Department.

SCOPE

The scope of this project is to construct a representation of the Student Course Registration System for Marmara University's Computer Engineering Department. This simulated version will consist of the procedures and regulations that students need to follow during course registration, in accordance with the department's policies. It is designed as an educational resource for students, with a focus on simplicity to ensure easy usability for everyone.

PROBLEM STATEMENTS

The aim of this project is to help students to add courses to their schedules. After the person sucessfully login the system based on their identity, system authenticate them. If person is authenticated as a student, s/he can register and enroll the courses, If person is authenticated as a lecturer, it uses the system to approve the registration.

The primary objectives of the Course Registration System are:

- To authenticate person as a student or advisor, and assign them a authority to use the system as a their authentication.
- Authenticating as a students and provide them with the ability to enroll in courses.
- Authenticating as advisors and provide them with the ability to approve or not approve the taken courses.
- To maintain information about courses, lecturers, and students.

FUNCTIONAL REQUIREMENTS

- Students are created by using a json file for each semester.
- Advisors are added by using a json file
- Students are defined in advisor json file
- The system must make sure that prerequisites for a course are taken before adding it to a student.
- Advisor has a authentication to remove courses from student's transcript.
- The system must provide a mechanism for person to:
    I. Enter a username and password for authentication.
    II. Successfully authenticate against stored information.
- Add selected courses to the student's course list.
- Store enrolled courses for each student in an ArrayList.
- Generate a JSON file with a .json extension for each student, including their enrolled courses.

NON-FUNCTIONAL REQUIREMENTS

- The system should be easy for students and staff to use without needing a lot of training.
- The system should be working almost all the time so that students can register when they need to.
- The system should be able to handle lots of students using it at the same time, especially during busy registration periods
- The system should do things the way Marmara University's Computer Engineering Department wants, following their rules and guidelines.
- The system should make sure that student data doesn't get mixed up or lost during registration.
- Students should be able to create and remember their passwords easily, without too many rules or restrictions.
- The system is implemented using the Java programming language.
- Diagrams for the project were created using Draw.io.

- We write our documents using Google Docs.
- No Graphical User Interface was used in this project.

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.
- **Transcript**: A student transcript is an official document that provides a comprehensive record of a student's academic history
- **Semester:** A specific academic term within a school or university year, numerical value.

- **Technical elective:** A technical elective is a course in a specific technical or specialized area,
- **Non-technical elective:** is a course outside the major field of study
- **Authentication**: Authentication is the process of verifying the identity of a user
- **Unit Test:** A test that evaluates the individual components.

USE CASE

Use Case: Course Registration System

Actor:

- User (Student/Advisor)

Basic Flow:

1. User Authentication:

a. The user enters the system.

b. The system prompts the user to identify themselves as a student or an lecturer.

c. The user selects their role (student or lecturer).

d. The system then guides the user through the authentication process based on their chosen role.

2. Course Registration (Student):

a. If the user identifies as a student:

i. The system directs them to the course selection screen after successful authentication.

ii. The student chooses their desired courses.

iii. Upon completing the course selection, the student seeks approval from their advisor.

3. Advisor Approval (Advisor):

a. If the user identifies as an advisor:

i. The system directs them to the authentication process after choosing the role.

ii. After successful authentication, the advisor sees a list of students who have selected courses and are under their guidance.

iii. The advisor reviews the list of courses selected by each student.

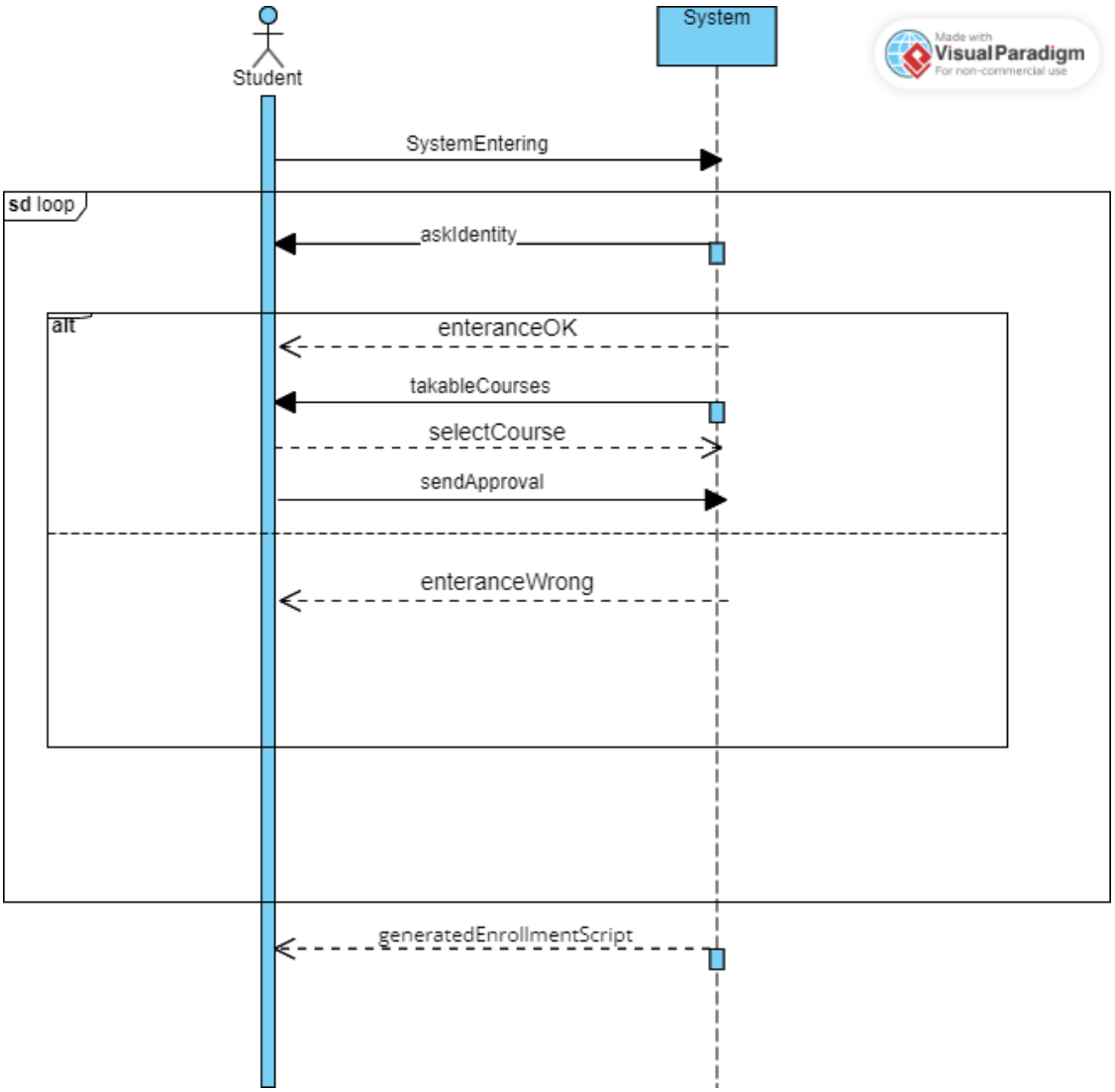
iv. The advisor has the option to approve or reject the selected courses for each student.

v. If approved, the system finalizes the course registration for the student.

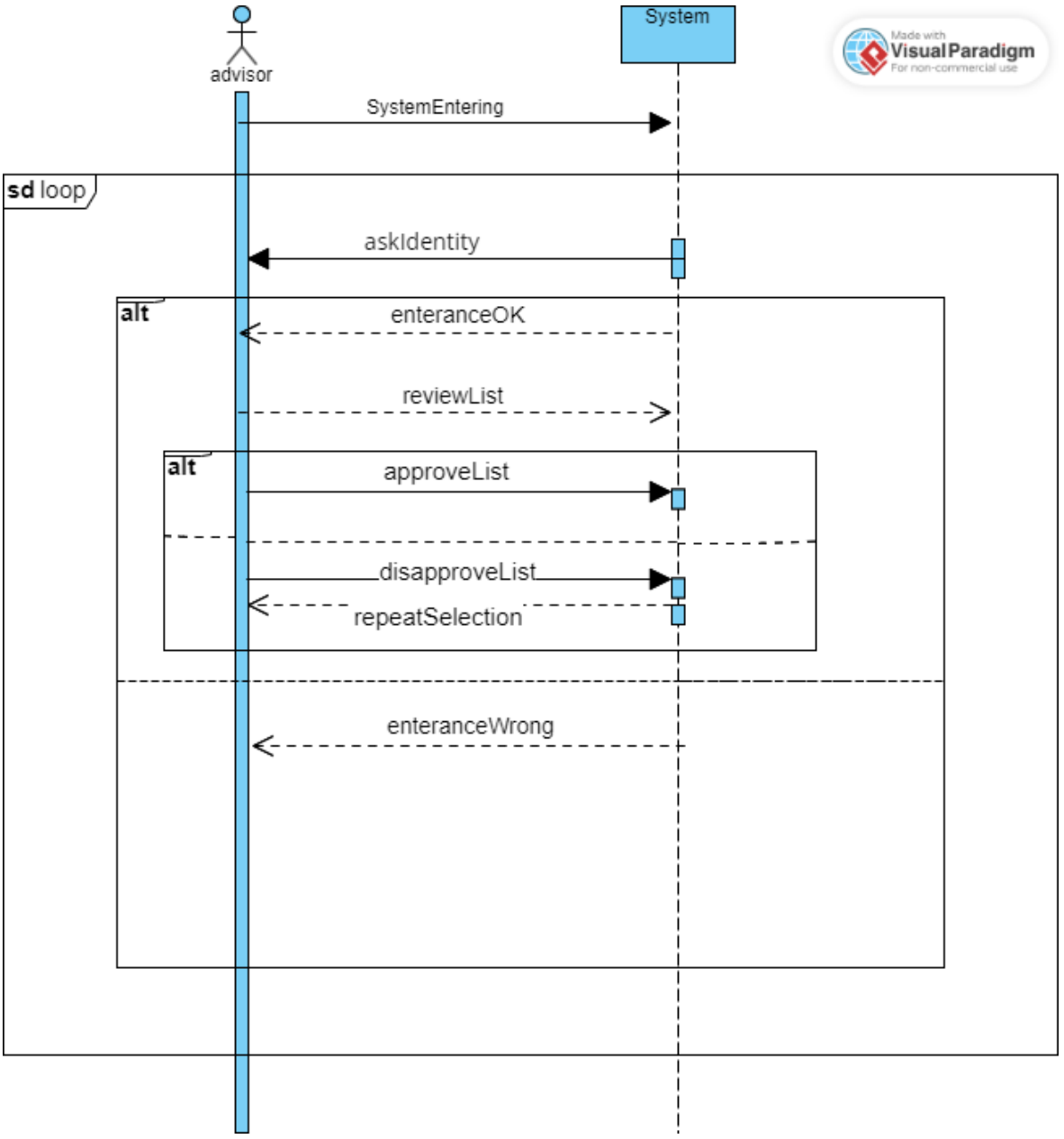
vi. If rejected, the system notifies the student of the rejection, and the student must repeat the course selection process.

SYSTEM SEQUENCE DIAGRAM

Student use case:



Advisor use case:



DOMAIN MODEL