CSE-3063

OBJECT-ORIENTED SOFTWARE DESIGN

ITERATION - 3

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.
- **Authentication**: Authentication is the process of verifying the identity of a user

- **Unit Test:** A test that evaluates the individual components.

USE CASES

Use Case: Course Registration System

Actor:

User (Student/Advisor)

Basic Flow:

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.

Use Case: Advisor Login and Course Approval

- Actor: Advisor
- Description: An advisor logs into the course registration system to view and approve course enrollments for their assigned students.
- Basic Flow:
    - Advisor enters their email address and password to log in.
    - The system authenticates the advisor's authantication.
    - After successful login, the system displays a list of the advisor's assigned students.
    - Advisor selects a specific student as their student ID to view their course enrollment.
    - The system displays the student's pending course enrollments.
    - Advisor reviews pending course enrollments and decides whether to approve or reject each course by selecting yes/no.
- Alternate Flow:
    - If the advisor enters incorrect login credentials, the system displays an error message, and the advisor is prompted to re-enter their credentials.
    - If the advisor chooses to exit, the use case ends.

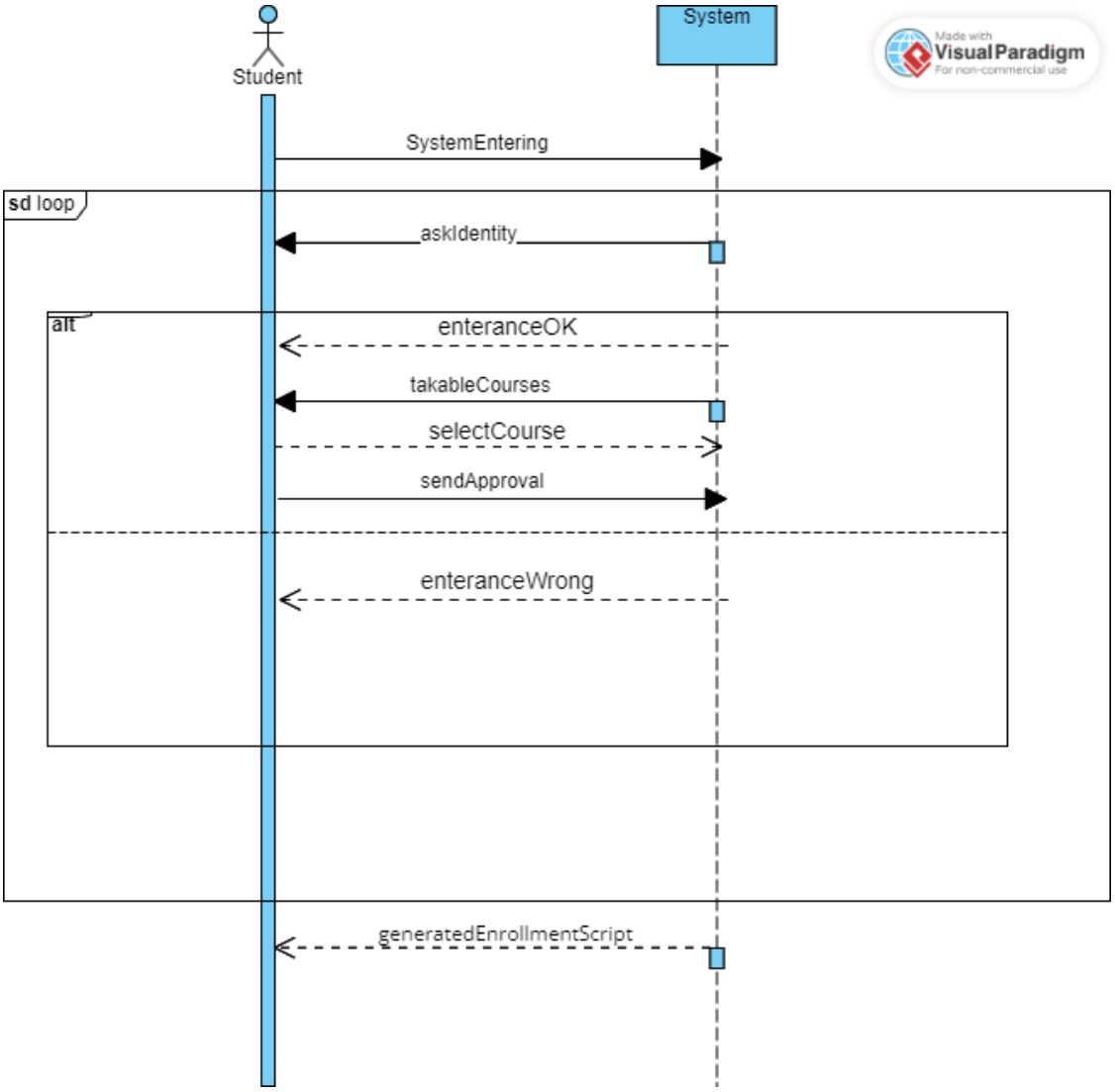
Use Case: Student Login and Course Enrollment

- Actor: Student
- Description: A student logs into the course registration system to enroll in courses for the upcoming semester.

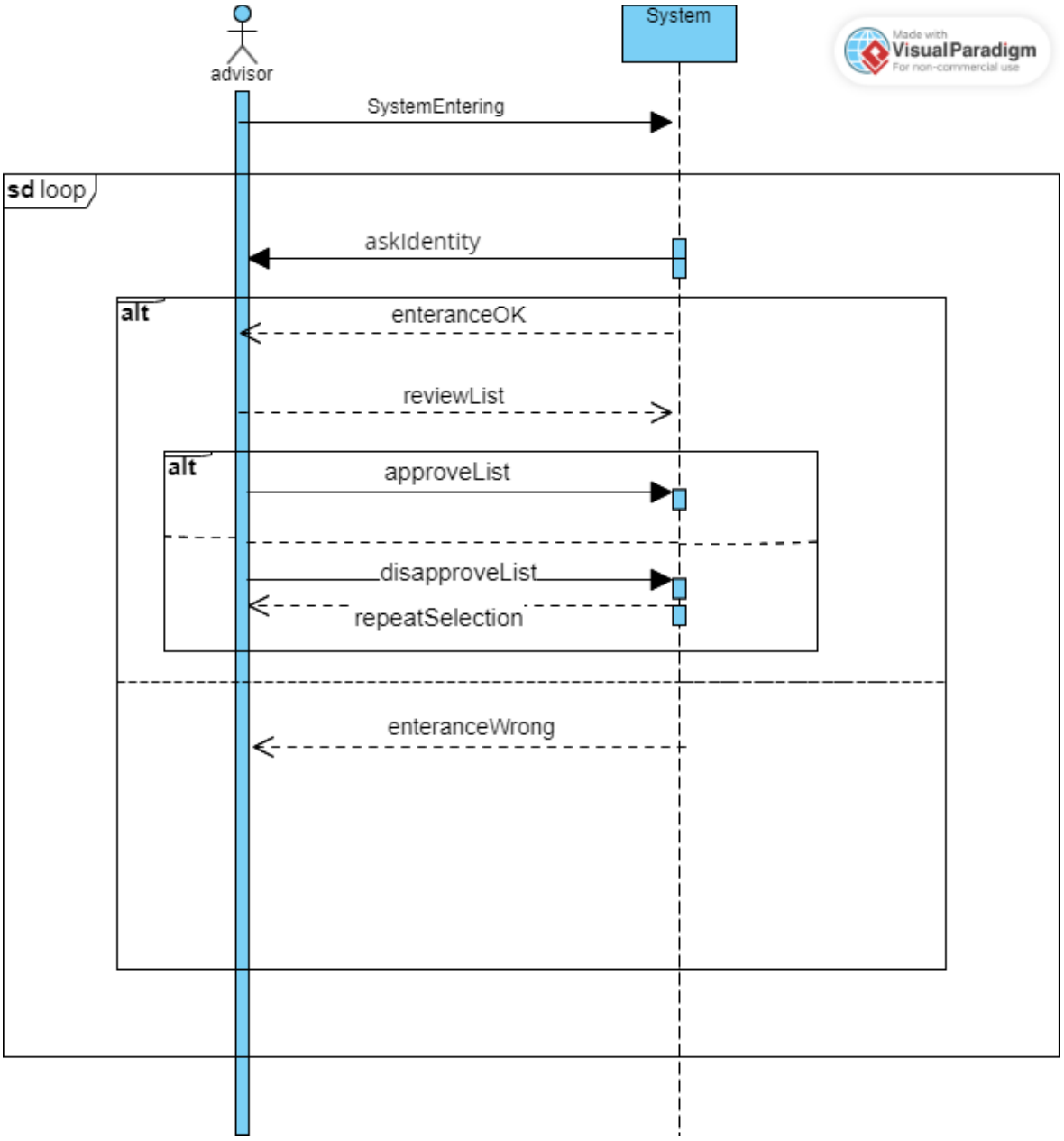
- Basic Flow:
  - Student enters their email address and password to log in.
  - The system authenticates the student's personal informations.
  - After successful login, the system displays a list of available courses for the upcoming semester.
  - Student views the list of courses and their details, including course name and course code.
  - Student selects the courses they wish to enroll in by the given number.
  - The system updates the student's course enrollment status and transcript.
  - Student receives a confirmation of successful course enrollment.
- Alternate Flow:
  - If the student enters incorrect login credentials, the system displays an error message, and the student is prompted to re-enter their credentials.

SYSTEM SEQUENCE DIAGRAM

Student:



Advisor:


advisor
System
SystemEntering
sd loop
askIdentity
alt
enteranceOK
reviewList
alt
approveList
disapproveList
repeatSelection
enteranceWrong
Made with VisualParadigm
For non-commercial use

DOMAIN MODEL