T.C.

MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER SCIENCE ENGINEERING DEPARTMENT

Object Oriented Software Design CSE3063 – Fall 2024

Title of the project: Course Registration System Iteration 1

Group Number: 17

EREN AHMET YEŞİLTAŞ - 150121017

İLKER İSLAM YETİMOĞLU - 150123823

ENES BERK YETİM - 150122507

MELİSA ARAÇ - 150122011

ATA SELÇUK - 150122075

İBRAHİM TINAS - 150719046

Project Description

The Course Registration System is a software project designed to simplify and streamline the process of enrolling students in academic courses at an educational institution. The first version of this system focuses on building the foundational structure and basic functionality needed to handle course registration.

The system will allow students and advisors to log in and interact with the platform. Students will be able to view available courses, check if they meet course prerequisites, and register for up to five courses. The system will also include features for advisors to assist students with course selection.

Key components of the system include:

1. **Student and Advisor Login**: Login functionality for both students and advisors to access the system.
2. **Core System Elements**: Creation of basic classes such as Student, Course, CourseSection, Advisor, and the CourseRegistrationSystem to manage the registration process.
3. **Course Registration with Prerequisite Checking**: Students can register for courses while ensuring they meet any prerequisites before enrolling.
4. **Data Management**: Information on students, courses, and registration will be stored in files for easy access and updating.
5. **User Interface**: A simple, command-line interface for users to interact with the system.
6. **Initial Data Setup**: The system will be populated with 10 third-year students, 2 advisors, and 10 courses, including prerequisites for certain courses.

This version of the system is designed as a basic prototype, focusing on implementing essential features that will be expanded in later versions. The goal is to create a functional and user-friendly course registration system that can be easily scaled and improved upon in future iterations.

Glossary

Advisor

A faculty member responsible for guiding students in course selection and ensuring they meet academic requirements.

Authentication

The process of verifying the identity of a user by validating their credentials (username and password).

Capacity Check

A system feature that ensures the maximum number of students allowed in a course is not exceeded during registration.

Command-Line Interface (CLI)

A text-based user interface that allows users to interact with the system by entering commands.

Course

An academic unit of study that includes a title, description, prerequisites, credit hours, and schedule.

Course Registration System

A software system designed to facilitate the process of enrolling students in academic courses.

CourseSection

A specific offering of a course, potentially including details such as instructor, time, and location.

Data Integrity

The accuracy and consistency of stored data throughout its lifecycle, ensuring no data corruption or loss.

Eligibility Check

A system function that determines whether a student meets academic level, program requirements, or credit limits for a specific course.

Error Handling

Mechanisms in the system that detect, manage, and provide feedback on errors during user operations, such as login failures or registration issues.

JSON (JavaScript Object Notation)

A lightweight data-interchange format used for storing and retrieving data in the system.

Login

The process of gaining access to the system by entering valid credentials (username and password).

Modularity

A design principle where the system is divided into distinct components or modules that can be independently developed, maintained, and updated.

Prerequisite

A course or set of courses a student must complete before enrolling in a specific course.

Prerequisite Verification

A feature that checks whether a student has completed the necessary prerequisites before allowing them to register for a course.

Registration Confirmation

A notification provided to the student upon successful registration for a course, confirming their enrollment.

Role-Based Access Control (RBAC)

A security mechanism that restricts system access based on the user's role, ensuring users can only perform actions appropriate to their permissions.

Scalability

The system's ability to handle increased load or number of users without compromising performance.

Session

A period during which a user interacts with the system, typically beginning with login and ending with logout.

Student

A user of the system who can view available courses, check prerequisites, and register for courses.

System Response Time

The time taken by the system to process a user request and provide feedback.

Usability

The degree to which the system is easy to use and provides a satisfying user experience.

User

Any individual interacting with the system, including students and advisors.

User Feedback

Messages or prompts provided by the system to inform users about the status of their actions or errors.

Use Case

A specific scenario describing how a user interacts with the system to achieve a particular goal.

Validation

The process of checking that user inputs meet the required criteria before they are processed by the system.

Functional Requirements

Functional Requirements for Login Functionality

1. **User Login**
    - The system must allow users to submit their username and password.
2. **Input Validation**
    - The system must check if both the username and password fields are filled before processing the login request.
    - If either field is empty, the system must prompt the user to fill in all required fields.
3. **Authentication**
    - The system must retrieve the user information from JSON to validate the credentials.
    - The system must securely verify if the entered password matches the stored password for the username provided.
    - Password verification should be done using a secure method, such as comparing hashed passwords rather than plain text.
4. **Error Handling**
    - If the username does not exist in the database, the system should return an error message such as "Username not found."
    - If the password is incorrect, the system should return an error message such as "Invalid password. Please try again."
5. **Logout**
    - The system should provide a "Logout" button that allows users to manually end their session at any time.
    - Upon logout, the system must remove or invalidate the session data and redirect the user to the login page or homepage.

Functional Requirements for Basic Course Registration with Prerequisite Checking

1. **Course Registration Interface**

    - The system must provide a course registration interface where students can view available courses and select the ones they wish to enroll in.
    - The interface must display essential course information, including course title, description, prerequisites, credit hours, and schedule.

2. **Prerequisite Verification**
    - The system must verify if the student has completed the prerequisite courses before allowing them to register.
    - If a student has not met the prerequisites for a course, the system must display a message indicating the missing prerequisites and prevent registration.

3. **Eligibility and Capacity Check**
    - The system must check if the student is eligible to register for the selected course based on academic level, program requirements, or credit limits.
    - The system must check if the course has reached maximum enrollment capacity and notify the student if the course is full.

4. **Registration Confirmation**
    - Upon successful registration, the system must provide a confirmation message and update the student's course schedule.
    - The system must update course enrollment data to reflect the new registration.

5. **Error Handling**
    - If any errors occur during registration, the system must display an error message and prompt the student to try again.

Non-Functional Requirements

1. Security Requirements

- **Authentication and Authorization:**
    - The system must implement secure authentication for all users (students and advisors).

2. Performance Requirements

- **System Response Time:**
    - The system must respond to 95% of user actions (e.g., login, course registration) within 3 seconds under normal load.

3. Usability Requirements

- **Ease of Use:**
    - The command-line interface (CLI) must be intuitive, with clear prompts and error messages guiding users through each process.
- **User Feedback:**
    - The system must provide immediate feedback for user actions, such as successful login, registration errors, or course availability.

4. Reliability Requirements

- **Availability:**
    - The system must be operational during the life of the program.
- **Data Integrity:**
    - The system must ensure that all data (e.g., student records, course registrations) is accurately saved and retrieved without corruption.

5. Maintainability Requirements

- **Modularity:**
    - The system must be designed with modular components to facilitate easy updates and maintenance.
- **Error Logging:**
    - The system must log all errors and exceptions for debugging and analysis, with logs stored securely.

6. Portability Requirements

- **Platform Independence:**
    - The system must be compatible with major operating systems (Windows, macOS, Linux) for development and deployment.
- **Deployment Flexibility:**
    - The system should be easily deployable on local machines or cloud environments for future scalability.

7. Data Management Requirements

- **Storage Format:**
    - The system must store data in JSON format for easy parsing and updating.

Domain Model

Domain model also can be viewed at DomainModel.pdf.

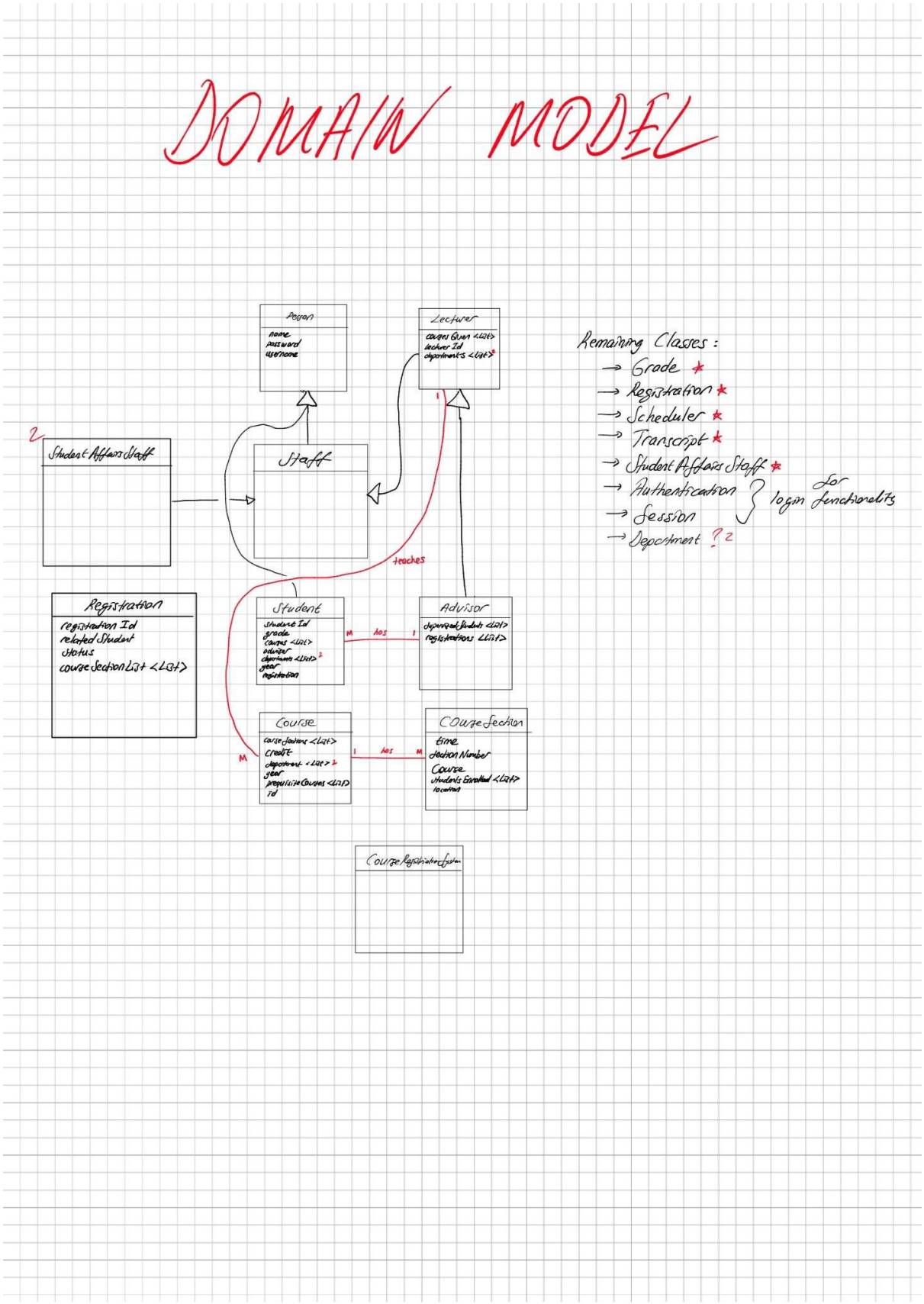


Figure 1: Domain Model

Use Cases

Use Case: Login

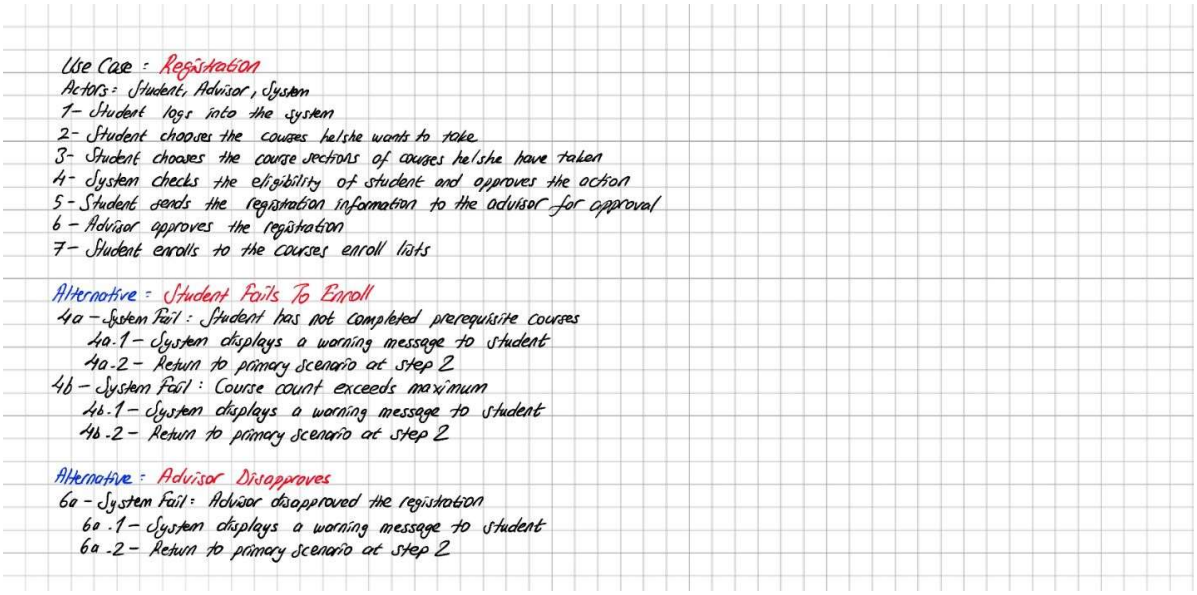
This use case describes how a user logs into the Course Registration System.

Actors: User, System

1. User selects the login option.
2. System displays a login form and asks for username and password.
3. User enters username and password.
4. System validates the user by checking username and password.
5. System displays the homepage of the user.
6. Use case ends successfully.

Alternative: Login Failure

4.a System fails to validate the user, displays an error message and allows re-entry of username and password.



System Sequence Diagrams (SSDs)

System Sequence Diagram of Login Functionality

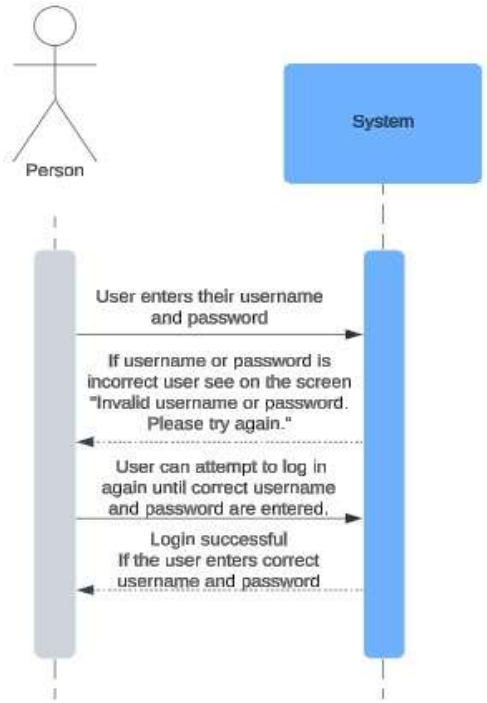


Figure 2: System Sequence Diagram of Login Functionality

System Sequence Diagram of Registration System