Requirement Analysis

CSE3063.1 – Object-Oriented Software Design

Group 16

Iteration 1

1. Description

This is a system helps university with registration process by using defined rules and restrictions. System allows students to register their desired courses by considering easily. A software that can manage day to day operations for the university. Student registration system is a structure that provides a simple set-up of programs for student enrollment. It is a method that enables the university to have better control over growing number of students, radically reduces the work and costs by getting rid of paper.

2. Requirements

- **Functional Requirements**
  - Student Information
    1. Semester Details
    2. Passed Courses
    3. Failed Courses
    4. Advisor
    5. GPA
    6. Completed Credits

  - Course Information
    1. Course Credits
    2. Credits
    3. Quota
    4. Pre & Co-requisite Courses
    5. Availability in semesters
    6. Lecture Hours
    7. Lecture Room
    8. Section
    9. Professor

- Current courses for all students

- Course enrollment
    1. Check if the student is allowed to take the selected courses
        a) If a selected course is from a higher semester, the student should have a GPA above 3.
        b) A student can't select more than 10 courses.
        c) Advisor will approve the selected courses if the student can take the courses, it will be added to the student's schedule.
        d) Advisor won't approve the selected courses if the student can't take them and inform him/her about it.

- **Non-Functional Requirements**
    - Easy to handle system
    - Program is implemented with JAVA
    - Logo of the university should be displayed
    - Provide necessary information when an error is committed
    - Registration process should be logged

3. Iteration Plan

1) *Planning and Requirements*

Determine the initial requirements, gather the connected documents, and build a timeline for the primary iterative cycle.

2) *Analysis and Design*

Finalize the business needs, database models, and technical requirements based on the plan. Produce a design or algorithm that satisfies your requirements.

3*) Implementation*

Develop the functionality and design required to meet the specifications.

4) *Testing*

Identify and locate what's not working or performing to expectations. Fix sections that do not work as expected.

5) *Evaluation and Review*

Compare this iteration with the requirements and expectations.

4. Use Case: Course Registration

Scope: University Course Registration System

Level: User-goal

Primary Actor**:** Student

Stakeholders/Interests:

- Student: Wants easy and quick way to register to courses as well as delete courses or modify existing schedule.
- Advisor/Registrar: Wants ability to view student /course information and approve/delete/modify courses picked by students easily.
- Department: Wants the ability to view and receive information about all events taking place in the system.
- Billing System: Wants to receive information about payments of tuition fees and store information about payment details.

Preconditions:

1. Student must be logged in and authenticated by student ID.

2. Student must have paid the tuition fee (if there's any) for the current semester before being able to register for courses.

3. Student must be assigned an advisor to overlook/supervise registration process.

***Success Guarantee*:** Courses taken are saved for each student and added to transcript, Advisor informed, Schedule created for the semester, Current quota for each course updated, and list of students taking specific course updated for the lecturer.

Main Success Scenario:

Basic Flow:

1. Student clicks on course registration section in the student information system.
2. System displays student's blank courses list.
3. System retrieves and displays courses available for the semester.
4. Student selects at most 10 courses as well as a section for each lab course (if there's any and if there are choices of different sections).
5. Student sends selected courses to advisor for approval.
6. System sends courses to advisor, pending approval.
7. Courses are approved, added to the transcript and system displays and saves created schedule.

Alternative Flows/Events:

1a. Course registration closed

1. System displays message to student and returns to main page.

1b. System fails to load

1. System returns to main menu of Student Information System prompting the user/student to try again later.

4a. Student chooses more than 10 courses

1. System displays error message and prompts student to delete subjects till allowed maximum amount is reached.

4b. Student selects courses he/she has failed or has not taken from the previous semester (if there's any).

1. Advisor checks if the course hours clash with the any of the current semester's courses.
2. If hours clash, Advisor removes courses and informs student of any alternatives he/she can make.

4c. Student selects NTE, FTE and TE courses.

1. Advisor checks if the course hours clash with any of the current semester's courses.
2. If hours clash, Advisor removes courses and informs student of any alternatives he/she can make

4d. Student deletes a course

1. If course not sent to advisor's approval yet, message is displayed asking student permission to delete
2. Course is deleted and removed from list of courses
3. If course sent to advisor's approval, request for deletion is sent to the advisor pending his/her approval.
4. If approval granted, course is deleted from list of chosen courses.

4e. Student has not taken prerequisite course for course(s) he/she selected.

1. Error message displays that the course cannot be taken due to missing prerequisite.
2. System prevents addition of course**.**

4f. Student selects a course that needs to have a specific number of credits completed.

1. Advisor checks whether the required number of credits has already been taken by the student.
2. If credit number/hours not met, advisor rejects selected course.
3. Message displays to the student informing him/her.

4g. Student selects a Technical Elective course.

1. Whether or not student has already taken two Technical Elective courses in the fall semester is checked.
2. If student has indeed taken only two, advisor rejects course.

4h. Student selects a Faculty Technical Elective course.

1. Current status of student is checked (graduating/not graduating).
2. If requirements are not met, advisor rejects course.

4i. Course quota full

1. System displays message stating quota is full for course
2. System rejects addition of the course.

4j. Lab section quota full

1. System displays message stating quota is full for particular section and prompts the student to pick another section.
2. System returns to list of courses page.

System repeats these steps until a section with non-complete quota is chosen

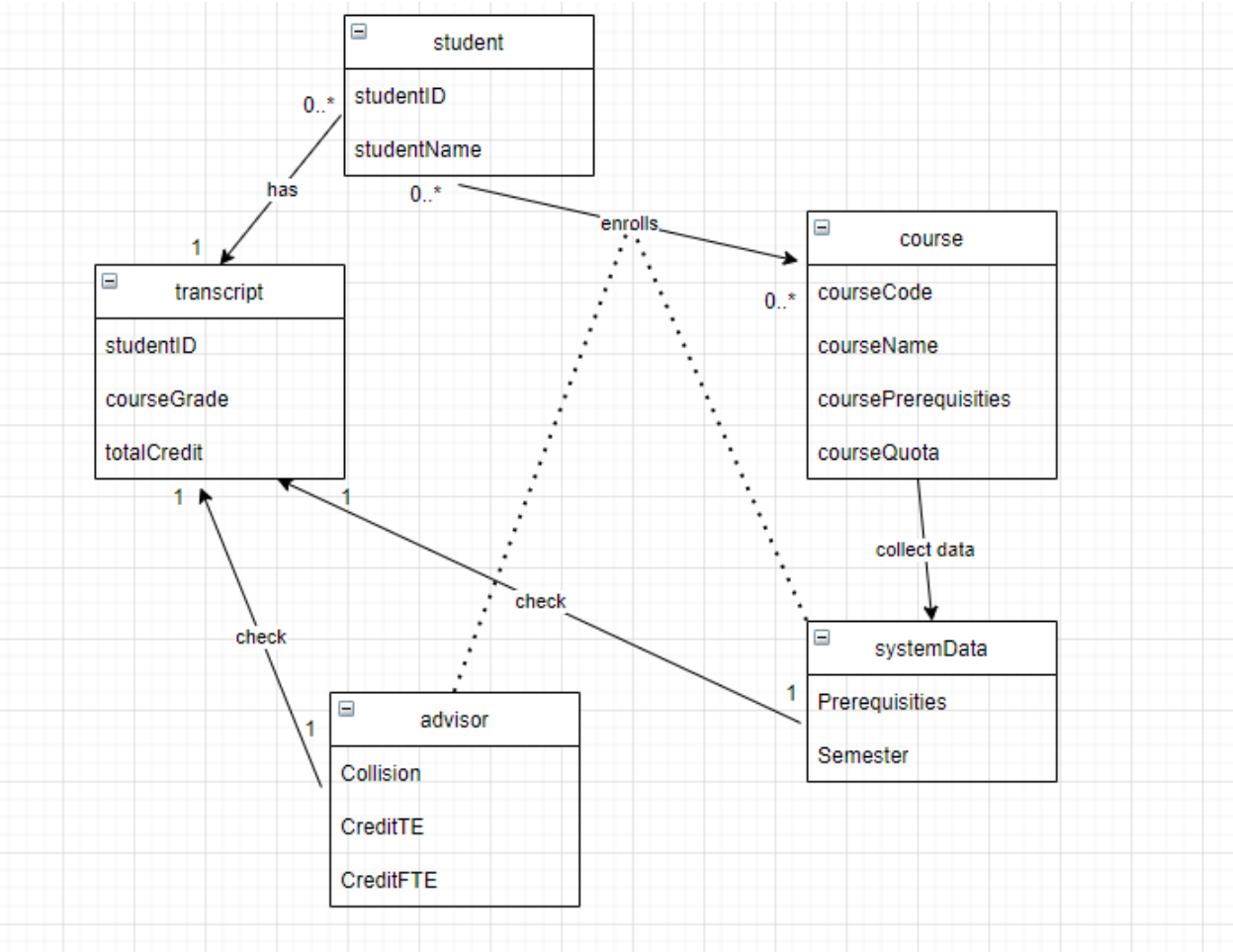
4k. Student selects a summer course.

1. System checks if payment has been made for the selected subject (if eligible).
2. If payment is not made, System displays error and prompts student to pay for the subject first and course is not added to student's transcript.
3. If payment made, course is approved.

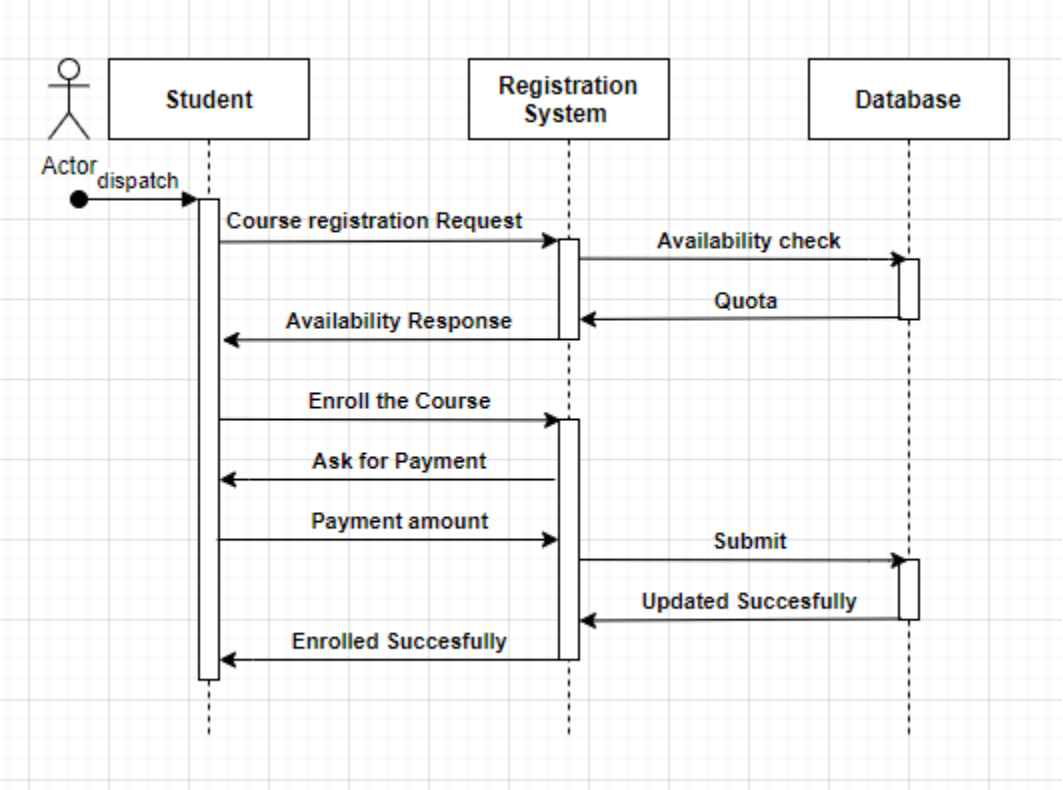
Special Requirements:

1. System must be multilingual providing both Turkish and English for foreign students.
2. UI to provide readability and ease of use for the user.

5. Domain Model



6. SSD



7. Glossary

Course -A class offered by the university.

GPA - The evaluation of a particular student for all the courses he they completed.

Professor- A person teaching classes at the university.

Schedule -A person enrolled in classes at the university.

Roster - All the students enrolled in a particular course offering.