Description

This software simply simulates a course registration environment in a university. There are two kinds of actors in this system: student and advisor. A student may register to a list of courses with certain regulations. Advisor's duty is to approve or disapprove the course list of the student he/she is advising to.

There is no GUI, so communication with the user of program is done through input/output files and command prompt. This software works in a randomized fashion, so every time it is executed different results are outputted to the user.

Requirements

Functional Requirements

- When an instructor receives a list of courses to approval, he/she can approve or disapprove it.
- The software will output the randomized results in related json files and command prompt.
- The software will take the list of courses, types of courses, prerequisites, and semester (fall/spring) from a json file.
- While adding a course to a student, system must ensure that all prerequisites of that course is satisfied.
- While adding a course to a student, system will check that sum of credits does not exceed a certain limit.

Non-functional Requirements

- Python will be used to develop the program.
- No database system will be used for input or output operations.
- Json file will be used to store input and output.
- There will be no GUI.
- Artifacts will be shared by using GitHub.
- Drawio and Paint is used to draw and prepare the analysis and design documents.

Glossary

- Student: Someone who is studying at a university.
- Course: A series of lessons about a particular subject.
- Elective Course: Non-mandatory course.
- Compulsory Course: Mandatory course.
- Advisor: someone whose job is to give advice about a course registration.
- Transcript: A record of courses and their grades.
- Input: Any action user can do to software or computer.
- Output: Software's reaction to the user action.
- GUI: Stands for Graphical User Interface. It is a way to communicate with user employing icons, dropdown menus, windows and more.
- Command Prompt: It is a text-based program to the provide communication between user and the computer.
- Json: Stands for Open Standard File Format, it is a standard to exchange data between different computer software.
- Functional Requirement: Describes what a software does with different inputs.
- Non-functional Requirement: Describes how a software achieves a feature in a more technical view.
- Python: A multi paradigm programming language.

Use Cases

Student applies for registering the courses successfully

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Main Success Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Sum of credits does not exceed limit, student had passed prerequisite courses, has not taken any FTE in fall, eligible to take engineering project, has not taken more than 2 TE in fall.
5- System approves registration

Student applies for registering the courses with sum of credits exceeds the limit

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Sum of credits exceeds the limit
5- System disapproves registration
6- Student goes back to step 2

Student applies for registering the courses with unsatisfied prerequisites

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Student had not passed prerequisite courses
5- System disapproves registration
6- Student goes back to step 2

Student applies for registering the courses with not eligible to register FTE in FALL semester

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Student took FTE in FALL semester and he/she is not graduating this semester
5- System disapproves registration
6- Student goes back to step 2

Student applies for registering the courses with not being eligible to take engineering project

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Student is not eligible to take engineering project
5- System disapproves registration
6- Student goes back to step 2

Student applies for registering the courses with having more than two technical electives in fall

Scope: Course Registration Simulation Application

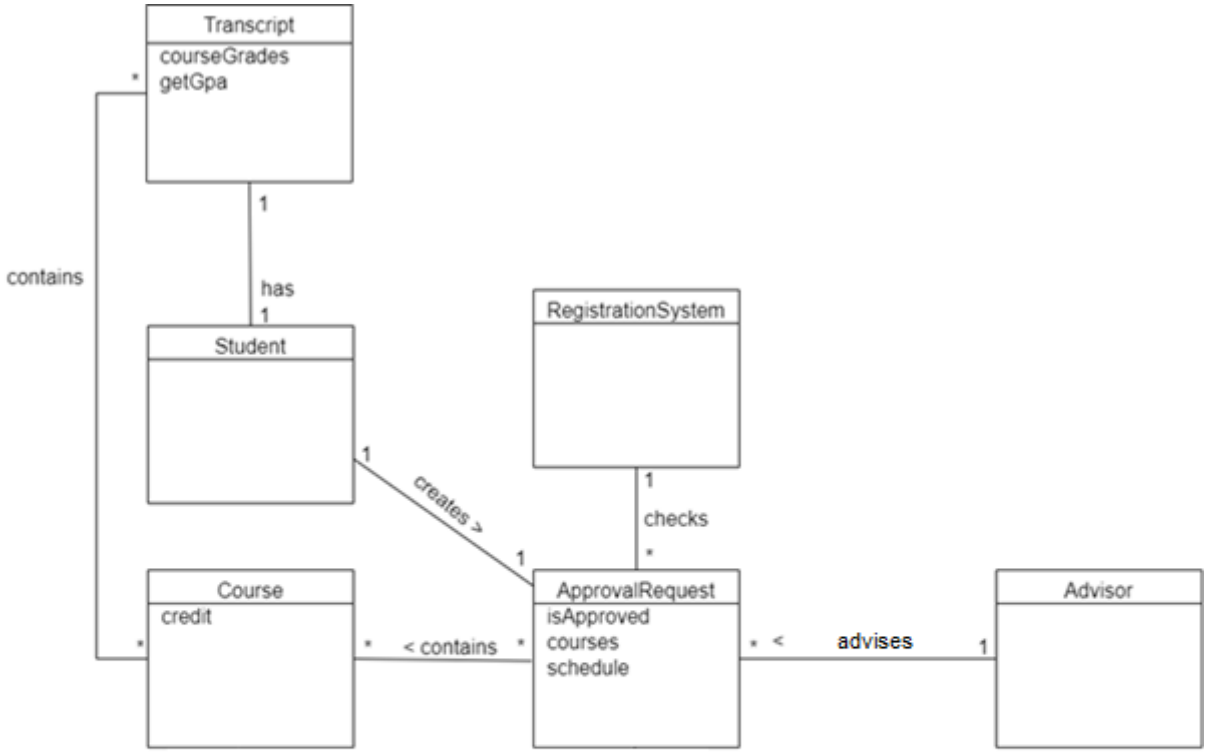
Primary Actor: Student

Level: Student Goal

Failure Scenario:

1- Student wants to enroll in a course that she/he can enroll in during her/his term
2- The system presents the courses for the student
3- Student selects courses
4- Student selected more than two technical electives in fall
5- System disapproves registration
6- Student goes back to step 2

Domain Model



System Sequence Diagram

Main Success Scenario

1. Student wants to enroll in a course that she/he can enroll in during her/his term
2. The system presents the courses for the student
3. Student selects courses
4. There are no collisions in schedule of student, sum of credits does not exceed limit,student had passed prerequisite courses
5. Advisor approves registration