

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 a student wants to add a course to his/her schedule, it will be sent to advisor by registration system.
- 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, quota, and semester (fall/spring) from a json file.
- When a student wants to add a course to schedule, registration system will ensure there are no collisions in Student's schedule.
- 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

- Object Oriented Programming language Java 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 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.
- Technical Elective Course: Technical electives are courses (generally in computer science, mathematics, engineering, or business) oriented toward the design or use of computers.
- Faculty Technical Elective: At most 2 technical elective courses can have non ISE codes and these courses can be selected from the Faculty Technical Elective Course Pool with the approval of the student's advisor.
- University Elective: Electives are courses you can choose outside core or compulsory courses in your degree. They can be chosen from all courses available at the University that do not have any other conditions (such as a course requisite) applied to them.
- Compulsory Course: Mandatory course.
- Advisor: someone whose job is to give advice about a course registration.
- Transcript: A record of courses and their grades.
- Schedule: A list of times of courses.
- 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 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.

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 indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- There are no collisions in schedule of student, sum of credits does not exceeds limit, student had passed prerequisite courses.
- 5- Advisor approves registration
- 6- Student successfully enrolls the semester.

Student applies for registering the courses with collision in the schedule

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

- 1- Student indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- There are more than one collision in schedule of the student
- 5- System disapproves registration
- 6- Student goes back to step 2

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 indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- Sum of credits of selected courses exceeds the limit
- 5- System disapproves registration
- 6- Student goes back to step 2

Student applies for registering the courses with unsatisfied prerequisite

Scope: Course Registration Simulation Application

Primary Actor: Student

Level: Student Goal

Failure Scenario:

- 1- Student indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- Student had not passed prerequisite courses for some courses that he/she selects for this term
- 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 indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- Student took FTE in FALL semester and he/she is not graduating this semester
- 5- Advisor disapproves registration
- 6- Student goes back to step 2

Advisor receives students registration

Scope: Course Registration Simulation Application

Primary Actor: Advisor

Level: Advisor Goal

Failure Scenario:

- 1- Student indicates that he/she 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- He/she selects the courses according to his/her wish
- 4- Student selects a FTE course in FALL semester and he/she is not graduating this semester
- 5- Advisor checks and finds this situation and disapproves registration
- 6- Student goes back to step 2

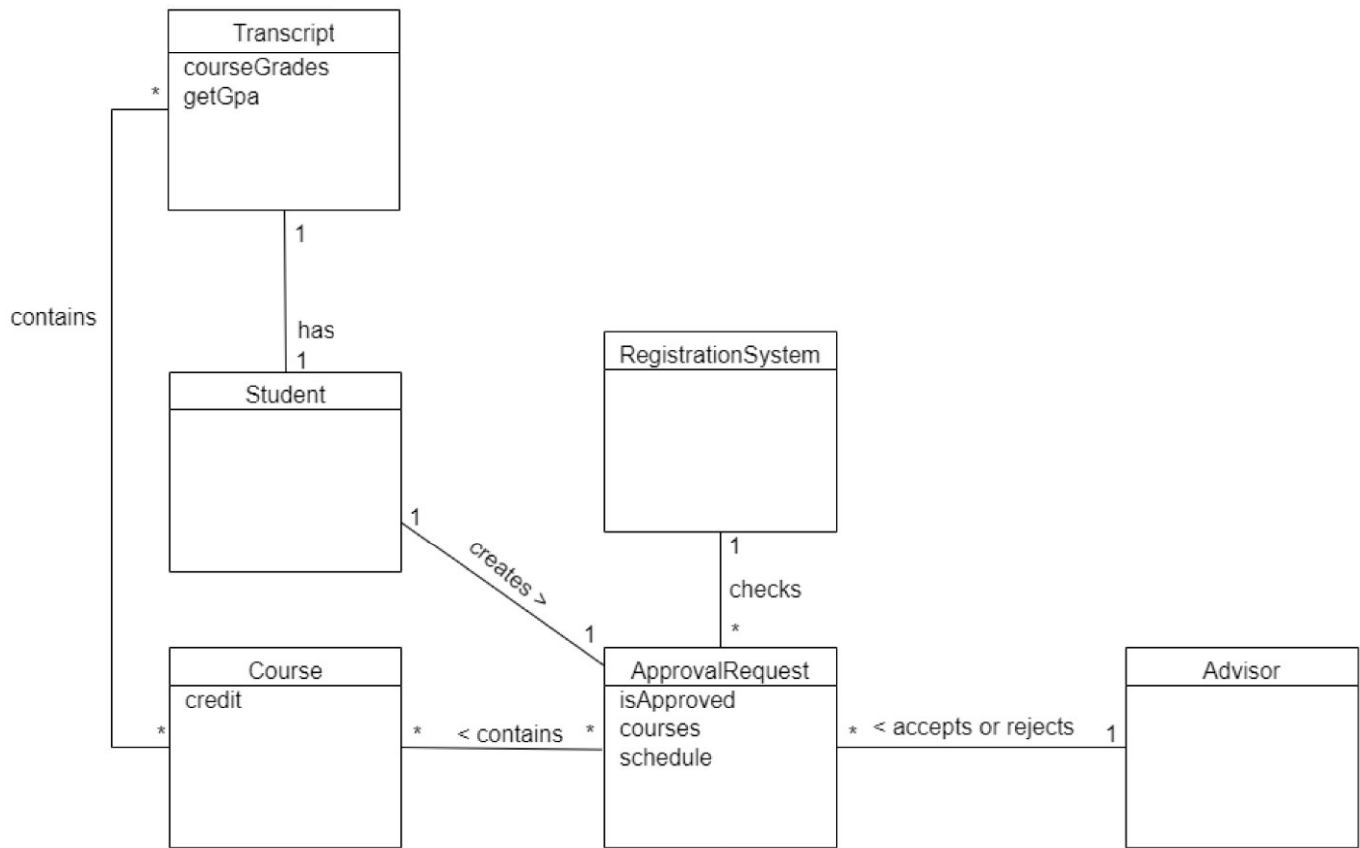
Iteration Plan

The iteration plan, also known as the spring plan, is the detailed execution plan for a given (usually current) iteration. It defines the iteration goals and commitments by specifying the user stories, work tasks, priorities and team member work assignments required to complete the iteration. The iteration plan is normally produced by the entire development unit during the iteration planning session.

The Iteration Plan defines the work for the current and next iteration not all iterations. It includes risk management. For our project the biggest risk is not to complete a functional working software for a registration system. In order to eliminate the biggest risk we use iterative development process. We divide the problem into small subsets and each iteration we will try to complete only one subset.

For the first iteration we will focus on the documentation especially analysis phase. In order not to miss any particular specialty and function ability we will try to keep the design and software phase. In 2 or 3 weeks we plan to finish the analysis part completely and also a small running software which randomizes student and his/her json files which includes prior semesters transcript and current semesters courses which he/she try to enroll . Also this software will be able to give some feedback about some failures on enrolling courses and make a schedule for the current semester.

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

