REQUIREMENT ANALYSIS DOCUMENT

Introduction

In this document, purpose of the first iteration of the project, list of functional requirements and overall description is provided. Additionally, system functionality using domain model and SSD (System Sequence Diagram) are provided.

Purpose

The purpose of the first iteration of the project is tracking registration process of the students according to department regulations as a simulation. In the simulation, user can observe how distance education period will be handled. Also, project can be used in order to test if the system works properly.

Functional Requirements

- System requires an input file to start simulation.
- System creates instructors.
- System creates courses.
- System assigns courses to instructors.
- System creates 70 students per semester.
- System provides available courses of the current semester.
- Students register courses according to regulations.
- System creates transcript files for each student.

Overall Description

In the registration process of the simulation, system will check whether student can take available courses in current semester. If there is a situation that prevents course taking, the system handles this situation as an error and adds to transcript file of student. For instance, if the capacity of elective course is full, system does not approve the registration of this course. If students can meet the conditions necessary for the course, they may register for the course. In the end of the semester the grades of courses will be generated and GPA will be created. After finishing the process, system provides transcript files of students. Transcript files includes summary of registration process. This process will continue until end of 8 semesters.
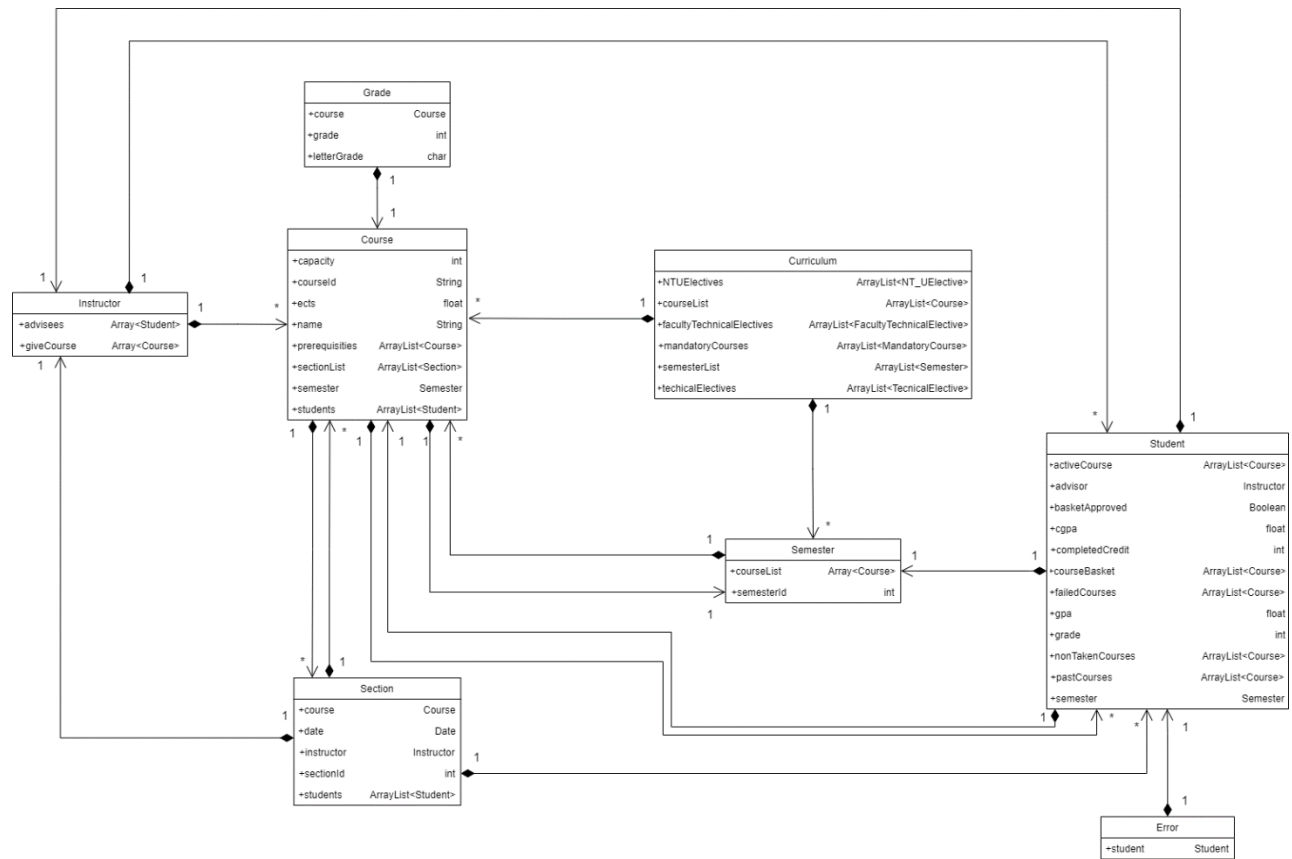
Customer:

- Murat Can Ganiz

Developers:

- Muhammed Eren Atala
- Hüseyin Kerem Mican
- Ahmet Faruk Güzel
- Hakan Kenar
- Ahsen Yağmur Kahyaoğlu
- Emre Demir

Domain Model

Grade
+course	Course
+grade	int
+letterGrade	char

Course
+capacity	int
+courseId	String
+ects	float
+name	String
+prerequisities	ArrayList<Course>
+sectionList	ArrayList<Section>
+semester	Semester
+students	ArrayList<Student>

Curriculum
+NTUElectives	ArrayList<NT_UElective>
+courseList	ArrayList<Course>
+facultyTechnicalElectives	ArrayList<FacultyTechnicalElective>
+mandatoryCourses	ArrayList<MandatoryCourse>
+semesterList	ArrayList<Semester>
+techicalElectives	ArrayList<TecnicalElective>

Instructor
+advisees	Array<Student>
+giveCourse	Array<Course>

Semester
+courseList	Array<Course>
+semesterId	int

Student
+activeCourse	ArrayList<Course>
+advisor	Instructor
+basketApproved	Boolean
+cgpa	float
+completedCredit	int
+courseBasket	ArrayList<Course>
+failedCourses	ArrayList<Course>
+gpa	float
+grade	int
+nonTakenCourses	ArrayList<Course>
+pastCourses	ArrayList<Course>
+semester	Semester

Section
+course	Course
+date	Date
+instructor	Instructor
+sectionId	int
+students	ArrayList<Student>

Error
+student	Student

System Sequence Diagram