Specification of Project Design Document

Title: Ashesi University Academic Management System

Authors

Bamah Asiamah Nyanteh (Cohort C)

Kojo Darkoh Opoku(Cohort C)

Kwamena Duker(Cohort C)

Yoni Kwasi Appiah-Berko (Cohort C)

Introduction

Our Java program is designed to streamline administrative tasks and facilitate academic operations within Ashesi University. The aim is to provide the registry and faculty with a centralised platform for managing courses, student records, and grades by automating processes such as course management, grade tracking, and student enrollment to enhance efficiency and organisation within the university.

Solution Design:

Our solution consists of several classes that interact with each other to manage different aspects of university operations.

Here's an overview of the classes and their relationships:

- Person (Superclass):

Attributes: firstName, lastName, email

Methods: getters and setters for attributes

- Student (Subclass of Person):

Represents a student enrolled at Ashesi University. Attributes include student ID, name, email, and enrolled courses. Methods allow students to enroll in courses, drop courses, and generate transcripts.

Attributes: studentID, coursesEnrolled, numCourses, maxCourses

Methods: enrollCourse(), dropCourse(), getTranscript()

- Instructor (Subclass of Person):

Represents a faculty member responsible for teaching courses. Attributes include name and email. Methods allow instructors to manage course offerings and record grades for students.

Attributes: No additional ones

Methods: teach()

Relationships:

Teaches Course objects

- Course:

Represents a course offered at the university. Attributes include course code, name, credits, instructor, and maximum enrollment capacity. Methods allow for course creation, enrollment management, and grade tracking.

Attributes: courseCode, name, credits, instructor, enrolledStudents, numEnrolled, maxStudents

Methods: enrollStudent(), dropStudent(), recordGrade(), displayEnrolledStudents()

Relationships:

Depends on the Student class for enrollment and grade recording

- Grade:

Represents a student's grade for a specific course. Attributes include the score, letter grade, and grade point. Methods allow for grade conversion and calculation.

Attributes: score, letterGrade, gradePoint

Methods: convertToLetter(), convertToGradePoint()

Relationships:

Used by the Course and Transcript classes for grade conversion and calculation

- Transcript:

Represents a student's academic transcript. Attributes include the student, enrolled courses, letter grades, grade points, and GPA. Methods allow for grade recording, GPA calculation, and transcript generation.

Attributes: student, courses, letterGrades, gradePoints, numCourses

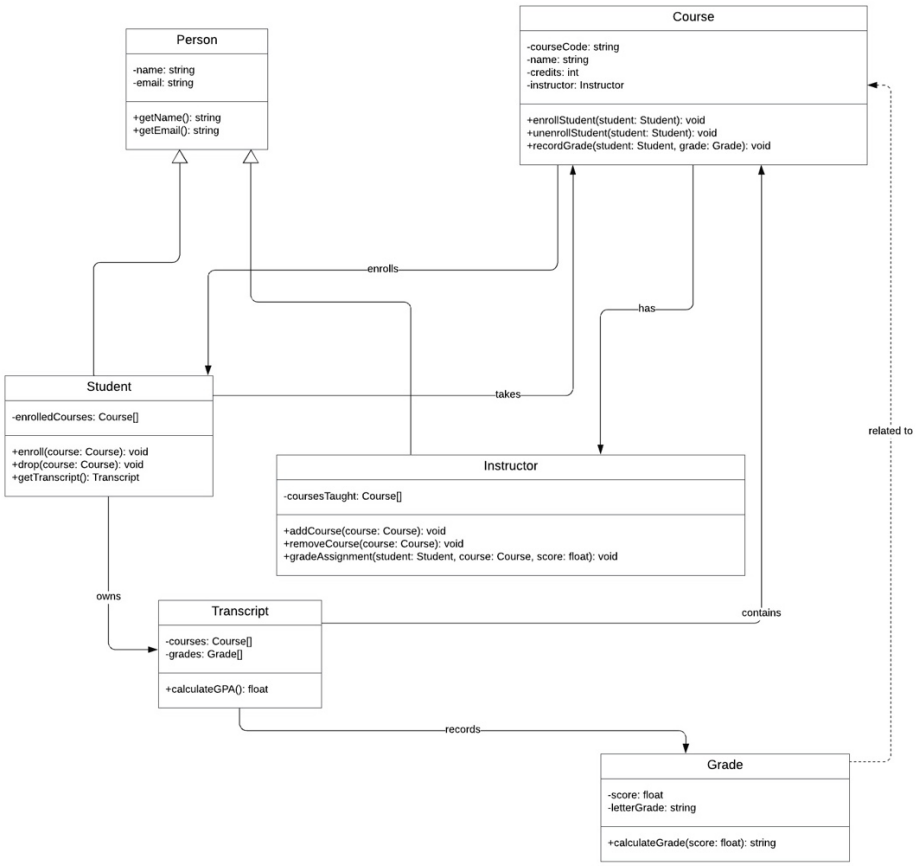
Methods: addCourseGrade(), calculateGPA(), displayTranscript()

Relationships:

Uses Grade class for grade conversion and calculation

Depends on the Student class for generating transcripts

UML Diagram



In terms of the interactions of the classes:

The Student class allows the user to enroll students in courses, drop courses, and generate transcripts.

Instructors teach Courses.

Courses enroll students, record grades, and display enrolled students.

Grades are converted and calculated for courses and transcripts.

Transcripts aggregate course information and grades for students.