College Database System
Design Document

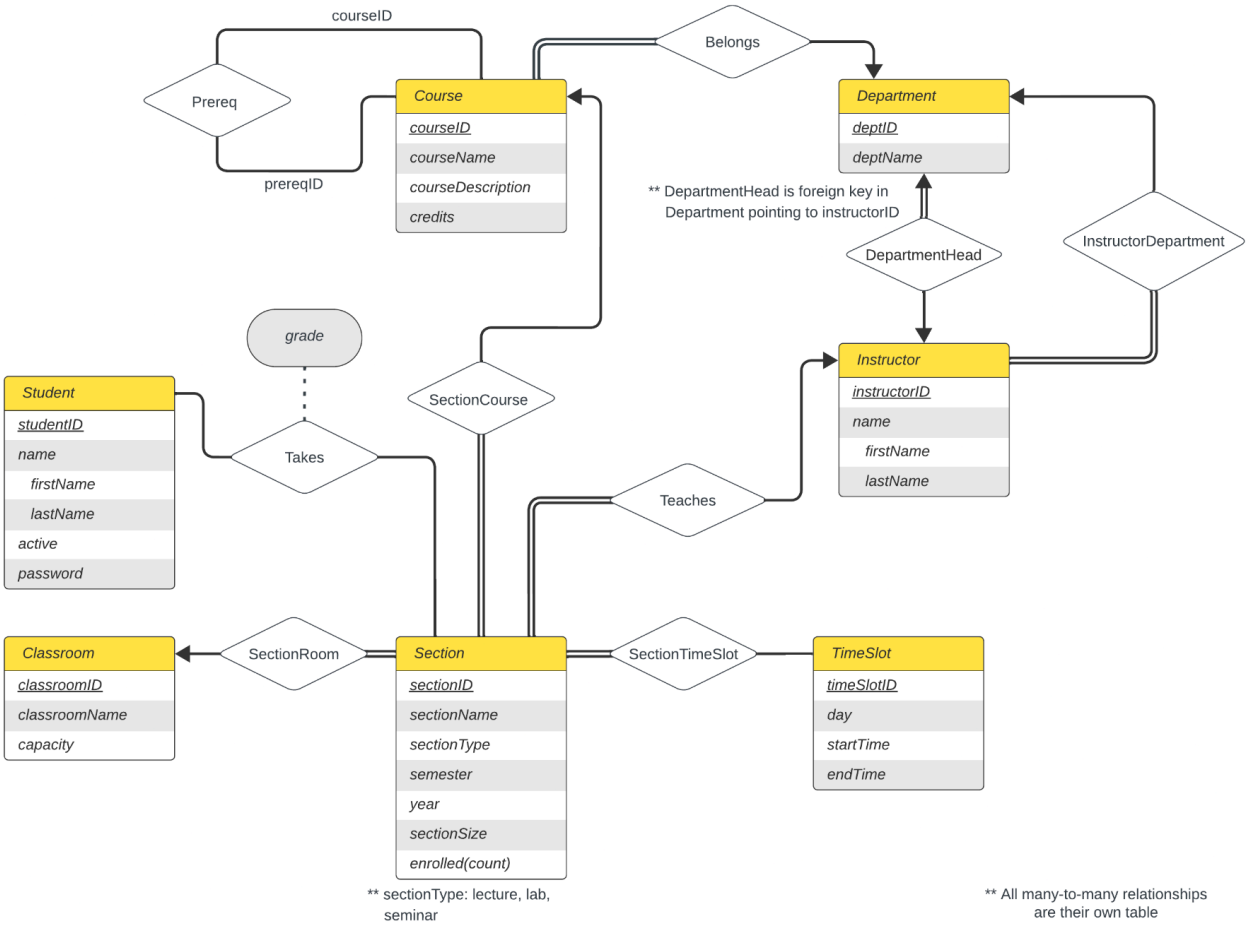
CMPT 391 - Project 1
February 14, 2024
Group 2

Nadir Bakridi
Jesse Emery
Yeji Kim
Kelsey Pascual
Michael Sin

Table of Contents

Entity-Relationship Diagram **3**
Database Tables **4**
Screen Mock-ups **5**
Business Rules of Our System **7**
Data Linking **7**
Integrity constraints **8**
Stored Procedures **8**

Entity-Relationship Diagram



Database Tables

Classroom(classroomID, classroomName, capacity)

Course(courseID, courseName, courseDescription, credits, departmentID[FK])
departmentID references Department

Prereq(prereqID[FK], courseID[FK])
prereqID references Course
courseID references Course

Department(departmentID, departmentName, departmentHeadID[FK])
departmentHeadID references Instructor

Instructor(instructorID, firstName, lastName, departmentID[FK])
departmentID references Department

Timeslot(timeslotID, day, startTime, endTime)

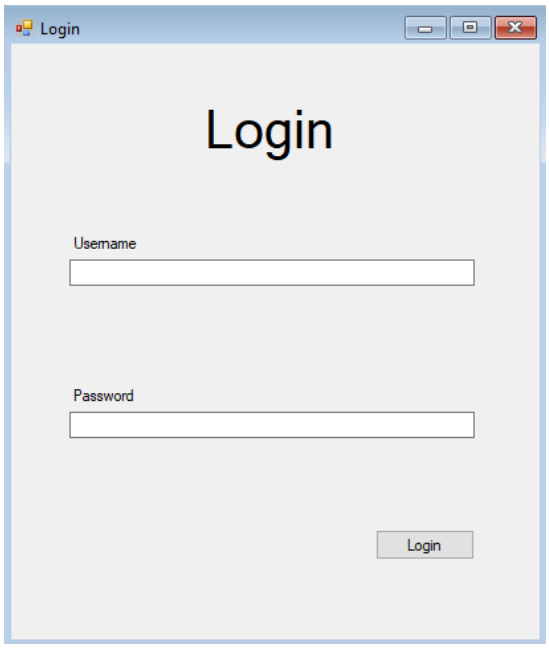
SectionTimeSlot(sectionID[FK], timeslotID[FK])
sectionID references Section
timeslotID references Timeslot

Section(sectionID, sectionName, sectionType, semester, year, sectionSize, enrolled, instructorID[FK], courseID[FK], classroomID[FK])
instructor ID references Instructor
courseID references Course
classroomID references Classroom

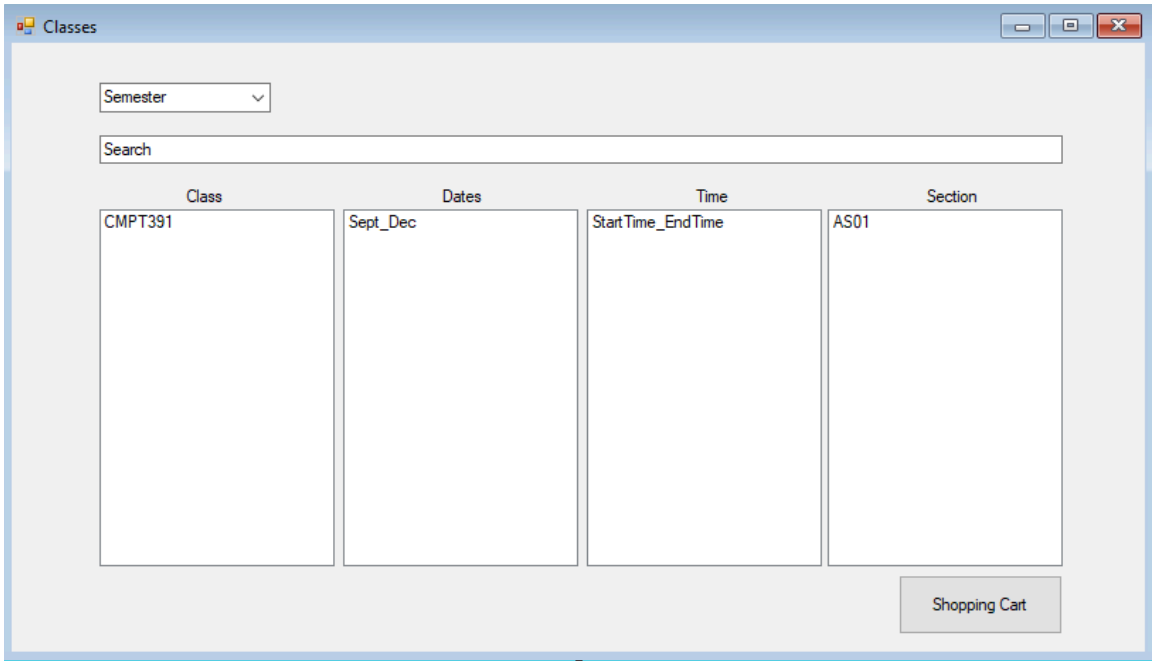
Student(studentID, firstName, lastName, activeStudent, password)

Takes(studentID[FK], sectionID[FK], grade)
studentID references Student
sectionID references Section

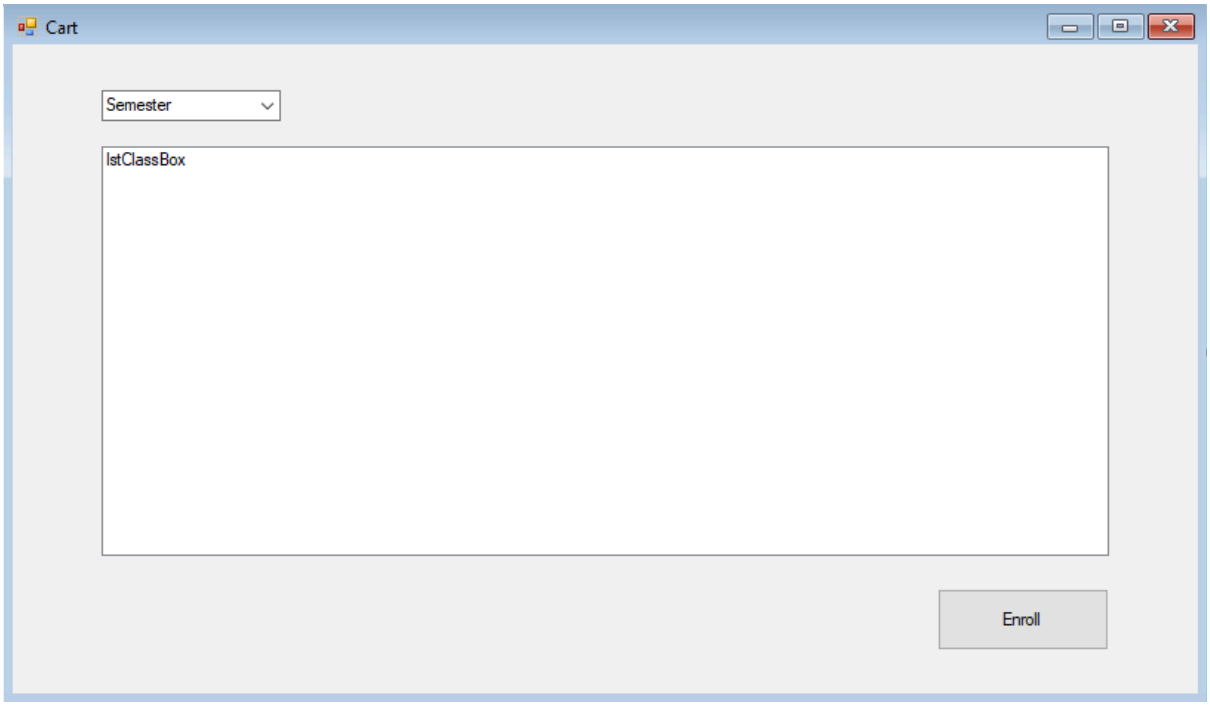
Screen Mock-ups



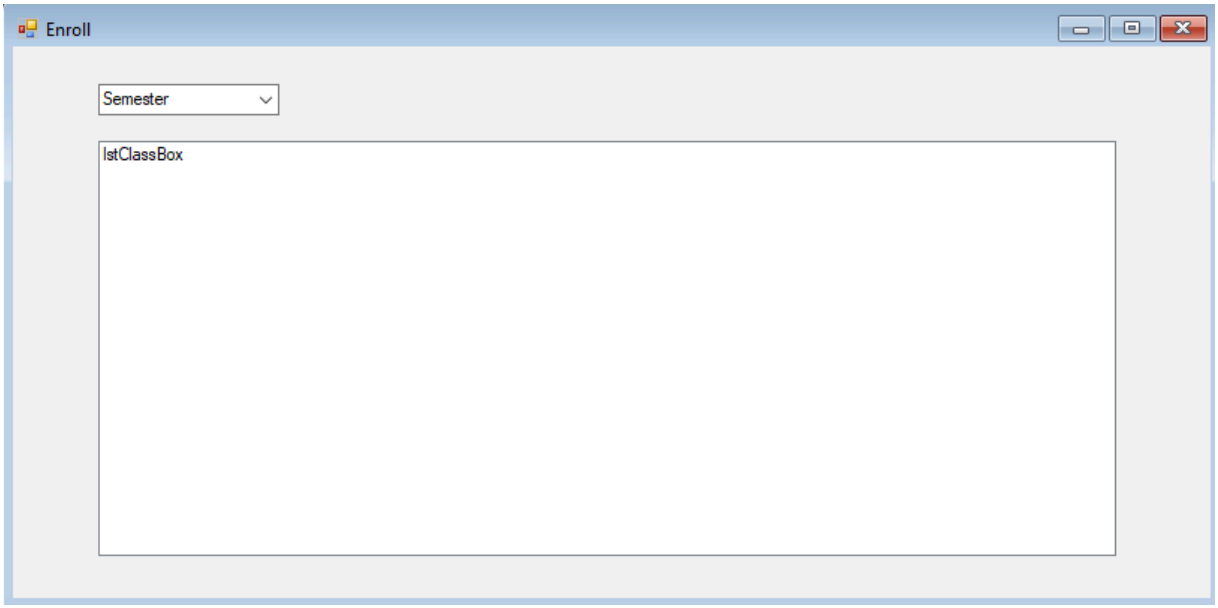
- Login screen that students will see first. Students use their id number as username to login



- Course search result
- Displays course information and section with date and time, as well as open seats available



- Students can select a course and add it to cart until they are ready to enroll.
- They can enroll to multiple courses at once



- Student's course history screen will display the courses they have taken
- We will use this information to see pre-req qualifications

Business Rules of Our System

1) Student must login to register into a class
2) Student must have prerequisites (if needed) to enroll into a class
3) Class enrollment must not be full for student to enroll
4) Class must not have time conflict with another class to enroll

Data Linking

Student and Section are a many-to-many relationship in the *Takes* table, since each student can enroll in multiple sections and each section can have many students. The *Takes* table also serves as a record for previously taken courses, where the grade is displayed once completed.

Section and Classroom are a many-to-one relationship since each section should only be happening in one classroom at a time, but classrooms can be used for multiple different sections.

Section and TimeSlot are a many-to-many relationship in the *SectionTimeSlot* table since each section can have multiple timeslots (ie. M/W/F 9:00-10:00, or Tu/Th 14:00-15:00) and each timeslot can be used for multiple different sections.

Section and Course are a many-to-one relationship since each section is for a specific course, while a course can have multiple sections (ex. AS01, AS02).

Section and Instructor are a many-to-one relationship since each section needs one instructor, and instructors can teach multiple sections. Since a section is either a lecture, lab, or seminar, a different instructor can teach a different lecture section, or the lab section for instance.

Course has a many-to-many relationship with itself in the *Prereq* table, where each course can require multiple courses as prerequisites, and each course can act as a prerequisite for multiple courses.

Course and Department are a many-to-one relationship since each course belongs to a single department (ex. CMPT101 belongs to Computer Science), but each department can have multiple courses.

Instructor and Department have two relationships with each other:

- Many-to-one where each instructor works in a single department, and a department can have many instructors working in it.
- One-to-one where each department must have an instructor as the department head, and each instructor can only be the head of at most one department.

Note: to get around this double-relationship, the departmentHeadID in the *Department* table is nullable, and then updated with instructors after both tables have been filled.

Integrity constraints

Each section in the *Section* table has a constraint checking the number of enrolled students is less than or equal to the section size.

Each course taken in the *Takes* table has a grade that is nullable to account for students enrolling in a section they haven't completed and received a grade yet.

Stored Procedures

spEnroll (@studentID, @sectionID) – uses transaction
Adds studentID and sectionID into Takes table using transaction. The procedure checks if the timeslotID does not already exist in the student's system, then compares the section capacity and the number of enrolled students. If the class is at its capacity, it will rollback.

spLogin (@studentID, @password)
Checks if the studentID and password entered match the record.

spStudentCourses (@studentID)
Returns the list of all the course information in the selected student's record.

spStudentCoursesByYear (@studentID, @semester, @year)
Returns the list of all the course information in the specific semester and year of the student.

spSectionEnrolled (@sectionID)
Returns the number of available seats in a section.

spStudentCourseSchedule (@studentID)
Returns the schedule with time.

spHistoricalStudentCount
Returns the number of all students, active students, and nonactive students.

spSearchCourseByCourseID (@courseID)
Returns the list of courses that match the search keyword.

spAllCourses
Returns the list of available courses for enrollment.

spViewClassesinSemester (@semester)
Returns list of courses that are offered in the selected semester, and it only displays the ones in the upcoming semesters. The procedure checks for the current year using CURRENT_TIMESTAMP, and it will filter out the courses that were offered in previous years and only display upcoming semesters.