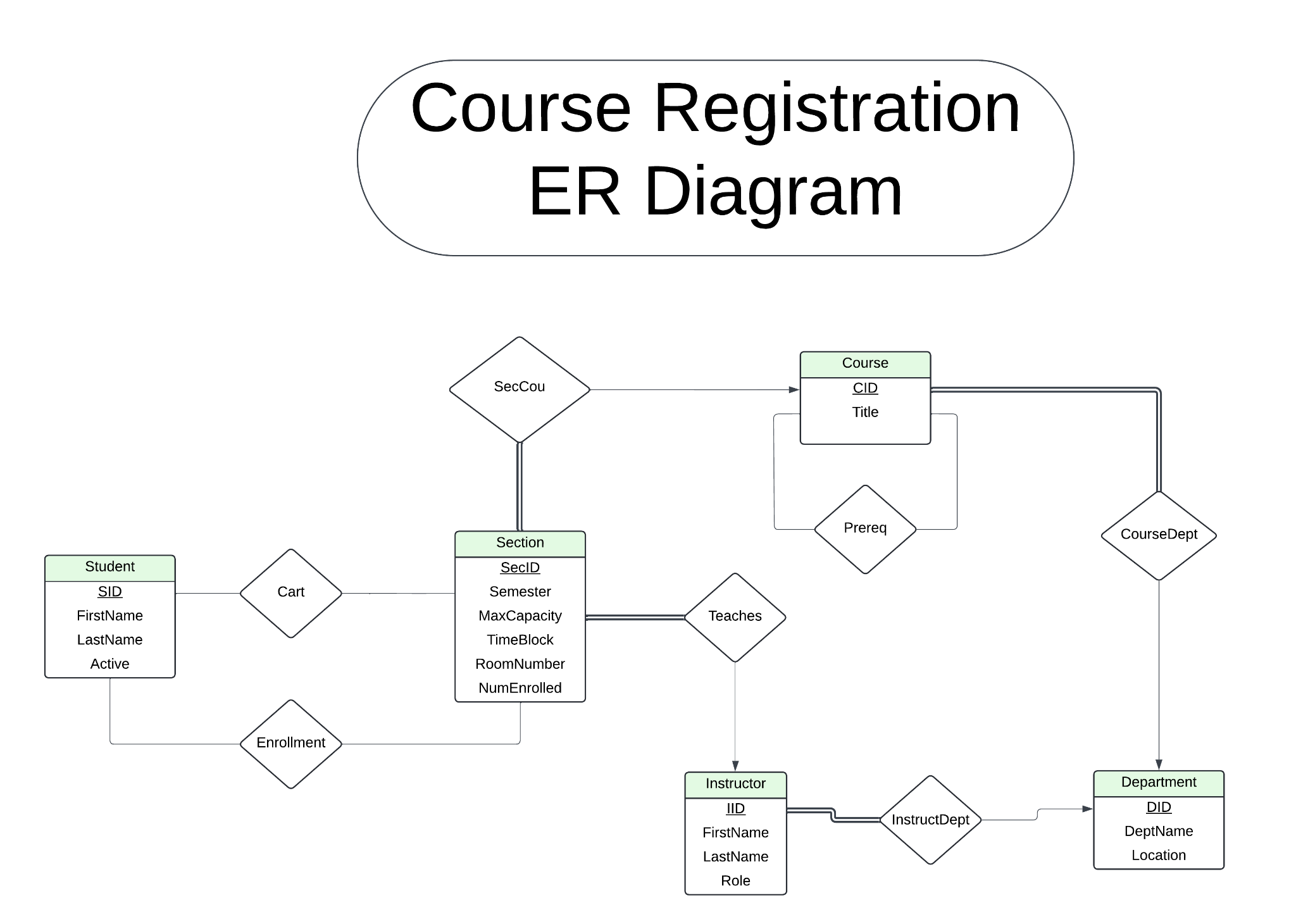
**Documentation For The School Registration System**

**ER Diagram:**

Entity-Relationship (E-R) Diagram: This diagram illustrates the complete database schema for your system.



**Relational Database Schema:**

The Student table has information on the student, including their first and last name, as well as a flag describing their current status (active or inactive). It contains a unique student ID that acts as the primary key for the relation.

The Department table contains information on departments in the university. It has the department name, location, and a unique department ID that acts as the primary key for the relation.

The Course table contains information regarding the courses offered. Its attributes are the title of the course, and a department ID (which is a foreign key referencing the Department relation). It uses a unique Course ID that acts as the primary key for the relation.

The Instructor table has information on the instructors in the university. Its attributes include the first and last name of the instructor, their role (sessional, associate, or head) and the department ID of the department they belong to (which is a foreign key referencing the Department relation). It uses a unique instructor ID that acts as the primary key for the relation.

The Section table stores all the information on each section of an offered course. It tracks the semester that the section is in, its max capacity, the timeblock (MWF or TR and the time of the course) and the room number. Importantly, it contains the course ID of the course associated with it (which is a foreign key referencing the Course relation) and the ID of the associated instructor (which is a foreign key referencing the Instructor relation). It uses a unique Section ID that acts as the primary key for the relation.

The Cart relation tracks information on the courses that have been added to the cart by each student. It contains the student ID of the student (a foreign key referencing the Student relation) and the section ID of the course section that the student has added to the cart (a foreign key referencing the Section relation). The primary key for this relation includes both the student ID and the section ID.

The Enrollment relation tracks information on the courses that have been enrolled in by each student, which are populated from rows from the Cart relation. Like the Cart relation, it contains the student ID of the student (a foreign key referencing the Student relation) and the section ID of the course section that the student has been enrolled in (a foreign key referencing the Section relation). The primary key for this relation includes both the student ID and the section ID.

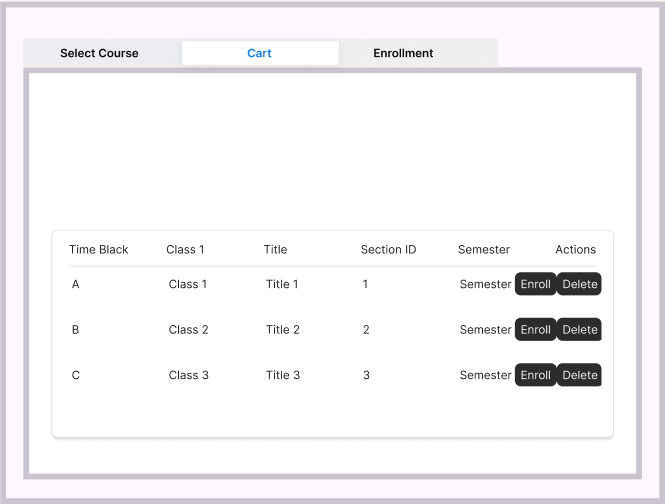
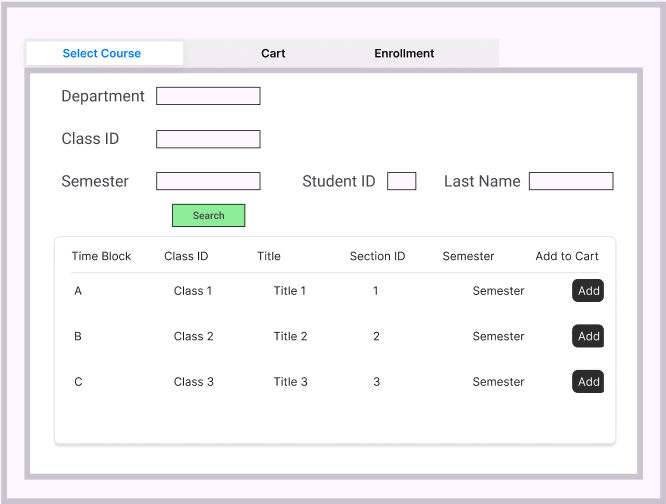
The Prerequisites relation contains information regarding the prerequisites required for each course, if such exist(s). It contains the prerequisite ID (which is a foreign key referencing the course relation) and the course ID of the course that it acts as a prerequisite for (which is a foreign key referencing the Course relation). The primary key for this relation includes both the prerequisite course ID and the course ID of the course it acts as a prerequisite for.

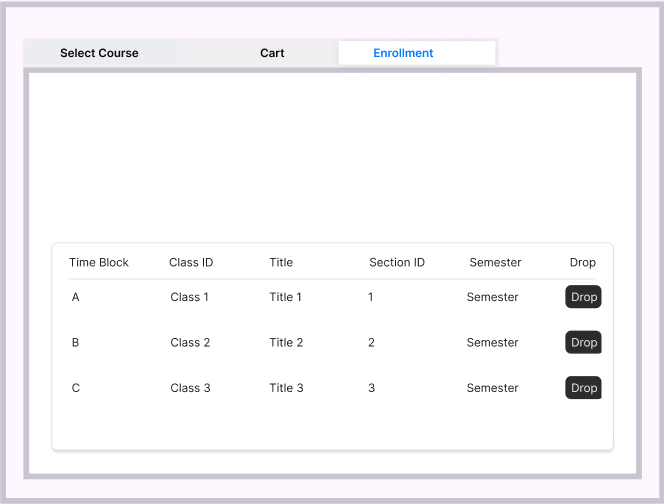
Design Rationale:

We attempted to follow the BNCF format, ensuring as little redundant information is contained within the tables as possible. Tables are modeled after real-world entities, which ensures that our schema is easy to understand and conceptualize. Each table also contains unique primary keys to identify each entry, and uses foreign keys from the other tables to ensure the consistency of data across the entire schema. Stored procedures store queries and their execution plan, saving resources on replicate queries and increasing performance. Transactions help ensure the atomicity, consistency, isolation and durability of the database. Materialized views physically store the result of often-referenced queries which do not need frequent updating, increasing the performance of queries related to the information stored. Thus, joins between three or more tables (for example: Student, Enrollment, and Section) do not have to be processed every time the information from the table is referenced if up-to-date records are not frequently required.

**Screen Mock-Ups:**

Visual representations of your system's user interface.





**Summary of Business Rules:**

A brief overview of the fundamental business rules that govern your system.

We currently have three tabs all on one screen. They are course selection, user cart, and enrollment page.  
  
Course Selection:

* It allows users to enter their student ID and last name to authenticate themselves, search for courses, put courses in their cart, and show their enrolled courses.
* The student can search for courses using the fields department, course ID, and semester.
* If the student wishes to add a course to their cart, they can click a button on the right-hand side of the course to add it to their cart.
* Once a student has decided to add a course to their cart, the system double-checks if the student has satisfied the prerequisites to enroll them into the course.
  + A student cannot add the course to the cart multiple times.

User Cart:

* Displays the courses the user currently has in their cart.
* The student can either remove the course from their cart or enroll in that course using one of two buttons on the right-hand side of the row.
* When a student enrolls in a course, the system automatically checks if the course is available and that no existing time conflicts exist.
  + If there are none, the student can freely enroll in the course.
  + Just like adding to the cart, the student cannot enroll in the same section ID OR course\_id.

Enrollment Page:

* Shows the courses the student is currently enrolled in for the current semester.
* If students wish to unenroll in a course, they can select the button on the right-hand side of the row to unenroll.