**ACADEMIC RECORD MANAGEMENT SYSYTEM**

**Phase 2: Business Process Modeling**

**Definition:**

**Objective:** To streamline and automate the management of academic records in educational institutions, including student registration, course enrollment, and grade tracking. This aligns with the MIS framework by ensuring data centralization, improving process efficiency, and enabling better decision making.

**Expected Outcomes:** A seamless academic records process that reduces manual tasks, minimizes data errors, and improves accessibility for students, instructors, and administrators.

**Key Entities and their Roles**

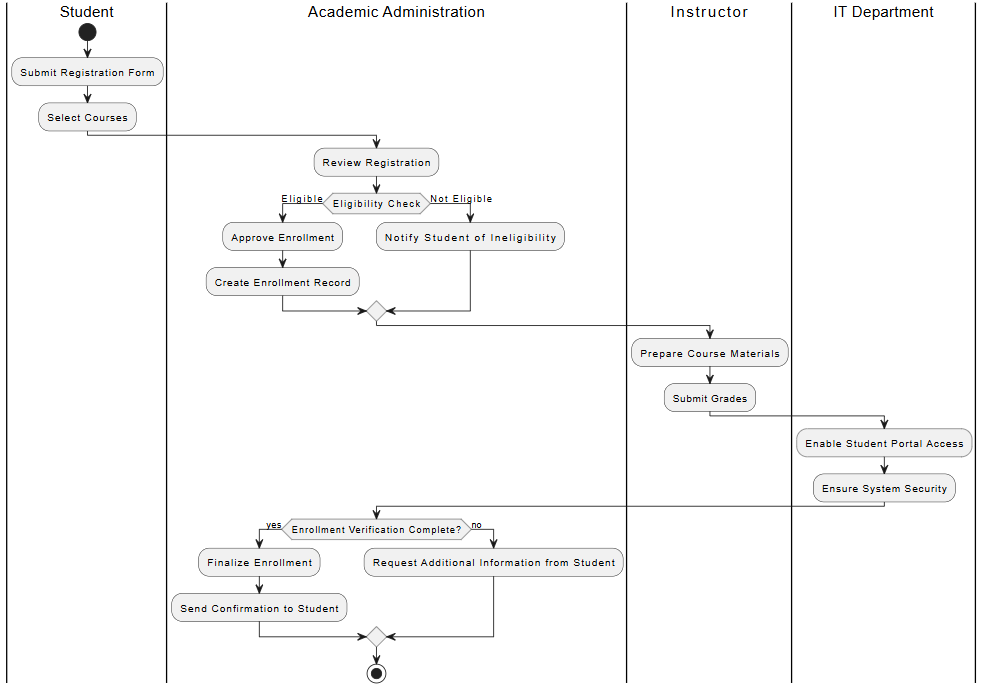
1. **Student:** Register for courses, view grades, and access academic information.
2. **Instructors:** Update course details, manage student grades, and view enrollment data.
3. **Academic Administration:** Processes student registrations, manages course and enrollment data, and oversees records for compliance and reporting.
4. **IT Department:** Manages system access, data security, and technical support for the Academic Records Management System (ARMS).
5. **Data Elements:** Include Student profile, Course catalog, Enrollment records, and Grade record.

**Swimlanes Diagram**

**Logical flow:**

1. **Registration request:** Students initiate registration, which goes to administration for approval.
2. **Course Enrollment:** Approved students select courses, creating enrollment records.
3. **Grade Submission:** Instructors submit grades and add courses for enrolled students.
4. **Record Review:** Administration review and finalizes records for accuracy and compliance.
5. **Security:** IT department ensures the security and maintains system integrity.

**Diagram:**

****

**Explanation:**

**Logical Flow:** the diagram begins with the student submitting a registration form and selecting courses, academic administration then verifies the student’s eligibility, with a decision point: if the student is eligible, enrollment is approved and recorded. If not, the student is notified of ineligibility.

**Phase 3: Logical Model Design**

**Entities and Attributes**

1. **Student:** StudentID (PK), Name, DOB, Email, ContactInfo, EnrollmentStatus.
2. **Courses:** CourseID (PK), Name, Credits, Department.
3. **Enrollments:** EnrollmentID (PK), StudentID (FK), CourseID (FK), EnrollmentDate.
4. **Instructors:** InstructorID (PK), Name, DOB, Email, ContactInfo.
5. **Grades:** GradeID (PK), StudentID (FK), CourseID (FK), Grade.

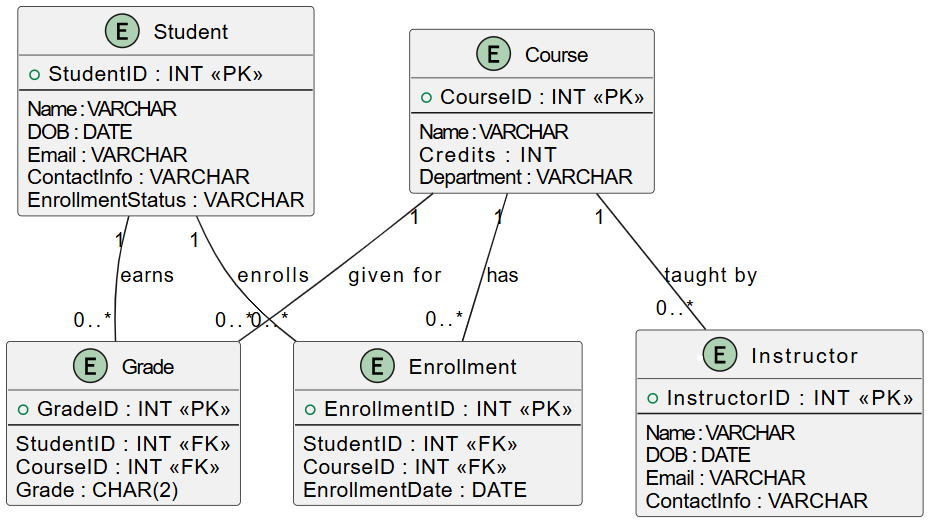
**Relationships:**

* **Students – Courses:** Many-to-many relationship, managed by Enrollments table.
* **Course – Instructors:** One-to-many relationship, each course can have multiple instructors.
* **Students – Grades:** One-to-many relationship, where each student can have multiple grades for different courses.

**Keys and Constraints:**

* **Primary Keys:** StudentID, CourseID, InstructorID, EnrollmentID, GradeID.
* **Foreign Keys:** StudentID and CourseID in Enrollment and Grades tables.
* **Other Constraints:** Set NOT NULL Constraint on the Primary Keys and enforce unique Constraint on necessary details.

**Entity Relationship Diagram (ERD):**

****