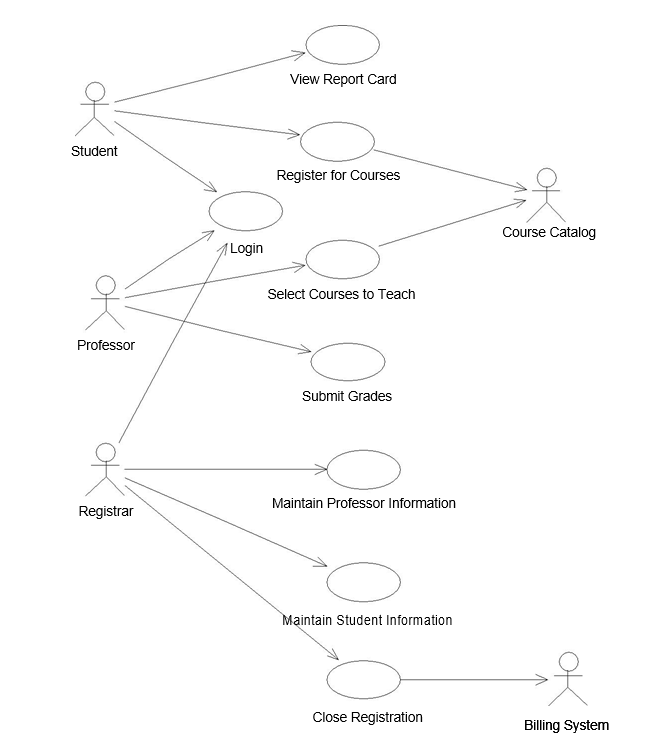
**Use-Case Model**

**Course Registration System Use-Case Model Main Diagram**



**Login**

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** Course Registration System | | |
| **Use Case name:** Login | |  |
| **Primary actor:** Student, Professor and Registrar | | **Other actors:** |
| **Description:** This use case describes how a user logs into the Course Registration System. | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Pre-conditions:**  The system is in the login state and has the login screen displayed. | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. This use case starts when the actor select login option 2. The actor provides his/her name and password. | * 1. The system request username and password.   2. The system validates the entered name and password and logs the actor into the system. | |
| **Alternative and exceptional flows:**  In step 2.1. : If the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to cancel the login, at which point the use case ends, or repeat from step 2. | | |
| **Post-conditions:**  If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged. | | |

**Select Courses to Teach**

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** Course Registration System | | |
| **Use Case name:** Select courses to teach | | |
| **Primary actor:** Professor | | **Other actors:** Course Offering Catalog |
| **Description:** This use case allows a Professor to select the course offerings from the course catalog for the courses that he/she is eligible for and wishes to teach in the upcoming semester. | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Pre-conditions:**  The Professor must be logged onto the system before this use case begins. | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. This use case starts when a Professor select choose courses to teach option. 2. The professor selects and/or de-selects the course offerings that he/she wishes to teach for the upcoming semester. | * 1. The system retrieves and displays the list of course offerings the professor is eligible to teach for the current semester. The system also retrieves and displays the list of courses the professor has previously selected to teach.   2. The system removes the professor from teaching the de-selected course offerings.   3. The system verifies that the selected offerings do not conflict (i.e., have the same dates and times) with each other or any course offerings that the professor has previously signed up to teach. If there is no conflict, the system updates the course offering information for each offering the professor selects (i.e., records the professor as the instructor for the course offering). | |
| **Alternative and exceptional flows:**  In step 2.2 : If the systems find a schedule conflict when trying to establish the course offerings the Professor should take, the system will display an error message indicating that a schedule conflict has occurred. The system will also indicate which the conflicting courses are. The Professor can either resolve the schedule conflict (i.e., by canceling his selection to teach one of the course offerings), or cancel the operation, in which case, any selections will be lost, and the use case ends. | | |
| **Post-conditions:**  If the use case was successful, the course offerings a Professor is scheduled to teach have been updated. Otherwise, the system state is unchanged. | | |

**Submit Grades**

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** Course Registration System | | |
| **Use Case name:** Submit Grades | | |
| **Primary actor:** Professor | | **Other actors:** |
| **Description:** This use case allows a Professor to submit student grades for one or more classes completed in the previous semester. | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Pre-conditions:**  The Professor must be logged onto the system before this use case begins. | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. This use case starts when a Professor select submit grades option 2. The Professor selects a course offering. 3. For each student on the list, the Professor enters a grade: A, B, C, D, F, or I. The system records the student’s grade for the course offering. If the Professor wishes to skip a particular student, the grade information can be left blank and filled in at a later time. The Professor may also change the grade for a student by entering a new grade. | * 1. The system displays a list of course offerings the Professor taught in the previous semester.   2. The system retrieves a list of all students who were registered for the course offering. The system displays each student and any grade that was previously assigned for the offering.   3. The system update students grades | |
| **Alternative and exceptional flows:**  In step 1.1 : If the Professor did not teach any course offerings in the previous semester, the system will display an error message. The Professor acknowledges the message, and the use case ends. | | |
| **Post-conditions:**  If the use case was successful, student grades for a course offering are updated. Otherwise, the system state is unchanged. | | |