Modeling a Problem Scenario with UML

1 Table of Contents

2 Problem Statement

Develop a client-server student registration system that will replace a legacy system that was based on mainframe technology. The new system will allow students to register for courses and view report cards from personal computers attached to the campus LAN. Professors will be able to access the system to sign up to teach courses as well as record grades. The system must be able to use the existing course catalog database available on the college’s DEC VAX machine.

3 Overview

3.1 Background

The character-based interface to the current mainframe registration system is requires repetitive keyboard entry and as such, is subject to erroneous data entries. As the student population of the college grows over time, the volume of student registration and data-entry intensive process of recording, retrieving and updating each record is getting to be tremendously ineffective for students and professors, as well as for the registrar and office support staff.

The college would like a new client-server system with a graphical “point-and-click” user interface to replace its much older system that was originally developed around mainframe technology.

3.2 Overall Description

The existing course catalog database is available on the college's DEC VAX machine. Among others, other systems of the college like Student Grading System, Financial Aid, and Billing Systems are on the same platform. These systems will remain as is, and the new registration system will be developed on top of the existing systems. Due to budget limitations, the client-server registration system will be developed first. Other systems on the DEC VAX system will follow upgrade path subject to the availability of college funds. However they should remain operational as each upgrade phase is delivered.

4 Constraints

Due to a decrease in federal funding, the college cannot afford to replace the entire system at once. The college will keep the existing course catalog database in the current platform where all course information is maintained.

5 Functional Specifications

Besides an enhanced graphical user interface, the client-server registration system is shall provide enhanced functionalities that shall primarily address the student's registration needs and professor's course and grade data-entry functions.

5.1 Student

- The Registration System shall authenticate students before using the system
- The Registration System shall allow students to register for courses
- The Registration System shall allow students to view report cards

5.2 Professor

- The Registration System shall authenticate Professors before using the system
- The Registration System shall allow Professors to indicate which courses they will be teaching
- The Registration System shall allow Professors need to see which students signed up for their course offerings
- Professors shall be allowed to record the grades for the students in each class

5.3 Registrar

- The registrar's office will continue to maintain course information through another system
- Maintain related information about each course, such as professor, department, and prerequisites, etc.

5.4 Billing

- Registration system sends information to the billing system so the student can be billed for the semester

6 Business Requirements

- The Registrar sets up the curriculum for a semester
- Course offerings will have a maximum of ten students and a minimum of three students
- A course offering with fewer than three students will be canceled
- Students select 4 primary courses and 2 alternate courses
- Once a student registers for a semester, the billing system is notified so the student may be billed for the semester

- For each semester, students may use the system to add/drop courses for a period of time after registration
- If a course fills up during the actual registration process, the student must be notified of the change before submitting the schedule for processing
- Professors use the system to receive their course offering rosters
- Users of the registration system are assigned passwords that are used at logon validation

7 UML

Modeling the problem scenario(s) using unified modeling language

7.1 Use Case Analysis

7.1.1 Description

7.1.1.1 Student registration

1. Enter the Student ID and PIN
2. Validate entered Student ID and PIN
3. Enter/select upcoming academic semester-year
4. Display course catalog
5. Enter/select course to register
6. Determine course pre-requisite(s)
7. Add course number to student's registered courses
8. Display/refresh list of registered courses

7.1.1.2 View Student Grades

1. Enter the Student ID and PIN
2. Validate entered Student ID and PIN
3. Enter/select academic semester-year of courses taken
4. Display grade(s) earned for courses taken

7.1.1.3 Professor's intent to teach course(s)

1. Enter the Professor's ID and PIN
2. Validate entered Professor's ID and PIN
3. Enter/select upcoming academic semester-year
4. Display course catalog
5. Enter/select intended course to teach
6. Determine instructor's eligibility to teach chosen course
7. Add course number to professor's intended courses to teach
8. Display/refresh list of professor's intended courses to teach

7.1.1.4 Professor's grade entries

1. Enter the Professor's ID and PIN
2. Validate entered Professor's ID and PIN
3. Enter/select current or recently concluded academic semester
4. Enter/select current or recently concluded course(s) taught
5. Enter grade to each student's record in class
6. Send grades to Grading System

7.1.1.5 View Student sign-up for course to teach

1. Enter the Professor's ID and PIN
2. Validate entered Professor's ID and PIN

3. Enter/select upcoming academic semester-year
4. Display list of professor's intended courses to teach
5. Display names of students who signed up for course to teach
6. Display count of students who signed up for course to teach

7.1.1.6 Maintain Updates to Professor Information

1. Enter the Registrar's ID and PIN
2. Validate entered Registrar's ID and PIN
3. Display/View current list of college professors
4. Select professor record to update
5. Update record
6. Send updated record Professor database

7.1.1.7 Maintain Updates to Student's Information

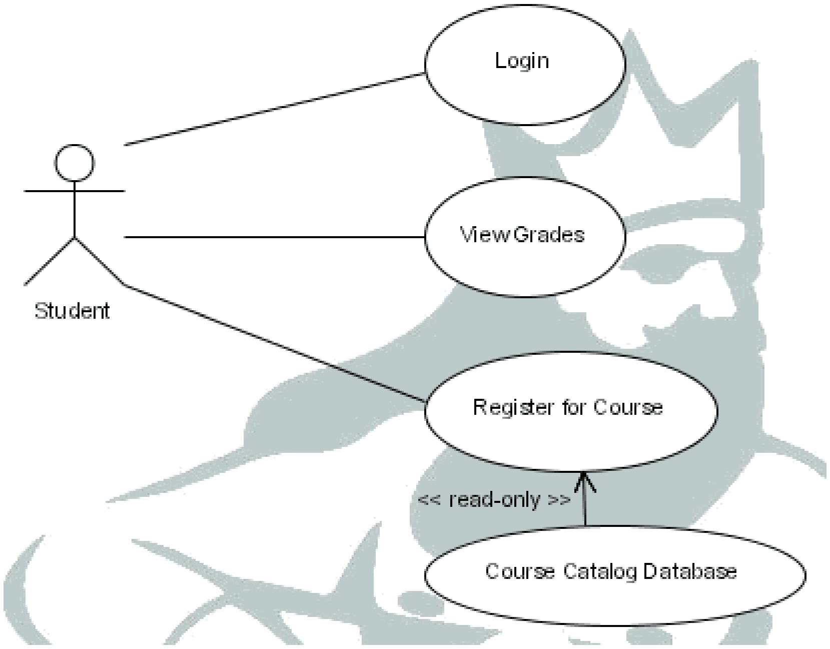
1. Enter the Registrar's ID and PIN
2. Validate entered Registrar's ID and PIN
3. Display/View current list of current students
4. Select student record to update
5. Update record
6. Send updated record Student database

7.1.1.8 Student Billing

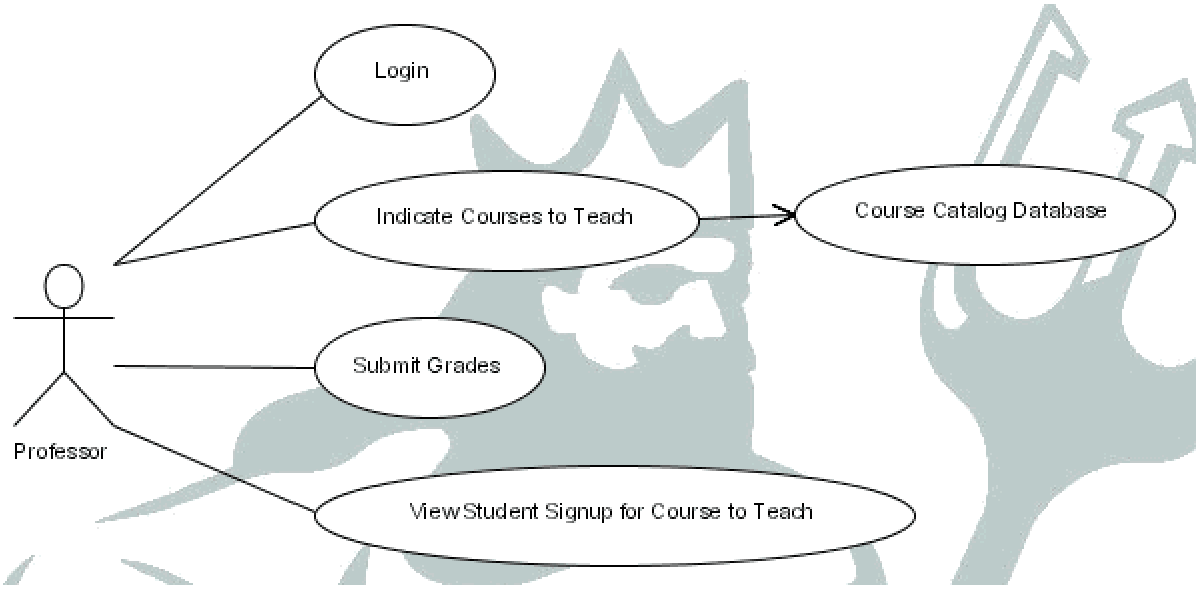
1. Retrieve completed registration records
2. Compute student's total bill amount for courses
3. Retrieve student name and mailing address
4. Send bill information to student

7.1.2 Diagrams

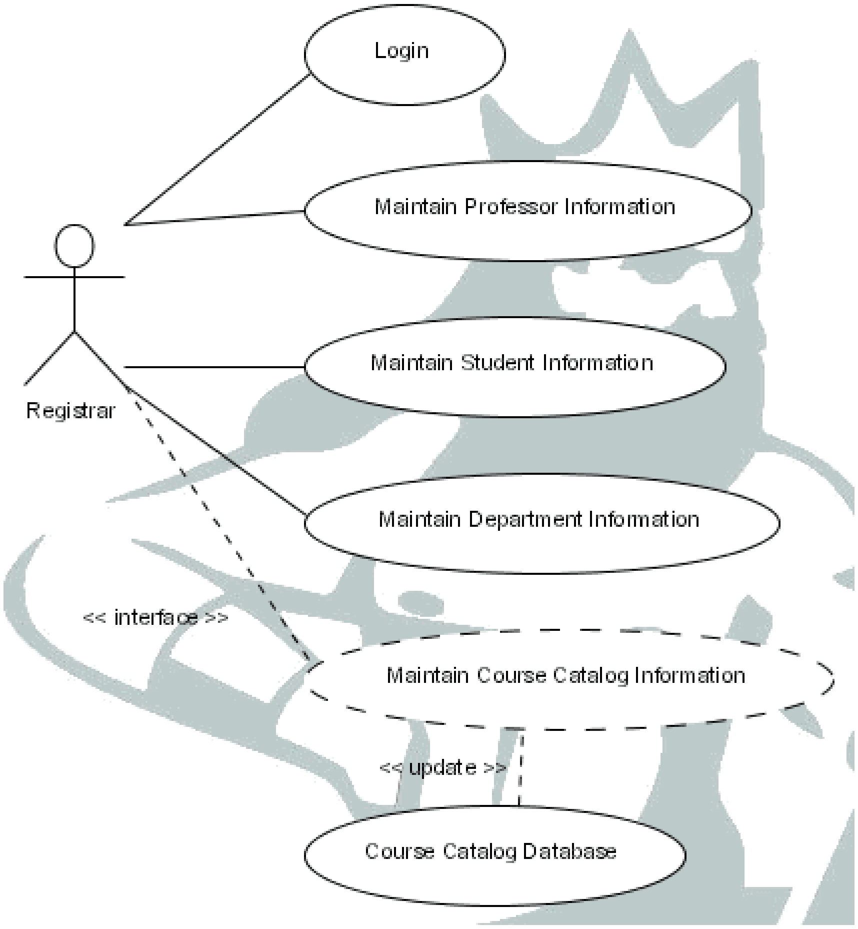
7.1.2.1 Student registration use case



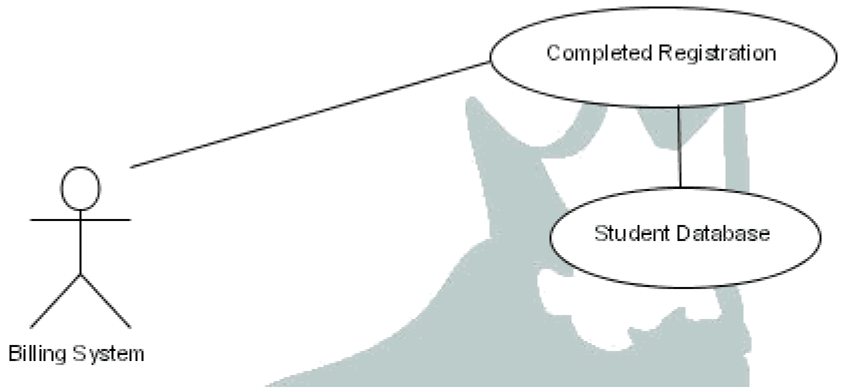
7.1.2.2 Professor grade entry use case


Login
Indicate Courses to Teach
Course Catalog Database
Submit Grades
Professor
ViewStudent Signup for Course to Teach

7.1.2.3 Registrar data maintenance use case


Login
Maintain Professor Information
Maintain Student Information
Registrar
Maintain Department Information
<< interface >>
Maintain Course Catalog Information
<< update >>
Course Catalog Database

7.1.2.4 Student billing use case



7.1.3 Specifications

Use Case ID: 7.1.2.1

Use Case Name: Student registration

Relevant requirements: Functional specification section 5.1

Primary Actor: Student

Pre-conditions: Student must be admitted to college and assigned a user identification and PIN

Post-conditions: Student will be registered for at least one course offered by the college

Basic Flow or Main Scenario: *See Use case description 7.1.1.1*

Extensions or Alternate Flows: None

Exceptions:

1. Pre-requisite courses not met
2. De-activated user id or PIN
3. Registration stop for non-payment of outstanding bill
4. Course catalog unavailable

Related Use Cases: Student view grade

Revision History –

Date	Description	By
11/20/2003	Student registration Use Case	RGP

Use Case ID: 7.1.2.2

Use Case Name: Professor grade entry

Relevant requirements: Functional specification section 5.2

Primary Actor: Professor

Pre-conditions: Professor must have taught at least one course at the college

Post-conditions: Students who finish course from instructor will have a grade for course taken

Basic Flow or Main Scenario: *See Use case description 7.1.1.4*

Extensions or Alternate Flows: None

Exceptions:

1. Course taught not in course catalog
2. Professor's login is de-activated

Related Use Cases: Professor intent to teach course

Revision History –

Date	Description	By
11/20/2003	Professor Grade Entry Use Case	RGP

Use Case ID: 7.1.2.3

Use Case Name: Registrar Data Maintenance Use Case

Relevant requirements: Functional specification section 5.3

Primary Actor: Registrar

Pre-conditions: Registrar has been defined in system through user identification and PIN with Registrar privileges

Post-conditions: Professor and or Student college records will be updated

Basic Flow or Main Scenario: *See Use case description 7.1.1.7*

Extensions or Alternate Flows: None

Exceptions:

1. Student dropped from student database (inactivated due to dismissal, separation, etc)
2. Professor dropped from staff/instructor database (inactivated due to dismissal, separation, etc)

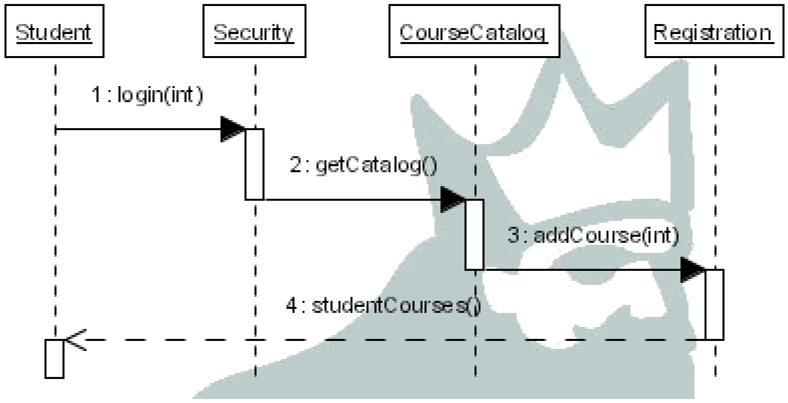
Related Use Cases: Student view grade

Revision History –

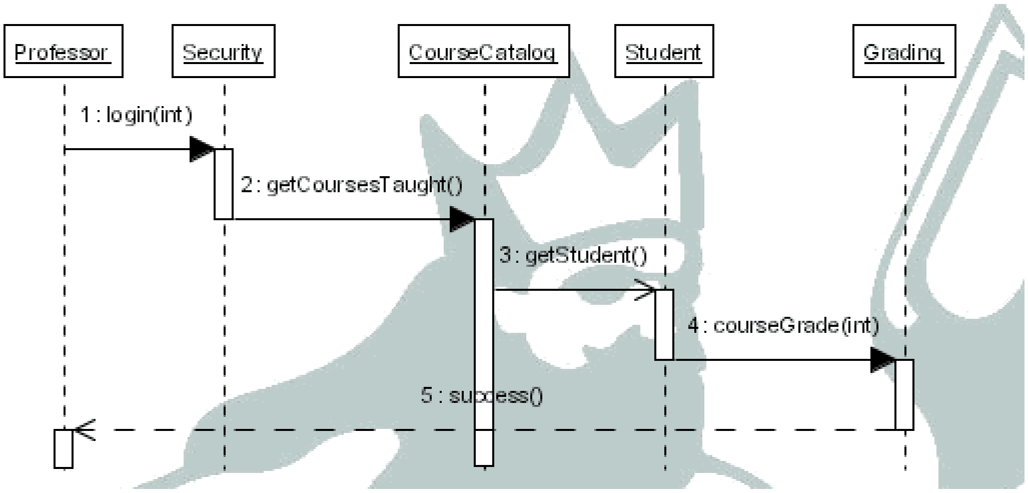
Date	Description	By
11/20/2003	Registrar Data Maintenance Use Case	RGP

8 Sequence Diagrams

8.1 Student Registration



8.2 Grade Entry


Professor
Security
CourseCatalog
Student
Grading
1 : login(int)
2 : getCoursesTaught()
3 : getStudent()
4 : courseGrade(int)
5 : success()