Software Requirements Specification

for

Course Management System

Version 2.0 approved

Prepared by Ujjwal Pasupulety

Department of Information Technology, NITK

27th February, 2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Ujjwal Pasupulety	**27-02-2018**	**Added UML Diagrams and NFRs**	**2.0**

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the first version of the "Course Management System" (CMS) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications.

1.2 Document Conventions

- *Convention for Main Title*
  i) Font face: Times New Roman
  ii) Font style: Bold
  iii) Font size: 18

- *Convention for Sub Title*
  i) Font face: Times New Roman
  ii) Font style: Bold
  iii) Font size: 14

- *Convention for Body*
  i) Font face: Arial
  ii) Font style: Italic
  iii) Font size: 11

- *Entire document must be justified*

1.3 Intended Audience and Reading Suggestions

This document is primarily intended to be proposed to a customer for the approval of the software and also serves as a reference for developing the first version of the system for the development team.

1.4 Product Scope

The CMS software helps college-going students manage their semester credit requirements through an easy-to-use application. It displays all the available courses in a given semester which are uploaded by the faculty through a separate application and a student can enroll for a given course. It also warns the student in case he/she falls short of the minimum credit requirements. This avoids any confusion during course registration.

1.5 References

1. Title: Course Management System
Version: Prototype 1.0, Date: November 2016,
Authors: Ujjwal Pasupulety, Rajeev Gundavarapu, Aiman Abdullah Anees, Jyoti Prakash Sahoo
Source: https://github.com/ujdcodr/javaproject

2. Hardware Requirements
Source:https://www.umass.edu/it/support/hardware/recommended-minimum-computer-configurations-windows

2. Overall Description

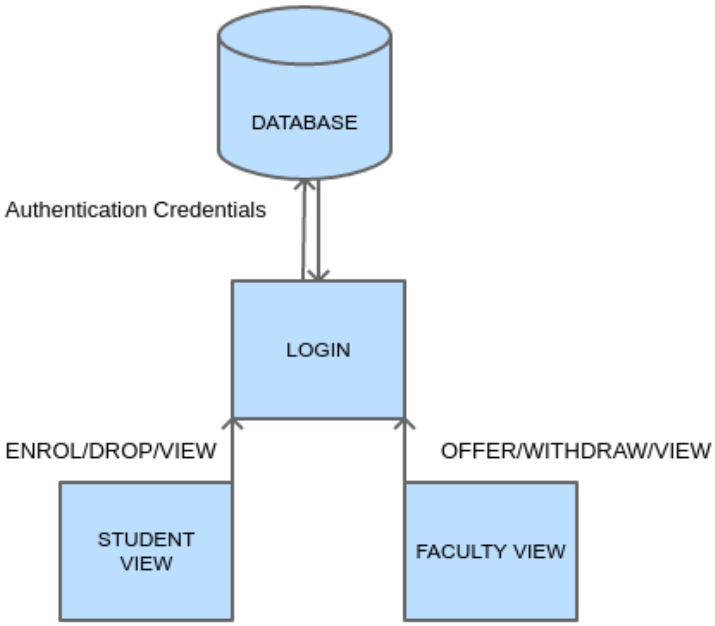
2.1 Product Perspective

The manual method of course registration requires students to be physically present and fill out forms which is quite inconvenient. There is the added hassle of fulfilling credit requirements where not all students are sure as to how many courses to take. Making a course available for students to opt for by the faculty is also a very tedious procedure and involves a lot of paperwork. The need of the hour is a robust, online and simpler way to register for courses and keep track of the semester credit requirements for students as well as for faculty to release new course offerings. The CMS software is a new self-contained product made for the ease of both parties,the students and faculty and serves as a replacement for the old system of manually registering and maintaining spreadsheets containing information about the courses offered, the number of students opting, etc.

2.2 Product Functions

Major functions that can be performed include:

- *View all available courses for a given semester*
- *User Login*
- *Select courses to take and resolve credit requirement discrepancies(if any)*
- *Upload information for new courses or withdraw them*
- *View all students who have opted for their course*



2.3 User Classes and Characteristics

The product has two primary users who use the CMS through different application interfaces, namely:

- *Student*
  i) View all available courses for a given semester
  ii) Select courses to take and resolve credit requirement discrepancies(if any)

- *Faculty*
  i) Upload information for new courses
  ii) View all students who have opted for their course

Each one uses a separate subset of the functions described in section 2.2 and both are disjoint. Both student and faculty accounts are secured through a common login page.

2.4 Operating Environment

Recommended

Processor (CPU): Intel Core i5-6xxx or equivalent

Operating System:	*Ubuntu 16.0 LTS and above*
Memory:	*8 GB RAM*
Storage:	*500 GB internal hard drive*

Monitor/Display: 14" LCD monitor, resolution of 1600 x 900 or better.

Network Adapter: 802.11ac 2.4/5 GHz wireless adaptor

Minimum

Processor (CPU): Intel Core i3-3xxx or equivalent

Operating System: Ubuntu 14.0 LTS and above

Memory: 4 GB RAM

Storage: 500 GB internal hard drive

CD-ROM: DVD/CD-RW (optional)

Monitor/Display: 13" LCD monitor

Network Adapter: Dual-band 802.11a/g - compliant adapter

2.5 Design and Implementation Constraints

Institute policies state that no form of internal data analytics must take place at the backend of the application to ensure security and privacy of both the student and faculty data. Application must be designed keeping low-grade hardware in mind as well based on the minimum specifications mentioned in Section 2.4 so that the user experience doesn't fall below a certain level of expected performance. The application will store the data on a document-oriented database program(like MongoDB). The Python programming language will be used for creating the modules. The application user interface will be constructed using GUI Programming libraries such a Tkinter or PyQt. Since the CMS software makes use of Open Source Software, the customer's organization will be responsible for maintaining the delivered software.

2.6 User Documentation

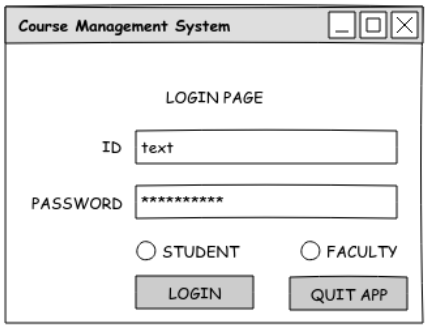
On-line quick start guide – HTML Webpage
User manual – Portable Document Format

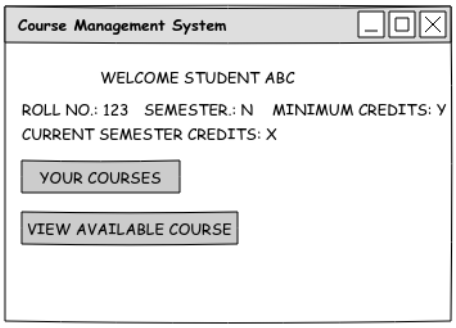
2.7 Assumptions and Dependencies

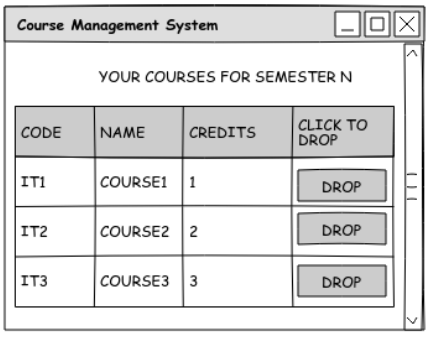
The CMS software has the following dependencies:
Python
MongoDB
Tkinter/PyQt(for GUI)

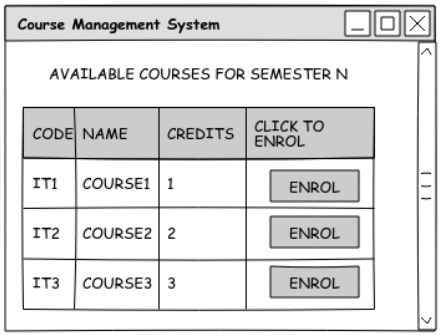
3. External Interface Requirements

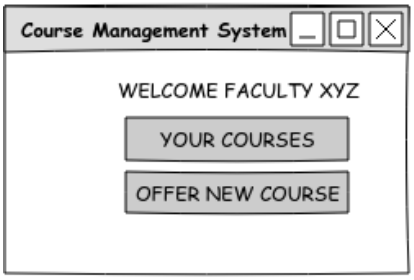
3.1 User Interfaces

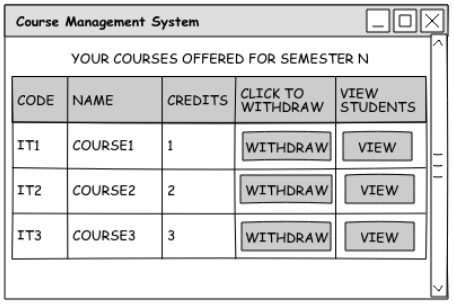


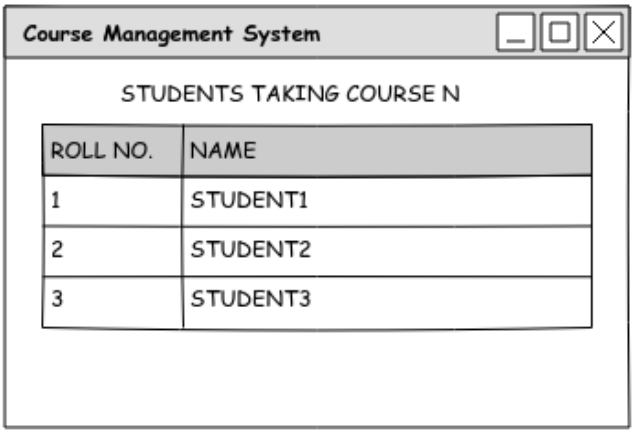












Protootypes were constructed using the Evolus Pencil tool. The application has a separate login for faculty and students. A student can view all the courses available and enrol for them. He/she can also view the courses they are enrolled in and drop them. Faculty can offer new courses and also view the details of the students who have enrolled for their courses.

3.2 Hardware Interfaces

The CMS software can be run on the minimum hardware settings specified in Section 2.4 and basic peripherals such as a mouse and keyboard for interacting with the graphical user interface to enter text for login authentication or new course details and clicking buttons respectively.

3.3 Software Interfaces

The CMS application connects to a document oriented database(such as MongoDB) to store the data of the students and faculty and makes use of GUI libraries such as Tkinter/PyQt for the user interface.

Important incoming data items and messages include:
- *User Login Credentials*
- *New Course Information*
- *Mouse click events for Enrolling/Dropping/Withdrawing a course or Viewing Students*

Important outging data items and messages include:
- *New student information to be added to the enrolled list for a coursework*
- *New course Information added to the available course list*

3.4 Communications Interfaces

There are two routes that can be taken. The CMS is designed to work only on a single standalone system for prototyping and presentation purposes, no communication interface will be required.

4. System Features

4.1 Login Screen

4.1.1 Description and Priority

Allows the main end users, students and faculty to log into their respective accounts through a password protected interface. Priority: 9

4.1.2 Stimulus/Response Sequences

For Students:
Enter Student ID and password and check "Student" Option → Display student start screen

For Faculty:
Enter Faculty ID and password and check "Faculty" Option → Display faculty start screen

4.1.3 Functional Requirements

REQ-1: Takes in Institute email ID as input. Reports error and refreshes the login screen if any other input is provided.

REQ-2: Accepts password containing alphanumeric characters along with special characters Reports error and refreshes the login screen if any other input is provided.

REQ-3: Login procedure only commences after an appropriate checkbox is checked. Pressing "Login" without checking a box results in an error message and refreshes the login page.

REQ-4: "Quit App" button shutsdown the entire application, but retains the student and Faculty data.

4.2 View available courses for a given semester

4.1.1 Description and Priority

This feature lets students view the courses they can enrol for so that they can make the decision as to which courses to choose so that they can manage their semester credit requirements. Priority: 4

4.1.2 Stimulus/Response Sequences

Log in as a student → Display student start screen
Click the "View Available Course" button → Display list of courses in a table
Click "Enrol" → Return to student start screen with updated credits

4.1.3 Functional Requirements

REQ-1: The list displayed after clicking "View Available Course" should only display the courses that have not already been taken by the given student in a tabular form.

REQ-2: After selecting the course, the associated credits must be added to the students total semester credits. The app should return to the student start page displaying the updated credits and a message should be displayed whether the student fulfills the minimum credit requirement or not.

REQ-3: The new course must be appended to the list which is displayed on clicking the "Your Courses" button in the student start page and also in the faculty's "View Student" list who is handling that course. It must not be present in the list displayed after clicking "View Available Course".

4.3 Drop a Course

4.1.1 Description and Priority

This feature lets students view the courses they have been enrolled to so that they can make the decision as to which courses to drop so that they can manage their semester credit requirements. Priority: 6

4.1.2 Stimulus/Response Sequences

Log in as a student → Display student start screen
Click the "Your Courses" button → Display list of courses in a table
Click "Drop" → Return to student start screen with updated credits

4.1.3 Functional Requirements

REQ-1: The list displayed after clicking "Your Courses" should only display the courses that have been taken by the given student in a tabular form.

REQ-2: After selecting the course, the associated credits must be subtracted from the students total semester credits. The app should return to the student start page displaying the updated credits and a message should be displayed whether the student fulfills the minimum credit requirement or not.

REQ-3: The new course must be appended to the list which is displayed on clicking the "View Available Course" button and must not be present in the list displayed after clicking "Your Courses" in the student start page and also in the faculty's "View Student" list who is handling that course.

4.4 Offer a course

4.1.1 Description and Priority

This feature lets faculty offer a new course. Priority: 6

4.1.2 Stimulus/Response Sequences

Log in as a faculty → Display faculty start screen
Click the "Offer New Course" button → Display text fields for course information
Fill detailes and Click "Offer Course" → Return to faculty start screen

4.1.3 Functional Requirements

REQ-1: Accepts course information containing alphanumeric characters. Reports error and returns to faculty start screen the login screen if any other input is provided.

REQ-2: After offering the new course, it must be appended in the list diaplayed after clicking "Your courses" button in the faculty start page and also update the list of available courses for students.

4.5 View students taking a Course

4.1.1 Description and Priority

This feature lets faculty view the details of the students who have enrolled for a course offered by them.. Priority: 6

4.1.2 Stimulus/Response Sequences

Log in as a student → Display student start screen
Click the "Your Courses" button → Display list of courses in a table
Click "View Students" → Enrolled Students screen

4.1.3 Functional Requirements

REQ-1: After withdrawing course, it must be removed in the list diaplayed after clicking "Your courses" button in the faculty start page and also from the list of available courses for students.

4.6 Withdraw a Course

4.1.1 Description and Priority

This feature lets faculty withdraw a new course. Priority: 6

4.1.2 Stimulus/Response Sequences

Log in as a faculty → Display faculty start screen
Click the "Your Courses" button → Display list of courses in a table
Click "Withdraw" → Return to faculty start screen

4.1.3 Functional Requirements

REQ-1: After withdrawing course, it must be removed in the list diaplayed after clicking "Your courses" button in the faculty start page and also from the list of available courses for students.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Since system is closed and doesn't involve an internet connection, the different GUI forms should ideally load instantly.

5.2 Safety Requirements

Error checking in input fields to prevent unaccepted inputs

5.3 Security Requirements

Login password must have atleast one special character and atleast one number

5.4 Software Quality Attributes

Robust and Reliable. Minimal to no crashing even on high system loads
Usability: GUI must be intuitive
Testability:Must pass all test cases before shipping

5.5 Business Rules

Student can
i) View available courses
ii) Enroll for a course
iii) View courses taken

Instructor can
i) Offer a new Course
ii) withdraw a coursework
iii) View students who have taken their course

6. Other Requirements

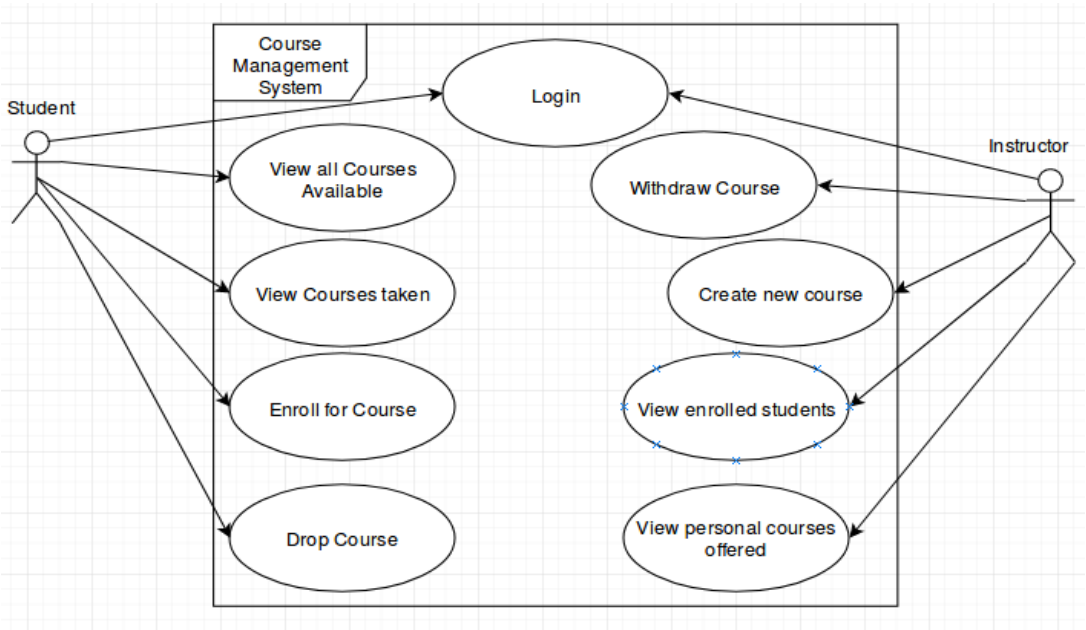
Database must possess ACID properties and be able to handle concurrent operations on it with out affecting the consistency of the stored data.

Appendix A: Glossary

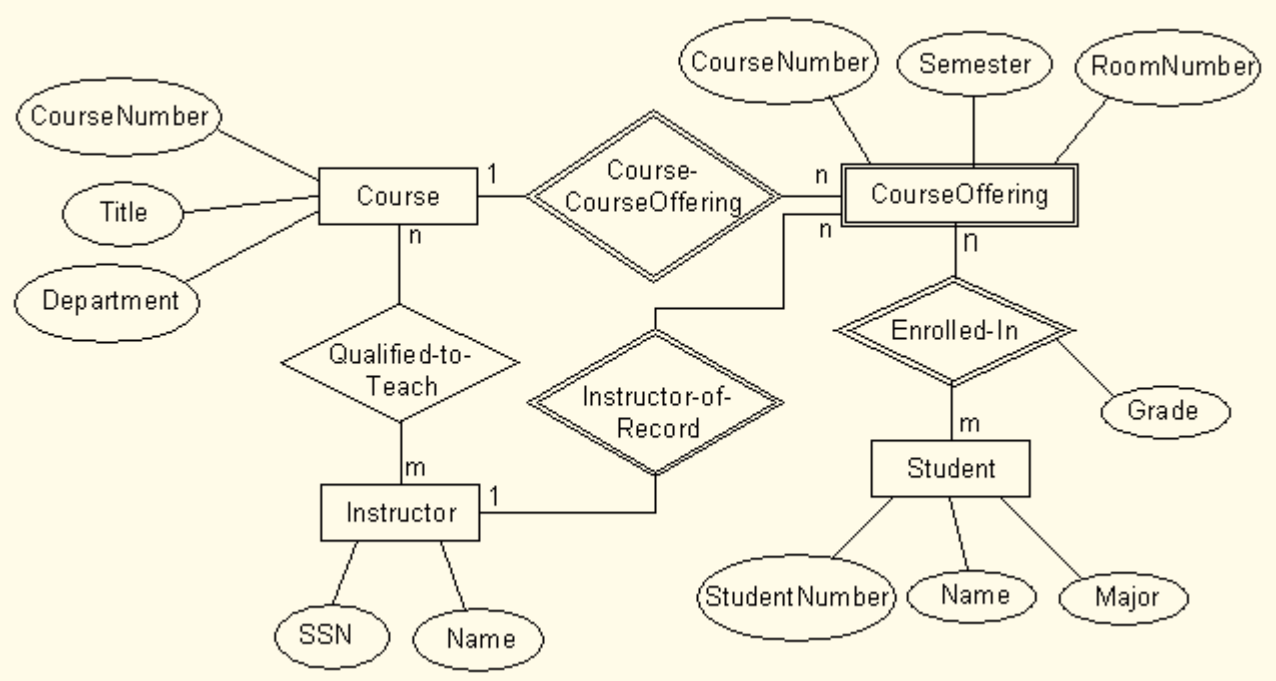
CMS – Course Management System
GUI – Graphical User Interfaces
HTTP – Hyper Text Transfer Protocol
ACID – Atomic, Consistent, Isolated, Durable
TBD – To Be Discussed
REQ - Requirement

Appendix B: Analysis Models

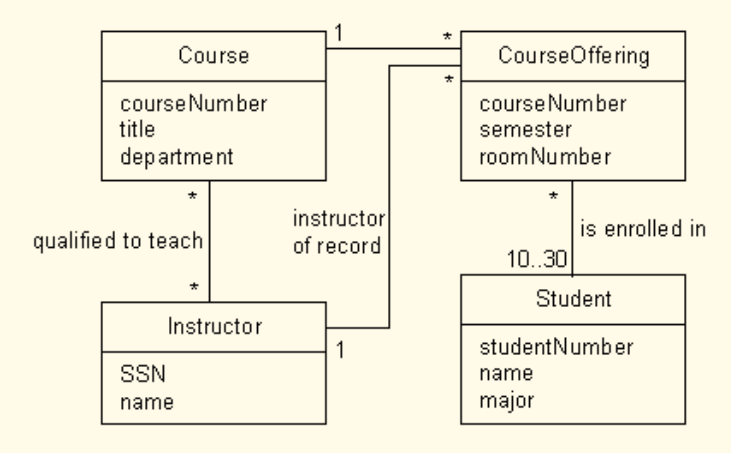
Use case



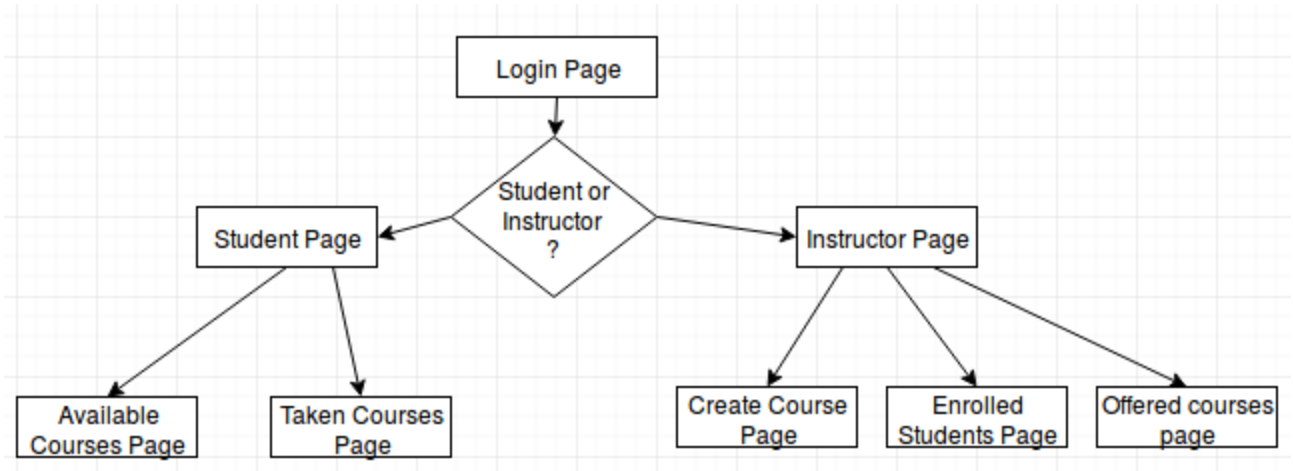
Entity-Relationship



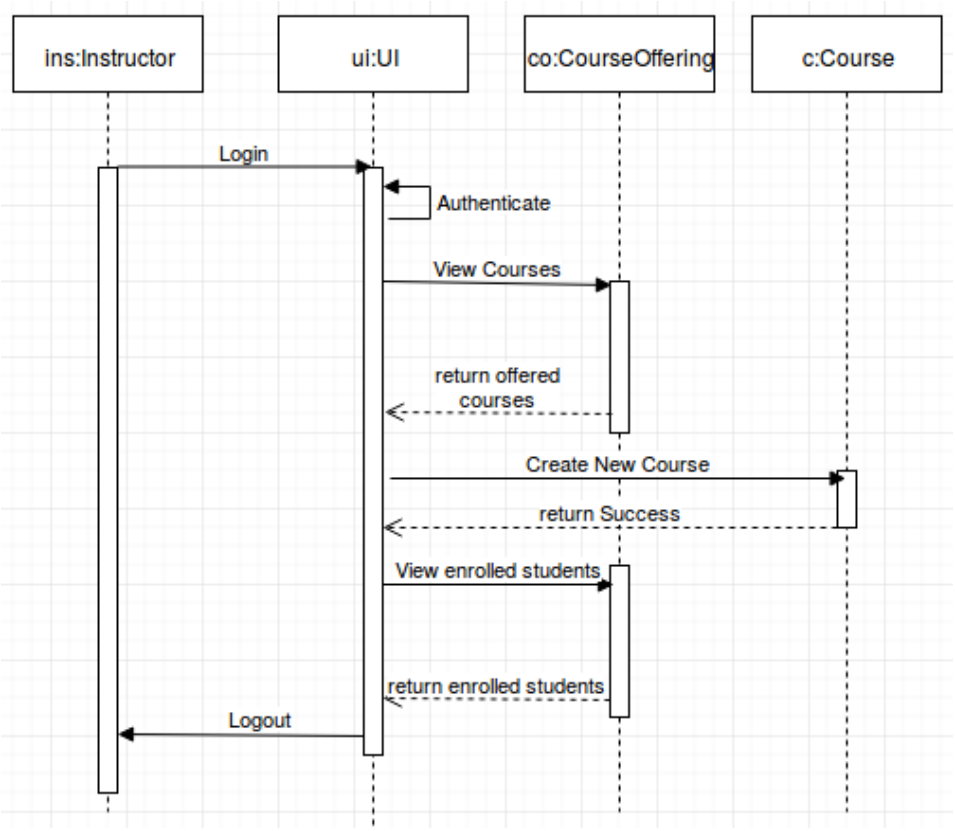
Class

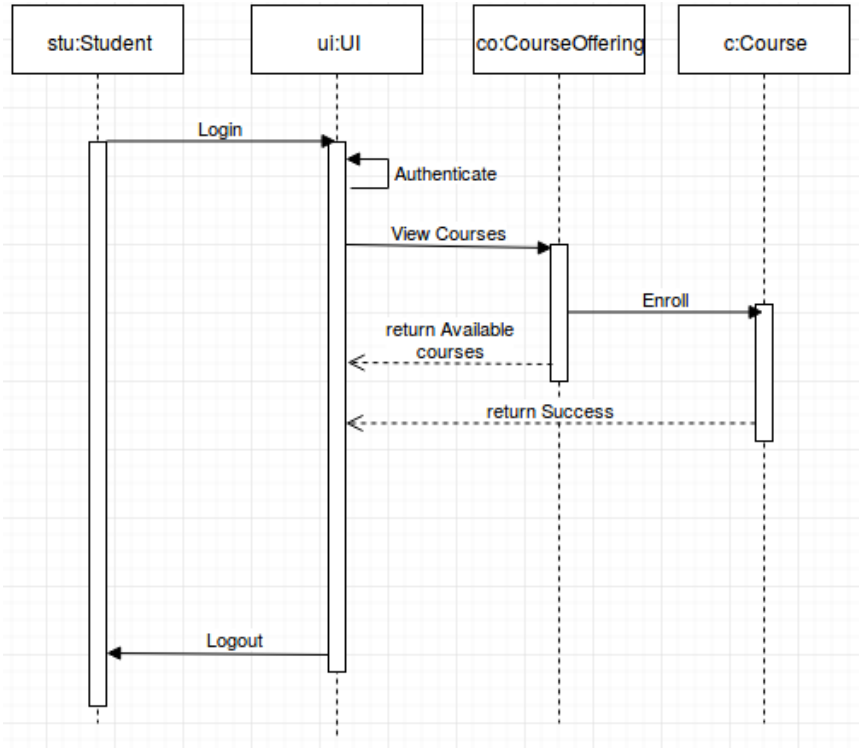


Navigation

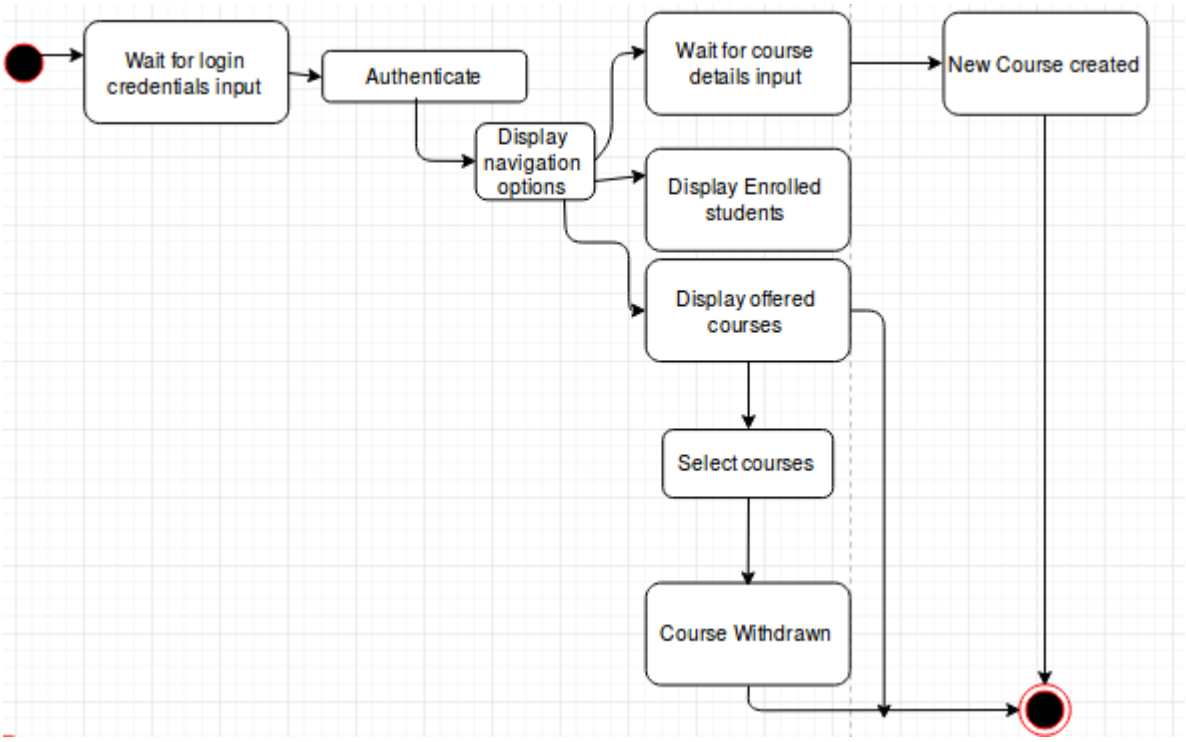


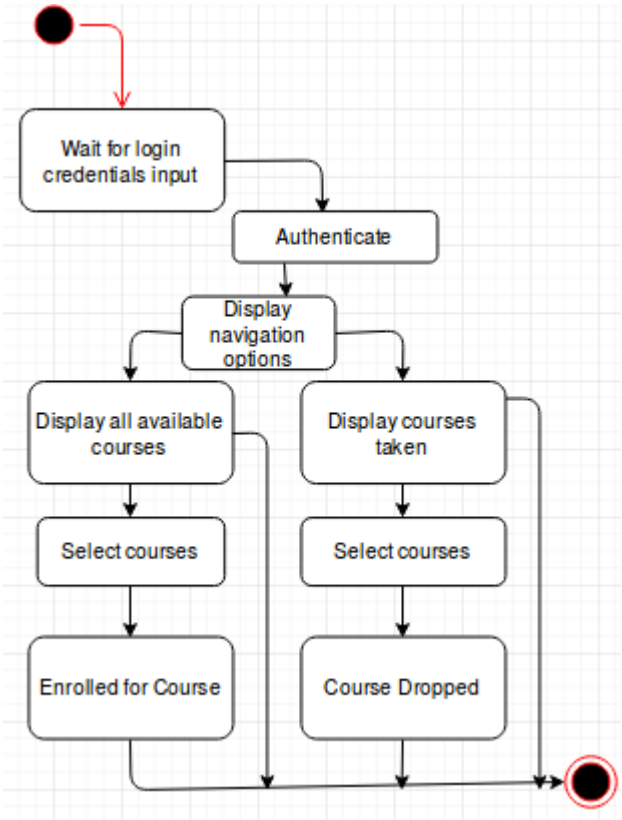
Sequence


ins:Instructor
ui:UI
co:CourseOffering
c:Course
Login
Authenticate
View Courses
return offered courses
Create New Course
return Success
View enrolled students
return enrolled students
Logout

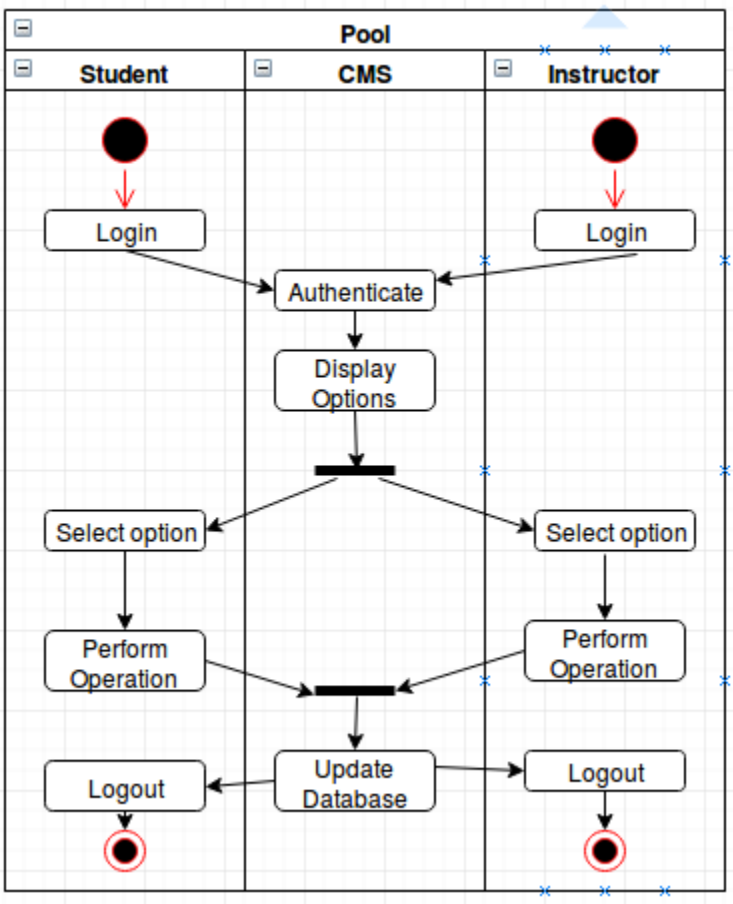

stu:Student
ui:UI
co:CourseOffering
c:Course
Login
Authenticate
View Courses
Enroll
return Available courses
return Success
Logout

State


Wait for login credentials input
Authenticate
Display navigation options
Wait for course details input
New Course created
Display Enrolled students
Display offered courses
Select courses
Course Withdrawn


Wait for login credentials input
Authenticate
Display navigation options
Display all available courses
Display courses taken
Select courses
Select courses
Enrolled for Course
Course Dropped

Swimlane



Appendix C: To Be Determined List