Software Design & Architectures

SOFE 3650 Case Study: Course Management System (CMS)
CRN: 43963

Prepared By:
     100617713   Samantha Husack
     100620049   Alexander Hurst

Due: December 5, 2018
Instructor: Professor Ramiro Liscano
Possible Marks: 60

Case Study: Course Management System (CMS)

Table of contents

1: Introduction

This report represents the ADD process for the Content Management System for a Course management system.

2: System Requirements

Requirement elicitation activities had previously been performed, and the following is a summary of the most relevant requirements collected. To view the entirety of the requirements for this business case, please refer to the Requirements Document here.

2.1 Use Case Model

The following use case model presents the most relevant use cases that support the CMS Model in the system. Other use cases are not shown.
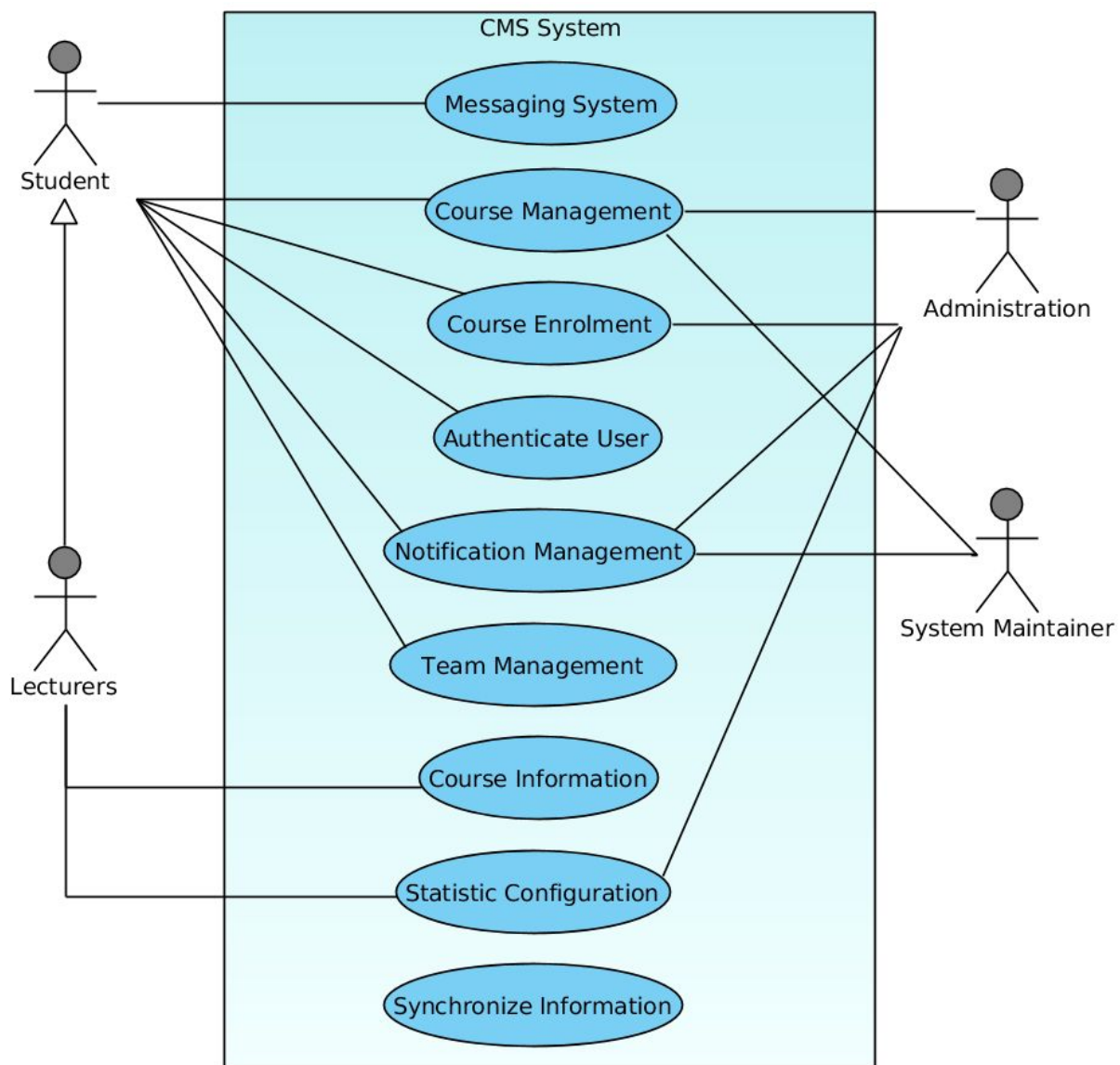

Figure 2.1   Use Case Model for CM System

Each of these use cases is described in the following table:

Use Case	Description
UC-1: Messaging System	The system can provide a messaging system for all users where: Students may send and receive messages to and from individuals, teams and all course participants including lecturer(s). Lecturers may send and receive messages to and from individuals, teams, other lecturers, and all course participants as well as post announcements.
UC-2: Course Management	The system shall allow for file management for all courses. Lecturers will have permissions to duplicate and import courses and materials, manage static information, create and recreate courses as well as register the assistant lecture and manage files. Students can search through and receive static and dynamic content. The system maintainer can specify content restrictions whereas the administrator may manage all courses and specify course requirements.
UC-3: Course Enrolment	The system shall allow course subscription by students, administration and enrolment policies shall be assigned by lecturers.
UC-4: Authenticate User	When a student, lecturer, administrator or maintainer would like to use the system, he/she needs an established session. If one is not in session, they are prompted to enter a user ID and password. Not all pages are available to all users.
UC-5: Notification Management	The system provides a notification manager for announcements, course updates and messages.
UC-6: Team Management	The system supports teams and teams management where lecturers can create and manage teams via the means of sending information, viewing, inserting and removing students as well assigning assistant lecturers. Students may join, create, leave, share to, upload, download from and message through teams as well.
UC-7: Course Information	The system shall provide, store, and represent static and dynamic information.
UC-8: Statistical Configuration	The system will allow statistical data to be drawn from the application records.
UC-9: Synchronize Information	The system can synchronize with secondary university systems.

2.2: Quality Attribute Scenarios

In addition to these use cases, a number of quality attribute scenarios were elicited and documented. The eight most relevant ones are presented in the following table. For each scenario, we also identify the associated use case(s) [AUC].

ID	Quality Attribute	Scenario	AUC
QA1	Security	A user can only make specified changes to the system based on their status and allowances. It is possible to know who performed the operations and when it was performed 100% of the time.	All
QA2	Availability	A failure occurs in the CMS during normal operation. The downtime is regulated to a maximum of 4 hours per month. All expected downtime has been announced a minimum of 48 hours in advance. Expected downtime occurs only during low-intensity hours.	All
QA3	Performance	Peak load performance even when usage is high.	All
QA4	Privacy, Availability, User Friendliness, Accurate	A student requests to see their grades. The grade given should be accurate to all of the data that the system has been provided and available only to that student.	UC-2
QA5	Compatibility, Usability	If a user wants to access information provided by a secondary university system, it will synchronise the content and display correctly.	UC-9
QA6	Durability	If the system crashes while a student enrols in a course when the system recovers the student should still be enrolled	UC-3
QA7	Simplicity	Lecturers should have the ability to send messages to all individuals, teams, etc with one click functionality	UC-1
QA8	Efficiency, Usability, Simplicity	The UI is consistent, descriptive and intuitive for all users, including the disabled. When a user wants to access any content it is less than 3 clicks away.	UC-7

2.3: Constraints

A set of constraints on the system and its implementation were collected. These are presented in the following table.

ID	Constraint
CON-1	A minimum of 1000 simultaneous users must be supported.
CON-2	The system must be accessed through a web browser (Chrome 70, Safari 12, Firefox 63, Internet Explorer 11, Opera 56) on different platforms: Windows, OSX, Linux.
CON-3	An existing relational database server must be used.
CON-4	The system shall provide an export to commonly used calendar formats
CON-5	The system shall be able to import BOZ roster information into the course Roster.
CON-6	The system shall have an intuitive, consistent and descriptive UI. The system shall be accessible by disabled (blind) users, who should be able to navigate the system and

	have access to all content and functionality.
CON-7	The system shall not allow users to change information which is contained and maintained by secondary university systems. The system shall automatically synchronize with secondary university systems.
CON-8	The system shall be easily testable, scalable, extensible, evolvable and maintainable.

2.4: Architectural Concerns

As this is a content management system under a greenfield development, only a few architectural concerns are identified initially, as shown in the following table.

ID	Concern
CRN-1	Establishing an overall initial system structure.
CRN-2	Leverage of the team's knowledge of various programming languages.
CRN-3	Allocate work to members of the development team.

3: Design Process

The following section is divided into four parts. First, we review the inputs. This is essential to identify the requirements chosen to be drivers in design decisions through the design process. Next, we iterate through three iterations of the ADD Process steps 2 through 7.

3.1: ADD Step 1: Review Inputs

In the first step of the ADD method, we review the inputs and identify which requirements will be considered as drivers. These inputs are summarised in the following table.

Category	Details
Design purpose	The purpose is to produce a sufficiently detailed design to support the construction of the system.
Primary functional requirements	From the uses cases presented in Assignment 2, the primary ones were determined to be: UC-2: Because it directly linked to the core features of the system. UC-5: Because of the technical issues associated with it. UC-6: Because it directly linked to the core features of the system. UC-9: Because of the technical aspects, as well as core functionality.
Quality attribute scenarios	The scenarios described in Assignment 2 have now been prioritized as follows:

	ID	Importance to CMS Stakeholders	Implementation Difficulty
	QA-1	HIGH	MEDIUM
	QA-2	MEDIUM	MEDIUM
	QA-3	HIGH	HIGH
	QA-4	MEDIUM	LOW
	QA-5	HIGH	HIGH
	QA-6	LOW	MEDIUM
	QA-7	HIGH	HIGH
	QA-8	HIGH	HIGH
	QA-9	HIGH	MEDIUM
	Only QA-1, QA-3, QA-5, QA-7 and QA-8 are selected as drivers		
Constraints	All of the constraints discussed in Assignment 2 are included as drivers.		
Architectural concerns	All of the architectural concerns discussed in the above section 2.4 are included as drivers.		

3.2: Iteration 1

The first iteration of the ADD design process aims to establish an overall system structure.

The entirety of Iteration 1 can be found under the ITERATION 1 Folder.

3.3: Iteration 2

The second iteration of the ADD design process aims to identify structures to support primary functionality.

The entirety of Iteration 2 can be found under the ITERATION 2 Folder.

3.4: Iteration 3

The third iteration of the ADD process aims to complete an final initial architecture structure for the Course Management System.

The entirety of Iteration 3 can be found under the ITERATION 3 Folder.