Iteration 1: CMS System

This section presents the results after main requirements have been defined and divided into smaller problems for future iterations. The steps of ADD are applied to establish a basic design for a CMS system.

Table of Contents

1.1 Step 1: Review Inputs

The requirements are carefully reviewed from the requirements, only the primary ones are chosen to represent use cases and constraints. These constraints and use cases are to provide the key quality attributes that the system needs. In Figure 1.1 a use case diagram is drawn representing all the main cases and the groups/actors that use them.

1.1.1 Use Cases

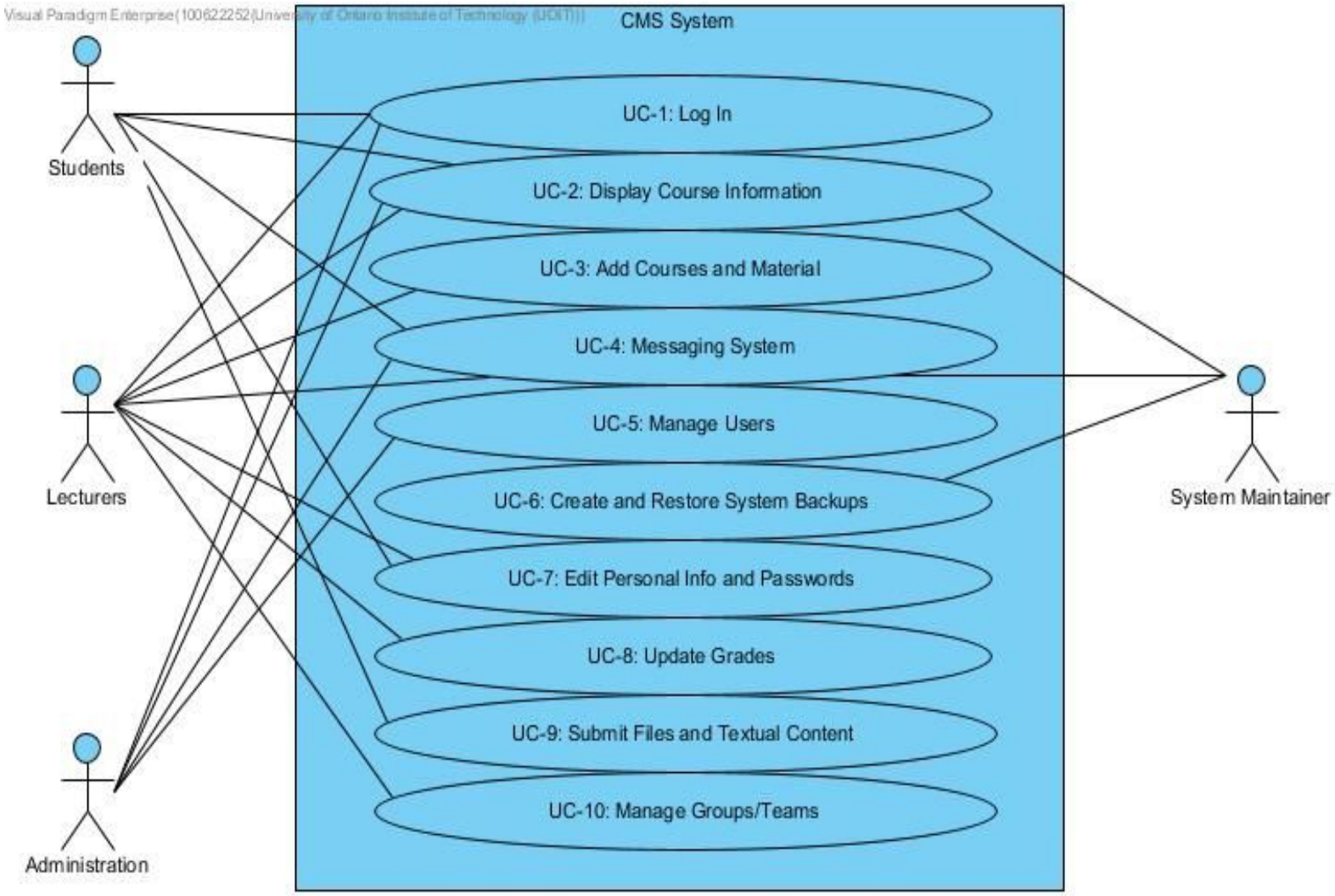


FIGURE 1.1

In Table 1.1, the use cases from Figure 1.1 are explained with a short description of the significance of the use case.

Use Case	Description
UC-1: Log in	A user logs into the system through a login/password screen. Upon successful login, the user is presented with different options according to their role.
UC-2: Display Course Information	Users should be able to view course information for courses they are enrolled in
UC-3: Add Course and Material	Lecturers should be able to upload course related material
UC-4: Messaging System	Students can message their lecturers and other students/teams. Lecturers can message everyone in their course, teams, and individuals
UC-5: Manage Users	Administration can assign users to courses (add and/or remove)
UC-6: Create and Restore Backups	System management is able to make backups and image copies of the entire system often. These backups can be restored when needed.
UC-7: Edit Personal Info and Passwords	Students and Lecturers can edit their personal information and change their login password
UC-8: Update Grades	Lecturers can change and update the grades given to students
UC-9: Submit Files and Textual Content	The user can submit a file or textual contents to their lecturer for a course for grading.
UC-10: Manage Groups/Teams	Lectures can manage groups made and allow students to join groups.

TABLE 1.1

1.1.2 Quality Attributes

In Table 1.2, a handful of quality attributes were extracted from the use cases and the requirement. In the table, it gives a short description of the scenario applying to the attribute, the use case associated with it, and a rating of how important it is to the system and how difficult it is to implement.

ID	Quality Attribute	Scenario	Use Case	Importance to User	Difficulty to

					Implement
QA-1	Security	The system shall be secure and only allow students to change the study information of others	UC-1, UC-7	High	Low
QA-2	Availability	Unexpected downtimes in the system are to be fixed immediately. The system will only be brought down for maintenance during low-intensity hours.	All	High	High
QA-3	Performance	Take a back up of the entire system before a maintenance is about to happen.	UC-6	Low	Medium
QA-4	Modifiability	A new course is added to the system by adding it to a database. It is immediately available for registration and use by lecturers and students	UC-3	Medium	Low
QA-5	Privacy	Users can only view their information. For example, students can only view their personal information and only see their grades for the course. Lecturers can only view content and students for their course(s).	UC-5	High	High
QA-6	Usability	Have an easy to use UI that is simple yet descriptive and consistent. A single login to access all the courses a user is taking.	All	HIgh	High

TABLE 1.2

1.1.3 Constraints

Table 1.3 displays all the constraints on the system

ID	Constraint
CON-1	The system shall have downtime at most 4 hours/month
CON-2	The system shall not allow users to change information which is contained and maintained by a secondary university system
CON-3	The systems shall only allow students, lecturers, faculty administration and maintenance to log in to the system
CON-4	The system must be accessed through a web browser in different platforms: Windows, OSX and Linux
CON-5	The System must have a consistent UI

TABLE 1.3

1.2 Step 2: Establish Iteration Goal

The iteration goal of the first iteration is to establish a structure for the system by focusing on and building on the primary inputs. The table below covers which inputs are of focus:

Category	Details
Design Purpose:	To create a design that supports the overall system structure.
Primary Functional Use Cases	UC-1: Because users need to be able to log in UC-2: Because it is one of the main functions UC-3: Because it is one of the main functions UC-6: Because it is a backup and security measure
Quality Attribute Scenarios	QA-1, QA-2, QA-5, QA-6 are selected as drivers
Constraints	All of them

TABLE 1.4 Main Drivers

1.3 Step 3: Choose Elements to Decompose

The system is a greenfield development system meaning that the system is based on existing architectures. This also means the entire CMS is an element that needs to be decomposed.

1.4 Step 4: Choose Design Concepts that Satisfy the Drivers

Below is a table which goes over different types of architecture references, data organization, and frameworks they system would need. The choices below help layout a path for the next iterations to build upon. The decisions below are explained as to why they are a prime option as well as why other options have been discarded.

Design Decisions	**Rationale**
Web Application	This reference architecture is oriented towards the development of applications that are accessed from a web browser. The use cases that are being introduced in the system require interaction through a web browser(CON-4) **Discarded Alternative:** ● Rich Client Application- supports a strong user interface and user experience however it is only meant for PC applications which break CON-4. This also would require internet support as the messaging, course information, and the material is not local and consistently updating. ● Rich Internet Application- Does not have strong security features which affect its access to storage data. It does support a rich user interface but it makes server-side tasks difficult to implement. ● Mobile Applications- Supports portability and accessibility but it is not considered as an option to access the system and poses a risk of having an inconsistent design in order to fit the screen size.
SIS Framework	The SIS Integration Framework is a Building Block-extensible framework that provides common functionality to all integrations while facilitating integration creation, configuration, and management. The SIS Framework provides the UI-based create, configure, and operation/maintenance capabilities as well as the Integration Types (based on Building Blocks) to provide the logic specific to each SIS integration type **Discarded Alternatives** ● ASP.NET- server-side web application framework designed for web development to produce dynamic web pages. Was considered but discarded because it's only operable on Windows ● React Native-discarded because a mobile application is not necessary
Hierarchical	The Hierarchical database allows for a tree-like system which

Database	makes accessing the database and modifying it simple and easy to use by the administration who design the courses. This works since courses are under a certain year which is under a certain program which is under the university. This supports QA-4 and QA-5 since the administration team is responsible for modifying courses and allowing access to them. This also would support all the users who are allowed in the system as their information would also be in a database. This means there is support for QA-1, CON-1, CON-2, and CON-3 because only those in the database can log in to the system and see the courses they are registered for. Also, the database would only be accessible to certain people for specific items. The information is on a need to know bases.
3-Tier Deployment	Since the system must be accessed from a web browser(CON-4) and an existing database server must also be used, a three-tier deployment is appropriate.

TABLE 1.5 Concepts to be refined

1.5 Step 5: Instantiate Architectural Elements and Define Interfaces

At this moment there are no interfaces to be defined. In Table 1.6, an architectural element is defined below.

Design Decision and Location	Rationale
Host the database in a separate server	This choice secures the information and allows only the administration to modify it. (CON-2)

TABLE 1.6

1.6 Step 6: Sketch Views and Record Design Decisions

The diagram below displays a sketch of the architecture reference chosen from the design decision. In the diagrams, the system is seen being adapted to the architectures chosen below.

A 3-Tier Layered diagram is used to model each of the components and show the interaction between them. This diagram follows the web-application architecture interface.

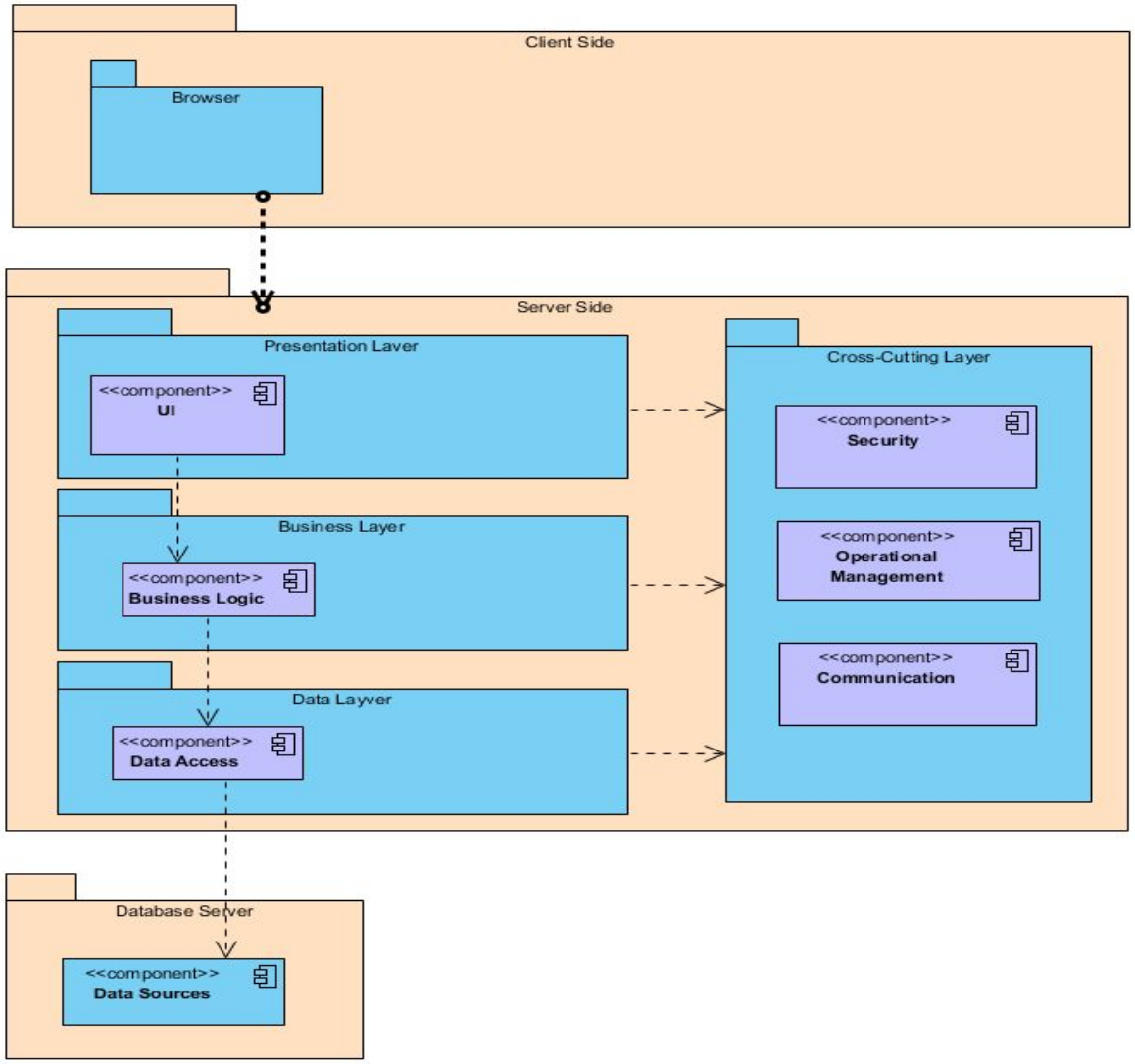


FIGURE 1.2 Three-Tier layered architecture of a Course Management System

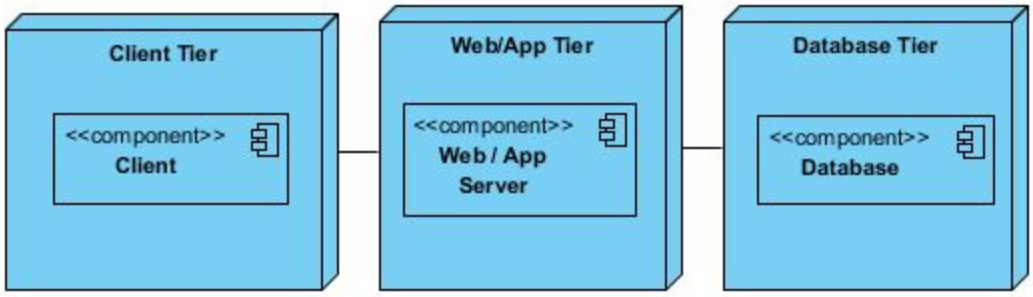
The following table below gives a short description of the main functionalities of each element.

Element	Responsibility
Browser (Client Side)	A web browser running on the client's machine (CON-4)
User Interface (Server Side)	These components receive user interaction and present information to the user. Includes buttons, text fields, links, containers, and notifications. This layer is significant in achieving a consistent and functional UI (UC-2, QA-6, CON-5)
Business Logic (Server Side)	This element performs business logic operations such as adding, deleting, and editing courses and material
Security (Server Side)	These components handle security aspects such as

	authorization and authentication
Operational Management (Server Side)	This component includes management or exception handling, logging, and instrumentation validation
Communications (Server Side)	This component is responsible for communication between the client side and server side.
Data Access (Server Side)	This element is responsible for handling connections to the database. It uses the operations provided from the business logic component to perform queries and mapping of databases specific to hierarchical databases.
Data Sources (Database)	This element is responsible for containing the information provided by the Data Access element and storing the information in a hierarchical structure.

TABLE 1.7 Functionalities of Elements

The deployment diagram in Figure 1.3 sketches an allocation view that illustrates where the components associated with the modules in the previous diagram will be developed.



Element	Responsibility
Client Tier	Hosts the client's side logic of the application on the user's browser
Web/ App Tier	Hosts the servers side logic of the application and also serves web pages and content.
Database Tier	Holds all the information

TABLE 1.8

1.7 Step 7: Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

The following Kanban board summarizes the design progress made after the first iteration

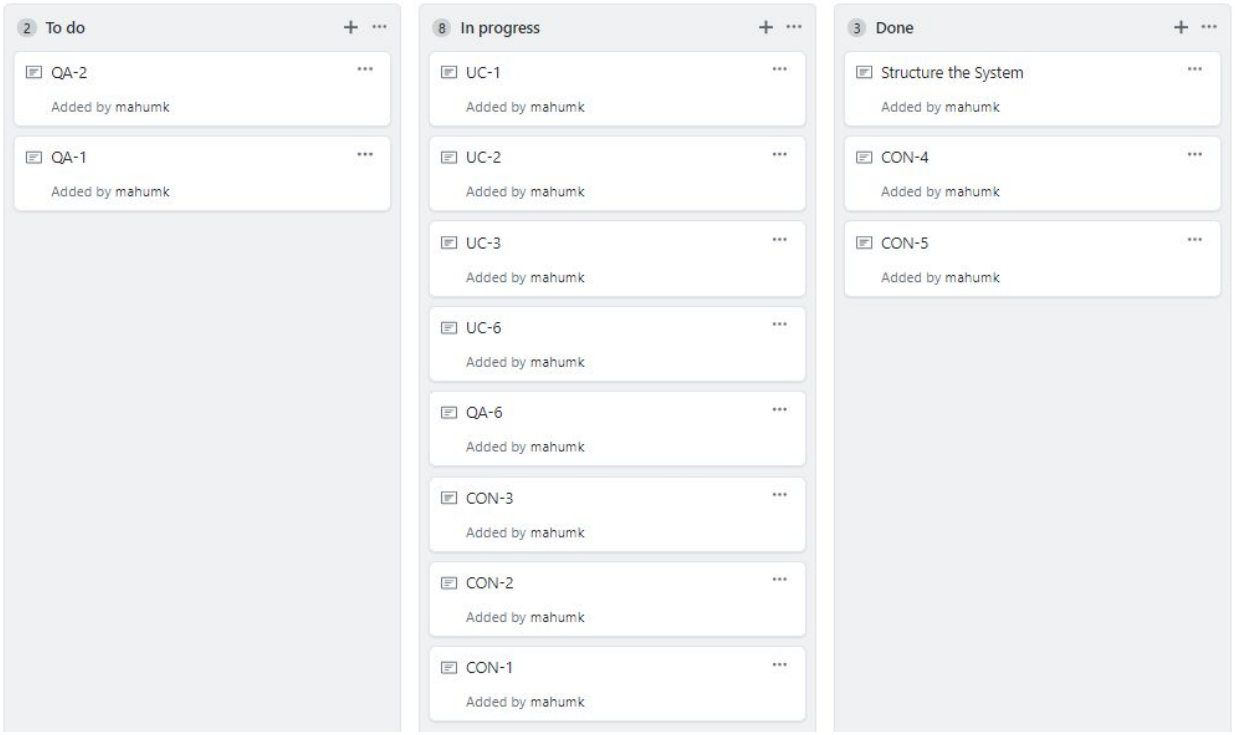


FIGURE 1.4 Kanban board after iteration 1