



COUSE MANAGEMENT SYSTEM

Iteration 1: Establishing an Overall System Structure



MOHAMMED HAMEEDUDDIN - 100587893
ABRAHAM ISAAC DURAN - 100731964

The Design Process

Step 1: Review Inputs

The primary use cases chosen for our architecture are:

UC 2: Course Selection

UC 3: Submissions

UC 7: Course Management

UC 10: Editing Dynamic Course Information

UC 11: Editing Static Course Information.

The quality attributes have also been characterized in user importance and architect difficulty for implementation:

ID	Quality Attribute	User Importance – Architect Difficulty
QA-1	Performance	H – M
QA-2	Modifiability	H – H
QA-3	Security	H – M
QA-4	Availability	H – M
QA-5	Usability	M – L
QA-6	Performance, Usability	M – M
QA-7	User Availability	H – H
QA-8	Extensibility	M – H

Iteration 1: Establishing an Overall System Structure

Step 2: Establish Iteration Goal by Selecting Drivers

For this iteration the main goal is to establish an overall system structure while keeping the design drivers in mind that will have an influence on the overall architecture.

The main drivers selected are:

QA-1: Performance

QA-2: Modifiability

QA-3: Security

QA-4: Availability

QA-7: User Availability

CON-1: Students cannot view other student's personal information or grades.

CON-2: The system can only have a max of 4 hours'/month downtime during low intensity hours.

CON-3: The system must be accessible to disabled users who should be able to access full system content.

CON-4: The system shall be interoperable with secondary university systems.

CON-5: The system shall be scalable as more data will be uploaded each year.

Step 3: Choose One or More Elements of the System to Refine

We are working on the design of a greenfield system for a domain that is novel. We will be establishing a roadmap and working to refine structures to address driver concerns. The entire Course Management System will be refined.

Step 4: Choose One or More Design Concepts that Satisfy the Selected Drivers

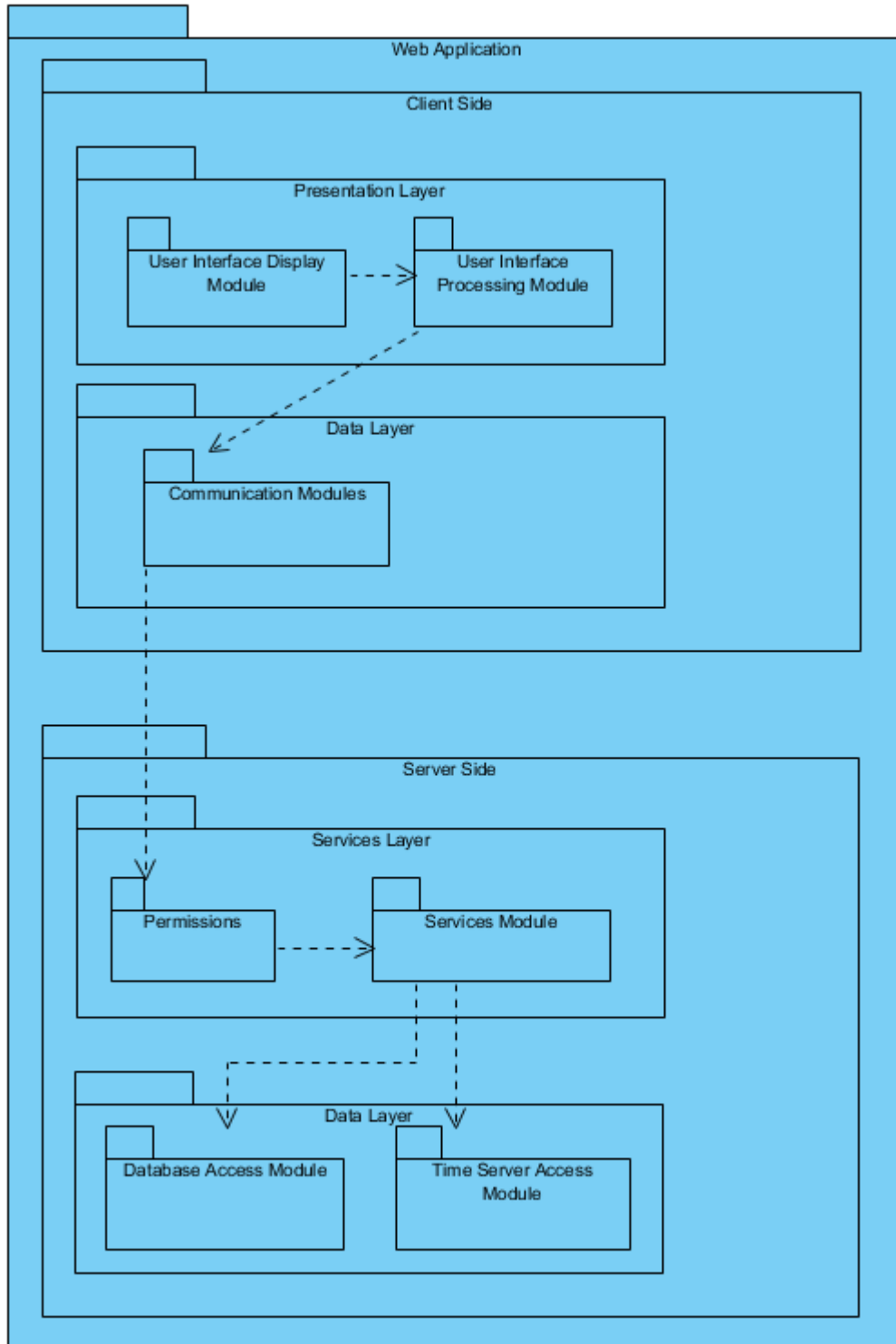
Design Decisions	Rationale
Structure the client-side part of the system using Rich Internet Applications reference architecture	Rich Internet Application (RIAs) typically run inside a browser and may be developed using code that is executed by the browser. The user would be logging into the system through a web browser and access the course system. Therefore we would require a rich internet application to create a rich user interface.
Structure the server part of the system using the Service Application reference architecture	Service applications are non-interactive applications that expose functionality through public interfaces. Services will be invoked by service consumer components remotely.
Physically structure the application using the three-tier deployment pattern	This system will be accessed through a web browser and it will be pulling/pushing data from/to and from existing database through the application. Therefore, a three tier deployment pattern seems best for this project.

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The user interface will have 2 modules that handle the display and the user requests. There will also be a dedicated time server as submissions and uploads to the system by the user are time critical. This time server will serve as the unified clock to the services module handling the user requests and database transactions.

Step 6: Sketch Views and Record Design Decisions

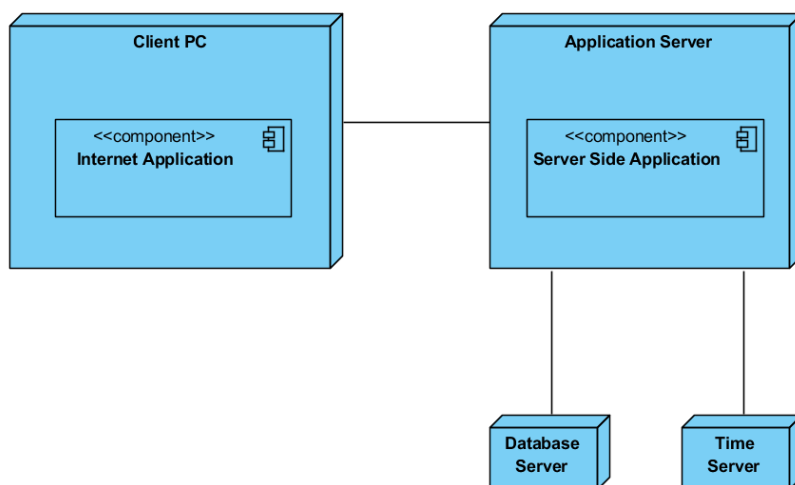
The flow of how our system will operate is shown in the diagram below:



Each layer in the diagram contains modules that have specific functions. This is an early design with few details that will be later defined.

Element	Responsibility
Presentation Layer	The presentation layer contains the user interface display module where the program screen is displayed to the user. It also houses the user interface processing module that handles the user interaction requests.
Data Layer Client Side	This layer contains a communication module that handles the requests with the server side of the application.
Services Layer	The permission module will interpret the request and determine if the user has account authorization to proceed. A services module exists here that translates the requests from the client side to retrieve or push data to the database.
Data Layer	The data layer holds the database access module that will contain the database system for information storage. The time server access module is also in this layer that will give the accurate time to all application systems.

The deployment diagram below shows the allocation view where the components associated with the modules in the previous diagram will be deployed.



The responsibilities of the components in the deployment diagram are as follows:

Element	Responsibility
Client PC	The computer or laptop that the end user uses to access the web based application.
Application Server	The server side system that process the query's and handles the information sent to and retrieved form the database.
Database Server	The server that hosts the database.
Time Server	The server that will provide accurate time information for the application.

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

The following table summarizes the design process of this iteration using the Kanban board technique.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During Iteration
	UC-2		System architecture design contains modules that supports these requirements.
	UC-3		
	UC-7		
	UC-10		
	UC-11		
QA-1			No particular decisions were made to address this requirement.
	QA-2		Data and application separation allow for changes to be made without affecting the rest of the system.
		QA-3	A permissions module was placed in the server side which will only allow users with authorization to access certain features/data.
QA-4			No particular decisions were made to address this requirement.
QA-7			No particular decisions were made to address this requirement.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During Iteration
		CON-1	A permissions module was placed in the server side which will only allow users with authorization to access certain features/data.
CON-2			No particular decisions were made to address this requirement.
CON-3			No particular decisions were made to address this requirement.
	CON-4		With data and application services separated, third-party applications can be integrated to work with the course management system.
		CON-5	A database server is being implemented and can have more storage added to it to increase storage capacity when needed.