Iteration 1: Establishing an Overall System Structure

ADD step 1: review Inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design Purpose</td><td>This is a greenfield system from a mature domain. The purpose is to produce a sufficiently detailed design to support the construction of the system</td></tr>
<tr><td>Primary functional requirements</td><td>UC1: Manage Courses
UC7: Calculate grade statistics
UC8: Create and Restore Backup
UC10: Retrieve Course Information
UC11: Subscribe/Unsubscribe to courses
UC13: Share files and messages with team
UC25: Email students</td></tr>
<tr><td>Quality Attribute scenarios</td><td>
<table>
<tr><th>Scenario ID</th><th>Importance to the Customer</th><th>Difficulty of implementation according to architect</th></tr>
<tr><td>QA1</td><td>High</td><td>High</td></tr>
<tr><td>QA2</td><td>High</td><td>Medium</td></tr>
<tr><td>QA3</td><td>Low</td><td>Low</td></tr>
<tr><td>QA4</td><td>Low</td><td>Low</td></tr>
<tr><td>QA5</td><td>High</td><td>High</td></tr>
<tr><td>QA6</td><td>Low</td><td>High</td></tr>
<tr><td>QA7</td><td>Medium</td><td>High</td></tr>
<tr><td>QA8</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA9</td><td>Medium</td><td>Medium</td></tr>
</table>
</td></tr>
</table>

	<table><tr><td>QA10</td><td>High</td><td>Medium</td></tr></table> From this list, only QA1, QA2, QA5, QA10 are selected as drivers.
Constraints	CON1, CON2, CON4 are selected as the drivers
Architectural concerns	CRN1 Establishing an overall initial system structure. CRN2 Leverage the team's knowledge about Object Oriented Programming Languages and scripting language like Node JS CRN3 Allocate work to members of the development team. ALL of the architectural concerns are included as the drivers.

Step 2: establish iteration goal by selecting Drivers

The goal of the first iteration is to establish an overall system structure

- QA 1 - Privacy
- QA 2 - Availability
- QA 5 - Security
- QA 10 - Maintainability
- CON 1 - System must be accessible over different web browsers on different platforms
- CON 2 - A minimum of 200 simultaneous users must be supported
- CON 4 - All course information since the start must be stored
- CRN1- Establishing an overall initial system structure.
- CRN2- Leverage the team's knowledge about Object Oriented Programming Languages and scripting language like Node JS
- CRN3 -Allocate work to members of the development team.

Step 3: Choose One or More elements of the system to refine

The Course Management System (CMS) will be a greenfield system in a mature domain. Meaning it is a new system based on existing architecture patterns and styles. As it is a greenfield system, the element to refine is the entire CMS. Refinement will be carried out through decomposition.

Step 4: Choose One or More Design Concepts that satisfy the selected Drivers

Design Decision	Rational
Web Application reference architecture	This reference architecture is orientated towards the development of applications that are accessed from a web browser. This architecture is helpful in achieving QA-10 as the application can be easily maintained in the back-end of the server by the admin. This architecture also allows connection to a database (UC-10).

Deployment Pattern	Rational
3-Tier	Since system must be accessed from a web browser and an existing database server must also be in use (CON-4), a three-tier deployment is ideal with a presentation layer, business logic layer and a database layer. Dividing the application into distinct layers help improve maintainability of the system (QA-10).

Architectural Design	Rational
Database Access	We need to insulate applications from the details of how data is represented in persistent storage. Introduce a data mapper for each type of persistent application object. The responsibility of this mapper is to transfer

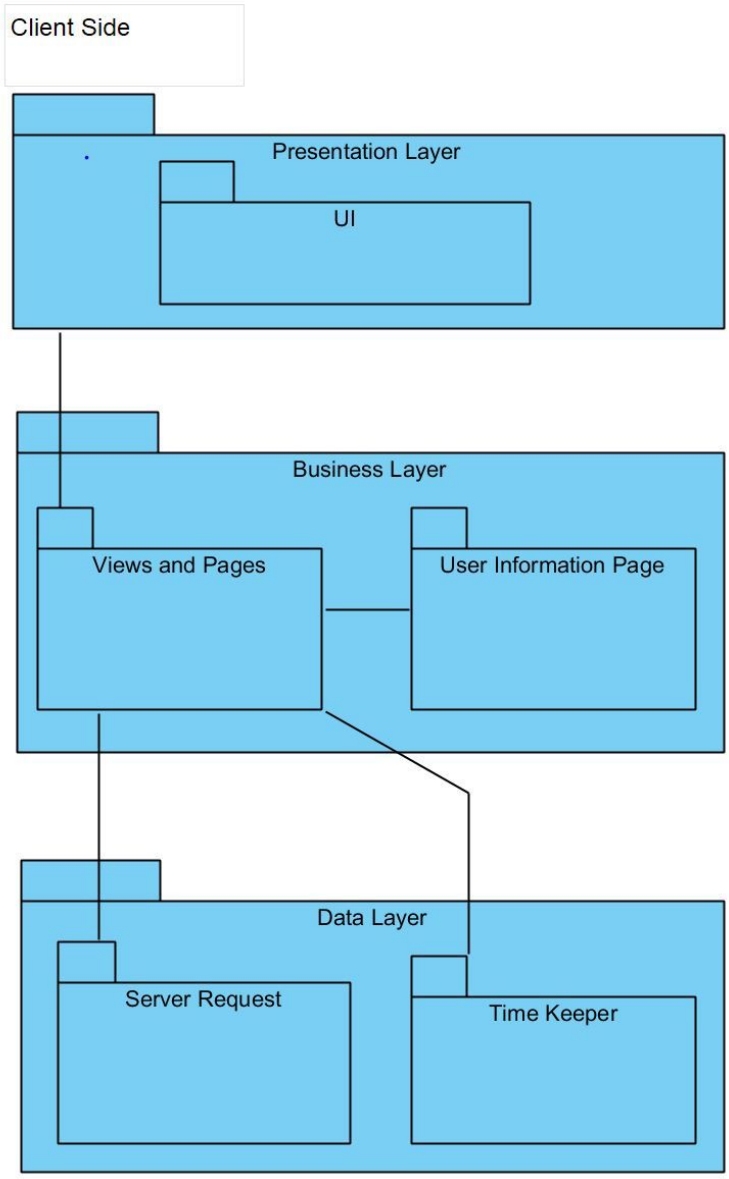
	data from the objects to the database, and vice versa.

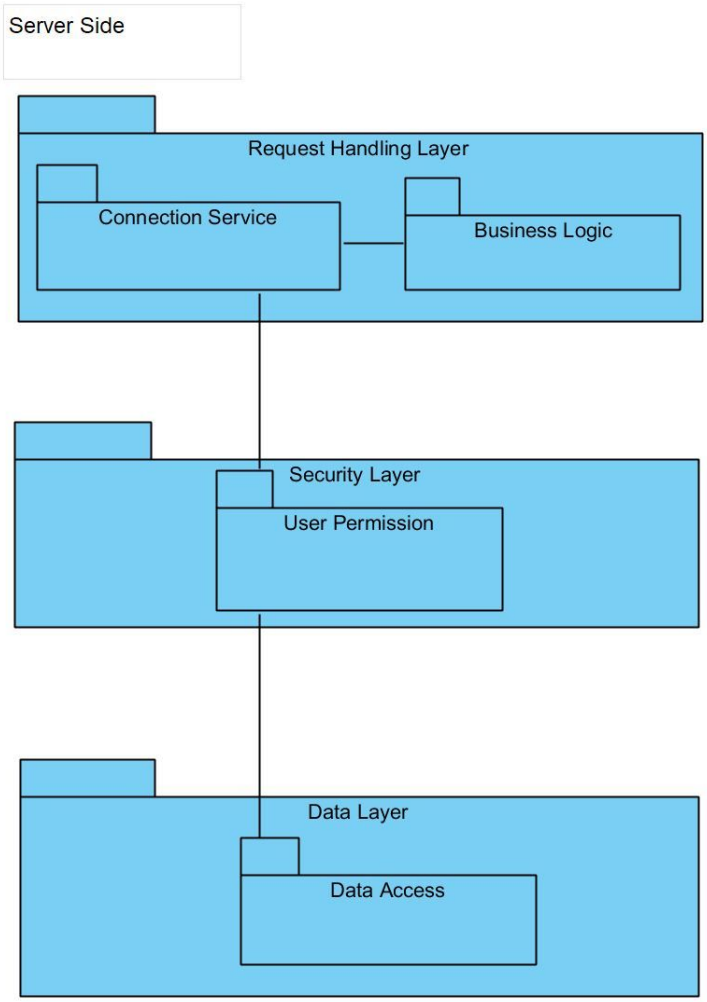
Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

As this is the first iteration, it is too early to precisely define functionality and interfaces.

Design Decision and Location	Rationale
Remove local data storage from the client side	-Network application is assumed to be reliable therefore there is no need for local data storage - Communication to the server is handled by the communication components and thus any internal communication is done using method calls and variables stored in cache - Full backups of application is supported (UC8) - Supports simpler maintenance of system (QA10)
Create a module for timekeeping in the data layer on the client side	- Aids in syncing of timings with the server - Supports UC13 by providing support for real-time messaging capabilities among users
Isolate important business logic onto server side	- Business layer on client side should contain simple logic that processes user input - Protects user information and ensures system security as server-side logic is harder to tamper (QA1,QA5,UC7) -Provides avenue for extensibility and evolvability

Step 6: Sketch Views and Record Design Decisions





Element	Responsibility
Presentation Layer (client)	This layer contains UI modules that control user interaction
Business Layer (client)	This layer contains modules that perform business logic operations on the client side
Data layer (client)	This layer contains data
UI modules (client)	These modules render the user interface and receive user inputs.
Views and Pages modules (client)	These modules handle the business logics of page viewing function
User information page module (client)	This module handles the business logics of retrieving user information and data.

Server Request (client)	This module handles the http request to retrieve/update data.
Time Keeper (client)	This module is responsible for communication with the time servers.
Business Logic (server)	This layer manages calls from Views and communicates/request information from the database.
Connection Service (server)	This module receives the connection from the business logic layer.
User Permission (server)	This module carries the permissions each user has to access stored information in the database.
Data Access (server)	This layer is for stored information and will return to previous modules and layers it was called from.

Step 7: Perform Analysis of current Design and review iteration Goal and Achievement of Design Purpose

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
UC1			No relevant decisions made
	UC7		Management of grades is kept secure by logic isolation
		UC8	By removing local data, admin can manage backups of the system
	UC10		Architecture supports a database to retrieve

			information
UC11			No relevant decision.
	UC13		Timekeeping module allows for users to keep communication in real-time
UC25			No relevant decision.
	QA-1		User information is protected from exposure by business logic isolation on the server side
QA-2			No relevant decision
	QA-5		Business logic isolation hampers tampering of data by users
	QA-10		Tiered architecture and removing local data allows for easy maintenance of the system
CON-1			No relevant decision.
CON-2			No relevant decision.
	CON-4		Database access allows for storage of information
		CRN-1	Selection of application architectures and patterns
CRN-2			No relevant decision.
CRN-3			No relevant decision.