

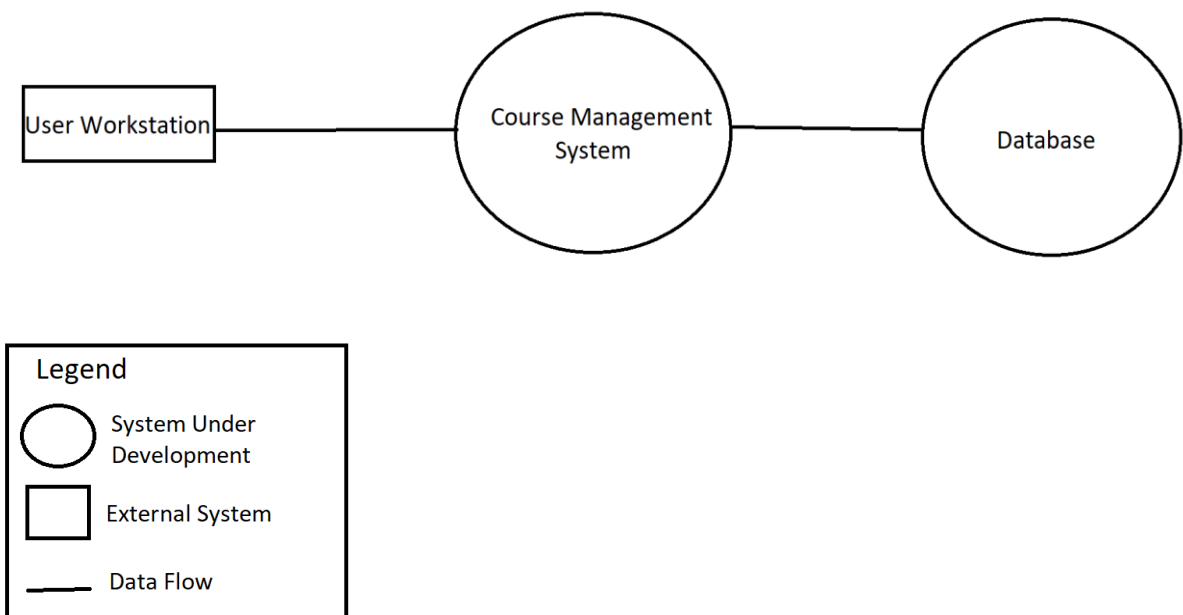
Iteration 1: Establishing a System Structure

In this iteration we will mainly be establishing all the concern that are related to the system. After reviewing the Quality attributes some striking attributes that stand out are the following.

Step 2 Establish Iteration Goal by Selecting Drivers

- QA-2 Availability
- QA-4 Useability
- QA-10 Scalability
- CON-2 Multiple simultaneous connection is required, size > 300
- CON-5 A large relational database is required to maintain student progression and enrollment requirement
- CRN-1 Organizing information without nesting two tables into one.

Context Diagram



Step 4 Choosing one or more Elements to refine

The two elements needed to be refined will be the course management system and the database since they are the only two systems under development.

Design Decisions and Location	Rationale
Build the server using Amazon Web Services Cloud Architecture.	One of our concerns CRN-2 states that multiple simultaneous connection is

	<p>required, size > 300, in order to do this we will need a powerful web hosting service that has a good reputation with handling large amounts of traffic. AWS is a highly scalable type of cloud web hosting architecture which deals with quality attributes that we deemed important. QA-10 has high difficulty of implementation and deals with scalability. Cloud architecture will ensure that the web hosting will not be a detriment to the architects since it is a highly scalable cloud service that can handle high traffic for large and small applications. AWS has robust security measures which is a Quality attribute that is of high importance to the customers. This web hosting application is very reliable and is used by many large companies which helps deal with Quality attribute 2 (the system should be running 24/7) which was deemed a high priority and importance for both the architects and users. Cloud web architecture uses data splitting measure to make sure that users cannot have their data shared. This system does not require a lot of computation which is one of cloud architectures main benefits however with this type of architecture the user can choose as much and as little computing resources as they like.</p>
Structure the Database using Model View Controller architecture.	<p>MVC seems the most appropriate and most used for the implementation of the database. The views which would contain course information instructor information etc. would be part of the top layer of the architecture. The middle layer would consist of the controller that can transform the data and is a gateway for changing the way the users see the data from the view. The inner layer is the model which is where the data is stored in the database. This architecture is the most appropriate because of its high maintainability and testability. The architecture can isolate layers so that they aren't affected by testing</p>

	and maintenance. This is one of the most important points of the system. It must be running most of the time with as little downtime as possible, and it must be scalable (QA-4 and QA-10).
Build the user interface using standard languages such as HTML CSS Javascript	These languages are the easiest to pick up, they are sufficient for the job. The system does not need any overcomplicated fancy UI. The main focus of the UI is to be simple and easy to use. These languages meet this requirement and most developers are familiar with them.

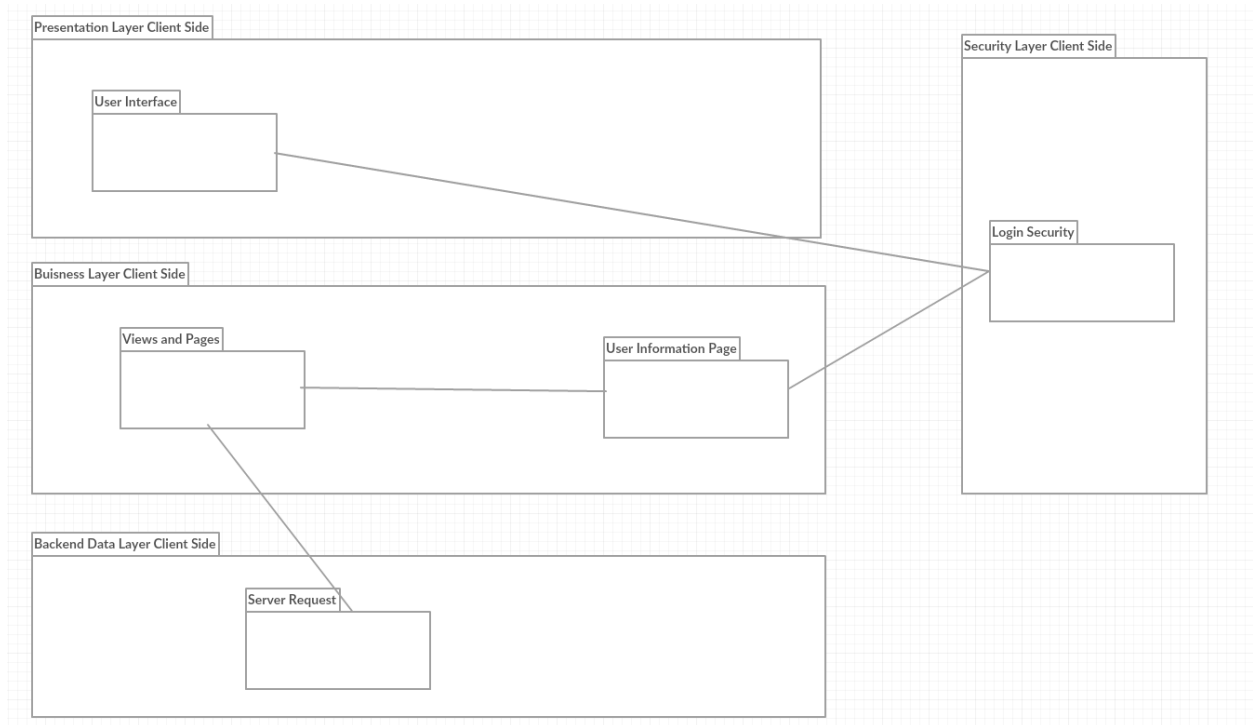
Step 5 Instantiate Architectural Elements, Allocate Responsibilities, and define interfaces

Below we will instantiate the two elements in the system.

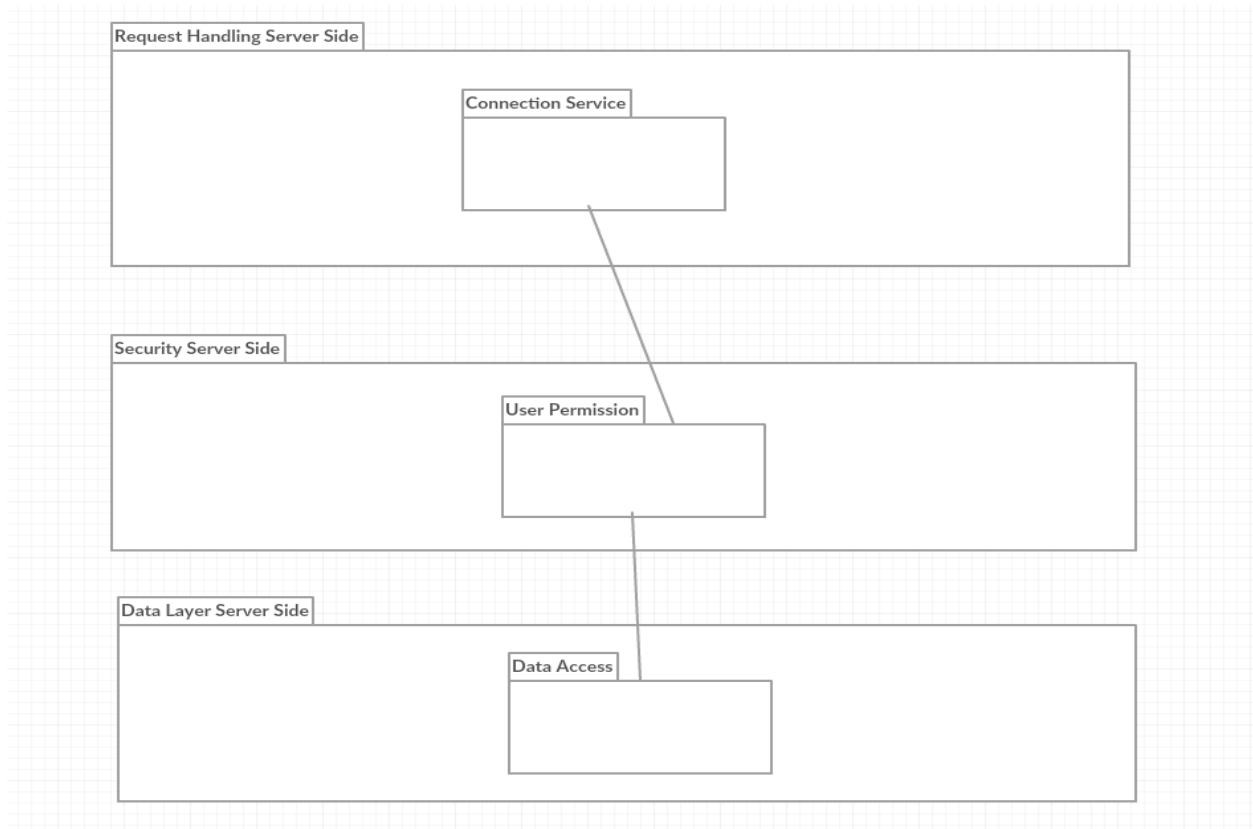
Design Decisions and Location	Rationale
Create cloud server without UI with basic UI pages	The UI will be simple but there is still not enough planning to decide what the UI will have. We know the bare essentials such as a login screen, homepage, and course pages. However we do not have any details so for this iteration these will be the only parts instantiated.
Create data types and tables for the model layer of the database architecture	We have a rough idea on what data will be used in the tables of our database as well as the primary keys. We will instantiate the database by filling some of those tables with our key data types and primary keys.

Step 6 Sketch Views

Client Side



Server Side



Element	Responsibility
User Interface	This layer deals with user input and output so the user can communicate with the web server
Login Security	This layer deals with security and identification of the user so the web client can determine privileges
User Information Page	This module deals with the personal information of users
Views and Pages	This module deals with the various views the user will access in order to get information
Server Communication	This layer deals is called from the view and is needed to communicate and request information from the database
Server Side Connection Service	This module receives the connection from the server communication layer.
User Permission	This module identifies what access the user has to the information stored in the database
Data Access	This layer is the location of stored information and will return to the previous modules and layers it was called from.

Responsibilities of Elements

Element	Responsibility
User Workstation	Users PC allows user to use the application interface to request or input information.
Application Server	Web pages and server that provide a medium for the user to view data and access data from the database
Database Server	Relational database that stores tables of data

Step 7: Perform Analysis of Current Design and Review Iteration

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
	UC-1		In our client side view our user has a user interface and that connects to a security which validates the users identity. Furthermore they can be connected to their own personal information page
	UC-2		In the client side view multiple web pages are accessed after login. This includes an add/drop course.
	UC-3		In the client side view multiple web pages are accessed after login. This includes viewing course information page
UC-14			No decisions have been made yet.
	QA-2		As stated in iteration 1 a cloud service for the web server will be used which will keep downtime to a mininum.
	QA-4		The UI will be simple and this will be communicated to the development team. The languages used are basic languages and ease of usage is emphasized.

	QA-10		As stated in iteration 1 a cloud service for the web server will be used which will allow us to add processing units that suit our needs. This allows us to scale up or down depending on what we want and the demand of our system.
CON-1			No decisions have been made yet.
	CON-2		The use of a cloud server helps fulfill this concern.
CON-3			No decisions have been made yet.
	CON-4		With a 3 tier database we can add massive amounts of information and testing can be done without being a detriment to the users since it won't affect downtime. The layers can be separated so the testing can be done on one layer without interrupting the other.
	CON-5		With a 3 tier database we can add massive amounts of information and testing can be done without being a detriment to the users since it won't affect downtime. The layers can be separated so the testing can be done on one layer

			without interrupting the other.
	CRN-1		By properly identifying primary and foreign keys we can eliminated any conflicts within the tables in the database
	CRN-2		By properly identifying primary and foreign keys we can eliminated any conflicts within the tables in the database
	CRN-3		Languages have been considered and have been taken into account with respect to the knowledge of the developers.