Iteration 3: Addressing Quality Attribute Scenario Driver

QA-2

Step 2: Establish Iteration Goal by Selecting Drivers
For this scenario the main focus is on QA-2: The system should work 24/7 without any error.
During the expected downtime, maximum 4 hours/month.

Step 3: Choose one or more Elements in the System to Refine
For this scenario the elements that are in need of refinement are the Database Server and
Application Server.

**Step 4: Choose one or more Design Concepts that Satisfy the Selected Drivers**

Design Decisions and Location	Rationale and Assumptions
Regularly make backups of the Server Database	In the event of a hardware crash or malfunction the server data is the most precious information and having a backed up server is the best way to prevent data loss.
Deploy a load balancer to increase capacity and reliability of the server.	If one application server becomes over capacity the load balancer can combat this by directing the traffic to an available application server, assuming the system adapt to a multiple application server system. This load balancing process can be done with a variety of algorithms such as round robin, least bandwidth etc.
Implement a server monitoring tool	The use of a server monitoring tool is beneficial for quickly identifying problems and notifying the maintenance. With server monitoring you are able to monitor traffic, unusual amounts of bandwidth being used. In addition to that server monitoring can track processor and memory usage. This can help predict and manage any minor problems before they become larger. Small problems being detected and isolated early is the best way to decrease downtime to a small amount per month.

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

Design Decisions and Location	Rationale
Implement Software Element to backup server	When the server is constantly backed up if any major issues occur the backup server image can be used to restore any lost or damaged information. This helps satisfy QA-6 and QA-2
Deploy a Load Balancer	Load balancer is deployed before server. Multiple application servers are created and load balancer is deployed to regulate traffic before traffic gets to the server. This helps satisfy QA-2 and CON-2/
Server monitoring software is implemented	A server monitoring software keeps track of the server in case any small issue occurs which can further help prevent any large issue. This helps satisfy QA-2 by potentially decreasing downtime.

Step 6: Sketch Views and Record Design Decisions
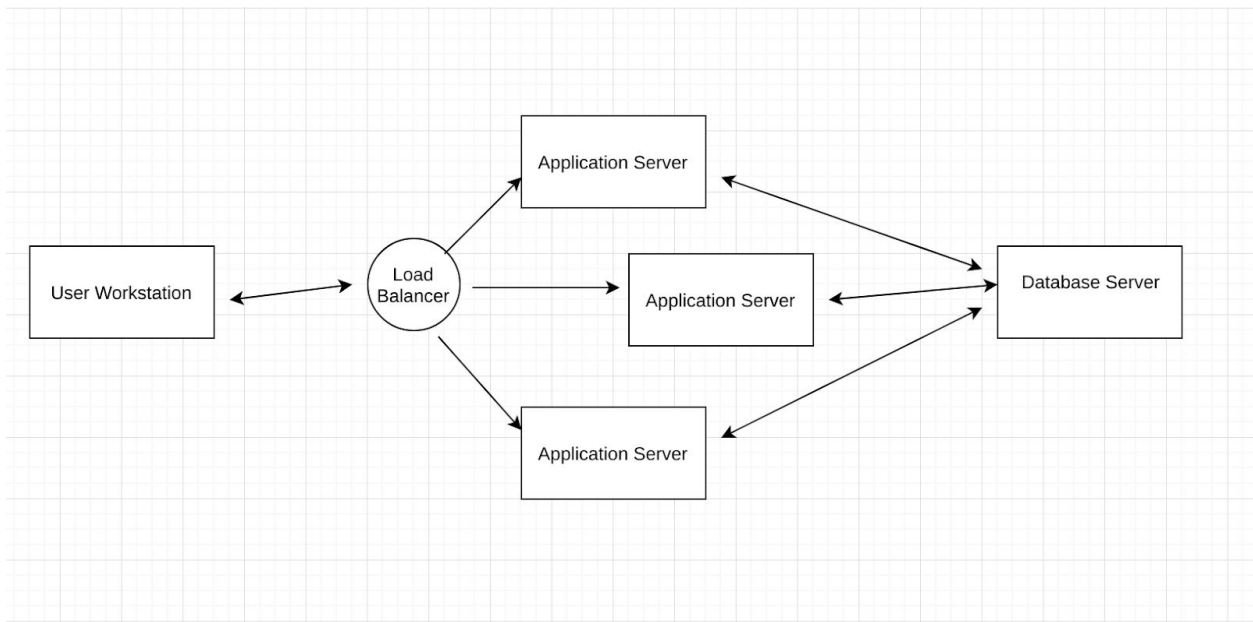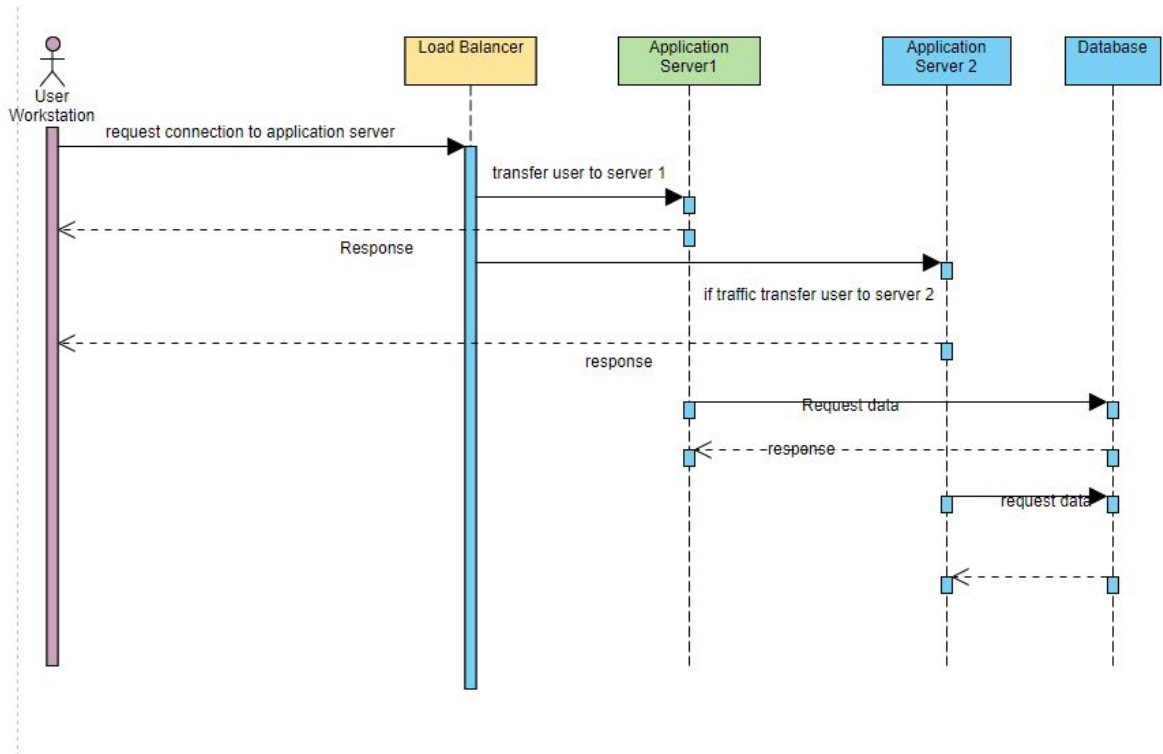
Step 7: Perform Analysis of Current Design and Review Iteration
Completly addressed elements have been removed

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
	UC-4, UC-6		The discussion and team service provides basis for allowing discussion board, but the actual mechanism has yet to be determined
	UC-5		The course management service provides functionality to perform file upload and organization.
	QA-1		The security manager determines user login and assign privilege based on user type
	QA-2		As stated in iteration 1 a cloud service for the web server will be used which will keep downtime to a minimum. Load balancer, server monitoring and server backup help keep server downtime to a

			minimum.
	QA-6		The extended relational database system provides good scalability and extensibility as well as interoperability so extraction of the data from the database becomes manageable
	QA-8		Using of DBMS provides high maintainability to the database. Constant server monitoring algorithm helps predict and prevent problems.
	CON-2		The use of a cloud server helps fulfill this concern. As well as implementation of Load balancer.
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		The basic relational database is designed and further improvement is required
	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.
	CRN-4		With the database structure and basic architecture been designed, the work can be distributed to complete individual part of the system.

Summary

The course management system was requested for implementation and though on the outside seems like a simple task, meeting stakeholder expectations and avoiding concerns turns out to be a difficult process. This system used bare components and combined all those. Each iteration took components and refined them by looking at driving use cases, quality attributes and concerns. By refining the system through multiple iterations most of these requirements were addressed to form a functioning system.