Iteration 3: Addressing Quality Attribute Scenario Driver (QA-2)

In this iteration of the ADD process, the aim is to address quality attribute driver QA - 2.

ADD Step 2: Establish Iteration Goal by Selecting Drivers

The main focus for this scenario is QA-2: the system should work 24 hours a day without any errors.

ADD Step 3: Choose one or more Elements in the System to Refine

In this scenario of availability, the refined elements are the physical nodes identified during the first iteration which are Application server and Database server.

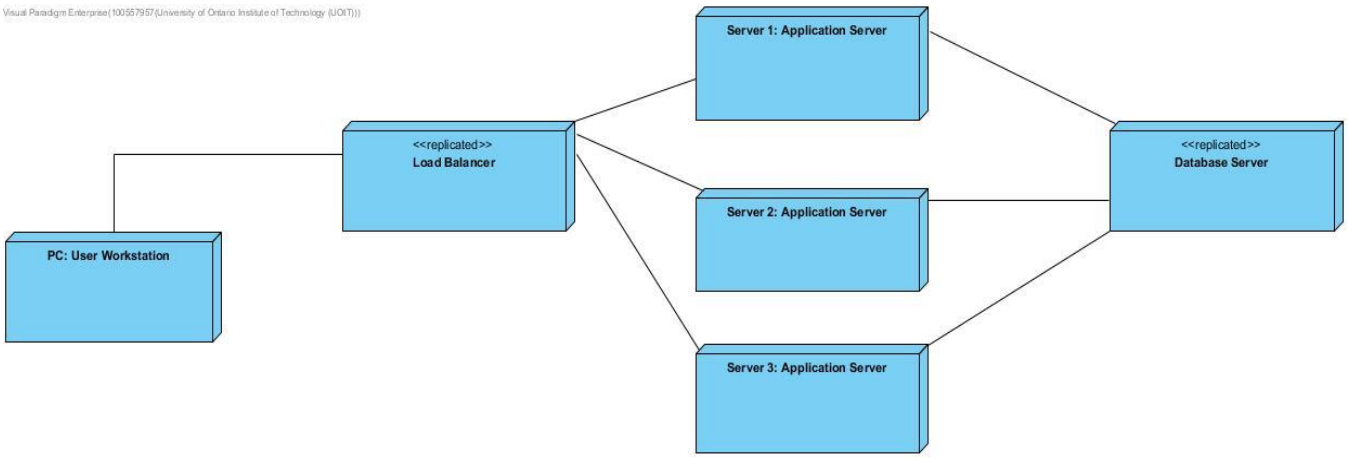
ADD Step 4: Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Location	Rationale and Assumptions
The server database needs to be backup all the time	In case of a hardware crash or malfunction, server data is the most valuable information and the best way to avoid data loss is to have a backed - up server.
Use a load balancer to improve the server's capacity and reliability	A load balancer is a reverse proxy that distributes network or application traffic over a number of servers. Load balancers are used to increase the capacity (competitive users) and application reliability.
Implement a monitoring tool for the server	Server monitoring software is an essential tool for system administrators because it enables automated reporting, scheduled checks and preventive health warnings of your many servers in your operating environment.

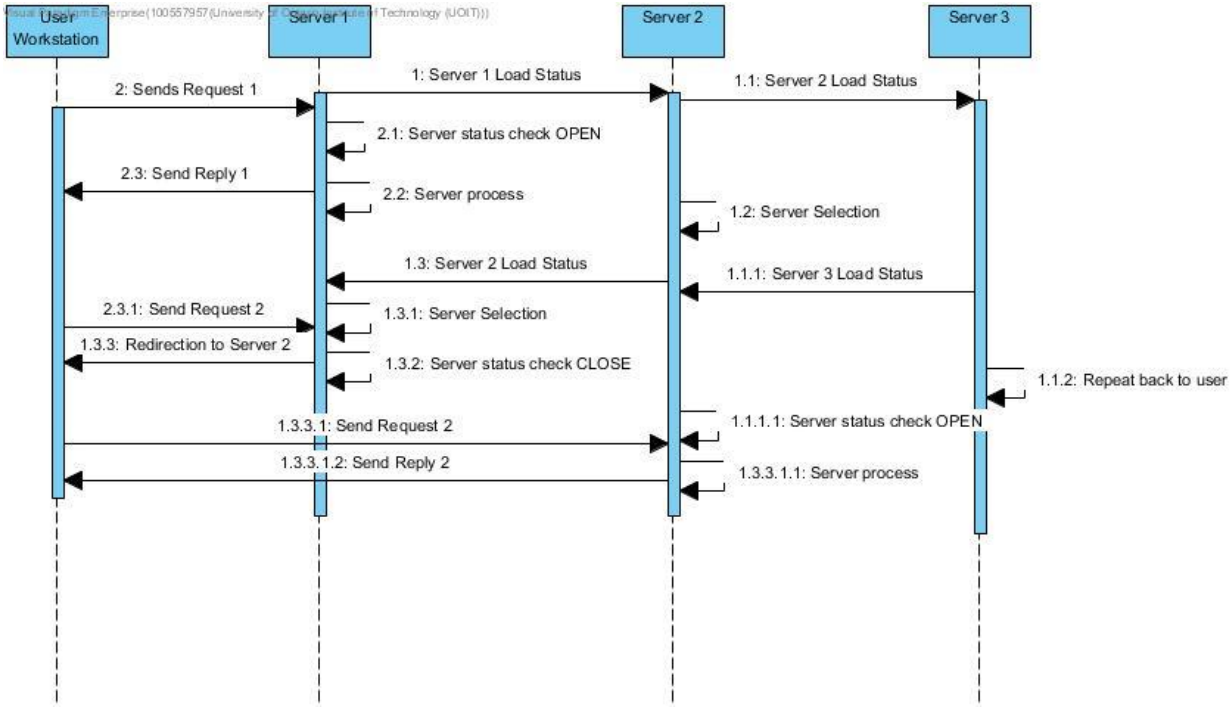
ADD Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Location	Rationale
To backup a server we need to implement software element	At the point when the server is always backed up if any significant issues happen the reinforcement server picture can be utilized to re-establish any lost or harmed data. This fulfills QA-6 and QA-2
Upload a load balancer	Load balancer is conveyed before server. Various application servers are made and stack balancer is sent to direct movement before movement gets to the server. This fulfills QA-2 and CON-2
Implementing a Server monitoring software	A server monitoring software monitors the server in the event that any little issue happens which can additionally help keep any expansive issue. This fulfills QA-2

ADD Step 6: Sketch Views and Record Design Decisions



Element	Responsibility
Load Balancer	Dispatches (and parities the heap of) demands originating from customers to the application servers. The load balancer too presents a one of a kind IP address to the customers.



This sequence diagram explains the load balancer

ADD 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
	QA-1		The security director decides client login and allocate benefit in view of client type

	QA-2		As expressed in iteration 1 a cloud benefit for the web server will be utilized which will hold downtime to a least. Load balancer, server observing and server reinforcement help keep server downtime to a least.
	QA-6		The all-encompassing relational database framework gives great versatility what's more, extensibility and additionally interoperability so extraction of the information from the database progresses toward becoming sensible
	QA-8		Utilizing of the Database Management System gives high viability to the database.

			Consistent server observing calculation predicts and anticipate issues
	CON-2		The utilization of a cloud server makes a difference satisfy this worry. And additionally usage of Load balancer
CON-3			No decisions have been made yet
	CON-4		With a 3 level database we can include monstrous measures of data furthermore, testing should be possible without being an impairment to the clients since it won't influence downtime. The layers can be isolated so the testing should be possible on one layer without interfering with the other.
	CON-5		With a 3 level database we can include monstrous measures of data

			furthermore, testing should be possible without being an impairment to the clients since it won't influence downtime. The layers can be isolated so the testing should be possible on one layer without interfering with the other.
	CRN-1		The fundamental relational database is structured and further enhancement is required
CRN-2			No decisions have been made yet
	CRN-3		Dialects have been considered what more is, have been considered as for the learning of the engineers
	CRN-4		With the database structure and essential engineering been structured, the work can be

			circulated to finish singular piece of the framework