Iteration 3: Addressing Quality Attribute Scenario Driver

In this section, we are building on the fundamental structural decisions made in the previous iterations. We can reason about the fulfillment of one of the most important quality attributes: this iteration focuses on quality attribute scenario QA-2 (Availability).

Step 2: Establish Iteration Goal by Selecting Drivers

For this iteration, the focus is on the QA-2 (Availability) quality attribute scenario. The system should keep functioning in case of a failure within a short period of time.

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

In this availability scenario, the elements that will be refined were identified during the first iteration:

- Application server
- Database server

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

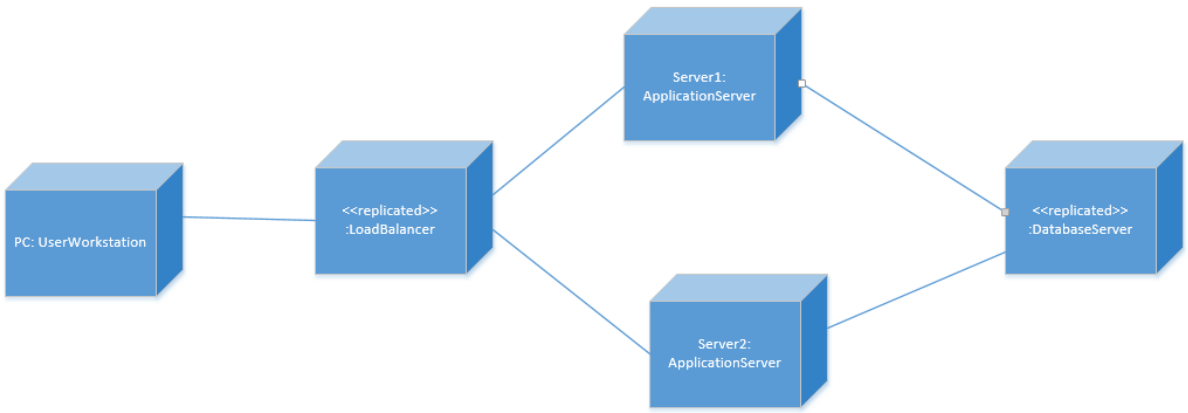
Design Decisions and Location	Rationale and Assumptions
Active redundancy is introduced by replicating the application server and other critical components such as the database	The functionality of the system will not be affected in case of a failure when the critical parts are replicated.
Create backup servers for both the application server and the database server	By having backups of the servers the system can be easily restored in case of an error.

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

Design Decisions and Location	Rationale
Use active redundancy and load balancing in the application server	Since there are two servers that are copies of each other are running at the same time, it is recommended to distribute the load between two servers so that the system can perform in an efficient way.

Step 6: Sketch Views and Record Design Decisions

The figure below shows the updated deployment diagram that introduces the active redundancy and load balancing.



Element	Responsibility
LoadBalancer	Balances the load of requests coming to the main application server
Server2:ApplicationServer	The copy of Server1: ApplicationServer. It helps balancing the load and it operates in case of a failure that might happen in Server1: ApplicationServer.

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
		UC-2	Modules across the layers and preliminary interfaces to support this use case have been identified
		UC-4	Modules across the layers and preliminary interfaces to support this use case have been identified
	QA-1		Some modules across layers and preliminary interfaces to support this have been identified
		QA-2	The load balancer is introduced as well as a replica of the application server. By using a load balancer we are dividing the requests coming to the server. Also, replicated server keeps the system functioning in a short period of time which improves the availability.
	QA-3		No relevant decisions made, but system is modular
		QA-4	Use of rich UI interface support this quality attribute
QA-5			No relevant decisions made
	CON-1		Some modules across layers and preliminary interfaces to support this have been identified
CON-2			No relevant decisions made
		CON-3	The elements that support the associated use case have been identified
		CON-4	The elements that support the associated use case have been identified
		CRN-1	Initial system structure have been established after iterations
		CRN-2	Team's knowledge of Java and related frameworks have been leveraged
		CRN-3	Application has been broken down into units to be distributed across members of team