

SOFE 3650U - Iteration 3 Final Project

Theatre Reservation System

Student Name	Number
Banujan Sutheswaran	100750620
Nathan Bojczuk	100749012
Cyrus Lee	100748627
Yehchan Park	100754068

Step 2: Establish Iteration Goal by Selecting Drivers

The goal during this iteration is to focus on the QA-4 quality attribute scenario: The existing database specified in CON-4 should be able to retrieve and update data to and from the client side within 5 seconds. The database system will schedule a restart once a day for 1 minute to ensure its data integrity.

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

For this scenario, the elements that we will refine are the physical nodes that were identified during the first iteration:

- Application server
- Database server

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

Design Decision and Location	Rationale and Assumptions
Establish performance tactics to manage sample rate, limit event response, and prioritize events.	By reducing the sampling frequency between the Theatre database and the application database, the demand can be reduced resulting in faster response times. Through creating a priority stream that ranks events based on importance and frequency, event processes can be predicted and scaled accordingly. This design decision completes the project's quality attribute 4.
Introduce Database Backup	The addition of this function will prevent real-time database failure, as there will always be another backed up database. This functionality will support constraint 6 that was defined in the previous iterations.

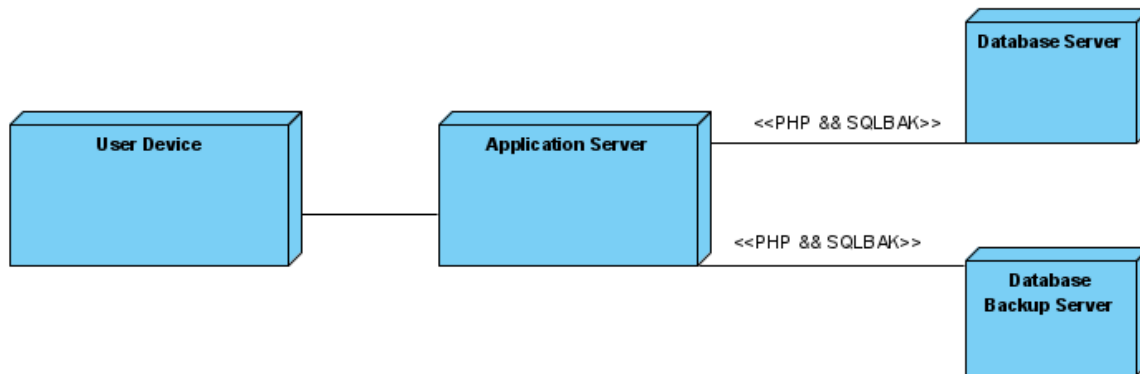
Step 5: Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

The instantiation design decisions are summarized in the table:

Design Decisions and Location	Rationale
Deploy the database replica	Utilizing SqlBak, we can deploy the database contents on a separate database to ensure that our clients integrity is upheld.
Reduce sampling frequency and create a priority stream.	In order to establish performance tactics for minimising load and organising events. It would be logical to reduce the sampling frequency of events and to create a priority stream that sorts events based on their importance.

Step 6: Sketch Views and Record Design Decisions

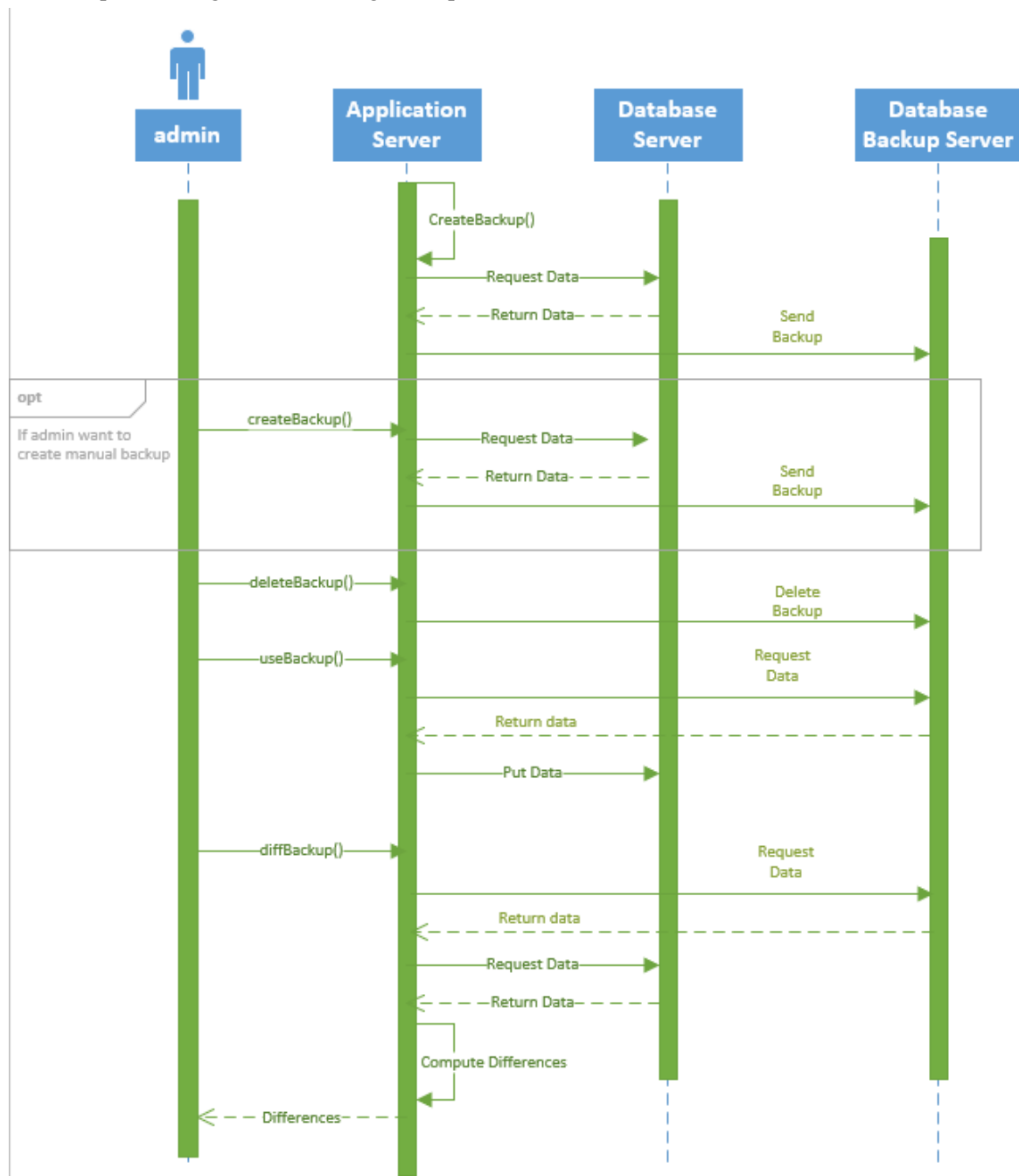
Updated Deployment Diagram:



The following table describes the elements applied in this iteration:

Element	Responsibility
createBackup()	Creates a backup of the current information in the database server.
deleteBackup()	Delete the current backup inside the backup server
useBackup()	Transfer information from backup server to database. Overwrites the information.
diffBackup()	Returns the differences between the database server and the backup server currently.

UML Sequence Diagram Illustrating backup functions:



Step 7: Perform Analysis of Current Design and Review Iteration

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions
	QA-4		Introduced performance tactics to increase overall system efficiency.
		CON-4	The theater's existing database is still in use; an additional backup server was introduced , which does not replace the original database server. This allows the website to be more secure and reliable in the event of data loss.
		CON-6	Using SQLBAK the server will now perform an automatic backup of its contents every 30 days. An admin also has the ability to initiate a backup. Other functions related to it have been created as well.
	QA-1		No relevant decisions have been made for this iteration
	QA-5		
	QA-7		
	QA-9		
	CON-1		
	CON-2		
	CON-3		
	CON-5		