SOFE 3650U: Software Design & Architecture

Addressing Quality Attribute Scenario Driver - ADD Iteration 3

Date: November 19th, 2021

Group Members:

Foram Gandhi (100699245)

Kinjal Shah (100743551)

Rutvi Shah (100747171)

Danial Shaikh (100698628)

Iteration 3: Addressing Quality Attribute Scenario Driver

Step 2: Establish Iteration Goal by Selecting Drivers

This iteration will focus on the QA-4 (Performance) quality attribute scenario. QA-4 states that the website must be able to produce transactions and display tickets in a timely manner. There should be little delay, <1 minute, when performing these transactions.

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

The elements to be refined in this iteration will be the physical nodes identified in iteration 1, which are the time servers and database servers.

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

Design Decision and Locations	**Rationale and Assumptions**
Introducing an element from the **message queue**.	Tickets received from the time servers are placed in the message queue then later retrieved by the application. Use of a queue will guarantee that tickets are processed and delivered in a timely manner.
Introducing the **performance** tactic by **prioritizing events.**	Prioritizing transactions by importance of events according to the service provided to the user. Ticket booking transactions made first will get processed first (according to the time).

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

Design Decision and Locations	**Rationale and Assumptions**
Deploy message queue on a separate node	Doing so will guarantee that no transaction data will be lost in the case that the time servers are down according to CON-3.
Identifying the events to be prioritized.	Create the prioritization scheme to rank the order of services according to importance.

Step 6: Sketch Views and Record Design Decisions

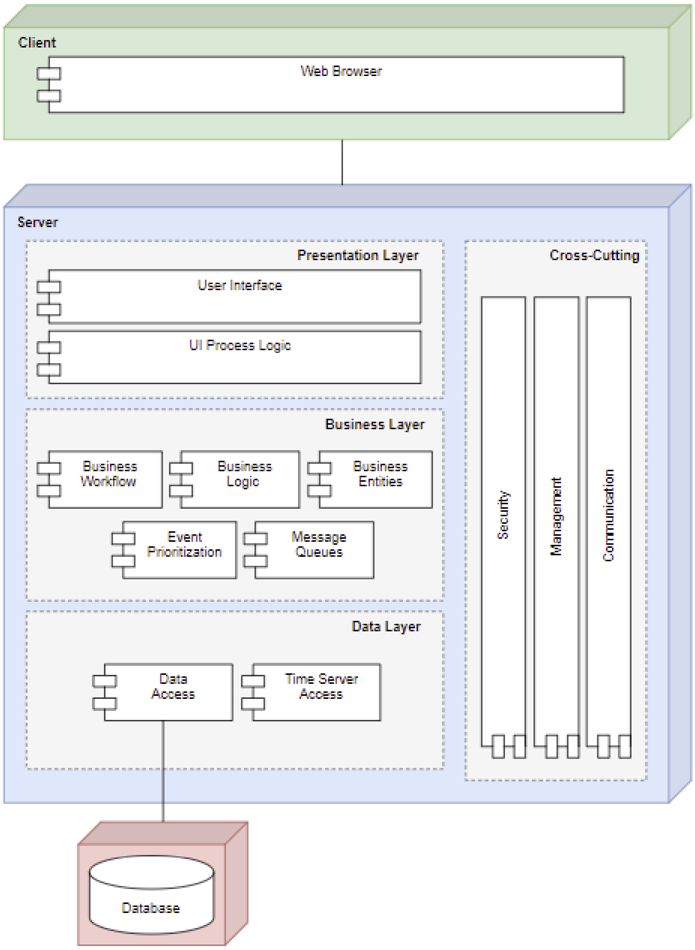

Fig 10: Refined Layered Architecture Model (Web Application Reference Architecture)

The responsibilities for the elements identified in Figure 10 are summarized in the following table:

Layer	**Module**	**Responsibility**
Client	Web Browser	Movie Ticket Booking Application runs on the client's web browser.
Server - Presentation Layer	User Interface	Receives user interactions and presents the information to the users. It contains UI elements as buttons and text

		fields as well as a form in the ticket checkout page.
Server - Presentation Layer	UI Process Logic	Manages the control flow of the application’s use cases and provides data coming from the business layer to the user interface components. It is also responsible for data validation. An example is the Login/Signup Page.
Server - Business layer	Business Workflow	Involves the execution of multiple use cases (such as ticket selection and seat selection since seats selected must be equal to number of tickets previously selected)
Server - Business Layer	Business Logic - Payment - Ticket Booking	Retrieves and processes application data and applies business rules on the data (such as seat booking page and checkout page).
Server - Business Layer	Business Entities - User - Movie Information - System	Represents the entities from the business domain and the associated business logic.
Server - Business Layer	Event Prioritization - First come first serve	Prioritizing the events from level of importance to the user.
Server - Business Layer	Message Queues	Using message queues to prioritize the events to be received by the application and to make sure the system time server does not fail.
Server - Data Layer	Data Access - MySQL database - OMDB API	Provides the common components needed to retrieve and store information. It uses the OMDB API for retrieval of movies and movie info, as well as retrieval from the

		database to verify/check login/signup info.
Server - Data Layer	Time Server Access	This module isolates and abstracts operations to support communication between different types of time servers partially addressing QA-4.
Server - Cross-Cutting Layer	Security - User authentication - Password encryption/decryption	Handles security aspects such as user authorization and authentication, as well as password encryption/decryption
Server - Cross cutting Layer	Management	Handles validation, exception management and logging across all layers.
Server - Cross-Cutting Layer	Communication - Data layer communicates with database - Time Servers	Handles communication mechanisms across layers and physical tiers
External Data Sources	Database	Stores all data from the system, later accessed by the DataAccess module in the Data Layer (using MySQL to store databases).

UC-5: Confirmation Response Time

The UML sequence diagram shown in Fig 11 shows the system and Time Servers components satisfying UC-5 (Confirmation Response Time), which is associated with QA-2(Interoperability), where the time servers must be able to send a confirmation email to the user after booking in <1 minute. The user is sent confirmations for booking and making the Payment for the ticket in a timely manner within the restrictions.

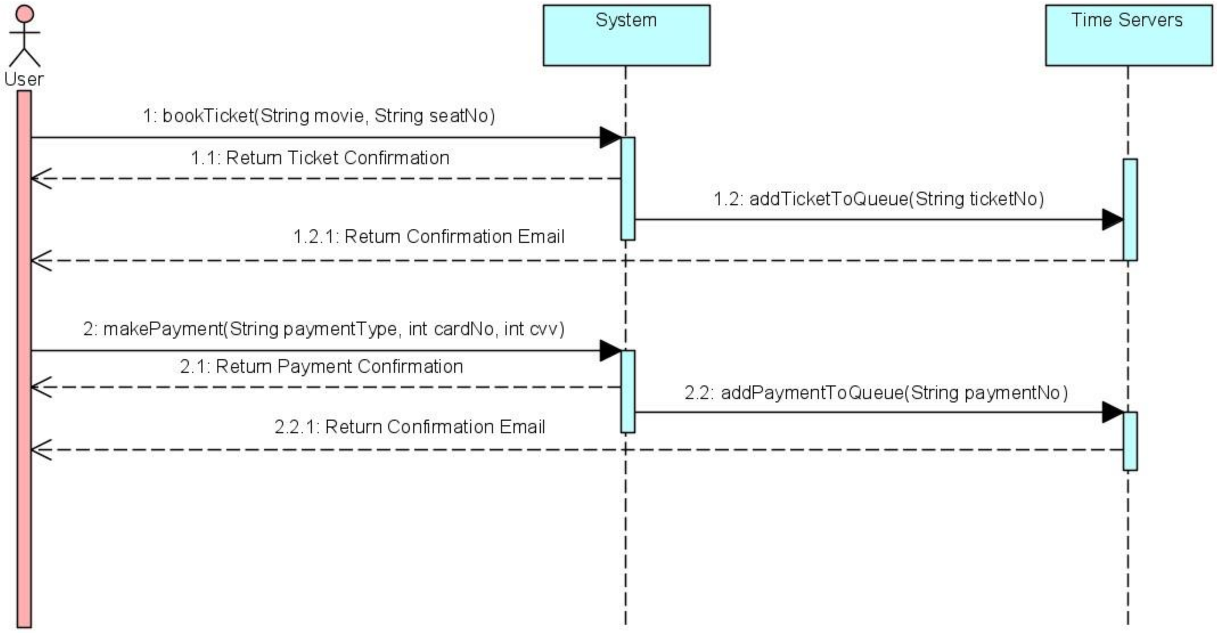
User
System
Time Servers
1: bookTicket(String movie, String seatNo)
1.1: Return Ticket Confirmation
1.2: addTicketToQueue(String ticketNo)
1.2.1: Return Confirmation Email
2: makePayment(String paymentType, int cardNo, int cvv)
2.1: Return Payment Confirmation
2.2: addPaymentToQueue(String paymentNo)
2.2.1: Return Confirmation Email

Fig 11: UML Sequence Diagram for UC-5

From the interactions identified in Fig 11, methods for the interfaces of the interacting elements are identified in the following table:

Element	**Method Name**	**Description**
System	String bookTicket(String movie, String seatNo)	The user must book the ticket, the function returns a confirmation to the user of the ticket booking with a ticket number. The ticket number gets sent to the Time Servers to be put into a queue.
	String makePayment(String payment Type, int cardNo, int cvv)	The user must choose and make a payment, the function returns a confirmation back to the user of payment with the payment number. The payment number gets sent to the Time Servers to be put into a queue.
Time Servers	String addTicketToQueue(String ticketNo)	The Time Servers adds the ticket next in the queue, the

	String addPaymentToQueue(String paymentNo)	function returns an email back to the user in less than a minute with a confirmation that their ticket has been booked. The Time Servers adds the payment next in the queue, the function returns an email back to the user in less than a minute with a confirmation that their payment has been processed.

Step 7: Perform Analysis of Current Design and Review Iteration

The status of the various drivers, as well as the decisions taken throughout the iteration, are summarised in the table below. The table has been cleared of drivers that were fully handled in the previous iteration.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions Made During the Iteration
	UC-6		Partial addressing of this use case in the refined model.
		UC-7	The refined modules in the architectural model address this use case.
	UC-8		Partial addressing of this use case in the refined model.
		UC-9	The refined modules in the architectural model address this use case.
		QA-4	This quality attribute is the primary goal for this iteration and has been supported through the refined models.
QA-5			No relative decision has been made for this particular quality attribute.

	CON-1		Decisions for the work efficiency of the technician have not been made yet.
	CON-2		Decisions for the communication relationship between the developer and technician have not been made yet.
		CON-3	Constraint is referenced by the architecture model that has been refined.
	CON-6		Constraint has been partially addressed with the introduction of message queues and event prioritization.
	CRN-2		Technologies and reference architecture that have been considered up to this point have taken into account the knowledge of the developers.