SOFE 3650U - Iteration 1 Final Project

Theatre Reservation System

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

ADD Step 1: Review inputs

<table>
<tr><th>Category</th><th>Details</th></tr>
<tr><td>Design Purpose</td><td>This is a greenfield system from a mature domain. The purpose is to create a sufficient design to support the construction of the system.</td></tr>
<tr><td>Primary Functional Requirements</td><td>From the use cases presented in Deliverable 2, the primary ones were determined to be:
<ul>
<li>UC-1: Because it directly supports the core business</li>
<li>UC-4: Because it directly supports the core business</li>
<li>UC-6: Because it directly supports the core business</li>
<li>UC-7 Because it grants the ability to troubleshoot and modify in real time</li>
</ul></td></tr>
<tr><td>Quality Attribute Scenarios</td><td>The scenarios were described in Deliverable 2. They have now been prioritized as follows:
<table>
<tr><th>Scenario ID</th><th>Importance to Customers</th><th>Difficulty of Implementation according to architect</th></tr>
<tr><td>QA-1</td><td>High</td><td>Medium</td></tr>
<tr><td>QA-2</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-3</td><td>Low</td><td>Low</td></tr>
<tr><td>QA-4</td><td>High</td><td>Medium</td></tr>
<tr><td>QA-5</td><td>Medium</td><td>Medium</td></tr>
<tr><td>QA-6</td><td>Low</td><td>Low</td></tr>
<tr><td>QA-7</td><td>High</td><td>High</td></tr>
<tr><td>QA-8</td><td>High</td><td>High</td></tr>
<tr><td>QA-9</td><td>Medium</td><td>Low</td></tr>
</table>
From this list quality attributes 1, 4, 7, and 8 are selected as drivers.</td></tr>
<tr><td>Constraints</td><td>All the constraints discussed in the second deliverable are included as drivers.</td></tr>
<tr><td>Architectural Concerns</td><td>
<table>
<tr><td>ID</td><td>Concern</td></tr>
</table></td></tr>
</table>

	CRN-1	Establishing an overall initial system structure.
	CRN-2	Leverage the team's knowledge about Web Programming, including HTML, CSS, Javascript, PHP, SQL and AJAX .
	CRN-3	Allocate work to members of the development team.

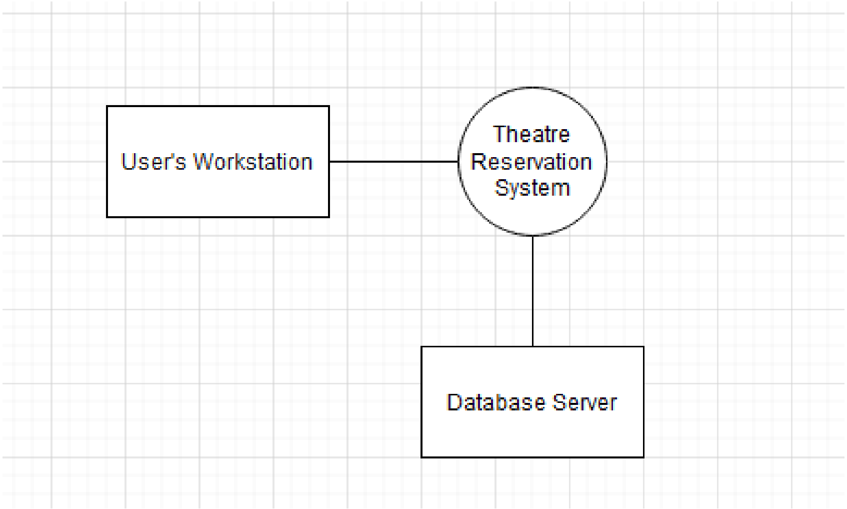
Step 2: Establish Iteration Goal by Selecting Drivers

This is the first iteration in the design of a greenfield system, so the iteration goal is to achieve the first architectural concern of establishing an overall system structure .

Although this iteration is driven by a general architectural concern, the architect must keep in mind all of the drivers that may influence the general structure of the system. In particular, the architect must be mindful of the following:

QA-1: Security
QA-4: Performance, Availability
QA-7: Security
QA-8: Security
CON-2: User workstations use the following operating systems: Windows and Linux.
CON-4: Must use the theatre's existing databases.
CON-5: The system must be accessed through a web browser such as Chrome or Firefox while on different operating systems.
CRN-2: Leverage the team's knowledge about Web technologies, including HTML, CSS, Javascript, PHP, SQL and AJAX .

Context Diagram for the theatre reservation system.



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

Our system is a greenfield system, so the element to refine is the entire theatre reservation system, which is shown in the diagram above. Refinement is performed through decomposition.

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

<table>
<tr><th>Design decisions and Location</th><th>Rationale</th></tr>
<tr><td>Theatre Reservation System will be structured using the <b>Rich Internet Application</b> reference architecture</td><td>The Rich Internet Application reference architecture supports the development of applications that utilize a rich user interface and runs inside a web browser. This type of application supports business logic on the client side which allows a responsive exchange of information between the client and server part of the system. This advantage is especially useful in achieving QA-4, where data is required to be updated/retrieved quickly. Constraint 5 is also satisfied with the Rich Internet Application, as it allows the system to be accessed through the user's choice of web browser.</td></tr>
<tr><td></td><td><b>Discarded alternatives:</b>
<table>
<tr><th>Alternative</th><th>Reason for discarding</th></tr>
<tr><td>Web Application</td><td>While the design does include a very mobile and web based option, it does have limitations for what it can accomplish. This design concept was ultimately discarded due to the difficulty of implementing a rich user interface.</td></tr>
<tr><td>Rich Client Application</td><td>This architecture is designed to be highly responsive by running on the users’ machines. The design was discarded as the desired system requires high network connectivity and does not want to heavily use the user’s machine resources.</td></tr>
<tr><td>Mobile Applications</td><td>This architecture is focused toward the development of applications that are to be deployed on mobile devices. This alternative was discarded because mobile devices were not considered for accessing this system.</td></tr>
</table>
</td></tr>
<tr><td>Physically structure the application using the <b>three-tier deployment</b></td><td>Since the system will be accessed from a web browser (CON-5) and the theatre's existing database must also be used (CON-4), a three tier deployment is appropriate. Includes the presentation tier, application tier, and data tier.</td></tr>
</table>

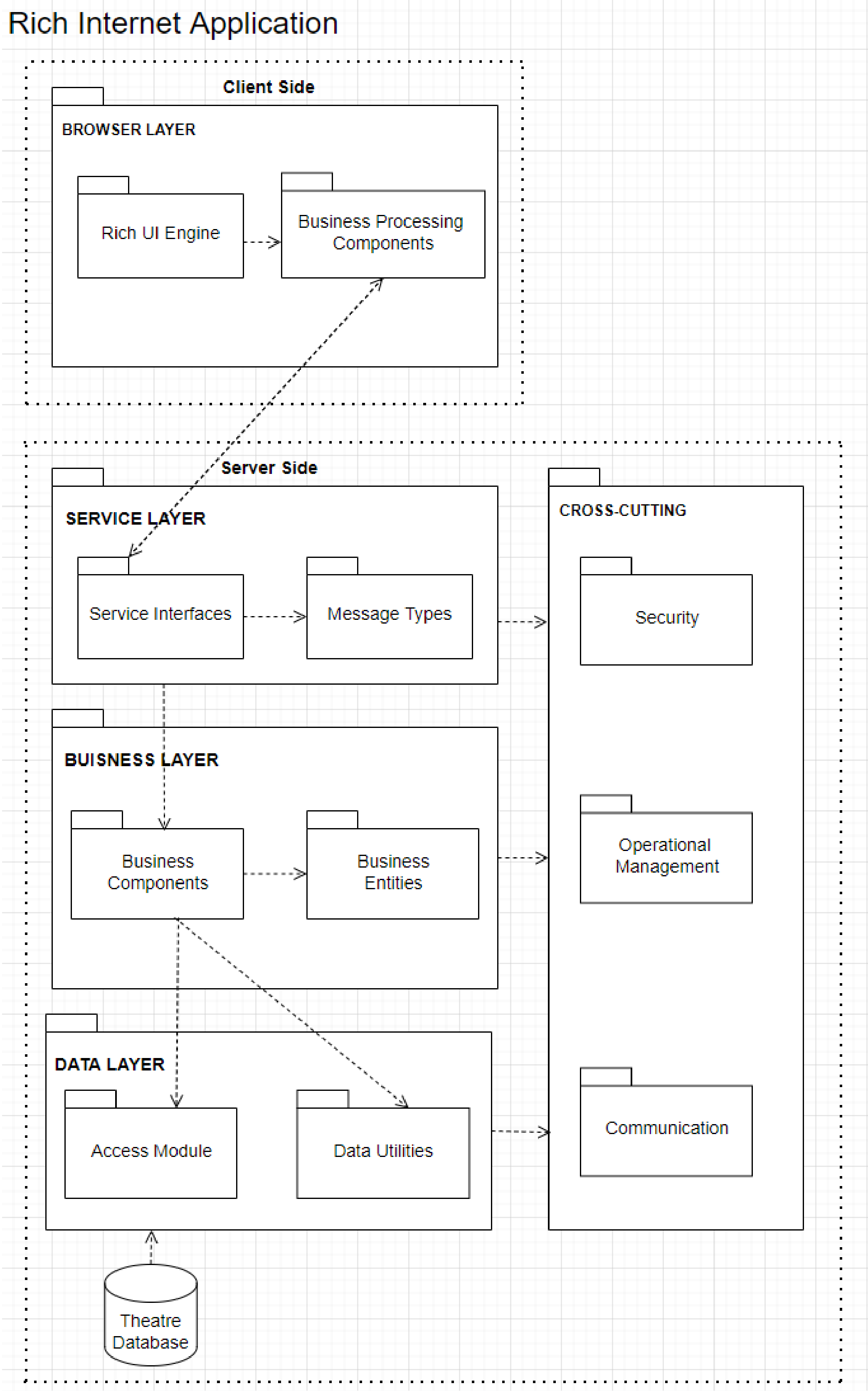
pattern	
Build the user interface of the client using HTML and other **web technologies**	The standard for building web applications ensures proper compatibility (CON-5) and it is what the developers are already familiar with (CRN-2). Discarded Alternatives: Java servlets were considered , but the developers understanding of the technologies was not sufficient enough.

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

In the initial iteration, interfaces are not yet defined.

Design Decision and Location	**Rationale**
Remove local data sources in the Rich Internet Application	No need to store data locally as information will be stored in the Theatre Database.

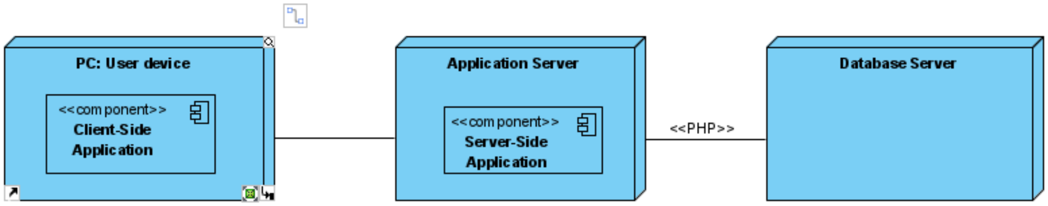
Step 6: Sketch Views and Record Design Decisions



The following table summarizes the information that is captured by the sketch :

Element	**Responsibility**
Browser Layer	This layer contains the canvas for user interaction and use case control flow. Will use HTML and CSS to implement.
Service Interface	Various service components are utilized by the end users.
Business Layer	This layer contains the modules that perform the business logic operations. This can be carried out on the server side.
Data Layer	This layer houses the components needed to communicate with the theatre server.
Cross Cutting	This extra hidden layer contains the website's security, operational management, and communication with the theatre database server. The service, business, and data layer all make use of this helpful layer.
Rich UI Engine	This highly interactive user interface will make the software feel more responsive to the end user.
Business Processing Component	Element will send and receive delicate information from the client and server sides.
Message Types	Contains the appropriate class of variable to send or receive.
Business Components	Incorporate business operations that require processing on the server side.
Business Entities	These entities make up the domain model.
Access Module	Retrieves data from theatre database.
Data Utilities	Tools in order to retrieve and collect data. Used in the business layer.

Deployment Diagram:



The responsibilities of the elements are:

Element	Responsibility
User Device	The device (most likely a pc), which houses the client side logic part of the program
Application Server	The server that host the server side logic of the program, it also serves the web pages
Database Server	Server that hosts the the relational database

The relationship between elements:

Relationship	Description
Between application and database server	Communication with the database will be done using PHP.

Step 7: Perform Analysis of Current Design and Review Iteration

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions
	UC-1		Selected reference architecture establishes the modules that will support this functionality.
	UC-4		
	UC-6		
	UC-7		
	QA-4		Introduced business logic module on the client side to provide fast and secure data transfer.

		CON-5	Use of the Rich Internet Application being programmed in Java, which supports the website across different browsers and operating systems.
	QA-1		Selected relevant drivers.
	QA-4		
	QA-7		
	QA-8		
		CRN-2	Leverage the team's knowledge about Web technologies, including HTML, CSS, Javascript, PHP, SQL and AJAX .
	CON-2		Took into account the following elements, mindful while deciding project drivers.
	CON-4		
UC-2			No relevant decisions made, elements did not fit this iterations scenario.
UC-3			
UC-5			
UC-8			
QA-2			
QA-3			
QA-5			
QA-6			
QA-9			
CON-1			
CON-3			
CON-6			