Project Progress

SOFE 3650U

Nov 09, 2022

Nathan Hacault	100778576
Zainab Nomani	100784761
Evidence Okeke	100755328

1. **Review Inputs**

Use Case	**Description**
UC-1 Managing Dynamic Course Information	Lecturer can post messages and manage who can see archived items
UC-2 Managing Static Course Information	The administrator updates the lectures and modules of the course

ID	**Quality Attributes**	**Description**
QA-1	Usability	Lecturers are able to modify course content, grades, and messages.
QA-2	Modifiability	Administrators and lecturers are able to recreate and modify courses.

ID	**Description**
CON-1	The system must be accessed through a web browser (Chrome, Firefox, etc)

2. **Establish Iteration Goals by Selecting Drivers**

This is the first iteration in the design of a greenfield system, so the iteration goal is to achieve the overall architecture of the system keeping in mind:

- QA-1: Usability
- QA-2: Modifiability
- CON-1: The system must be accessed through a web browser

3. **Choose One or More Elements of the System to Refine**

We want to refine the entire Course Management System (CMS)

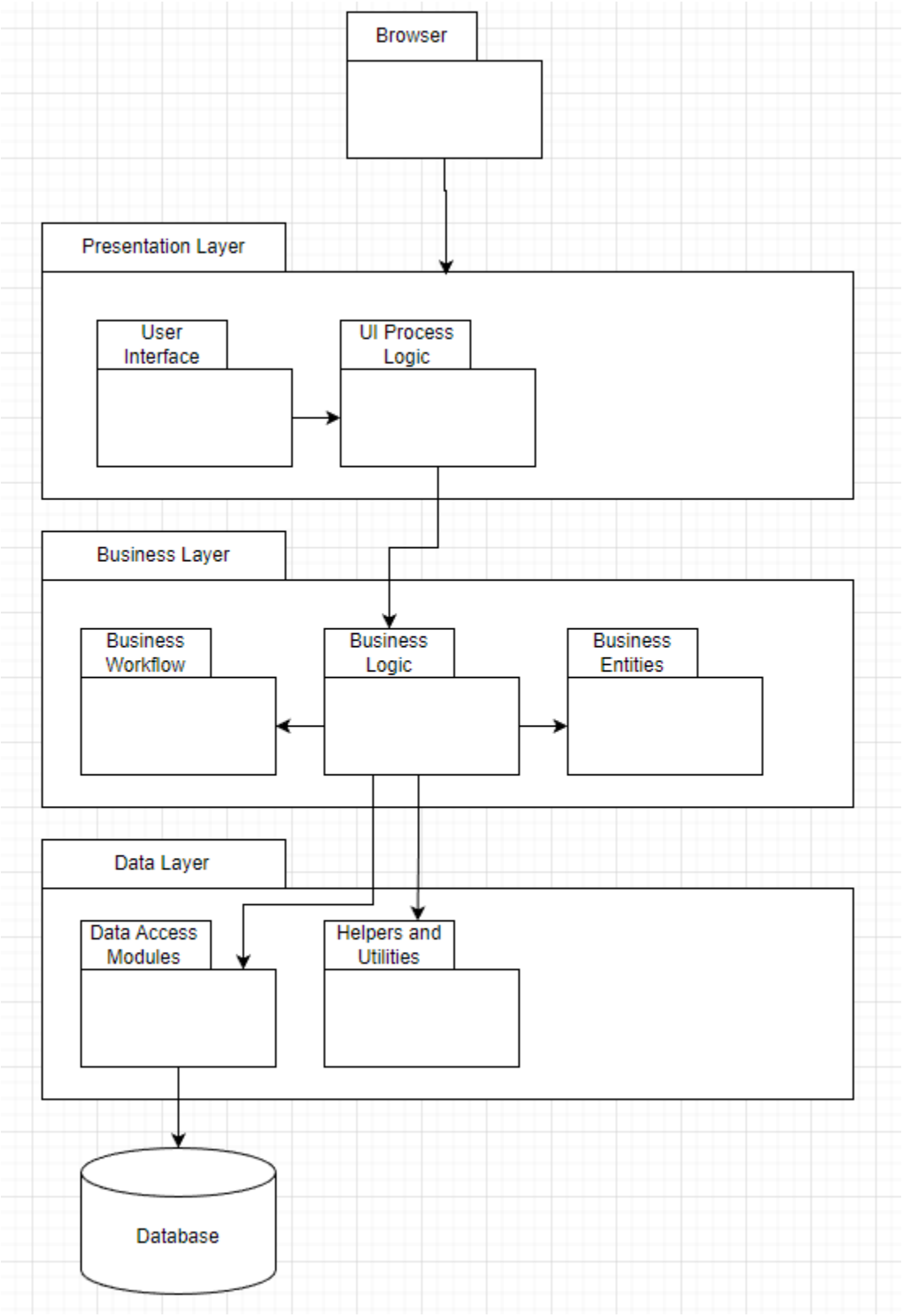
4. Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Location	Rationale
Logically structure the system using the Web Application reference architecture	The Web Application reference architecture supports the development of systems that are accessible over the internet through a web browser (CON-1). The system must support the capability of allowing a lecturer to post messages and manage who can see archived items (UC-1). Administrators must also be able to update the lectures and modules of a course (UC-2).
Include a edit feature for lecturers to modify their courses	Lecturers must be able to modify course content, grades, and messages (QA-1).
Include a copy function to duplicate courses and an edit function to modify them	Administrators and lecturers must be able to recreate and modify courses (QA-2).
Build the website using JavaScript and other web development tools.	As this is a web browser application (CON-1) and the development team has expertise in JavaScript.

5. Instantiate Architectural Element, Allocate Responsibilities, and Define Interfaces

Design Decisions and Location	Rationale
Remove Application Facade	Since the UI of the system will be simple and straightforward, no facade is needed.
Remove Service Agents	The system will not be communicating with any other systems. All content received from other Course Managements Systems will be done through export/import of CSV or other files.

6. Sketch View and Record Design Decisions



Element	Responsibility
Browser	A web browser running on the client machine
User Interface	These components are responsible for receiving user interactions and presenting information to the users. They contain UI elements such as buttons and text fields.
UI Process Logic	Responsible for managing the control flow of the application's use cases. Responsible for other aspects such as data validation and orchestrating interactions with the business logic.
Business Workflow	Responsible for managing long running business processes.
Business Logic	Responsible for retrieving and processing application data and applying business rules on the data.
Business Entities	These components represent the entities from the business domain and their associated business logic.
Data Access Modules	Provide common operations used to retrieve and store information in the database.
Helpers and Utilities	These components are responsible for the logic needed for data access tasks. Both helper and utility components can often be reused in other applications.
Database	Hosted database which is responsible for storing all information (course, student, instructor, admin).

7. **Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose**

Not Addressed	Partially Addressed	Completely Addressed
	UC-1	
	UC-2	
	QA-1	
	QA-2	
	CON-1	