Iteration 2: Identifying Structures to Support Primary Functionality

The goal of this iteration is to address the general architectural concern of identifying structures that support the primary functionality of the application.

Step 2: Establish Iteration Goal by Selecting Drivers

The following primary use cases will be addressed:

UC-2: Create/remove content

UC-4: View Course Information

Step 3: Choose One or More Design Concepts That Satisfy the Selected Drivers

In this iteration we will be refining the server's architecture in the Rich Client Application Architecture.

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

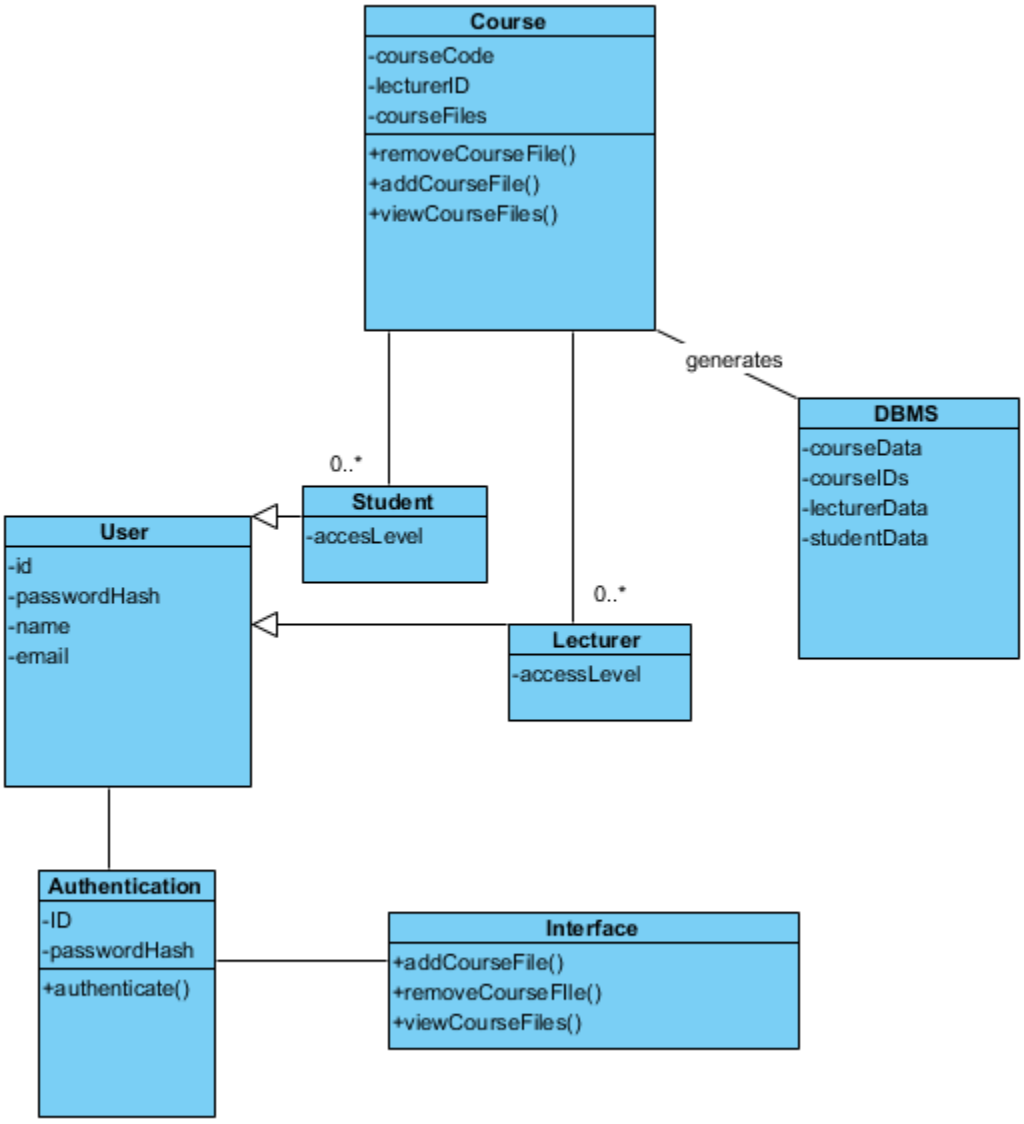
Design Decisions and Location	Rationale and Assumptions
Create a Domain Model for the application	In order to identify the major entities of the system and their relationship with each other in the domain, a domain model needs to be created. No alternatives exist for the domain model as it eventually needs to be discussed.
Identify Domain Objects that map to functional requirements	A domain object is a building block of the application as it encapsulates the functional elements.
Decompose Domain Objects into general and specialized components	Functionality is represented by domain objects but some small-scaled extra elements located within the layers might be needed. Each module is associated with a layer.
Use Spring framework	Spring is selected as the development team is already familiar with it. It is a popular Java framework that is light and it has a good tool support.

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

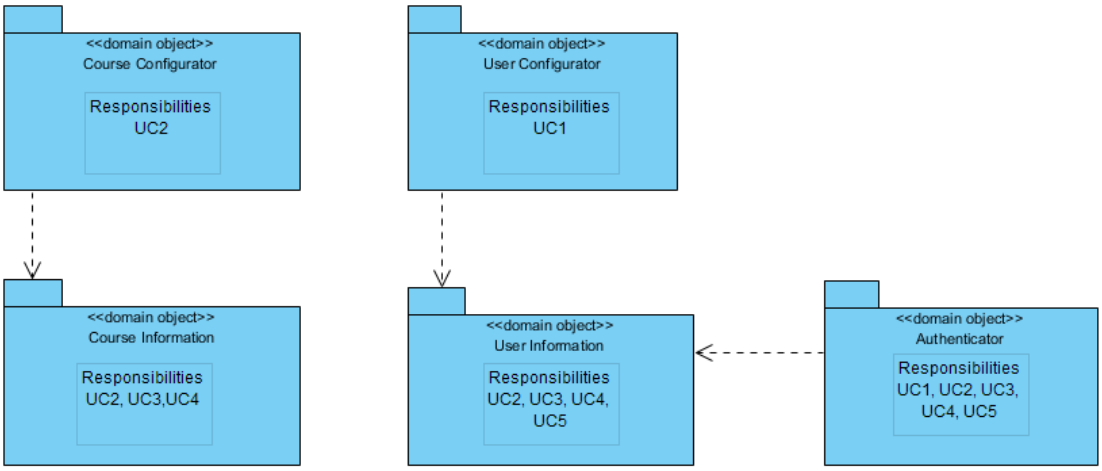
Design Decisions and Location	Rationale
Create only an initial domain model	In order to accelerate the design process only an initial domain model is created.
Map the system use cases to domain objects	The system's use cases need to be analyzed to identify the domain objects. For each use case, domain objects are identified to achieve CRN-3.
Decompose the domain objects across the layers to identify layer-specific modules with an explicit interface	The architect identifies the modules based on the primary use cases. It is recommended that other team members contribute to this process by identifying the rest of the modules. CRN-3 is addressed again by dividing the work between the members in the team. As modules are being identified, a new architectural concern is realized: CRN-4: Most of the module should be unit tested.
Connect components associated with modules using Spring	With Spring the modules can be unit tested and different aspects can be supported.
Associate frameworks with a module in the data layer	Spring is associated with the modules in the server side.

Step 6: Sketch Views and Record Design Decisions

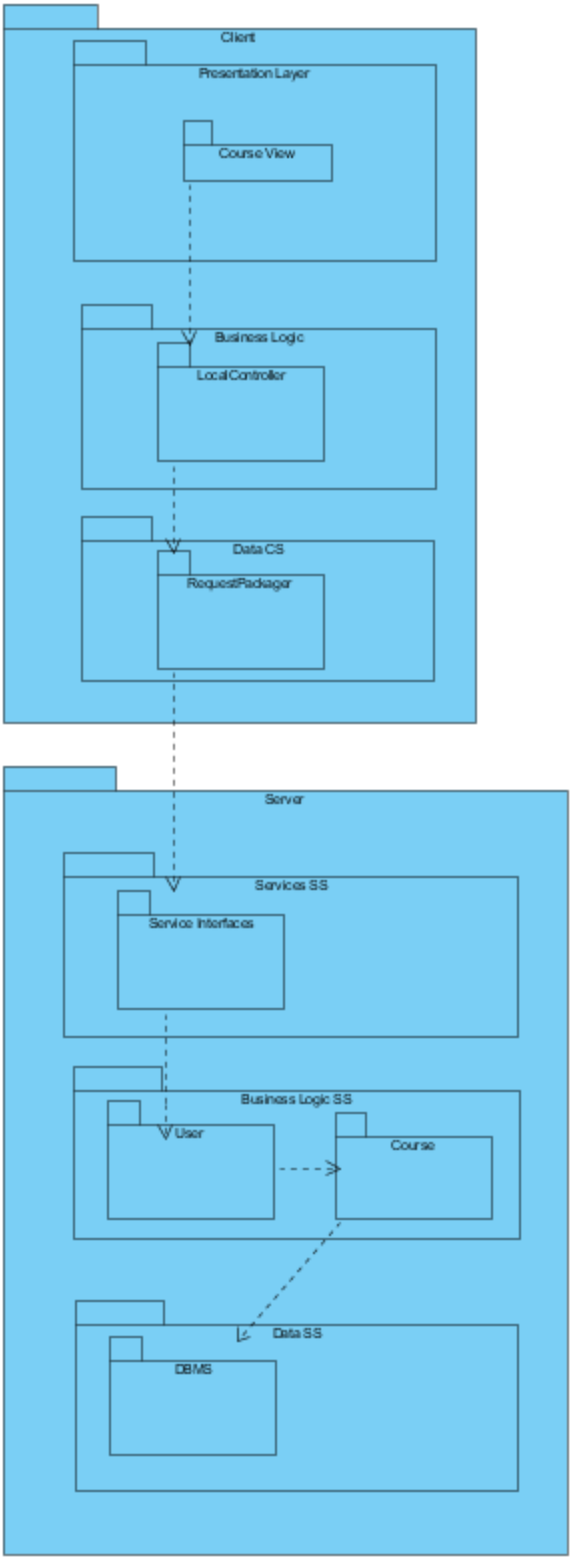
Initial Domain Model



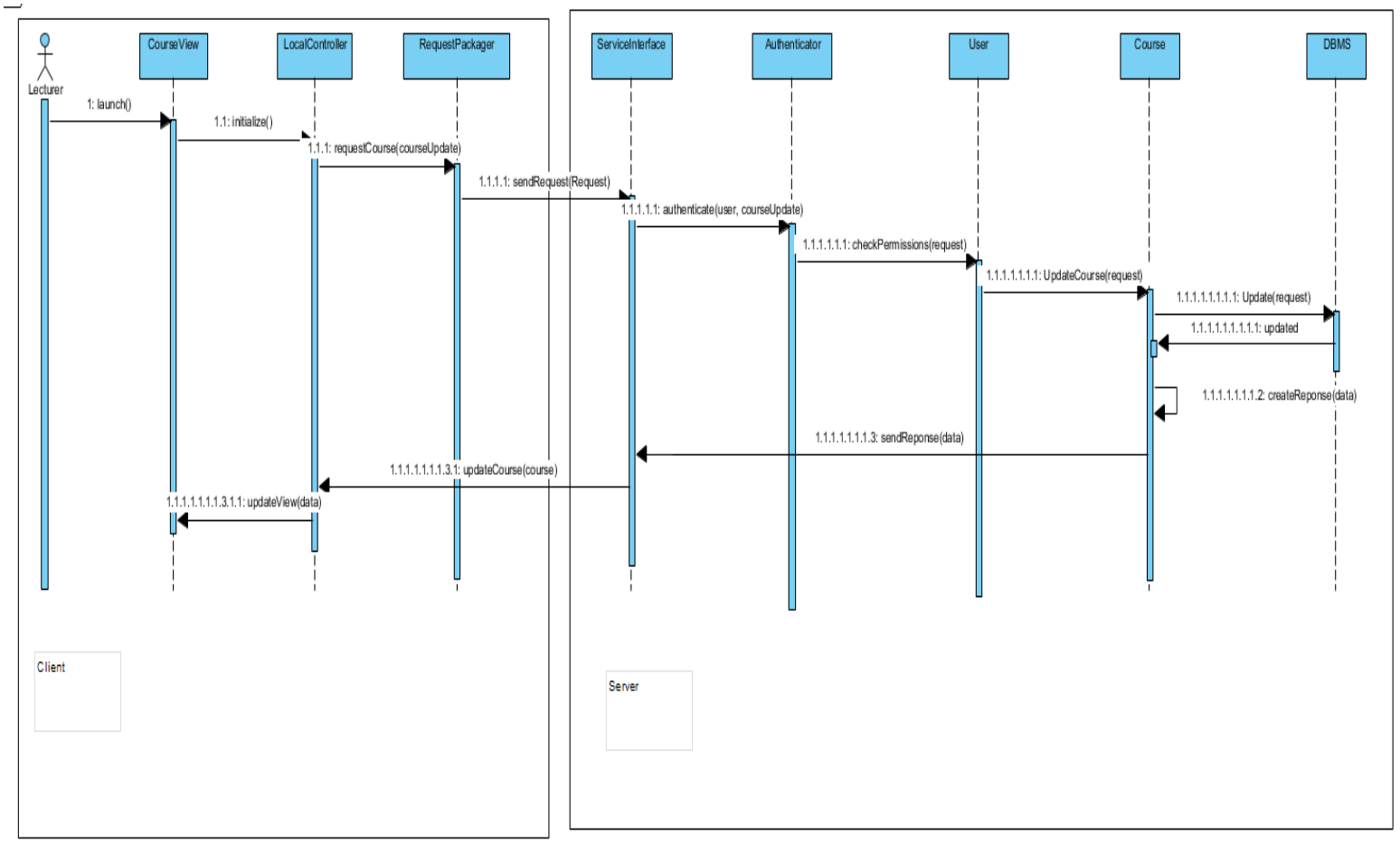
Domain Objects



Module View


Client
Presentation Layer
Course View
Business Logic
Local Controller
Data CS
RequestPackager
Server
Services SS
Service Interfaces
Business Logic SS
User
Course
Data SS
DBMS

Sequence Diagram



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		Some modules across layers and preliminary interfaces to support this have been identified
	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