

Iteration 2: Identify structures to support primary functionality

The objective of this iteration is to identify the structures that support the primary functionality. The structures are from the previous iterations, we now look for further method of implementation. This iteration will be applied in a expeditious manner, addressing the biggest risks first and moving down from there.

Step 2: Establish Iteration Goal by Selecting drivers

In this iteration, we want to address the general architectural concerns of managing the system and communication servers.

In this iteration we pick the following use cases as drivers:

- UC-5
- UC-12
- UC-13

Step 3: Choose One or More elements of the System to refine

The structures that will be refined in this iteration are the design architectures relating to relaying information. These structures are a mixture of many components, each component plays a vital part and are all interconnected.

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

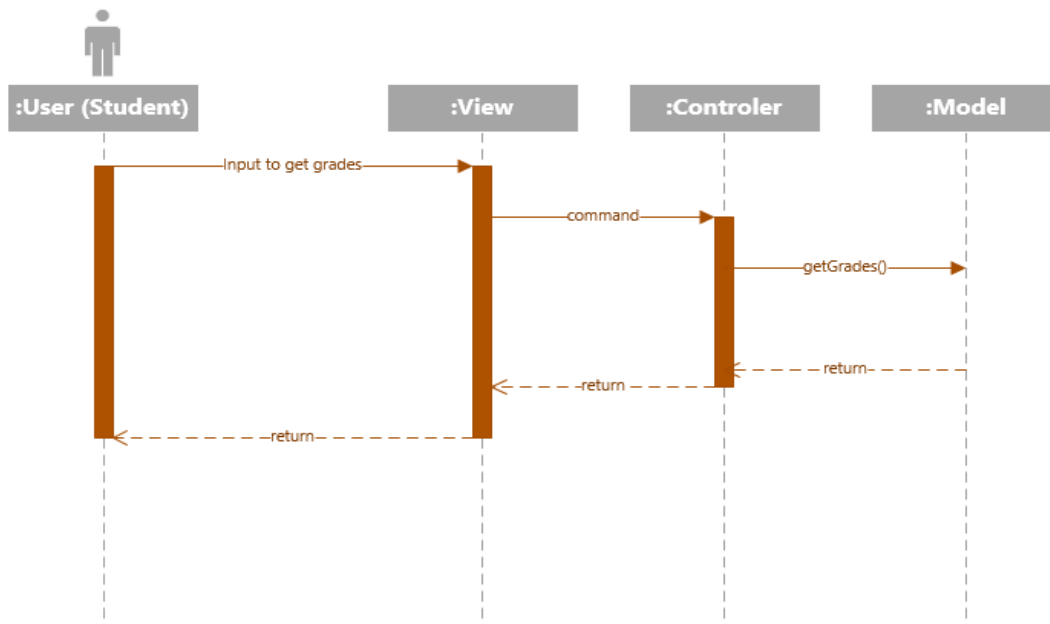
| Design Decision and Location | Rationale and Assumptions |
|--|---|
| Create Domain Model for the application | It is very important to commence the process to create a domain model for CMS. Once the process has been commenced, we must immediately identify the primary drivers and how they are interrelated to each other. |
| Identify Domain Objects that map to functional requirements | We need to figure out the domain objects. They will be primarily integrated within the system. |
| Decompose Domain Objects into general and specialized Components | The Domain Objects will represent their own functionality. They are all supported by more in depth within the system by the other layers. The objects are all related to their corresponding layers. |
| Use Spring framework and Hibernate | Allows all the objects to form and maintain a connection. Improvements in performance can be observed within the layers. |

Step 5: Insatiate Architectural Elements, Allocate Responsibilities

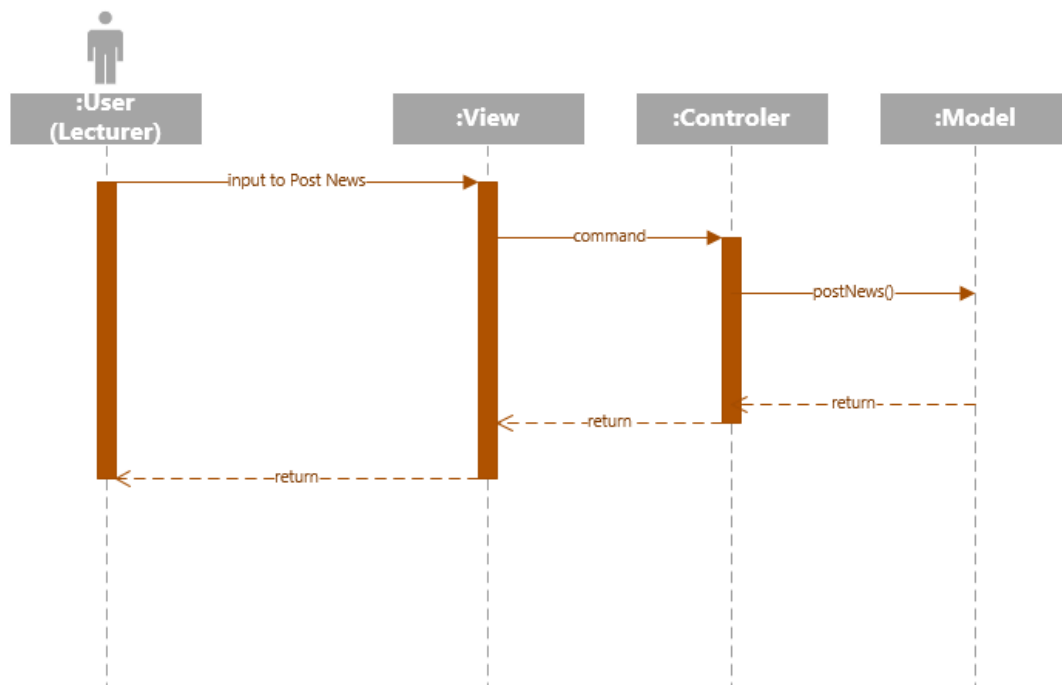
| Design Decision and Location | Rationale and Assumptions |
|---|---|
| Create the initial domain model | We must make the first domain model and ensure it contains ours selected drivers. |
| Map the system usecases to the domain objects | We can create a map of our domain by analyzing our usecases. |
| Decompose the domain objects across the layers to layer-specific modules with an explicit interface | Once all functionality have been identified, we will have a clearer image of which ones must be tested. This will reduce testing by skipping over lesser used ones. |

Step 6: Sketch Views and Record Design Decisions.

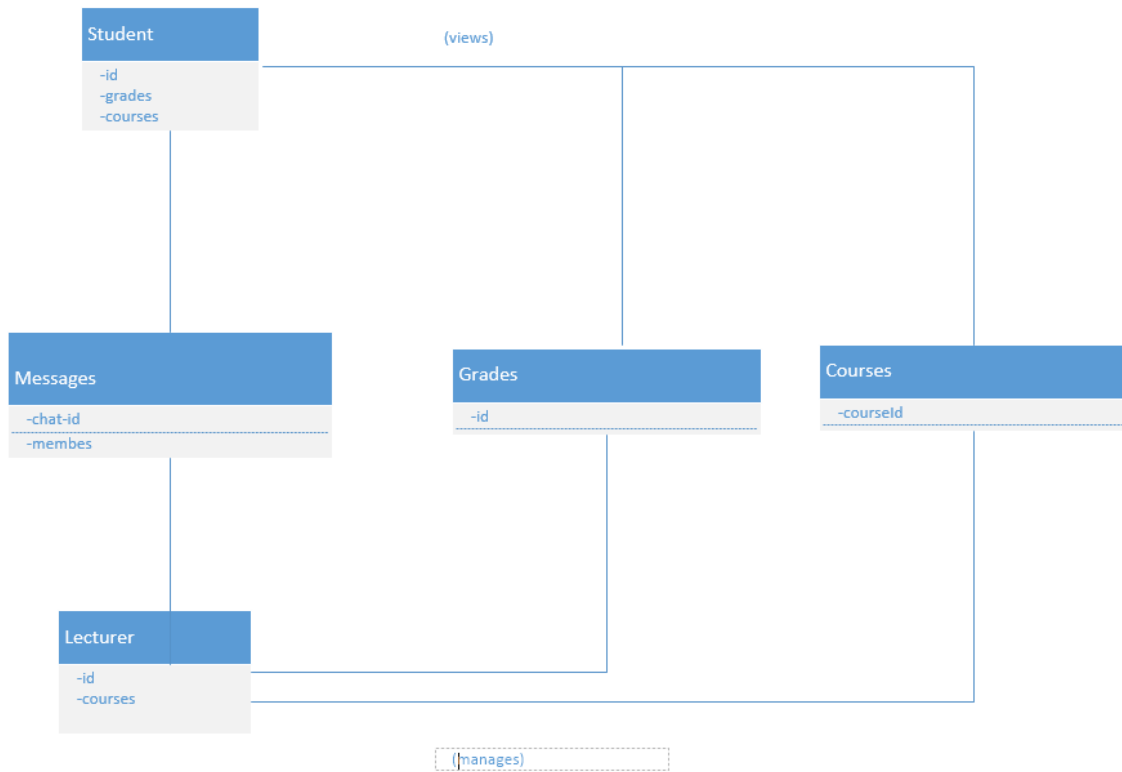
Following are sequence diagrams that provide views of two use cases - Lectures posting news messages and students getting grades from a database



Sequential diagram for checking grades



Sequential diagram for posting news messages



Domain model for Lecturers and Students using message, grade and course entities

Step 7: : Perform Analysis of Current Design and Review Iteration.

| Not Addressed | Partially Addressed | Completely addressed | Design Decisions Made During the Iteration |
|---------------|---------------------|----------------------|--|
| | | UC5 | Knowing the layout of the system allows us to know which components are involved in notifications. |
| | UC2 | | No relevant decision made. |
| | | UC-7 | No relevant decision made. |
| | | UC-12 | Identified the components that need to be implemented for the uwe case. |
| | | UC-13 | The mapping of the network allows to visualize and implement messaging use case for all needs.. |

