# SOFE 3650U - Final Project

## ADD Iteration 2

| USE CASE | DESCRIPTION |
|---|---|
| UC 1 - Subscribe and unsubscribe | Students subscribe or unsubscribe from a course or exam if they meet the requirements. |
| UC 2 - Manage static course information | The administrator updates the name of a lecturer and study materials for a course. |
| UC 3 - Manage dynamic course information | The lecturer can post messages and manage who can see archived items |
| UC 4 - Create course | The lecturer and/or the administrator can create new courses and recreate courses in the system |
| UC 5 - Manage grades | Lecturers can update, insert and calculate grades in the system. |
| UC 6 - Manage space size | Maintainers allocate space for a course and limit the size of files lecturers and students can upload. |
| UC 7 - Create teams | Students and lecturers can create teams for students to collaborate and share files in the system. |
| UC 8 - Send messages | Lecturers and students can send messages through mail or the messaging system to other participants |

| ID | QUALITY ATTRIBUTES | SCENARIO | ASSOCIATED USE CASE |
|---|---|---|---|
| QA-1 | Usability | Lectures are able to create and manage courses, grades, messages and create teams. | UC-4<br>UC-5<br>UC-7<br>UC-8 |

| | | | |
|---|---|---|---|
| QA -2 | Maintainability | Administrators and Lecturers are able to recreate and modify courses. | UC-2<br>UC-3<br>UC-4<br>UC-5<br>UC-7<br>UC-8 |
| QA -3 | Reliability | Program will recover to its previous state in 30 seconds if there is an error. | All |
| QA -4 | Security | A user performs a change in the system during normal operation. It is possible to know who performed the change in the system. | All |

| **ID** | **CONSTRAINTS** |
|---|---|
| CON-1 | The system must be accessed through a web browser (Chrome, Firefox, etc) in different platforms ( Windows, Linux) |
| CON-2 | Events from the previous school term must be stored. |
| CON-3 | Up to 100 users must be supported. |
| CON-4 | Users network should be in proper conditions to use system |
| CON-5 | Database for the system must not be used for any other purpose |

| **ID** | **CONCERNS** |
|---|---|
| CRN-1 | The system must validate user input to make sure it is of the correct type |
| CRN-2 | The codebase must stay organized and well commented for future development |
| CRN-3 | The system must handle and log exceptions and errors to facilitate troubleshooting |
| CRN-4 | The system must be fast and responsive despite a large amount of users being logged in at once |

**Iteration 2:**

***Step 1: Review Inputs***

| **Category** | **Details** |
|---|---|
| Design Purpose | To produce a sufficiently detailed design to support the construction of |

| | |
|---|---|
| | the system |
| Primary Functional Requirements | UC-1 Subscribe and unsubscribe<br>UC-2 Managing Dynamic Course Information<br>UC-3 Managing Static Course Information |
| Quality Attributes | QA-1 Usability<br>QA-2 Modifiability |

Iteration 2: Identifying structures to support primary functionality

***Step 2: Establish Iteration Goal by Selecting Drivers***

Select drivers that address the general architectural concern of identifying structures to support primary functionality.

- UC-1: Subscribe and Unsubscribe
- UC-2: Manage static course information
- UC-3: Manage dynamic course information

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

We want to refine the modules located in different layers defined by the reference architectures from iteration 1.

- We want to refine the Data Access Modules.
- We want to refine the Helpers and Utilities module.

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

| **Design Decisions and Location** | **Rationale and Assumptions** |
|---|---|
| Decompose the elements in iteration 1 into domain specific components. | Before starting functional decomposition, it is necessary to have an initial domain model of the system. Identification of the major entities based on the Use Cases using a partial domain model will help. |
| Choose framework: Spring and Hibernate | These frameworks will help with interfacing and database relations. |

***Step 5: Instantiate Architectural Elements, Allocate Responsibilities, Define Interface***

| **Design Decision** | **Rationale Assumptions** |
|---|---|
| Create only an initial domain model from the elements in the rich client reference model into domain specific modules with explicit interfaces. | It shows the entities that participate in the primary use cases and helps accelerate design process as you would not have to create one |

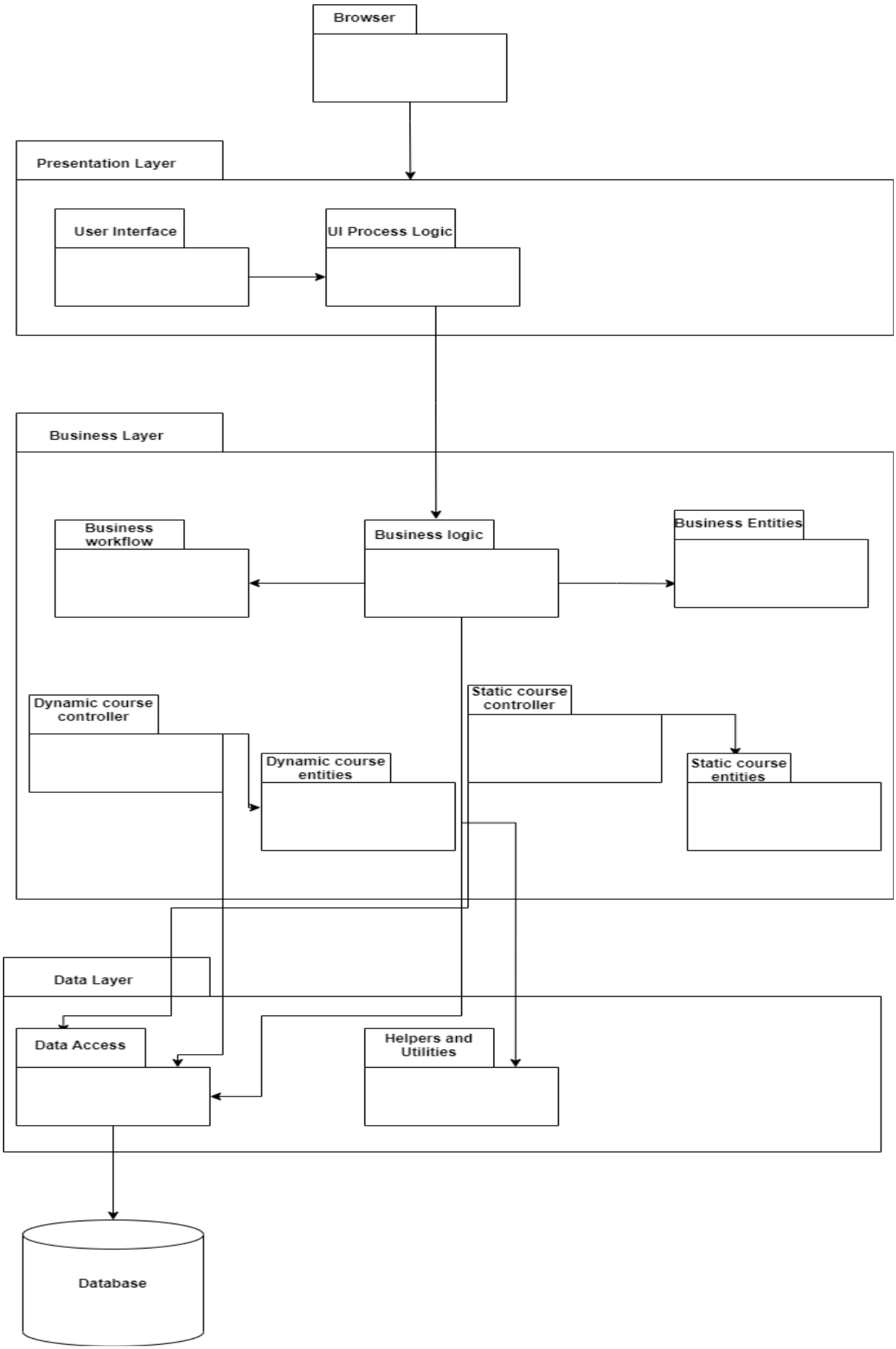
***Step 6: Sketch Views and Record Design Decisions***

Below are the Use Cases highlighting some key entities

| **Use Case** | **Description** |
|---|---|
| UC-1 Subscribe and Unsubscribe | Students subscribe or unsubscribe from a course or exam if they meet the requirements. |
| UC-2 Managing Dynamic Course Information | Lecturer can post messages and manage who can see archived items |
| UC-3 Managing Static Course Information | The administrator updates the lectures and modules of the course |

This is the modified version of the Rich Client architecture based on the decision made in step 5.
The table below lists key domain specific elements and whether they will be components or entities.

| **Element** | **Domain Component** | **Domain Entity** |
|---|---|---|
| Messages | | Property of dynamic course component |
| Lectures | | Property of static course component |
| Modules | | Property of static course component |
| Course | | Property of static course component |
| Archived items | | Property of dynamic course component |

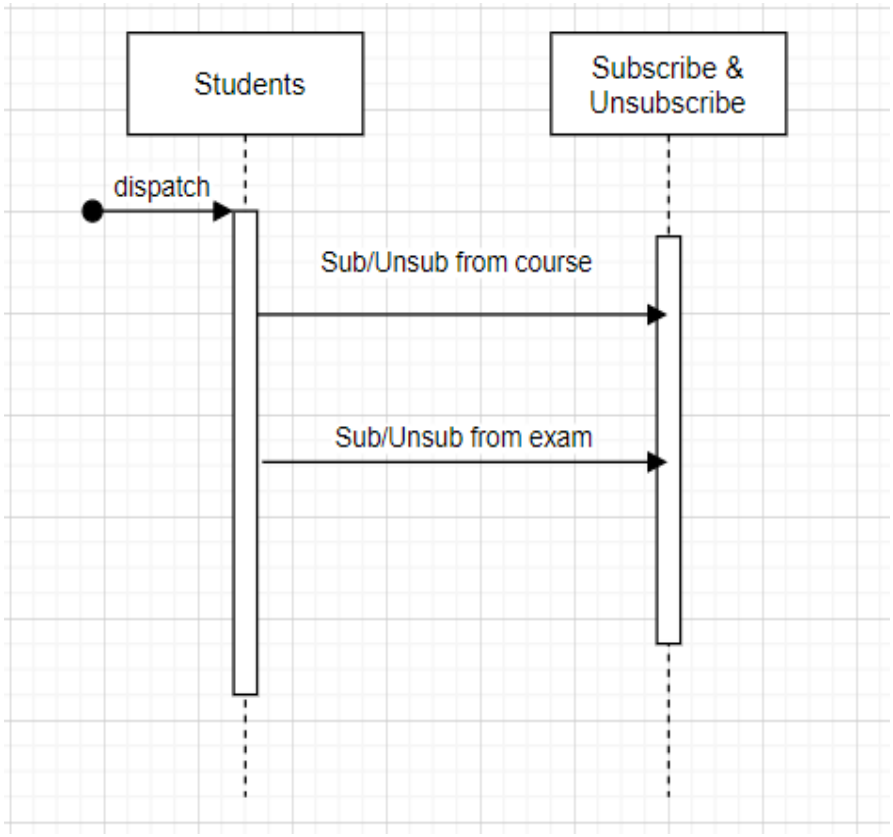

Browser
Presentation Layer
User Interface
UI Process Logic
Business Layer
Business workflow
Business logic
Business Entities
Dynamic course controller
Static course controller
Dynamic course entities
Static course entities
Data Layer
Data Access
Helpers and Utilities
Database

| 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. |
| Dynamic Course Controller | Responsible for providing the necessary information to the presentation layer, displaying dynamic course interactions in the system. |
| Static Course Controller | Responsible for providing the necessary information to the presentation layer, displaying static course representation in the system. |
| Dynamic course entities | Contains domain specific objects that represent messaging in the system. |
| Static course entities | Contains domain specific objects that represent lectures, courses and modules in the system. |
| Data Access | 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). |
| 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 |

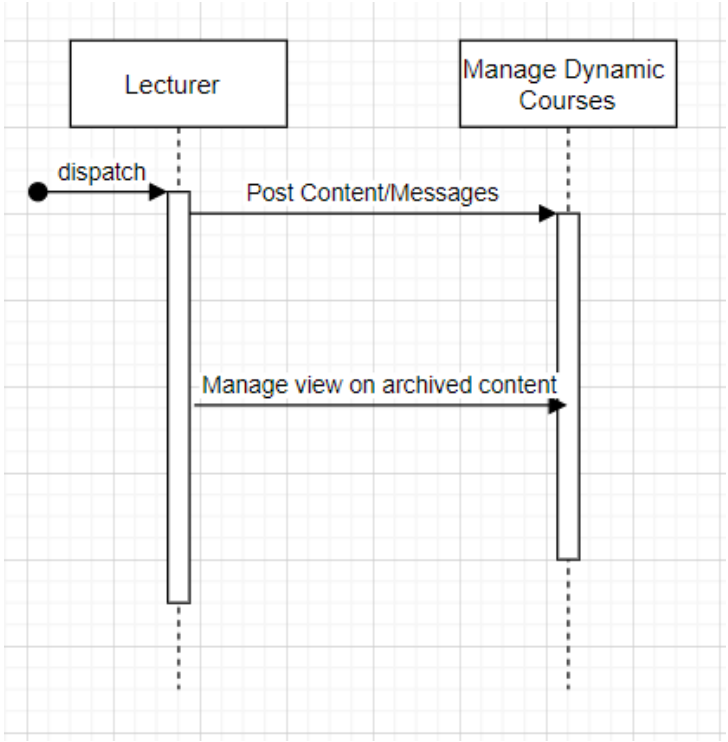
| | |
|---|---|
| | domain and their associated business logic. |

**Sequence Diagram**

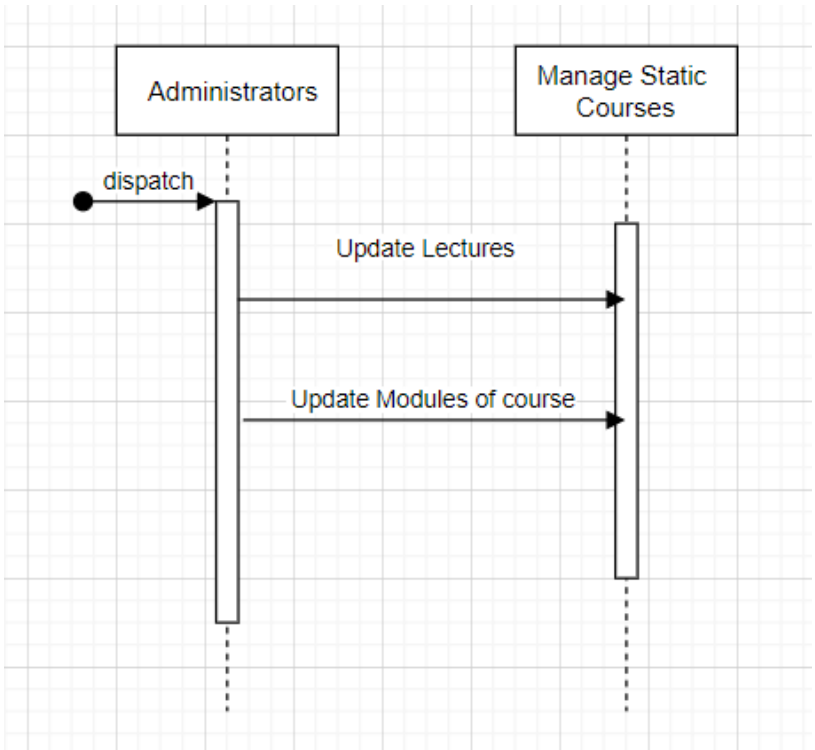
*UC-1 Subscribe and Unsubscribe*



*UC-2 Managing Dynamic Course Information*




*UC-3 Managing Static Course Information*



**Interfaces**

| **Method Name** | **Description** |
|---|---|
| **Element:** Dynamic course Controller | |
| session_start() | Commences Login session for student and admin |
| session_destroy() | Ends session |

***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-1 | |
| | | UC-2 | |
| | | UC-3 | |
| | | | |