Learning Management System

**Software Architecture Document**

**Version 1.0**

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 27/11/2024 | 1.0 | Final version | Nguyễn Thị Hiền Anh Trần Thị Kim Ngân Vũ Thị Liên Nguyễn Duy Thùy Dương Đào Quang Khánh Nguyễn Quang Dũng Tạ Thị Tuyết Lan |

**Table of Contents**

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 5

4. Use-Case View 5

5. Logical View 6

5.1 Overview 6

5.2 Architecturally Significant Design Packages 7

5.3 Use-Case Realizations 8

6. Process View 18

7. Deployment View 18

8. Implementation View 18

8.1 Overview 19

8.2 Layers 19

9. Data View (optional) 19

10. Size and Performance 19

11. Quality 19

**Software Architecture Document**

# **Introduction**

## **Purpose**

This document offers a thorough architectural overview of the system, showing various system components utilizing a variety of architectural viewpoints. Capturing and communicating the important architectural choices made for the system is its goal.

## **Scope**

This document is implemented on the Learning Management System, which is developed by group 10

## **Definitions, Acronyms, and Abbreviations**

User: People use the system, such as students and lecturers

Admin: People manage the whole system

## **References**

N/A

## **Overview**

The Software Architecture Document is organized into multiple sections:

* Introduction: This section outlines the document's purpose and scope, defines relevant terms, acronyms, and abbreviations, and includes references and an overview.
* Architectural Representation: This part explains how the software architecture of the Learning Managements System is structured and represented.
* Architecture Goals and Constraints: This section highlights the factors influencing the system's architecture. It also includes the Use-case View, which details the system's use cases.
* Additional Sections: Other key parts of the document include the Logical View, Process View, Deployment View, Implementation View, as well as considerations for Size and Performance and Quality.

# **Architectural Representation**

The views essential for presenting the architecture in this project are the Use-Case View, Logical View, Deployment View, and Data View. These are illustrated using Unified Modeling Language (UML) diagrams created on StarUML, WhiteStarUML and draw.io

The **Use-Case View** focuses on identifying key user scenarios and their interactions with the system. It highlights the most important system features through use cases and scenarios, making it particularly relevant to end users and stakeholders. Artifacts associated with this view include the Use-Case Model, Analysis Model, and Use-Case-Realization documents.

The **Logical View** addresses the system’s functional requirements by presenting the object model of the design, including the application’s major components, packages, and classes, along with their relationships. This view is targeted at programmers and designers, with the Design Model as its primary artifact.

The **Deployment View** illustrates the physical infrastructure where the application will be deployed, focusing on the mapping of software to hardware and the system’s distribution topology. It provides critical insights for deployment managers and system administrators, supported by the Deployment Model artifact.

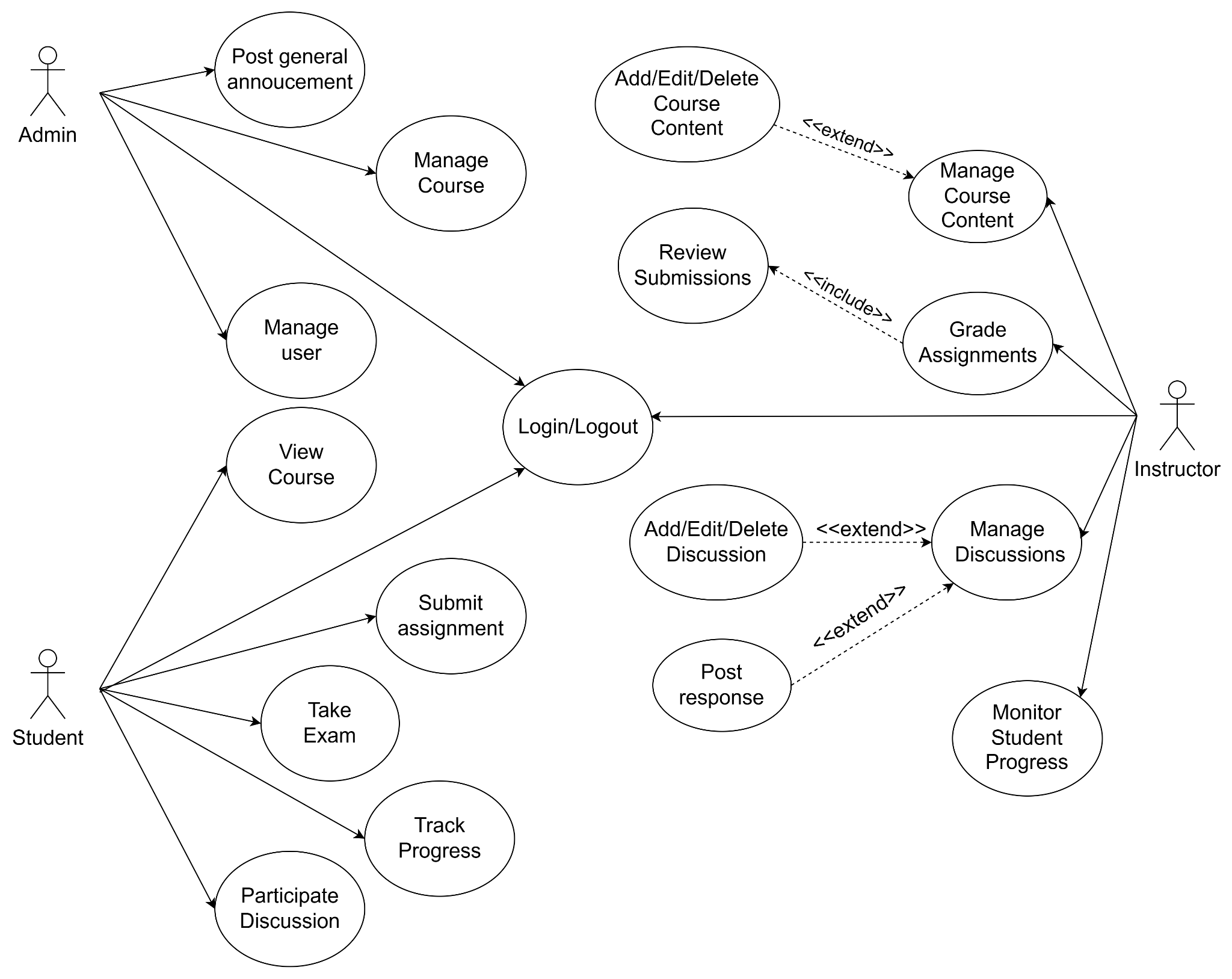
Lastly, the **Data View** defines the enduring, architecturally significant components of the data model, ensuring data persistence. It is intended for data specialists and database administrators, with the Data Model serving as its primary artifact.

# **Architectural Goals and Constraints**

The system should be designed to support the management of academic tasks for a university department, with the ability to handle a large user load effectively. It should be structured to simplify administrative procedures and allow clear role-based access to reduce complexity. The system must ensure high availability and minimal downtime, providing users with uninterrupted access. The user interface should be intuitive and easy to navigate, allowing users to quickly find books and perform transactions. Additionally, the system must be optimized for fast loading times and responsiveness to enhance the user experience, reducing latency during page loading and book searches.

The system will follow the MVC design pattern to ensure clarity and maintainability, though it will introduce a layered structure. Performance and scalability requirements will be defined later based on usage patterns. The project must remain cost-effective, considering technology selection, infrastructure, and maintenance without sacrificing quality. It must work within technology constraints, such as server capacity, bandwidth, and compatibility with various devices and browsers.

# **Use-Case View**

**

# **Logical View**

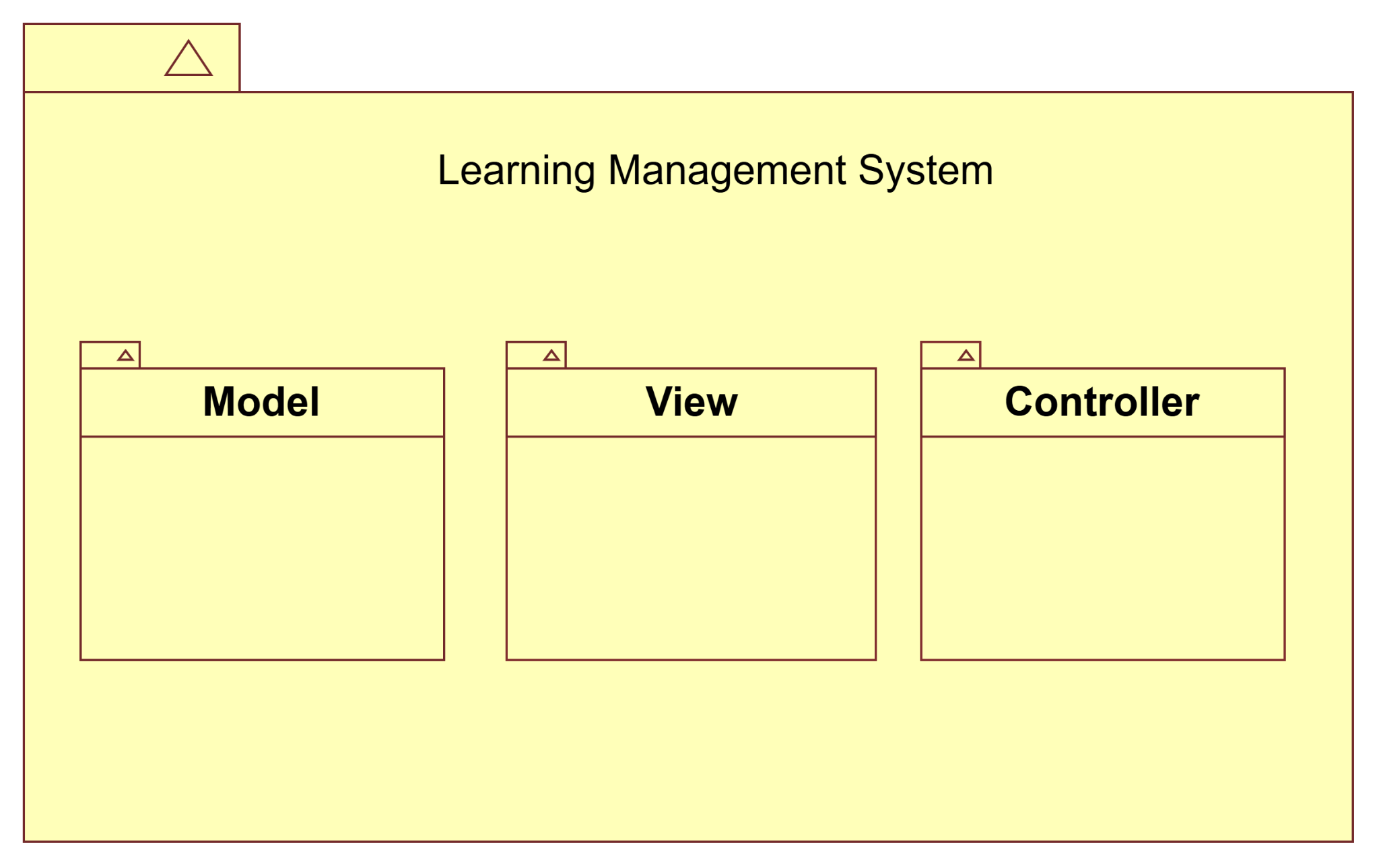
## **Overview**

An explanation of the architecture's logical perspective explains how the design model is broken down into layers and package hierarchy overall. Three major packages make up the logical view of the Learning Management System:

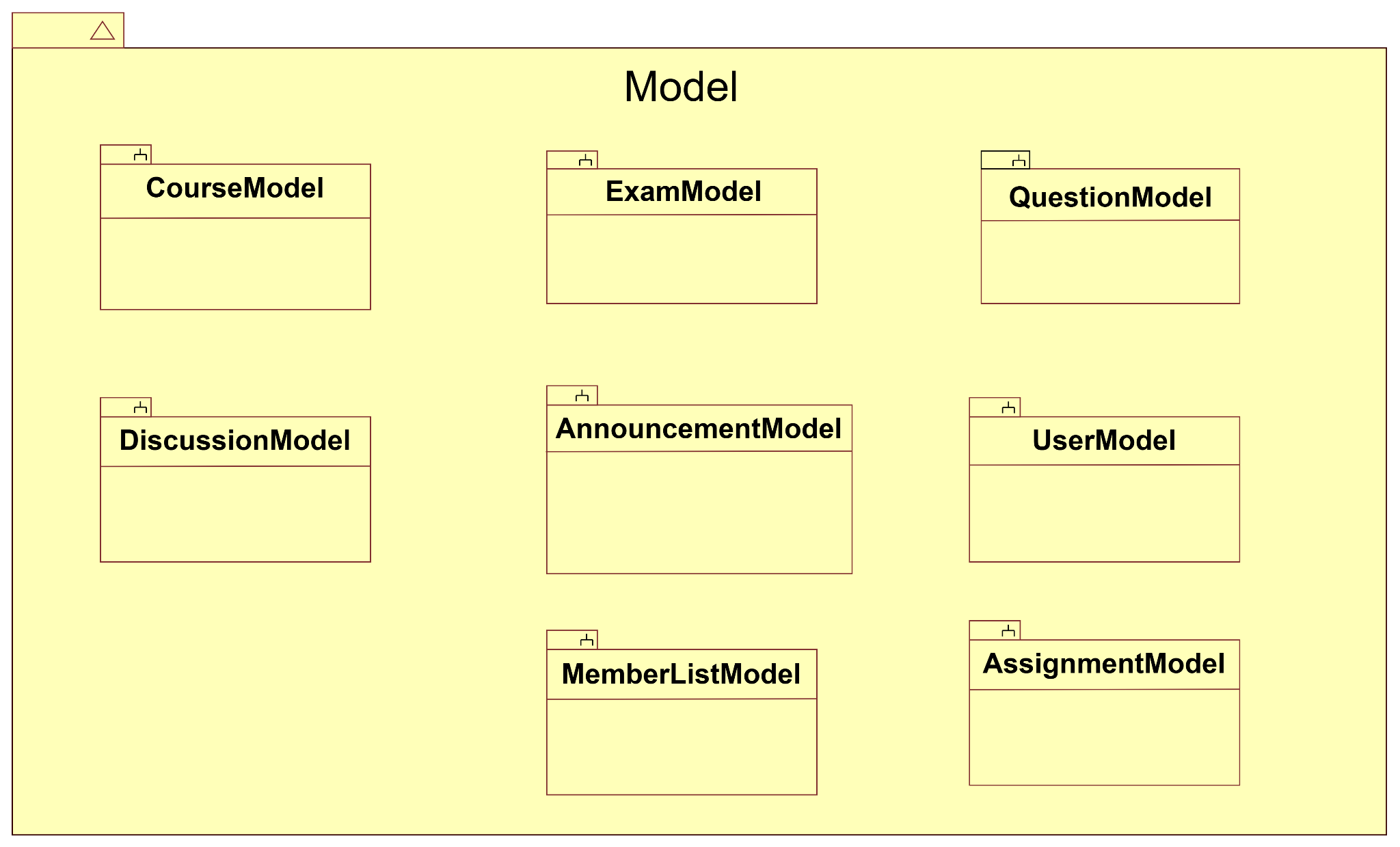
Model is a collection of classes that directly maintains and shows the LMS’s data, logic, and rules in the view.

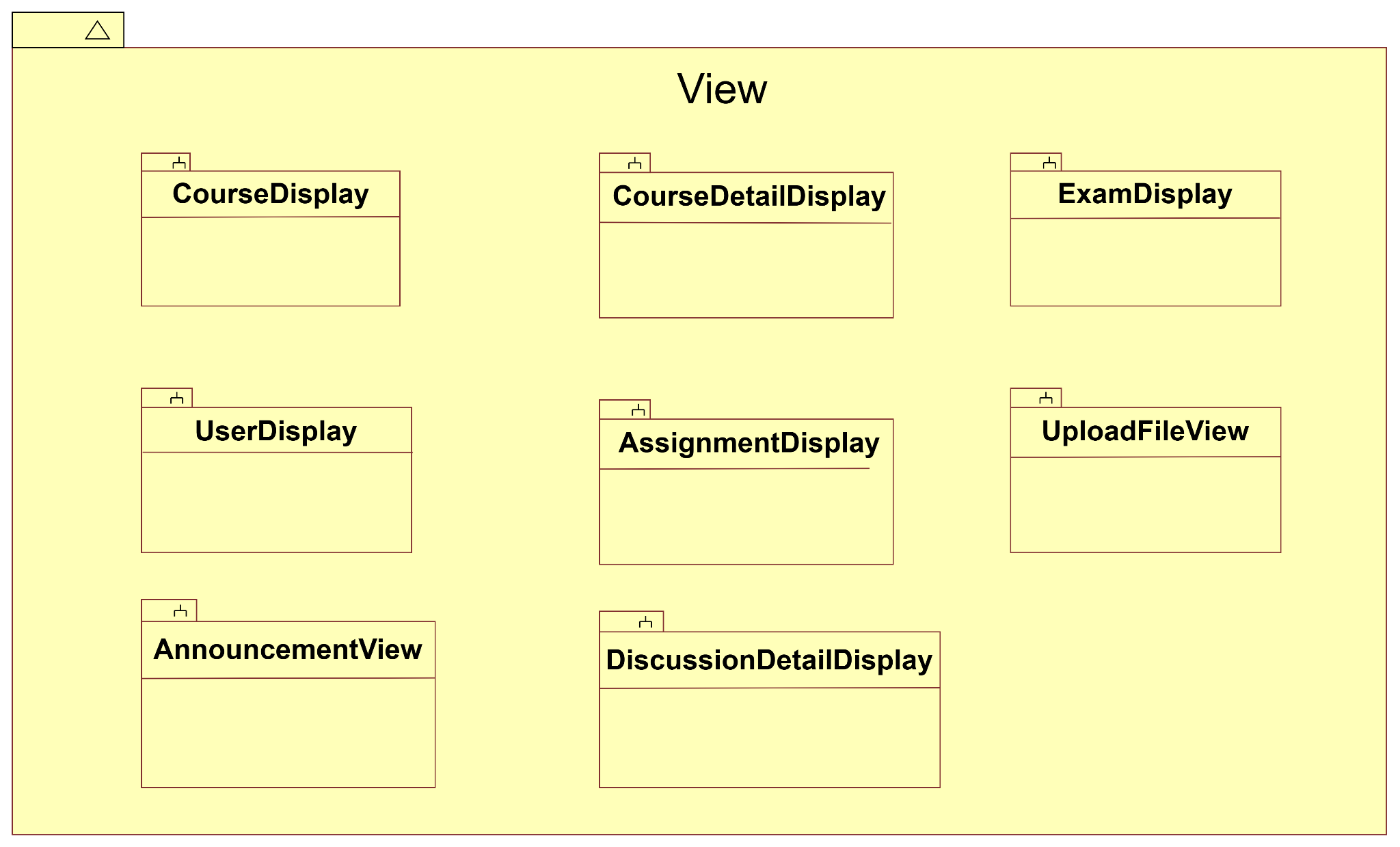
Views are classes that, in response to modifications in the model, provide output representations of information for the users.

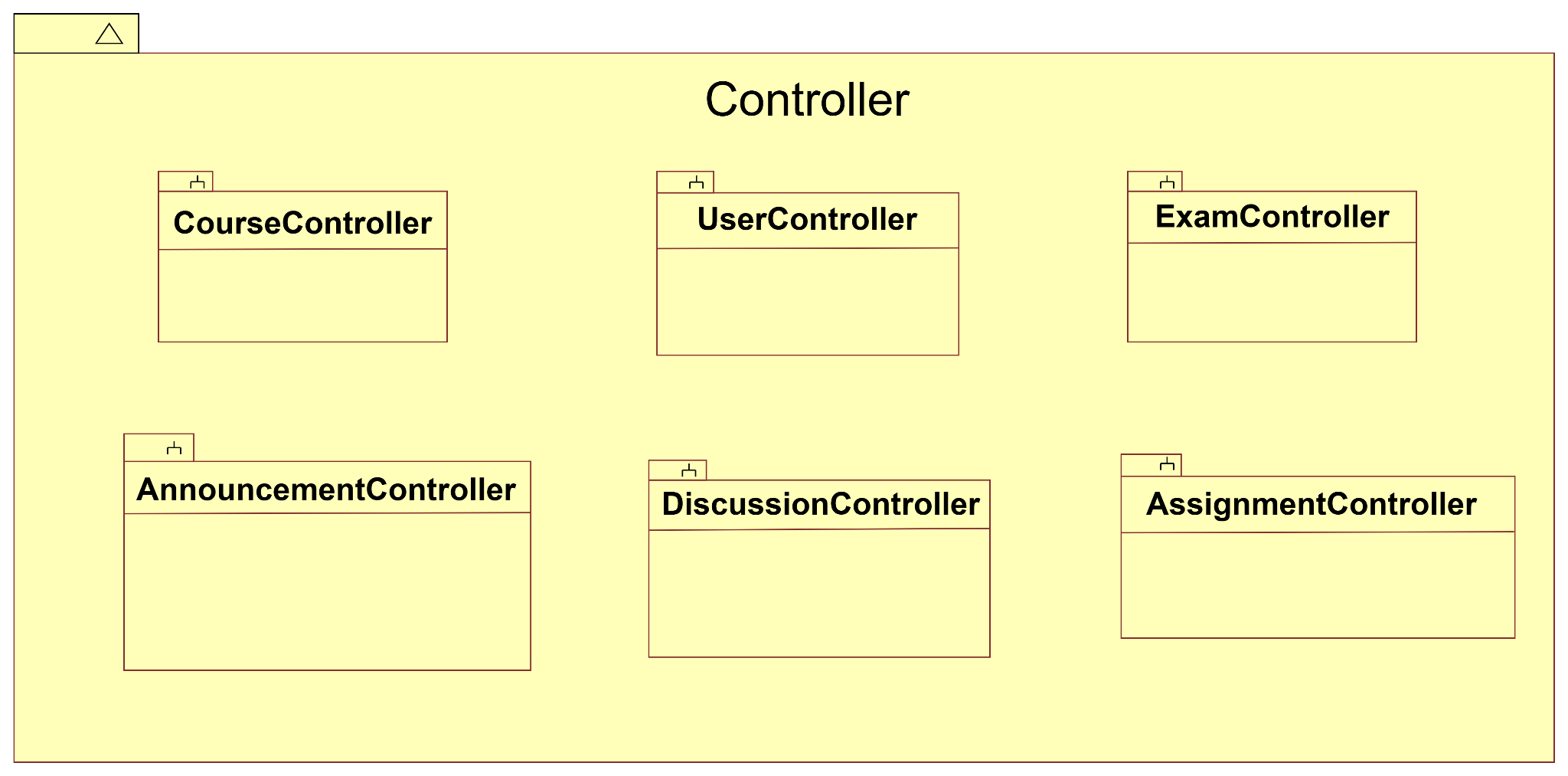
Controller has classes that can communicate with the model to update its state and can also provide data to its associated view to modify how the model is shown.



## **Architecturally Significant Design Packages**

**

**

**

## **Use-Case Realizations**

* + 1. **Submit Assignments**

| Use-case Name | Submit Assignments | |
| --- | --- | --- |
| Scenario | Submit assignments through the LMS. | |
| Brief Description | This use case enables students to submit assignments through the LMS. | |
| Actors | Users: Students | |
| Related Use cases | None | |
| Precondition | User connects to Internet successfully | |
| Post Conditions | A confirmation email is sent to both the student and the instructor. | |
| Flow of Events | Basic Flow | After logging in and completing the assignment, the user clicks "Add Assignment", the system displays a file upload dialog box where the user can select and upload their assignment file. After the user uploads an assignment, The system provides two options including “Submit” and “Cancel”. If the user selects “Submit”, the system validates the file, there will be a notification confirming the user's successful submission and updating the status "Done: Receive a grade" on the assignment details page or in the list of assignments. If the user selects “Cancel”, the system discards the uploaded file and redirects the user back to the assignment upload page without making any changes |
| Alternative Flow | None |
| Exception Condition | None | |

* + 1. **Take Exam**

| Use-case Name | Take Exam | |
| --- | --- | --- |
| Scenario | Students take an online exam test to get their midterm or final grade | |
| Brief Description | "Take Exam" use-case describes how the users access, take the online exam test and submit their work | |
| Actors | Users: Students | |
| Related Use cases | View Course | |
| Precondition | None | |
| Post Conditions | None | |
| Flow of Events | Basic Flow | In a course, users need to click on the test section that the teacher assigned, enter the password that the teacher has provided when making the test, and take the test within the period that the teacher has created. Students can choose the answer options, fill in the blanks, and mark a question with a flag. Then, students can click submit and confirm submission, or the system will automatically submit after the time is up. Students can see the test results depending on the teacher's settings after completing the test. |
| Alternative Flow | None |
| Exception Condition | None | |

**5.3.3. Manage Courses Content**

| Use-case Name | Manage Courses Content | |
| --- | --- | --- |
| Scenario | The Manage Courses Content use case allows instructors to effectively manage the structure and content of their courses within the Learning Management System (LMS). This includes edit course settings, add new modules, add new sections (Activities/Resources), delete modules/sections, organize course structure. The use case ensures that instructors can keep course content up to date and structured in a way that supports student engagement and learning outcomes. Administrators may also interact with this use case in relation to course creation and role management. | |
| Brief Description | The Manage Courses Content use case enables instructors to create, modify, and organize course materials. It involves the following key actions:  Adding new course modules: Creating new sections or units for the course.  Uploading course materials: Adding documents, videos, links, and other resources.  Editing existing content: Modifying or updating course details, resources, and assignments.  Deleting or removing outdated content: Removing old materials that are no longer relevant.  Organizing course structure: Reordering course modules and sections to improve the flow of content.  Managing activities: Creating assignments, quizzes, and forums, as well as configuring access and grading. | |
| Actors | Users: Administrators, and instructors  **Instructors** are the primary users responsible for managing course content, uploading materials, creating assignments, and ensuring that the course remains up to date.  **Administrators** have a role in managing course creation, enrolling instructors, and ensuring that courses are properly structured within the LMS. | |
| Related Use cases | This use case stands independently in relation to the others in terms of managing the content of the course itself. | |
| Precondition | Instructor is Logged In:The instructor must be logged into the LMS with appropriate credentials to access course content management functionality. **Course is Created and assigned to instructor:** A course must already exist and must be assigned to instructor in the system by admin before any course content can be uploaded or activities created. | |
| Post Conditions | Course Content is Updated: Once the instructor uploads or edits course materials, the course content is updated and made immediately available to students. **Students Can Access Updated Content:** After the course materials and activities are updated, the students can access the new content based on the instructor’s configured access settings. | |
| Flow of Events | Basic Flow | **Login and Access the Course**: The instructor logs into the LMS and navigates to the course management section.  **Enter Edit Mode**: The instructor clicks on the "Edit Mode" button to activate course editing functions.  **Add New Modules**: The instructor adds new modules to the course by selecting "Add Module" and providing relevant details (name, description, and materials).  **Edit Existing Modules**: The instructor selects an existing module to edit its content, such as updating files, changing descriptions, or modifying grading criteria.  **Upload Course Materials**: The instructor uploads documents, videos, links, or other resources into the course structure.  **Reorganize Content**: The instructor may use drag-and-drop functionality to rearrange the order of modules or resources to ensure a logical flow of content.  **Save Changes**: Once the course content is updated, the instructor saves the changes, which are reflected immediately in the course structure.  **Exit Edit Mode**: The instructor exits "Edit Mode," and the updated course content is now available to students. |
| Alternative Flow | None |
| Exception Condition | None | |

**5.3.4. Manage Announcements**

| Use-case Name | Manage Announcement | |
| --- | --- | --- |
| Scenario | The Manage Announcement use case allows instructors and administrators to create, edit, and delete announcements within the LMS. Instructors can input details like title, content, and target audience, with the system validating and saving changes. Notifications are sent to the intended audience for new or updated announcements. Administrators can manage announcements at a broader level. The system ensures effective communication while providing feedback for errors or permission issues. | |
| Brief Description | This use case involves managing announcements in the LMS, including creating, editing, and deleting announcements. It ensures smooth communication between instructors, administrators, and students by sending notifications and managing the visibility of announcements. | |
| Actors | Students:   * View announcements. * Receive notifications for new or updated announcements.   Instructors:   * Create, edit, and delete announcements. * Send notifications to the relevant audience. | |
| Related Use cases | View Announcements | |
| Precondition | User connects to the Internet successfully and has the appropriate role (instructor or student). | |
| Post Conditions | Users must be logged into the LMS with instructor or administrator privileges.  System must be connected to a stable database. | |
| Flow of Events | Basic Flow | 1. The instructor or administrator logs into the LMS and accesses the Announcement Page.  2. The system displays a Summary Panel showing the total number of announcements, the most recent announcement, and an option to add a new announcement.  3. The user clicks the "Add New Announcement" button, which opens a form to input the title, content, and other details.  4. After submitting valid announcement details, the system validates, saves the announcement, and sends notifications to the intended audience.  5. The announcement is added to the list, and the Summary Panel is updated to reflect the new announcement.  6. Users can click on existing announcements to edit or delete them.  7. If editing, the user makes the necessary changes, and the system updates the announcement after validation.  8. If deleting, the system confirms the action and updates the list accordingly. |
| Alternative Flow | None |
| Exception Condition | None | |

**5.3.5. Manage Users**

| Use-case Name | Manage Users | |
| --- | --- | --- |
| Scenario | The Manage Users use case allows administrators to create, view, edit, and delete user's information including instructors and students in the LMS. | |
| Brief Description | This business use case describes how an admin manages the users on the LMS, including adding new users, viewing user information, editing user information, and deleting users. | |
| Actors | Administrators | |
| Related Use cases | None | |
| Precondition | The administrator must be logged in to the system as an authorized user. | |
| Post Conditions | The system saves any changes made to the user's information and updates the database. | |
| Flow of Events | Basic Flow | To add a new user to the system, the administrator clicks on the "plus" icon in the top right corner of the existing user list. Then, the "Add New User" box will appear on the system interface, and the administrator fills in all the information the user wants to add, especially the required information marked with a "red star". After filling in all the information, the administrator clicks on the "Save" located in the top right corner of the "Add New User" box screen to save the new user information. The notification box that the new user has been successfully added "The User Account has been added successfully" appears, and the administrator clicks "OK" to return to the user list interface.  The administrator clicks on the "..." icon at the end of the user's information row. A menu will appear with 3 items: "View", "Edit" and "Delete", the administrator clicks on the "View" button to view the detailed information of that user. Then, the "View User Information" box appears and the detailed information of the selected user will be displayed on the screen. After viewing, the administrator clicks the "X" button in the top left corner of the "View User Information" box to return to the user list interface.  The administrator clicks on the "..." icon at the end of the user's information row. A menu will appear with 3 items: "View", "Edit" and "Delete", the administrator clicks on the "Edit" button to edit the detailed information of that user. Then, the "Edit User Information" box appears and the detailed information of the selected user to edit will be displayed on the screen. At the end of each information box about 'FirstName', 'LastName', 'ID', 'Email', 'Phone Number', 'Address' and 'Description', there is a "Pen on Paper" icon, after the administrator clicks on that icon located on the information box that needs to be edited, the interface will display the "Change Information" box so that the administrator can edit the content of that information. After completing the edit information, the administrator clicks on the "Save" button located on the top right corner of the "Edit User Information" box to save the information that has just been edited and the screen will display the box "The User's Information has been edited successfully" to notify that the administrator has completed editing the user information, click the "OK" button to return to the user list.  The administrator clicks on the "..." icon at the end of the user's information row. A menu will appear with 3 items: "View", "Edit" and "Delete", the administrator clicks on the "Delete" button to delete that user. Then the screen will display the box "Are you sure you want to delete this User and all related information?" to confirm that the administrator is sure wants to delete this user and all information related to that person on the system. To delete the user, the administrator clicks on the red "Delete" button in the lower right corner of the box "Are you sure you want to delete this User and all related information?" or to cancel the previous operation, the administrator clicks on the gray "Cancel" button in the left corner of the box and returns to the user list screen. In case the administrator has completed deleting the user and all related information the screen will display the box "The User Account and all related information has been deleted successfully" to notify that the administrator has completed deleting the user information, click the "OK" button to return to the user list. |
| Alternative Flow | If there are no existing students, the system displays a message indicating that there are no students to manage. The administrator can choose to add a new student |
| Exception Condition | None | |

**5.3.6. Manage Course**

| Use-case Name | Manage Course | |
| --- | --- | --- |
| Scenario | The administrator manages the courses on the system. | |
| Brief Description | The administrator manages the courses, including creating courses, editing course information, or deleting courses. The admin only manages the course information and member lists of the course, including instructors and students. The admin has no right to adjust the content of the course. | |
| Actors | Users: Administrator. | |
| Related Use cases | None | |
| Precondition | The administrator logged in successfully. | |
| Post Conditions | None | |
| Flow of Events | Basic Flow | * Admins can manage courses through a view that includes a list of courses with filters, arranger and search bar. * Clicking "Create new course" prompts a form to enter several course details like name, ID, description,..., and assign members (instructors and students). Members gain access to the course based on their role. Once the form is complete, the admin can click "Save" to create the course. * Selecting "Edit" brings up a form to modify course details. Admins can update basic information and adjust the member list by adding or removing members. Clicking "Save" updates the course. * Choosing "Delete" prompts an alert if not able to or a confirmation message. If the admin selects "Yes," the course is deleted with a success notification; if "No," the admin is returned to the previous view, and the course remains. |
| Alternative Flow | None |
| Exception Condition | None | |

**5.3.7. Participate in Discussions**

| Use-case Name | Participate in Discussions | |
| --- | --- | --- |
| Scenario | Participate in Discussions use case allow students to create, post response, and view discussions within the course in LMS. Students can input details such as subject, message and also answer into discussion section | |
| Brief Description | This use case involves joining in discussion of a course in LMS, including creating, posting responses, and viewing other discussions. It ensure a seamless communication between lecturers, students in order to solve student’s problems | |
| Actors | Users: Students | |
| Related Use cases | None | |
| Precondition | Students logging in successfully | |
| Post Conditions | None | |
| Flow of Events | Basic Flow | Students can access their courses via the “Courses” button in the left navigation bar and select a course. In the course’s “Discussion” section, clicking the triangle reveals available discussions.  To join a topic, click “Join,” then participate by selecting a discussion, typing a response, and clicking “Post response.” Use “Back to discussion” to return.  To start a new discussion, click “Start discussion,” fill in the details, and click “Post discussion” or “Cancel” to exit.  Students can highlight discussions with the three dots, navigate with “Next discussion,” or return to the dashboard via “Back to dashboard.” |
| Alternative Flow | None |
| Exception Condition | None | |

# **Size and Performance**

The major sizing characteristics and performance constraints for the LMS are as follows:

The LMS should support up to 3000 active users concurrently, considering a growing number of students, instructors, and administrators. The system should be scalable to support up to 5000 users as the number of users increases over time. The response time for any user interaction should not exceed 3 seconds. The system should provide an optimal experience by reducing latency during course access, grading, and other key functions.

# **Quality**

The LMS's user interface needs to be intuitive and simple enough for students, teachers, and administrators who may not be highly technical. It should require minimal training for users to navigate and complete tasks. The LMS should include an online help feature with detailed instructions on system usage and FAQs. The help section should also include definitions of academic terms and acronyms to assist users in understanding course content and navigation.