UML Design and Modeling of an E-learning System

1. **Objective**

In an increasingly connected world, online learning (e-learning) platforms play a vital role in the dissemination of knowledge and access to education. They allow learners and trainers to interact and collaborate regardless of geographical or temporal constraints.

The objective of this document is to present the design and UML modeling of an e-learning system. Through a detailed analysis of user needs, key functionalities and interactions, we provide a clear and structured representation of the envisaged system.

To meet the requirements of this type of platform, we use UML diagrams such as use case, class and sequence diagrams, which provide a solid basis for the technical development of the project.

This document is intended for project stakeholders, including developers, analysts and decision makers, in order to ensure a common understanding of the objectives and functionalities of the system.

1. **Needs Analysis**

**Functional Needs**

Functional needs define the main and specific functionalities of the e-learning system. They include:

User management:

* Registration, login, and logout of users (administrators, teachers, students).
* Management of user profiles (modification of personal information and passwords).

Management of courses and resources:

* Creation, modification, and deletion of courses by administrators and teachers.
* Assignment of courses to teachers and associated resources (texts, files, videos, etc.).

Management of assessments:

* Creation and publication of tests, quizzes, and assignments.
* Consultation of assessment results by students.

Communication and collaboration:

* Setting up discussion forums, workshops, and messaging to encourage interaction.
* Notifications to inform users of important updates.

Access to educational content:

* Consultation of available courses, schedules, and resources.
* Access to course histories for students.

**Non-Functional Requirements**

Non-functional requirements ensure the quality, performance, and security of the system. They include:

Accessibility:

* Compatibility with different devices (computers, tablets, smartphones).
* Compliance with accessibility standards for people with disabilities.

Performance:

* Optimal response time during navigation and user actions.
* Ability to manage a large number of users simultaneously.

Security:

* Protection of users' personal data.
* Access control via roles (administrator, teacher, student).

Maintainability:

* Well-structured and documented code to facilitate future updates.
* Implementation of regular tests to detect and correct possible errors.

Reliability:

* System available 24/7 with minimal failures or interruptions.

1. **Use Case diagrames**

The Use Case diagram that describes the rights granted to administrators

is shown on **Fig. 1**. Administrators have more rights in comparison with the other users. Except the rights related with the administration (for example user groups management, course management etc.) administrators has also the rights of the lecturers and students. This is the reason that the Use Case diagram of the administrator is the most complex in the developed model.

On **Fig. 2** is shown the Use Case diagram that describes the rights granted

to lecturers/instructors. This user type has all rights needed for course

management (publishing course structure), resources management (publishing resources and associating them to a course item) and course preview.

The Use Case diagram that describes the rights granted to students is

shown on Fig. 2. They are related with user profile management, access to

specific courses etc.

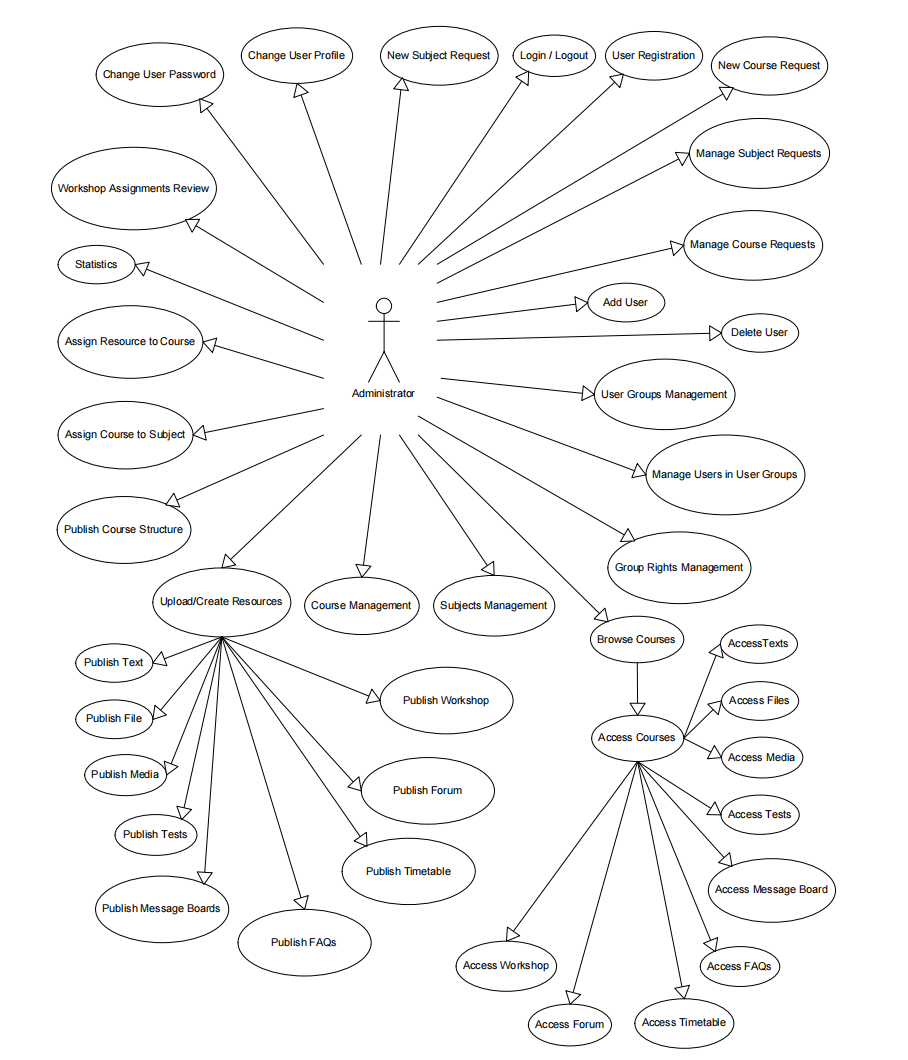


Fig.1. Use Case diagram of administrator's functions

#### ****1. User Management****

* **Change User Password**: Reset or update a user's password.
* **Change User Profile**: Modify user details such as name, contact, or roles.
* **Add User**: Add new users to the system.
* **Delete User**: Remove users from the system.
* **User Groups Management**: Organize users into groups for better access control.
* **Manage Users in User Groups**: Assign or remove users from specific groups.
* **Group Rights Management**: Define access rights and permissions for user groups.

#### ****2. Course and Subject Management****

* **New Course Request**: Approve or reject requests for creating new courses.
* **Manage Course Requests**: Oversee all course creation requests.
* **Assign Course to Subject**: Link a course to its respective subject.
* **Subjects Management**: Add, modify, or delete subjects in the system.
* **Course Management**: Oversee all course-related activities, including updates and deletions.
* **Publish Course Structure**: Organize and release the structure of courses.

#### ****3. Resource Management****

* **Upload/Create Resources**: Add course content such as files, media, tests, and text.
* **Publish Text**: Make textual materials available to students.
* **Publish File**: Upload files like PDFs, presentations, or assignments.
* **Publish Media**: Add multimedia resources such as videos or audio files.
* **Publish Tests**: Publish quizzes, exams, or assignments for students.
* **Publish Message Boards**: Create forums or boards for discussions.
* **Publish FAQs**: Provide frequently asked questions for user assistance.

#### ****4. Access Management****

* **Access Courses**: Grant access to specific courses.
* **Access Texts, Files, Media, Tests**: Enable users to view or interact with various types of course resources.
* **Access Message Board**: Allow users to participate in discussions.
* **Access FAQs**: Let users browse the FAQ section for guidance.

#### ****5. Workshop and Forum Management****

* **Publish Workshop**: Create and manage online workshops for students.
* **Access Workshop**: Provide access to workshop sessions.
* **Publish Forum**: Enable the creation of discussion forums.
* **Access Forum**: Allow users to participate in forum discussions.

#### ****6. Timetable Management****

* **Publish Timetable**: Create and share schedules for courses, exams, or workshops.
* **Access Timetable**: Enable users to view the published timetable.

#### ****7. Statistical Analysis****

* **Statistics**: View or generate reports related to system usage, course performance, and user activity.

#### ****8. Requests and Approvals****

* **New Subject Request**: Approve or reject requests for adding new subjects.
* **Workshop Assignments Review**: Review and approve assignments submitted for workshops.

#### ****9. Login/Logout****

* Basic functionality for system access and security.

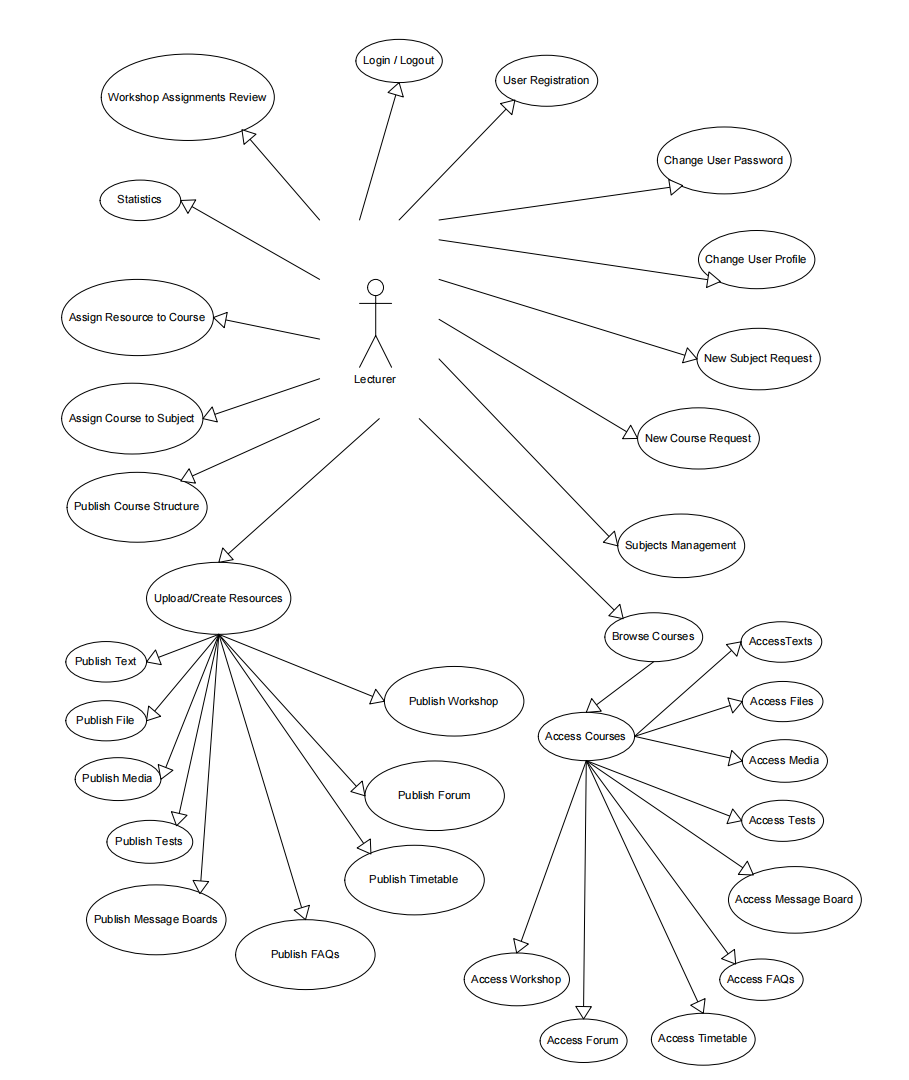
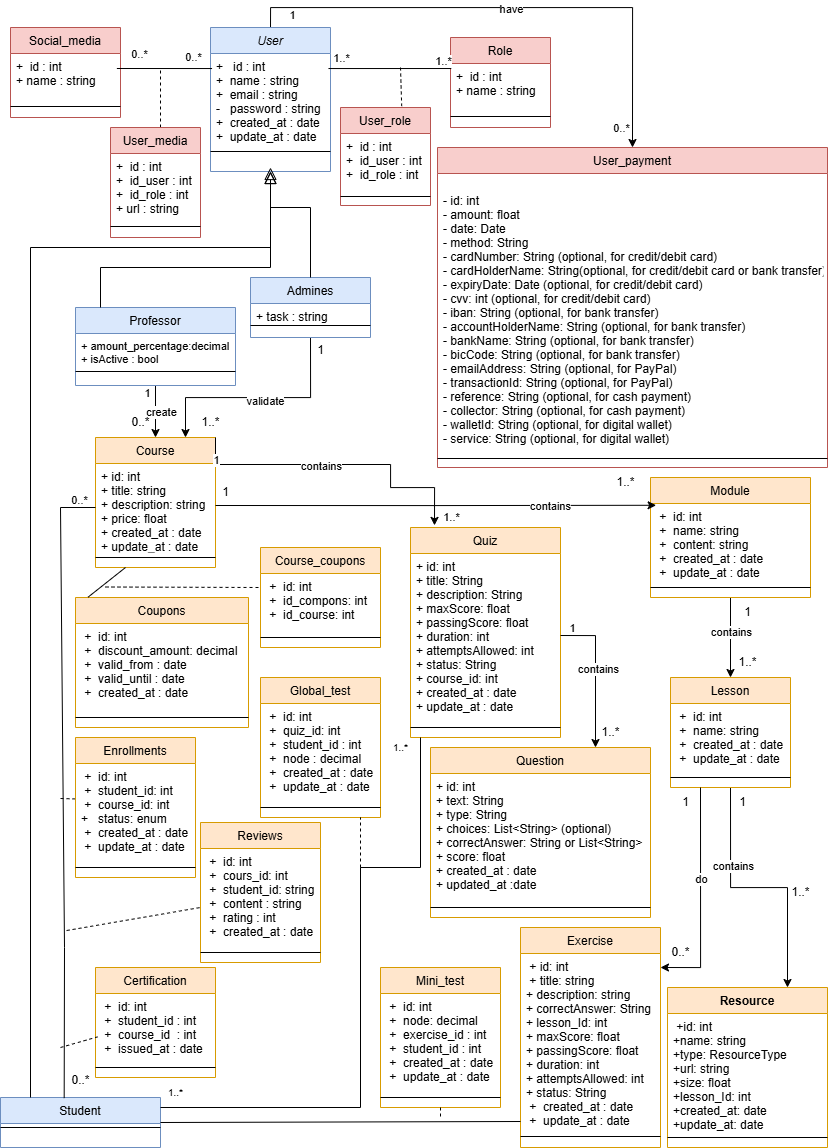


Fig. 2. Use Case diagram of lecturer’s functions



Fig. 3. Use Case diagram of student's functions

1. **Class diagram**

****

1. **kk**