POLITEHNICA UNIVERSITY OF BUCHAREST

SOFTWARE ENGINEERING

Moodle++

Software Requirements Specification

Team:
Stefanescu Bogdan-Laurentiu, gr.1241A
Stan-Soponaru Ioan-Alexandru, gr. 1241A
Cristea George-Razvan, gr. 1241A

Coordinator:
Prof. Nicolae GOGA

Date created:
Wednesday, February 12, 2025

Delivery Report

(will be delivered along with the project)

Name	Group	Project implementation [%, reason]	Signature
Stefanescu Bogdan-Laurentiu	1241A	33%	
Stan-Soponaru Ioan-Alexandru	1241A	33%	
Cristea George-Razvan	1241A	33%	

Delivery date: 31.10.2024

Table of Contents

Requirements Analysis

According to the IEEE STD-830-1993, *IEEE Recommended Practice for Software Requirements Specification*.

1. Introduction

1.1. Purpose

The purpose of this project is to give teachers an easy way to give out materials, homeworks and tasks to the students and for the students to have an organized place where they can see all these materials and assignments without missing anything.

1.2. History

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open-source learning management system (LMS) created by Australian educator and computer scientist **Martin Dougiamas** in 2002.

Initially launched as a free tool for educators, Moodle quickly gained popularity due to its flexibility, user-friendly design, and support for open educational practices. It allowed institutions, from schools to universities, to customize and scale their online courses without high licensing costs. The platform supports a wide range of features like quizzes, discussion forums, assignments, and multimedia integration, catering to diverse educational needs.

1.3. Scope

The scope of this project includes creating a courses platform that can hold different courses for students and teachers alike, each course being able to hold all kinds of documents, tasks and homeworks. The teachers will have access to put the materials and the students only to view them, take part in quizzes, or upload homeworks/tasks.

1.4. Definitions, Acronyms and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendices in the SRS or by reference to other documents.

GDPR: General Data Protection Regulation

LMS: Learning management system

1.5. References

List of references used in this paper, possible legislation that governs the application domain of the project and/or this document.

- **Data Protection Laws:** The platform must abide by applicable data protection laws, such as the General Data Protection Regulation (GDPR) in the EU if it gathers any user data (such as grades or personal information). These regulations control the gathering, storing, and processing of user data. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

- **European Accessibility Act:** This law guarantees that all users, including those with impairments, can use websites and online platforms throughout the European Union. It covers organizations in the public sector as well as other online services, such as learning environments.

  Directive (EU) 2019/882 of the European Parliament and of the Council of 17 April 2019 on the accessibility requirements for products and services (Text with EEA relevance)

1.6. Structure

Describe the structure of the document, with one phrase for each chapter.

The document is organized as follows:

• **The introduction:** Provides an overview of the project, including its goal, history, and scope.

• **General Description:** Outlines the product's features, functions, user interactions, restrictions, and assumptions.

• **System Requirements:** Specify the system's external interfaces, functional specifications, performance measurements, and design limitations.

• **Appendices:** Supplementary documents include client interviews, system diagrams, use case diagrams, class diagrams, sequence diagrams, and team meeting summaries.

2. General description

2.1. Product Description

Describe the finished product that will result from the project. The description will focus mainly on the relationship between the product and the world: which are the external interfaces (system, user, hardware and software).

The "**Moodle++**" is a digital teaching environment that enables seamless interaction between students and teachers. The platform uses software, mobile devices to improve learning, collaboration, and classroom administration. It offers an intuitive and secure environment in which teachers can quickly share resources, assign projects, and track class attendance. Students can use modern QR code scanning technology to authenticate their presence in class.

Key Features:

- **Teacher Dashboard**: Teachers can submit course materials, set tasks, and grade students. The dashboard provides an overview of all students' actions, deadlines, and performance measures to facilitate class administration.

- **Task & Resource Sharing**: Teachers can submit a wide range of documents, such as PDFs, slideshows, and spreadsheets. Students can download these files at any time for class or research purposes.

- **Mobile-Friendly Interface**: The platform is available through mobile phones and online browsers, allowing students and teachers to interact with the system from anywhere.

External Interfaces:

- **User Interface**: Users can access the platform's important features via a web or mobile interface. Teachers and students can use their dashboards to manage tasks, assignments, and attendance records.
- **Software Interfaces**: It uses common APIs to synchronize data between mobile devices and the central database, resulting in a seamless user experience and safe data storage.

2.2. Product Functions

Briefly describe how the product will be used. Don't forget to obtain as many information as possible form interviews (see Appendix 1). Do not confuse this information with that obtained in the design/brainstorming processes.

The product will be used by students and teachers.

- **Students** will have a dashboard where they can see their upcoming assignments, a calendar where they can see in each day what assignments they have/had, the courses page that contains each course they are assigned to. Each course will in turn hold everything related to that course. For each course, students will be able to see the total grade for that course as well as the individual grading assigned by teachers to them.

- **Teachers** will be able to upload/modify anything about the course they have. This includes documents, tasks, homeworks, quizzes and other activities. In the courses page, they will see all the groups they are teaching to. They will be able to review the students' activity in the course and assign grading to each of their activity.

2.3. User description

Describe how the product will be used by users and possible limitations imposed by the technical level of the users.

Users will be able to register/login on the platform. They will then be sent on the home page where they will be able to see a few upcoming activities and select from a menu to go to either: dashboard, calendar, courses or logout. There they will see various courses and activities they are enrolled to. Regarding technical constraints, the system will be designed to prioritize ease of use and intuitive navigation.

The teachers who are inexperienced with software systems may find it challenging to create certain tasks such as a quiz with certain types of questions. Users who are inexperienced with applications may find it challenging to see tasks that have passed the expiration date.

2.4. Constraints

Hardware limitations, interfaces with other applications, application importance (if the application is critical or not, if the application must always generate correct results), functions depending on the legislation specified in chapter 1.5, minimum/maximum number of users, parallelism.

Application Importance:

- **Mission-Critical Application**: The Moodle platform remake is considered **mission-critical** for educational institutions, where disruptions can significantly impact learning and administrative operations.
- **Accuracy**: The platform must **always generate correct results**, particularly for grading, course progression, and assessments. Any incorrect results could have legal and academic consequences.
- **Compliance with Legislation**: As outlined in Chapter 1.5, the system must ensure compliance with data privacy laws (e.g., GDPR), and education-specific regulations.

Minimum/Maximum Number of Users:

- **Minimum Users**: The system should support a minimum of 1,000 active users.
- **Maximum Users**: The platform must handle up to 20,000 registered users, though not all simultaneously active. Concurrent users should be capped at around 5,000 to maintain system stability.

Parallelism:

- **Concurrent User Handling**: The system must allow parallel handling of requests. Load balancing is required to distribute tasks efficiently across multiple servers.
- **Task Execution**: Background tasks, such as content uploads, database updates, and batch processing (e.g., grades or reports generation), should not affect the user experience.

2.5. Assumptions and Dependencies

Describe the machine on which the application will run, the dependencies of the data base systems/ program libraries/existing programs.

The **Moodle++** platform relies on a number of external elements and assumptions to enable effective functionality and integration. These include hardware, software requirements, and environmental considerations.

- **Hardware Assumptions**:

  The application should run on any current laptop, desktop, or mobile device with internet connectivity. It will use cloud-based or on-premise servers to manage storage and computing. Servers must have enough capability to support up to 20,000 registered users, with a maximum concurrency of 5,000.

- **Software Dependencies**:

  Moodle++ requires integration with many libraries and APIs. The system must work with PostgreSQL or MySQL databases and be deployable in cloud settings like AWS or Microsoft Azure.

- **Operating Systems**:

  The platform is expected to support key server operating systems like Linux, Windows Server, and macOS. Client-side devices can run a variety of operating systems, including Windows, MacOS, Android, and iOS.

- **Database and Storage**:

  Moodle++ will use a centralized database for storing user data, course content, and activity logs. The system's operation is based on a strong database solution (PostgreSQL, MySQL) and cloud storage solutions like Amazon S3 to handle massive volumes of course material, such as videos and documents.

- **Internet and Network**:

  The application assumes a reliable internet connection for both users and servers. As a cloud-based system, any disruptions in internet access could have an impact on the user experience, especially for real-time interactions such as video conferencing.

- **Security Dependencies:**

  The system will use secure protocols like HTTPS and TLS to encrypt data in transit. Furthermore, data at rest will be encrypted (AES-256), and the platform will rely on regular security audits to ensure compliance with GDPR and other relevant requirements.

3. System Requirements

3.1. External Interface Requirements

User Interfaces

- **User Interface Design**: The Moodle remake should provide a modern, responsive, and intuitive UI compatible with multiple devices (desktop, tablets, smartphones).
- **Multilingual Support**: Interface and content translation capabilities, supporting multiple languages.

Hardware Interfaces

- **Server Hardware**: Requires integration with cloud-based or on-premise server infrastructure capable of handling 20,000 registered users, with a focus on session-based scaling (e.g., AWS EC2, Azure).
- **User Devices**: Moodle should be accessible via standard devices such as PCs, laptops, tablets, and mobile phones.
- **Storage**: Integration with scalable storage systems (e.g., S3 or a comparable service) for data, video, and document storage.

Software Interfaces

- **Database**: Compatibility with PostgreSQL, MySQL, or other relational databases for user management, course materials, and activity logs.
- **Operating Systems**: Linux, Windows Server, or macOS.
- **Third-Party Libraries**: Integration with libraries for file sharing (e.g., Google Drive, OneDrive), and authentication (e.g., LDAP, OAuth).
- **APIs**: Support for RESTful and GraphQL APIs to integrate with external systems (student information systems, HR systems).

Communication Interfaces

- **Protocols**: Secure HTTP (HTTPS), WebSocket for real-time interactions.
- **Notification Systems**: Integration with email servers, SMS services, and push notifications for communication with users.
- **Load Balancers**: For optimizing traffic to multiple instances of Moodle to handle spikes in user activity.

3.2. Functional Requirements

Product requirements (as in chapter 2.2, but in detail)
Use cases
Data flow, data management

- **User Management**: Ability to manage user roles (students, teachers, administrators), with 20,000+ registered users, including automated enrollment based on external databases.
- **Course Management**: Creation, management, and scheduling of courses, assignments, and quizzes.
- **Grading & Analytics**: Real-time data tracking for course progress, grading, and reporting, with analytics dashboards for instructors and administrators.
- **Collaboration Tools**: Built-in forums, chats, and group workspaces.

Use Cases

- **Login and Authentication**: Students authenticate using SSO (Single Sign-On) or OAuth, access their courses, and participate in discussions.
- **Course Enrollment**: Automated and manual enrollment in courses based on predefined rules.
- **Assignment Submission**: Students upload files for review, and instructors grade with feedback.
- **Real-Time Learning**: Users participate in live video lectures integrated into Moodle.
Data Flow and Data Management

- **Student Data**: Managed securely in the database, with options for backups and data recovery.
- **Content Distribution**: Optimized data flow for multimedia content (videos, presentations), ensuring minimal load times.
- **Activity Logs**: Recording user activity for audit purposes.

3.3. Performance Requirements

Clear values regarding the measurable characteristics of the system (e.g. number of users, number of frames per second, etc.)

- **Concurrent Users**: The system should handle up to 5,000 concurrent users during peak times without significant degradation in performance.
- **Response Time**: Average page load time must be under 3 seconds.
- **Scalability**: Automatic scaling to manage up to 20,000 total registered users, with dynamic load balancing during peak times.
- **Uptime**: Minimum uptime of 99.9% over a 12-month period.

3.4. Design Constraints

The compliance of the rules described in the normative documents (see chapter 1.5)
The compliance of the hardware constraints.
The compliance of other constraints.

3.5. Software System Attributes

Trust
Reliability
Security
Maintenance
Portability
Fault tolerance

Trust

- **User Trust**: The platform should be transparent about data usage and ensure that user information is secure and accessible to authorized individuals only.

Reliability

- **Failover Mechanism**: Automated failover for mission-critical components, ensuring continuous access.
- **Data Backup**: Nightly backups, with the ability to recover within 12 hours in case of failure.

Security

- **Authentication & Authorization**: Support for multi-factor authentication (MFA), role-based access control (RBAC).
- **Data Encryption**: Use of SSL/TLS for data in transit and encryption for data at rest (AES-256).
- **Security Audits**: Regular security assessments and vulnerability testing.

Maintenance

- **Automated Updates**: Support for rolling updates with minimal downtime.
- **Error Logging**: Comprehensive error logging and diagnostic tools for system administrators.

Portability

- **Cloud Deployment**: Ability to deploy on different cloud environments (AWS, Azure, Google Cloud).
- **Backup Migration**: Support for exporting user data, course content, and grades for migration between systems.

Fault Tolerance

- **Redundancy**: The platform should have redundancy in its critical components to ensure that failure of a single component does not impact the entire system.
- **Auto-Recovery**: In case of failure, the system should have automated processes for recovery and restart.

3.6. Other System Requirements

- **Legal Compliance**: Must meet global legal requirements for data protection, user privacy, and content ownership (e.g., GDPR, DMCA).
- **Documentation**: Complete system documentation for both end-users and administrators, with training materials for onboarding new users.
- **Support**: 24/7 technical support for critical system failures and a community forum for minor issues.

Appendices

A1. Interview with the customer

During the interview with the customer, several key pain points and improvement ideas for the Moodle platform were discussed, reflecting the user experience and functionality that need enhancement.

One issue raised was the quiz navigation. The current quiz system doesn't allow students to easily go back to previous questions. The customer suggested implementing a feature that lets users navigate freely between questions, which would result in a more fluid quiz-taking experience. Additionally, there was a desire for an easier quiz interface, where the design is simplified, making it more approachable and user-friendly for students.

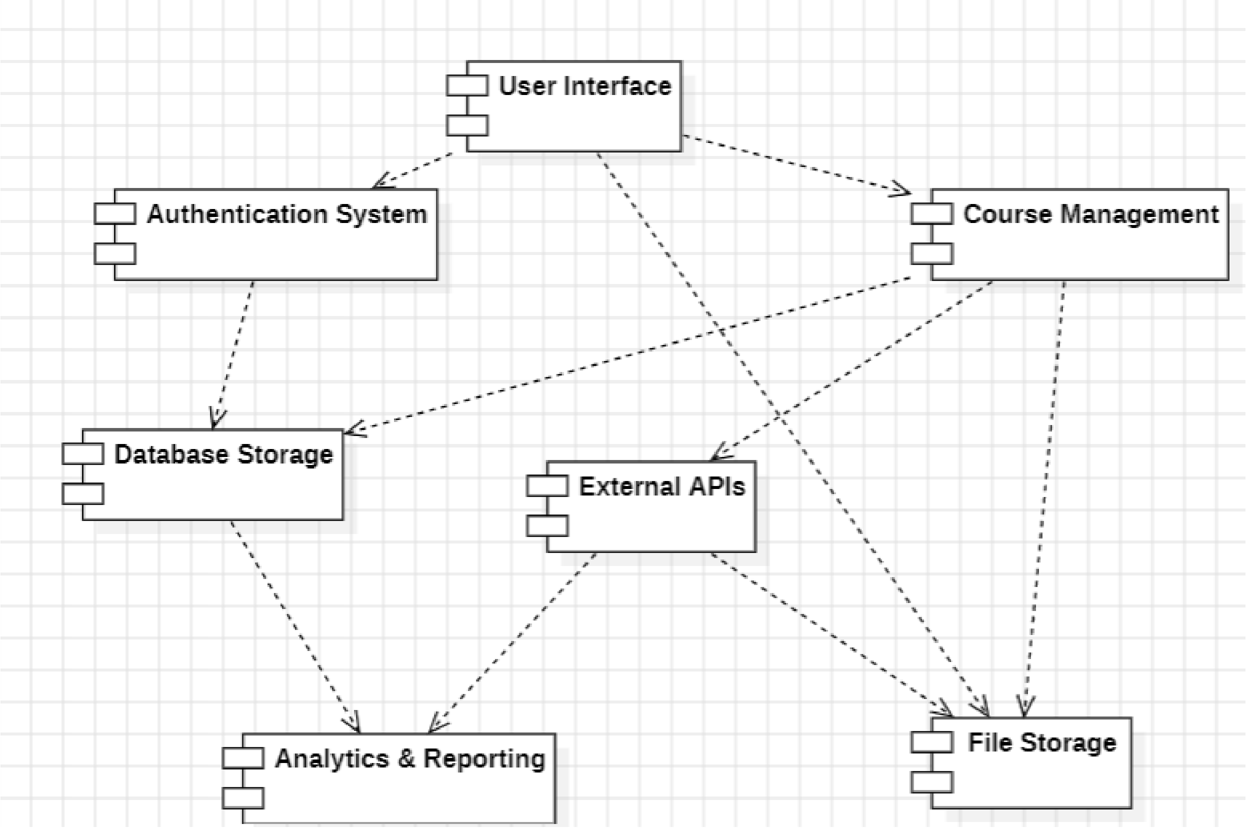
Another major concern was the difficulty in finding certain resources. The customer noted that finding specific documents or materials in the current version of Moodle is cumbersome, requiring too much effort from users. To address this, the interface should be restructured to prioritize document accessibility, ensuring that important resources can be found quickly, potentially through an improved search function or a more logical organization of content.

The overall interface of the platform was also highlighted as needing improvement. The customer emphasized that the interface should be much more intuitive. An easier-to-navigate design would reduce frustration and enhance the learning experience. This is particularly important for students and instructors who need to manage courses and assignments efficiently. The mobile application was singled out as particularly problematic, with its design and usability lagging behind expectations. Given the increasing reliance on mobile devices for accessing learning platforms, a more user-friendly mobile experience is essential.

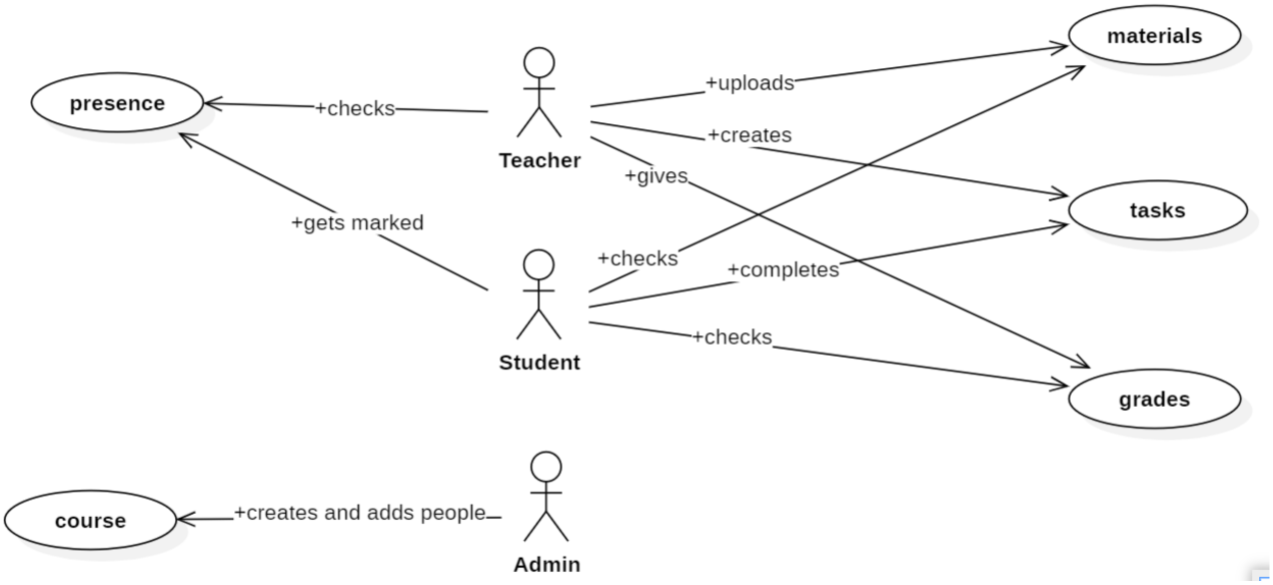
Another significant concern mentioned was the perception that feedback on assignments is not anonymized. This raises privacy concerns among students, who might hesitate to provide honest evaluations or peer feedback. Addressing this by ensuring that feedback mechanisms are properly anonymized would enhance user trust and promote a more transparent learning environment.

These insights point to the need for an enhanced user experience that focuses on intuitiveness, accessibility, and usability, ensuring both desktop and mobile users have seamless access to essential tools and resources.

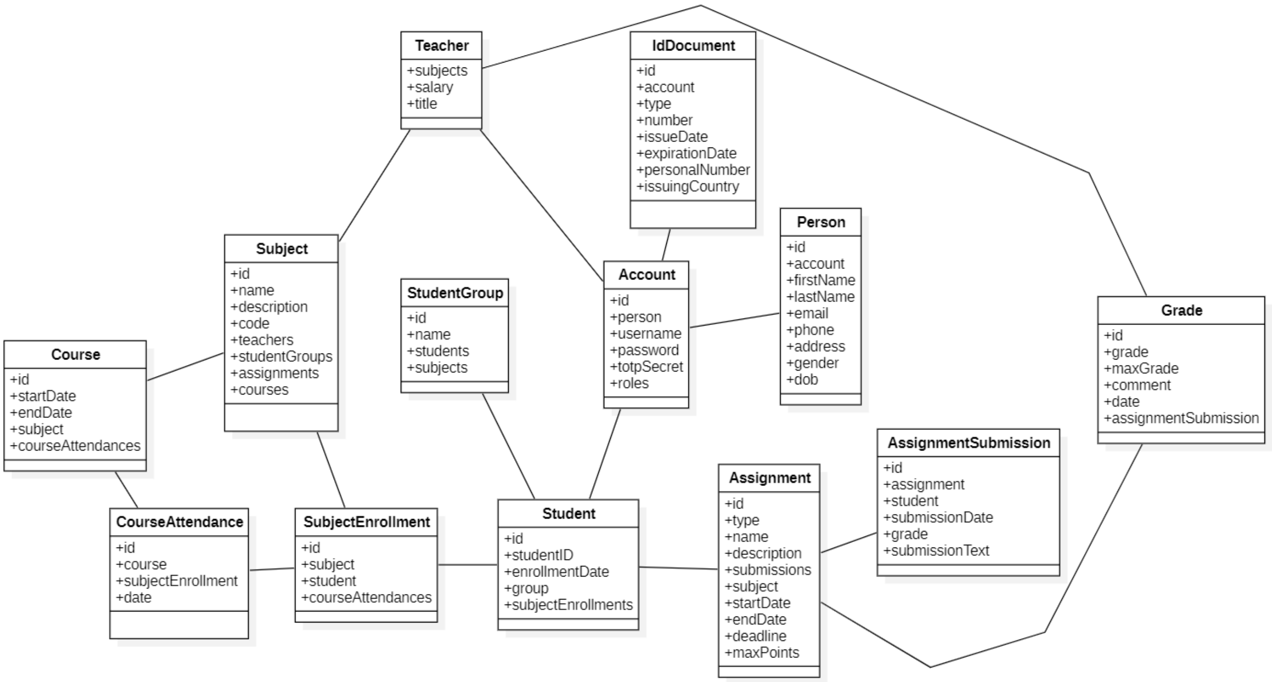
A2. System diagram



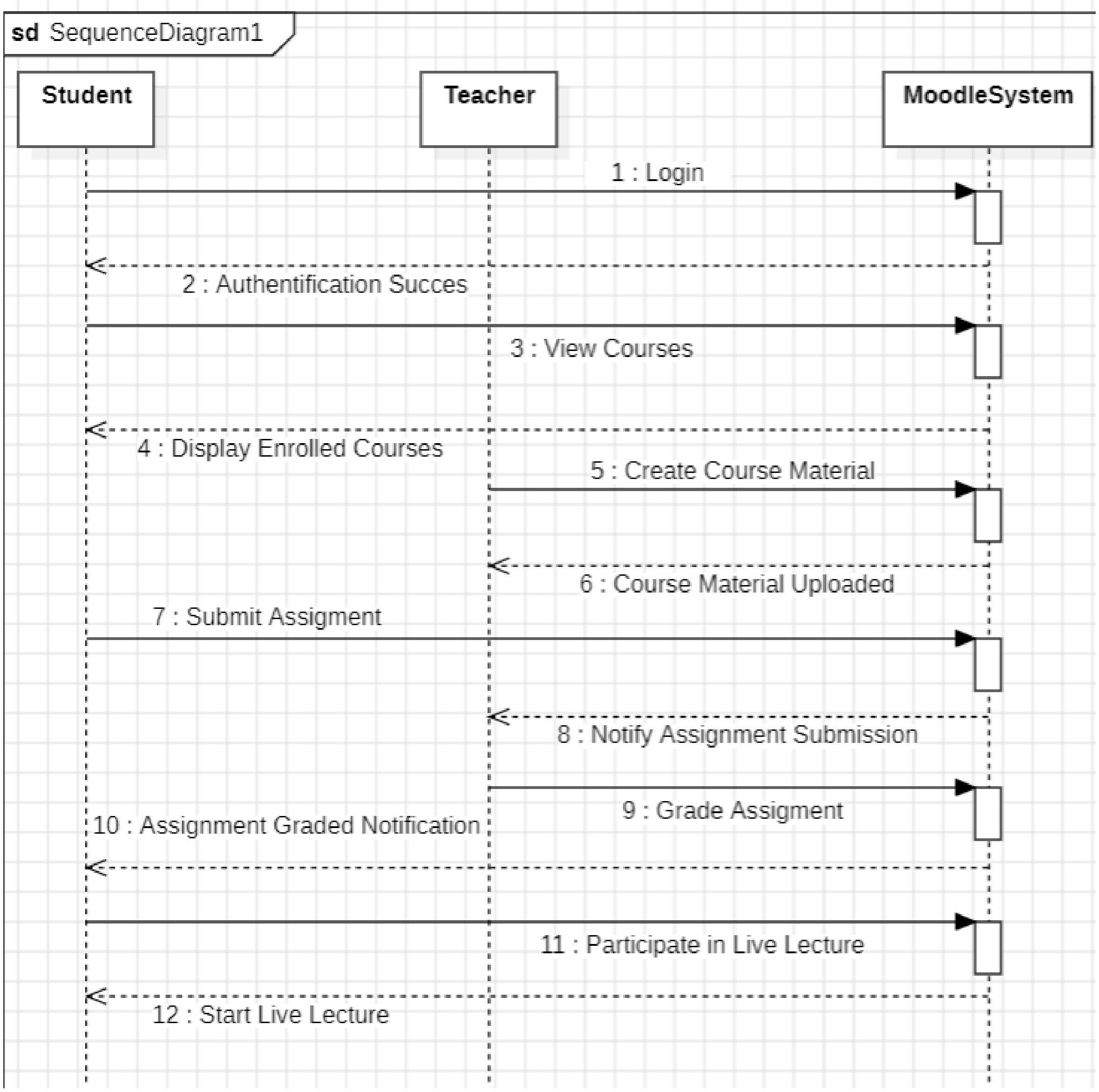
A3. Use Cases Diagrams (put it in the right UML format)



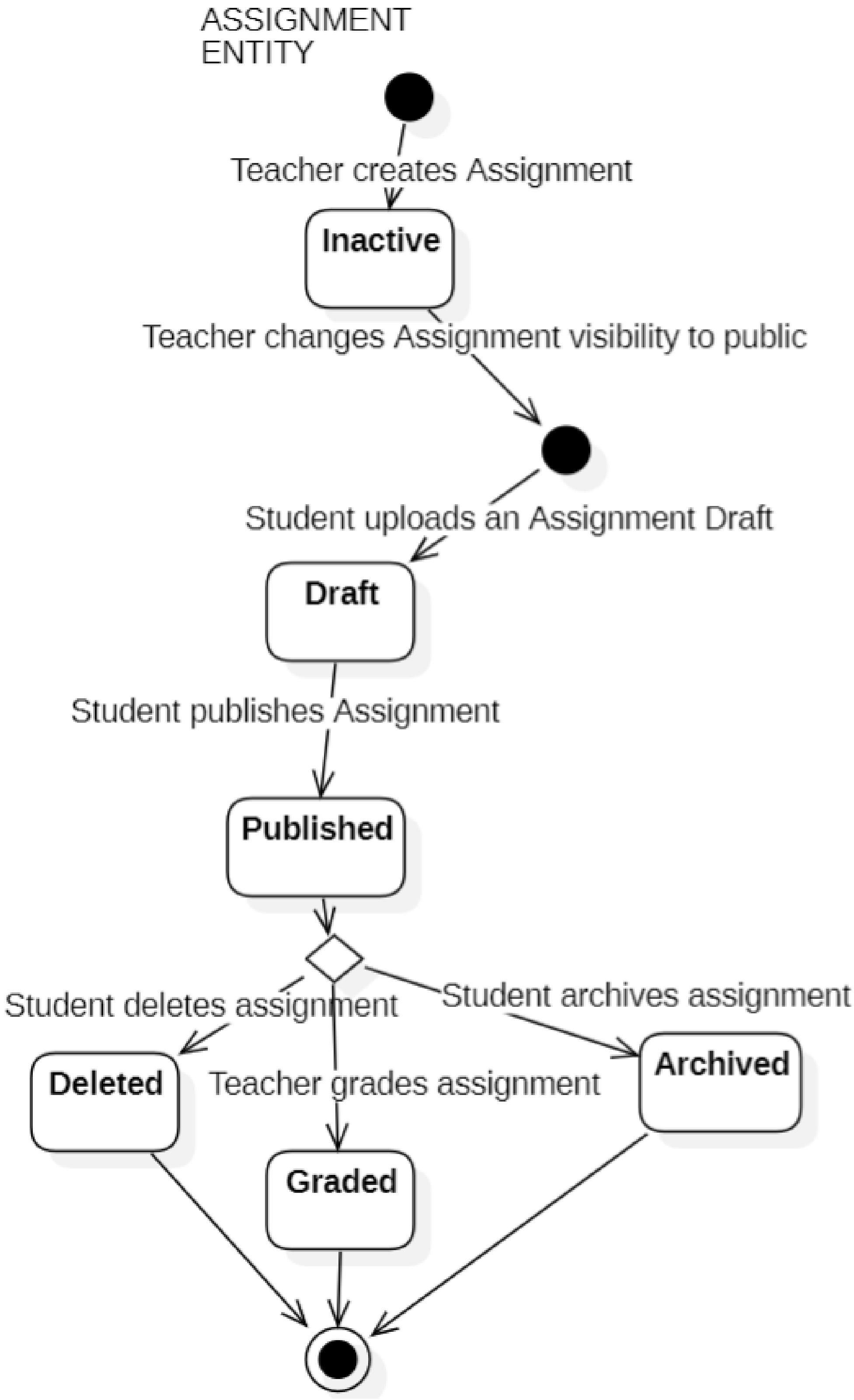
A4. Class Diagrams



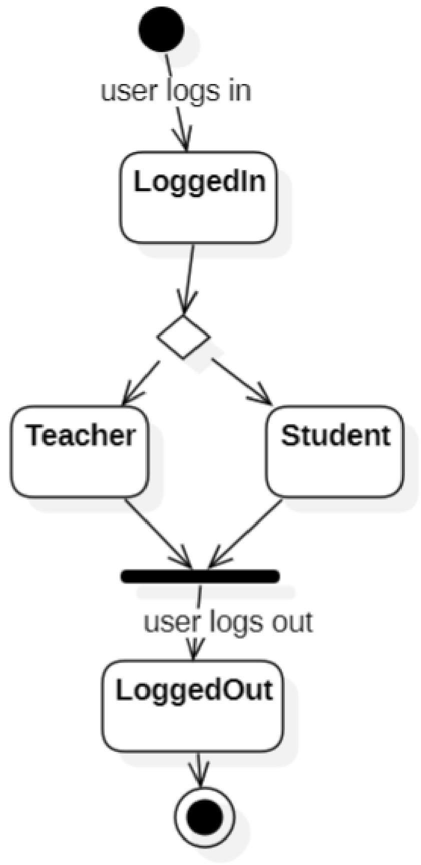
A5. Sequence Diagrams


sd SequenceDiagram1
Student
Teacher
MoodleSystem
1 : Login
2 : Authentification Succes
3 : View Courses
4 : Display Enrolled Courses
5 : Create Course Material
6 : Course Material Uploaded
7 : Submit Assigment
8 : Notify Assignment Submission
9 : Grade Assigment
10 : Assignment Graded Notification
11 : Participate in Live Lecture
12 : Start Live Lecture

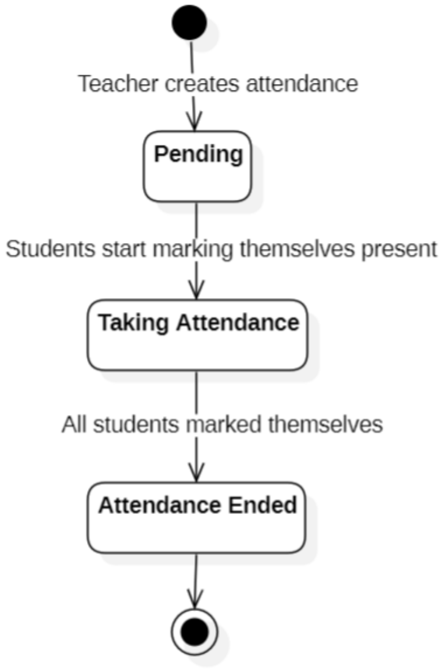
A6. State Diagrams


ASSIGNMENT ENTITY
Teacher creates Assignment
Inactive
Teacher changes Assignment visibility to public
Student uploads an Assignment Draft
Draft
Student publishes Assignment
Published
Student deletes assignment
Student archives assignment
Deleted
Teacher grades assignment
Archived
Graded

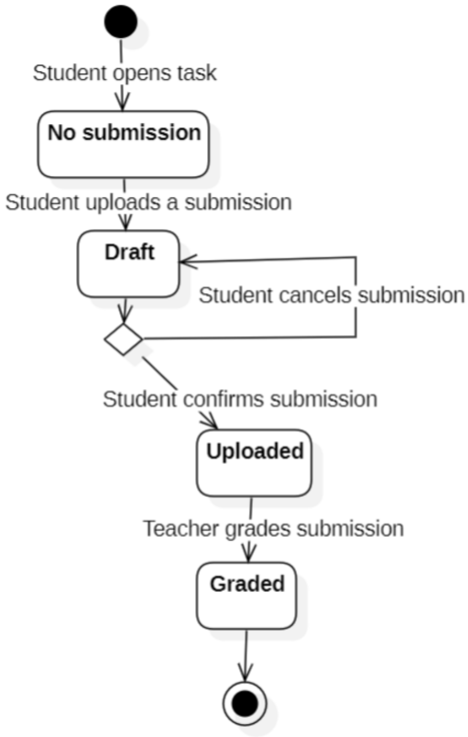
USER ENTITY



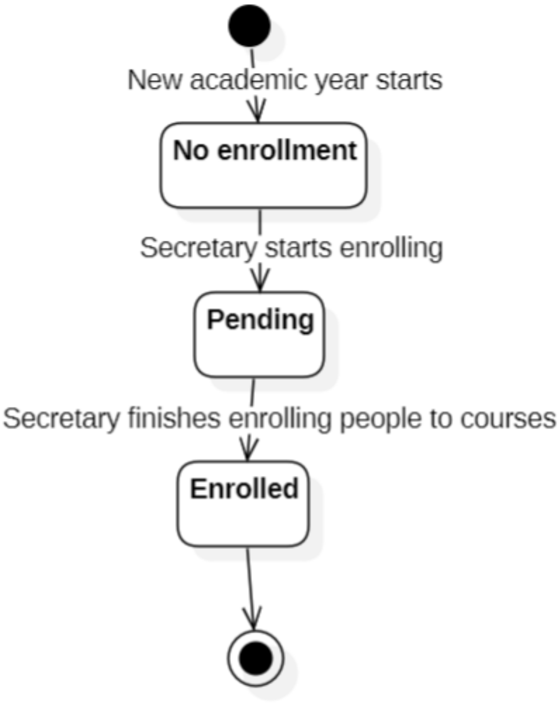
ATTENDANCE


Teacher creates attendance
Pending
Students start marking themselves present
Taking Attendance
All students marked themselves
Attendance Ended

ASSIGNMENT
SUBMISSION


Student opens task
No submission
Student uploads a submission
Draft
Student cancels submission
Student confirms submission
Uploaded
Teacher grades submission
Graded

SUBJECT
ENROLLMENT


New academic year starts
No enrollment
Secretary starts enrolling
Pending
Secretary finishes enrolling people to courses
Enrolled

A7. Document Evolution

This section displays the changes and updates made to the document throughout time, demonstrating its evolution and iterations.

Date: 21.10.2024
Version: 1.0.
Summary of changes: First draft of the Software Requirements Specification (SRS). The introduction now includes sections 1.1 (Purpose), 1.2 (History), and 1.3 (Scope). The general description (Section 2.1) gave an overview of the product. Sections 3.1 and 3.2 include the initial brainstorming of functional and external interface needs.

Date: 22.10.2024
Version: 1.1.
Summary of changes: Completed system requirements. Sections 3.1–3.5. Concentrated on defining the platform's external interfaces and functional needs in more depth. User, course, and collaboration tools now have descriptions.

Date: 23.10.2024
Version: 1.2.
Summary of changes: Updates to the performance criteria (Section 3.3) to assure 5,000 concurrent users and 20,000 total users. Modified assumptions and dependencies (Section 2.5) for cloud-based hosting solutions. Final review and minor edits were completed in preparation for submission.

A8. Report regarding team meetings

Date:
Location:
Participants:
Summary:
Activity:
Title
Description (4 lines)
Results:
Element
Description (2-3 lines)

Date: 21.10.2024
Location: Online
Participants: Stefanescu Bogdan-Laurentiu, Stan-Soponaru Ioan-Alexandru, Cristea George-Razvan
Summary: Discussing Project Topic and Requirements
Activity:
The team examined the project scope and outlined Moodle++'s primary objectives. We brainstormed the key elements, such as resource sharing and student task submission.
Results:
Established the project goals and began drafting the initial document.

Date: 22.10.2024
Location: Online

Participants: Stefanescu Bogdan-Laurentiu, Stan-Soponaru Ioan-Alexandru, Cristea George-Razvan
Summary: Discussing System Requirements
Activity:

The team examined the public interface and functional requirements, concentrating on user roles, grading systems, and collaborative tools. We also started tweaking the performance limits.

Results:

Completed points 3.1 (External Interface Requirements), 3.2 (Functional Requirements), 3.3 (Performance Requirements), and 3.5 (Software System Attributes).

Date: 23.10.2024
Location: Building CB, UNSTB Campus
Participants: Stefanescu Bogdan-Laurentiu, Stan-Soponaru Ioan-Alexandru, Cristea George-Razvan
Summary: Reviewing Design Diagrams
Activity:

Title: Diagrams Review

The team discussed and drew initial drafts of the system diagrams, which included use case and sequence diagrams.

Results:

Element: Use Case Diagrams

Description: Developed draft use case diagrams based on system functionality and user interactions.

Date: 23.10.2024
Location: Online
Participants: Stefanescu Bogdan-Laurentiu, Stan-Soponaru Ioan-Alexandru, Cristea George-Razvan
Summary: Discussing Appendices
Activity:

Title: Finalizing Appendices

The team went over the appendices, which included the system and class diagrams. The emphasis was on assuring completion and coherence with the primary material.

Results:

Element: Sequence Diagrams

Description: Finalized the sequence diagrams for core functionalities like task submission and grading.

Date: 27.10.2024
Location: Online
Participants: Stefanescu Bogdan-Laurentiu, Stan-Soponaru Ioan-Alexandru, Cristea George-Razvan
Summary: Discussing Review and final changes
Activity: Collaborative Document Review and Revision

Title: Final Changes

Results: The team examined the document, found areas that needed clarification or refinement, and recommended final changes to ensure consistency, correctness, and alignment with project objectives. Each participant provided feedback on specific areas of the content, with an emphasis on improving its quality and coherence.

Element: Grammar and Style Edits

Description: Minor grammar problems were corrected, and the style was uniformed across the document to maintain a professional tone.

Element: Formatting Adjustments

Description: Standardised layout for headings, bullet points, and numbering ensures a consistent and clean presentation.

Element: Summary and Key Takeaways

Description: To increase readability and ease of reference, a succinct summary and key takeaways were added in each section.

A9. Conclusions regarding the activity

Until now, we have worked effectively as a team, encouraging open communication to discuss ideas and distribute tasks efficiently. We kept a clear vision for the **Moodle++** platform by establishing the project concept and core functionality from the beginning. We have also created detailed documentation that includes the project's goal, needs, and schematics, so improving clarity and use. Our final review process guaranteed consistency and quality throughout all sectors, preparing us for the next stages of growth. Overall, our systematic approach and teamwork established a solid foundation for the successful development of **Moodle++,** and we are eager to continue forward.