



Faculty of Computing and Technology University
of Kelaniya

Learning Management System

“Padama”

Software Requirement Specification

Table of Contents

Software Requirements Specification (SRS) for Padama	1
1. Introduction.....	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope.....	2
2. Overall Description of the System.....	2
2.1 Product Functions.....	2
2.2 System Modelling.....	4
2.3 Main Flow	6
3. System Features	7
3.1 Use Case 1: Course Management.....	7
3.2 Use Case 2: User Registration and Management.....	10
3.3 Use Case 3: Content Delivery	12
3.4 Use Case 4: Progress Tracking	14
3.5 Use Case 5: Communication Tools	16
3.6 Use Case 6: File Storage and Management	18
4. Architectural Design	20
4.1 Non-functional Requirements.....	20
4.2 Architectural Design.....	22
4.3 Design and Implementation	24
4.4 Database Design	25
5. Project Plan	26
5.1 Project Schedule.....	26
5.2 Project Budget.....	27
References.....	28
Appendices	29
Appendix A: Glossary	29
Appendix B: Acronyms	29

Software Requirements Specification (SRS) for Padama

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to serve as a comprehensive guide detailing the requirements, functionalities, and architecture of the Learning Management System (LMS) named "Padama." This document aims to provide clarity and direction to stakeholders involved in the development, implementation, and usage of Padama. It outlines the scope of the project, identifies key stakeholders, and establishes the foundation for designing, building, testing, and deploying the LMS.

1.2 Intended Audience and Reading Suggestions

Intended Audience

The intended audience for this SRS includes:

- **Project Managers:** To understand project scope, timelines, and deliverables.
- **System Architects and Developers:** To implement the system based on detailed requirements and design specifications.
- **Quality Assurance Teams:** To develop comprehensive test cases and ensure the system meets specified requirements.
- **End Users (Instructors, Students):** To understand the features and functionalities available in Padama.
- **Stakeholders (Educational Institutions, Corporate Training Providers):** To align business objectives with system capabilities.

Reading Suggestions

Depending on their roles, readers are encouraged to focus on specific sections of the SRS:

- **Project Managers** should review Sections 1, 2, 5, and 6 for an overview of project goals, system functionalities, project plan, and references.
- **System Architects and Developers** should concentrate on Sections 2, 3, 4, and 6 for detailed system description, requirements, architectural design, and references.

- **Quality Assurance Teams** should refer to Sections 3, 4, and 5 for functional requirements, architectural design, and project plan details.
- **End Users and Stakeholders** should explore Sections 2 and 3 to understand user interactions, system capabilities, and benefits.

1.3 Product Scope

Padama is envisioned as a robust Learning Management System designed to support educational institutions, corporate training programs, and online course providers. It aims to streamline course management, enhance learning experiences, and facilitate collaboration among users. The system will include comprehensive features for course creation, content delivery, user management, progress tracking, communication, and assessment. Padama's scope encompasses both frontend user interactions and backend system functionalities, ensuring a seamless and efficient educational platform.

2. Overall Description of the System

2.1 Product Functions

Padama will offer a wide range of functionalities to meet the diverse needs of its users:

Course Management

Instructors will have the capability to create, organize, and manage courses within Padama. They can upload various types of course materials, including documents, videos, and interactive quizzes. Course content can be structured into modules and lessons, allowing for flexible course design. Instructors can also set up assessments, such as assignments and quizzes, and manage course settings like enrollment options and prerequisites.

User Management

Padama will support user registration, authentication, and management. Users, including students, instructors, and administrators, can create and manage their profiles within the system. They can enroll in courses, track their progress, and interact with course materials based on their roles and permissions. User management functionalities ensure secure access to Padama and personalized user experiences.

Content Delivery

The system will facilitate the distribution and access of course materials to users. Instructors can upload, organize, and update content regularly, ensuring seamless access for students. Content delivery mechanisms will support multimedia formats, interactive elements, and accessibility features to cater to diverse learning needs.

Progress Tracking

Padama will provide tools for monitoring and reporting user progress and performance. Instructors can track individual and group progress, view participation metrics, and assess learning outcomes. Progress tracking features enable personalized feedback and interventions to support student success and course effectiveness.

Communication Tools

The system will include communication and collaboration tools to facilitate interaction among users. Features such as discussion forums, real-time chat, and messaging will promote collaborative learning environments. Instructors and students can engage in discussions, ask questions, share resources, and provide feedback, enhancing communication and knowledge sharing within Padama.

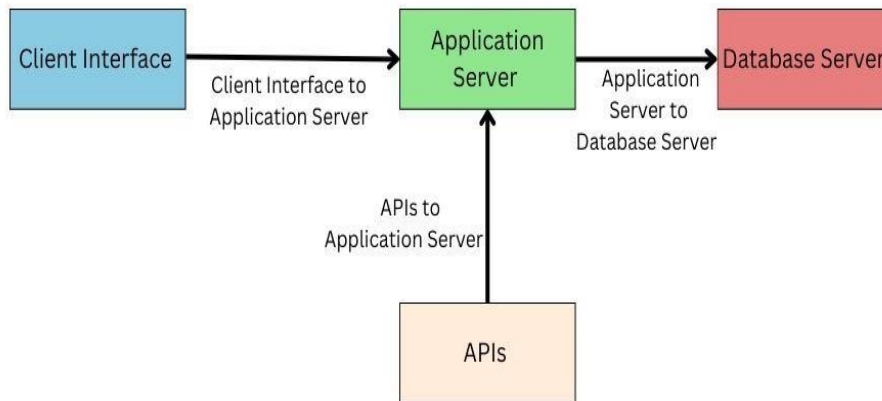
Assessment and Evaluation

Padama will support various assessment methods to evaluate student learning. Instructors can create and manage assignments, quizzes, and exams within the system. Automated grading tools will streamline assessment processes, providing timely feedback to students. Assessment functionalities will ensure fair evaluation practices and alignment with learning objectives.

File Storage and Management

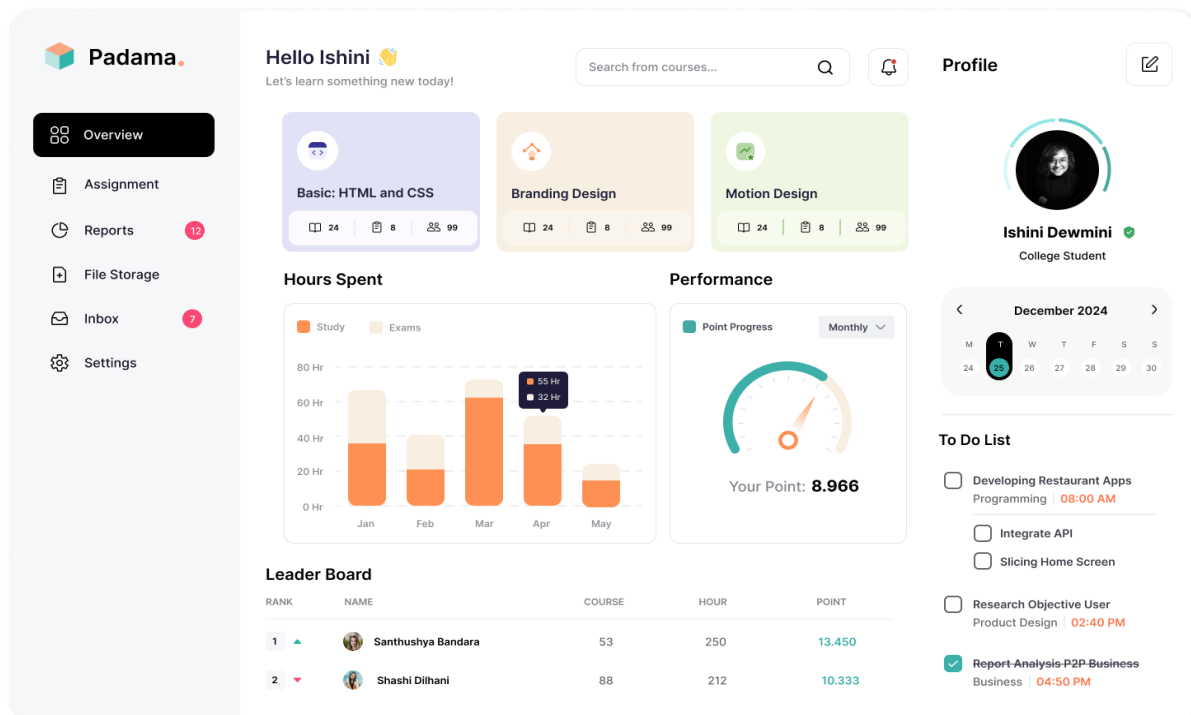
Users will have access to secure file storage and management capabilities within Padama. They can upload, store, and organize files associated with courses, assignments, and collaborative projects. Version control, document sharing, and access permissions will enhance file management efficiency and data security across the platform.

2.2 System Modelling



Padama will adopt a client-server architecture to manage data flow and interactions between users and the system:

Client Interface

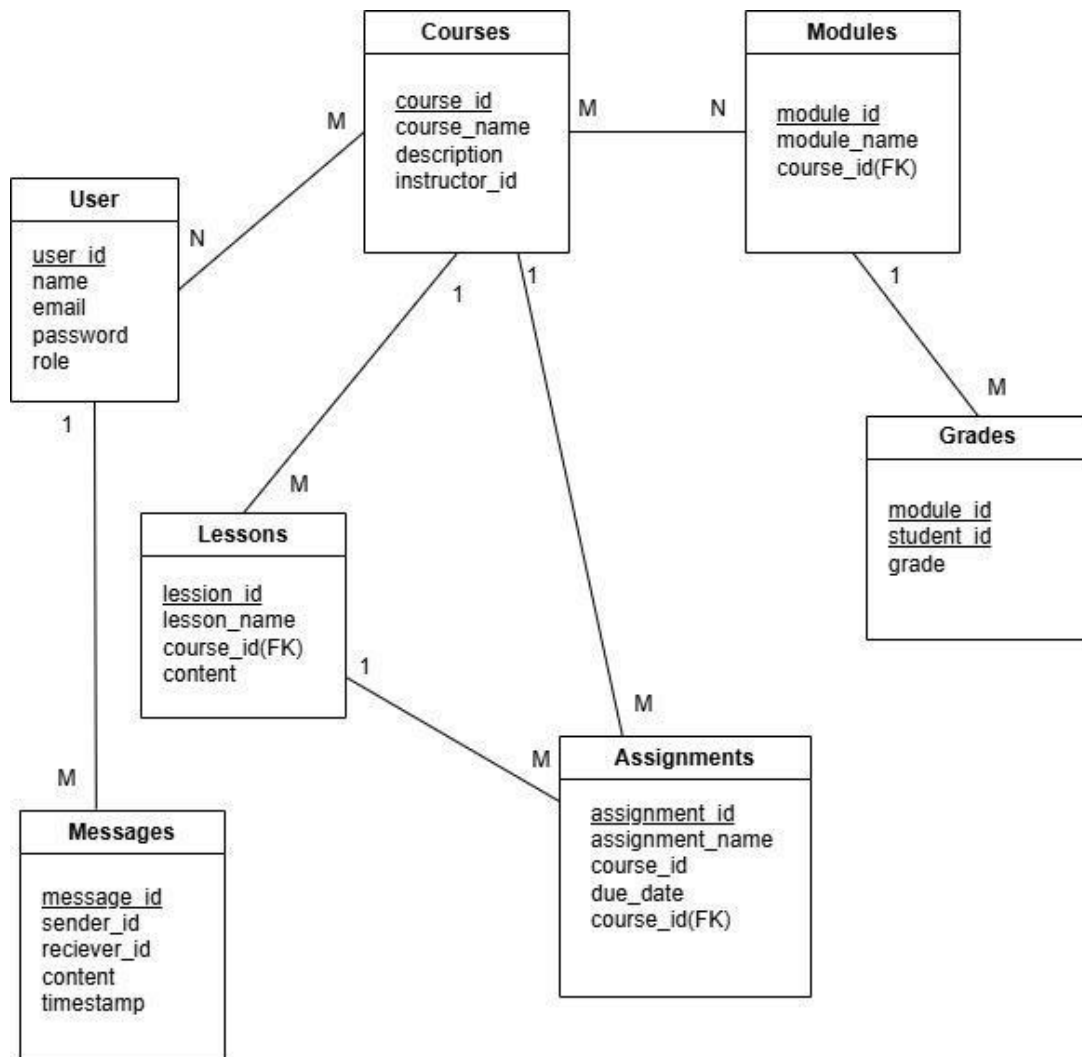


The frontend of Padama will be accessible via web browsers, ensuring compatibility across devices and platforms. The user interface (UI) will be designed for intuitive navigation and user-friendly interactions. Modern web technologies such as HTML5, CSS3, and JavaScript (React.js) will be employed to create responsive and dynamic user experiences.

Application Server

The backend of Padama will host the core application logic, user authentication, and business processes. It will utilize Node.js and Express.js frameworks to handle server-side operations efficiently. The application server will manage data transactions, process user requests, and communicate with external systems through RESTful APIs, ensuring scalability and performance.

Database Server



Database schema

Padama will rely on MongoDB as the database management system to store and retrieve data securely. The database server will manage user profiles, course information, content metadata, and

system configurations. MongoDB's document-oriented architecture will support flexible data storage and querying, accommodating Padama's evolving requirements and data volumes.

APIs

RESTful APIs will facilitate communication between the client interface and the application server in Padama. APIs will enable seamless data exchange, integration with external services, and support for mobile applications. Standardized API endpoints will ensure interoperability and maintainability of the system's backend components.

2.3 Main Flow

The main flow of Padama encompasses typical user interactions and system operations:

User Registration and Login

Users will register with Padama using their email addresses and creating passwords. Upon successful registration, they can log in to access their profiles and course materials. User authentication mechanisms will verify credentials securely, protecting user accounts from unauthorized access.

Course Creation and Management

Instructors will create new courses within Padama by providing titles, descriptions, and metadata such as keywords and categories. They can upload course materials, organize content into modules and lessons, and configure course settings. Course management features will empower instructors to customize learning experiences and optimize course structures based on educational goals.

Course Enrolment and Progress Tracking

Students will browse available courses, review course details, and enroll in desired courses through Padama's interface. They can access course materials, participate in discussions, submit assignments, and take quizzes online. Progress tracking tools will monitor individual and group performance, providing insights into learning outcomes and identifying areas for improvement.

Communication and Collaboration

Padama will facilitate communication among users through dedicated forums, real-time chat, and messaging features. Instructors can initiate discussions, facilitate peer-to-peer interactions, and provide academic support. Collaboration tools will foster community engagement, knowledge sharing, and collaborative learning experiences within Padama's virtual learning environment.

Assessment and Feedback

Instructors will create assessments, such as assignments and quizzes, within Padama. Students can complete assessments online, submit responses, and receive automated grading and feedback. Instructors will review submissions, provide personalized feedback, and evaluate student performance based on predefined criteria. Assessment functionalities will promote continuous learning, skill development, and academic achievement among users.

File Storage and Management

Users can upload, store, and manage files securely within Padama. They can organize files into folders, set access permissions, and collaborate on shared documents. Version control features will track document revisions, ensuring data integrity and facilitating collaborative projects. File management capabilities will support seamless integration of multimedia content and resources across Padama's educational platform.

3. System Features

3.1 Use Case 1: Course Management

3.1.1 Description and Priority

Course management functionalities enable instructors to create, organize, and manage courses within Padama. This use case is critical for delivering educational content and supporting diverse learning experiences.

3.1.2 Functional Requirements

F1:	Course Creation
Input:	Course title, description, metadata (keywords, categories)
Processing:	Instructors input course details into the system.
Output:	A newly created course with associated metadata.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Course title: The name of the course.• Description: A summary of the course content.• Metadata: Keywords and categories associated with the course. <p>Output: Course creation confirmation.</p>

F2:	Upload Course Materials
Input:	Course materials (documents, videos, quizzes).
Processing:	Instructors upload materials to the course.
Output:	Course materials available to students.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Documents: Text files, PDFs, etc.• Videos: MP4, AVI, etc.• Quizzes: Interactive question sets. <p>Output: Successful upload confirmation.</p>

F3:	Organize Course Content
Input:	Course content, module structure, lesson details.
Processing:	Instructors arrange content into modules and lessons.
Output:	Organized course structure.

Definition:	Inputs: <ul style="list-style-type: none"> • Course content: Materials and resources. • Module structure: Groupings of lessons. • Lesson details: Specific content within modules.
	Output: Structured course layout.

F4:	Setup Assessments
Input:	Assessment details (assignments, quizzes, exams).
Processing:	Instructors create and configure assessments.
Output:	Assessments integrated into the course.
Definition:	Inputs: <ul style="list-style-type: none"> • Assignments: Tasks for students to complete. • Quizzes: Short assessments. • Exams: Comprehensive tests. Output: Assessment availability for students.

F5:	Manage Course Settings
Input:	Course settings (enrollment rules, duration, prerequisites).
Processing:	Instructors configure course settings.
Output:	Updated course settings.
Definition:	Inputs: <ul style="list-style-type: none"> • Enrollment rules: Criteria for joining the course. • Duration: Length of the course. • Prerequisites: Required prior knowledge or courses. Output: Confirmation of updated settings.

Explanation

Course management in Padama empowers instructors to create rich, engaging courses tailored to educational objectives. Instructors can define course structures, upload multimedia resources, and configure assessments to assess student learning. The intuitive interface facilitates efficient content organization and management, ensuring instructors can focus on delivering high-quality educational experiences.

3.2 Use Case 2: User Registration and Management

3.2.1 Description and Priority

User registration and management functionalities enable users to create accounts, authenticate securely, and manage profiles within Padama. This use case is fundamental for ensuring personalized user experiences and system security.

3.2.2 Functional Requirements

F6:	User Registration
Input:	Email address, password.
Processing:	Users enter registration details; system stores credentials.
Output:	User account creation confirmation.
Definition:	<div>Inputs:</div> <ul style="list-style-type: none">Email address: User's email for communication.Password: Secure password for account access. <div>Output: Account creation success message.</div>

F7:	User Credential Verification
Input:	Email address, password.
Processing:	System checks credentials against stored data.
Output:	Access granted or denied.

Definition:	Inputs: <ul style="list-style-type: none"> Email address: User's registered email. Password: User's entered password. Output: Authentication success or failure message.
-------------	--

F8:	Update User Profiles
Input:	Personal information, preferences.
Processing:	Users modify their profile details.
Output:	Updated user profile.
Definition:	Inputs: <ul style="list-style-type: none"> Personal information: Name, contact details, etc. Preferences: User's settings and preferences. Output: Profile update confirmation.

F9:	Role-Based Access Control
Input:	User roles, permissions.
Processing:	System assigns permissions based on roles.
Output:	Access to functionalities based on role.
Definition:	Inputs: <ul style="list-style-type: none"> User roles: Designated roles for users. Permissions: Access rights for each role. Output: Role-specific access controls.

F10:	Forgot Password (Authentication and Access Control)
Input:	email address for initiating password reset

Processing:	The system sends a password reset link or code to the user's registered email address. The user clicks the link or enters the code to access a password reset page, verifies their identity, and creates a new password.
Output:	Password reset confirmation and successful login with the new password.
Definition:	Inputs:
	<ul style="list-style-type: none"> Username or email address: Identifier for the user requesting the password reset. <p>Outputs:</p> <ul style="list-style-type: none"> Password reset link or code: Sent to the user's email for initiating the reset process. Password reset confirmation: Notification that the password has been successfully reset.

Explanation

User registration and management features in Padama streamline account creation and authentication processes, ensuring secure access for all users. Role-based access control enhances data security by assigning appropriate permissions to users based on their roles within the educational environment. Users can manage their profiles, update information, and engage with Padama's functionalities based on their roles and responsibilities.

3.3 Use Case 3: Content Delivery

3.3.1 Description and Priority

Content delivery functionalities facilitate the distribution and access of course materials to users within Padama. This use case is critical for delivering educational content effectively and enhancing learning experiences.

3.3.2 Functional Requirements

F11:	Upload and Storage of Course Materials
Input:	Course materials (documents, presentations, multimedia files).
Processing:	Instructors upload materials to the system.
Output:	Stored course materials accessible to students.
Definition:	Inputs:
	<ul style="list-style-type: none">• Documents: Text files, PDFs, etc.• Presentations: Slideshows, PowerPoint files, etc.• Multimedia files: Videos, audio files, images, etc. Output: Confirmation that materials are stored and accessible.

F12:	Organizing and Categorizing Content
Input:	Course content (modules, lessons, topics).
Processing:	Instructors organize and categorize content.
Output:	Organized course structure.
Definition:	Inputs:
	<ul style="list-style-type: none">• Modules: Major sections of a course.• Lessons: Subsections within modules.• Topics: Specific subjects within lessons. Output: Structured and categorized course content.

F13:	File Format Compatibility
Input:	Various file formats and multimedia content.
Processing:	System ensures compatibility with input formats.
Output:	Compatible content available for use.

Definition:	Inputs: <ul style="list-style-type: none"> • File formats: Types of files supported (e.g., .doc, .pdf, .mp4). • Multimedia content: Videos, audio, interactive elements. Output: Files and content that work seamlessly within the system.
-------------	--

F14:	Optimized Content Delivery
Input:	Course materials and user devices.
Processing:	System optimizes content delivery for different devices and platforms.
Output:	Seamless content access across devices.
Definition:	Inputs: <ul style="list-style-type: none"> • Course materials: Documents, presentations, multimedia files. • User devices: Various devices such as PCs, tablets, smartphones. Output: Optimized display and performance of content.

Explanation

Content delivery in Padama ensures accessibility and usability of educational resources across diverse learning environments. Instructors can upload and organize course materials, ensuring compatibility with multimedia formats and accessibility standards. The system optimizes content delivery to support seamless access and interaction for users, promoting enhanced engagement and learning outcomes within Padama's educational ecosystem.

3.4 Use Case 4: Progress Tracking

3.4.1 Description and Priority

Progress tracking functionalities enable instructors to monitor and evaluate user progress and performance within Padama. This use case is critical for assessing learning outcomes and providing personalized feedback to users.

3.4.2 Functional Requirements

F15:	Tracking User Progress
Input:	User activities (course completion, module progress, assignment submissions).
Processing:	System tracks and records user progress.
Output:	Progress data accessible to users and instructors.
Definition:	Inputs: <ul style="list-style-type: none">• User activities: Actions taken by users within the course.
	Output: Reports and metrics showing user progress.

F16:	Reports and Analytics
Input:	User engagement, participation, performance data.
Processing:	System generates reports and analytics.
Output:	Detailed reports and performance metrics.
Definition:	Inputs: <ul style="list-style-type: none">• User engagement: Interaction metrics.• Participation: Attendance and activity levels.• Performance data: Scores, grades, completion rates. Output: Visual and textual reports summarizing data.

F17:	Personalized Feedback Mechanisms
Input:	Instructor feedback, student submissions.
Processing:	Instructors provide personalized feedback to students.
Output:	Feedback accessible to students.

Definition:	Inputs: <ul style="list-style-type: none"> • Instructor feedback: Comments and suggestions. • Student submissions: Assignments, quizzes, exams. Output: Personalized comments and grades visible to students.
-------------	---

F18:	Data Visualization Tools
Input:	Progress and performance data.
Processing:	System visualizes data using graphs, charts, and other tools.
Output:	Visual representation of data.
Definition:	Inputs: <ul style="list-style-type: none"> • Progress data: Metrics on user advancement. • Performance data: Scores and grades. Output: Graphs, charts, and dashboards displaying data.

Explanation

Progress tracking in Padama empowers instructors with insights into user engagement and performance metrics, facilitating data-driven decision-making and personalized learning interventions. Reporting and analytics tools enable instructors to assess learning outcomes, identify areas for improvement, and optimize educational strategies within Padama. The system fosters a supportive learning environment where instructors can provide timely feedback, monitor progress, and promote student success effectively.

3.5 Use Case 5: Communication Tools

3.5.1 Description and Priority

Communication tools facilitate interaction and collaboration among users within Padama. This use case is essential for promoting engagement, knowledge sharing, and community building within the educational environment.

3.5.2 Functional Requirements

F19:	Discussion Forums
Input:	User posts and replies.
Processing:	Users initiate and participate in discussions.
Output:	Active discussion threads.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• User posts: Questions, comments, topics.• Replies: Responses to posts. <p>Output: Threaded discussions visible to users.</p>

F20:	Real-Time Chat Features
Input:	Chat messages.
Processing:	System facilitates real-time messaging.
Output:	Instant messages between users.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Chat messages: Text communications between users. <p>Output: Real-time chat interface and message history.</p>

F21:	Messaging Capabilities
Input:	Private messages.
Processing:	System handles user-to-user messaging.
Output:	Private messages sent and received.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Private messages: Direct communications between users. <p>Output: Inbox and sent messages accessible to users</p>

F22:	Notification Systems
Input:	Updates, messages, course activities.
Processing:	System sends notifications to users.
Output:	Alerts and notifications.
Definition:	Inputs: <ul style="list-style-type: none"> • Updates: System changes, new content. • Messages: User communications. • Course activities: Assignment due dates, new discussions. Output: Pop-up alerts, email notifications, system messages.

Explanation

Communication tools in Padama enhance collaboration and interaction among users, fostering a supportive and engaging learning community. Discussion forums enable academic discussions, knowledge sharing, and peer-to-peer interactions among students and instructors. Real-time chat and messaging features facilitate instant communication, promoting collaborative projects, and providing academic support within Padama's virtual learning environment. Notification systems ensure users stay informed about course activities, updates, and relevant communications, enhancing user engagement and communication efficiency.

3.6 Use Case 6: File Storage and Management

3.6.1 Description and Priority

File storage and management functionalities enable users to upload, store, and manage files securely within Padama. This use case is essential for supporting course materials, assignments, and collaborative projects effectively.

3.6.2 Functional Requirements

F23:	File Storage Feature
Input:	User files.
Processing:	Users upload and store files in the system.

Output:	Stored files accessible to users.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> User files: Documents, images, multimedia content. <p>Output: Securely stored files with user access.</p>

F24:	Organizing Files
Input:	User files and folder structure.
Processing:	Users organize files into folders.
Output:	Efficiently managed files and folders.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> User files: Documents, images, multimedia content. Folder structure: User-created organization system. <p>Output: Hierarchical organization of files and folders.</p>

F25:	Access Control for Files
Input:	File permissions settings.
Processing:	Users set view and edit permissions.
Output:	Controlled access to files.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> File permissions: Read, write, edit settings. <p>Output: Access control lists (ACLs) defining permissions.</p>

F26:	Secure Storage and Access
Input:	User files, security protocols.
Processing:	System ensures secure storage and access.
Output:	Protected user data and privacy.

Definition:	Inputs: <ul style="list-style-type: none"> • User files: Documents, images, multimedia content. • Security protocols: Encryption, authentication methods. Output: Secure storage system and access logs.
-------------	--

Explanation

File storage and management in Padama facilitate secure storage, organization, and access to files associated with courses, assignments, and collaborative projects. Users can upload diverse file types, organize content into folders, and manage permissions for viewing and editing files. Access control mechanisms ensure data security and privacy, safeguarding user information within Padama's educational environment. The system supports efficient file management, version control, and collaborative workflows, enhancing productivity and collaboration among users.

4. Architectural Design

4.1 Non-functional Requirements

4.1.1 Performance Requirements

NFR 1:	Scalable User Access
Description:	The system shall robustly support concurrent access by up to 10,000 users simultaneously, ensuring seamless performance under peak load conditions.

NFR 2:	Dashboard Loading Time
Description:	The system shall load the dashboard within 3 seconds for at least 95% of users, optimizing user experience and efficiency.

NFR 3:	Efficient Data Transactions
--------	-----------------------------

Description:	Data transactions within the system shall be processed within 1 second for 95% of operations, maintaining responsiveness and operational speed.
--------------	---

4.1.2 Security Requirements

NFR :4	Data Encryption
Description:	The system shall utilize strong encryption protocols such as SSL/TLS to encrypt data transmitted between clients and servers, ensuring data integrity and confidentiality.

NFR 5:	Authentication and Authorization
Description:	Robust user authentication and authorization mechanisms shall be implemented to prevent unauthorized access and protect sensitive data.

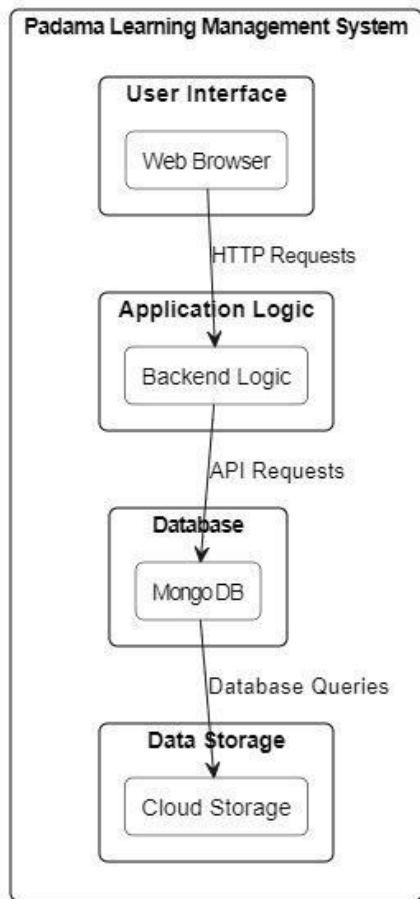
NFR 6:	Data Privacy
Description:	The system shall enforce strict access controls and permissions to safeguard sensitive information, ensuring compliance with data privacy regulations.

NFR 7:	Intuitive User Interface
Description:	The system shall feature a user-friendly interface with intuitive navigation and clear instructions, enhancing usability for all user roles.

NFR 8:	Device Compatibility
Description:	The system shall be accessible across desktop and mobile devices, offering a consistent and responsive user experience.

NFR 9:	Comprehensive Documentation
Description:	Comprehensive documentation and user guides shall be provided to support users in effectively utilizing all system features and functionalities.

4.2 Architectural Design



4.2.1 System Requirements

F27:	Browser Compatibility
Input:	Various web browsers (Chrome, Firefox, Safari, Edge).
Processing:	System ensures compatibility.
Output:	Accessible and usable system across major browsers.
Definition:	<div>Inputs:<ul style="list-style-type: none">Web browsers: Different types of browsers used by users.<div>Output: Consistent user experience across browsers.</div></div>

F28:	Responsive Design
Input:	Various mobile devices.
Processing:	System adapts interface layouts and functionalities.
Output:	Optimized user experience on mobile devices.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> • Mobile devices: Smartphones, tablets. <p>Output: Responsive design that adjusts to screen sizes.</p>

F29:	Accessibility Standards
Input:	WCAG 2.0 guidelines.
Processing:	System adheres to accessibility standards.
Output:	Inclusive and accessible system for users with disabilities.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> • WCAG 2.0 guidelines: Web Content Accessibility Guidelines. <p>Output: Compliance with accessibility standards.</p>

F30:	Logging and Monitoring
Input:	System activities and performance data.
Processing:	System logs and monitors activities.
Output:	Real-time troubleshooting, performance analysis data.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none"> • System activities: User actions, system events. • Performance data: Metrics on system operation. <p>Output: Logs and monitoring dashboards.</p>

4.3 Design and Implementation

4.3.1 System Requirements

F31:	Modular Architecture
Input:	System components and modules.
Processing:	System architecture designed for modularity.
Output:	Scalable, maintainable, and upgradeable system.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• System components: Independent parts of the system. <p>Output: Modular system architecture.</p>

F32:	API Integration
Input:	External APIs.
Processing:	System integrates with external APIs.
Output:	Enhanced functionality and seamless data exchange.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• External APIs: Application Programming Interfaces from other systems. <p>Output: Integrated features and data flows.</p>

F33:	Caching Mechanisms
Input:	Frequently accessed data.
Processing:	System caches data.
Output:	Optimized performance and reduced retrieval times.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Frequently accessed data: Data that is often requested by users. <p>Output: Improved system responsiveness.</p>

4.4 Database Design

4.4.1 Data Requirements

F34:	Database Management System
Input:	User data, course information, system configurations.
Processing:	Data stored and managed using MongoDB.
Output:	Efficient data storage and management.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• User data: Personal information, progress, preferences.• Course information: Titles, descriptions, materials.• System configurations: Settings, preferences. <p>Output: Structured and managed database.</p>

F35:	Scalable Data Storage
Input:	Growing user base and course catalog.
Processing:	Database supports scalable storage and querying.
Output:	Seamless accommodation of expanding data.
Definition:	<p>Inputs:</p> <ul style="list-style-type: none">• Growing user base: Increasing number of users (students, instructors, administrators).• Course catalog: Expanding number of courses offered within the system. <p>Output:</p> <ul style="list-style-type: none">• Scalable data storage: The ability to store increasing amounts of data efficiently.• Querying capabilities: The ability to retrieve and manage data effectively as the database grows.

5. Project Plan

5.1 Project Schedule

Milestones and Deliverables

Phase	Activities and Deliverables	Timeline
Phase 1	Requirements Gathering and Analysis	Aug 1, 2024-Aug 15, 2024
	- Completion of SRS document	
	- Approval by stakeholders	
Phase 2	System Design	Aug 16, 2024 - Aug 31, 2024
	- Architectural design	
	- UI/UX design	
	- Database schema design	
Phase 3	Development	Sep 1, 2024-Sep 30, 2024
	- Frontend and backend development	
	- Integration of system modules	
Phase 4	Testing	Oct 1, 2024-Oct 15, 2024

	- Unit testing	
	- Integration testing	
	- User acceptance testing	
Phase 5	Deployment	Oct 16, 2024-Oct 31, 2024
	- System deployment	
	- User training sessions	
	- Production launch	

5.2 Project Budget

Estimated Costs

Category	Details	Estimated Amount (LKR)
Development Costs	Salaries Software licenses Development tools	2,500,000.00
Infrastructure Costs	Cloud hosting Server maintenance Database management	500.000.00

Testing Costs	QA resources Testing tools Performance testing	300,000.00
Training Costs	User training sessions Documentation development	200,000.00
Total		<u>3,500,000.00</u>

References

ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

- <https://www.iso.org/standard/35733.html>

W3C Web Content Accessibility Guidelines (WCAG) 2.0

- <https://www.w3.org/TR/WCAG20/>

MongoDB Documentation

- <https://docs.mongodb.com/>

Amazon Web Services (AWS) Documentation

- <https://docs.aws.amazon.com/>

Project Management Institute (PMI) - Project Management Body of Knowledge (PMBOK) Guide

- <https://www.pmi.org/pmbok-guide-standards>

Appendices

Appendix A: Glossary

1. **Learning Management System (LMS)**
 - A software application for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, or learning and development programs.
2. **Software Requirements Specification (SRS)** ○ A document that captures the complete software requirements for the system, or a portion of the system.
3. **System Architect** ○ A person responsible for defining the overall structure of a software system, including its components and their interactions.
4. **Quality Assurance (QA)**
 - A way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers.
5. **WCAG 2.0**
 - Web Content Accessibility Guidelines developed through the W3C process in cooperation with individuals and organizations around the world, with a goal of providing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments internationally.
6. **MongoDB**
 - A source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.
7. **Amazon Web Services (AWS)** ○ A subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.
8. **Project Management Body of Knowledge (PMBOK)** ○ A set of standard terminology and guidelines for project management, maintained by the Project Management Institute (PMI).

Appendix B: Acronyms

- **LMS:** Learning Management System
- **SRS:** Software Requirements Specification
- **QA:** Quality Assurance
- **WCAG:** Web Content Accessibility Guidelines
- **AWS:** Amazon Web Services
- **PMBOK:** Project Management Body of Knowledge