SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Student Learning Platform (SLP) for ICT and Computer Science

EduTech

Smart Learning, Better Results

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of the requirements for the Student Learning Platform (SLP) called **EduTech** with slogan **Smart Learning, Better Results**. It is intended to serve as a reference for developers, system architects, project managers, testers, and stakeholders involved in the design, development, and deployment of the system.

1.2 Scope

The Student Learning Platform (SLP) is a web-based educational system designed to support Ordinary Level (O/L) and Advanced Level (A/L) students studying ICT and Computer Science. The platform enables students to register, subscribe to learning plans, access structured learning materials, complete assessments, track progress, and interact with a learning community.

1.3 Definitions, Acronyms, and Abbreviations

- **SLP** – Student Learning Platform
- **ICT** – Information and Communication Technology
- **O/L** – Ordinary Level

- **A/L** – Advanced Level
- **SRS** – Software Requirements Specification

2. Overall Description

2.1 Product Perspective

The SLP is a standalone web-based application accessible through modern web browsers. It may integrate with third-party services such as payment gateways, email notification systems, and video hosting platforms.

2.2 Product Functions

The main functions of the system include:

- User registration and authentication
- Subscription management
- Delivery of learning materials
- Online assessments and feedback
- Progress tracking and certification
- Discussion forums and support services

2.3 User Classes and Characteristics

- **Students**: Primary users who access learning content, complete assessments, and track progress.
- **Instructors/Content Creator**: Manage content, assessments.
- **System Administrator**: Manages users, subscriptions, platform usage and system settings.

2.4 Operating Environment

- **Frontend:** HTML5, CSS3, JavaScript, React
- **Backend:** Node.js
- **Database server** (e.g., MySQL, PostgreSQL)

2.5 Design and Implementation Constraints

- Web-based architecture
- Secure handling of user data and payments
- Compliance with basic data protection regulations

2.6 Assumptions and Dependencies

- Users have access to the internet
- Payment gateway availability for subscription services
- Content availability from instructors

3. Functional Requirements

3.1 User Registration and Authentication

- The system shall allow users to create an account using email, name, password and educational level (O/L or A/L).
- The system shall validate user credentials during login.
- The system shall allow users to reset forgotten passwords.

3.2 Subscription Management

- The system shall provide multiple subscription plans.
- The system shall allow users to subscribe, renew, or cancel subscriptions.
- The system shall restrict access to premium content based on subscription status.

3.3 Learning Materials

- The system shall provide structured notes for ICT and Computer Science topics.
- The system shall support multimedia content such as videos and infographics.
- The system shall allow instructors/admins to upload and manage learning materials.

- The system shall allow users to bookmark/save content for later access.

3.4 Assessments and Practice

- The system shall allow users take quizzes/question sets per topic
- The system shall provide immediate feedback on submitted answers.
- Question types shall include multiple-choice and short-answers.
- The system shall allow users review previous quiz attempts and reviews.

3.5 Progress Tracking and Certification

- The system shall track student progress across courses and modules, quiz scores and time spent.
- The system shall display progress reports to students(charts, completed percentages).
- The system shall generate certificates upon successful completion of courses or modules.

3.6 Community and Support

- The system shall provide discussion forums for students and instructors.
- The system shall allow users to post, reply, and moderate discussions.
- The system shall provide support resources such as FAQs and contact options.

3.7 Subscription Plans and Access Control

3.7.1 Overview

The Student Learning Platform shall implement a subscription-based access control mechanism to regulate user access to learning resources and platform features. Subscription plans act as feature gates that determine the scope, duration, and level of access available to each user.

3.7.2 Subscription Plan Types

a) Free Plan

The Free Plan shall be automatically assigned to users upon registration.

Features:

- Access to limited introductory notes
- Access to sample learning materials
- No access to assessments
- No access to certificates

Restrictions:

- Premium content shall be restricted
- Progress tracking shall be limited

b) Basic Plan

The Basic Plan shall provide entry-level paid access to learning resources.

Features:

- Access to all standard notes
- Limited assessments and practice questions
- Basic progress tracking
- Limited forum access

Restrictions:

- Certificate generation shall not be available
- Assessment attempts may be capped

c) Pro Plan

The Pro Plan shall provide full access to the platform and represents the primary revenue tier.

Features:

- Full access to all courses and modules
- Unlimited assessments and quizzes
- Certificate generation upon course/module completion
- Full progress analytics
- Full community forum access

3.7.3 Subscription Lifecycle Management

- The system shall allow users to subscribe to a plan via an online payment gateway.
- The system shall activate subscriptions upon successful payment confirmation.
- Each subscription shall have a defined start date, end date, and status (active, expired, cancelled).
- The system shall automatically downgrade users to the Free Plan upon subscription expiry.

3.7.4 Upgrade and Downgrade Policy

- The system shall allow users to upgrade their subscription plans with immediate effect.
- Subscription downgrades shall take effect at the end of the current billing cycle.
- Subscription billing adjustments shall be handled by the payment service provider.

3.7.5 Feature-Based Access Control

- The system shall enforce access control based on enabled subscription features.
- All premium actions (e.g., attempting assessments, downloading certificates) shall be validated on the server side.
- Client-side checks shall be used only for user interface convenience and shall not be relied upon for security.

3.7.6 Subscription Notifications

- The system shall notify users of successful subscriptions and renewals.
- The system shall notify users prior to subscription expiry.
- The system shall notify users upon subscription cancellation or downgrade.

4. Non-Functional Requirements

4.1 Performance Requirements

- The system shall support concurrent users without performance degradation.
- Page load times shall not exceed 3 seconds under normal conditions.

4.2 Security Requirements

- The system shall encrypt sensitive user data.
- Payment data shall not be store internally (handled by a payment gateway).
- Role-based access control (student, instructor, admin).

4.3 Usability Requirements

- The system shall have a user-friendly and intuitive interface.
- The system shall be accessible to users with basic computer literacy.

4.4 Reliability and Availability

- The system shall be available 99% of the time.

4.5 Scalability

- The system shall be scalable to accommodate an increasing number of users and content.

5. External Interface Requirements

5.1 User Interface

- **Home Page:** Clean layout with login/registration, featured content and testimonials
- **Dashboard:** Personal content feed, progress summary, recent activities.
- **Content Viewer:** Topic-wise materials with navigation (previous/next).
- **Quiz Interface:** Timed questions with submit/retry options.

5.2 Hardware Interface

- No specific hardware requirements beyond an internet-enabled device

5.3 Software Interface

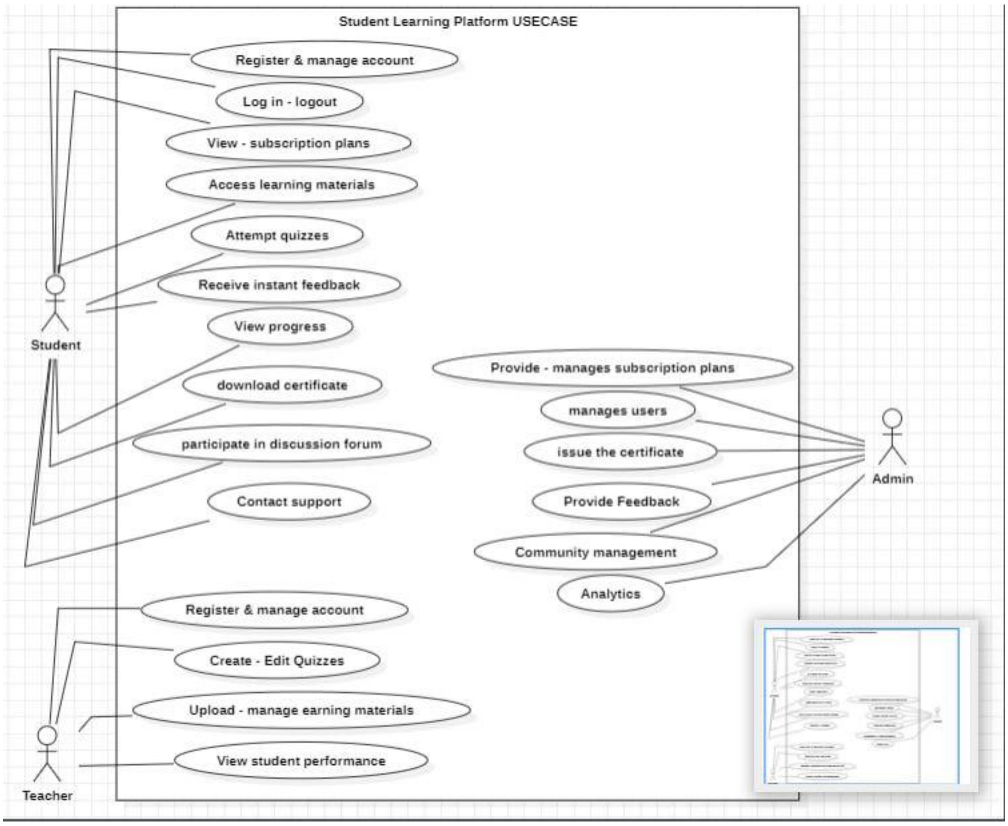
- Integration with payment gateways API
- Integration with email notification services

5.4 Communication Interface

- HTTPS for secure communication

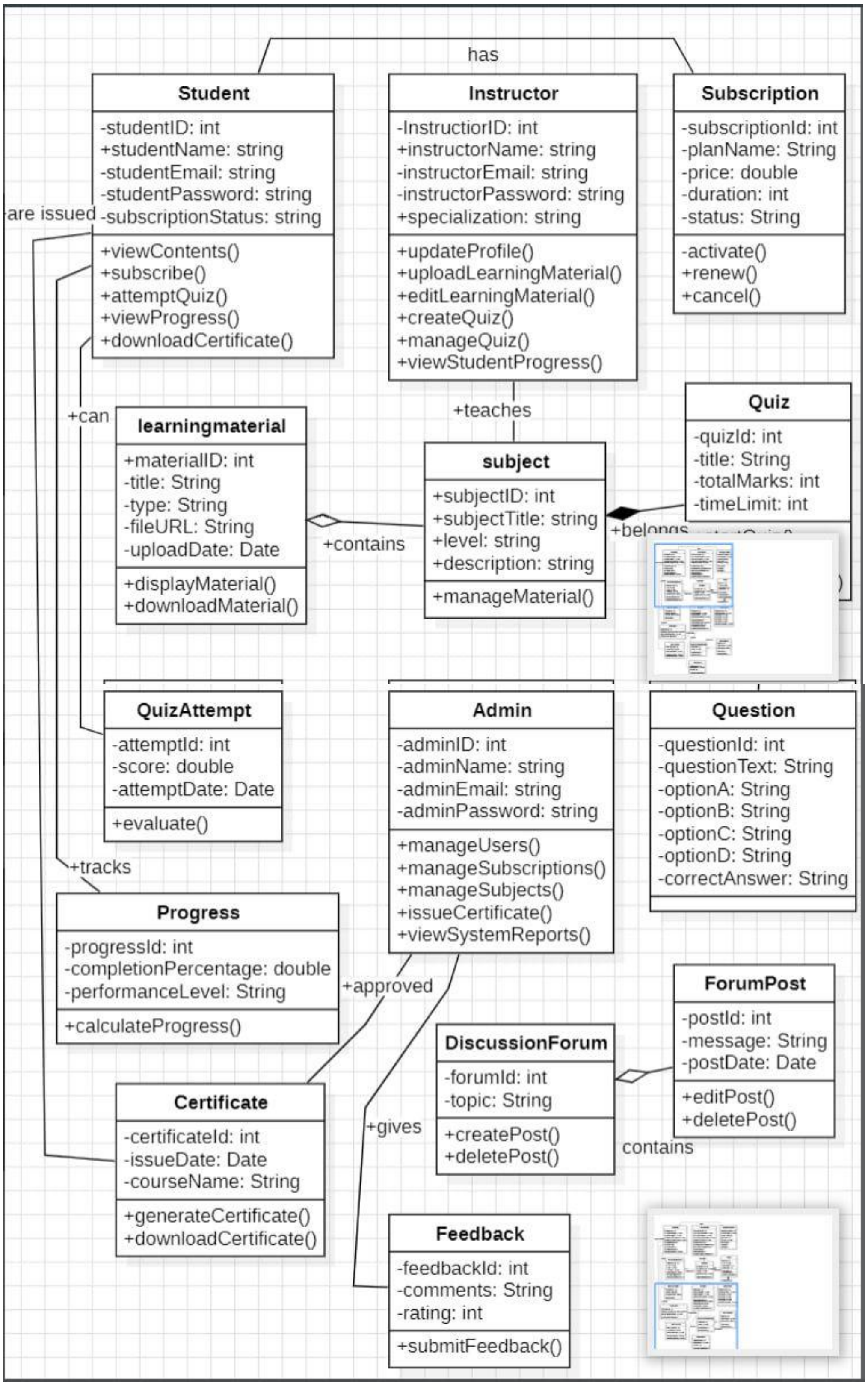
6. UML DIAGRAMS

6.1 Use Case Diagram



6.2 Class Diagram

The class diagram describes the static structure of the system.


has
Student
-studentID: int
+studentName: string
-studentEmail: string
-studentPassword: string
-subscriptionStatus: string
+viewContents()
+subscribe()
+attemptQuiz()
+viewProgress()
+downloadCertificate()
Instructor
-InstructiorID: int
+instructorName: string
-instructorEmail: string
-instructorPassword: string
+specialization: string
+updateProfile()
+uploadLearningMaterial()
+editLearningMaterial()
+createQuiz()
+manageQuiz()
+viewStudentProgress()
Subscription
-subscriptionId: int
-planName: String
-price: double
-duration: int
-status: String
-activate()
+renew()
+cancel()
are issued
+can
+teaches
learningmaterial
+materialID: int
-title: String
-type: String
-fileURL: String
-uploadDate: Date
+displayMaterial()
+downloadMaterial()
+contains
subject
+subjectID: int
+subjectTitle: string
+level: string
+description: string
+manageMaterial()
Quiz
-quizId: int
-title: String
-totalMarks: int
-timeLimit: int
+belongs
QuizAttempt
-attemptId: int
-score: double
-attemptDate: Date
+evaluate()
Admin
-adminID: int
-adminName: string
-adminEmail: string
-adminPassword: string
+manageUsers()
+manageSubscriptions()
+manageSubjects()
+issueCertificate()
+viewSystemReports()
Question
-questionId: int
-questionText: String
-optionA: String
-optionB: String
-optionC: String
-optionD: String
-correctAnswer: String
+tracks
Progress
-progressId: int
-completionPercentage: double
-performanceLevel: String
+calculateProgress()
+approved
ForumPost
-postId: int
-message: String
-postDate: Date
+editPost()
+deletePost()
DiscussionForum
-forumId: int
-topic: String
+createPost()
+deletePost()
contains
Certificate
-certificateId: int
-issueDate: Date
-courseName: String
+generateCertificate()
+downloadCertificate()
+gives
Feedback
-feedbackId: int
-comments: String
-rating: int
+submitFeedback()

7. Technology Stack

Category	**Language/Framework**
Frontend	Next.js/Typescript
Backend	Node.js
Database	PosgreSQL Cloudinary for videos/files etc

8. Future Enhancements

- AI-based personalized learning recommendations
- Live classes and real-time chat support