**1. Sprint Summaries**

This section outlines the development progress across several sprints, detailing completed tasks and future.

**Sprint 1: Project Initialization, Core Models, and Basic CRUD**

* **Tasks Done:**
  + Finalized project requirements and created the initial proposal document.
  + Set up the Spring Boot project with all necessary dependencies (Spring Web, Data JPA, Security, Thyme leaf).
  + Designed and implemented the core database models (Entities) for Student and Tutor.
  + Developed full CRUD (Create, Read, Update, Delete) functionality for Student Profile Management, accessible by an administrator.
  + Established the initial MySQL database connection and schema generation.

**Sprint 2: Feature Implementation & Security Integration**

* **Tasks Done:**
  + Developed the Lesson Management module, allowing Tutors to upload educational materials.
  + Implemented the Class Scheduling module, enabling Admins to create and manage class timetables, assign tutors, and enroll students.
  + Built the Attendance Management feature for Tutors to mark student attendance.
  + Integrated Spring Security to establish a role-based authentication system (ADMIN, TUTOR, STUDENT, PARENT).
  + Created the core User and Role entities to handle login and permissions.

**Sprint 3: User Dashboards, Portals, and Communication**

* **Tasks Done:**
  + Developed the **Tutor Dashboard**, showing assigned classes.
  + Developed the **Student Dashboard**, showing enrolled classes and available lessons.
  + Built the **Parent Portal**, allowing parents to view their children's attendance records.
  + Implemented a simple, text-based **messaging system** for communication between Parents and Tutors.
  + Created a universal public registration system with an **Admin approval workflow** for Tutor and Parent roles, enhancing security.

**Sprint 4: UI Refinement, Final Testing, and Bug Fixing**

* **Tasks Done:**
  + Designed and implemented a professional, themed UI for all public-facing pages (Homepage, Login, Register).
  + Created and applied a consistent master layout and theme to all internal application pages for a cohesive user experience.
  + Conducted thorough testing of all features across all user roles.
  + Identified and fixed major bugs related to database constraints (foreign key errors on delete) and UI inconsistencies.
  + Finalized all CRUD operations, including adding missing "Update" and "Delete" functionalities.

**Future Sprints**

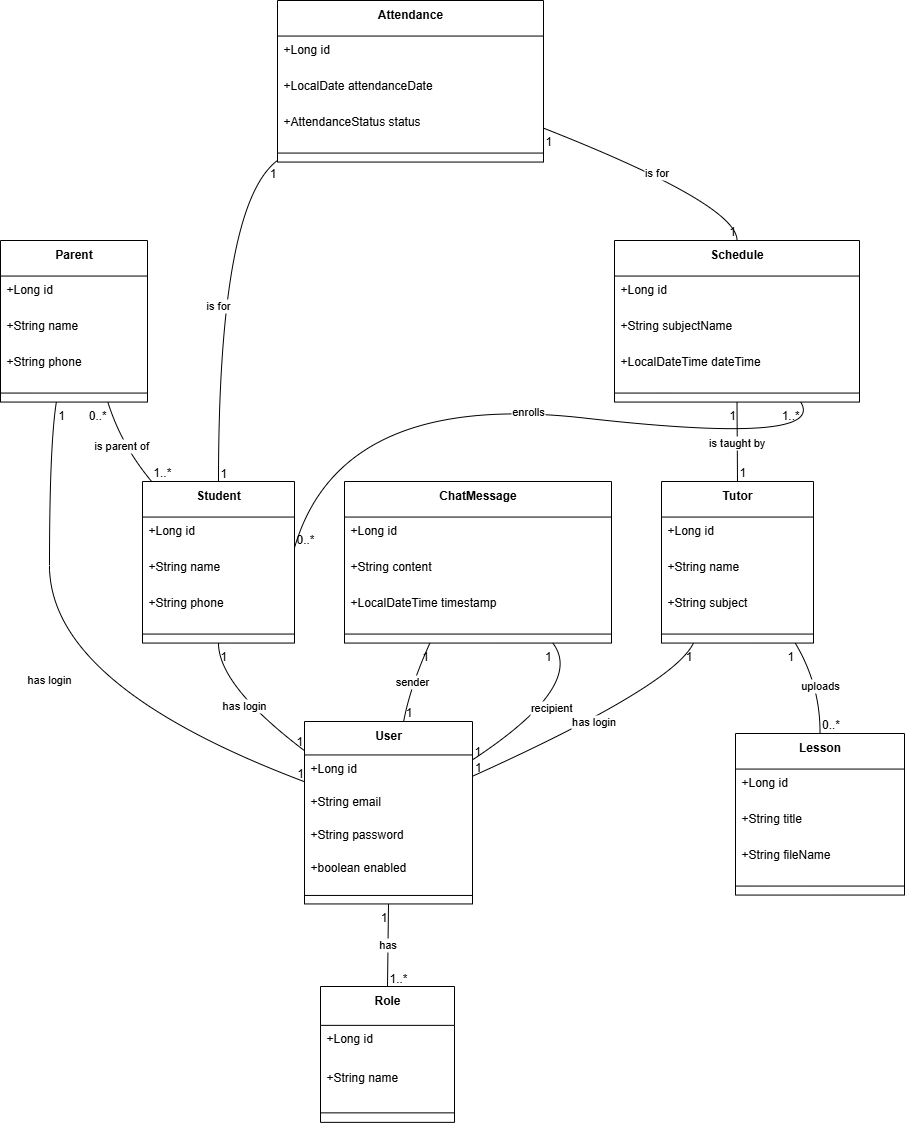
* **Payments Module:** Integrate a payment gateway to manage and track student fee payments.
* **Notifications System:** Implement an automated email or in-app notification system for class reminders, cancellations, and new messages.
* **Reporting Module:** Develop a feature for Admins to generate reports on attendance, finances, and student performance.

**2. Final Use Case Diagram**

A screen shot of a cell phone

AI-generated content may be incorrect.

**3. Class Diagram**



**4. Sequence Diagrams**

**1. Student Profile Management – IT24100958**

**Use Case:** Admin updates a student's profile.

A black screen with white text

AI-generated content may be incorrect.

**2. Lesson Management – IT24101454**

**Use Case:** Tutor uploads a new lesson.

A black screen with white text

AI-generated content may be incorrect.

**3. Access Lessons – IT24101277**

**Use Case:** Student views the list of available lessons from their dashboard.

A black screen with white text

AI-generated content may be incorrect.

**4. Class Scheduling – IT24101149**

**Use Case:** Admin creates a new class schedule.

A black screen with white text

AI-generated content may be incorrect.

**5. Attendance Management (Dulith)**

**Use Case:** Tutor marks attendance for a class.

A black screen with white text

AI-generated content may be incorrect.

**6. Tutor Dashboard & Communication – IT24101261**

**Use Case:** Parent sends a message to a Tutor.

A screenshot of a computer

AI-generated content may be incorrect.

**5. Ethical Considerations**

The design and implementation of the Tuition Management System prioritize ethical standards, focusing on user privacy, informed consent, and data security.

* **Privacy and Data Minimization:** The system is built on a principle of role-based access control. Each user role (Admin, Tutor, Student, Parent) has access only to the data necessary for their function. For example, a Parent can only view information related to their own linked children, and a Tutor can only manage the students within their assigned classes. This minimizes unnecessary data exposure and protects student privacy.
* **Consent:** All users (or their guardians) must actively provide consent by completing the registration process. For roles with access to sensitive information, such as Tutors and Parents, a two-step consent process is implemented: initial registration followed by mandatory approval from a System Administrator. This ensures that all privileged users are vetted and have been granted explicit permission to access the system.
* **Data Security:** User security is paramount. All passwords are not stored in plain text. Instead, they are securely hashed using the industry-standard BCrypt algorithm. This one-way encryption ensures that even in the unlikely event of a database breach, user passwords remain protected and cannot be reverse engineered.
* **Accessibility:** The user interface is built using standard web technologies (HTML and Bootstrap), ensuring a baseline level of accessibility. The design avoids overly complex elements, uses clear navigation, and maintains a consistent layout, making it usable for individuals with varying levels of digital literacy. Future iterations will focus on further improving compliance with Web Content Accessibility Guidelines (WCAG).