Architectural Design Document for Student Tracker
Version: 1.0
Last Updated: 12/01/2025

1. System Overview

The Student Tracker is a web application designed to track the location and hours of students during field training. It allows students to check in via geolocation and allows instructors to monitor their assigned students and exchange messages.

2. High-Level Architecture

The system follows a standard 3-Layer Monolithic Architecture. It is self-contained and hosted as a single Blazor Server instance.

- Presentation Layer (UI): Blazor Components handling user interaction.
- Service/Data Layer: Repository classes that handle business logic and database commands.
- Database Layer: A local SQLite database managed via Entity Framework Core.

Technology Stack

- Framework: .NET 8 Blazor Server
- Database: SQLite
- ORM: Entity Framework Core

3. Key Components

3.1 User Interface (Frontend)

The UI is divided into role-based views managed by a shared layout:

- Public: Login and Account Creation.
- Student View: Dashboard for "Check-In" actions and viewing personal logs.
- Instructor View: Dashboard for viewing a list of assigned students and their status.
- Communication: A shared messaging interface for Student-Instructor chat.

3.2 Logic & Data Access (Backend)

The application uses the Repository Pattern to decouple the UI from the database.

- User Repository: Handles authentication, registration, and role assignment.
- Check-In Repository: Receives raw Geolocation data from the UI and timestamps it into the database.
- Association Repository: Manages the link between a specific Student and a specific Instructor.
- Message Repository: Handles the persistent chat history between users.

4. Data Design

The SQLite database consists of five core entities:

Entity	Purpose
User	User contains all the base information for every account.
StudentTeacher	Stores studentid and teacherid so there is a relationship.
CheckInLog	Stores Timestamp, Latitude, Longitude, Notes, and UserID.

Message	Stores SenderID, ReceiverID, Message, and Timestamp.

5. Security & Workflow

- Authentication: Custom session management using an in-memory state provider.
- Authorization: The UI renders specific aspects of the dashboard based on the UserRole stored in the session.
- Data Privacy: Instructors are strictly limited to viewing the location logs of students explicitly assigned to them in the database.