Version & Metadata

Version: v1.0

Title: PeerLearn — Use Cases

Project: PeerLearn (Personal project)

Author: Aleksandar Ivanov

Date: 2025-09-11 (Europe/Amsterdam)

1) Purpose

This document lists the **functional use cases** for PeerLearn, aligned with the technical stack decisions (ASP.NET Core, React, SQL, MongoDB, SignalR, Yjs, Slate, WebRTC). Each use case defines the actors, scenarios, and supporting technology with external references.

2) Use Cases

UC1 — User Authentication & Room Management

- **Actors:** Student, System.
- **Scenario:** A student signs up/logs in with email and password. The backend validates credentials and issues a JWT stored in an HttpOnly cookie. The student can create or join rooms.
- **Tech Justification:** ASP.NET Core Identity + JWT bearer. ASP.NET Core Identity OWASP HttpOnly

UC2 — Collaborative Note Editing

- **Actors:** Students in a room.
- **Scenario:** Multiple users edit the same shared note in real time. All changes are merged seamlessly and reflected for all participants.
- **Tech Justification:** React + Slate rich-text editor, synced via Yjs CRDT with `y-websocket`. Slate docs Yjs docs y-websocket docs

UC3 — Real-Time Chat

- **Actors:** Students.
- **Scenario:** Students send and receive instant text messages in a room while editing notes or studying.
- **Tech Justification:** ASP.NET Core SignalR hubs with React client. SignalR intro

UC4 — Flashcard Generation

- **Actors:** Student, System.
- **Scenario:** A student highlights text in notes and clicks “Generate Flashcards.” The system calls an AI API to produce Q&A pairs.
- **Tech Justification:** OpenAI API or Hugging Face Inference Endpoints. OpenAI API Hugging Face

UC5 — Quiz Taking & Progress Tracking

- **Actors:** Student.
- **Scenario:** A student answers flashcard-based quizzes. Scores are recorded, XP is updated, and progress is displayed in the dashboard.
- **Tech Justification:** MongoDB for storing quiz results and progress metrics. MongoDB .NET driver

UC6 — Voice Co-Study (Optional, Post-MVP)

- **Actors:** Students.
- **Scenario:** Students start a voice channel in a study room for live discussions.
- **Tech Justification:** WebRTC peer connections with TURN relay for NAT traversal. WebRTC API (MDN) TURN (MDN) coturn project

3) References

- ASP.NET Core Identity: https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-9.0
- OWASP HttpOnly: https://owasp.org/www-community/HttpOnly
- Slate docs: https://docs.slatejs.org
- Yjs docs: https://docs.yjs.dev
- y-websocket docs: https://docs.yjs.dev/ecosystem/connection-provider/y-websocket
- SignalR intro: https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-9.0
- OpenAI API: https://platform.openai.com/docs/api-reference
- Hugging Face endpoints: https://huggingface.co/docs/inference-endpoints/index
- MongoDB .NET driver: https://www.mongodb.com/docs/drivers/csharp/
- WebRTC API (MDN): https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
- TURN (MDN): https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Protocols#turn
- coturn: https://github.com/coturn/coturn