AI Study Buddy — Use Case Details

Date: 2025-10-09 19:19 UTC

Detailed tables for the four requested use cases.

## Login & Registration

|  |  |
| --- | --- |
| Primary actor(s) | Student; Email Service; optional SSO Provider |
| Preconditions | User not logged in; network available; account may or may not exist. |
| Trigger | User clicks 'Log in' or 'Sign up'. |
| Postconditions | User is authenticated; refresh token stored securely; audit log updated. |
| Non-functional notes | Security: rate-limit, lockout after N attempts; refresh rotation; JWT expiry; CSRF-safe refresh endpoint; accessibility AA forms. |

Main Success Scenario:

1. 1. System shows login/registration form.
2. 2. User submits credentials or SSO intent.
3. 3. System validates input; on sign-up, creates account in DB.
4. 4. System sends verification email with time-limited token.
5. 5. User verifies email; system marks account verified.
6. 6. System issues access token + refresh token; session started.

Alternate / Exception Flows:

* A1: Wrong password
* - Reject login; increment attempt count; show safe error; offer reset.
* A2: Unverified account
* - Prompt to verify email; resend verification token; limit resends.
* A3: Password reset
* - User requests reset; system emails link; user sets new password.
* A4: SSO failure
* - Show fallback to username/password; log event.

## Create Module & Add Notes

|  |  |
| --- | --- |
| Primary actor(s) | Student; Object Storage; Background Worker |
| Preconditions | User is authenticated and has at least one course/module context. |
| Trigger | User selects 'New Module' or opens a module and selects 'Add note'. |
| Postconditions | Module exists with associated notes; searchable with embeddings; artefacts stored in object storage. |
| Non-functional notes | Performance: uploads stream and never hit DB; reliability via queued processing; signed URLs for downloads. |

Main Success Scenario:

1. 1. System shows module form (name, semester, tags).
2. 2. User submits; system creates module record.
3. 3. User selects 'Add note' and uploads file(s) or pastes text.
4. 4. API streams file to object storage; returns Job ID.
5. 5. System enqueues processing job (OCR/ASR if needed).
6. 6. Worker processes file; generates summary + embeddings; updates DB.
7. 7. System notifies user when ready; note appears in module.

Alternate / Exception Flows:

* B1: Unsupported file
* - Reject with message; show allowed types; keep UI state for retry.
* B2: Large file
* - Use chunked upload; show progress; allow resume.
* B3: Processing failed
* - Mark job failed; expose retry; capture error for observability.

## Pomodoro Timer while Studying

|  |  |
| --- | --- |
| Primary actor(s) | Student; Notification Service |
| Preconditions | User authenticated and inside study session view. |
| Trigger | User clicks 'Start Pomodoro'. |
| Postconditions | Study stats updated; streaks/badges awarded where applicable; next session ready. |
| Non-functional notes | Usability: large, accessible controls; offline-safe tick; resilient to tab loss; P95 drift < 2s per 25-min block. |

Main Success Scenario:

1. 1. System starts 25-minute timer; shows remaining time.
2. 2. At end, system triggers 5-minute break; play alert/notification.
3. 3. After 4 sessions, schedule long break.
4. 4. System logs completed focus/break blocks to user stats.
5. 5. User can pause/adjust presets within allowed limits.

Alternate / Exception Flows:

* C1: Browser sleep/tab closed
* - Persist timer server-side or via service worker; resync on return.
* C2: Notification disabled
* - Fallback to in-app sound/banner; prompt to enable notifications.

## Seek AI Help while Studying

|  |  |
| --- | --- |
| Primary actor(s) | Student; AI Provider |
| Preconditions | User authenticated; relevant notes exist with embeddings; daily usage quota available. |
| Trigger | User asks a question from study view (optionally highlights text). |
| Postconditions | Answer shown with citations; saved QA available later; analytics updated. |
| Non-functional notes | Quality: cite sources; latency target < 4s with streaming; privacy — redact PII in prompts. |

Main Success Scenario:

1. 1. System retrieves top-k relevant chunks (pgvector).
2. 2. System sends prompt with citations to AI provider.
3. 3. AI responds; system formats answer with sources and follow-ups.
4. 4. User can rate/save answer; content added to knowledge base.
5. 5. System adapts suggestions based on weak areas.

Alternate / Exception Flows:

* D1: No relevant context
* - Answer with general guidance and ask to upload notes; suggest module.
* D2: AI provider error/rate limit
* - Retry with backoff; switch provider; show gentle message.
* D3: Safety/guardrails
* - Filter prompts/outputs; avoid disallowed content; log for review.