# Bilkent University
# Department of Computer Engineering

## Senior Design Project

**Team ID:** T2521

**Project Name:** Ubien: The Immersive Inquiry Based Learning Platform

## Analysis and Requirement Report

**Group Members:**

*22203367, Ahmet Deniz Gelir, deniz.gelir@ug.bilkent.edu.tr*
22202254, Efe Can Tatar, can.tatar@ug.bilkent.edu.tr
22201820, Kemal Onur Özkan, onurozkan@ug.bilkent.edu.tr
22001736, Eren Uslu, eren.uslu@ug.bilkent.edu.tr
22203328,Simay Uygur, simay.uygur@ug.bilkent.edu.tr

**Supervisor:**

Uğur Güdükbay

**Course Instructors:**

İlker Burak Kurt

Mert Bıçakçı

**19.12.2025**

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

# Contents

# Analysis and Requirement Report

*Project Short-Name: Ubien*

## 1. Introduction

It is a commonplace in the present day to hear students bemoan that while they have “learned” a great deal on paper, they have retained only a fraction of the information they gathered over the course of their education. It is natural, then, to ask: what has been the point of learning all that information if it functioned merely as a temporary exercise in memorization? The current paradigms of education, from structured lectures to solution-driven problem sets and exam-oriented milestones, are built around the efficient delivery and assessment of knowledge rather than the cultivation of genuine understanding. In such systems, students often become passive recipients, carrying information only long enough to reproduce it when asked, and quickly losing it afterward.

This widespread pattern reveals a fundamental problem: the absence of active self-discovery. Educational psychology and centuries of pedagogical tradition alike affirm that concepts uncovered through one’s own reasoning are remembered more deeply, understood more clearly, and integrated more permanently into one’s intellectual framework [1]. Yet most modern educational tools unintentionally short-circuit this process. They present polished explanations, prepackaged insights, and step-by-step solutions that resolve uncertainty too quickly, preventing learners from engaging with the material in a way that forms resilient mental models.

Ubien arises from the recognition that true understanding occurs when students do not merely receive ideas, but are made to come up with it themselves. Drawing on the principles of Inquiry-Based Learning (IBL), Ubien transforms traditional IBL textbooks into interactive courses where students iteratively advance through answering a sequence of questions. Instead of being told what to think, learners are socratically questioned into discovering why concepts hold together and how the structures of a discipline emerge organically from first principles.

In this environment, knowledge is not an external artifact handed down by an instructor, it becomes something the student builds. The result is a form of understanding that is permanent, self-sourced, and meaningfully tied to the learner’s own cognitive efforts. Ubien seeks to restore this mode of discovery-based learning to the digital age, offering a platform where students engage with material in a way that mirrors the curiosity-driven process that underlies real intellectual growth.

## 2. Current System

The current systems in place that incorporate artificial intelligence into a learning platform allow you to chat with a generative AI model that aids the user to understand a topic by giving summaries and re-explaining in different words and with different examples. Although this is a valid learning strategy for short-term cramming and getting higher grades, Ubien approaches the learning from a different perspective.

Ubien incorporates IBL principles into the learning cycle of the user since real learning occurs when you are pushed to come up with the solution yourself. Instead of being given a list to memorize as fast as possible, Ubien will push users towards understanding why such items are on the list to create a long-lasting learning experience that will stick with the user for the rest of their lives.

Although Ubien will not be the first to market as an AI assisted learning platform, It will be the first platform where users do not upload their course material to summarize or quiz themselves on, but rather to learn more deeply about the topic they are interested in. The current solutions to AI assisted learning are all based on summarization, and compacting information to allow users to easily digest the most amount of information in the shortest amount of time. This is far from the goal of Ubien, which is to make sure the topic at hand is fully understood to its last details and connections with the underlying topics are made.

## 3. Proposed System

Ubien proposes an AI assisted inquiry based approach to learning which aims to teach the underlying principles about why and how phenomena are the way they are. This ensures that the users are left with life-long lessons that they can remember and make connections with more easily. The system will be comprised of 3 main elements that will separately handle the inquiry based learning course creation and the responses to the user, an animated instructor embodiment that will provide expressive interaction and speech output and a user interaction layer that is mainly made up of the frontend website elements that are used to interact with the service.

### 3.1. Overview

Ubien is a next-generation digital education platform designed to transform the passive retention of information into active, enduring understanding. Born from the recognition that traditional educational paradigms often lead to temporary memorization rather than deep comprehension, Ubien leverages the principles of Inquiry-Based Learning (IBL) to shift the student's role from a passive recipient to an active discoverer of knowledge.

The core innovation of Ubien lies in its ability to convert static IBL textbooks into dynamic, interactive courses using Large Language Models (LLMs). Instead of presenting pre-packaged solutions, the system guides students through a Socratic method of questioning, allowing them to construct mental models and derive concepts from first principles organically.

The system allows users to upload standard IBL textbooks, which are then automatically converted into interactive, question-driven courses. To heighten engagement and simulate the experience of learning from a real tutor, Ubien uses 3D virtual avatars capable of facial expressions, gestures, and speech synthesis.

The project deliverables include a fully operational web application, backend services for content processing, and the virtual instructor model. By merging the depth of inquiry-based learning with the capabilities of generative AI, Ubien aims to provide a tool that supports not only academic success but the development of independent, rigorous, and enduring understanding.

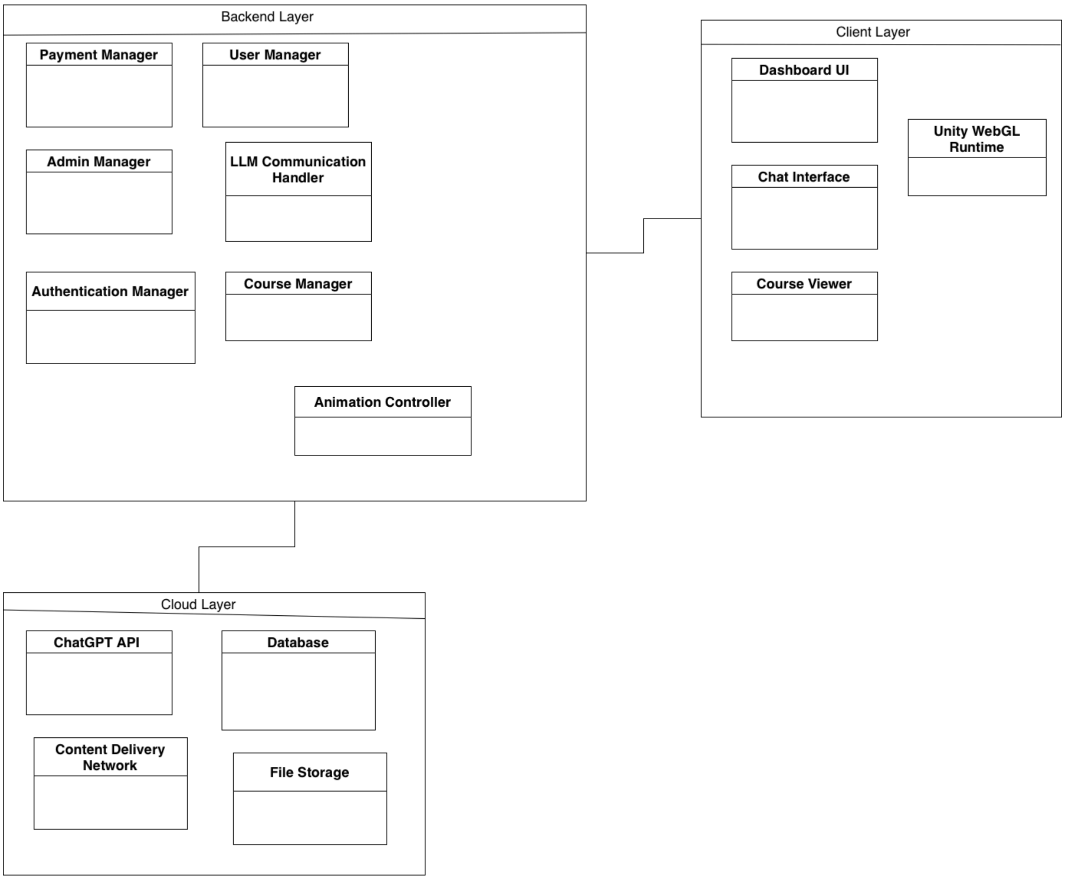


Figure 1. Proposed High Level System Architecture

## 3.2. Functional Requirements

### 3.2.1 User Functionalities

- The user should be able to create an account by entering the required information, including an email address, password, age, and school level.
- The user should be able to activate their account using a verification link or code sent to them via email.
- The user should be able to log in only after verifying their email address.
- The user should be able to upload an IBL textbook in PDF format.
- The user should be able to view the computed price before creating the course.
- The user should be able to confirm or cancel the creation of a course after seeing the price.
- The user should be able to view the list of courses they have created.
- The user should be able to access any course they have created.
- The user should be able to interact with the course material by asking questions, replying to the instructor, and seeking clarifications through the system’s interface.
- The user should be able to submit up to 600 GPT-tokens (approximately 400–500 words) of English text per message.
- The user should be able to select between Mode A, Mode B, and Mode C as instructor types.

#### 3.2.2 System Functionalities

- The system should send a confirmation email containing a verification link or code.
- The system should extract and process text from uploaded PDFs.
- The system should compute estimated token counts.
- The system should compute the estimated course creation cost.
- The system should generate course content using the ChatGPT API upon user approval.
- The system should store all generated course content (sections, questions, metadata).
- The system should track user progress (completed tasks and questions).
- The system should store token estimates and cost calculations.
- The system should generate responses using the ChatGPT API.
- When Study Mode A is selected, the system should display text-only responses.
- When Study Mode B is selected, the system should generate a neutral upper-body avatar.
- When Study Mode C is selected, the system should generate animation data including gaze direction, lip-sync motions, and an emotion tag (e.g., “happy,” “serious”).
- The system should synchronize avatar mouth movement with TTS audio.
- The system should use TTS to generate audio output.
- The system should generate animation data from LLM output (timing, gestures, expressions).
- The system should ensure TTS–animation sync delay ≤150 ms.
- The system should detect invalid/unreadable PDFs and show an error.
- The system must prevent course creation when the PDF is invalid.
- The system should alert the user if the PDF exceeds the maximum token limit.
- The system should alert the user if ChatGPT or TTS API failures occur.

## 3.3. Non-functional Requirements

### 2.2.1. Usability

- The system should allow a first-time user with no technical background to upload a PDF and start a course within 2 minutes of initial use.
- All core actions, for instance, uploading textbooks, accepting the cost, selecting study modes, and asking questions, should be accessible in no more than four clicks from the dashboard.

### 2.2.2. Reliability

- In the event of network failures, API timeouts, or invalid uploads, the system displays an error message within 10 seconds and allows users to resubmit the request with descriptive error messages.
- The backend system will maintain an availability of at least 99% during regular operational hours.
- User progress, including completed tasks, questions asked, and lessons viewed, will be saved instantly and remain secure, even if the browser is refreshed or closed.

#### 2.2.3. Performance

- Under typical network conditions, which include a connection with 50–100 ms latency and at least 10 Mbps download bandwidth, only text ChatGPT API responses will be delivered within 3-5 seconds.
- Once the user approves the cost, the system will initiate the course generation process within one minute.
- Text extraction and token estimation for PDFs up to 50 MB should take 10 seconds or less on the backend.
- The system will perform text summarization, content chunking, and filtering to avoid exceeding the ChatGPT API token limit or context buffer size.
- WebGL-based animations will maintain a minimum of 25 frames per second on mid-range laptops with 8 GB of RAM and an integrated GPU.

#### 2.2.4. Supportability

- Each of the system's distinct modules, PDF parsing, token estimation, LLM processing, and animation control, can be upgraded or changed without impacting TTS, and animate the system as a whole.
- The system should record failed API requests, PDF extraction errors, and token estimation failures, along with the timestamps and request details. To facilitate maintenance and troubleshooting, logs must be retained for a minimum of 30 days.
- API keys, price values, and animation choices should all be customizable using environment variables rather than being hard-coded. This will eliminate the need for recompilation of the backend.

#### 2.2.5. Scalability

- The backend should be able to run multiple instances simultaneously and support 200 active users without any loss of performance.
- To reduce response time by at least 50% for repeated requests, frequently reused LLM outputs, such as textbook summaries and chapter metadata, will be cached.
- The system should load the list of courses within two seconds when a user opens the dashboard, even if the database contains a large number of courses, with a total of 10,000 or more courses stored.

### 3.4. Pseudo Requirements

- The application shall be developed as a web application.
- GitHub shall be used for code management throughout the project.
- The Go programming language shall be used to implement system processes and business logic.
- PostgreSQL database management system shall be utilized for user accounts and system data storage.
- Unity’s WebGL shall be used for visualization tasks.
- The OpenAI’s ChatGPT artificial intelligence model shall be integrated as a core part of the system's functionality.
- The project shall be completed in May 2026

## 3.5. System Models

### 3.5.1. Scenarios

#### 3.5.1.1 User Registration

**Use Case Name:** Register User

**Actors:** User

**Entry conditions:**
The user is not logged in.

**Event flow:**

1. The user initiates the registration process.
2. The user provides the required registration information.
3. The system creates a user account in an unverified state.
4. The system provides a verification mechanism.
5. The user completes the verification step.
6. The system marks the account as verified.

**Alternate event flows:**
2.1 The provided information is invalid or incomplete and the system rejects the registration.
5.1 Verification fails or expires and the system allows retrying.

**Exit conditions:**
The user account is verified or the registration process is aborted.

#### 3.5.1.2 User Login

**Use Case Name:** Log In

**Actors:** User

**Entry conditions:**
The user has a verified account.

**Event flow:**

1. The user initiates the login process.
2. The user provides the required credentials.
3. The system validates the credentials.
4. The system establishes a user session.
5. The system grants access to the dashboard.

**Alternate event flows:**
3.1 Credentials are invalid and the system rejects the login.
3.2 The account is not verified and the system blocks login.

**Exit conditions:**
The user is logged in or the login attempt fails.

#### 3.5.1.3 Subscription

**Use Case Name:** Subscribe to Ubien

**Actors:** User, Payment Processor

**Entry conditions:**
The user is logged in.

**Event flow:**

1. The user initiates the subscription process.
2. The system presents available subscription options.
3. The user selects a subscription option.
4. The system processes payment through the payment processor.
5. The system activates the subscription.
6. The system initializes the monthly course creation quota.

**Alternate event flows:**
4.1 Payment fails and the subscription is not activated.
4.2 The user cancels the process and no subscription is created.

**Exit conditions:**
The subscription is active or the subscription process ends without activation.

#### 3.5.1.4 Upload IBL Textbook and Preview Course Generation

**Use Case Name:** Upload IBL Textbook

**Actors:** User

**Entry conditions:**
The user is logged in.

**Event flow:**

1. The user initiates the upload process for an IBL textbook.
2. The user provides a textbook file in the supported format.
3. The system stores the uploaded file.
4. The system extracts content from the uploaded textbook.
5. The system estimates the resources required to generate a course from the textbook.
6. The system displays the estimate to the user.

**Alternate event flows:**
2.1 The file format is unsupported and the system rejects the upload.
4.1 The textbook cannot be processed and the system displays an error.
5.1 The estimate exceeds system limits and the system warns the user.

**Exit conditions:**
An estimate is displayed or the upload process is aborted.

#### 3.5.1.5 Create Course from IBL Textbook

**Use Case Name:** Create Course

**Actors:** User, LLM Service (ChatGPT API)

**Entry conditions:**
The user is logged in.
The user has an active subscription.
The user has remaining monthly course creation quota.
A textbook has been uploaded successfully.

**Event flow:**

1. The user confirms course creation for the uploaded textbook.
2. The system validates that the user's subscription is active.
3. The system validates that the user has remaining monthly course creation quota.
4. The system splits the textbook into chapters.
5. The system splits chapters into sections.
6. The system generates expository slides for each section using the LLM service.
7. The system extracts IBL-type questions for each section using the LLM service.
8. The system stores the generated chapters, sections, slides, and questions persistently.
9. The system creates a new course and makes it visible on the user's dashboard.
10. The system decrements the user's monthly course creation quota.

**Alternate event flows:**
2.1 Subscription is inactive and the system blocks course creation.
3.1 Monthly quota is exhausted and the system blocks course creation.
6–7.1 Generation fails and the course is not created and the quota is not decremented.

**Exit conditions:**
The course is created and stored or course creation is blocked or fails.

#### 3.5.1.6 Access Course Content

**Use Case Name:** Access Course Content

**Actors:** User

**Entry conditions:**
The user is logged in.
A course exists for the user.

**Event flow:**

1. The user selects a course.
2. The system displays the course structure consisting of chapters and sections
3. The user selects a chapter and a section.
4. The system displays the section content.

**Alternate event flows:**
None.

**Exit conditions:**
The user is viewing the selected course content or exits the course.

**3.5.1.7 Navigate Slides within a Section**
**Use Case Name:** Navigate Slides

**Actors:** User

**Entry conditions:**
The user is viewing a section that contains slides.
The user has selected an interaction mode (text-based or animated instructor).

**Event flow:**

1. The system displays slide content according to the selected interaction mode.
2. If the text-based mode is selected, the system presents slide content in textual form.
3. If the animated instructor mode is selected, the system presents slide content through an animated instructor.
4. The user navigates forward and backward between slides.
5. The system records the user's last visited slide for progress tracking.

**Alternate event flows:**
None.

**Exit conditions:**
The user continues navigation or leaves the section.

**3.5.1.8 Open Question**
**Use Case Name:** Open Question

**Actors:** User

**Entry conditions:**
The user is viewing a section that contains questions.

**Event flow:**

1. The user selects a question.
2. The system loads the question prompt.
3. The system loads the persistent chat history for the question.
4. If no interaction mode has been selected for the question, the system prompts the user to select an interaction mode (text-based or animated instructor).
5. The user selects an interaction mode.
6. The system stores the selected interaction mode for the question.
7. The system displays the question according to the selected interaction mode.

**Alternate event flows:**
None.

**Exit conditions:**
The question is opened with an interaction mode selected, or the user returns to the section content.

#### 3.5.1.9 Question Q&A

**Use Case Name:** Question Q&A

**Actors:** User, LLM Service (ChatGPT API)

**Entry conditions:**
A question is open.
The user has an active subscription.
The user has selected an interaction mode (text-based or animated instructor).

**Event flow:**

1. The user submits an attempt or a request for help according to the selected interaction mode.
2. The system appends the user input to the question's chat history.
3. The system sends the question context and chat history to the LLM service.
4. The system receives a tutor response from the LLM service.
5. The system presents the tutor response according to the selected interaction mode.
6. Steps 1–5 may repeat as the user continues interacting within the question.
7. The system evaluates whether the student has arrived at the correct answer.
8. If the answer is reached, the system marks the question as solved.

**Alternate event flows:**
The subscription is inactive and the system blocks message submission.
The answer is not reached and the question remains unsolved.
The LLM service fails and the system reports an error without losing existing chat history.
The animated instructor mode is unavailable and the system falls back to text-based mode.

**Exit conditions:**
The question remains in progress or is marked as solved.

#### 3.5.1.10 View Progress

**Use Case Name:** View Progress

**Actors:** User

**Entry conditions:**
The user is logged in.
A course exists for the user.

**Event flow:**

1. The user opens the progress view for a course.
2. The system displays progress information, including the last visited slide and question statuses.
3. The user reviews the progress information.

**Alternate event flows:**
2.1 No progress exists and the system displays an empty progress view.

**Exit conditions:**
Progress information is displayed or the user leaves the progress view.

### 3.5.1.11 Flag Slide or Section for Reconstruction
**Use Case Name: Flag Content for Reconstruction**

**Actors:** User, LLM Service (ChatGPT API)

**Entry conditions:**
The user is viewing a slide or section.

**Event flow:**

1. The user flags the current slide or section as misclassified.
2. The user optionally provides an explanation.
3. The system records the reconstruction request.
4. The system regenerates the affected content using the LLM service.
5. The system updates the stored content with the regenerated version.

**Alternate event flows:**
4.1 Regeneration fails and the system retains the original content and reports an error.

**Exit conditions:**
The content is updated or the reconstruction process fails.

### 3.5.1.12 Monthly Course Creation Quota Reset
**Use Case Name:** Reset Monthly Course Quota

**Actors:** System

**Entry conditions:**
The quota reset time is reached for an active subscription.

**Event flow:**

1. The system resets the user’s monthly course creation quota according to the active subscription plan.
2. The system records the reset event.

**Alternate event flows:**
None.

**Exit conditions:**
The monthly course creation quota is available again.

### 3.5.2. Use-Case Model

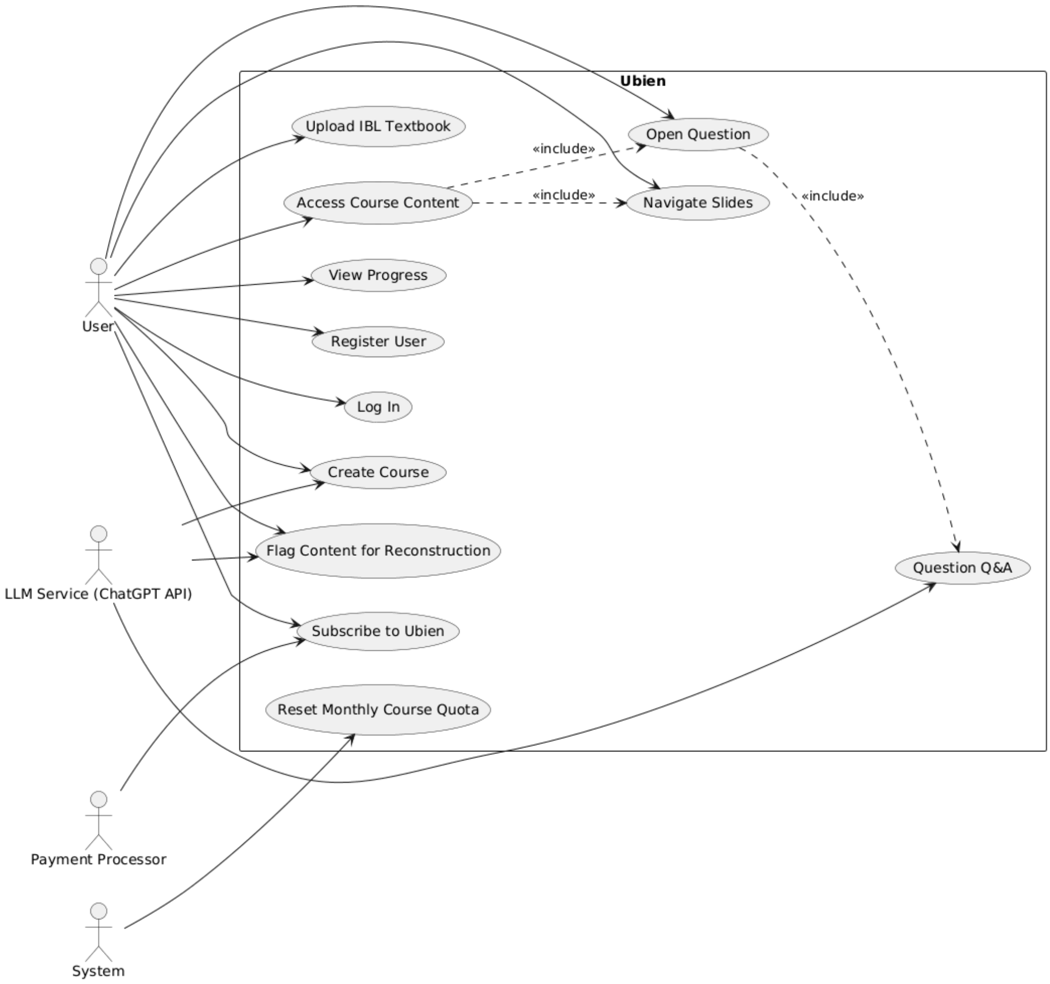


Figure 2. Use Case Model

### 3.5.3. Object and Class Model

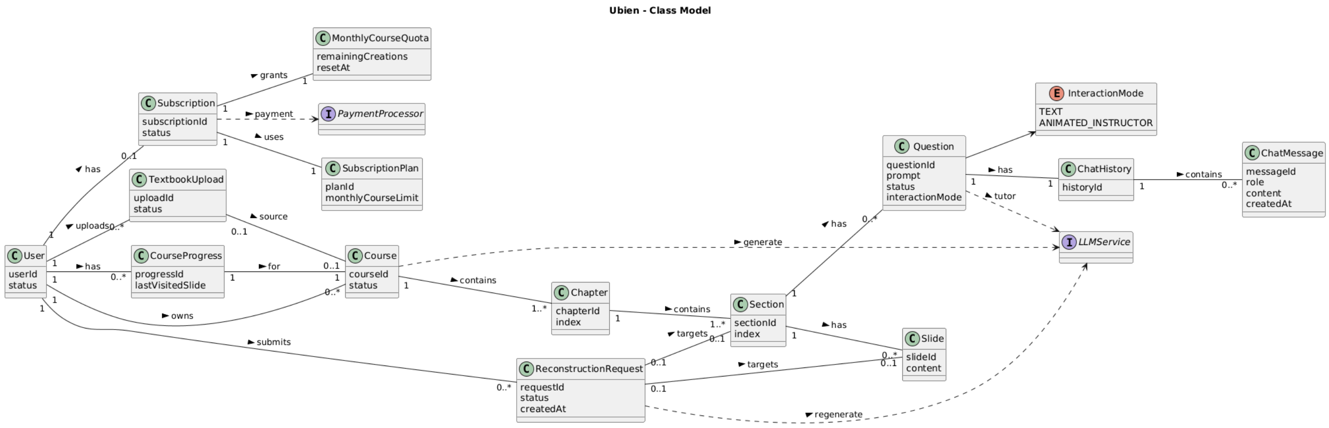


Figure 3. Class Model

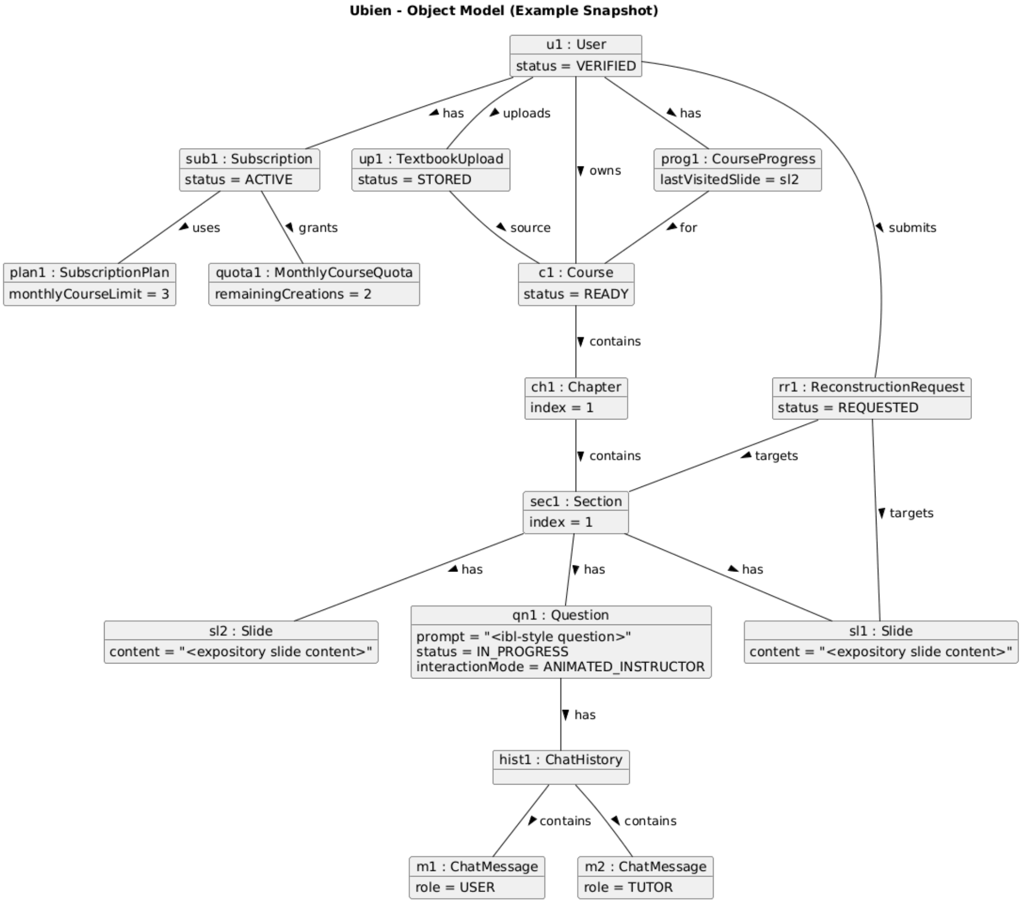


Figure 4. Object Model

### 3.5.4. Dynamic Models

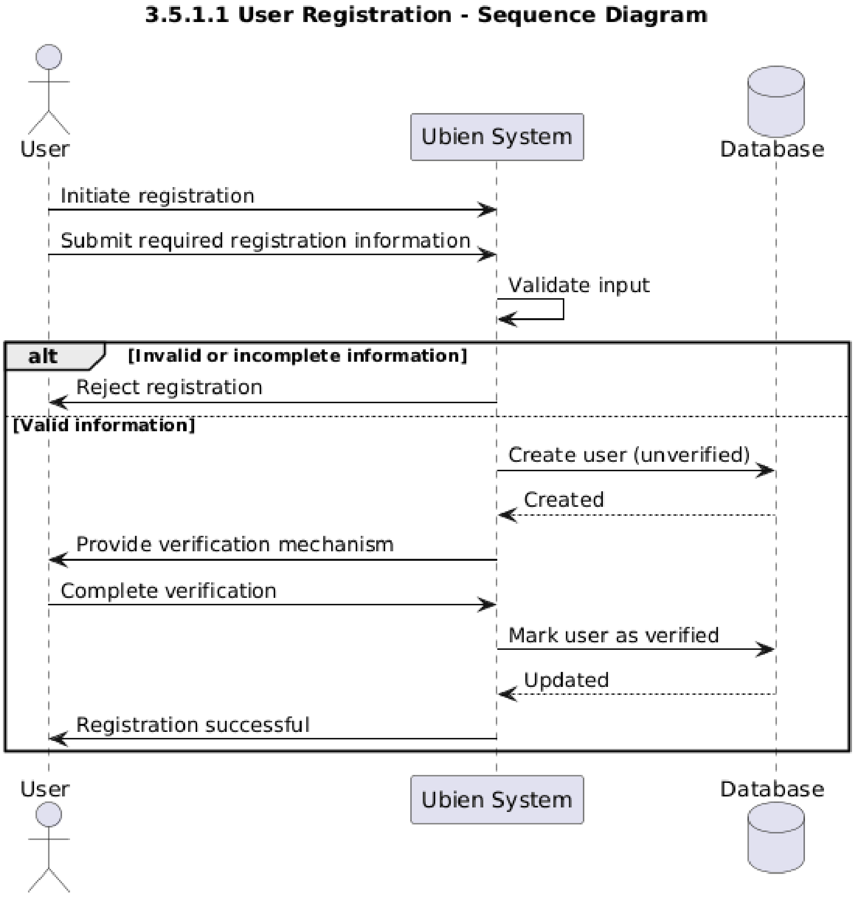


Figure 5. User Registration Sequence Diagram

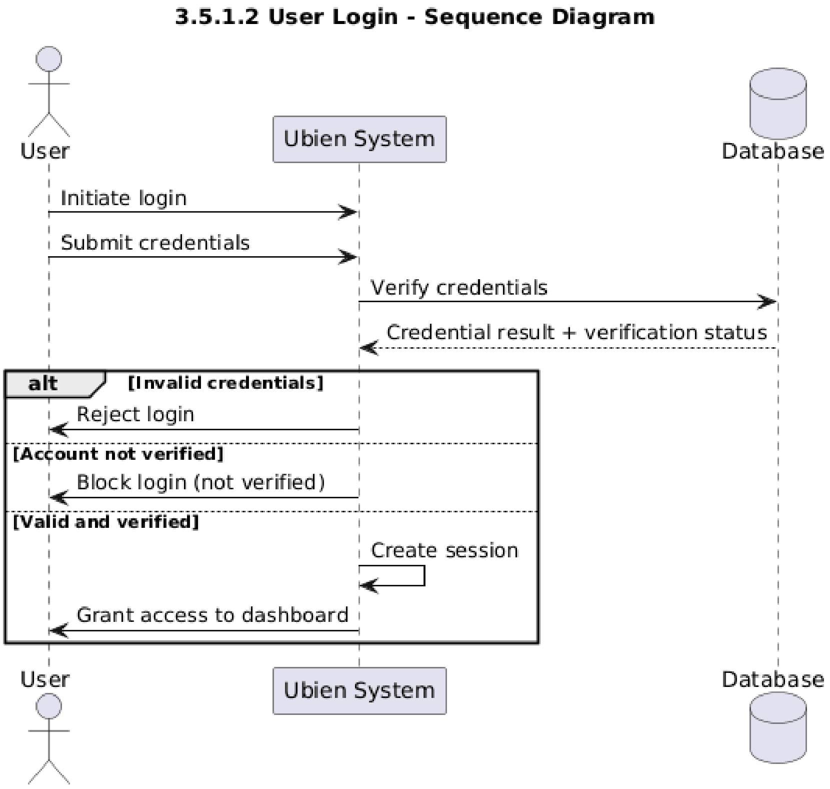


Figure 6. User Login Sequence Diagram

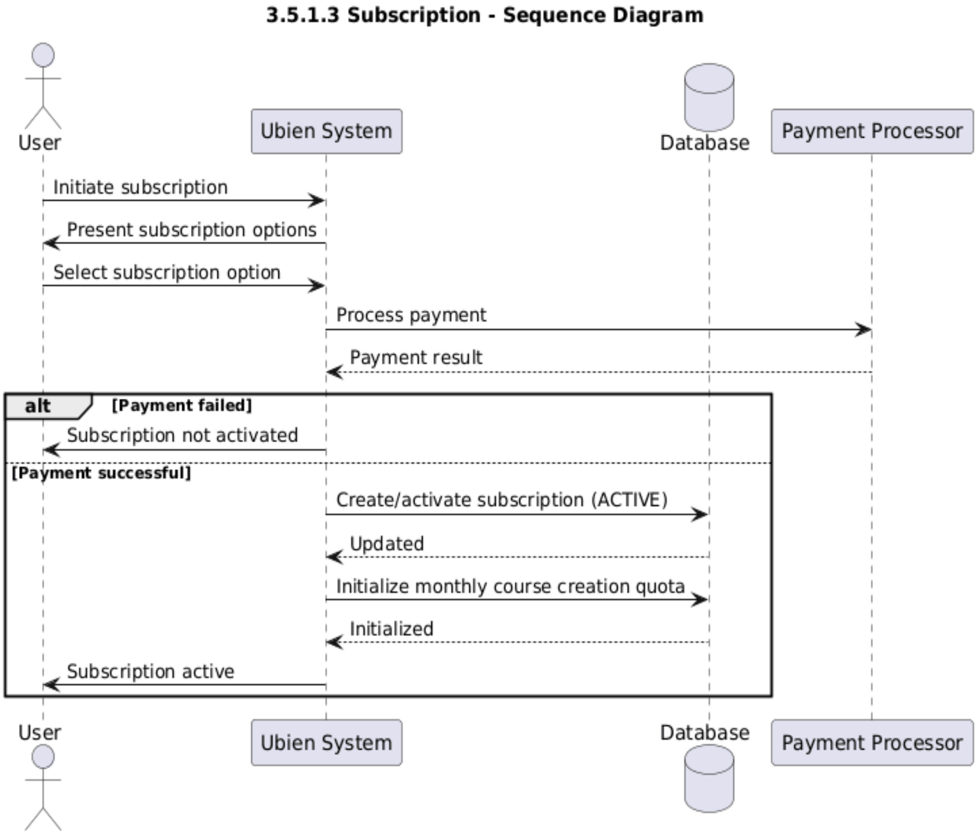


Figure 7. Subscription Sequence Diagram

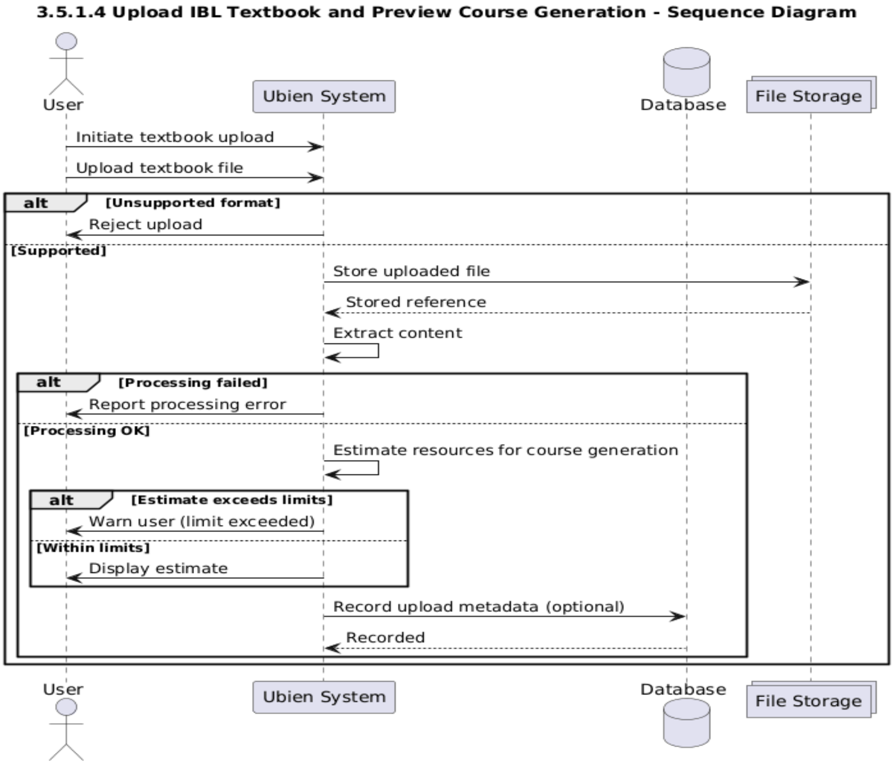


Figure 8. Course Generation Sequence Diagram

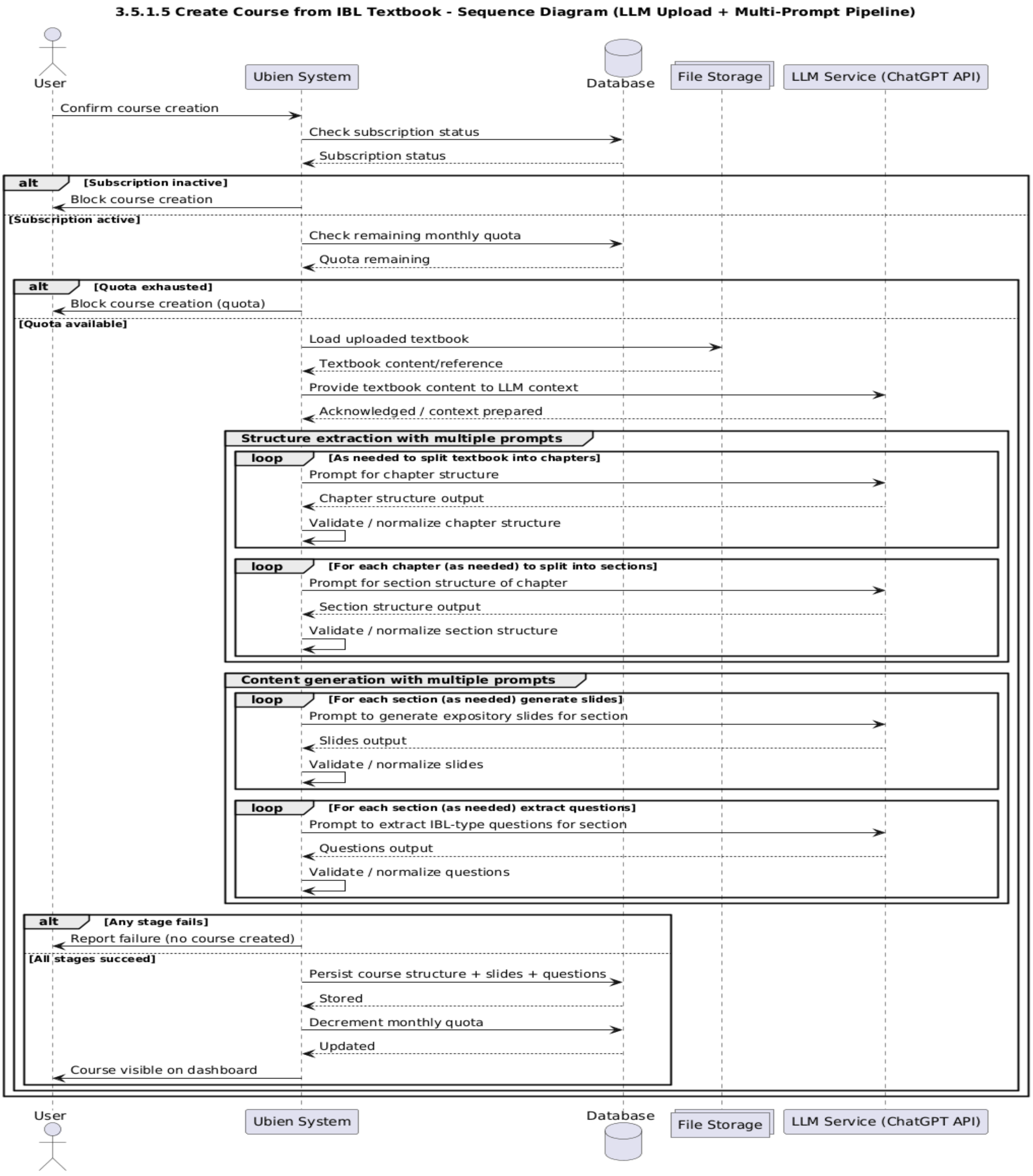

3.5.1.5 Create Course from IBL Textbook - Sequence Diagram (LLM Upload + Multi-Prompt Pipeline)
User
Ubien System
Database
File Storage
LLM Service (ChatGPT API)
Confirm course creation
Check subscription status
Subscription status
alt
[Subscription inactive]
Block course creation
[Subscription active]
Check remaining monthly quota
Quota remaining
alt
[Quota exhausted]
Block course creation (quota)
[Quota available]
Load uploaded textbook
Textbook content/reference
Provide textbook content to LLM context
Acknowledged / context prepared
Structure extraction with multiple prompts
loop
[As needed to split textbook into chapters]
Prompt for chapter structure
Chapter structure output
Validate / normalize chapter structure
loop
[For each chapter (as needed) to split into sections]
Prompt for section structure of chapter
Section structure output
Validate / normalize section structure
Content generation with multiple prompts
loop
[For each section (as needed) generate slides]
Prompt to generate expository slides for section
Slides output
Validate / normalize slides
loop
[For each section (as needed) extract questions]
Prompt to extract IBL-type questions for section
Questions output
Validate / normalize questions
alt
[Any stage fails]
Report failure (no course created)
[All stages succeed]
Persist course structure + slides + questions
Stored
Decrement monthly quota
Updated
Course visible on dashboard
User
Ubien System
Database
File Storage
LLM Service (ChatGPT API)

Figure 9. Course Creation Sequence Diagram

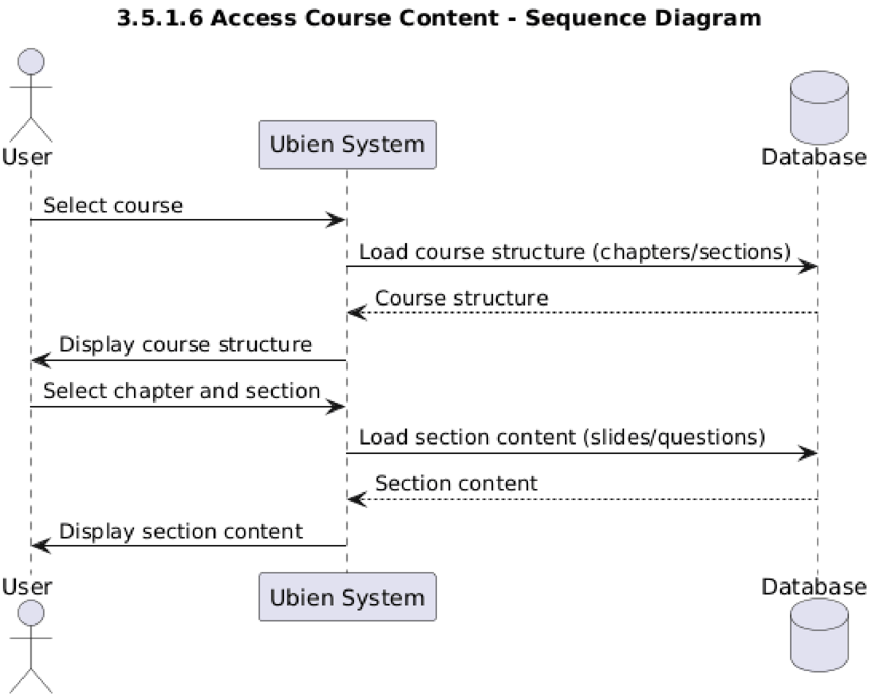

3.5.1.6 Access Course Content - Sequence Diagram
User
Ubien System
Database
Select course
Load course structure (chapters/sections)
Course structure
Display course structure
Select chapter and section
Load section content (slides/questions)
Section content
Display section content
User
Ubien System
Database

Figure 10. Course Content Access Sequence Diagram

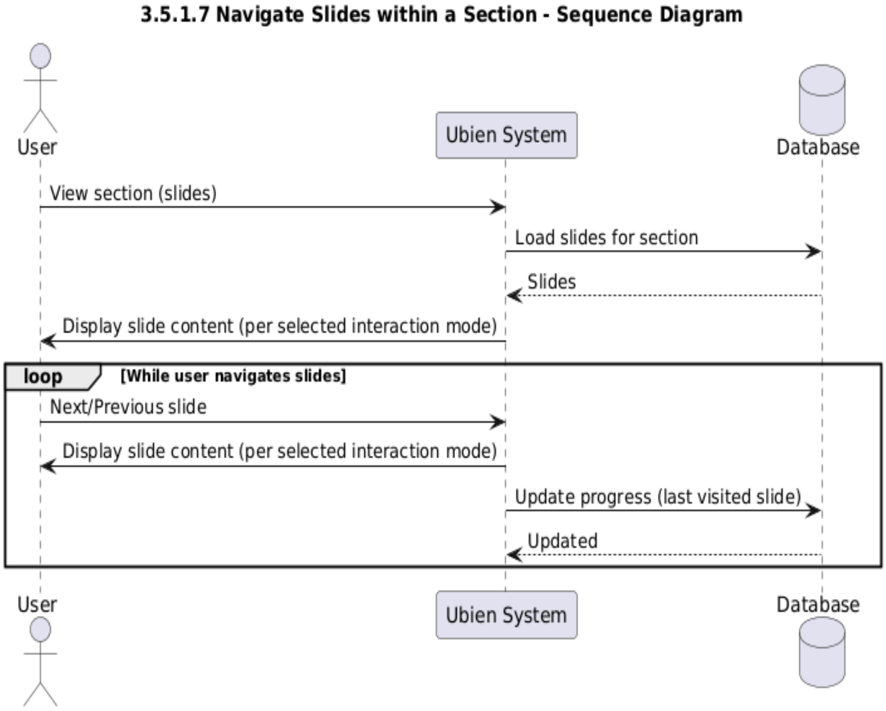

3.5.1.7 Navigate Slides within a Section - Sequence Diagram
User
Ubien System
Database
View section (slides)
Load slides for section
Slides
Display slide content (per selected interaction mode)
loop
[While user navigates slides]
Next/Previous slide
Display slide content (per selected interaction mode)
Update progress (last visited slide)
Updated
User
Ubien System
Database

Figure 11. Navigation Sequence Diagram

### 3.5.1.8 Open Question - Sequence Diagram

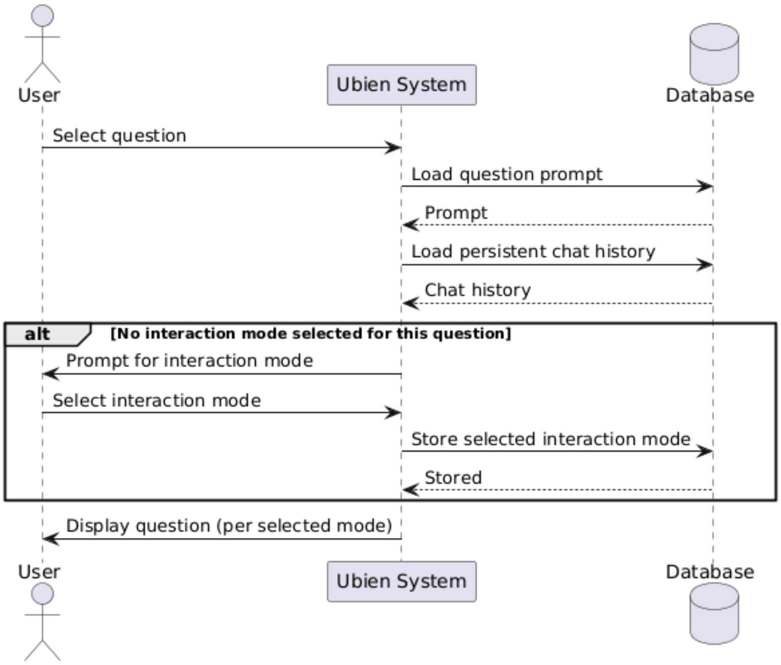

User
Ubien System
Database
Select question
Load question prompt
Prompt
Load persistent chat history
Chat history
alt
[No interaction mode selected for this question]
Prompt for interaction mode
Select interaction mode
Store selected interaction mode
Stored
Display question (per selected mode)
User
Ubien System
Database

Figure 12. Open Question Sequence Diagram

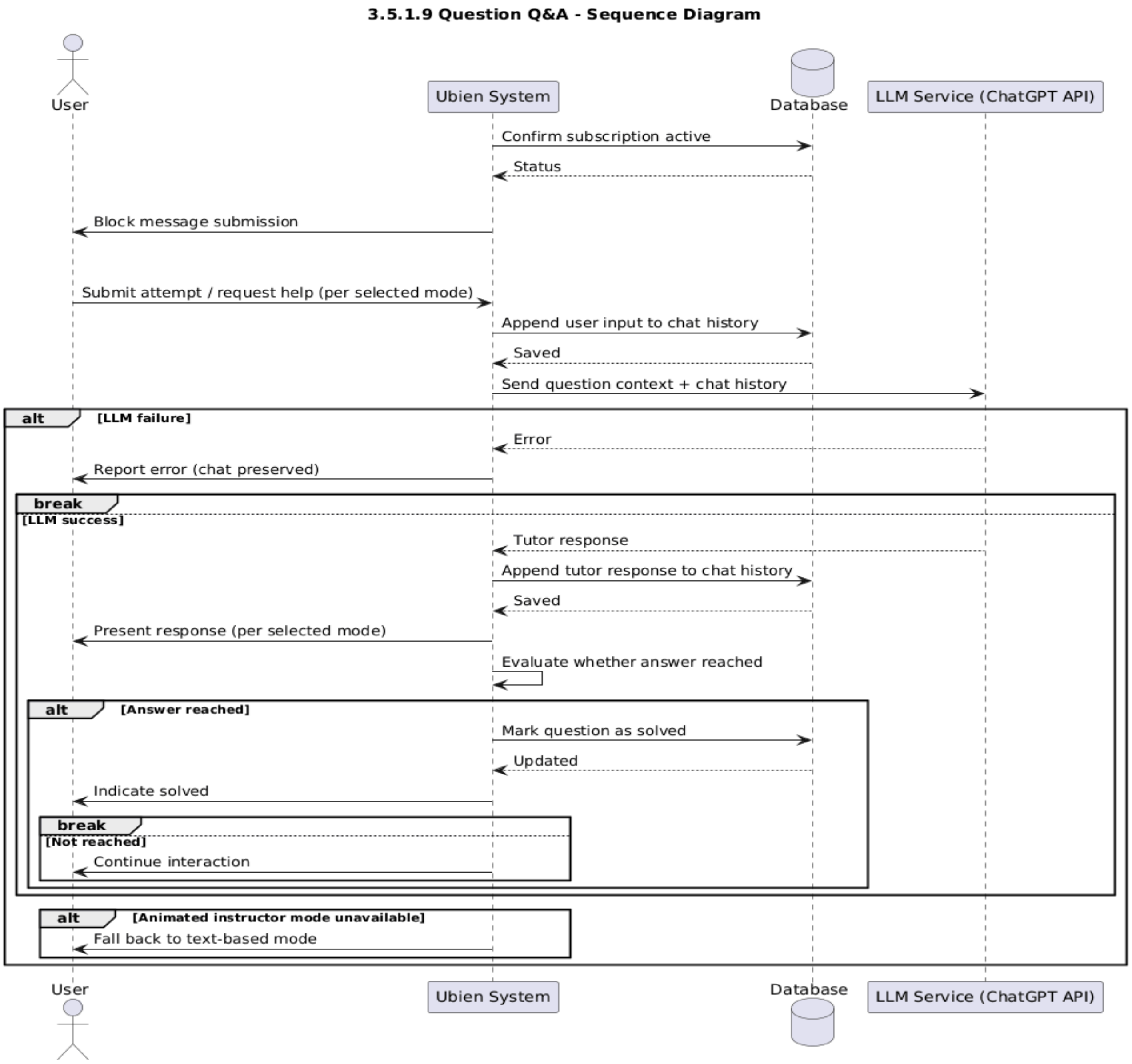

3.5.1.9 Question Q&A - Sequence Diagram
User
Ubien System
Database
LLM Service (ChatGPT API)
Confirm subscription active
Status
Block message submission
Submit attempt / request help (per selected mode)
Append user input to chat history
Saved
Send question context + chat history
alt
[LLM failure]
Error
Report error (chat preserved)
break
[LLM success]
Tutor response
Append tutor response to chat history
Saved
Present response (per selected mode)
Evaluate whether answer reached
alt
[Answer reached]
Mark question as solved
Updated
Indicate solved
break
[Not reached]
Continue interaction
alt
[Animated instructor mode unavailable]
Fall back to text-based mode
User
Ubien System
Database
LLM Service (ChatGPT API)

Figure 13. Q&A Sequence Diagram

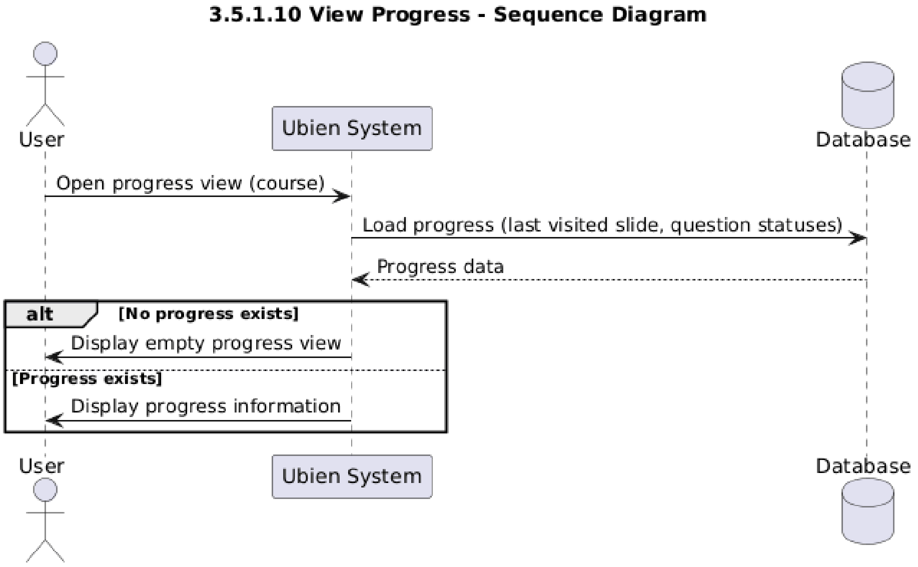

3.5.1.10 View Progress - Sequence Diagram
User
Ubien System
Database
Open progress view (course)
Load progress (last visited slide, question statuses)
Progress data
alt
[No progress exists]
Display empty progress view
[Progress exists]
Display progress information
User
Ubien System
Database

Figure 14. View Progress Sequence Diagram

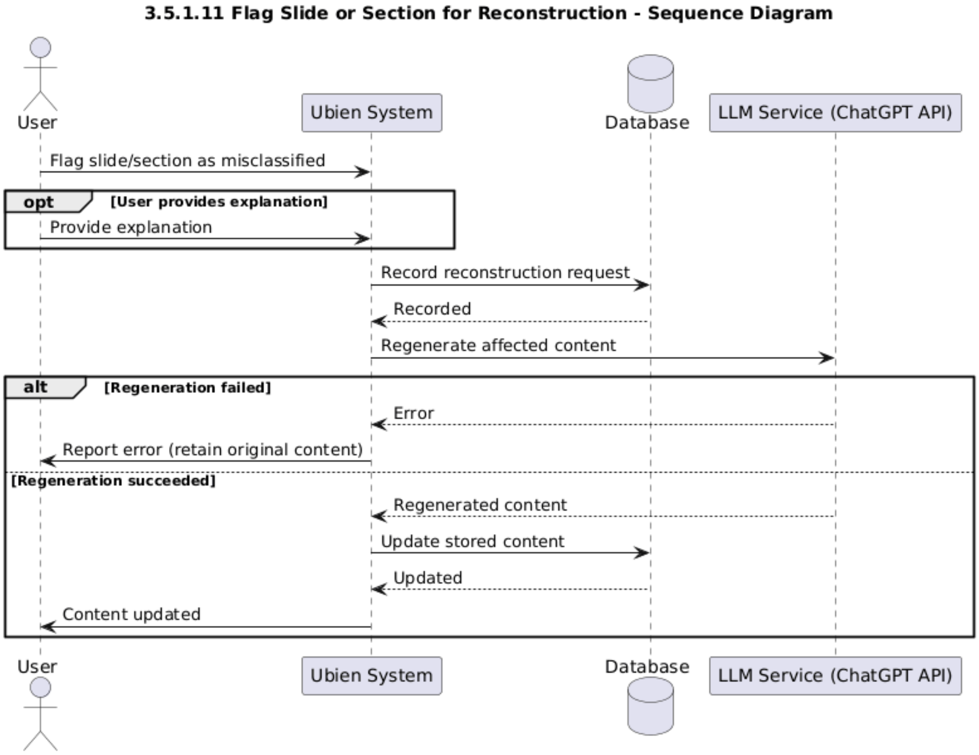

3.5.1.11 Flag Slide or Section for Reconstruction - Sequence Diagram
User
Ubien System
Database
LLM Service (ChatGPT API)
Flag slide/section as misclassified
opt
[User provides explanation]
Provide explanation
Record reconstruction request
Recorded
Regenerate affected content
alt
[Regeneration failed]
Error
Report error (retain original content)
[Regeneration succeeded]
Regenerated content
Update stored content
Updated
Content updated
User
Ubien System
Database
LLM Service (ChatGPT API)

Figure 15. Reconstruction Sequence Diagram

### 3.5.1.12 Monthly Course Creation Quota Reset - Sequence Diagram

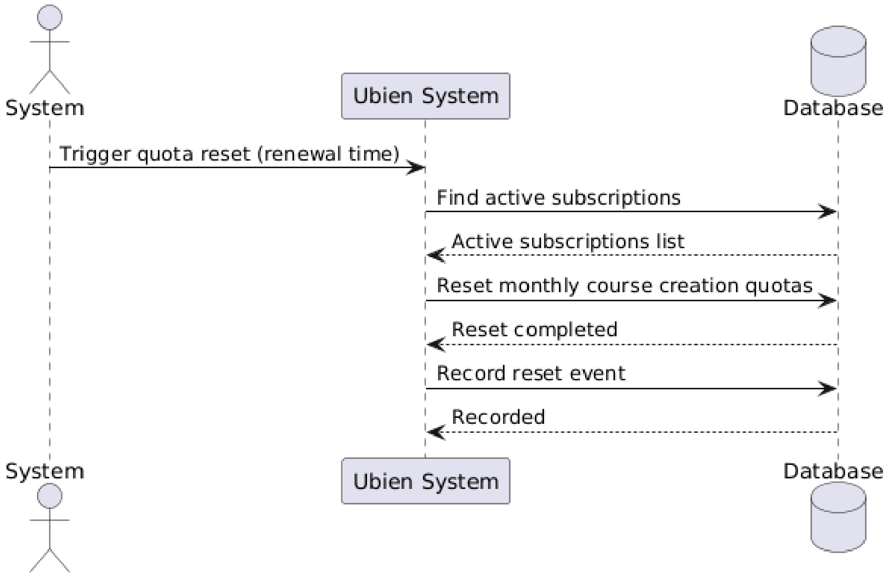

System
Ubien System
Database
Trigger quota reset (renewal time)
Find active subscriptions
Active subscriptions list
Reset monthly course creation quotas
Reset completed
Record reset event
Recorded
System
Ubien System
Database

Figure 16. Quota Reset Sequence Diagram

### 3.5.5. User Interface

#### 3.5.5.1 Navigational Path

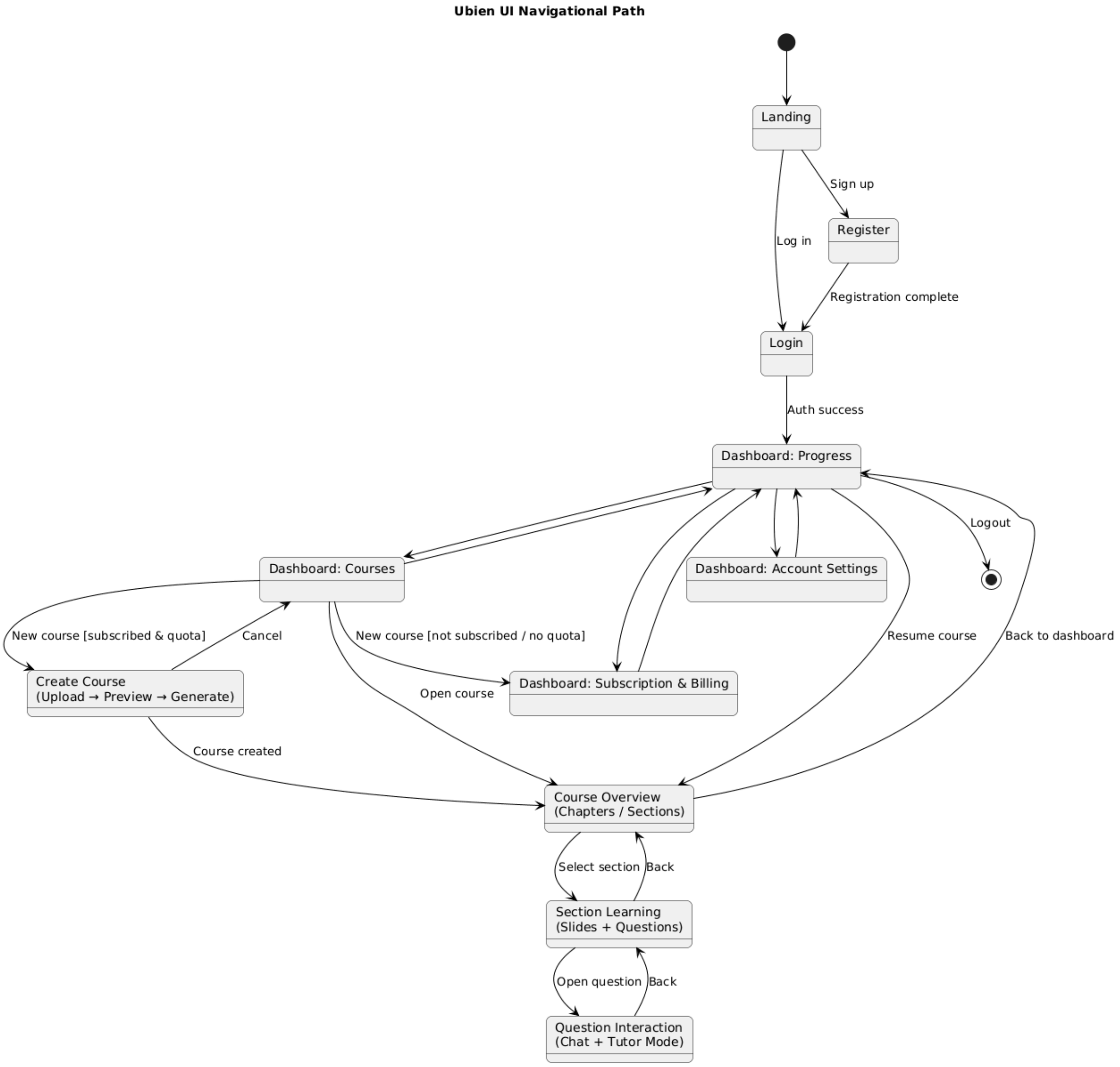
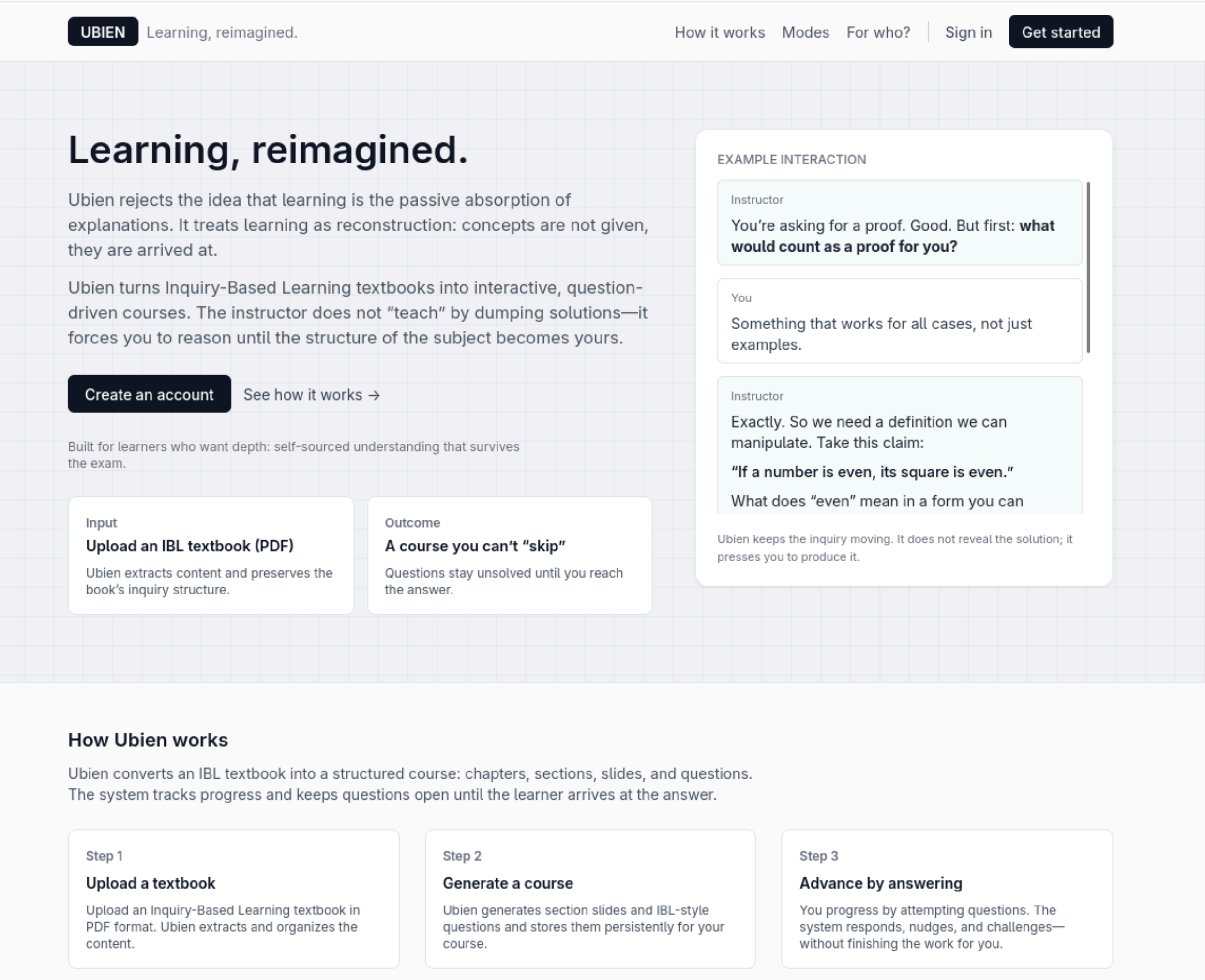


Figure 17. UI Navigational Path

### 3.5.5.2 UI Mock-Ups

#### 3.5.5.2.1 Landing Page



## Learning, reimagined.

Ubien rejects the idea that learning is the passive absorption of explanations. It treats learning as reconstruction: concepts are not given, they are arrived at.

Ubien turns Inquiry-Based Learning textbooks into interactive, question-driven courses. The instructor does not “teach” by dumping solutions—it forces you to reason until the structure of the subject becomes yours.

Built for learners who want depth: self-sourced understanding that survives the exam.

Input

**Upload an IBL textbook (PDF)**

Ubien extracts content and preserves the book’s inquiry structure.

Outcome

**A course you can’t “skip”**

Questions stay unsolved until you reach the answer.

EXAMPLE INTERACTION

Instructor

You’re asking for a proof. Good. But first: **what would count as a proof for you?**

You

Something that works for all cases, not just examples.

Instructor

Exactly. So we need a definition we can manipulate. Take this claim:

**“If a number is even, its square is even.”**

What does “even” mean in a form you can

Ubien keeps the inquiry moving. It does not reveal the solution; it presses you to produce it.

**How Ubien works**

Ubien converts an IBL textbook into a structured course: chapters, sections, slides, and questions. The system tracks progress and keeps questions open until the learner arrives at the answer.

Step 1

**Upload a textbook**

Upload an Inquiry-Based Learning textbook in PDF format. Ubien extracts and organizes the content.

Step 2

**Generate a course**

Ubien generates section slides and IBL-style questions and stores them persistently for your course.

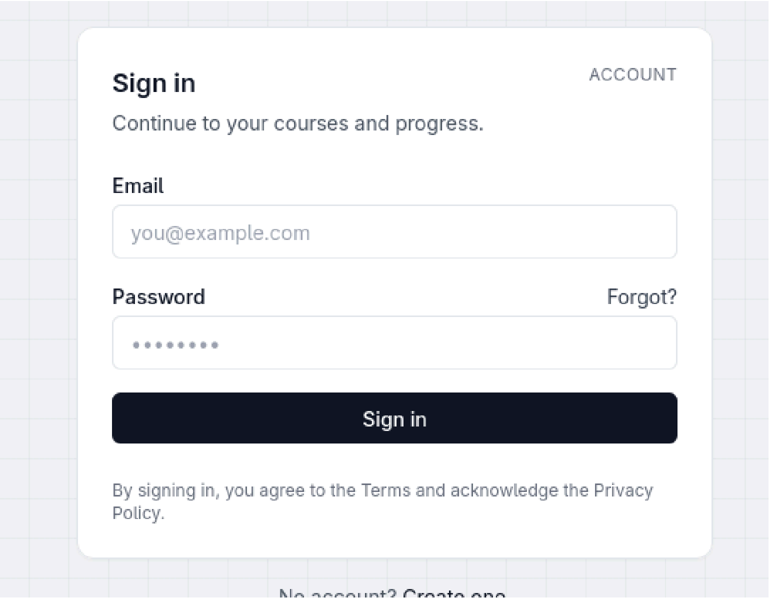
Step 3

**Advance by answering**

You progress by attempting questions. The system responds, nudges, and challenges—without finishing the work for you.

Figure 18. Main Page

### 3.5.5.2.2 Login & Signup Pages



Sign in

ACCOUNT

Continue to your courses and progress.

Email

you@example.com

Password

Forgot?

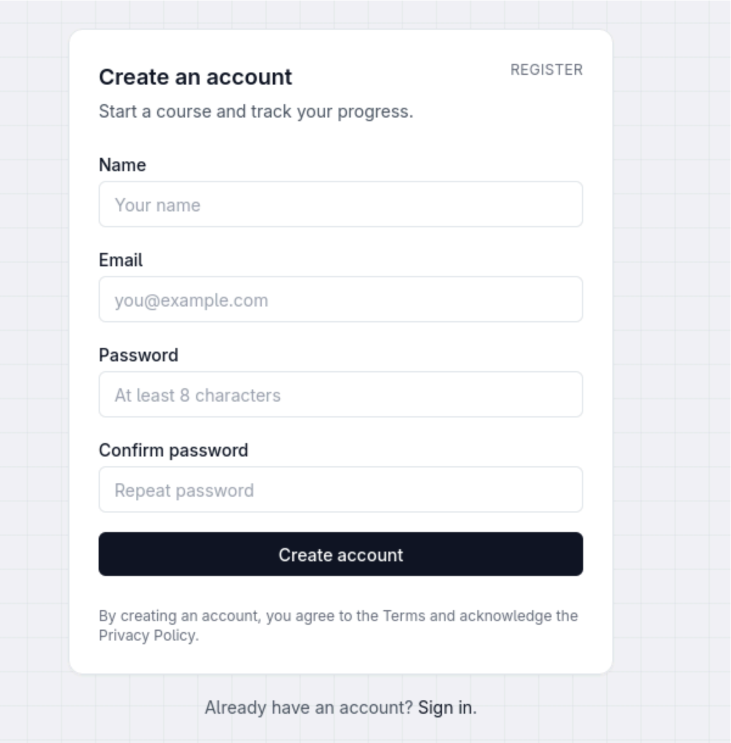
••••••••

Sign in

By signing in, you agree to the Terms and acknowledge the Privacy Policy.

No account? Create one

Figure 19. Login Page



Create an account

REGISTER

Start a course and track your progress.

Name

Your name

Email

you@example.com

Password

At least 8 characters

Confirm password

Repeat password

Create account

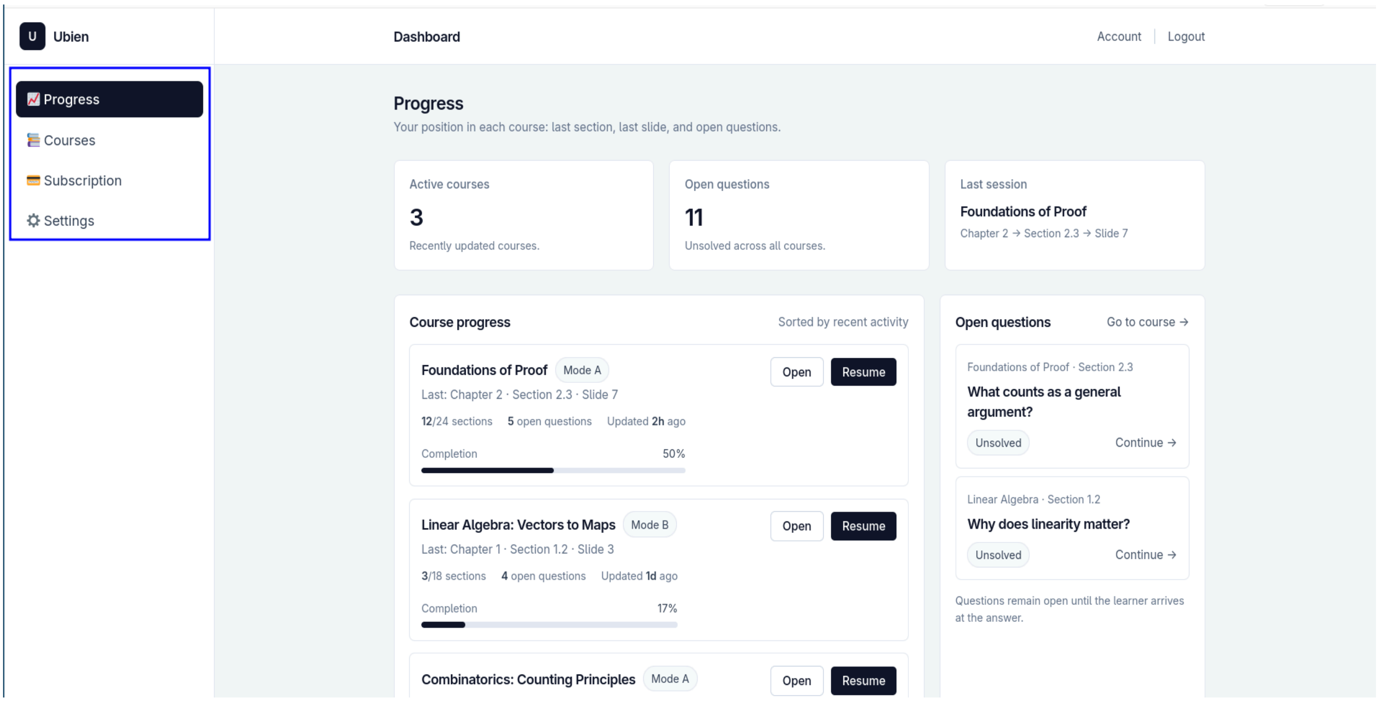
By creating an account, you agree to the Terms and acknowledge the Privacy Policy.

Already have an account? Sign in.

Figure 19. Sign Up Page

The login and signup pages provide a minimal entry point into Ubien, allowing users to authenticate or create an account with only the required information.

### 3.5.5.2.3 Dashboard



Progress

Your position in each course: last section, last slide, and open questions.

Active courses
3
Recently updated courses.

Open questions
11
Unsolved across all courses.

Last session
Foundations of Proof
Chapter 2 → Section 2.3 → Slide 7

Course progress — Sorted by recent activity

Foundations of Proof — Mode A
Last: Chapter 2 · Section 2.3 · Slide 7
12/24 sections  5 open questions  Updated 2h ago
Completion 50%

Linear Algebra: Vectors to Maps — Mode B
Last: Chapter 1 · Section 1.2 · Slide 3
3/18 sections  4 open questions  Updated 1d ago
Completion 17%

Combinatorics: Counting Principles — Mode A

Open questions

Foundations of Proof · Section 2.3
What counts as a general argument?
Unsolved

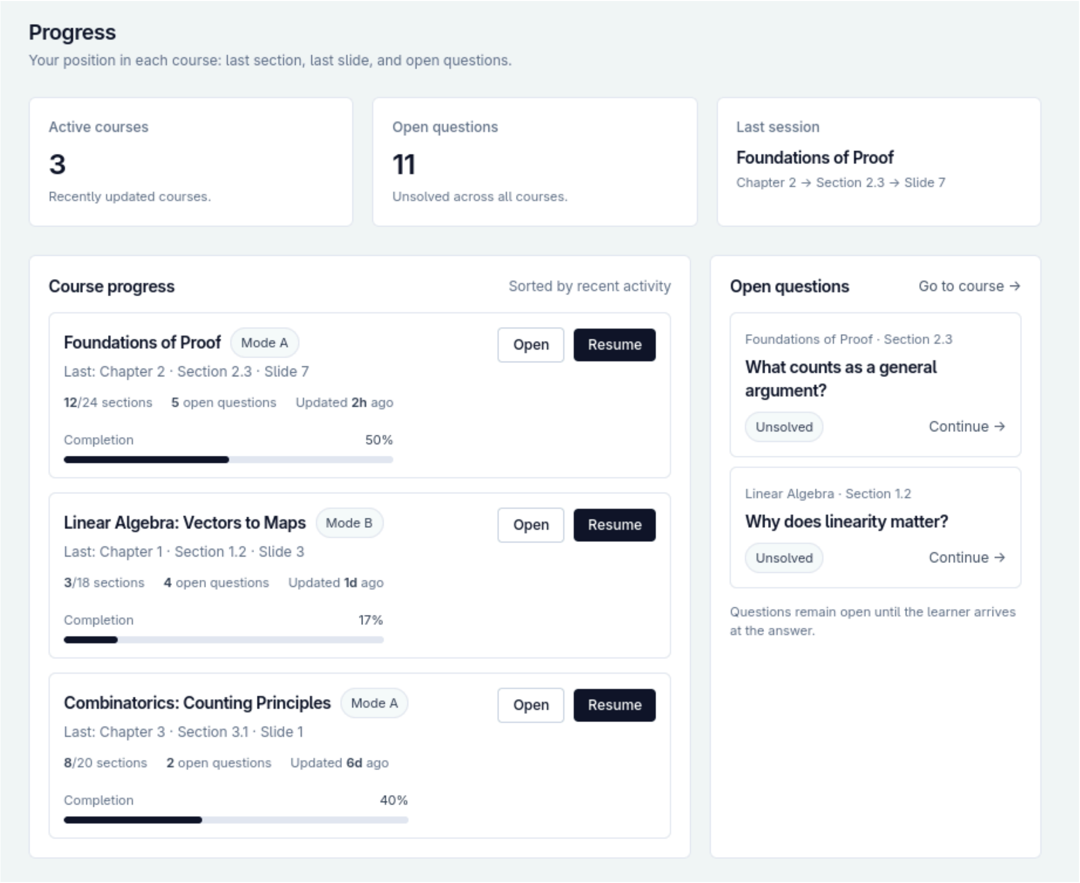
Linear Algebra · Section 1.2
Why does linearity matter?
Unsolved

Questions remain open until the learner arrives at the answer.

Figure 20. Dashboard

The dashboard serves as the central interface through which users manage their interaction with Ubien. It provides access to course progress, course creation, subscription status, and account settings, and acts as the primary navigation hub once a user is authenticated.

### 3.5.5.2.4 Dashboard - Progress



Progress

Your position in each course: last section, last slide, and open questions.

Active courses: 3. Recently updated courses.

Open questions: 11. Unsolved across all courses.

Last session: Foundations of Proof. Chapter 2 → Section 2.3 → Slide 7

Course progress (Sorted by recent activity)

Foundations of Proof (Mode A)
Last: Chapter 2 · Section 2.3 · Slide 7
12/24 sections · 5 open questions · Updated 2h ago
Completion 50%

Linear Algebra: Vectors to Maps (Mode B)
Last: Chapter 1 · Section 1.2 · Slide 3
3/18 sections · 4 open questions · Updated 1d ago
Completion 17%

Combinatorics: Counting Principles (Mode A)
Last: Chapter 3 · Section 3.1 · Slide 1
8/20 sections · 2 open questions · Updated 6d ago
Completion 40%

Open questions

Foundations of Proof · Section 2.3
What counts as a general argument?
Unsolved

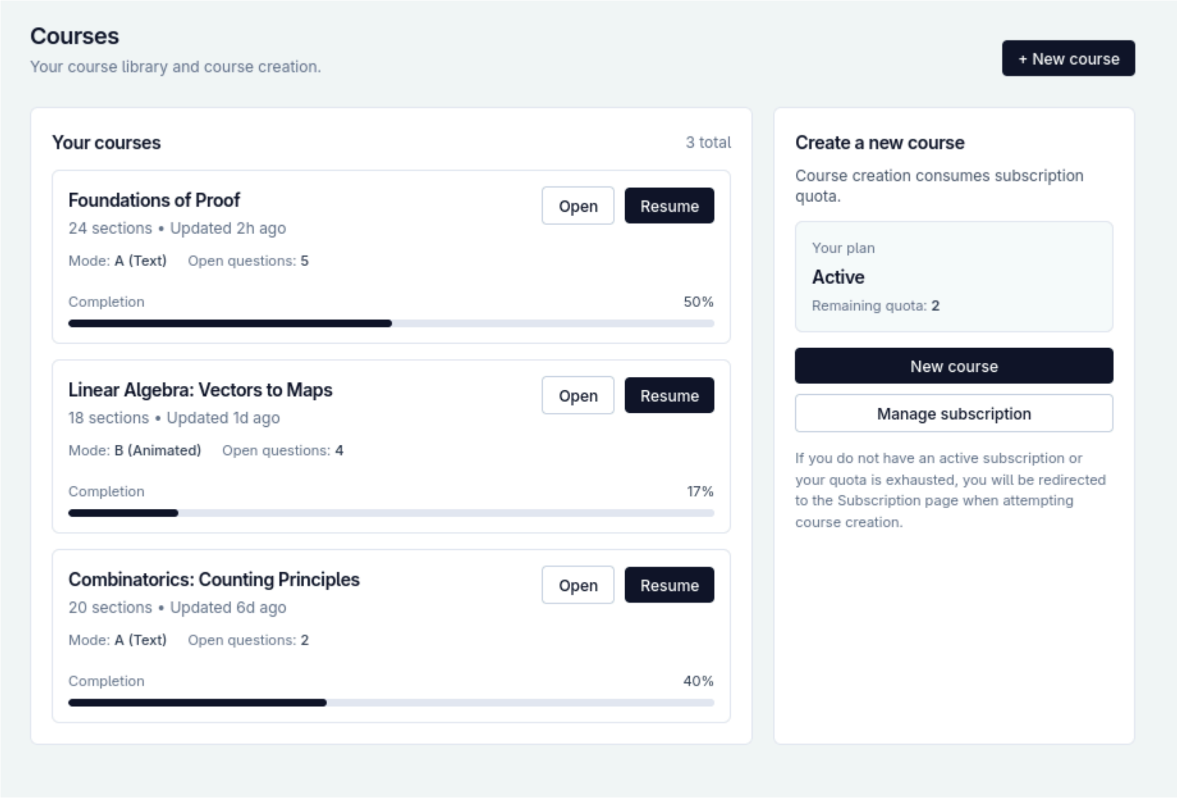
Linear Algebra · Section 1.2
Why does linearity matter?
Unsolved

Questions remain open until the learner arrives at the answer.

Figure 21. Dashboard-Progress

The progress view presents an overview of the user's current learning state across all enrolled courses. It summarizes progress at the course and section level, indicating completion status and outstanding questions, and allows users to resume learning from the appropriate point.

### 3.5.5.2.5 Dashboard - Courses



Courses

Your course library and course creation.

Your courses

3 total

Foundations of Proof

24 sections • Updated 2h ago

Mode: A (Text) Open questions: 5

Completion 50%

Linear Algebra: Vectors to Maps

18 sections • Updated 1d ago

Mode: B (Animated) Open questions: 4

Completion 17%

Combinatorics: Counting Principles

20 sections • Updated 6d ago

Mode: A (Text) Open questions: 2

Completion 40%

Create a new course

Course creation consumes subscription quota.

Your plan

Active

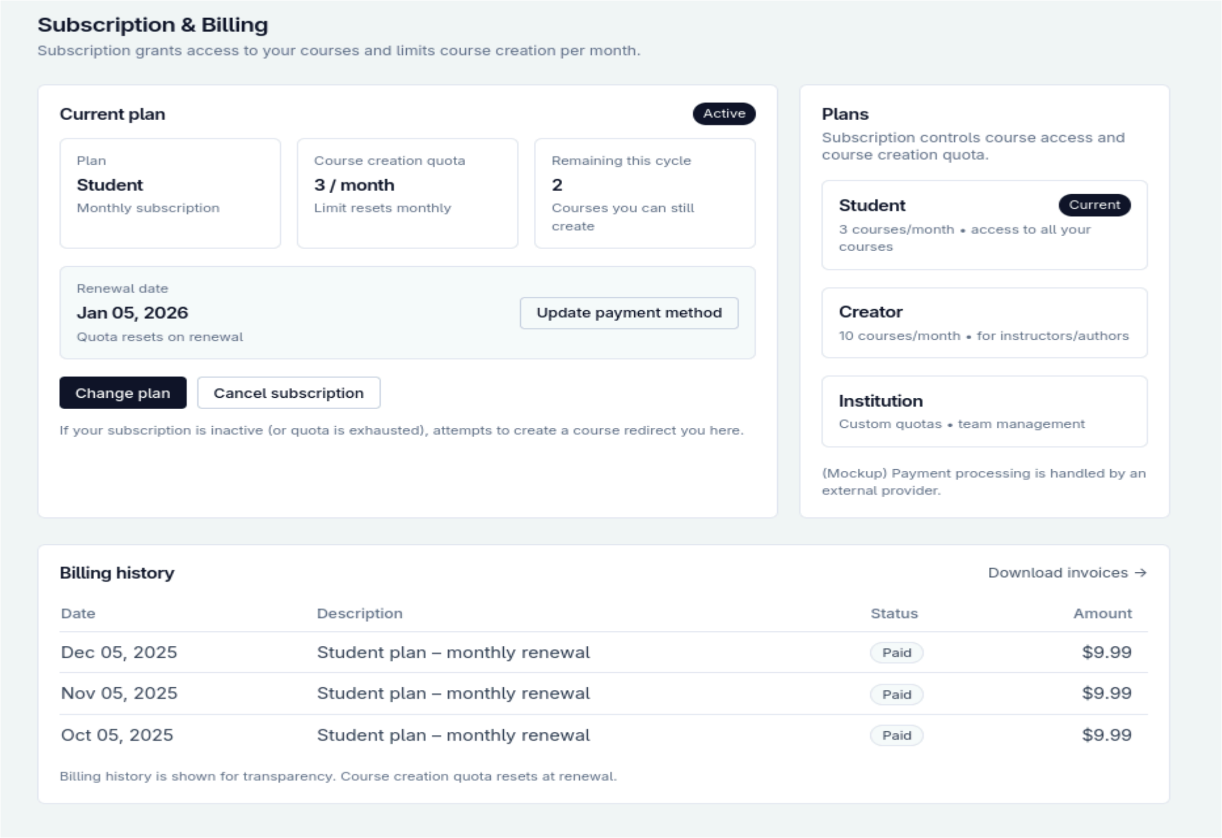
Remaining quota: 2

If you do not have an active subscription or your quota is exhausted, you will be redirected to the Subscription page when attempting course creation.

Figure 22. Dashboard-Courses

The courses view lists all courses associated with the user and provides access to course-specific actions. From this page, users can view existing courses, navigate to individual course overviews, and initiate the creation of new courses when permitted by their subscription.

#### 3.5.5.2.6 Dashboard - Subscription & Billing



Subscription & Billing

Subscription grants access to your courses and limits course creation per month.

Current plan — Active

| Plan | Course creation quota | Remaining this cycle |
|---|---|---|
| Student<br>Monthly subscription | 3 / month<br>Limit resets monthly | 2<br>Courses you can still create |

Renewal date

Jan 05, 2026

Quota resets on renewal

If your subscription is inactive (or quota is exhausted), attempts to create a course redirect you here.

Plans

Subscription controls course access and course creation quota.

- Student — Current: 3 courses/month • access to all your courses
- Creator: 10 courses/month • for instructors/authors
- Institution: Custom quotas • team management

(Mockup) Payment processing is handled by an external provider.

Billing history

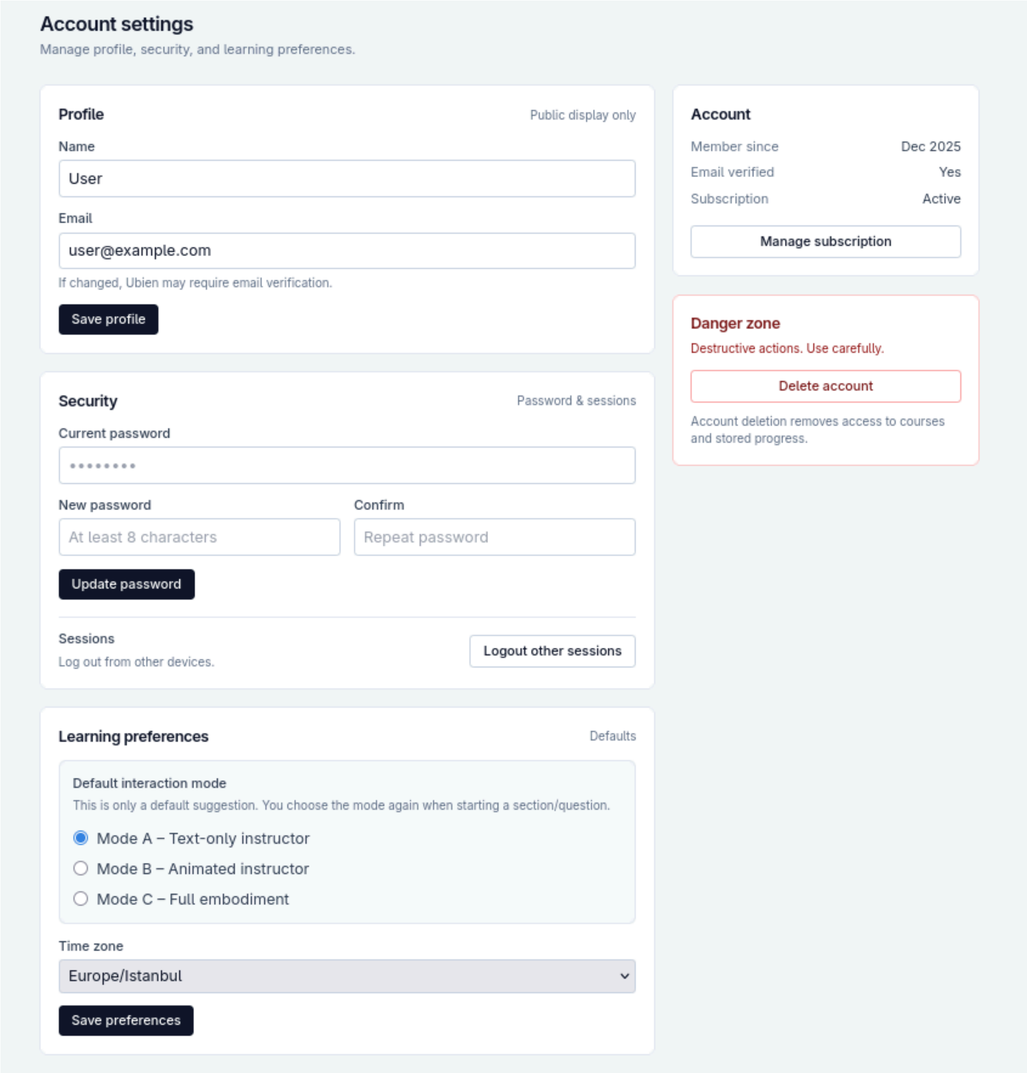
| Date | Description | Status | Amount |
|---|---|---|---|
| Dec 05, 2025 | Student plan – monthly renewal | Paid | $9.99 |
| Nov 05, 2025 | Student plan – monthly renewal | Paid | $9.99 |
| Oct 05, 2025 | Student plan – monthly renewal | Paid | $9.99 |

Billing history is shown for transparency. Course creation quota resets at renewal.

Figure 23. Dashboard-Subscription & Billing

The subscription and billing view displays the user’s current subscription status, course creation quota, and billing information. It allows users to review their active plan, manage payment details, and view billing history, and serves as the point of redirection when a user attempts an action that requires an active subscription.

### 3.5.5.2.7 Dashboard - Account Settings



Account settings

Manage profile, security, and learning preferences.

Profile — Public display only

Name: User

Email: user@example.com

If changed, Ubien may require email verification.

Security — Password & sessions

Current password: ••••••••

New password: At least 8 characters

Confirm: Repeat password

Sessions

Log out from other devices.

Learning preferences — Defaults

Default interaction mode

This is only a default suggestion. You choose the mode again when starting a section/question.

● Mode A – Text-only instructor

○ Mode B – Animated instructor

○ Mode C – Full embodiment

Time zone: Europe/Istanbul

Account

| | |
|---|---|
| Member since | Dec 2025 |
| Email verified | Yes |
| Subscription | Active |

Danger zone

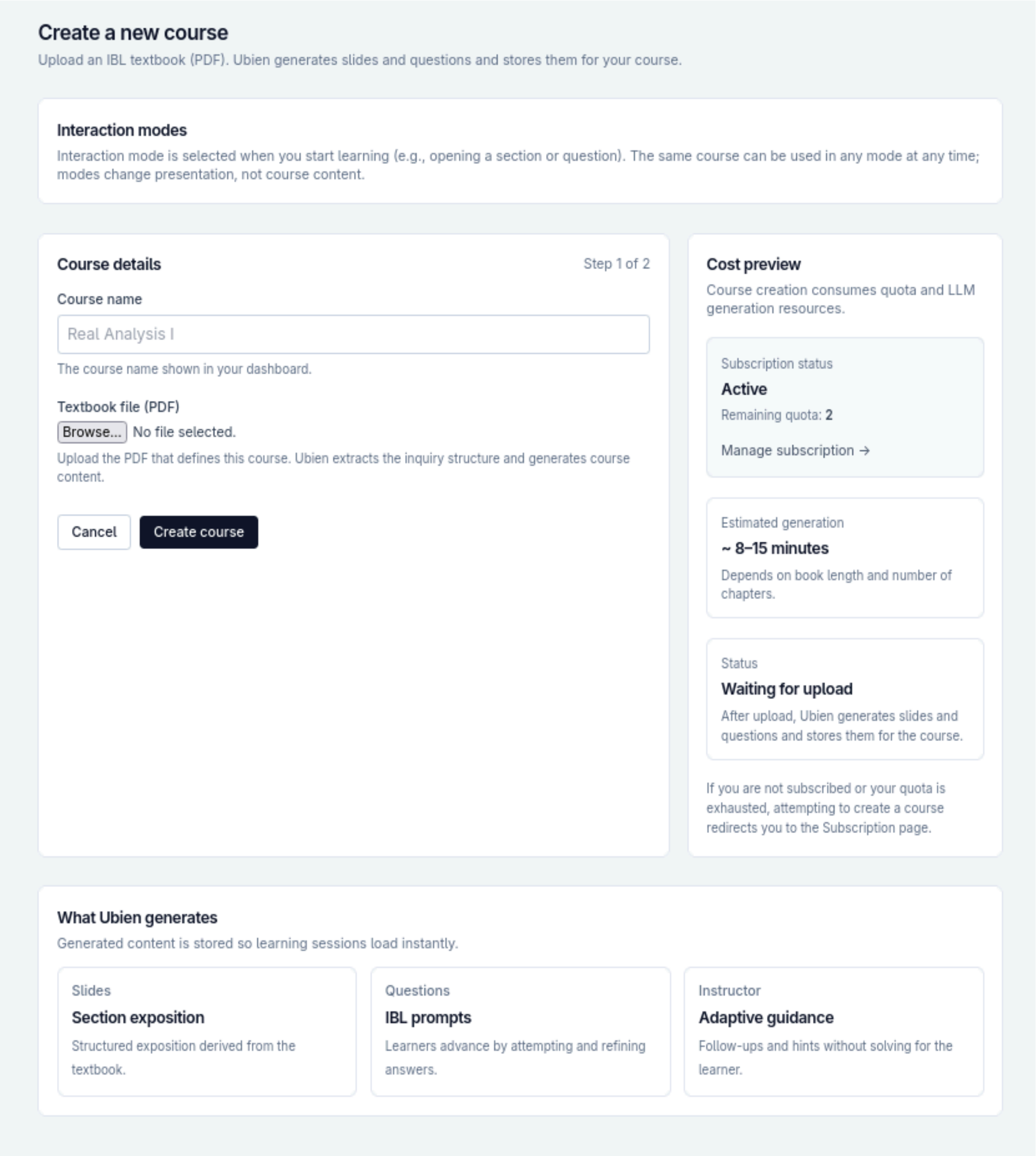
Destructive actions. Use carefully.

Account deletion removes access to courses and stored progress.

Figure 24. Account Settings

The dashboard account settings view allows users to manage personal and system-level preferences. It provides controls for updating profile information, changing security settings, and configuring default learning preferences.

### 3.5.5.2.8 Create Course Pages



Create a new course

Upload an IBL textbook (PDF). Ubien generates slides and questions and stores them for your course.

Interaction modes

Interaction mode is selected when you start learning (e.g., opening a section or question). The same course can be used in any mode at any time; modes change presentation, not course content.

Course details

Step 1 of 2

Course name: Real Analysis I

The course name shown in your dashboard.

Textbook file (PDF): No file selected.

Upload the PDF that defines this course. Ubien extracts the inquiry structure and generates course content.

Cost preview

Course creation consumes quota and LLM generation resources.

Subscription status: Active

Remaining quota: **2**

Estimated generation: ~ 8–15 minutes

Depends on book length and number of chapters.

Status: Waiting for upload

After upload, Ubien generates slides and questions and stores them for the course.

If you are not subscribed or your quota is exhausted, attempting to create a course redirects you to the Subscription page.

What Ubien generates

Generated content is stored so learning sessions load instantly.

Slides: Section exposition

Structured exposition derived from the textbook.

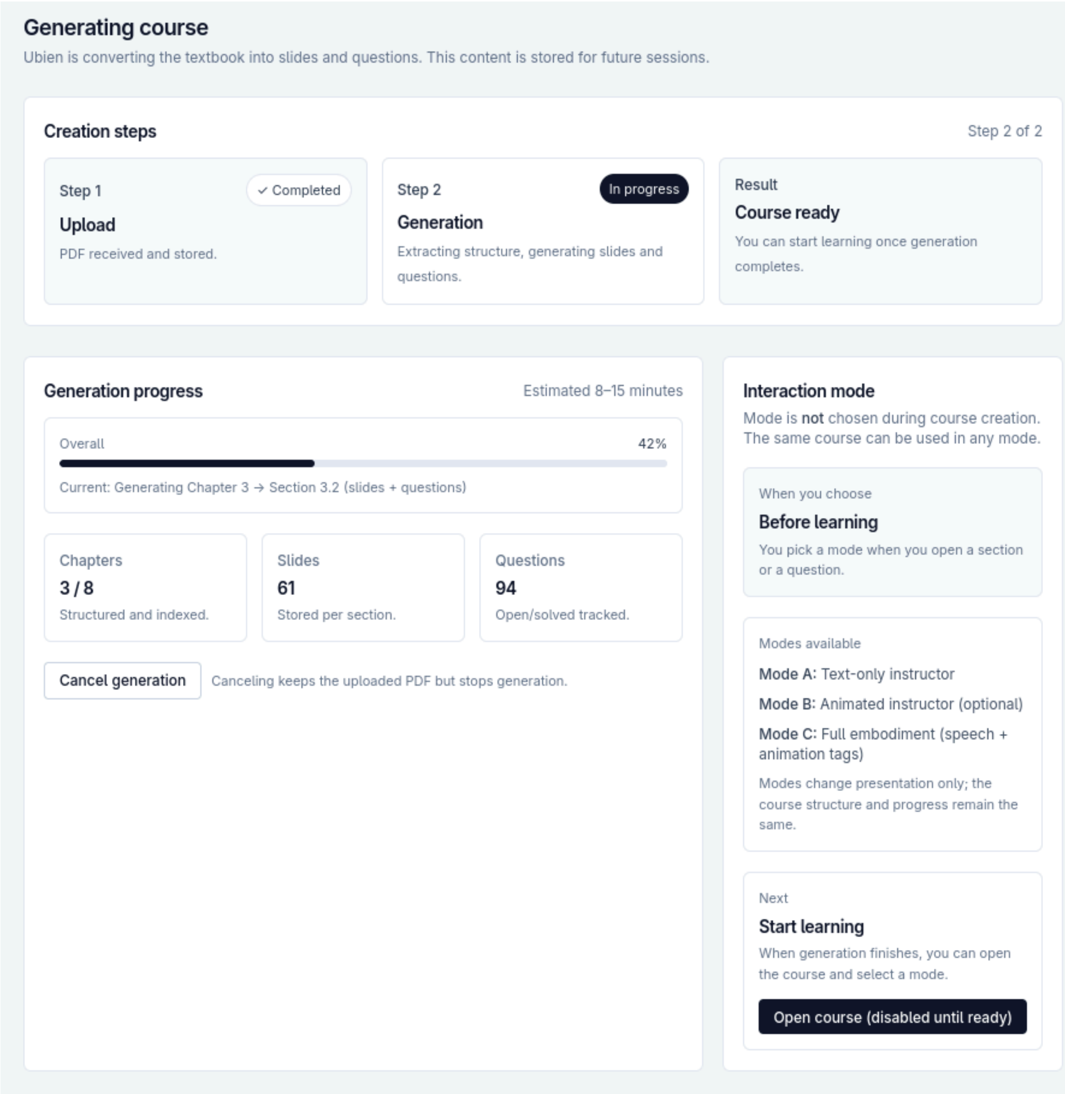
Questions: IBL prompts

Learners advance by attempting and refining answers.

Instructor: Adaptive guidance

Follow-ups and hints without solving for the learner.

Figure 25. Course Creation Page



Generating course

Ubien is converting the textbook into slides and questions. This content is stored for future sessions.

Creation steps

Step 2 of 2

Step 1 ✓ Completed

Upload

PDF received and stored.

Step 2 In progress

Generation

Extracting structure, generating slides and questions.

Result

Course ready

You can start learning once generation completes.

Generation progress

Estimated 8–15 minutes

Overall 42%

Current: Generating Chapter 3 → Section 3.2 (slides + questions)

Chapters

3 / 8

Structured and indexed.

Slides

61

Stored per section.

Questions

94

Open/solved tracked.

Cancel generation

Canceling keeps the uploaded PDF but stops generation.

Interaction mode

Mode is **not** chosen during course creation. The same course can be used in any mode.

When you choose

Before learning

You pick a mode when you open a section or a question.

Modes available

**Mode A:** Text-only instructor

**Mode B:** Animated instructor (optional)

**Mode C:** Full embodiment (speech + animation tags)

Modes change presentation only; the course structure and progress remain the same.

Next

Start learning

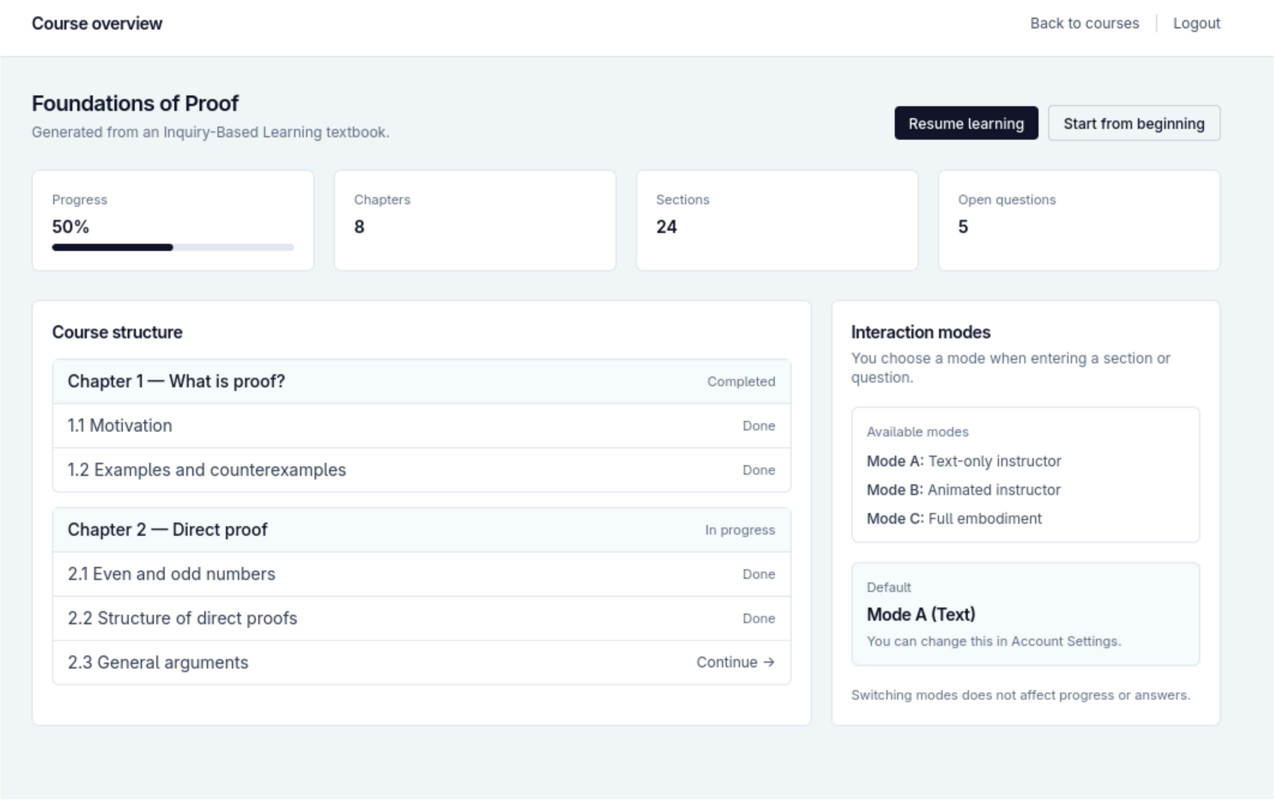
When generation finishes, you can open the course and select a mode.

Open course (disabled until ready)

Figure 26. Course Creation Page Showing the Progress of Book Upload

The create course page enables users to generate a new course by providing a course name and uploading an Inquiry-Based Learning textbook. It guides the user through the initial setup process and reflects the current creation state (e.g., uploading, processing, or generation in progress). Access to this page is constrained by the user’s subscription and course creation quota, with users redirected to the subscription view when limits are exceeded.

### 3.5.5.2.9 Course Overview Page



Course overview

Foundations of Proof

Generated from an Inquiry-Based Learning textbook.

| Progress | Chapters | Sections | Open questions |
|---|---|---|---|
| 50% | 8 | 24 | 5 |

Course structure

| Chapter 1 — What is proof? | Completed |
|---|---|
| 1.1 Motivation | Done |
| 1.2 Examples and counterexamples | Done |

| Chapter 2 — Direct proof | In progress |
|---|---|
| 2.1 Even and odd numbers | Done |
| 2.2 Structure of direct proofs | Done |
| 2.3 General arguments | Continue → |

Interaction modes

You choose a mode when entering a section or question.

Available modes

**Mode A:** Text-only instructor

**Mode B:** Animated instructor

**Mode C:** Full embodiment

Default

**Mode A (Text)**

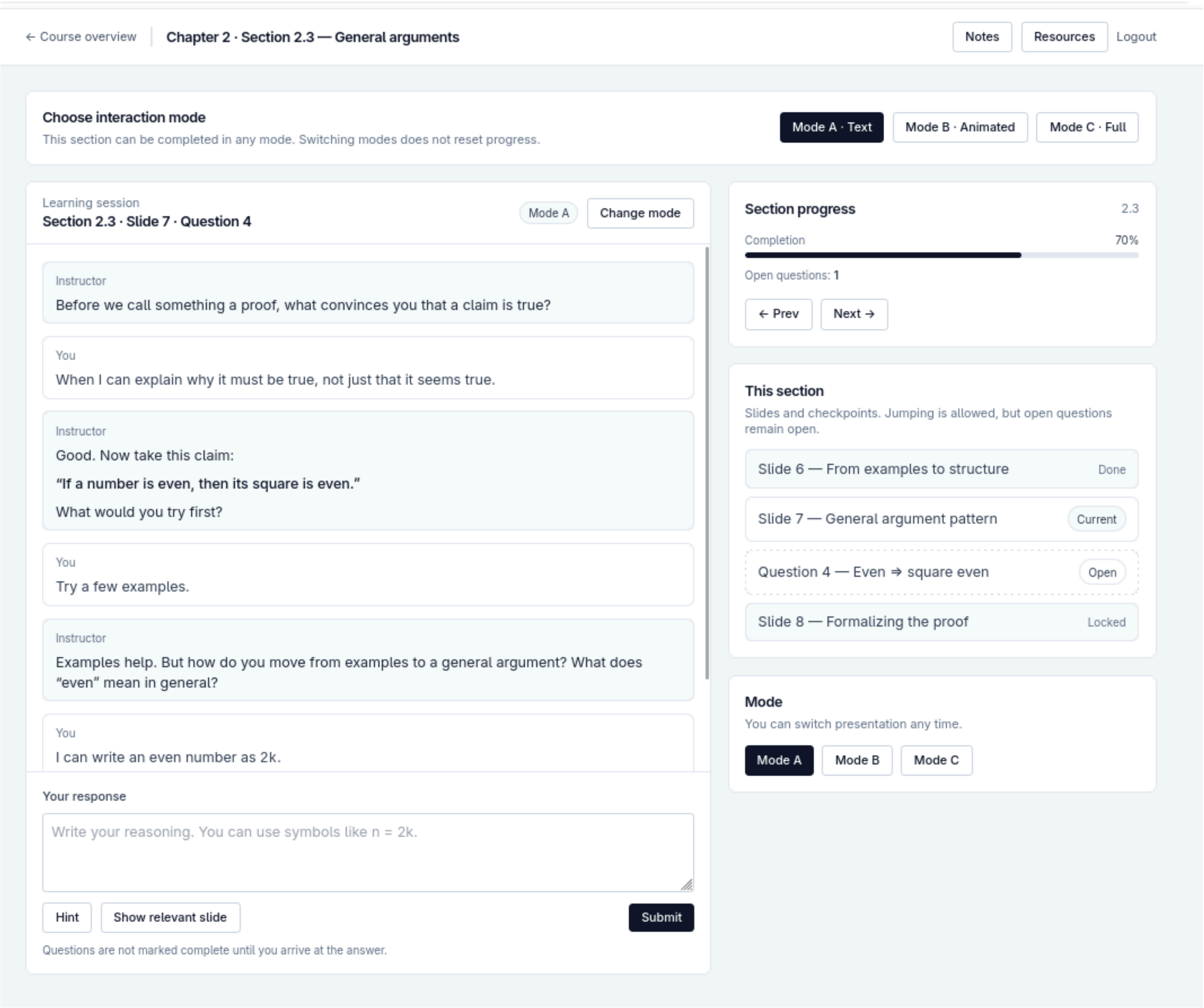
You can change this in Account Settings.

Switching modes does not affect progress or answers.

Figure 27. Overview Page

The course overview page presents the structure and current state of a single course. It summarizes overall progress, lists chapters and sections, and indicates which parts are completed, in progress, or locked. From this page, users can resume learning, start from the beginning, or navigate directly to specific sections

#### 3.5.5.2.10 Learning Pages



← Course overview | Chapter 2 · Section 2.3 — General arguments

Choose interaction mode

This section can be completed in any mode. Switching modes does not reset progress.

Learning session

Section 2.3 · Slide 7 · Question 4

Instructor

Before we call something a proof, what convinces you that a claim is true?

You

When I can explain why it must be true, not just that it seems true.

Instructor

Good. Now take this claim:

**"If a number is even, then its square is even."**

What would you try first?

You

Try a few examples.

Instructor

Examples help. But how do you move from examples to a general argument? What does "even" mean in general?

You

I can write an even number as 2k.

Your response

Write your reasoning. You can use symbols like n = 2k.

Questions are not marked complete until you arrive at the answer.

Section progress 2.3

Completion 70%

Open questions: **1**

This section

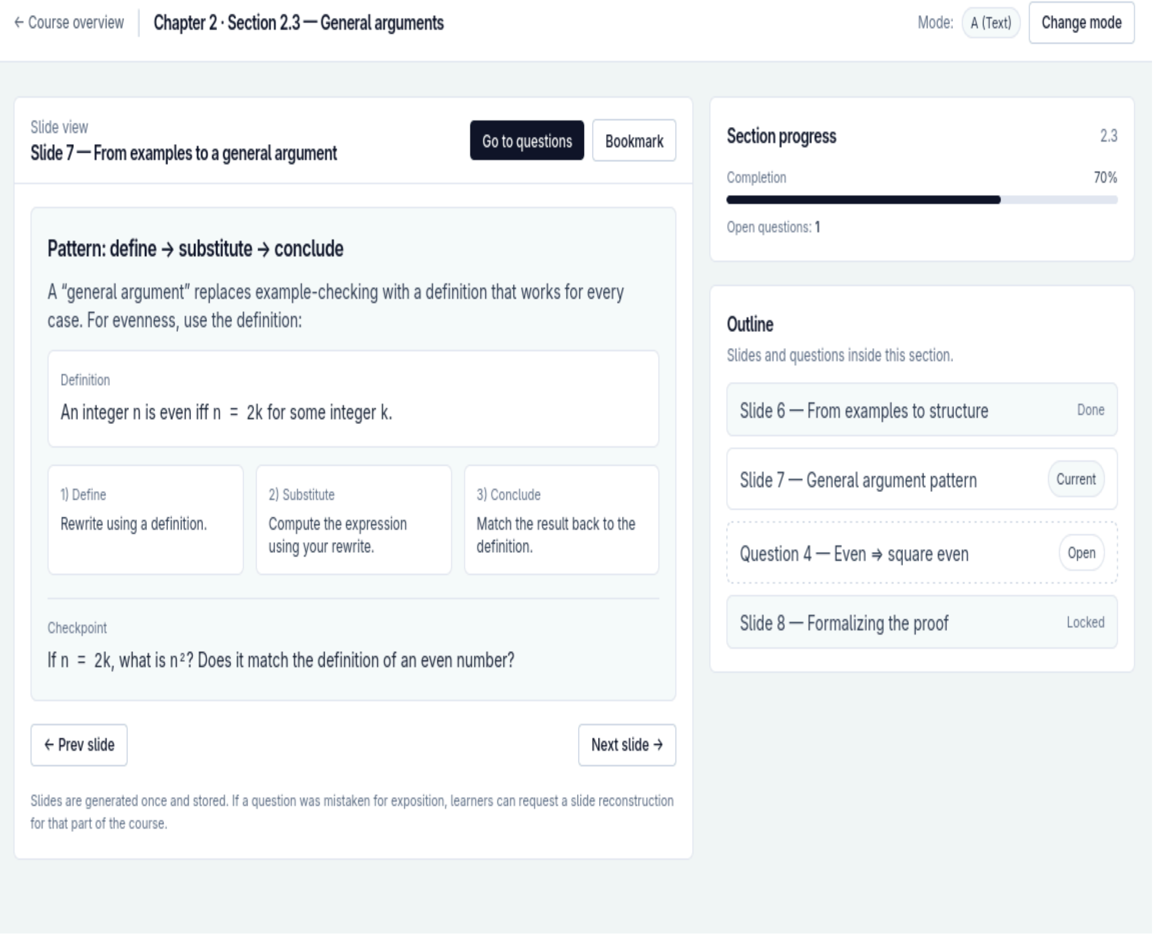
Slides and checkpoints. Jumping is allowed, but open questions remain open.

| | |
|---|---|
| Slide 6 — From examples to structure | Done |
| Slide 7 — General argument pattern | Current |
| Question 4 — Even ⇒ square even | Open |
| Slide 8 — Formalizing the proof | Locked |

Mode

You can switch presentation any time.

Figure 28. Student Learning Page Interacting through Text



Chapter 2 · Section 2.3 — General arguments

Slide view

Slide 7 — From examples to a general argument

Pattern: define → substitute → conclude

A "general argument" replaces example-checking with a definition that works for every case. For evenness, use the definition:

Definition

An integer n is even iff n = 2k for some integer k.

1) Define

Rewrite using a definition.

2) Substitute

Compute the expression using your rewrite.

3) Conclude

Match the result back to the definition.

Checkpoint

If n = 2k, what is $n^2$? Does it match the definition of an even number?

Slides are generated once and stored. If a question was mistaken for exposition, learners can request a slide reconstruction for that part of the course.

Section progress 2.3

Completion 70%

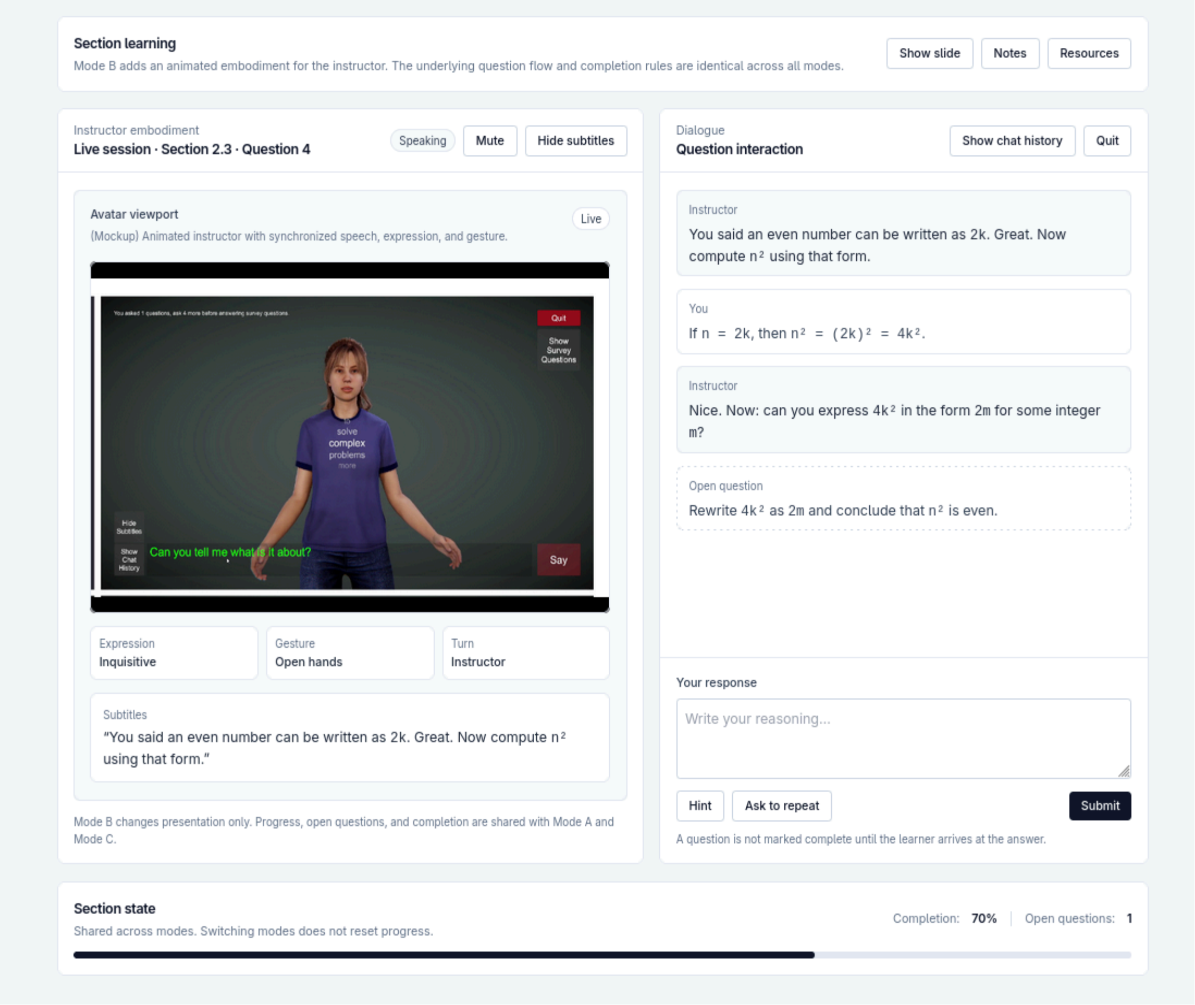
Open questions: **1**

Outline

Slides and questions inside this section.

| | |
|---|---|
| Slide 6 — From examples to structure | Done |
| Slide 7 — General argument pattern | Current |
| Question 4 — Even ⇒ square even | Open |
| Slide 8 — Formalizing the proof | Locked |

Figure 29. Slides Review Page



**Section learning**

Mode B adds an animated embodiment for the instructor. The underlying question flow and completion rules are identical across all modes.

Instructor embodiment

**Live session · Section 2.3 · Question 4**

Avatar viewport

(Mockup) Animated instructor with synchronized speech, expression, and gesture.

| Expression | Gesture | Turn |
|---|---|---|
| Inquisitive | Open hands | Instructor |

Subtitles

"You said an even number can be written as 2k. Great. Now compute $n^2$ using that form."

Mode B changes presentation only. Progress, open questions, and completion are shared with Mode A and Mode C.

Dialogue

**Question interaction**

Instructor

You said an even number can be written as 2k. Great. Now compute $n^2$ using that form.

You

If $n = 2k$, then $n^2 = (2k)^2 = 4k^2$.

Instructor

Nice. Now: can you express $4k^2$ in the form $2m$ for some integer $m$?

Open question

Rewrite $4k^2$ as $2m$ and conclude that $n^2$ is even.

Your response

Write your reasoning...

A question is not marked complete until the learner arrives at the answer.

**Section state**

Shared across modes. Switching modes does not reset progress.

Completion: **70%** | Open questions: **1**

Figure 30. Student Learning Page Interaction with Animation

The learning page is where active learning takes place in Ubien and maintains the context of a specific course section. It alternates between slides and questions derived from the source textbook: slides present necessary exposition or definitions and may be revisited at any time, while questions require the learner to actively reason and respond before progress can continue. Each question acts as a checkpoint and is not marked complete until the learner arrives at the intended reasoning outcome. Different interaction modes affect only how the instructor is presented and do not change the underlying learning flow or progress state.

## 4. Other Analysis Elements

### 4.1. Consideration of Various Factors in Engineering Design

| | Effect level | Effect |
|---|---|---|
| Public health | 2/10 | Ubien is an educational software system and does not directly interact with physical health. However, extended screen usage and cognitive load are considered by promoting short interaction cycles and inquiry-based pacing rather than continuous passive consumption. |
| Public safety | 1/10 | The system does not pose direct safety risks since it operates entirely in a digital environment. Safety considerations are limited to preventing misuse, misinformation, and unauthorized access to user accounts. |
| Public welfare | 7/10 | Ubien aims to improve long-term learning outcomes by encouraging deep understanding rather than memorization, which contributes positively to educational quality and individual intellectual development. |
| Global factors | 5/10 | The system is designed as a web-based platform accessible globally. Differences in internet bandwidth, latency, and cloud availability influence performance considerations and API usage optimization. |

| | Effect level | Effect |
|---|---|---|
| Cultural factors | 4/10 | Learning styles and educational expectations vary across cultures. To address this, Ubien adopts inquiry-based learning principles that are adaptable, while initially limiting content generation to English to reduce complexity. |
| Social factors | 6/10 | Ubien addresses the growing reliance on AI tools for learning and responds by promoting active reasoning instead of passive AI-generated answers. |
| Environmental factors | 3/10 | The system has no direct environmental footprint beyond server energy consumption. Efficient backend processing and limited regeneration of content are considered to reduce unnecessary compute usage. |
| Economic factors | 8/10 | API usage costs, hosting expenses, and subscription pricing heavily influence system design. Token limits, course quotas, caching strategies, and pricing transparency are implemented to ensure financial sustainability. |

Table 1. Factors That Can Affect Analysis and Design

### 4.1.1. Constraints

#### 4.1.1.1 Accessibility Constraints

- Interface must be usable by individuals with basic digital literacy.
- Course content is initially limited to English; multilingual accessibility is out of scope.

#### 4.1.1.2 Maintainability Constraints

- Logging, observability, and modular code structure must support rapid troubleshooting.
- Clean Code principles must guide implementation.

#### 4.1.1.3 Marketability Constraints

- Competitive pricing depends on minimizing backend and API cost overhead.
- Must differentiate from existing tools such as YouLearn, Mindgrasp, and TurboAI [8, 9, 10].

#### 4.1.1.4 Implementation Constraints

- All course generation and TTS must use the OpenAI API through the backend.
- The server must be implemented in Go, with WebSockets for live interaction.
- User accounts, courses, and progress must be stored in PostgreSQL.
- User login must rely on secure server-side sessions (cookie-based), also required for WebSocket upgrades.
- The system must run in Docker containers, both for development and deployment.
- The project must use GitHub for code, issues, and basic CI.

#### 4.1.1.5 Economic Constraints

- API costs and token restrictions may limit the number of interactions that a user can make in a month of subscription.
- The API costs will be covered from the funds obtained from monthly subscriptions. Users will be allowed to use as much as they pay.
- All software will be built using freely available technologies.
- For an initial setup intended to serve up to 100 users, the monthly hosting costs amount to approximately 45 $, broken down as follows: 15 $ (backend server) + 15 $ (managed PostgreSQL DB) + 5 $ (storage) + 10 $ (bandwidth) + 1 $ (domain/DNS) + 0 $ (SSL) + 0 $ (CDN, Cloudflare Free Tier). In the long run, these hosting costs will be covered by allocating a portion of the revenue obtained from monthly subscriptions, allowing Ubien to operate sustainably and profitably.

#### 4.1.1.6 Ethical Constraints

- User data and their chats must stay confidential and encrypted
- Users won't be allowed to share chat logs and courses with others as the course may include personalized or copyrighted material for the user.
- All datasets and educational resources used will comply with fair-use, privacy, and ethical data-handling principles
- All visual and instructional material must be correctly attributed and comply with fair-use principles.

### 4.1.2. Standards

#### 4.1.2.1 Software & System Modelling Standards

- **UML 2.5.1 (ISO/IEC 19505-1:2012)**
  Used for class diagrams, use-case diagrams, activity diagrams, and sequence diagrams to maintain formal and consistent documentation of system models [2].

#### 4.1.2.2 Requirements Engineering Standards

- **IEEE 830 – Software Requirements Specification (SRS)**
  Guides the writing of functional and non-functional requirements to ensure traceability, clarity, and completeness [3].

#### 4.1.2.3 Information Security Standards

- **ISO/IEC 27001 – Information Security Management**
  Applied to user accounts, course metadata, chat logs, and uploaded textbook files. Ensures proper access control, secure storage, and risk-aware handling of sensitive data [4].

#### 4.1.2.4 Ethics and Professional Conduct Standards

- **ACM Code of Ethics and Professional Conduct**
  Guides principles regarding privacy, fairness, transparency, and avoidance of harm during development and interaction with users [5].

#### 4.1.2.5 API and Cloud Service Standards

- **OpenAI API Usage Policies (2024)**
  Governs all AI interactions including course generation, content extraction, tutoring messages, and text-to-speech [6].

#### 4.1.2.6 Code Quality and Maintainability Standards

**Clean Code Principles (Robert C. Martin)**
Ensures readable, maintainable, and extensible backend and frontend implementation [7].

## 4.2. Risks and Alternatives

Ubien is comprised of several unique parts that run in parallel. The three main parts of the project include the backend AI model that creates and stores the relevant information to the course that the user is on, the optional interactive learning assistant that will bring more life to the platform by mimicking human behaviour and speech patterns and the interface of the project which is used to communicate with the backend. All 3 of these main parts can be further divided into subgroups depending on whether the user created their own IBL course with their own material or whether they are using a pre-made course about a topic they found interesting. Having many independent parts that all have to work in parallel makes risk management and having alternatives to some of these systems very important.

Most of the chokepoints of the system lie within the backend where the data-processing and querying of the information occurs. There are many scenarios where the backend may not be able to give sufficient information to the other modules, which may cause issues related to the teacher model speech pattern suffering all the way to the contents of the course being faulty. Other problems in the display may also affect the teacher model or communication with the backend. Possible problems that may occur for every module are listed in the table below with a rate of severity and fallback plans for them.

| Risk | Likelihood | Effect | Fallback plan |
|---|---|---|---|
| Interface cannot connect to the backend AI model | 3/10 | The course cannot be created or the next response cannot be generated. | Error message is displayed to direct the user to other features while the backend issue is being fixed |
| Interface errors stop the user from typing anything or reading something on the screen | 1/10 | The user cannot effectively converse with the AI teacher and the learning experience is mortified. | Reloading the page to refresh the elements that are not working properly. |
| The backend AI model runs out of context window and cannot create a comprehensive and fully developed course | 7/10 | The user experience is mortified and learning quality drops. | Directing the user to other pre-planned courses while the backend re-checks the course to evaluate quality standards. |
| AI model hallucinates and spews out wrong information | 7/10 | The course causes misinformation and the quality of the education is affected. | Backend quality ensurement layer re-checks previous responses to clear out any misinformation and correct them in the future |
| Teacher model lip-syncing is out of sync with the sound | 2/10 | Immersiveness of the teaching is affected negatively. | The teacher model is closed and the user can keep up the conversation in the chatbox. |
| Teacher model face expressions change with no regard to pronunciation | 4/10 | Immersiveness of the teaching is affected negatively. | The teacher model is closed and the user can keep up the conversation in the chatbox. |

Table 2. Potential Risks and Fallback Plans

### 4.3. Project Plan

The project plan is divided into the following work package structure with an explanation of the objectives and due dates of the tasks.

| WP# | Work package title | Leader | Members involved |
|---|---|---|---|
| WP1 | Project Information Form | Ahmet Deniz Gelir | All members |
| WP2 | Assessment of Innovation Form | Ahmet Deniz Gelir | All members |
| WP3 | Project Specification Document | Efe Can Tatar | All members |
| WP4 | Analysis and Requirements Report | Simay Uygur | All members |
| WP5 | Implementation of interactive learning platform and course creator logic | Ahmet Deniz Gelir | Simay Uygur, Kemal Onur Özkan |
| WP6 | Presentation and Demo | Efe Can Tatar | All Members |
| WP7 | Implementation of the virtual instructor | Ahmet Deniz Gelir | Efe Can Tatar, Eren Uslu |
| WP8 | Detailed Design Report | Simay Uygur | All members |
| WP9 | Implementation of pre-trained courses and quality testing of the product | Ahmet Deniz Gelir | Eren Uslu, Kemal Onur Özkan, Simay Uygur |
| WP10 | Final Report | Efe Can Tatar | All members |
| WP11 | Final Presentation and Demo | Ahmet Deniz Gelir | All members |

Table 3. List of work packages

<table>
<tr><td colspan="4"><b>WP 1:</b> <i>Project Information Form</i></td></tr>
<tr><td colspan="4"><b>Start date:</b> <i>10.10.2025</i> <b>End date:</b> <i>24.10.2025</i></td></tr>
<tr><td><b>Leader:</b></td><td><i>Efe Can Tatar</i></td><td><b>Members involved:</b></td><td><i>All members</i></td></tr>
<tr><td colspan="4"><b>Objectives:</b> <i>Fill out and finalize the project proposal. It shall include a high-level summary of the project and the main features to implement.</i></td></tr>
<tr><td colspan="4"><b>Tasks:</b><br><b><i>Task 1.1 Name the project</i>:</b> <i>Come up with a name for the project</i><br><b><i>Task 1.2 Project description</i>:</b> <i>The planned contents of the project shall be summarized in one paragraph</i><br><b><i>Task 1.3 Finalize project supervisor and innovation expert:</i></b> A relevant project supervisor and innovation expert shall be found to continue the project with.<br><b><i>Task 1.4 Project URL creation:</i></b> Either a web page or github repository shall be created to host the code and workings of the project.</td></tr>
<tr><td colspan="4"><b>Deliverables</b><br><b><i>D1.1:</i></b> <i>Filled out project information form according to the format given in the CS491 course page.</i></td></tr>
<tr><td colspan="4"><b>WP 2:</b> <i>Assessment of Innovation Form</i></td></tr>
<tr><td colspan="4"><b>Start date:</b> <i>24.10.2025</i> <b>End date:</b> <i>31.10.2025</i></td></tr>
<tr><td><b>Leader:</b></td><td><i>Ahmet Deniz Gelir</i></td><td><b>Members involved:</b></td><td><i>All members</i></td></tr>
<tr><td colspan="4"><b>Objectives:</b> <i>Finding an expert in the field that can analyze the innovative aspects of the project and guide the team in a better direction that utilizes the newest technologies.</i></td></tr>
<tr><td colspan="4"><b>Tasks:</b><br><b><i>Task 2.1</i></b> Communicate with the project’s innovation expert for an analysis of the project innovation aspect<br><b><i>Task 2.1</i></b> Upload the filled out innovation form to the project page.</td></tr>
</table>

| |
|---|
| **Deliverables**<br>***D2.1:*** *The project innovation form filled out by the project's assigned innovation expert* |

| | | | |
|---|---|---|---|
| **WP 3:** *Project Specification Document* | | | |
| **Start date:** *7.11.2025* **End date:** *28.11.2025* | | | |
| **Leader:** | Efe Can Tatar | **Members involved:** | All members |
| **Objectives:** Filling out the project specification document that details the deliverable list and objectives of the project on a high level. This document shall also contain information about the project constraints such as economic, environmental and ethical concerns. | | | |
| **Tasks:**<br>***Task 3.1:*** *Discuss and finalize the deliverable list for the project as a team.*<br>***Task 3.2:*** *Fill out the project specification document outlined in the CS491 web-page and upload it to the course page.* | | | |
| **Deliverables**<br>***D3.1:*** Filled out version of the project specification document | | | |

| | | | |
|---|---|---|---|
| **WP 4:** *Analysis and Requirements Report* | | | |
| **Start date:** *28.11.2025* **End date:** *19.12.2025* | | | |
| **Leader:** | Simay Uygur | **Members involved:** | All members |
| **Objectives:** Filling out the analysis and requirements report which outlines the project requirements in detail and analyzes the main problem that the project aims to solve. This includes doing market research and comparing the product with competitors. Feasibility of the project and how the building process of the system will be conducted shall be carefully explained in this report. | | | |
| **Tasks:** | | | |

***Task 4.1:*** Discuss and finalize the system architecture that will be implemented.

***Task 4.2:*** Conduct market research and academic analysis to compare with products already in market and detail the elements that make the project different from what is already on the market.

***Task 4.3:*** Fill out the analysis and requirements report that is outlined in the CS491 course page.

**Deliverables**

***D4.1:*** Filled out version of the analysis and requirements report

**WP 5:** *Implementation of interactive learning platform and course creator logic*

**Start date:** *01.11.2025* **End date:** *26.12.2025*

| | | | |
|---|---|---|---|
| **Leader:** | Ahmet Deniz Gelir | **Members involved:** | Simay Uygur, Kemal Onur Özkan |

**Objectives:** Implement the web page and large language model integration of the system on a basic scale. The user should be able to converse with many models through the web page. The basic structure of the project shall be created to a working extent and inquiry based learning trials should be conducted to gain insight into what models are efficient for the task.

**Tasks:**

***Task 5.1:*** Creation of the web-page.

***Task 5.2:*** Implementation of the large language model conversation with the servers.

***Task 5.3:*** Fine tuning of the large language model to fit the inquiry based learning principles

***Task 5.4:*** Ensuring quality control in the inquiry based conversation structure of the platform.

**Deliverables**

***D5.1:*** Basic working version of the product web-page.

***D5.2:*** Conversational large language model implementation that is fined tuned for inquiry based learning.

| | | | |
|---|---|---|---|
| **WP 6:** *Presentation and Demo* | | | |
| **Start date:** *26.12.2025* **End date:** *26.12.2025* | | | |
| **Leader:** | Efe Can Tatar | **Members involved:** | All members |
| **Objectives:** Present the current workings of the project to the course instructors and supervisors. | | | |
| **Tasks:**<br>***Task 6.1:*** Prepare a slide to go over the implemented features and do a live demo with the working project model. | | | |
| **Deliverables:** None | | | |

| | | | |
|---|---|---|---|
| **WP 7:** *Implementation of the virtual instructor* | | | |
| **Start date:** *01.01.2026* **End date:** *28.02.2026* | | | |
| **Leader:** | Ahmet Deniz Gelir | **Members involved:** | Efe Can Tatar, Eren Uslu |
| **Objectives:** Implement the virtual visual instructor for a more immersive learning experience. This should include a working 2D or 3D model with facial expressions and lip syncing capabilities. The instructor should be able to speak to the user with a voice model that utilizes artificial intelligence. | | | |
| **Tasks:**<br>***Task 7.1:*** Create a model for the instructor<br>***Task 7.2:*** Rig the model to have precise facial expressions.<br>***Task 7.3:*** Implement a context understanding model to adjust the facial expression and tonal changes of the instructors. | | | |
| **Deliverables**<br>***D7.1:*** Working model of the human-like inquiry based learning instructor | | | |

**WP 8:** *Detailed Design Report*

**Start date:** *01.02.2026* **End date:** *10.03.2026*

| | | | |
|---|---|---|---|
| **Leader:** | Simay Uygur | **Members involved:** | All members |

**Objectives:** Transforming the system analysis into a complete detailed design by defining design goals, subsystem decomposition, architectural decisions, and implementation strategies. The report documents the design principles, models, and diagrams used to realize the system, while also addressing public, social, environmental, and economic considerations. The report also documents integration test procedures to ensure the system meets its requirements prior to final implementation.

**Tasks:**

***Task 8.1:*** Decompose the system into clear subsystems and describe their responsibilities.

***Task 8.2:*** Define functional and non-functional integration test cases with clear procedures and expected results. Ensure tests can be executed by test engineers without access to the source code.

**Deliverables**

***D8.1:*** Filled out version of the detailed design report

**WP 9:** *Implementation of pre-trained courses and quality testing of the product*

**Start date:** *01.02.2026* **End date:** *01.03.2026*

| | | | |
|---|---|---|---|
| **Leader:** | Ahmet Deniz Gelir | **Members involved:** | Eren Uslu, Kemal Onur Özkan, Simay Uygur |

**Objectives:** Ensuring quality of the product before finalizing every change to the system. This includes rigorous testing of the large language models we have implemented and quality assurance of the inner workings of the system. Multiple pre-trained courses about various topics shall be created at this step for the product.

**Tasks:**

***Task 9.1:*** Checking all functionalities of the product to see whether they work as intended.

***Task 9.2:*** Comparing functionalities of the product with deliverables set in the prior reports to see if any functionality is missing.

***Task 9.3:*** Setting up integration and unit tests for all the outlined functionalities to ensure proper structure.

**Deliverables**

***D9.1:*** None

**WP 10:** *Final Report*

**Start date:** *01.02.2026* **End date:** *02.05.2026*

| | | | |
|---|---|---|---|
| **Leader:** | Efe Can Tatar | **Members involved:** | All members |

**Objectives:** Present a complete documentation of the finished system, including refined requirements, final architecture and design, implementation details, testing results and maintenance plans.

**Tasks:**

***Task 10.1:*** Refine and document the final system requirements, architecture and detailed design decisions.

***Task 10.2:*** Define how the system will be maintained and updated in the future

***Task 10.3:*** Complete final code commits and prepare a user manual for easy use.

**Deliverables**

***D10.1:*** Filled out version of the final report document

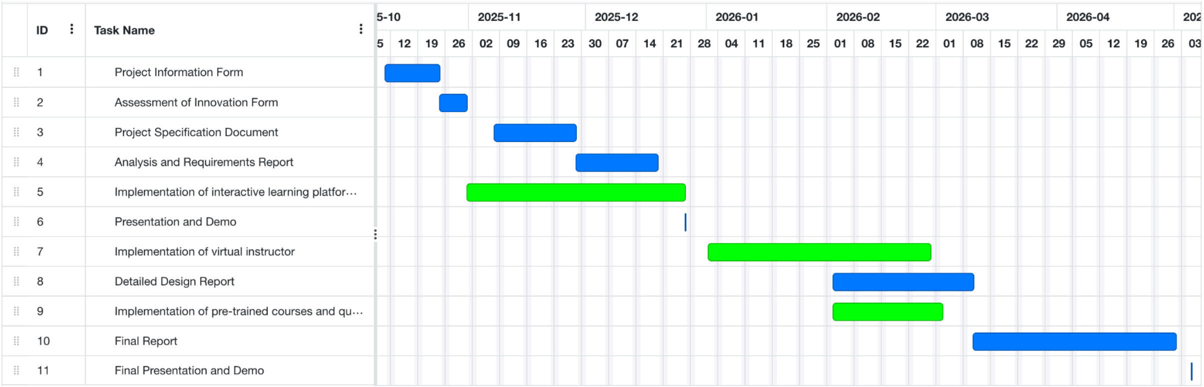
**WP 11:** *Final Presentation and Demo*

**Start date:** *05.05.2026* **End date:** *05.05.2026*

| | | | |
|---|---|---|---|
| **Leader:** | Ahmet Deniz Gelir | **Members involved:** | All members |

|  |
|---|
| **Objectives:** Present the final version of the project to course instructors and supervisors as a complete and standalone product. |
| **Tasks:**<br>***Task 11.1:*** Organize slides to go over newest implementations to the project<br>***Task 11.2:*** Launch the web-service to demonstrate the project capabilities live. |
| **Deliverables**<br>***D11.1:*** None |

The project will follow the structure of the following Gantt chart. Blue tasks represent report writing and presentation activities while the green tasks represent the implementation and engineering activities. As seen, the project follows a hybrid structure with the report writing being done in a waterfall style while the implementation tasks are done in month-long sprint sessions alongside the report writing.



| ID | Task Name |
|---|---|
| 1 | Project Information Form |
| 2 | Assessment of Innovation Form |
| 3 | Project Specification Document |
| 4 | Analysis and Requirements Report |
| 5 | Implementation of interactive learning platfor… |
| 6 | Presentation and Demo |
| 7 | Implementation of virtual instructor |
| 8 | Detailed Design Report |
| 9 | Implementation of pre-trained courses and qu… |
| 10 | Final Report |
| 11 | Final Presentation and Demo |

Figure 31. Project Gantt Chart

## 4.4. Ensuring Proper Teamwork

The project plan is decided before work begins on any deadline or task that is shared by the team members. This plan includes details of when each member should contribute to each feature and what additions should be done.

It is the job of the project manager to ensure every task is done on time by all members and if not, rescheduled accordingly. Efe Can Tatar assumes the position of the project manager alongside the developer role. Every member is free to help out and discuss the workload and communicate it accordingly with everyone to re-distribute the workload in a fair way.

The team will follow a mix between agile and waterfall project management methods where planned features are distributed among each member to implement. New features may be added if the ideas are approved by most members and the distribution may change multiple times in a week with short notice.

This structure ensures every member can develop as much functionality as possible with easy ways to reschedule and redistribute their workload and get help from other teammates if necessary.

### 4.5. Ethics and Professional Responsibilities

Throughout the development and future operation of Ubien, the team recognizes and adheres to ethical and professional responsibilities concerning user privacy, data protection, intellectual property rights, and the responsible use of artificial intelligence. The system is designed under the principle that all educational materials, including potentially copyrighted textbooks, are provided directly by users who possess the right to use them for personal and educational purposes. Ubien does not maintain independent copies of copyrighted books for redistribution, nor does it offer shared access to uploaded materials. Uploaded content is processed solely to generate personalized, user-specific learning experiences and is stored only to the extent necessary to provide the intended service, in accordance with fair-use principles.

User data, including uploaded documents, chat interactions, and progress records, is handled in compliance with the Turkish Personal Data Protection Law (KVKK, Law No. 6698). The system enforces confidentiality, access control, and secure storage mechanisms to protect personal data against unauthorized access, misuse, or disclosure. Data processing practices remain transparent and limited to clearly defined purposes, and no personal data is shared with third parties beyond required external services operating under contractual and legal obligations. In alignment with the ACM Code of Ethics, the team prioritizes fairness, transparency, and the prevention of harm, particularly in the use of large language models. Ubien is designed to support critical thinking rather than present generated responses as absolute truth, and safeguards are continuously considered to reduce misinformation, bias, and misuse both during development and in future iterations of the system.

### 4.6. Planning for New Knowledge and Learning Strategies

The Ubien project is structured around continuous learning, as its scope requires the acquisition and application of new knowledge across software architecture, artificial intelligence systems, inquiry-based learning pedagogy, data security, and legal compliance. From the initial analysis phase through deployment and future maintenance, the team actively identifies knowledge gaps and plans learning activities to address them. This includes studying academic literature, official documentation, industry standards, and legal frameworks such as KVKK and AI usage policies to ensure both technical correctness and regulatory compliance.

Learning strategies emphasize a combination of self-directed research, collaborative knowledge sharing, and iterative experimentation. Team members document findings, share insights during regular meetings, and apply newly acquired concepts directly through prototyping and incremental system improvements. This approach ensures that learning is not limited to theoretical understanding but is continuously translated into practical design and implementation decisions. As the system evolves, the team plans to maintain this adaptive learning mindset to respond to emerging technologies, updated regulations, and new ethical considerations, ensuring the long-term sustainability and responsibility of the Ubien platform.

## 5. Glossary

TTS: Text-to-Speech

API: Application Programming Interface

# 6. References

[1] I. Kaiser, J. Mayer, and D. Malai, “Self-Generation in the Context of Inquiry-Based Learning,” *Frontiers in Psychology*, vol. 9, 2018. doi: 10.3389/fpsyg.2018.02440.

[2] International Organization for Standardization, "Information technology -- Object Management Group Unified Modeling Language (OMG UML)," ISO/IEC 19505-1:2012, 2012.

[3] *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998, 1998.

[4] International Organization for Standardization, "Information security, cybersecurity and privacy protection -- Information security management systems -- Requirements," ISO/IEC 27001:2022, 2022.

[5] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," 2018. [Online]. Available: https://www.acm.org/code-of-ethics

[6] OpenAI, "Usage Policies," 2024. [Online]. Available: https://openai.com/policies/usage-policies

[7] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River, NJ, USA: Prentice Hall, 2008.

[8] YouLearn, "YouLearn: The AI Tutor for Students," 2024. [Online]. Available: https://www.youlearn.ai

[9] Mindgrasp AI, "Mindgrasp: The World's #1 AI Learning Assistant," 2024. [Online]. Available: https://mindgrasp.ai

[10] TurboLearn AI, "TurboLearn AI: Instant Notes & Flashcards," 2024. [Online]. Available: https://www.turbolearn.ai