YEDİTEPE UNIVERSITY

FACULTY OF ENGINEERING DEPARTMENT OF

INDUSTRIAL ENGINEERING AND COMPUTER ENGINEERING

ISE402-CSE344 SPRING 2025 TERM PROJECT DESIGN REPORT

GROUP 6

TutorUP

ISE TEAM

EZGİ DENİZER - 20200703089

BEGÜM DOBRUCALI - 20200703081

CSE TEAM

EMRECAN BOZKURT – 20210702007

MUHARREM EFE BAŞBAYRAM - 20210702093

ALİ OZAN OCAK - 20210702037

ALPER ÖZDEMİR – 20210702058

ARDA EGEMEN İPEK – 20210702139

FURKAN UYSAL - 20210702076

Instructor: Dr. Eylül Damla Gonül Sezer

Instructor: Assoc. Prof. Dr. Mert Özkaya

Spring 2025

TABLE OF CONTENT

1 Introduction 4

1.1 Purpose of the Document 4

1.2 Purpose of the System 4

1.2.1 Describe the New System 4

1.3 Background 5

1.4 Motivation 6

1.5 Structure of the Document 7

2 Functional Requirements 8

2.1 Description of the system functionalities 10

2.1.1 Authentication Page 10

2.1.2 Main Page 12

2.1.3 Lesson Search Page 14

2.1.4 Course Creation Page 14

2.1.5 Profile Page 14

2.1.6 Booking Page 15

2.1.7 Instructor Dashboard 15

2.1.8 Notification System 15

2.1.9 Feedback & Rating System 16

2.1.10 Messaging System 16

2.2 Description of the system users 16

2.2.1 Learners 16

2.2.2 Instructors 17

2.2.3 Administrators (Platform Managers) 17

2.3 Specific Requirements 17

2.3.1 Use Case Diagram 19

2.3.2 Use Case Priority List 20

2.3.3 Use Case Specifications 21

3 Non-Functional Requirements 29

3.1 Volere Template 30

4 System Models 39

4.1 UML Class Diagram for Domain Analysis 39

4.2 User Interface - Navigational Paths and Screen Mock-Ups 40

5 Customer Experience 42

6 Systems Architecture Models 44

6.1 UML Component Diagram (Software Architecture) 45

6.2 2.2. UML Package Diagram (Physical Architecture) 46

7 Detailed Design Class Diagram 47

8 Dynamic Models 48

8.1 Sequence Diagrams 48

8.1.1 User Login Sequence 48

8.1.2 Instructor Schedules a Lesson 49

8.1.3 Student Registration Process 49

8.1.4 Student Submits a Review for a Lesson 50

8.1.5 Instructor Creates a New Lesson 50

8.1.6 4.1.6. Student Books a Lesson 50

8.2 State Diagrams 51

8.2.1 Instructor Mode State Diagram 51

8.2.2 Learner Mode State Diagram 52

8.3 Activity Diagrams 53

8.3.1 Student Login and Registration 53

8.3.2 Instructor Creates Lesson and Manages Schedule 54

8.3.3 Student Lesson Booking and Review 55

9 Entity Relationship Diagram 56

10 Meeting Minutes 57

11 Glossary & References 70

1 Introduction

1.1 Purpose of the Document

This report aims to provide information about the software development project between the ISE and CSE groups. The project involves the design and development of an innovative educational platform where university students can take private lessons from one another. The report outlines the system requirements, user interfaces, various sequences of user-initiated events, and the behavioral patterns of the system, accompanied by relevant diagrams. Additionally, it serves as a guide to facilitate the software development process.

1.2 Purpose of the System

This project aims to provide a web-based platform to support private tutoring processes among university students. After registering on the system, users will be able to choose either the "taking lessons" or "giving lessons" option and create personalized accounts accordingly. The system allows users to specify their preferred subjects, schedules, and price ranges, while offering profile creation and evaluation mechanisms for tutors.

Additionally, the project aims to ensure the privacy of user data, maintain high system performance, incorporate user feedback for continuous improvements, and provide technical support. The platform is designed with scalability in mind, making it adaptable to increasing user numbers and data volumes. With its user-friendly interface and advanced search features, the platform enables students to quickly meet their needs and enjoy an efficient learning experience.

1.2.1 Describe the New System

The new system is a web-based platform designed to make the processes of taking and giving lessons among university students more efficient and secure. During the registration process, users can choose either the "taking lessons" or "giving lessons" option to create a personalized account. The system addresses individual learning needs by matching students who want to take lessons with upperclassmen who have previously completed the course successfully. Users can select lesson topics, specify price ranges, and schedule sessions at suitable times.

One of the key features of the platform is the password recovery mechanism, which ensures secure access to user accounts. If users forget their passwords, they can quickly initiate a password reset request via email or phone number, allowing them to regain access to their accounts safely and seamlessly.

Users who wish to give lessons can create profiles based on their areas of expertise and share lesson details on the platform. To foster trust between lesson providers and recipients, the system includes features like reviews and ratings. The database securely stores user information and all interactions on the platform, while advanced security protocols protect data privacy.

The platform is developed with a user-friendly interface and advanced search and filtering capabilities, enabling users to quickly find solutions tailored to their needs. Its scalable infrastructure maintains performance even under heavy usage, adapting to increasing user demands. Technical support and regular system updates are provided to continuously improve the platform's functionality and ensure maximum user satisfaction. The system aims to offer a safe, fast, and accessible learning environment, contributing to users' academic success.

1.3 Background

In recent years, significant changes have occurred in the methods of taking and giving private lessons. The rise of digitalization and online platforms has introduced new ways to facilitate educational processes for individuals. Although current online tutoring platforms accelerate learning and teaching processes, they often face issues such as a lack of personalization and high costs. Additionally, the variability of course content from one university to another and according to the priorities of professors makes general solutions less effective. For this reason, the idea of developing a personalized and university-focused platform has emerged.

These trends have highlighted the need for more targeted solutions. Observations and surveys conducted among students have revealed that learning a course from upperclassmen who have successfully completed it previously can strengthen individual learning experiences. Furthermore, the pricing policies and teacher-centered approaches of existing platforms fail to fully meet the needs of university students. This situation has underscored the necessity of creating a platform specifically for university students, which promotes knowledge sharing and academic collaboration.

Our platform aims to address these shortcomings. Our system distinguishes itself from traditional online tutoring platforms by exclusively providing tutoring services among university students. After registering on the system, users can select either the "taking lessons" or "giving lessons" option and personalize their accounts based on their needs. Additionally, the opportunity to learn from upperclassmen offers students a chance to better understand the course content and engage in exam-focused learning processes. The campus-focused approach of the system ensures the creation of a safe and targeted community for users.

This innovative platform is expected to support knowledge sharing, simplify learning processes, and enhance academic success across universities. Furthermore, with its affordable pricing policies and user-friendly interface, it aims to become widely adopted among students.

1.4 Motivation

Knowledge sharing is one of the most essential elements that supports learning processes and academic success. Organizing the processes of university students taking or giving lessons to one another presents a significant opportunity to address individual needs in education. However, existing platforms often fail to fully support personalized learning and teaching experiences for students, which reduces effectiveness in education. In particular, the fact that course content varies from one university to another, or even according to the priorities of professors, makes learning through traditional methods more challenging.

The idea of taking lessons from upperclassmen creates opportunities for knowledge sharing tailored to individual needs and practical-focused learning. This system facilitates the transfer of knowledge and experience while contributing to the academic success of both students who wish to take lessons and those who wish to give them. The shortcomings of traditional tutoring platforms and the challenges encountered in this process have highlighted the need to develop a more innovative and user-friendly system.

One disadvantage of traditional systems is their inability to fully ensure user security and data privacy. This can reduce students' trust in platforms and hinder their efficient use. For these reasons, developing a software-based educational platform can make learning and teaching processes more secure, faster, and more accessible. A campus-focused system specifically

designed for university students aims to address their needs more clearly and organize knowledge sharing more effectively.

Through this platform, students can easily find the lessons they need, benefit from suitable pricing options, and optimize their learning processes with just a few clicks. Creating a better environment for both students giving lessons and those receiving them forms the core motivation of this project.

1.5 Structure of the Document

This document provides a comprehensive analysis and design overview of the peer-to-peer learning platform that enables university students to offer and receive private lessons from one another. The document is structured to reflect each phase of system planning, functional modeling, and requirement specification, ensuring clarity for both technical and non-technical stakeholders.

The document begins with an Introduction, which outlines the project's purpose, background, and the motivation behind developing such a system. It identifies the limitations of traditional and existing online learning platforms, particularly in their lack of peer-based, university-specific support networks. It then introduces the proposed solution: a dynamic web-based application where students can become both learners and instructors within a controlled and trusted academic environment.

Following the introduction, the System Overview explains the key roles involved in the system (Learner, Instructor, and Admin) and how the core components interact. This section is essential for understanding how the system behaves at a macro level before diving into specific details.

The Functional Requirements section enumerates the expected system behaviors using precise "The system shall…" statements. These are followed by the Non-Functional Requirements, which define performance metrics, security standards, usability expectations, scalability needs, and compliance with data protection laws.

A central part of the document is the Use Case Modeling. It includes a detailed Use Case Diagram illustrating all possible interactions between users and the system, and a Use Case Priority Table that highlights the significance of each function. Each individual use case is then

explained thoroughly in the Use Case Specifications section, using a structured tabular format that defines actors, preconditions, main and alternative flows, and postconditions.

The User Description section introduces the three primary user types—learners, instructors, and administrators—detailing their permissions, responsibilities, and how they interact with the platform.

The Interface Design and Navigation section provides descriptions of each page in the system, such as the authentication page, search and booking pages, course creation, messaging, and profile pages. This helps bridge the gap between user experience and technical design.

Finally, the document concludes with a brief summary of how the proposed system addresses the initially identified problems and offers a scalable, flexible, and user-centered approach to academic peer tutoring. Future enhancements and possible integrations are also briefly considered to support the project's long-term vision.

2 Functional Requirements

- The system shall allow users to register using their university email.
- The system shall require student identity verification during registration.
- The system shall allow users to log in using their registered credentials.
- The system shall support password reset functionality via email verification.
- The system shall prevent unauthorized access by implementing secure authentication.
- The system shall allow users to choose between being an instructor or a learner during registration.
- The system shall allow users to switch roles between instructor and learner.
- The system shall enable users to create and edit their profiles with personal and academic details.
- The system shall allow users to upload a profile picture.
- The system shall display user ratings and reviews on their profile.
- The system shall allow instructors to create courses with titles, descriptions, and categories.
- The system shall allow instructors to set lesson availability (date, time, and duration).

- The system shall allow learners to search and filter lessons by subject, instructor, or price.
- The system shall enable learners to book lessons from available slots.
- The system shall allow instructors to approve or reject lesson bookings.
- The system shall provide a calendar view for instructors to manage their schedule.
- The system shall allow instructors to cancel scheduled lessons with a valid reason.
- The system shall send notifications to learners if a lesson is canceled.
- The system shall allow learners to cancel their lesson bookings before a specified deadline.
- The system shall send automatic reminders for upcoming lessons to both instructors and learners.
- The system shall support secure online payments for paid lessons.
- The system shall allow instructors to set lesson prices.
- The system shall process refunds in case of canceled lessons according to the refund policy.
- The system shall provide a transaction history for both learners and instructors.
- The system shall ensure that payments are securely stored and processed.
- The system shall allow users to send and receive messages within the platform.
- The system shall support real-time chat between instructors and learners.
- The system shall allow learners to ask instructors questions before booking a lesson.
- The system shall send notifications for new messages.
- The system shall restrict messaging for unverified users.
- The system shall allow learners to rate and review instructors after a lesson.
- The system shall display average ratings on instructor profiles.
- The system shall allow instructors to respond to reviews.
- The system shall prevent users from editing or deleting their reviews after submission.
- The system shall allow admins to remove inappropriate reviews.
- The system shall send lesson confirmation notifications to learners and instructors.
- The system shall send reminders for upcoming lessons via email and app notifications.
- The system shall notify users about new messages.
- The system shall send alerts for failed payments or transaction issues.
- The system shall notify instructors when a learner books a lesson.

- The system shall allow administrators to manage user accounts (suspend, delete, verify).
- The system shall provide a dashboard for tracking platform activity.
- The system shall allow administrators to review reported users.
- The system shall enable administrators to monitor transactions for security purposes.
- The system shall allow administrators to enforce platform rules by issuing warnings or bans.

2.1 Description of the system functionalities

2.1.1 Authentication Page

This page consists of two distinct text areas that accept nickname and password inputs from the user, two different buttons called login and signup, also two different clickable text for "forgot my account" and "forgot my password" functionalities. When one of the "forgot my account" or "forgot my password" texts are clicked, the authentication page will be redirected to a related page in the context. When the login button is clicked, the page will be redirected to the main page if the nickname and password are matched with the database. If the nickname and password are not matched then an information message will be shown as text on the authentication page. When the signup button is clicked, the page will redirect to an updated version of the authentication page that also includes a text area for collecting email address input.

2.1.1.1 Login

User identities consist of three parts; email address, nickname, and password. It's enough to input a nickname and password for login, entering an email address for login is not accepted. Login will be handled on the authentication page. The login process will begin when the user submits his/her nickname and password by clicking the login button. When the login button is clicked, the page will be redirected to the main page if the nickname and password are matched with the database. Technically in the database, the hash and added "password salt" (for extra security), will be checked. If the nickname and password are not matched, then an information message will be shown as text on the authentication page.

2.1.1.2 Sign Up

As mentioned in the Login part, user identities consist of three parts; email address, nickname, and password. All three parts are needed for signup. When the signup button is clicked, the page will redirect to an updated version of the authentication page that also includes a text area for collecting email address input. When all information is entered and clicked the submit button; the email address will be checked whether it satisfies the email address rules or not. Technically, using regular expressions, the email address will be checked. Nicknames will be searched through the entire database, looking for a match or not. If the nickname is already in the database, the system will not permit to the creation of another account with the same nickname. Also, passwords will be checked whether they fit password rules or not by regular expressions. At the end of the process, if sign-up is successful, an information message will be sent to the user's email address

2.1.1.3 Forget My Account

When a user forgets the nickname, he/she cannot log in because of a lack of information. The page consists of one text area that waits for email address input. The email address will be checked whether matched an account in the database or not. If it's matched, then related nickname information will be sent to the email address. Otherwise, an error message will be 8 shown about there being no account found related to the given email address as a text on the page.

2.1.1.4 Forget My Password

When a user forgets the password, he/she cannot log in because of a lack of information. The page consists of one text area that waits for email address input. The email address will be checked whether matched an account in the database or not. If it's matched, a link will be sent to the email address. The link will expire when its timestamp is finished. Adding a timestamp to the link will protect accounts from bad-minded people

2.1.2 Main Page

The main page is designed to provide users with a comprehensive overview of the platform's offerings and impact. The top navigation bar allows users to easily access different sections of the platform, such as the lesson search page, profile settings, and course creation options. It also features quick links to manage bookings and access personalized lesson recommendations.

At the center of the main page are dynamic tables showcasing the top daily, weekly, monthly, and special courses offered by instructors. Each table highlights essential information, including the course title, description, instructor credentials, available lesson slots, and pricing, enabling users to make informed decisions about which courses to enroll in. This leaderboard setup encourages active participation among instructors to enhance their offerings and stand out in the rankings.

Users have the flexibility to book lessons directly or explore recommended lessons based on their academic needs and preferences. Advanced filtering options allow learners to search by subject, instructor, or price, making the process of finding the ideal lesson straightforward and personalized.

The "Book a Lesson" button on the main page simplifies the process, ensuring users can quickly schedule sessions with their chosen instructor. For those unsure of their exact needs, the platform offers tools to refine their options based on academic goals or feedback from peers. This seamless design enhances user experience and fosters engagement among the student community.

Lastly, the main page features live updates on the number of lessons booked and the ratings achieved by top instructors. This real-time information promotes transparency and inspires trust in the platform while instilling a sense of progress and community achievement. Users can see the tangible benefits of their engagements and the collective efforts of the platform, fostering a sense of collaboration and academic growth.

2.1.2.1 Navigation Bar

The navigation bar consists of essential buttons that provide quick access to various sections of the platform. These buttons include "Search Lessons," "Create a Lesson," "Profile

Settings," "Messages," and "Log In/Sign Up." Its primary function is to simplify navigation and help users easily locate the features they need.

2.1.2.2 Top Courses Selection

The top courses section organizes lessons into categories based on the specific academic needs of learners. These categories include Exam-Focused Lessons, Topic-Focused Lessons, and Assignment-Focused Lessons. Each category offers tailored options to address students' unique goals and challenges.

2.1.2.3 Exam-focuses Lessons

This section is dedicated to lessons aimed at helping students prepare for their upcoming exams. These sessions focus on critical topics, problem-solving strategies, and exam techniques to improve performance. Lessons in this category are ideal for students seeking intensive support to ace their exams and are often booked close to exam dates for maximum effectiveness

2.1.2.4 Topic-focused Lessons

Topic-focused lessons are designed for students who wish to deepen their understanding of specific subjects or concepts. Whether it's mastering a challenging topic in mathematics or clarifying difficult theories in physics, this category caters to students looking for in-depth exploration and clarification in targeted areas.

2.1.2.5 Assignment-focused Lessons

This section is tailored to students needing guidance on completing homework, projects, or assignments. Instructors assist learners by offering step-by-step explanations, problem-solving methods, and project planning strategies to ensure assignments are completed effectively and accurately. These sessions are particularly helpful for students managing tight deadlines or struggling with specific tasks.

2.1.3 Lesson Search Page

The Lesson Search Page allows users to discover available lessons based on subject areas, instructor profiles, user ratings, and pricing. The search interface includes a prominent filter panel where users can specify criteria such as subject, date and time availability, lesson format (online/offline), price range, and instructor rating.

The search results dynamically update based on the selected filters, offering a personalized and efficient experience. Each result card displays the lesson title, instructor name, a short description, average rating, available time slots, and a “Book Now” button for direct booking.

To enhance discoverability, the system also integrates recommendation algorithms that suggest lessons based on the user’s profile, past bookings, and liked courses. Hover effects and bookmark buttons allow users to save lessons for later.

2.1.4 Course Creation Page

The Course Creation Page is dedicated to instructors who want to create and publish their own lessons. To begin, instructors choose a lesson category (e.g., math, language, coding) from a predefined list. They are then prompted to fill out key information, including the course title, description, duration, target audience, and pricing.

Instructors can upload an optional cover image and list the available times for bookings through a calendar interface. The platform supports both single-session and recurring lesson options. Instructors are required to agree to platform guidelines and ethical standards before publishing their course.

Upon submission, the course enters a pending review state and becomes visible to users once approved. Instructors can track enrollments, respond to student feedback, and edit course details via their dashboard.

2.1.5 Profile Page

The Profile Page serves as the personal hub for both learners and instructors. For learners, it displays enrolled lessons, upcoming bookings, completed lessons, feedback history, and

personalized lesson suggestions. For instructors, it shows created courses, upcoming and past sessions, student reviews, and earnings.

Users can update personal details such as name, email, profile photo, and preferred learning topics. Privacy settings allow users to manage visibility and communication preferences. The dashboard includes quick access to booking history and payment information.

2.1.6 Booking Page

The Bookings Page provides a centralized space for managing upcoming and past lesson reservations. Users can view their scheduled sessions, join live lessons, reschedule or cancel bookings (if within allowed time frames), and submit feedback post-lesson.

Each booking entry includes the lesson title, instructor name, date and time, status (confirmed, pending, completed), and links to contact the instructor or join the session. Notifications and reminders are integrated to ensure timely attendance.

2.1.7 Instructor Dashboard

The Instructor Dashboard is tailored for lesson creators to manage their courses efficiently. It includes real-time data on enrollment numbers, revenue, student ratings, and feedback trends. Instructors can edit course details, add new time slots, and communicate with enrolled students through the dashboard.

A performance summary panel highlights best-performing courses, average ratings, and suggestions for improvement. The dashboard also provides access to financial statements and withdrawal options.

2.1.8 Notification System

The Notification System ensures that users are kept informed about important updates related to their learning journey. These notifications include upcoming lesson reminders, new messages, instructor announcements, course updates, and promotional offers. Notifications are accessible through the website interface and may also be sent via email or push notifications for mobile users.

Users can manage their notification preferences through the profile settings, enabling or disabling different types of alerts. This helps maintain a balance between staying informed and avoiding notification fatigue.

2.1.9 Feedback & Rating System

The Feedback & Rating System allows learners to evaluate instructors after completing a lesson. Ratings are based on multiple criteria such as clarity, engagement, knowledge, and punctuality. Learners can also leave written reviews to provide more detailed feedback.

Instructors receive aggregated feedback in their dashboards and can respond to reviews when appropriate. Highly-rated instructors gain visibility in the platform's search and recommendation systems, incentivizing quality teaching.

2.1.10 Messaging System

The Messaging System facilitates direct communication between learners and instructors. Users can discuss lesson content, scheduling preferences, and other relevant details. All conversations are stored securely within the platform and accessible via the "Messages" section on the navigation bar.

The system includes features like read receipts, mute options, and reporting tools to ensure respectful and productive communication. Spam filters and moderation tools are in place to maintain a safe environment.

2.2 Description of the system users

2.2.1 Learners

Learners are users who wish to receive private lessons to enhance their understanding of specific subjects or prepare for exams. They can search for available lessons, filter results based on subject, instructor ratings, and pricing, and book sessions with their desired instructors. Learners have the ability to create an account on the platform, log in, reset their passwords, and edit personal or academic details such as their field of study and academic level. They can also leave ratings and reviews for the lessons they have attended to provide feedback for instructors and other learners.

2.2.2 Instructors

Instructors are users who provide lessons to learners, sharing their expertise in specific subjects. They can create personalized course listings with titles, descriptions, lesson categories, and pricing details. Instructors have the ability to specify their availability by setting lesson schedules, manage their booked sessions through a calendar view, and accept or reject lesson booking requests. Additionally, they can edit their profiles to include academic and professional credentials and upload profile pictures to build trust and credibility. Instructors also receive ratings and reviews from learners, which are displayed on their profiles to showcase their teaching success.

2.2.3 Administrators (Platform Managers)

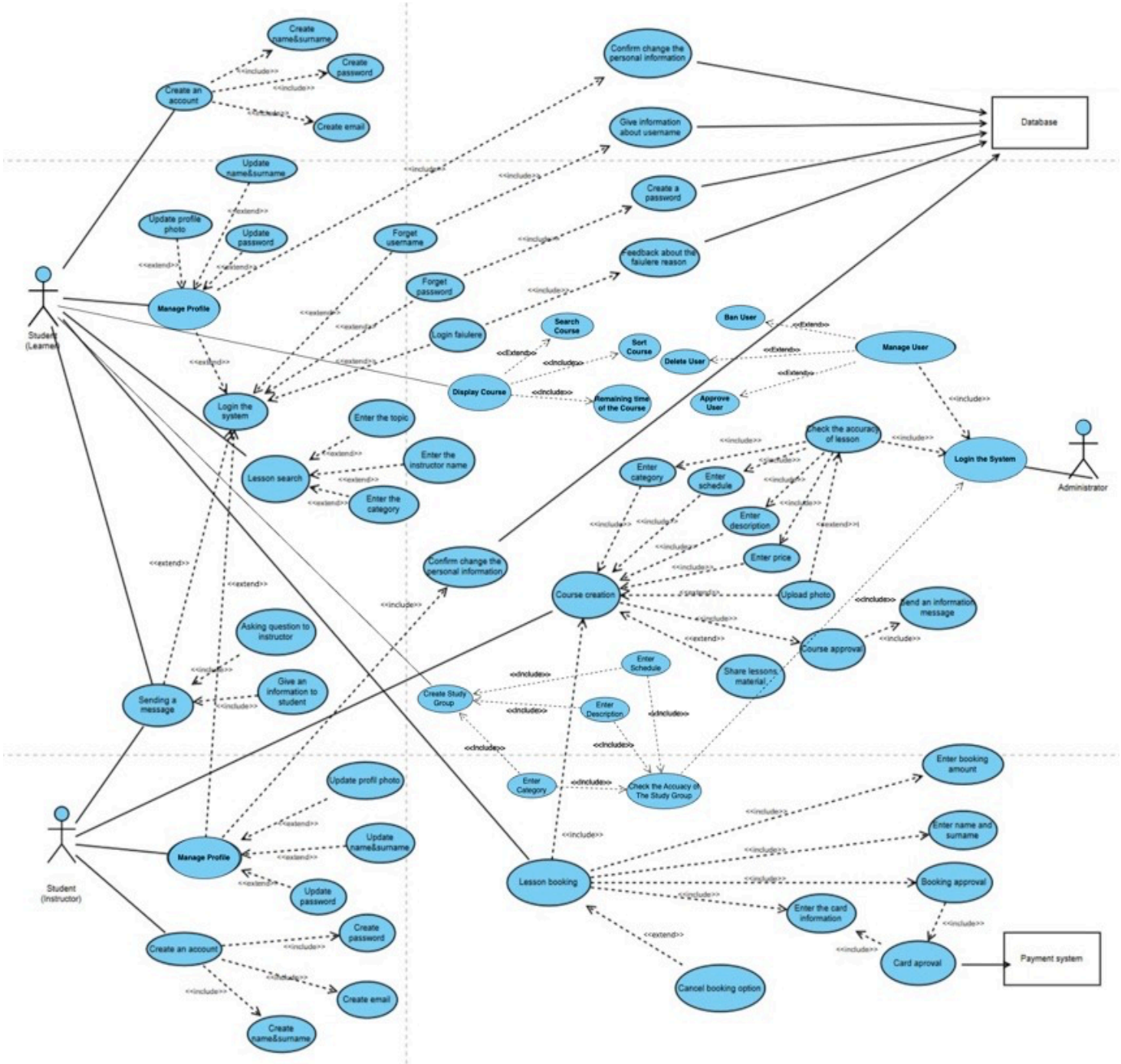
Administrators oversee the operations of the platform and ensure it functions efficiently. They have the authority to verify, suspend, or delete user accounts, approve instructor-created course listings, and manage reported users or inappropriate content such as harmful reviews. Administrators can also monitor platform analytics, including total sessions booked, ratings, and feedback, to make improvements. Their responsibilities include resolving technical issues faced by users and ensuring that the platform complies with data security and privacy standards

2.3 Specific Requirements

- The system shall allow users to register using a university email, nickname, and password.
- The system shall reject duplicate nicknames during the registration process.
- The system shall hash and salt all passwords before storing them in the database.
- The system shall allow users to reset their password via a time-limited email link.
- The system shall send the user's nickname to their email when the “Forgot My Account” function is used.
- The system shall validate the email address format using regular expressions during sign-up.
- The system shall display a proper error message if login credentials are incorrect.
- The system shall redirect users to the main page after successful login.

- The system shall allow learners to search for lessons using filters such as subject, price, rating, and time.
- The system shall display search results dynamically based on user-selected filters.
- The system shall allow instructors to create and publish new courses with a title, description, category, and pricing.
- The system shall require instructors to set available time slots before publishing a course.
- The system shall notify the instructor when a new booking is made.
- The system shall allow learners to book available lessons directly from the search results.
- The system shall allow both learners and instructors to message each other after a booking is made.
- The system shall allow users to rate and review a completed lesson.
- The system shall update instructor ratings based on learner feedback.
- The system shall display real-time notifications for messages, bookings, and system updates.
- The system shall provide an admin interface for managing users, courses, and reports.
- The system shall allow file or note sharing within forum posts.

2.3.1 Use Case Diagram



2.3.2 Use Case Priority List

Use Case ID	Use Case	Priority Rank	Reason
UC01	User Registration	5	Fundamental for account creation and platform access.
UC02	User Login	5	Essential for secure authentication and user sessions.
UC03	Forget My Password	5	Critical for regain account access and security.
UC04	Lesson Search	5	Allows learners to find suitable lessons.
UC05	Lesson Booking	5	Enables the core function of reserving sessions.
UC06	Course Creation	5	Main activity for instructors to offer lessons.
UC07	Create Study Group	4	Promotes collaborative learning and peer support.
UC08	Share Learning Resource	4	Supports free exchange of knowledge and notes.
UC09	Profile Management	4	Allows users to personalize and manage visibility.
UC10	Notification System	5	Keeps users informed of relevant activities.
UC11	Courses Display	3	Improves engagement, not core to basic usage.
UC12	Filter & Recommendation Engine	3	Enhances search experience.
UC13	Booking Management	5	Centralized handling of scheduled sessions.
UC14	Messaging System	5	Facilitates communication and lesson coordination.
UC15	Feedback & Rating	4	Supports transparency and continuous improvement.
UC16	Payment System	5	Manages financial transactions securely.
UC17	Admin User Management	4	Supports moderation and platform security.

2.3.3 Use Case Specifications

Use case ID: UC01	**Use Case:** User Registration
Primary Actors: Student	
MAIN FLOW	
Preconditions: User is not registered	
Main Flow: 1. User opens registration form. 2. Enters nickname, password, and email. 3. System checks input validity. 4. Confirmation email is sent. 5. User verifies email. 6. Account is created.	
Post conditions: User is redirected to login page	
ALTERNATIVE FLOW	
Preconditions: User enters invalid or duplicate info.	
Alternative Flow: System shows error for invalid email or existing nickname.	
Post conditions: User is redirected to login page.	

Use case ID: UC02	**Use Case:** User Login
Primary Actors: Student	
MAIN FLOW	
Preconditions: User is already registered	
Main Flow: 1. User opens login page. 2. Enters nickname and password. 3. System checks credentials. 4. If correct, redirects to main page. 5. If incorrect, shows error.	
Post conditions: User is logged into the system.	
ALTERNATIVE FLOW	
Preconditions: User enters wrong credentials.	
Alternative Flow: System shows incorrect login error message.	
Post conditions: User is logged into the system.	

Use case ID: UC03	**Use Case:** Password Reset
Primary Actors: Student	
MAIN FLOW	
Preconditions: User forgets password	
Main Flow: 1. User clicks 'Forgot Password'. 2. Enters email address. 3. System checks if email exists. 4. If valid, sends reset link. 5. User sets new password via link.	
Post conditions: User reset and successfully updates password.	
ALTERNATIVE FLOW	
Preconditions: Email not found in database.	
Alternative Flow: System displays email-not-found error.	
Post conditions: User password is reset.	

Use case ID: UC04	**Use Case:** Lesson Search
Primary Actors: Learner	
MAIN FLOW	
Preconditions: User is successfully logged in	
Main Flow: 1. User goes to search page. 2. Applies filters (subject, rating, price, etc.). 3. System displays matching lessons. 4. User browses and selects a lesson.	
Post conditions: List of filtered lessons shown.	
ALTERNATIVE FLOW	
Preconditions: No lessons match criteria.	
Alternative Flow: System shows 'no matching results' message.	
Post conditions: List of filtered lessons shown.	

Use case ID: UC05	**Use Case:** Lesson Booking

Primary Actors: Learner
MAIN FLOW
Preconditions: User selects a lesson
Main Flow: 1. User clicks 'Book Now'. 2. Chooses a time slot. 3. Proceeds with booking. 4. Receives confirmation message. 5. Instructor is notified.
Post conditions: Booking is saved to calendar.
ALTERNATIVE FLOW
Preconditions Selected slot is no longer available.
Alternative Flow: System prompts to choose another time or notifies unavailability.
Post conditions: Booking is saved to calendar.

Use case ID: UC06	**Use Case:** Course Creation
Primary Actors: Instructor	
MAIN FLOW	
Preconditions: Instructor is logged in	
Main Flow: 1. Instructor navigates to course creation. 2. Fills out title, description, category, time slots. 3. Sets pricing and uploads optional image. 4. Agrees to terms. 5. Submits for review.	
Post conditions: Course is published after approval.	
ALTERNATIVE FLOW	
Preconditions: Missing/invalid course data.	
Alternative Flow: System displays error and prevents submission.	
Post conditions: Course is published after approval.	

Use case ID: UC07	**Use Case:** Create Study Group

Primary Actors: Learner, Instructor
MAIN FLOW
Preconditions: User is logged in
Main Flow: 1. User navigates to 'Study Groups' section. 2. Clicks 'Create New Group'. 3. Fills in group name, subject, and visibility settings. 4. Clicks 'Create'. 5. Group is added to group list.
Post conditions: Study group created successfully and visible to others.
ALTERNATIVE FLOW
Preconditions: User is not logged in.
Alternative Flow: System redirects to login page.
Post conditions: Study group created successfully and visible to others.

Use case ID: UC08	**Use Case:** Share Learning Resource
Primary Actors: Learner, Instructor	
MAIN FLOW	
Preconditions: User is logged in	
Main Flow: 1. User navigates to a forum thread. 2. Clicks 'Attach Resource'. 3. Selects file or adds link to resource. 4. Adds description (optional). 5. Submits post.	
Post conditions: Resource is uploaded and visible within the forum.	
ALTERNATIVE FLOW	
Preconditions: Unsupported file type uploaded.	
Alternative Flow: System shows error message and prevents upload.	
Post conditions: Resource is uploaded and visible within the forum.	

Use case ID: UC09	**Use Case:** Profile Management

Primary Actors: User
MAIN FLOW
Preconditions: User is logged in
Main Flow: 1. User navigates to profile settings. 2. Updates personal info and preferences. 3. Saves changes.
Post conditions: Profile is updated successfully.
ALTERNATIVE FLOW
Preconditions: Incomplete or invalid inputs.
Alternative Flow: System highlights fields and shows validation messages.
Post conditions: Profile is updated successfully.

Use case ID: UC10	**Use Case:** Notification System
Primary Actors: User	
MAIN FLOW	
Preconditions: Relevant event occurs	
Main Flow: 1. System detects an event (e.g., booking, message). 2. Creates and delivers notification. 3. User sees notification via UI or email.	
Post conditions: User stays updated with platform events.	
ALTERNATIVE FLOW	
Preconditions: User disabled notifications.	
Alternative Flow: System stores notification in notification center for later viewing.	
Post conditions: User stays updated with platform events.	

Use case ID: UC11	**Use Case:** Courses Display
Primary Actors: User	
MAIN FLOW	
Preconditions: User opens main page	

Main Flow: 1. System ranks courses by popularity. 2. Displays top courses with instructor info and slots.
Post conditions: Top courses are shown to user.
ALTERNATIVE FLOW
Preconditions: No courses meet ranking criteria.
Alternative Flow: System displays message or generic course listing.
Post conditions: Top courses are shown to user.

Use case ID: UC12	**Use Case:** Filter & Recommendation Engine
Primary Actors: Learner	
MAIN FLOW	
Preconditions: User has booking history or preferences	
Main Flow: 1. System analyzes user activity. 2. Generates recommended course list. 3. Displays on main or search page.	
Post conditions: Recommendations help user discover courses	
ALTERNATIVE FLOW	
Preconditions: No relevant history.	
Alternative Flow: System shows popular or trending courses instead.	
Post conditions: Recommendations help user discover courses.	

Use case ID: UC13	**Use Case:** Booking Management
Primary Actors: Learner	
MAIN FLOW	
Preconditions: User has upcoming or past bookings	
Main Flow: 1. User opens booking page. 2. Views, edits or cancels lessons. 3. Joins live lessons if enabled.	
Post conditions: User manages their bookings easily.	

ALTERNATIVE FLOW
Preconditions: Booking too close to start time.
Alternative Flow: System disallows change or warns about refund policy.
Post conditions: User manages their bookings easily.

Use case ID: UC14	**Use Case:** Messaging System
Primary Actors: Learner, Instructor	
MAIN FLOW	
Preconditions: Users share a booking	
Main Flow: 1. User accesses Messages tab. 2. Chooses recipient from contact list. 3. Sends and receives messages.	
Post conditions: Communication is established.	
ALTERNATIVE FLOW	
Preconditions: User not enrolled or blocked.	
Alternative Flow: System blocks access to chat and displays notice.	
Post conditions: Communication is established.	

Use case ID: UC15	**Use Case:** Feedback & Rating
Primary Actors: Learner	
MAIN FLOW	
Preconditions: Lesson is completed	
Main Flow: 1. Learner accesses feedback form. 2. Rates instructor and adds optional comments. 3. Submits review.	
Post conditions: Review saved and visible on profile.	
ALTERNATIVE FLOW	
Preconditions: User skips feedback.	
Alternative Flow: System sends reminder or marks as no review.	

Post conditions: Review saved and visible on profile.

Use case ID: UC16	**Use Case:** Payment System
Primary Actors: Learner, Instructor	
MAIN FLOW	
Preconditions: Learner books paid lesson	
Main Flow: 1. User initiates payment. 2. Chooses payment method. 3. Completes transaction. 4. Funds held in escrow.	
Post conditions: Payment successful and secured.	
ALTERNATIVE FLOW	
Preconditions: Transaction fails.	
Alternative Flow: System requests retry or alternate method.	
Post conditions: Payment successful and secured.	

Use case ID: UC17	**Use Case:** Admin User Management
Primary Actors: Admin	
MAIN FLOW	
Preconditions: Admin accesses dashboard	
Main Flow: 1. Admin views reported users and flagged content. 2. Takes action: warn, suspend, delete. 3. Updates reflected in user database.	
Post conditions: Platform integrity maintained.	
ALTERNATIVE FLOW	
Preconditions: Admin takes no action.	
Alternative Flow: No change occurs; flagged user remains active	
Post conditions: Platform integrity maintained.	

3 Non-Functional Requirements

- The system shall respond to user requests within 2 seconds under normal load conditions. (NFR01)
- The system shall support up to 1,000 concurrent users without performance degradation. (NFR02)
- The system shall implement account protection via 2FA and login attempt limits. (NFR04)
- The system shall maintain 99.9% uptime and restore service within 5 minutes after a failure. (NFR06)
- The system shall back up all user data every 24 hours and ensure data integrity on recovery. (NFR03)
- The system shall provide a responsive and intuitive interface across devices. (NFR07)
- The system shall support accessibility standards including screen readers and keyboard navigation. (NFR14)
- The system shall be usable on Firefox, Opera, Chrome, and Safari. (NFR09)
- The system shall scale to support 100,000 users and handle 10x peak traffic spikes. (NFR15)
- The system shall use modular architecture and allow zero-downtime updates. (NFR16)
- The system shall comply with GDPR and allow users to delete their data. (NFR17)
- The system shall store user data in an encrypted format and support data export. (NFR13)
- The system shall log errors and retry failed operations before notifying the user. (NFR18)
- The system shall provide real-time performance monitoring and admin usage reports. (NFR19)
- The system shall allow users to reset their passwords securely via time-limited email links. (NFR12)
- The system shall provide a real-time notification for bookings, messages and schedule changes. (NFR11)
- The system shall provide weekly performance reports to administrators for ongoing improvement. (NFR10)
- The system shall automatically log out users after 30 minutes of inactivity for security. (NFR05)

- The system shall allow students to access the web site from multiple devices with synchronized data. (NFR08)

3.1 Volere Template

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR01	NFR (Performance)	UC04-UC05-UC14
Description: The system shall respond to user actions within 2 seconds under normal load conditions.		
Rationale: The system should be responsive, and have minimal delay or error. It should be able to manage a lot of traffic, especially during busy donation seasons.		
Fit Criteria: • The website shall load within 3 seconds of the user requesting the website. • The load time shall be measured from the user's request to the website's full display on the user's device. • The load time shall be consistent across multiple tests performed under different network conditions. • The website's load time shall be monitored and optimized regularly to ensure compliance with the requiremen		
Priority: 4 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR02	NFR (Scalability)	UC04, UC05, UC16
Description: The system should be capable enough to handle 1000 users at the same time.		
Rationale: To ensure the system remains stable during high traffic usage.		
Fit Criteria: • The system shall be able to handle up to 1000 concurrent users accessing the website. • The response time for each user's request shall not exceed 5 seconds under maximum load. • The system shall maintain data integrity and prevent data loss under heavy load. • The system shall be tested using load testing tools to ensure that it meets the requirement for concurrent users. • The system shall be designed with scalability in mind to handle future growth in user traffic.		

• The system shall be monitored regularly to ensure that it is able to handle increased traffic and demand over time.
Priority: 4 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR09	NFR (Usability)	UC01
Description: The system shall be usable on Firefox, Opera, Chrome, and Safari.		
Rationale: The system shall be usable on Firefox, Opera, Chrome, and Safari web browsers. The design should be intuitive and easy to navigate, catering to users who are at least 7 years old with basic knowledge of web browsers.		
Fit Criteria: • The system shall be tested on the latest version of Firefox, Opera, Chrome, and Safari web browsers. • The website shall be able to load in less than 3 seconds on a desktop device. • The website shall be able to load in less than 5 seconds on a mobile device with a 3G connection.		
Priority: 4 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR03	NFR (Reliability)	UC01, UC10
Description: The system shall back up all user data every 24 hours and ensure data integrity on recovery.		
Rationale: To protect user data from loss and corruption.		
Fit Criteria: • Automated daily backups shall be configured. • Recovery tests shall verify data completeness and accuracy.		
Priority: 5 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR06	NFR (Reliability)	UC01
Description: The system shall maintain 99.9% uptime and restore service within 5 minutes after a failure.		
Rationale: To ensure system availability for users at all times.		
Fit Criteria: • Downtime shall not exceed 8 hours annually. • Monitoring system will log uptime metrics. • Recovery scripts shall complete in under 5 minutes.		
Priority: 5 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR07	NFR (Usability)	UC01
Description: The system shall provide a responsive and intuitive interface across devices**.**		
Rationale: To ensure ease of use on all platforms and screen sizes.		
Fit Criteria: • The UI shall adapt to mobile, tablet, and desktop resolutions. • Usability testing shall confirm intuitive navigation.		
Priority: 4 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR08	NFR (Performance)	UC10
Description: The system shall allow students to access the web site from multiple devices with synchronized data.		
Rationale: Students often switch between devices (e.g., laptop, tablet, phone) during their study sessions. To ensure a seamless user experience, the platform must provide real-time synchronization of their data, including profiles, bookings, and messages.		

Fit Criteria:

- A change, such as booking a lesson or editing a profile, shall be reflected across all logged-in devices within 5 seconds.
- Data synchronization shall be verified through testing on at least three device types: desktop, tablet, and mobile.
- Users shall remain logged in on all devices unless they explicitly choose to log out from all sessions.
- Real-time synchronization events shall be logged and monitored during quality assurance (QA) testing.

Priority: 3 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR05	NFR (Security)	UC02

Description: The system shall automatically log out users after 30 minutes of inactivity for security**.**

Rationale: To prevent unauthorized access and protect sensitive user data when users leave their sessions unattended, an automatic logout mechanism is required.

Fit Criteria:

- The system shall track user activity such as mouse movement, clicks, and keyboard input.
- If no activity is detected for 30 minutes, the system shall terminate the session and redirect the user to the login page.
- The system shall display a warning message 1 minute before automatic logout.
- This functionality shall be tested across different roles (student, instructor, admin) and devices.

Priority: 4 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR10	NFR (Monitoring & Reporting)	UC10

Description: The system shall provide weekly performance reports to administrators for ongoing improvement.

Rationale: Regular performance insights allow administrators to monitor system usage, detect bottlenecks, and implement necessary improvements. This supports proactive management and strategic decision-making.

Fit Criteria:

- The system shall generate and deliver performance reports to designated admin emails every 7 days.

• Reports shall include metrics such as user activity, booking frequency, most viewed lessons, and system response times. • Reports shall be delivered in PDF or dashboard summary format. • The system shall maintain a report history accessible for the last 3 months.
Priority: 2 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type:**	**Event/Use case #:**
NFR11	NFR (Usability / Real-time Communication)	UC05, UC11, UC16
Description: The system shall provide a real-time notifications for bookings, messages and schedule changes.		
Rationale: Users need to be instantly informed about updates that affect their interactions (e.g., new messages, lesson confirmations, or schedule alterations) to take timely action and avoid missed sessions or miscommunication.		
Fit Criteria: • The system shall send real-time notifications via the user interface and optionally via email or push notification. • Notifications shall be triggered within 2 seconds of an event (e.g., booking confirmation, message received). • Users shall be able to manage which types of notifications they receive through profile settings. • Notification delivery shall be tested across web and mobile interfaces for all user roles.		
Priority: 5 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type:**	**Event/Use case #:**
NFR12	NFR (Security / Account Recovery)	UC03
Description: The system shall allow users to reset their passwords securely via time-limited email links.		
Rationale: To ensure account recovery without compromising user data, the password reset process must be secure, authenticated, and time-bound.		
Fit Criteria: • The system shall send a password reset link to the user's registered email upon request. • The reset link shall expire within 15 minutes of being generated. • The link shall be single-use and become invalid after a successful password change or timeout.		

• The system shall not expose any user data during the reset process. • Penetration tests shall confirm that the password reset flow is secure.
Priority: 5 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR13	NFR (Data Management & Privacy)	UC01, UC10
Description: The system shall store user data in an encrypted format and support data export.		
Rationale: To protect personal information and comply with privacy regulations, user data must be stored securely and users should have the ability to export their information when requested.		
Fit Criteria: • All user data (e.g., profile details, bookings, messages) shall be stored using industry-standard encryption (e.g., AES-256). • Users shall be able to request and download their personal data in a readable format (e.g., JSON or CSV) via the profile settings page. • Exported data shall include only the requesting user's content and be downloadable within 24 hours of the request. • Encryption methods and export functionality shall be validated during security audits and QA testing.		
Priority: 5 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR04	NFR (Security)	UC02
Description: The system shall implement account protection via two-factor authentication (2FA) and login attempt limits.		
Rationale: To reduce the risk of unauthorized access and brute-force attacks, user accounts must be protected by multi-step verification and safeguards against repeated login failures.		
Fit Criteria: • Users shall be required to complete 2FA using email or mobile verification when logging in from a new device or location. • The system shall lock accounts after 5 consecutive failed login attempts within a 10-minute window. • Locked accounts shall require email verification to unlock or wait for a 15-minute cooldown period.		

• All login attempts and lockout events shall be logged for auditing.
Priority: 5 (Lowest: 1, Highest: 5)

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR14	NFR (Accessibility)	UC01, UC04, UC05, UC09
Description: The system shall support accessibility standards including screen readers and keyboard navigation.		
Rationale: To make the platform usable for users with disabilities and comply with accessibility laws.		
Fit Criteria: • The platform shall pass WCAG 2.1 AA compliance checks to ensure accessibility standards are met. • Screen reader tests shall pass on all interactive elements to support visually impaired users. • Keyboard-only navigation shall be possible across all features for users with motor impairments.		
Priority: 4 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR15	NFR (Scalability)	UC04, UC05, UC16
Description: The system shall scale to support 100,000 users and handle 10x peak traffic spikes.		
Rationale: To prepare the system for future expansion and ensure availability during peak loads.		
Fit Criteria: • The system shall scale up to 100,000 users in a stress test. • 10x average load shall be handled without system failure.		
Priority: 5 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR16	NFR (Maintainability)	UC01
Description: The system shall use modular architecture and allow zero-downtime updates.		
Rationale: To improve maintainability and support continuous delivery and upgrades.		
Fit Criteria: • Updates shall be applied without downtime. • Code shall be structured in independently testable modules.		
Priority: 4 (Lowest: 1, Highest: 5)		

Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR17	NFR (Legal Compliance)	UC17
Description: The system shall comply with GDPR and allow users to delete their data.		
Rationale: To comply with European data protection regulations and user privacy rights.		
Fit Criteria: • Users can request deletion and system completes it within 30 days. • GDPR audit check passed.		
Priority: 5 (Lowest: 1, Highest: 5)		

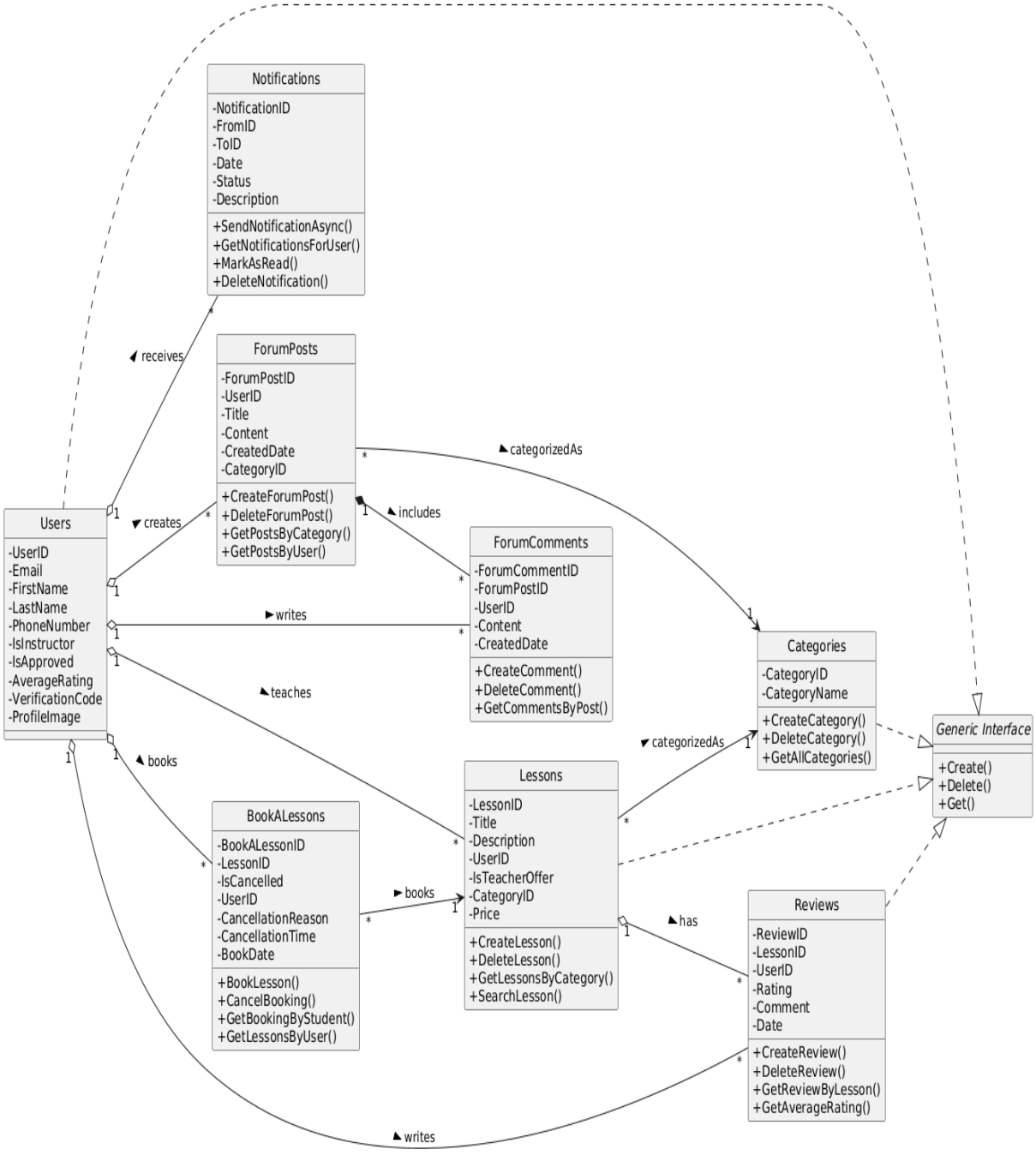
Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR18	NFR (Error Handling)	UC05
Description: The system shall log errors and retry failed operations before notifying the user.		
Rationale: To reduce user frustration and assist in debugging and error recovery.		
Fit Criteria: • 100% of errors shall be logged with timestamps to ensure accurate tracking and debugging.		

• Failed requests shall be retried automatically up to 3 times to enhance reliability and error handling.
Priority: 4 (Lowest: 1, Highest: 5)

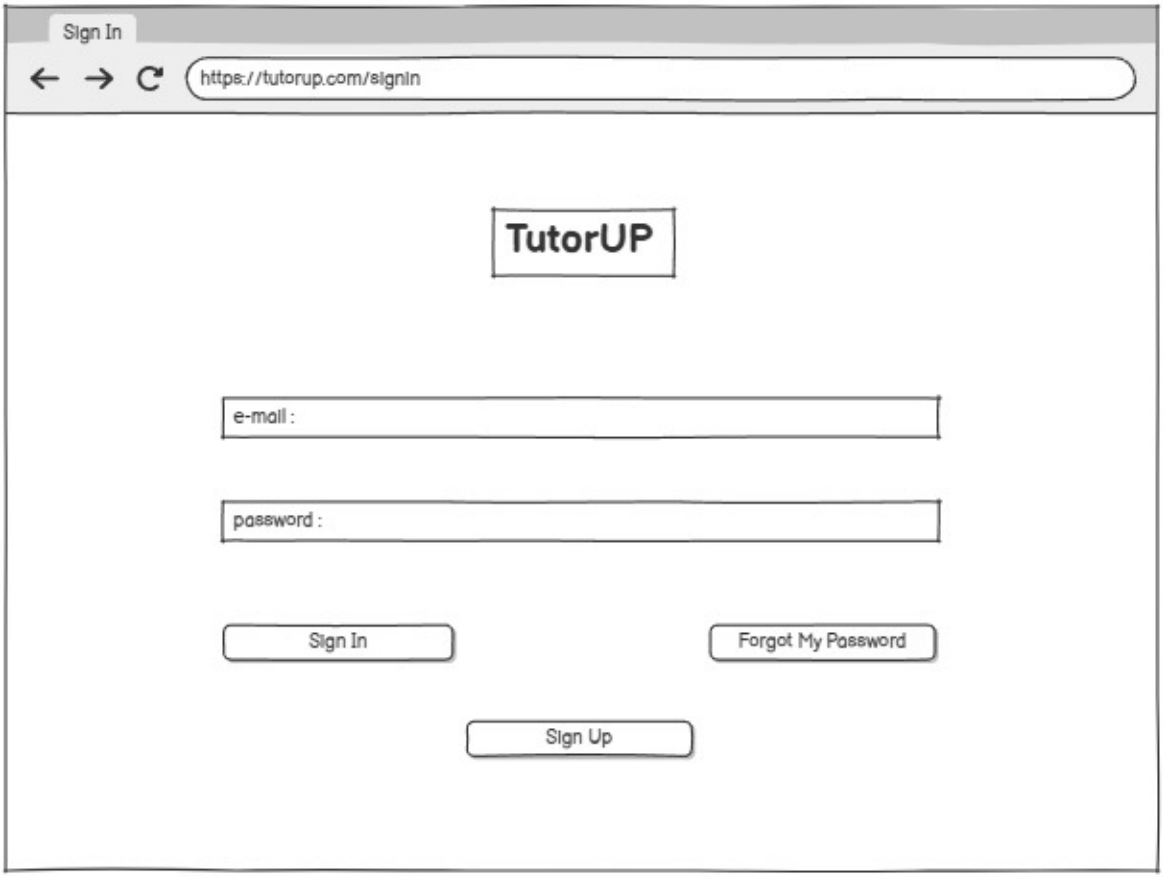
Requirement ID:	**Requirement Type**:	**Event/Use case #:**
NFR19	NFR (Monitoring)	UC17
Description: The system shall provide real-time performance monitoring and admin usage reports.		
Rationale: To ensure system performance is observable and support operational decisions.		
Fit Criteria: • A dashboard shall be available with live server and usage metrics to monitor system performance in real-time. • Monthly usage reports shall be downloadable by administrators for analytical and operational purposes.		
Priority: 3 (Lowest: 1, Highest: 5)		

4 System Models

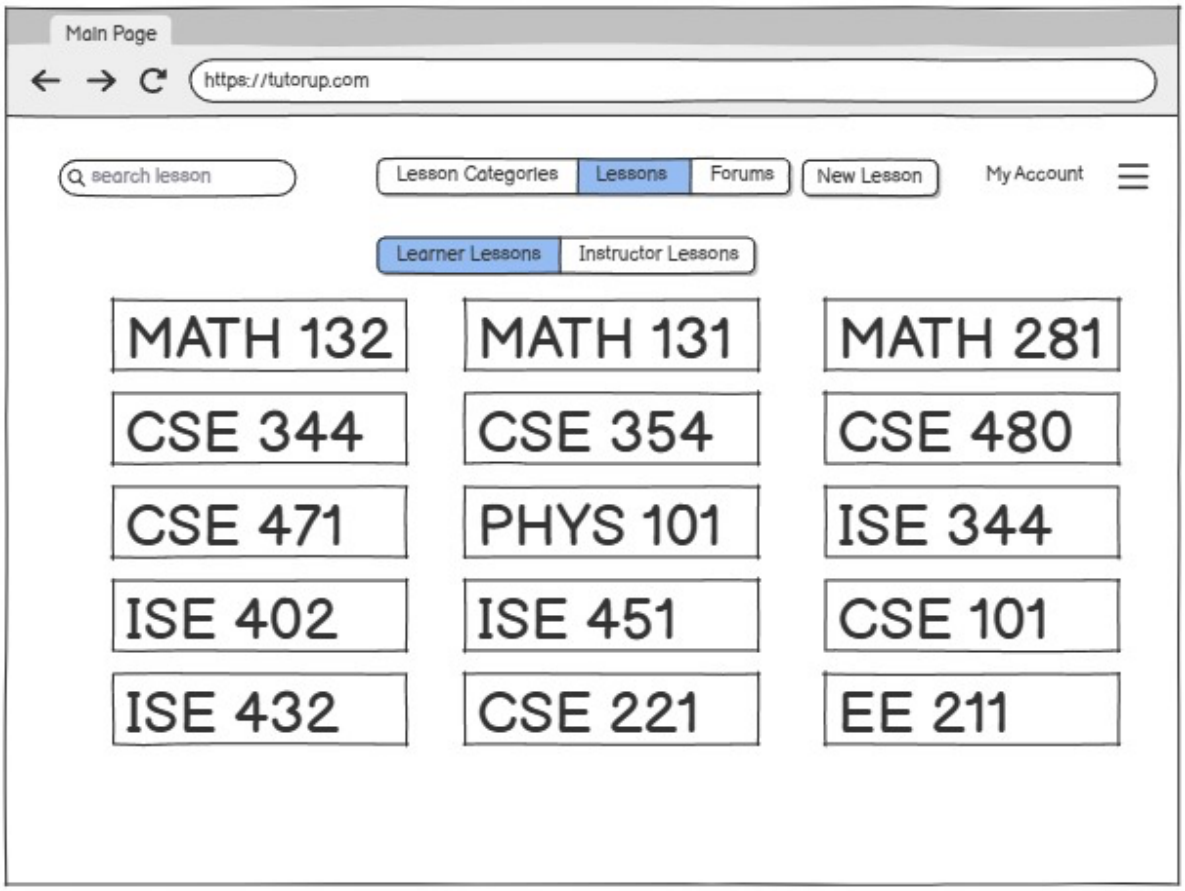
4.1 UML Class Diagram for Domain Analysis

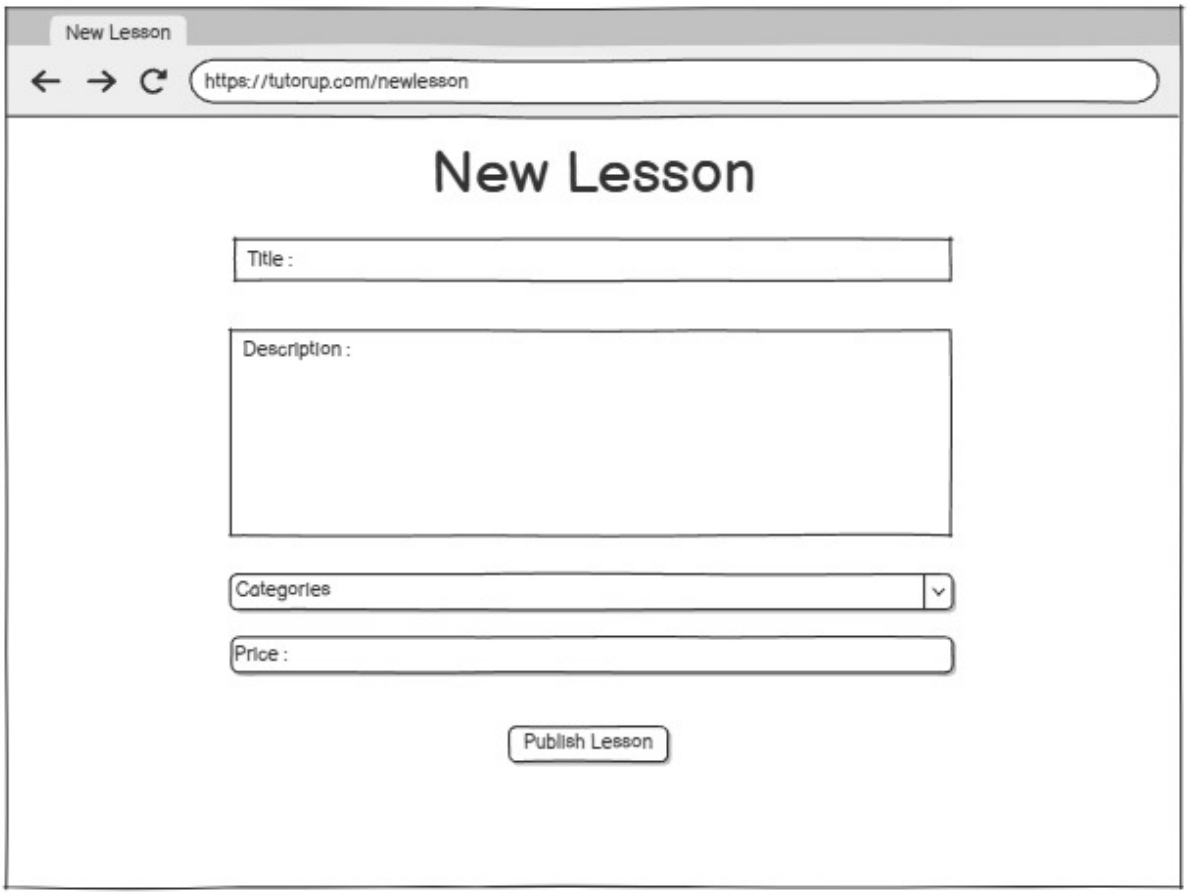


4.2 User Interface - Navigational Paths and Screen Mock-Ups




Sign Up
https://tutorup.com/signup
TutorUP
First Name :
Last Name :
Phone Number :
e-mail :
e-mail again :
password :
Role
Instructor
Learner
Sign Up


Main Page
https://tutorup.com
search lesson
Lesson Categories
Lessons
Forums
New Lesson
My Account
Learner Lessons
Instructor Lessons
MATH 132
MATH 131
MATH 281
CSE 344
CSE 354
CSE 480
CSE 471
PHYS 101
ISE 344
ISE 402
ISE 451
CSE 101
ISE 432
CSE 221
EE 211



5 Customer Experience

As a customer (learner or instructor), TutorUP platform provides a highly personalized and user-centered experience that significantly enhances the process of academic support among university students. Below is a detailed evaluation of the customer experience:

- **Easy Registration and Login:**
  The registration process is intuitive and secure. Features such as password reset and nickname reminders ensure that users can easily access their accounts, even if they forget their credentials. This makes the platform accessible for users with varying technical proficiency.
- **Personalized Learning Environment:**
  The system allows students to choose whether they want to give or receive lessons, and tailor their profiles accordingly. Learners can filter lessons by topic, price, and rating, while instructors can define their own schedules, prices, and expertise areas.
- **Seamless Booking and Scheduling:**
  The lesson booking process is straightforward, with real-time availability and confirmation notifications. Users can easily reschedule or cancel sessions if needed, which supports flexibility and minimizes frustration.

- **Real-Time Notifications and Messaging:**
  Learners and instructors stay informed with instant notifications for bookings, cancellations, and new messages. The internal messaging system enables them to communicate directly and clarify any lesson-related details in advance.
- **Transparency through Ratings and Reviews:**
  The feedback and rating system helps build trust within the platform. Learners can view instructors' performance and comments from past students, making it easier to make informed decisions.
- **Security and Privacy:**
  Personal and academic data are securely stored and encrypted. The platform is compliant with GDPR standards, and privacy settings allow users to manage their visibility and interaction preferences.

In conclusion, TutorUP provides a functional, secure, and learner-oriented experience. With a few interface and accessibility enhancements, the platform has strong potential to become a preferred tool for peer-to-peer learning in academic communities.

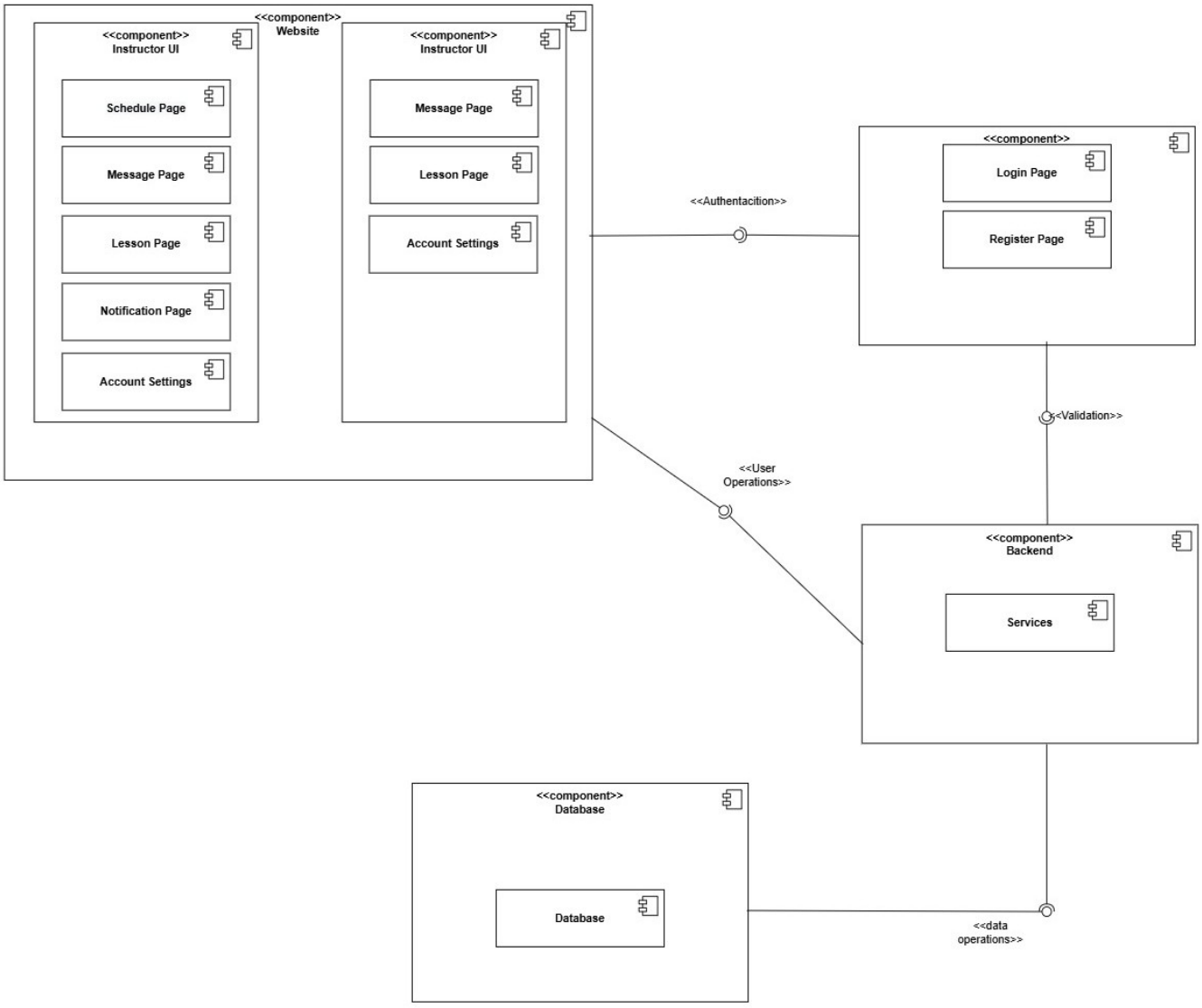
Requirement Description	FR/NFR	Implemented	Notes
Users shall register with university email and create account	FR	Yes	Signup via nickname, password, and email; email validation enforced
Users shall log in and securely authenticate	FR	Yes	Authentication page with feedback on failed login
Users shall reset password via email link	FR	Yes	Forgot password flow redirects to reset via secure link
Instructors shall create and publish lessons	FR	Yes	Instructors define course details and availability, review-based publish
Learners shall search lessons with filters	FR	Yes	Search via subject, rating, price, and personalized recommendations
Learners shall book and manage sessions	FR	Yes	Booking includes time slot selection, instructor approval, notifications
System shall provide real-time notifications	NFR	Yes	Notifications via UI, email, and push for key events
Platform shall support messaging and communication	FR	Yes	Secure messaging system with moderation and real-time chat

Users shall submit and view feedback & ratings	FR	Yes	Ratings and written feedback shown on instructor profile
System shall support secure online payments	FR	Yes	Payment module with escrow, refund handling, secure processing
System shall ensure data privacy (encryption, GDPR)	NFR	Yes	Encrypted data storage; GDPR-compliant data export and deletion
System shall auto-log out inactive users	NFR	Yes	30-min inactivity timeout, warning before logout
System shall support accessibility and multi-device use	NFR	Partially	Responsive UI noted; but WCAG/accessibility testing not shown
System shall provide admin tools for user/course management	FR	Yes	Admin dashboard allows user and course management
System shall maintain high uptime and fast performance	NFR	Yes	Uptime target and performance goals defined in NFR section
Users shall be able to create study groups	FR	Partially	Mentioned as feature but no UI, sequence or flow shown
Users shall be able to share learning resources	FR	Partially	Mentioned in use case but not supported by UI or diagrams
System shall provide real-time performance monitoring	NFR	No	Monitoring reports claimed but no interface/dashboard shown
System shall support zero-downtime updates	NFR	No	Zero-downtime mentioned; no architecture proof provided

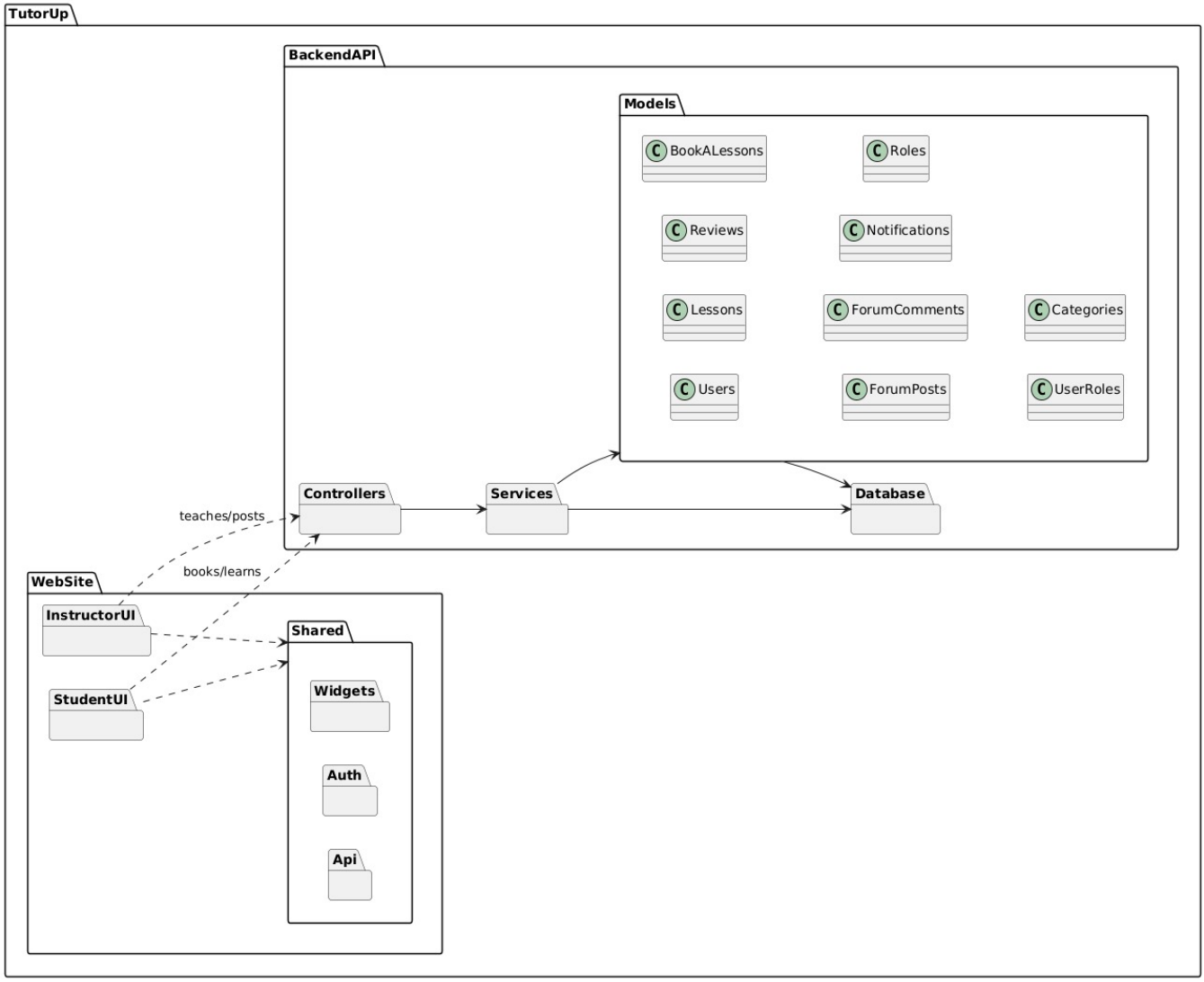
6 Systems Architecture Models

This section defines the high-level architecture of the TutorUP - Online Tutoring Platform.

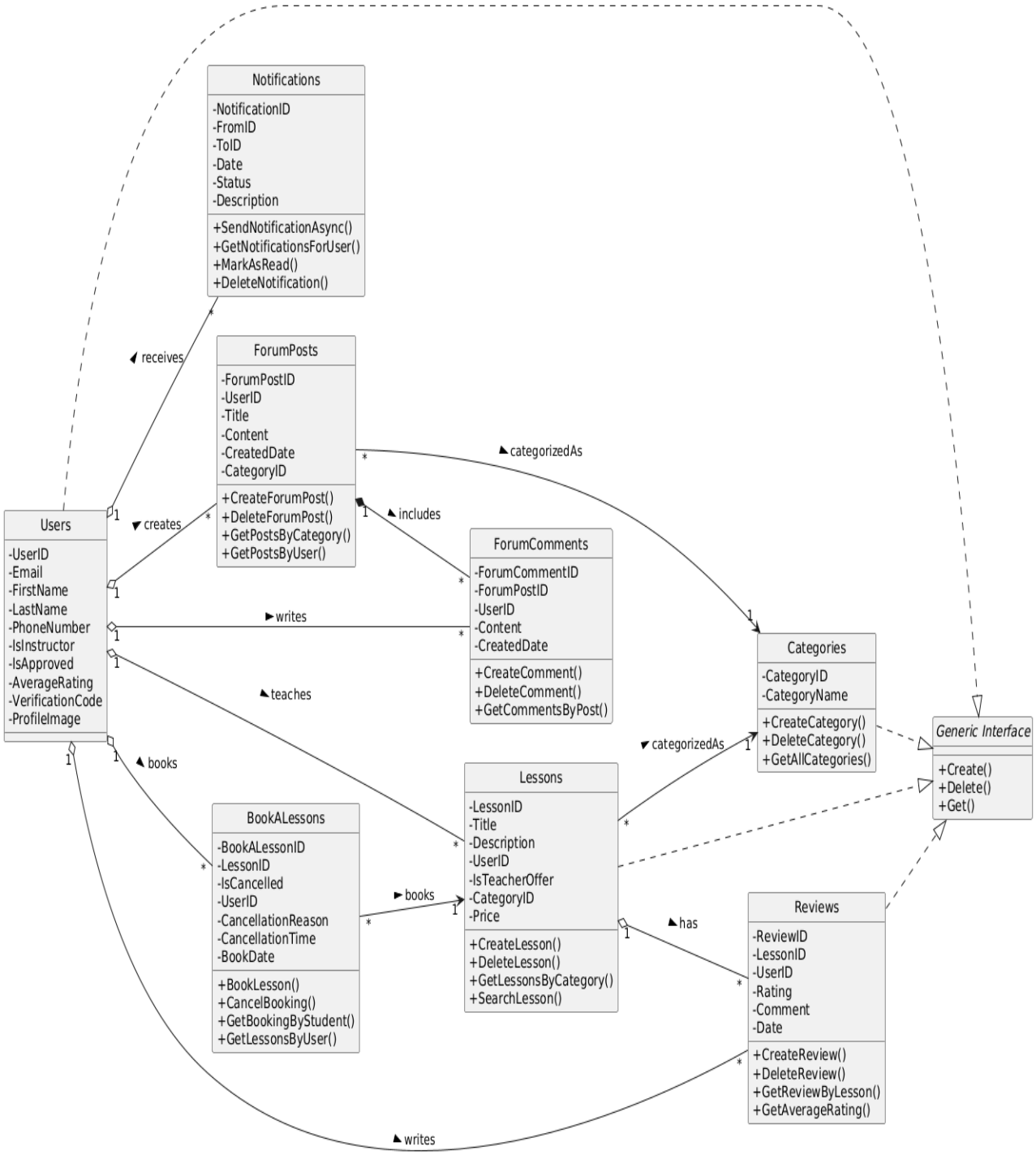
6.1 UML Component Diagram (Software Architecture)



6.2 2.2. UML Package Diagram (Physical Architecture)



7 Detailed Design Class Diagram

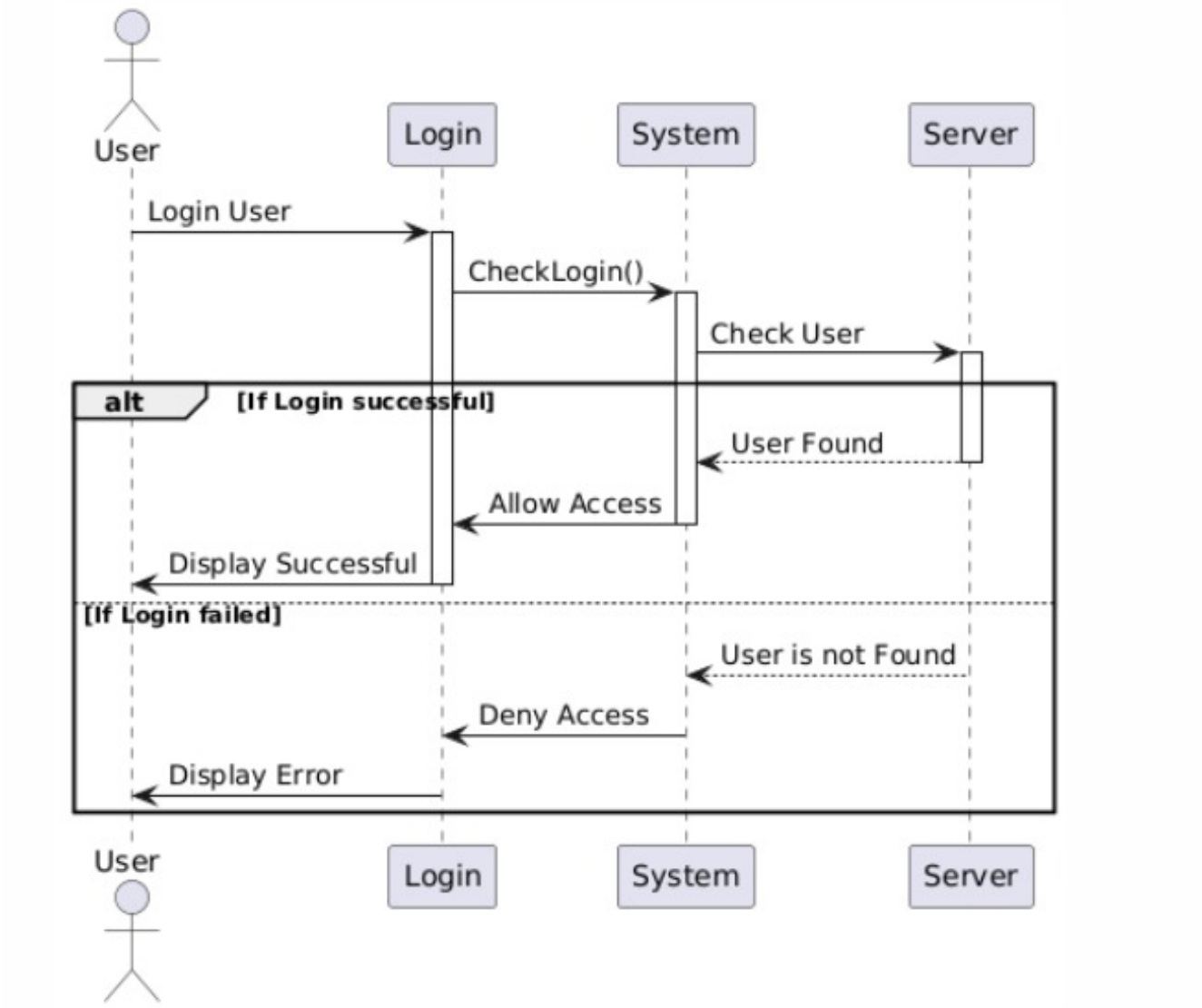


8 Dynamic Models

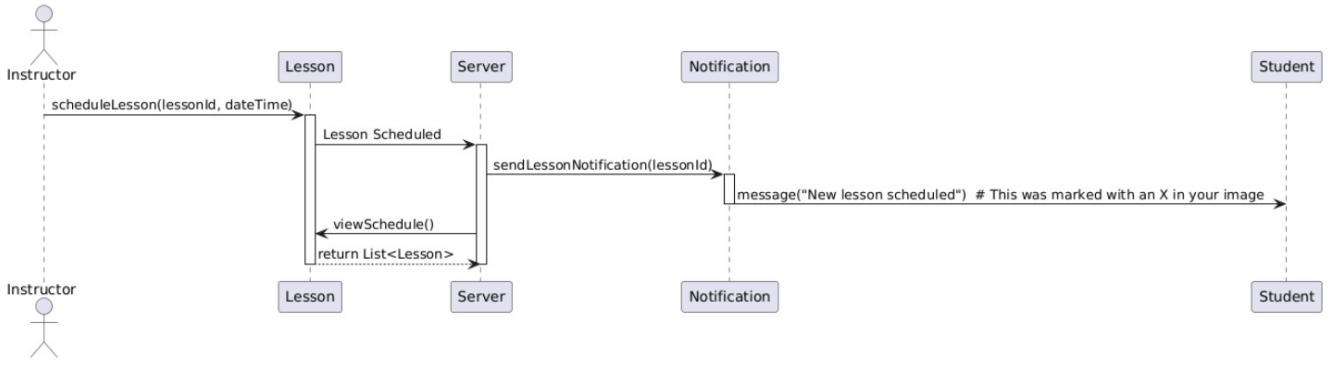
8.1 Sequence Diagrams

This section describes key interactions within the TutorUP system using sequence diagrams. These diagrams illustrate how different components or objects collaborate over time to achieve specific functionalities.

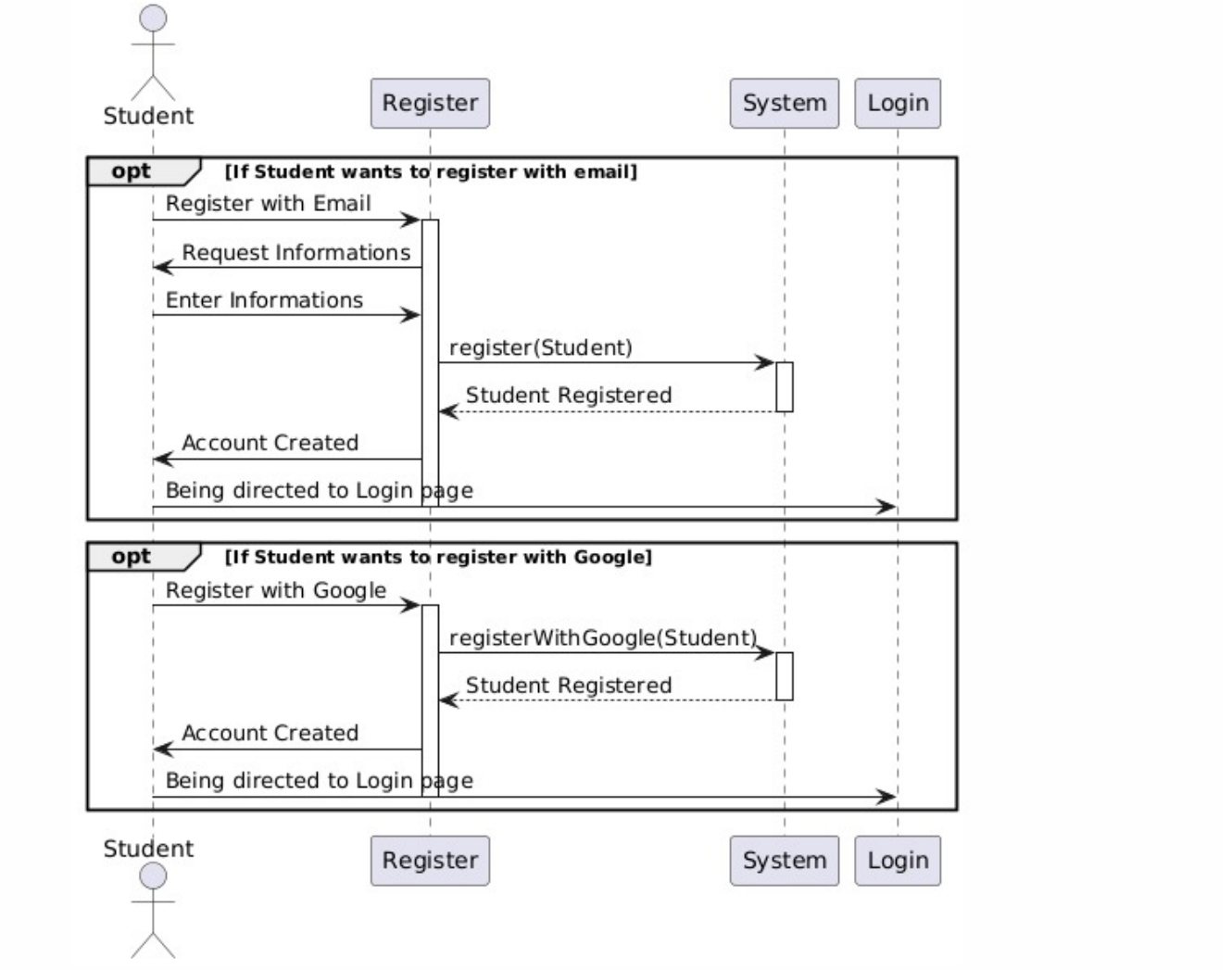
8.1.1 User Login Sequence



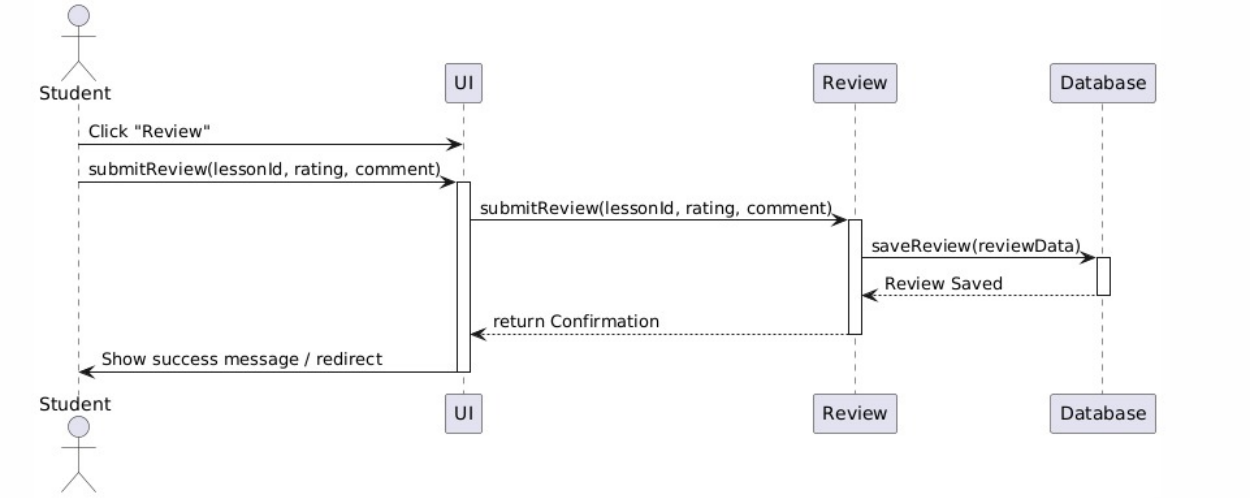
8.1.2 Instructor Schedules a Lesson



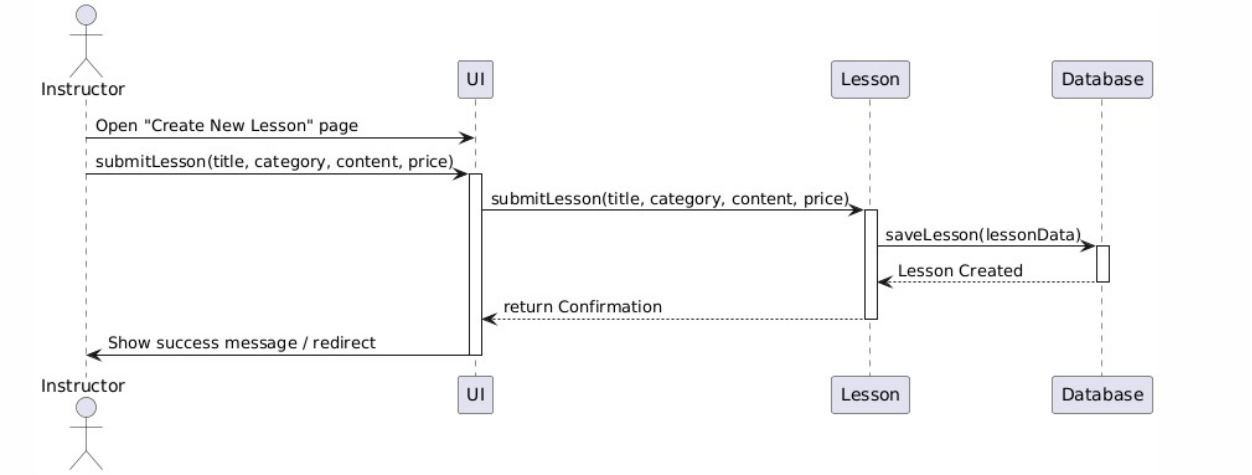
8.1.3 Student Registration Process



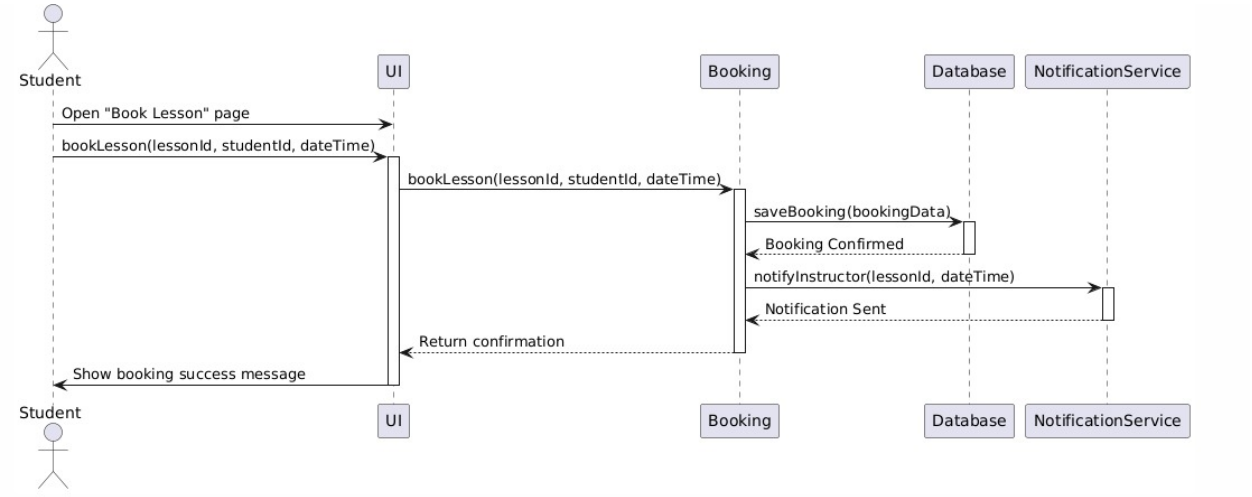
8.1.4 Student Submits a Review for a Lesson



8.1.5 Instructor Creates a New Lesson



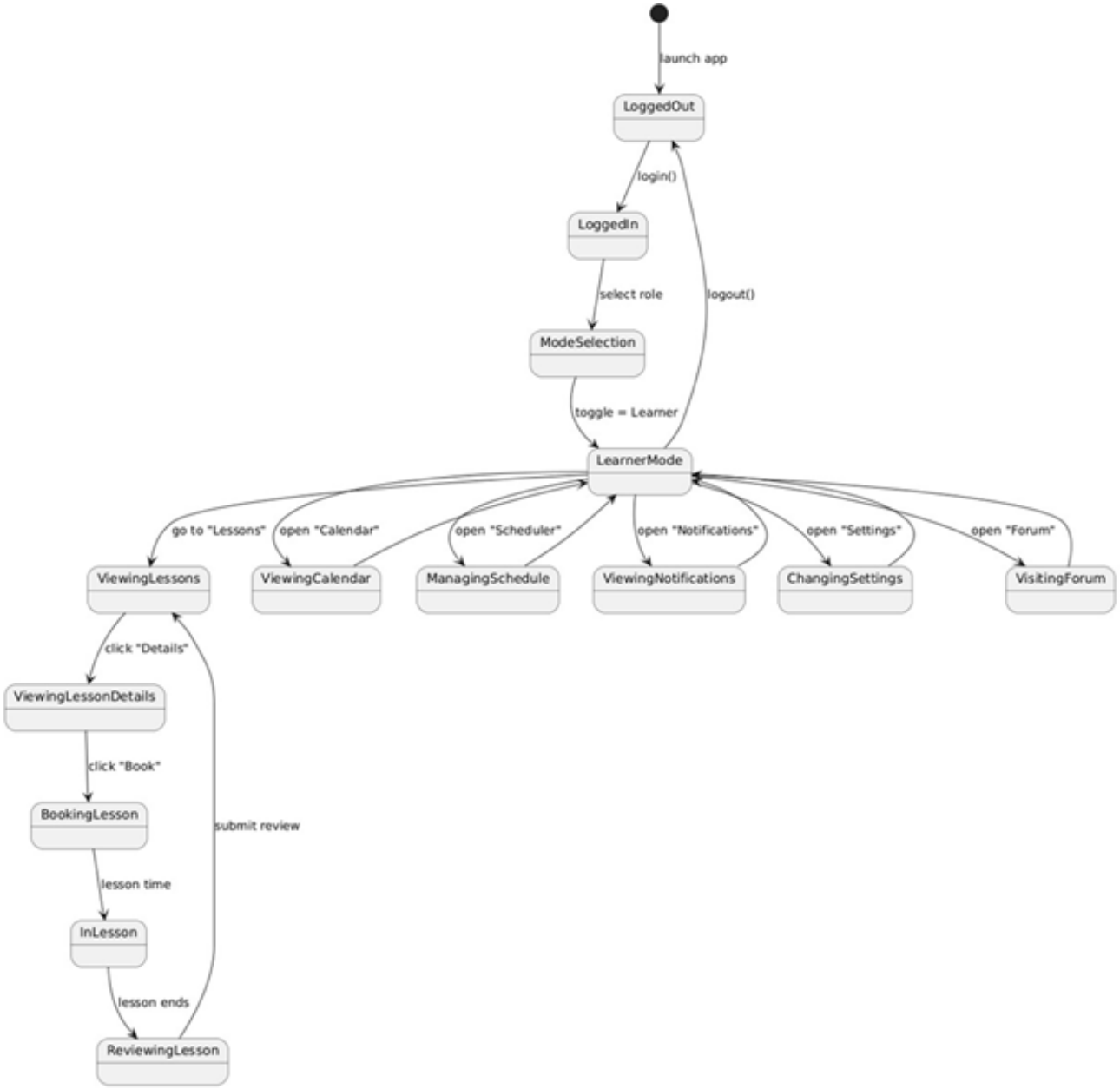
8.1.6 4.1.6. Student Books a Lesson



8.2 State Diagrams

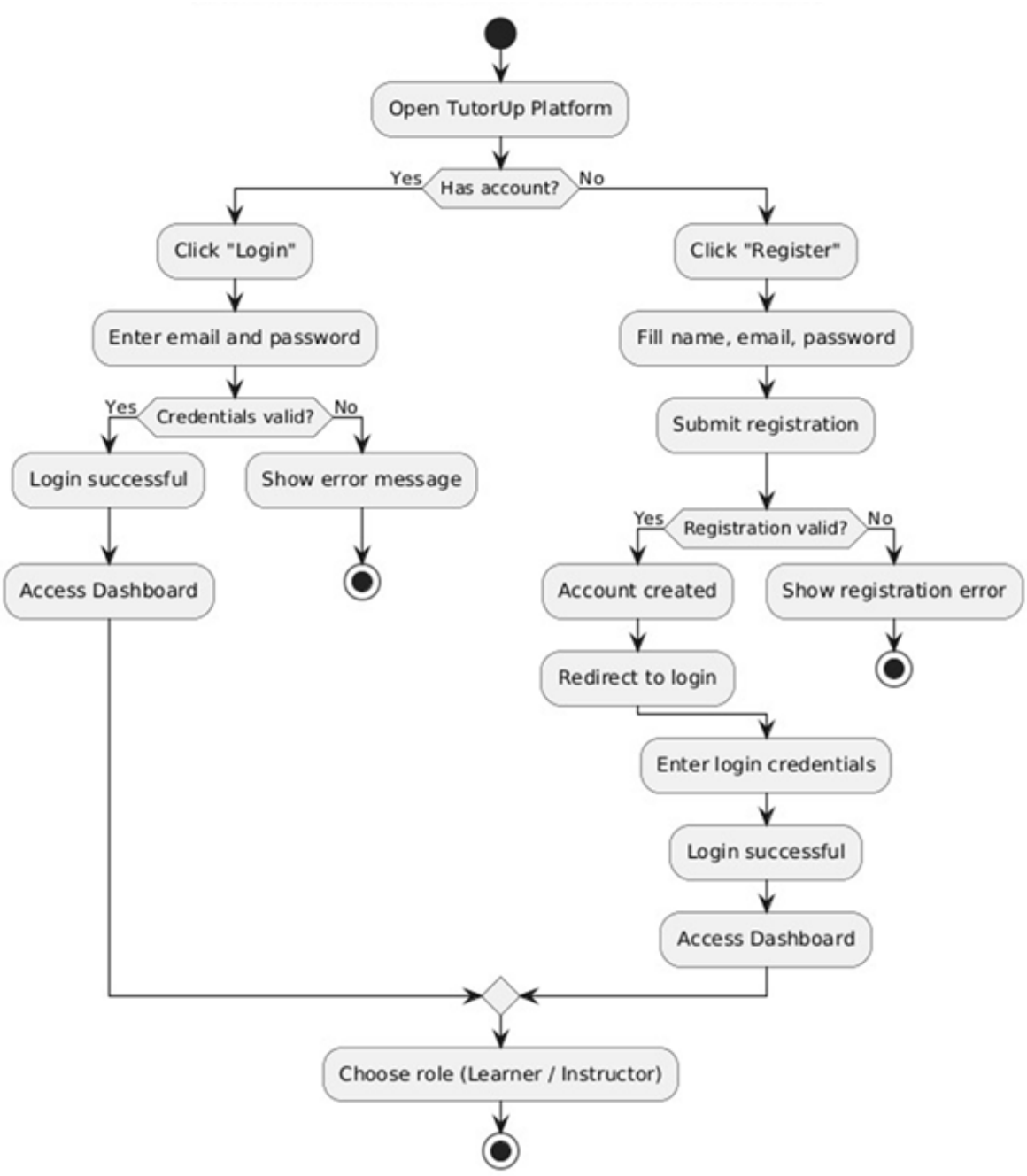
8.2.1 Instructor Mode State Diagram

8.2.2 Learner Mode State Diagram

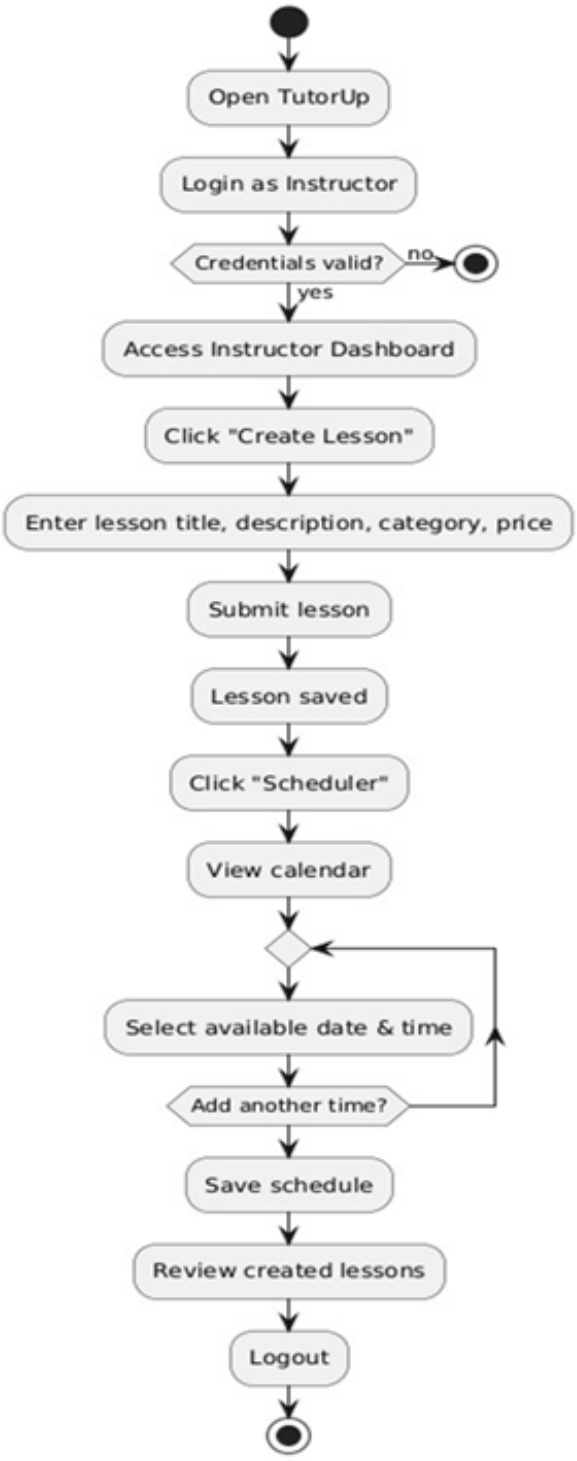


8.3 Activity Diagrams

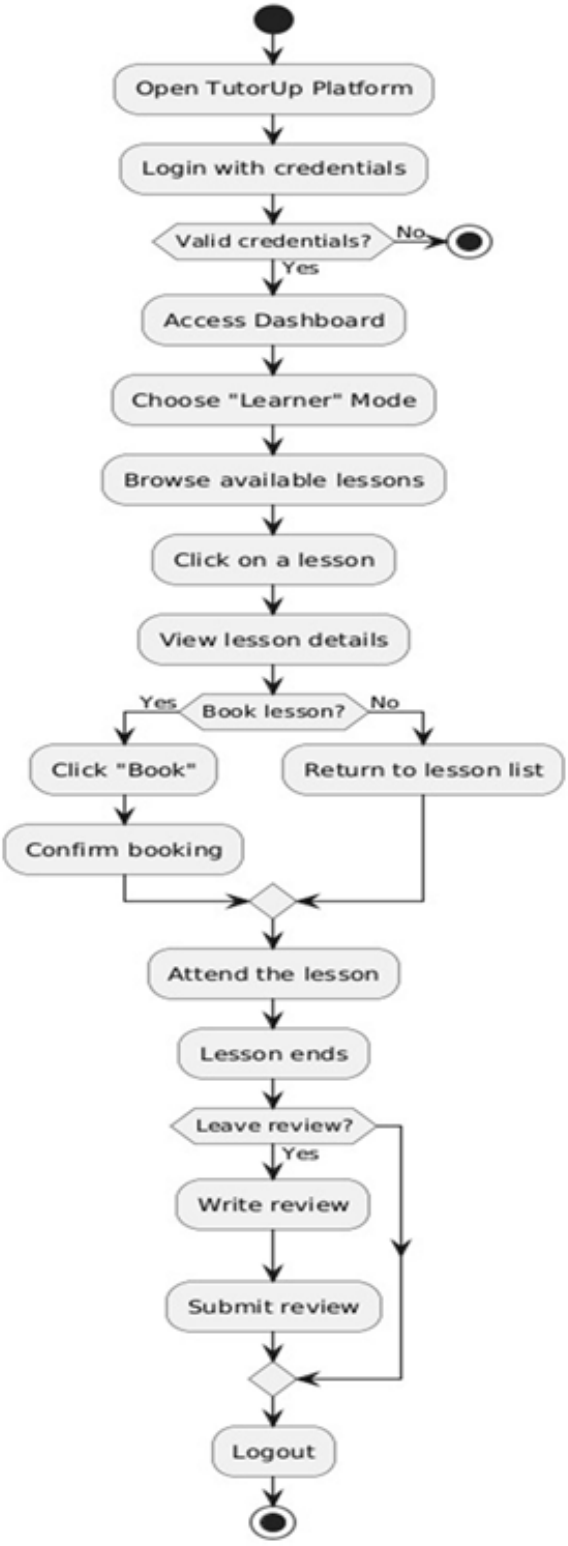
8.3.1 Student Login and Registration



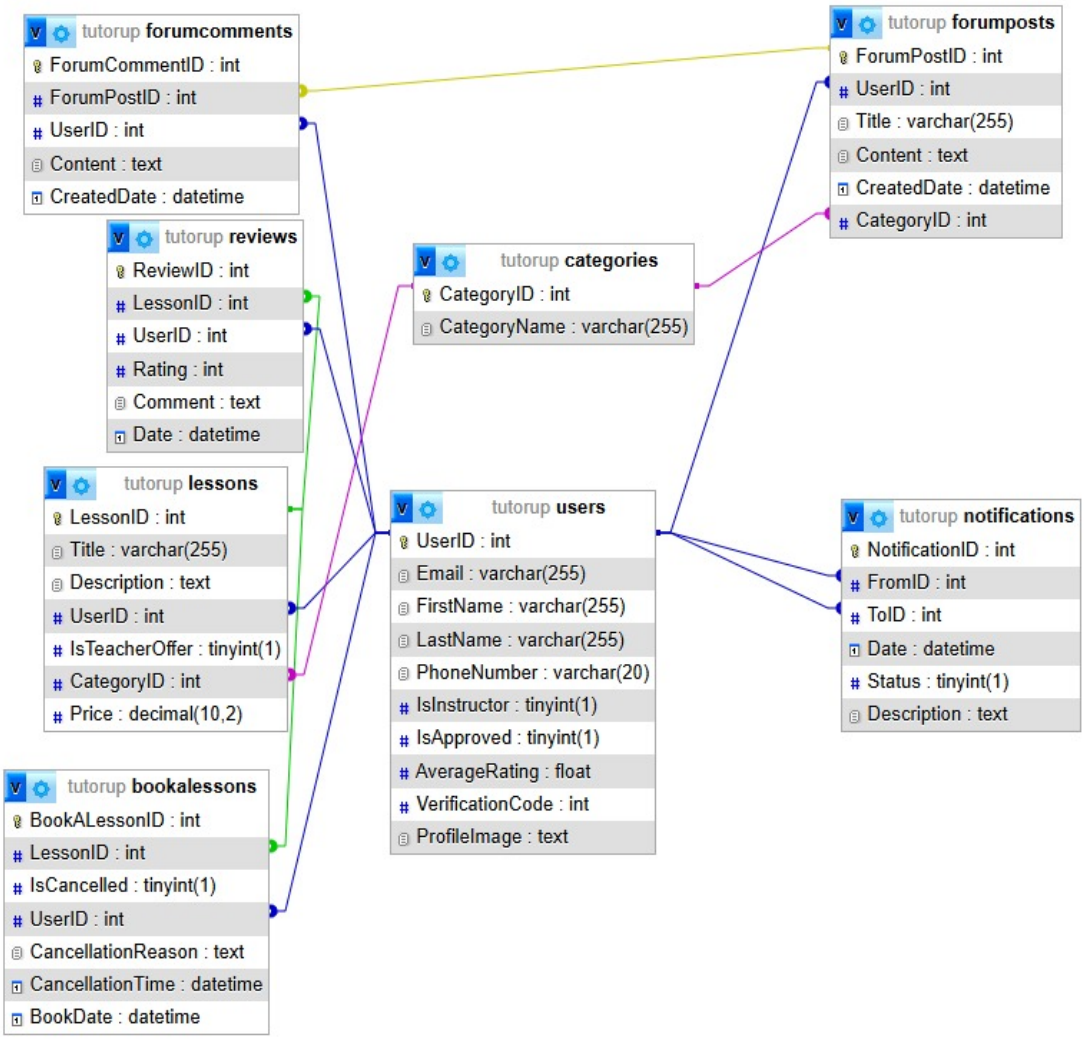
8.3.2 Instructor Creates Lesson and Manages Schedule



8.3.3 Student Lesson Booking and Review



9 Entity Relationship Diagram



10 Meeting Minutes

Meeting 1

Date: 09.03.2025

Time: 16.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Introductory meeting to get to know each other and brainstorm potential project topics.

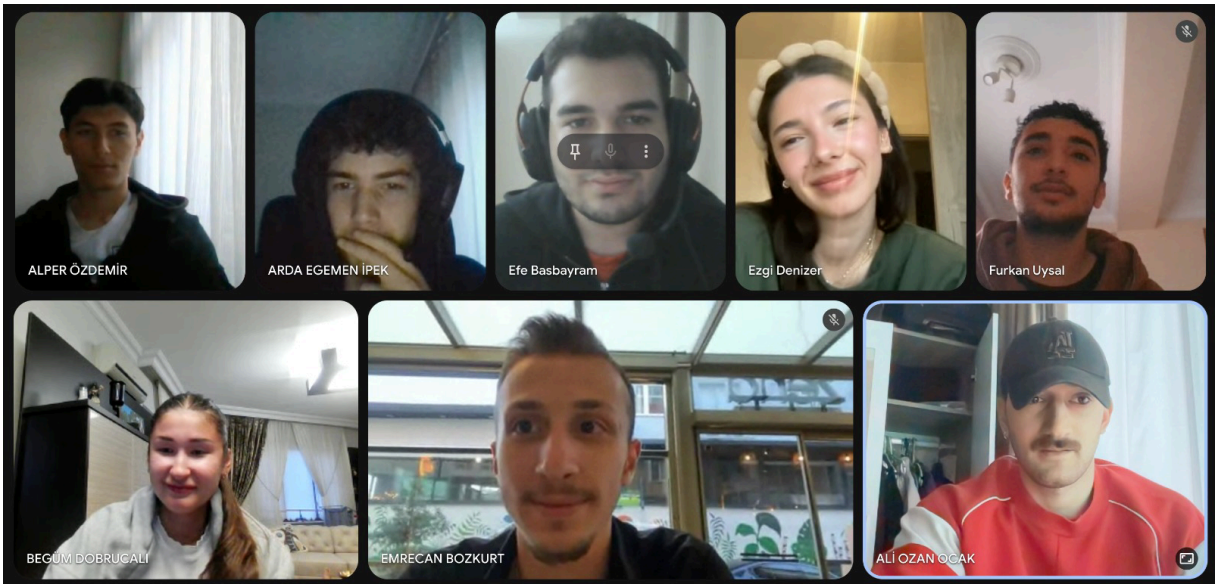
Meeting Notes: Students introduced themselves and briefly discussed expectations from the collaboration. A general discussion was held regarding potential systems to develop. TutorUP idea emerged and was favored.

Individual Contributions: All team members explore educational platforms.

Next Steps: Finalize the project idea in the next meeting

Action Items: Conduct competitor and technical feasibility research

Submitted by: Ezgi Denizer and Begüm Dobrucalı



Meeting 2

Date: 15.03.2025

Time: 17.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Define objectives and finalize project scope

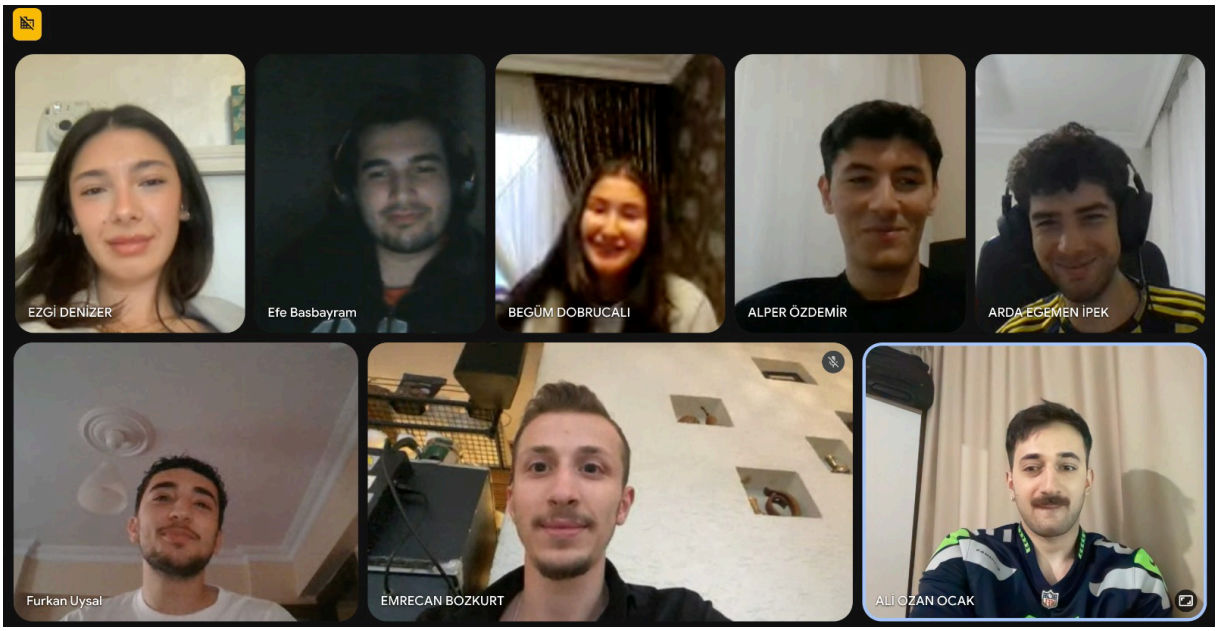
Meeting Notes: "TutorUP" name was chosen; agreed on goals like accessibility and platform structure.

Individual Contributions: Ezgi and Begüm define user types, main modules, and use case boundaries

Next Steps: Start writing the introduction and background sections

Action Items: Complete purpose and user role sections in the report

Submitted by: Begüm Dobrucalı



Meeting 3

Date: 29.03.2025

Time: 14.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Functional requirements

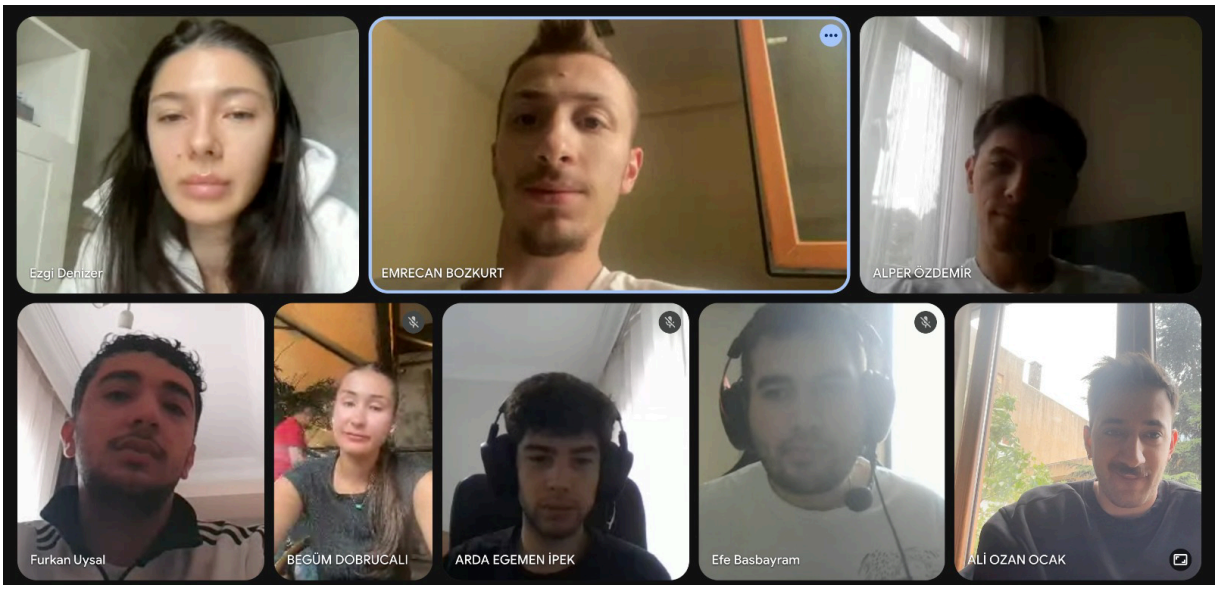
Meeting Notes: Core functions such as lesson search, booking, and user authentication were discussed

Individual Contributions: Ezgi and Begüm focused on refining functional requirements and use cases. Emrecan and Ali Ozan worked on frontend development of booking and login modules.

Next Steps: Complete detailed requirement writing

Action Items: Finalize functional requirement documentation

Submitted by: Ezgi Denizer



Meeting 4

Date: 06.04.2025

Time: 16.00

Location: Top Roasters

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Use case design

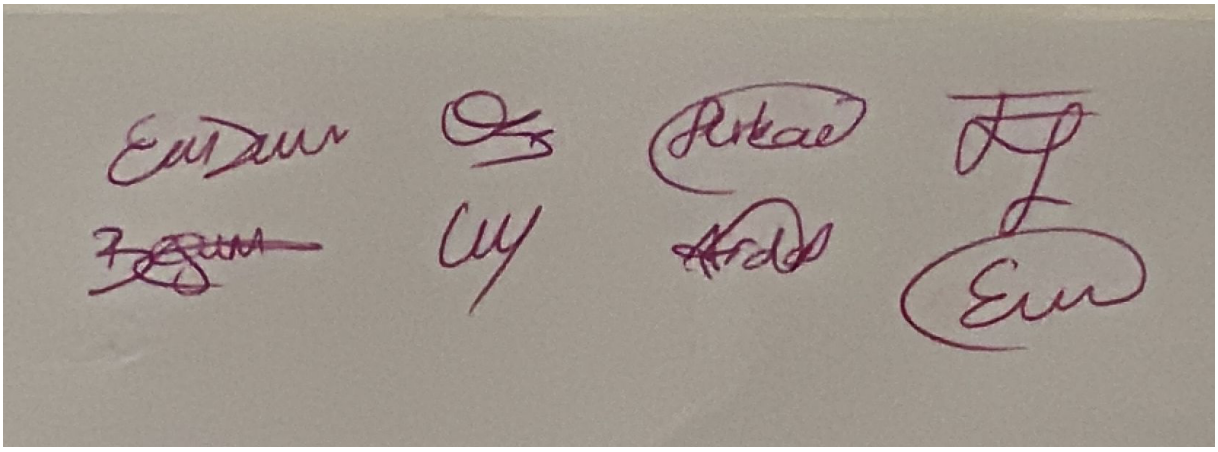
Meeting Notes: Prioritized and defined initial use cases UC01–UC06

Individual Contributions: Begüm will write use case flows. Ezgi will identify actors and alternatives.

Next Steps: Finalize all use case specifications

Action Items: Complete use case descriptions and diagrams.

Submitted by: Muharrem Efe Başbayram



Meeting 5

Date: 09.04.2025

Time: 22.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Preparation of the analysis report

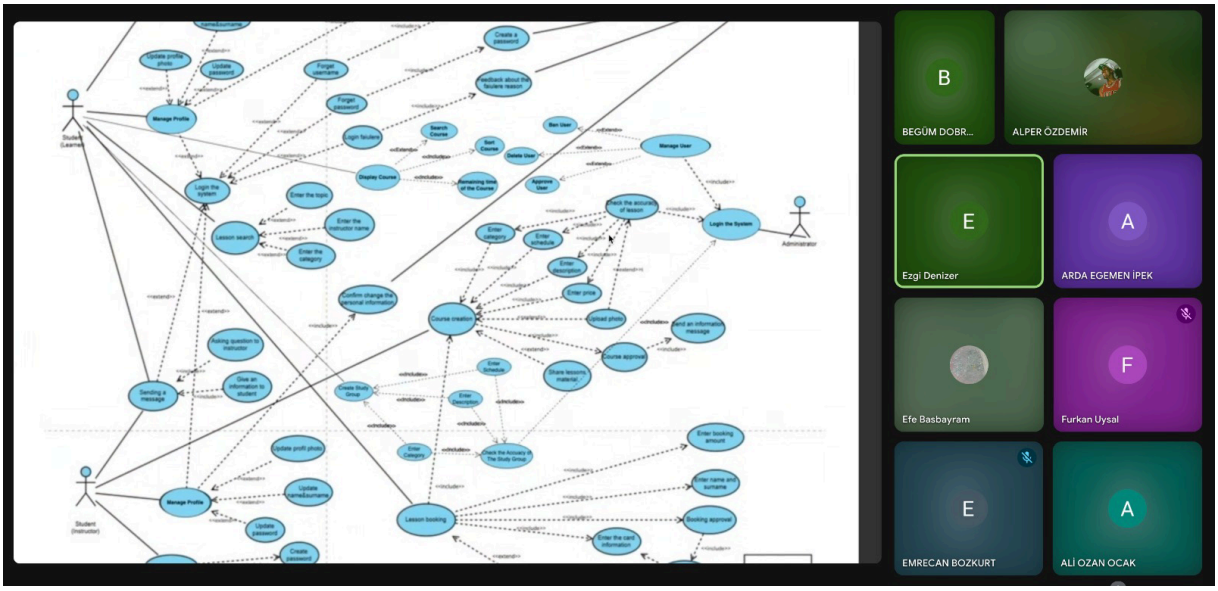
Meeting Notes: Collected and reviewed requirements and use cases for the report

Individual Contributions: Ezgi and Begüm will organize and write NFRs and Volere tables.

Next Steps: Submit the analysis report

Action Items: Assemble and review the final analysis report

Submitted by: Emrecan Bozkurt



Meeting 6

Date: 18.04.2025

Time: 19.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: User interface mockups

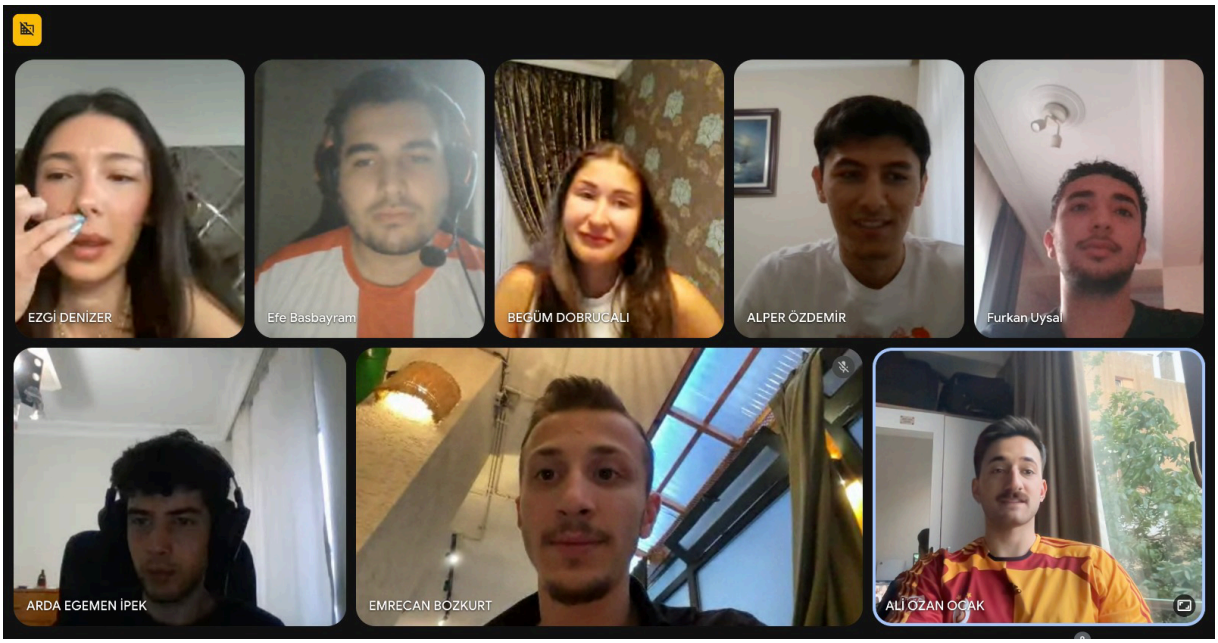
Meeting Notes: Initial screen designs were created; page-to-page flow reviewed

Individual Contributions: Efe and Emrecan will sketch UI pages and align interface with requirements.

Next Steps: Build initial GUI prototype

Action Items: Prepare low-fidelity screen designs

Submitted by: Furkan Uysal



Meeting 7

Date: 25.04.2025

Time: 18.00

Location: Schollars Cafe

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Prototype development.

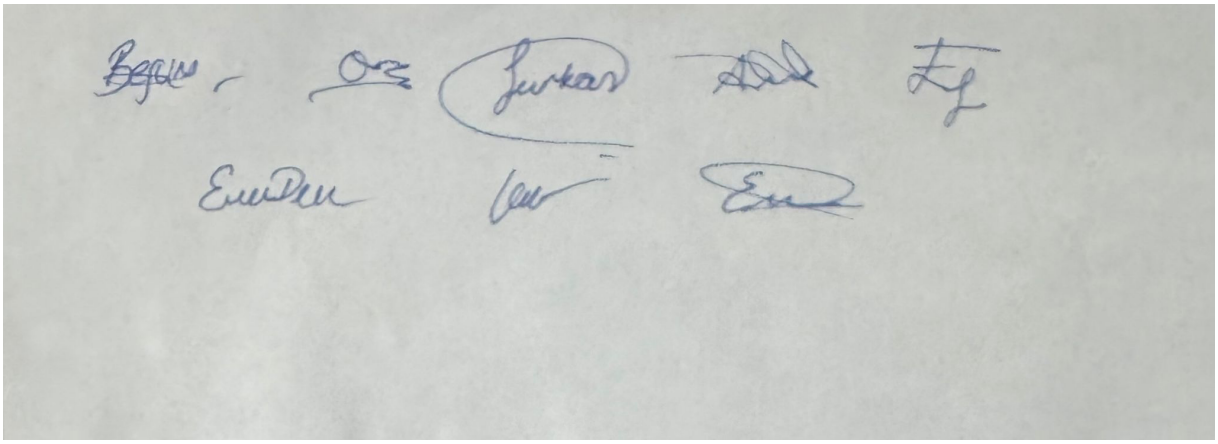
Meeting Notes: Started frontend implementation using design drafts

Individual Contributions: Ozan and Alper worked on frontend development of booking and login modules. Arda and Furkan implement main UI pages and integrate input features

Next Steps: Test initial GUI screens and interactions

Action Items: Complete and test interface functionality

Submitted by: Arda Egemen İpek and Furkan Uysal



Meeting 8

Date: 27.04.2025

Time: 19.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Backend structure planning and task assignment

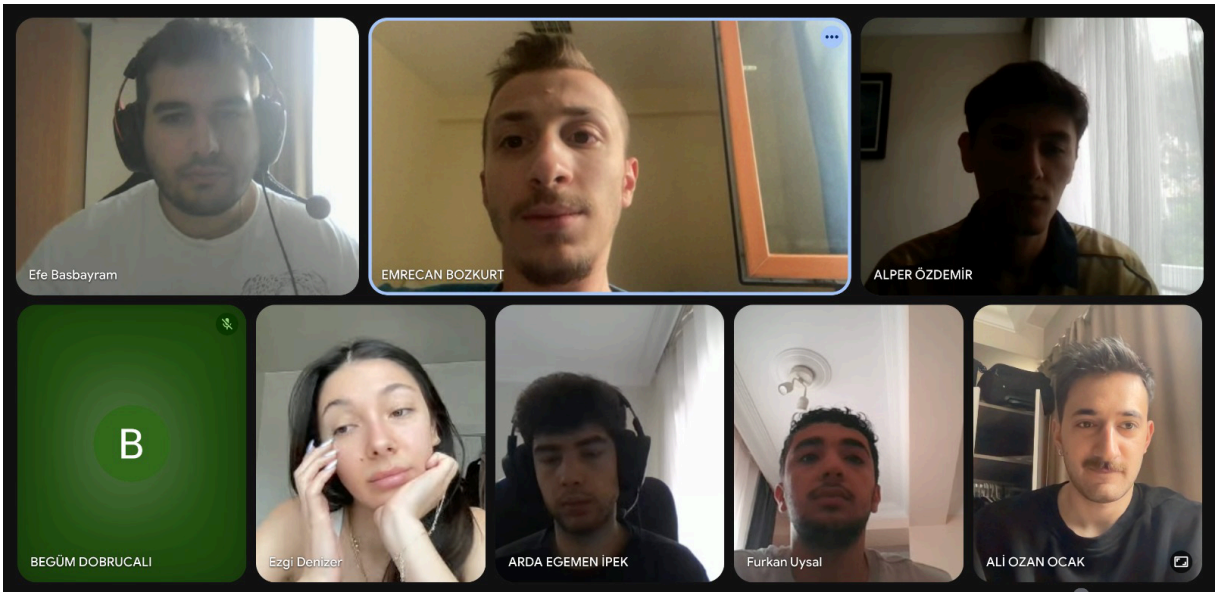
Meeting Notes: Discussed the overall backend development architecture. Key components such as API design, authentication flow, and database connections were planned.

Individual Contributions: Emrecan and Alper planned the core API endpoints. Efe and Furkan worked on designing the data structures required for frontend-backend interaction.

Next Steps: Start implementing backend modules

Action Items: Develop user management and authentication modules

Submitted by: Muharrem Efe Başbayram



Meeting 9

Date: 28.04.2025

Time: 21.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Backend testing scenarios and integration plan

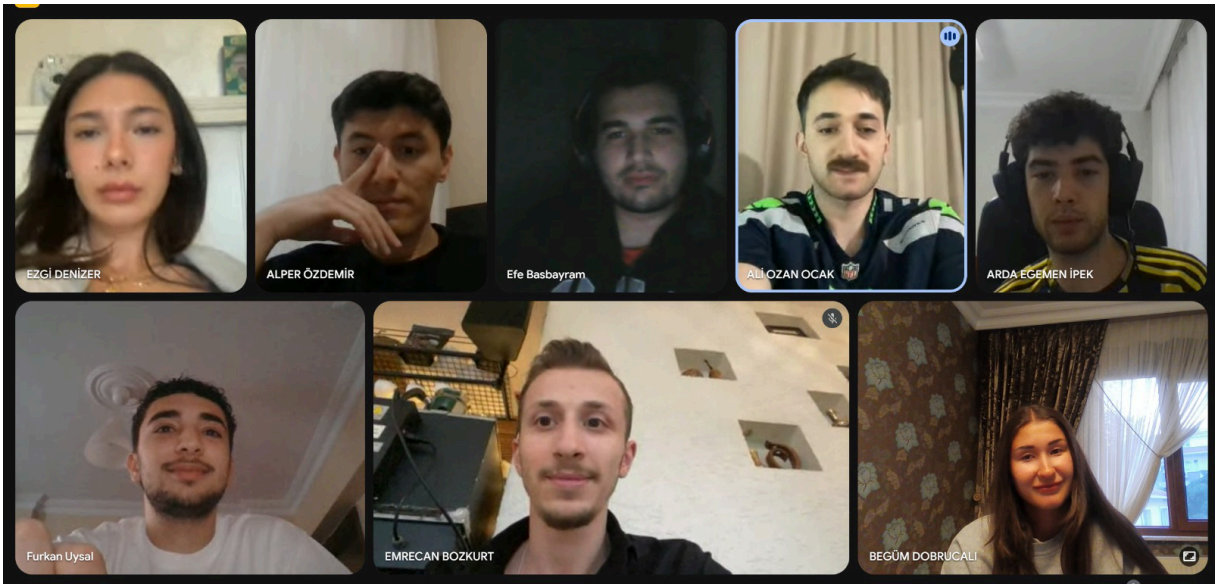
Meeting Notes: Discussed backend testing strategy, including unit test cases and expected outcomes. The team decided on tools and environments for testing.

Individual Contributions: Alper was responsible for setting up the test environment. Arda and Ozan prepared endpoint test scenarios.

Next Steps: Begin implementing and running tests

Action Items: Write and execute test cases for backend modules

Submitted by: Alper Özdemir



Meeting 10

Date: 30.04.2025

Time: 21.30

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Database and class structure

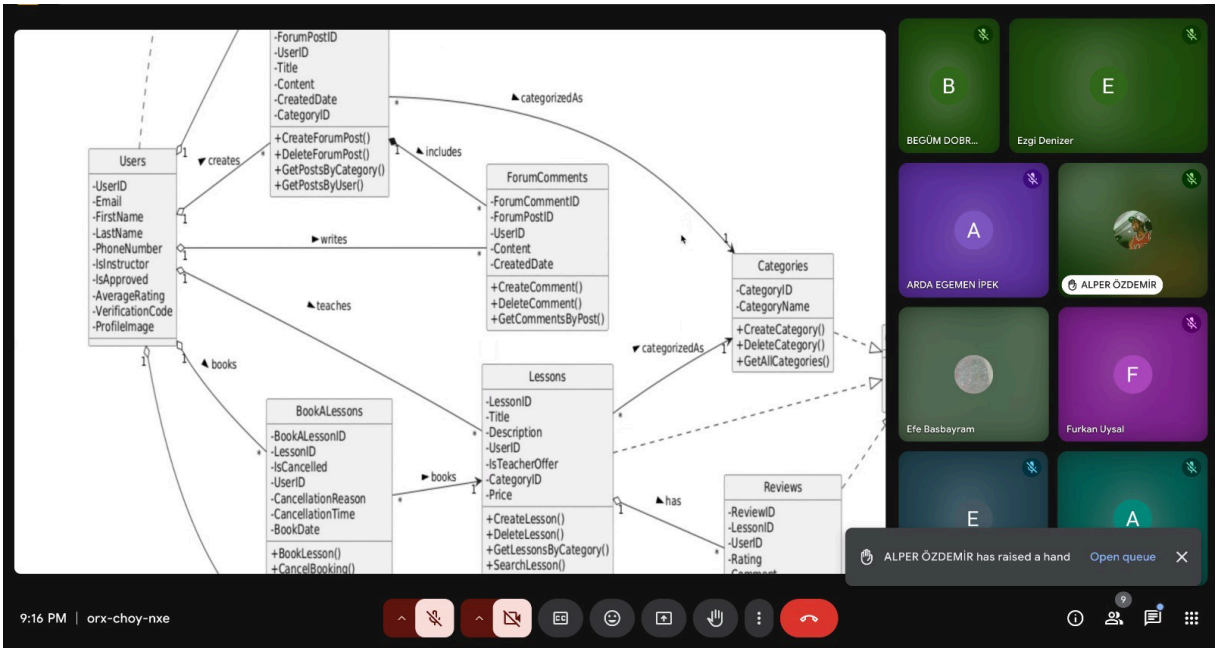
Meeting Notes: Designed database and relationships between system components

Individual Contributions: Ali Ozan and Arda define entity relationships and object classes

Next Steps: Initiate backend implementation

Action Items: Finalize class and ER diagrams

Submitted by: Muharrem Efe Başbayram



Meeting 11

Date: 09.02.2025

Time: 17:00

Location: Schollars Cafe

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Backend and API integration

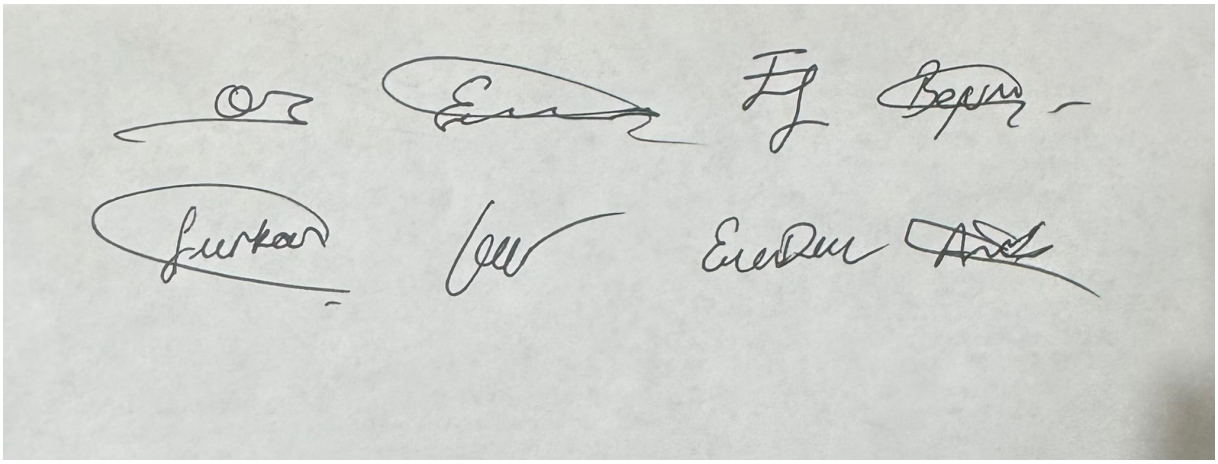
Meeting Notes: Backend APIs and database connectivity were partially integrated

Individual Contributions: Emrecan and Alper implement REST services and define test scenarios

Next Steps: Integrate frontend with backend

Action Items: Develop and connect core API endpoints

Submitted by: Ali Ozan Ocak



Meeting 12

Date: 13.05.2025

Time: 21.30

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Final system integration

Meeting Notes: All modules were integrated and tested for consistency

Individual Contributions: Perform system testing and correct UI bugs

Next Steps: Submit the full implementation

Action Items: Conduct final testing and documentation updates

Submitted by: Alper Özdemir

Meeting 13

Date: 22.05.2025

Time: 21.00

Location: Online (Google Meet)

Attendees: Ezgi Denizer, Begüm Dobrucalı, Emrecan Bozkurt, Muharrem Efe Başbayram, Ali Ozan Ocak, Alper Özdemir, Arda Egemen İpek, Furkan Uysal

Agenda: Project closure

Meeting Notes: All members reflected on the project and shared insights

Individual Contributions: Review final report and ensure all components are ready

Next Steps: Submit the final report and meeting minutes

Action Items: Upload all deliverables and complete documentation

Submitted by: Begüm Dobrucalı and Ezgi Denizer

11 Glossary & References

StarUML. (n.d.). *StarUML - UML/MDA platform* from https://staruml.io

Balsamiq. (n.d.). *Balsamiq Wireframes* from https://balsamiq.com

Visual Paradigm. (n.d.). *Visual Paradigm Online* https://online.visual-paradigm.com

Hasan, H., & Bhuiyan, A. (2024). *User Centered Non-Functional Requirements Specification – An Extended Use-Case Diagram*. *Procedia Computer Science, 235*, 240–249. https://doi.org/10.1016/j.procs.2024.04.026

https://plantuml.com

Timothy C. Lethbridge and Robert Laganière, Object-Oriented Software Engineering: Practical Software Development using UML and Java, any edition, McGraw Hill

Walter Savitch, Absolute C++, any edition, Addison-Wesley Longman

https://www.geeksforgeeks.org/