

Software Requirements Specification (SRS)

Project: Tutor Uberization Platform

Prepared by:

Mashael Saeed (S23108689)
Sarah Elshiaty (S23108195)
Seifeldin Elshiaty (S23108194)

Course: CS31511 – Software Engineering

Instructor: Dr. Passent Elkafrawy

Effat University – Spring 2025

Date: December 4, 2025

Contents

Executive Summary	4
1 Introduction	8
1.1 Purpose	8
1.2 Scope	9
1.3 Definitions and Acronyms	11
1.4 Competitors Analysis and Stakeholder Views	11
1.4.1 Existing Competitors	12
1.4.2 Existing Solutions and Gaps	12
1.4.3 Feature Comparison	13
1.4.4 Customer Feedback and Stakeholder Views	13
1.4.5 Market Opportunity and Differentiation	14
1.4.6 High-Level Go-to-Market Plan	14
1.5 References	15
2 Overall Description	16
2.1 System Perspective	16
2.2 User Characteristics	17
2.3 Constraints	18
2.4 Assumptions and Dependencies	18
3 Functional Requirements	19

3.1	Requirement Summary Table	20
3.2	Main Use Cases	20
3.3	Sustainable Development Goals (SDG) Alignment	21
3.4	Requirements–Use Case Traceability Matrix	22
4	Non-Functional Requirements	23
4.1	Performance Requirements	23
4.2	Security Requirements	23
4.3	Reliability and Availability	23
4.4	Usability and Accessibility	24
4.5	Maintainability and Scalability	24
5	System Features	25
5.1	Account Management	25
5.2	Tutor Search & Recommendation	25
5.3	Booking and Scheduling	25
5.4	Payments, Refunds, Receipts	25
5.5	Session Attendance	26
5.6	Reviews and Ratings	26
5.7	Verification and Compliance	26
6	System Models	27
6.1	Use-Case-Centred Behavioural Models	27
6.1.1	UC1 – Book Tutoring Session	27
6.1.2	UC2 – Manage Payment and Refunds	32
6.1.3	UC3 – Manage Tutor Availability and Sessions	37
6.1.4	UC4 – Tutor Verification and Compliance	41
6.1.5	UC5 – Review and Rating System	45
6.1.6	UC6 – Partner and Regulator Reporting	47
6.2	Domain Model (Class Diagram)	50

6.3	Entity Relationship Diagram (ERD)	52
6.4	Entity Relationship Diagram (ERD)	54
6.5	System Architecture	56
7	Design Patterns	58
7.1	Observer Pattern – Session Notifications	58
7.2	Strategy Pattern – Payment Methods	59
7.3	State Pattern – Booking Lifecycle	60
8	User Interface / User Experience (UI/UX)	62
8.1	Home Screen	63
8.2	Tutor Profile Screen	64
8.3	Booking and Payment Screen	65
8.4	Why This UX Works	66
8.5	Conclusion	66
9	Validation and Acceptance Criteria	67
10	Appendices	69
10.1	Business Model Canvas	69
10.2	Gantt Chart	70
10.3	Project Workload Distribution	70
10.4	Glossary	71

Executive Summary

The **Tutor Uberization Platform** is a hybrid web and mobile application that connects learners with verified tutors in a safe, flexible, and intelligent way. It addresses a clear gap in the Saudi tutoring market by combining **local exam focus** (Qiyas, Tahsili), **global exam support** (IELTS, SAT, ACT), **strict tutor verification**, and **AI-based personalization** in one platform.

Problem Context:

Parents and students struggle to find trusted tutors who:

- Are *actually qualified* and vetted.
- Support both online and safe face-to-face sessions.
- Understand local curricula and high-stakes exams (Qiyas, Tahsili).
- Can handle diverse learner needs, including special needs cases.

Existing platforms and traditional centers cover some of these needs, but not all together, and often lack strong verification and personalization.

Proposed Solution:

The Tutor Uberization Platform provides:

- A **marketplace** where students and parents can search for verified tutors using filters (subject, level, exam, city, mode, language).
- **Hybrid tutoring:** online sessions (video meeting links) and regional face-to-face sessions with clear parental approval and safety rules.
- **AI-based tutor recommendation** to suggest the most relevant tutors for each learner based on profile, history, and preferences.

- **End-to-end booking flow:** availability management, booking confirmation, secure payment, reminders, attendance logging, and post-session reviews.
- **Verification and reporting tools** for platform managers, regulators, and partner institutions (schools and universities).

Key Objectives:

- Simplify and secure the process of finding and booking qualified tutors.
- Improve exam readiness and academic performance for K–12, university, and professional learners.
- Provide a transparent, data-driven platform with clear KPIs for partners and investors.
- Align with Saudi Vision 2030 goals for digital transformation and education quality.

Target Users:

- K–12 students and their parents (core segment for school and Qiyas/Tahsili tutoring).
- University students and professionals needing subject-specific help and certification prep.
- Learners with special needs (ADHD, dyslexia, mild autism, dyscalculia) who require patient, structured teaching.
- Tutors (individuals and centers) who want a reliable channel to reach students and manage bookings.
- Educational partners (schools, universities) and regulators (Ministry of Education) who need quality assurance and compliance.

Major Functional Requirements (High-Level):

- Search and filter only *verified* tutors based on multi-criteria filters.
- Book tutoring sessions and manage availability without conflicts.

- Process secure online payments, generate digital receipts, and handle refund requests.
- Provide AI-based recommended tutors per search or learner profile.
- Record attendance, session outcomes, and post-session ratings and reviews.
- Provide dashboards and exportable reports for partners and regulators.

Major Non-Functional Requirements:

- **Performance:** Search and booking operations should respond within 2 seconds for 95% of requests under normal load.
- **Security:** All communication over HTTPS; personal data encrypted at rest; payments processed via PCI-compliant gateways.
- **Reliability:** Core services (booking, payments) should achieve at least 99.5% uptime per month.
- **Usability:** Mobile-first, bilingual Arabic/English UI; simple flows suitable for parents and younger learners.
- **Accessibility:** Screen layouts and flows designed to support special needs personas (short steps, clear feedback, minimal clutter).

Expected Benefits:

- Increased trust and safety for parents compared to unverified private tutors.
- Better exam preparation and continuous support for students.
- New revenue streams and structured tutoring programs for partner institutions.
- Stronger evidence and analytics for investors and policy makers.

Risks and Dependencies:

- Dependence on third-party services (payment, SMS/email notifications, identity verification).

- Risk of low adoption if trust is not established early (e.g., insufficient verified tutors at launch).
- Regulatory changes in education or data privacy that may require design or operational updates.
- Need for ongoing tutor quality monitoring to maintain high ratings and student satisfaction.

In conclusion, the Tutor Uberization Platform aims to become a **trusted, localized, and intelligent tutoring ecosystem** that supports diverse learners, empowers tutors, and enables schools, universities, and regulators to collaborate around quality, safety, and performance.

Chapter 1

Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to formally define the requirements of the Tutor Uberization Platform. It serves as a shared reference for:

- **Developers and architects** – to design and implement the software according to agreed functionalities and constraints.
- **Testers and QA engineers** – to derive test cases, scenarios, and acceptance criteria.
- **Project managers and platform managers** – to plan, monitor, and validate project scope.
- **Stakeholders** (students, parents, tutors, schools, universities, regulators, investors) – to understand what the platform will and will not provide in its initial versions.

This SRS focuses on the MVP and near-term releases while remaining extensible for future features such as advanced gamification and more complex analytics. The system is primarily intended for K–12 learners and their parents, university students, professional learners, verified tutors, educational partners, and regulators, who will rely on it as a secure and trusted environment for tutoring and exam preparation.

1.2 Scope

The Tutor Uberization Platform is a cloud-hosted, hybrid tutoring marketplace. Its main goals are:

- To connect learners (K–12, university, professionals) with verified tutors who can provide online and face-to-face sessions.
- To support local Saudi exams (Qiyas, Tahsili) and global exams (IELTS, SAT, ACT), in addition to regular school and university subjects.
- To ensure trust and safety through verification workflows, ratings and reviews, and clear policies for cancellations and refunds.
- To provide AI-based tutor recommendations and basic learning analytics for improved matching and engagement.

At a high level, the system will:

- Offer registration and profile management for students, parents, tutors, and partners.
- Allow students/parents to search, filter, and book sessions with tutors.
- Allow tutors to manage availability and sessions.
- Handle payment transactions, receipts, disputes, and refunds.
- Send reminders and track session attendance.
- Generate reports and dashboards for platform managers, partners, and regulators.

From a business perspective, the platform operates in a rapidly growing Saudi education and tutoring market. According to recent studies, there are over 7 million K–12 students in Saudi Arabia, and the private tutoring market is valued at approximately \$90 million in 2024, projected to reach around \$300 million by 2033 with a CAGR of about 13.2%. This indicates a strong demand for innovative, localized, and secure digital tutoring solutions that cover both national exam preparation (Qiyas, Tahsili) and international standardized tests (IELTS, SAT, ACT). The scope of the system is therefore to:

- Provide a single trusted platform where parents and students can find verified tutors for local and global exams.
- Address key pain points identified in the market study, such as lack of tutor verification, weak personalization, and limited support for learners with special needs.
- Offer hybrid learning (online and face-to-face) with built-in safety controls and parental oversight.

Out of scope for this SRS:

- Full Learning Management System (LMS) capabilities (e.g., complete course authoring, assignment management, and grading).
- Detailed curriculum content creation; tutors are responsible for their pedagogical content within broad guidelines.
- In-depth social networking features (e.g., public groups, feeds) beyond what is needed for reviews and notifications.

1.3 Definitions and Acronyms

Term	Definition
Tutor Uberization Platform	The system being developed; a hybrid tutoring marketplace for verified tutors and learners.
Student	Any learner using the platform (K–12, university, or professional).
Parent / Guardian	The decision-maker and payer for underage learners.
Tutor	A verified educator providing tutoring sessions through the platform.
Session	A scheduled tutoring meeting (online or face-to-face) booked through the platform.
Booking	A confirmed reservation for a session between a student and a tutor at a specific date and time.
MVP	Minimum Viable Product, the first deployable version with essential features.
PDPL	Personal Data Protection Law of Saudi Arabia.
KPI	Key Performance Indicator; metrics used to evaluate success (e.g., completion rate, repeat bookings).
AI Recommendation	Algorithm that ranks tutors based on learner profile, history, and preferences.
Hybrid Tutoring	Combination of online and in-person tutoring under one coordinated platform.

1.4 Competitors Analysis and Stakeholder Views

This section summarizes the market study conducted for the Tutor Uberization Platform, focusing on existing competitors, customer feedback, stakeholder perspectives, and differentiation.

1.4.1 Existing Competitors

Several competitors already exist in the tutoring and EdTech sector, both regionally and internationally:

1. **Noon Academy (KSA)** – Offers online tutoring and group classes but lacks rigorous tutor verification and advanced personalization features.
2. **Najahi (KSA)** – Focused on K–12 tutoring, but limited scalability and absence of hybrid learning models.
3. **Preply (Global)** – Strong in online language tutoring but not tailored to local Saudi educational needs such as Qiyas or Tahsili exams.
4. **VIPKid (Global)** – Specializes in English learning for younger students but lacks subject variety and local exam preparation.
5. **Superprof (Global)** – Wide coverage of subjects and tutors, but minimal vetting process and inconsistent quality control.
6. **Wyzant (US-based)** – Large pool of tutors and strong subject variety, but weak focus on Middle East regional exams and hybrid safety.
7. **Traditional Tutoring Centers (Jeddah, Riyadh)** – Provide face-to-face tutoring but offer limited flexibility, no AI personalization, and weaker safety guarantees.

1.4.2 Existing Solutions and Gaps

Current solutions fall into three broad categories:

- **Global online tutoring platforms** (e.g., Preply, VIPKid) – Focus on languages and generic subjects but do not address Saudi-specific exams.
- **Regional EdTech apps** (e.g., Noon Academy, Najahi) – Emphasize school subjects and national curricula but often lack deep personalization, robust verification, and hybrid flexibility.
- **Traditional tutoring centers** – Effective for exam prep but constrained by physical location, rigid scheduling, and limited data-driven monitoring.

Across these solutions, several critical gaps were identified:

- Limited or no rigorous tutor verification (ID checks, certifications, background checks).
- Weak support for hybrid models that safely combine online and in-person tutoring.
- Limited use of AI for tutor matching and adaptive learning analytics.
- Insufficient support for learners with special needs.
- Lack of strong engagement features such as gamification and long-term progress tracking.

1.4.3 Feature Comparison

Feature	Noon	Preply	Traditional	Wyzant	Superprof	Our Project
Verified tutors (ID & degrees)	✗	✗	✗	✗	✗	✓
AI tutor matching	✗	✗	✗	✗	✗	✓
Hybrid (online & in-person)	✓	✗	✓	✗	✗	✓
Local exam prep (Qiyas/Tahsili)	✓	✗	✓	✗	✗	✓
Special needs support	✗	✗	✗	✗	✗	✓
Gamification/engagement tools	✗	✗	✗	✗	✗	✓
Global exam prep (IELTS, SAT)	✗	✓	✗	✓	✓	✓
Rating and review	✓	✓	✗	✓	✓	✓

Table 1.1: Feature comparison between existing platforms and the Tutor Uberization Platform

1.4.4 Customer Feedback and Stakeholder Views

Feedback from parents, students, and tutors on existing platforms highlights:

- **Positive:** Convenience of finding tutors online; time and cost savings compared to travelling to centers; flexible scheduling and broad subject coverage.
- **Negative:** Concerns about unverified tutors and variable teaching quality; limited exam preparation for Qiyas and Tahsili in global platforms; motivation drops after a few sessions; weak personalization and follow-up.

Key stakeholder groups and their priorities include:

- **Students:** Need flexible, engaging support tailored to their curriculum and exam goals.
- **Parents/Guardians:** Seek trusted, verified tutors, transparent prices, and clear progress updates.
- **Tutors:** Want fair pay, reliable bookings, and tools to manage their schedule and reputation.
- **Schools & Universities:** Look for structured academic support and progress evidence for their students.
- **Regulators and Government Agencies:** Focus on safety, PDPL compliance, and quality assurance.
- **Investors and Partners:** Care about scalability, market growth, and strong retention metrics.

1.4.5 Market Opportunity and Differentiation

The market study shows a strong opportunity for a localized platform that combines:

- Rigorous tutor verification and background checks.
- Hybrid tutoring (secure face-to-face and online) in one app.
- AI-driven tutor matching and adaptive learning analytics.
- Built-in support for learners with special needs.
- Gamified engagement and long-term progress tracking.

The Tutor Uberization Platform is differentiated by directly targeting Saudi and MENA learners, focusing on Qiyas/Tahsili and global exams, and embedding trust, safety, localization, and personalization into its core design rather than as add-ons.

1.4.6 High-Level Go-to-Market Plan

- **Phase 1 (MVP, 3–4 months):** Launch with a curated group of verified tutors in major cities (e.g., Jeddah and Riyadh); enable core flows: search → book → pay → review.

- **Phase 2 (Scale, 6–9 months):** Add Qiyas/Tahsili prep bundles, partner with schools and universities, introduce loyalty and referral programs.
- **Phase 3 (Optimization, 12+ months):** Expand to GCC countries, add deeper gamification and advanced analytics, strengthen partnerships with certification bodies.

Key early metrics include: number of verified tutors onboarded, first-session completion rate, repeat bookings (Day 30 / Day 90), Net Promoter Score (NPS), dispute and refund rates, and Customer Acquisition Cost (CAC) versus Lifetime Value (LTV).

1.5 References

- Saudi Ministry of Education. *Education Statistics 2023*.
- ResearchAndMarkets. *Saudi Arabia Private Tutoring Market Report 2024–2033*.
- Preply Official Website. <https://preply.com>
- Noon Academy. <https://www.learnatnoon.com>
- Wyzant Official Website. <https://www.wyzant.com>
- Superprof Official Website. <https://www.superprof.com>
- Internal project documents: Business Model Canvas; User Personas; Requirements and UML Models; Architectural Design and Design Patterns.

Chapter 2

Overall Description

2.1 System Perspective

The Tutor Uberization Platform (TUP) is designed as a cloud-hosted SaaS solution. It interacts with users (students, parents, tutors, managers), external services (payment gateways, notification systems), and partner institutions.

System Context Overview:

- **Frontend:** Mobile App (iOS/Android), Browser-based Web App.
- **Backend:** RESTful API, authentication and access control, business logic.
- **Database:** Centralized relational DB storing users, bookings, payments, and logs.
- **External Systems:** Payment services, SMS/email notification providers, ID verification APIs.

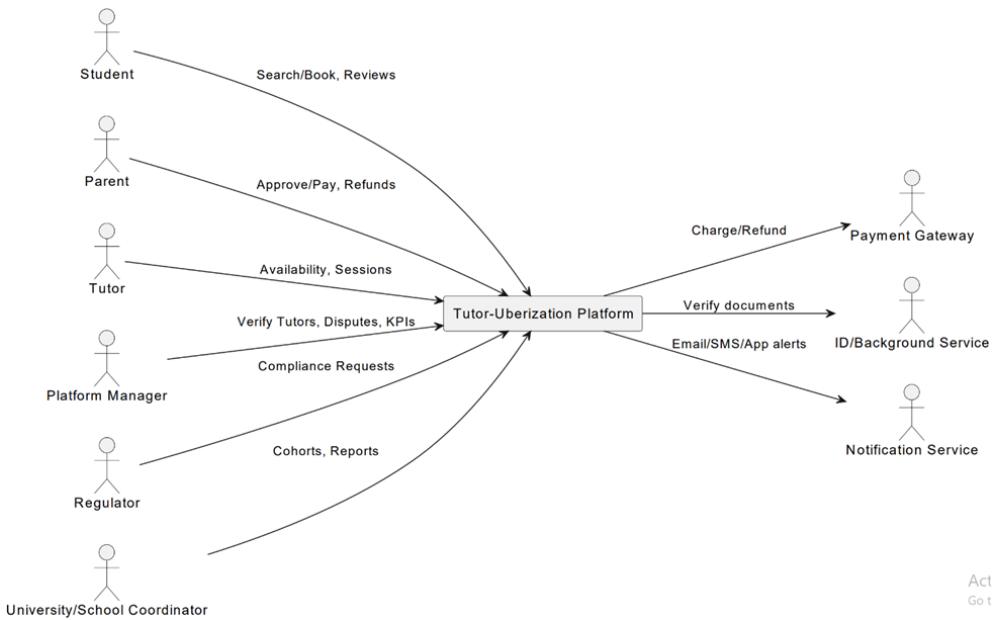


Figure 2.1: System Context Diagram – Tutor Uberization Platform

The system supports future scalability, including adding smart analytics, loyalty points, and institutional integrations.

2.2 User Characteristics

Based on persona analysis:

- **K–12 Learners:** Simple UI, gamified engagement, shorter task flow.
- **University Students:** Flexibility in time and budget, detailed tutor profiles.
- **Parents:** Trust and safety, transparent pricing, receipts.
- **Tutors:** Professional tools to build reputation, fair payouts.
- **Special Needs Students:** Calm interfaces, visual cues, structured steps.
- **Managers:** Dashboards, quality oversight, dispute resolution.
- **Regulators/Institutions:** Compliance monitoring, statistical KPIs.

2.3 Constraints

- **Regulatory:** Must comply with Saudi PDPL and payment security standards.
- **Technical:** Must operate under limited mobile network bandwidth.
- **Safety:** Parental approval required for minors' face-to-face sessions.
- **Financial:** Gateway fees affect revenue margins; offline refunds must be restricted.

2.4 Assumptions and Dependencies

- Tutors provide accurate educational background documents.
- Internet connection available for online sessions.
- Third-party APIs are reliable and available.
- External providers (SMS, payment) maintain their security policies.

Chapter 3

Functional Requirements

This chapter presents a **detailed list of what the Tutor Uberization Platform must do**. Each functional requirement describes observable behaviour of the system from the point of view of its external actors (students, parents, tutors, platform managers, partners, or regulators). The focus is on *what* the system must provide — such as searching for tutors, booking sessions, processing payments, sending reminders, or generating reports — rather than *how* these behaviours will be implemented internally.

In this SRS, the functional requirements are organised and expressed using **use cases**, **user stories**, and **functional modules**, consistent with the course template and Lab 4 deliverables. Each goal-level use case (e.g., “Book Tutoring Session”, “Manage Payment and Refunds”, “Tutor Verification and Compliance”) is supported by one or more SMART requirements (RQ0–RQ13). The requirements are written so that they can be traced to concrete test cases: for every requirement we define inputs, expected system behaviour, and clear acceptance criteria in a Given/When/Then style.

The remainder of this chapter is structured as follows:

- A set of **SMART functional requirements** (RQ0–RQ13) with acceptance criteria and priority.
- A **use case list** that groups related requirements into goal-level interactions.
- A **traceability matrix** connecting requirements to use cases.
- **Brief and fully dressed use case descriptions** for the most important flows (e.g., booking and payment).

3.1 Requirement Summary Table

ID	Requirement Description	Acceptance Criteria
RQ0	User authentication and profile creation	Successful login/registration with valid credentials; errors shown for invalid attempts
RQ1	Search verified tutors using filters	Returns matching verified tutors within 2 seconds
RQ2	Book a tutoring session	Booking created with confirmation and reminders
RQ3	Secure payment processing	Payment completes or fails with clear feedback
RQ4	Generate digital receipt	Receipt stored and viewable anytime
RQ5	Tutor availability management	Prevent overlapping availability
RQ6	Conduct session (online/face-to-face)	Record attendance and meeting link or location
RQ7	Tutor verification workflow	Approval stored, visible badge, and logs
RQ8	Compliance reporting	Export CSV/PDF logs for managers
RQ9	Ratings and reviews	Average updates and suspicious reviews flagged
RQ10	Disputes	Resolve within admin panel with refund API integration
RQ11	AI tutor recommendations	Ranked suggestions available for each search
RQ12	Notifications and reminders	Logs for reminder dispatch are stored
RQ13	KPI dashboards for partners	Visual analytics by time period

3.2 Main Use Cases

For this lab, the main goal-level use cases are:

- UC1: Book Tutoring Session.
- UC2: Manage Payment and Refunds.
- UC3: Manage Tutor Availability and Sessions.

- UC4: Tutor Verification and Compliance.
- UC5: Review and Rating System.
- UC6: Partner and Regulator Reporting.

Short descriptions of these use cases are given in this chapter; the fully dressed templates and their related UML diagrams (use case, activity, sequence, and state where applicable) are documented in Chapter 6.

3.3 Sustainable Development Goals (SDG) Alignment

This platform supports Saudi Vision 2030 digital transformation goals and the Kingdom's **Human Capability Development Program**, which aims to enhance learning opportunities for all citizens. The system is designed to align directly with two key United Nations Sustainable Development Goals (SDGs): **SDG 4: Quality Education** and **SDG 10: Reduced Inequalities**.



Figure 3.1: Alignment with SDG 4, SDG 10, and Saudi Vision 2030

SDG 4 – Quality Education: The platform expands access to verified tutors and specialized exam support (Qiyas, Tahsili, IELTS, SAT), helping students improve academic performance and future opportunities.

How the system supports SDG 4:

- UC1 & UC2 – Faster access to safe and qualified tutors.
- UC4 – Hybrid tutoring ensures learning continuity online and in-person.
- UC5 – Transparent ratings strengthen teaching quality.
- UC8 – Data insights improve educational outcomes.

SDG 10 – Reduced Inequalities: The system ensures equal educational access across different regions, income levels, and learner needs, including students with disabilities or limited local resources.

How the system supports SDG 10:

- UC1 – Inclusive search supports underserved regions and subjects.
- UC3 – Flexible scheduling for busy families and working students.
- UC7 – Fair policies reduce drop-out risk due to emergencies.
- UC8 – KPI reporting identifies inequities for improvement.

In conclusion, the Tutor Uberization Platform contributes to national sustainability priorities by promoting **high-quality, inclusive, and equitable education** while reducing **access barriers, geographic inequality, and safety concerns** in private tutoring.

3.4 Requirements–Use Case Traceability Matrix

Req. ID	Priority	UC1	UC2	UC3	UC4	UC5	UC6
RQ1	M	✓	✗	✗	✗	✗	✗
RQ2	M	✓	✗	✗	✗	✗	✗
RQ3	C	✗	✗	✓	✗	✗	✗
RQ4	S	✓	✗	✗	✗	✗	✗
RQ5	M	✓	✓	✗	✗	✗	✗
RQ6	S	✗	✓	✗	✗	✗	✗
RQ7	M	✗	✓	✗	✗	✗	✗
RQ8	M	✗	✗	✗	✓	✗	✗
RQ9	M	✗	✗	✗	✓	✗	✗
RQ10	M	✗	✗	✗	✗	✓	✗
RQ11	S	✗	✗	✗	✗	✓	✗
RQ12	S	✓	✗	✗	✗	✓	✗
RQ13	S	✗	✗	✗	✓	✓	✓

Table 3.1: Traceability matrix mapping Requirements (RQ1–RQ13) to Use Cases (UC1–UC6).

Chapter 4

Non-Functional Requirements

4.1 Performance Requirements

- 95% of search requests respond in ≤ 2 seconds.
- Login, payment, and booking events must respond in less than 3 seconds.
- The system must support 1000 concurrent users in the MVP.

4.2 Security Requirements

- Data encrypted with AES-256 at rest and TLS 1.3 in transit.
- Role-based access control (student, tutor, parent, admin).
- Payment API must follow PCI-DSS compliance.
- No resource may leak personal data in logs.

4.3 Reliability and Availability

- Booking and payment uptime must be at least 99.5%.
- Session reminders must retry 3 times before finally failing.

4.4 Usability and Accessibility

- Mobile-first, bilingual Arabic/English.
- Accessibility aligned with WCAG AA guidelines for special needs students.
- All core flows should be completed in 4 steps or fewer.

4.5 Maintainability and Scalability

- Modular architecture ready for horizontal scaling.
- Future AI analytics components deployable independently.

Chapter 5

System Features

5.1 Account Management

Inputs: Name, email/phone, role, password, preferences.

Processing: Validate and encrypt data.

Outputs: Profile created and stored in the database.

5.2 Tutor Search & Recommendation

Inputs: Subject, level, city, mode.

Processing: Filter tutors and apply AI-based ranking.

Outputs: Tutor list ordered by relevance.

5.3 Booking and Scheduling

Inputs: Tutor ID, selected time.

Processing: Conflict detection and reservation lock.

Outputs: Booking ID and reminders.

5.4 Payments, Refunds, Receipts

Inputs: Card or online wallet information.

Processing: Payment API authorization and optional refunds.

Outputs: Receipt and updated booking status.

5.5 Session Attendance

Inputs: Session join action.

Processing: Log timestamps and verify presence.

Outputs: Attendance record.

5.6 Reviews and Ratings

Inputs: Stars (1–5) and comment.

Processing: Publish moderated review and update averages.

Outputs: Updated average rating displayed on tutor profile.

5.7 Verification and Compliance

Inputs: IDs and certifications.

Processing: Approval workflow by managers.

Outputs: Verified badge and compliance logs.

Chapter 6

System Models

This chapter presents a set of models that describe how the Tutor Uberization Platform behaves at runtime and how the data is structured. The models are organised around the main **use cases**. For each key use case we provide:

- A **use case diagram** showing the actors and high-level interaction.
- An **activity diagram** outlining the internal flow of actions and data.
- A **sequence diagram** (where applicable) to show message exchange between objects over time.
- A **state machine diagram** (where applicable) to describe how core entities such as *Booking* change state during the use case.

6.1 Use-Case-Centred Behavioural Models

6.1.1 UC1 – Book Tutoring Session

This model cluster captures how a student or parent books a tutoring session with a verified tutor. The use case is tightly connected to requirements RQ1 (search), RQ2 (booking), RQ5 (availability), RQ6 (session), and RQ12 (notifications). The diagrams show the journey from selecting a tutor, choosing a time slot, and confirming a booking, while ensuring that double-booking is prevented and reminders are scheduled.

Detailed Use Case Description (Style A)

Name: UC1 – Book Tutoring Session

Primary Actor: Student or Parent

Stakeholders:

- **Student/Parent:** Wants an appropriate tutor at a convenient time.
- **Tutor:** Wants a clear, conflict-free schedule.
- **Platform Manager:** Wants reliable bookings and accurate usage metrics.

Preconditions:

- The user (student or parent) is authenticated in the system.
- At least one verified tutor with available time slots exists.

Postconditions:

- A new booking is created with status *Pending Payment* or *Paid*.
- The selected time slot is locked to avoid double-booking.
- A confirmation notification is scheduled or sent to both student/parent and tutor.

Main Success Scenario:

1. User opens the tutor search results and selects a tutor profile.
2. System displays tutor details and the availability calendar.
3. User selects the subject, mode (online/face-to-face), and a free time slot.
4. User reviews price, duration, and session details and clicks “Confirm Booking”.
5. System validates that the chosen slot is still free and creates the booking.

6. System updates the tutor's availability to show the slot as reserved.
7. System sends or schedules a booking confirmation notification and session reminder.

Extensions:

- 2a. *No availability*: If the tutor has no free slots, the system suggests alternative tutors or times.
- 4a. *Slot conflict*: If the slot becomes unavailable during confirmation, the system rejects the booking and shows other free slots.
- 7a. *Notification failure*: If the notification service is down, the system logs the error and retries later; booking still succeeds.

Related Requirements: RQ1, RQ2, RQ5, RQ6, RQ12.

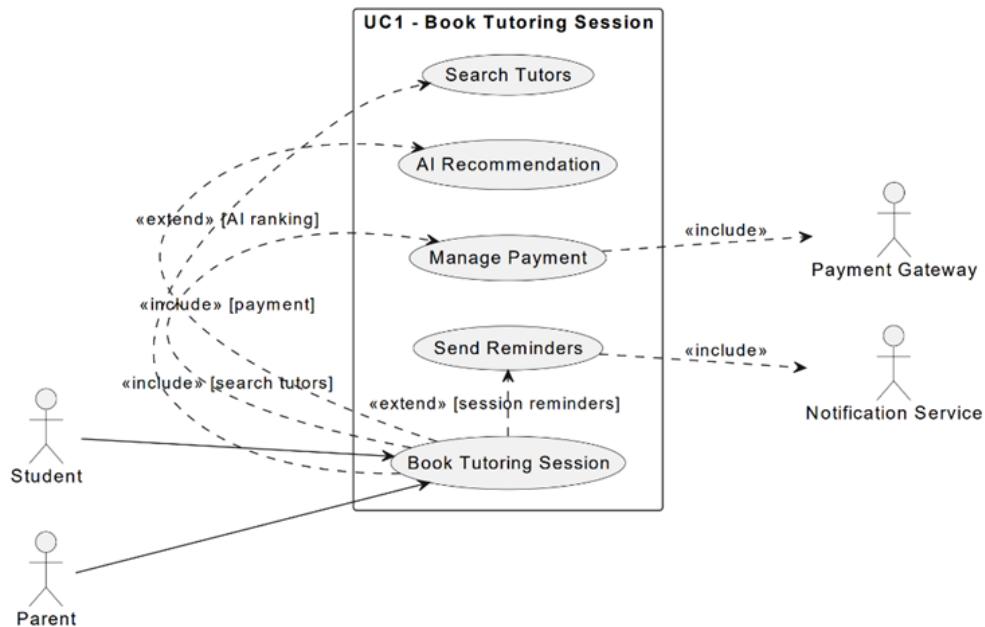


Figure 6.1: UC1 – Book Tutoring Session (Use Case Diagram)

The activity diagram below refines this behaviour into a step-by-step workflow starting from the *Search/Select Tutor* action and ending when the booking is stored and confirmation is sent.

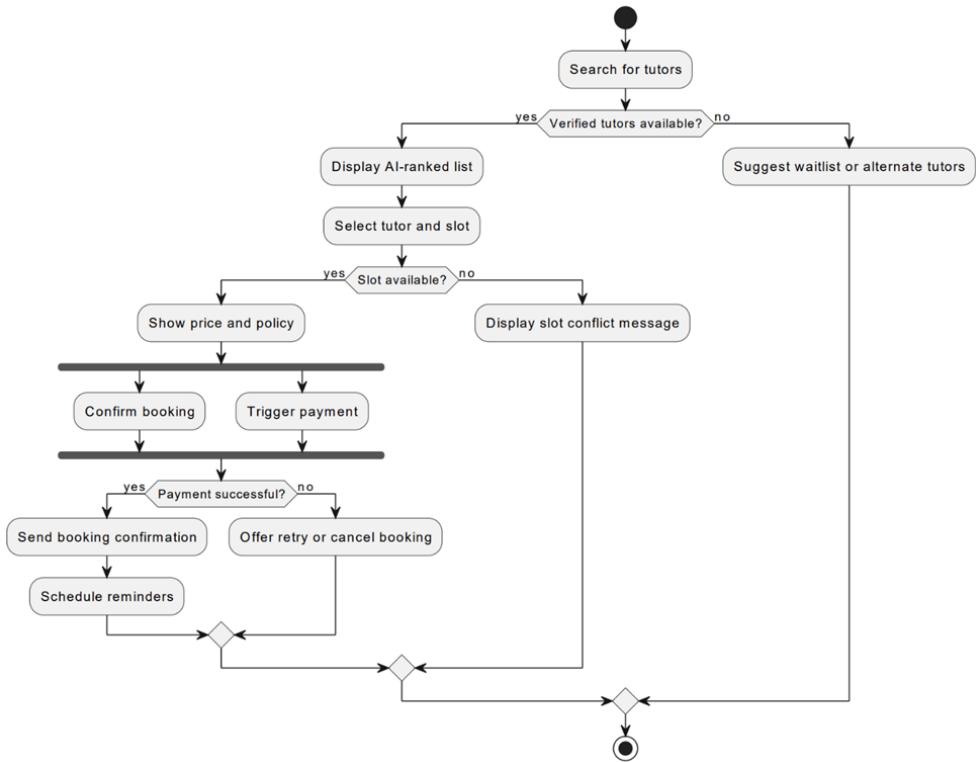


Figure 6.2: Activity Diagram – Book Tutoring Session

The sequence diagram focuses on the interaction between key objects (Student/Parent, UI, Booking Service, Availability Service, Notification Service). It highlights how the system locks the selected slot, creates the booking, and schedules notifications.

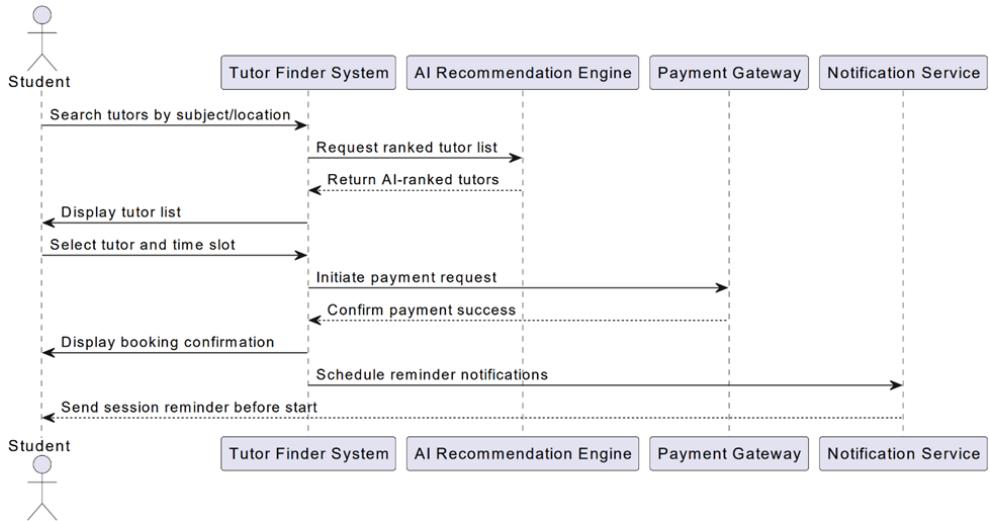


Figure 6.3: Sequence Diagram – Booking and Session Reminder

Finally, the booking state machine below shows how a booking progresses from initial creation to completion and rating. This lifecycle is triggered by UC1 and then extended by other use cases such as payment and cancellation.

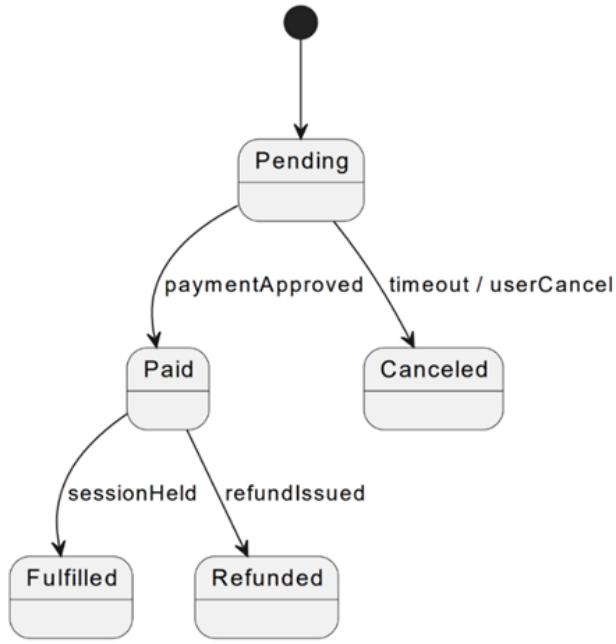


Figure 6.4: State Diagram – Basic Booking Lifecycle

6.1.2 UC2 – Manage Payment and Refunds

UC2 covers secure payment processing and optional refunds for existing bookings. It is mainly related to requirements RQ3 (payment), RQ4 (receipt generation), and RQ10 (disputes and refunds). The models describe how the system interacts with the payment gateway, updates booking status, and records financial transactions.

Detailed Use Case Description (Style A)

Name: UC2 – Manage Payment and Refunds

Primary Actor: Parent (payer) or adult Student

Stakeholders:

- **Student/Parent:** Wants a safe and simple checkout.
- **Tutor:** Wants timely payment for completed sessions.
- **Platform/Finance:** Needs accurate payment and refund records.

Preconditions:

- A booking with status *Pending Payment* exists.
- The payment gateway is online and reachable.

Postconditions:

- On success, booking status is *Paid* and a receipt is stored.
- On failure, booking remains *Pending Payment* with an error message.
- For approved refunds, payment and booking records are updated accordingly.

Main Success Scenario (Payment):

1. User opens the details of a pending booking and clicks “Pay Now”.
2. System shows the total amount and available payment methods (Mada, Visa, Apple Pay).

3. User selects a method and provides payment details.
4. System sends a payment authorisation request to the external gateway.
5. Gateway approves the transaction.
6. System marks the booking as *Paid* and stores a payment record.
7. System generates and stores a digital receipt.
8. System notifies both tutor and student/parent that payment succeeded.

Alternative Flow (Refund):

9. Admin or authorised staff opens the booking in the admin panel and selects “Issue Refund”.
10. System calls the refund endpoint of the payment gateway.
11. If refund succeeds, system updates payment information and booking status to *Refunded*.

Extensions:

- 5a. *Payment declined*: System shows gateway error; user can retry or choose a different method.
- 5b. *Timeout*: System reports timeout; booking stays *Pending Payment*.
- 11a. *Refund failure*: System logs the error and informs admin to try again or escalate.

Related Requirements: RQ3, RQ4, RQ10, RQ12.

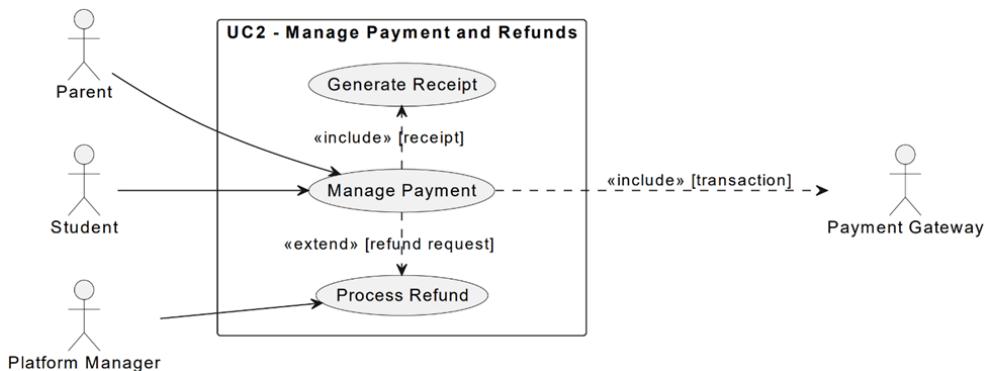


Figure 6.5: UC2 – Manage Payment and Refunds (Use Case Diagram)

The activity diagram details the internal flow from displaying the payable amount, through payment authorisation, to generating the digital receipt or handling errors.

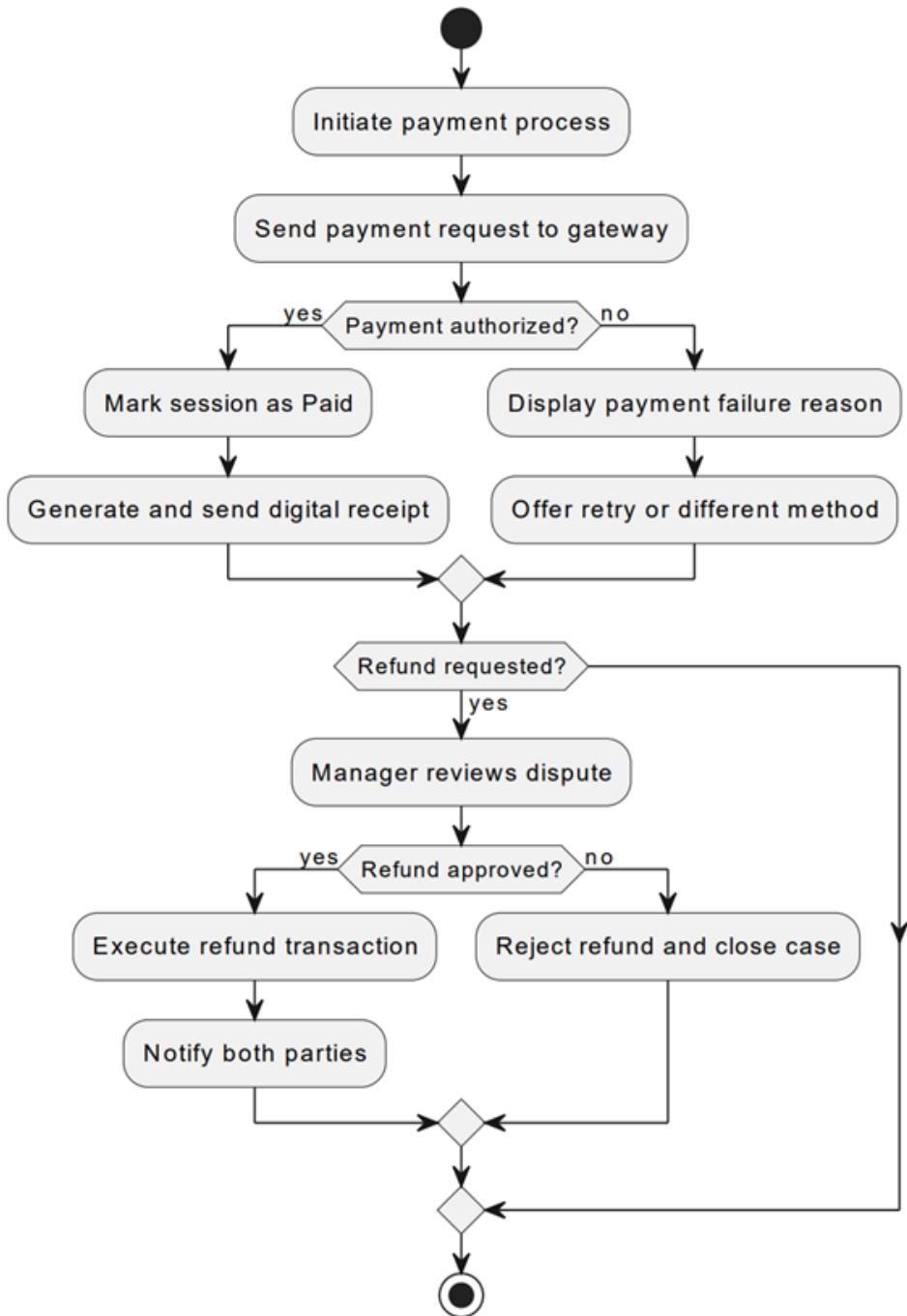


Figure 6.6: Activity Diagram – Payment and Refund Flow

The corresponding sequence diagram shows the messages exchanged between

the UI, Payment Service, external Payment Gateway, and Booking Service during both payment and refund operations.

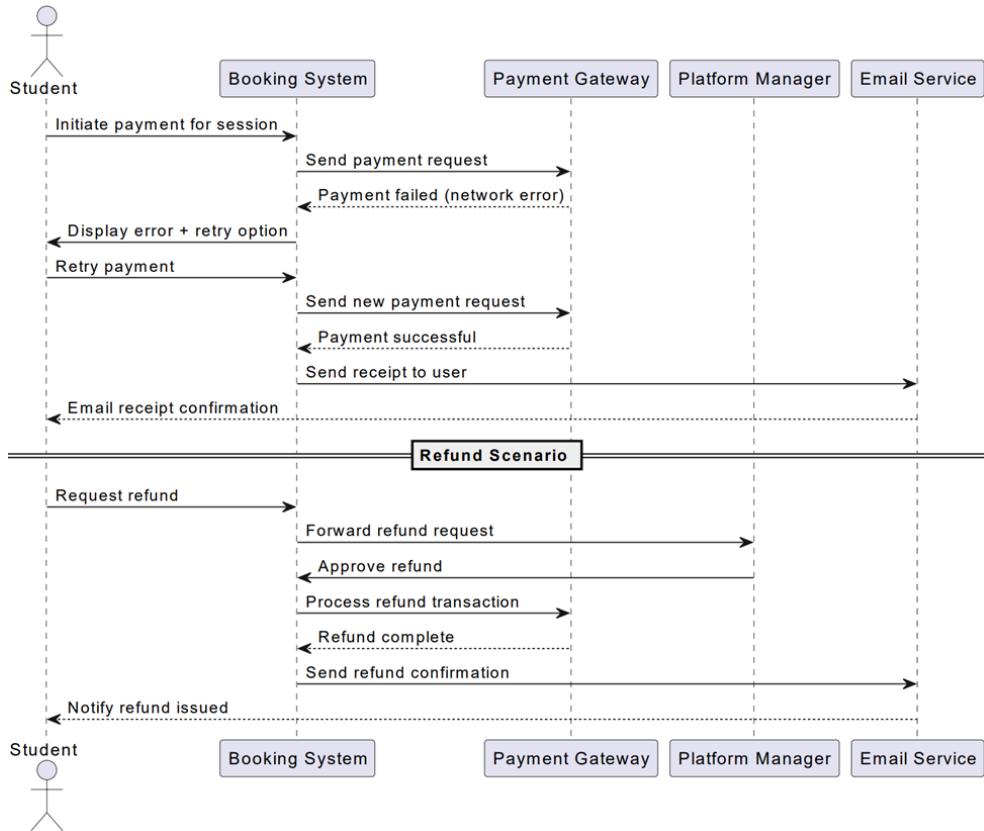


Figure 6.7: Sequence Diagram – Payment and Refund Process

From a lifecycle perspective, payments extend the booking state machine by adding transitions for *Paid*, *Refund Requested*, and *Refunded* states. The following state diagram focuses on these combined booking and refund transitions.

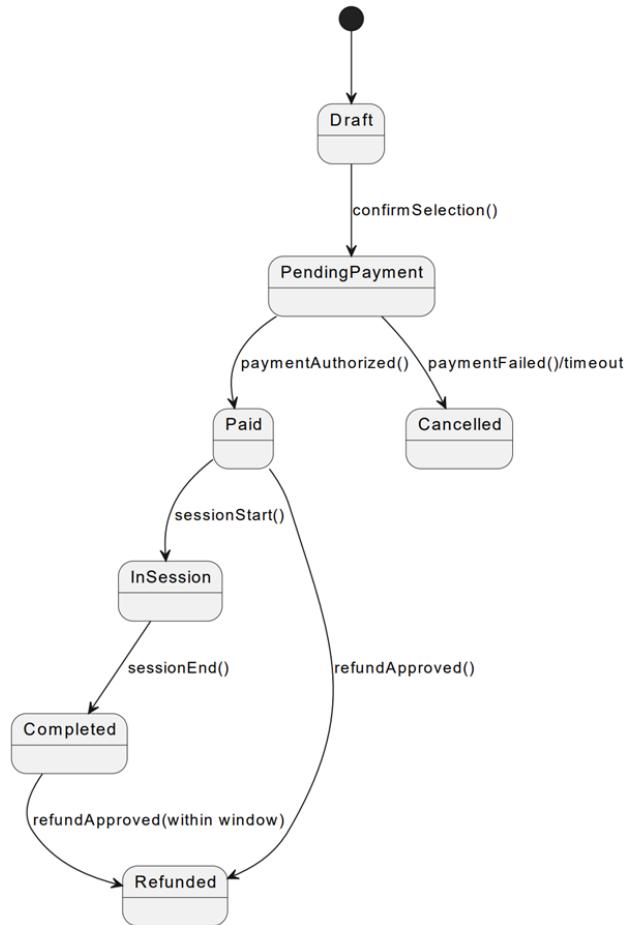


Figure 6.8: State Diagram – Full Booking and Refund Lifecycle

6.1.3 UC3 – Manage Tutor Availability and Sessions

UC3 models how tutors define and adjust their availability and how the system keeps availability consistent with existing bookings. It supports requirements RQ2 (booking) and RQ5 (availability) by ensuring that only valid time slots can be booked and that schedule changes are propagated to students.

Detailed Use Case Description (Style A)

Name: UC3 – Manage Tutor Availability and Sessions

Primary Actor: Tutor

Stakeholders:

- **Students/Parents:** Need reliable information about when tutors are free.
- **Platform Manager:** Wants consistent schedules and fewer booking errors.

Preconditions:

- Tutor is authenticated.
- Tutor profile exists in the system (ideally verified).

Postconditions:

- Availability slots are stored without overlaps.
- Any changes are reflected in search and booking flows.

Main Success Scenario:

1. Tutor opens the “Availability and Sessions” dashboard.
2. System displays a calendar view with existing sessions and free slots.
3. Tutor adds new availability slots or edits existing ones.
4. System checks that new availability does not conflict with confirmed bookings.
5. Valid changes are saved and immediately visible to UC1 (booking).

Extensions:

- 4a. *Overlap detected:* System rejects conflicting changes and highlights the problematic time range.
- 5a. *Upcoming session cancelled:* When a booking is cancelled, system frees the corresponding slot and updates the view for tutor and students.

Related Requirements: RQ2, RQ5, RQ6.

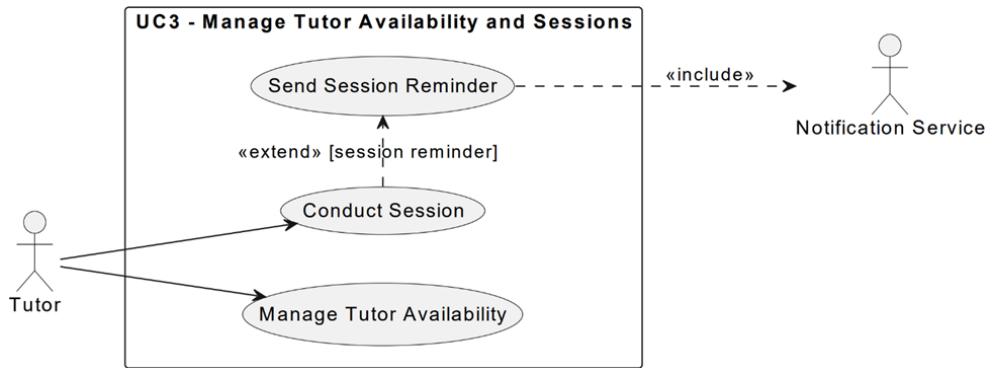


Figure 6.9: UC3 – Manage Tutor Availability and Sessions (Use Case Diagram)

The activity diagram explains how a tutor adds, edits, or deletes availability slots, and how the system validates conflicts with current bookings before accepting changes.

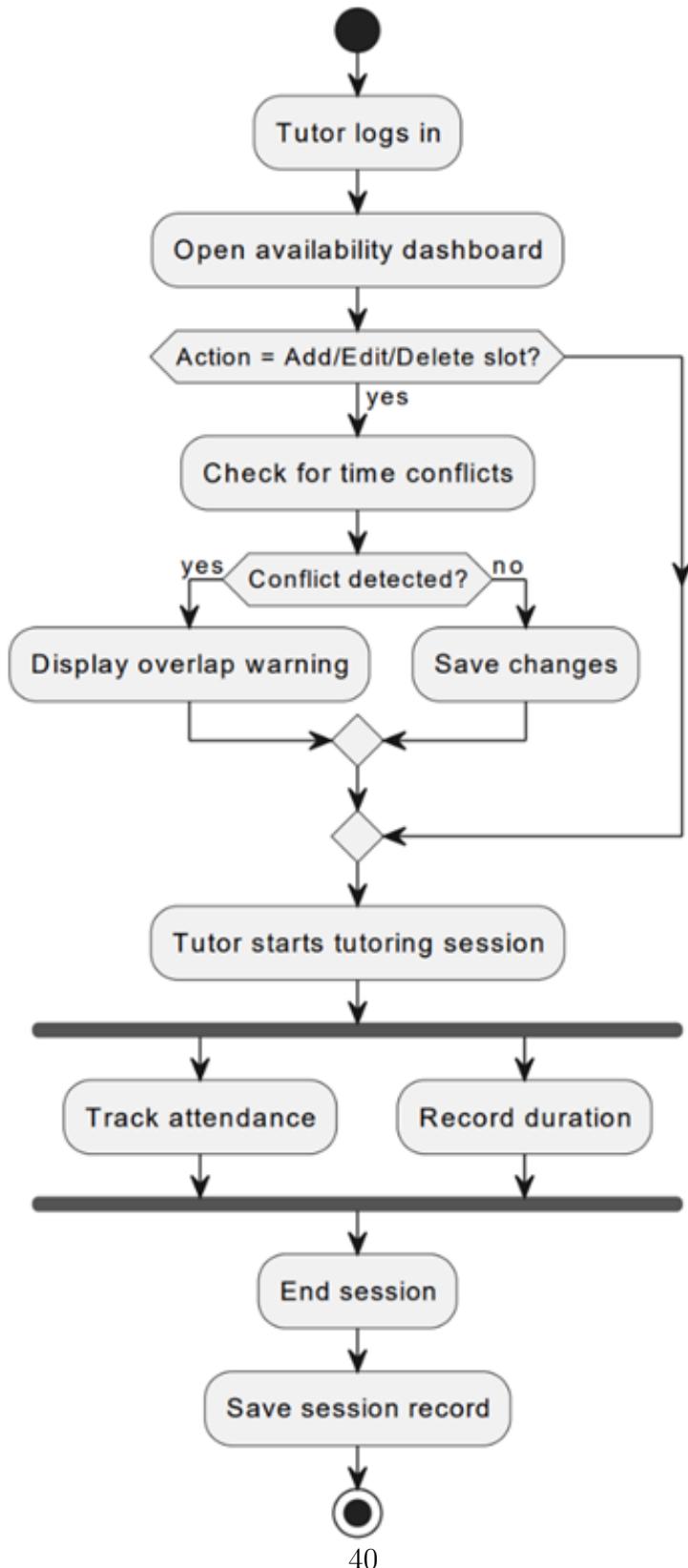


Figure 6.10: Activity Diagram – Tutor Availability and Sessions

6.1.4 UC4 – Tutor Verification and Compliance

UC4 covers the verification workflow that platform managers use to approve or reject tutors and to maintain compliance logs. It addresses requirements RQ7 (tutor verification) and RQ8 (compliance reporting), ensuring that only verified tutors appear as selectable in UC1 and UC2.

Detailed Use Case Description (Style A)

Name: UC4 – Tutor Verification and Compliance

Primary Actor: Platform Manager / Admin

Stakeholders:

- **Tutors:** Need a clear, fair verification process.
- **Students/Parents:** Rely on the verified badge for trust.
- **Regulators:** Require traceable verification decisions.

Preconditions:

- Tutor has submitted required documents (e.g., ID, certificates).
- Tutor account exists in the system.

Postconditions:

- Tutor is marked as *Verified*, *Rejected*, or *Needs More Info*.
- A compliance log entry records the decision, time, and responsible admin.

Main Success Scenario:

1. Admin opens the “Tutor Verification” panel.
2. System lists tutors with *Pending* verification status.
3. Admin reviews the submitted documents and checks them against policies.
4. Admin approves the tutor.

5. System updates tutor status to *Verified* and shows a badge on tutor profile.
6. System logs the decision for future compliance reports.

Extensions:

- 3a. *Insufficient documents*: Admin sets status to *Needs More Info*; system requests missing documents from tutor.
- 4a. *Tutor rejected*: Admin rejects tutor; system records reason and hides tutor from UC1 search.

Related Requirements: RQ7, RQ8, RQ13.

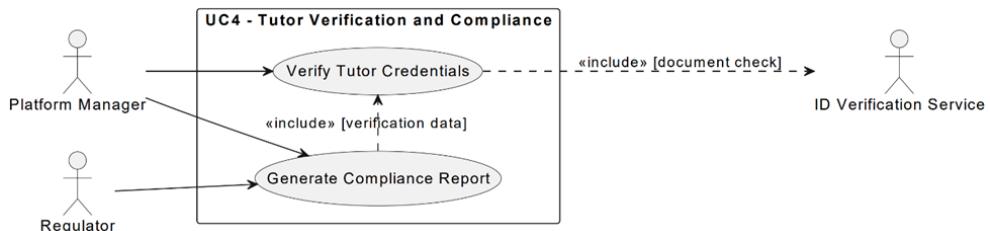


Figure 6.11: UC4 – Tutor Verification and Compliance (Use Case Diagram)

The activity diagram shows the steps of submitting documents, reviewing them, recording decisions, and updating the tutor's visible verification badge.

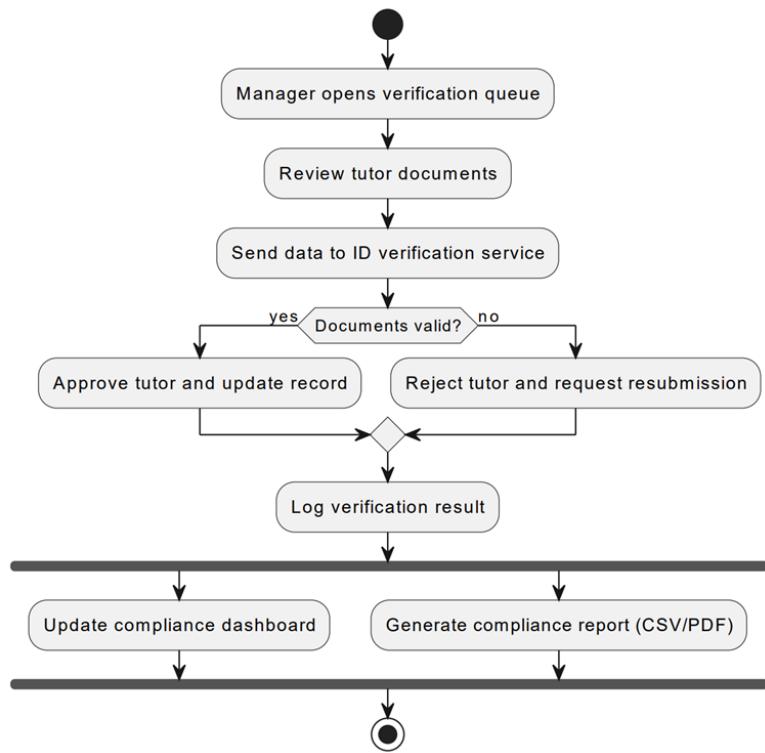


Figure 6.12: Activity Diagram – Tutor Verification Workflow

Sequence Diagram – Tutor Verification (UC4)

The sequence diagram in Figure 6.13 shows the full workflow of tutor verification, including submission, review, validation, and approval/rejection. It demonstrates the interaction between Tutor, Platform Manager, system backend, external ID API, and the database to ensure platform trust and safety.

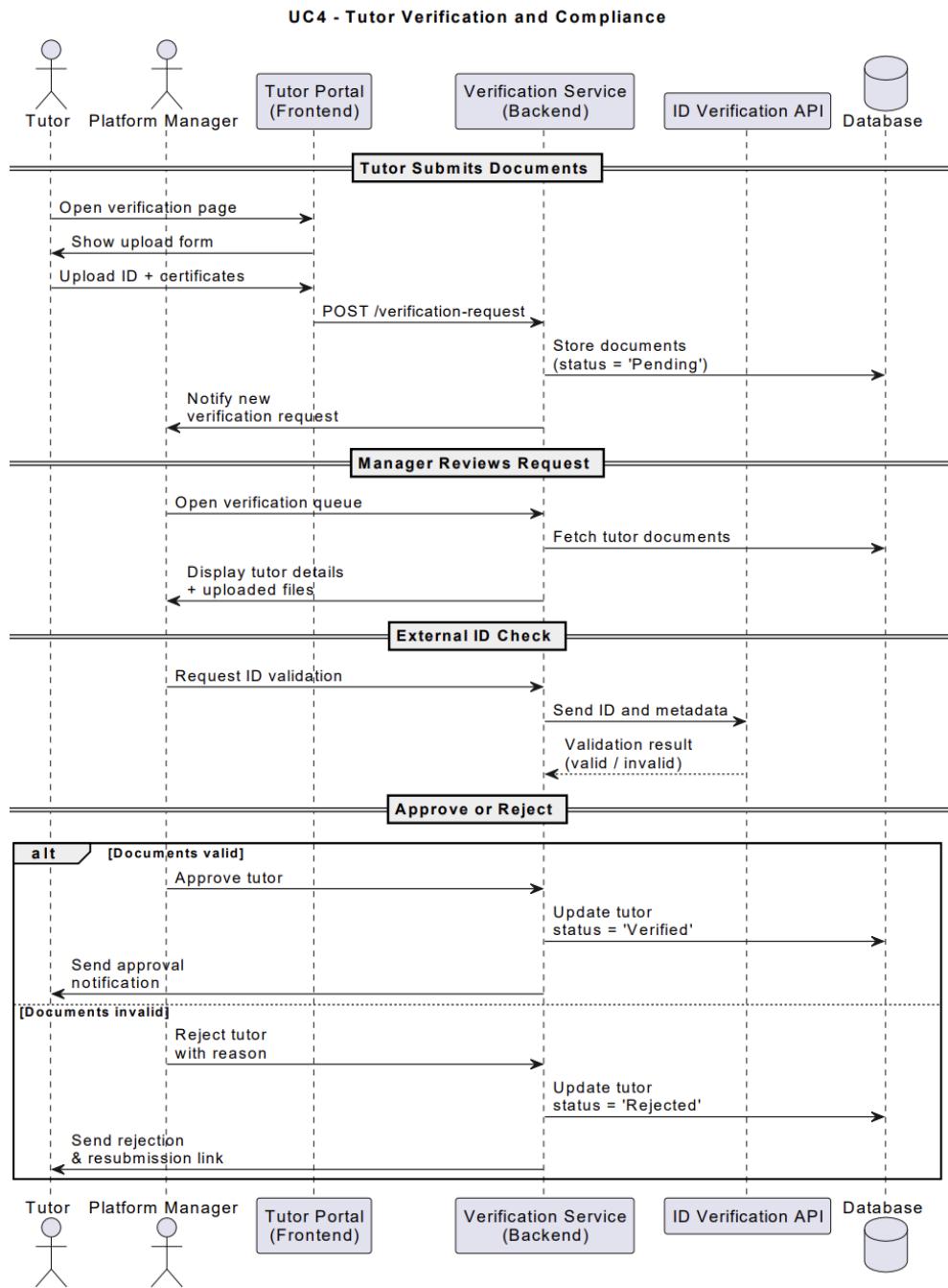


Figure 6.13: Sequence Diagram for UC4 — Tutor Verification Process

6.1.5 UC5 – Review and Rating System

UC5 models the post-session feedback process in which students or parents rate tutors and leave comments. It is primarily linked to RQ9 (ratings and reviews) and contributes data to RQ11 (AI tutor recommendations).

Detailed Use Case Description (Style A)

Name: UC5 – Review and Rating System

Primary Actor: Student or Parent

Stakeholders:

- **Tutors:** Want feedback to improve and build reputation.
- **Other Students/Parents:** Use reviews to choose tutors.
- **Platform Manager:** Uses reviews to monitor quality.

Preconditions:

- The booking status is *Completed*.

Postconditions:

- A review with rating and optional comment is stored.
- Tutor average rating is recalculated and displayed.

Main Success Scenario:

1. After a session, system prompts student/parent to rate the tutor.
2. User selects a rating (1–5 stars) and optionally writes a short comment.
3. System validates the input (non-empty rating, acceptable comment length).
4. System stores the review and updates the tutor's average rating.
5. Updated rating and new review appear on the tutor profile.

Extensions:

- 2a. *User skips review:* System allows skipping and may send a reminder later.
- 3a. *Offensive or suspicious content:* System flags review for moderation and hides it until approved.

Related Requirements: RQ9, RQ11.

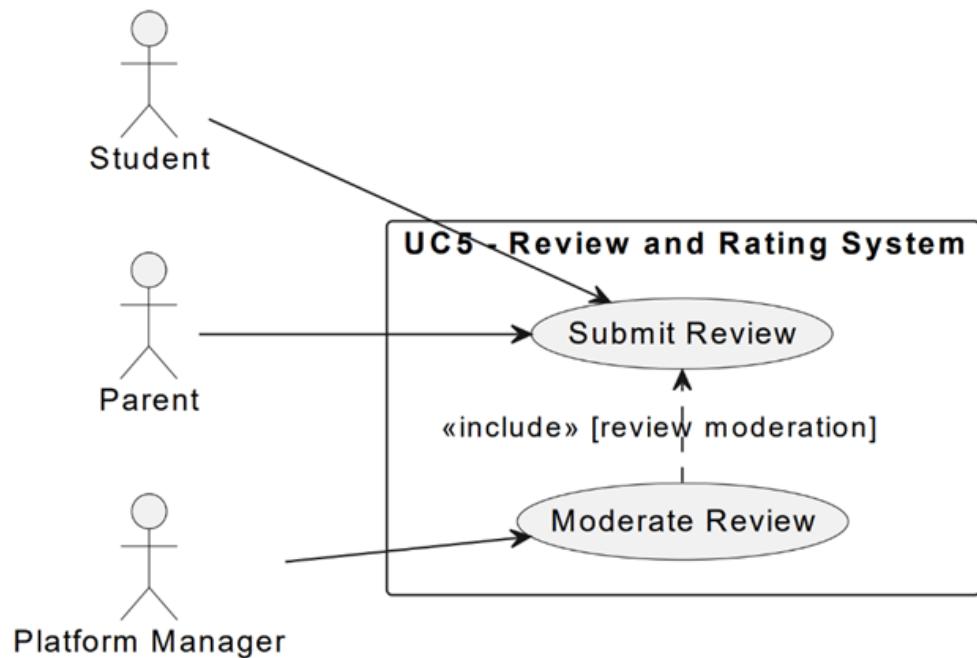


Figure 6.14: UC5 – Review and Rating System (Use Case Diagram)

The activity diagram describes how the platform prompts for reviews, validates the content, updates average ratings, and flags suspicious reviews for moderation.

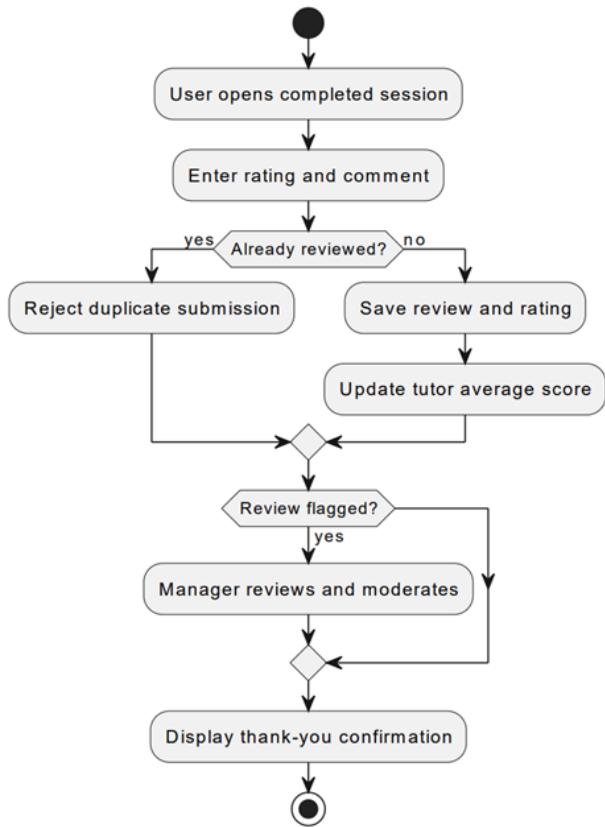


Figure 6.15: Activity Diagram – Review and Moderation Flow

6.1.6 UC6 – Partner and Regulator Reporting

UC6 focuses on generating KPI and compliance reports for schools, universities, and regulators. It supports RQ8 (compliance reporting) and RQ13 (dashboards for partners).

Detailed Use Case Description (Style A)

Name: UC6 – Partner and Regulator Reporting

Primary Actor: Platform Manager / Institution / Regulator

Stakeholders:

- **Partners (Schools/Universities):** Need insights into student usage and outcomes.

- **Regulators:** Need statistics for compliance and policy-making.
- **Investors/Management:** Monitor growth, revenue, and quality indicators.

Preconditions:

- Historical data exists for bookings, payments, and reviews.
- Actor has permission to view reporting dashboards.

Postconditions:

- A filtered report is generated and optionally exported (CSV/PDF).

Main Success Scenario:

1. Actor opens the “Reporting & KPIs” dashboard.
2. Actor selects a date range and optional filters (city, subject, institution).
3. System aggregates the relevant data from the database.
4. System displays charts and tables summarising key KPIs (e.g., number of sessions, revenue, average ratings).
5. Actor exports the report as CSV or PDF for sharing.

Extensions:

- 2a. *Invalid filter combination:* System displays validation messages and asks user to correct filters.
- 5a. *Export error:* System informs the user and logs the error for technical follow-up.

Related Requirements: RQ8, RQ13.

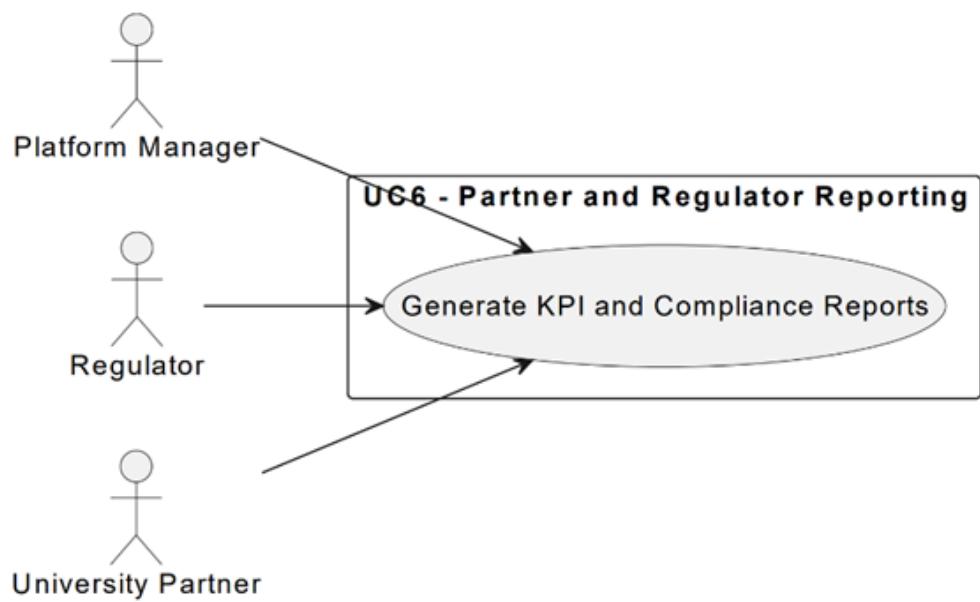


Figure 6.16: UC6 – Partner and Regulator Reporting (Use Case Diagram)

The related activity diagram explains how managers choose reporting periods and filters, how data is aggregated, and how the reports are exported to CSV/PDF formats.

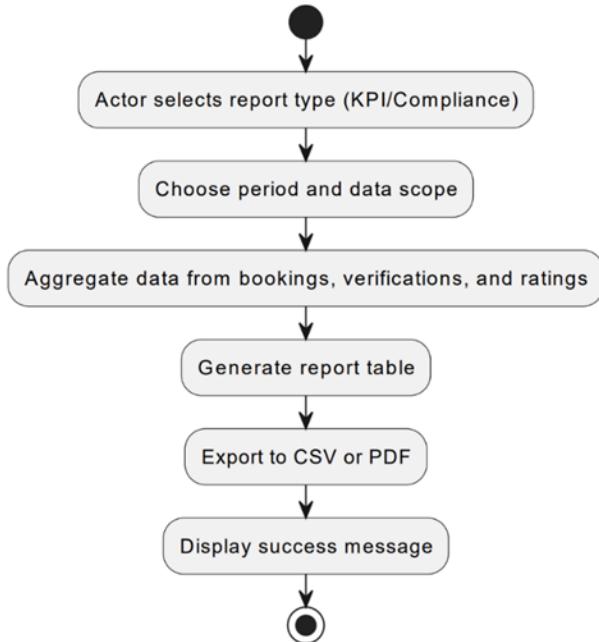


Figure 6.17: Activity Diagram – KPI and Compliance Reporting Flow

6.2 Domain Model (Class Diagram)

The domain model represents the main data structures and relationships that underpin all use cases. It defines entities such as *User*, *Tutor*, *Student*, *Parent*, *Booking*, *Payment*, *AvailabilitySlot*, *Review*, and others. These classes provide the persistent backbone required by the behavioural models described earlier.

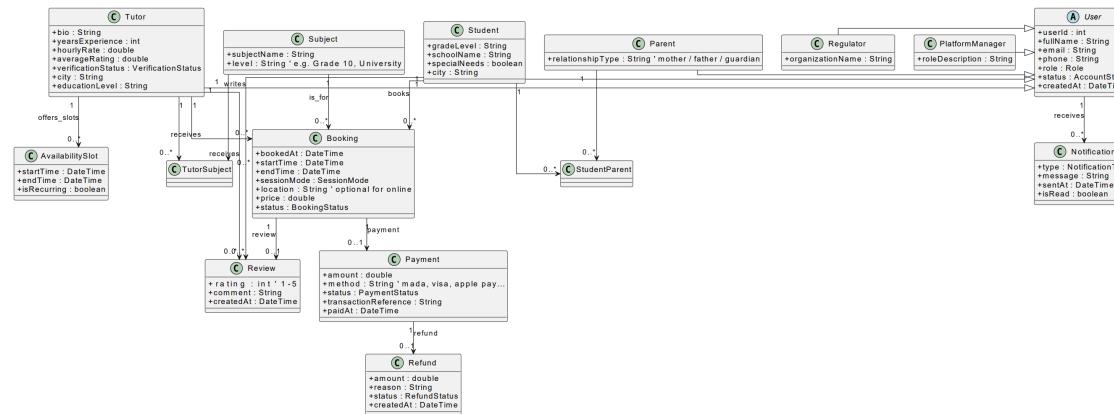


Figure 6.18: Domain/Class Diagram – Tutor Uberization Data Model

6.3 Entity Relationship Diagram (ERD)

The Entity Relationship Diagram refines the domain model into a database-oriented view. It focuses on primary keys, foreign keys, and cardinalities required to implement:

- User and role hierarchies (Student, Parent, Tutor, Manager, Regulator),
- Booking and Payment transactions,
- Availability management,
- Ratings, reviews, and notifications,
- Compliance and reporting data.

The ERD ensures that every use case (UC1–UC6) is backed by a consistent relational schema, and that constraints such as “one payment per booking” or “review only after completed session” can be enforced at the database level.

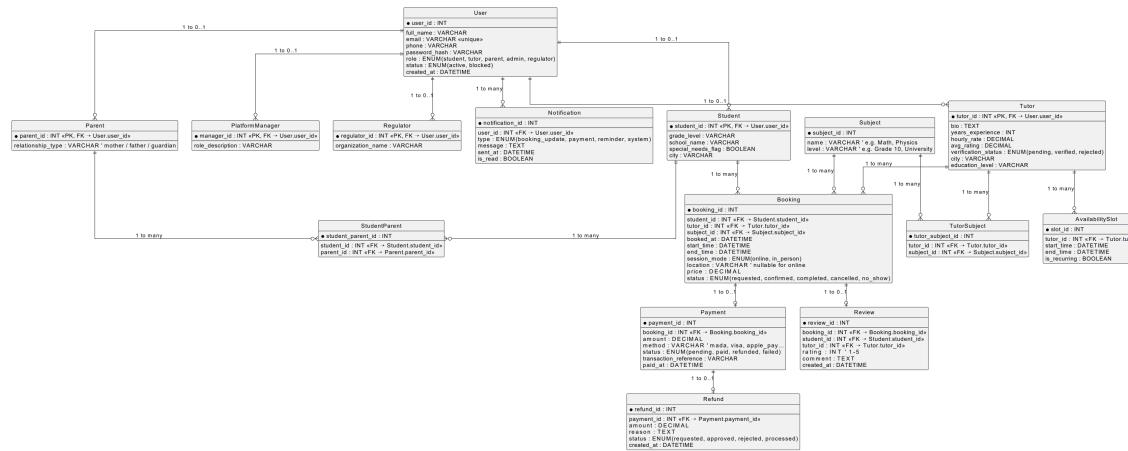


Figure 6.19: Entity Relationship Diagram – Tutor Uberization Platform

6.4 Entity Relationship Diagram (ERD)

This ERD describes the core data structure for the Tutor Uberization Platform, including primary entities such as User, Tutor, Student, Parent, Booking, Payment, Reviews, Subjects, and Availability. It defines the relationships, cardinalities, and database rules necessary to enforce data integrity and support essential platform use cases such as booking sessions, processing payments, submitting ratings, and generating compliance reports.

The ERD is derived from the functional requirements (RQ1–RQ13) and Use Cases (UC1–UC6) and ensures that every interaction in the system is backed by a valid data model. It incorporates super-type/sub-type relationships from the User hierarchy, many-to-many relationships for Parent–Student and Tutor–Subject associations, and transactional relations that maintain booking and payment history.

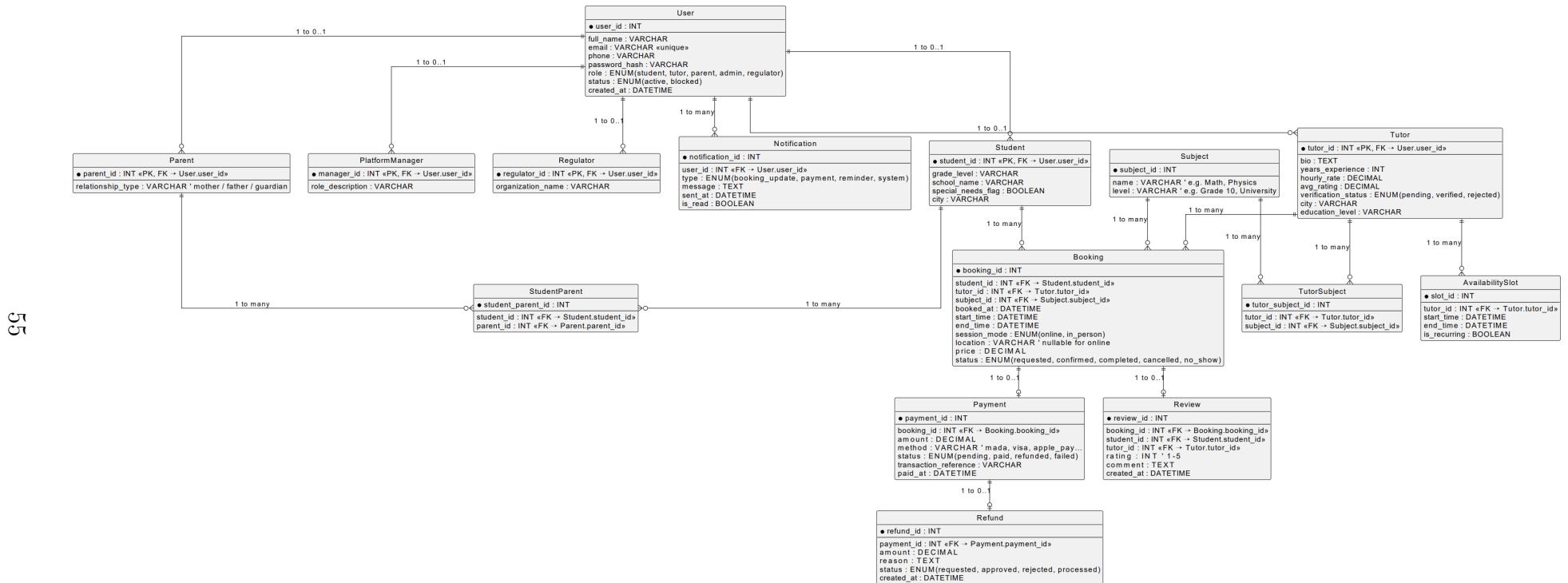


Figure 6.20: Entity Relationship Diagram – Tutor Uberization Platform

Key Entity Notes

- **User** is a super-entity with roles: Student, Parent, Tutor, Platform Manager, Regulator.
- **Booking** connects Student and Tutor with required Subject, price, and time.
- **Payment** is linked to one Booking; refunds extend from payments only.
- **AvailabilitySlot** ensures tutors can only be booked during valid time ranges.
- **Review** can only be submitted after a completed session.
- **Notification** logs reminders, alerts, and communications per user.
- **Compliance & KPI data** comes from booking, review, and session history.

This ERD supports traceability to the following core functions:

- Tutor search + filtering (RQ1)
- Booking and session lifecycle (RQ2, RQ5–RQ6, RQ12)
- Secure payments and refunds (RQ3–RQ4, RQ10)
- Verification, KPIs, and compliance reporting (RQ7–RQ8, RQ13)
- Ratings, reviews, and AI recommendations (RQ9, RQ11)

6.5 System Architecture

The architecture follows a layered design ensuring modularity, security, and scalability.

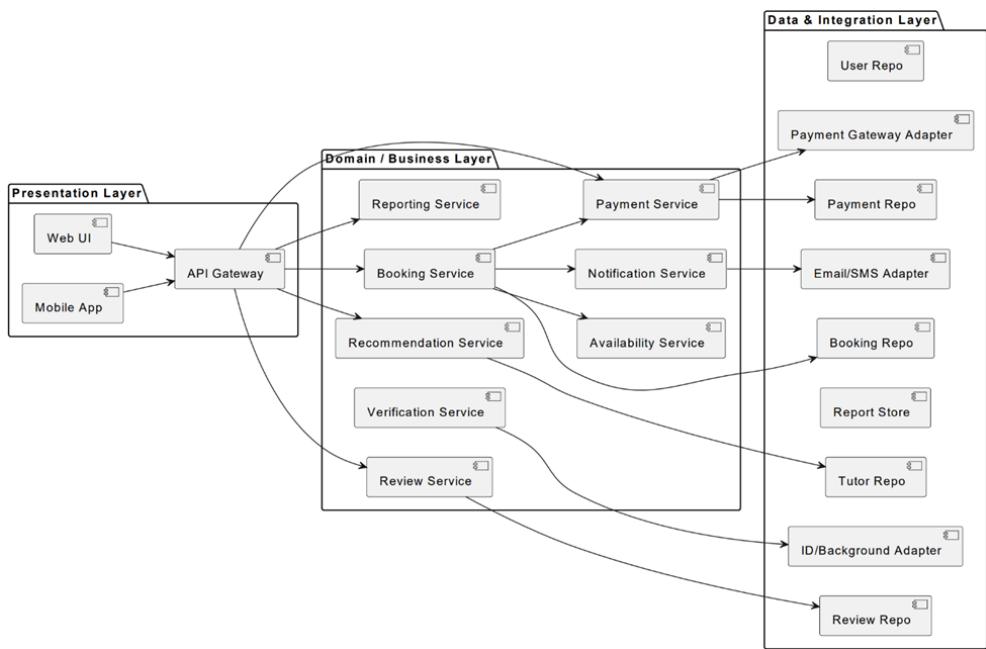


Figure 6.21: Layered Architecture for Tutor Uberization Platform

The platform includes:

- **Presentation Layer:** Mobile app and web client.
- **Application Layer:** API, session logic, analytics, notifications.
- **Data Layer:** SQL database and secure storage.

Chapter 7

Design Patterns

The system applies software design patterns to improve maintainability, reliability, and extensibility.

7.1 Observer Pattern – Session Notifications

Problem: Students and tutors must receive reminders and schedule changes in real time.

Solution: Observer links notification service to booking updates.

Participants:

- **Subject:** Booking.
- **Observers:** Student, Tutor.
- **Concrete Observer:** Notification Service.

Example Flow:

Booking status changes -> Notify() -> SMS / Email message sent

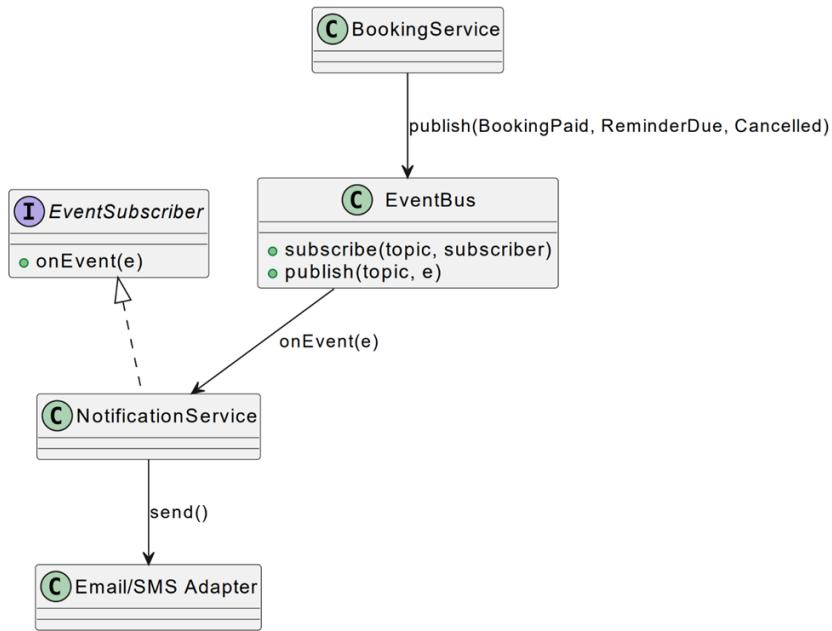


Figure 7.1: Observer Pattern applied to Booking Notifications

7.2 Strategy Pattern – Payment Methods

Problem: The system must support Mada, card payments, and future marketplaces without rewriting core logic.

Solution: Replaceable payment algorithms through Strategy.

Strategy Interface:

```
processPayment(amount, user, booking)
```

Concrete Strategies:

- MadaPaymentStrategy
- CreditCardPaymentStrategy

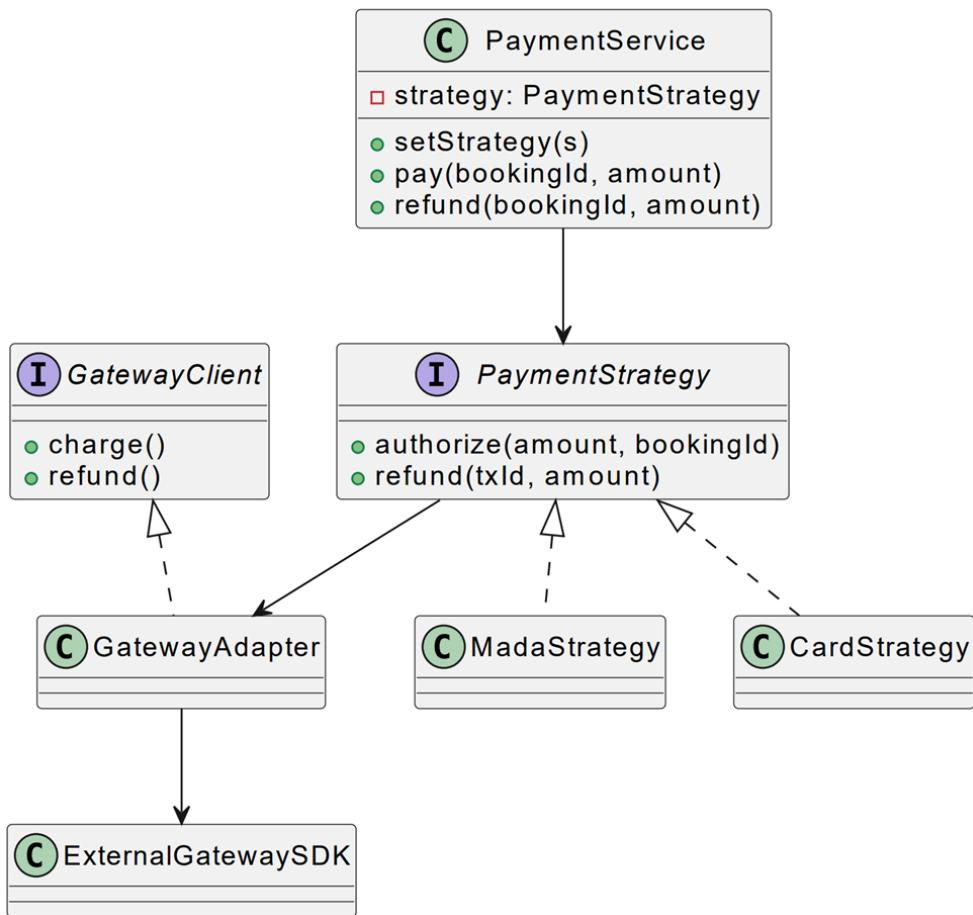


Figure 7.2: Strategy Pattern for Payment Methods

7.3 State Pattern – Booking Lifecycle

Problem: Booking behavior changes based on status (pending, paid, completed).

Solution: Each state implements rules and allowed transitions.

Pending -> Paid -> Completed -> Rated
 \-> Canceled

Benefits:

- Improves clarity in allowed actions.

- Reduces conditional logic in the booking class.

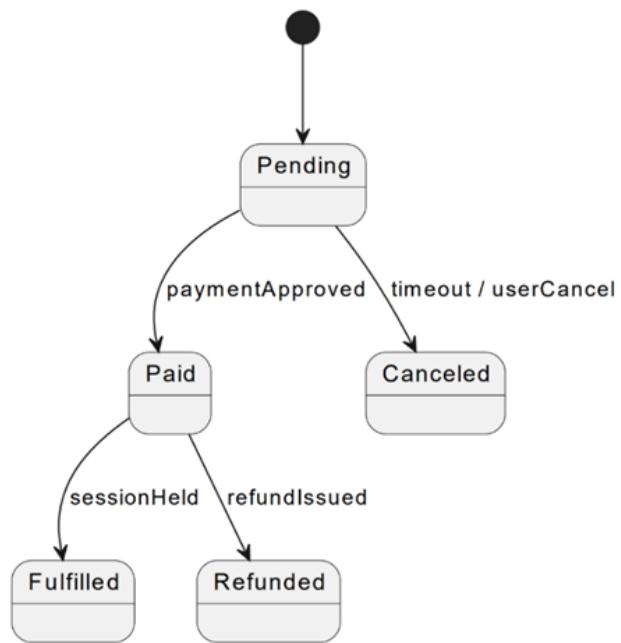


Figure 7.3: State Pattern in Booking Management

Chapter 8

User Interface / User Experience (UI/UX)

This section presents the main user interfaces of the Tutor Uberization Platform mobile application. The design follows a mobile-first approach with a friendly, minimal structure that supports both **Arabic and English localization**, targeting K–12 learners, university students, and parents.

The goal is to ensure:

- Fast and intuitive search and booking flows.
- Reduced cognitive load with clear visual hierarchy.
- High trust and professionalism communicated through clean design.
- Compliance with accessibility guidelines and cultural expectations in Saudi Arabia.

8.1 Home Screen

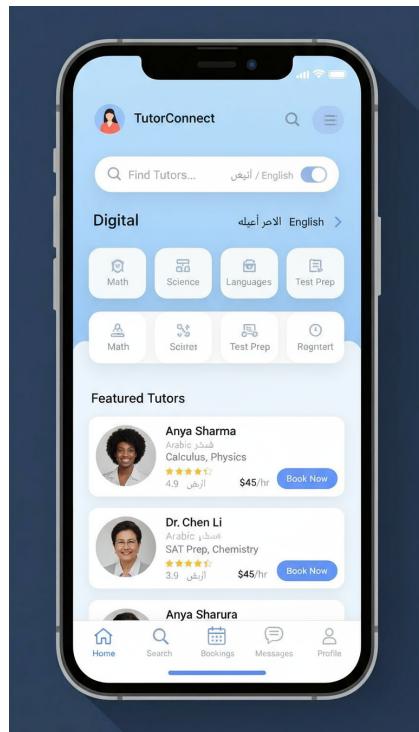


Figure 8.1: Home Screen – Explore subjects and find tutors

The Home screen allows students and parents to:

- Search for tutors using keywords or filters.
- Quickly access subject categories such as Math, Science, Languages, and Test Prep.
- View recommended and trending tutors based on AI ranking.

The clean layout and visual icons reduce friction for young users and parents, improving navigation.

8.2 Tutor Profile Screen



Figure 8.2: Tutor Profile – Ratings, distance, qualifications, and availability

This screen displays detailed tutor information including:

- Name, subject expertise, degrees, and total experience.
- Star ratings, reviews, and distance for face-to-face sessions.
- Clear call-to-action **Book Session** button.

The design supports better decision-making and transparency for parents seeking verified tutors.

8.3 Booking and Payment Screen

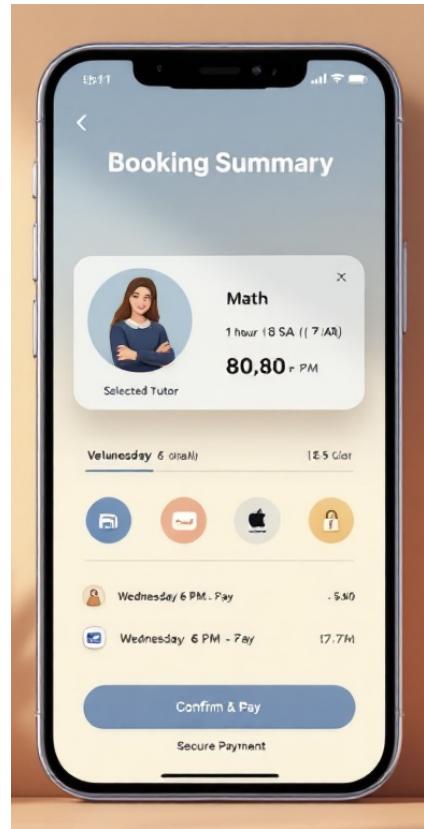


Figure 8.3: Booking and Payment – Confirm session details and secure checkout

The booking summary ensures:

- User confirms selected time slot before paying.
- Payment methods include Mada, Visa, and Apple Pay.
- Secure interface that builds trust in the transaction process.

A simplified design helps students/parents complete the booking successfully without confusion.

8.4 Why This UX Works

- **3-step booking flow** improves usability and reduces drop-off.
- **Localization** ensures accessibility for users in Saudi Arabia.
- **Clear visual hierarchy** lowers cognitive load for young learners.
- **Trust elements** such as secure payment icons reassure parents.
- **Consistency in layouts and icons** enhances learnability.

8.5 Conclusion

Tutorly delivers a smooth and intuitive user experience that supports fast search, informed decision-making, and secure session booking. Future updates will include in-app video sessions, gamified learning progress, and performance tracking dashboards for parents and institutions.

Chapter 9

Validation and Acceptance Criteria

This project uses the functional requirements table (RQ0–RQ13) as the main source of acceptance criteria. For each requirement, the “Acceptance Criteria” column defines measurable pass/fail conditions that will be evaluated through test cases (unit tests, integration tests, UI tests, and end-to-end validations).

Below are explicit validation conditions for key requirements to ensure the system behaves correctly:

- **RQ0 Login and Registration:** Users must be able to create an account, verify credentials, and log in successfully. Invalid credentials must be rejected with proper error messaging.
- **RQ1 Search and Filter Tutors:** When valid filters are applied, the system returns matching verified tutors in ≤ 2 seconds for 95% of requests.
- **RQ2 Booking Sessions:** Conflicting bookings must be blocked and the selected time slot is reserved instantly after confirmation.
- **RQ3 Payment Processing:** If payment succeeds, booking becomes *Paid* and a receipt is stored. If it fails, the booking remains *Pending Payment* with clear feedback.
- **RQ4 Receipt Generation:** A valid digital receipt must be accessible from the user’s dashboard at any time after payment.

- **RQ5 Tutor Availability Management:** Overlapping availability times are not allowed; changes are reflected within 2 seconds in search results.
- **RQ7 Tutor Verification:** Tutors appear with a *Verified* badge only after manager approval is completed and logged.
- **RQ8 Compliance Reporting:** Reports must export correctly to CSV/PDF formats with no missing data.
- **RQ9 Ratings and Reviews:** Submitted reviews are reflected instantly in the tutor's visible average rating. Inappropriate reviews trigger moderation alerts.
- **RQ10 Disputes:** Admin panel must allow issue resolution including refund integration with a maximum turnaround time defined by policy.
- **RQ11 AI Tutor Recommendations:** Recommended tutors must be ranked based on relevance (subject, location, success history) and ordered by descending match score.
- **RQ12 Notifications and Reminders:** Schedule-based reminders must be logged and delivered successfully via SMS/email with retry logic up to 3 times.
- **RQ13 KPI Dashboards:** Metrics and analytics must update dynamically based on date range filters and remain available to authorized stakeholders only.

All requirements listed in the RQ0–RQ13 table have traceable testing criteria to ensure full acceptance of the Tutor Uberization Platform MVP.

Chapter 10

Appendices

10.1 Business Model Canvas

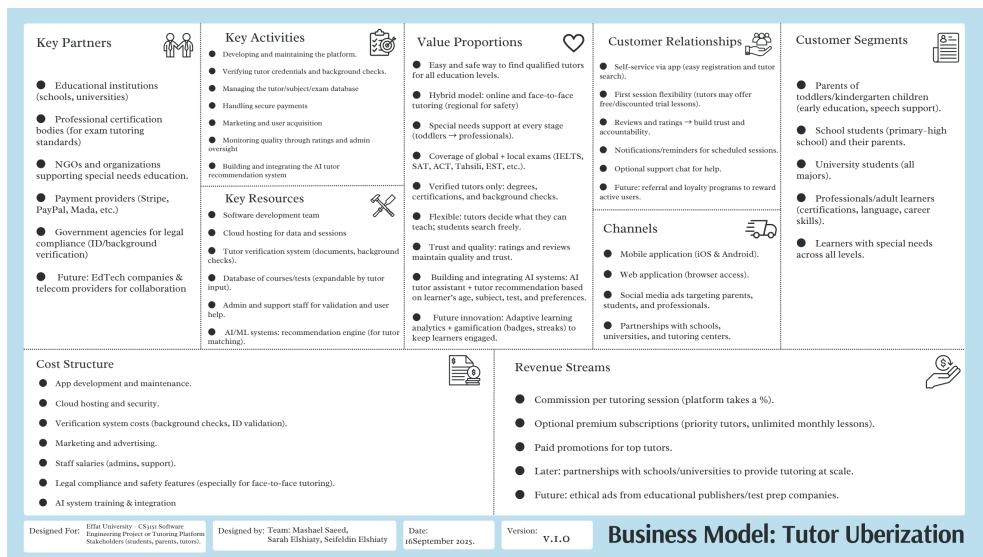


Figure 10.1: Business Model Canvas – Tutor Uberization Platform

10.2 Gantt Chart

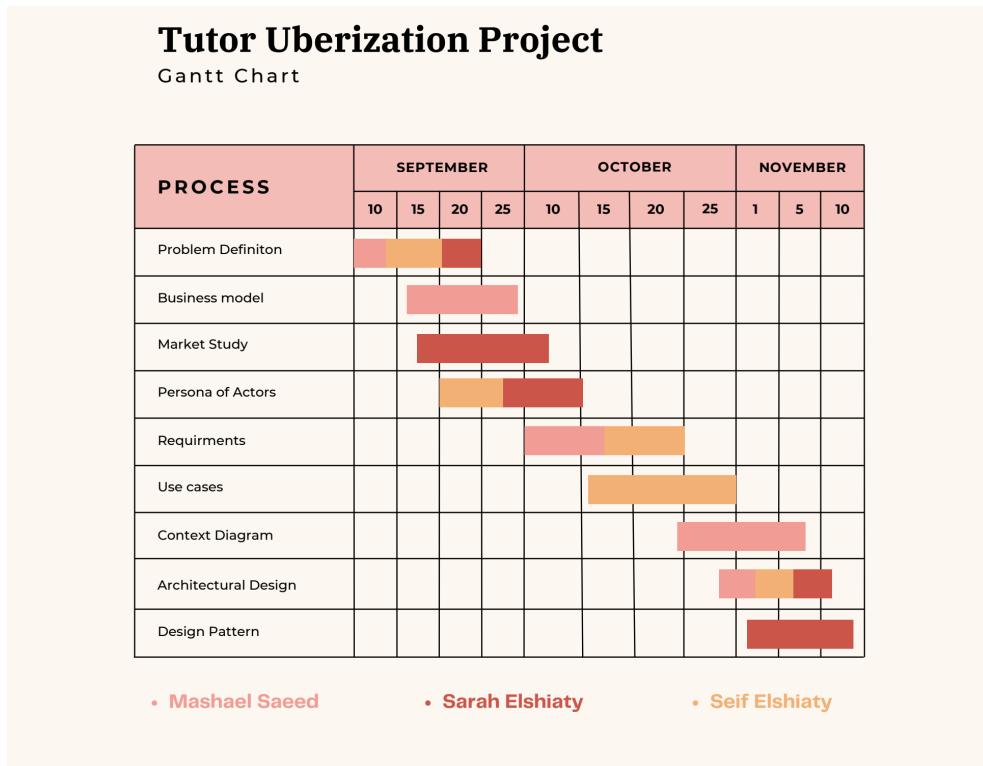


Figure 10.2: Project Gantt Chart

10.3 Project Workload Distribution

The following table summarizes the contributions of each team member across the major project tasks for the Tutor Uberization Platform. Percentages represent estimated division of effort and collaboration throughout the semester.

Task	Duration	Mashael Saeed	Sarah Elshiaty	Seifeldin Elshiaty
Problem Definition	Sep. 1 – Sep. 10	30%	30%	40%
Business Canvas	Sep. 15 – Sep. 25	40%	30%	40%
Market Study	Sep. 25 – Sep. 30	30%	50%	30%
Persona of Actors	Sep. 25 – Oct. 15	30%	35%	20%
Requirements	Oct. 15 – Oct. 25	30%	40%	30%
Use Cases	Oct. 20 – Oct. 31	25%	25%	50%
Context Diagram	Oct. 31 – Nov. 5	50%	30%	20%
Architectural Design	Nov. 1 – Nov. 10	30%	30%	40%

Table 10.1: Team Workload Allocation for Tutor Uberization Project

The workload shows balanced collaboration and clear task ownership among all three project members.

10.4 Glossary

- PDPL – Personal Data Protection Law (KSA).
- KPI – Key Performance Indicator.
- AI – Artificial Intelligence.
- Hybrid Tutoring – Combination of online and in-person learning.