

Tutor Uberization

Final Software Engineering
Presentation
Connecting Students with Verified Tutors

Team: Mashael Saeed · Sarah Elshiaty · Seif Elshiaty

Course: CS3151 - Software Engineering

Instructor: Dr. Passent Elkafrawy – Spring 2025

Why Tutor Uberization?

Addressing critical challenges in the modern tutoring landscape to ensure quality and accessibility for all learners.

1

Lack of Verified Tutors

Presents a significant quality and safety risk for students and parents.

2

No Hybrid Tutoring Model

Absence of integrated online and safe in-person learning options.

3

Weak Saudi Exam Support

Limited resources for specialized Saudi exams like Qiyas and Tahsili.

4

Limited Access for Special Needs

Inadequate support and accessibility for learners with special educational needs.

5

Parental Oversight

Parents lack crucial visibility over their child's progress and spending on tutoring services.



Existing Solutions vs. Our Vision

Analyzing current market offerings to highlight the unique strengths of our platform.

Feature	Now	Preply	Traditional	Wyzant	Superprof	Our
	Academy		Centers			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	✓	✓	✗	✓	✓	✓

None combine: Verification + AI + Hybrid + Local Exams + Safety

What Makes Us Special?

Our platform stands apart by integrating cutting-edge technology with user-centric design.



AI Tutor Recommendation



Verified Safety System
(Parental Approval)



Hybrid Smart Availability



Support for Special Needs



Gamified Retention



Data Reports for
Partners/Regulators

Impact: SDG 4 & SDG 10 + Saudi Vision 2030

Our platform directly contributes to national development goals and global sustainability targets.



Quality Education

Increased access to diverse, high-quality tutoring services for all learners.



Trusted & Safe

A secure environment for both students and tutors through robust verification processes.



Empowering Tutors

Providing independent income opportunities and flexible work for educators.



Data-Driven Learning

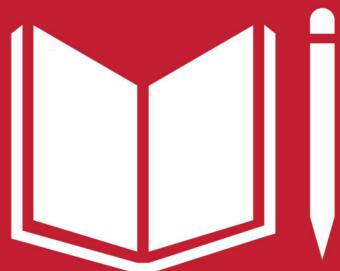
Utilizing analytics to continually improve learning outcomes and personalize experiences.



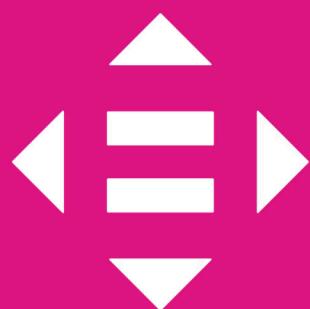
Local Economy Support

Fostering local economic growth aligned with Saudi Vision 2030 objectives.

4 QUALITY EDUCATION



10 REDUCED INEQUALITIES



رؤية
2030
المملكة العربية السعودية
KINGDOM OF SAUDI ARABIA

Functional Requirements

Defining the essential features that form the backbone of our tutoring platform.

Core Functional Features

- Search and filter verified tutors
- Book tutoring sessions (online or in-person)
- Manage bookings (view, cancel, reschedule)
- Submit reviews and view tutor ratings
- Generate basic reports for users and admins

Smart Functional Features

- AI-based tutor recommendations based on subject, level, and learner profile
- Hybrid scheduling (choose online or face-to-face)
- Personalized learning paths based on user history

Trust & Safety Functional Features

- Tutor verification workflow (documents + approval)
- Parental oversight for minors (approve bookings & monitor sessions)
- In-platform messaging for safe communication

Software Requirements

The diagram illustrates the software requirements structure. At the top, a large white circle labeled "Core" is positioned above a dark blue rectangular area labeled "Core Esmart & Smart". Below this is a large orange area. To the right, there is a dark blue circle labeled "Smart" and a light blue circle labeled "Trust & Safety Vintage blue". The background features abstract shapes in various colors (blue, orange, yellow) and a faint grid pattern.

Core

Core
Esmart & Smart

Smart

Trust &
Vintage blue

Non-Functional Requirements

Ensuring our platform operates seamlessly, securely, and efficiently to provide the best user experience.



Performance

Fast search and booking with sub-2-second response times for a smooth user journey.

Security

PDPL compliant and encrypted payments (TLS 1.3) using secure APIs to protect user data.

Reliability

Available 24/7 with minimal downtime, guaranteeing continuous access to tutoring services.



Usability

Accessible UI in both Arabic and English, mobile-first design, and simple navigation.



Scalability

Designed to support a growing number of tutors and students without performance degradation.



Maintainability

Modular architecture with clean design patterns for easy updates and future enhancements.

Traceability Matrix: Use Case → Requirements

Ensuring every requirement is linked to a specific use case, guaranteeing comprehensive coverage.

This matrix visually demonstrates the robust connection between our system's requirements and its intended functionalities, ensuring full accountability and development integrity.

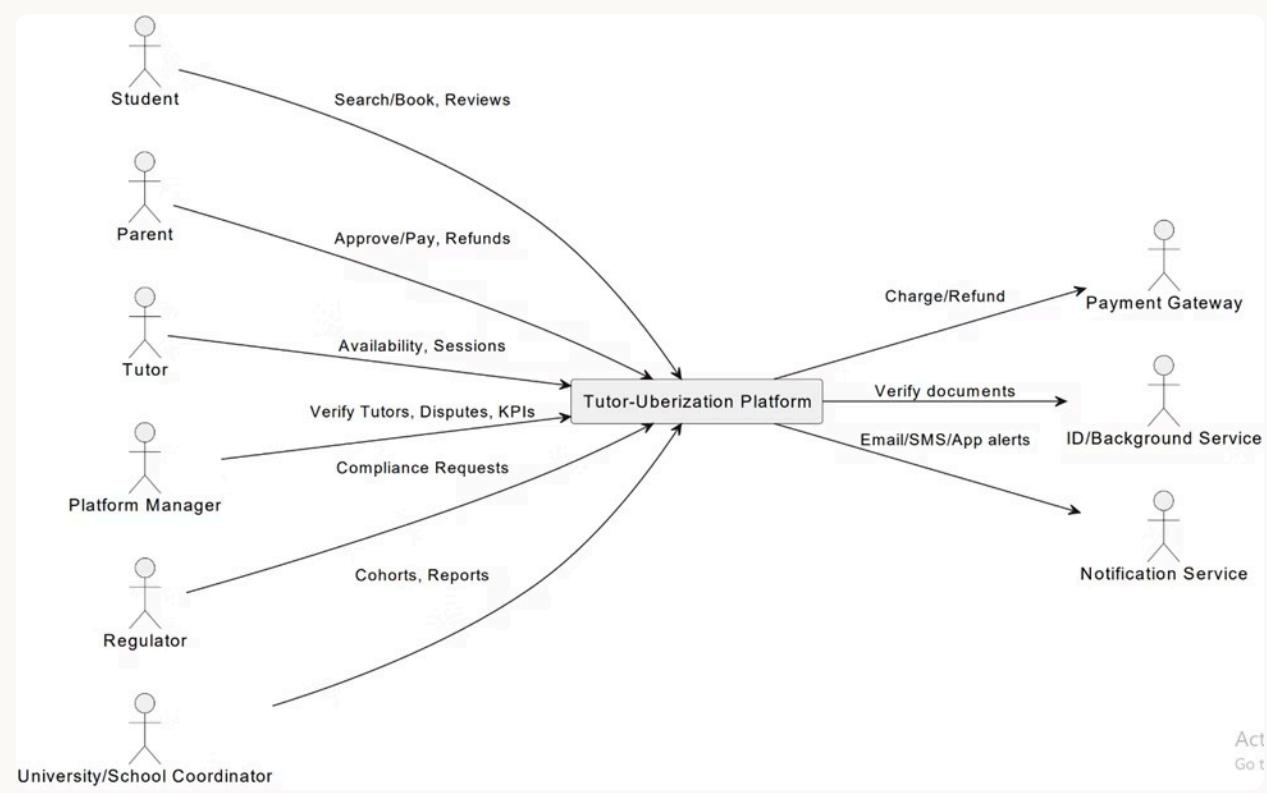
The main goal-level use cases are:

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

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

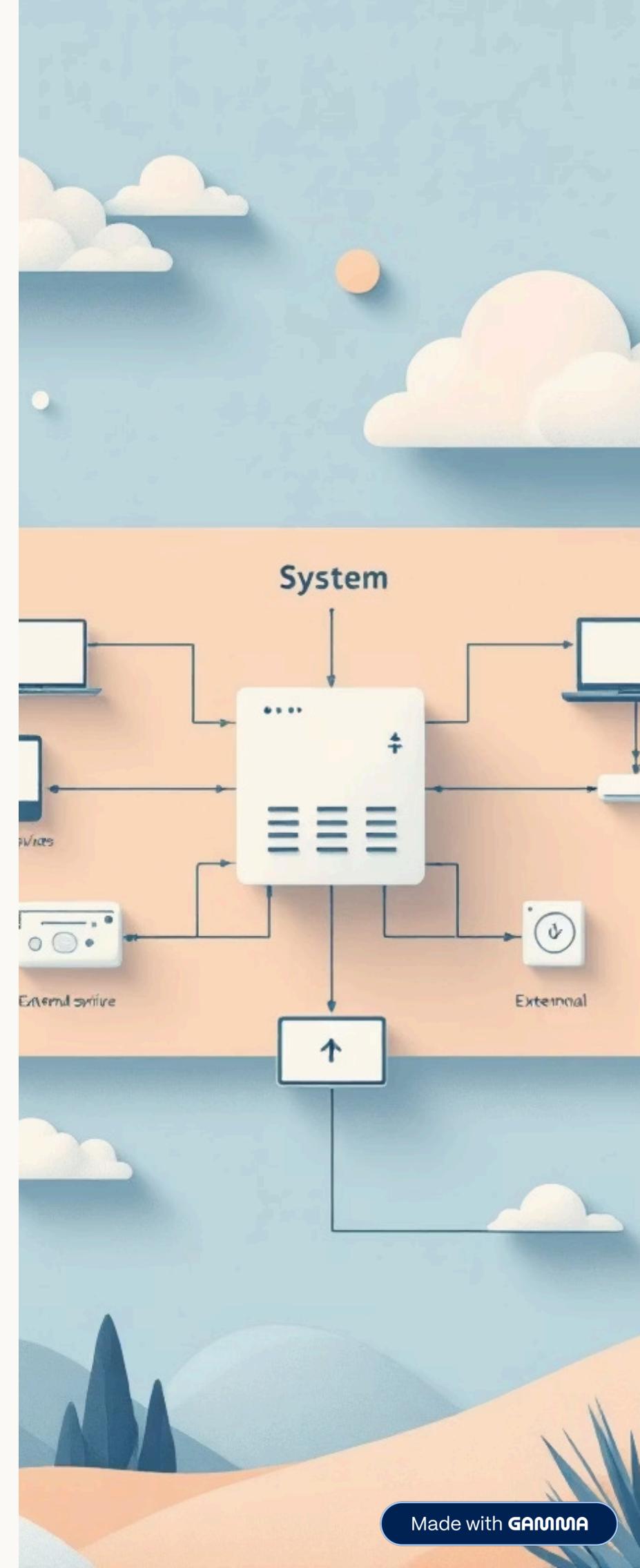
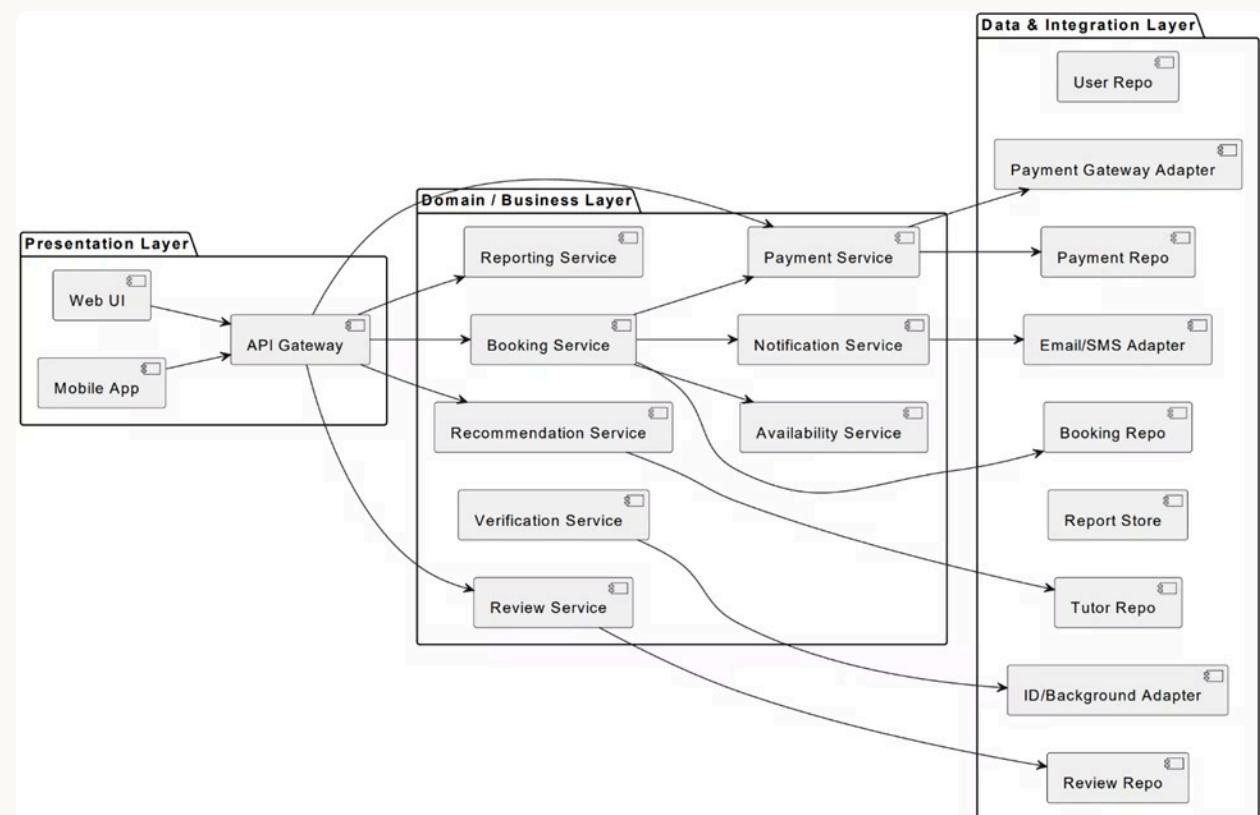
Requirements	Priority	UC1	UC2	UC3	UC4	UC5	UC6
RQ1 : Search for Tutors	M	X					
RQ2 : Book a Tutor Session	M	X					
RQ3 : Make a Secure Payment	C	X	X				
RQ4 : Provide Digital Receipt	S	X		X			
RQ5 : Management of Tutor Availability	M	X	X				
RQ6 : Conduct Tutoring Session	S		X				
RQ7 : Verify Tutor Credentials	M		X		X		
RQ8 : Generate Compliance Logs	M			X			
RQ9 : Submit Review	M			X			
RQ10 : Handle Refund or Dispute	M				X		
RQ11: AI Tutor Recommendation	S				X		
RQ12 : Send Reminders	S	X				X	
RQ13 : Generate KPI and Partner Report	S				X	X	X

Context



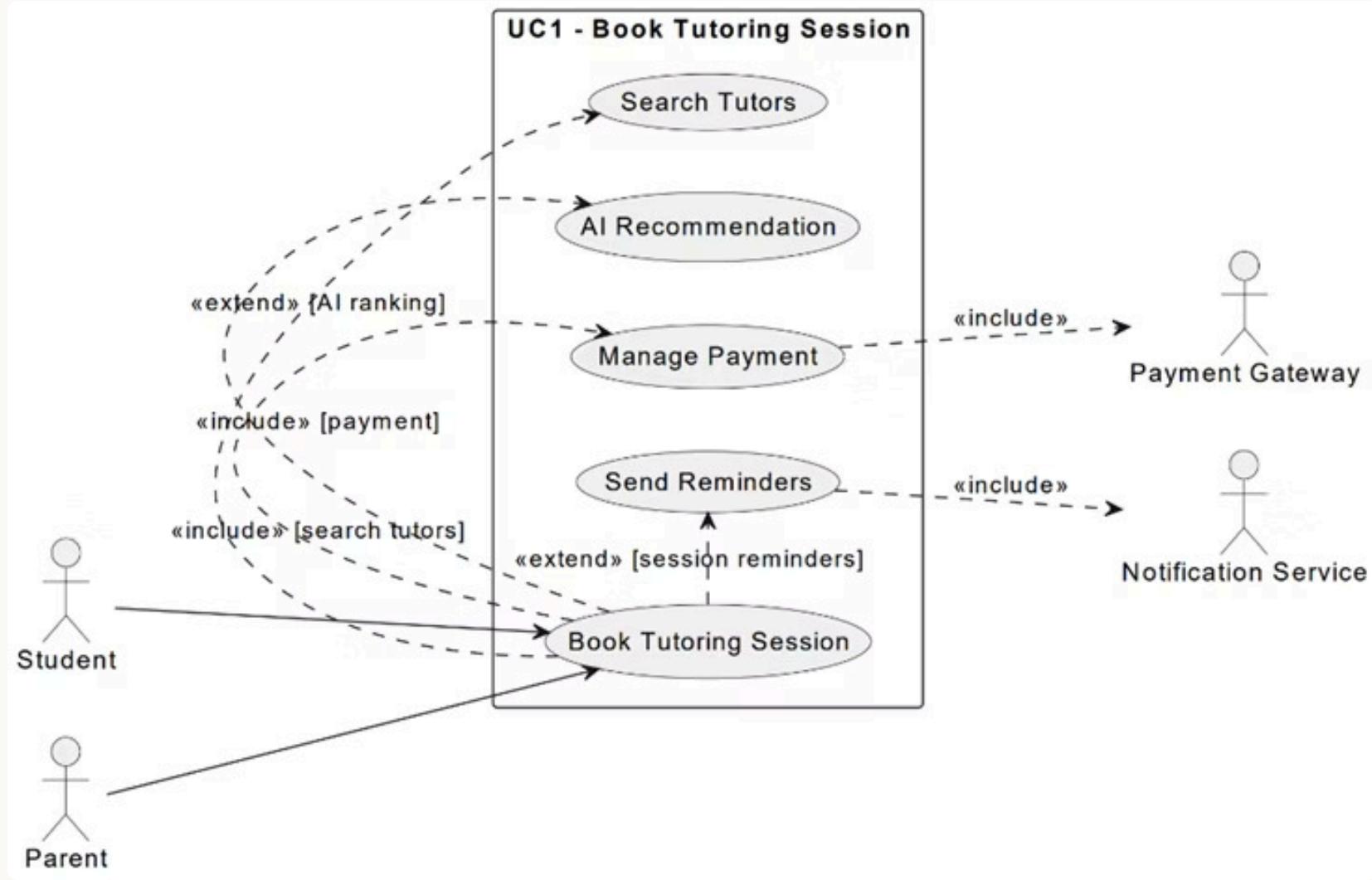
Architecture

Secure | Scalable | Cloud-first

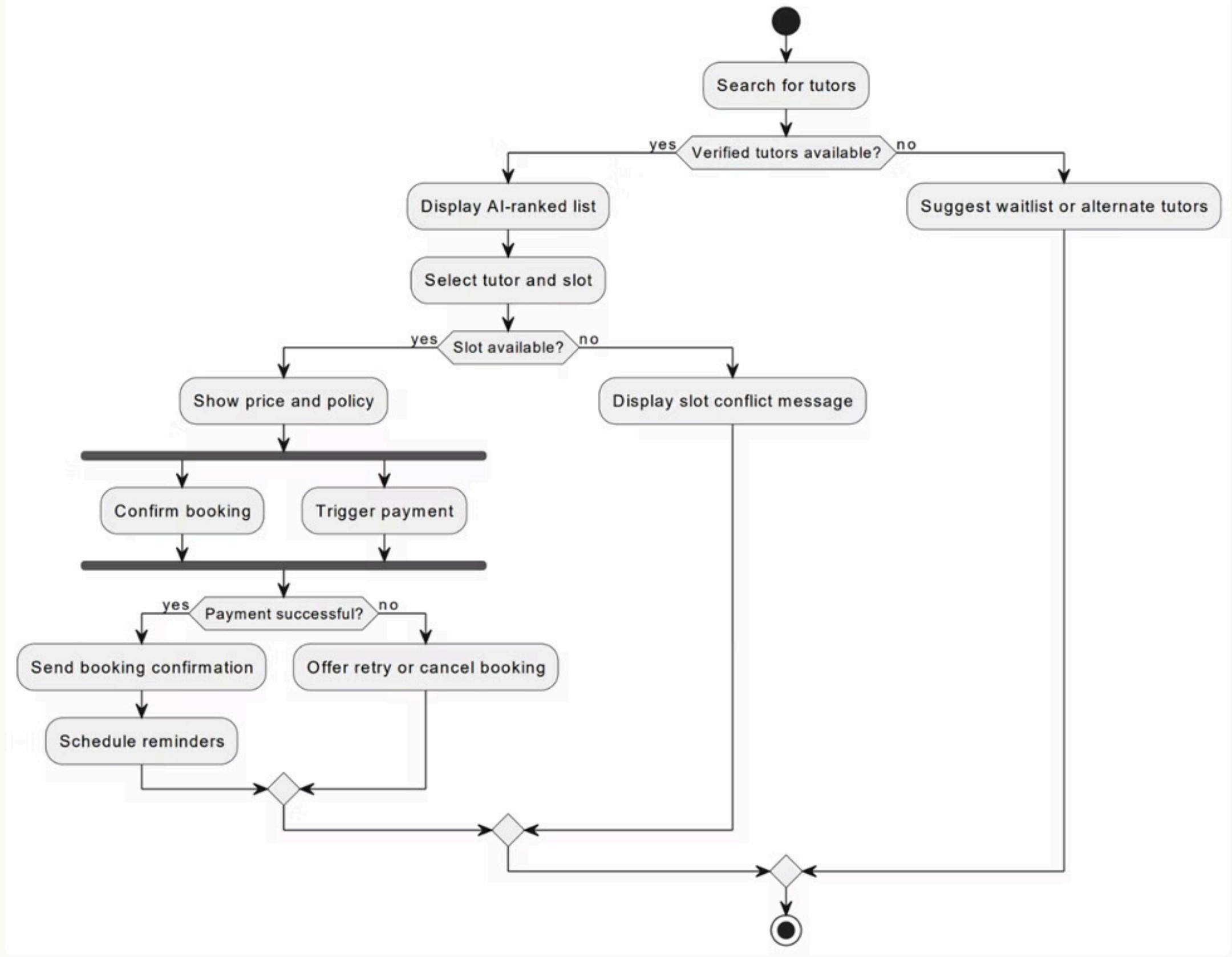


Main Use Cases: Book

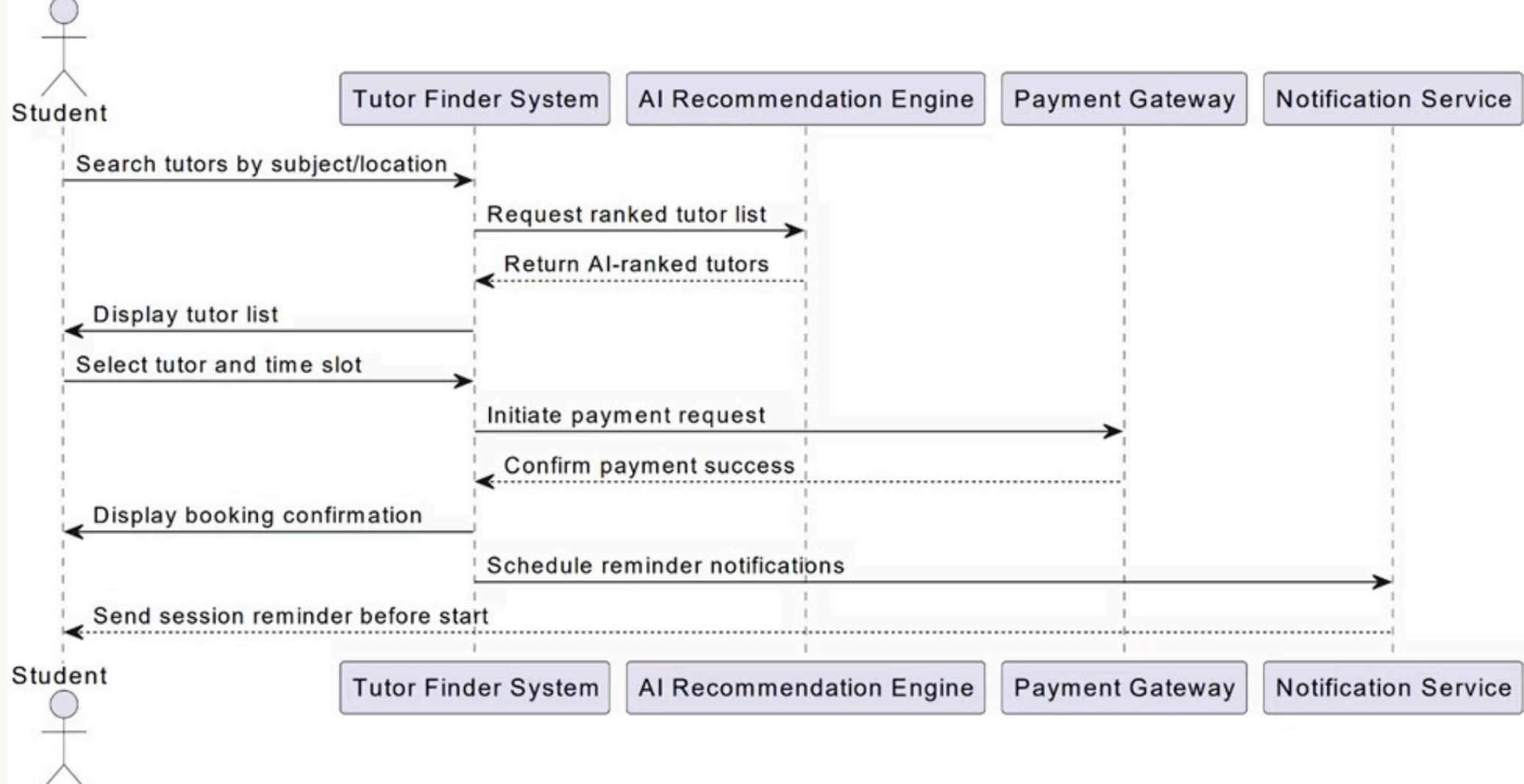
UC1 – Book Session



Activity Diagram

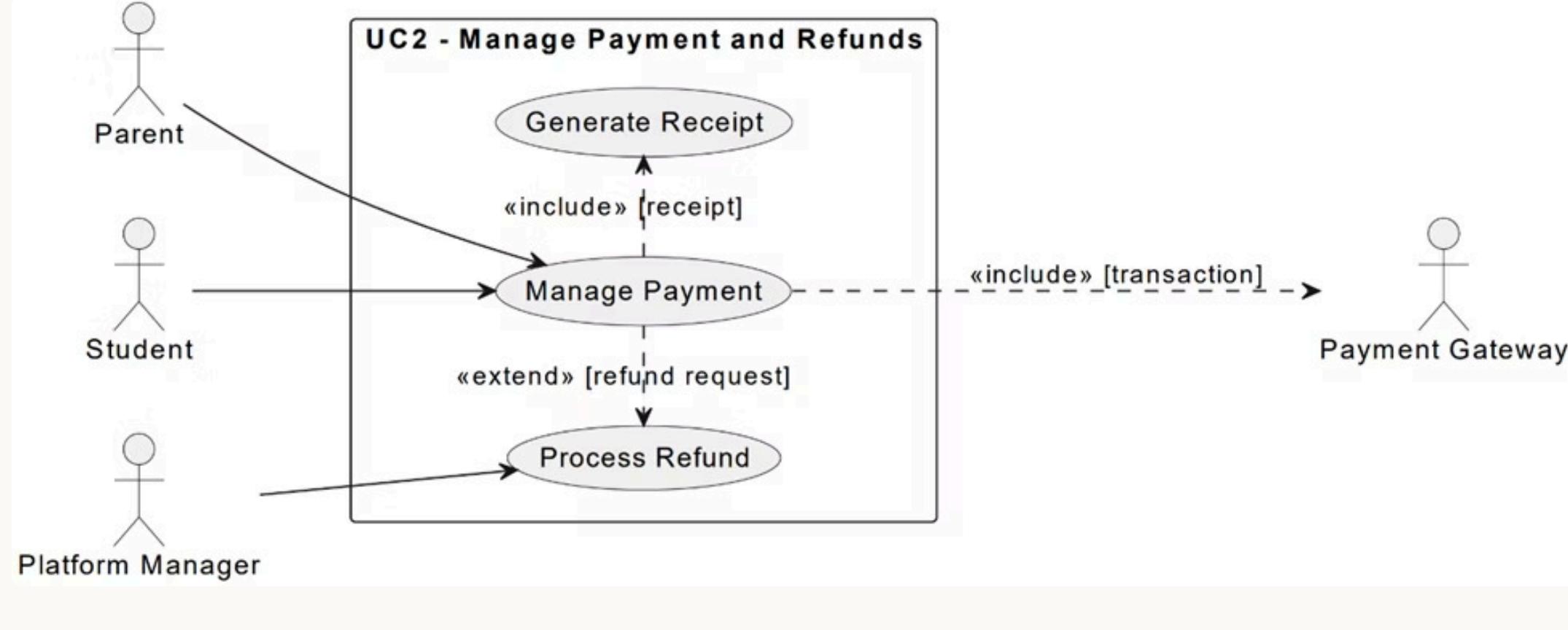


Sequence Diagram

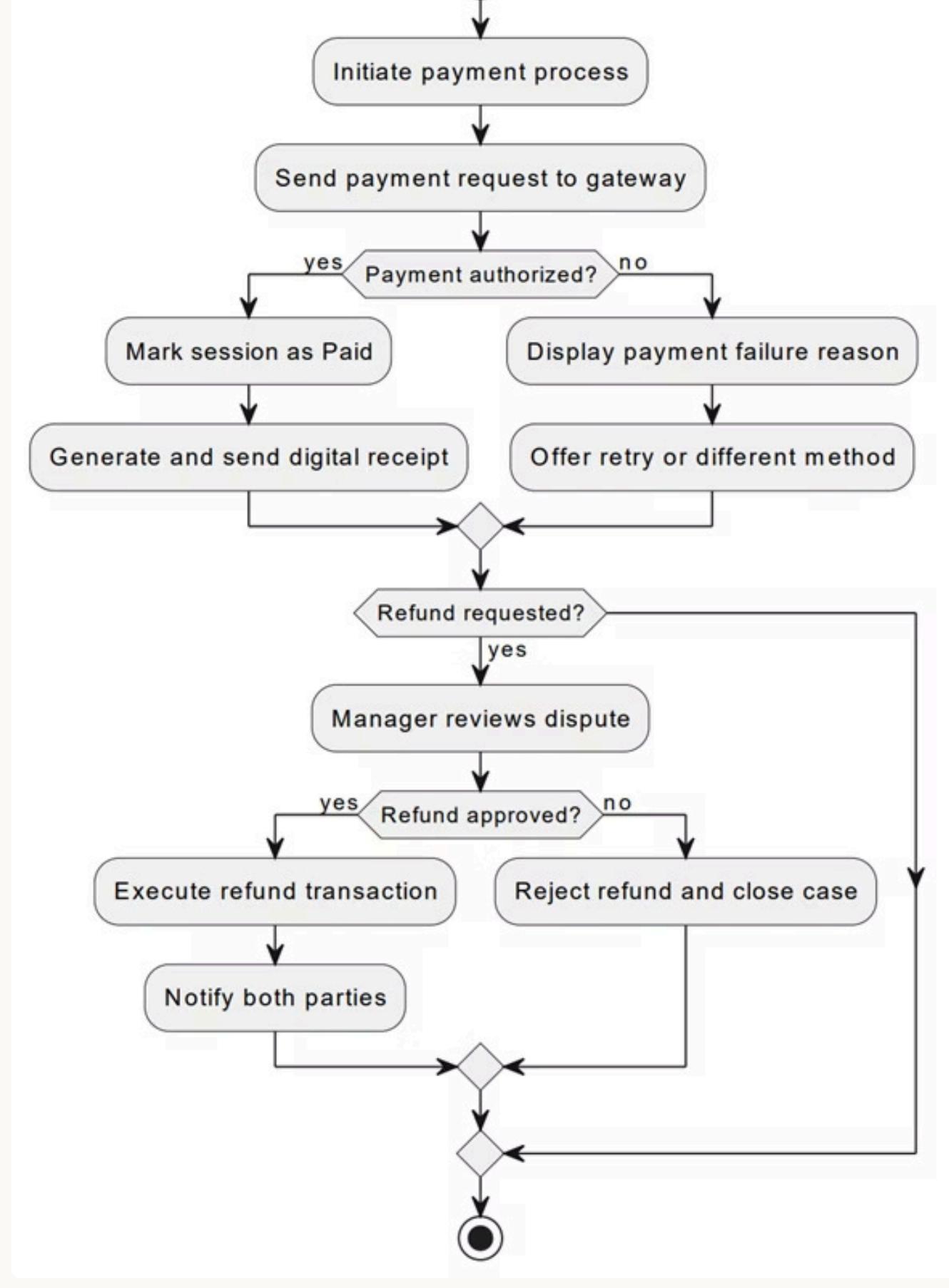


Main Use Cases: Pay

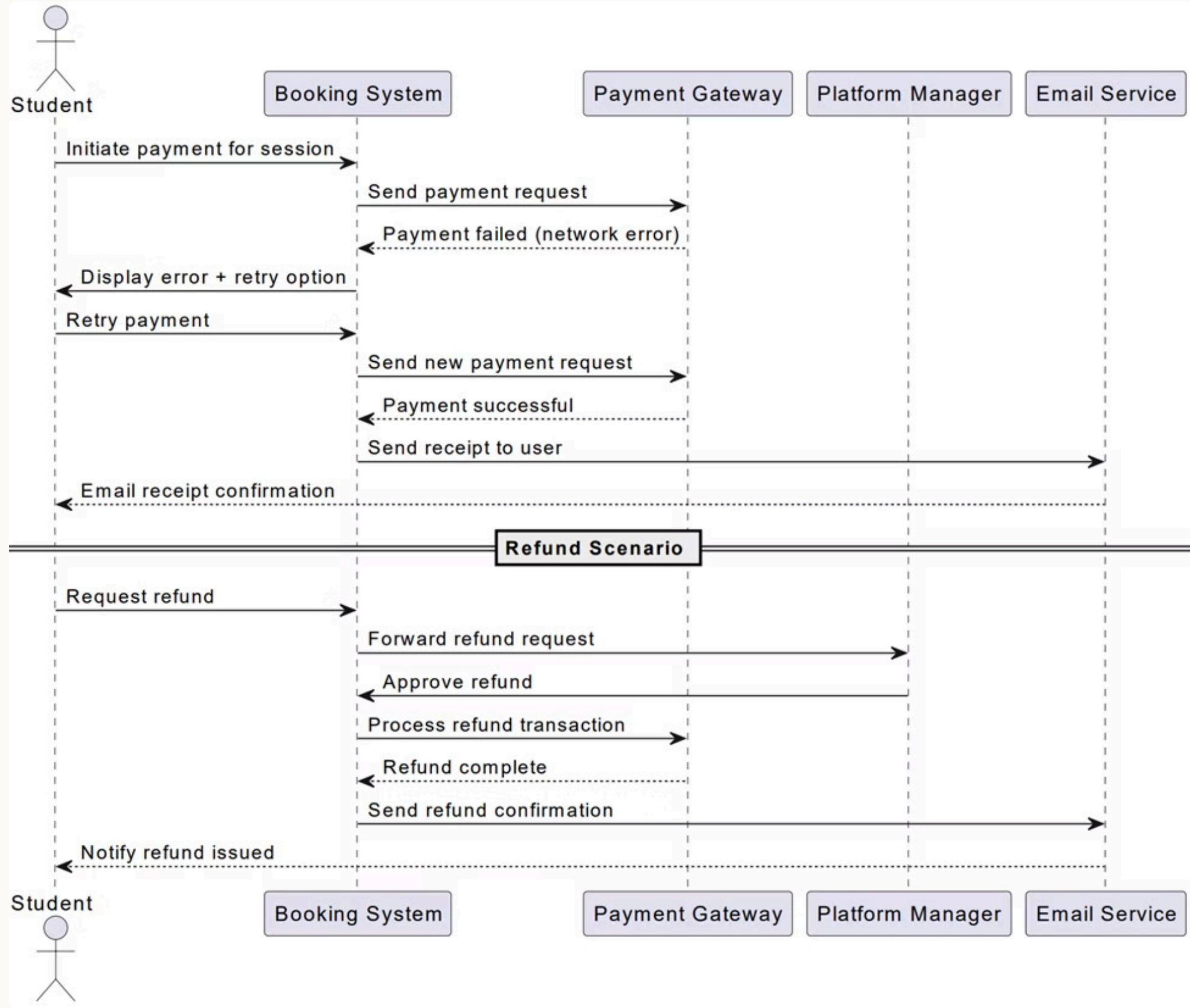
UC2 – Payment



Activity Diagram

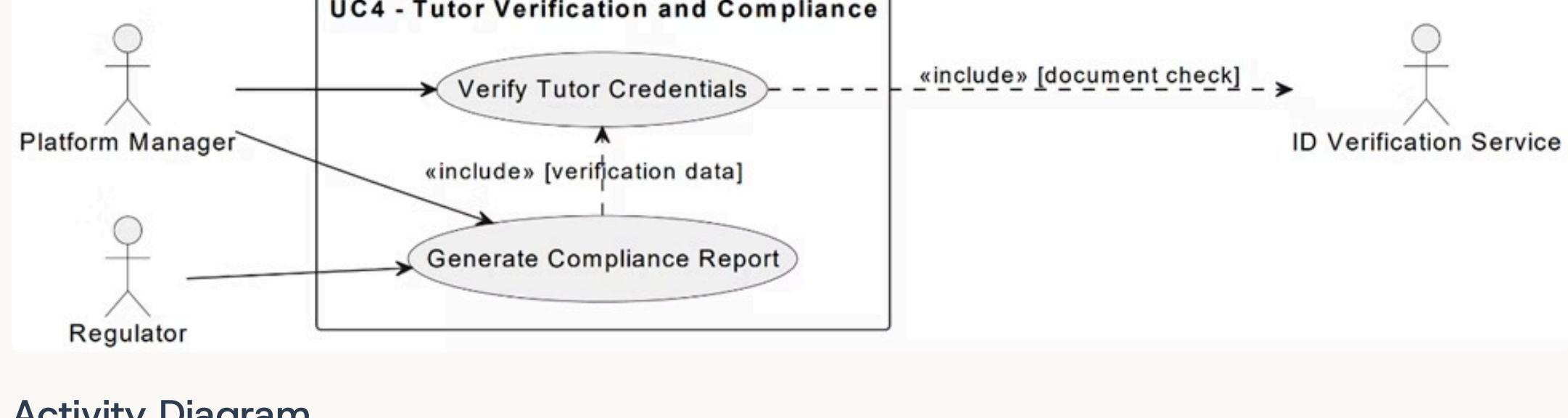


Sequence Diagram

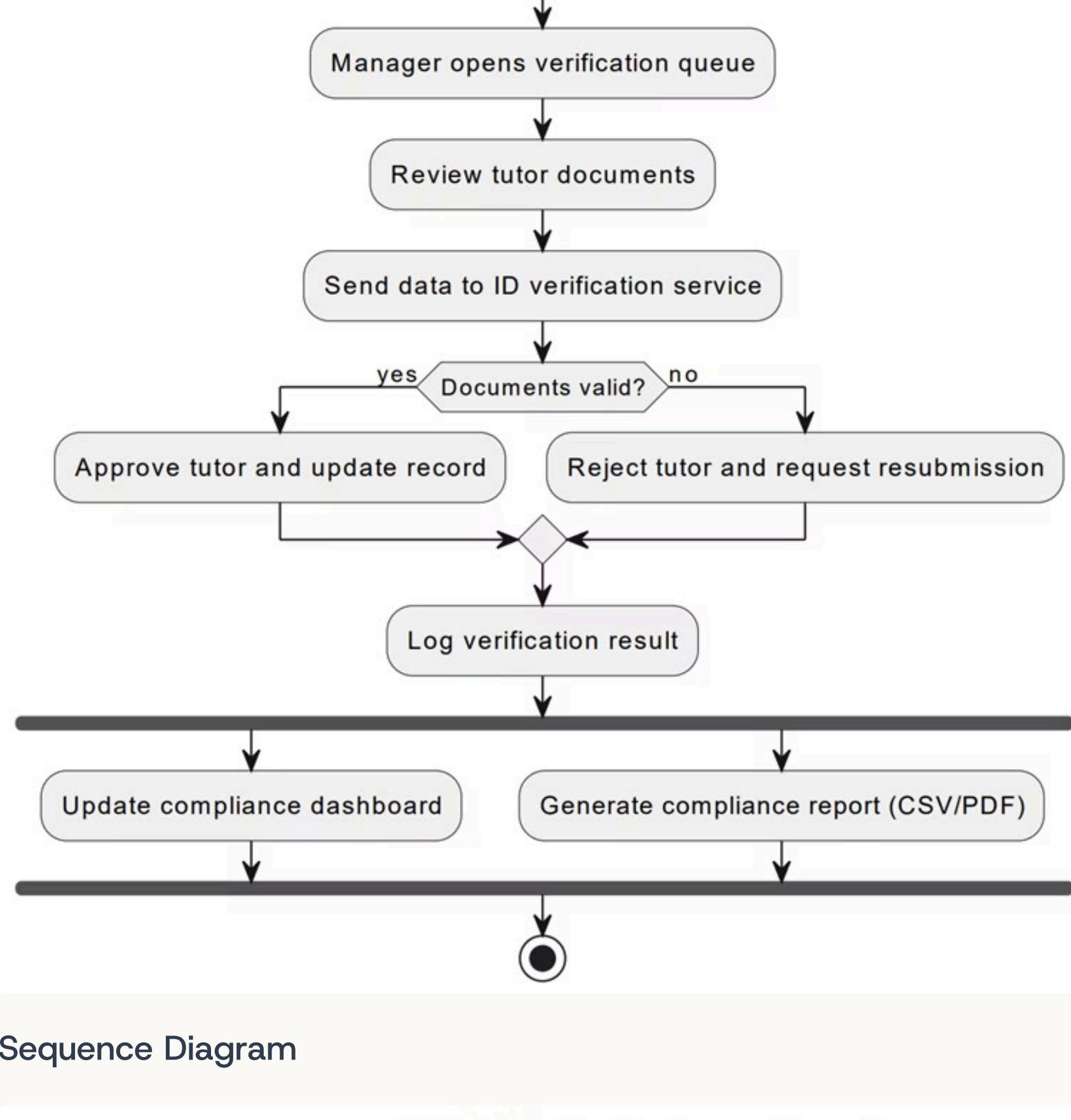


Main Use Cases: Verify

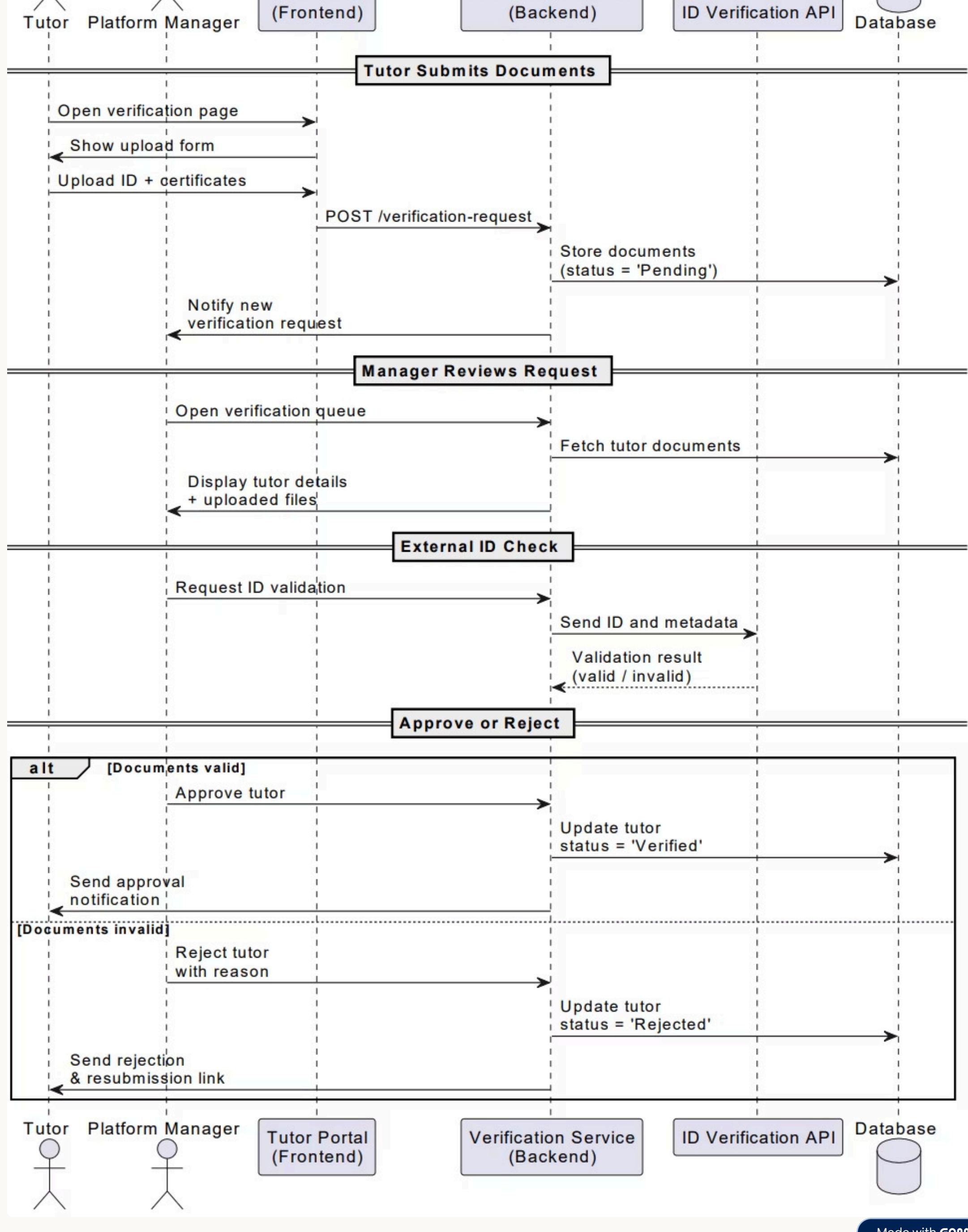
UC4 — Tutor Verification



Activity Diagram



Sequence Diagram



Design Pattern & Class Diagram

Patterns Used

Observer

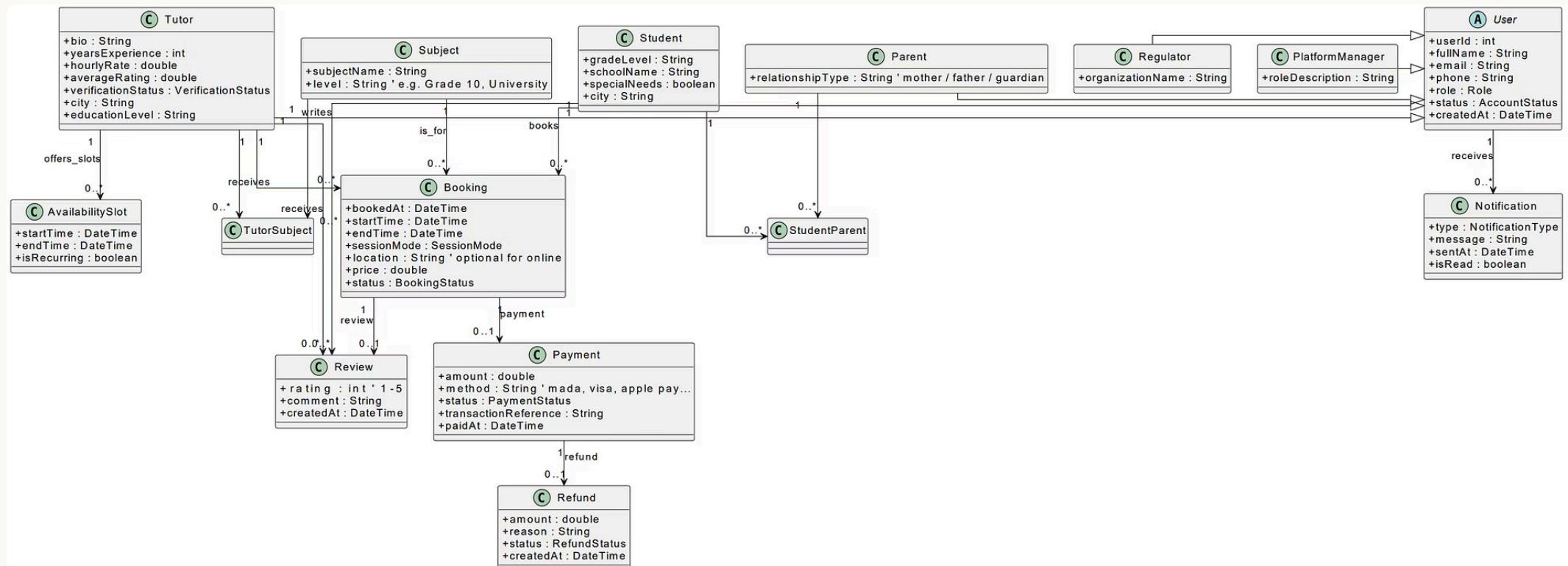
For seamless notifications across the platform.

Strategy

To handle diverse payment methods flexibly.

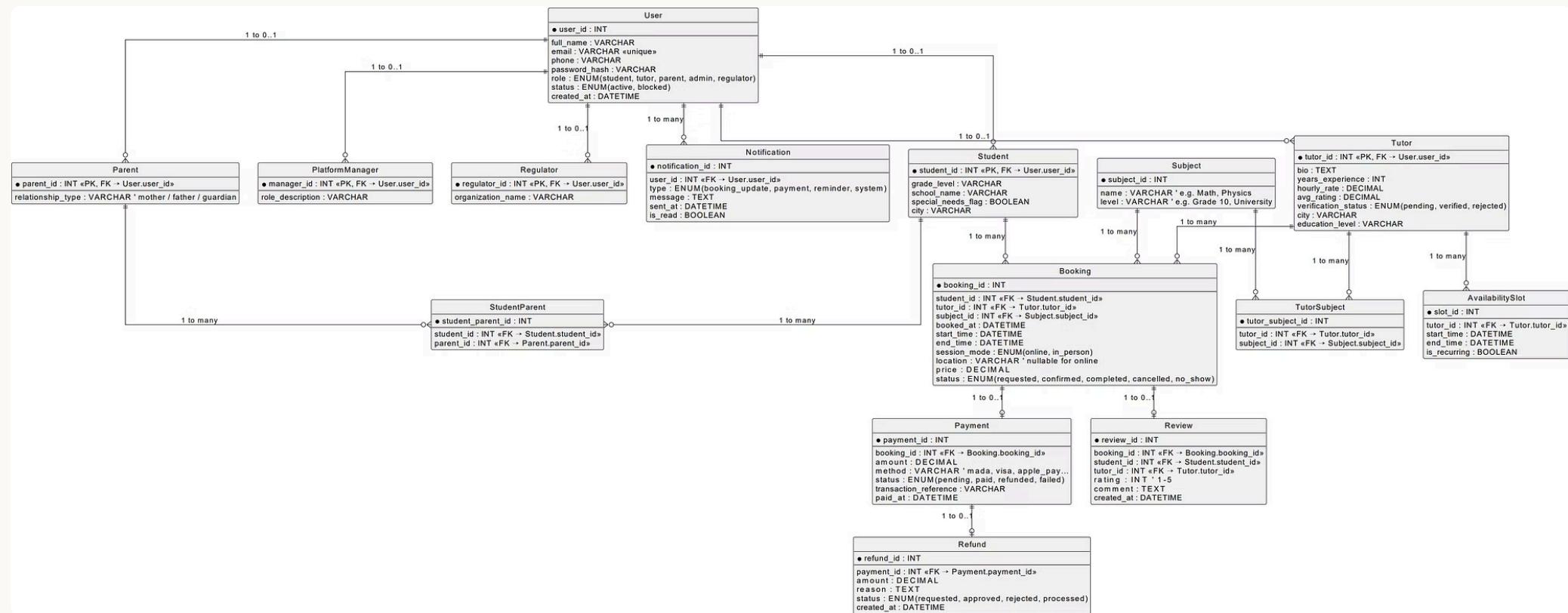
State

Managing the dynamic lifecycle of booking processes.



Reusable · Maintainable

Designed Database



Supports scalability + compliance

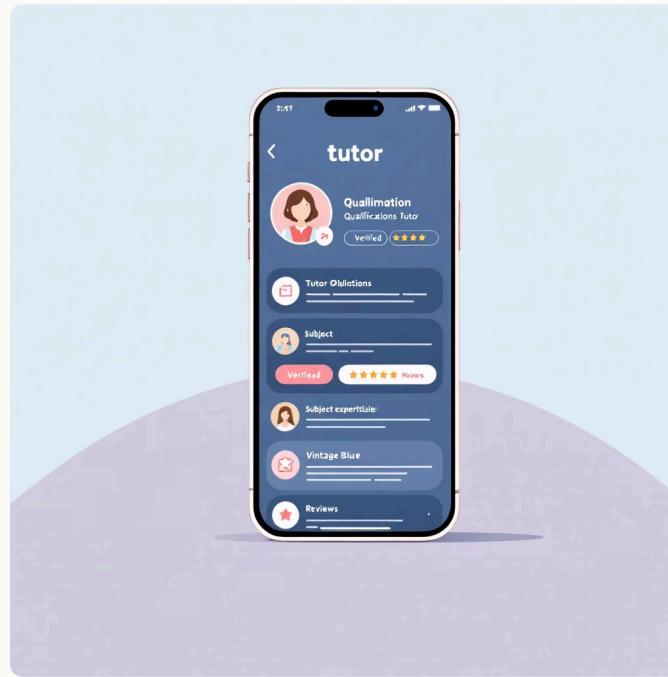
UI Mockups

Visualizing the user experience with intuitive designs for key interactions.

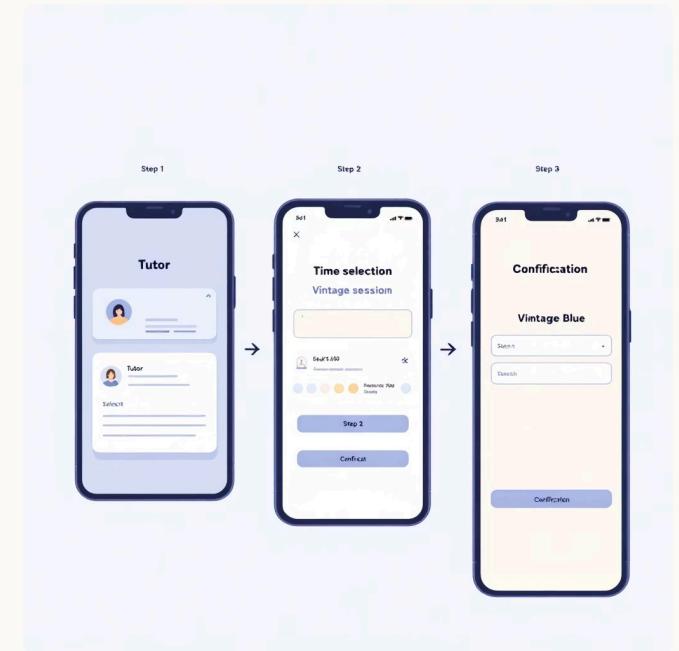
Simple for kids and parents



Verified trust badges

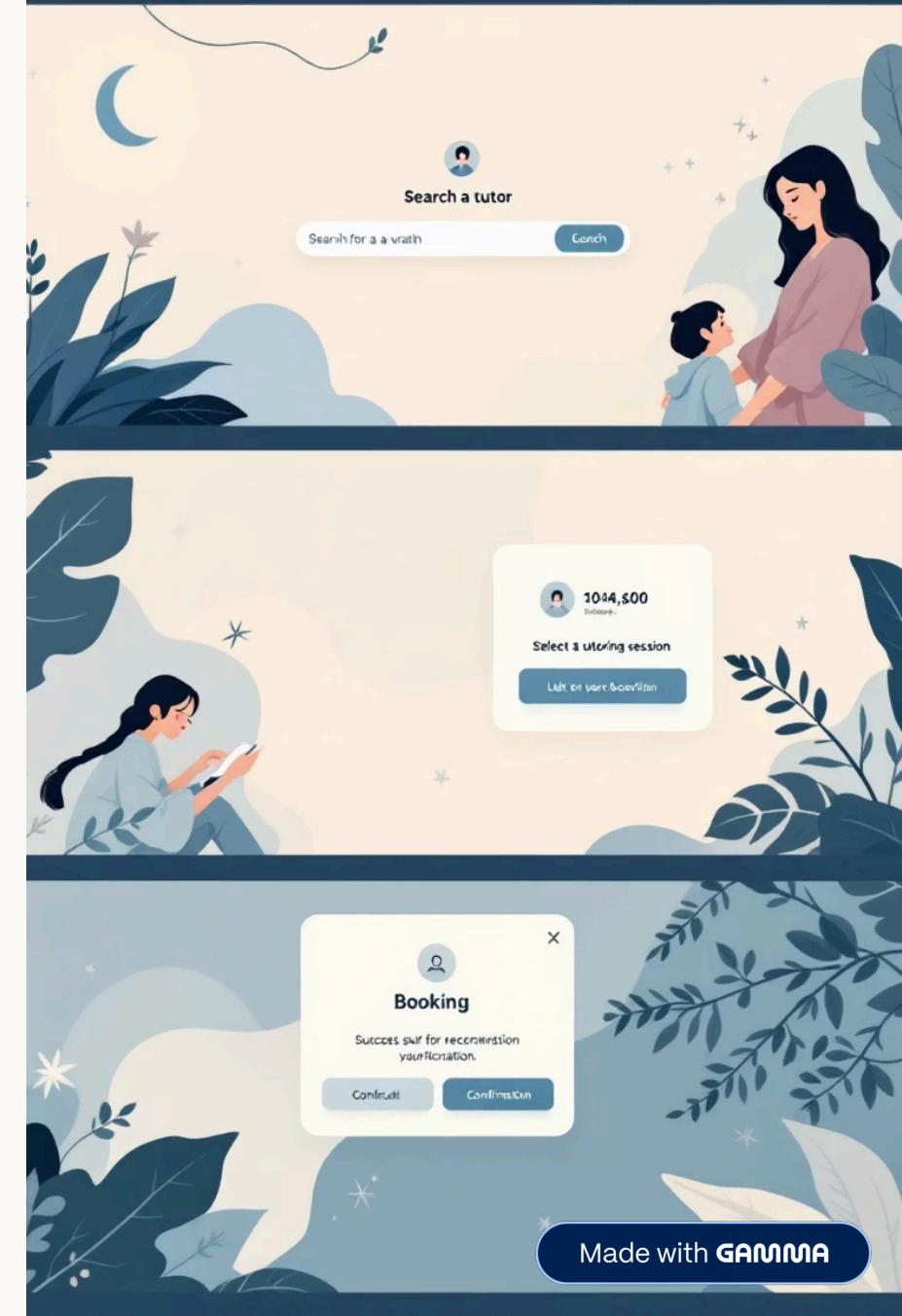
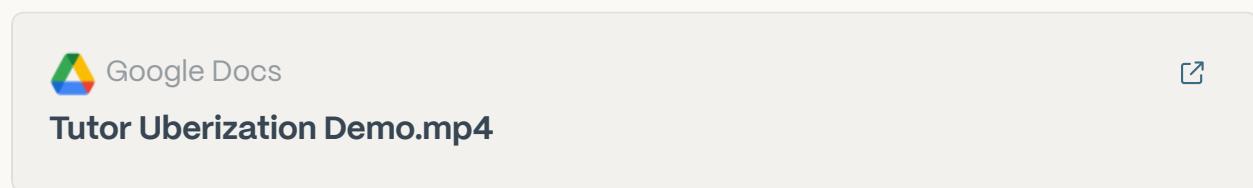


Faster 3-step booking



Live Prototype Demo – Search → Book → Confirm

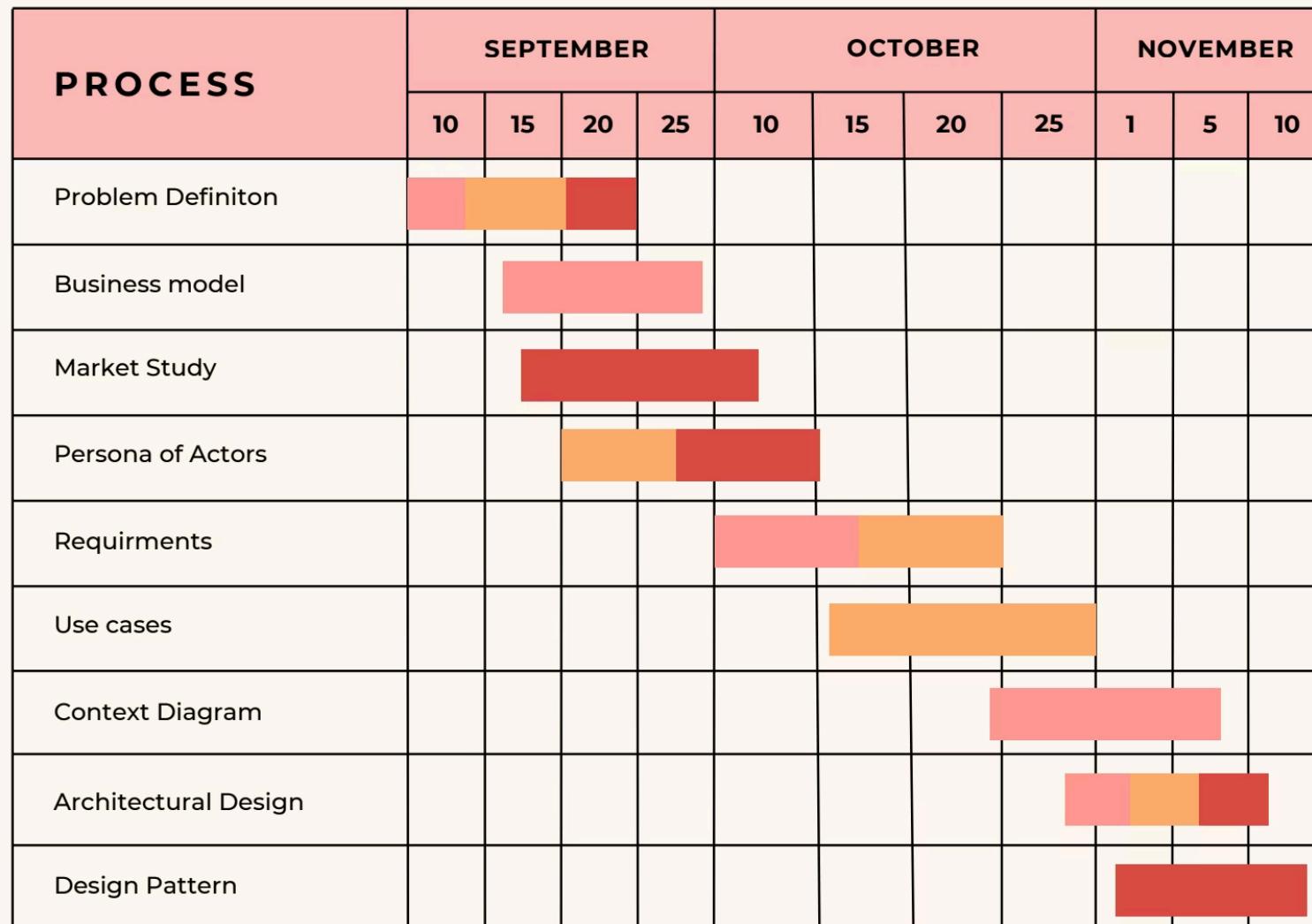
Our interactive prototype demonstrates the core user journey, showcasing a seamless search and booking experience with verified tutors. This 30–45 second demo highlights the intuitive interface and key functionalities that make Tutor Uberization stand out as a safe, hybrid, and AI-powered educational solution aligned with SDG 4: Quality Education.



Team Division Work

Tutor Uberization Project

Gantt Chart



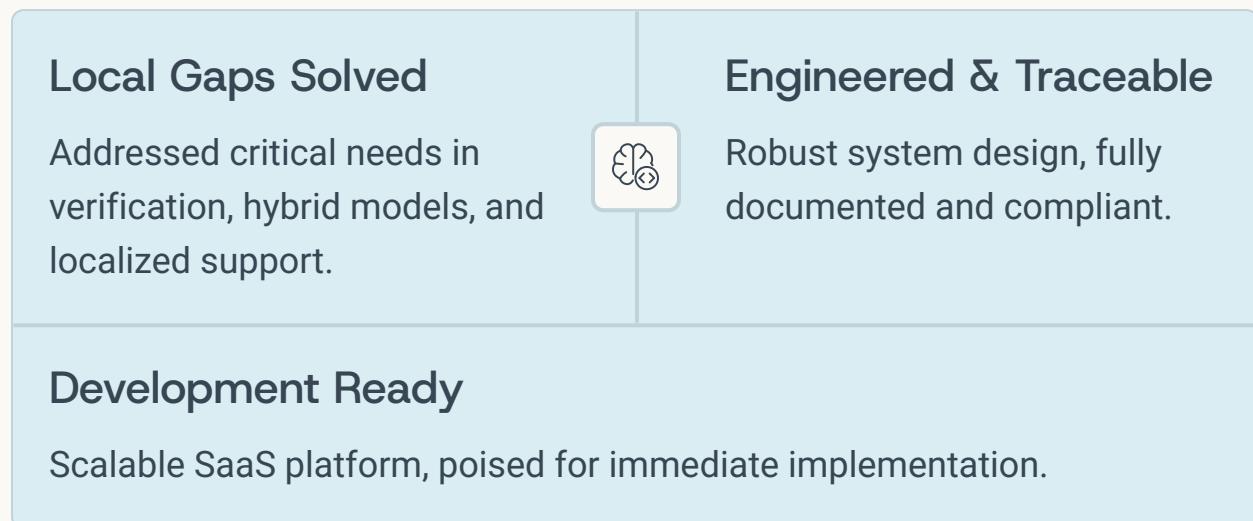
• Mashael Saeed

• Sarah Elshiaty

• Seif Elshiaty

Conclusion & Implementation

Next



Roadmap: Next Steps



MVP

Launch core features to early adopters.

Institutional Partnerships

Expand reach through collaborations with schools and universities.

Gamification

Enhance user engagement and retention.

AI Analytics

Personalize learning and optimize platform performance.



Research Paper Summary

Research Contribution — Secure Distributed Software Agents

Topic

Security vulnerabilities in distributed software agents and how modern SE can protect them.

Research Methods

Structured literature review + analytical synthesis
Evaluation of SE methodologies: Agile, DevOps, DevSecOps, MDE
Study of security risks: impersonation, message tampering, DoS attacks

Key Findings

- Security is often missing at runtime in agent systems
- DevSecOps + MDE + cryptographic trust models strengthen resilience
- Need for **security-by-design** in decentralized environments

Academic Value

Contributes to secure engineering practices in IoT, AI, and cloud-based agents

Supports building **trustworthy digital ecosystems**

Security of Distributed Software Agents in Modern Software Engineering Systems

Sarah Khaled Elshiyaty (ID: S23108195), Mashael Saeed (ID: S23108689), Seif Khaled Elshiyaty (ID: S23108194)

Department of Computer Science, Effat University, Jeddah, Saudi Arabia.

Contributing authors: skelshiyaty@effatuniversity.edu.sa;
mmsaeed@effatuniversity.edu.sa; skhelshiyaty@effatuniversity.edu.sa;

Abstract

Distributed software agents are autonomous programs that operate across interconnected environments such as cloud computing, the Internet of Things (IoT), and artificial intelligence (AI) ecosystems. Their capacity for independent decision-making and collaboration has transformed modern computing but also introduced serious security vulnerabilities including impersonation, message tampering, and coordination failure. This study examines how contemporary software engineering (SE) methodologies can be extended to protect these agents. Using a structured literature review and analytical synthesis, it explores agent-specific security risks, engineering frameworks such as DevSecOps (Development-Security-Operations), and emerging tools like blockchain-enabled trust models. Findings reveal that although cryptographic and decentralized approaches are advancing, agent-level security remains insufficiently embedded in current SE practices. The paper concludes by advocating security-by-design principles and adaptive frameworks that combine formal verification, automation, and trust modeling for secure, scalable multi-agent systems.

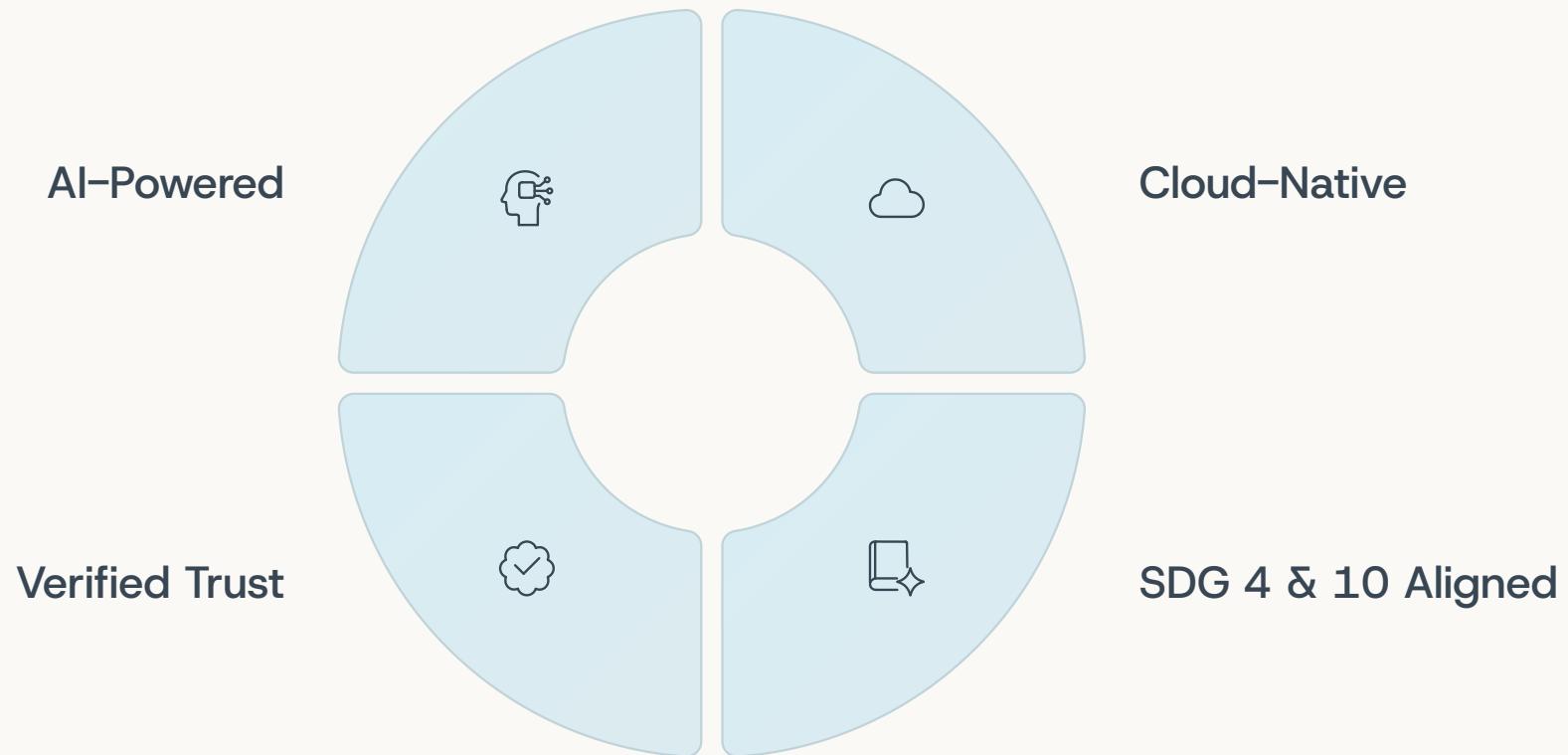
Keywords: Distributed Agents, Software Security, Software Engineering Methodologies, Multi-Agent Systems, Trust Models

1 Introduction

Distributed software agents are self-governing computational entities that perform tasks across multiple systems or networks to achieve common objectives. Unlike static

Thank You & Q&A

Your questions and feedback are invaluable as we move towards our MVP launch.



We look forward to collaborating with you to empower learners and educators.

References

1. United Nations. *Sustainable Development Goals (SDG 4 & SDG 10)*.
<https://sdgs.un.org/goals>
2. Saudi Vision 2030 Official Portal.
<https://www.vision2030.gov.sa>
3. Ministry of Education, Saudi Arabia – Programs and Initiatives.
<https://www.moe.gov.sa/en/Pages/default.aspx>
4. Preply Tutoring Platform.
<https://preply.com>
5. Wyzant Tutoring Marketplace.
<https://www.wyzant.com>
6. Noon Academy – EdTech Learning Platform.
<https://www.noonacademy.com>
7. Superprof Tutor Network.
<https://www.superprof.com>
8. Stripe Secure Payment API Documentation.
<https://stripe.com/docs>
9. VerifyMe Global ID Verification Services.
<https://www.verifyme.ng>
10. IEEE Software Engineering Body of Knowledge (SWEBOk).
<https://www.computer.org/education/bodies-of-knowledge/software-engineering>