ENGLISH LANGUAGE SCHOOL MANAGEMENT

PLATFORM

Technical Presentation – Microservices Architecture

Academic Project :

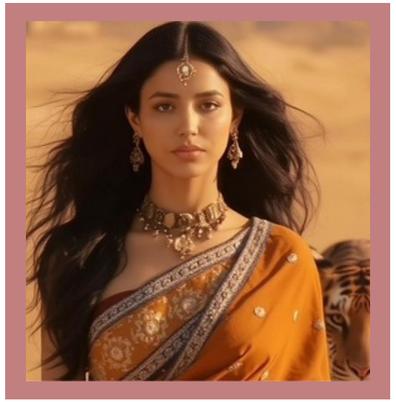
Applications web distribuées

Academic Year: 2025–2026

Table of CONTENTS

Team MEMBERS

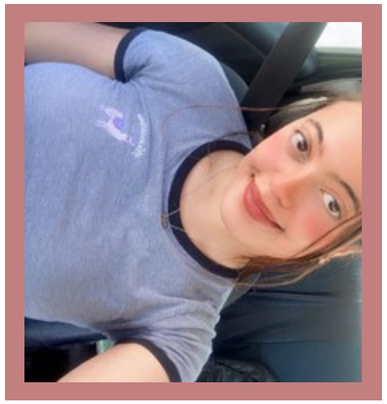
RAGHAD KHEDHIRI



NADINE RAZKI

AYMEN BEN ABDALLAH

MED KHALIL ESSOURI



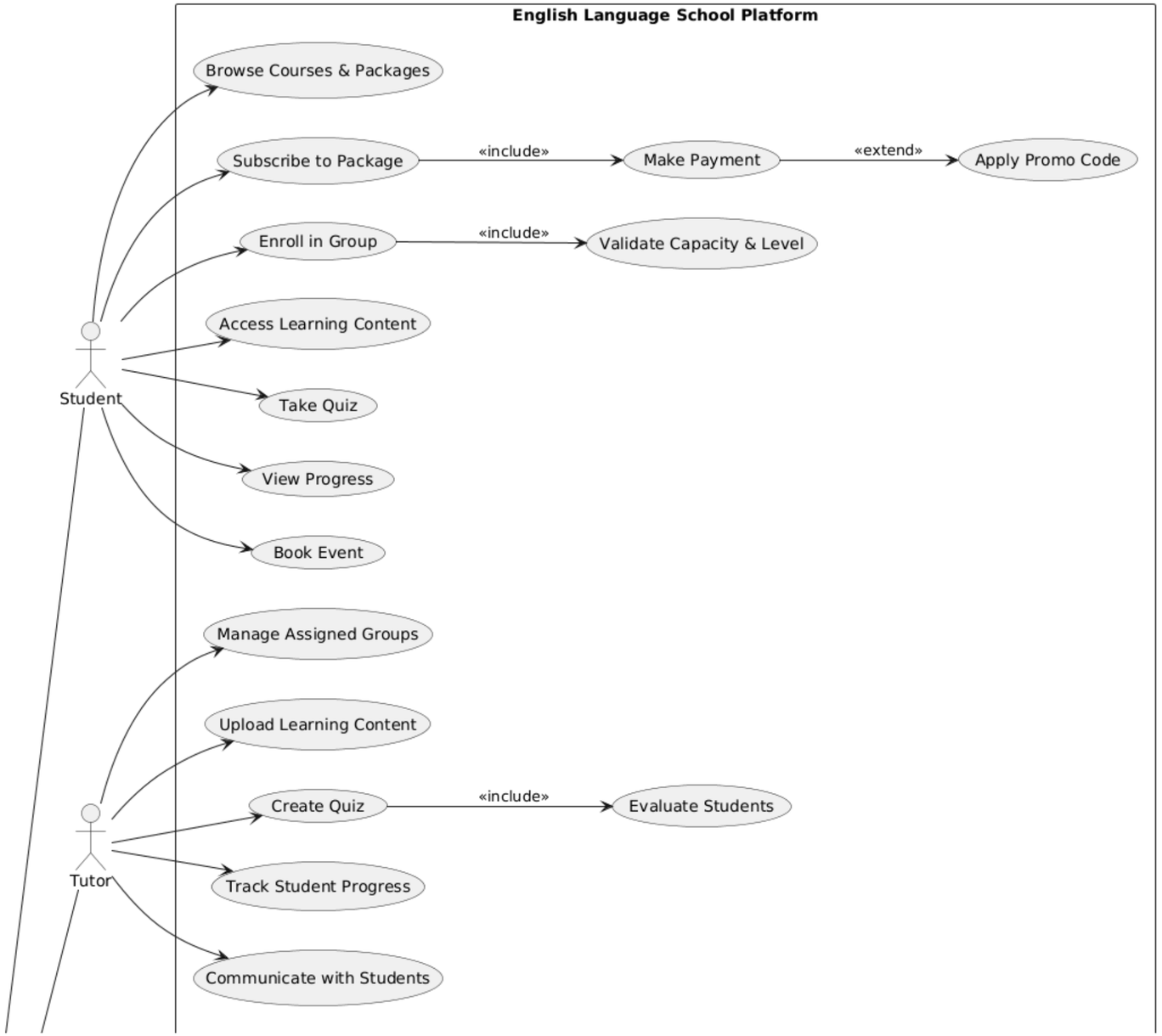
YASMINE OUERTATANI

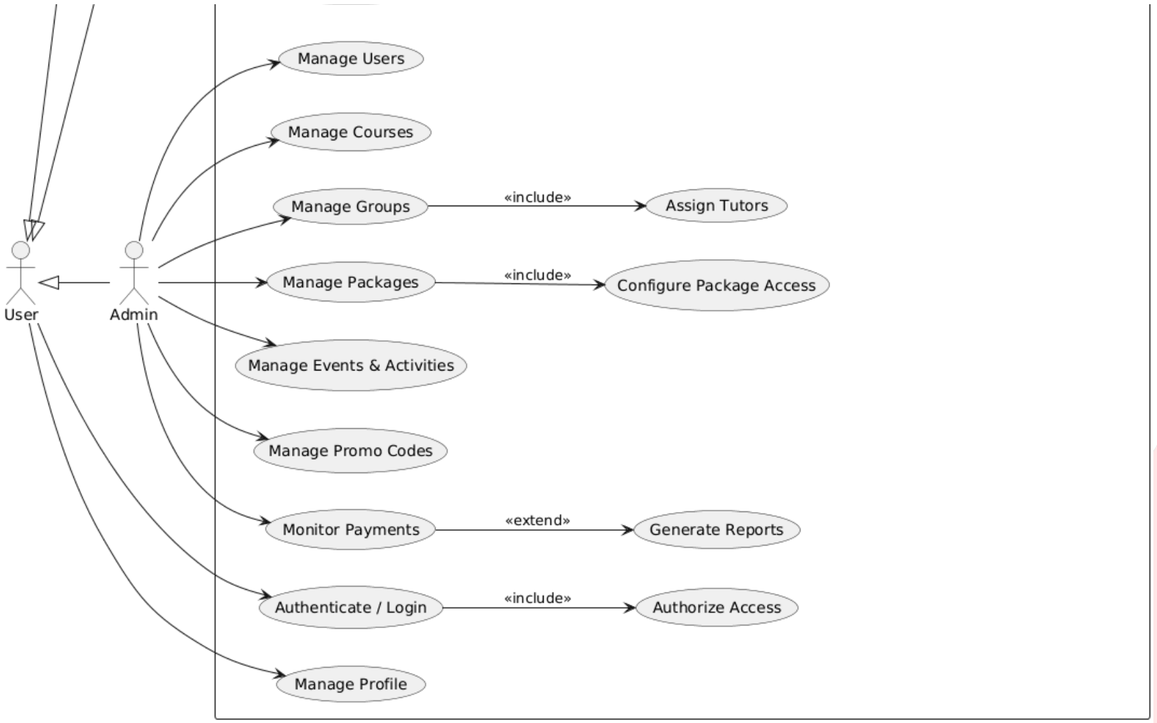
KHALIL SAHNOUN

Project Objective

The project is a microservices-based platform for managing a language school, covering user management, courses, groups, packages, payments, and events. Each business domain is an independent microservice, ensuring scalability, flexibility, and secure communication. The system serves students, tutors, and administrators, enabling enrollment, content access, academic management, and platform supervision.

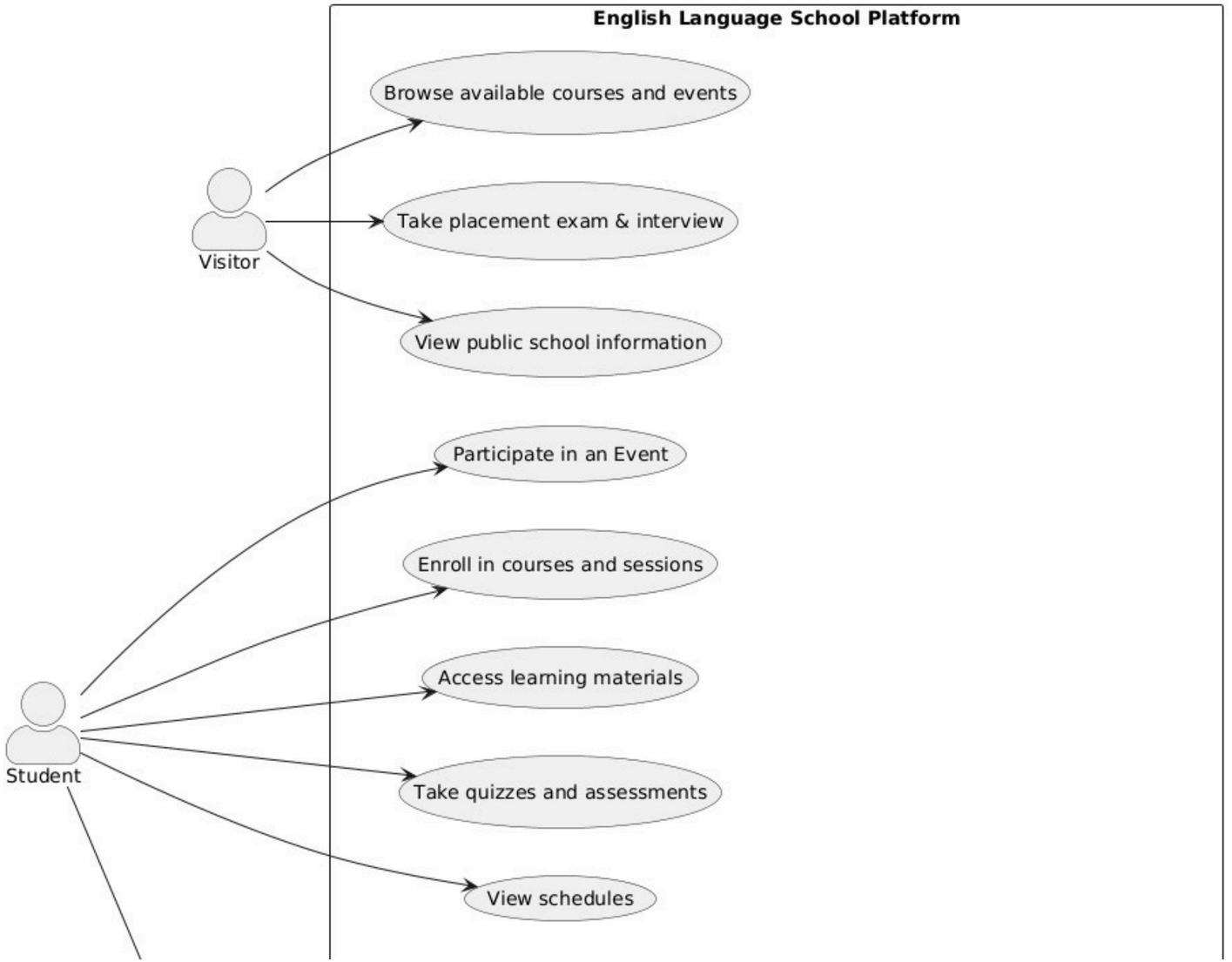
USE CASE DIAGRAM

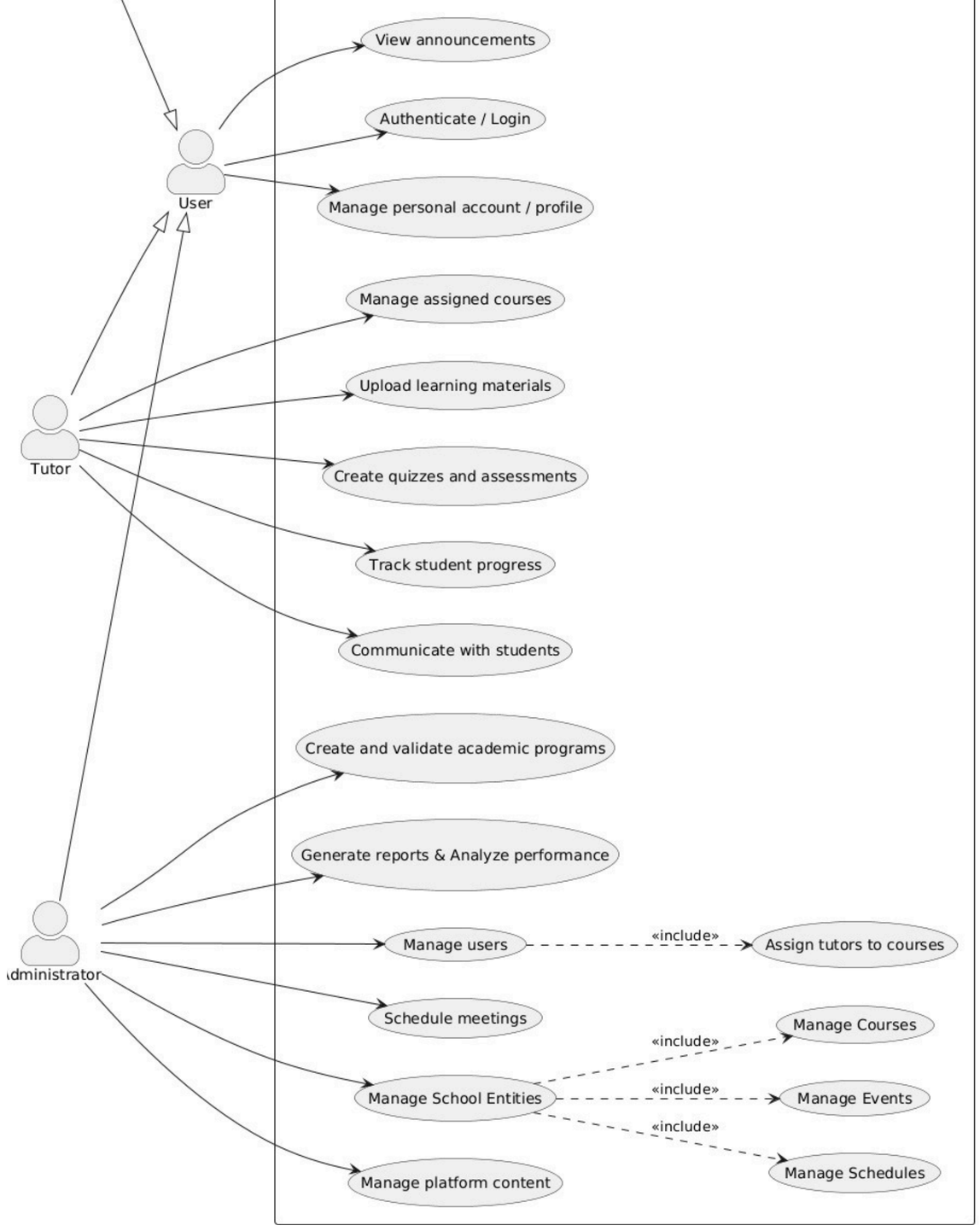




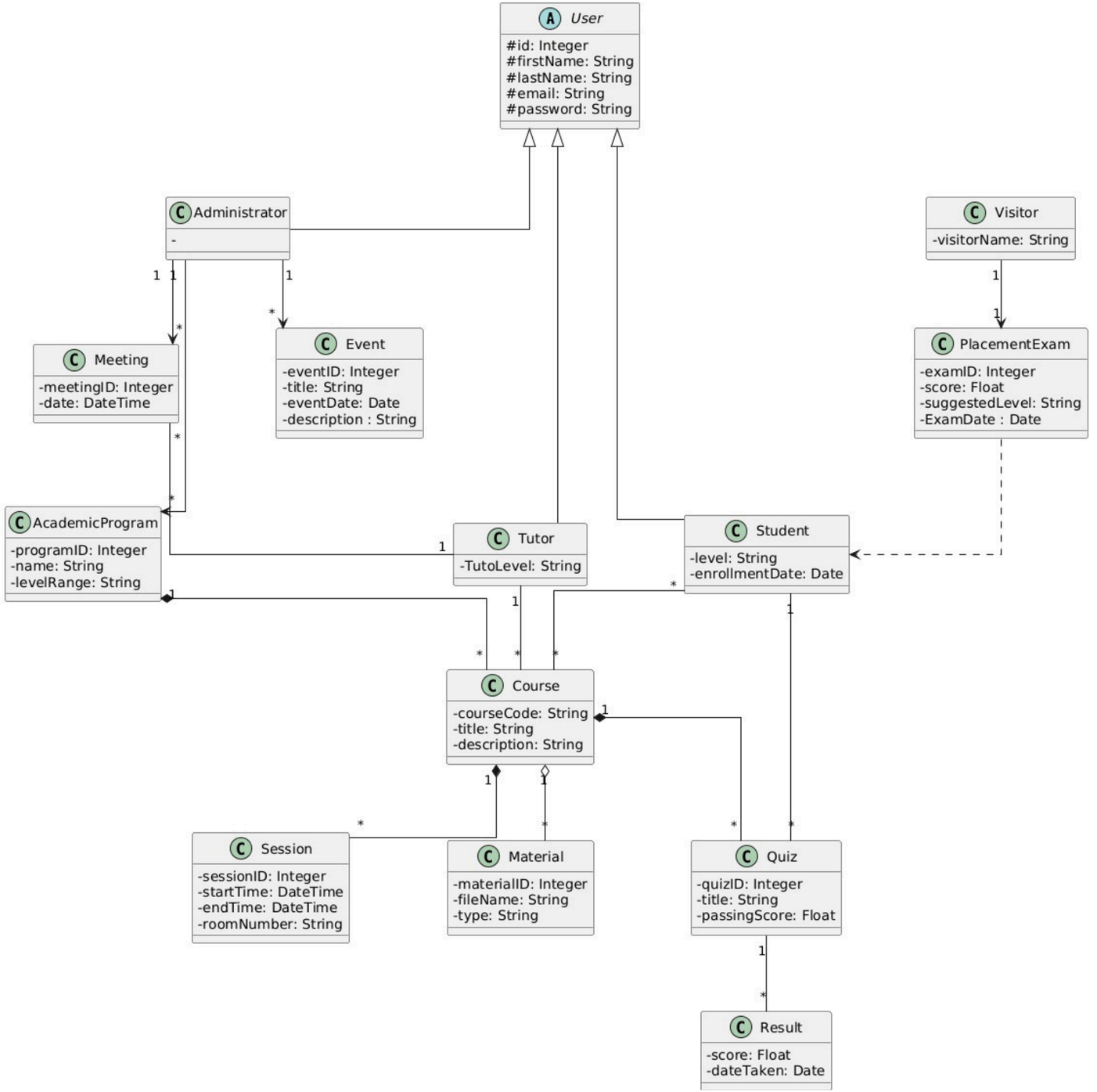
Global Architecture

USE CASE DIAGRAM

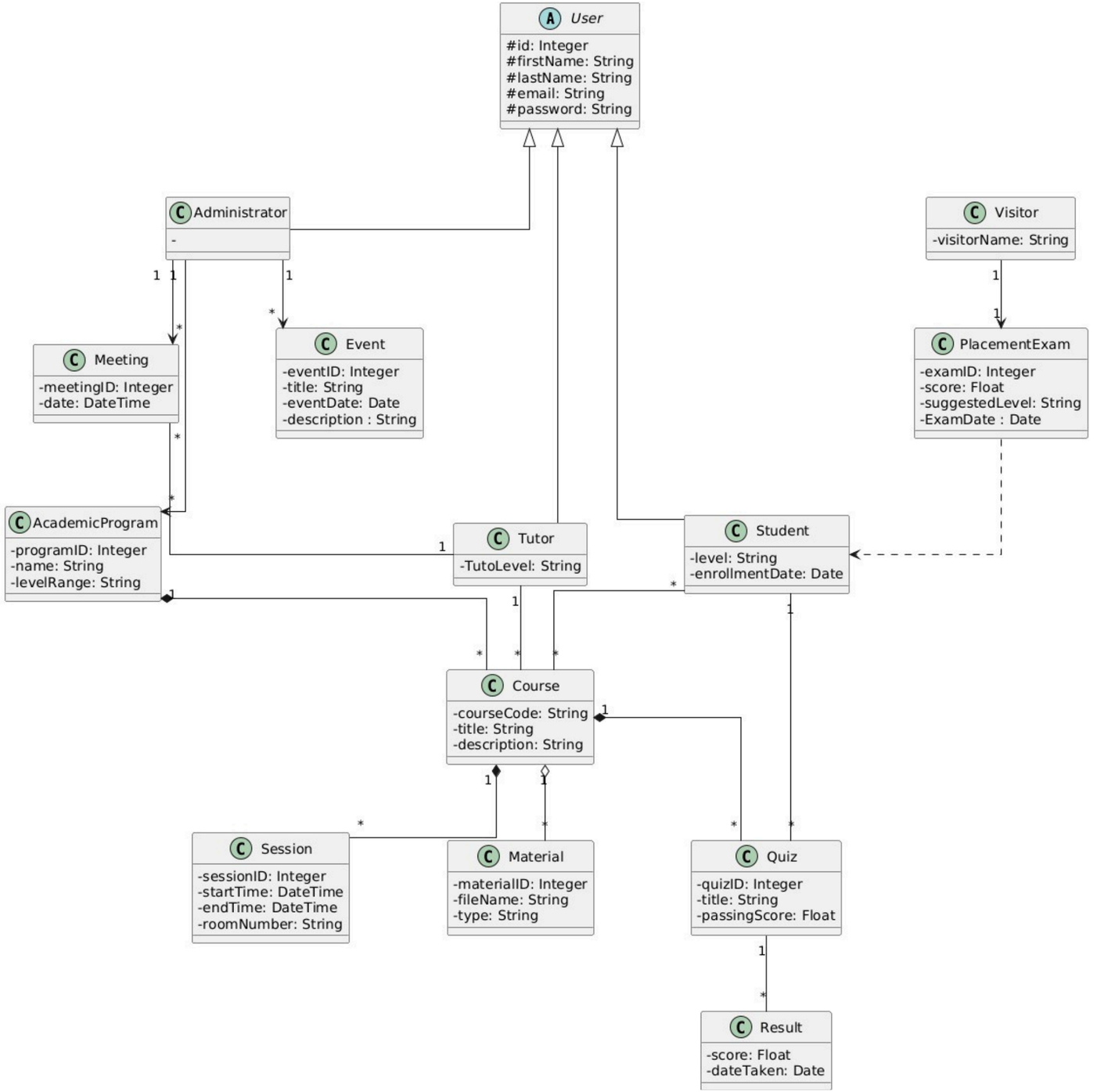



View announcements
Authenticate / Login
Manage personal account / profile
User
Manage assigned courses
Upload learning materials
Create quizzes and assessments
Track student progress
Communicate with students
Tutor
Create and validate academic programs
Generate reports & Analyze performance
Manage users
«include»
Assign tutors to courses
Schedule meetings
Manage School Entities
«include»
Manage Courses
«include»
Manage Events
«include»
Manage Schedules
Manage platform content
dministrator

CLASS DIAGRAM

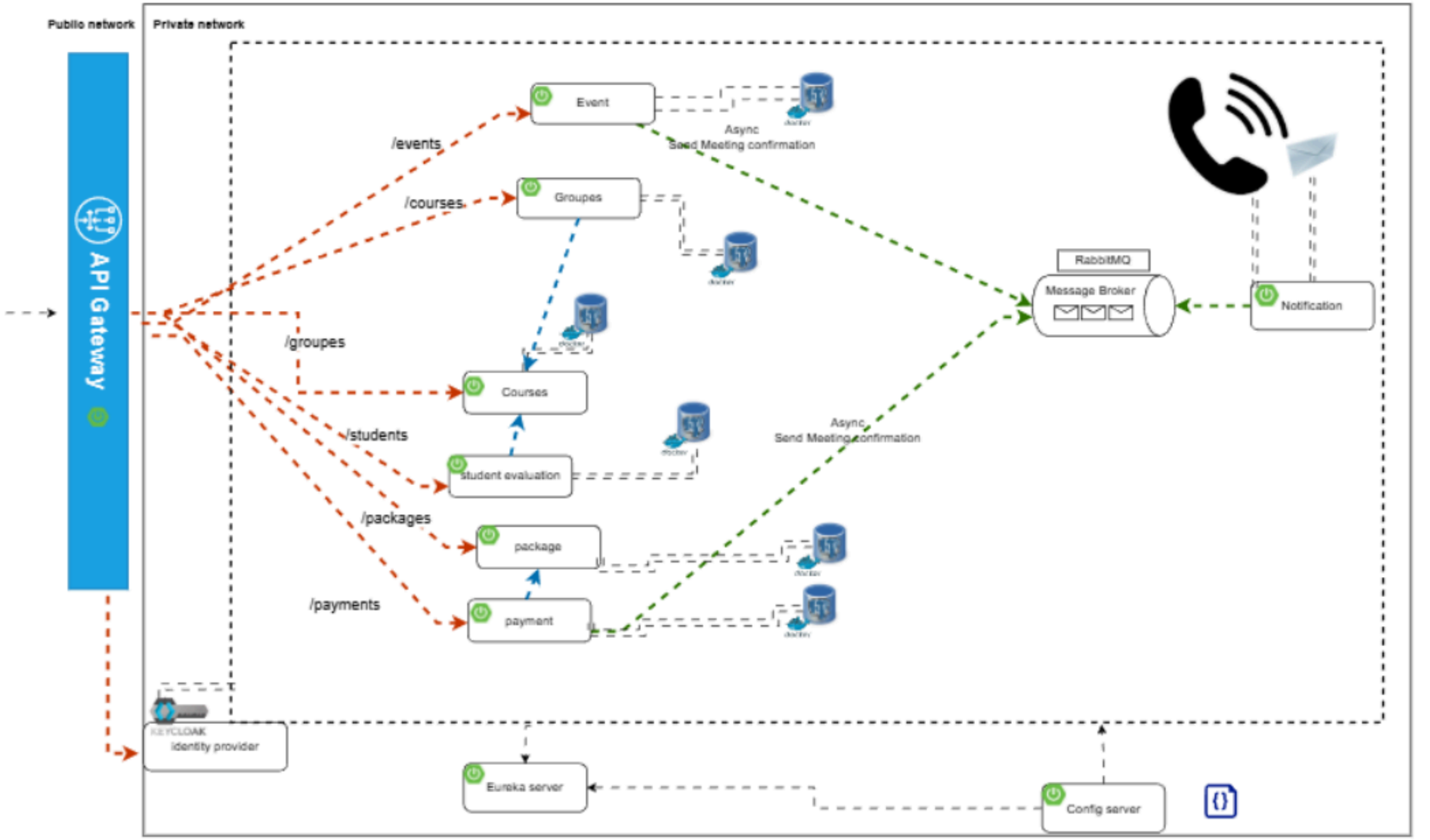


CLASS DIAGRAM



Overall architecture

EnglishForU



Microservices Description

Microservice Name	Responsibility	Technology Used
User Service	Management of users (students, teachers,	Spring Boot, REST API
Auth Service	User authentication and authorization, role and permission management	Keycloak, OAuth2, JWT
Course Service	Management of courses, levels, programs, and	Spring Boot, REST API
Enrollment Service	Management of student enrollments in courses and	Spring Boot, REST API
Schedule Service	Scheduling of sessions, course timetables, and	Spring Boot, REST API
Payment Service	Management of payments, invoices, and financial	Spring Boot, REST API
Notification Service	Sending notifications (emails, confirmations, reminders) to	Spring Boot, SMTP / REST
API Gateway	Single entry point of the system, routing requests to	Spring Cloud Gateway
Frontend Service	User interface for students, teachers, and administrators	Angular

Communication & Data Flow

1. Communication Protocols :

- **REST APIs :**

Primary communication between microservices (User, Course, Enrollment, Payment, etc.).

2. Main Flows Between Services :

User Service ↔ Auth Service → Login, authentication, and role validation.

User Service ↔ Course Service → Students enroll in courses, tutors manage content.

Enrollment Service ↔ Payment Service → Enrollment confirmed only after successful payment .

Enrollment Service ↔ Schedule Service → Synchronizes student enrollment with timetables and group capacity.

Notification Service ↔ All Services → Sends confirmations, reminders, and progress updates.

API Gateway ↔ All Services → Single entry point, routes requests securely to the right microservice.

Conclusion

The chosen microservices architecture ensures scalability, clear separation of concerns, and independent deployment of services.
The use of Docker, REST APIs, and asynchronous messaging provides a robust and production-ready system.