

# Service Management System

Kuruguntla Asritha

[asritha.kuruguntla@gmail.com](mailto:asritha.kuruguntla@gmail.com)

# Introduction:

## Problem Statement

Service operations often rely on manual coordination across request handling, technician assignment, and billing, resulting in slow response times and limited visibility.

## Technology Stack

### Frontend

Framework: Angular 20.3.0

Language: TypeScript 5.9.2

Build Tool: Angular CLI 20.3.5

Styling: Bootstrap 5.3.8

Reactive Programming: RxJS 7.8.0

Testing: Jasmine 5.9.0, Karma 6.4.0

Package Manager: npm

Backend (Microservices Architecture)

Language: Java 17

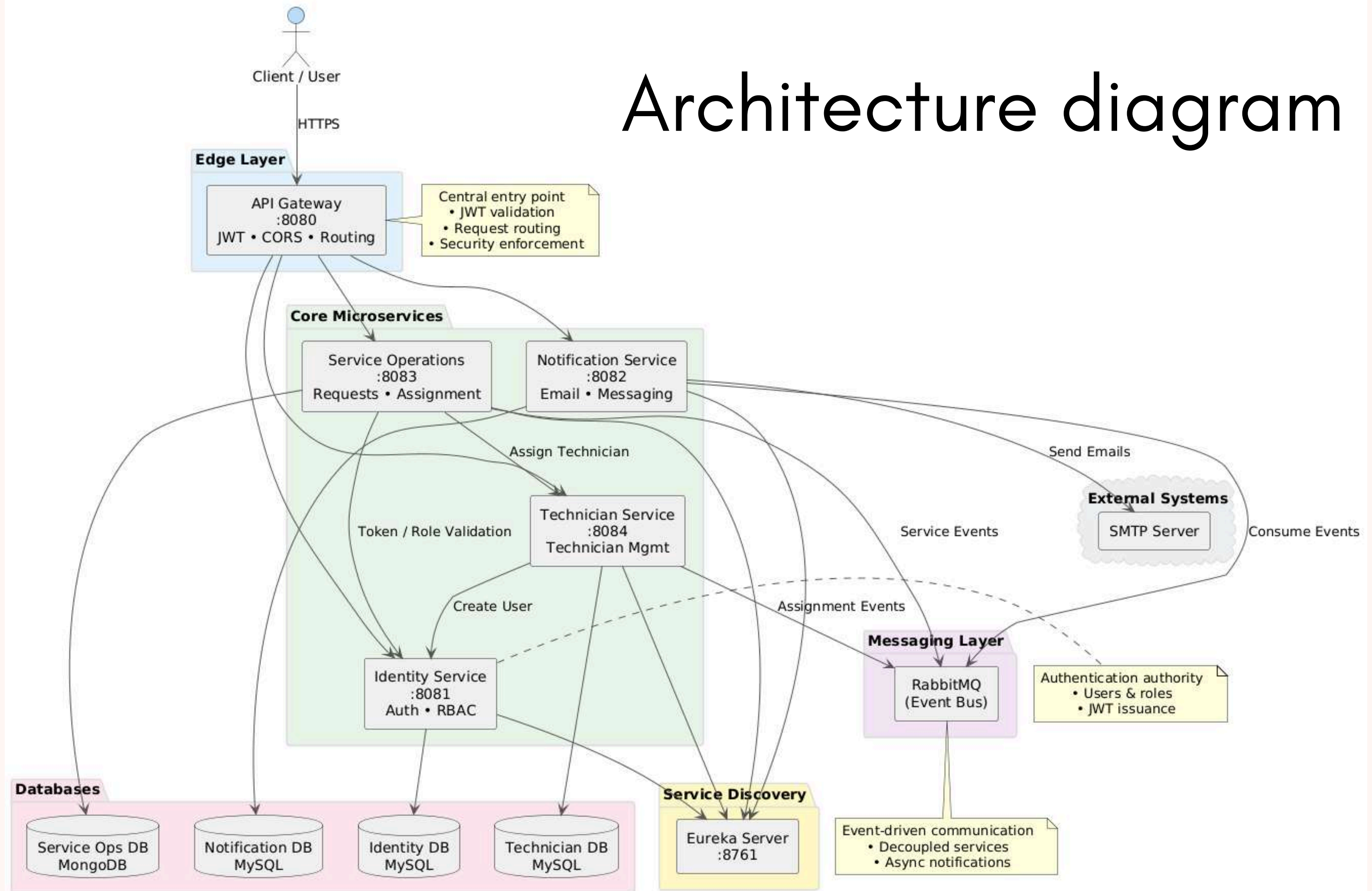
Framework: Spring Boot 3.2.5

Cloud Framework: Spring Cloud 2023.0.3

Build Tool: Maven

Containerization: Docker

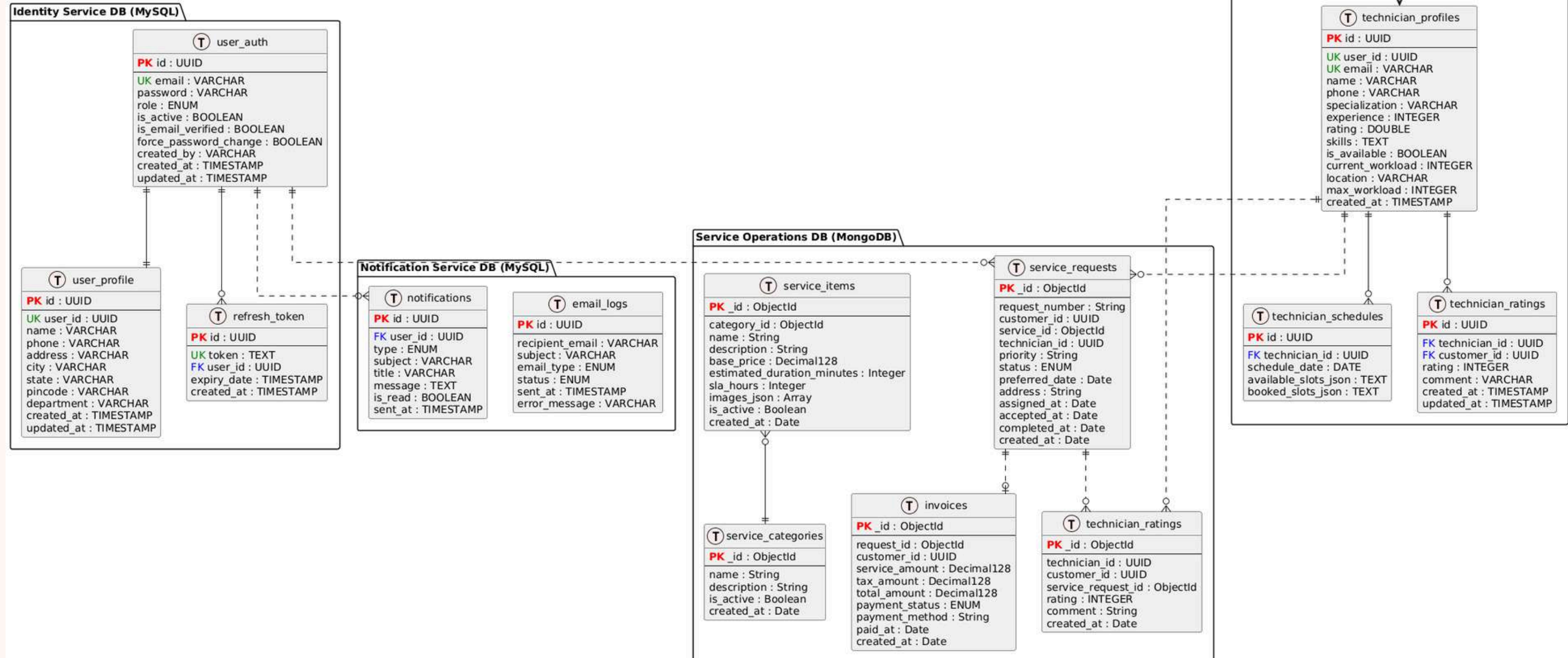
# Architecture diagram



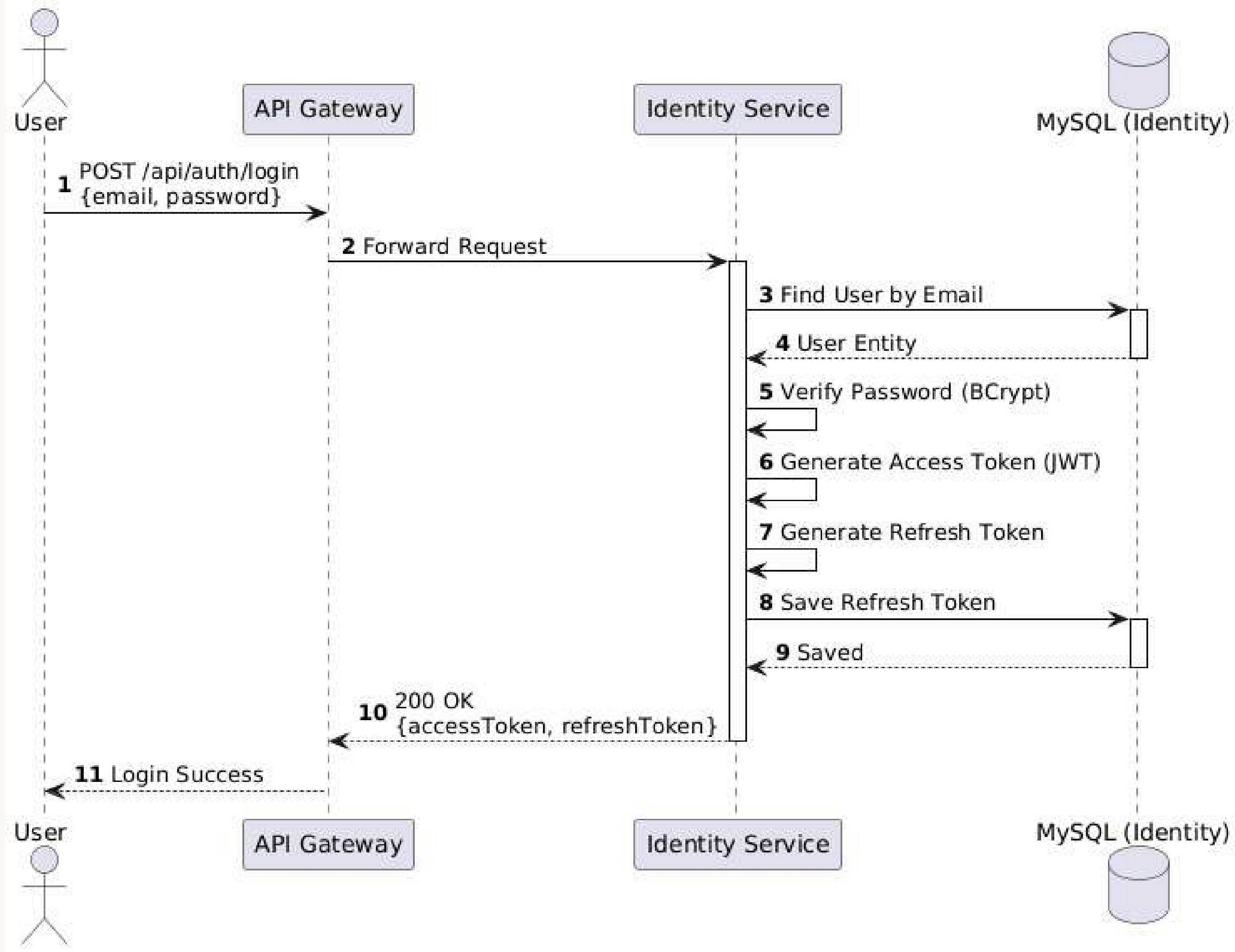


# ER DIAGRAM

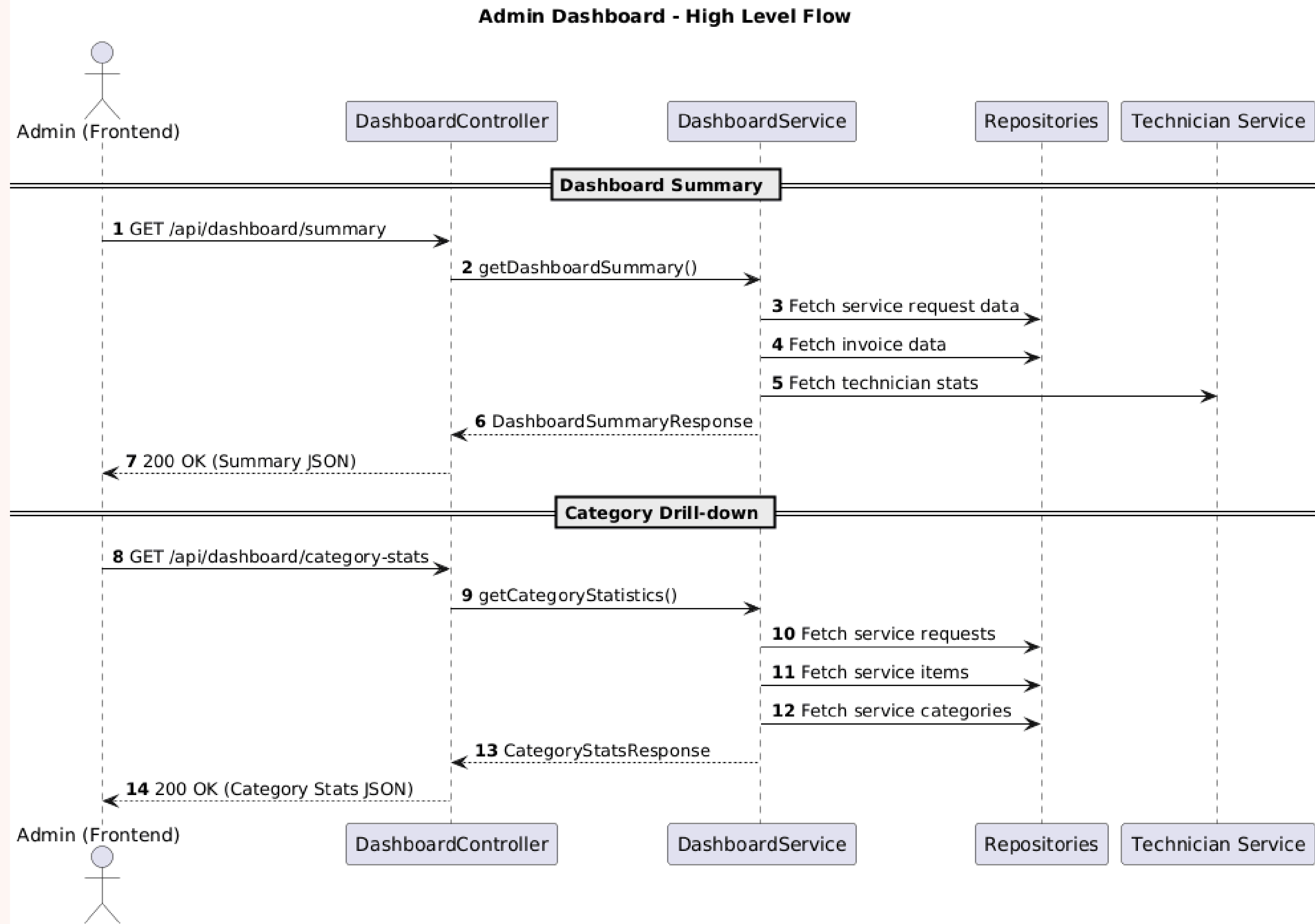
Databases used: MySQL, MONGODB



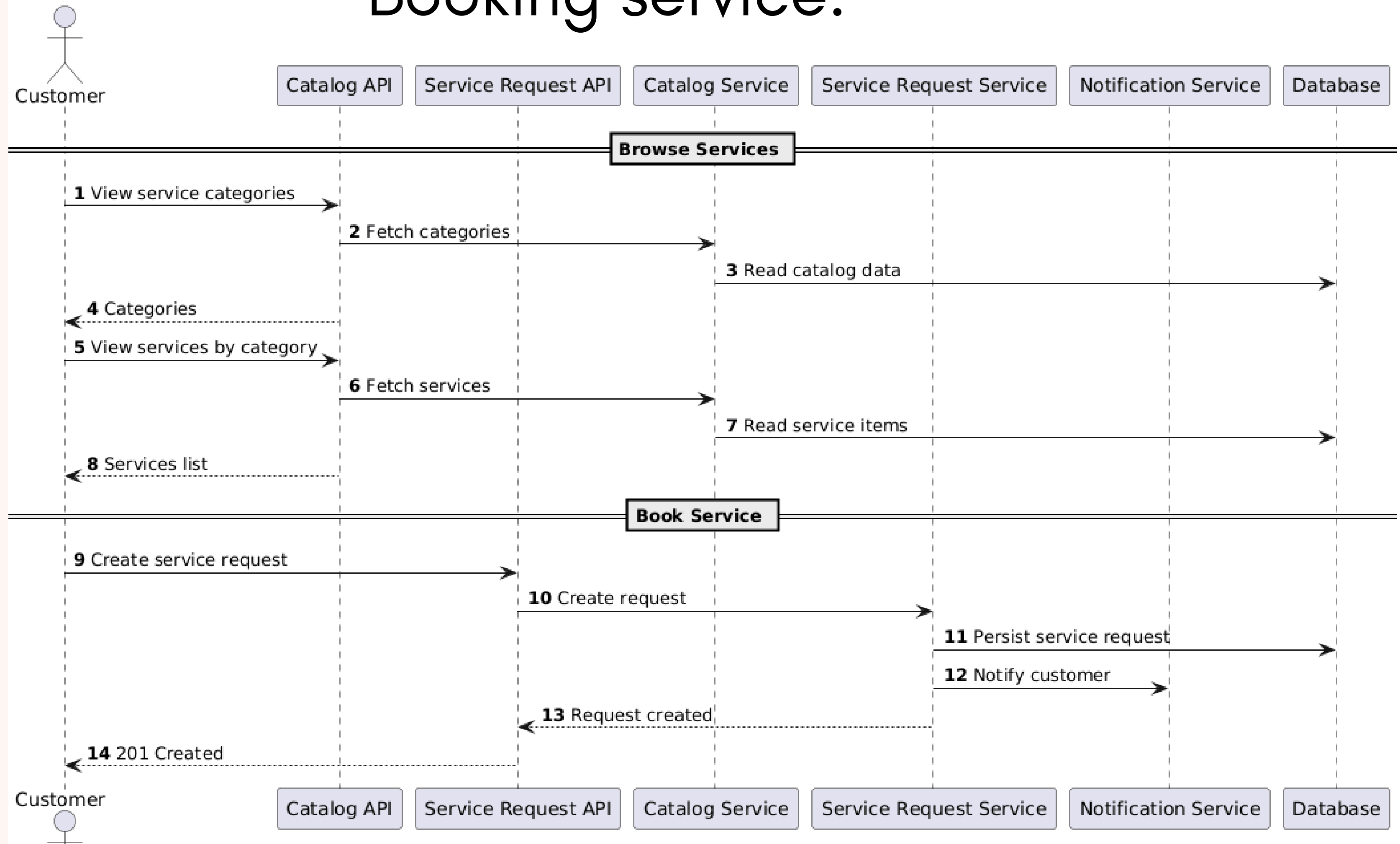
# Login Flow:



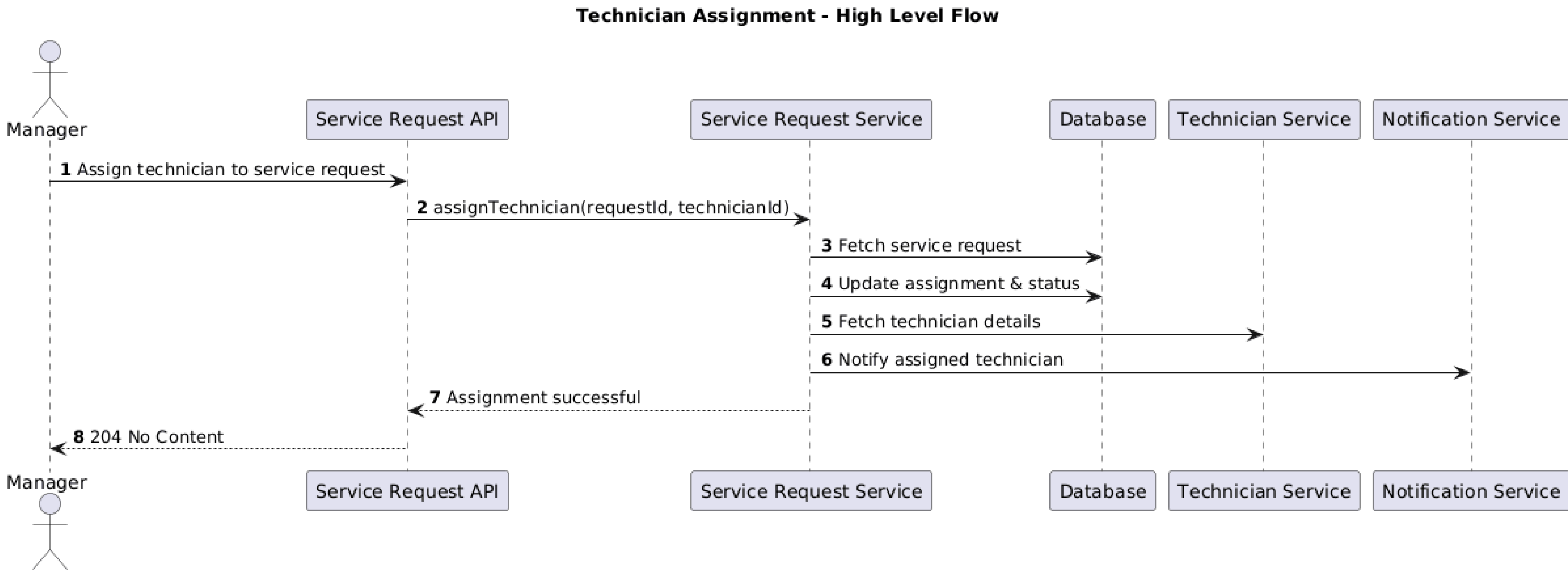
# Admin Dashboard:



# Booking service:

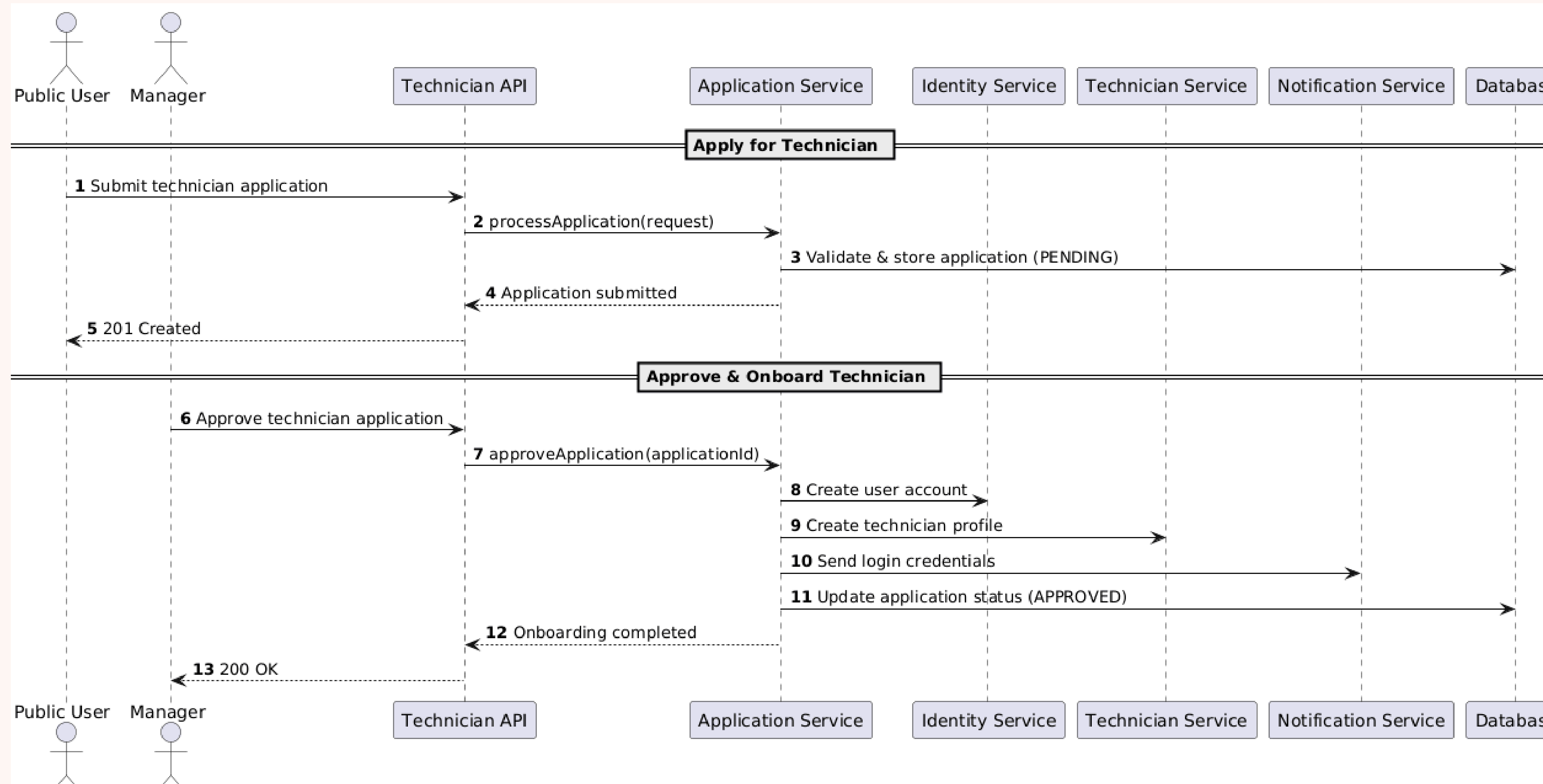


# Technician assignment flow:

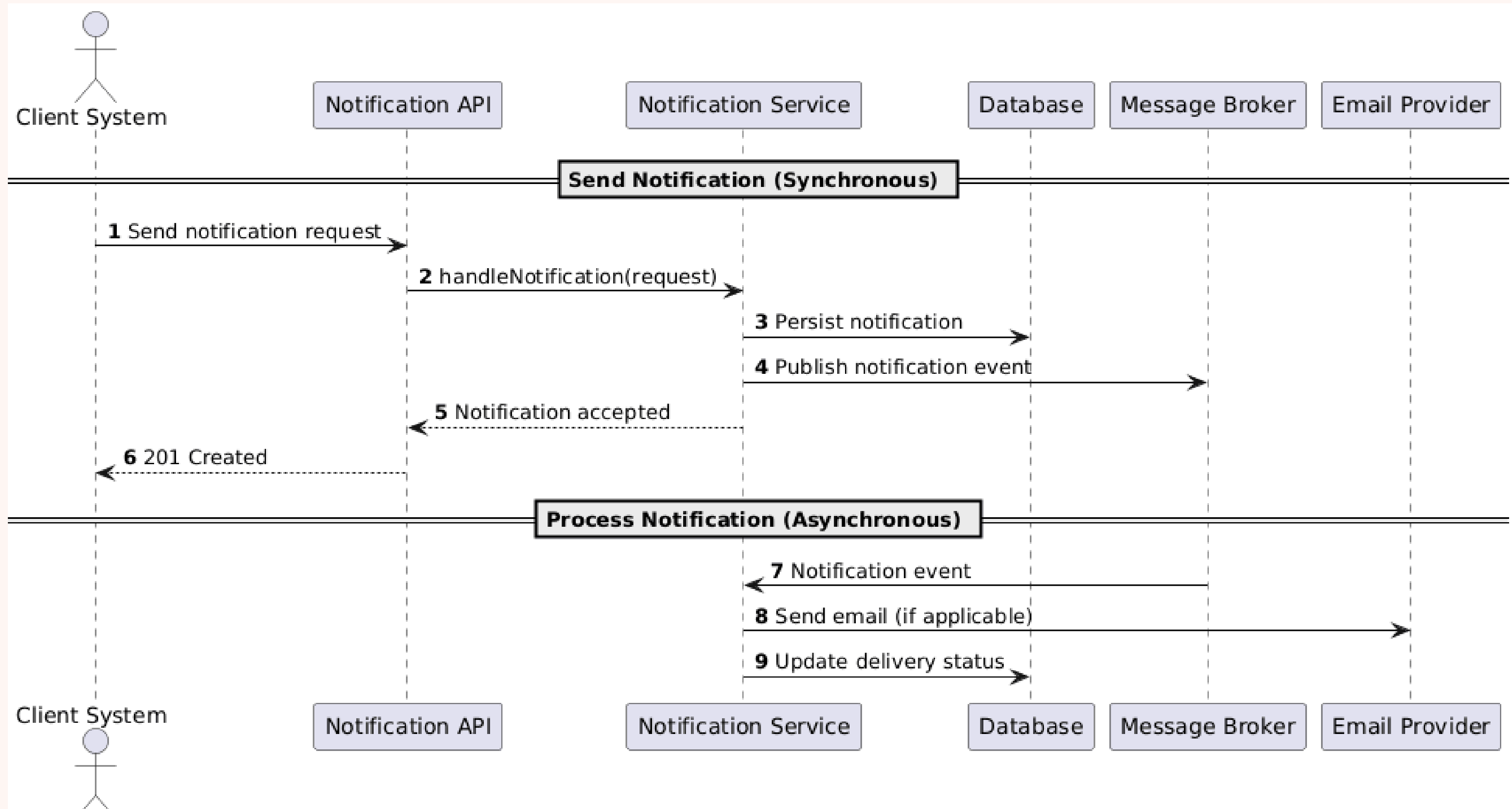




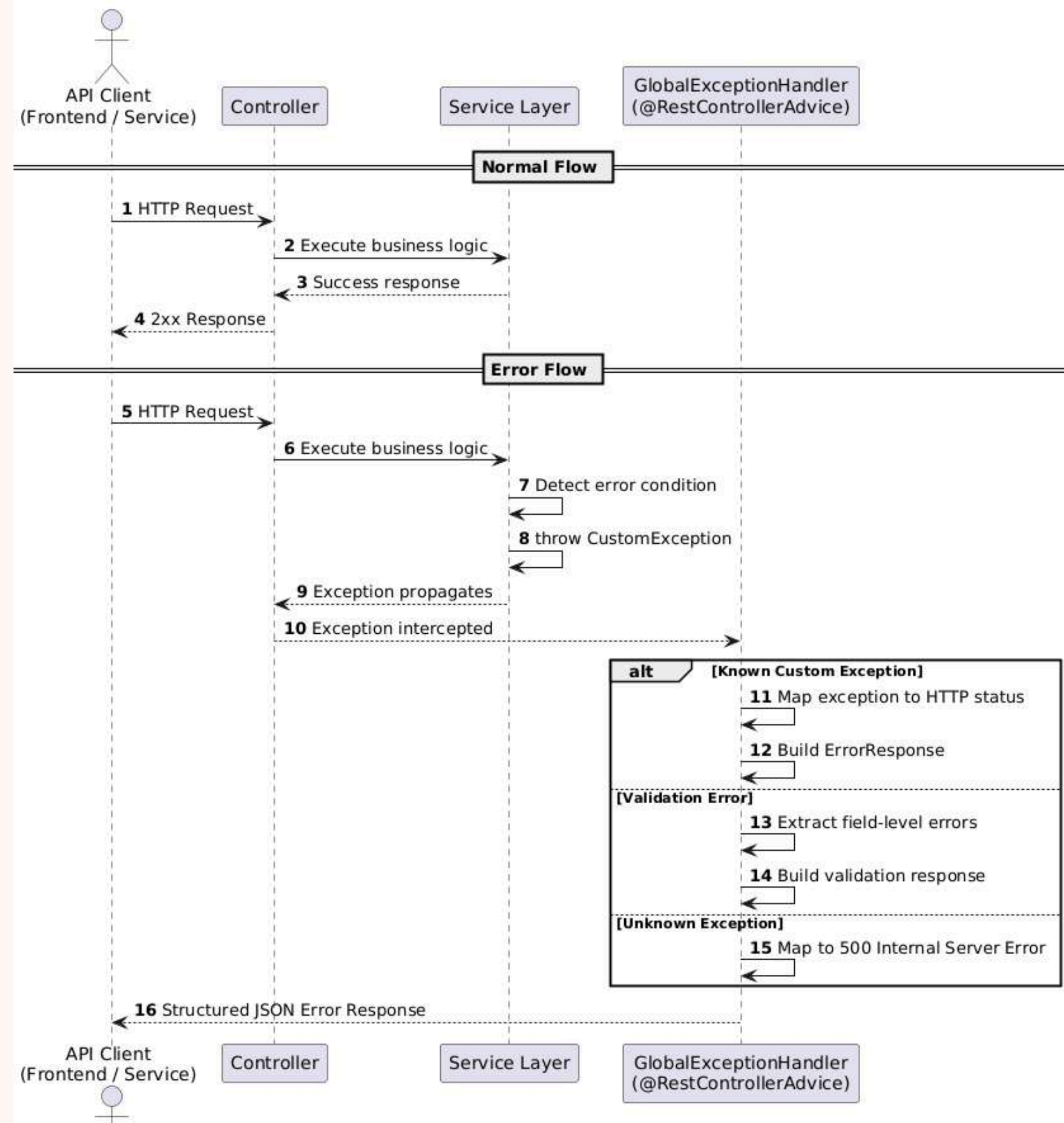
# Technician service flow:



# Notifications flow:



# Exception Handling:



# Business rules

- Reapplication cooldown: Technician cannot reapply within 30 days after rejection
- Rejection requirements
- **Smart Technician Suggestion Algorithm**
- Only technicians with  $\text{currentWorkload} < \text{maxWorkload}$  are considered available
- Workload updates
- Users cannot register with an email that already exists in the system

## Code quality and testing (sonar):

### Main Branch Summary

8.2k Lines of Code  • Version 1.0.0 • Last analysis 18 hours ago •  [bb6a23fd](#)

Quality Gate: [Sonar way](#) 



## Passed

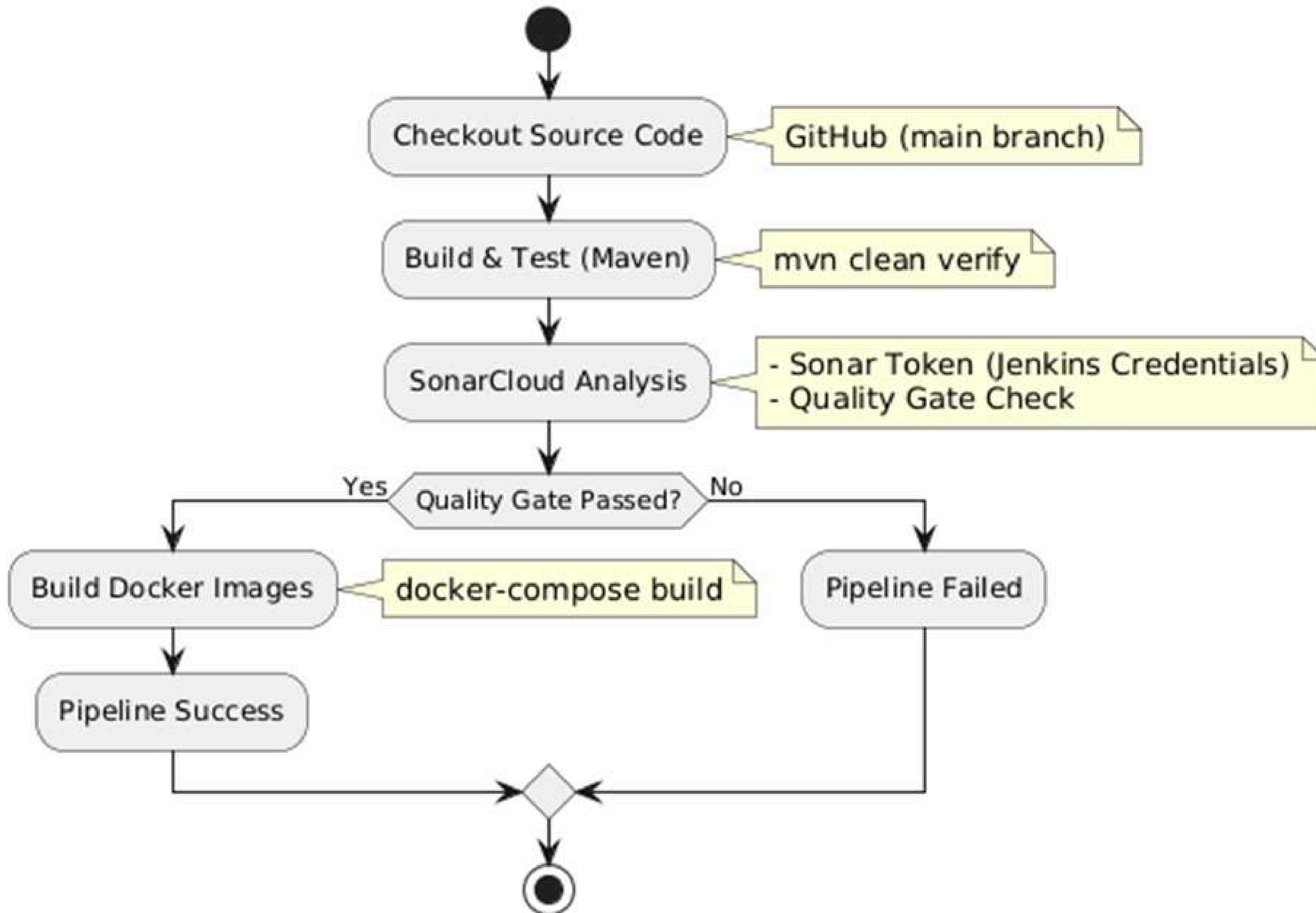
Coverage

## 90.6%

No conditions set  
on 1.5k Lines to cover



## Jenkins CI Pipeline - Service Management System Backend



# Challenges faced:

- Planning & Architecture Decisions
- Securing Inter-Service Communication with OpenFeign
- Centralized Authentication & Authorization using API Gateway
- Improving code quality by analysis with sonar



# Improvements :

- Google Maps API: Route planning and distance calculation
- Calendar Integration: Sync with Google Calendar
- Payment Gateways: Stripe, Razorpay, PayPal
- API Gateway Features : Request throttling, Response caching, Load balancing
- Payment Reminders: Automated payment reminder emails/SMS
- Dynamic Pricing: Implement surge pricing based on demand
- WebSocket Integration: Real-time updates without polling
- Smart Technician Assignment: ML-based optimal technician matching
- Caching Layer: Implement Redis for caching frequently accessed data
- Two-Factor Authentication (2FA): Add OTP-based authentication for sensitive operations
- API Rate Limiting: Implement rate limiting to prevent API abuse and DDoS attacks

**Thank  
You**