SERVICE MANAGEMENT SYSTEM

POLAKU JOSHIKA SREE
Joshikasreepolaku@gmail.com

Introduction:

Problem Statement:

- Manual service request handling via calls.
- No centralized tracking of service requests andtechnician assignments.
- Difficulty managing technician availability andservice schedules.
- Delayed service resolution and poor customerexperience.
- Have to manage monthly billing manually.

Proposed Solution:

- Centralized, cloud-ready Service Management Platform.
- Secure role-based access for Admin, Manager, Technician, and Customer.
- Automated service lifecycle management.
- Real-time dashboards and reports for operational visibility.

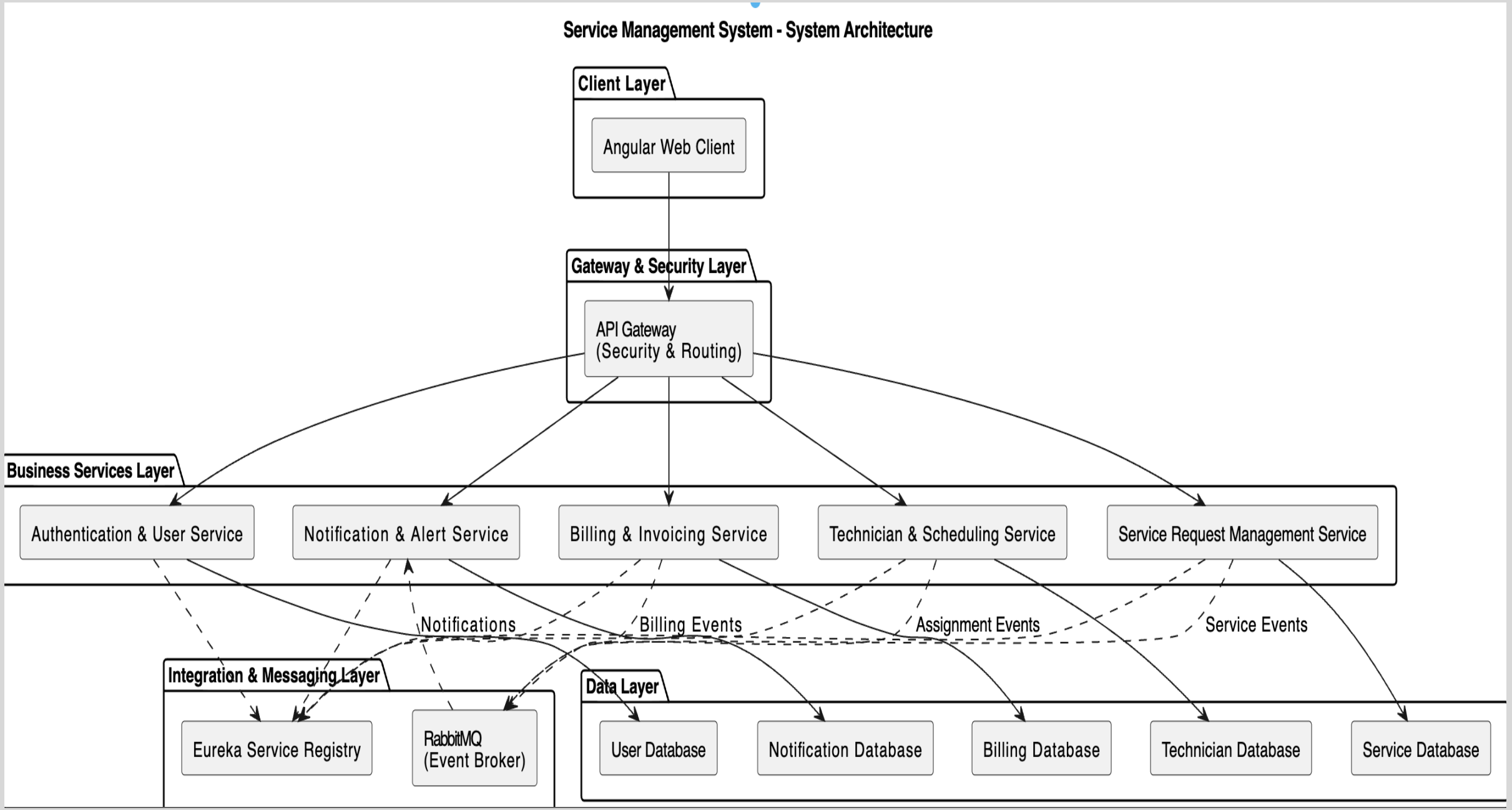
Technology Stack

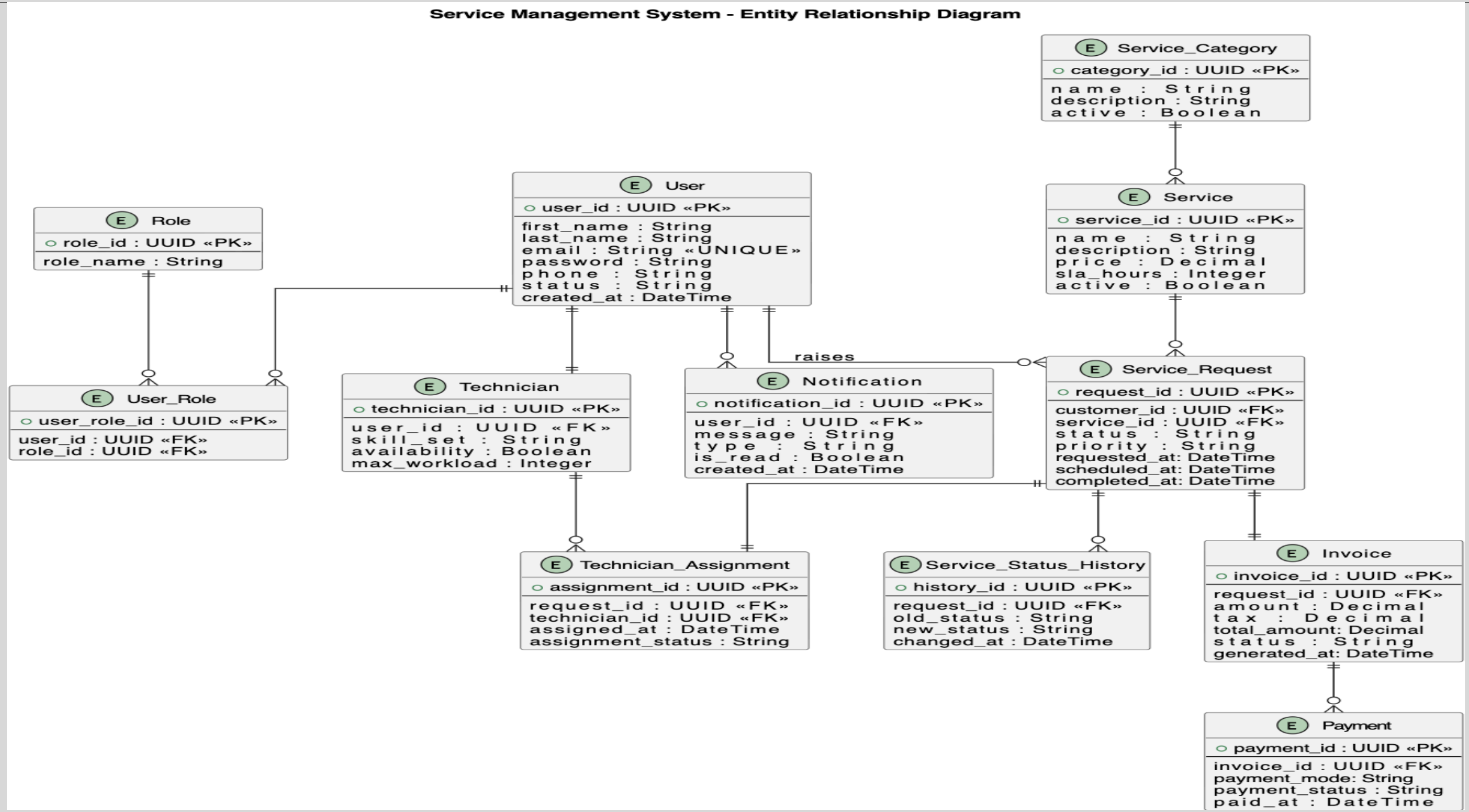
Backend

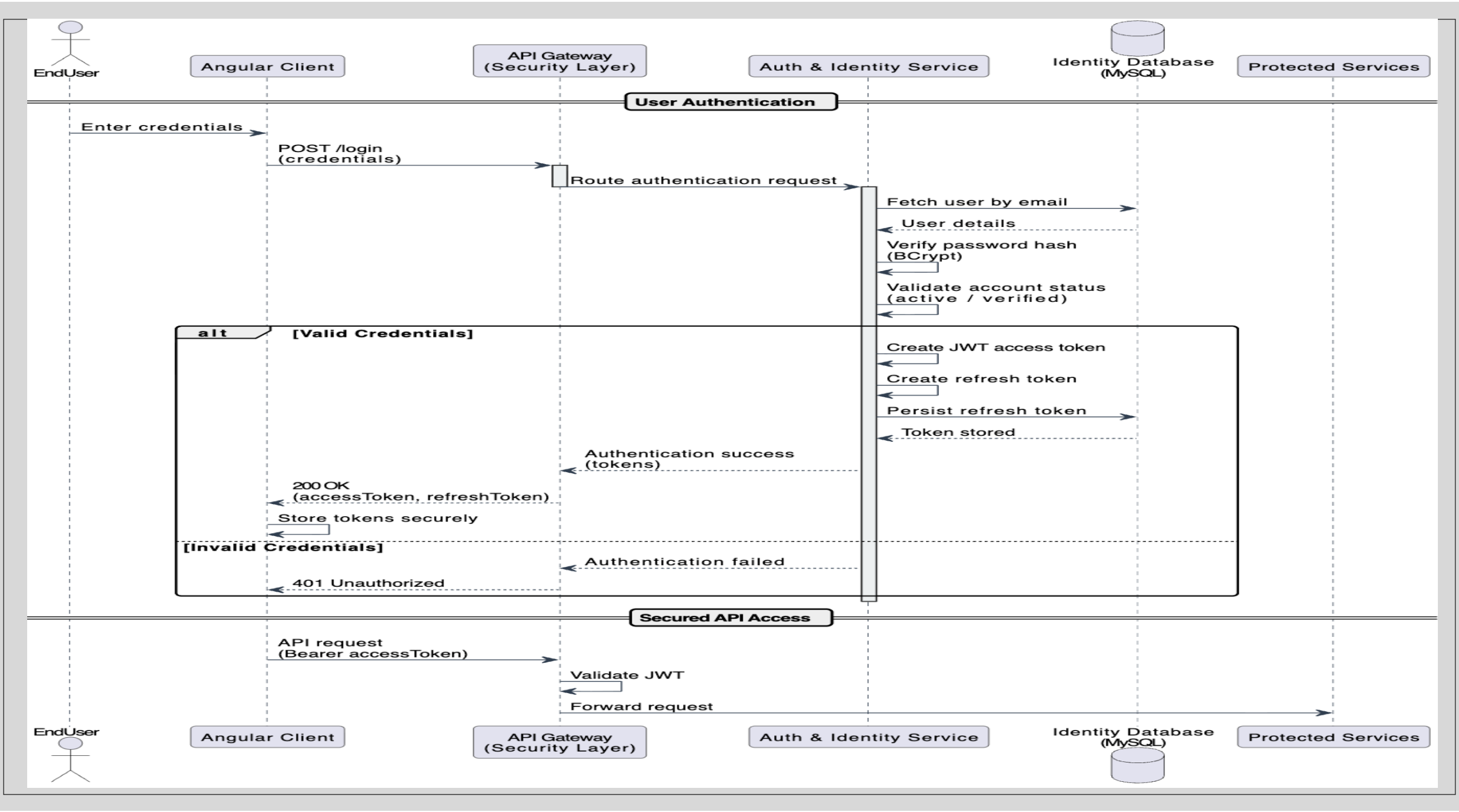
- **Language:** Java 17+
- **Framework:** Spring Boot 3.x
- **Architecture:** Microservices-ready architecture
- **API Design:** RESTful APIs using Spring Web
- **Database:** MongoDB
- **Security:** JWT-based Authentication
- **Authorization:** Role-Based Access Control (RBAC)
- **API Documentation:** Swagger
- **Build Tool:** Maven

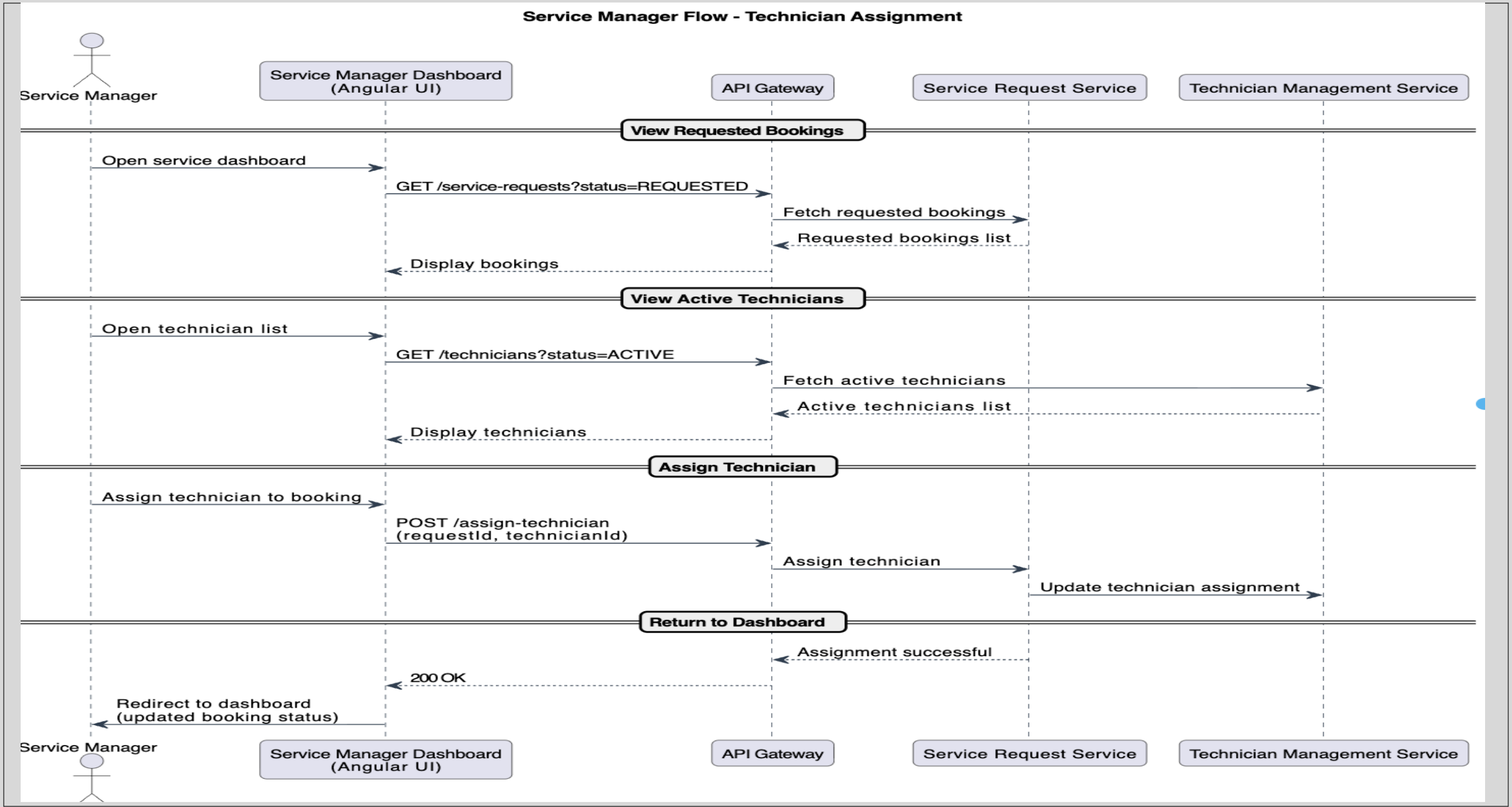
Frontend

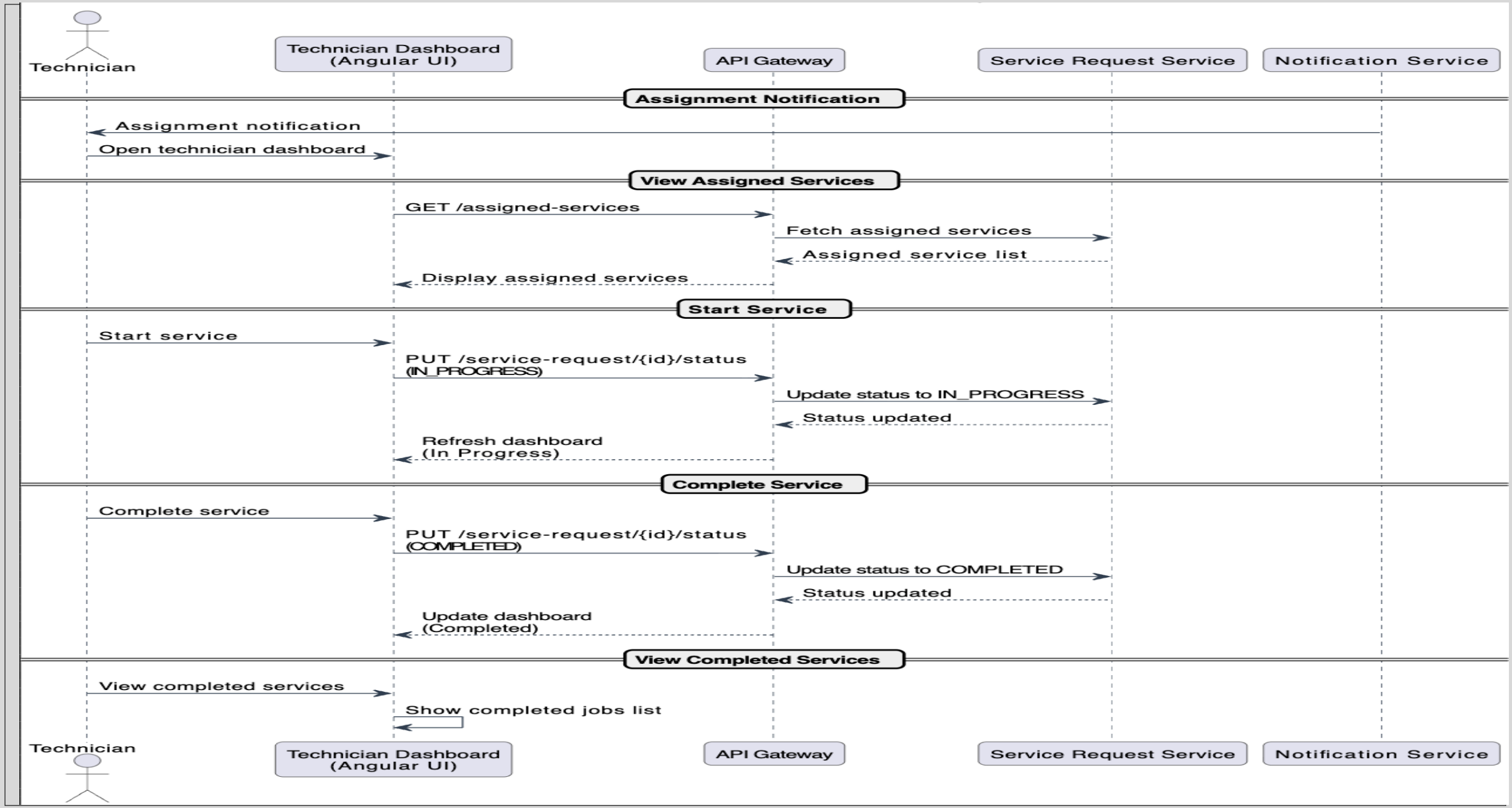
- **Framework:** Angular (Latest Version)
- **Language:** TypeScript
- **UI Design:** Angular Material / Bootstrap
- **Forms:** Reactive Forms with validation
- **HTTP Communication:** HttpClient
- **Security Handling:** HTTP Interceptors for JWT token attachment
- **Routing:** Role-based route guards

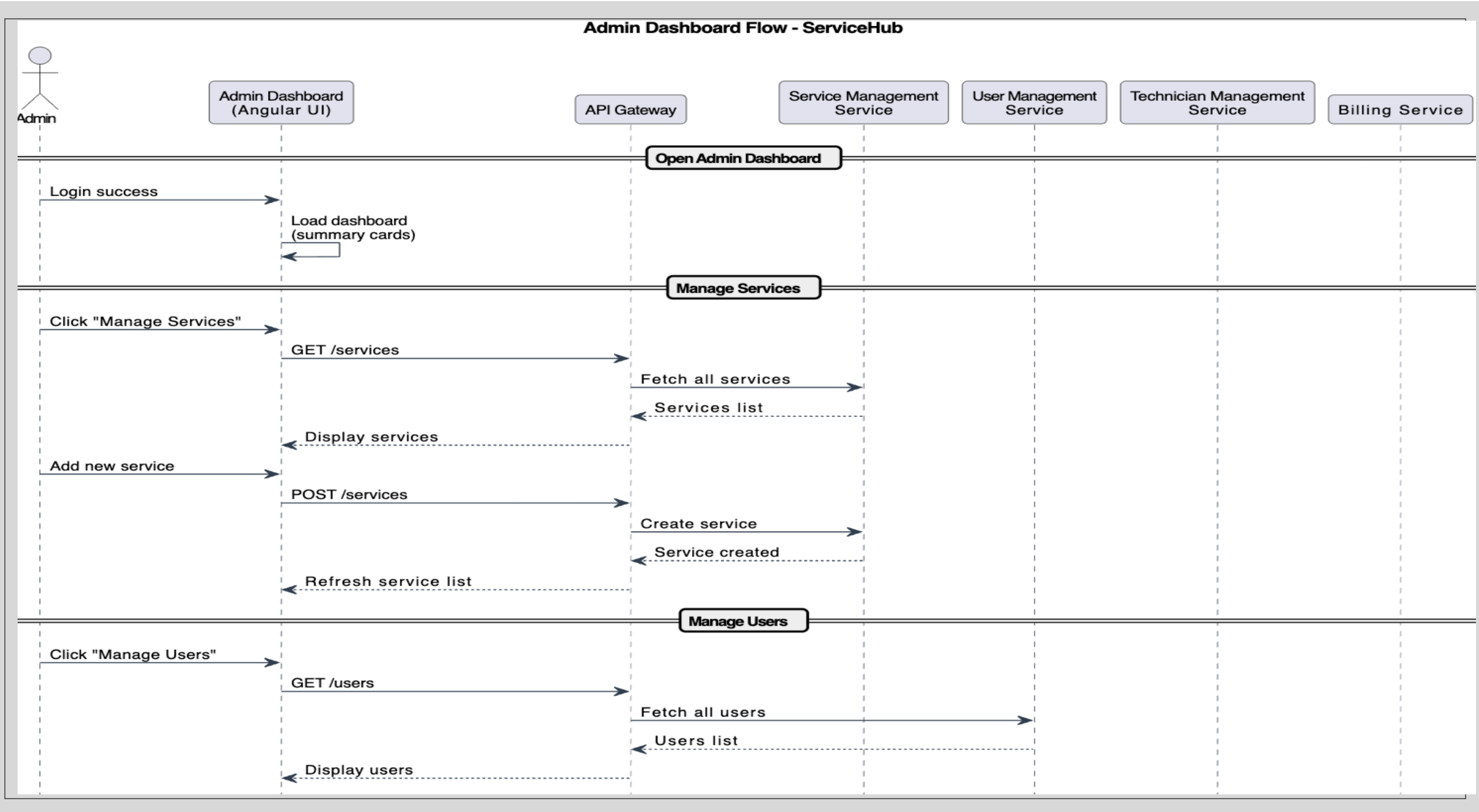

Service Management System - System Architecture
Client Layer
Angular Web Client
Gateway & Security Layer
API Gateway
(Security & Routing)
Business Services Layer
Authentication & User Service
Notification & Alert Service
Billing & Invoicing Service
Technician & Scheduling Service
Service Request Management Service
Notifications
Billing Events
Assignment Events
Service Events
Integration & Messaging Layer
Eureka Service Registry
RabbitMQ
(Event Broker)
Data Layer
User Database
Notification Database
Billing Database
Technician Database
Service Database

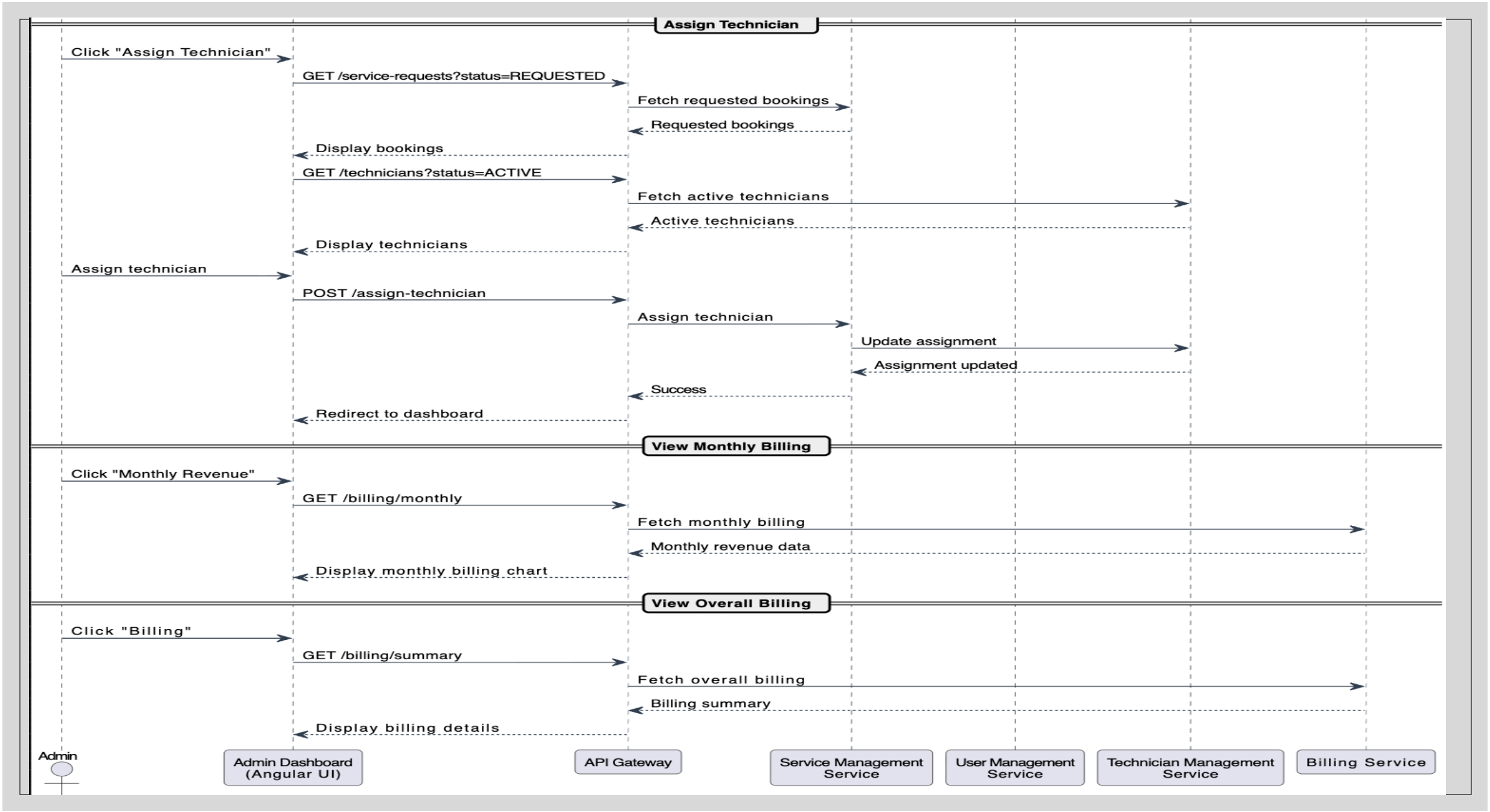

Service Management System - Entity Relationship Diagram
Service_Category
category_id : UUID «PK»
name : String
description : String
active : Boolean
User
user_id : UUID «PK»
first_name : String
last_name : String
email : String «UNIQUE»
password : String
phone : String
status : String
created_at : DateTime
Service
service_id : UUID «PK»
name : String
description : String
price : Decimal
sla_hours : Integer
active : Boolean
Role
role_id : UUID «PK»
role_name : String
raises
User_Role
user_role_id : UUID «PK»
user_id : UUID «FK»
role_id : UUID «FK»
Technician
technician_id : UUID «PK»
user_id : UUID «FK»
skill_set : String
availability : Boolean
max_workload : Integer
Notification
notification_id : UUID «PK»
user_id : UUID «FK»
message : String
type : String
is_read : Boolean
created_at : DateTime
Service_Request
request_id : UUID «PK»
customer_id : UUID «FK»
service_id : UUID «FK»
status : String
priority : String
requested_at: DateTime
scheduled_at: DateTime
completed_at: DateTime
Technician_Assignment
assignment_id : UUID «PK»
request_id : UUID «FK»
technician_id : UUID «FK»
assigned_at : DateTime
assignment_status : String
Service_Status_History
history_id : UUID «PK»
request_id : UUID «FK»
old_status : String
new_status : String
changed_at : DateTime
Invoice
invoice_id : UUID «PK»
request_id : UUID «FK»
amount : Decimal
tax : Decimal
total_amount: Decimal
status : String
generated_at: DateTime
Payment
payment_id : UUID «PK»
invoice_id : UUID «FK»
payment_mode: String
payment_status : String
paid_at : DateTime

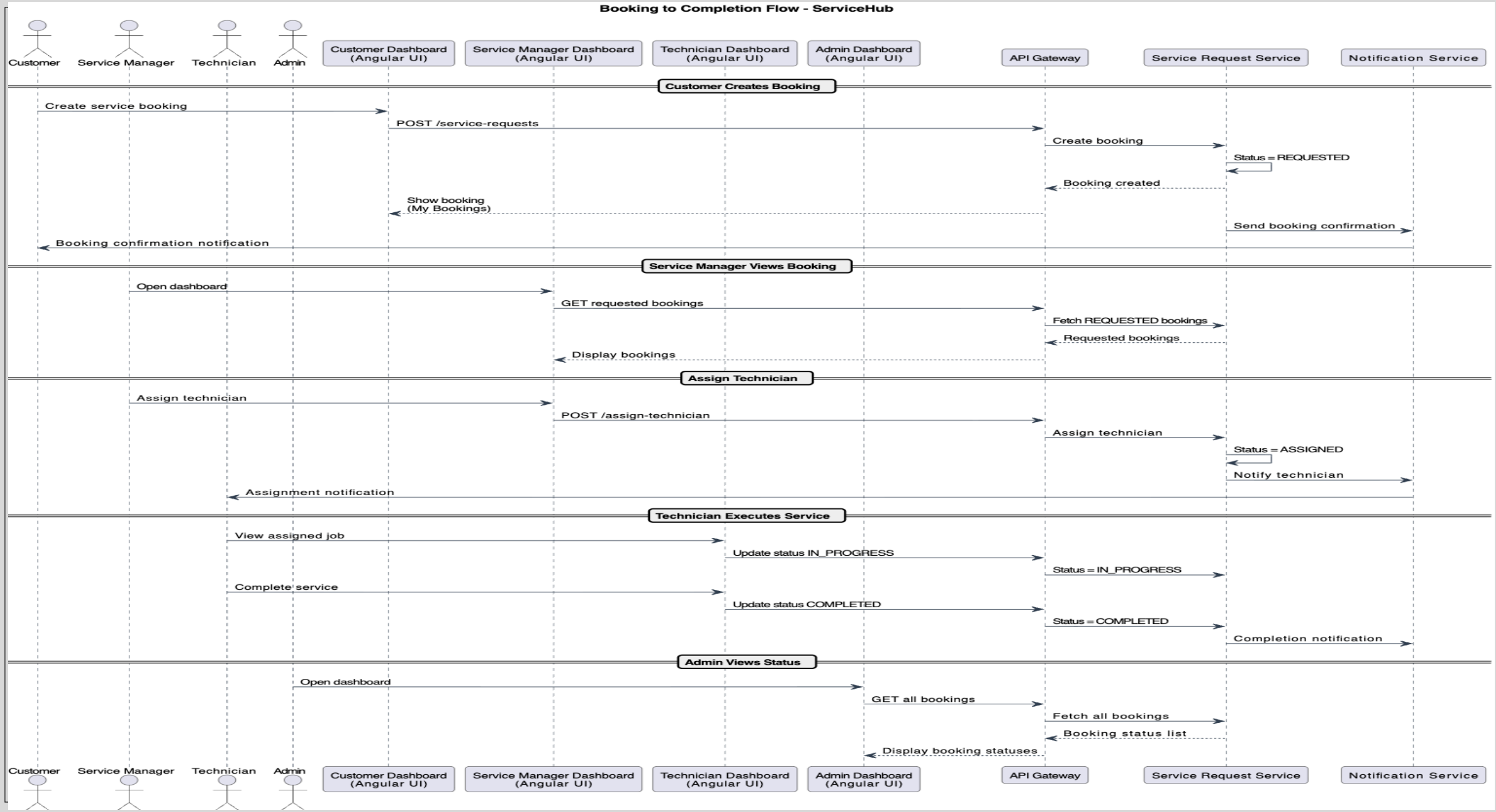

EndUser
Angular Client
API Gateway
(Security Layer)
Auth & Identity Service
Identity Database
(MySQL)
Protected Services
User Authentication
Enter credentials
POST /login
(credentials)
Route authentication request
Fetch user by email
User details
Verify password hash
(BCrypt)
Validate account status
(active / verified)
alt
[Valid Credentials]
Create JWT access token
Create refresh token
Persist refresh token
Token stored
Authentication success
(tokens)
200 OK
(accessToken, refreshToken)
Store tokens securely
[Invalid Credentials]
Authentication failed
401 Unauthorized
Secured API Access
API request
(Bearer accessToken)
Validate JWT
Forward request
EndUser
Angular Client
API Gateway
(Security Layer)
Auth & Identity Service
Identity Database
(MySQL)
Protected Services

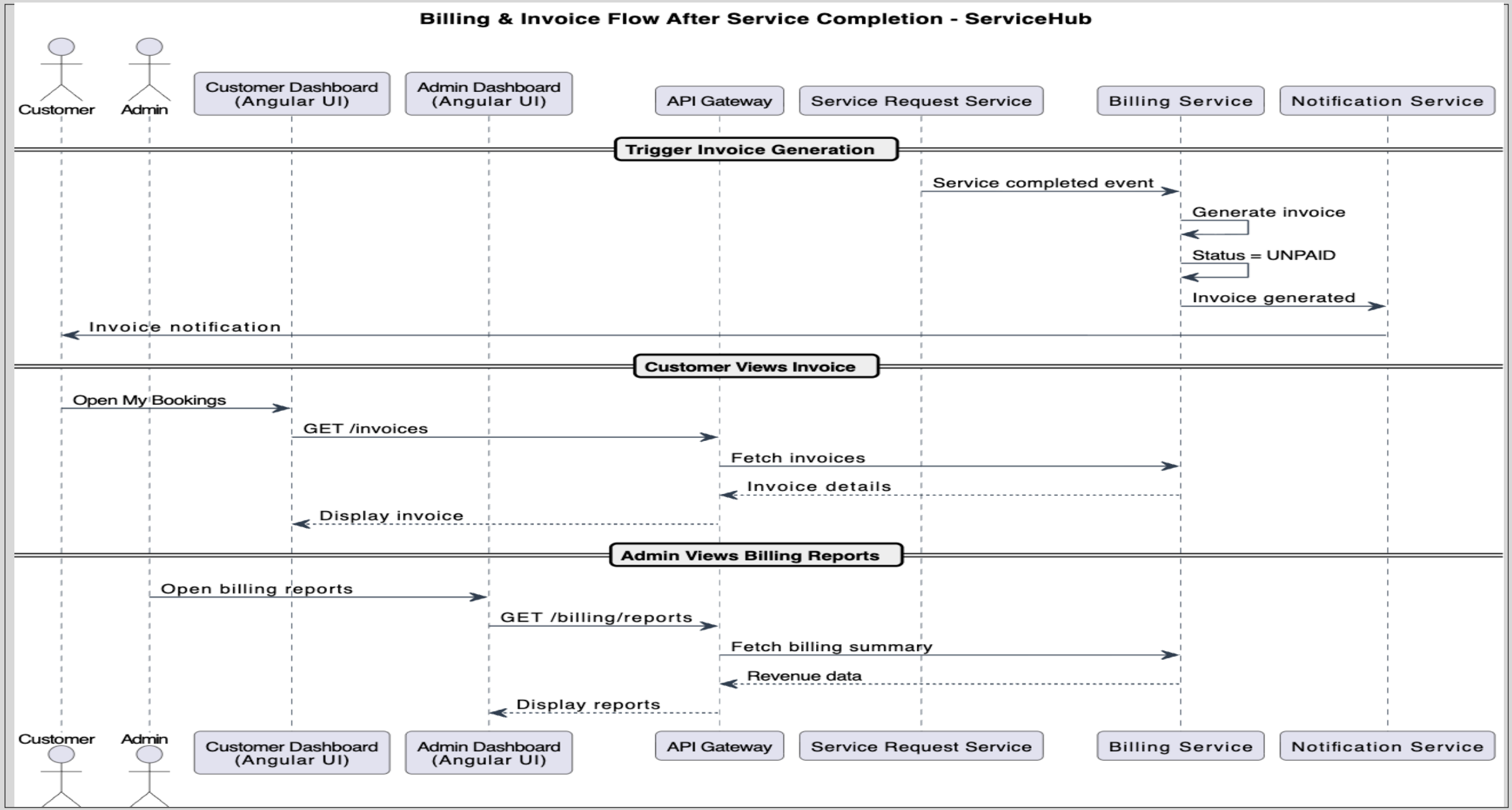

Service Manager Flow - Technician Assignment
Service Manager
Service Manager Dashboard (Angular UI)
API Gateway
Service Request Service
Technician Management Service
View Requested Bookings
Open service dashboard
GET /service-requests?status=REQUESTED
Fetch requested bookings
Requested bookings list
Display bookings
View Active Technicians
Open technician list
GET /technicians?status=ACTIVE
Fetch active technicians
Active technicians list
Display technicians
Assign Technician
Assign technician to booking
POST /assign-technician (requestId, technicianId)
Assign technician
Update technician assignment
Return to Dashboard
Assignment successful
200 OK
Redirect to dashboard (updated booking status)
Service Manager
Service Manager Dashboard (Angular UI)
API Gateway
Service Request Service
Technician Management Service


Technician
Technician Dashboard (Angular UI)
API Gateway
Service Request Service
Notification Service
Assignment Notification
Assignment notification
Open technician dashboard
View Assigned Services
GET /assigned-services
Fetch assigned services
Assigned service list
Display assigned services
Start Service
Start service
PUT /service-request/{id}/status (IN_PROGRESS)
Update status to IN_PROGRESS
Status updated
Refresh dashboard (In Progress)
Complete Service
Complete service
PUT /service-request/{id}/status (COMPLETED)
Update status to COMPLETED
Status updated
Update dashboard (Completed)
View Completed Services
View completed services
Show completed jobs list
Technician
Technician Dashboard (Angular UI)
API Gateway
Service Request Service
Notification Service

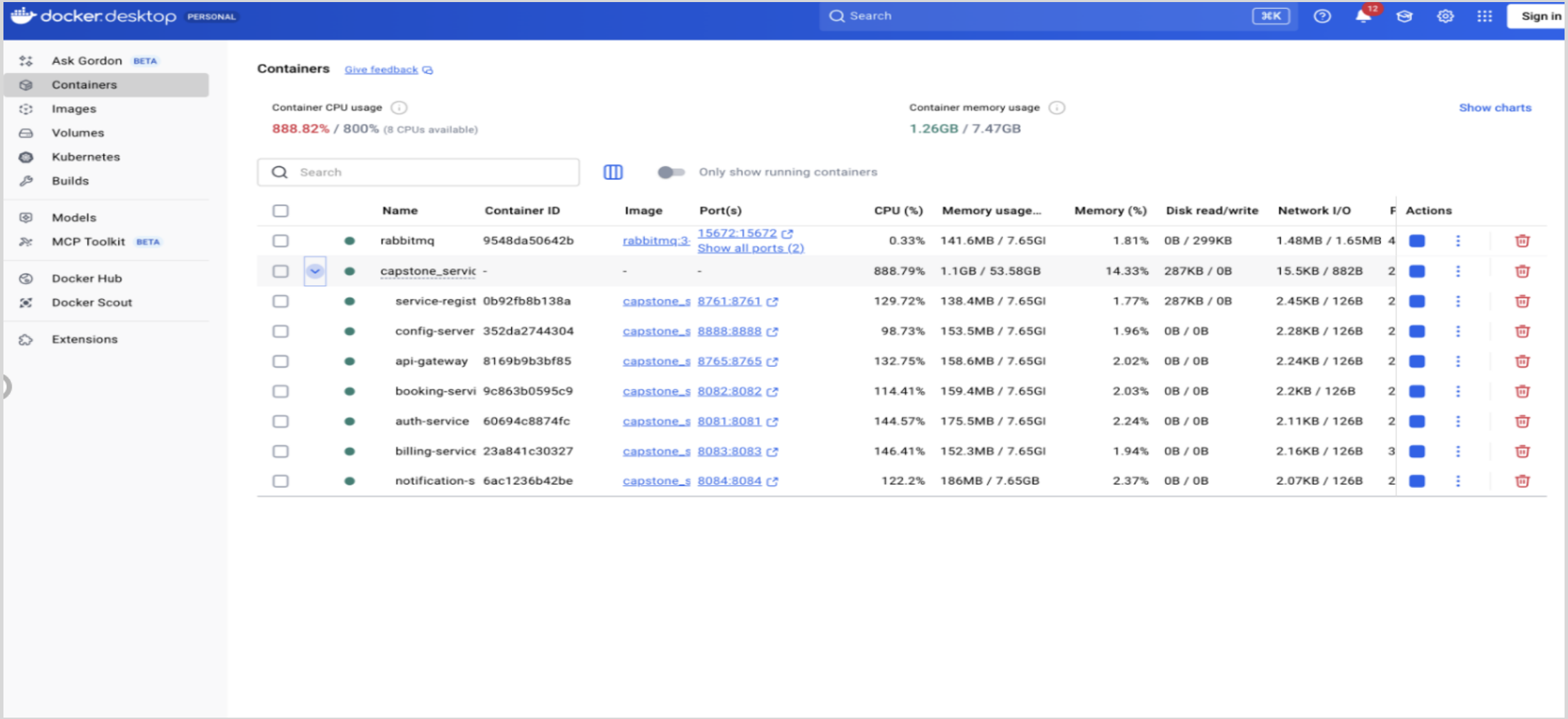

Admin Dashboard Flow - ServiceHub
Admin
Admin Dashboard (Angular UI)
API Gateway
Service Management Service
User Management Service
Technician Management Service
Billing Service
Open Admin Dashboard
Login success
Load dashboard (summary cards)
Manage Services
Click "Manage Services"
GET /services
Fetch all services
Services list
Display services
Add new service
POST /services
Create service
Service created
Refresh service list
Manage Users
Click "Manage Users"
GET /users
Fetch all users
Users list
Display users


Assign Technician
Click "Assign Technician"
GET /service-requests?status=REQUESTED
Fetch requested bookings
Requested bookings
Display bookings
GET /technicians?status=ACTIVE
Fetch active technicians
Active technicians
Display technicians
Assign technician
POST /assign-technician
Assign technician
Update assignment
Assignment updated
Success
Redirect to dashboard
View Monthly Billing
Click "Monthly Revenue"
GET /billing/monthly
Fetch monthly billing
Monthly revenue data
Display monthly billing chart
View Overall Billing
Click "Billing"
GET /billing/summary
Fetch overall billing
Billing summary
Display billing details
Admin
Admin Dashboard (Angular UI)
API Gateway
Service Management Service
User Management Service
Technician Management Service
Billing Service


Booking to Completion Flow - ServiceHub
Customer
Service Manager
Technician
Admin
Customer Dashboard (Angular UI)
Service Manager Dashboard (Angular UI)
Technician Dashboard (Angular UI)
Admin Dashboard (Angular UI)
API Gateway
Service Request Service
Notification Service
Customer Creates Booking
Create service booking
POST /service-requests
Create booking
Status = REQUESTED
Booking created
Show booking (My Bookings)
Send booking confirmation
Booking confirmation notification
Service Manager Views Booking
Open dashboard
GET requested bookings
Fetch REQUESTED bookings
Requested bookings
Display bookings
Assign Technician
Assign technician
POST /assign-technician
Assign technician
Status = ASSIGNED
Notify technician
Assignment notification
Technician Executes Service
View assigned job
Update status IN_PROGRESS
Status = IN_PROGRESS
Complete service
Update status COMPLETED
Status = COMPLETED
Completion notification
Admin Views Status
Open dashboard
GET all bookings
Fetch all bookings
Booking status list
Display booking statuses


Billing & Invoice Flow After Service Completion - ServiceHub
Customer
Admin
Customer Dashboard (Angular UI)
Admin Dashboard (Angular UI)
API Gateway
Service Request Service
Billing Service
Notification Service
Trigger Invoice Generation
Service completed event
Generate invoice
Status = UNPAID
Invoice generated
Invoice notification
Customer Views Invoice
Open My Bookings
GET /invoices
Fetch invoices
Invoice details
Display invoice
Admin Views Billing Reports
Open billing reports
GET /billing/reports
Fetch billing summary
Revenue data
Display reports
Customer
Admin
Customer Dashboard (Angular UI)
Admin Dashboard (Angular UI)
API Gateway
Service Request Service
Billing Service
Notification Service

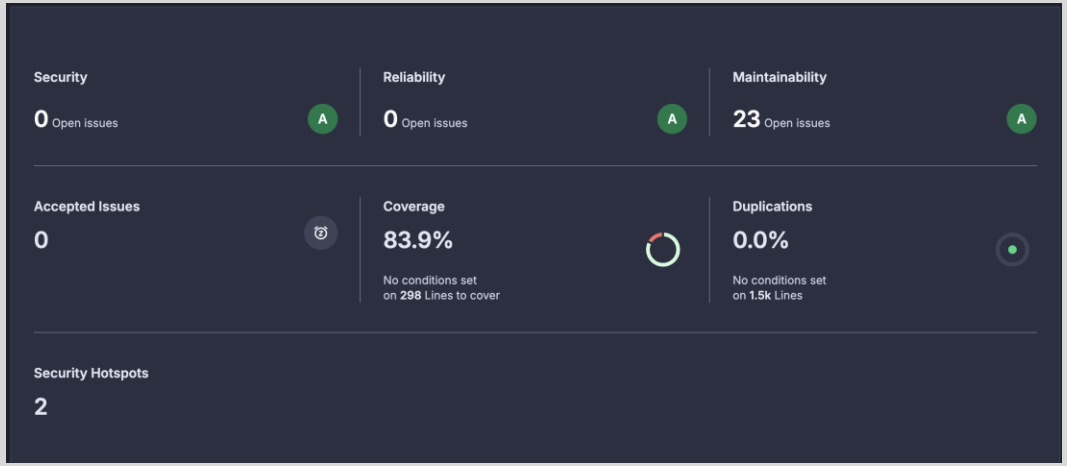
Docker Deployment

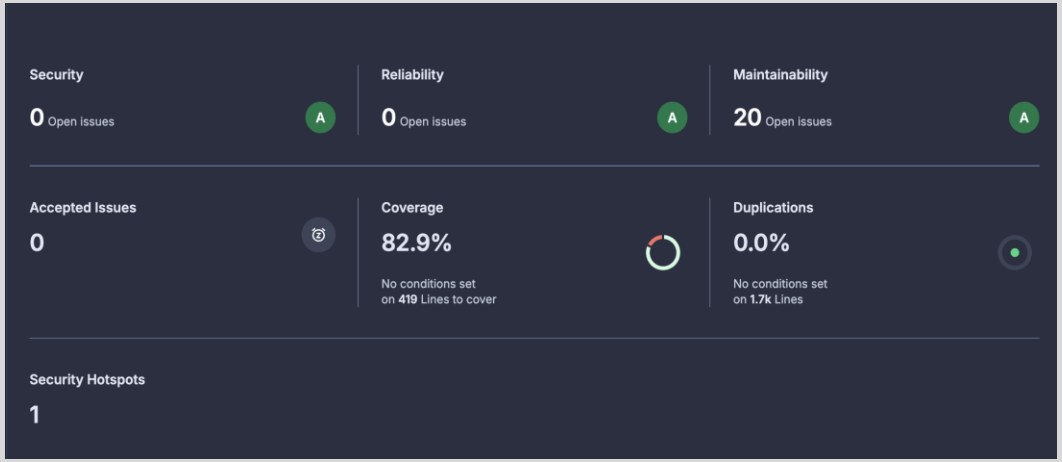


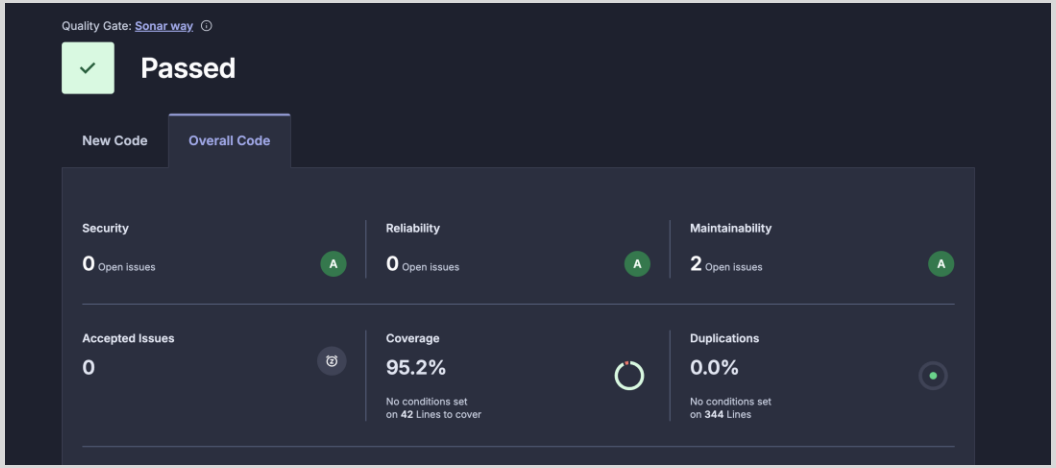
Business Rules

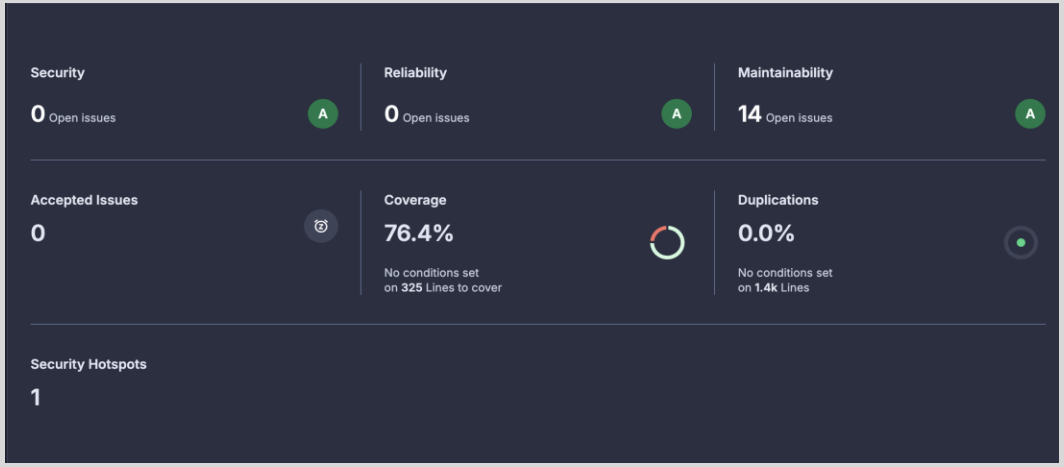
- Only registered customers can create service bookings
- Booking status flow:
  REQUESTED → ASSIGNED → IN_PROGRESS → COMPLETED
- Customers can view **only their own bookings**
- Only **Service Manager** can assign technicians
- Only **active technicians** can be assigned
- Role-based access (Customer, Manager, Technician, Admin)
- Only technicians can update service status
- Completed bookings cannot be edited

Code Quality and Testing





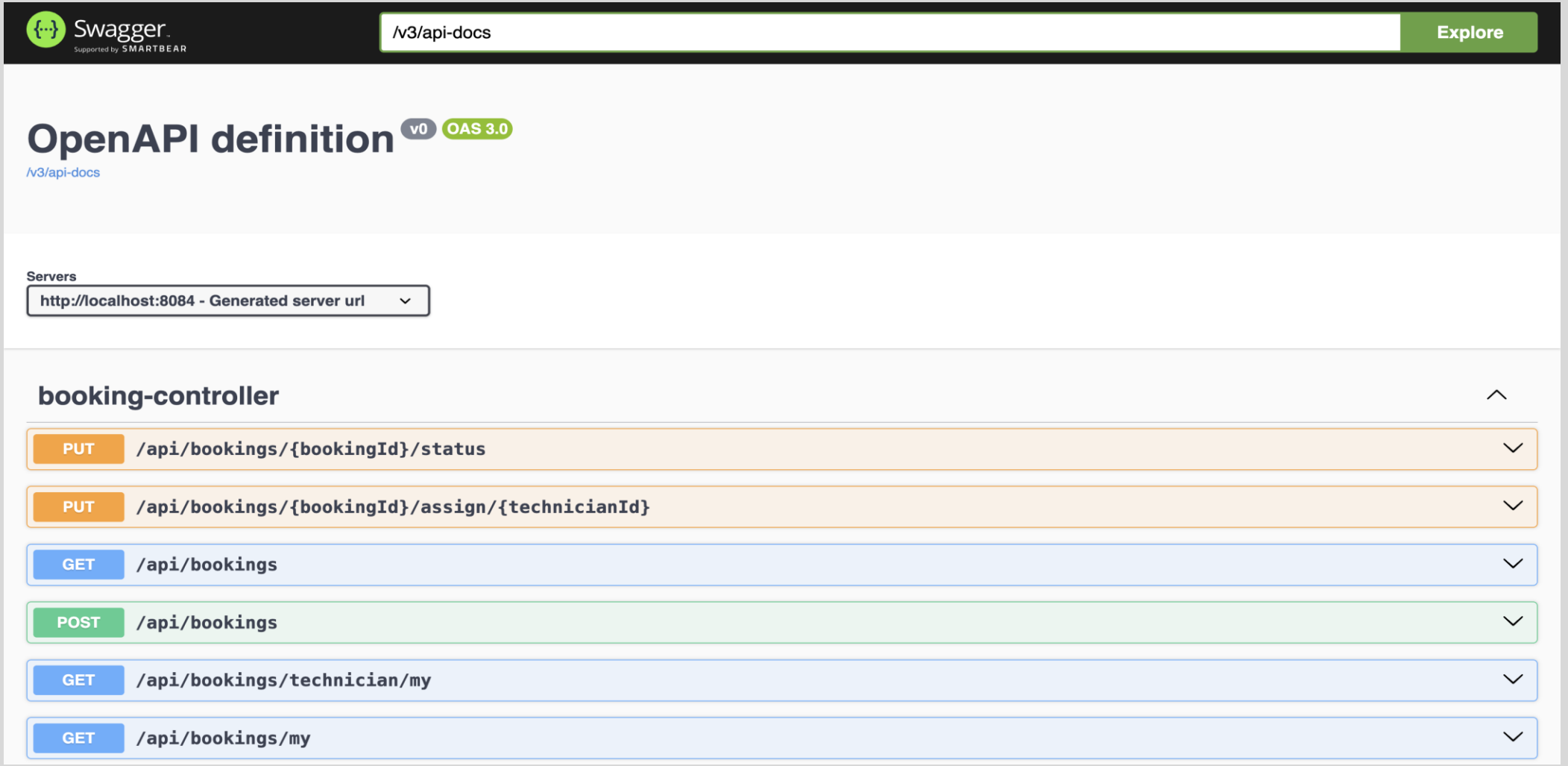




Improvements:

- **Smart Technician Assignment**
  Auto-assign technicians based on availability and workload
- **Advanced Reports & Analytics**
  Service trends, technician performance, revenue insights
- **Mobile Application**
  Android/iOS app for customers and technicians
- **Mobile Application**
  Android/iOS app for customers and technicians
- **GPS & Location Tracking**
  Display service address as a GPS point and enable route navigation for technicians

API EndPoints




Thank You!