Servista

Service Booking system

THE PROBLEM

Market Fragmentation & Mistrust

Finding Reliability

People struggle to find trustworthy service providers for household and tutoring needs.

Fragmented Market

The current market is a maze of solutions, with no single, easy-to-use platform.

No Quality-Check

It's hard to verify provider quality due to a lack of a consistent, reliable provider rating system.

Lack of Transparency

A Black Box of Services

Users face a significant lack of transparency in both **pricing** for on-demand services and difficulty in **scheduling** conveniently. This creates friction and mistrust.

User Frustration

> I just want to book a reliable plumber without visiting ten websites and wondering if I'm being overcharged.
>
> — Every Frustrated User

Our Solution

A single, transparent, and reliable platform for all on-demand services.

Centralization & Easy Booking

One Platform, Total Control

We provide a single, unified web platform to bring order to the chaos.

- **Centralize:** We bring all services, providers, and bookings into one place.
- **Easy Booking:** A simple, 3-step process: search, select a timeslot, and book.

Workflows & Oversight

For Service Providers

A clear **job visibility and assignment workflow** allows providers to manage their work efficiently, accept jobs, and track earnings.

For Administrators

We enable **administrative oversight** for monitoring all bookings, payments, and service quality via a powerful admin dashboard.

Building Trust & Security

- **Secure Transactions:** Implement secure, reliable transactions (with partners like Razorpay) to protect both customers and providers.
- **Reliable Feedback:** A robust feedback and ratings mechanism to ensure service quality, build user trust, and reward the best providers.

The User Journey

Search & Find

User searches for a service (e.g., "Tutor").

Book & Pay

User selects a provider, books a timeslot, and pays securely.

Get Service

Provider accepts the job and completes the service.

Rate & Review

User and provider rate the experience, ensuring quality.

Target Users

- Customers: Search, book, pay, and rate services.
- Service Providers: Accept assignments, manage availability, and update job status.
- Admins: Monitor operations, manage users/services, and track revenue.

System Environment

- **Application Type:** Web Application only (initially).
- **Single Responsive Interface:** A single, responsive web application will serve all three user roles (Customer, Provider, Admin).
- **Role-Based Dashboards:** The system will use Role-Based Access Control (RBAC) to display different dashboards and features to each user type after they log in.
- **Device Compatibility:** The responsive web design is intended to work seamlessly on various devices, including desktops, tablets, and mobile phone browsers.
- **Cloud Deployment:** The final application will be deployed to a live, public-facing URL.

Requirements Specification

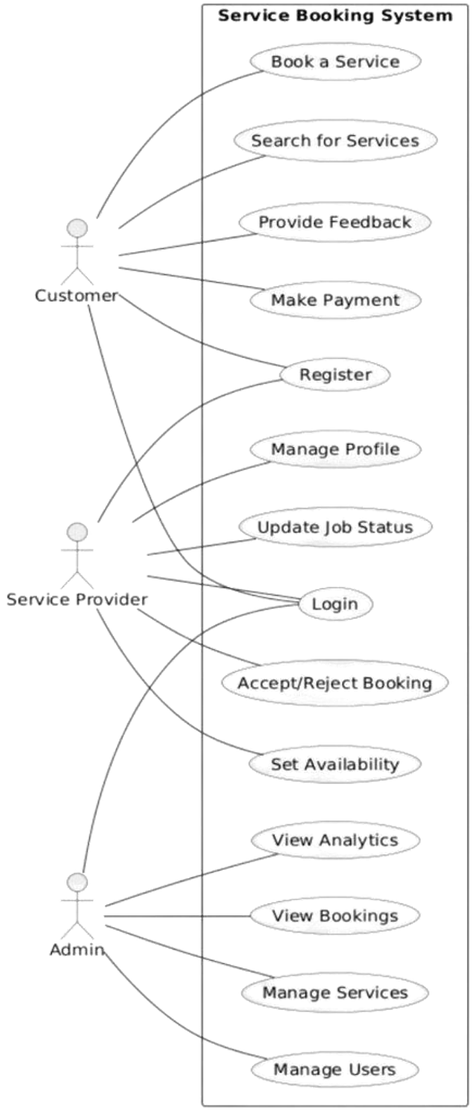
Functional Requirements

- **User Management:** 3 distinct roles (Customer, Provider, Admin) with secure registration, login, and role-based access.
- **Service Catalog:** Admin can manage services; Customers can browse and search.
- **Booking & Scheduling:** Customers book slots; Providers set availability and accept/reject bookings.
- **Payment & Billing:** Secure online payments via Razorpay and invoice generation.
- **Ratings & Feedback:** Customers can rate and review completed jobs.
- **Admin Dashboard:** Central hub to monitor bookings, payments, and user activity with basic analytics.

Non-Functional Requirements

- **Project Scope:** Focus on core booking & payment functionality in a simple, monolithic web application.
- **Key NFRs:** Nested list for Key NFRs
  - **Performance:** Fast response time (< 3-second page loads).
  - **Security:** JWT for authentication, bcrypt for password hashing, and secure payment processing.
  - **Usability:** Intuitive user interface, easy-to-navigate, and responsive UI.
  - **Availability:** High availability (target 99.9% uptime).

User Case Diagram



User Stories

Customer

- As a customer, I want to create an account and log in, so I can access the platform's services.
- As a customer, I want to search for services (by category, location) and view provider profiles, including ratings and reviews, so I can make an informed decision.
- As a customer, I want to book a service for a specific date and time, so I can schedule it at my convenience.

Service Provider

- As a service provider, I want to create a profile with my business details and the services I offer, so I can attract customers.
- As a service provider, I want to set my availability, so I only get booking requests for times I'm working.
- As a service provider, I want to receive notifications for new booking requests and be able to accept or reject them, so I can manage my workload promptly.

Admin

- As an admin, I want to manage the list of services and sub-services offered on the platform, so I can keep the catalog up-to-date.
- As an admin, I want to view and manage all users (customers and service providers), so I can maintain a healthy and safe community.
- As an admin, I want to monitor all bookings and their statuses, and view financial reports, so I can track the platform's overall activity and financial performance.

TECHNOLOGY STACK

Backend
Spring Boot

Database
PostgreSQL

Frontend
React.js

VCS & Repo
Git/Github

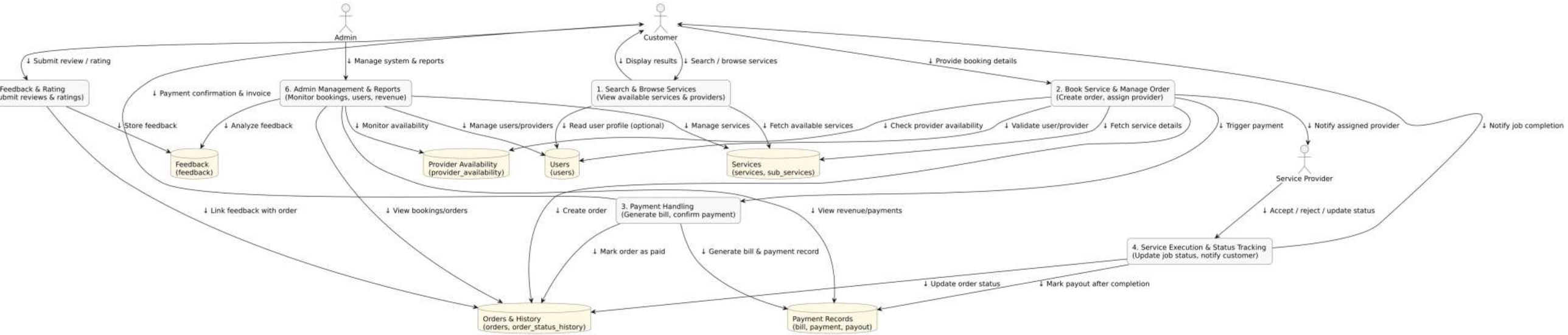
Authentication
JWT

Deployment
Vercel / Render

Payments
Razorpay

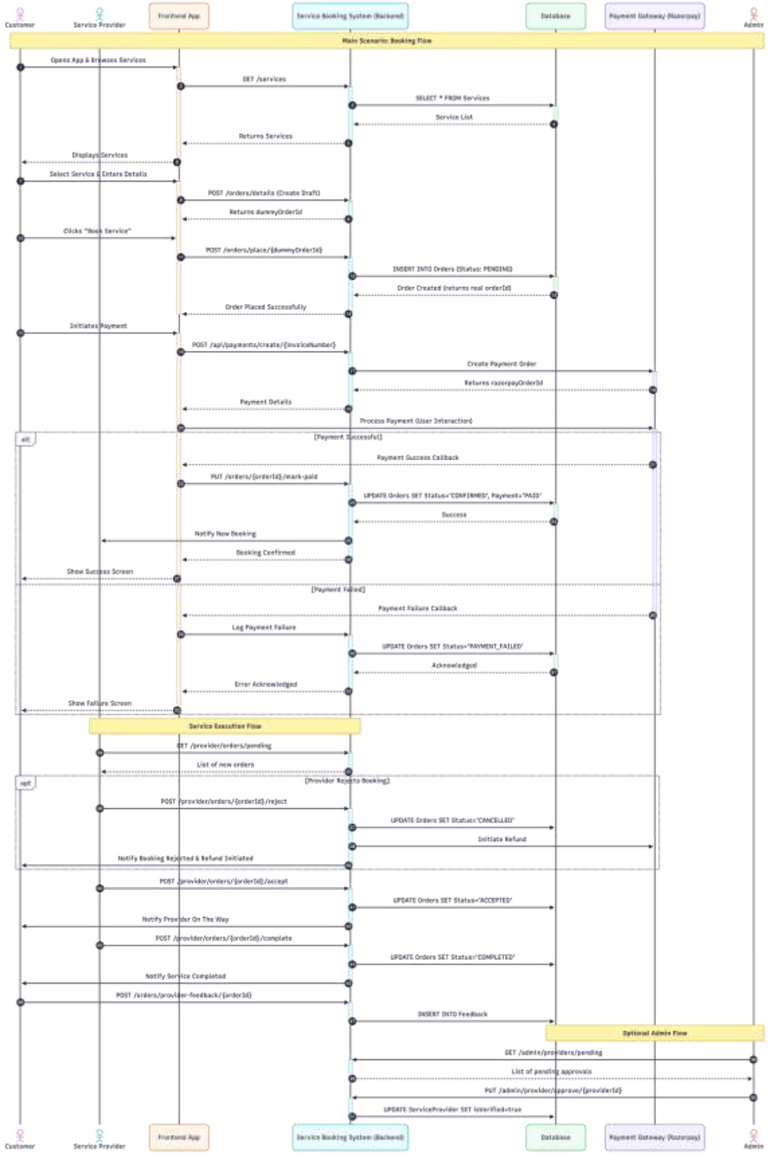
System Architecture

DFD Level 1



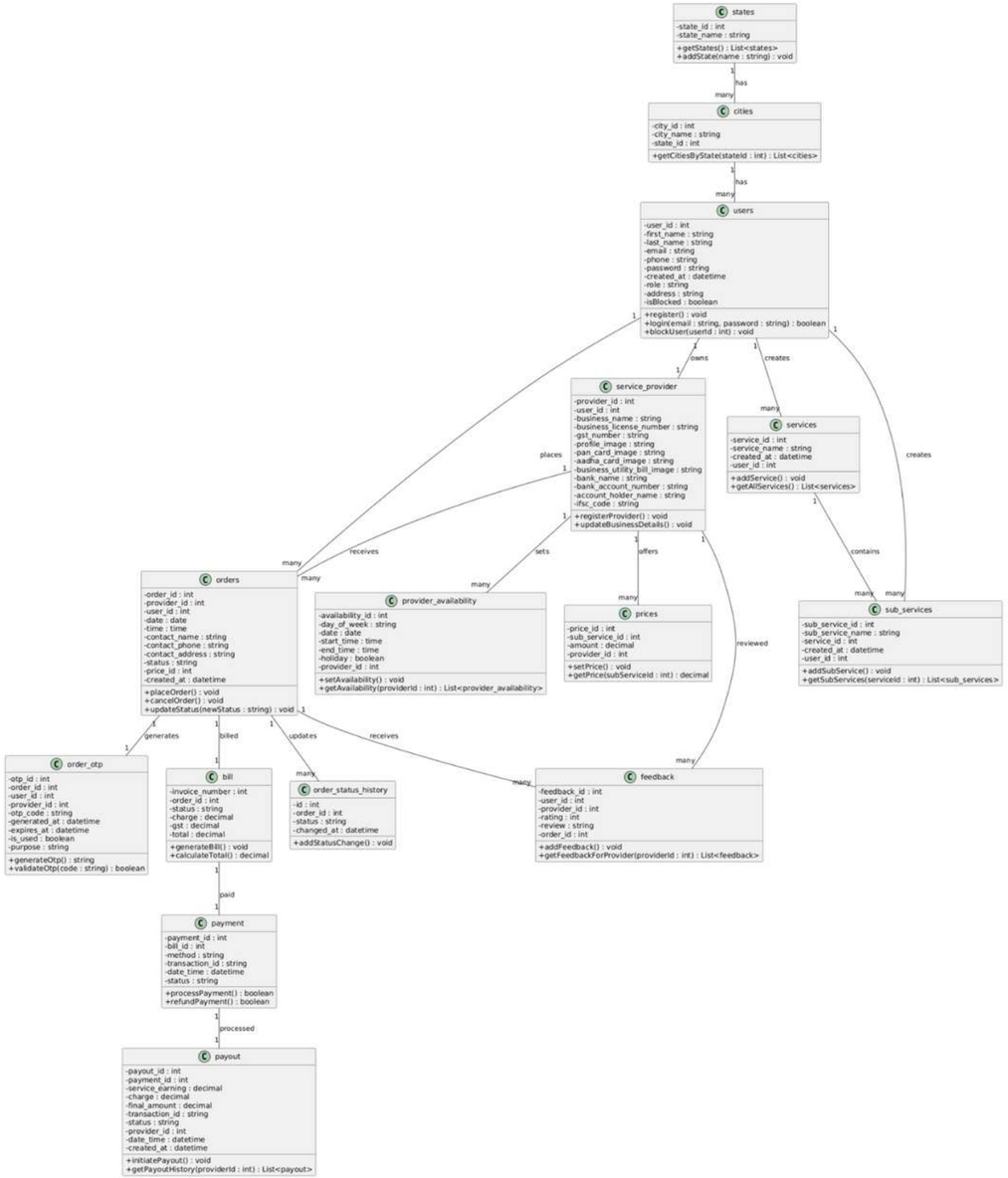
[Link]

Sequence Diagram



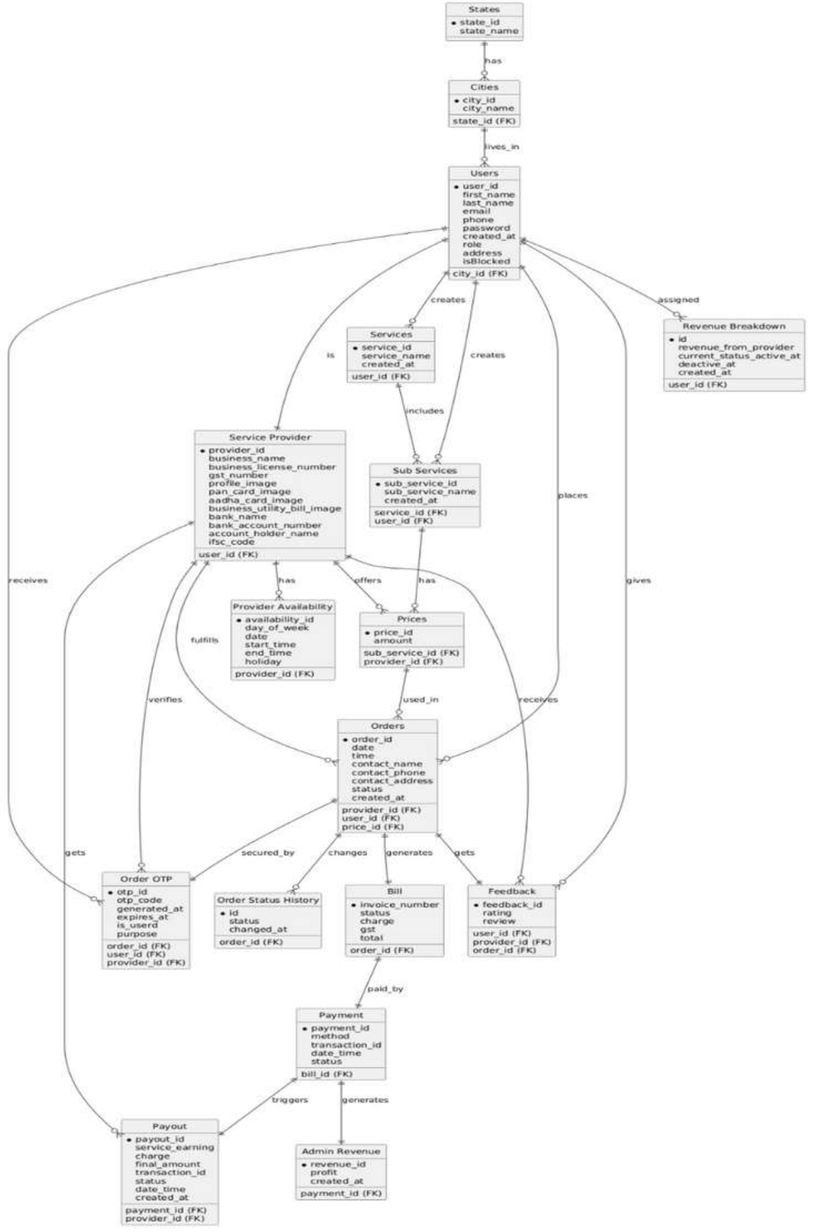
[Link]

Class Diagram



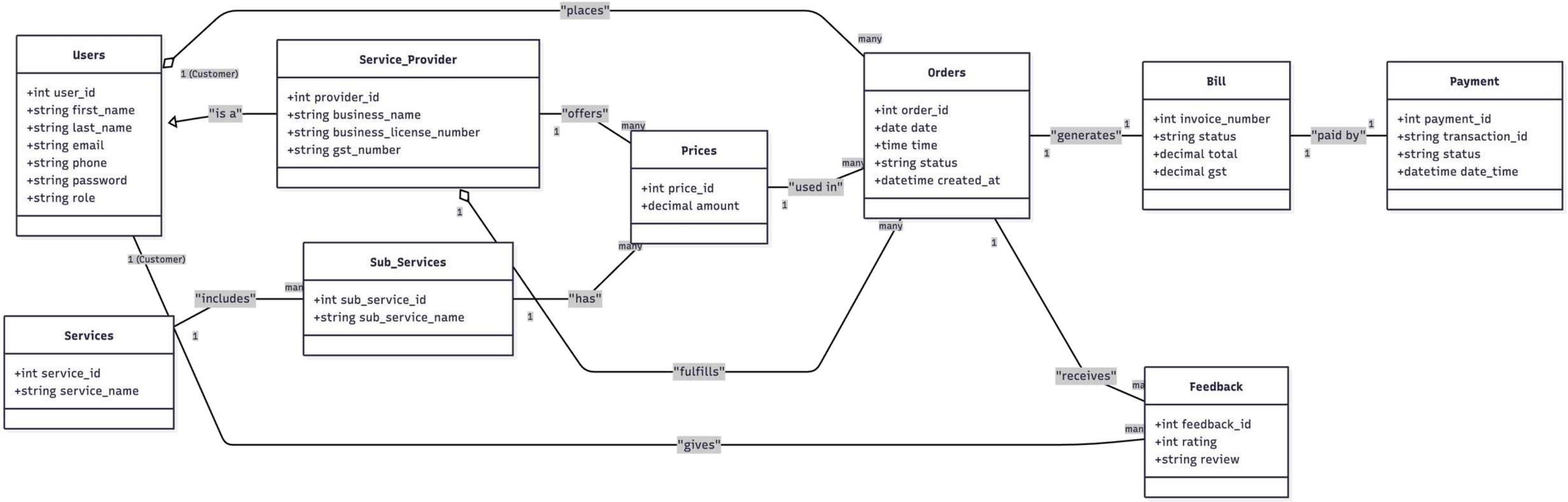
[Link]

ER Diagram



[Link]

Schema Design



Code walkthrough of key modules.

Authentication

Key Features

- Handles User Identity:
  - Manages registration, login, and password resets for all users (customers and providers).
- JWT-Based Security:
  - Generates a JSON Web Token (JWT) on successful login, which is required for accessing secure API endpoints.
- OTP Verification:
  - Implements a robust OTP (One-Time Password) system for email verification and secure password resets.

API Endpoints

- POST /register
- POST /login
- POST /change-password

Code Snippet

Payment Module

Key Features

- Payment Gateway Integration:
  - Connects the application to the Razorpay payment system to handle all financial transactions.
- Order Creation:
  - Creates a payment order with Razorpay, linking it to a specific booking ID in our system.
- Secure Webhook:
  - Provides a callback endpoint for Razorpay to confirm a successful payment.

API Endpoints

- POST /api/payments/create/{id}
- POST /api/payments/success

Code Snippet

Customer Order Module

Key Features

- Service Booking: The core module for customers to create new service bookings.
- Handles Order Data: Gathers all required details for a booking, including service ID, address, and scheduled date/time.
- View Bookings: Allows customers to retrieve their complete booking history ("My Bookings").
- View Details: Provides an endpoint to get all specific details for a single order.

API Endpoints

- POST /order/book-service
- GET /order/my-bookings
- GET /order/{orderId}

Code Snippet

Design Patterns

Builder Pattern

- This is a creational pattern used to construct complex objects.

```
RevenueBreakDown revenueBreakDown = RevenueBreakDown.builder()
        .serviceProvider(revenueSettingRequestDTO.getServiceProviderRevenue())
        .currentStatus(revenueSettingRequestDTO.getCurrentStatus())
        .activeAt(active)
        .user(user)
        .build();
```

Facade Pattern

- Our Service Layer classes act as Facades. They provide a simple, unified interface (e.g., bookService(...)) that hides the complex underlying logic of coordinating multiple repositories, checking business rules, and saving different entities

```
public String createRazorpayOrder(Bill bill) {
    try {
        //  Initialize Razorpay client
        RazorpayClient razorpay = new RazorpayClient(keyId, keySecret);

        //  Create order options JSON
        JSONObject options = new JSONObject();
        options.put("amount", (int) Math.round(bill.getTotal() * 100)); // Razorpay accepts amount in paise
        options.put("currency", "INR");
        options.put("receipt", "INV-" + bill.getInvoiceNumber());
        options.put("payment_capture", 1); // auto capture

        //  Create order on Razorpay
        Order order = razorpay.orders.create(options);

        return order.get("id");
    } catch (Exception e) {
        throw new RuntimeException(" Razorpay order creation failed: " + e.getMessage(), e);
    }
}
```

Github Repository

hitiksha-jagani Some changes in environment variables	707b777 · yesterday	81 Commits
Api spec	Revenue controller added for admin	last month
backend	Some changes in environment variables	yesterday
design	Add files via upload	2 months ago
docs	Add files via upload	last month
frontend	Updated the folder strucutre	3 months ago
reports	Add files via upload	last month
timesheets	Test commit XLSX via Apps Script	2 months ago
.env-example	Some changes in environment variables	yesterday
.gitignore	Added some initial folders	2 months ago
LICENSE	Initial commit	3 months ago
README.md	Admin profile controller's testing done	last month
docker-compose.yml	Docker setup done	2 days ago

Thank you

Servista.

Service Booking system

Testing & Quality Assurance

Objectives

- Evaluate **verification and validation** processes.
- Ensure **system robustness** under real usage.
- Define **testing strategy and coverage** across modules.
- Execute **test cases** and maintain **bug tracking**.
- Perform **performance & security testing** for reliability.

Testing Strategy

Integration Testing

Integration testing verifies how multiple modules work together within Servista system.

- API Endpoints ↔ Database (PostgreSQL):
  - Fetching Services, Creating Bookings, Updating provider status, Storing payments.
- Backend ↔ Payment Gateway:
  - Order creation, Payment signature validation
- Backend ↔ Frontend (React):
  - Booking workflow API calls, Dashboard data rendering, State managment

Unit Testing

Unit testing focuses on validating individual components of the Servista backend and frontend to ensure each function behaves as expected before integration.

- **Server Level Logic:**
  - Booking Creation Logic, Service Filtering and Search
- **Controller Level Validation:**
  - Api Input Validation, Error Handling(400,401,404,500)
- **Authentication Module:**
  - JWT Token generation & Verification, Password Hashing with bcrypt, Role-based checks.
- **Utility Functions:**
  - Date and Time slot validation, price calculation.

System Testing

System testing validates the **entire Servista application** in a real-life scenario from start to end.

A) Customer Workflow

Login / Signup , Search for a service, View service details, Book a time slot, Proceed to payment.

B) Provider Workflow

Login as provider, View assigned bookings, Accept / Reject jobs, Update job status

C) Admin Workflow

Login as admin, View total bookings, revenue, charts ,Manage providers & customers, Resolve disputes / edit bookings, Monitor KPIs and analytics

- System Wide Test:
  - Responsiveness across devices, Session handling & token expiry, , Error page rendering

Test Cases

- Login & Signup Authentication
- Service search & filtering
- Booking creation & provider assignment
- Job status updates
- Payment completion
- Admin KPIs & analytics

Test Case PDF : Github Link

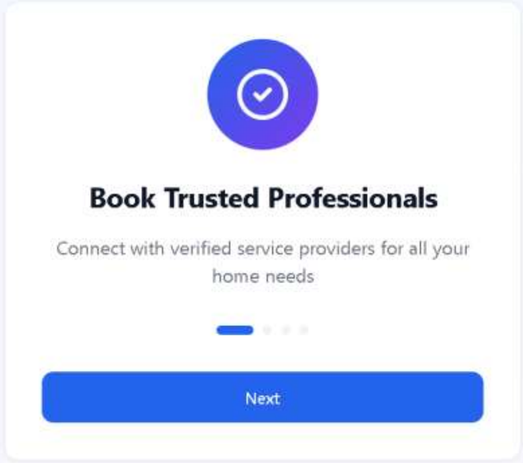
Thank you

Servista.

Service Booking system

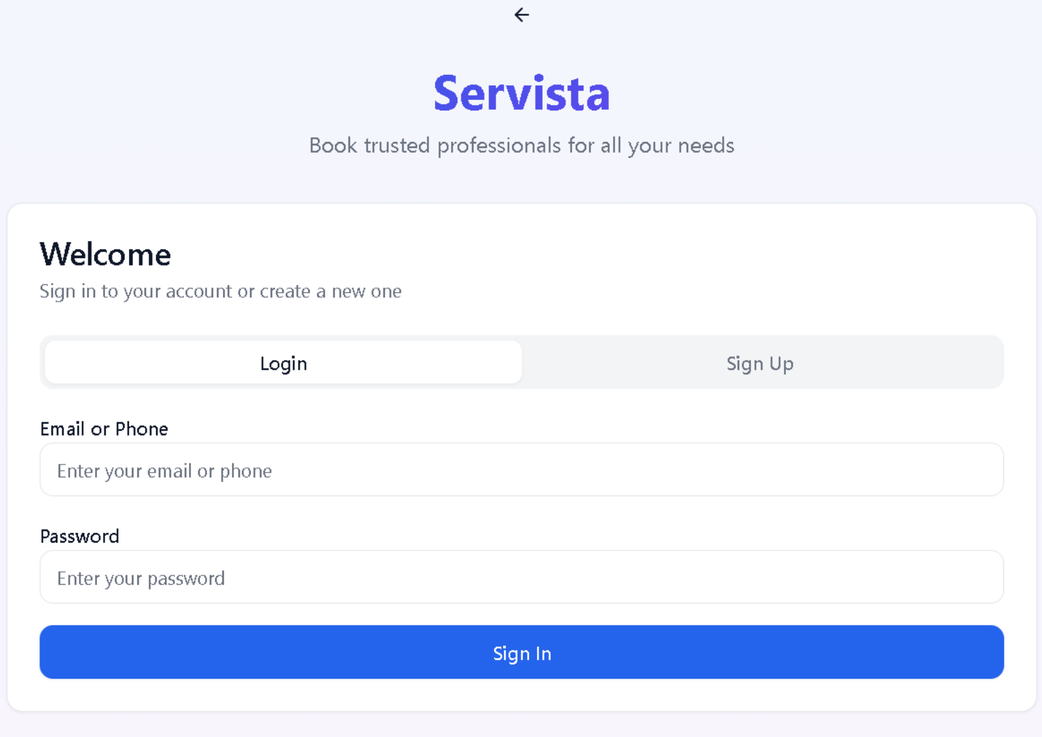
User manual.

Skip



Onboarding

Login



Home

Servista

Home | My Bookings | Profile

Quality Home Services At Your Doorstep

Book trusted professionals for all your home service needs.

Search for services...

Login successful!

Service Details

Leak Repair

3.6 Average rating

★★★½☆ 5 reviews

Verified reviews

Fixing pipe leaks, sealing joints, and resolving water leakage issues.

Select a Package

Leak Repair
Fixing pipe leaks, sealing joints, and resolving water leakage issues.

₹499

Customer Reviews

MP **Malhar Prajapati** 11/30/2025 — 3★ ★★★☆☆

MP **Malhar Prajapati** 11/30/2025 — 1★ ★☆☆☆☆

Book Now

Service

Servista

Home My Bookings Profile

My Bookings

Manage your service bookings

Upcoming (2) | In Progress (1) | Completed (3) | Cancelled (5)

3 December 2025
21:07
Rajkot

Booking ID: 802 ₹499

11 December 2025
10:39
Mumbai

Booking ID: 1002

Bookings

Booking details

Booking Details

Service

UPCOMING

Package

Booking ID: 1002

Date
11 December 2025

Time
10:39

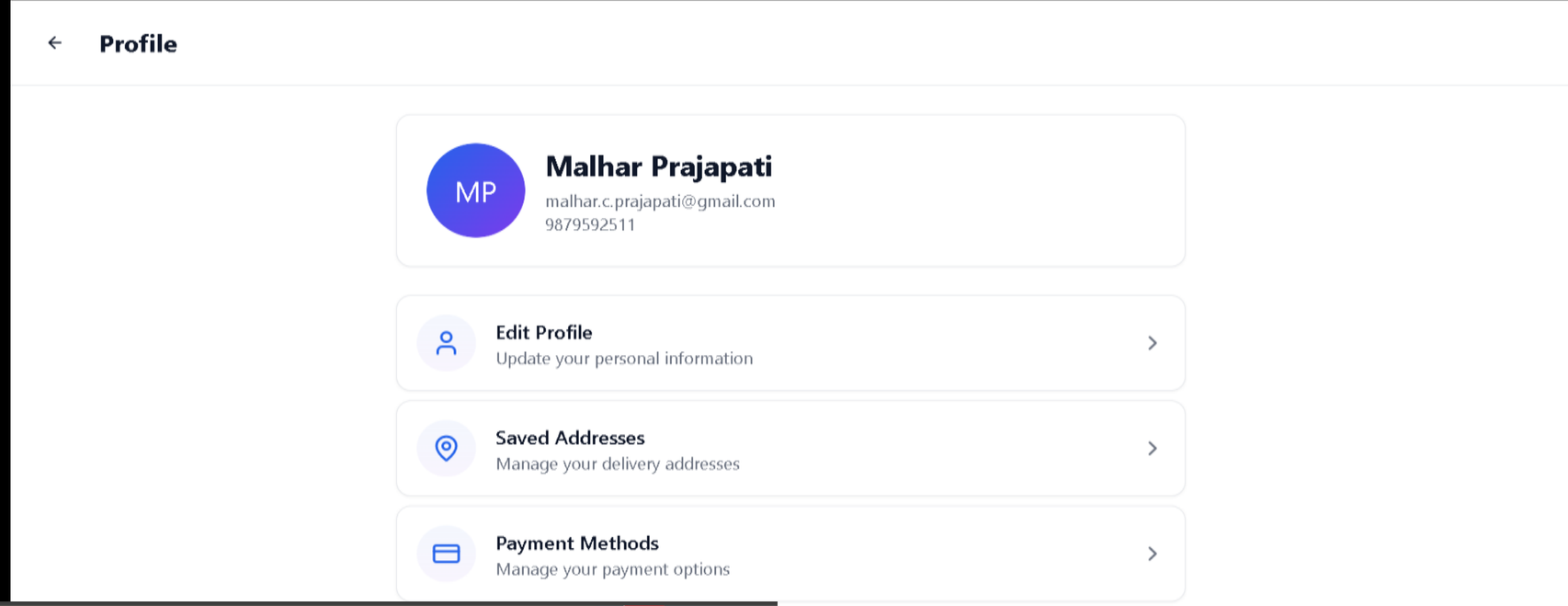
Address
Mumbai

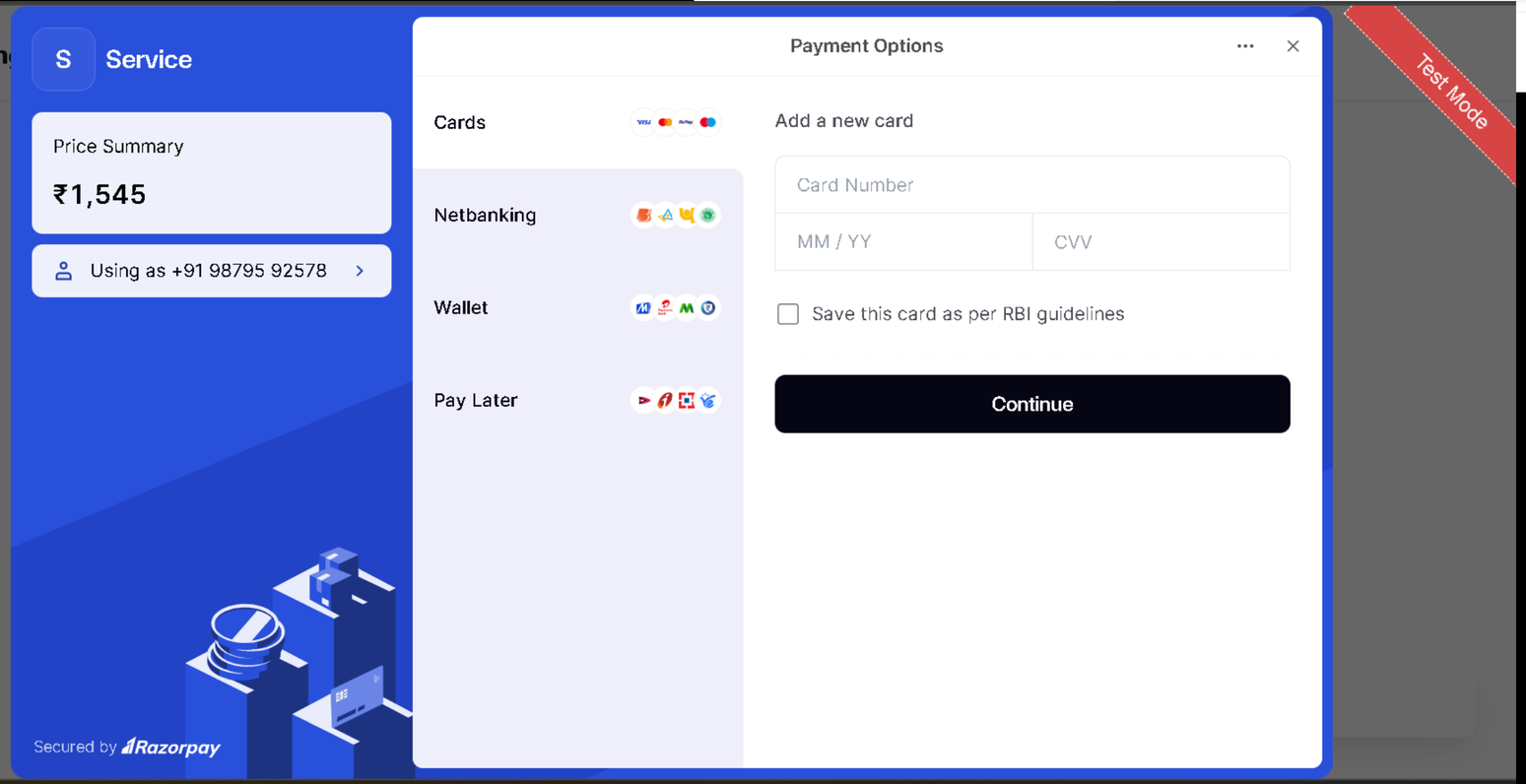
Payment Summary

Invoice 1002

Service Charge ₹1,500.00

Profile





Payment Gateway

Thank You