



UE22CS352B - Object Oriented Analysis & Design

Mini Project Report

“Urban-Connect: A One-Stop Service Booking and Management Solution”

Submitted by:

SAI MOURYA N DODDAMANI	PES1UG22CS502
SANATH SHETTY	PES1UG22CS520
SHASHANK R PATIL	PES1UG22CS550
VIDHAAN VISWAS	PES1UG23CS835

Semester 6th | Section I

Under the guidance of

Mahitha G

Assistant Professor

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India.

Problem Statement

In today's fast-paced urban lifestyle, accessing reliable home services ranging from plumbing and electrical repairs to housekeeping and appliance maintenance often becomes a tedious and inefficient process. Customers struggle to find trusted professionals, while independent service providers face challenges in reaching a wider client base due to the lack of a unified platform.

Urban-Connect aims to bridge this gap by providing a **Software-as-a-Service (SaaS)** web-based platform that facilitates seamless interaction between customers and service providers. It streamlines service discovery, booking, communication, and dispute resolution within a single application. The system ensures security and quality through an admin approval mechanism before allowing providers to operate, thus promoting trust on both sides.

This solution leverages modern web technologies to deliver an intuitive, scalable, and reliable service booking and management system that addresses the evolving demands of urban service ecosystems.

❖ Key Features

- **For Customers**

- ✓ **Secure Signup/Login:** Role-based access with secure authentication.
- ✓ **Personalized Profile:** Each customer has a unique profile for managing bookings and preferences.
- ✓ **Browse Services:** Explore a wide range of home services with filters and search functionality.
- ✓ **Book Services:** Real-time booking with date/time preferences and location details.
- ✓ **In-App Chat:** Communicate with assigned service providers directly within the app.
- ✓ **Booking History:** View past and upcoming bookings with detailed status updates.
- ✓ **Dispute Resolution & Help Desk:** Raise concerns or disputes and access help resources.

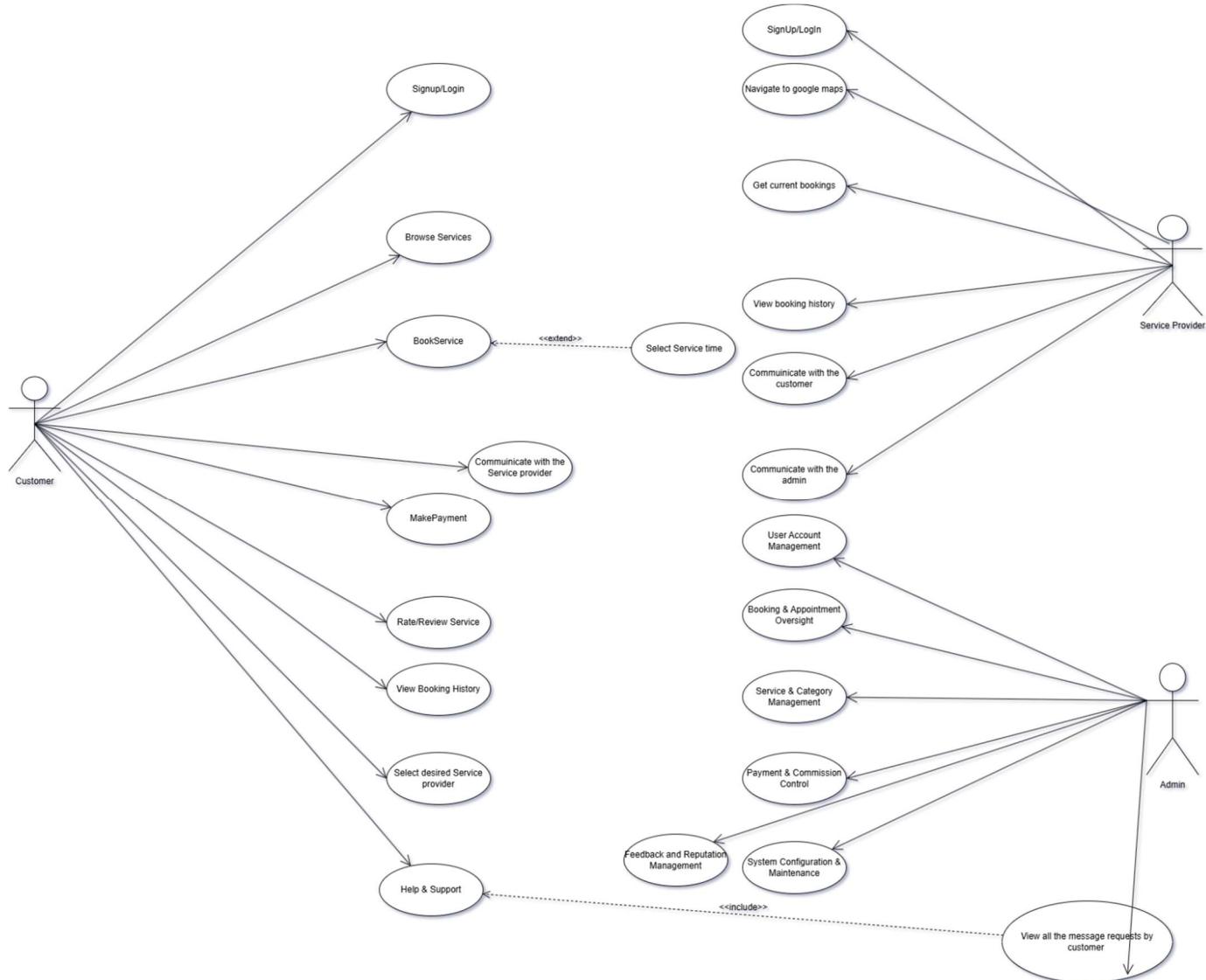
- **For Service Providers**

- ✓ **Signup/Login with Verification:** Onboarding begins with registration, followed by admin approval.
- ✓ **Booking Dashboard:** View service requests with relevant customer details.
- ✓ **Location Access:** Receive accurate customer location for service delivery.
- ✓ **Customer Communication:** Integrated messaging to discuss details with customers.
- ✓ **Service History:** Track past services rendered, ratings, and earnings.

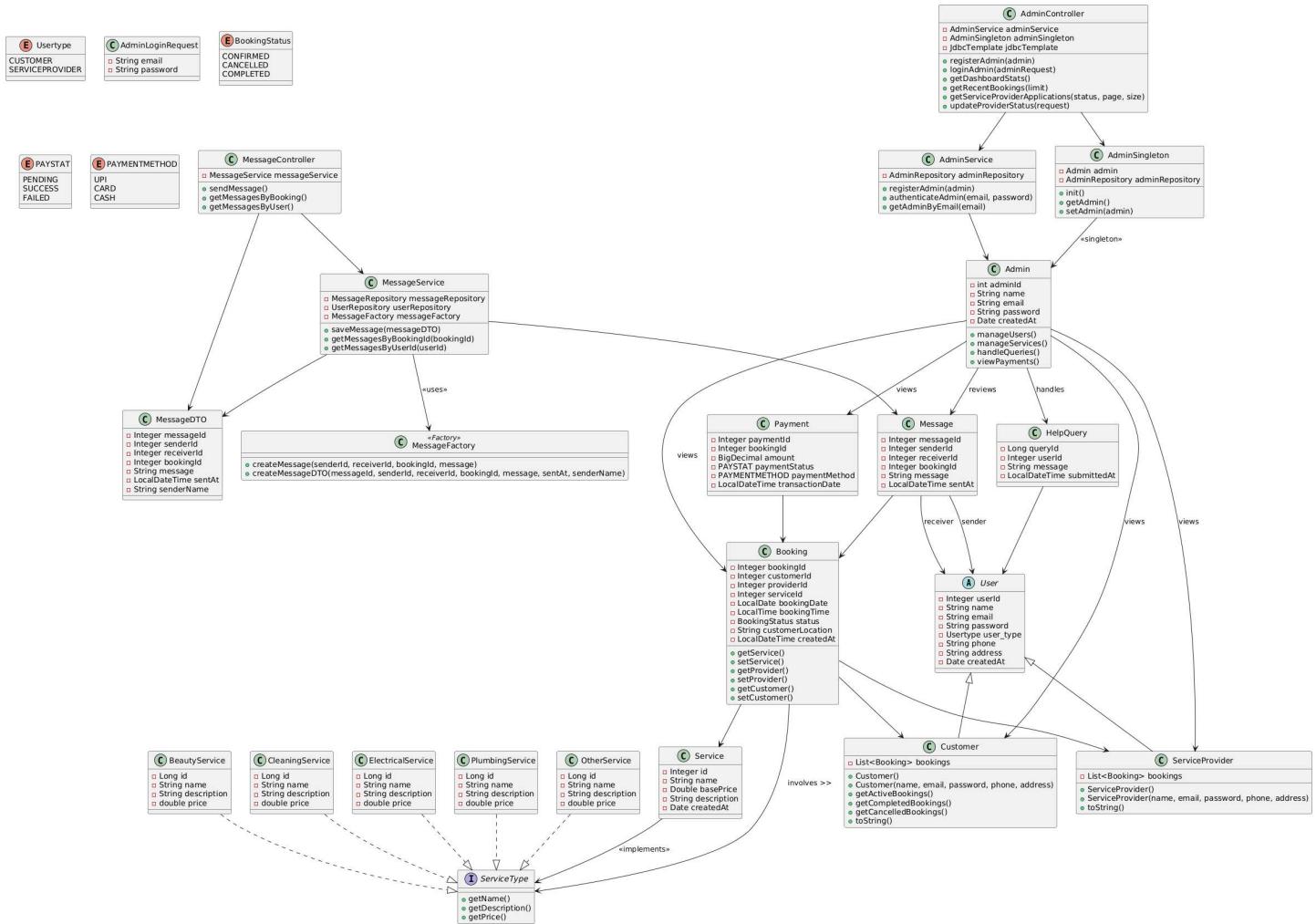
- **For Admin**
 - ✓ **Complete Platform Control:** Monitor and manage users, services, bookings, and disputes.
 - ✓ **Provider Approval Workflow:** Review service provider registrations and grant access after validation.
 - ✓ **Data Analytics Dashboard (Optional Enhancement):** Visual insights into platform usage, peak service times, and satisfaction trends.
 - ✓ **Dispute Management:** Oversee dispute resolutions and ensure quality control.

❖ Models

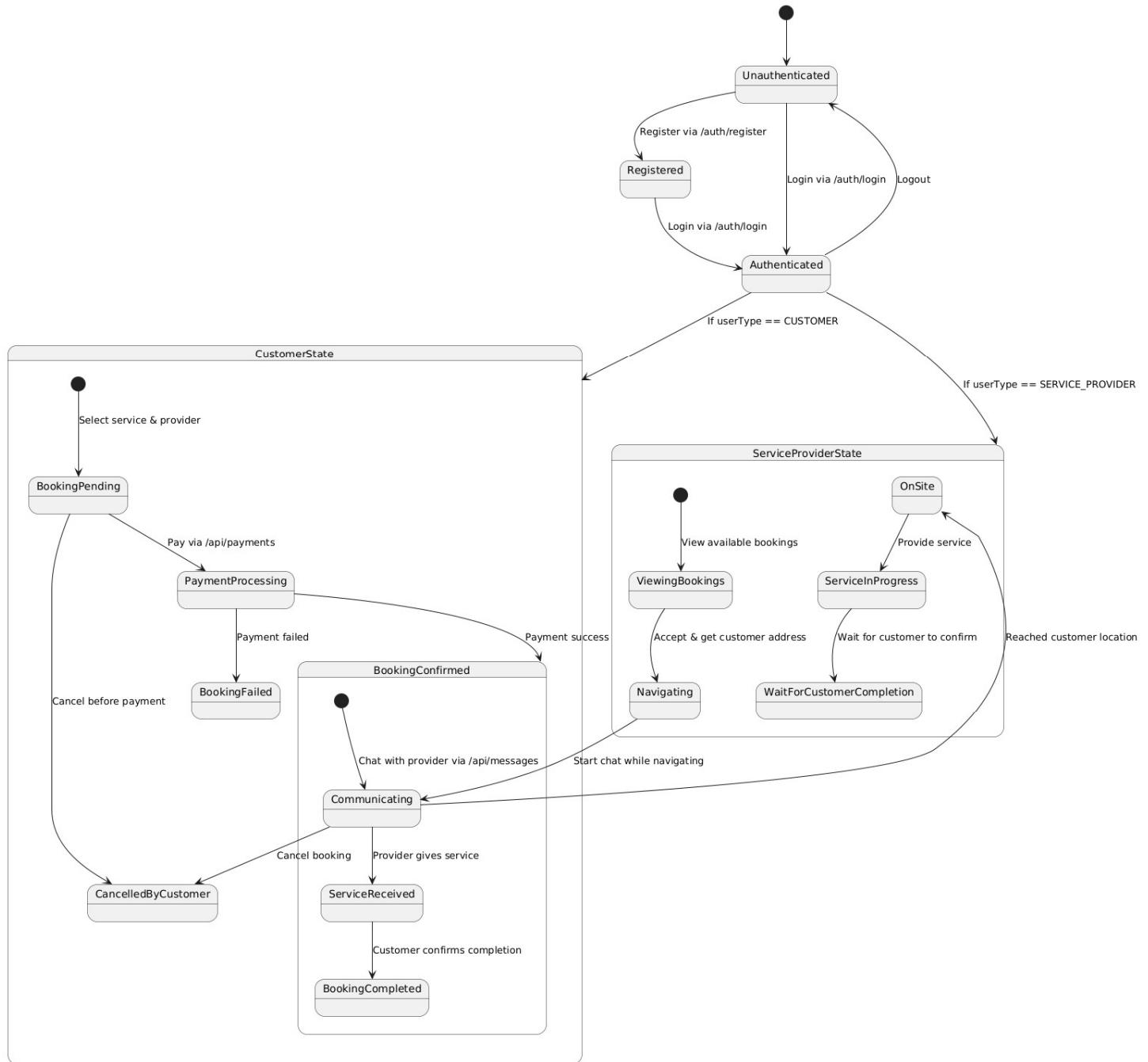
- Use Case Diagram



- **Class Diagram**



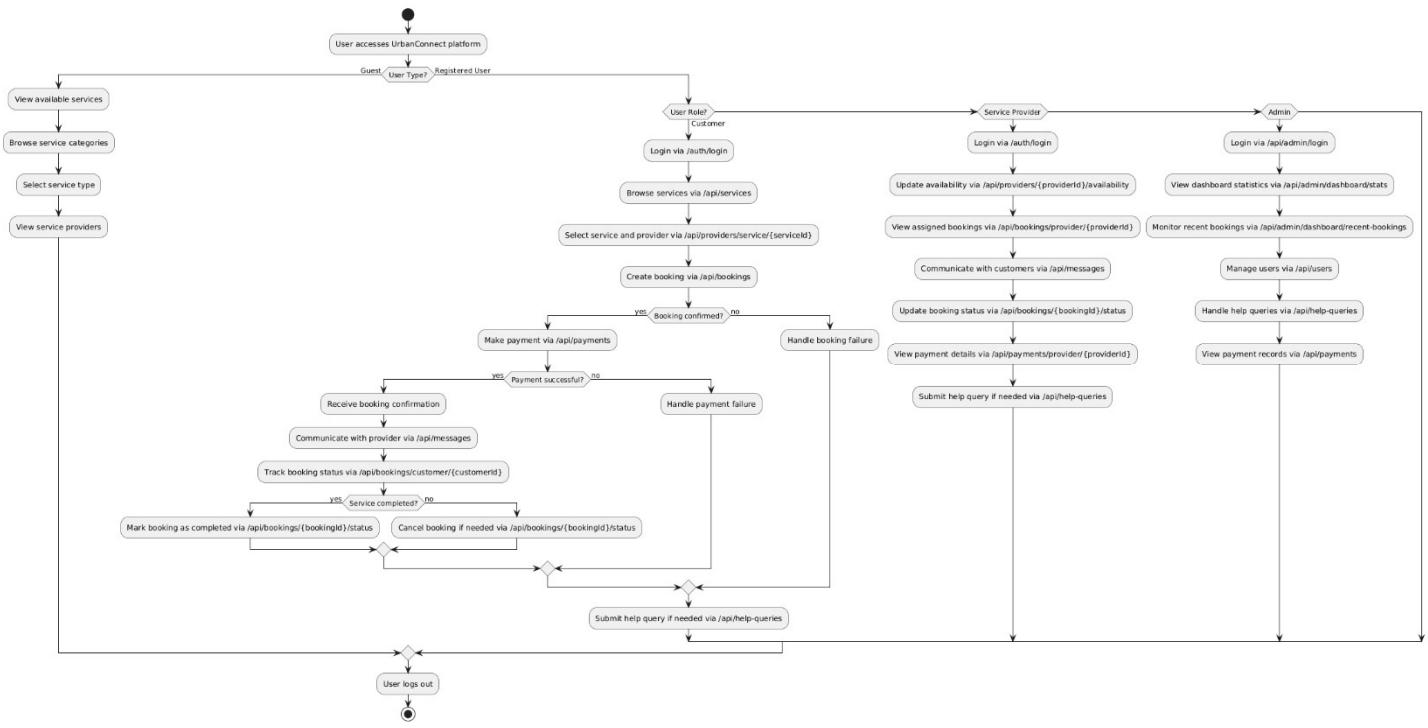
- State Diagram



- **Activity Diagrams**

1. Major Use-case

We have one complete activity diagram which covers almost all the use cases (we have combined those 4 diagrams)



❖ Architecture Patterns

- **Model - View - Controller Pattern (MVC)**

- **Layered Architectural pattern**

- 1. Presentation Layer (Controller)**

- a. Handles HTTP requests/responses.
 - b. Example: Admin-Controller, Message-Controller
 - c. Communicates with the service layer.

- 2. Business Logic Layer (Service)**

- a. Contains business rules and application logic.
 - b. Example: Admin-Service, Message-Service
 - c. Acts as a bridge between Controller and Repository.

- 3. Data Access Layer (Repository)**

- a. Handles CRUD operations with the database.
 - b. Example: Admin-Repository, User-Repository
 - c. Usually uses JPA or raw SQL (as in your case).

❖ Design Principles

1. S - Single Responsibility Principle (SRP):

- Each class in the diagram has a single, well-defined responsibility.
 - Message-Service handles message-related logic.
 - Admin-Controller deals only with admin operations.
 - Booking class handles booking data and related actions.
- This ensures code modularity and simplifies future changes.

2. O - Open/Closed Principle (OCP):

- The Service Type interface allows the system to be extended with new service types (like Beauty Service, Plumbing Service, etc.) without modifying existing code.
- This supports future scalability and flexibility.

3. I - Interface Segregation Principle (ISP):

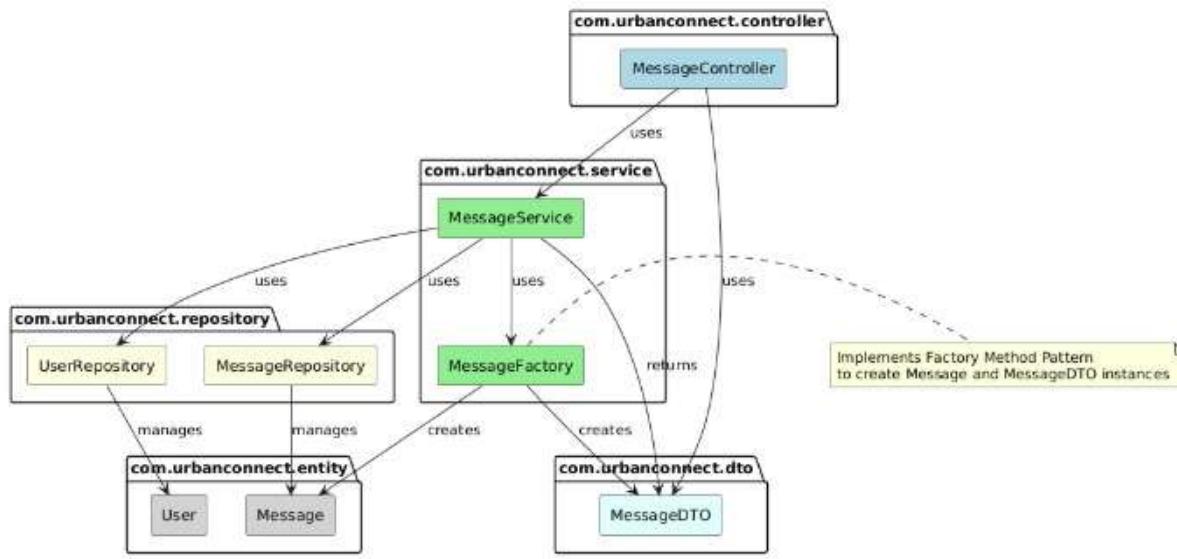
- Interfaces are designed with specific roles.
- The Service Type interface defines only service-related behavior, keeping it minimal and focused.

4. D - Dependency Inversion Principle (DIP):

- High-level classes like Message Controller and Admin Controller depend on abstractions (Message Service, Admin Service) rather than concrete implementations.
- This promotes loose coupling and makes unit testing easier.

❖ Design Patterns

We have used two patterns Singleton pattern and factory pattern.



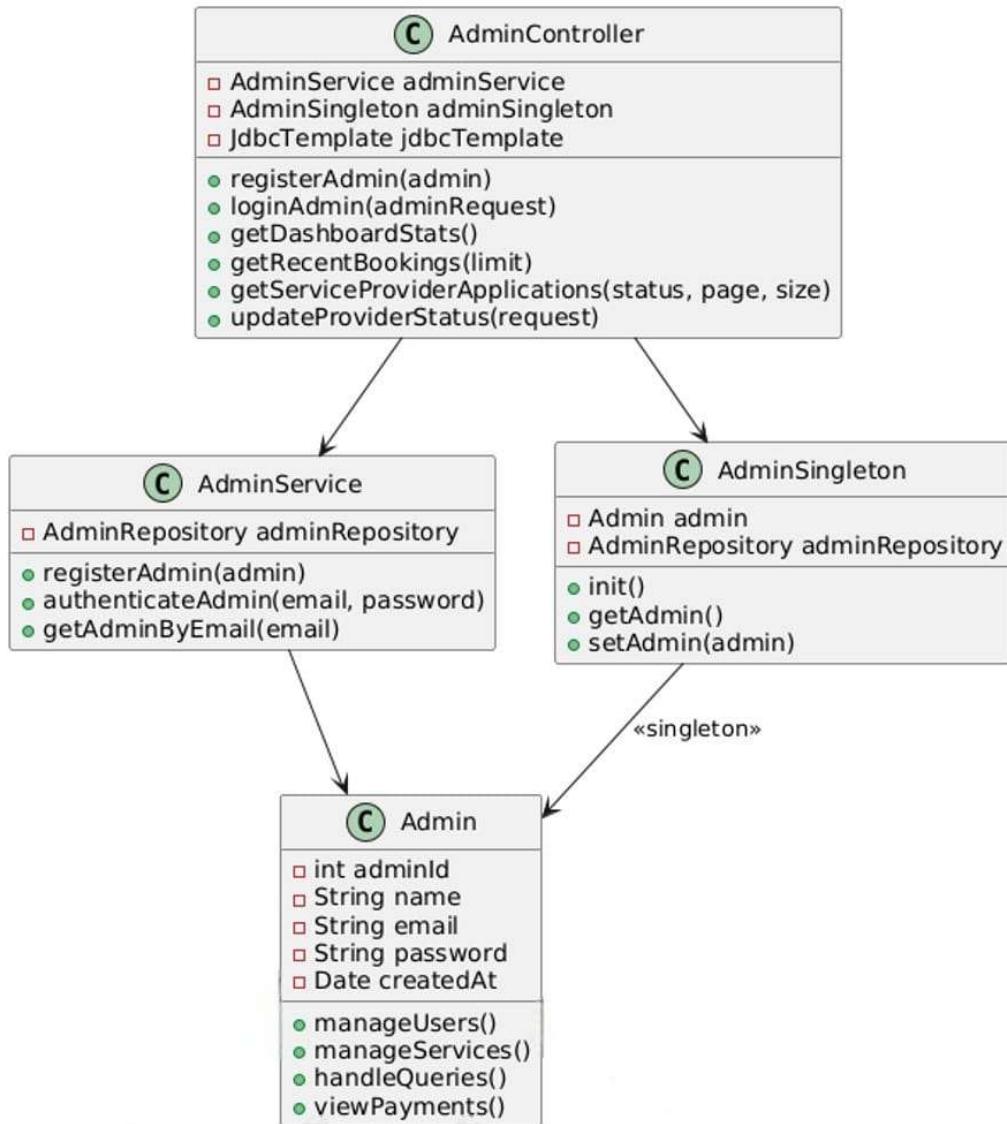
1. Structure and Components

- ✓ **Message-Factory:** This is the factory class, annotated with `@Component` to make it a Spring-managed bean. It contains two key methods:
 - `createMessage(senderId, receiverId, bookingId, message):` Creates a `Message` entity with the provided parameters and sets the `sentAt` timestamp to `LocalDateTime.now()`.
 - `createMessageDTO(messageId, senderId, receiverId, bookingId, message, sentAt, senderName):` Creates a `MessageDTO` object, populating it with the given fields, including the sender's name for display purposes.
- ✓ **MessageService:** This is the client of the factory. It uses `@Autowired` to inject `MessageFactory` and relies on its methods to create objects instead of directly using `new Message()` or `new MessageDTO()`.
- ✓ **MessageController:** Remains unchanged, interacting with `MessageService` as before, unaware of the factory's role.

2. Workflow

- ✓ **Saving a Message (saveMessage):** When `MessageController` receives a `POST /api/messages` request with a `MessageDTO`, it passes it to `MessageService.saveMessage`.

- ✓ Inside saveMessage, the service calls messageFactory.createMessage to instantiate a Message object with the DTO's data.
- ✓ The Message is saved to the database via messageRepository.save.
- ✓ After retrieval, messageFactory.createMessageDTO is used to create a response MessageDTO, incorporating the sender's name from UserRepository.



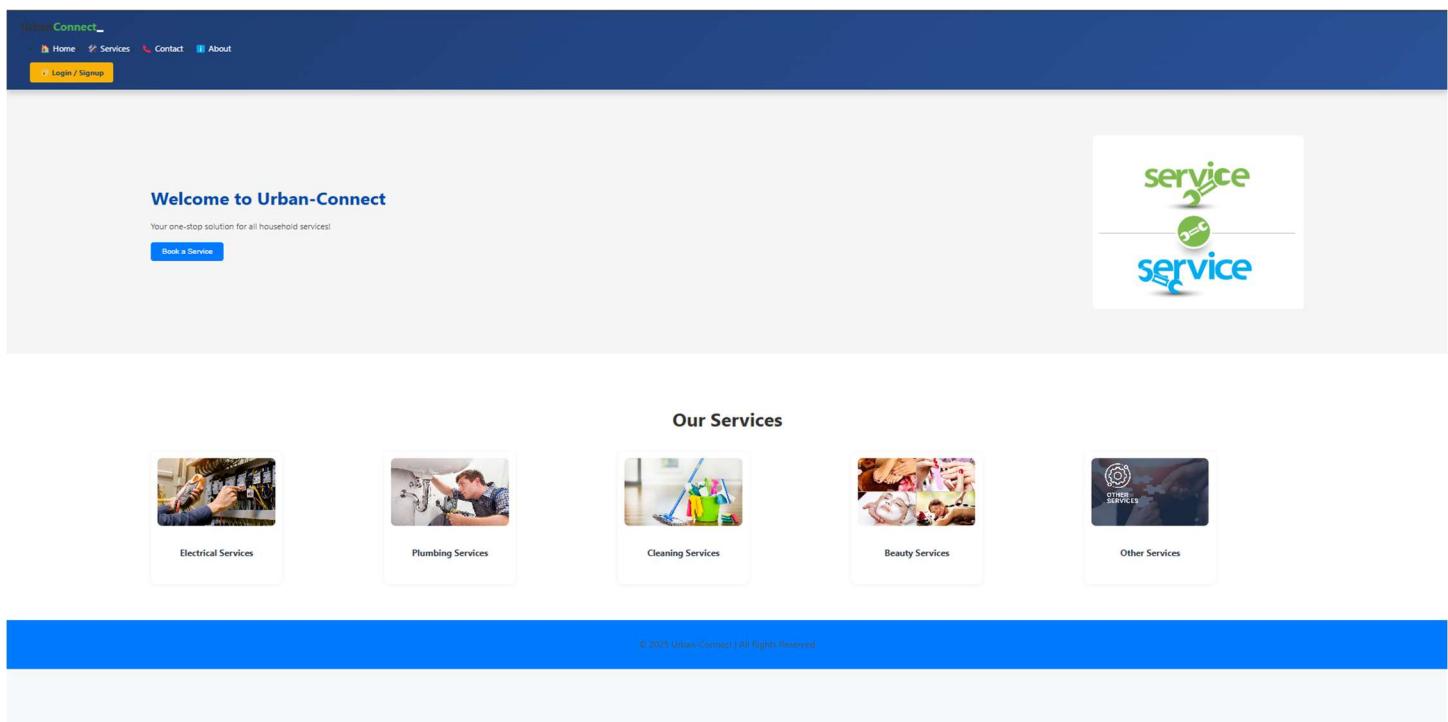
We made sure there can be only one admin for our project, we have used singleton pattern for this, when if an admin is made to register it will automatically fail to register more than one admin.

❖ GitHub link to the Codebase:

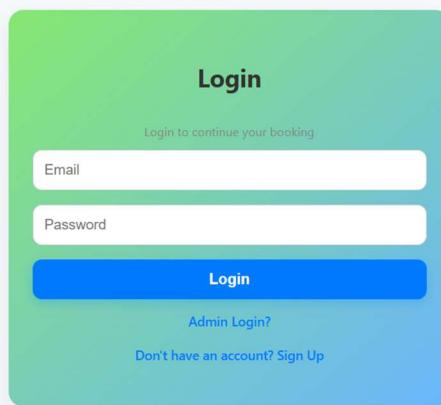
- ✓ https://github.com/vidhaanviswas/ooad_mini_project

❖ User-Interface Screen Shots:

○ Urban Connect:

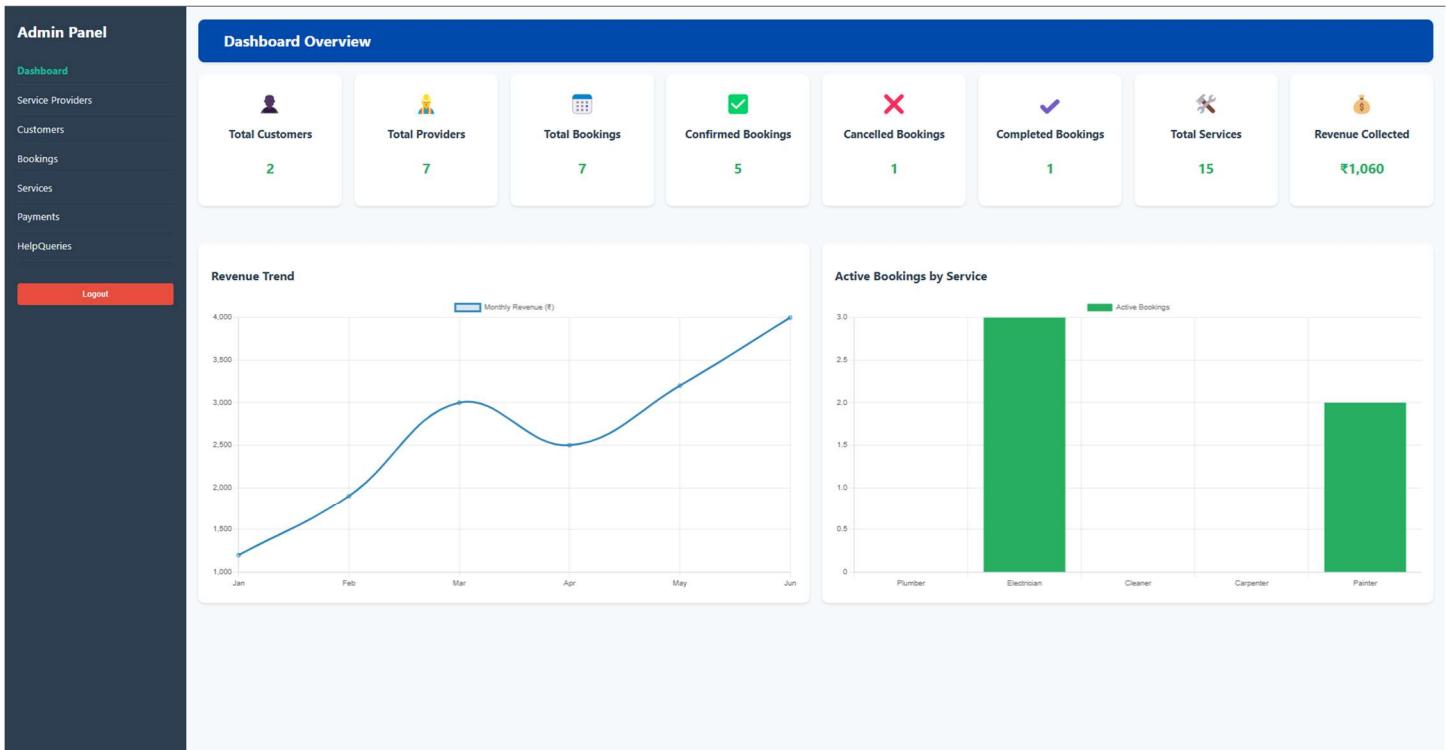


Screen-Shot 1 Urban Connect Home-Page

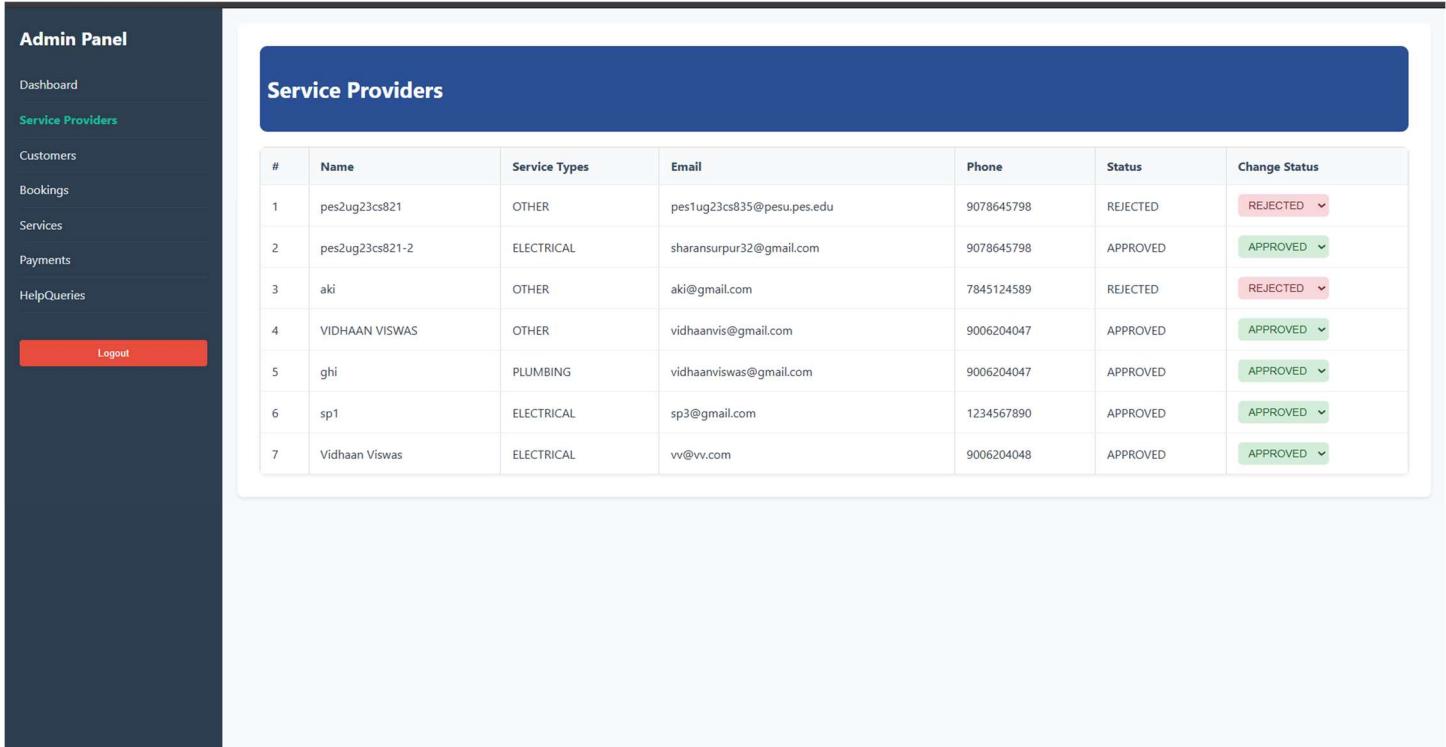


Screen-Shot 2 Urban Connect Login/Signup-Page

○ Admin Panel:



Screen-Shot 3 Admin Dashboard.



Screen-Shot 4 Admin Dashboard - Service Providers Management Section.

Admin Panel

- Dashboard
- Service Providers
- Customers**
- Bookings
- Services
- Payments
- HelpQueries

[Logout](#)

Customers

Items per page:

10

#	Name	Email	Phone	Joined On
1	VV	vidhaanviswas9006@gmail.com	9006204047	13-04-2025
2	Byte-Busters	Hi@gmail.com	9006204047	13-04-2025

Screen-Shot 5 Admin Dashboard - Customers List Section.

Admin Panel

- Dashboard
- Service Providers
- Customers
- Bookings**
- Services
- Payments
- HelpQueries

[Logout](#)

Bookings

Filter by Status

All Bookings

Items per page

10

Booking ID	Customer	Provider	Service	Date	Status
6	VV	Vidhaan Viswas	Circuit Troubleshooting	4/24/2025	CANCELLED
7	VV	Vidhaan Viswas	Circuit Troubleshooting	4/17/2025	CONFIRMED
8	VV	Vidhaan Viswas	Circuit Troubleshooting	4/19/2025	CONFIRMED
9	VV	aki	Painting	4/26/2025	CONFIRMED
10	VV	aki	Painting	4/26/2025	CONFIRMED
11	VV	sp1	AC Repair	4/26/2025	CONFIRMED
12	VV	pes2ug23cs821-2	AC Repair	4/19/2025	COMPLETED
13	VV	sp1	AC Repair	4/25/2025	CONFIRMED

Screen-Shot 6 Admin Dashboard - Bookings Section.

Admin Panel

Dashboard

Service Providers

Customers

Bookings

Services

Payments

HelpQueries

Logout

Services				
Service ID	Name	Category	Base Price (₹)	Description
1	AC Repair	ELECTRICAL	150.00	Fix air conditioner issues
2	Wiring Installation	ELECTRICAL	200.00	New electrical wiring setup
3	Circuit Troubleshooting	ELECTRICAL	120.00	Diagnose electrical circuits
4	Pipe Repair	PLUMBING	100.00	Fix water leaks and pipes
5	Drain Cleaning	PLUMBING	80.00	Clear blocked drains
6	Water Heater Installation	PLUMBING	250.00	Install new water heaters
7	House Cleaning	CLEANING	120.00	Full home cleaning service
8	Carpet Cleaning	CLEANING	90.00	Deep carpet cleaning
9	Window Washing	CLEANING	70.00	Clean all windows
10	Haircut	BEAUTY	40.00	Professional haircut service
11	Manicure	BEAUTY	30.00	Nail care service
12	Facial Treatment	BEAUTY	60.00	Relaxing facial care
13	Handyman Services	OTHER	150.00	General home repairs
14	Furniture Assembly	OTHER	100.00	Assemble furniture pieces
15	Painting	OTHER	200.00	Interior wall painting

Screen-Shot 7 Admin Dashboard - Services list Section.

Admin Panel

Dashboard

Service Providers

Customers

Bookings

Services

Payments

HelpQueries

Logout

Payments					
Payment ID	Booking ID	Amount (₹)	Status	Method	Date
6	6	₹120.00	SUCCESS	CASH	Invalid Date
7	7	₹120.00	SUCCESS	CASH	Invalid Date
8	8	₹120.00	SUCCESS	UPI	Invalid Date
9	9	₹200.00	SUCCESS	CARD	Invalid Date
10	10	₹200.00	SUCCESS	CARD	Invalid Date
11	11	₹150.00	SUCCESS	CARD	Invalid Date
12	12	₹150.00	SUCCESS	CASH	Invalid Date
13	13	₹150.00	SUCCESS	CASH	Invalid Date

Screen-Shot 8 Admin Dashboard - Payment Record Section.

Admin Panel

- Dashboard
- Service Providers
- Customers
- Bookings
- Services
- Payments
- HelpQueries**

[Logout](#)

Query ID	User Name	Message
1	Vidhaan Viswas	fdsbgtrn
2	Vidhaan Viswas	Hi
3	Vidhaan Viswas	HHGG
4	Vidhaan Viswas	dsvfeb

Screen-Shot 9 Admin Dashboard - Help Queries Record Section.

○ **Vendor or Service Provider Panel:**

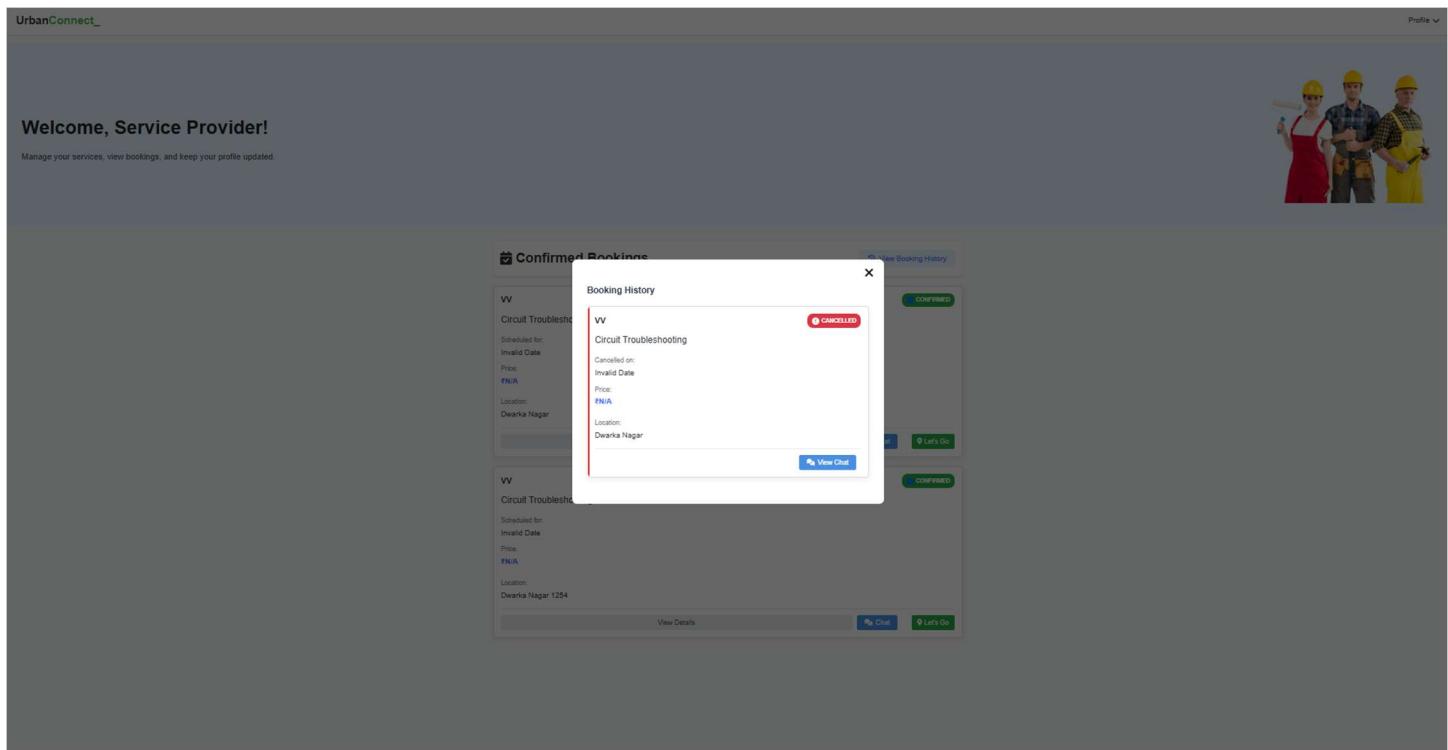
Welcome, Service Provider!

Manage your services, view bookings, and keep your profile updated.

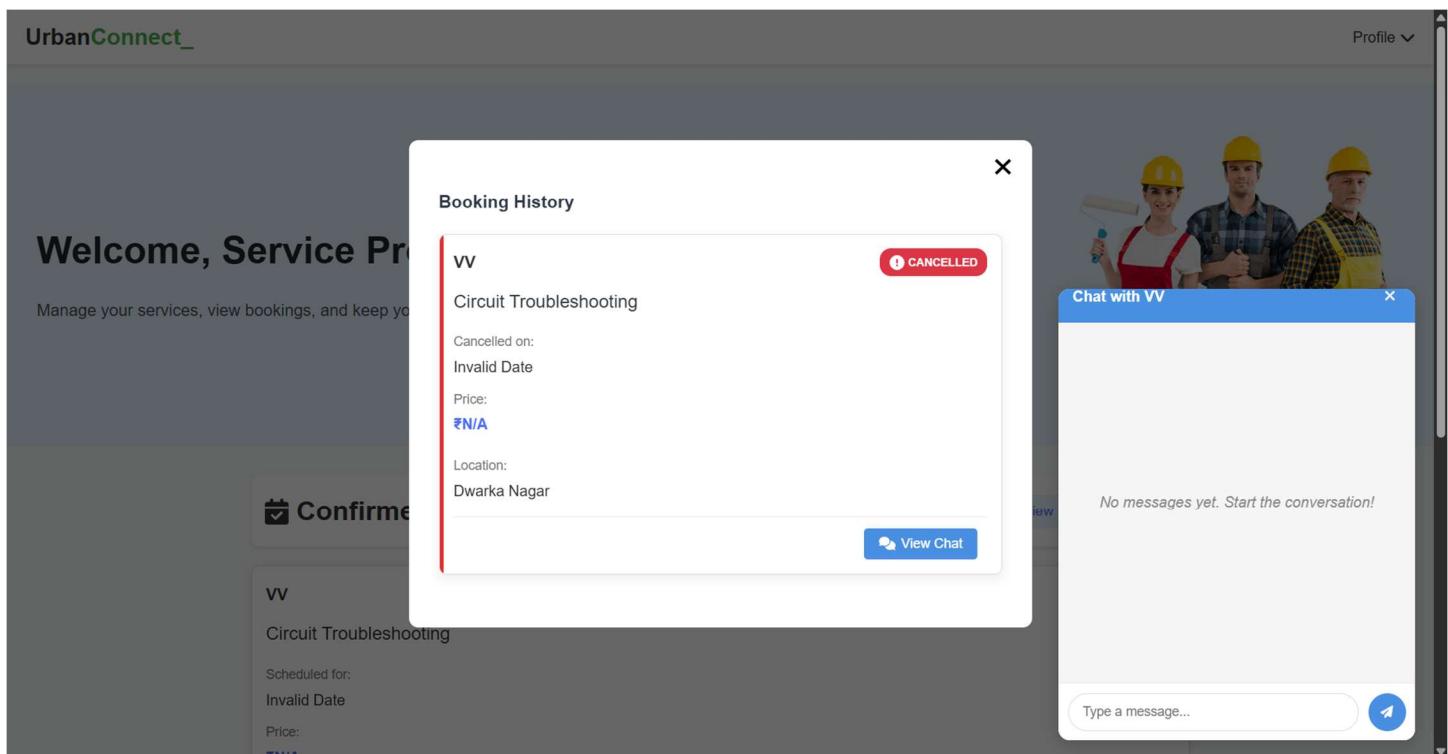


Confirmed Bookings		View Booking History
VV	Circuit Troubleshooting	CONFIRMED
Scheduled for:	Invalid Date	
Price:	IN/A	
Location:	Dwarka Nagar	
View Details Chat Let's Go		
VV	Circuit Troubleshooting	CONFIRMED
Scheduled for:	Invalid Date	
Price:	IN/A	
Location:	Dwarka Nagar 1254	
View Details Chat Let's Go		

Screen-Shot 10 Vendor Dashboard - Bookings received list.



Screen-Shot 11 Vendor - Booking History list.



Screen-Shot 12 Vendor - Chat window to connect with customer.

[← Back to Dashboard](#)

⌚ Received Payments

🔍 Search by booking ID, customer name, or service...

▼ All Payments

Booking #6	SUCCESS	
Customer_6		
Service_6		
Service Date: N/A	Payment Method: CASH	
Amount: ₹120	Transaction Date: Invalid Date	

Booking #7	SUCCESS	
Customer_7		
Service_7		
Service Date: N/A	Payment Method: CASH	
Amount: ₹120	Transaction Date: Invalid Date	

Booking #8	SUCCESS	
Customer_8		
Service_8		
Service Date: N/A	Payment Method: UPI	
Amount: ₹120	Transaction Date: Invalid Date	

Screen-Shot 13 Vendor - Received payments history.

○ Customer Panel:

Welcome, Valued Customer!

Book reliable services and manage your appointments with ease.



Explore Our Services



Electrical Services



Plumbing Services



Cleaning Services



Beauty Services



Other Services

Screen-Shot 14 Customer - Home Page.

Electrical Services

Available Services

AC Repair Fix air conditioner issues	₹150
Wiring Installation New electrical wiring setup	₹200
Circuit Troubleshooting Diagnose electrical circuits	₹120

Book Service

Selected Service:
AC Repair - ₹150

Select Service Provider:
sp1 - Price: ₹150

Select Date:
04/25/2025

Select Time:
07:35 AM

Your Location:
PES University

Your saved address has been loaded. You may edit it for this booking if needed.

Proceed to Payment

Screen-Shot 15 Customer - Service Booking Page.

Electrical Services

Available Services

AC Repair Fix air conditioner issues
Wiring Installation New electrical wiring setup
Circuit Troubleshooting Diagnose electrical circuits

Complete Payment

Booking Summary

Service: AC Repair
Provider: sp1
Date: 2025-04-25
Time: 07:35
Location: PES University
Total Amount: ₹150

Select Payment Method:

UPI Card Cash on Service

You'll pay ₹150 in cash after the service is completed.

Cancel **Pay ₹150**

Screen-Shot 16 Customer - Payment Page.

Back to Dashboard

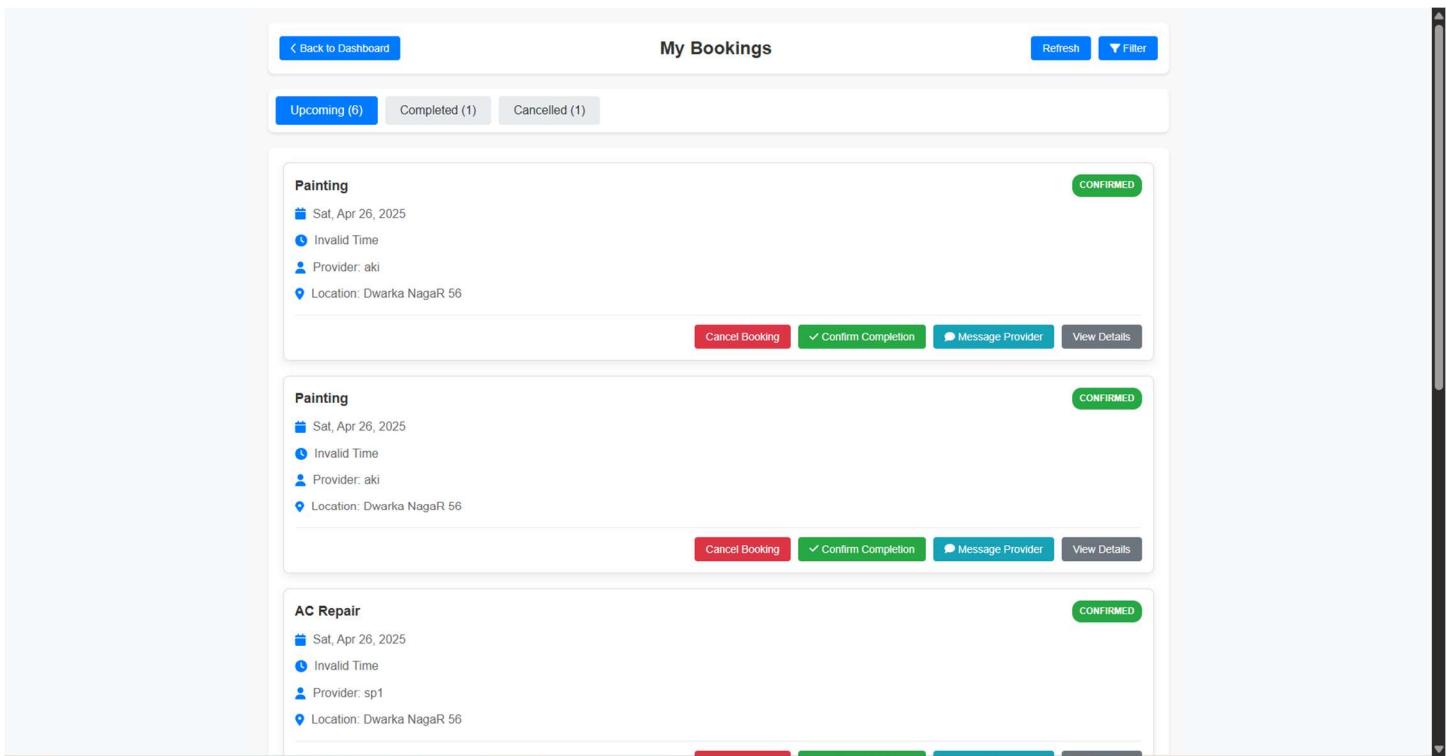
My Bookings

Upcoming (6) Completed (1) Cancelled (1)

Painting
Sat, Apr 26, 2025 CONFIRMED
Invalid Time
Provider: aki
Location: Dwarka NagaR 56

Painting
Sat, Apr 26, 2025 CONFIRMED
Invalid Time
Provider: aki
Location: Dwarka NagaR 56

AC Repair
Sat, Apr 26, 2025 CONFIRMED
Invalid Time
Provider: sp1
Location: Dwarka NagaR 56



Screen-Shot 17 Customer - Booking History Page.

Cancel Booking Confirm Completion Message Provider View Details

AC Repair
Sat, Apr 26, 2025 CONFIRMED
Invalid Time
Provider: sp1
Location: Dwarka NagaR 56

AC Repair
Fri, Apr 25, 2025 CONFIRMED
Invalid Time
Provider: sp1
Location: PES University

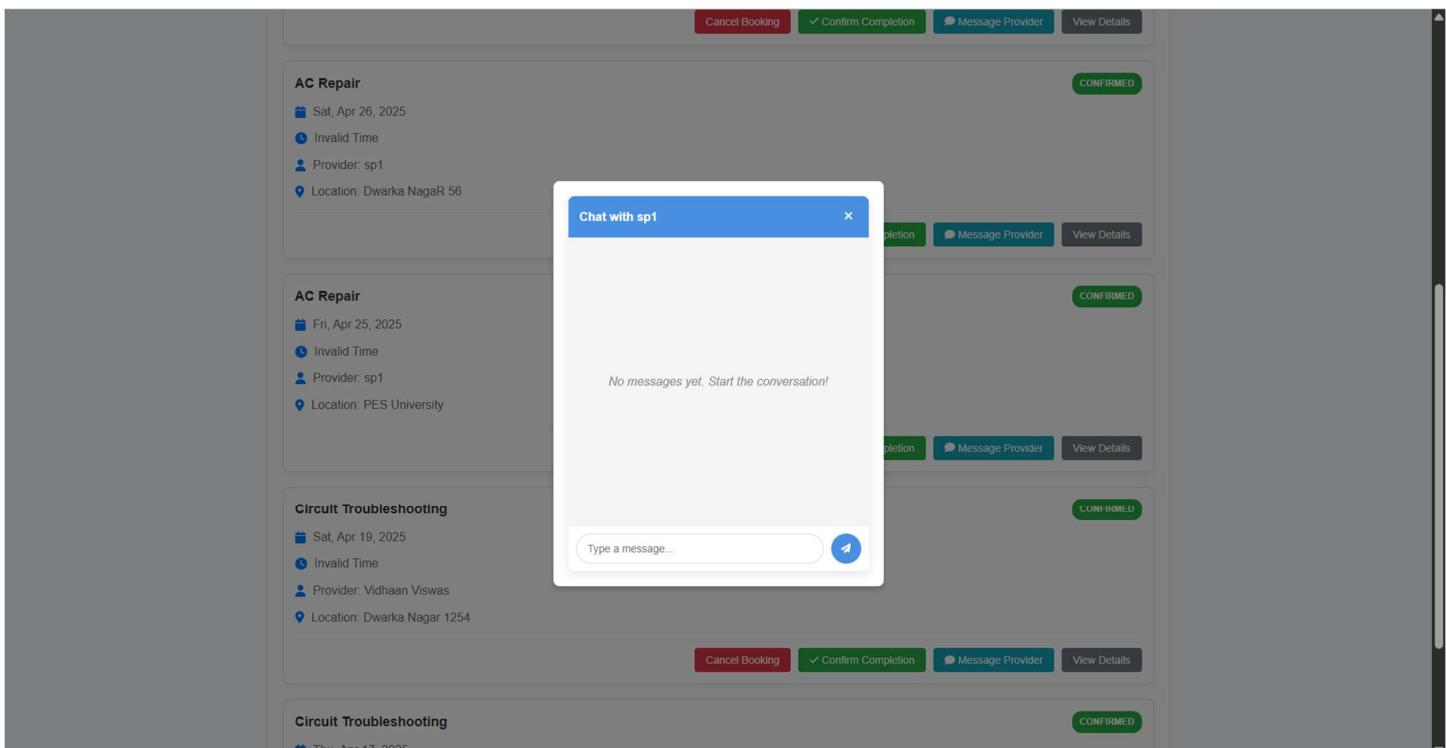
Circuit Troubleshooting
Sat, Apr 19, 2025 CONFIRMED
Invalid Time
Provider: Vidyaaan Viswas
Location: Dwarka Nagar 1254

Circuit Troubleshooting
Thu, Apr 17, 2025 CONFIRMED

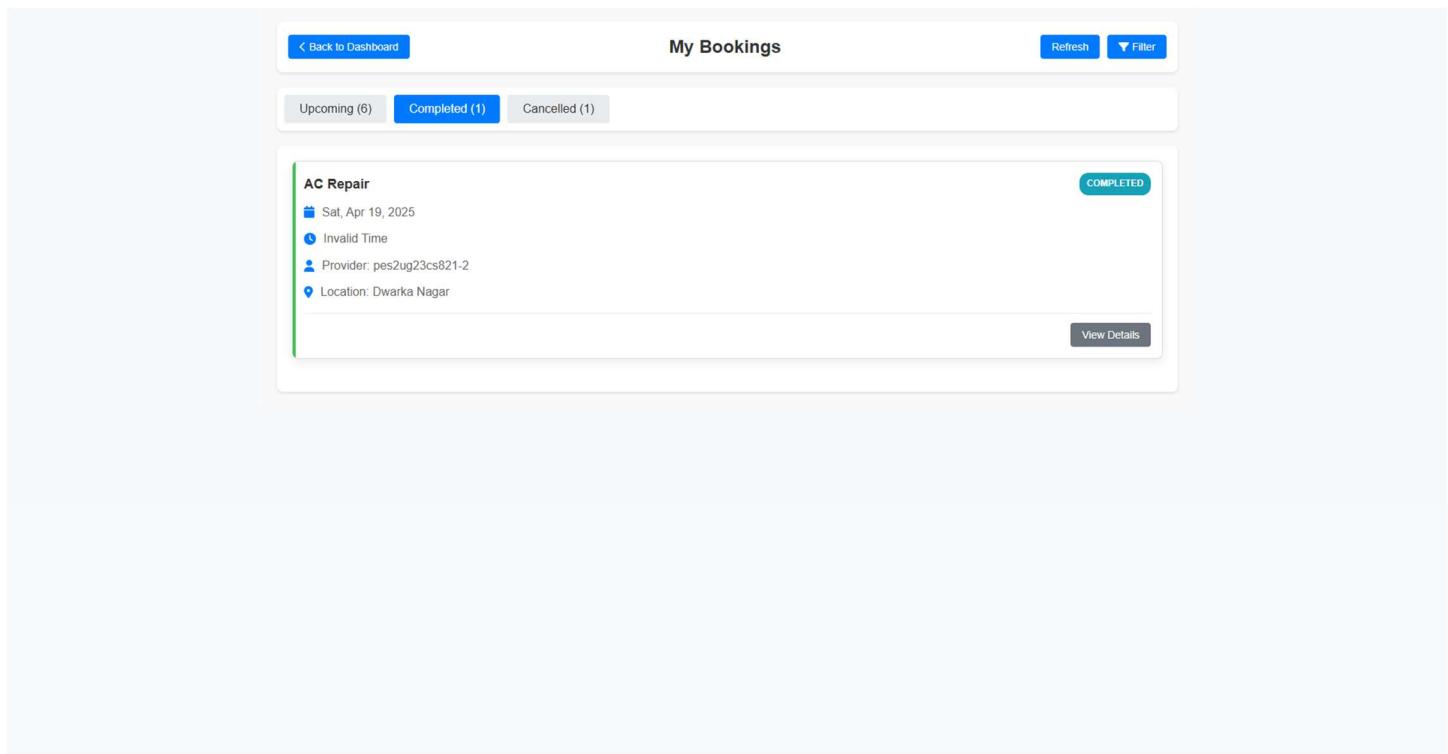
Chat with sp1

No messages yet. Start the conversation!

Type a message...



Screen-Shot 18 Customer - Chat window to connect with vendor.



Screen-Shot 19 Customer – Service completed Booking list.

A screenshot of the 'UrbanConnect Help Center' page. The page features a header with the 'UrbanConnect' logo and 'Help Center' text, followed by a sub-header 'Find answers or ask your own question.' Below this is a 'Frequently Asked Questions' section. The first question, 'How do I book a service?', is answered with the text: 'You can book any service by selecting a category on the dashboard and filling out the booking form.' The second question, 'Can I cancel or reschedule a booking?', is answered with the text: 'Yes, visit the "My Bookings" section to cancel or reschedule an appointment.' At the bottom of the page is a 'Still Have a Question?' section, which contains a text input field for users to enter their query and a 'Submit Query' button.

Screen-Shot 20 Customer - Help Page and Query submission.

❖ Individual contributions of the team members

Name	Contribution to Module
SAI MOURYA N DODDAMANI	I implemented the Factory Pattern, where the service class interacts with a factory instead of directly accessing Message.java and Message-DTO. I developed key features that manage the customer-service provider interaction from booking confirmation to completion including real-time chat, booking status updates, and navigation via Google Maps. I also ensured only customers can cancel bookings, as guided by our course instructor. Additionally, I handled booking detail retrieval on both frontend and backend, designed the overall class diagram and a 3NF compliant database schema. Vidhaan and I also resolved a critical infinite recursion issue caused by Jackson configuration.
SANATH SHETTY	In our project, I was responsible for developing the complete booking module, covering both the front end and back end. I implemented a system where the user can book from five different services, and each service includes three specific subcategories. Additionally, I developed the Help Centre feature, which allows users to directly send their queries to the admin through a dedicated button. I also worked on the profile management section, enabling both users and service providers to edit and update their profiles. Moreover, during the booking process, the system intelligently matches the booking request with the service provider who has registered for the selected service and subcategory, ensuring smooth and relevant assignments.
SHASHANK R PATIL	I implemented the complete authorization flow for both users and the admin, including the frontend login and signup pages. I applied the Singleton Pattern to ensure only one admin can be registered, blocking any further attempts. For payments, I handled both frontend and backend, adhering to the Single Responsibility Principle by keeping Payment.java focused solely on payment logic. I also designed the state and activity diagrams to capture the overall SaaS workflow. Additionally, access to the service provider dashboard is restricted until admin approval - a feature implemented by my teammate Vidhaan.
VIDHAAN VISWAS	I initiated the project implementation and designed the admin panel with full control to manage customers, service providers, bookings, payments, and real-time booking tracking. I built a dashboard displaying total income, booking counts, and statuses. I also created the use case diagram and ensured all planned features were delivered. A key functionality I implemented was restricting service provider access to the dashboard until admin approval. Saimourya and I resolved a critical infinite recursion issue in an endpoint related to user details post-booking by configuring Jackson and using a custom mapper.