**TRAVELLING COURIER SERVICE**

**Technical Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** |
| **Name** | 1.Naveen Vemuri- 232883  2.Ramyanath Chakraborty- 175014  3.Ganesh Narayanan-143698 | Sharath Kumar | Sharath Kumar |
| **Role** | POD Leader  POD Member  POD Member |  |  |
| **Signature** |  |  |  |
| **Date** |  |  |  |

**Table of Contents**

[**1.0 Introduction** **3**](#_Toc9818)

[**1.1 Purpose of this document** **3**](#_Toc9819)

[**1.2 Project overview** **3**](#_Toc9820)

[**2.0 Solution Summary** **3**](#_Toc9821)

[**2.1 Scope** **3**](#_Toc9822)

[**2.2 Assumptions** **3**](#_Toc9823)

[**2.3 Dependencies** **4**](#_Toc9824)

[**2.4 Risks** **4**](#_Toc9825)

[**3.0 Schematic Diagram** **5**](#_Toc9826)

[**4.0 System Design** **6**](#_Toc9827)

[**4.1 Proposed design** **6**](#_Toc9828)

[**5.0 Database Design** **7**](#_Toc9830)

[**5.1 Data Model** **7**](#_Toc9831)

[**6.0 Terms and conditions** **9**](#_Toc9833)

[**7.0 Application flow** **9**](#_Toc9836)

# 1.0 Introduction

## 1.0 Purpose of this document

The purpose of this document is to document the technical design, component details and Database design. This will also capture the scope, assumptions, risk, dependencies of this project.

## 1.1 Project overview

This project aims to create an application that allows travelers to deliver items or packages on behalf of others while they are traveling from Chennai to Bangalore. Travelers can update their travel details, including the source, destination, date and time of departure and arrival, in the application. If someone wants to deliver an item or package to the destination that the traveler is going to, they can contact the traveler and transfer the items to be delivered in the destination without depending on a courier company. The traveler can also charge a small fee for delivering the courier.

# 2.0 Solution Summary

## 2.0 Scope

## The application will have three modules: Admin, Traveler, and Courier. The Traveler module allows travelers to update their travel details and search for delivery opportunities. The Courier module enables individuals who need to deliver items to update their details and package details, and search for travelers to deliver their packages. The Admin module allows the admin to view, verify, and validate the details of travelers and couriers, and delete false entries and expired details.

## 2.1 Assumptions

* Administrator have access to add and remove user in database.
* Administrator and User can login with their credentials.
* User can add or update details of his profile.
* User search for other user travelers and book them for courier transfer.

## 2.2 Dependencies

1. Hardware Requirement:

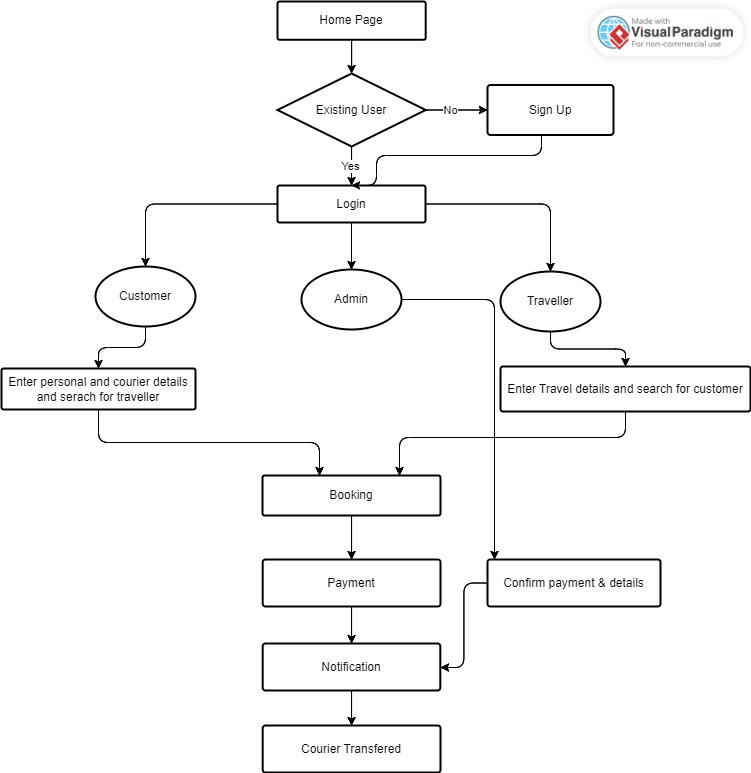
● Developer Laptop with minimum 8 GB RAM

1. Software Requirement:
   * Framework 3.0.2 and above
   * STS / Intelij Idea
   * MySQL Management studio 2019 and above
   * Kubernetes
   * Docker

## 2.3 Risks

* Payment Integration
* Deployment (AWS)

# 3.0 Schematic Diagram



# 4.0 System Design

## 4.0 Proposed design

1. **Customer Module:**
   * Customer Sign up - After successful Sign up , Customer can login with user name and password.
   * Customer can Search for traveler and book him for transfer courier.
   * customer can update profile information and can do the payment for booking
2. **Admin Module:**
   * Admin can sign up and login with his credentials
   * Admin can manage all the details of other users
   * Admin can generate bills and send notifications

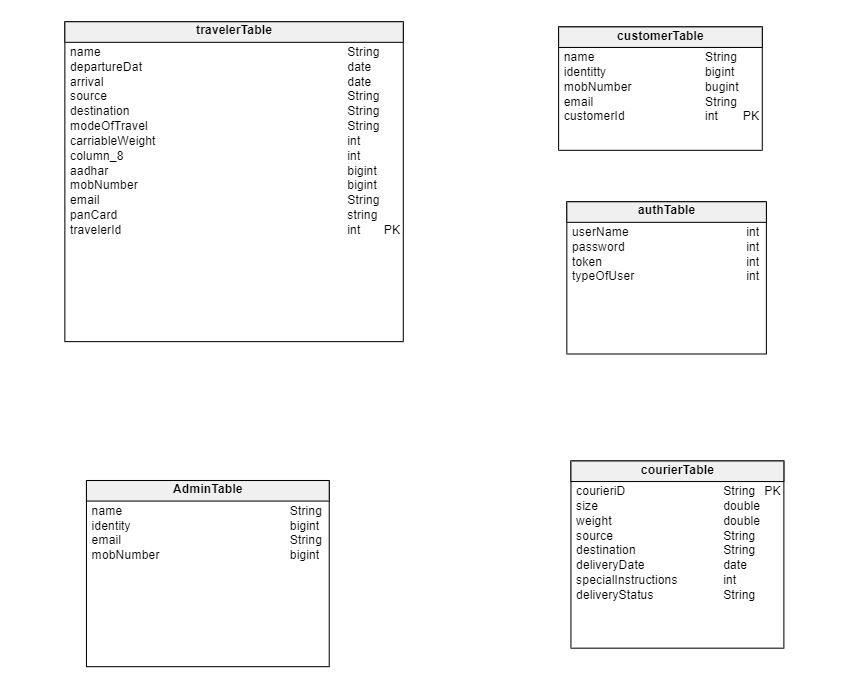
## 4.1 Component inventory

**● Components:**

1. Administrator
2. Customer
3. Traveler

# 5.0 Database Design

## 5.0 Data Model



# 6.0 Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of***

***UST Network, We have a Zero Tolerance Policy. Kindly adhere to 100%***

***Compliance at all times.***

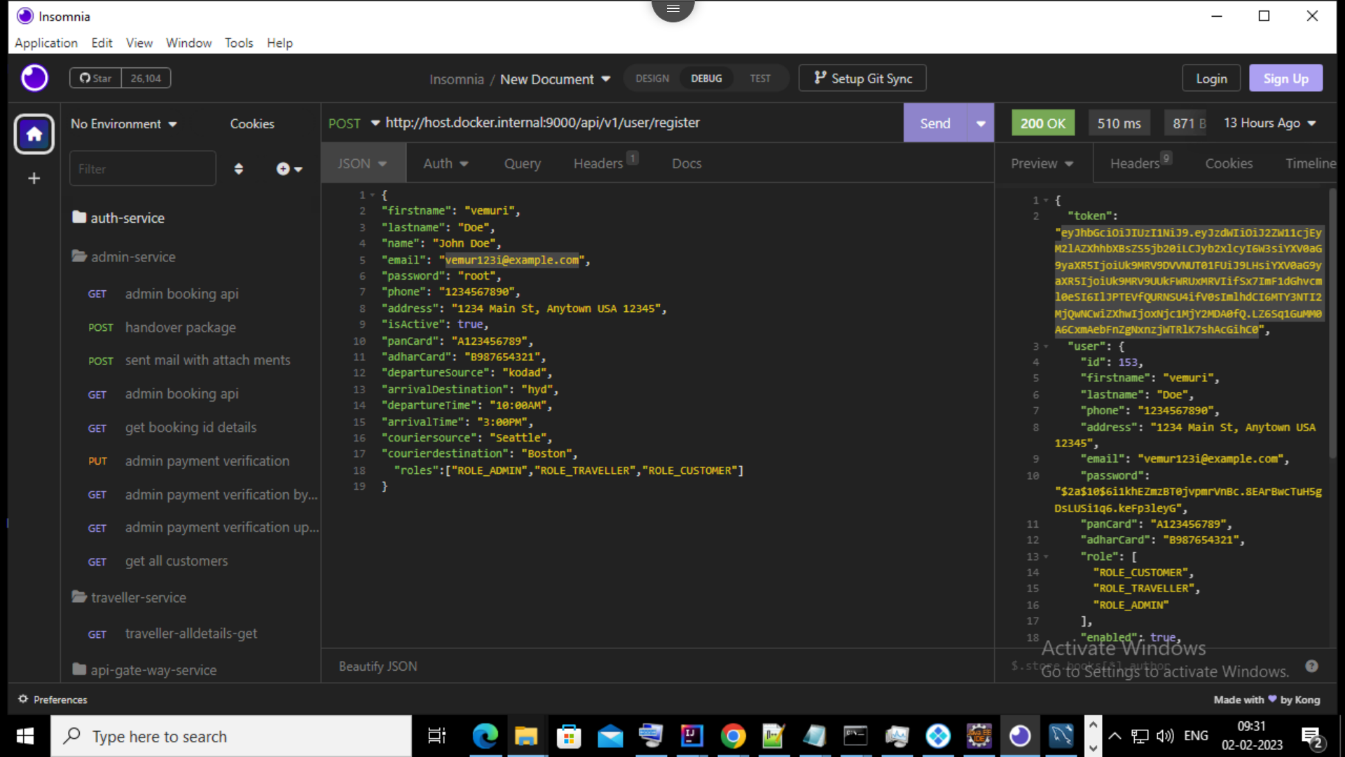
|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

# 7.0 Application Flow

1. User Sign up

User can enter all the basic details including email and password for log in in the register api. Existing email id won’t allow.

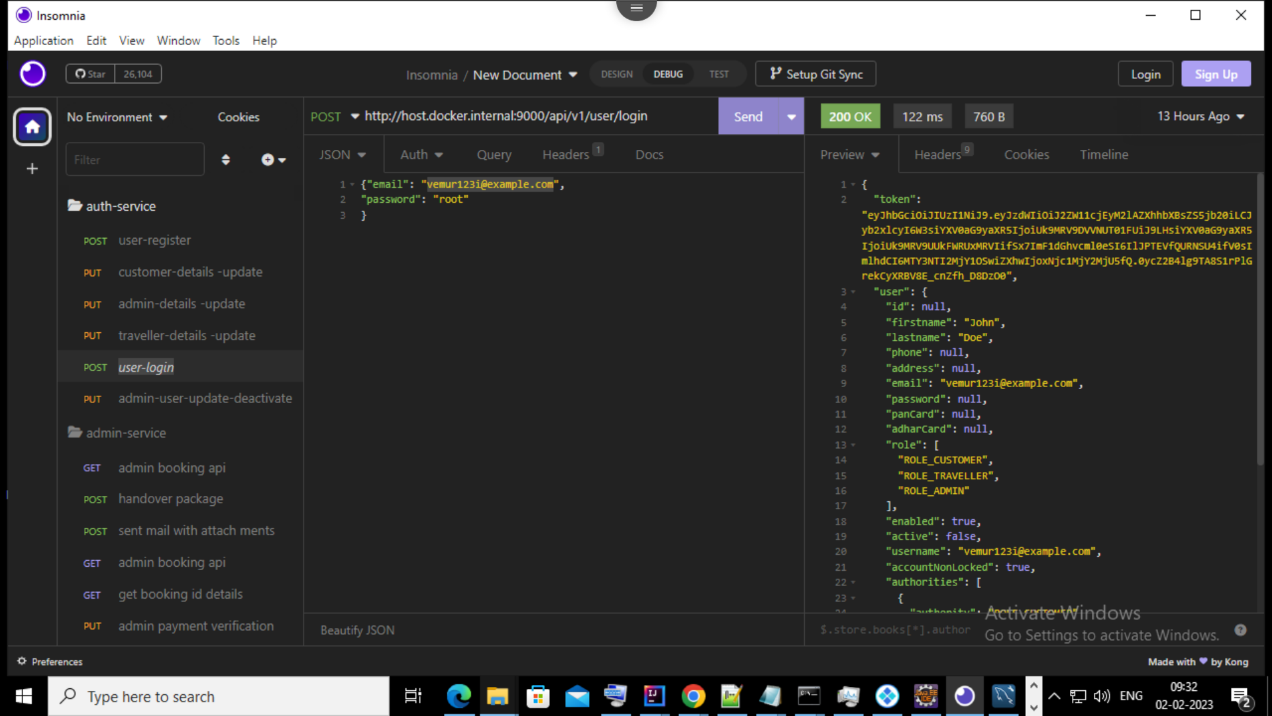
http://host.docker.internal:9000/api/v1/user/register



1. User log in

User can login to the application using email and password.

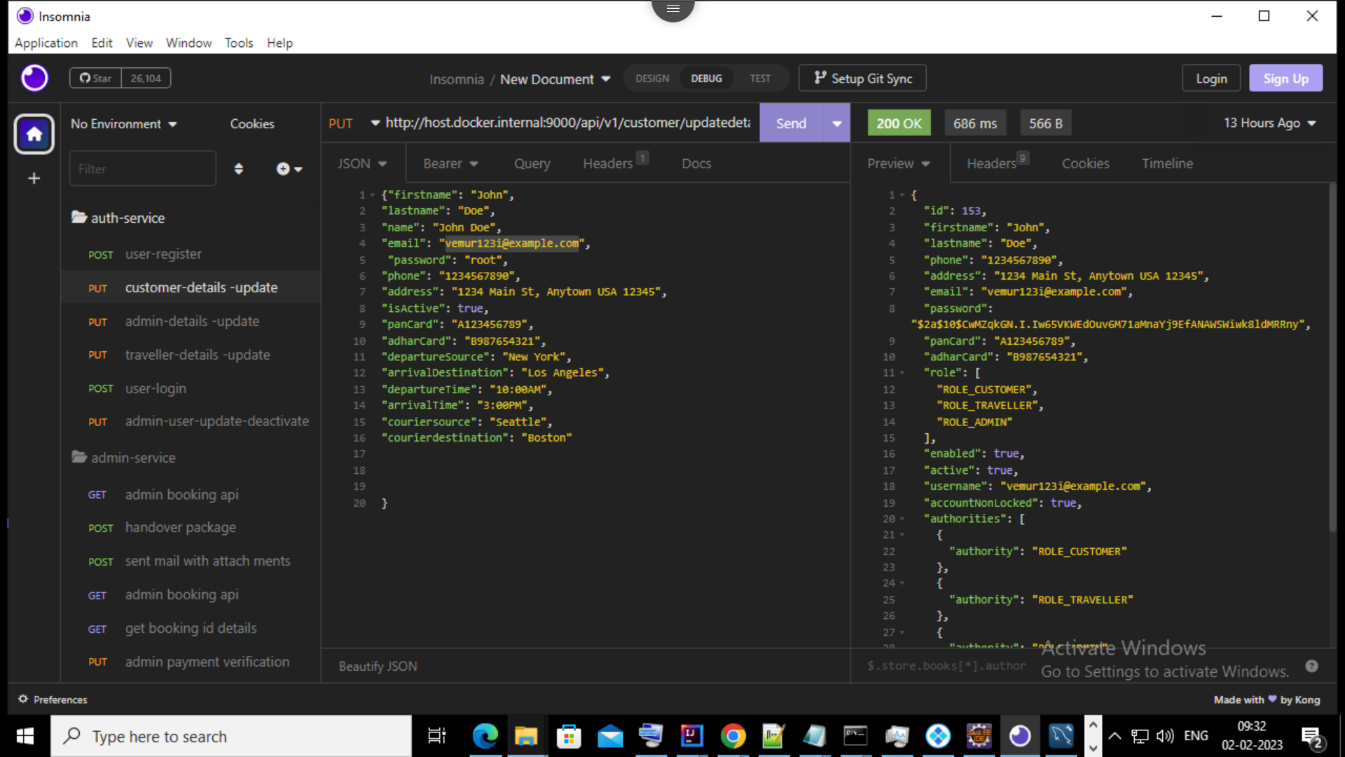
http://host.docker.internal:9000/api/v1/user/login

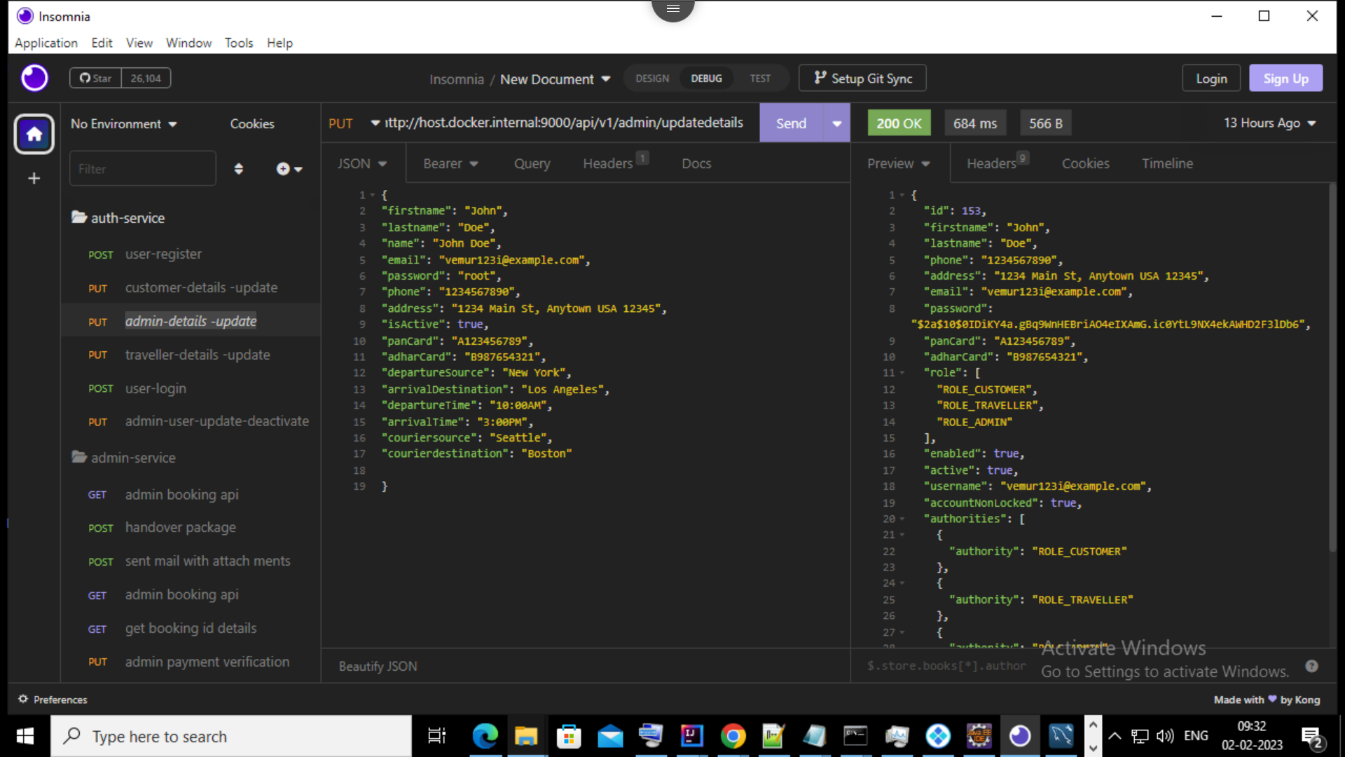


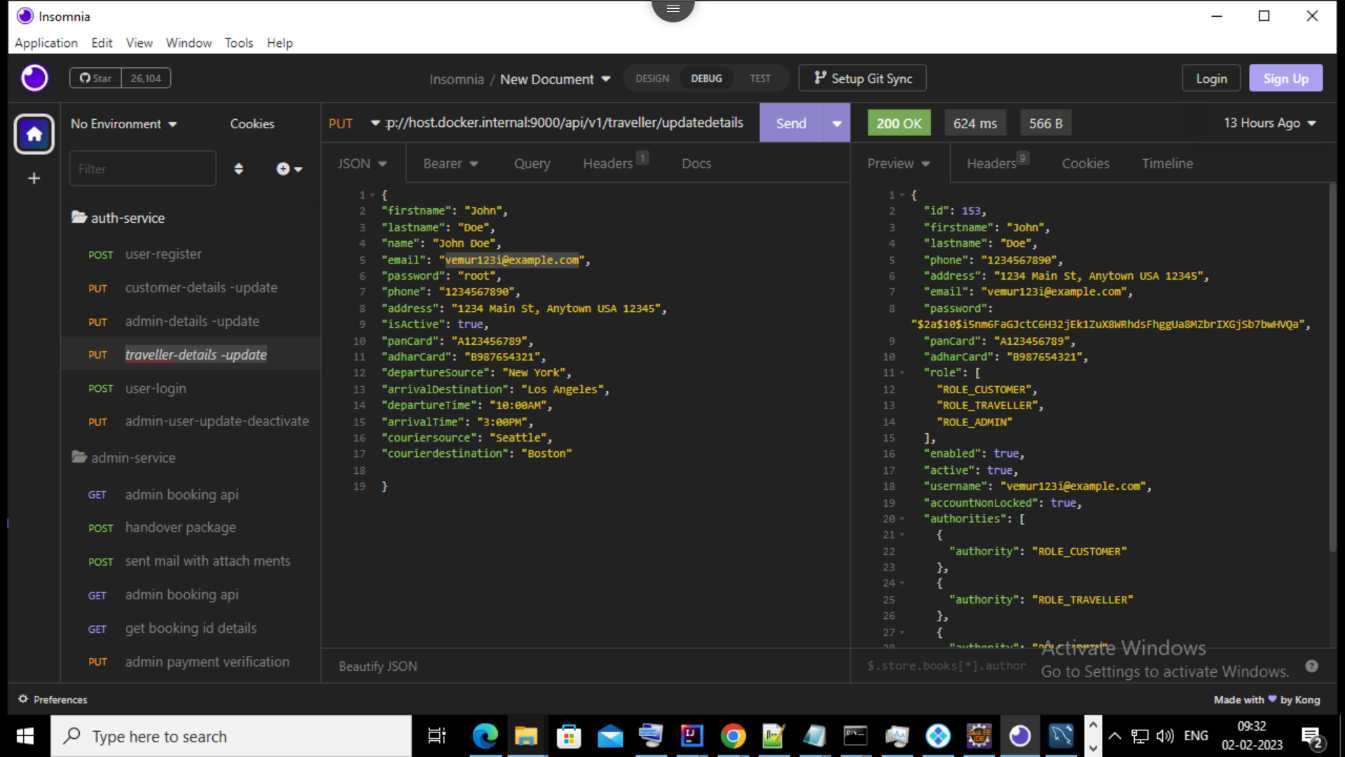
1. Update details

Admin can update all the details and users can update only their own details.

http://host.docker.internal:9000/api/v1/customer/updatedetails



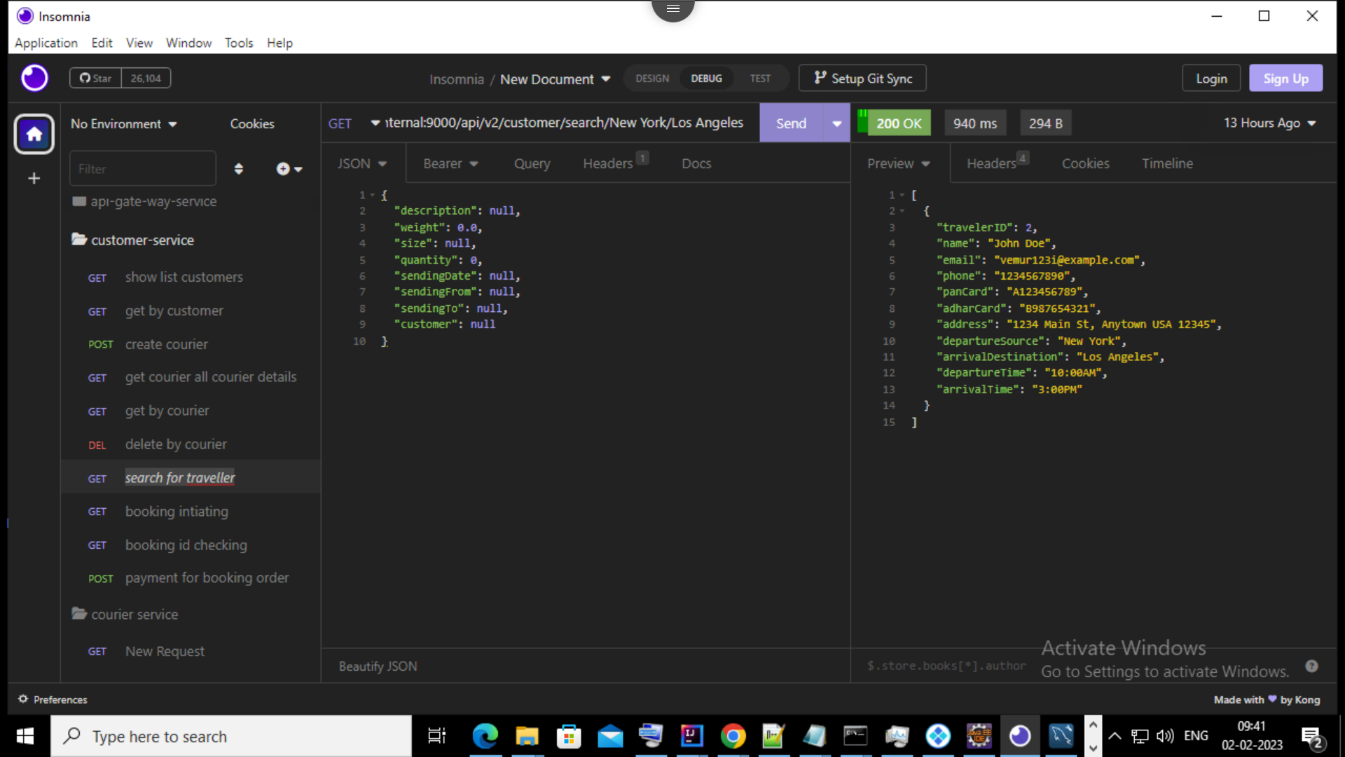




1. Search for traveler

Customer can search for traveler using the source and destination

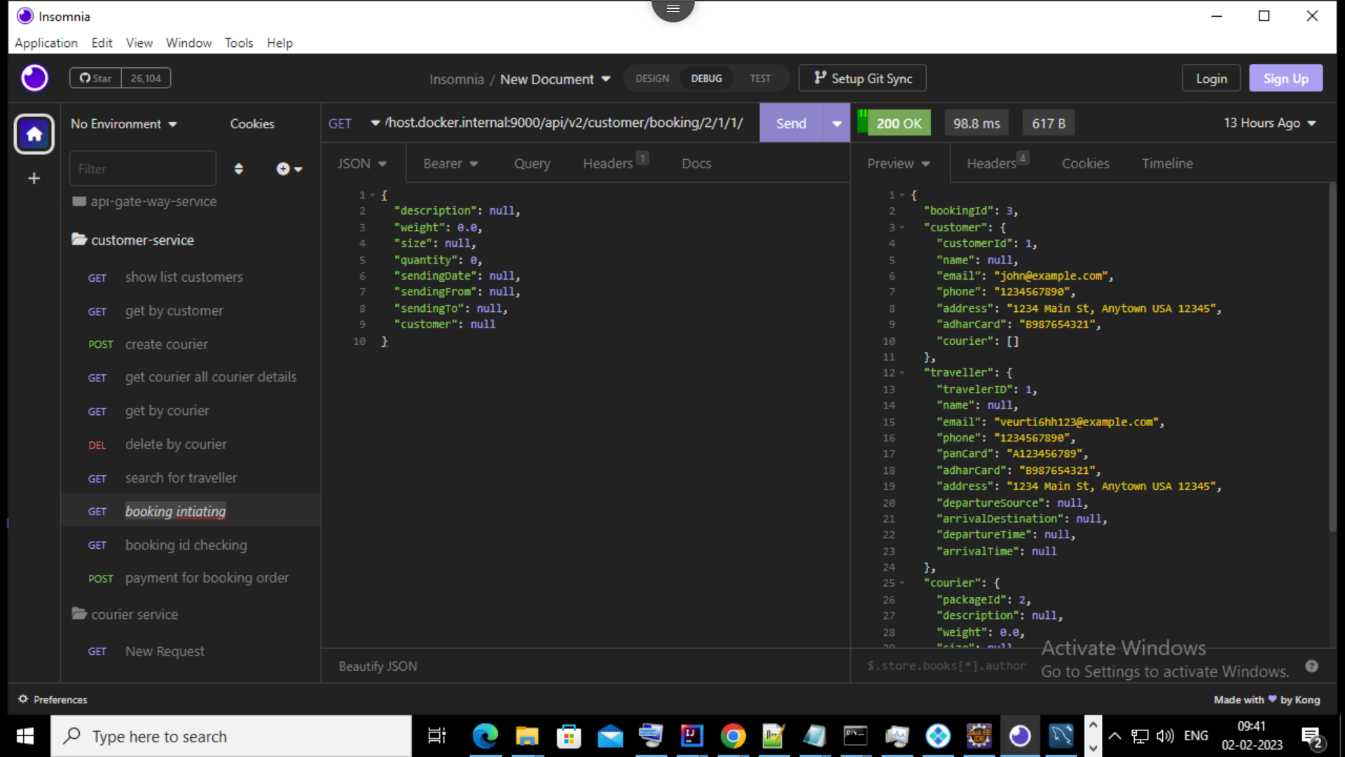
<http://host.docker.internal:9000/api/v2/search/new> york/losangels



1. Booking

Customer can book the traveler for courier transfer

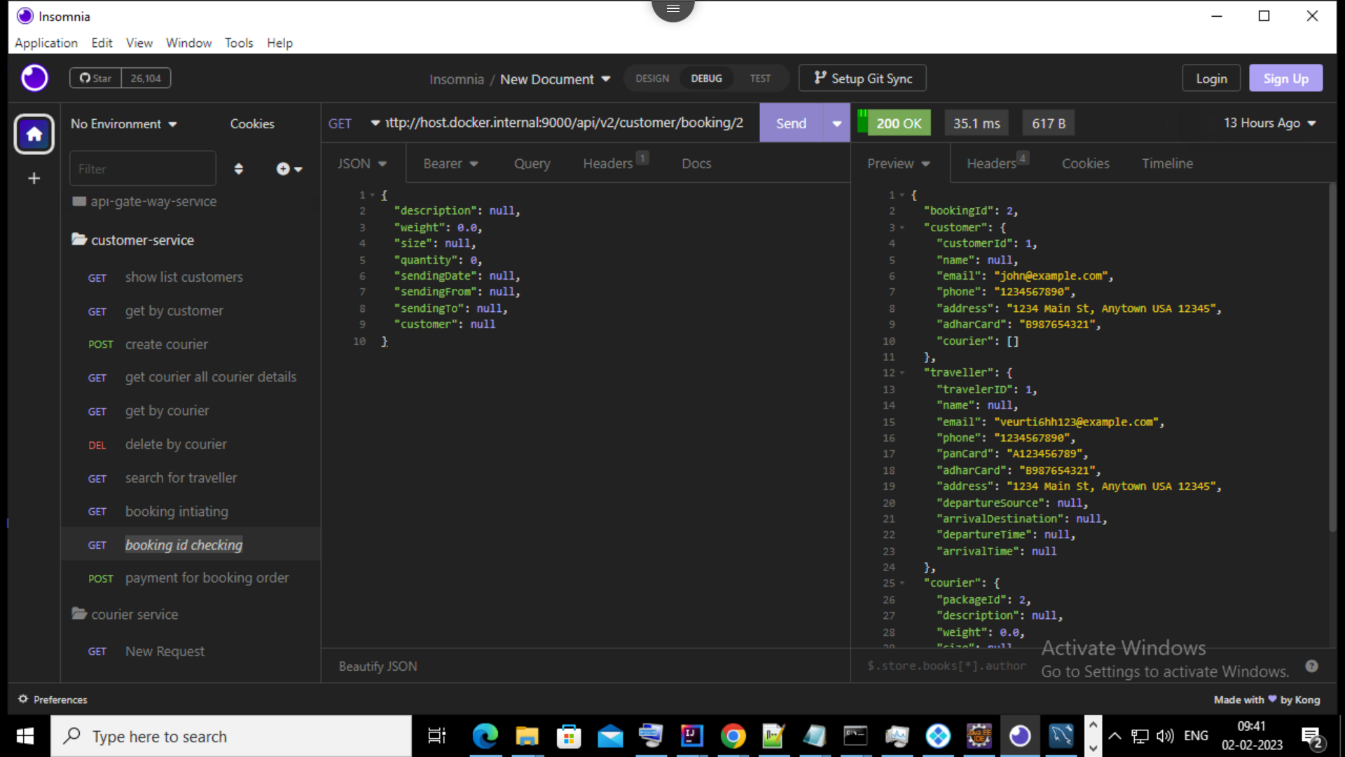
http://host.docker.internal:9000/api/v2/customer/booking/2/1/1



1. Booking Id Checking

Admin will verify the booking id

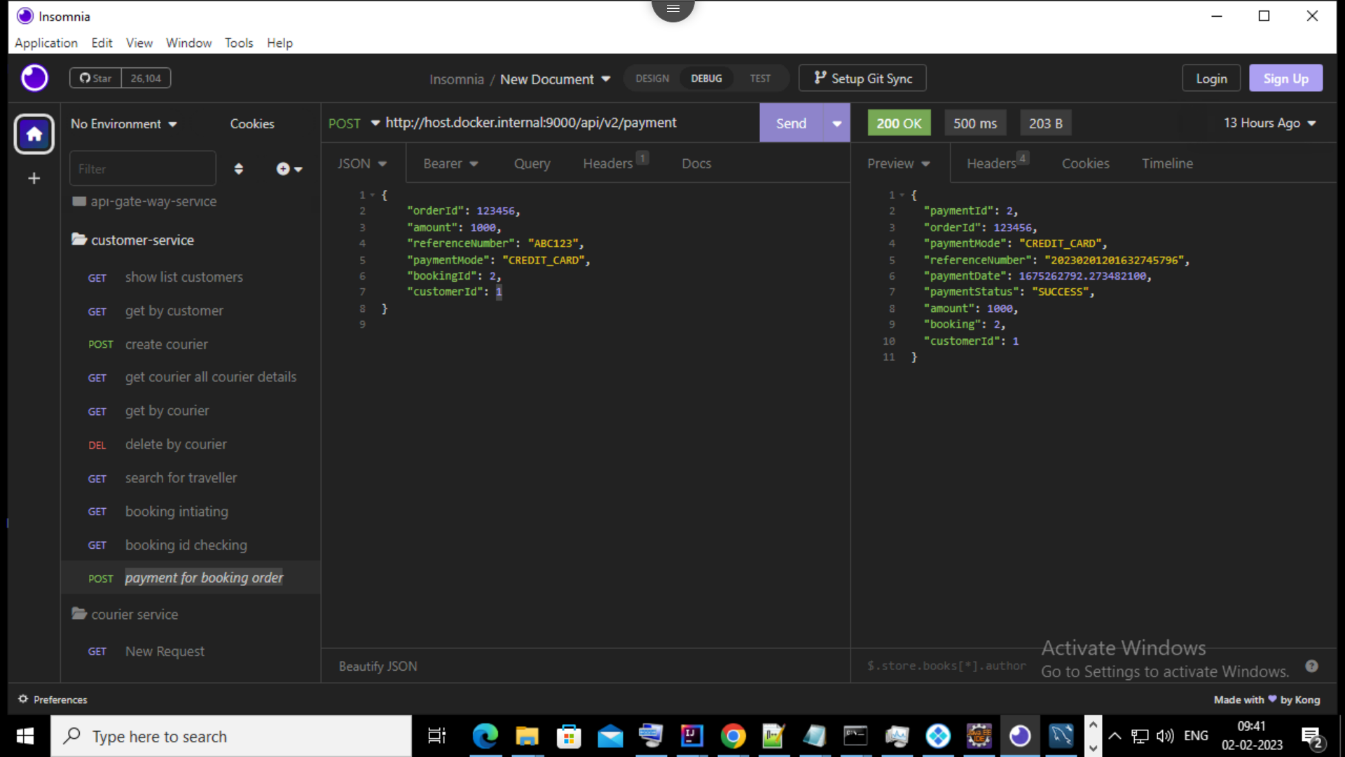
http://host.docker.internal:9000/api/v2/customer/booking/2



1. Payment

Customer needs to do the payment

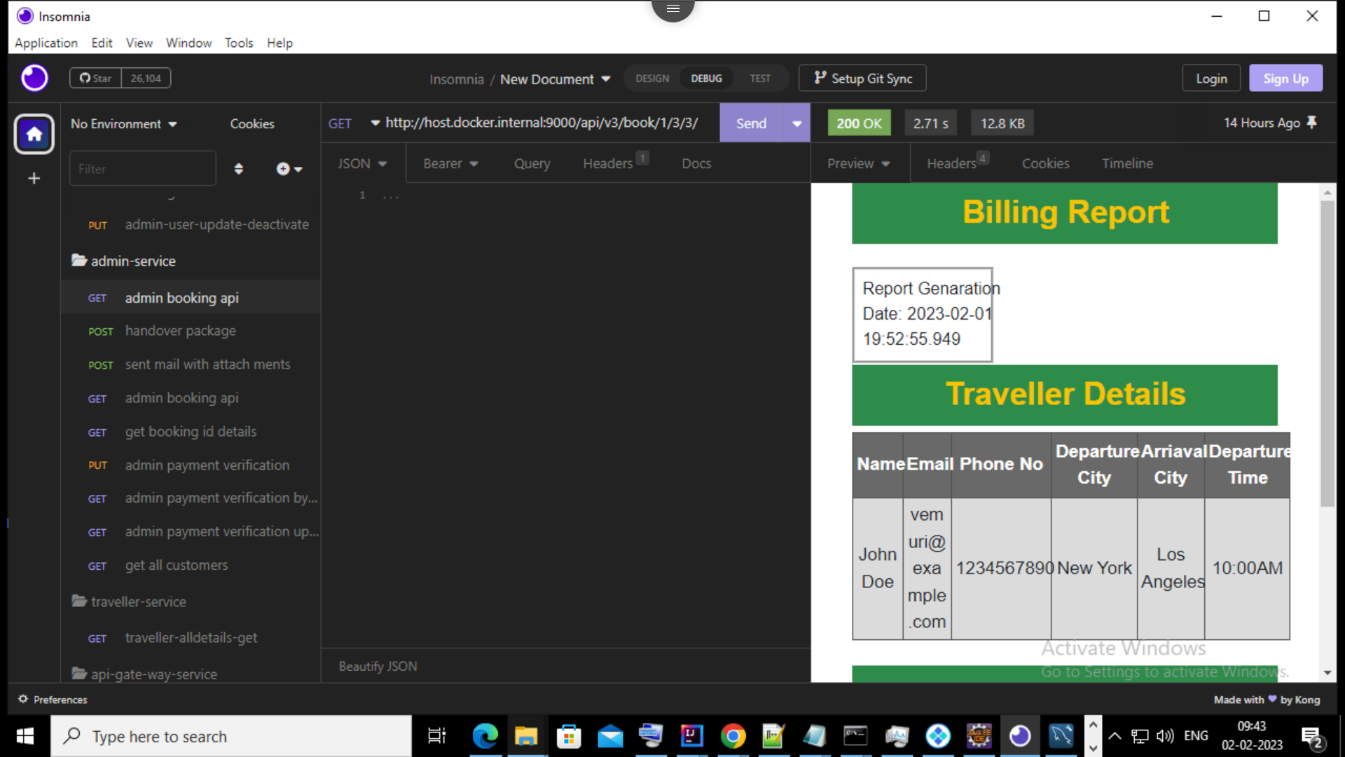
http://host.docker.internal:9000/api/v2/payment



1. Bill Generation

Admin can generate bills for payments and courier transfer.

http://host.docker.internal:9000/api/v3/book/1/3/3



1. Notification

Admin can send the notification to both customer and traveler with and without attachment.

http://host.docker.internal:9000/api/v3/handover

