Informatics Institute of Technology

in collaboration with

University of Westminster, UK

5COSCO19

# Object Oriented Programming

Coursework

# Documentation


| | |
|---|---|
| Name: | **Gaindu Amarasinghe** |
| IIT No: | **20230161** |
| UoW No: | **w2053783** |
| Module Leader: | **Poravi Guganatha** |

# Table of Contents

# 1. System Introduction.

The Real-Time Ticketing System is one of those big projects which simulates and manages the behind-the-scenes real-time ticketing mechanism. It is a typical model for integrating contemporary software engineering methodologies with practical ticketing scenarios to develop a framework that efficiently manages ticket issuance, acquisition, and observation.

The system can be divided into three major subsystems:

1. **Command-Line Interface:**

A Java-based interface for the simulation of back-end processes of ticket management. The vendors can publish tickets, customers can buy tickets, and all of the system's general behaviour is visible to the administrators in real time. Real-time update in the terminal with complete details about the system's operation.

2. **Frontend (Angular):**

It is an Angular-based web application that supports quick and intuitive non-fuzzy interaction with the system. Features include monitoring ticket availability, viewing real-time logs, and configuring ticketing parameters. WebSockets provide real-time updates and allow users to interact dynamically.

3. **Backend (Spring Boot):**

A resilient Spring Boot application with business logic and publishing of the REST API endpoint. Allows immediate updates via WebSockets so clients can observe live and make changes. Configuration, ticket operations, and logging of activities of systems are included.

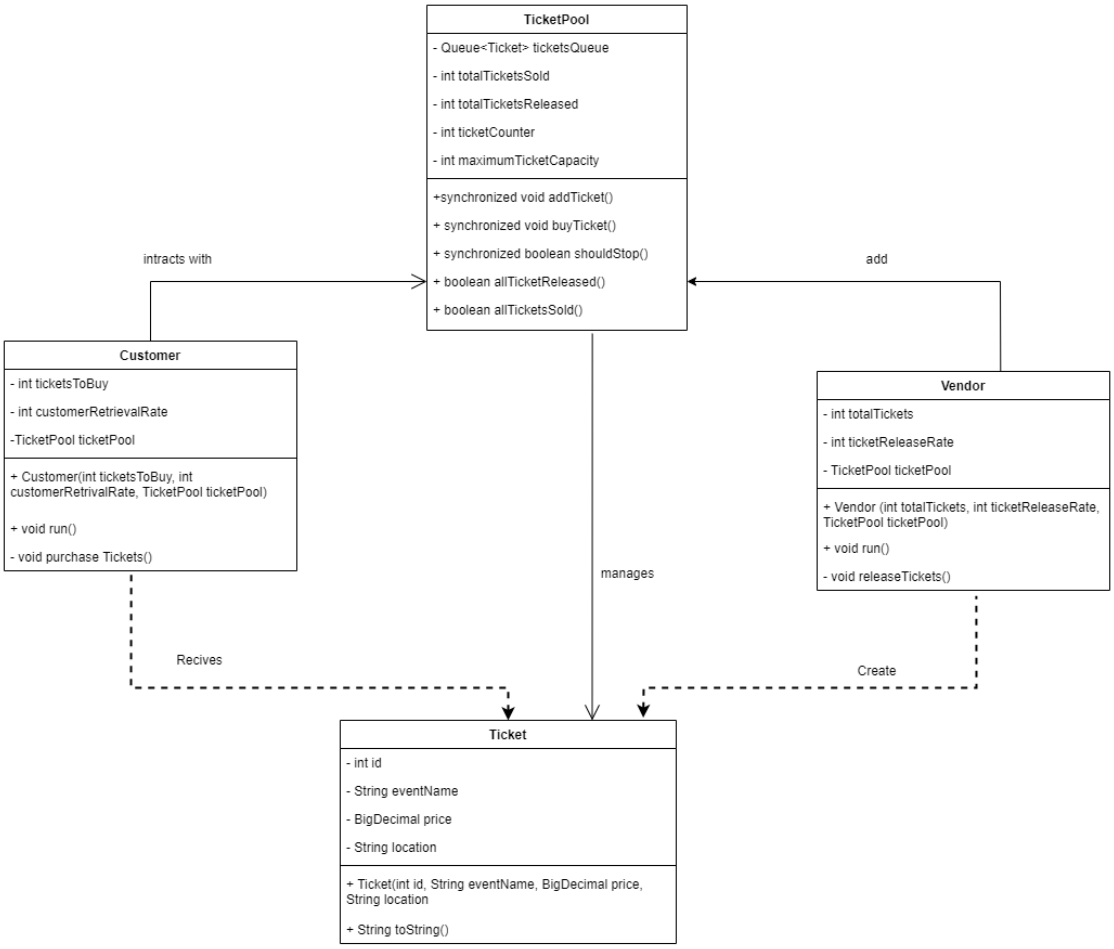
**Key Features**

- Real-time Updates: WebSockets updates CLI and web clients' live logs and tickets.
- Multi-threaded simulation means vendors and customers can work simultaneously, simulating a real-life ticketing scenario.
- Dynamic Configuration: Using the Web interface, the administrator can adjust ticket capacity, release, and retrieval rates.
- Extensive logging mechanisms have been implemented to log every significant event in the system: ticket purchasing, system releases, error incidents, and real-time visualisation.

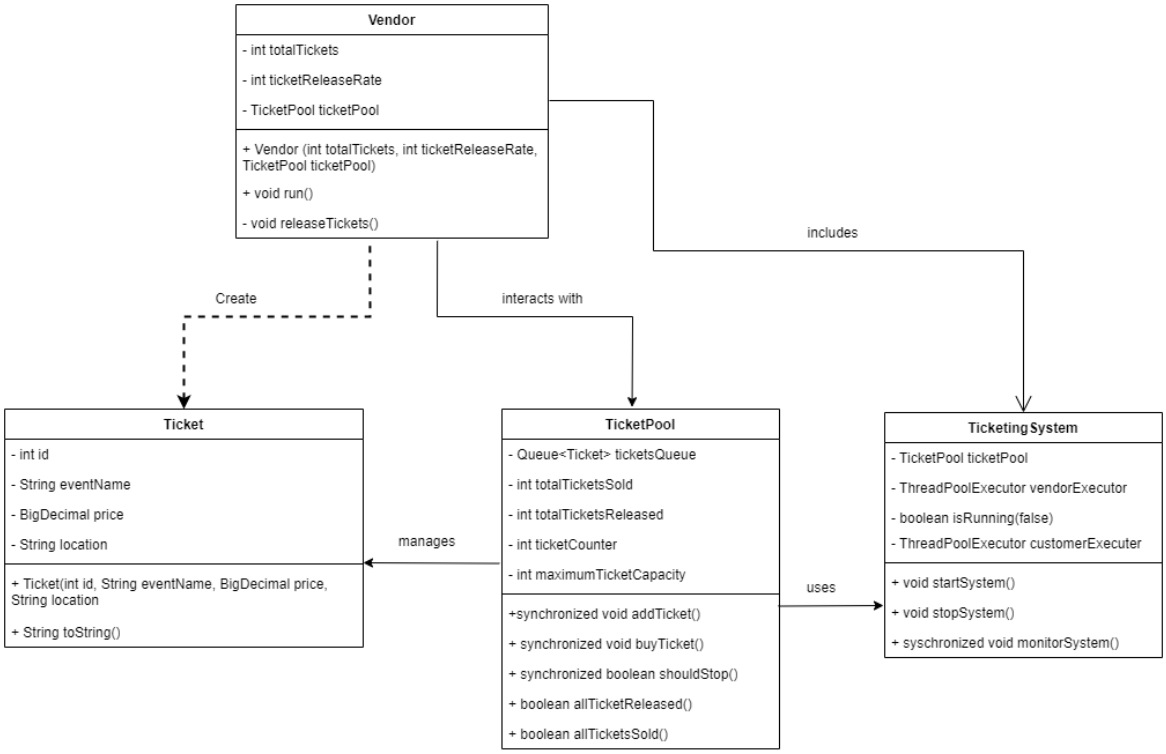
# 2. Diagrams

## 2.1. Class Diagrams

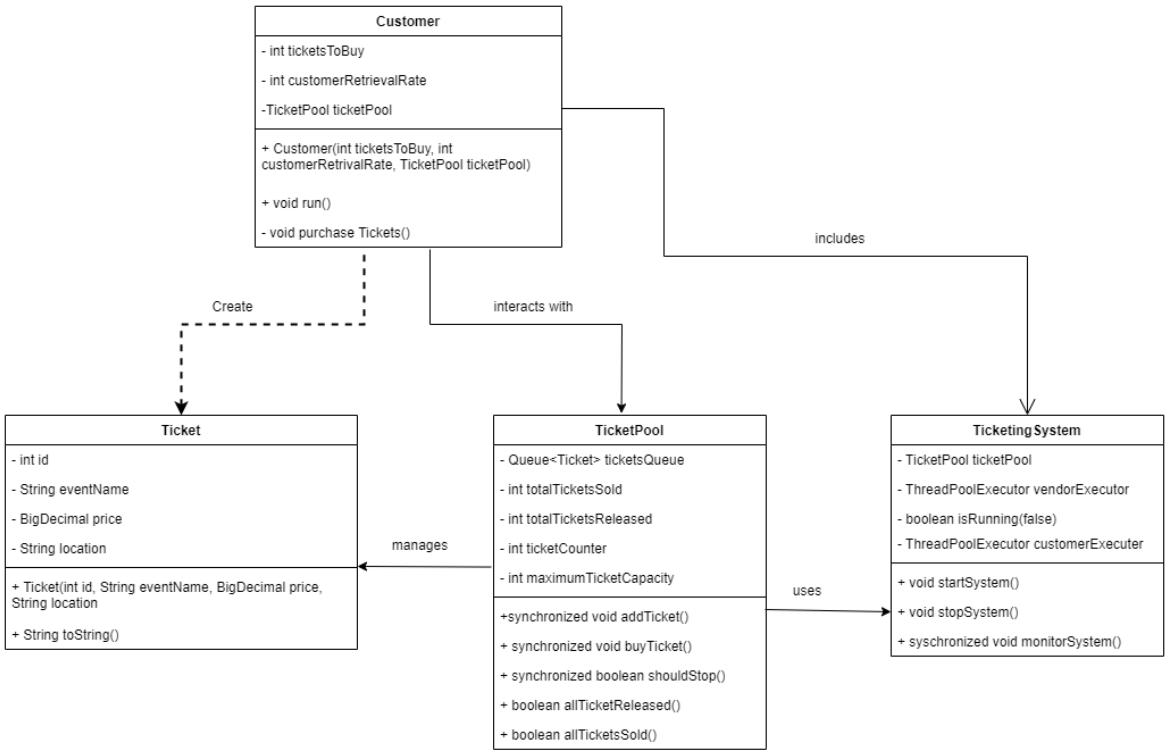
**Ticket Pool Class**



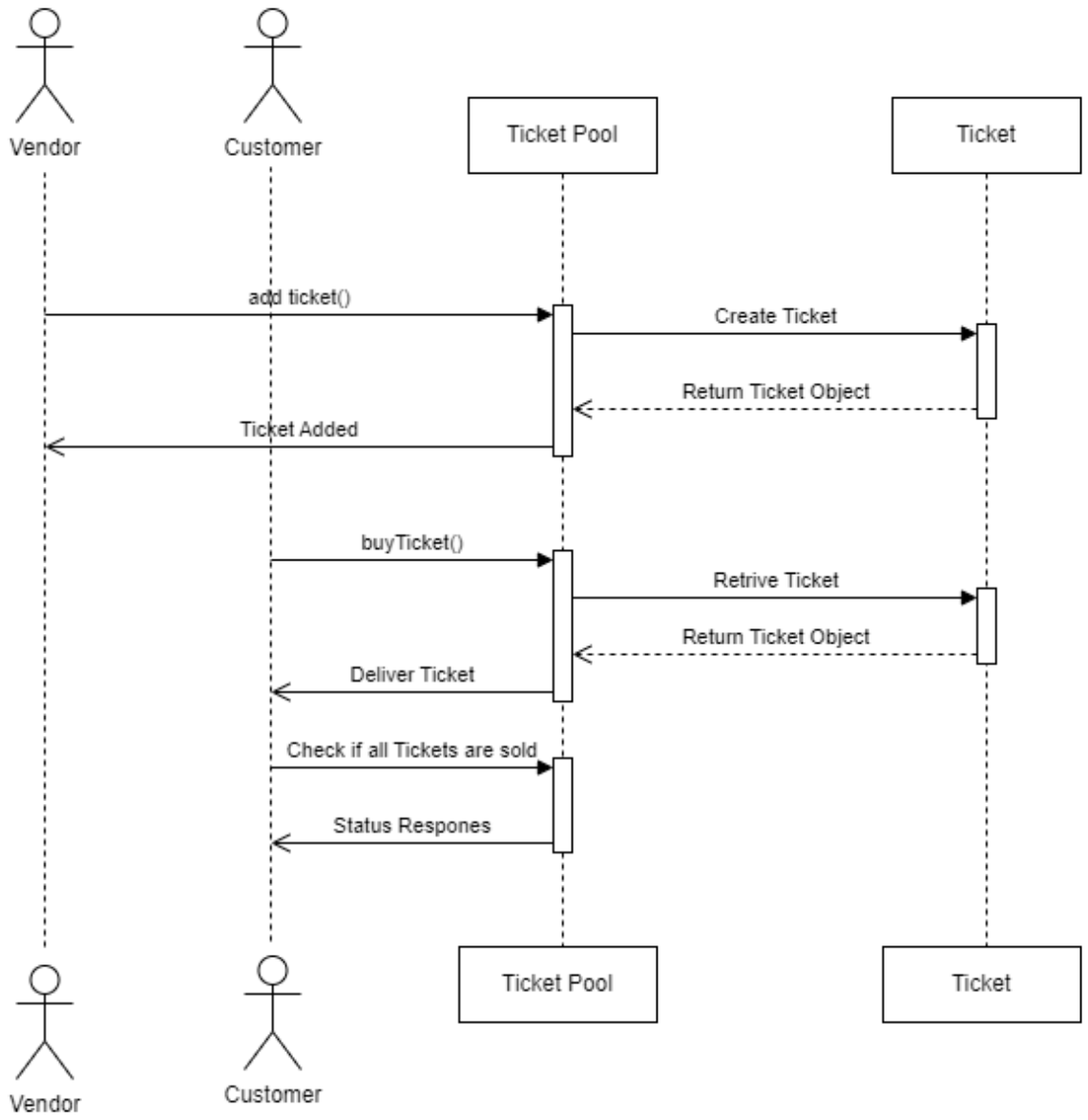
## Vendor Class




## Customer Class



## 2 .2. Sequence Diagram



# 3. Test Cases

## 3.1.CLI Test Cases

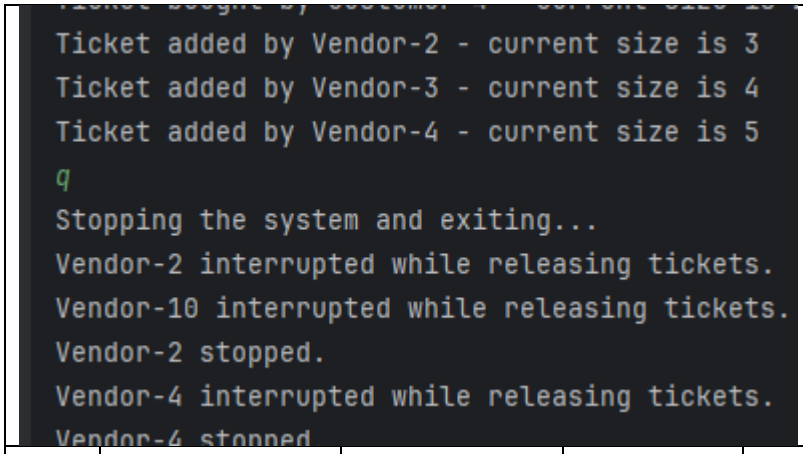
| Test No. | Test case | Self-assessment | Test input | Expected output | Test case status |
|---|---|---|---|---|---|
| 01. | Display the configuration in the console. | Fully implemented. | Run project | -----Configurations-----<br>Enter the total number of tickets: | Pass |
| Actual output<br>"C:\Program Files\Java\jdk-21\bin\java.exe"<br>-----Configurations-----<br>Enter the total number of tickets: | | | | | |
| 02. | Enter Values to the 4 configurations and get the summary | Fully implemented. | 100<br>20<br>10<br>50 | -----Configurations-----<br>Enter the total number of tickets:<br>100<br>Enter the ticket release rate (in seconds):<br>20<br>Enter the customer retrieval rate (in seconds):<br>10<br>Enter the maximum number of tickets:<br>40<br><br>All inputs validated successfully!<br><br>Total Tickets: 100<br>Ticket Release Rate: 20<br>Customer Retrieval Rate: 10<br>Maximum Ticket Capacity: 40<br>Configuration saved as JSON successfully.<br>----------<br>Enter command to start or stop the system (start/q): | Pass |
| Actual Output | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | "C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\<br>-----Configurations-----<br>Enter the total number of tickets:<br>100<br>Enter the ticket release rate (in seconds):<br>20<br>Enter the customer retrieval rate (in seconds):<br>10<br>Enter the maximum number of tickets:<br>40<br><br>All inputs validated successfully!<br><br>Total Tickets: 100<br>Ticket Release Rate: 20<br>Customer Retrieval Rate: 10<br>Maximum Ticket Capacity: 40<br>Configuration saved as JSON successfully.<br>----------<br>Enter command to start or stop the system (start/q): | |
| 03. | Enter the Start command to start the programme. | Fully Implemented | start | Enter command to start or stop the system (start/q): start<br>Starting the ticketing system...<br>----------<br>Ticketing system is now running.<br>----------<br><br>Ticket added by Vendor-1 - current size is 1<br>Ticket added by Vendor-10 - current size is 2<br>Ticket added by Vendor-9 - current size is 3<br>Ticket added by Vendor-8 - current size is 4…… | Pass |

<table>
<tr><td colspan="5">Actual Outcome<br><pre>Enter command to start or stop the system (start/q):
start
Starting the ticketing system...
--------------------------------
Ticketing system is now running.
--------------------------------

Ticket added by Vendor-1 - current size is 1
Ticket added by Vendor-10 - current size is 2
Ticket added by Vendor-9 - current size is 3
Ticket added by Vendor-8 - current size is 4
Ticket bought by Customer-5 - current size is 3 - Ticket is
Ticket added by Vendor-7 - current size is 4</pre></td><td></td></tr>
<tr><td>04.</td><td>Stop the system before starting the simulation</td><td>Fully implemented.</td><td>q</td><td>Enter the command to start or stop the system (start/q):<br>q<br>Stopping the system and exiting...<br>All threads have been stopped.</td><td rowspan="2">Pass</td></tr>
<tr><td colspan="5">Actual Output<br><pre>Enter command to start or stop the system (start/q):
q
Stopping the system and exiting...
All threads have been stopped.

Process finished with exit code 0</pre></td></tr>
<tr><td>05.</td><td>Stop the system middle of the simulation.</td><td>Fully implemented</td><td>q</td><td>Enter command to start or stop the system (start/q):<br>q<br>Stopping the system and exiting...<br>All threads have been stopped.</td><td rowspan="2">pass</td></tr>
<tr><td colspan="5">Actual Output</td></tr>
</table>

| | | | | | |
|---|---|---|---|---|---|
|  | | | | | |
| 06. | The system Terminates after the total tickets are sold. | Fully implemented. | Automatic | Ticket bought by Customer-2 - current size is 0 - Ticket is {ticketID=100, eventName='Simple Event', ticketPrice=1000, location='Maharagama'}<br>Vendor-6 stopped.<br>Vendor-3 stopped.<br>Vendor-5 stopped.<br>Vendor-4 stopped.<br>Vendor-9 stopped.<br>Vendor-7 stopped.<br>Vendor-2 stopped.<br>Vendor-1 stopped.<br>Vendor-8 stopped.<br>Vendor-10 stopped.<br>Customer-5 stopped.<br>Customer-4 stopped.<br>Customer-3 stopped.<br>Customer-1 stopped.<br>Customer-2 stopped. | Pass |
| Actual Output | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Ticket bought by Customer-2 - current size is 0 - Ticket is {ticket}<br>Vendor-6 stopped.<br>Vendor-3 stopped.<br>Vendor-5 stopped.<br>Vendor-4 stopped.<br>Vendor-9 stopped.<br>Vendor-7 stopped.<br>Vendor-2 stopped.<br>Vendor-1 stopped.<br>Vendor-8 stopped.<br>Vendor-10 stopped.<br>Customer-5 stopped.<br>Customer-4 stopped.<br>Customer-3 stopped.<br>Customer-1 stopped.<br>Customer-2 stopped.<br><br>Process finished with exit code 0 | |
| 07. | Validation for configurations. | Fully Implemented | -1<br>10<br>12<br>5<br>6<br>4<br>12<br>8 | -----Configurations-----<br>Enter the total number of tickets:<br>-1<br>Total tickets must be a positive integer!<br>Enter the total number of tickets:<br>10<br>Enter the ticket release rate (in seconds):<br>12<br>The ticket release rate cannot exceed the total available tickets!<br>Enter the ticket release rate (in seconds):<br>5<br>Enter the customer retrieval rate (in seconds):<br>6<br>The customer retrieval rate cannot exceed the ticket release rate!<br>Enter the customer retrieval rate (in seconds):<br>4 | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Enter the maximum number of tickets:<br>12<br>The maximum ticket capacity cannot exceed the total tickets and must be greater than or equal to the ticket release rate!<br>Enter the maximum number of tickets:<br>8<br><br>All inputs validated successfully! | |

Actual Output

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\lib\i
-----Configurations-----
Enter the total number of tickets:
-1
Total tickets must be a positive integer!
Enter the total number of tickets:
10
Enter the ticket release rate (in seconds):
12
The ticket release rate cannot exceed the total available tickets!
Enter the ticket release rate (in seconds):
5
Enter the customer retrieval rate (in seconds):
6
The customer retrieval rate cannot exceed the ticket release rate!
Enter the customer retrieval rate (in seconds):
4
Enter the maximum number of tickets:
12
The maximum ticket capacity cannot exceed the total tickets and must be greater than or equal to the ticket rel
Enter the maximum number of tickets:
8

All inputs validated successfully!
```

| | | | | | |
|---|---|---|---|---|---|
| 08. | Validate the integers | Full implemented | q<br>10<br>e<br>5<br>w<br>4<br>x<br>8 | -----Configurations-----<br>Enter the total number of tickets:<br>q<br>Invalid input! Please enter a positive integer.<br>Enter the total number of tickets:<br>10<br>Enter the ticket release rate (in seconds):<br>e<br>Invalid input! Please enter a positive integer.<br>Enter the ticket release rate (in seconds):<br>5 | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Enter the customer retrieval rate (in seconds):<br>w<br>Invalid input! Please enter a positive integer.<br>Enter the customer retrieval rate (in seconds):<br>4<br>Enter the maximum number of tickets:<br>x<br>Invalid input! Please enter a positive integer.<br>Enter the maximum number of tickets:<br>8<br><br>All inputs validated successfully! | |

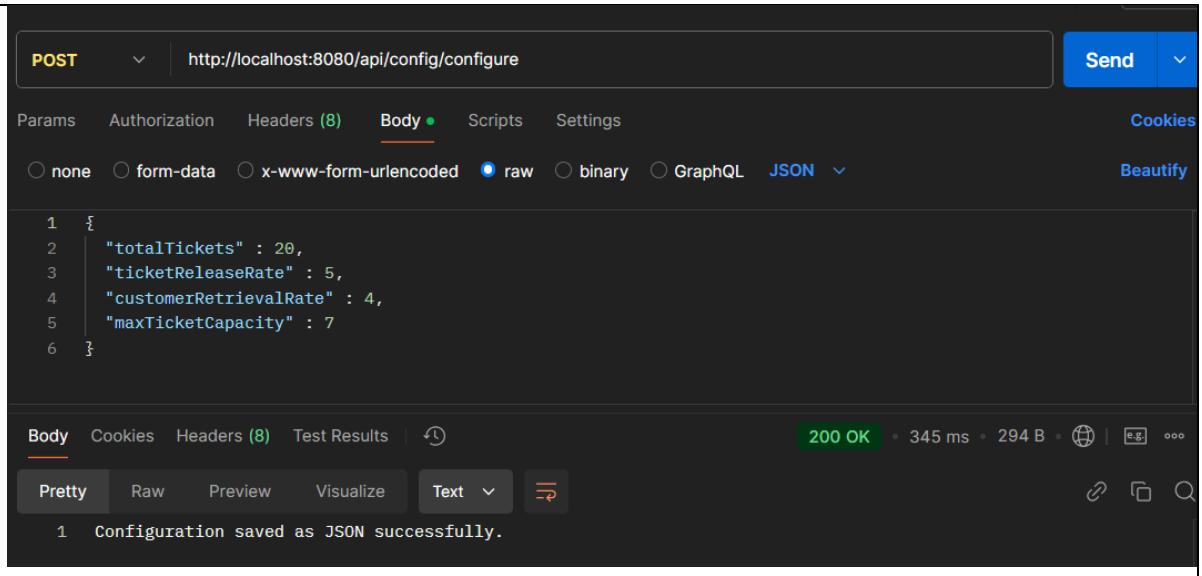
Actual Output

```
-----Configurations-----
Enter the total number of tickets:
q
Invalid input! Please enter a positive integer.
Enter the total number of tickets:
10
Enter the ticket release rate (in seconds):
e
Invalid input! Please enter a positive integer.
Enter the ticket release rate (in seconds):
5
Enter the customer retrieval rate (in seconds):
w
Invalid input! Please enter a positive integer.
Enter the customer retrieval rate (in seconds):
4
Enter the maximum number of tickets:
x
Invalid input! Please enter a positive integer.
Enter the maximum number of tickets:
8

All inputs validated successfully!
```

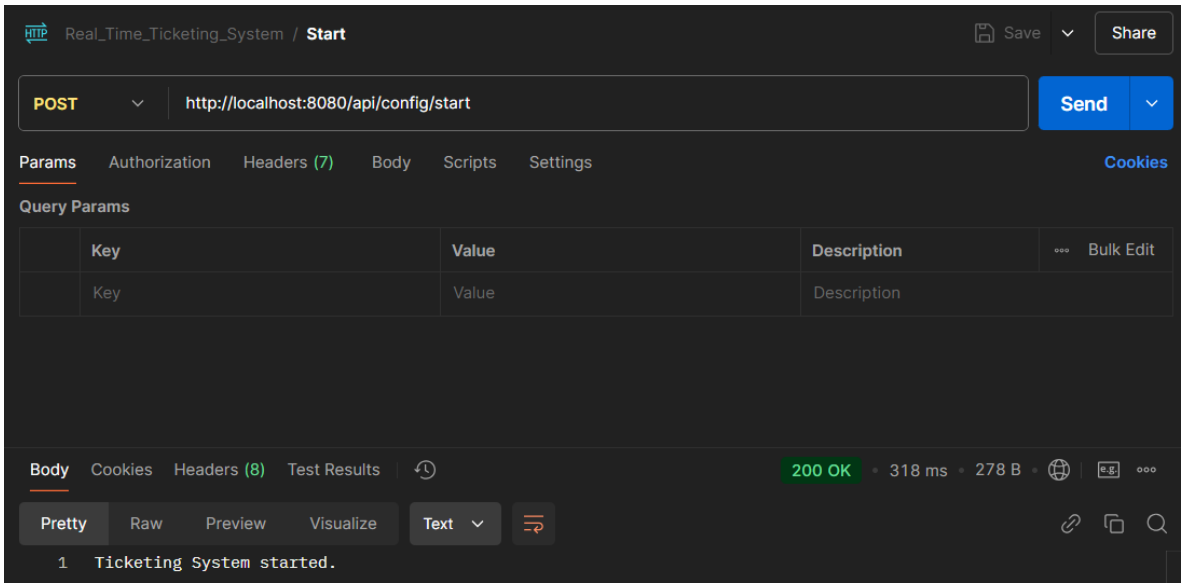
## 3.2. Backend Test Cases

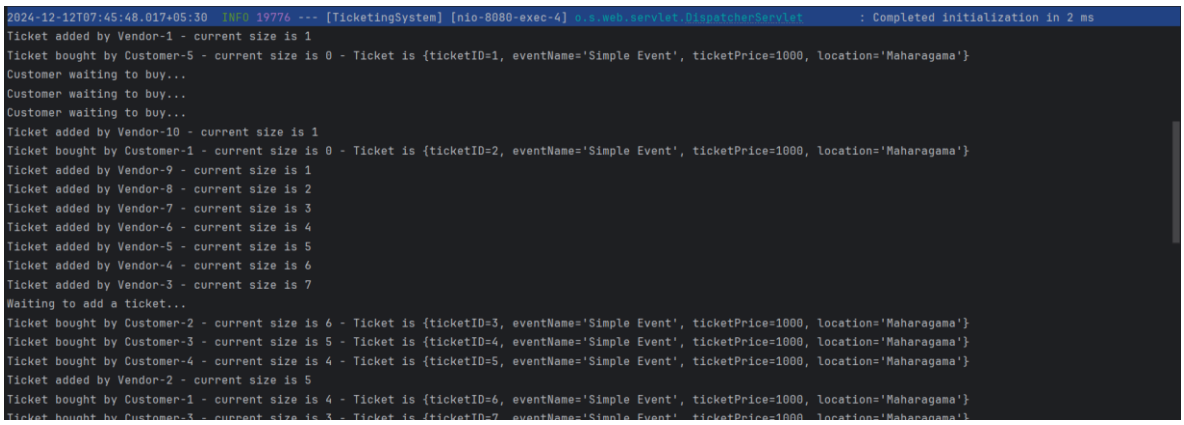
1. Send the API request from the postman to the backend

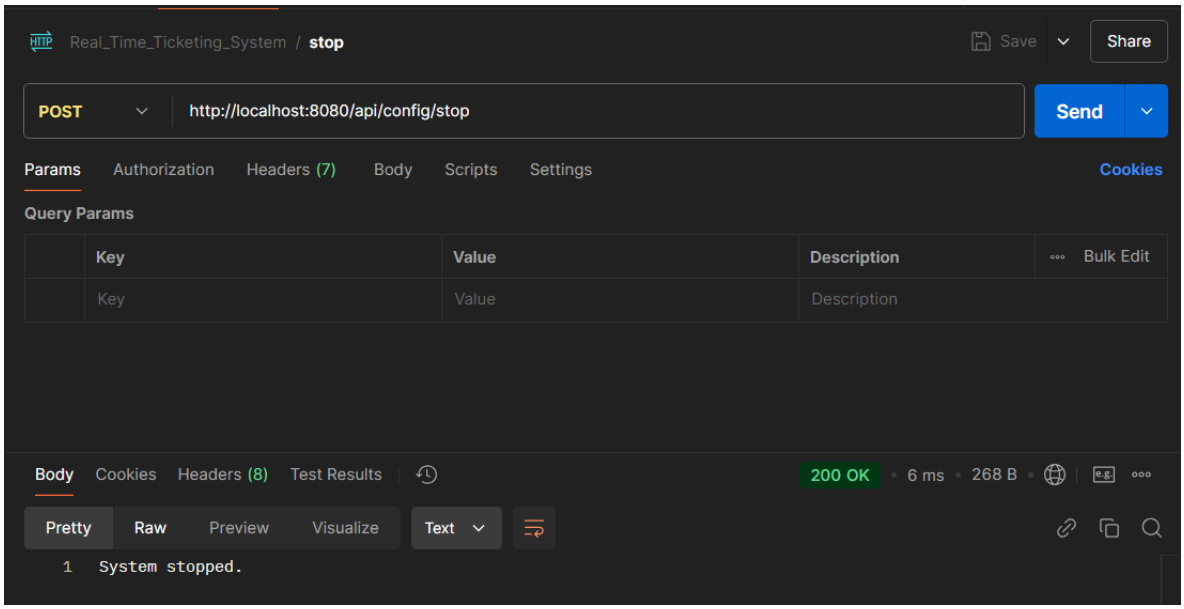
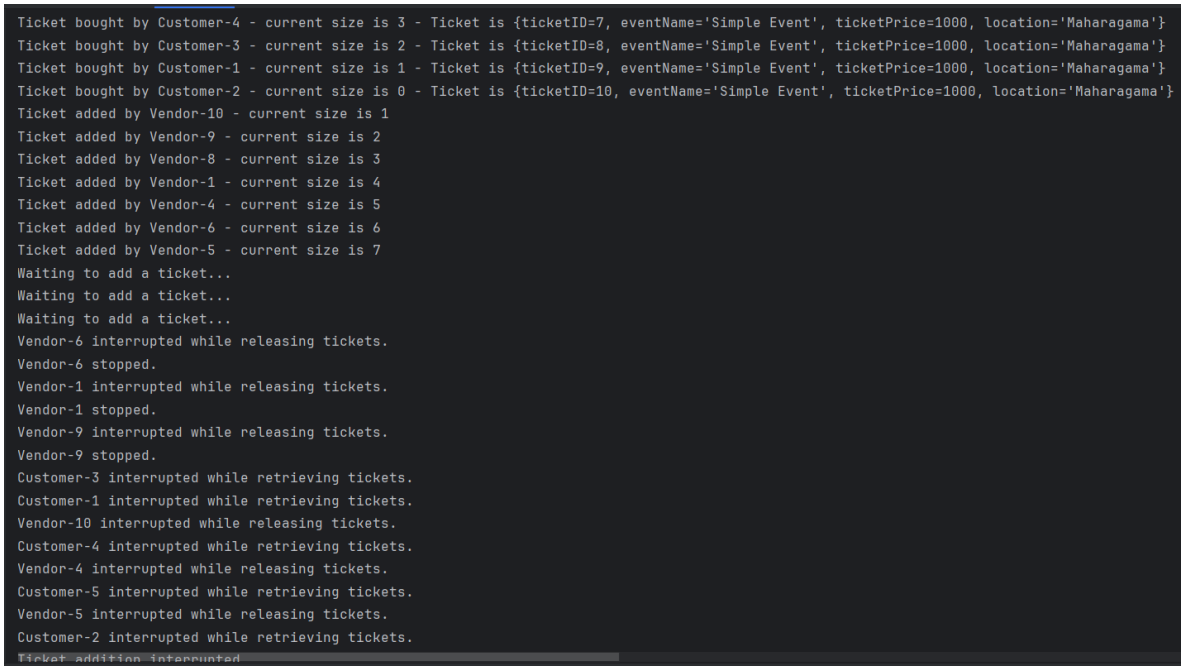



2024-12-12T07:45:48.017+05:30 INFO 19776 --- [TicketingSystem] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms

2. Start the simulation through the Postman API

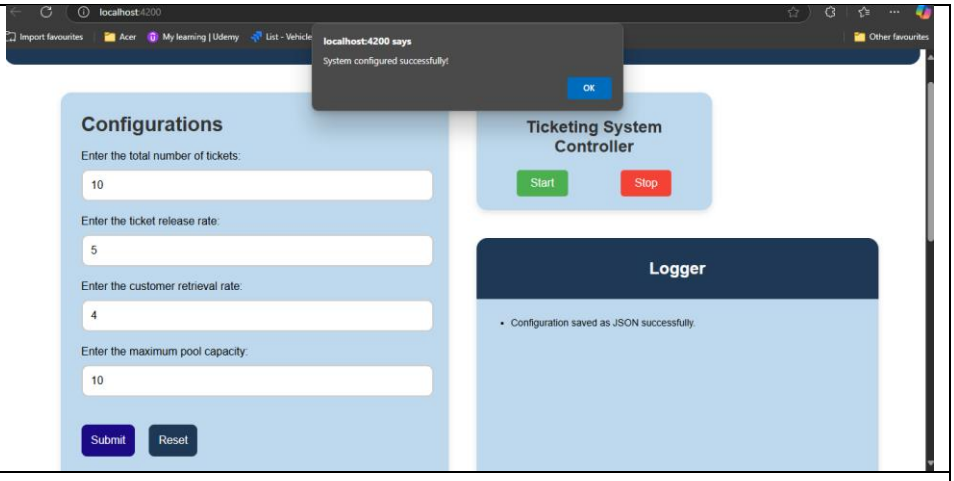
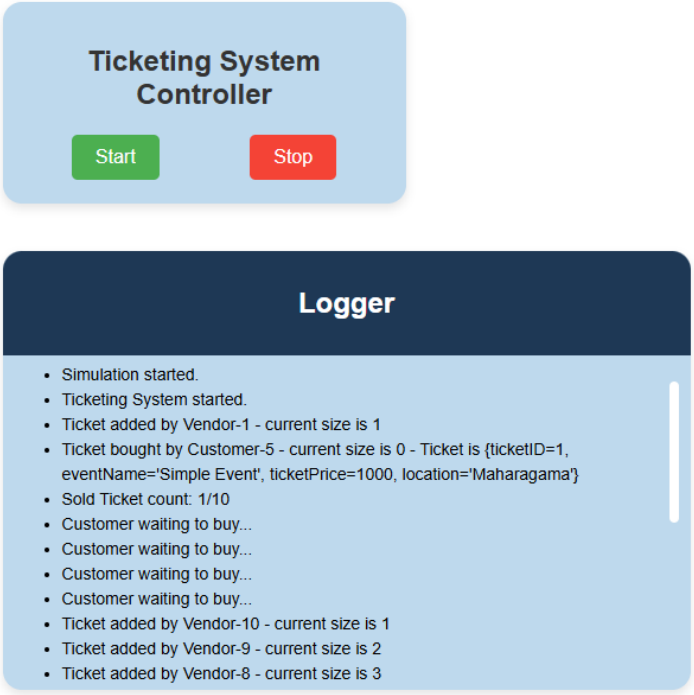


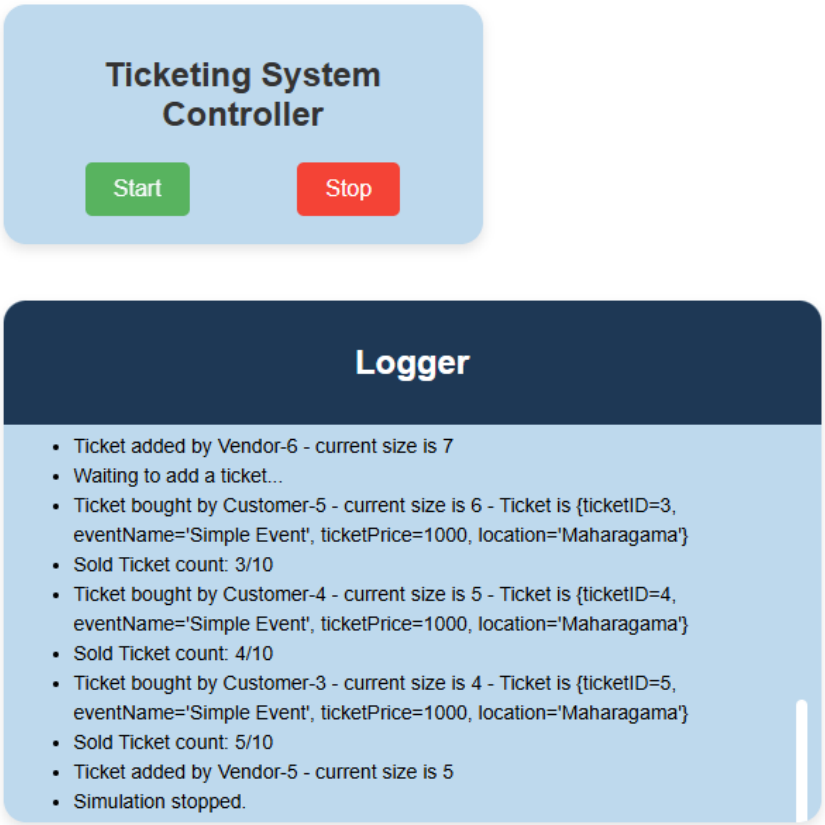



| | |
|---|---|
| 4. Stop the simulation using the Postman API request |  <br>  |

## 3.3. GUI Test Cases.

| | |
|---|---|
| 1.Save the configuration Successfully. | |

| | |
|---|---|
| |  |
| 2.Start Simulation |  |
| 3.Stop the simulation. | |

<table>
<tr>
<td></td>
<td>


</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</table>