UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

University of Westminster

Department of Computing

BSc (Hons) Computer Science

Module Name: Object Oriented Programming

Module Code: 5COSC019C

Full Name: Janudi Yatagampitiya

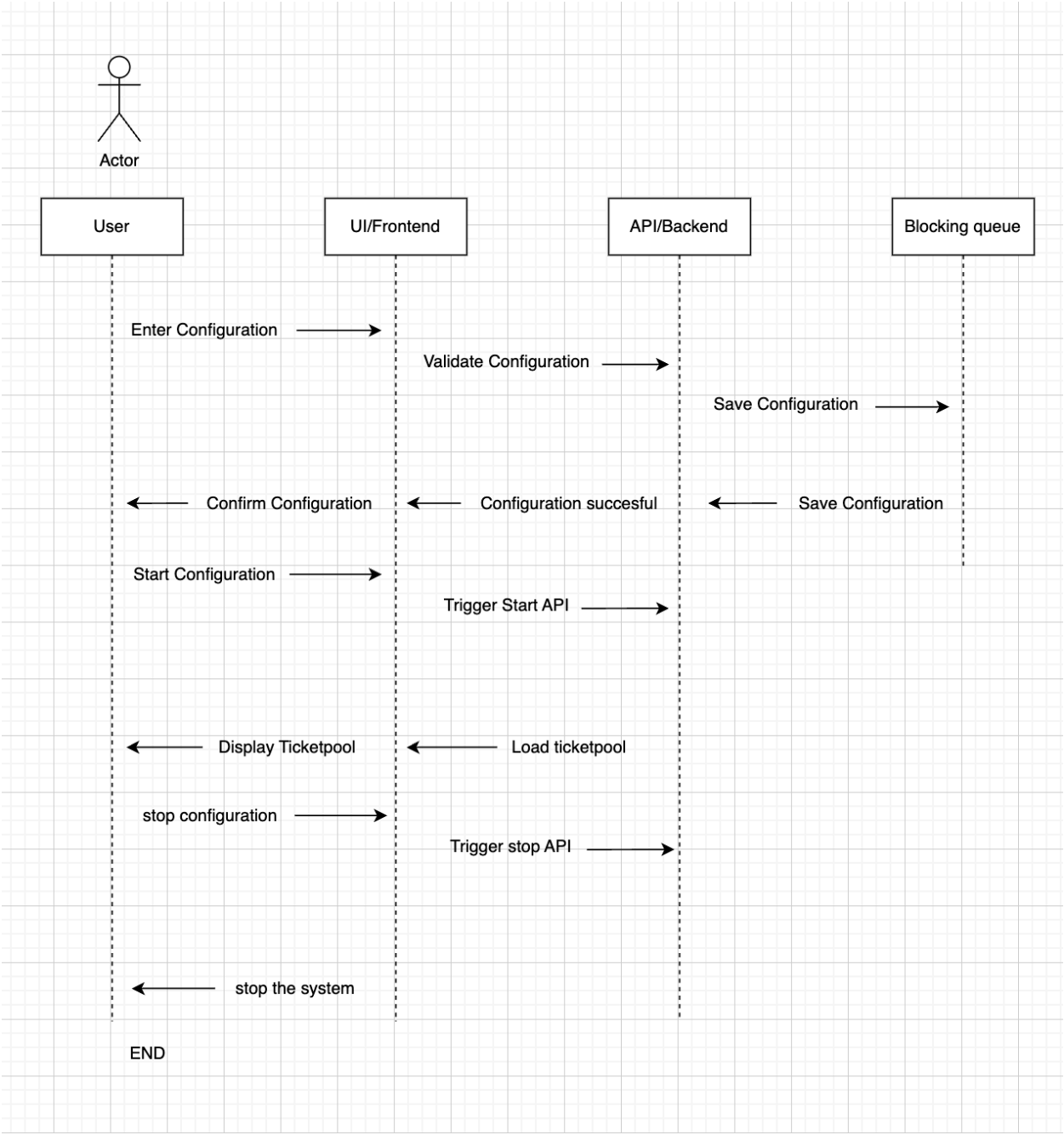
UOW ID: w20518932

IIT ID: 20231201

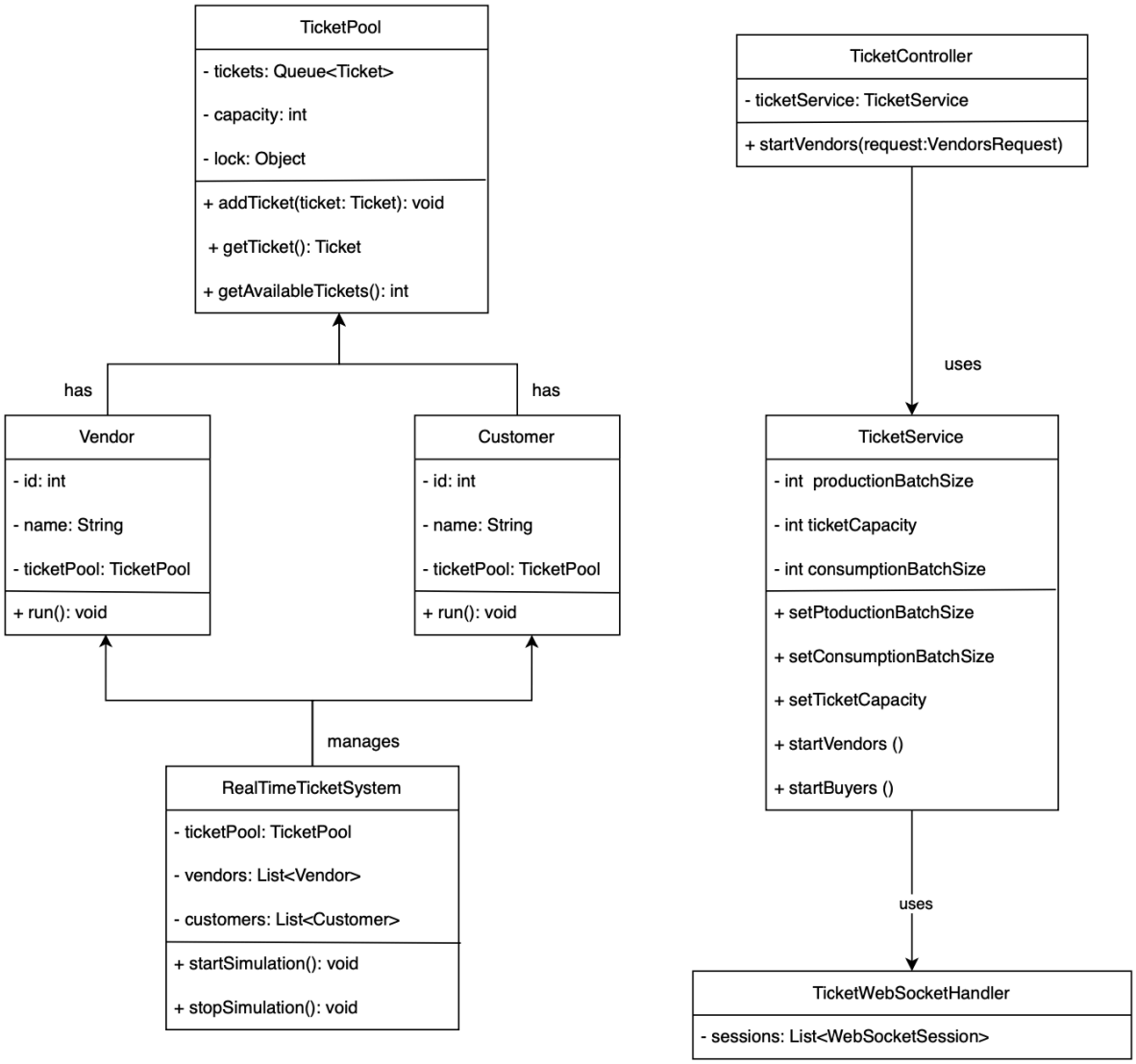
Contents

1. Sequence diagram 3
2. Class diagram 4
3. Test Cases 5
4. CLI 6
5. Frontend 7
6. Backend 8

1. Sequence diagram



2. Class diagram

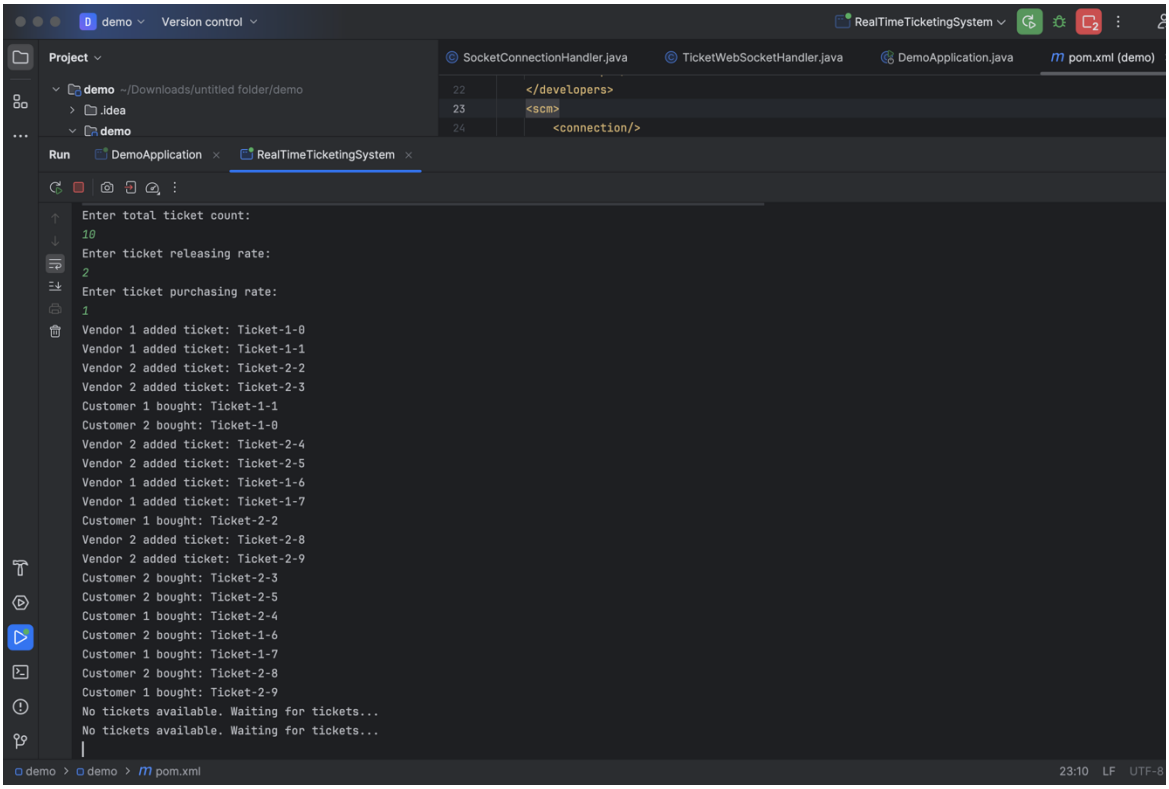


3. Test Cases

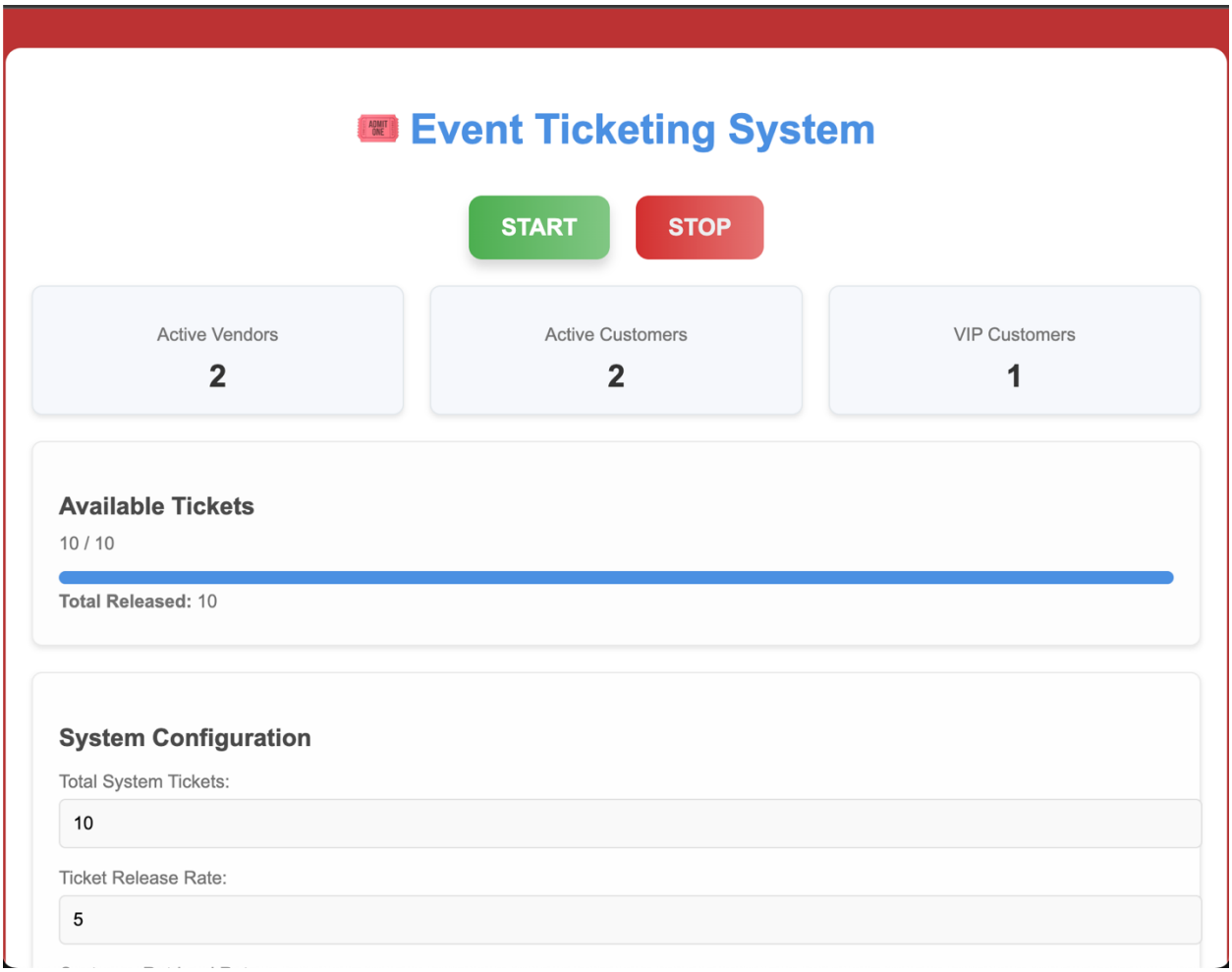
Test case scenario	Scenario description	Result	Status
Display a menu for start and stop commands	Enter the correct option to start the system	Program started	Pass
	Enter the correct option to exit the system	Exit the system	
Start the system	Enter all valid inputs		Pass
	Enter invalid inputs		
System configuration	Test manual configuration using user inputs		Pass
	Configuration using previous data		
Ticket releasing process	Ticket releasing process stops when total tickets reach the maximum ticket capacity		Pass

Ticket retrieval process	Customer retrieval rate is higher than vendor release rate		Pass
	Customer retrieval rate is equal to Vendor release rate		
	Customer retrieval rate is less than vendor release rate		
Minimal configuration			
Configuration input validation	Entering invalid characters		Pass
Save config data to JSON file			Pass

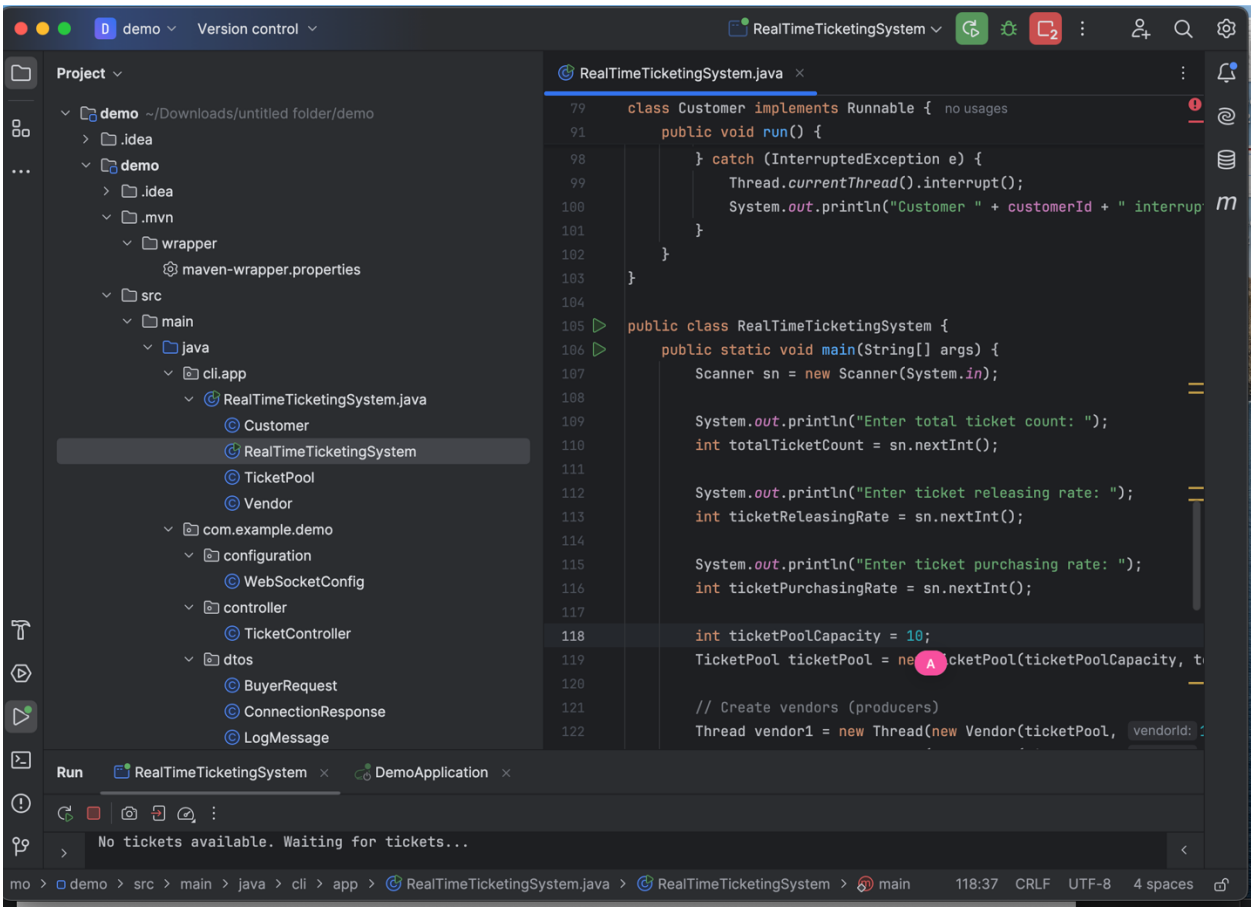
4. CLI



5. Frontend


Event Ticketing System
START
STOP
Active Vendors
2
Active Customers
2
VIP Customers
1
Available Tickets
10 / 10
Total Released: 10
System Configuration
Total System Tickets:
10
Ticket Release Rate:
5

6. Backend


demo
Version control
RealTimeTicketingSystem
Project
demo ~/Downloads/untitled folder/demo
.idea
demo
.idea
.mvn
wrapper
maven-wrapper.properties
src
main
java
cli.app
RealTimeTicketingSystem.java
Customer
RealTimeTicketingSystem
TicketPool
Vendor
com.example.demo
configuration
WebSocketConfig
controller
TicketController
dtos
BuyerRequest
ConnectionResponse
LogMessage
RealTimeTicketingSystem.java
class Customer implements Runnable { no usages
public void run() {
} catch (InterruptedException e) {
Thread.currentThread().interrupt();
System.out.println("Customer " + customerId + " interrup
}
}
}
public class RealTimeTicketingSystem {
public static void main(String[] args) {
Scanner sn = new Scanner(System.in);
System.out.println("Enter total ticket count: ");
int totalTicketCount = sn.nextInt();
System.out.println("Enter ticket releasing rate: ");
int ticketReleasingRate = sn.nextInt();
System.out.println("Enter ticket purchasing rate: ");
int ticketPurchasingRate = sn.nextInt();
int ticketPoolCapacity = 10;
TicketPool ticketPool = ne cketPool(ticketPoolCapacity, t
// Create vendors (producers)
Thread vendor1 = new Thread(new Vendor(ticketPool, vendorId:
Run
RealTimeTicketingSystem
DemoApplication
No tickets available. Waiting for tickets...
mo > demo > src > main > java > cli > app > RealTimeTicketingSystem.java > RealTimeTicketingSystem > main
118:37
CRLF
UTF-8
4 spaces