University of Westminster

Informatics Institute of Technology

Object Oriented Programming

5COSC019C.1

1.A Coursework

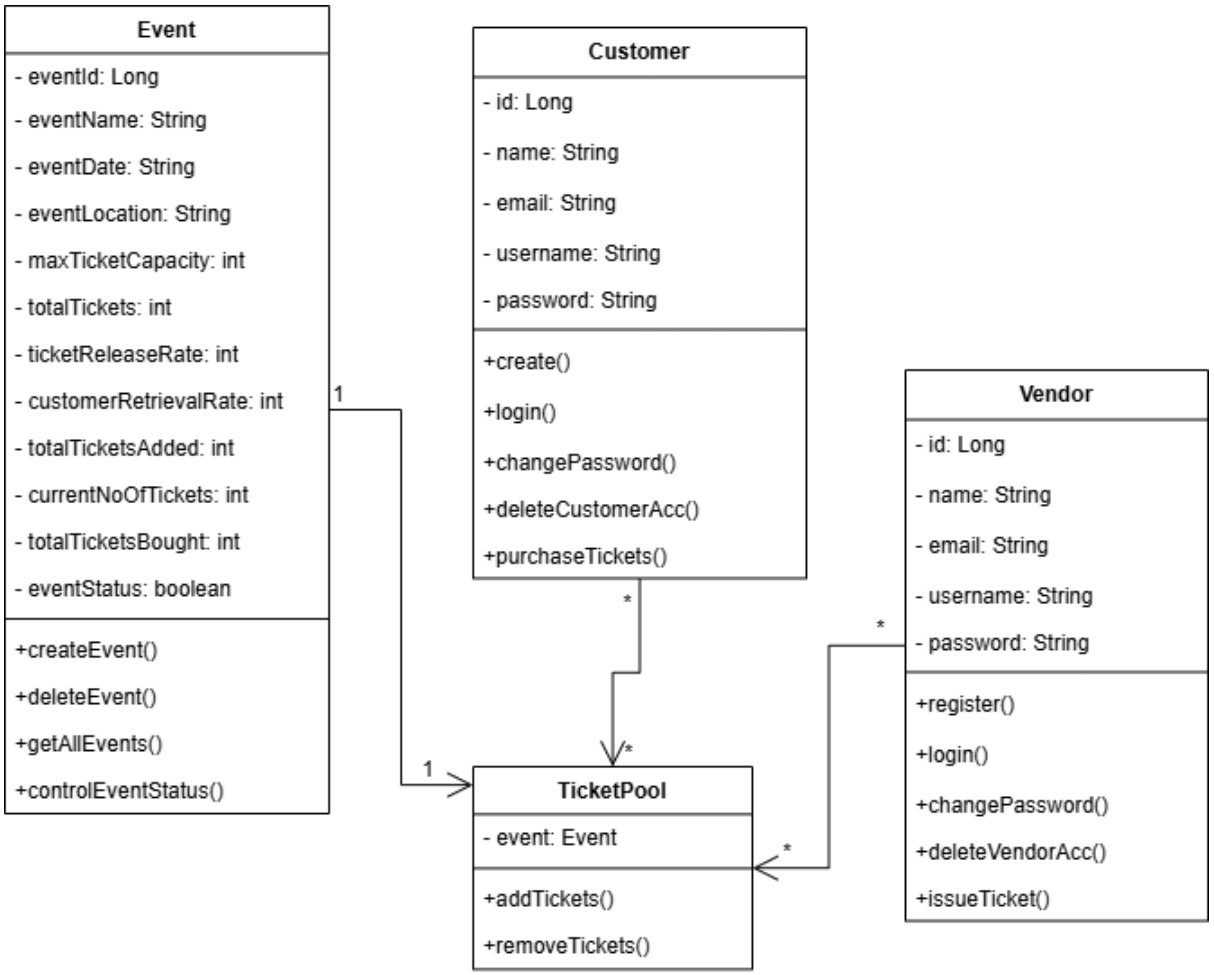
Meepe Gamage Binuk Yehan Dias

UoW Student ID - 20527837

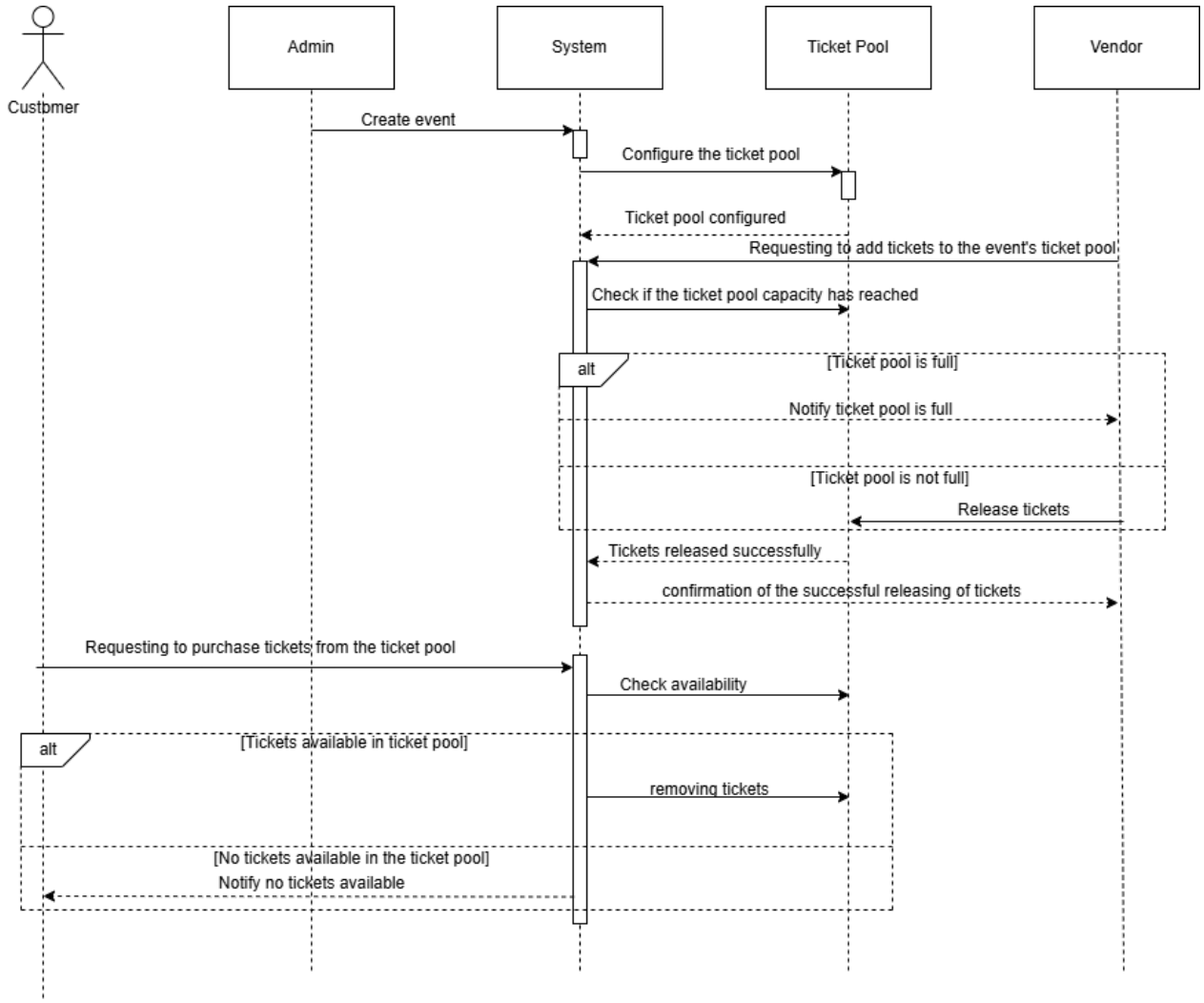
IIT Number - 20221411

Table of Contents

Class Diagram



Sequence Diagram



Test Cases

Test Case ID	Scenario	Test Steps	Expected Result	Outcome
GUI				
1	Handling ticket purchases by multiple customers	1. Configure the customer retrieval rate 2. Send multiple customer ticket purchase requests concurrently. 3. Monitor the ticket availability and logs	There should not be any race conditions, after the purchases of all requests are done, the available ticket count should be correct. (Not oversold or undersold)	Pass
2	Handling the adding of tickets by multiple vendors	1. Configure the vendor ticket release rate 2. Send multiple vendor tickets adding requests concurrently. 3. Monitor the ticket availability and logs.	There should not be any race conditions, after all the ticket-adding requests are done, the remaining ticket count should be correct.	Pass
3	Reaching the maximum capacity of the ticket pool. (Total tickets)	1. Send vendor tickets adding requests until it reach the ticket pool limit ("Total tickets"). 2. Monitor the system's response when it reaches that limit.	The adding of tickets should be stopped, and a message should be sent to the vendor saying the limit has been reached, vendors should not be able to add any more tickets till a customer buys a ticket.	Pass
4	Reaching the maximum capacity of the event. (Max ticket capacity)	1. Send vendor and customer requests until the total tickets released by vendors reach the "max ticket capacity". 2. Monitor the system's response when it reaches that limit.	The adding of tickets should be stopped, even when customers buy the remaining tickets in the ticket pool, the vendors should not be able to add tickets to that event ticket pool.	Pass
5	Real-time update of the available ticket count of each event.	1. Send several vendor and customer requests. 2. Monitor if the displayed available tickets count updates in real-time without refreshing for each vendor/customer.	When vendors and customers add or purchase tickets from the displayed events, the available ticket count should change in real time.	Pass

6	Start / Stop of a specific event.	1. Admin selects the event to start and gives the start command. 2. Send vendor and Customer requests to check if they can add/purchase. 3. Admin selects the event to stop and gives the stop command. 4. Send vendor and customer requests to check if they cannot add/purchase from that stopped event	When an event is not started, the vendors and customers should not be able to add or purchase tickets from that event, when the event is started by the admin, the vendors and customers must be able to add or purchase tickets from that event.	Pass
7	Real-time update of the start/stop status in the events list.	1. Admin starts a previously stopped event. 2. Admin stops an event that has been started before.	The Vendors and Customers should see the red indicator when the event is stopped, and the green indicator when the event is started.	Pass
8	Input Validation	1. Insert a negative number when creating an event, when adding tickets, and when purchasing tickets. 2. Monitor the system's response	The system should display a customized error message to the user when they do an invalid input	Pass
9	Thread synchronization	1. Send a vendor adding ticket requests. 2. Send a customer ticket purchase request at the same time. 3. Monitor the logs.	There should not be any race conditions or deadlocks. The vendors and customers should be able to buy and purchase tickets at the same time. The available tickets in the ticket pool should be accurate.	Pass
10	Error handling	1. Simulate a database failure. 2. Send a ticket purchase request/ ticket adding request 3. Monitor the system's response.	The system should not crash, it should display a customized error message	Pass
11	Vendor and Customer access control.	1. Visit the vendor login page and try to log in with an invalid username 2. Monitor the system's response.	The vendors and customers who have not registered should not be able to log in, and the vendors and customers who have previously	Pass

		3. Now login with a valid username and password. 4. Monitor the system's response. 5. Do the same process on the customer login page.	registered should be able to log in smoothly and go to the tickets adding / purchasing page.	
CLI				
12	Starting the ticketing process	1. Enter the start command	Vendor and customer threads should be started and the ticketing process should be running.	Pass
13	Trying to start when the process is already started.	1. Enter "Start" when the process is already started.	A message should be displayed saying the ticketing process is already running.	Pass
14	Stopping the ticket process	1. Enter "stop" to stop the ticketing process.	The ticketing process should stop and a message should be displayed saying the system has stopped.	Pass
15	Resetting the ticketing system after stopping	1. Enter "Stop" to stop the ticketing process. 2. Enter "yes" when asked to configure again.	Should ask the user to configure again.	Pass
16	Exiting the system	1. Enter "Stop" to stop the ticketing process 2. Enter "No" when asked to configure again.	The system should exit with a message displayed.	Pass
17	Input validation	1. Enter negative values or enter an invalid input.	A customized error message should be shown and should ask the user to enter the value again.	Pass
18	Verifying that several customer threads run at the same time.	1. In configuration, enter the number of customers. 2. Then start the system. 3. Monitor the logs	The logs should display the customer's number when logging as "Customer 1", "Customer 2"	Pass
19	Stopping the process when the total tickets added by the vendors reach the max ticket capacity.	1. Configure the system. 2. Start the ticketing process. 3. Monitor the system logs.	The process should automatically stop when the total number of tickets added by vendor reaches the maximum ticket capacity as configured before.	Pass
20	Checking for proper thread synchronization	1. Configure the system. 2. Start the ticketing process. 2. Monitor the system logs	There should be no race conditions, vendor and customers should be adding and purchasing tickets concurrently.	Pass