UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Module Leader - Mr. Guhanathan Poravi

Coursework Report – 5COSC019C Object Oriented Programming

Student Name: Hewathanthirige Sathmi Minanga Hewathanthirige
Student ID: 2030073
UOW No: w2052077

Demo Video UrL - https://drive.google.com/file/d/14o8jxEWyDNxzc2_Afxwk2H2jMisceKRF/view?usp=drive_link

TEST PLAN

Test ID	Test Case	Expected Result	Actual Result	Pass / Fail
1.Select the type of the user.	A Situation where a user wants access relevant portal.	To display successfully directly to the relevant portal.	Display successfully directly to the relevant portal	Pass
2.Add a customer by entering name to the system.	A Situation where customers want to enter the system	To display an alert “Customer account created successfully.” And navigate to the customer dashboard.	Display an alert “Customer account created successfully.” And navigate to the customer dashboard.	Pass
3. Book a Ticket for the event.	A Situation where customers want to purchase tickets.	When clicking the Book a ticket button successfully direct to the Purchase ticket page.	When click the Book a ticket button successfully direct to the Purchase ticket page.	Pass
4. Show the number of Tickets available for the customer.	Display all real time available tickets.	In the Ticket booking page displays the current available tickets amount.	In the Ticket booking page displays the current available tickets amount.	Pass
5. Request for a ticket.	A Situation where the user wants to purchase tickets by entering their name, number of tickets and	To display a customer name, number of tickets and VIP true or false in a sweet alert.	Display a customer name, number of tickets and VIP true or false in a sweet alert.	Pass

	selecting VIP or not.			
6. Add a vendor by entering name to the system.	A Situation where Vendor wants to enter the system	To display an alert “Vendor account created successfully.” And navigate to the vendor dashboard.	Display an alert “Vendor account created successfully.” And navigate to the vendor dashboard.	Pass
7. Start system.	A Situation where the vendor wants to start the system and display the ticket pool size.	To display the ticket pool size when clicking the Start System button.	Display the ticket pool size when clicking the Start System button.	Pass
8. Stop system.	A Situation where the vendor wants to stop the running system.	To display system running as stopped when clicked the Stop System button.	Display system running stopped when clicked the Stop System button.	Pass
9. Update configuration.	A situation where vendor can see rates and the total number of tickets.	To display all the rate and the total number of tikcets.	Display all the rate and the total number of tikcets.	Pass
10. Check status.	A Situation where the vendor can get a visual representation of purchasing tickets.	To display graphical visual representation of ticket purchases with the time.	Display graphical visual representation of ticket purchases with the time.	Pass

[✓ Yes] No

Task	**Did you attempt the task?**	**Student's comments**
Design a UML Use Case Diagram and Sequential Diagram	[✓ Yes] No	Class diagrams were created for all the classes with the relationships between the classes.
Implementation Class TicketingSystem	[✓ Yes] No	Class TicketingSystem created with some methods like getValidatInput, saveConfigurationToFile , getTicketConfiguration to get inputs for rates.
Implementation Class Customer	[✓ Yes] No	Customer class was successfully created with requirements.
Implementation Class Vendor	[✓ Yes] No	Class Vendor was created successfully created with all requirements.
Implementation Class TicketPool	[✓ Yes] No	TicketPool class implements synchronization in its methods to ensure thread safety.
Implementation Class TciketingConfiguartion	[✓ Yes] No	Class TciketingConfiguartion was created successfully with all the attributes and getters and setters.
Implementation Class Ticket	[✓ Yes] No	Ticket class was successfully created with requirements.
Users can select a path depend on their needs.	[✓ Yes] No	Task was implemented successfully to direct to the dashboards.
Customers can purchase tickets and cancel tickets	[✓ Yes] No	The task was implemented successfully.

and visible the available number of tickets.		
Vendors can Start System, Stop System, Check Status and update configurations.	[✓ Yes] ☐ No	The task was implemented successfully.
Error handling across all the code, input validation and code quality.	[✓ Yes] ☐ No	Task was completed with error handling and try/catch blocks to improve the code quality and check input validation.