|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number ID | Related Systems/ User Story | Pre-Conditions | Test Description (Steps) | Expected Results | R2 Results |
| 1 | 1. Delivery Location Database | System is Running | 1. Creates a DeliveryLocation with the name, "RIT", and delivery time 26 minutes | 1. The DeliveryLocationStorage should have a DeliveryLocation in its map, the associated string called RIT and the DeliveryLocation having the aforementioned traits | Pass |
| 2 | 1. Delivery Location Database | System is Running | 1. Attempts to create a DeliveryLocation with either a null name, a name that already exists in the system, or a negative time | 1. The DeliveryLocationStorage should not store the location. | Pass |
| 3 | 1. Delivery Location Database | #1 | 1. Ask the DeliveryLocation database for a location of name, "RIT" | 1. Returns the object created in test number 1. | Pass |
| 4 | 1. Delivery Location Database | System is Running | 1. Attempts to get a location that does not exist in the database | 1. Returns a null object | Pass |
| 5 | 1.  Delivery Location Database | #1 | 1. Asks the Delivery Location database if a location with name "RIT" exists | 1. Returns true | Pass |
| 6 | 1. Delivery Location Database | System is Running | 1. Asks if the Delivery Location database if it contains a location that does not exist in the database | 1. Returns false | Pass |
| 7 | 1. Delivery Location Database | #1 | 1. Attempts to remove a location with name, "RIT" | 1. The DeliveryLocation database should be empty | Pass |
| 8 | 1. Delivery Location Database | #1 | 1. Saves the DeliveryLocation database settings | 1. A file should exist that is called DeliveryLocationList.settings | Pass |
| 9 | 1. Delivery Location Database | #8  The system is restarted | 1. Gets the instance of the DeliveryLocation database | 1. The instance of the RIT DeliveryLocation should still exist | Pass |
| 10 | 1. Delivery System | System is Running | 1. Set the number of drivers in the Delivery System to 2 | 1. The number of drivers should be 2 | Pass |
| 11 | 1. Delivery System | System is Running | 1. Attempts to set the number of drivers in the Delivery System to a negative number | 1. The number of drivers should be whatever it last was | Pass |
| 12 | 1. Delivery System | #11 | 1. Save the DeliverySystem settings | 1. A file should exist that is called DeliverySystem.settings | Pass |
| 13 | 1. Delivery System | #12  The system is restarted | 1. Get the Delivery Location instance | 1. The number of drivers should be 2. | Pass |
| 14. | 1. Employee Database | System is running | 1. Create an employee with first name, "Employee,", last name "Test", user ID "Case", password "Password", and manager status false | 1. The Employee database should contain an employee with the aforementioned characteristics | Pass |
| 15 | 1. Employee Database | System is Running | 1. Attempts to create a DeliveryLocation with either a null string as any of its constructor parameters, or a user ID that already exists | 1. The EmployeeStorage should not store the employee | Pass |
| 16 | 1. Employee Storage | #14 | 1. Ask the Employee database for a user ID of name "Case". | 1. Returns the object created in test number 15. | Pass |
| 17 | 1. Employee Storage | System is Running | 1. Attempts to get an employee user ID that does not exist in the database | 1. Returns a null object | Pass |
| 18 | 1. Employee Storage | #14 | 1. Asks the Employee database if there is an employee with user ID, "Case" | 1. Returns true | Pass |
| 19 | 1. Employee Storage | System is Running | 1. Asks the Employee database if there an employee ID that does not exist in the database | 1. Returns false | Pass |
| 20 | 1. Employee Storage | #14 | 1. Attempts to remove an employee with user ID "Case" | 1. The Employee database should be empty | Pass |
| 21 | 1. Employee Storage | #14 | 1. Save the Employee database | 1. A file should exist called EmployeeList.settings | Pass |
| 22 | 1. Employee Storage | #21  The system is restarted | 1. Gets the Employee database instance | 1. The database should contain the Employee object created in #15 | Pass |
| 23 | 1. Employee Storage | #14 | 1. Change the first name, last name, and password of the employee with user ID "Case" to "Testing", as well as grant the employee manager status. | 1. The employee object should reflect all the changes made | Pass |
| 24 | 1. Menu Item Storage | System is running | 1. Add a topping by the name, "Bacon" to the Menu Item database | 1. The Menu Item database instance should contain a topping named Bacon | Pass |
| 25 | 1. Menu Item Storage | #24 | 1. Adds a new MenuItem that is called Pizza, has a base price of $10, preparation time of 5 minutes, cook time of 10 minutes, oven space required of 4 units, whole topping cost of $1.50, flag that it can have half toppings, and say it can have Bacon as a topping | 1. The Menu Item database should have a menu item with the given qualities.  2. The Menu Item database should have an association between "Pizza" and "Bacon". | Pass |
| 26 | 1. Menu Item Storage | System is running | 1. Attempts to add a new menu item with either a null name or a name that already exists, or negative-value parameters | 1. The object is not added to the Menu Item Storage | Pass |
| 27 | 1. Menu Item Storage | #25 | 1. Add a topping by the name, "Pepperoni" to the Menu Item database  2. Add Pepperoni as an associating topping to the Pizza MenuItem | 1. The Menu Item database should have a new topping available, Pepperoni.  2. Pizza should also have Pepperoni as an associating topping | Pass |
| 28 | 1. Menu Item Storage | #25 | 1. Removes the Pizza Menu Item from the database | 1. The Menu Item data is removed, however, associating toppings stay | Pass |
| 29 | 1. Menu Item Storage | #27 | 1. Remove Pepperoni as a topping | 1. Pepperoni is disassociated as a topping from Pizza.  2. Pepperoni does not exist as a topping anymore | Pass |
| 30 | 1. Menu Item Storage | #29 | 1. Asks the Menu Item database if Pizza is a menu item | 1. Returns true | Pass |
| 31 | 1. Menu Item Storage | System is Running | 1. Asks the Menu Item database if an item that does not exist in the system exists | 1. Returns false | Pass |
| 32 | 1. Menu Item Storage | #29 | 1. Asks the Menu Item database if the topping Bacon exists | 1. Returns true | Pass |
| 33 | 1. Menu Item Storage | System is Running | 1. Asks if a topping that does not exist in the Menu Item database exists | 1. Returns false | Pass |
| 34 | 1. Menu Item Storage | #29 | 1. Asks if Bacon is an associated topping to Pizza | 1. Returns true | Pass |
| 35 | 1. Menu Item Storage | System is Running | 1. Asks if a topping that is not associated with Pizza is associated with Pizza. | 1. Returns false | Pass |
| 36 | 1. Menu Item Storage | #29 | 1. Save the Menu Item database | 1. A file should exist called MenuItemList.settings | Pass |
| 37 | 1. Menu Item Storage | #37  The system is restarted | 1. The instance of the Menu Item storage is obtained | 1. The Pizza item should exist with its settings.  2. The topping Bacon should exist  3. The topping association between Pizza and Bacon should exist | Pass |
| 38 | 1. Oven Kitchen | The system is running | 1. Create an oven with size 10 | 1. The Oven Kitchen should have an oven of size 10. | Pass |
| 39 | 1. Oven Kitchen | #38 | 1. Save the Oven Kitchen settings | 1. A file should exist called OvenKitchen.settings | Pass |
| 40 | 1. Oven Kitchen | #39  The system is reset | 1. Get the instance of the Oven Kitchen | 1. It should contain an oven of size 10 inside. | Pass |
| 41 | 1. Oven Kitchen | #38 | 1. Removes the oven of size 10 | 1. The Oven Kitchen's list of Ovens should be empty | Pass |
| 42 | 1. Preparation Kitchen | System is running | 1. Set the number of cooks in the Preparation Kitchen to 2 | 1. The number of cooks should be 2. | Pass |
| 43 | 1. Preparation Kitchen | #42 | 1. Save the Preparation Kitchen settings | 1. A file should exist named PreparationKitchen.settings | Pass |
| 44. | 1  Preparation Kitchen | #43  The system is restarted | 1. Get the instance of the Preparation Kitchen | 1. The Preparation Kitchen should have 2 cooks. | Pass |
| 45 | 1. Customer Storage | #1 | 1. Create a Customer with order ID of 5558675309, last name Rivings, first name Rose, and the delivery location being the object created in #1 | 1. The Customer database should have a Customer with the prior characteristics | Pass |
| 46 | 1. Customer Storage | System is running | 1. Creates a Customer with an already existing or negative order ID or has a last name, first name, or delivery location of null | 1. The customer should not be stored in the database | Pass |
| 47 | 1. Customer Storage | #45 | 1. Asks for the customer with the order ID of 5558675309 in the system | 1. Returns the object created in #46 | Pass |
| 48 | 1. Customer Storage | System is running | 1. Asks for a customer with an order Id that is not in the system | 1. Returns a null object | Pass |
| 49 | 1. Customer Storage | #45 | 1. Asks if a customer with the order ID number of 5558675309 exists | 1. Returns true | Pass |
| 50 | 1. Customer Storage | System is running | 1. Asks if a customer with an order ID that does not exist in the system exists | 1. Returns false | Pass |
| 51 | 1. Customer Storage | #45 | 1. Saves the Customer database | 1. A file should exist called CustomerList.settings | Pass |
| 52 | 1. Customer Storage | #51  The system is restarted | 1. Get the instance of the Customer database | 1. Should have the customer created in #46 | Pass |
| 53 | 1. Ordered Item | #27 | 1. Creates an ordered item with the name Pizza, left hand toppings of Bacon, no right hand toppings, and no associated order | 1. An OrderedItem with the prior defined parameters is created.  2. Proper values are places, like the time the order was taken | Pass |
| 54 | 1. Ordered Item | System is running | 1. Creates an item with a non-existing or null name | 1. The OrderedItem is not properly created | Pass |
| 55 | 1. Order | #45, 53 | 1. Creates an Order with the customer being the object created in #46 and the order contents being a list containing the OrderedItem created in #55  2. Waits 30 seconds | 1. The Order has those as part of its contents, and saves appropriate data (like the cost of the order) | Pass |