SOFTWARE TEST SUITE

for

TransportX

Transportation Company Computerization Software

Version 1.0.4 approved

Prepared by:
Abhay Kumar Keshari (20CS10001)
Aniket Kumar (20CS10083)
Tanmay Mohanty (20CS10089)

IIT Kharagpur

Contents

Revision History

Name	Date	Reason For Changes	Version
1	15/03/22	The entire SRS has been formulated	1.0.0
2	17/03/22	The Use Cases and diagrams are updated	1.0.1
3	18/03/22	Class Diagram are updated	1.0.2
4	19/03/22	Test Cases planned	1.0.3
5	20/03/22	Minor Bugs Found and Fixed	1.0.4

1 Class Testing

1.1 Test Cases for Manager Class

1.1.1 Constructor Testing

Input:

Name = "Tanmay Mohanty"
Street Name = "C-210, LBS Hall of Residence, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"
D.O.B = "21/09/2002"
email = "tanmaymohanty@gmail.com"
Password = "Tanmay@12345"

Response:

Assert if all inputs are correctly assigned to each attribute of Manager class. A dialogBox appears with the generated Manager ID.

1.1.2 Method Testing

Methods:

1. setID()
2. getID()
3. setPassword()
4. getPassword()
5. getDOB()
6. getAddress()
7. getName()
8. getEmail()

Response: Assert that all the inputs to be supplied to the constructors are of appropriate data types and the Setters-Getters are working correctly, by cross - verification via manual check from database.

1.2 Test Cases for Office Class

1.2.1 Constructor Testing

Input:

Street Name = "Technology Market, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"
Zipcode = "721301"
Rate = 12
PhoneNumber = "033-7843982892"

Input:

Street Name = "Anand Vihar"
City = "New Delhi"
State = "New Delhi"
Zipcode = "110092"
Rate = 15
PhoneNumber = "011-841364626"

Response:

Assert if all inputs are correctly assigned to each attribute of Office class. A dialogBox appears with the generated Office ID

1.2.2 Method Testing

Input

1. getID()
2. setID()
3. setAddress()
4. getAddress()
5. setPhoneNumber()
6. getZIP()
   Response: Assert that all the inputs to be supplied to the constructors are of appropriate data types and the Setters-Getters are working correctly, by cross - verification via manual check from database.
7. getNumConsignment()
8. addConsignment()
   Response: Assert that the consignment is added to office database, by cross - verification via manual check from database.
9. getNumTrucks()

10. getAvgWaitTime()
    Response: Assert that the idle time is correctly calculated by the formula avg(ConsignmentDepatureT
    - ConsignmentArrivalTime), by cross - verification via manual check from database.

11. getRate()

12. updateRate()
    Response: Assert that rate, if changed at any time, the change is reflected in the database, by cross - verification via manual check from database.

13. getStoredVol()

14. updateStoredVolume()
    Response: Assert that total volume of stored is increasing by increments of volume of consignments, by cross - verification via manual check from database.

15. getTotalRevenue()

16. updateRevenue()
    Response: Assert that office revenue increasing by increments of cost of consignments handled, by cross - verification via manual check from database.

17. addTruck()
    Response: Assert that truck is added to database upon arrival at an office, by cross - verification via manual check from database.

18. dispatchTruck()
    Response: Assert that a truck is removed from office database, upon departure from office, by cross - verification via manual check from database.

19. getDispatchedVol()

20. updateDispatchedVolume()
    Response: Assert that total volume of consignments delivered in its entire existence period is increasing by increments of volume of consignments, by cross - verification via manual check from database.

1.3 Test Cases for Employee Class

1.3.1 Constructor Testing

Input

Name = "Neeraj Venkat Naidu"
Street Name = "C-208, LBS Hall of Residence, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"
Office = "721301"

email = "neeraj.naidu@gmail.com"
DOB = "01/01/2002"
Password = "Neeraj@12345"

Input

Name = "Shashank Goud"
Street name = "E-702, Anand Vihar"
City = "New Delhi"
State = "New Delhi"
Office = "110092"
email = "shashank.goud@gmail.com"
DOB = "02/01/2002"
Password = "Shashank@12345"

Input

Name = "Dharavath Yuvaraj"
Street Name = "C-111, LBS Hall of Residence, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"
Office = "721301"
email = "dharavath.yuvaraj@gmail.com"
DOB = "03/01/2002"
Password = "Yuvaraj@12345"

Response:

Assert if inputs are correctly assigned to each attribute of Employee class. A dialogBox appears with the generated Employee ID.

1.3.2 Method Testing

Input

1. setID()
2. getID()
3. getName()
4. setPassword()
5. getPassword()
6. setOffice()
7. getOffice()
8. getAddress()
9. getDOB()

10. getEmail()

Response: Assert that all the inputs to be supplied to the constructors are of appropriate data types and the Setters-Getters are working correctly, by cross - verification via manual check from database.

1.4 Test Cases for Customer Class

1.4.1 Constructor Testing

Input
Name = "Subhajyoti Halder"
Street Name = "C-214, LBS Hall of Residence, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"

Input
Name = "Aditya Ranjan Jha"
Street Name = "C-215, LBS Hall of Residence, IIT Kharagpur"
City = "Kharagpur"
State = "West Bengal"

Input
Name = "Rushil Venkateshwar"
Street Name = "D-500, Anand Vihar"
City = "New Delhi"
State = "New Delhi"

Response:
Assert if inputs are correctly assigned to each attribute of Customer class. A dialogBox appears with the generated Customer ID.

1.4.2 Method Testing

Input

1. setID()
2. getID()
3. getName()
4. getAddress()
   Response: Assert that all the inputs to be supplied to the constructors are of appropriate data types and the Setters-Getters are working correctly, by cross - verification via manual check from database.

5. addConsignment()
   Response: Assert that consignment is added appropriately into customer-specific database.

6. viewConsignmentDetails()
   Response: Assert that Customer is able to view details and status of his booked consignment.

1.5 Test Cases for Truck Class

1.5.1 Constructor Testing

Input
License Number = "WB5556EF"
Current Office = "721301"
Input
License Number = "WB5615CD"
Current Office = "721301"
Input
License Number = "WB6541AB"
Current Office = "110092"
Response:
Assert if inputs are correctly assigned to each attribute of Truck class.
A dialogBox appears with the generated Truck ID.

1.5.2 Method Testing

Input

1. setID()

2. getID()

3. setCurrOffice()

4. getCurrOffice()

5. getVolume()

6. updateVolume()
   Response: Assert that the total volume handled by truck, in its entire usage period, is increasing by increments of every delivery's total volume, by cross - verification via manual check from database.

7. setArrivalTime()

8. getArrivalTime()
   Response: Assert that the time at the moment of arrival of truck at an office is set as its ArrivalTime, by cross - verification via manual check from database.

9. setDepartureTime()

10. getDepartureTime()
    Response: Assert that the time at the moment of departure of truck at an office is set as its DepartureTime, by cross - verification via manual check from database.

11. getAvgIdleTime()
    Response: Assert that the idle time is correctly calculated by the formula avg(DepatureTime - ArrivalTime), by cross - verification via manual check from database.

1.6 Test Cases for Consignment Class

1.6.1 Constructor Testing

Input

Volume = "396"
Sender = "Subhajyoti Halder"
Receiver = "Rushil Venkateshwar"
srcOffice = "721301"
destOffice = "110092"
ArrivalTime = java.time.LocalDateTime.now()

Input

Volume = "150"
Sender = "Aditya Ranjan Jha"
Receiver = "Rushil Venkateshwar"
srcOffice = "721301"
destOffice = "110092"
ArrivalTime = java.time.LocalDateTime.now()

Response:

Assert if inputs are correctly assigned to attributes of Consignment class.
A dialogBox appears with the generated Consignment ID.

1.6.2 Method Testing

Input

1. getID()

2. setID()

3. getVolume()

4. getSender()
5. getReceiver()
6. getsrcOffice()
7. getdestOffice()
8. setArrivalTime()
9. getArrivalTime()
   Response: Assert that all the inputs to be supplied to the constructors are of appropriate data types and the Setters-Getters are working correctly, by cross - verification via manual check from database.
10. setCost()
11. getCost()
    Response: Assert that cost is calculated correctly by the formula (Cost decided by Office / Volume)*(Volume of Consignment), by cross - verification via manual check from database.
12. getDelStatus()
13. setDelStatus()
    Response: Assert that delivery status is updated from time to time, depending on Truck allotment and total inventory volume, by cross - verification via manual check from database.
14. setAssignedTruck()
15. getAssignedTruck()
    Response: Assert that whenever total volume of consignments to a destination office is ¿ 500 m^3, a truck at the office is alloted for their delivery, or whenever the next truck is available, by cross - verification via manual check from database.
16. setDepartureTime()
17. getDepartureTime()
    Response: Assert that whenever a truck is alloted for a delivery, the departure time is same as the current time at allotment, by cross - verification via manual check from database.

2 Test Cases for Authentication

2.1 Test cases for Login

Requirements:
Login Credentials (UserID, Password)
Actor
Manager/Employee
Case 1:
UserID is not found.
A DialogBox appears with an error message "UserID not found".
Case 2:
Incorrect password is entered.
A DialogBox appears with an error message "Incorrect password".
Case 3:
Correct credentials entered. Logged-in into respective interface.

2.2 Test cases for Signup

Requirements:
Manager logged in into his account
Employee details
Case 1:
Email authentication Failed.
A dialogBox appears with an error message "Invalid Email ID".
Case 2:
No Branch with the entered Branch ID exits.
A DialogBox with the an error message "No Branch with ID $\langle enteredID \rangle$ exists".
Case 1:
User successfully created.
A dialogBox appears with a message "Successfully registered with UserID: $\langle UserID \rangle$".

3 Database Testing

3.1 Test Connection Setup with database

Requirements:
Initialization of application
Case 1:
Communications link failure. Application testing aborted.
Case 2:
Successful initialization of application.

3.2 Testing data insertion

Requirements:
Creation of a Manager, Office, Employee, Truck, Consignment or Customer.
Assert whether error-free data is inserted into database.

3.3 Testing database revision

Requirements:
Reset password of Manager/Employee.
Truck Allotment.
Dispatch or arrival of Truck.
Rate changes in office.
Back-end activities.
Assert whether desired revision of database occurred.

3.4 Testing data retrieval

Requirements:
Manager queries about Consignments, Office or Truck
Customer queries about his Consignment
Bill generation.
Print Consignments details to be forwarded with truck.
Data validation and verification.
Back-end Activities.
Assert whether error-free data is retrieved from database.

4 Test Cases for Application

Below is the procedure pathway through the program to achieve the execution:

1. Rate:
   - Currently decided on office to office basis, by he manager, per unit m^3.
2. Manager 1 created:
   - Name: Tanmay Mohanty
   - Street Name = C-210, LBS Hall of Residence, IIT Kharagpur
   - City = Kharagpur
   - State = West Bengal
   - D.O.B = 21/09/2002
   - email = tanmaymohanty@gmail.com
   - Password = Tanmay@12345
3. User ID popped up for Manager 1 (Consider: M1)
4. Manager 1 logs in:
   - User ID: M1
   - Password = Tanmay@12345
5. Now, Manager has got rights to perform many tasks
6. Let us first create an office
7. Office 1 Establishment:
   - Street name = Technology Market, IIT Kharagpur
   - City = Kharagpur
   - State = West Bengal
   - Zipcode = 721301
   - Rate = 12
   - Phone Number = 033-7843982892
8. Office ID popped up for Office 1 (Consider: O1)
9. Office 2 Establishment:

- Street name = Anand Vihar
- City = New Delhi
- State = New Delhi
- Zipcode: 110092
- Rate = 15
- Phone Number = 011-841364626

10. Office ID popped up for Office 2 (Consider: O2)
11. Let him now create 3 employee.
12. Employee 1 Account Creation:
    - Name = Neeraj Venkat Naidu
    - Street Name = C-208, LBS Hall of Residence, IIT Kharagpur
    - City = Kharagpur
    - State = West Bengal
    - Office = 721301
    - email = neeraj.naidu@gmail.com
    - DOB = 01/01/2002
    - Password = Neeraj@12345
13. User ID popped up for Employee 1 (Consider: E1)
14. Employee 2 Account Creation:
    - Name = Shashank Goud
    - Street Name = E-702, Anand Vihar
    - City = New Delhi
    - State = New Delhi
    - Office = 110092
    - email = shashank.goud@gmail.com
    - DOB = 02/01/2002
    - Password = Shashank@12345
15. User ID popped up for Employee 2 (Consider: E2)
16. Employee 3 Account Creation:
    - Name = Dharavath Yuvaraj
    - Street Name = C-111, LBS Hall of Residence, IIT Kharagpur

- City = Kharagpur
- State = West Bengal
- Office = 721301
- email = dharavath.yuvaraj@gmail.com
- DOB = 03/01/2002
- Password = Yuvaraj@12345

17. User ID popped up for Employee 3 (Consider: E3)
18. Now the Manager adds new Trucks
19. Adding Truck 1:
    - License Plate Number = WB5556EF
    - Current Office = 721301
20. Truck ID popped up for Truck 1 (Consider: T1)
21. Adding Truck 2:
    - License Plate Number = WB5615CD
    - Current Office = 721301
22. Truck ID popped up for Truck 2 (Consider: T2)
23. Adding Truck 3:
    - License Plate Number = WB6541AB
    - Current Office = 110092
24. Truck ID popped up for Truck 3 (Consider: T3)
25. Manager 1 logs out
26. Let's take the case where a customer comes in with a consignment at O1, and sends to O2.
27. Employee 1 is logs in
    - User ID: E1
    - Password = Neeraj@12345
28. Sender registration:
    - First, if customer already has an User ID, we will update his old data.
    - If new customer comes in (as this case)
        - Name = "Subhajyoti Halder"

- Street Name = "C-214, LBS Hall of Residence, IIT Kharagpur"
- City = "Kharagpur"
- State = "West Bengal"
- Office = 721301

- ID popped up for Customer 1 (Consider: CS1)

29. Now, employee registers the receiver as a customer, if not already in our database
    - Name = "Rushil Venkateshwar"
    - Street Name = "D-500, Anand Vihar"
    - City = "New Delhi"
    - State = "New Delhi"
    - Office = 110092
30. ID popped up for Customer 2 (Consider: CS2)
31. Now the employee verifies the consignment, and enters the details:
32. Consignment registration:
    - Volume = "396"
    - Sender = CS1
    - Receiver = CS2
    - srcOffice = 721301
    - destOffice = 110092
    - arrivalTime = java.time.LocalDateTime.now()
33. Consignment ID popped up for Consignment 1 (Consider: CN1)
34. In the back-end, the volume is added to the destination office and cost is calculated and truck allotment is being checked.
35. Let us again take another consignment from O1 to O2
36. New Customer registration:
    - Name = "Aditya Ranjan Jha"
    - Street Name = "C-215, LBS Hall of Residence, IIT Kharagpur"
    - City = "Kharagpur"
    - State = "West Bengal"
    - Office = 721301
37. User ID popped up for Customer 3 (Consider: CS3)

38. Consignment 2 registration:
    - Volume = "150"
    - Sender = CS3
    - Receiver = CS2
    - srcOffice = 721301
    - destOffice = 110092
    - arrivalTime = java.time.LocalDateTime.now()
39. Consignment ID popped up for Consignment 2 (Consider: CN2)
40. Now, the sum of volumes of CN1 and CN2 is ¿500 m^3, so a truck is alloted to them, which is T1.
41. T1 travels from O1 to O2, and the truck is now having current office as O2.
42. If a customer enquires comes at anytime about his consignment status, an employee inputs the following:
    - Consignment ID
43. Status of consignments changes from time to time, depending on truck allotment or inventory storage.
44. Employee 1 logs out
45. Manager 1 logs in
46. If he wants to check the statistics of the logistics, revenue, cost and consignment delivery
47. The following options can be seen by Manager:
    - Truck Usage
    - Average Consignment Waiting Time
    - Revenue Generated
    - Volume Handled
    - Average Idle Waiting Time of Trucks
    - Truck Status
    - Consignment Status
48. These would help him to take important business decisions.
49. Manager 1 logs out.