

Transport Company Computerisation **Software**

Test Suite Design Document and **Test Results**

B419013 Akshit Gangwar

B419052 Satyam Pandey

B419039 Sumeru Das

B419039 Pawan Kumar Mishra

Introduction

The test suite design is a high-level overview which provides a way that helps in formulating a testing strategy for the Transport Company Computerisation Software. The objective of the test suite is to communicate the quality of project standards and procedure. We will use the methods of white-box testing, black-box testing and system-testing in order to lay out an effective test plan for our software. The test suite also portrays a snapshot of the project as it was at the end of the planning phase. The test suite is designed to look into and follow all the standards of unit-testing, integration-testing and system-testing. The paradigm includes but is not limited to methods, testing criteria and designing test cases for the overall design. The documentation specifications described in the IEEE Standard 829-1983 for Software Test Documentation will be applied.

Test Objective

The main objective of the test suite is to find out and report as many bugs as there are in the program and hence improve the overall integrity and performance of the software. We cannot carry out exhaustive testing as the number of such test cases would be very large possibly infinite. But we have tried to include the most important test cases and most importantly remove the redundant ones in order to reduce the size of the test suite. Our software the Transport Company Computerisation Software (TCCS) has various utilities to be tested such as login, placing consignment, despatching orders, issuing bills, allotting trucks, adding new employee as well as branches. All of these functionalities have been implemented in the GUI and we tried to keep the interface as simple as possible so that even a layman can use the software to the utmost of his potential.

Process Overview

The following represents the overall flow of the testing process employed:

1. The very first step is to identify all the requirements to be tested. This step is done with the help of the Software Requirements Specification or SRS prepared in an earlier phase and hence we derive all the required test cases.
2. The next step is to identify the test cases that will test the various modules.
3. We then move on to verify that the test cases and test data are adequate to test the various units of the Transport Company Computerisation Software
4. Before we apply the test cases on the software it is necessary to identify the expected results so that they can later be used to verify the outputs that result from the application of the test cases in the TCCS.
5. Then we document the test data, test configuration and the expected results as found out in the earlier step.
6. Finally, the tests are performed on the system developed.
7. Again, the outputs are documented in the Test Report.

8. Successful unit testing is essential before one can proceed to the integration and finally the system testing.
9. In case of a unsuccessful testing a bug report is generated and it describes the test case, the point where the problem was generated and its possible cause and the sequence of events that led to the mismatch between the generated and expected outputs. This will form the future basis for our technical analysis in removing the bug.

Test Process

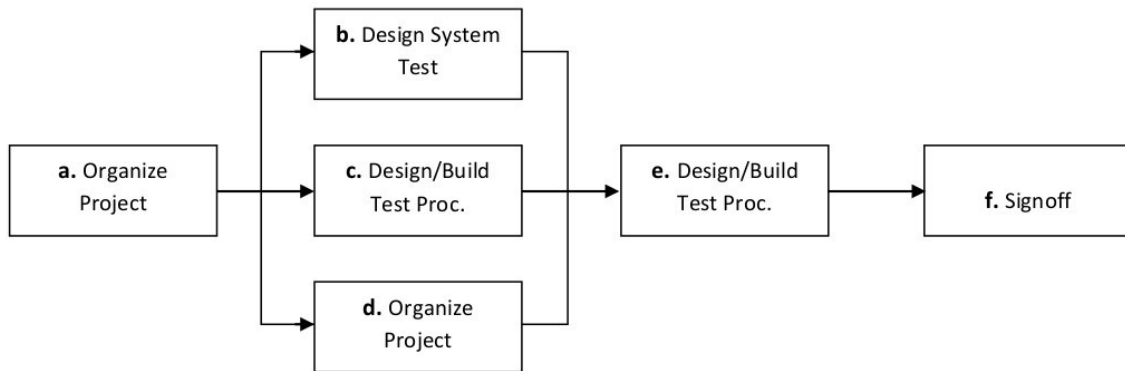


Figure – Test Process

The diagram above outlines the Test Process that has been followed in the test suite design.

- a. Organising the project implies creating the system test plan, schedule the test plan, approach and assign the various responsibilities.
- b. To design and build system test means identifying the test cases, test cycles, the entrance and exit criterion and most importantly the expected results. The test data and test cases are identified and the conditions in which the software is to be tested are derived from the system requirements specifications.
- c. Through the designing of building of the test procedure we set up procedures such as error management systems and reporting the status of the status under various test conditions.
- d. While building the test environment we build the hardware, software and the data paths or may request them from reliable sources.
- e. Finally the test cases are executed and the results are compared with that of the expected results and the bugs are documented, and finally forwarded to the development team for improvement.
- f. The signing off happens when a predefined exit criteria has been achieved.

A failure is a manifestation of an error (also defect or bug). Mere presence of an error may not lead to a failure. So we see that whether the outputs match the expected output and in case they do not do so we debug the program and correct the errors.

Black Box Testing For Add Manager to the Company

The details are entered into the user interface after clicking on the register button the login screen. The details after being entered into the screen are shown in the following screenshot.



A screenshot of a Windows-style dialog box titled "Register as Manager". It contains several text input fields for registration details: "Id" (containing "1"), "Name" (containing "Sachin Bansal"), "Address" (containing "C-125, JN Rd., Kol-103"), "Mobile" (containing "9433420449"), "Email Id" (containing "bansal@gmail.com"), "Branch Id" (containing "Kol"), "Password" (containing four dots), and "Confirm Password" (containing four dots). At the bottom right, there are "OK" and "Cancel" buttons.

The expected output is that the system will register the manager by updating its database. The following screen shot proves the same and the databases are updated accordingly. So the functionality of the system in adding a new manger has been successfully implemented by the Transport Company Computerisation Software.



Black box Testing For Trying to Register a Second Manager

When the user tries to register a second manager without considering the fact that there already exists a Manager. The following error message is displayed.



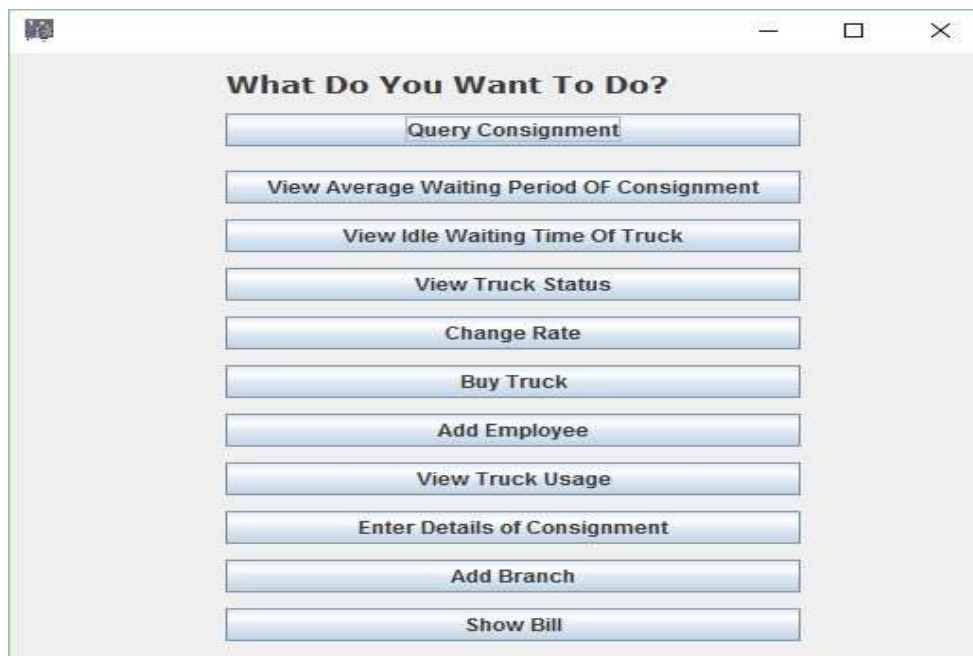
Hence the expected behaviour of the system is obtained in the test case, as the Transport Company is supposed to have a single manager.

Black Box Testing with correct Login credentials for the Manager

The userID and corresponding password are keyed into the respective fields in the user interface of Manager Login as the test case inputs. The screenshot is as follows.

A screenshot of a 'Manager Login' dialog box. It has a title bar with a close button. Inside, there are two input fields: 'UserId' with the value '1' and 'Password' with four dots. At the bottom, there are four buttons: 'Remove', 'Register', 'OK', and 'Cancel'.

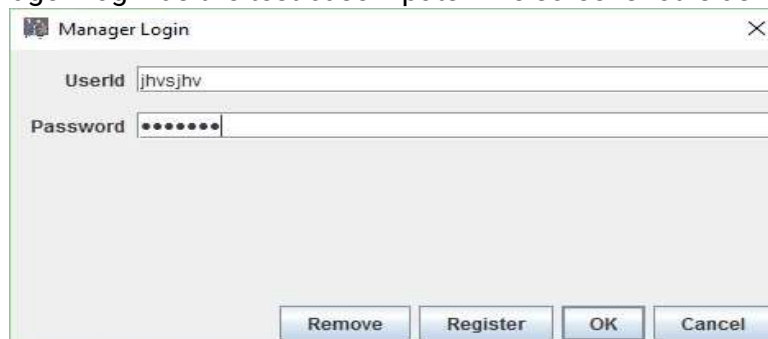
In the case of correct password and userID the manger is allowed the access to perform a large amount of activities on the system and the corresponding manager activity screen pops up. The screen shot for supporting the same follows.

A screenshot of a manager activity screen titled 'What Do You Want To Do?'. It contains a vertical list of eleven buttons: 'Query Consignment', 'View Average Waiting Period OF Consignment', 'View Idle Waiting Time Of Truck', 'View Truck Status', 'Change Rate', 'Buy Truck', 'Add Employee', 'View Truck Usage', 'Enter Details of Consignment', 'Add Branch', and 'Show Bill'.

So, the expected results are displayed in providing the activity screen of the manager and this prove the correct functioning of the TCCS in implementing the functionality of Manager Login.

Black Box Testing with incorrect Login credentials for the Manager

The userID and mismatching password are keyed into the respective fields in the user interface of Manager Login as the test case inputs. The screenshot is as follows.

A screenshot of the 'Manager Login' dialog box. The 'UserId' field contains 'jhvsjhv' and the 'Password' field contains seven dots. The buttons at the bottom are 'Remove', 'Register', 'OK', and 'Cancel'.

The result of pressing OK button on the user interface is shown in the next screenshot.



Hence the output of the test case containing invalid login credentials gives the expected output.

Black box Testing for Adding a Branch by the Manager

The test case input is fed into the system by clicking on the Add Branch option in the activity screen of the manager. The details of the Branch are entered in the corresponding fields in the user interface and the following screen shot was taken.

A screenshot of a "Add Branch" dialog box. It contains four input fields: "Branch Id" with the value "Kol", "Branch Address" with the value "Kolkata", "Branch Phone" with the value "8902552884", and "rate" with the value "40". At the bottom right, there are two buttons: "OK" and "Cancel".

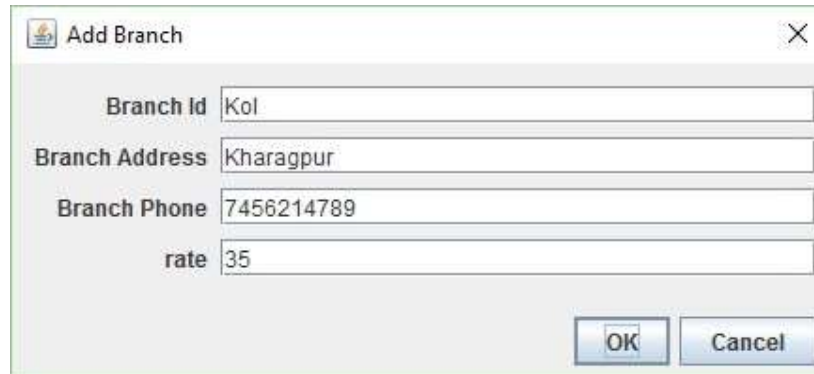
On confirming the credentials and pressing OK button the following assertive message is displayed.



In this case to the expected and observed outputs are the same, confirming the functionality of the adding new branches by the Manager.

Black box Testing for Adding a New Branch with duplicate BranchID

The test case was devised to test the case when the manger while entering the details of new branch keys in the BranchId already used by some other branch. This should not be allowed as the BranchId is set as the primary key in the database for the branch office. The details are entered as shown below, these were entered when the database already contained a branch office with the barnchId "Kol" and hence the error message was displayed as follows subsequently.

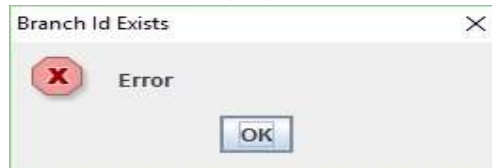


Dialog box titled "Add Branch" with the following fields and values:

Field	Value
Branch Id	Kol
Branch Address	Kharagpur
Branch Phone	7456214789
rate	35

Buttons: OK, Cancel

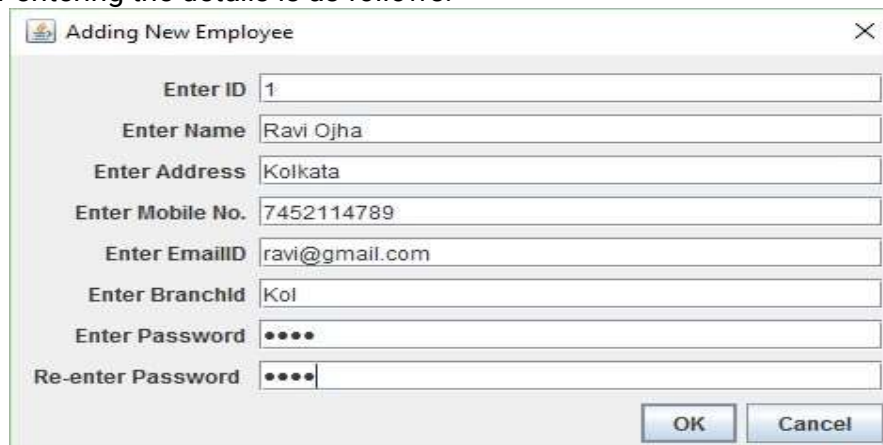
Below is the screenshot of the error message generated.



Hence the system performs as expected in case of the duplicity and doesn't add the new branch.

Black Box Testing for Add new Employee by the Manger

The test case is executed as follows, from the manager activity screen the button for Add Employee is clicked and the screen for entering the credentials of the new Employee opens up. The details are keyed into the fields without any ambiguity in this case, the screen after entering the details is as follows.

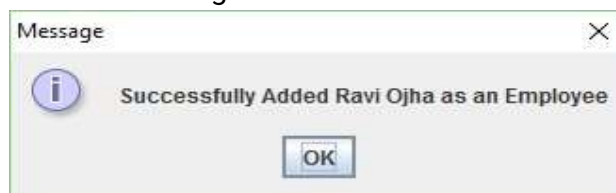


Dialog box titled "Adding New Employee" with the following fields and values:

Field	Value
Enter ID	1
Enter Name	Ravi Ojha
Enter Address	Kolkata
Enter Mobile No.	7452114789
Enter EmailID	ravi@gmail.com
Enter BranchId	Kol
Enter Password	••••
Re-enter Password	••••

Buttons: OK, Cancel

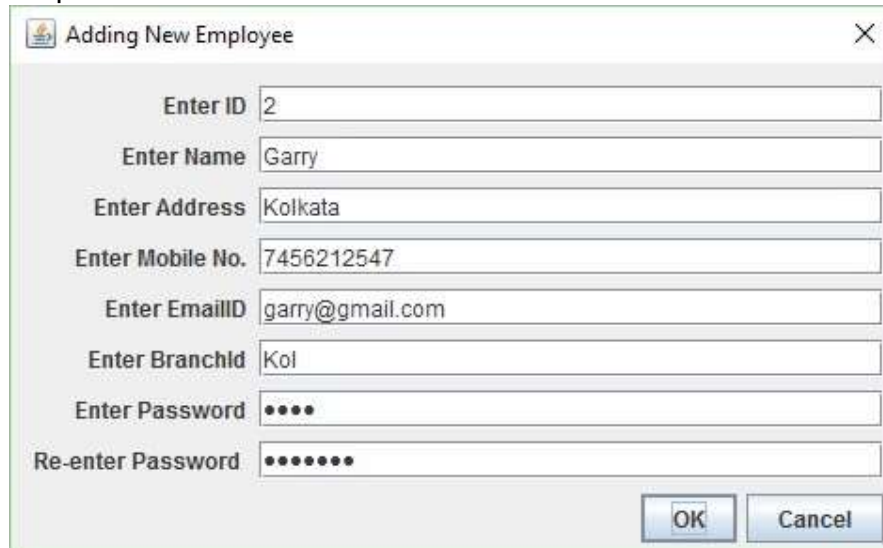
On pressing the OK button the message for successful insertion is diaplyed as follows



So the behaviour of the system is as expected and the functionality of the add employee is confirmed as the employee database is updated accordingly.

Black Box Testing for the case when the password and confirm password fields don't match

This test case is basically the same as the previous one except from the fact that while entering the details in the add employee user interface the two password fields are keyed with different inputs as shown below.



Adding New Employee

Enter ID: 2

Enter Name: Garry

Enter Address: Kolkata

Enter Mobile No.: 7456212547

Enter EmailID: garry@gmail.com

Enter BranchId: Kol

Enter Password: ****

Re-enter Password: *****

OK Cancel

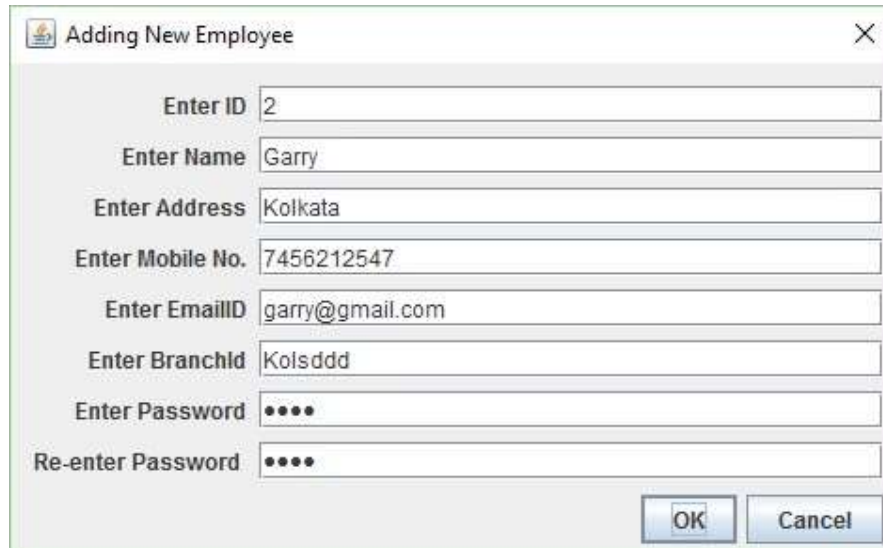
The following error message pops up on pressing the Ok button.



This test case devised to test whether the system takes care of the fact that the user confirms the password properly before adding the employee. The output was as expected and the system does not make any addition to the database for employees in such a erroneous situation.

Black Box Testing for Adding employee to a non Existent Branch Office

This case handles the case when the manager tries to add an employee to a branch which does not exist. The credentials are entered in the user interface as shown below. The test is performed when the BranchId "Kolsddd" did not exist in the database for the branch office. The screenshot for the same is same below.



Adding New Employee

Enter ID: 2

Enter Name: Garry

Enter Address: Kolkata

Enter Mobile No.: 7456212547

Enter EmailID: garry@gmail.com

Enter BranchId: Kolsddd

Enter Password:

Re-enter Password:

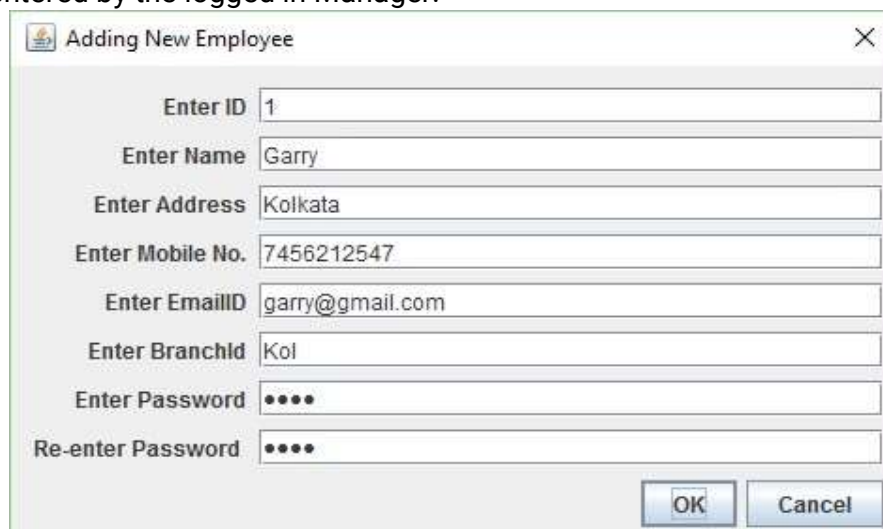
OK Cancel

On pressing the OK button the following error message is displayed



Black Box Testing for adding Employee with duplicate Employee ID

This test case is executed to find out whether the system functions in a correct way when there is a delicacy in the primary key of the employee database. The inputs are keyed in at a time when there already exists a user with the ID "1". The following screenshot is that of the inputs entered by the logged in Manager.



Adding New Employee

Enter ID: 1

Enter Name: Garry

Enter Address: Kolkata

Enter Mobile No.: 7456212547

Enter EmailID: garry@gmail.com

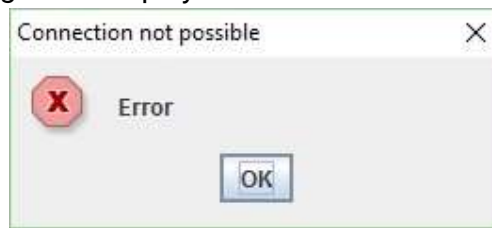
Enter BranchId: Kol

Enter Password:

Re-enter Password:

OK Cancel

The following error message was displayed.

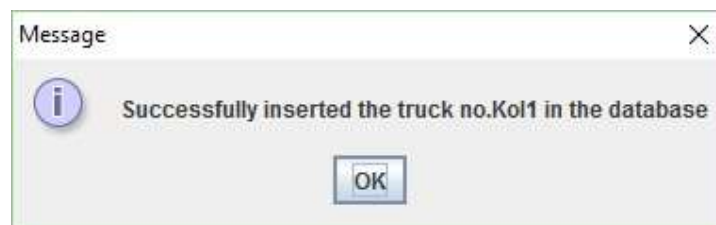


Black Box Testing for Successfully adding a truck to the database

This test case is executed without any ambiguity by entering the truck number and current branch in the user interface as shown in the screenshot below.

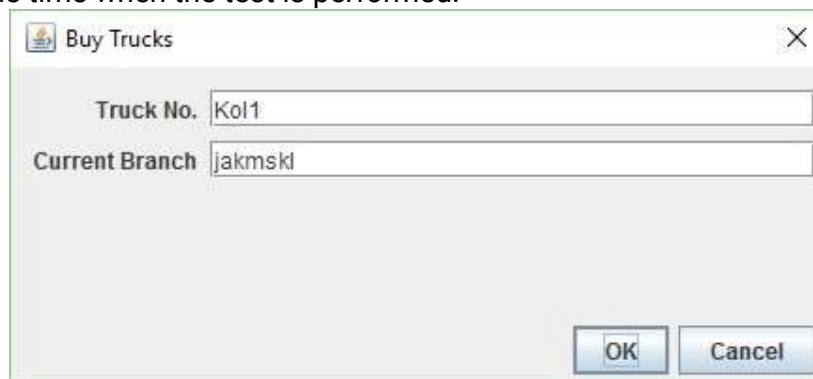


The following message pops up and hence the expected behavior is same as the observed behaviour.



Black Box Testing for adding a truck to the database with a BranchId that does not exist

This test case is executed with by entering the truck number and current branch in the user interface as shown in the screenshot below. The current branch entered is not there in the database at the time when the test is performed.



The observed output was that a error message was displayed and the truck could not be added which is as expected and the following screenshot supports the same.



Black Box Testing for adding a truck to the database with duplicate Truck number

This test case is executed with by entering the truck number "Kol1" which already existed in the database of the trucks and current branch in the user interface as shown in the screenshot below.

A screenshot of a dialog box titled "Buy Trucks" with a close button (X) on the right. It contains two text input fields. The first field is labeled "Truck No." and contains the text "Kol1". The second field is labeled "Current Branch" and contains the text "Kol". At the bottom right are two buttons: "OK" and "Cancel".

The expected output was observed and the truck could not be added by the manager. The screenshot was taken of the error message that followed.



Hence the system supports the functionality of adding a truck as was specified in the software requirements specifications.

Black Box Testing for Changing Rate in the Branch by the Manager

The test case is performed with happy test cases where the BranchId entered is a valid one. The new rate for the corresponding branch is also entered. The screenshot below is the representation of the same.

A screenshot of a dialog box titled "Change The Rate for Any Branch (per cubic metres)" with a close button (X) on the right. It contains two text input fields. The first field is labeled "Branch Id:" and contains the text "Kol". The second field is labeled "New Rate" and contains the text "45". At the bottom right are two buttons: "OK" and "Cancel".

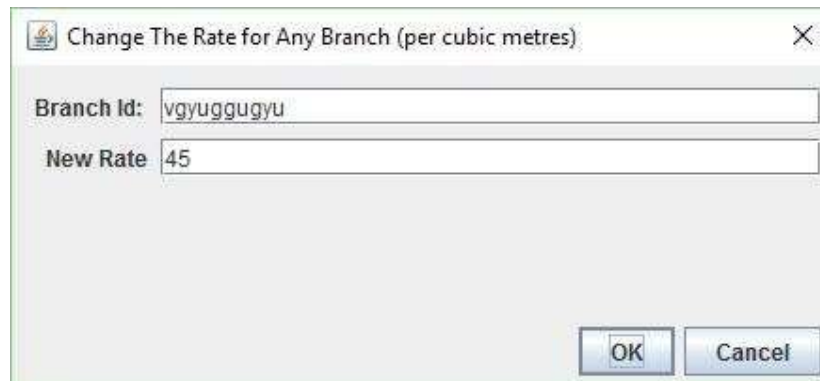
The message displayed is as follows.



This shows that observed and expected outputs are same and the database is updated accordingly.

Black Box Testing for Changing Rate in the Branch by the Manager when Branch is non existent

The test case is performed with unhappy test cases where the BranchId entered is a invalid one. The new rate for the corresponding branch is also entered. The screenshot below is the representation of the same.



The following error message is displayed.



This test case ensures the correct functionality of the change rate function implemented as the observed and expected output are the same.

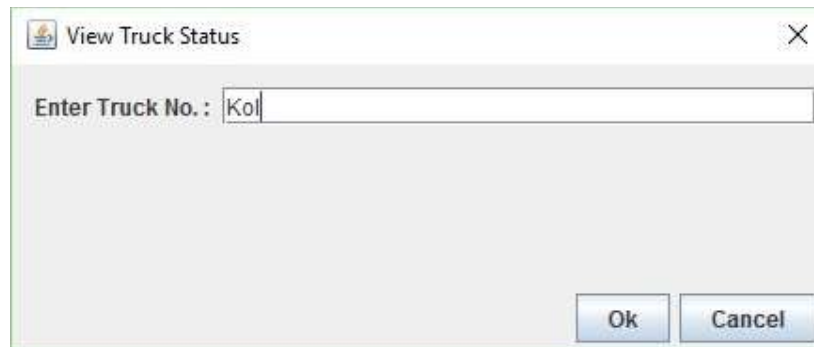
Black Box Testing for finding the average waiting time of Consignment when the consignment count is still zero

The test case is executed at the instance when the database for consignments is empty. The following error message was displayed which shows the query is mathematically invalid as is expected.



Black Box Testing for viewing Truck Status

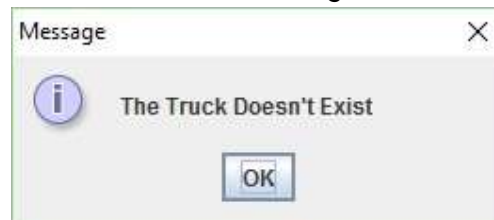
This test case is performed to view the status of any truck. The user is prompted to enter the truck no.



If the Truck exists then it shows the status of the corresponding truck as follows.

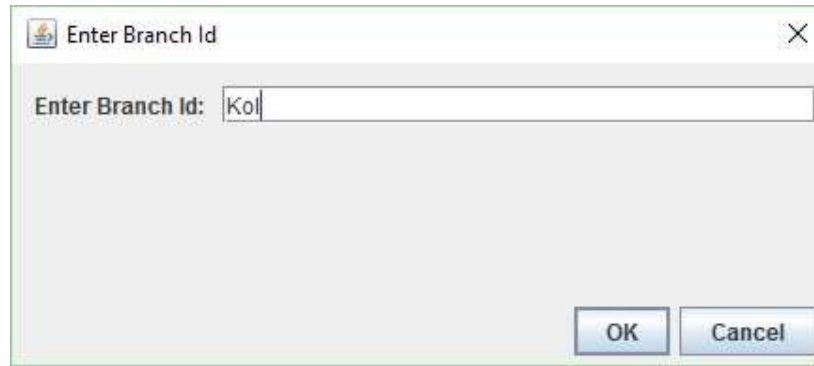


Else if the truck number does not exist the following error message is displayed.

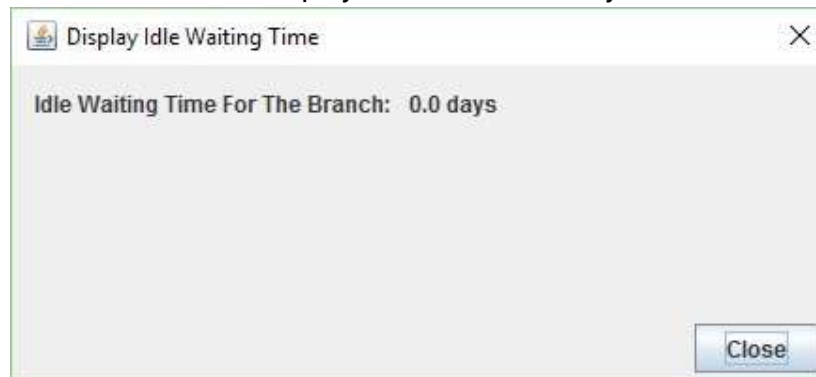


Black Box Testing for Idle waiting time of truck in a BranchId

This test case asks for the Branch ID.



If the branch Id exists the time is displayed in number of days.



Else in case the branch does not exist the error message is displayed.



As the expected and observed outputs are the same so the correctness of the above functionality is confirmed.

Black Box Testing for showing Bill

This test case is performed with unhappy test cases where no consignment with the entered Id exists. The error message that follows is also displayed.





Hence, we see that the observed and the expected outputs are the same and hence the Black Box Testing was successful.

White Box Testing

In the White Box Testing, we will verify that all the paths in a function are correct through basic path testing. White Box tests for different functions are as following.

White Box Testing For allot truck:

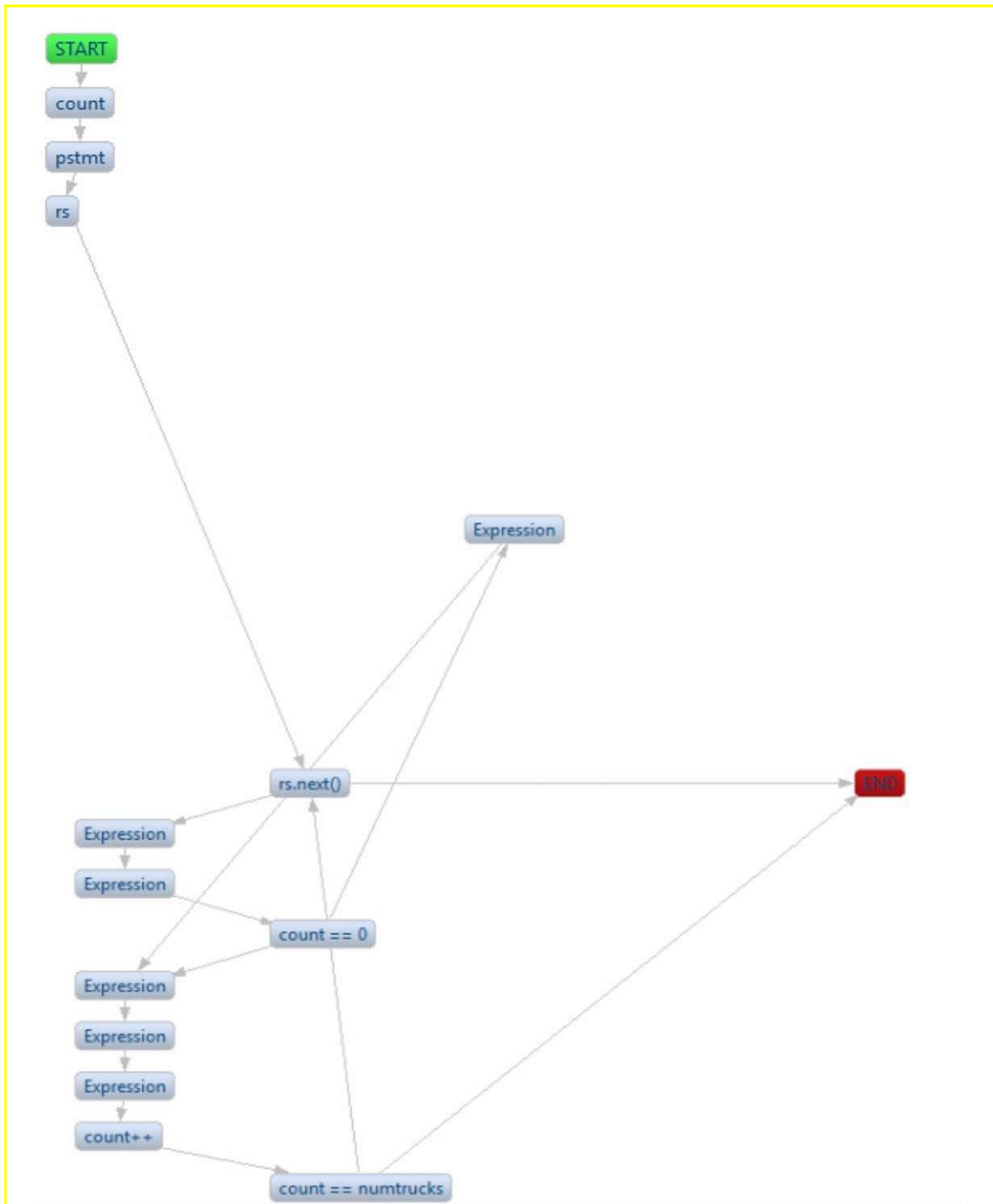
Class Name: CompanyDAO

Code Segments:

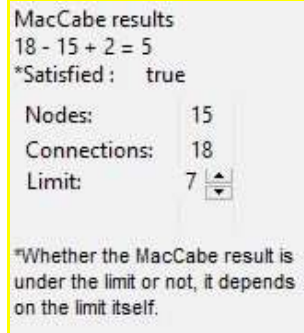
```
void allot_truck(int numtrucks, String branchid, String receiverid, String  
consignmentid) throws SQLException{ int count = 0;  
    PreparedStatement pstmt = con.prepareStatement("select * from truck  
where currentBranch = '"+branchid+"'");  
    ResultSet rs = pstmt.executeQuery();  
    while(rs.next()){ change_status_con(consignmentid,rs.getString(1));  
        change_status_truck(rs.getString(1),receiverid); if(count==0)  
        change_status_branch(branchid,receiverid,consignmentid);  
        increase_truck(1, receiverid); decrease_truck(1, branchid);  
        assign_trucks(rs.getString(1), receiverid,false);  
        count++;  
        if(count == numtrucks) break;  
    }  
}
```

Control Flow Diagram:

The Control Flow Diagram is as follows:



McCabe Result For The CFG:



White Box Testing For assign_truck:

Class Name: CompanyDAO

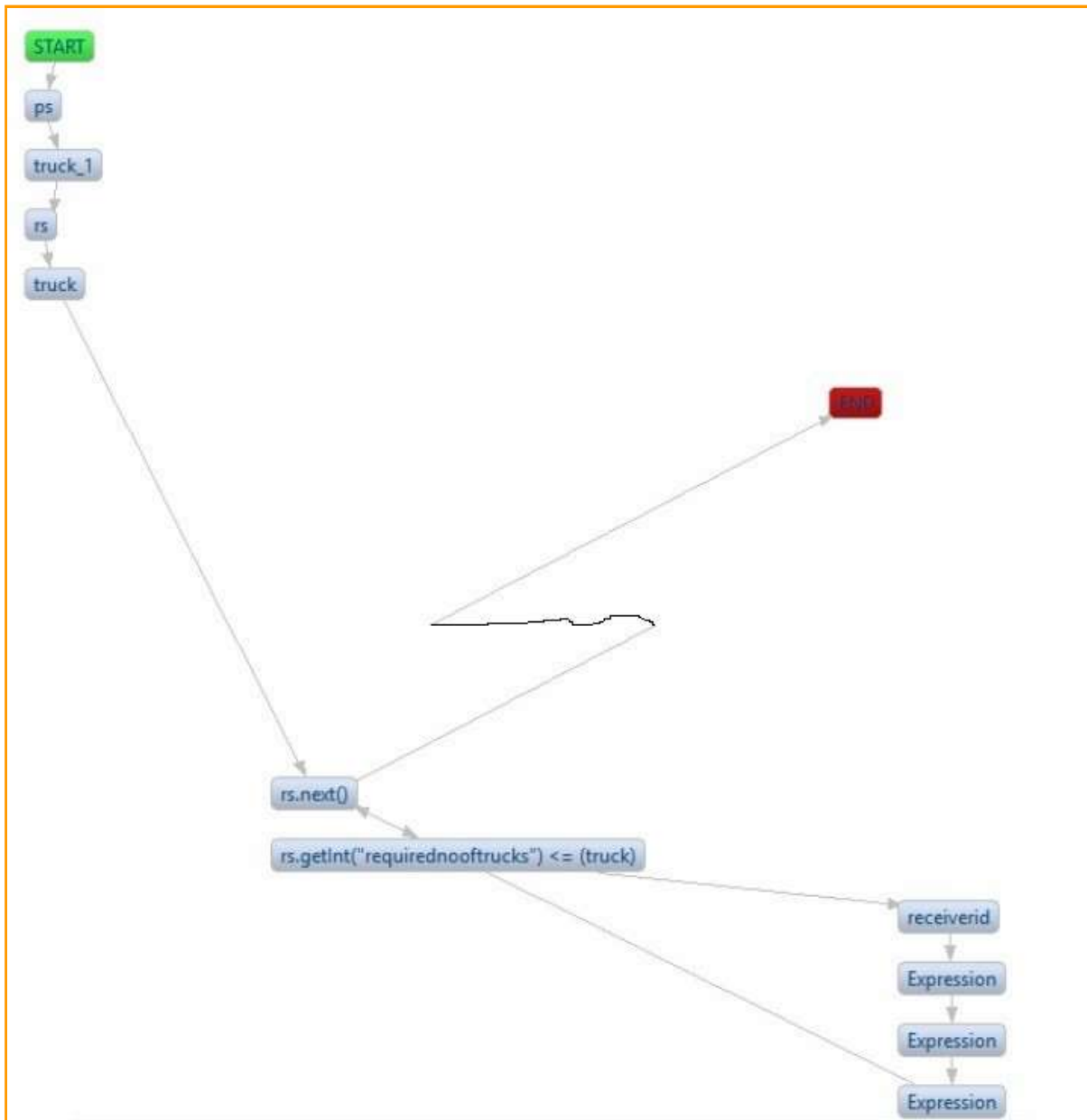
Code Segments:

```
void assign_trucks(String truckno , String branchid,boolean buy) throws SQLException{
    PreparedStatement ps = con.prepareStatement("select * from
consignment where sourceId = '"+branchid+"' and isTruck = 0 order by volume");
    int truck_1 = get_number_of_trucks(branchid);
    ResultSet rs = ps.executeQuery();
    /*if(!buy){
        PreparedStatement pst = con.prepareStatement("update
branchoffice set currentnumoftrucks ="+(truck_1+1)+" where branchId =
 '"+branchid+"'");
        PreparedStatement pst_1 = con.prepareStatement("update
branchoffice set numberoftrucks ="+(truck_1+1)+" where branchId =
 '"+branchid+"'"); pst.executeUpdate();
        pst_1.executeUpdate();
    }*/
    int truck = get_number_of_trucks(branchid); while(rs.next()){
        if(rs.getInt("requirednooftrucks")<=(truck)){
            String receiverid = rs.getString("destId");
            String consignmentid = rs.getString(1);

            allot_truck(rs.getInt("requirednooftrucks"),branchid,receiverid,consignmentid);
            JOptionPane.showMessageDialog(null, "consignment
            "+rs.getInt(1)+" dispatched");
        }
    }
}
```

Control Flow Graph:

The CFG for this code snippet is as follows:



McCabe Result For This CFG:

McCabe results
 $13 - 12 + 2 = 3$
 *Satisfied: true
 Nodes: 12
 Connections: 13
 Limit: 7

*Whether the McCabe result is under the limit or not, it depends on the limit itself.

White Box Test For actionPerformed(ActionEvent e):

Class name: Buy_Truck

Code Segments:

```

public void actionPerformed(ActionEvent e) {
    String truckno = textField.getText();

```

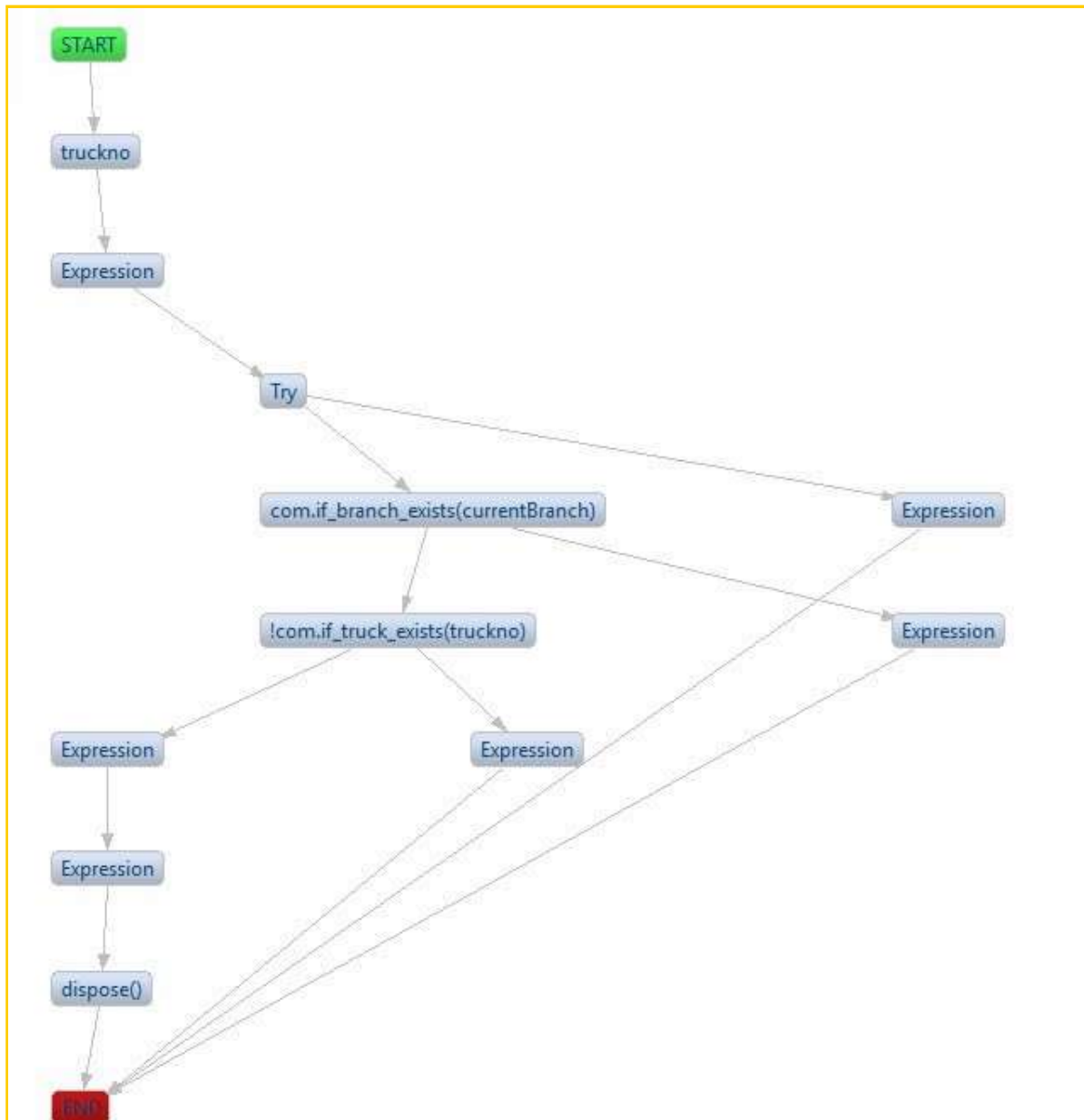
```

        String currentBranch = textField_1.getText();
        try {
            if(com.if_branch_exists(currentBranch)){ com.insert_new_truck(truckno, currentBranch);
            JOptionPane.showMessageDialog(null, "Successfully inserted the truck
no."+truckno+" in the database"); dispose();
            }
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    }

```

Control Flow Diagram:

The CFG For this code snippet is as follows:



McCabe Result For The CFG:

MacCabe results	
15 - 13 + 2 = 4	
*Satisfied: true	
Nodes:	13
Connections:	15
Limit:	7
*Whether the MacCabe result is under the limit or not, it depends on the limit itself.	

White Box Testing For actionPerformed(ActionEvent e)

Class Name: Details_Consignment

Code Segments:

```

public void actionPerformed(ActionEvent arg0) {
    String id = textField.getText();

```

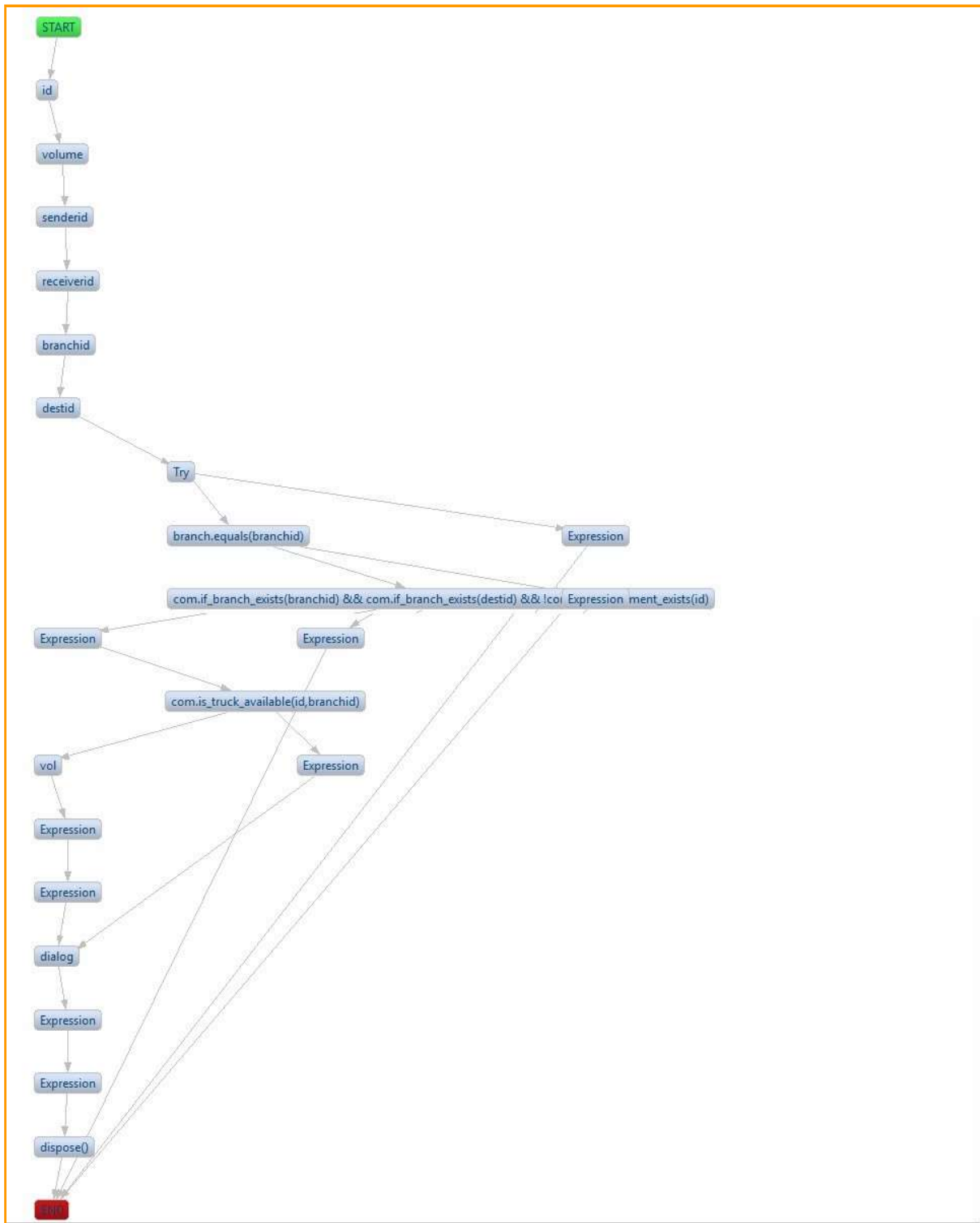
```

int volume = Integer.valueOf(textField_1.getText());
String senderid = textField_2.getText();
String receiverid = textField_3.getText();
String branchid = textField_4.getText();
String destid = textField_5.getText();
try {
    if(com.if_branch_exists(branchid) && com.if_branch_exists(destid) &&
!com.if_consignment_exists(id)){
        com.insert_consignment(id, volume, senderid, receiverid, branchid,
destid);
        if(com.is_truck_available(id, branchid)){ com.allot_truck(volume/500 +1, branchid,
destid, id); dispatched = 1;
        }else{ dispatched = 0;
            } sender_Details dialog = new
sender_Details(dispatched,id,senderid,receiverid);
        dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        dialog.setVisible(true); dispose();
    }else{
        JOptionPane.showMessageDialog(null,"Error","The Branch Doesn't Exist or
The consignment exists beforehand",JOptionPane.ERROR_MESSAGE);
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

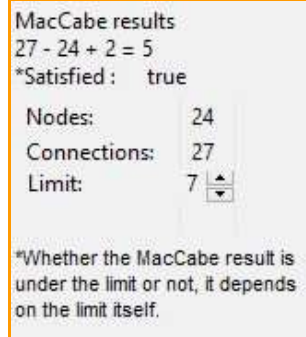
```

Control Flow Diagram:

The CFG For this code snippet is as follows:



McCabe Result For The CFG:



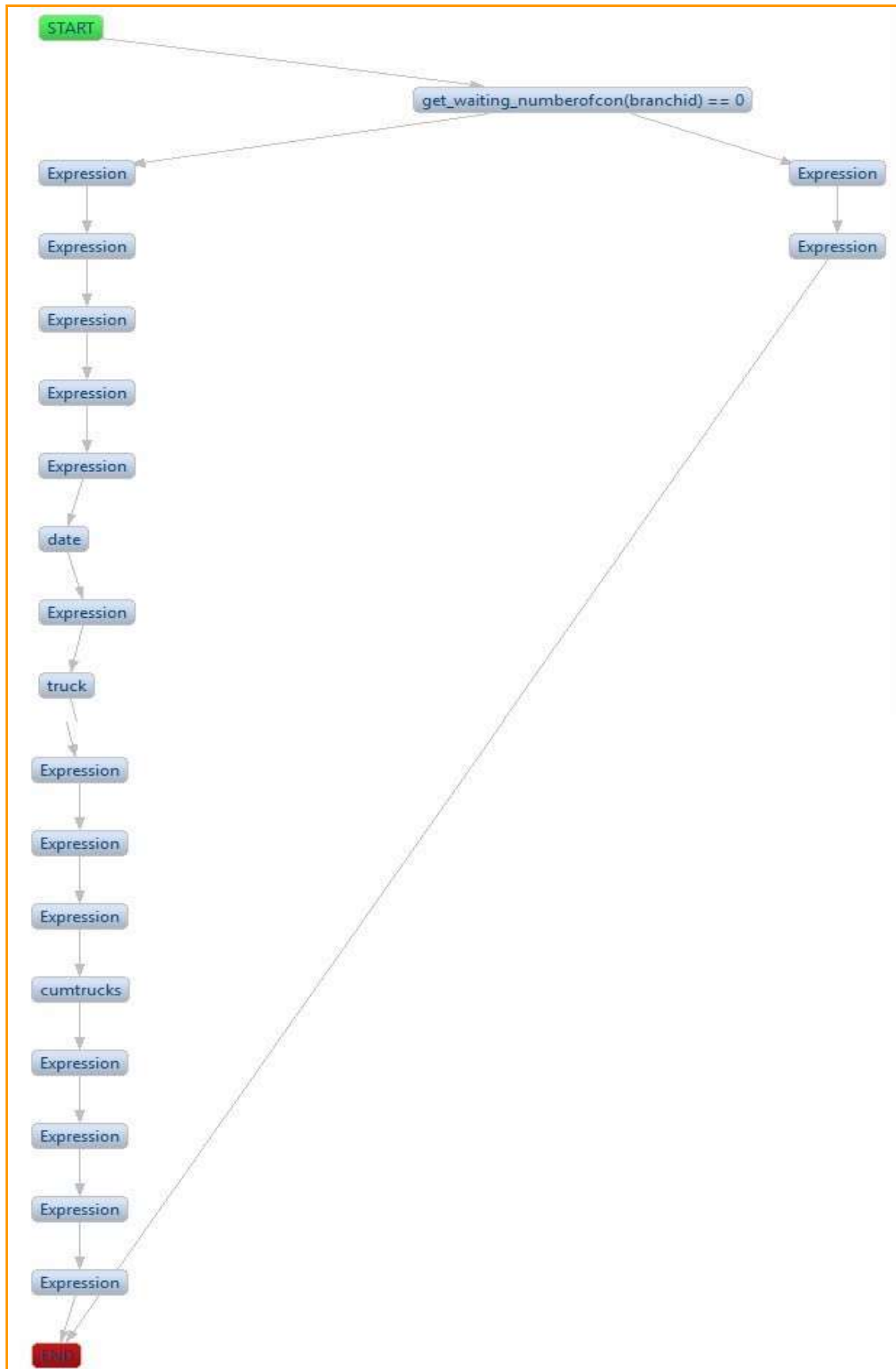
White Box Testing For insert_new_truck:

Class Name: CompanyDAO

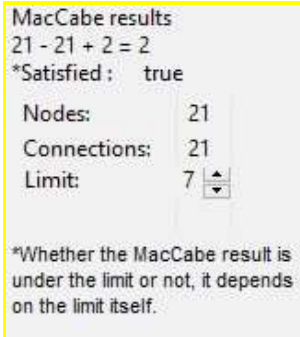
Code Segments:

```
void insert_new_truck(String truckno , String branchid) throws
SQLException{ if(get_waiting_numberofcon(branchid)==0){ stmt3 =
con.prepareStatement("insert into
truck(truckNo,currentBranch,noOfCon,status,buyingdate) values(?,?,?,?,?)");
stmt3.setString(1, truckno); stmt3.setString(2, branchid);
stmt3.setInt(3, 0);
stmt3.setString(4, "waiting in the branchid = "+branchid); java.sql.Date date = new
java.sql.Date(System.currentTimeMillis());
stmt3.setDate(5, date);
int truck = get_number_of_trucks(branchid); stmt1 =
con.prepareStatement("update branchoffice set
currentnumoftrucks = "+(truck+1)+" where branchid = '"+branchid+"'");
int cumtrucks = get_number_of_cum_trucks(branchid); stmt2 =
con.prepareStatement("update branchoffice set
numberoftrucks = "+(cumtrucks+1)+" where branchid = '"+branchid+"'");
stmt3.executeUpdate(); stmt1.executeUpdate();
stmt2.executeUpdate();
}else{ temp_inc_truck(truckno,branchid);
assign_trucks(truckno,branchid,true);
}
} Control Flow
```

Diagram:



McCabe Result For The CFG:



The following sequences of test cases were fed as input to the system in order to test its overall functionality.

The screenshots shown just below represent the initial configuration of the databases where no consignment has yet been added and there are only three branches.

Database of Employees

	Id	Name	Address	Mobile	Email	BranchId	Password
▶	1	Ravi Ojha	Kolkata	7452114789	ravi@gmail.com	Kol	zwFo0ywkM9lb3L2s/9cs91dB7wmikcxBIJGDP40/...
	2	Garry	Kolkata	7456212547	garry@gmail.com	Kol	oUpWYOQ5XPjYG5gtVzpl+Ml/jmSYhZgmiDO3n5j...
	3	Suman Prakash	Kharagpur	7412336987	prakash.suman@gmail.com	Kgp	Vlg+d4OK1M3C20J+CIJ6jQnWmo5G6wXTd+rFb...
	4	Sourav Das	Jamshedpur	7412336547	das.sourav@gmail.com	Jsm	eIM43JK4qfdoyXIFvApzdO15NtDbmReZ3lfURqr...
*		NULL	NULL	NULL	NULL	NULL	NULL

Database of Branch Offices

	branchId	branchAddress	branchphone	numberOftrucks	numberOfemployees	volumehandled	revenue	idlewaitingtime	rate	currentnumoftrucks	numberOfwaitingcon
▶	Jsm	Jamshedpur	7456214789	0	1	0	0	0	35	0	0
	Kgp	Kharagpur	7456214789	0	1	0	0	0	35	0	0
	Kol	Kolkata	8902552884	3	2	0	0	0	45	3	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Trucks

	truckNo	currentBranch	noOfCon	status	buyingdate	arrivedate	dispatchdate
▶	Kol1	Kol	0	waiting in the branchid = Kol	2016-04-18	2016-04-18	NULL
	Kol2	Kol	0	waiting in the branchid = Kol	2016-04-18	2016-04-18	NULL
	Kol3	Kol	0	waiting in the branchid = Kol	2016-04-18	2016-04-18	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Manager

	Id	Name	Address	Mobile	EmailId	BranchId	Password
▶	1	Sachin Bansal	C-125, JN Rd., Kol-103	9433420449	bansal@gmail.com	Kol	OZDCtp9e0ftrqgB0JpKdJQf5bkmwvOPeSU6O...
*		NULL	NULL	NULL	NULL	NULL	NULL

Now the as per our test plan we feed input such that a consignment of volume 700 cubic meters arrives at Kolkata and is destined to Kharagpur. Since 2 trucks are already available the consignment is immediately dispatched to Kharagpur. The following set of Screen shots suggest the entire sequence of events that take place in the Transport Company Computerisation Software(TCCS).

 Employee Activity


×

What do You Want to do?

Enter Consignment Details

Show Bill

Close

 Enter Details Of Consignment

×

Id

1

Volume

700

SenderId

s1

ReceiverId

r1

SourceId


Kol

Destination Id

Kgp

Cancel

Next

 Enter Sender Details

×

Id

s1

Name

John

Address

Kolkata


Mobile No.

74569874123

Email Id

john@gmail.com

Next

 Enter Receiver Details ✕

Id

r1

Name

Garry

Address

Kharagpur

Mobile No.

7456321478

Email Id

garry@gmail.com

OK



The following are the screen shots taken after this dispatch and clearly shows that the databases have been updated accordingly. **Database of Branch Offices**

[illegible]

Database of Consignments

[illegible]

Database of Customers

Id	Name	Address	Mobile	Email	ConsignmentId
r1	Garry	Kharagpur	7456321478	garry@gmail.com	1
s1	John	Kolkata	74569874123	john@gmail.com	1

Database of trucks

truckNo	currentBranch	noOfCon	status	buyingdate	arrivedate	dispatchdate
Kol1	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol2	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol3	Kol	0	waiting in the branchid = Kol	2016-04-18	2016-04-18	NULL

This is the bill generated by the system for the consignment which has been despatched



Displaying Bill For The Consignment

Consignment Id: 1

Volume : 700

Sender Name: John

Receiver Name: Garry

Revenue 28000

Dispatch and Received: Yes

Truck No. Assigned: ,Kol1,Kol2

Source Branch Id: Kol

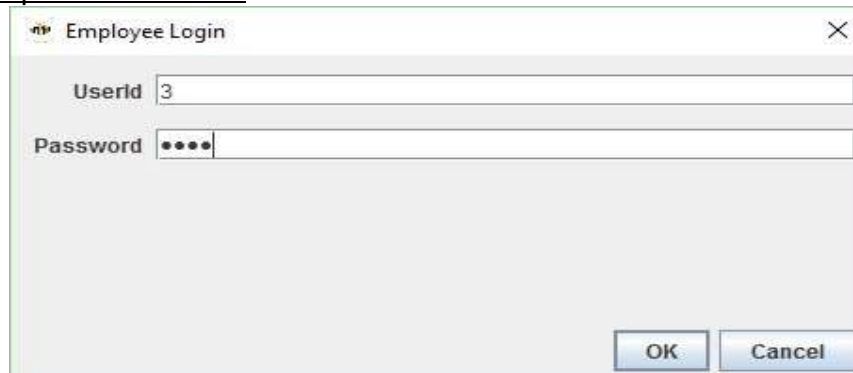
Destination Branch Id: Kgp

Order Date: 18/04/2016

Despatch Date: 18/04/2016

Close

In the next attempt we try to add a consignment at a branch through an employee at some other branch and it is rejected by the system as it should be. The sequence of screenshots explain it in details.



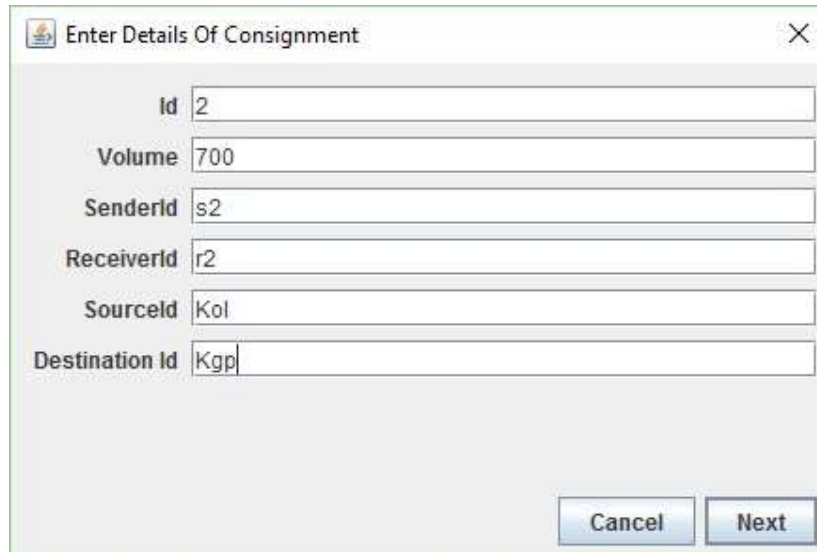
Employee Login

UserId 3

Password

OK Cancel

Here the employee with id 3 is in branch with ID Kgp but will try to enter consignment to branch Id Kol




A screenshot of a Windows-style dialog box titled "Enter Details Of Consignment". It contains six text input fields with labels to their left: "Id" (value: 2), "Volume" (value: 700), "SenderId" (value: s2), "ReceiverId" (value: r2), "SourceId" (value: Kol), and "Destination Id" (value: Kgp). At the bottom right, there are two buttons: "Cancel" and "Next".

On clicking Next button error message is displayed.




Now there is a fresh attempt which succeeds and is shown.

 Enter Details Of Consignment ✕


Id	2
Volume	700
SenderId	s2
ReceiverId	r2
SourceId	Kol
Destination Id	Kgp

Cancel Next

 Enter Sender Details ✕

Id	s2
Name	Martin
Address	Kolkata
Mobile No.	7412336987
Email Id	martin@gmail.com

Next

 Enter Receiver Details ✕

Id	r2
Name	Thomas
Address	Kharagpur
Mobile No.	74563214785
Email Id	thomas@gmail.com

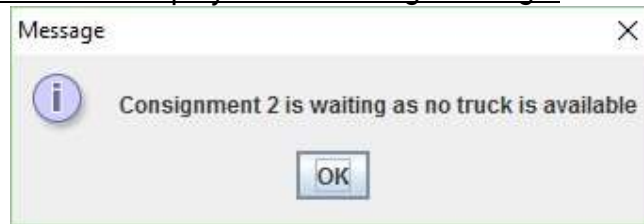
OK

✕ Message

 **Successfully inserted**

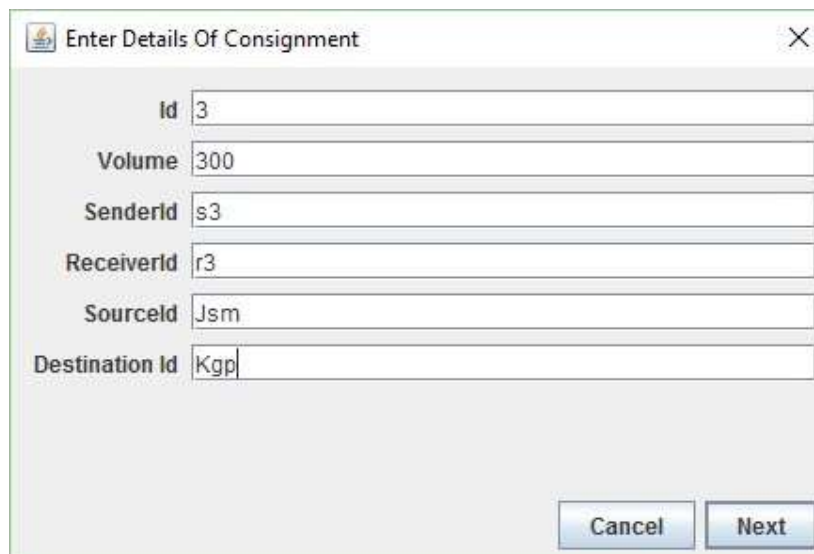
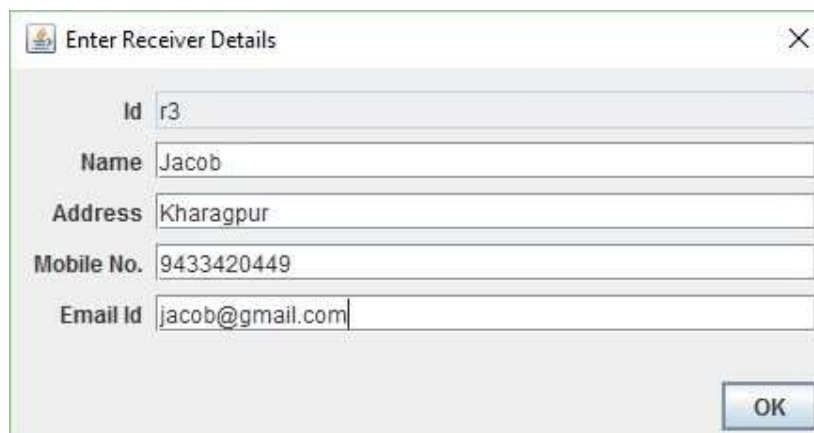
OK

Now the consignment is to be kept waiting since there are not enough trucks as per the requirements specifications we need 2 trucks to transport a volume of 700 cubic metres. The TCCS does the same and displays the following message.



We have incremented the system date by one in order to see whether the various waiting times are updated or not.

Now as the next step we add another consignment at Branch Id "Jsm". Since there are no trucks at "Jsm" yet, so in a similar argument the consignment is kept waiting. The following screen shots suggest the same and are in the same sequence as the sequence of events.

A screenshot of a form titled "Enter Details Of Consignment". It contains several input fields: "Id" with value "3", "Volume" with value "300", "SenderId" with value "s3", "ReceiverId" with value "r3", "SourceId" with value "Jsm", and "Destination Id" with value "Kgp". At the bottom right are "Cancel" and "Next" buttons.A screenshot of a form titled "Enter Receiver Details". It contains several input fields: "Id" with value "r3", "Name" with value "Jacob", "Address" with value "Kharagpur", "Mobile No." with value "9433420449", and "Email Id" with value "jacob@gmail.com". At the bottom right is an "OK" button.



The integrity of the TCCS is further tested.
Following are the updated databases.

Database of Trucks

truckNo	currentBranch	noOfCon	status	buyingdate	arrivedate	dispatchdate
Kol1	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol2	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol3	Kol	0	waiting in the branchid = Kol	2016-04-18	2016-04-18	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Consignments

consignmentId	volume	senderId	receiverId	revenue	isTruck	truckno	dispatchstatus	sourceId	destId	orderdate	despatchdate	requirednooftrucks
1	700	s1	r1	28000	1	,Kol1,Kol2	1	Kol	Kgp	2016-04-18	2016-04-18	2
2	700	s2	r2	28000	0	0	0	Kol	Kgp	2016-04-18	NULL	2
3	300	s3	r3	13500	0	0	0	Jsm	Kgp	2016-04-19	NULL	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Branch Offices

branchId	branchAddress	branchphone	numberoftrucks	numberofemployees	volumehandled	revenue	idlewaitingtime	rate	currentnumoftrucks	numberofwaitingcon
Jsm	Jamshedpur	4589614789	0	1	0	0	0	45	0	1
Kgp	Kharagpur	7412336981	2	1	700	28000	0	35	2	0
Kol	Kolkata	74563214789	3	2	0	0	0	40	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Customers

Id	Name	Address	Mobile	Email	ConsignmentId
r1	Garry	Kharagpur	7456321478	garry@gmail.com	1
r2	Thomas	Kharagpur	74563214785	thomas@gmail.com	2
r3	Jacob	Kharagpur	9433420449	jacob@gmail.com	3
s1	John	Kolkata	74569874123	john@gmail.com	1
s2	Martin	Kolkata	7412336987	martin@gmail.com	2
s3	Francis	Jamshedpur	78963214789	francis@gmail.com	3
NULL	NULL	NULL	NULL	NULL	NULL

Now we log in as manager and see whether the waiting times have been updated. The following screenshot are taken to support the same.



The screenshot below displays the idle waiting time of a truck in the branch with BranchId "Kgp".



The system date is incremented again.

And this time a consignment is despatched from Kharagpur to Kolkata by a single truck. This in turn triggers the despatching of consignment 2 which was kept waiting at Kolkata as two trucks were not available. Now it is shown as despatched. The following screenshots are in the support of the same.

A screenshot of a Windows-style form titled "Enter Details Of Consignment". The form has a close button (X) in the top right corner. It contains several text input fields with labels: "Id" (value: 4), "Volume" (value: 400), "SenderId" (value: s4), "ReceiverId" (value: r4), "SourceId" (value: Kgp), and "Destination Id" (value: Kol). At the bottom right, there are two buttons: "Cancel" and "Next".

Enter Sender Details

Id s4

Name Harish

Address Kharagpur

Mobile No. 7412354789

Email Id harish@gmail.com

Next

Enter Receiver Details

Id r4

Name Ajoy

Address Kolkata

Mobile No. 7412336987

Email Id ajoyk.mondal@gmail.com

OK

Message

i Successfully inserted

OK

Message

i Consignment 4 is successfully dispatched

OK

Message

i consignment 2 dispatched

OK

Again, we use the manager use cases to query the average waiting time of consignment and the idle waiting time of truck in the branch "Kol".



Database of Branch Office

[illegible]


consignmentid	volume	senderId	receiverId	revenue	isTruck	truckno	dispatchstatus	sourceId	destId	orderdate	despatchdate	requirednooftrucks
1	700	s1	r1	28000	1	,Kol1,Kol2	1	Kol	Kgp	2016-04-18	2016-04-18	2
2	700	s2	r2	28000	1	,Kol1,Kol3	1	Kol	Kgp	2016-04-18	2016-04-20	2
3	300	s3	r3	13500	0		0	Jsm	Kgp	2016-04-19	NULL	1
4	400	s4	r4	14000	1	,Kol1	1	Kgp	Kol	2016-04-20	2016-04-20	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Consignment

Database of Trucks

truckNo	currentBranch	noOfCon	status	buyingdate	arrivedate	dispatchdate
Kol1	Kgp	3	waiting in the branch Kgp	2016-04-18	2016-04-20	NULL
Kol2	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol3	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-20	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The manager then queries the consignment 3 and the following output is generated by the TCCS.


Display consignment status
✕

Consignment Id: 3

Volume : 300

Sender Name: Francis

Receiver Name: Jacob

Revenue 13500

Dispatch and Received: No

Truck No. Assigned:

Source Branch Id: Jsm

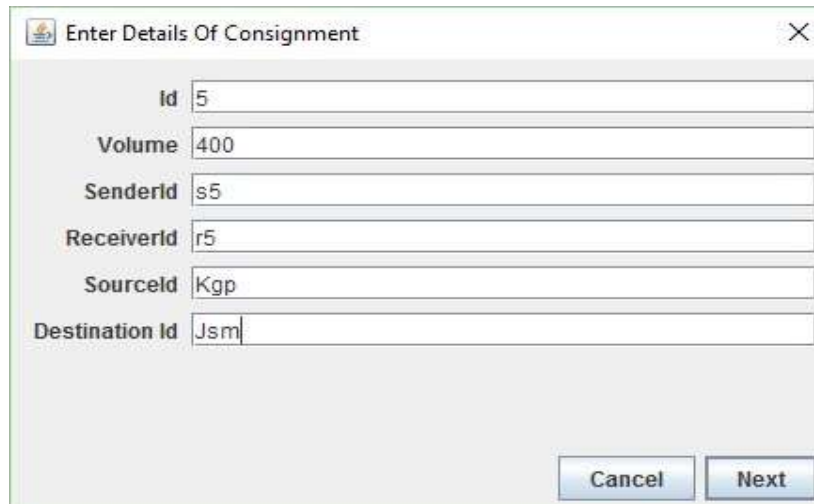
Destination Branch Id: Kgp

Order Date: 19/04/2016

Despatch Date: To be dispatched

Close

The system date is incremented once again and then a consignment is added to be despatched from Kharagpur to Jamshedpur.



A screenshot of a software dialog box titled "Enter Details Of Consignment". It contains several input fields with the following values: Id (5), Volume (400), SenderId (s5), ReceiverId (r5), SourceId (Kgp), and Destination Id (Jsm). At the bottom right, there are "Cancel" and "Next" buttons.

Field	Value
Id	5
Volume	400
SenderId	s5
ReceiverId	r5
SourceId	Kgp
Destination Id	Jsm



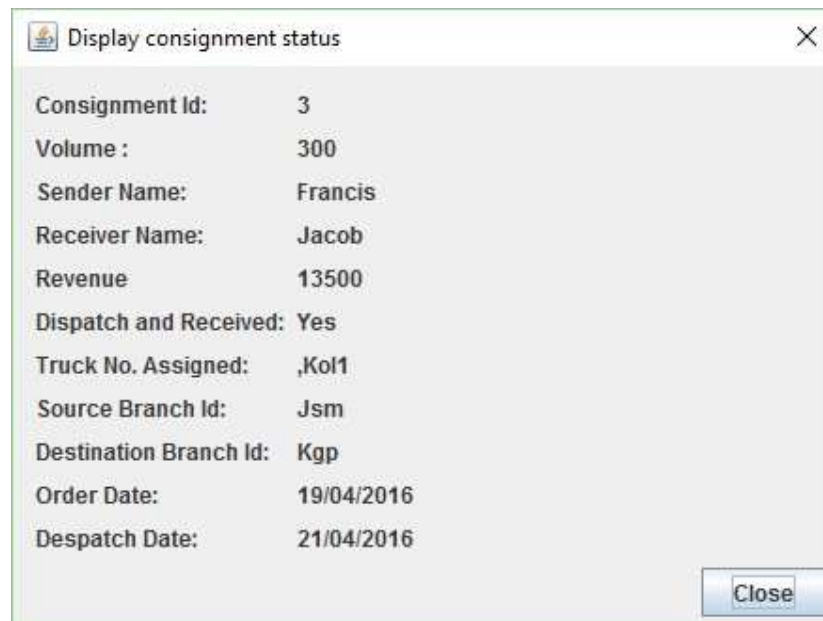
Due to availability of truck at Kharagpur the Consignment 5 is immediately despatched.



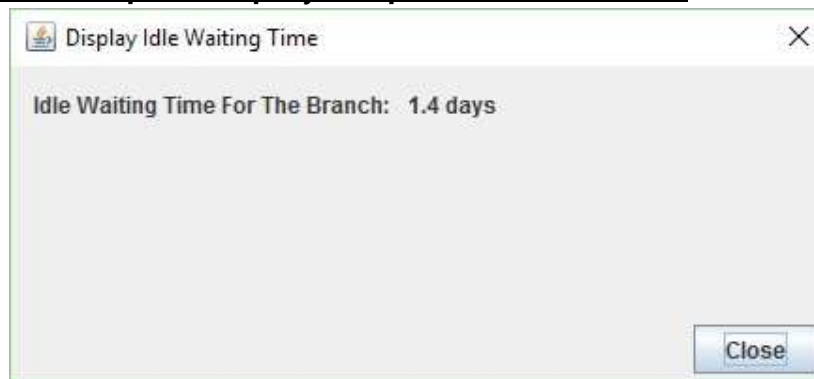
This truck on reaching Jamshedpur is given Consignment 3, which had been waiting so long and finally despatched.



The manager use case of Query Consignment is again used to Query the status of the Dispatched Consignment 3. The output generated is shown in the next screenshot.



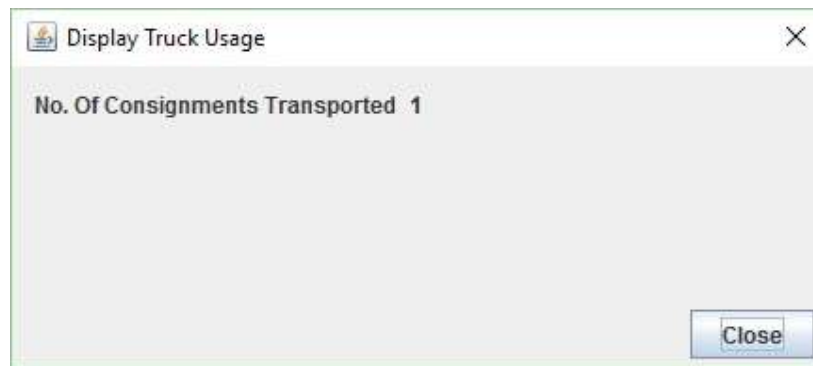
The idle waiting time of a truck at branch Kgp is queried the following output was generated by the Transport Company Computerization Software.



The next query was to find the average waiting period of consignments. The following message was displayed.

Finally, a Query was made on the usage of Truck No." Kol2". The TCCS generated the following output.





The final updated databases are being displayed below, all consignments are despatched and received.

Database of Branch Office

branchId	branchAddress	branchphone	numeroftucks	numerofoemployees	volumehandled	revenue	idlewaitingtime	rate	currentnumoftucks	numerofowaitingcon
Jsm	Jamshedpur	4589614789	1	1	400	14000	0	45	0	0
Kgp	Kharagpur	7412336981	5	1	1700	69500	3	35	3	0
Kol	Kolkata	74563214789	4	2	400	14000	2	40	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Consignment

consignmentId	volume	senderId	receiverId	revenue	isTruck	truckno	dispatchstatus	sourceId	destId	orderdate	despatchdate	requirednooftucks
1	700	s1	r1	28000	1	,Kol1,Kol2	1	Kol	Kgp	2016-04-18	2016-04-18	2
2	700	s2	r2	28000	1	,Kol1,Kol3	1	Kol	Kgp	2016-04-18	2016-04-20	2
3	300	s3	r3	13500	1	,Kol1	1	Jsm	Kgp	2016-04-19	2016-04-21	1
4	400	s4	r4	14000	1	,Kol1	1	Kgp	Kol	2016-04-20	2016-04-20	1
5	400	s5	r5	14000	1	,Kol1	1	Kgp	Jsm	2016-04-21	2016-04-21	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Database of Trucks

truckNo	currentBranch	noOfCon	status	buyingdate	arrivedate	dispatchdate
Kol1	Kgp	5	waiting in the branch Kgp	2016-04-18	2016-04-21	NULL
Kol2	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-18	NULL
Kol3	Kgp	1	waiting in the branch Kgp	2016-04-18	2016-04-20	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL