

# CCC Software Documentation

## Version 1.1

### April 22, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

**Software Requirements Specification  
Version 1.1  
March 6, 2011**

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

# Table of Contents

<b>Table of Contents .....</b>	
<b>1.0. Introduction .....</b>	2
1.1. Purpose.....	3
1.2. Intended Audience .....	3
1.3. Assumptions and Dependencies .....	3
1.4. An Overview .....	3
1.5. Additional Information .....	4
1.6. Contact Information .....	4
1.7. References .....	4
<b>2.0. DFD .....</b>	5-7
2.1. Context Diagram.....	5
2.1. DFD.....	6
2.1. Data Dictionary For DFD .....	7
<b>3.0. Functional requirements .....</b>	8-12
3.1. Order a Consignment .....	8
3.2. Transport Office Section .....	9-10
3.2.1.Issue Truck Driver Receipt .....	9
3.2.2.Update Truck Status .....	10
3.3Account Manager's Section .....	10-11
3.3.1.Generate Pay Slip .....	10
3.3.2.Maintain Firm's Account.....	11
3.4. Manager's Section .....	12-14
3.4.1.View Truck's Status .....	12
3.4.2.View Truck's Usage.....	12
3.4.3. View Consignment Details .....	13
3.4.4. View Performance.....	14
<b>4.0. Non-functional requirements .....</b>	15
4.1 Security requirements.....	15
4.2 Performance requirements .....	15

# 1.0. Introduction

## 1.1. Purpose

The main purpose of the software is to automate much of the task which was earlier done manually. Various accounts keeping functions keep track of company's progress and are very essential for a firm's Manager for improvising.

## 1.2. Intended Audience

The expected audience of this document are the courier companies who wish to computerize various book keeping activities associated with its daily operation.

## 1.3. Assumptions and Dependencies

- Six types of users are assumed. First, Manager which is single for the whole Firm. Second Clerk, Third Account Manager, both of them one for each Branch, and Truck Drivers, Customers and Employees of Firm.
- Transport Section and Booking Consignment to be handled by Clerk.
- Each Truck has a unique ID.

## 1.4. An Overview

The software will consist of a Main User Interface with four selections.

The first selection is to **Order a consignment**. This section is for user clerk's use. The entries on the receipt will be filled by a user clerk. The receipt will contain entries like volume, weight, destination address, sender address. This information will be retained on the Firm's Head Office Server. The software will compute the charge for the consignment and print a bill indicating a unique id indicating the consignment number assigned to the consignment. All payments and receipts are entered into the system.

The second selection is **Transport Office Section**. When the volume of consignments for any particular destination (branch office) becomes 500 cubic meters, the system will automatically allot the next available truck that is present at the branch office. The software will print the details of the consignment like number, volume, sender's name and address, and the receiver's name and address. When a truck reaches a branch office, its arrival status will be updated here. The fuel and repair charges for a truck will be entered here.

The third selection is **Account Manager's Section**. Regular expenses of the courier company including staff salaries, rental charges for the branch offices and truck maintenance charges will be maintained here. Company's profit/loss is maintained here. The software will assist

the Account's Manager in generating the pay slip of all the employees every month and automatically credit the salaries to their respective bank accounts.

The fourth selection is **Manager's Section**. It includes a profit-loss account (taking all the branches and the entire operation into account) to enable manager to view branch-wise revenue generated, consignments handled, expenses. The manager will be able to view the status of different trucks at any time like the branch office at which it is waiting. The manager will be able to view truck usage (overall as well as for individual trucks) over a given period of time. The truck usage will be given in terms of load factor (average capacity utilization) and number of Kilometres covered over the given period. The manager of the courier company will be able to query the status of any particular consignment and the details of volume of consignments transported between any two branches and the corresponding revenue generated.

The manager will be able to view the average waiting period for the consignments over a given period of time (day, month, or year) and that for various source destination pairs. The manager will also be able to see the average idle times of trucks over a given period time.

## 1.5. Additional Information

The remainder of this document is two chapters, the first providing a full description of the project for the owners of the CCC. It lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full for the software developers' assistance.

These two sections are cross-referenced by topic; to increase understanding by both groups involved. Also for better understanding a DFD (Data Flow Diagram) has been provided.

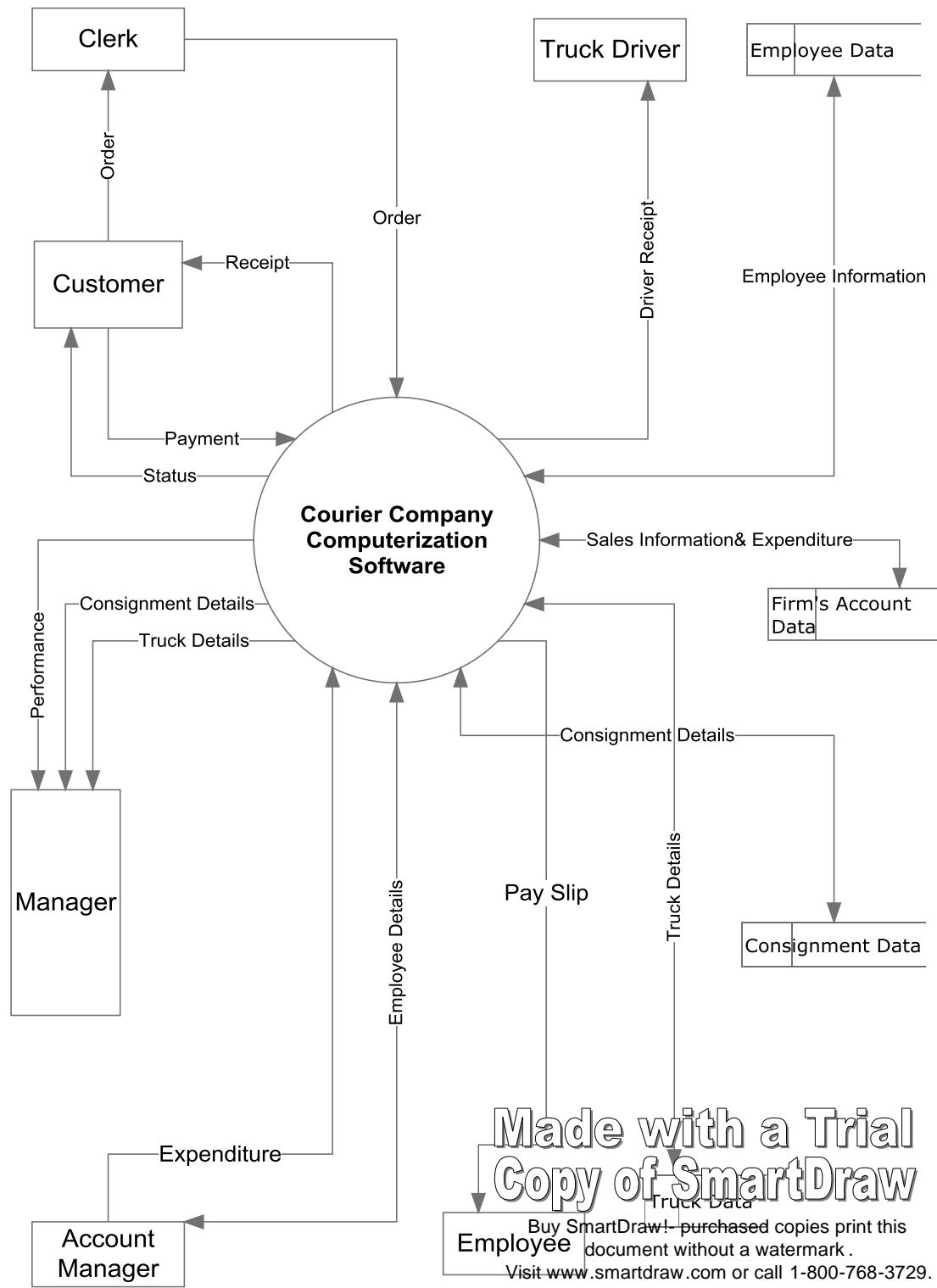
## 1.6. Contact Information

Team Member Pranjal Sahu  
Roll No. 09CS1036  
ID 36  
Phone No. 9476481262

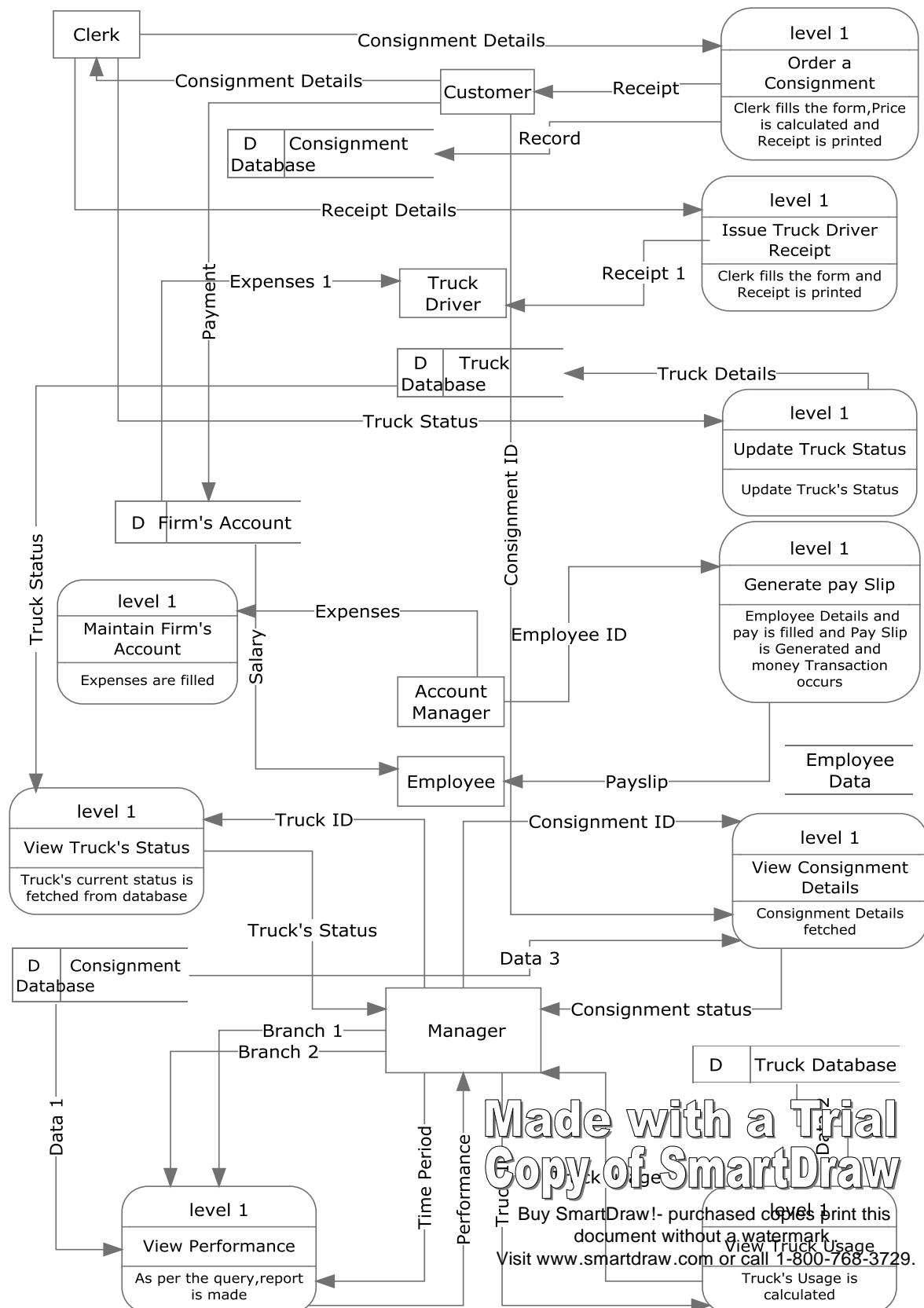
## 1.7. References

1. [IEEE] the applicable IEEE standards are published in "IEEE Standards Collection," 2001 edition.
2. Writing Software Requirements Specifications by *Donn Le Vie, Jr.*
3. An Integrated Approach to Software Engineering by Pankaj Jalote.

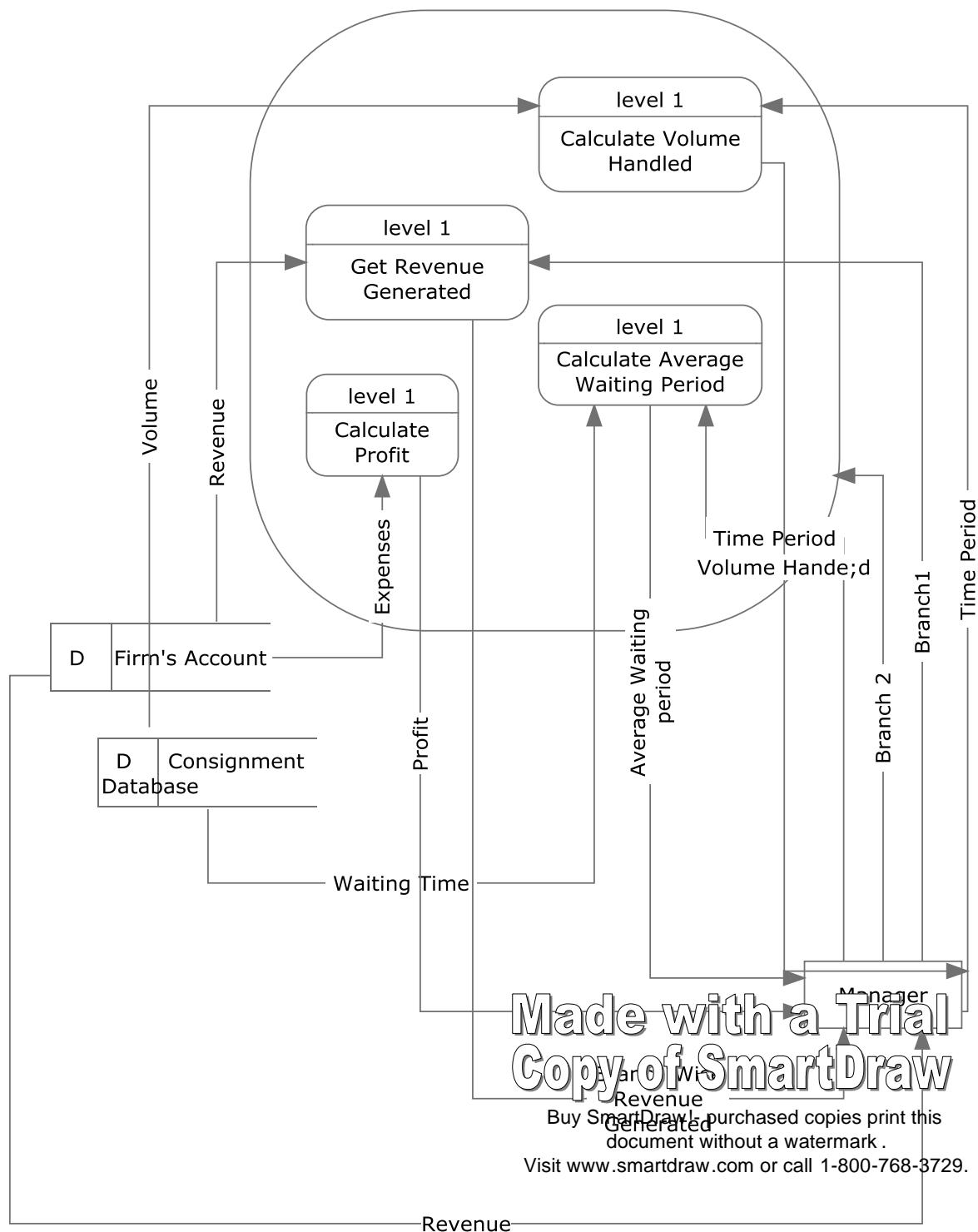
## 2.0. DFD



Context Diagram for the Courier Company



## **View Performance Level 2**



## Data Dictionary for DFD:

<b>Consignment Details</b>	Refer 3.1.1 ----- <b>Consignment Details</b> .
<b>Data 1</b>	Refer 3.3.3----- <b>Show Status +Show Details</b> .
<b>Data 2</b>	Refer 3.3.3----- <b>Show Status +Show Details</b> .
<b>Data 3</b>	Refer 3.4.2----- <b>Truck's Usage</b> .
<b>Receipt Details</b>	Refer 3.2.1 ----- <b>Receipt Details</b> .
<b>Receipt 1</b>	Refer 3.2.1 ----- <b>Receipt Details</b> .
<b>Record</b>	Refer 3.1.1 ----- <b>Receipt Details</b> .
<b>Receipt</b>	Refer 3.1.1 ----- <b>Receipt Details</b> .
<b>Payment</b>	Money (in exchange of service).
<b>Salary</b>	Money (in exchange of service).
<b>Pay Slip</b>	Refer 3.3.1 <b>Employees Record</b> .
<b>Expenses 1</b>	Fuel and Repair Charges.
<b>Truck Status</b>	Refer 3.4.1 ----- <b>Truck's Status</b> .
<b>Truck's Status</b>	Refer 3.4.1 ----- <b>Truck's Status</b> .
<b>Consignment Status</b>	Refer 3.4.3 ----- <b>Show Status</b> .
<b>Truck ID</b>	Unique ID of Each Truck.
<b>Consignment ID</b>	Unique ID of Each Consignment.
<b>Employee ID</b>	Unique ID of Each Employee of each Branch.
<b>Performance</b>	Refer 3.4.4----- <b>Explanation</b>
<b>Branch 1</b>	Branch Name.
<b>Branch 2</b>	Branch Name.
<b>Time Period</b>	Can be in Day, Month Or Year.

NOTE-In cases of any discrepancy please refer 1.4 An Overview.

## 3.0. Functional Requirements

### 3.1. Order a Consignment

#### 3.1.1 Order a Consignment

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	Computer is connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Clerk selects the “Order a Consignment” option from the main UI.</li> <li>2. The software presents the clerk with a receipt form.</li> <li>3. The Clerk fills in the form and click submit.</li> <li>4. The software checks to see if all required fields are not empty.</li> <li>5. A receipt is printed with <b>Receipt details</b> containing the price calculated by <b>Formula 1</b>.</li> </ol>
<b>Alternate Path</b>	N/A
<b>Post condition</b>	<ul style="list-style-type: none"> <li>• The user is again on the Main UI.</li> <li>• A record is created in the Consignment ID table of the Consignment Database.</li> </ul>
<b>Exception Path</b>	<ul style="list-style-type: none"> <li>• If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.</li> <li>• If any of the required fields are empty, the software returns a message and returns the clerk to the receipt form.</li> </ul>
<b>Explanation</b>	<p><b>Consignment Details :</b></p> <ol style="list-style-type: none"> <li>1. Weight</li> <li>2. Volume(measured in cubic meters).</li> <li>3. Order Date.</li> <li>4. Receiving Date. (To be automatically filled when the consignment reaches).</li> <li>5. Sending Branch Office.</li> <li>6. Receiving Branch Office.</li> <li>7. Receiver's name and address.</li> <li>8. Sender's name and address.</li> </ol> <p><b>Formula 1:</b></p> <ol style="list-style-type: none"> <li>1. Base=Rs. 5,000/- per cubic meter (for distances till 500km).</li> <li>2. If Distance is greater than 500 then-  <math display="block">(10/100) * (\text{per extra } 100\text{km}) + \text{Base.}</math> </li> <li>3. For small articles and letters, Rs.50/- per 100gms.</li> <li>4. If weight is more than 100kg per cubic meter, then –  <math display="block">(10/100) * (\text{per extra } 20 \text{ kg}) + \text{Base.}</math> </li> </ol> <p><b>Receipt details:</b></p> <ol style="list-style-type: none"> <li>1. <b>Consignment Details.</b></li> <li>2. Unique ID number.</li> <li>3. Price by <b>Formula 1</b>.</li> </ol>

## 3.2. Transport Office Section

### 3.2.1 Issue Truck Driver Receipt

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The Computer must be connected to the Internet and to the Firm's Head Office Server.</li> <li>2. Truck allotted signal is positive.</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Clerk selects the "Transport Office Section" option from the main UI.</li> <li>2. The software displays a new window form.</li> <li>3. The Clerk clicks the "Issue Truck DriverReceipt" option from the new window displayed.</li> <li>4. The software displays a new Driver Receipt form.</li> <li>5. The Clerk fills in the form and clicks submit.</li> <li>6. The software checks to see if all required fields are not empty.</li> <li>7. If the required fields are not empty, a receipt is printed with <b>Receipt details</b>.</li> <li>8. If any of the required fields are empty, the software returns a message and returns the Clerk to the form window.</li> </ol>
<b>Alternate Path</b>	N/A
<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• A record is created in the Truck's Dispatch.</li> <li>• The user is again on "Transport Office Section" window.</li> </ul>
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.
<b>Explanation</b>	<p><b>Receipt Details:</b></p> <ol style="list-style-type: none"> <li>1. All <b>Consignment Details</b> it is carrying. (Refer <b>Consignment Details</b> of 3.1.1.)</li> <li>2. Total volume of all consignments.</li> <li>3. Fuel and Repair Charges.</li> </ol>

### 3.2.2 Update Truck Status

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	The Computer must be connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Clerk selects the "Transport Office Section" option from the main UI.</li> <li>2. The software displays a Transport Office Section window.</li> <li>3. The Clerk clicks the "Update Truck Status".</li> <li>4. Clerk enters the ID of the truck and submits.</li> <li>5. <b>Truck's record</b> is shown.</li> <li>6. Clerk will update the <b>Truck's record</b>.</li> <li>7. Confirm message is displayed.</li> <li>8. If yes is selected then the record of that truck is updated.</li> <li>9. Else If No is selected then Truck's record is again returned.</li> </ol>
<b>Alternate Path</b>	N/A

<b>Post-condition</b>	<ul style="list-style-type: none"> <li>• <b>Truck's record</b> is updated.</li> <li>• The user is again on “Transport Office Section” window.</li> </ul>
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.
<b>Explanation</b>	<p><b>Truck's record :</b></p> <ol style="list-style-type: none"> <li>1. Total volume transported.</li> <li>2. Presently transporting OR not transporting.</li> <li>3. If currently transporting- <ul style="list-style-type: none"> <li>• Names of two Branches between which it is transporting. <ul style="list-style-type: none"> <li>• Date of Start of Journey.(To be filled when the truck has been dispatched from a branch).</li> <li>• Arrival Date. (To be filled when the truck has arrived at a branch).</li> <li>• Kilometres to be travelled. (To be filled when the truck has been dispatched from a branch).(Equals to the distance between the two branches).</li> </ul> </li> </ul> </li> </ol>

### 3.3. Account Manager's Section

#### 3.3.1 Generate Pay Slip

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	Computer is connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Account manager selects the “Account Manager's Section” option from the main UI.</li> <li>2. Username and Password field is provided.</li> <li>3. On successful logging in a new Account's Section window is opened.</li> <li>4. Account manager enters the employees name and clicks enter.</li> <li>5. If employee's name is present in Employees database then that <b>Employees Record</b> is shown.</li> <li>6. Account Manager enters the <b>Details</b> and clicks Generate Pay Slip.</li> </ol>
<b>Alternate Path</b>	N/A
<b>Post condition</b>	<ul style="list-style-type: none"> <li>• Pay Slip is generated.</li> <li>• On successful operation the software credits the salaries in that employees' bank account and from the firm's account.</li> </ul>
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given
<b>Explanation</b>	<b>Employees record :</b>

	<ol style="list-style-type: none"> <li>1. Name of Employee.</li> <li>2. Unique ID number of Employee.</li> <li>3. Address of Employee.</li> <li>4. Telephone number of Employee.</li> <li>5. Basic Pay.</li> </ol> <p><b>Details:</b></p> <ol style="list-style-type: none"> <li>1. Employee's record.</li> </ol>
--	---

### 3.3.2 Maintain Firm's account

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	Computer is connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Follow 1-3 steps from Basic Path of 3.3.1.</li> <li>2. Account manager can update <b>Firm's account</b>.</li> </ol>
<b>Alternate Path</b>	N/A
<b>Post condition</b>	<ul style="list-style-type: none"> <li>• The user is again on the Main UI.</li> <li>• <b>Firm's account</b> is updated.</li> </ul>
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given
<b>Explanation</b>	<p><b>Firm's Account :</b></p> <ol style="list-style-type: none"> <li>1. Rent charges of office. (Of that branch).</li> <li>2. Staff Salaries. (Of that branch).</li> <li>3. Expenditure on Trucks. (Equals to Total Fuel and repair charges of trucks, common for all branches).</li> <li>4. Total revenue generated. (Equals to the sum of all price of all the <b>Receipt</b> issued from that office).</li> </ol>
<b>Reference</b>	<ul style="list-style-type: none"> <li>• For <b>Receipt</b> refer to Receipt Details of 3.1.1.</li> </ul>

## 3.4. Manager's Section

### 3.4.1 View Truck's Status

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	The Computer must be connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>Manager selects the “Manager’s Section” option from the main UI.</li> <li>Username and Password field is provided.</li> <li>On successful logging in a new Manager’s Section window is opened.</li> <li>Manager selects the “View Truck’s Status” option.</li> <li>New window is displayed.</li> <li>Manager selects the ID of the truck and clicks “Show Status”.</li> </ol>
<b>Alternate Path</b>	N/A
<b>Post-condition</b>	<b>Truck’s status</b> is shown.
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.
<b>Explanation</b>	<p><b>Truck’s status:</b></p> <ol style="list-style-type: none"> <li>If currently transporting then the two branches will be mentioned.</li> <li>If currently not transporting then the branch at which it is waiting is shown.</li> </ol>

### 3.4.2. View Truck’s Usage

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	The Computer must be connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>Follow 1-3 points in Basic Path of 2.4.1.</li> <li>Manager selects the “View Truck’s Usage” option.</li> <li>New window is displayed.</li> <li>(A) Manager selects the ID of the truck and selects the time interval and clicks “Show Usage”. <b>Truck’sUsage</b> is shown. (B) If Manager selects “Show overall usage” then <b>Total usage of trucks</b> is displayed.</li> </ol>
<b>Alternate Path</b>	N/A

<b>Post-condition</b>	<b>Truck's Usage</b> is shown.
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.
<b>Explanation</b>	<p><b>Truck's Usage:</b></p> <ol style="list-style-type: none"> <li>1. <b>Load factor</b> (average capacity utilization) i.e. (total weight carried in given time)/(given time)</li> <li>2. No. of kilometre travelled in the given period.</li> </ol> <p><b>Total Usage:</b></p> <ol style="list-style-type: none"> <li>1. Average <b>load factor</b> of all trucks in the given period.</li> <li>2. Average kilometres travelled by all trucks in the given period.</li> </ol>

### 3.4.3. View Consignment Details

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	The Computer must be connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Follow 1-3 points in Basic Path of 2.4.1.</li> <li>2. <b>(A)</b> Manager selects the “View Consignment Details” option.             <ol style="list-style-type: none"> <li>(i) New window is displayed.</li> <li>(ii) Manager selects the ID of the consignment clicks “<b>Show Details</b>”.</li> </ol> <b>(B)</b> Manager selects the “View Average Waiting Period” option.             <ol style="list-style-type: none"> <li>(i) New window is displayed.</li> <li>(ii) Manager selects “Between Branches” option.</li> <li>(iii) Manager selects the two branches.</li> <li>(iv) Manager enters <b>Time Period</b> and clicks “<b>Show Details</b>”.</li> </ol> </li> </ol>
<b>Alternate Path</b>	<p>Customer enters the Companies site and clicks “View Consignment Status”.</p> <p>New window is displayed.</p> <p>Customer enters the ID of the consignment clicks “<b>Show Status</b>”.</p>
<b>Post-condition</b>	<ol style="list-style-type: none"> <li>(I) ID's Consignment details are shown.</li> <li>(II) Average Waiting Period.</li> </ol>
<b>Exception Path</b>	If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.
<b>Explanation</b>	<p><b>Show Details:</b></p> <ol style="list-style-type: none"> <li>1. Show Consignment's Detail.</li> </ol>

	<p>2. Revenue Generated.</p> <p><b>Show Status:</b></p> <p>Consignment Status is shown.</p>
--	---

### 3.4.4. View Performance

<b>Priority</b>	Essential
<b>Trigger</b>	Menu selection
<b>Precondition</b>	The Computer must be connected to the Internet and to the Firm's Head Office Server.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>Follow 1-3 points in Basic Path of 3.4.1.</li> <li>Manager selects the "View Performance" option.</li> <li>New window is displayed.</li> <li>Manager selects "Overall Profit". <b>Profit</b> of whole Firm is shown.</li> <li>Manager selects the "View Average Waiting Period" option. <ul style="list-style-type: none"> <li>(i) New window is displayed.</li> <li>(ii) Manager selects "Between Branches" option.</li> <li>(iii) Manager selects the two branches.</li> <li>(iv) Manager enters "<b>Time Period</b>" and clicks "<b>Show Details</b>".</li> </ul> </li> <li>Manager selects "Branch Wise Performance". <ul style="list-style-type: none"> <li>(I) Manager selects the "Revenue Generated" and enters the two branches. Manager enters "<b>Time period</b>". (Revenue generated between the two branches is shown).</li> <li>(II) Manager selects the "Expenses" and enters the branch. Manager enters "<b>Time period</b>". (Expenses of that branch are shown).</li> <li>(III) Manager selects "View Volume Handled" option and enters "<b>Time Period</b>". Manager selects the two branches and clicks "<b>Show</b>".</li> </ul> </li> </ol>
<b>Alternate Path</b>	N/A
<b>Post-condition</b>	As per Manager's selection, required data is shown.
<b>Exception Path</b>	<ul style="list-style-type: none"> <li>In case of any field not entered message to be delivered and redirected to present window.</li> <li>If there is a connection failure, message of failure is delivered. Message to repeat the process is also given.</li> </ul>
<b>Explanation</b>	<p><b>Profit</b>= Total revenue generated-Total expenses.</p> <p><b>Show Details:</b> Show the <b>Average Waiting Period</b> in that given period and for that branch pairs is shown.</p> <p><b>Time period:</b> Can be in Day, Month OR Year.</p>

**Show:** Show Volume of consignments transported between the two branches.

**Average Waiting Period = (Waiting time)/Time period.**

**Waiting time = (Average time duration of delivery all consignments).**

## 4.0. Non-Functional Requirements

### 4.1. Security Requirements

1. Account Manager and Manager to be provided with Log In option.
2. Account related Data to be kept secure.

### 4.2 .Performance Requirements

1. Software to be able to run on a desktop with Pentium processor and 128 MB Ram.

Functional Oriented Design  
Version 1.1  
March 2, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>1.0. Purpose .....</b>	<b>2</b>
<b>2.0. Architecture views .....</b>	<b>3-4</b>
2.1. Element Catalog .....	3
2.2. Architecture Rationale.....	3
2.3. Diagrammatic Representation .....	4
<b>3.0. Structured Chart .....</b>	<b>5-</b>
3.1.1. First Level Factoring .....	5
3.1.1.1 Interfaces .....	5
3.1.2. Second Level Factoring.....	6-10
3.1.2.1 Interfaces .....	10
3.1.3. Third Level Factoring .....	11-14
3.1.3.1 Interfaces .....	14
<b>4.0. Assessment .....</b>	<b>15</b>

## 1.0. Purpose

---

After the creation of SRS the main purpose of this document is to provide the development team with the necessary details to carry out the project successfully. The document comprises of software architecture which determines how the client, software and the external world are interacting. Secondly the Structured charts provide the details of functions to be used with the required interfaces and finally it gives and self assessment of the design on the basis of goodness criteria.

## 2.0. Architecture Views

---

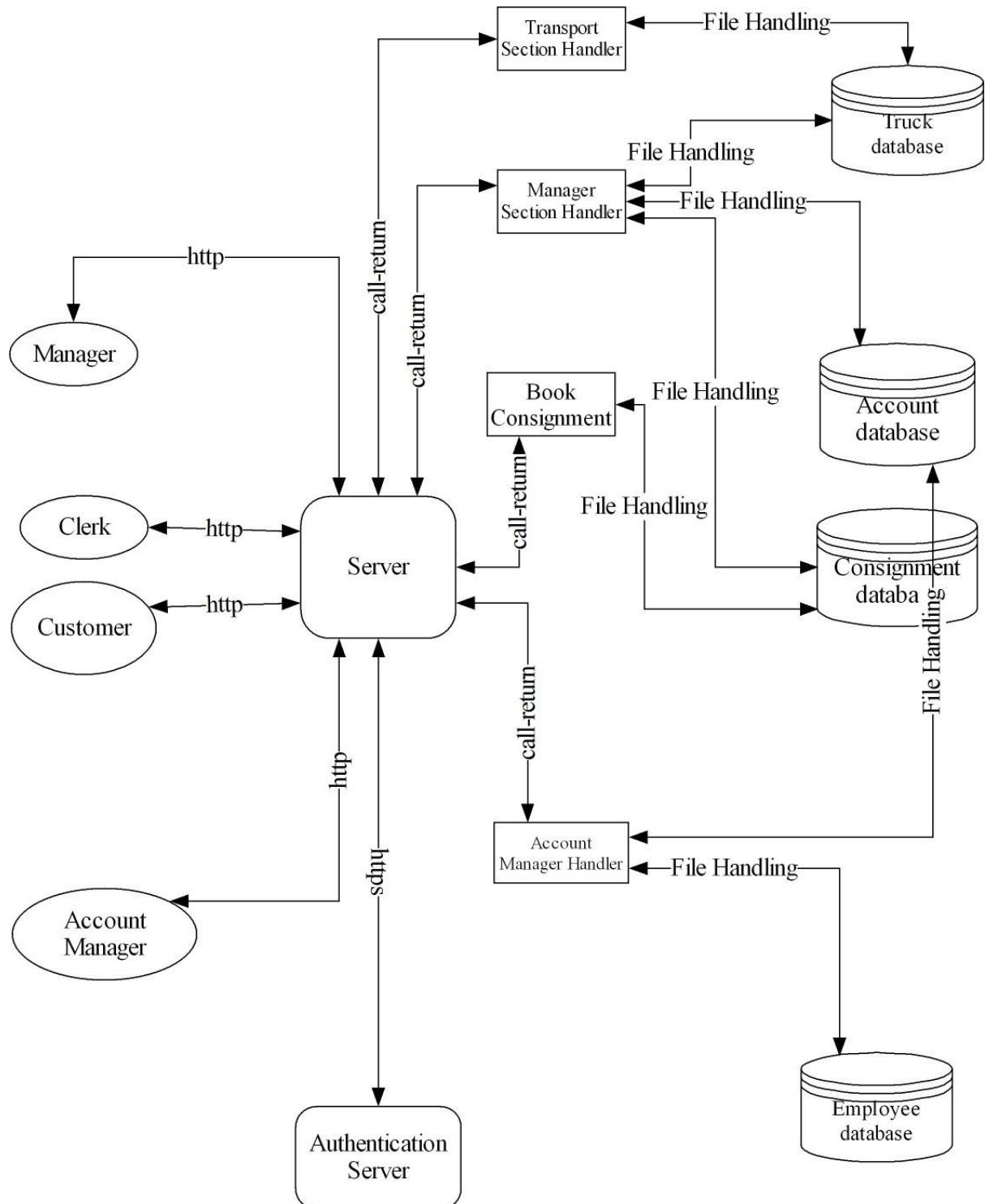
### 2.1 Element Catalog:

1. Database:
  - i. Account Database:  
Takes record of revenues, expenses, profit.
  - ii. Truck Database:  
Takes every Truck's record
  - iii. Employee Database:  
Takes record each branch's every employee.
  - iv. Consignment Database:  
Takes record of each consignment.
2. Clients:
  - i. Manager
  - ii. Clerk
  - iii. Account Manager
  - iv. Customer
3. Connectors:
  - i. Call-return
  - ii. http
  - iii. https
  - iv. file handling
4. Servers:
  - i. Main Server
  - ii. Authentication Server
5. Applications:
  - i. Transport Section Handler
  - ii. Manager Section Handler
  - iii. Account Manager section Handler
  - iv. Book Consignment.

### 2.2 Architecture Rationale

As the problem's analysis suggests, various clients are interacting with the system and modifying various components. Therefore it is very similar to Shared Architecture, with the central repository containing all the four Databases.

## 2.3 Diagrammatic Representation

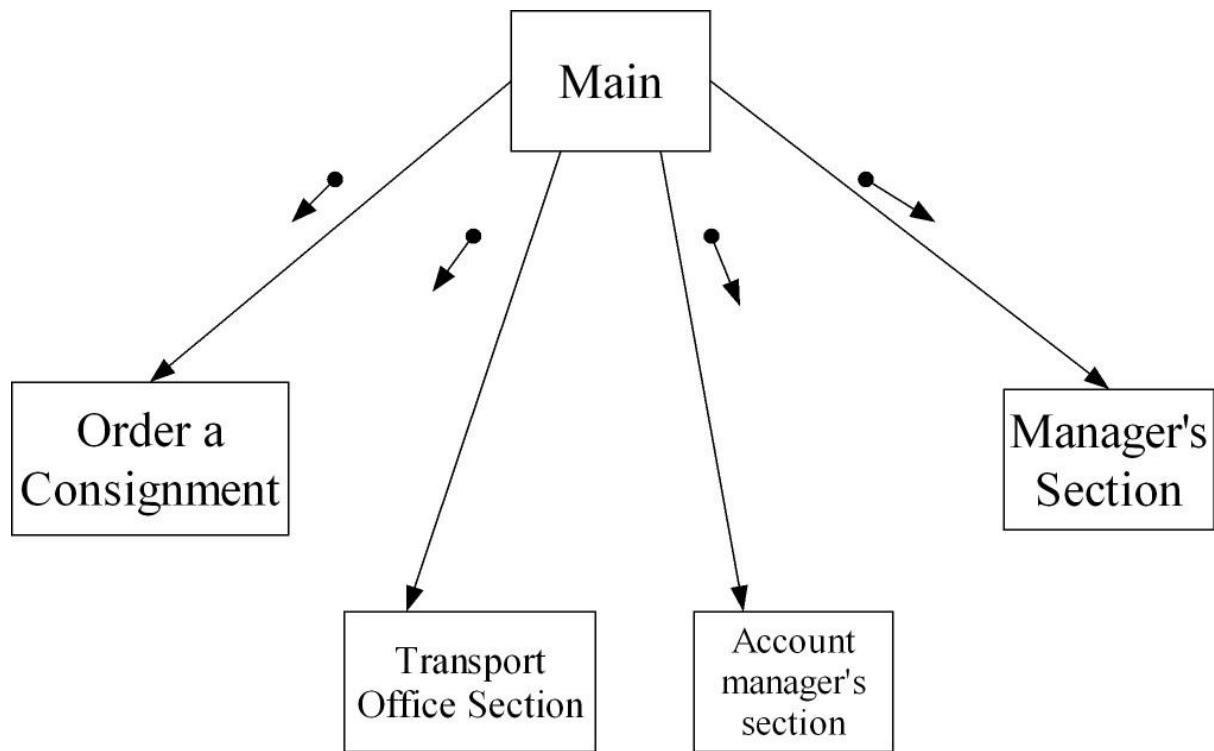


Component And Connector View of CCC software

## 3.0. Structured Chart

---

### 3.1.1 First Level Factoring



#### 1.1 First Level Factoring of CCC

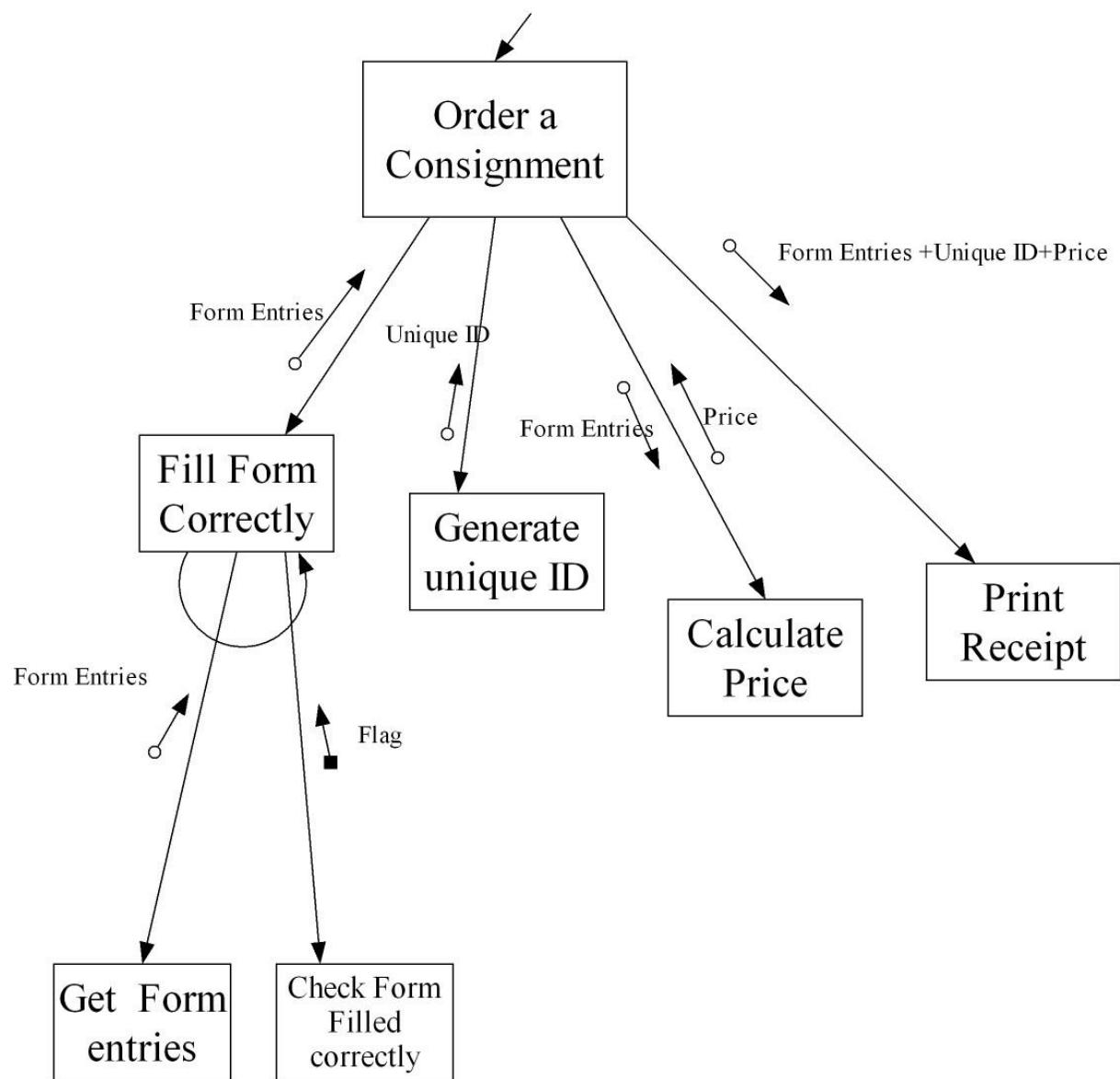
Fig.2

Interfaces (Level 1 factoring)(Fig.2):

- All the modules are initiated by a flag indicating selection.
- Above shown modules are only control modules managing the required operation to be done.

### 3.1.2 Second Level Factoring

#### 1.2 Second Level factoring of CCC software



**First Level Factoring of Order a Consignment**

Fig.3

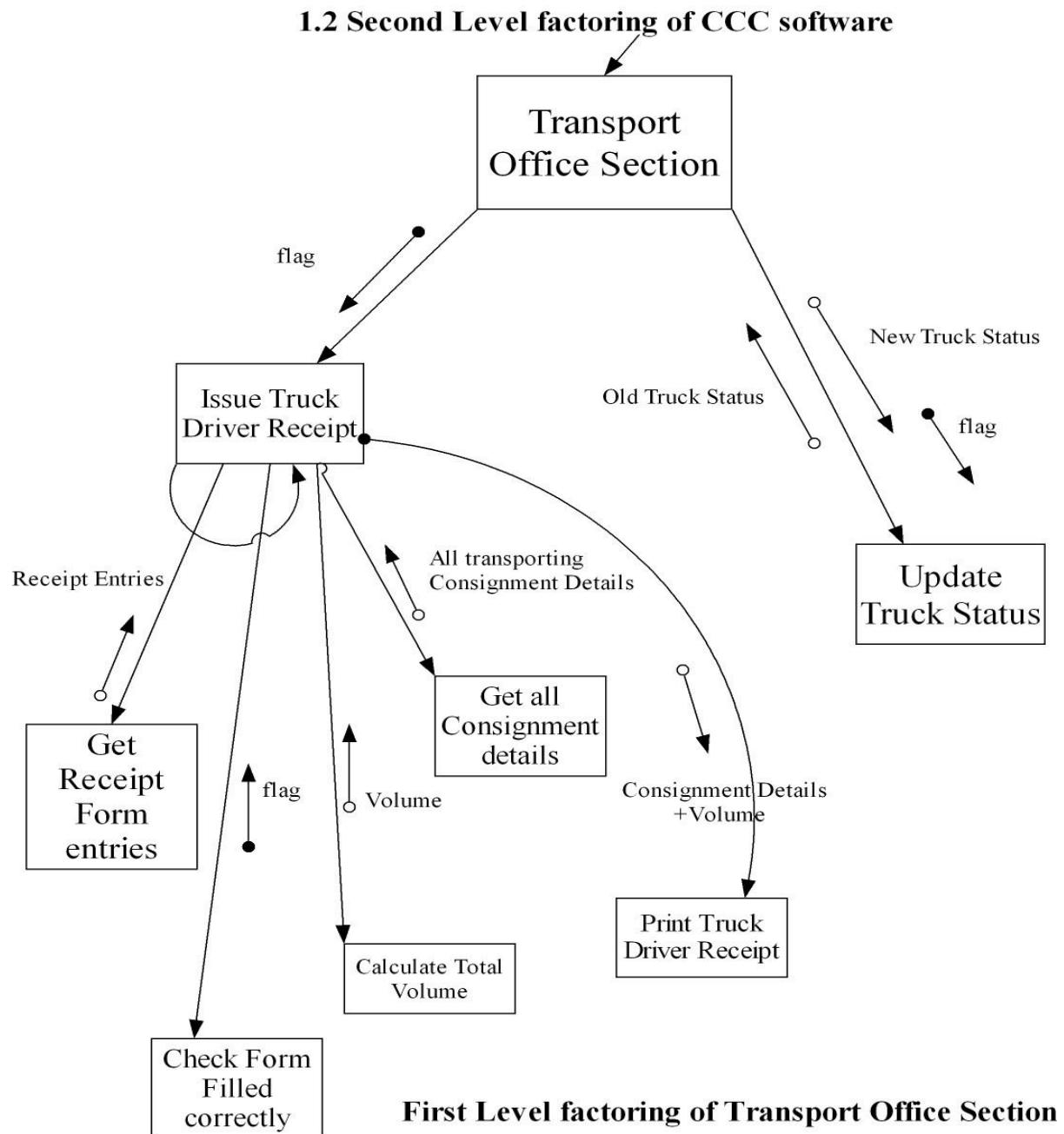
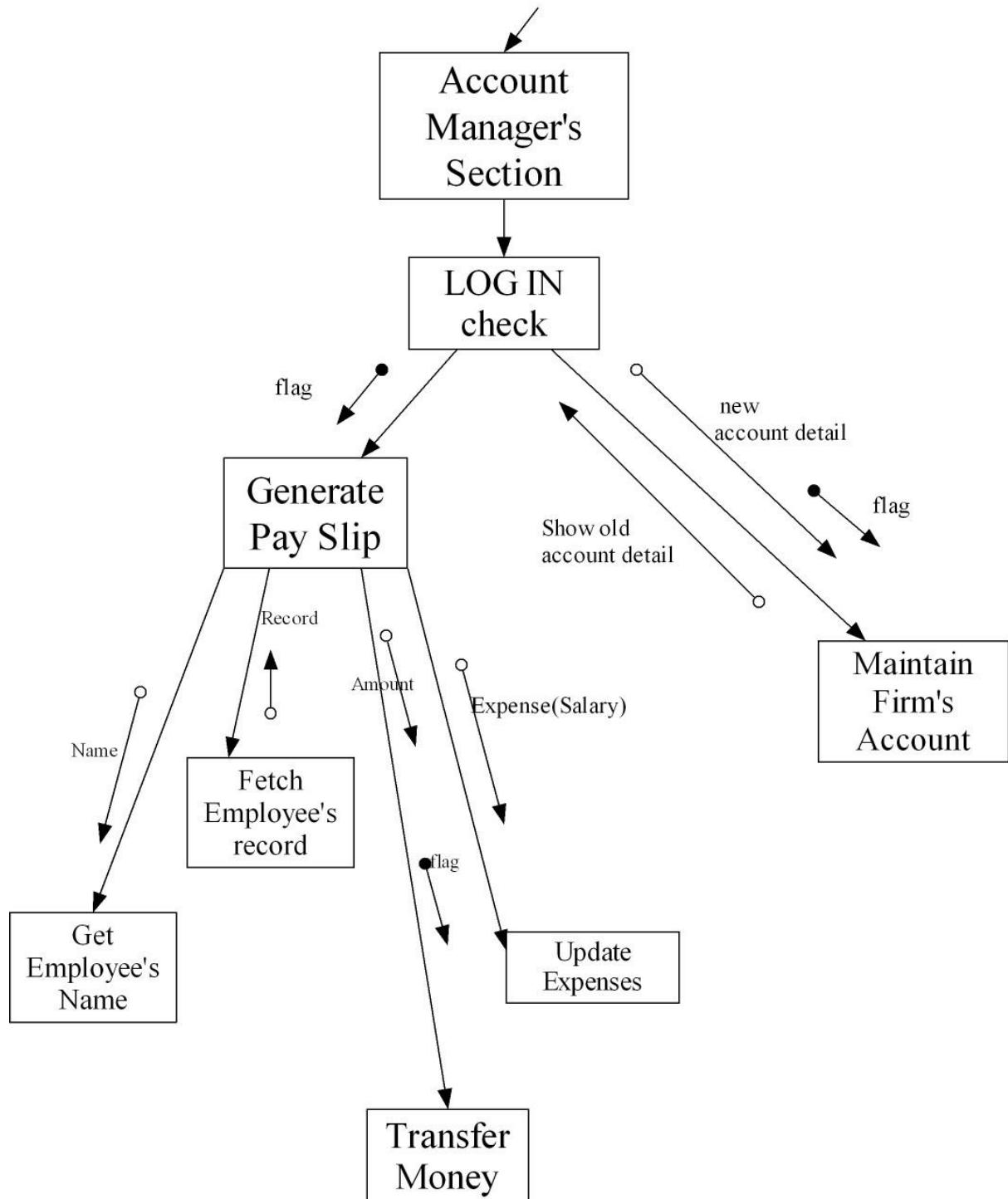


Fig.4

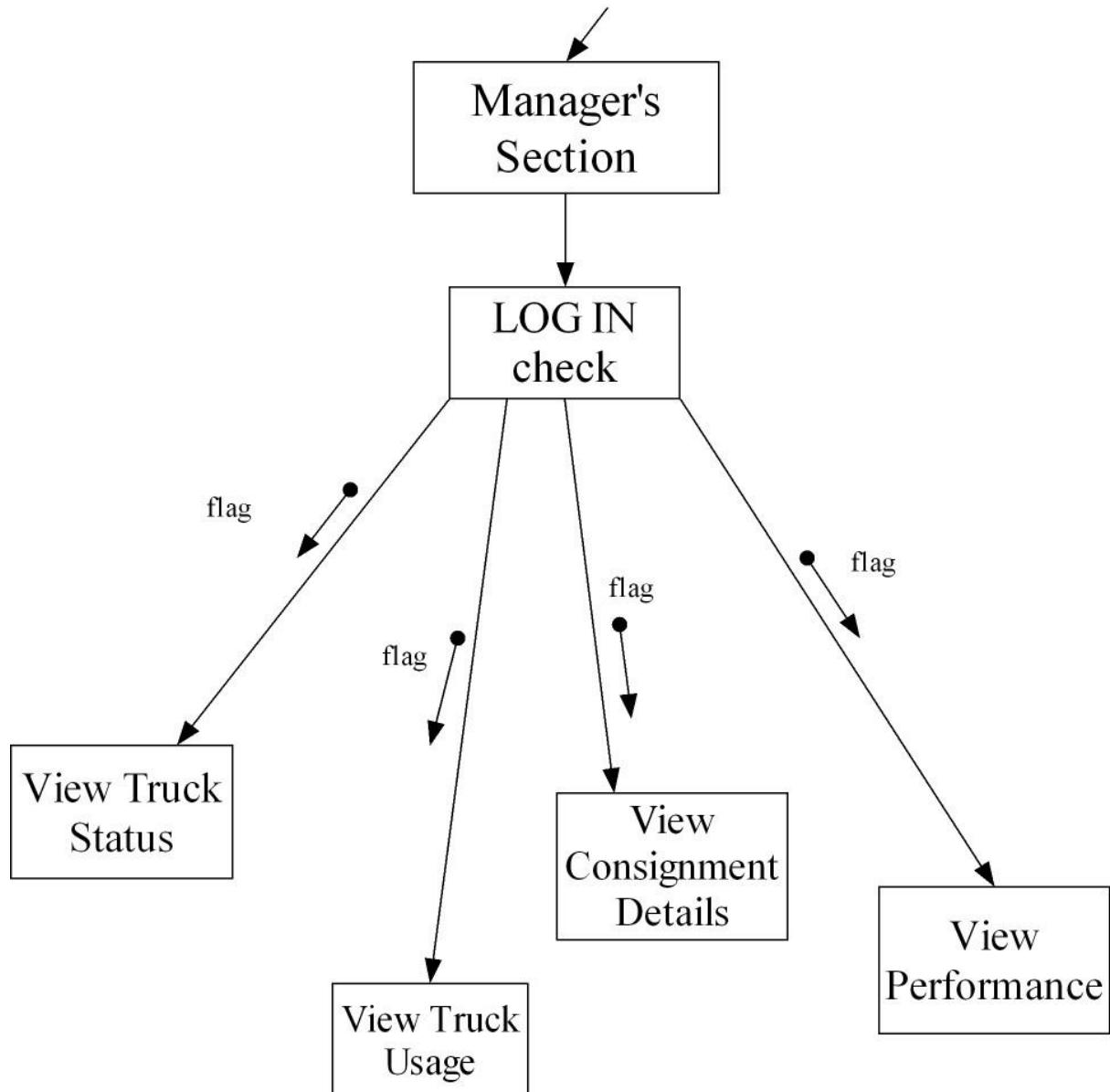
## 1.2 Second Level factoring of CCC software



First Level Factoring of Transport Office Section

Fig.5

## 1.2 Second Level factoring of CCC software



**First Level Factoring of Manager's Office Section**

Fig.6

## **INTERFACES (Level 2 factoring):**

### **Interfaces for ORDER A CONSIGNMENT (Fig.3):**

1. Form Entries:
  - i. String: Sending Branch Office, Receiving Branch Office, Order date, Receiving date, Receiver's name and address, Sender's name and address.
  - ii. Float: Volume, Weight.
2. Integer: Price, Unique ID

### **Interfaces for TRANSPORT OFFICE SECTION (Fig.4):**

1. Receipt Entries = Consignment Details:
  - i. Same as Form Entries
  - ii. Float: Volume
  - iii. Integer: Fuel and Repair Charges.
2. Old Truck Status = New Truck Status:
  - i. String: Sender Branch Name, Receiver branch Name, Current Branch Name

### **Interfaces for ACCOUNT MANAGER'S SECTION (Fig.5):**

1. Integer : Amount, Salary
2. String: Name
3. Record =Old Account Detail = New Account Detail:
  - i. String: Name of Employee, Address
  - ii. Integer: Unique ID, Telephone No., Basic pay

### **Interfaces for MANAGER'S SECTION (Fig.6):**

1. Integer : Amount, Salary
  2. String: Name
  3. Record =Old Account Detail = New Account Detail:
  4. String: Name of Employee, Address
- Integer: Unique ID, Telephone No., Basic pay

### 3.1.3 Third Level Factoring

#### 1.3 Third Level factoring of CCC software

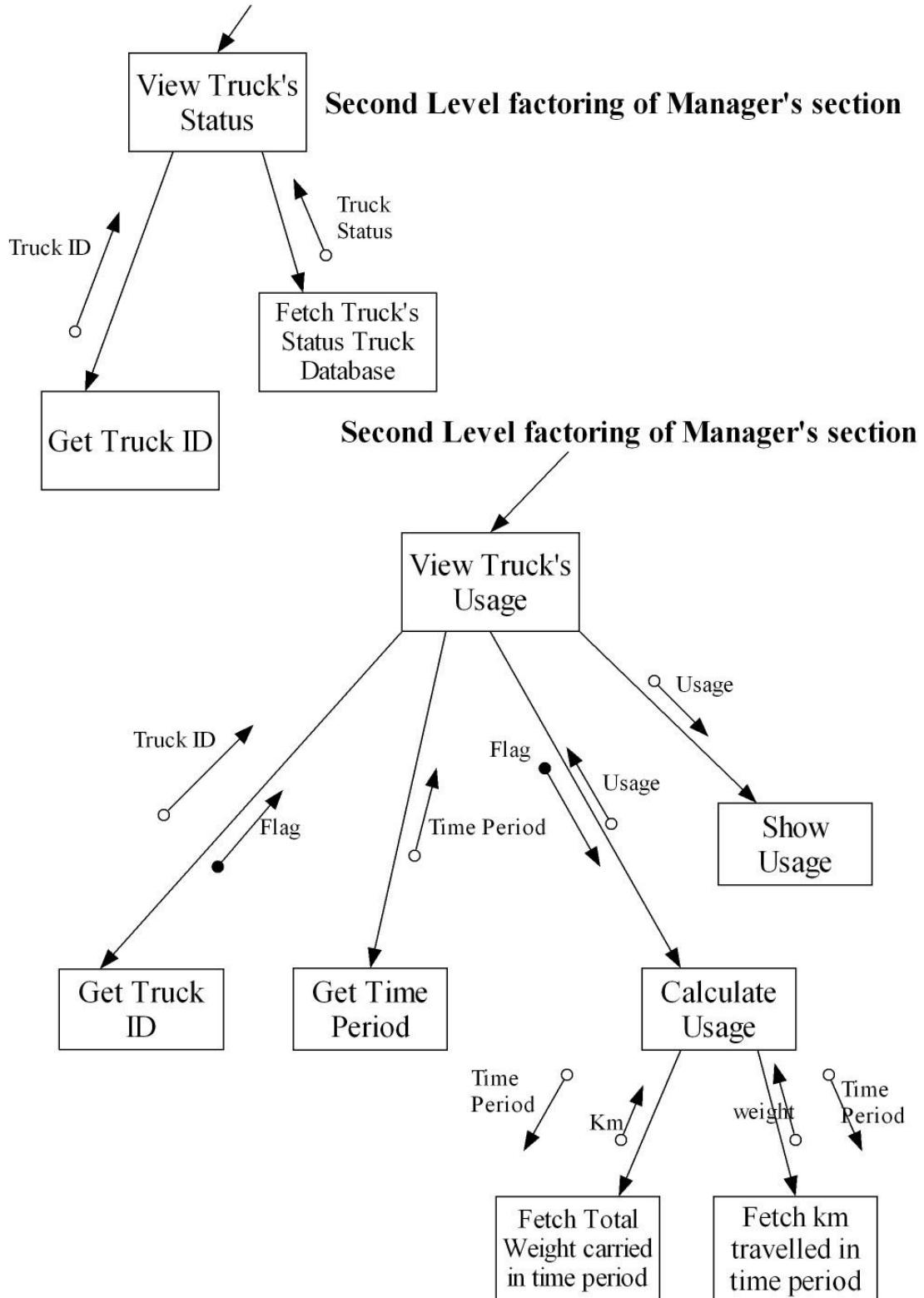
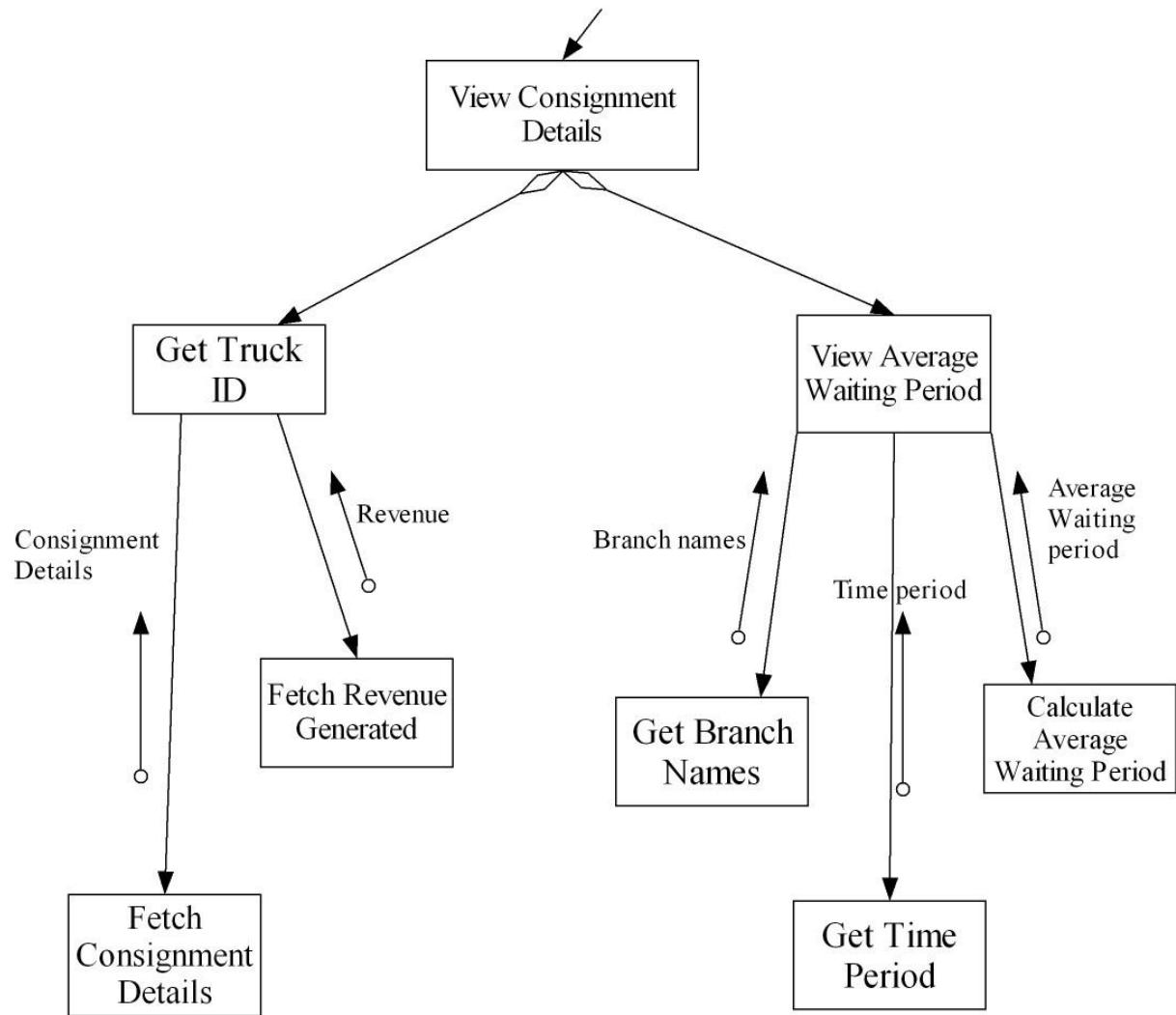


Fig.7

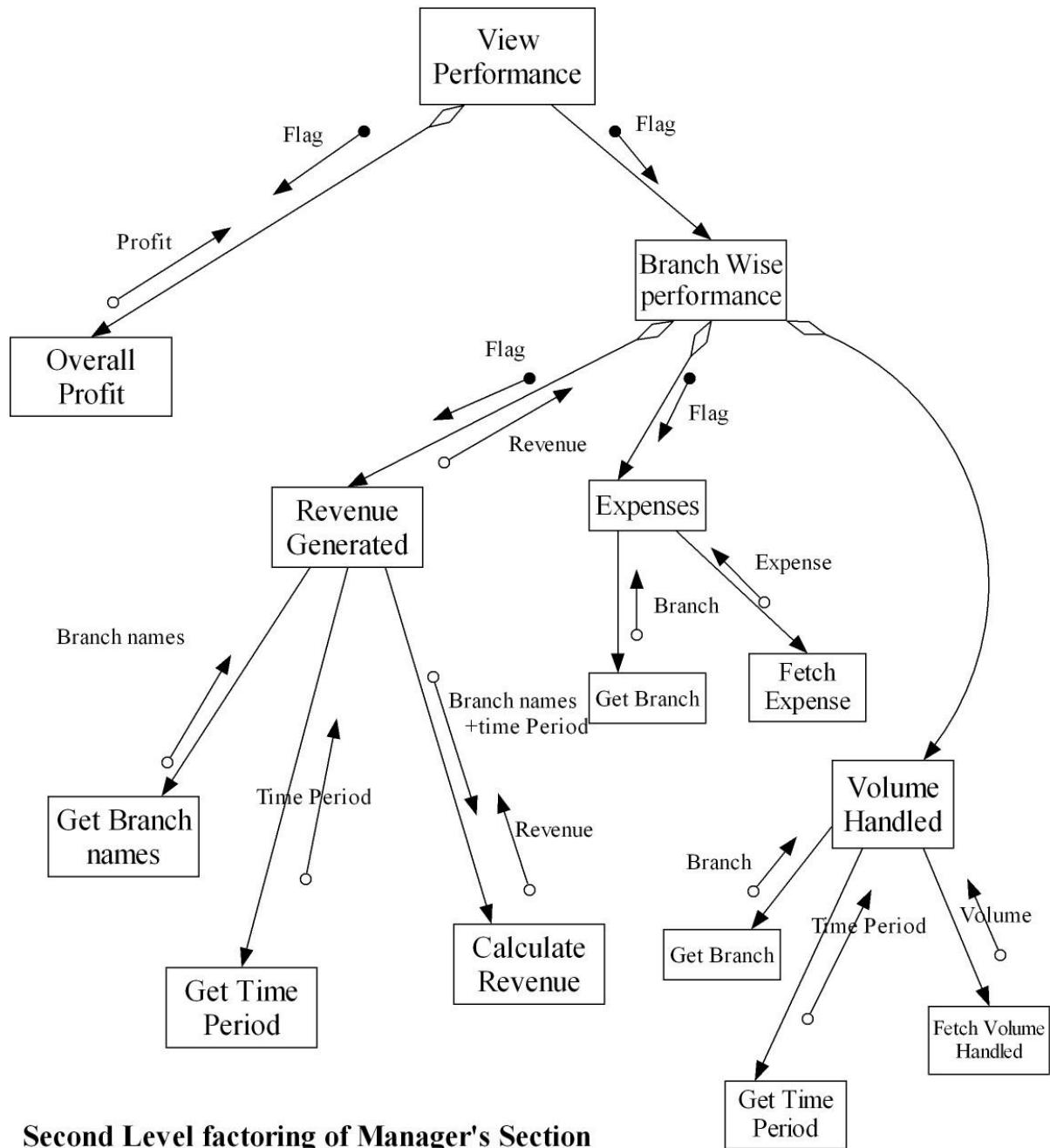
### 1.3 Third Level factoring of CCC software



### Second Level Factoring of Manager's Office Section

Fig.8

### Third Level factoring of CCC software



### Second Level factoring of Manager's Section

Fig.9

### **INTERFACES (Level 3 factoring):**

#### **Interfaces for VIEW TRUCK'S STATUS & VIEW TRUCK'S USAGE (Fig.7):**

1. Truck Status:  
String: Sender Branch Name, Receiver branch Name, Current Branch Name
2. Integer: Truck ID ,Km
3. Float: Weight
4. Time Period: String: Starting date, End Date.
5. Usage:
  - i. Float: Load Factor
  - ii. Integer: Km travelled

#### **Interfaces for VIEW CONSIGNMENT DETAILS (Fig.8):**

1. Consignment Details:
  - i. Same as Form Entries
  - ii. Float: Volume
  - iii. Integer: Revenue Generated
2. Integer: Revenue
3. Average Waiting Time Period: Integer: Days, Months
4. String: Branch Name 1, Branch Name 2.

#### **Interfaces for VIEW PERFORMANCES (Fig.9):**

1. Integer : Profit, Revenue, Expense
2. String: Branch, Branch names , (Branch name 1, Branch name 2)
3. Time Period-Integer-Day, Month, Year
4. Float: Volume

## 4.0 Assessment

---

1. Architecture Rationale is already given at the start.
2. As a matter of fact a good factorisation is one in which the higher functionalities only control the control flow and do not participate in low level functions. Our design satisfies these criteria because till first level only control is decided and actual work is done in level two and level three.
3. As one can see in the structured charts, no two modules are interconnected leading to low coupling.
4. Each function has different task and is mostly dependent on low level functions like editing, saving, searching. Hence each function has high cohesion value.
5. Many Data structures are reused in many functions.
6. We have decided to pass the whole data structure like consignment detail, receipt detail for transforming even single data field. This causes little higher coupling but better understanding.
7. Inclusion of https protocol increases security component of software.
8. Separate databases for different types of data have been used in the architecture.
9. New functionalities can be added later as much of the higher level components have only control factor (flag) which can be easily modified.
10. Thus our conclusion is that the design is understandable and with moderate coupling and high cohesion and with the capability to be modified later without much difficulty.

# Object Oriented Design

## Version 1.1

### March 2, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

## **Table of Contents**

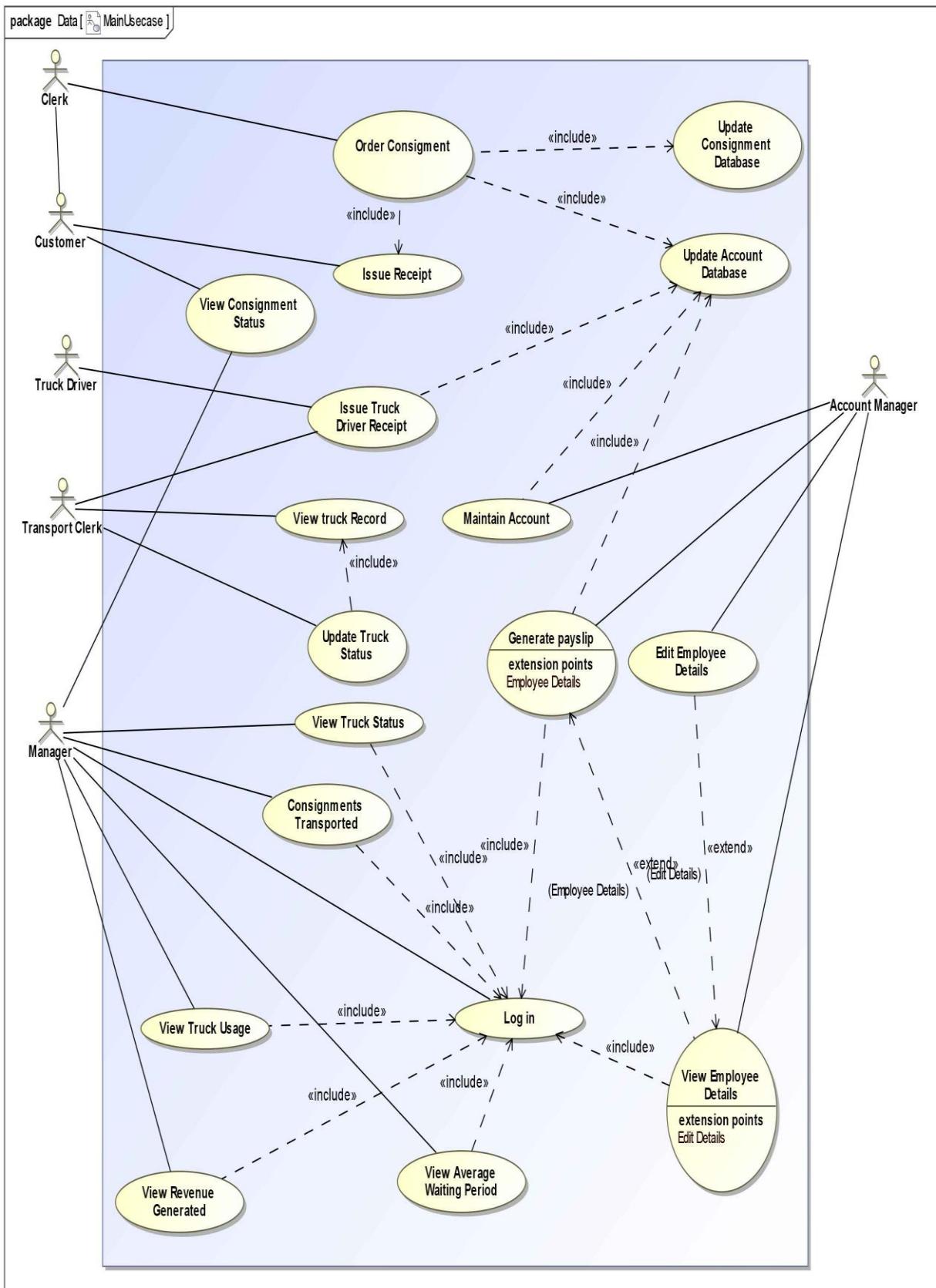
### **Table of Contents**

**1.0. Use Case Diagram**

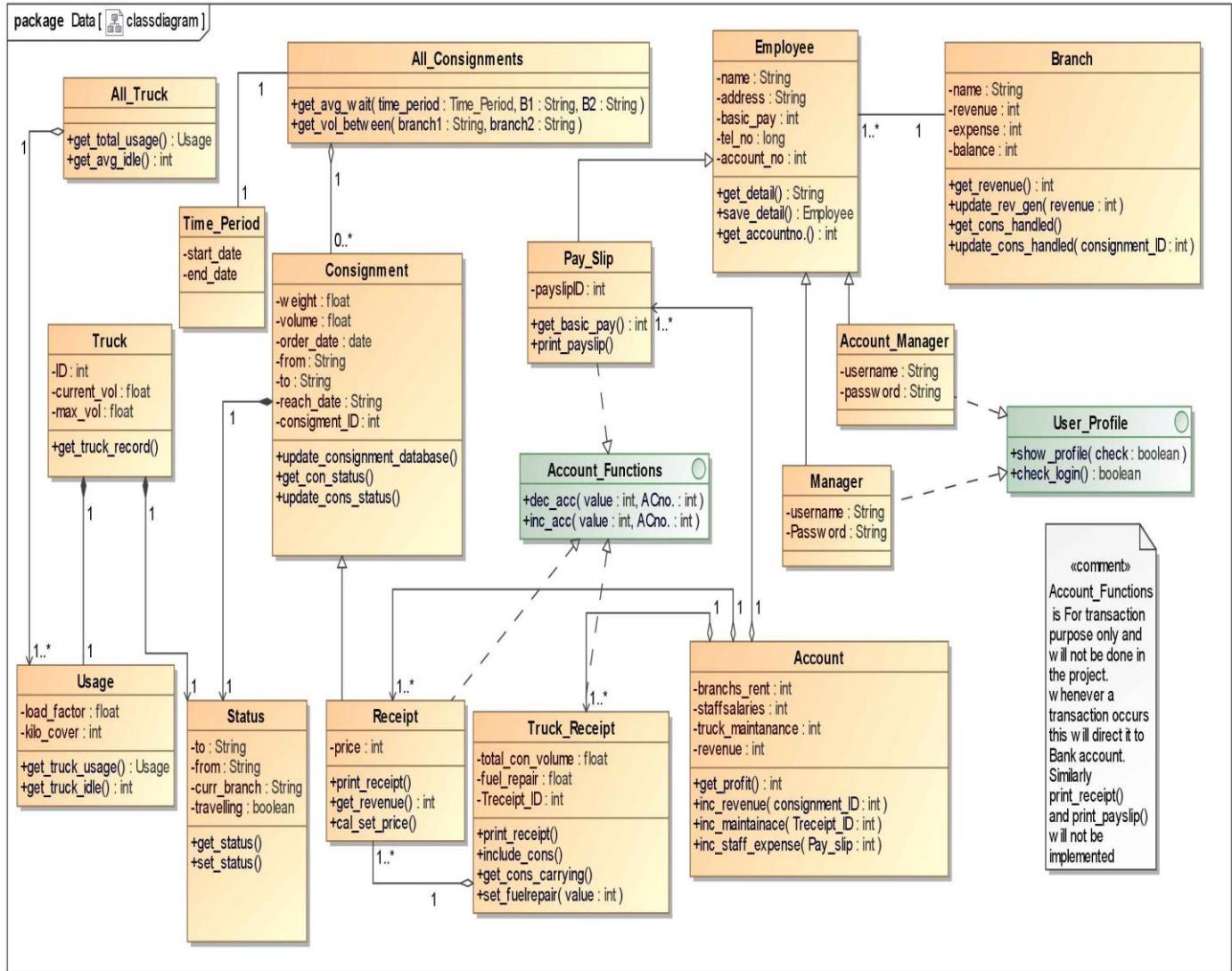
**2.0. Class Diagram**

**3.0. Sequence Diagram**

# 1.0. Use Case Diagram

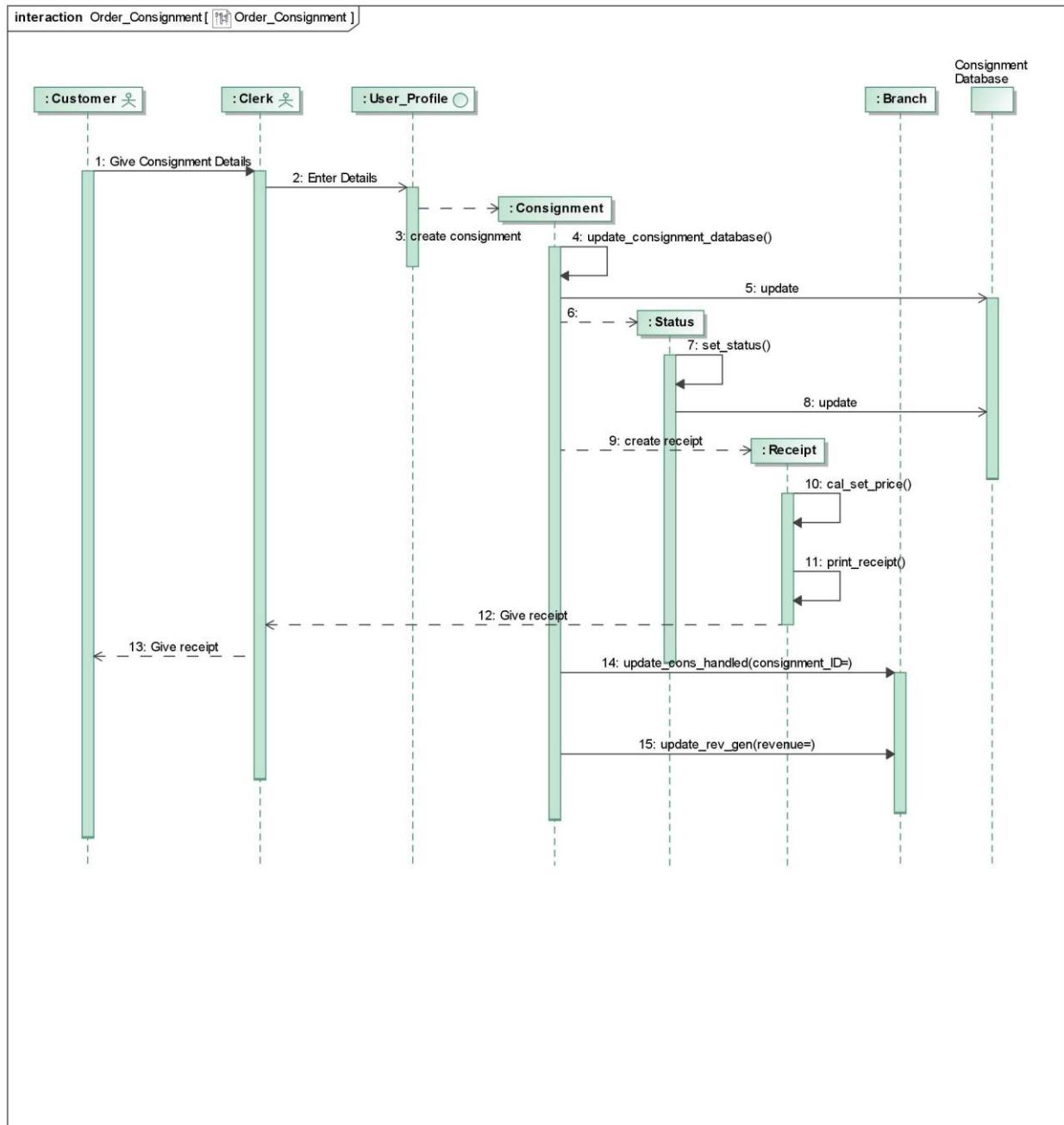


## 2.0. Class Diagram

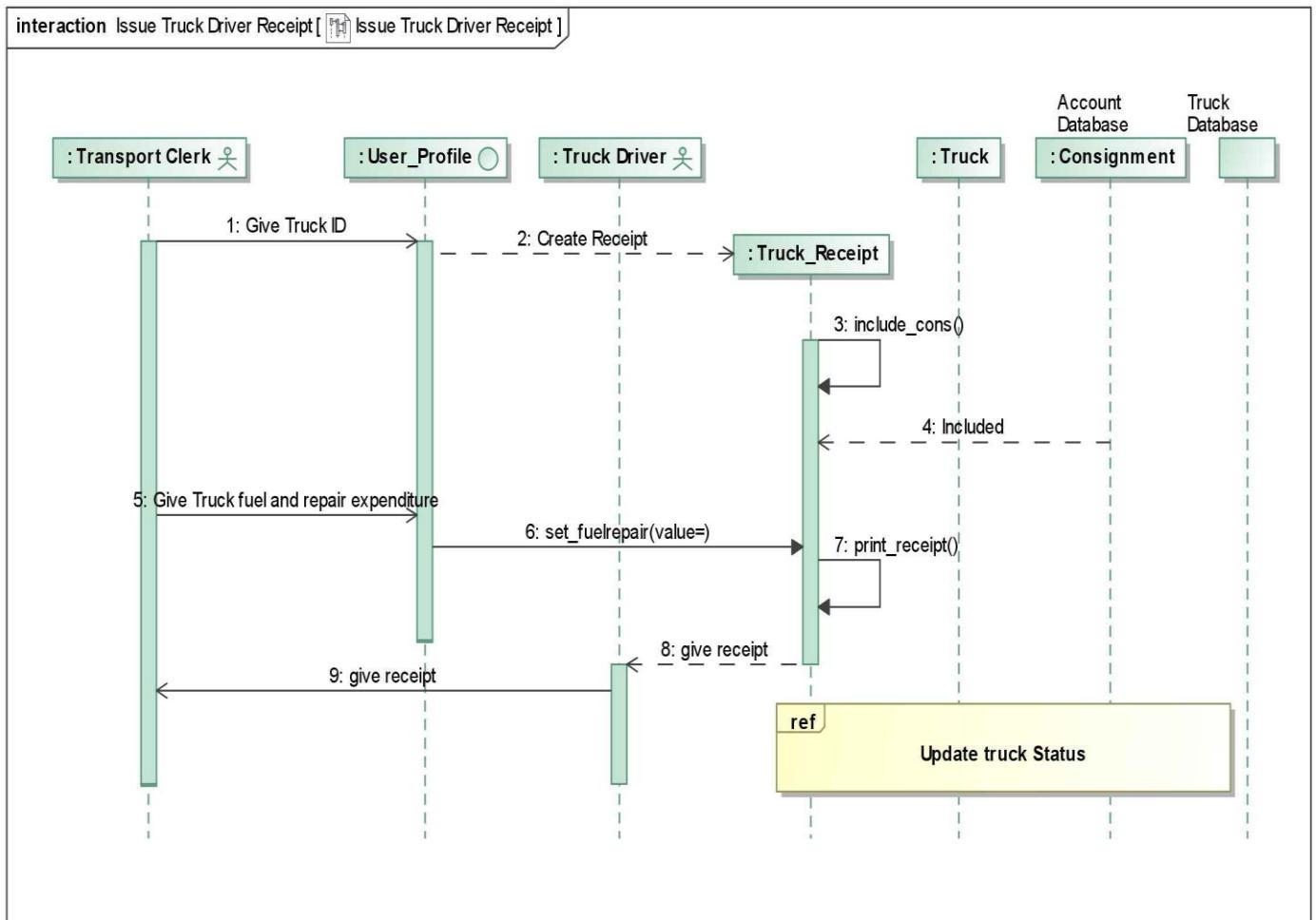


## 2.0.Sequence Diagram

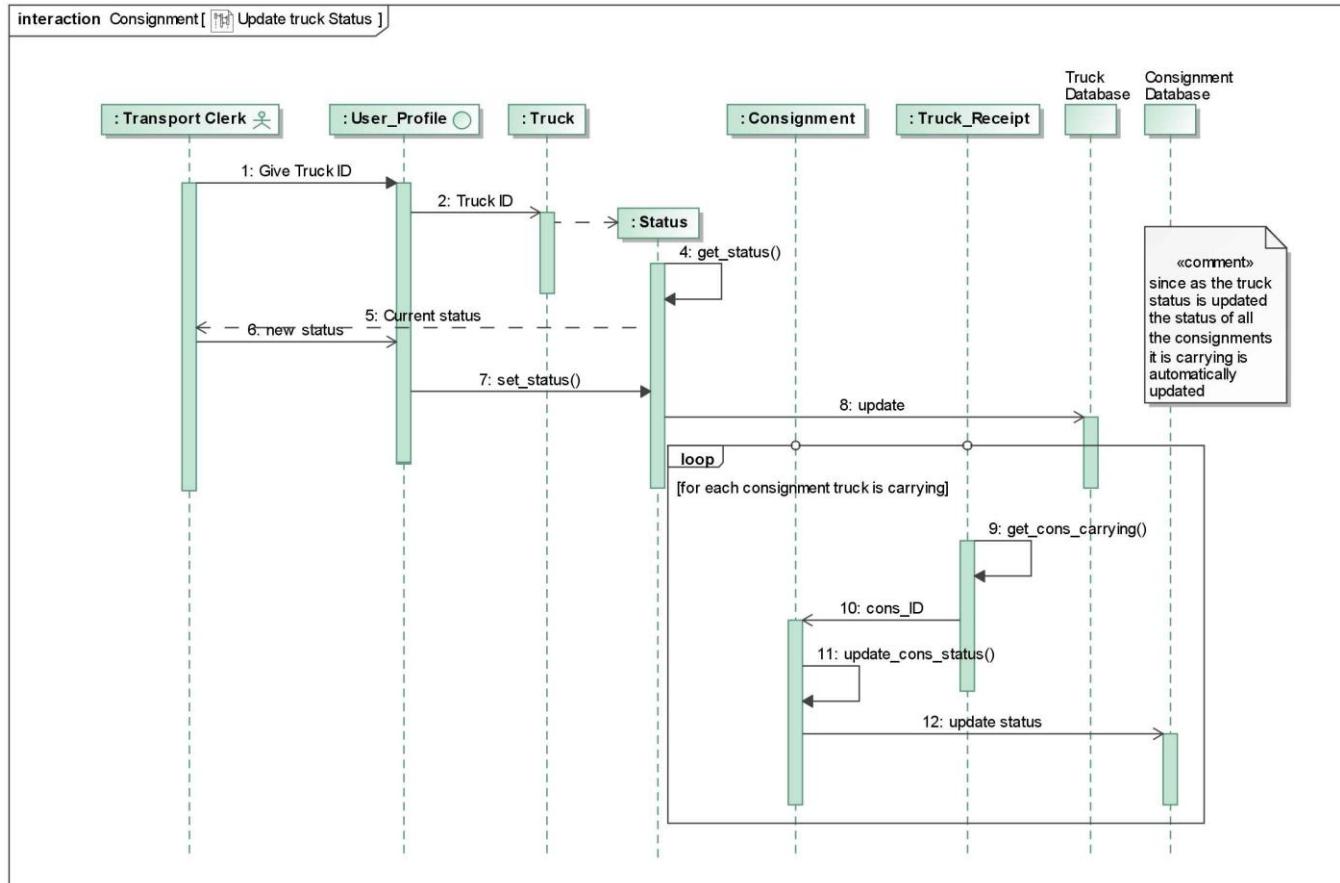
### 2.1.Order a consignment



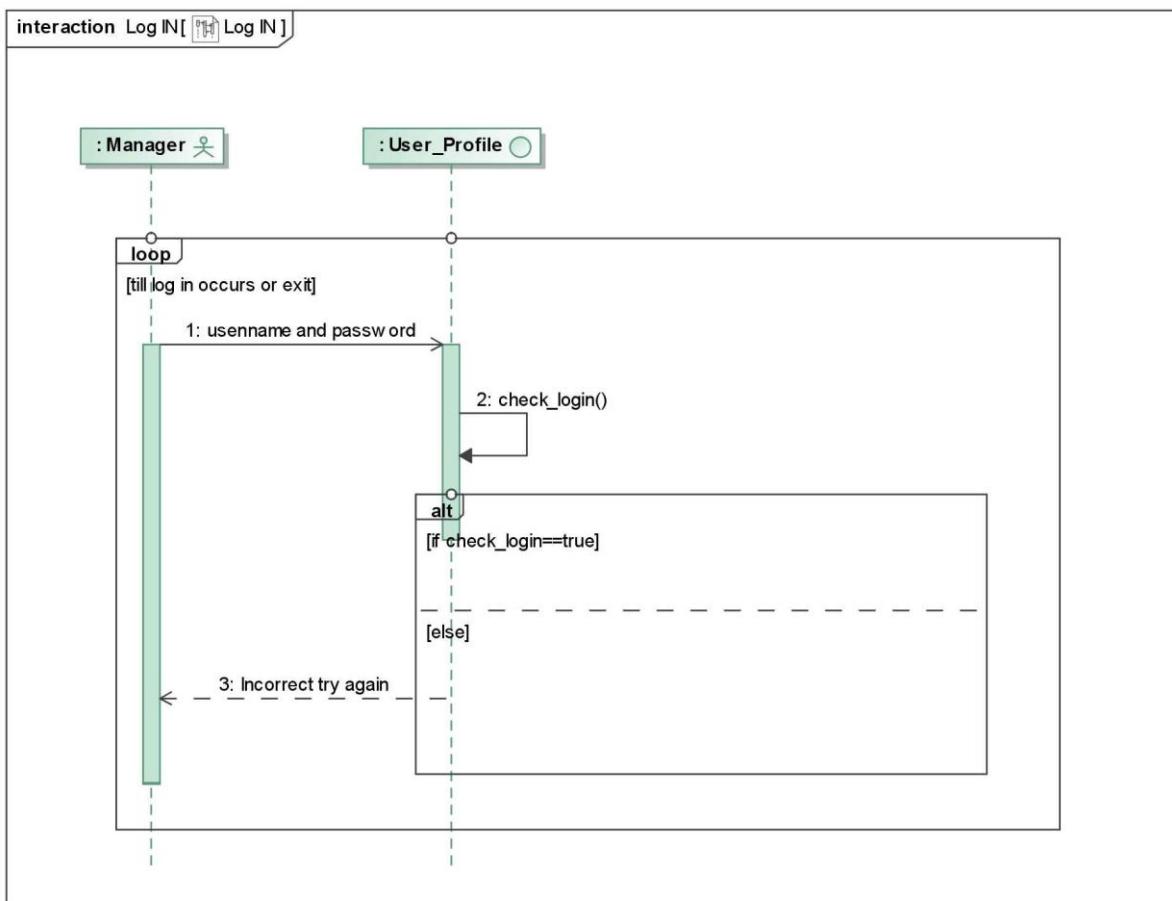
## 2.2.Issue Truck Driver Receipt



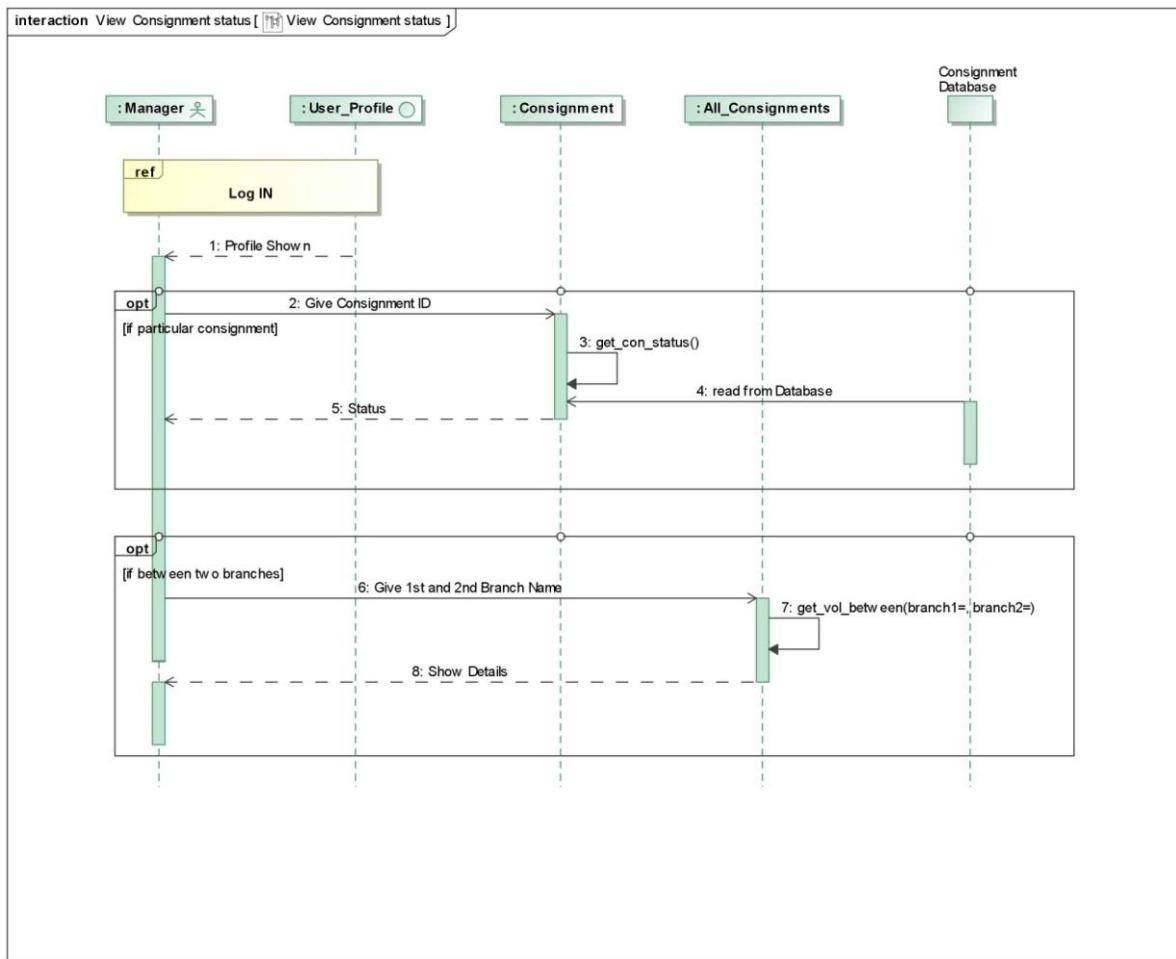
## 2.3. Update Truck Status



## 2.4. Log In

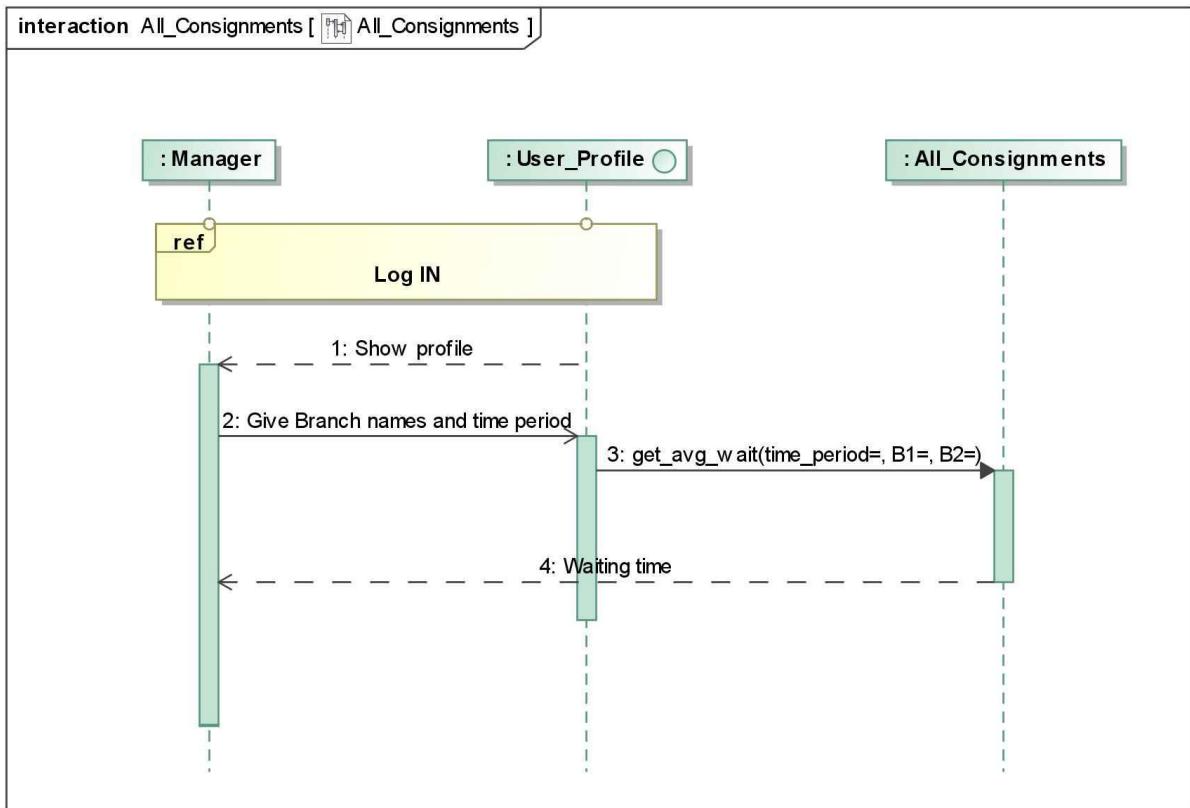


## 2.5. View Consignment Status

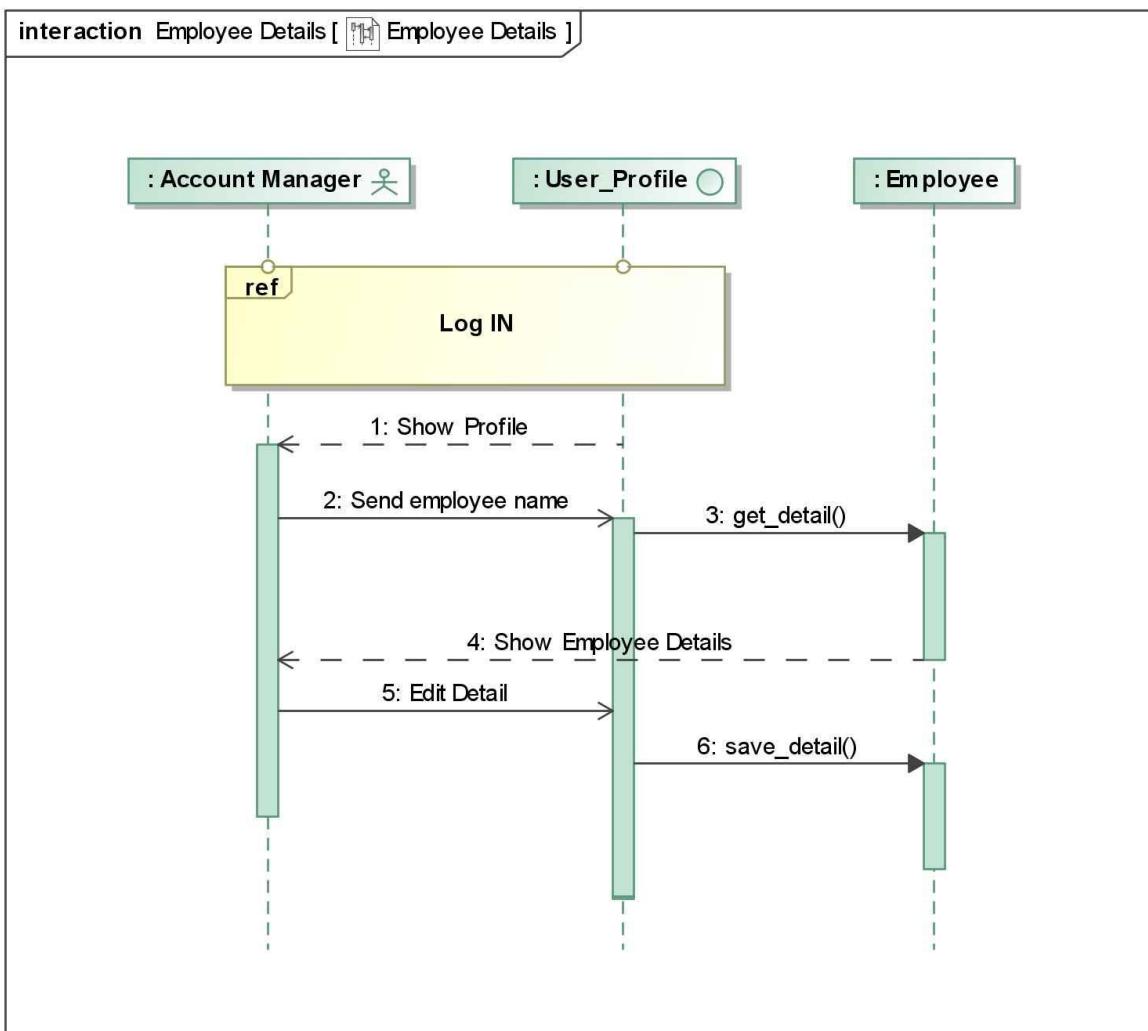


## 2.6.

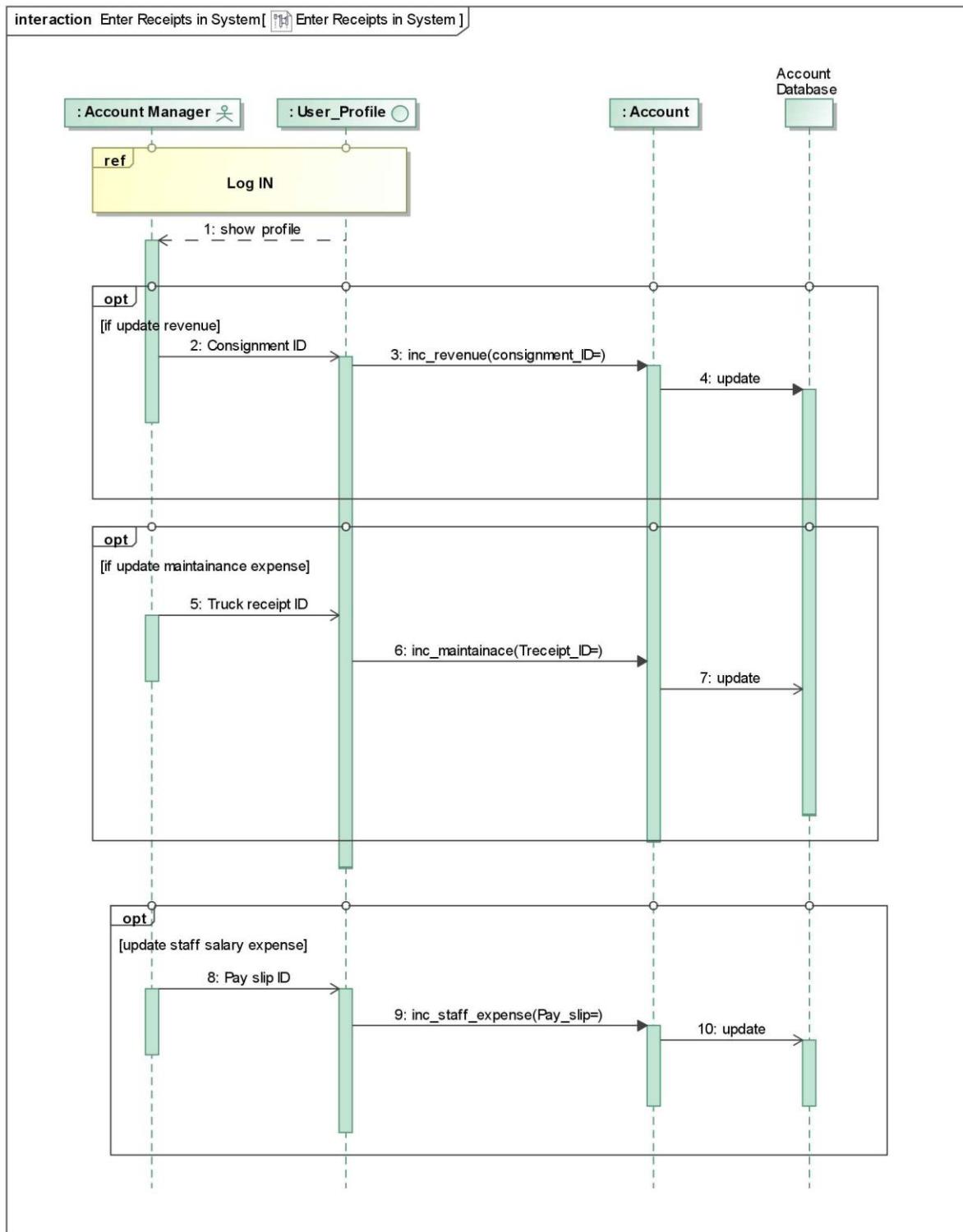
### View Average Waiting time



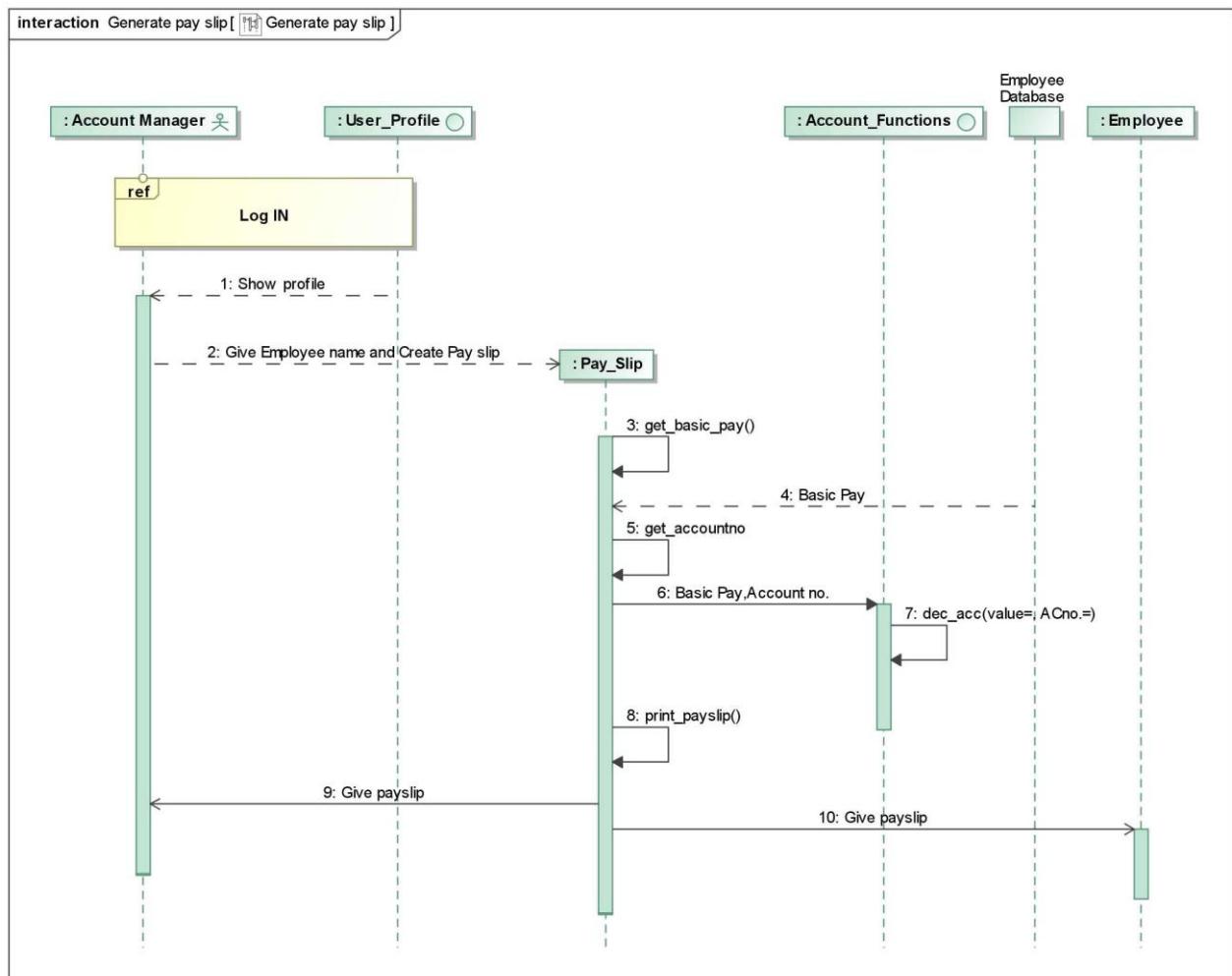
## 2.7. View Employee Details



## 2.8. Enter Receipts in System

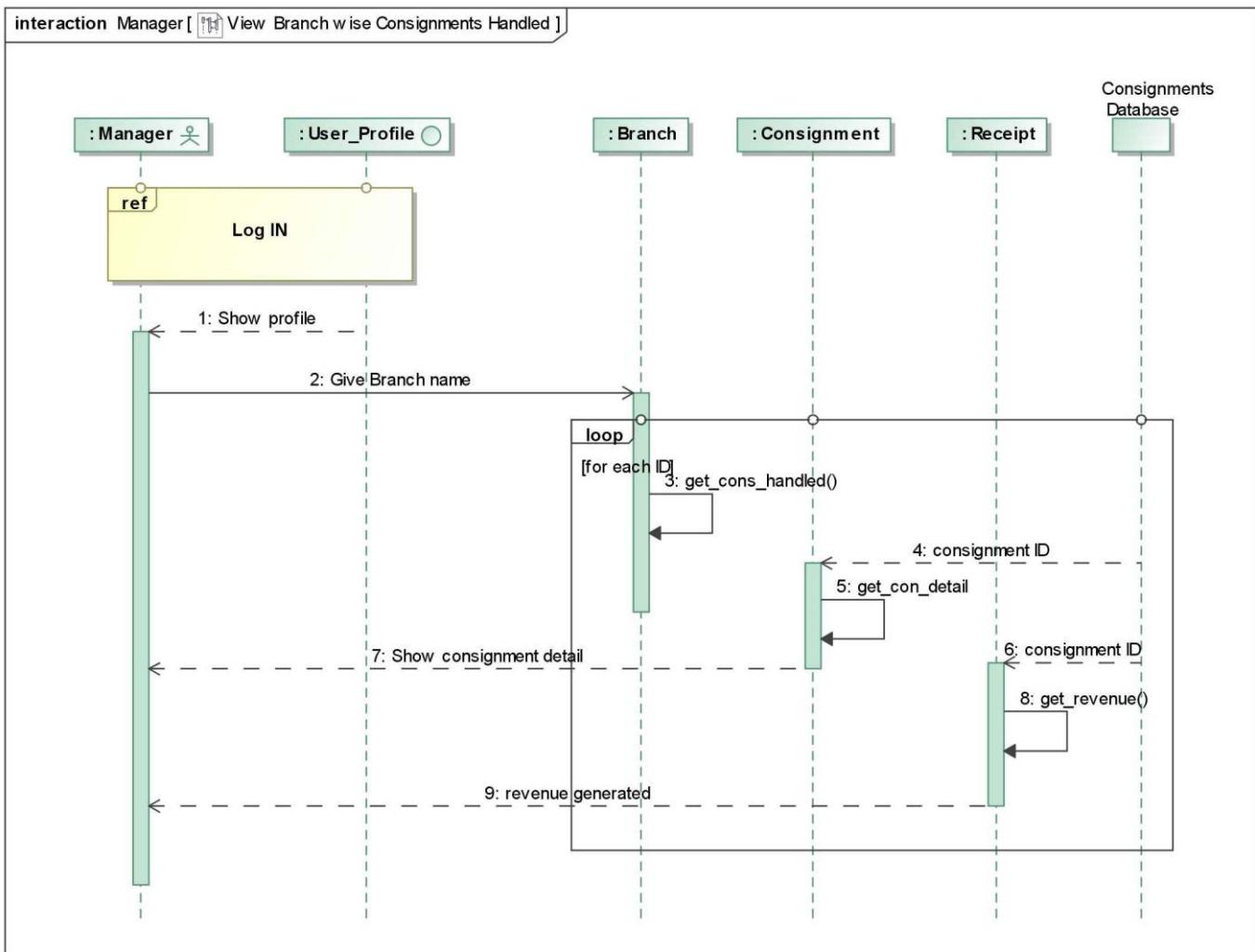


## 2.9. Generate pay Slip

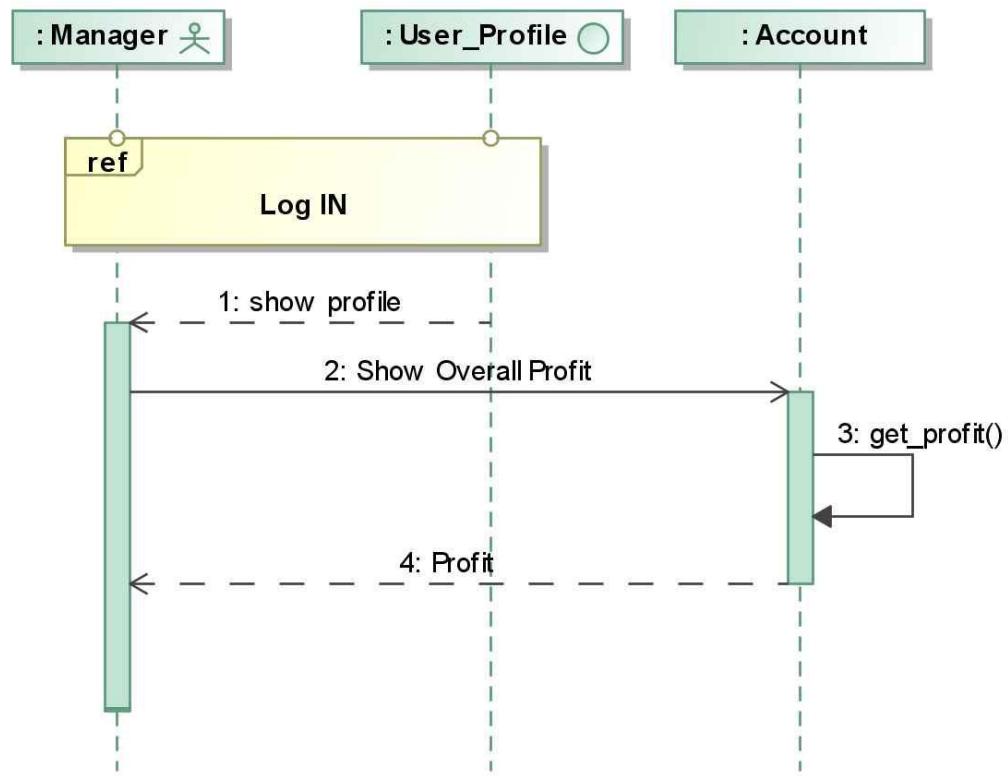


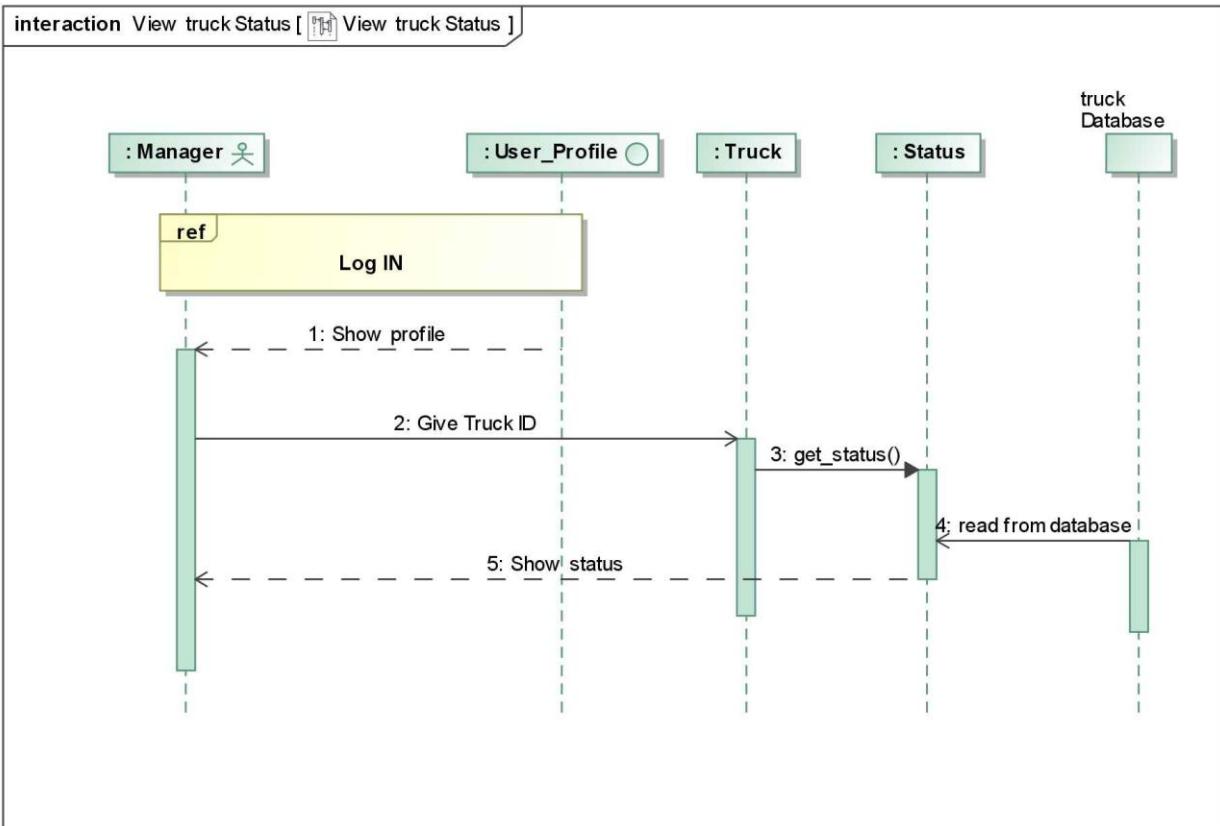
2.10.

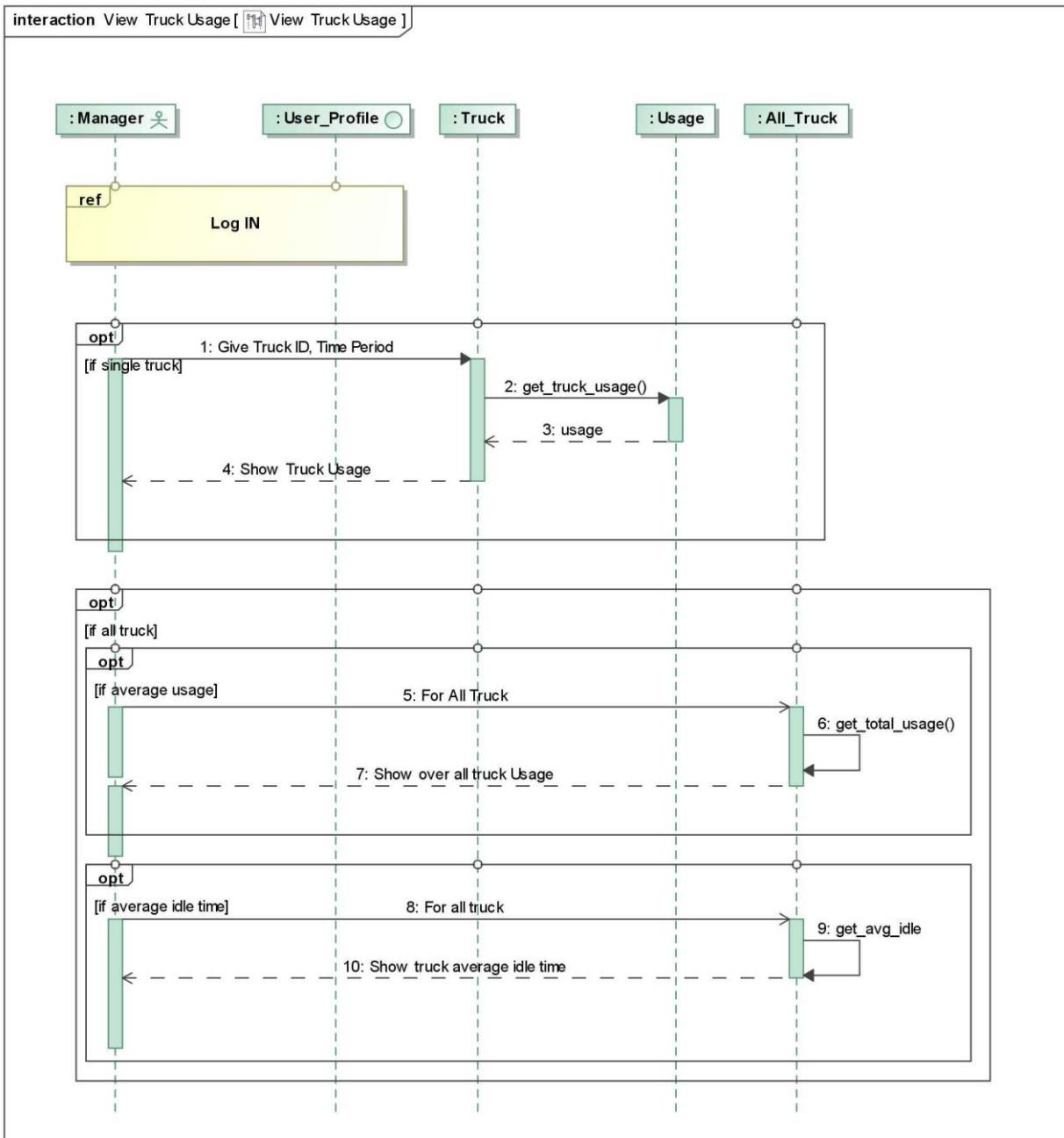
## View Branch Wise Consignment Handled

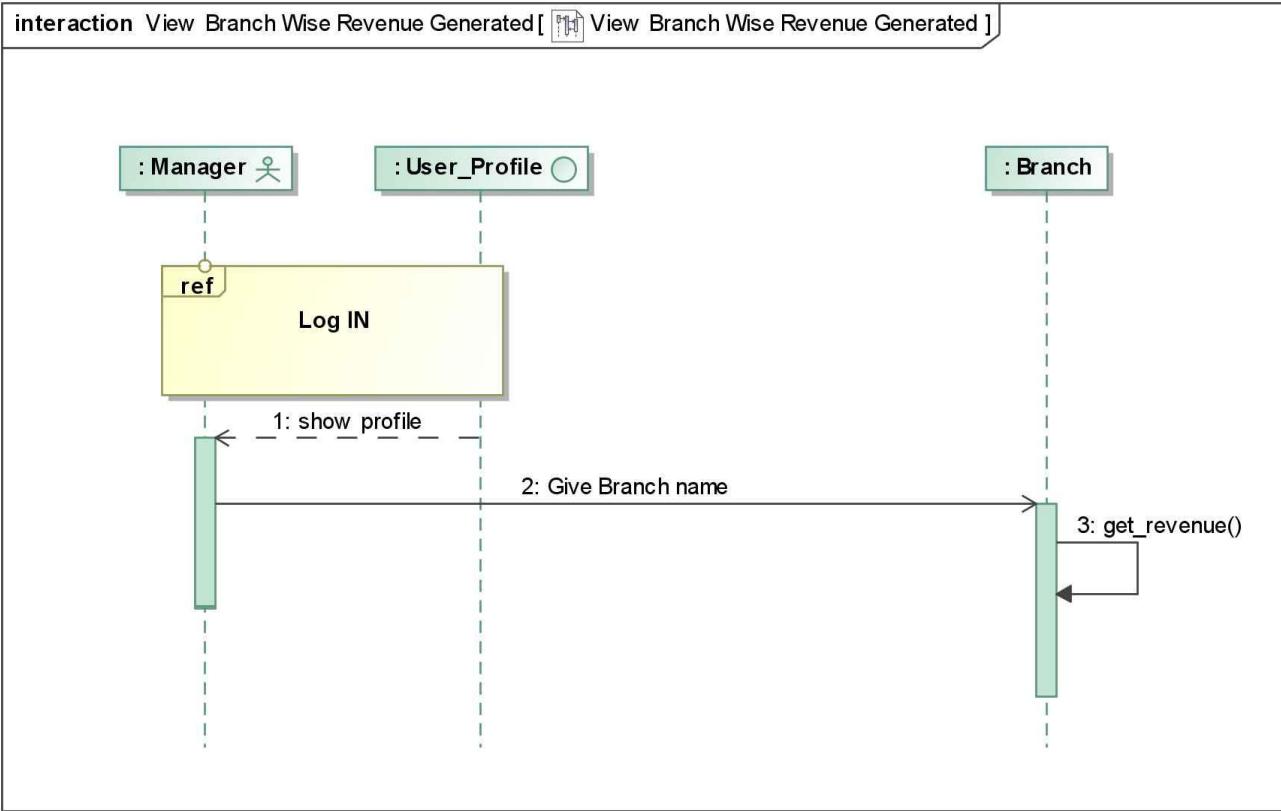


interaction View profit/Loss [  View profit/Loss ]

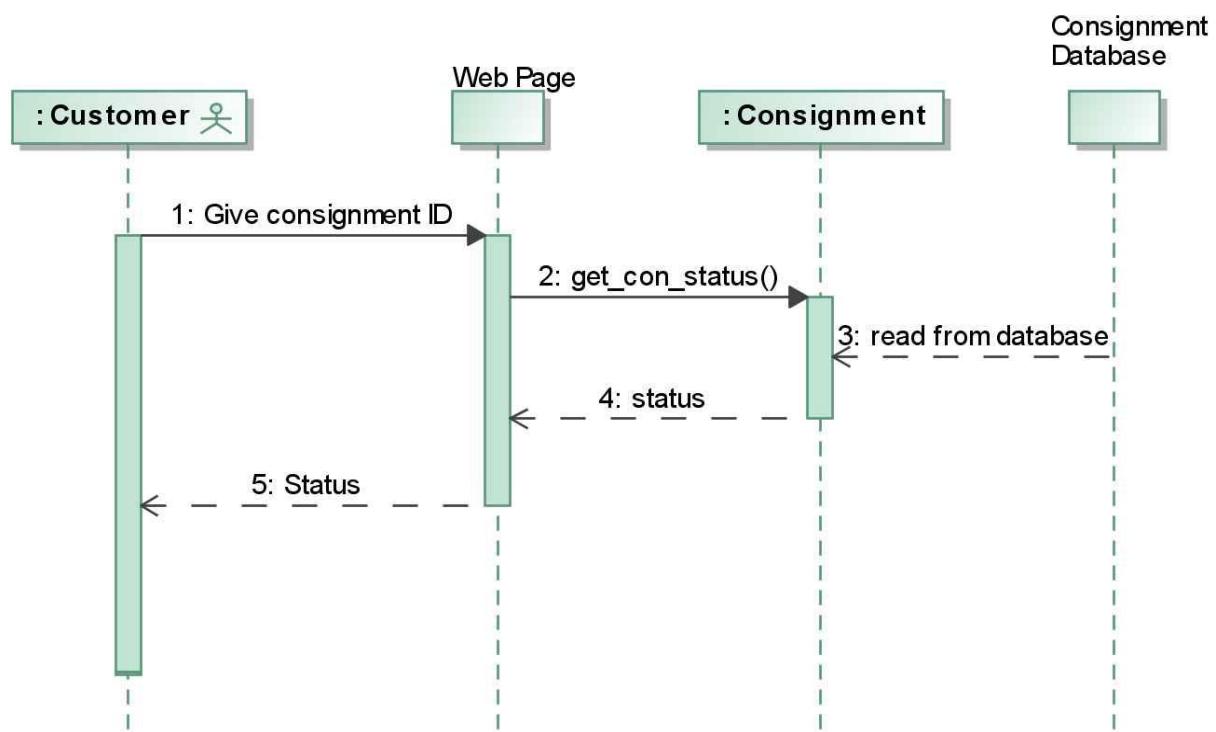








interaction View Status [  View Status ]

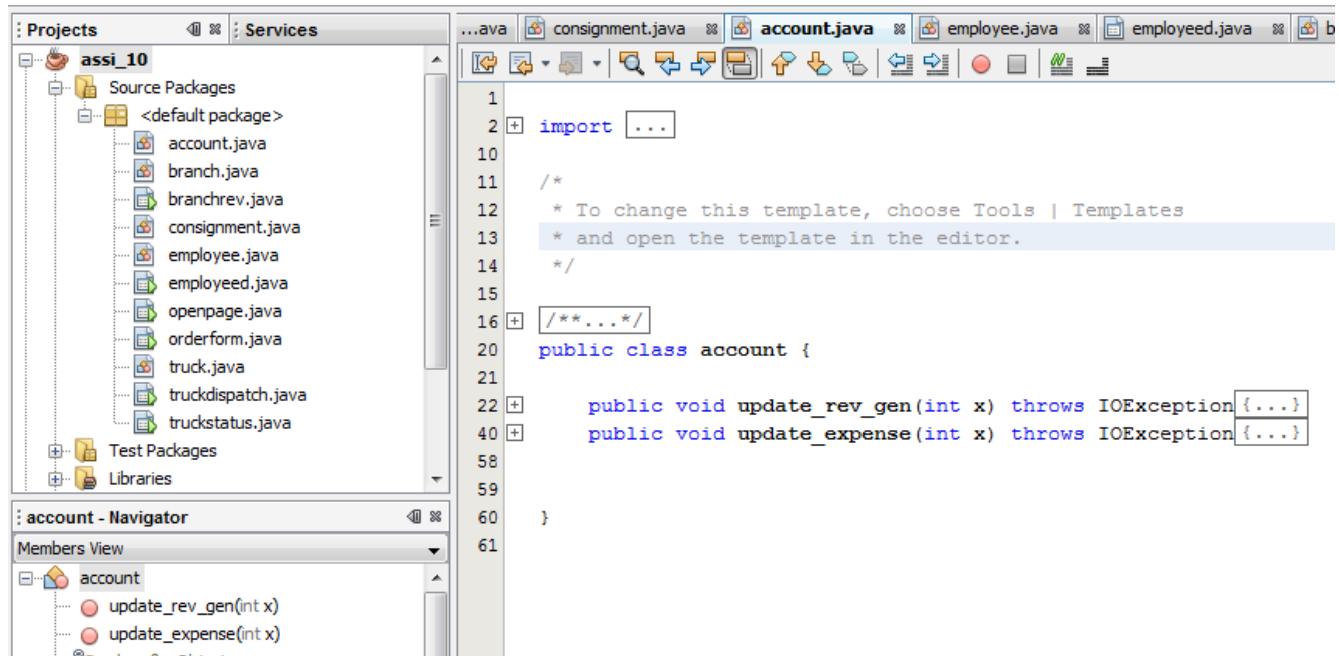


Object Oriented Design  
Java Code Snippets  
Version 1.1  
April 20, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

## Class Account code snippet

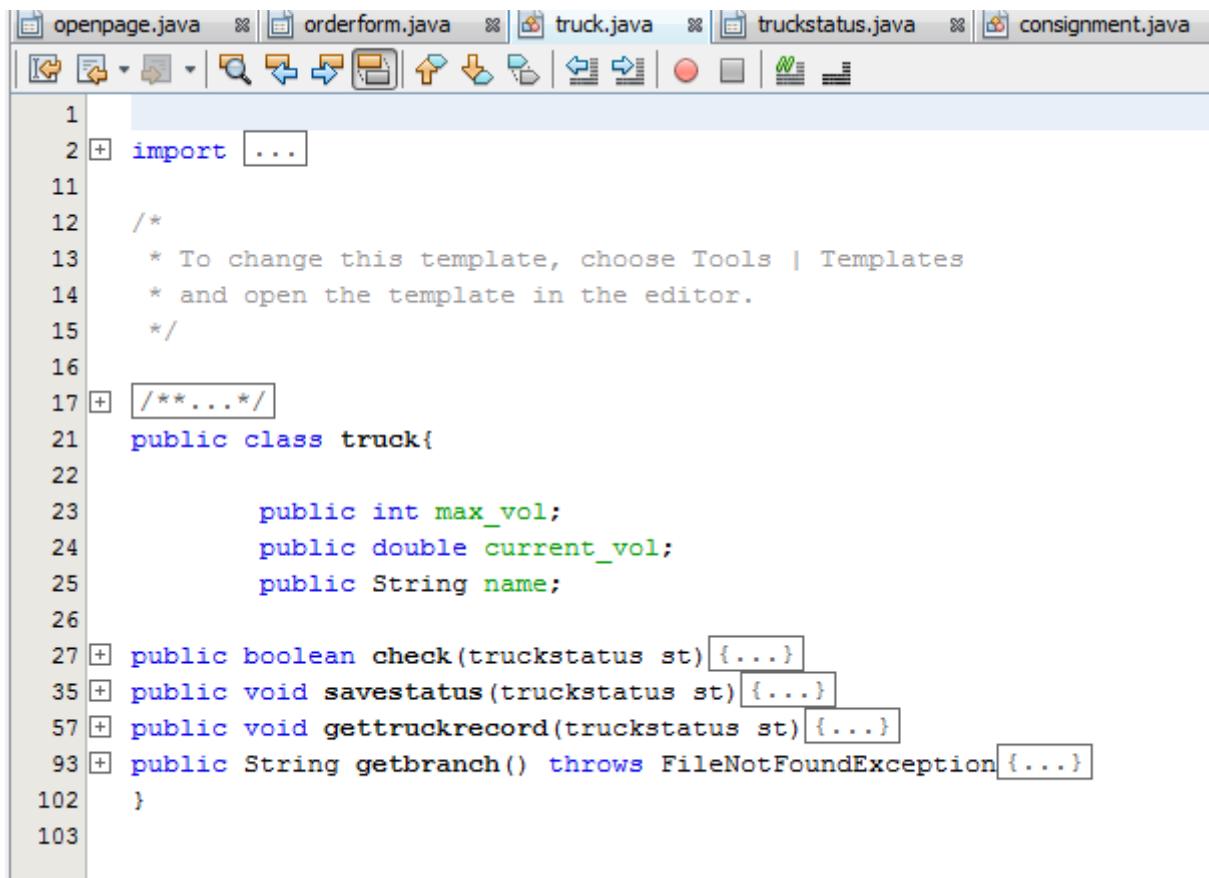


```
16 /**
17  */
18 public class account {
19
20     public void update_rev_gen(int x) throws IOException{...}
21     public void update_expense(int x) throws IOException{...}
22
23     BufferedWriter out = null;
24
25     int rev = 0;
26     Scanner in = null;
27     try {
28         in = new Scanner(new File("Account/expense.txt"));
29         } catch (FileNotFoundException ex) {
30             Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
31         }
32     rev=in.nextInt();
33     rev=rev+x;
34     out = new BufferedWriter(new FileWriter("Account/expense.txt"));
35     out.write(Integer.toString(rev));
36     out.close();
37     return;
38 }
39
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57 }
```

```
16  /**...*/
20  public class account {
21
22      public void update_rev_gen(int x) throws IOException{
23
24          BufferedWriter out = null;
25
26          int rev = 0;
27          Scanner in = null;
28          try {
29              in = new Scanner(new File("Account/revenue.txt"));
30          } catch (FileNotFoundException ex) {
31              Logger.getLogger(account.class.getName()).log(Level.SEVERE, null, ex);
32          }
33          rev=in.nextInt();
34          rev=rev+x;
35          out = new BufferedWriter(new FileWriter("Account/revenue.txt"));
36          out.write(Integer.toString(rev));
37          out.close();
38          return;
39      }

```

## Class Truck code snippet



```
1
2  import ...
11
12  /*
13  * To change this template, choose Tools | Templates
14  * and open the template in the editor.
15  */
16
17  /**...*/
21  public class truck{
22
23      public int max_vol;
24      public double current_vol;
25      public String name;
26
27  + public boolean check(truckstatus st){...}
35  + public void savestatus(truckstatus st){...}
57  + public void gettruckrecord(truckstatus st){...}
93  + public String getbranch() throws FileNotFoundException{...}
102 }
103
```

```
26
27 [+ public boolean check(truckstatus st){...}
35 [- public void savestatus(truckstatus st){
36     String m2,m1;
37     m1=st.trucknum.getSelectedItem().toString();
38     m2=st.combobranch2.getSelectedItem().toString();
39
40
41     BufferedWriter out = null;
42     try {
43         //out = new BufferedWriter(new FileWriter("studentfiles/"+fname.getText()+sna
44         out = new BufferedWriter(new FileWriter("truck/"+m1+".txt"));
45
46     } catch (IOException ex) {
47         Logger.getLogger(truck.class.getName()).log(Level.SEVERE, null, ex);
48     }
49     try {
50         out.write("0 "+m2);
51         out.close();
52     } catch (IOException ex) {
53         Logger.getLogger(truckstatus.class.getName()).log(Level.SEVERE, null, ex);
54     }
55
56
57 [+ /**...*/
58
59 public class truck{
60
61
62     public int max_vol;
63     public double current_vol;
64     public String name;
65
66
67 [+ public boolean check(truckstatus st){...}
68 [+ public void savestatus(truckstatus st){...}
69 [+ public void gettruckrecord(truckstatus st){...}
70 [- public String getbranch() throws FileNotFoundException{
71     String m = null;
72     int s;
73     Scanner in;
74     in = new Scanner(new File("truck/"+name+".txt"));
75     s=in.nextInt();
76     m=in.next();
77     return m;
78 }
79 }
```

```
27 [+ public boolean check(truckstatus st){...}
35 [+ public void savestatus(truckstatus st){...}
57 [- public void gettruckrecord(truckstatus st){
58     st.nopanel.setVisible(false);
59     st.yespanel.setVisible(false);
60     int status=0;
61     String m=null;
62     Scanner in = null;
63     if(check(st)==false){
64         return;
65     }
66     m=st.trucknum.getSelectedItem().toString();
67     try {
68         in = new Scanner(new File("truck/"+m+".txt"));
69     } catch (FileNotFoundException ex) {
70         Logger.getLogger(truckstatus.class.getName()).log(Level.SEVERE, null, ex);
71     }
72     status=in.nextInt();
73     if(status==1){
74         st.yespanel.setVisible(true);
75         st.currentlyyes.setText("Currently Transporting YES");
76         st.fromtext.setText("From "+in.next());
77         st.totext.setText("To "+in.next());
78     }
79     else{
80         st.nopanel.setVisible(true);
81         st.currentlyno.setText("Currently Transporting NO");
82         st.attext.setText("Currently at "+in.next());
83     }
84     return;
85 }
```

## Class Consignment code snippet

```
  public void update_consignment_database() throws IOException{
    File myfile = null;
    BufferedWriter out = null;

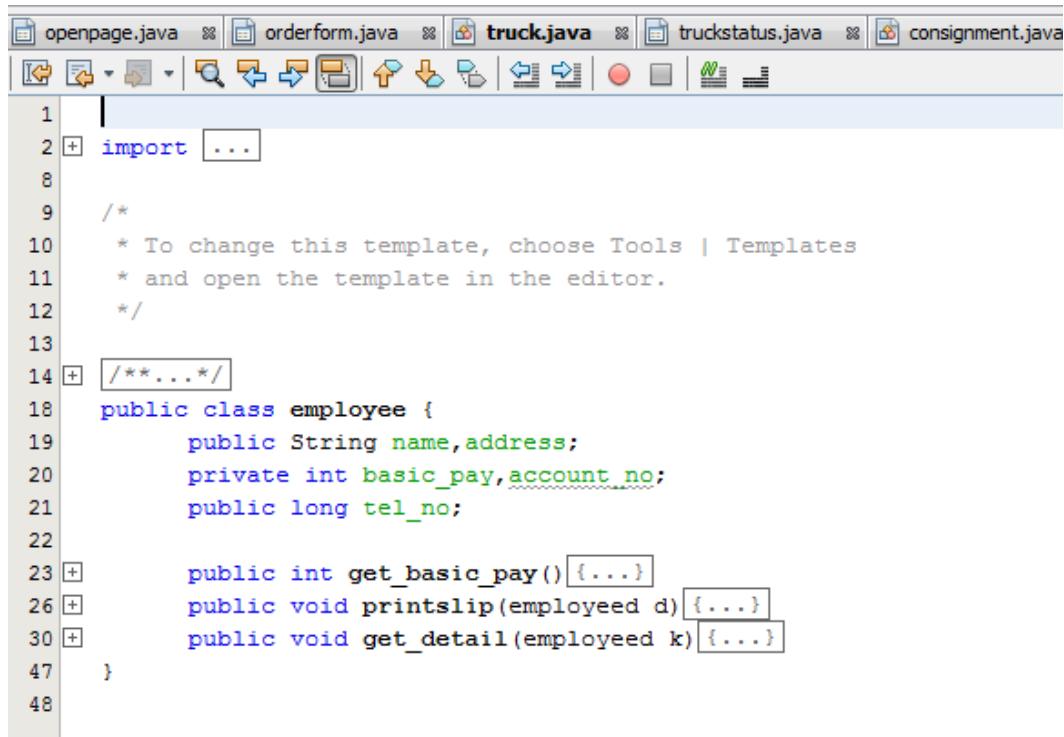
    if(type==1){
        myfile=new File("orders/sorders/"+Integer.toString(ID)+".txt");
        myfile.createNewFile();

        out = new BufferedWriter(new FileWriter("orders/sorders/"+Integer.toString(ID)+".txt"));
        out.write(Integer.toString(ID));out.newLine();
        out.write(Integer.toString(price));out.newLine();
        out.write(Double.toString(weight));out.newLine();
        out.write(order_date);out.newLine();
        out.write(to);out.newLine();
        out.write(from);out.newLine();
        out.write(senderb);out.newLine();
        out.write(recb);out.newLine();
        out.close();
    }
    else{
        myfile=new File("orders/lorders/"+Integer.toString(ID)+".txt");
        myfile.createNewFile();
        out = new BufferedWriter(new FileWriter("orders/lorders/"+Integer.toString(ID)+".txt"));
        out.write(Integer.toString(ID));out.newLine();
        out.write(Integer.toString(price));out.newLine();
        out.write(Double.toString(weight));out.newLine();
        out.write(Double.toString(volume));out.newLine();
        out.write(Integer.toString(distance));out.newLine();
        out.write(order_date);out.newLine();
    }
    else{
        myfile=new File("orders/lorders/"+Integer.toString(ID)+".txt");
        myfile.createNewFile();
        out = new BufferedWriter(new FileWriter("orders/lorders/"+Integer.toString(ID)+".txt"));
        out.write(Integer.toString(ID));out.newLine();
        out.write(Integer.toString(price));out.newLine();
        out.write(Double.toString(weight));out.newLine();
        out.write(Double.toString(volume));out.newLine();
        out.write(Integer.toString(distance));out.newLine();
        out.write(order_date);out.newLine();
        out.write(to);out.newLine();
        out.write(from);out.newLine();
        out.write(senderb);out.newLine();
        out.write(recb);out.newLine();
        out.close();
    }
    //ID price weight volume distance order_date to from senderb recb reach_date
    //ID price weight order_date to from senderb recb reach_date
}
public int price(){...}

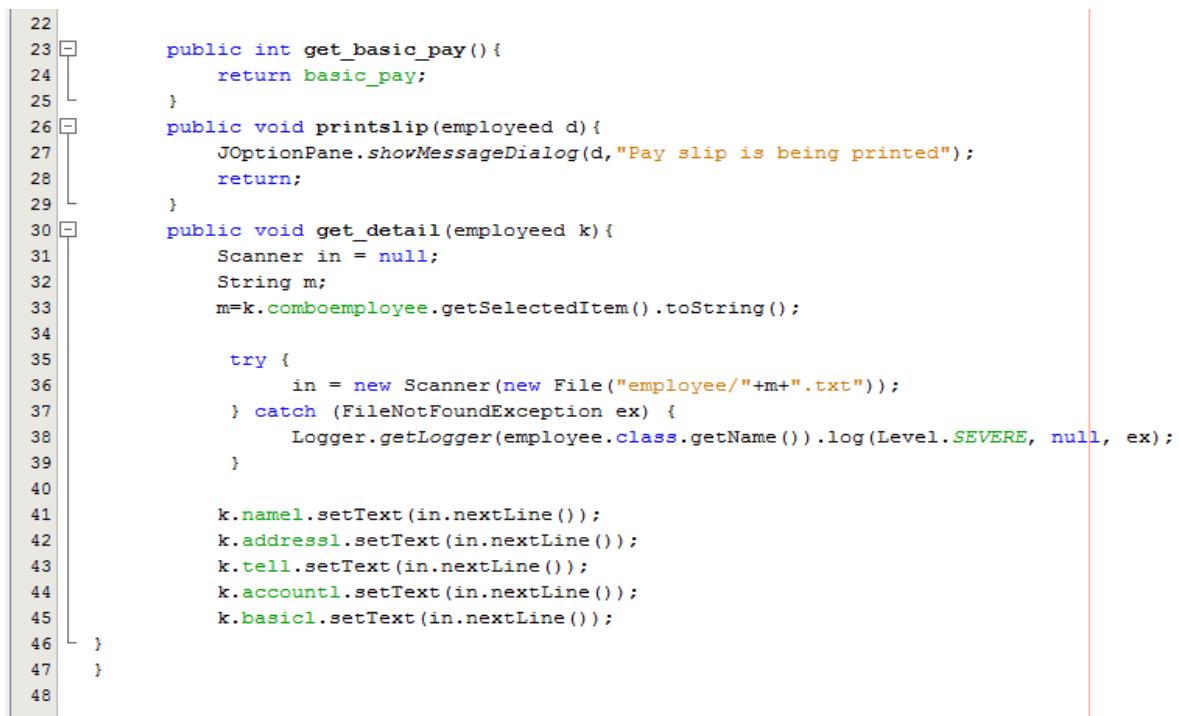
```

```
121     public int price() {
122         int price=0,y,a,b = 0;
123         double density = 0;
124         double x;
125
126         if(type==1){
127             price=(int) (Math.ceil(weight / 100) * 50);
128             return price;
129         }
130         else if(type==0){
131             price=(int) (5000 * volume);
132             density=weight/volume;
133
134             if(density>100){
135                 x=density-100;
136                 b=(int) (Math.ceil((x / 20.0))*price*( 1/10.0));
137             }
138             if(distance<=500){
139                 price=price+b;
140             }
141             else {
142                 y=distance-500;
143                 a=(int) Math.ceil(y/100.0);
144
145                 price=price+a+b;
146             }
147             return price;
148         }
149         return price;
150     }
```

## Class Employee code snippet

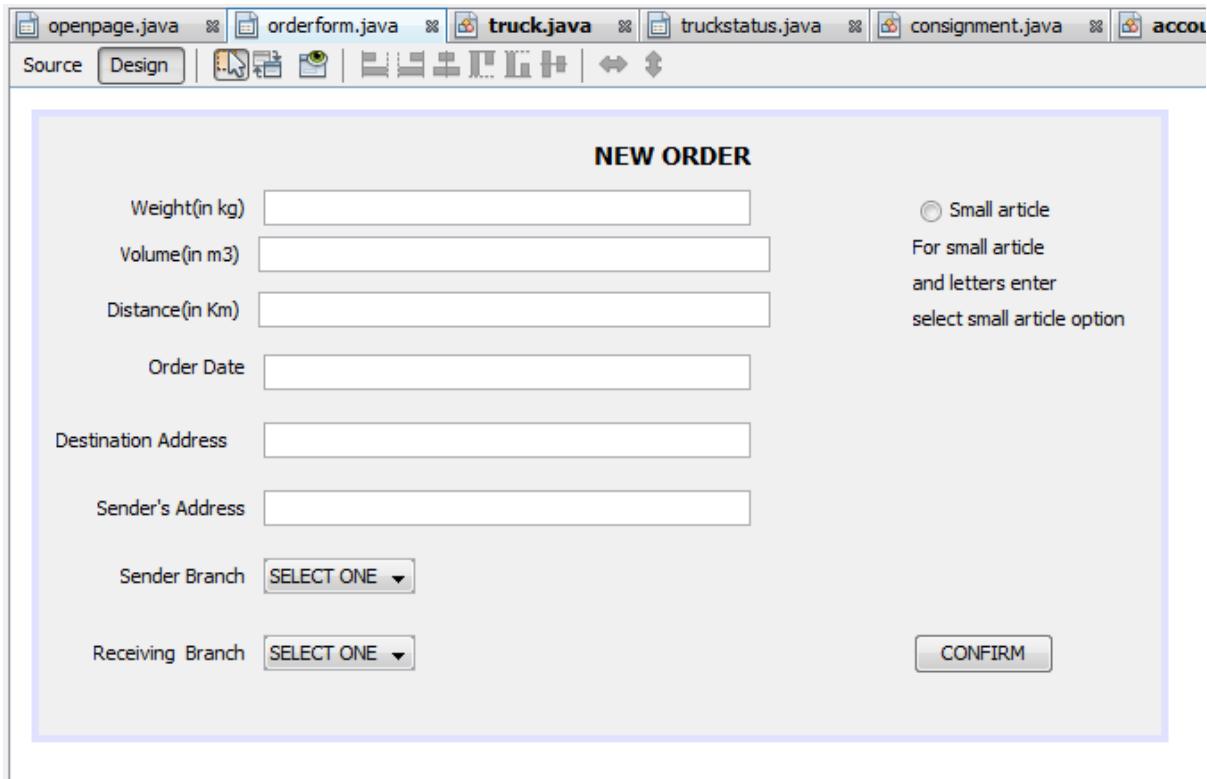


```
1
2 import ...
8
9 /*
10  * To change this template, choose Tools | Templates
11  * and open the template in the editor.
12  */
13
14 /**
15  * ...
16  */
17 public class employee {
18     public String name,address;
19     private int basic_pay,account_no;
20     public long tel_no;
21
22     public int get_basic_pay(){...}
23     public void printslip(employee d){...}
24     public void get_detail(employee k){...}
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```



```
22
23     public int get_basic_pay(){
24         return basic_pay;
25     }
26     public void printslip(employee d){
27         JOptionPane.showMessageDialog(d,"Pay slip is being printed");
28         return;
29     }
30     public void get_detail(employee k){
31         Scanner in = null;
32         String m;
33         m=k.comboemployee.getSelectedItem().toString();
34
35         try {
36             in = new Scanner(new File("employee/"+m+".txt"));
37         } catch (FileNotFoundException ex) {
38             Logger.getLogger(employee.class.getName()).log(Level.SEVERE, null, ex);
39         }
40
41         k.name1.setText(in.nextLine());
42         k.address1.setText(in.nextLine());
43         k.tell.setText(in.nextLine());
44         k.account1.setText(in.nextLine());
45         k.basic1.setText(in.nextLine());
46     }
47 }
48
```

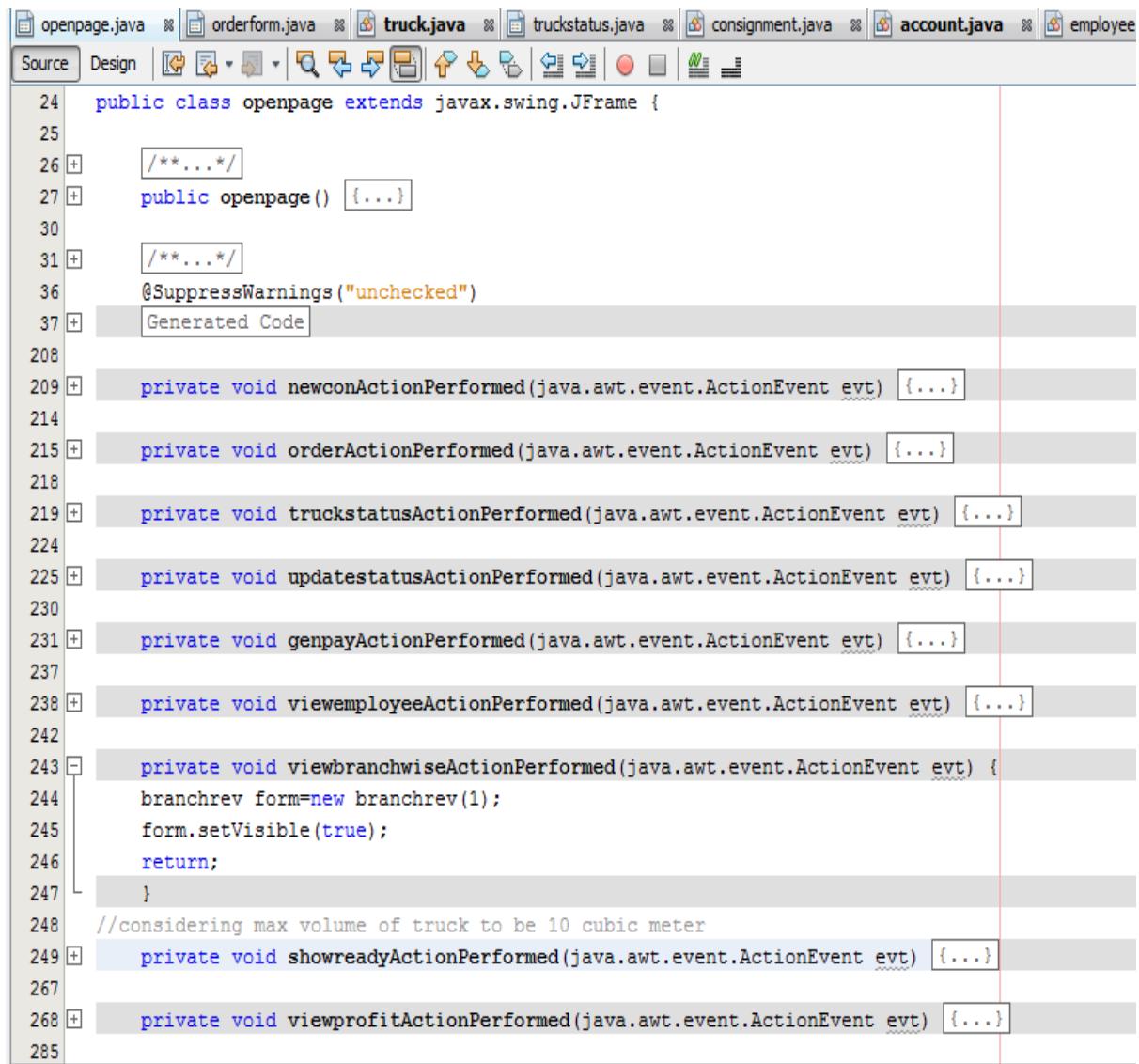
## Order form code snippet



```
27  public class orderform extends javax.swing.JFrame {
28
29  +     /**
30  -     public orderform() {
31  -         initComponents();
32  /**
33  -         Scanner in = null;
34  -         String m;
35  -         try {
36  -             in = new Scanner(new File("branch/branches.txt"));
37  -         } catch (FileNotFoundException ex) {
38  -             Logger.getLogger(orderform.class.getName()).log(Level.SEVERE, null, ex);
39  -         }
40  //add the name of branches in the combo boxes
41  -         while(in.hasNext()){
42  -             //m=in.next();
43  -             m=in.nextLine();
44  -             combosender.addItem(m);
45  -             comborec.addItem(m);
46  -         }
47  /**
48  -     }
49  }
50 }
```

```
23 [+]  
27  public class orderform extends javax.swing.JFrame {  
28  
29 [+]  
30      /**...*/  
31      public orderform() {...}  
32  
33 [+]  
34      /**...*/  
35      @SuppressWarnings("unchecked")  
36      Generated Code  
37  
38 [+]  
39      private void tweightActionPerformed(java.awt.event.ActionEvent evt) {...}  
40  
41      //checks whether the form is correctly filled or not  
42      public boolean formisempty() {...}  
43      //correct for non integer entries  
44      void clearall() {...}  
45  
46      private void confirmActionPerformed(java.awt.event.ActionEvent evt) {...}  
47  
48      private void radiosmallActionPerformed(java.awt.event.ActionEvent evt) {...}  
49  
50      /**...*/  
51      public static void main(String args[]) {...}  
52  
53
```

## Open Page Form code snippet



```
24  public class openpage extends javax.swing.JFrame {
25
26  /**
27  * @Generated Code
28  */
29
30 /**
31 * @Generated Code
32 */
33 @SuppressWarnings("unchecked")
34 Generated Code
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
```

```
237
238  private void viewemployeeActionPerformed(java.awt.event.ActionEvent evt) {
239      employeed x=new employeed(0);
240      x.setVisible(true);
241  }
242
243  private void viewbranchwiseActionPerformed(java.awt.event.ActionEvent evt) {
244      branchrev form=new branchrev(1);
245      form.setVisible(true);
246      return;
247  }
248 //considering max volume of truck to be 10 cubic meter
249  private void showreadyActionPerformed(java.awt.event.ActionEvent evt) {
250      int i;
251      String m;
252      Double vol;
253      branch branch1=new branch();
254      branch1.name="Bangalore";
255      m=branch1.get_truckava();
256      if(m.compareTo("")==0){
257          JOptionPane.showMessageDialog(this,"Sorry no truck available currently");
258      }
259      else{
260          vol=branch1.get_current_vol();
261          m=branch1.get_truckava();
262          truckdispatch truckdispatch1 = new truckdispatch(m,vol);
263          truckdispatch1.setVisible(true);
264      }
265      return;
266  }
267
```

# Test Plan Document

## Version 1.1

### April 20, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

## **Items to be tested:**

- 1.) Price () - Refer Consignment class  
To check if given a form details the price function
- 2.) Update\_consignment\_database() – Refer Consignment Class.
- 3.) Confirm\_action() – Refer Order Form

## **Test/Fail Criteria:**

- 1.) **Price():**  
At each of its branch office and other retail outlets, the courier company receives consignments of various weights and sizes (measured in cubic meters). The charges are at present Rs. 5,000/- per cubic meter for distances up to 500km. For larger distances, 10% additional charge is levied for every 100km or part thereof. For packets weighing more than 100kg per cubic meter, an additional 10% levy is charged for every 20kg/cubic meter. For small articles and letters, Rs.50/- per 100gms is charged.  
**The function should satisfy the above given criteria.**
- 2.) **Update\_consignment\_database():**  
On call of this function a new file with integer ID should be created with consignments detail written in it in the following order.
  - a.) Consignment ID
  - b.) Price
  - c.) Weight

- d.) Volume
- e.) Distance
- f.) Order date
- g.) To (Receiving person's address)
- h.) From (Sending person's address)
- i.) Sender branch
- j.) Receiving branch

3.) **Confirm\_action():**

To check if the form is completed with proper details. (Refer the Order form Class and Order in SRS).

If the format doesn't matches than fail.

**Features Not to Be Tested:**

All the functions taken from the java library and exception handling not to be tested.

Software  
Project Management Plan  
Version 1.1  
April 20, 2011

Courier company computerization (CCC) software  
Pranjal Sahu  
Roll No.-09CS1036

Submitted in partial fulfillment  
Of the requirements of  
CS20006 (Software Engineering)

## Project Plan:

### Estimation of Size:

From our previous experience of ERP software we can guess the size of this software:

- 1.) Three core users namely student, administrator and teacher each contributing ~1KLOC.
- 2.) 2 helping forms with each contributing ~200LOC.

Therefore an approximate total of  $3 \times 1$  KLOC +  $2 \times 200$  LOC i.e. approximately 3.4 KLOC.

In our present project we have:

- 1.) 4 main classes for truck, account, employee, branch each of ~500 LOC.
- 2.) 10 helping forms with each of ~300 LOC for view truck status, update truck status, show ready truck, view consignment status, order, view branch wise revenue/expense, view profit, view employee details, enter receipts, generate pay slip,

Therefore we estimate the total size of our software to be  $(4 \times 500 + 10 \times 300)$  LOC ~ **5 KLOC**.

Since the software is of organic category, therefore the coefficients used are-

Software project	$a_b$	$b_b$	$c_b$	$d_b$
<b>Organic</b>	2.4	1.05	2.5	0.38

$$\text{Effort Applied (E)} = a_b(KLOC)^{b_b} \quad [\text{man-months}]$$

$$\text{Development Time (D)} = c_b(\text{Effort Applied})^{d_b} \quad [\text{months}]$$

$$\text{People required (P)} = \text{Effort Applied} / \text{Development Time} \quad [\text{count}]$$

Therefore in our software:

$$\text{Effort Applied (E)} = 2.04(5)^{1.05} = 11.05 \quad [\text{man-months}]$$

$$\text{Development Time (D)} = 2.5(11.05)^{-0.38} = 6.23 \quad [\text{months}]$$

$$\text{People required (P)} = \text{Effort Applied} / \text{Development Time} \quad [\text{count}]$$

$$= 11.05 / 6.23 = 1.77 \text{ OR } 2$$

## In case of our project Management plan is-

### Start of Project:

Wednesday, 09 February 2011

<b>S. No.</b>	<b>Project Module</b>	<b>Date of Completion</b>	<b>~Days</b>	<b>Priority</b>
1.	SRS Document with DFDs	Sunday, 06 March 2011.	27	Highest
2.	Functional oriented design - Architecture - Top-level design - Detailed design	Tuesday, 15 March 2011.	13	Moderate
3.	UML diagrams - Use Case Diagram - Class Diagram - Sequence Diagram	Tuesday, 05 April 2011.	21	Highest
4.	Coding	Sunday, 17 April 2011	26	Higher
5.	Deployment	Wednesday, 20 April 2011.	20	Higher

Note: More than one process will be done in parallel.