

Sri Lanka Institute of Information Technology

Supply Chain Management(SCM).

**Project Report**

Information Technology Project 2017

Project ID: ITP-17-MLB-09

Submitted by:

|  |  |
| --- | --- |
| IT16097902 | Velivitiya G.C. |
| IT16225848 | K.A.D.L.S. Wijayawickrama |
| IT16131866 | W.D.I.Fernando |
| IT16125094 | S.S.D.Peellapitiya |
| IT16112292 | Devindi D.G.L |
| IT16112674 | H.M.C.S. Wijayananda |
| IT16111516 | K.A Kalana Bimsara |
| IT16110380 | Dharmasena P.H.R |

Submitted to:

…………………………..

Yashodya Wijesinghe

20.11.2017

Abstract

Emerging media (Pvt)Ltd was a company formed in 2009. Idea behind its formation was to make a revolution in the field of advertising of Sri Lanka. The company introduced the bus tv and indoor advertising firstly. And with the advancement of it they introduced LED advertising for the first time in Sri Lanka, which was a great benefit for the companies who were with the need of an effective and large-scale advertising. At present, Emerging media is the market leader in digital signage and location based advertising with most number of its screens and touch points throughout Colombo. It also does events such as musical shows, movie launchers, beauty pageants. They are hold either under the basis fully coverage or sponsorship wise. Its head office is situated in the World Trade Center at building Bank of Ceylon Mawatha Colombo Sri Lanka.

The company is in need of a fully functional software system for their supply chain management(SCM). Besides they are maintaining most of the process related to SCM manually (ex:- excel sheets ). And due to the unavailability of a complete interconnected (chained) system for their functions, they are facing a problem of improper(inconvenient) process handling. The absence of a central data base(DB) for company’s data, causing them to be at a risk information loss and lesser security. And with the consideration of other issues as well (which are hoped to be discussed later in this document), Our solution for these unfavorable circumstances is a software product which consists of following 8 main functions as follow.

* Project Management
* Event Management
* Client Management
* Employee Management
* Supplier Management
* Financial Management
* Inventory and ware house management
* Transportation Management

Acknowledgement

We have taken efforts in this project. However, it would not have become a success without the guidance and supervision of many individuals and organizations.

First and foremost, we have to thank our project supervisor Ms Yashodya Wijesinghe. Without her dedicated involvement and supervision though out the project, this would not have become a success at all.

Also we would like to thank the Managing Director of Emerging Media Mr Sanjeewa Rajapakse for handing over the project to our group with a value of trust. Our special thank goes to Software Engineer team lead Mr Thranga Kodithuwakku of the company for guiding us throughout the project and providing necessary information.

Finally our thank goes to our group members for their collaboration and commitment towards the completion of the project.

Declaration

We declare that the this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

|  |  |
| --- | --- |
| Project Title | Supply chain management system |
| Project ID | **ITP-17-MLB-09** |

Group Members

|  |  |  |
| --- | --- | --- |
| **Reg. No** | **Name** | **Signature** |
| IT16097902 | Velivitiya G.C. |  |
| IT16225848 | K.A.D.L.S. Wijayawickrama |  |
| IT16131866 | W.D.I.Fernando |  |
| IT16125094 | S.S.D.Peellapitiya |  |
| IT16112292 | Devindi D.G.L |  |
| IT16112674 | H.M.C.S. Wijayananda |  |
| IT16111516 | K.A Kalana Bimsara |  |
| IT16110380 | Dharmasena P.H.R |  |

Table of Contents

Abstract i

Acknowledgement ii

Declaration iii

Table of Contents iv

List of Figures v

List of Tables v

List of Acronyms and Abbreviations v

1. Introduction 1

1.1 Problem Statement 1

1.2 Product Scope 1

1.3 Project Report Structure 1

2. Methodology 2

2.1 Requirements and Analysis 2

2.2 Design 2

2.3 Implementation 2

2.4 Testing 2

3. Evaluation 3

3.1 Assessment of the Project results 3

3.2 Lessons Learned 3

3.3 Future Work 3

4. Conclusion 4

5. References 5

Appendix A: Design Diagrams 6

Appendix B: Test Results 7

Appendix C: Selected Code Listings 8

**List of Figure**

Page No

Figure 2.1.1: Overall Use Case Diagram .......................................................................... 06

Figure 2.1.2: Sales Management Use Case ....................................................................... 07

Figure 2.1.3: Style Planning & Resource Allocation Use ................................................. 11

Figure 2.1.4: Quality Checking Use Case ......................................................................... 22

Figure 2.1.5: Employee Management System Use Case ................................................... 25

Figure 2.1.6: Maintenance Management & Event Handling Use Case ............................. 31

Figure 2.1.7: Account Management Use Case .................................................................. 41

Figure 2.1.8: Stock Management Use Case ...................................................................... 45

Figure 2.1.9: Transportation Management Use Case ........................................................ 49

Figure 1.0: EER Diagram ................................................................................................. 56

Figure 2.0: Class Diagram of Style and Resources Planning & Resource Allocation ...... 57

Figure 3.0: Class Diagram of Maintenance Management & Event Handling ................... 58

Figure 4.0: Class Diagram of Return Goods Handling & Quality Checking .................... 59

Figure 5.0: Class Diagram of Employee Management ..................................................... 60

Figure 6.0: Class Diagram of Accounts Management ...................................................... 61

Figure 7.0: Class Diagram of Transport Management .................................................... 61

Figure 8.0: Class Diagram of Raw Material & Supplier Management ........................... 62

Figure 9.0: Class Diagram of Sales Management ........................................................... 62

Figure 10.0: Home Page .................................................................................................... 64

Figure 11.0: Style Registration New Style Tab ................................................................. 65

Figure 12.0: Style Registration Add Quantity Tab (Submit Validations) ......................... 65

Figure 13.0: Style Resource Difference (Search) .............................................................. 66

Figure 14.0: Style Resource Allocation (Choose Button Validations) ............................. 66

Figure 15.0: Style Resource Allocation (Choose) ............................................................. 67

Figure 16.0: Style Resource Allocation (Assign Button Validation) ................................ 67

Figure 17.0: Style Resource Allocation (Remove Button Validation) .............................. 68

Figure 18.0: Send Email .................................................................................................... 68

Figure 19.0: Accessary Inspection (Submit Button Validation) ....................................... 69

Figure 20.0: Final Product Inspection (Delete Button Validation) ................................... 69

Figure 21.0: Fabric Inspection (Submit Button Validation).............................................. 70

Figure 22.0: Packing Inspection (Update Button Validation) .......................................... 70

Figure 23.0: Activity Diagram of Sales Management .................................................... 119

Figure 24.0: Activity Diagram of Style Planning & Resource Allocation ...................... 120

Figure 25.0: Activity Diagram of Stock Management ................................................... 121

Figure 26.0: Activity Diagram of Account Management .............................................. 122

**List of Tables**

Page No

Table 01: List of Abbreviations & Acronyms .................................................................... vi

Table 02: Test Results of Login Page ............................................................................. 123

Table 03: Test Results of Home Page ............................................................................. 123

**List of Acronyms and Abbreviations**

1. **Introduction**

**1.1 Problem statement**

Despite being a company specializing in an innovative field of technology, Emerging Media functions with the use of outdated techniques for its systems. With the exception of a few processes, the majority of Emerging Media’s systems function manually, using data stored on hard copies. Even the few processes that are carried out using software have to be carried out manually, and these processes are not linked to automatically share data and resources. And thus, the inaccuracies and time consumption that entail manual processing and data transfer has resulted in a far below optimum level of efficiency. The specific issues that cause this lack of efficiency have been identified as follows.

* Corruption of data during storage and transfer. Since most tasks require employees to store crucial data and transfer this data from process to process and from section to section, the high probability of occurrence of human errors tend to cause data to be corrupted and result in potentially massive losses of both profit, and time due to the necessity of identifying and correcting these errors.
* Necessity of a large workforce to tackle a small workload. A myriad manual task means copious amounts of time spent by employees engaged in actions like entering and retrieving data. An interconnected and automated system would not only carry out these tasks instantaneously, but also reduce the true workload to be handled by employees to a surprisingly small amount. This means that the company spends an unnecessary amount on an excessive number of employees in order to obtain an unsatisfactory output.
* Inability to modify data. Apart from a few tasks carried out using manually operated software like MS Excel, the company’s records are kept using physical, written media. This means the data thus stored cannot be modified significantly. And so, oftentimes it is necessary to store create new, unnecessary records to overcome this problem, and even recreate existing records with slight modifications. As the company’s number of tasks increase and become more complex, it becomes difficult to store the growing cache of physical records and even more difficult to retrieve particular pieces of data from these records.
* Data theft and intentional corruption of records. Because of the size of the cache of physical records, it is impossible to double check the connections between the sets of data across all the records on a regular basis. Therefore, an employee or outsider with access to this cache can modify certain vital records with malicious intent, often for his or her financial gain. There is also a possibility of data important for future decisions or information pertaining to the internal affairs of the company being leaked to outside sources, possibly to competitors, which in turn can cripple the company.

**1.2 Product Scope**

The supply chain management system is coming with the centralized and large database. It also give a simple interface to user to handle the system easily. It makes sure that the data access is reliable and efficient also overcome from the existing system. It helps in mining and controlling of all the activities and operations being carried out in the company. It also maintains the record of the data of the entire employees which can be used as and when required. The supply chain management software helps in providing various tools which helps in effective controlling and functioning of company. Main advantage of the system is to centralize all the functions of the system such as Staff management system which maintains staff details, Customer management system which maintains Customer details, Transport management which maintains transport and vehicle details, Finance management system which maintains money related details, Inventory management system which maintain inventory details, supply management system which manage all the stock details, project management system which manage the all project details and event management system handle all the events . The benefits of our systems are

Centralized database

User friendly GUI

Easy to handle without extra knowledge in IT

No redundancy

Easy calculations

Ensure security

**1.3 Project report structure**

1. **Methodology**
   1. **Requirements and analysis**

The requirements were gathered via observations and interviews. Though the company has separate software’s for its processes, the lack of an integrated module is a fact which makes the tasks harder to the control. Therefore, Supply Chain Management system for Emerging Media was the best solution to overcome this inconvenience.

* Project Management

This function is responsible for maintaining details related to company’s projects and assigning proper resources for its execution.

* Event Management

This function is responsible for maintaining details related to events hold by the company, booking tickets and hiring equipment’s for events which are outside the companies’ control.

* Customer Management

In here the handling of customer base is done. Maintaining the involvements of the customer. Handling details of the customer and customer orders, payments.

* Employee Management

This function is handling employee details including leave details, record attendance and payroll handling of the employees.

* Inventory Management

This function has stock handling, order handling and maintaining warehouse activity details.

* Transportation Management

This function is responsible of transporting raw materials and employees. This checks the availability of the vehicles and maintain the facts of the vehicles such as insurance, fuel etc.

* Finance Management

This function is responsible of calculating monthly sales and income in order to get the profit. And also whole company’s expenses and income.

* Supplier Management

In this function the details related to the suppliers of the company is maintained. Their identification details as well as what they supply to the company.

* + 1. **Full System Overview**

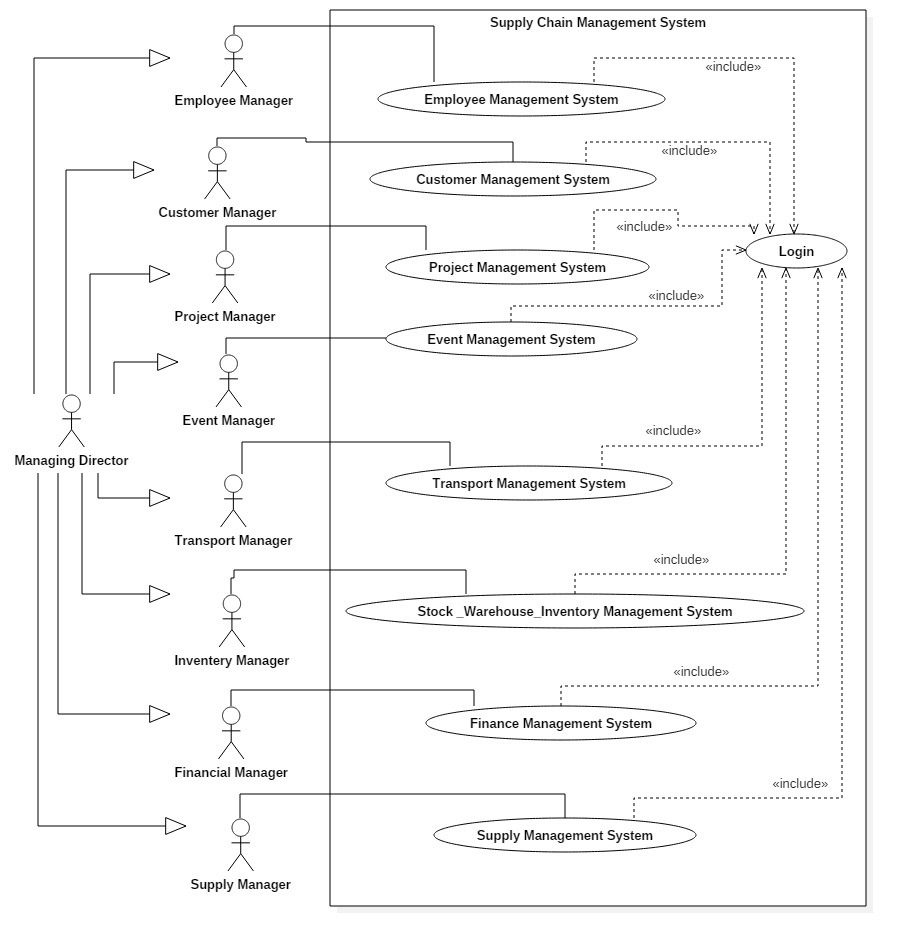


Figure 2.1.1 Overall Use Case Diagram

**2.1.2 Employee Management**

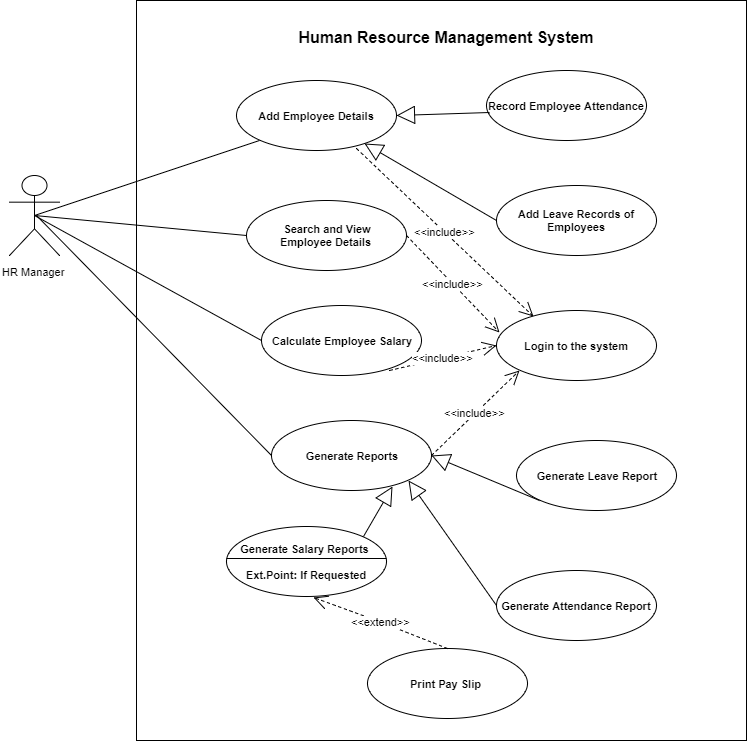


Figure 2.1.2 Employee Use Case Diagram

**2.1.2.1 Add Employee Details**

|  |
| --- |
| **Use Case ID:** HR-UC-1 |
| **Use Case:** Add Employee Details |
| **Brief Description:** This Use Case describes the procedure of adding Employee Details to the  system |
| **Primary Actor:**  HR Manager |
| **Pre-Condition:** HR Manager has logged in to the system |
| **Main Success Scenario:**  1. Select “Personal Details” Tab in the Menu Bar.  2. Click “Add Employee” Button.  3. Add the Employee Details in the given Text fields.  4. Click Submit Button.  5. If the details are Valid display Submission Confirmation Message to the HR Manager  6. If HR Manager confirms the Submission details will be added to the System.  7. Click “Clear” Button to add another Employee |
| **Extensions:**  3a Add invalid details in the Text fields  3a1. Display proper Error Messages below the relevant Text fields when moving to another  text field.  4a Add invalid details in the Text fields  4a1. Display proper Error Messages below the relevant Text fields.  6a Manager rejects the confirmation or doesn’t confirm the Submission  6a1. Details will not be added to the system  7a HR Manager doesn’t need to add another Employee  7a1. Click “Exit” Button.  7a2. Log out from the system or select another tab from the Menu Bar |

|  |
| --- |
| **Use Case ID:** HR-UC-2 |
| **Use Case:** Add Leave Record |
| **Brief Description:** This Use Case describes the procedure of adding Leave Records to the  system |
| **Primary Actor:**  HR Manager |
| **Pre-Condition:** HR Manager has logged in to the system |
| **Main Success Scenario:**  1. Select “Apply Leave” Tab in the Menu Bar.  2. Enter the Employee Code  3. Display leaves left for the Employee in the grid.  4. Select the leave type.  5. Selects the “half day” or “full day” and select the dates of applying leave.  6. Click “Add” button.  7. If the details are valid, leaves left for the Employee is updated and leave will be added to the “Leaves taken” grid.  8. Click submit button.  9. If the details are Valid display Submission Confirmation Message to the HR Manager  10. If HR Manager confirms the Submission details will be added to the System. |
| **Extensions:**  3a Employee Code is Invalid  3a1. Display proper Error Messages below the Text field when moving to the next  text field.  3b Employee’s Office Details has not entered to the system  3b1. Display proper Error Messages below the Text field when moving to the next  text field.  5a If “half day” selects after selecting the leave type as annual, or sick leave.  5a1. Display proper Error Messages  5b If “full day” is selected and the employee is in his probation period.  5b1. Display proper Error Messages  6a. If the Employee has no any remaining leaves.  6a1. Display of asking of applying “No Pay”  9a. If the details are invalid or empty  9a1. Display error Messages |

**2.1.2.2 Add Leave Record**

**2.1.3 Inventory Management**

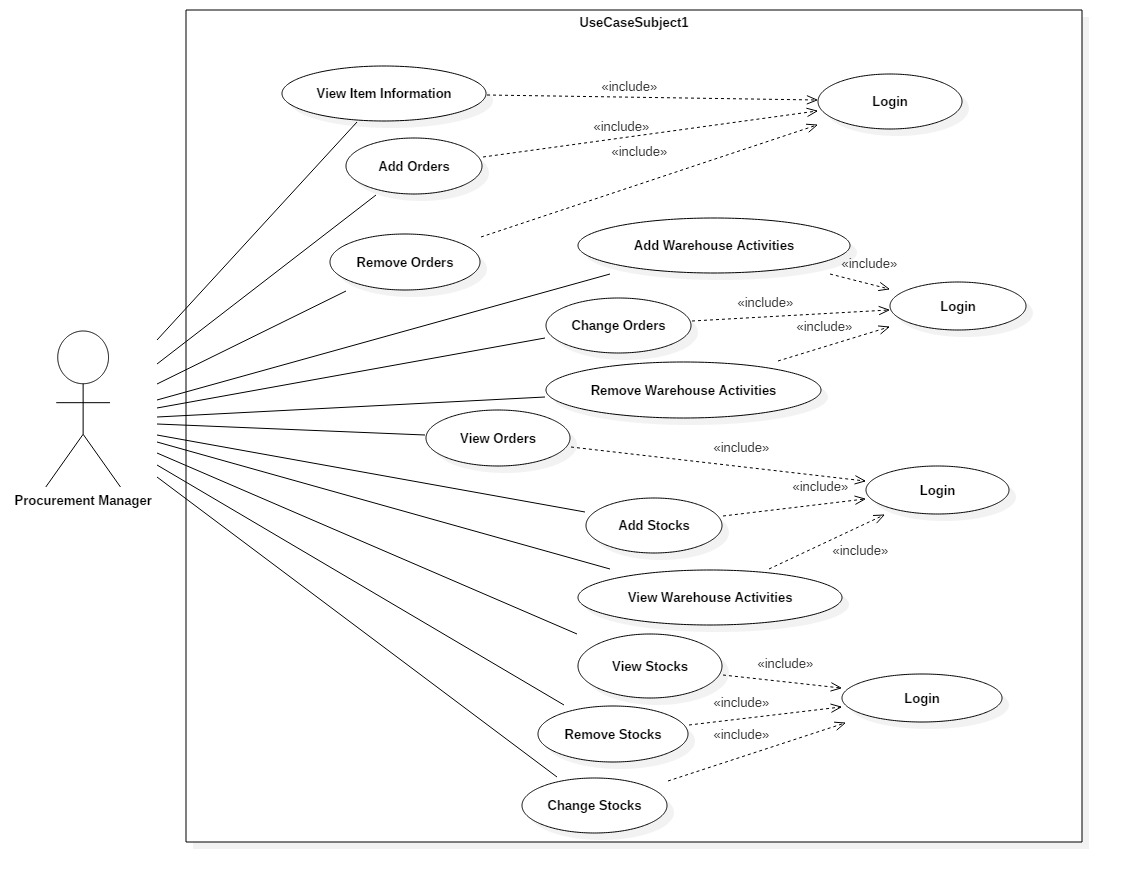


Figure 2.1.3 Inventory Use Case Diagram

**2.1.3.1 Add Orders**

|  |  |
| --- | --- |
| Use case name | Add Orders |
| Pre-conditions | Inventory manager should log in to the system and there must be a proper internet connection. |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the add order button 3. Add order form will be prompted. 4. User selects the items need to order and fills the quantity and click OK button 5. An auto-generated email will be sent to the supply manager and database and chart in the interface will be updated. |
| Extensions | 4.a. If user entered any invalid information on  form an error message will be displayed.  4.b. If user selects the same item again, after  previous selection an error message will be  displayed. |

**2.1.3.2 Remove Orders**

|  |  |
| --- | --- |
| User case name | Remove Orders |
| Pre-conditions | Inventory manager should log in to the system and there must be a proper internet connection. |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the remove order button 3. Remove order form will be prompted. 4. User selects the orders need to removed and click OK button 5. An auto-generated email will be sent to the supply manager and database and chart in the interface will be updated. |
| Extensions | 4.a. If user selects the same order again, after  previous selection an error message will be  displayed. |

**2.1.3.3 Change Orders**

|  |  |
| --- | --- |
| Use case name | Change Orders |
| Pre-conditions | Inventory manager should log in to the system and there must be a proper internet connection. |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the change order button 3. Change order form will be prompted. 4. User selects the orders need to change and click OK button after doing relevant changes. 5. An auto-generated email will be sent to the supply manager and database and chart in the interface will be updated. |
| Extensions | 4.a. If user entered any invalid information on  form an error message will be displayed.  4.b. If user selects the same order again, after  previous selection an error message will be  displayed. |

**2.1.3.4 View Orders**

|  |  |
| --- | --- |
| Use case name | View Orders |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the view order details in tabular form button. 3. Then he or she can view order details in tabular format |
| Extensions | 3.a. If there are no order data in database an  error message will be displayed.  3.b. If user pressed the button again he or she  can view relevant order data in a chart |

**2.1.3.5 Add Stocks**

|  |  |
| --- | --- |
| Use case name | Add Stocks |
| Pre-conditions | Inventory manager should log in to the system and there must be at least one order in the database. |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the add stock button 3. Add stock form will be prompted. 4. User selects the items that ordered and fills the quantity and click OK button 5. Database and chart in the interface will be updated. |
| Extensions | 4.a. If user entered any invalid information on  form an error message will be displayed.  4.b. If user selects the same item again, after  previous selection an error message will be  displayed. |

**2.1.3.6 Remove Stocks**

|  |  |
| --- | --- |
| User case name | Remove Stocks |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the remove stock button 3. Remove stock form will be prompted. 4. User selects the stock need to removed and click OK button. 5. Database and chart in the interface will be updated. |
| Extensions | 4.a. If user selects the same stock again, after  previous selection an error message will be  displayed. |

**2.1.3.7 Change Stock**

|  |  |
| --- | --- |
| Use case name | Change Stock |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the change stocks button 3. Change stock form will be prompted. 4. User selects the stock need to change and click OK button after doing relevant changes. 5. Database and chart in the interface will be updated. |
| Extensions | 4.a. If user entered any invalid information on  form an error message will be displayed.  4.b. If user selects the same stock again, after  previous selection an error message will be  displayed. |

**2.1.3.8 View Stocks**

|  |  |
| --- | --- |
| Use case name | View Stock |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the view stock details in tabular view button. 3. Then he or she can view stock details in tabular format |
| Extensions | 3.a. If there are no stock data in database an  error message will be displayed.  3.b. If user pressed the button again he or she  can view relevant stock data in a chart |

**2.1.3.9 Add Warehouse Activities**

|  |  |
| --- | --- |
| Use case name | Add Warehouse Activities |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the add warehouse activities button 3. Add warehouse activities form will be prompted. 4. User selects the appropriate vehicle from vehicle list and fill necessary information and click OK button 5. Database and chart in the interface will be updated. |
| Extensions | 4.a. If user entered any invalid information on  form an error message will be displayed.  4.b. If user selects the same vehicle again, after  previous selection an error message will be  displayed. |

**2.1.3.10 Remove Warehouse Activities**

|  |  |
| --- | --- |
| User case name | Remove Warehouse Activities |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the remove warehouse activities button 3. Remove warehouse activities form will be prompted. 4. User selects the warehouse activities need to removed and click OK button 5. Database and chart in the interface will be updated. |
| Extensions | 4.a. If user selects the same warehouse activity  again, after previous selection an error  message will be displayed. |

**2.1.3.11 View Warehouse Activities**

|  |  |
| --- | --- |
| Use case name | View Warehouse Activities |
| Pre-conditions | Inventory manager should log in to the system |
| Primary actor | Procurement manager |
| Main flow | 1. Include ::<login> 2. Use case begin when procurement manager clicks the view warehouse activity details in tabular form button. 3. Then he or she can view warehouse activity details in tabular format |
| Extensions | 3.a. If there are no warehouse activity data in  database an error message will be  displayed.  3.b. If user pressed the button again he or she  can view relevant warehouse activity data  in a chart |

**2.1.4 Event Management**

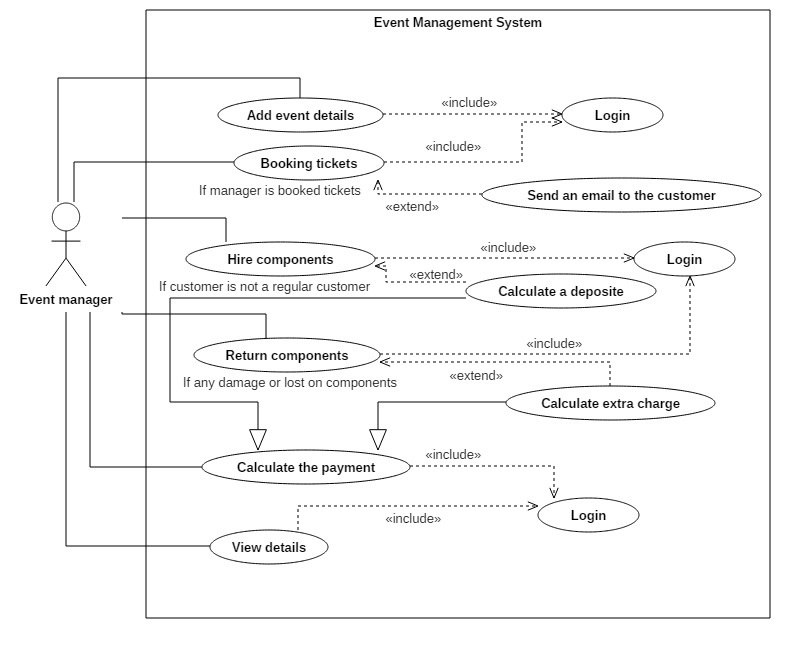


Figure 2.1.4 Event Management Use Case

##### 2.1.4.1 Booking Tickets

|  |  |
| --- | --- |
| **Use Case Name** | Booking Tickets |
| **Brief Description** | Event manager can book tickets for the events according to the customer requirements. |
| **Primary Actor** | Event manager |
| **Main Scenario**  **Extensions** | 1. Require: Login 2. The Event manager can fill the form in the GUI by adding relevant details of the customer and calculate the amount. 3. A message saying details of the customer have been successfully added to the database. 4. Send an email to the customer including the booking details 5. a.) If wrong EmpId or password given, system ask the user to re-enter. 6. a.) If any text field is not filled it will display an error message. 7. a.) If booking details have not been successfully added to database display and error message. 8. a.) If the receiver’s email address is invalid or email is not send it will display an error message. |

**2.1.4.2 Add An Event**

|  |  |
| --- | --- |
| **Use Case Name** | Add an event |
| **Brief Description** | Event manager can form events by assigning the relevant details about the event |
| **Primary Actor** | Event manager |
| **Pre-Conditions** | Approval for the event from head office |
| **Main Scenario**  **Extensions** | 1. Require: Login 2. the Event manager can fill the form in the GUI by adding relevant details about the event 3. A message saying event details have been successfully added to the database. 4. a.) If wrong EmpId or password given, system ask the user to re-enter. 5. a.) If any text field is not filled it will display an error message. 6. a.) If event details have not been successfully added to database display and error message. |

##### 2.1.4.3 Hire Components

|  |  |
| --- | --- |
| **Use Case Name** | Hire Components |
| **Brief Description** | Event manager record the details of hired components. |
| **Primary Actor** | Event manager |
| **Main Scenario**  **Extensions** | 1. Require: Login 2. Event manager select components related to the customer information and assign charges for hired components. 3. A message saying details of the hired component have been successfully added to the database. 4. a.) If wrong EmpId or password given, system ask the user to re-enter. 5. a.) If any text field is not filled it will display an error message.   2. b.) If a customer is not a regular customer they have to make a deposit.   1. a.) If hired components details have not been successfully added to database display and error message |

**2.1.4.4 Payment**

|  |  |
| --- | --- |
| **Use Case Name** | Payment |
| **Brief Description** | Event manager calculate the total amount for hired components and add transport charges and calculate full payment. |
| **Primary Actor** | Event manager |
| **Main Scenario**  **Extensions** | 1. Require: Login  2. The Event manager can fill the form in the GUI by adding relevant payment details of the hired components  3. A message saying details of the payment have been successfully added to the database.  1. a.) If wrong EmpId or password given, system ask the user to re-enter.  2. a.) If any text field is not filled it will display an error message.  3. a.) If return components details have not been successfully added to database display and error message |

##### 2.1.4.5 Return Components

|  |  |
| --- | --- |
| **Use Case Name** | Return Components |
| **Brief Description** | Event manager check the hired components details and compare with return components details |
| **Primary Actor** | Event manager |
| **Main Scenario**  **Extensions** | 1. Require: Login  2. The Event manager can fill the form in the GUI by adding relevant details of the return components and check the damage and lost and calculate the amount.  3. A message saying details of the return components have been successfully added to the database.  1. a.) If wrong EmpId or password given, system ask the user to re-enter.  2. a.) If any text field is not filled it will display an error message.  2. b.) If any damage or lost in the return components extra charge will be calculate and added to the total amount.  3. a.) If return components details have not been successfully added to database display and error message |

##### 2.1.4.6 View Details and Update and Delete

|  |  |
| --- | --- |
| **Use Case Name** | View Details and Update and Delete |
| **Brief Description** | Event manager can view the details events, bookings, hired and return components and payments. He can update or delete the data from the database. |
| **Primary Actor** | Event manager |
| **Main Scenario**  **Extensions** | 1. Require: Login  2. The Event manager can view, update or delete details using datagrideview.    1. a.) If wrong EmpId or password given, system ask the user to re-enter. |

**2.1.5 Project Management**

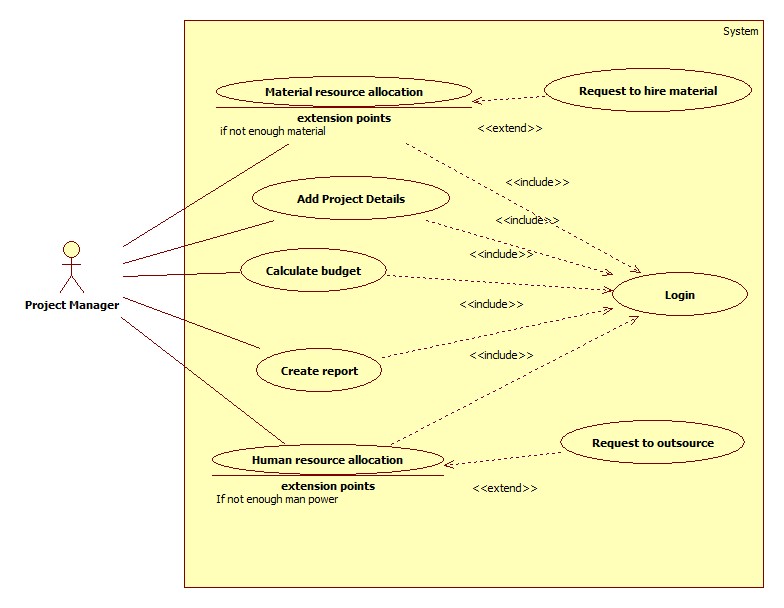


Figure 2.1.5 Project Management Use Case

**2.1.5.1 Material Resource Allocation**

|  |  |
| --- | --- |
| Use Case Name | Material Resource Allocation |
| Pre-Conditions | System is up and running, user is logged in with correct credentials |
| Primary Actor | Project Manager |
| Main Success Scenario | 1.User selects the Project  2.Selects the Particular Subtask to allocate the materials.  3.Fill in the details necessary for the allocation process  4.Submit details and obtain the Allocated ID. |
| Extensions | 2.a) If no Subtask added for the particular project, user will not have options to select.  3.a) if valid details are not entered prompts error message to the user. |

**2.1.5.2 Human Resource Allocation**

|  |  |
| --- | --- |
| Use Case Name | Human Resource Allocation |
| Pre-Conditions | System is up and running ,user is logged in with correct credentials. |
| Primary Actor | Project Manager |
| Main Success Scenario | 1.User selects the Project  2.Selects the Particular Subtask to allocate the materials.  3.Fill in the details necessary for the allocation process  4.Submit details and obtain the Job ID.  5.Send an email to the necessary employees assigned for the project. |
| Extensions | 2.a) If no Subtask added for the particular project, user will not have options to select.  3.a) if valid details are not entered, prompts error message to the user. |

**2.1.5.3 Calculate Budget**

|  |  |
| --- | --- |
| Use Case Name | Calculate Budget |
| Pre-Conditions | System is up and running, user is logged in with correct credentials |
| Primary Actor | Project Manager |
| Main Success Scenario | 1.User selects Project  2.Enters taxation percentage  3.Calculate budget  4.Use case ends once the user enters the submit button |
| Extensions |  |

**2.1.5.4 Create Report**

|  |  |
| --- | --- |
| Use Case Name | Create Report |
| Pre-Conditions | System is up and running, user is logged in with correct credentials |
| Primary Actor | Project Manager |
| Main Success Scenario | 1.Select |
| Extensions |  |

**2.1.6 Customer Management**

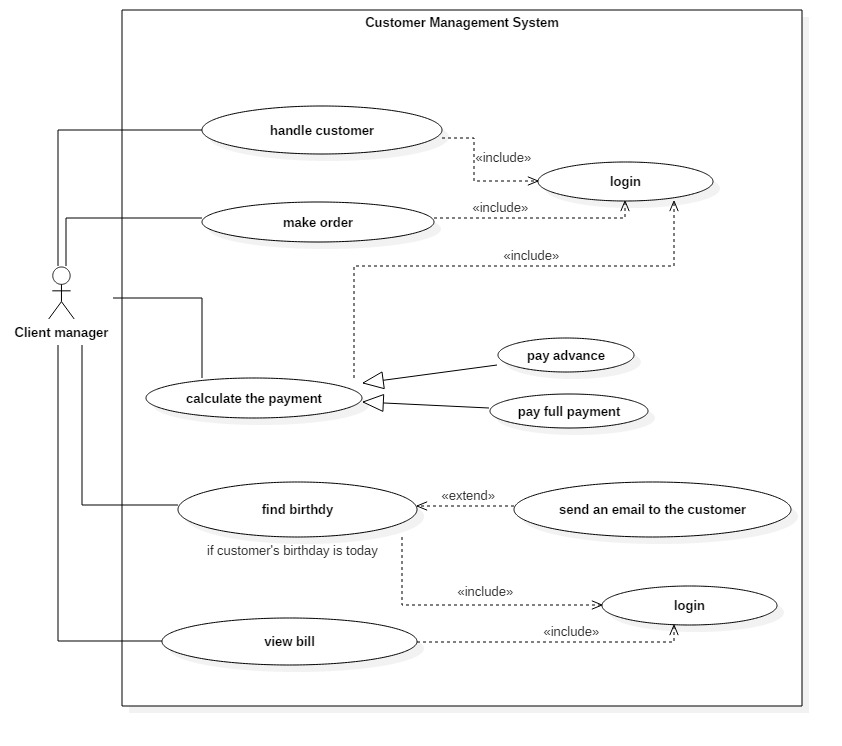


Figure 2.1.6 Customer Management Use Case

|  |  |
| --- | --- |
| Use Case Name | Handle customer |
| Brief Description | Client manager can Add a customer, update a customer, remove a customer and search relevant data about a customer. |
| Primary Actor | Client manager |
| Pre-Condition |  |
| Main Scenario | 1. Require: login  2. The Client manager can fill the form in GUI by adding relevant data.  3. Search a customer and make any changes to the form then update details.  4. Search and delete a customer  5. A message saying customer details have been successfully saved. |
| Extensions | 1.If any text field is not entered it will show an error message |

**2.1.6.1 Handle Customer**

**2.1.6.2 Calculate Payment**

|  |  |
| --- | --- |
| Use Case Name | Calculate payment |
| Brief Description | After calculate the bill prepare bill for advance payment. Then prepare bill for full payment |
| Primary Actor | Client manager |
| Pre-Condition | Make an order |
| Main Scenario | 1.make an order  2.prepare an advance bill  3.prepare full payment bill  4.A message saying payment is done successfully |
| Extensions |  |

**2.1.6.3 Find Birthday**

|  |  |
| --- | --- |
| Use Case Name | find birthday |
| Brief Description | Find weekly birthdays and daily birthdays of customers |
| Primary Actor | Client manager |
| Pre-Condition |  |
| Main Scenario | 1.find today birthdays  2.send an email to the customer |
| Extensions |  |

**2.1.6.4 View Bill**

|  |  |
| --- | --- |
| Use Case Name | View bill |
| Brief Description | After complete full payment client manager is able to view bills |
| Primary Actor | Client manager |
| Pre-Condition | Complete full payment |
| Main Scenario | 1.pay advance  2.pay full payment |
| Extensions |  |

**2.1.6.5 Make Order**

|  |  |
| --- | --- |
| Use Case Name | Make order |
| Brief Description | When customer select an item get all details and prepare a bill to be paid. |
| Primary Actor | Client manager |
| Pre-Condition | Should be a registered customer |
| Main Scenario | 1. Require: login  2. select items  3. send selected items to the inventory manager  4. calculate the bill |
| Extensions | 1.If any necessary field is not entered it will show an error message |

**2.1.7 Transport Management**

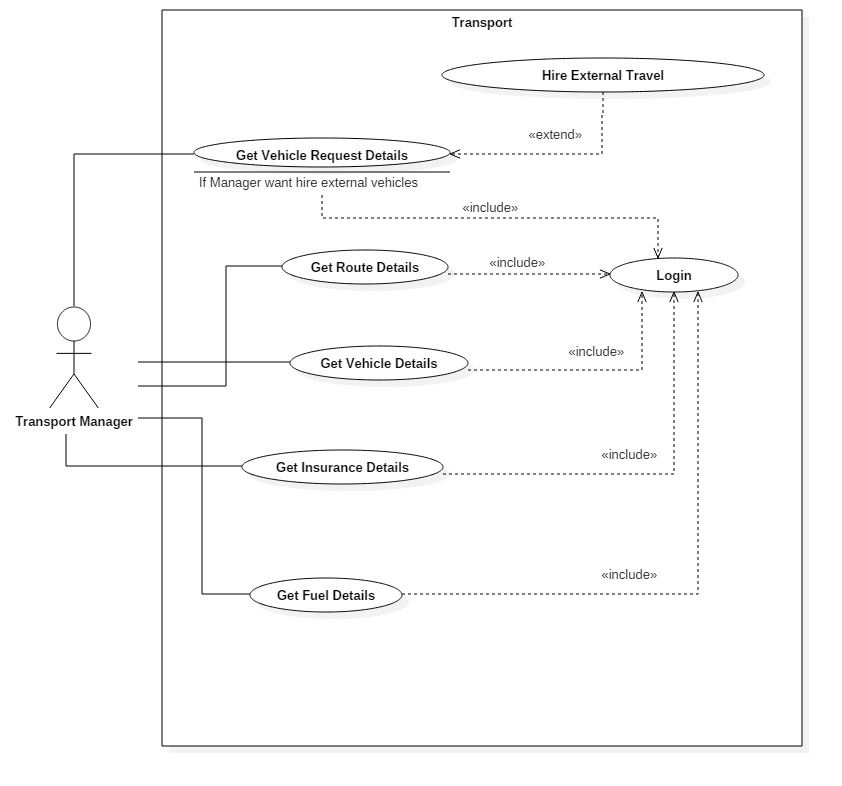


Figure 2.1.4 Transport Management Use Case

**2.1.7.1 Get Route Details**

|  |  |
| --- | --- |
| Use case Name | Get Route details |
| Pre-condition | System is up and running, transport manager should logged into the system properly. |
| Primary Actor | Transport Manager |
| Main Scenario | 1. Use case begins when the transport manager successfully logs into the system. |
| 1. Transport Manager selects the transportation option. |
| 1. Include:(External vehicles) |
| 1. Transport Manager fill in the relevant fields and submits |
| Extensions | 1. a) if invalid details entered, display an error message |

**2.1.7.2 Get Fuel Details**

|  |  |
| --- | --- |
| Use case Name | Get Fuel details |
| Pre-condition | System is up and running, transport manager should logged into the system properly. |
| Primary Actor | Transport Manager |
| Main scenario | 1. Use case begins when the transport manager successfully logs into the system. |
| 1. Transport manager get efficiency of transportation |

* + - 1. **Get Insurance Details**

|  |  |
| --- | --- |
| Use case Name | Get Insurance details |
| Pre-condition | System is up and running, transport manager should logged into the system properly. |
| Primary Actor | Transport Manager |
| Main Scenario | 1. Use case begins when the transport manager successfully logs into the system. |
| 1. Transport manager selects the insurance type and company name for insurance details |

**2.1.7.4 Get Vehicle Request Details**

|  |  |
| --- | --- |
| Use case Name | Get Vehicle Request details |
| Pre-condition | System is up and running, transport manger should logged into the system properly. |
| Primary Actor | Transport Manager |
| Main Scenario | 1. Use case begins when the transport manager successfully logs into the system. |
| 1. Transport manager get vehicle details . |
| 1. Transport manager select External Vehicle and get details(if hire external vehicles). |
| Extensions | 1. a)if invalid details entered, display an error message |

**2.1.8 Supplier Management System**

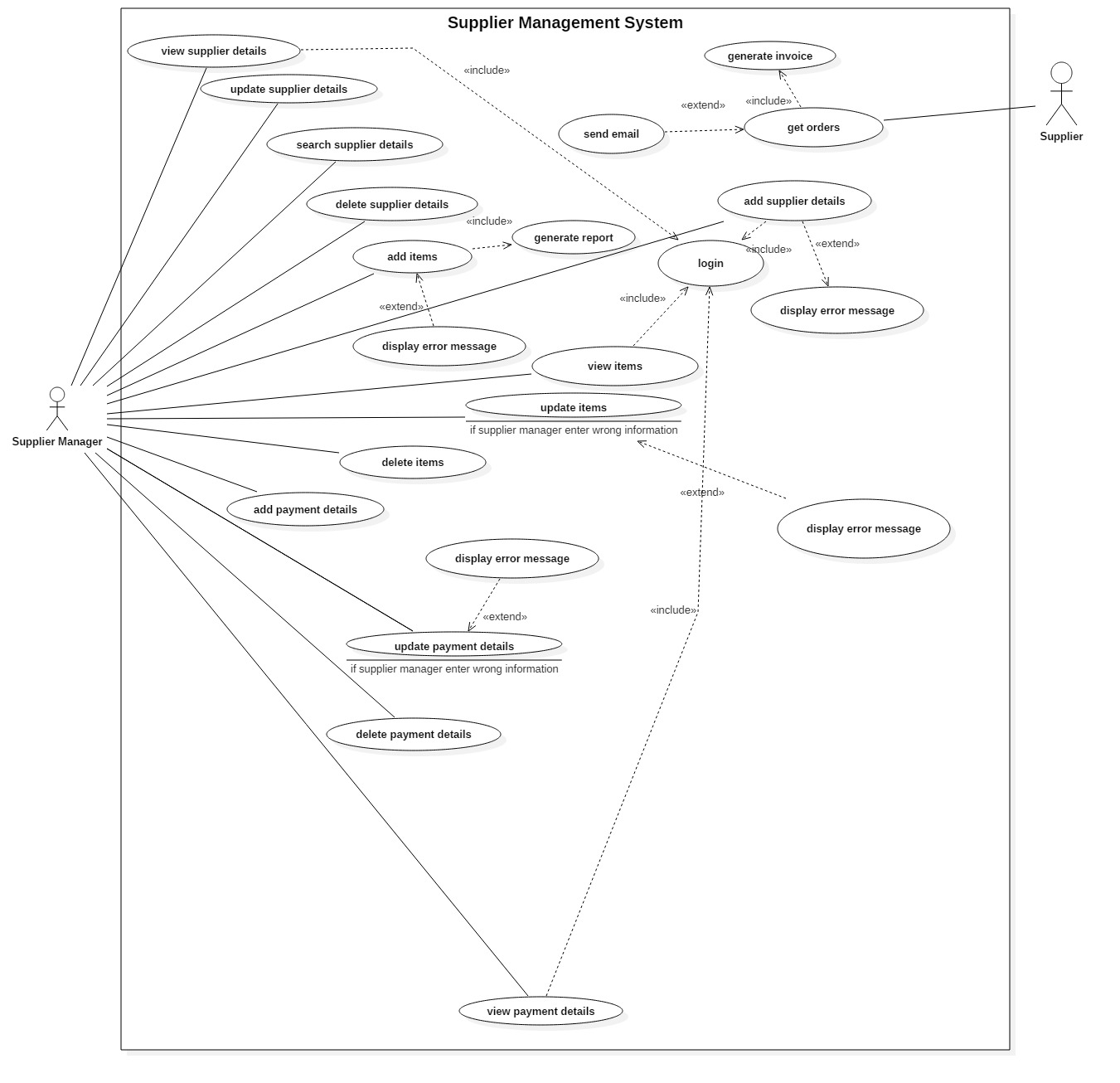


Figure 2.1.8 Supplier Management Use Case

**2.1.8.1 View Supplier Details**

|  |  |
| --- | --- |
| Use Case Name | View Supplier Details |
| Pre-Condition | The system is up and running supplier manager should be logged into the system |
| Primary Actor | Supplier Manager |
| Main Success Scenario | * Use case begin when supplier manager manage select the supplier management system * Supplier manager should be enter his login into correctly to the system * System will load the main interface * Manager manage relevant details about supplier details * Manager can choose search button according to the supplier specific details * All the due supplier details will be displayed |
| Extensions | If the login information which provided by the user is correct, error message will be displayed |

**2.1.8.2 Add Item Details**

|  |  |
| --- | --- |
| Use Case Name | Add Item Details |
| Pre-Condition | The system is up and running supplier manager should be logged into the system |
| Primary Actor | Supplier Manager |
| Main Success Scenario | * Login <<include>> * Generate report * Use case begin when supplier manager manage select the supplier management system * User can select add item details option * User can enter the relevant details about the items and save |
| Extensions | If the login information which provided by the user is correct, error message will be displayed  If the user enter wrong information, error message will be displayed |

**2.1.8.3 Update Item Details**

|  |  |
| --- | --- |
| Use Case Name | Update Item Details |
| Pre-Condition | The system is up and running supplier manager should be logged into the system |
| Primary Actor | Supplier Manager |
| Main Success Scenario | * Display error message <<extend>> * When the user place items wrong * When the order table update by supplier manager * User can select update button and place correct data |
| Extensions | If the login information which provided by the user is correct, error message will be displayed  If the user enter wrong information, error message will be displayed |

**2.1.8.4 Get Order Details**

|  |  |
| --- | --- |
| Use Case Name | Get Order Details |
| Pre-Condition | The system is up and running supplier manager should be logged into the system |
| Primary Actor | Supplier |
| Main Success Scenario | * Generate invoice <<include>> * Use case begin when supplier select the supplier management division * Supplier can get the order option * Supplier send the relevant details about order to supplier manager |
| Extensions | If the order is OK, supplier send a notification for supplier manager |

**2.1.8.5 View Payment Details**

|  |  |
| --- | --- |
| Use Case Name | View Payment Details |
| Pre-Condition | The system is up and running supplier manager should be logged into the system |
| Primary Actor | Supplier Manager |
| Main Success Scenario | * Login <<include>> * User can begin when the user select the supplier management division * When the user place payment details into the system and save * User can choose view button and payment details will be displayed |
| Extensions | If the login information which provided by the user is correct, error message will be displayed |

**2.1.9 Financial System**

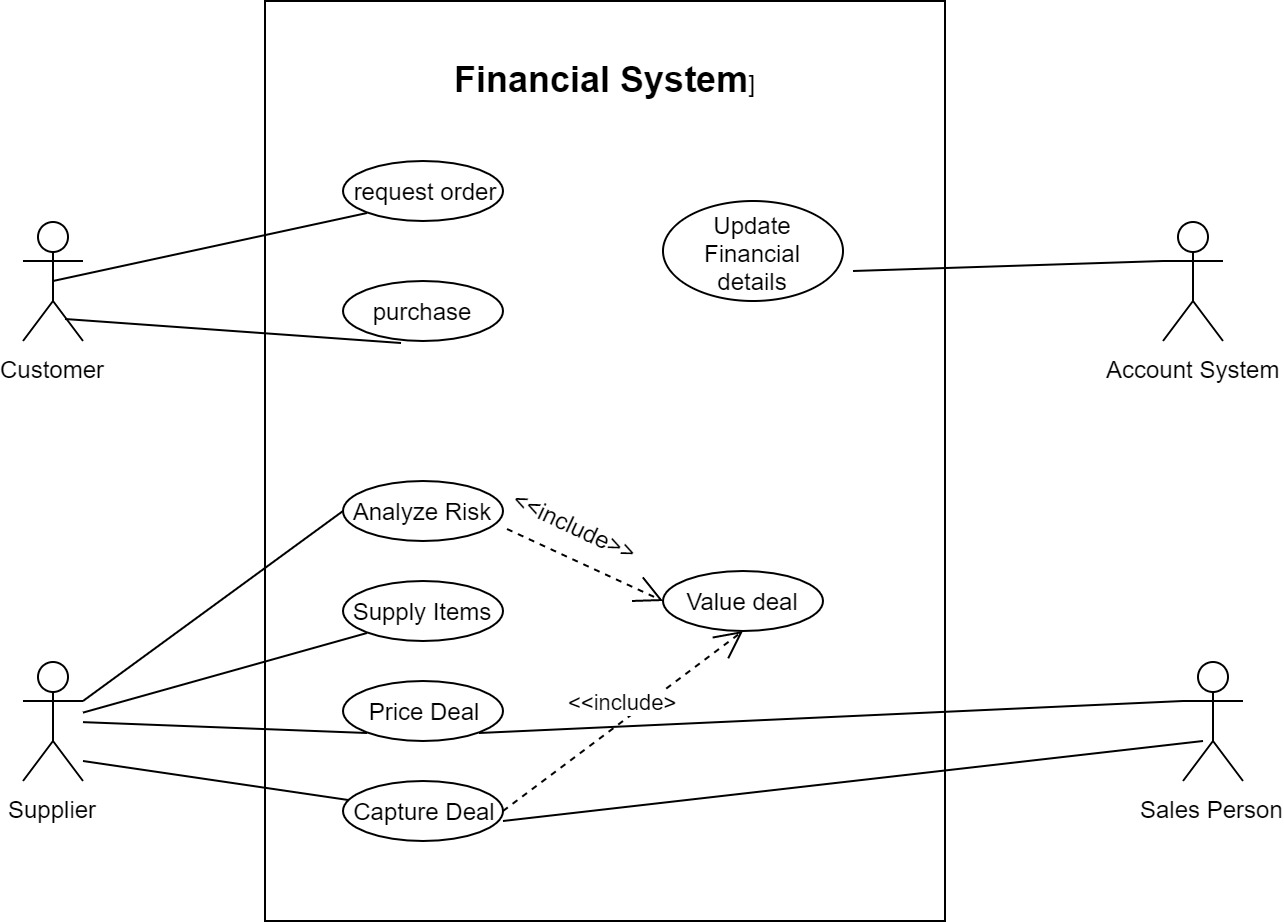


Figure 2.1.9 Financial System Use Case

|  |  |
| --- | --- |
| Use case name | Request order |
| Pre-conditions | Financial Manger should log into the system  And there must be a proper internet connection |
| Primary actor | Financial Manager |
| Main flow | 1. Include ::<login> 2. Use case begin financial manager   Enters data |
| Extensions | User cant enter wrong type values |

* + - 1. **Request Order**

**2.1.9.2 Purchase**

|  |  |
| --- | --- |
| User case name | purchase |
| Pre-conditions | Financial Manager should log in to the system and there must be a proper internet connection. |
| Primary actor | Financial Manager |
| Main flow | 1. Include ::<login> 2. Use case begin when Financial manager clicks the save button 3. then all of data will be saved |
| Extensions | If user entered same invoice no error will be displayed |

**2.1.9.3 Supply items**

|  |  |
| --- | --- |
| Use case name | Supply items |
| Pre-conditions | Financial should log in to the system and there must be a proper internet connection. |
| Primary actor | Financial Manager |
| Main flow | 1. Include ::<login> 2. Use case begin when Financial manager click   Purchase interface save button   1. save all the data of purchase details that entered |
| Extensions | If user entered same purchase invoice no error will be displayed |

**2.1.9.4 Price deal**

|  |  |
| --- | --- |
| Use case name | price deal |
| Pre-conditions | Financial should log in to the system and there must be a proper internet connection. |
| Primary actor | Financial Manager |
| Main flow | 1.Include ::<login>  2.Use case begin when FInancial manager  Clicks save button on expenditures interface  3.amount of expense calculate after that |
| Extensions | If user entered same expense no error will be displayed |

**2.1.9.5 Update financial details**

|  |  |
| --- | --- |
| Use case name | Update financial details |
| Pre-conditions | Financial manager should log in to the system and there must be at least one order in the database. |
| Primary actor | Financial manager |
| Main flow | 1. Include ::<login> 2. Use case begin when Financial manager clicks income statement button on main frame   3.all report will be displayed |
| Extensions |  |

**2.1.10 Non-Functional Requirements**

* **Security Requirements**

The user account security is ensured by passwords and their password will be secret and it can only access by them. All the data that are entered by user will save on a centralized server database. Emails are secured by the user.

* **Performance Requirements**

The system is efficient if and only if the computer where the system going to be install must be with a good hardware and software requirements. The user verification can done in few seconds and user can easily add or edit or remove data with no time consuming. Calculation parts also can be done less than 15 seconds and system doesn’t take longer than 20 sec to complete an action.

* **Software Quality Attributes**

**Usability:**

The system should provide uniform look and feel between all the user interfaces. Because of the user-friendly interfaces, the system makes easy to use and easy to get used to it too.

**Reliability and availability:**

The system should run at any instance without any obstacles. Unauthorized people cannot access the system.

**Correctness:**

Correct and accurate database should be maintained.

**2.2 Design**

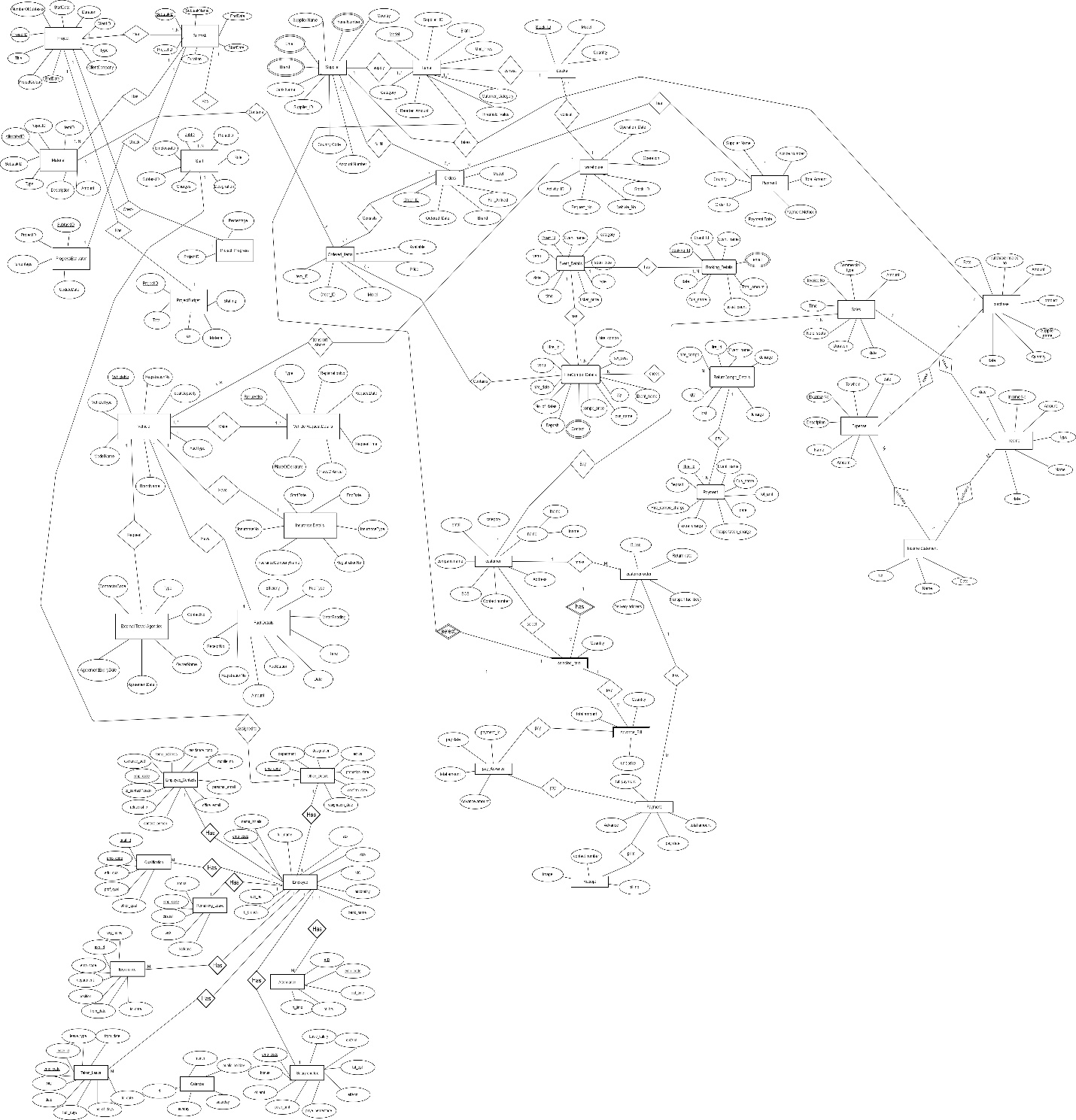
****

Figure 2.2.0: EER Diagram

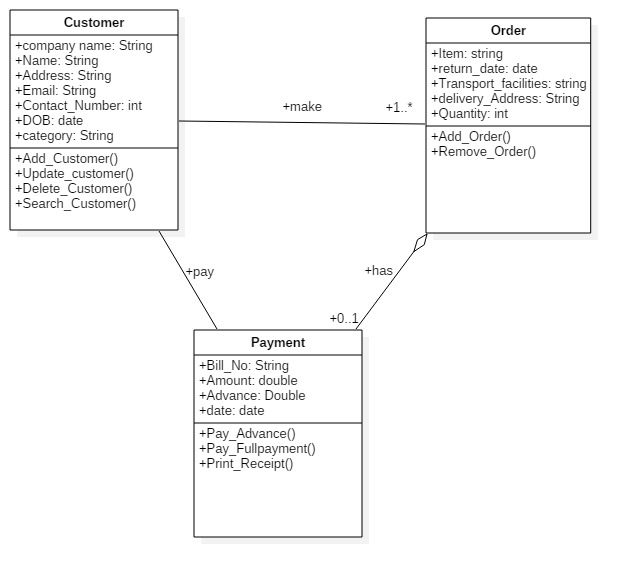
****

Figure 2.2.1: Customer management

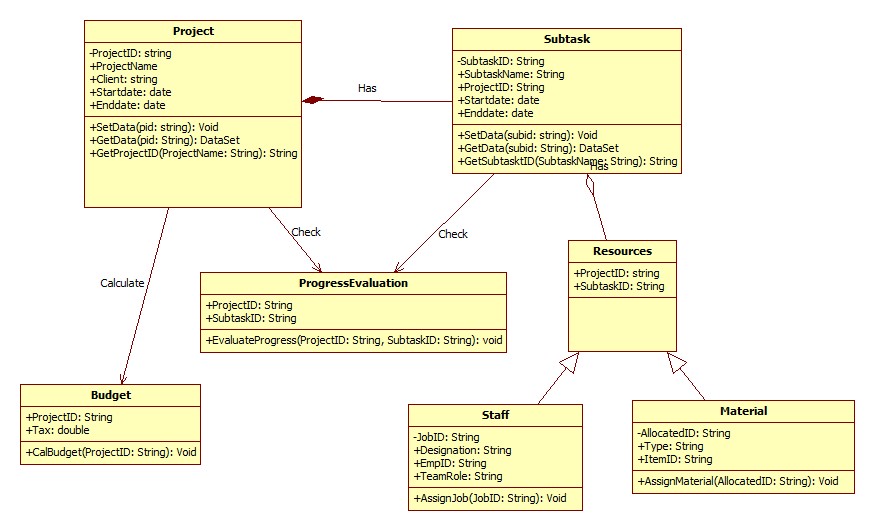


Figure 2.2.2: Project management

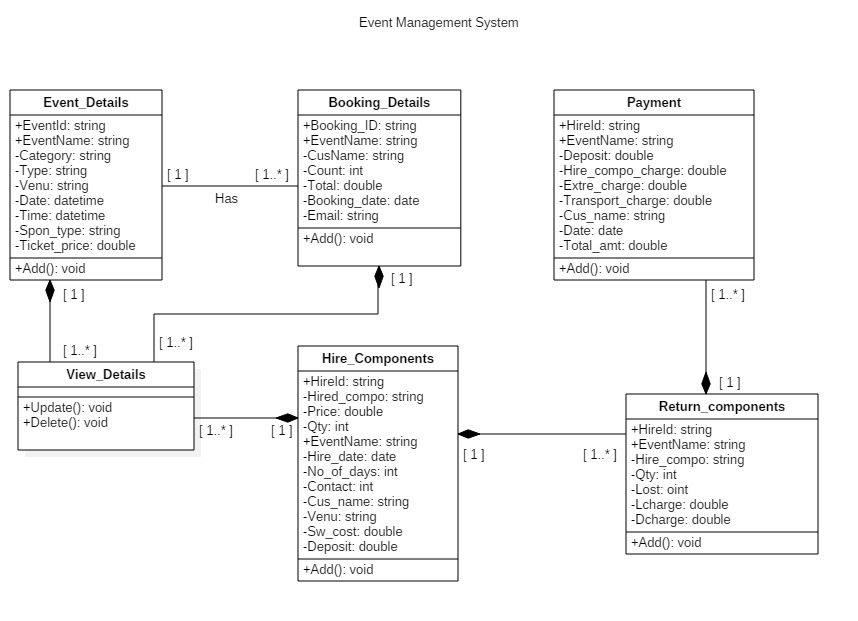


Figure 2.2.3: Event management

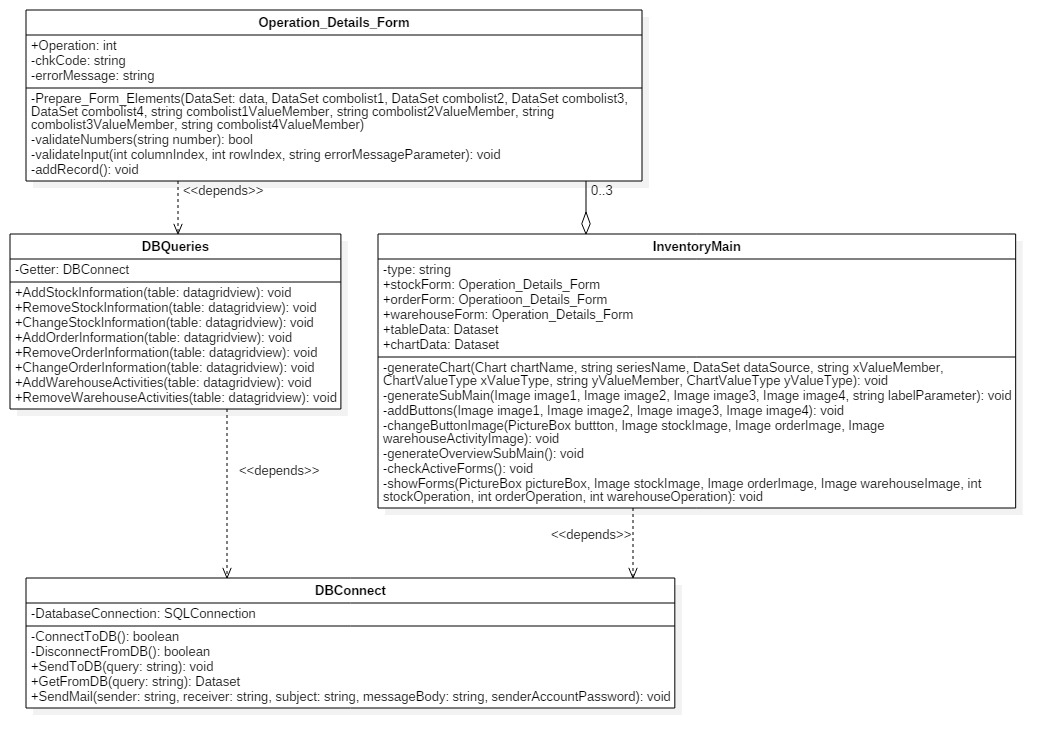


Figure 2.2.4: Inventory management

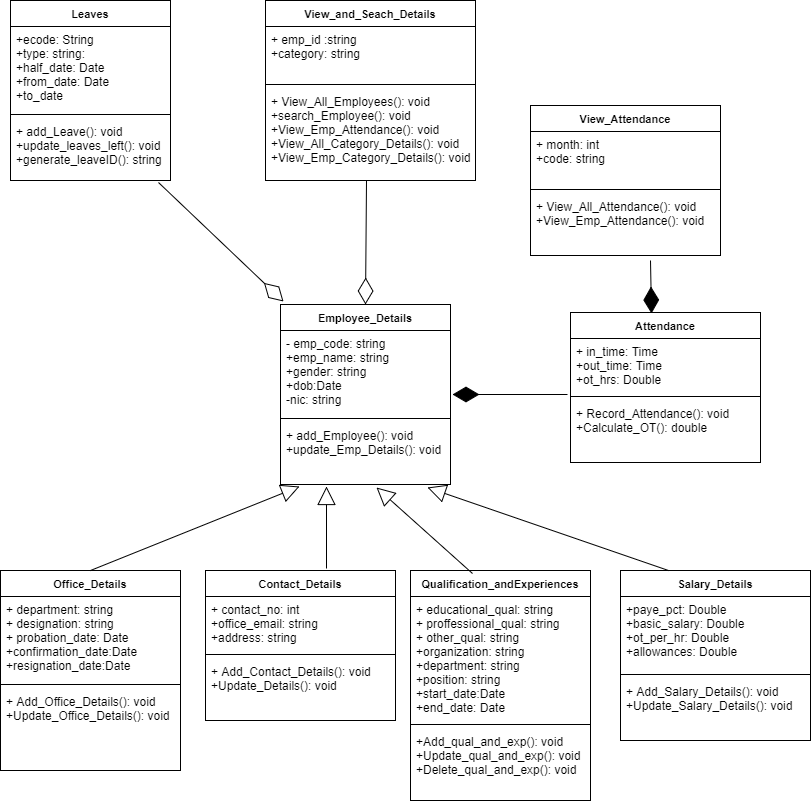


Figure 2.2.5: Employee management

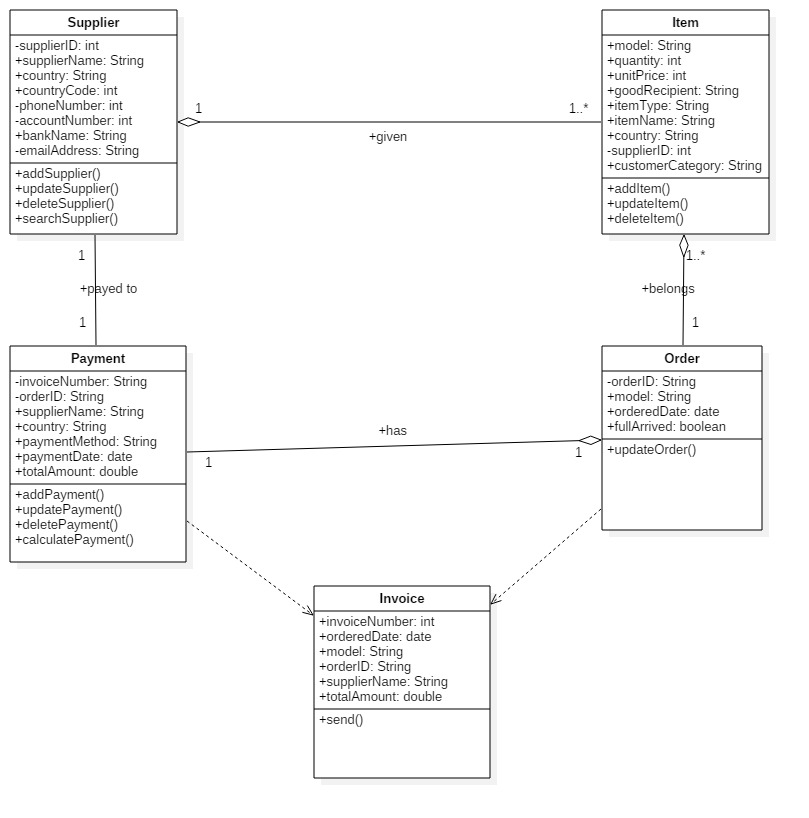


Figure 2.2.6: Supply management

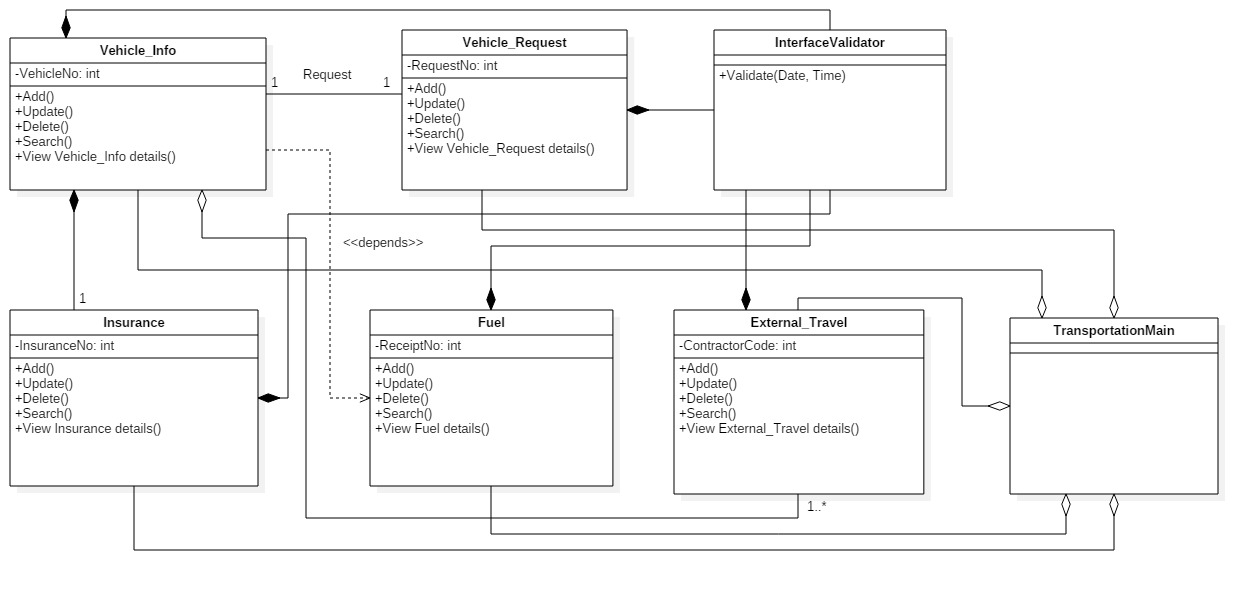


Figure 2.2.7: Transport management

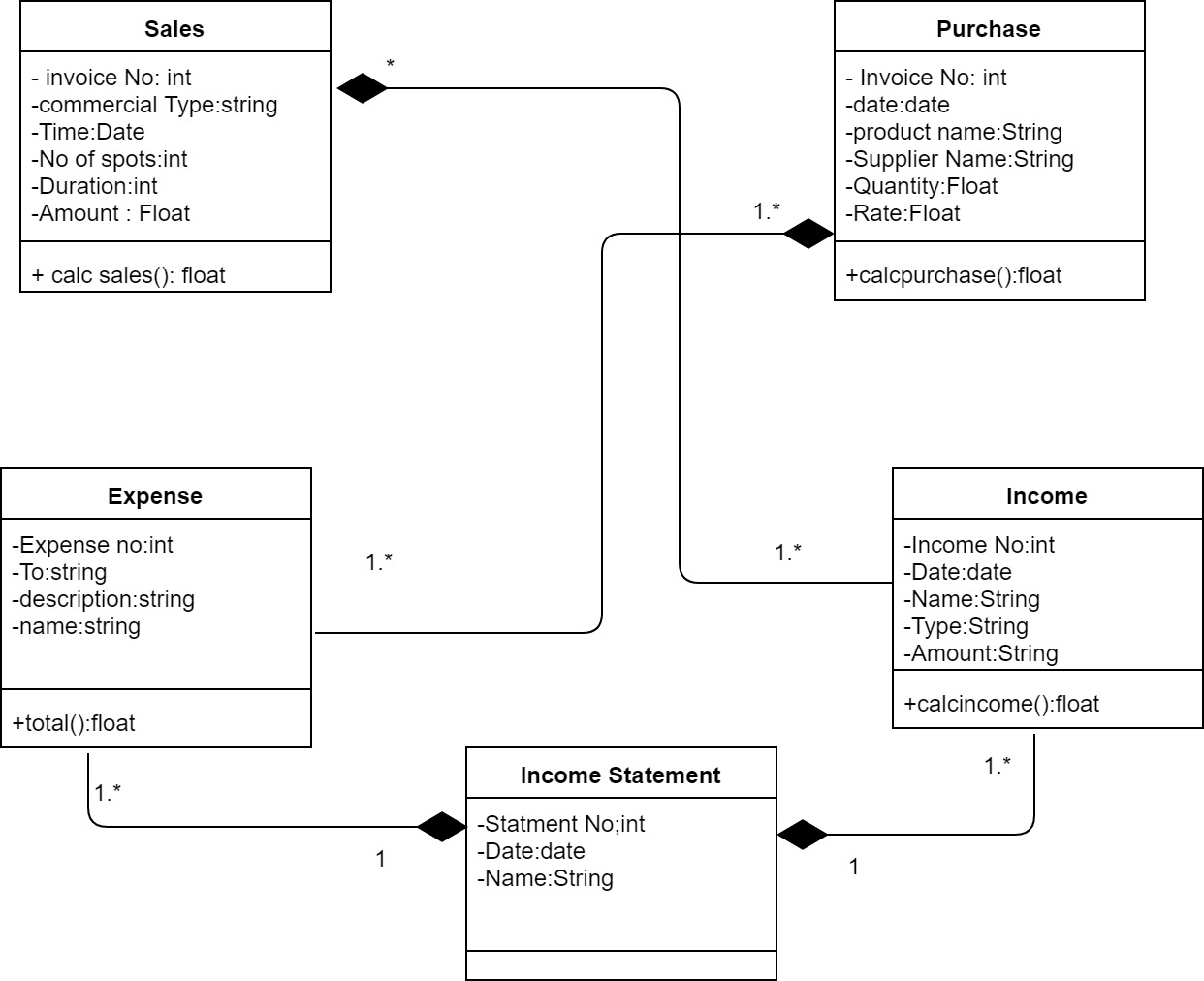


Figure 2.2.8: Finanace management

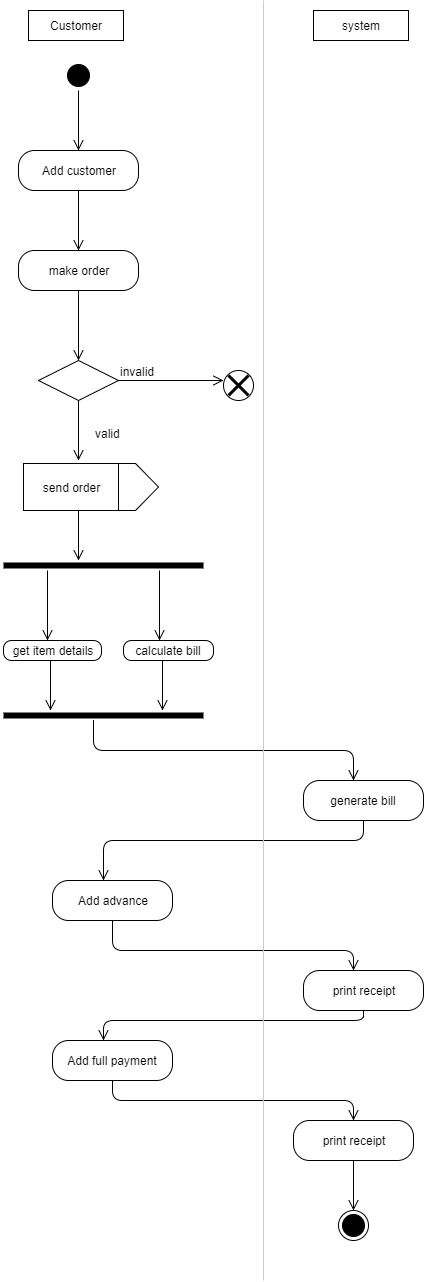


Figure 2.2.9: Customer management

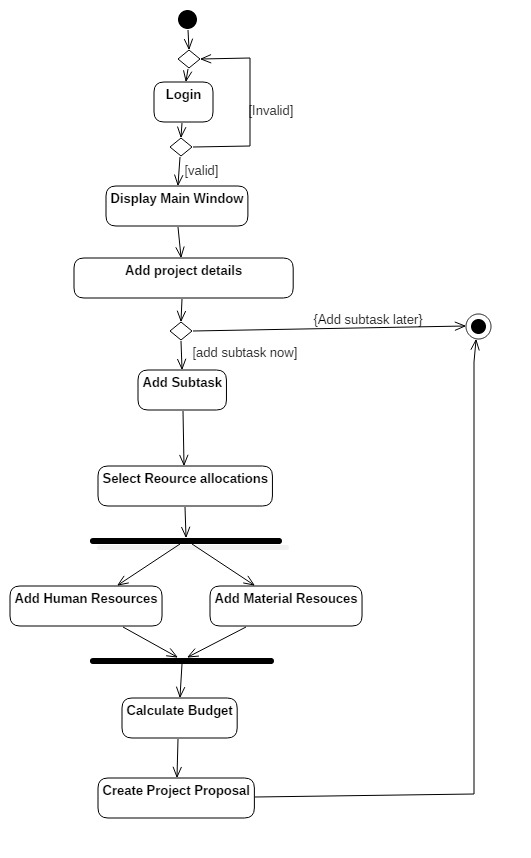


Figure 2.2.10: Project management

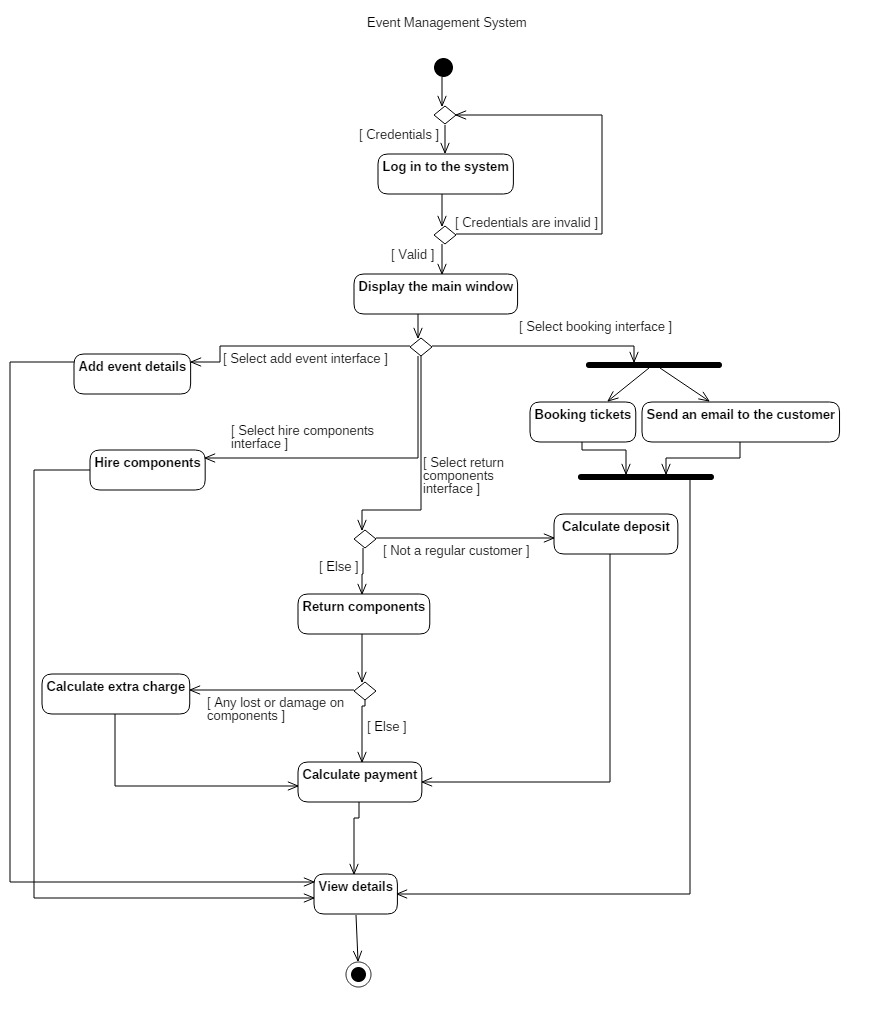


Figure 2.2.11: Event management

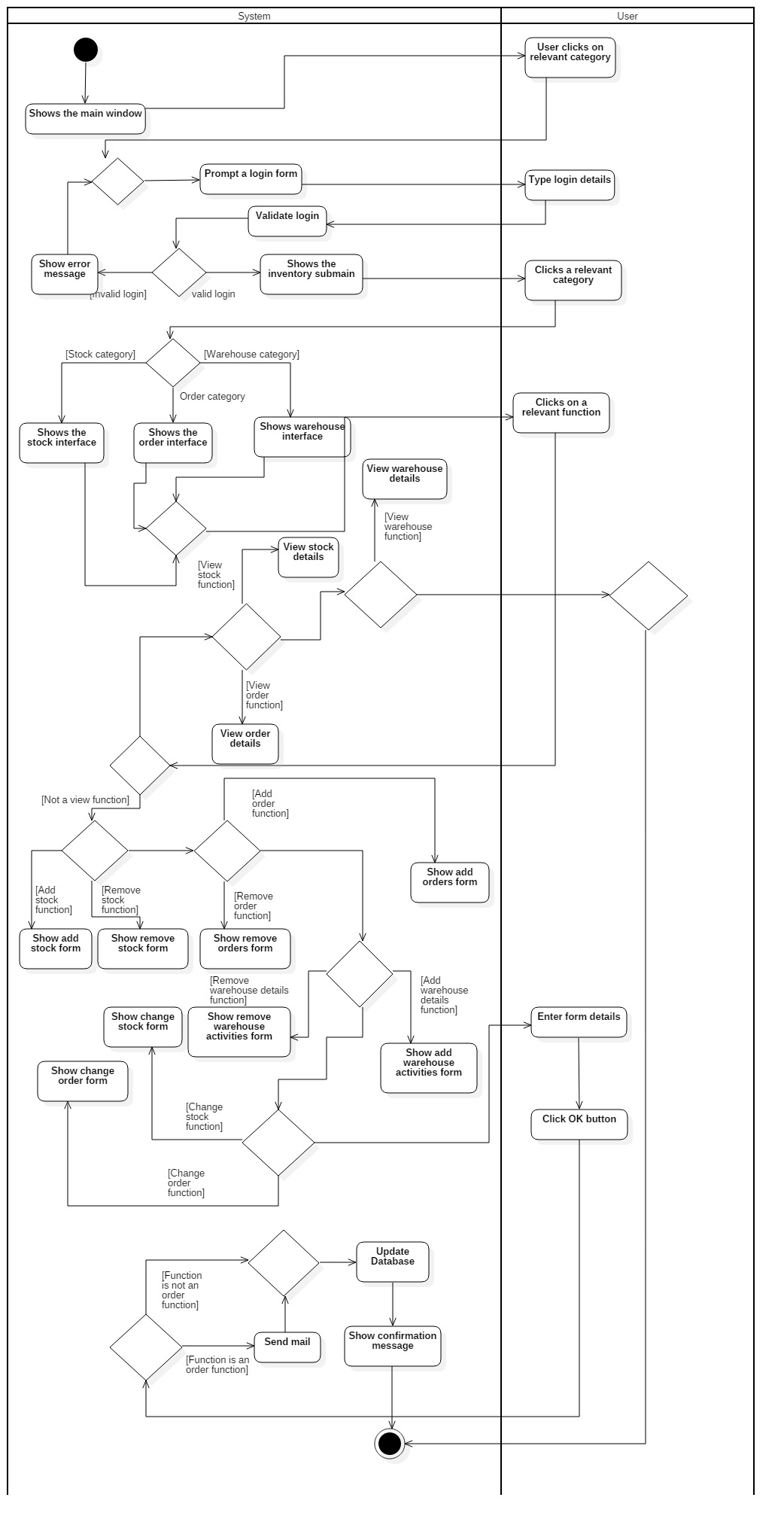


Figure 2.2.12: Inventry management

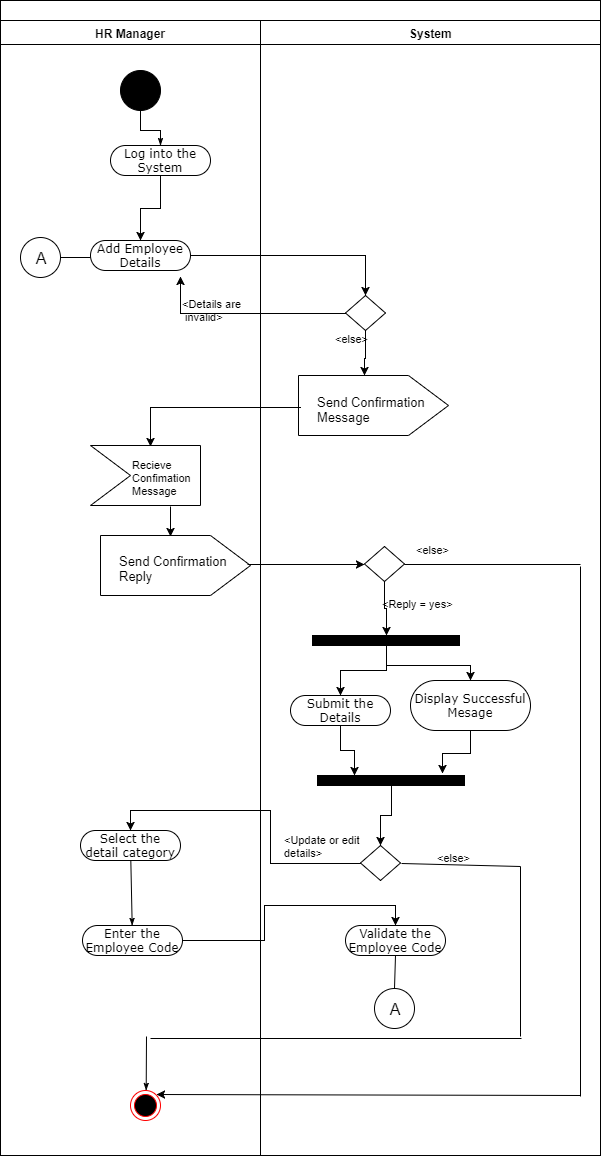


Figure 2.2.13: Employee management

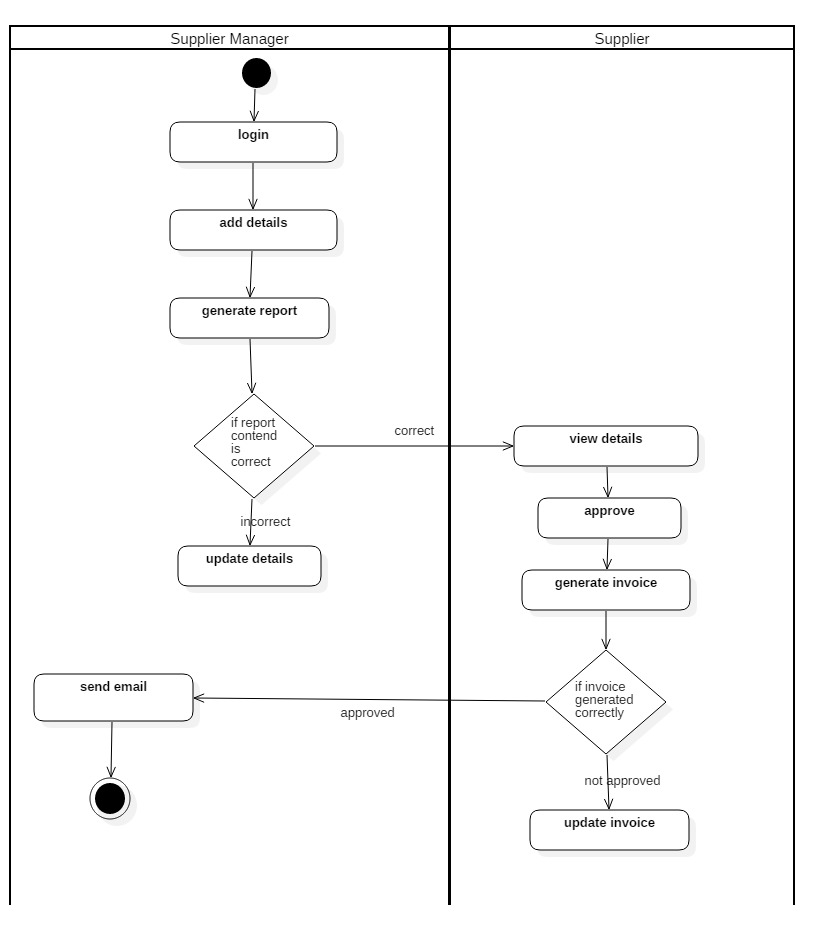


Figure 2.2.14: Supply management

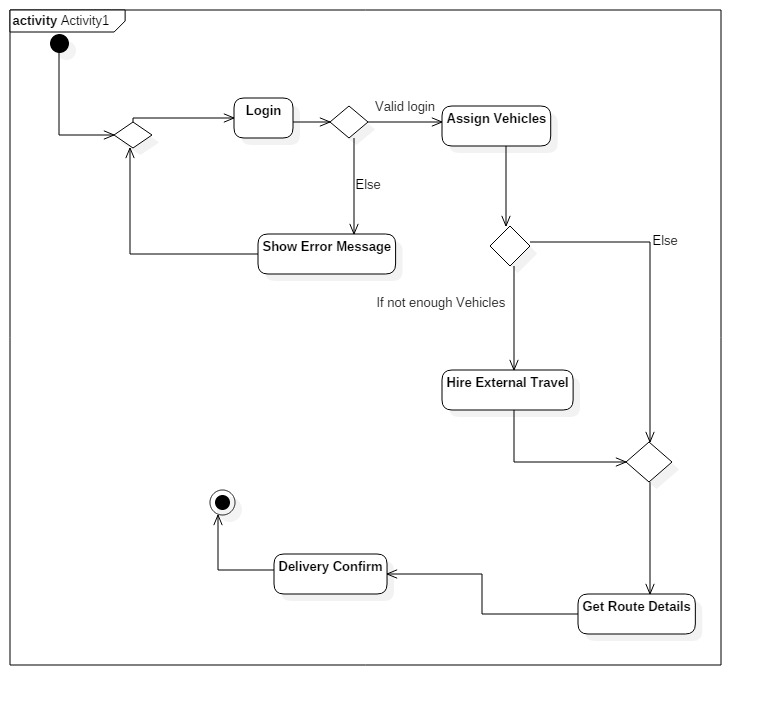


Figure 2.2.15: Transport management

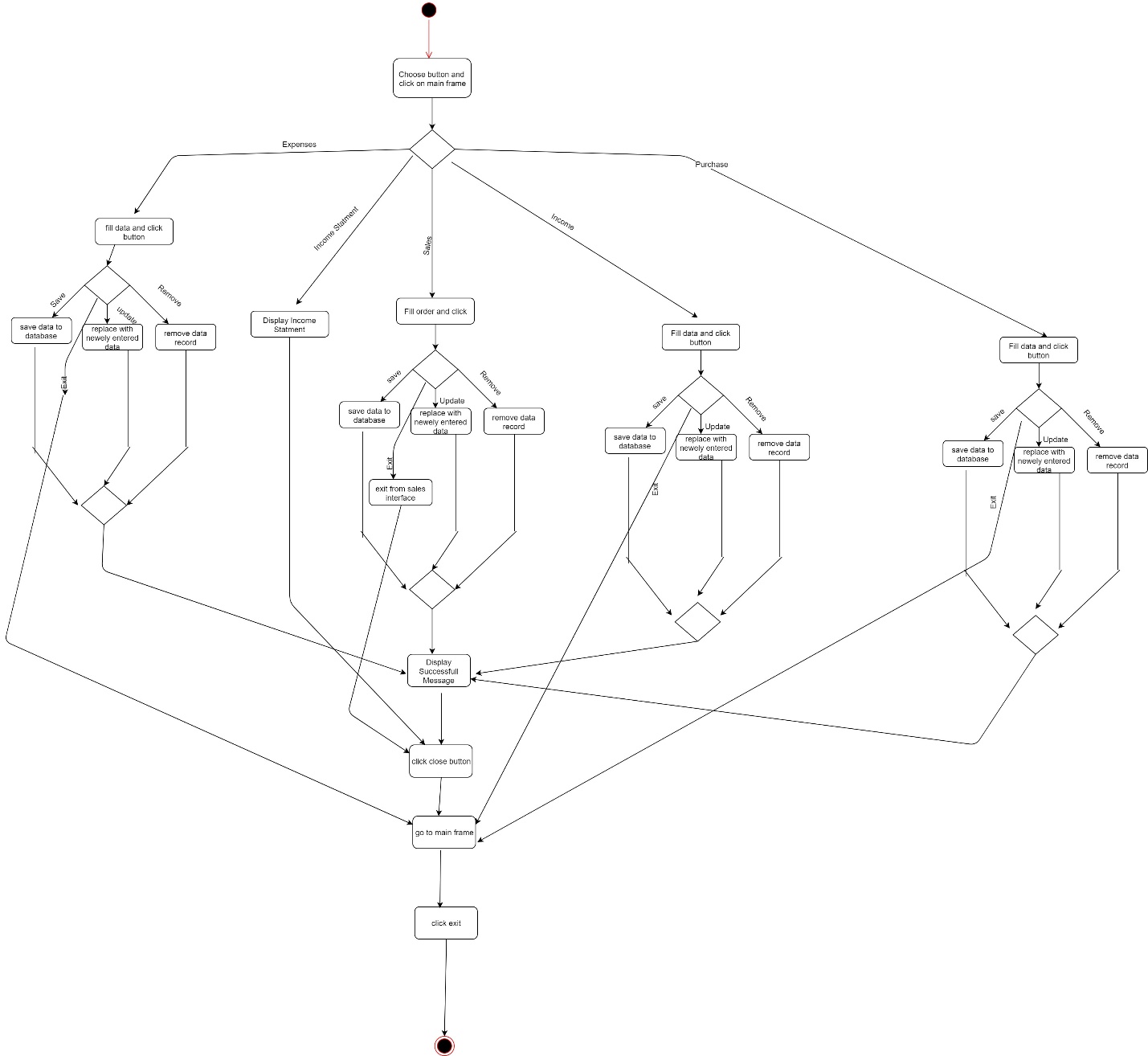


Figure 2.2.16: Finance management

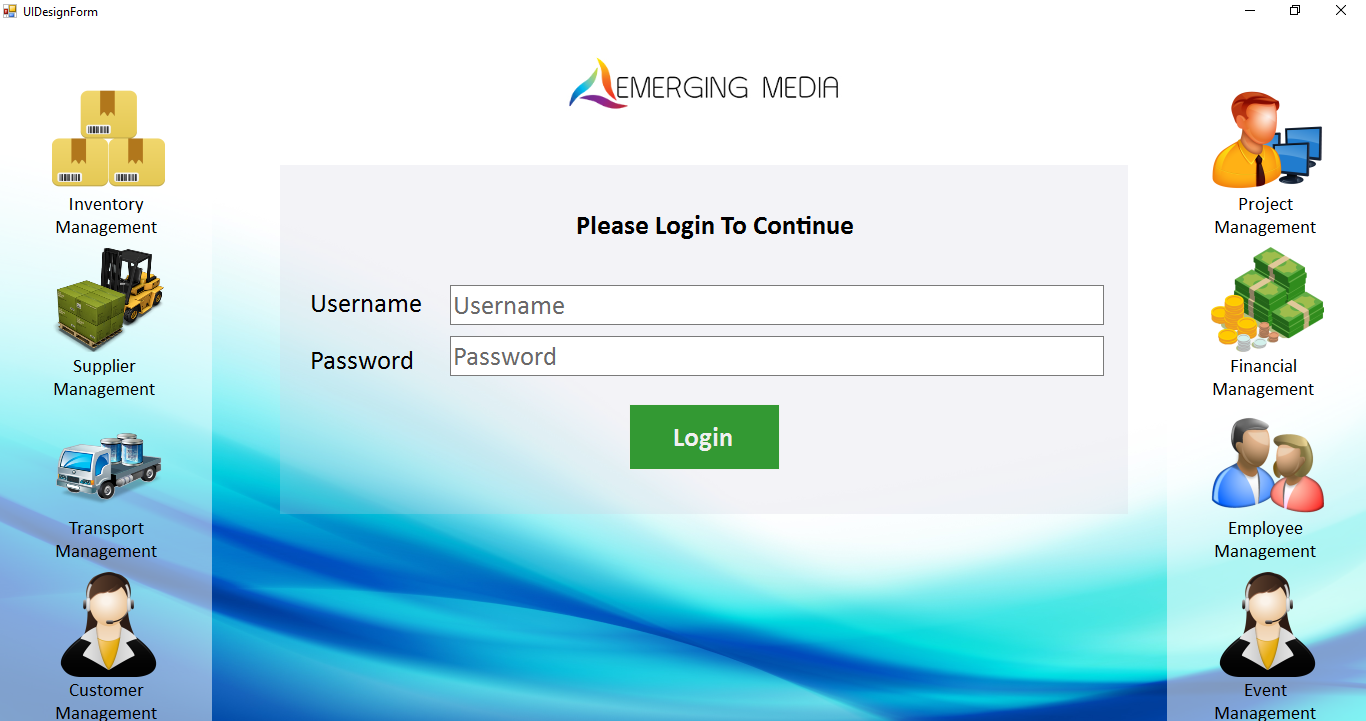
**2.3 Implementation**

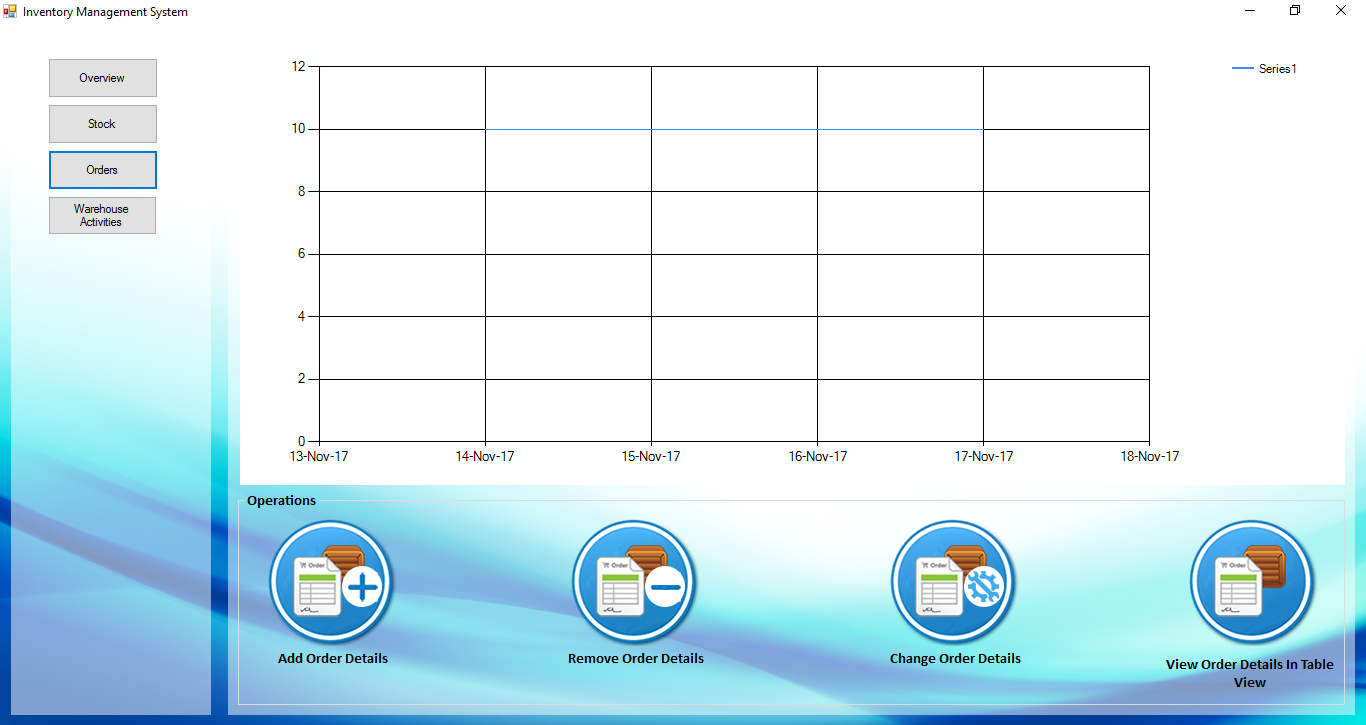
The Implementation of Supply Chain Management System for Emerging Media was planned after conducting several discussions with the company IT department and team members. Once identified the systems main functionalities, the development of each function was distributed among the members with equal weight in modules. Which are Project Management, Event Management, Transport Management, Inventory Management, Supplier management, financial Management Employee Management, Customer Management. After developing each module, they were integrated to one fully functional system. The team had to be hold to several constraints in the process of development of their module.

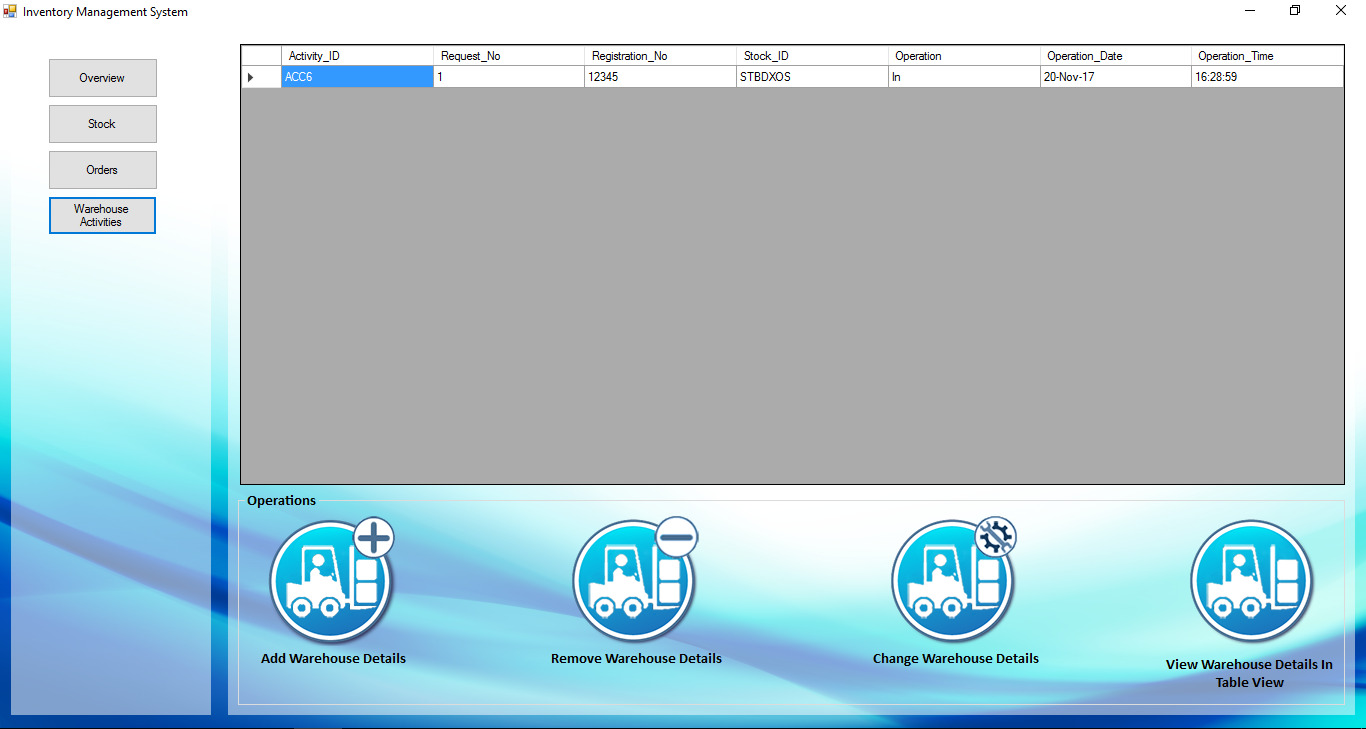
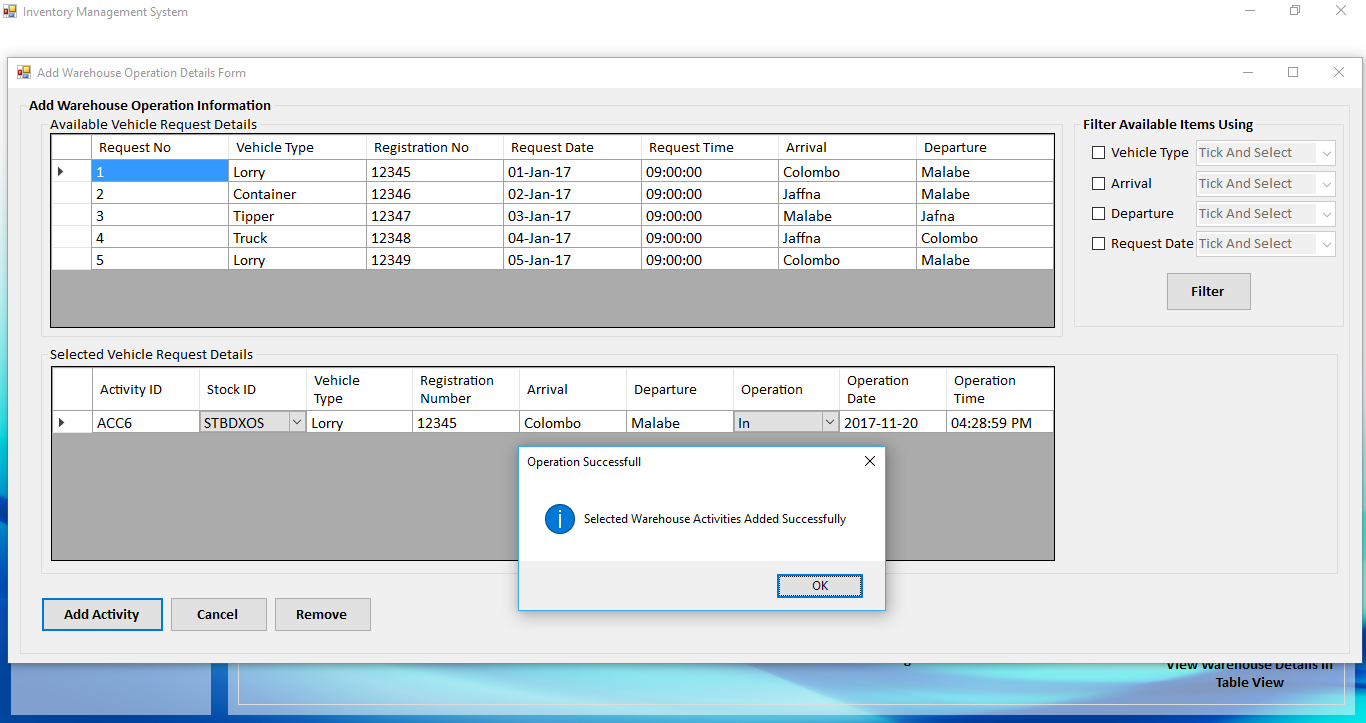
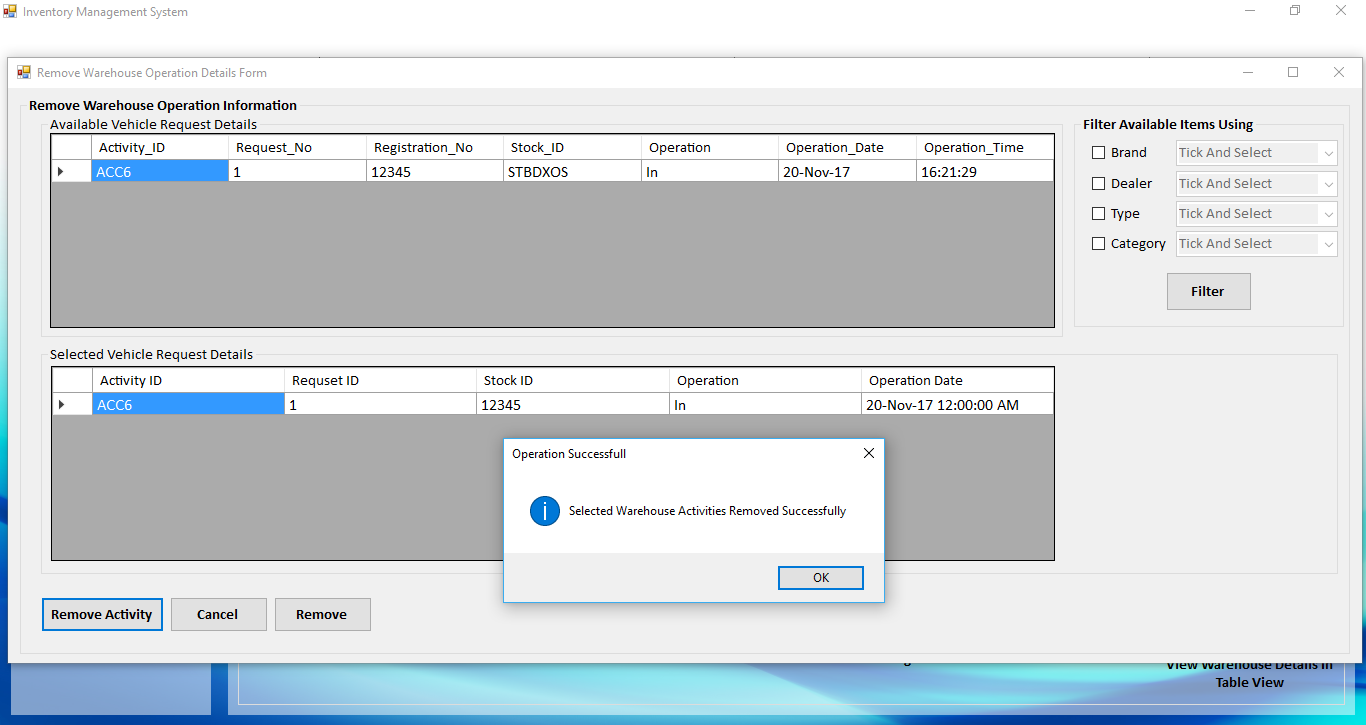
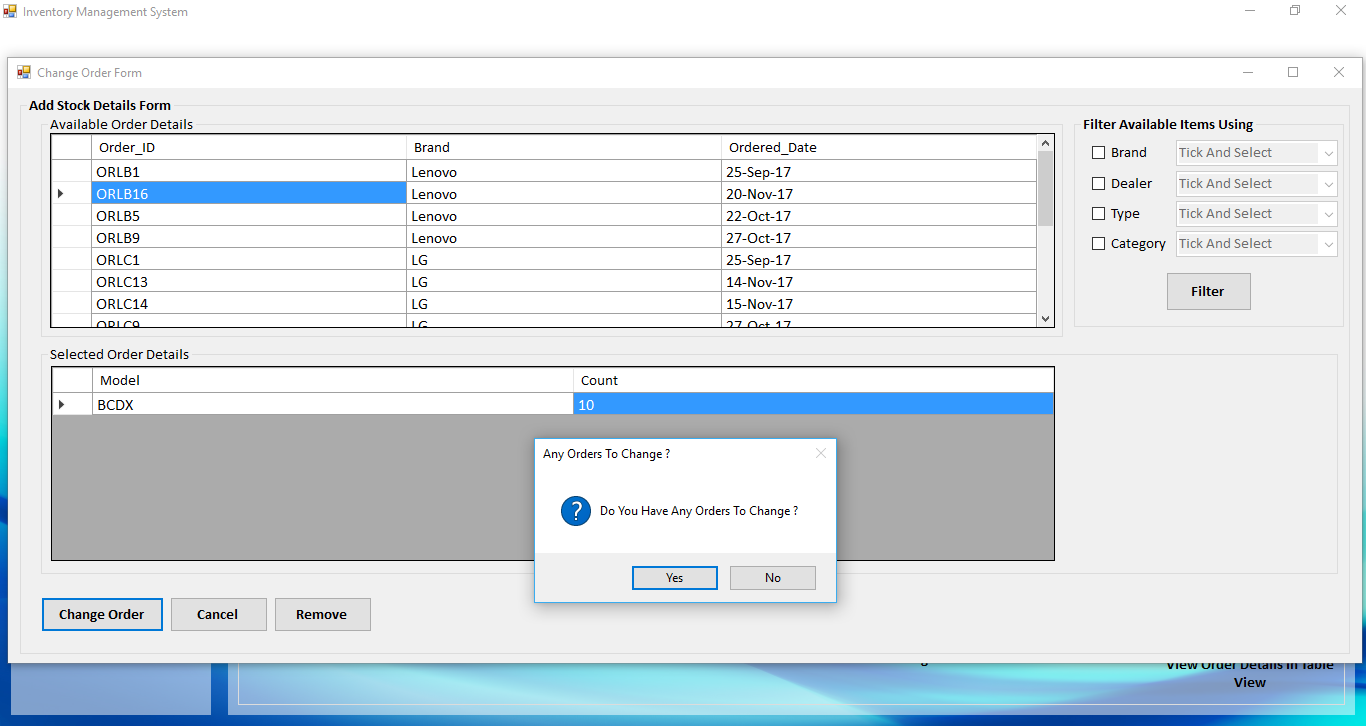
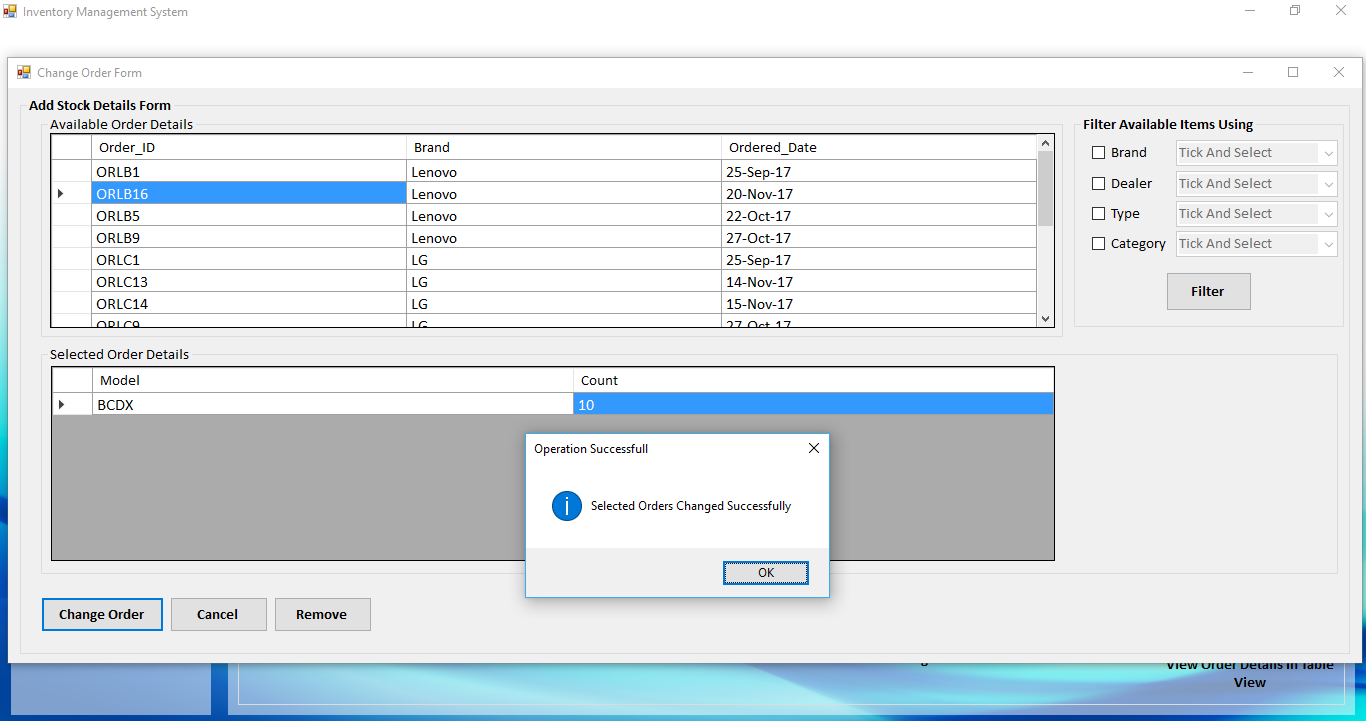
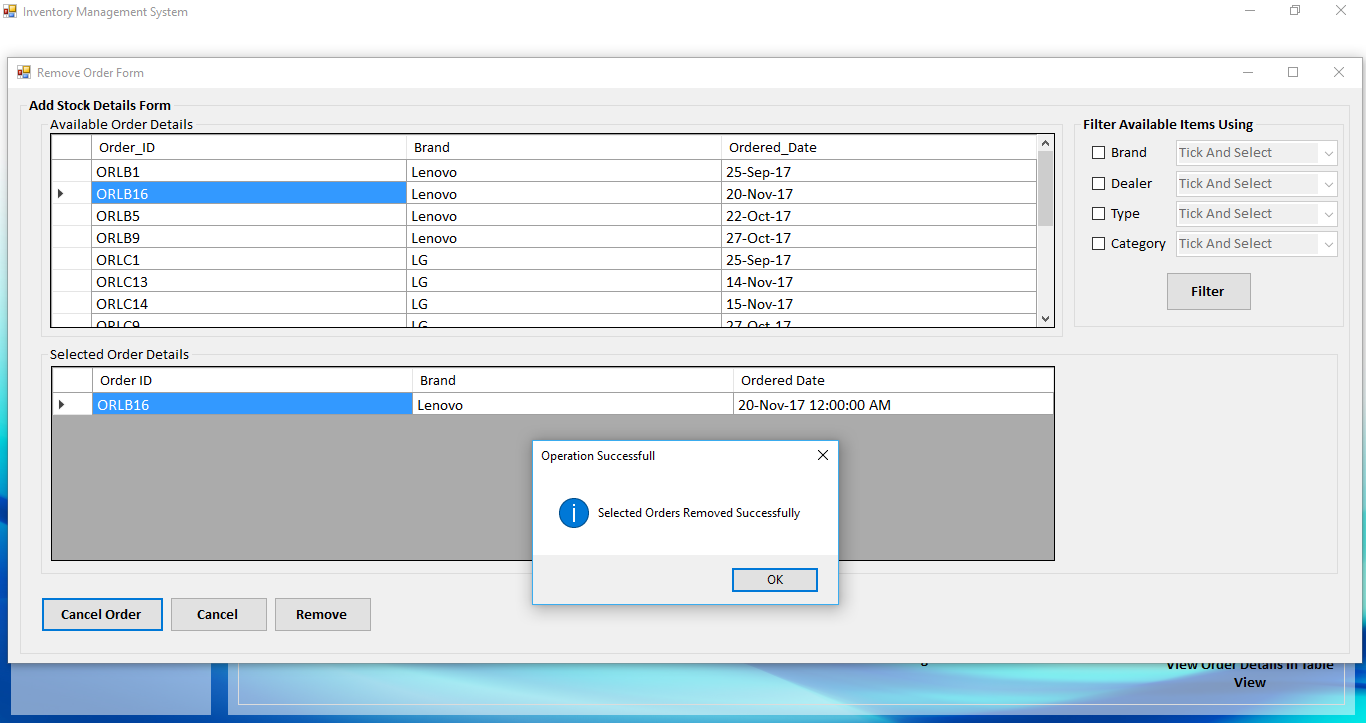
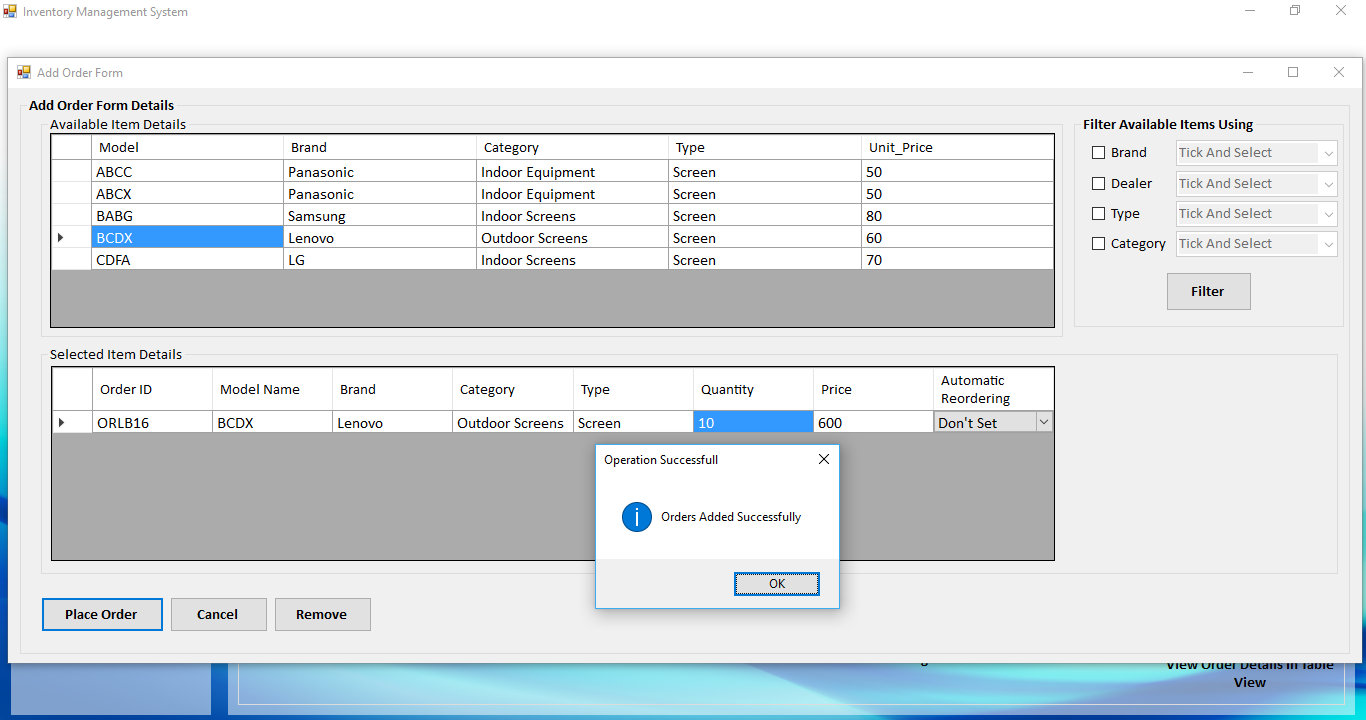
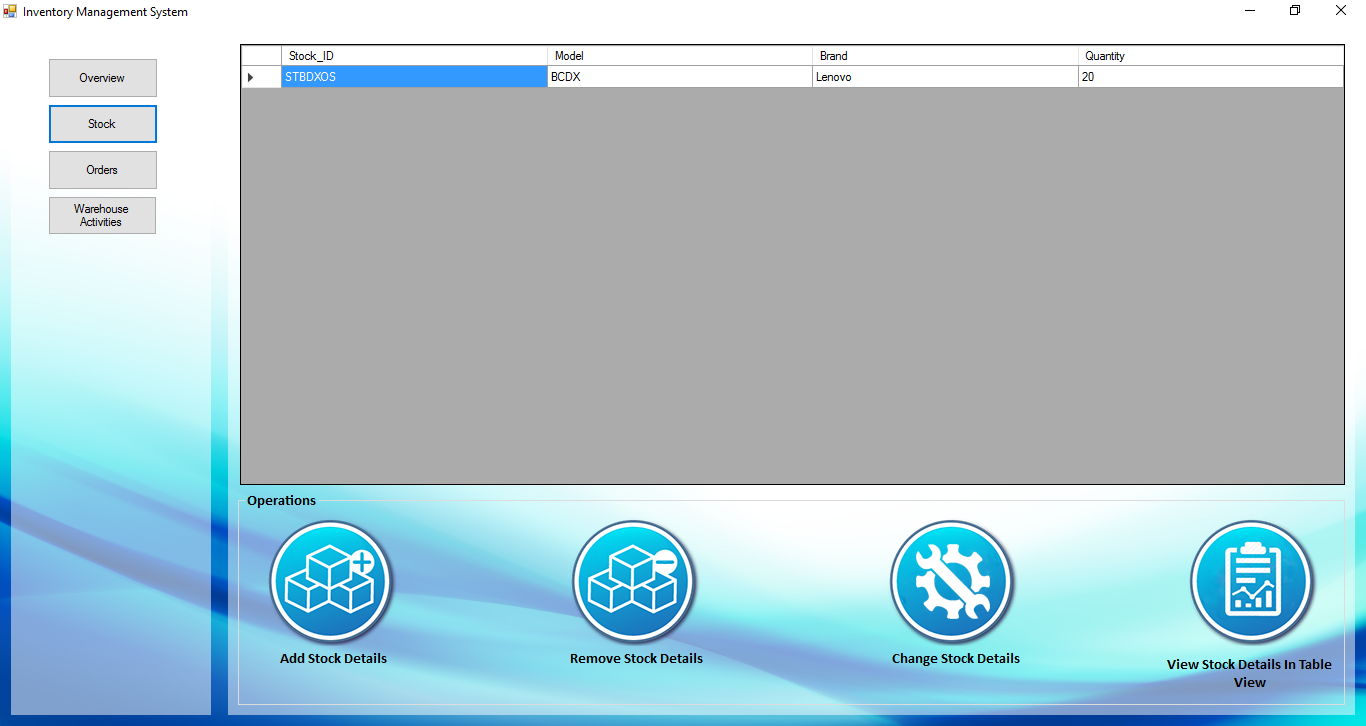
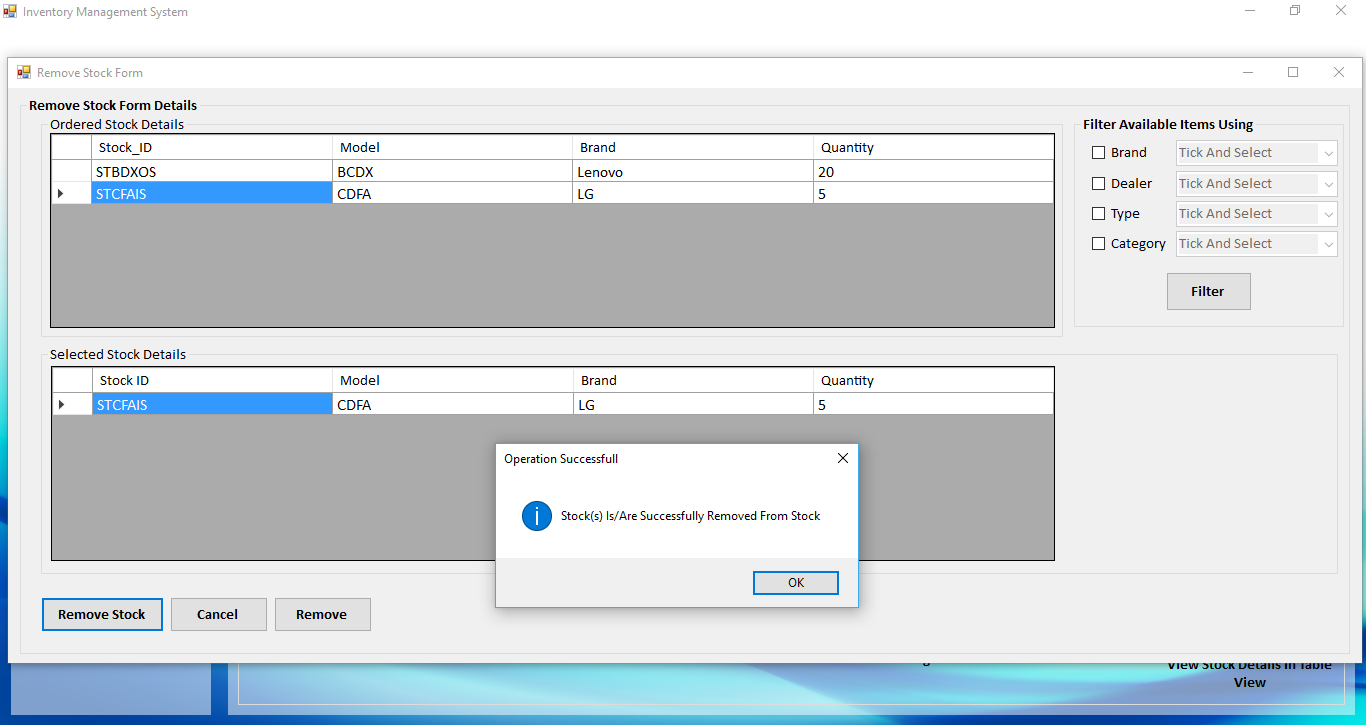
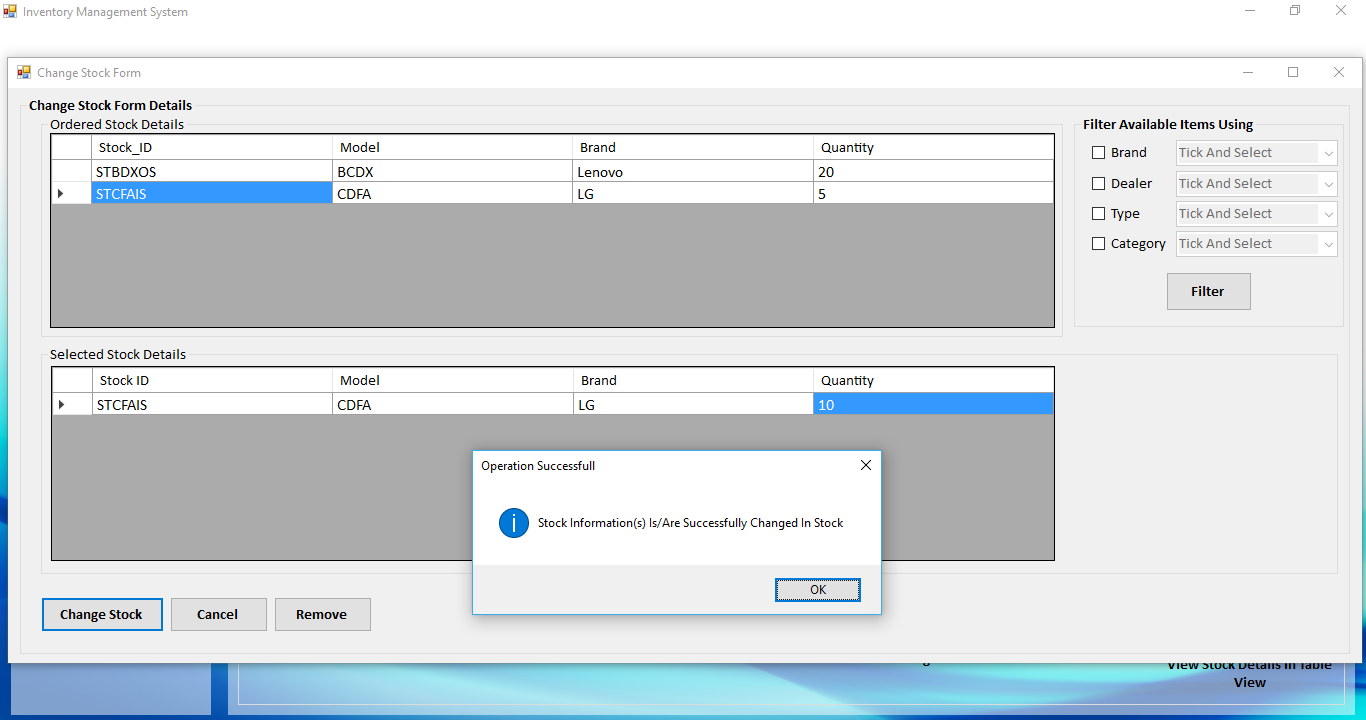
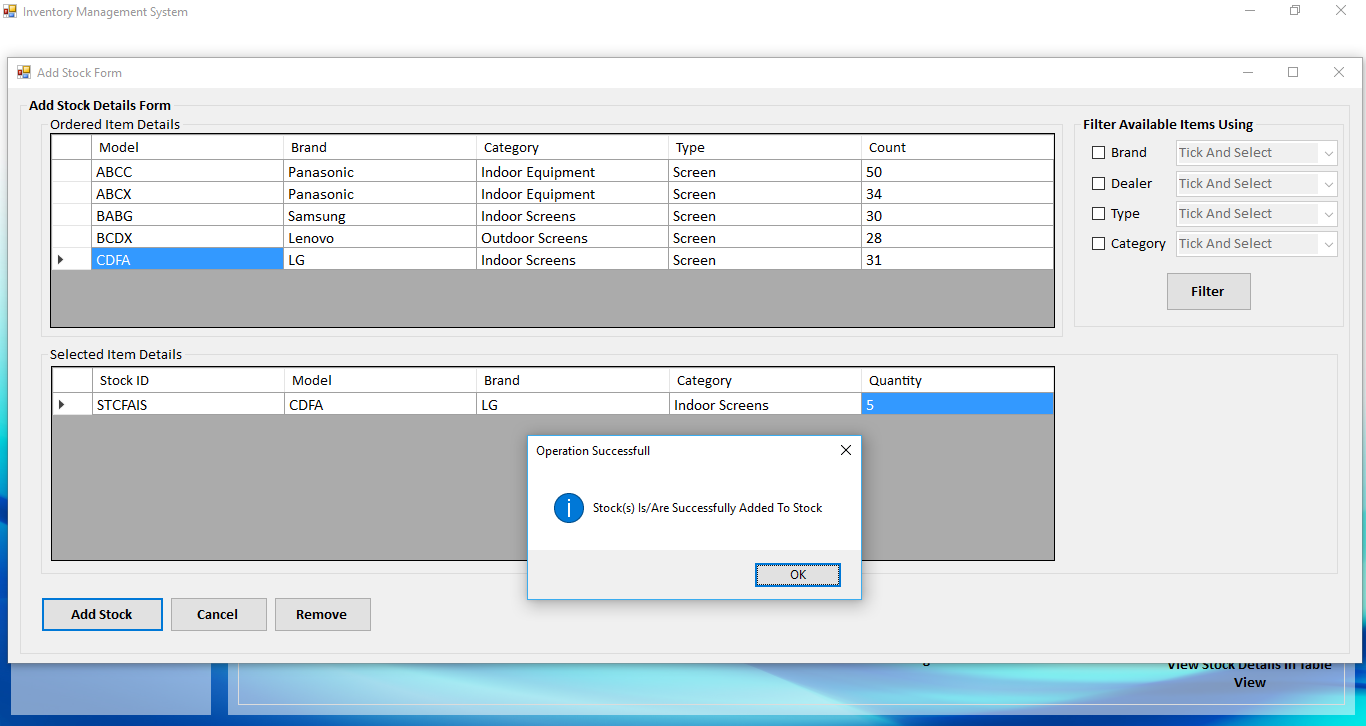
* The system should be a standalone system.
* Required language is c#.net, required database management system is MS SQL, and reporting too should be crystal reports.

Following constraints as mentioned above the functions were developed by each individual. Each member had the freedom to add own exciting functions, with beholding to some common functionalities to ensure the balance between the sub modules of the system.

**2.3.1 Graphics and Interfaces**



**

**

**2.4 Testing**

With the intension of ensuring the reliability and efficiency of the system testing phase did a remarkable job. The testing phase was carried out in two stages

* Developer side testing- to make sure the system was free from implementation and logical errors.
* Client-side testing- to ensure that the system meet the client requirements.

Developer side testing was carried out during the entire period of development to ensure that the system is free from implementation testing and logical errors. Then after the modules were unit tested before the implementation to make sure whether they are up and running without any crashes and according to the necessities of the client.

After that the entire system was integrated and tested for the same conditions as before mentioned.

Client-side testing was done at the client company site with the presence of them to make sure the system meets their necessities. Following test cases will give you an idea how the system was tested to meet its expectations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case**  **ID** | **Test Case Description** | **Test**  **Inputs** | **Test Procedure** | **Expected Results** |
| Login Test01 | Valid Login Test | Gehan123,Gv1234 | Type username and password in relevant field and click login button | Load the relevant module. |
| Login Test02 | Invalid Login Test | Abcd,abcd123 | Type username and password in relevant field and click login button | Display an error message (“Invalid username or password”). |
| Module Button Test01 |  |  |  |  |
| PM Test01 | Project Management Interface: Project interface: input fields  Check(search) | Search key: Project name | Type a project name in the search field | Relevant project names details will be displayed in the datagridview |
| PM Test02 | Project Management Interface: Project interface: input fields  Check(add) | Seylan sirisewana event, web application,  Seylan,Sampath,  11/12/2017,  30/12/2017,  5,0color | Fill in the text fields with correct values and click add | Details get added to the db. and datagrid view and prompt message  “Project added successfully” |
| PM Test03 | Project Management interface: Project interface: input fields check(Update) | Update the relevant fields | select the relevant projectID to be updated and click Update button | the respective values will be updated with the new values in the fields and prompt a message  “Updated Successfully” |
| PM Test04 | Project  Management interface: Project interface: check(Delete) | Delete the relevant Project | Select the relevant ProjectID to be deleted and click Delete button | The respective details related to the ProjectID will be removed from the db. |
| PM Test05 | Project Management interface: Project interface: check (date validation) | select end date behind the start date | Select an end date for the project behind the start date and click Add button | Display error message in front of the datetimepicker of end date |
| PM Test06 | Project management interface: Subtask interface: (date validation) | Select start date to the subtask ahead of the respective projects end date | Select start date ahead the respective projects deadline and click Add | Prompts error message indicating that the selected date is exceeding the projects deadline |
| PM Test07 | Project Management interface: Subtask interface: (date validation) | Select end date to the subtask ahead of the respective projects end date | Select the end date ahead the respective projects deadline and click Add | Prompts error message indicating that the selected date is exceeding the projects deadline |
| PM Test08 | Project Management interface: Subtask interface: (date validation) | select a start date greater than the end date | select start date greater than end date and click Add button | Prompts error message showing that start date is greater than end date |
| PM Test09 | Project Management interface: Calculate Budget interface: (percentage validation) | Enter Percentage greater than 100 or less than 0 | Enter percentage value less than 0 or greater than 100 and click  Calculate | Prompt error message |
| PM Test10 | Project Management interface: Resource allocation interface: Staff interface: (send job status mail) | Seylan Sirisewana event, s001, developer,100,  Group leader,10000 | Fill in the fields with respective values and click Finish, click ok in the send email prompt | prompt successful message |
| PM Test11 | Project Management interface:Subtask interface: (Update click) | Update the relevant fields | Select the relevant and click the Update button | The details of the Subtask will be updated, details related project (project budget, resources, progress) will be updated |
| PM Test12 | Project Management interface: Subtask interface: (Delete click) | Delete the relevant SubtaskID | Select the relevant SubtaskID and click Delete button | Details related to the entire subtask will be deleted and the relevant details of the project (project budget, resources, progress)  will be updated |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case description | Test Inputs | Test Procedure | Expected result |
| CM test 1 | Add customer interface:  Valid email address | Maliban.com  maliban@gmail  maliban | Click the next text field | Display an error message when go to next text field |
| CM test 2 | Add customer interface:  Valid contact number | 011236547  01qwx124447  0112365478956  wedsvtyhgf | Enter the number and click on Add customer button. | Display an error message. Cannot save details. |
| CM test 3 | Add customer interface:  Valid Input fields check | Update the relevant fields | Click on the Add customer button. | Display a successful message |
| CM test 4 | Add customer interface:  Invalid Input field check | - | Click on Add customer button | Display an error message cannot save details |
| CM test 5 | Update customer interface:  Select empty data grid | - | Update the relevant record by clicking on the record. | Display an error message |
| CM test 6 | Update customer interface:  Search Button Check blank text fields | Relevant input for search text box. | Fill the text box and click on Search button | Display relevant error message |
| CM test 7 | Delete customer interface:  Select data from data grid | - | Delete the relevant record by clicking on the button. | Display information message |
| CM test 8 | Make order interface:  blank text fields | - | Click on the relevant main function button. | Display an error message |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Input** | **Test Procedure** | **Expected Results** |
| HRTest1 | Personal Details Interface: Input fields check for correct details | ‘E0001’,  ‘A.B.Perera’,  ‘Allison Ben  Perera’,  Female,  26-09-1996,  123456789V,  ‘Sinhala’,  ‘Nations Trust  Bank’,  ‘Malabe’, 11111 | Enter the inputs and click submit button | 1. Display submit confirmation message.  2. Display Successful message and send Employee personal data to the database after selecting ‘yes.’ |
| HRTest2 | Personal Details Interface: Input fields check for invalid and empty details | ‘E0001’,  ‘A.B.Perera’,  ‘Allison Ben  Perera’,  Female,  26-09-1996,  12345,  ‘Sinhala’,’ ’,  ‘Malabe’,  ‘abcde’ | Enter the inputs and click submit button | Display error message due to validation failure. |
| HRTest3 | Personal Details  Interface: Edit details with valid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Allison Ben  Perera’,  Male,  26-11-1996,  123456789V,  ‘Sinhala’,  ‘Sampath Bank’,  ‘Malabe’, 11111 | Enter the inputs and click Save button | 1. Display update confirmation message.  2. Display Successful message and update the details to the database after selecting ‘yes’. |
| HRTest4 | Personal Details  Interface: Edit details with invalid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Allison Ben  Perera’,’ ’,  26-09-1996,  56789V,  ‘Sinhala’,  ‘Nations Trust  Bank’,  ‘657’, 11111 | Make changes and click Save button | Display error message due to validation failure. |
| HRTest5 | Official Details Interface: Input fields check for correct details | ‘E0001’,  ‘A.B.Perera’,  ‘Digital and IT’,  ‘developer’,  ‘Yes’,  ’07-18-2017’,  ’12-18-2017’,  ‘ ’ | Enter the inputs and click submit button | 1. Display submit confirmation message.  2. Display Successful message and send data to the database after selecting ‘yes.’ |
| HRTest6 | Official Details Interface: Input fields check for invalid and empty details | ‘E0002’,  ‘A.B.Perera’,  ‘Digital and IT’,  ‘developer’,  ‘No’,  ’07-18-2017’,  ’12-18-2017’,  ‘ ’ | Enter the inputs and click submit button | Display error message due to validation failure. |
| HRTest7 | Official Details Interface: Edit details with valid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Digital and IT’,  ‘programmer’,  ‘Yes’,  ’07-18-2017’,  ’12-18-2017’,  ‘ ’ | Enter the inputs and click Save button | 1. Display update confirmation message.  2. Display Successful message and update the details to the database after selecting ‘yes’. |
| HRTest8 | Official Details Interface: Edit details with invalid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Digital and IT’,  ‘developer’,  ‘No’,  ’07-18-2017’,  ’12-18-2017’,  ‘ ’ | Enter the inputs and click Save button | Display error message due to validation failure. |
| HRTest9 | Contact Details Interface: Input fields check for correct details | ‘E0001’,  ‘A.B.Perera’,  ‘Colombo’,  ‘Colombo’,  0123456789  0771236589  ‘ben@gmail.com’  ‘ben@my.abc.lk’  ‘D.E.Fernando’  ‘Mother’,  0112233665 | Enter the inputs and click submit button | 1. Display submit confirmation message.  2. Display Successful message and send data to the database after selecting ‘yes.’ |
| HRTest10 | Contact Details Interface: Input fields check for invalid and empty details | ‘E0002’,  ‘A.B.Perera’,  ‘Colombo’,  ‘’,  0123456789  ‘abcde’  ‘ben@gmail.com’  ‘ben.my@lk  ‘D.E.Fernando’  ‘Mother’,  0112233665 | Enter the inputs and click submit button | Display error message due to validation failure. |
| HRTest11 | Contact Details Interface: Edit details with valid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Negombo’,  ‘Colombo’,  0123456789  0771236589  ‘ben@gmail.com’  ‘ben@my.abc.lk’  ‘D.E.Fernando’  ‘Father’,  0112233665 | Enter the inputs and click Save button | 1. Display update confirmation message.  2. Display Successful message and update the details to the database after selecting ‘yes’. |
| HRTest12 | Contact Details Interface: Edit details with invalid updating details | ‘E0001’,  ‘A.B.Perera’,  ‘Colombo’,  ‘’,  0123456789  ‘abcde’  ‘ben@gmail.com’  ‘ben.my@lk  ‘D.E.Fernando’  ‘Mother’,  0112233665 | Enter the inputs and click Save button | Display error message due to validation failure. |
| HRTest13 | Experiences and Qualifications: Add Qualification | ‘Pass O/L with 9A’s’ | Enter the inputs and click Add button | Add details to the Qualifications  grid |
| HRTest14 | Experiences and Qualifications: Add empty field Qualification | - | Enter the inputs and click Add button | Display error message due to validation failure. |
| HRTest15 | Experiences and Qualifications: Add valid experience details | ‘Orange’, ‘IT’,  ’developer’,  ’01-01-2017’,  ’01-06-2017’ | Enter the inputs and click Add button | Add details to the Experiences  grid |
| HRTest16 | Experiences and Qualifications: Add invalid or blank experience details | - | Enter the inputs and click Add button | Display error message due to validation failure. |
| HRTest17 | Experiences and Qualifications: Submit valid qualifications | - | Enter the inputs and click Submit button | 1. Display submit confirmation message.  2. Display Successful message and send data to the database after selecting ‘yes.’ |
| HRTest18 | Experiences and Qualifications: Submit valid experience details | - | Enter the inputs and click Submit button | 1. Display submit confirmation message.  2. Display Successful message and send data to the database after selecting ‘yes.’ |
| HRTest19 | Salary Details Interface: Input fields check for correct details | ‘E0001’,  Basic Sal =30000  OT = 100 | Enter inputs and click Add button | Add details to the grid. |
| HRTest20 | Salary Details Interface: Input fields check for invalid and empty details | ‘E0001’,  Basic Sal =  OT = ‘abc’ | Enter inputs and click Add button | Display error message due to validation failure. |
| HRTest21 | Salary Details  Interface: Edit details with valid updating details | ‘E0001’,  Basic Sal =35000  OT = 100  Allowances=1000 | 1.Enter inputs and click Add button  2. Click Save button | 1. Add details to the grid.  2. Update the details in the database after clicking “Yes”  in the confirmation message |
| HRTest22 | Salary Details  Interface: Edit details with invalid updating details | ‘E0001’,  Basic Sal =30000  OT = ‘abc’ | Enter inputs and click Add button | Display error message due to validation failure. |
| HRTest23 | Salary Details  Interface: Submit details | - | Click submit Button | 1. Display submit confirmation message.  2. Display Successful message and send data to the database after selecting ‘yes.’ |
| HRTest24 | Get Attendance Interface: Add attendance with valid details. | Code=’E0001’  In:  01-01-2017 08:00  Out:  01-01-2017 08:00 | Enter inputs and click Add button | Add details to the grid. |
| HRTest25 | Get Attendance Interface: Submit attendance | - | Click Submit button | Add details in the grid to the database |
| HRTest26 | Apply Leave Interface: Enter valid Leave details | Code=’E0001’  leave type=casual  date=11-18-2017 | Enter the details and click Add Leave button | 1. Update the “leaves left” grid  2. Add leave details to the “leaves taken” grid |
| HRTest27 | Apply Leave Interface: Submit the |  |  | 1. Display submit confirmation message.  2. Display Successful message and send leaves taken data to the database after selecting ‘yes.’  3. update the leaves left table in the database |
| HRTest28 | View Employee Interface: Search Employee | Code=’E001’  or  Name=  ‘A.B.Perera’ | Enter the details and click Search button | Select the row containing the valid name or the employee code |
| HRTest29 | View Employee Interface: Delete Employee | Code=’E001’  or  Name=  ‘A.B.Perera’ | Enter the details and click Delete button | 1.Display confirmation message  2. Delete every record related to the employee after selecting ‘yes’ button  3. Display successful message |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case ID** | **Test case description** | **Test Inputs** | **Test Procedure** | **Expected result** |
| EM test 1 | Add event interface:  Check valid input fields | Mo-EI01  Minion  Indoor  Majestic city  16/12/2017  10.00.00 AM  Full coverage  1000 | Enter the relevant inputs and click Submit Button | Display a successful message and enter the event details to the database |
| EM test 2 | Add event interface  Check valid input fields | Mo-EI01  Minion  Indoor  16/12/2017  10.00.00 AM  Full coverage  1000 | Enter the relevant inputs and click the submit button | Display an error message due to validation |
| EM test 3 | Booking event interface:  Check customer name field | Kamal15 | Typing letters and digits on customer name field | Digits are not displayed on customer name field. |
| EM test 4 | Booking event interface:  Check valid email address | event.manage.gmail.com | Click the next text field | Display an error icon. |
| EM test 5 | Booking event interface:  Check valid email address (Submit) | event.manage.gmail.com | Click the submit button | Cannot save details and display an error message. |
| EM test 6 | Hire component interface:  Check contact number validation | 01127654345682 | Typing more than 10 digits | Display an error message cannot exceed 10 digits. |
| EM test 7 | Hire component interface:  Check contact number validation | 011326ehjs | Typing letters on contact number field | Letters are not display on the on field |
| EM test 8 | Return component interface:  Check event name validation | Minion432 | Typing letters or digits on the combox field (event name) | Display an error icon and display an message “Please select the event name” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Inputs** | **Test Procedures** | **Expected Results** |
| Login Test 01 | Valid login test | Devi123 | Enter the credentials and click on login | Display the loading interface |
| Login Test 02 | Invalid login test | Dev123 | Enter the credentials and click on login | Display an error message |
| Supplier Test 01 | Supplier details interface: Text fields check | Panasonic  China  86  130456781123  [panasonic@gmail.com](mailto:panasonic@gmail.com)  Shin  12246578 | Enter the details and click on the save button | Display relevant message and load the table accordingly |
| Supplier Test 02 | Supplier details interface: Text fields check | Panasonic1@gmail.com | Update the relevant details and click on update button | Display relevant message and load the table accordingly |
| Supplier Test 03 | Supplier details interface: Text fields check | \_ | Delete the relevant record by clicking on the record | Display relevant message load the table accordingly |
| Supplier Test 04 | Supplier details interface: Text fields check | pan | Search the relevant records by clicking on the search button | Display relevant |
| Supplier Test 05 | Supplier details interface: Text field check | \_ | View all the all records by clicking on the view all button | Display all the details |
| Item Test 01 | Item details interface: Text field check | ABCC  Panasonic  Indoor Screen  Screen  Indoor Advertising  Panasonic  China  5  50000  Singhegiri | Enter the details and click on the save button | Display relevant message and load the table accordingly |
| Item Test 02 | Item details interface: Text field check | Outdoor Screen  Outdoor Advertising | Update the relevant details and click on update button | Display relevant message and load the table accordingly |
| Item Test 03 | Item details interface: Text field check | \_ | Delete the relevant record by clicking on the record | Display relevant message load the table accordingly |
| Item Test 04 | Item details interface: Text field check | \_ | View all the all records by clicking on the view all button | Display all the details |
| Payment Test 01 | Payment details interface: Text field check | A23S  E34  Panasonic  China  Credit Card  Monday, November 3, 2017  150000 | Enter the details and click on the save button | Display relevant message and load the table accordingly |
| Payment Test 02 | Payment details interface: Text field check | Wednesday, November 5, 2017 | Update the relevant details and click on update button | Display relevant message and load the table accordingly |
| Payment Test 03 | Payment details interface: Text field check | \_ | Delete the relevant record by clicking on the record | Display relevant message load the table accordingly |
| Payment Test 04 | Payment details interface: Text field check | \_ | Calculate the total amount entering the unit price and quantity by clicking on calculate button | Calculate and load the text box accordingly |
| Payment Test 05 | Payment details interface: Text field check | \_ | View all the all records by clicking on the view all button | Display all the details |
| Send Email Test 01 | Email details interface: Text field check | [panasonic@gmail.com](mailto:panasonic@gmail.com)  Order Details  openFileDialog1  This is the order of the Emerging Media | Send email to relevant supplier | Display the successful message |
| Send Email Test 02 | Email details interface: Text field check | No internet connection  Attachment unavailable | Send email to relevant supplier | Display an error message |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case ID** | **Test case Description** | **Test Input** | **Test Procedure** | **Expected Result** |
| Vehicle Details 01 | Vehicle Info Interface:  Input fields  Check(Save) | 14567,Lorry,Tundra-  Toyota(Lorry),Toyota  ,15,Diesel | Enter relevant details and click on save button | Display success  Message and the enter the Vehicle info to the database |
| Vehicle Details 02 | Vehicle Info  Interface:  Input fields  Check(Update) | - | Click update button | Display a success message and update the relevant record |
| Vehicle Details 03 | Vehicle Info  Interface:  Input fields  Check(Remove) | - | Click remove button | Display a success message and remove the relevant record |
| Vehicle Details 04 | Vehicle Info  Interface:  Input fields  Check(Search) | Search category, search key word | Type a search key word in the field (Registration No) | Relevant search results will be load to the table while typing |
| Insurance Details 01 | Insurance  Interface:  Input fields  Check(Save) | 14567,Srilanka Insurance, Motor Plus - Srilanka Insurance, 19/11/2017,  19/11/2020 | Enter relevant details and click on save button | Display success message and enter the Insurance details to the database |
| Insurance Details 02 | Insurance  Interface:  Input fields  Check(Remove) | - | Click remove button | Display a success message and remove the relevant record |
| Insurance Details 03 | Insurance  Interface:  Input fields  Check(Update) | - | Click update button | Display a success message and update the relevant record |
| Fuel Details  01 | Fuel  Interface:  Input fields  Check(Save) | 14567,Diesel | Click save button | Display an error message due to validation failure |
| Fuel Details  02 | Fuel  Interface:  Input fields  Check(Remove) | - | Click remove button | Display a success message and remove the relevant record |
| External Travel Agencies  01 | External Travel  Interface:  Input fields  Check(Save) | 0717240325 ,  Lorry | Enter relevant details and click on save button | Display an error message due to validation failure |
| IMTest001 | Inventory Management Interface:Add Stock Form Interface:Input fields check(Add) | Quantity | Select the item from available items datagrid view and type the quantity in the relevant field in the selected items datagrid view and click add button | Display a successful message and enter the stock details in to the database. |
| IMTest002 | Inventory Management Interface:Add Stock Form Interface:Input fields check(Filter) | Brand, dealer, type, category | Tick one or more checkboxes from brand, type, dealer and category checkboxes and type relevant fields and then click Filter button. | Displays the relevant search result in the available items datagridview. |
| IMTest003 | Inventory Management Interface:Add Stock Form Interface:Input fields check(Delete) | - | Select the stock item from available stock and click Delete button. | Displays a successful message and remove the selected stock details from the database. |
| IMTest004 | Inventory Management Interface:Add Stock Form Interface:Input fields check(Change) | Quantity | Select the item from available items datagrid view and type the quantity in the relevant field in the selected items datagrid view and click change button | Displays a successful message and update the selected stock details in the database. |
| IMTest005 | Inventory Management Interface:Input fields check(View Stock Details In Tabular View) | - | Click the View Stock Details In Tabular View button | Shows the relevant stock details retrieved from the database in a datagrid view |
| IMT  est006 | Inventory Management Interface:Input fields check(View Stock Details In Chart View) | - | Click the View Stock Details In Chart View button | Shows the relevant stock details retrieved from the database in a chart |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case ID** | **Test case Description** | **Test inputs** | **Test Procedure** | **Expected**  **Result** |
| FM-test1 | Financial System Interface: Sales interface: input fields check(save) | Invoice No, Commercial type, client name, cost per day, duration,  No of spots | Enter data in to the relevant fields and click save button | Display successful message and save data to database and  Display datagridview of relevant data |
| FM-test2 | Financial System  Interface: Sales  Interface: input  Fields check(update) | Commercial type or client name or cost per day or duration  No of spots or can input all of above without invoice No | Enter data in to the relevant fields and click update button | Display successful message and update(replace) data to database and  Display datagridview of relevant data |
| FM-test3 | Financial System  Interface: Sales  Interface: input  Fields check(remove) | No input | Select datagridview row  And click remove button | Display successful message and remove  Data row from database and datagridview |
| FM-test4 | Financial System  Interface:Sales  Interface:input  Fields check(exit) | No input | Click exit button | Exit from the sales interface |
| FM-test5 | Financial System  Interface:Purchase  Interface:input  Fields check(exit) | Invoice no,Name,product,  Quantity,rate,amont | Enter data in to the relevant fields and click save button | Display successful message and save data to database and  Display datagridview of relevant data |
| FM-test6 | Financial System  Interface:Purchase  Interface:input  Fields check(remove) | No input | Select datagridview row  And click remove button | Display successful message and remove  Data row from database and datagridview |
| FM-test7 | Financial System  Interface:Purchase  Interface:input  Fields check(exit) | No input | Click exit button | Exit from the purchase interface |
| FM-test8 | Financial System  Interface: Expenditures  Interface:input  Fields check(save) | Expense No,name,amount,to whom,  description | Enter data in to the relevant fields and click save button | Display successful message and save data to database and  Display datagridview of relevant data |
| FM-test9 | Financial System  Interface: Expenditures  Interface:input  Fields check(update) | No,name,amount,to whom,description or all of above without expense No | Enter data in to the relevant fields and click update button | Display successful message and update(replace) data to database and  Display datagridview of relevant data |
| FM-test10 | Financial System  Interface: Expenditures  Interface:input  Fields check(remove) | No input | Select datagridview row  And click remove button | Display successful message and remove  Data row from database and datagridview |
| FM-test11 | Financial System  Interface: Expenditures  Interface:input  Fields check(remove) | No input | Click exit button | Exit from the Expenditures interface |

**3.Evaluation**

* 1. **Assessment of the Project results**

The aim of the project was to automate the supply chain management system of Emerging Media(Pvt)Ltd, to give a software solution which is integrated to perform all the functions from one main software itself. User friendliness, accuracy and reliability of the software were the key factors which were taken into consideration during the analysis of the project results.

When we were carrying on with the project work we came up with several issues during the requirement analysis as well as developing stages. Therefore, we had to visit them back and forth to clear those issues.

Discussing with the company IT department, according to their view and our ideas we both found that a standalone application is well suiting for their company since the operations which we took into consideration is performed within the head office only. Supply Chain Management System for Emerging Media(Pvt)Ltd is customized software with bit different interface and allowing the basic functionalities such as add, update, delete. Centralizing database, providing flexibility for the employees and use of many innovative methodologies and technologies, use of optimized queries which will not introduce errors and in turn providing optimum results were the initial objectives of the project which we used may innovative technologies and methodologies to achieve those.

We used face to face conversations with the client most of the time to how well they react to the interfaces and what changes they needed over the other common interfaces. With those choices, we came out with more applicable and suitable concepts to them. Once the implementation of the product was completed, the system was tested with real data values of the company to ensure the productivity of it that we expected

**3.2 Lessons Learned**

There were many lessons we learned from doing this project and it was a priceless experience we gained from it.

From the initial step itself, identifying functions meeting the requirements of the client as well as the project scope was an uneasy task. At the beginning we had to visit the client several times to gather functions. After all, to clear the issues came across during the analyzing of the function we had to be in contact with them throughout the period.

We frequently kept contacts with the IT department of the company during the working period. Developing the system with flexibility and suiting to company’s environment was a very crucial task. Since this system was a bespoke identifying the clients need and successfully providing a solution was a better experience for us in learning customer satisfaction and customization products.

Developing frameworks, new technologies were directed to us by the client and we researched for them and learned them by our own effort.

**3.3Future Work**

We can introduce more efficient coding strategies and threading concepts to efficiently use the product. More user-friendly GUI’s exciting customers usage and increase user friendliness. New frameworks and database structures can be introduced to successfully operate the system and manage data effectively.

1. **Conclusion**

The main purpose of this project was to develop an integrated software product to automate the functions related to companies supply chain management. This goal was successfully achieved from the product we implemented.

Even though some processes of the company are automated using software product, the absence of a fully integrated product automating each function is a drawback faced by the company. The product was a very helpful solution for overcoming this issue. Each functionality of the system was automated as needed for the company.

Intercommunication among inventory management and supplier management is efficiently achieved through this system. Project management function and event management function was newly introduced to the company.

A secured centralized database was used to maintain all the necessary information related to the functions. Likewise, several functions were implemented to provide necessary services to the company.

Hence, Supply Chain Management System was a successful reality which we were capable of achieving within the time frame we had for the project.