



# **Production and Distribution Management System for Jayalath Enterprise**

**Prasadinee B.A.S**  
**BIT Registration Number: R110458**  
**Index Number: 1104586**

**Name of the supervisor:**  
**Mr. J.H.P.B Jayasinghe**

**Date of Submission: 2017 November**

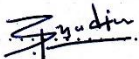


**This dissertation is submitted in partial fulfillment of the requirement of  
the Degree of Bachelor of Information Technology of the University of  
Colombo School of Computing**

# Declaration

## Declaration


I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate: ...  .....

Date: 05/11/2017

Name of Candidate: B.A.S. Prasadinec

Countersigned by:

Signature of Supervisor(s)/Advisor(s): ...  .....

Date: 05/11/2017

Name(s) of Supervisor(s)/Advisor(s): Mr. J.H.P.B. Jayasinghe

# Abstract

Jayalath Enterprise is a small shoe factory which produces their high Quality and uniquely designed shoes with about 50 employees there. Here raw materials are purchased from various suppliers and finished products are sold to selected customers throughout the Srilanka. Though huge set of information are processed and kept here all they are manual which is not effective for such an active factory. So here we proposed a solution based on information communication technology.

This system was proposed to build as a web base system because some data has to be uploaded, processed and viewed both inside and outside the factory. Many functions hope to be carried out by proposed system like supply management, stock management, sales management, employee management, transport and accounts handlings.

The system has been designed with logins for few selected employees in first level considering user friendliness and easy data communication.

Considering the present Software Engineering theories and by studying the problem domain, the Rational Unified Process (RUP) was identified as the most relevant development methodology based on its flexible developing theory and related advantages compared to other software methodologies. When developing the system, Unified Modeling Language (UML) was used for analysis and designing. When drawing necessary diagrams in design phase Argo UML was used as designing tool. Proposed system was developed using PHP (PHP Hypertext Preprocessor) as server-side scripting language while using JS (Java Script) and CSS (Cascading Style Sheet) with it and MYSQL as database tool, in XWAMP Development environment. And NetBeans would be used as coding tool.

Now the proposed system is completed and is achieved anticipated benefits relevant to identified requirements of Jayalath Enterprise.

# Acknowledgement

I express my gratitude to everyone who helped me and guided me to complete the project successfully. I wish to thank my project supervisor Mr.W.A.P.Jayasinghe, ICT officer at Department of Agriculture, who gave the technical guidance and constant supervision despite his busy schedule, and also I should thank to Mr.Susantha Subhasinghe, The Manager and Mrs. B.C.Jayalath, the proprietied of the factory for giving me the domain knowledge and providing necessary information and documents regarding the project.

My sincere appreciation goes to the staff of Jayalath Enterprise, who enlightened me about the process and functionalities of the factory and co-operated in the course of development of the project. I wish to express my gratitude to the BIT coordinator of University of Colombo School of Computing (UCSC) and project evaluation board of Bachelors of Information Technology (BIT) for giving me the opportunity to apply the knowledge gain through the BIT degree program.

I also grateful to my family and friends especially to my parents, for supporting and encouraging me to complete this project successfully.

# TABLE OF CONTENTS

## Contents

Declaration.....	ii
Abstract .....	iii
Acknowledgement .....	iv
TABLE OF CONTENTS.....	v
List of Figures .....	viii
List of Tables .....	x
List of Acronyms .....	xi
CHAPTER 01 – INTRODUCTION .....	1
1.1 Background .....	1
1.2 Motivation for project .....	1
1.3 Objectives and scope of proposed project .....	4
CHAPTER 02 – ANALYSIS.....	6
2.1 Introduction.....	6
2.2 Fact Gathering Techniques .....	6
2.3 Current Manual System .....	7
2.4 Similar Systems and Literature Review.....	9
2.4.1 Ultriva Lean Factory Management .....	9
2.4.2 Classic Factory Management System with ERP.....	10
2.5 Functional Requirements .....	11
2.6. Non- Functional Requirements .....	13
2.7 Users of Proposed System .....	14
2.8 Methodology .....	15
2.8.1 Waterfall Model .....	15
2.8.2 Evolutionary Development .....	15
2.8.3 Spiral Model.....	15
2.8.4 Prototyping Model .....	15
2.8.5 Rapid Application Development.....	15
2.8.6 Rational Unified Process(RUP) – Selected methodology for the System .....	16
CHAPTER 03 – DESIGN.....	18
3.1 Introduction.....	18
3.2 Alternate solutions Evaluation.....	18

3.2.1 developing from scratch.....	18
3.2.2 Standalone system.....	18
3.2.3 Software collection .....	19
3.3 Selected Solution .....	19
3.4 Design Techniques.....	19
3.5 Production and Distribution Management System Design .....	20
3.5.1 Use Case Diagram.....	21
3.5.2 Class Diagram.....	33
3.5.3 Sequence Diagram .....	34
3.6 Database Design of the system .....	35
3.6.1 Database diagram.....	35
3.6.2 ER diagram .....	36
3.7 User Interface Design .....	37
3.7.1 Login Interface.....	38
3.7.2 Dash board .....	38
3.7.3 Register Employee Interface.....	39
3.7.4 Release material interface .....	39
3.7.5 Supply order note Interface.....	40
3.7.6 Place Order interface.....	40
3.7.7 Issue invoice interface.....	41
CHAPTER 04 – IMPLEMENTATION.....	42
4.1 Introduction.....	42
4.2 Implementation Environment .....	42
4.3 Reused Modules.....	43
4.4 Major Code Segments.....	44
CHAPTER 05 – EVALUATION .....	51
5.1 Introduction.....	51
5.2 Software Testing Methods .....	51
5.3 Software Testing Types.....	52
5.4 Test Plans and Test Results.....	53
5.4.1 Test Cases of Sales Module .....	53
5.4.2 Test Cases of Product Module .....	55
5.4.3 Test Cases of Stock Module .....	56
5.5 User Evaluation (Acceptance testing).....	58

CHAPTER 06 – CONCLUSION.....	60
6.1 Introduction.....	60
6.2 Future Improvements .....	60
6.3 Lesson Learnt .....	61
Reference .....	62
Appendix A - System Documentation .....	64
Appendix B - Design Documentation.....	66
Appendix C - User Documentation.....	73
Appendix D - Management Reports .....	78
Appendix E – Test Results.....	79
Appendix F - Code Listing .....	82
Appendix G - Client Certificate.....	87
Glossary .....	88
Index .....	89

# List of Figures

Figure 2.1 Use Case Diagram for the existing system.....	8
Figure 2.2 Customer order interface of the system.....	9
Figure 2.3 Rational Unified Process Model.....	17
Figure 3.1 high level Use-Case diagram of the system .....	21
Figure 3.2 high level Use-Case of the sales module.....	22
Figure 3.3 high level Use-Case of the Production module .....	25
Figure 3.4 high levels Use-Case of the stock module.....	27
Figure 3.5 high level Use-Case of the supply module.....	31
Figure 3.6 Class Diagram of the Proposed System.....	33
Figure 3.7 Sequence diagram of the proposed system.....	34
Figure 3.8 Database diagram .....	35
Figure 3.9 ER diagram.....	36
Figure 3.10 Login Interface .....	38
Figure 3.11 Dash board.....	38
Figure 3.12 Register Employee Interface .....	39
Figure 3.13 Release materials Interface.....	39
Figure 3.14 Supply Order note Interface .....	40
Figure 3.15 Place order Interface.....	40
Figure 3.16 Issue invoice Interface .....	41
Figure 4.1 MCV stucture .....	44
Figure 5.1 Feedback Questionnaire .....	59
Figure B.1 high level Use-Case of the HR module .....	66
Figure B.2 high level Use-Case of the transport module.....	67
Figure B.3 high level Use-Case of the account module.....	69
Figure C.1 Logging Form .....	73
Figure C.2 dash board.....	73
Figure C.3 navigation bar .....	74
Figure C.4 Register material interface .....	74
Figure C.5 place order interface.....	75
Figure C.6 Buttons in the table .....	75
Figure C.7 alert .....	76
Figure C.8 Interface of register material after clicking update button.....	76



Figure C.9 Interface of place order after clicking update button .....	77
Figure C.10 Interface of place order after clicking update button .....	77
Figure C.11 pop up modal of place order .....	77
Figure D.1 pop up modal of place order .....	78
Figure D.2 income chart .....	78

# List of Tables

Table 2.1: Details of users of the proposed System.....	14
Table 3.1: use case narrative of register shop .....	23
Table 3.2: use case narrative of place an order .....	<b>Error! Bookmark not defined.</b>
Table 3.3: use case narrative of issue invoice.....	24
Table 3.4: use case narrative of Record return products.....	24
Table 3.5: use case narrative of generate reports .....	25
Table 3.6: use case narrative of mark attendance .....	26
Table 3.7: use case narrative of Add daily work .....	26
Table 3.8: use case narrative of mark attendance .....	27
Table 3.9: use case narrative of register material .....	28
Table 3.10: use case narrative of register product .....	28
Table 3.11: use case narrative of add materials .....	29
Table 3.12: use case narrative of add product.....	29
Table 3.13: use case narrative of issue materials .....	30
Table 3.14: use case narrative of issue product .....	30
Table 3.15: use case narrative of make supply order.....	32
Table 3.16: use case narrative of make supply order.....	32
Table 4.1 Implementation Environment .....	42
Table 5.1: Test cases and results of sales module.....	54
Table 5.2: Test cases and results of product module .....	55
Table 5.3: Test cases and results of stock module .....	57
Table B.1: use case narrative of register employee .....	66
Table B.2: use case narrative of add logging details .....	67
Table B.3: use case narrative of register vehicle .....	68
Table B.4: use case narrative of register charges.....	68
Table B.5: use case narrative of Allocate cash for sales trips.....	69
Table B.6: use case narrative of Allocate extra cash .....	70
Table B.7: use case narrative of Report income .....	70
Table B.8: use case narrative of Allocate advance .....	71
Table B.9: use case narrative of calculate payment.....	71
Table B.10: use case narrative of Allocate advance .....	72
Table E.1: Test cases and results of Supply module.....	80
Table E.2: Test cases and results of HR module .....	81

# List of Acronyms

DB – data Base

ER - Entity Relationship

Etc. – etcetera

Ex- Examples

NIC – National Identity Card

No. – Number

PADMS -Production and Distribution Management System

RUP- Rational unified process model

SMS – short Message Sending

# **CHAPTER 01 – INTRODUCTION**

## **1.1 Background**

Production is essential factor in a society, which is spread various directions. Production is taken place in modern society in a competitive environment, because same product is produced by different producers adding unique features and quality. And when we consider a certain product, some companies play a giant role with a good market both local and foreign. So, improving of small scale production system in same item is not easier.

And in any production process huge set of information has to be handled for an effective and efficient production process.

After the presence of information communication technology in the world, numerous positive changes took place in life pattern as well as in industry. So it is wise to implement a computer based information system in production environment with precious use of internet.

Jayalath Enterprise is a regional shoe production factory with a good market over many cities in Srilanka which run under the above mentioned condition with a set of considerable and important information handling aspects. So, it is essential to manage supply, stock, sales, accounts and employees using a proper system when overcome challenges and approach their goals. Then only they will be able to make their mission, a reality.

## **1.2 Motivation for project**

Jayalath Enterprise is a small-scale garment factory which produces high quality male shoes for local market in Sri Lanka. This factory has been established about 20 year back and now managed by the second generation which is run as a family business. And the factory is situated in a rural area I nakkawaththa. Though it's not much famous as an industry, they distribute their quality products throughout the Sri Lanka in a competitive

environment with branded shoes. And now they have improved their business with child shoes which aimed at school boys in Sri Lanka.

Following functioning units can be seen here

- Supply process
- Production
- Stock keeping
- Sales and delivery
- Accounts management
- Employee handling

Jayalath Enterprise has introduced their own unique designs for their products with many variations. And the quality and selective materials are exclusively expended in producing excellent merchandises.

They purchase material from suppliers in Colombo market places who import them directly. They order them according to the necessity in the production and some are purchased regularly in the process. And some soles are acquired from a manufacture at Kaluthara.

In the factory about 50 employees works including Manager, Supervisors, accountant, skill workers, helpers and drivers.

Material transportation and products distribution take place through their own vehicles and some rented vehicles.

Sales executive take orders from their regular customers all over the Sri Lanka and inform the factory supervisor. He checks the availability at the shoe stock and if stock is not enough direct production process towards the fulfilling of the order. Sales executive issues invoice for customer and handover money to accountant after delivery of product take place.

Manager observe entire process while supervisor keep records in production, stock keeper keeps both material and finishing stocks and accountant maintain cost, income and payment details.

Though Jayalath Enterprise is well functioning garment factory, each information handling is done manually. It makes the situation in the industry in to a poor efficiency and prevents improvements of the business.

Under these conditions, prevailing manual system is much time and unnecessary energy consuming and poor to survive in the industrial contest. On other hand marketing procedure here is not strong enough in this shining information era based on internet. So, achievements of their goals have become slow.

In this circumstance the project is focused on building an online Production and Distribution Management system for Jayalath Enterprise, which can be used to handle their information efficiently and effectively.

As a result of a complete research on the main functional areas of the Jayalath Enterprise I was able to discover the following problematic areas which make the factory less efficient

- Material Stock data is not kept. So, material supply process is not starched. It disrupts the efficient production process.
- Data in finishing stock is not kept, and it is checked when an order take place. It's time consuming and not supported in both production and supply processes.
- Employee handling is done total manually.
- All accounts are kept manually through accountant. So, accuracy and auditing become problematic.
- Sales executives place orders formally after visiting the factory again. So, there is no proper and effective coordination between sales executive and factory.
- Poor Security of accounts data.
- Data of product returns is not managed well.
- Manual record keeping makes administrators busier and it is time consumed to find records quickly.
- Periodical report generation done in manually which is time and energy consumed.

Because of the mentioned reasons I have decided to propose a Production and Distribution Management System to improve efficient and cost-effective way with more benefits for Jayalath Enterprise.

## **1.3 Objectives and scope of proposed project**

### **Objectives of the project:**

- Our main objective was to get the experience of working with a real client. When we are working with the client we learn how to take responsibility and how can we identify the user requirements and their expectations.
- Make the supply chain effective.
- Keep stocks data formally which would support in supply, production and sales.
- Increase the productivity of the whole production process.
- Provide an easy to use and reliable sales information handling and communication method.
- Improve employee management process.
- Keep accounts accurately effectively and securely.
- Make transport service data formal according to the service and through it reduce unnecessary cost.
- Manage returns well.
- Allow administrators to observe entire process.
- Help administrators to analyze things and generate timely reports for future improvements.

### **Scope of the project:**

- Supply management
- stock management
- Production data management

- Transport data management
- Employee handling
- Accounts handling
- Sales management
- Returns handling
- Login management
- Report generation
- High security of data access
- User management
- Database management



# CHAPTER 02 – ANALYSIS

## 2.1 Introduction

“System analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way" [1]

To create a most efficient and effective system, in this phase functional and non-functional requirement are collected. And this can be named as operations research.

In this chapter, fact gathering techniques of this project are presented, and prevailing manual system is studied and is expressed briefly using a top-level use case diagram. Functional and non- functional requirements to be accomplished in the new systems are forwarded in this chapter.

## 2.2 Fact Gathering Techniques

Among many Fact Gathering Techniques, several required and suitable techniques were selected to collect information about the prevailing system and to get an idea about new system which is hoping to be implemented. Following techniques were used.

- Interview
- Observation
- Sampling & Documentation

Both direct and indirect observations were used to identify the basic function of the factory environment. Few stakeholders were interviewed to understand the system further with their requirements. Open ended, close ended, formal and informal interviews were done to collect more information about prevailing system requirements, scope, and roles with various responsibilities, problematic areas and critical conditions. And few documents were reviewed get a clear idea about the system. Meeting with domain experts made it easy to understand the real necessity.

Above mentioned techniques were appropriate and useful when understanding manual system and designing new system with additional requirements.

## **2.3 Current Manual System**

Main functions of the current manual system can be identified as follows.

- Purchase raw materials according to order need or regular basis by manager
- Make sales orders from customers by sales executive (rep)
- Supervise production process and have records by supervisor
- Keep both material and production stock by stock manager
- Keep cost and payment details by accountant
- Keep employee work records, attendance, advance payment and make payments
- Managers manage whole process while handling employees and checking payments details.
- Create reports manually as necessary.

### **Material Supply**

There are few regular suppliers to Jayalath Enterprise from Colombo and nearby regions. Most of materials are purchased regular basis and sometimes due to quick product orders, quick material supply take place. Through the manager material purchase happened and payments are done through accountant

### **Sales**

Sales executive (rep) coordinate sales process. Regular customers of jayalath enterprise are visited by sales executive and take orders. After visiting the factory orders are handed over to supervisor. Through accountant an invoice is prepared and with product load it is brought to customer by factory transport service.

### **Stock Keeping**

Both materials and finished goods are kept in stores under a store keeper. But store records are not kept well. Products and materials are stored and release accordingly and store is checked only for need.

### **Employee Handling**

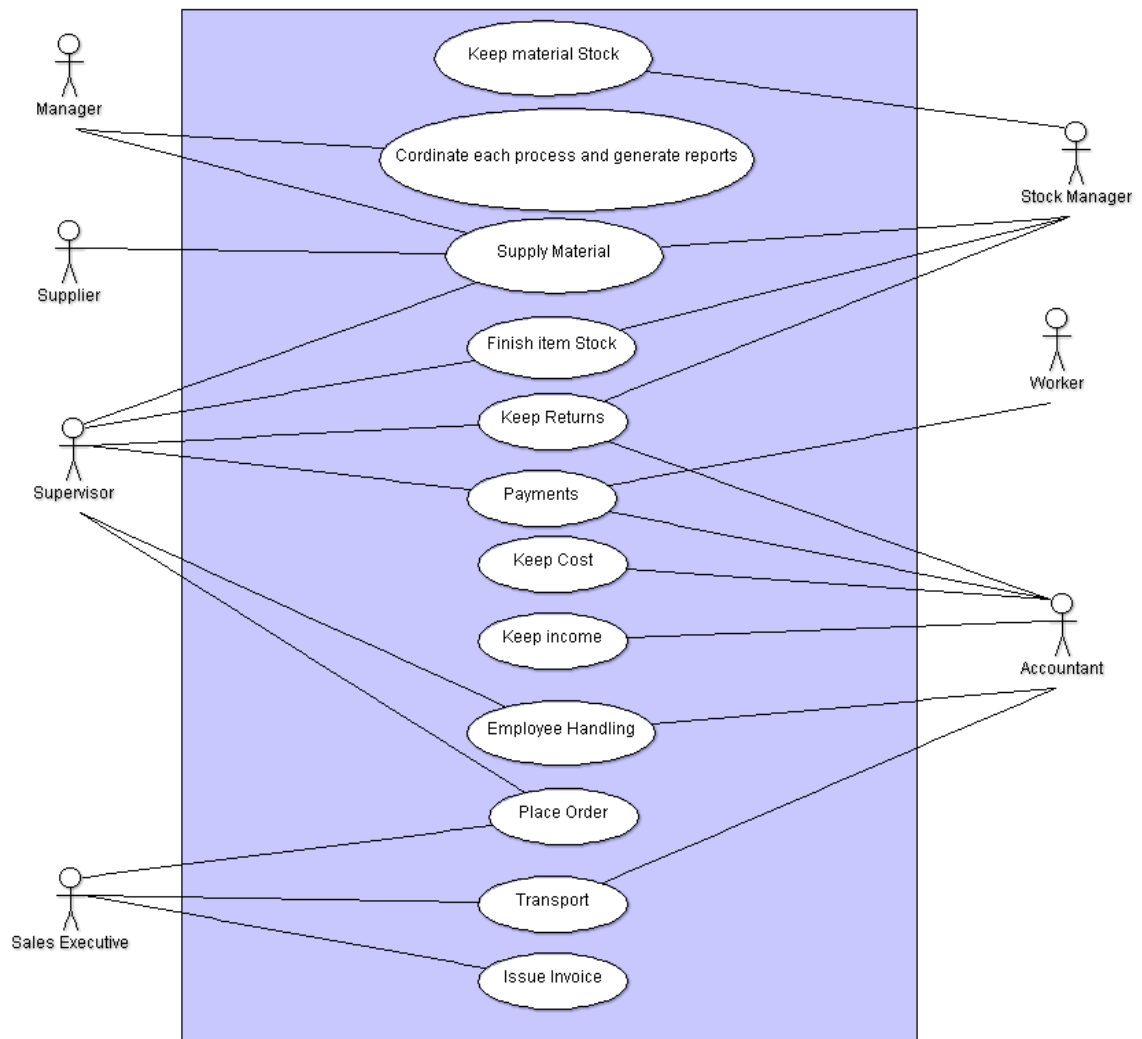
Employee attendance is marked, and work pieces are recorded daily for each and every employee for salary calculation, while top level workers are given a fix salary amount. And employees are able to get salary advance.

### **Accounts managing**

Payments are done for material purchases through accountant. And the salary for employees is calculated using employee data. Invoices are issued for orders and data like return values and Report pending of income and balance of payments. And cash allocation for transport also done through accountant.

### Management

Manager coordinates all above processes while making necessary reports for future decisions.



**Figure 2.1 Use Case Diagram for the existing system**

## 2.4 Similar Systems and Literature Review

Many similar web based systems including same functionalities of our proposed systems can be seen. By studying such current systems more ideas can be obtained. Following are few similarly systems which were reviewed to build the system.

### 2.4.1 Ultriva Lean Factory Management

“Ultriva addresses the challenges of production inefficiency with Ultriva Lean Factory Management (LFM), a cloud-based application that optimizes production job sequencing and supply chain material replenishment. LFM can work in conjunction with other Ultriva LeanSuite applications or function as an independent application.” [2]

In this system modules relevant to our system could be seen. And there were many models of interfaces and features which can be exemplified for our proposed system.



The screenshot displays the Ultriva Customer order interface. At the top left is the ULTRIVA logo. Below it, the text "Active Job is :Z40960-001" is shown. A legend indicates job status: a green box for "Active Job", a yellow box for "Stopped Not Finished", and a red box for "ShipToday". Below the legend are two buttons: "Start Production" and "Assemble From Components". The main part of the interface is a table with a header row and several data rows. The header row is labeled "Today" and "CinCom Status". The table columns are: M/C, Part #, Customer Name, Order, and Cust. Want Date. The first row in the table is highlighted in red and green, indicating it is the active job.

M/C	Part #	Customer Name	Order	Cust. Want Date
MB	MB251-PA	KESFIELD INDUSTRIES - LA	K40960-001	01/05/13
LB	FX3Y216E3	KESFIELD INDUSTRIES - LA	K40962-001	01/09/13
MB	MB251-PA	KESFIELD IND - COL WHS	K40961-001	01/09/13
MB	NFC2251	KESFIELD IND - COL WHS	K36971-001	01/11/13
MB	NFC4251	KESFIELD IND - COL WHS	K36973-001	01/11/13
LB	WP3S216E	KESFIELD INDUSTRIES - LA	K33414-001	01/12/13
MB	ER16-HFF6	KESFIELD IND - COL WHS	K40383-001	01/13/13
LB	UG216N	KESFIELD IND - COL WHS	K37920-001	01/13/13

Figure 2.2 Customer order interface of the system

## 2.4.2 Classic Factory Management System with ERP

Features were studied relevant to our system

### “ Features:

- Address Maintenance:
  - Buyer Master [Area Wise].
  - Supplier Master.
  - Buyer's Listing.
  - Label Printing.
- Store Maintenance:
  - M.R.R. Entry.
  - M.R.I.R. Entry.
  - M.R.I.R. Printing [Supplier wise, Product Wise, Pending].
  - M.R.I.R. Printing [Supplier wise, Product Wise, Pending].
  - Rejection Statement.
  - Inward Chalan Register.
  - D.C. Entry [Coded, Sundries].
  - D.C. Printing [Coded, Sundries].
  - Issue Entry.
  - Issue Register [Dept. Wise, Item Wise].
  - Material Return Entry.
  - Return Register [Dept. Wise, Contractor Wise].
  - Product Listing.
- Purchase Maintenance:
  - Purchase Order Entry.
  - Purchase Order Printing.
  - Purchase Order Listing [Supplier Wise, Item Wise, Status].
  - Purchase Order Outstanding [Supplier Wise, Item Wise].
  - Purchase Order Annexure Printing.
  - P.O. Excess Material Received Report.
- Account Maintenance:
  - Account Code Entry.
  - Group Code Entry.
  - Graphical Analysis of all Heads [Bar / Pie / Line].
  - Opening Balance Entry.
  - Voucher Entry [Cash / Bank / Journal Vouchers].
  - Post Dated Cheque Handling Incoming Cheque.
  - Post Dated Cheque Handling Outgoing Cheque.
  - Today Collection Cheque Listing.
  - Return Cheque Entry.
  - Cash Book Listing.
  - Day Book Listing.
  - Trail Balance [Opening, Monthly, Cumulative, as on date]
  - Profit and Loss Report.
  - Balance Sheet.

- Utility:
  - • Daily Backup.” [3]

Above mentioned features were studied and some of them were captured to our system.

## 2.5 Functional Requirements

“a functional requirement’ is that it essentially specifies something the system should do. Typically, functional requirements will specify a behavior or function” [4]

All functional requirements for the computer based system of Jayalath Enterprise are gathered during requirement analysis are listed below.

- log in for various users
  - Except skilled workers and drivers, other employees of the factory can log in to the system. User Accounts are created By General Manager. All system users have their user name and password to log the system.
- Supply Module
  - Varied materials are supplied to the factory from various suppliers. In the supply process following functional requirements are available.
    - Register Suppliers
    - Place an order
    - Check Material Stock
    - Generate Reports
- Sales Module
  - Sales is a main functional unit because income is based on sales. In sales, following functions happens.
    - Select sales points
    - Place orders
    - Issue invoice
    - Report Returns
    - View sales
    - Check Product Availability

- Put material release not
- Generate Reports
- Stock Module
 

Both material and product stocks are there, where a stock keeper engages in following functions.

  - Add material
  - Return damages to production
  - Deduct material
  - Add products
  - Deduct products
  - Accept Returns
  - Report shortage
  - View Stock
  - Add new items
- Production Module
 

Product supervisor supervise work units and update relevant data on production.

  - Count daily pieces of works
  - Collect finish items and update stock
  - Release material and update stock
- HR Module
 

Different type of employees works here. After recruiting them, many functionalities are kept going regarding to them.

  - Add employee
  - Remove Employee
  - View worker information
  - Keep attendance
  - Allocate advance
  - Monthly payment
- Transport Module
 

Transport is used in supply and sales. In transport following functions should be accomplished.

- Add charges
  - Add rent vehicles
  - Add sales ad supply points
  - View sales points and transport information
  - Request extra cash and allocate cash for transport
  - Schedule sale trip
  - Accounts Module
- Accounts are handled by accountant. Cost and income should be calculated.
- Following functions can be seen in accounts module.
- Report income
  - Report cost
  - Allocate cash
  - Add return values
  - Accept signed invoice
  - Allocate advance
  - Calculate payments
  - View accounts
  - Generate reports

## **2.6. Non- Functional Requirements**

“A non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes for of a system.” [5]

- Accuracy of information and generated reports.
- Security of information
- User friendliness.
- Reliability



## 2.7 Users of Proposed System

Not all users but users having few job titles hope to use the propose system when, fulfilling above given requirements.

Users of proposed system, Number of that type of users, their relevant jobs and supposing frequency of using the proposed system is given in table 2.1

Users	No. of Users	Job of Users	Frequency of using the System
Manager	2	Register Employees, products, materials, shops and suppliers Make supply orders Observe all functionalities of System View all reports	Once a day
Production Supervisor	4	Record attendance of employees Record daily work out of workers at production	Two thee times per day
Stock Manager	2	Request material shortage from management Issue materials to production department Issue products for Sale Keep Stock Information updated	Two thee times per day according to need
Accountant	2	Allocate cash for material, transport and employees Record invoice details of supply Record cash of sales	Working hours of the day
Sales Representative	5	Place orders for shops Deliver products and issue invoice Request necessary requirement for the sales trips (Cost, vehicles, drivers)	Once or twice a week

Table 2.1: Details of users of the proposed System

## **2.8 Methodology**

“A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system” [6]

### **2.8.1 Waterfall Model**

“The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.” [7]

### **2.8.2 Evolutionary Development**

Evolutionary development this approach interleaves the activities of specification, development and validation. An initial system is rapidly developed from abstract specifications. This is then refined with customer input to produce a system that satisfies the customers’ needs.

### **2.8.3 Spiral Model**

The Spiral Model is the most flexible and agile of all traditional software process models.

The process begins at the center position.

From there it moves clockwise in traversals. Each traversal of the spiral usually results in a deliverable

### **2.8.4 Prototyping Model**

“The prototyping model is a systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved” [8]

### **2.8.5 Rapid Application Development**

“Rapid application development (RAD) is both a general term used to refer to alternatives to the conventional waterfall model of software development as well as the name for James Martin's approach to rapid development.” [9]

“Rapid-development language” is a general term that refers to any programming language that offers speedier implementation than do traditional third-generation languages such as C/C++, Pascal, or Fortran. Rapid-Development Languages (RDLs) produce their savings by reducing the amount of construction needed to build a product” [10]

### **2.8.6 Rational Unified Process(RUP) – Selected methodology for the System**

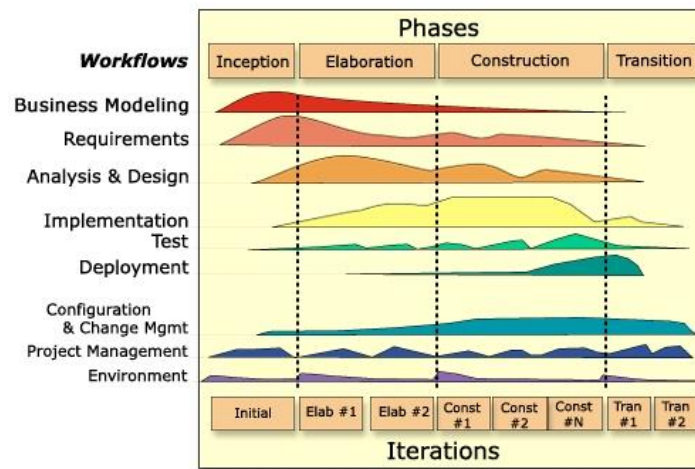
Rational unified process model (RUP) was chosen despite from the above explained models. RUP is an adaptable process framework, where the elements of the processes can be changed according to the user’s needs. This mainly concentrates on risk reduction due to past failures of other projects. The RUP lifecycle comprises four main phases namely: Inception, Elaboration, Construction and Transition. descriptions of each phase are given Below.

- **Inception phase** is where the business case is defined which includes the business context of the project. If the project does not pass this milestone, called the Lifecycle Objective Milestone, it either can be cancelled or repeated after being redesigned to better meet the criteria.
  
- **Elaboration phase** is where the primary objective is to mitigate the key risk items identified by analysis up to the end of this phase. The elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form. Lifecycle Architecture Milestone is reached in this phase.
  
- **Construction phase** is primary forces to build the software system. In this phase, the main focus is on the development of components and other features of the system. This is the phase when the bulk of the coding takes place. In larger projects, several construction iterations may be developed in an effort to divide the use cases into manageable segments that produce demonstrable prototypes. This phase produces the first external release of the software. Its conclusion is marked by the Initial Operational Capability Milestone.
  
- **Transition phase** the primary objective is to 'transit' the system from development into production, making it available to and understood by the end user. The activities of this

phase include training the end users and maintainers and beta testing the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase. If all objectives are met, the Product Release Milestone is reached, and the development cycle is finished.

[11]

Figure 2.3 illustrates the phases and activities done in each phase of the RUP life cycle.



**Figure 2.3 Rational Unified Process Model**

# CHAPTER 03 – DESIGN

## 3.1 Introduction

“Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development” [12]

Throughout this chapter, Alternate solutions which can be used for the analyzed requirements are discussed. And, despite all alternative solutions, selected solution is also presented. The Object-Oriented design techniques (UML diagrams) were used throughout the process and ER diagrams were used in Data Base Design.

## 3.2 Alternate solutions Evaluation

Other than developing web based Production and Distribution system, manage with a standalone system and have a software collection to fulfill tasks done which are done manually can be presented as alternate systems.

### 3.2.1 developing from scratch

Designing a website from scratch means creating a website from starting, without taking help of any theme or template. In this solution, all of the code is custom written by a web developer. This can be expensive unless it is a small system that does very little.

### 3.2.2 Standalone system

It is a computer system that functioning independently without any connection of other functioning systems. When compare standalone system as an alternative it may cost a lot when developing. Also, manipulation and maintain data and information is difficult and time consuming because there is no any interrelation in between each terminal. All terminals are also functioning separately. So, changes of data are maintained separately to keep integrity and consistent of data. Because a change done in one terminal not affecting to others.

### **3.2.3 Software collection**

Using separate software to perform different tasks in each section also can complete their day to day work as necessary. As an example;

- Use Billing system to maintain payment details
- Use Report generating software to generate collective reports

But using separate software for separate tasks cause big money wastage. And it consumes lot of time when interacting with each section because data handle using separate software has no interconnection. Also the factory should pay money for features that the factory dose not even use.

## **3.3 Selected Solution**

Despite of above discussed ways of development, a web based system is decided to be developed using some open source components like bootstrap, icons, calendrers etc...

### **Reasons to choose the web based system**

- Some top-level employees want to access the system to retrieve information when they are outside the factory.
- System is platform independent because users interact with the system via web browser.
- Easy deployment.
- Can be implemented on client-server architecture.
- Maintenance is easy, because the database is centralized, and everything is synchronized.
- Easy to manage because using minimum requirements on the user work station.
- Cost effective and timeliness.

## **3.4 Design Techniques**

Every well-functioning system has a good requirement definition and good designing methodology. A methodology can be defined as underlying principles and rules control a system. Although there are various designing practices like Modern Structured design, Prototyping, Rapid Application Development; Object Oriented design technique was

chosen out of them for the main design concepts like abstraction, inheritance, modularity, reusability, encapsulation and message passing. Unified Modeling Language (UML) plays an important role in Object Orientation. UML allows us to build easy to use and easy to understand models of objects so that programmers can easily write software. Where,

- Use-Case diagrams with some narratives – shows what the system needs to do.
- Sequence diagrams – shows how the objects interact overtime.
- Activity diagrams – shows object states at a specific timeline.
- Class diagrams – shows the needed objects and relationships between them.

Since the object-oriented methodology uses all classes and objects to make use of its main features, the classes and the relationships between them were identified to implement the business logic of the system. Attributes of the classes were identified to build the data layer where all the relevant tables were designed, and normalization was carried out to validate and improve the logical design.

### **3.5 Production and Distribution Management System Design**

To describe the design of the proposed system, use case diagrams, use case narratives, class diagrams and sequence diagrams were used.

Use case diagram for total system is presented and use case diagrams for modules also used separately to clarify the design furthermore. Figure 3.1 shows high level Use-Case diagram of the system.

### 3.5.1 Use Case Diagram

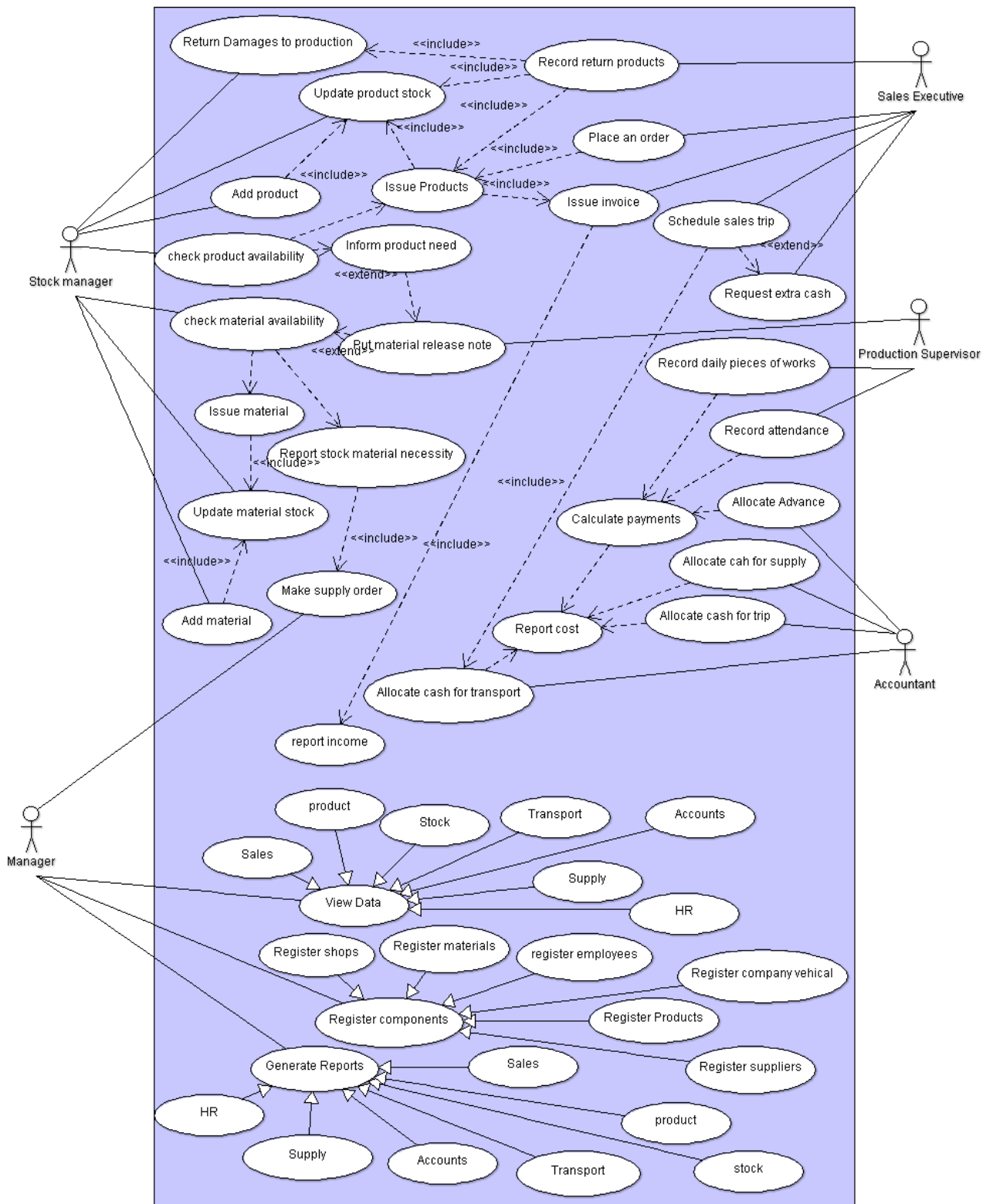


Figure 3.1 high level Use-Case diagram of the system



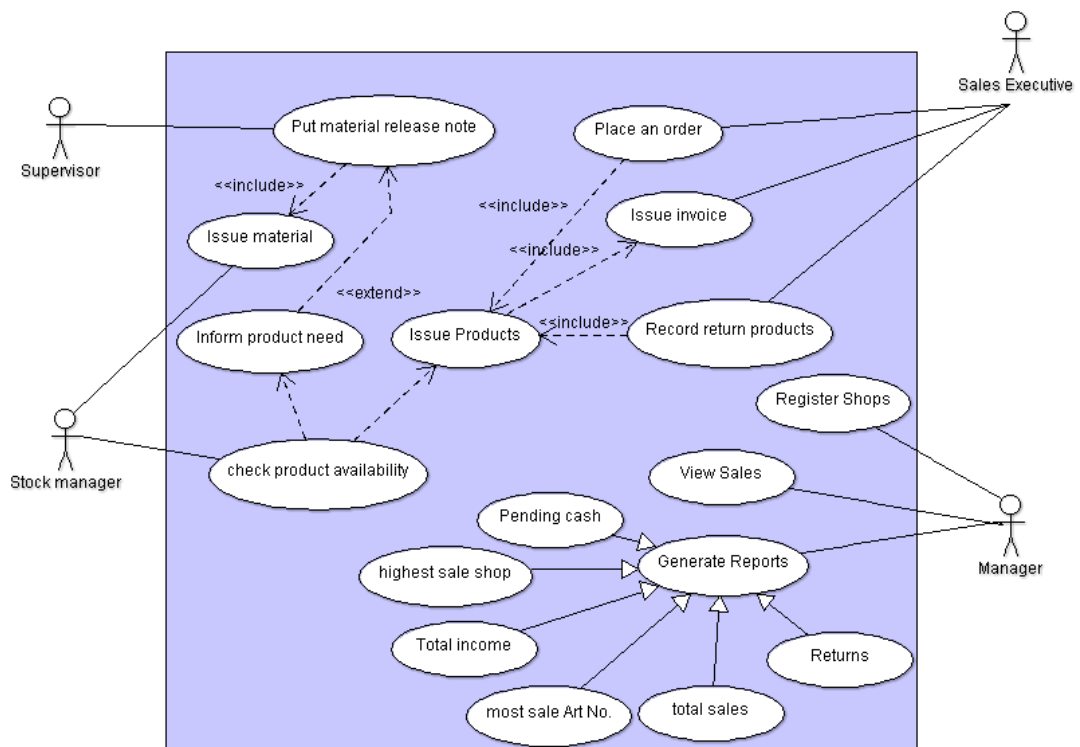
The proposed system is divided in to several modules in order to make the development easier. The modules as follows

- Sales module
- Supply module
- Stock module
- Production module
- HR module
- Accounts module
- Transport module

The modules are described below with main use cases along with narratives.

### Sales Module

Figure 3.2 shows high level Use-Case diagram of sales module.



**Figure 3.2 high level Use-Case of the sales module**

This module is used to describe how sales order is placed, issue products and issue invoice. In sales module sales executive and stock manger involve essentially while supervisor and manger also contribute.

1. Use case description for register shop

<b>Use case</b>	Register shop
<b>Actor</b>	Manager
<b>Over view</b>	
Factory deliver their products only to their registered customers. Therefore, to sale products shops should be registered in the system early	
<b>Precondition</b>	
Shop should be a customer of the factory	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form with relevant information</li> <li>• submit</li> </ul>	
<b>Postcondition</b>	
Shop details can be removed or edited later. Shop details are used in place order, schedule sales trip and reports	

Table 3.1: use case narrative of register shop

2. Use case description for place order

<b>Use case</b>	Place an Order
<b>Actor</b>	Sales Executive
<b>Over view</b>	
Through a visit or phone, Sales Executive places a shoe order for a shop. In one order several shoe types and sizes with quantities would be contained. A due date is also mentioned.	
<b>Precondition</b>	
Shop should be a registered one of factory.	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form with shoe sets having art no., size, amount and due date</li> <li>• submit</li> </ul>	
<b>Postcondition</b>	
Order can be removed or update later. Oder details are used in releasing product from stock and issuing invoice	

Table 3.2: use case narrative of place an orde

### 3. Use case description for Issue invoice

<b>Use case</b>	Issue invoice
<b>Actor</b>	Sales Executive
<b>Over view</b>	
When stock release products relevant to the order, sales executive carry away products to the shop and issue invoice. Consider payment method of shop and apply discounts.	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Oder placement for the shop</li> <li>• Release product from product stock</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select relevant order</li> <li>• Select payment method. (if it is cash only 10% discount is applied by design)</li> <li>• Save invoice and print invoice</li> </ul>	
<b>Postcondition</b>	
Issuing invoice can be undo if there is any change of payment. Printed invoice is given to shop. (If Cheque is accepted as payment will be changed after 30 days)	

Table 3.3: use case narrative of issue invoice

### 4. Use case description for Record return products

<b>Use case</b>	Record return products
<b>Actor</b>	Sales Executive
<b>Over view</b>	
Due to damages of shoes or sales problem over time some products are returned. When return take place, cash is not returned. New products are replaced for reasonable returns. Returns are recorded as removing items and repairing items	
<b>Precondition</b>	
There should be acceptable reason to record a return	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form and submit</li> </ul>	
<b>Postcondition</b>	
Repairable products are sent to production. Releasing new products for returned products or repair returns and deliver back. Lost for the factory, responsible person is recorded	

Table 3.4: use case narrative of Record return products

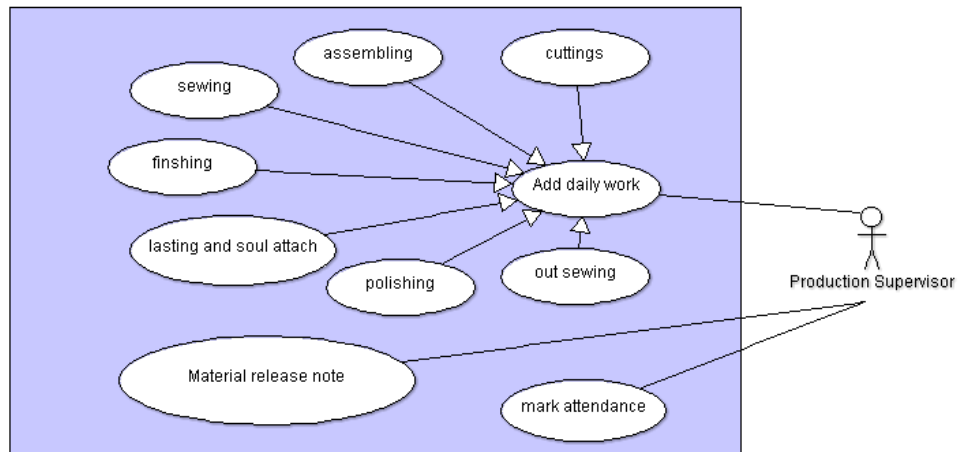
5. Use case description for Generate reports in sales

<b>Use case</b>	Generate reports
<b>Actor</b>	Manager
<b>Over view</b>	In sales process placing sale order, issue invoice, accept returns are happened typically. Information flow through these processes are used to generate reports
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Placed orders</li> <li>Issued invoices</li> <li>Accepted returns</li> </ul>
<b>Flow of events</b>	<ul style="list-style-type: none"> <li>Select relevant button</li> <li>Generate reports (income of month, most sale art no, highest sale shop etc.)</li> </ul>
<b>Postcondition</b>	Reports should be accurate as Management use reports in decision making

Table 3.5: use case narrative of generate reports

**Production Module**

Figure 3.3 shows high level Use-Case diagram of production module.



**Figure 3.3 high level Use-Case of the Production module**

1. Use case description for mark attendance

<b>Use case</b>	Mark attendance
<b>Actor</b>	Production supervisor
<b>Over view</b>	
By default, system mark presence for all worker. Production manager should select absentees only to keep daily attendance.	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>Employee(worker) should be registered</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>Select absentees</li> <li>Save attendance</li> </ul>	
<b>Postcondition</b>	
Attendance is used in adding daily work and calculating salary end of the month	

Table 3.6: use case narrative of mark attendance

2. Use case description for Add daily work

<b>Use case</b>	Add daily work
<b>Actor</b>	Production supervisor
<b>Over view</b>	
Manufacturing of a pair of shoe is divided in to seven works units.one skilled worker works at one work unit. Rarely one worker engages in another work unit. No of pieces, a worker goes through in a day is recorded by Production manager.	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>Register workers</li> <li>Mark daily attendance</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>Record No. of pieces completed by a worker at a particular work unit</li> </ul>	
<b>Postcondition</b>	
When recording No. of pieces completed by a worker, considering the work unit is vital as payments are differ for different work units. And these daily work details are used in monthly salary calculation.	

Table 3.7: use case narrative of Add daily work

### 3. Use case description for material release note

<b>Use case</b>	Material release note
<b>Actor</b>	Production supervisor
<b>Over view</b>	Production manager request necessary materials for the production from the stock
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Material should be registered</li> </ul>
<b>Flow of events</b>	<ul style="list-style-type: none"> <li>Fill the form</li> <li>Submit</li> </ul>
<b>Postcondition</b>	Stock manger receive the request note. He releases materials if available in the stock

Table 3.8: use case narrative of mark attendance

### Stock Module

Figure 3.4 shows high level Use-Case diagram of stock module.

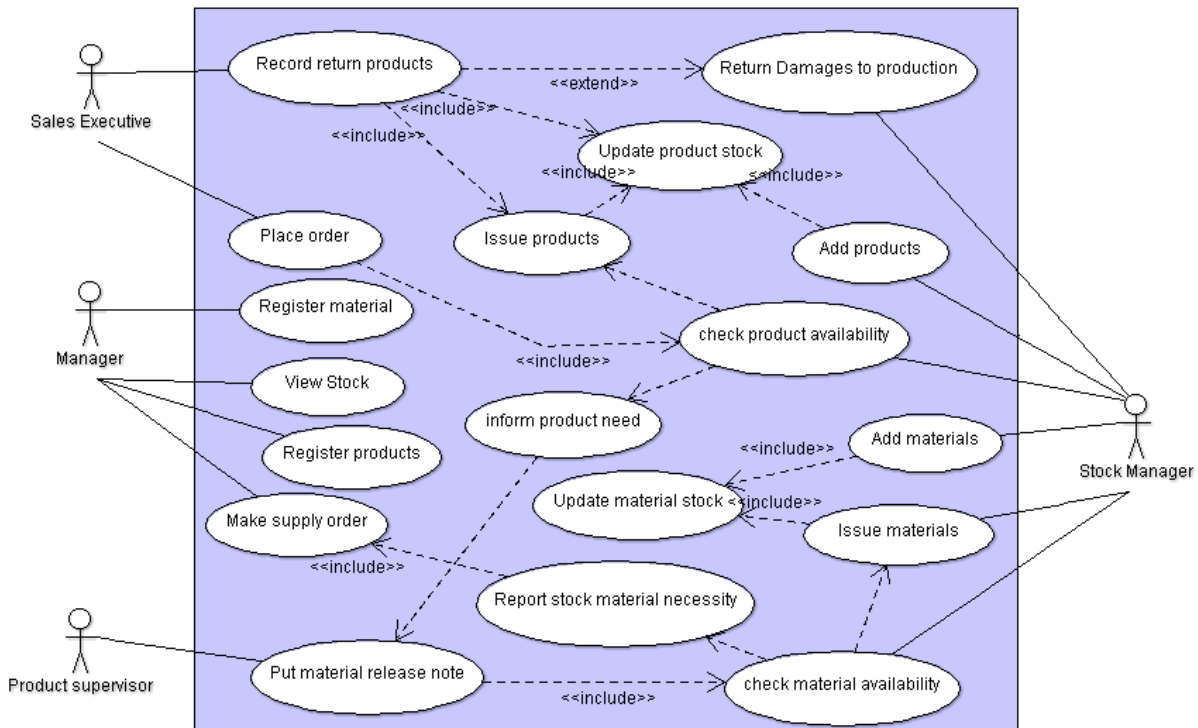


Figure 3.4 high levels Use-Case of the stock module

1. Use case description for Register material

<b>Use case</b>	Register material
<b>Actor</b>	Manager
<b>Over view</b>	
All materials should be registered in the system	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Using that material in any step of production</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form</li> <li>• save</li> </ul>	
<b>Postcondition</b>	
These information is important in production and supply	

Table 3.9: use case narrative of register material

2. Use case description for Register Product

<b>Use case</b>	Register product
<b>Actor</b>	Manager
<b>Over view</b>	
All products should be registered in the system	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Being a manufacturing product in the factory</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form</li> <li>• save</li> </ul>	
<b>Postcondition</b>	
This information is important in production and sales	

Table 3.10: use case narrative of register product

### 3. Use case description for Add material

<b>Use case</b>	Add materials
<b>Actor</b>	Stock Manager
<b>Over view</b>	
Supplied materials are added to stock	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Supply material</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form</li> <li>• save</li> </ul>	
<b>Postcondition</b>	
Material stock should be updated. These information is important in production, stock and supply	

Table 3.11: use case narrative of add materials

### 4. Use case description for Add Product

<b>Use case</b>	Add products
<b>Actor</b>	Stock Manager
<b>Over view</b>	
Production is completed they are added to the stock	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Complete production of the day</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form</li> <li>• save</li> </ul>	
<b>Post condition</b>	
Product stock should be updated. This information is important in sales and stock	

Table 3.12: use case narrative of add product



5. Use case description for Issue material

<b>Use case</b>	Issue materials
<b>Actor</b>	Stock Manager
<b>Over view</b>	
Materials are issued for the request from product supervisor	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Product supervisor put material release note</li> <li>• Stock Manager check material availability</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select material release note</li> <li>• Fill the form and <b>save</b></li> </ul>	
<b>Postcondition</b>	
Material stock should be updated. If material is not available stock material necessity is informed to manager. Manager sends supply order note to supplier as an email. These information is important in stock and supply	

Table 3.13: use case narrative of issue materials

6. Use case description for Issue Product

<b>Use case</b>	Issue products
<b>Actor</b>	Stock Manager
<b>Over view</b>	
Products are issued for the request from sales executive for orders and returns.	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Sales executive place order or record return products</li> <li>• Stock Manager check product availability</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select placed order or return product record</li> <li>• Fill the form and save</li> </ul>	
<b>Postcondition</b>	
Product stock should be updated. Repairable returns should be sent to production. If product is not available product need is informed to production supervisor. This information is important in sales and stock	

Table 3.14: use case narrative of issue product

## Supply Module

Figure 3.5 shows high level Use-Case diagram of supply module.

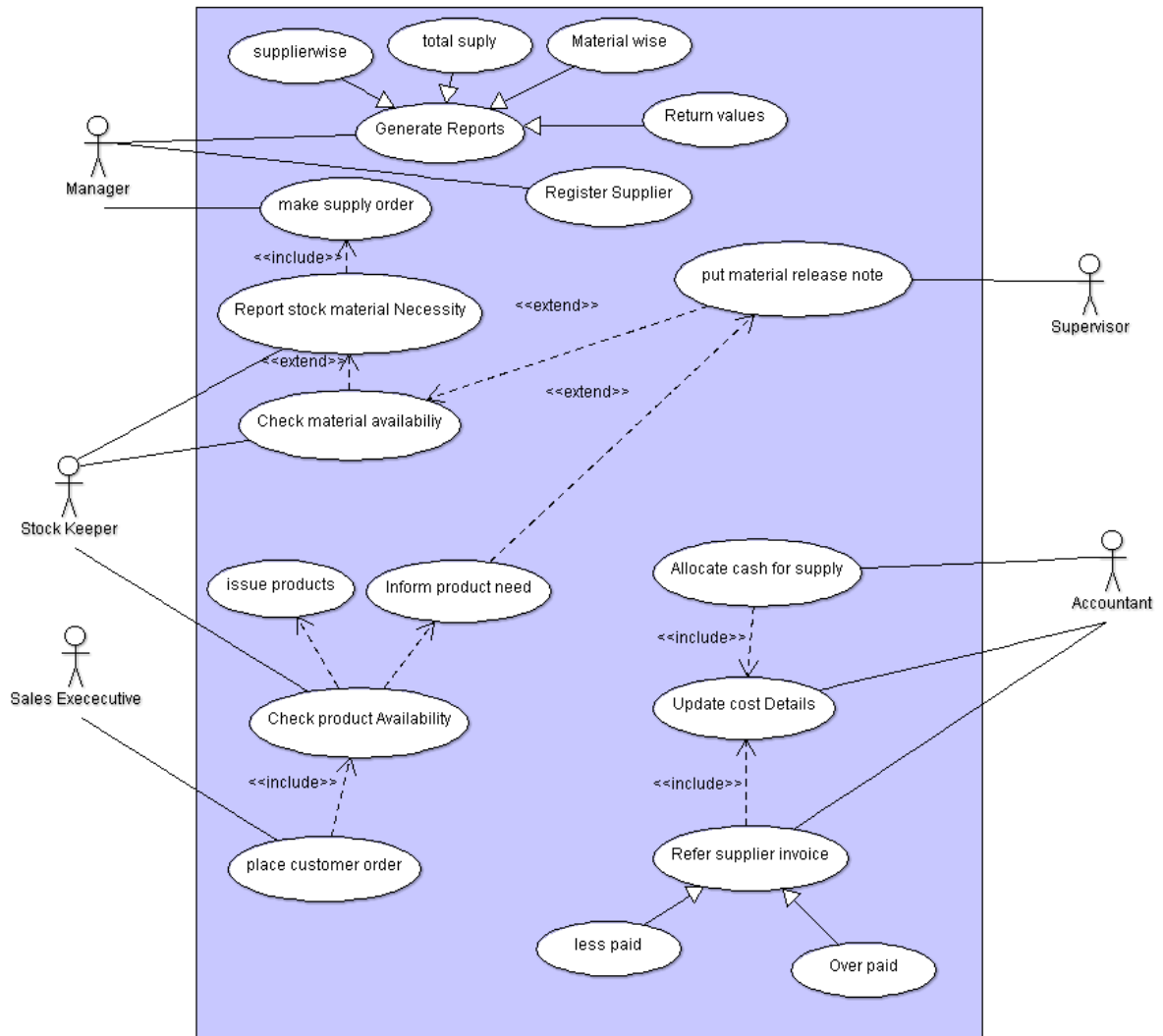


Figure 3.5 high level Use-Case of the supply module

1. Use case description for make supply order

<b>Use case</b>	Make supply order
<b>Actor</b>	Manager
<b>Over view</b>	
When stock manager informs about material necessity manager make a supply order note and send it to supplier via an email	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Stock manger inform about material necessity</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select material necessity note</li> <li>• Fill the form</li> <li>• Save and email</li> </ul>	
<b>Postcondition</b>	
This information is important in accounts and transport	

Table 3.15: use case narrative of make supply order

2. Use case description for refer supplier invoice

<b>Use case</b>	Refer supplier invoice
<b>Actor</b>	Accountant
<b>Over view</b>	
When supply order is received with invoice, accountant enter invoice details	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Make supply order by manager</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Receive supply order</li> <li>• Select supply order note</li> <li>• Fill the form</li> <li>• Save</li> </ul>	
<b>Postcondition</b>	
This information should be accurate they are used in accounts update and report generation	

Table 3.16: use case narrative of make supply order

### 3.5.2 Class Diagram

The class diagram describes the attributes and operations of classes. Figure 3.6 shows class diagram of PADMS

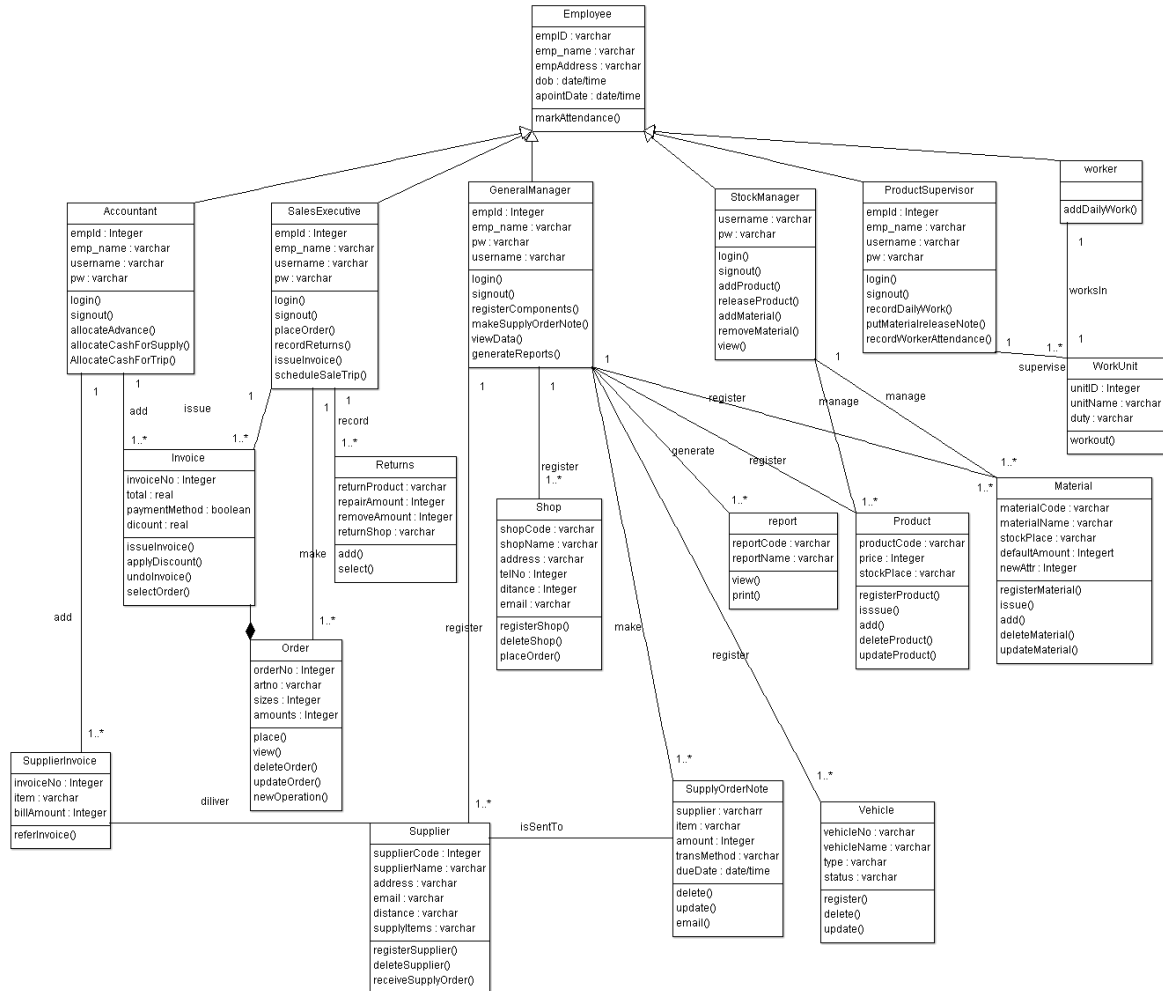
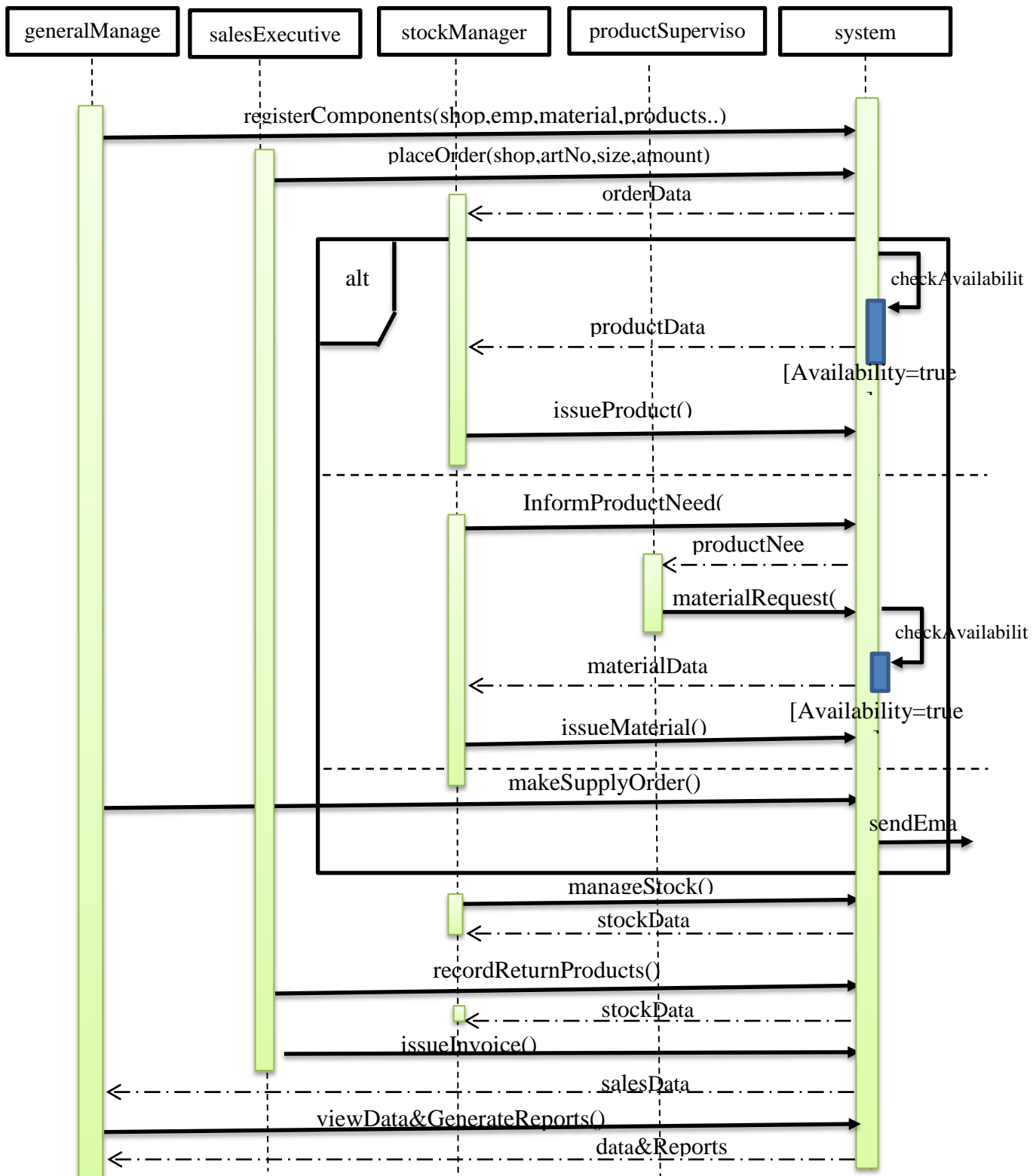


Figure 3.6 Class Diagram of the Proposed System

### 3.5.3 Sequence Diagram



**Figure 3.7 Sequence diagram of the proposed system**

Figure 3.7 shows sequence diagram of the proposed PADMS.

## 3.6 Database Design of the system

### 3.6.1 Database diagram

Here describes database design use for the system. This is a technique used for defining business requirement for a database. Figure 3.8 depicts the structure of the system's relational composition.

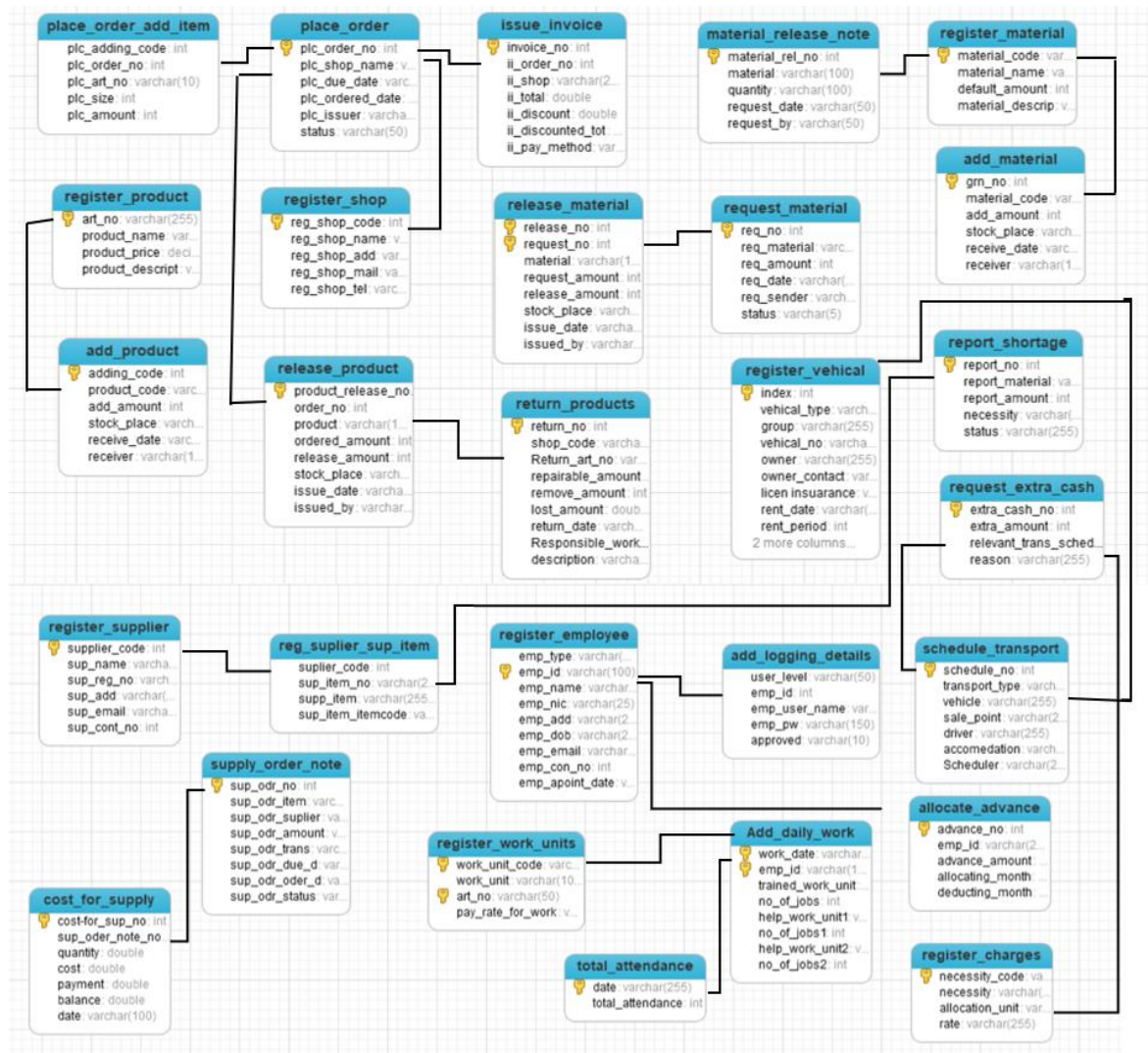


Figure 3.8 Database diagram

### 3.6.2 ER diagram

Figure 3.9 shows ER (Entity Relationship) diagram of the proposed PADMS.

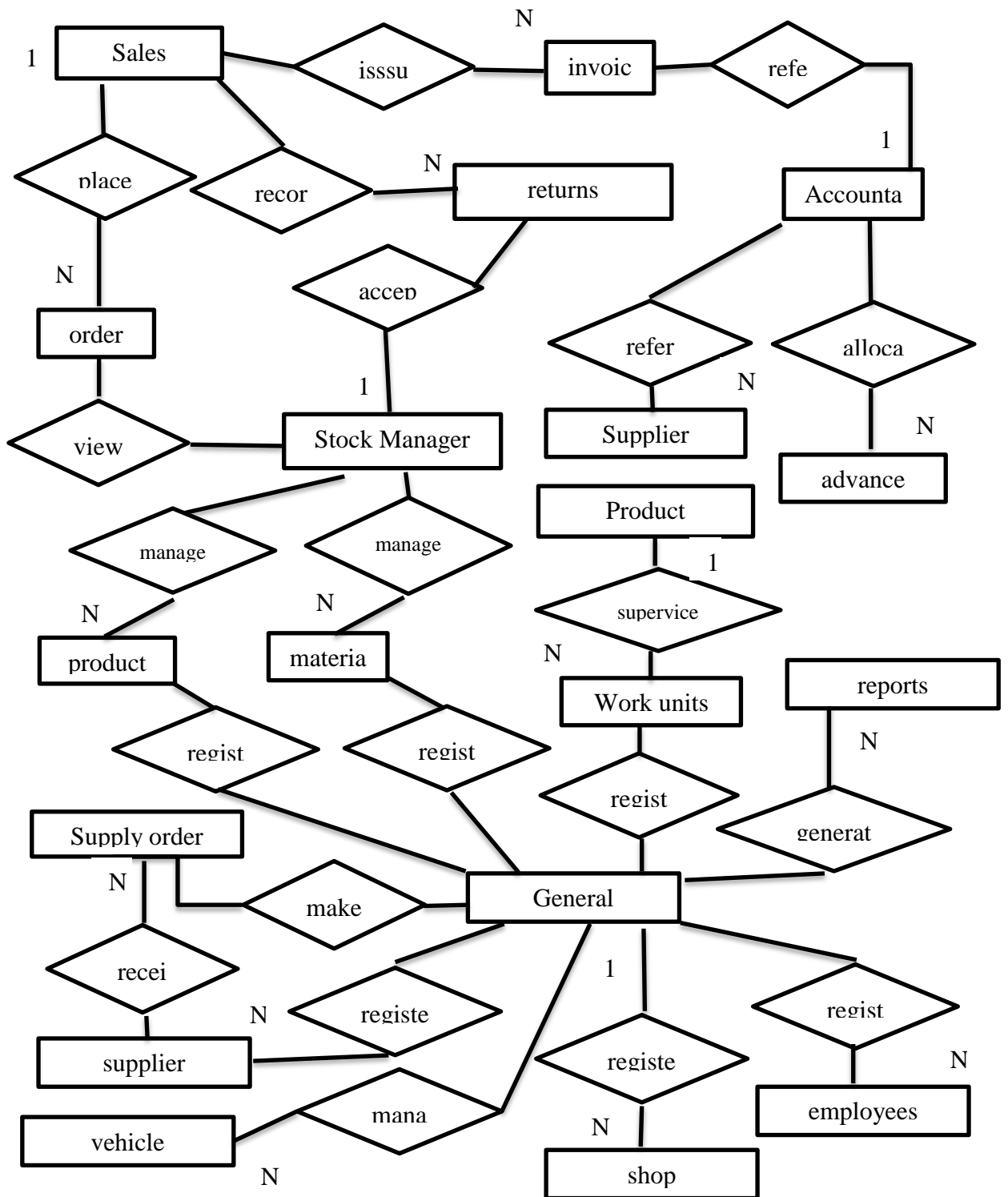


Figure 3.9 ER diagram

## 3.7 User Interface Design

User interface design is a most important task and same way not that much easy to design with relevant characteristics. User interface connect the user with the system. Through these user interfaces users can access data ad complete task easily without worrying internal processes, further user interface hides internal complex logic from user and provides user friendly attractive front for its users.

### **Human factors in interface design**

- Limited short-term memory

People can instantaneously remember about seven items of information. If you present more than this, they are more liable to make mistakes.

- People make mistakes

When people make mistakes and systems go wrong, inappropriate alarms and messages can increase stress and hence the likelihood of more mistakes

- People are different

People have a wide range of physical capabilities. Designers should not just design for their own capabilities.

- People have different interaction preferences **Some like pictures, some like text**

### **User interface design principles**

- User familiarity

The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system

- Consistency

The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way

- Minimal surprise

Users should never be surprised by the behavior of a system.

- Recoverability

The interface should include mechanisms to allow users to recover from errors

- User guidance

The interface should provide meaningful feedback when errors occur and provide help facility

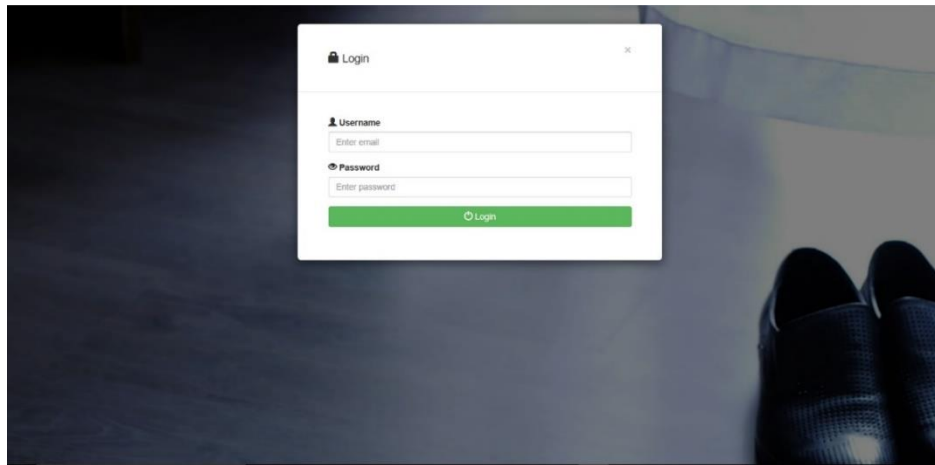


- User Diversity

The interface should provide appropriate interaction facilities for different type of user[13]

### 3.7.1 Login Interface

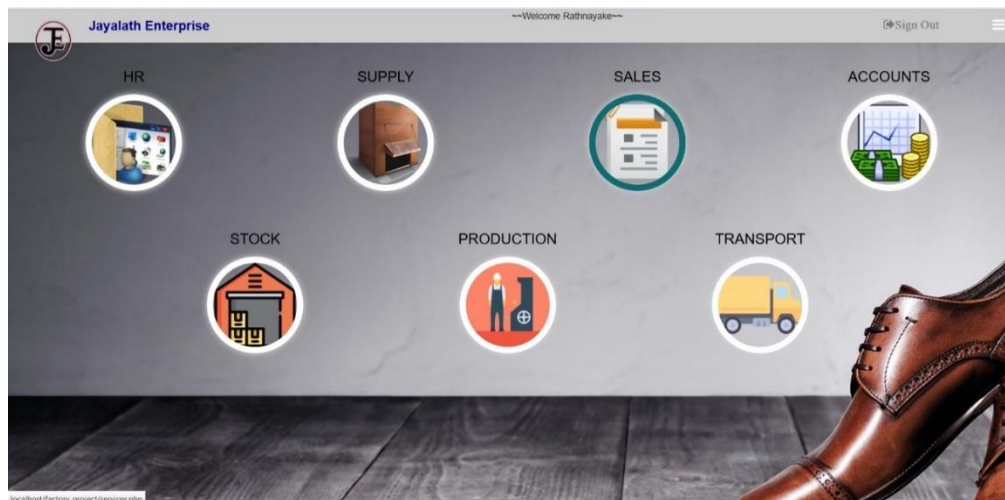
The system login page which belongs to the developed system is displayed by figure 3.10. System user meets the Logging interface first. So, it should be a pleasant impression to user and should be simple.



**Figure 3.10 Login Interface**

### 3.7.2 Dash board

The system dash board of manager which belongs to the developed system is displayed by figure 3.11. dash board allows user to link with main modules of the system. Dash board differ to user according to their privileges.



**Figure 3.11 Dash board**

### 3.7.3 Register Employee Interface

Register Employee interface which belongs to the developed system is displayed by figure 3.12

No.	Employee ID	Type	Name	NIC
1	0001	Manager	rathnayaka	543756987V
2	0002	Supervisor	indra	567832684V
3	0003	Accountant	darshika	865972686V
4	0004	Stock Manager	poorna	913679642V

Figure 3.12 Register Employee Interface

### 3.7.4 Release material interface

Remove material interface which belongs to the developed system is displayed by figure 3.13. This interface is an opened tab of manage material stock interface. Add material and view material stock tabs also can be seen here.

Select	Req.No	Material	Amount	Date/Time	Requested By
<input type="checkbox"/>	2	fffff	100	19/10/2017	sathyajith
<input type="checkbox"/>	4	laese	100	19/10/2017	sathyajith
<input type="checkbox"/>	5	00000	78	21/10/2017	sathyajith

Ret_No.	Material	Amount	Release Date	Release By
<input type="checkbox"/>	JEM2	50	25/10/2017	poorna
<input type="checkbox"/>	JEM1	50	22/10/2017	poorna

Figure 3.13 Release materials Interface

### 3.7.5 Supply order note Interface

Supply order note interface which belongs to the developed system is displayed by figure

Pending Order Data *						
	No.	Item	Supplier	Amount	Ordered date	Status
	18	laese	nimal	300	15/10/2017	
	19	ggggg	sunil	30	15/10/2017	
	21	laese	nimal	775	17/10/2017	

Figure 3.14 Supply Order note Interface

### 3.7.6 Place Order interface

Place order interface which belongs to the developed system is displayed by figure 3.15.

Order Data *					
	No.	Shop Name	Due Date	Ordered date	Ref.
	221	aaaa	27/10/2017	19/10/2017	6
	220	aaaa	01/12/2017	17/10/2017	ref 2
	218	cccc	29/11/2017	15/10/2017	ref 1
	217	cccc	09/11/2017	15/10/2017	ref 3

Figure 3.15 Place order Interface

### 3.7.7 Issue invoice interface

Issue invoice interface which belongs to the developed system is displayed by figure 3.16.

The screenshot shows the 'Issue Invoice' interface. At the top, there is a header with the 'Jayalath Enterprise' logo, the text 'Welcome Rathnayake', and a 'Sign Out' button. Below the header, the page title 'Issue Invoice' is displayed. The main content area is divided into several sections:

- Form Fields:** There are two input fields for 'Order No.' and 'Shop Name'. Below these are buttons for 'Art No.', 'Size', 'Amount', 'Unit rate', and 'total Rate'. A 'Total:' field is also present.
- Payment Method:** Radio buttons for 'cheque' and 'Cash' are available.
- Action Buttons:** 'Save' and 'Goto Print' buttons are located at the bottom of the form area.
- Select order to issue invoice:** A table with columns: Select, Index, Order No., Shop, Order Date, and ref. It contains three rows of data.
- issued invoice:** A table with columns: Undo, Invoice No, Order No., Shop, Total, and Total with Discount. It contains three rows of data.

**Figure 3.16 Issue invoice Interface**

# CHAPTER 04 – IMPLEMENTATION

## 4.1 Introduction

Implementation is the process of converting system design to a working system. Here choosing correct programming language is critical. If programming language does not support the functionality of the system, the system cannot be implemented as designed. In this phase a relevant programming language is used in a suitable way with relevant comments and it may be easier to build the proposed system in a brief time and it will be a big help for future maintenance.

## 4.2 Implementation Environment

Some important aspects were taken into consideration when finalizing the implementation environment. When selecting the development software, most of them were free and open source which won't cause much trouble when getting the copyrights of the system. Some of these technologies are targeted at a specific application domain (e.g., Web-site design and implementation); others focus on a technology domain (e.g., object-oriented systems). The following components were used in the implementation environment.

Implementation environment used for our system is given in Table 4.1 below.

Hardware	Software
Intel(R) Core(TM) i5-5200U CPU @ 2.27GHz 4GB RAM 930GB Hard Disk	Microsoft Windows 10 XAMPP V3.2.2 PHP 5.5 Apache MySQL Net beans 8.1 Navicat Premium Adobe Photoshop CS6

Table 4.1 Implementation Environment

Although the system was developed under windows 10 operating system, the PADMS is fully compatible with windows 8 as well as with windows 7.

## Development Tools

- Net beans – for coding.
- Navicat – to handle database easily cooperated with MySQL
- Adobe Photoshop 7.0 – for image retouching.
- PHP (Hypertext Pre- Processor) was the main development language used to develop the main system and its logics.
- MySQL was used to handle all the development related to the database.
- HTML was used to build the base Interfaces of the system.
- CSS was used to make the plain HTML interfaces more attractive and user friendly, which also decided the look and feel of the system.
- JavaScript was used to code all the client-side validation.
- AJAX which is based on JavaScript was used to get data from the server without refreshing it repetitively.
- JQuery - It is a JavaScript library and used as reusable component when developing.
- JSON – this syntax is used for storing and exchanging text information.
- Ms word – for PDF creation technology

## 4.3 Reused Modules

The following pre-coded modules were reused in the system during implementation process.

- Bootstrap - Bootstrap is a "sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development." [14]
- Alertify – a styling sheet class used to generate meaningful and attractive alerts.
- JavaScript SHA1 - a 160 bit encryption algorithm.
- Bootstrap-datepicker - provides a flexible datepicker widget in the Bootstrap style.

## 4.4 Major Code Segments

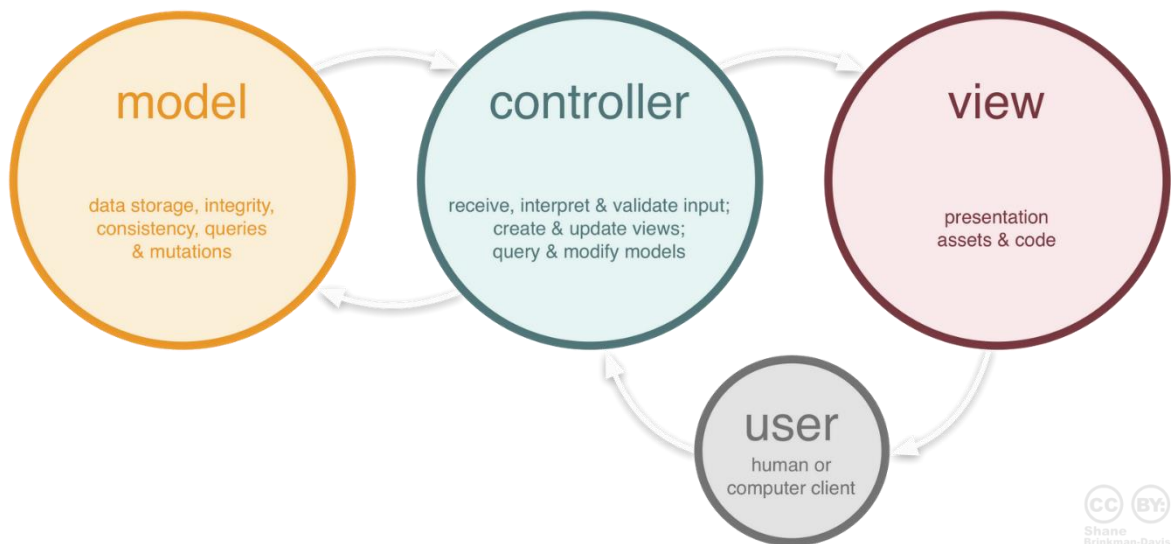
MCV structure was followed, when doing coding.

**M – MODEL** – include code segments which make connections with the database.

**C – CONTROL** – include JavaScript functions which make connection with both views and model

**V – VIEW** – include basic html which make view of the page.

The main code modules developed in the system have been mentioned below by briefly describing their functionality.



[15]

**Figure 4.1 MCV structure**

Figure 4.1 describes how MCV structure works in coding.

- **Data base connection**

```
class dbConnect {
// this function is used to connect with DB
private function connetDB() {
    $link = mysqli_connect(DB_SERVER, DB_USER, DB_PASS, DB_NAME);
    if (mysqli_connect_errno()) {
        die ("Couldn't make connection" . mysqli_connect_error() .
            ".mysqli_connect_errno() . ");
    } else {
        return $link;
    }
}
// this function is used to close the DB connection
private function closeDB($link) {
    if (isset($link)) {
        mysqli_close($link);
    }
}
}
```

This code segment is place in a different page as “connectDB.php” in our system. That file is included in every MODEL page of our system. And “DbConnect” class is used by that MODEL pages by making an object from this “DbConnect” class.

**Including connectDB.php” page** - require\_once './include/connectDB.php';

**making an object from this “DbConnect” class** - \$db = new dbConnect();

In “DbConnect” class, other common functions are placed which can be used in MODEL pages.



Ex-:

```
class dbConnect {
    // insert data to DB
    public static function setResults($query) {
        $db_conn = new dbConnect();
        $conn = $db_conn->connetDB();
        $result = mysqli_query($conn, $query);
        $last_id = mysqli_insert_id($conn);
        $db_conn->confirmQuery($result, $conn, $last_id);
        $db_conn->closeDB($conn);
    }
    // Select data from data base
    public static function getResult($query) {
        $data = array();
        $db_conn = new dbConnect();
        $conn = $db_conn->connetDB();
        $result = mysqli_query($conn, $query);

        while ($row = mysqli_fetch_assoc($result)) {
            $data[] = $row;
        }

        mysqli_free_result($result);
        $db_conn->closeDB($conn);
        echo json_encode($data);
    }
    // Delete data from data base
    public static function getResultforDelete($query) {

        $db_conn = new dbConnect();
        $conn = $db_conn->connetDB();
        $result = mysqli_query($conn, $query);

        $db_conn->confirmQuery1($result, $conn);
        $db_conn->closeDB($conn);
    }
}
```

Above code segments are used to insert data to DB, select data from DB and to delete data from BD. These code segments can be used again and again by various set of data.

- **Validations of the System**

Validation codes used in the system is presented below using “register employee” page. Following given validation is used in VIEW page

```

<script type="text/javascript">

// insert only Number validation

function isNumberKey(evt) {
    var charCode = (evt.which) ? evt.which : evt.keyCode;
    if (charCode == 110 || charCode == 190 || charCode == 46)
        return true;
    if (charCode > 31 && (charCode < 48 || charCode > 57))
        return false;
    return true;
}

// NIC validation
$('#reg_emp_nic').on('keyup', function () {
    if ($(this).val() !== "") {
        var valid = /^[0-9]{9}[VvXx]{1}$/.test(this.value) && this.value.length;
        if (valid) {
            $('#nic_valok').html('<i class="glyphicon glyphicon-ok-sign"></i>
Valid NIC number. ');
            $('#nic_val').html("");
            $('#reg_emp_save').removeClass('hidden');
        } else {
            $('#nic_valok').html("");
            $('#nic_val').html('<i class="glyphicon glyphicon-warning-sign"></i>
NIC number is not Valid. ');
            $('#reg_emp_save').addClass('hidden');
        }
    } else {
        $('#nic_val').html("Please Enter NIC");
        $('#reg_emp_save').addClass('hidden');
    }
});
</script>

```

‘Only number validation’ is used when a form field should be filled only with number data. ‘NIC validation’ is used when a form field require the date which follow NIC character format.

```

<script type="text/javascript">
// Email Validation
$('#reg_emp_email').on('keyup', function () {
    if ($(this).val() !== "") {
        var valid = /^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}$/;
        if (valid.test(this.value) && this.value.length) {
            if (valid) {
                $('.emailvalue').removeClass('has-error');
                $('#em_val').html("");
                $('#reg_emp_save').removeClass('hidden');
            } else {
                $('.emailvalue').addClass('has-error');
                $('#em_val').html('<i class="glyphicon glyphicon-warning-sign"></i> E-Mail address is not valid. ');
                $('#reg_emp_save').addClass('hidden');
            }
        } else {
            $('.emailvalue').removeClass('has-error');
            $('#em_val').html("");
            $('#reg_emp_save').removeClass('hidden');
        }
    }
});
</script>

```

Above email validation is used to help system users to only enter correct email address in email field.

- **MCV structure in main code segments**

Using the “Register Employee” page in our system following main segments are presented which follow MCV model all over the related process. In This examples whole code of interface design is not presented within view code segments. Only essential codes to describe is given below.

- **Save to System**

#### VIEW

```

<script type="text/javascript">
    $('#reg_emp_save').click(function () {
        reg_employee();
    });
</script>

```

When ‘save’ button is clicked ‘reg\_employee()’ function is called. And that function is called from ‘CONTROL’ section , shown as below.

## CONTROL

```
function reg_employee() {

    var emp_type = $('#reg_emp_type').val();
    var emp_id = $('#reg_emp_id').val();
    var emp_name = $('#reg_emp_name').val();
    var emp_nic = $('#reg_emp_nic').val();
    var emp_add = $('#reg_emp_addres').val();
    var emp_dob = $('#reg_emp_dob').val();
    var emp_email = $('#reg_emp_email').val();
    var emp_cont = $('#reg_emp_tell').val();
    var emp_date = $('#reg_emp_date').val();

    $.post("./model/regis_emp_d.php",
        {action_key: "insert", key1: emp_type, key2: emp_id, key3: emp_name, key4:
emp_nic, key5: emp_add, key6: emp_dob, key7: emp_email, key8: emp_cont, key9:
emp_date},
        function (return_val) {
            $.each(return_val, function (index, msgDataAny) {
                if (msgDataAny.msgType === 1) {
                    alertify.success("Employee Registered");
                } else {
                    alertify.error("error");
                }
            });
            tableload_regis_emp();
            clear_reg_emp_form_data();
        }, "json");
}
```

When 'control' code segment run, form data is passed to php file at the path `./model/regis_emp_d.php`. In that file in 'MODEL' section, relevant SQL query is run shown as below.

## MODEL

```
if (array_key_exists("action_key", $_POST)) {
    if ($_POST['action_key'] == 'insert') {

        $query = "INSERT INTO `factory_db`.`register_employee` (`emp_type`, `emp_id`,
`emp_name`, `emp_nic`, `emp_add`, `emp_dob`, `emp_email`,
`emp_con_no`, `emp_apoint_date`) VALUES
({$_POST['key1']}, {$_POST['key2']}, {$_POST['key3']}, {$_POST['key4']}, {$_POST['ke
y5']}, {$_POST['key6']}, {$_POST['key7']}, {$_POST['key8']}, {$_POST['key9']}";

        echo $db->setResults($query);
    }
}
```

- **Delete**

## **VIEW and CONTROL**

```

$('.del_emp').click(function () {
    var delete_emp_ID = $(this).val();
    alertify.confirm("Are you sure want to detele this record?", function (e) {
        if (e) {
            $.post("./model/regis_emp_d.php", {action_key: 'delete', del_employee_id:
delete_emp_ID}, function (e) {
                tableload_regis_emp();
                alertifyMsgDisplay(e, 2000);
            }, "json");
        } else {
            alertify.log("Cancel process", "Done", 1000);
        }
    });
});

```

## **MODEL**

```

if (array_key_exists("action_key", $_POST)) {
    if ($_POST['action_key'] === 'delete') {
        $id = $_POST['del_employee_id'];
        $query = "DELETE FROM `register_employee` WHERE `emp_id` = " . $id;
        echo $db->getResultforDelete($query);
    }
}

```

In Appendix F other code segments (update, Load data to a table from DB, load data to a combo box from DB, Using Modal in the system are given.

# CHAPTER 05 – EVALUATION

## 5.1 Introduction

“Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.”

[16]

System testing is one element of a broader topic that is often referred to as verification and validation (V&V). Verification refers to the set of tasks that ensure that system correctly implements a specific function. Validation refers to a different set of tasks that ensure that the system that has been built is traceable to customer requirements. Boehm states this way:

Verification: —Are we building the product, right?

Validation: —Are we building the right product?

## 5.2 Software Testing Methods

### **Black-Box Testing**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. [16]

### **White-Box Testing**

White-box testing is the detailed investigation of internal logic and structure of the code.

White-box testing is also called glass testing or open-box testing. In order to perform

white-box testing on an application, a tester needs to know the internal workings of the code. [16]

## 5.3 Software Testing Types

### **Unit Testing**

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

### **Integration Testing**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom up integration testing and Top-down integration testing.

Bottom-up integration -This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Top-down integration - In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.

### **System Testing**

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards.

### **Regression Testing**

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation.

### **Acceptance Testing**

This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement. [16]

## 5.4 Test Plans and Test Results

Test plan is the guide for testing the system. This covers all components in the system. Test plan is prepared during the design phase, before starting the implementation. A test plan includes test objectives, schedule and logistics, test strategies and especially test cases. Test cases are prepared as a part of test plan. Test cases consist of data, procedure and expected result, and represent just one situation under which the system or part of the system might run.

In here test cases are designed as modular wise to reduce the complexity of testing process. The following tables specify some test cases.

### 5.4.1 Test Cases of Sales Module

Test cases, expected results and actual result status of sales module are displayed in table 5.1

Test No.	Test Case	Step to Test	Expected Result	Actual Result Status
1	Place order	When load the interface	<ul style="list-style-type: none"> <li>• Table should be loaded with placed orders</li> <li>• Shop name combo box should be loaded with registered shops</li> <li>• Art no combo box should be loaded with registered product Art No s.</li> <li>• Issued by combo box should be loaded with registered Sales representatives</li> </ul>	pass
2	Place order	When click on due date text box	Calendar should be appeared	pass
3	Place order	When type text in amount test box	Text are not allowed to be typed	pass
4	Place order	When click on “Add” button in the form	Art No., size and amount should be added to the below table again and again	pass
5	Place order	Click “save” button with necessary field filled	“order sent” message should be appeared	pass
6	Place order	Click edit button in the table	Order form should be changed with buttons “delete previous items” and “add more items”. Save button is removed and update button appeared	pass



7	Place order	Click on new “delete previous items” button	Modal should be appeared having saved items where allow user to delete that items	pass
8	Place order	Click on “add more items” button	Art No., size and amount fields are appeared with “add” button	pass
9	Place order	Click “Update” button	“Successfully Updated” alert should be appeared	pass
10	Issue Invoice	When load the interface	Two tables should be loaded with relevant data	pass
11	Issue Invoice	When click select button in “Select order to issue invoice” table	<ul style="list-style-type: none"> <li>• Left side form should be filled.</li> <li>• Total should be calculated and displayed</li> <li>• Displayed data can not be edited</li> </ul>	pass
12	Issue Invoice	When click “cash” option	Discount and discounted total should be calculated and displayed	pass
13	Issue Invoice	When click “cheque” option	Discount and discounted total should be removed	pass
14	Issue Invoice	When click “save” button	<ul style="list-style-type: none"> <li>• “Invoice is issued” alert should be appeared</li> <li>• Relevant record should be disappeared from “Select order to issue invoice” table</li> <li>• Issued record should be appeared in “Issued invoice” table</li> </ul>	pass
15	Issue Invoice	When click on “undo” button in “Issued invoice” table	<ul style="list-style-type: none"> <li>• Relevant record should be disappeared from “Issued invoice” table and should be appeared again in “Select order to issue invoice” table</li> </ul>	pass
16	Issue Invoice	When click on “Go to Print” button	<ul style="list-style-type: none"> <li>• User should be directed to “invoice.php” page</li> </ul>	pass

Table 5.1: Test cases and results of sales module

### 5.4.2 Test Cases of Product Module

Test cases, expected results and actual results status of sales module are displayed in table 5.2

Test No.	Test Case	Step to Test	Expected Result	Actual Result Status
1	Mark attendance	When load the interface	• Table should be loaded with marked attendance before	pass
2	Mark attendance	When click on marking date text box	Calendar should be appeared	pass
3	Mark attendance	When click on “Employee list” button	Employee list table should be appeared with green buttons	pass
4	Mark attendance	Click green button	Green button should be converted in to green	pass
5	Mark attendance	Click on converted red button in the table	red button should be converted in to green	pass
6	Mark attendance	Click on “record attendance” button	“Attendance is recorded” alert is appeared.	pass
7	Add daily work	When click on working date text box	Calendar should be appeared	pass
8	Add daily work	When click on “Select Employee” combo box and select a worker	Relevant trained unit of worker is displayed in “helping work unit” text box	pass
09	Add daily work	When click on “helping work unit” combo box	List of all work units are displayed	pass
10	Add daily work	When type text in “No. of jobs” test box	Text are not allowed to be typed	pass
11	Add daily work	When click on “add” button	Record is appeared in below table	pass
12	Add daily work	When click “save” button	“Daily work added” alert should be appeared.	pass
13	Material Release note	When type text in “quantity” test box	Text are not allowed to be typed	pass
14	Material Release note	When click on “material” combo box	All registered materials are displayed.	

Table 5.2: Test cases and results of product module

### 5.4.3 Test Cases of Stock Module

Test cases, expected results and actual results status of Stock module are displayed in table 5.3

Test No.	Test Case	Step to Test	Expected Result	Actual Result Status
1	Register material	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with registered materials before</li> </ul>	pass
2	Register material	When click “save” button	“Material is registered” alert should be appeared.	pass
3	Register material	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
4	Register material	When click on a “update” button in table	Left side form is filled with Data relevant to that record	pass
5	Register product	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with registered products before</li> </ul>	pass
6	Register product	When click “save” button	“product is registered” alert should be appeared.	pass
7	Register product	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
8	Add material	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with added materials before</li> </ul>	pass
9	Add material	When click on “material” combo box	All registered materials are displayed.	pass
10	Add material	When click on “receiver” combo box	Authenticated employees are displayed.	pass
11	Add material	When type text in “Adding amount” test box	Text are not allowed to be typed	pass
12	Add material	When click “save” button	“Material is added” alert should be appeared.	pass
11	Release material	When load the interface	Two tables should be loaded with relevant data	pass
12	Release material	When click on “Issued by” combo box	Authenticated employees are displayed.	pass
13	Release material	When click select button in Select Request from Production Department table	<ul style="list-style-type: none"> <li>Left side form should be filled partially</li> </ul>	pass
14	Release material	When click “save” button	<ul style="list-style-type: none"> <li>“Material is released” alert should be appeared.</li> <li>Record in “Select Request from Production Department” table is disappeared</li> <li>Relevant record is loaded in</li> </ul>	pass

			“released material” table	
15	Release material	When click undo button in “released material” table	<ul style="list-style-type: none"> <li>Record in “released material” table is disappeared</li> <li>Relevant record is appeared in Select Request from Production Department” table</li> </ul>	pass
16	View Stock	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with materials with amounts in the stock</li> </ul>	pass
17	Add product	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with added product before</li> </ul>	pass
18	Add product	When click on “product” combo box	All registered product are displayed.	pass
19	Add product	When click on “receiver” combo box	Authenticated employees are displayed.	pass
29	Add product	When type text in “Adding amount” test box	Text are not allowed to be typed	pass
21	Add product	When click “save” button	“product is added” alert should be appeared.	pass
22	Release product	When load the interface	Two tables should be loaded with relevant data	pass
23	Release product	When click on “Issued by” combo box	Authenticated employees are displayed.	pass
24	Release product	When click select button in “Select orders” table	<ul style="list-style-type: none"> <li>Left side form should be filled partially</li> </ul>	pass
25	Release product	When click “save” button	<ul style="list-style-type: none"> <li>“product is released” alert should be appeared.</li> <li>Record in “Select orders” table is disappeared</li> <li>Relevant record is loaded in “released product” table</li> </ul>	pass
26	product	When click undo button in “released product” table	<ul style="list-style-type: none"> <li>Record in “release product” table is disappeared</li> <li>Relevant record is appeared in “Select orders” table</li> </ul>	
27	View Stock	When load the interface	<ul style="list-style-type: none"> <li>Table should be loaded with products with amounts in the stock</li> </ul>	pass

Table 5.3: Test cases and results of stock module

In Appendix E, Test cases of other modules (Supply, HR, transport) are given.

## **5.5 User Evaluation (Acceptance testing)**

A one of another important test type in a system is acceptance test. The acceptance test of this system is done with real environment that system work with real data and real system users. Here after implementing the system next most important thing is acceptance testing. After implement the system in users' environment the system is tested by selected set of users with real time operations. By the test is carrying out users' activities are minted. This help to further system tuning in future maintenance. The final result of the test indicated that the system is easy to handle and work with, user friendly with pleasant working environment.

In addition, it can be used to perform transactions in an efficient way. But it was few problems are encountered during the work, but all the bugs were fixed right away. Some feedbacks and suggestions given by the client were implemented to give better performance and acceptance. Finally, the overall achievement of this project was satisfied and considered that all the objectives of the project have been met.

Following questionnaire form was presented to potential users of the system including Managers, Sales Executive, Stock managers, product supervisors and accountants. All together 15 users were used in getting feedback. And Helpful feedback was received.

Presented questionnaire and a set of results are shown in figure 5.1below.

Client mentioned that with the newly developed system the company could function efficiently and smoothly rather than continue using the old method. The certificate received from the client has been appended to Appendix G.

## User Evaluation questionnaire

New Production And Distribution Management System – Jayalath Enterprise

User Name: <i>Jasanthra Subasinghe</i>		User Role: <i>Manager</i>			
No	Inquiry	P	Q	R	S
1	Ease of understanding the module functions	✓			
2	Interactivity of the system	✓			
3	Coverage of required functionalities		✓		
4	Consistency of navigation menus and forms	✓			
5	Accessing to data through correct links	✓			
6	Speed of transaction per minute		✓		
7	Ability to understand error messages	✓			
8	Potential benefits gained through the system		✓		
9	Ability to maintain data up to date		✓		
10	Ease of handle form data and manipulate them	✓			

- ✓ P – very good
- ✓ Q – good
- ✓ R – acceptable
- ✓ S – weak

**Figure 5.1 Feedback Questionnaire**

# CHAPTER 06 – CONCLUSION

## 6.1 Introduction

Jayalath Enterprise is a private business factory which is continuing about two decade in Nakkawththa , where big competition takes place with Branded and famous shoe products in Sri lanka. But it could step forward significantly, in a short period of time under the vision of High quality products. Under this situation factory management take many actions to deliver an efficient and effective product while gaining a better income. So the management had to concentrate on managing information in manufacturing and distribution basically. They are achieved manually before which are not very much effective. This Online PADMS was developed to address those drawbacks as well as to optimize their overall business process and through this to help them to income while reduce cost.

## 6.2 Future Improvements

The online Production and Distribution Management System has fulfilled the entire functional and non-functional requirements requested by the client; along with additional features, and have managed to overcome most of the problems the factory faced with the manual system, some future enhancements are essential to make this as a fully qualified Management Information System (MIS). The following are the future enhancements which are considered to implement in PADMS with the client's permission.

- ❖ Improve the security by placing a good firewall and use cryptographic methods to constrict further security as the Intranet deals with large amount of sensitive information.
- ❖ Provide outside users like registered customers and registered suppliers to use the system and access some limited information relevant to their functions
- ❖ Add Bag production to the system which is a rare business unit of company.
- ❖ Improve the system to mark attendance through Bar code reader or finger print scanner
- ❖ Sending a SMS informing about supply order to supplier instead of e amil

- ❖ Add more effective bar charts and pie charts in report generation module to support top management in decision making.
- ❖ Collect some additional features to make the system more attractive.

## **6.3 Lesson Learnt**

The knowledge gained throughout the project was really valuable. In addition this project gave me an exceptional experience of being involved in a complete Software Development Lifecycle Cycle, starting from the feasibility studies to the conclusion of the project. This project gave me a chance to test and implement most important theories and technologies learnt throughout the BIT degree program. It also helped me to find out and learn some very interesting new and upcoming technologies (AJAX, jQuery, PHP, MySQL) and theories (OO) in order to enhance the system performance.

Furthermore, working on the project helped me to improve technical skills as well as intellectual skills by collaborating with many individuals from collective fields. And, through completing project dissertation , knowledge was gained about how to write reports according to the recognized standard .



## Reference

- [1] Wikipedia, Systems\_analysis, 2017. [Online]. Available:[https://en.wikipedia.org/wiki/Systems\\_analysis](https://en.wikipedia.org/wiki/Systems_analysis). [Accessed: 10-June-2017]
- [2] uplandsoftware.com,Ultriva’s Lean Factory Management (LFM), 2017.[Online]. Available: <https://uplandsoftware.com/ultriva/product/features/lean-factory-management/>. [Accessed: 20-June-2107]
- [3] classicsoftech. com,Classic Factory Management System with ERP, 2017. [Online]. Available:[http:// www.classicsoftech.com / factory\\_management\\_system.html](http://www.classicsoftech.com/factory_management_system.html) [Accessed: 20-June-2107]
- [4] reqtest.com, functional-vs-non-functional-requirements, 2017. [Online]. Available: [http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements /](http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/) .[Accessed: 20-Jul-2107]
- [5] reqtest.com, functional-vs-non-functional-requirements, 2017. [Online]. Available: [http://reqtest.com/requirements-blog/functional-vs-non-functional- requirements/](http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/) .[Accessed: 20-Jul-2107]
- [6] itinfo.am, Software Development Methodologies, 2017. [Online]. Available: <http://www.itinfo.am/eng/software-development-methodologies/>. [Accessed: 3-Aug-2017]
- [7] Wikipedia, Waterfall\_model, 2017. [Online]. Available:[https://en.wikipedia.org/wiki/ Waterfall\\_model](https://en.wikipedia.org/wiki/Waterfall_model). [Accessed: 3-Aug-2017]
- [8] techtarget.com, Prototyping Model, 2017. [Online]. Available: [http://searchcio.techtarget.com /definition/Prototyping-Model](http://searchcio.techtarget.com/definition/Prototyping-Model). [Accessed: 3-Aug-2017]
- [9] Wikipedia, Rapid application development, 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Rapid\\_application\\_development](https://en.wikipedia.org/wiki/Rapid_application_development) . [Accessed: 3-Aug-2017]
- [10] itinfo.am, Software Development Methodologies, Rapid Application Development (RAD) Methodology,2017. [Online]. Available: [http://www.itinfo.am/eng/ software-development-methodologies/](http://www.itinfo.am/eng/software-development-methodologies/). [Accessed: 3-Aug-2017]
- [11] Wikipedia, Rational Unified Process, 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Rational\\_Unified\\_Process](https://en.wikipedia.org/wiki/Rational_Unified_Process). [Accessed: 04-Aug-2017]

- [12] Wikipedia, Systems\_design, 2017. [Online]. Available:[https:// en.wikipedia.org/wiki/Systems\\_design](https://en.wikipedia.org/wiki/Systems_design). [Accessed: 10-Aug-2017]
- [13] ifs.host.cs.st-andrews.ac.uk, Books, 2017. [Online]. Available: <https://ifs.host.cs.st-andrews.ac.uk/Books/SE7/Presentations/PDF/ch16.pdf>. [Accessed: 20-Aug-2017]
- [14] startbootstrap.com, help,2017. . [Online]. Available:<http://startbootstrap.com/help/>. [Accessed: 10-sep-2017]
- [15] waqar.me, mvc-model-view-controller, 2017. [Online]. Available: <http://www.waqar.me/learn/mvc-model-view-controller/> [Accessed: 30-sep-2017]
- [16] tutorialspoint.com, software\_testing\_overview, 2017. [Online]. Available: [http://www.tutorialspoint.com/software\\_testing/software\\_testing\\_overview.htm](http://www.tutorialspoint.com/software_testing/software_testing_overview.htm) . [Accessed: 30- sep -2017]

# Appendix A - System Documentation

System documentation provides guidelines prior to the setup of the PADMS. This is to assist the software engineers, Administrators and managers to install the PADMS in their PC's, Servers or Workstations. The system documentation can be referred if there are any changes to be made in the PADMS.

In order to install the system, the Device chosen for installation should meet the following prerequisites of Hardware and Software.

## Hardware requirements

Hardware	Minimum requirement
Processor	1.8GHz Intel core 2 Duo or similar or newer processor
Memory	1GB RAM or more
Hard Disk Space	Minimum 1GB free disk space or Higher
Display	1024x768 or resolutions above High. Color 16bit display
Printer	Dot-matrix printer or Ink jet printer or Laser printer
Internet	Minimum 2.1Mbps ADSL connection

## Software requirements

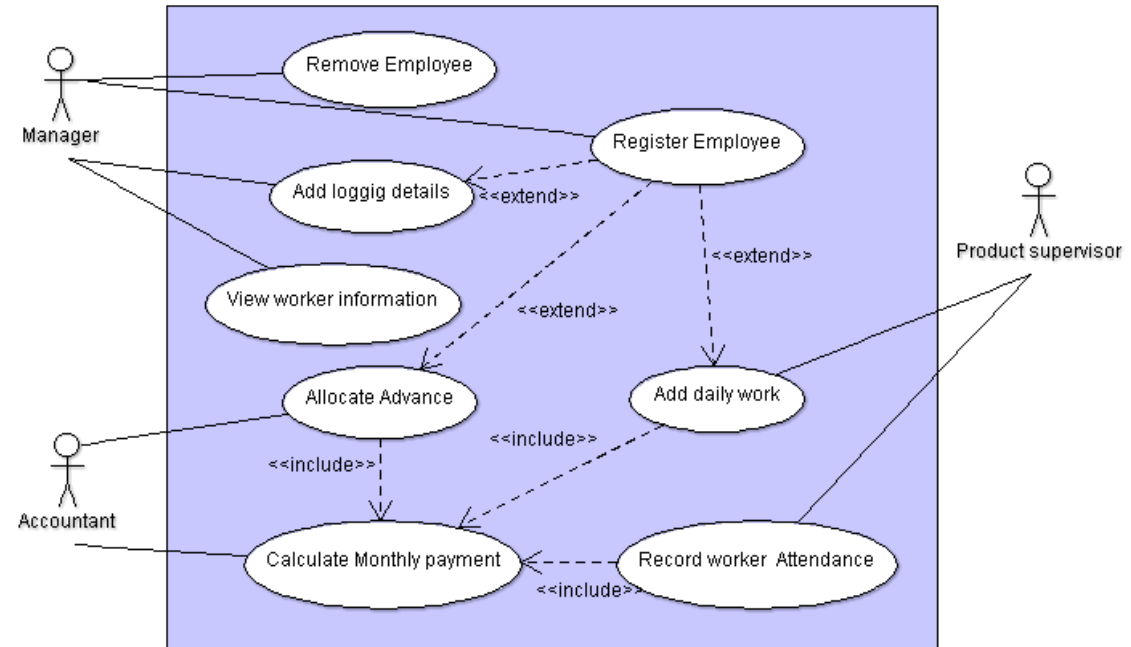
Software	Minimum requirement
Operating System	Microsoft Windows XP/Vista/Windows 7/Windows 8
Bundle package	XAMPP Or Individual packages MySQL5.5 ,PHP 5.4, Apache2.2
Code editor	Net Beans or suitable editor
Image editor	Adobe Photoshop CS3 version or higher version
Web browser	Internet explorer/Fire Fox/Google Chrome/Opera/Safari

## PADMS Setup

1. Install relevant software packages according to their user manuals.
  - ❖ XAMPP
  - ❖ Web browser
2. Copy the “padms” folder given in the supplementary CD and paste it inside the htdocs folder in the following paths
  - ❖ Windows Environment with XAMPP installed the path would be - C:\xampp\htdocs
  - ❖ Linux Environment with XAMPP installed the path would be - /opt/lampp/htdocs
3. Data base installation
  - ❖ Open the web browser and type the URL <http://localhost/phpmyadmin/>
  - ❖ Create empty database by providing name as “factory\_db” and navigate to the “Import” tab and click “choose file” button. Then browse the CD and select the “factory\_db.sql” file by opening Database folder.
  - ❖ Then Press “GO” button located in the bottom of the page.
4. Launching the system
  - ❖ You can open preferred web browser and type the following URL in the address bar: <http://localhost/padms> or <http://127.0.0.1/padms>
  - ❖ And Login to gain access, by providing correct username and password

# Appendix B - Design Documentation

## HR Module



**Figure B.1 high level Use-Case of the HR module**

1. Use case description for Register employee

<b>Use case</b>	Register employee
<b>Actor</b>	Manager
<b>Over view</b>	All employees including mangers, supervisors, sales executives and skilled workers should be registered in the system
<b>Precondition</b>	<ul style="list-style-type: none"> <li>They should be permanent employees of the factory</li> </ul>
<b>Flow of events</b>	<ul style="list-style-type: none"> <li>Fill the form and save</li> </ul>
<b>Postcondition</b>	This information is used in adding logging details, mark attendance, adding daily work, allocating advance and finally calculating salary. Registered employees can be removed or updated

Table B.1: use case narrative of register employee

## 2. Use case description for Add logging details

<b>Use case</b>	Add logging details
<b>Actor</b>	Manager
<b>Over view</b>	Employees including mangers, supervisors, sales executives who hope to use the proposed system, are used to give logging privilege. Logging details with user name and password should be added for that employees.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Employees should be registered</li> <li>• Employee type should be manager, sales executive, stock manger, product supervisor or accountant</li> </ul>
<b>Flow of events</b>	<ul style="list-style-type: none"> <li>• Select employee</li> <li>• Fill the form and Submit</li> </ul>
<b>Postcondition</b>	This information is used when logging to the system

Table B.2: use case narrative of add logging details

## transport Module

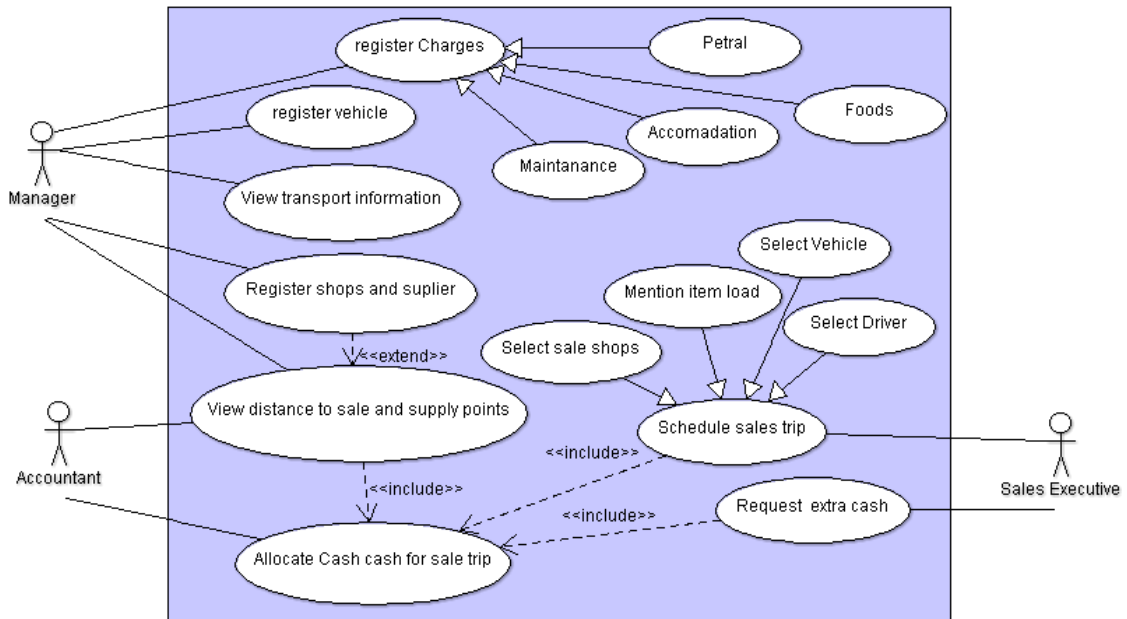


Figure B.2 high level Use-Case of the transport module

1. Use case description for register vehicle

<b>Use case</b>	Register vehicle
<b>Actor</b>	Manager
<b>Over view</b>	
Both company vehicle and rented vehicle should be registered in the system	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Having company vehicles and vehicles for rent</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form and save</li> </ul>	
<b>Postcondition</b>	
This information is used when scheduling sales trip and accounts	

Table B.3: use case narrative of register vehicle

2. Use case description for register charges

<b>Use case</b>	Register charges
<b>Actor</b>	Manager
<b>Over view</b>	
For sales trips, cash should be allocated in few needs (petrel, vehicle maintenance, accommodation, foods)	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Having sales trip</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form and save</li> </ul>	
<b>Postcondition</b>	
This information is used when scheduling sales trips and requesting extra cash	

Table B.4: use case narrative of register charges

### 3. Use case description for Allocate cash for sales trip

<b>Use case</b>	Allocate cash for sales trips
<b>Actor</b>	Accountant
<b>Over view</b>	Sales executive schedule sales trip. Accountant should allocate adequate cash for sales trip
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Scheduled sales trip</li> </ul>
<b>Flow of events</b>	<ul style="list-style-type: none"> <li>Select the sales trip</li> <li>Allocate cash and submit</li> </ul>
<b>Postcondition</b>	This information is used when in accounts

Table B.5: use case narrative of Allocate cash for sales trips

### Account Module



Figure B.3 high level Use-Case of the account module



1. Use case description for Allocate extra cash

<b>Use case</b>	Allocate extra cash
<b>Actor</b>	Accountant
<b>Over view</b>	
When sales executive request extra cash accountant can allocate requested cash if possible	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Requesting extra cash by sales executive</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select cash request</li> <li>• Allocate cash and submit</li> </ul>	
<b>Postcondition</b>	
This information is used when in accounts	

Table B.6: use case narrative of Allocate extra cash

2. Use case description for report income

<b>Use case</b>	Report income
<b>Actor</b>	Accountant
<b>Over view</b>	
When sales executive hand over sales cash and cheques accountant should add them to income reports	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Issue invoice</li> <li>• Sales executive hand over income</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select issued invoice record</li> <li>• Fill the form</li> <li>• save</li> </ul>	
<b>Postcondition</b>	
This information is used when reports generation in accounts.	

Table B.7: use case narrative of Report income

### 3. Use case description for allocate advance

<b>Use case</b>	Allocate advance
<b>Actor</b>	Accountant
<b>Over view</b>	
When worker request an advance by a letter, accountant can allocate advance if worker has not taken advance before. Advance is set to deduct from monthly payment in a determined period.	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Advance request from the worker through a letter</li> <li>• No advances before for that worker</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Fill the form and save</li> </ul>	
<b>Postcondition</b>	
This information is used when calculating monthly salary of workers	

Table B.8: use case narrative of Allocate advance

### 4. Use case description for calculate payment

<b>Use case</b>	Calculate payment
<b>Actor</b>	Accountant
<b>Over view</b>	
Considering daily works, attendance, advances payments is calculated for skilled workers. Other employees are given a permanent salary	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Mark attendance of workers</li> <li>• Record daily work</li> <li>• Consider advances</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select month</li> <li>• Generate payment information. Print payment</li> </ul>	
<b>Postcondition</b>	
This information is used in reporting cost	

Table B.9: use case narrative of calculate payment

5. Use case description for report cost

<b>Use case</b>	Report cost
<b>Actor</b>	Accountant
<b>Over view</b>	
Considering payments, cash for sales trip, extra cash and cash for supply, total cost is generated	
<b>Precondition</b>	
<ul style="list-style-type: none"> <li>• Employee payments</li> <li>• Allocate cash for sales trips</li> <li>• Allocate extra cash</li> <li>• Allocate cash for invoice (refer supplier invoice)</li> </ul>	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Select month</li> <li>• Generate cost information.</li> </ul>	
<b>Postcondition</b>	
This information is used in decision making	

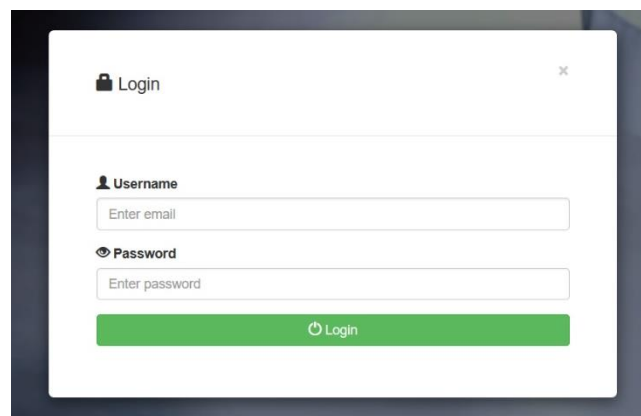
Table B.10: use case narrative of Allocate advance

# Appendix C - User Documentation

This document is for the reference of the users who intend to use the GMS intranet to carry out transactions with system functionalities.

## Login form

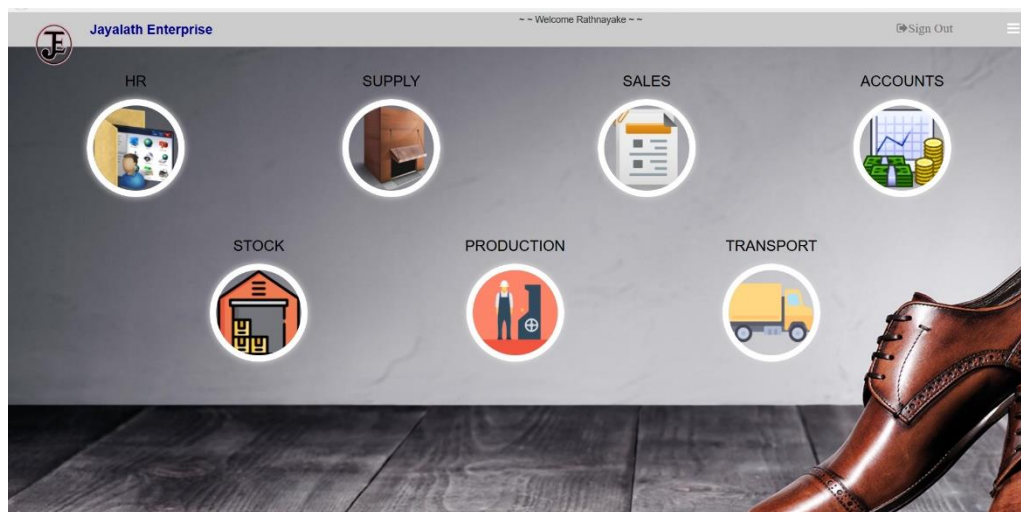
Following Figure C.1 shows user login form which allows users to log into the system. All levels of users can log into the system in one form. When user try to log into the system check whether this user is valid user or not otherwise system display error message.



**Figure C.1 Logging Form**

## Dash board

If user name and password is correct you are directed to your dash board shown in figure C.2



**Figure C.2 dash board**

By clicking on circles on dash board you can navigate relevant modules.

## Logout



**Figure C.3 navigation bar**

Right corner of navigation bar(shown if figure c.3) “sign out” option is available. By clicking on it you can logout the system

## Submit data by a form

Using two interfaces, submitting data by a form is describe. (register material interface- figure C.4 and Place order interface – figure C.5 is used here)

### Register Material

No.	M_Code	Material	amount	description
1	JEM1	brown ledher	50	this is for modern shoes
2	JEM2	laese	500	www
3	JEM4	99999	150	99999
4	JEM5	fffff	57	ffffff

**Figure C.4 Register material interface**

- ❖ Fill all data using text boxes.
- ❖ click on “save” button
- ❖ If everything ok alert would be appeared as “Successfully Registered”. For other forms this alert may differ

registered material

If something wrong “Error” Message would be appeared

error

- ❖ Just after saving, your saving will be appeared in right side table

## Place Order interface

No.	Shop Name	Due Date	Ordered date	Ref.
221	aaaa	27/10/2017	19/10/2017	6
220	aaaa	01/12/2017	17/10/2017	ref 2
218	cccc	29/11/2017	15/10/2017	ref 1
217	cccc	09/11/2017	15/10/2017	ref 3

**Figure C.5 place order interface**

- ❖ If there is combo box click on it and select relevant value
- ❖ For above table select Art no, size and insert amount and click “add” button. You can add several records like this
- ❖ After filling all data of form click on “save” button
- ❖ If everything ok alert would be appeared as “Order sent”. For other forms this alert may differ



- ❖ Just after saving, your saving will be appeared in right side table

## Delete Data

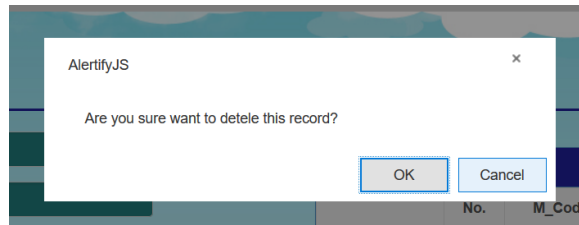
No.	Shop Name
222	bbbb
221	aaaa
220	aaaa
218	cccc

**Figure C.6 Buttons in the table**

In figure C.6, orange color update button and red color delete button can be seen

When click delete button in above mentioned interfaces, an alert appears as below figure

C.7



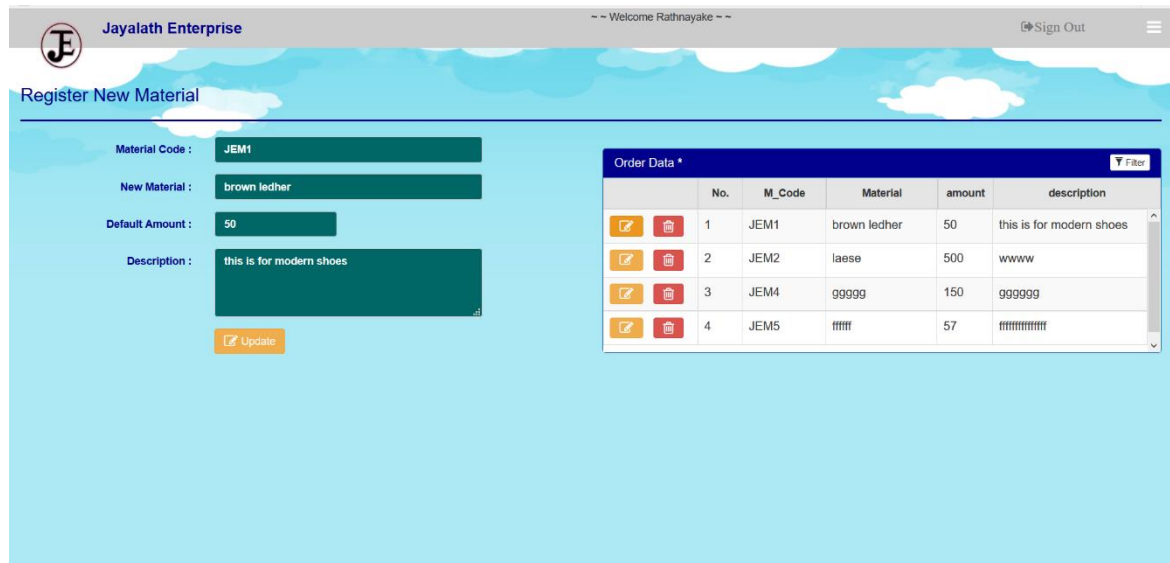
**Figure C.7 alert**

If you want to delete that record, click on “ok”. Else, click “cancel”

## Update Data

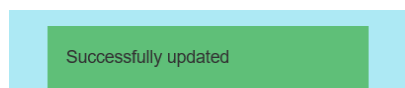
When click on update button following changes happen as following figure C.8 – register material interface and figure C.9 – place order interface

## Register Material



**Figure C.8 Interface of register material after clicking update button**

- ❖ Now you can change values of text boxes
- ❖ Then Click “Update” button
- ❖ If everything ok alert would be appeared as “Successfully updated”. For other forms this alert may differ



## Place order

No.	Shop Name	Due Date	Ordered date	Ref.
218	cccc	29/11/2017	15/10/2017	ref 1
217	cccc	09/11/2017	15/10/2017	ref 3
215	cccc	08/11/2017	08/10/2017	ref 2
214	aaaa	11/10/2017	08/10/2017	ref 1

**Figure C.9 Interface of place order after clicking update button**

- ❖ If you want to add more items, click on blue color “add more items” button. Then form will be changed as figure C.10 below

Art No.	Size	Amount

**Figure C.10 Interface of place order after clicking update button**

- ❖ If you want to delete previous items click on red color “delete previous items” button. Then following modal will be pop up which is in figure C.11

Art No.	Size	Amount
JEP4	37	55
JEP5	37	55
JEP5	37	56

**Figure C.11 pop up modal of place order**



- ❖ Now you can delete items as previously mentioned



# Appendix D - Management Reports

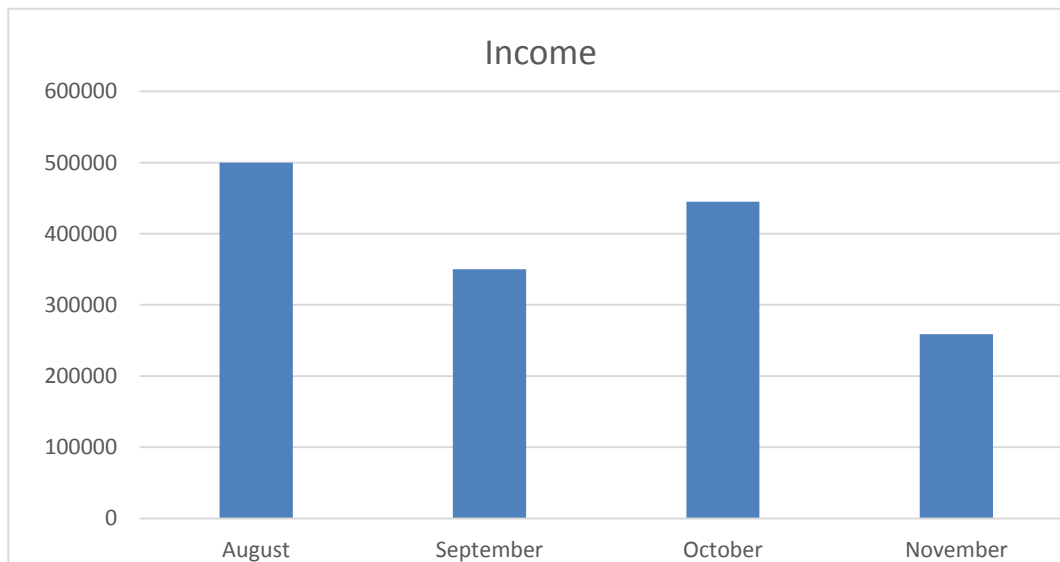
Many Management Reports can be generated by this PADMS.

- Following figure D.1 shows a invoice which is given to customer

 <b>JAYALATH ENTERPRISE</b>				
Manufactures and Distributors of High Quality Shoes Ranasgalla - Nakkawatta ☎ 037 - 2286183 / 071 - 4431980				
B.R.No. :24/1120			Date :2017 Oct.20	
Name :Sampath and brothers				
Address : No. 32,Main street, Alawwa				
Bill No. :5			Salesman : Udara	
Art No.	Size	Qty	Rate	TotRate
JEP6	30	56	500	28 000
JEP6	37	10	500	5000
Total :				33 000
Discount :				3300
Discounted Total :				29 700
 print				

**Figure D.1 pop up modal of place order**

- Following chart given in figure C.11 shows the total income difference in months



**Figure D.2 income chart**

# Appendix E – Test Results

## Test Cases of Supply Module

Test cases, expected results and actual results status of Stock module are displayed in table E.1

Test No.	Test Case	Step to Test	Expected Result	Actual Result Status
1	Make Supply note	When load the interface	• Table should be loaded with registered materials before	pass
2	Make Supply note	When click “save” button	“Material is registered” alert should be appeared.	pass
3	Make Supply note	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
4	Make Supply note	When click on a “update” button in table	Left side form is filled with Data relevant to that record	pass
5	Make Supply note	When load the interface	• Table should be loaded with registered products before	pass
6	Make Supply note	When click “save” button	“product is registered” alert should be appeared.	pass
7	Make Supply note	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
8	Make Supply note	When load the interface	• Table should be loaded with added materials before	pass
9	Refer supplier invoice	When click on “material” combo box	All registered materials are displayed.	pass
10	Refer supplier invoice	When click on “receiver” combo box	Authenticated employees are displayed.	pass
11	Refer supplier invoice	When type text in “Adding amount” test box	Text are not allowed to be typed	pass
12	Refer supplier invoice	When click “save” button	“Material is added” alert should be appeared.	pass
11	Refer supplier invoice	When load the interface	Two tables should be loaded with relevant data	pass
12	Refer supplier invoice	When click on “Issued by” combo box	Authenticated employees are displayed.	pass
13	Refer	When click select button in	• Left side form should be	pass

	supplier invoice	Select Request from Production Department table	filled partially	
14	Refer supplier invoice	When click “save” button	<ul style="list-style-type: none"> <li>• “saved” alert should be appeared.</li> <li>• Record in “Select Request from Production Department” table is disappeared</li> <li>• Relevant record is loaded in “referred invoices” table</li> </ul>	pass

Table E.1: Test cases and results of Supply module

### Test Cases of HR Module

Test cases, expected results and actual results status of Stock module are displayed in table E.2

Test No.	Test Case	Step to Test	Expected Result	Actual Result Status
1	Register Employee	When load the interface	• Table should be loaded with registered materials before	pass
2	Register Employee	When click “save” button	“Material is registered” alert should be appeared.	pass
3	Register Employee	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
4	Register Employee	When click on a “update” button in table	Left side form is filled with Data relevant to that record	pass
5	Register Employee	When load the interface	• Table should be loaded with registered products before	pass
6	Register Employee	When click “save” button	“product is registered” alert should be appeared.	pass
7	Register Employee	When click on a “delete” button in table	Message is appeared as “are you sure want to delete this record”	pass
8	Register Employee	When load the interface	• Table should be loaded with added materials before	pass
9	Register Employee	When click on “material” combo box	All registered materials are displayed.	pass
10	Register Employee	When click on “receiver” combo box	Authenticated employees are displayed.	pass
11	Add logging details	When type text in “Adding amount” test box	Text are not allowed to be typed	pass
12	Add logging details	When click “save” button	“Material is added” alert should be appeared.	pass

<b>11</b>	Add logging details	When load the interface	Two tables should be loaded with relevant data	pass
<b>12</b>	Add logging details	When click on “Issued by” combo box	Authenticated employees are displayed.	pass
<b>13</b>	Add logging details	When click select button in Select Request from Production Department table	<ul style="list-style-type: none"> <li>• Left side form should be filled partially</li> </ul>	pass
<b>14</b>	Add logging details	When click “save” button	<ul style="list-style-type: none"> <li>• “saved” alert should be appeared.</li> <li>• Record in “Select Request from Production Department” table is disappeared</li> <li>• Relevant record is loaded in “referred invoices” table</li> </ul>	pass

Table E.2: Test cases and results of HR module

# Appendix F - Code Listing

## MCV structure in main code segments

- Update

### VIEW

```
<script type="text/javascript">

        $('#emp_update_btn').click(function () {
            Save_reg_emp_update_btn();
        });

</script>
```

### CONTROL

```
function Save_reg_emp_update_btn() {
    var emp_type = $('#reg_emp_type').val();
    var emp_id = $('#reg_emp_id').val();
    var emp_name = $('#reg_emp_name').val();
    var emp_nic = $('#reg_emp_nic').val();
    var emp_add = $('#reg_emp_addres').val();
    var emp_dob = $('#reg_emp_dob').val();
    var emp_email = $('#reg_emp_email').val();
    var emp_cont = $('#reg_emp_tell').val();
    var emp_date = $('#reg_emp_date').val();

    $.post("./model/regis_emp_d.php",
        {action_key: "update", key1: emp_type, key2: emp_id, key3: emp_name, key4:
emp_nic, key5: emp_add, key6: emp_dob, key7: emp_email, key8: emp_cont, key9:
emp_date},
        function (return_val) {

            $.each(return_val, function (index, msgDataAny) {
                if (msgDataAny.msgType === 1) {
                    alertify.success("Successfully updated");
                } else {
                    alertify.error("error");
                }
            });

            clear_reg_emp_form_data();
            $('#reg_emp_update_btn').addClass('hidden');
            $('#reg_emp_save').removeClass('hidden');
            tableload_regis_emp();
        }, "json");
}
```

## MODEL

```
if (array_key_exists("action_key", $_POST)) {
    if ($_POST['action_key'] == 'update') {
        $id = $_POST['key2'];
        $query = "UPDATE `factory_db`.`register_employee` SET
`emp_type`='{$_POST['key1']}', `emp_name`='{$_POST['key3']}',
`emp_nic`='{$_POST['key4']}', `emp_add`='{$_POST['key5']}',
`emp_dob`='{$_POST['key6']}', `emp_email`='{$_POST['key7']}',
`emp_con_no`='{$_POST['key8']}', `emp_apoint_date`='{$_POST['key9']}' WHERE
`emp_id`=" . $id;
        echo $db->setResults($query);
    }
}
```

- Load data to a table from DB

## VIEW

```
<script type="text/javascript">
    $(window).load(function () {
        tableload_regis_emp();
    });
</script>
```

## MODEL

```
if (array_key_exists("action_key", $_POST)) {
    if ($_POST['action_key'] == 'update') {
        $id = $_POST['key2'];
        $query = "UPDATE `factory_db`.`register_employee` SET
`emp_type`='{$_POST['key1']}', `emp_name`='{$_POST['key3']}',
`emp_nic`='{$_POST['key4']}', `emp_add`='{$_POST['key5']}',
`emp_dob`='{$_POST['key6']}', `emp_email`='{$_POST['key7']}',
`emp_con_no`='{$_POST['key8']}', `emp_apoint_date`='{$_POST['key9']}' WHERE
`emp_id`=" . $id;
        echo $db->setResults($query);
    }
}
```

## CONTROL

```
function tableload_regis_emp() {

    var t_content;

    $.post("./model/regis_emp_d.php",
        {action_key: "search2"},
        function (return_val) {
            $.each(return_val, function (index, tabelDataArray) {

                t_content += '<tr>' +
                    '<td style="width:17%;">' +
                    '<button class="btn btn-warning update_emp" value = ' +
                    tabelDataArray.emp_id + '><i class="fa fa-edit fa-lg"></i></button>' + '&nbsp; &nbsp; &nbsp;' +
                    '<button class="btn btn-danger del_emp" value = ' + tabelDataArray.emp_id
                    + ' ><i class="fa fa-trash-o fa-lg"></i></button>' +
                    '</td>' +
                    '<td style="width:8%">' + ++index + '</td>' +
                    '<td style="width:15%">' + tabelDataArray.emp_id + '</td>' +
                    '<td style="width:20%">' + tabelDataArray.emp_type + '</td>' +
                    '<td style="width:20%">' + tabelDataArray.emp_name + '</td>' +
                    '<td style="width:17.3%">' + tabelDataArray.emp_nic + '</td>' +
                    '</tr>'

            });
            $('table_load tbody').html("").append(t_content);

            //////////// DELETE Employee ////////////
            $('del_emp').click(function () {
                var delete_emp_ID = $(this).val();
                alertify.confirm("Are you sure want to detele this record?", function (e) {
                    if (e) {
                        $.post("./model/regis_emp_d.php", {action_key: 'delete', del_employee_id:
                        delete_emp_ID}, function (e) {
                            tableload_regis_emp();
                            alertifyMsgDisplay(e, 2000);
                        }, "json");
                    } else {
                        alertify.log("Cancel process", "Done", 1000);
                    }
                });
            });

            //////////// UPDATE Employee ////////////
            $('update_emp').click(function () {

                emp_update(($this).val());
                $('#reg_emp_save').addClass('hidden');
                $('#reg_emp_update_btn').removeClass('hidden');
            });
        });
    });
}
```

- **Load data to a combo box from DB**

To present that code segment place order interface code segment is used

### **VIEW**

```
<script type="text/javascript">
$(window).load(function () {
    tableload_plc_order();
    plcodr_Combo_box1_load_option();
});
</script>
```

### **MODEL**

```
if (array_key_exists("action_key", $_POST)) {
    if ($_POST['action_key'] === 'combo_artno') {
        $query = "SELECT
            register_product.art_no
            FROM
            register_product";
        echo $db->getResult($query);
    }
}
```



## CONTROL

```
function plcodr_Combo_box1_load_option(selected, callBack) {
    var comboData = '<option selected="true" disabled = "disabled"> -- Select Art No --
</option>';
    $.post("./model/place_order_d.php", { action_key: 'combo_artno'}, function (e) {

        if (e === undefined || e.length === 0 || e === null) {
            comboData += '<option value="0"> -- No Data Found -- </option>';
            $('#place_odr_artno').html("").append(comboData);
            chosenRefresh();
        } else {
            $.each(e, function (index, qData) {
                if (selected !== undefined || e !== null || e.length !== 0) {
                    if (parseInt(selected) === parseInt(qData.art_no)) {
                        comboData += '<option value="" + qData.art_no + "" selected>' + qData.art_no
+ '</option>';
                    } else {
                        comboData += '<option value="" + qData.art_no + "">' + qData.art_no +
'</option>';
                    }
                } else {
                    comboData += '<option value="" + qData.art_no + "">' + qData.art_no +
'</option>';
                }
            });
        }

        $('#place_odr_artno').html("").append(comboData);
        chosenRefresh();
        if (callBack !== undefined) {
            if (typeof callBack === 'function') {
                callBack();
            }
        }

    }, "json");
}
```

# Appendix G - Client Certificate



**JAYALATH ENTERPRISE**

*Manufactures of High Quality Gents Shoes*

Ranasgalla, Nakkawaththa, Tel : 0773 605444, 071-4431980, 027-2286183  
E-mail : jayalathenterprise@yahoo.com

Coordinator,  
External Degree Center of UCSC,  
No 17,  
Swarna Road,  
Colombo 06  
28/10/2017  
Dear Sir/Madam,

## LETTER OF CERTIFICATION

This is to certify that Miss. B.A.S. Prasadinee, who is studying at University of Colombo School of Computing(UCSC) has successfully developed a Production And Distribution management system for Jayalath Enterprise.

And it is glad to say that new system facilitates high productivity when managing day today functions. She has successfully completed our requirements.

This certificate is issued upon the request of Miss B.A.S. prasadinee .

Thankyou.

Yours faithfully,



S.A.Y.L. Subasinghe

(Manger - Operations)

**Jayalath Enterprise**  
**RANASGALLA - NAKKAWATHTHA;**

# Glossary

**RUP** - (Rational Unified Process ) is a software development process from Rational, a division of IBM

**PHP** - (PHP Hypertext Preprocessor) is a server-side scripting language

**UML** - (Unified Modeling Language) is a general-purpose, developmental, modeling language in the field of software engineering

**JS** – (JavaScript) is a scripting languages, primarily used on the Web

**CSS** - (Cascading Style sheets) is a language that describes the style of an HTML document

**My SQL** -MySQL is a relational database management system (RDBMS) based on Structured Query Language (SQL)

**XWAMP** - XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends

**Use case diagrams** - representation of a user's interaction with the system that shows the relationship between the user and the different functions

**Class Diagram** - a type of static structure diagram including the system's classes, their attributes, operations and the relationships among objects.

**sequence diagram** - A diagram which shows object interactions arranged in time sequence

**ER diagram** - entity relationship diagram shows the relationship among the entities of database

# Index

## A

accountant ..... 2, 3, 4, 7, 8, 14, 35, 72, 76, 77  
accounts ..... iii, 1, 4, 5, 14, 35, 74, 75, 76  
Accounts ..... 2, 5, 8, 12, 14, 24  
administrators ..... 4, 5  
AJAX ..... 47, 66  
allocate ..... 14, 74, 76, 77  
analysis ..... iv, 6, 12, 18, 67  
Analysis ..... 11, 16

## B

Bootstrap ..... 47  
business ..... 2, 3, 6, 18, 22, 38, 57, 65, 66

## C

characteristics ..... 40  
Class diagrams ..... 22  
client ..... 4, 21, 47, 57, 63, 65  
CSS ..... iv, 47, 97  
customers ..... iii, 2, 7, 8, 16, 25, 66

## D

dash board ..... ix, 42, 80  
data .. iii, xi, 3, 4, 5, 8, 13, 19, 20, 22, 40, 47, 51, 52, 53,  
54, 56, 57, 58, 61, 62, 63, 80, 81, 82, 87, 89, 91, 93  
design .. iv, 16, 19, 21, 22, 26, 38, 40, 41, 45, 46, 52, 57,  
68  
Design ..... vi, vii, 16, 20, 21, 22, 38, 40, 71  
development .. iv, 16, 17, 18, 19, 21, 23, 46, 47, 67, 68,  
97

## E

effective ..... iii, 1, 4, 6, 21, 65, 66  
email ..... 33, 35, 52  
Employee vii, ix, 2, 3, 5, 8, 13, 28, 42, 43, 52, 59, 72, 78,  
87, 88  
Entity Relationship ..... xi, 39  
environment ..... iv, 1, 2, 7, 46, 63  
executive ..... 2, 4, 7, 8, 24, 26, 33, 72, 74, 76

## F

factory .. iii, iv, 1, 2, 3, 4, 7, 8, 12, 21, 25, 27, 31, 65, 67,  
70, 72  
Features ..... 10, 11

## G

generation ..... 2, 5, 17, 76

## H

high level ... viii, ix, 22, 23, 24, 27, 28, 30, 34, 71, 73, 75  
HTML ..... 47, 97

## I

Implementation ..... vii, xi, 16, 45, 46  
income ..... ix, xi, 3, 9, 12, 14, 27, 65, 76, 85  
industry ..... 1, 2, 3  
information ... iii, iv, 1, 3, 5, 7, 13, 14, 15, 16, 20, 21, 25,  
31, 32, 33, 35, 40, 47, 65, 66, 72, 73, 74, 75, 76, 77,  
78  
interface .... vii, viii, ix, 10, 40, 41, 42, 43, 44, 52, 56, 57,  
58, 59, 60, 61, 62, 81, 82, 83, 86, 87, 88, 89, 93  
Interface ..... vi, vii, ix, 40, 41, 42, 43, 44, 45, 83, 84  
invoice .. vii, ix, x, 3, 8, 12, 14, 15, 16, 24, 25, 26, 27, 35,  
44, 45, 58, 59, 76, 78, 84, 86, 87  
Issue vii, ix, 11, 12, 15, 25, 26, 32, 33, 44, 45, 58, 59, 76

## J

JavaScript ..... 47, 48  
Jayalath Enterprise ..... i, iii, iv, 1, 2, 3, 4, 8, 12, 65  
jQuery ..... 66

## L

Login ..... vii, ix, 5, 41, 42, 71, 79

## M

maintenance ..... 46, 63, 74  
management ..... iii, 2, 5, 15, 65, 66, 67, 97  
material ... vii, ix, x, 2, 3, 7, 8, 10, 13, 15, 29, 30, 31, 32,  
33, 35, 43, 60, 61, 81, 83, 86, 88  
materials ... iii, ix, x, xi, 2, 7, 8, 12, 15, 29, 30, 31, 32, 33,  
43, 60, 61, 62, 86, 87, 88  
method ..... 5, 17, 26, 63  
Model ..... vi, viii, 16, 17, 19, 68  
Module vii, 12, 13, 14, 24, 27, 30, 33, 57, 59, 60, 71, 73,  
75, 85, 87  
MySQL ..... 46, 47, 66, 97

## O

online ..... 3, 65

**P**

password ..... 12, 71, 72, 80  
 payment ..... xi, 3, 7, 8, 13, 20, 26, 77, 78  
 process iv, xi, 1, 2, 3, 4, 5, 6, 7, 8, 12, 16, 17, 19, 20, 27,  
 45, 47, 52, 55, 57, 65, 97  
 Production...i, vi, viii, xi, 1, 2, 3, 4, 5, 13, 15, 16, 20, 22,  
 24, 27, 28, 29, 32, 61, 65, 87, 89  
 programming..... 17, 45

**R**

Reliability ..... 15  
 report generation ..... 4, 35, 66  
 request .....29, 32, 33, 76, 77  
 requirement ..... ii, 6, 12, 14, 16, 21, 38, 57, 69, 70  
 requirements...iv, 4, 6, 7, 12, 14, 15, 19, 21, 55, 65, 67,  
 69, 70  
 Returns ..... 5, 12, 13, 26  
 RUP .....iv, vi, xi, 17, 18, 97

**S**

Sales vii, 2, 4, 5, 8, 12, 16, 24, 25, 26, 33, 57, 63, 74, 76  
 Sequence diagrams ..... 22  
 shoes .....iii, 1, 26

similar .....10, 69  
 software.... iv, 16, 17, 18, 20, 21, 46, 55, 56, 67, 68, 69,  
 70, 97  
 stock... iii, viii, xi, 1, 2, 3, 5, 7, 13, 24, 25, 26, 29, 30, 31,  
 32, 33, 35, 43, 62, 72  
 Stock ...vii, 2, 3, 8, 12, 13, 15, 24, 29, 31, 32, 33, 35, 60,  
 62, 63, 86, 87  
 structure ..... 16, 38, 48, 52, 56, 90, 97  
 Supply ... vii, ix, xi, 2, 5, 8, 12, 24, 31, 33, 43, 44, 62, 85,  
 86, 87

**T**

techniques ..... 6, 7, 20  
 Testing .....vii, 16, 55, 56, 57  
 transport..... iii, ix, 5, 8, 9, 14, 15, 35, 62, 73

**U**

UML ..... iv, 20, 21, 97  
 Use case diagram .....22

**V**

Validation.....51, 55  
 VIEW ..... 48, 51, 52, 54, 90, 91, 93