



# **Sales and Inventory Management System for Eraj Pharmaceuticals**

A.M.A De Silva

BIT registration number: R090150

Index number: 0901504

Name of the supervisor: Mr. R.D Jayaweera

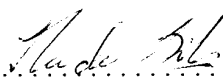
**December 2015**



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

# DECLARATION

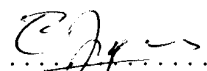
I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where 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: 27/07/15.

Name of Candidate: . . . A.M.A. De Silva . . . . .

Countersigned by:

Signature of Supervisor(s): 

Date: 27/07/15.

Name(s) of Supervisor(s): . . . R.D. Jayaseera . . . . .

# ABSTRACT

Information Technology takes most important place in the modern competitive business world. Without having usage of IT, companies may not be able to do their daily process efficiently and reliably. And business will not be able to compete with competitors. As a business, Eraj Pharmaceuticals decide to gain benefit through Information Technology. They are handling well reputed business and main part of this business is importing and distributing pharmaceuticals and surgical.

Quality of business has been decreased due to lot of manual works. Frequency of occurring inconsistencies is increased due to lot of paper works. Because of low security level data and information could be hacked or destroyed. Generating reports are not accurate. Due to this decision making power has been decreased. Because of these difficulties they are facing many problems.

The proposed system is developing to overcome the problems that are faced by Eraj Pharmaceuticals. Sales and Inventory are the two main areas of the business. The system covered all functionalities of these areas and trying to increase efficiency and effectiveness of the business.

This system is developed according to object oriented principals by using Rational Unified Process and Unified Modeling Language. The project is being developed NetBeans 7.4 as IDE. Microsoft SQL Server is being used as database server. Coding is being done using Java. Reports generated using Jasper Reports.

Finally the SIMS was developed as a solution for problems which are faced by Eraj Pharmaceuticals. The system would provide assistance to perform their company functions efficiently and effectively while decreasing time wasting.

This document fully and formally describes the requirements of the proposed said project system. It sets out the functional and non-functional requirements and includes a description of the user interface and documentation and training requirements.

## **ACKNOWLEDGEMENT**

I would like to express my deepest appreciation to all those who provide me the possibility to complete this project. A special gratitude I give to University of Colombo School of Computing for offer us a great opportunity to improve our practical knowledge and experience of software development. And also I would like to express my thanks to all the coordinators, e-facilitators and other staff of BIT degree programme to provide proper guidance to complete the project.

I would like to acknowledge with much appreciate to my project supervisor Mr. R. D. Jayaweera for his excellent support through the project development and for grate guidance and advices.

I am very grateful and extend my sincere thanks to Mr. R. N. E. C De Silva, the Proprietor of Eraj Pharmaceuticals and the staff for their cooperation by giving me the necessary information to this project.

Finally, a very special thank need to be given to my family, specially my parents and friends who are giving me a support and encourage me to complete this project successfully.

# TABLE OF CONTENT

<b>DECLARATION.....</b>	<b>ii</b>
<b>ABSTRACT.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iv</b>
<b>TABLE OF CONTENT .....</b>	<b>v</b>
<b>LIST OF FIGURES.....</b>	<b>viii</b>
<b>LIST OF TABLES.....</b>	<b>x</b>
<b>LIST OF ACRONYMS.....</b>	<b>xi</b>
<b>CHAPTER 01: INTRODUCTION .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 About the Client .....	1
1.3 Motivation for the project .....	2
1.4 Objectives of the Project .....	2
1.5 Scope of the Project .....	3
1.6 Structure of the Dissertation .....	3
<b>CHAPTER 02: ANALYSIS .....</b>	<b>4</b>
2.1 Introduction to analysis .....	4
2.2 Problems of the Current System .....	5
2.3 Existing Similar Systems .....	5
2.4 Fact Finding Techniques .....	6
2.5 Functional Requirements .....	7
2.6 Non-Functional Requirements .....	8
2.7 Requirement Validation .....	9
	10
<b>CHAPTER 03: DESIGN .....</b>	<b>11</b>
3.1 Introduction to Design .....	11
3.2 Process Models .....	11
3.2.1 Waterfall Module .....	11
3.2.2 Prototyping Module .....	12
3.2.3 Spiral Module .....	12

3.2.4 RAD Module.....	12
3.3 Alternate Solutions .....	12
3.3.1 Manual System .....	13
3.3.2 Already Developed Software .....	13
3.3.3 Reasons to Choose the Standalone System .....	13
3.4 Selected Module for the Proposed System .....	13
3.5 Object Oriented Designing .....	14
3.5.1 Class Diagram for the Proposed System .....	15
3.5.2 Overall High Level Use case Diagram for the Proposed System .....	16
3.6 Database Designing .....	17
3.6.1 Database Normalization .....	18
3.7 User Interface .....	18
<b>CHAPTER 04: IMPLEMENTATION.....</b>	<b>19</b>
4.1 Introduction.....	20
4.2 Hardware and Software Requirements.....	20
4.2.1 Software Environment.....	20
4.2.2 Hardware Environment.....	20
4.3 Development Tools.....	21
4.4 Major Code Segments.....	21
4.4.1 Manage Employee Module.....	23
4.5 Reused Existing Code.....	28
<b>CHAPTER 05: EVALUATION.....</b>	<b>29</b>
5.1 Introduction.....	29
5.2 Techniques of Testing.....	29
5.3 Types Testing.....	29
5.4 Test Plan.....	30
5.5 Test Case Generation.....	31
5.5.1 Test Cases for Login Module.....	31
5.5.2 Test Cases for Manage Customer Module.....	32
5.5.3 Test Cases for Manage Supplier Module.....	33
5.5.4 Test Cases for Manage Product Module.....	33
5.5.5 Test Cases for Manage Stock Module.....	34
5.5.6 Test Cases for Purchase Module.....	34

5.5.5 Test Cases for Manage Stock Module.....	34
5.5.6 Test Cases for Purchase Module.....	34
5.5.7 Test Cases for Sale Module.....	35
5.6 Client Satisfaction Evaluation.....	35
<b>CHAPTER 06: CONCLUSION.....</b>	<b>37</b>
6.1 Introduction.....	37
6.2 Critical Evaluation.....	37
6.3 Problem Encounter.....	38
6.4 Lesson Learnt.....	38
6.5 Future Enhancement.....	39
<b>REFERENCES.....</b>	<b>40</b>
<b>APPENDIX – A SYSTEM DOCUMENTAION.....</b>	<b>43</b>
<b>APPENDIX – B DESIGN DOCUMENTATION.....</b>	<b>44</b>
<b>APPENDIX – C USER DOCUMENTATION.....</b>	<b>55</b>
<b>APPENDIX – D MANAGEMENT REPORTS.....</b>	<b>71</b>
<b>APPENDIX – E TEST RESULTS.....</b>	<b>74</b>
<b>APPENDIX – F CODE LISTING.....</b>	<b>81</b>
<b>APPENDIX – G CLIENT CERTIFICATION.....</b>	<b>86</b>
<b>GLOSSARY.....</b>	<b>87</b>
<b>INDEX.....</b>	<b>88</b>

# LIST OF FIGURES

<i>Figure 2.1 - Main interface of Peachtree Accounting Software.....</i>	<i>6</i>
<i>Figure 2.2 - Main interface of The Inventory Keeper.....</i>	<i>7</i>
<i>Figure 3.1 - Rational Unified Process Model.....</i>	<i>15</i>
<i>Figure 3.2 – Class Diagram for the Sales and Inventory Management System...</i>	<i>16</i>
<i>Figure 3.3 – Overall Use Case Diagram.....</i>	<i>17</i>
<i>Figure B.1 Use-Case Diagram for Sales Module.....</i>	<i>44</i>
<i>Figure B.2 Use-Case Diagram for Purchase Module.....</i>	<i>45</i>
<i>Figure B.3 Use-Case Diagram for Manage Customer Module.....</i>	<i>46</i>
<i>Figure B.4 Use-Case Diagram for Manage Supplier Module.....</i>	<i>47</i>
<i>Figure B.5 Use-Case Diagram for Manage Product Module.....</i>	<i>48</i>
<i>Figure B.6 Use-Case Diagram for Manage Stock Module.....</i>	<i>49</i>
<i>Figure B. Use-Case Diagram for Administration.....</i>	<i>50</i>
<i>Figure B.8 - Activity diagram for overall sales management module.....</i>	<i>51</i>
<i>Figure B.9 - Activity diagram for overall stock management module.....</i>	<i>52</i>
<i>Figure B.10 - Sequence diagram for overall sales management module.....</i>	<i>53</i>
<i>Figure B.11 - Sequence diagram for overall stock management module.....</i>	<i>55</i>
<i>Figure C.1 – Login Screen.....</i>	<i>55</i>
<i>Figure C.2 – Main Screen.....</i>	<i>56</i>
<i>Figure C.3 – Menu Items.....</i>	<i>56</i>
<i>Figure C.4 – Sub Menu Items.....</i>	<i>57</i>
<i>Figure C.5 – Change Password Screen.....</i>	<i>57</i>
<i>Figure C.6 – Add Customer Details Screen.....</i>	<i>58</i>
<i>Figure C.7 – Customer Search Screen.....</i>	<i>58</i>
<i>Figure C.8 – Edit Customer Screen.....</i>	<i>59</i>
<i>Figure C.9 – Add Supplier Details Screen.....</i>	<i>60</i>



<i>Figure C.10 – Supplier Search Screen.....</i>	<i>60</i>
<i>Figure C.11 – Supplier Search Screen.....</i>	<i>61</i>
<i>Figure C.12 – Enter New Stock Details Screen.....</i>	<i>62</i>
<i>Figure C.13 – Add Product Details Screen.....</i>	<i>62</i>
<i>Figure C.14 – Sale Screen.....</i>	<i>63</i>
<i>Figure C.15 – Purchases Screen.....</i>	<i>64</i>
<i>Figure C.16 –Add User Role Screen.....</i>	<i>65</i>
<i>Figure C.17 –Edit User Role Screen.....</i>	<i>65</i>
<i>Figure C.18 –Create User Account Screen.....</i>	<i>66</i>
<i>Figure C.19 –Edit User Account Screen.....</i>	<i>67</i>
<i>Figure C.20 –Search User Account Screen.....</i>	<i>67</i>
<i>Figure C.21 –Edit User Account Screen.....</i>	<i>68</i>
<i>Figure C.22 –Change User Password Screen.....</i>	<i>68</i>
<i>Figure C.23 –All Messages Screen.....</i>	<i>69</i>
<i>Figure C.24 –Product Re-Order Message Screen.....</i>	<i>69</i>
<i>Figure C.25 –Message Received Notification.....</i>	<i>70</i>
<i>Figure C.26 –New Message Screen.....</i>	<i>70</i>
<i>Figure D.1 - Custom Wise Analysis Report.....</i>	<i>71</i>
<i>Figure D.2 - Slow Moving Item Report.....</i>	<i>72</i>
<i>Figure D.3 – Invoice.....</i>	<i>73</i>

# LIST OF TABLES

<i>Table 5.1 – Test cases for Login Module.....</i>	<i>32</i>
<i>Table 5.2 – Test cases for Manage Customer Module.....</i>	<i>33</i>
<i>Table 5.3 – Test cases for Manage Supplier Module.....</i>	<i>33</i>
<i>Table 5.4 – Test cases for Manage Product Module.....</i>	<i>33</i>
<i>Table 5.5 – Test cases for Manage Stock Module.....</i>	<i>34</i>
<i>Table 5.6 – Test cases for Purchase Module.....</i>	<i>34</i>
<i>Table 5.7 – Test cases for Sale Module.....</i>	<i>35</i>
<i>Table B.1- Use-Case Narrative for Sales Module.....</i>	<i>44</i>
<i>Table B.2- Use-Case Narrative for Purchase Module.....</i>	<i>45</i>
<i>Table B.3- Use-Case Narrative for Manage Customer Module.....</i>	<i>46</i>
<i>Table B.4- Use-Case Narrative for Manage Supplier Module.....</i>	<i>47</i>
<i>Table B.5- Use-Case Narrative for Manage Product Module.....</i>	<i>48</i>
<i>Table B.6- Use-Case Narrative for Manage Stock Module.....</i>	<i>49</i>
<i>Table B.7- Use-Case Narrative for Administration.....</i>	<i>50</i>
<i>Table E.1 – Test Results for Login Module.....</i>	<i>75</i>
<i>Table E.2 – Test Results for Customer Module.....</i>	<i>76</i>
<i>Table E.3 – Test Results for Manage Supplier Module.....</i>	<i>77</i>
<i>Table E.4 – Test Results for Manage Product Module.....</i>	<i>77</i>
<i>Table E.5 – Test Results for Manage Stock Module.....</i>	<i>79</i>
<i>Table E.6 – Test Results for Purchase Module.....</i>	<i>79</i>
<i>Table E.7 – Test Results for Sale Module.....</i>	<i>80</i>

# LIST OF ACRONYMS

**BIT** - Bachelor of Information Technology

**CD-ROM** - Compact Disk Read Only Memory

**E-R Diagram** -Entity Relationship Diagram

**GB** -Giga Byte

**GUI** - Graphical User Interface

**HR** – Human Resource

**IDE** - Integrated Development Environment

**IT** - Information Technology

**RAD** - Rapid Application Development

**RAM** -Random Access Memory

**RUP** - Rational Unified process

**SIMS** – Sales and Inventory Management System

**SQL** - Structured Query Language

**UML** - Unified Modeling Language

**WWW** - World Wide Web

# CHAPTER 01: INTRODUCTION

## 1.1 Introduction

In today's advanced technological environment Information Communication Technology is widely used for many applications. Most of the processes in companies are computerized to bring more reliability and efficient in them. The proposed Sales and Inventory Management System (SIMS) is developed to gain the advantages of IT over the problems faced by Eraj Pharmaceuticals in their manual process.

Sales and proper management of stocks are the critical parts of this business. Sale is the activity which provides product or service to the customer. When doing sales, customer interaction is the most important thing. Customer satisfaction of products and keep willingness of the customer is critical thing to keep reputation of the business. Proper management of stocks leads to keep stocks up to date and eliminate over stocking and stock-out.

To implement critical parts of business which are described above using computerized system is more effective and efficient than a manual system. So this project will be helpful to reduce manual paper works and get benefit over IT.

## 1.2 About the Client

Eraj pharmaceuticals is a well known whole sale dealers for distributing of pharmaceuticals and surgical products in western province. It was established in year 2000. Eraj pharmaceuticals is situated in Colombo 11. They are importing and distributing pharmaceuticals and surgical.

### 1.3 Motivation for the Project

Although Eraj Pharmaceuticals earned a good reputation in the market, they have faced lots of difficulties when they are doing their business due to the manual system.

The essential of computerized system had been identified by the Eraj Pharmaceuticals to improve their accuracy and efficiency. As a BIT student I identified their necessity and accept to develop their required system. They had been faced many problems because of manual handling. They can be listed as follows.

- Transaction details are not recorded properly.
- Stock information is not maintained properly.
- Purchase orders are not properly handled by the existing manual system.
- Employees have been demotivated by extensive work load.

### 1.4 Objectives of the project

The main objective of the project is to develop standalone software system to reduce the work load of the employees and improve efficiency and accuracy.

Following are the list of objectives tried to achieve by the proposed system.

- Automated the business process to reduce occurring of errors when doing manual works, minimize delays and eliminate unnecessary documents in manual sales and inventory handling processes to gain improved competitive advantage.
- Allow collecting of all necessary information in centralized database.
- Information should be up to date and accurate to create various important reports that provide valuable information to the top level management.
- Reduce inconsistency of stock and get more accurate and proper control of stocks.
- User friendliness interfaces should be provided to improve efficiency and motivation of users.
- Improve security of the business information.

## 1.5 Scope of the Project

Major barriers for completing this project are available resources and time schedule. The SIMS is the first of introducing a computer based solution to the Eraj Pharmaceuticals.

Sales and Inventory Management are the main areas that the system will implement. Other than that report generation and administrative security functions are provided.

The scope of the system as follows.

- System will keep records of every customer details such as customer ID, name, address, contact number etc. Add, edit, delete of customers are done by using records of customer details. Viewing of particular customer is consisted in the system.
- Recording of transaction details, calculating of total and last price, generation of invoices are included within the transaction handling modeling
- Recording of stock details and product details are include with stock and product modules.
- Recording, editing, deleting of supplier details and viewing that recorded details of suppliers are included with supplier module.
- Accurate report generating and administrative functions are also given.

## 1.6 Structure of the Dissertation

Five main chapters are including with Sales and Inventory Management System after the Introduction chapter.

### **Chapter 02: Analysis**

Main task of this phase is gathering of customer requirements. This chapter documents fact gathering techniques which are used to gather requirements, functional and non-functional requirements, as well as detailed description of current problems. It also describes existing manual system and use case diagram to analyses existing similar systems which are studied to get help to design the new system.

### **Chapter 03: Design**

Design phase of the development process for proposed system is discussed in this chapter. According to gathered requirements, the solution software system has to be designed. UML diagrams are drawn for obtain more appropriate design solution. User interfaces which are helping user to interact with the system are created and databases are designed. All of these are documented within this chapter.

### **Chapter 04: Implementation**

Developing of the system using coding is indicated within this chapter. Which describe the applied coding step by step that is used to develop main modules. Implementation environment such as software requirements, hardware requirements, also mention in this chapter

### **Chapter 05: Evaluation**

The developed system has to be tested using generated test cases. This chapter is consists with test cases, test results, occurred errors, how to correct those errors.

### **Chapter 06: Conclusion**

This is the last main chapter of the dissertation. This will indicate lessons learnt during the project work and future enhancement of the system.

## **CHAPTER 02: ANALYSIS**

### **2.1 Introduction to Analysis**

Systems analysis is a process of collecting factual data; understand the processes involved, identifying problems and recommending feasible suggestions for improving the functioning of the system. This involves studying the business processes, gathering operational data, understanding the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes.

The major objectives of systems analysis are to find answers for each business process: What is being done? How is it being done? Who is doing it? When is he doing it? Why is it being done? How can it be improved?

It is more of a thinking process and involves the creative skills of the System Analyst. It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. [WWW1]

### **2.2 Problems of the Current System**

One of the majority parts of the analysis is identify the problems of the current system. Due to these problems, company will face a lot of problems. Efficiency of the company could be decreased. So the main intention is to overcome problems by developing a new system.

Presently transaction details are not properly recorded and this leads to data inconsistencies. There is no any proper methods for generate invoices. Customer details are recorded poorly and this leads to poor customer interaction. It is also taking a very long time period to find supplier details. Because of the improper stock management lots of data inconsistencies are occurred in this manual system. There is no any security features added to this current manual system. Therefore important confidential data could be accessed by anyone. It is very difficult to generate reports by using this manual system.



Some times reports that generated are not compatible with actual data. Due to this accurate decision making ability is decreased.

## 2.3 Existing Similar Systems

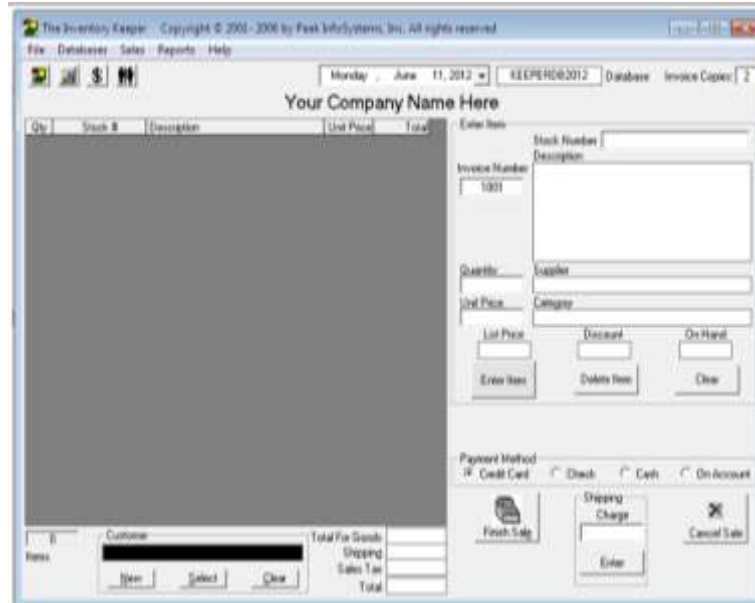
It is valuable to study existing operational system to get idea about main processes, user interfaces etc.

Peachtree accounting software was selected to study about Sales, Transactions, Invoices, Customers, Suppliers/Vendors and Purchases, Inventory Management. Peachtree accounting software is business management software developed by Sage software. There are several software versions has been released and Peachtree complete accounting 2009 version was selected to study.



**Figure 2.1 - Main interface of Peachtree Accounting Software**

Peak InfoSystems Inventory Keeper 5.0.7 is another software which is selected to study an existing similar systems. The Inventory Keeper is used to get an idea about invoicing, stock management, sales management and how to keep track of sales, customers and inventory.



*Figure 2.2 - Main interface of The Inventory Keeper*

## 2.4 Fact Finding Techniques

To study any system the analyst needs to do collect facts and all relevant information. The facts when expressed in quantitative form are termed as data. The success of any project is depended upon the accuracy of available data. Accurate information can be collected with help of certain methods/ techniques. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used by the analyst. The analyst may use more than one technique for investigation. [WW2]

### Interviews

Mostly implemented fact finding technique, where by the system analyst gather and verify requirements from stake holders through face-to-face interaction. Both formal and informal interviews are carried out.

### Document Review

Documents which are used in the current system were analyzed to get a better understanding about document layouts they used. Pay sheets, pay slips. Attendance sheets, leave applications are also analyzed.

### **Observation of the Work Environment**

Study of how the stake holders are performing activities with the existing system. Using this technique, more reliable information can be collected. This is an effective technique for understanding the system.

### **Scenarios**

This is a useful method to gather requirements from stakeholders. Because they can be easily understand the scenario and give feedback for it. Developing of more understandable and flexible scenarios is helped to collect clear and more visible requirements.

## **2.5 Functional Requirement**

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. [WWW3]

Following are the functional requirements for proposed system.

- System should give facility to add, edit and delete customer details.
- System should provide facility to view recorded customer details.
- System should provide facility to add, edit and delete supplier details.
- System should give facility to view recorded supplier details.
- System should provide facility to record transaction details for each and every transaction.
- Transaction details should not be able to edit or delete.
- System should calculate total price and last price for a transaction.
- System should generate invoices for transactions.
- System should provide facility to add product details.
- Product details should not be able to edit or delete. System should provide facility to add stock details.
- Stock details should not be able to edit or delete.
- System should indicate items which have to be purchased.

- System should generate required reports.

## 2.6 Non Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-functional requirements define how a system is supposed to be. They are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements. [WWW4]

Following are the non-functional requirements for proposed system.

- System should provide user friendliness. User interfaces of the system should be simple and easy to understand. System should provide user interaction messages which are understandable and meaningful for the user. Because system should be able to use with less training by its user. Processing time and responsible time of the system should be less.
- System should provide accuracy and consistency. System should process data and information accurately while keeping consistency of them. Proper error messages should be provide to eliminate entering incorrect and ambiguous information.
- System should provide security of data and information. System should provide secure login facility to its user. System should have the responsibility to safe of confidential information.
- System should be reliability. To generate reports, system should be up to date.
- System should be able to install easily and it should not consume too much of memory space and have work under minimum system requirements.

## 2.7 Requirement Validation

Requirement validation is concerned with the process of checking whether the requirements gathered are compatible with customer expectation from the system. During this process, different type of checks should be carried out.

### **Validity Checks**

Collected requirements are validated against customer's actual need. During this process additional or different functions may be identified as requirements which were not gathered before. And wrong judgments were eliminated.

### **Consistency Checks**

These checks to identify whether there are any conflicting requirements in the system. If identified any conflicting requirements, further analysis is done for make them clear.

### **Completeness Checks**

Completeness checks were performed to check whether all requirements were gathered during analysis phase. Missing requirements were identified and collected during this process.

### **Realism Check**

Using knowledge of existing technology, the requirements should be checked to ensure that they can actually be implemented. Check whether the system can be implemented within the time schedules and given system environment.

## CHAPTER 03: DESIGN

### 3.1 Introduction to Design

A new system must be based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the development of a system. The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase the SDLC process continues to move from to what questions of the analysis phase to the how. The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. Data structure, control process, equipment source, workload and limitation of the system, Interface, documentation, training, procedures of using the system, taking backups and staffing requirement are decided at this stage. [WWW5]

### 3.2 Process Models

The software development process involves many activities and it consists of number distinct phases. However, there are number of ways in which these activities are organized into a complete development process. Each of these is called a “Software Development Process Model” or “Software Engineering Model”, which shows how the activities are organized and how they are, linked. [BOOK1]

#### 3.2.1 Waterfall Model

The waterfall model is a sequential design process, often 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. The waterfall development model originates in the manufacturing and construction industries: highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time,

this hardware-oriented model was simply adapted for software development.

[WWW6]

### 3.2.2 Prototyping Model

Software prototyping refers to the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. A prototype typically simulates only a few aspects of, and may be completely different from, the final product. Prototyping has several benefits: The software designer and implementer can get valuable feedback from the users early in the project. The client and the contractor can compare if the software made matches the software specification, according to which the software program is built. [WWW7]

### 3.2.3 Spiral Model

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model (or spiral development). This model of development combines the features of the prototyping and the waterfall model. [WWW8]

### 3.2.4 RAD Model

Rapid Application Development (RAD) is an incremental software development process model that emphasizes an extremely short development cycle. If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a fully functional system within very short time periods (eg. 60 to 90 days). [NOTE1]

## 3.3 Alternate Solutions

Alternate solutions for the SIMS are buying already developed software and stick to the traditional manual system.

### 3.3.1 Manual System

The manual Sales and Inventory Management system is an inexpensive method with no startup costs. A manual SIMS relies heavily on the actions of people. That will increase the possibility of human errors. People might forget to record some transaction details. People might miscount inventory and that will lead to needless additional orders that increase the company's inventory carrying costs and use precious storage space. Inaccurate physical counts of items could also result in not ordering enough of a product. It is very easy to make mistakes when processing transactions manually and the penalty of mistakes especially in calculations of total and last price. An automated system is very easy to implement, quicker and more accurate than a manual system.

### 3.3.2 Already Developed Software

It is not cost effective to buy already developed software. They could be high in price. And there are some disadvantages of buying and using of already developed software. Because those kind of software has not all features that customer needs and some features that couldn't understand. That software may not be included customized features for customer. Because of these reasons, customer may not be satisfied with the already developed software.

### 3.3.3 Reasons to Choose the Standalone System

- Client requested the standalone system with customized features for their company.
- It is easy to develop and maintain a standalone system.
- Hacking possibilities are less because of the security is high.

All features that are needed to company Sales and Inventory Management can be included within the system.



### 3.4 Selected Model for the Proposed System

The Rational Unified Process (RUP) is an iterative software development process framework created by the Software Corporation, a division of IBM since 2003. RUP is not a single concrete prescriptive process, but rather an adaptable process framework. [WWW9]

Reasons of selecting the RUP model for the proposed system are listed below.

- RUP typically helps to resolve project risk, because it is line-up with customer evolving requirements.
- RUP support iterative development
- Therefore it is easy to control changes of the system.
- RUP provides flexibility to handle the project.

RUP is consists with four main phases. They are described bellow

#### **Inception Phase**

This is the smallest phase in the project & ideally it should be quite short. In this phase establish the business case for the system. Identify all external entities & define entity interactions. Use these information to access the contribution that the system makes to the business

#### **Elaboration**

In this phase, develop an understanding of the problem domain; establish an architectural frame work for the system, develop the project plan & identify key project risks.

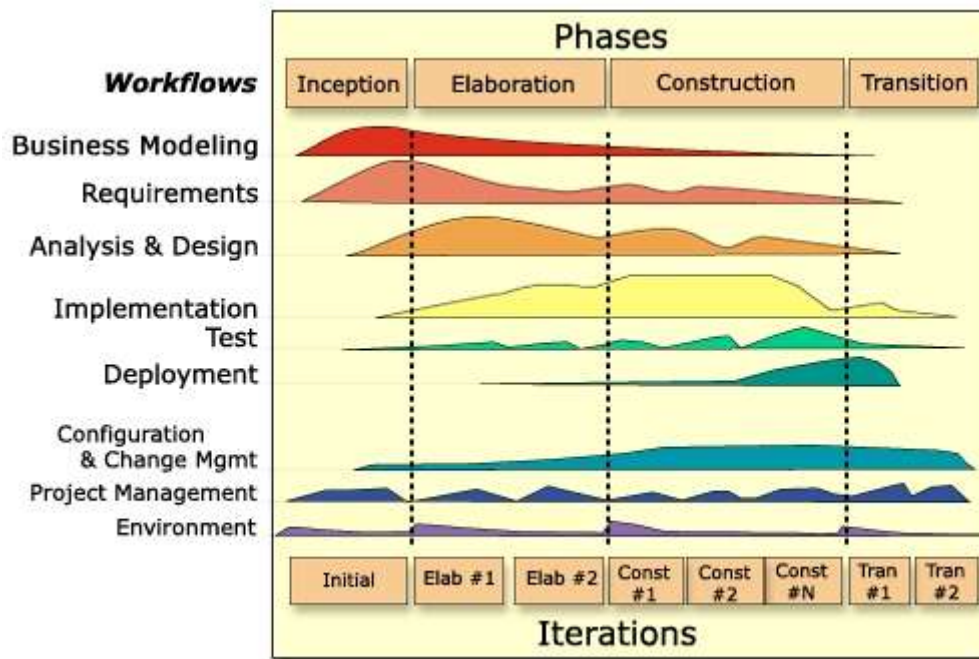
#### **Constructions**

This is the largest phase in the project. This is concerned with system design; programming & testing parts of the system are developed in parallel & integrated during this phase.

#### **Transition**

The final phase is transition. In this phase, system is moving from the development community to the user community. System works in the real environment.

Following figure [WWW10] illustrates the Four Phases of Rational Unified Process



*Figure 3.1 - Rational Unified Process Model*

### 3.5 Object Oriented Designing

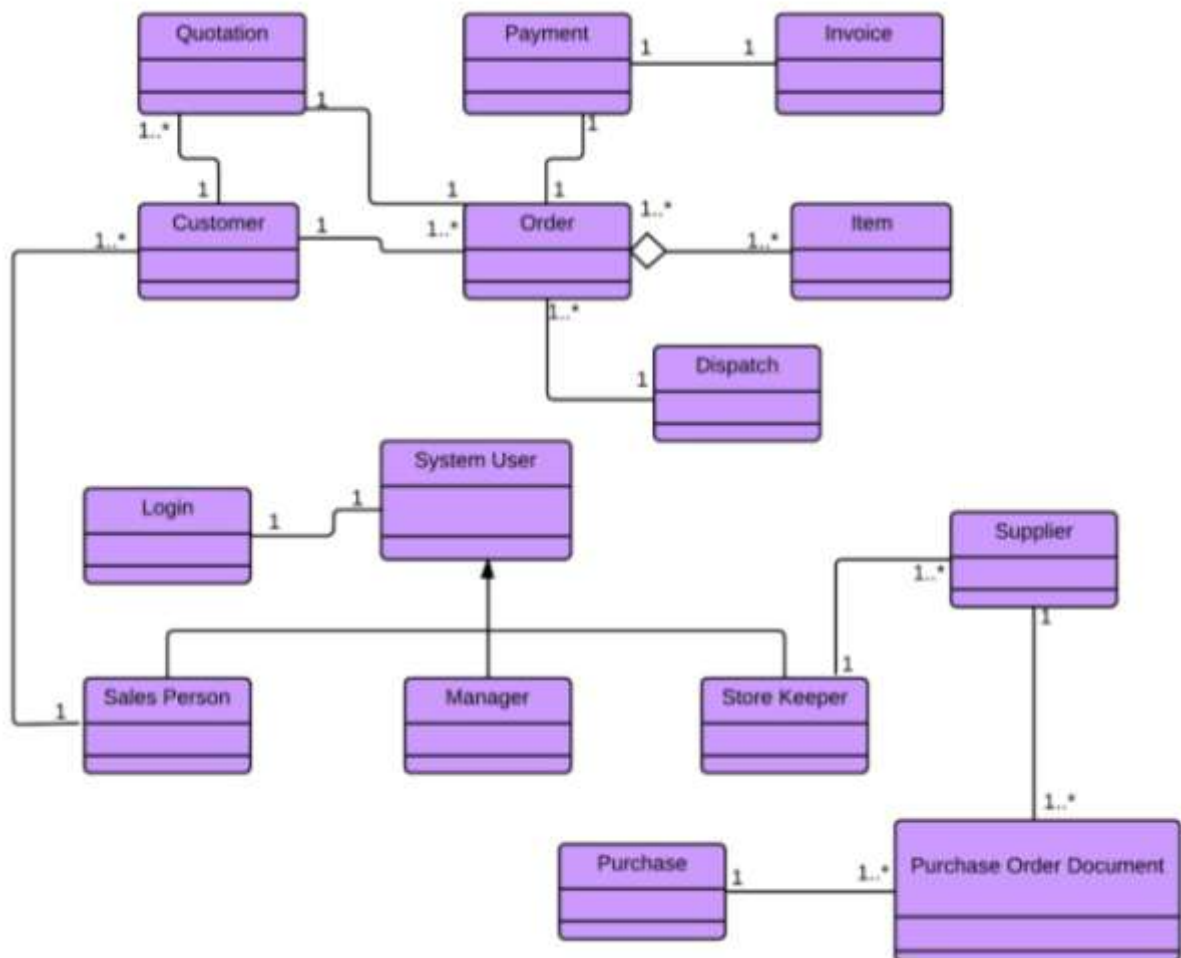
Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its class, its state (data elements), and its behavior. Various models can be created to show the static structure, dynamic behavior, and run-time deployment of these collaborating objects. There are a number of different notations for representing these models, such as the Unified Modeling Language (UML). [WWW11]

Most common design diagrams that are defined in the UML are Use Case diagrams, Activity Diagrams, Sequence diagrams, Class diagrams etc.

### 3.5.1 Class Diagram for the System

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. [WWW12]

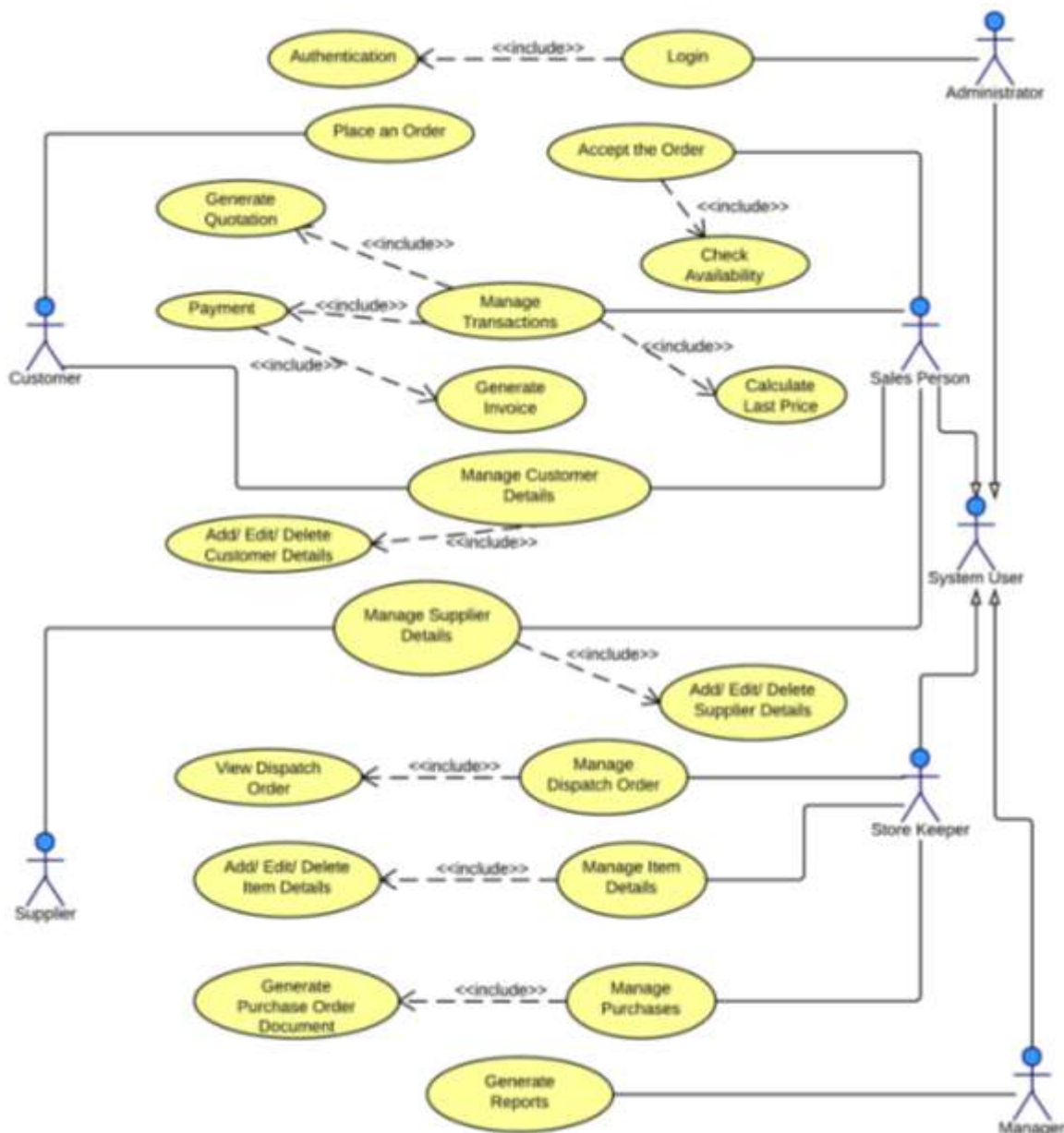
Following figure illustrates the class diagram for the proposed system.



**Figure 3.2 – Class Diagram for the Sales and Inventory Management System**

### 3.5.2 Overall High Level Use Case Diagram for the Proposed System

User view of the UML is represented by the use case diagrams. Use cases and actors are consisted within use case diagrams. Actors may be human or external system. Following figure depicts interaction between actors and use cases for the Sales and Inventory Management System.



**Figure 3.3 – Overall Use Case Diagram**

Please Refer Appendix B – Design Documentation for the rest of the use case diagrams.

## 3.6 Database Design

Database design is the process of producing a detailed data model of a database. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. [WWW13]

### 3.6.1 Database Normalization

In the field of relational database design, normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics—insertion, update, and deletion anomalies—that could lead to a loss of data integrity. [WWW13]

Database Normalization consists of various normal forms and the tables created to proposed system are in 3rd normal form.

**1st Normal Form** - If a relation does not contain repeating groups or multi valued attributes, the relation is in 1st normal form.

**2nd Normal Form** - A relation is in 2nd normal form when there are No partial dependencies on a concatenated key.

**3rd Normal Form** - When a relation is in 3rd normal form there are no transitive dependencies. This means no non-key attributes functionally depends on non-key attributes.

## 3.7 User Interfaces Design

User interface design or user interface engineering is the design of computers. Appliances, machines, mobile communication devices, software applications and website with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible. [WWW14].

**Consistency** - It is important to keep consistency when designing interfaces. All interfaces have to be included same font, same button style, same navigation style and same wording style. All controls have to place consistence places in interfaces.

**Flexibility** – Sometimes stakeholder's requirements are different from the design ones. So interfaces have to be flexible to change according to these requirements.

**Easy Navigation** – Navigation between main user interfaces is make system easy to use, flexibility of the system is increased and customer frustration is decreased.

**User Friendliness** – When displaying user interface, user has to feel comfortable and willingness to use it. To achieve these proper colour combinations, meaningful, understandable wordings and, proper navigation links, proper alignment of fields have to be used. Elimination of crowded screens and grouping of controls effectively leads to get user satisfaction.

**Recoverability** – When designing interfaces, designers must expect mistakes from the user. Meaningful error messages are used to recover errors. Then users are allowed to eliminate errors and doing mistakes.

Please Refer Appendix C – User Documentation for the interface designs.

## CHAPTER 04: IMPLEMENTATION

### 4.1 Introduction

Implementation is the process where software engineers actually program the code for the project. In this phase system converts to its real practical behavior according to its design process. All the system specifications should be implemented during this process. Functional and non-functional requirements which are identified in analysis phase should be satisfied. This is the longest phase of the project

To develop the system appropriate and familiarized programming language and tools should be chosen. Meaningful comments should be applied to the code to get a readability of the coding.

During this phase programmers should make it their central goal to fulfill the functional and non-functional requirements of the company and to meet the design outlined in the design phase. System that does not meet company needs is wasteful.

### 4.2 Hardware and Software Requirements

To get a proper solution of the system, hardware and software environment should be prepared before starting the implementation of the system.

Hardware and software requirements for the Sales and Inventory Management System are as follows.

#### 4.2.1 Software Environment

- Microsoft Windows 7 Ultimate
- NetBeans IDE 7.4
- MySQL Workbench 6.1 CE

#### 4.2.2 Hardware Environment

- 2GB RAM
- Pentium(R)4 CPU 3.20 GHz Processor
- 120GB Hard Disk

## 4.3 Development Tools

**Java** - Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. [WWW15]

**NetBeans IDE 7.4** - NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5.<sup>[3]</sup> It is also an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers. [WWW16]

NetBeans IDE 7.4 extends the advanced HTML5 development support introduced in NetBeans IDE 7.3 to Java EE and PHP applications, while offering new support for mobile web development on the Android and iOS platforms, including support for the Cordova framework. Additional highlights include support for preview versions of JDK 8, and continued enhancements to JavaFX, PHP, C/C++, Maven and other features. [WWW17]

**MySQL Server 5.6** - MySQL is the world's second most widely used open-source relational database management system (RDBMS). The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL



databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use. [WWW18]

**MySQL Workbench 6.1 CE** - MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DB Designer 4 from fabFORCE.net, and replaces the previous package of software, MySQL GUI Tools Bundle. On January 23, 2014 the MySQL Workbench Team announced its first public beta release of Version 6.1. The first general-availability release was made on March 31, 2014. New features include improved Visual Explain output, a Performance dashboard, Performance Schema support, additional query result views, and MSAA support. [WWW19]

**Jasper Reports** – Jasper Reports is an open source Java reporting tool that can write to a variety of targets, such as: screen, a printer, into PDF, HTML, Microsoft Excel, RTF, ODT, Comma-separated values or XML files. It can be used in Java-enabled applications, including Java EE or web applications, to generate dynamic content. It reads its instructions from an XML or .jasper file. Jasper Reports is part of the Lisog open source stack initiative. Jasper Reports is an open source reporting library that can be embedded into any Java application. Features include, Scriptlets may accompany the report definition,<sup>[1]</sup> which the report definition can invoke at any point to perform additional processing. The scriptlet is built using Java, and has many hooks that can be invoked before or after stages of the report generation, such as Report, Page, Column or Group and Sub-reports. For users with more sophisticated report management requirements, reports designed for Jasper Reports can be easily imported into the Jasper Server - the interactive report server. [WWW20]

## 4.4 Major Code Segments

This section illustrates the major code segments which are used to build the system effectively and efficiently. Following are the major code segments for some of main modules. Other code listing can be found in Appendix F.

### 4.4.1 Manage Employee Module

Following code segment is used to add personal details into the system.

```
public class CreateCustomer extends javax.swing.JFrame {
    MasterService masterService = new MasterServiceImpl();
    Validator val = new Validator();
    /**
     * Creates new form Customer
     */
    public CreateCustomer() {
        initComponents();
        Loading();
    }
    //Add customer details
    private void btnCustomerAddActionPerformed(java.awt.event.ActionEvent evt) {
        if (txtCustomerId.getText().trim().equals("") ||
            txtCustomerName.getText().trim().equals("") ||
            txtCustomerAddress.getText().trim().equals("") ||
            txtCustomerMobile.getText().trim().equals("")) //checks whether all required fields are filled
        {
            JOptionPane.showMessageDialog(null, "One or more Required Fields are Empty !", "Save Customer Details", 2);
            return;
        } else {
            if (val.validateCustomerId(txtCustomerId.getText().trim()))//validates customer ID
            format
            {
                try { //to check whether the customer ID is already exists
                    com.abc.salesinventory.model.newpackage.Customer customer =
                    masterService.getCustomer(txtCustomerId.getText().trim());
                    if(customer != null){
                        JOptionPane.showMessageDialog(null, "Customer Already Exists !",
                        "Save Customer Details", 2);
                        return; // Customer ID is already exists.
                    }
                } catch (Exception x) { //Exception handling
                    JOptionPane.showMessageDialog(null, "Error Occured !", "Save Customer
                    Details", 2);
                    return;
                }
            } else { //email address format is wrong
                JOptionPane.showMessageDialog(null, "Incorrect Email Address Format !",
                "Save Customer Details", 2);
            }
        }
    }
}
```

```

txtCustomerEmail.requestFocus(); return; } }
else { //incorrect mobile number format
JOptionPane.showMessageDialog(null, "Incorrect Mobile Number Format !", "Save Customer
Details", 2); txtCustomerMobile.requestFocus(); return;}
} else { //incorrect supplier ID format
JOptionPane.showMessageDialog(null, "Incorrect Customer ID Format", "Save Customer
Details", 2); txtCustomerId.requestFocus(); return; } }
//add customer details into the database
com.abc.salesinventory.model.newpackage.Customer customer = new
com.abc.salesinventory.model.newpackage.Customer();
customer.setId(txtCustomerId.getText().trim());
customer.setName(txtCustomerName.getText().trim());
customer.setMobile(txtCustomerMobile.getText().trim());
customer.setHome(ftxtCustomerHome.getText().trim());
customer.setOffice(ftxtCustomerOffice.getText().trim());
customer.setEmail(txtCustomerEmail.getText().trim());
customer.setAddress(txtCustomerAddress.getText().trim());
masterService.saveOrUpdateCustomer(customer);
JOptionPane.showMessageDialog(null, "Successfully Added", "Save Customer
Details", 2); Loading(); }

catch (Exception e) { //Exception handling
JOptionPane.showMessageDialog(rootPane, e); } }
private void txtCustomerMobileKeyTyped(java.awt.event.KeyEvent evt) {
if (txtCustomerMobile.getText().length() == 10) {
evt.consume(); } //set customer mobile number field length into 10 digits }
private void btnCustomerClearActionPerformed(java.awt.event.ActionEvent evt) {
Loading(); //click cancel button, clear all text fields }
private void btnCustomerCloseActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(1); //click close button, exit from the window }
private void txtCustomerIdKeyTyped(java.awt.event.KeyEvent evt) {
if(txtCustomerId.getText().length()==6)
{ evt.consume(); } //sets customer ID length into 6 digits }
//method to set text fields null
private void Loading() {
txtCustomerId.setText(null);
txtCustomerName.setText(null);
txtCustomerAddress.setText(null);
ftxtCustomerHome.setText(null);
txtCustomerMobile.setText(null);
ftxtCustomerOffice.setText(null);
txtCustomerEmail.setText(null);
txtCustomerId.requestFocus();
}
}

```

This code segment is used to search customer details.

```
public class SearchCustomer extends javax.swing.JFrame {
    MasterService masterService = new MasterServiceImpl();
    /**
     * Creates new form UserSearch
     */
    public SearchCustomer() {
        initComponents();
        txtCustomerName.requestFocus();
    }
    //Query for search customer details
    private static String QUERY_BASED_ON_NAME = "from Customer a where a.name like
    ";
    private static String QUERY_BASED_ON_MOBILE = "from Customer a where a.mobile
    like ";
    private static String QUERY_ALL = "from Customer";
    private void searchCustomer(String hql) {
        try {
            Session session = HibernateUtil.getSessionFactory().openSession();
            session.beginTransaction();
            Query q = session.createQuery(hql);
            List resultList = q.list();
            displayResult(resultList);
            session.getTransaction().commit();
        } catch (HibernateException he) {
            he.printStackTrace();
        }
    }
    //Display customer details in the table
    private void displayResult(List resultList) {

        Vector<String> tableHeaders = new Vector<String>();
        tableHeaders.add("id");
        tableHeaders.add("name");
        tableHeaders.add("address");
        tableHeaders.add("mobile");
        tableHeaders.add("home");
        tableHeaders.add("office");
        tableHeaders.add("email");

        Vector tableData = new Vector();
        for (Object o : resultList) {
```

```
        oneRow.add(customer.getAddress());
        oneRow.add(customer.getMobile());
        oneRow.add(customer.getHome());
        oneRow.add(customer.getOffice());
        oneRow.add(customer.getEmail());
        tableData.add(oneRow);
    }
    resultTable.setModel(new DefaultTableModel(tableData, tableHeaders));
}

//Search customer details
public void search() {
    //Search by customer name
    if (!txtCustomerName.getText().trim().equals("") &&
        txtMobileNumber.getText().trim().equals("")) {
        searchCustomer(QUERY_BASED_ON_NAME + txtCustomerName.getText() +
"%");
    }
    //Search by mobile number
    else if (txtCustomerName.getText().trim().equals("") &&
!txtMobileNumber.getText().trim().equals("")) {
        searchCustomer(QUERY_BASED_ON_MOBILE + txtMobileNumber.getText() +
"%");
    }
    //search by customer name and mobile number
    else if (!txtCustomerName.getText().trim().equals("") &&
!txtMobileNumber.getText().trim().equals("")) {
        searchCustomer("from Customer a where a.name like " +
txtCustomerName.getText().trim() + "%' and a.mobile like " +
txtMobileNumber.getText().trim() + "%");
    }
    //get all data
    else if (txtCustomerName.getText().trim().equals("") &&
txtMobileNumber.getText().trim().equals("")) {
        searchCustomer(QUERY_ALL);
    }
}

//Search customer details
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    search();
}

//Delete selected customer
private void btnDeleteCustomerActionPerformed(java.awt.event.ActionEvent evt) {
    int row = resultTable.getSelectedRow();
    DefaultTableModel model = (DefaultTableModel) resultTable.getModel();
    Vector dataModel = model.getDataVector();
    Customer customer = new Customer();
```

```
Iterator it = dataModel.iterator();
int x = 0;
while (it.hasNext()) {
    Vector vector = (Vector) it.next();
    if (x == row) {
        customer.setId((String) vector.get(0));
        break;
    }
    x++;
}
int dialogButton = JOptionPane.YES_NO_OPTION;
int dialogResult = JOptionPane.showConfirmDialog(null, "Would You Like to Delete
this selected Customer?", "Warning", dialogButton);
if (dialogResult == JOptionPane.YES_OPTION) {
    masterService.removeCustomer(customer);
    search();
    JOptionPane.showMessageDialog(null, "Selected Customer ID is Deleted !", "Delete
Customer Details", 2);
}
}
```

This code segment is used to edit customer details

```
//edit customer
private void btnEditCustomerActionPerformed(java.awt.event.ActionEvent evt) {

    int row = resultTable.getSelectedRow();
    DefaultTableModel model = (DefaultTableModel) resultTable.getModel();
    Vector dataModel = model.getDataVector();
    Customer customer = new Customer();
    Iterator it = dataModel.iterator();
    int x = 0;
    while (it.hasNext()) {
        Vector vector = (Vector) it.next();
        if (x == row) {
            customer.setId((String) vector.get(0));
            customer.setName((String) vector.get(1));
            customer.setAddress((String) vector.get(2));
            customer.setMobile((String) vector.get(3));
            customer.setHome((String) vector.get(4));
            customer.setOffice((String) vector.get(5));
            customer.setEmail((String) vector.get(6));
            break;
        }
        x++;
    }
    EditCustomer ec = new EditCustomer(customer);
    ec.setVisible(true);
}
```

## 4.5 Reused Existing Code

Some code segments used to build the system are got after referring websites and forums. They are directly used to implement some important features of the system. Following indicate the reused existing codes used by

MD5 Hash is used to encrypt and decrypt passwords for user accounts of the system. Following code segment is found from brezovsky.net [WWW21]

```
//http://blog.brezovsky.net/en-text-2.html
//MD5 Hash is used to encrypt the password
System.Security.Cryptography.MD5CryptoServiceProvider x = new
System.Security.Cryptography.MD5CryptoServiceProvider();
byte[] bs = System.Text.Encoding.UTF8.GetBytes(input);
bs = x.ComputeHash(bs);

System.Text.StringBuilder s = new System.Text.StringBuilder();

foreach (byte b in bs)
{
    s.Append(b.ToString("x2").ToLower());
}
string password = s.ToString();
return password;
```

## CHAPTER 05 – EVALUATION

### 5.1 Introduction

System testing is an important part of the software development life cycle. During this phase, should perform variety of tests on a system to clear bugs such as errors and defects. The entire system is tested to verify whether the system accomplished its goals. System quality can be increased, when the system is facing the tastings. Validation and verification are the main objectives of testing.

**Validation** - Refers to set of activities that ensure that the software that has been built is the one that customer needs which matches with their requirements.

**Verification** - Refers to the set of activities that ensure that software correctly implements the specified functions.

### 5.2 Techniques of Testing

#### **White Box Testing**

Also known as transparent box testing, clear box testing, structural testing and glass box test. This is a methodology that used to ensure and validate the internal frame work, mechanisms, objects and components of software application and compares actual result and expected result. This is used to examine whether the code works as expected.

#### **Black Box Testing**

Also known as functional testing. Testing software based on output requirement and without knowing any of the internal structure or coding in the program. Tester only knows the inputs and what the expected outcomes should be.

### 5.3 Types of Testing

#### **Unit Test**

Unit testing follows white box testing approach, where developer will test units of source code like statements, functions, methods, interfaces or classes. The purpose of unit testing is to find whether the implemented unit is producing expected output against given input.



### **Integration Testing**

In this testing, units and group of components are integrated to produce an output. Interaction between software and hardware is also testing when there is a relation between software & hardware. This falls under both category of white box and black box testing.

### **System Testing**

This includes multiple testing types that will enable to validate the software as a whole (software, hardware and network) against the requirements for which it was built. Different types of tests are carried out to complete system testing such as GUI testing, functional testing, regression testing, smoke testing etc. This falls under the category of black box testing.

### **Stress testing**

In this type of testing, software is subjected to peak loads and even to a breakpoint to observe how the software would behave at breakpoints. This belongs to black box testing.

### **Acceptance Testing**

This is a formal type of testing that is performed by the customer to ensure that the system meet their business needs and the requirements and work as they expected. This is a kind if a black box testing.

### **Regression Testing**

After done modifications of the system, regression testing is conducted to ensure the system, its components and group of related unit works properly without unexpected results.

## **5.4 Test Plan**

A testing plan is a methodological and systematic approach to testing a system such as a machine or software. It can be effective in finding errors and flaws in a system. In order to find relevant results, the plan typically contains experiments with a range of operations and values, including an understanding of what the eventual workflow will be. [WWW22].

SIMS is included both techniques of white box and black box testing. Unit testing is conducted when coding the system. Integration testing is done when integrate different units of the system. System testing is performed by implementing the system in different environment. Acceptance testing is done by the customer to ensure the requirements that he expected from the system are satisfied. Regression testing is conducted when system modifications are done. When a modification is done in a one code in a unit, sometimes it might affect another unit of the system. In situations like that, system checks to avoid crashes and unexpected results.

### Advantages of test plan

- Established a test schedule.
- Can identify what will be tested and what will be not tested.
- Can identify test cases.
- To achieve 100% correct code. Ensure all functional and non functional and designed requirements are implemented as specified.

Test planning is the way to avoid problematic testing and test cases.

## 5.5 Test Case Generation

Most important part of the test planning is test case generation. Test cases are set of conditions and variables which determine whether the system is working correctly or not.

Following tables illustrates test cases which are designed for each module separately. (Test result for designed test cases is included in Appendix E-Test result with test screen shots.)

### 5.5.1 Test Cases for Login Module

ID	Test Case Description	Expected Result/s	Priority
1	Username is not entered	Prompt user to fill required fields	High
2	User role is not entered	Prompt user to fill required fields	High
3	Password is not entered	Prompt user to fill required fields	High
4	Confirm password is not entered	Prompt user to fill required fields	High
5	Confirm password is not match with previous entered password	Prompt as confirmation is failed	High
6	Entered already exist username	Prompt as username is already entered	High

7	Click submit button to save account	Create account and prompt as successfully added	High
8	Click cancel button	Clear all text boxes	High
9	Username is not entered	Prompt user to fill required fields	High
10	Existing password is not entered	Prompt user to fill required fields	High
11	New Password is not entered	Prompt user to fill required fields	High
12	Confirm password is not entered	Prompt user to fill required fields	High
13	Click change button to change password	Change the password and prompt as successfully changed	High
14	Click delete button to delete an account	Prompt user to get confirmation	High
		Delete account and prompt as successfully deleted	Medium
15	User enter wrong username and correct password	Prompt as invalid username or password	High
16	User enter wrong username and wrong password	Prompt as invalid username or password	High
17	User enter correct username and wrong password	Prompt as invalid username or password	High
18	User enter correct username and correct password	Login to the system	High
19	User click cancel button	Clear all text boxes	High

**Table 5.1 – Test cases for Login Module**

### 5.5.2 Test Cases for Manage Customer Module

ID	Test Case Description	Expected Result/s	Priority
1	Customer ID is not entered	Prompt user to fill required fields	High
2	Customer name is not entered	Prompt user to fill required fields	High
3	Customer address is not entered	Prompt user to fill required fields	High
4	Customer mobile number is not entered	Prompt user to fill required fields	High
5	Customer ID format is incorrect	Prompt as wrong format	High
6	Customer mobile number format is incorrect	Prompt as wrong format	High
7	Customer email address format is incorrect	Prompt as wrong format	High
8	Customer ID is already exist	Prompt as customer ID is already exist	High
9	Click add button	Add details and prompt as successfully added	High
10	Click cancel button	Exit from the window	Medium
11	Click clear button	Clear all text fields	Medium
12	Click search button	View all customer details	High
13	Enter customer name and click search button	View relevant details	High
14	Enter mobile number and click search button	View relevant details	High
15	Click delete button on the selected row in table	Delete data from database and prompt as deleted successfully	High

<b>15</b>	Click edit button	Goes to edit customer window	High
<b>16</b>	Enter required details and click update button	Prompt as updated successfully	High

**Table 5.2 – Test cases for Manage Customer Module**

### 5.5.3 Test Cases for Manage Supplier Module

<b>ID</b>	<b>Test Case Description</b>	<b>Expected Result/s</b>	<b>Priority</b>
<b>1</b>	Supplier ID is not entered	Prompt user to fill required fields	High
<b>2</b>	Supplier name is not entered	Prompt user to fill required fields	High
<b>3</b>	Supplier address is not entered	Prompt user to fill required fields	High
<b>4</b>	Supplier mobile number is not entered	Prompt user to fill required fields	High
<b>5</b>	Supplier ID format is incorrect	Prompt as wrong format	High
<b>6</b>	Supplier mobile number format is incorrect	Prompt as wrong format	High
<b>7</b>	Supplier email address format is incorrect	Prompt as wrong format	High
<b>8</b>	Supplier ID is already exist	Prompt as supplier ID is already exist	High
<b>9</b>	Click add button	Add details and prompt as successfully added	High
<b>10</b>	Click cancel button	Exit from the window	Medium
<b>11</b>	Click clear button	Clear all text fields	Medium
<b>12</b>	Click search button	View all customer details	High
<b>13</b>	Enter supplier name and click search button	View relevant details	High
<b>14</b>	Enter mobile number and click search button	View relevant details	High
<b>15</b>	Click delete button on the selected row in table	Delete data from database and prompt as deleted successfully	High
<b>16</b>	Click edit button	Goes to edit supplier window	High
<b>17</b>	Enter required details and click update button	Prompt as updated successfully	High

**Table 5.3 – Test cases for Manage Supplier Module**

### 5.5.4 Test Cases for Manage Product Module

<b>ID</b>	<b>Test Case Description</b>	<b>Expected Result/s</b>	<b>Priority</b>
<b>1</b>	Product code is not entered	Prompt user to fill required fields	High
<b>2</b>	Product Name is not entered	Prompt user to fill required fields	High
<b>3</b>	Category is not selected	Prompt user to fill required fields	High
<b>4</b>	Re order level is not entered	Prompt user to fill required fields	High
<b>5</b>	Unit is not selected	Prompt user to fill required fields	High
<b>6</b>	Click save button	Add details and prompt as successfully added	High
<b>7</b>	Click clear button	Clear all text fields	Medium
<b>8</b>	Click close button	Exit from the window	Medium

**Table 5.4 – Test cases for Manage Product Module**

## 5.5.5 Test Cases for Manage Stock Module

ID	Test Case Description	Expected Result/s	Priority
1	Stock ID is not entered	Prompt user to fill required fields	High
2	Product Name is not selected	Prompt user to fill required fields	High
3	Supplier Name is not selected	Prompt user to fill required fields	High
4	Quantity is not entered	Prompt user to fill required fields	High
5	Purchased unit price in not entered	Prompt user to fill required fields	High
6	Selling unit price is not entered	Prompt user to fill required fields	High
7	Expiry date is not entered	Prompt user to fill required fields	High
8	Click save button	Add details and prompt as successfully added	High
9	Click clear button	Clear all text fields	Medium
10	Click close button	Exit from the window	Medium

*Table 5.5 – Test cases for Manage Stock Module*

## 5.5.6 Test Cases for Purchase Module

ID	Test Case Description	Expected Result/s	Priority
1	Transaction ID is not entered	Prompt user to fill required fields	High
2	Date is not entered	Prompt user to fill required fields	High
3	Purchase number is not entered	Prompt user to fill required fields	High
4	Supplier name is not selected	Prompt user to fill required fields	High
5	Select a supplier name	Other fields of supplier details fills with relevant data	High
6	Payment method is not selected	Prompt user to fill required fields	High
7	Product name is not selected	Prompt user to fill required fields	High
8	Select a product name	Other fields of product details fills with relevant data	High
9	Purchased quantity is not entered	Prompt user to fill required fields	High
10	Press Enter button	Gets the amount	High
11	Amount Field is empty	Prompt user to fill required fields	High
12	Click add button	Add data to the data table	High
13	Click total button	Gives the total purchased value	High
14	Total field is empty	Prompt user to fill required fields	High
15	Select a row and click remove selected item button	Prompt as selected item is removed	High
16	Click add transaction	Add transaction to the database and prompt as transaction added successfully	High
17	Click cancel transaction button	Cancel the transaction and clear the table	High
18	Click view purchased receipt	View the purchased receipt	High
19	Click view older purchased receipts	View older purchase receipts	Medium
20	Click close button	Exits from the window	medium

*Table 5.6 – Test cases for Purchase Module*

### 5.5.7 Test Cases for Sale Module

ID	Test Case Description	Expected Result/s	Priority
1	Transaction ID is not entered	Prompt user to fill required fields	High
2	Date is not entered	Prompt user to fill required fields	High
3	Purchase number is not entered	Prompt user to fill required fields	High
4	Customer name is not selected	Prompt user to fill required fields	High
5	Select a customer name	Other fields of customer details fills with relevant data	High
6	Payment method is not selected	Prompt user to fill required fields	High
7	Product name is not selected	Prompt user to fill required fields	High
8	Select a product name	Other fields of product details fills with relevant data	High
9	Purchased quantity is not entered	Prompt user to fill required fields	High
10	Press Enter button	Calculates the price	High
11	Total value Field is empty	Prompt user to fill required fields	High
12	Click add button	Add data to the data table	High
13	Click net invoice button	Gives the total invoice value	High
14	Net Invoice value field is empty	Prompt user to fill required fields	High
15	Select a row and click remove selected item button	Prompt as selected item is removed	High
16	Click add transaction	Add transaction to the database and prompt as transaction added successfully	High
17	Click cancel transaction button	Cancel the transaction and clear the table	High
18	Click view invoice	View the invoice	High
19	Click view older invoice	View older invoices	Medium
20	Click close button	Exits from the window	medium

**Table 5.7 – Test cases for Sale Module**

## 5.6 Client Satisfaction Evaluation

Client satisfaction evaluations are an excellent opportunity to involve client in the process of evaluating the system. Amount of users in SIMS is considerably low in this stage of the system, because current stage of system is only available for internal users. One manager, three store keepers, four salesmen were selected as target population for the evaluation.

Questionnaire was given to target population. Following questionnaire was used when doing user evaluation survey.

### CLIENT EVALUATION SHEET

User Name : .....

Role:.....

Please tick appropriate answer.

(1 – Very Poor, 2 – Poor, 3 – Average, 4 – Good, 5 – Very Good)

(1) Over roll reaction : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(2) Character readability : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(3) Colour scheme : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(4) System navigation : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(5) Ease of usage : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(6) Functionalities : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(7) Interfaces : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(8) Ease of learning : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(9) Response time : 1 ( ) 2 ( ) 3 ( ) 4 ( ) 5 ( )

(10) Other comments :

--

Test results are included in Appendix E – Test Results

## CHAPTER 06: CONCLUSION

### 6.1 Introduction

Eraj pharmaceutical is a well known whole sale dealers for distributing of pharmaceuticals and surgical products in western province. The proposed Sales and Inventory Management system is developed to gain the advantages of an automated system over the problems faced by the Eraj pharmaceutical. The main functional area of the business is sales management and stock management. Most of the functions are manually handle, so lots of inconsistencies had been happened while doing transactions and stock handling. And it is a most time consuming task. All invoices, purchase reciepts are generated manually. These reasons lead to decrease the effectiveness and efficiency of the business.

The developed system facilitated all functional areas of the business. Relationship among those areas decreases most of inconsistencies that were happened. It is a big advantage for the business. Handling of customer details increases the relationship with customers. Developed system generates invoices and purchase receipts automatically. According to client opinion the developed system will provide easiness to their works and increase efficiency and effectiveness of the business.

### 6.2 Critical Evaluation

Functional and non functional requirements are discovered in analysis phase and review and check back in every phase to verify whether they are satisfied in the system. Different fact finding techniques are used in analysis phase. They are interviews, document review, observation of the work environment and scenarios. Requirement Validation is performed using different type of checks such as validity checks, consistency checks, completeness checks and realism checks. Object oriented designing and implementation is used and several UML diagrams were drawn at the design phase. RUP Methodology is used to develop the system. Simple, user friendly and understandable user interfaces are created.



System testing is performed since development phase. Different type of testing such as unit testing, integration testing, system testing, stress testing, acceptance testing and regression testing are performed. Test Case generation is also done in testing phase.

The language used to develop the system is java and NetBeans IDE 7.4 and MySQL Server 5.6 with MySQL Workbench 6.1 CE is used as implementation environment. Jasper reports are used to generate reports.

### 6.3 Problem Encounter

One of the major problem is faced is the interaction with client because of their busy work schedule. System developed using new programming language and there were no previous experience with language. Getting familiar with the language is one of initial effort of the project.

### 6.4 Lesson Learnt

Implementation of theoretical knowledge which is gained through past three years of the BIT degree program in a practical manner is the major lesson that I learnt. When I was going through the Software Development Life Cycle step by step I got the actual experience of how a software system becomes its final state. I learnt how to interaction with client and how to get requirements effectively and get experience to manage different stake holder requirements and how to do suggestion to change them in professional manner. Time management is the most critical experience of the project. Developed the system with new programming language increased my knowledge and gave me an experience of how to cope with an inexperienced language. The project helped me to learn new features of NetBeans and also get vast experience of creating and handling database and database tables in MySQL through MySQL Workbench. Dissertation writing is another usable experience of project work. It is done while developing the system. I learnt how to write a project report in a professional way and it is increased my knowledge of professional English also.

## 6.5 Future Enhancement

System quality and functionality can be improved by adding new features, modules and hardware.

- The system would be integrated with a bar code reader to improve efficiency of data entry.
- Increase more security features to improve quality of the system further.

## REFERENCES

- [WWW1]- oer.nios.ac.in, "Phases of System Development Life Cycle"  
[http://oer.nios.ac.in/wiki/index.php/Phases\\_of\\_System\\_Development\\_Life\\_Cycle](http://oer.nios.ac.in/wiki/index.php/Phases_of_System_Development_Life_Cycle)  
[15/01/2014]
- [WWW2]- systemanalysisanddesign.blogspot.com, "System Analysis and Design:  
Fact Finding Techniques"  
<http://systemanalysisanddesign.blogspot.com/2008/11/fact-finding-techniques.html>  
[15/01/2014]
- [WWW3]- en.wikipedia.org, "Functional Requirements"  
[http://en.wikipedia.org/wiki/Functional\\_requirement](http://en.wikipedia.org/wiki/Functional_requirement) [19/01/2014]
- [WWW4]- en.wikipedia.org, "Non-Functional Requirements"  
[http://en.wikipedia.org/wiki/Non-functional\\_requirement](http://en.wikipedia.org/wiki/Non-functional_requirement) [31/01/2014]
- [WWW5]- oer.nios.ac.in, "Phases of System Development Life Cycle"  
[http://oer.nios.ac.in/wiki/index.php/Phases\\_of\\_System\\_Development\\_Life\\_Cycle](http://oer.nios.ac.in/wiki/index.php/Phases_of_System_Development_Life_Cycle)  
[31/01/2014]
- [BOOK1]- IT-3203 Software Engineering, 2010, Esoft Computer Studies (Pvt)  
Ltd, Galle Road, Colombo 4 [14/02/2014]
- [WWW6]- en.wikipedia.org, "Waterfall Model"  
[http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model) [17/02/2014]
- [WWW7]- en.wikipedia.org, "Software Prototyping"  
[http://en.wikipedia.org/wiki/Software\\_prototyping](http://en.wikipedia.org/wiki/Software_prototyping) [17/02/2014]

**[WWW8]**-en.wikipedia.org, "Spiral Model"

[http://en.wikipedia.org/wiki/Spiral\\_model](http://en.wikipedia.org/wiki/Spiral_model) [21/02/2014]

**[NOTE1]**-University of Colombo School of Computing, 2010, "IT3203-Software Process Models" [28/02/2014]

**[WWW9]**-en.wikipedia.org, "IBM Rational Unified Process"

[http://en.wikipedia.org/wiki/IBM\\_Rational\\_Unified\\_Process](http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process) [05/03/2014]

**[WWW10]** - era.nih.gov, "RUP Fundamentals Presentation"

[http://era.nih.gov/docs/rup\\_fundamentals.htm](http://era.nih.gov/docs/rup_fundamentals.htm) [12/03/2014]

**[WWW11]** - en.wikipedia.org, "Object-oriented analysis and design"

[http://en.wikipedia.org/wiki/Object-oriented\\_analysis\\_and\\_design](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) [14/03/2014]

**[WWW12]** - en.wikipedia.org, "Class diagram"

[http://en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram) [20/03/2014]

**[WWW13]**- en.wikipedia.org, "Database Design"

[http://en.wikipedia.org/wiki/Database\\_design](http://en.wikipedia.org/wiki/Database_design) [09/04/2014]

**[WWW14]** - en.wikipedia.org, "User interface design"

[http://en.wikipedia.org/wiki/User\\_interface\\_design](http://en.wikipedia.org/wiki/User_interface_design) [19/04/2014]

**[WWW15]**- en.wikipedia.org, "Java (programming language)"

[http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)) [05/05/2014]

**[WWW16]** - en.wikipedia.org "NetBeans"

<http://en.wikipedia.org/wiki/NetBeans> [05/05/2014]

**[WWW17]** - netbeans.org “NetBeans IDE 7.4 Information”

<https://netbeans.org/community/releases/74/> [05/05/2014]

**[WWW18]** en.wikipedia.org “MySQL server”

[http://en.wikipedia.org/wiki/MySQL\\_server](http://en.wikipedia.org/wiki/MySQL_server) [05/05/2014]

**[WWW19]** – en.wikipedia.org “MySQL Workbench”

[http://en.wikipedia.org/wiki/MySQL\\_Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) [05/05/2014]

**[WWW20]** - en.wikipedia.org “Jasper Reports”

<http://en.wikipedia.org/wiki/JasperReports> [05/05/2014]

**[WWW21]**- <http://blog.brezovsky.net>, "How to get MD5 hash in C#.NET"

<http://blog.brezovsky.net/en-text-2.html>[03/06/2013]

**[WWW22]**-testingsoftware.blogspot.com "what is test plan"

[http://testingsoftware.blogspot.com/2005/11/what-is-test-plan\\_30.html](http://testingsoftware.blogspot.com/2005/11/what-is-test-plan_30.html)[01/07/2014]

# APPENDIX - A

## SYSTEM DOCUMENTATION

### Minimum Hardware Requirements

- 2GB RAM
- Pentium(R)4 CPU 3.20 GHz Processor
- 120GB Hard Disk
- CD-Rom Driver
- Printer
- 

### Minimum Software Requirements

- Microsoft Windows 7
- NetBeans IDE 7.4
- MySQL Server 5.6
- MySQL Workbench CE 6.1

### Database Setup

**Step 1** - Install the MySQL Server 5.6

**Step 2** - Install the MySQL Workbench CE 6.1

**Step 3** - Create database called SIMS in the server

**Step 4** - Select Eraj Pharmaceuticals script file and open it in the a query window

**Step 5** - Execute the script file

### Sales and Inventory Management System Setup

**Step 1**- Download the Java Runtime Environment

**Step 2**- Create a folder called SIMS

**Step 3** - Copy the Eraj Pharmaceuticals folder given in the supplementary CD and paste it inside that folder

**Step 4**- Open the Eraj Pharmaceuticals folder

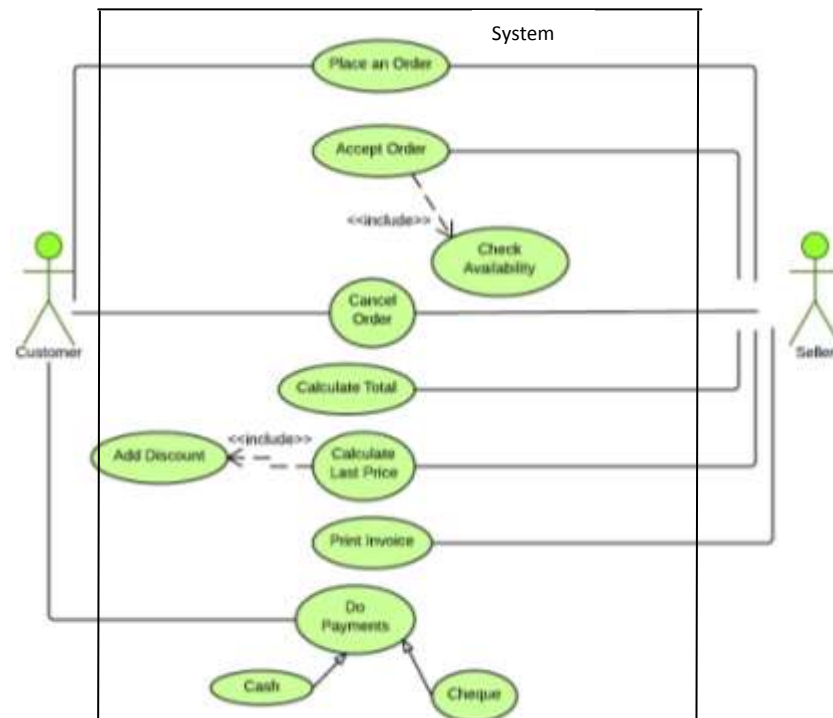
**Step 5**- Click on the SIMS.jar file and open with java platform (or Set "Java platform" as the default program to open .jar files and then double click and run.)

# APPENDIX – B

## DESIGN DOCUMENTATION

### Use Case Diagrams and Narratives

#### 1. Sales Module

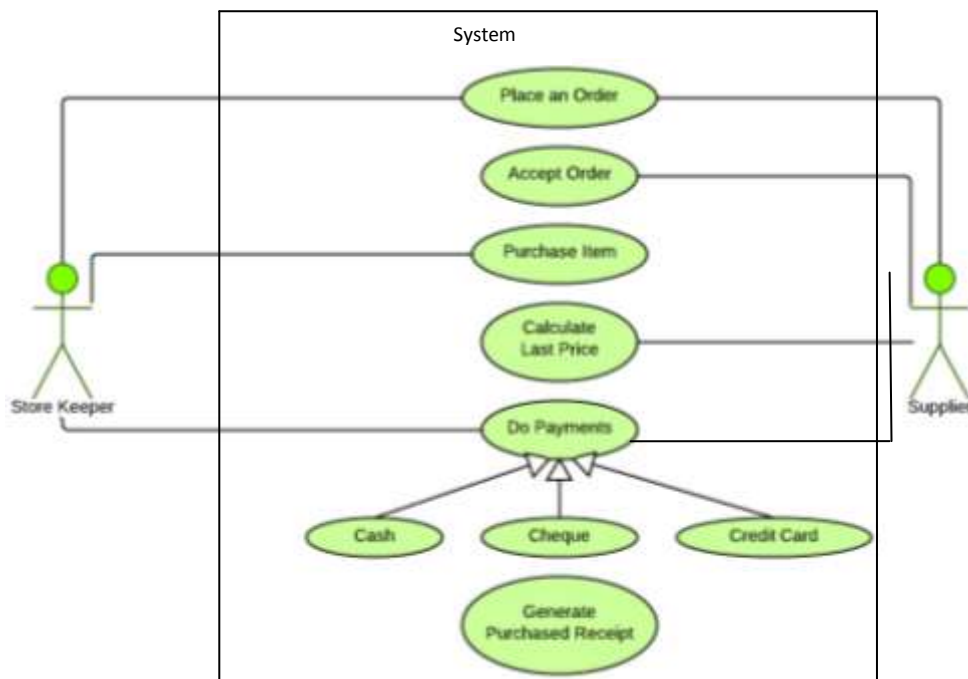


**Figure B.1 Use-Case Diagram for Sales Module**

<b>Use Case</b>	Sales Module
<b>Actors</b>	1) Customer 2) Seller
<b>Description</b>	Customer purchases items and seller sells items while entering transaction details to the system
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Seller should login to the system</li> <li>Ordered items should available in the stock</li> </ul>
<b>Flow of Events</b>	1) Customer place an order 2) Seller provides quotation 3) Seller checks item availability and accept order 4) System calculates last price 5) System generates and print invoice 6) Customer make payment
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>System consist a new sales details.</li> </ul>

**Table B.1- Use-Case Narrative for Sales Module**

## 2) Purchase Module



**Figure B.2 Use-Case Diagram for Purchase Module**

<b>Use Case</b>	Purchase Module
<b>Actors</b>	1) Store Keeper 2) Supplier
<b>Description</b>	Store keeper purchases items from supplier while entering transaction details to the system
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Store keeper should login to the system</li> <li>Supplier details should be entered in o the system</li> </ul>
<b>Flow of Events</b>	1) Store keeper place an order 2) Supplier accepts order 3) Supplier calculates last price 4) Store keeper do the payments 5) Store keeper purchases items 6)Store keeper generate purchased receipt
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>System consists a new purchase details</li> </ul>

**Table B.2- Use-Case Narrative for Purchase Module**



## 3) Manage Customer Module

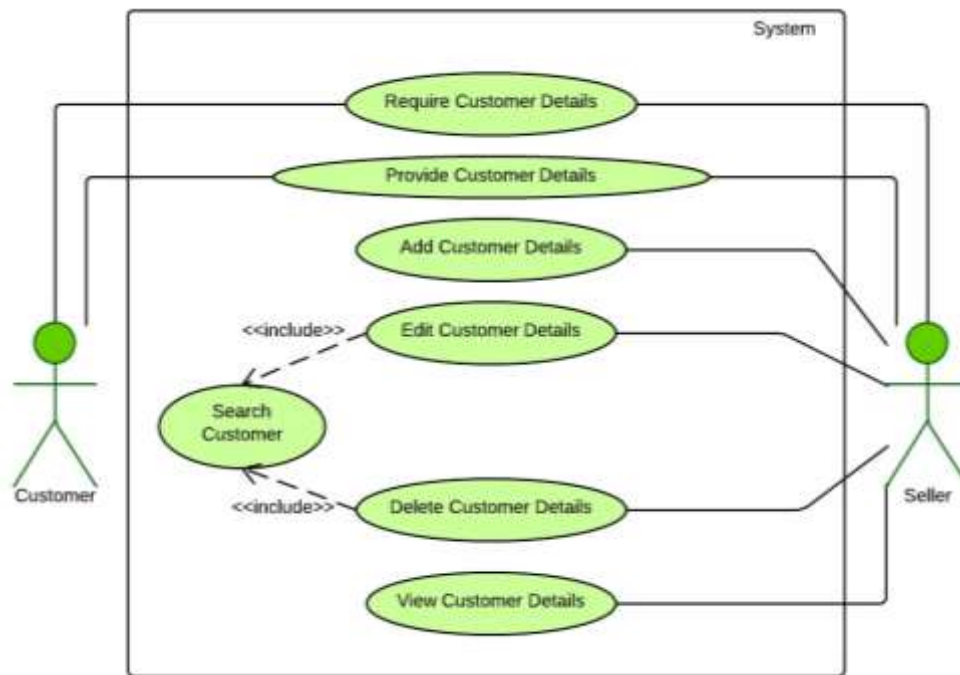
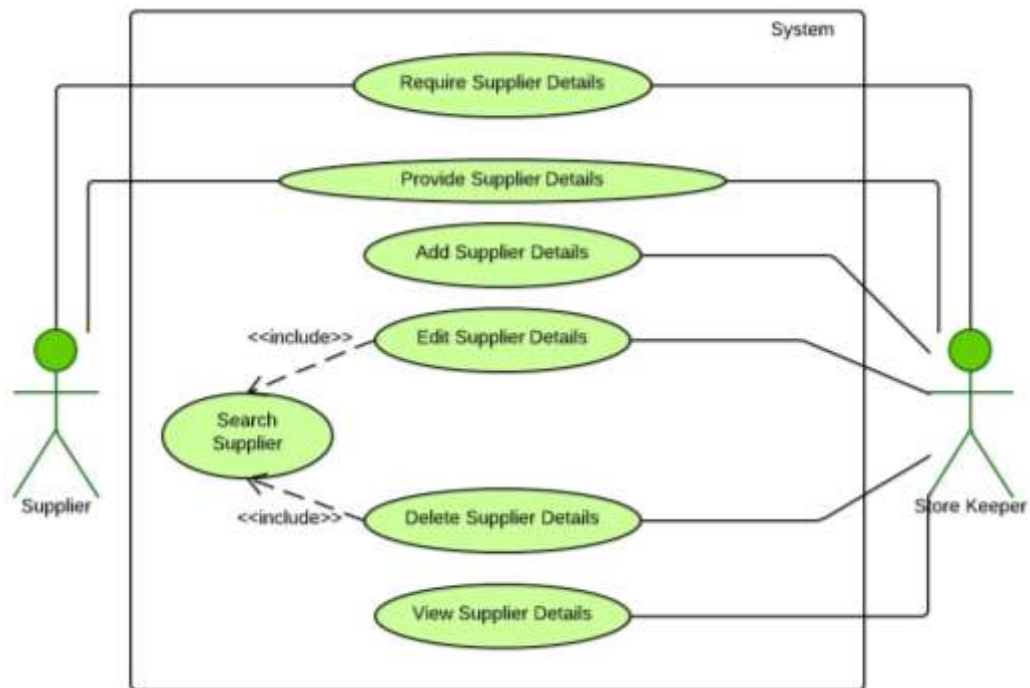


Figure B.3 Use-Case Diagram for Manage Customer Module

<b>Use Case</b>	Manage customer
<b>Actors</b>	1) Customer 2) Seller
<b>Description</b>	Recording of customer details
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>• Seller should login to the system</li> <li>• Customer prefer to provide details</li> </ul>
<b>Flow of Events</b>	1) Seller requires customer details 2) Customer provides details 3) Seller enter customer details to the system 4) Seller save customer details 5) Seller edit customer details 6) Seller view customer details 7) Seller view customer details
<b>Post-Conditions</b>	System consists a new customer detail record

Table B.3- Use-Case Narrative for Manage Customer Module

## 4) Manage Supplier Module



*Figure B.4 Use-Case Diagram for Manage Supplier Module*

<b>Use Case</b>	Manage supplier
<b>Actors</b>	1) Supplier 2) Store keeper
<b>Description</b>	Recording of supplier details
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Store keeper should login to the system</li> <li>Supplier prefer to provide details</li> </ul>
<b>Flow of Events</b>	1) Store keeper requires customer details 2) supplier provides details 3) Store keeper enter customer details to the system 4) Store keeper save customer details 5) Store keeper edit customer details 6) Store keeper view customer details 7) Store keeper view customer details
<b>Post-Conditions</b>	System consists a new supplier detail record

*Table B.4- Use-Case Narrative for Manage Supplier Module*

## 5) Manage Product Module

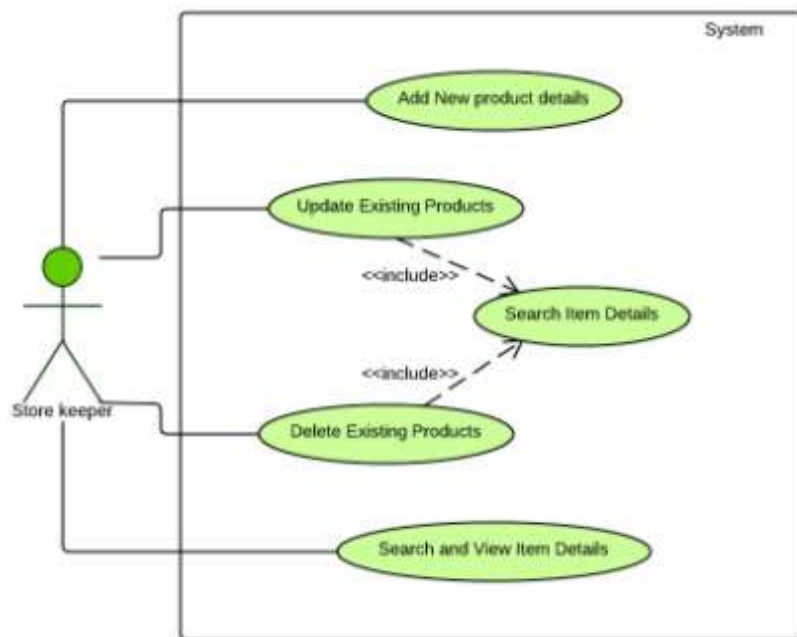
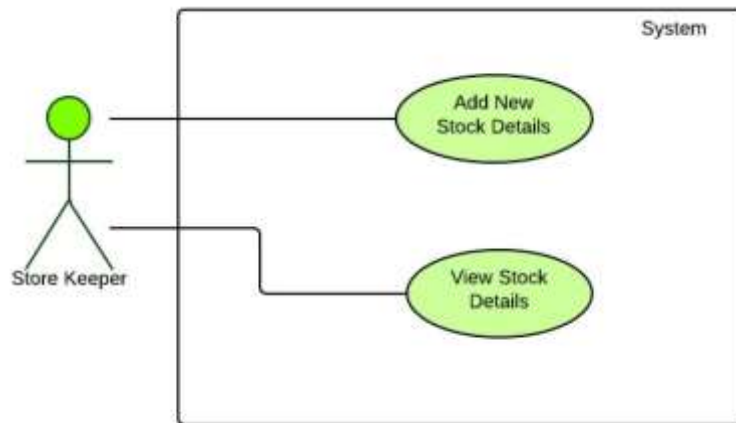


Figure B.5 Use-Case Diagram for Manage Product Module

<b>Use Case</b>	Manage product
<b>Actors</b>	1) Store keeper
<b>Description</b>	Store Keeper add, search, update, view, delete item details
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Store keeper must logging to the system</li> </ul>
<b>Flow of Events</b>	1) Store keeper add product details 2) Store keeper search product details 3) Store keeper update product details 4) Store keeper view product details 5) Store keeper delete product details
<b>Post-Conditions</b>	System consists new product detail records

Table B.5- Use-Case Narrative for Manage Product Module

## 6) Manage Stock Module

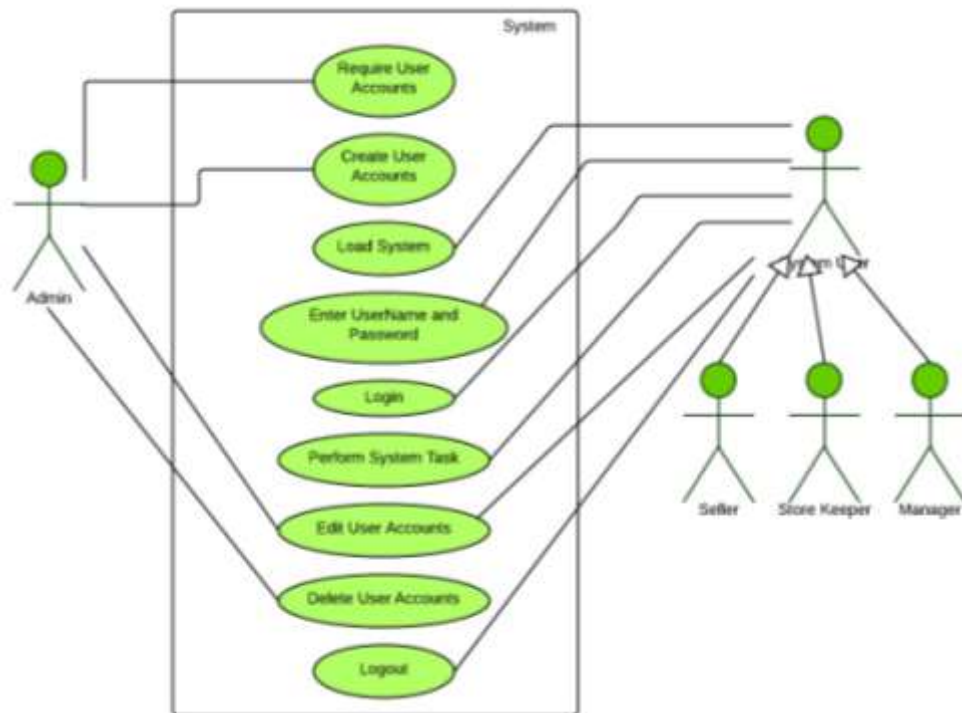


*Figure B.6 Use-Case Diagram for Manage Stock Module*

<b>Use Case</b>	Manage stock
<b>Actors</b>	1) Store keeper
<b>Description</b>	Store Keeper add and view stock details
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Store keeper must logging to the system</li> </ul>
<b>Flow of Events</b>	1) Store keeper add stock 2) Store keeper view product details
<b>Post-Conditions</b>	System consists new stock detail records

*Table B.6- Use-Case Narrative for Manage Stock Module*

## 7) Administration



**Figure B. Use-Case Diagram for Administration**

<b>Use Case</b>	Administration
<b>Actors</b>	1) Administrator 2) Manager 3) Seller 4) Store Keeper
<b>Description</b>	Administrator creates different views of the system to different stake holders and give access permission to the system
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>Administrator should logging to the system</li> <li>New stake holder require a new account</li> <li>Existing stake holder already have an account</li> </ul>
<b>Flow of Events</b>	1) New stake holder require an account 2) Administrator creates an account 3) Stake holder load the system 4) Stake holder enter user name, password and then logging to the system 5) Stake holder edit their accounts 6) Stake holder logout from the system 7) Administrator delete accounts 8) Administrator logout from the system
<b>Post-Conditions</b>	<ul style="list-style-type: none"> <li>New stake holder has an account and has permission to access the system</li> </ul>

**Table B.7- Use-Case Narrative for Administration**

## Activity Diagrams

### 1. Activity diagram for overall sales management module

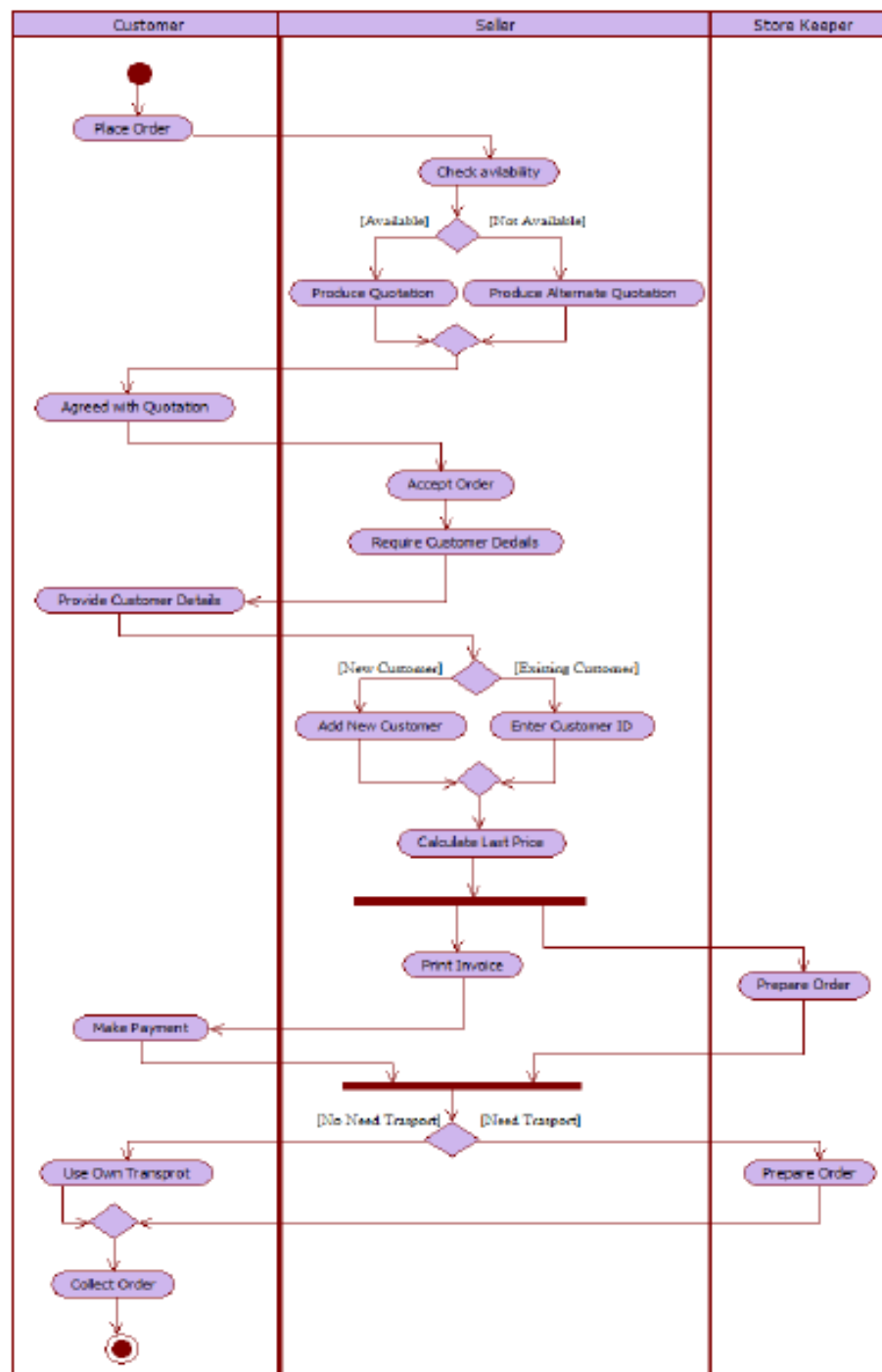
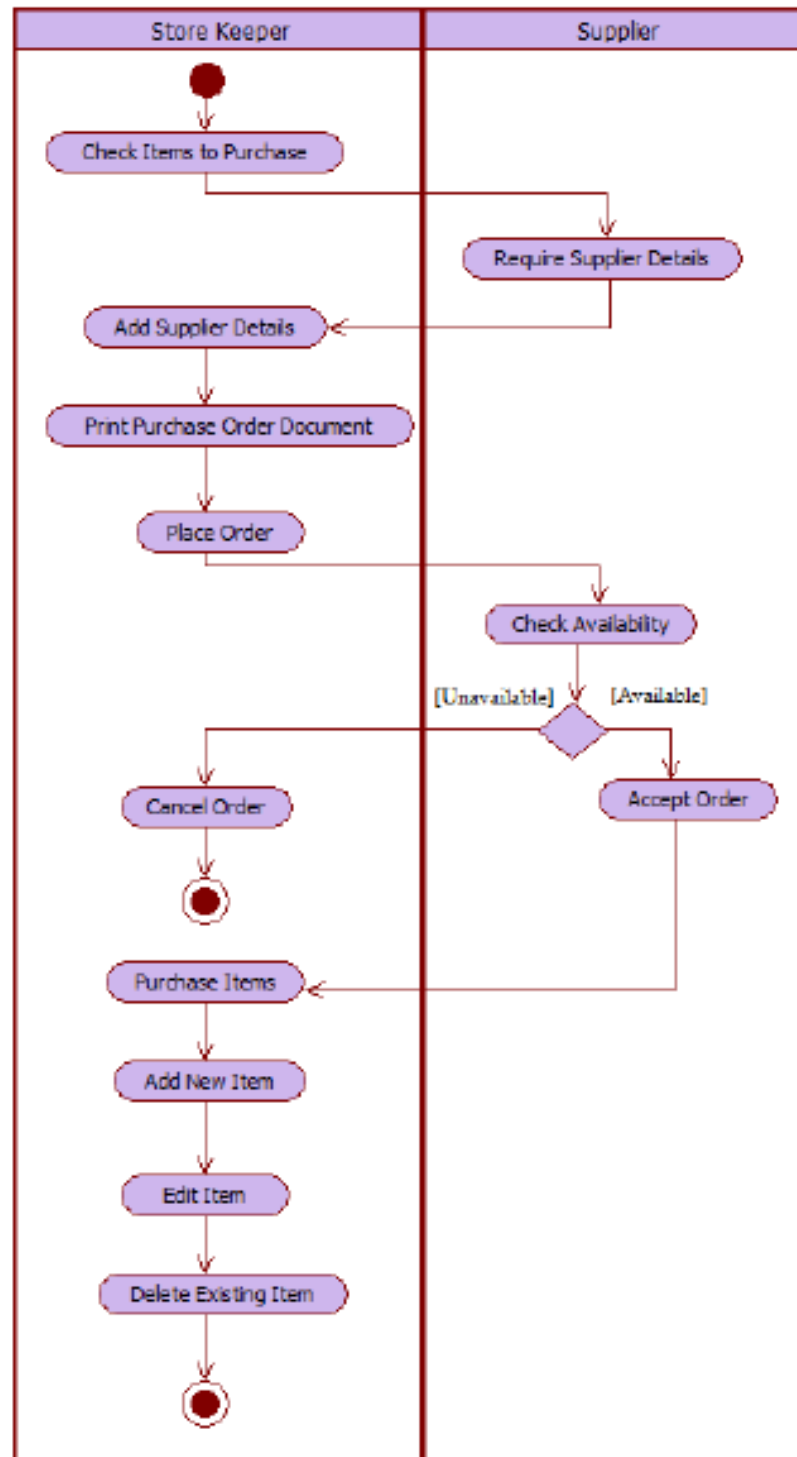


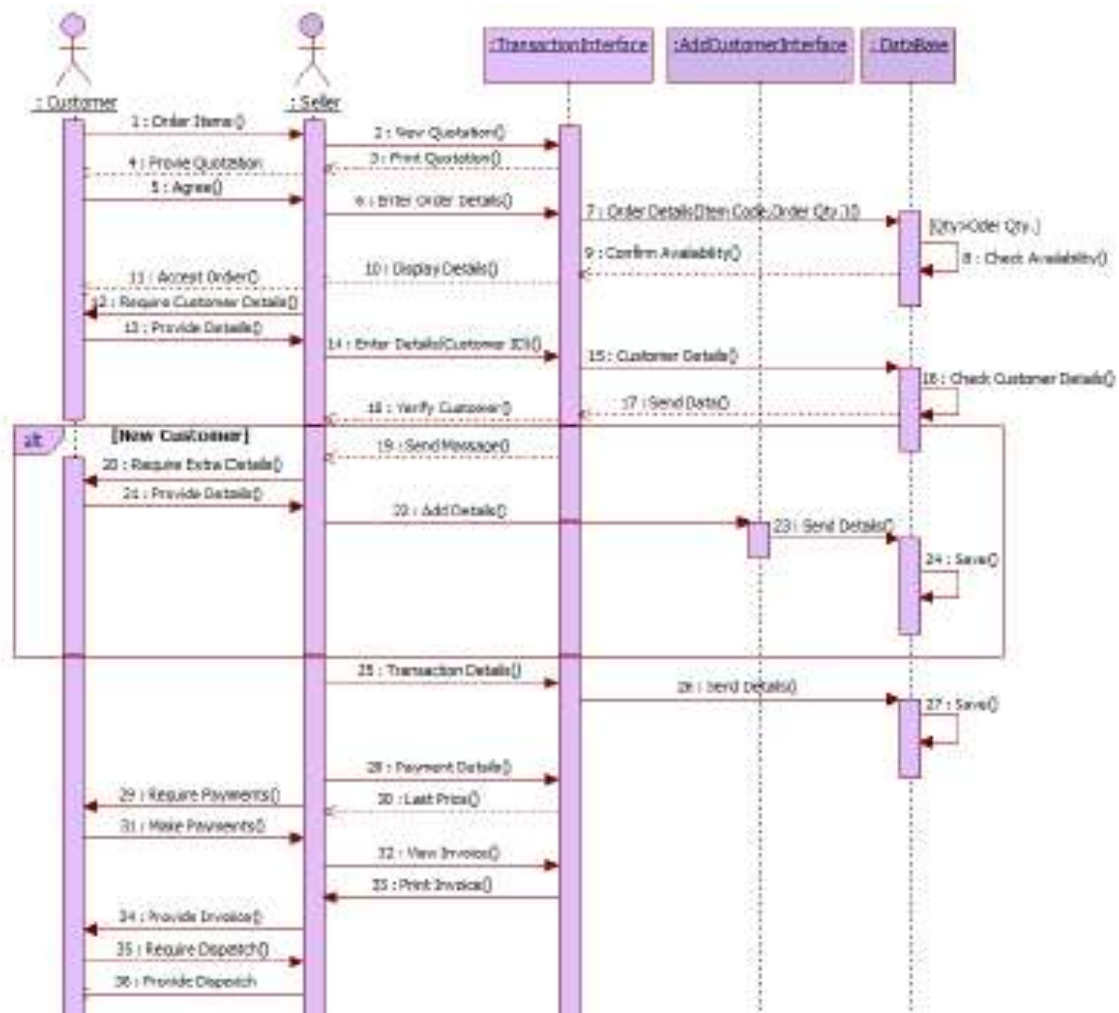
Figure B.8 - Activity diagram for overall sales management module

## 2. Activity Diagram for overall Stock Management Module

*Figure B.9 - Activity diagram for overall stock management module*

## Sequence Diagrams

### 1. Sequence Diagram for overall Sales Management Module



**Figure B.10 - Sequence diagram for overall sales management module**



## 2. Sequence Diagram for overall Sales Management Module

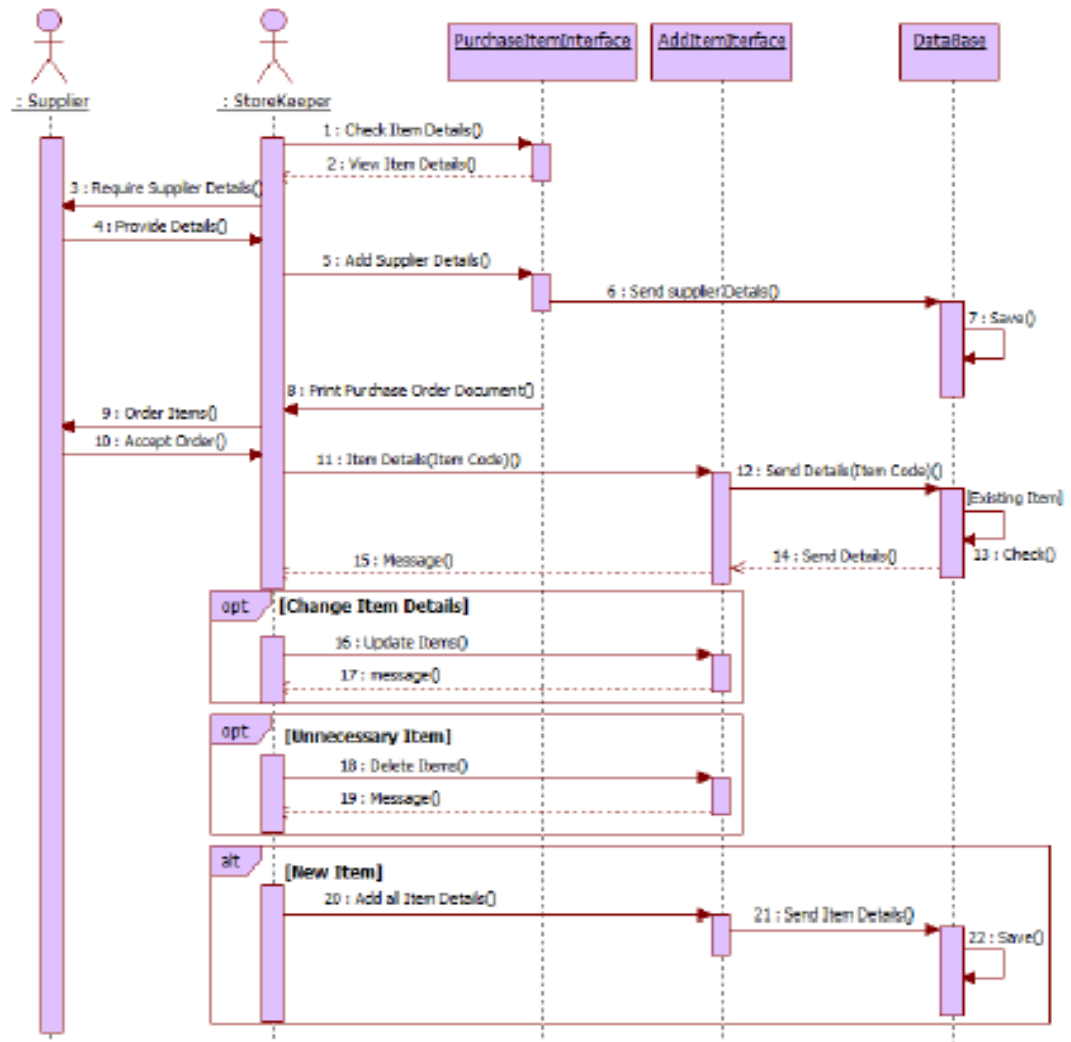


Figure B.11 - Sequence diagram for overall stock management module

## APPENDIX – C

# USER DOCUMENTATION

This documentation is a user guide for the users of the system. Sales and Inventory Management System is developed without complexity and included easily understandable features. Administrator, Manager, Seller and Store Keeper are the users of the system.

### Login screen

Authorized users who already have an account can login to the system. Default user name is 'admin' and password is 'pass'



*Figure C.1 – Login Screen*

### Main Screen

Main screen contains menu bar. When Click on menu items under this menu bar relevant interfaces will appear and then user can interact with these interfaces and perform functions. Main menus of the menu bar are File, Customer, Supplier, Stock, Sales, Purchases, Reports, Account and User role.

By using file menu item, user can change his/her password and exit from the system.

Under customer menu there are two sub menus. By using add customer submenu user can add customer details. By using search customer sub menu user can search customer details as well as delete and edit selected customers

Under supplier menu there are two sub menus. By using add supplier sub menu user can add supplier details to the system. By using search supplier sub menu user can search supplier details as well as edit and delete selected supplier.

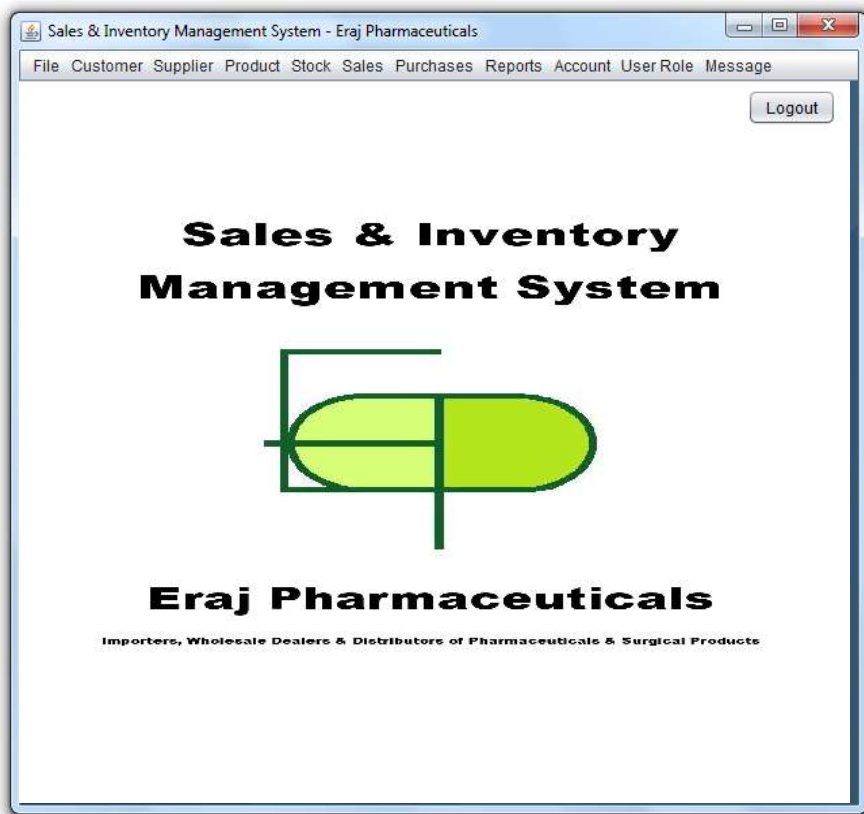
By using stock menu item, user can add new stock details. Sales menu item is using for add sales details and purchase menu item is using for add purchase details to the system.

Reports menu item is using for view reports.

There are two sub menu items under accounts menu item. Create account sub menu item is use for create user accounts and edit user account is for edit and delete user accounts.

Under user role menu item there are two sub menu items. To create a user role user have to use create user role sub menu item and edit user role sub menu item is use for edit and delete user roles.

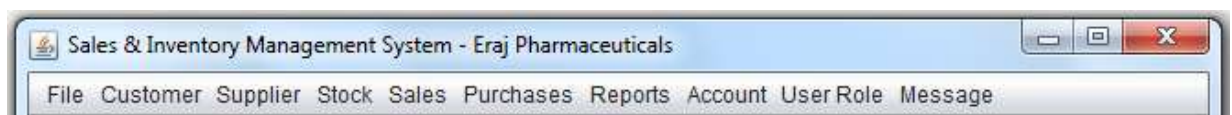
Logout from the system user have to press logout button of the top right corner.



*Figure C.2 – Main Screen*

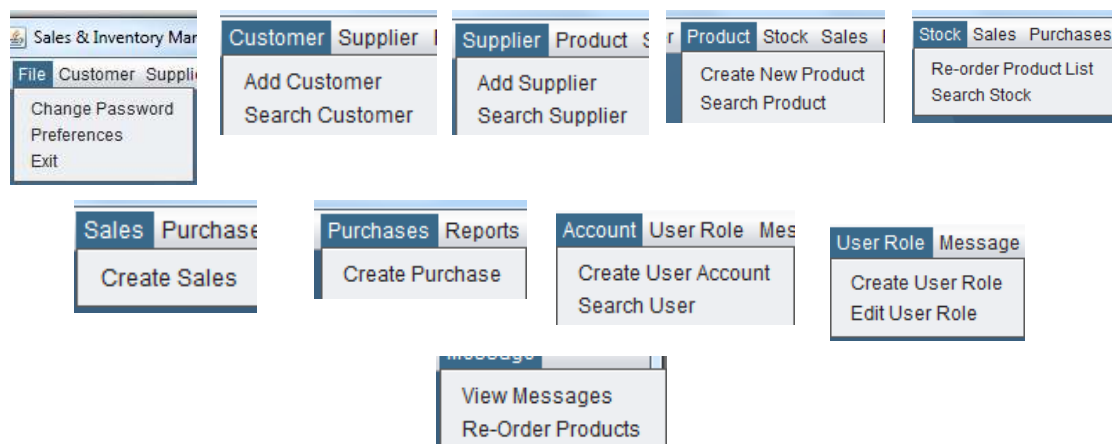
### Menu Items

Important navigations can be done by using menu bar. Following illustrates the menu bar of the system



*Figure C.3 – Menu Items*

## Sub Menu Items



*Figure C.4 – Sub Menu Items*

## Change Password Screen

*Figure C.5 – Change Password Screen*

By clicking change password menu item, it will appear the change password interface. By using this user can change his/her own password. Red stars will indicate the required fields to fill. Those fields cannot be empty

To change the password user have to give the correct information and fill all the required fields and have to click the submit button. Clear button is for clear all text fields.

Cancel button is for existing from the change password window.

## Customer

### Add Customer Details Screen

**Figure C.6 – Add Customer Details Screen**

By clicking add customer menu item it will appear add customer details interface. This interface is use to add customer details to the system. Red stars will indicate required fields to be filled. After filling required fields, user have to click the add button to submit the data. If customer ID, mobile number and email field are different from the given format it will prompt an error message. Home and office number fields should be integers.

If user wants to clear all text fields, he/she have to click the clear button. By clicking close button, user can exit from the add customer details interface.

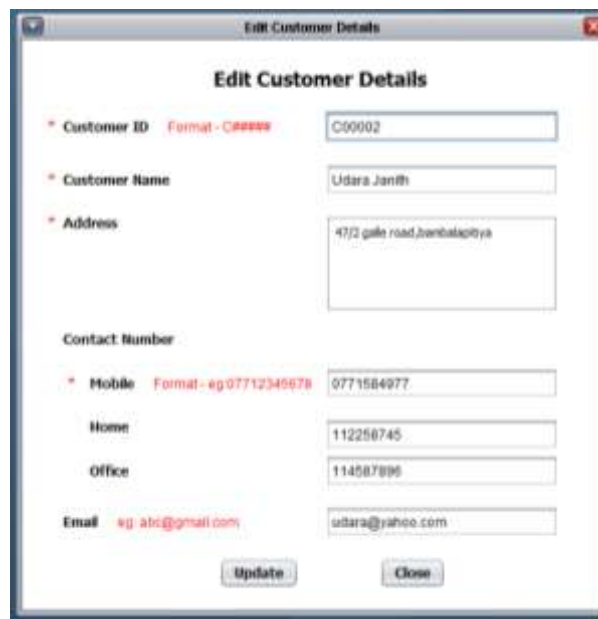
### Customer Search Screen

Customer ID	Name	Address	Mobile	Home	Office	Email
C00001	Heshan Pe...	21/3 Jem...	0771589459	112735895	112789456	hesh@gm...
C00002	Udara Janth...	47/2 galle r...	0771584977	112258745	114567890	udara@yah...
C00003	Dishan de ...	56, temple r...	0773866141	112588769	112987568	dishan@g...
C00004	Nihal Perera	566, camp...	0725877458	112735671	112987568	niha1@yah...
C00005	Anura de Si...	345, katu...	0718334539	112577025	112578890	anura@gm...
C00006	Namal Per...	No.12 Flow...	0773866141			namal@g...
C00007	Sajeewa H...	No 52 Field...	0718334539			s_sajeewa...

**Figure C.7 – Customer Search Screen**

By clicking search customer menu it will appear customer search screen. User can search customer by customer name or his/her phone number or by using both. Then it will shows particular customer details in the table. If user wants to delete a customer he/she have to select the customer from the table and have to click the delete button. Then it will prompt a window to ask for the user confirmation. If user wants to edit any customer detail, he/she should select the particular customer and click edit button. Then it will appear the edit customer interface to do the edits.

### Edit Customer Details Screen



*Figure C.8 – Edit Customer Screen*

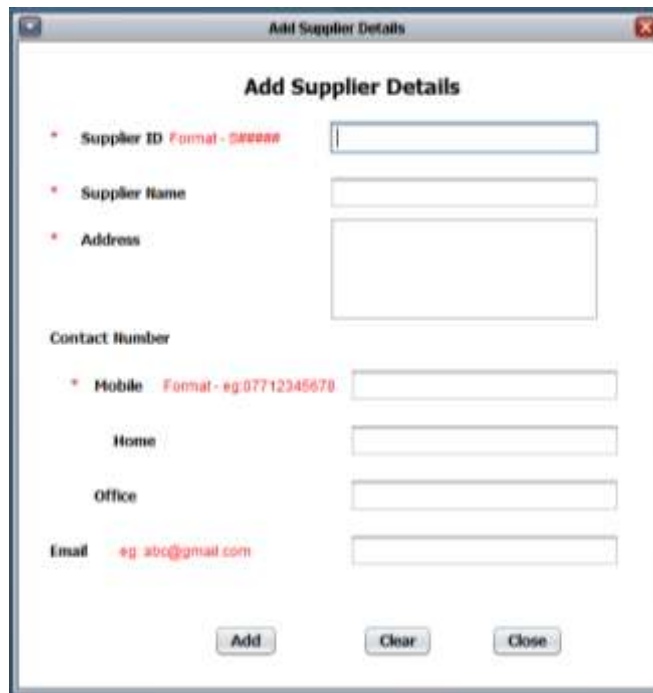
User can edit customer details by using this interface. Red stars will indicate the required fields to be filled. User only cannot edit the customer ID field. After done the editing, have to click the update button to update data to the system. By clicking close button, user can exit from the current window.

## Supplier

### Add Supplier Details Screen

By clicking add supplier menu item it will appear add supplier details interface. This interface is use to add supplier details to the system. Red stars will indicate required fields to be filled. After filling required fields, user have to click the add button to submit the data. If supplier ID, mobile number and email field are different from the given format it will prompt an error message. Home and office number fields should be integers.

If user wants to clear all text fields, he/she have to click the clear button. By clicking close button, user can exit from the add customer details interface.



**Add Supplier Details**

\* Supplier ID Format - 000000

\* Supplier Name

\* Address

Contact Number

\* Mobile Format - eg:07712345670

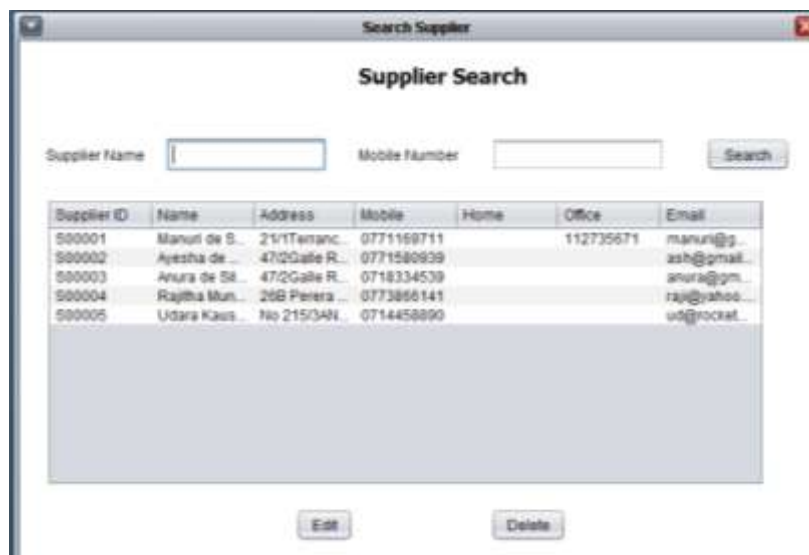
Home

Office

Email eg: abc@gmail.com

**Figure C.9 – Add Supplier Details Screen**

### Supplier Search Screen



**Search Supplier**

**Supplier Search**

Supplier Name  Mobile Number

Supplier ID	Name	Address	Mobile	Home	Office	Email
S00001	Manuri de S...	21/1Terranc...	0771159711		112735671	manuri@g...
S00002	Ajesh de ...	47/2Gate R...	0771580939			ash@gmail...
S00003	Anura de S...	47/2Gate R...	0718334539			anura@gm...
S00004	Rajitha Mun...	20B Patera ...	0773886141			raj@yahoo...
S00005	Udara Kas...	No 215/34N...	0714458890			ud@rocket...

**Figure C.10 – Supplier Search Screen**

By clicking search supplier menu it will appear supplier search screen. User can search supplier by supplier name or his/her phone number or by using both. Then it will shows particular supplier details in the table. If user wants to delete a supplier he/she have to select the supplier from the table and have to click the delete button. Then it will prompt a window to ask for the user confirmation. If user wants to edit any supplier detail, he/she should select the particular supplier and click edit button. Then it will appear the edit supplier interface to do the edits.

## Edit Supplier Details Screen

**Figure C.11 – Supplier Edit Screen**

User can edit supplier details by using this interface. Red stars will indicate the required fields to be filled. User only cannot edit the supplier ID field. After done the editing, have to click the update button to update data to the system. By clicking close button, user can exit from the current window.

## Search Stock Details Screen

To appear this screen user have to select search stock sub menu item under stock menu item.

By using this interface user can search stock details by product code.

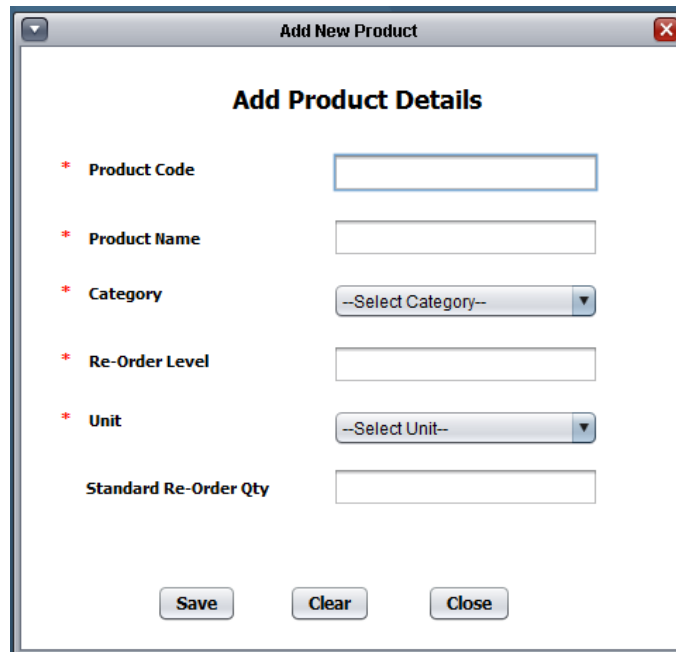
User cannot edit or delete stock details.

Stock ID	Product Code	Supplier ID	Qty	Expiry Date	Purchased Pt.
1a3d743b-89b...	P00006	Ayesha de Silva	250.0	2015-07-04 00	200.0
321c18e8-e8e...	P00002	Manuel de Silva	150.0	2015-06-28 00	150.0
39c1c7b5-c3d...	P00007	Manuel de Silva	10.0	2015-06-28 00	250.0
3ae7c986-462...	P00010	Ayesha de Silva	500.0	2015-07-04 00	100.0
40c9dec3-1f4...	P00006	Ayesha de Silva	1000.0	2015-06-28 00	75.0
ae33c8f8-e48...	P00003	Ayesha de Silva	2000.0	2015-07-04 00	50.0
84e3a74-dcd...	P00004	Manuel de Silva	25.0	2015-06-28 00	150.0
8ae6807-80c...	P00001	Manuel de Silva	2000.0	2015-06-28 00	5.0

**Figure C.12 – Search Stock Details Screen**



## Add Product Details Screen



The image shows a software window titled "Add New Product" with a close button in the top right corner. Inside the window, the title "Add Product Details" is centered. Below the title, there are six input fields, each preceded by a red asterisk indicating it is required. The fields are: "Product Code" (text box), "Product Name" (text box), "Category" (dropdown menu with "--Select Category--" selected), "Re-Order Level" (text box), "Unit" (dropdown menu with "--Select Unit--" selected), and "Standard Re-Order Qty" (text box). At the bottom of the window, there are three buttons: "Save", "Clear", and "Close".

*Figure C.13 – Add Product Details Screen*

By using this interface, user can enter product details in to the system. To appear this interface, user have to click add new product sub menu item under product menu. Red stars are indicating the required fields to be filled. After entering all required data user have to click the save button to add data into the system. Once user adds details into the system he/she cannot do edits or deletes. Clear button for clear all text fields and close button for exit from the system.

## Sales

The screenshot shows a web application window titled "Sales". It contains several sections for data entry:

- Transaction ID:** A text field with the value "3156-45e81d4d910".
- Date:** A text field with the value "2015-7-28".
- Customer Details:**
  - Customer Name:** A dropdown menu with "--Select Customer Name--".
  - Customer ID:** A text field.
- Item Details:**
  - Product Name:** A dropdown menu with "--Select Product--".
  - Product Code:** A text field.
  - Category:** A text field.
  - Discount (%):** A text field.
  - Unit Price (Rs):** A text field.
  - Order Qty:** A text field.
  - Stock Balance:** A text field.
  - Buttons:** "Add" and "Clear" buttons are located below the unit price field.
- Payment Method:** Radio buttons for "Cash" and "Cheque".
- Table:** A table with columns: Product Code, Product Name, Qty, Discount, Price, and Amount. The table is currently empty.
- Bottom Section:**
  - Remove Selected Item:** A button.
  - Net Invoice Value:** A text field.
  - Buttons:** "Save" and "Cancel" buttons.

**Figure C.14 – Sale Screen**

By clicking Create Sales menu item under Sales menu sale interface will appear. This screen is used to handle sales transactions. User has to fill all the fields that indicate with red stars. When user selects the customer system will automatically fill the customer ID. When user selects the product name system will fill product code, category and expire date. Unit price should be entered by user. If giving a discount to the customer user have to enter the percentage into the discount field. Remaining stock balance will be shown in the stock balance field. User has to enter products one by one. To do that user have to press the add button. After adding all items, if user wants to remove any item from the table, have to select the item and click remove selected item button.

To get the net invoice value user has to click the Net Invoice Value button. Then the system will calculate the net invoice value. To add the sales transaction into the system user have to click the Save button. After saving the transaction it will ask to view the invoice. If user wants to view it he/she has to click the yes button and it will show the invoice.

By clicking cancel transaction button transaction will be canceled and all text fields and table will clear.

Once a sales transaction is entered into the system, it will not be able to delete or do edits.

When the product reorder level is reached it will pop up a form asking to send a message to a supplier. User has to select the product that is need to reorder. Then the message will be automatically shown below the message field. If user wants he/she can edit the message. Then have to click the send message to send the message to the supplier.



## User Role

### Add User Role Screen

ID	Name	Code	Description
10	Preferences	SELECT_P	Select Rep.
4	Roles	MANAGE_R	Manage rol.
9	Messages	SEND_ME	Send & rec.
3	Users	MANAGE_U	Manage us.
8	Reports	GENERATE	Generate re.
6	Suppliers	MANAGE_S	Manage su.
7	Products	MANAGE_P	Manage pro.
5	Customers	MANAGE_C	Manage cu.
2	Purchases	DO_PURC	Perform pur.
1	Sales	DO_SALES	Perform sal.

**Figure C.16 –Add User Role Screen**

By clicking add user role sub menu item under user role menu item this interface will be appeared. Using this user can add user roles into the system. Red stars will indicate the required fields to be filled. After entering required data, user have to click the submit button to add data into the system. All text fields will be cleared by clicking clear button. To exit from the system, user has to click cancel button.

### Edit User Role Screen

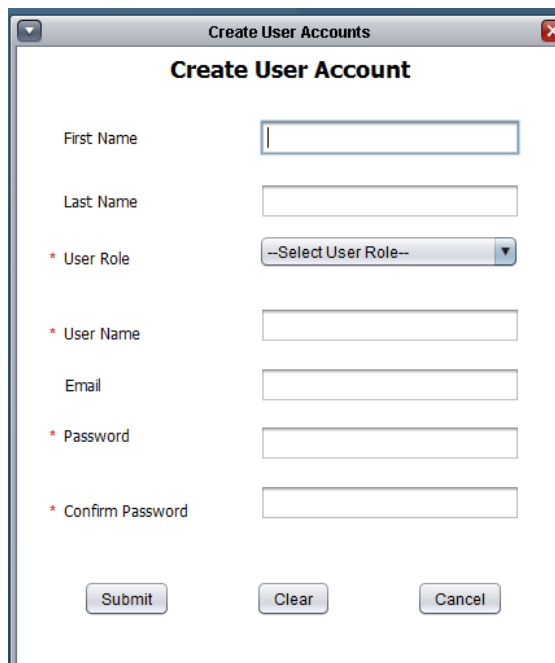
ID	Name	Code	Descr.
10	Prefer.	SELE	Sele.
4	Roles	MANA	Mana.
9	Mess.	SEND	Send
3	Users	MANA	Mana.
8	Reps.	GENE	Gene.
6	Suppl.	MANA	Mana.
7	Produ.	MANA	Mana.
5	Custo.	MANA	Mana.
2	Purch.	DO_P	Perfor.
1	Sales	DO_S	Perfor.

**Figure C.17 –Edit User Role Screen**

By clicking edit user role sub menu item under user role menu this screen will appear. When user selects the user role ID user role name field and description field will be filled automatically. User only cannot edit the ID. After done the editing user have to click the update button to update data into the system. To delete a user role user have to select a user ID and then have to click the delete button. By clicking cancel button user can exit from the current window.

### User Account

#### Create User Account Screen



**Figure C.18 –Create User Account Screen**

By clicking create user account sub menu item under user account menu this interface will be appeared. Using this user can create user accounts. Red stars will indicate the required fields to be filled. After entering required data, user have to click the submit button to add data into the system. All text fields will be cleared by clicking clear button. To exit from the system, user has to click cancel button.

#### Edit User Account Screen

By clicking edit user account sub menu item under user account menu item this screen will appear. When user selects the username user role name field will be filled automatically. User only cannot edit the username. To change the password user have to select the username, enter a new password and re enter that password and then have to click the change password button. To change the user role user have to select the user name and select a new user role and then have to click the change user role button. To delete a user account user have to select a user name and then have to click the delete button. By clicking cancel button user can exit from the current window.



**Figure C.19 –Search User Account Screen**

To appear this screen user have to select search stock sub menu item under stock menu item.

By using this interface user can search stock details by product code.

User cannot edit or delete stock details.

### Search User Account Screen

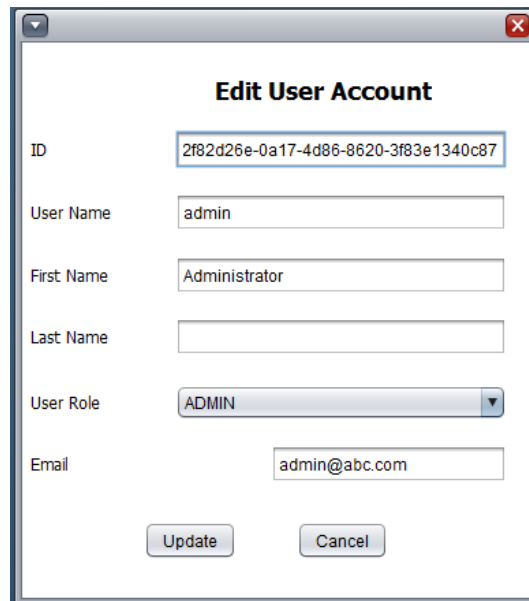
To appear this screen user have to select search user sub menu item under user role menu item.

By using this interface user can search user by username or email address.



**Figure C.20 –Search User Account Screen**

To edit the user account user has to select a username and has to click the edit button. Then the edit user account screen will be appeared. Using that user can edit details except ID and username.

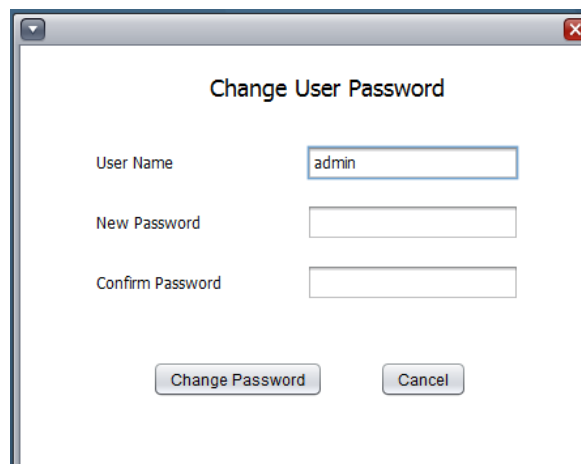


The 'Edit User Account' window contains the following fields and controls:

- ID:** 2f82d26e-0a17-4d86-8620-3f83e1340c87
- User Name:** admin
- First Name:** Administrator
- Last Name:** (empty)
- User Role:** ADMIN (dropdown menu)
- Email:** admin@abc.com
- Buttons:** Update, Cancel

**Figure C.21 –Edit User Account Screen**

To change the password of a user account user has to select a username and then has to click change password button on search user screen. Then the user password screen will be appeared.



The 'Change User Password' window contains the following fields and controls:

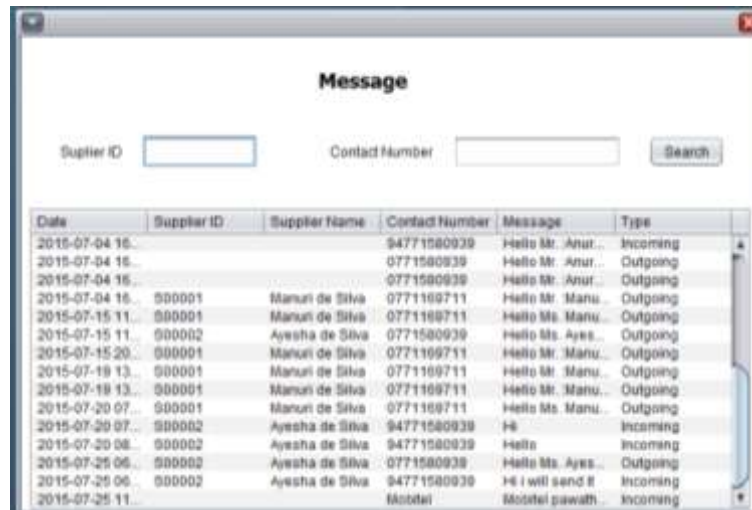
- User Name:** admin
- New Password:** (empty)
- Confirm Password:** (empty)
- Buttons:** Change Password, Cancel

**Figure C.22 –Change User Password Screen**

To delete a user account user has to select a username and then has to click delete button search user account screen.

## Messages Screen

To view all messages user has to select view messages sub menu item under messages menu.

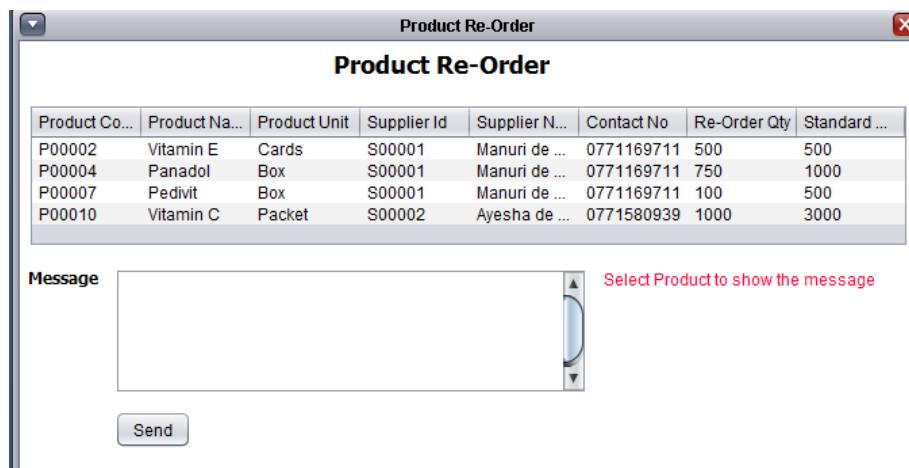


Date	Supplier ID	Supplier Name	Contact Number	Message	Type
2015-07-04 16...			94771580939	Hello Mr. Anur...	Incoming
2015-07-04 16...			0771580939	Hello Mr. Anur...	Outgoing
2015-07-04 16...			0771580939	Hello Mr. Anur...	Outgoing
2015-07-04 16...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-15 11...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-15 11...	S00002	Ayesha de Silva	0771580939	Hello Mr. Ayes...	Outgoing
2015-07-15 20...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-19 13...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-19 13...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-20 07...	S00001	Manuri de Silva	0771169711	Hello Mr. Manu...	Outgoing
2015-07-20 07...	S00002	Ayesha de Silva	94771580939	Hi	Incoming
2015-07-20 08...	S00002	Ayesha de Silva	94771580939	Hello	Incoming
2015-07-25 06...	S00002	Ayesha de Silva	0771580939	Hello Mr. Ayes...	Outgoing
2015-07-25 06...	S00002	Ayesha de Silva	94771580939	Hi I will send it	Incoming
2015-07-25 11...			Moddel	Moddel pawath...	Incoming

*Figure C.23 –All Messages Screen*

## Product Re-Order Message Screen

To send a message to re-order a product user has to select re-order products sub menu under messages menu item. To place an order via sms user has to select a product and then message will be automatically shown in the message field. If user wants he/she can edit the message and has to click send button.



Product Co...	Product Na...	Product Unit	Supplier Id	Supplier N...	Contact No	Re-Order Qty	Standard ...
P00002	Vitamin E	Cards	S00001	Manuri de ...	0771169711	500	500
P00004	Panadol	Box	S00001	Manuri de ...	0771169711	750	1000
P00007	Pedivit	Box	S00001	Manuri de ...	0771169711	100	500
P00010	Vitamin C	Packet	S00002	Ayesha de ...	0771580939	1000	3000

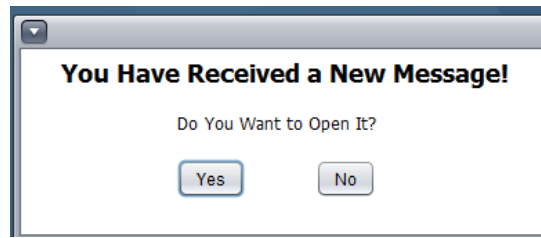
Message Select Product to show the message

Send

*Figure C.24 –Product Re-Order Message Screen*

If a message received system will show a notification. If user wants to view that message he/she has to click the yes button. Then the new message will be shown. If she/he clicked the no button the notification will be disappeared.





*Figure C.25 –Message Received Notification*



*Figure C.26 –New Message Screen*

## APPENDIX – D

### MANAGEMENT REPORTS

#### Customer Wise Sales Analysis Report

This report provides facility to get idea about total purchases which are done by each customer. Bar chart provides information in graphical manner to get idea at a glance. More valuable customers according to their total purchase can identify by using this report.

#### Customer Wise Sales Analysis Report

Customer ID	Item Code	Total Amount (Rs.)
C0001	BCS111011201054	15,000
	BCS131483021054	1,500
	<b>Total Purchase (Rs.) :</b>	<b>16,500</b>
C0002	BCS131483021054	1,500
	BCS271871171104	1,000
	BCS281761011104	3,000
	<b>Total Purchase (Rs.) :</b>	<b>5,500</b>
C0003	BCS121021101054	6,000
	BCS151066061054	2,500
	<b>Total Purchase (Rs.) :</b>	<b>8,500</b>
C0004	BCS124578931054	900
	BCS141061081054	12,000
	BCS161811001054	3,000
	<b>Total Purchase (Rs.) :</b>	<b>15,900</b>

*Figure D.1 - Custom Wise Analysis Report*

## Slow Moving Items Report


Items not issued for last three months are depicts in Slow Moving Items Report. This is a more use full report for managements to take a decision about purchasing of items.

### Slow Moving Items Report

item code	stock balance	purchase price	Last Date of Issue
BCS101268257353	2	35,000.00	1/1/2012
BCS111011201054	8	10,000.00	4/5/2012
BCS121021101054	19	4,500.00	4/7/2012
BCS131483021054	48	1,100.00	3/20/2012
BCS151066061054	19	1,200.00	4/7/2012
BCS271871171104	19	800.00	3/20/2012
BCS281761011104	48	1,000.00	3/20/2012
BCS346561086934	8	800.00	1/1/2012
BCS453287931054	6	3,800.00	1/1/2012

*Figure D.2 - Slow Moving Item Report*

## Invoice



## Eraj Pharmaceuticals

Importers, Wholesale Dealers & Distributors of Pharmaceutical & Surgical Products

No: 96-1/10 Consistory Building, Front Street, Colombo 11, Sri Lanka.

Tel: 011-2434554, 011-3075106, 011-5855714, 011-5219138, 011-5219137

---

Invoice To: Nihal Perera

Invoice No 00001

Date 10/07/2014

<u>Product Code</u>	<u>Description</u>	<u>Qty</u>	<u>(%)</u>	<u>Unit Price</u>	<u>Discount</u>	<u>Amount</u>
EQ03	Equal 300 TAB(6%)	300	6	5.10	91.80	1530.00
OBMI05T	Obmet TAB(15%)	200	15	3.20	96.00	640.00
ARL-10	Atorlip 10(15%)	50	15	12.00	90.00	600.00
PIOT	PIOTAZ-30(15%)	100	15	3.75	56.25	375.00

---

Total No Of Items 4	Total Value	3145.00
	(-)Discount	334.05
	Net Invoice Value	2810.25

---






*Note: Please verify item, Price and Quantity. No returns will be accepted after the goods are cleared from the premises*



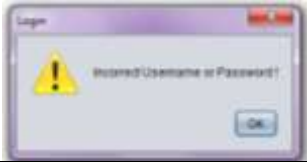
Figure D.3 - Invoice

# APPENDIX – E

## TEST RESULTS




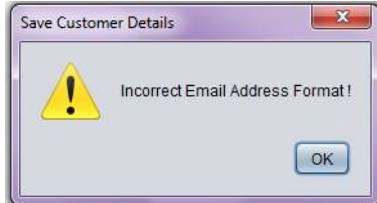
### Test Results for Login Module



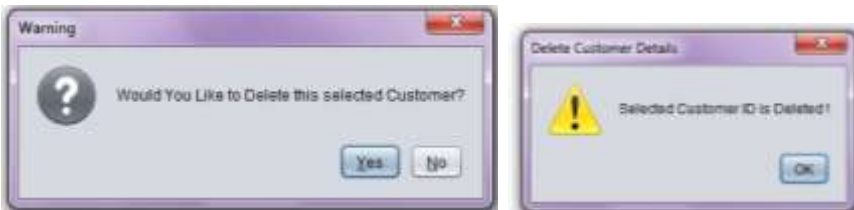

ID	Test Data Inputs and Received Data	Status
1	Username is not entered 	√
5	Confirm password is not match with previous entered password 	√
6	Entered already exist username 	√
7	Click submit button to save account 	√
8	Click cancel button Clear all text boxes	√
13	Click change button to change password 	√

14	Click delete button to delete an account and confirms as “Yes”	 	√
15	User enter wrong username and correct password		√
18	User enter correct username and correct password	Login to the system	√
19	User click cancel button	Clear all text boxes	√

**Table E.1 – Test Results for Login Module**

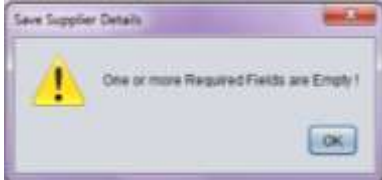

## Test Results for Manage Customer Module







ID	Test Data Inputs and Received Data	Status
1	Customer ID is not entered 	√
5	Customer ID format is incorrect Prompt as wrong format 	√
6	Customer mobile number format is incorrect 	√
7	Customer email address format is incorrect 	√
8	Customer ID is already exit	√

		
9	<p>Click add button</p> 	√
10	Click cancel button - Exit from the window	√
11	Click clear button - Clear all text fields	√
12	Click search button - View all customer details	√
13	Enter customer name and click search button - View relevant details	√
14	Enter mobile number and click search button - View relevant details	√
15	<p>Select a row and click delete button and confirm as “Yes”</p> 	√
15	Click edit button - Goes to edit customer window	√
16	<p>Enter required details and click update button</p> 	√

*Table E.2 – Test Results for Customer Module*

## Test Results for Manage Supplier Module

ID	Test Data Inputs and Received Data	Status
1	<p>Supplier ID is not entered</p> 	√
5	<p>Supplier ID format is incorrect</p> 	√

6	Supplier mobile number format is incorrect 	√
7	Supplier email address format is incorrect 	√
8	Supplier ID is already exist 	√
9	Click add button 	√
10	Click cancel button - Exit from the window	√
11	Click clear button - Clear all text fields	√
12	Click search button - View all customer details	√
13	Enter supplier name and click search button - View relevant details	√
14	Enter mobile number and click search button - View relevant details	√
15	Click delete button on the selected row in table 	√
16	Click edit button - Goes to edit supplier window	√
17	Enter required details and click update button 	√

**Table E.3 – Test Results for Manage Supplier Module**



## Test Results for Manage Product Module


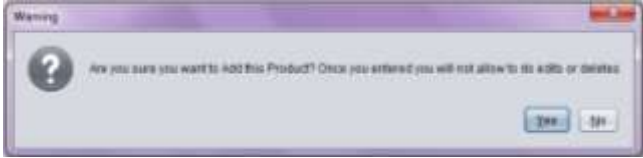

ID	Test Data Inputs and Received Data	Status
1	Product code is not entered 	✓
6	Click save button  	✓
7	Click clear button - Clear all text fields	✓
8	Click close button - Exit from the window	✓

Table E.4 – Test Results for Manage Product Module

## Test Results for Manage Stock Module


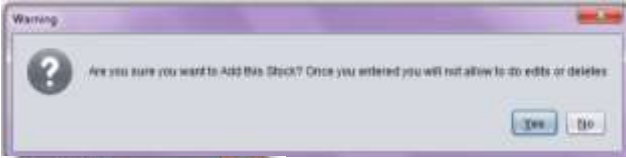

ID	Test Data Inputs and Received Data	Status
1	Stock ID is not entered 	✓
8	Click save button  	✓
9	Click clear button - Clear all text fields	✓
10	Click close button - Exit from the window	✓

Table E.5 – Test Results for Manage Stock Module

## Test Results for Purchase Module

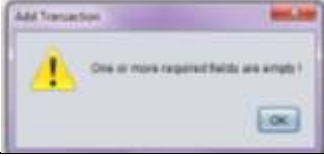

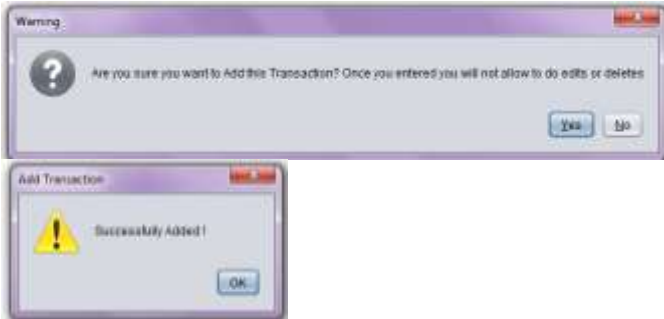
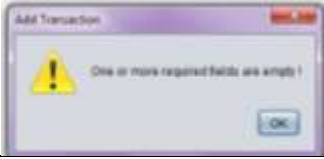

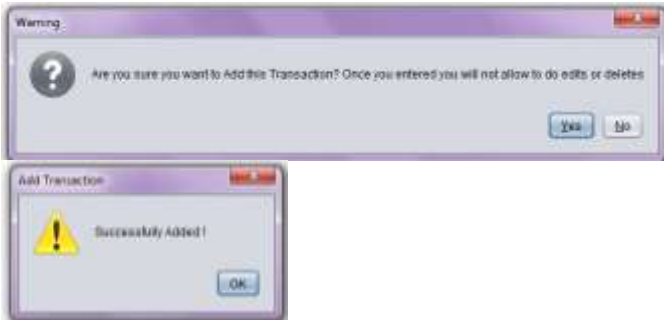
ID	Test Data Inputs and Received Data	Status
1	Transaction ID is not entered 	✓
5	Select a supplier name - Other fields of supplier details fills with relevant data	✓
8	Select a product name - Other fields of product details fills with relevant data	✓
10	Press Enter button - Gets the amount	✓
12	Click add button - Add data to the data table	✓
13	Click total button - Gives the total purchased value	✓
15	Select a row and click remove selected item button 	✓
16	Click add transaction 	✓
17	Click cancel transaction button Cancel the transaction and clear the table	✓
18	Click view purchased receipt View the purchased receipt	✓
19	Click view older purchased receipts View older purchase receipts	✓
20	Click close button Exits from the window	✓

Table E.6 – Test Results for Purchase Module

## Test Results for Sale Module

ID	Test Data Inputs and Received Data	Status
1	Transaction ID is not entered 	✓
5	Select a customer name - Other fields of customer details fills with relevant data	✓
8	Select a product name - Other fields of product details fills with relevant data	✓
10	Press Enter button - Calculates the price	✓
12	Click add button - Add data to the data table	✓
13	Click net invoice button - Gives the total invoice value	✓
15	Select a row and click remove selected item button 	✓
16	Click add transaction 	✓
17	Click cancel transaction button - Cancel the transaction and clear the table	✓
18	Click view invoice - View the invoice	✓
19	Click view older invoice - View older invoices	✓
20	Click close button - Exits from the window	✓

*Table E.7 – Test Results for Sale Module*

## APPENDIX – F

### CODE LISTING

Summarized overview of code segments are included in this appendix, other than sample code segments included in Chapter 4 – Implementation. Due to the length of system coding, only some important code segments are included here. Please refer read-only CD for the complete source code.

#### Manage Customer

##### Edit customer details

```
public class EditCustomer extends javax.swing.JFrame {
    MasterService masterService = new MasterServiceImpl();
    Validator val = new Validator();
    /**
     * Creates new form EditCustomer
     */
    public EditCustomer(Customer c) {
        initComponents();
        //set data to relevant fields
        txtCustomerId.setText(c.getId());
        txtCustomerName.setText(c.getName());
        txtCustomerMobile.setText(c.getMobile());
        ftxtCustomerHome.setText(c.getHome());
        ftxtCustomerOffice.setText(c.getOffice());
        txtCustomerEmail.setText(c.getEmail());
        txtCustomerAddress.setText(c.getAddress());
    }
    //sets mobile number field length to 10 digits
    private void txtCustomerMobileKeyTyped(java.awt.event.KeyEvent evt) {
        if (txtCustomerMobile.getText().length() == 10) {
            evt.consume();
        }
    }
    private void btnCustomerUpdateActionPerformed(java.awt.event.ActionEvent evt) {

        if (txtCustomerId.getText().trim().equals("") ||
            txtCustomerName.getText().trim().equals("") ||
            txtCustomerAddress.getText().trim().equals("") ||
            txtCustomerMobile.getText().trim().equals("")) //checks whether all required fields are filled
        {
            JOptionPane.showMessageDialog(null, "One or more Required Fields are Empty !",
                "Save Customer Details", 2);
            return;
        }
    }
}
```

```
        } else {
            if (val.validateCustomerId(txtCustomerId.getText().trim())//validates customer ID
format
                { if (val.validatePhoneNumber(txtCustomerMobile.getText().trim())//validates
customer mobile number format
                    { if (txtCustomerEmail.getText().trim().equals("") ||
val.validEmail(txtCustomerEmail.getText().trim())) {
                        try {} catch (Exception x) {
JOptionPane.showMessageDialog(null, "Error Occured !", "Save Customer Details", 2);
                            return; } }
                        else {JOptionPane.showMessageDialog(null, "Incorrect Email Address Format !", "Save
Customer Details", 2); txtCustomerEmail.requestFocus(); return; } }
                        else { JOptionPane.showMessageDialog(null, "Incorrect Mobile Number Format !", "Save
Customer Details", 2); txtCustomerMobile.requestFocus(); return; } }
                        else {
                            JOptionPane.showMessageDialog(null, "Incorrect Customer ID Format", "Save
Customer Details", 2);
                            txtCustomerId.requestFocus();
                            return;
                        }
                    }
                }
            }

//update database

        com.abc.salesinventory.model.newpackage.Customer customer = new
com.abc.salesinventory.model.newpackage.Customer();
        customer.setId(txtCustomerId.getText().trim());
        customer.setName(txtCustomerName.getText().trim());
        customer.setMobile(txtCustomerMobile.getText().trim());
        customer.setHome(ftxtCustomerHome.getText().trim());
        customer.setOffice(ftxtCustomerOffice.getText().trim());
        customer.setEmail(txtCustomerEmail.getText().trim());
        customer.setAddress(txtCustomerAddress.getText().trim());
        try {
            masterService.saveOrUpdateCustomer(customer);
            JOptionPane.showMessageDialog(null, "Successfully Updated", "Edit Customer
Details", 2);
            this.setVisible(false);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, e);
        }
    }

//Close button
    private void btnCustomerCloseActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(1);
    }
```

## Manage Supplier

### Add Supplier Details

```

public class CreateSupplier extends javax.swing.JFrame {

    /**
     * Creates new form CreateSupplier
     */
    MasterService masterService = new MasterServiceImpl();
    Validator val = new Validator();

    public CreateSupplier() {
        initComponents();
        Loading();
        masterService.getAllSuppliers();
    }
    private void btnSupplierAddActionPerformed(java.awt.event.ActionEvent evt) {
        if(txtSupplierId.getText().trim().equals("") || txtSupplierName.getText().trim().equals("")
        || txtSupplierAddress.getText().trim().equals("")||
        txtSupplierMobile.getText().trim().equals(""))
            //checks whether all required fields are filled
            { JOptionPane.showMessageDialog(null, "One or more Required Fields are Empty !",
            "Save Supplier Details", 2); return; }
        else
            { if(val.validateSupplierId(txtSupplierId.getText().trim()))//validates supplier ID format
                {if(val.validatePhoneNumber(txtSupplierMobile.getText().trim()))//validates supplier
                mobile number format
                {
                    if(txtSupplierEmail.getText().trim().equals("")||val.validEmail(txtSupplierEmail.getText().trim
                    ())))//validates supplier email format
                        { try {Supplier supplier =
                        masterService.getSupplier(txtSupplierId.getText().trim()); if(supplier != null){
                        JOptionPane.showMessageDialog(null, "Supplier Already Exists !", "Save Supplier Details",
                        2); return; // Supplier ID is already exists. } }
                        catch(Exception x)
                        {JOptionPane.showMessageDialog(null, "Error Occured !", "Save Supplier
                        Details", 2);} }
                        else{JOptionPane.showMessageDialog(null, "Incorrect Email Address Format !", "Save
                        Supplier Details", 2); txtSupplierEmail.requestFocus(); return;}}
                        else{ JOptionPane.showMessageDialog(null, "Incorrect Mobile Number Format
                        !", "Save Supplier Details", 2); txtSupplierMobile.requestFocus(); return;} }
                        else{JOptionPane.showMessageDialog(null, "Incorrect Supplier ID Format", "Save
                        Supplier Details", 2); txtSupplierId.requestFocus(); return; } }
                        com.abc.salesinventory.model.newpackage.Supplier supplier = new
                        com.abc.salesinventory.model.newpackage.Supplier();
                        supplier.setId(txtSupplierId.getText().trim());
                        supplier.setName(txtSupplierName.getText().trim());
                        supplier.setAddress(txtSupplierAddress.getText().trim());
                        supplier.setMobile(txtSupplierMobile.getText().trim());

```

```

supplier.setHome(ftxtSupplierHome.getText().trim());
supplier.setOffice(ftxtSupplierOffice.getText().trim());
supplier.setEmail(txtSupplierEmail.getText().trim());
try {masterService.saveOrUpdateSupplier(supplier);
OptionPane.showMessageDialog(null, "Successfully Added", "Save Supplier Details", 2);
Loading();
} catch (Exception e) {
OptionPane.showMessageDialog(rootPane, e); } }

//set mobile number field length to 10 digits
private void txtSupplierMobileKeyTyped(java.awt.event.KeyEvent evt) {
    if(txtSupplierMobile.getText().length()==10)
    {
        evt.consume();
    }
}

private void btnSupplierClearActionPerformed(java.awt.event.ActionEvent evt) {
    Loading(); }

private void btnSupplierCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(1);
}

private void txtSupplierIdKeyTyped(java.awt.event.KeyEvent evt) {
    if(txtSupplierId.getText().length()==6)
    {
        evt.consume();
    }
}

```

## Search Supplier

```

public class SearchSupplier extends javax.swing.JFrame {
    MasterService masterService = new MasterServiceImpl();
    /**
     * Creates new form SearchSupplier
     */
    public SearchSupplier() {
        initComponents();
        txtSupplierName.requestFocus();
    }
    //method for search supplier
    private void searchSupplier (String hql) {
        try {
            Session session = HibernateUtil.getSessionFactory().openSession();
            session.beginTransaction();
            Query q = session.createQuery(hql);
            List resultList = q.list();

```

```

        displayResult(resultList);
        session.getTransaction().commit();
    } catch (HibernateException he) {
        he.printStackTrace();
    }
}

//method for display results
private void displayResult(List resultList) {

    Vector<String> tableHeaders = new Vector<String>();
    tableHeaders.add("id");
    tableHeaders.add("name");
    tableHeaders.add("address");
    tableHeaders.add("mobile");
    tableHeaders.add("home");
    tableHeaders.add("office");
    tableHeaders.add("email");

    Vector tableData = new Vector();
    for (Object o : resultList) {

        Supplier supplier = (Supplier) o;
        Vector<Object> oneRow = new Vector<Object>();
        oneRow.add(supplier.getId());
        oneRow.add(supplier.getName());
        oneRow.add(supplier.getAddress());
        oneRow.add(supplier.getMobile());
        oneRow.add(supplier.getHome());
        oneRow.add(supplier.getOffice());
        oneRow.add(supplier.getEmail());
        tableData.add(oneRow);
    } resultTable.setModel(new DefaultTableModel(tableData, tableHeaders));
}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    search();
}

//Delete Supplier details
private void btnDeleteSupplierActionPerformed(java.awt.event.ActionEvent evt) {
    int row = resultTable.getSelectedRow();
    DefaultTableModel model = (DefaultTableModel) resultTable.getModel();
    Vector dataModel = model.getDataVector(); Supplier supplier = new Supplier();
    Iterator it = dataModel.iterator(); int x = 0; while (it.hasNext()) { Vector vector =
(Vector) it.next(); if (x == row) { supplier.setId((String) vector.get(0)); break; } x++; }
    int dialogButton = JOptionPane.YES_NO_OPTION;
    int dialogResult = JOptionPane.showConfirmDialog(null, "Would You Like to Delete
this selected Supplier?", "Warning", dialogButton);
    if (dialogResult == JOptionPane.YES_OPTION) {
        masterService.removeSupplier(supplier); search();
        JOptionPane.showMessageDialog(null, "Selected Supplier ID is Deleted !", "Delete
Customer Details", 2); }
}

```



# APPENDIX - G

## CLIENT CERTIFICATE



**Eraj Pharmaceuticals**

**Importers, Wholesale Dealers & Distribution of Pharmaceuticals  
& Surgical Products**

No: 96-1/10 Consistory Building, Front Street, Colombo 11, Sri Lanka  
Tel: 011-2434553, 011-3075106, 011-5855714, 011-5219138, 011-5219137  
Fax: (0094)2434553  
Email: erajmedi@hotmail.com

---

Project Examination  
External Degree Center,  
University of Colombo School of Computing,  
No. 221/2A,  
Dharmapala Mawatha,  
Colombo 07.

July 14, 2014

Dear Sir/ Madam,

Letter to Certification

This is to certify that Miss A.M.A De Silva who is an undergraduate at University of Colombo School of Computing has successfully developed and delivered the proposed Sales and Inventory Management System. The system has been developed as the purpose of completion of the Bachelor of Information Technology Degree Program.

The Sales and Inventory Management System fulfills our company requirements and we believe that the system will provide us a great contribution to perform our sales and inventory functions easily and effectively.

Thank you.

Yours Sincerely,  
**ERAJ PHARMACEUTICALS**  


---

**Proprietor**  
Mr. R. N. E. C De Silva  
Proprietor

Eraj Pharmaceuticals  
**ERAJ PHARMACEUTICALS**  
No. 96-1/10, Front Street,  
Consistory Building  
(Former Siedlos Building)  
COLOMBO - 11.

# GLOSSARY

- **Black Box Testing** – This testing ignores the internal structure of the system and focus on output generated according to any input.
- **Database** – Collections of organized data that are accessed by user.
- **E-R Diagram** – The diagram which is used to represent database design graphically.
- **Feasibility Study** – The process of analyzing feasibility to develop the proposed system.
- **Graphical User Interface (GUI)** – Interfaces that are use to graphic controls to interact with the system.
- **Normalization** – The process of split tables to eliminate data redundancy.
- **Object Oriented Designing** – This is the methodology of designing proposed system that uses object to design the system.
- **Rational Unified Process (RUP)** – is an iterative software development process framework created by the Rational Software Corporation
- **Unified Modeling Language (UML)** – Object Oriented Modeling language which is used to object oriented designing of proposed system.
- **Validation** – The process of evaluate whether the system satisfy the stakeholder requirements.
- **Verification** – The process of evaluate the system implement the required function correctly.

**White box Testing** – This testing takes into account the internal structure of the system or component and compares actual result and expected result.

# INDEX

<b>1</b>		<b>G</b>	
1 <sup>st</sup> Normal Form -----	18	Group-----	15,22,30
<b>2</b>		<b>H</b>	
2 <sup>nd</sup> Norma Form -----	18	Hardware requirements -----	vi,4,43
<b>3</b>		<b>I</b>	
3 <sup>rd</sup> Normal Form -----	18	Implementation environment -----	4,38
<b>A</b>		Install-----	9,43
Acceptance testing -----	31,38	Integration testing -----	31
Activity diagram -----	51,52	IT -----	1
Analysis -----	5,10,11,15,20,37,40,41,71	<b>J</b>	
Analysis phase -----	11,20,37	Jasper Report-----	iii,22,38,42
<b>B</b>		Java-----	iii,21,22,38,41,43
BIT -----	i,iv,2,38	<b>M</b>	
Black box testing -----	29,87	Menu bar -----	55,56
Button-----	19,32,33,35,56,58,60,63,66,74,77,80	Menu items-----	vii,55,56,57,58,59,62,64,65,66
<b>C</b>		MySQL Server -----	38,43
Class diagram -----	vi,viii,16	Microsoft windows 7-----	20
Completeness checks -----	10,37	<b>N</b>	
Conclusion -----	vii,4,37	Navigations -----	56
<b>D</b>		<b>O</b>	
Database-----	iii,vi,2,4,11,18,21,22,32,33,38,41,43,87	Object oriented designing---	vi,15,37,87
Database normalization -----	vi,18	Object oriented Modeling -----	87
Development Process -----	4,11,12,14,87	Observation	
Dissertation -----	i,v,38	<b>P</b>	
<b>E</b>		Processing time-----	9
Email -----	32,33,58,59,75,77	Process models -----	11
Evaluation -----	35	Programming language -----	11,20,21,38
<b>F</b>		Prototype -----	12
Fact gathering techniques -----	3	Prototyping Model -----	12
Functional requirements -----	iii,v,8,9,20,37	<b>Q</b>	
		Query -----	22,43
		Questionnaire -----	35

## **R**

RAD -----	vi,12
Realism check -----	10,37
Regression testing -----	30,31,38
Requirement validation-----	10 ,37
RUP -----	14,37,41,87

## **S**

Scenarios -----	8
Script file -----	43
Sequence diagram -----	15,53,54
Software engineering -----	11,15
Software requirements -----	20
Software system -----	2,4,8,38
Spiral model -----	12,41
Stakeholder -----	87
Stress testing -----	30,38
Sub menu items -----	56
System design -----	11
System quality -----	29,39
System testing -----	29,30,31,38

## **T**

Test cases -----	31,32,33,34
Test results -----	74,75,76,78,79

## **U**

UML diagrams -----	4,37
Unit testing -----	29,31
Use case diagram -----	3,17
User interfaces -----	4,6,9,18,19,37

## **W**

Waterfall model -----	11,12,40
White box testing -----	29,87