



Stock Management and Demand Forecasting System for Karunaratna Shopping Centre

**A dissertation submitted for the Degree of Master of
Information Technology**

**R.R.Sandaruwani
University of Colombo School of Computing
2017**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: R.R.Sandaruwani

Registration Number: 2015/MIT/047

Index Number:15550472

Signature:

Date:

This is to certify that this thesis is based on the work of Ms. **Ranasinghe Ranawalage Sandaruwani** under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr K. L.Jayaratne

Signature:

Date:

ABSTRACT

Karunarathna Shopping Centre is a modern retail business which located in Thalduwa, Avissawella town. Currently they are selling large varieties of items including textiles, stationaries, house hold items, foods etc... They are planning to expand their business by modifying as super market within next two years. Predicting of items which they required to improve has become a challenge for them. Also currently they maintain their inventory using a computerize system with very primary features. Predicting demand of items and inventory positioning may definitely become a tough task to them, particularly when they enlarge business. Because of the poorly maintained inventory, currently they faced more difficulties in continuous supply of goods to customers and failed to actively monitor the movement of items.

This system intends to improve their stock management process and plan their item ordering process systematically by analysing and forecasting existing selling items. For analyse forecasting next two years sales use as measurement. System also upgrade with fully computerize stock management features.

Considering their present situation a computerize system to measure demand forecasting for stock item with inventory handling functions is a great opportunity. Business administration can manage stocks efficiently and enable profit maximization. Also they can manage a more organized warehouse and improve accuracy of inventory orders. Furthermore they can plan and anticipate for future demand, in advance they can react immediately.

The system has adapted MVC (Model View Controller) architecture and object oriented techniques. Unified Modelling Language (UML) was used for analysis and design. C# language with .NET framework 4.5 and visual studio 2017 was used to deploy the system and Crystal Reports XIII was used for report purposes, meanwhile SQL Server 2016 R2 was used to handle databases. System support in any web browsers including chrome, Internet Explorer, Firefox. Since this is a web based system, it could be operated either in Windows or Linux environment.

System was effectively developed and successfully implemented. According to user evaluation feedbacks, developed stock management and forecast system fulfills and satisfies requirements of Karunarathne Shopping Centre.

ACKNOWLEDGEMENT

This would not have been successful without the support of many people. This is a good opportunity to give sincere thanks of the people who gave their maximum support to make this effort in success.

First I would like to express my sincere gratitude to University of Colombo School of Computing for offering this master's degree program to individuals like myself whom seeking for knowledge in the IT industry, and also to all the staff members who guide students from the beginning of the program.

Secondly I would like to express my gratitude to Dr. L. Jayarathne, my supervisor for the invaluable guidance, encouragement and precious support given throughout the journey. More importantly the valuable advice and criticisms drive project more successful.

I also take this opportunity to express a deep sense of gratitude to my client Mr.Karunaratne for the guidance, assistance given during the all-time period starting from requirement gathering process. Also another word of thanks goes to the Karunaratne shopping centre staff who support by providing valuable data to make success the analysing.

Last, but not least, I express my heartfelt gratitude to my parents, sisters and my office colleagues and friend who were with me during the project period by providing advice, guidance and valuable support to make this effort success.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF ACRONYMS	x
1. INTRODUCTION	1
1.1 Problem Statement.....	1
1.2 Motivation	1
1.3 Aim and objectives	1
1.4 Methodology.....	2
1.5 Scope of Project.....	2
1.6 Chapter Breakdown	3
2. BACKGROUND	5
2.1 Background Introduction.....	5
2.2 Business Background	5
2.3 Overview of current process and Drawbacks	5
2.4 Review of similar Systems	6
2.4.1 OpenBoxes :	6
2.4.2 Loked:.....	7
2.5 Implementation Tools and Technologies	8
3. ANALYSIS AND DESIGN	10
3.1 Introduction	10
3.2 Analysing the current system	10
3.3 Requirements gathering.....	11
3.4 Problem of existing process and propose solutions.....	12
3.5 Requirements of the propose system.....	13

3.5.1 Functional requirements	13
3.5.2 Non-functional requirements	14
3.6 Proposed architecture for the system.....	14
3.7 Design techniques.....	15
3.8 Propose System Design	16
3.9 Database design	18
3.10 Class diagram of the system	19
3.11 Sequence Diagram.....	21
3.12 User interface design	21
4. IMPLEMENTATION	22
4.1 Introduction	22
4.2 Implementation environment.....	22
4.2.1 Server environment.....	22
4.2.2 Client environment	22
4.3 Development tools and techniques	23
4.3.1 Tools	23
4.3.2 Technologies.....	23
4.4 Reusable components	24
4.5 Network architecture	24
4.6 Application Development Architecture.....	25
4.7 Main User Interfaces	26
5. USER EVALUATION AND TESTING.....	29
5.1 Introduction	29
5.2 Testing Procedure	29
5.3 Test Plan and Test Cases	29
5.3.1 Master Data modules	30
5.3.2 Report Module.....	31
5.3.3 Sales Module	32

5.3.4 Login Module	33
5.4 Acceptance Testing.....	33
5.4 Summary.....	33
6. CONCLUSION.....	34
6.1 Introduction.....	34
6.2 Problems Encountered.....	34
6.3 Lessons Learnt.....	34
6.4 Future enhancements	35
REFERENCES	36
APPENDIX A - SYSTEM DOCUMENTATION	37
APPENDIX B – DESIGN DOCUMENTATION.....	39
APPENDIX C - USER DOCUMENTATION	46
APPENDIX D – REPORTS	53
APPENDIX E - TEST RESULTS.....	58
APPENDIX F – CODE LISTINGS.....	63
GLOSSARY	66
INDEX.....	67

LIST OF FIGURES

Figure 2. 1: Main view of openbox software.....	7
Figure 2. 2: View of sales forecasting in Loked software	8
Figure 3. 1: The high level use case diagram for existing manual system	11
Figure 3. 2: Interaction followed in the MVC framework	15
Figure 3. 3: Overall high level Use-Case of the proposed system	17
Figure 3. 4: Structure of the relational database configuration of the system.....	19
Figure 3. 5: Overall classes system of the system	20
Figure B. 1: Sales module use case	39
Figure C. 1: Login page.....	46
Figure C. 2: Home page.....	46
Figure C. 3: Display login user privileges.....	47
Figure C. 4: Navigation bar	47
Figure C. 5: Common button set	47
Figure C. 6: Id generate for all category.....	47
Figure C. 7: Date picker using system.....	48
Figure C. 8: Common grid view used in modules.....	48
Figure C. 9: Display error messages.....	48
Figure C. 10: Save message display	49
Figure C. 11: User Information module	49
Figure C. 12: E-mail validation	50
Figure C. 13: User details interface	50
Figure C. 14: Purchase order generation module	51
Figure C. 15: Sales Module	51
Figure C. 16: Item interface.....	52

Figure C. 17: Item interface.....	52
Figure C. 18: Report module	52
Figure D. 1: Inventory reports	53
Figure D. 2: Purchase Order	54
Figure D. 3: Stock valuation.....	54
Figure E. 1: Overall feedback.....	62

LIST OF TABLES

Table 3. 1: Problems encountered and solutions	12
Table 4. 2: Hardware and software requirements of client environment	23
Table 5. 1: Test case for master data module	31
Table 5. 2: test case for report module	32
Table 5. 3: Test case for sales Module	32
Table 5. 4: Test case for login module	33
Table A. 1: Hardware	37
Table A. 2: Software.....	37
Table B. 1: Use-Case Description for Sales module	39
Table B. 2: Use-Case Description for purchase module	40
Table B. 3: Use-Case Description for good received module	40
Table B. 4: Use-Case Description for Master page Maintain module.....	41
Table B. 5: Use-Case Description for report generate.....	41
Table E. 1: Common Functions	58
Table E. 2: Item Creation	59
Table E. 3: Purchase order.....	60
Table E. 4: Acceptance test results	62

LIST OF ACRONYMS

AJAX - Asynchronous JavaScript and XML

CD ROM - Compact Disc Read Only Memory

CSS - Cascading Style Sheet

DBMS - Database Management Systems

ER - Entity Relationship

GUI - Graphical User Interface

HTML - Hypertext Mark-up Language

MVC - Model-View-Controller

OO - Object Orient

OOA - Object Oriented Analysis

OOD - Object Oriented Design

PDF - Portable Document Format

RAM - Random Access Memory

RUP - Rational Unified Process

SQL - Structured Query Language

UML - Unified Modeling Language

URL - Uniform Resource Locator

1. INTRODUCTION

1.1 Problem Statement

Today we are living in a competitive world. Therefore, businesses need to adapt for changing circumstances in future. All requirements of the business sector need to be technically accurate and practical. Demand forecasting is the best solution for any business to access the probable demand for its items and to manage purchasing and ordering items accordingly [1]. In common practice, perfect demand forecasts lead to efficient and satisfactory customer service.

Karunaratna Shopping Centre is a two story building, selling large varieties of items which have more than 50 years in retail business experience. They are going to expand their business within next two years. Predicting of items which they required to increase has become a challenge for them. Currently they are maintaining inventory handling system with very primary features. This system intend to plan their item ordering process systematically by analysing and forecasting existing selling items. System also upgrade with fully computerize stock management features.

1.2 Motivation

Demand forecasting is very effective concept in Inventory Management Process [2]. Most of supermarkets in Sri Lanka do not follow proper forecasting methodology for their item handling process.

At the moment Karunaratna Shopping Centre selling more than thousands variety of items which purchasing from large number of suppliers. They are going to enhance their business within next two years by increasing varieties of items. Currently they have inventory management system with very basic functions and most of operations are done manually. But with the competitiveness, they are looking for new stock handling system with new features.

Considering their present situation a computerize system to measure demand forecast of stock item with inventory handling functions is a great opportunity to them. They can plan and anticipate for future demand, in advance they can react immediately for any situation in stock handling.

1.3 Aim and objectives

The main aim of the system is to forecast item demand, control stock management and support to expand the business efficiently and effectively. Following present the objectives of the project.

1. Investigate the existing inventory controls and identify potential capability to expand the business
2. Planning inventory levels. (ex. Increase order quantity, new items, promotions, Seasonal variations etc.)
3. Developing a mechanism to reduce item obsolescence cost and item receiving delays.
4. Make important business decisions to expand the business by identifying expanding areas, risking areas and areas required to develop.
5. Develop a system to identify item demand forecast and manage stocks which supports to expand the business

1.4 Methodology

System methodology will be fundamentally based on quantitative analysis. Karunarathna Shopping Centre staff going to use this system within next two years for gain knowledge about item demand. Demand forecast will be predicted by investigating and analyzing historical stock data. System will be developed as Auto detect probabilistic forecast of each and every item in stocks Sales compare by two items, item sales compare with customer and item sales compare with supplier.. Also system will be facilitated the statistical correlation of items with analyzing entire probability distribution.

1.5 Scope of Project

Forecasting is a good measure to identify the life cycle of a business. Therefore, demand forecast is very essential requirement for the survival of business. Scope of the project is listed below.

1. Generate reports for daily, monthly, yearly and selected date range to measure forecast of each and every item.
2. Facility of daily update system history.
3. Customer registration with main customer details to measure forecast such as age, status etc.
4. Identify most customer favourable items in each hours of the day and manage showroom items.
5. Master screens to maintain stock details such as item details, usage of items, ordering process.

6. Manage inventory control levels such as reorder level, order quantity maintenance, return items etc.
7. Maintain supplier details to send item order requests via emails.
8. Analyse every item individually to measure its demand forecasting using detail tables and chart views.
9. Report generation
 - Forecasting related reports for hourly, daily basic, monthly, quarterly, yearly.
 - Customer base, Supplier base and item comparison reports for measure demand
 - Inventory control reports

1.6 Chapter Breakdown

Dissertation is the document which contains overall information of the project in chapter wise. This dissertation contains six main chapters followed by reference and appendix.

Chapter 01: Introduction

This chapter provides an overview of the content to motivate reader. Introduction to project, scope and objectives included here.

Chapter 02: Background

This chapter is written to provide an idea about background of implementing the project. Other than that, chapter provides a review of similar systems, tools and technologies which used in the system implementation.

Chapter 03: Analysis and Design

All fact gathering techniques, functional and non-functional requirements which need to specify software requirement specification and analysis will be included in this chapter. Design chapter includes methodologies, tools and techniques used to design the system. Design selection is based on sound justifications and user interface, database and modules of system will be designed.

Chapter 04: Implementation

This chapter describes how specifications were gathered (in analysis and design phases) are converted into executable program. Furthermore chapter incorporates with Implementing technologies and concepts.

Chapter 05: User Evolution and Testing

This chapter describes how the system was tested by using various testing methods. It also reported errors and how to overcome those errors and how to modify the system.

Chapter 06: Conclusion

This chapter include the all summarize details of the projects and also include findings and lessons learned during the project with further improvements of the project.

Reference

All the referred books, URL references and other materials will be included in this section.

Appendix

This include further details and supplementary parts relating to chapters.

2. BACKGROUND

2.1 Background Introduction

Demand forecast play a critical role in supermarket inventory management process. Providing progressive forecasts in a supermarket allows all representatives of the particular chain to enhance their inventory management decisions [3].

There are many forecasting techniques and that can be divided into four main groups such as follows [4].

- Qualitative methods: Depend on human opinion to decide a forecast
- Time series methods: Use historical figures to make a forecast
- Causal methods: Involve assuming that the demand forecast is highly connected with environment factors
- Simulation methods: Imitate the consumer choices that give increase to demand to arrive at a forecast

All of this comprises with several advantages and disadvantage. Therefore best way is develop a system combing above mentioned technologies to predict demand forecast.

2.2 Business Background

Karunarathna Shopping Centre is established in 1941 in centre of Avissawella, Thalduwa town. Supermarket heads to main road, therefore huge number of transactions are occurred for a day. This shop offers a broad range of products purchased from a large number of distributors and manufacturers. Now they are planning to expand their business by adding more space to business area. So they plan to purchase more items. In order to effectively provide such a variety of items to their customers at competitive prices, the supermarket has to manage their stock successfully. Therefore, this system will develop to determine the reliable prediction of future demand and to maintain their stocks.

2.3 Overview of current process and Drawbacks

Currently Karunarathna Shopping centre handling inventory management system with very basic features. Current system provide few operations such as item creation, item usage, purchasing items and stock adjustment. However, they manually send their purchase orders to supplier, manage good received notes and invoices handling and insert stock in quantities to the systems. They forecast demand of items by looking the usage quantities and measuring usage

percentages. In common practice, perfect demand forecasts lead to efficient and satisfactory customer service. There are some drawbacks of existing system are listed below,

- System was not updated within past few years, therefore some processes are run with errors.
- System user cannot predict future demand and future stability of items.
- Many functions which can be operated in systematic way, but currently operate manually with lots of paper works.
- System generates few reports. But they also have mistakes such as unbalancing stock balances, errors in stock limits and re order levels.

2.4 Review of similar Systems

2.4.1 OpenBoxes :

URL: <https://openboxes.com/>

Open Box is open source and free supply chain managements system basically design to support health care facilities. System maintains inventory functions and track stock movements. Not only inventory movements, system provides all historical information of transactions. This may facilitate to item forecast demand using historical data for future planning periods. Figure 2.1 illustrates the main view of ‘openbox’ software.

This software also provide Inventory visibility and tracking with max min and reorder points which allow s the users to quick access if supplies are over stocks or out of stocks. System support windows, Ubuntu, Mac platforms with basic system requirements such as Chrome/Firefox, Java 7, Tomcat 7, MySQL 5.5, SMTP server [optional] and written primarily in Groovy. System is fully computer base. So business owners have to hire computer literate employees to handle this software. [13]

Showing 1 - 10 of 465 results

Product code	Name	Manufacturer	Brand name	Manufacturer code	Qty +	Qty -	Qty
Drugs and pharmaceutical products (10)							
EM30	Abacavir sulfate, 20mg/mL, Oral solution, 240mL bottle	Aurobindo			0	0	0
EB12	Abacavir sulfate, 300mg, tablet	Aurobindo			0	0	180
XJ13	Acetazolamide, 250 mg, tablet	Micro Labs		10031	0	0	1000
HD92	Acetylcysteine, 200 mg/mL (20 %), 10 mL ampoule	Aurum		10069	0	0	450
BG90	Acetylsalicylic acid, 100 mg, tablet	IDA		024500	0	0	5970
CH56	Acetylsalicylic acid, 100 mg, tablet	Nanjing Beijingyu Pharmaceuticals		10892	0	0	11550
EU48	Aciclovir, 200 mg, capsule	Mylan Pharmaceuticals, Inc.			0	0	15390
DC96	Aciclovir, 200 mg, tablet	IDA			0	0	0
PJ94	Aciclovir, 200 mg, tablet	Medopharm		34772	0	0	74700
ZU93	Aciclovir, 3% eye ointment, 4.5 gram tube	Galentia		10096	0	0	115

1 2 3 4 5 6 7 8 9 10 ... 47 Next

Results per page: 10 25 50 100 All

Figure 2. 1: Main view of openbox software

2.4.2 Loked:

URL: <http://www.lokad.com/>

Lokad is a quantitative supply chain management system. Software helps to improve purchasing and make better replenishment decisions. Maintaining stock and decide how much to buy, when, where to store and when to recount and etc. Instead of predicting on value per product, Lokad predicts the entire probability distribution. Figure 2.2 illustrates the view of sales forecasting in Loked software.

System support any platform and compatible with any browser. The main advantages as said by users this software perform high reactivity, adaptability and especially trustable inventory optimization. But main problem is computer literate employees or trained staff must essential in business to manage stocks using this software. [4]

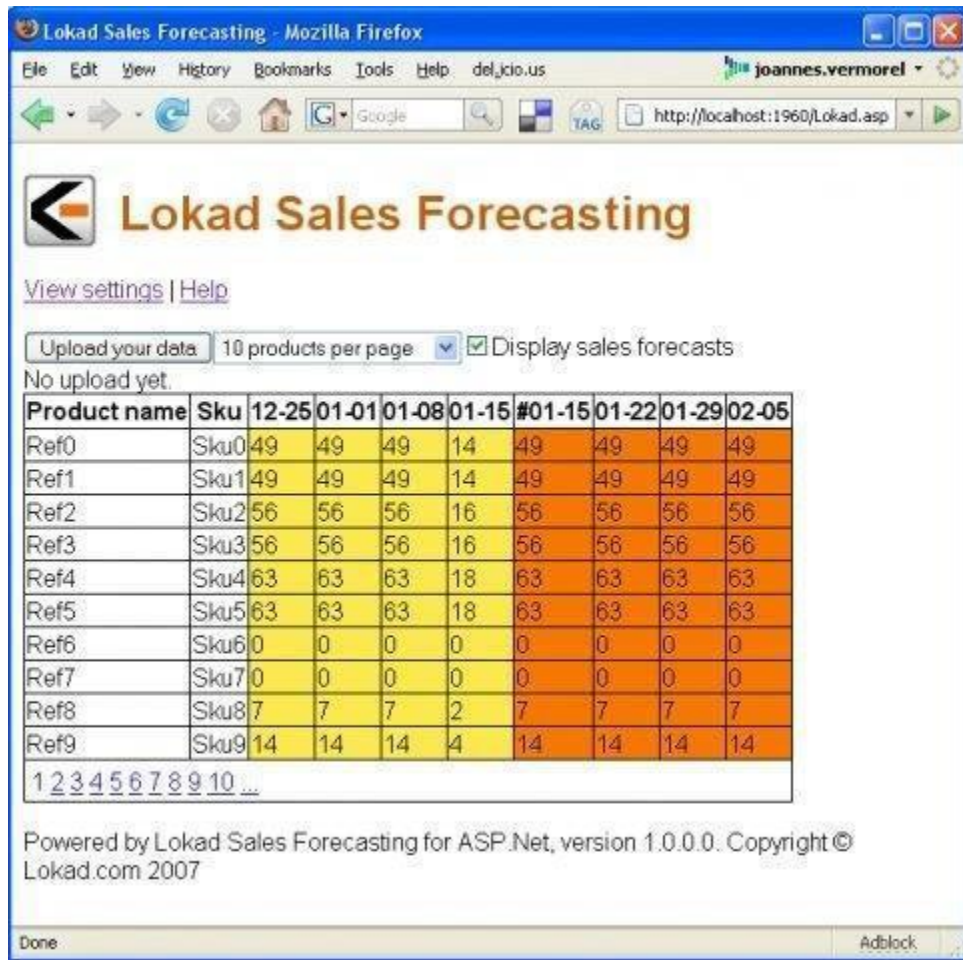


Figure 2. 2: View of sales forecasting in Loked software

2.5 Implementation Tools and Technologies

Proposed system will be used by business users who are not familiar with programming. Therefore system should be understandable and user friendly. Object oriented working environment used for developments. Latest technologies and tools will be used for development. Development framework is ASP.NET 4.5, MS SQL as database, Crystal reports XII for report generation. Other than system will use bootstrap framework 3.3, CSS for styles and java scripts with JSON format for validations. MVC 5 used to develop architectural pattern with entity frameworks.

.NET will add many object oriented rich feature to system development. Platform like .NET framework supports many languages, such as C#, VB.NET, Visual C++, JavaScript applications. MS SQL is Microsoft authenticated and secured RDMS which can integrate with other common databases easily [5].

2.6 Summery

Over the last few decades, many different forecasting techniques have been developed in number of different application areas, including engineering and economics [6]. Proposed Demand forecasting and stock management system is developed for a supermarket which handling huge number of transactions per a day. This project follows quantitative analysis method which considered two years of historical stock data distribution with stock management functions. End user understandable report also generated through the system. System use latest technologies and tools for developments.

3. ANALYSIS AND DESIGN

3.1 Introduction

This chapter give an overview of system analysis and system design. Requirements gathering and analysis is a significant practice for a successful project. The main processes of this phase include domain understanding, requirements collection, classification, structuring, prioritization and validation. Appropriate methods and processes were engaged to carry out the analysis phase in an effective way.

Software design phase is an iterative process in which requirements gathered in analysis are translated into a “blueprint” in building the system. Afterwards this can be elaborated into detailed functional and behavioural requirements [7]. The output of the design phase is the system specification.

3.2 Analysing the current system

To recognize main system functionalities, it is vital to analyse the current system methodologies. Domain analysis was carried out through certain fact gathering techniques such as interviews, observations and consulting domain experts in the domain's own terminology. Current stock handling system is using with very few computer base functions and lot of paper works are there. Also they haven't proper method to measure item demand or forecasting, stock reordering and moving patterns.

According to analysis following system functionalities were identified.

- Issue stock
- Stock verify and Dispose
- Reorder Level & Purchase
- Query and Respond stock
- Stock receive and Return
- Monitor and reports

Figure 3.1 represents the high level use case diagram for existing manual system

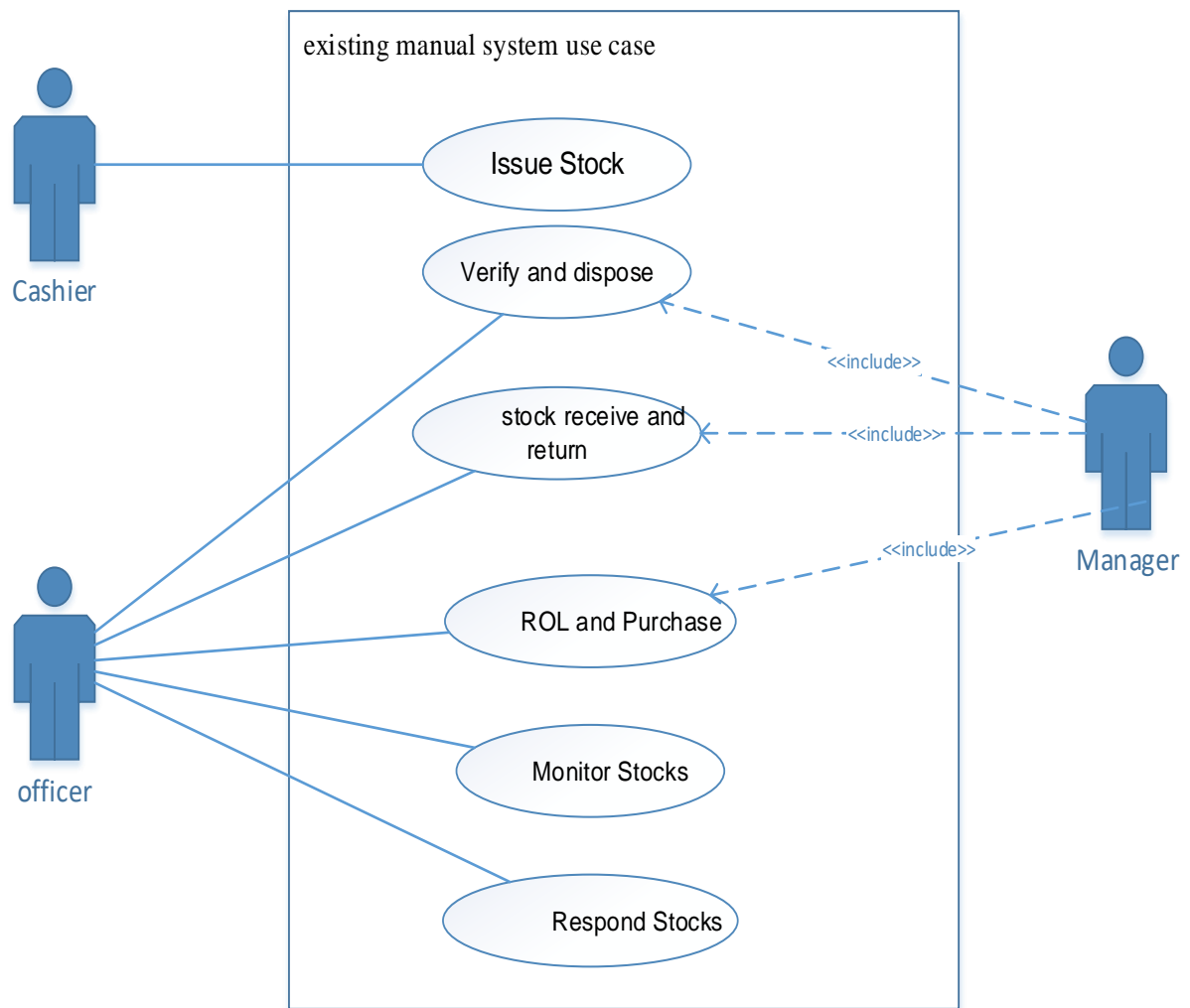


Figure 3. 1: The high level use case diagram for existing manual system

3.3 Requirements gathering

Several kinds of techniques were engaged here to capture requirements from different stakeholder perspectives. The main approaches used for requirement gathering process were; interview, observation, document review and scenarios.

Both open ended and close ended interviews were conducted for prompt existing and also to identify new problems. Relevant documents and current system were analysed and studied through a sampling observation methodology. Captured facts were represented using scenarios. Use case diagram of each module was designed to get clear picture of the gathered requirements.

3.4 Problem of existing process and propose solutions

The requirements gathering phase identify the problems in current process and following discuss the propose solutions for relevant problems. Table 3.1 display Problems encountered and solutions

Problem	Propose Solution
At the end of a day of sales, the sales bills were collected from cashier and enter records and reduced the item quantities sold in system.	System improves to reduce stocks concurrently with sales.
Customer returns and returns of damaged items to the supplier are handling manually when stock balancing.	System should support to reduces such quantities immediately
Sales person have to check stock levels every day and balance it manually for every single individual item to keep the stock levels up-to date	System should automatically update on sales real time and should provide vital and accurate stock levels.
Current system updates are not happen in proper flow and very difficult to instantly track records.	System should provide better interconnection with all modules and should allowed easy filtering.
Failing to monitor fast moving items and slow moving items and their ordering arrangements.	System should provide reporting facility to capture re-order level and identify the items which needs to order.
Current system hasn't present demand of item or forecasting of the items.	System should measure demand of each and every item with their forecasting patterns.
Past stock manage patterns are not available in current system.	System should allow to analyse historical data patterns with report generation facility.

Table 3. 1: Problems encountered and solutions

3.5 Requirements of the propose system

3.5.1 Functional requirements

Functional requirements express the required behaviour of the system to be built or what the system supposed to do. Below listed are the main functional requirements of the propose system.

- **Stock management and maintenance**

User should be able to add, modify information, view, search and delete an item from the stock index. User should be able to instantly view up-to date, inventory status, cost of the inventory, and retail value of the inventory, number of active items and number of items reached low stock level, fast moving items and slow moving items.

- **Item issue and return process**

Record and calculate daily sales and daily expenditure. Record returned items from customers with reasons to expand sales quality.

- **Item ordering process**

System support users to check item moving patterns and manage ordering according to such patterns. User should be able to view purchase orders status instantly when previous processed purchase orders and up-to date number of items reached to low stock level.

- **Stock analysing and demand forecasting**

The system should allow the relevant users to generate and view the timely reports related to sales, purchase, current stocks and measure demand of each and every item. The system should allow the users to print the reports and documents.

- **Customer management**

User should be able to view customer information, add new customer, search a customer, modify and measure item demand according to customer type.

- **System administration**

The system should allow the administrator to create new users accounts with differing privileges for the employees who needs to access to the system. The system should allow the administrator to take backups and restore files when needed.

3.5.2 Non-functional requirements

Non-functional requirements are the requirements which are not directly affected with the exact basic functions delivered by the system. But, failing to meet non-functional system requirements may make the whole system unusable. Non-functional requirements are relevant with quality attributes, quality of service requirements and non-behavioural requirements. The non-functional requirements of the system are listed below.

- The system should provide a user friendly environment including flexible interfaces,
- Person who has average computer skills can work with the system with a short period of training.
- The system should be accurate and consistent, when manipulating the fed data in proper way and displaying correct information,
- The system should keep up security and reliability, because the system handles important data related to business processes of the company.
- Occasionally backups should be taken to maintain reliability and necessary security measures.
- The system should be reusable and maintainable.
- It should be able to adopt future implementations to the system.

3.6 Proposed architecture for the system

System was used MVC architecture for developments. MVC which represent **Model View Controller** is a software design pattern for developing web applications. It divides given software application into three interconnected portions, which has a full skills to support rapid web application development and dynamic interactivity with the database [8]. Figure 3.2 shows the interaction followed in the MVC framework. [8]

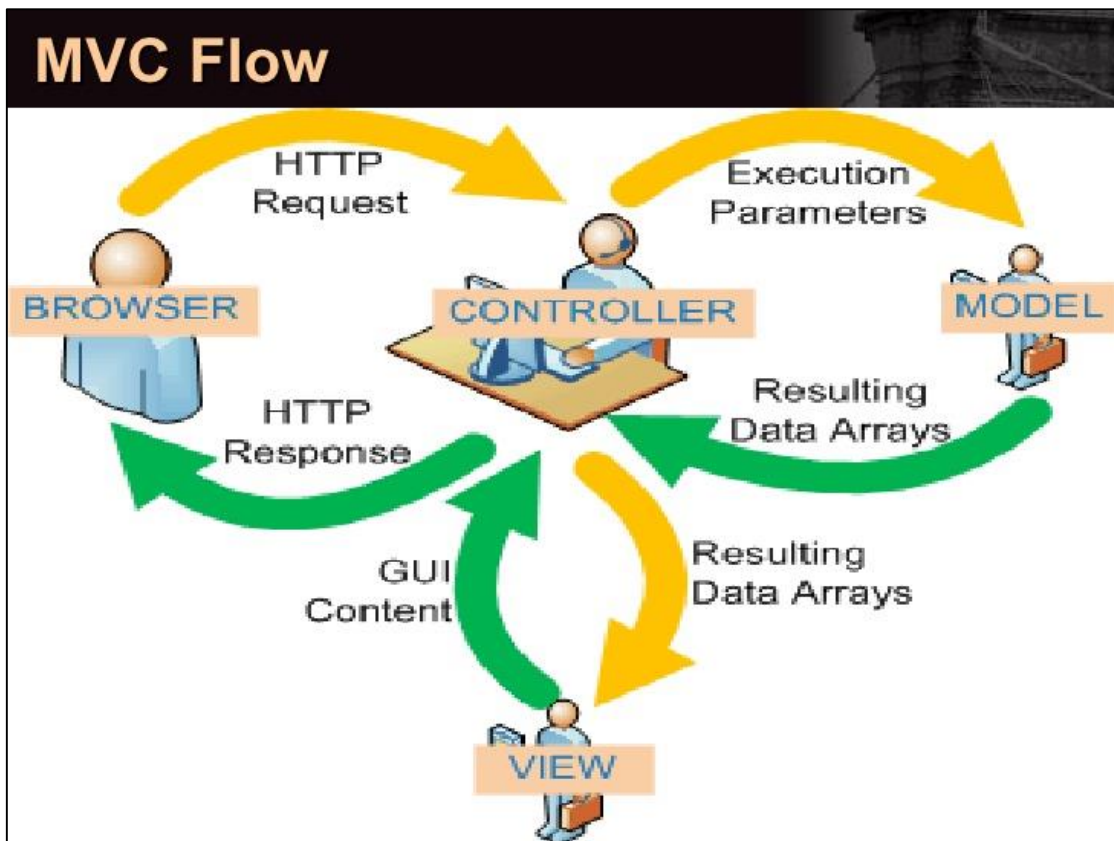


Figure 3. 2: Interaction followed in the MVC framework

Controller place at the heart of the architecture and it interact with the user through the web browser. Behind the scene, it links with the Model and the View components of the architecture.

The **Model** in point of fact encloses the database. Therefore the Controller can express the database according to the user actions on the browser. It meant that the database is modified in accordance with the requests given by the user. This develops user interactivity and therefore the overall experience.

The controller switches the Model according to the query request given by the user; and in return Model informs **View** to update the user interface (UI) on the screen in accordance to the database change. The cycle restart when the user enters another request as query [9].

3.7 Design techniques

Design techniques are the techniques use to model the system. There are different design techniques such as Modern Structured design, Rapid application development, and prototyping. Object oriented design technique was chosen out of them because of code reusable and recycling facility, design benefits and maintainable facilities with Objects and classes.

Unified Modelling Language (UML) plays a significant role in Object Orient designing. UML allows programmers easy understand models of objects so that programmers can easily write software. Some of structural and behavioural UML diagrams use for design proposes systems are mentioned below [10].

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

.NET C# is the language which was used to develop the system, it does have wide-range of object-oriented features. To make the development context simpler and strong, the system has been built on top of Model View Controller framework. Using MVC architecture, the system can be structured in order to support reuse of program code, which is an important feature of Object orient development [9].

3.8 Propose System Design

Diagram in Figure 3.3 depicts the overall high level Use-Case of the proposed system.

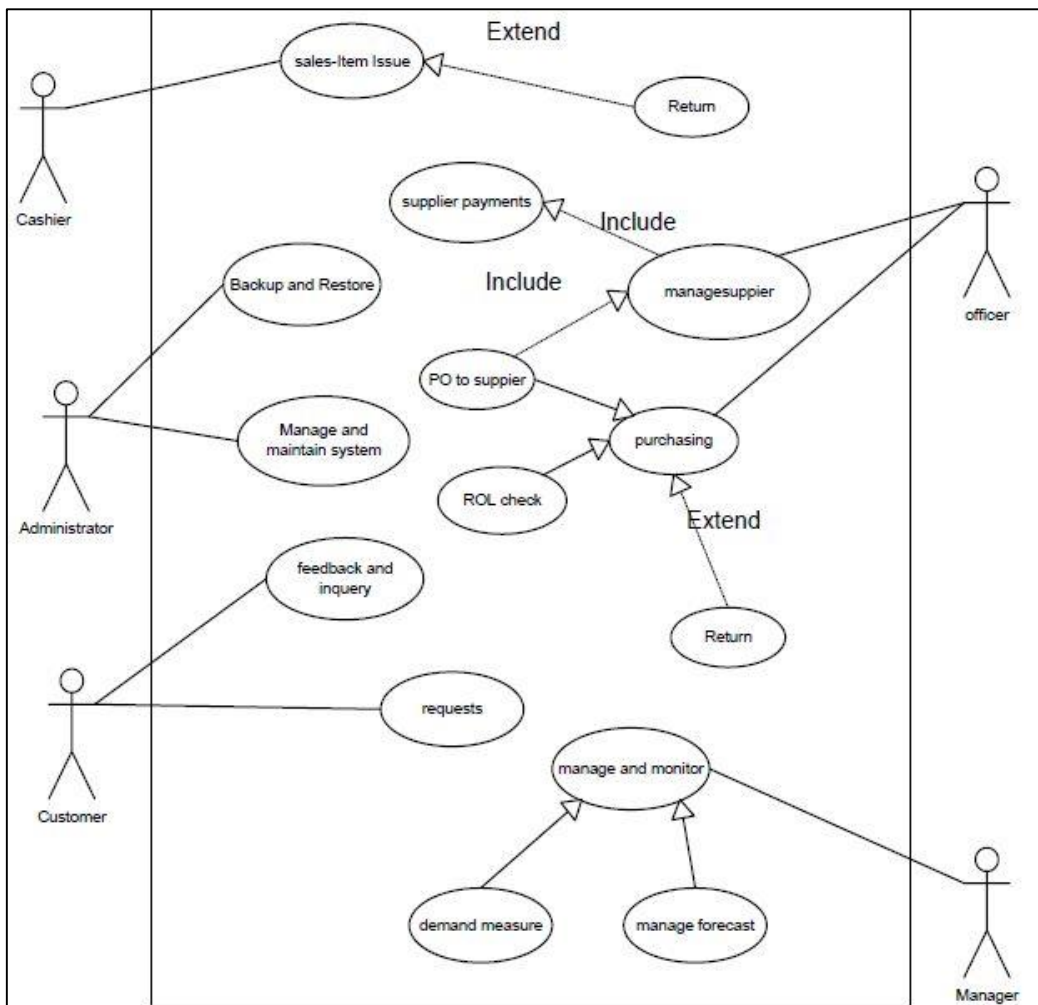


Figure 3. 3: Overall high level Use-Case of the proposed system

System will promote the users to bring out their responsibilities effectively and efficiently by reducing the manual human effort and by generating reports to take decisions on time. The suggested solution has been separated into few modules in order to make the development and comprehension easier. The modules are as follows,

1. Stock management and maintenance module
2. Item issue and return module
3. Re-order level maintain & Purchase module
4. Stock analysing and demand forecasting module
5. Customer management module
6. System administration module

- Stock management and maintenance module

User should be able to add, modify information, view, search and delete an item from the stock index. User should be able to instantly view up-to date, inventory status, cost of the inventory, retail value of the inventory, number of active items and number of items reached low stock level, fast moving items and slow moving items.

- Item sales and return module

Record and calculate daily cash and credit sales and also calculate daily expenditure. Record returned items from customers with reasons to expand sales quality. In this module hourly, daily and monthly sales was analysed using historical data.

- Re-order level maintain& Purchase module

System support users to check item moving patterns and manage ordering according to such patterns. System catch slow moving and fast moving item patterns with their ordering qualities. User should be able to view instantly purchase orders status, previous processed purchase orders and up-to date number of items reached low stock level.

- Stock analysing and demand forecasting module

The system should allow the relevant users to generate and view the timely reports related to sales, purchase, current stocks and measure demand of each and every item. The system should allow the users to print the reports and documents.

- Customer management module

User should be able to view customer information, add new customer, search a customer, modify and send an email to a relevant customer when necessary. System also provide interface for customers to add requests and feedbacks.

- System administration module

The system should allow the administrator to create new users accounts with differing privileges for the employees who needs to access the system. Allow the administrator to take backups and restore files when needed.

3.9 Database design

Database design is completed through data modelling. Database design is used to specify the structure of business objects used in the client/server system. The database tables were normalized to 3rd normal form to avoid redundancies. The following Entity Relationship diagram

in Figure 3.4 depicts the structure of the relational database configuration of the system. Refer appendix B for detailed description of the database ER diagrams.

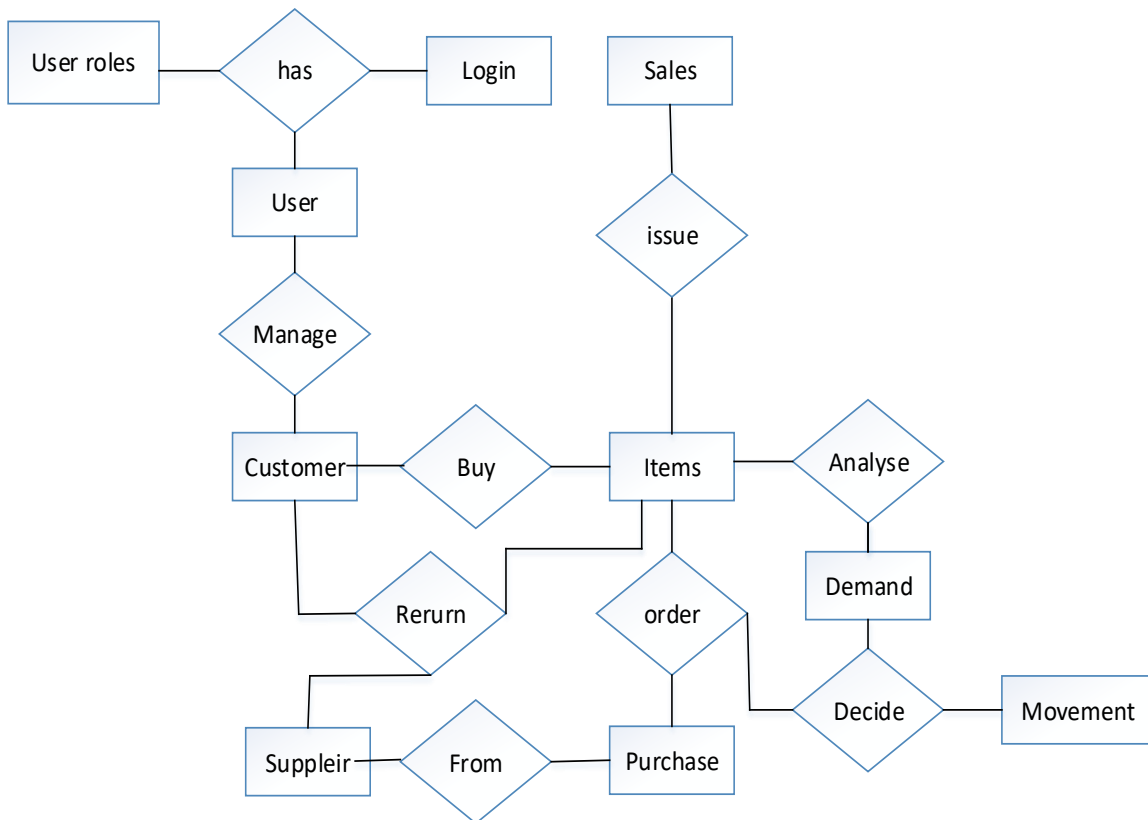


Figure 3. 4: Structure of the relational database configuration of the system

3.10 Class diagram of the system

Class diagrams support to carry out conceptual and domain modelling. A conceptual model represents objects and classes from a particular system user’s perspective. The following class diagram in Figure 3.9 depicts the overall classes system of the system.

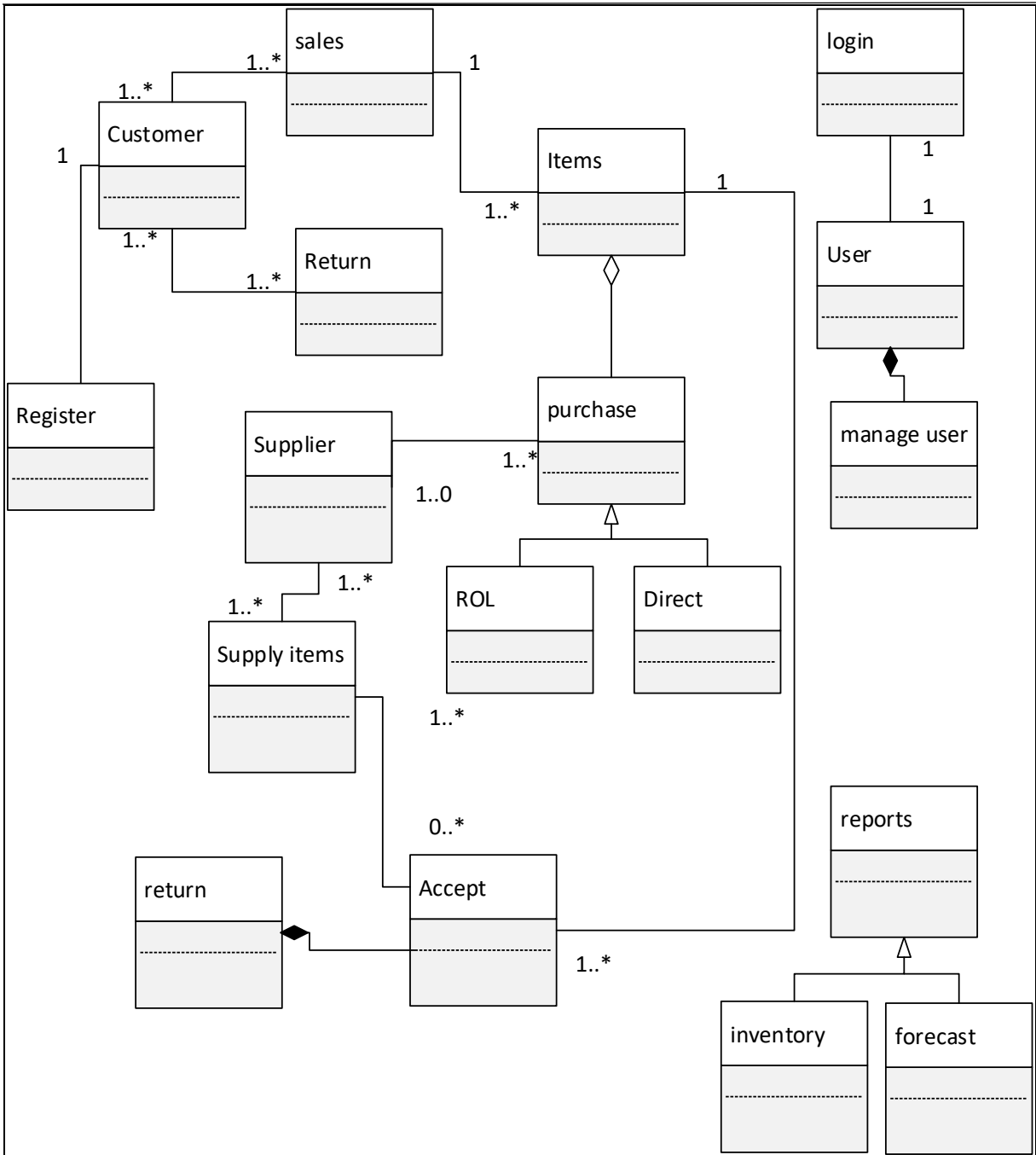


Figure 3. 5: Overall classes system of the system

3.11 Sequence Diagram

Below figure 3.6 display sequence diagram for report module.

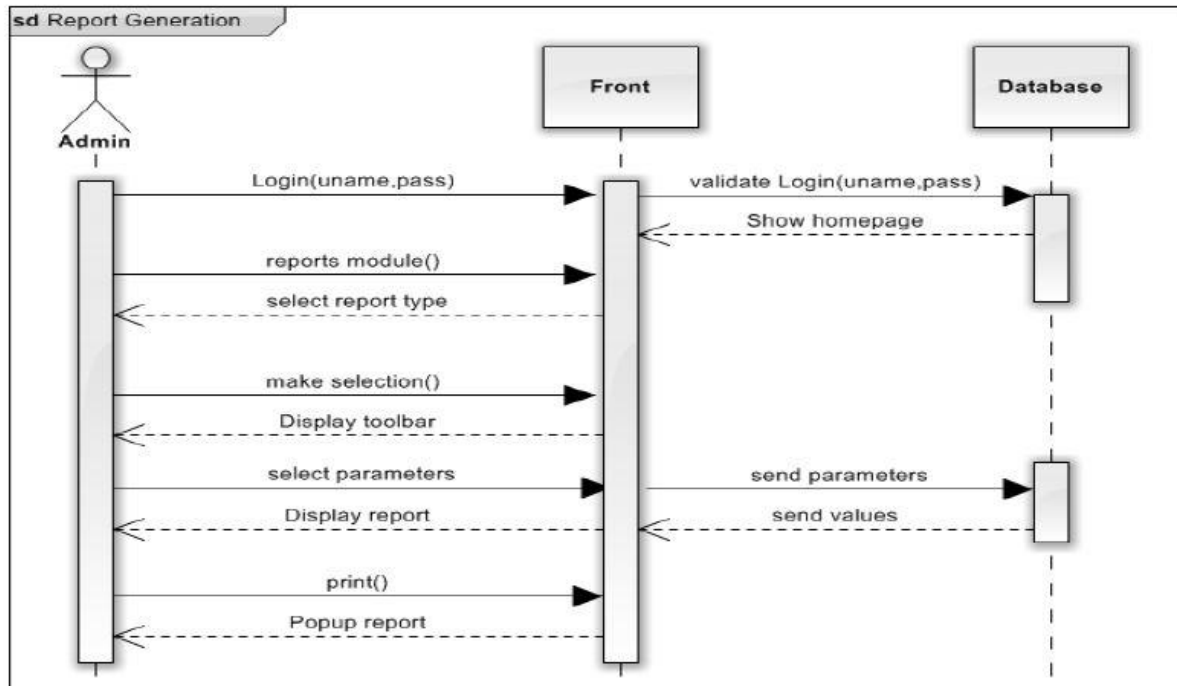


Figure 3.6 Report generation sequence diagram

3.12 User interface design

User interface is the primary and very first interaction with user and system. So it should be interactive to user. User Interface (UI) Design focuses on what users expect through the system and ensuring that the interface has fundamental facilities which are easy to understand, access, and use [11]. There are some properties that a user interface must have,

- Avoid unnecessary elements and keep the interface simple:
- Purposefully use colour and texture: make direct attention toward or redirect attention away from items using colour, texture, contrast.
- Use auto select options as well as let the users select options for better flexibility.
- Use common UI elements and build consistency: By using more comfortable common elements, for users and able to get things done fast.
- Use typography to create hierarchy and clarity: Different sizes, fonts, and arrangement of the text to help increase scalability, legibility and readability.

4. IMPLEMENTATION

4.1 Introduction

Implementation process use to convert system specifications to executable system. In this phase convert design to user friendly views with front end interfaces and back end systematic logics for perform functionalities. A comfortable language and suitable tools were selected in the process of coding and development. The codes were written and arranged in a readable format. Comments added for special logics to help for future implementations.

4.2 Implementation environment

Server side environment and client side environment were consider in implementation process. These environment requirements basically divide in to software requirements and hardware requirements. These requirements are described below.

4.2.1 Server environment

System hosted location and client server architecture operators are considering in this environment. Server environment is used to host developed system, which can handles all the data processing activities. Server should be a Mid-upper range server with 12 GB RAM and 300 GB Storage.

4.2.2 Client environment

Client software run in between the user and server be responsible for a user interface for user to cooperate with the server, for example internet explorer like web browser act as client software. System is rich with latest technologies and tools, so updated web browsers should use to avoid browser compatible issues. System is capable of working with different hardware and software requirements, minimum system requirements are listed below. Table 4.2 identify hardware and software requirements of client environment

Hardware	Software
2.4GHz CPU or upwards	Windows 2008 (64 bit) or higher Operating System
2GB RAM	SQL Server 2016 R2

At least 120GB hard disk	Dot Net framework 4.5 (64 bit) , Microsoft visual studio 2017
Printer for report printing	Crystal Reports XIII
	IIS 7.0 Server or higher

Table 4. 1: Hardware and software requirements of client environment

4.3 Development tools and techniques

Below described the tools and technologies that were used to develop the system.

4.3.1 Tools

- Visual Studio 2017.
Visual Studio 2017 Professional edition was used as the IDE for this project. Visual Studio make development easy and quicker development with features such as IntelliSense which is an automatically complete system that enables to find and use parameters, classes etc... Also helping automatically generate code in the Code Editor.
- SQL Server 2016 R2
SQL Server Management Studio was used to handle all database related developments. This offers a set of tools for development, administration objects.
- Cristal report VS_13_0_20
Cristal report used for design and generate reports graphically. Using this tool table data can link with different formulas and capable to presenting user understandable formats.
- Adobe Photoshop CC
Used for photo editing purposes.

4.3.2 Technologies

- C# as development Language
C# used to build applications that run on the .NET framework and programming language is a fully object oriented language.
- .NET framework
The .NET framework is a software development environment. It permits build and deploy applications quickly. It consists of the common language runtime and .NET class library which has set of reusable types. .
- HTML
HTML was used to build the base Interfaces of the system.
- CSS

CSS Styles was used to make interfaces more user friendly and attractive. Styles basically decided the look of the system.

- JavaScript

JavaScript was used to code all the client-side validation.

- AJAX

Which is based on JavaScript was used to get data from the server without refreshing it repetitively

- JQuery

Which is also based on JavaScript was used to implement some features such as the pre-coded time picker module

4.4 Reusable components

- Date picker – To select a date from the calendar.
- Data table – Table adept of searching and filtering data, pagination of data.
- CSS – Cascading Style Sheet to design an interactive user interface.
- Cristal Report format – Provide report generation facility which used to create PDF documents
- Charts - Use to generate HTML based JavaScript based interactive charts to display information graphically to user.

4.5 Network architecture

The system contains with central web server users connects to the server via internal network. Users interacts the system via web browser, and internal network protected by firewall.

Figure 4.1 portrays sample diagram for system implementation



Figure 4.1: Network Diagram of System

4.6 Application Development Architecture

As discussed in the previous chapters, the system is built on top of MVC. The Model is used for database interaction. The Controller is to send requests and responses back and forth between models, views. Graphical user interfaces represent by Views [9]. Figure 4.2 shows designed architecture for the development.

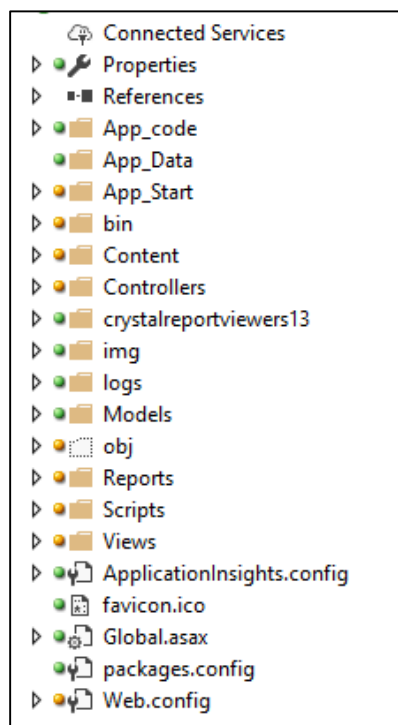


Figure 4. 2: Application Development Architecture

4.7 Main User Interfaces

The below section of the document provides only a few main user interfaces in order to show system interface structure. Please Refer Appendix C - User Documentation for the rest of the interface designs.

Login Page

This interface (Figure 4. 3) allows the user to log in to access the system with correct user name.

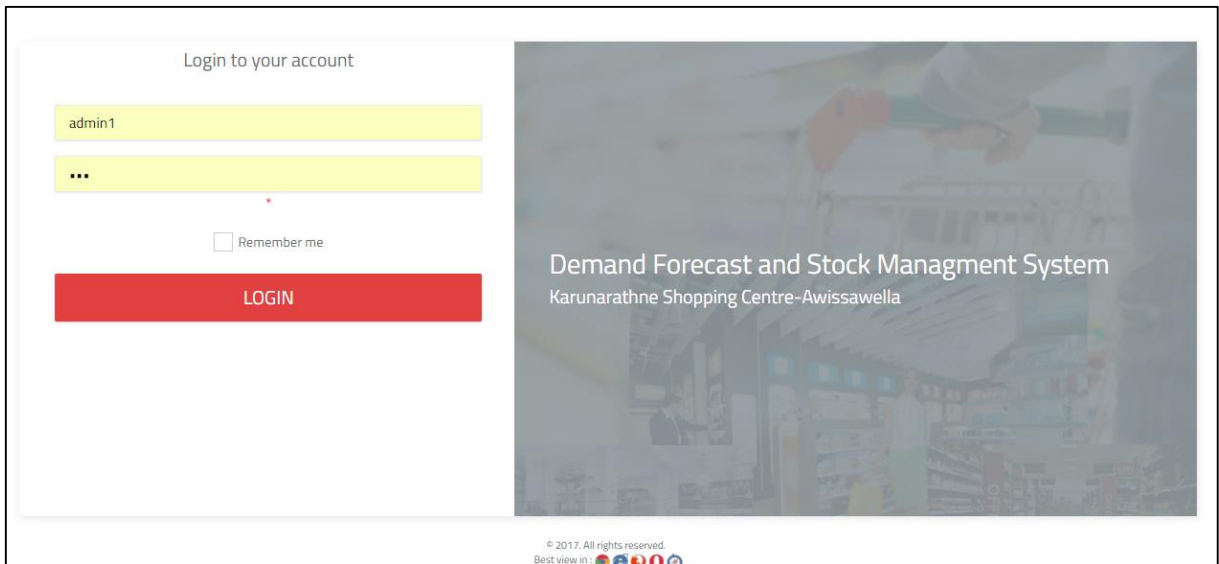


Figure 4. 3: Login Page interface

Home Page

The homepage provides the links and the menu bar to let the user to navigate easily through the system. This pages display according to user privileges. Figure 4.4 shows home page of Management level user.

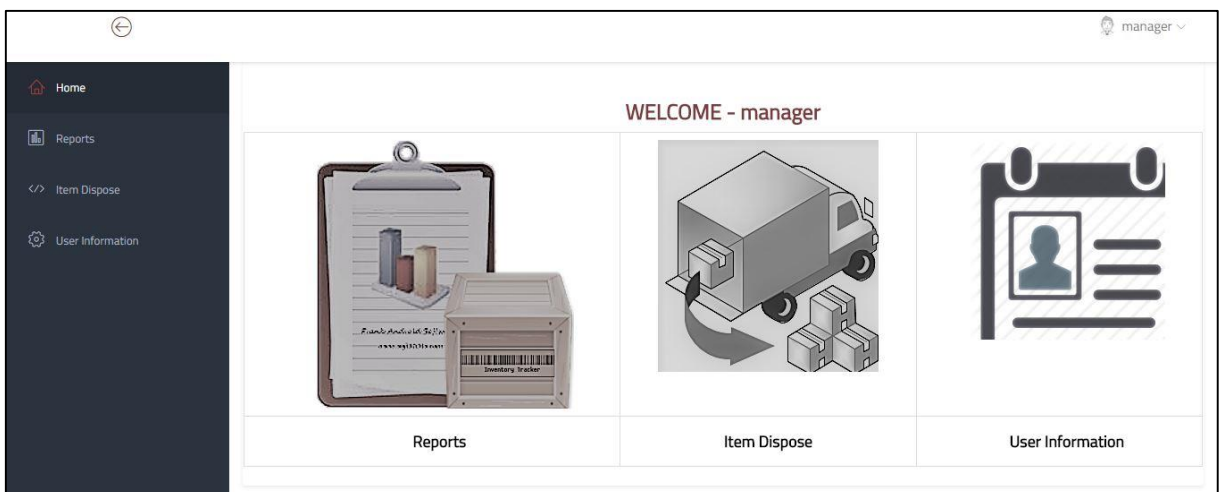


Figure 4. 4: Manager Home page

Report View

Mainly two views use for report generation. Inventory report view and forecast report view. Toolbars are used to generate different reports in reports module. Figure 3.8 show forecasting report view tabular format. In this table user can filter reports according to their needs.

Report No	Report Name	View
01. Item wise Sales Demand		
1.1	Daily	Select
1.2	Date Range	Select
1.3	Monthly	Select
1.4	Yearly	Select
1.5	Date Range - Hourly	Select
02. All Sales Details		
03. Top Sales Item Details		
04. Low Ten Sales Item Details		
05. Item Comparison with Item		
06. Sales Comparison with Customer		
07. Sales Comparison with Supplier		

Figure 4. 5: Forecast Report Selection view

Data entry form

The system have forms in each module to input/update data. Figure 4.5 shows sample form module. System developed with both client-side JavaScript validation and server-side validation to confirm the right data is fed into the database

Item Creation			
Item Code	ITM00009		
Item Category	--Select--	Item Name	<input type="text"/>
Item Quantity	0	Re-Order Level	0
Active	<input checked="" type="checkbox"/>	Unit Price	0.00
Discription	<input type="text"/>		
<input type="button" value="Save"/> <input type="button" value="Search"/> <input type="button" value="Cancel"/>			

Figure 4. 6: Data entry form

Report and Charts

Report generation and graphical chart views display according to requirements. This reports display according to given selected parameters such as date range, year, month and item. Figure 4.5 display report generate for selected item by selected date.

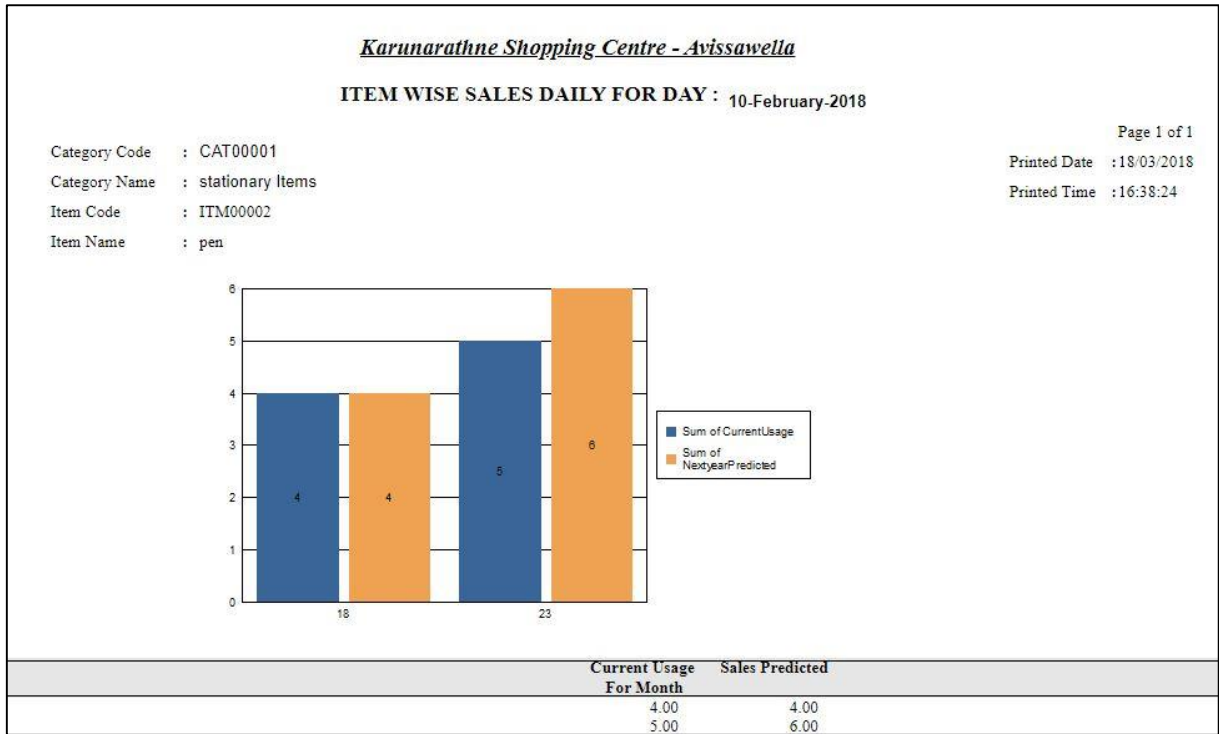


Figure 4. 7: Report and Charts

Please Refer to Appendix-F for basic code listings and the CD-ROM for the entire codes

5. USER EVALUATION AND TESTING

5.1 Introduction

Software testing is a critical component of software quality assurance that represents the ultimate analysis of specification, design, and code generation of software product. The testing method is basically combine with Verification and Validation. Validation refers to testing whether the system satisfies the requirements while verification refers to whether the system implements the specified functions properly [12]. Basic goals of test evaluation is determining whether the promises about the invention by the supplier and the requirements of the customer are met on an acceptable level.

5.2 Testing Procedure

Software testing and implementation are iterative processes. Most of the time both stages work as simultaneous system components. The propose system was tested while the development was ongoing. Following iterative RUP development life cycle made it possible to test Iterative increments of the software.

Structural testing techniques were followed in this phase including “white box” testing which tests “how a program/system does something”. [12] Functional testing techniques were used, which includes “black box” testing which tests the behavior of a system or program. These techniques were exploited in different testing levels like unit testing..

System followed object oriented methodology, so object oriented testing was also carried out in this phase. Individual operations associated with objects were tested initially, followed by testing individual classes and clusters of objects, and finally test the system as whole. User acceptance testing was completed in client site, participation of few staff members.

5.3 Test Plan and Test Cases


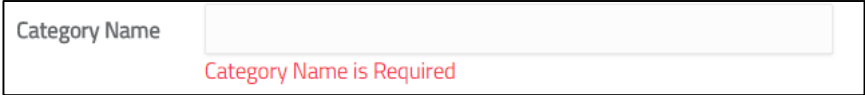
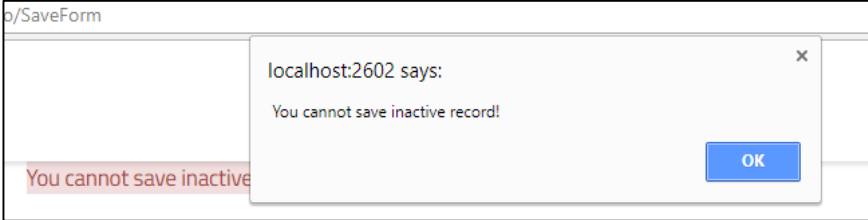
Testing begins with the implementation, code is reviewed while developing stage for testing. Test plan included all phases of testing and also used as a guide for the overall testing process. Before the system implementation, the test plan was designed. A test plan includes: test objectives, schedule and logistics, test strategies and especially test cases.

Test cases were created according to the designed test plan. That contains data, procedure, and expected result and represents which use to system or part of the system run. To reduce complexity of the testing process test cases were designed for each module independently. The following tables specify some test cases.

Manual testing method and procedures used for testing rather than automation tools and technologies.

Below tables tabulates some test cases for test basic functions.

5.3.1 Master Data modules

Test case Id	01		
Tested component	Master Data Maintenance		
Module name	Stock Item Management		
Test case	Add a new item		
Expected output	Empty fields are highlighted in yellow fields with invalid data highlighted in red colour with the relevant error messages		
Test case description			
No	Test case	Actual output	Status
01	Auto generate category code	<p>When page load new category code generated in un-editable textbox field</p> 	pass
02	Press save button without fields	<p>Mandatory fields required message display</p> 	pass
03	Save inactive record	<p>Cannot save inactive record message display.</p> 	pass
04	Save record successful	<p>Save successful message display</p>	pass

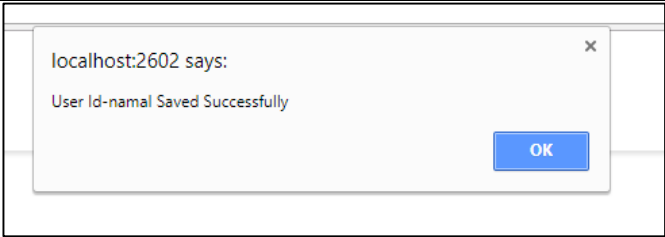
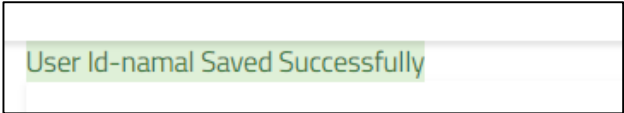
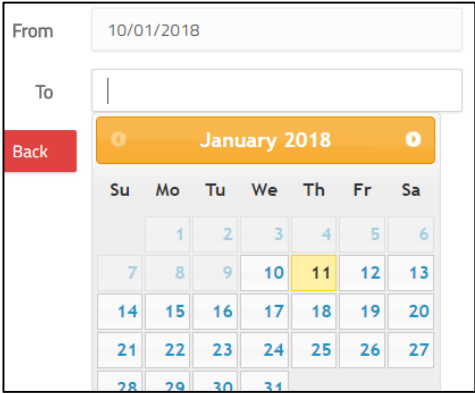
		 	
--	--	--	--

Table 5. 1: Test case for master data module

5.3.2 Report Module

Test case Id	02		
Tested component	Reports		
Module name	Report		
Test case	Report generate fields		
Expected output	When error occur display red colour with the relevant error messages		
Test case description			
No	Test case	Actual output	Status
01	Date validation	<p>To date validate according to From date. Pervious dates disable</p> 	pass
02	Press save button without fill fields	fields required message display	pass

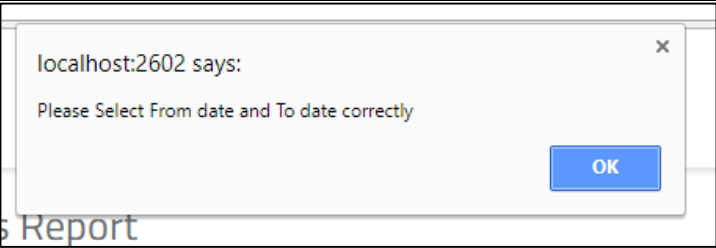
			
03	No data captured in selected range	No records to display message.	pass
04	enter feilds successf ul	successful goes to report page	pass

Table 5. 2: test case for report module

5.3.3 Sales Module

Test case Id	03		
Tested component	Sales component		
Module name	Sales Module		
Test case	Sales screen		
Expected output	Validations. Empty fields are highlighted. When error occur display red colour with the relevant error messages		
Test case description			
No	Test case	Actual output	Status
01	Stock up to date	Stock quantities update with GRNs, Unit price change according to lease updated prices of item.	pass
02	Validations	Quantity validate with system and physical values.	pass
03	Table update	When transaction complete item quantities in out update correct way.	pass
04	Customer	Already registered customer validate when enter phone number	pass

Table 5. 3: Test case for sales Module

5.3.4 Login Module

Test case Id	04		
Tested component	Login		
Module name	Login Module		
Test case	Login screen		
Expected output	User friendly view. When invalid login occur display red colour with the relevant error messages		
Test case description			
No	Test case	Actual output	Status
1	Login (positive).	Enter valid username and password. Validate and redirect to correct user home page	pass
2	Login (Negative).	Enter invalid username and password. Error message display with reason	pass

Table 5. 4: Test case for login module

5.4 Acceptance Testing

The user acceptance testing was conducted in actual working environment of Karunarathne Shopping Centre with real transaction data sets. Users were selected from the employees, and were asked to work with related modules in system according to their privileges. Activities were monitored while users working with the system.

To consider as a successful project the final outcome need to be accepted by the end users. Therefore system was handover to users for test run with questionnaire to capture the customer feedbacks. Therefor a sample questionnaire was design to give users.

5.4 Summary

After full testing cycle user can comment system was developed successfully and according to positive user reactions shows they accepted the system. When consider user acceptance test results all the functional requirements given by the client were fulfilled and the users can carry out their tasks effectively and efficiently with new system. Few minor modifications were made to the system according to the user feedback.

6. CONCLUSION

6.1 Introduction

This last chapter discuss about the objectives archived and the future encasements of the developed system. Karunaratne shopping Centre planning to become a super market within next two years. So it has been introduced a high-tech computerize systems from the beginning to reduce labor and increase productivity. As initial point they start to measure demand forecast their existing items and analyst which items they need to increase ordering and how to maintain stocks accordingly.

This system introduce for maintain stock levels with proper stock management technics and systematic demand forecasting measuring facilities. The system was built using these following technologies MVC, object oriented, C# with .NET framework, HTML, CSS and JavaScript.

6.2 Problems Encountered

From the beginning capturing the correct requirements was difficult because of the narrow IT knowledge of the client. Client was unwilling to demonstrate sensitive documents initially therefore the system analysis and design was delayed. So time needed to assign for analysis expand than planed time. Some major requirements were changed while developing the system and also database structure was changed according to changed requirements. Hence required to re modify code segments of those specific modules which consumed lot of time.

Other major problem which encountered during the development process is the lack of knowledge regarding the development tools, and languages. Online tutorials and books were used to capture the essential level of knowledge.

6.3 Lessons Learnt

The knowledge added throughout the project was actually valuable. Starting from the feasibility studies, to the end of development this process gave incomparable experience in many ways. This project gave a chance to test and implement most important theories and technologies learnt throughout the MIT degree programme. It also facilitated to learn very interesting new and updated technologies (AJAX, jQuery, Entity frameworks) in order to improve the system performance. Moreover, special efforts were taken to learn the MVC design pattern.

Furthermore working on the project encourage me to improve technical skills as well as intellectual skills by collaborating with many individuals from collective fields. Proper time

management and planning is very important aspects learn out during this project. Should carry out the work according to the schedule

6.4 Future enhancements

According to the customer acceptance test conceded in the evaluation phase, the customer feedback was optimistic and convincing. Some future enhancements are essential to make this as a fully qualified Enterprise Resource Planning (ERP) system by adding all modules.

- Add an accounting module and allow the management to generate trial balances, profit and loss statements.
- Add more effective charts in report generation module to support top management in decision making.
- Allow to access the system from anywhere for authorize persons which make easy for management.
- Online payment gate way can be introduced for online order process and online selling process.

REFERENCES

- [1] Luis A and Richard W, “Improved supply chain management based on hybrid demand forecasts”, in *Applied Soft Computing*, 2007,pp.136-144.
- [2] Chopra S and Meindl, P, “Supply chain management: Strategy, planning and operation. NJ: Prentice-Hall”,2007
- [3] C. Anley ,” Advanced SQL injection in SQL server applications,” 2002.
- [4] J. H. Rob and K. Yeasmin, “Automatic Time Series Forecasting: The forecast Package for R,”,2008.
- [5] H. Stadtler, ”Supply chain management and advanced planning—basics, overview and challenges,” in *European journal of operational research*, 163(3), pp.575-588.
- [6] L. Lapide “Demand Forecasting, Planning, and Management” Lecture to MLOG Class, Massachusetts Institute of Technologies, 2006.
- [7] P. Roger , *Software Engineering*, 5th edition, Thomas Casson, 2001.
- [8] W. Romsaiyud, "Applying MVC data model on hadoop for delivering the business intelligence", in *ICT and Knowledge Engineering (ICT and Knowledge Engineering)*, 2014, pp. 78-82.
- [9] H. Mcheick and Y. Qi, “Dependency of components in MVC distributed architecture,” in Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference, 8-11 May 2011, pp. 6-10.
- [10] I.Sommerville, *Software Engineering*, 8th edition, Addison-Wesley, 2007.
- [11] G. Booch, The Unified Modeling Language User Guide, Pearson Education India, 2005.
- [12] G. J. Myers, C. Sandler, T. Badgett, The Art of Software Testing, John Wiley & Sons, Sep 23, 2011 - Business & Economics
- [13] <https://openboxes.com/home>
- [14] <http://www.lokad.com/en/latest>

APPENDIX A - SYSTEM DOCUMENTATION

This documentation provides guidelines prior to the setup of the system. This is to assist the administrators and managers to install the system in their PC's, Servers or Workstations. In order to install the system, the chosen device should meet the following requirements of Hardware and Software.

Hardware

Hardware	Recommended Minimum Requirements
Processor	Intel Core i3 processor 2.3GHz
Memory	2GB RAM or more.
Hard disk space	Minimum 2GB free disk space or Higher.
Display	1024 X 768, High colour 32-bit display
Internet	Minimum 512kbps ADSL connection

Table A. 1: Hardware

Software

Software	Recommended Minimum Requirements
Operating System	Microsoft Windows XP/Vista/Windows 7 Or above.
Code editor	Dot Net framework 4.5 (64 bit) , Microsoft visual studio 2017
Database	SQL server 2016 R2
Web browser	Internet explorer/Fire Fox/Google Chrome
Report Generator	Cristal report VS 13.2.0
JDK toolkit	JDK 8

Table A. 2: Software

SETUP

- Install above mentioned software's in suitable hardware installed server.
- Open SQL server and create new empty database according to given name. Then copy backup database from given folder and restore backup file to newly created database.

- Copy the given host folder given in the supplementary CD and paste it inside the suitable folder in the Server. Open IIS manager and apply host folder with given database connection features.
- Run the system in browser given URL notations.
- Press login to access, by providing correct username and password.

APPENDIX B – DESIGN DOCUMENTATION

Use case diagrams and descriptions

Use-Case Description for Sales module

Use-Case	Sales
Actors	Cashier, customer
Overview	The sales items sold to the customer.
Pre-conditions	Enough stock in showroom
Flow of events	<ol style="list-style-type: none">1. Customer comes cashier with items2. If customer already in system check availability. If not asked about registering and register customer3. Sales person check items and add then into the system4. The system calculates the amount payable
Post conditions	Sales details updated to system for further analysis.

Table B. 1: Use-Case Description for Sales module

Sales module use case

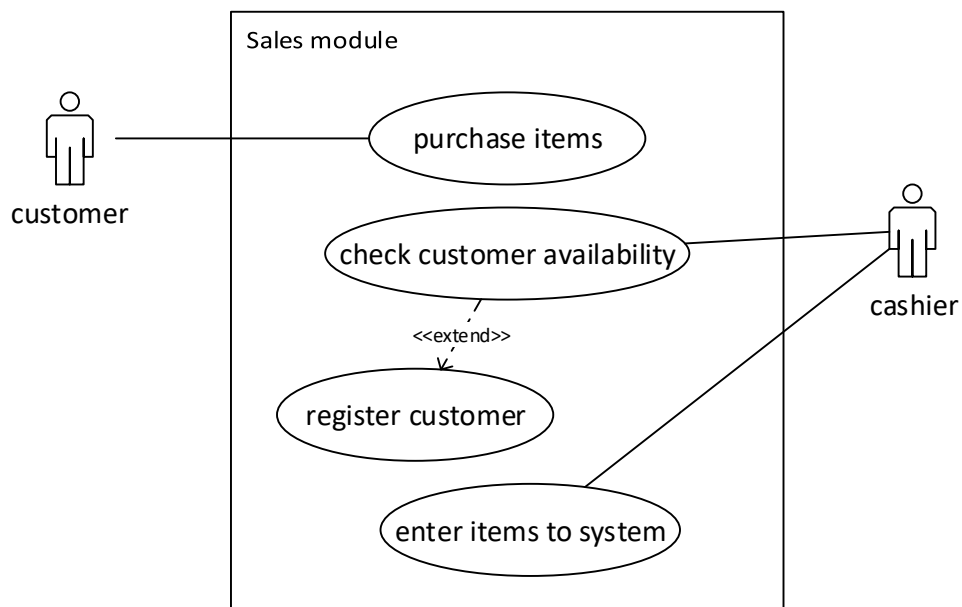


Figure B. 1: Sales module use case

Use-Case Description for purchase module

Use-Case	purchase
Actors	Officer, Supplier.
Overview	
The order details are added into the system	
Pre-conditions	
Item must have a one or more supplier	
Flow of events	
<ol style="list-style-type: none"> 1. Check re-order level reach items or select special request need items. 2. Select suitable supplier with order quantities 3. Generate purchase order and print purchase order 4. Send generated purchase order to suppliers 	
Post conditions	
Item Purchases process is maintained and monitored weekly	

Table B. 2: Use-Case Description for purchase module

Use-Case Description for good received module

Use-Case	good received
Actors	Supplier, officer
Overview	
Supplier received goods	
Pre-conditions	
Purchase order request sent to supplier	
Flow of events	
<ol style="list-style-type: none"> 1. Good received by supplier with good receive note. 2. Enter received good details according to purchase order. 3. System automatically update all table fields and if all order quantity received order closed. 4. Generate invoice and print copy send to supplier. 	
Post conditions	
System update with new received quantities	

Table B. 3: Use-Case Description for good received module

Use-Case Description for Master page Maintain module

Use-Case	Master data
Actors	Admin, manager
Overview	
Master data maintain	
Pre-conditions	
All master pages should well structured	
Flow of events	
<ol style="list-style-type: none"> 1. Authorize use enter users and user roles 2. Master pages view according to user privileges 3. Authorize person add edit update delete master data. 	
Post conditions	
All data update in database for transaction usage purposes	

Table B. 4: Use-Case Description for Master page Maintain module

Use-Case Description for report generate

This module allows the user to generate yearly monthly, daily reports, reports by ID and print them as well.

Use-Case	Report generation
Actors	Officer, manager
Overview	
Report generation for inventory maintain and analyst demand forecast	
Pre-conditions	
Generated report transaction data must entered in database table fields	
Flow of events	
<ol style="list-style-type: none"> 1. User enter appropriate fields need to generate report 2. User can print report as pdf or excel sheet when view report 	
Post conditions	
Report data use for analysis purposes	

Table B. 5: Use-Case Description for report generate

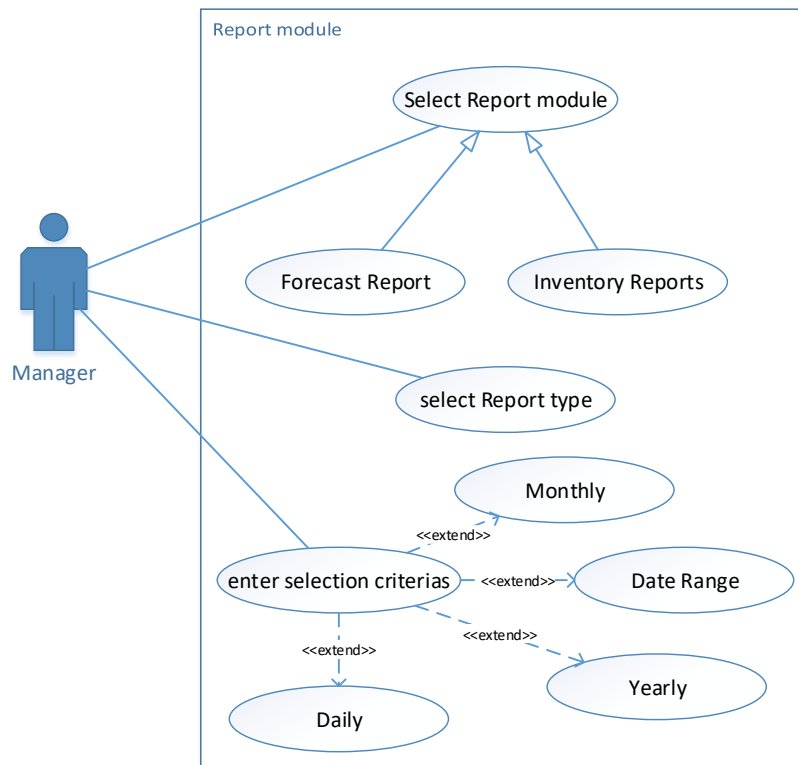


Figure B. 2: Report Use case

Sequence diagram

Diagram for Report Generation

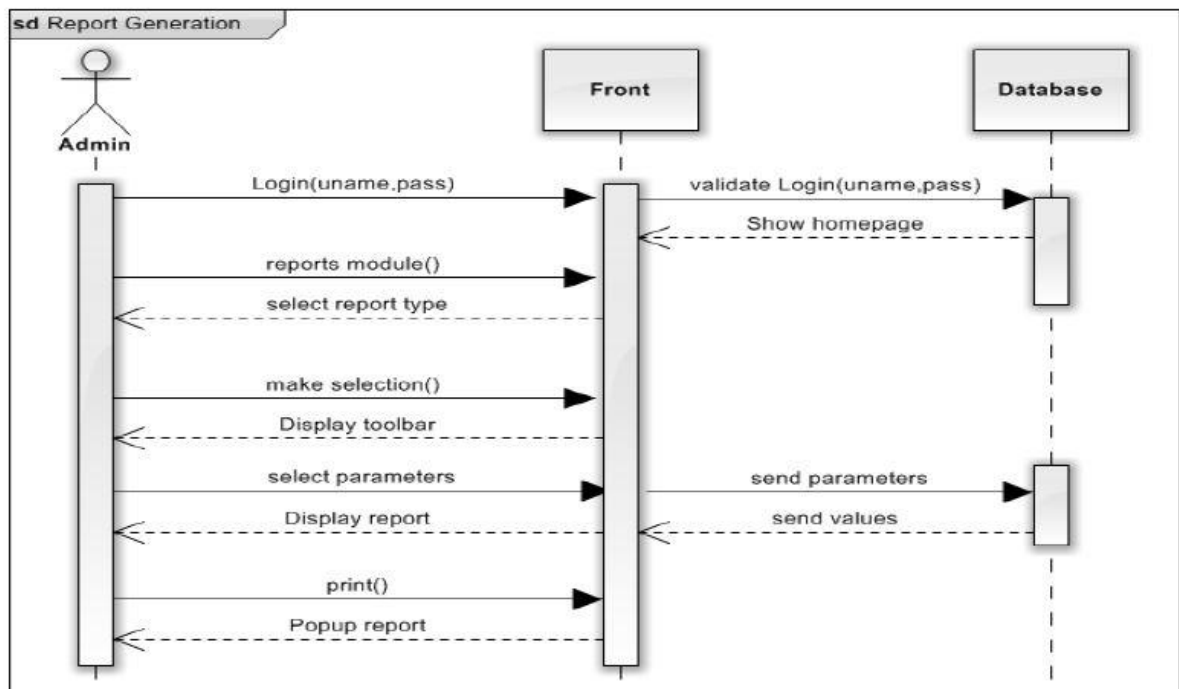


Figure B. 3: Diagram for report generation

Activity diagram

The following diagram covers activities from purchase order generation to good received.

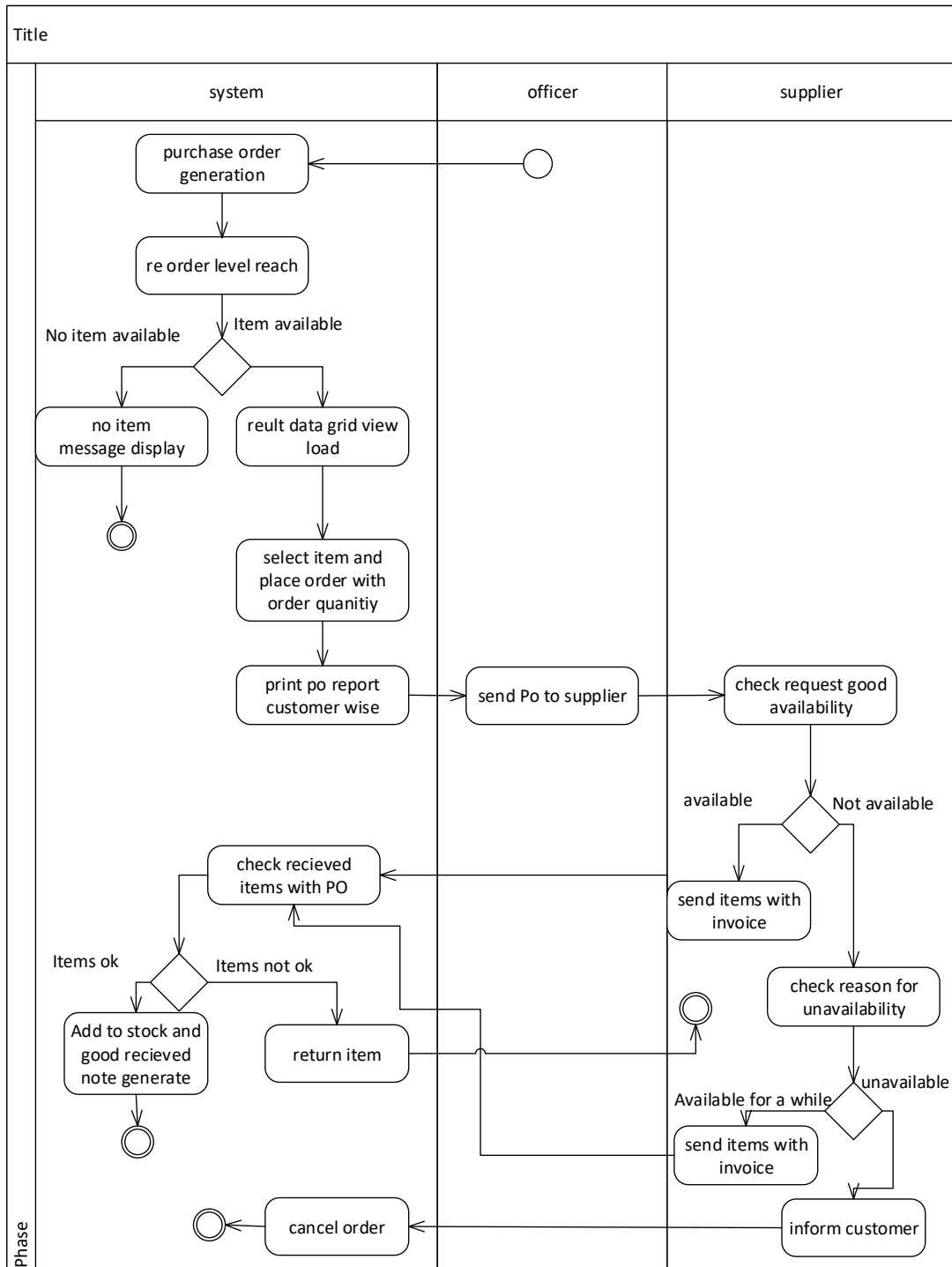


Figure B. 4: Activity diagram for PO

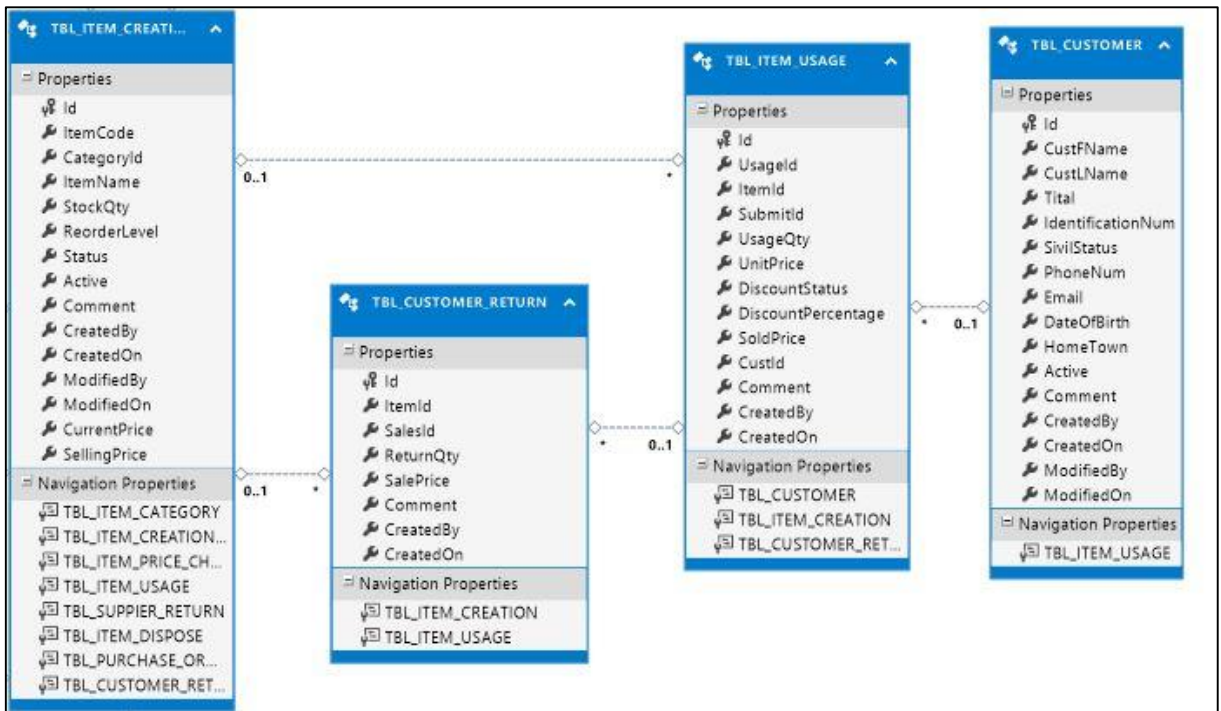
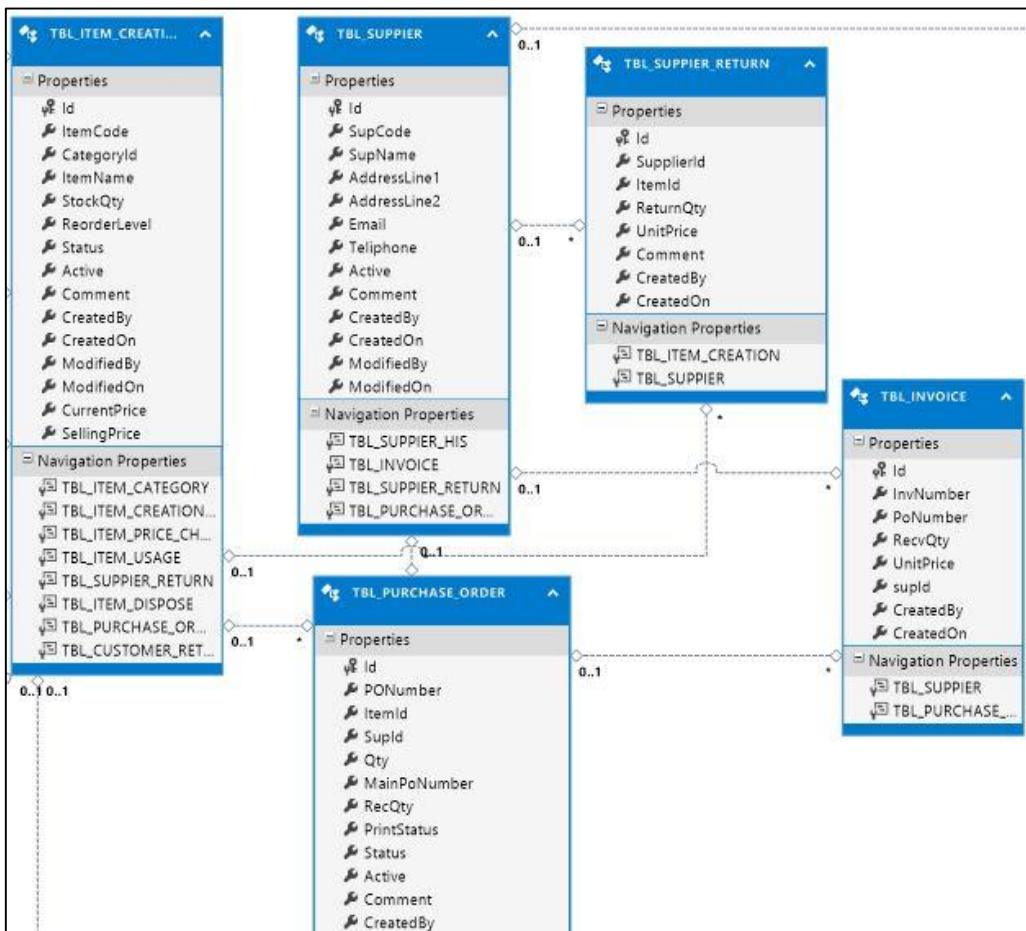


Figure B.6: Sales module ER diagram

Item Ordering module table relationship diagram display in figure B.7.



B.7: Order module ER diagram

APPENDIX C - USER DOCUMENTATION

Developed system has built for several users which are admin, manager, accountant, officer, owner and customer, the system offers wide variety of functionalities for these users. This manual is guidance for users how to use system features which helps to adhere management expectations.

Administrator has all the rights to operate all the modules. Explain all the module functions in the system, the documentation is compiled for the user who holds administrative privileges.

Getting familiar with the system

System was built in user friendly understandable interface views. Proper flow include in each module.

Typing the URL in the address bar provides the Login page (Figure C.1).

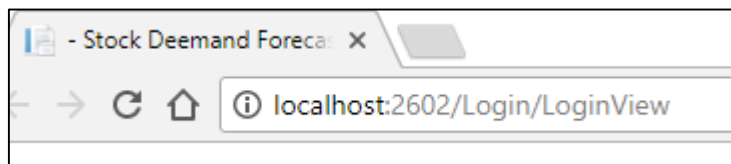


Figure C. 1: Login page

The user should provide correct username and password to gain authorised access to the system.

Once logged in, the user is directed to the home page (Figure C.2).

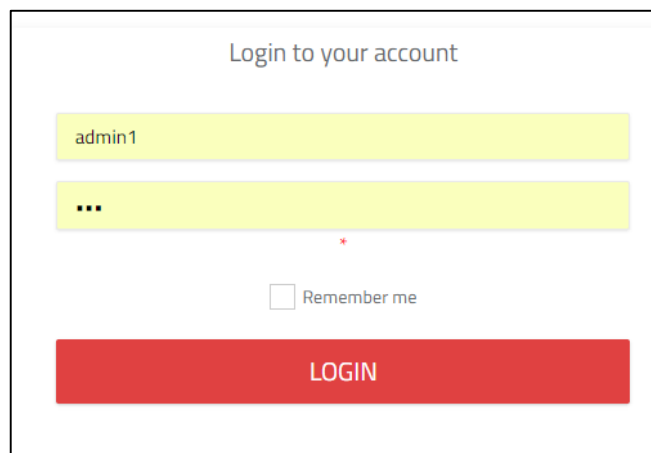


Figure C. 2: Home page

User can log in system after giving correct user name and password. Dashboard home page view display after initial login. The modules can be browsed through the menu bar drop down

in left side navigation bar. And right side top user log name navigation display log out and user profile view module. Figure C.3 display navigation bar for main modules and Figure C.4 display login user privileges.

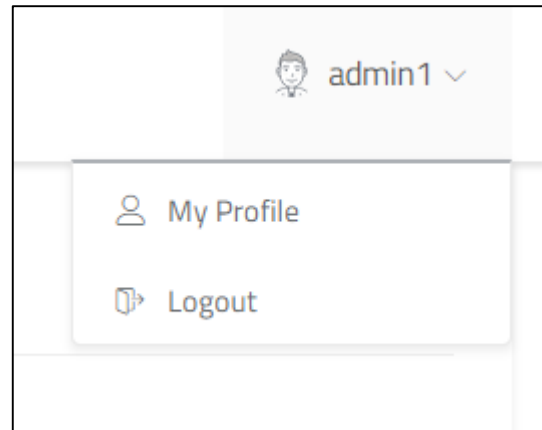
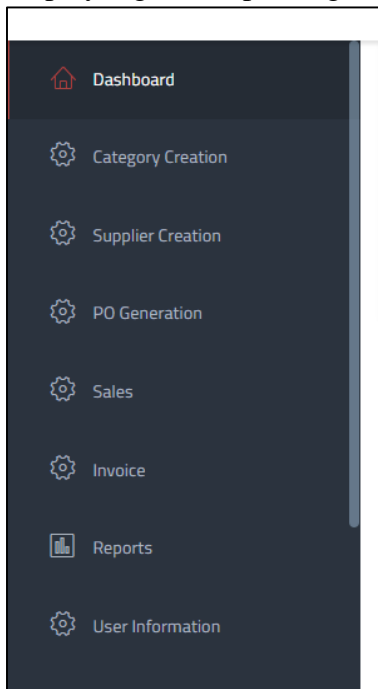


Figure C. 3: Display login user privileges

Figure C. 4: Navigation bar

General Guidelines in Using Common Form Controls

Clicking the button helps to complete appropriate function named by each button. Figure C.5 shown common button set

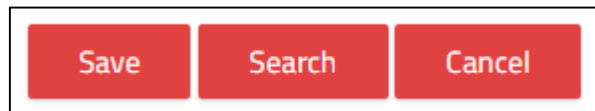


Figure C. 5: Common button set

Auto increment Id generate for main master modules and some transaction pages for give unique identification for each data. Figure C.6 display example Id generate for all category.



Figure C. 6: Id generate for all category

Common calendar display using reusable java scrip. Figure C.7 display date picker using system.

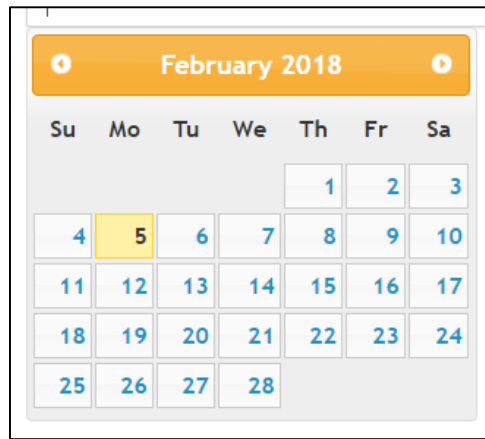


Figure C. 7: Date picker using system

General grid view use for all data search according to given filtrations. Select and history data view links include in most grid tables for easy functionalities. Figure C.8 show common grid view used in modules.

	User Id	User Name	Group Name	Active	Status	
Select	admin1	saku	ADMIN	True	SAVED	History
Select	admin2	saku	ADMIN	True	SAVED	History
Select	Manager	Manager	Manager	True	SAVED	History
Select	officer1	namal	Staff	True	SAVED	History
Select	officer2	kumara	Staff	True	SAVED	History
Select	namal	namal fernando	Customer	True	SAVED	History

Figure C. 8: Common grid view used in modules

Validations were added to popup when save or edit details without main functions. Figure C.9 display error messages in item creation view save without mandatory fields.

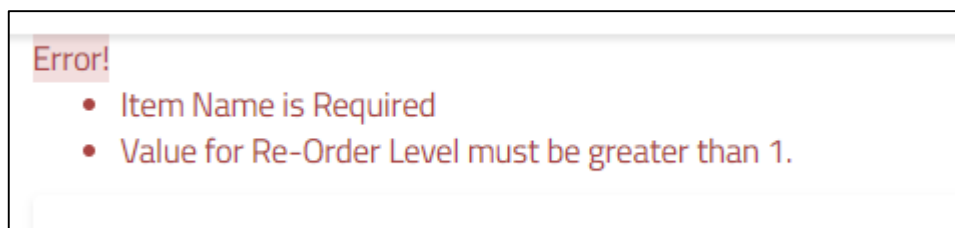


Figure C. 9: Display error messages

When save with correct details and fill all mandatory fields successfully save message display (Figure C.10) and detail view shown.

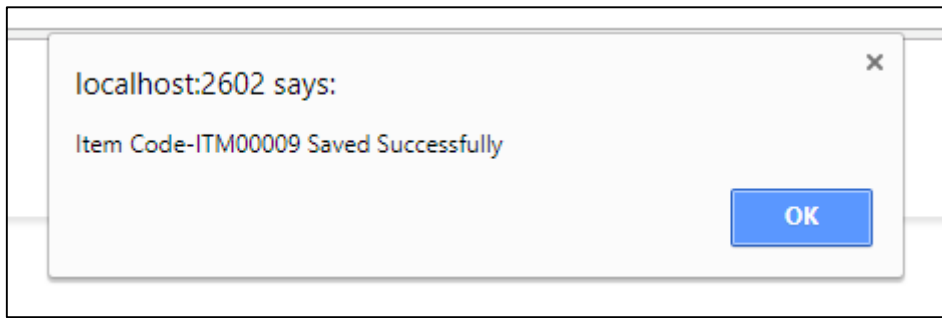


Figure C. 10: Save message display

Modules

The below section gives brief descriptions on some main modules and features. The data update and deletion has been explained in the general guidelines which are the same for all the module features of the system. Therefore only the special features add forms are explained below.

User Information module

This module use to create system users. Admin create users and when user log login user can edit his profile. As common module describe main functions included this page (figure C.11).

Figure C. 11: User Information module

Other than that email validation added when wrong email format added. (Figure C.12)

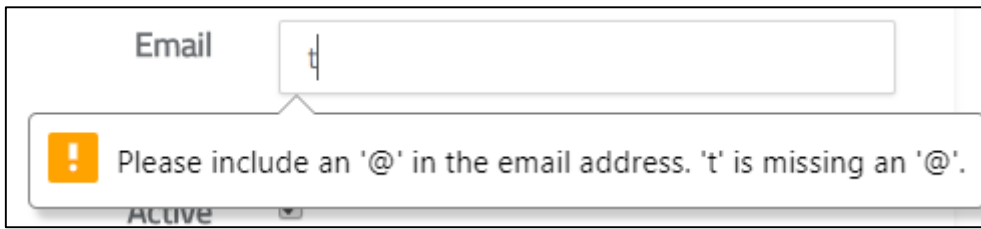


Figure C. 12: E-mail validation

When save data detail view displayed with edit button. Press edit button navigate again main view for edit and update data.

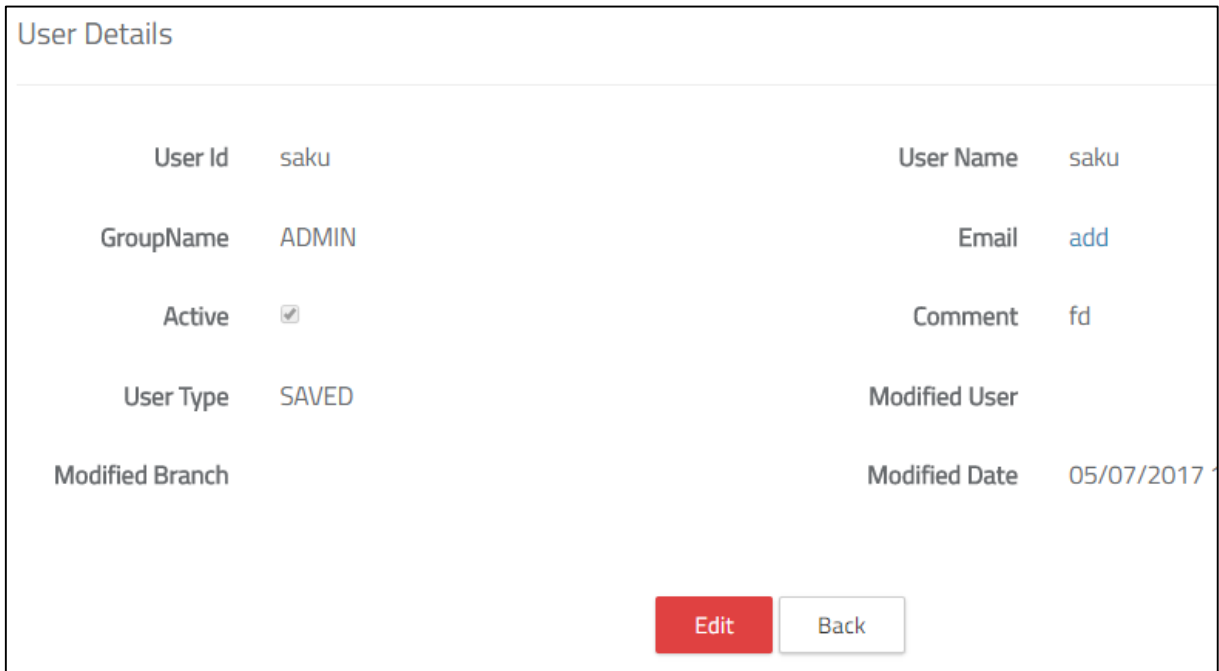


Figure C. 13: User details interface

As describe below all other master data entering modules such as Item creation, category creation, supplier creation, customer creation were operate same way.

Purchase order generation module

This module officer can generate purchase order for check item levels. Button include call re order level reach facilitate display all reorder level reach items before they exceed minimum quantity in stocks. Other than this user can manually select item and place order. Figure C.14 display this module interface.

Purchase Order Generation

PO Number: PON00002

Category Name: --Select--

Item Name: *

Supplier Name: --Select-- *

Order Quantity: 0

Unit Price: 0.00

Stock Quantity: 0

Re-order Level: 0

Buttons: Save, Search, Re-Order Reach List, Cancel

Figure C. 14: Purchase order generation module

Auto generated PO number added for each order items. Print facility given by supplier wise and end of the day user can print and send PO to suppliers.

Sales Module

This module use cashier to enter customer purchasing items to system. Basically this module can divided in to main three parts.

First part(figure C.15) check customer availability. When customer already registered customer name display after enter identification number. If new customer cashier can register customer before add sales items.

Sales Item

Customer

Customer Mobile Number: [Input Field] New Customer

Customer Name: [Input Field] 0

Figure C. 15: Sales Module

Then user can add each item to system by select item and entering quantity. When user press add button item details add grid view table and user can delete entered data if he make wrong order. (figure C.16)

Items

Usageld: SAL00011

Item Category: --Select-- Item Name: --Select--

Quantity: 0 discount percentage(%): 0.00

Current Stock Quantity: 0 Price: 0.00

Comment:

	Item Name	Item Quantity	Total Price
Select	chocolet	1	500.00

Figure C. 16: Item interface

After all items enter user can press submit button and close order. New sales order id generated automatically. When click clear button order canceled.(figure C.17) Bill generated through billing system.

	Item Name	Item Quantity
Select	pen	2
Select	pen	3

Figure C. 17: Item interface

Report module

One of the purpose of this system is report generation. This system data use for measure item demand forecast using report modules. Also main inventory control reports included. Figure C.18 display the main interface for report viewer.

Reports

Inventory Reports	Forecasting Reports
<input type="button" value="Inventory"/>	<input type="button" value="Forecast"/>

Figure C. 18: Report module

APPENDIX D – REPORTS

The system allows the administrators and managers to generate well documented reports with a filtering Yearly, Monthly, Daily, Date range reports. The report indicators helps the management to get idea about item movement and demand forecasting. These reports are given in clear format which helps the user to easily browse, view and print.

Inventory Reports

Inventory reports display separately with forecast reports. Below figure D.1 display inventory report selected view. All inventory related reports include in this table format. Printed purchase order re-printed option is added advantage in this module. Report module access permission only accept to management user group.

Inventory Reports		
Report No	Report Name	View
01	Current Stock Details- As at Date	Select
02	Sales Details -Date Range	Select
03	Purchase Order Details - Date Range	Select
04	Invoice Details- Date Range	Select
05	Supplier Return Details- Date Range	Select
06	Ex-change Details- Date Range	Select
07	Supplier Details	Select
08	Category Details	Select
09	Item Order Details- Item Wise	Select
10	Customer Details	Select

Figure D. 1: Inventory reports

Below figure D.2 shown purchase order print report for selected supplier. This report can download as pdf format and can email or print and send to supplier.

Purchase Order						Print Date : 06/02/2018
						Print Time : 3:43 pm
Supplier Name	: Santo Printers					
Supplier Address	: No 26,sen Sebastian Road,colombo 12.					
Supplier Contact No	: 075-713539/					
Supplier Email	: santop777@gmail.com					
Please supply following items on our Account						
No	Purchase Date	Item Name	Unit Price Rs/=	VAT %	Quantity	Amount
1	01/11/2016	Account Opening Activators New	2.20	0.00	10000	22000.00
2	01/11/2016	Account Opening Forms - Limited Liabilities	4.60	0.00	3000	13800.00
3	01/11/2016	Account Opening Forms - Sole Proprietorship/Partne	4.60	0.00	3000	13800.00
4	01/11/2016	Account Opening Mandate New	3.10	0.00	20000	62000.00
5	01/11/2016	Account Opening Mandate New Sinhala	3.10	0.00	5000	15500.00
6	13/10/2016	Account Opening Mandate New Tamil	3.10	0.00	2000	6200.00
7	01/11/2016	CRD 100 - Loan Agreement	7.75	0.00	500	3875.00
8	01/11/2016	Credit files	295.00	0.00	50	14750.00
9	01/11/2016	Files For Trade - Green	27.50	0.00	200	5500.00
10	01/11/2016	Files For Trade - Orange	27.50	0.00	250	6875.00
11	01/11/2016	Form No.001 - Balance Inquiry	82.50	0.00	1000	82500.00
12	01/11/2016	Form No.004 - Remittance Application Forms	255.00	0.00	500	127500.00
13	01/11/2016	Form No.010 - Cash Transfer Slip	63.00	0.00	500	31500.00
14	01/11/2016	Form No.050 - Denomination Labels	0.95	0.00	15000	14250.00
15	01/11/2016	Form No.051 - Request Form	175.00	0.00	100	17500.00
16	01/11/2016	Form No.054 Voucher Summary	1.45	0.00	500	725.00
17	01/11/2016	Form No.060 - Overdraft Agreement	7.75	0.00	1000	7750.00
18	01/11/2016	Form No.070 - General Agreement - English	16.50	0.00	5000	82500.00

Figure D. 2: Purchase Order

Stock valuation report include item wise good receive details for given time period. Unit price change according to each purchase highlighted here.

Stock Valuation						Print By100
						Print Date6/02/2018
						Print Time3:10:34PM
Item Name	Trans Date	Item Code	Quantity	Unit Price	Amount	
Gift Items						
Activation T -shirt	10/31/2016	GFI0025	98	300.00	29,400.00	
Camera	10/31/2016	GFI0004	13	550.00	7,150.00	
Caps	10/31/2016	GFI0005	500	155.40	77,700.00	
	10/31/2016		167	155.44	25,958.48	
	05/24/2017		2000	155.40	310,800.00	
	05/31/2017		499	155.40	77,544.60	
	06/09/2017		10000	178.71	1,787,100.00	
	06/07/2017		500	155.40	77,700.00	
Carry Bag	06/02/2017	GFI0003	2500	500.00	1,250,000.00	
	06/02/2017		75	250.00	18,750.00	
	06/02/2017		45	500.00	22,500.00	
	06/02/2017		55	500.00	27,500.00	
	06/02/2017		55	500.00	27,500.00	
	06/02/2017		5	575.00	2,875.00	
	06/02/2017		120	155.40	18,648.00	
	06/01/2017		20	575.00	11,500.00	
	06/01/2017		15	575.00	8,625.00	
	05/31/2017		25	575.00	14,375.00	
	05/31/2017		15	575.00	8,625.00	
	05/30/2017		18	300.00	5,400.00	
	06/07/2017		5	575.00	2,875.00	
	07/07/2017		120	155.40	18,648.00	

Figure D. 3: Stock valuation

Forecast Reports

Forecast report generation is the one of main feature in this system development. This reports display both printed and graphical views. Forecast view reports divided into areas such as item wise sales demand, top sales items, low sales items, customer base item demand, supplier base item demand etc.. Below figure show screen of selection table.

Report No	Report Name	View
01. Item wise Sales Demand		
1.1	Daily	Select
1.2	Date Range	Select
1.3	Monthly	Select
1.4	Yearly	Select
1.5	Date Range - Hourly	Select
02. All Sales Details		
2.1	All Sales in Year	Select
03. Top Sales Item Details		
3.1	Daily	Select
3.2	Date Range	Select

Figure D. 4 : Forecast view table

Each select redirect to special page which have select criteria's such as date range, month, year or day. According to selected fields reports were view with print facility. Below figure D.5 display selected item monthly sales with next year predicted quantities.

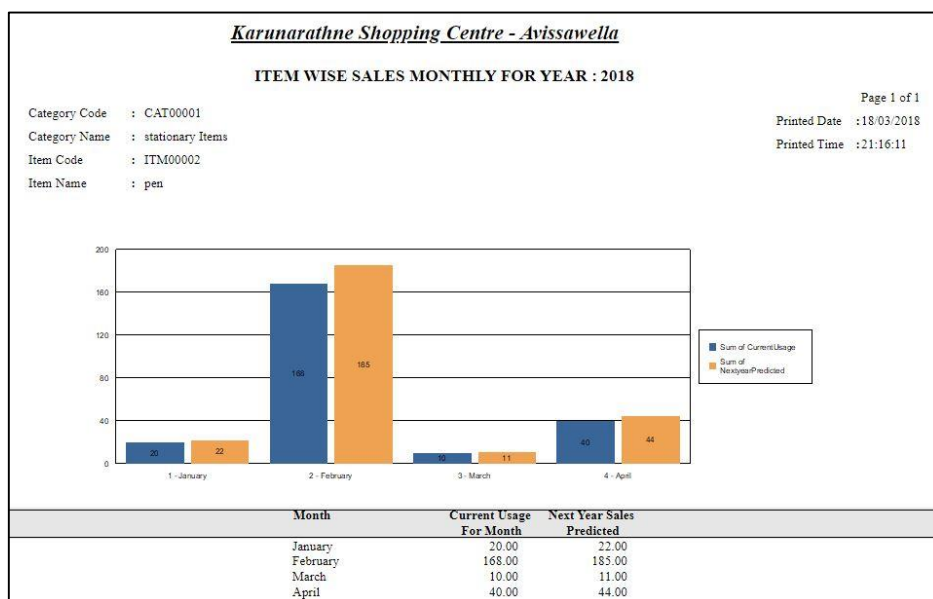


Figure D .5 Selected Item Monthly Sales

Figure D.6 display sales of all items in selected year and also there is a field for predicted value for next year sales. Therefore management can decide next year ordering quantities using this values.

<i>Karunaratne Shopping Centre - Avissawella</i>							
Monthly Item Demand For Year: 2018							
Page 1 of 1							
Printed Date : 18/03/2018							
Printed Time : 21:07:08							
Category Name	Item Code	Item Name	Unit Price	Current Usage For Month	Cost	Next Year Sales Predicted	
April							
CAT00001	stationary Items	ITM00002	pen	85.00	40	3,400.00	44
Monthly Total :						3,400.00	
February							
CAT00001	stationary Items	ITM00002	pen	45.00	159	7,155.00	175
		ITM00002	pen	80.00	4	320.00	4
		ITM00002	pen	85.00	5	425.00	6
CAT00002	Cosmetics	ITM00001	revlon	480.00	2	960.00	2
		ITM00001	revlon	490.00	7	3,430.00	8
CAT00003	Fancy Items	ITM00003	chocolet	40.00	3	120.00	3
		ITM00003	chocolet	490.00	3	1,470.00	3
		ITM00003	chocolet	500.00	1	500.00	1
		ITM00005	biscuits	56.00	2	112.00	2
CAT00004	Footware Items	ITM00011	table cloth	400.00	4	1,600.00	4
Monthly Total :						16,092.00	
January							
CAT00001	stationary Items	ITM00002	pen	45.00	20	900.00	22
Monthly Total :						900.00	
March							
CAT00001	stationary Items	ITM00002	pen	85.00	10	850.00	11
Monthly Total :						850.00	
Grand Total :						21,242.00	

Figure D. 6: Forecast report-Sales Demand of Year 2018

Below report display reconciliation of stock for given date period. Started balance to end balance of selected date range was displayed with all stock changers.

		Print By : 100
		Print Date : 06/02/2018
		Print Time : 3:22:57PM
From Date	: 05/07/2017	
To Date	: 14/02/2018	
<u>Store Stock Reconciliation Report</u>		
<u>Description</u>	<u>Amount Rs/=</u>	
Opening Balance	25,454,420.50	25,454,420.50
Total Purchase for the month	21,646.50	
Total Branch Return for the month	768.06	
Total Item Reject From Branch	27,337.60	
Addition of stock Adjustment	471.90	50,224.06
Total Branch Allocations for the Month	41,994.56	
Total Supplier return for the Month	1,111.46	
Total Damage stock value for the Month	114.10	
Reduction of stock Adjustment	12,351.24	55,571.36
Closing Balance	25,449,073.20	25,449,073.20
Branch Pending Acknowledge	220.00	220.00
Grand Total	25,449,293.20	25,449,293.20

Figure D. 7: Reconciliation Report

Item wise stock change in given month, month wise, year wise and selected date range was display for get idea for future ordering purposes.

Below figure shown category wise sales on given year.

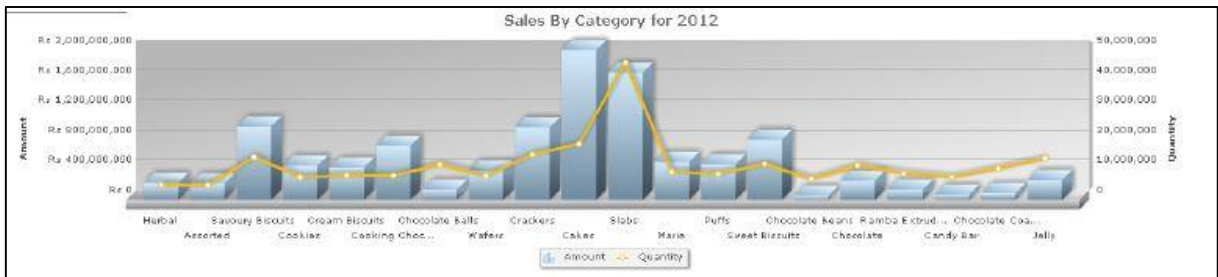


Figure D. 8: Sales on given year

APPENDIX E - TEST RESULTS

This system developed on MVC architecture generally supports us in code reuse, hence same error messages were re-used in appropriate places. Test cases were used to get check system with appropriate errors and responses. Chapter 5 describe main test cases included in project test plan. Rest was discuss here.

Common Functions

Test case Id	05		
Tested component	Common		
Module name	common		
Test case	common		
Expected output	Satisfy common system functions need to run system accurately.		
Test case description			
No	Test case	Actual output	Status
01	Login time out.	Idle for 15 minutes logout automatically	pass
02	Logout.	Click logout button. Logout user and direct to the login page.	pass
03	Test with browser	Access & load system in different browsers (Firefox, Chrome, Opera, IE, Safari).	pass
04	DB connection	Give wrong DB user name/ password/ DB name in db helper config file. Display db error message	pass

Table E. 1: Common Functions

Item Creation


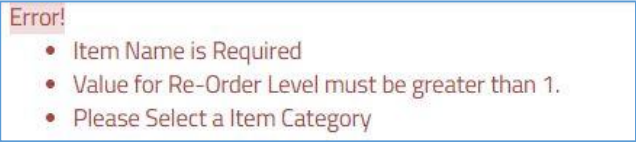
Test case Id		06	
Tested component		Item Creation	
Module name		Item Creation	
Test case		Item Creation	
Expected output		Successfully save item and error messages display when wrong data enter.	
Test case description			
No	Test case	Actual output	Status
01	Mandatory fields highlighted.		pass
02	Save without mandatory fields	Error message display and return to data view 	pass
03	No data captured in selected range	No data message display	pass
04	Re order level and enter stock validate	When initial item create re order level quantity should not validate with item quantity.	pass

Table E. 2: Item Creation

Purchase Order


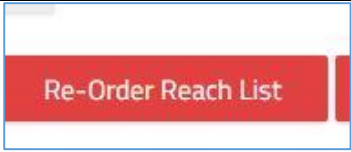
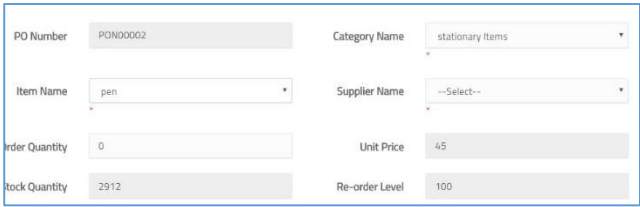
Test case Id		07	
Tested component		PO Creation	
Module name		Po Creation	
Test case		Purchase order Creation	
Expected output		Successfully save create PO and error messages display when wrong data enter.	
Test case description			
No	Test case	Actual output	Status
01	Auto generated Id create for each order.		pass
02	Re order level reach items automatically display button click	 If no item to show message display	pass
03	No data captured in selected range	No data message display	pass
04	Automatically fill fields when item select from item dropdown	Item dropdown enable after item category select. 	pass

Table E. 3: Purchase order

Acceptance Test Results

Main purpose of the system built is to satisfy the customer requirements, the final system should be accepted by the end users to complete project as successful project. Thus a questionnaire was designed and make system assess to a set of the users by capturing the customer feedbacks on the survey. The following is a sample questionnaire created.

Stock Management and Demand Forecasting System – User Evaluation Form

Date :	Name :	Designation :
--------	--------	---------------

I appreciate your help in evaluating new software to your shop by filling short answers to below questions. Put “X” to appropriate column.

#	Question	Strongly Agree	Agree	Disagree	Strongly Disagree
1	Interfaces are simple and understandable				
2	System display information more familiar				
3	Easily navigate through system				
4	Functions are more reliable				
5	System provide sufficient information				
6	Degree of Information provided in Reports.				
7	Ability to maintain data, to keep it up-to date.				
8	Ease of entering/handling form fields, and handle manipulations.				

Comments

Thank You.

Table E. 4: Acceptance test results

The Whole Feedback ratings given by each user were taken and converted into a graphical evaluation chart. The following bar chart in Figure E.1 display the overall feedback.

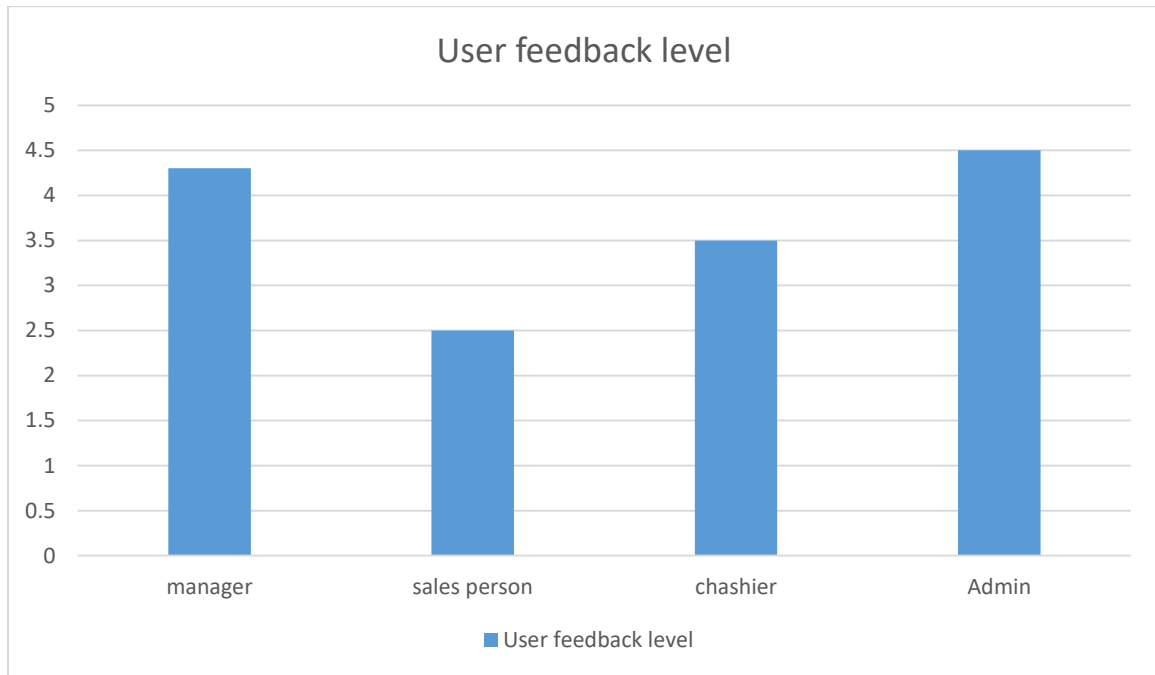


Figure E. 1: Overall feedback

The final feedback received from the survey was significantly positive. Some suggestions were taken and hope to consider them at the implementation stages.

APPENDIX F – CODE LISTINGS

This section use to give fundamental impression about the code structure of the system. Since the codes are very lengthy, only important code snippets are represented in this segment. Developed system has adapted MVC design pattern. Codes related to each function, comprises of separate Model View and Controller. Some main code segments are described below.

As example below describe code segments for Item Category creation view MVC architecture. All other pages were designed base on this pattern. Green colour commented lines describe reasons for the each feature in code.

View/Controller/Model

View:

The user interacts with the system through this view. View also displays error messages passed from the model=>controller.

```
<div class="form-horizontal">
  @*validation check*@
  @Html.ValidationSummary(true, "", new { @class = "text-danger" })
  <div class="clearfix">&nbsp;</div>
  <div class="row">
    @*name lable create here*@
    @Html.LabelFor(m => m.ItemCategory.CategoryCode, htmlAttributes:
    new { @class = "control-label col-md-2" })
    <div class="col-md-6">
      @*text box for data enter*@
      @Html.EditorFor(m => m.ItemCategory.CategoryCode,
      new { htmlAttributes = new { id = "txtCatCode", @readonly = "readonly",
      @class = "form-control" } })
      @*validate text box*@
      @Html.ValidationMessageFor(m => m.ItemCategory.CategoryCode, "",
      new { @class = "text-danger" })
    </div>
  </div>
</div>
```

Controller:

Controller work as a messenger between the models and views. It gets the requests from view pass to model get responses. Then updates the view according to the results. The statements below show how files are loaded.

```

public class ItemCatController : Controller
{
    // call model class
    public static ItemCategoryViewModel masterDataViewModel = new ItemCategoryViewModel();
    IdCounterCRTLLogic idCounterCRTLLogic = new IdCounterCRTLLogic();
    CategoryCRLLogic categoryCRLLogic = new CategoryCRLLogic();
    LoginInformation login = new LoginInformation();
    bool success = false;
    // main areas need to display when visit page
    0 references | 0 requests | 0 exceptions
    public ActionResult MainView(string mode) [...]
    //capture user enter values and pass it to model
    0 references | 0 requests | 0 exceptions
    public ActionResult SaveForm(ItemCategoryViewModel ugvm) [...]
    // search database saved values
    0 references | 0 requests | 0 exceptions
    public ActionResult SearchForm(ItemCategoryViewModel ugm) [...]
}

```

Model:

The model is where the system carries out the CRUD operations in the database tables. The following code fragment shows the code for insertion of a record.

```

TBL_ITEM_CATEGORY CATINF = new TBL_ITEM_CATEGORY();
CATINF.CategoryCode = catInfo.CategoryCode;
CATINF.CategoryName = catInfo.CategoryName;
CATINF.Active = catInfo.Active == true ? 1 : 0;
CATINF.CreatedOn = login.LoginDate;
CATINF.CreatedBy = login.UssrId;

db.TBL_ITEM_CATEGORY.Add(CATINF);
db.SaveChanges();

```

The following code fragment shows the selection of a record from database.

```

ItemCategoryViewModel ugvm = new ItemCategoryViewModel();
try
{
    using (ARAEEntities db = new ARAEntities())
    {
        string selectQuery = "";
        selectQuery = "select * from TBL_ITEM_CATEGORY where Active = '" + (catInfo.Active == true ? 1 : 0) + "'";
        var selectList = db.TBL_ITEM_CATEGORY.SqlQuery(selectQuery);
        foreach (var item in selectList.ToList())
        {
            ugvm.ItemCategory = new ItemCategoryView();
            ugvm.ItemCategory.Id = (int)item.Id;
            ugvm.ItemCategory.CategoryCode = item.CategoryCode;
            ugvm.ItemCategory.CategoryName = item.CategoryName;
            ugvm.ItemCategory.Comment = item.Comment;
            ugvm.ItemCategory.Active = item.Active == 1 ? true : false;
            ugvm.ItemCategory.CreatedBy = item.CreatedBy;
            ugvm.ItemCategory.CreatedOn = DateTime.Parse(item.CreatedOn.ToString());

            ugvm.itemCategoryList.Add(ugvm.ItemCategory);
        }
    }
}

```

Reusable components

Below script use to auto generate Item code, category code, purchase order number, GRN number and supplier number.

```
<script>
    $(document).ready(function () {
    var txID = $("#txtCatCode");
    var value = "1";
    $.ajax({
        type: 'Post',
        url: '@Url.Action("GetCategoryCode", "ItemCat")',
        data: '{value: "' + value + '"}',
        contentType: 'application/json; charset=utf-8',
        success: function(data) {
            $('#txtCatCode').val(data.IDNumber);
        }
    });
    });
</script>
```

Common date picker java script use for all reports which use to display calendar date selection functions.

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

    $("#txtFrom").datepicker({
        dateFormat: "dd/mm/yy",
        maxDate: "+60D",
        numberOfMonths: 1,
        onSelect: function (selected) {
            $("#txtTo").datepicker("option", "minDate", selected)
        }
    });
</script>
```

GLOSSARY

CSS (Cascading Style Sheet) - is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a mark-up language.

Database - is an organized collection of data for one or more purposes, usually in digital form.

Graphical User Interface - is a type of user interface that allows users to interact with electronic devices with images rather than text commands.

Internet - is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.

JavaScript - is a prototype-based, object-oriented client-side scripting language which is dynamic. It is also considered as a functional programming language.

JQuery - is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML.

Object Oriented Development - is a standard approach to software development based on objects and its instances

Structured Query Language - is a database computer declarative language designed for managing data in relational database management systems (RDBMS).

Unified Modeling Language (UML) - is a standardized general-purpose modeling language in the field of object-oriented engineering. This includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.

Web Browser - is a software which allows the user to access WWW.

WWW (World Wide Web) - is a system of interlinked hypertext documents accessed via the Internet.

INDEX

A

acceptance, 43, 48, 49
Activity, vii, viii, 28, 58, 59
analysing, ii, iii, v, xii, 26, 30

B

black box, 43
business, ii, xii, 13, 14, 17, 20, 27, 31, 51

C

C#, ii, 20, 28, 36, 37, 49
Centre, ii, iii, xii, 13, 17, 48
chapter, 15, 22, 48
Charts, viii, 37, 41, 42
Class, vi, 28, 32, 51
Client, vi, 35, 49
Controller, ii, xi, 27, 28, 29, 38, 80
Cristal report, 36, 52
customer, xii, 14, 18, 26, 31, 43, 47, 48, 49, 54, 62, 67, 68, 71, 77

D

database, viii, 15, 20, 27, 28, 31, 36, 38, 41, 49, 53, 56, 57, 60, 81, 83
demand, ii, v, xii, 13, 14, 17, 18, 22, 25, 26, 30, 48, 50, 57, 69, 71
design, ii, vi, viii, 15, 18, 22, 27, 28, 31, 33, 34, 36, 37, 43, 48, 49, 60, 80
diagram, vi, vii, viii, 23, 24, 31, 32, 38, 57, 58, 59, 60, 61

E

Entity Relationship, xi, 31
ER, xi, 31, 60, 61

F

feedbacks, iii, 14, 31, 48, 78
forecasting, ii, v, viii, xii, 14, 17, 19, 20, 21, 22, 25, 26, 30, 40, 48, 69
framework, ii, viii, 20, 27, 29, 36, 37, 49, 52

H

hardware, 35, 53
historical, 14, 17, 18, 21, 25, 30

I

inventory, ii, xii, 13, 14, 16, 17, 18, 25, 30, 57, 69
issue, v, 25, 30
items, ii, xii, 13, 14, 17, 18, 24, 25, 26, 30, 34, 48, 54, 55, 67, 68, 71, 72, 77

J

JavaScript, xi, 20, 37, 41, 49, 83

M

management, ii, iii, v, xii, 13, 16, 17, 19, 21, 25, 26, 29, 30, 31, 48, 49, 50, 51, 62, 69, 72, 83
measure, ii, 13, 14, 22, 25, 26, 30, 48, 69
Model, ii, xi, 27, 28, 29, 38, 80, 81
MVC, ii, viii, xi, 20, 27, 29, 38, 49, 51, 75, 80

N

Network, vi, 37
Non-functional, v, 26

O

Object Orient, xi, 28
ordering, ii, xii, 14, 24, 25, 30, 48, 72, 74

P

patterns, 22, 25, 30
predicting, 19
project, iii, 13, 14, 15, 21, 22, 36, 48, 49, 75, 78
Purchase, v, vii, ix, xi, 22, 30, 56, 67, 70, 77
purchase orders, 17, 25, 30

R

Report, vi, viii, ix, x, 14, 37, 40, 41, 42, 45, 46, 52, 57, 69,
74
requirements, iii, v, x, xii, 15, 18, 22, 23, 24, 25, 26, 35,
36, 41, 43, 48, 49, 52, 77

S

Sales, vi, vii, viii, ix, x, 14, 24, 46, 47, 54, 55, 60, 61, 68,
73, 74
Sequence, vii, 28, 57
software, viii, x, 15, 18, 19, 20, 27, 28, 35, 36, 37, 43, 53,
78, 83
SQL, ii, xi, 20, 36, 51, 52, 53
stock, ii, iii, xii, 13, 14, 17, 18, 19, 21, 22, 24, 25, 26, 30,
48, 54, 73, 74, 76

T

techniques, ii, v, vi, 15, 17, 21, 22, 23, 28, 36, 43, 83
test, xi, 43, 44, 48, 49, 75, 79
Test case, 44, 45, 46, 47, 75, 76, 77

U

UML, ii, xi, 28, 83
Use case, vii, 24, 54, 57
User Interface, xi, 33, 83

V

View, ii, viii, xi, 20, 27, 28, 29, 40, 80

W

white box, 43

Y

yearly, 14, 56