COMPUTER SHOP MANAGEMENT SYSTEM FOR WISDOM COMPUTER TECHNOLOGIES

M.W.A. CHANDRAWANSHA

R141303

1413031

Name of the Supervisor: O.K.A. Pathmasiri

November 2017

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

DECLARATION

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

Signature of Candidate:.. Date: 07/11/2017

Name of the Candidate: M.W Athula Chandrawansha

Signature of Supervisor(s)/Advisor(s):.. Date: 08/11/2017

Name of Supervisor(s)/Advisor(s): O.K.A.P. Mohottala

ABSTRACT

Wisdom Computer Technologies is a shop in Balangoda. When selling computers, parts, accessories and repair computers of their customers. In this shop existing system is manual system. All day to day activities done and handled on paper based. The shop employees and their customers time waste, inefficient to do some activities and sometime more hard. For all these existing system problems and fill the all other requirements propose computer shop management system as a solution.

The computer shop management system is solved all those problems and increase efficiency and effective of their works. Proposed system provide facilities such as customers, suppliers and all items details keep save, selling and repair items, maintain stock, generate reports, calculate income and expenses etc.

This project consist following modules. Master data handling module, category manage module, inventory maintain module for stock, sales module for direct sales and higher purchases, account module for calculate incomes expenses, warranty claim module, repair pc module, notification module, works history module, access control module and reporting module.

Develop computer shop management system use RUP frame work. Used UML for design system with full fill client requirements. Draw Use Case, Class, Activity, Sequence and ER diagram. For data base design based on RDBMS methodology.

For implement the project use ASP.NET Entity Framework with code fist. For UI handling part and validations done by use Java Script, JQuery and Ajax. Use HTML, CSS and bootstrap for design UI in this project. The system create based on MVC architecture. C# use based on OOP concept in controller. Visual Studio is use as IDE of this project. Crystal report XI use for reporting. MSSQL Server use as DBMS. LINQ Queries use for call to database. Use ISS Server and web browser to run project local host.

After completion project tested by client and other acceptable users. Then handed over successful computer shop management system to the client for enhance their works.

ACKNOWLEDGEMENT

First of all I would like to express my heartfelt gratitude to BIT Coordinator and the Project Examination Board of the University of Colombo School Of Computing for giving me this precious opportunity to follow a world recognized degree program.

My heartfelt gratitude also goes to Project Supervisor, Mr. O.K. Amila Pathmasiri for his correct guidance and great assistance.

Finally I would like to express my special thanks to Client and the managing director of the Wisdom Computer Technologies, Mr. W.A.W.N. Chandrasekara and staff members for giving me the great support and for dedicating their valuable time to help me for successful completing final year project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

IDE	Integrated Development Environment
SQL	Structured Query Language
MSSQL	Microsoft SQL
C#	C-Sharp
NF	Normalization Form
DB	Database
DBMS	Data Base Management System
RAD	Rapid Application Development
NIC	National Identity Card Number
MVC	Model View Controller
OOP	Object Oriented Programming
UML	Unified Modeling Language
PDF	Portable Document Format
ER	Entity Relationship
SDLC	Software Development Life Cycle
CSS	Cascading Style Sheet
IIS	Internet Information Services
ASP	Active Server Pages
PC	Personal Computer
HDD	Hard Disk Drive
RAM	Random Access Memory
Bit	Binary Digit
GRN	Goods Received Note
PO	Purchase Order
NIC	National Identity Card
SSADM	Structured System Analysis and Design Methodology
RUP	Rational Unified Process
UI	User Interface

CHAPTER 1 – INTRODUCTION

1.1. About Wisdom Computer Technologies

Information Technology is the most information factor in this era. Many organizations provide different type of technologies and services for their clients. Clients get services form service providers to achieve their targets more easily and efficiency adapting new technologies.

Wisdom Computer Technologies is supplies various type of IT related equipment's for various type of clients in Balangoda area. Computer servicing and repairing is doing in additionally. This organization opened every day except holidays and other special holidays.

This organization buying branded and reconditioned computers and other IT related equipment's from suppliers and dealers. Selling items for individual clients or organizations as whole sales or retails as direct sales or hire purchases. Currently they have large customer crowed around Balangoda area.

The managing director is owner of this organization. He is final decision maker. Other employees are working under supervision by him. The organization need to handle huge amount of data, activities and make decisions effectively.

1.2. Importance of Computer Shop Management System

As a business organization, Wisdom Computer Technologies faced lot of challenges as internal and external. This organization adopted new technologies to achieve their targets wining challenges.

Internally, the business environment is day to day growing with purchasing and selling. They happened to face difficulties handle large amount of data and activities with in minimum time. Externally, wisdom's needs to competitive with others improving businesses.

Through the web based computer shop management system they can solve all problems of current system more accurately.

1.3. Motivation of Project

The web based system is beneficial to owner and all other different level employees of the wisdom computer technologies. The Wisdom Computer Technologies currently done their activities in a manual way. This manual system is very inefficient, time consume and heavy paper works day to day. In current system their services are very poor. Following problems are encountered in current existing manual system.

There is no methods to maintain customers, suppliers, employees, guarantors and stock items information in systematic way. Stock physical counting dispatching problem and damage item returning process more complex. Hard to knowing available items, its quantities and to provide item description quickly.

Prepare purchase orders, goods received note and sales for their customers and suppliers process are more complex in current. There is no systematic way to generate and print purchase orders, goods received notes, higher purchase agreements, cash receipts and invoices. Repairing also same, because there is no maintain queue for repair accepted computers with in detail.

Lower responsibility and poor security are types of problem in day to day activities and transactions in the existing system. Difficulties of validate and conform, warranty provide periods, customers and items are problems related warranty claim. Check expiring problems and there is no method to check and calculate receivable payments quickly. Cannot be calculate the company incomes and expenses in efficient way.

Due to all above mentioned existing problems successfully, propose a computer shop management system for this organization. It will minimize time consume and cost day to day activities furthermore increasing the efficiency. The proposed system is simplest all interactions and provide simple environments to do employee works easily.

1.4. Objectives Of The Project

Create web base database to store system data properly. Improve system accessible anywhere using any devices. As well as minimize time consume, reduce paper works.

- Minimize time to create invoice, quotations and keep high consistency of the employee works.

- Increase efficiency of register employees, register suppliers, register customers and add items into stock.
- Improve communication methodology between system and clients.
- Minimizing complexity of day to day activities of the system easy to use.
- Provide reports for effective decision making.
- Manage users and privileges to protect system security.
- Keep track items movement, price history and works history for easily monitoring.
- Calculate income - outcome, lost - profits for achieve future targets.

1.5. Scope Of The Project

Scope of the project is represent system boundary clearly.

- Maintain customers, suppliers, guarantors, items categories and stock items details. Improve find, filter and communicate with clients.
- Adjust stock item quantity, special removals, set opening balance and provide item movement, filter reorder level items in the stock.
- Manage direct sales, hire purchase sales. Calculate total, installments and rates by system. Payments handle trough card, cash or checks
- Manage purchase orders, goods received notes for purchase orders based on a purchaser order can have one or many goods received notes.
- Handle repairing computers include making repair accept note, repair description and warranty claims.
- Manage system users, assign privileges.
- Notify receivable payments, expiring checks, reorder level items, receivable items, accepted repairs and warranty completing dates.
- Reports for master data, transactions, repairing and inventory bellow shown. As well as profit - lost, incomes - expenses, price history and user working history.

1.6. Structure Of The Dissertation

This dissertation shown overall descriptions about the proposed computer shop management system. Dissertation structure shown following chapters.

The introduction of the project is shown overall brief description about the proposed computer shop management system using existing system proposed system, scope of the project and objectives.

The system is analyze based on clearly gathered user requirements. In this chapter finally choose best solution form alternative solutions to design.

Design system using use case, entity relationship and sequence and class diagrams. The proposed system of computer shop management system user interface design process is late. Not included image of user interface design process in proposed system.

Provide description about used coding styles, tools and techniques to develop the project and required environment to run this project on real world.

Under the evaluation shown high level test plan, details about software testing with software testing methodologies.

Provide description about future improvements of the project with lessons and newly learnt things during doing this overall project completion.

Show more details about the content in detail of the dissertation chapters. Those details are not included in to bellow chapters.

CHAPTER 2 - ANALYSIS

2.1. Introduction

“The act, process, or profession of studying an activity typically by mathematical means in order to define its goals or purposes and to discover operations and procedures for accomplishing them most efficiently.” [www1]

System analysis phase is one of a very essential in software development process. Gathering requirements is the main attraction of this phase. It will help to get overall image of the system and produce a high level description of the system. To get the system domain completely and correctly system analyze phase is very important. End of the analysis phase require full fill client’s requirements respectively. Overview current system, determine proposed system functional and non-functional requirements and review existing similar systems which is related to the proposed system using differentiate fact gathering methodologies.

2.2. Fact Gathering Techniques

“Fact finding is process of collection of data and information based on techniques which contain sampling of existing documents, research, observation, questionnaires, interviews, prototyping and joint requirements planning. System analyst uses suitable fact-finding techniques to develop and implement the current existing system. Collecting required facts are very important to apply tools in System Development Life Cycle because tools cannot be used efficiently and effectively without proper extracting from facts.” [www2]

For gather client’s requirements choose possible techniques according to the system environment. Selecting different techniques can be gather different type requirements and information about different processes.

- Interviews

Interview is the most common and important used technique to collect information. Because face to face interview has more probability to get correct, verify and prove information. In this method can interview owner of the organization, managerial level, technicians, cashiers and other employees. It will help to understand clearly what their expectations from proposed system and identify the current system business processes.

- Observation

Observation is help to validate gathered requirements. Identify how workers involve in to the system and how they perform their day to day activities in the system. In the observation can see original scenario really. Because minimize effected other personal ideas and system analysis get information directly.

- Sampling existing documents

Questionnaires are prepared by system analyst for specific area to gather requirements and other related information. Analyst can gather requirements, throw taking answers for that questionnaire by giving to possible persons of the organization. Provide questioner about receive higher purchase payments and then provide suggested possible answers to select to the answer provider.

- Sampling existing documents

Studying existing system documents by the analyst can be understand the client's requirements correctly. As an example analyst can define what details need to be included to the invoice using existing invoice. Problem of this method is some clients are lack of willingness to provide their some documents.

2.3. Current System

Wisdom computer technologies their current works doing by paper based manual system. All information about day to day activities and transactions are update manually. As an example create invoice using preprinted structured sheet.

Proposed web based computer shop management system provide all features and facilities to do day to day activities. Following figure 2.1 use case diagram represent the existing manual system.

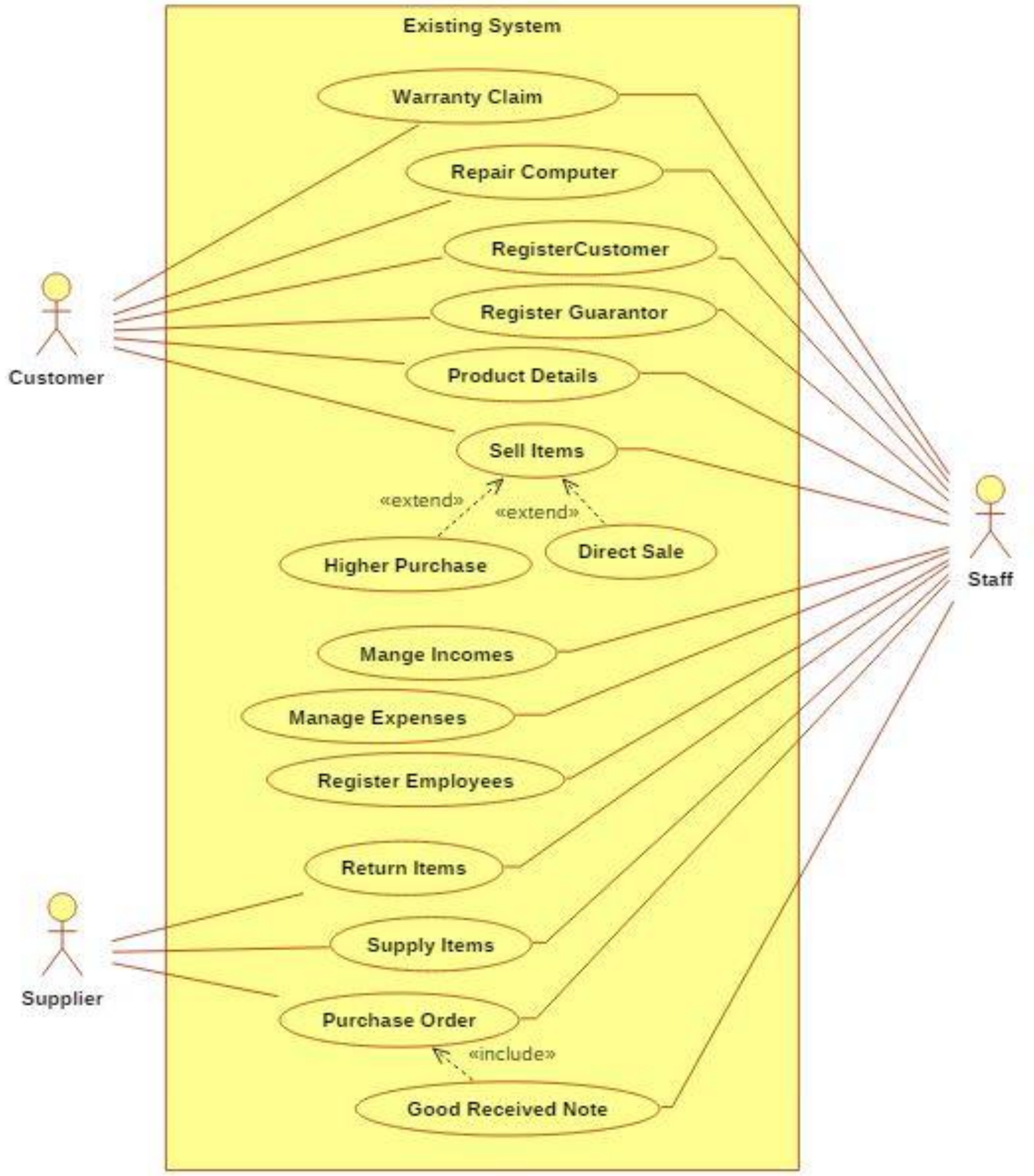


Figure 2.1 Use case diagram for existing manual system

2.3.1. Register Customer

Current system customer name, address, telephone, NIC and signature are add to invoice. They register customers for both purposes mainly as sell items and repairing computers. Customer details are not maintain separately. Carbon copies for invoices is kept in billing book. Customer details are included into invoices.

2.3.2. Register Supplier

Suppliers are registered doing by using name, address, NIC, telephone and company. Purchase orders are prepare for registered suppliers. Like customer's, supplier's details are not maintain separately.

2.3.3. Register Employee

Employees of this shop are registered based on their working levels. Such as Management, Technician, cashier and operational levels. Employee details name, address, start date, birth date, NIC, post and gender are used for to the register. After register, appointment letter is issued for registered employees of this organization.

2.3.4. Register Guarantor

Guarantors are registered done for hire purchases. In this registration use guarantor name, address, NIC and telephone mainly.

2.3.5. Provide Product Details

Most of times customers ask about product details in deeply. Employees happen provide all about details products. According to customer required things provide details about products was done by employees.

2.3.6. Sell Items

Selling items is doing in two ways. A one is direct sale and another is hire purchase. One way is direct sale and another one is hire purchase. In the direct sales, customer is pay full payment of the total. Then customer receive invoice.

In the hire purchase, customer complete down payment firstly. Remain amount is pay in defined time period as installments. For each payment, cash receipt is issued manually. After completing all installments customer receive invoice. For hire purchases they add rates additionally under their organization condition.

2.3.7. Calculate Sales

Keep records manually about sales items. Under the date keep how many items issues and what are the values of that day sales invoice to invoice.

2.3.8. Profit/Lost Calculation

This profit and lost calculations done monthly. In addition kept records about daily sales and other expenses.

2.3.9. Expenses/Incomes Calculation

Normally expenses and incomes are calculated monthly. For this calculations use daily records about all incomes expenses.

2.3.10. Repair PC Order

Repair computers is accept from repairing section of this organization. Technicians are check computer problem and after take customer telephone number for refer that computer. After completing repairing of the computer an invoice with repair description is issue to customer.

2.3.11. Communicate with Clients

Currently this organization communicate with their clients through telephone calls.

2.4. Functional Requirements

2.4.1. Master Module

- Under the master module include employees, customers, suppliers, guarantors, categories and items in stock data.
- Users can perform search data under name, telephone, code, NIC. As well as insert, update and delete can done.
- Provided features to generating different type reports for all about master data by this system.

2.4.2. Inventory Module

- Based on transactions system will automatically update stock items quantities.
- In start of the system by setting opening balance of items can be add items into the stock.
- Using goods received note user can added items into the stock on running system.
- Stock items quantities can be up to date by creating physical adjustments and special removals.
- Provide features to define reorder levels for stock items in the stock.
- The system keeps records about item movements in the stock automatically.
- Can generate reports as whole items report, current stock balance report, reorder level items report and individual items report using possible ways.

2.4.3. Purchasing Module

- By selecting items under the supplier, users can prepare purchase orders.
- System user can cancel or apply change created purchase order. Under conditions user can perform purchase order cancel in this system.
- Reports about purchase order and purchase order list can be generate.

2.4.4. Goods Received Note Module

- Create goods received note is done by selecting exist purchase order.
- Can prepare one or more goods received notes for one purchase order. That is based on received quantity of items.
- As generating reports, all goods received notes under the purchase order and individual goods received note and goods received note list can be generate.

2.4.5. Repair PC Module

- Can save repair computer and description by selecting customer in the system.
- Repair computer order can be cancel under conditions applied.
- Provide features by system to issue repair accept notice and repair invoice for clients.

2.4.6. Sales Module - Direct Sale

- Create a new sale process is done by choosing exist customer or add new customer in to the system properly.
- If the customer paid full payment of invoice, system will detect it as direct sale.
- System will generate invoice to customer automatically.

2.4.7. Sales Module - Hire Purchase

- Create a new sale process is done by choosing exist customer or add new customer in to the system properly.
- If the customer pay a part of the full payment system will continue it as hire purchase.

- In hire purchase must provide details in to the system about guarantor, down payment, number of installments and rates of the higher purchase.

- The system provide feature for receive hire purchase installments to complete all installments.

- If the customer completed all installments of the hire purchase, that hire purchase account will close automatically by system.

- Hire purchase agreement and down payment cash receipt are generated by the system at opening the hire purchase account.

- After all installments payments complete, system will generate invoice of the hire purchase.

2.4.8. User Management Module

- In the registration process of employees, system will generate username and password automatically and sending is done those details to employees email address.

- Using received user name and password in to email address, user can logon to the system then can reset their username and password under conditions.

2.4.9. Notification Module

- System show notifications about notification about expire checks, receivable higher purchase installments, reorder level items in the stock and receivable items in purchase orders automatically.

2.4.10. Warranty Claim Module

- Based on details about item, invoice and customer user can check warranty claimed items.

- If item is not expire system will accept warranty claim.

- System will issue warranty accepted notice receipt after accept warranty. Then at the end of the warranty claim process warranty completed notice receipt is issue for customer.

2.4.11. Profit and Lost Module

- System done auto calculate income and expenses using purchasing and sales.

- System provide feature to add other expenses and incomes in to the system.
- Using both of these methods system will calculate final income and expenses.
- Reports about profit, lost, incomes and expenses are generated by system as daily, weekly and monthly.

2.4.12. History Module

- System provide history about all activities of this system.
- Kept save details about activity as user, date time, activity and reference of activity by system.
- Provide feature to generate history report in system based on user defined time range.
- System provide history about stock items price changes.
- Item price history report can be generate based on item wise.

2.4.13. Report Module

- Generate report about stock current availability.
- Black listed guarantors, suppliers and customers report.
- Payments complete and incomplete hire purchases report.
- Repair complete and repair accept order report.
- Direct sales and hire purchase sales report.
- Report part of all above modules are include in the report module.

2.5. Non Functional Requirements

Non-functional requirements are consider when developing system. System usability is incrementing nonfunctional requirements consideration is very important. Non-functional requirements consideration is very important to increment the system usability. For keep high accuracy of the system, the database design part need to be done carefully. Because SQL queries written based on database structure. Database design is complete using relational database design methodology with normalization forms.

2.5.1. Security

Application security is very important factor. It’s considering in two ways. One is security in application side and another one is security in data base side. Protect application side security provide user logins and data encryption mechanisms. In application side payments and check managements are very important to protect security. More weight is kept for payment handling and check management processes to protect security of the data. Using database management system kept high security in the database server side by creating user logins and assigning roles and privileges for users.

2.5.2. Usability

The system user interface designs is done keep simple minimizing complexity. Always user interface design is based on user interface design principles. As another way is very complex processes are break down in to simple parts. Providing different ways to do same thing can increment easy to use of the system. Usage of graphical icons, colors, messages, wizards and menus more important for quick and easily understand, protect correctively and maximize the system efficiency. These methodologies usability and understandability are incrementing of the system.

2.5.3. Accuracy

Keep high accuracy of data is an important factor. Minimize saving of inconsistent data in database is done by using rules and data validations. For enter correct data provide suggestions and helps to system users in possible situations. Increment system accuracy by using possible data types, field sizes and minimizing null values adding and data repetitions in the database.

2.6. Methodology

Whole system development process is based on structure of selected methodology. Agile Software Development, Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum and Spiral are various types of methodologies. The Rational Unified Process (RUP) was selected as this system methodology. Reasons for that selection also mentioned in following.

2.6.1. Agile Software Development

"Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.

Agile Alliance was formed shortly after this gathering to encourage practitioners to further explore and share ideas and experiences. Agile Alliance continues to curate resources to help you adopt agile practices and improve your ability to develop software with agility." [www3]

2.6.2. Rapid Application Development

"RAD or Rapid Application Development process is an adoption of the waterfall model it targets at developing software in a short span of time. RAD follow the iterative RAD model has following phases as Business Modeling, Data Modeling, Process Modeling, Application Generation, Testing and Turnover.

It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD model are that it focuses on the reuse of templates, tools, processes and code." [www4]

2.6.3. Rational Unified Process

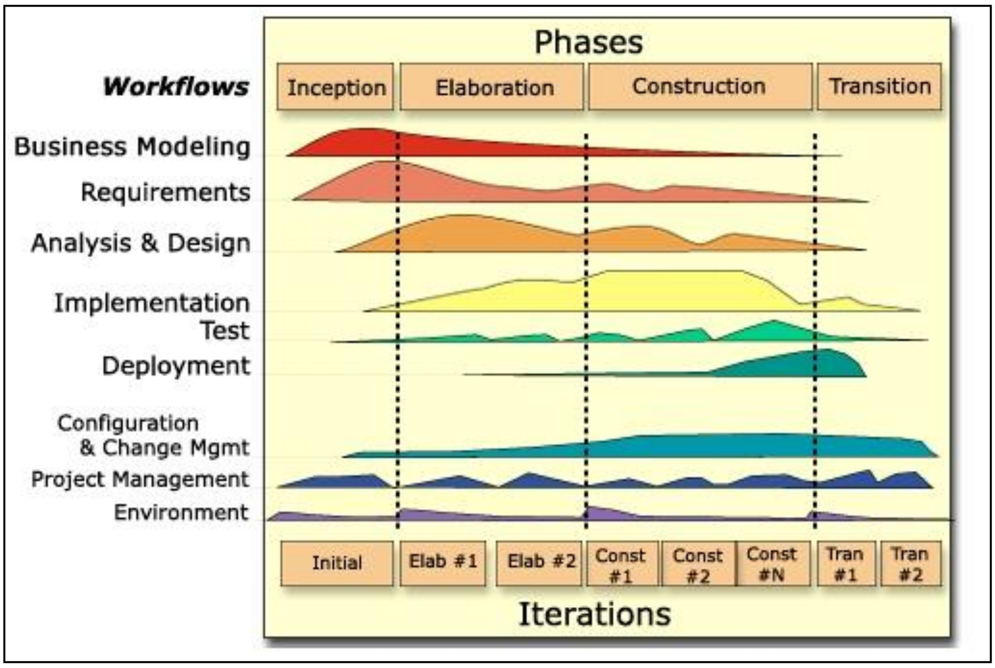


Figure 2.2 Rational Unified Process Architecture

The Rational Unified Process Methodology was selected for this computer shop management system. System design and documenting processes was done by using Unified Modeling Language (UML). It is more supported for Object Oriented Methodology and component reusability. Rational Unified Process is integrated with many software development tools. One development cycle is divided into four phases. Each phase has specific purpose and well defined milestones in RUP process.

- Inception Phase - In this phase prepare a vision document. An initial use-case model is drawing. As well as initial risk assessment business model is done. A project plan, showing phases and iterations are design.
- Elaboration Phase - A use-case model at least 80% are complete in this phase. Gathered supplementary requirements. Software architectural design and prototype is done this phase. Development plan for overall project is done.
- Construction Phase - The software product integrated on the adequate platforms. Provide user manuals and a description of the current release in this step of RUP.
- Transition Phase - Validate new system by using beta testing. Parallel operation is done with legacy system that it is replacing. Conversion of operational databases and training users and maintain were done in this step.

2.7. Existing Similar Systems

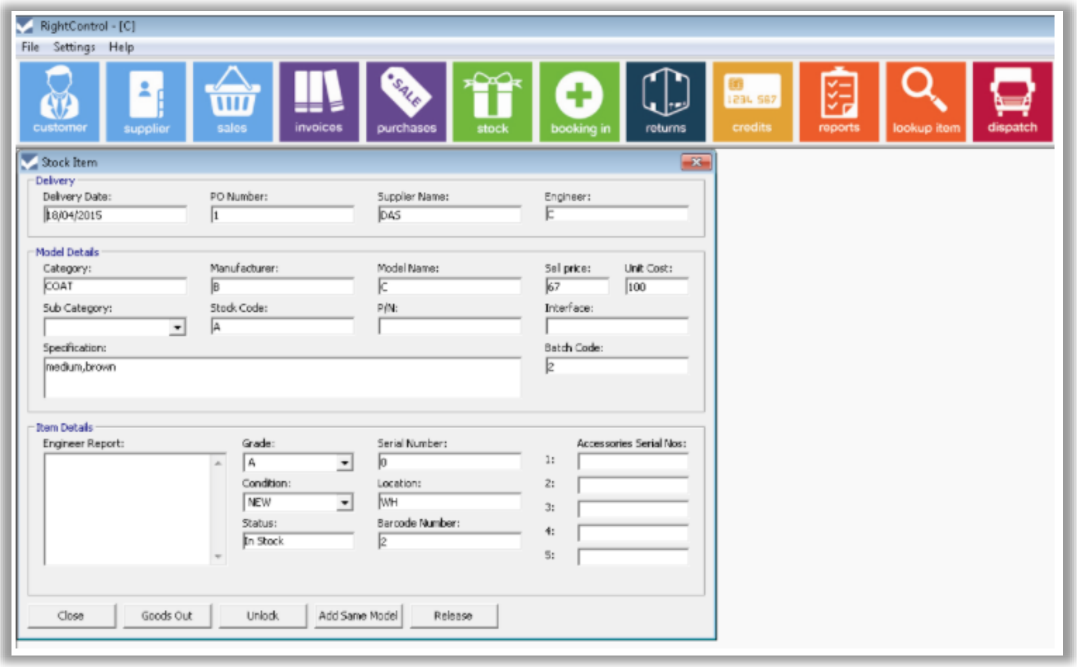
Existing similar systems was studied well before start computer shop management system. It was helpful for improve and verify knowledge about computer shop management system.

2.7.1. Right Control Lite

Right Control Lite is an inventroy management system. It provide more functionalities for control products in stock. [www6]

Following features were helped for develop proposed system functionalities.

- Reminder reorder level items
- Product categorize organize
- Return items handling
- Monitor stock level
- Privileges assigns
- Purchasing handling



Stock Item

Delivery

Delivery Date: 18/04/2015
PO Number: 1
Supplier Name: DAS
Engineer: E

Model Details

Category: COAT
Manufacturer: B
Model Name: C
Sel price: 67
Unit Cost: 100
Sub Category:
Stock Code: A
P/N:
Interface:
Specification: medium,brown
Batch Code: 2

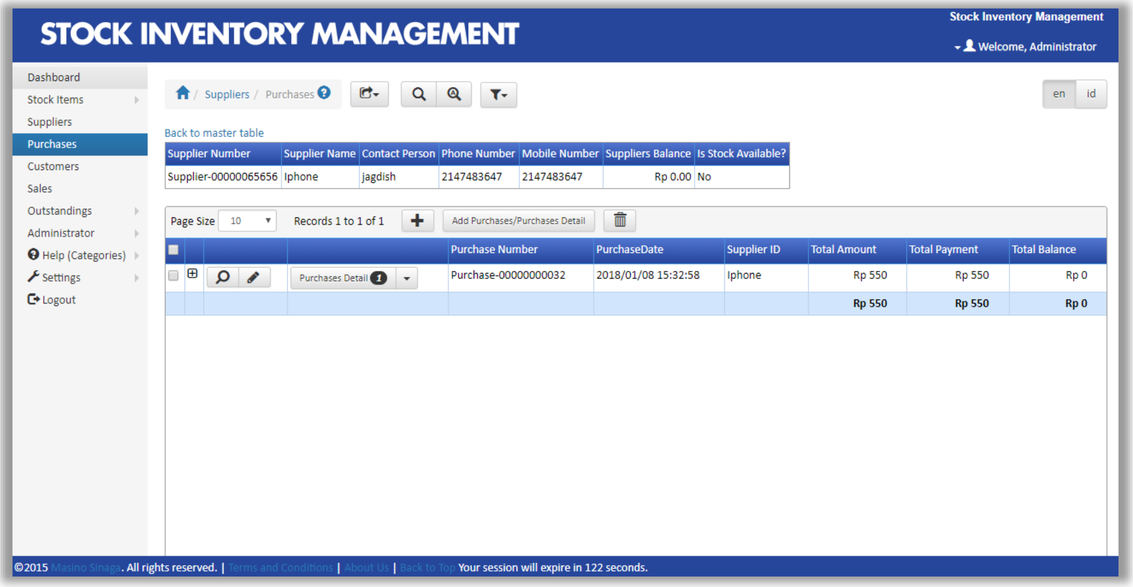
Item Details

Engineer Report:
Grade: A
Condition: NEW
Status: In Stock
Serial Number: 0
Location: WH
Barcode Number: 2
Accessories Serial Nos:
1:
2:
3:
4:
5:

Figure 2.3 An interface of Right Control Light

2.7.2. Stock Inventory Management

This is a web based application. This system mainly can done manage stock, sales and purchasing mainly. As well as available features for stock management, dash board, purchasing, sales, suppliers and customers manage in this system. [www7]



Supplier Number	Supplier Name	Contact Person	Phone Number	Mobile Number	Suppliers Balance	Is Stock Available?
Supplier-00000065656	Iphone	jagdish	2147483647	2147483647	Rp 0.00	No

Page Size 10 Records 1 to 1 of 1

Purchase Number	PurchaseDate	Supplier ID	Total Amount	Total Payment	Total Balance
Purchase-0000000032	2018/01/08 15:32:58	Iphone	Rp 550	Rp 550	Rp 0
			Rp 550	**Rp 550**	**Rp 0**

©2015 Masino Sinaga. All rights reserved. | Terms and Conditions | About Us | Back to Top Your session will expire in 122 seconds.

Figure 2.4 Interface of Stock Inventory Management System

Following options were useful to develop computer shop management system form this.

- Dash board design
- Master data handing
- Direct sales and hire purchases
- Responsive design
- Payment handling
- Data exports

CHAPTER 3 - DESIGN

3.1. Introduction

Software design phase is a very essential in software development. Because this chapter consists all about design factors including user interfaces, classes, architectures and modules. Chosen the best solution from alternative solutions and justified it.

Various approaches are available for system design. Such as Traditional Waterfall SDLC, Agile Methodology, Object Oriented Analysis Design, Joint Application Development and Rapid Application Development. Object oriented is widely used methodology because it has characteristics about reusability, understandability and minimize complexity of the code.

3.2. Alternative Solutions

The best solution was selected form alternative solutions before start the development. This is pointing in below.

- Web based system

The proposed system development is done as web based system. It develop as client server system.

- Development tools

System development and implementation is done using Microsoft Visual Studio, Microsoft SQL Server and Crystal Reports. ISS Server is use as local host during development time.

- System software

As system software choose Microsoft Windows operating system. This is based on client's requirements.

3.3. Selected Solution Justification

After consideration of clients requirements chosen the best solution as following.

- Web based system can be access throw the internet using a web browser easily. It is independent from the operating system and can use with any devices as computer, tablet and mobile.
- Used above discussed tools for system development to keep high compatibility with system running environment and sub parts of system. As an example database management system and application compatibility.
- Development tools are more supported for Microsoft Windows operating system. Employees of this organizations are well known to working the windows environment.

After consideration of all aspects in client's requirements, web based system development using ASP.Net MVC was chosen for full fill the client's requirements.

3.4. The Design Architecture

- **Structured System Analysis And Design Method (SSADM)**

Structured system analysis and design method is a water fall method for design information systems. This is can be thought to represent a pinnacle of the rigorous document approach.

Stages of SSADM are including,

- Investigation of the environment
- Business system options
- Definition of requirements
- Technical system options
- Logical design
- Physical design

Some most important factors of this SSADM is shown as follows.

- Before the start the project done feasibility study.
- Project is dividing into sub modules with well-defined objectives.
- Diagrammatic representation and modeling with documentation.
- Simply understandable for both parties of clients and developers.

Computer shop management system design process was structured based on SSADM. Unified modeling language diagrams are drawing for design architecture of the new system. The architecture is illustrated by using use case, class, activity and sequence diagrams. It is easy the system implementation with designed system to fulfill the clients requirements.

- **System Development Life Cycle (SDLC)**

Software development life cycle is define steps software development. It is included requirement analysis, design, implementation, testing, deployment and maintenance. Software development process is keeps on standard way using software development life cycle. Available different methodologies interact with software development life cycle as waterfall model, incremental model, rapid application development model and agile model.

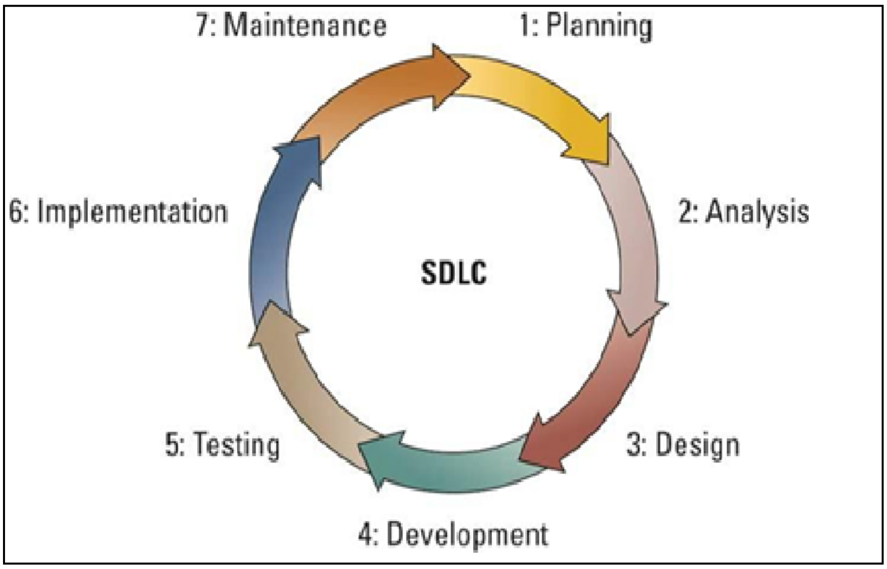


Figure 3.1 System development lifecycle

3.5. Object Oriented Designing

"Object Oriented Design is the concept that forces programmers to plan out their code in order to have a better flowing program. The origins of object oriented design is debated, but the first languages that supported it included Simulate and Smalltalk.

Object Oriented Design is defined as a programming language that has five conceptual tools to aid the programmer. These programs are often more readable than non-object oriented programs, and debugging becomes easier with locality." [www8]

3.5.1. High Level Use Case Diagram

Following Figure shows the high level use case diagram for proposed computer shop management system. This diagram represent main interactions between new system and users. Detailed information of this diagram included in Appendix B.

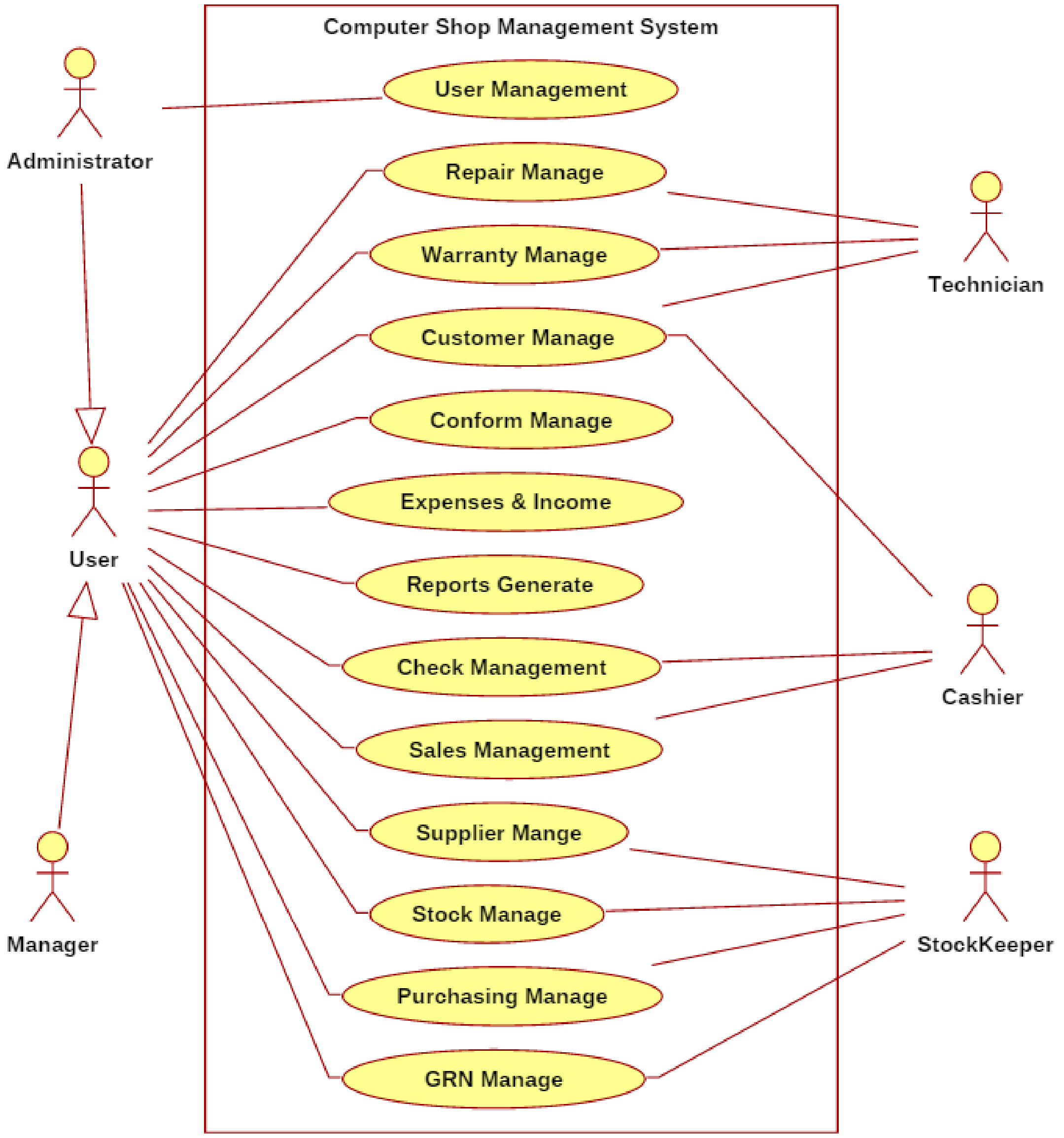


Figure 3.2 High level use case diagram for proposed system

- **Customer Details Management**

The customer details management use case diagram is displayed using figure 3.2. For prepare new sale must register customer in this system. Use case description is show bellow Table 3.1, Table 3.2, Table 3.3 and Table 3.4 respectively.

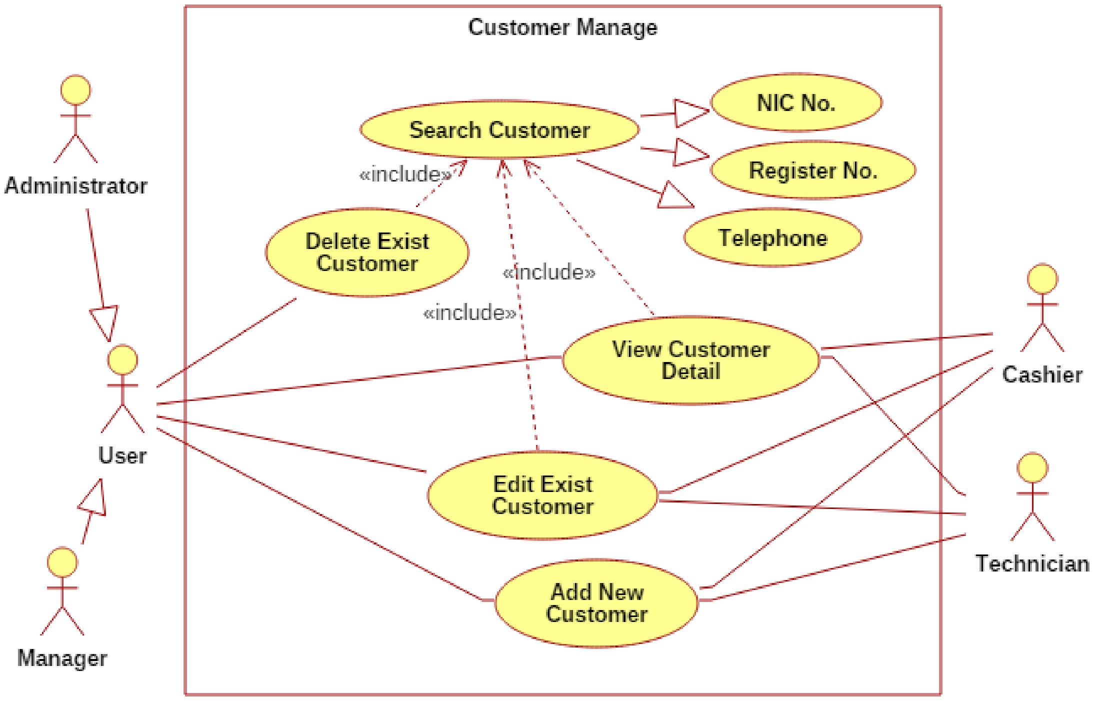


Figure 3.3 Customer Details Management

Use Case	Add new customer
Actor	Administrator, Manager, Technician, Cashier
Description	Register new customer in organization
Pre-Conditions	
User should be logged in to the system. The customer not exist in the organization.	
Flow of Events	
1. Customer provide his details. 2. User fill customer details on customer registration window. 3. User save customer details and validate data by system.	
Post-Conditions	
New customer record store in the system.	

Table 3.1 Add new customer

Use Case	View customer detail
Actor	Administrator, Manager, Technician, Cashier
Description	See details about already registered customer in the organization
Pre-Conditions	
Customer already registered in the organization.	
Flow of Events	
1. User go to the search window in the system.	

2. Enter Register No., NIC No. or Telephone by the user. 3. Validate data by system and user click on view details button.
Post-Conditions
Display customer record data in the system.

Table 3.2 View details of existing customer

Use Case	Edit exist customer
Actor	Administrator, Manager, Technician, Cashier
Description	Change details about already registered customer in the organization
Pre-Conditions	
Customer already registered in the organization.	
Flow of Events	
1. User go to the search window in the system. 2. Enter Register No., NIC No. or Telephone by the user. 3. Validate data by system and user click on edit details button. 4. Enter new details and click on save button for change the customer details by user. 5. Validate input information and save changes by system.	
Post-Conditions	
Update record with changed data about customer in the system.	

Table 3.3 Edit existing customer

Use Case	Delete exist customer
Actor	Administrator, Manager
Description	Delete already registered customer in the organization
Pre-Conditions	
Customer already registered in the organization.	
Flow of Events	
1. User go to the search window in the system. 2. Enter Register No., NIC No. or Telephone by the user. 3. Validate data by system and user click on edit details button. 4. Click on save delete for change the customer details by user. 5. Validate input information and delete customer by system.	
Post-Conditions	
Update status as false of the customer record in the system.	

Table 3.4 Delete existing customer

3.5.2. Class Diagram

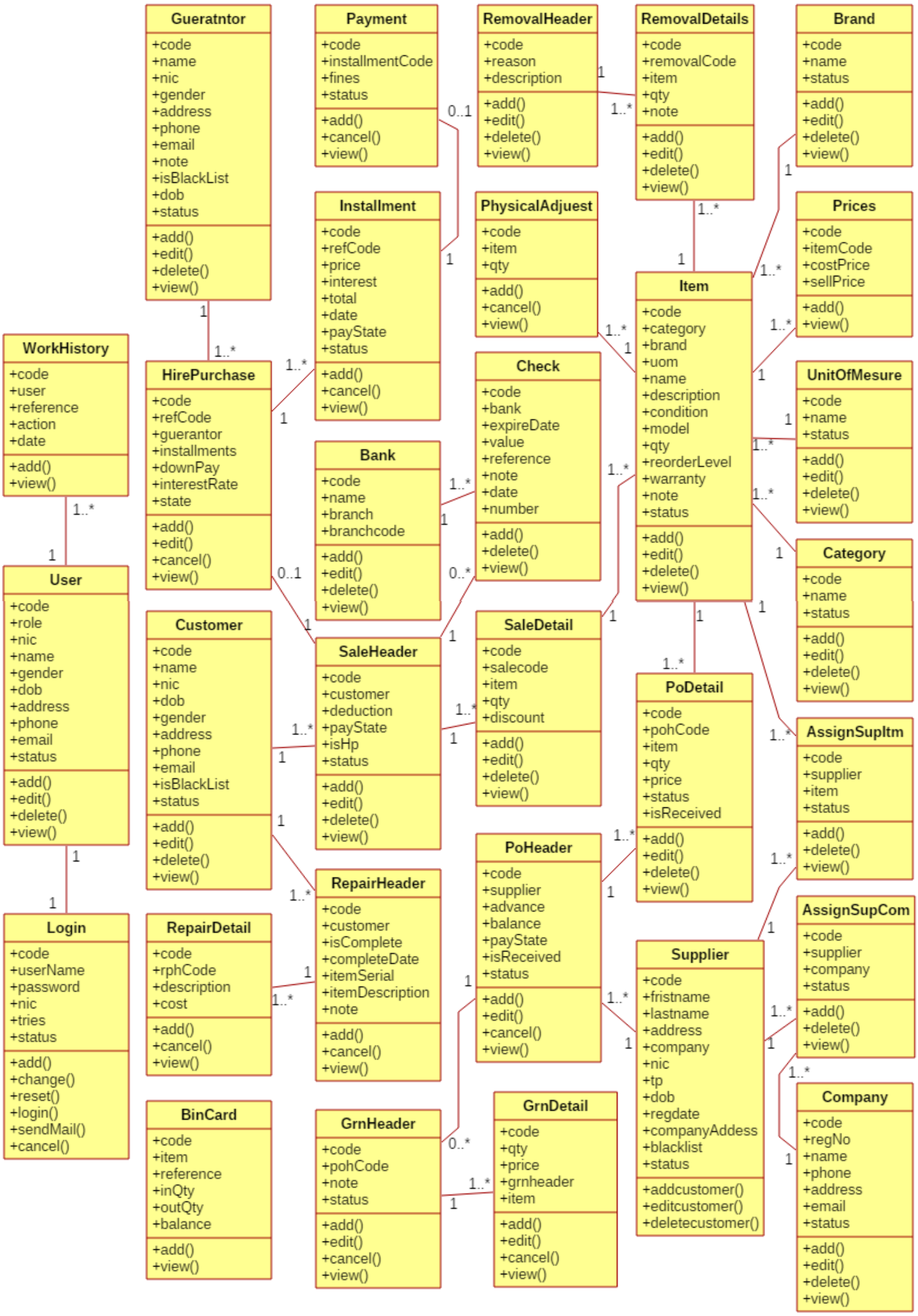


Figure 3.4 Class Diagram

3.5.3. Activity Diagram

Activity diagram describe the dynamic aspects of the system. It is basically flowchart to represent the flow from one activity to another activity. Can be describe an operation of the system by using activity. More activity diagrams are included in Appendix B. Register new customer in the organization is represent in activity diagram Figure 3.5.

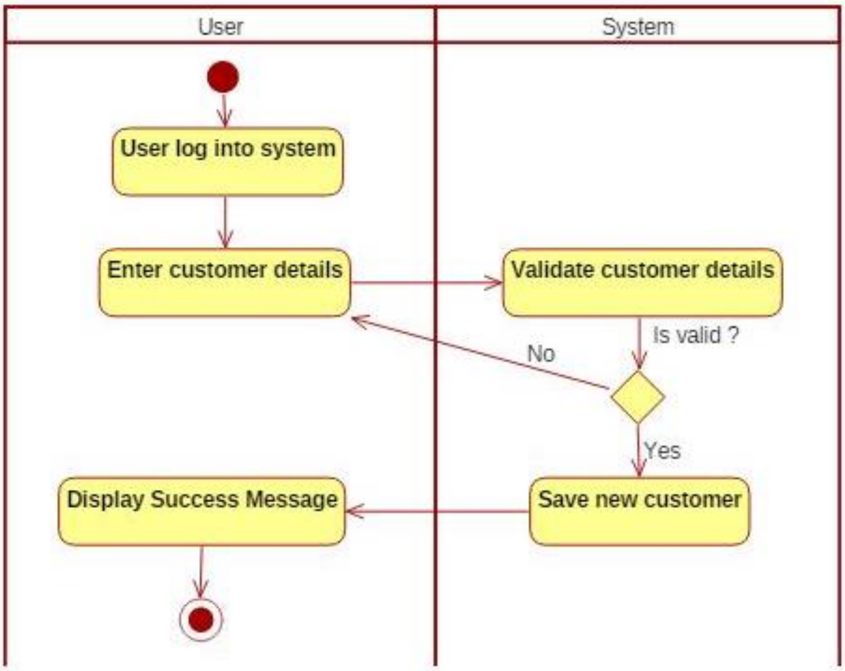


Figure 3.5 Activity Diagram for Register New Customer

3.5.4. Sequence Diagram

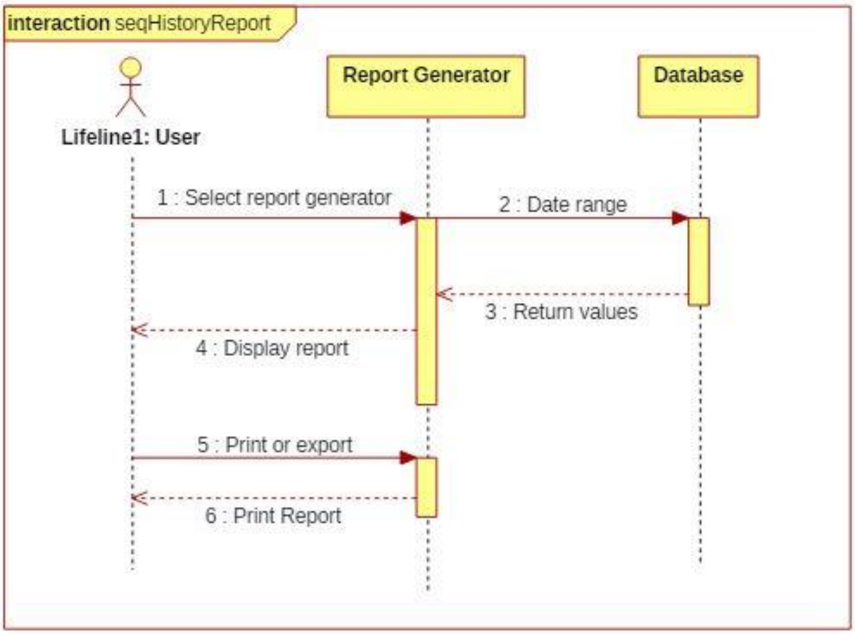


Figure 3.6 Generate History Report

Sequence diagram is an interaction diagram. This is shown object operate with one another in order. It is construct of message sequence chart. A sequence diagram shows object and their interactions arranged time in sequence. More sequence diagrams are included in Appendix B.

3.6. Database Design

The database design process included conceptual design, logical design and physical design phases. Using entity relationship diagram (ER) shown entities in the system and relationships between entities with cardinality of them. For computer shop management system data base design process done base on relational data base management system (RDBMS). In Figure 3.6 show data base diagram with entity relationships.

3.6.1. Database Normalization

Database normalization is the process of decomposing unsatisfactory relations into similar relations. For minimize and remove insert, update and delete animalize in relational data base designing process use data base normalization. Minimize data redundancy and null values saving problems in the database using database normalization process.

- Zero Normal Form (0NF)

Multi valued attributes are exists.

- First Normal Form (1NF)

Any multivalued attributes have been removed.

- Second Normal Form (2NF)

Any partial functional dependencies have been removed.

- Third Normal Form (3NF)

Any transitive dependencies have been removed.

3.6.2. Entity Relation Ship Diagram

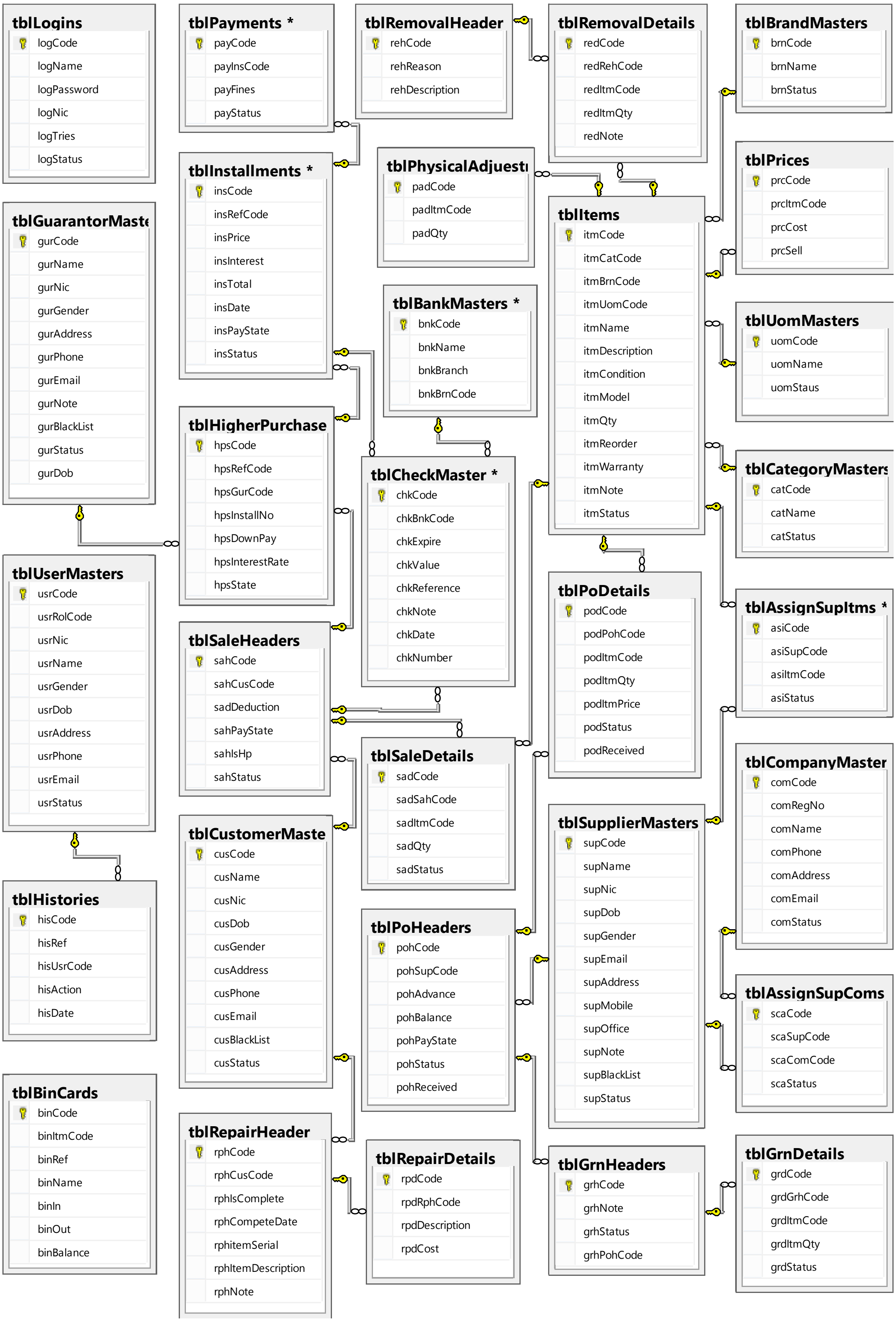


Figure 3.7 Entity Relationship Diagram

3.7. Interface Design

Interface design process is very important phase of the project. Because user interact with the system using the interface. If interfaces are not good user are not likes to works with it. Poor user interface design is risk to failure system. Good inter face design is reason to success the project. In user interface design use possible eye friendly colors, icons and keep design is simple. Because interface is complex users hard to learn.

In the user interface show to user where user in now in the system. Provide different ways to do same thing to user.

UI design guidelines.

- User friendly

Design user interface very user friendly. User possible components, order, colors, text and headers with possible sizes for keep high user friendly.

- Easy to learn

Minimize complexity of the user interface. User can identify easily what can do, what data enter to the system using the interface clearly.

- One click done all

For complete a task user not need going to more interfaces. Most of user interfaces provide user to perform their task using one interface.

- Wizard UI

If tasks are more complex it divided few interfaces. System provide wizard one by one interfaces user to complete their task simply.

- Validate UI

Save only correct data in to the system done validations. Show error, conformation and perform action status messages to user for higher data creativity and user understandability.

- Responsive UI

Keep responsive user interface design for works in different size screens of this system.

3.7.1. System Login

The system user must login with providing correct username and password. If user name and password are mismatch the system will detect its invalid. User can try maximum five times only. After system lock login automatically. If user forget his login information system provide link to reset users username and password with system registered email address. The system login page represent in Figure 3.8.

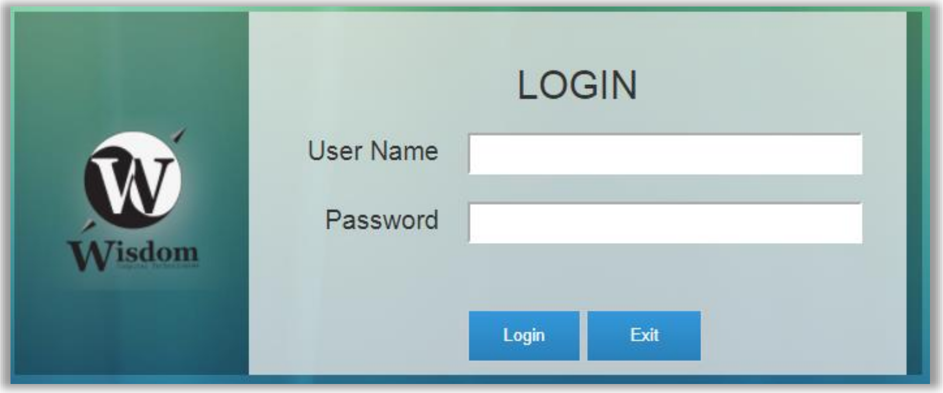
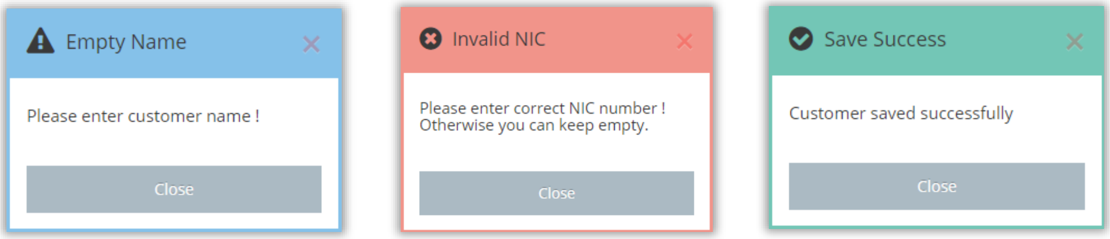


Figure 3.8 Login interface of the system

3.7.2. Messages

Message boxes design of the system was done based on principles of user interface design. Use possible colors, text and images for easily understand and identify for system users. Figure 3.9 shown message boxes of the system.



Please enter customer name !

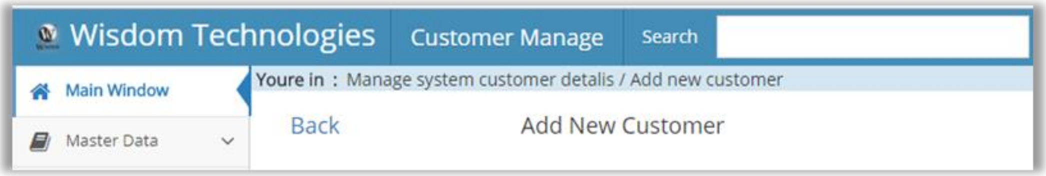
Please enter correct NIC number !
Otherwise you can keep empty.

Customer saved successfully

Figure 3.9 Message boxes of the system

3.7.3. Address bar

The window address will helped to user for identify what is the path from main dashboard to current position of the system. Using this address bar easily jump between interfaces in the address bar. Following Figure 3.10 represent address bar.

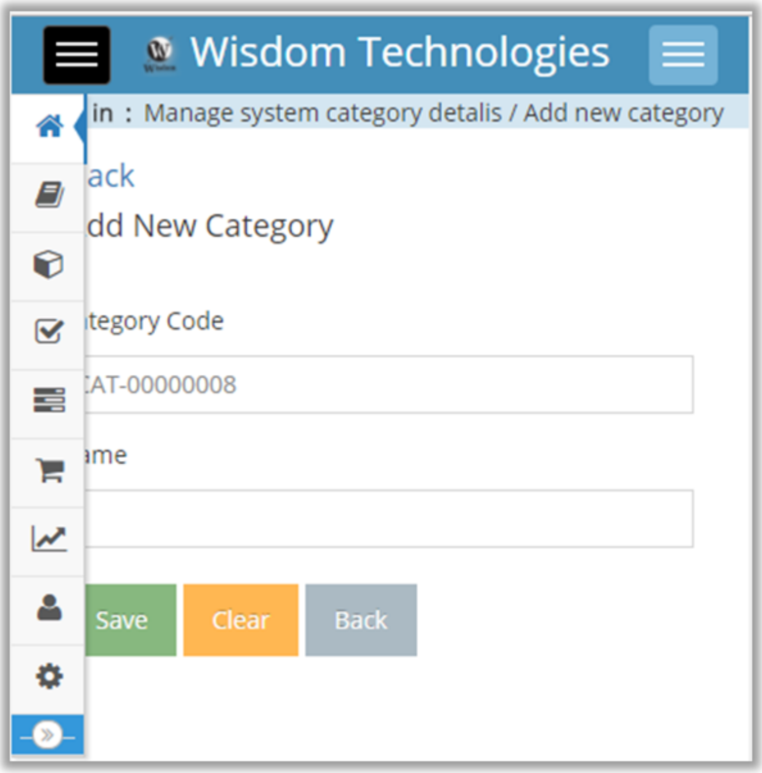


Youre in : Manage system customer detalis / Add new customer

Figure 3.10 Address bar of the system

3.7.4. Responsive design

System design was compatible to run any of devices as computer, tablet and mobile. Main menu and other components of the system are support run with different sizes displays in many devices. Figure 3.11 represent responsive menu of the system.



Wisdom Technologies

in : Manage system category detalis / Add new category

ack

dd New Category

tegory Code

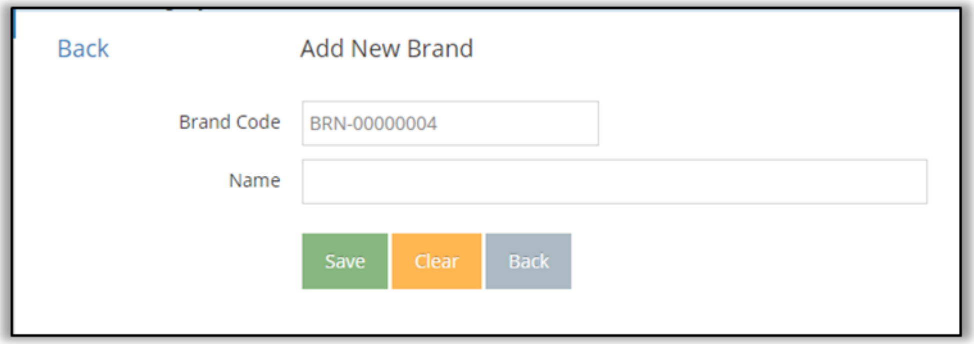
AT-00000008

ıme

Figure 3.11 Responsive menu of the system

3.7.5. Accessible

Available more improved accessible features in the system. To do one thing provide different ways for users. Figure 3.12 represent two different type of back ways for go to back in the system.



Back

Add New Brand

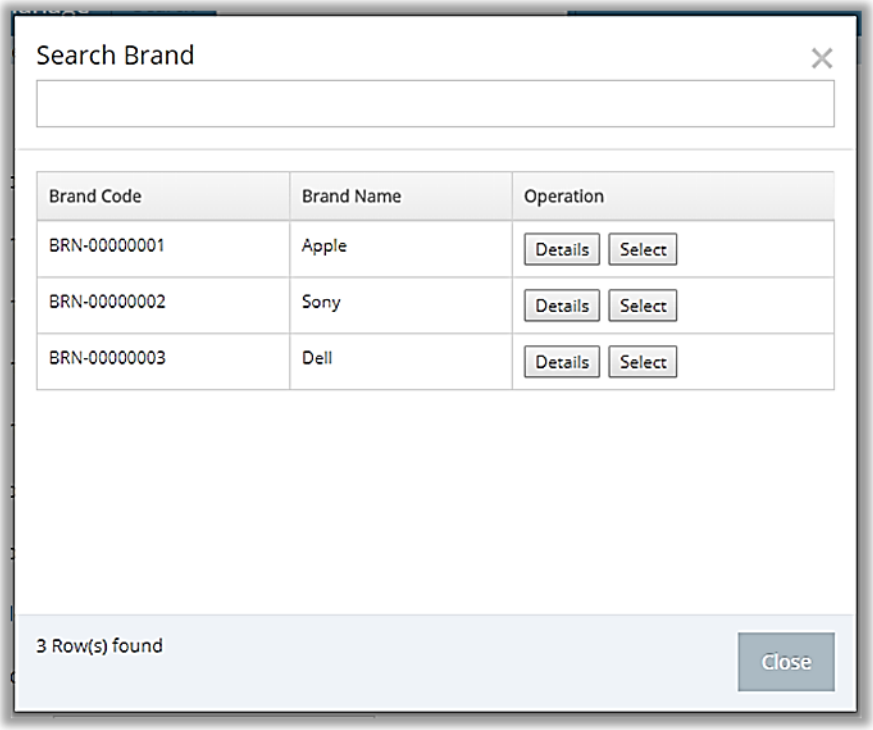
Brand Code: BRN-00000004

Name:

Figure 3.12 Accessible of the system

3.7.6. Quick Search

The quick search box provide features to users search data based on different attributes. Easily get more details clicking details button on searched things. Search box is reusable component all-around of this system. Figure 3.13 shown search dialog.



Search Brand

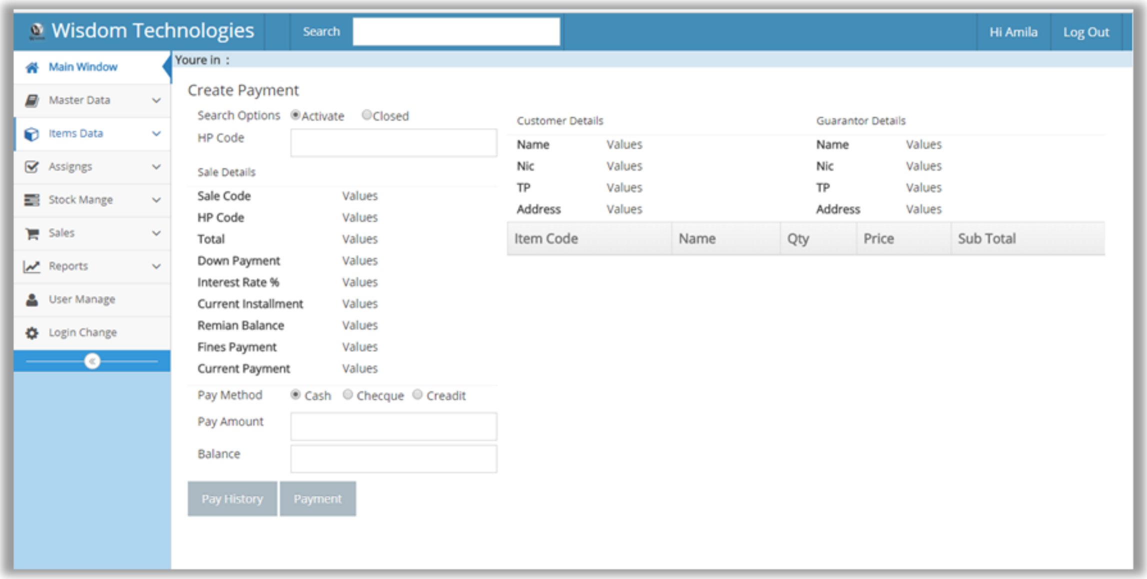
Brand Code	Brand Name	Operation
BRN-00000001	Apple	Details Select
BRN-00000002	Sony	Details Select
BRN-00000003	Dell	Details Select

3 Row(s) found

Figure 3.13 Quick search of the system

3.7.7. Create payment window

The interface of create payment facilitated for users receive hire purchase payments in the system. Following Figure 3.14 represent create payment window.



Youre in :

Create Payment

Search Options ● Activate ○ Closed

HP Code

Sale Details

Sale Code	Values
HP Code	Values
Total	Values
Down Payment	Values
Interest Rate %	Values
Current Installment	Values
Remian Balance	Values
Fines Payment	Values
Current Payment	Values

Pay Method ● Cash ○ Checque ○ Creadit

Pay Amount

Balance

Customer Details

Name	Values
Nic	Values
TP	Values
Address	Values

Guarantor Details

Name	Values
Nic	Values
TP	Values
Address	Values

Item Code	Name	Qty	Price	Sub Total

Figure 3.14 Create payment window of the system

CHAPTER 4 - IMPLEMENTATION

4.1. Introduction

The main objective is this phase convert planning project to working system and implementing it in clients organization for practically use. This phase require minimum time in software development life cycle. The code project convert to web application in this phase.

The computer shop management system implement on hosted web server to access client computers using web browsers throw internet.

Code writing process included using comments and regions. It’s very important thing coding best practices. Comments very useful to developer for understand coding without time consuming. Regions use to collapse the large number of line code area in the code window.

4.2. Interaction Between Modules

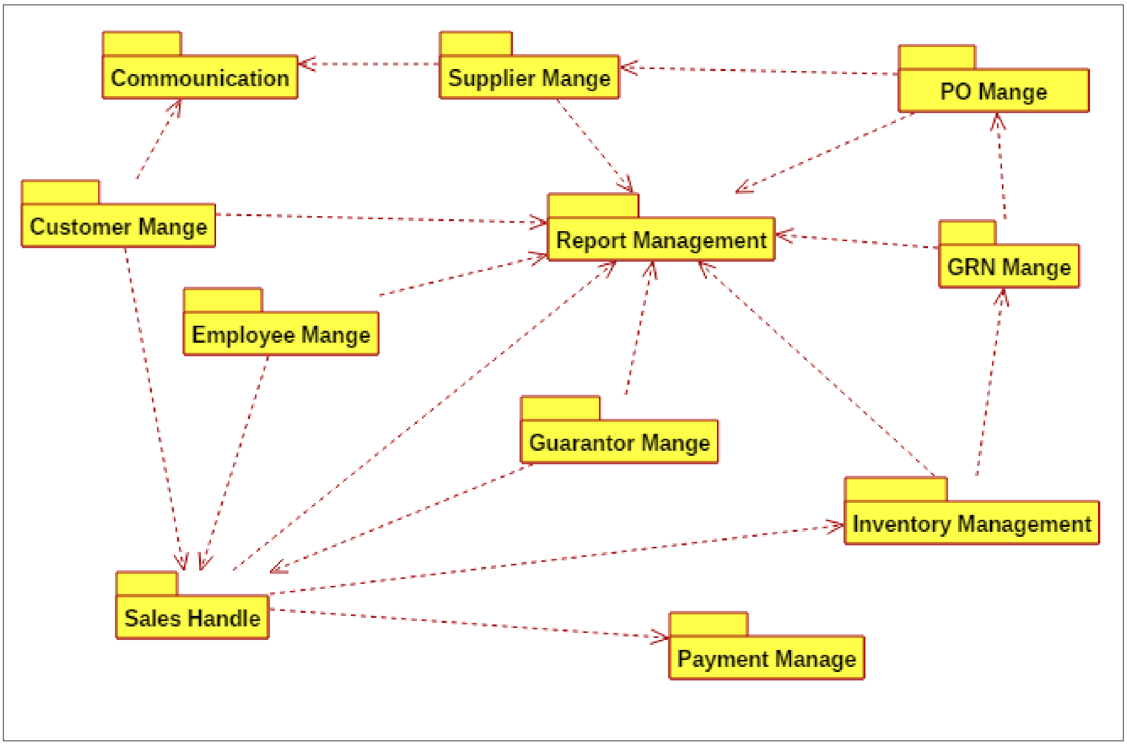


Figure 4.1 Interaction between modules of the system

4.3. System Requirements

The system requirements identify three different ways as hardware, software and live ware.

4.3.1. Hardware Requirements

- Computer with HDD 500GB, RAM 6GB, Processor Core i3 3rd Gen
- Router or Internet connection
- Printer

4.3.2. Software Requirements

- OS Microsoft Windows 8.1 64bit later one
- Web Browser
- MSSQL Server 2014 Express
- Microsoft Office 2013 or later one

4.3.3. Livewire Requirements

- System users for different levels

This computer shop management system is compatible with bellow explained minimum requirements and resources.

4.4. Development Tools

Development tools can be classified as follows.

4.4.1. Software Tools

- **Visual Studio**

Microsoft Visual Studio 2015 use as the application development IDE. Visual studio IDE code window provide suggestions for all languages use this project. Developer can see application data flow and code performances and code analysis using visual studio IDE. Compatibility with Microsoft SQL severer is an advantage.

- **MSSQL Server 2014**

MSSQL server use as the DBMS as this application. This server provide more services to run systems database. Some services as user management, grant privileges, query window and ER Design window.

- **Crystal Reports XI**

Crystal report XI Business Object used to design reports and verify reports of this application according to client requirements. For this include runtimes 32 bit and 64 bit.

- **Google Chrome**

Google chrome web browser used to run the wisdom shop management web application.

4.4.2. Programming Languages

Different type of programing languages are used for design user interface, keep responsive design, database programming and application programming.

- **Visual C-Sharp**

C-Sharp is object oriented and component oriented programming language. In MVC architecture the controller methods are written by using C-Sharp language.

- **Structured Query Language**

SQL use rarely. Because SQL Replaced with LINQ queries in ASP.Net MVC architecture. Used LINQ queries to interact with database.

- **Java Script**

Java Script use for client side validations. Using JS validate GUI that running on web browser. Defined functions using JavaScript.

- **JQuery**

JQuery use for event handling and other validations in user interface of this web application.

- **Ajax**

Ajax used for pass data from UI to controller method. UI variables declared using java script. Used get, post method and JSON arrays to retrieve data.

- **Hyper Text Markup Language**

UI design mainly done by using html. All components define using HTML language, between html tags.

- **Cascade Style Sheets**

CSS used for design user interface that are not effected from bootstrap. Use inline CSS, external cuss methods.

4.4.3. Frameworks and Architecture

- **Entity Framework - Code Fist**

Entity framework use for create asp application. Used code first migration data base method to create application database in MSSQL server.

- **Bootstrap**

Bootstrap used for keep responsive UI design in this system. Used bootstrap library and apply bootstrap for classes for html elements in web UI.

- **MVC Architecture**

The computer shop management system is based on Model View Controller architecture. Following figure shown MVC architecture from [www9].

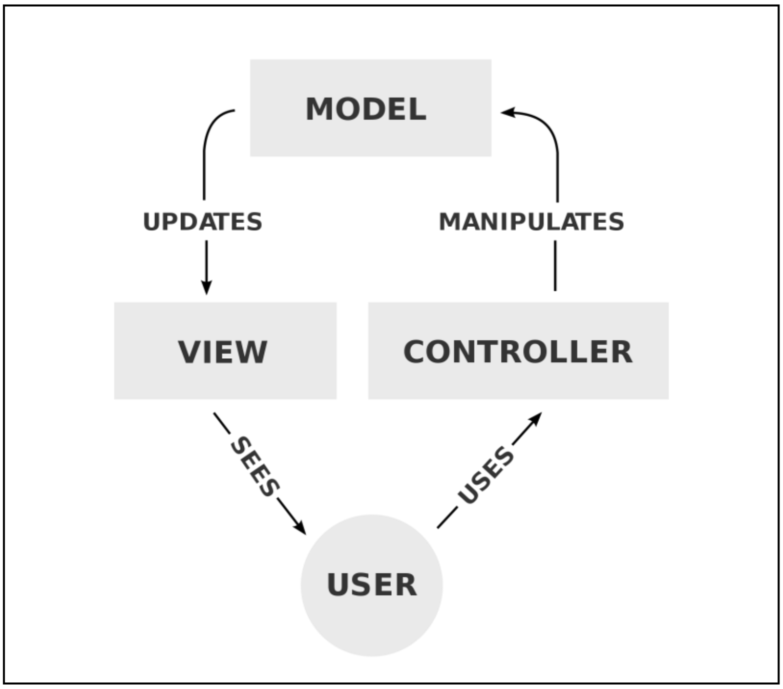


Figure 4.2 MVC Architecture

4.5. Code Segments

4.5.1. CURD Operations

Create (insert), Update, Read (select) and Delete operations are represent under module wise of the computer shop management system. Major code snippets were shown with description. Comment lines of the code are provide description for the code.

- **User Management Module**

Validate login

Login handling included in user management module. Following code segment check the user enter or not enter login details on system login. If user name or password empty system provide error message. Else not empty login information called “LoginCheck()” function.

```
$('#btnLogin').on('click', function () {
    var un = $('#txtUn').val();
    var pw = $('#txtPw').val();
    if (un=="") {
        $('#error').text('Enter Username');
    } else if(pw=="") {
        $('#error').text('Enter Paaword');
    } else {
        LoginCheck();
    }
});
```

Login check function send login information form view to controller using post method for compare with database saved login information. If the comparison is match user can login to main dashboard.

```
function LoginCheck() {
    var un = $('#txtUn').val();
    var pw = $('#txtPw').val();
    $.ajax({
        url:'@Url.Action("LoginCheck","SystemLogin")',
        type: 'post',
        data: {
            'pun': un,
            'ppw':pw
        },
        success: function (res) {
            $('#error').text(res);

            if (res == "Please Wait...") {
                var url = $("#RedirectTo").val();
                location.href = url;
            }
        },
        complete: function () {
        },
        error: function () {
            alert('error login check request')
        }
    });
}
```

Controller method of “LoginCheck” check user enter login information correct or incorrect in the database. If correct assign logged user name for the login session. Else provide error message about login information mismatch. The code segment represent on next page.

```
public string LoginCheck(string pun, string ppw)
{
    string error = "";
    var res = db.Login.Where(x => x.logName == pun).FirstOrDefault();
    if (res != null)
    {
        if (res.logPassword == ppw)
        {
            error = "Please Wait...";
            Session["log"] = res.logName;
        }
        else
        {
            error = "Username password dismatch !";
        }
    }
    else
    {
        error = "Username password dismatch !";
    }
    return error;
}
```

User open logged system without any working, after user defined time duration system will logout automatically and come back to login screen. This process done with login session expiring time.

```
//define the sesson expire time
Session.Timeout = 525600;

@*link to login page*@
@Html.Hidden("RedirectTo", Url.Action("Index", "SystemLogin"))

//redirect to login page
function GotoLogin(){
    if (@Session["log"] == null) {
        var url = $("#RedirectTo").val();
        location.href = url;
    }
}
```

- **Maser Module**

All about registrations and assignations in the system is included in to master module.

Register New Supplier

Following code of function Insert() is done passed validated data from user interface to controller method using post method for insert data into database.

```
//send data for insert
function Insert() {
    var Code = $('#txtCode').val();
    var Name = $('#txtName').val();
    var Nic = $('#txtNic').val();
    var Dob = $('#datDob').val();
    var Gender = $('input:radio[name=rbtnGender]:checked').val();
    var Email = $('#emaEmail').val();
    var Address = $('#txtAddress').val();
    var Mobile = $('#telMobile').val();
    var Office = $('#telOffice').val();
    var Note = $('#txtNote').val();
    $.ajax({
        url: '@Url.Action("Insert", "SupplierMasters")',
        data: {
            'pSupCode': Code,
            'pSupName': Name,
            'pSupNic': Nic,
            'pSupDob': Dob,
            'pSupGender': Gender,
            'pSupEmail': Email,
            'pSupAddress': Address,
            'pSupMobile': Mobile,
            'pSupOffice': Office,
            'pSupNote': Note
        },
        type: "post",
        success: function (res) {
            alert(res);
        },
        error: function () {
            alert('Error insert supplier request !');
        }
    });
}
```

Following code snippets shown supplier master controller method of Insert is done insert new supplier data into database. If the insert is success or failed return message for show to user.

```
//save data into database
public string Insert(
     string pSupCode,
     string pSupName,
     string pSupNic,
     DateTime pSupDob,
     bool pSupGender,
     string pSupEmail,
     string pSupAddress,
     string pSupMobile,
     string pSupOffice,
     string pSupNote
     )
{
    tblSupplierMaster objSup = new tblSupplierMaster
    {
        supCode = pSupCode,
        supName = pSupName,
        supNic = pSupName,
        supDob = pSupDob,
        supGender = pSupGender,
        supEmail = pSupEmail,
        supAddress = pSupAddress,
        supMobile = pSupMobile,
        supOffice=pSupOffice,
        supNote = pSupNote,
        supBlackList = false,
        supStatus = true
    };
    db.Supplier.Add(objSup);
    db.SaveChanges();
    //save history data
    new HistoriesController().Insert(
       pSupCode, Global.logedUserCode, "Create Supplier"
    );
    return "Saved";
}
```

Assign Items for Suppliers

The system stock items supply by suppliers. A supplier supply one or more items. An item has one or more suppliers. For complete this scenario system provide this feature including the master module. Follows code snippet represent items assign for suppliers without preventing pre assigns.

```
//andd new assing data
public string Assigns(string supCode, string itmCode)
{
    string ret = "";
    tblAssignSupItm obj = new tblAssignSupItm();
    obj.asiCode = GnerateCode();
    obj.asiSupCode = supCode;
    obj.asiItmCode = itmCode;
    obj.asiStatus = true;
    //if not exist
    var isExist = db.AssignSupCom.Where(o => o.scaSupCode == obj.asiSupCode)
                    .Where(x => x.scaComCode == obj.asiItmCode)
                    .Where(y => y.scaStatus == obj.asiStatus);


    if (isExist.FirstOrDefault() == null)
    {
        new HistoriesController().Insert(
              obj.asiCode, Global.logedUserCode, "Assign item to supplier"
        );
        db.AssignSupItm.Add(obj);
        if (db.SaveChanges() > 0)
        {
            ret = "Saved";
        }
        else
        {
            ret = "Save Failed";
        }
    }
    else
    {
        ret = "Existed";
    }
    return ret;
}
```

- **History Module**

Add History

About all activities of the computer shop management system provide history. Following code section represent data insert in to database for save history about an activity.

```
//insert
[ValidateAntiForgeryToken]
public void Insert(string reference, string user, string action)
{
    tblHistory tblHistory = new tblHistory();
    tblHistory.hisCode = GnerateCode();
    tblHistory.hisDate = DateTime.Now;
    tblHistory.hisRef = reference;
    tblHistory.hisUsrCode = user;
    tblHistory.hisAction = action;

    db.History.Add(tblHistory);
    db.SaveChanges();
}
```

Following code is done added history record about create a new sale using bellow history inserting method.

```
//add to history
new HistoriesController().Insert(objSah.sahCode, Global.logedUserCode, sale);
```

- **Inventory Module**

Add to Bin card

Inventory module included set opening balance, physical adjustments, special removals, up to date items quantity and bin card. The bin card provide item movements in stock as in or out. Following method use for save record to bin card.

```
//insert bin card
[ValidateAntiForgeryToken]
public void Insert(
       string itemCode, string refCode, string name, int inQty, int outQty
)
{
    var itm = db.Item.Where(i => i.itmCode == itemCode)
                     .Where(i => i.itmStatus == true).First();
    tblBinCard objBin = new tblBinCard
    {
        binCode = GnerateCode(),
        binItmCode = itemCode,
        binName = name,
        binRef = refCode,
        binIn = inQty,
        binOut = outQty,
        binBalance = itm.itmQty
    };
    db.BinCard.Add(objBin);
    db.SaveChanges();
    //add to history
    new HistoriesController().Insert(
       objBin.binCode,Global.logedUserCode, "Bin Card"
    );

}
```

- **Goods Received Note Module**

Create Goods Received Note

Use bellow method in GRN creation and save details about stock in quantities GRN items represent following code snippet.

```
//save grn
public string Insert(string pGrhCode, string pGrhPohCode, string pGrhNote)
{
    //grh save------------------------------------
    tblGrnHeader objGrh = new tblGrnHeader
    {
        grhCode = pGrhCode,
        grhPohCode = pGrhPohCode,
        grhNote = pGrhNote,
        grhStatus = true
    };
    db.GrnHeader.Add(objGrh);
    db.SaveChanges();
    //grd save------------------------------------
    //grd-collect
List<tblPoDetail> lstPod = new PoDetailsController().GetPodItems(pGrhPohCode);
List<tblGrnDetail> lstGrd = new List<tblGrnDetail>();
    foreach (var item in lstPod)
    {
        //mix pod and grd
        var grdItem = lstGrdRecQty.Where(g => g.grdItmCode == item.podItmCode)
                              .Where(g => g.grdStatus == true).FirstOrDefault();
            tblGrnDetail objGrd = new tblGrnDetail
                {
                    grdGrhCode=objGrh.grhCode,
                    grdItmCode = item.podItmCode,
                    grdItmQty = (grdItem!=null?grdItem.grdItmQty:0),
                    grdStatus = true
                };
                if (objGrd.grdItmQty>0)
                {
                    lstGrd.Add(objGrd);
                }
            }
            //grd save----------------------------------------
            foreach (var item in lstGrd)
            {
                item.grdCode = GnerateCode();
                db.GrnDetail.Add(item);
                db.SaveChanges();
            }
            //grh recevide status update-----------------------
     //get po items
```

```
var resPod = (from poh in db.PoHeader
              join pod in db.PoDetail on poh.pohCode equals pod.podPohCode
                     where poh.pohCode==pGrhPohCode
                     select new
                     {
                         poh.pohCode,
                         pod.podItmCode,
                         pod.podItmQty
                     });
//get grn items
var resGrd = (from grh in db.GrnHeader
              join grd in db.GrnDetail on grh.grhCode equals grd.grdGrhCode
                     where grh.grhPohCode == pGrhPohCode
                     group grd by grd.grdItmCode into grdItm
                     select new
                     {
                         grdItm.FirstOrDefault().grdItmCode,
                         grdItmqQtySum = grdItm.Sum(x => x.grdItmQty)
                     });
var chkResGrd = resGrd;
//compare and update grd items
foreach (var itemResGrd in resGrd)
{
   if (chkResGrd.FirstOrDefault()!=null)
   {
      int podQty = resPod
                  .Where(x=>x.podItmCode==itemResGrd.grdItmCode)
                  .FirstOrDefault().podItmQty;
      int grdty = itemResGrd.grdItmqQtySum;
      if (podQty==grdty)
      {
          string pohCode = resPod.FirstOrDefault().pohCode;
          var tblPod = db.PoDetail.Where(x => x.podPohCode == pohCode)
                       .Where(x => x.podItmCode == itemResGrd.grdItmCode)
                       .ToList().FirstOrDefault();
          tblPod.podReceived = true;
          db.Entry(tblPod).State = EntityState.Modified;
       }
    }
 }
 db.SaveChanges();
 //grd recevide status update-----------------------
 //check all grn items receive status true
 //refer resGrd,resPod
 //update grnh
 if (resGrd.FirstOrDefault()!=null)
 {
     bool isAllReceived = true;
     foreach (var itemResGrd in resGrd)
     {
         int podQty = resPod
                      .Where(x => x.podItmCode == itemResGrd.grdItmCode)
                      .FirstOrDefault().podItmQty;
         int grdty = itemResGrd.grdItmqQtySum;
         if (podQty>grdty)
         {
            isAllReceived = false;
         }
      }
      if (isAllReceived)
      {
           var tblPoh = db.PoHeader
```

```
                             .Where(x => x.pohCode == resPod.FirstOrDefault()
                             .pohCode).FirstOrDefault();
                tblPoh.pohReceived = true;
                db.Entry(tblPoh).State = EntityState.Modified;
            }
        }
        db.SaveChanges();
        //stock update-------------------------------------
        foreach (var item in lstGrd)
        {
            var itm = db.Item.Where(i => i.itmCode == item.grdItmCode)
                      .Where(i => i.itmStatus == true).First();
            itm.itmQty += item.grdItmQty;
            db.Entry(itm).State = EntityState.Modified;
        }
        db.SaveChanges();
        //bin save------------------------------------------
        foreach (var item in lstGrd)
        {
            new BinCardsController()
            .Insert(item.grdItmCode, item.grdGrhCode, "GRN", item.grdItmQty, 0);
            new HistoriesController()
            .Insert(item.grdCode, Global.logedUserCode, "Add GRD Item");
        }
        //his save------------------------------------------
        new HistoriesController()
        .Insert(pGrhCode, Global.logedUserCode, "Create GRN");
        return "Save success";
    }
```

4.5.2. Data Validation

The data validation is most import for prevent saving incorrect or impossible data into the database. It will helped to minimize system errors and keep input data correct format. Following code snippets shown user input fields validations.

```
//input text only
function TextValid(txtId, length) {
    var x = $(txtId).val().replace(/[^A-Za-z ]/, '');
    $(txtId).val(x);
    $(txtId).val(x.substring(0, length));
}

//input nubers only
function NumValid(id, length) {
    var x = $(txtId).val().replace(/[^0-9]/, '');
    $(txtId).val(x);
    $(txtId).val(x.substring(0, length));
}

//validate nic number
function NicValid(id, length) {
    var x = $(txtId).val().replace(/[^0-9]/, '');
    $(txtId).val(x.substring(0, length));
}
```

Bellow validation methods used in customer saving view validation function represent in following code snippet.

```
<script src="~/Library/Validates/Validate.js"></script>
//validations customer data
$('#txtNic').on('keyup', function () {
    NicValid('#txtnic', 13);
});
$('#txtName').on('keyup', function () {
    NicValid('#txtName', 35);
});
$('#txtPhone').on('keyup', function () {
    NicValid('#txtPhone ', 10);
});

//validate customer data
function IsValid() {
    if ($('#txtName').val()=='') {
        MessageShow('Empty Name', 'Please enter customer name !')
        return false;
    } else if ($('#txtPhone').val() == '') {
        MessageShow('Empty TP', 'Please enter customer telephone !')
        return false;
    } else if ($('#txtAddress').val() == '') {
        MessageShow('Empty Address', 'Please enter customer address !')
        return false;
    } else {
        return true;
    }
}
```

Following code snippets all input data are valid insert the customer into database on save button click.

```
//save customer data
$('#btnSave').on('click', function () {
    if (IsValid()) {
        Insert();
    }
});
```

4.5.3. SQL Injection

Structured Query Language usage is minimized in ASP.Net EFK. It’s replaced with LINQ Queries. Following code represent SQL query used for purchase order code generate automatically.

```
//get table row count--------------
private int GetRowCount()
{
    var blogs = db.PoHeader.SqlQuery("SELECT * FROM tblPoHeaders").ToList();
    return blogs.Count;
}
//generate id----------------------
public string GnerateCode()
```

```
{
    int max = 0;
    if (GetRowCount() > 0)
    {
        foreach (tblPoHeader item in db.PoHeader.ToList())
        {
            if (max < Convert.ToInt32(item.pohCode.Substring(4, 8)))
            {
                max = Convert.ToInt32(item.pohCode.Substring(4, 8));
            }
        }
        return "POH-" + (max + 1).ToString("00000000");
    }
    else
    {
        return "POH-00000001";
    }
}
```

- **LINQ Queries**

LINQ SQL Query inner join is represent following code snippets for select item full description for show item details to user.

```
public JsonResult LoadItem(string pItmCode)
{
    //collect basic details
    var resItm = (from itm in db.Item
                  join cat in db.Category on itm.itmCatCode equals cat.catCode
                  join brn in db.Brand on itm.itmBrnCode equals brn.brnCode
                  join his in db.History on itm.itmCode equals his.hisRef
                  join usr in db.Login on his.hisUsrCode equals usr.logCode
                  join uom in db.UnitOfMeasure on itm.itmUomCode equals
                  uom.uomCode
                  where his.hisAction == "Create Item"
                  where itm.itmStatus == true
                  where itm.itmCode == pItmCode
                  select new
                  {
                      itmCode = itm.itmCode,
                      itmCategory = cat.catName,
                      itmCatCode = cat.catCode,
                      itmBrand = brn.brnName,
                      itmBrnCode = brn.brnCode,
                      itmUom = uom.uomName,
                      itmUomCode = uom.uomCode,
                      itmName = itm.itmName,
                      itmDescription = itm.itmDescription,
                      itmModel = itm.itmModel,
                      itmReorder = itm.itmReorder,
                      itmWarranty = itm.itmWarranty,
                      itmNote = itm.itmNote,
                      itmCondition = itm.itmCondition,
                      itmDate = his.hisDate,
                      itmUsr = usr.logNic
                  }).ToList().First();
    //collect latest price
    var resPri = (from pri in db.Price
                  join his in db.History on pri.prcCode equals his.hisRef
                  orderby his.hisDate descending
                  join usr in db.Login on his.hisUsrCode equals usr.logCode
                  where pri.prcItmCode == pItmCode
                  select new
                  {
                      prcCost = pri.prcCost,
                      prcSell = pri.prcSell,
                      prcUsr = usr.logNic,
                      prcDate = his.hisDate
                  }).ToList().First();
    //assign item collected details
    ItemUi itemUiObj = new ItemUi
    {
        items = new tblItem
        {
            itmCode = resItm.itmCode,
            itmName = resItm.itmName,
            itmDescription = resItm.itmDescription,
            itmModel = resItm.itmModel,
            itmReorder = resItm.itmReorder,
            itmWarranty = resItm.itmWarranty,
            itmNote = resItm.itmNote,
            itmCondition = resItm.itmCondition
        },
        categories = new tblCategoryMaster
        {
            catName = resItm.itmCategory,
            catCode = resItm.itmCatCode
        },
        brands = new tblBrandMaster
        {
            brnName = resItm.itmBrand,
            brnCode = resItm.itmBrnCode
        },
        uoms = new tblUomMaster
        {
            uomName = resItm.itmUom,
            uomCode = resItm.itmUomCode
        },
        prices = new tblPrice
        {
            prcCost = resPri.prcCost,
            prcSell = resPri.prcSell
        },
        itemUsers = new tblUserMaster
        {
            usrNic = resItm.itmUsr
        },
        priceUsers = new tblUserMaster
        {
            usrNic = resPri.prcUsr
        },
        itemHistories = new tblHistory
        {
            hisDate = resItm.itmDate
        },
        priceHistories = new tblHistory
        {
            hisUsrCode = resPri.prcUsr
        }
    };
    return Json(itemUiObj ,JsonRequestBehavior.AllowGet);
}
```

4.6. Re-Usable Codes & Components

4.6.1. Code Re-Use

This project search widow and message box reuse in everywhere. Follow method call with different parameters for reuse according to the purpose. Using search main method call search window different places. Call message show shown message different purposes in this application.

- **Search popup and Message box popup**

```
//search settings
function SearchMainShow(title, colCode, colName, colOpr) {
    $('#mmTitle').html(title);
    $('#mmttCode').html(colCode);
    $('#mmttName').html(colName);
    $('#mmttOpr').html(colOpr);
    $('#modMain').modal('show');
}
//message settings
function MessageShow(title, message) {
    $('#msgTitle').html(title);
    $('#msgMessage').html(message);
    $('#modMessage').modal('show');
}
//search company widow settings
SearchMainShow('Search Company', 'Company Code', 'Company Name', 'Operation');
//search supplier window settings
SearchMainShow('Search Supplier', 'Supplier Code', 'Supplier Name',
               'Operation');
```

History recording method is used in everywhere of this system. Bin card assign method is called in many of times in inventory module. Following code snippets displayed above method reused in the system. Saved data about guarantor add, edit and delete using history insert method reusing.

- **Record History**

```
new HistoriesController().Insert(
       pGurCode, Global.logedUserCode, "Create Guarantor"
);
new HistoriesController().Insert(
       pGurCode, Global.logedUserCode, "Change Guarantor"
);
new HistoriesController().Insert(
       pGurCode, Global.logedUserCode, "Delete Guarantor"
);
```

4.6.2. Component Re-Use

- **ASP.Net C#**
    - Email sender facility
    - Pdf file generator in crystal reports

    - Excel exporting in crystal reports
    - Data encryptions for logins

- **JavaScript JQuery**
    - Validation method for control validations
    - Popup interfaces
    - Date time format handle functions.
    - Event handling of tools
- **AJAX JSON**
    - Pass data from view to controller
    - Load retrieved data from data base
- **Bootstrap CSS HTML**
    - User interface format design
    - Search boxes and message boxes design.
    - Data table in user interfaces
    - Control designs in interfaces

CHAPTER 5 – EAVALUATION

5.1. Introduction

"Evaluation is a process that critically examines a program. It involves collecting and analyzing information about a program's activities, characteristics, and outcomes. Its purpose is to make judgments about a program, to improve its effectiveness, and/or to inform programming decisions.

It is important to periodically assess and adapt your activities to ensure they are as effective as they can be. Evaluation can help you identify areas for improvement and ultimately help you realize your goals more efficiently. Additionally, when you share your results about what was more and less effective, you help advance environmental education.

Evaluation enables you to demonstrate your program's success or progress. The information you collect allows you to better communicate your program's impact to others, which is critical for public relations, staff morale, and attracting and retaining support from current and potential funders." [www10]

5.2. System Testing

System testing is consist range of different time periods, different methodologies, different ways and different persons. Application user totally different one from each other. Produce system must be compatible with any users of the system perfectly. Client satisfaction very reasonable thing to minimize application failure rate of the application. Error freely and verified the client requirements, system testing is very important factor.

5.3. Software Testing Techniques

Available many of ways for testing software systems. Choosing one or more suitable methodologies can be done system testing phase completed successfully.

- **Black box testing**

A software testing method in which the internal structure, design, implementation of the

Item being tested is not known to the tester. This test can be functional or non-functional, though usually functional. Test design techniques include: Equivalence partitioning, Boundary Value Analysis, Cause Effect Graphing.

- **White box testing**

A software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. Test design techniques include: Control flow testing, Data flow testing, Branch testing, and Path testing.

- **Gray box testing**

A software testing method which is a combination of Black Box Testing method and White Box Testing method.

- **Agile testing**

A method of software testing that follows the principles of agile software development.

- **Ad hoc testing**

A method of software testing without any planning and documentation.

5.4. Types of Testing

5.4.1. Unit Testing

The unit testing is done by developer in between system developments. Objective of the unit testing is check and verify the correctness of the module. Check the individual parts of the coding.

5.4.2. Integration Testing

After completed process of module testing integrated those modules as a group. A software system consist with multiple modules. Different modules are developed by different developers. Integration test is check after integration of modules is two modules are communicate with each other or not.

5.4.3. System Testing

The overall completed system is check before issue the software system as complete product. System testing is first time end to end test before the launching.

5.4.4. Acceptance Testing

Acceptance testing is performed by clients. If the system meetup the clients requirements that was agree. User acceptance testing is beta test of the product and evaluated by the system end users. Validate end to end business flow in this testing.

5.4.5. Regression Testing

Regressing testing is done by after the system modifications. The modifications of system, component, group or related units verify and conform those modifications are works correctly without damaging other modules of the system.

5.5. Test Plan & Cases

The test plan include details about scope, schedule of testing and deliverables. Based on bellow testing types prepare test plan using modules of the system. All aspects of the system are test with following test cases. Test plan is represent about tester, place, time and quality of performed.

Test cases are create for testing process. Description, expected output and status are consisted in test cases. As base using modules major reason for minimize the complexity of the system testing procedure. (Appendix E mentioned test cases of the system)

- **Test Result for System Login (User Management Module)**

Test No.	Description	Steps to test	Expected result
1	Validate user input details	Enter correct user name Enter correct password Click on login button	Show message "Please wait…" Login successful
2	Validate user input details	Empty user name Empty password Click on login button	Show message "Enter user name"
3	Validate user input details	Enter correct user name Empty password Click on login button	Show message "Enter password"
4	Validate user input details	Enter correct password Empty user name Click on login button	Show message "Enter user name"
5	Validate user input details	Enter correct user name Enter incorrect password Click on login button	Show message "User name password mismatch"
6	Validate user input details	Enter incorrect user name Enter correct password Click on login button	Show message "User name password mismatch"

7	Validate user input details	Enter incorrect user name Enter incorrect password Click on login button	Show message “User name password mismatch”
8	Validate user input details	Try to login more than five times with entering incorrect login information	Show message “Account locked, Click here to reset login”
9	Validate user input details	Click on “Account locked, Click here to reset login” link	Show message “Login info sent to your email address”

Table 5.1 Test results for system Login

- **Test results for add customer - Customer master (Master module)**

Test No.	Description	Steps to test	Expected result
1	Validate customer adding	Enter customer all data with name, gender, address, phone Click on save button	Show message “Save successful”
2	Validate customer adding	Enter customer name, gender, address, phone only Click on save button	Show message “Save successful”
3	Validate customer adding	Load customer code generated by system	Show customer code
4	Validate customer adding	Click on save button without entering any customer data	Enter customer name
5	Validate customer adding	Enter customer data without name Click on save button	Show message “Please enter customer name”
6	Validate customer adding	Enter data with empty address Click on login button	Show message “Enter customer address”
7	Validate customer adding	Enter customer data without phone Click on save button	Show message “Enter correct phone number”
8	Validate customer adding	Enter customer data with invalid phone Click on save button	Show message “Enter correct phone”
9	Validate customer adding	Enter customer data with invalid NIC Click on save button	Show message “Enter correct NIC or keep empty”
10	Validate customer adding	Enter customer data with invalid email Click on save button	Show message “Enter correct email or keep empty”
11	Validate customer adding	Enter customer data with already added NIC Click on save button	Show message “Customer NIC already exist”
12	Validate customer adding	Enter customer data with already added phone Click on save button	Show message “Customer phone already exist”
13	Validate customer adding	Enter customer data with already added email Click on save button	Show message “Customer email already exist”
14	Validate customer adding	Try to customer name with invalid character to the name	Auto clear invalid character by system
15	Validate customer adding	Try to customer NIC with invalid character to the NIC	Auto clear invalid character by system
16	Validate customer adding	Try to customer address with invalid character to the address	Auto clear invalid character by system

17	Validate customer adding	Try to customer phone with invalid character to the phone	Auto clear invalid character by system
18	Validate customer adding	Try to customer email with invalid character to the email	Auto clear invalid character by system
19	Validate customer adding	Click on back button	Back to main page
20	Validate customer adding	Click on back link	Back to main page
21	Validate customer adding	Click on clear button	Clear customer data without customer code

Table 5.2 Test results for add customer

- **Test Results for Create Direct Sale (Sales Module - Direct)**

Test No.	Description	Steps to test	Expected result
1	Validate creation new direct sale	Select customer search by name, code or NIC Select item search by item name, code Enter item quantity Click on add button Enter deductions Select payment method Enter pay amount (greater than or equal too grand total) Click on pay button	Calculate total and balance Generate invoice for sale
2	Validate creation new direct sale	Enter all sale data without selecting customer	Select a customer
3	Validate creation new direct sale	Click on add button without selecting an item	Select an item
4	Validate creation new direct sale	Click on add button without enter quantity or zero quantity	Enter quantity
5	Validate creation new direct sale	Click on add button without selecting item and empty quantity or zero quantity	Select an item
6	Validate creation new direct sale	Add already added item into sale again	Add new quantity to old quantity for item
7	Validate creation new direct sale	Click on remove button in selected items list	Remove item form list
8	Validate creation new direct sale	Click on pay button without entering pay amount or zero pay amount or less pay amount	Enter pay amount
9	Validate creation new direct sale	Enter invalid characters in quantity	Auto clear character
10	Validate creation new direct sale	Enter invalid characters in pay amount	Auto clear character
11	Validate creation new direct sale	Click on clear button	Clear all and set as zero values without effecting sale code Set pay method as cash
12	Validate creation new direct sale	If customer is not available Click on add new button	Popup customer add window
13	Validate creation new direct sale	If check select as pay method	Popup check add window

14	Validate creation new direct sale	If credit select as pay method	Popup credit card add window
15	Validate creation new direct sale	Click on back button	Back to main page
16	Validate creation new direct sale	Click on back link	Back to main page

Table 5.3 Test results for create direct sale

- **Test Results for Assign Items for Suppliers (Master Module)**

Test No.	Description	Steps to test	Expected result
1	Validate item assign for supplier	Select supplier using search Select items using search Click on assign button	Show message “Item assigned to supplier successfully”
2	Validate item assign for supplier	Select supplier using search	Load already assigned items
3	Validate item assign for supplier	Click on assign button without select a supplier and items	Show message “Please select supplier first”
4	Validate item assign for supplier	Click on assign button without select a supplier	Show message “Please select supplier first”
5	Validate item assign for supplier	Click on assign button without select minimum one item	Show message “Please select item”
6	Validate item assign for supplier	Select supplier search by supplier name, code or NIC	Load supplier search popup
7	Validate item assign for supplier	Select an item search by item name or code	Load item search popup
8	Validate item assign for supplier	Click on clear button	Clear form data
9	Validate item assign for supplier	Click on back button	Back to main page
10	Validate item assign for supplier	Click on back link	Back to main page

Table 5.4 Test results for assign items for suppliers

5.6. User Accepting Testing

User accept testing was done by the system different levels of users. System users are the administrator, the manager, cashier, stock keeper and technician. For collect user accepting testing feedback.

Next page Table 5.1 type formatted questioner was provided. The questioner has four satisfy levels as very good, good, poor and very poor.

User Name:			**Date Time:**	
Criteria	**Very good**	**Good**	**Poor**	**Very poor**
System functionalities				
Navigations of the system				
Interface designs				
Complexity level of activities				
System efficiency				
Improvements / suggestions	....			

Table 5.5 user evaluation questioner

Using above formatted collected individual user's evaluation feedbacks summarization calculate overall system user acceptance testing summery in figure 5.2 as follows.

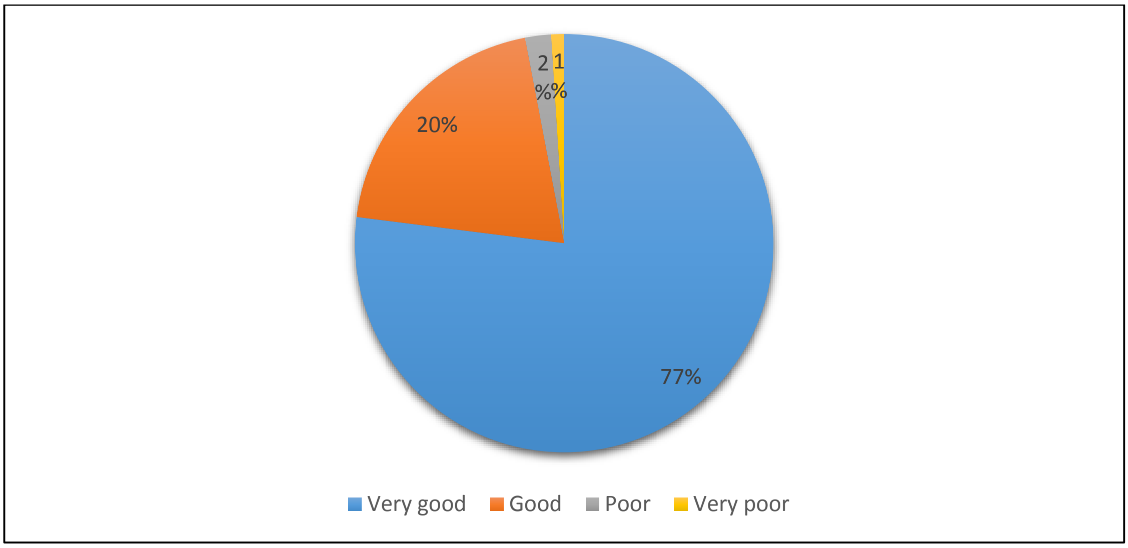


Figure 5.1 Overall System User Acceptance Testing Summery

CHAPTER 6 – CONCLUSION

6.1. Summary

The wisdom computer technology is a leading shop in Balangoda area. They are selling all type of computers, parts and related items as direct sales and hire purchase for customers as retails and stocks. Other way they are repairing and servicing computers in their technical section. Technicians, stock keeper, cashiers and other employees works under the shop manager properly.

From beginning to current, the wisdom done their day to day activities by using paper based manual way. Using manual way they faced lots of problems around when maintaining customers, suppliers and employee's data, preparing purchasing and selling areas. Based on current manual system more difficult to prepare reports and check current availabilities of the shop. It is negative effected to their business with problem to decision making, more time and cost wasting. Proposed new computer shop management system was developed to solve those issues and problems effectively in this organization.

Gather client's requirements for the proposed system, allocated more times. To identify stakeholder's requirements completely used different type of fact finding techniques considering the situation. The manager, stock keeper, cashiers, technicians and other employees are interviewed. Other time provide simple questioners as well as study their documents are in detailed.

When system analysis proposed system was done choosing RUP methodology. For gathering client's requirements clearly, different types of UML diagrams were drew including use case diagrams, class diagram, activity diagrams and sequence diagrams. System database design process was based on ER diagram.

Considering after clients requirements proposed system implemented as a web based system. Computer shop management system concern the staff of the wisdom computer technologies. All users can access the system throw the internet. For server side of the system c-sharp. MSSQL server used as data base management system. Client side of the system is developed using JS, JQuery, Ajax, HTML, CSS and Bootstraps.

System is divided to modules for minimize the system complexity. Prepare test plan for the system by module wise. System test is done in module’s test cases are testing. Finally overall system was tested after completing module testing.

Full filled the clients requirements and completed the project objectives when completion the system. Implemented web based system with hosted data based for store their data and access data using any device and anywhere. System provide features to handle all type of data and generating required reports according to the client’s requirements. Provide email facility to communicate with their clients, employees and customers. System has more secure for all activities doing on the system.

Generating different type of reports according to user levels is very useful for taking decisions easily in this organization more effectively. Using the system history the company manager or administrator can monitor users all activities in the system from login to logout. Providing different user levels and given privileges are important to clearly define the users working area of the system. The system minimize user involves of every each activity. As well as minimize typing and data entering for keep high efficiency of activities. Most of times system automated features are provide for possible activities to remove the user involvements. The system is run using well designed data base. Database is design considering efficiency large amount of data handling and effectively for the future.

The new system is replace manual system to computerized system, paper based convert to computer web based, most manual process convert to automate in this organization. System archived client’s requirements completing functional and non-functional requirements. Considering user feedbacks, overall system test results prove that system meet up the clients requirements and it completed the client satisfactory level successfully.

6.2. Problems Encountered

Start of the project requirements gathering part more difficult. For interview workers, study documents, verify scopes and clearly define system boundaries take more time. As well as allocated more weight to clearly identify requirements for minimize failure rate of the project.

Analyze and check critical functions of the project given more priority. Because success the software project based on successfully of the critical functions of the project. Some unfamiliar real world activates converting to system process and developing it by facing to problems solving them.

6.3. Lessons Learnt

This was a great opportunity convert theoretical knowledge as practical working system. Developing the new system take huge experience from gathering requirements to implement and hosting system. Its included working with clients, provide demonstrations and analyze, design, implementation, testing and project management.

Standard way proposal writing, proposal writing with preparing agreements with clients very useful for improve communication skills. Improved knowledge of how to solve the real world problem choosing the best solution form alternative solutions. Then selected solution convert to web based practical system following standard way.

6.4. Achievements

- Successfully completed proposed system within time including expected functionalities for full fill the client’s requirements.
- Huge experience works with different software development IDE’s, tools, languages, architectures with new technologies.
- Valuable opportunity to introduce my own production in software industry including unique innovative things to the system.
- Improved problem solving skills in every phases of SDLC, according to the situation using previously learnt theoretical lessons and knowledge.

6.5. Deviations from the proposal

- Requirement gathering process was completed with late few days. Because for conformation with re modifications of clients requirements get more days.

- Allocated more time for system implement part. Because in coding with testing error handling get more time.

6.6. Future Enhancement

Day to day growing of the workings system's improvements are possible.

- View available items details on stock to customer using system separately.
- Add feature to customer login to system separately.
- Add items to cart and online payments done throw web.
- Improve home delivery part of this system.
- Add barcode reader, generation works automated.

References

[www1] Wikipedia - Systems analysis [online]. Available: https://en.wikipedia.org/wiki/Systems_analysis [Accessed: January 28, 2018]

[www2] UKEssays - Definition of Fact Finding Techniques Information Technology [online]. Available: https://www.ukessays.com/essays/information-technology/definition-of-fact-finding-techniques-information-technology-essay.php [Accessed: January 28, 2018]

[www3] Agile Alliance - What is Agile Software Development [online]. Available: https://www.agilealliance.org/agile101/ [Accessed: January 28, 2018]

[www4] Meet Guru99 - Free Training Tutorials & Video for IT Courses - What is RAD Model? Advantages & Disadvantages [online]. Available: https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html [Accessed: January 28, 2018]

[www5] ReQtest - Software Testing & Bug Reporting Tool [online]. Available: https://reqtest.com/agile-blog/how-to-successfully-migrate-from-rup-to-scrum/ [Accessed: January 28, 2018]

[www6] @PDFelement - Top 4 Inventory Software [online]. Available: https://pdf.wondershare.com/business/inventory-software.html [Accessed: January 28, 2018]

[www7] SourceForge - Stock Inventory Management [online]. Available: https://sourceforge.net/projects/stock-inventory-management/ [Accessed: January 28, 2018]

[www8] Process Maturity | FAQ - What is Object Oriented Design? (OOD) [Online]. Available: http://www.selectbs.com/process-maturity/what-is-object-oriented-design [Accessed: January 28, 2018]

[www9] Danilo Maccioni - Model View Controller (traduzione tutorial di Kevin Waterson) [online]. Available: http://www.danilomaccioni.it/kevin-waterson-mvc/ [Accessed: January 28, 2018]

[www10] Evaluation: What is it and why do it? | Meera - Evaluation: What is it and why do it? [Online]. Available: http://meera.snre.umich.edu/evaluation-what-it-and-why-do-it#what [Accessed: January 28, 2018]

Main Menu Vertical part: Ace Master Dashboard used under fully modified [online] Available: https://themelot.net/ace-v1-3-3-responsive-admin-template.html [Accessed: January 28, 2018]

Main Menu Horizontal part: Bootstrap Menu Bar used under fully modified [online] http://formoid.com/articles/bootstrap-navbar-example-1382.html [Accessed: January 28, 2018]

Appendix A - System Documentation

The following steps are explain how to install web based system and what are the required minimum requirement to run application.

- **1st Step**

Hardware Requirements

Processor:	Core i3 3rd Gen 1.8GHz or above
Ram:	6GB
HDD:	500GB
Network:	LAN & Internet

Software Requirements

OS:	Windows 8.1 64bit or latest version
SQL Server:	Micro Soft SQL Server Express 2014
Web browser:	Google Chrome / Mozilla Firefox or another one
IDE:	Visual Studio 2015 or Latest version
Reports:	Crystal Reports XI or Latest version
Office:	Microsoft Office 2013 or Latest Version

- **2nd Step**

Install MSSQL Server

Visiting Microsoft website can download and install MSSQL Server 2014 Express version.
https://www.microsoft.com/en-us/download/details.aspx?id=42299

Install web browser

Visiting Google Chrome web site can download and install Google Chrome web browser.
https://www.google.com/chrome/browser/desktop/index.html

- **3rd Step**

Restore database

Create empty database in MSSQL Server
Open empty database SQL Script using MSSQL Server query window.
Change created data base name as use database.
Run the empty database SQL Script in MSSQL Server query window. Now created database in MSSQL server successful.

- **4th Step**

Extract Files

Web application file extract into specific place in the pc.

- **5th Launch**

Open web application typing the URL in the web browser.

Appendix B - Design Documentation

Use case diagrams and description

The following segment describes use case diagrams for other modules of the system continuing chapter 3.

Purchasing Manage

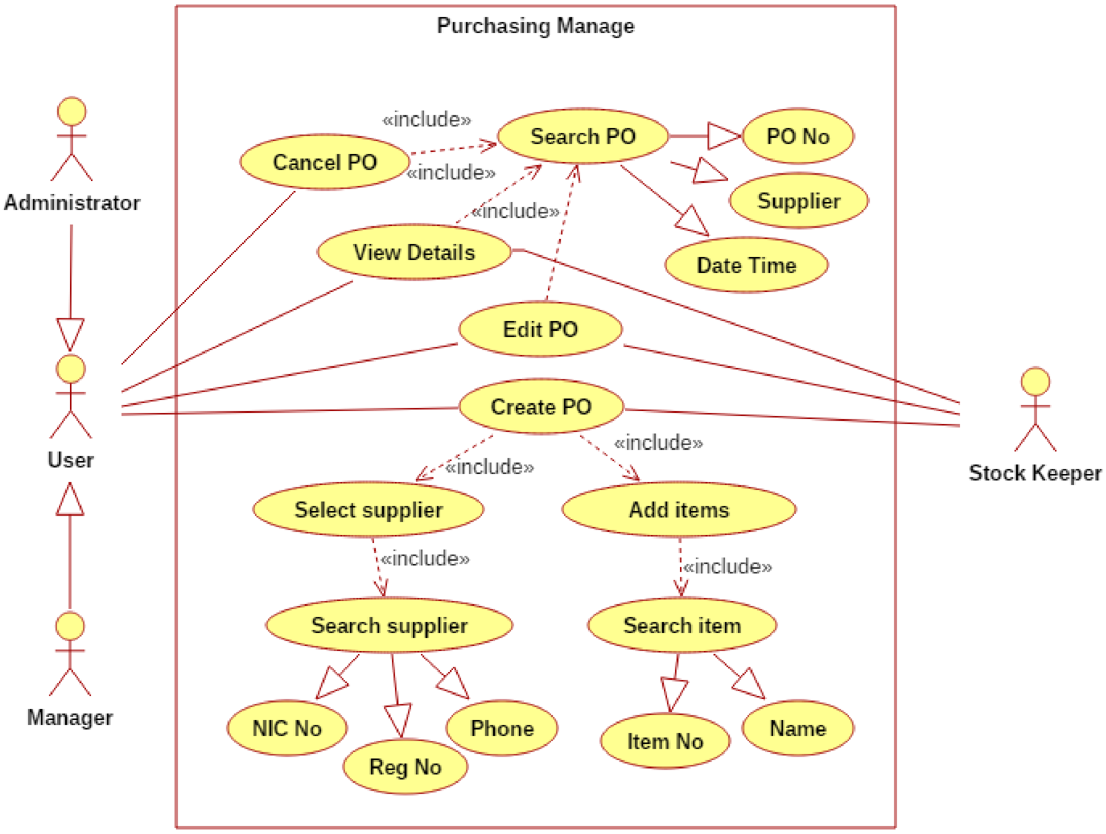


Figure B.1 Use case diagram for purchase order management

Use Case	Create purchase order
Actor	Administrator, Manager, Stok keeper
Description	Create purchase order for already registered suppliers
Pre-Conditions	
Supplier already registered in the organization. Item is already added into the system. User should logged to the system	
Flow of Events	
1. User select supplier use supplier search by given criteria. 2. User select one or more items use items search by given criteria. 3. Enter order quantity or the selected item. 4. Click on add button for add to purchase order. 5. Click on save button for save purchase order.	
Post-Conditions	

Save new purchase order in for future use Send purchase order to supplier email address as an email. Generate purchase order for printing.

Table B.1 Create new purchase order

Use Case	Edit purchase order
Actor	Administrator, Manager, Stok keeper
Description	Edit already crated purchase order by changing supplier or added items in that.
Pre-Conditions	
Supplier already registered in the organization. Item is already added into the system. Purchase order already available in the system. User should logged to the system	
Flow of Events	
1. User select purchase order search by given criteria. 2. User change supplier using supplier search by given criteria. 3. User change selected items use items search by given criteria and remove button. 4. Enter order quantity or the selected item. 5. Click on change button for change purchase order.	
Post-Conditions	
Change exist purchase order for future use. Send modified purchase order to supplier email address with a notice as an email. Generate modified purchase order for printing.	

Table B.2 Edit exist purchase order

Use Case	View details purchase order
Actor	Administrator, Manager, Stok keeper
Description	View full details about already added purchase order
Pre-Conditions	
Purchase order already available in the system. User should logged to the system	
Flow of Events	
1. User select purchase order use search by given criteria. 2. Click on detail button for view details purchase order.	
Post-Conditions	
Popup detail window and show all details about purchase order	

Table B.3 View details purchase order

Use Case	Cancel purchase order
Actor	Administrator, Manager
Description	Cancel already added purchase order for supplier
Pre-Conditions	
Purchase order already available in the system. User should logged to the system	
Flow of Events	
1. User select purchase order use search by given criteria. 2. Click on delete button for cancel purchase order.	

3. Confirm message about purchase order delete confirmation.	
Post-Conditions	
Purchase order states changed as deleted one. Save changes of the purchase order as deleted one. Purchase order cancellation notice send to supplier email address as an email.	

Table B.4 Cancel purchase order

Use Case	Search purchase order
Actor	Administrator, Manager, Stock Keeper
Description	Search exist purchase order in the system
Pre-Conditions	
Purchase order already available in the system. User should logged to the system	
Flow of Events	
1. User go to the purchase order search window. 2. Enter PO No. or Date or Supplier for search purchase order. 3. Listed all related purchase orders with best match one. 4. Click on select button for select a purchase order.	
Post-Conditions	
Purchase order search window will close automatically. Purchase order details assign in to required view.	

Table B.5 Search purchase order

Use Case	Search suppler
Actor	Administrator, Manager, Stock Keeper
Description	Search already registered supplier in the system
Pre-Conditions	
Supplier already registered in the system. User should logged to the system	
Flow of Events	
1. User go to the supplier search window. 2. Enter Supplier No. or NIC No. or Phone for search supplier. 3. Listed all related supplier with best match. 4. Click on select button for select a supplier.	
Post-Conditions	
Supplier search window will close automatically. Supplier details assign in to required view.	

Table B.6 Search suppler

Use Case	Search item
Actor	Administrator, Manager, Stock Keeper
Description	Search already added item in the system
Pre-Conditions	
Item already available in the system. User should logged to the system	
Flow of Events	
1. User go to the item search window. 2. Enter Item No. or Name for search an item.	

3. Listed all related items with best match in stock. 4. Click on select button for select an item.	
Post-Conditions	
Item search window will close automatically. Item details assign in to required view.	

Table B.7 Search item

Goods Received Note Management

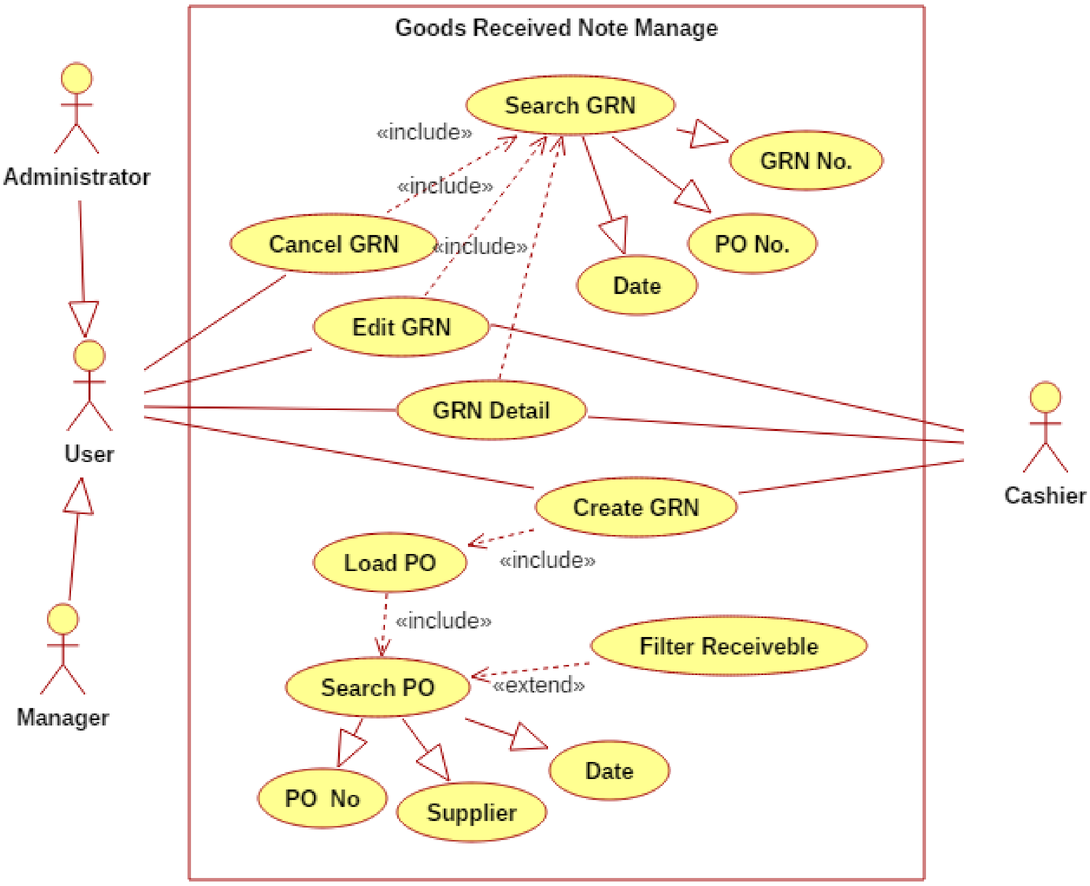


Figure B.2 Use case diagram for goods received note management

Use Case	Create goods received note
Actor	Administrator, Manager, Stock Keeper
Description	Create goods received note for already added purchase order. A purchase order may have one or more goods received notes.
Pre-Conditions	
Purchase order already available in to the system. User should logged to the system	
Flow of Events	
1. User load purchase order search by given criteria. 2. Fill received quantities of the loaded purchase order. 3. If some items not received, remove those items in loaded purchase order item list. 4. Click on save button for save purchase order.	
Post-Conditions	

Goods received note store for future use. Stock item quantities up to date according to goods received note. Purchase order receivable quantities are up to date. Generate goods received note for print.

Table B.8 Create goods received note

Use Case	Edit goods received note
Actor	Administrator, Manager, Stock Keeper
Description	Change already added goods received note received quantities for already added purchase order.
Pre-Conditions	
Purchase order already available in to the system. Goods received note already added in to the system User should logged to the system	
Flow of Events	
1. User load goods received note search by given criteria. 2. Fill received quantities changes of the goods received note. 3. If some items not received, remove those items in loaded purchase order item list. 4. Click on change button for change purchase order.	
Post-Conditions	
Goods received note store for future use. Stock item quantities roll back and up to date according to goods received note. Purchase order receivable quantities are up to date. Generate goods received note for print.	

Table B.9 Change goods received note

Use Case	Cancel goods received note
Actor	Administrator, Manager
Description	Change already added goods received note status as deleted one.
Pre-Conditions	
Purchase order already available in to the system. Goods received note already added in to the system User should logged to the system	
Flow of Events	
1. User select goods received note search by given criteria. 2. Click on delete button for delete goods received note.	
Post-Conditions	
Change goods received note status as deleted. Stock item quantities roll back according to deleted goods received note quantities.	

Table B.10 Delete goods received note

Use Case	Goods received note details
Actor	Administrator, Manager, Stok keeper
Description	View full details about already added goods received note
Pre-Conditions	
Purchase order already available in the system. Goods received note already available in the system. User should logged to the system	

Flow of Events
1. User select goods received note use search by given criteria. 2. Click on detail button for view details of goods received note.
Post-Conditions
Popup detail window and show all details about goods received note

Table B.11 View details goods received note

Use Case	Search goods received note
Actor	Administrator, Manager
Description	Search already added goods received note in the system
Pre-Conditions	
Purchase order already available in to the system. Goods received note already added in to the system User should logged to the system	
Flow of Events	
1. User go to the goods received note search window. 2. Enter GRN No., PO No. or Date of created for search an item. 3. Listed all related goods received notes with best match in stock. 4. Click on select button for select goods received note.	
Post-Conditions	
Goods received note search window will close automatically. Selected goods received note details assign in to required view.	

Table B.12 Search goods received note

Use case diagram for sales manage - direct

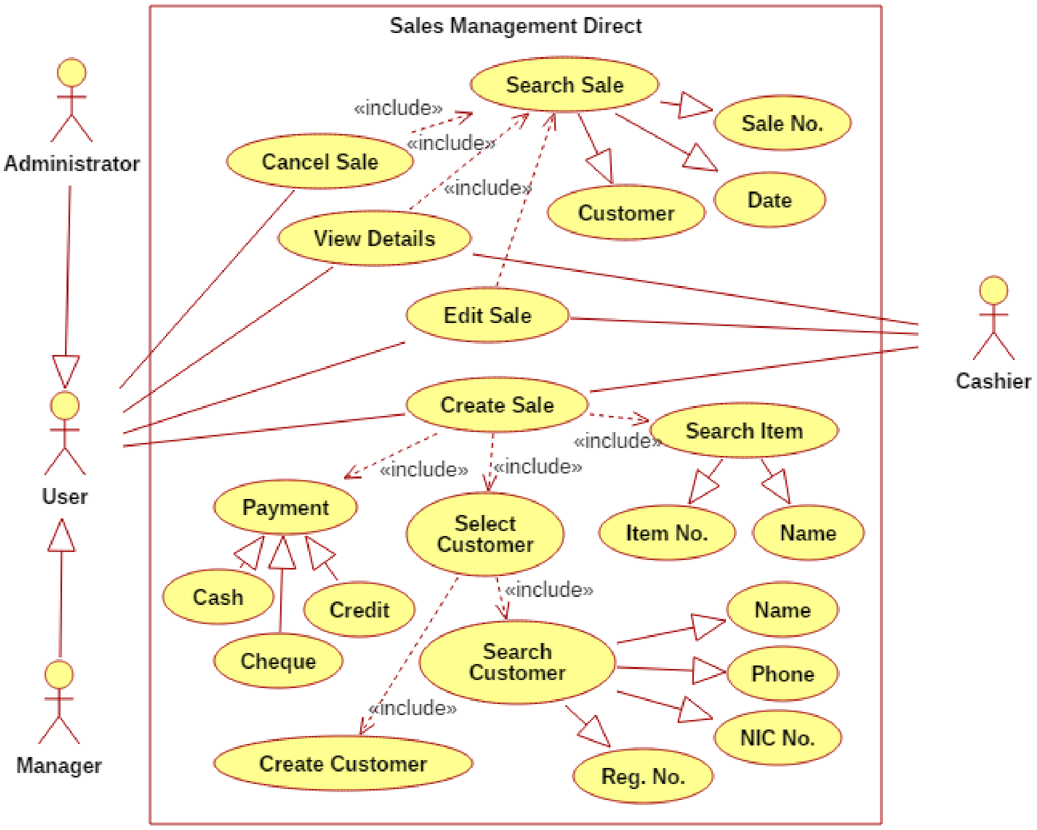


Figure B.3 Use case diagram for direct sales

Use Case	Create sale direct
Actor	Administrator, Manager, Cashier
Description	Create a new sale for already registered customer. If customer is not registered need to register customer firstly in the system.
Pre-Conditions	
Customer already registered in the system. Items are already available in the system and stock. User should logged to the system	
Flow of Events	
1. User select a customer using search by given criteria or registering new customer. 2. Select one or more items for sale by items search by given criteria. 3. Enter item quantities and total discounts for sale. 4. Click on save button for save direct sale.	
Post-Conditions	
Direct sale store for future use. Stock item quantities and accounts up to date according to direct sale. Generate invoice for print.	

Table B.13 Create direct sale

Use Case	Edit sale direct
Actor	Administrator, Manager, Cashier
Description	Edit already created direct sale details.
Pre-Conditions	
Direct sale already available in to the system. User should logged to the system	
Flow of Events	
1. User load direct sale search by given criteria. 2. Change relevant details editing the selected direct sale. 3. Click on change button for change direct sale.	
Post-Conditions	
Direct sale changed and store for future use. Stock item quantities and accounts roll back and up to date according to direct sale changes. Generate changed direct sale invoice for print.	

Table B.14 Edit exist direct sale

Use Case	Cancel sale direct
Actor	Administrator, Manager
Description	Change direct sale status as deleted and store direct sale.
Pre-Conditions	
Direct sale is already available in the system. User should logged to the system	
Flow of Events	
1. User select direct sale search by given criteria. 2. Click on delete button for delete direct sale.	
Post-Conditions	
Change direct sale status as deleted and store for future use. Stock item quantities and accounts are roll back and up to date according to direct sale.	

Table B.15 Delete exist direct sale

Use Case	View details
Actor	Administrator, Manager, Cashier
Description	Show direct sale full description include customer, items and prepared user details.
Pre-Conditions	
Direct sale already available in the system. User should logged to the system	
Flow of Events	
1. User select direct sale search by given criteria. 2. Load direct sale details on direct sale details window.	
Post-Conditions	
load direct sale details on direct sale details window	

Table B.16 View details exist direct sale

Activity Diagrams

Create new repair PC order accept - Repair PC module

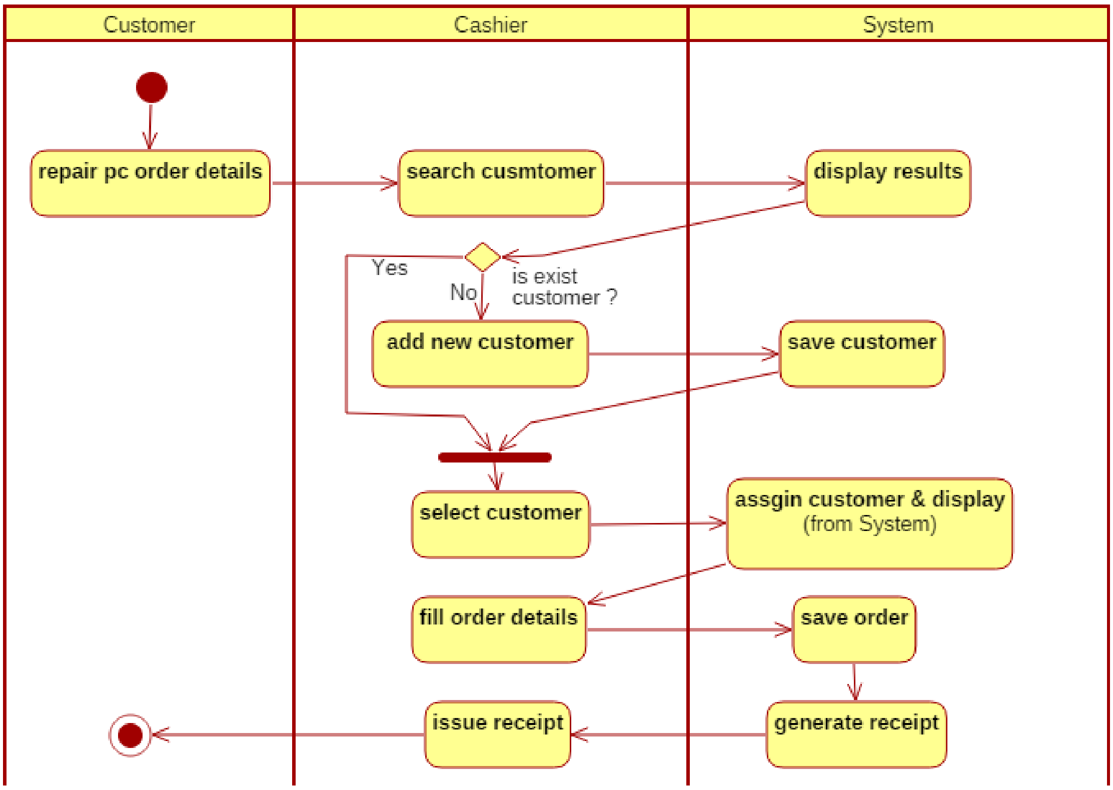


Figure B.4 Activity diagram for create new repair PC order accept - Repair PC module

Create new direct sale - Direct sale module

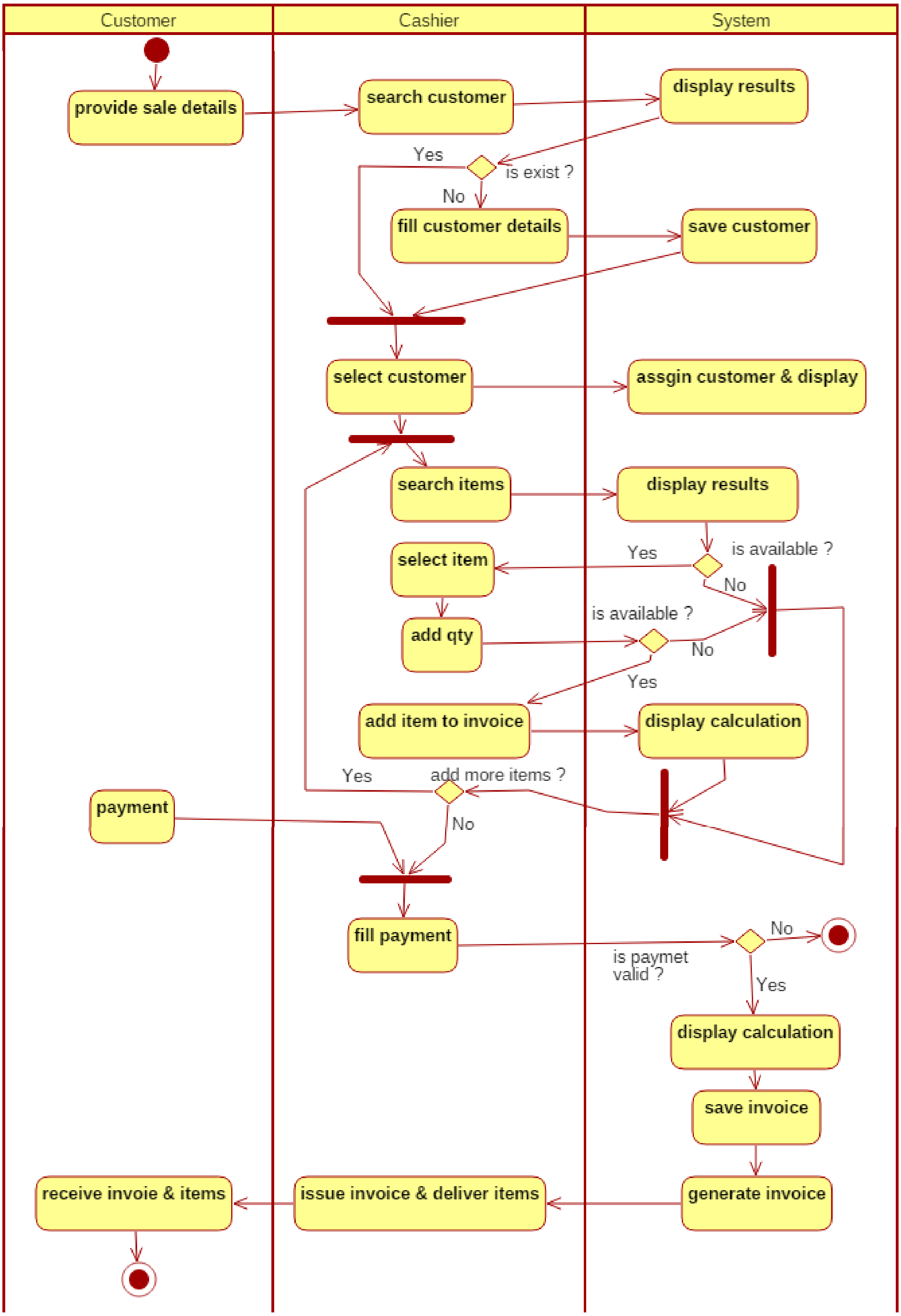


Figure B.5 Activity diagram for create new direct sale - direct sale module

Create invoice for repair PC order - Repair PC module

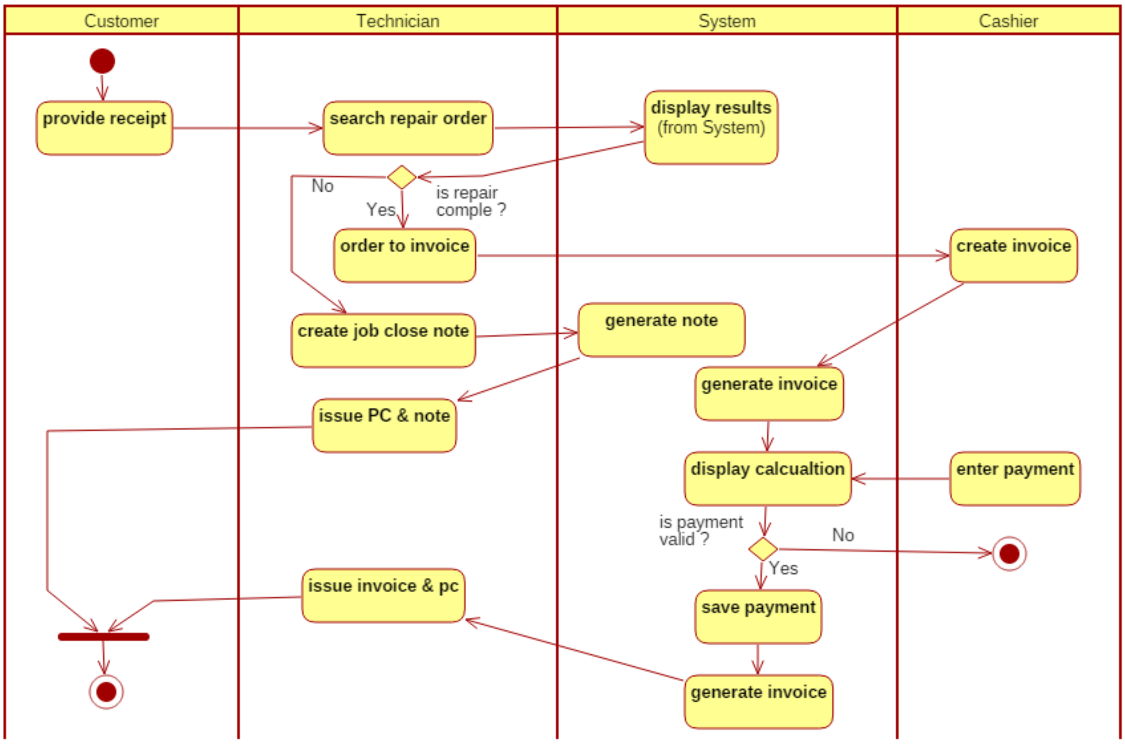


Figure B.6 Activity diagram for create invoice for repair PC order - Repair PC module

Sequence Diagrams

Sequence diagram for create new repair PC order accept - Repair PC module

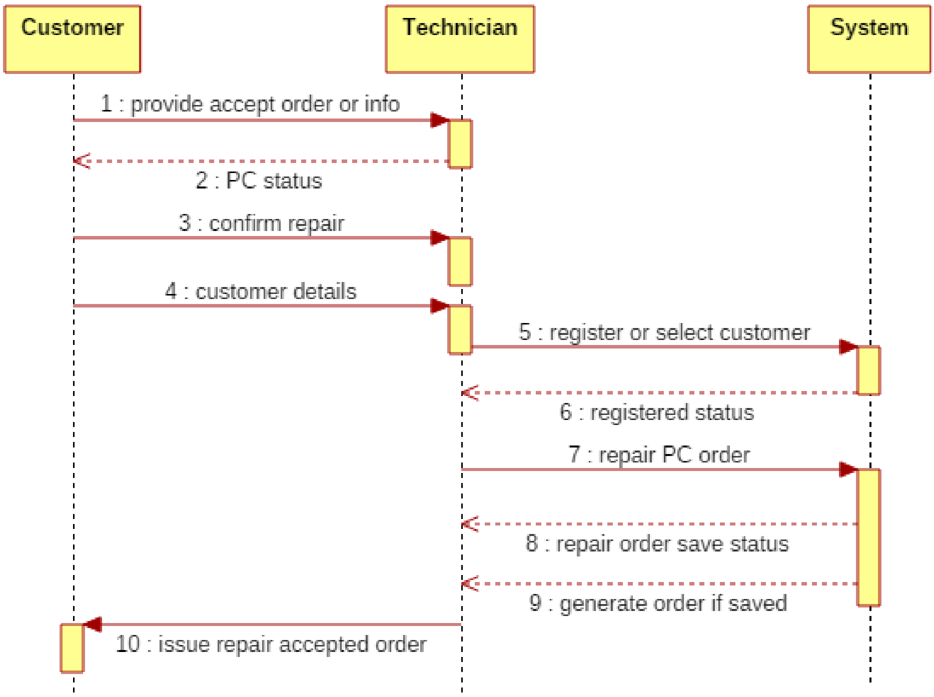

Customer
Technician
System
1 : provide accept order or info
2 : PC status
3 : confirm repair
4 : customer details
5 : register or select customer
6 : registered status
7 : repair PC order
8 : repair order save status
9 : generate order if saved
10 : issue repair accepted order

Figure B.7 Sequence diagram for create new repair PC order accept

Sequence Diagrams

Sequence diagram for create invoice for repair PC order - Repair PC module

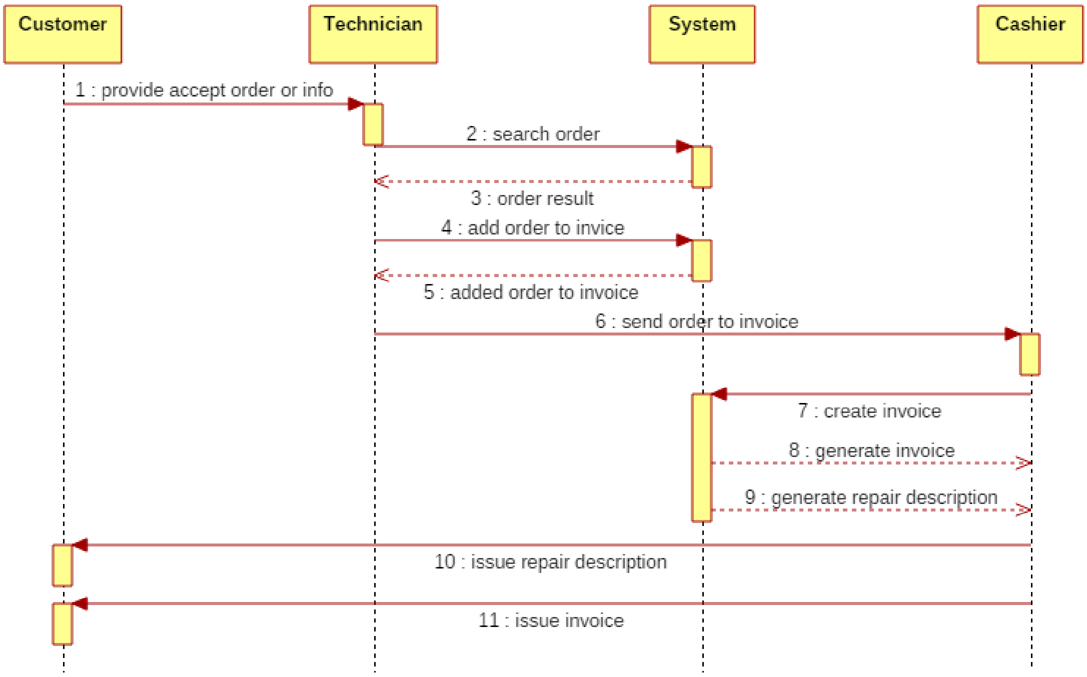

Customer
Technician
System
Cashier
1 : provide accept order or info
2 : search order
3 : order result
4 : add order to invice
5 : added order to invoice
6 : send order to invoice
7 : create invoice
8 : generate invoice
9 : generate repair description
10 : issue repair description
11 : issue invoice

Figure B 1 Sequence diagram for create invoice for repair PC order

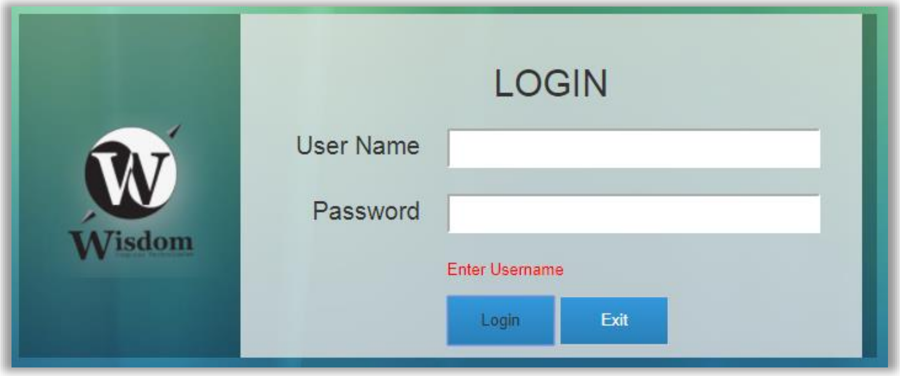
Appendix C - User Documentation

Administrator (super admin), manager (admin), cashier, technician and other users can access the system. Only super admin has privileges for user's management and privileges assign for other users. Other users access the under received privileges.

System Login & Dashboard

- **Login form**

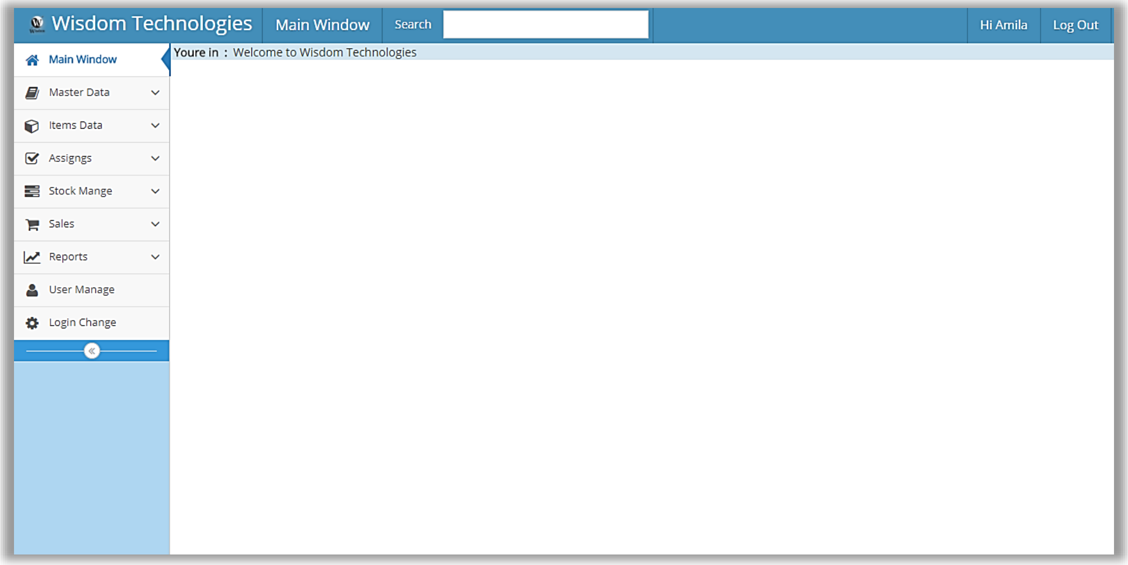
Application start form login page. Block redirect for inside other pages without login to application. All system user can be login to system using this form. If user provide incorrect details block login to system. After define login try time lock account system automatically. If user provide correct login info redirect to main dash board according to user type.



Enter Username

Figure C.1 System login page

- **Dashboard**



Youre in : Welcome to Wisdom Technologies

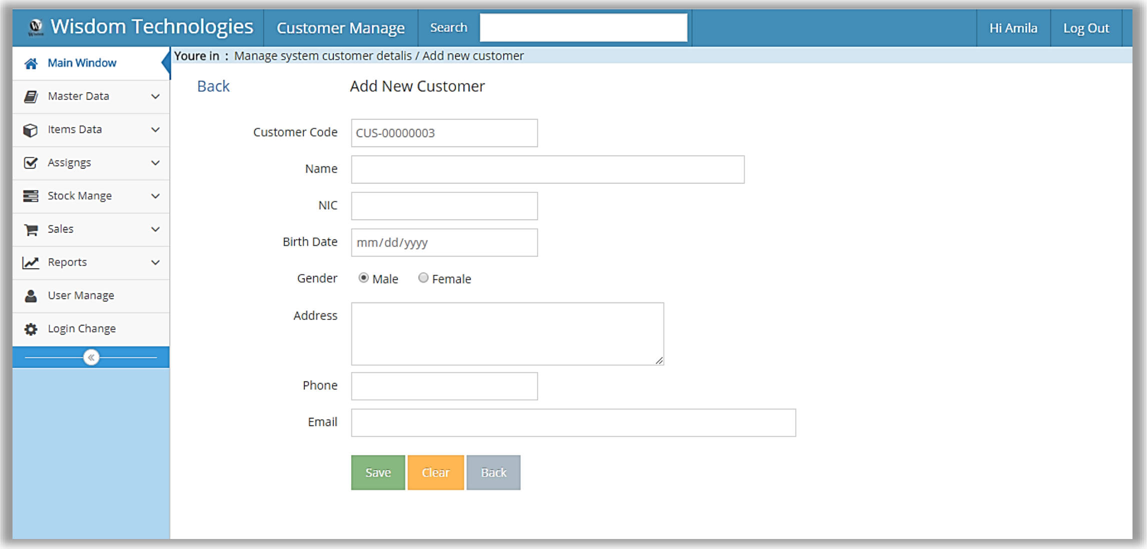
Figure C.2 Dashboard of administrator page

Bellow *Figure C 2 Dashboard* include all system functionalities according to the logged user type. Top menu of the dashboard provide login user, log out, searching and page title. Left side expandable menu bar represent the system main menu.

Customer Management

- **Register New Customer**

Entering valid customer details to the system by customer provided information can register new customer correct way. Customer code is generated by system. Need enter valid NIC number, NIC number is optional. As well as date of birth can choose use calendar optionally. Name, address, gender and phone are required.



Youre in : Manage system customer detalis / Add new customer

Back   Add New Customer

Customer Code: CUS-00000003

Name:

NIC:

Birth Date: mm/dd/yyyy

Gender: ● Male ○ Female

Address:

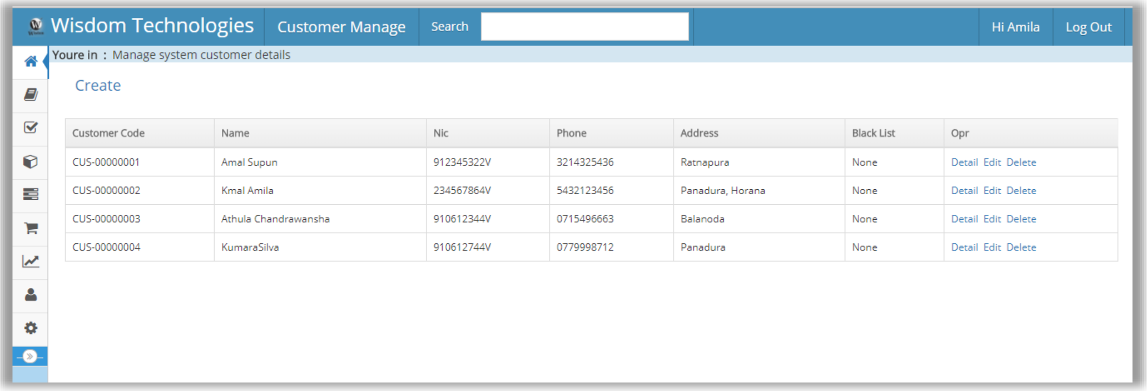
Phone:

Email:

Figure C.3 Customer register page

- **Customer List**

Show all customers who registered in the system. Using search box in top menu of the page can search customer by name, NIC no., Reg. No. or telephone. Get customer details, change customer or delete customer can be done using customer operations of Add, Edit and Delete.

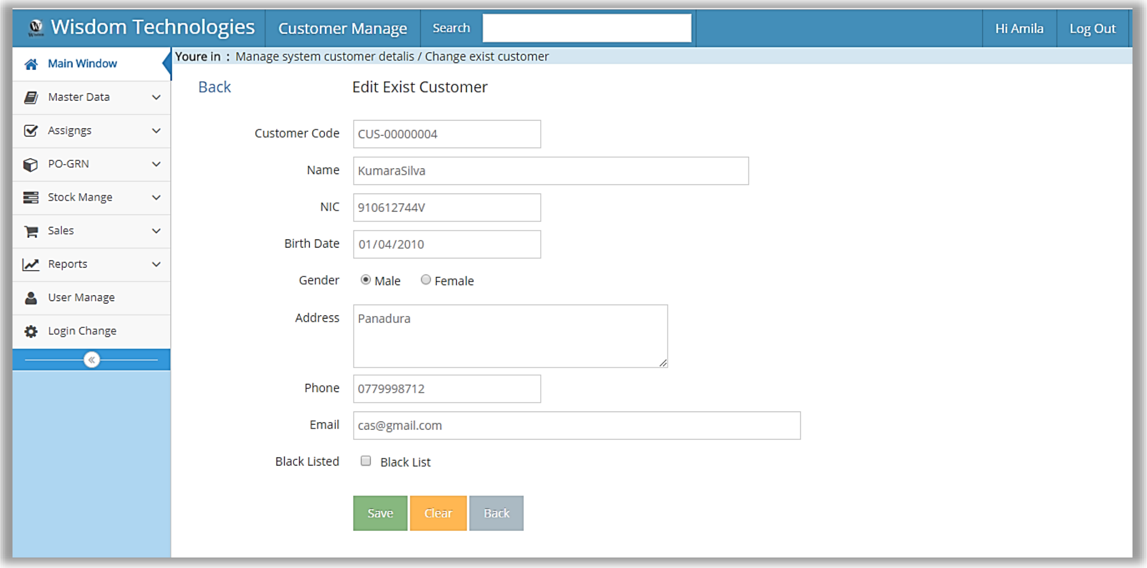


Youre in : Manage system customer details

Customer Code	Name	Nic	Phone	Address	Black List	Opr
CUS-00000001	Amal Supun	912345322V	3214325436	Ratnapura	None	Detail Edit Delete
CUS-00000002	Kmal Amila	234567864V	5432123456	Panadura, Horana	None	Detail Edit Delete
CUS-00000003	Athula Chandrawansha	910612344V	0715496663	Balanoda	None	Detail Edit Delete
CUS-00000004	KumaraSilva	910612744V	0779998712	Panadura	None	Detail Edit Delete

Figure C.4 Customer list page

- **Change Customer**



Youre in : Manage system customer details / Change exist customer

Back Edit Exist Customer

Customer Code: CUS-00000004

Name: KumaraSilva

NIC: 910612744V

Birth Date: 01/04/2010

Gender: ● Male ○ Female

Address: Panadura

Phone: 0779998712

Email: cas@gmail.com

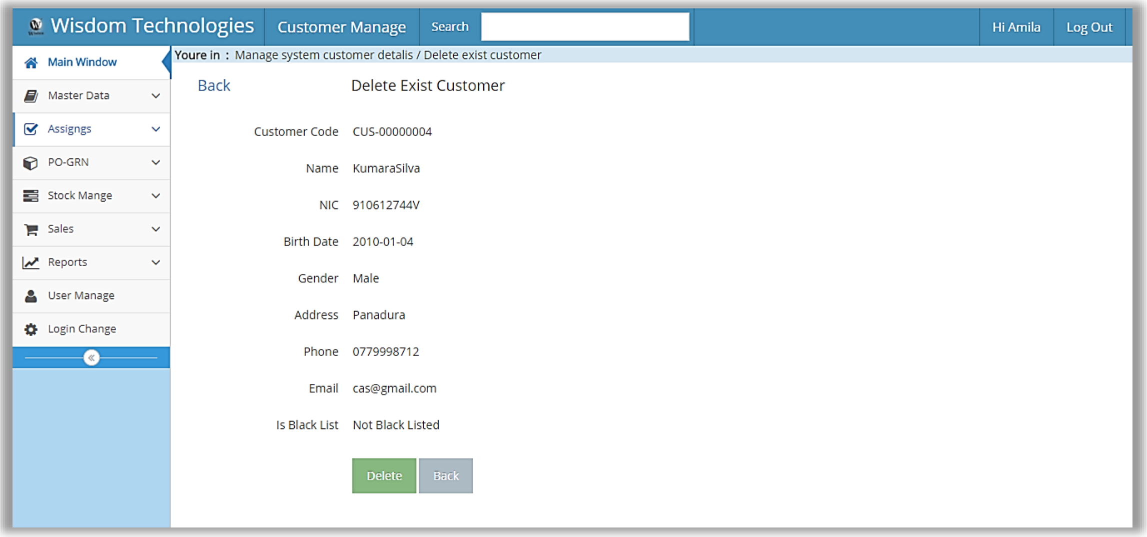
Black Listed: □ Black List

Figure C.5 Change customer page

Click on Edit option in searched customer in customer list redirect to this edit page with auto filled customer current details. User can save editing all customer details without customer code. User can enter edited data under the system validation. As example NIC No., Telephone and Email cannot be duplicated. Using back button or link go to the customer list page.

- **Delete Customer**

Click on Edit option in searched customer in customer list redirect to following delete confirmation page. Delete confirmation page represent all details about going to be deleted customer. After click on delete button delete customer and redirect to customer list page of the system automatically.



Youre in : Manage system customer details / Delete exist customer

Back Delete Exist Customer

Customer Code: CUS-00000004

Name: KumaraSilva

NIC: 910612744V

Birth Date: 2010-01-04

Gender: Male

Address: Panadura

Phone: 0779998712

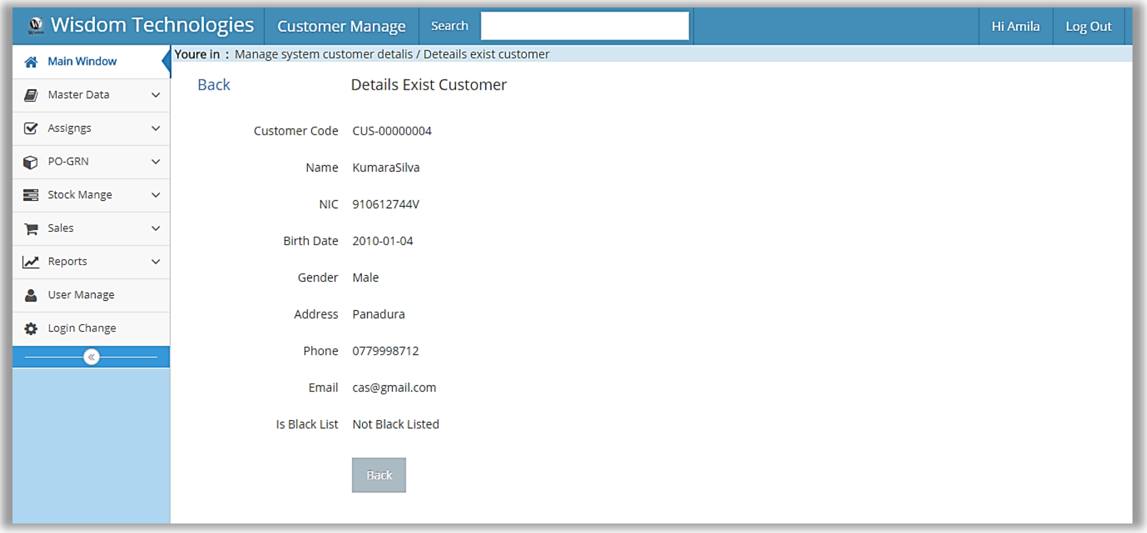
Email: cas@gmail.com

Is Black List: Not Black Listed

Figure C.6 Customer delete page

- ### Customer Details

Click on Detail option in searched customer in customer list redirect to following details page. The details exist customer page show all details about the selected customer as following *Figure C 7*.



Youre in : Manage system customer detalis / Deteails exist customer

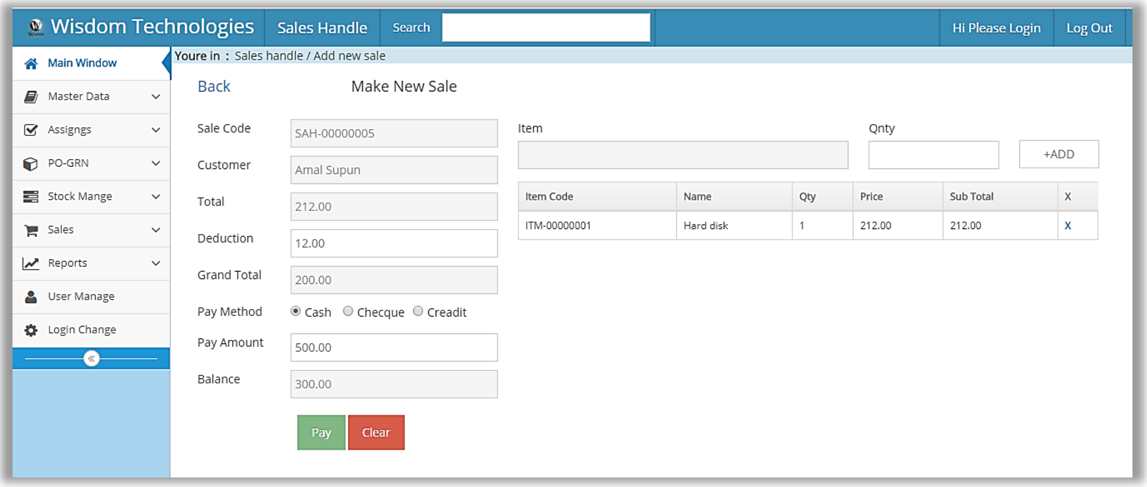
Back Details Exist Customer

Customer Code	CUS-00000004
Name	KumaraSilva
NIC	910612744V
Birth Date	2010-01-04
Gender	Male
Address	Panadura
Phone	0779998712
Email	cas@gmail.com
Is Black List	Not Black Listed

Figure C.7 Customer details page

Direct Sale Management

- ### Create Direct Sale



Youre in : Sales handle / Add new sale

Back Make New Sale

Sale Code	SAH-00000005
Customer	Amal Supun
Total	212.00
Deduction	12.00
Grand Total	200.00
Pay Method	● Cash ○ Checque ○ Creadit
Pay Amount	500.00
Balance	300.00

Item Qnty

Item Code	Name	Qty	Price	Sub Total	X
ITM-00000001	Hard disk	1	212.00	212.00	X

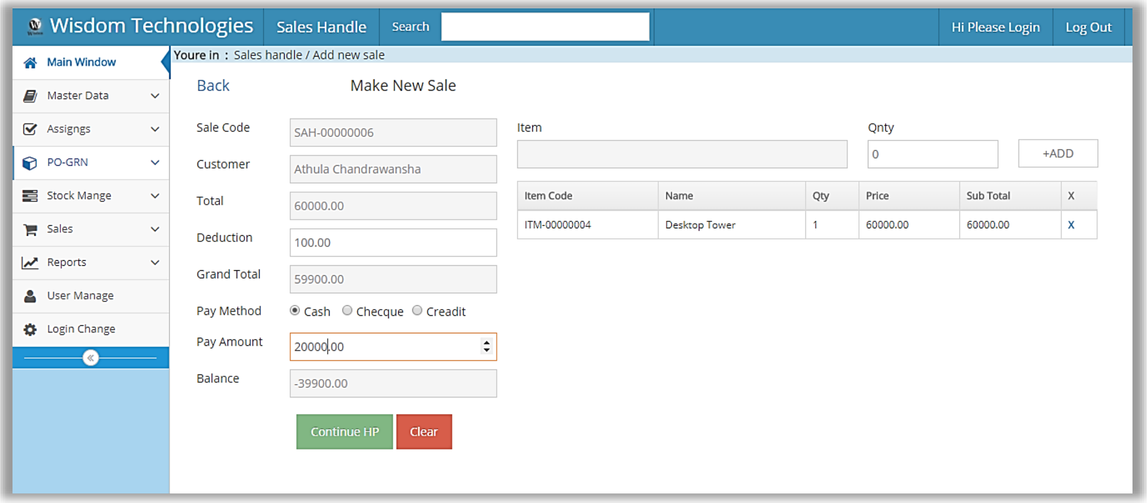
Figure C.8 Create direct sale page

Using bellow *Figure C 8* user can create direct sale. Sale header code is auto generate by system. Click on customer field popup customer search window. Customers search can be done by Reg. No., NIC No., phone or name. After click on select button in customer search box can add customer to direct sale. For add items click on item field and then popup item search window. Items search can be done by item Reg. No. or name. Can add one or more items by selecting search box items. Click on x button can remove item for list. Click on pay button save sale and invoice will generate by system.

Hire Purchase Sale Management

▪ Create New Hire Purchase

Hire purchase creating process broken to tow steps as wizard. In first step user need provide sale information. This is different from direct sale by user entered pay amount. According to pay amount system will continue as hire purchase. Entering selling item quantities system will check with selling quantity with available quantity. User can select payment methodologies for received payment. If user select check or credit card need its information add to system firstly.



Youre in : Sales handle / Add new sale

Back Make New Sale

Sale Code	SAH-00000006
Customer	Athula Chandrawansha
Total	60000.00
Deduction	100.00
Grand Total	59900.00
Pay Method	● Cash ○ Checque ○ Creadit
Pay Amount	20000.00
Balance	-39900.00

Continue HP Clear

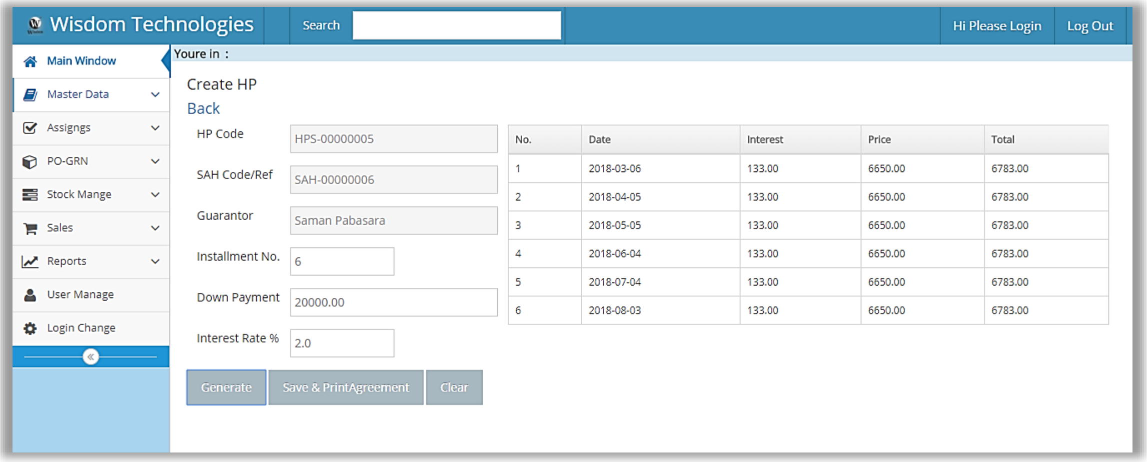
Item

Qnty 0 +ADD

Item Code	Name	Qty	Price	Sub Total	X
ITM-00000004	Desktop Tower	1	60000.00	60000.00	X

Figure C.9 Hire purchase sale info page

Click on Continue HP button, system provide following Figure C 10 page as second step of the wizard. HP Code and SAH Code will generated by system automatically.



Youre in :

Create HP

Back

HP Code	HPS-00000005
SAH Code/Ref	SAH-00000006
Guarantor	Saman Pabasara
Installment No.	6
Down Payment	20000.00
Interest Rate %	2.0

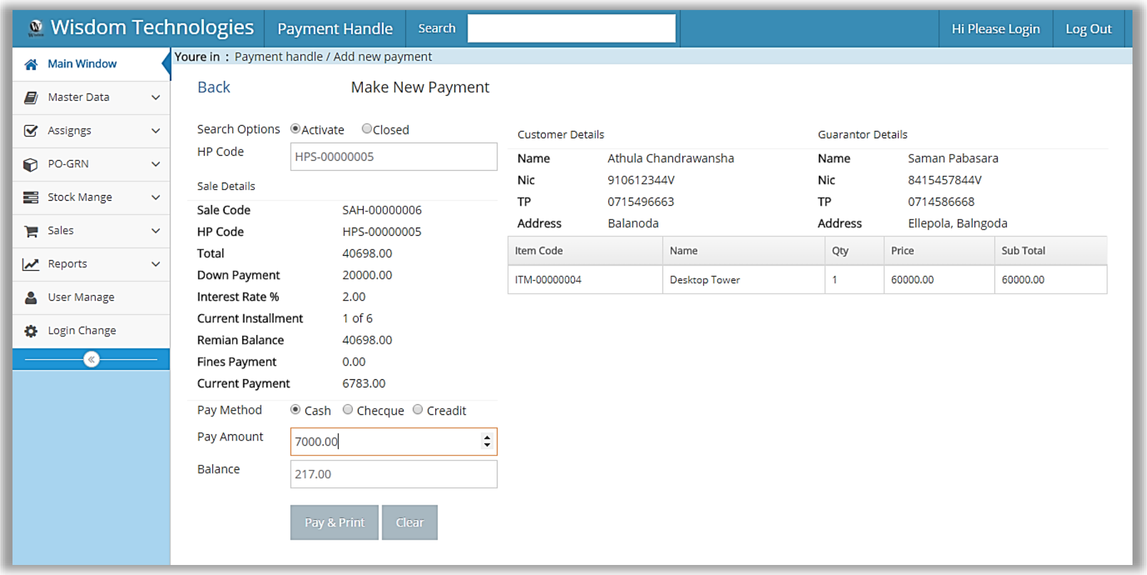
Generate Save & PrintAgreement Clear

No.	Date	Interest	Price	Total
1	2018-03-06	133.00	6650.00	6783.00
2	2018-04-05	133.00	6650.00	6783.00
3	2018-05-05	133.00	6650.00	6783.00
4	2018-06-04	133.00	6650.00	6783.00
5	2018-07-04	133.00	6650.00	6783.00
6	2018-08-03	133.00	6650.00	6783.00

Figure C.10 Hire purchase payment info

Bellow create hire purchase window user enter valid number of installments, down payment with rates. Then click on Generate button generate right side table included information about all installments. Finally user click on Save & Print Agreement button system will generate agreement automatically.

- **Create Hire Purchase Payment**



Youre in : Payment handle / Add new payment

Make New Payment

Search Options ● Activate ○ Closed

HP Code: HPS-00000005

Sale Details

Sale Code	SAH-00000006
HP Code	HPS-00000005
Total	40698.00
Down Payment	20000.00
Interest Rate %	2.00
Current Installment	1 of 6
Remian Balance	40698.00
Fines Payment	0.00
Current Payment	6783.00

Pay Method ● Cash ○ Checque ○ Creadit

Pay Amount: 7000.00

Balance: 217.00

Customer Details

Name	Athula Chandrawansha
Nic	910612344V
TP	0715496663
Address	Balanoda

Guarantor Details

Name	Saman Pabasara
Nic	8415457844V
TP	0714586668
Address	Ellepola, Balngoda

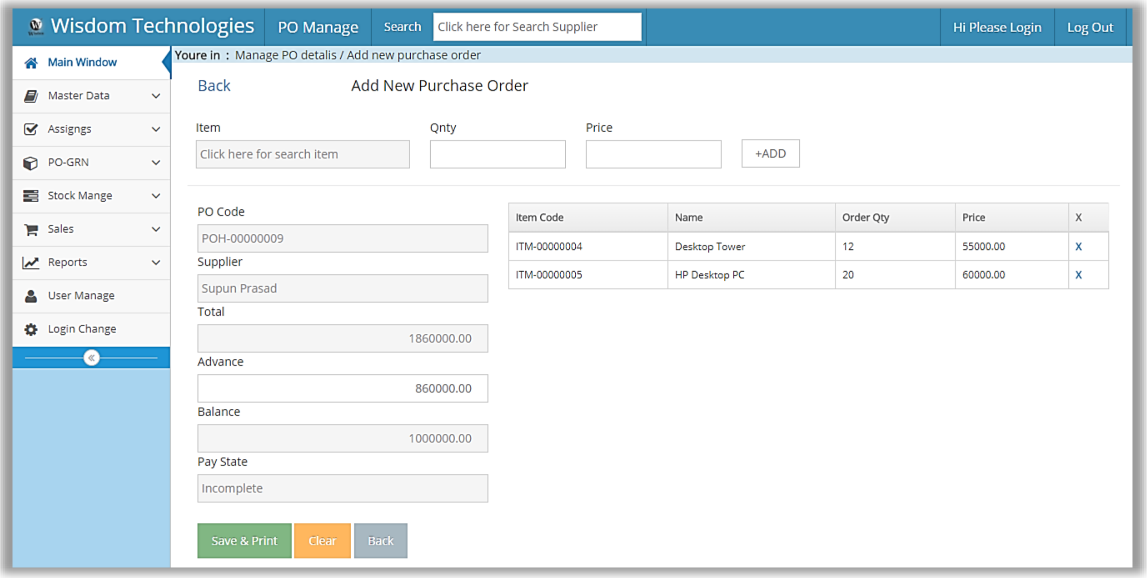
Item Code	Name	Qty	Price	Sub Total
ITM-00000004	Desktop Tower	1	60000.00	60000.00

Figure C.11 Hire purchase payment form

For complete hire purchase account need to be received all installments. Bellow form is provide receive installment payments of the system. Firstly need select hire purchase account using hire purchase search pop window. After select account show all details about hire purchase account in this form. Then need to be add pay amount of the installment selecting payment method. Then click on “Pay & Print” button system will generated cash receipt. If all installment are completed generate invoice.

Purchase Order Mange

- **Create Purchase Order**



Youre in : Manage PO details / Add new purchase order

Add New Purchase Order

Item: Click here for search item

Qnty

Price

PO Code: POH-00000009

Supplier: Supun Prasad

Total: 1860000.00

Advance: 860000.00

Balance: 1000000.00

Pay State: Incomplete

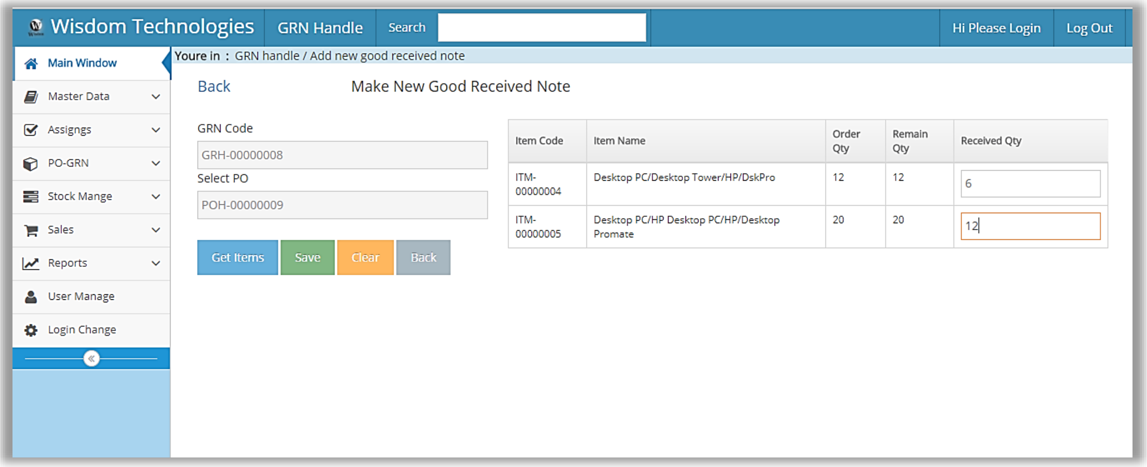
Item Code	Name	Order Qty	Price	X
ITM-00000004	Desktop Tower	12	55000.00	X
ITM-00000005	HP Desktop PC	20	60000.00	X

Figure C.12 Create purchase order page

For create hire purchase user need select supplier firstly by searching supplier. Supplier search can be done by code, name, NIC or telephone. Then need add items to purchase order by searching items. Items can be search by item code, name or address. System will provide items for search filter the selected supplier supply items. User can add one or more items for purchase order with relevant quantities using +ADD button. Then providing advance payment and click on Save & Print button save purchase order and system will generate purchase order automatically.

Goods Received Note Mange

- **Create Goods Received Note**



Youre in : GRN handle / Add new good received note

Make New Good Received Note

GRN Code: GRH-00000008

Select PO: POH-00000009

Item Code	Item Name	Order Qty	Remain Qty	Received Qty
ITM-00000004	Desktop PC/Desktop Tower/HP/DskPro	12	12	6
ITM-00000005	Desktop PC/HP Desktop PC/HP/Desktop Promate	20	20	12

Figure C.13 Create goods received note page

Using *Figure C 13* page can create goods received note for selected purchase order. One purchase order can create one or more goods received notes. User selecting purchase order click on Select PO field, will pop up purchase order search window. Purchase order can search by PO Code or supplier.

After selecting purchase order click on Get Items button, system will generate right side table. It included details about order items of the selected purchase order. Now user can fill the received quantities of order items. User entered received quantities check with remaining quantities by system.

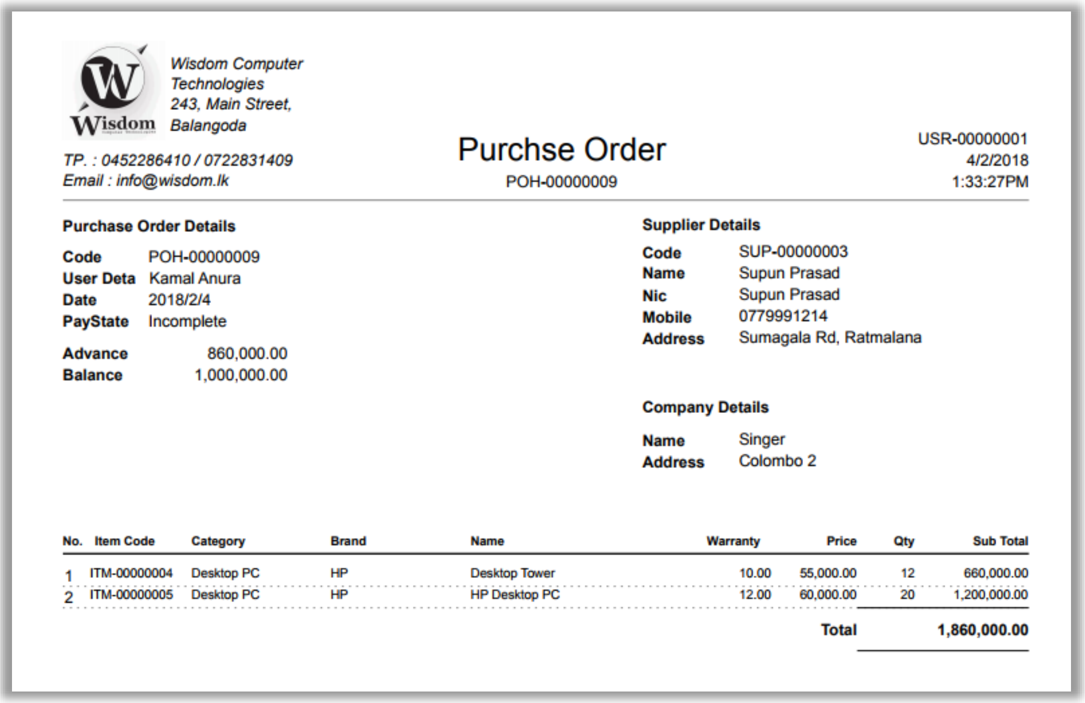
Finally click on “Save” button save purchase order and system will generate purchase order automatically.

Appendix D - Management Reports

Individual Master Reports

Shown following figures individual master data report, generated after save data.

- **Purchase Order**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

Purchse Order
POH-00000009

USR-00000001
4/2/2018
1:33:27PM

Purchase Order Details

Code	POH-00000009
User Deta	Kamal Anura
Date	2018/2/4
PayState	Incomplete
Advance	860,000.00
Balance	1,000,000.00

Supplier Details

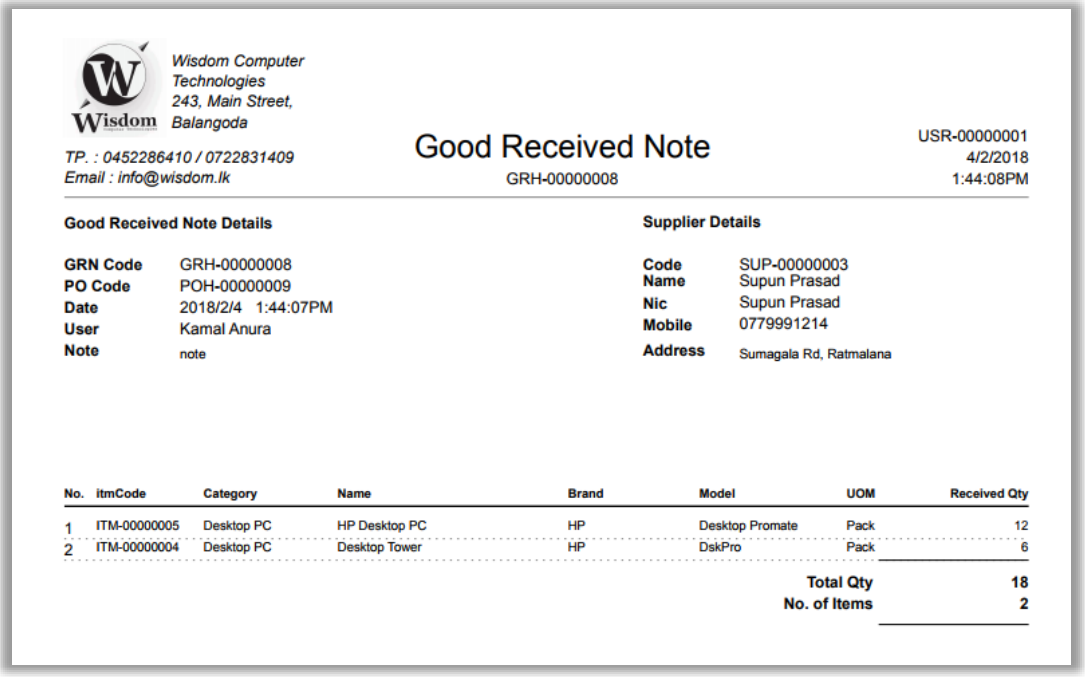
Code	SUP-00000003
Name	Supun Prasad
Nic	Supun Prasad
Mobile	0779991214
Address	Sumagala Rd, Ratmalana

Company Details

Name	Singer
Address	Colombo 2

No.	Item Code	Category	Brand	Name	Warranty	Price	Qty	Sub Total
1	ITM-00000004	Desktop PC	HP	Desktop Tower	10.00	55,000.00	12	660,000.00
2	ITM-00000005	Desktop PC	HP	HP Desktop PC	12.00	60,000.00	20	1,200,000.00
						Total		**1,860,000.00**

Figure D.1 Purchase order report



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

Good Received Note
GRH-00000008

USR-00000001
4/2/2018
1:44:08PM

Good Received Note Details

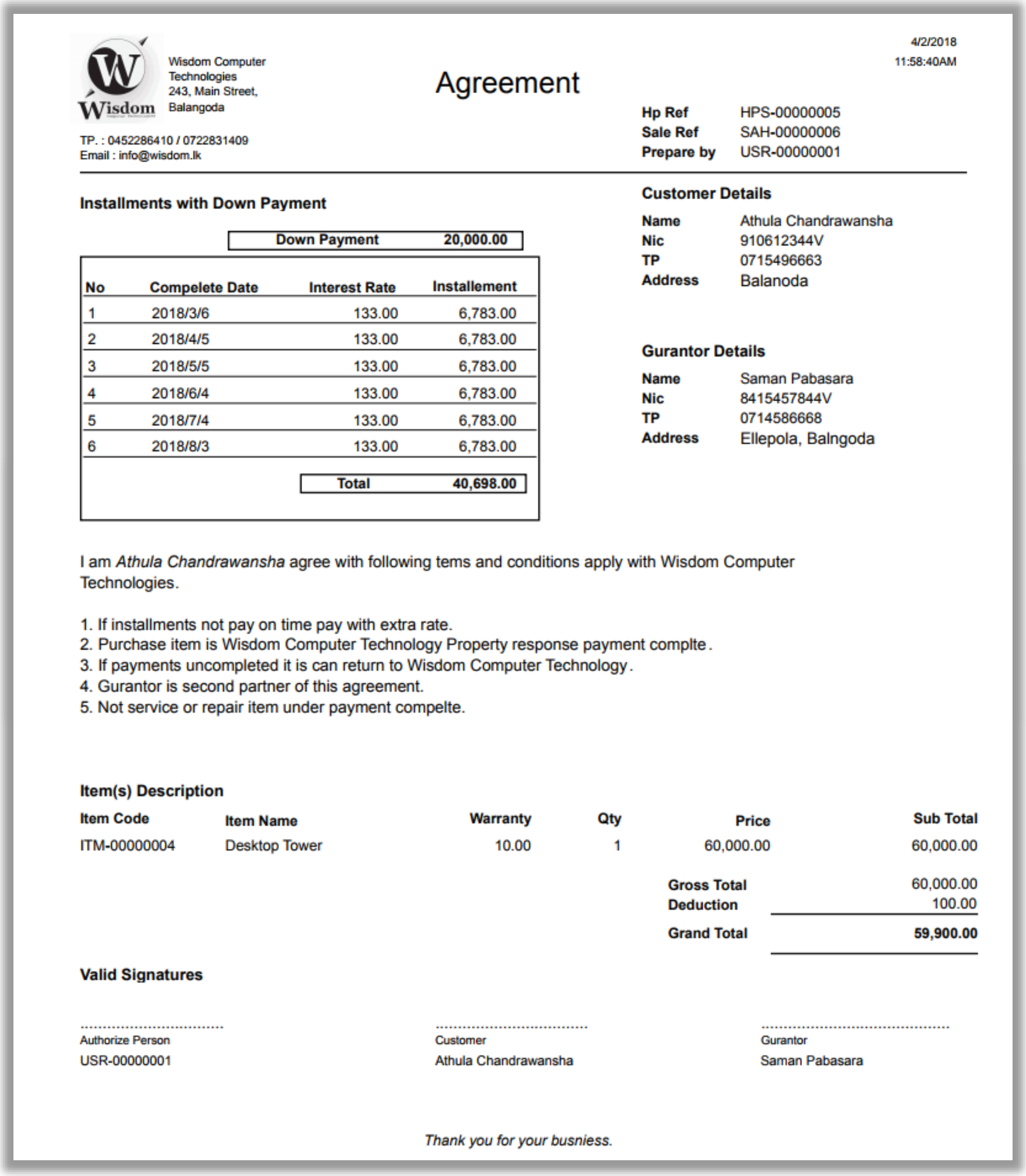
GRN Code	GRH-00000008
PO Code	POH-00000009
Date	2018/2/4 1:44:07PM
User	Kamal Anura
Note	note

Supplier Details

Code	SUP-00000003
Name	Supun Prasad
Nic	Supun Prasad
Mobile	0779991214
Address	Sumagala Rd, Ratmalana

No.	itmCode	Category	Name	Brand	Model	UOM	Received Qty
1	ITM-00000005	Desktop PC	HP Desktop PC	HP	Desktop Promate	Pack	12
2	ITM-00000004	Desktop PC	Desktop Tower	HP	DskPro	Pack	6
						Total Qty	**18**
						No. of Items	**2**

Figure D.2 Goods received note report



Wisdom Computer Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

4/2/2018
11:58:40AM

Agreement

Hp Ref	HPS-00000005
Sale Ref	SAH-00000006
Prepare by	USR-00000001

Installments with Down Payment

Down Payment	**20,000.00**

No	Compelete Date	Interest Rate	Installement
1	2018/3/6	133.00	6,783.00
2	2018/4/5	133.00	6,783.00
3	2018/5/5	133.00	6,783.00
4	2018/6/4	133.00	6,783.00
5	2018/7/4	133.00	6,783.00
6	2018/8/3	133.00	6,783.00
		Total	**40,698.00**

Customer Details

Name	Athula Chandrawansha
Nic	910612344V
TP	0715496663
Address	Balanoda

Gurantor Details

Name	Saman Pabasara
Nic	8415457844V
TP	0714586668
Address	Ellepola, Balngoda

I am *Athula Chandrawansha* agree with following tems and conditions apply with Wisdom Computer Technologies.

1. If installments not pay on time pay with extra rate.
2. Purchase item is Wisdom Computer Technology Property response payment complte.
3. If payments uncompleted it is can return to Wisdom Computer Technology.
4. Gurantor is second partner of this agreement.
5. Not service or repair item under payment compelte.

Item(s) Description

Item Code	Item Name	Warranty	Qty	Price	Sub Total
ITM-00000004	Desktop Tower	10.00	1	60,000.00	60,000.00

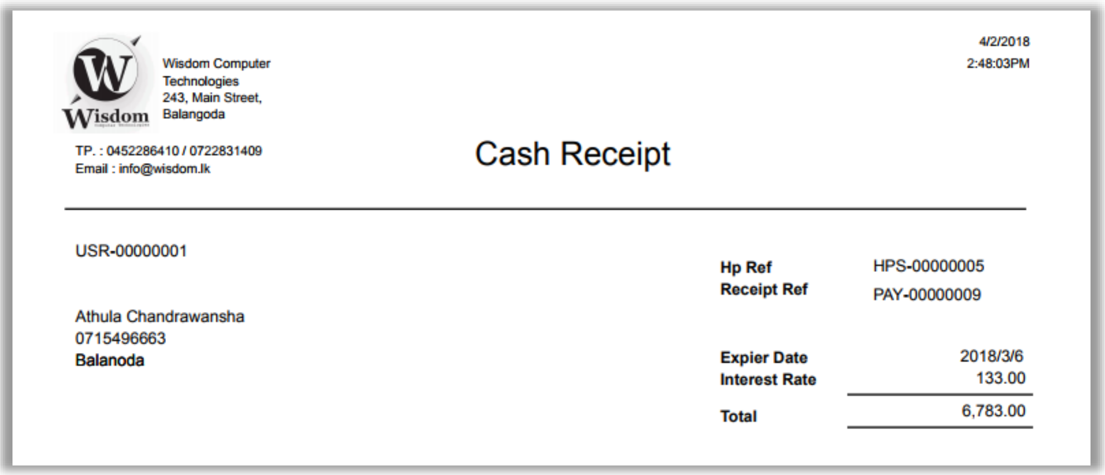
Gross Total	60,000.00
Deduction	100.00
Grand Total	**59,900.00**

Valid Signatures

....	....	....
Authorize Person	Customer	Gurantor
USR-00000001	Athula Chandrawansha	Saman Pabasara

Thank you for your busniess.

Figure D.3 Hire purchase agreement report



Wisdom Computer Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

4/2/2018
2:48:03PM

Cash Receipt

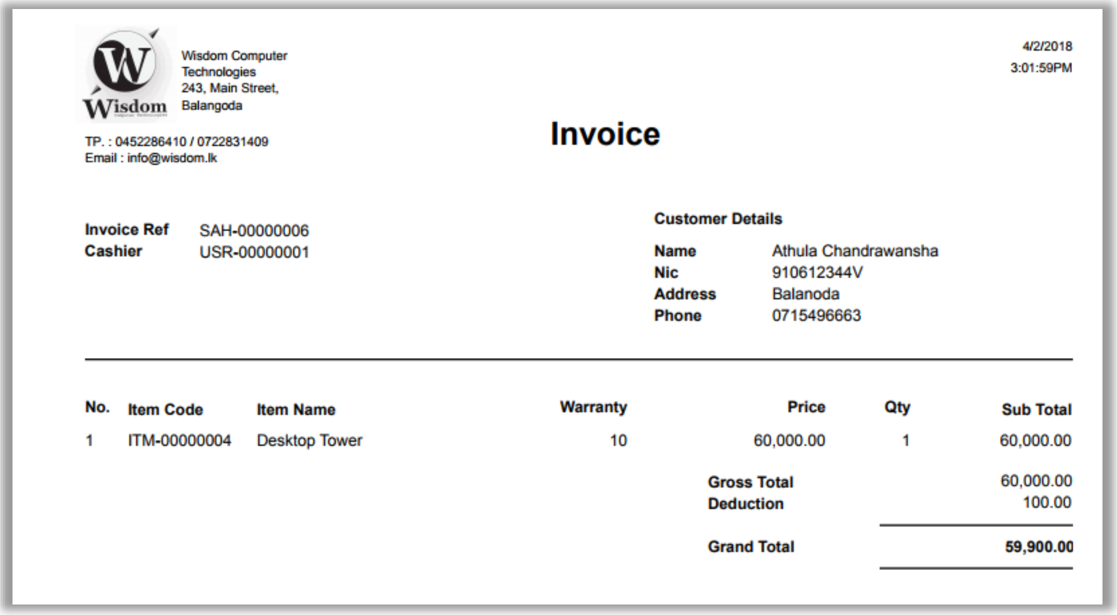
USR-00000001

Athula Chandrawansha
0715496663
Balanoda

Hp Ref	HPS-00000005
Receipt Ref	PAY-00000009
Expier Date	2018/3/6
Interest Rate	133.00
Total	6,783.00

Figure D.4 Cash receipt for hire purchase installment

- **Invoice for Hire Purchase and Direct Sale**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

4/2/2018
3:01:59PM

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

Invoice

Invoice Ref	SAH-00000006
Cashier	USR-00000001

Customer Details

Name	Athula Chandrawansha
Nic	910612344V
Address	Balanoda
Phone	0715496663

No.	Item Code	Item Name	Warranty	Price	Qty	Sub Total
1	ITM-00000004	Desktop Tower	10	60,000.00	1	60,000.00

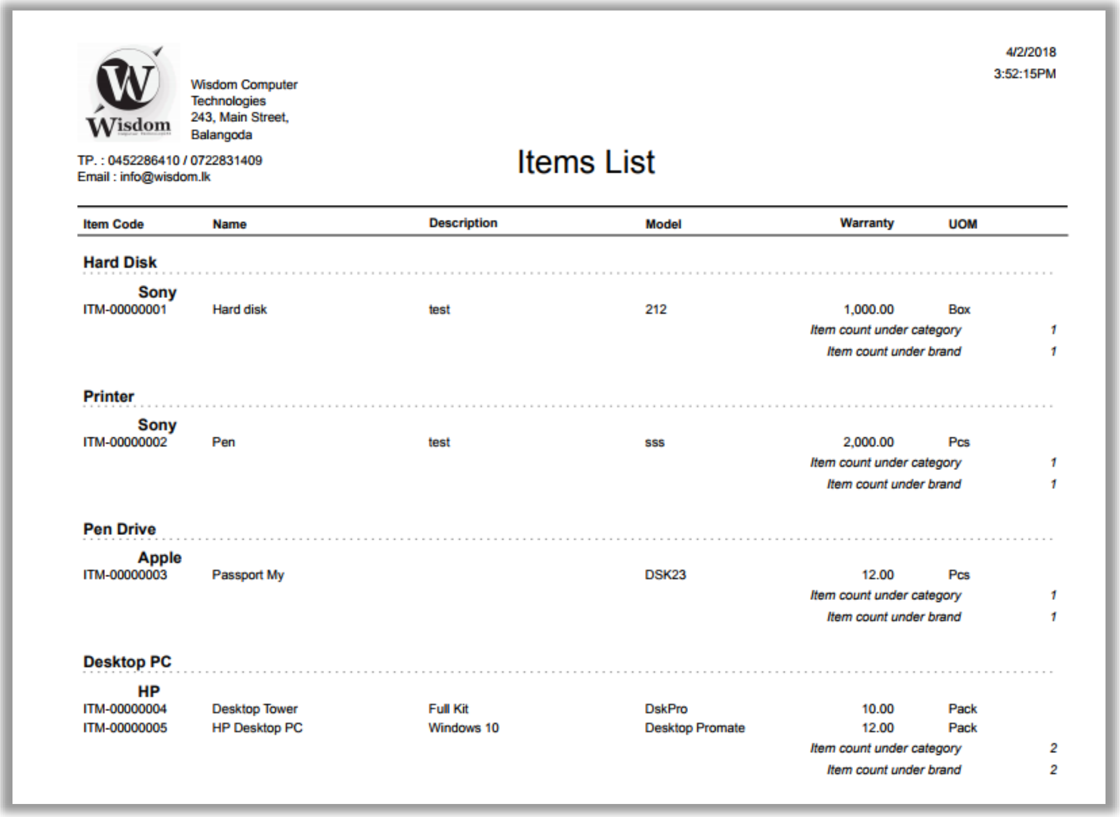
Gross Total	60,000.00
Deduction	100.00
Grand Total	**59,900.00**

Figure D.5 Invoice for hire purchase and direct sale

Master Data Summery Reports

Shown sample of master data summaries of following reports. Following items list report items are categorized and summarized by category and brand

- **List of Items**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

4/2/2018
3:52:15PM

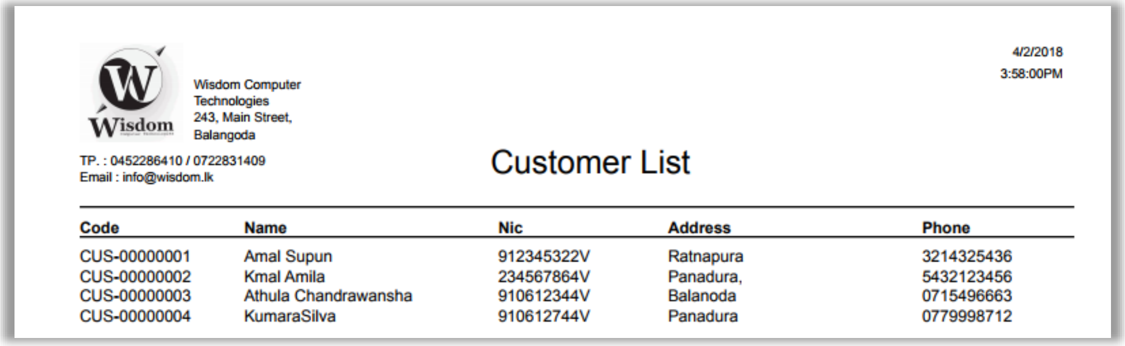
TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

Items List

Item Code	Name	Description	Model	Warranty	UOM	
Hard Disk						
Sony						
ITM-00000001	Hard disk	test	212	1,000.00	Box	
				Item count under category		*1*
				Item count under brand		*1*
Printer						
Sony						
ITM-00000002	Pen	test	sss	2,000.00	Pcs	
				Item count under category		*1*
				Item count under brand		*1*
Pen Drive						
Apple						
ITM-00000003	Passport My		DSK23	12.00	Pcs	
				Item count under category		*1*
				Item count under brand		*1*
Desktop PC						
HP						
ITM-00000004	Desktop Tower	Full Kit	DskPro	10.00	Pack	
ITM-00000005	HP Desktop PC	Windows 10	Desktop Promate	12.00	Pack	
				Item count under category		*2*
				Item count under brand		*2*

Figure D.6 Items list report

- **List of All Customers**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

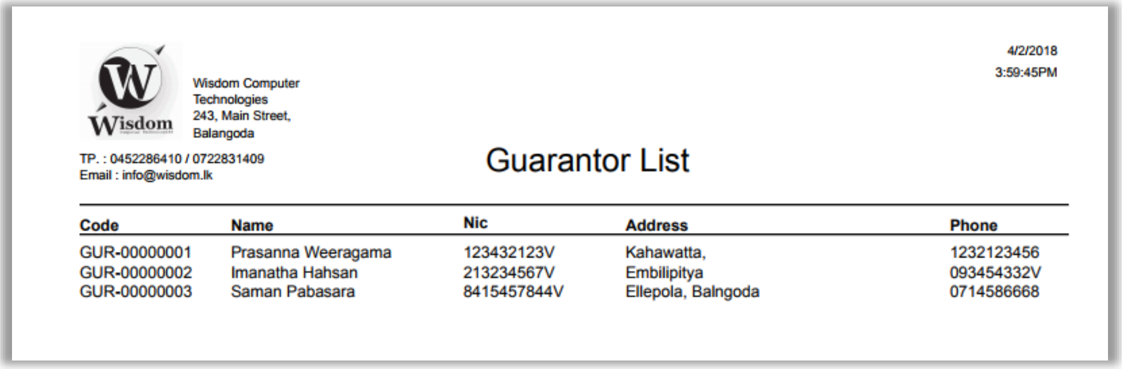
4/2/2018
3:58:00PM

Customer List

Code	Name	Nic	Address	Phone
CUS-00000001	Amal Supun	912345322V	Ratnapura	3214325436
CUS-00000002	Kmal Amila	234567864V	Panadura,	5432123456
CUS-00000003	Athula Chandrawansha	910612344V	Balanoda	0715496663
CUS-00000004	KumaraSilva	910612744V	Panadura	0779998712

Figure D.7 Customers list report

- **List of Guarantors**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

4/2/2018
3:59:45PM

Guarantor List

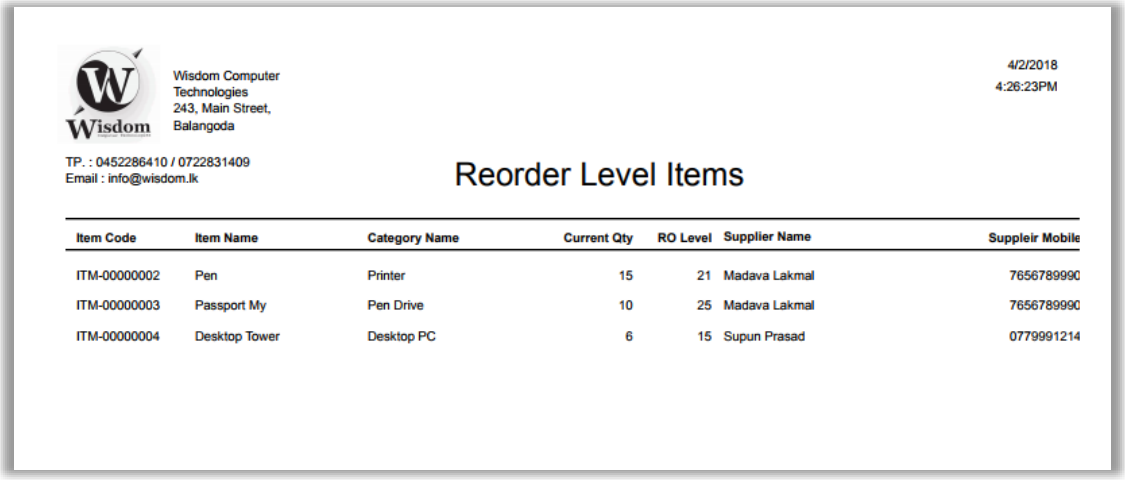
Code	Name	Nic	Address	Phone
GUR-00000001	Prasanna Weeragama	123432123V	Kahawatta,	1232123456
GUR-00000002	Imanatha Hahsan	213234567V	Embilipitya	093454332V
GUR-00000003	Saman Pabasara	8415457844V	Ellepola, Balngoda	0714586668

Figure D.8 Guarantors list

Stock Reports

Following figure shown reports about stock items. The reorder level items report shown reorder level items in the stock according to user set reorder level. Stock movement report represent items are in out in the stock.

- **Reorder Level Items**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

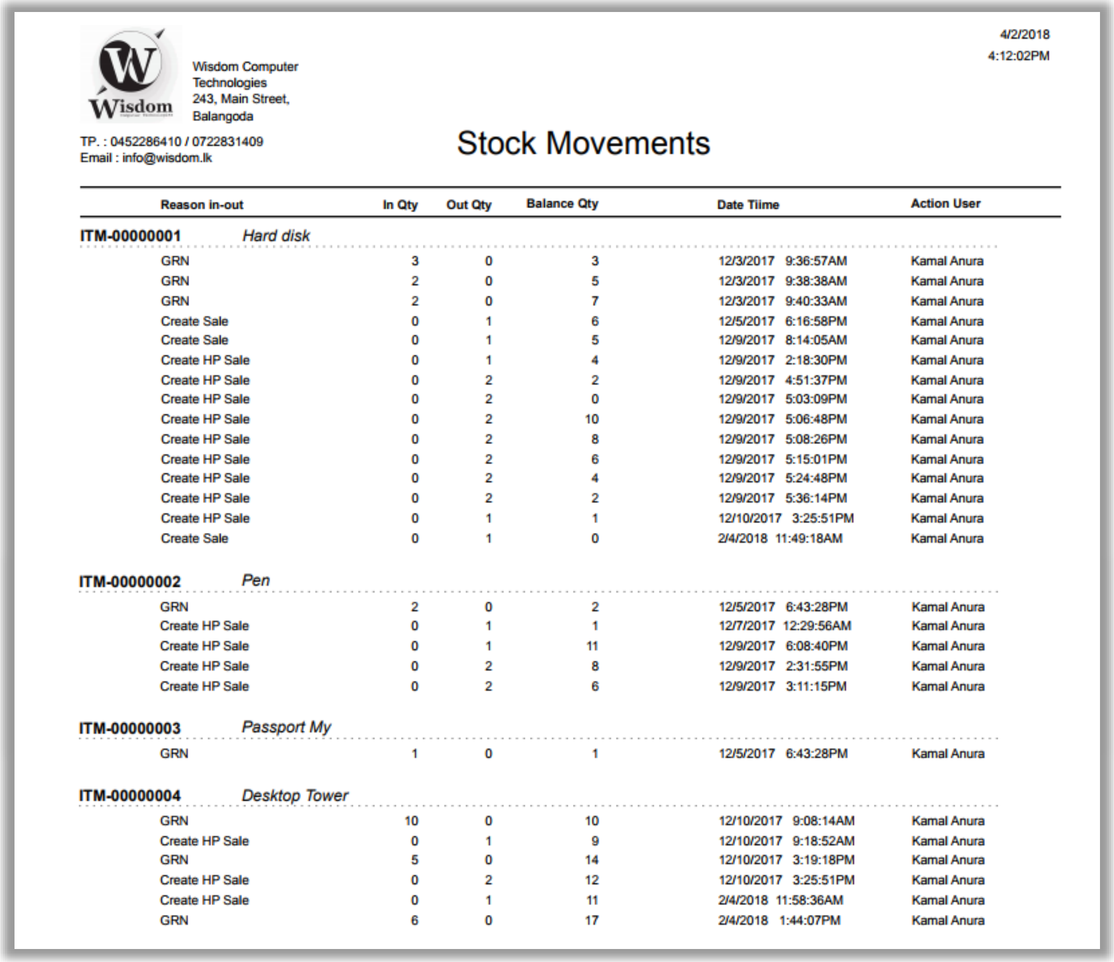
4/2/2018
4:26:23PM

Reorder Level Items

Item Code	Item Name	Category Name	Current Qty	RO Level	Supplier Name	Suppleir Mobile
ITM-00000002	Pen	Printer	15	21	Madava Lakmal	7656789990
ITM-00000003	Passport My	Pen Drive	10	25	Madava Lakmal	7656789990
ITM-00000004	Desktop Tower	Desktop PC	6	15	Supun Prasad	0779991214

Figure D.9 Reorder level items

- **Stock Movement Report**



4/2/2018
4:12:02PM

Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

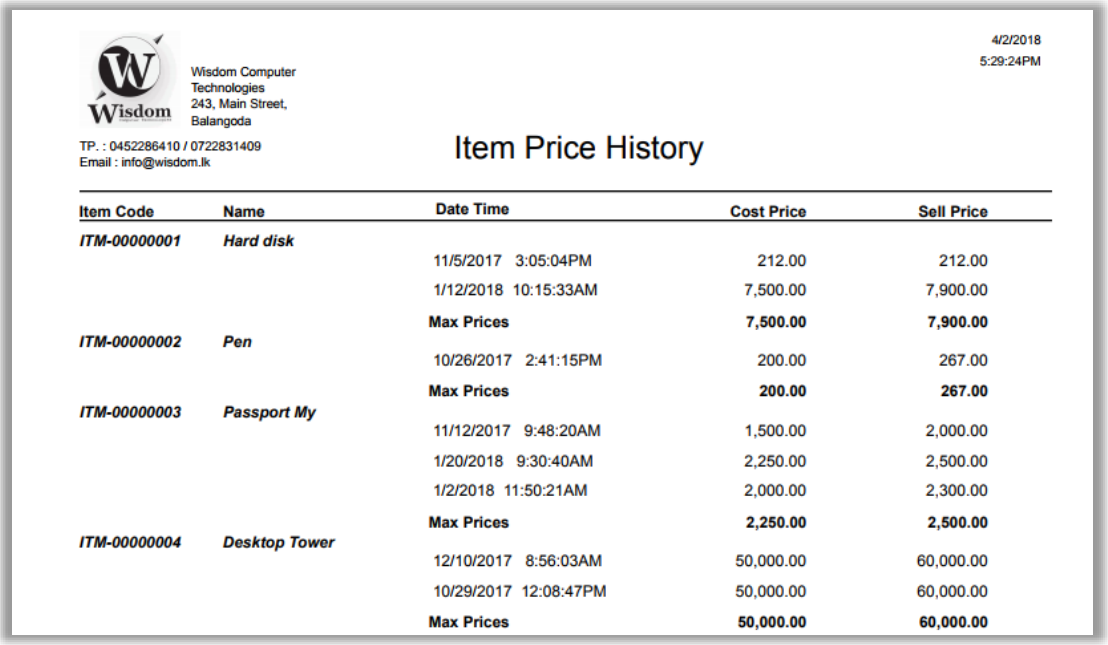
Stock Movements

	Reason in-out	In Qty	Out Qty	Balance Qty	Date Tiime	Action User
ITM-00000001	*Hard disk*					
	GRN	3	0	3	12/3/2017 9:36:57AM	Kamal Anura
	GRN	2	0	5	12/3/2017 9:38:38AM	Kamal Anura
	GRN	2	0	7	12/3/2017 9:40:33AM	Kamal Anura
	Create Sale	0	1	6	12/5/2017 6:16:58PM	Kamal Anura
	Create Sale	0	1	5	12/9/2017 8:14:05AM	Kamal Anura
	Create HP Sale	0	1	4	12/9/2017 2:18:30PM	Kamal Anura
	Create HP Sale	0	2	2	12/9/2017 4:51:37PM	Kamal Anura
	Create HP Sale	0	2	0	12/9/2017 5:03:09PM	Kamal Anura
	Create HP Sale	0	2	10	12/9/2017 5:06:48PM	Kamal Anura
	Create HP Sale	0	2	8	12/9/2017 5:08:26PM	Kamal Anura
	Create HP Sale	0	2	6	12/9/2017 5:15:01PM	Kamal Anura
	Create HP Sale	0	2	4	12/9/2017 5:24:48PM	Kamal Anura
	Create HP Sale	0	2	2	12/9/2017 5:36:14PM	Kamal Anura
	Create HP Sale	0	1	1	12/10/2017 3:25:51PM	Kamal Anura
	Create Sale	0	1	0	2/4/2018 11:49:18AM	Kamal Anura
ITM-00000002	*Pen*					
	GRN	2	0	2	12/5/2017 6:43:28PM	Kamal Anura
	Create HP Sale	0	1	1	12/7/2017 12:29:56AM	Kamal Anura
	Create HP Sale	0	1	11	12/9/2017 6:08:40PM	Kamal Anura
	Create HP Sale	0	2	8	12/9/2017 2:31:55PM	Kamal Anura
	Create HP Sale	0	2	6	12/9/2017 3:11:15PM	Kamal Anura
ITM-00000003	*Passport My*					
	GRN	1	0	1	12/5/2017 6:43:28PM	Kamal Anura
ITM-00000004	*Desktop Tower*					
	GRN	10	0	10	12/10/2017 9:08:14AM	Kamal Anura
	Create HP Sale	0	1	9	12/10/2017 9:18:52AM	Kamal Anura
	GRN	5	0	14	12/10/2017 3:19:18PM	Kamal Anura
	Create HP Sale	0	2	12	12/10/2017 3:25:51PM	Kamal Anura
	Create HP Sale	0	1	11	2/4/2018 11:58:36AM	Kamal Anura
	GRN	6	0	17	2/4/2018 1:44:07PM	Kamal Anura

Figure D.10 Stock movements report

History Reports

- **Item Price History**



4/2/2018
5:29:24PM

Wisdom Computer
Technologies
243, Main Street,
Balangoda

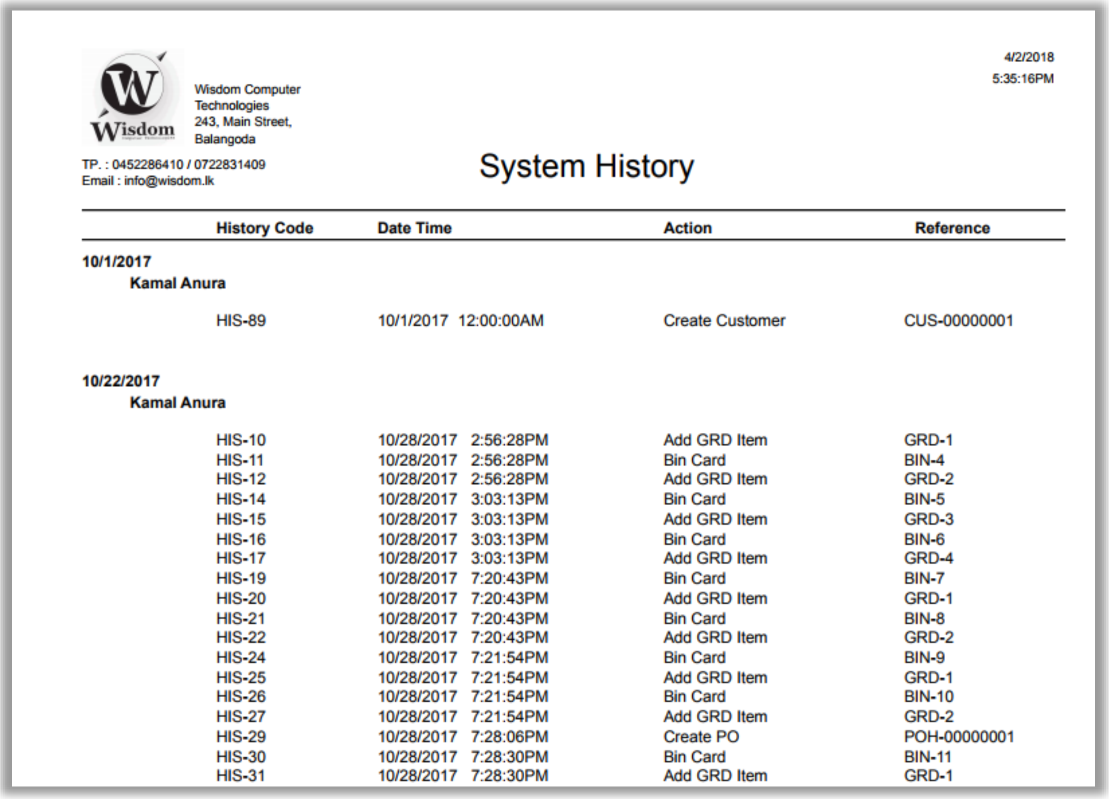
TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

Item Price History

Item Code	Name	Date Time	Cost Price	Sell Price
ITM-00000001	***Hard disk***			
		11/5/2017 3:05:04PM	212.00	212.00
		1/12/2018 10:15:33AM	7,500.00	7,900.00
		Max Prices	**7,500.00**	**7,900.00**
ITM-00000002	***Pen***			
		10/26/2017 2:41:15PM	200.00	267.00
		Max Prices	**200.00**	**267.00**
ITM-00000003	***Passport My***			
		11/12/2017 9:48:20AM	1,500.00	2,000.00
		1/20/2018 9:30:40AM	2,250.00	2,500.00
		1/2/2018 11:50:21AM	2,000.00	2,300.00
		Max Prices	**2,250.00**	**2,500.00**
ITM-00000004	***Desktop Tower***			
		12/10/2017 8:56:03AM	50,000.00	60,000.00
		10/29/2017 12:08:47PM	50,000.00	60,000.00
		Max Prices	**50,000.00**	**60,000.00**

Figure D.11 Items price history

- **System History Report**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

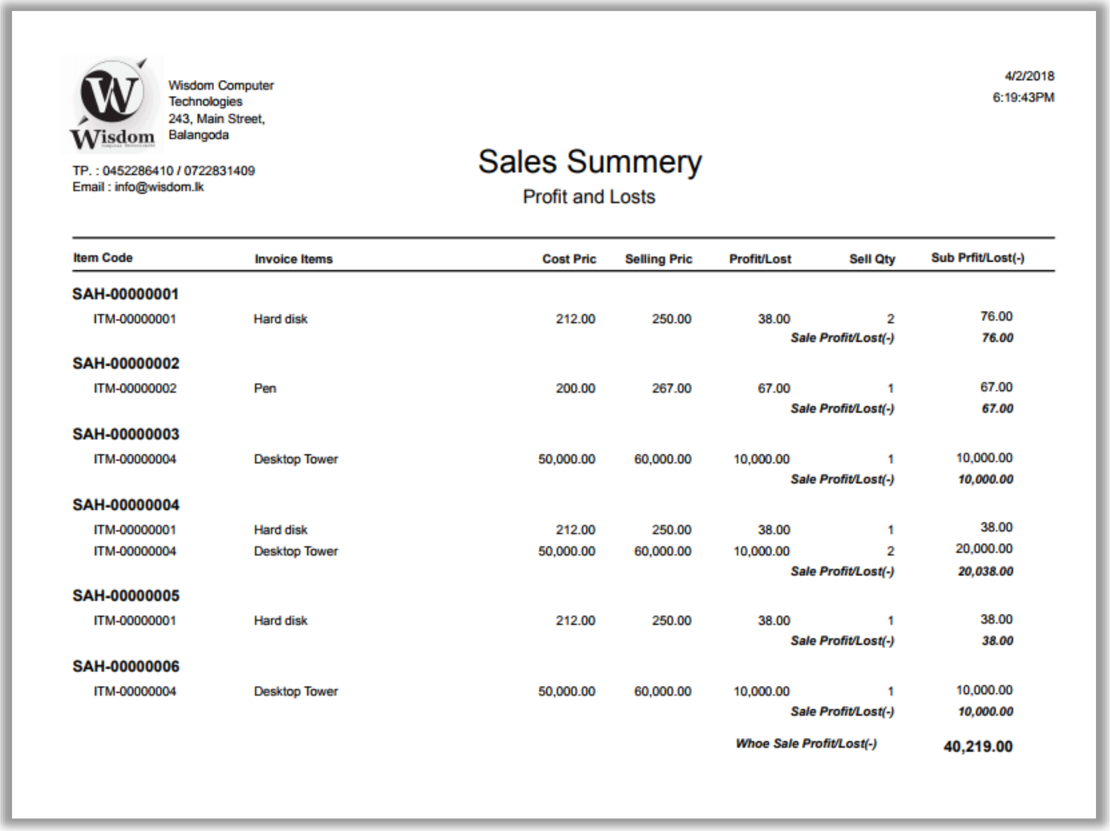
4/2/2018
5:35:16PM

System History

		History Code	Date Time	Action	Reference
10/1/2017					
	Kamal Anura				
		HIS-89	10/1/2017 12:00:00AM	Create Customer	CUS-00000001
10/22/2017					
	Kamal Anura				
		HIS-10	10/28/2017 2:56:28PM	Add GRD Item	GRD-1
		HIS-11	10/28/2017 2:56:28PM	Bin Card	BIN-4
		HIS-12	10/28/2017 2:56:28PM	Add GRD Item	GRD-2
		HIS-14	10/28/2017 3:03:13PM	Bin Card	BIN-5
		HIS-15	10/28/2017 3:03:13PM	Add GRD Item	GRD-3
		HIS-16	10/28/2017 3:03:13PM	Bin Card	BIN-6
		HIS-17	10/28/2017 3:03:13PM	Add GRD Item	GRD-4
		HIS-19	10/28/2017 7:20:43PM	Bin Card	BIN-7
		HIS-20	10/28/2017 7:20:43PM	Add GRD Item	GRD-1
		HIS-21	10/28/2017 7:20:43PM	Bin Card	BIN-8
		HIS-22	10/28/2017 7:20:43PM	Add GRD Item	GRD-2
		HIS-24	10/28/2017 7:21:54PM	Bin Card	BIN-9
		HIS-25	10/28/2017 7:21:54PM	Add GRD Item	GRD-1
		HIS-26	10/28/2017 7:21:54PM	Bin Card	BIN-10
		HIS-27	10/28/2017 7:21:54PM	Add GRD Item	GRD-2
		HIS-29	10/28/2017 7:28:06PM	Create PO	POH-00000001
		HIS-30	10/28/2017 7:28:30PM	Bin Card	BIN-11
		HIS-31	10/28/2017 7:28:30PM	Add GRD Item	GRD-1

Figure D.12 System history report

- **Whole Sales Summery Profit & Lost Report**



Wisdom Computer
Technologies
243, Main Street,
Balangoda

TP. : 0452286410 / 0722831409
Email : info@wisdom.lk

4/2/2018
6:19:43PM

Sales Summery

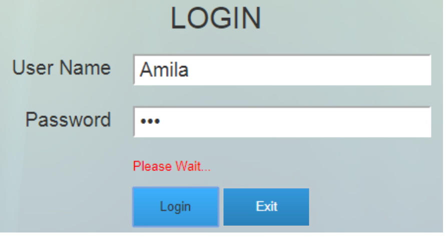
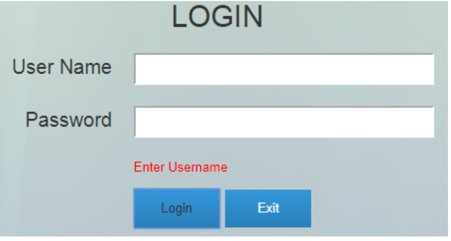
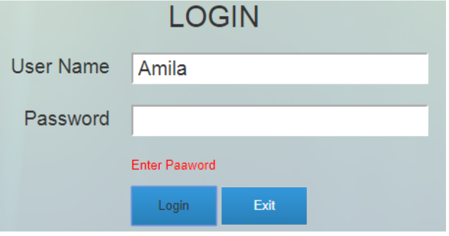
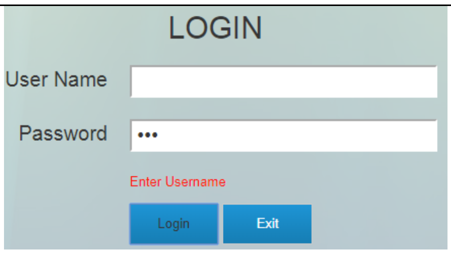
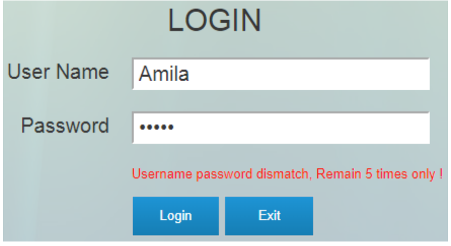
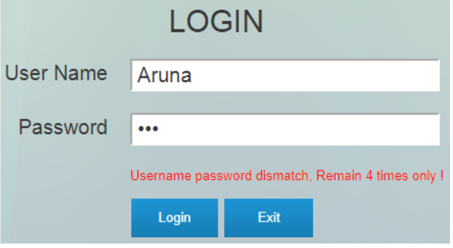
Profit and Losts

Item Code	Invoice Items	Cost Pric	Selling Pric	Profit/Lost	Sell Qty	Sub Prfit/Lost(-)
SAH-00000001						
ITM-00000001	Hard disk	212.00	250.00	38.00	2	76.00
					Sale Profit/Lost(-)	***76.00***
SAH-00000002						
ITM-00000002	Pen	200.00	267.00	67.00	1	67.00
					Sale Profit/Lost(-)	***67.00***
SAH-00000003						
ITM-00000004	Desktop Tower	50,000.00	60,000.00	10,000.00	1	10,000.00
					Sale Profit/Lost(-)	***10,000.00***
SAH-00000004						
ITM-00000001	Hard disk	212.00	250.00	38.00	1	38.00
ITM-00000004	Desktop Tower	50,000.00	60,000.00	10,000.00	2	20,000.00
					Sale Profit/Lost(-)	***20,038.00***
SAH-00000005						
ITM-00000001	Hard disk	212.00	250.00	38.00	1	38.00
					Sale Profit/Lost(-)	***38.00***
SAH-00000006						
ITM-00000004	Desktop Tower	50,000.00	60,000.00	10,000.00	1	10,000.00
					Sale Profit/Lost(-)	***10,000.00***
				Whoe Sale Profit/Lost(-)		**40,219.00**

Figure D.13 Whole sales profit & lost report

Appendix E - Test Result

- **Test case for user authentication - System login**

No	Test Case	Expected Output	Actual output	Status
1	Successful login	Redirecting message for dash board	 LOGIN User Name Amila Password ••• Please Wait... Login Exit	Pass
2	Try to login with empty user name and password	Show error message	 LOGIN User Name Password Enter Username Login Exit	Pass
3	Try to login with empty password	Show error message	 LOGIN User Name Amila Password Enter Paaword Login Exit	Pass
4	Try to login with empty user name	Show error message	 LOGIN User Name Password ••• Enter Username Login Exit	Pass
5	Try to login with correct user name and incorrect password	Show error message	 LOGIN User Name Amila Password ••••• Username password dismatch, Remain 5 times only ! Login Exit	Pass
6	Try to login with incorrect user name and correct password	Show error message	 LOGIN User Name Aruna Password ••• Username password dismatch, Remain 4 times only ! Login Exit	Pass

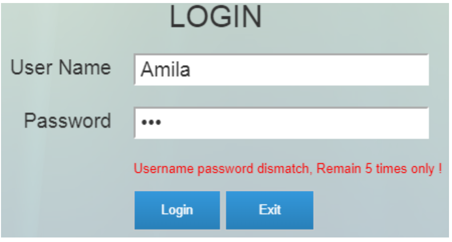
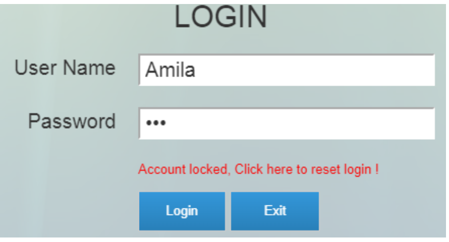
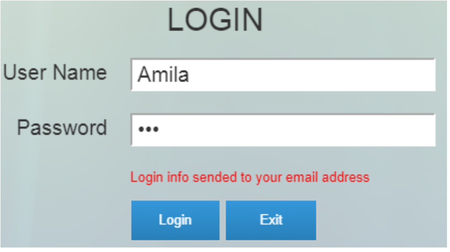
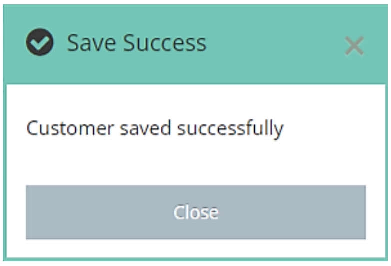
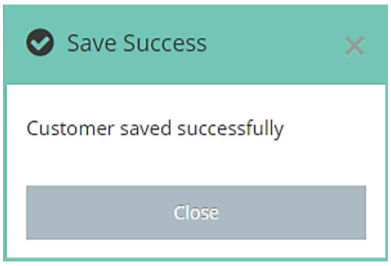
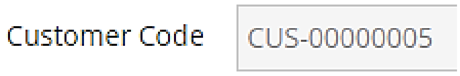
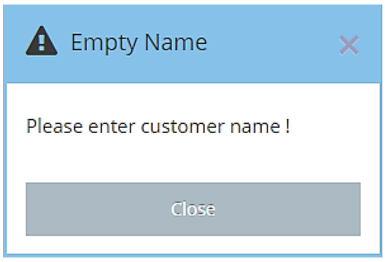
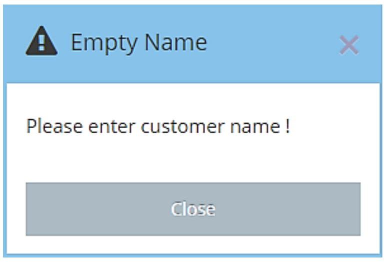
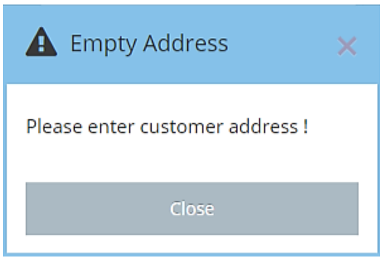
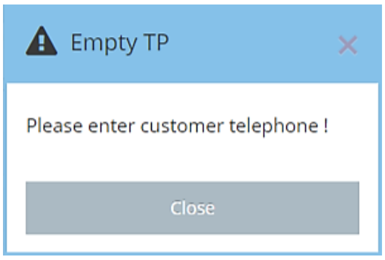
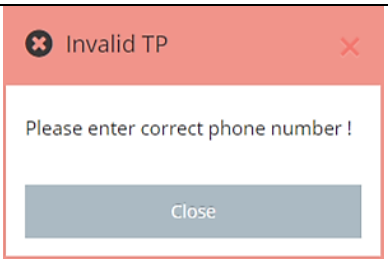
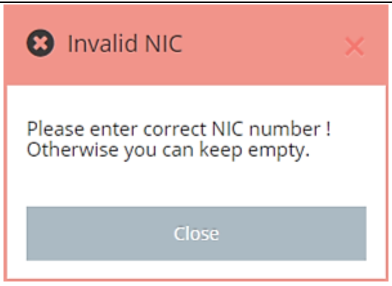
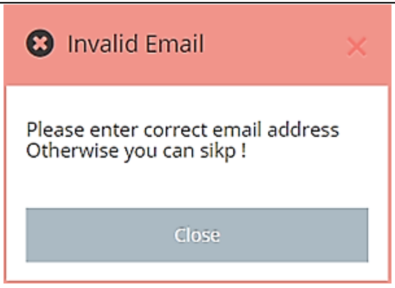
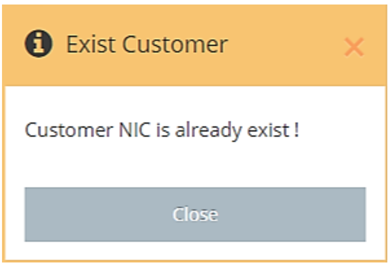
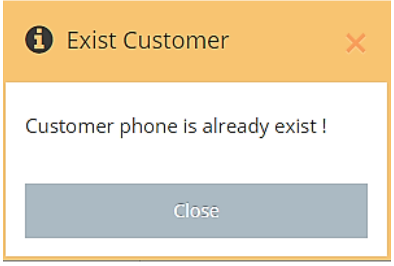
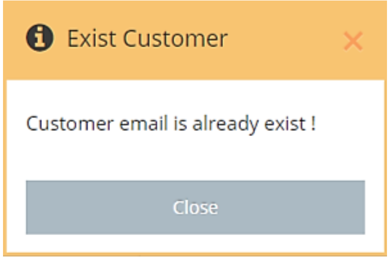
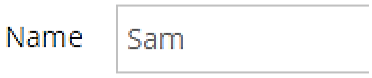
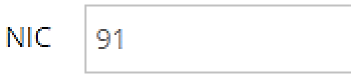
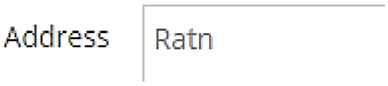
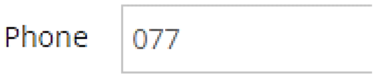
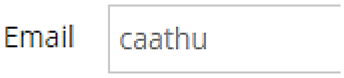
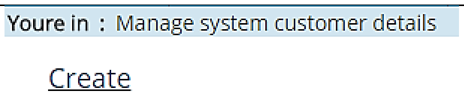
7	Try to login with incorrect username and password	Show error message	 LOGIN User Name: Amila Password: ••• Username password dismatch, Remain 5 times only ! Login Exit	Pass
8	Try to login more than five times with incorrect login information	Show reset login information message	 LOGIN User Name: Amila Password: ••• Account locked, Click here to reset login ! Login Exit	Pass
9	Send login reset information to user email address	Show login message	 LOGIN User Name: Amila Password: ••• Login info sended to your email address Login Exit	Pass

Table E.1 Test case for user authentication

- **Test results for add customer - Customer master (Master module)**

No	Test Case	Expected Output	Actual output	Status
1	Try to save with correct all details	Show success message	 Save Success Customer saved successfully Close	Pass
2	Try to save with correct required details only	Show success message	 Save Success Customer saved successfully Close	Pass
3	Load customer code	Generate customer code by system	 Customer Code: CUS-00000005	Pass

4	Try to save customer keep all fields empty without customer code	Show empty message	 Empty Name Please enter customer name ! 	Pass
5	Try to save customer without name	Show empty message	 Empty Name Please enter customer name ! 	Pass
6	Try to save customer without address	Show empty message	 Empty Address Please enter customer address ! 	Pass
7	Try to save customer without telephone	Show empty message	 Empty TP Please enter customer telephone ! 	Pass
8	Try to save customer with invalid telephone	Show invalid message	 Invalid TP Please enter correct phone number ! 	Pass
9	Try to save customer with invalid NIC number	Show invalid message	 Invalid NIC Please enter correct NIC number ! Otherwise you can keep empty. 	Pass

10	Try to save customer with invalid email address	Show invalid message	 Invalid Email Please enter correct email address Otherwise you can sikp ! 	Pass
11	Try to save customer with exist NIC number	Show already exist message	 Exist Customer Customer NIC is already exist ! 	Pass
12	Try to save customer with exist phone	Show already exist message	 Exist Customer Customer phone is already exist ! 	Pass
13	Try to save customer with exist email address	Show already exist message	 Exist Customer Customer email is already exist ! 	Pass
14	Enter invalid characters to customer name	Auto clear by system	 Name: Sam	Pass
15	Enter invalid characters to customer name	Auto clear by system	 NIC: 91	Pass
16	Enter invalid characters to customer name	Auto clear by system	 Address: Ratn	Pass
17	Enter invalid characters to customer name	Auto clear by system	 Phone: 077	Pass
18	Enter invalid characters to customer name	Auto clear by system	 Email: caathu	Pass
19	Click on back button	Back to main page	 Youre in : Manage system customer details 	Pass

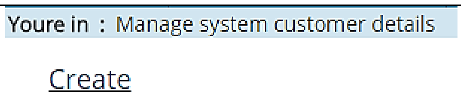
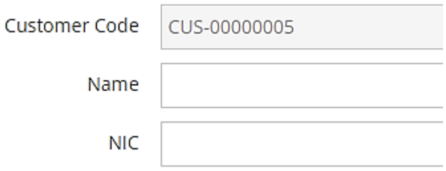
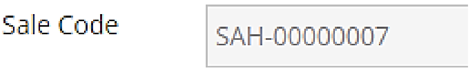
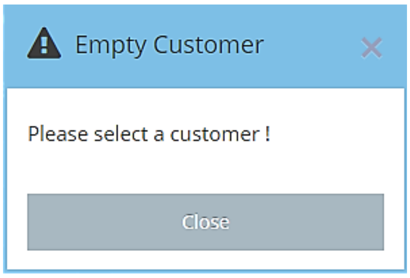
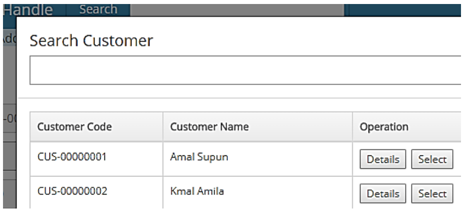
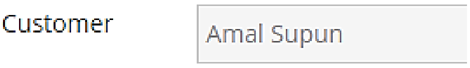
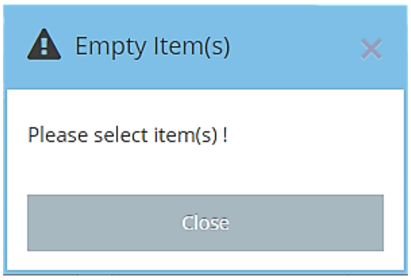
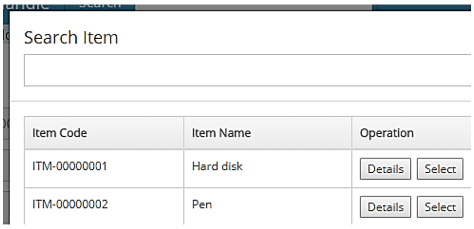
20	Click on back link	Back to main page	 Youre in : Manage system customer details Create	Pass
21	Click on clear button	Clear customer data without customer code	 Customer Code: CUS-00000005 Name: NIC:	Pass

Table E.2 Test case for master module

- **Test case for direct sale - Direct sale module**

No	Test Case	Expected Output	Actual output	Status
1	Generate sale code	Show sale code	 Sale Code: SAH-00000007	Pass
2	Try to create sale without select a customer	Show empty message	 Empty Customer Please select a customer ! Close	Pass
3	Search customer	Search dialog load with customers	 Search Customer Customer Code / Customer Name / Operation CUS-00000001 / Amal Supun / Details Select CUS-00000002 / Kmal Amila / Details Select	Pass
4	Select a customer	Show customer detail	 Customer: Amal Supun	Pass
5	Try to create sale or click on Add button without selecting items	Show empty message	 Empty Item(s) Please select item(s) ! Close	Pass
6	Search item	Search dialog load with items	 Search Item Item Code / Item Name / Operation ITM-00000001 / Hard disk / Details Select ITM-00000002 / Pen / Details Select	Pass

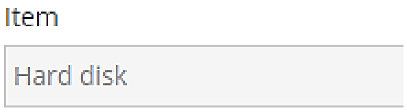
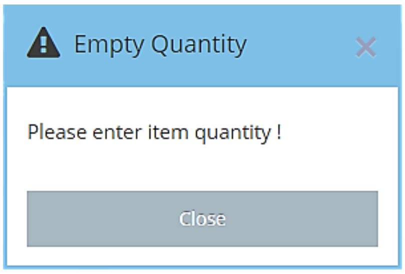
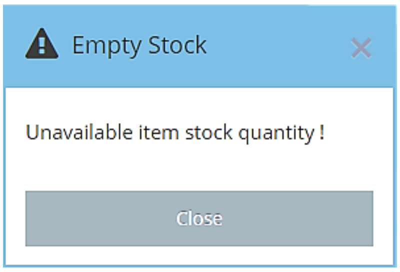
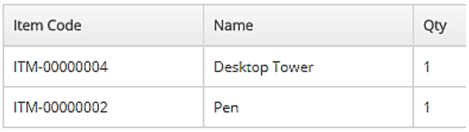
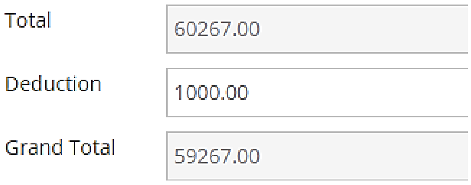
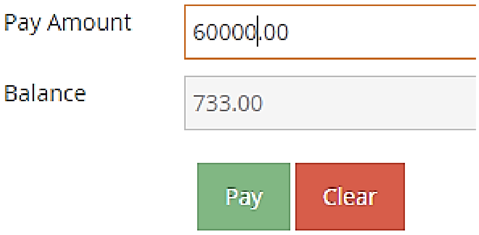
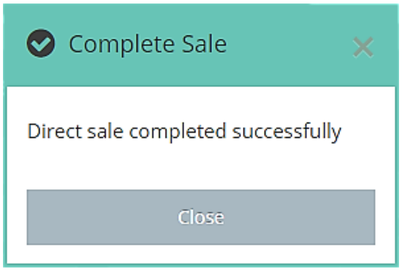
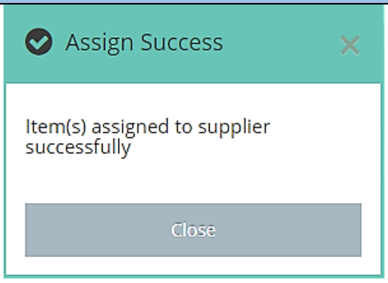
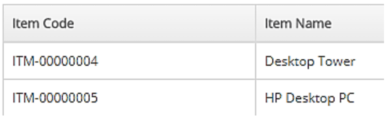
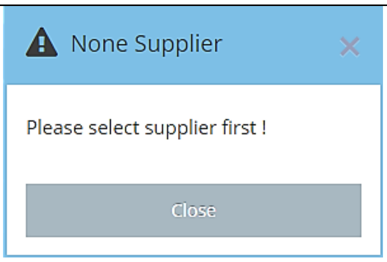
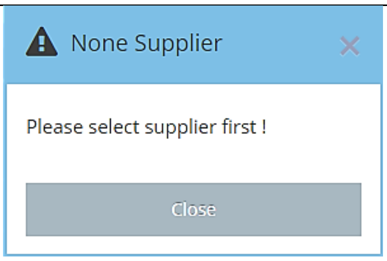
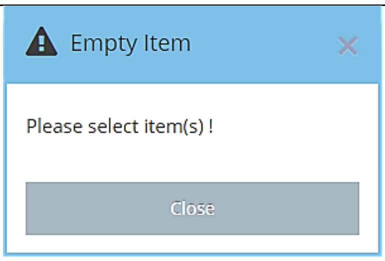
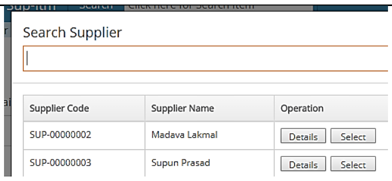
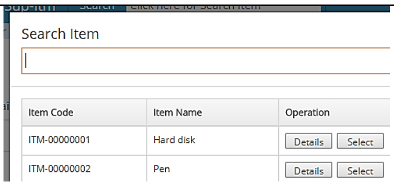
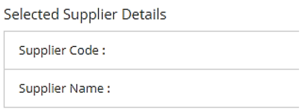
7	Select an item	Show item details	 Item: Hard disk	Pass
8	Try to create sale or click on Add button with empty or zero quantity	Show empty message	 Empty Quantity Please enter item quantity ! 	Pass
9	Try to enter quantity stock empty item	Show empty message	 Empty Stock Unavailable item stock quantity ! 	Pass
10	Add selling items in to list	Show items list	 Item Code | Name | Qty ITM-00000004 | Desktop Tower | 1 ITM-00000002 | Pen | 1	Pass
11	Calculate grand total	Show calculation	 Total: 60267.00 Deduction: 1000.00 Grand Total: 59267.00	Pass
12	Calculate balance and Pay button	Show calculation and button	 Pay Amount: 60000.00 Balance: 733.00 	Pass
13	Create sale with all relevant data	Show complete message	 Complete Sale Direct sale completed successfully 	Pass

Table E.3 Test case for direct sale module

- **Test Results for Assign Items for Suppliers (Master Module)**

No	Test Case	Expected Output	Actual output	Status
1	Assign items for supplier correctly	Show success message	 Assign Success Item(s) assigned to supplier successfully 	Pass
2	Select supplier	Load items already assign to supplier	 Item Code | Item Name ITM-00000004 | Desktop Tower ITM-00000005 | HP Desktop PC	Pass
3	Click on assign without select supplier and items	Show empty message	 None Supplier Please select supplier first ! 	Pass
4	Click on assign without select supplier	Show empty message	 None Supplier Please select supplier first ! 	Pass
5	Click on assign without select item	Show empty message	 Empty Item Please select item(s) ! 	Pass
6	Click on supplier search field	Popup supplier search prompt	 Search Supplier Supplier Code | Supplier Name | Operation SUP-00000002 | Madava Lakmal | Details Select SUP-00000003 | Supun Prasad | Details Select	Pass
7	Click on item search field	Popup item search prompt	 Search Item Item Code | Item Name | Operation ITM-00000001 | Hard disk | Details Select ITM-00000002 | Pen | Details Select	Pass
8	Click on clear button	Clear all form	 Selected Supplier Details Supplier Code : Supplier Name :	Pass

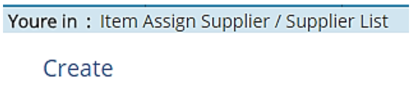
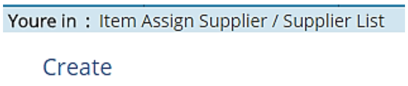
9	Click on back button	Redirect to supplier list	 Youre in : Item Assign Supplier / Supplier List Create	Pass
10	Click on back link	Redirect to supplier list	 Youre in : Item Assign Supplier / Supplier List Create	Pass

Table E.4 Test Results for Assign Items for Suppliers

Appendix F - Code Listing

Anyone can get an idea about system functionalities and system developments referring code fragments. Here represent sample codes of the project. For more can refer source code of the project.

- **Login Page**

Following code segment represent interface (front end) of the login page HTML and CSS mainly.

```
<body>
    @Html.Hidden("RedirectTo", Url.Action("GetMain", "SystemLogin"))
    <div id="main" class="col-md-6" style="background-color:rgba(0, 0, 0, 0.3)">
        <div style="height:auto; background-color:transparent; margin-
             left:0px;padding-left:0px" class="col-md-3">
           <img id="logo" src="~/Images/System/wisLogoP.png"/>
        </div>
        <div style="background-color: rgba(255,255,255, 0.7);padding-
             bottom:10px" class="col-md-9">
            <div style="margin-top:20px" class="row">
                <h2 style="text-align:center;padding-bottom:10px">LOGIN</h2>
                <p id="userName" class="col-md-4">User Name&ensp;</p>
                <input id="txtUn" class="" type="text" />
            </div>
            <div style="margin-top:15px" class="row">
                <p id="password" class="col-md-4">Password&ensp;</p>
                <input id="txtPw" class="" type="password"/>
            </div>
            <div style="margin-top:15px" class="row">
                <div class="col-md-4"></div>
                <div class="col-md-8" style="padding-left:0px;height:15px"><a><p
                id="error"></p></a></div>
            </div>
            <div style="margin-top:15px" class="row">
                <div class="col-md-4"></div>
                <div id="btns" class="col-md-8" style="padding-left:0px;
                width:200px">
                    <button id="btnLogin" class="button btn btn-
                    default">&ensp;Login&ensp;</button>
                    <button id="btnExit" class="button btn btn-
                    default">&emsp;Exit&emsp;</button>
                </div>
            </div>
        </div>
    </div>
</body>
```

Following code segment represent design using CSS in external file.

```
        input[type=text], input[type=password], p{
            font-size:20px;
        }
        input[type=text], input[type=password]{
            padding:0px 5px 0px 5px;
            border-color:white;
            background-color:white
        }
        #main{
        }
        body{
            background-image:url(../../Images/System/logBg.jpg)
        }
        #error{
            font-size:13px;
            color:red;
            cursor:pointer
        }
        #userName, #password{
        }
        #logo {
            display: block;
            margin-left: auto;
            margin-right: auto;
            height: auto;
        }
        .btn {
            background: #3498db;
            background-image: -webkit-linear-gradient(top, #3498db, #2980b9);
            background-image: -moz-linear-gradient(top, #3498db, #2980b9);
            background-image: -ms-linear-gradient(top, #3498db, #2980b9);
            background-image: -o-linear-gradient(top, #3498db, #2980b9);
            background-image: linear-gradient(to bottom, #3498db, #2980b9);
            -webkit-border-radius: 0;
            -moz-border-radius: 0;
            border-radius: 0px;
            font-family: Arial;
            color: #ffffff;
            font-size: 13px;
            padding: 10px 20px 10px 20px;
            text-decoration: none;
        }
        .btn:hover {
            background: #3cb0fd;
            background-image: -webkit-linear-gradient(top, #3cb0fd, #3498db);
            background-image: -moz-linear-gradient(top, #3cb0fd, #3498db);
            background-image: -ms-linear-gradient(top, #3cb0fd, #3498db);
            background-image: -o-linear-gradient(top, #3cb0fd, #3498db);
            background-image: linear-gradient(to bottom, #3cb0fd, #3498db);
            text-decoration: none;
        }
```

Following code segment represent interface responsive using JS, CSS and JQuery external file.

```
        function Responsive() {
            var height = $(window).innerHeight() - 20;
            var width = $(window).innerWidth() - 20;
            $('#main').width(width / 2);

            $('#main').css('margin-left', (width / 4));
            $('#main').css('margin-right', (width / 4));
            $('#main').css('margin-top', (height / 4));

            if (width < 964 && width> 364) {
                $('#userName').css('text-align', 'center');
                $('#password').css('text-align', 'center');
                $('#error').css('text-align', 'center');
                $('#logo').css('width', '150px');
                $('#logo').css('margin', 'auto');
                $('#logo').css('padding-top', '0px');

                $('#main').css('margin-top', (height / 8));

                $('input[type=text]').css('width', '100%');
                $('input[type=password]').css('width', '100%');

                $('#btns').css('margin-left', 'auto');
                $('#btns').css('margin-right', 'auto');

            } else if (width < 365) {

                $('#main').css('margin-left', '0px');
                $('#main').css('margin-right', '0px');
                $('#main').css('width', '100%');


                $('#logo').css('width', '150px');
                $('#logo').css('margin', 'auto');
                $('#logo').css('padding-top', '0px');

                $('#main').css('margin-top', (height / 8));

                $('input[type=text]').css('width', '100%');
                $('input[type=password]').css('width', '100%');

                $('#btns').css('margin-left', 'auto');
                $('#btns').css('margin-right', 'auto');
            } else {
                $('#userName').css('text-align', 'right');
                $('#password').css('text-align', 'right');

                $('#logo').css('width', '150px');
                $('#logo').css('padding-top', '80px');

                $('input[type=text]').css('width', '60%');
                $('input[type=password]').css('width', '60%');

                $('#error').css('text-align', 'left');
            }
        }

        $(document).ready(function () {
            Responsive();
            $(window).on('resize', function () {
                Responsive();
                if (un=="") {
                    $('#error').text('Enter Username');
                } else if(pw=="") {
                    $('#error').text('Enter Paaword');
                } else {
                    LoginCheck();
                }
            });
        });
```

- **Main Page**

Libraries used for Main dashboard

```
<!-- bootstrap & fontawesome -->
<link href="~/Bootstrap/bootstrap.min.css" rel="stylesheet" type="text/css" />
<link href="~/Design/FontAwaysome/4.5.0/css/font-awesome.min.css"
rel="stylesheet" />
<!-- text fonts -->
<link href="~/Design/MainWindow/fonts.googleapis.com.css" rel="stylesheet" />
<!-- ace styles -->
<link href="~/Design/Ace/ace.min.css" rel="stylesheet" type="text/css"
class="ace-main-stylesheet" id="main-ace-style">
<link href="~/Design/Ace/ace-skins.min.css" rel="stylesheet" type="text/css" />
<!-- basic scripts -->
<script src="~/Library/Jquery/jquery-2.1.4.min.js"
type="text/javascript"></script>
<script src="~/Bootstrap/bootstrap.min.js" type="text/javascript"></script>
<!-- ace scripts -->
<script src="~/Library/Ace/ace.min.js" type="text/javascript"></script>
<script src="~/Library/MainWindow/main.js" type="text/javascript"></script>
<script src="~/Scripts/jquery-1.10.2.js"></script>

<script src="~/Library/Jquery/jquery-2.1.4.min.js"></script>

@Scripts.Render("~/bundles/modernizr")
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
```

Login code segment in main dashboard

```
@{
    var sesLog = (Session["log"] != null ? Session["log"].ToString() : "Please
                  Login");
}
$(document).ready(function () {
    //check if user is logged
    $('#logOut').on('click', function () {
        @Session["log"] = null;
        var url = $("#RedirectTo").val();
        location.href = url;
    });
});

@*link to login page*@

@Html.Hidden("RedirectTo", Url.Action("Index", "SystemLogin"))

<ul class="nav navbar-nav navbar-right">
    <li><a style="font-size:15px" href="#">Hi @sesLog</a></li>
    <li><a style="font-size:15px" id="logOut" href="">Log Out</a></li>
</ul>

<a style="cursor:pointer"
   onclick="@("window.location.href='"+@Url.Action(null,"SystemLogin",
   "GetMain")+"'");">
   <i class="menu-icon fa fa-home"></i>
   <span class="menu-text">Main Window</span>
</a>
```

Sample code of main menu - Horizontal

```
<div class="main-container ace-save-state" id="main-container" >
    <div style="background-color: #aed6f1" id="sidebar" class="sidebar
               responsive ace-save-state">
        <ul class="nav nav-list">
            <li class="active">
                <a style="cursor:pointer"
                   onclick="@("window.location.href=
                           '"+@Url.Action(null,"SystemLogin", "GetMain")+"'");">
                    <i class="menu-icon fa fa-home"></i>
                    <span class="menu-text">
                        Main Window
                    </span>
                </a>
                <b class="arrow"></b>
            </li>
            <li class="">
                <a href="#" class="dropdown-toggle">
                    <i class="menu-icon fa fa-book"></i>
                    <span class="menu-text">
                        Master Data
                    </span>
                    <b class="arrow fa fa-angle-down"></b>
                </a>
                <b class="arrow"></b>

                <ul class="submenu">
                    <li class="">
                        @Html.ActionLink("Brands", "Index", "BrandMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("Categories", "Index",
                                         "CategoryMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("Companies", "Index", "CompanyMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("Customers", "Index",
                                        "CustomerMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("Guarantors", "Index",
                                         "GuarantorMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("Suppliers", "Index",
                                         "SupplierMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
                        @Html.ActionLink("UnitOfMeasures", "Index",
                                         "UomMasters")
                        <b class="arrow"></b>
                    </li>
                    <li class="">
```

- **Payment Create Page**

Sample code for interface design

```
<div class="row">
    <div class="col-md-8">
        <h4 class="col-md-3">@Html.ActionLink("Back", "#", "#")</h4>
        <h4 class="col-md-9">Make New Payment</h4>
    </div>
    <input id="txtPaidInstall" class="col-md-8" type="hidden" />
    <input id="txtAllInstall" class="col-md-8" type="hidden" />
    <input id="txtPayCode" class="col-md-8" type="hidden" />
    <input id="txtInsCode" class="col-md-8" type="hidden" />
</div>
<div class="row" style="height:20px"></div>

<div>
    <div class="col-md-4">
        <!--Search-->
        <div class="row">
            <label class="col-md-4">Search Options</label>
            <label><input id="rbtnActive" name="hpAcc" type="radio"
                          value="false" checked />Activate &emsp;</label>
            <label><input id="rbtnClosed" name="hpAcc" type="radio" value="true"
                    />Closed &emsp;</label>
        </div>
        <div class="row">
            <label class="col-md-4">HP Code</label>
            <input id="txtHpCode" class="col-md-8" type="text" />
        </div>
        <!--Details-->
        <div class="row" style="height:1px;background-color:transparent"></div>
        <h6>Sale Details</h6>
        <div class="row" style="height:1px;background-color:whitesmoke"></div>
        <div class="row">
             <label class="col-md-6"><b>Sale Code</b></label>
             <label id="lblSaleCode" class="col-md-6 values">Values</label>
        </div>
        <div class="row">
             <label class="col-md-6"><b>HP Code</b></label>
             <label id="lblHpCode" class="col-md-6 values">Values</label>
        </div>
        <div class="row">
             <label class="col-md-6"><b>Total</b></label>
             <label id="lblTotal" class="col-md-6 values">Values</label></div>
        <div class="row">
        <label class="col-md-6"><b>Down Payment</b></label>
        <label id="lblDownPay" class="col-md-6 values">Values</label></div>
        <div class="row">
             <label class="col-md-6"><b>Interest Rate %</b></label>
             <label id="lblRate" class="col-md-6 values">Values</label></div>
        <div class="row">
             <label class="col-md-6"><b>Current Installment</b></label>
             <label id="lblInstall" class="col-md-6 values">Values</label>
        </div>
        <div class="row">
             <label class="col-md-6"><b>Remian Balance</b></label>
             <label id="lblBalance" class="col-md-6 values">Values</label>
        </div>
        <div class="row">
             <label class="col-md-6"><b>Fines Payment</b></label>
             <label id="lblFinesPay" class="col-md-6 values">Values</label>
        </div>
        <div class="row">
             <label class="col-md-6"><b>Current Payment</b></label>
             <label id="lblCurrentPay" class="col-md-6 values">Values</label>
        </div>
```

Sample validations

```
function IsValid() {
    if ($('#txtHpCode').val()=='') {
        MessageShow('Empty Hp', 'Please select HP !');
        return false;
    } else if ($('#numPayAmount').val() == '' || $('#numPayAmount').val() <= 0)
{
        MessageShow('Empty PayAmount', 'Please enter pay amount HP !');
        return false;
    } else if ($('#numBalance').val() == '' || $('#numBalance').val() < 0) {
        MessageShow('Invalid PayAmount', 'Pay amount must greater than or equal
current payment !');
        return false;
    }else {
        return true;
    }
}
```

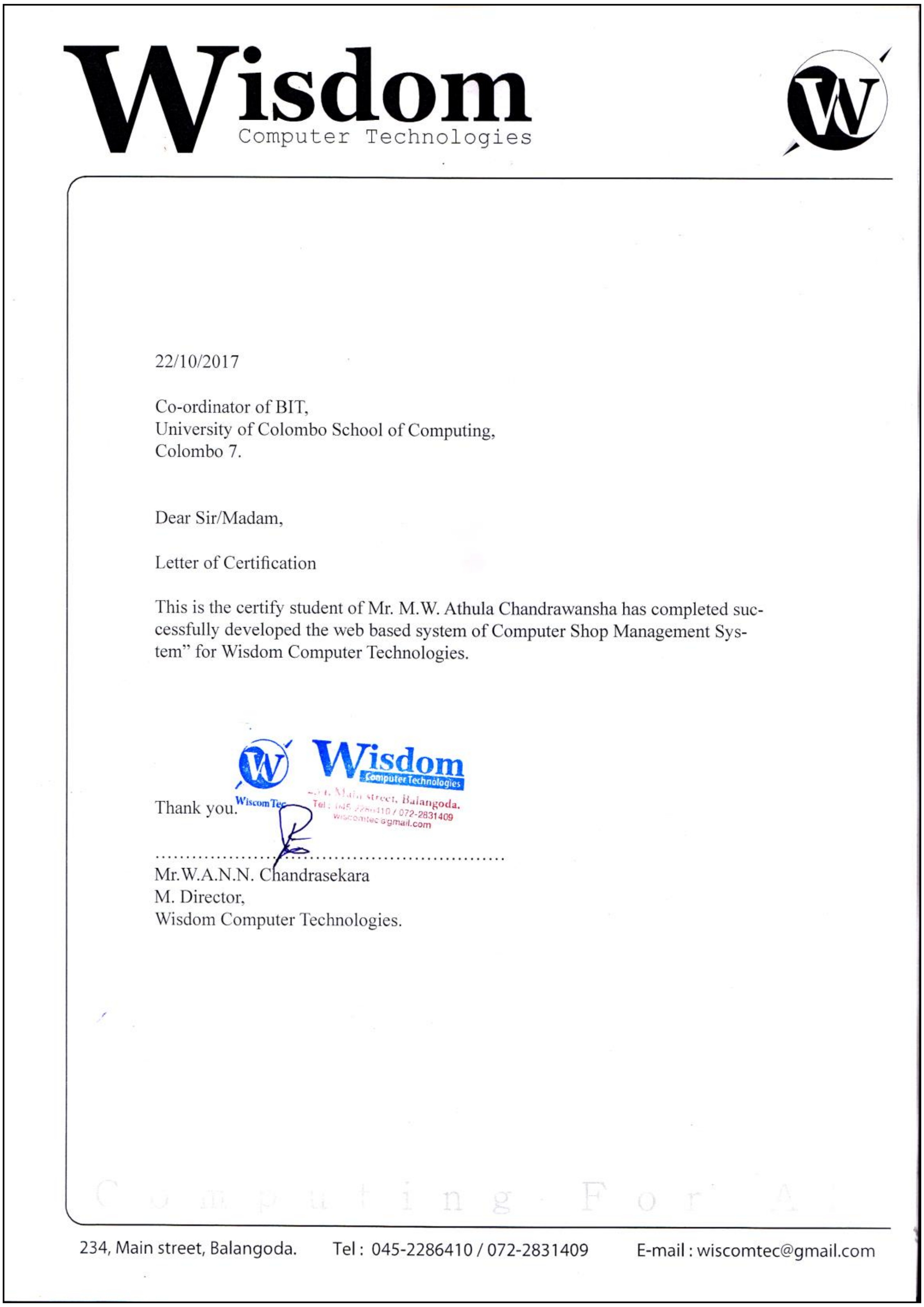
Sample code functions - Font end

```
//search
function SearchHp() {
    var cond = $('#mmSearch').val();
    var payState = $('input[name="hpAcc"]:checked').val();
    $('#tblDatam tbody tr').remove();
    $.ajax({
        url: '@Url.Action("Search", "HigherPurchases")',
        data: {
            'condition': cond,
            'payState': payState
        },
        type: "post",
        cache: false,
        success: function (data) {
            var cou = 0, item = '';
            $.each(data, function (i, item) {
                var rows = "<tr>"
                                + "<td>" + item.hpsCode + "</td>"
                                + "<td>" + item.cusName + "</td>"
                                + "<td>"
                                        + "<button class='d' value='" +
item.hpsCode + "|" + item.cusName + "'>Details</button>&ensp;"
                                        + "<button class='c' value='" +
item.hpsCode + "|" + item.cusName + "'>Select</button>"
                                + "</td>"
                         + "</tr>";
                $('#tblDatam').append(rows);
                $('#resCount').html((cou += 1) + ' Row(s) found');
            });
        },
        complete: function () {

        },
        error: function () {
            alert('Error Load Data in Search hp requst !');
        }
    });
}

//save payemnt
function SavePayment() {
    var payCode = $('#txtPayCode').val();
    var insCode = $('#txtInsCode').val();
    var fines = parseFloat($('#lblFinesPay').text());
    var hpsCode = $('#lblFinesPay').text();

    var payStateHp = false;
    var paidInstall = parseInt($('#txtPaidInstall').val());
    var totalInstall = parseInt($('#txtAllInstall').val());
    if ((paidInstall + 1) == totalInstall) {
        payStateHp = true;
    }
    var sahCode = $('#lblSaleCode').text();

    $.ajax({
        url: '@Url.Action("Insert", "Payments")',
        data: {
            'pPayCode': payCode,
            'pInsCode': insCode,
            'pFines': fines,
            'pHpsCode': hpsCode,
            'pHpsPayState': payStateHp,
            'pSahCode': sahCode
        },
        type: "post",
        cache: false,
        success: function (res) {
            alert('Complete payment');
            //print receipt
            PrintReceipt(payCode);
            if (isFinalPayment) {
                PrintInvoice(sahCode);
            }
        },
        complete: function () {
            ClearAll();
        },
        error: function () {
            alert('Error Load Data in saved requst !');
        }
    });
}
```

Sample code functions - Back end C#

```
//get table row count
private int GetRowCount()
{
    var blogs = db.Payment.SqlQuery("SELECT * FROM tblPayments").ToList();
    return blogs.Count;
}
//generate id
public string GnerateCode()
{
    int max = 0;
    if (GetRowCount() > 0)
    {
        foreach (tblPayment item in db.Payment.ToList())
        {
            if (max < Convert.ToInt32(item.payCode.Substring(4, 8)))
            {
                max = Convert.ToInt32(item.payCode.Substring(4, 8));
            }
        }
        return "PAY-" + (max + 1).ToString("00000000");
    }
    else
    {
        return "PAY-00000001";
    }
}

//insert payment
public bool Insert(string pPayCode, string pInsCode, double pFines, string
pHpsCode, bool pHpsPayState, string pSahCode)
{
    //insert payment
    tblPayment objPay = new tblPayment
    {
        payCode = pPayCode,
        payFines = pFines,
        payInsCode = pInsCode,
        payStatus = true
    };
    db.Payment.Add(objPay);
    db.SaveChanges();
    //update installment pay state
    var insObj = db.Installment.Where(x => x.insCode ==
pInsCode).FirstOrDefault();
    insObj.insPayState = true;
    db.Entry(insObj).State = EntityState.Modified;
    db.SaveChanges();
    //insert history of payment
    new HistoriesController().Insert(objPay.payCode, Global.logedUserCode, "Done
Paymennt");
    //update hp sale state
    if (pHpsPayState)
    {
        var objSah = db.SaleHeader.Where(x => x.sahCode ==
pSahCode).FirstOrDefault();
        objSah.sahPayState = true;
        db.SaveChanges();
        //insert history of sah complete pay for hp
        new HistoriesController().Insert(objSah.sahCode, Global.logedUserCode,
"Complete HP Payments");
    }
    return true;
}
```

Appendix G - Client Certificate



Wisdom
Computer Technologies

22/10/2017

Co-ordinator of BIT,
University of Colombo School of Computing,
Colombo 7.

Dear Sir/Madam,

Letter of Certification

This is the certify student of Mr. M.W. Athula Chandrawansha has completed successfully developed the web based system of Computer Shop Management System" for Wisdom Computer Technologies.

Wisdom
Computer Technologies
WiscomTec
234, Main street, Balangoda.
Tel : 045-2286410 / 072-2831409
wiscomtec@gmail.com

Thank you.

....
Mr.W.A.N.N. Chandrasekara
M. Director,
Wisdom Computer Technologies.

Computing For All

234, Main street, Balangoda. Tel : 045-2286410 / 072-2831409 E-mail : wiscomtec@gmail.com

Figure G.1 Client certificate

Glossary

ER Diagram Entity Relationship diagram show all entities with relations between of them.

DBMS Database Management System is software create and manage databases.

UI User interface.

SQL Structured Query Language is data base programming language.

IDE Integrated Develop Environment.

IIS Internet Information Services is an extensible web server.

UML Unified Modeling Language is visualize the design of system using diagrams.

OS Operating System is a system software.

OOP Object Oriented Programming is a programming paradigm based on concept of objects.

MVC Model View Controller is software architectural pattern.

HTML Hyper Text Markup Language for creating web pages and applications.

CSS Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.

.NET FK Dot NET Framework is a software framework developed by Microsoft.

ASP Active Server Pages is Microsoft's first server-side script engine.

MSSQL Microsoft SQL Server is a relational database management system developed by Microsoft.

SDLC Systems Development Life Cycle is a term used in systems engineering.

Index