



SALES AND INVENTORY MANAGEMENT SYSTEM for IMPERIAL AUTO CARE

Name: PBCI DE SILVA

BIT Registration No: R080878

Index No: 0808784

Name of the Supervisor: Mr. NKMN Amarasooriya

2017

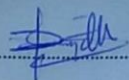


**This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology (external) of the
University of Colombo School of Computing**

Declaration

Declaration

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

Signature of Candidate : 

Date: 05.11.2017

Name of Candidate : PBCI De Silva

Countersigned by:

Signature of Supervisor : 

Date: 5/11/2017

Name of Supervisor : Mr Milan Amarasooriya

Abstract

Imperial Auto Care is a well-known service station in Dehiwala. What they do for business is undertake servicing and detailing vehicles and sell other vehicle related goods such as lubricants, filters, wipers, air fresheners, etc. This process needs various activities to be carried out to fulfill all the requirements for the organization. Currently all the activities are done by manual book keeping. This process requires more man power and also highly inefficient. So the client required a fully functional Sales and Inventory Management System for their Service Station. They were proposed few systems already in the market but the process of those systems were bit complicated and detailed, so those were too much for the organization. They requires a solution with simple steps and suffix to carry out all the areas required. Their main focus was on the inventory management, invoicing and reporting. But also required to maintain the security as well. Therefore this project endeavors to create an application to produce accurate and efficient output of the business to the management by reducing the paperwork, decrease number of human errors and maintain the standards, quality.

Iterative waterfall method is used as the software development life cycle. Coding was handled through an Object-oriented approach. Above mentioned methodologies made project work load light and provided the ease of developing. The system is evaluating by several people regarding user levels of the developed system. Results of the evaluation helped for further maintenance of the product.

System is using Object Oriented techniques. Unified Modeling Language was used in the analysis and design stages of the project. Microsoft Visual Studio 2017 was used to implement the system in C#.NET language with the help of Microsoft SQL Server 2014 in the Windows environment.

Acknowledgements

I would like to gratefully acknowledge people who have been journeyed with me in past few months as I have worked on this Dissertation. Above all, I would like to be thankful to External Degrees Centre University of Colombo School of Computing with including all related parties for giving me such valuable opportunity to be a part of this well recognized degree program. Special thanks for the Project Examination Board for the most precious guidance and support throughout the period.

I am grateful to my supervisor, Mr. NKMN Amarasooriya who is a fellow Software developer at Kingslake (pvt) Ltd, whose expertise, understanding, generous guidance and support made it all possible for me to work on such project that was of great interest to me. It was a pleasure to working with him.

Mr. D.D.P. Lal, the Assistant Accountant of Imperial Auto Care owes a lot of credit as his contribution to provide his most valuable time and enormous amount of knowledge about the domain of the related industry. Next part of the credit goes to the staff of the company for allowing me to interfere with them to understand the work flow and the valuable help on the practical scenarios.

My heartfelt thanks to my parents and my sisters for being the encouragement of this dissertation. Many thanks to Mr. Chaminda Wijesuriya, who made me to be in this industry, Mr. Yogaraja Mackshan, Mr. Aruna Athulathmudali, Mr. Manjula Peiris, Mr. Sampath De Silva who supported me throughout my career in IT Industry and all the colleagues who helped me with the technical matters with this dissertation.

Contents

Declaration.....	ii
Abstract.....	iii
Contents	v
List of Figures	xi
List of Tables	xiii
List of Acronyms	xiv
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Problems identified and proposed solution	1
1.3 Motivation	2
1.4 Scope of the Proposed Solution	2
1.5 Objectives of the Proposed Solution	3
1.6 Structure of Dissertation.....	4
Chapter 01 – Introduction.....	4
Chapter 02 - Analysis	4
Chapter 03 - Design	4
Chapter 04 - Implementation.....	4
Chapter 05 - Evaluation	4
Chapter 06 - Conclusion	4
Appendixes	5
Chapter 2: Analysis.....	6
2.1 Introduction	6
2.2 Facts gathering techniques	6
2.3 Requirement Analysis	7

2.4	Analysis of the current system	8
2.5	Existing Similar Systems.	9
	QuickBooks	9
	Salesbinder.....	10
2.6	Functional Requirements of the System.....	10
	Customer Management.....	11
	User Administration	11
	Product Management.....	12
	Inventory Management.....	13
	Billing and Payments.....	13
	Report Centre.....	13
	Backup Plan.....	13
2.7	Non-Functional Requirements	13
2.8	Selected Methodology.....	16
Chapter 3: Design of solution		18
3.1	Introduction	18
	Architectural design.....	18
	Logical design.....	18
	Physical design.	18
3.2	Alternate solutions evaluation.....	19
	Mobile applications	19
	Rich client applications	19
	Rich Internet applications.....	20
	Service applications.....	20
	Web applications	20
3.3	Rich Internet Application description and justification	21

Advantages using Rich Internet Application type	22
Advantages using cloud computing than in-house IT infrastructure.....	23
Solution Architecture.....	25
3.4 Object Oriented Approach	26
3.5 Database Design.....	27
Use Case Diagram	27
Database Design	33
3.6 Activity Diagram.....	36
3.7 Sequence Diagram.....	37
3.8 User Interface (UI) / User experience (UX).....	38
Login Screen.....	38
Menu.....	39
Customer Management.....	39
Billing Management	39
Report Center.....	40
Administration	40
UI Components.....	40
Chapter 4: Implementation	43
4.1 Introduction	43
4.2 Implementation Environment.....	43
Server Side Hardware	43
Server Side Software	44
Client Side Hardware.....	44
Client Side Software.....	44
4.3 Development Tools	45
Visual Studio Enterprise 2015	45

C# 6.....	45
.Net Framework 4.6	46
SQL Server 2014	46
TortoiseSVN.....	46
VisualSVN Server	46
AnkhSVN	46
ReSharper	47
SQL Tool belt	47
Microsoft Office Professional Plus 2016.....	47
Chrome Dev Tools.....	48
4.4 Solution Architecture	48
4.5 Re-used existing code.....	50
Entity Framework	50
ASP.NET Web API.....	50
Telerik MVC UI/ Kendo-UI.....	51
jQuery [21]	51
4.6 Major code segments.....	52
Simple CRUD Operation.....	52
Chapter 5: Evaluation	56
5.1 Introduction	56
Validation	56
Verification.....	56
5.2 Testing Levels	56
Unit Testing	56
Integration Testing.....	56
System Testing	57

Acceptance Testing.....	57
5.3 Test Cases.....	57
5.4 User Acceptance Testing (UAT).....	58
Chapter 6: Conclusion.....	60
6.1 Introduction	60
6.2 Critical Assessment of the Project	60
6.3 Lessons Learnt.....	60
6.4 Future Developments	61
SMS/Email Alert system	61
Mobile Version	61
Appendix A : System Documentation	67
A.1 Enable IIS Server.....	67
A.2 Install MSSQL Server	70
A.3 Host Application.....	70
Appendix B : Design Documentation	71
B.1 Use case Diagrams.....	71
B.2 ER Diagrams.....	74
B.3 Activity Diagrams.....	75
B.4 Sequence Diagrams	76
Appendix C : User Documentation.....	77
C.1 Common Master Data Grid	77
C.2 Customer Management.....	80
C.3 Invoicing.....	83
Appendix D : Management Reports.....	84
D.1 Invoice Report	84
Appendix E : Test Results	86

E.1 Log in.....	86
E.2 Master Data screen.....	87
Appendix F : Code Listing	88
F.1 Web Configuration file important code sections.	88
F.2 BL method for Add New Customer	88
F.3 Kendo Helper on View Code Sample	89
F.4 View JavaScript on View Code Sample	90
F.5 Controller Class Code Sample	90
1. Model Class Sample.....	91
Appendix G : Client Certificate	92
Glossary	93
Index	94

List of Figures

Figure 2.1 Existing Process Diagram.....	8
Figure 2.2 QuickBooks Online Screen	9
Figure 2.3 Salesbinder Inventory Screen	10
Figure 2.4 Engineering disciplines iteration	17
Figure 2.5 RUP phase and disciplines	17
Figure 3.1 The typical structure of a rich internet application.....	22
Figure 3.2 Collaboration of the MVC components	25
Figure 3.3 System administrator use case.....	28
Figure 3.4 Product manager use case.....	29
Figure 3.5 Stores clerk use case.....	30
Figure 3.6 Office manager use case.....	31
Figure 3.7 Office clerk use case.....	32
Figure 3.8 Management user use case	33
Figure 3.9 Entity relationship diagram	34
Figure 3.10 Activity diagram for search	36
Figure 3.11 Sequence diagram for add new user	37
Figure 3.12 Login user interface	38
Figure 3.13 Customer list view	39
Figure 3.14 Customer detail view	39
Figure 3.15 Access control view.....	40
Figure 3.16 Date time picker	40
Figure 3.17 Numeric textbox	41
Figure 3.18 Notification partial view.....	42
Figure 4.1 Visual studio 2015	45
Figure 4.2 Solution architecture.....	49
Figure 4.3 Projects in solution	50
Figure 4.4 Sample model class	52
Figure 4.5 Sample view page.....	53
Figure 4.6 Sample partial view page.....	54

Figure 4.7 Sample controller class.....	55
Figure 4.8 Sample business login class.....	55
Figure A.6.1 IIS install screen 01	67
Figure A.6.2 IIS install screen 02	68
Figure A.6.3 IIS install screen 03	68
Figure A.6.4 IIS install screen 04	69
Figure A.6.5 IIS install screen 05	69
Figure A.6.6 IIS install screen 06	70
Figure B.6.7 Entity relationship diagram.....	74
Figure B.6.8 Activity diagram for search	75
Figure B.6.9 Sequence diagram for add new user	76
Figure C.6.10 Common popup layout.....	80
Figure C.6.11 Add new customer view	81
Figure C.6.12 Customer list view	81
Figure C.6.13 Customer view page 01.....	82
Figure C.6.14 Customer invoice history	82
Figure C.6.15 Customer debtors and payment history	82
Figure C.6.16 Invoice header details	83
Figure C.6.17 Invoice lines and totals	83
Figure D.6.18 Invoice report.....	84
Figure D.6.19 Stock on hand report.....	85

List of Tables

Table 3.1 Benefits and considerations for the common application types	21
Table 4.1 Server Side Hardware	43
Table 4.2 Server Side Software	44
Table 4.3 Client Side Hardware.....	44
Table 4.4 Client Side Software	44
Table 5.1 Sample Test Cases	58
Table E.6.1 Login test cases	86
Table E.2 Master data test cases	87

List of Acronyms

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
EF	Entity Framework
ER	Entity Relationship
GPS	Global Positioning System
IAAS	Infrastructure as a service
IIS	Internet Information Service
IT	Information Technology
MVC	Model View Controller
PAAS	Platform as a Service
RAM	Random Access Memory
RIA	Rich Internet Application
SAAS	Software as a Service
SMS	Short Message Service
SSE2	Streaming SIMD Extensions 2
SVN	Subversion
UAT	User Acceptance Testing
UCSC	University of Colombo School of Computing
UI	User Interface
UML	Universal modeling language
UX	User experience
WDDM	Windows Display Driver Model

Chapter 1: Introduction

1.1 Introduction

Since Imperial Auto Care is following the manual book keeping mythology, it is very much difficult to manage everything in books with the huge workload in their business. Also there is a need of speedy and accurate work flow in order to serve the customer in a better manner.

The main objective of the proposed application is to transform this manual book keeping mythology to easy, efficient, reliable and accurate system and help the management focus more on improving the production and help the employees with much more efficient work for the organization.

1.2 Problems identified and proposed solution

Information Technology is widely utilized on almost every business areas at present. Because of the development on Information Technology, people are used to get the advantage on computerized work than manual work because of the efficiency, reliability and accuracy of using computerized systems. People have the tendency to think that machines are more reliable to handle sensitive and non-sensitive data. That is why most of the processing is done by machines. Sri Lanka, being a developing country, encourages automation of the processes of each and every level in business areas.

The motive of our project is to control various activities performed in the office using the computer with the work of Sales and Inventory Management System. From the beginning, the important thing in our mind is that we should concentrate our project work on a subject that is easy to understand and is according to our daily requirement. By keeping this in mind, the subject chosen by is the management. In the present time there is a great rush in these kind of service stations, as these have become necessity for middle and upper class of the society. The project is developed keeping in mind the general needs of the client's day to day operations.

1.3 Motivation

As we know a manual paper based system requires huge man power and unnecessary work load for the employees to maintain every function properly. And the possibility for errors are numerous such as in calculations, etc. Handling their daily operations with a manual paper based system, the management faced various kinds of problems as follows.

- Office staff waste lot of time to manage documents.
- Accuracy of calculations might be doubtful.
- Could not easily retrieve information on stock, billing or purchases because of maintain several books for record keeping on customer inquiry and customers are really unhappy about the process.
- No proper way to handle stock.
- Management cannot take a reports easily.
- Management cannot monitor the day to day activities easily.
- Analysing the trend of customers is a hard work.
- Every office employee can access to every document and there is a security threat.
- Monitor the financial state is digging through a pile of papers

The proposed solution is to overcome these issues. It will minimize the time and cost that will have to spend for the operations in the company. Further, it can improve the efficiency and the productivity of their services, so that, they can also provide a better and a reliable service for their customers.

1.4 Scope of the Proposed Solution

Scope of the proposed solution as follows.

- Provide a useful, highly valuable, platform independent web based Sales & Inventory Management system for the client.
- To provide accessibility to the system anytime from anywhere.
- Facilitate to have all the current activities managed through the system.

- Improve the efficiency and accuracy of the operation of the client by providing user-friendly data entering and monitoring screens.
- Generate view and print project reports easily.
- Fulfill non-functional requirements such as security, quality, scalability, maintainability and reusability etc.
- To practically apply the knowledge gathered throughout the period of BIT Degree Program.

1.5 Objectives of the Proposed Solution

An automated web based system was proposed to perform their internal and external activities accurately. Objectives of the proposed solution are listed below.

- Guarantee the accuracy of calculations.
- Increase the efficiency of employees.
- Reduce the paper work.
- Guarantee the security of data with proper user management.
- Daily tracking of business in detail.
- Generate proper reports for the management to monitor activities and to make predictions for future decisions.
- Management can handle the business remotely, without attending to the office.
- Provide a good and accurate service to customers.

The system should be able to satisfy the following requirements:

- The system should be able to keep the records of the customers.
- The system should be able to keep the records of the products and services.
- The system should be able to keep the inventory management and reordering process
- The system should be able to generate the invoice for a customer.
- The system should provide management reports for given criteria.

1.6 Structure of Dissertation

This Dissertation provides the overall knowledge about the Transport Management System. Dissertation structure as follows

Chapter 01 – Introduction

The introduction chapter describes about the problem domain, introduction about the client, the motivation, objectives and scope of the project.

Chapter 02 - Analysis

The analysis chapter describes analysis of the current process and functional and non-functional requirements of the project. Also it contains requirement gathering techniques and details of existing solutions to address the issues.

Chapter 03 - Design

The design chapter describes the details about the how designed the proposed system. This chapter contains use-case diagram, ER diagram, Class diagram and sequence diagram and also include the user interface Designs.

Chapter 04 - Implementation

The implementation chapter describes the implementation of the system. It will explain systems' major source codes and module structures also; the implementation environment and it will describe tools Used to development, data structure and platform dependence of the software.

Chapter 05 - Evaluation

The evaluation chapter describes test plan that was used to verify and validate the system and also small description of the effect of errors.

Chapter 06 - Conclusion

This will be the conclusion of the project. It will discuss what we have achieved, problems we encountered and lessons learned by this project. It will also discuss suggestions of future improvements could be made.

Appendixes

These are provided further details about the content of the dissertation chapters which were not included in the chapters.

Chapter 2: Analysis

2.1 Introduction

Requirement gathering is a very important part of the software development lifecycle. All the rest of the life cycle depends on the requirements. Requirements gathered will be helpful to understand the business domain and to get the overall picture of the system. Fact gathering techniques like field visits, observation, questionnaires, face to face interviews and debriefings will be used to gather requirements.

2.2 Facts gathering techniques

Requirement gathering techniques are used for understanding about the business domain. It is necessary to identify the main functional areas and non-functional areas of the manual system.

In analysis stage the client knows more about the domain and we must acquire their knowledge about the activities by using different fact gathering techniques such as interviewing, questionnaires etc. Requirement gathering is very time consuming activity because the selecting suitable people to be interviewed and to be asked questions about the process for the understanding about the system.

So the people who are working in several departments and the people who are in administration level of industry were interviewed according to the pre-defined check list and go through their manual project management process.

There are several fact gathering techniques were used to determine the domain and for preparation of the new solution

Interviews

Interview is the most commonly used technique to collect information from the face-to-face interviews. The purpose of interview is to find, verify, clarify facts, motivate end-users involved, identify requirements and gather ideas and opinions. The role of interview includes

interviewer who is system analyst and interviewee who are system owner or user. Interviewing technique needs good communication skills for interaction between system analyst and user.

Interviews were arranged with office clerk and accountant of the organization to gather information about the data and process of the manual system.

Sampling of existing documents and forms

In order to get a wide understanding of the process it is much easier to go through the current documentation and forms which were used. The organization has invoice documents, ledger books and stock books which were used in the process. And also had excel forms which were used. And all those were referred on this phase.

Observation of the work environment

Another fact finding technique is observation. In this technique, system analyst participates in the organization, studies the flow of documents, applies the existing system, and interacts with the users. Observation can be a useful technique when the system analyst has user point of view. Sampling technique called work sampling is useful for observation. By using this technique, system analyst can know how employees spend their days.

2.3 Requirement Analysis

Requirement analysis can be said to identify investigation, documenting and analysing requirements of the system. The first investigation step of the need of a system is a major activity of problem analysis. This is the role of a system analyst to determine the actual requirement of a system. Sometimes, system analyst may make mistake in discovering the requirements and in trying to analyze a problem. As a result, they may come up with the wrong solution in designing and implementation of a solution. And the solution will not be able to solve the real problem and even cause a new problem in later system analysis process. Therefore, this paper looks at one of the effective methods (fact-finding) for system analyst to help them in gathering information. [1]

In Software engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish [3].

Mainly requirement analysis is divided into two levels namely functional requirement and non-functional requirement

Functional requirements describe functionality or services that the system is expected to provide. Functional requirements of system are mentioned below according to the scope.

2.4 Analysis of the current system

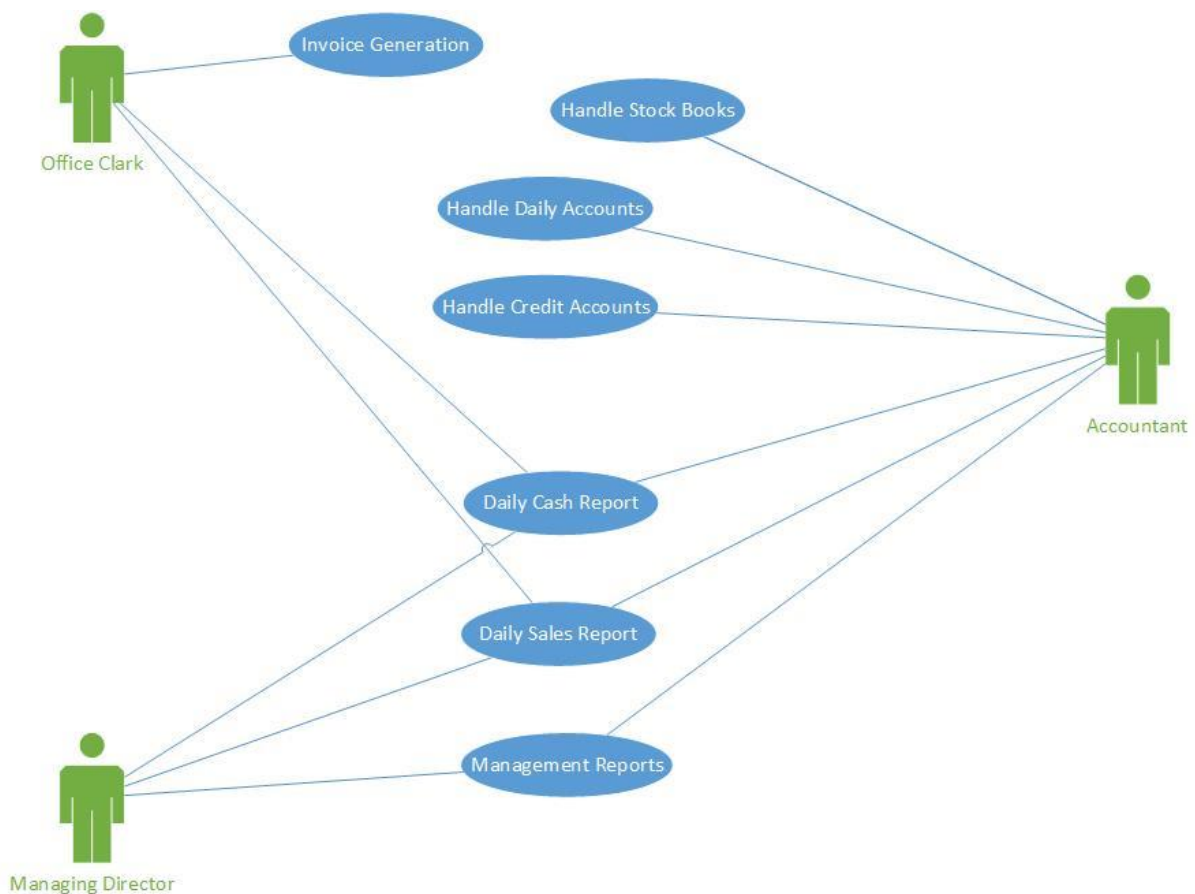


Figure 2.1 Existing Process Diagram

2.5 Existing Similar Systems.

QuickBooks

QuickBooks is the leading accounting software for small businesses. Manage all your finances with either our licensed or online version. Instant access to customer, vendor and employee information. Free support and upgrades with online version. [2]

Quickbooks is an applications which were used by many small organizations as there accounting software. This have a very detail accounting module with covering most of the needs. Also facilitate to maintain customers,vendors,invoices and purchase orders in the system. Since this software is a standalone windows application one main computer should be in place to have the software and there is only one user access to the system. If we need to get access with multiple users a higher subscription should be purchased and it will cost a significant amount for the purchase.

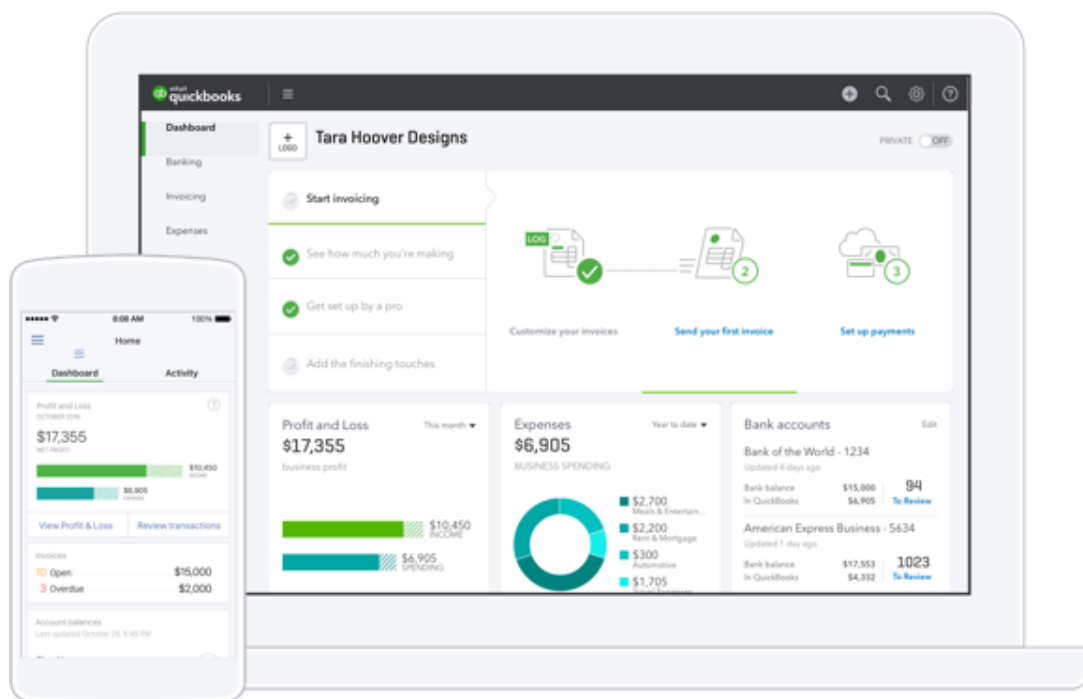


Figure 2.2 QuickBooks Online Screen

Salesbinder

Salesbinder is the easiest-to-use, full-featured online inventory management system for any small-medium enterprise (SME). Key features include: invoicing, estimates, purchase orders, barcoding, reporting, account management (CRM), multiple locations, item variations, custom user permissions, a full API and many online integrations.

Salesbinder is subscription based web application which handle by a USA company. The subscription can be monthly or annually. There are difficulties to add more fields to the system as required for the client to keep client and vehicle information since the software have no feature as such.

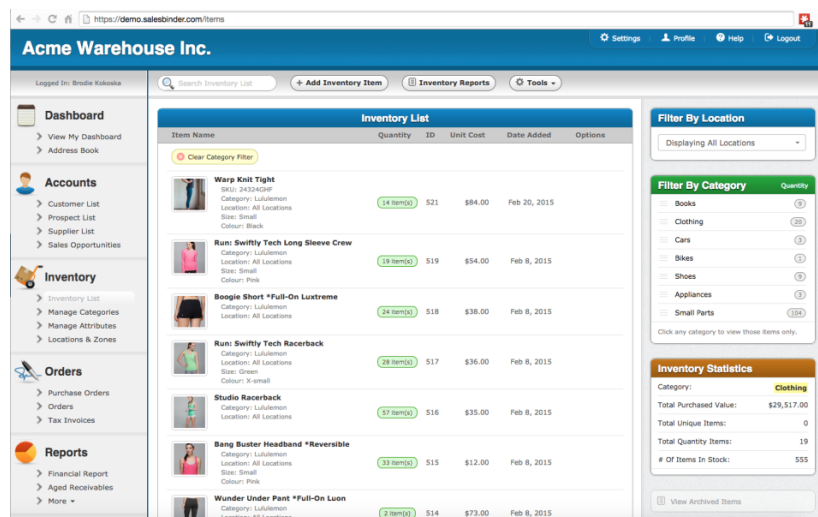


Figure 2.3 Salesbinder Inventory Screen

2.6 Functional Requirements of the System

With the requirements of the client, the system will define under following categories,

- Customer Management
- User Management
- Product Management
- Inventory Management
- Billing & Payments
- Reporting

- Backup Plan

Customer Management

Customer management module includes every function related to customer. Creating new customer, updating customer information and lookup on history transactions are key functions. Customers are categorized into two areas as general customer and vehicle customer. General customers are buying goods without a vehicle no reference. And vehicle customer needs to have a vehicle no. Also vehicle customers then categorized with their vehicle type.

All the transactions and payments are shown on customer account interface and can be filtered by multiple parameters.

User Administration

All user management activities are done in User Management sub module.

Manage Users

The system shall support the concept of user. Every user of the system has a name and a password. The name must be unique within the installed instance of the system. In addition, every user has a set of properties: Full Name, Full Business Title (Company Name, Position), E-Mail Address, Phone, Working Address, Alternative Phone, and Alternative Working Address. Each user is uniquely identified by its name within the system.

The system shall provide the authenticated user the ability to change his or her password and to store the change. The user profile contains the user-specific configurable parameters of the system. The user profile is associated with one and only one user that is registered in the system (has a user name and a password). The user must be able to change his or her profile and save the changes.

The system should be capable of view users, adding new users, modify existing users and delete existing users and the accessibility to the options is managed by the permission levels.

Manage Roles

The system shall support the concept of user role. The role has the unique name within the installed instance of the system and a set of permissions that are assigned to this role. The permission determines explicitly what the user belonging to this role allowed to do in the system. Every user of the system must be associated with at least one of the roles. The user can belong to many roles. If the user is member of several roles, the deny permission take over the grant one.

The system should be capable of view roles, adding new roles, modify existing roles and delete existing roles and the accessibility to the options is managed by the permission levels.

Manage User Logins

The user must login to the system by specifying his or her name and password before he or she can work with the system. If the password is invalid or the user name does not exist in the system, the user is not allowed to login and must enter the name and password again. There is no limit of the login tries. After successful login the system shall associate the user with the user roles and configure appearance of GUI according the user profile. After the login the main functionality of the system according the user's permissions is available. After the login the user becomes the authenticated and authorized user.

User Activity Monitor

Every user function is captures and logged in a user log for further use. The accessibility to the options is managed by the permission levels.

Product Management

Keep track of the products with status and availability for invoicing. This includes product management with altering product details, reorder levels and stocks, manage rates on products.

Inventory Management

All the inventory management handled by the stock keeper is maintained in Inventory Sub Module. This handles update GRN, Stock Counts, Stock Adjustments.

Billing and Payments

Invoice generation and printing is handled. Method of payment and details of payment is updated through this sub module.

Report Centre

All related reports are generated to the purpose of Management Information Reports.

Backup Plan

Backing up the system and the database to an external location as an initiative of disaster recovery plan.

2.7 Non-Functional Requirements

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behaviour. One could also think of non-functional requirements as quality attributes for of a system.

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviours, for example: “Modified data in a database should be updated for all users accessing it within 2 seconds.” [2]

Performance

The system should be fast and quickly accessible. Customers are always in a hurry so the user needs to be able to give the service to the customer as quick as possible. In order to increase the system speed certain areas are improved in data retrieval and update procedures. The system loading time, processing time and query and reporting time should be fast.

Accuracy

Every calculation is accurate to the fraction. User will hesitate to use the system if there is a single deviation in the application. It is critical to be accurate because financial transactions also handle by the system.

Availability

The system needs to be available from anywhere by any device. The management needs to be able to logging and see what is happening in the system any time. Availability of system for each licensed user. This system will available for 24 hours. The number of concurrent users will be limited according to the number of licenses mentioned in the proposal.

Reliability

Since the customer will depend on the application, it should be always available and can be accessed easily. Application maintenance and disaster recovery plans has to be taken in place in case of emergency. Such as up to date firewalls, anti-virus applications and database backups, application host backups.

Security Requirements

The users who have permission to use the system will have the ability to change their password time to time.

The user authentication will be done by the system. There will be a facility to assign expiry dates for the users when necessary.

User Friendliness

System has a simple, clean and eye catching user interface. User interface design has been done by considering user experience standards. Minimized mouse clicks. User can navigate through the form elements via keyboard entries. Refreshing page for every user action is not recommended in modern web.

Fault Recovery

In case of a failure, system should be able to recover within a short period.

Maintainability

The system is designed to adapt the new changes without affecting other components.

Response Time

Speed is a very crucial fact of a web application. If a user does not get what expected within 7 seconds it will decrease the interest of the task. To speed up the web application content compression and AJAX technologies are used.

Processing Time

Processing time can be highly effected to the response time of a web application. Powerful processing power and optimized server side coding will lead the application to master processing.

Notifications

Automated notification generation is a must in modern web designing. Important notification like low stock levels will save the time that user spend to digging over the data.

2.8 Selected Methodology

RUP Stands for "Rational Unified Process." RUP is a software development process from Rational, a division of IBM since 2003. It divides the development process into four distinct phases that each involve business modeling, analysis and design, implementation, testing, and deployment. The four lifecycle phases are:

- **Inception**

The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.

- **Elaboration**

The project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.

- **Construction**

The project is developed and completed. The software is designed, written, and tested.

- **Transition**

The software is released to the public. Final adjustments or updates are made based on feedback from end users.

The RUP development methodology provides a structured way for companies to envision create software programs. Since it provides a specific plan for each step of the development process, it helps prevent resources from being wasted and reduces unexpected development costs [5].

Within each iteration of software development lifecycle, the tasks are categorized into six engineering disciplines as follows. Iteration of disciplines depicted in Figure 2.3.

- Business modelling
- Requirements
- Analysis and design
- Implementation
- Test
- Deployment

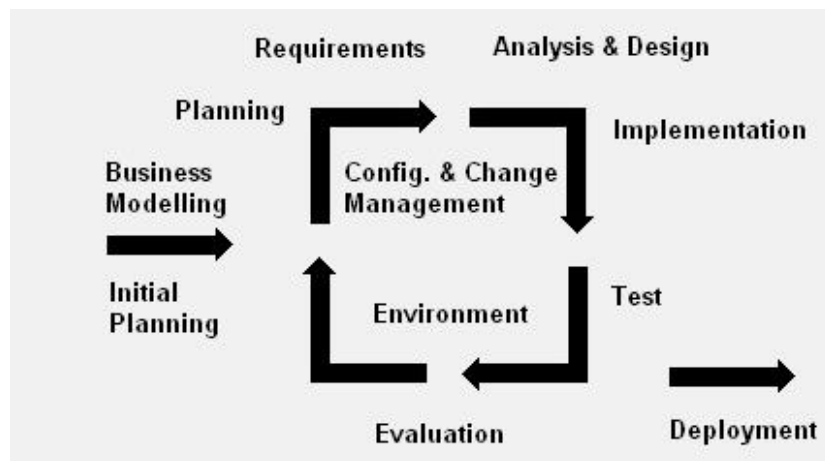


Figure 2.4 Engineering disciplines iteration

With the Iterative cycles, business modeling, Requirement gathering, Analyzing and Design and implementation discipline tasks weights fade away. Deployment tasks will increase at the end of the time. Testing has been spared through the whole period. Engineering disciplines spread through the lifecycle phases as demonstrated in Figure 2.4.

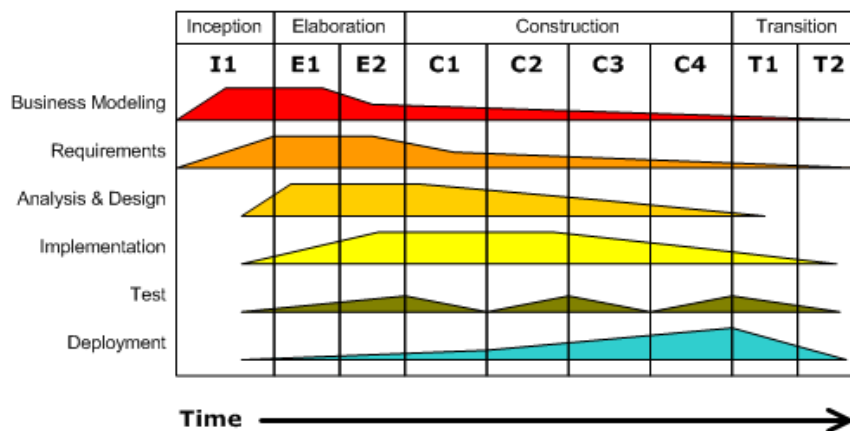


Figure 2.5 RUP phase and disciplines

Chapter 3: Design of solution

3.1 Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements captured in requirement analysis phase. System designing comes under elaboration phase of software developing. Object-oriented methods are becoming the most widely used methods for systems design. The UML has become the standard language in object-oriented analysis and design. Object oriented techniques and UML modeling languages were used to designing system.

If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user [6].

There are three levels of design.

Architectural design

The architectural design of a system emphasizes the design of the systems architecture that describes the structure, behaviour and more views of that system and analysis [6].

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams) [6].

Physical design.

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks.

- User Interface Design
- Data Design
- Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase [6].

System designers may use various system design approaches. Such as modern structured design, prototyping, Joint Application Design (JAD), Rapid application development (RAD) and object-oriented design.

3.2 Alternate solutions evaluation

Since it was decided to design this application in Microsoft [7] development platform, this paragraph indicates the common application types which identified by the company.

Mobile applications

Applications of this type can be developed as thin client or rich client applications. Rich client mobile applications can support disconnected or occasionally connected scenarios. Web or thin client applications support connected scenarios only. Device resources may prove to be a constraint when designing mobile applications.

Rich client applications

Applications of this type are usually developed as stand-alone applications with a graphical user interface that displays data using a range of controls. Rich client applications can be designed for disconnected and occasionally connected scenarios if they need to access remote data or functionality.

Rich Internet applications

Applications of this type can be developed to support multiple platforms and multiple browsers, displaying rich media or graphical content. Rich Internet applications run in a browser sandbox that restricts access to some features of the client.

Service applications

Services expose shared business functionality and allow clients to access them from a local or a remote system. Service operations are called using messages, based on XML schemas, passed over a transport channel. The goal of this type of application is to achieve loose coupling between the client and the server.

Web applications

Applications of this type typically support connected scenarios and can support different browsers running on a range of operating systems and platforms.

The following table indicates the benefits and considerations for the common application types.

Application type	Benefits	Considerations
Mobile applications	Support for handheld devices. Availability and ease of use for out of office users. Support for offline and occasionally-connected scenarios.	Input and navigation limitations. Limited screen display area.
Rich client applications	Ability to leverage client resources. Better responsiveness, rich UI functionality, and improved user experience. Highly dynamic and responsive interaction. Support for offline and occasionally connected scenarios.	Deployment complexity; however, a range of installation options such as Click Once, Windows Installer, and XCOPY are available. Challenging to version over time. Platform specific.
Rich Internet applications (RIA)	The same rich user interface capability as rich clients. Support for rich and streaming media and graphical display. Simple deployment with the same distribution capabilities (reach) as Web clients.	Larger application footprint on the client compared to a Web application. Restrictions on leveraging client resources compared to a rich client application. Requires deployment of a suitable runtime framework on the client.

	Simple upgrade and version updating. Cross-platform and cross-browser support.	
Service applications	Loosely coupled interactions between client and server. Can be consumed by different and unrelated applications. Support for interoperability.	No UI support. Dependent on network connectivity.
Web applications	Broad reach and a standards-based UI across multiple platforms. Ease of deployment and change management.	Dependent on continual network connectivity. Difficult to provide a rich user interface.

Table 3.1 Benefits and considerations for the common application types

After considerations of pros and cons with the application types and the requirement of the client, it was decided to use the application type of Rich Internet Application as the perfect application type for the system.

3.3 Rich Internet Application description and justification

A rich Internet application (RIA) runs in the browser in a sandbox. The benefits of a RIA over traditional Web applications include richer user experience, improved user responsiveness, and improved network efficiency. A RIA will normally be structured as a multi-layered application consisting of user experience (presentation), service, business, and data layers, as shown in following figure 3.1.

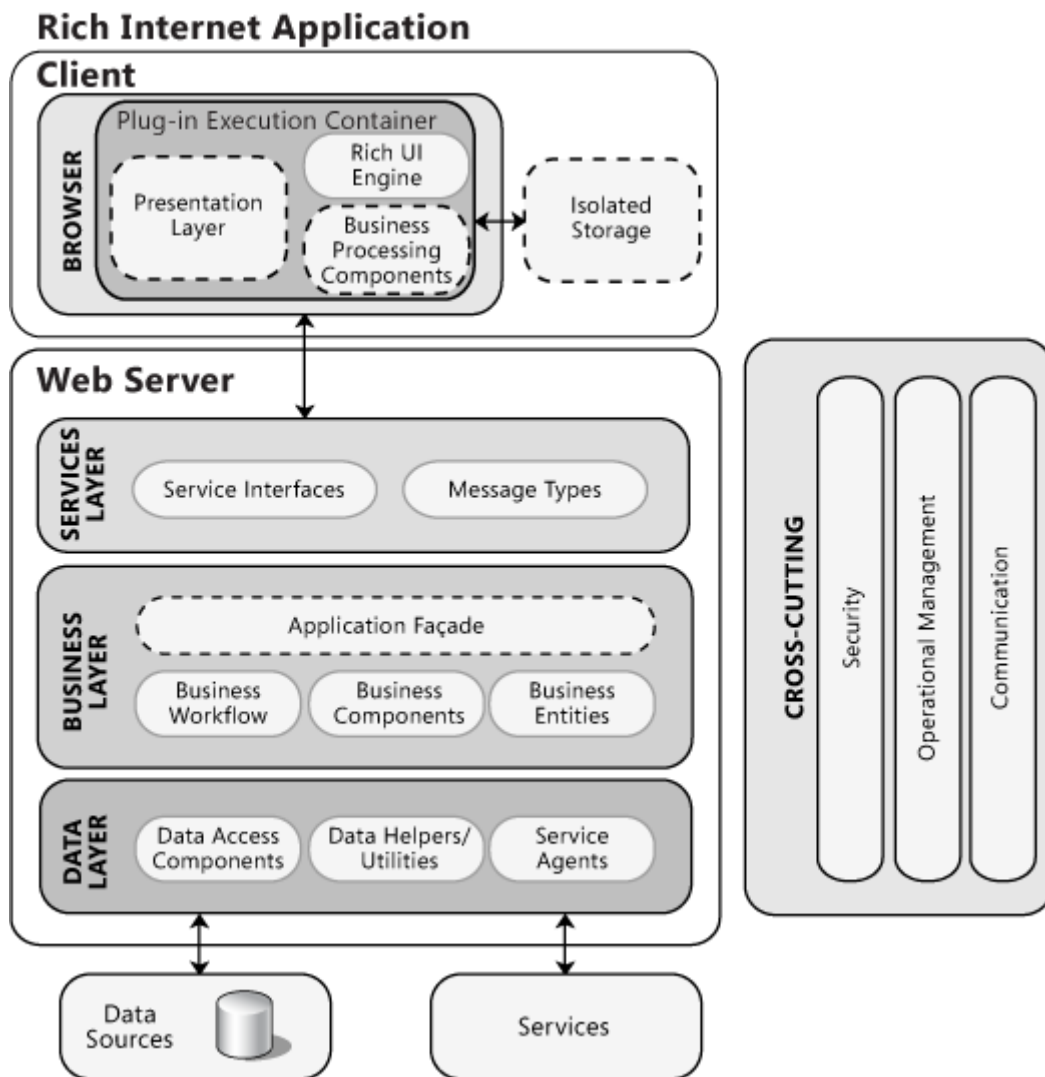


Figure 3.1 The typical structure of a rich internet application

With the RIA client would have the continuous access to the application any place that world wide web accessible. Plus, the weight of the application mostly bared by the server side, hence the client side would not need to be very rich with resources such as processing power and memory. In that case client would reduce their cost on hardware. Web browser is the only extra software that would be needed by the client to access the application which, most of the leading modern browsers are freely available to download and install.

Advantages using Rich Internet Application type

- **A better user experience**

It is a lot easier and cheaper to make a web based system user friendly across multiple platforms and various screen sizes.

- **Worldwide Access**

Employees can work from multiple offices, a client's site, a hotel, or even from home.

- **Client secure login**

Impress clients with a modern web portal and improve customer service with a more automated process.

- **Easy install**

It takes a couple of minutes to get someone set up on a web app. Provide them with the URL, a user name, and password then they are away.

- **Always up-to-date**

As everyone is accessing the same web app via the same URL they will all be seeing the most up-to-date version of your system.

- **Cross platform**

Your apps can be easily ported to virtually any platform with a web browser.

After considering the facts, it was decided to host the application on virtual machine in cloud. There are couple of major cloud service providers in the market. Among those competitors Microsoft Azure [9] was chosen since the service covers all major cloud services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [10].

Advantages using cloud computing than in-house IT infrastructure.

- **Cost Savings**

Perhaps, the most significant cloud computing benefit is in terms of IT cost savings. Businesses, no matter what their type or size, exist to earn money while keeping capital and operational expenses to a minimum. With cloud computing, you can save substantial capital costs with zero in-house server storage and application requirements. The lack of on-premises infrastructure also removes their associated operational costs in the form of power, air conditioning and administration costs. You pay for what is used and disengage whenever you like - there is no invested IT capital to worry about. It's a common misconception that only

large businesses can afford to use the cloud, when in fact, cloud services are extremely affordable for smaller businesses.

- **Reliability**

With a managed service platform, cloud computing is much more reliable and consistent than in-house IT infrastructure. Most providers offer a Service Level Agreement which guarantees 24/7/365 and 99.99% availability. Your organization can benefit from a massive pool of redundant IT resources, as well as quick failover mechanism - if a server fails, hosted applications and services can easily be transited to any of the available servers.

- **Manageability**

Cloud computing provides enhanced and simplified IT management and maintenance capabilities through central administration of resources, vendor managed infrastructure and SLA backed agreements. IT infrastructure updates and maintenance are eliminated, as all resources are maintained by the service provider. You enjoy a simple web-based user interface for accessing software, applications and services – without the need for installation - and an SLA ensures the timely and guaranteed delivery, management and maintenance of your IT services.

- **Strategic Edge**

Ever-increasing computing resources give you a competitive edge over competitors, as the time you require for IT procurement is virtually nil. Your company can deploy mission critical applications that deliver significant business benefits, without any upfront costs and minimal provisioning time. Cloud computing allows you to forget about technology and focus on your key business activities and objectives. It can also help you to reduce the time needed to market newer applications and services.

Since the development platform became Microsoft it is an advantage to do the develop on the Windows operating system (OS). Personal Computer with the OS Windows 10 [11] Pro 64-Bit has been selected as the development platform. Visual studio 2015 Enterprise, the official Integrated Development Environment (IDE) was chosen to develop the system.

Solution Architecture

With regarding the application architectural design there will be multi layered architecture. Layers as follows.

- **Presentation Layer**

Because this application constrained to the browser, they tend to work best when designed as one central interface. Applications with multiple pages require that consideration how to link between pages. It was decided to use C# [9] ASP.Net MVC [9] framework to develop the presentation layer, which is the most successful web application methodology invented and contributed by the Microsoft.

Model View Controller (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. Collaboration of the MVC components depicted in Figure 3.2.

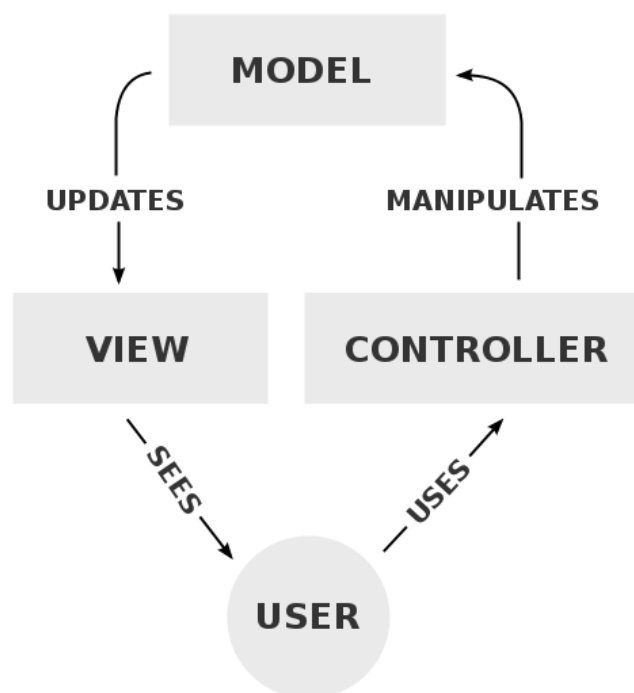


Figure 3.2 Collaboration of the MVC components

The model directly manages the data, logic and rules of the application and stores data that is retrieved according to commands from the controller and displayed in the view. View generates new output to the user based on changes in the model and a view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. The third part, the controller, accepts input and converts it to commands for the model or view and the controller itself can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model. Above figure explains the collaboration between three components.

Models will be generated according to the data tables and will be included in Data layer. Controllers would be created according to the UI design.

- **Business Layer**

In most scenarios, RIAs will access data or information located outside the application. While the nature of the information will vary, it is likely to be extracted from a business system. To maximize performance and usability, it will be considered locating some of the business processing tasks on the client.

- **Data Layer**

Also described as Data Access Layer. Only authorized layer to connect to the database and access data. Data models will also be created in data layer.

- **Database Management System**

A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases.

3.4 Object Oriented Approach

Object oriented approach is the widely used designing methodology in the world. Objects are close to real world entities and easy to handle and understand. And object oriented code is reusable and easy to maintain.

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design.

An object contains encapsulated data and procedures grouped together to represent an entity. The 'object interface' defines how the object can be interacted with. An object-oriented program is described by the interaction of these objects. Object-oriented design is the discipline of defining the objects and their interactions to solve a problem that was identified and documented during object-oriented analysis [8].

3.5 Database Design

Normalized relational database is the most suitable database to achieve the goal in object oriented approach. Normalization minimize the data redundancy by dividing large tables to number of small tables. Normalization involves arranging attributes in tables based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished through applying some formal rules either by a process of synthesis or decomposition. Synthesis creates a normalized database design based on a known set of dependencies. Decomposition takes an existing (insufficiently normalized) database design and improves it based on the known set of dependencies [9].

Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is

involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

The use case module mainly has following Actors.

- System Administrator
- Product Manager
- Stores Clarke
- Office Manager
- Office Clarke
- Management User

Actors and their responsible processes are listed below.

- System Administrator

System administrator is responsible for manage users and user permissions throughout the system. System administrator's use case diagram showing in figure 3.4

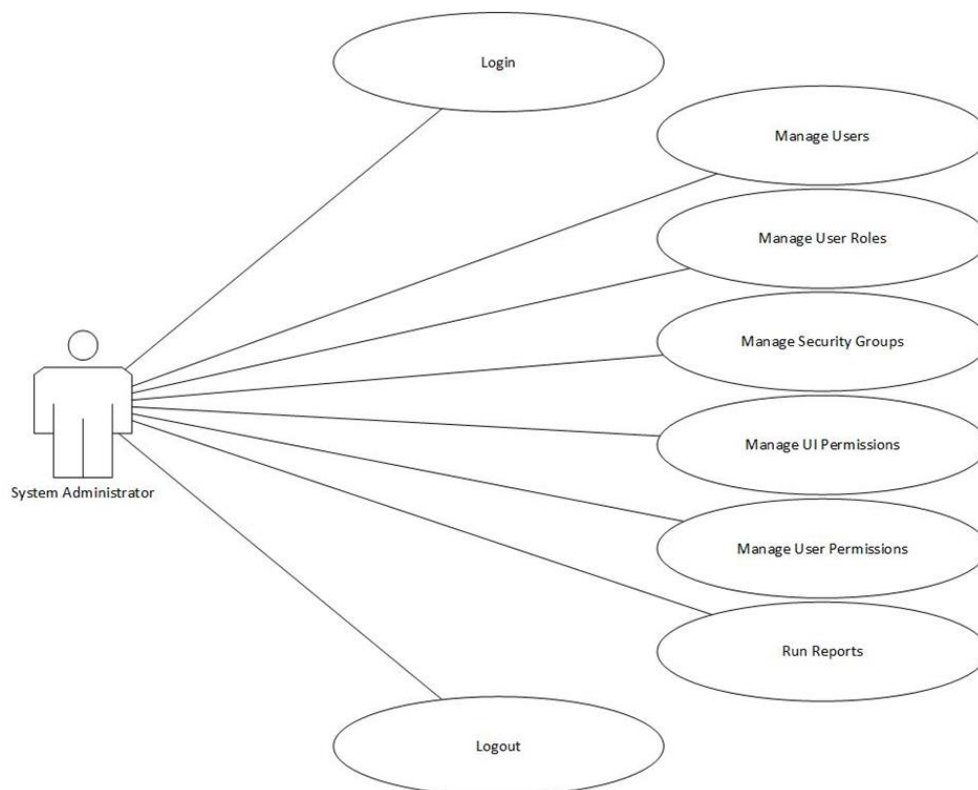


Figure 3.3 System administrator use case

- Product Manager

Product manager is responsible for create, update products, price lists and reordering. Product manager's use case showing on figure 3.5

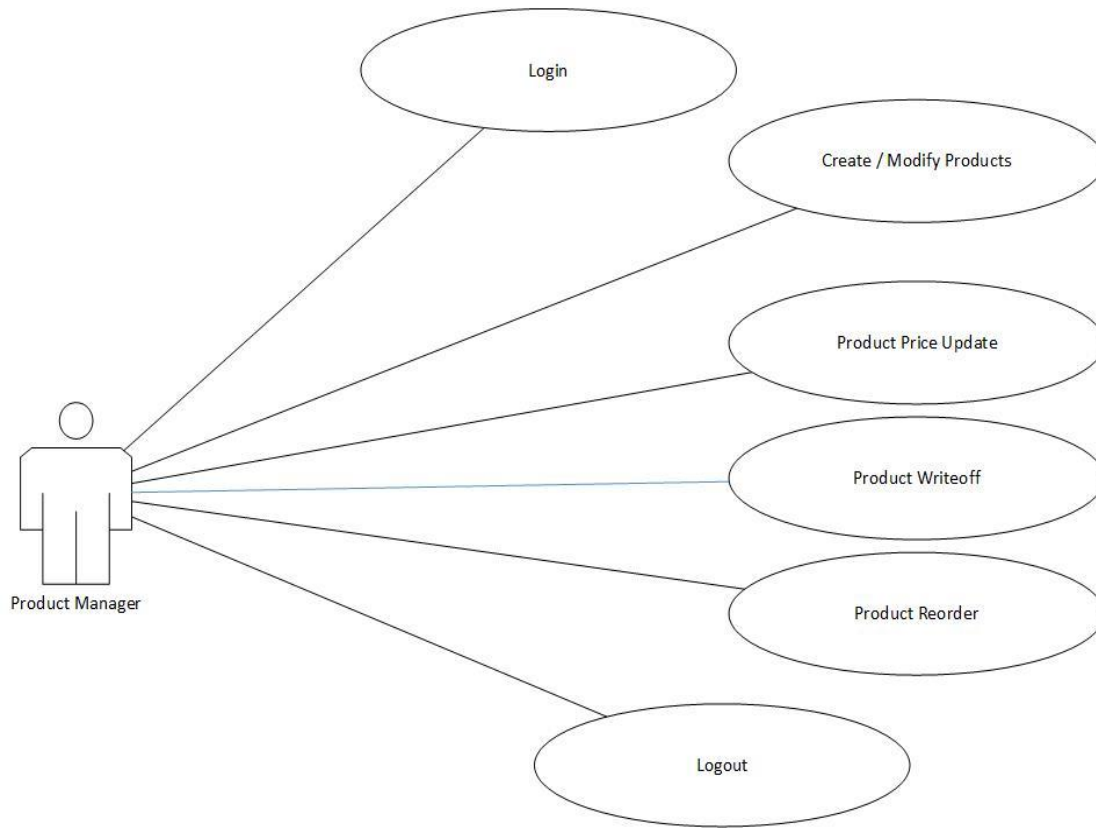


Figure 3.4 Product manager use case

- Stores Clerk

Stock in out update, damage stock update, bin management will be main tasks for stores Clarke. Stores clerk's use case diagram is showing on figure 3.6

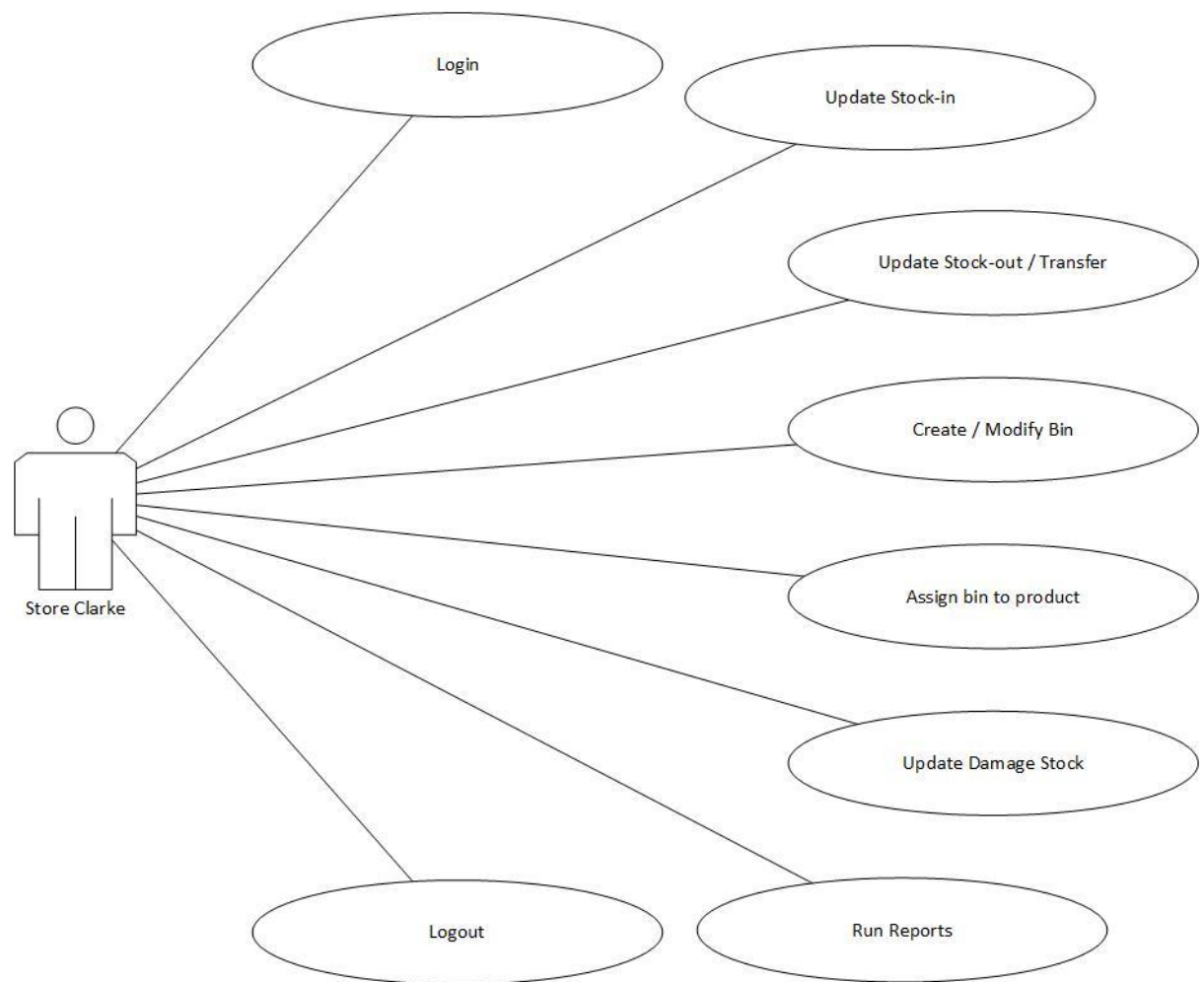


Figure 3.5 Stores clerk use case

- Office Manager

Office Manager will approve credit notes and change requests and get daily, monthly reports. Manager's use case is showing on figure 3.7

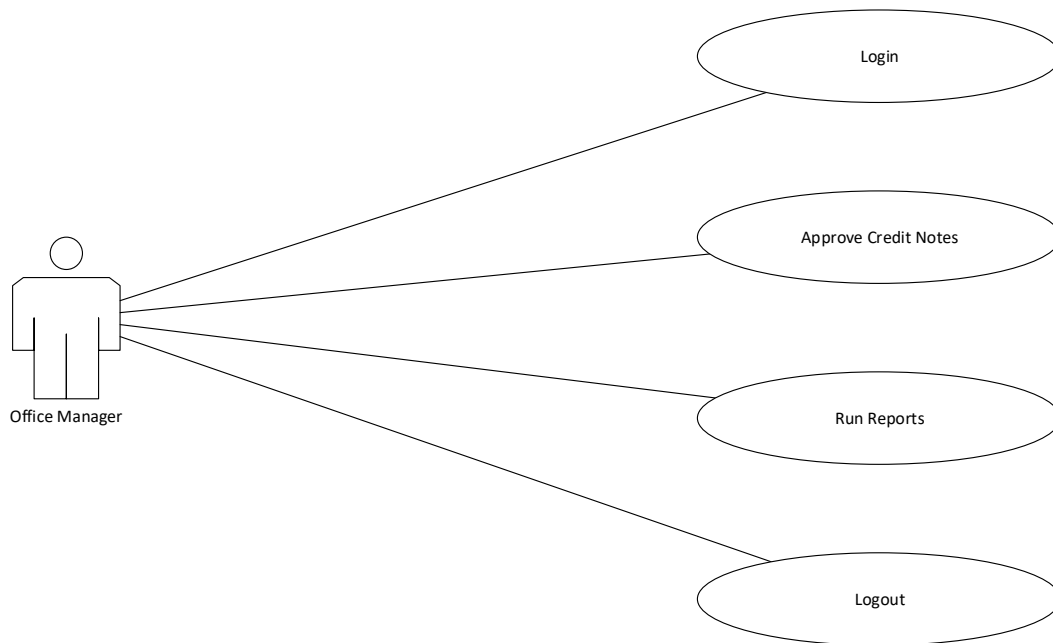


Figure 3.6 Office manager use case

- Office Clark

Create/Update Customer info, Create Invoices/Credit notes will be Office Clark's role.

Clerk's use case is showing on figure 3.8

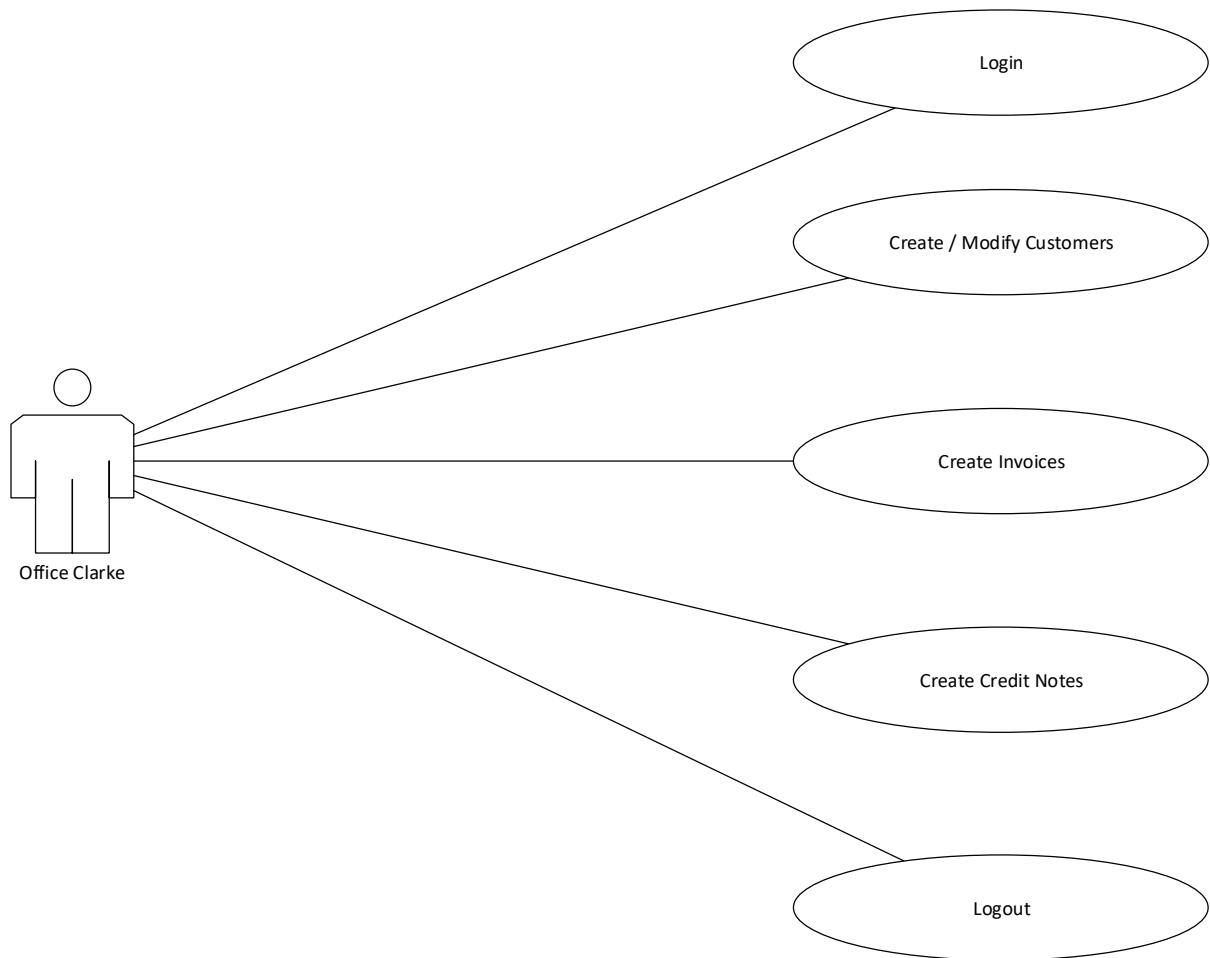


Figure 3.7 Office clerk use case

- Management User

Management User will process reports for required information. Management user's use case is showing on figure 3.9

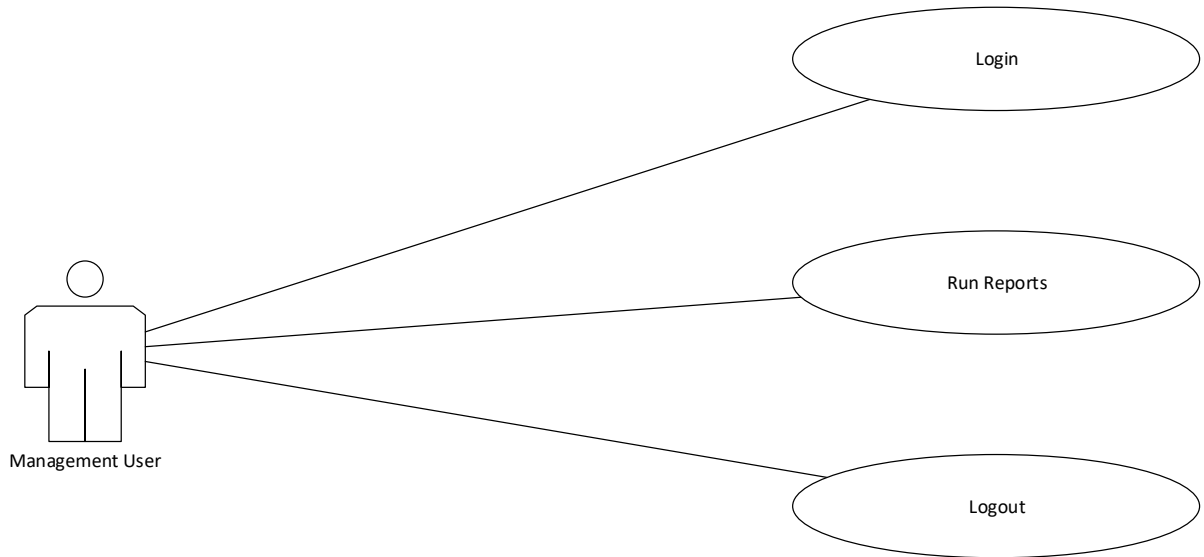


Figure 3.8 Management user use case

Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and view. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and

queries used as part of the overall database application within the database management system (DBMS).

- **Entity Relationship Diagram**

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

- Entities
- Relationships
- Attributes

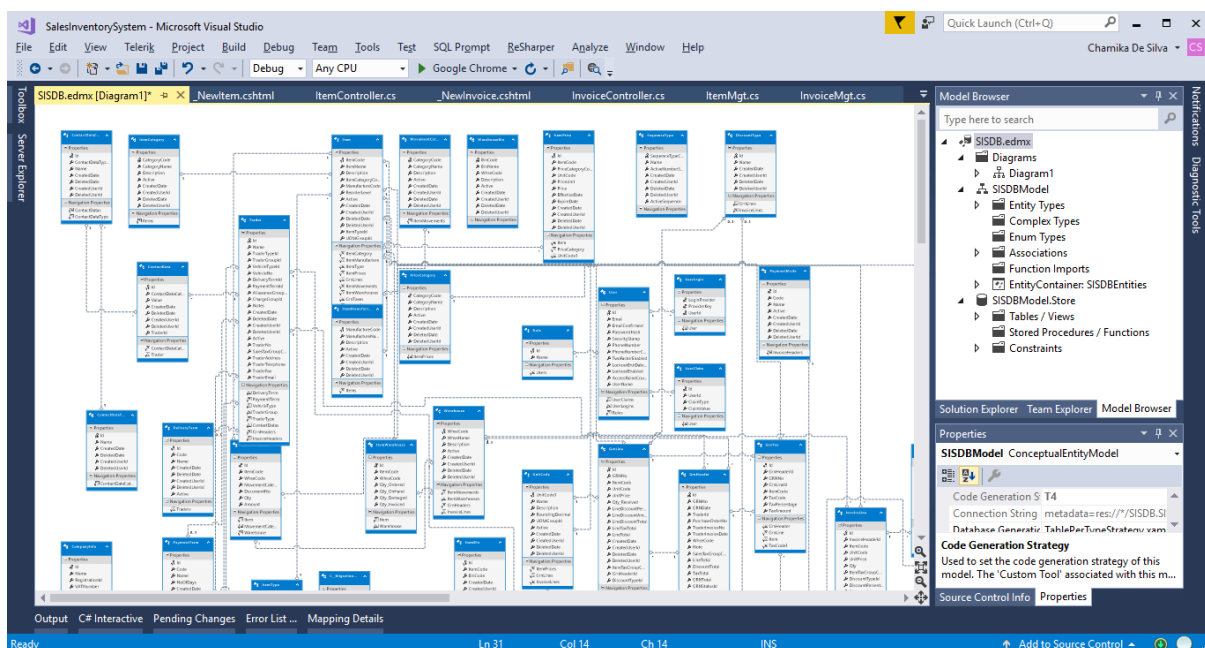


Figure 3.9 Entity relationship diagram

- **Database Normalization**

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

- **Levels of Normalization**

First Normal Form (1NF)

First normal form (1NF) sets the very basic rules for an organized database:

- Eliminate duplicative columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second Normal Form (2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data:

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign.

- **Third Normal Form (3NF)**

Third normal form (3NF) goes one large step further:

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

3.6 Activity Diagram

In UML, an activity diagram is used to display the sequence of activities. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.

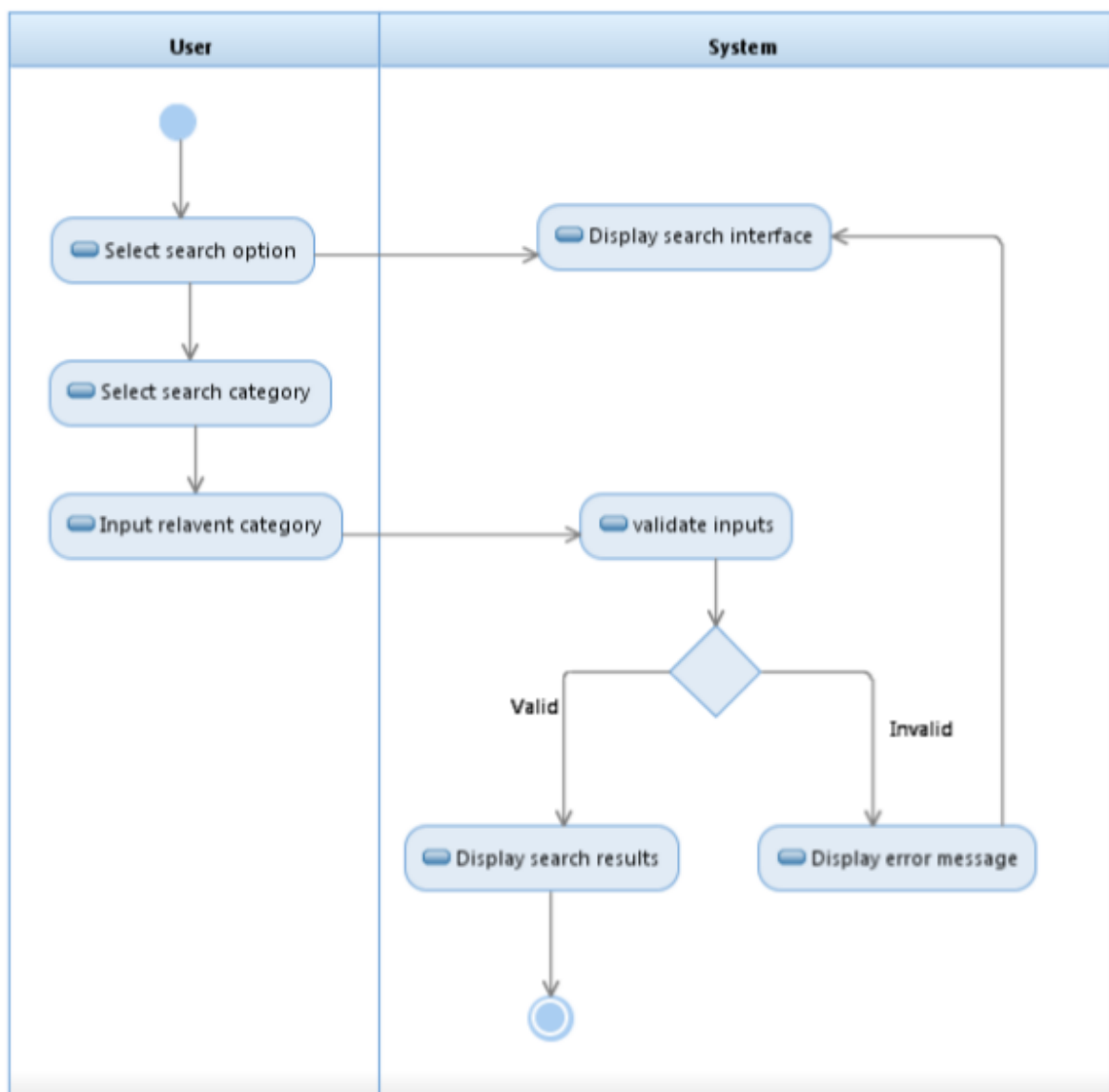


Figure 3.10 Activity diagram for search

3.7 Sequence Diagram

A sequence diagram is a form of interaction diagram which shows objects as lifelines running down the page, with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. Sequence diagrams are good at showing which objects communicate with which other objects; and what messages trigger those communications. Sequence diagrams are not intended for showing complex procedural logic.

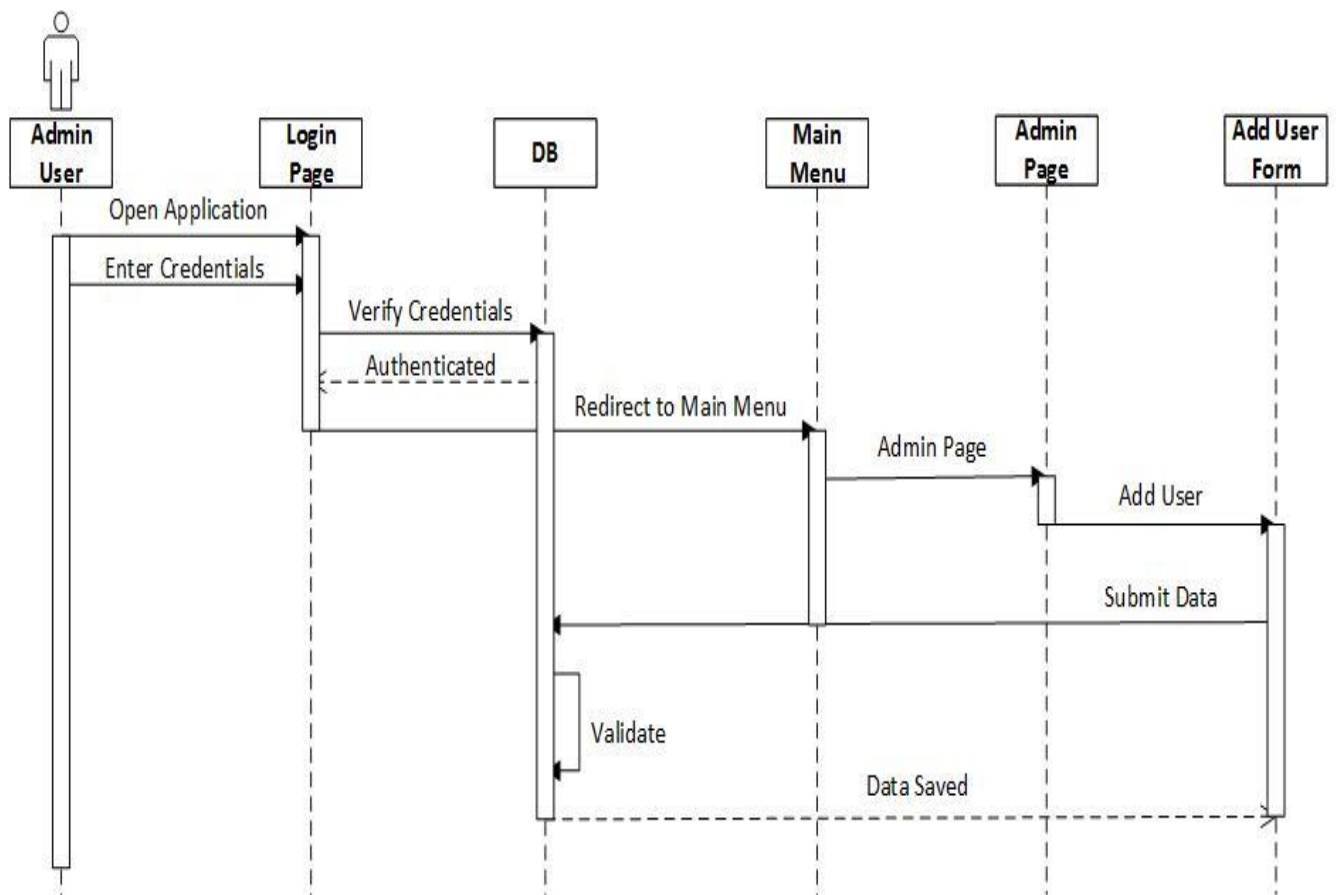


Figure 3.11 Sequence diagram for add new user

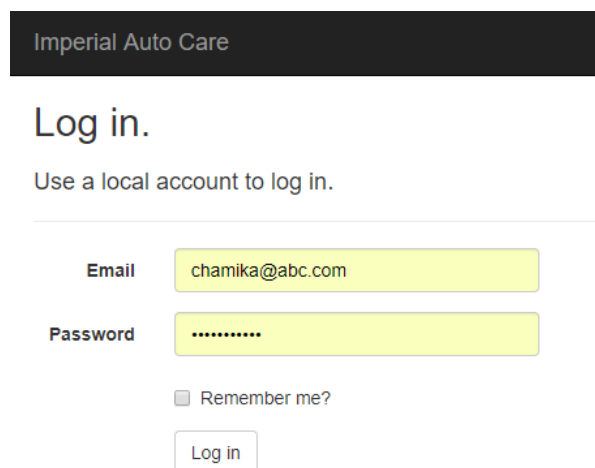
3.8 User Interface (UI) / User experience (UX)

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Login Screen

This is the first thing user will see in the system. Modern and eye catching look should contain in this screen. Apart from the clean and charm looking login form, background image was placed to feel professional at the first sight. Login screen view showing on figure 3.13



The image shows a login screen for 'Imperial Auto Care'. At the top, there is a dark header with the text 'Imperial Auto Care' in a light blue font. Below the header, the text 'Log in.' is displayed in a large, bold, dark blue font. Underneath, a smaller line of text says 'Use a local account to log in.' followed by a horizontal line. The login form consists of two input fields: 'Email' with the value 'chamika@abc.com' and 'Password' with masked characters '.....'. Both fields have a light yellow background. Below the password field, there is a checkbox labeled 'Remember me?'. At the bottom of the form, there is a 'Log in' button with a light blue border and text.

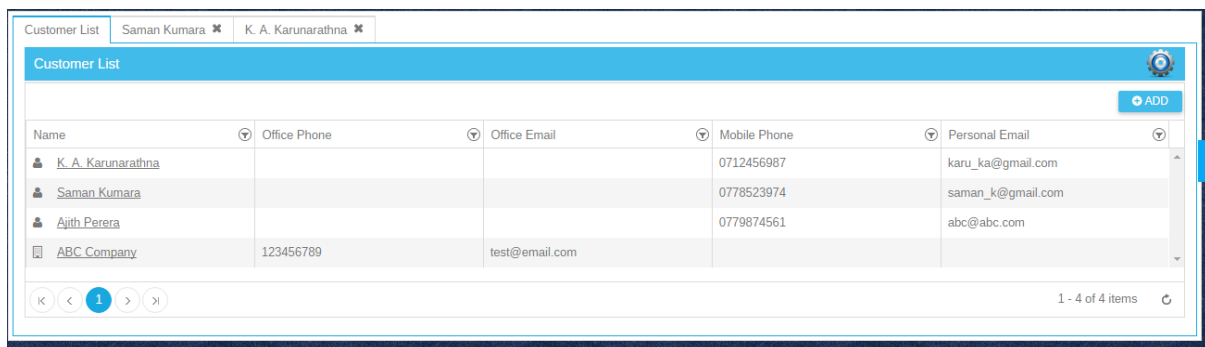
Figure 3.12 Login user interface

Menu

Successful login attempt will allow the user to enter the application landing page. main menu of the application would only appear the appropriate menu items as the user

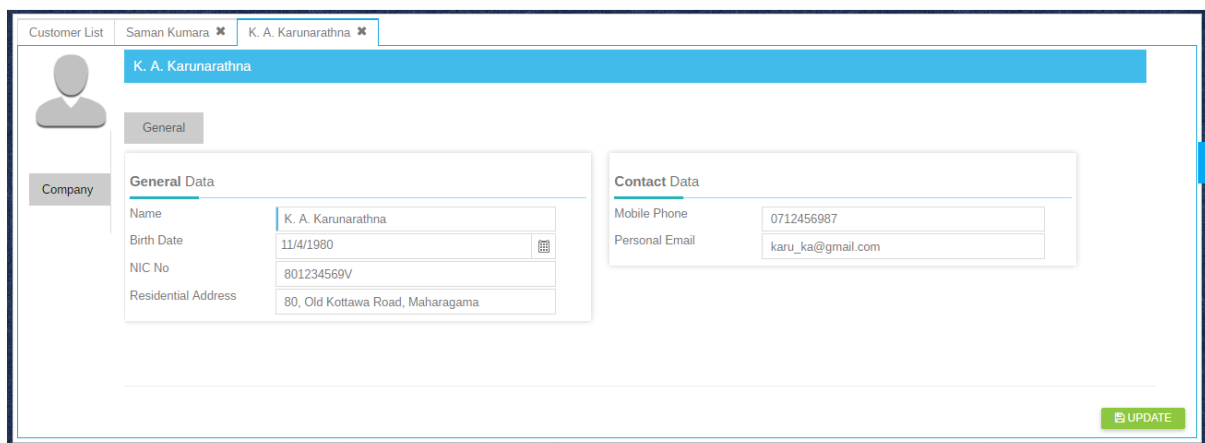
Customer Management

Entering customer information.



Name	Office Phone	Office Email	Mobile Phone	Personal Email
K. A. Karunaratna			0712456987	karu_ka@gmail.com
Saman Kumara			0778523974	saman_k@gmail.com
Ajith Perera			0779874561	abc@abc.com
ABC Company	123456789	test@email.com		

Figure 3.13 Customer list view



K. A. Karunaratna	
General	
General Data	
Name	K. A. Karunaratna
Birth Date	11/4/1980
NIC No	801234569V
Residential Address	80, Old Kottawa Road, Maharagama
Contact Data	
Mobile Phone	0712456987
Personal Email	karu_ka@gmail.com
UPDATE	

Figure 3.14 Customer detail view

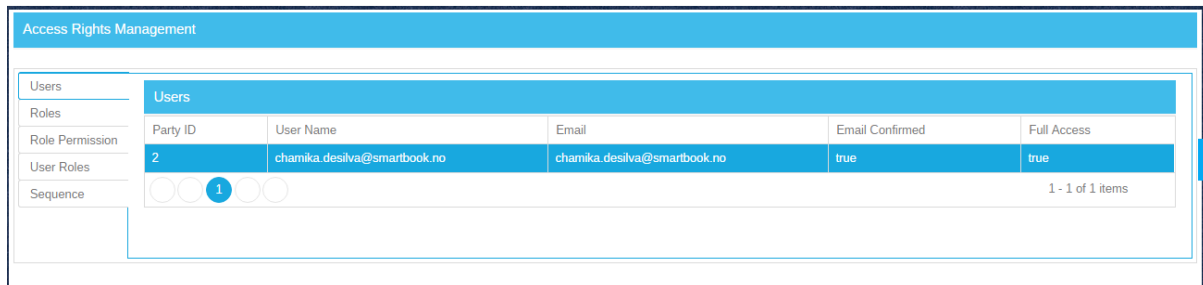
Billing Management

For managing payments and prepare invoices are handles in Billing menu. The user can enter charges such as meal charges and also update payments received from the client to the system and final invoicing is handled by the Billing Menu.

Report Center

Administration

User access rights management is with Administration menu.



The screenshot shows a web application interface titled "Access Rights Management". On the left is a sidebar menu with options: Users, Roles, Role Permission, User Roles, and Sequence. The "Users" option is selected. The main content area displays a table with the following data:

Party ID	User Name	Email	Email Confirmed	Full Access
2	chamika.desilva@smartbook.no	chamika.desilva@smartbook.no	true	true

Below the table, there are five circular pagination controls, with the second one (containing the number 1) being active. The text "1 - 1 of 1 items" is shown at the bottom right of the table area.

Figure 3.15 Access control view

UI Components

Set of components has been selected to gain the user friendliness, and for the benefit of good user experience.

- **Date Time Picker**

Date time pickers can avoid the confusion of user by restricting user to input only the correct format.

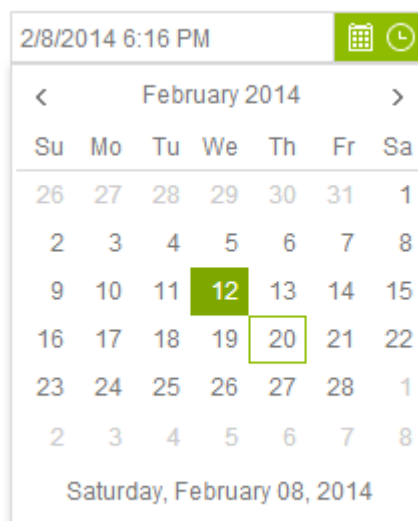


Figure 3.16 Date time picker

- **Numeric Textbox**

Number formats such as currency, percentage, unit of measure and quantities has different kind of presentation and validations. This UI component fulfil the above mentioned requirements.

PRICE:

\$30.00

PRICE DISCOUNT:

5 %

WEIGHT:

2.00 kg

CURRENTLY IN STOCK:

17.00

Figure 3.17 Numeric textbox

- **Notification**

Notification messages for the user interactions such as errors, warnings and information will be screened using this component.

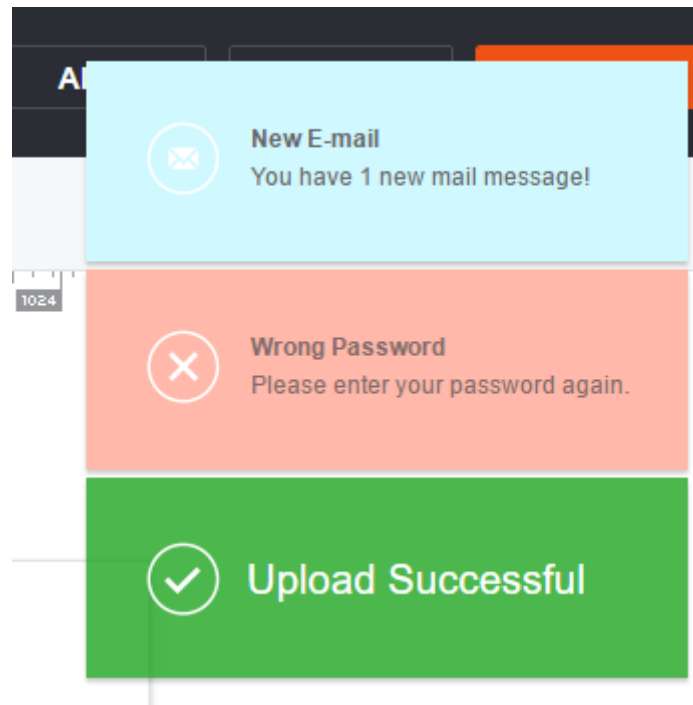


Figure 3.18 Notification partial view

Chapter 4: Implementation

4.1 Introduction

The Implementation Phase has one key activity. That is deploying the new system in its target environment. Supporting actions include training end-users and preparing to turn the system over to maintenance personnel. After this phase, the system enters the Operations and Maintenance Phase for the remainder of the system's operational life. Multiple-release projects require multiple iterations of the Implementation Phase one for each release.

4.2 Implementation Environment

Server side implementation has been done in the Azure virtual machine located in Netherlands, Western European Azure region.

Server Side Hardware

Category	Requirement	Actual
Random Access Memory	Minimum of 1 GB. More RAM will allow more files to be cached, reducing disk requests.	3.5 GB
Physical Memory	More that have allowed virtual paging to disk grows as well.	126 GB
Processor	Can be achieved significant performance improvements with a large processor caches and clock speeds.	Processor Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz, 2394 MHz, 1 Core(s), 1 Logical Processor(s)
Network	Increase the bandwidths will improve the traffic control with the application.	Microsoft Hyper-V Network Adapter 10.0 Gbps

Table 4.1 Server Side Hardware

Server Side Software

Category	Requirement	Actual
Operation System	Windows Vista SP1 and later (x86 and x64) Or Windows Server 2008 (x86 and x64) or later	Windows 10 Enterprise N x64
IIS Server	IIS 7.5 or later	IIS 10.0
SQL Server	2008 or later	2014
.Net Framework Client Runtime (CLR)	4.5 or later	4.6

Table 4.2 Server Side Software

Minimal requirements depend on the modern major browser requirements. Considering four major browsers (Chrome [22], Firefox [23], Edge [24], Opera [24]) it was declared the minimum client side requirements as follows.

Client Side Hardware

Category	Requirement
Random Access Memory	Minimum of 1 GB.
Physical Memory	Minimum 300 MB. More that have allowed browser cache to disk.
Processor	Pentium 4 or newer processor that supports SSE2.
Network	Minimum of 256 kbps. Increase the bandwidths will improve the user experience.
GPU	Display Driver Model (WDDM) 1.0 driver.
Display	Super VGA (800 x 600) or higher-resolution monitor with 256 colours.
Printer	According to client decided print quality.

Table 4.3 Client Side Hardware

Client Side Software

Category	Requirement
Browser	Chrome, Firefox, Safari, Opera, Edge
Graphics	Microsoft DirectX 9 or later with Windows

Table 4.4 Client Side Software

4.3 Development Tools

Visual Studio Enterprise 2015

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. [26]

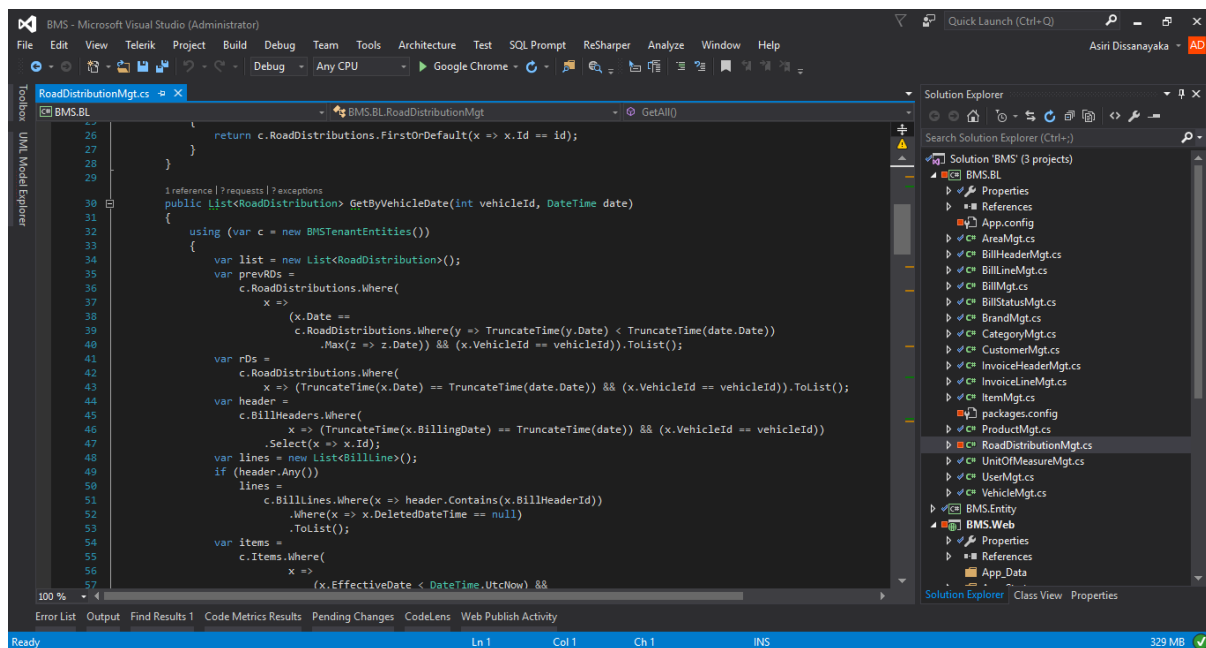


Figure 4.1 Visual studio 2015

C# 6

C# (pronounced "C sharp") is a programming language that is designed for building a variety of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented. The many innovations in C# enable rapid application development while retaining the expressiveness and elegance of C-style languages. Sixth version of C# has been used in the application. [12]

.Net Framework 4.6

.NET is the development platform used by millions of developers to create apps and services on any device and OS, with amazing performance and great developer productivity. [27]

SQL Server 2014

SQL Server is Microsoft's relational database management system (RDBMS). It is a full-featured database primarily designed to compete against competitors Oracle Database (DB) and MySQL. Like all major RDBMS, SQL Server supports ANSI SQL, the standard SQL language. However, SQL Server also contains T-SQL, its own SQL implementation. SQL Server Management Studio (SSMS) is SQL Server's main interface tool, and it supports 32-bit and 64-bit environments. SQL Server is sometimes referred to as MSSQL and Microsoft SQL Server. [28]

TortoiseSVN

TortoiseSVN is a really easy to use Revision control / version control / source control software for Windows. It is based on Apache™ Subversion (SVN)® [30]; TortoiseSVN provides a nice and easy user interface for Subversion. [28]

VisualSVN Server

VisualSVN Server allows you to easily install and manage a fully-functional Subversion server on the Windows platform. Thanks to its robustness, unbeatable usability and unique enterprise-grade features, VisualSVN Server is useful both for small business and corporate users. [29]

AnkhSVN

AnkhSVN is a Subversion Source Control Provider for Microsoft Visual Studio 2005-2013 and 2015. AnkhSVN provides Apache™ Subversion® source code management support to all

project types supported by Visual Studio and allows you to perform the most common version control operations directly from inside the Microsoft Visual Studio IDE. [30]

ReSharper

ReSharper is a popular developer productivity extension for Microsoft Visual Studio. It automates most of what can be automated in your coding routines. It finds compiler errors, runtime errors, redundancies, code smells, and possible improvements right as you type, suggesting intelligent corrections for them. ReSharper helps you explore code by visualizing the structure of files, type and style hierarchies, call and value chains, project dependencies. It allows you to instantly traverse your entire solution and jump right to the exact file and line that you are looking for, decompiling library code if necessary. [31]

SQL Tool belt

Red gate's SQL Tool belt contains the industry-standard products for SQL Server development, deployment, backup, and monitoring. Together, they make you productive, your team agile, and your data safe. Thousands of SQL Server professionals rely on the SQL Tool belt every day, because it's reliable, easy to use, and plugs in to the tools they already use for application development. [33]

Microsoft Office Professional Plus 2016

Suite of products developed by Microsoft Corporation that includes Microsoft Word, Excel, Access, Publisher, PowerPoint, and Outlook. Each program serves a different purpose and is compatible with other programs included in the package. The suite of programs is compatible with both the Windows and Macintosh operating system. Microsoft Office is the most common form of software used in the western world. [34]

Chrome Dev Tools

The Chrome Developer Tools (DevTools for short), are a set of web authoring and debugging tools built into Google Chrome. The DevTools provide web developers deep access into the internals of the browser and their web application. Use the DevTools to efficiently track down layout issues, set JavaScript breakpoints, and get insights for code optimization. [35]

4.4 Solution Architecture

Solution was designed with four layers with internal dependencies. When an input generated by the user presentation layer forward the details to the service layer as a request. Service layer process and validate the request. Request will be forwarded to the Business Logic Layer. Only Data layer will be allowed to communicate with the DBMS. Logic will be applied and data will be manipulated with the use of Data Layer and forward the result to the Service layer. Then the response will be received by the Presentation Layer from Service Layer. Cycle of request will be completed at that point. Solution architecture showing on figure 4.2

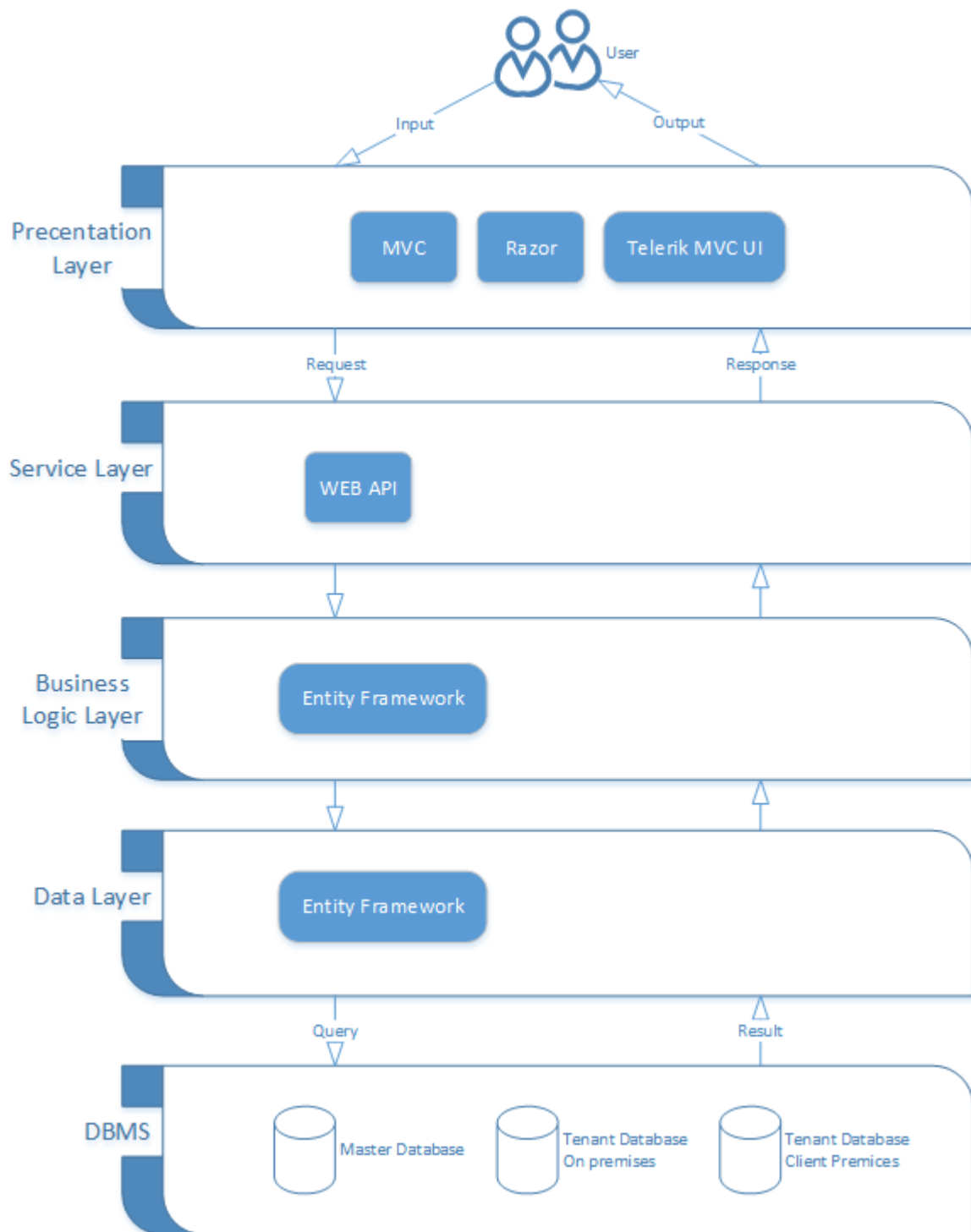


Figure 4.2 Solution architecture

Solution was named as ‘SalesAndInventorySystem’ because of the name of the project. It was decided to have three projects under application solution. Since it was decided to use the same host with both Presentation and Service layers, both will be integrated under web project ‘SalesAndInventorySystem.Web’ namespace and will be compiled as assembly

‘SalesAndInventorySystem.Web.dll’. ‘SalesAndInventorySystem.BL’ project contain the Business logic and the class library will be compiled in to ‘SalesAndInventorySystem.BL.dll’. ‘SalesAndInventorySystem.Entity.dll’ assembly will be the binary file of the Data layer which the class library namespace identified as ‘SalesAndInventorySystem.Entity’.

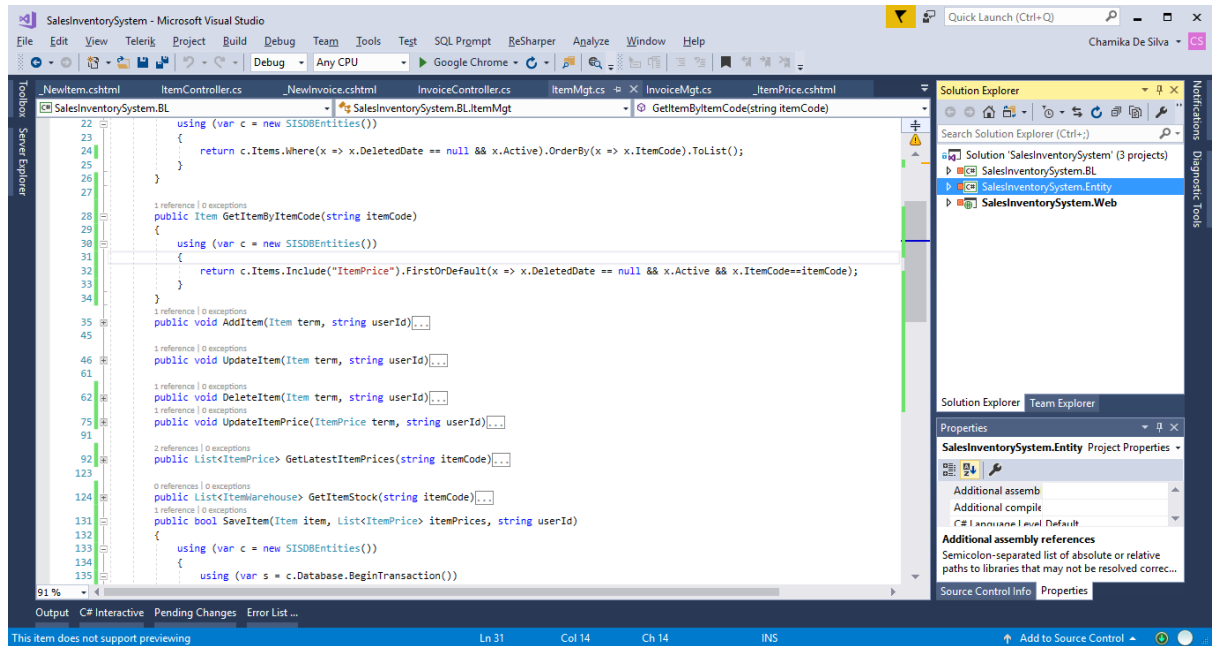


Figure 4.3 Projects in solution

4.5 Re-used existing code

Entity Framework

Entity Framework is Microsoft’s recommended data access technology for new applications. It is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. It eliminates the need for most of the data-access code that usually need to write. It is open source and contribution is allowed in GitHub [36]. Implemented in all three projects in order to use the major contributed Entity project.

ASP.NET Web API

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal

platform for building RESTful applications on the .NET Framework. Implemented in Web project. Open source code can be found in GitHub [37].

Telerik MVC UI/ Kendo-UI

It was chosen as the UI framework for the application. Kendo-UI is pure client side with JavaScript, CSS and resource files. Rich integration with the MVC structure is an advantage to the developer. Telerik MVC UI was built as helpers for the MVC applications to render Kendo components in server side. Web project contains the Kendo files and the Telerik MVC binaries.

jQuery [21]

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. Web project contains the files and freely distributed open source framework.

4.6 Major code segments

Simple CRUD Operation

Vehicle screen related code will be discussed under this topic.

- **Model**

Sample model class showing in figure 4.4

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SalesInventorySystem.Entity.SISDB
8 {
9     39 references
10     public partial class Item
11     {
12         4 references | 0 exceptions
13         public string CostUnitCode { get; set; }
14         4 references | 0 exceptions
15         public decimal CostPriceUnit { get; set; }
16         4 references | 0 exceptions
17         public decimal CostPrice { get; set; }
18         4 references | 0 exceptions
19         public string SellUnitCode { get; set; }
20         4 references | 0 exceptions
21         public decimal SellPriceUnit { get; set; }
22         4 references | 0 exceptions
23         public decimal SellPrice { get; set; }
24         4 references | 0 exceptions
25         public decimal WhseQty { get; set; }
26     }
27 }
```

Figure 4.4 Sample model class

- **View**

Sample view page showing on figure 4.5

```

1  @using Kendo.Mvc.UI
2
3  @{
4      ViewBag.Title = "Index";
5      Layout = "~/Views/Shared/_Layout.cshtml";
6  }
7  <h2>Items</h2>
8  @Html.Kendo().TabStrip()
9      .Name("ItemStrip")
10     .Items(i =>
11     {
12         i.Add().Text("New Product").LoadContentFrom("_NewItem", "Item");
13         i.Add().Text("Product List").LoadContentFrom("_Item", "Item");
14     })
15     .SelectedIndex(0)
16 }
17
18 <script type="text/javascript">
19     function error_handler(e) {
20         if (e.errors) {
21             var message = "Errors:\n";
22             $.each(e.errors, function (key, value) {
23                 if ('errors' in value) {
24                     $.each(value.errors, function() {
25                         message += this + "\n";
26                     });
27                 }
28             });
29             alert(message);
30         }
31     }
32 </script>

```

Figure 4.5 Sample view page

- **Partial View**

Sample partial view page showing on figure 4.6

```

1  @using Kendo.Mvc.UI
2  @using SalesInventorySystem.Entity.SISDB
3
4  @(Html.Kendo().Grid<ItemCategory>()
5      .Name("gItemCategory")
6      .Columns(columns =>
7      {
8          //columns.Bound(p => p.Id);
9          columns.Bound(p => p.CategoryCode);
10         columns.Bound(p => p.CategoryName);
11         columns.Bound(p => p.Description);
12         columns.Bound(p => p.Active);
13         columns.Command(command => command.Destroy()).Width(110);
14     })
15     .ToolBar(toolbar =>
16     {
17         toolbar.Create();
18         toolbar.Save();
19     })
20     .Editable(editable => editable.Mode(GridEditMode.InCell))
21     .Pageable(pageable => pageable
22         .Refresh(true))
23     .Navigatable()
24     .Sortable()
25     .Scrollable()
26     .DataSource(dataSource => dataSource
27         .Ajax()
28         .Batch(true)
29         .PageSize(20)
30         .ServerOperation(false)
31         .Events(events => events.Error("error_handler"))
32         .Model(model => model.Id(p => p.CategoryCode))
33         .Create("ItemCategory_Create", "ItemCategory")
34         .Read("ItemCategory_Read", "ItemCategory")
35         .Update("ItemCategory_Update", "ItemCategory")
36         .Destroy("ItemCategory_Destroy", "ItemCategory")
37     )
38 )
39 </script>...</script>

```

Figure 4.6 Sample partial view page

- **Controller**

Sample controller class showing on figure 4.7

```

13 {
14     0 references
15     public class ItemTypeController : Controller
16     {
17         // GET: ItemType
18         0 references | 0 requests | 0 exceptions
19         public ActionResult Index()
20         {
21             return View();
22         }
23         0 references | 0 requests | 0 exceptions
24         public ActionResult _ItemType()
25         {
26             return PartialView();
27         }
28         0 references | 0 requests | 0 exceptions
29         public ActionResult GetAll()
30         {
31             return Json(new ItemTypeMgt().GetAll(), JsonRequestBehavior.AllowGet);
32         }
33         0 references | 0 requests | 0 exceptions
34         public ActionResult ItemType_Read([DataSourceRequest] DataSourceRequest request)
35         {
36             return Json(new ItemTypeMgt().GetAll().ToDataSourceResult(request), JsonRequestBehavior.AllowGet);
37         }
38
39         [AcceptVerbs(HttpVerbs.Post)]
40         0 references | 0 requests | 0 exceptions
41         public ActionResult ItemType_Create([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)...
42
43         [AcceptVerbs(HttpVerbs.Post)]
44         0 references | 0 requests | 0 exceptions
45         public ActionResult ItemType_Update([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)...
46
47         [AcceptVerbs(HttpVerbs.Post)]
48         0 references | 0 requests | 0 exceptions
49         public ActionResult ItemType_Destroy([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)...
50     }
51 }

```

Figure 4.7 Sample controller class

- **Business logic**

Sample business logic class showing on figure 4.8

```

1 using ...
2
3 namespace SalesInventorySystem.BL
4 {
5     7 references
6     public class ItemTypeMgt
7     {
8         4 references | 0 exceptions
9         public List<ItemType> GetAll()
10         {
11             using (var c = new SISDBEntities())
12             {
13                 return c.ItemTypes.Where(x => x.DeletedDate == null).OrderBy(x => x.Id).ToList();
14             }
15         }
16
17         1 reference | 0 exceptions
18         public void AddItemType(ItemType term, string userId)
19         {
20             using (var c = new SISDBEntities())
21             {
22                 term.CreatedDate = DateTime.Now;
23                 term.CreatedUserId = userId;
24                 c.ItemTypes.Add(term);
25                 c.SaveChanges();
26             }
27         }
28
29         1 reference | 0 exceptions
30         public void UpdateItemType(ItemType term, string userId)...
31
32         1 reference | 0 exceptions
33         public void DeleteItemType(ItemType term, string userId)...
34     }
35 }

```

Figure 4.8 Sample business logic class

Chapter 5: Evaluation

5.1 Introduction

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Validation

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the software development life cycle (SDLC). If the software matches requirements for which it was made, it is validated.

Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

5.2 Testing Levels

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified. Testing separately is done just to make sure that there are no hidden bugs or issues left in the software.

Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updating etc.

System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

Functionality testing - Tests all functionalities of the software against the requirement.

Performance testing - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.

Security & Portability - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

Alpha testing - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.

Beta testing - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

5.3 Test Cases

Test Design & Execution Documents has been prepared for each functional area. Sample test document with results of the login screen as follows.

	Test Case	Result		Status
		Expected	Actual	

1	Click Login button with Username = "", Password = "", Remember me = uncheck	Refresh page with error messages for missing username and password.	Expected	Pass
2	Click Login button with Username = "{username}", Password = "", Remember me = uncheck	Refresh page with error messages for missing password.	Expected	Pass
3	Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = uncheck	Successfully login to landing page	Expected	Pass
4	Click Login button with Username = "{correct username}", Password = "{incorrect password}", Remember me = uncheck	Refresh page with error messages for incorrect username and password combination.	Expected	Pass
5	Click Login button with Username = "{incorrect username}", Password = "{correct password}", Remember me = uncheck	Refresh page with error messages for incorrect username and password combination.	Expected	Pass
6	Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = check. Then close the browser. Go to home page.	Load the landing page with the last login user details.	Expected	Pass
7	Logout if logged in. Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = uncheck. Then close the browser. Go to home page.	Load the login page.	Expected	Pass

Table 5.1 Sample Test Cases

5.4 User Acceptance Testing (UAT)

User acceptance testing is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

The goal of User Acceptance Testing is to assess if the system can support day-to-day business and user scenarios and ensure the system is sufficient and correct for business usage.

Application was hosted in the client site as an evaluation version for use of one month with real data. Proprietor with two accountants involved with the testing. At the end of the period system was accepted by the client with some new feature and change requests. Those will be mentioned under future development section.

User acceptance document can be found in the appendix.

Chapter 6: Conclusion

6.1 Introduction

This Chapter will contain the critical evaluation of the overall system with assessment of results of project, whether the project objectives were satisfied and lessons learnt during the course of the project.

6.2 Critical Assessment of the Project

On completion of the project it was assessed against the objectives. Accuracy of the calculations guaranteed with computing. With compared to the manual documents, system only require simple data entry effort to carry on the daily process and it increase the efficiency of the employees. User friendliness and the eye-catching interfaces were able to attract the users to the system. By using UI component such as date time picker, notifications messages and Numeric textboxes for various types of numeric inputs, application was able to reduce the margin of error which could made by user. System has been able to reduce most of the paperwork and also reduced the size of document storage in the organization. Company dashboard presentation with animated graphics and the very valuable information with trending, was helpful for the management to monitor the business on the fly.

For period of month, application was tested side by side along with the manual paper work and assured the reliability, accuracy and functional requirement fulfilment of the system. At the point of client acceptance, the system has been thoroughly tested by both the developer and the users. By hosting the web application on cloud, it has guaranteed the high availability of the system to the users. Additional costs for backups and maintenance costs also been reduced with the decision. Mobile device availability was not up to expected satisfaction level, although the responsive interface design has able to present most of the mobile experience. Mobile push notifications or SMS alerts would add a lot of value to the system.

6.3 Lessons Learnt

Your client will ask you for a Rolls Royce, but he actually needs a bicycle to solve his problem. It is hard to explain the customer that actually what he needs. Client may ask for

hundred features but, he will only use ten of them for his whole business. It is a responsibility of business analyser to stand with the client to do the necessary things.

Continuous updates regarding the system progress will make the customer happy and grow the relationship bond towards the entrepreneur.

When selecting the technology for the development it is better to use the well-established versions than the latest. Other than the unidentified bugs in the latest, lack of documentation on the internet also cost you more time to even identify the bug.

6.4 Future Developments

SMS/Email Alert system

Client expects to have an alert system with short text messages and email. Alert trigger points and the receiving group of users could be configured through system.

Mobile Version

Since everybody has a smartphone these days, proprietor pointed out the importance of a mobile app.

Reference

- [1] Wikipedia, "Functional_requirement," Wikimedia Foundation, Inc., 12 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/Functional_requirement. [Accessed 12 September 2016].
- [2] "QuickBooks Online," Intuit, [Online]. Available: <https://quickbooks.intuit.com/global/?Country=Sri+Lanka>.
- [3] "Functional Requirements Vs Non Functional Requirements," ReQtest, [Online]. Available: <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>.
- [4] Wikipedia, "System Design," Wikimedia Foundation, Inc., 14 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/Systems_design. [Accessed 14 September 2016].
- [5] "Agile Development Methodology," [Online]. Available: <https://www.pinterest.com/pin/304555993521699172/>.
- [6] Microsoft, "Microsoft – Official Home Page," Microsoft, 2016. [Online]. Available: <https://www.microsoft.com>. [Accessed 16 September 2016].
- [7] Microsoft, "Chapter 23: Designing Rich Internet Applications," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ee658083.aspx>. [Accessed 16 September 2016].
- [8] Microsoft, "Microsoft Azure: Cloud Computing Platform & Services," Microsoft, 2016. [Online]. Available: <https://azure.microsoft.com>. [Accessed 16 September 2016].
- [9] Wikipedia, "Cloud computing - Wikipedia, the free encyclopedia," Wikimedia Foundation, Inc., 2016. [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing. [Accessed 16 September 2016].

- [10 Microsoft, "Windows | Official Site For Microsoft Windows 10 OS, Laptops, PCs, Tablets & More," Microsoft, 2016. [Online]. Available: <https://www.microsoft.com/en-us/windows/>. [Accessed 16 September 2016].
- [11 Microsoft, "C#," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/kx37x362.aspx>. [Accessed 16 September 2016].
- [12 Microsoft, "ASP.NET | The ASP.NET Site," Microsoft by Neudesic, LLC., 2016. [Online]. Available: <http://www.asp.net>. [Accessed 16 September 2016].
- [13 Wikipedia, "Model–view–controller," Wikimedia Foundation, Inc., 22 January 2005. [Online]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Accessed 16 September 2016].
- [14 Wikipedia, "Object-oriented design," Wikimedia Foundation, Inc., 31 January 2007. [Online]. Available: https://en.wikipedia.org/wiki/Object-oriented_design. [Accessed 14 September 2016].
- [15 Wikipedia, "Database normalization," Wikimedia Foundation, Inc., 23 April 2014. [Online]. Available: https://en.wikipedia.org/wiki/Database_normalization. [Accessed 16 September 2016].
- [16 Google, "Download and install Google Chrome," Google, 2016. [Online]. Available: <https://support.google.com/chrome/answer/95346?co=GENIE.Platform%3DDesktop&hl=en-GB>. [Accessed 16 September 2016].
- [17 mozilla.org, "Firefox 48.0.2 System Requirements," mozilla.org, 2016. [Online]. Available: <https://www.mozilla.org/en-US/firefox/48.0.2/system-requirements/>. [Accessed 16 September 2016].
- [18 Microsoft, "Microsoft Edge requirements and language support," Microsoft, 2016. [Online]. Available: <https://technet.microsoft.com/en-us/itpro/microsoft-edge/hardware-and-software-requirements>. [Accessed 16 September 2016].

- [19] Opera, "Opera system requirements," Opera Software ASA, 2016. [Online]. Available: <https://www.opera.com/download/requirements/>. [Accessed 16 September 2016].
- [20] Microsoft, "Visual Studio - Microsoft Developer Tools," Microsoft, 2016. [Online]. Available: <https://www.visualstudio.com/>. [Accessed 16 September 2016].
- [21] Microsoft, ".NET - Powerful Open Source Cross Platform Development," Microsoft, 2016. [Online]. Available: <https://www.microsoft.com/net/>. [Accessed 16 September 2016].
- [22] Microsoft, "SQL Server 2016 | Microsoft," Microsoft, 2016. [Online]. Available: <https://www.microsoft.com/en-us/cloud-platform/sql-server>. [Accessed 16 September 2016].
- [23] THE TORTOISESVN TEAM, "Home · TortoiseSVN," THE TORTOISESVN TEAM, 2016. [Online]. Available: <https://tortoisesvn.net/>. [Accessed 16 September 2016].
- [24] The Apache Software Foundation, "Apache Subversion," The Apache Software Foundation, 2016. [Online]. Available: <https://subversion.apache.org/>. [Accessed 21 September 2016].
- [25] VisualSVN Limited, "VisualSVN Server | Subversion Server for Windows," VisualSVN Limited, 2016. [Online]. Available: <https://www.visualsvn.com/server/>. [Accessed 16 September 2016].
- [26] CollabNet, Inc., "ankhsvn: Subversion Support for Visual Studio," CollabNet, Inc., 2016. [Online]. Available: <https://ankhsvn.open.collab.net/>. [Accessed 16 September 2016].
- [27] JetBrains s.r.o., "ReSharper," JetBrains s.r.o., 2016. [Online]. Available: <https://www.jetbrains.com/resharper/>. [Accessed 16 September 2016].
- [28] Red Gate Software Ltd, "SQL Toolbelt – Everything you need to work with SQL Server," Red Gate Software Ltd, 2016. [Online]. Available: <http://www.red-gate.com/products/sql-development/sql-toolbelt/>. [Accessed 16 September 2016].
- [29] Microsoft, "Microsoft Office Professional 2016," Microsoft, 2016. [Online]. Available: <https://products.office.com/en/professional>. [Accessed 16 September 2016].

- [30] Google, "Chrome DevTools Overview - Google Chrome," Google, 2016. [Online].
] Available: <https://developer.chrome.com/devtools>. [Accessed 16 September 2016].
- [31] Microsoft, "aspnet/EntityFramework6," GitHub, Inc., 2016. [Online]. Available:
] <https://github.com/aspnet/EntityFramework6>. [Accessed 17 September 2016].
- [32] Microsoft, "WebApiContrib/WebAPIContrib," GitHub, Inc., 2016. [Online]. Available:
] ASP.NET Web API . [Accessed 17 September 2016].
- [33] JQuery, "JQuery," The jQuery Foundation, 2016. [Online]. Available:
] <https://jquery.com/>. [Accessed 16 September 2016].
- [34] Microsoft, "Install SQL Server 2016 from the Installation Wizard (Setup)," Microsoft,
] 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ms143219.aspx>.
[Accessed 21 September 2016].
- [35] Wikipedia, "Rational Unified Process," Wikimedia Foundation, Inc., 2004. [Online].
] Available: https://en.wikipedia.org/wiki/Rational_Unified_Process. [Accessed 14
September 2016].
- [36] Wikipedia, "QuickBooks," Wikimedia Foundation, Inc., 22 March 2006. [Online].
] Available: <https://en.wikipedia.org/wiki/QuickBooks>. [Accessed 12 09 2016].
- [37] Wikipedia, "Non-functional requirement," Wikimedia Foundation, Inc., 16 March 2007.
] [Online]. Available: https://en.wikipedia.org/wiki/Non-functional_requirement.
[Accessed 13 October 2016].
- [38] Intuit, "Intuit Home Page," Intuit Inc., 12 October 2016. [Online]. Available:
] <http://global.intuit.com/row/index.jsp>. [Accessed 12 October 2016].
- [39] Wikipedia, "Systems development life cycle," Wikimedia Foundation, Inc., 3 April 2006.
] [Online]. Available: https://en.wikipedia.org/wiki/Systems_development_life_cycle.
[Accessed 14 September 2016].
- [40] Progress Sitefinity, "ASP.NET MVC Controls Based on jQuery," Progress Software
] Corporation, 2016. [Online]. Available: <http://www.telerik.com/aspnet-mvc>. [Accessed
16 September 2016].

- [41] Progress Software Corporation, "Kendo UI HTML Framework," Progress Sitefinity, 2016. [Online]. Available: <http://www.telerik.com/kendo-ui>. [Accessed 16 September 2016].
- [42] Mark Otto, Jacob Thornton, "Bootstrap · The world's most popular mobile-first and responsive front-end framework," Twitter, Inc., 2016. [Online]. Available: <http://getbootstrap.com/>. [Accessed 16 September 2016].
- [43] D. Gandy, "Font Awesome, the iconic font and CSS toolkit," Dave Gandy, 2016. [Online]. Available: <http://fontawesome.io/>. [Accessed 16 September 2016].
- [44] Microsoft, "Editions and Components of SQL Server 2016," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ms144275.aspx>. [Accessed 21 September 2016].
- [45] Microsoft, "Editions and Components of SQL Server 2016," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ms144275.aspx>. [Accessed 21 September 2016].
- [46] Microsoft, "Features Supported by the Editions of SQL Server 2016," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/cc645993.aspx>. [Accessed 21 September 2016].
- [47] Microsoft, "Instance Configuration," Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ms143531.aspx>. [Accessed 21 September 2016].

Appendix A : System Documentation

A.1 Enable IIS Server

IIS or Internet Information Services is a technology included in Windows Server, which provides a security-enhanced, easy-to-manage web server for developing and reliably hosting web applications and web services. This enables organizations to deliver rich, web-based experiences. IIS is not turned on by default on the Windows desktop editions. You have to turn it on.

Let us see now how to enable or turn on Internet Information Services on Windows 10. The procedure remains the same as it was in earlier editions.

Press Windows with R key in combination to bring up the 'Run' dialog box. In it, type 'appwiz.cpl' and hit the 'OK' button.

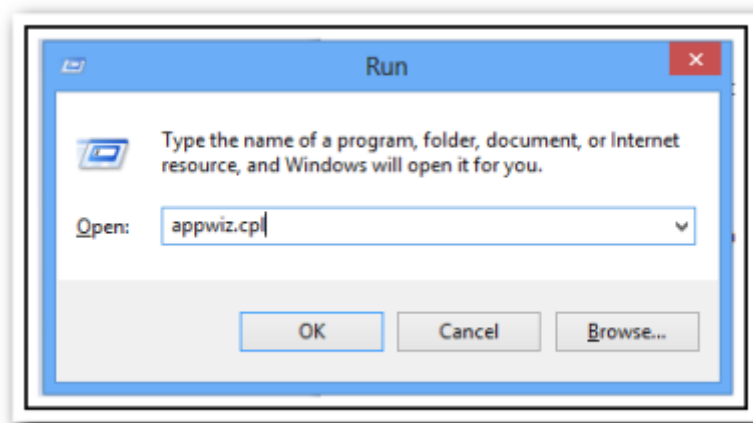


Figure A.6.1 IIS install screen 01

You will be directed to 'Programs and Feature' screen of the 'Control Panel'. There, in the left pane you will find a link to Turn Windows features on/off. Click on that link.

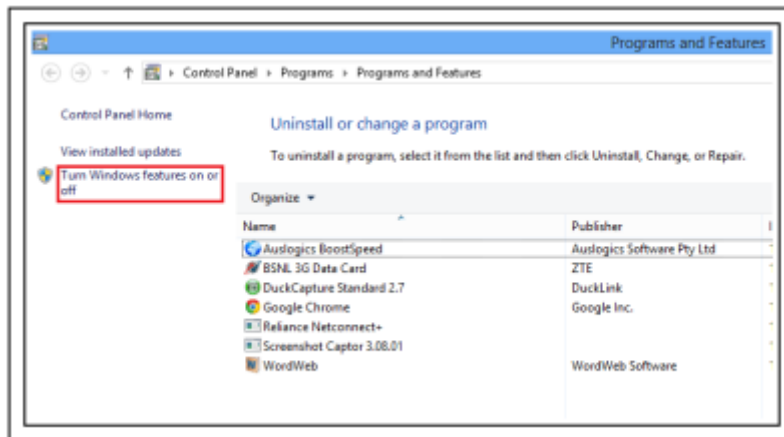


Figure A.6.2 IIS install screen 02

Now, a new 'Windows Features' screen should be visible to you. Scroll and search for the Internet Information Services option and select its check box, to turn on the feature.

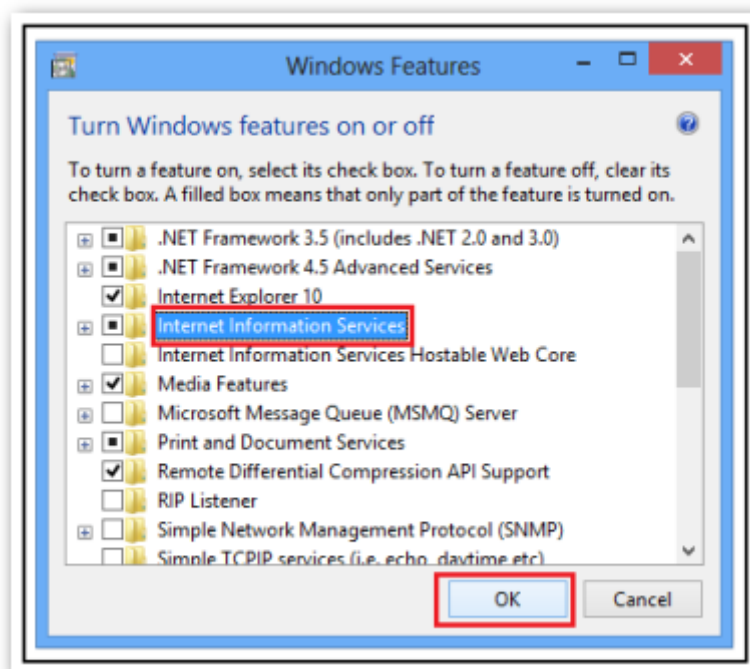


Figure A.6.3 IIS install screen 03

At this point you can leave settings as default and say OK, but you can certainly expand it and check the other components as well, if you are interested in selecting all or deselecting some.

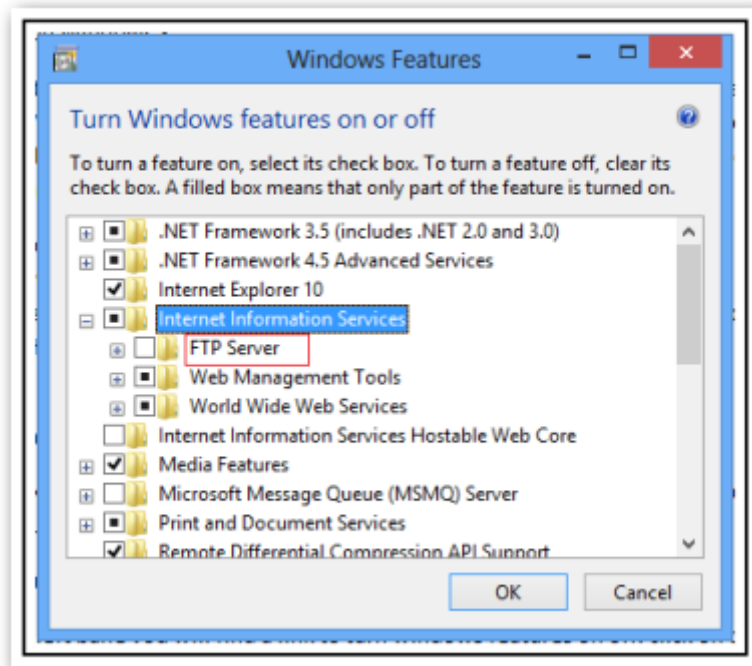


Figure A.6.4 IIS install screen 04

Once you have configured the desired options as well, hit the 'OK' button and wait patiently for few minutes to allow Windows Features to search for required files, install them and apply the changes.

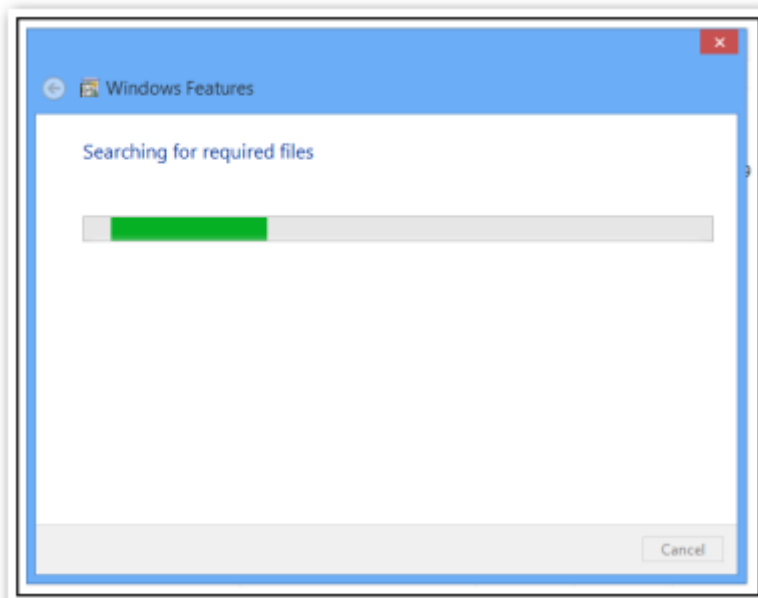


Figure A.6.5 IIS install screen 05

That is all. You will have now successfully installed the new version of IIS Internet Information Services on Windows 8, you will see a new localhost start page. Check out the screen-shot of below.



Figure A.6.6 IIS install screen 06

This is how you can enable or turn on Internet Information services (IIS) on Windows 10.

A.2 Install MSSQL Server

Full Installation instruction of the MSSQL Server 2016 can be found on the internet [39]. After that you can restore the two backup databases in to the server.

A.3 Host Application

Create a new web application in IIS server. Copy the provided host content in to the new host folder. Update the 'web.config' file in the root directory with the database connection strings.

Use the provided admin username and password and configure your system.

Appendix B : Design Documentation

B.1 Use case Diagrams

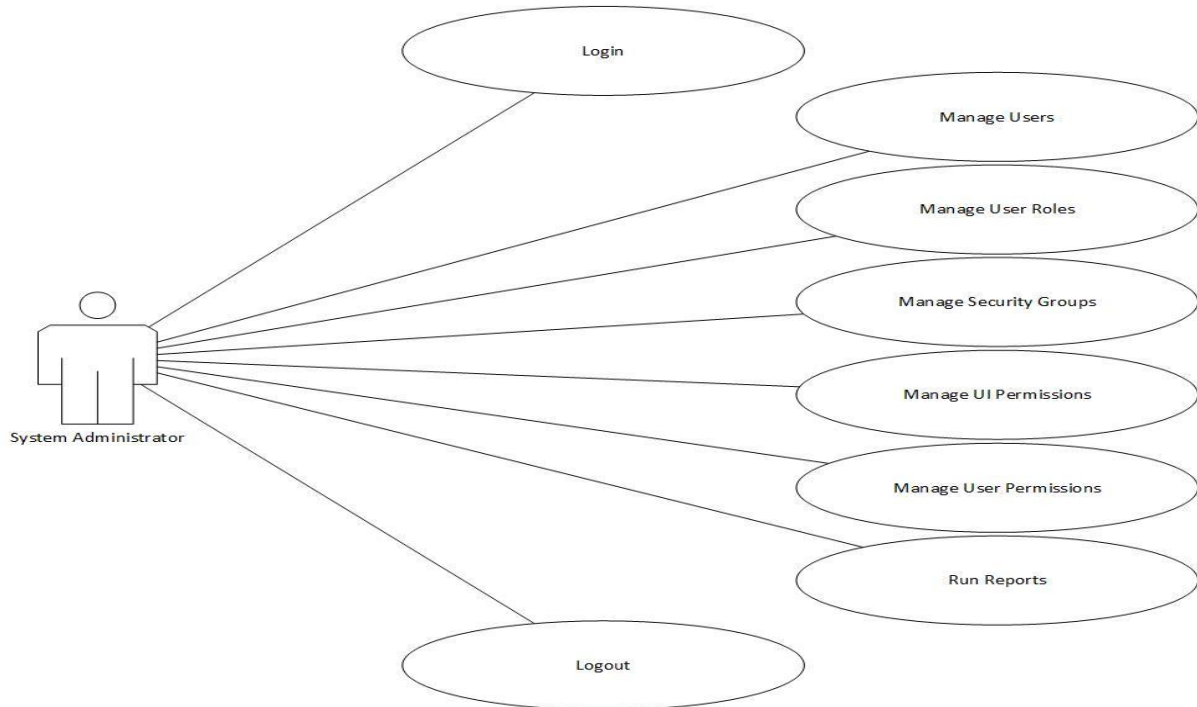


Figure B.1 System administrator use case

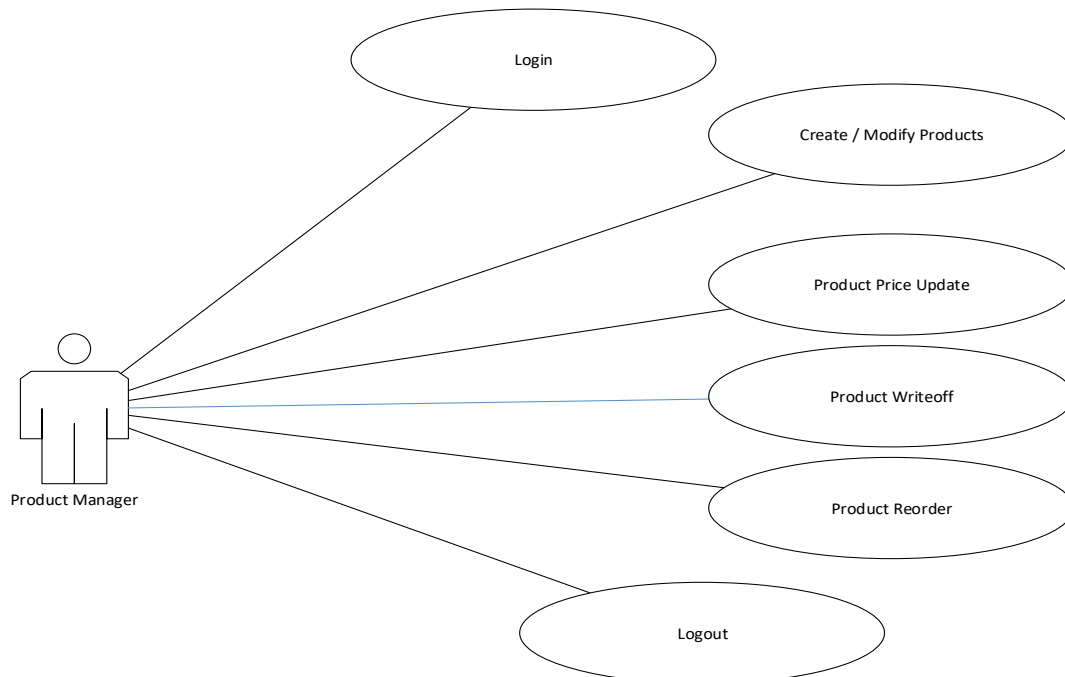


Figure B.2 Product manager use case

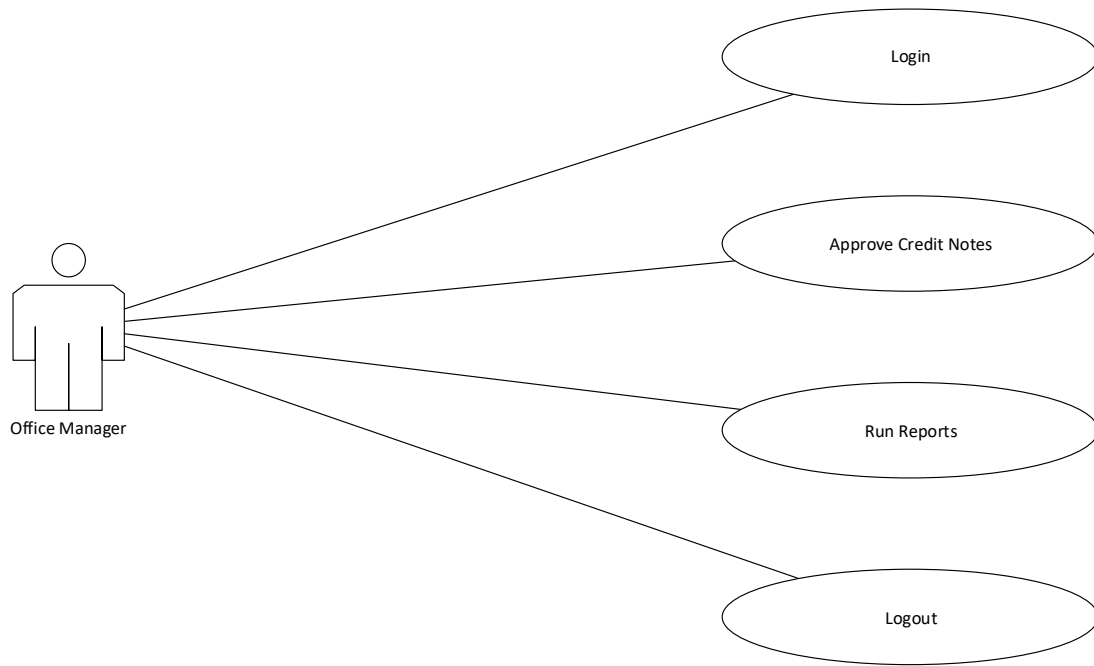


Figure B.3 Office manager use case

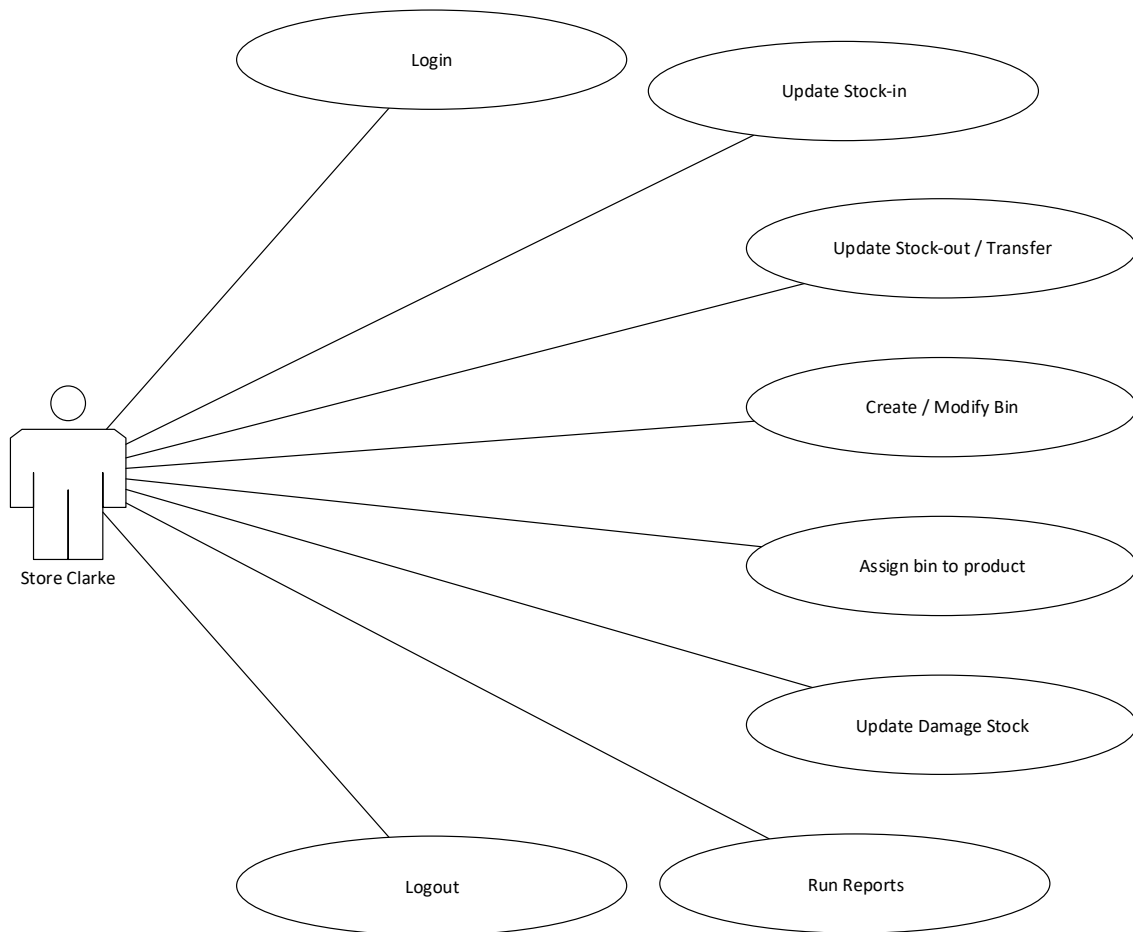


Figure B.4 Stores clerk use case

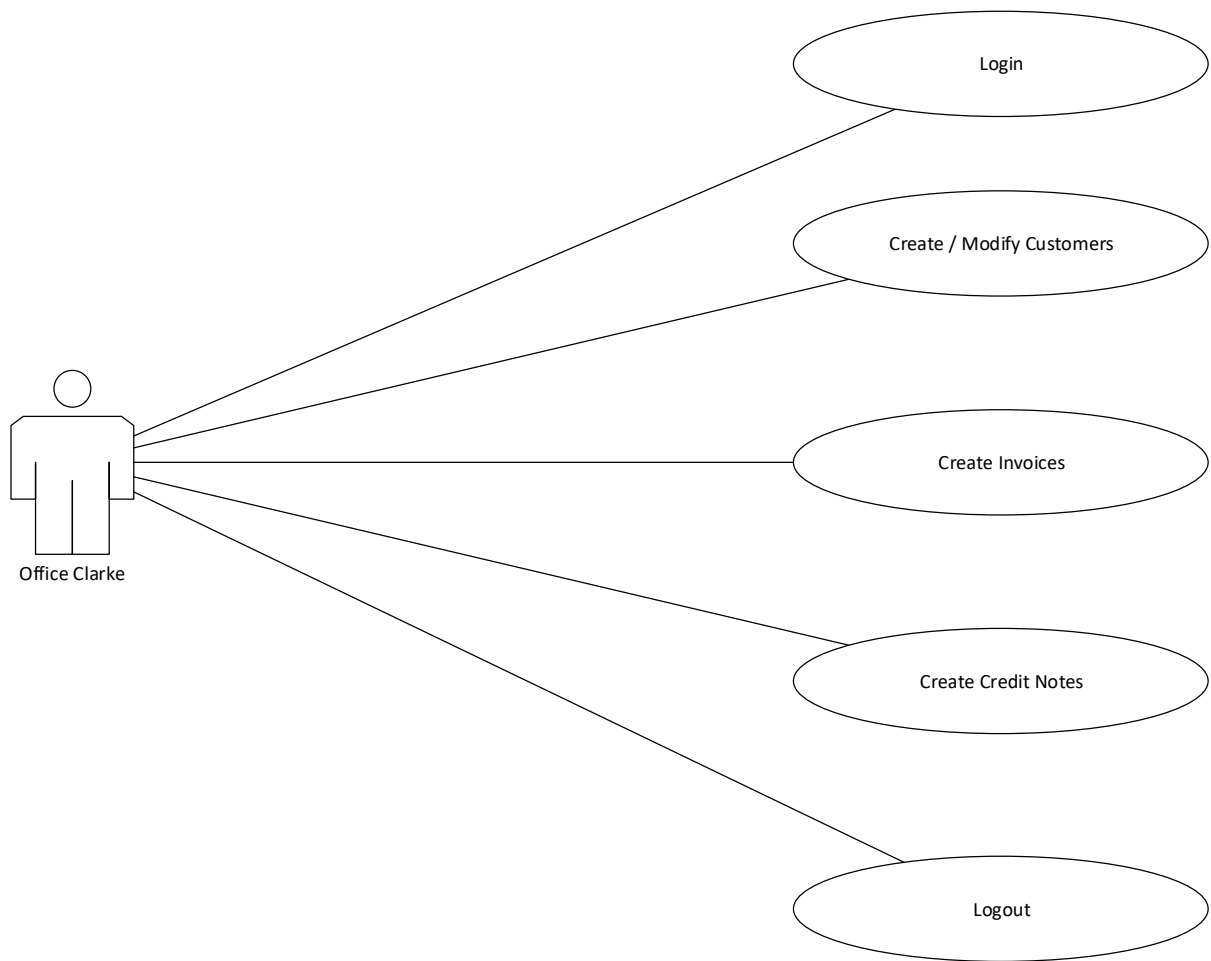


Figure B.5 Office clerk use case

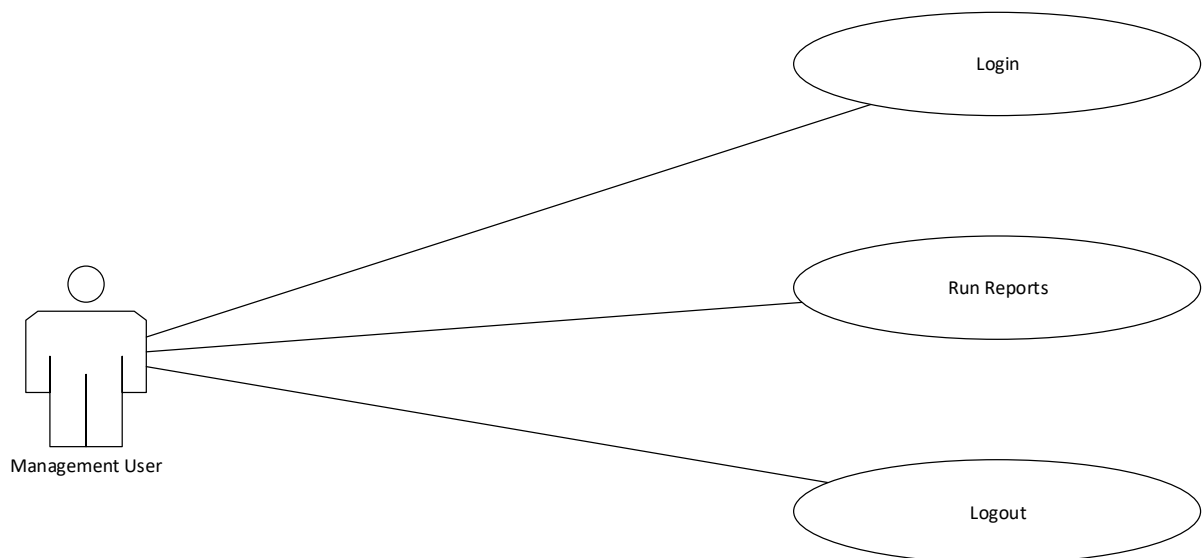


Figure B.6 Management user use case

B.2 ER Diagrams

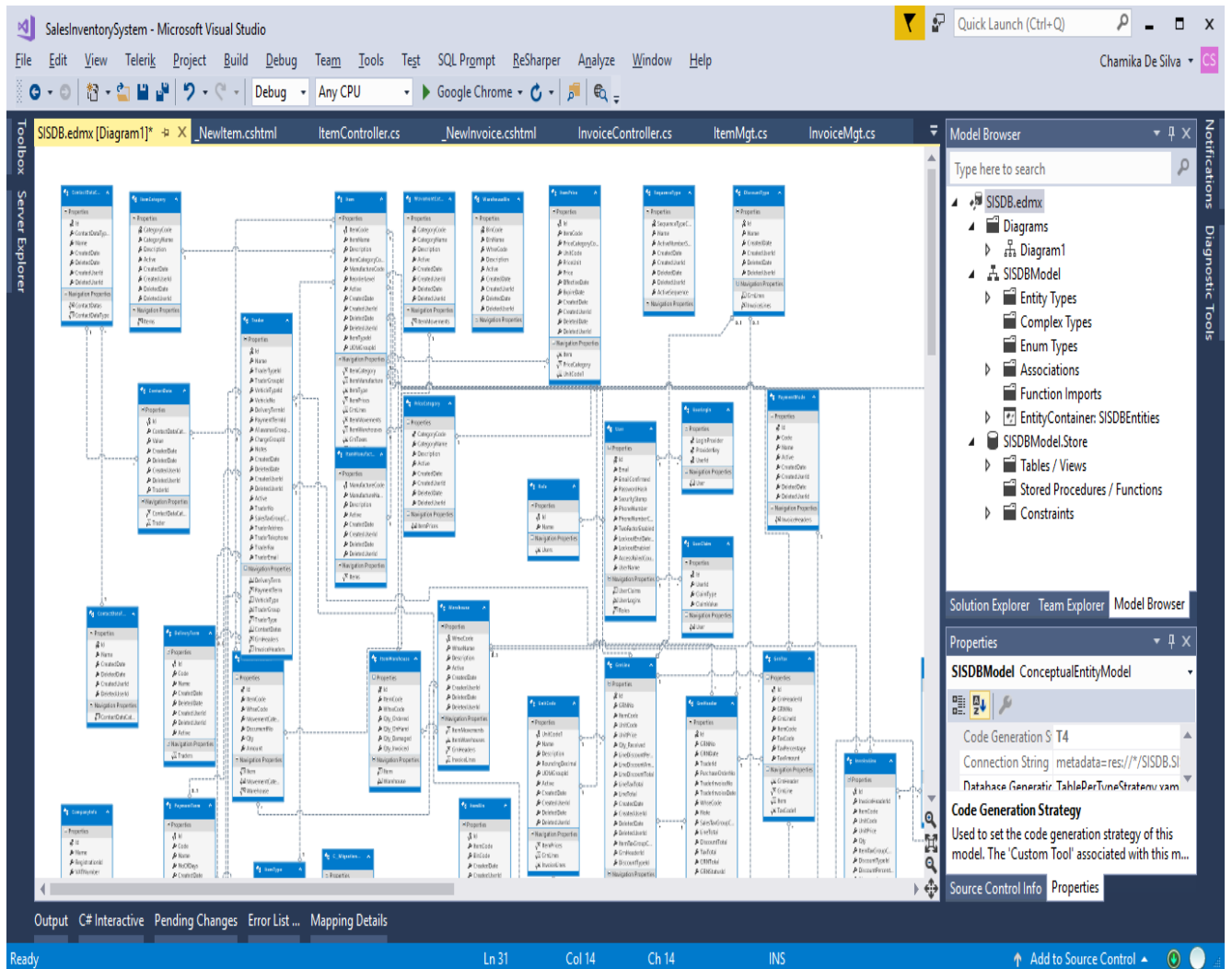


Figure B.6.7 Entity relationship diagram

B.3 Activity Diagrams

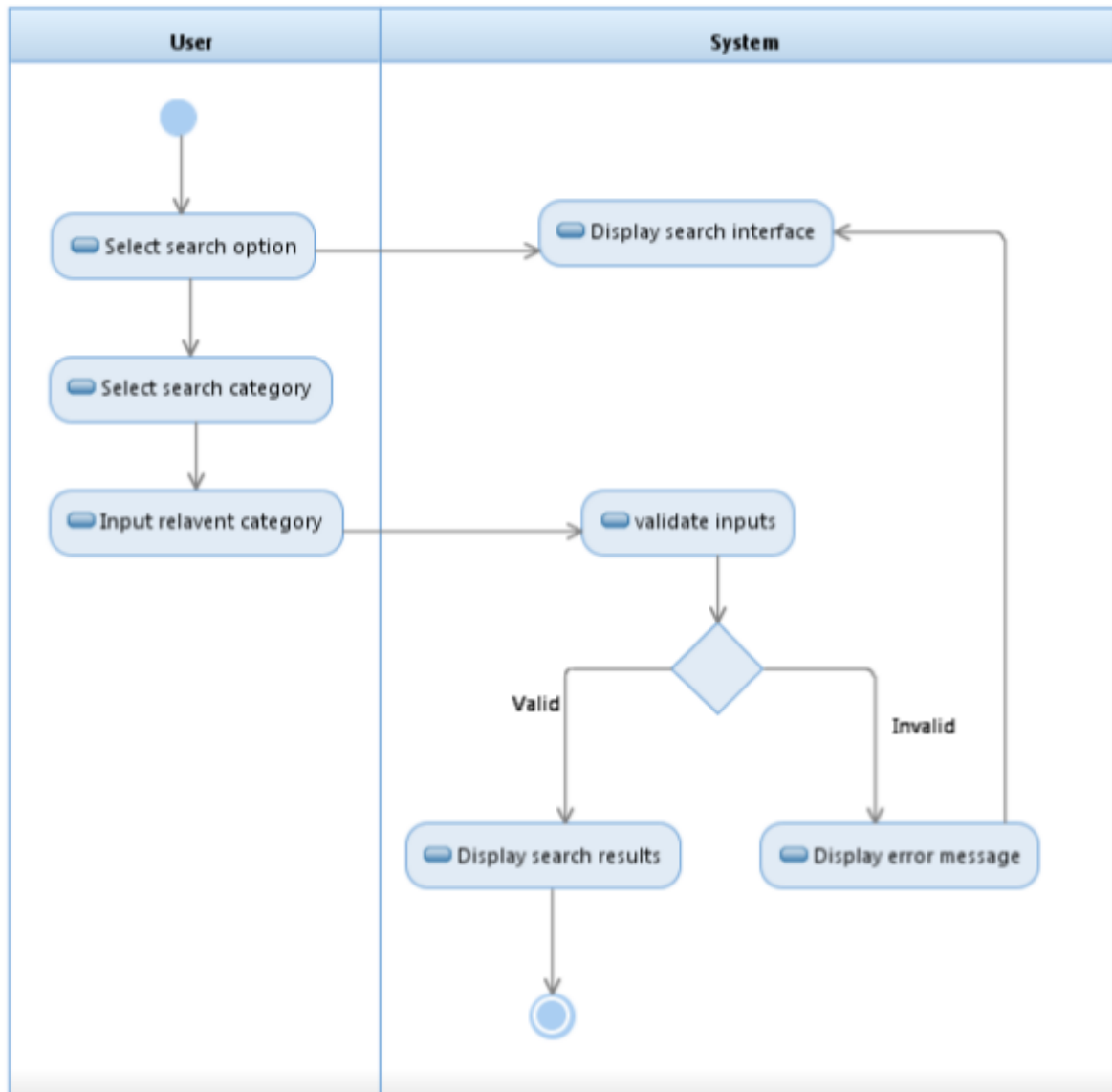


Figure B.6.8 Activity diagram for search

B.4 Sequence Diagrams

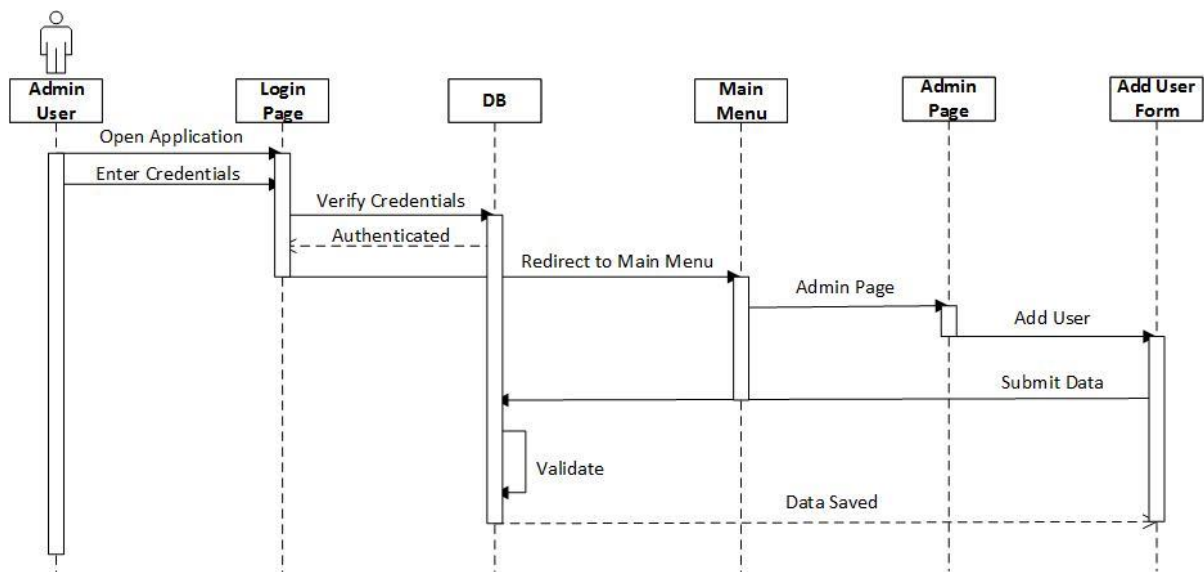


Figure B.6.9 Sequence diagram for add new user

Appendix C : User Documentation

C.1 Common Master Data Grid

All the screens in with the data grid has the same type of behaviour. Following mentioned grid functionality is almost same throughout the application. Add, Save and cancel buttons have placed on the top of the data grid. All the data editing will happen in the grid as in cell editing. Last column of the grid will contain the delete button for each and every row of data. There will be a confirmation message for deleting a record.

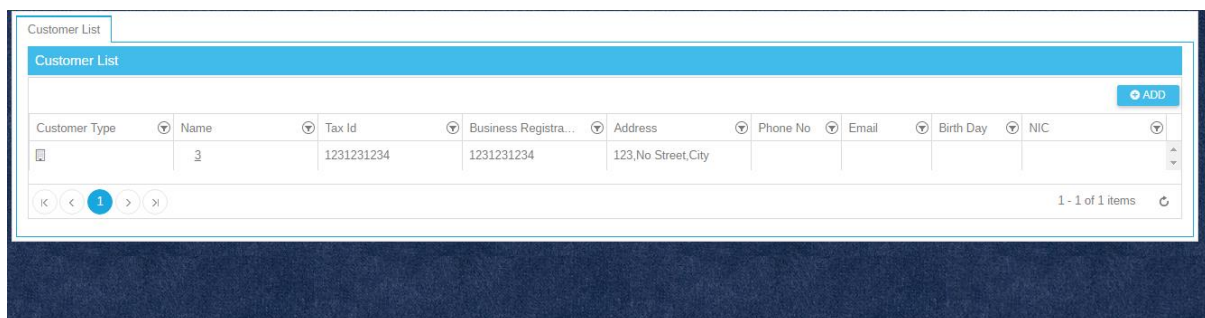


Figure C.1 Common master data screen

Every column will have filter icon on the right corner of the heard cell. It will open the filter menu for the column. Filter criteria and value can be updated first and click the filter button for apply it. The applied filter can be cleared by the clear button.

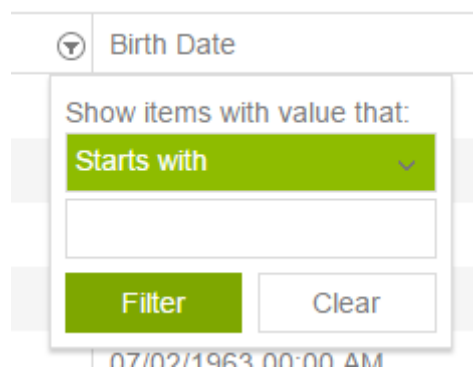


Figure C.2 Grid filter

Sorting can be done by clicking on the column heading. First time for sort ascending and then for descending and again to clear sorting.

	Order Date ^
	04/07/1996
	05/07/1996

Figure C.3 Grid sorting

Add button will add an empty row for the grid and can be filled cell by cell and can be saved by the save button o top at last. Specific data types will display specific types of data editing components in cells as follows.

	From Date	Thi
	7/13/2016	
	7/13/2016	

July 2016

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Thursday, September 22, 2016

Figure C.4 Date editor

Units In Stock
39.00 ^

Figure C.5 Number editor

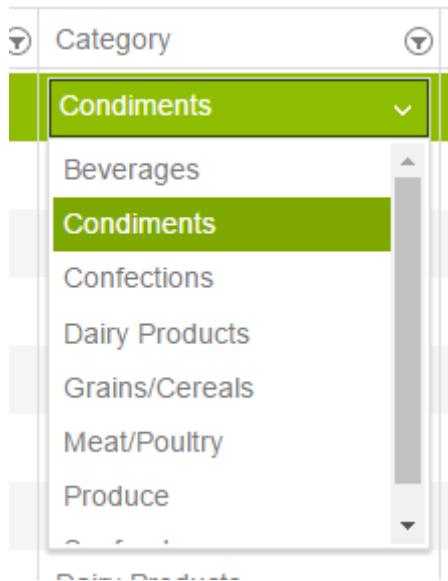


Figure C.6 Dropdown select list



Figure C.7 Boolean editor

The data grid allows to save multiple changes at once as a batch update. User can make changes in random cells and the changes will indicate with a red mark on the top left of the cell. Click the save button will save all the changes to database.

ProductName	Unit Price	Units In Stock	Discontinued
Chai	\$18.00	38	false
Chang	\$18.00	17	true
Aniseed Syrup1	\$10.00	12	false
Chef Anton's Cajun Seasoning	\$22.00	53	true

Figure C.8 Grid batch edit

Footer of the grid will contain the paging section on the left. Navigations through the pages of data grid facilitates here.



Figure C.9 Grid footer

Right side of the footer will display the number of records a range in selected page. Refresh button next to the message will refresh the current selection of data by reading from the database.

For Add Records using a common popup layout.

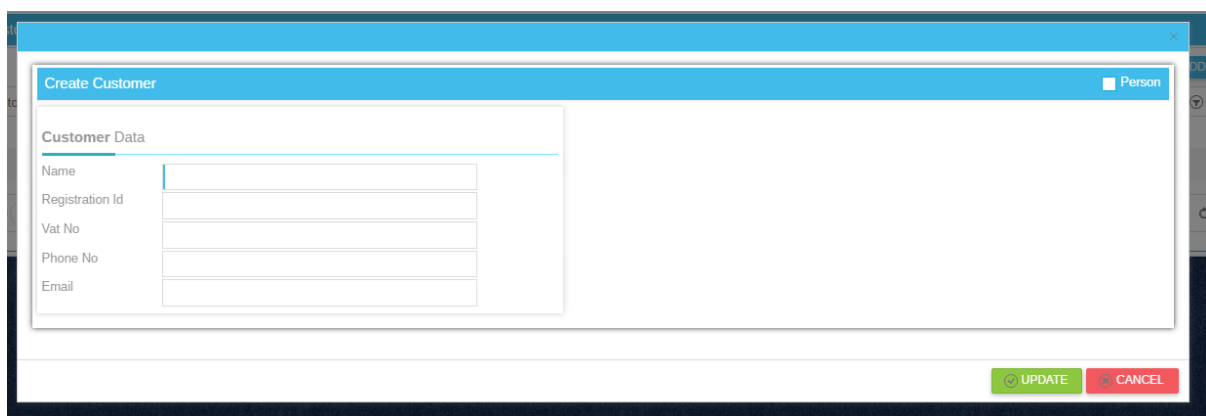


Figure C.6.10 Common popup layout

C.2 Customer Management

Customer management consists of two main functions.

Create Customer

The system made adding a customer as simple as possible. Customer adding screen showing on figure C.11

Customer Management

New Customer
Customer List

Save & Update
Clear All

Name
Vehicle Type
Select Vehicle Type
Vehicle No
Address
Telephone
Fax
Email

Trader Group
Select Group
Delivery Term
Select Delivery Term
Payment Term
Select Payment Term
Sales Tax Group
Select Tax Group
Note

Figure C.6.11 Add new customer view

Modify & view customer

There is a view with listing all customers (figure C.12). By selecting on a customer code, will be redirecting to view information on the customer and can modify if necessary. Also will display invoice history under the customer and debtors and payment history. (figure C.13, figure C.14, figure C.15)

Customer Management

New Customer
Customer List

Drag a column header and drop it here to group by that column

Trader No	Name	Trader Group Id	Vehicle Type Id	Vehicle No	Active	Credit Total	Debit Total	Balance
CUS000002	Akila Perera	Cash	Car	KA-5847	true	0	0	0
CUS000003	Jayantha Perera	Cash	Van	56-2565	true	0	0	0
CUS000004	Duminda Rathnayaka	Cash	Car	CAJ-8977	true	0	0	0

1
1 - 3 of 3 items

Figure C.6.12 Customer list view

Customer Management

New CustomerCustomer ListCUS000002 x

Customer InfoInvoice HistoryPayment History

Save & Update

NameAkila Perera

Vehicle TypeCar

Vehicle NoKA-5847

Address

Telephone

Fax

Email

Active☒

Trader GroupCash

Delivery TermPICKUP

Payment TermCASH

Sales Tax GroupVATE

Note

Credit Total148.50

Debit Total148.50

Balance Amount0.00

Figure C.6.13 Customer view page 01

Customer Management

New CustomerCustomer ListCUS000002 x

Customer InfoInvoice HistoryPayment History

Drag a column header and drop it here to group by that column

Invoice No	Invoice Date	Invoice Type Id	Invoice Status Id	Purchase Orde...	Due Date	Payable Total	Paid Total
INV000018	11/28/2017	Invoice	Invoked		11/28/2017	148.5	0

1 - 1 of 1 items

Figure C.6.14 Customer invoice history

Customer Management

New CustomerCustomer ListCUS000002 x

Customer InfoInvoice HistoryPayment History

Below30	Above30 To60	Above60 To90	Over90
0	0	0	0

Drag a column header and drop it here to group by that column

Payment No	Created Date	Payment Type Id	Payment Mode Id	Payment Reference	Payment Total
PAY000004	11/28/2017	Customer Pay	Cheque	no-125487	148.5

1 - 1 of 1 items

Figure C.6.15 Customer debtors and payment history

82

C.3 Invoicing

Invoicing screen contains two main sections. Invoice header contains the Invoice No, Date of invoice and whether the invoice had been finalized or not. The grid will become editable on user clicks on a cell. Date can be selected from the date picker and there is a checkbox for finalize field. Invoice number can be entered as a plain text. You can edit multiple records and save with the save changes button in top as a batch. Add button will allow you to add new Invoice. When you select a row from the header, the related lines will be loaded automatically to the lines grid. Delete button can be found on each row of data.

Invoice lines grid have a dropdown for select an item. Accordingly, no of cases and units can be inserted as numbers. Same pattern of add, save and delete buttons placed in this grid as well as invoice header. When finalize the invoice the lot of items will be added in to the stock for sales.

Figure C.6.16 Invoice header details

W/H	Item Code	Unit	Unit Price	Qty	Disc %	Disc Amt	Disc Total	Tax Total	Line Total
SS	TYT0W30	EACH	2585	1	0	0	0	0	2585
OS	TYT0W20	EACH	2750	1	0	0	0	0	2750
OS	HRXSAE50	EACH	148.5	1	0	0	0	0	148.5

Line Total	5,483.50
Disc Total	0.00
Tax Total	0.00
Paid	0.00
Balance	5,483.50

Figure C.6.17 Invoice lines and totals

Appendix D : Management Reports

D.1 Invoice Report

IMPERIAL AUTO CARE				
# 44, Galle Road, Dehiwala. Tel : 2 713 767				
Invoice No	17588	Invoice Date	28-June-2017	
Vehicle No	07-0985			
Customer Name	Mr Rohan jayawardhana			
Address	35 A Rajamahavihara rd Pitakotte			
Next Service	40792			
Particulars	Quantity	Unit Price	Discount	Total Price
Full Service Charge (Car)	1	1990	190	1800
Hyrax - Supra(SAE 40) 4L	1	2640	0	2640
SUB TOTAL				4,440.00
TOTAL DISCOUNT				0.00
TAX 0.00 %				0.00
GRAND TOTAL				4,440.00

Figure D.6.18 Invoice report

IMPERIAL AUTO CARE

44, Galle Road, Dehiwala. Tel : 2 713 767

Printed On 11/5/2017

Periodic Stock Report

Category	Company	ID	Name	TotalValue	OnHand
Air Freshner					
	Asia global traders				
		709	Exotica scent-mix	2,280.00	3
		710	Exotica scent a vent	2,480.00	2
		711	Exotica Zeroder	3,400.00	5
		712	Exotica palm tree	560.00	2
		713	Exotica x-card	280.00	1
	C.S.R Marketing				
		730	Air freshener CSR-Armorall	200.00	2
	godrej				
		438	godrej Stella Car Air Freshner	9,200.00	40
		460	godrej aer Twist air Freshner	2,396.00	4
		461	godrej Stella Matic Set	0.00	0
		524	godrej refil	1,240.00	4
		633	godrej daily Freshness	3,029.00	13
		715	godrej Car perfume	0.00	0
		731	Godrej aer pocket	4,800.00	48
	STC				
		702	STC Tokyo	150.00	1
		756	STC auto fresh air freshner	0.00	0
		781	STC- pandora air freshner	0.00	12
	Test				
		745	D-Fresh	2,220.00	6
C.S.R Marketing					
	C.S.R Marketing				
		563	Wash & Wax Detailer Spray	750.00	1
		634	STP Tuff Stuff Stain Remover	750.00	1

Figure D.6.19 Stock on hand report

Appendix E : Test Results

E.1 Log in

	Test Case	Result		Status
		Expected	Actual	
1	Click Login button with Username = "", Password = "", Remember me = uncheck	Refresh page with error messages for missing username and password.	Expected	Pass
2	Click Login button with Username = "{username}", Password = "", Remember me = uncheck	Refresh page with error messages for missing password.	Expected	Pass
3	Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = uncheck	Successfully login to landing page	Expected	Pass
4	Click Login button with Username = "{correct username}", Password = "{incorrect password}", Remember me = uncheck	Refresh page with error messages for incorrect username and password combination.	Expected	Pass
5	Click Login button with Username = "{incorrect username}", Password = "{correct password}", Remember me = uncheck	Refresh page with error messages for incorrect username and password combination.	Expected	Pass
6	Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = check. Then close the browser. Go to home page.	Load the landing page with the last login user details.	Expected	Pass
7	Logout if logged in. Click Login button with Username = "{correct username}", Password = "{correct password}", Remember me = uncheck. Then close the browser. Go to home page.	Load the login page.	Expected	Pass

Table E.6.1 Login test cases

E.2 Master Data screen

	Test Case	Result		Status
		Expected	Actual	
1	Click Add Button	Add a new empty row of data on the top row and first column become editable.	Expected	Pass
2	Click Cancel button	Discard changes and refresh the grid.	Expected	Pass
3	Click on the data cell	Become editable	Expected	Pass
4	Edit cell and make the focus out from the cell	Red marker on the top left corner of the cell	Expected	Pass
5	Click save button	Update the data in to database	Expected	Pass
6	Sort Ascending, Descending, Clear sorting	Sort data Ascending, Descending, Clear sort	Expected	Pass
7	Filter	Filter data according to criteria	Expected	Pass
8	Paging	Display first/Last, Previous/Next pages of data with buttons directly select page with page buttons	Expected	Pass
9	Click Delete	Ask for confirmation for delete	Expected	Pass
10	Delete confirmation cancel	Close the confirmation message	Expected	Pass
11	Delete confirmation confirm	Delete the record	Expected	Pass

Table E.6.2 Master data test cases

Appendix F : Code Listing

F.1 Web Configuration file important code sections.

```
6 <configuration>
7 <configSections>
8 <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
9 <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
10 Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
11 <section name="Telerik.Reporting" type="Telerik.Reporting.Configuration.ReportingConfigurationSection, Telerik.Reporting"
12 allowLocation="true" allowDefinition="Everywhere" />
13 </configSections>
14 <connectionStrings>
15 <!--<add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-SalesInventorySystem.Web-2f
16 providerName="System.Data.SqlClient" />-->
17 <add name="IdentityConnection" connectionString="Data Source=.;Initial Catalog=SIISDB;persist security info=True;user id=SIISUser;password=Chamika@123;"
18 providerName="System.Data.SqlClient" />
19 <add name="SIISDBEntities" connectionString="metadata=res://*/SIISDB.SIISDB.csdl|res://*/SIISDB.SIISDB.ssdl|res://*/SIISDB.SIISDB.msl;
20 provider=System.Data.SqlClient;provider connection string="data source=ROBOT02;initial catalog=SIISDB;persist security info=True;
21 user id=SIISUser;password=Chamika@123;MultipleActiveResultSets=True;App=EntityFramework";" providerName="System.Data.EntityClient" />
22 </connectionStrings>
23 <appSettings>
24 <add key="webpages:Version" value="3.0.0.0" />
25 <add key="webpages:Enabled" value="false" />
26 <add key="ClientValidationEnabled" value="true" />
27 <add key="UnobtrusiveJavaScriptEnabled" value="true" />
28 </appSettings>
--
```

Figure F.1 Configuration code sample

F.2 BL method for Add New Customer

```
61 public bool AddTrader(Trader term, string userId)
62 {
63     using (var c = new SIISDBEntities())
64     {
65         string traderNo = "";
66         switch (term.TraderTypeId)
67         {
68             case 1:
69                 traderNo = new SequenceMgt().GetNextNo("CustSeq");
70                 break;
71             case 2:
72                 traderNo = new SequenceMgt().GetNextNo("SupSeq");
73                 break;
74         }
75         term.TraderNo = traderNo;
76         term.Active = true;
77         term.CreatedDate = DateTime.Now;
78         term.CreatedUserId = userId;
79         c.Traders.Add(term);
80         c.SaveChanges();
81         return true;
82     }
83 }
84
```

Figure F.2 Add new customer business logic

F.3 Kendo Helper on View Code Sample

[illegible]

Figure F.3 Kendo view code sample 01

```

39 @Html.Kendo().Grid<Trader>()
40 .Name("gTrader")
41 .Columns(columns =>
42 {
43     columns.Bound(p => p.TraderNo).ClientTemplate("<span style='text-decoration: underline;cursor:pointer;' class='trader-no' data-id='#{Id#}'>#{Id#}</span>");
44     columns.Bound(p => p.Name);
45     columns.ForeignKey(p => p.TraderGroupId, (System.Collections.IEnumerable) ViewData["TraderGroups"], "Id", "Name");
46     if (Model== (int)TraderTypeEnum.Customer)
47     {
48         columns.ForeignKey(p => p.VehicleTypeId, (System.Collections.IEnumerable) ViewData["VehicleTypes"], "Id", "Name");
49         columns.Bound(p => p.VehicleNo);
50     }
51     columns.Bound(p => p.Active);
52     columns.Bound(p => p.CreditTotal);
53     columns.Bound(p => p.DebitTotal);
54     columns.Template(c => { }).ClientTemplate("#{DebitTotal - CreditTotal}").Title("Balance");
55 })
56 .Pageable(pageable => pageable
57     .Refresh(true))
58 .Navigatable()
59 .Sortable()
60 .Scrollable()
61 .Groupable()
62 .DataSource(dataSource => dataSource
63     .Ajax()
64     .PageSize(20)
65     .ServerOperation(false)
66     .Events(events => events.Error("error_handler"))
67     .Model(model => model.Id(p => p.Id))
68     .Read(read=> { read.Action("Trader_Read", "Trader").Data("TraderTypeInfo"); })
69 )
70 .Events(e => e.DataBound("gTraderOnDataBound"))
71 )

```

Figure F.4 Kendo view code sample 02

F.4 View JavaScript on View Code Sample

```
17 function onChange(arg) {
18     var grid1 = $("#ItemWarehouse").data("kendoGrid");
19     var selectedItem = grid1.dataItem(grid1.select());
20     if (selectedItem.Qty_OnHand - selectedItem.Qty_Invoiced > 0) {
21         var grid = $("#InvoiceLine_0").data("kendoGrid");
22         var datasource = grid.dataSource;
23
24         var count = grid.dataSource.data().length;
25
26         for (var i = 0; i < count; i++) {
27             if (grid.dataItems()[i].ItemCode === selectedItem.ItemCode && grid.dataItems()[i].WhseCode === selectedItem.WhseCode) {
28                 notify("error", "Item already added.");
29                 $('#ItemWarehouse').hide();
30                 $('#ItemCodeSearch').data('kendoDropDownList').select(0);
31                 return;
32             }
33         }
34
35         datasource.insert({
36             WhseCode: selectedItem.WhseCode,
37             ItemCode: selectedItem.ItemCode,
38             ItemTaxGroupCode: selectedItem.ItemTaxGroupCode,
39             UnitCode: selectedItem.UnitCode,
40             UnitPrice: selectedItem.Price,
41             Qty: 1,
42             DiscountPercentage: 0,
43             DiscountAmount: 0,
44             TaxPercentage: selectedItem.TaxPercentage
45         });
46     } else {
47         notify("error", "No Stocks Available");
48     }
49     $('#ItemWarehouse').hide();
50     $('#ItemCodeSearch').data('kendoDropDownList').select(0);
51     updateTotals();
52 }
```

Figure F.5 javascript code sample

F.5 Controller Class Code Sample

```
13 {
14     0 references
15     public class ItemTypeController : Controller
16     {
17         // GET: ItemType
18         0 references | 0 requests | 0 exceptions
19         public ActionResult Index()
20         {
21             return View();
22         }
23         0 references | 0 requests | 0 exceptions
24         public ActionResult _ItemType()
25         {
26             return PartialView();
27         }
28         0 references | 0 requests | 0 exceptions
29         public ActionResult GetAll()
30         {
31             return Json(new ItemTypeMgt().GetAll(), JsonRequestBehavior.AllowGet);
32         }
33         0 references | 0 requests | 0 exceptions
34         public ActionResult ItemType_Read([DataSourceRequest] DataSourceRequest request)
35         {
36             return Json(new ItemTypeMgt().GetAll().ToDataSourceResult(request), JsonRequestBehavior.AllowGet);
37         }
38
39         [AcceptVerbs(HttpVerbs.Post)]
40         0 references | 0 requests | 0 exceptions
41         public ActionResult ItemType_Create([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)
42         {
43         }
44
45         [AcceptVerbs(HttpVerbs.Post)]
46         0 references | 0 requests | 0 exceptions
47         public ActionResult ItemType_Update([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)
48         {
49         }
50
51         [AcceptVerbs(HttpVerbs.Post)]
52         0 references | 0 requests | 0 exceptions
53         public ActionResult ItemType_Destroy([DataSourceRequest] DataSourceRequest request, [Bind(Prefix = "models")]IEnumerable<ItemType> terms)
54         {
55         }
56     }
57 }
```

Figure F.6 Controller code sample

1. Model Class Sample

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace SalesInventorySystem.Entity.SISDB
8 {
9     39 references
10    public partial class Item
11    {
12        4 references | 0 exceptions
13        public string CostUnitCode { get; set; }
14        4 references | 0 exceptions
15        public decimal CostPriceUnit { get; set; }
16        4 references | 0 exceptions
17        public decimal CostPrice { get; set; }
18        4 references | 0 exceptions
19        public string SellUnitCode { get; set; }
20        4 references | 0 exceptions
21        public decimal SellPriceUnit { get; set; }
22        4 references | 0 exceptions
23        public decimal SellPrice { get; set; }
24        4 references | 0 exceptions
25        public decimal WhseQty { get; set; }
26    }
27 }
```

Figure F.7 Model class sample code

Appendix G : Client Certificate

IMPERIAL AUTO CARE

#44, Galle Road, Dehiwala.

Tel: 011 2 713 767

Project Examination Board,
University of Colombo School of Computing,
Colombo 07

Dear Sir / Madam,

Letter of Certification

This is to certify that Mr. P.B.C.I. De Silva has successfully designed and developed a Sales and Inventory Management System for Imperial Auto Care. The project was undertaken by him as a partial fulfillment of a requirement for the Bachelor of Information Technology Degree program.

The system has enabled the smooth operation of the business activities. It has also enhanced the sales, financial management. I am pleased to certify that the system developed by Mr. P.B.C.I. De Silva fulfill the requirements and could be used as our Sales and Inventory Management System.

This certification is issued on the request of Mr. P.B.C.I. De Silva.

Thank You,
Yours Faithfully,



.....
D.D.P. Lal
Assistant Accountant,
Imperial Auto Care.

Glossary

Actor: A person, organization or external system that plays a role in one or more interactions with the system. Actors are drawn as stick figures.

AJAX: A free framework for creating efficient and interactive web applications. It is the art of exchanging data with a server, and update parts of a web page without re loading the whole page.

Black Box Testing: Focusing the functions and behaviours of the system.

CSS: Language that is used to format the content of web pages.

Database: Structured collection of records or data. A computer database relies upon software to organize the storage of data.

Domain: Knowledge about the environment in which the target system operates.**HTML:** Predominant mark-up language for web pages.

Foreign Key: A column in a table that is used as a link to matching columns in other tables.

Razor: Server side programming language based on C#.

Object Oriented: A software design method that models the characteristics of abstract or real objects using classes and objects.

Primary Key: Key that uniquely identifies a record in a database table.

Transaction: An electronic operation that changes the contents of a database by adding, modifying, transferring or deleting data.

UML: Standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems.

Use Case: Actions that provide something of measurable value to an actor and are drawn as a horizontal ellipse.

Web Based: Information and/or application made available via the World Wide Web.

White box testing: Focusing the internal structure and logics of the system

Index

.	
.Net	43, 44
A	
Alpha testing	59
Analysis	5, 7
AnkhSVN	45
ASP.NET	48
B	
Beta testing	59
C	
C# 25, 44	
Chrome	43, 46
Chrome Dev Tools.....	46
CRUD.....	49
D	
Date Time Picker	40
DBMS	26, 46
DirectX	43
E	
Edge	24, 43
Email	63
Entity Framework	48
F	
Firefox	43
J	
jQuery	49
K	
Kendo-UI	49
M	
MVC	25, 49
O	
Opera	43
R	
ReSharper	45
RIA.....	20, 21, 22
S	
Safari	43
SMS	63
SQL Server 2014	44
SQL Toolbelt.....	45
T	
Telerik MVC UI	49
TortoiseSVN	45
U	
UAT	60
V	
Validation	58
Verification.....	58
Visual Studio	43, 44, 45
VisualSVN	45

W

Windows 10	24, 42
Web API	48, 54