



Daffodil
International
University

Project Title

"STOCK MANAGEMENT SYSTEM"

Supervised by:

Sarwar Hossain Mollah,

Dept. Head of CIS

Submitted by:

Name: MD. ABU SAEM

ID: 181-16-274

Submission Date: 22th June, 2020.

Spring, 2020

Department of Computing and Information System (CIS),
DAFFODIL INTERNATIONAL UNIVERSITY

APPROVAL

This Project titled “STOCK MANAGEMENT SYSTEM”, Submitted by Md. Abu Saem, ID No: 18-16-274 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on 19-07-2020.

BOARD OF EXAMINERS



Mr. Md Sarwar Hossain Mollah
Assistant Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Chairman



Ms. Nayeema Rahman
Sr. Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Minhaj Hosen
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Saifuddin Md. Tareeq
Professor
Department of Computer Science and Engineering
Dhaka University, Dhaka

External Examiner

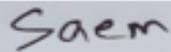
DECLARATION

I hereby declare that, this project has been done by me under the supervisor of “**Sarwar Hossain Mollah**” Assistant Professor & Head of the department of CIS of Daffodil International University. It is also declared that neither this project nor any part of there has been submitted anywhere else for the award of any degree, diploma or other qualification.



Mr. Md Sarwar Hossain Mollah
Assistant Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Supervisor



Md. Abu Saem

Student

ID: 181-16-274

Department of Computing & Information System (CIS)

Faculty of Science & information Technology

Daffodil International University.

Acknowledgement

First of all I would like to give all the credit to almighty Allah who gives me physical and mental strength during my project to do it properly. This project is the final stage to complete my BSc (Honors) degree in Computing and Information System from Daffodil International University.

I would like to thank my mentor, beloved and most respected teacher, my project supervisor Mr. Sarowar Hossain Mollah, Head of the Department of CIS, Daffodil International University. During my project he helped me a lot, he supervised me properly, give proper inspiration and provide the actual guideline I have needed. Without his help and proper guideline I can't complete my project properly.

I would also like to thanks my classmates, others teacher and my family member for their greatest support during project time. They help me and give me proper support.

And finally again all the praises goes to almighty Allah and I'm really very much grateful to him

MD. ABU SAEM

ID: 181-16-274, Dept. of CIS

Daffodil International University

Executive Summary

Stock management system is a web based system which is developed for managing stock in dynamic way. In the age of modern world everything has been digitalized and people comes out from the traditional things. Stock management system is platform independent system. It can used for any kind of business to manage stock. Such as Super shop's stock management, Medicine sector's stock management, Garments' stock management etc. People or organizations who wants to manage their stock dynamically, only for them I developed this system. By using this system they can easily manage their stock and can know the current status of their stock. The can easily prevent business losses and can increase the level of profit by using this system.

Table of Contents

Chapter 1 - Introduction	1
Initial description of the Project:	1
Document Context of the project:	2
Chapter 2 – Initial Study	3
2.1 Project Proposal:	3
1. Initial Conception	3
a) Brief description of the project:	3
b) Proof of the concept:	4
i) Prototyping:	4
ii) Initial Research	4
2. Feasibility Study	5
a) Operational Feasibility:	5
b) Technical Feasibility	5
c) Economic feasibility:	5
d) Market research analysis based on feasibility factors:	7
3. Foundation	8
a) Defined goals/objectives of the project:	8
High level features/requirement to achieve goals/objectives:	9
4. Exploration and Engineering	10
a) Iterative development:	10
Time box:	10
Gantt chart:	10
2.2 Background of the project:	12
2.2.1 Goals of the project:	13
2.2.2 Objectives of the project:	13
2.3 Problem areas:	13
2.4 Possible solution:	14
Chapter 3 - Literature review	15
3.1 Discussion on problem domain based on published articles:	15
3.2 Discussion on problem solution based on published articles:	15
3.3 Comparison of 3/4 leading solution:	16
3.3.1 Best features:	16

3.3.2 Limitations:	18
3.4 Recommended approach:	18
Chapter 4 - Methodology	19
4.1 What to use and why to use:	19
4.1.1 Waterfall approach:	19
4.1.2 Rapid action development methodology:	19
4.1.3 Dynamic system development methodology:	20
4.3 Section of Methodology:	20
4.4 Implementation plans	21
Chapter 5 – Planning	23
5.1 Project Plan	23
5.1.1 Work Break down Structure (WBS):	23
5.1.2 Resource Allocation	24
5.1.3 Time Boxing	25
5.1.4 Gantt Chart	25
5.2 Test Plan	26
5.2.1 Required Testing	26
5.2.2 Test Case:	27
5.2.3 User acceptance testing:	28
Chapter 6 - Feasibility	29
6.1 All possible types of feasibility	29
6.1.1 Economic feasibility:	29
6.1.2 Operational Feasibility:	30
6.1.3 Technical Feasibility	31
6.2 Cost Benefit Analysis	31
6.3 DSDM – Good or Not for this project:	32
Chapter 7 - Foundation	33
7.1 Overall requirement list	33
7.2 What technology to be implemented:	34
7.3 Recommendations and justifications:	35
Chapter 8 - Exploration	36
8.1 Old full system use case:	36
8.2 Old system full activity diagram:	37

8.3 Prototype of new system:	39
Chapter 9 - Engineering	44
9.1 New system modules:	44
9.2 Use case Diagram:	45
9.3 Class diagram	46
9.4 Sequence Diagram	47
9.5 System Interface	48
Chapter 10 - Deployment.....	53
10.1 Core modules coding samples of my system:	53
10.2 Possible problem breakdown	61
Chapter 11 - Testing	62
11.1 Unit Testing	62
11.2 Module Testing	64
11.3 Integration Testing	67
Chapter 12 - Implementation	69
12.1 Training	69
12.2 System Implementation	69
12.3 Recommended implementation process	70
Chapter 13 - Critical appraisal and evaluation	71
13.1 Objectives that could be met	71
13.1.1 Success rate against each objective	71
13.1.2 How much better could have been done	71
13.1.3 How better is the features of the solution?	72
13.1.4 Which features couldn't be touched:	72
13.2.1 Why this feature couldn't be touched	72
13.2 Objective totally not met / touched.....	72
13.2.1 Why it could not be touched	72
13.2.2 What could have been done?	72
Chapter 14 - Conclusion.....	73
14.1 Summary of the project.....	73
14.2 Goals of the project	73
14.3 Success of the project	73
14.4 Value of the project.....	73

14.5 My experiences..... 74
References **Error! Bookmark not defined.**
Plagiarism Report: 97

Table of Figures

Figure 1: Sales Binder Inventory Management System application.....	16
Figure 2: Inflow Inventory Management System application.....	17
Figure 3: Trade decko Inventory Management System application	17
Figure 4: Work break down structure of proposed system.....	23
Figure 5: Resource allocation of proposed system	24
Figure 6: Time box of proposed system	25
Figure 7: Gantt chart of proposed system	25
Figure 8: acceptance testing template.	28
Figure 9: Use case diagram for stock management system	45
Figure 11: Sequence diagram for stock management system.....	47
Figure 16: Product added successfully	65
Figure 17: Supplier added successfully.....	65
Figure 18: Purchase product from supplier	66
Figure 19: Purchase product from supplier	67
Figure 20: Sold product successfully	68
Figure 21: Category added and showing successfully	76
Figure 28: Purchase product from supplier	81
Figure 29: Sold product successfully	82

Chapter 1 - Introduction

Initial description of the Project:

Now a day's people wants everything digitally. Because it is very easy to handle things in dynamic way. And also in system based operation data can be stored and you can access these data anytime when you need. You can't do all the things in traditional approach. It is very tough for an organization or business to handle and manage their stock traditionally. And this approach also time consuming. They don't get their report when they want that's why they can't take decision based on report. As a result organization have to suffer.

From this point of view I decided to develop Stock Management system and in developed it.

As I mentioned previously there are lot of problems to manage stock traditionally. There are lot of possibilities of getting loss if ones manage their stock traditionally. That's why I decided to build up a web based system which name is Stock Management system. The expected result of Stock Management System is Manage Stock dynamically. After fully development of this system, system should be able to setup category, product under category, supplier and customer. It should be able to purchase product and sale product and required functionalities under purchase and sale module. After development admin can get stock current status, periodically report on sale and purchase.

Document Context of the project:

Document context of the project are given below according serial number:

- **Chapter 1: Introduction**
This chapter is all about the introduction and explanation of your proposed system
- **Chapter 2: Initial study**
Here I have explained about the background, aims, objectives of proposed system and finding problem area of this system.
- **Chapter 3: Literature Review**
In this chapter I have discussed about domain problem and the proposed solution for it and compared my project (because it already exist) with world leading software.
- **Chapter 4: Methodology**
Here I have discussed about the methodology what should use and which I have used and reasons behind it.
- **Chapter 5: Planning**
Here I have discussed about the way how I managed my plan according breakdown structure, how I handled risk management and how I execute test plan.
- **Chapter 6: Feasibility**
Here I have explained about the economical, operational, social and technical feasibility of my proposed system.
- **Chapter 7: Foundation**
In this chapter I have explained the whole process of project with proper diagram.
- **Chapter 8: Exploration**
Here I have discussed about old system and its boundaries with different types of diagram.
- **Chapter 9: Engineering**
Activities and engineering part of my proposed system are described here.
- **Chapter 10: Deployment**
Development with work prioritization, its technique and steps are described here.
- **Chapter 11: Testing**
Here I have showed all the possible test cases of my system.
- **Chapter 12: Implementation**
Here I have explained the implantation step and technique of my project.
- **Chapter 13: Critical Appraisal and Evaluation**
Here I have provide the description about the evaluation of this project that are measured.
- **Chapter 14: Conclusion**
Here I have provide a short descript about project and process of project to complete.

Chapter 2 – Initial Study

2.1 Project Proposal:

1. Initial Conception

a) Brief description of the project:

Now a day's people wants everything digitally. Because it is very easy to handle things in dynamic way. And also in system based operation data can be stored and you can access these data anytime when you need. You can't do all the things in traditional approach. It is very tough for an organization or business to handle and manage their stock traditionally. And this approach also time consuming. They don't get their report when they want that's why they can't take decision based on report. As a result organization have to suffer.

From this point of view I decided to develop Stock Management system and in developed it.

As I mentioned previously there are lot of problems to manage stock traditionally. There are lot of possibilities of getting loss if ones manage their stock traditionally. That's why I decided to build up a web based system which name is Stock Management system. The expected result of Stock Management System is Manage Stock dynamically. After fully development of this system, system should be able to setup category, product under category, supplier and customer. It should be able to purchase product and sale product and required functionalities under purchase and sale module. After development admin can get stock current status, periodically report on sale and purchase.

b) Proof of the concept:

i) Prototyping:

Stock

Product	<input type="text"/>	Start Date	<input type="text" value="mm/dd/yyyy"/>
Category	<input type="text"/>	End Date	<input type="text" value="mm/dd/yyyy"/>
<input type="checkbox"/> Reorder Level			
<input type="checkbox"/> Expire			

SI	Code	Name	Category	Reorder Level	Ex. Date	Ex. Qty	Opening Blance	IN	Out	Closing Blance
1	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	20	440	12	22	500
2	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	50	600	12	12	500
3	M101	Mobile	Electronics	12	12/14/2019 12:00:00 AM	10	100	12	10	500
4	Mob11	iPhone	Electronics	5	12/23/2019 12:00:00 AM	1	12	45	12	40

ii) Initial Research

Market Viability:

Stock management system is a web based system which is developed for managing stock in dynamic way. In the age of modern world everything has been digitalized and people comes out from the traditional things. Stock management system is platform independent system. It can used for any kind of business to manage stock. Such as Super shop's stock management, Medicine sector's stock management, Garments' stock management etc. People or organizations who wants to manage their stock dynamically, only for them I developed this system. By using this system they can easily manage their stock and can know the current status of their stock. The can easily prevent business losses and can increase the level of profit by using this system.

In our country there are lots of Inventory management system which is very much vast and costly. Small and medium organizations can't afford them. From that point of view I would like to build up this system to grab this market

2. Feasibility Study

a) Operational Feasibility:

By using this system different types of organization can handle their stock dynamically. There will be one user in this system called admin. To manage stock he can setup category, product, customer, supplier, can know current status of stock, can see periodically report. Also can take decision based on stock status. Here I will provide better type of operational feasibility for my system:-

- All kind of user such as customer and admin easily access to my system.
- All kind of product added easily.
- All kind of stock maintain easily.

b) Technical Feasibility

To run developed system which is a web based system we need just one computer and one server. Where PC is in the client end and server is in server end. All the information pushed through the system will stored in server. According necessity data will be fetched from server to the client end.

Hardware	PC/Laptop (4 GB RAM, 1TB HDD), Wi-Fi Router
Software	<ul style="list-style-type: none">• Adobe creative Cloud• MS office 2016• OS windows-10• Visual Basic.

c) Economic feasibility:

As we want to build this project for managing stock we just need one pc and one server. So, we can manage it in low cost because we don't need a large number of pc and servers.

Costs:

- Development cost
- Software cost.
- Hosting cost
- Storage cost (Anon., n.d.).

S.N	Component	Price (BDT)
01.	Hardware	30,000/=
02.	Software	20,000/=
03.	Cost of Web Hosting	1,500/=
04.	Other Components	10,000/=
Total		61,500/=

Hardware Price: Required hardware's prices are given below:

S.N	Component	Price (BDT)
01.	Desktop/pc	40,000/=
02.	Wi-Fi Router	2,000/=
03.	Electrical components	2,000/=
Total		44,000/=

Software Price: For making this system required software list and prices are

S.N	Component	Price (BDT)
01.	MS Office	10,000/=
02.	Browser (Free)	0/=
03.	Adobe Create Cloud	10,000/=
04.	OS (Windows)	10,000/=
05.	Visual Basic	15,000/=
Total		40,000/=

Hosting Price:

S.N	Hosting Name	Price(BDT)
01.	https://www.StockManagementSystem.com	1,500/=

d) Market research analysis based on feasibility factors:

After analyzing all the feasibility factor I can say that this system is possible to develop technically, economically and physically. The system would be good for small and medium size company.

3. Foundation

a) Defined goals/objectives of the project:

Goals:

The main goal of the proposed system is manage stock or warehouse to prevent loss. This system was wanted as a supportive tool to manage stock or warehouse properly.

Objectives:

1. Category setup: Admin can setup category to setup products under each category. He can edit and delete category
2. Product setup: Admin can setup product under each category. He also can set product reorder level that's why system can inform him if product is going below reorder level. There are also an option from where admin can easily access if needed during add product under category.
3. Supplier setup: Admin can setup supplier with their details so that he can track from where they purchase product. Here also he can edit and delete supplier.
4. Customer setup: Admin can setup customer with their details so that he can track to whom they sale their product. Here also he can edit and delete customer.
5. Purchase: Admin can purchase product under supplier multiple times. Here product code and total cost of product will auto appear, if admin set unit price and quantity. Admin can delete purchase before save it to database because there is a partial table where purchase store temporarily before sending it to database.
6. Sale: Admin can sell product to customer and he can sell product multiple times. Here customer will rewarded on specific product according their loyalty point. Admin can delete sell before save it to database because there is a partial table where sell store temporarily before sending it to database.
7. Stock: Admin can check stock for expire product, sold product, damage product etc. He can also which product is going under reorder level. System will notify to admin to restore product again. He can check how many product are in, how many products are out, product opening balance, product closing balance etc.

8. Report: Admin can see report periodically on sale and purchase. He can also check business loss and profit. He can make decision based on statistics.

High level features/requirement to achieve goals/objectives:

- Add product
- purchase product
- Selling product
- Check stock periodically
- Common searching feature
- Generate report periodically
- Live notification

Nonfunctional requirements:

1. System validation and verification
2. System security
3. Reliability
4. Response time of each function
5. Testability
6. Efficiency of system

4. Exploration and Engineering

a) Iterative development:

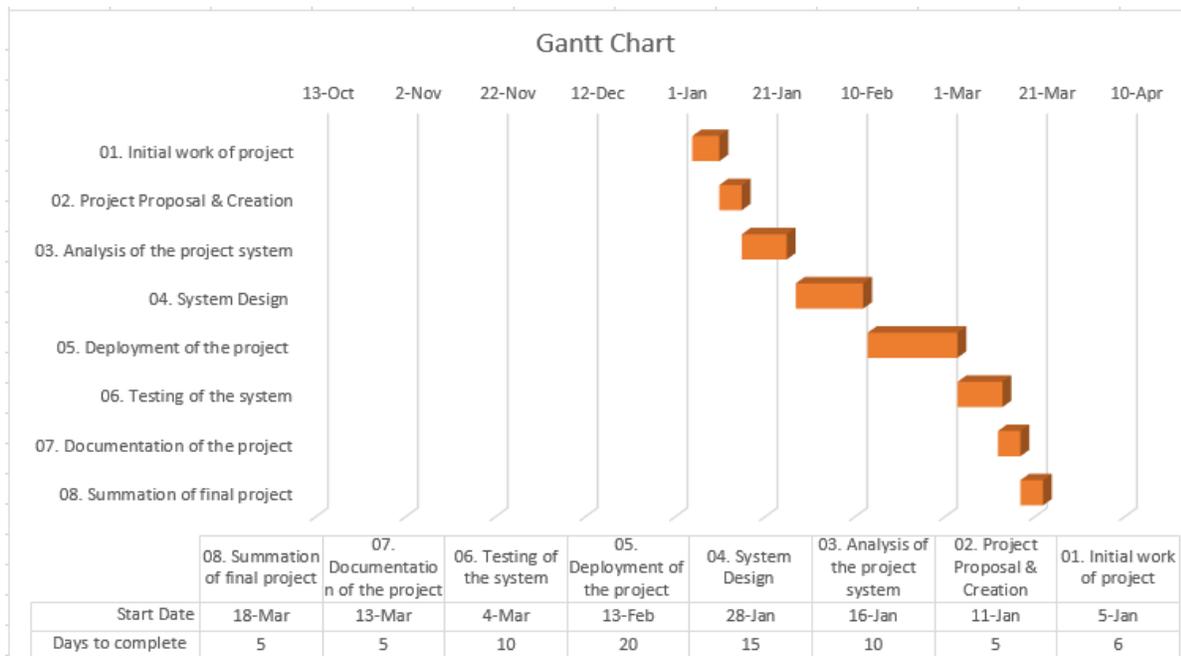
Iterative development are listed below:

- Checking stock
- Notify user

Time box:

Task Name	Start	End Date	Days to complete
01. Initial work of project	5-Jan	10-Jan	6
02. Project Proposal & Creation	11-Jan	15-Jan	5
03. Analysis of the project system	16-Jan	27-Feb	10
04. System Design	28-Jan	12-Feb	15
05. Deployment of the project	13-Feb	3-Mar	20
06. Testing of the system	4-Mar	12-Mar	10
07. Documentation of the project	13-Mar	17-Mar	5
08. Summation of final project	18-Mar	22-Mar	5

Gantt chart:



Features wise development:

First of all I have to complete user authentication part, then category setup, product setup, supplier setup, customer setup, purchase product, sale product, check stock, report generate and going on accordingly

Deployment:

I will develop web based application and there will be only one user in my system known as admin. After comparing with others I think web server means web based application will be best suited for our proposed system. It has lot of features and advantages such as accessible from anywhere through internet connection, effective cost, doesn't need to be installed on any devices, fast and easy updating etc.

I will complete my project by using ASP.NET as backend, jQuery for UI and make it first and more responsive, HTML and CSS for structure, MsSQL for store data and many more according needs.

2.2 Background of the project:

Now a day's people lives in a modern era. They made everything according to their needs. They tried all the possible way to make their life easy and comfortable. Stock house or warehouse is a very common term right now. People stores their needed thing for different purpose. It is a very toughest thing to manage stock in proper way. After that people manage their stock in analogue way. Their stock management was paper or oral based. As a result, most of the time they have fallen in a great loss. From that point of view I have decided to build a system which will be known as stock management system.

Stock management system is a kind of system which one will be developed to manage any kind of business's stock. As I mentioned previously there are lot of problems to manage stock traditionally. There are lot of possibilities of getting loss if ones manage their stock traditionally. That's why I decided to build up a web based system which name is Stock Management system. The expected result of Stock Management System is Manage Stock dynamically. After fully development of this system, system should be able to setup category, product under category, supplier and customer. It should be able to purchase product and sale product and required functionalities under purchase and sale module. After development admin can get stock current status, periodically report on sale and purchase.

There are only one user in this system. There won't multiple user. Only admin can handle it because there is no need of multiple user to manage stock. All the types of functional and non-functional requirements will be implemented for this system. The system has total 8 modules. Here first of all admin need to setup category and setup product under category. Then he need to setup supplier and customer from where the organization buy products and to whom they sell their product. When admin setup products he can set reorder level against individual product. And when setup customer he can add loyalty point for customer and next time during purchasing customer will get reward according their loyalty point. There also purchase and sale module. Admin can store all the information about purchase. Such as which kind of products they bought, total products, product MRP etc. Admin can also store sale information. He can check stock. Such as which product get expired, number of damaged product, number of sold product, number of stored product etc. there also report module in this system where admin can check daily or monthly report of purchase and sale. He can also get statistics about loss and profit.

2.2.1 Goals of the project:

- Selling and purchasing easily
- Manage stock in proper way
- Generate periodical report to make decision
- Customer prioritization

2.2.2 Objectives of the project:

- The main objectives of my proposed system is manage stock properly to prevent any kind of losses
- Help organization to make right decision in right time
- Keep each and every calculation

2.3 Problem areas:

There are always some problems in every project. Before I built stock management system for one organization they use traditional approach to run their business and manage their stock. They used paper based system to manage and store their sensitive information like customer information, transaction information, and purchase and sell information etc. That's why the management of this organization wants to make this system fully digital. In spite of having too short time, they has a lot of task to do. But I figure out main functionality of proposed system to be done:

- Product Catalog a. Product Category Setup (Entry, Edit, Search) b. Product Setup with Category (Entry, Edit, Search)
- Party Module a. Customer Setup (Entry, Edit, Search)
- Supplier Setup (Entry, Edit, Search)
- Purchase Module a. Purchase Entry b. Purchase Edit
- Purchase Confirm d. Purchase Search
- Sales Module a. Sales Entry b. Sales Edit c. Sales Confirm d. Sales Search
- Stock Module a. Periodical Stock Report (From - To, Opening Balance, In, Out, Closing Balance) b. Find Low product & Expired product.
- Reporting Module a. Periodic Income Expense Report on Sales b. Periodic Income Expense Report on Purchase

2.4 Possible solution:

As I mentioned previously there are lot of problems to manage stock traditionally. There are lot of possibilities of getting loss if ones manage their stock traditionally. That's why I decided to build up a web based system which name is Stock Management system. The expected result of Stock Management System is Manage Stock dynamically. After fully development of this system, system should be able to setup category, product under category, supplier and customer. It should be able to purchase product and sale product and required functionalities under purchase and sale module. After development admin can get stock current status, periodically report on sale and purchase.

Chapter 3 - Literature review

A literature review is a kind of article review. Literature review basically helps to gain knowledge about present and past. It also helps to know about theoretical and methodological contribution to a particular topic.

3.1 Discussion on problem domain based on published articles:

To grow business by using stock management system, there are several activities which should be considered.

- Products are damageable: Products can be damaged during transportation. So it is very difficult to keep track.
- Lack of basic facilities: In third world country like us, there are always have problem like electricity problem, transportation problem, telecommunication problem etc. as a result it may very costly to run a single/small business
- Financial unavailability: Some business organization wants to apply it to run their business properly but it may not be financially feasible.
- Support of government: In our country, recently government announced and investing on IT development but government's help is not enough. Without this there are lot of boundaries like vat and taxes on product for importing and exporting.

3.2 Discussion on problem solution based on published articles:

- Provide basic services like electricity, telecommunication, easy transportation system, banking services etc.
- Training: Train people to carry and manage things properly on shop
- Advertise campaign: Make audio and video advertisement for rural people and for urban people make web advertisement.
- Information system for inventory: Information system can be used for sharing information and knowledge.
- Mobile service for inventory system: To connect with huge number of peoples mobile features should be enabled.
- The mandatory thing is flexibility and adaptability in terms of technology.

3.3 Comparison of 3/4 leading solution:

Here I will compare leading 3/4 project to give best solution for organization. These leading websites are given below and will be compared.

- <https://www.salesbinder.com/>
- <https://www.inflowinventory.com/software-features>
- <https://www.tradegecko.com/>

3.3.1 Best features:

Sales Binder:

- Cloud based online inventory system
- Has ability to organize supplier, customer, invoice and many more
- They are currently providing different type of features like Multi ware-house, cloud based, barcoding etc.
- They provides trial period for 30 days.
- Hass different subscription plan (Anon., n.d.).



Figure 1: Sales Binder Inventory Management System application

Inflow:

- Online inventory management system on the cloud
- Has ability to organize supplier, customer, invoice and many more
- Has live chat option for customer
- They are currently providing different type of features like Multi ware-house, cloud based, barcoding etc.

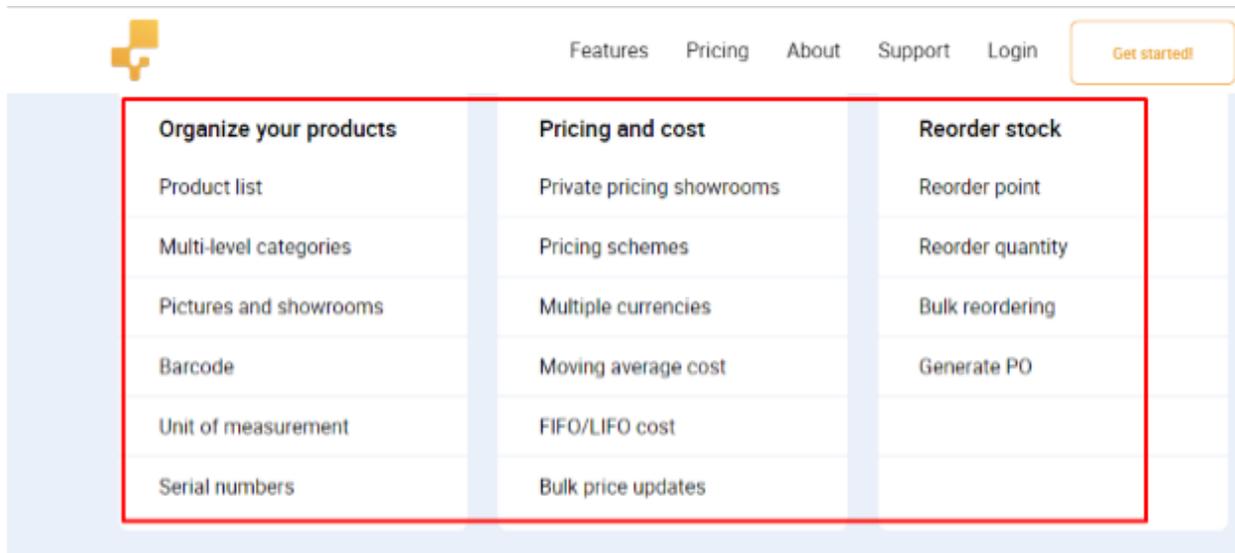


Figure 2: Inflow Inventory Management System application

Trade decko:

- Has ability to organize supplier, customer, invoice and many more
- It has 14 day trial period for customer
- Provide multiple feature like order management, purchasing, inventory management etc (Anon., n.d.).

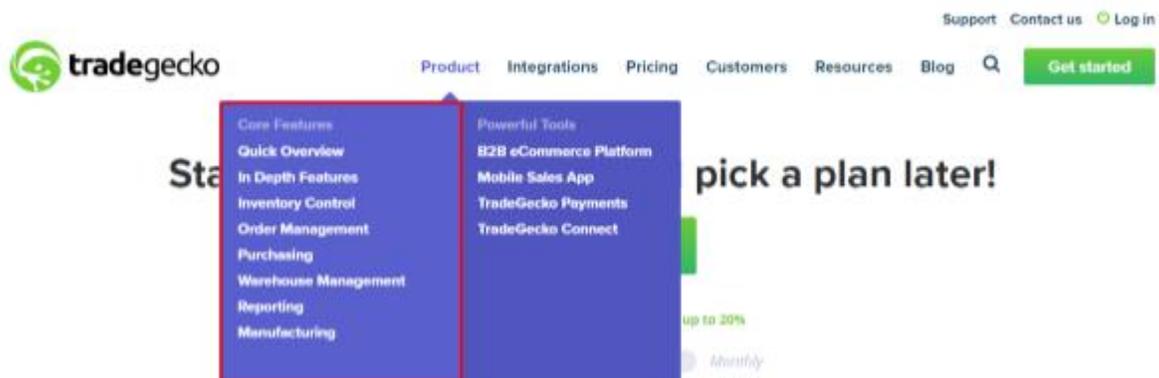


Figure 3: Trade decko Inventory Management System application

3.3.2 Limitations:

Sales binder:

- Need good internet service as the provide cloud based service
- In cloud you may face lot of problems during changes

Inflow:

- Need good internet service as the provide cloud based service
- In cloud you may face lot of problems during changes (Anon., n.d.).

3.4 Recommended approach:

It would be better to developed by using agile philosophy where the system developed according prioritization of their demand and entire requirement would be stored properly so that they can easily and properly go through with their business without any hassle. Before start the development proper risk management plan and implementation of development should be analyzed. After development acceptance criteria should be matched. Anyhow this project should be done with massive features within fixed timeframe.

Chapter 4 - Methodology

Methodology always leads a project to complete within fixed budget and timeframe

4.1 What to use and why to use:

To develop this proposed system I have chosen DSDM which is belong to agile methodology. There are lot of reasons of choosing it. It helps to create focus on complete project within fixed budget and timeframe (Anon., n.d.).

Main reason of using DSDM are explained in below by comparing with different methodology:

4.1.1 Waterfall approach:

Waterfall model was the first model introduced with development. It is also called linear approach.

Advantages and disadvantages of waterfall approach:

Advantages:

- Easy to arrange time and task
- simple and easy to understand
- Milestones are well understood and sets requirements stability

Disadvantages:

- Very difficult to go back and change things
- Uncertainty and high amount of risk (Anon., n.d.).

4.1.2 Rapid action development methodology:

Rapid action methodology is all about do more and less talk. It has four phases: Requirement, planning, user design, construction and cutover.

Advantages and disadvantages of Rapid action methodology:

Advantages:

- Break down the whole project into smaller task
- Optimizing team efficiency and assign specific task to experienced one
- One can get desire product into short time frame

Disadvantages:

- Technical risk is very high
- Short time, less feature

4.1.3 Dynamic system development methodology:

It is an agile development methodology. It focusses on early delivery, strategic goals, and completed project on time and real benefit to the business

Advantages and disadvantages of DSDM:

Advantages:

- User involvement is very high in development
- completes project on time
- rapidly provide basic functionality
- Follow MoSCoW prioritization

Disadvantages:

- Not suitable for smallest one
- DSDM is restrictive and difficult to work

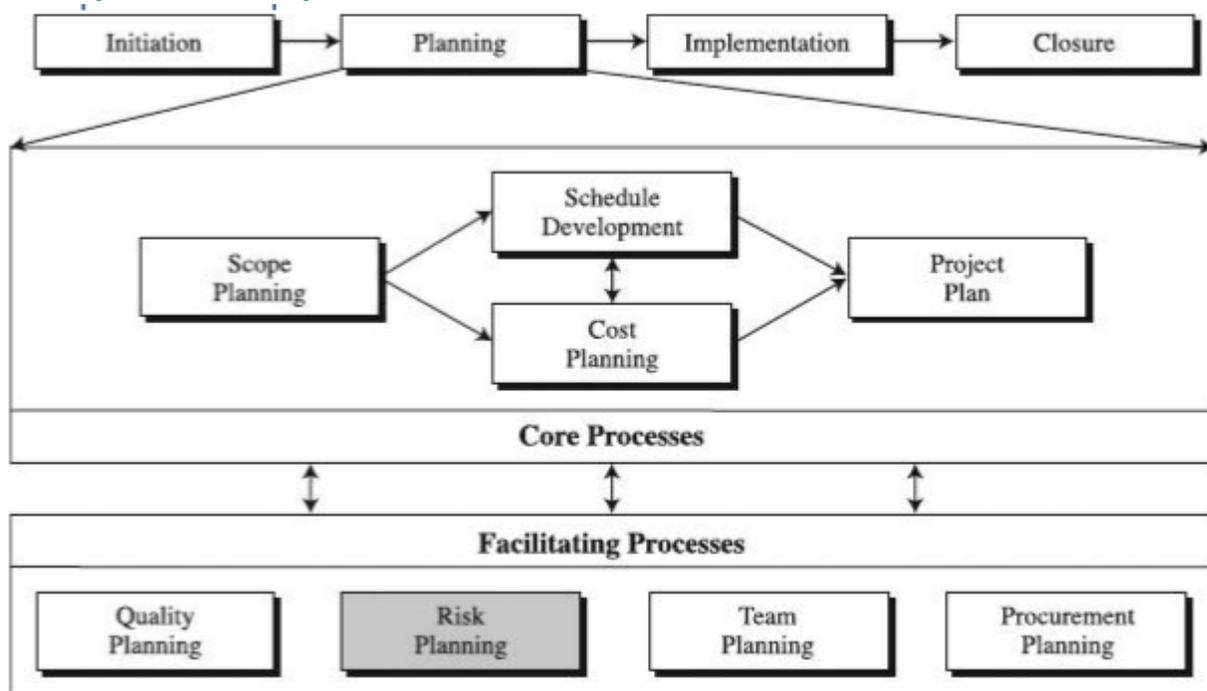
After comparing with all other methodology I think DSDM is the best methodology to work. That's why I'm going with DSDM.

4.3 Section of Methodology:

- Pre-project: Initial phase where project is selected and project proposal is made. Business problems are also identified here. Initial budget will be allocated here according project. This types of problem and solution are identified in this pre project.
- Feasibility: Here feasibility study specify that proposed system is financially and technically feasible or not to develop. It helps to make decision about proposed system that will work or not.
- Foundation: It defines problem, establishes goals and objectives, and builds a project plan to manage the resources. It provides tip to keep track of project performance to increase performance and gain customer acceptance.
- Exploration: All the functional requirements identify here and prioritize these requirements according Moscow analysis. It will help me to make decision on activities and deliverables to be achieved in any time box within the project,

- Engineering: This process is incremental and very highly iterative development process to achieve product's operational preparation. During engineering and exploration phase system testing is taken
- It involves with physical implementation. If any types of problem can be found then back to engineering phase and solve the problem
- Post-project: After completing the whole project deploy it where realize benefit to the business

4.4 Implementation plans



There are lot of key elements which need to be considered to implement our chosen methodology (DSDM Attern). These are given below:

- Acceptance of DSDM methodology: Before start working developer and organization must have to agree with DSDM Attern else it loses it philosophy.
- A supportive commercial relationship: For development purpose developer team need to access an organization data. That's why company must have willingness to help them.
- Only one team collaboration: Developer and organization should think themselves as only one team else system can't reach its goal.
- Solution development team stability: Development team of solution must need to be stable. Anyone from outside shouldn't be allowed at mid of the project.

- Solution development team skill: Selected team should have proper knowledge about project and development before agreement.
- Development team size: Development team size shouldn't be more than 10 and the ideal size is 6.
- Incremental Delivery: Systems' delivery needs to be in an incremental process. So that early business benefit can come in short time.
- Considering risk: Some external and internal risk can be happen during development like electricity loss, employee sick leave etc. In that case before start project, risk planning should be taken.

Chapter 5 – Planning

For any kind of project planning is very important. Because it shows the way how to and in which way one can complete their project properly.

5.1 Project Plan

5.1.1 Work Break down Structure (WBS):

It is a working process where we divided out total task into multiple small task in order to complete it properly and easily within time.

Serial	Task	Starting	Ending	Duration
1.	Initial work for the project.	05/01/2020	10/01/2020	6 days
2.	Project proposal and creation	11/01/2020	15/01/2020	5 days
3.	Analysis of the project	16/01/2020	27/01/2020	10 days
4.	System Designing	28/01/2020	12/02/2020	15 days
5.	Deployment of the project	13/02/2020	3/03/2020	20 days
6.	System Testing	04/03/2020	12/03/2020	10 days
7.	Project Documentation	13/03/2020	17/03/2020	5 days
8.	Conclusion	18/03/2020	22/03/2020	5 days
Total:				76

Figure 4: Work break down structure of proposed system

5.1.2 Resource Allocation

Time Boxes	Task Name	Resource Name
T - 01	Introductions	Analyst
	Initial study	Analyst
	Literature Review	Analyst
	Methodology	Analyst
T - 02	Planning	Analyst, user
	Feasibility	Analyst, Developer
	Foundation	Analyst, Developer
	Exploration	Analyst, Developer
T - 03	Engineering	Analyst, Developer
	Deployment	Developer team, Tester Team
	Testing	Tester
	Implementation	Analyst, Developer
T - 04	Evaluation	Analyst, Developer
	Critical Appraisal	Analyst, Developer
	Conclusion	Analyst, Developer, Tester

Figure 5: Resource allocation of proposed system

5.1.3 Time Boxing

It shows the duration that will be taken to complete a task.

Task Name	Start	End Date	Days to complete
01. Initial work of project	5-Jan	10-Jan	6
02. Project Proposal & Creation	11-Jan	15-Jan	5
03. Analysis of the project system	16-Jan	27-Feb	10
04. System Design	28-Jan	12-Feb	15
05. Deployment of the project	13-Feb	3-Mar	20
06. Testing of the system	4-Mar	12-Mar	10
07. Documentation of the project	13-Mar	17-Mar	5
08. Summation of final project	18-Mar	22-Mar	5

Figure 6: Time box of proposed system

5.1.4 Gantt Chart

It shows the activity within fixed timeframe.

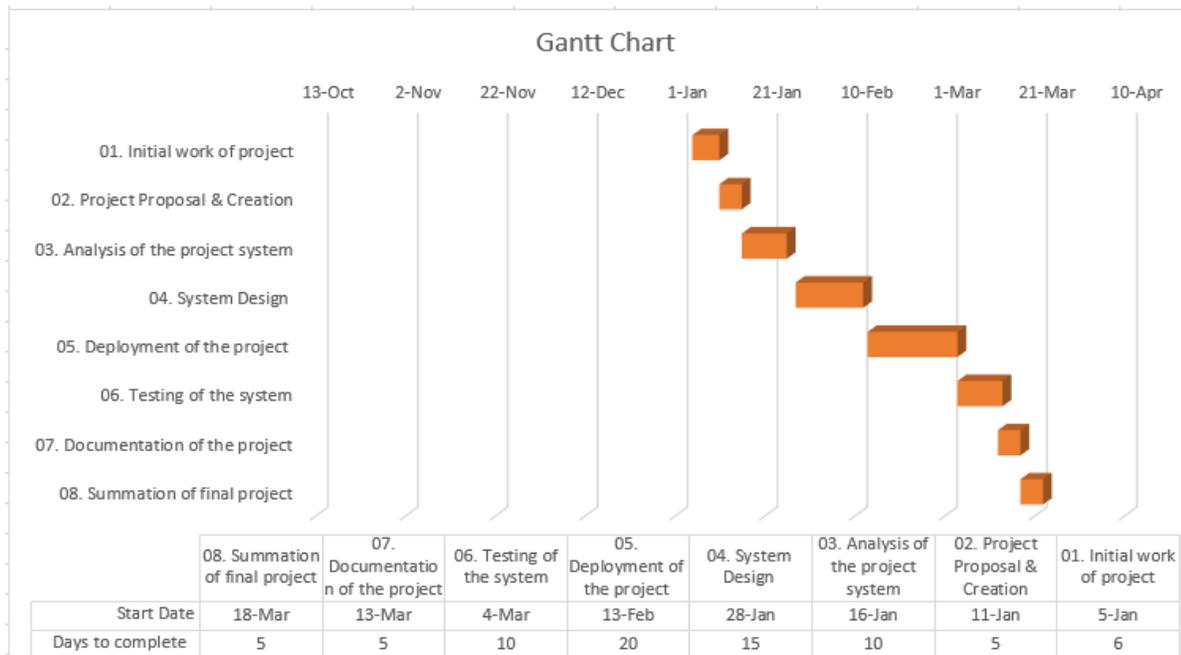


Figure 7: Gantt chart of proposed system

5.2 Test Plan

5.2.1 Required Testing

In this time there has lot of testing method for software testing. I have chosen some of them, like- Unit Testing, Integration Testing, Module Testing, and Acceptance Testing. Here I have describe about these type of testing:-

- **Unit Testing:** Unit testing is a procedure which mainly used for validate any kind of software or individual unit of source code is working perfectly or not. It is a smallest testable part of an application. When this type of testing is running, it usually has one or few input and usually single input for identify actual result of testing.
- **Integration Testing:** Integration testing is known as second level of testing method for any kind of software or application. It comes after unit testing. The main purpose of this testing is individual components of a software are tested in a group and find out software faults (Anon., 2020).
- **Module Testing:** A module testing is a technique, which is a process of testing individual software components, classes, subroutines, subprograms in a program. This type of testing are mainly carried out by a group of software testers. This group's main target is testing software components in isolation also check about business requirement match or not (Anon., 2020).
- **Acceptance Testing:** It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not (Anon., 2020).

5.2.2 Test Case:

A test case is a document, which has a set of data, precondition, post condition, excepted result, actual result and test description. It developed for a particular test scenario to check the specific function or requirement. Test case are starting point of test execution. Some input values applying for application testing and get actual result. After completing testing, test document is ready for test description (Anon., 2020).

Here I have provided a test case template:

Test case ID:		Test Case Name:		
Design By:		Design Date:		
Executed by:		Execution date:		
Post-condition:				
Pre-condition				
Step	Action	Excepted Result	Actual Result	Comment

5.2.3 User acceptance testing:

User acceptance testing is a methodology where end users involved in this testing. End users testing that the product is validate against their requirements or not. Generally it performed at client location developer site. Here I have provided an acceptance testing template: -

Test case ID		User Name	
Test Type		Role	
Test Name			
Pre-Condition			
Post-Condition			
Steps	Excepted Result	Actual Result	Comments

Figure 8: acceptance testing template.

Chapter 6 - Feasibility

It measure that the proposed system would be technically and operationally feasible or not.

6.1 All possible types of feasibility

6.1.1 Economic feasibility:

As we want to build this project for managing stock we just need one pc and one server. So, we can manage it in low cost because we don't need a large number of pc and servers.

Costs:

- Development cost
- Software cost.
- Hosting cost
- Storage cost.

S.N	Component	Price (BDT)
05.	Hardware	30,000/=
06.	Software	20,000/=
07.	Cost of Web Hosting	1,500/=
08.	Other Components	10,000/=
Total		61,500/=

Hardware Price: Required hardware's prices are given below:

S.N	Component	Price (BDT)
04.	Desktop/pc	40,000/=
05.	Wi-Fi Router	2,000/=
06.	Electrical components	2,000/=
Total		44,000/=

Software Price: For making this system required software list and prices are

S.N	Component	Price (BDT)
06.	MS Office	10,000/=
07.	Browser (Free)	0/=
08.	Adobe Create Cloud	10,000/=
09.	OS (Windows)	10,000/=
10.	Visual Basic	15,000/=
Total		40,000/=

Hosting Price:

S.N	Hosting Name	Price(BDT)
02.	https://www.StockManagementSystem.com	1,500/=
Total Price		1,500/=

6.1.2 Operational Feasibility:

By using this system different types of organization can handle their stock dynamically. There will be one user in this system called admin. To manage stock he can setup category, product, customer, supplier, can know current status of stock, can see periodically report. Also can take decision based on stock status. Here I will provide better type of operational feasibility for my system:-

- All kind of user such as customer and admin easily access to my system.
- All kind of product added easily.
- All kind of stock maintain easily.

6.1.3 Technical Feasibility

To run developed system which is a web based system we need just one computer and one server. Where PC is in the client end and server is in server end. All the information pushed through the system will stored in server. According necessity data will be fetched from server to the client end.

Hardware	PC/Laptop (4 GB RAM, 1TB HDD), Wi-Fi Router
Software	<ul style="list-style-type: none">• Adobe creative Cloud• MS office 2016• OS windows-10• Visual Basic.

6.2 Cost Benefit Analysis

Cost Estimation:

Basic for	Cost (BDT)
Hardware	30,000/=
Software	30,000/=
Hosting	1,500/=
Others	10.000/=
Maintenance	5,000/=
Implementation	10,000/=
Total	86 ,500/=

Benefit Estimation:

Basic for	Benefit (BDT)
Business Improvement	50,000/=
Improvement of Productivity	20,000/=
Saving from Changes	30,000/=
Total	1,00,000/=

6.3 DSDM – Good or Not for this project:

DSDM is a very good approach and we should use DSDM approach. Reasons behind using DSDM approaches are given below:

- It provides techniques independent solution and process
- very much flexible during requirement gathering and evaluating
- Time frame is very much strict
- Tester is required in each team
- Designed from root level by corporate people
- has approach to specify each task impotency

So we can say that, DSDM is good for this proposed system

Chapter 7 - Foundation

7.1 Overall requirement list

There are two types of requirement in software engineering and they are functional requirement and non-functional requirement. Functional and non-functional requirements of this proposed system are given below:

List of functional requirement:

9. Category setup: Admin can setup category to setup products under each category. He can edit and delete category
10. Product setup: Admin can setup product under each category. He also can set product reorder level that's why system can inform him if product is going below reorder level. There are also an option from where admin can easily access if needed during add product under category.
11. Supplier setup: Admin can setup supplier with their details so that he can track from where they purchase product. Here also he can edit and delete supplier.
12. Customer setup: Admin can setup customer with their details so that he can track to whom they sale their product. Here also he can edit and delete customer.
13. Purchase: Admin can purchase product under supplier multiple times. Here product code and total cost of product will auto appear, if admin set unit price and quantity. Admin can delete purchase before save it to database because there is a partial table where purchase store temporarily before sending it to database.
14. Sale: Admin can sell product to customer and he can sell product multiple times. Here customer will rewarded on specific product according their loyalty point. Admin can delete sell before save it to database because there is a partial table where sell store temporarily before sending it to database.
15. Stock: Admin can check stock for expire product, sold product, damage product etc. He can also which product is going under reorder level. System will notify to admin to restore product again. He can check how many product are in, how many products are out, product opening balance, product closing balance etc.

16. Report: Admin can see report periodically on sale and purchase. He can also check business loss and profit. He can make decision based on statistics.

List of non-functional requirements:

7. System validation and verification
8. System security
9. Reliability
10. Response time of each function
11. Testability
12. Efficiency of system

7.2 What technology to be implemented:

Client server application

Client server application is a kind of application which installed on client side and it runs on client machine. Core features of client server application is given below:

- Need to be installed on client pc.
- Against client request server sends response to it
- Due to increasing uses of mobile service, it will add more values to customer
- As they follow common protocol, it is very easy to communicate with each other's (client and server).
- Server can accommodate only limited number of request

Web server application:

It's a kind of application which doesn't need to be installed on client machine. Client can use this type of application by using HTTP communication process. Core features of web server application are given below:

- It can be accessible from anywhere, anytime through internet connection if you have capable device.
- It doesn't required to be installed on client machine
- Availability of storage (cloud)
- Up to date always and upgrade and updates are very fast.
- Web based application's interface is very much easier and user friendly than others
- More secured than other systems
- It can also run from mobile device

7.3 Recommendations and justifications:

In order to find out best possible solution for our proposed system and we have already discussed about different types of application and its core features and advantages for our proposed system.

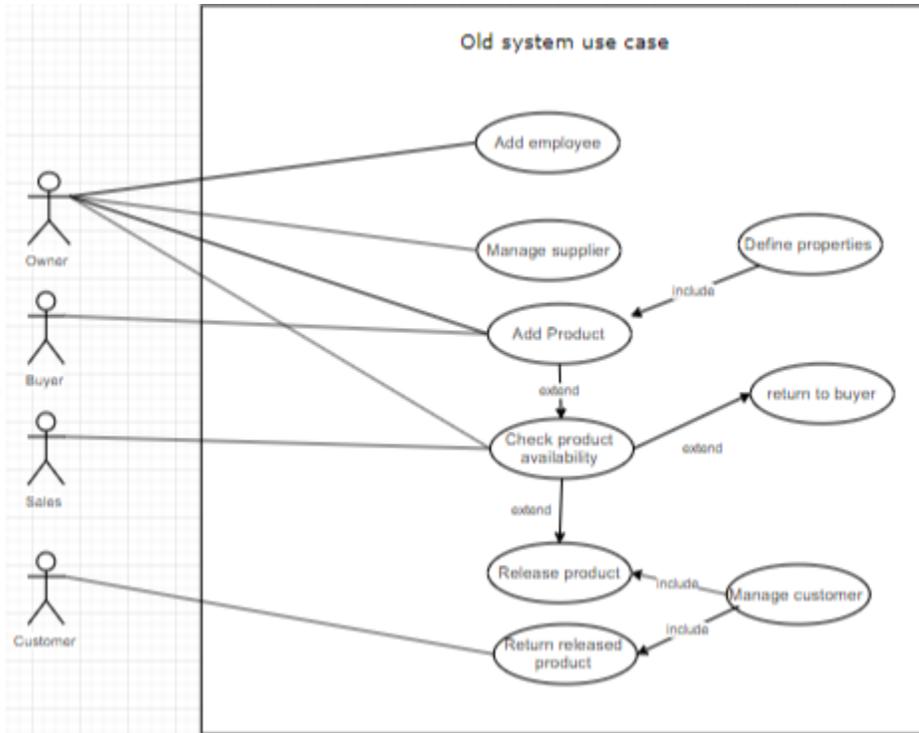
After comparing I think web server means web based application will be best suited for our proposed system. It has lot of features and advantages such as accessible from anywhere through internet connection, effective cost, doesn't need to be installed on any devices, fast and easy updating etc. So after all it is clear that web server application is best for our proposed system.

Chapter 8 - Exploration

Main purpose of exploration is find out requirements and better understanding of system by using different types of tools like diagram, prioritization etc.

8.1 Old full system use case:

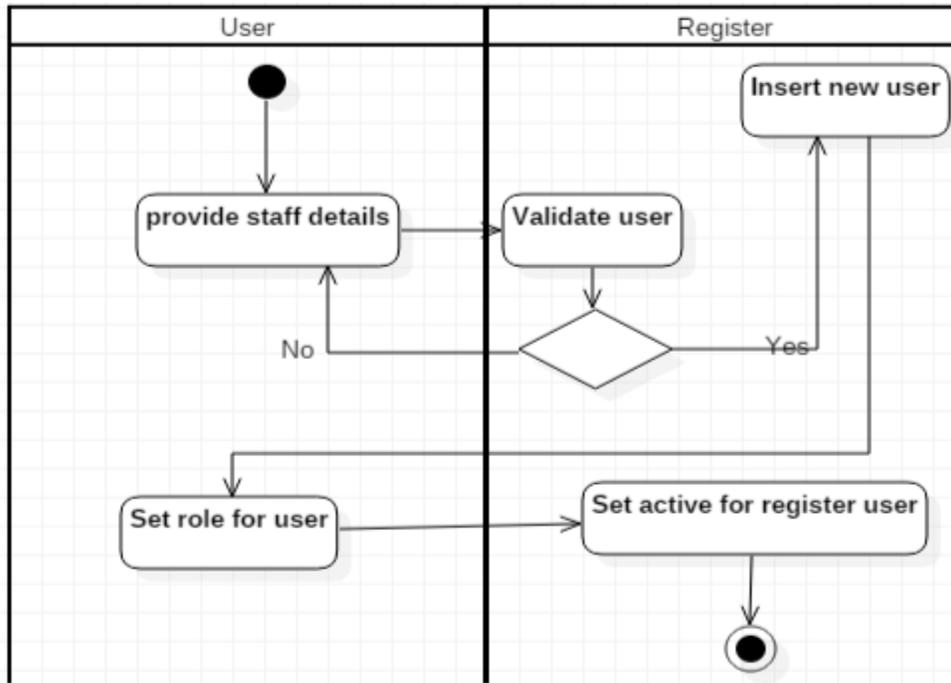
Use case is an easy and simplest representation of interaction between user and system. Here I have provide use case for old system:



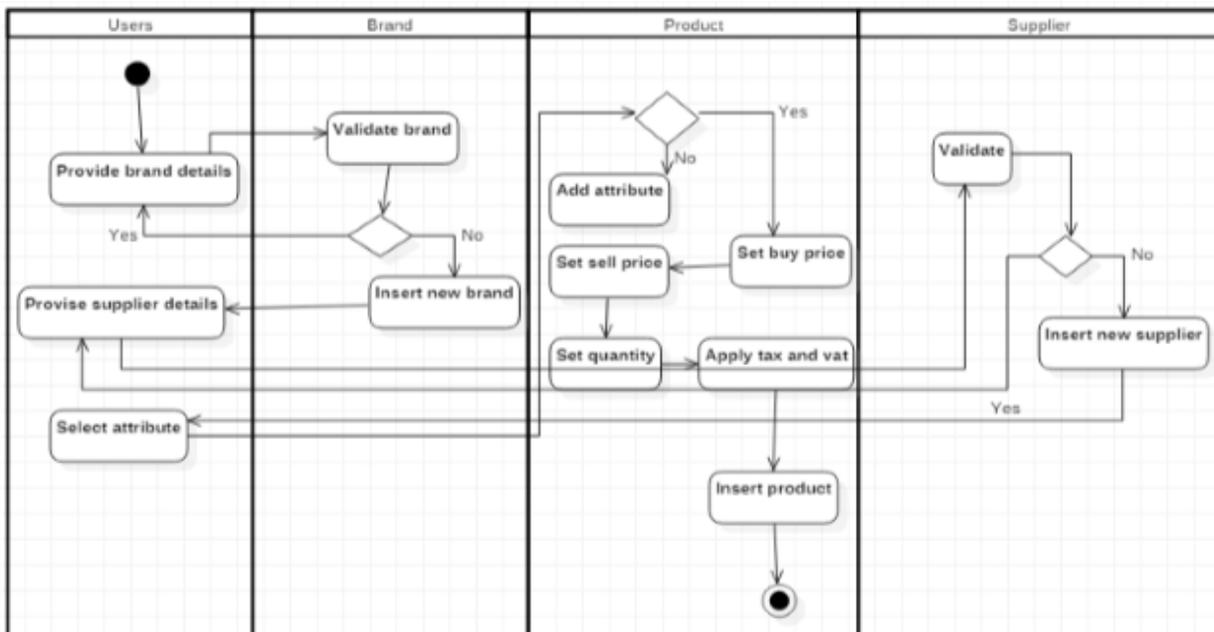
8.2 Old system full activity diagram:

Activity diagram is a visual representation of a series of a system. Here I have provide an activity diagram for old inventory management system:

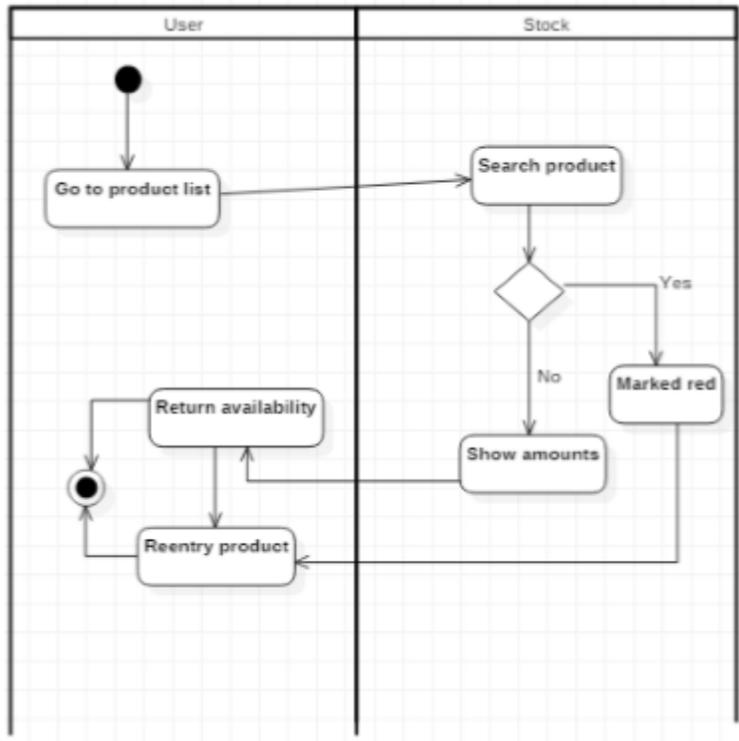
User management:



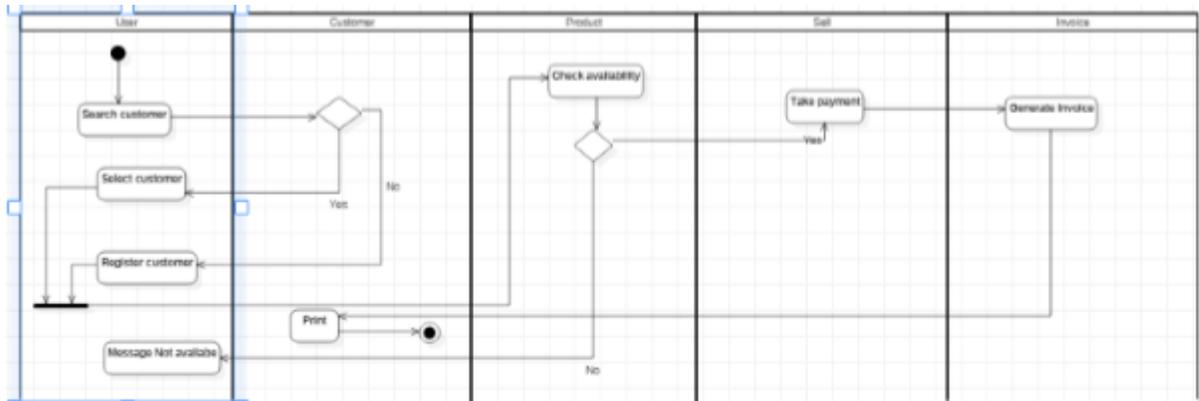
Buy product:



Availability of product:

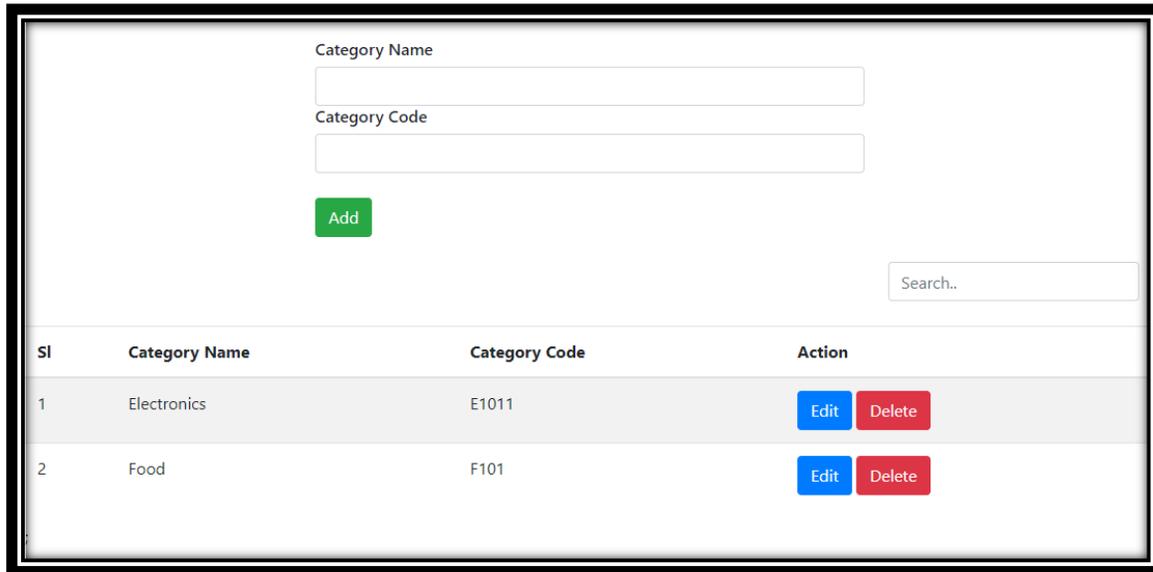


Product release:



8.3 Prototype of new system:

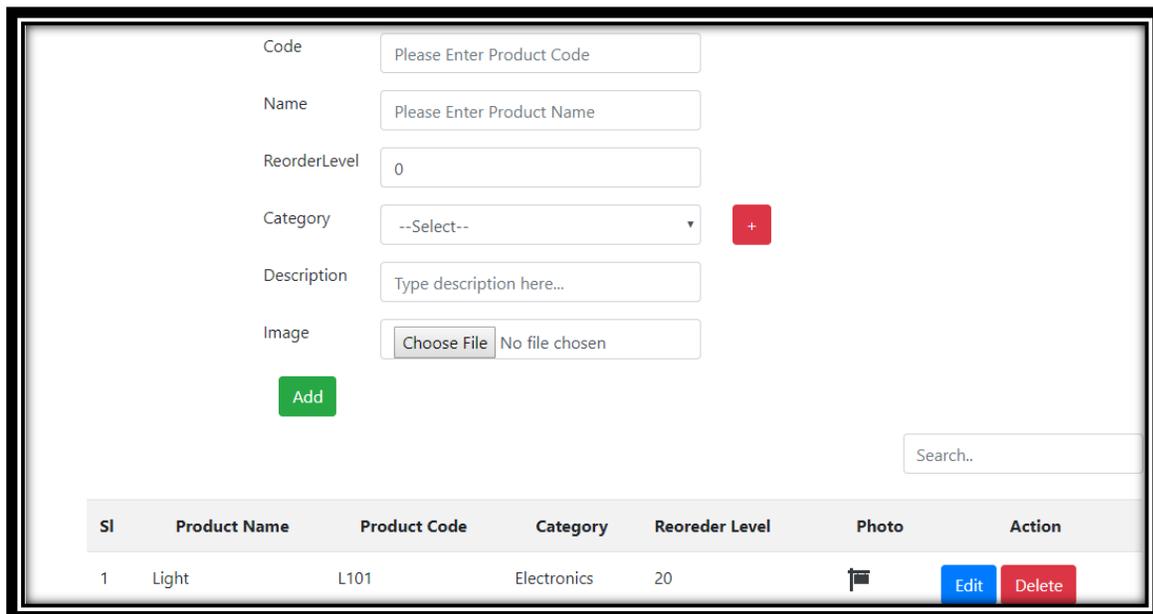
Category Setup: Prototype of category setup is given below



The image shows a web form for category setup. It includes two input fields: 'Category Name' and 'Category Code'. Below these is a green 'Add' button. To the right is a search box labeled 'Search..'. Below the form is a table with the following data:

SI	Category Name	Category Code	Action
1	Electronics	E1011	Edit Delete
2	Food	F101	Edit Delete

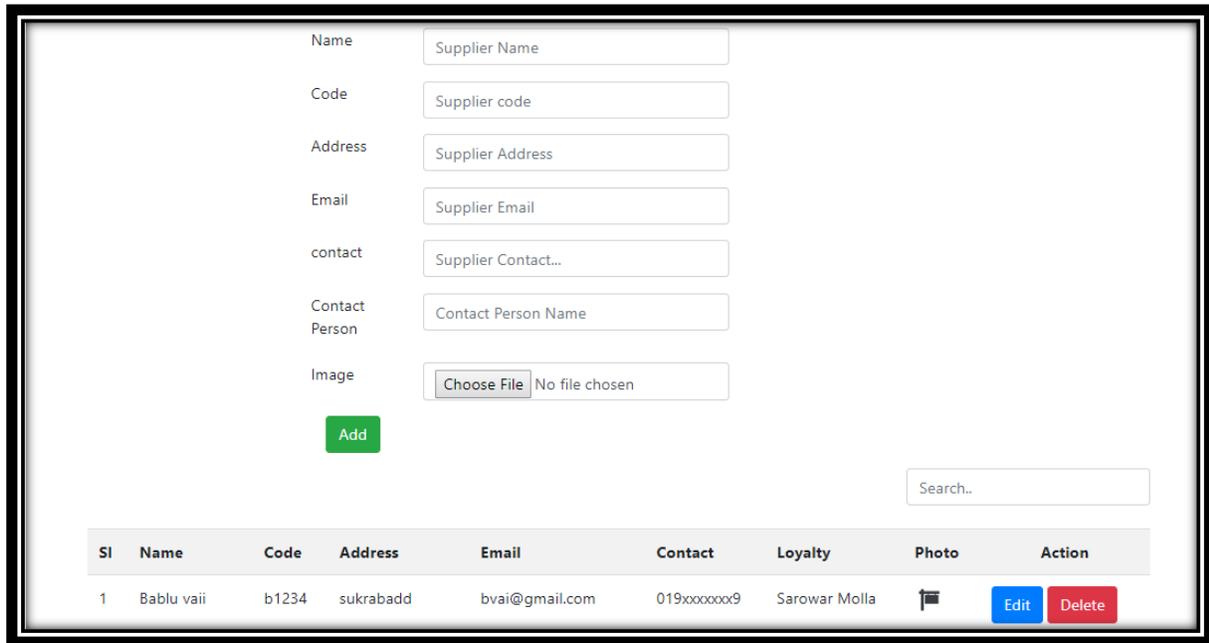
Product setup: Prototype of product setup is given below



The image shows a web form for product setup. It includes several input fields: 'Code' (placeholder: 'Please Enter Product Code'), 'Name' (placeholder: 'Please Enter Product Name'), 'ReorderLevel' (value: '0'), 'Category' (dropdown menu: '--Select--'), 'Description' (placeholder: 'Type description here...'), and 'Image' (button: 'Choose File', text: 'No file chosen'). Below these is a green 'Add' button. To the right is a search box labeled 'Search..'. Below the form is a table with the following data:

SI	Product Name	Product Code	Category	Reorder Level	Photo	Action
1	Light	L101	Electronics	20		Edit Delete

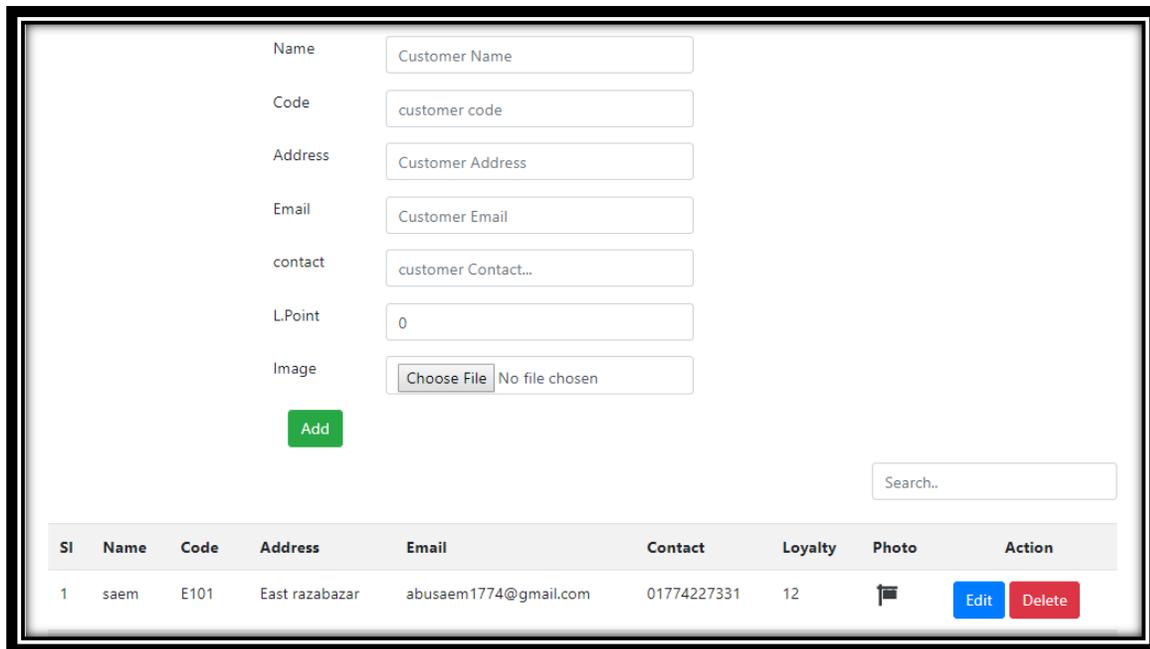
Supplier setup: Prototype of supplier setup is given below



The image shows a web form for adding a new supplier. The form includes input fields for Name, Code, Address, Email, contact, Contact Person, and Image. Below the form is a green 'Add' button and a search bar. At the bottom, there is a table listing existing suppliers.

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	Bablu vaii	b1234	sukrabadd	bvai@gmail.com	019xxxxxxx9	Sarowar Molla		Edit Delete

Customer setup: Prototype of customer setup is given below



The image shows a web form for adding a new customer. The form includes input fields for Name, Code, Address, Email, contact, L.Point, and Image. Below the form is a green 'Add' button and a search bar. At the bottom, there is a table listing existing customers.

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	saem	E101	East razabazar	abusaem1774@gmail.com	01774227331	12		Edit Delete

Purchase: Purchase prototype is given below

Add Some Data

Supplier

SupplierID

Date

invoiceNo

Product

ProductID

Code

manufactureDate

expireDate

expireDate

remarks

quantity

unitPrice

Total Price

Previous Cost Price

Previous MRP

newMRP

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
<input type="button" value="Save"/>									

Sale: Sale prototype is given below

Customer

Customer:

Date:

Loyalty point:

Product

ProductID:

Available Quantity:

Quantity:

MRP:

Total MRP:

Product Details

SI	Product	Quantity	MRP	Total MRP	Action
	Payable				<input type="text" value="Payable"/>
	Discount(%)				<input type="text" value="Discount(%)"/>
	Discount				<input type="text" value="Discount"/>
	Grand-Total				<input type="text" value="0"/>

Stock: Prototype of stock status is given below

Stock

Product

Category

Reorder Level

Expire

Start Date

End Date

SI	Code	Name	Category	Reorder Level	Ex. Date	Ex. Qty	Opening Blance	IN	Out	Closing Blance
1	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	20	440	12	22	500
2	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	50	600	12	12	500
3	M101	Mobile	Electronics	12	12/14/2019 12:00:00 AM	10	100	12	10	500
4	Mob11	iPhone	Electronics	5	12/23/2019 12:00:00 AM	1	12	45	12	40

Report: Prototype of report is given below

Date

Date

SI	Code	Name	CP(TK)	MRP	Profit
1	L101	Electronics	10	22	440
2	L101	Electronics	10	12	600
3	M101	Electronics	10	10	100
4	Mob11	Electronics	20	12	12

Chapter 9 - Engineering

9.1 New system modules:

There are different type of modules in my proposed system which are developed according to customer requirements. These are given below:

- Product Catalog a. Product Category Setup (Entry, Edit, Search) b. Product Setup with Category (Entry, Edit, Search)
- Party Module a. Customer Setup (Entry, Edit, Search)
- Supplier Setup (Entry, Edit, Search)
- Purchase Module a. Purchase Entry b. Purchase Edit
- Purchase Confirm d. Purchase Search
- Sales Module a. Sales Entry b. Sales Edit c. Sales Confirm d. Sales Search
- Stock Module a. Periodical Stock Report (From - To, Opening Balance, In, Out, Closing Balance) b. Find Low product & Expired product.
- Reporting Module a. Periodic Income Expense Report on Sales b. Periodic Income Expense Report on Purchase

9.2 Use case Diagram:

Use case diagram for stock management system is given below:

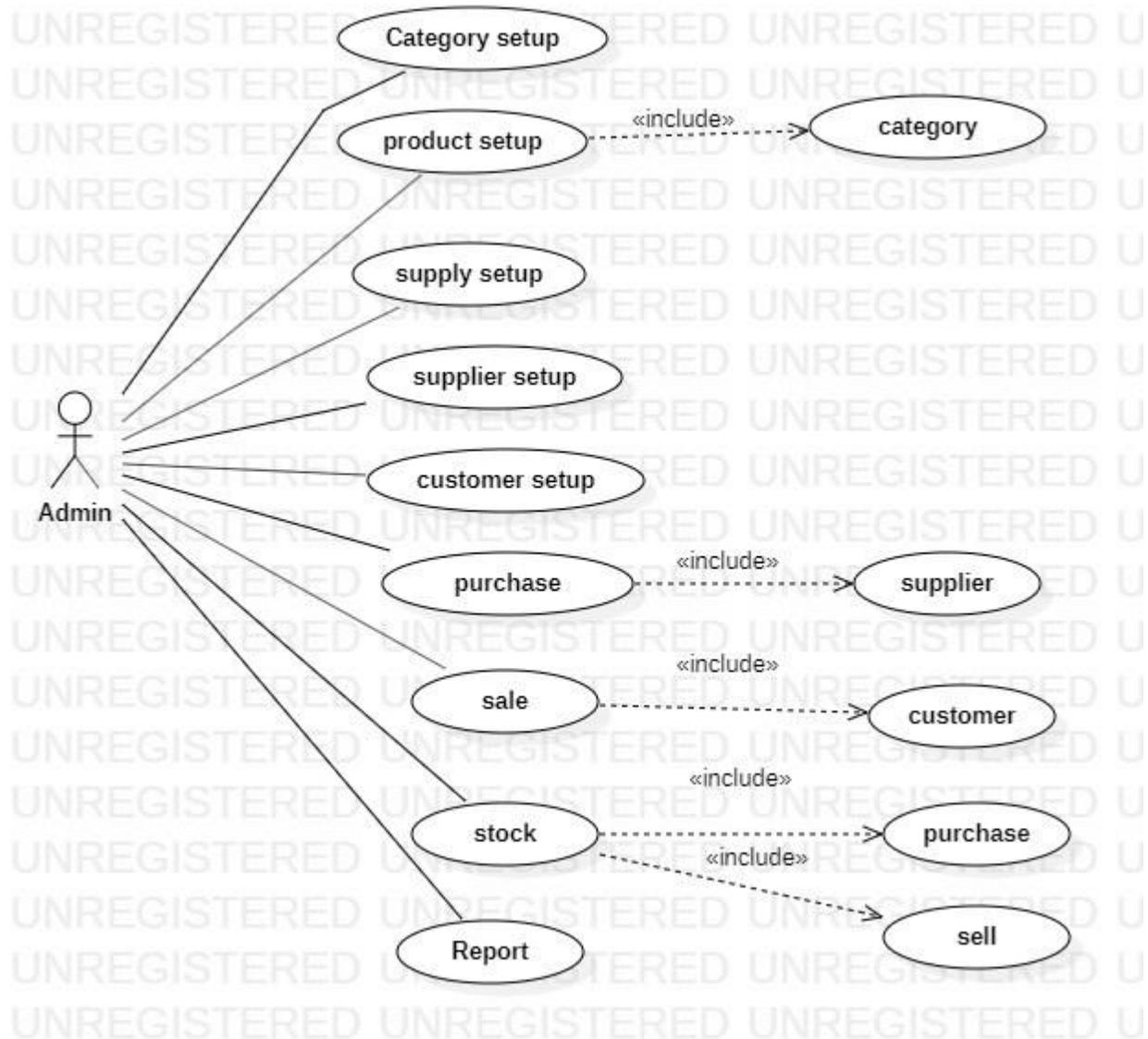


Figure 9: Use case diagram for stock management system

9.3 Class diagram

Class diagram for stock management system is given below:

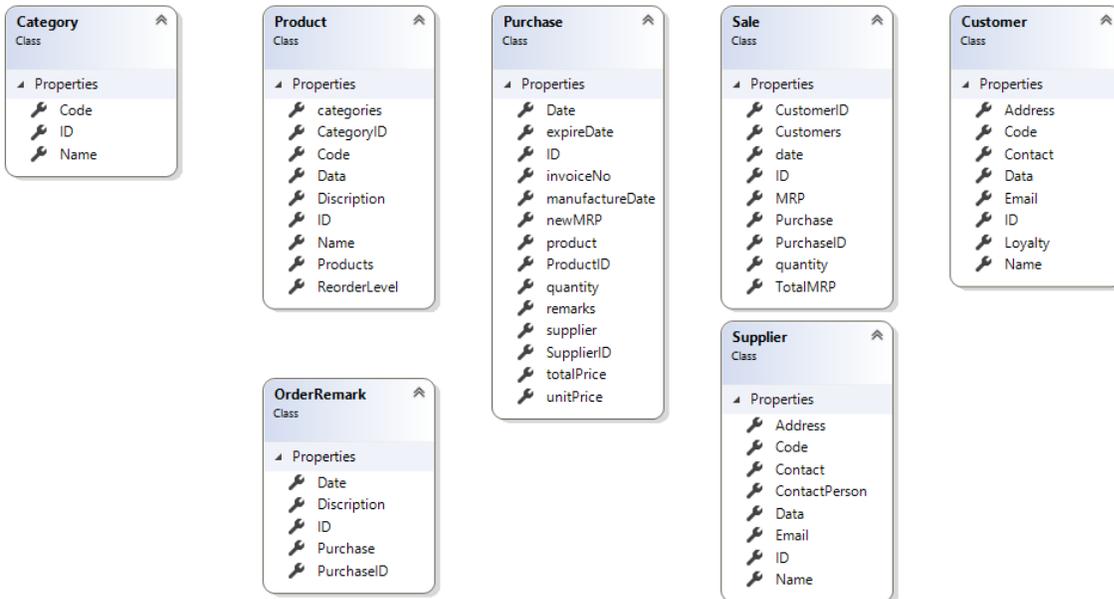


Figure 10: Class diagram of stock management system

9.4 Sequence Diagram

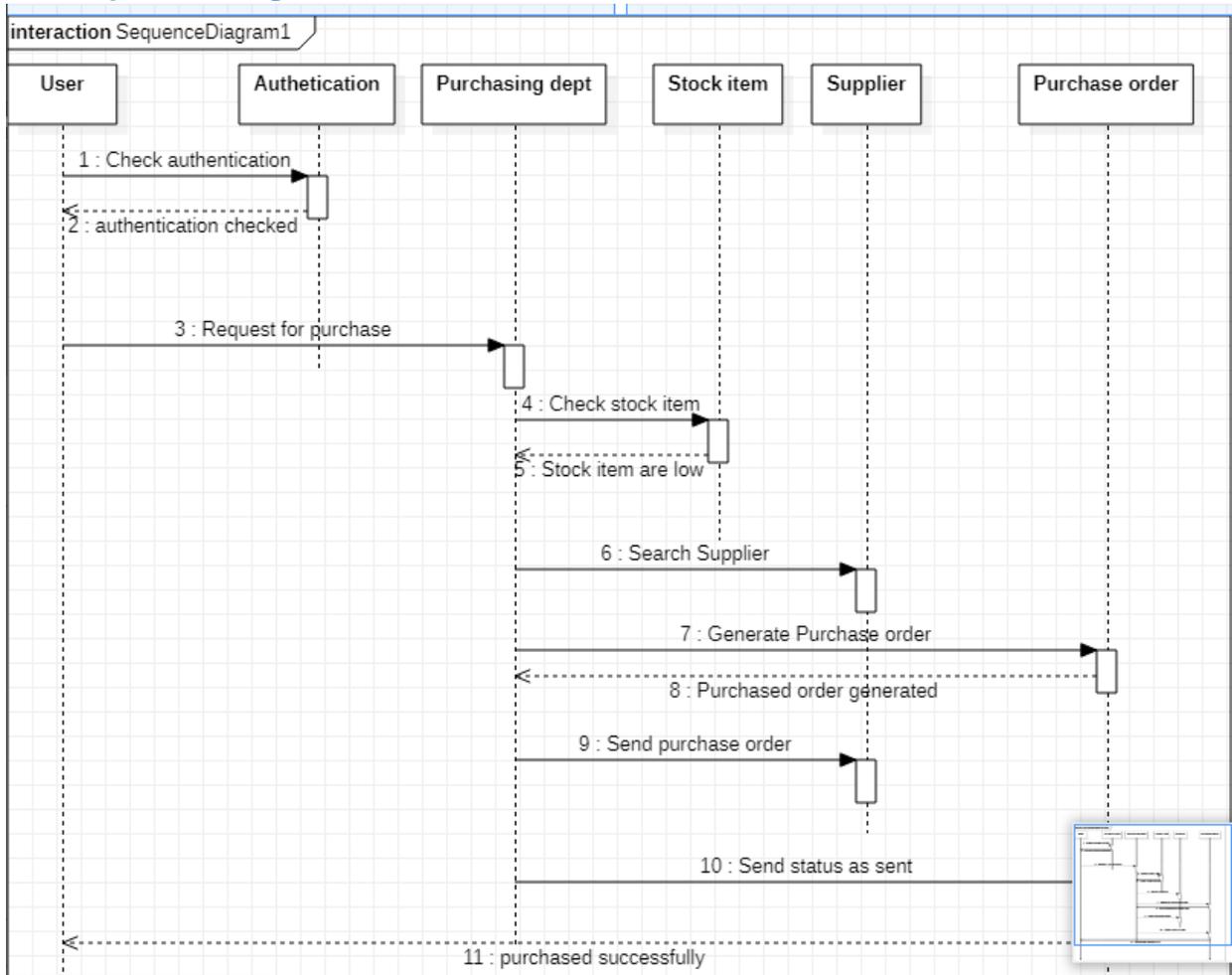


Figure 11: Sequence diagram for stock management system

9.5 System Interface

Customer Setup:

In customer setup module here admin can setup category to setup products under each category. He can edit and delete category. There is also an option for admin to search category what he wants to see or wants to manipulate.

Category Name

Category Code

SI	Category Name	Category Code	Action
1	Electronics	E1011	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Food	F101	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

;

Category setup:

In category setup module he can setup product under each category. He can edit and delete each product. He also can set product reorder level that's why system can inform him if product is going below reorder level. There are also an option from where admin can easily access if needed during add product under category. He can also search specific category from list of products.

Code

Name

ReorderLevel

Category

Description

Image

SI	Product Name	Product Code	Category	Reoreder Level	Photo	Action
1	Light	L101	Electronics	20		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Supplier setup:

He can setup supplier with their details so that he can track from where they purchase product. Here also he can edit and delete supplier. He can also search specific supplier

Name

Code

Address

Email

contact

Contact Person

Image No file chosen

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	Bablu vaii	b1234	sukrabadd	bvai@gmail.com	019xxxxxx9	Sarowar Molla		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Customer setup:

He can setup customer with their details so that he can track to whom they sale their product. Here also he can edit and delete customer. He can also search specific customer

Name

Code

Address

Email

contact

L.Point

Image No file chosen

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	saem	E101	East razabazar	abusaeem1774@gmail.com	01774227331	12		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Purchase:

Admin can purchase product under supplier multiple times. Here product code and total cost of product will auto appear, if admin set unit price and quantity. Admin can delete purchase before save it to database because there is a partial table where purchase store temporarily before sending it to database.

Add Some Data

Supplier

SupplierID

Date

invoiceNo

Product

Check Remarks

ProductID

Code

manufactureDate

expireDate

expireDate

remarks

quantity

unitPrice

Total Price

Previous Cost Price

Previous MRP

newMRP

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
----	---------------	---------	----------	----------	------------	-------------	-----------	---------	--------

Save

Sale:

©Daffodil International University

50

Admin can sell product to customer and he can sell product multiple times. Here customer will rewarded on specific product according their loyalty point. Admin can delete sell before save it to database because there is a partial table where sell store temporarily before sending it to database.

Customer

Customer

Date

Loyalty point

Product

ProductID

Available Quantity

Quantity

MRP

Total MRP

Product Details

SI	Product	Quantity	MRP	Total MRP	Action
Payable			<input type="text" value="Payable"/>		
Discount(%)			<input type="text" value="Discount(%)"/>		
Discount			<input type="text" value="Discount"/>		
Grand-Total			<input type="text" value="0"/>		

Stock:

Admin can check stock for expire product, sold product, damage product etc. He can also which product is going under reorder level. System will notify to admin to restore product again. He can check how many product are in, how many products are out, product opening balance, product closing balance etc.

Stock

Product	<input type="text"/>	Start Date	<input type="text" value="mm/dd/yyyy"/>
Category	<input type="text"/>	End Date	<input type="text" value="mm/dd/yyyy"/>
<input type="checkbox"/> Reorder Level <input type="checkbox"/> Expire			
<input type="button" value="Submit"/>			

SI	Code	Name	Category	Reorder Level	Ex. Date	Ex. Qty	Opening Blance	IN	Out	Closing Blance
1	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	20	440	12	22	500
2	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	50	600	12	12	500
3	M101	Mobile	Electronics	12	12/14/2019 12:00:00 AM	10	100	12	10	500
4	Mob11	iPhone	Electronics	5	12/23/2019 12:00:00 AM	1	12	45	12	40

Report:

Admin can see report periodically on sale and purchase. He can also check business loss and profit. He can make decision based on statistics.

Date	<input type="text" value="mm/dd/yyyy"/>
Date	<input type="text" value="mm/dd/yyyy"/>
<input type="button" value="Search"/>	

SI	Code	Name	CP(TK)	MRP	Profit
1	L101	Electronics	10	22	440
2	L101	Electronics	10	12	600
3	M101	Electronics	10	10	100
4	Mob11	Electronics	20	12	12

Chapter 10 - Deployment

After completing all of analysis deployment/development starts from this chapter. I have divided all of my task into small and specific portion and then it was very easy to develop all of those task.

10.1 Core modules coding samples of my system:

Category setup coding samples:

```
15     StockManageManger _stockManageManger = new StockManageManger();
16     Category _category = new Category();
17     CategoryVM _categoryvm = new CategoryVM();
18
19     public ActionResult Add()
20     {
21         _categoryvm.categories = _stockManageManger.Show();
22
23         return View(_categoryvm);
24     }
25
26     [HttpPost]
27     public ActionResult Add(Category category)
28     {
29         if (ModelState.IsValid)
30
31         {
32             bool invalid = true;
33             if (category.Name != null && category.Code != null)
34             {
35                 foreach (var item in _stockManageManger.Show())
36                 {
37                     if (item.Name==category.Name)
38                     {
39                         invalid = false;
40                     }
41                 }
42             }
43             if(invalid!=false)
44             {
45                 _stockManageManger.Add(category);
46                 ViewBag.message = "Category has been added ";
47                 ViewBag.alert = "alert alert info";
48             }
49         }
50     }
51 }
```

Product setup coding samples:

```
15 // GET: Product
16 StockManageProductManager _stockManageManger = new StockManageProductManager();
17
18 StockManageManager _stockManageManagerCategory = new StockManageManager();
19 Product _product = new Product();
20 // ProductVM _productvm = new ProductVM();
21
22
23 0 references
24 public ActionResult Add()
25 {
26     ProductVM _productvm = new ProductVM();
27     _productvm.CategoryList = _stockManageManagerCategory.Show()
28         .Select(c => new SelectListItem()
29             {
30                 Value = c.ID.ToString(),
31                 Text = c.Name
32             }).ToList();
33
34
35     _productvm.products = _stockManageManger.Show();
36
37     return View(_productvm);
38 }
39 [Route("Add")]
40 [HttpPost]
41 0 references
42 public ActionResult Add(ProductVM model)
43 {
44     Product product = Mapper.Map<Product>(model);
45     HttpPostedFileBase image = Request.Files["ImageData"];
46     var name = _stockManageManger.Show();
47     bool isValid = true;
```

Customer setup coding samples:

```
41         //     return view();
42         //}
43
44         [Route("Add")]
45         [HttpPost]
46         0 references
47         public ActionResult Add(CustomerVM model)
48         {
49             if (ModelState.IsValid)
50             {
51                 Customer customer = Mapper.Map<Customer>(model);
52
53                 HttpPostedFileBase image = Request.Files["ImageData"];
54
55                 _stockManager.Add(customer, image);
56             }
57             _customerVM.customers = _stockManager.Show();
58             return View(_customerVM);
59         }
60
61
62
63         [HttpPost]
64         0 references
65         public ActionResult Show(CustomerVM model)
66         {
67             Customer customer = Mapper.Map<Customer>(model);
68             if (customer.Name != null)
69             {
70                 _customerVM.customers = _stockManager.search(customer);
71             }
72         }
```

Supplier setup coding samples:

```
{
    StockManageSupplierManager _stockManageManger = new StockManageSupplierManager();
    StockManageDbContext Db = new StockManageDbContext();
    Supplier _supplier = new Supplier();
    SupplierVM _supplierVM = new SupplierVM();

    0 references
    public ActionResult Add()
    {
        SupplierVM model = new SupplierVM();
        model.suppliers = _stockManageManger.Show();

        return View(model);
    }

    [Route("Add")]
    [HttpPost]
    0 references
    public ActionResult Add(SupplierVM model)
    {
        if (ModelState.IsValid)
        {
            Supplier supplier = Mapper.Map<Supplier>(model);

            HttpPostedFileBase image = Request.Files["ImageData"];

            _stockManageManger.Add(supplier, image);
        }
        _supplierVM.suppliers = _stockManageManger.Show();
        return View(_supplierVM);
    }
}
```

Product purchase coding samples:

```
0 references
public class PurchaseController : Controller
{
    bool add = false;
    PurchaseManager _stockManageManger = new PurchaseManager();
0 references
    public ActionResult Add()
    {
        PurchaseVM model = new PurchaseVM();

        model.ProductList = _stockManageManger.ShowProduct()
            .Select(c => new SelectListItem
            {
                Value = c.ID.ToString(),
                Text = c.Name
            })
            .ToList();

        model.SupplierList = _stockManageManger.ShowSupplier()
            .Select(c => new SelectListItem
            {
                Value = c.ID.ToString(),
                Text = c.Name
            })
            .ToList();

        if(add==true)
        {
            ViewBag.mess = "Add Purchase Success";
        }
        else
        {
            ViewBag.mess = "Add Gene Data";
        }
    }
}
```

Product sale coding samples:

0 references

```
public class SaleController : Controller
```

```
{
```

```
    StockManageSaleManager _stockManageManger = new StockManageSaleManager();
```

```
    PurchaseManager _perchaseManager = new PurchaseManager();
```

```
    Sale _supplier = new Sale();
```

```
    Customer _Customer = new Customer();
```

```
    // GET: Sales
```

0 references

```
public ActionResult Index()
```

```
{
```

```
    return View();
```

```
}
```

0 references

```
public ActionResult Add()
```

```
{
```

```
    var model = new SalesVm();
```

```
    model.CustomerList = _stockManageManger.Show()
```

```
        .Select(c => new SelectListItem
```

```
        {
```

```
            Value = c.ID.ToString(),
```

```
            Text = c.Name
```

```
        })
```

```
        .ToList();
```

```
    var List = _stockManageManger.ShowProduct()
```

```
        .Select(c => new SelectListItem
```

```
        {
```

```
            Value = c.ID.ToString(),
```

```
            Text = c.product.Name
```

```
        }) .ToList();
```

Stock check coding samples:

```
//model.Purchases = _manager.purchases();

if (model.Reordercheck == null && model.Expirecheck == null)
{
    model.Sales = (Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.Date >= model.sDate && c.Purchase.Date <=
        model.eDate).ToList();
}
else if (model.Reordercheck == "true" && model.Expirecheck == null)
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.quantity <= c.Purchase.product.ReorderLevel).ToList();
}

else if (model.Reordercheck == "true" && model.Expirecheck == "true")
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.quantity <= c.Purchase.product.ReorderLevel && c.Purchase.expireDate <= model.eDate).ToList()
}

else if (model.Reordercheck == null && model.Expirecheck == "true")
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.Purchase.expireDate <= model.eDate).ToList();
}

//model.stockoutRIana=model.Purchases.Where(c=>c \
```

Report generate coding samples:

```
using StockManage.BLL.BLL;
using StockManage.DatabaseContext.DatabaseContext;
using StockManageWeb.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace StockManageWeb.Controllers
{
    0 references
    public class ReportController : Controller
    {
        StockManageDbContext Db = new StockManageDbContext();
        StockManager _manager = new StockManager();
        0 references
        public ActionResult Index()
        {
            Report model = new Report();
            model.ProductList = _manager.ShowProduct().Select(c => new SelectListItem { Value = c.ID.ToString(), Text = c.Name }).ToList();

            model.Sales = _manager.sales();
            return View(model);
        }
        0 references
        [HttpPost]
        0 references
        public ActionResult Index(Report model)
        {

            model.Sales = Db.sales.Where(c => c.Purchase.Date >= model.sDate && c.Purchase.Date <= model.eDate).ToList();

            return View(model);
        }
    }
}
```

10.2 Possible problem breakdown

In this chapter, first of all I have find out possible problems which have to develop. Then I divided all the possible problem into multiple small task to complete easily. Possible problem breakdowns of my proposed system is given below:

Database designing and query manipulation:

- Gather data from sources
- Create ERD
- Create database
- Manipulate query

Front-End designing:

- Create prototype on paper
- Then design on IDE using html, CSS and bootstrap
- Link pages with each other's

Back-End Developing:

Purchase product:

- Develop category setup function
- Develop Product add function
- Develop Supplier setup function

Sale product:

- Develop category setup function
- Develop Product add function
- Develop Customer setup function

Stock checking:

- Develop category setup function
- Develop Product add function
- Develop Customer setup function
- Develop Supplier setup function
- Purchase product setup
- sale product setup

Chapter 11 - Testing

Different types of testing and test cases are explained here. Testing is must do parameter in software development to check all of requirements are met or not. It works to detect the difference between actual and expected result.

Test case:

11.1 Unit Testing

Username:	MD. ABU SAEM		Role:	Tester
SI No	Test name	Expected Result	Success Factor	Comments
1.	Category Setup	Category add and show them	Category can be added successfully	Successful
2.	Customer setup	Customer add and show them	Customer can be added successfully	Successful
3.	Product Setup	Product add and show them	Product showed successfully	Successful

Result:

Category setup:

Category has been added

Category Name

Category Code

SI	Category Name	Category Code	Action
1	Electronics	E101	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 12: Category added and showing successfully

Customer setup:

Name

Code

Address

Email

contact

L.Point

Image No file chosen

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	saem	S101	East razabazar	saem@gmail.com	01774227331	12		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 13: Customer added and showing successfully

Product Setup:

Code

Name

ReorderLevel

Category

Description

Image No file chosen

SI	Product Name	Product Code	Category	Reoreder Level	Photo	Action
1	Light	L101	Electronics	10		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 14: Product added and showing successfully

11.2 Module Testing

Username:	MD. ABU SAEM	Role:	Tester	
SI No	Test name	Expected Result	Success Factor	Comments
1.	Product Purchasing	Product add then purchase	Purchase should be successful	Successful

Category Add:

First of all I have to add category to add product under each category.

Category Name

Category Code

SI	Category Name	Category Code	Action
1	Electronics	E101	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 15: Category added successfully

Product Setup:

In order to purchase product, add product under each category

Code

Name

ReorderLevel

Category 

Description

Image No file chosen

SI	Product Name	Product Code	Category	Reorder Level	Photo	Action
1	Light	L101	Electronics	10		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 16: Product added successfully

Add Supplier:

In order to purchase add supplier:

Name

Code

Address

Email

contact

Contact Person

Image No file chosen

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	ABU SAEM	As1001	west razabazar	saem@gmail.com	01774227331	kamal		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 17: Supplier added successfully

Purchase product from supplier:

Supplier

SupplierID	ABU SAEM
Date	06/20/2020
invoiceNo	P101

Product

[Check Remarks](#)

ProductID	Light
Code	
manufactureDate	06/01/2020
expireDate	06/30/2020
remarks	100 v Light
quantity	5
unitPrice	200
Total Price	1000
Previous Cost Price	Previous Cost Price
Previous MRP	Previous MRP
newMRP	250

[Add](#)

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
1	LightNaN		2020-06-01	2020-06-30	100 v Light	5	200	1000	250 Delete

[Save](#)

Figure 18: Purchase product from supplier

11.3 Integration Testing

Username:	MD. ABU SAEM		Role:	Tester
SI No	Test name	Expected Result	Success Factor	Comments
1.	Sale product	Product purchase then sale	Selling product should be successful	Successful

Purchase product: Purchase product at first in order to sale

Supplier

SupplierID:

Date:

invoiceNo:

Product

Check Remarks

ProductID:

Code:

manufactureDate:

expireDate:

remarks:

quantity:

unitPrice:

Total Price:

Previous Cost Price:

Previous MRP:

newMRP:

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
1	LightNaN		2020-06-01	2020-06-30	100 v Light	5	200	1000	250 <input type="button" value="Delete"/>

Figure 19: Purchase product from supplier

Sale product:

Customer: --Select--

Date: 06/21/2020

Loyalty point: []

Product

ProductID: Light

Available Quantity: 3

Quantity: 1

MRP: 300

Total MRP: 300

Add

Product Details

SI	Product	Quantity	MRP	Total MRP	Action
0	Light	1	300	300	Delete

Payable: NaN

Discount(%): Discount(%)

Discount: NaN

Grand-Total: 300

Save

Figure 20: Sold product successfully

Chapter 12 - Implementation

12.1 Training

Here all types of user will learn how to use the system and I will describe the Training area, Time limit for Training and details of the training.

As my system has only one user called admin, so I have to just train admin to run this system properly:

Serial No.	Name of the User	Training Area	Time Limit	Details
01	Admin			
		Each module	About 2 hour	Admin can understand how to conduct the system

12.2 System Implementation

There are four types of implementation process in projects. Using in those process system has been implemented in the real world. I will describe those process.

- **Big Bang:** it is a process of implementing a new system by turning off the old system. In that process, data loss can be happening because this process is much quicker than others.
- **Direct Implementation:** it is a process of implementing a system by removing the old system after completing all the testing procedure. In that process there will be no existing of old systems.

- **Pilot:** it's a process of running new system as test implementation purposes. In that process in some organizations, new system has been implemented. After a several period of times, if all the features of system have been working well, then the final installation will be done of the system.
- **Parallel:** it is a process of running new system with the old system and the old system works as the backup of the new system.

12.3 Recommended implementation process

I think the pilot schema implementation process can be install because it is the process of implementing system in various branches of organization as pilot testing purposes. If all the features of a system work as expected, then the system will launch finally for real world implementation.

Chapter 13 - Critical appraisal and evaluation

Here in this chapter I will evaluate and critically analyze the system and system objectives. So many objectives are identified in the above (previous part). It will help system to be more successful.

13.1 Objectives that could be met

Though I tried to meet all the objectives but all objectives are not met.

13.1.1 Success rate against each objective

Academic objectives:

- User friendly system
- Suitable methodology for this system has been used
- Feasibility study and risk analysis has been done
- Database designing and developing
- Analysis and documenting properly to make it rich system
- Standard and understandable documentation for all

Business objectives:

- Easy and proper view
- Purchase and sale product
- Generate report
- Show product status

Personal objectives:

- Different types of report
- Efficient and user friendly system

13.1.2 How much better could have been done

- Common search option for whole system
- Displaying product rating on product view
- Real time notification option

13.1.3 How better is the features of the solution?

- Proposed system should be full responsive
- It should be fully user friendly
- It should be efficient and smooth
- It should have performance parameter
- It should have option so that it can be maintained easily
- All the data should be secured

13.1.4 Which features couldn't be touched:

- Common search option
- Setting price range
- product rating on product

13.2.1 Why this feature couldn't be touched

Due to have time limitation and budget problem and less number of developer (only me) I have not touched these given (in above) features.

13.2 Objective totally not met / touched

- Common search option for whole system
- Setting price range
- Some types of report

13.2.1 Why it could not be touched

Though I tried to fulfill all the requirement but some things are not complete because of time limitation. Without having anyone help in deployment I completed this project and here I have played different types of role or all the role exist in software development. That's why I couldn't met all the mentioned requirements.

13.2.2 What could have been done?

As much as I know, hiring 4-6 developer could complete this project within fixed time frame but for that we have to have more budget. Without that we have to follow agile methodology for its proper focuses on early delivery and strategic goals. If we can follow above things then project can complete on time and added value in business,

Chapter 14 - Conclusion

14.1 Summary of the project

Within given time, I have completed all the mentioned task and I have tested it and working properly. During project time I have faced lot of problems and risks. I have worked hard to complete it within proposed time and budget. I have learned lot of thing during my project development and documentation time. This project taught me how to complete a project/ task within short time and how to complete it properly. I have given all the effort I had and tried to fulfill all the requirements. Adding some more features can make it more perfect to manage stock or warehouse. I hope, in future this project will contribute to manage warehouse properly.

14.2 Goals of the project

The main goal of the proposed system is managed stock or warehouse to prevent loss. This system was wanted as a supportive tool to manage stock or warehouse properly.

14.3 Success of the project

Basically, success rate of project depends on how many requirements or objectives has been completed properly. I think I have done 65 % of objectives and 35 % are rest. This rest portion will be further development. As I done this project in single hand, it was a very big challenge for me to complete this project within fixed time frame. Having 4-6 more developer is plus to complete this project successfully. Success ratio of project not depends on having any extra feature.

As I completed all the features so I can say that the project is successful.

14.4 Value of the project

Now a day's people all over the world wants everything in their hand and wants to does everything in technical way. This system will create big platform for small/big organization to manage and control their stock in dynamically. This project is a platform independent project. So it has high chances to gather a huge attention for business organization. It will create a new area in warehouse or stock management area.

14.5 My experiences

During my project completion, I have learned a lot of things such as how to collect data, how to complete task within fixed time frame, how to face problems and overcome it. Developing a full project in single hand was very tough but was great opportunity for me to handle it and complete it properly. I have completed all the requirements within fixed budget and time. Hope that this project knowledge will be very helpful in future for my career. I want to thanks almighty Allah who gave me all the physical and mental strength and also to my module leader for his great support.

References

Anon., 2020. *geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>
[Accessed 17 April 2020].

Anon., 2020. *javatpoint*. [Online]

Available at: <https://www.javatpoint.com/integration-testing>
[Accessed 17 april 2020].

Anon., 2020. *professionalqa*. [Online]

Available at: <https://www.professionalqa.com/module-testing>
[Accessed 17 april 2020].

Anon., 2020. *tutorialspoint*. [Online]

Available at: https://www.tutorialspoint.com/software_testing_dictionary/test_case.htm
[Accessed 17 april 2020].

Anon., n.d. [Online]

Available at: <https://osarome.blogspot.com/2011/10/1-technical-feasibility-2-operational.html>

Anon., n.d. [Online]

Available at: <https://www.salesbinder.com/>

Anon., n.d. [Online]

Available at: <https://www.tradegecko.com/>

Anon., n.d. [Online]

Available at: https://www.inflowinventory.com/?utm_expid=.5nDW8CphQjWfBA-Y1uENQQ.0&utm_referrer=https%3A%2F%2Fwww.google.com%2F

Anon., n.d. [Online]

Available at: <https://www.agilebusiness.org/page/TheDSDMAgileProjectFramework>

Anon., n.d. [Online]

Available at: <https://www.guru99.com/what-is-sdlc-or-waterfall-model.html>

Appendices:

Test case:

11.1 Unit Testing

Username:	MD. ABU SAEM		Role:	Tester
SI No	Test name	Expected Result	Success Factor	Comments
1.	Category Setup	Category add and show them	Category can be added successfully	Successful
2.	Customer setup	Customer add and show them	Customer can be added successfully	Successful
3.	Product Setup	Product add and show them	Product showed successfully	Successful

Result:

Category setup:

Category has been added

Category Name

Category Code

SI	Category Name	Category Code	Action
1	Electronics	E101	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

:

Figure 21: Category added and showing successfully

Customer setup:

Name

Code

Address

Email

contact

L.Point

Image No file chosen

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	saem	S101	East razabazar	saem@gmail.com	01774227331	12		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 22: Customer added and showing successfully

Product Setup:

Code

Name

ReorderLevel

Category

Description

Image No file chosen

SI	Product Name	Product Code	Category	Reoreder Level	Photo	Action
1	Light	L101	Electronics	10		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 23: Product added and showing successfully

11.2 Module Testing

Username:	MD. ABU SAEM	Role:	Tester	
SI No	Test name	Expected Result	Success Factor	Comments
1.	Product Purchasing	Product add then purchase	Purchase should be successful	Successful

Category Add:

First of all I have to add category to add product under each category.

Category Name

Category Code

SI	Category Name	Category Code	Action
1	Electronics	E101	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

;

Figure 24: Category added successfully

Product Setup:

In order to purchase product, add product under each category

Code

Name

ReorderLevel

Category 

Description

Image No file chosen

Search..

SI	Product Name	Product Code	Category	Reorder Level	Photo	Action
1	Light	L101	Electronics	10		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 25: Product added successfully

Add Supplier:

In order to purchase add supplier:

Name

Code

Address

Email

contact

Contact Person

Image No file chosen

Search..

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	ABU SAEM	As1001	west razabazar	saem@gmail.com	01774227331	kamal		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 26: Supplier added successfully

Purchase product from supplier:

Supplier

SupplierID	ABU SAEM
Date	06/20/2020
invoiceNo	P101

Product

[Check Remarks](#)

ProductID	Light
Code	
manufactureDate	06/01/2020
expireDate	06/30/2020
remarks	100 v Light
quantity	5
unitPrice	200
Total Price	1000
Previous Cost Price	Previous Cost Price
Previous MRP	Previous MRP
newMRP	250

[Add](#)

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
1	LightNaN		2020-06-01	2020-06-30	100 v Light	5	200	1000	250 Delete

[Save](#)

Figure 27: Purchase product from supplier

11.3 Integration Testing

Username:	MD. ABU SAEM		Role:	Tester
SI No	Test name	Expected Result	Success Factor	Comments
1.	Sale product	Product purchase then sale	Selling product should be successful	Successful

Purchase product: Purchase product at first in order to sale

Supplier

SupplierID:

Date:

invoiceNo:

Product

Check Remarks

ProductID:

Code:

manufactureDate:

expireDate:

remarks:

quantity:

unitPrice:

Total Price:

Previous Cost Price:

Previous MRP:

newMRP:

SI	Product(Code)	M. Date	Ex. Date	Quantity	Unit Price	Total Price	New Price	Remarks	Action
1	LightNaN		2020-06-01	2020-06-30	100 v Light	5	200	1000	250 <input type="button" value="Delete"/>

Figure 28: Purchase product from supplier

Sale product:

Customer: --Select--

Date: 06/21/2020

Loyalty point: []

Product

ProductID: Light

Available Quantity: 3

Quantity: 1

MRP: 300

Total MRP: 300

Add

Product Details

SI	Product	Quantity	MRP	Total MRP	Action
0	Light	1	300	300	Delete

Payable: NaN

Discount(%): Discount(%)

Discount: NaN

Grand-Total: 300

Save

Figure 29: Sold product successfully

User Guide:

First of all admin need to complete his registration to the system then he has to login in the system in order to conduct the system. Then sequentially he has to setup category, then product under each category, then supplier setup, then customer setup, then purchase setup, then sales, then check stock. Here below I'm showing steps through screenshots:

Customer Setup:

In customer setup module here admin can setup category to setup products under each category. He can edit and delete category. There is also an option for admin to search category what he wants to see or wants to manipulate.

Category Name

Category Code

[Add](#)

Sl	Category Name	Category Code	Action
1	Electronics	E1011	Edit Delete
2	Food	F101	Edit Delete

Category setup:

In category setup module he can setup product under each category. He can edit and delete each product. He also can set product reorder level that's why system can inform him if product is going below reorder level. There are also an option from where admin can easily access if needed during add product under category. He can also search specific category from list of products.

Code

Name

ReorderLevel

Category 

Description

Image

SI	Product Name	Product Code	Category	Reoreder Level	Photo	Action
1	Light	L101	Electronics	20		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Supplier setup:

He can setup supplier with their details so that he can track from where they purchase product. Here also he can edit and delete supplier. He can also search specific supplier

Name

Code

Address

Email

contact

Contact Person

Image

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	Bablu vaii	b1234	sukrabadd	bvai@gmail.com	019xxxxxxx9	Sarowar Molla		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Customer setup:

He can setup customer with their details so that he can track to whom they sale their product. Here also he can edit and delete customer. He can also search specific customer

Name

Code

Address

Email

contact

L.Point

Image No file chosen

[Add](#)

SI	Name	Code	Address	Email	Contact	Loyalty	Photo	Action
1	saem	E101	East razabazar	abusaem1774@gmail.com	01774227331	12		Edit Delete

Purchase:

Admin can purchase product under supplier multiple times. Here product code and total cost of product will auto appear, if admin set unit price and quantity. Admin can delete purchase before save it to database because there is a partial table where purchase store temporarily before sending it to database.

Add Some Data

Supplier

SupplierID	--Select--
Date	mm/dd/yyyy
invoiceNo	Bill/Invoice No

Product

Check Remarks

ProductID	--Select--
Code	
manufactureDate	mm/dd/yyyy
expireDate	mm/dd/yyyy
expireDate	mm/dd/yyyy
remarks	Remarks
quantity	Quantity
unitPrice	Unit Price
Total Price	Total Price
Previous Cost Price	Previous Cost Price
Previous MRP	Previous MRP
newMRP	New MRP
Add	

SI Product(Code) M. Date Ex. Date Quantity Unit Price Total Price New Price Remarks Action

Save

Sale:

Admin can sell product to customer and he can sell product multiple times. Here customer will rewarded on specific product according their loyalty point. Admin can delete sell before save it to database because there is a partial table where sell store temporarily before sending it to database.

Customer

Customer:

Date:

Loyalty point:

Product Details

SI	Product	Quantity	MRP	Total MRP	Action
	Payable		<input type="text" value="Payable"/>		
	Discount(%)		<input type="text" value="Discount(%)"/>		
	Discount		<input type="text" value="Discount"/>		
	Grand-Total		<input type="text" value="0"/>		<input type="button" value="Save"/>

Product

ProductID:

Available Quantity:

Quantity:

MRP:

Total MRP:

Stock:

Admin can check stock for expire product, sold product, damage product etc. He can also check which product is going under reorder level. System will notify to admin to restore product again. He can check how many product are in, how many products are out, product opening balance, product closing balance etc.

Stock

Product	<input type="text"/>	Start Date	<input type="text" value="mm/dd/yyyy"/>
Category	<input type="text"/>	End Date	<input type="text" value="mm/dd/yyyy"/>
<input type="checkbox"/> Reorder Level <input type="checkbox"/> Expire			

SI	Code	Name	Category	Reorder Level	Ex. Date	Ex. Qty	Opening Blance	IN	Out	Closing Blance
1	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	20	440	12	22	500
2	L101	Light	Electronics	20	12/14/2019 12:00:00 AM	50	600	12	12	500
3	M101	Mobile	Electronics	12	12/14/2019 12:00:00 AM	10	100	12	10	500
4	Mob11	iPhone	Electronics	5	12/23/2019 12:00:00 AM	1	12	45	12	40

Report:

Admin can see report periodically on sale and purchase. He can also check business loss and profit. He can make decision based on statistics.

Date	<input type="text" value="mm/dd/yyyy"/>
Date	<input type="text" value="mm/dd/yyyy"/>
<input type="button" value="Search"/>	

SI	Code	Name	CP(TK)	MRP	Profit
1	L101	Electronics	10	22	440
2	L101	Electronics	10	12	600
3	M101	Electronics	10	10	100
4	Mob11	Electronics	20	12	12

System code

Category setup coding samples:

```
15     StockManageManger _stockManageManger = new StockManageManger();
16     Category _category = new Category();
17     CategoryVM _categoryvm = new CategoryVM();
18
19     public ActionResult Add()
20     {
21         _categoryvm.categories = _stockManageManger.Show();
22
23         return View(_categoryvm);
24     }
25
26     [HttpPost]
27     public ActionResult Add(Category category)
28     {
29         if (ModelState.IsValid)
30
31         {
32             bool isvalid = true;
33             if (category.Name != null && category.Code != null)
34             {
35                 foreach (var item in _stockManageManger.Show())
36                 {
37                     if (item.Name==category.Name)
38                     {
39                         isvalid = false;
40                     }
41                 }
42             }
43             if(isvalid!=false)
44             {
45                 _stockManageManger.Add(category);
46                 ViewBag.message = "Category has been added ";
47                 ViewBag.alert = "alert alert info";
48             }
49         }
50     }
51 }
```

Product setup coding samples:

```
15 // GET: Product
16 StockManageProductManager _stockManageManger = new StockManageProductManager();
17
18 StockManageManager _stockManageManagerCategory = new StockManageManager();
19 Product _product = new Product();
20 // ProductVM _productvm = new ProductVM();
21
22
23 0 references
24 public ActionResult Add()
25 {
26     ProductVM _productvm = new ProductVM();
27     _productvm.CategoryList = _stockManageManagerCategory.Show()
28         .Select(c => new SelectListItem()
29             {
30                 Value = c.ID.ToString(),
31                 Text = c.Name
32             }).ToList();
33
34
35     _productvm.products = _stockManageManger.Show();
36
37     return View(_productvm);
38 }
39 [Route("Add")]
40 [HttpPost]
41 0 references
42 public ActionResult Add(ProductVM model)
43 {
44     Product product = Mapper.Map<Product>(model);
45     HttpPostedFileBase image = Request.Files["ImageData"];
46     var name = _stockManageManger.Show();
47     bool isValid = true;
```

Customer setup coding samples:

```
41         //     return view();
42         //}
43
44         [Route("Add")]
45         [HttpPost]
46         0 references
47         public ActionResult Add(CustomerVM model)
48         {
49             if (ModelState.IsValid)
50             {
51                 Customer customer = Mapper.Map<Customer>(model);
52
53                 HttpPostedFileBase image = Request.Files["ImageData"];
54
55                 _stockManager.Add(customer, image);
56
57             }
58             _customermv.customers = _stockManager.Show();
59             return View(_customermv);
60         }
61
62
63         [HttpPost]
64         0 references
65         public ActionResult Show(CustomerVM model)
66         {
67             Customer customer = Mapper.Map<Customer>(model);
68             if (customer.Name != null)
69             {
70                 _customermv.customers = _stockManager.search(customer);
71             }
72         }
```

Supplier setup coding samples:

```
{
    StockManageSupplierManger _stockManageManger = new StockManageSupplierManager();
    StockManageDbContext Db = new StockManageDbContext();
    Supplier _supplier = new Supplier();
    SupplierVM _supplierVM = new SupplierVM();

    0 references
    public ActionResult Add()
    {
        SupplierVM model = new SupplierVM();
        model.suppliers = _stockManageManger.Show();

        return View(model);
    }

    [Route("Add")]
    [HttpPost]
    0 references
    public ActionResult Add(SupplierVM model)
    {
        if (ModelState.IsValid)
        {
            Supplier supplier = Mapper.Map<Supplier>(model);

            HttpPostedFileBase image = Request.Files["ImageData"];

            _stockManageManger.Add(supplier, image);
        }
        _supplierVM.suppliers = _stockManageManger.Show();
        return View(_supplierVM);
    }
}
```

Product purchase coding samples:

```
0 references
public class PurchaseController : Controller
{
    bool add = false;
    PurchaseManager _stockManageManger = new PurchaseManager();
0 references
    public ActionResult Add()
    {
        PurchaseVM model = new PurchaseVM();

        model.ProductList = _stockManageManger.ShowProduct()
            .Select(c => new SelectListItem
            {
                Value = c.ID.ToString(),
                Text = c.Name
            })
            .ToList();

        model.SupplierList = _stockManageManger.ShowSupplier()
            .Select(c => new SelectListItem
            {
                Value = c.ID.ToString(),
                Text = c.Name
            })
            .ToList();

        if(add==true)
        {
            ViewBag.mess = "Add Purchase Success";
        }
        else
        {
            ViewBag.mess = "Add Gene Data";
        }
    }
}
```

Product sale coding samples:

0 references

```
public class SaleController : Controller
{
```

```
    StockManageSaleManager _stockManageManger = new StockManageSaleManager();
    PurchaseManager _perchaseManager = new PurchaseManager();
    Sale _supplier = new Sale();
    Customer _Customer = new Customer();
    // GET: Sales
```

0 references

```
public ActionResult Index()
{
    return View();
}
```

0 references

```
public ActionResult Add()
{
```

```
    var model = new SalesVm();
    model.CustomerList = _stockManageManger.Show()
        .Select(c => new SelectListItem
        {
            Value = c.ID.ToString(),
            Text = c.Name
        })
        .ToList();
```

```
    var List = _stockManageManger.ShowProduct()
        .Select(c => new SelectListItem
        {
            Value = c.ID.ToString(),
            Text = c.product.Name
        })
        .ToList();
```

Stock check coding samples:

```
//model.Purchases = _manager.purchases();

if (model.Reordercheck == null && model.Expirecheck == null)
{
    model.Sales = (Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.Date >= model.sDate && c.Purchase.Date <=
        model.eDate).ToList();
}
else if (model.Reordercheck == "true" && model.Expirecheck == null)
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.quantity <= c.Purchase.product.ReorderLevel).ToList();
}

else if (model.Reordercheck == "true" && model.Expirecheck == "true")
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.quantity <= c.Purchase.product.ReorderLevel && c.Purchase.expireDate <= model.eDate).ToList();
}

else if (model.Reordercheck == null && model.Expirecheck == "true")
{
    model.Sales = ((Db.sales.Where(c => c.Purchase.product.Name == model.productName).ToList()).Where(c => c.Purchase.expireDate >= model.sDate &&
        c.Purchase.expireDate <= model.eDate).ToList()).Where(c => c.Purchase.expireDate <= model.eDate).ToList();
}

//model.stockoutRIana=model.Purchases.Where(c=>c \
```

Report generate coding samples:

```
using StockManage.BLL.BLL;
using StockManage.DatabaseContext.DatabaseContext;
using StockManageWeb.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace StockManageWeb.Controllers
{
    0 references
    public class ReportController : Controller
    {
        StockManageDbContext Db = new StockManageDbContext();
        StockManager _manager = new StockManager();
        0 references
        public ActionResult Index()
        {
            Report model = new Report();
            model.ProductList = _manager.ShowProduct().Select(c => new SelectListItem { Value = c.ID.ToString(), Text = c.Name }).ToList();

            model.Sales = _manager.sales();
            return View(model);
        }
        0 references
        [HttpPost]
        0 references
        public ActionResult Index(Report model)
        {

            model.Sales = Db.sales.Where(c => c.Purchase.Date >= model.sDate && c.Purchase.Date <= model.eDate).ToList();

            return View(model);
        }
    }
}
```

Plagiarism Report:

Turnitin Originality Report

Processed on: 19-Aug-2020 14:54 +06
ID: 1371335337
Word Count: 9804
Submitted: 1

181-16-274_StcokManagementSystem_Documentation.pdf By
Md. Abu Saem

	Similarity by Source	
Similarity Index	Internet Sources:	1%
16%	Publications:	0%
	Student Papers:	16%

3% match (student papers from 19-Aug-2020)
[Submitted to NCC Education on 2020-08-19](#)

3% match (student papers from 19-Aug-2020)
[Submitted to NCC Education on 2020-08-19](#)

2% match (student papers from 18-Dec-2017)
[Submitted to University of Greenwich on 2017-12-18](#)

1% match (student papers from 15-Sep-2019)
Class: Final_Project_Summer 2019
Assignment: CIS Project Final 2019
Paper ID: [1172815400](#)

1% match (student papers from 29-Dec-2019)
Class: Final_project_Fall2019
Assignment: Project_Fall_2019
Paper ID: [1238621981](#)

< 1% match (Internet from 28-Jul-2020)
<https://www.geeksforgeeks.org/acceptance-testing-software-testing/>