

Shop Management System

Team Name: Xenon

Course Title: Software Project Lab I

Course Code: SE 2112

Team Mentor

Dipanita Saha

Assistant Professor

Institute of Information Technology

Noakhali Science and Technology University

Team Members

1. Prosanto Deb (ASH1925005M)

2. Sultana Marjan (BKH1925010F)

3. Md Alamgir Hossain (ASH1925016M)

Date of Submission: 09-10-2021

Table of Contents

1	Introduction	2
2	Software Project Description	2
2.1	Story	2
2.2	Requirements.....	2
2.2.1	Functional Requirements	2
2.2.2	Non-functional	3
2.3	Proposed Process Model	3
2.3.1	Traditional Model	3
2.3.2	Agile Model	4
2.4	Project Team	4
2.5	Proposed Timeline and Actual Timeline.....	4
2.6	Requirements Traceability Matrix	6
2.7	Tools.....	7
2.7.1	Language: JAVA	7
2.7.2	IDE: NetBeans	7
2.8	User Interface	7
2.9	Future Directions.....	11
3	Software Project Matrix.....	11
3.1	Code Level	11
3.1.1	Average LOC in a Class	12
3.2	Design Level	13
3.3	Collaboration.....	13
3.3.1	LOC addition and deletion.....	13
4	Software Project Deliverables	15
5	Summery.....	15
6	References	15

1 Introduction

Our software project named **Shop Management System** stores records of suppliers, sale and purchase and customer's records are maintained and manipulated. These works were done and managed manually hence leading to the chances of human errors that may create some problems. Thus, a secured and reliable system is required to handle it.

Shop Management System, as described above, can lead to error free, reliable and fast management system. It can assist the staff to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

2 Software Project Description

2.1 Story

Today in Digital Bangladesh everyone is going towards smart and efficient ideas. Like every sectors our most important sector is business sector. Sometime a businessman needs some information about his current state which requires many effort, calculation and time. So our target is to provide a businessman or a shopkeeper such a software which will make him more smart to take perfect decision.

2.2 Requirements

The requirements for a system are the description of what the system should do. We need to work very closely with stakeholders and to draw on their communication and technical skills to surface underlying business needs that might be addressed by a system solution. It can be divided into functional requirements and nonfunctional requirements.

2.2.1 Functional Requirements

- Create Sales Invoice
- Create Purchase Invoice
- Stock Check
- Due Amount Check and Process after Another Transaction
- Add Cost
- Show Daily Income-Cost
- Staff Attendance
- New Member Add and Remove Existing Member
- Use File Input Output

2.2.2 Non-functional

- Store Detailed information including Name, Address and Mobile Number
- Smart searching with any keyword or information stored in file
- Auto Capitalization at Name and Address input
- Showing Digit hint at Mobile Number input
- Showing Available Stock at Sales Quantity input
- Auto Amount Calculation while Creating Invoice
- Monthly Income Cost Information
- Daily and Monthly Stock Information
- Easy Login, Logout and Password Change
- Printing Income Cost, Stock, Attendance Information and Invoice
- Attendance, Password Change, New Model Add only performed by Owner
- Smart Input Validation and Showing message for Invalid Input

2.3 Proposed Process Model

A software process model is an abstraction of the software development process. The models specify the stages and order of a process. In our development we implemented,

1. Traditional Model
2. Agile Model

2.3.1 Traditional Model

At the beginning of our project development we started with Traditional Model. It is the most commonly used approach by organizations whereby software development activities are completed sequentially.

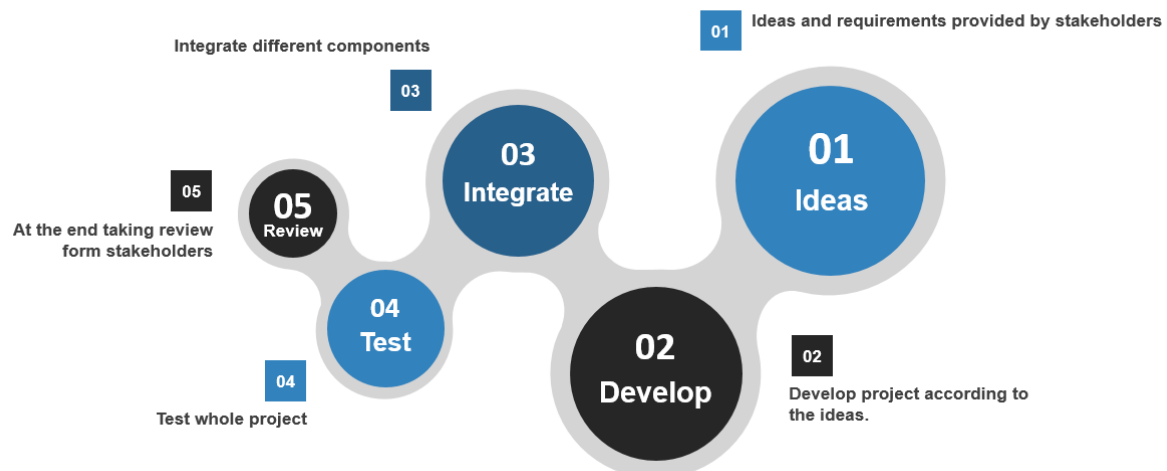


Figure 1: Traditional Process Model

2.3.2 Agile Model

After following Traditional Model, we focused on Agile Model. Agile Model break tasks into smaller iterations. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

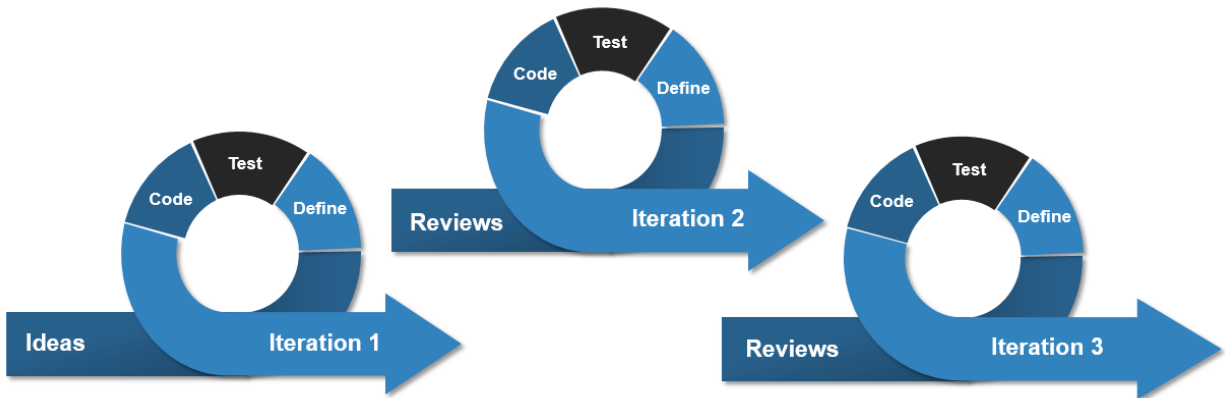


Figure 2: Agile Process Model

2.4 Project Team

Team Name: Xenon

Members:

1. Prosanto Deb (ASH1925005M)
2. Sultana Marjan (BKH1925010F)
3. Md Alamgir Hossain (ASH1925016M)

2.5 Proposed Timeline and Actual Timeline

We proposed an initial timeline to complete our project but due to Covid-19 our timeline was disturbed and we took more than the proposed timeline which is shown at the actual timeline section.

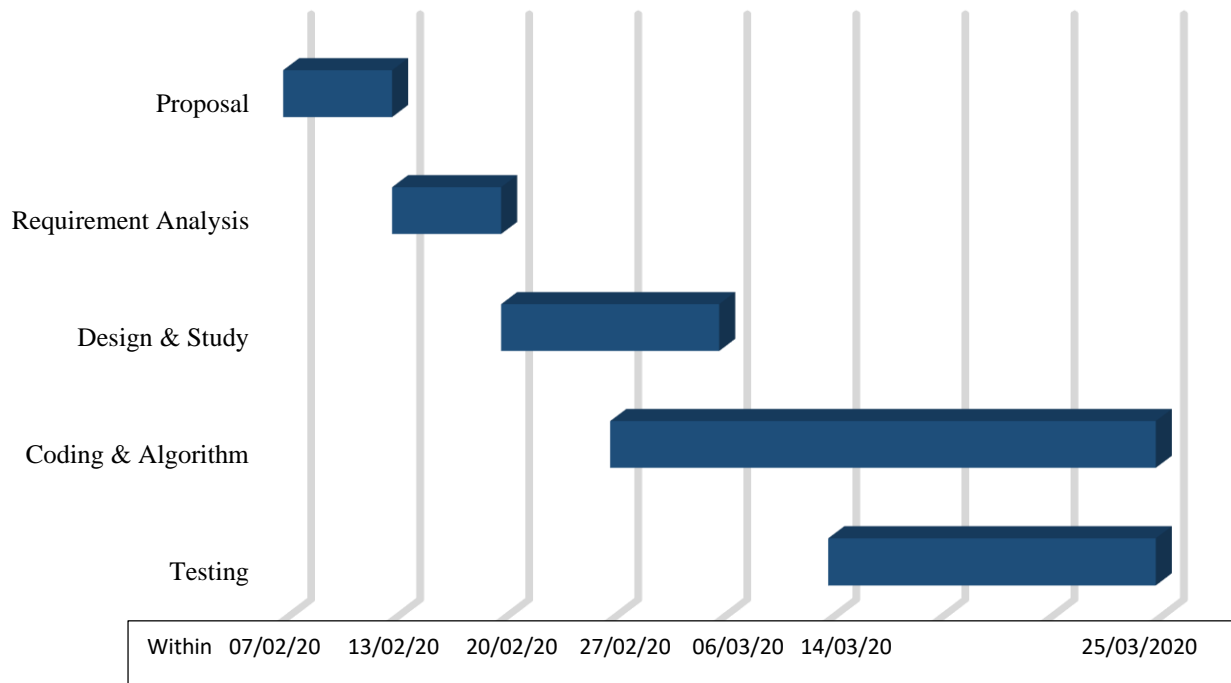


Figure 3: Proposed Timeline

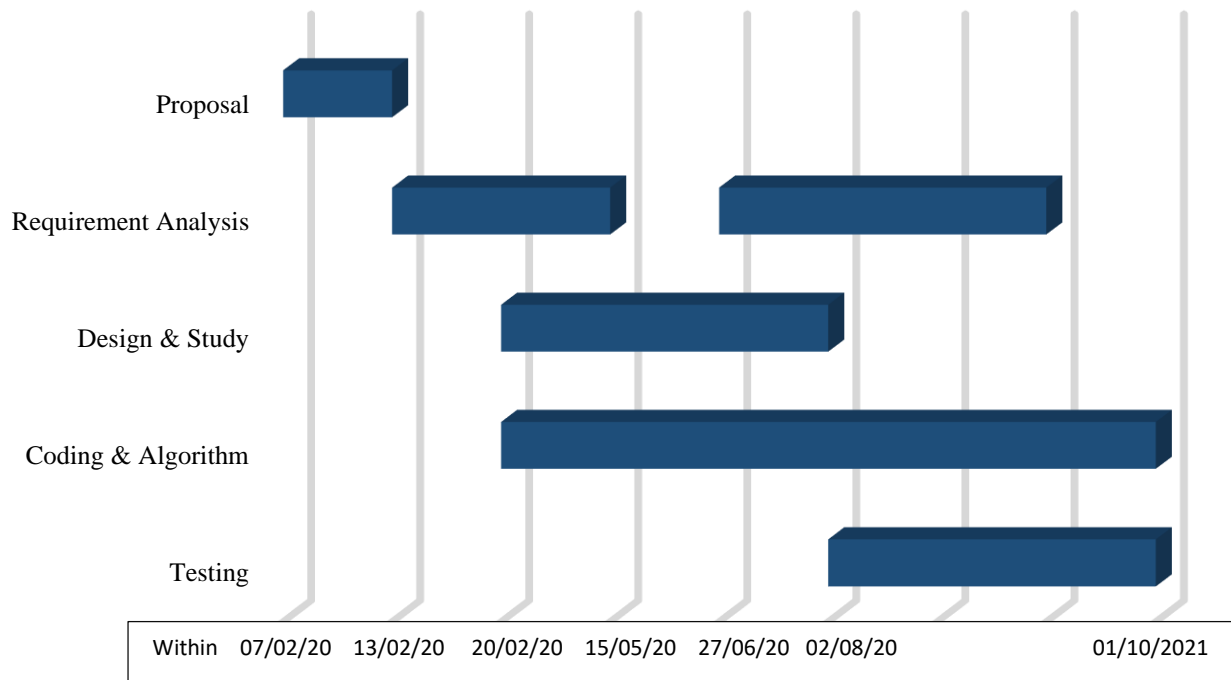


Figure 4: Actual Timeline

Here in actual timeline we can see that a gap in **Requirement Analysis** which shows that our requirements changed and we started our analysis again.

2.6 Requirements Traceability Matrix

Requirements Traceability Matrix is a document, usually in the form of a table, used to assist in determining the completeness of a requirement. It is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled.

Requirement 1 (IDR01)	Create Sales and Purchase Invoice and Print
Requirement 2 (IDR02)	Add Cost
Requirement 3 (IDR03)	Stock Check
Requirement 4 (IDR04)	Advanced Stock Check (Print and show information of Day and Month)
Requirement 5 (IDR05)	Due Check (Print and show information of Day and Month)
Requirement 6 (IDR06)	Daily and Monthly Income Cost including Print Option
Requirement 7 (IDR07)	Staff Attendance
Requirement 8 (IDR08)	Smart Search Feature
Requirement 9 (IDR09)	Settings (Password Change, Member add and Remove, New Model Add)

Table 02: Code Level for Every Classes

Requirement Test Case	Req 1 (IDR01)	Req 2 (IDR02)	Req 3 (IDR03)	Req 4 (IDR04)	Req 5 (IDR04)	Req 7 (IDR05)	Req 8 (IDR08)	Req 9 (IDR09)
Test Case 1	✓		✓				✓	✓
Test Case 2		✓						
Test Case 3			✓					✓
Test Case 4				✓				✓
Test Case 5					✓		✓	
Test Case 6						✓	✓	
Test Case 7						✓		
Test Case 8							✓	✓

2.7 Tools

2.7.1 Language: JAVA

2.7.2 IDE: NetBeans

2.8 User Interface



Figure 5: Loading Page



Figure 6: Login Page

SHOP ASSISTANT

Home

Create Sales Invoice

Create Purchase Invoice

Log Out

Sales Id	S200000002		
Date & Time	09/10/2021 Sat 09:53:46 PM		
Enter Customer's Name			
Enter Customer's Address			
Enter Mobile Number			
Enter Product's Type			
Enter Product's Model			
Enter Product's Quantity		Subtotal	
Enter Per Item Price		Discount	
Product's Payment		Total Payment	
		Total Paid Amount	
		Total Due Amount	

Clear

Add to Cart

Finish

Print

Figure 7: Create Sales Invoice Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Due Check

Staff Attendance

Settings

Date & Time

27/05/2020 Wed 04:01:06 PM

Cost Type

Electricity Bill

Amount

Submit

Figure 8: Add Cost Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Advanced Stock Check

Due Check

Staff Attendance

Settings

Select Date

Day

Month

Year

Submit

Print

Figure 9: Daily Income Cost Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Advanced Stock Check

Due Check

Staff Attendance

Settings

Product Type

Direct Cool Refrigerator

Model Number

Total Direct Cool Refrigerator is 10

Figure 10: Stock Check Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Advanced Stock Check

Due Check

Staff Attendance

Settings

Date & Time

09/10/2021 Sat 09:22:50 PM

Customer Total Due

40000

Company Total Due

20000

Customer Due ID's

Compnay Due ID's

Amount

Amount

Submit

Submit

Figure 11: Due Check Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Advanced Stock Check

Due Check

Staff Attendance

Settings

Date & Time

09/10/2021 Sat 09:23:18 PM

Select Member

Present

Absent

Submit

Show Details

Print Details

Figure 12: Staff Attendance Page

SHOP ASSISTANT

Shop Name

Create Sales Invoice

Create Purchase Invoice

Log Out

Home

Add Cost

Daily Income Cost

Stock Check

Advanced Stock Check

Due Check

Staff Attendance

Settings

Password Change

Enter Old Password

Enter New Password (2 Times)

Submit

Add New Model

Select Category

Enter New Model (2 Times)

Add Model

Add New Staff

Enter New Staff Name

Add Staff

Remove Staff

Select Staff

Remove Staff

Figure 13: Settings Page

2.9 Future Directions

We will develop a web base version using database so that user can use from any place and using any device with any kind of web browsers.

3 Software Project Matrix

3.1 Code Level

Table 02: Code Level for Every Classes

SL	Class	LOC	NCLOC	CLOC	Density of Comments
1	Main Page	126	117	9	7.14 %
2	CreatePurchaseInvoicePage	77	65	12	15.58 %
3	CreateSalesInvoicePage	99	87	12	12.12 %
4	DailyIncomeCostPage	349	325	24	6.87 %
5	DueCheckPage	512	478	34	6.64 %
6	HomePage	299	284	15	5.01 %

7	LoginPage	109	100	9	8.25 %
8	SettingsPage	621	577	44	3.38 %
9	StaffAttendancePage	383	352	31	8.09 %
10	DashBoardTemplate	356	286	70	19.66 %
11	FrameSetup	316	224	92	25.84 %
12	InvoiceGeneratorTemplate	1002	921	81	8.08 %
13	StartingTemplate	77	63	14	18.18 %
14	AddcostPage	210	195	15	7.14%
15	AdvancedstockCheckPage	317	299	18	5.67 %
16	StockcheckPage	174	159	15	8.62 %
	Total Project	5027	4532	495	10.39%

LOC - Line of code or total number of code lines in whole project = 5027

NCLOC - Non-comment line of code in whole project = 4532

CLOC - Comment line of code in whole project = 495

Density of Comments - Number of comment line compared to average code = 10.39%

3.1.1 Average LOC in a Class - The average number of line of code in each class = 314

3.2 Design Level

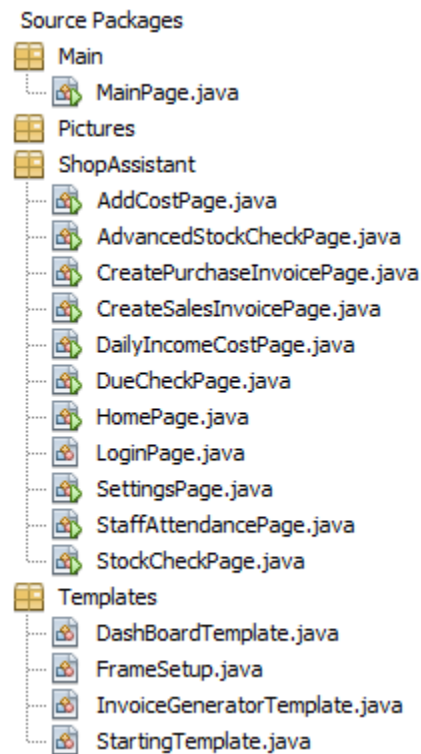


Figure 14: Package and Classes

3.3 Collaboration

3.3.1 LOC addition and deletion

May 10, 2020 – Oct 9, 2021

Contributions: Commits ▼

Contributions to master, excluding merge commits and bot accounts



Figure 15: Total Contribution

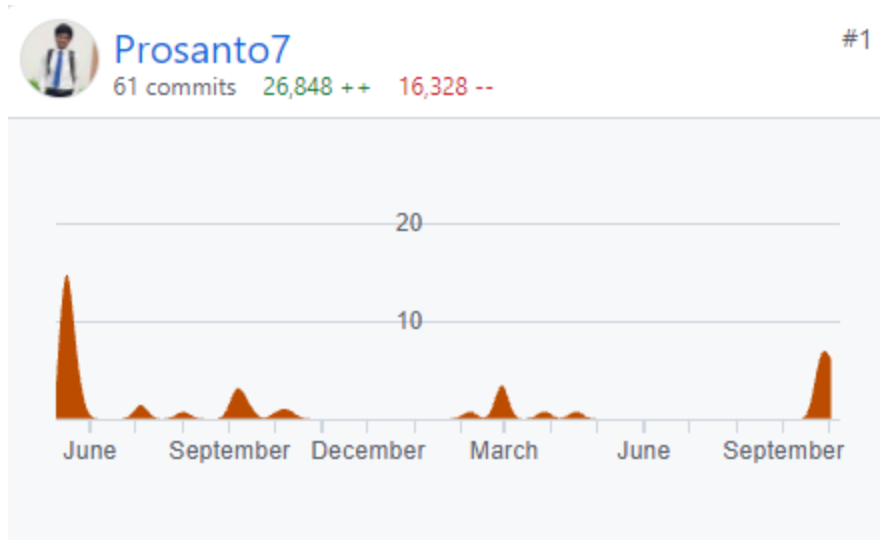


Figure 16: Total Contribution of Prosanto Deb

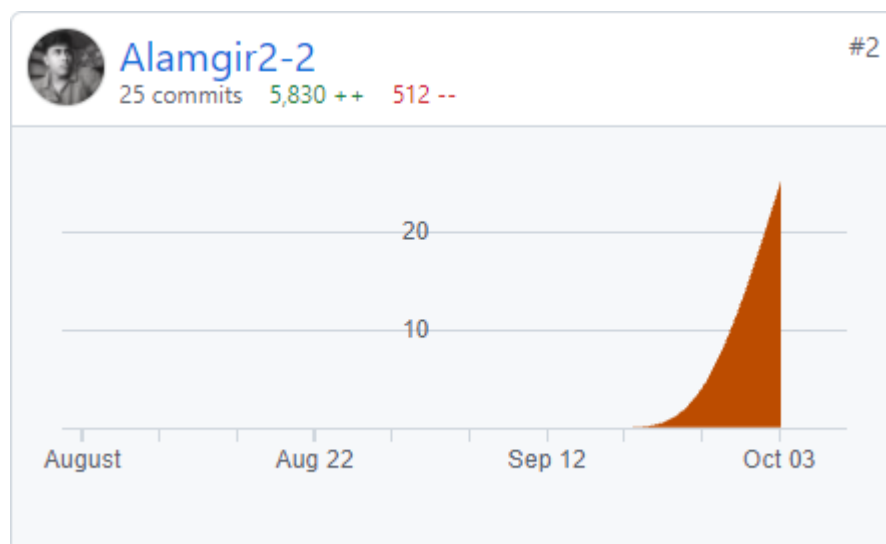


Figure 17: Total Contribution of Alamgir



Figure 18: Total Contribution of Sultana Marjan

4 Software Project Deliverables

1. PowerPoint Presentation
2. Project report
3. Video Presentation of User Manual
4. Source Code

5 Summery

Regarding our implementations its quite clear that our application is fully user friendly and as developer we are also experienced enough situations and gone through many concepts which will cut a good figure in our internship.

6 References

1. <https://github.com/IIT-NSTU> [accessed at: 10.02.2020]
2. <https://stackoverflow.com> [accessed at: 05.02.2020]
3. <https://wikipedia.com> [accessed at: 08.02.2020]
4. <https://quora.com> [accessed at: 10.02.2020]
5. <https://wikipedia.com> [accessed at: 08.02.2020]
6. <https://www.javatpoint.com/java-io> [accessed at: 08.02.2020]