# A Major Project Final Report on

# **S-Mart Stock Management System**

Submitted in Partial Fulfillment of the Requirements for

The **Bachelor of Engineering** in **Software Engineering**

under **Pokhara University**

Submitted by:

**Rabin Panthi 14725**

**Suman K Khanal 14735**

Date: 08-12-2018

 **Department of Software Engineering**

**NEPAL COLLEGE OF**

**INFORMATION TECHNOLOGY**

**Balkumari, Lalitpur, Nepal**

**ACKNOWLEDGEMENT**

It is indeed our great pleasure to express our gratitude to all the people who individually contributed in the making of our project. We would like to thank them for providing us with their valuable suggestions, time and expertise without which we could not have done our project on **“S-Mart Stock Management System”.**

We would like to thank our supervisor **Dr. Roshan Chitrakar** under whose constant guidance we undertook our project. His constant guidance and willingness to share his vast knowledge made us enhance our knowledge and helped us to complete the assigned tasks to perfection. Without his effort and full support & an astonishing testing ability this project may not have succeeded.

It would be our privilege to thank **Er. Birendra Bista**, Head of Department of Software

Engineering for his support. We would also like to thank **Er. Tulsi Dahal** for his valuable suggestion. We are thankful to our college, NCIT that provided us an opportunity to apply our theoretical knowledge through the project. We feel obliged in submitting this project as part of our curriculum.

Although, there may be still many who are unacknowledged in this humble vote of thanks, there are none who remain unappreciated.

Rabin Panthi

Suman Kumar Khanal

Nepal College of Information Technology

Balkumari, Lalitpur

**ABSTRACT**

This project makes use of the codeigniter web framework which is a powerful php framework that is utilized to create this huge sized project. The main issues for this system is to save the products and update the status as the products are sold out. For these issues we solely utilize the core principle of codeignitor i.e MVC pattern. The database creation, manipulation and deletion for the products included in the invoices are coded in the Model pattern. Similarly, each and every event handled within the system falls in the Controller section. Finally, the design or the layout presented is handled in the View section.

**TABLE OF CONTENTS**

1 **Introduction** ………………………………………………………………………… 8

1.1 Problem Statement …………………………………………………………………..8

1.2 Project Overview …………………………………………………………………....8

1.3 Project Objective …………………………………………………………………….9

1.4 Project Scope and Limitations ……………………………………………………….9

1.5 Significance of Study ………………………………………………………………...9

2 . **Literature Review**  ………………………………………………………………….......10

2.1 Review…………………………………………………………………………………....10

2.2 Comparison with the existing system…………………………………………………….11

2.3 Basic Structure of S-Mart Stock Management System…………………………………..12

2.4 Steps involved in the project……………………………………………………………..12

3. **Methodology**………………………………………………………………………………15

3.1 Software development life cycle………………………………………………………....15

3.2 Develop an overall model………………………………………………………………...16

3.3 Build a features list……………………………………………………………………….16

3.4 Pros of FDD………………………………………………………………………………17

3.5 Tools used…………………………………………………………………………………18

3.6 Technologies used…………………………………………………………………………18

4. **Requirement Analysis**...........................................................................................................19

4.1 System requirement Specification ......................................................................................19

4.1.1 Functional requirements ..................................................................................................19

4.1.2 Interface required………………………………………………………………………...19

**4 System Design and UML Models** ......................................................................................20

4.1 Use Case Diagram ...............................................................................................................20

4.2 Operation Contract...........................................................................................................21-22

4.3 Data flow diagram…………………………………………………………………………23

4.4 System sequence diagram…………………………………………………………………24

4.5 Design class diagram………………………………………………………………………25

5  **Testing** ..................................................................................................................................26

5.1 Test 1.1 ...............................................................................................................................26

5.2 Test 1.2 ...............................................................................................................................27

5.3 Test 1.3 ...............................................................................................................................28

5.4 Test 1.4 ...............................................................................................................................29

6. **Project Task And Time Schedule**  ......................................................................................30

7. **Task Completed** ...................................................................................................................31

8. **Conclusion** ..........................................................................................................................32

9.. **Bibliography** ........................................................................................................................33

**LIST OF FIGURES**

1. **Basic Structure of the system……………………………………………………..12**
2. **Graphical illustration of agile methodology……………………………………..15**
3. **Agile feature driven development lifecycle………………………………………16**
4. **Use case diagram…………………………………………………………………..20**
5. **Data flow diagram…………………………………………………………………23**
6. **System sequence diagram…………………………………………………………24**
7. **Design class diagram………………………………………………………………25**
8. **Test case 1.1………………………………………………………………………...26**
9. **Test case 1.2………………………………………………………………………...27**
10. **Test case 1.3………………………………………………………………………...28**
11. **Test case 1.4………………………………………………………………………...29**
12. **Gantt Chart………………………………………………………………………...30**

**LIST OF TABLES**

1. **Comparison with the existing system…………………………………………………11**
2. **Table for tools and purpose……………………………………………………………18**
3. **Table for task completion time schedule……………………………………………...30**
4. **Table for task completion………………………………………………………………31**
5. **INTRODUCTION**

Our project **“S-Mart Stock Management System”** is a web application which provides a storage facility of various categories of products within the central system and their subsequent distributors. Our project is a web-based stock management tool which manage the product stock in manageable way. The products that are stored in the system has well defined set of attributes, categories, brand, descriptions etc. These details are provided once the products are received. In addition, essential services like billing, multi-user accessibility, report etc are included thereby making it a complete and efficient system. This system deals with selling of the stored product stocks to the customers and maintaining the data of sold products so that we can understand the current market trend.

* 1. **PROBLEM STATEMENT**

As we are creating the system, critical problems are about analyzing what attributes are to be added in the system. Similarly, the products have several features like brands, price, quantities etc. They should be filled according to the specifications. Another problem is about giving user accessibility in the fields of creating, updating, viewing and deleting. Every user is not given the same accessibility as the admin is given. In addition, this system is a multi user interface and is delivered to various other retailers. The retailers or the sub-company should have their own address, phone number and vat rates. The main system should have the information of those branches who are distributing the related products. Apart from that, general report of the products that are sold needs to be created so that an idea of what kind of stocks is popular among the customers is gained. The report should be generated on a monthly basis. Once the order is placed and released, the related data like total pricing and address should be maintained.

* 1. **PROJECT OVERVIEW**

One of the key feature of this system is it can maintain a track record of stocks in the company. It is different than an e-commerce in the sense that e-commerce sells the product to the general public online but this system maintains the record of stocks so that the admin can buy the stocks when it is finished and can ship the product to its sub-companies when demanded. In general, it is the admin who is given the authority for total management.

* 1. **PROJECT OBJECTIVE**

In this project, we primarily focused on providing the admin total control of the stocks in the company. The main objectives of the project are:-

1. Management of stocks according to its essential features.

2. Precision of the stock information and location specific shipment.

3. Lessen the errors.

4. Expansion of business through sub-companies.

5. Reduce the possible wrong handling of the account.

* 1. **SCOPES AND LIMITATIONS**

The major scope of the system is that it not only can be used by the central company but can also be distributed to its sub-companies so they can manage their stock properly and place orders once the stock is about to be finished. So the system can take a larger scope in terms of usability.

Limitation of this system is that the admin is the sole responsible person for using it so problems like mismanagement of stocks is highly probable to occur due to wrong judgment. The admin can alter the quantity of stocks released and benefit from it. Constant monitoring is not possible only through this system.

* 1. **SIGNIFICANCE OF STUDY**

The study investigates the need for a complete web based system which is totally handled by a trustworthy admin. There are many online sites to sell the products but the work of ordering and releasing stocks that happens at the back by the authorized personnel is very significant as it held privately by a single person. Major significance can be pointed out as follows:-

1. Customer can choose and identify the stocks according to his favored choices.
2. Admin can note the placed order and command to package and ship the products on the go.
3. Admin can see the data of the sold items, he can therefore purchase the out of stock products efficiently.
4. Admin can handle the request for distribution of the sub-companies and ship the stocks to them and maintain their records.
5. The billing system is integrated in the system which includes the vat rates and shipping prices. The total price of the shipped product can be calculated.
6. The monthly revenue of the company can be generated to comprehend the profit or loss status.

**2. LITERATURE REVIEW**

**2.1 REVIEW**

S-Mart Stock Management System is the basically a web based tool to manage the stock available within the factory, placing orders and releasing them to the demanding customers. We have extensively used codeigniter in order to complete this project. Different applications have different requirements of the stock management technique. For example, some applications require online transactions that enables you to manage orders and inventory. With multi-channel selling, shipping integrations and powerful inventory controls, we can optimize inventory and order management right from purchase to packing to payments. Whereas, our system is mainly targeted for the admin and customer interaction for the purpose of purchase, taking orders and shipping.

**2.2 COMPARISON WITH THE EXISTING SYSTEM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Features** | **Zoho Inventory** | **APICBASE** | **Our System** |
| **1.** | Cross-platform | Yes | no | yes |
| **2.** | Monthly revenue system | No | no | yes |
| **3.** | Multi-user accessibility | No | yes | yes |
| **4.** | Sub-company models | No | no | yes |

Table 1: Comparison with the existing system

**2.3 BASIC STRUCTURE OF S-MART STOCK MANAGEMENT SYSTEM**

S-Mart Stock Management system is the warehouse of the stocks managed in a proper and organized fashion where a customer can place the orders and admin takes them. The shipping is done with as much less time consumption of time as possible.

**ADMIN**

**CUSTOMER**

Places Order Takes Order

Fig 1: Basic Structure of the system

**2.4 STEPS INVOLVED IN THE PROJECT**

**2.4.1 Planning our Layout**

The basic plan was to create a layout that could have good UI/UX experience. This was mostly done with the front-end programming and structuring using Javascript, HTML,Css and bootstrap. We figured out that the left slider menu must contain the necessary features of the project which can be approached just by a simple click. As we select those features, the event was handled and displayed just on the right side, an entire page was not created.

**2.4.2 Adding all the required features**

Adding all the features that might be necessary for the system was analyzed and created on the left slider menu.

**2.4.3 Adding the forms**

Once the features were added, we needed them to function through the forms where a user could input the data and be stored. The input data were stored in the database which could be retrieved in the located fields.

**2.4.4 Adding the attributes**

The necessary attributes were thought and were added to the slider where the admin can add or remove any kind of attribute he desires. These attributes are necessary for the detail information regarding the stocks.

**2.4.5 Outputting the details**

Once the fine details of the stocks are added, they could be displayed dynamically by viewing the Products bar.

**2.4.6 Adding Orders**

The orders could be placed by the customer by entering the name, address and phone number. Once these details were filled up, the total pricing could be displayed dynamically on the right side with the added vat rates.

**2.4.7 Creating the Reports**

Total revenue of the products on the monthly basis were displayed on the Reports section of the bar. This feature lets the admin know whether he is on profit or loss.

**2.4.8 Authorizing other users**

As this system is a multi user interface, we should be able to give access to other users as well. But they should not be given all out access as the admin gets. The process of creating, updating, deleting is partially given to them by the admin. The personal information of those users are stored in the database.

**3. METHODOLOGY**

We have planned to work following these methodologies for the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of our project. This section presents a detailed information about the software development process, project approach and the tool that we used for our project.

**3.1 SOFTWARE DEVELOPMENT LIFE CYCLE: AGILE**

The framework we will be using for developing this project is Agile model. This is an umbrella term for several iterative and incremental software development methodologies. While each of the agile methodologies is unique in its specific approach, they all share a common vision and core values. They all fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system. They all involve continuous planning, continuous testing, continuous integration, and other forms of continuous evolution of both the project and the software. They are all lightweight, especially compared to traditional waterfall-style processes, and inherently adaptable. What is more important about agile methods is that they all focus on empowering people to collaborate and make decisions together quickly and effectively.

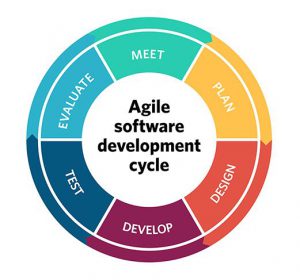


Fig 2: Graphical illustration of agile methodology

As you see in Figure below, there are five main activities in FDD that are performed iteratively.

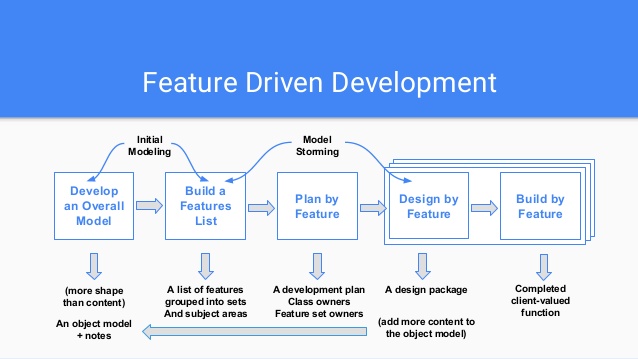


Fig 3: Agile feature driven development life cycle

**3.1.1 DEVELOP AN OVERALL MODEL**

The initial result being a high-level object model and notes. At the start of a project your goal is

to identify and understand the fundamentals of the domain that your system is addressing, and

throughout the project you will flesh this model out to reflect what you're building.

**3.1.2 BUILD A FEATURES LIST**

The second step is grouping them into related sets and subject areas. We collect the list of all the

features found in our project and specialized them accordingly.

The features are listed below:-

1. Brands
2. Categories
3. Attributes
4. Products
5. Order
6. Report
7. Multi-user accessibility

**3.1.3Analysis by Feature**

The end result being a development, the identification of class owners, and the identification of

feature set owners.

**3.1.4 Design by Feature**

The majority of the effort on an FDD project, roughly 75%, is comprised of the fourth and fifth

steps. In this step, we continuously model the feature in detail and build the feature iteratively

until all the features are implemented.

**3.1.5 Build by Features**

In this step, the detailed design gotten from previous step are programmed, tested and packaged

iteratively.

**3.2 PROS OF FDD**

1. Lets to move to large projects and obtain repeatable success.

2. Practicing the five processes helps to bring new staff in with a shorter ramp-up time.

3. Feature-Driven Development is built around a core set of industry-recognized best practices

4. Risk Reduction via iteration of design & build in small chunks. FDD helps in reducing risks

using shorter iterations of designing, understanding of the requirements and the system- in a

clear and distinct way, thereby leading to a state where there are no ambiguities, as the needs and

expectations are already understood very well.

5. Clarity of requirements and better understanding of system to be built is gained through the

develop overall model process. This process includes high-level walkthrough of the scope of the

system and its context. Next, detailed domain walkthroughs are held for each modeling area.

1. Costing the project by feature leads to greater accuracy.

**3.3 WHY DID WE CHOSE FDD**

The first process, developing the overall model makes us have a deep understanding of the scope

and the context of the project.

The fact that we have a deeper understanding of the requirements and the expectations, that we

do small iterations and build small parts, one by one, implies that the risk is really reduced. Less

unwanted surprises!

**3.4 TOOLS USED**

|  |  |
| --- | --- |
| **TOOLS** | **PURPOSE** |
| Google Chrome | Debugging and continuous testing |
| Xampp | Testing suite for connecting web to localhost |
| Github | Manage source code |

Table 2: Table for tools and purpose

**3.5 TECHNOLOGIES USED**

1. Javascript for event handling and client side programming
2. Php for server side programming
3. MySql for database
4. Css and Bootstrap for design
5. **REQUIREMENT ANALYSIS**

Requirement analysis, in software engineering encompasses those tasks that go into determining

the need and conditions to meet for a new or altered product, taking account of possibly

conflicting requirements of the various stakeholders, such as users. It is the early stage activity of

requirement engineering which encompasses all activities concerned with eliciting, analyzing,

documenting, validating and managing system requirements.

**4.1SYSTEM REQUIREMENT SPECIFICATIONS**

**4.1.1FUNCTIONAL REQUIREMENTS**

Provide the admin total control over listing the product features ranging from names, prices to brands.

Provide the admin with the user interface to take the order of product of their choice.

**4.1.2DATA ENTRY MODEL**

The following input formats were applied:-

1. The order is taken from the click of a mouse and entry through the keyboard.
2. The admin approves the order and goes for the billing system and updates the database.

**4.1.3 INTERFACE IS REQUIRED**

The admin is provided with the web browser in order to use the web tool.

1. **SYSTEM DESIGN AND UML MODELS**

**5.1 USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system that

shows the relationship between the user and the different use cases in which the user is involved.

admin customer

driver

Fig 4: Use Case Diagram

**5.2 OPERATION CONTRACT**

**Use Case UC1:Take Orders**

Primary Actor: Order taker

Cross References: S-Mart Stock Management System

Secondary Actor: none

Stakeholders:

Customer: Select the Products

Order taker: Takes the order

Preconditions:

none

Post conditions:

The Orders must be taken by the admin

The Billing system should be generated

**5.3 Operation Contract**

**Use Case UC2: Take Orders**

Primary Actor: Admin

Cross References: S-Mart Management System

Secondary Actor: Order taker

Stakeholders:

Order taker: Place the orders

Preconditions:

Orders should be placed

Post conditions:

The products should be shipped

The database should be updated

Reports should be maintained

**5.4 DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. We used DFD as a preliminary step to create an overview of the system, which can later be elaborated also be used for the visualization of data processing (structured design) .

Fig 5: Data flow diagram

Input Text

Customer receives Order

Customer Selects Order

Output Text+Images

**5.5 SYSTEM SEQUENCE DIAGRAM**

Sequence Diagram is an interaction diagram. It shows how the events occur and in what order. For our system we have designed sequence diagrams for most critical and influential activities which are shown below:

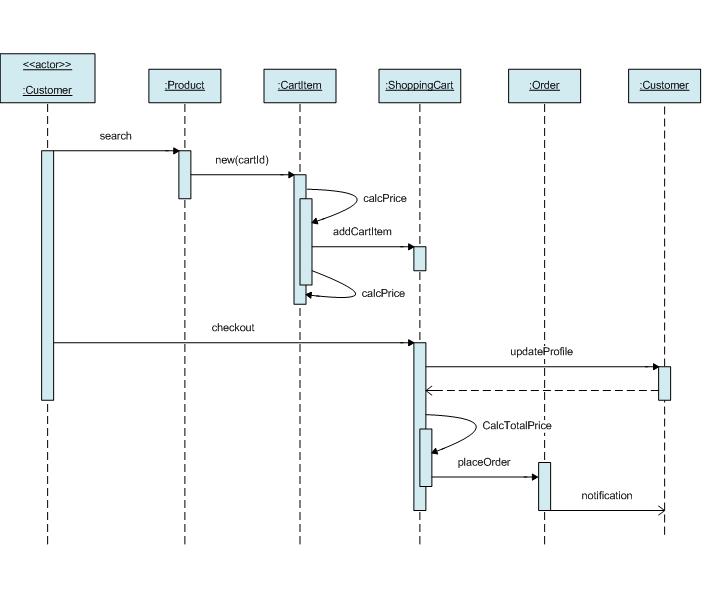
****

Fig 6: System sequence diagram

**5.6 DESIGN CLASS DIAGRAM**

Design class diagrams are the mainstay of object-oriented analysis and design. Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling.

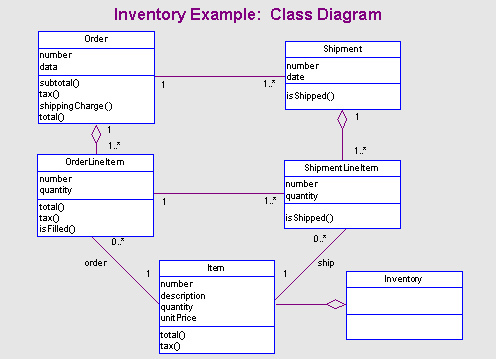


Fig 7: Design Class Diagram

**6. TESTING**

We wanted to make sure that all the component of the developed worked functioned properly.

For this, we created a test cases for our work, in which elements such as validation, reliability

and user acceptance will be tested. The system will be tested for normal condition, primarily.

**6.1 Test 1.1**

Component: Log in

Purpose: accessibility

Expected output : The admin logs in successfully

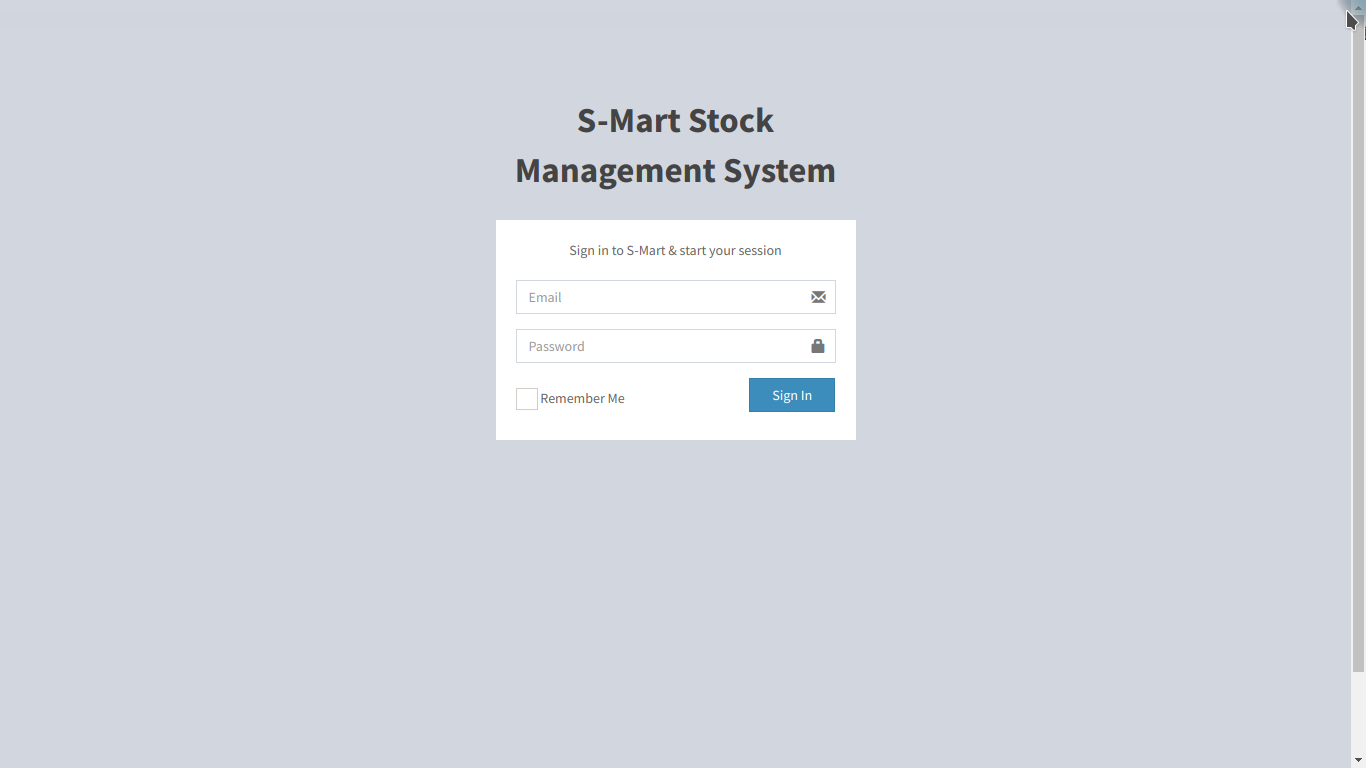
****

Fig 8: Test 1.1

**Test 1.2**

Component: Layout

Purpose: General appearance

Expected output: Display of Dashboard and left slider

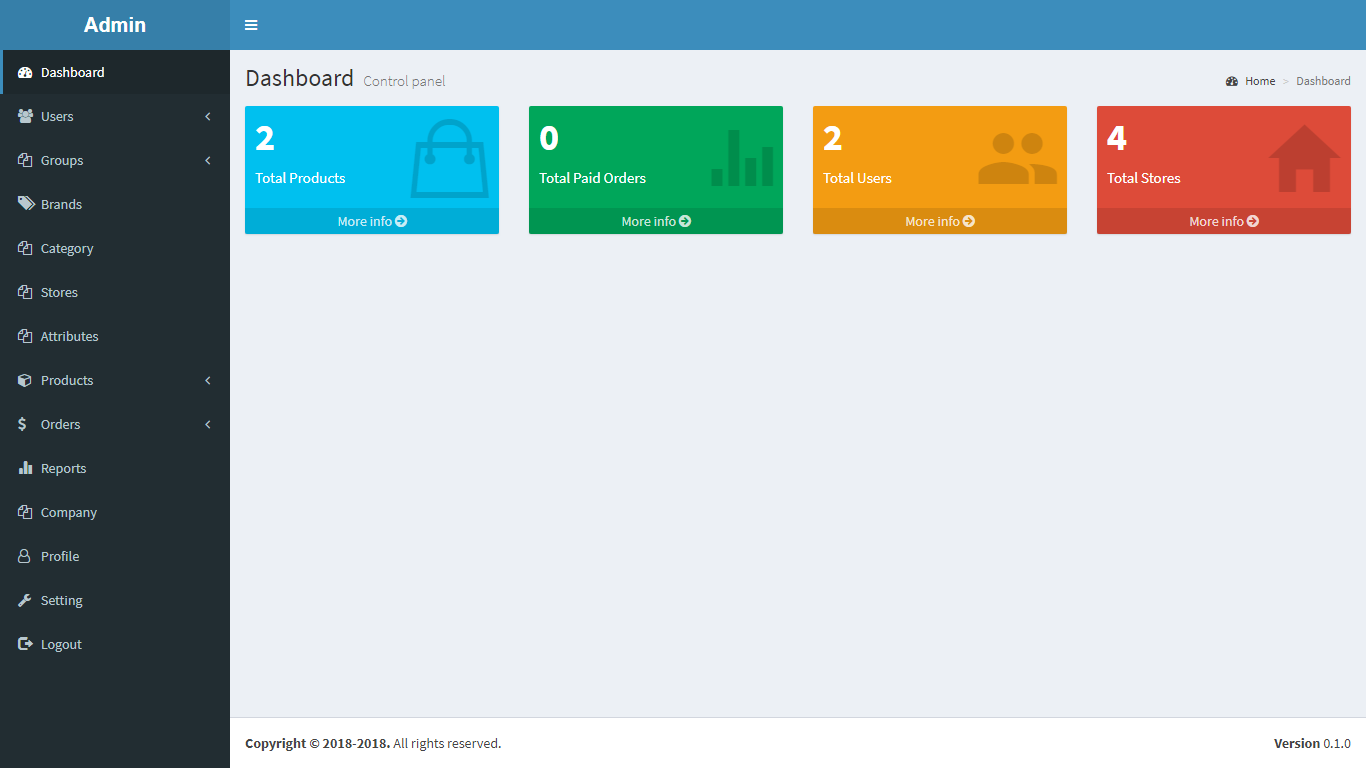
****

Fig 9: Test 1.2

**Test 1.3**

Component : Entry of details

Purpose : Description of products

Expected output: Storage successful

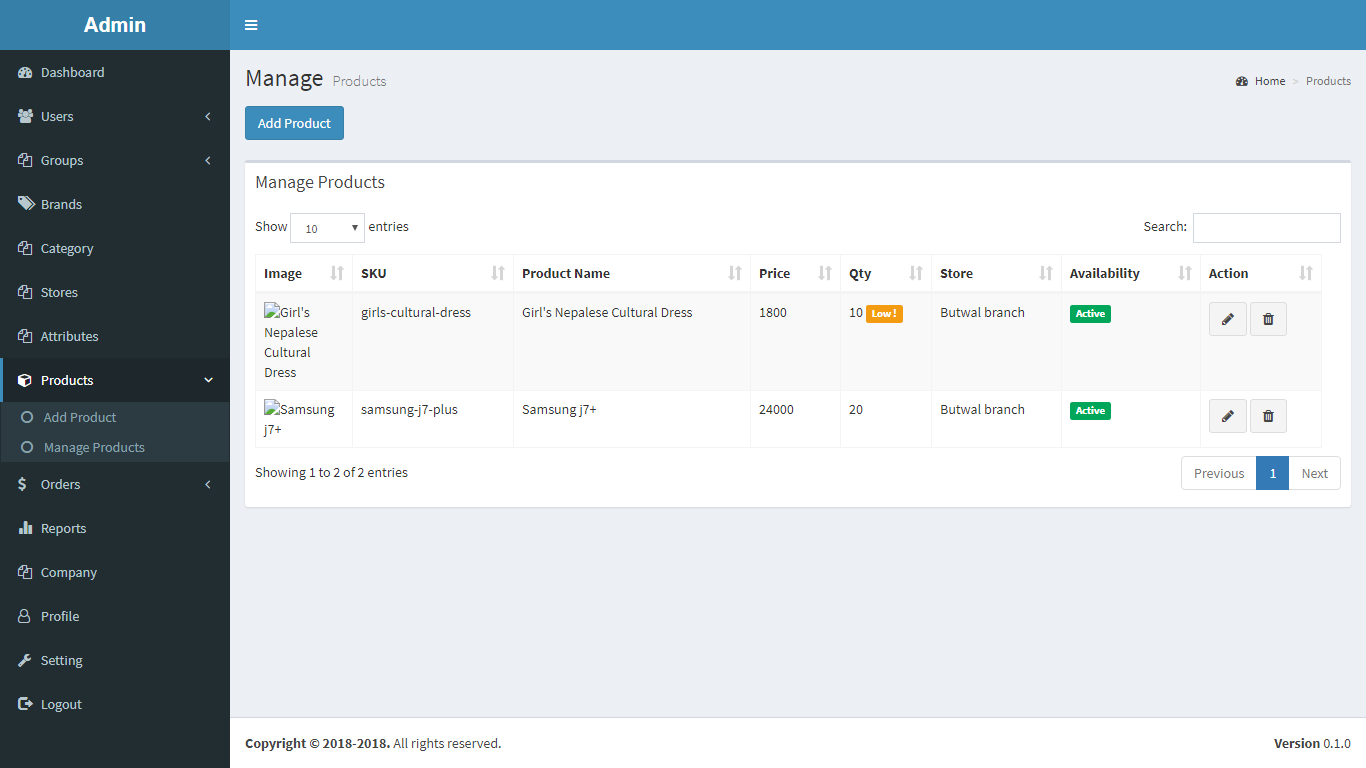
****

Fig 10: 1.3

**Test 1.4**

Component: Billing system

Purpose: To manage economics

Expected Result: successful

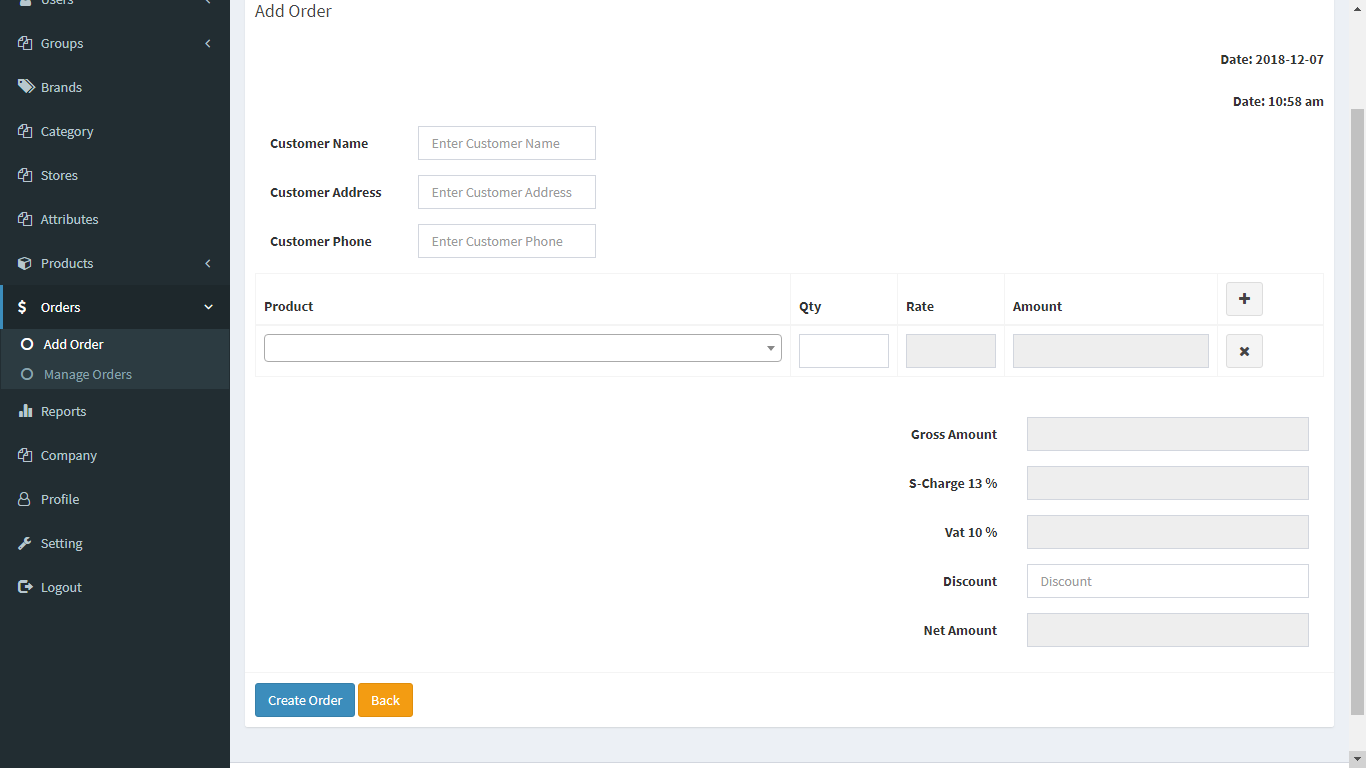


Fig11: test 1.4

**7.PROJECT TASK AND TIME SCHEDULE**

The project schedule has been designed as per requirements and constraints involved. This project is scheduled to be completed in about 4 weeks. Requirement analysis have been

given more emphasis. Research is to be done first and well documented.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tasks | Login to the system | Layout design | Forms | Categories and brands | Attributes and product |
| Requirement Analysis | 1d | 2d | 1d | 1d | 1d |
| Analysis of system | 1d | 1d | 1d | 1d | 1d |
| Design System | 1d | 3d | 2d | 1d | 1d |
| Implementation | 2d | 2d | 2d | 1d | 1d |
| Approx duration | 5d | 7d | 6d | 4d | 4d |

Table 3: Table for task completion time schedule

Overall timeline for building the project is as follows:-

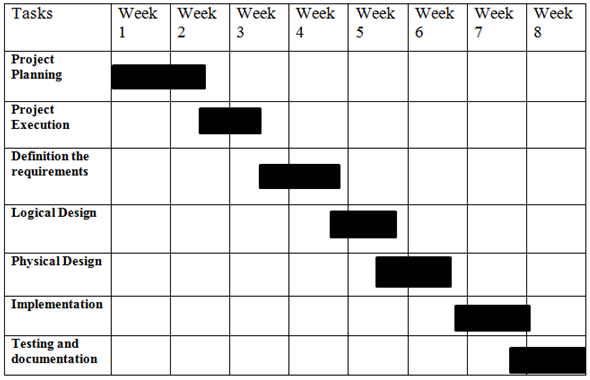


Fig 12: Gantt chart for project to complete

1. **TASK COMPLETED**

The following table depicts the task status of our project and lists the tasks we have

completed, currently working on and still remaining.

|  |  |  |
| --- | --- | --- |
| SN | Tasks | Task Status |
| 1 | Login system | Completed |
| 2 | Graphical User Interface for Dashboard | Completed |
| 3 | Adding of categories | Completed |
| 4 | Adding of brands and products | Completed |
| 5 | Adding of stores | Completed |
| 6 | Adding attributes | Completed |
| 7 | Billing | Completed |
| 8 | Place Order | Completed |
| 9 | Generation of reports | Completed |

Table 3: Table for task completion

**9. CONCLUSION**

It is observed that this dashboards and left slider is completed along with the respective forms. Whenever the products run out, the admin purchases them and enter the details. The data entered into the form about the products are backed in the database. The customer can place order which is validated by the admin. The products are shipped to the customers and the flow is preserved in the system so that we can gain the idea of the products that are in high demand.

**10.BIBLIOGRAPHY**

1. <https://www.capterra.com/inventory-management-software/>

2. <https://ecommercetrainingacademy.com/tools/list-inventory-management-software-ecommerce/>

3. <https://www.softwareadvice.com/inventory-management/>

4. <https://medium.freecodecamp.org/making-an-awesome-inventory-management-application-in-php-and-mysql-from-start-to-finish-90bc5996680a?gi=f5154b282c4e>

5. <https://www.peopleperhour.com/hourlie/make-custom-php-backend-frontend-admin-panel-software/306816>