

A Project Report On

INVENTORY MANAGEMENT SYSTEM

Submitted in partial fulfilment of the
MASTER OF COMPUTER APPLICATION

By

Sana Afrin

Enrollment No. AJU/210020

Under the esteemed guidance of

Dr.Arvind Kumar Pandey

&

Ms. Alka Kumari

(Internal Guide)



DEPARTMENT OF COMPUTER SCIENCE & IT
ARKA JAIN UNIVERSITY, JHARKHAND

Jamshedpur

2021-2023

A photograph of the main entrance gate of Arka Jain University. The gate is a large, white, rectangular structure with a flat top. The words "ARKA JAIN UNIVERSITY" are written in large, dark, capital letters across the top of the gate. Below the gate, there is a wooden fence and a person standing near the entrance. The background shows some green trees and a clear sky.

ARKA JAIN UNIVERSITY

GUIDED BY

Dr. ARVIND KUMAR PANDEY

&

Prof. ALKA KUMARI

SUBMITTED BY

Sana Afrin

SUBMITTED TO

**DEPARTMENT OF COMPUTER
SCIENCE & IT**

A Project Report On

INVENTORY MANAGEMENT SYSTEM

by

Sana Afrin

Enrollment Number: AJU/210020

Under the esteemed guidance of

Dr.Arvind Kumar Pandey

&

Ms. Alka Kumari

(Internal Guide)



DEPARTMENT OF COMPUTER SCIENCE & IT

ARKA JAIN UNIVERSITY, JHARKHAND

JAMSHEDPUR

2021-2023

ARKA JAIN UNIVERSITY, JHARKHAND

DEPARTMENT OF COMPUTER SCIENCE & IT



CERTIFICATE

This is to certify that the project entitled, " **Inventory Management System**", is bonafied work of **Sana Afrin** bearing **Enrollment no- AJU/210020** submitted in partial fulfilment of the requirements for the award of degree of MASTER OF COMPUTER APPLICATION (MCA) from ARKA JAIN University, JHARKHAND.

Aruka Singh

Internal Guide

Anand Parly
HOD



University Seal

Date: 23.7.2022

ABSTRACT

The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorize individual aspects for the sales and inventory management system. Inventory Management System is important to ensure quality control in businesses that handle transactions resolving around consumer goods. Without proper inventory control, a large retail store may run out of stock on an important item. Inventory Management System is also on important means of tracking large shipment. An Inventory Management System helps to minimize the errors while recording the stock.

The software “Inventory Management System” is a web-based software developed by using the markup language “HTML”, implemented on PHP. It would be a distributed system so that multiple systems can access and manipulate the database. The database would be maintained in MySQL.

From this project, shop can handle the details of vendor, customer, stock, sales, items and login details.

ACKNOWLEDGEMENT

It is a genuine pleasure to express my profound gratitude and deep regard to my guide “**Ms. Alka Kumari**” for her exemplary guidance, monitoring and constant encouragement.

I would like to specially thank “**Mr. Arvind Kumar Pandey**“ our Head of the Department who gave me the golden opportunity to do this wonderful project on the topic “Inventory Management System”, which helped me in research and I came to know about so many things.

With Regards

Sana Afrin

AJU/210020

Roll no: 002

TABLE OF CONTENTS

Chapter 1

Introduction

1.1. Overview	1
1.2. Objectives	1
1.3. Benefits of New System	1
1.4. Identification of Needs	2

Chapter 2

Survey of Technologies

2.1. Visual Studio Code	3
2.2. Languages	3
2.2.1. Html	3
2.2.2. CSS	3
2.2.3. JavaScript	3
2.2.4. PHP	4
2.2.5. MySQL	4

Chapter 3

Requirement & Analysis

3.1. SRS(Software Requirement Analysis)	5
3.2. Preliminary Investigation	7
3.3. Feasibility Study.....	8
3.4. System Analysis	9
3.5. Hardware and Software Requirements	10
3.6. DFD	11
3.6.1. Context Level DFD	12
3.6.2. DFD Level 1	13

3.6.3. DFD Level 2	15
3.7. ER-Diagram	19
3.8. Flowchart	20
3.9. Normalization	21
3.10. Gantt Chart.....	23

Chapter 4

System Design

4.1. Input to the project	24
4.2. Output to the project	25
4.3. Modularization details	26
4.4. Data Integrity	27
4.5. Data Dictionary	28

Chapter 5

Program Code and Testing

5.1. Code Details.....	30
5.2. Testing Approach	39
5.2.1. Use Case	41
5.3. Validation Check	41

Chapter 6

Result and Discussion

6.1. User Documentation	42
-------------------------------	----

Chapter 7

Conclusion

7.1. Limitations	45
7.2. Future Scope	45

7.3. Conclusion	45
7.4. References	46

Ch 1. INTRODUCTION

1.1. OVERVIEW

Inventory management information system is high performance software, which speed up the business operation of the organization . Every organization , which deals with the raw materials, put its great effort in the efficient utilization of its raw, material according to its need and requirement . The organization has to perform number of tasks and operations in order to run its business in manual system .

The “Inventory Management System”, is used for checking and listing of the sales of the company, it is faster and more reliable rather than doing manually. The system can minimize human errors in editing and be easily access anytime by the company. Sales and inventory system make the company more effective, more productive and is convenient for the company and its customers.

1.2. OBJECTIVES

This project is aimed at developing a web-based inventory management System Tool, which is of importance to a small shop . The Inventory Management System is a software-based application works as a simple software to maintain record and maintain reports.

1.3. BENEFITS OF EXISTING SYSTEM

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. There is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.

- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency.

1.4. IDENTIFICATION OF NEEDS

The existing system is a manual one. After studying the problems of the existing system, the following requirements have been identified.

- Develop a new system that will reduce the manual effort maintaining stock details.
- Develop a system that will built-up the database to facilitate future information and retrieval for analysis and other statements.
- Develop a system that will automate the monitoring of any problem during analysis.
- Develop a system that has a flexible form design.
- The system should have provision to view performance during working with system.

After completing the requirement determination and doing requirement analysis new system is designed with could solve the problem of existing system and fulfill the requirement of the users.

Ch 2. SURVEY OF TECHNOLOGY

2.1. SOFTWARE DESCRIPTION

Visual studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

Languages

HTML

HTML stands for **Hypertext Markup Language**. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Why Using PHP?

There are a lot of reasons to know and love PHP, probably the most potent and valid of which is this: it's used and runs everywhere the web does. Your cheap little \$3 per month hosting account *may* let you run a web application in Python or Ruby if you shop carefully. But it'll definitely run PHP. This means that you can count on it wherever you are.

And because it runs everywhere, and is easy to get started with, *a lot* of very popular software is written in PHP. **WordPress** is the example that's both largest and most familiar to me, but tools like Joomla, Drupal, Magento, ExpressionEngine, vBulletin (yep, that's still around), MediaWiki, and more are all running PHP on the server.

Why using MYSQL?

Many of the world's largest and fastest-growing organizations including Facebook, Google, Adobe, Alcatel Lucent and Zappos rely on MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software.

Since then, the performance & scalability, reliability, and ease of use of the world's most **popular** open source database, characteristics that made **MySQL** the #1 choice for web applications, have relentlessly been improved.

Ch 3. REQUIREMENT AND ANALYSIS

3.1. SOFTWARE REQUIREMENTS SPECIFICATIONS

A software requirements specification is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

1. **Product perspective:** The software product is a Web application. The application will be made up of two parts, one administrator who has all the rights and the other user who has limited rights to handle the application. The two users of the system, namely the Service Manager (Admin) and Customers (User) interact with the system in different ways.
2. **Product Functions:** First of all, it will authenticate the user whether he is Admin or User the unauthorized person can't get access to the application. The Admin will be able to Add, delete, and modify Product details. He can also Add, delete and modify Service Request made by Customers. He can use this application to check all reports related to Product Sell, and assign work order as well as he can manipulate the data of Repair Request. The User has some less function compare to Admin. He will be able to Submit Service Requests, Update Own Profile etc. He can check request status.
3. **Safety Requirements:** All the data will be saved to database for safety purpose so there will be no data loss. These data can be accessed only by an authorized person so data theft is also not possible in this application.
4. **Security Requirements:** For preventing unauthorized access to the application, this application has login feature so only granted user can access with defined rights.
5. **System Analysis and Design:** It refers to the process of examining a business situation with the intent of improving it through better methods and procedures. System Analysis is the process of gathering and interpreting the facts, diagnosis the problems and using the information to recommend the improvements to the replace or compliment an existing system.
6. **Preliminary investigation:** The system is investigated. The objective of this phase is to conduct an initial analysis and findings of the system.

Requirement analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Feasibility study

To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

System analysis

Gather, analyze, and validate the information. Define the requirements and prototypes for new system.

Hardware and software study

The hardware and software is developed or maintained for the smooth running of the project.

System design

The design phase comes after a good understanding of customer's requirements, this phase defines the elements of a system, the components, the security level, modules, architecture and the different interfaces and type of data that goes through the system.

System testing and implementation

Testing is becoming more and more important to ensure customer's satisfaction, and it requires no knowledge in coding, hardware configuration or design. In this phase, the system is ready to be deployed and installed in customer's premises.

Evaluation

This phase identifies whether the system meets the initial requirements and objectives. This is when the system is evaluated for weaknesses. The objective of the evaluation phase of the systems development life cycle is to deploy the system and train the system end users.

Maintenance and modification

In this phase, periodic maintenance for the system will be carried out to make sure that the system won't become obsolete, this will include replacing the old hardware and continuously evaluating system's performance, it also includes providing latest updates for certain components to make sure meets the right standards and the latest technologies to face current security threats.

3.2. PRELIMINARY INVESTIGATION

The first step in any system development is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is rather concerned with collecting of information that helps to evaluate the merits of the project that is to be undertaken and make an informed judgment about the feasibility of the proposed project.

While working on the INVENTORY MANAGEMENT SYSTEM, I kept in mind that following objectives must be accomplished:

- Understand the project requested.
- Determine the size of the project
- Assess costs and benefits of alternative approaches
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance of the proposal.

While conducting the investigation I have gone through the following activities:

1. **On-Site Observation:** Another technique I used to collect the training information is on-site observation. During this activity, I have observed the activities of the system directly. One main purpose was to get as close to the real system being studied. During on-site observation, I have seen the office environment, working style of staff and teachers, and method of work.
2. **Conducting the interviews:** Written document and on-site observation technique helped me to understand how the Registration System operates, but again they don't include enough details to allow a decision to be made about the merits of the system proposed, nor do they present user views about current operations. To learn these details I have conducted the interviews.

Requirement determination technique involves studying the current system to find out how it works and where improvement should be made. System studies results in evaluation of how current methods are working and where adjustments are necessary.

Requirement is a feature that is must for any new system. The determination of requirement thus entails studying the existing system and collecting details about it to find out what these requirements are.

After studying all the details of the existing system, the exact problem is found out, new requirements of the user are studied and main objectives of the proposed system are outlined. After thorough study of the exact requirement of the system is clear and presentation diagram of the proposed system is designed and presented to the user, several solution strategies with regard to the problem is outlined.

3.3. FEASIBILITY REPORT

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the small shops. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

1. Does the necessary technology exist to do what is suggested?
2. Do the proposed equipments have the technical capacity to hold the data required to use the new system?
3. Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
4. Can the system be upgraded if developed?
5. Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface. Thus, it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

1. Is there sufficient support for the management from the users?
2. Will the system be used and work properly if it is being developed and implemented?
3. Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies. There is nominal expenditure and economical feasibility for certain.

3.4. SYSTEM ANALYSIS

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

Study of The System

GUI'S

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browses interface. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

3.5. HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement :-

- Dual core processor and above
- 2GB of RAM & above
- 20GB of hard disk & above

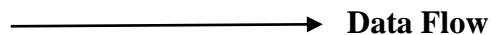
Software Requirement :-

- Operating system: Windows 10
- Front end: Html, CSS, JavaScript
- Frameworks and library: Bootstrap, Font awesome, Google fonts
- Backend: PHP
- Text Editor: Visual Studio Code
- Database: MySQL
- Web Server: Apache
- Drawing tools: Star UML

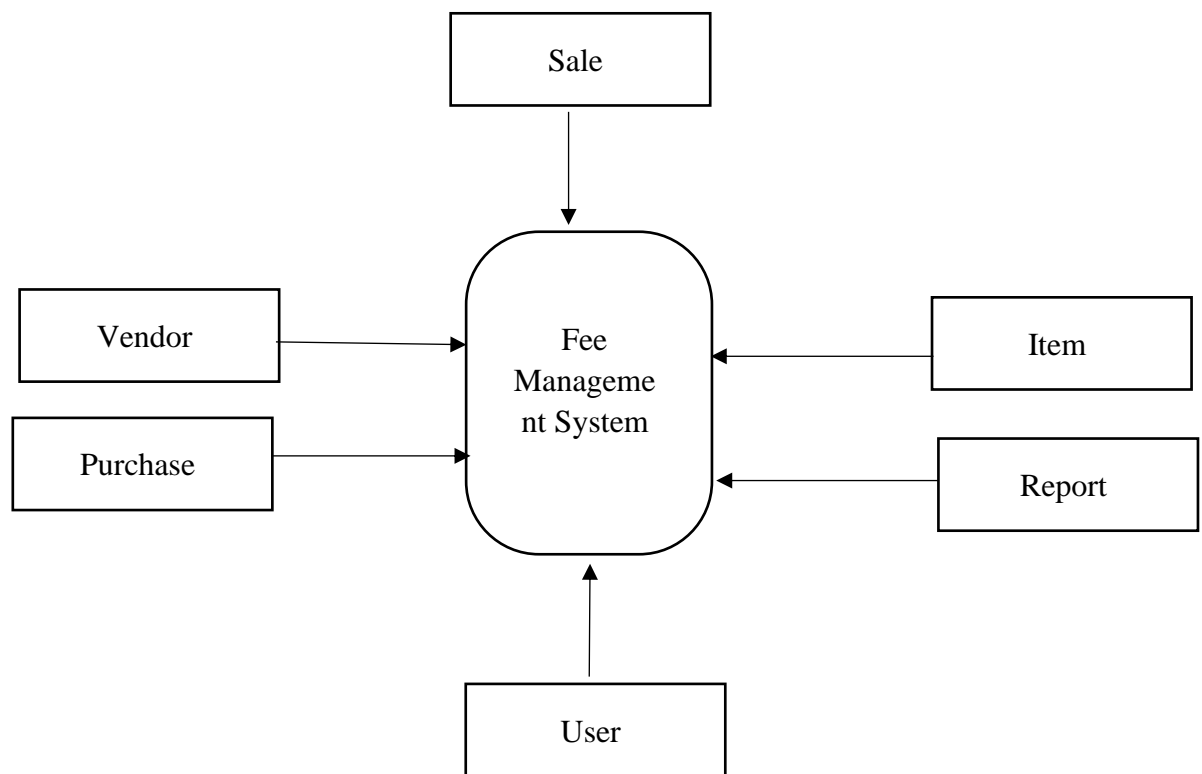
3.6 DATA FLOW DIAGRAM

DFD is an important tool used by system analysis. A data flow diagram model, a system using external entities from which data flows through a process which transforms the data and creates output data transforms which go to other processes external entities such as files. The main merit of DFD is that it can provide an overview of what data a system would process.

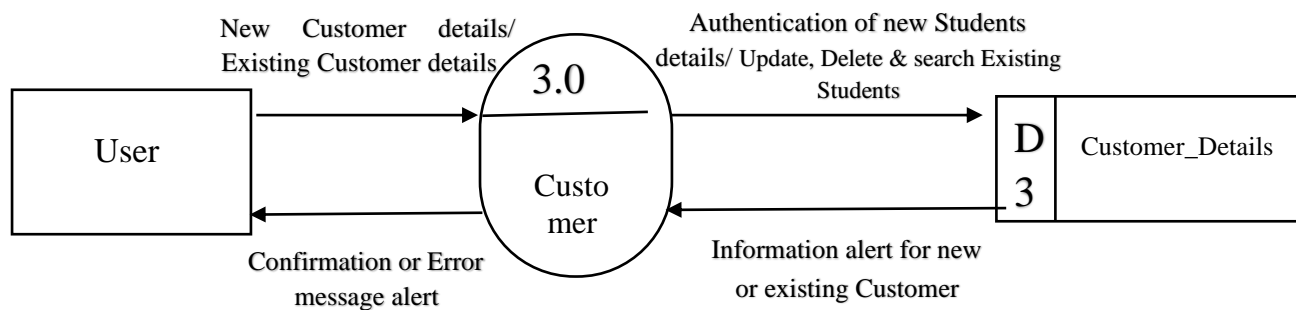
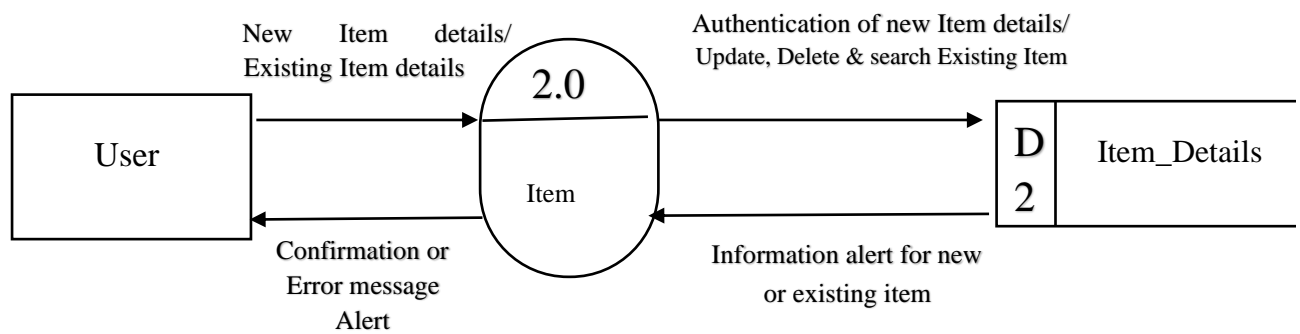
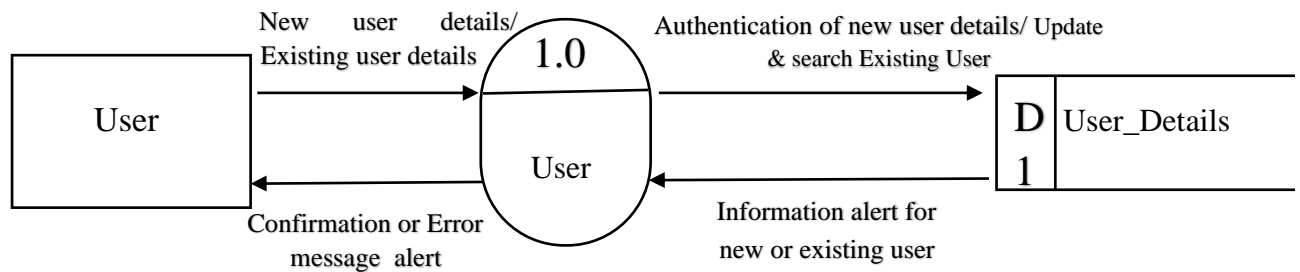
Symbols:

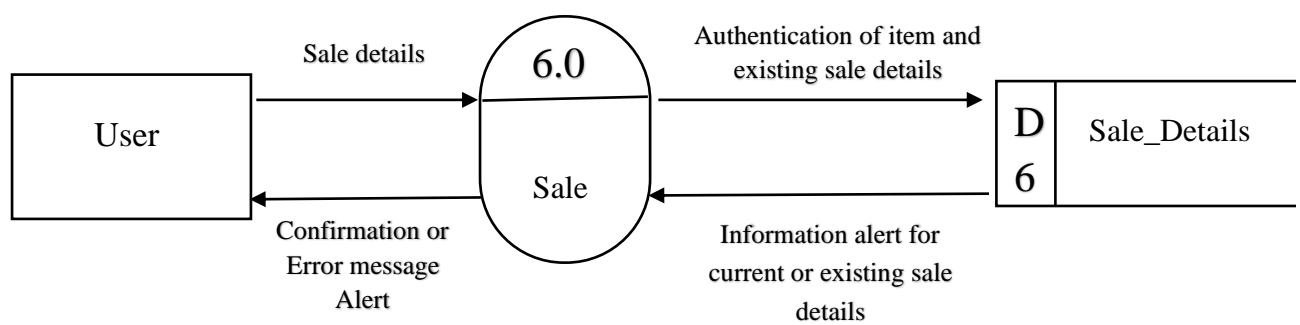
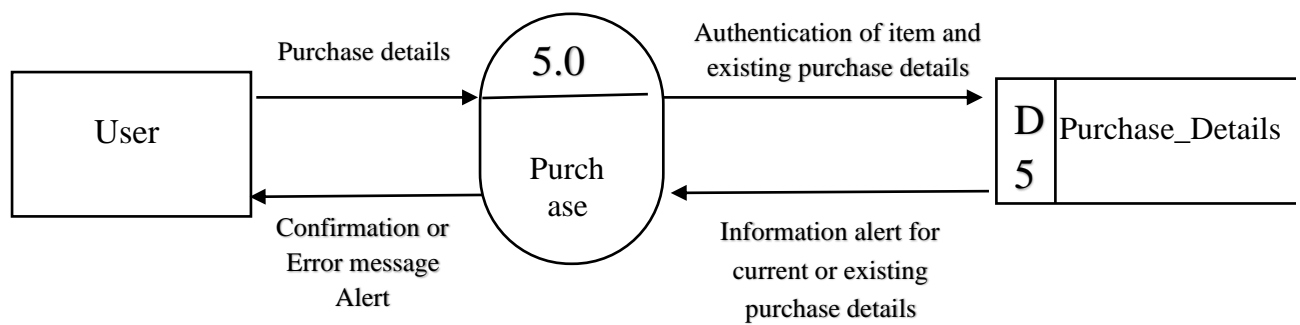
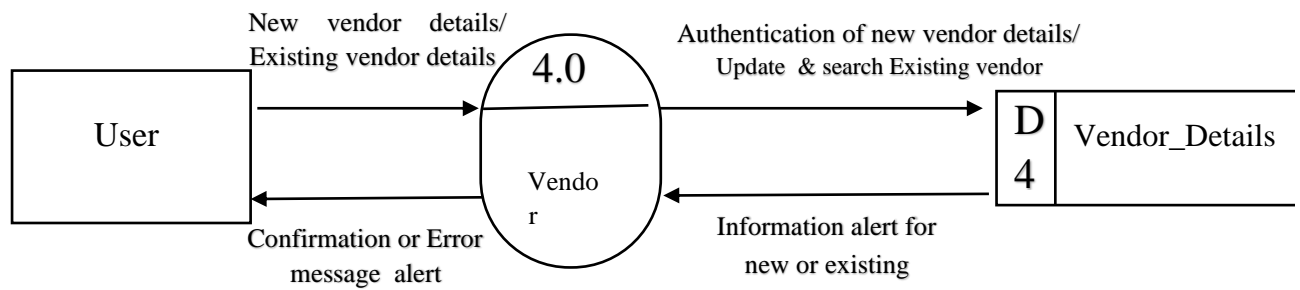


3.6.1. Context Level DFD

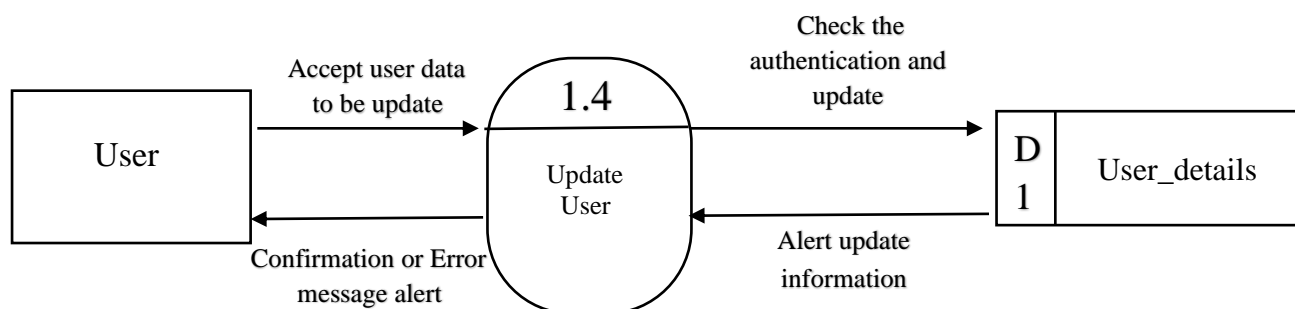
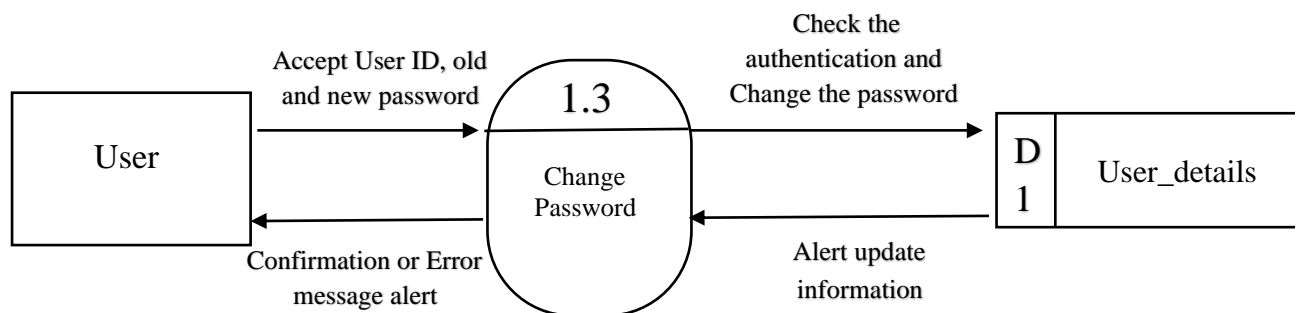
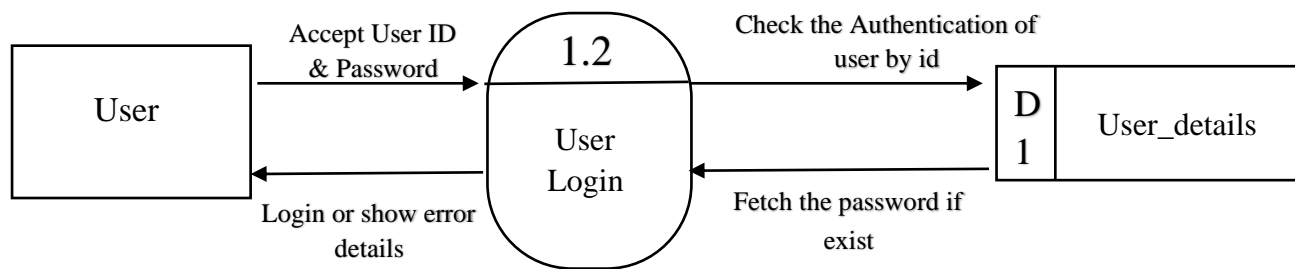
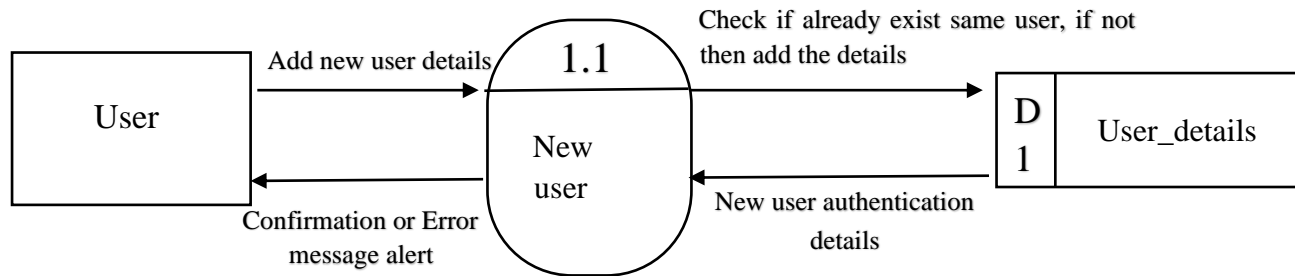


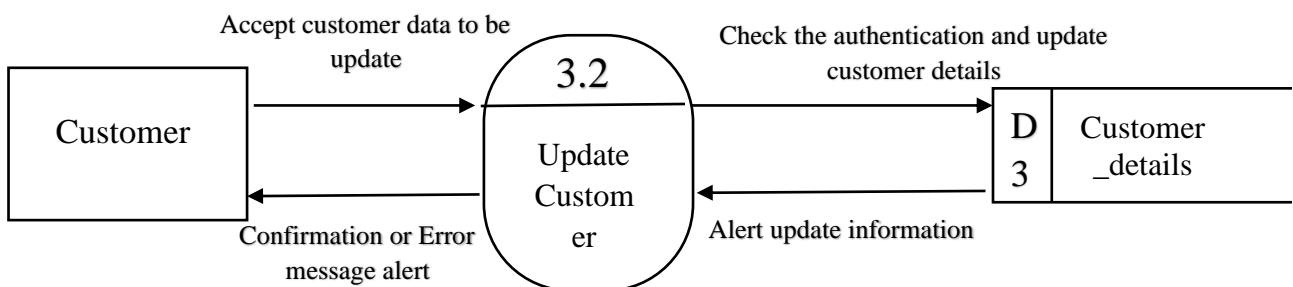
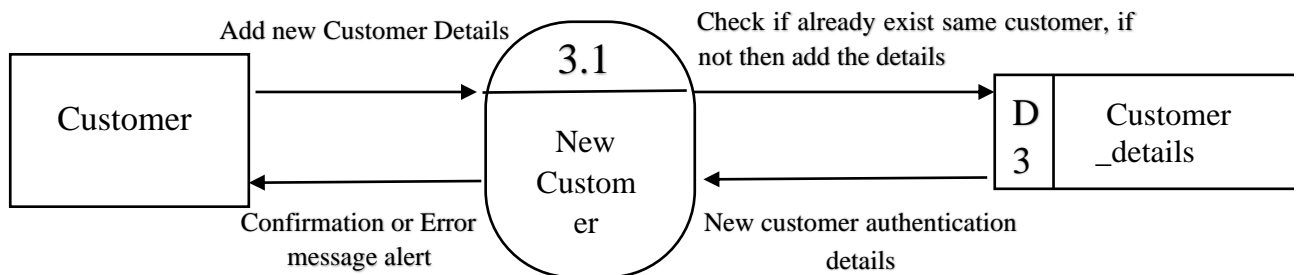
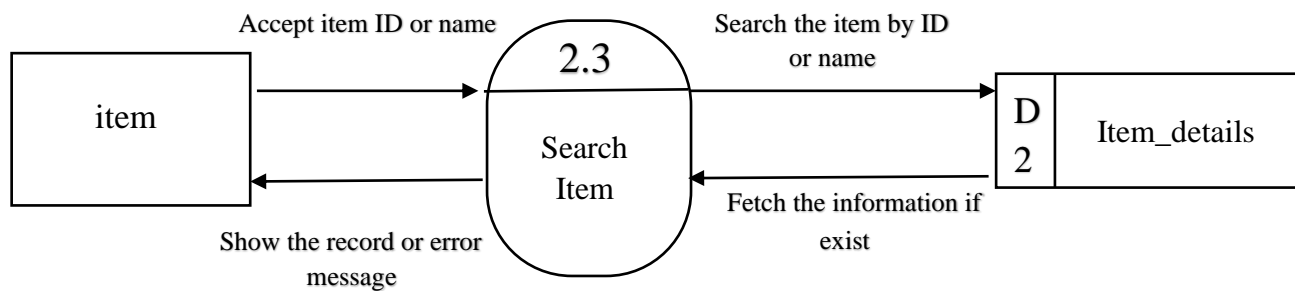
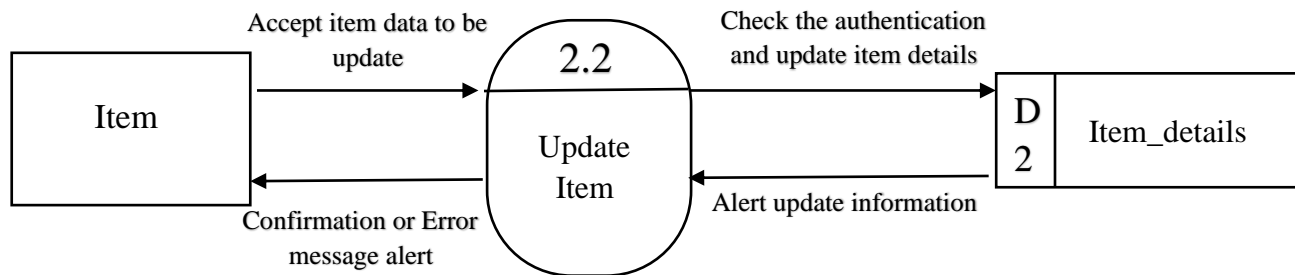
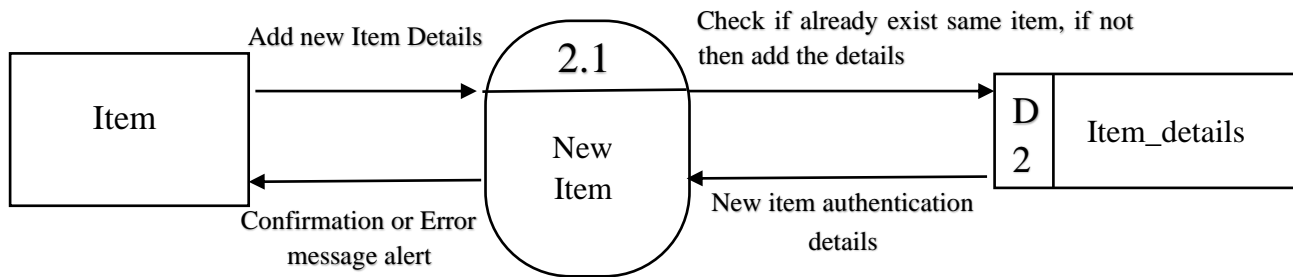
3.6.2 1st Level DFD

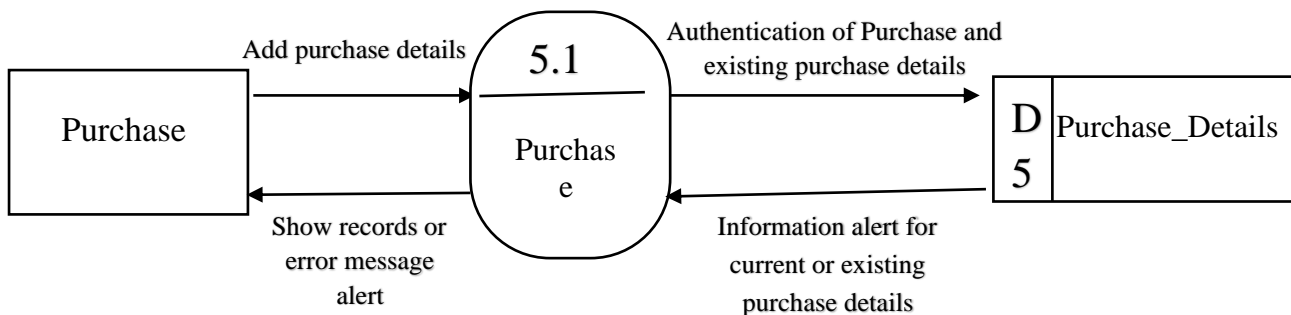
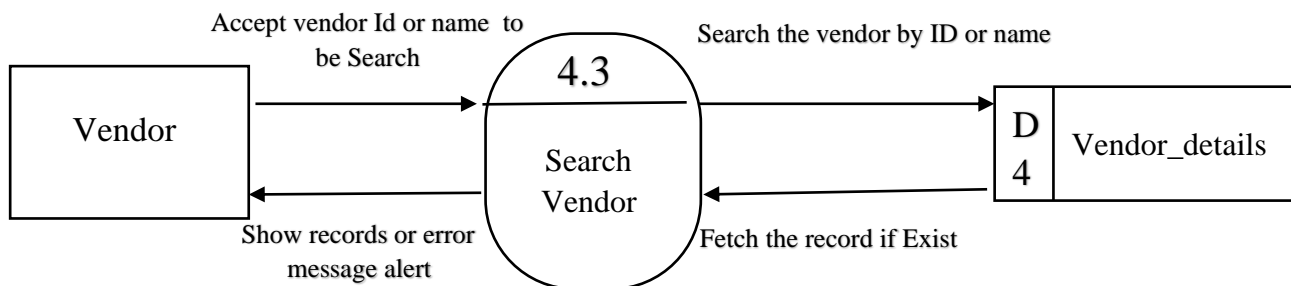
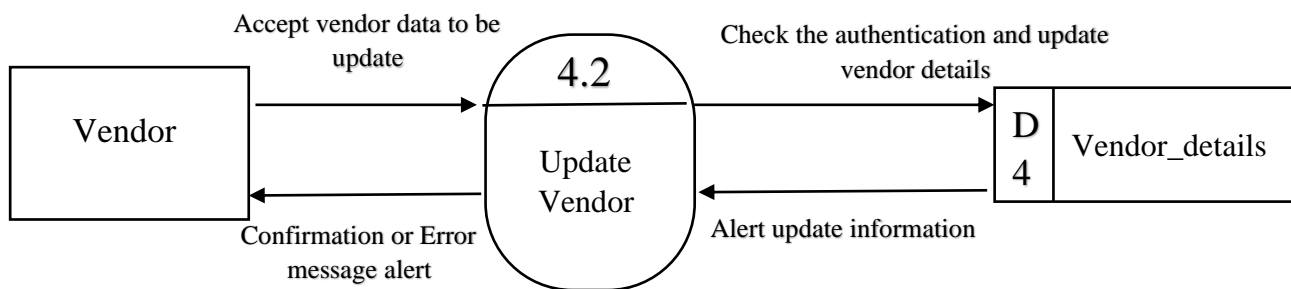
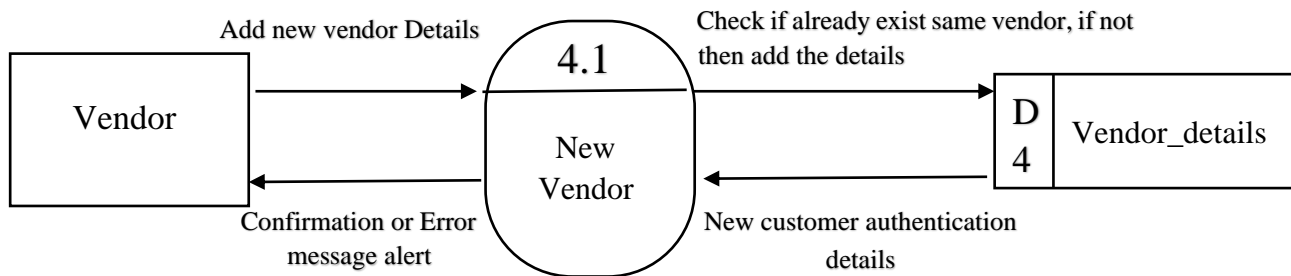
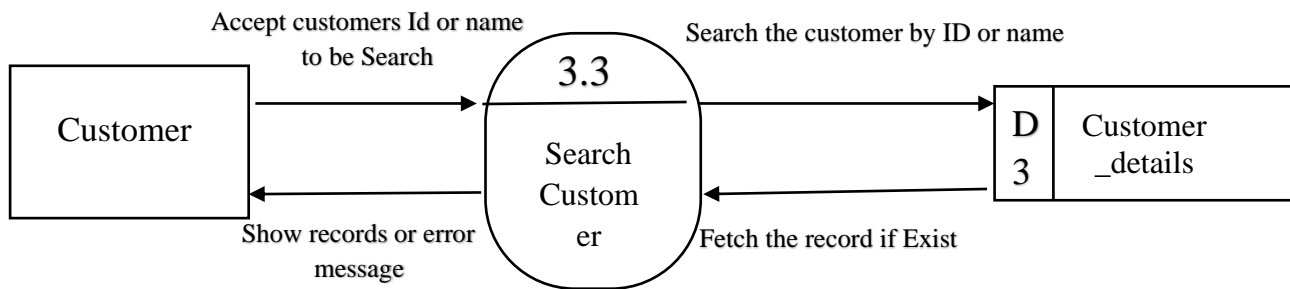


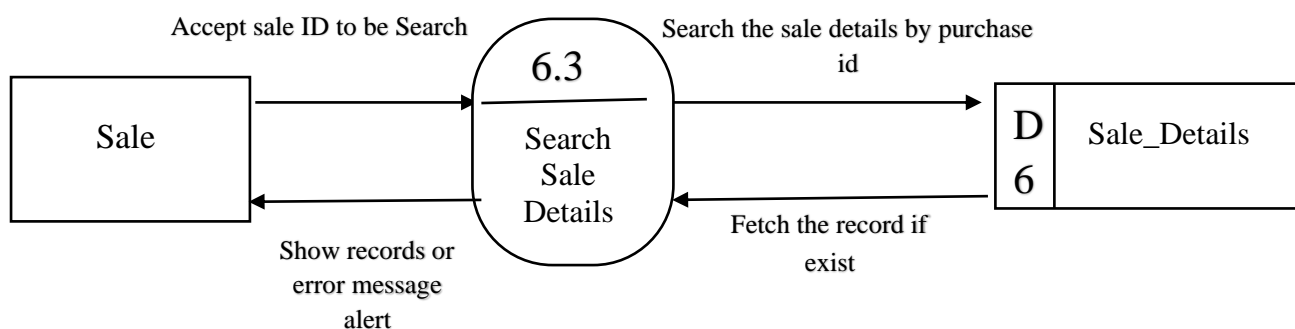
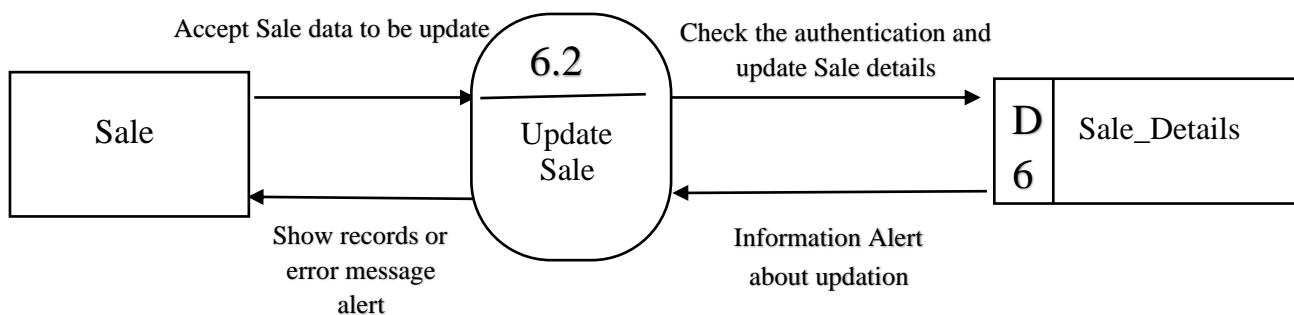
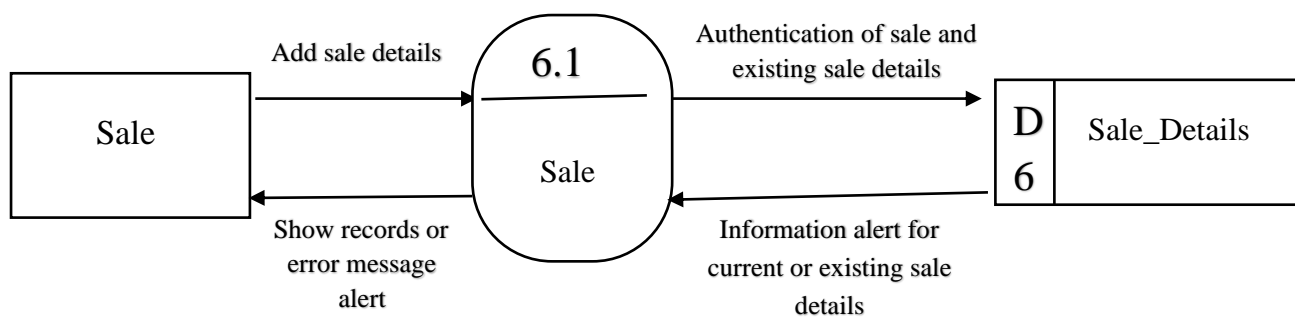
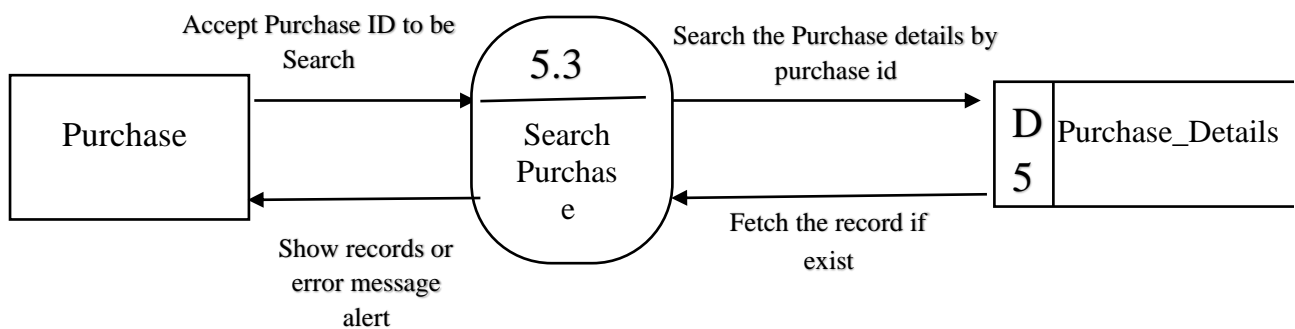
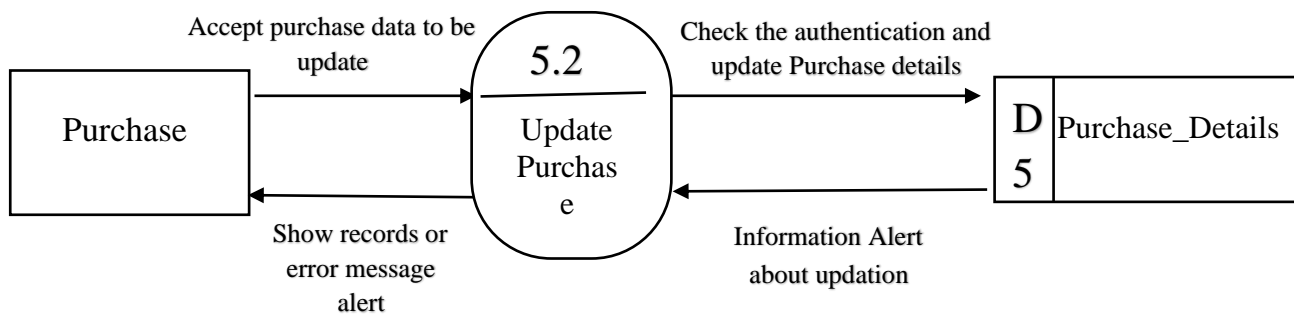


3.6.3 2nd Level DFD

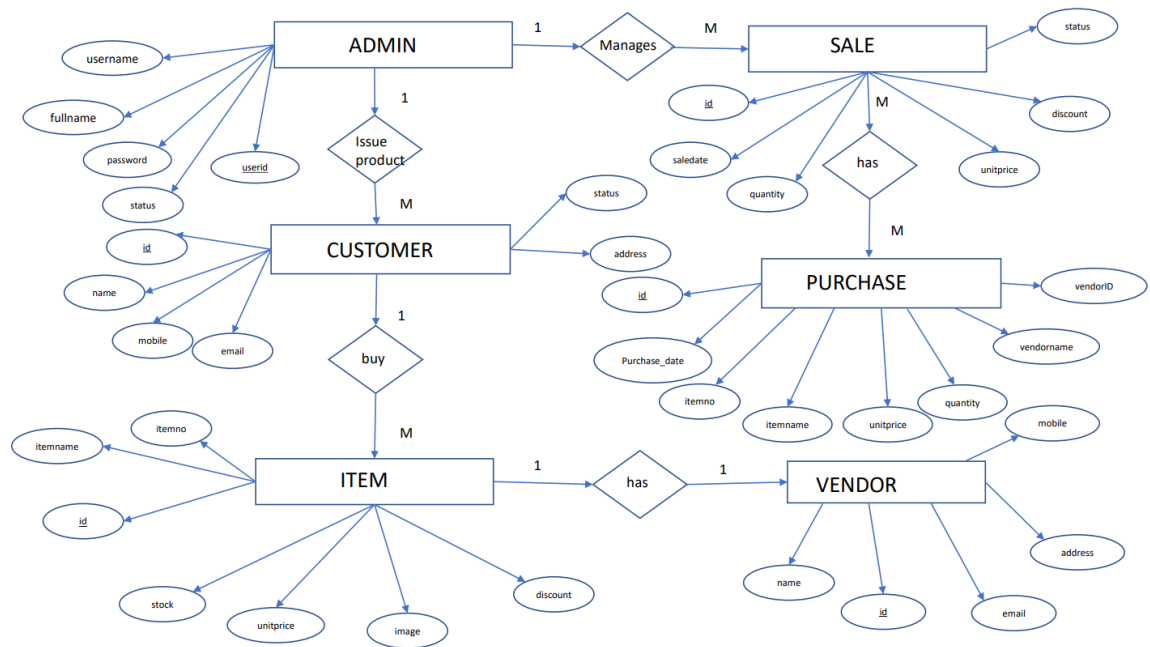








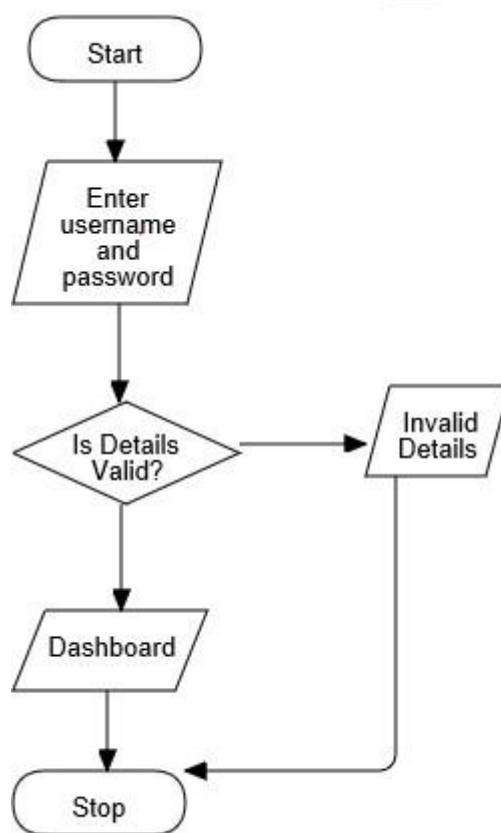
3.7. ER DIAGRAM



3.8. FLOW CHART

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

Login:



3.9. Normalization

Why Normalization:

Database normalization is the process of removing redundant data from your tables in order to improve storage efficiency, data integrity, and scalability .Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued.

0NF

COLUMN NAME	DATA TYPE
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
Address2	varchar(255)
City	varchar(30)
District	varchar(30)
status	varchar(255)
ProductID	Int(11)
itemNumber	varchar(255)
itemNumber	varchar(255)
Discount	float
stock	Int(11)
unitPrice	float
imageURL	varchar(255)
description	text
purchaseID	Int(11)
itemNumber	varchar(255)
purchaseDate	date
itemName	varchar(255)
quantity	Int(11)
unitPrice	float
VendorName	varchar(255)
vendorId	Int(11)
saleID	Int(11)
itemNumber	varchar(255)
customerID	Int(11)
customerName	varchar(255)
itemName	varchar(255)
saleDate	date
discount	float
quantity	Int(11)
unitPrice	float
userID	Int(11)
fullname	varchar(255)
username	varchar(255)
password	varchar(255)
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
City	varchar(30)
District	varchar(30)

1 NF

Customer table

COLUMN NAME	DATA TYPE
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
Address2	varchar(255)
City	varchar(30)
District	varchar(30)
status	varchar(255)

Item table

COLUMN NAME	DATA TYPE
ProductID	Int(11)
itemNumber	varchar(255)
itemName	varchar(255)
Discount	float
stock	Int(11)
unitPrice	float
imageURL	varchar(255)
status	varchar(255)
description	text

Purchase table

COLUMN NAME	DATA TYPE
purchaseID	Int(11)
itemNumber	varchar(255)
purchaseDate	date
itemName	varchar(255)
quantity	Int(11)
unitPrice	float
VendorName	varchar(255)
vendorId	Int(11)

Sale table

COLUMN NAME	DATA TYPE
saleID	Int(11)
itemNumber	varchar(255)
customerID	Int(11)
customerName	varchar(255)
itemName	varchar(255)
saleDate	date
discount	float
quantity	Int(11)
unitPrice	float

user table

COLUMN NAME	DATA TYPE
userID	Int(11)
fullname	varchar(255)
username	varchar(255)
password	varchar(255)
stauts	varchar(255)

vendor table

COLUMN NAME	DATA TYPE
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
Address2	varchar(255)
City	varchar(30)
District	varchar(30)
status	varchar(255)

2NF

Customer table

COLUMN NAME	DATA TYPE
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
Address2	varchar(255)
City	varchar(30)
District	varchar(30)
status	varchar(255)

Item table

COLUMN NAME	DATA TYPE
ProductID	Int(11)
itemNumber	varchar(255)
itemName	varchar(255)
Discount	float
stock	Int(11)
unitPrice	float
imageURL	varchar(255)
status	varchar(255)
description	text

Purchase table

COLUMN NAME	DATA TYPE
purchaseID	Int(11)
itemNumber	varchar(255)
purchaseDate	date
itemName	varchar(255)
quantity	Int(11)
unitPrice	float
VendorName	varchar(255)
vendorId	Int(11)

Sale table

COLUMN NAME	DATA TYPE
saleID	Int(11)
itemNumber	varchar(255)
customerID	Int(11)
customerName	varchar(255)
itemName	varchar(255)
saleDate	date
discount	float
quantity	Int(11)
unitPrice	float

Data table

COLUMN NAME	DATA TYPE
purchaseID	Int(11)
itemNumber	varchar(255)
purchaseDate	date
vendorId	Int(11)

user table

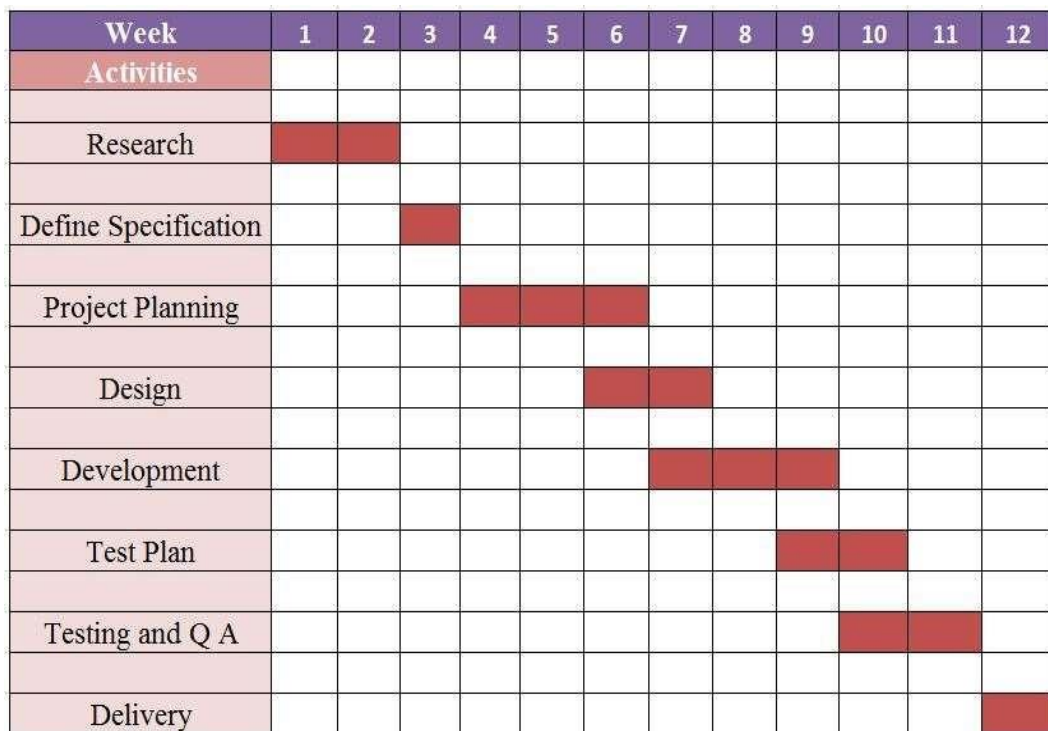
COLUMN NAME	DATA TYPE
userID	Int(11)
fullname	varchar(255)
username	varchar(255)
password	varchar(255)
status	varchar(255)

vendor table

COLUMN NAME	DATA TYPE
vendorID	Int(11)
fullName	varchar(100)
Email	varchar(100)
Mobile	Int(11)
Phone2	Int(11)
Address	varchar(255)
Address2	varchar(255)
City	varchar(30)
District	varchar(30)
status	varchar(255)

3.9. GANTT CHART

Gantt chart was invented by a mechanical engineer named Henry Gantt in 1910. A Gantt chart is simply a type of bar chart that visually represents a project plan over time. It shows start and end dates for tasks, displays milestones, and allows for dependencies between tasks. With all the features of Henry Gantt's project management system, it's no wonder that even now, more than 100 years later, the Gantt chart is still the preferred tool for managing projects of all sizes and types.



Ch.4. SYSTEM DESIGN

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirements have been specified and analyzed, system design is the first of the three technical activities design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.1.INPUT TO THE PROJECT

In order to complete the tasks of the Application and to get output by using this application work, there is need of some input based on the work that is to be carried out by using it. Input required for different purposes are:

1. Input for Login

- a. Username
- b. Password

2. Input for Update Admin Profile

- a. Name
- b. Username
- c. Email

3. Input for Add and Edit Customer

- a. Customer Name
- b. Customer Code

4. Input for Add and Edit Vendor

- a. Vendor Name
- b. Vendor Code

5. Input for Add and Edit Products

- a. Category Name
- b. Company Name
- c. Product Name
- d. Product Price

6. Input for Add and Edit Sale

- a. Sale Code

4.2. OUTPUT TO THE PROJECT

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

1. User friendliness is provided in the application with various controls.
2. The system makes the overall project management much easier and flexible.
3. There is no risk of data mismanagement at any level while the project development is under process.
4. It provides high level of security with different level of authentication.
5. User friendliness is provided in the application with various controls.
6. The system makes the overall project management much easier and flexible.
7. There is no risk of data mismanagement at any level while the project development is under process.
8. It provides high level of security with different level of authentication.

4.3. MODULARIZATION DETAILS

This section gives a brief description of the modules used in the project:

User Module

Input – User enter their details to create new account.

Process Definition – Checks whether he entered all details or not.

Output – User can login and manage the inventory details.

Customer Module

Input –User will enter the details of customer.

Process Definition – Checks whether he entered all details or not.

Output – Records will be added to the database.

Item Module

Input –User will enter the details of the new item or update existing item.

Process Definition – Checks whether he entered all details or not.

Output – Records will be added to the database .

Purchase Module

Input – User will enter the details of the purchased item.

Process Definition – Checks whether he entered all details or not.

Output – Record will be added to the database.

Sale Module

Input – User will enter the details of the total sale.

Process Definition – Checks whether he entered all details or not.

Output – Record will be added to the database.

Vendor Module

Input – User will enter the details of the vendor.

Process Definition – Checks whether he entered all details or not.

Output – Record will be added to the database.

4.5. DATA INTEGRITY

Data integrity is the overall completeness, accuracy and consistency of data. This can be indicated by the absence of alteration between two instances or between two updates of a data record, meaning data is intact and unchanged. Data integrity is usually imposed during the database design phase through the use of standard procedures and rules. The concept of data integrity ensures that all data in a database can be traced and connected to other data. This ensures that everything is recoverable and searchable. Having a single, well-defined and well-

controlled data integrity system increases stability, performance, reusability and maintainability. Data values are standardized according to a data model and data type. All characteristics of the data must be correct including business rules, relations, dates and definitions for data to be complete. Data integrity is imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation routines. As a simple example, to maintain data integrity numeric columns/cells should not accept alphabetic data .

4.6. DATA DICTIONARY

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.



































Database tables









User table: This table stores user login details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	userID	int(11)		No	None		AUTO_INCREMENT	Change Drop
<input type="checkbox"/>	2	fullName	varchar(255)	latin1_swedish_ci	No	None			Change Drop
<input type="checkbox"/>	3	username	varchar(255)	latin1_swedish_ci	No	None			Change Drop
<input type="checkbox"/>	4	password	varchar(255)	latin1_swedish_ci	No	None			Change Drop
<input type="checkbox"/>	5	status	varchar(255)	latin1_swedish_ci	No	Active			Change Drop





























☐ Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext









Customer table: This table stores customer details

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	customerID 	int(11)			No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	fullName	varchar(100)	latin1_swedish_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	email	varchar(100)	latin1_swedish_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	4	mobile	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	5	phone2	int(11)			Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	6	address	varchar(255)	latin1_swedish_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	7	address2	varchar(255)	latin1_swedish_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	8	city	varchar(30)	latin1_swedish_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	9	district	varchar(30)	latin1_swedish_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	10	status	varchar(255)	latin1_swedish_ci		No	Active			 Change  Drop  More
<input type="checkbox"/>	11	createdOn	timestamp			No	current_timestamp()			 Change  Drop  More

☐ Check all With selected:  Browse  Change  Drop  Primary  Unique  Index  Spatial  Fulltext

Item table: This table stores items details

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	productID 	int(11)			No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	itemNumber	varchar(255)	latin1_swedish_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	itemName	varchar(255)	latin1_swedish_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4	discount	float			No	0			 Change  Drop  More
<input type="checkbox"/>	5	stock	int(11)			No	0			 Change  Drop  More
<input type="checkbox"/>	6	unitPrice	float			No	0			 Change  Drop  More
<input type="checkbox"/>	7	imageURL	varchar(255)	latin1_swedish_ci		No	imageNotAvailable.jpg			 Change  Drop  More
<input type="checkbox"/>	8	status	varchar(255)	latin1_swedish_ci		No	Active			 Change  Drop  More
<input type="checkbox"/>	9	description	text	latin1_swedish_ci		No	None			 Change  Drop  More

☐ Check all With selected:  Browse  Change  Drop  Primary  Unique  Index  Spatial  Fulltext

Purchase table: This table stores purchase details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	purchaseID	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	itemNumber	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3	purchaseDate	date		No	None			Change Drop More
<input type="checkbox"/>	4	itemName	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	5	unitPrice	float		No	0			Change Drop More
<input type="checkbox"/>	6	quantity	int(11)		No	0			Change Drop More
<input type="checkbox"/>	7	vendorName	varchar(255) latin1_swedish_ci		No	Test Vendor			Change Drop More
<input type="checkbox"/>	8	vendorID	int(11)		No	0			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Sale table: This table stores sales details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	saleID	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	itemNumber	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3	customerID	int(11)		No	None			Change Drop More
<input type="checkbox"/>	4	customerName	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	5	itemName	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	6	saleDate	date		No	None			Change Drop More
<input type="checkbox"/>	7	discount	float		No	0			Change Drop More
<input type="checkbox"/>	8	quantity	int(11)		No	0			Change Drop More
<input type="checkbox"/>	9	unitPrice	float(10,0)		No	0			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Vendor table: This table stores vendors details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	vendorID	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	fullName	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3	email	varchar(255) latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4	mobile	int(11)		No	None			Change Drop More
<input type="checkbox"/>	5	phone2	int(11)		Yes	NULL			Change Drop More
<input type="checkbox"/>	6	address	varchar(255) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	7	address2	varchar(255) latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	8	city	varchar(30) latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	9	district	varchar(30) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	10	status	varchar(255) latin1_swedish_ci		No	Active			Change Drop More
<input type="checkbox"/>	11	createdOn	timestamp		No	current_timestamp()			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Ch 5. PROGRAM CODE AND TESTING

5.1. CODE DETAILS AND CODE EFFICIENCY

login.php

<?php

session_start();

// Check if user is already logged in

if(isset(\$_SESSION['loggedIn'])){\b

header('Location: index.php');

exit();

}

require_once('inc/config/constants.php');

require_once('inc/config/db.php');

require_once('inc/header.html');

?>

<body>

<?php

// Variable to store the action (login, register, passwordReset)

\$action = '';

if(isset(\$_GET['action'])){\b

\$action = \$_GET['action'];

if(\$action == 'register'){

?>

<div class="container">

<div class="row justify-content-center">

```

<div class="col-sm-12 col-md-5 col-lg-5">

    <div class="card">

        <div class="card-header">

            Register

        </div>

        <div class="card-body">

            <form action="">

                <div id="registerMessage"></div>

                <div class="form-group">

                    <label          for="registerFullName">Name<span
class="requiredIcon">*</span></label>

                    <input      type="text"      class="form-control"
id="registerFullName" name="registerFullName">

                    <!-- <small id="emailHelp" class="form-text text-
muted"></small> -->

                </div>

                <div class="form-group">

                    <label      for="registerUsername">Username<span
class="requiredIcon">*</span></label>

                    <input      type="email"      class="form-control"
id="registerUsername" name="registerUsername" autocomplete="on">

                </div>

                <div class="form-group">

                    <label      for="registerPassword1">Password<span
class="requiredIcon">*</span></label>

                    <input      type="password"      class="form-control"
id="registerPassword1" name="registerPassword1">

                </div>

                <div class="form-group">

                    <label          for="registerPassword2">Re-enter
password<span class="requiredIcon">*</span></label>

```



```

        <input type="password" class="form-control"
id="registerPassword2" name="registerPassword2">

        </div>

        <a href="login.php" class="btn btn-primary">Login</a>

        <button type="button" id="register" class="btn btn-
success">Register</button>

        <a href="login.php?action=resetPassword" class="btn btn-
warning">Reset Password</a>

        <button type="reset" class="btn">Clear</button>

    </form>

</div>

</div>

</div>

</div>

</div>

<?php

require 'inc/footer.php';

echo '</body></html>';

exit();

} elseif($action == 'resetPassword'){

?>

<div class="container">

    <div class="row justify-content-center">

        <div class="col-sm-12 col-md-5 col-lg-5">

            <div class="card">

                <div class="card-header">

                    Reset Password

                </div>

                <div class="card-body">

                    <form action="">

```

```

<div id="resetPasswordMessage"></div>

<div class="form-group">

    <label
for="resetPasswordUsername">Username</label>

        <input      type="text"      class="form-control"
id="resetPasswordUsername" name="resetPasswordUsername">

    </div>

    <div class="form-group">

        <label      for="resetPasswordPassword1">New
Password</label>

            <input  type="password"  class="form-control"
id="resetPasswordPassword1" name="resetPasswordPassword1">

        </div>

        <div class="form-group">

            <label  for="resetPasswordPassword2">Confirm
New Password</label>

                <input  type="password"  class="form-control"
id="resetPasswordPassword2" name="resetPasswordPassword2">

            </div>

            <a href="login.php" class="btn btn-primary">Login</a>

            <a href="login.php?action=register" class="btn btn-
success">Register</a>

            <button  type="button"  id="resetPasswordButton"
class="btn btn-warning">Reset Password</button>

            <button type="reset" class="btn">Clear</button>

        </form>

    </div>

</div>

</div>

</div>

</div>

</div>

<?php

```

```

        require 'inc/footer.php';

        echo '</body></html>';

        exit();
    }
}
?>

<!-- Default Page Content (login form) -->

<div class="container">

    <div class="row justify-content-center">

<div class="col-sm-12 col-md-5 col-lg-5">

    <div class="card">

        <div class="card-header">

            Login

        </div>

        <div class="card-body">

            <form action="">

                <div id="loginMessage"></div>

                <div class="form-group">

                    <label for="loginUsername">Username</label>

                    <input type="text" class="form-control" id="loginUsername"
name="loginUsername">

                </div>

                <div class="form-group">

                    <label for="loginPassword">Password</label>

                    <input type="password" class="form-control"
id="loginPassword" name="loginPassword">

                </div>

                <button type="button" id="login" class="btn btn-
primary">Login</button>

                <a href="login.php?action=register" class="btn btn-
success">Register</a>

```

```
                <a href="login.php?action=resetPassword" class="btn btn-  
warning">Reset Password</a>
```

```
                <button type="reset" class="btn">Clear</button>
```

```
            </form>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php
```

```
require 'inc/footer.php';
```

```
?>
```

```
</body>
```

```
</html>
```

customer

populateCustomerDetails.php

```
<?php
```

```
require_once(' ../inc/config/constants.php');
```

```
require_once(' ../inc/config/db.php');
```

```
// Execute the script if the POST request is submitted
```

```
if(isset($_POST['customerID'])){
```

```
    $customerID = htmlentities($_POST['customerID']);
```

```
    $customerDetailsSql = 'SELECT * FROM customer WHERE customerID =  
:customerID';
```

```
    $customerDetailsStatement = $conn->prepare($customerDetailsSql);
```

```

$customerDetailsStatement->execute(['customerID' => $customerID]);

// If data is found for the given item number, return it as a json object
if($customerDetailsStatement->rowCount() > 0) {
    $row = $customerDetailsStatement->fetch(PDO::FETCH_ASSOC);
    echo json_encode($row);
}
$customerDetailsStatement->closeCursor();
}
?>

```

populateLastCustomerID.php

```

<?php
require_once(' ../inc/config/constants.php');
require_once(' ../inc/config/db.php');

$sql = "SELECT MAX(customerID) FROM customer";
$stmt = $conn->prepare($sql);
$stmt->execute();
$row = $stmt->fetch(PDO::FETCH_ASSOC);

echo $row['MAX(customerID)'];
$stmt->closeCursor();
?>

```

saleReportsSearchTableCreator.php

```

<?php
require_once(' ../inc/config/constants.php');

```

```
require_once('../inc/config/db.php');
```

```
$uPrice = 0;
```

```
$qty = 0;
```

```
$totalPrice = 0;
```

```
$saleDetailsSearchSql = 'SELECT * FROM sale';
```

```
$saleDetailsSearchStatement = $conn->prepare($saleDetailsSearchSql);
```

```
$saleDetailsSearchStatement->execute();
```

```
$output = '<table id="saleReportsTable" class="table table-sm table-striped table-bordered table-hover" style="width:100%">
```

```
    <thead>
```

```
        <tr>
```

```
            <th>Sale ID</th>
```

```
            <th>Item Number</th>
```

```
            <th>Customer ID</th>
```

```
            <th>Customer Name</th>
```

```
            <th>Item Name</th>
```

```
            <th>Sale Date</th>
```

```
            <th>Discount %</th>
```

```
            <th>Quantity</th>
```

```
            <th>Unit Price</th>
```

```
            <th>Total Price</th>
```

```
        </tr>
```

```
    </thead>
```

```
    <tbody>';
```

```
// Create table rows from the selected data
```

```

while($row = $saleDetailsSearchStatement->fetch(PDO::FETCH_ASSOC)){

    $uPrice = $row['unitPrice'];

    $qty = $row['quantity'];

    $discount = $row['discount'];

    $totalPrice = $uPrice * $qty * ((100 - $discount)/100);

    $output .= '<tr>' .

        '<td>' . $row['saleID'] . '</td>' .

        '<td>' . $row['itemNumber'] . '</td>' .

        '<td>' . $row['customerID'] . '</td>' .

        '<td>' . $row['customerName'] . '</td>' .

        '<td>' . $row['itemName'] . '</td>' .

        '<td>' . $row['saleDate'] . '</td>' .

        '<td>' . $row['discount'] . '</td>' .

        '<td>' . $row['quantity'] . '</td>' .

        '<td>' . $row['unitPrice'] . '</td>' .

        '<td>' . $totalPrice . '</td>' .

        '</tr>';

}

```

```

$saleDetailsSearchStatement->closeCursor();

```

```

$output .= '</tbody>'

```

```

<tfoot>

    <tr>

        <th>Total</th>

        <th></th>

        <th></th>

        <th></th>

```

```

<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</tfoot>
</table>';

```

```
echo $output;
```

```
?>
```

5.2. TESTING APPROACH

Types Of Testing

The system was designed according to the requirement of the system. But we are not 100% confident. The lack of confidence stems from several things. First the system deals with large number of states, complex logic and activities. So some error might occur in the system. Error may be software, which is known as “SOFTWARE ERROR” i.e. the software doesn’t do what the requirement says. So an exhaustive and thorough testing must be conducted to ascertain. Whether the system produces right results. The project guide and the user both did testing.

Module Testing

The testing was done in several stages. First each program module was tested as a single program, which is also known as module testing or unit testing. In unit testing a set of data as input was given to the module and observed what output data is produced. In addition, the logic and boundary condition for input and output data was also checked. The interface between this module and others was checked for correctness. While collecting the input data for testing the program module it was kept in mind that input should be from all classes, so the entire condition of the program could also be checked.

Interrogating Testing

When the individual program modules were working properly, we combine the module in the working system. This integration is planned and coordinated so that when an error occurs, we have an idea of what caused it. Integration testing is the process of verifying that the components of a system work together as described in the program design specification. For testing, the system was viewed as a hierarchy of modules. We began with the module at the highest level of design and worked down. The next modules to be tested are those that call previously tested modules.

Function Testing

Once we sure that information is passed among modules according to the design prescription we tested the system to assure whether the function described the requirement specification are actually performed by the integrated system.

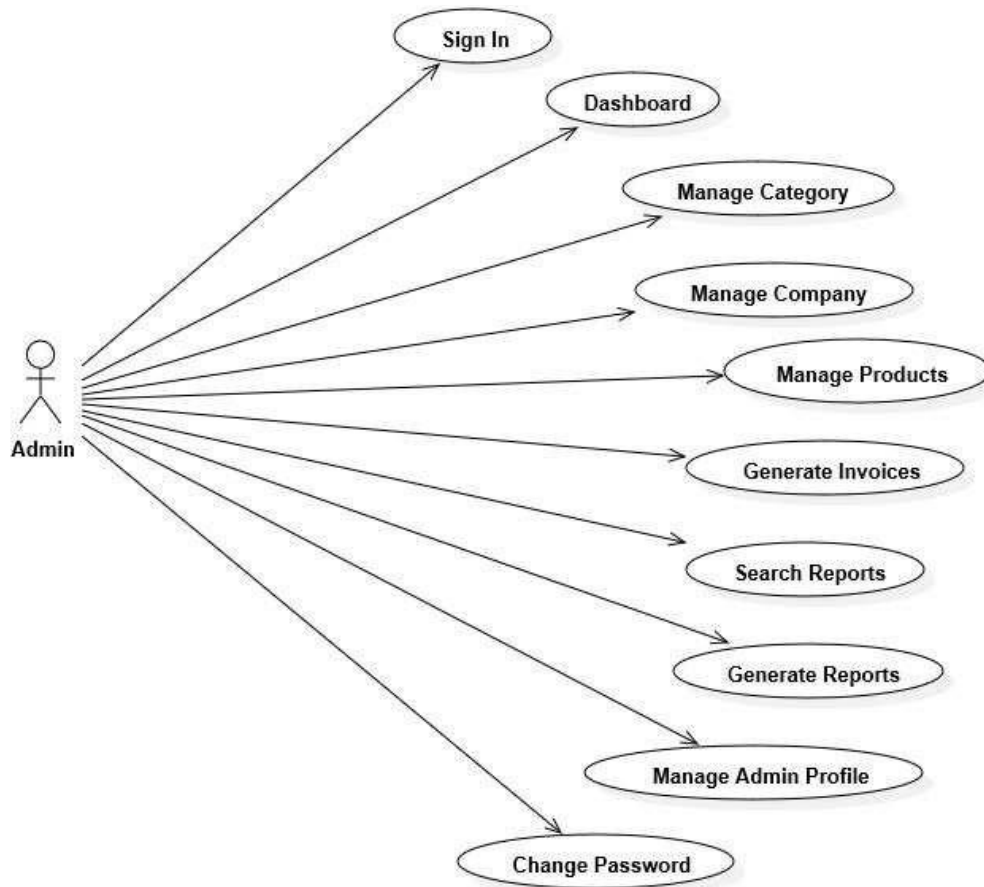
Acceptance Test

When the function test completes then we involved the user to make sure that system works according to the user's expectation. Thus, the user did the acceptance test.

Implementation

Once the system was tested (module wise as well as integrated) satisfactorily, and then comes the implementation of the system. Implementation is the process of changing from old system (manual) to the new system (computerized). Some training was also given to the user about how to work on the new system and finally the system was successfully adopted.

5.2.1 USE CASE DIAGRAM



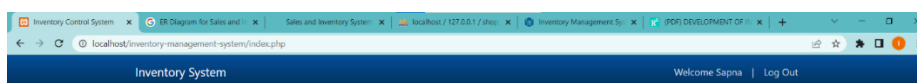
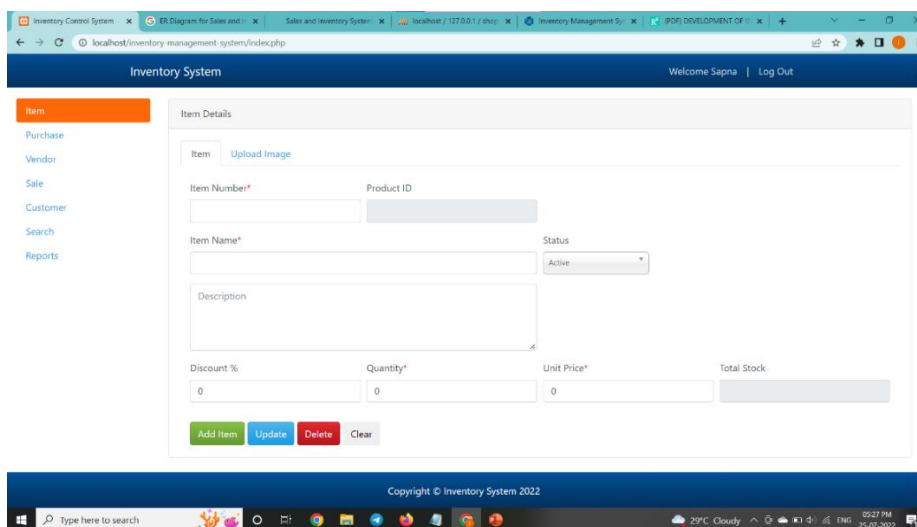
5.3. VALIDATION CHECK

The basic objective of the requirement validation activity is to ensure that “Inventory Management System” reflects the actual requirements accurately and clearly. Checking data for appropriateness and accuracy is called validating. Data entry checks can be built into the database packages to prevent invalid data from being entered. For instance, in database, format checking (determining if the data are in the correct format) is done automatically because at the time of creating a database file.

Range checking (determining if the data fall within an acceptable range) and accuracy checking (making sure that an entry is possible) can be built by writing a simple program in database.

Ch 6. RESULT AND DISCUSSION

6.1 USER DOCUMENTATION



Inventory Control System x ER Diagram for Sales and x Sales and Inventory System x localhost / 127.0.0.1 / shop x Inventory Management Sy x (PDF) DEVELOPMENT OF x +

localhost/inventory-management-system/index.php

Inventory System Welcome Sapna | Log Out

Item
Purchase
Vendor
Sale
Customer
Search
Reports

Vendor Details

Full Name* Status Vendor ID

Phone (mobile)* Phone 2 Email

Address*

Address 2

City District

Colombo

Add Vendor Update Delete Clear

Copyright © Inventory System 2022

Type here to search 29°C Cloudy 05:28 PM 25-07-2022

Inventory Control System x ER Diagram for Sales and x Sales and Inventory System x localhost / 127.0.0.1 / shop x Inventory Management Sy x (PDF) DEVELOPMENT OF x +

localhost/inventory-management-system/index.php

Inventory System Welcome Sapna | Log Out

Item
Purchase
Vendor
Sale
Customer
Search
Reports

Sale Details

Item Number* Customer ID* Customer Name Sale ID

Item Name Sale Date*

2018-05-24

Total Stock Discount % Quantity* Unit Price* Total

0 0 0

Add Sale Update Clear

Copyright © Inventory System 2022

Type here to search 29°C Cloudy 05:30 PM 25-07-2022

Inventory Control System x ER Diagram for Sales and x Sales and Inventory System x localhost / 127.0.0.1 / shop x Inventory Management Sy x (PDF) DEVELOPMENT OF x +

localhost/inventory-management-system/index.php

Inventory System Welcome Sapna | Log Out

Item
Purchase
Vendor
Sale
Customer
Search
Reports

Customer Details

Full Name* Status Customer ID

Phone (mobile)* Phone 2 Email

Address*

Address 2

City District

Colombo

Add Customer Update Delete Clear

Copyright © Inventory System 2022

Type here to search 29°C Cloudy 05:30 PM 25-07-2022

Inventory System

Welcome Sapna | Log Out

Vendor
Sale
Customer
Search
Reports

Item Customer Sale Purchase Vendor

Use the grid below to search all details of items

Show 10 entries

Search:

Product ID	Item Number	Item Name	Discount %	Stock	Unit Price	Status	Description
35	2	School Bag	0	12	500	Active	
36	3	Office Bag	0	5	1300	Active	
37	4	Leather Bag	2	6	3409	Active	
38	5	Travel Bag	2	17	1200	Active	
39	6	Gym Bag	0	0	3000	Active	
40	7	Handbag	1.5	10	1650	Active	
41	8	Laptop Bag	2.1	9	2300	Active	
43	10	Sports Bag	1	102	1000	Active	
49	14	Hiking Bag	1.5	5	1200	Active	This is a hiking bag. Ideal for long distance hikes. Light-weight and water proof.
51	15	handpurse	0	10	1000	Active	
Product ID	Item Number	Item Name	Discount %	Stock	Unit Price	Status	Description

Inventory System

Welcome Sapna | Log Out

Item
Purchase
Vendor
Sale
Customer
Search
Reports

Reports

Item Customer Sale Purchase Vendor

Use the grid below to get reports for items

Show 10 entries

Copy CSV Excel PDF Print

Search:

Product ID	Item Number	Item Name	Discount %	Stock	Unit Price	Status	Description
35	2	School Bag	0	12	500	Active	
36	3	Office Bag	0	5	1300	Active	
37	4	Leather Bag	2	6	3409	Active	
38	5	Travel Bag	2	17	1200	Active	
39	6	Gym Bag	0	0	3000	Active	
40	7	Handbag	1.5	10	1650	Active	
41	8	Laptop Bag	2.1	9	2300	Active	
43	10	Sports Bag	1	102	1000	Active	
49	14	Hiking Bag	1.5	5	1200	Active	This is a hiking bag. Ideal for long distance hikes. Light-weight and water proof.
51	15	handpurse	0	10	1000	Active	
Product ID	Item Number	Item Name	Discount %	Stock	Unit Price	Status	Description

Showing 1 to 10 of 10 entries

Previous 1 Next

Copyright © Inventory System 2022

Ch 7. CONCLUSION

7.1. LIMITATIONS

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.
- It don't have any industrial implementation, it is just applicable to be implemented for small shops to main its shops products details and for providing bills to the customer.

7.2. FURTHER SCOPE

- Some new features like **charts, pipes, day to day sale** can be added to the existing system as per requirements.
- This project can be further customized and used other shops and stock managements.
- Our project can be used in any shops and used for billing and keeping records of the products as well. It is best way to keep the information and data regarding products, vendor, customers for future aspects and saving time.

7.3. CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in PHP and MySQL web-based application. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

7.4. REFERENCES

For PHP

- <https://www.w3schools.com/php/default.asp>
- <https://www.sitepoint.com/php/>
- <https://www.php.net/>

For MySQL

- <https://www.mysql.com/>
- <http://www.mysqltutorial.org>

For XAMPP

- <https://www.apachefriends.org/download.html>