

**A MINI PROJECT REPORT ON**

**“COMPUTER HARDWARE SHOP MANAGEMENT”**

Submitted by

**Mr. BHOSALE DARSHAN BHIMRAO (24006)**

Under the Guidance of

**Dr. Ashwyn Kumar**

Submitted to



**SAVITRIBAI PHULE  
PUNE UNIVERSITY**

**As a partial fulfillment for the award of degree of  
MASTER IN COMPUTER APPLICATION**

**Semester – I**

**At**



**ASM's IBMR Pune – Institute of Business Management  
and Research, Chinchwad  
PUNE-19**

**(Affiliated to SPPU & Approved by AICTE)  
Session: 2021-23**

## Declaration

I hereby declare that the present mini project work “**COMPUTER HARDWARE SHOP MANAGEMENT**” is original work carried out under the guidance of Prof. Name of guide, MCA Department, ASM’s Institute of Business Management and Research, MCA, Chinchwad. It has not been submitted by me in part or full to any University for any examination before. This work has been carried out by me at the Savitribai Phule Pune University during the academic session 2021-2022.

Date:

**BHOSALE DARSHAN BHIMRAO**

(MCA. I Semester)

Department of Master of Computer Application,  
ASM’s Institute of Business Management and Research, MCA,  
Chinchwad

## CERTIFICATE

*This is to certify that the mini project entitled*

**“COMPUTER HARDWARE SHOP MANAGEMENT”**

*is a bonafied work and it is submitted to the Savitribai Phule Pune University*

*By*

**BHOSALE DARSHAN BHIMRAO**

*In the partial fulfillment of the degree of Master Of Computer Appliction, during the academic year **2021-2022** under my guidance.*

**Dr. Ashwyn Kumar**

*Guide*

*Department of MCA*

*ASM's IBMR, MCA,*

*Chinchwad*



**Master of Computer Application,  
ASM's Institute of Business Management and Research, MCA, Chinchwad  
Session 2021-2022**

## EXAMINER CERTIFICATE

*This is to certify that the mini project on*

**“COMPUTER HARDWARE SHOP MANGEMENT”**

*is examined by the following examiners on date     /     /     2022*

*Internal Examiner*

*External Examiner*

## ACKNOWLEDGEMENT

It is a pleasure to acknowledge the assistance of several people and institutions in this effort. Honestly speaking, this project has turned me into a debtor. First and foremost, I feel indebted to my guide, **Dr. Ashwyn Kumar**, Department of MCA, ASM's IBMR, MCA, Chinchwad for his/her valuable guidance, continuous support and advice and constant encouragement throughout my project work.

I would like to extend my gratitude to honourable **Dr. V. P. Pawar sir**, Director, ASM's IBMR, MCA, Chinchwad, for being a constant source of inspiration. Finally, I would like to extend my thanks to all those who have contributed, directly or indirectly to make this project successful.

**Mr. BHOSALE DARSHAN BHIMRAO (24006)**

## INDEX

Sr No	Topic	Page no.
Chapter 1	Introduction	8 to 12
1.1	Existing System and Need for System	9
1.2	Scope of Work	9
1.3	Operating Environment – Hardware and Software	10
1.4	Detail Description of Technology Used	11
1.5	Feasibility Study	12
Chapter 2	Proposed System	14 to 17
2.1	Proposed System	14
2.2	Objectives of System	14
2.3	User Requirements	14
Chapter 3	Analysis & Design	18 to 37
3.1	Use Case Diagram	18
3.2	Data Flow Diagram	20
3.3	E-R Diagram	23
3.4	Sequence Diagram	25
3.5	Class Diagram	28
3.6	User Interface Design (Screens etc.)	29
3.7	Table specifications	32
3.8	Testing And Implementation	36
Chapter 4	User Manual	38 to 39
4.1	User Manual	38

4.2	Operations Manual / Menu Explanation	38
Chapter 5	Drawbacks And Limitations	40
Chapter 6	Conclusion	41
Chapter 7	System Coding	42
Chapter 8	Bibliography	76

## CHAPTER -1

### INTRODUCTION

This software enables the products acquisition of sales and return operations on stalls and shop-in-shop systems with integrated product sale software and back-office functions. This software controls the entire lifecycle in the store through receiving, sales, inventory management. Role-based security ensures that access to specific functionality is limited to appropriate levels.

The Computer Hardware Shop project is about the working process of Computer store including purchase, sales, stock maintaining, manpower utilization, maintaining customer and supplier information. The project provides computerization of the whole work, thus it helps to improve the quality and productivity of the store. The project will help store employees to maintain all related information about the purchase products, sales product, stock available in the store, supplier and customer. This project decreases the work load of store employees and provides suitable information about each product so that with the help of this project store employees can easily and quickly find the products, supplier, customer information whenever necessary.

The Computer Hardware Shop is to develop software based information of Computer goods shopping. In global business market the electronic products are most important accessories in real life. Visualizing the huge opportunity, this is an effort to maximize the business through the development of this software and keeping the data and thus increasing the customer base from the local as well as global markets around the world.

My ongoing project is a software application because nowadays software is a prominent tool of marketing mantra. With the advent of the software technologies, world has become a global village. Every year, millions more people around the world are added to the existing customer base. So considering a big hike in the revenue in this booming sector and one of the successful businesses through this software one should be proud to have such a technical deal.



## 1.1 Existing System and Proposed System:

### Existing System:

The project on “COMPUTER HARDWARE SHOP” enables to keep the records in the computer system with JAVA as Front-end and Microsoft SQL Server as Backend.

1. The project keeps the detail record of Suppliers, Customers, Stock of parts, Employees of the Store.
2. The project provides the quotation of product to the customer which enables the customers to make his choice.
3. If the product not available in the store then Shopkeeper order to supplier for providing particular product which customer want.
4. Then related information about that ordered product stored in database of the store.
5. After the order given by the customer it generate the bill.

### Proposed System:

The proposed system that we are making will be completely different and more importantly it will be user friendly.

It can be operated by anyone. It comes with some up-to-date functions which you can perform by just clicking a mouse button.

The aim of this system is to advance to the next level by making everything computerized. It has advantages like this:

- 1) It's user-friendly and it's completely computerized.
- 2) It has a huge data base which can store lots and lots of information.
- 3) Data can be easily entered, updated. I.e. data can be managed easily.
- 4) No need to worry about the storage of the data as it will be stored in your computer.

Search results can be shown to the clients at the same place and proper and correct information can be provided.

## 1.2 Scope Of Work

The existing system follows the same old approach of dealing with the customers in person and storing their records and all other data in files and registers.

This manual system is time consuming and not feasible to use. The existing system is completely manual and depended on human beings.

It has following limitations:

- ✚ Data has to be stored manually in files and registers.
- ✚ Retrieval of this data is very slow and hectic because one has to search all the files manually.
- ✚ As the INTEGER of files increases, it takes up lot of space making it difficult to manage.
- ✚ Searching for old bills and files or registers takes a lot of time.
- ✚ Data loss is always a major concern as a file or a register can be easily misplaced.
- ✚ Natural calamity like floods, fire break-out and other such things can destroy the data, and it can lead to data loss.

## 1.3 Operating Environment:

### HARDWARE:

Sr No.	Name of Resource/ Material	specifications	Quantity
1.	Processor	Corei5 7 <sup>th</sup> / 1tb HDD & 120GB SSD/ nvidea 4GB graphics card	1
2.	Primary Memory	8GB	1
3.	Operating System	64-bit	1
4.	Computer System	Windows 10	1

### SOFTWARE:

1.	Software Required	Eclipse Java 2019-09	1
2.	Front End	JAVA	-
3.	Back End	MySQL	-

## 1.4 Detail Description of Technology Used:

### 1.4.1 About JAVA

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

Java Platform is a collection of programs that help programmers to develop and run Java programming applications efficiently. It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications. James Gosling developed the Java platform at Sun Microsystems, and the Oracle Corporation later acquired it.

#### **1.4.1.1 Here are some important Java applications:**

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

#### **1.4.1.2 History of Java Programming Language :**

- The Java language was initially called OAK.
- Originally, it was developed for handling portable devices and set-top boxes. Oak was a massive failure.
- In 1995, Sun changed the name to "Java" and modified the language to take advantage of the burgeoning www (World Wide Web) development business.
- Later, in 2009, Oracle Corporation acquired Sun Microsystems and took ownership of three key Sun software assets: Java, MySQL, and Solaris.

### 1.4.2 About MySQL

MySQL was created by a Swedish company, MySQL AB, founded by Swedes David Axmark, Allan Larsson and Finland Swede Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietary licensed) mSQL antecedent.

#### **1.4.2.1 RDBMS Terminology**

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- **Database** – A database is a collection of tables, with related data.
- **Table** – A table is a matrix with data. A table in a database looks like a simple spreadsheet.

- **Column** – One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** – A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** – Storing data twice, redundantly to make the system faster.
- **Primary Key** – A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- **Foreign Key** – A foreign key is the linking pin between two tables.
- **Compound Key** – A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** – An index in a database resembles an index at the back of a book.
- **Referential Integrity** – Referential Integrity makes sure that a foreign key value always points to an existing row.

#### 1.4.2.2 MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

### 1.5 FEASIBILITY STUDY :

- In preliminary investigation, we examine the feasibility study to understanding the possibility of proposed system being useful in future.
- A feasibility study is a management oriented activity.
- It gives us the cost & benefits of our software define for proposed system.
- The feasibility studied are categories as-

- ✓ **Technical Feasibility**
- ✓ **Economical Feasibility**
- ✓ **Operational Feasibility**

### 1.5.1 Technical Feasibility

1. A technical feasibility study is an excellent tool for trouble -shooting and long-term planning.
2. Proposed application will provide enough response to enquiries regardless of the INTEGER of users.
3. This software provides lot of security, accuracy & reliability rather than existing system.
4. The proposed new system can easily be expanded to satisfy the new requirements in future.
5. Technical feasibility also acts as flow chat to show how our application evolves & adapts as per requirement.
6. The proposed equipment can safely handle the data required to use the new system.

**As per all above point, proposed system is technically feasible.**

### 1.5.2 Economic Feasibility

1. Economic feasibility is a measure of the cost effectiveness of a project or a solution.
2. The cost to implement proposed system includes cost of full systems investigation, cost of hardware and software and the cost of user training.
3. The cost to implement the proposed system is one time investment later they need not spent any more.
4. The organization must get benefit if they invest on the proposed system development.

**As per the above reasons the proposed new system is economically feasible.**

### 1.5.3 Operational Feasibility

1. Operational feasibility is mainly concerned with issue like whether the system will be developed or implemented properly.
2. The proposed system will not cause any problem & show efficient performance.
3. This system reduces the work load & work time and leads to error free & accurate work.
4. The proposed system will reduce the space required to store paper work for the previous records.
5. With the help of this software, we get the information about employees, database & other detail easily.
6. From the beginning of proposed system, we made users to involve in some way reducing the resistance in all respects.

**As per all above point, proposed system is operationally feasible.**

## CHAPTER-2

# PROPOSED SYSTEM

### 2.1 Proposed System:

The proposed system that we are making will be completely different and more importantly it will be user friendly.

It can be operated by anyone .It comes with some up-to-date functions which you can perform by just clicking a mouse button.

The aim of this system is to advance to the next level by making everything computerized. It has advantages like this:

- 1) It's user-friendly and it's completely computerized.
- 2) It has a huge data base which can store lots and lots of information.
- 3) Data can be easily entered, updated. I.e. data can be managed easily.
- 4) No need to worry about the storage of the data as it will be stored in your computer.
- 5) Search results can be shown to the clients at the same place and proper and correct information can be provided.

### 2.2Objective of System:

- ✚ It is very user-friendly and having added more features.
- ✚ It is fully computerized and easy to access.
- ✚ To develop global partnership.
- ✚ Wastage of energy is avoided.
- ✚ Provide security to data.
- ✚ Wastage of time is avoided.
- ✚ Reduce manpower.
- ✚ Decrease manual mistakes.
- ✚ Easy maintenance of Import and Export document.

### 2.3 User Requirement:

#### 2.3.1: Functional Requirements

- software engineering, a functional requirement defines a function of a software system or its component.
- A function is described as a set of inputs, the behavior, and outputs.
- Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.
- Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for

implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

- As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

## **2.3.2: Non-Functional Requirements**

### **Product Requirements**

#### ➤ Usability requirements

Usability is the ease of use and learns ability of a human-made object. The object of use can be a software application, website, book, tool, machine, process, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. Usability includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed. Usability differs from user satisfaction and user experience because usability also considers usefulness.

#### ➤ Reliability requirements

Reliability deals with the study, evaluation, and lifecycle management of reliability: the ability of a system or component to perform its required functions under stated conditions for a specified period of time. Reliability engineering is a sub-discipline within systems engineering. Reliability is theoretically defined as the probability of failure, the frequency of failures, or in terms of availability, a probability derived from reliability and maintainability. Maintainability and maintenance may be defined as a part of reliability engineering. Reliability plays a key role in cost-effectiveness of systems.

#### ➤ Portability requirements

Portability in high-level computer programming is the usability of the same software in different environments. The pre requirement for portability is the generalized abstraction between the application logic and system interfaces.

When software with the same functionality is produced for several computing platforms, portability is the key issue for development cost reduction.

Transferring installed program files to another computer of basically the same architecture. Reinstalling a program from distribution files on another computer of basically the same architecture.

### **Efficiency requirements**

Resource consumption for given load describes efficiency of product and web site.



➤ Performance requirements

Performance metrics include availability, response time, channel capacity, latency, completion time, service time, bandwidth, throughput, relative efficiency, scalability, performance per watt, compression ratio, instruction path length and speed up.

- Short response time for a given piece of work
- High throughput (rate of processing work)
- Low utilization of computing resource(s)
- High availability of the computing system or application
- Fast (or highly compact) data compression and decompression High bandwidth / short data transmission time.

### **Organizational Requirements**

➤ Implementation requirements

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium – recommended specifications, and software development tools contain implementations of programming languages.

➤ Standard requirements

The project should be developed as per standard format specified by IEEE.

Typical platforms include a computer architecture, operating system, programming languages and related user interface. The product should be developed as per client's standard requirements.

### **External Requirements**

➤ Interoperability requirements

Interoperability is a property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation.

The IEEE Glossary defines interoperability as:

The ability of two or more systems or components to exchange information and to use the information that has been exchanged.

➤ Legislative requirements

In the proprietary software industry, an end-user license agreement or software license agreement is the contract between the licensor and purchaser, establishing the purchaser's right to use the software. The license may define ways under which the copy can be used. Software companies often make special agreements with large businesses and government entities that include support contracts and specially drafted warranties.



➤ Privacy requirements

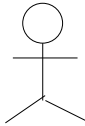


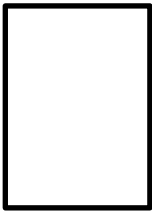
The term "privacy" means many things in different contexts. Different people, cultures, and nations have a wide variety of expectations about how much privacy a person is entitled to or what constitutes an invasion of privacy. Privacy is the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively. The boundaries and content of what is considered private differ among cultures and individuals, but share basic common themes. Privacy is sometimes related to anonymity, the wish to remain unnoticed or unidentified in the public realm.

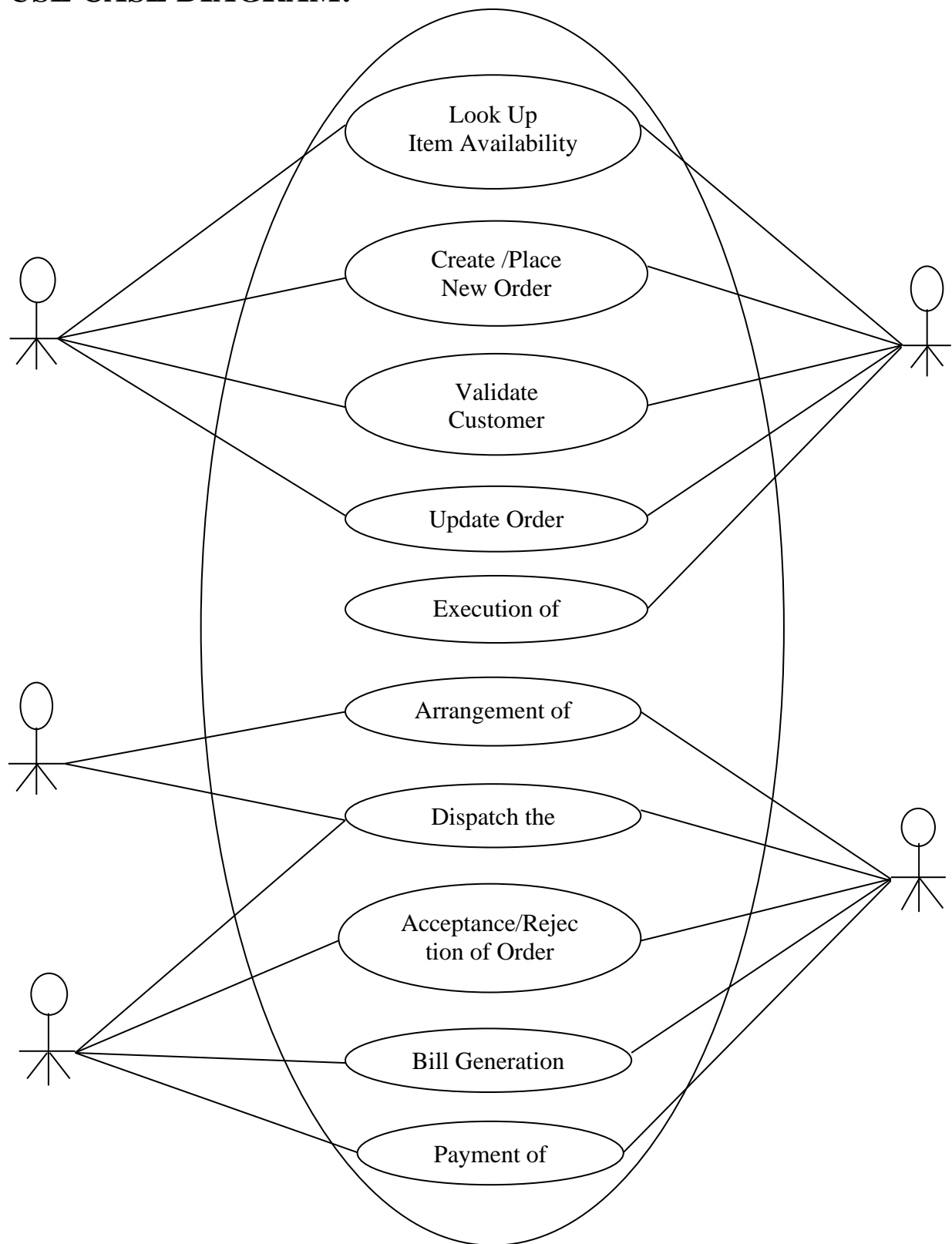
## CHAPTER -3

### ANALYSIS & DESIGN

#### 3.1 USE CASE DIAGRAMS

A use case defines behavioral features of a system. Each use case is named using a verb phase expresses a goal of the system. A use case diagram shows a set of use cases and actors & their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. It shows the graphical overview of functionality provided by the system intends actor.

Symbol	Symbol Name	Represents
	Actor	It represents a role that human, hardware device or another system plays when it communicates with the system.
	Use Case	It is the description of set of sequences of actions. It represents an action performed by a system.
	Communication Link	Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using message.
	System Boundary	For large and complex systems, each module may be the system boundary. The entire system can span all of these modules depicting the overall system boundary.

**USE CASE DIAGRAM:-**

## 3.2 DATA FLOW DIAGRAM

Data flow diagram is used to represent data & processes that manipulate it. The data flow diagram enables the software engineer to develop the model of information domain & functional domain at same time. As the DFD is refined into greater levels of details, the analyst performs implicit functional decomposition of the system.

A data flow Diagram (DFD) is one of the popular graphical tools uses to depict the flow of data through a system. DFD shows the processes, data stores, data flow & the source & destination entries. A few simple guidelines can aid immeasurably during derivation of data flow diagram.

1. The level 0 DFD should depict the system as a single bubble.
2. The primary input & output should be carefully noted.
3. Refinement should being by isolating candidate processes, data object & Stores to be represented at the next level.
4. All arrows & bubbles should be labelled with meaningful names.
5. Information flow continuity must be maintained from level to level. One bubble at time should be refined.

### TYPES OF DFD's:

There are two types of DFD's as follows: -

- 1). Physical DFD's
- 2). Logical DFD's

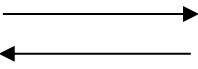

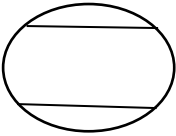
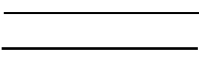
1). Physical DFD's: Physical DFD's depict the physical elements like people, report, documents, departments etc. Physical DFD's shows an implementation dependent view of the system.

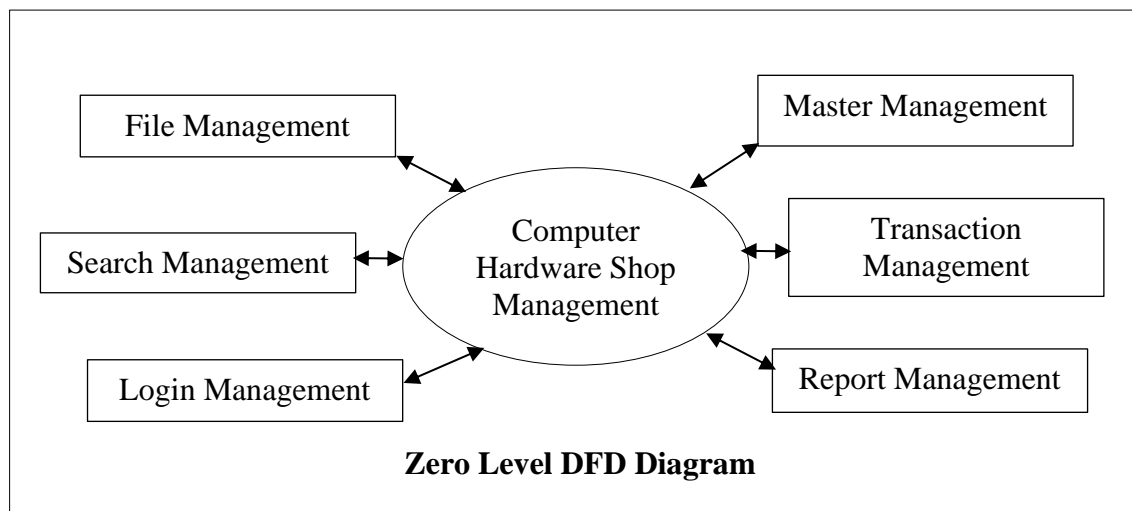
2). Logical DFD's: Logical DFD's depict the logical elements like data process & events those are abstract than physical DFD's. Logical DFD's shows an implementation independent view of the system.

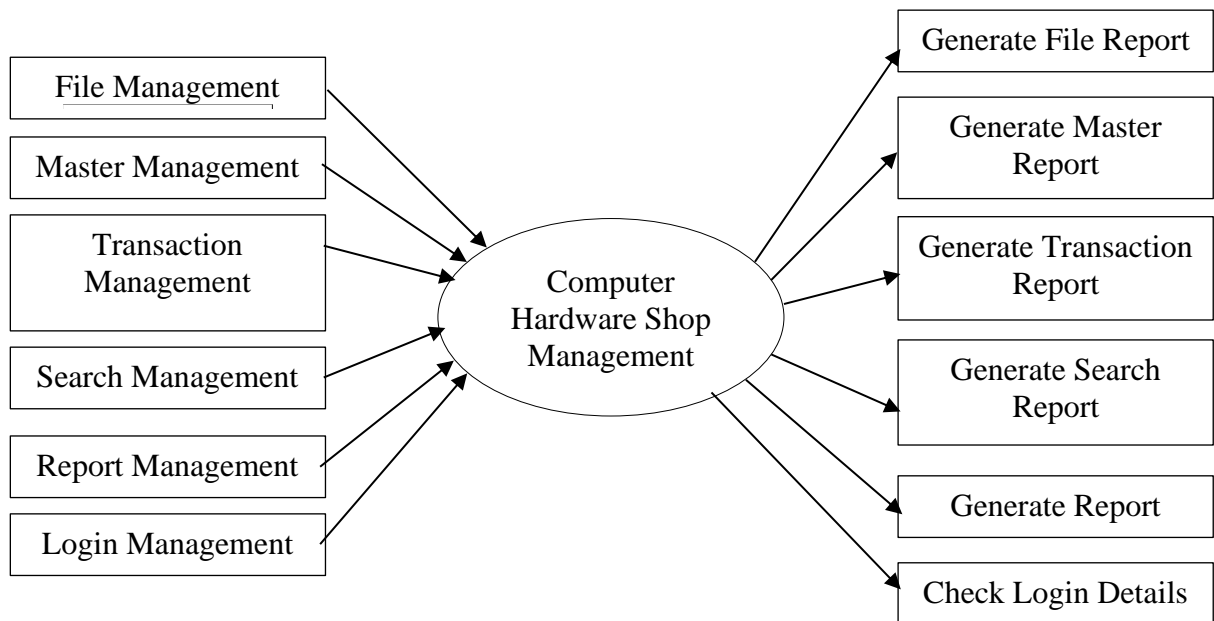
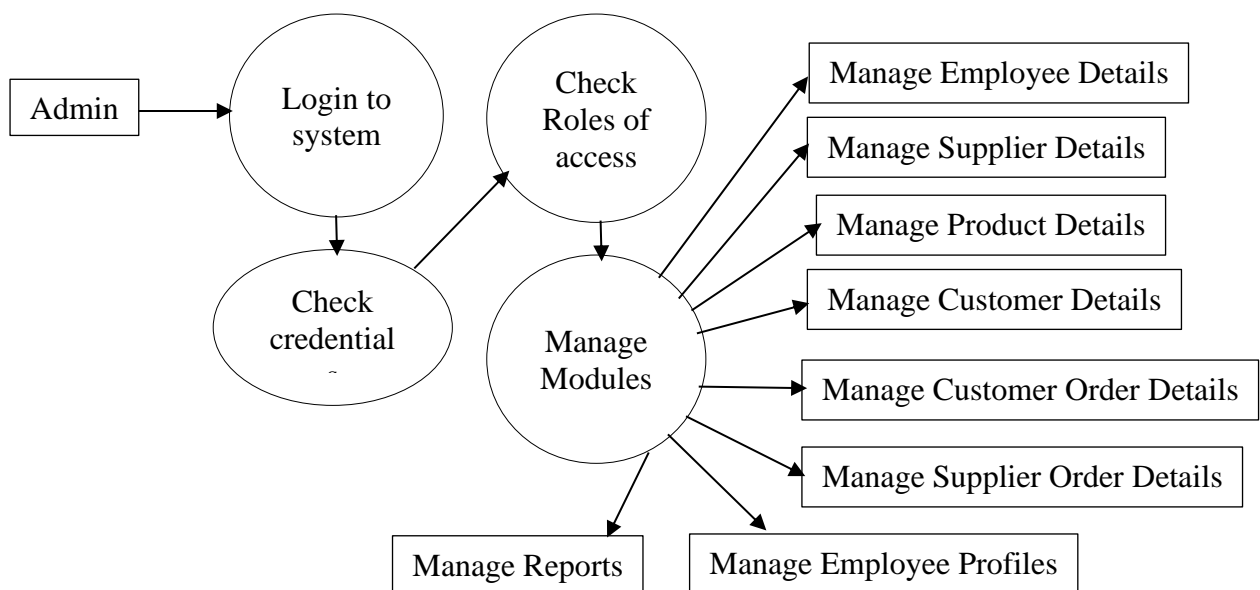
### CONTEXT DIAGRAM:

The top-level diagram is called as a 'context diagram'. Context diagrams contain single process, but it plays a very important role in studying the current system. Context diagram is constructed to show the highest level model of the system. This is the most general or broadcast picture of the current system. They are too representing the scope or boundaries of the system. Their purpose is identifying what is to include in the system under study.

**1). Symbol Used for Data Flow Diagram:**

Symbol	Symbol Name	Represents
	External Entity	A Source or destination of data which is external system.
	Data Flow	It is a packet of data. It may be in the form of document, letter, telephone call etc.
	Process	Here flow of data transferred.
	Data Store	Any store data but with no difference to physical method of storing.

**LEVEL 0:**

**LEVEL 1:****First Level DFD Diagram****LEVEL 2:****Second Level DFD Diagram**

### 3.3 ENTITY RELATIONSHIP DIAGRAMS

#### E-R DIAGRAM

##### SYMBOL USED IN E-R DIAGRAM:


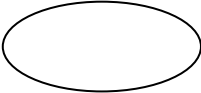
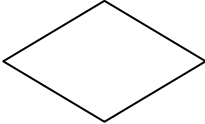

The E-R model uses few basic concepts in producing an E-R diagram.

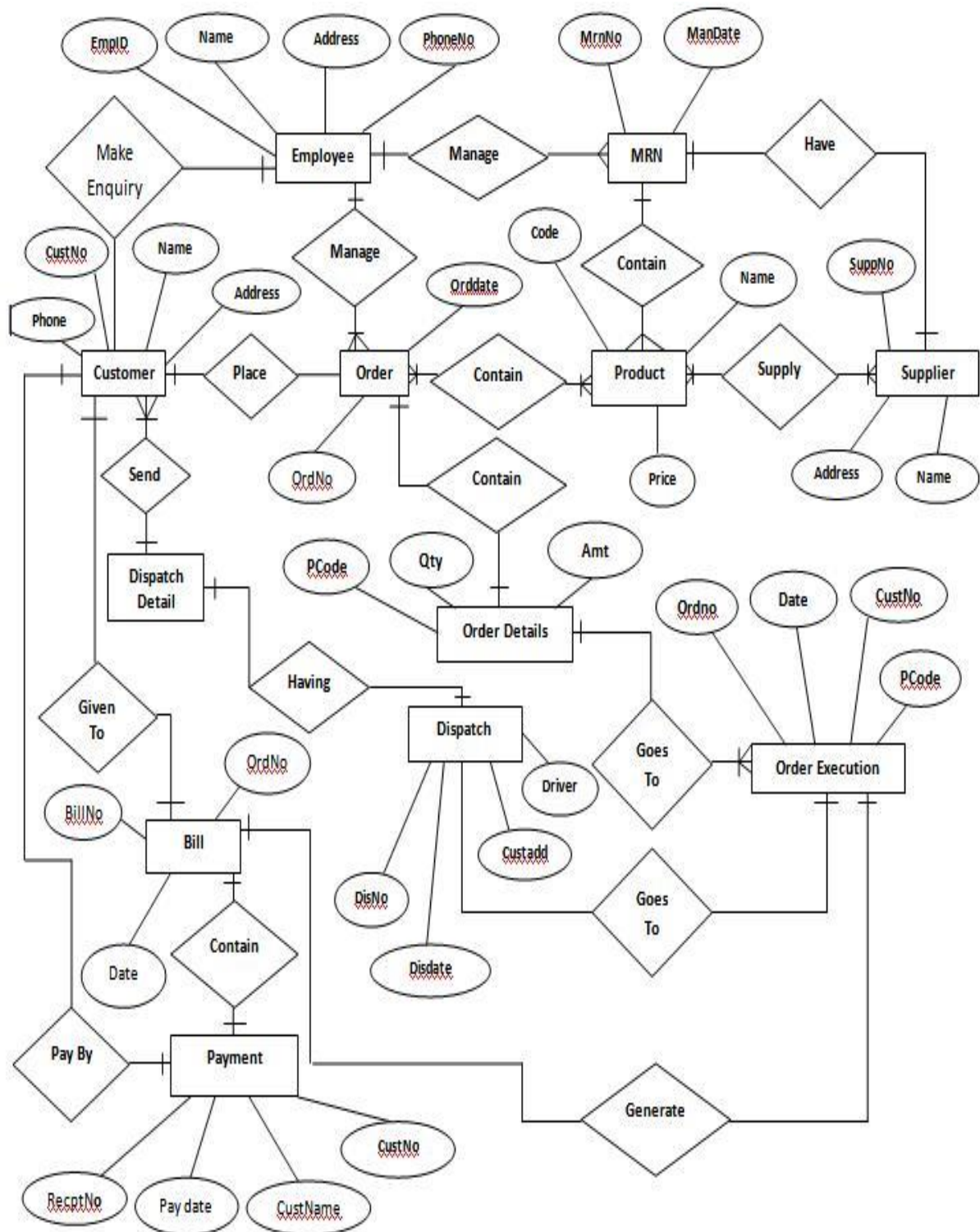
These concepts are: -

- 1) Entity
- 2) Relationship
- 3) Attribute

- 1). Entity: An entity is an object or anything, which is distinguishable from objects.
- 2). Relationship: A relationship is meaningful association, a linking or connection between entities.
- 3). Attribute: An attribute is any aspect quality or description of either an entity or relationship.

##### SYMBOL FOR E-R DIAGRAM:

Symbols	Symbol Name	Represents
	Rectangle	Entity Set
	Ellipse	Attribute
	Diamond	Relationship Set
	Line	Links between two Entity Set

**E-R DIAGRAM :-**




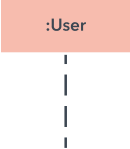



### 3.4 SEQUENCE DIAGRAMS

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.





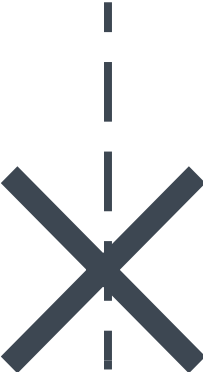
#### Basic symbols and components

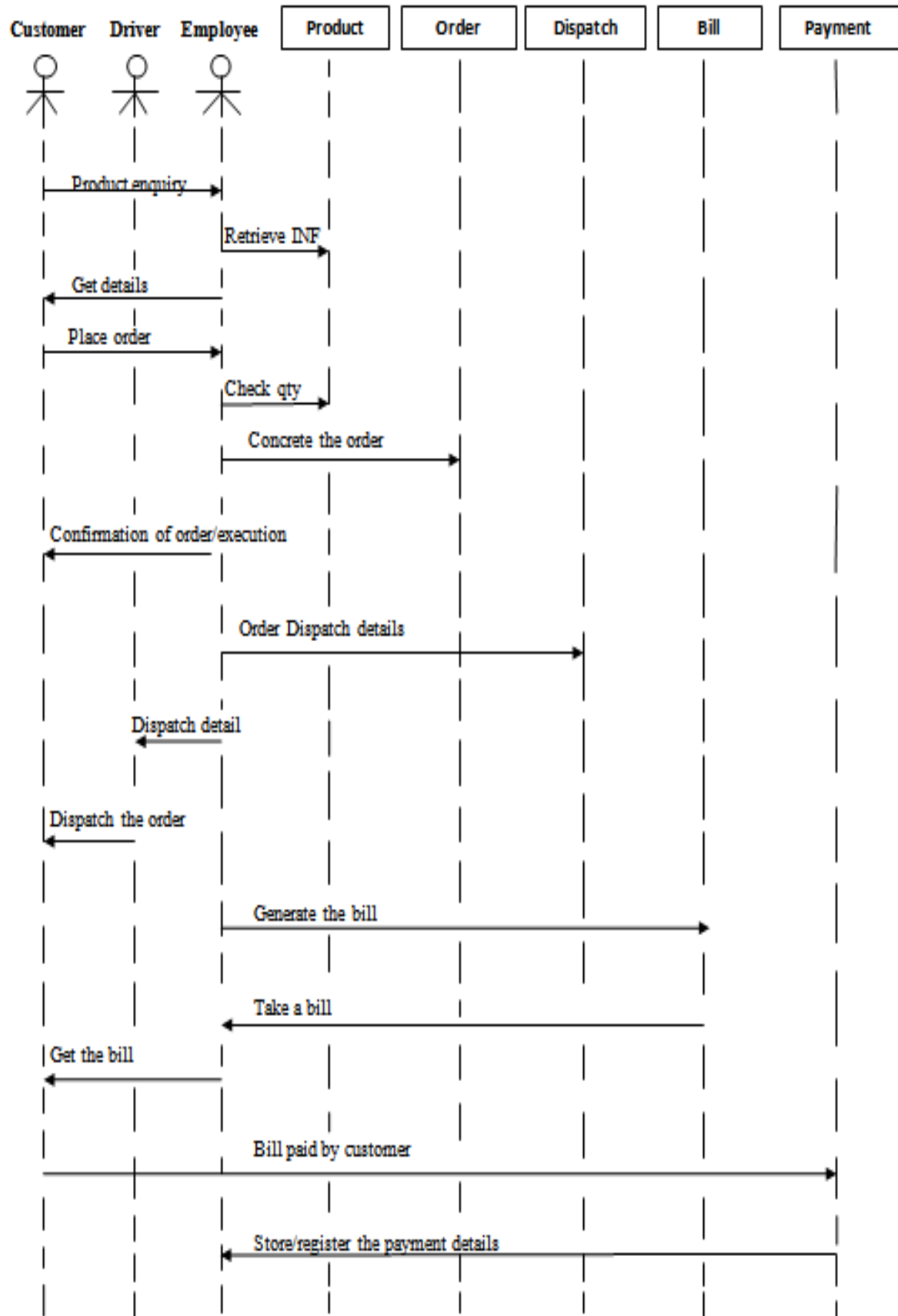
To understand what a sequence diagram is, you should be familiar with its symbols and components. Sequence diagrams are made up of the following icons and elements:

SYMBOL	NAME	DISCRIPTION
	Object symbol	Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape.
	Activation box	Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.
	Actor symbol	Shows entities that interact with or are external to the system.
	Lifeline symbol	Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.
	Alternative symbol	Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside.

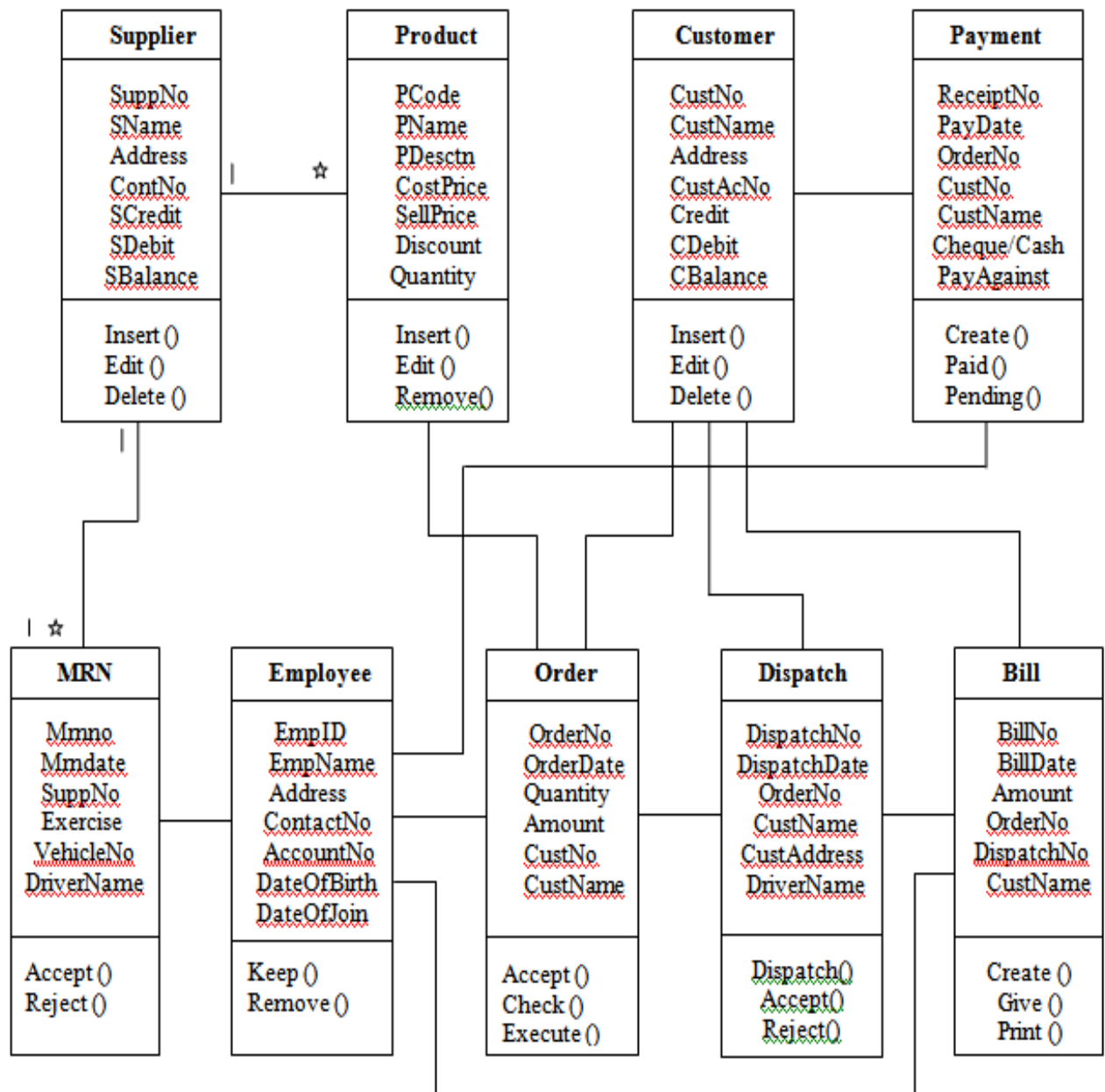
**Common message symbols**

Use the following arrows and message symbols to show how information is transmitted between objects. These symbols may reflect the start and execution of an operation or the sending and reception of a signal.

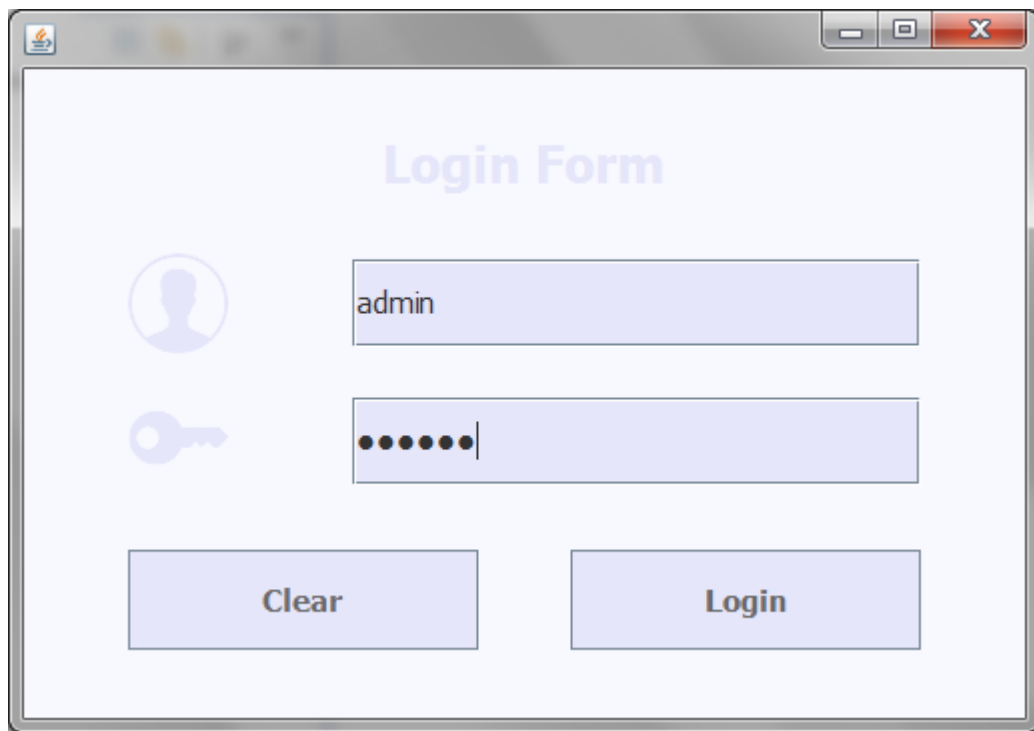
SYMBOL	NAME	DISCREPTION
	Synchronous message symbol	Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.
	Asynchronous message symbol	Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.
	Asynchronous return message symbol	Represented by a dashed line with a lined arrowhead.
	Asynchronous create message symbol	Represented by a dashed line with a lined arrowhead. This message creates a new object.
	Delete message symbol	Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

**SEQUENCE DIAGRAM**

### 3.5 CLASS DIAGRAM

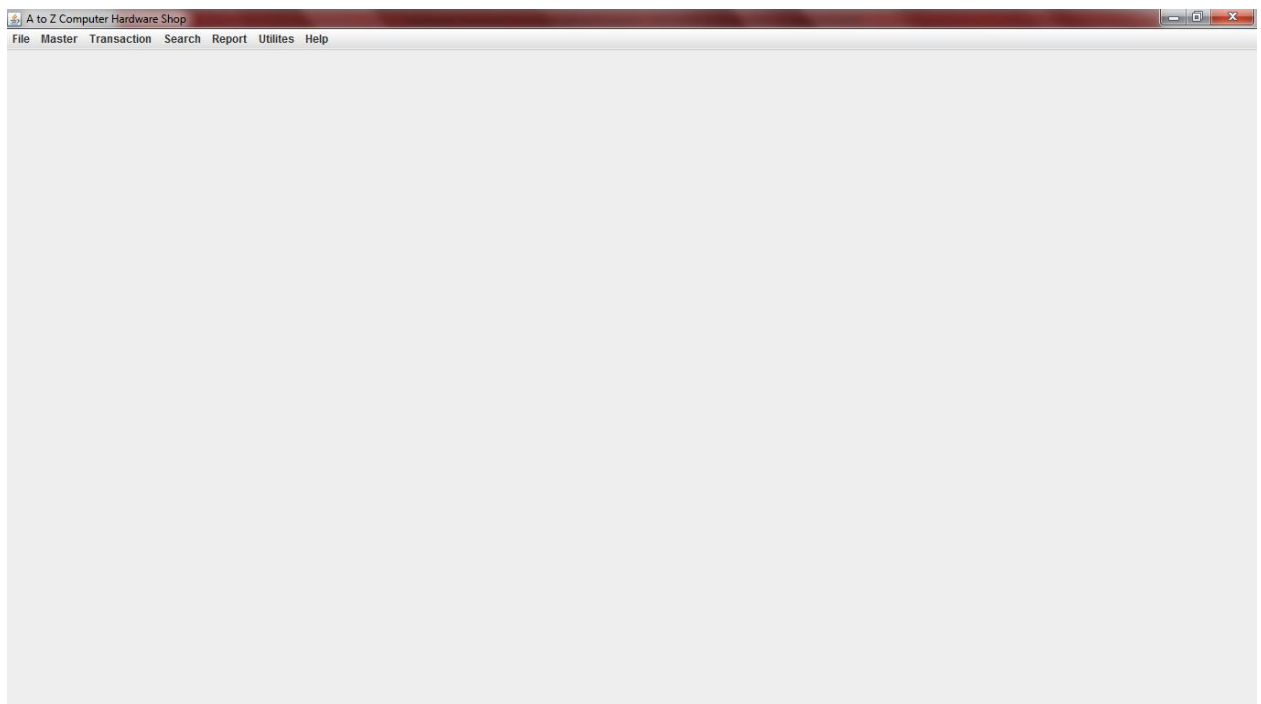


### 3.6 USER INTERFACE DESIGN (Screens etc.)



A screenshot of a login form window. The window has a title bar with standard minimize, maximize, and close buttons. The main content area is light blue and contains the text "Login Form" at the top. Below this, there are two input fields. The first field is preceded by a user icon and contains the text "admin". The second field is preceded by a key icon and contains a series of dots, indicating a password. At the bottom of the form, there are two buttons: "Clear" and "Login".

#### LOGIN FORM LAYOUT



#### MENU SCREEN

The screenshot shows a window titled "Adding\_Employee\_Information". It contains two main sections: "Employee\_Personal\_Information" and "Employee\_Official\_Information".

**Employee\_Personal\_Information:**

- Employee ID : 14
- Employee Name : Juber
- Employee Birth Date : Mar 1, 1996
- Employee Gender : ☒ Male ☐ Female
- Employee Address : pen
- Employee Contact : 777777777
- Employee Email ID : juber@gmail.com

**Employee\_Official\_Information:**

- Employee Joining Date : Mar 1, 2017
- Employee Designation : CEO
- Employee User Name : juber
- Employee Password : juber
- Employee Confirm Password : 10000

A "Message" dialog box is displayed in the center, stating "Employee Data Save Sucesfull..." with an "OK" button.

**Operation:**

- Save
- Clear
- Ext

## ADD EMPLOYEE

The screenshot shows a window titled "Searching\_Customer\_Information". It contains two main sections: "Customer\_Search" and "Customer\_Official\_Information".

**Customer\_Search:**

- Customer ID : 6
- Customer Name : raghav

**Customer\_Personal\_Information:**

- Customer Registered Date : Feb 7, 2017
- Customer Gender : ☒ Male ☐ Female
- Customer Address : parvel
- Customer Contact : 5668778899
- Customer Email ID : faa

**Customer\_Official\_Information:**

- Customer Credit Amount : 212
- Customer Debit Amount : 12
- Customer Balance Amount : 200

A "Message" dialog box is displayed in the center, stating "Update Customer Data Sucesfull..." with an "OK" button.

**Operation:**

- Update
- Clear
- Delete
- Exit

## MODIFY CUSTOMER

The screenshot shows a window titled "Searching\_Supplier\_Information". It contains two main sections: "Supplier\_Search" and "Supplier\_Official\_Information".

**Supplier\_Search:**

- Supplier Id : 2
- Supplier Name : ashok

**Supplier\_Personal\_Information:**

- Supplier Registered Date : Mar 7, 2017
- Supplier Gender : ☒ Male ☐ Female
- Supplier Address : parvel
- Supplier Contact : 7878787878
- Supplier Email ID : asd

**Supplier\_Official\_Information:**

- Supplier Credit Amount : 1000
- Supplier Debit Amount : 1000
- Supplier Balance Amount : 1000

A "Message" dialog box is displayed in the center, stating "Supplier Data delete Sucessfully..." with an "OK" button.

**Operation:**

- Update
- Clear
- Delete
- Exit

## MODIFY SUPPLIER

Customer\_Order\_Report

**Customer Order Report** Print

id	date	cust_id	cust_name	cust_cont...	cust_bam...	pro_id	pro_mno	pro_quntity	pro_rate	tot_amount	address
1	2017-02-...	1	Hrushu	8.888888...	10000.0	2	a12	12	100.0	1200.0	panvel
2	2017-02-...	5	reshma	6.665554...	99.0	2	a12	1	100.0	100.0	pune
3	2017-02-...	8	rekha	1.213232...	10.0	2	a12	4	100.0	400.0	12
4	2017-02-...	2	Rupesh	6.666666...	2000.0	3	mi89	1	12000.0	12000.0	panvel
5	2017-02-...	14	aniket	9.808989...	10000.0	2	a12	7	100.0	700.0	panvel
6	2017-03-...	16	ram	1.111111...	12.0	4	mh54	5	10000.0	50000.0	as
7	2017-03-...	10	karan	1.231231...	101.0	5	hp123	2	12000.0	24000.0	kalmoli

### VIEW CUSTOMER ORDER REPORT

Supplier\_Order\_Report

**Supplier Order Report** Print

id	date	sup_id	sup_name	pro_id	pro_mn	pro_qn	pro_rate	tot_amount
1	2017-03-02 0...	2	durga	3	mi89	4	12000.0	48000.0
2	2017-03-07 0...	1	durga	2	a12	19	100.0	1900.0
3	2017-03-02 0...	2	ashok	2	a12	1	100.0	100.0
5	2017-03-02 0...	2	ashok	3	mi89	10	12000.0	120000.0
6	2017-03-10 0...	5	deva	3	mi89	2	12000.0	24000.0
7	2017-03-15 0...	6	manoj	3	mi89	1	12000.0	12000.0

### SUPPLIER ORDER REPORT

### 3.7 TABLE SPECIFICATIONS

#### Table Name: Login

Field Name	Data Type	Description
Username	VARCHAR	It stores the Username
Password	VARCHAR	It stores the Password

#### Table Name: Supplier master

Field Name	Data Type	Description
Supplier INTEGER	VARCHAR	It stores the unique Supplier INTEGER.
Name	VARCHAR	It stores the unique Supplier Name.
Address	VARCHAR	It stores the unique Supplier Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of supplier.
Account INTEGER	VARCHAR	It stores the supplier bank account INTEGER.
Credit	INTEGER	It stores the supplier credit details.
Debit	INTEGER	It stores the supplier debit details.
Balance	INTEGER	It stores the supplier balance details.

#### Table Name: Employee master

Field Name	Data Type	Description
Employee ID	VARCHAR	It stores the unique employee INTEGER.
Name	VARCHAR	It stores the unique employee Name.
Address	VARCHAR	It stores the unique employee Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of employee
Date of Birth	DATETIME	It stores the date of birth of an employee.
Date of Join	DATETIME	It stores the date of joining of an employee.
Email ID	VARCHAR	It stores the Email-id of employee.
Salary	INTEGER	It stores the salary of employee.



**Table Name: Product master**

Field Name	Data Type	Description
Product Code	VARCHAR	It stores the unique product code.
Name	VARCHAR	It stores the unique product name.
Description	VARCHAR	It stores the product description.
Cost Price	INTEGER	It stores the product cost price.
Sell Price	INTEGER	It stores the product selling price.
Discount	INTEGER	It stores the product limited discount.
Quantity	INTEGER	It stores the product quantity.
Reorder Level	INTEGER	It stores the reorder level.

**Table Name: Customer master**

Field Name	Data Type	Description
Customer INTEGER	VARCHAR	It stores the unique customer INTEGER.
Name	VARCHAR	It stores the unique customer Name.
Address	VARCHAR	It stores the unique customer Address.
Phone INTEGER	INTEGER	It stores the contact INTEGER of customer.
Account INTEGER	VARCHAR	It stores customer bank account INTEGER.
Credit	INTEGER	It stores the customer credit details.
Debit	INTEGER	It stores the customer debit details.
Balance	INTEGER	It stores the customer balance details.

**Table Name: MRN master**

Field Name	Data Type	Description
MRN INTEGER	VARCHAR	It stores the unique MRN INTEGER.
Date	DATETIME	It stores the date.
Supplier INTEGER	VARCHAR	It retrieve the supplier INTEGER.
Product Excise	VARCHAR	It stores product excise code.
Vehicle INTEGER	VARCHAR	It stores the vehicle INTEGER.
Driver Name	VARCHAR	It stores the driver name.

**Table Name: MRN Transaction**

Field Name	Data Type	Description
MRN INTEGER	VARCHAR	It stores the unique MRN INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Quantity	INTEGER	It stores the product quantity.
Received Quantity	INTEGER	It stores the product received quantity.
Rejected Quantity	INTEGER	It stores the product rejected quantity.
Product Rate	INTEGER	It stores the single product rate.
Product Amount	INTEGER	It stores the total product amount.

**Table Name: Order master**

Field Name	Data Type	Description
Order INTEGER	VARCHAR	It stores the unique order INTEGER.
Order Date	DATETIME	It stores the order date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It retrieve the customer Name.

**Table Name: Order Transaction**

Field Name	Data Type	Description
Order INTEGER	VARCHAR	It retrieve order INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Name	INTEGER	It stores the product name.
Product Description	INTEGER	It stores the product description.
Product Quantity	INTEGER	It stores the product quantity.
Product Rate	INTEGER	It stores the single product rate.
Order Billed	Yes/No	It stores record from order.

**Table Name: Dispatch Master**

Field Name	Data Type	Description
Dispatch INTEGER	VARCHAR	It stores the unique dispatch INTEGER.
Dispatch Date	DATETIME	It stores the dispatch date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Address	VARCHAR	It retrieve the customer Name.
Order INTEGER	VARCHAR	It retrieve order INTEGER.
Vehicle INTEGER	VARCHAR	It stores the vehicle INTEGER.
Driver Name	VARCHAR	It stores the driver name.

**Table Name: Dispatch Transaction**

Field Name	Data Type	Description
Dispatch INTEGER	VARCHAR	It stores the unique dispatch INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Quantity	INTEGER	It stores the product quantity.
Received Quantity	INTEGER	It stores the product received quantity.
Rejected Quantity	INTEGER	It stores the product rejected quantity.
Product Rate	INTEGER	It stores the product rate.

**Table Name: Bill Master**

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.
Bill Date	DATETIME	It stores the bill generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It stores the customer name.
Order INTEGER	VARCHAR	It retrieve the order number.
Dispatch INTEGER	VARCHAR	It retrieve the dispatch INTEGER.

**Table Name: Bill Transaction**

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.
Product Code	VARCHAR	It retrieve the product code.
Product Description	INTEGER	It stores the product description.
Received Quantity	INTEGER	It stores the product received quantity.
Product Rate	INTEGER	It stores the product rate.
Product Discount	INTEGER	It stores the product limited discount.
Product Amount	INTEGER	It stores the total product amount.

**Table Name: Product Amount**

Field Name	Data Type	Description
Bill INTEGER	VARCHAR	It stores the unique bill INTEGER.
Bill Date	DATETIME	It stores the bill generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Order INTEGER	VARCHAR	It retrieve the order number.
Dispatch INTEGER	VARCHAR	It retrieve the dispatch INTEGER.
Product Amount	INTEGER	It stores the total product amount.

**Table Name: Payment Receipt**

Field Name	Data Type	Description
Receipt INTEGER	VARCHAR	It stores the unique receipt INTEGER.
Payment Date	DATETIME	It stores the payment generate date.
Customer INTEGER	VARCHAR	It retrieve the customer INTEGER.
Customer Name	VARCHAR	It stores the customer name.
Payment Against	VARCHAR	It check the payment against.
Bill Amount	INTEGER	It retrieve the payment bill amount.
Payment Mode	VARCHAR	It's type of payment mode.
Cheque INTEGER	INTEGER	It stores the cheque INTEGER.
Bank Name	VARCHAR	It stores the bank name.
Cheque Amount	INTEGER	It stores the cheque amount.

## 3.8 Testing And Implementation

**Software testing** is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect-free in order to produce a quality product.

### 3.8.1 Testing Approaches

There are three types of software testing approaches.

1. White Box Testing
2. Black Box Testing

#### **3.8.1.1 White Box Testing**

It is also called Glass Box, Clear Box, Structural Testing. White Box Testing is based on the application's internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.

#### **3.8.1.2 Black Box Testing**

It is also called Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

### 3.8.2 Unit Testing

Unit Testing is done to check whether the individual modules of the source code are working properly. i.e. testing each and every unit of the application separately by the developer in the developer's environment. It is AKA Module Testing or Component Testing.

### 3.8.3 Integration Testing

Integration Testing is the process of testing the connectivity or data transfer between a couple of unit tested modules. It is AKA I&T Testing or String Testing. It is subdivided into the Top-Down Approach, Bottom-Up Approach, and Sandwich Approach (Combination of Top-Down and Bottom-Up).

### 3.8.4 System Testing (End to End Testing)

It's a black box testing. Testing the fully integrated application is also called as an end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experiences with the application.

### 3.8.5 Acceptance Testing

To obtain customer sign-off so that software can be delivered and payments received. Types of Acceptance Testing are Alpha, Beta & Gamma Testing.

## **IMPLEMENTATION**

**The steps for making Class file of the project is as follows: -**

- Open your project.
- Compile the Java file.
- Then class file will be created.
- Execute that Java Class file.

## CHAPTER -4

### USER MANUAL

#### 4.1 USER MANUAL

Although the user interface of the system is constructed in such a way that anyone can use the system if he has the basic knowledge of the operating keyboard and mouse operation of the computer. All pages of the application contain the descriptive links and the buttons that will help the user to perform the required operation.

There are following links/module:

❖ ADMIN:

- ✚ Admin will have the full authority of the software.
- ✚ Admin will login by using his account.
- ✚ Admin will provide facility to view admin panel.
- ✚ Admin will have to make account of user and edit it.
- ✚ Admin will view/Edit the details.

#### 4.2 Operational Manual / Menu Explanation:

❖ Login:

This form is used to login the user in Computer Hardware Shop Management System. For this user has to enter with the proper id, password user get Menu Form otherwise he will get error message as “Username And Password is incorrect...!!!”.

❖ Employee Registration:

When new employee does not have account then through this form he can register himself to the system. The employee has to just do one thing that fill all details on employee registration form. Also every entry should be unique so that the problem of duplication is avoided and there no employee with same details in the database.

❖ Change Password:

When employee wants to change his password then he is transferred to this page. Here he has to enter user id and has to enter old password then new password then again enter new password for confirmations then click “SAVE BUTTON” then you get message “Password change successfully...”

❖ Employee Registration Form:

This Form contains the Registration details.

Step1: Click on Master →Add\_Employee now BUTTON.

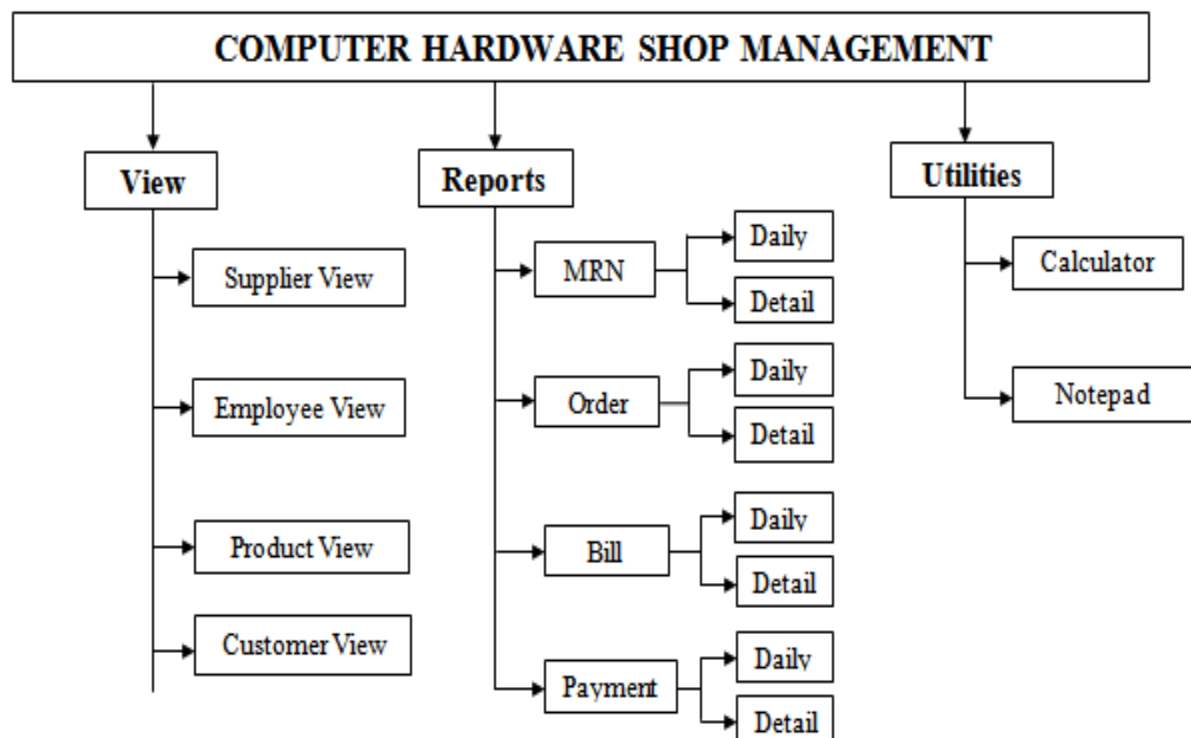
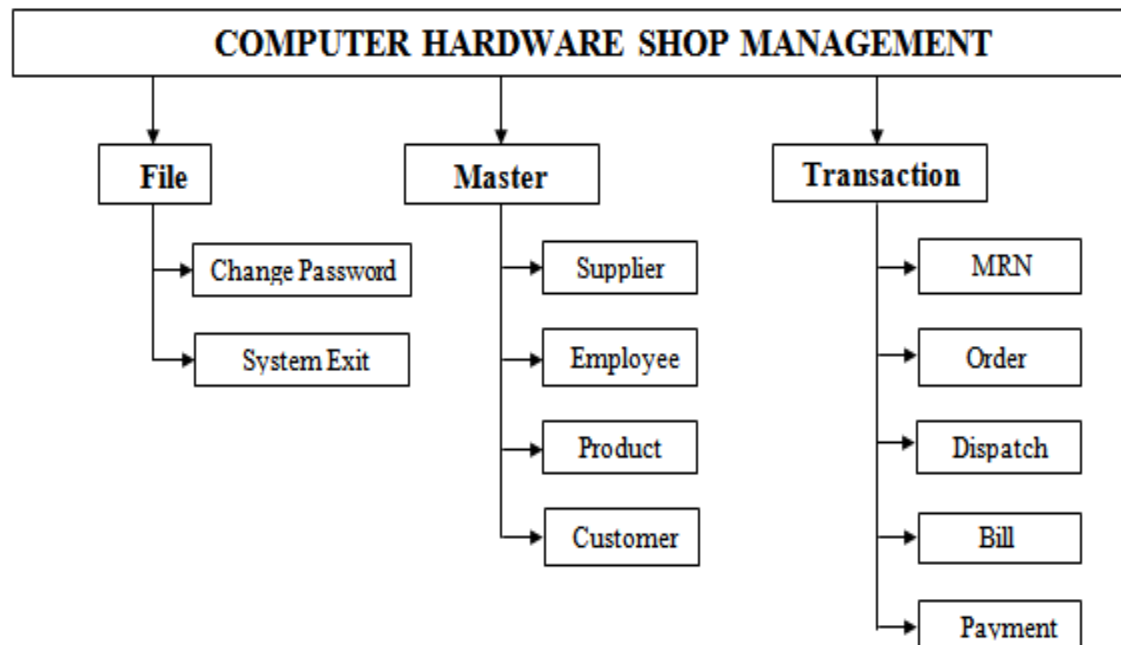
Step2: Enter all details as per the form fields.

Step3: You can see the user registration form in tab format i.e. Personal information, Mail ID.

Step4: Here you can continuously fill all tabs then click on “Submit Button” For Submit form

Step5: Reset Button for clear all fields of form...

❖ Menu Tree:



## **CHAPTER -5**

# **LIMITATIONS & FUTURE ENHANCEMENTS OF THE SYSTEM**

### **LIMITATIONS:**

It has following limitations:

- Data has to be stored manually in files and registers.
- Retrieval of this data is very slow and hectic because one has to search all the files manually.
- As the INTEGER of files increases, it takes up lot of space making it difficult to manage.
- Searching for old bills and files or registers takes a lot of time.
- Data loss is always a major concern as a file or a register can be easily misplaced.
- Natural calamity like floods, fire break-out and other such things can destroy the data, and it can lead to data loss.

### **FUTURE ENHANCEMENT:**

In my future version of this system I would like to add some following features in my project:

1. Some more reports will be added to make administrator task easier.
2. Interface will be made more user friendly.

Being a computer system, the system has lots of scope. It not only carried out work faster but also efficiently. A lots of manual work like searching ,personal alias is reduced to a large extent. As this is computerized, system the manager is completely depends on computer for accessing details about students. Here in case there is power failure or some hardware problem which cannot be fixed easily, the manager cannot continue this work. He gets struck in the middle of the day and this dealing need to be postponed.

Also the software need to be maintained properly from time to time, that is if it requires being updated or modified etc. then the manager need to take care of it.



## CHAPTER -6

# CONCLUSION

### 6.1 Conclusion of the COMPUTER HARDWARE SHOP MANAGEMENT SYSTEM

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objectives of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

### 6.2 At the end it is concluded that we have made effort on following points...

- ❖ A description of the background and context of the project and its relation to work already done in the area.
  - ❖ Made statement of the aims and objectives of the project.
  - ❖ The description of Purpose, Scope, and applicability.
  - ❖ We define the problem on which we are working in the project.
  - ❖ We describe the requirement Specifications of the system and the actions that can be done these things.
  - ❖ We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
  - ❖ We included features and operations in detail, including screen layouts.
  - ❖ We designed user interface and security issues related to system.
- Finally the system is implemented and tested according to test cases.

## CHAPTER -7

### SYSTEM CODING

- **Program Name:-** Login.java

**Program Description:-** This form is used for security with the help of password.

**Code:-**

```
package com.asms.form;
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import com.asms.form.db.DBLoginWindow;
import java.awt.event.*;

@SuppressWarnings("serial")
public class LoginWindow extends JFrame
{
    private JPanel contentPane;
    private JTextField txtLoginname;
    private JPasswordField txtPassword;
    private String name, pass;

    public LoginWindow()
    {
        setBounds(100, 100, 516, 362);
        //setAlwaysOnTop(true);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(0, 128, 0));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JPanel panel = new JPanel();
        panel.setLayout(null);
        panel.setBackground(null);
        panel.setBounds(10, 11, 480, 306);
        getContentPane().add(panel);

        txtLoginname = new JTextField();
        txtLoginname.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtLoginname.setColumns(10);
        txtLoginname.setBackground(new Color(230, 230, 250));
        txtLoginname.setBounds(196, 86, 227, 34);
        panel.add(txtLoginname);

        JButton btnLogin = new JButton("Login");
        btnLogin.addActionListener(new ActionListener()
        {
            @SuppressWarnings({ "deprecation" })
            public void actionPerformed(ActionEvent arg0)
```

```
        {
            name=txtLoginname.getText();
            pass=txtPassword.getText();
            DBLoginWindow db=new DBLoginWindow();
            int i = db.checkLogin(name,pass);
            if (i==0)
                setVisible(false);
        }
    });
    btnLogin.setForeground(SystemColor.controlDkShadow);
    btnLogin.setFont(new Font("Tahoma", Font.BOLD, 15));
    btnLogin.setBackground(new Color(230, 230, 250));
    btnLogin.setBounds(263, 229, 175, 50);
    panel.add(btnLogin);

    txtPassword = new JPasswordField();
    txtPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
    txtPassword.setBackground(new Color(230, 230, 250));
    txtPassword.setBounds(196, 148, 227, 34);
    panel.add(txtPassword);

    JLabel label = new JLabel("Login Form");
    label.setForeground(Color.DARK_GRAY);
    label.setFont(new Font("Tahoma", Font.BOLD, 25));
    label.setBackground(Color.BLACK);
    label.setBounds(169, 11, 141, 50);
    panel.add(label);

    JButton btnClear = new JButton("Clear");
    btnClear.setForeground(SystemColor.controlDkShadow);
    btnClear.setFont(new Font("Tahoma", Font.BOLD, 15));
    btnClear.setBackground(new Color(230, 230, 250));
    btnClear.setBounds(42, 229, 175, 50);
    panel.add(btnClear);

    JLabel lblUsername = new JLabel("User Name :");
    lblUsername.setForeground(Color.DARK_GRAY);
    lblUsername.setFont(new Font("Times New Roman", Font.BOLD, 22));
    lblUsername.setBackground(Color.BLACK);
    lblUsername.setBounds(45, 78, 141, 50);
    panel.add(lblUsername);

    JLabel lblPassword = new JLabel("Password :");
    lblPassword.setForeground(Color.DARK_GRAY);
    lblPassword.setFont(new Font("Times New Roman", Font.BOLD, 22));
    lblPassword.setBackground(Color.BLACK);
    lblPassword.setBounds(42, 140, 141, 50);
    panel.add(lblPassword);
    btnClear.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
```

```

        {
            txtLoginname.setText(null);
            txtPassword.setText(null);
        }
    });
}

```

- **Program Name:- Menuscreen.java**

**Program Description:-** This is the multiple documentation interfaces for the all of the form in the project

**Code:-**

```

package com.chsm.form;
import java.awt.event.*;
import javax.swing.*;
import com.chsm.form.modify.*;
import com.chsm.form.report.*;
import com.chsm.form.transaction.*;

@SuppressWarnings("serial")
public class MenuScreenForm extends JFrame
{
    private JMenuBar menuBar;
    private JMenu mnFile, mnMaster, mnNewMenu, mnSearch, mnReport, mnUtilites,
mnHelp;
    private JMenuItem mntmPasswordChange, mntmLogoutUser, mntmSystemExit,
mntmAddEmployee, mntmAddSupplier, mntmAddProduct, mntmAddCustomer,
mntmSupplierOrder,
        mntmCustomerOrder, mntmDispatchProduct, mntmSearchSupplier,
mntmSearchCustomer, mntmSearchCustomerOrder, mntmSearchEmployee,
        mntmSearchSupplierOrder, mntmEmployeeReport,
mntmCustomerReport, mntmSupplierReport, mntmProductReport,
mntmCustOrderReport, mntmNotepad,
        mntmCalculator, mntmSystemInfo, mntmContactUs;
    private JSeparator s1, s2;
    private JMenuItem mntmEmployeepayment;
    private JMenuItem mntmEmployeepayment
    public MenuScreenForm()
    {
        setTitle("A to Z Computer Hardware Shop");
        setBounds(100, 100, 816, 632);
        setExtendedState(JFrame.MAXIMIZED_BOTH);

        menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        mnFile = new JMenu("File");
        menuBar.add(mnFile);

        mntmPasswordChange = new JMenuItem("Password Change");

```

```
mntmPasswordChange.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new PasswordChange().setVisible(true);
    }
});
mnFile.add(mntmPasswordChange);

mntmLogoutUser = new JMenuItem("Logout User");
mntmLogoutUser.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        setVisible(false);
        new LoginWindow().setVisible(true);
    }
});
mnFile.add(mntmLogoutUser);

s1 = new JSeparator();
mnFile.add(s1);

mntmSystemExit = new JMenuItem("System Exit");
mntmSystemExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
    }
});
mnFile.add(mntmSystemExit);

mnMaster = new JMenu("Master");
menuBar.add(mnMaster);

mntmAddEmployee = new JMenuItem("Add_Employee");
mntmAddEmployee.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddEmployee().setVisible(true);
    }
});
mnMaster.add(mntmAddEmployee);

mntmAddSupplier = new JMenuItem("Add_Supplier");
mntmAddSupplier.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddSupplier().setVisible(true);
    }
});
```

```
});
mnMaster.add(mntmAddSupplier);

mntmAddProduct = new JMenuItem("Add_Product");
mntmAddProduct.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddProduct().setVisible(true);
    }
});
mnMaster.add(mntmAddProduct);

mntmAddCustomer = new JMenuItem("Add_Customer");
mntmAddCustomer.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new AddCustomer().setVisible(true);
    }
});
mnMaster.add(mntmAddCustomer);

mnNewMenu = new JMenu("Transaction");
menuBar.add(mnNewMenu);

mntmSupplierOrder = new JMenuItem("Supplier_Order");
mntmSupplierOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new SupplierOrder().setVisible(true);
    }
});
mnNewMenu.add(mntmSupplierOrder);
mntmCustomerOrder = new JMenuItem("Customer_Order");
mntmCustomerOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new CustomerOrder().setVisible(true);
    }
});
mnNewMenu.add(mntmCustomerOrder);

mntmDispatchProduct = new JMenuItem("Dispatch_Product");
mntmDispatchProduct.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new DispatchCustomerOrder().setVisible(true);
    }
});
```

```
        }
    });
    mnNewMenu.add(mntmDispatchProduct);

    mntmEmployeepayment = new JMenuItem("Employee_Payment");
    mntmEmployeepayment.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            new EmployeePayment().setVisible(true);
        }
    });
    mnNewMenu.add(mntmEmployeepayment);
    mnSearch = new JMenu("Search");
    menuBar.add(mnSearch);

    mntmSearchEmployee = new JMenuItem("Search_Employee");
    mntmSearchEmployee.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            new ModifyEmployee().setVisible(true);
        }
    });
    mnSearch.add(mntmSearchEmployee);

    mntmSearchSupplier = new JMenuItem("Search_Supplier");
    mntmSearchSupplier.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            new ModifySupplier().setVisible(true);
        }
    });
    mnSearch.add(mntmSearchSupplier);

    mntmSearchCustomer = new JMenuItem("Search_Customer");
    mntmSearchCustomer.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            new ModifyCustomer().setVisible(true);
        }
    });
    mnSearch.add(mntmSearchCustomer);

    mntmSearchCustomerOrder = new JMenuItem("Search_Customer_Order");
    mntmSearchCustomerOrder.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent arg0)
        {
            new ModifyCustomerOrder().setVisible(true);
        }
    });
```

```
});

mntmSearchSupplierOrder = new JMenuItem("Search_Supplier_Order");
mntmSearchSupplierOrder.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        new ModifySupplierOrder().setVisible(true);
    }
});

s2 = new JSeparator();
mnSearch.add(s2);
mnSearch.add(mntmSearchSupplierOrder);
mnSearch.add(mntmSearchCustomerOrder);

mnReport = new JMenu("Report");
menuBar.add(mnReport);

mntmEmployeeReport = new JMenuItem("Employee_Report");
mntmEmployeeReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdEmployeeInfo().setVisible(true);
    }
});
mnReport.add(mntmEmployeeReport);

mntmCustomerReport = new JMenuItem("Customer_Report");
mntmCustomerReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdCustomerInfo().setVisible(true);
    }
});
mnReport.add(mntmCustomerReport);

mntmSupplierReport = new JMenuItem("Supplier_Report");
mntmSupplierReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdSupplierInfo().setVisible(true);
    }
});
mnReport.add(mntmSupplierReport);

mntmProductReport = new JMenuItem("Product_Report");
mntmProductReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdProductInfo().setVisible(true);
    }
});
mnReport.add(mntmProductReport);
```



```
mntmCustOrderReport = new JMenuItem("Customer_Order_Report");
mntmCustOrderReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdCustomerOrder().setVisible(true);
    }
});

JSeparator separator = new JSeparator();
mnReport.add(separator);
mnReport.add(mntmCustOrderReport);

JMenuItem mntmSupplierOredrReport = new
JMenuItem("Supplier_Oredr_Report");
mntmSupplierOredrReport.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdSupplierOrder().setVisible(true);
    }
});
mnReport.add(mntmSupplierOredrReport);

mntmEmployeepaymnet = new JMenuItem("Employee_Paymnet");
mntmEmployeepaymnet.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        new RpdEmpPayment().setVisible(true);
    }
});
mnReport.add(mntmEmployeepaymnet);

mnUtilites = new JMenu("Utilites");
menuBar.add(mnUtilites);

mntmNotepad = new JMenuItem("Notepad");
mntmNotepad.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        try
        {
            Runtime.getRuntime().exec("Notepad.exe");
        }
        catch(Exception ex1)
        {
            JOptionPane.showMessageDialog(null,"Error,Cannot
start notepad","Application Error",JOptionPane.ERROR_MESSAGE);
        }
    }
});
mnUtilites.add(mntmNotepad);

mntmCalculator = new JMenuItem("Calculator");
```

```

mntmCalculator.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        try
        {
            Runtime.getRuntime().exec("calc.exe");
        }
        catch(Exception ex1)
        {
            JOptionPane.showMessageDialog(null,"Error,Cannot
start calculator","Application Error",JOptionPane.ERROR_MESSAGE);
        }
    }
});
mnUtilites.add(mntmCalculator);

mnHelp = new JMenu("Help");
menuBar.add(mnHelp);

mntmSystemInfo = new JMenuItem("System_Info");
mnHelp.add(mntmSystemInfo);

mntmContactUs = new JMenuItem("Contact_Us");
mnHelp.add(mntmContactUs);
getContentPane().setLayout(null);

}
}

```

- **Program Name:-** AddEmployee.java

**Program Desription:-** This form is used to Add Employee information.

**Code:-**

```

package com.chsm.form;
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.sql.*;

import com.chsm.form.db.DBEmployee;
import com.toedter.calendar.JDateChooser;

@SuppressWarnings("serial")
public class AddEmployee extends JFrame {

    private JPanel panel, panelEmppi, panelEmpoc, panelOperation;

```

```

        private JTextField txtEmployeeID, txtEmployeeName, txtEmployeeContact,
        txtEmployeeEmailID, txtEmployeeDesignation, txtEmployeeSalary, txtEmployeeUserName,
        txtEmployeePassword;
        private JLabel lblEmployeeId, lblEmployeeName, lblEmployeeBirthDate,
        lblEmployeeGender, lblEmployeeAddress, lblEmployeeContact, lblEmployeeEmailID,
        lblEmployeePassword;
        private JLabel lblEmployeeJoiningDate, lblEmployeeDesignation, lblEmployeeSalary,
        lblEmployeeUserName;
        private JRadioButton rbMale, rbFemale;
        private JButton btnSave, btnClear, btnExit;
        private ButtonGroup buttonGroup;
        private JTextArea txtaEmployeeAddress;
        private JScrollPane scrollPane;
        private JDateChooser txtEmployeeBirthDate, txtEmployeeJoiningDate;
        private DBEmployee db = new DBEmployee();

        public static java.sql.Date datemethad(java.util.Date date)
        {
            if(date != null)
            {
                java.sql.Date sqld = new Date(date.getTime());
                return sqld;
            }
            return null;
        }

        public AddEmployee()
        {
            setResizable(false);
            //setAlwaysOnTop(true);
            setTitle("Adding_Employee_Information");
            setBounds(100, 100, 1200, 525);
            panel = new JPanel();
            panel.setBorder(new EmptyBorder(5, 5, 5, 5));
            setContentPane(panel);
            panel.setLayout(null);

            panelEmppi = new JPanel();
            panelEmppi.setBorder(new
            TitledBorder(UIManager.getBorder("TitledBorder.border"),
            "Employee_Personal_Informations", TitledBorder.LEADING, TitledBorder.TOP, null, null));
            panelEmppi.setBounds(22, 19, 564, 458);
            panel.add(panelEmppi);
            panelEmppi.setLayout(null);

            lblEmployeeId = new JLabel("Employee ID :");
            lblEmployeeId.setFont(new Font("Tahoma", Font.PLAIN, 15));
            lblEmployeeId.setBounds(38, 44, 150, 25);
            panelEmppi.add(lblEmployeeId);

            txtEmployeeID = new JTextField();

```

```
txtEmployeeID.setColumns(10);
txtEmployeeID.setBounds(253, 44, 270, 25);
panelEmppei.add(txtEmployeeID);
txtEmployeeID.setText(db.setID());
txtEmployeeID.setEditable(false);
lblEmployeeName = new JLabel("Employee Name :");
lblEmployeeName.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeName.setBounds(38, 89, 150, 25);
panelEmppei.add(lblEmployeeName);

txtEmployeeName = new JTextField();
txtEmployeeName.setColumns(10);
txtEmployeeName.setBounds(253, 89, 270, 25);
panelEmppei.add(txtEmployeeName);

lblEmployeeBirthDate = new JLabel("Employee Birth Date :");
lblEmployeeBirthDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeBirthDate.setBounds(38, 134, 150, 25);
panelEmppei.add(lblEmployeeBirthDate);

lblEmployeeGender = new JLabel("Employee Gender :");
lblEmployeeGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeGender.setBounds(38, 183, 150, 25);
panelEmppei.add(lblEmployeeGender);

buttonGroup = new ButtonGroup();
rbMale = new JRadioButton("Male");
rbMale.setBackground(null);
buttonGroup.add(rbMale);
rbMale.setHorizontalAlignment(SwingConstants.CENTER);
rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbMale.setBounds(273, 184, 64, 23);
panelEmppei.add(rbMale);
rbMale.setActionCommand("Male");

rbFemale = new JRadioButton("Female");
rbFemale.setBackground(null);
buttonGroup.add(rbFemale);
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbFemale.setBounds(388, 184, 73, 23);
panelEmppei.add(rbFemale);
rbFemale.setActionCommand("Female");

lblEmployeeAddress = new JLabel("Employee Address :");
lblEmployeeAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeAddress.setBounds(38, 232, 150, 25);
panelEmppei.add(lblEmployeeAddress);

lblEmployeeContact = new JLabel("Employee Contact :");
lblEmployeeContact.setFont(new Font("Tahoma", Font.PLAIN, 15));
```

```
lblEmployeeContact.setBounds(38, 345, 150, 25);
panelEmppi.add(lblEmployeeContact);

txtEmployeeContact = new JTextField();
txtEmployeeContact.setColumns(10);
txtEmployeeContact.setBounds(253, 345, 270, 25);
panelEmppi.add(txtEmployeeContact);

lblEmployeeEmailID = new JLabel("Employee Email ID :");
lblEmployeeEmailID.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeEmailID.setBounds(38, 390, 150, 25);
panelEmppi.add(lblEmployeeEmailID);

txtEmployeeEmailID = new JTextField();
txtEmployeeEmailID.setColumns(10);
txtEmployeeEmailID.setBounds(253, 390, 270, 25);
panelEmppi.add(txtEmployeeEmailID);

scrollPane = new JScrollPane();
scrollPane.setBounds(253, 234, 270, 90);
panelEmppi.add(scrollPane);

txtaEmployeeAddress = new JTextArea();
scrollPane.setViewportViewView(txtaEmployeeAddress);

txtEmployeeBirthDate = new JDateChooser();
txtEmployeeBirthDate.setBounds(253, 134, 270, 25);
panelEmppi.add(txtEmployeeBirthDate);

panelEmpoc = new JPanel();
panelEmpoc.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Employee_Official_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelEmpoc.setBounds(608, 19, 564, 289);
panel.add(panelEmpoc);
panelEmpoc.setLayout(null);

lblEmployeeJoiningDate = new JLabel("Employee Joining Date :");
lblEmployeeJoiningDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeJoiningDate.setBounds(38, 27, 169, 25);
panelEmpoc.add(lblEmployeeJoiningDate);

lblEmployeeDesignation = new JLabel("Employee Designation :");
lblEmployeeDesignation.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeDesignation.setBounds(38, 79, 169, 25);
panelEmpoc.add(lblEmployeeDesignation);

txtEmployeeDesignation = new JTextField();
txtEmployeeDesignation.setColumns(10);
txtEmployeeDesignation.setBounds(253, 79, 270, 25);
panelEmpoc.add(txtEmployeeDesignation);
```

```
lblEmployeeSalary = new JLabel("Employee Salary :");
lblEmployeeSalary.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeSalary.setBounds(38, 235, 150, 25);
panelEmpoc.add(lblEmployeeSalary);

txtEmployeeSalary = new JTextField();
txtEmployeeSalary.setColumns(10);
txtEmployeeSalary.setBounds(253, 235, 270, 25);
panelEmpoc.add(txtEmployeeSalary);

lblEmployeeUserName = new JLabel("Employee User Name :");
lblEmployeeUserName.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeeUserName.setBounds(38, 131, 169, 25);
panelEmpoc.add(lblEmployeeUserName);

txtEmployeeUserName = new JTextField();
txtEmployeeUserName.setColumns(10);
txtEmployeeUserName.setBounds(253, 131, 270, 25);
panelEmpoc.add(txtEmployeeUserName);

lblEmployeePassword = new JLabel("Employee Password :");
lblEmployeePassword.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblEmployeePassword.setBounds(38, 183, 169, 25);
panelEmpoc.add(lblEmployeePassword);

txtEmployeePassword = new JTextField();
txtEmployeePassword.setColumns(10);
txtEmployeePassword.setBounds(253, 183, 270, 25);
panelEmpoc.add(txtEmployeePassword);

txtEmployeeJoiningDate = new JDateChooser();
txtEmployeeJoiningDate.setBackground(SystemColor.text);
txtEmployeeJoiningDate.setBounds(253, 27, 270, 25);
panelEmpoc.add(txtEmployeeJoiningDate);
panelOperation = new JPanel();
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 327, 564, 150);
panel.add(panelOperation);
panelOperation.setLayout(null);

btnSave = new JButton("Save");
btnSave.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        db.addemployee(Integer.parseInt(txtEmployeeID.getText()),
txtEmployeeContact.getText(), Integer.parseInt(txtEmployeeSalary.getText()),
```

```

        txtEmployeeName.getText(),
        buttonGroup.getSelection().getActionCommand(), txtaEmployeeAddress.getText(),
        txtEmployeeEmailID.getText(),
        txtEmployeeDesignation.getText(),
        datemethad(txtEmployeeBirthDate.getDate()),
        datemethad(txtEmployeeJoiningDate.getDate()));

        db.addlogin(Integer.parseInt(txtEmployeeID.getText()),
        txtEmployeeUserName.getText(), txtEmployeePassword.getText());
        JOptionPane.showMessageDialog(null, "Employee Data Save
        Sucesfull....");

        dispose();
    }
});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnSave.setBackground(new Color(230, 230, 250));
btnSave.setBounds(28, 50, 150, 50);
panelOperation.add(btnSave);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        txtEmployeeName.setText(null);
        txtEmployeeContact.setText(null);
        txtEmployeeEmailID.setText(null);
        txtEmployeeDesignation.setText(null);
        txtEmployeeSalary.setText(null);
        txtEmployeeUserName.setText(null);
        txtEmployeePassword.setText(null);
        txtaEmployeeAddress.setText(null);
        buttonGroup.clearSelection();
        txtEmployeeBirthDate.setCalendar(null);
        txtEmployeeJoiningDate.setCalendar(null);
    }
});
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(206, 50, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        JOptionPane.showMessageDialog(null, "Do you want to close
        Add_Employee....");
        dispose();
    }
}

```

```

    });
    btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
    btnExit.setBackground(new Color(230, 230, 250));
    btnExit.setBounds(384, 50, 150, 50);
    panelOperation.add(btnExit);

    }
}

```

- **Program Name:-** ModifyCustomer.java

**Program Description:-** This form is used to update the customer's information.

**Code:-**

```

package com.chsm.form.modify;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import com.chsm.form.db.DBCustomer;
import com.toedter.calendar.JDateChooser;
import DBmodel.sqlconnection;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

@SuppressWarnings("serial")
public class ModifyCustomer extends JFrame
{
    private JPanel panelCustoi, panel, panelCustpi, panelOperation, panelCustomerS;
    private JLabel lblCustomerId, lblCustomerEmailId, lblCustomerCreditAmount,
    lblCustomerContact, lblCustomerDebitAmount, lblCustomerName,
    lblCustomerBalanceAmount, lblCustomerRegisteredDate, lblCustomerGender,
    lblCustomerAddress;
    private JTextField txtCustomerName, txtCustomerBAmount, txtCustomerDeAmount,
    txtCustomerCrAmount, txtCustomerEmailId, txtCustomerContact;
    private JDateChooser txtCustomerRDate;
    @SuppressWarnings("rawtypes")
    private JComboBox txtCustomerId;
    private ButtonGroup buttonGroup;
    private JRadioButton rbMale, rbFemale;
    private JScrollPane scrollPane;
    private JTextArea txtCustomerAddress;
    private JButton btnExit, btnDelete, btnUupdate, btnClear;
    private int cid;
    private DBCustomer db = new DBCustomer();

    public static java.sql.Date datemethad(java.util.Date date)

```



```

    {
    if(date != null)
    {
        java.sql.Date sqld = new Date(date.getTime());
        return sqld;
    }
    return null;
    }

    @SuppressWarnings("unchecked")
    public void data()
    {
        try
        {
            Connection c = sqlconnection.getConnection();
            String sql = "Select * from customer";
            PreparedStatement pst = c.prepareStatement(sql);
            ResultSet rs = pst.executeQuery();
            if(!rs.isBeforeFirst())
            {
                JOptionPane.showMessageDialog(null, "there is no data of
Customer in table....");
            }
            else
            {
                while(rs.next())
                {
                    txtCustomerId.addItem(rs.getInt("id"));
                }
            }
            pst.close();
        } catch (SQLException e) { e.printStackTrace(); }
    }

    @SuppressWarnings("rawtypes")
    public ModifyCustomer()
    {
        setResizable(false);
        setTitle("Searching Customer Information");
        setBounds(100, 100, 1200, 525);
        panel = new JPanel();
        panel.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(panel);
        panel.setLayout(null);

        panelCustpi = new JPanel();
        panelCustpi.setLayout(null);
        panelCustpi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Customer_Personal_Informations", TitledBorder.LEADING, TitledBorder.TOP, null, null));

```

```
panelCustpi.setBounds(22, 163, 564, 307);
panel.add(panelCustpi);

lblCustomerRegisteredDate = new JLabel("Customer Registered Date :");
lblCustomerRegisteredDate.setForeground(Color.BLACK);
lblCustomerRegisteredDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerRegisteredDate.setBounds(35, 24, 189, 25);
panelCustpi.add(lblCustomerRegisteredDate);

lblCustomerGender = new JLabel("Customer Gender :");
lblCustomerGender.setForeground(Color.BLACK);
lblCustomerGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerGender.setBounds(35, 68, 189, 25);
panelCustpi.add(lblCustomerGender);

buttonGroup = new ButtonGroup();

rbMale = new JRadioButton("Male");
rbMale.setBackground(null);
buttonGroup.add(rbMale);
rbMale.setHorizontalAlignment(SwingConstants.CENTER);
rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbMale.setBounds(273, 69, 64, 23);
panelCustpi.add(rbMale);
rbMale.setActionCommand("Male");

rbFemale = new JRadioButton("Female");
rbFemale.setBackground(null);
buttonGroup.add(rbFemale);
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbFemale.setBounds(388, 69, 73, 23);
panelCustpi.add(rbFemale);
rbFemale.setActionCommand("Female");

lblCustomerAddress = new JLabel("Customer Address :");
lblCustomerAddress.setForeground(Color.BLACK);
lblCustomerAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerAddress.setBounds(35, 110, 189, 25);
panelCustpi.add(lblCustomerAddress);

lblCustomerContact = new JLabel("Customer Contact :");
lblCustomerContact.setForeground(Color.BLACK);
lblCustomerContact.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerContact.setBounds(35, 219, 189, 25);
panelCustpi.add(lblCustomerContact);

txtCustomerContact = new JTextField();
txtCustomerContact.setEditable(true);
txtCustomerContact.setColumns(10);
txtCustomerContact.setBackground(new Color(230, 230, 250));
```

```
txtCustomerContact.setBounds(259, 219, 270, 25);
panelCustpi.add(txtCustomerContact);

lblCustomerEmailId = new JLabel("Customer Email ID :");
lblCustomerEmailId.setForeground(Color.BLACK);
lblCustomerEmailId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerEmailId.setBounds(35, 263, 189, 25);
panelCustpi.add(lblCustomerEmailId);

txtCustomerEmailId = new JTextField();
txtCustomerEmailId.setEditable(true);
txtCustomerEmailId.setColumns(10);
txtCustomerEmailId.setBackground(new Color(230, 230, 250));
txtCustomerEmailId.setBounds(259, 263, 270, 25);
panelCustpi.add(txtCustomerEmailId);

scrollPane = new JScrollPane();
scrollPane.setBounds(259, 110, 270, 90);
panelCustpi.add(scrollPane);

txtCustomerAddress = new JTextArea();
scrollPane.setViewportViewView(txtCustomerAddress);

txtCustomerRDate = new JDateChooser();
txtCustomerRDate.setBounds(259, 24, 262, 25);
panelCustpi.add(txtCustomerRDate);

panelCustoi = new JPanel();
panelCustoi.setLayout(null);
panelCustoi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Customer_Official_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelCustoi.setBounds(608, 20, 564, 244);
panel.add(panelCustoi);

lblCustomerCreditAmount = new JLabel("Customer Credit Amount :");
lblCustomerCreditAmount.setForeground(Color.BLACK);
lblCustomerCreditAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerCreditAmount.setBounds(38, 42, 189, 25);
panelCustoi.add(lblCustomerCreditAmount);

txtCustomerCrAmount = new JTextField();
txtCustomerCrAmount.setEditable(true);
txtCustomerCrAmount.setColumns(10);
txtCustomerCrAmount.setBackground(new Color(230, 230, 250));
txtCustomerCrAmount.setBounds(253, 42, 270, 25);
panelCustoi.add(txtCustomerCrAmount);

lblCustomerDebitAmount = new JLabel("Customer Debit Amount :");
lblCustomerDebitAmount.setForeground(Color.BLACK);
lblCustomerDebitAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
```

```

lblCustomerDebitAmount.setBounds(38, 109, 189, 25);
panelCustoi.add(lblCustomerDebitAmount);

txtCustomerDeAmount = new JTextField();
txtCustomerDeAmount.setEditable(true);
txtCustomerDeAmount.setText("0");
txtCustomerDeAmount.setColumns(10);
txtCustomerDeAmount.setBackground(new Color(230, 230, 250));
txtCustomerDeAmount.setBounds(253, 109, 270, 25);
panelCustoi.add(txtCustomerDeAmount);
lblCustomerBalanceAmount = new JLabel("Customer Balance Amount :");
lblCustomerBalanceAmount.setForeground(Color.BLACK);
lblCustomerBalanceAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerBalanceAmount.setBounds(38, 176, 189, 25);
panelCustoi.add(lblCustomerBalanceAmount);

txtCustomerBAmount = new JTextField();
txtCustomerBAmount.setEditable(false);
txtCustomerBAmount.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int i=(Integer.parseInt(txtCustomerCrAmount.getText())-
Integer.parseInt(txtCustomerDeAmount.getText()));
        txtCustomerBAmount.setText(String.valueOf(i));
    }
});
txtCustomerBAmount.setColumns(10);
txtCustomerBAmount.setBackground(new Color(230, 230, 250));
txtCustomerBAmount.setBounds(253, 176, 270, 25);
panelCustoi.add(txtCustomerBAmount);

panelOperation = new JPanel();
panelOperation.setLayout(null);
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 275, 564, 195);
panel.add(panelOperation);

btnUupdate = new JButton("Update ");
btnUupdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        int i =
Integer.parseInt((txtCustomerId.getSelectedItem()).toString());
        db.updatecust(i, txtCustomerName.getText(),
datemethad(txtCustomerRDate.getDate()), buttonGroup.getSelection().getActionCommand(),
txtCustomerAddress.getText(),
txtCustomerContact.getText(), txtCustomerEmailId.getText(),
Integer.parseInt(txtCustomerCrAmount.getText()),

Integer.parseInt(txtCustomerBAmount.getText()));

```

```

        JOptionPane.showMessageDialog(null, "Update Customer Data
Sucesfull....");
        dispose();
    }
});
btnUupdate.setForeground(Color.BLACK);
btnUupdate.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnUupdate.setBackground(new Color(230, 230, 250));
btnUupdate.setBounds(88, 31, 150, 50);
panelOperation.add(btnUupdate);

btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int i =
Integer.parseInt((txtCustomerId.getSelectedItem()).toString());
        db.dlcust(i);
        JOptionPane.showMessageDialog(null, "Employee Data delete
Successfully....");
        dispose();
    }
});
btnDelete.setForeground(Color.BLACK);
btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnDelete.setBackground(new Color(230, 230, 250));
btnDelete.setBounds(88, 112, 150, 50);
panelOperation.add(btnDelete);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        txtCustomerName.setText(null);
        txtCustomerRDate.setDate(null);
        txtCustomerContact.setText(null);
        txtCustomerAddress.setText(null);
        txtCustomerContact.setText(null);
        txtCustomerEmailId.setText(null);
        txtCustomerCrAmount.setText(null);
        txtCustomerDeAmount.setText("0");
        txtCustomerBAmount.setText(null);
        buttonGroup.clearSelection();
    }
});
btnClear.setForeground(Color.BLACK);
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(326, 31, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {

```

```
        public void actionPerformed(ActionEvent arg0) {
            JOptionPane.showMessageDialog(null, "Do you want to close
Modify Customer....");
            dispose();
        }
    });
    btnExit.setForeground(Color.BLACK);
    btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
    btnExit.setBackground(new Color(230, 230, 250));
    btnExit.setBounds(326, 112, 150, 50);
    panelOperation.add(btnExit);

    panelCustomerS = new JPanel();
    panelCustomerS.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Customer_Search",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
    panelCustomerS.setBounds(22, 20, 564, 132);
    panel.add(panelCustomerS);
    panelCustomerS.setLayout(null);

    lblCustomerId = new JLabel("Customer ID :");
    lblCustomerId.setForeground(Color.BLACK);
    lblCustomerId.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblCustomerId.setBounds(35, 27, 189, 25);
    panelCustomerS.add(lblCustomerId);

    txtCustomerId = new JComboBox();
    txtCustomerId.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

cid=Integer.parseInt(txtCustomerId.getSelectedItem().toString());

            try
            {
                final Connection c =sqlconnection.getConnection();
                final String sql = "Select * from customer where id=?";
                final PreparedStatement pst = c.prepareStatement(sql);
                pst.setInt(1, cid);
                final ResultSet rs = pst.executeQuery();
                while(rs.next())
                {
                    txtCustomerName.setText(rs.getString("name"));

txtCustomerRDate.setDate(rs.getDate("reg_date"));

txtCustomerContact.setText(rs.getString("contact"));

txtCustomerAddress.setText(rs.getString("address"));

txtCustomerContact.setText(rs.getString("contact"));
```

```

txtCustomerEmailId.setText(rs.getString("email"));

txtCustomerCrAmount.setText(rs.getString("balance"));
        final String gender1=rs.getString("gender");
        if(gender1.equals("Male"))
        {
            rbMale.setSelected(true);
        }
        else
        {
            rbFemale.setSelected(true);
        }
    }
}
catch(final Exception ex)
{
    ex.printStackTrace();
}
});

txtCustomerId.setBounds(259, 27, 270, 25);
panelCustomerS.add(txtCustomerId);

lblCustomerName = new JLabel("Customer Name :");
lblCustomerName.setForeground(Color.BLACK);
lblCustomerName.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblCustomerName.setBounds(35, 79, 189, 25);
panelCustomerS.add(lblCustomerName);

txtCustomerName = new JTextField();
txtCustomerName.setEditable(true);
txtCustomerName.setColumns(10);
txtCustomerName.setBackground(new Color(230, 230, 250));
txtCustomerName.setBounds(259, 79, 270, 25);
panelCustomerS.add(txtCustomerName);

data();
    }
}

```

• **Program Name:-**ModifySupplier.java

**Program Description:-** This form is used to delete the Supplier info.

**Code:-**

```

package com.chsm.form.modify;
import java.awt.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;

```

```

import com.chsm.form.db.DBSupplier;
import com.toedter.calendar.JDateChooser;

import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class ModifySupplier extends JFrame {

    private JPanel panel, panelSupplierpi, panelSupplieroi, panelOperation,
    panelSupplierS;
    private JLabel lblSupplierName, lblSupplierRegisteredDate, lblSupplierId,
    lblSupplierBalanceAmount, lblSupplierGender, lblSupplierDebitAmount,
    lblSupplierCreditAmount, lblSupplierAddress, lblSupplierContact, lblSupplierEmailId;
    private JTextField txtSupplierName, txtSupplierContact, txtSupplierEmailId,
    txtSupplierCrAmount, txtSupplierDeAmount, txtSupplierBAmount;
    @SuppressWarnings("rawtypes")
    private JComboBox txtSuppliedId;
    private JDateChooser txtSupplierRDate;
    private JScrollPane scrollPane;
    private JRadioButton rbMale, rbFemale;
    private ButtonGroup buttonGroup;
    final JTextArea txtSupplierAddress ;
    private JButton btnUpadate, btnClear, btnExit, btnDelete;
    private int sid;
    private DBSupplier db = new DBSupplier();

    public static java.sql.Date datemethad(java.util.Date date)
    {
        if(date != null)
        {
            java.sql.Date sqld = new Date(date.getTime());
            return sqld;
        }
        return null;
    }

    @SuppressWarnings("unchecked")
    public void data()
    {
        try
        {
            Connection c =sqlconnection.getConnection();
            String sql = "Select * from supplier";
            PreparedStatement pst = c.prepareStatement(sql);
            ResultSet rs = pst.executeQuery();
            if(!rs.isBeforeFirst())
            {
                JOptionPane.showMessageDialog(null, "there is no data of
Supplier in table....");
            }
        }
    }
}

```



```

        }
        else
        {
            while(rs.next())
            {
                txtSupplierId.addItem(rs.getInt("id"));
            }
        }
        pst.close();
    } catch (SQLException e) { e.printStackTrace(); }
}

@SuppressWarnings("rawtypes")
public ModifySupplier()
{
    setTitle("Searching_Supplier_Information");
    setBounds(100, 100, 1200, 525);
    panel = new JPanel();
    panel.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(panel);
    panel.setLayout(null);

    panelSupplierpi = new JPanel();
    panelSupplierpi.setLayout(null);
    panelSupplierpi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"),
"Supplier_Personal_Information", TitledBorder.LEADING, TitledBorder.TOP, null, null));
    panelSupplierpi.setBounds(22, 163, 564, 307);
    panel.add(panelSupplierpi);

    lblSupplierRegisteredDate = new JLabel("Supplier Registered Date :");
    lblSupplierRegisteredDate.setForeground(Color.BLACK);
    lblSupplierRegisteredDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblSupplierRegisteredDate.setBounds(35, 24, 189, 25);
    panelSupplierpi.add(lblSupplierRegisteredDate);

    lblSupplierGender = new JLabel("Supplier Gender :");
    lblSupplierGender.setForeground(Color.BLACK);
    lblSupplierGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
    lblSupplierGender.setBounds(35, 68, 189, 25);
    panelSupplierpi.add(lblSupplierGender);

    buttonGroup = new ButtonGroup();

    rbMale = new JRadioButton("Male");
    rbMale.setBackground(null);
    buttonGroup.add(rbMale);
    rbMale.setHorizontalAlignment(SwingConstants.CENTER);
    rbMale.setFont(new Font("Tahoma", Font.PLAIN, 15));
    rbMale.setBounds(273, 69, 64, 23);
    panelSupplierpi.add(rbMale);

```

```
rbMale.setActionCommand("Male");

rbFemale = new JRadioButton("Female");
rbFemale.setBackground(null);
buttonGroup.add(rbFemale);
rbFemale.setHorizontalAlignment(SwingConstants.CENTER);
rbFemale.setFont(new Font("Tahoma", Font.PLAIN, 15));
rbFemale.setBounds(388, 69, 73, 23);
panelSupplierpi.add(rbFemale);
rbFemale.setActionCommand("Female");

lblSupplierAddress = new JLabel("Supplier Address :");
lblSupplierAddress.setForeground(Color.BLACK);
lblSupplierAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierAddress.setBounds(35, 110, 189, 25);
panelSupplierpi.add(lblSupplierAddress);

lblSupplierContact = new JLabel("Supplier Contact :");
lblSupplierContact.setForeground(Color.BLACK);
lblSupplierContact.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierContact.setBounds(35, 219, 189, 25);
panelSupplierpi.add(lblSupplierContact);

txtSupplierContact = new JTextField();
txtSupplierContact.setEditable(true);
txtSupplierContact.setColumns(10);
txtSupplierContact.setBackground(new Color(230, 230, 250));
txtSupplierContact.setBounds(259, 219, 270, 25);
panelSupplierpi.add(txtSupplierContact);

lblSupplierEmailId = new JLabel("Supplier Email ID :");
lblSupplierEmailId.setForeground(Color.BLACK);
lblSupplierEmailId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierEmailId.setBounds(35, 263, 189, 25);
panelSupplierpi.add(lblSupplierEmailId);

txtSupplierEmailId = new JTextField();
txtSupplierEmailId.setEditable(true);
txtSupplierEmailId.setColumns(10);
txtSupplierEmailId.setBackground(new Color(230, 230, 250));
txtSupplierEmailId.setBounds(259, 263, 270, 25);
panelSupplierpi.add(txtSupplierEmailId);

scrollPane = new JScrollPane();
scrollPane.setBounds(259, 110, 270, 90);
panelSupplierpi.add(scrollPane);

txtSupplierAddress = new JTextArea();
txtSupplierAddress.setEditable(true);
txtSupplierAddress.setBackground(new Color(230, 230, 250));
scrollPane.setViewportView(txtSupplierAddress);
```

```
txtSupplierRDate = new JDateChooser();
txtSupplierRDate.setBounds(259, 24, 270, 25);
panelSupplierpi.add(txtSupplierRDate);

panelSupplieroi = new JPanel();
panelSupplieroi.setLayout(null);
panelSupplieroi.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Supplier_Official_Information",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelSupplieroi.setBounds(608, 20, 564, 244);
panel.add(panelSupplieroi);

lblSupplierCreditAmount = new JLabel("Supplier Credit Amount :");
lblSupplierCreditAmount.setForeground(Color.BLACK);
lblSupplierCreditAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierCreditAmount.setBounds(38, 42, 189, 25);
panelSupplieroi.add(lblSupplierCreditAmount);

txtSupplierCrAmount = new JTextField();
txtSupplierCrAmount.setEditable(true);
txtSupplierCrAmount.setColumns(10);
txtSupplierCrAmount.setBackground(new Color(230, 230, 250));
txtSupplierCrAmount.setBounds(253, 42, 270, 25);
panelSupplieroi.add(txtSupplierCrAmount);

lblSupplierDebitAmount = new JLabel("Supplier Debit Amount :");
lblSupplierDebitAmount.setForeground(Color.BLACK);
lblSupplierDebitAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierDebitAmount.setBounds(38, 109, 189, 25);
panelSupplieroi.add(lblSupplierDebitAmount);

txtSupplierDeAmount = new JTextField();
txtSupplierDeAmount.setEditable(true);
txtSupplierDeAmount.setText("0");
txtSupplierDeAmount.setColumns(10);
txtSupplierDeAmount.setBackground(new Color(230, 230, 250));
txtSupplierDeAmount.setBounds(253, 109, 270, 25);
panelSupplieroi.add(txtSupplierDeAmount);

lblSupplierBalanceAmount = new JLabel("Supplier Balance Amount :");
lblSupplierBalanceAmount.setForeground(Color.BLACK);
lblSupplierBalanceAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierBalanceAmount.setBounds(38, 176, 189, 25);
panelSupplieroi.add(lblSupplierBalanceAmount);

txtSupplierBAmount = new JTextField();
txtSupplierBAmount.setEditable(false);
txtSupplierBAmount.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
```

```

        int i=(Integer.parseInt(txtSupplierCrAmount.getText())-
Integer.parseInt(txtSupplierDeAmount.getText()));
        txtSupplierBAmount.setText(String.valueOf(i));
    }
});
txtSupplierBAmount.setColumns(10);
txtSupplierBAmount.setBackground(new Color(230, 230, 250));
txtSupplierBAmount.setBounds(253, 176, 270, 25);
panelSupplieroi.add(txtSupplierBAmount);

panelOperation = new JPanel();
panelOperation.setLayout(null);
panelOperation.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Operation ",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelOperation.setBounds(608, 275, 564, 195);
panel.add(panelOperation);

btnUupdate = new JButton("Update ");
btnUupdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        int i =
Integer.parseInt((txtSuppliedId.getSelectedItem()).toString());

        db.updatesupplier(i, txtSupplierName.getText(),
datemethad(txtSupplierRDate.getDate()),
buttonGroup.getSelection().getActionCommand(), txtSupplierAddress.getText(),
txtSupplierContact.getText(),
txtSupplierEmailId.getText(), Integer.parseInt(txtSupplierCrAmount.getText()),
Integer.parseInt(txtSupplierBAmount.getText()));

        JOptionPane.showMessageDialog(null, "Update Supplier Data
Sucesfull....");

        dispose();
    }
});
btnUupdate.setForeground(Color.BLACK);
btnUupdate.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnUupdate.setBackground(new Color(230, 230, 250));
btnUupdate.setBounds(88, 31, 150, 50);
panelOperation.add(btnUupdate);

btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        int i =
Integer.parseInt((txtSuppliedId.getSelectedItem()).toString());
        db.dlsup(i);

```

```

        JOptionPane.showMessageDialog(null, "Supplier Data delete
Successfully....");
        dispose();
    }
});
btnDelete.setForeground(Color.BLACK);
btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnDelete.setBackground(new Color(230, 230, 250));
btnDelete.setBounds(88, 112, 150, 50);
panelOperation.add(btnDelete);

btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        txtSupplierAddress.setText(null);
        txtSupplierName.setText(null);
        txtSupplierContact.setText(null);
        txtSupplierEmailId.setText(null);
        txtSupplierCrAmount.setText(null);
        txtSupplierDeAmount.setText("0");
        txtSupplierBAmount.setText(null);
        txtSupplierRDate.setDate(null);
        buttonGroup.clearSelection();
    }
});
btnClear.setForeground(Color.BLACK);
btnClear.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnClear.setBackground(new Color(230, 230, 250));
btnClear.setBounds(326, 31, 150, 50);
panelOperation.add(btnClear);

btnExit = new JButton("Exit");
btnExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        JOptionPane.showMessageDialog(null, "Do you want to close
Modify_Supplier....");
        dispose();
    }
});
btnExit.setForeground(Color.BLACK);
btnExit.setFont(new Font("Tahoma", Font.PLAIN, 15));
btnExit.setBackground(new Color(230, 230, 250));
btnExit.setBounds(326, 112, 150, 50);
panelOperation.add(btnExit);

panelSupplierS = new JPanel();
panelSupplierS.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Supplier_Search",
TitledBorder.LEADING, TitledBorder.TOP, null, null));
panelSupplierS.setBounds(22, 20, 564, 132);
panel.add(panelSupplierS);

```

```
panelSupplierS.setLayout(null);

lblSupplierId = new JLabel("Supplier Id :");
lblSupplierId.setForeground(Color.BLACK);
lblSupplierId.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierId.setBounds(35, 27, 189, 25);
panelSupplierS.add(lblSupplierId);

lblSupplierName = new JLabel();
lblSupplierName.setText("Supplier Name :");
lblSupplierName.setForeground(Color.BLACK);
lblSupplierName.setFont(new Font("Tahoma", Font.PLAIN, 15));
lblSupplierName.setBounds(35, 79, 189, 25);
panelSupplierS.add(lblSupplierName);

txtSupplierName = new JTextField();
txtSupplierName.setEditable(true);
txtSupplierName.setBackground(new Color(230, 230, 250));
txtSupplierName.setBounds(259, 79, 270, 25);
panelSupplierS.add(txtSupplierName);
txtSupplierName.setColumns(10);

txtSuppliedId = new JComboBox();
txtSuppliedId.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

sid=Integer.parseInt(txtSuppliedId.getSelectedItem().toString());
        try
        {
            final Connection c =sqlconnection.getConnection();
            final String sql = "Select * from supplier where id=?";
            final PreparedStatement pst = c.prepareStatement(sql);
            pst.setInt(1, sid);
            final ResultSet rs = pst.executeQuery();
            while(rs.next())
            {
                txtSupplierName.setText(rs.getString("name"));

txtSupplierRDate.setDate(rs.getDate("reg_date"));

txtSupplierContact.setText(rs.getString("contact"));

txtSupplierAddress.setText(rs.getString("address"));

txtSupplierContact.setText(rs.getString("contact"));

txtSupplierEmailId.setText(rs.getString("email"));

txtSupplierCrAmount.setText(rs.getString("balance"));
                txtSupplierDeAmount.setText("0");
```

```

        final String gender1=rs.getString("gender");
        if(gender1.equals("Male"))
        {
            rbMale.setSelected(true);
        }
        else
        {
            rbFemale.setSelected(true);
        }
    }
}
catch(final Exception ex)
{
    ex.printStackTrace();
}
});
txtSuppliedId.setBounds(259, 27, 270, 25);
panelSupplierS.add(txtSuppliedId);

data();
}
}

```

- **Program Name:-**DBProduct.java

**Program Description:-** This form is used to save and modify the product info.  
**Code:-**

```

package com.chsm.form.db;
import java.awt.*;
import java.sql.*;

public class DBProduct
{
    private String query1, query2;
    private Connection con;
    private PreparedStatement pst;
    private ResultSet rs;
    private String id=null;

    public DBProduct()
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/chsm", "root", "root");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```
public String setID()
{
    try
    {
        query1 = "select id from product";
        pst = con.prepareStatement(query1);
        rs = pst.executeQuery();
        if(!rs.isBeforeFirst())
            id="1";
        else
        {
            rs.last();
            id = String.valueOf(rs.getInt(1) + 1);
        }
    }
    catch (HeadlessException | SQLException e)
    {
        e.printStackTrace();
    }

    return id;
}

public void addproduct(int id, String c_name, String p_name, String model_no, String
color, String description, int prise)
{
    try {
        query2="insert into product value(?,?,?,?);";
        pst=con.prepareStatement(query2);

        pst.setInt(1, id);
        pst.setString(2, c_name);
        pst.setString(3, p_name);
        pst.setString(4, model_no);
        pst.setString(5, color);
        pst.setString(6, description);
        pst.setInt(7, prise);
        pst.executeUpdate();

    } catch (HeadlessException | SQLException e) {
        e.printStackTrace();
    }
}
}
```

**Program Description:-** This form is used to view Customer Order Report

**Code:-**

```
package com.chsm.form.report;
import java.sql.*;
import java.text.*;
```



```
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.*;
import java.awt.*;
import java.awt.event.*;
import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class RpdCustomerOrder extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JLabel lblNewLabel;
    private JButton button;

    void view ()
    {
        try
        {
            Connection c=sqlconnection.getConnection();
            String s="select * from cust_order";
            PreparedStatement pst=c.prepareStatement(s);
            ResultSet rs=pst.executeQuery();
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }

        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    public RpdCustomerOrder() {
        setTitle("Customer_Order_Report");
        setBounds(100, 100, 900, 500);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(40, 100, 800, 300);
        contentPane.add(scrollPane);
        table = new JTable();
        scrollPane.setViewportView(table);
        lblNewLabel = new JLabel("Customer Order Report");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblNewLabel.setBounds(40, 36, 400, 30);
        contentPane.add(lblNewLabel);
        button = new JButton("Print");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
```

```

        MessageFormat header = new MessageFormat("Report print");
        MessageFormat footer = new
MessageFormat("Page{0,number,integer}");
        try{
            table.print(JTable.PrintMode.NORMAL,header,footer);
        } catch (java.awt.print.PrinterException ex){
            System.err.format("Can not print %s %n",ex.getMessage());
        }
    });
    button.setFont(new Font("Tahoma", Font.PLAIN, 15));
    button.setBackground(new Color(230, 230, 250));
    button.setBounds(740, 36, 100, 30);
    contentPane.add(button);
    view();
}
}

```

- **Program Name:-** RPDSuppOrder.java

**Program Description:-** This form is used for Show Supplier Order Report.

**Code:-**

```

package com.chsm.form.report;
import java.sql.*;
import java.text.*;
import javax.swing.*;
import javax.swing.border.*;
import net.proteanit.sql.*;
import java.awt.*;
import java.awt.event.*;
import DBmodel.sqlconnection;

@SuppressWarnings("serial")
public class RpdSupplierOrder extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JButton button;
    private JLabel lblSupplierOrderReport;

    void view()
    {
        try
        {
            Connection c=sqlconnection.getConnection();
            String s="select * from sup_order";
            PreparedStatement pst=c.prepareStatement(s);
            ResultSet rs=pst.executeQuery();
            table.setModel(DbUtils.resultSetToTableModel(rs));
        }
    }
}

```

```
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    public RpdSupplierOrder() {
        setTitle("Supplier_Order_Report");
        setBounds(100, 100, 900, 500);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(40, 100, 800, 300);
        contentPane.add(scrollPane);

        table = new JTable();
        scrollPane.setViewportView(table);

        button = new JButton("Print");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                MessageFormat header = new MessageFormat("Report print");
                MessageFormat footer = new
MessageFormat("Page{0,number,integer}");

                try{
                    table .print(JTable.PrintMode.NORMAL,header,footer);
                } catch (java.awt.print.PrinterException ex){
                    System.err.format("Can not print %s %n",ex.getMessage());
                }
            }
        });
        button.setFont(new Font("Tahoma", Font.PLAIN, 15));
        button.setBackground(new Color(230, 230, 250));
        button.setBounds(740, 36, 100, 30);
        contentPane.add(button);

        lblSupplierOrderReport = new JLabel("Supplier Order Report");
        lblSupplierOrderReport.setFont(new Font("Tahoma", Font.BOLD, 25));
        lblSupplierOrderReport.setBounds(40, 36, 400, 30);
        contentPane.add(lblSupplierOrderReport);
        view();
    }
}
```

## CHAPTER -8

# BIBLIOGRAPHY

Before and at the time of developing the project following books are feared which gear us seem important guidelines for designing and developing the project and project reports.

### 8.1 BIBLIOGRAPHICAL BOOKS:

1. Java Programming: Advanced Topics by Joe Wigglesworth and Paula McMillan, Thomson Course, Third Edition 2004
2. Java 2: The Complete Reference by Herbert Schildt, Tata McGraw Hill, Fifth Edition
3. Web Enabled Commercial Applications Development Using Java 2 by Ivan Bayross, BPB Publication, Revised Edition 2006

### 8.2 BIBLIOGRAPHICAL SITES:

- <http://www.google.co.in/>
- <http://sun.java.com/>
- <http://www.yahoo.co.in/>
- <http://www.w3school.com/>
- <http://www.syncjedit.co.in/>
- [www.mysql.com](http://www.mysql.com)
- [www.roseindia.net](http://www.roseindia.net)
- [www.tutorialspoint](http://www.tutorialspoint)