RAJARSHI SCHOOL OF MANAGEMENT AND TECHNOLOGY

MAJOR PROJECT

TOPIC :- STRUCTURE OF INVOICE

DEPARTMENT OF BIG DISCOUNT SHOP

BACHELOR OF | COMPUTER | APPLICATION | 2017-2020

Made By :- PRAKHAR PANDEY

course :- BCA 3RD YEAR

roll no. :- 1181840 7081

RAJARSHI SCHOOL OF MANAGEMENT AND TECHNOLOGY

Major Project

STRUCTURE OF INVOICE

DEPARTMENT OF

BIG DISCOUNT SHOP

Submitted BY: PRAKHAR PANDEY

Course : BCA 3RD YEAR

Roll no. : 11818407081

Submitted to:

DR. C.P SINGH

"SIR"

Major Project On

Structure Of

Invoice

Department Of

Big Discount

Shop

Certificate

Name : Prakhar Pandey

Course : Bachelor Of Computer Application

Roll No. : 11818407081

College : **RAJARSHI SCHOOL OF MANAGEMENT AND TECHNOLOGY**

This is certified to be the bonafide work of student in MAJOR PROJECT on "**STRUCTURE OF INVOICE DEPARTMENT OF BIG DISCOUNT SHOP" during the academic year 2019-2020.**

_____ _____

COORDINATOR'S SIGNATURE EXAMINER SIGNATURE

DATE: _____

Acknowledgement

I would like to express my special thanks of gratitude to my teacher. **Mr. Vijay Pandey** Sir as well as our Co-ordinator **Dr. C.P Singh** Sir who gave me the golden opportunity to do this wonderful Major project on the topic "**STRUCTURE OF INVOICE DEPARTMENTOF BIG DISCOUNT SHOP**", which also help me in doing a lot research and I came to know about so many new things I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project report within the limited time frame.

THANK YOU

TABLE OF CONTENT

INTRODUCTION OF THE PROJECT "STRUCTURE OF INVOICE DEPARTMENT OF BIG DISCOUNT SHOP":

The "Structure of Invoice Department of Big Discount Shop" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Structure of Invoice Department of Big Discount Shop, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Invoice, Customer, Sales, Invoice Receipt, and Payment. Every Shop has different customer needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

ORGANIZATION OVERVIEW

The project is on Structure of Invoice Department of Big Discount Shop. The shop is a place where customer comes to purchase their daily using products & pay for that. So there is a need to calculate what product they are buying & to generate the invoice for the customers.

In this project we have 2 users. First is the cashier who will generate the invoice toward that product which customers are buying & print the invoice. Second one is the administrator (admin) who will enter the product in the database & will decide the discount for product & also deal with software security.

OBJECTIVE OF THE PROJECT

The main objective while implementing the project Structure of Invoice Department of Big Discount Shop was to minimize the work and at the same time increase the speed of the work done.

THIS NEW SYSTEM IS BUILT WITH THE FOLLOWING OBJECTIVE:

1. Information retrieval will become easy.
2. Maintenance of database as well as overall project will become easy.
3. Security measure will be adopted, by maintaining the login of username and the password.

SPECIFIC OBJECTIVE:

1. Analyze the existing system.
2. To give fast and accurate access for customers.
3. To Avoid Workload.
4. System analysis and object design.
5. Requirement analysis of the system.

6. Collecting data about the organization.

7. Identifying the problem under the existing system.

CURRENT SYSTEM

Many Shop use pen & paper for creating invoice for a decade. It is improved many times according to requirements of sellers & customers. It does the same work i.e. calculating the invoice, gives it to the customer & maintain proper database.

A new concept is added in the invoice system is that they have to generate invoice & update their database in a digitalization way. In this way they will able to maintain good relationship with their customer by taking his less time in compare to making pen paper invoice.

The owner will get advantage for the security that they don't have to put a lot of register for maintain database. Here they will able to maintain their database in very less time with secure & unique login ID & PASSWORD.

LIMITATIONS OF CURRENT SYSTEM

Every system has pros & cons so existing system have many advantages & disadvantages.

So, the limitation of existing system are as follows:

- Processing Speed:

  Processing speed of the human is not so much good to operate fast as computer can.

- Workload:

  Sometimes the human being stops working or not working in a correct manner because of sudden increment in work.

- Error Free:

  Sometimes the human being gives error in the calculation in making invoice & in the information of the product due to workload.

- Man Power:

  Existing system uses so many people to do a single task.

- Resources:

  The existing system does not using the resources.

ABOUT C++

C++ (pronounced cee plus plus) is a general purpose programming language. It has imperative, object-oriented and generic programming features, while also providing the facilities for low level memory manipulation.

It is designed with a bias for systems programming (e.g. embedded systems, operating system kernels), with performance, efficiency and flexibility of use as its design requirements. C++ has also been found useful in many other contexts, including desktop applications, servers (e.g. e-commerce, web search, SQL), performance critical applications (e.g. telephone switches, space probes) and entertainment software, such as video games.

It is a compiled language, with implementations of it available on many platforms. Various organizations provide them, including the FSF, LLVM, Microsoft and Intel.

Before standardization (1989 onwards), C++ was developed by Bjarne Stroustrup at Bell Labs, starting in 1979, who wanted an efficient flexible language (like C) that also provided high level features for program organization.

Many other programming languages have been influenced by C++, including C#, Java, and newer versions of C (after 1998).

ELEMENT OF C++

OBJECTS:

C++ introduces object-oriented programming (OOP) features to C. It offers classes, which provide the four features commonly present in OOP (and some non-OOP) languages: abstraction, encapsulation, inheritance, and polymorphism. One distinguishing feature of C++ classes compared to classes in other programming languages is support for deterministic destructors, which in turn provide support for the Resource Acquisition is Initialization (RAII) concept.

ENCAPSULATION:

Encapsulation is the hiding of information to ensure that data structures and operators are used as intended and to make the usage model more obvious to the developer. C++ provides the ability to define classes and functions as its primary encapsulation mechanisms. Within a class, members can be declared as either public, protected, or private to explicitly enforce encapsulation. A public member of the class is accessible to any function. A private member is accessible only to functions that are members of that class and to functions and classes explicitly granted access permission by the class ("friends"). A protected member is accessible to members of classes that inherit from the class in addition to the class itself and any friends.

The OO principle is that all of the functions (and only the functions) that access the internal representation of a type should be encapsulated within the type definition. C++ supports this (via member functions and friend functions), but does not enforce it: the

programmer can declare parts or all of the representation of a type to be public, and is allowed to make public entities that are not part of the representation of the type.

Therefore, C++ supports not just OO programming, but other weaker decomposition paradigms, like modular programming.

It is generally considered good practice to make all data private or protected, and to make public only those functions that are part of a minimal interface for users of the class. This can hide the details of data implementation, allowing the designer to later fundamentally change the implementation without changing the interface in any way.

INHERITANCE:

Inheritance allows one data type to acquire properties of other data types. Inheritance from a base class may be declared as public, protected, or private. This access specifier determines whether unrelated and derived classes can access the inherited public and protected members of the base class. Only public inheritance corresponds to what is usually meant by "inheritance". The other two forms are much less frequently used. If the access specifier is omitted, a "class" inherits privately, while a "struct" inherits publicly. Base classes may be declared as virtual; this is called virtual inheritance. Virtual inheritance ensures that only one instance of a base class exists in the inheritance graph, avoiding some of the ambiguity problems of multiple inheritance.

Multiple inheritance is a C++ feature not found in most other languages, allowing a class to be derived from more than one base class; this allows for more elaborate inheritance relationships.

For example, a "Flying Cat" class can inherit from both "Cat" and "Flying Mammal". Some other languages, such as C# or Java, accomplish something similar (although more limited) by allowing inheritance of multiple interfaces while restricting the number of base classes to one (interfaces, unlike classes, provide only declarations of member functions, no implementation or member data). An interface as in C# and Java can be defined in C++ as a class containing only pure virtual functions, often known as an abstract base class or "ABC". The member functions of such an abstract base class are normally explicitly defined in the derived class, not inherited implicitly. C++ virtual inheritance exhibits an ambiguity resolution feature called dominance.

POLYMORPHISM:

Polymorphism enables one common interface for many implementations, and for objects to act differently under different circumstances.

C++ supports several kinds of static (compile-time) and dynamic (runtime) polymorphisms. Compile-time polymorphism does not allow for certain run-time decisions, while run-time polymorphism typically incurs a performance penalty.

ADVANTAGE OF PROPOSED SYSTEM

To reduce the limitations of the existing system there is a need to develop a new system. The new system should concern the requirements of the customers & the sellers.

It has following quality:

- Reduction in processing cost.
- Error reduction.
- Faster response time.
- Ability to meet user requirement.
- Reduced dependency.
- Reduction in use of papers.
- Improve resource users.
- Reduction in use of man power.
- Generate bill.
- Generate Report.

RISK ANALYSIS

1. Shortage of Time. We managed it by using additional time from our rest time.

2. Virus can attack our project. We use antivirus to manage this problem.

3. Damaging the computers that we work on, we try to manage by using backup by using external hard disk.

4. Power fluctuation problem. By using laptop that have high power back ups are used.

WORK FLOW

Work in the Shop will be done in the following way:

1. The product will come in the shop.

2. The Admin will enter the information of the product in database.

3. The Administrator (admin) will enter the discount for each product.

4. The customer will come and take the basket with them and choose the product and took it to the counter.

5. The cashier will check the products with product_no. then it will match with product_no saved in database then it will show its information and price and the invoice will be calculated and total payment will shown.

6. Customer will pay for the products.

7. All the products will be packed and delivered to the customer.

FEASIBILITY STUDY

After doing the project Structure of Invoice Department of Big Discount Shop, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

"**Feasibility Study**" is a test of the system according to its workability, impact of the organization, ability to meet user needs and effective use of the resources.

We can test our system by different type of the feasibilities.

TYPES OF THE FEASIBILITIES:

There are 5 types of the feasibilities which are discussed here. These are as follows:

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility
4. Schedule Feasibility
5. Behavioral Feasibility

TECHNICAL FEASIBILITY:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

ECONOMICAL FEASIBILITY:

In this we consider following costs:

1. The cost to conduct a full system investigation.
2. The cost of hardware and software for class of application being considered.
3. The benefit in the form of the reduced cost.

Our system has a lot of features at a minimum cost so it is feasible to implement and it will be very much beneficial to the sellers

in the reduced cost. It's software and hardware cost is also low then the existing system.

OPERATIONAL FEASIBILITY:

In this feasibility we consider following points:

1. What changes will be brought with the system?
2. What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

   In the new system we made some major changes for the user so that they have to be little bit trained to use the newly updated facilities. These major changes are possible and will give a new era in the Big Discount Shop in sales and invoice management.

SCHEDULE FEASIBILITY:

Time evaluation is most important consideration in development of the project. So the project is concerned should be completed with fixed in scheduled time as far as company is concerned. New system is not so much big so it is easy to make in few days.

BEHAVIORAL FEASIBILITY:

People are inherently resisted to change and a computer means "change is the only certainty". An estimate should be made of how strong a reaction the user staff in going to have towards development of new system. Thus special efforts can be made to educate and train the staff.

SYSTEM DESIGN

INTRODUCTION:

System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system is designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts prepared in the study time received and decomposed until all functions in the system perform evidently.

Design is a multistep process that focuses on data structures, software architecture, procedural details(algorithms etc) and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design. Modularization of task is made in the mode. The success of any integrated system depends on the planning of each and every fundamental module. Usually a project is revised in step by step sequence. Inter phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve.

Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test. The techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test that are required to build and verify software. Each activity transforms information, which validates the software. The design system

converts theoretical solution introduced by the feasibility study into a logical reality.

DESIGN STRATEGY:

The design strategy is a vital aspect of the system to be developed. The design of the software reflects the basic understanding of the problem. For designing a good system what we have to be is to get correct definition of the problem and analyze the problem thoroughly.

The design of a system should be such that if a small portion is changed. The rest of the system should be unaffected. This is the flexibility of the system. Greater the system flexibility greater will be the system reliability. While carrying out the job of designing of a new system one has to consider many factors. These factors include the drawbacks and limitations of the present manual system as well as of the features and advantages of the proposed system. It should be designed in such a manner that even a layman can run it without any difficulty.

An important quality of a software must enjoy is “user friendliness”. It can be achieved in many ways like providing menu, giving context sensitive help, doing automatic validation to input data, etc. Another main factor is speed efficiency. In order to achieve speed efficiency, the program should be designed accordingly and the user is provided with a compiled copy of the software package with necessary data file format rather than source code.

Design of input and output formats is equally important for any design. The output format should be designed in such a way that it must reflect all the required information in detail. The design of the database itself such as type of data stored, size of data etc. Some of the decisions made during database design are:

- Which data items are to be recorded and in which database.
- Length of each record, based on the characteristics of the data item
- Data who’s unauthorized change must be prevented.
- Maintenance of data integrity etc.
- Prevents invalid data access and changes.

Having all this, a positive interaction with clients at every stage of development is the core around which the software is built.

INPUT DESIGN:

Input design is the process of converting user-originate inputs to a computer-based format. The goal of design input data is to make data entry as easy, logical and free. The most common source of data processing errors is inactive input data. Effective design of the input data minimizes the error made by data entry operators. Catching errors on input is far less costly than correcting after data storage is complete.

User-friendly input design enables quick error detecting and correction. Verification and validation is the most important in input design. Since the system is used interactively, it has two types of inputs. Interactive input-which is the point contact of the user with the system and the input to the internal system i.e. Databases. For full efficiency of the system, it is necessary that the input must be accurate. Since the user of the system may not be a technical person and may not know input concepts so it is required that he warn, prevent and correct invalid data entry.

There are many ways that can be designed to handle such a situation. We can prevent the user entering invalid data into the databases by warning, neglecting or messaging appropriately. The user is then allowed to input correct data. Some help provisions may aid the user to point out the error. In this system inputs are collected from terminals through keyboard.

OUTPUT DESIGN:

Output design has been an ongoing activity from the very beginning of the project. The objective of the output design is to convey the information of all past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system.

The output design of the system is accomplished keeping in mind the following activities:

- Determine what information is to display.
- Decide whether to display or print the information retrieved, processed, generated from the system.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to the intended recipients.

In the output design phase one or more output media can be selected. Out of which the most common ones are CRT displays and print out. Here only CRT display has been attempted. A rapid enquiry is obtained from CRT displays. From design is made interesting and attractive. Easy understanding and effectiveness is made possible.

TABLE DESIGN:

i) PRODUCT:

NAME(add product)	TYPE
PRODUCT NUMBER	NUMBER
PRODUCT NAME	TEXT
PRICE	NUMBER
DISCOUNT	NUMBER

ii) USER (CASHIER):

NAME(USER LOGIN)	TYPE
NAME	TEXT
USER NAME	TEXT
PASSWORD	TEXT

iii) INVOICE DETAILS:

NAME (INVOICE)	TYPE
CNAME	TEXT
DATE	NUMBER
Pr No.	NUMBER
Pr NAME	TEXT
QUANTITY	NUMBER
PRICE	NUMBER
AMOUNT	NUMBER
AMOUNT AFTER DISCOUNT	NUMBER

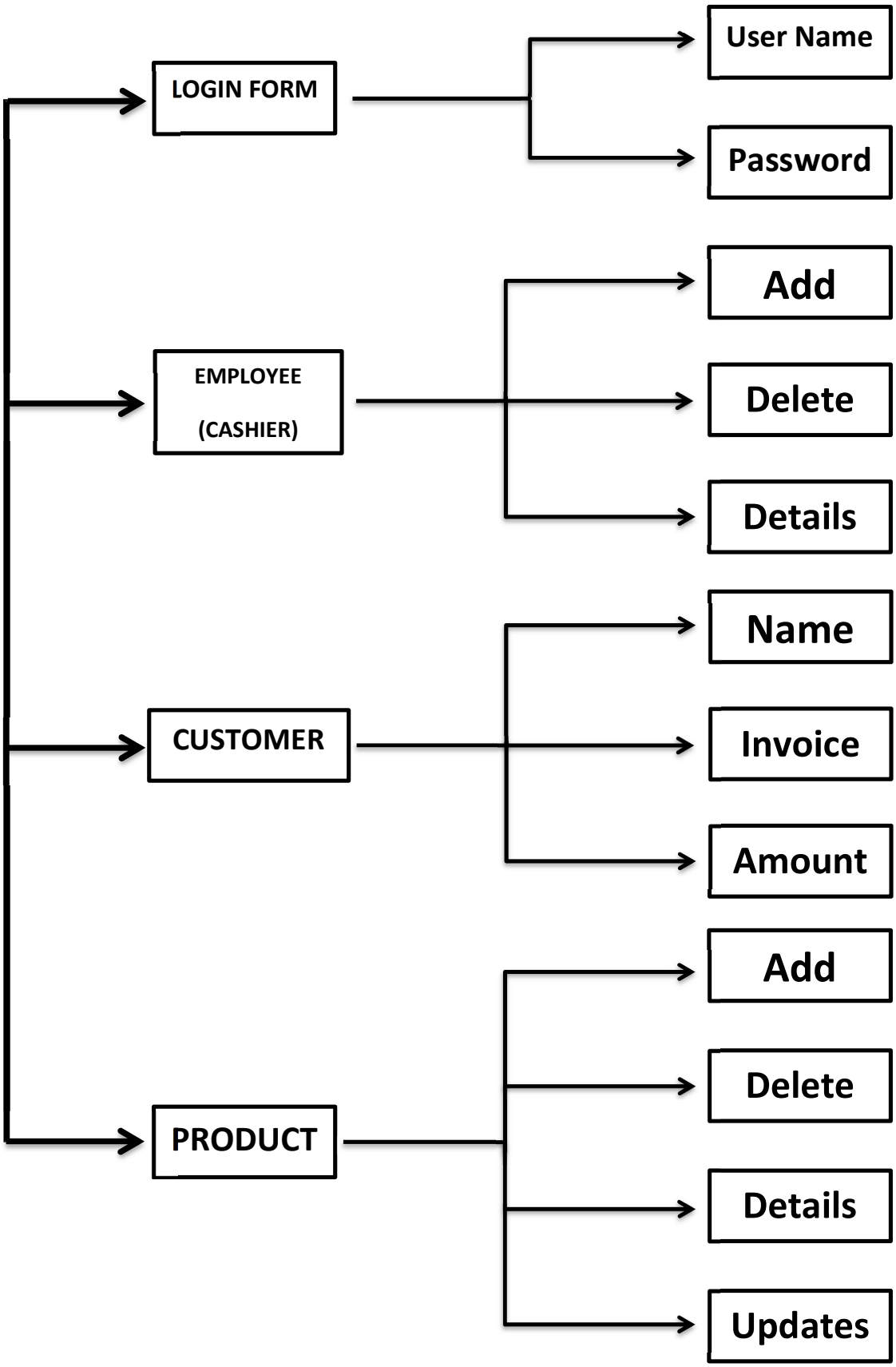
iv) ADMIN:

NAME (ADMIN LOGIN)	TYPE
NAME	TEXT
PASSWORD	TEXT

v) REPORT:

REPORT CONTENT	TYPE
CUSTOMER NAME	TEXT
DATE	NUMBER
TOTAL PAID AMOUNT	NUMBER

MENU TREE


LOGIN FORM
User Name
Password
EMPLOYEE
(CASHIER)
Add
Delete
Details
CUSTOMER
Name
Invoice
Amount
PRODUCT
Add
Delete
Details
Updates

PROJECT DESIGN

THIS SOFTWARE HAS BEEN DESIGNED TO:

I. Store the product details.

II. Delete the product.

III. Modify the product details.

IV. Store the customer's name.

V. Store the date of purchasing.

VI. Store the report.

VII. Generate the bill.

VIII. Generate the report.

IX. Provide security to data.

TABLE STRUCTURE

THREE FILES HAVE BEEN USED IN THIS PROJECT

1. SHOP.DAT (STORE ALL INFORMATION RELATED TO SHOP AND PRODUCT)

- P_No. : PRODUCT NUMBER
- P_NAME : PRODUCT NAME
- PRICE : SELLING PRICE OF PRODUCT
- DISCOUNT : DISCOUNT PERCENTAGE ON PARTICULAR PRODUCT

2. SECURITY.DAT (STORE ALL INFORMATION RELATED TO USERS)

- U_NAME : NAME OF USER (CASHIER)
- U_ID : USER LOG ID OF USER (CASHIER)
- U_PASSWORD : USER LOG IN PASSWORD OF THE USER (CASHIER)

3. CUSTOMER.DAT (STORE ALL INFORMATION OF CUSTOMERS)

- CNAME : NAME OF THE CUSTOMER
- DoP : DATE OF PURCHASING PRODUCT
- AMOUNT : TOTAL AMOUNT OF PURCHASING (TOTAL PAYABLE AMOUNT)

PROGRAM STRUCTURE

THREE CLASSES HAVE BEEN USED IN THIS PROJECT

1. PRODUCT CLASS

It contains two functions

A. Create product

This user defined function has created to store the all information related to product, like:

a) Product number.
b) Product name.
c) Price.
d) Discount.

B. Show product

This user defined function has created to display the information related to product, like:

a) Product number.
b) Product name.
c) Price.
d) Discount.

2. AUTHENTICATION CLASS

It contains two functions

A. Create Security

This user defined function has created to store the all information related to user, like:

a) Name.
b) User ID.
c) Password.

B. Show User

This user defined function has created to display the information related to product, like:

a) Name.
b) User ID.

3. REPORT CLASS

It contains two functions

A **Create Report**

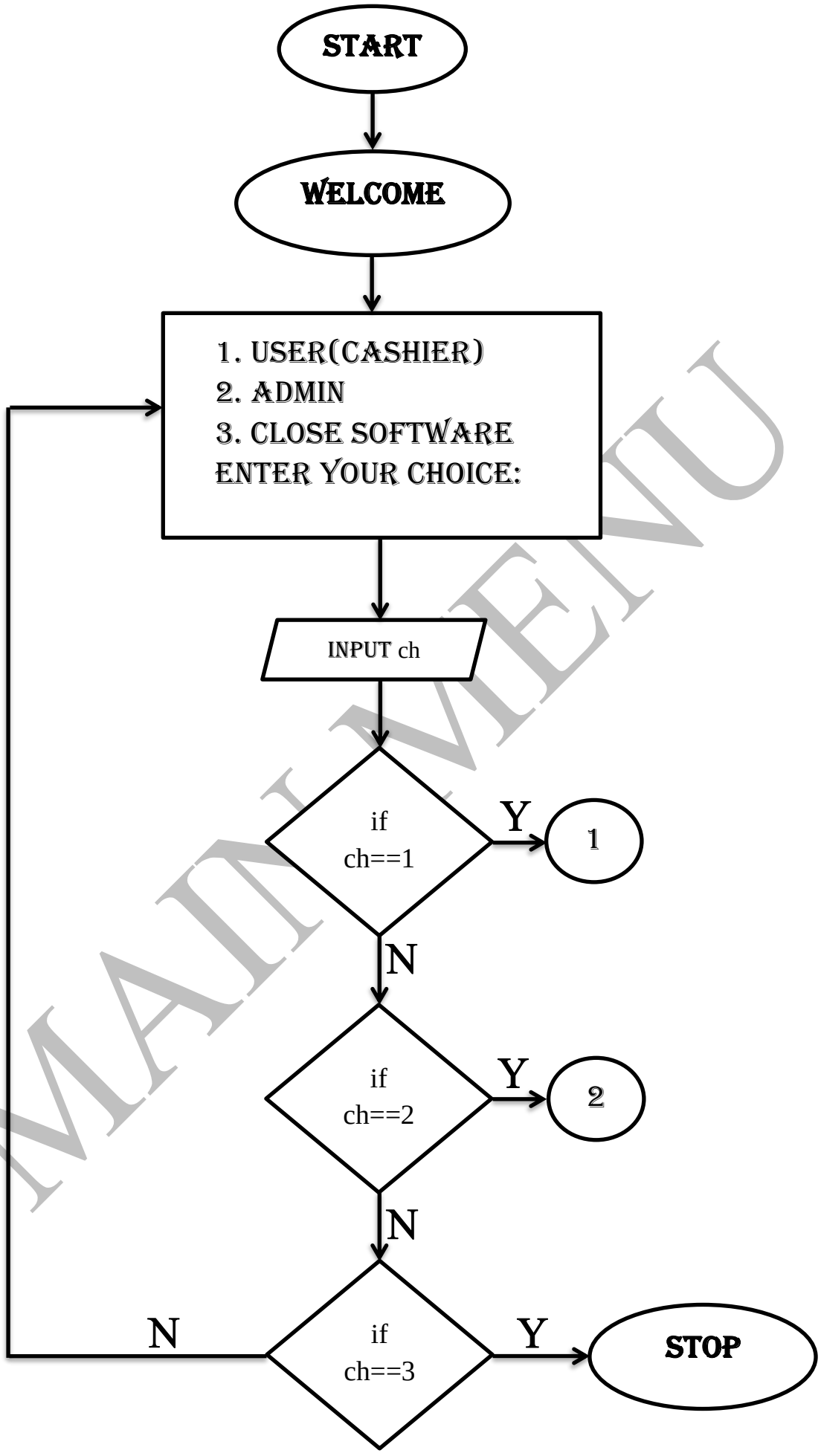
This user defined function has created to store the all information related to customer & sales, like:

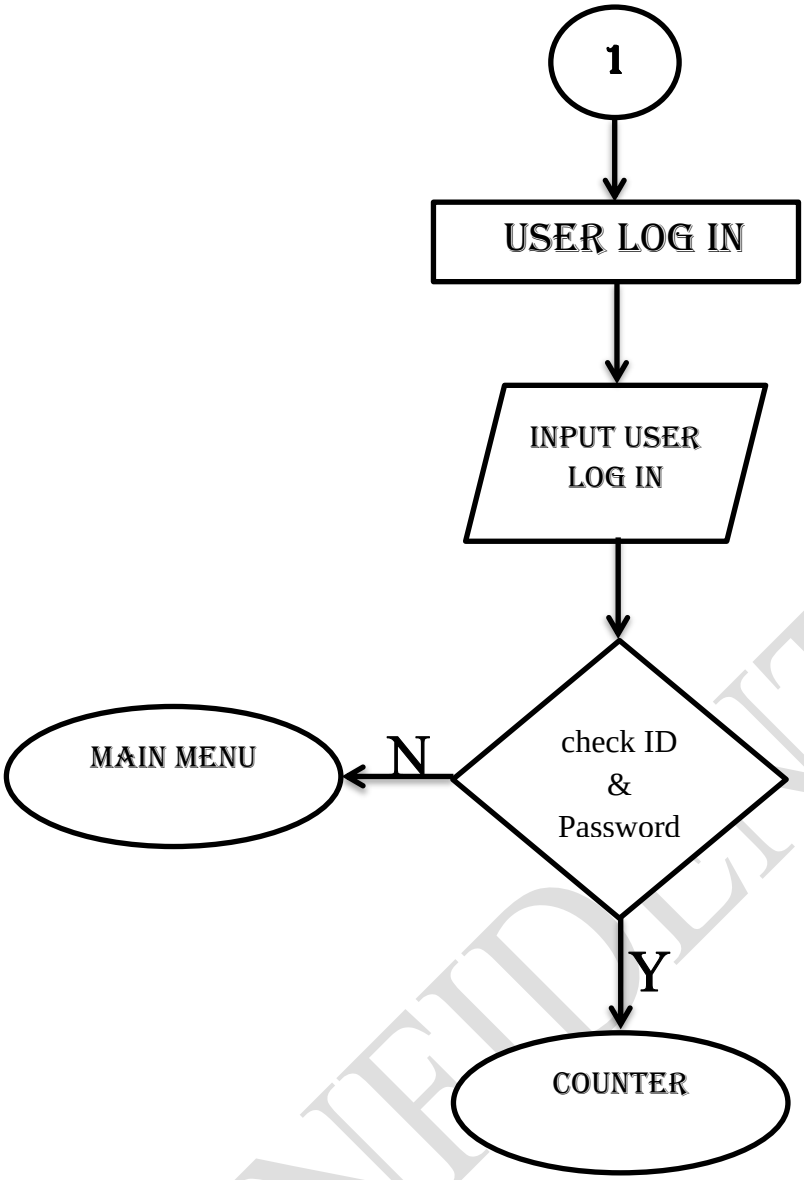
a) Customer Name.
b) Date of Purchasing.
c) Total Paid Amount.

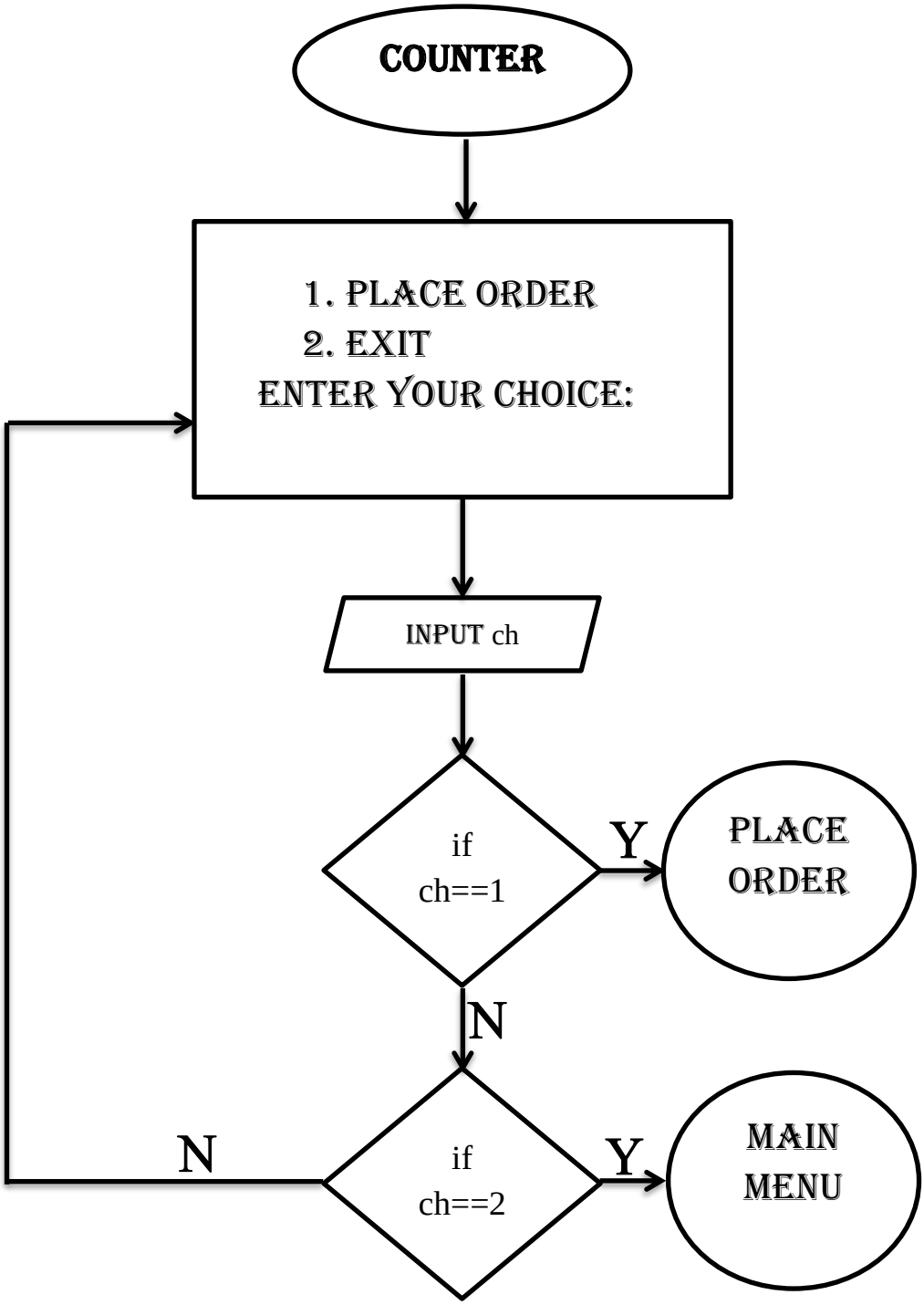
B. Show Report

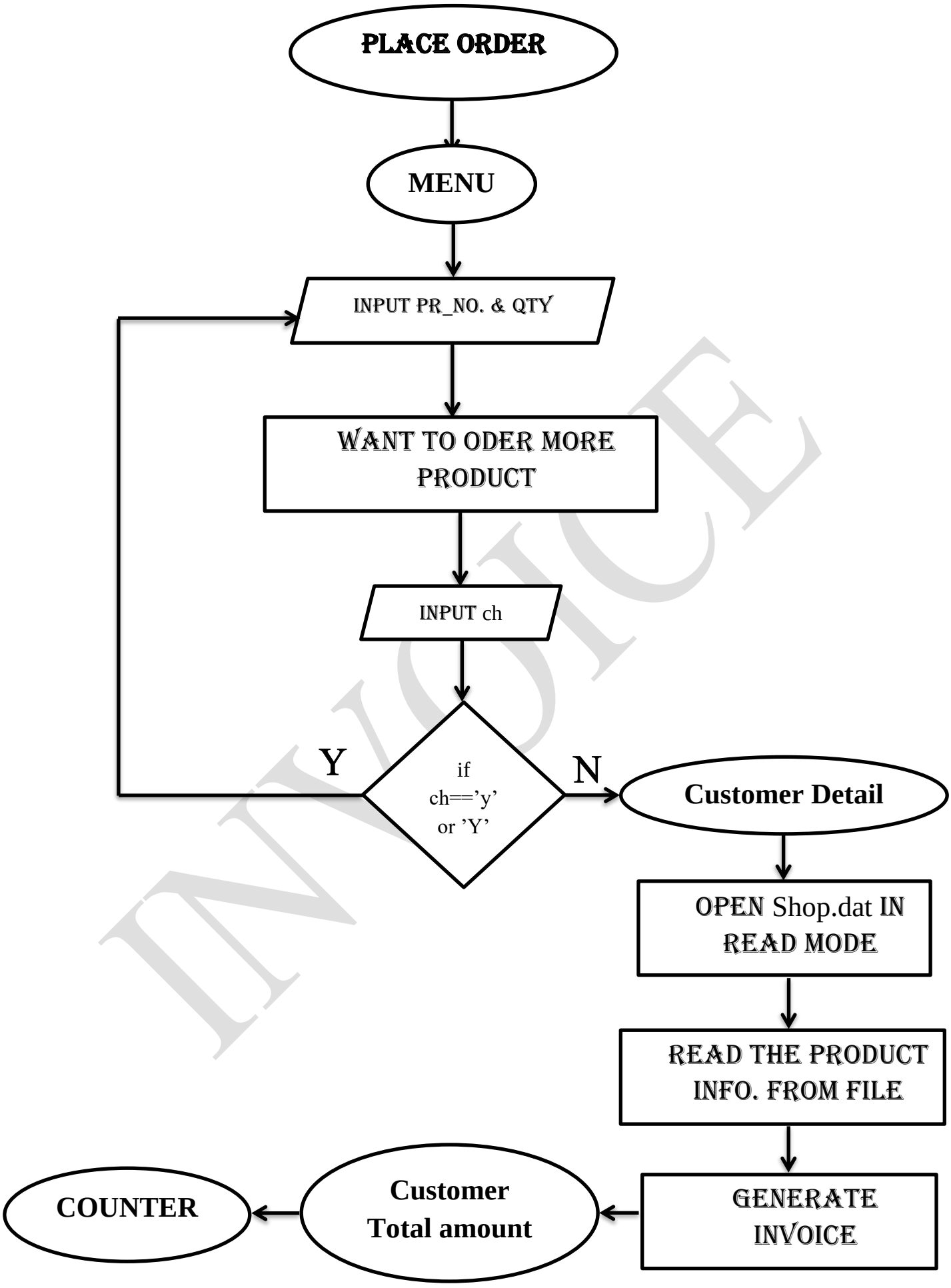
This user defined function has created to display the information related to customer & sales, like:

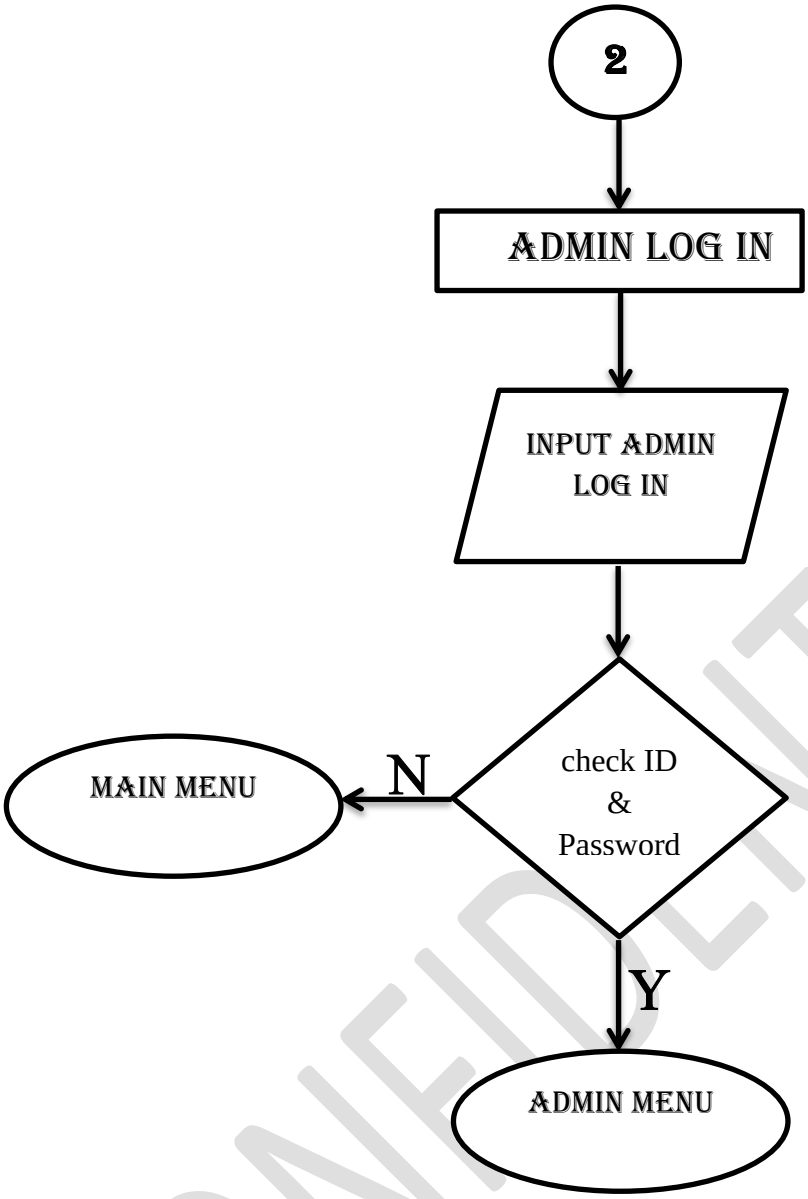
a) Customer Name.
b) Date of Purchasing.
c) Total Paid Amount.

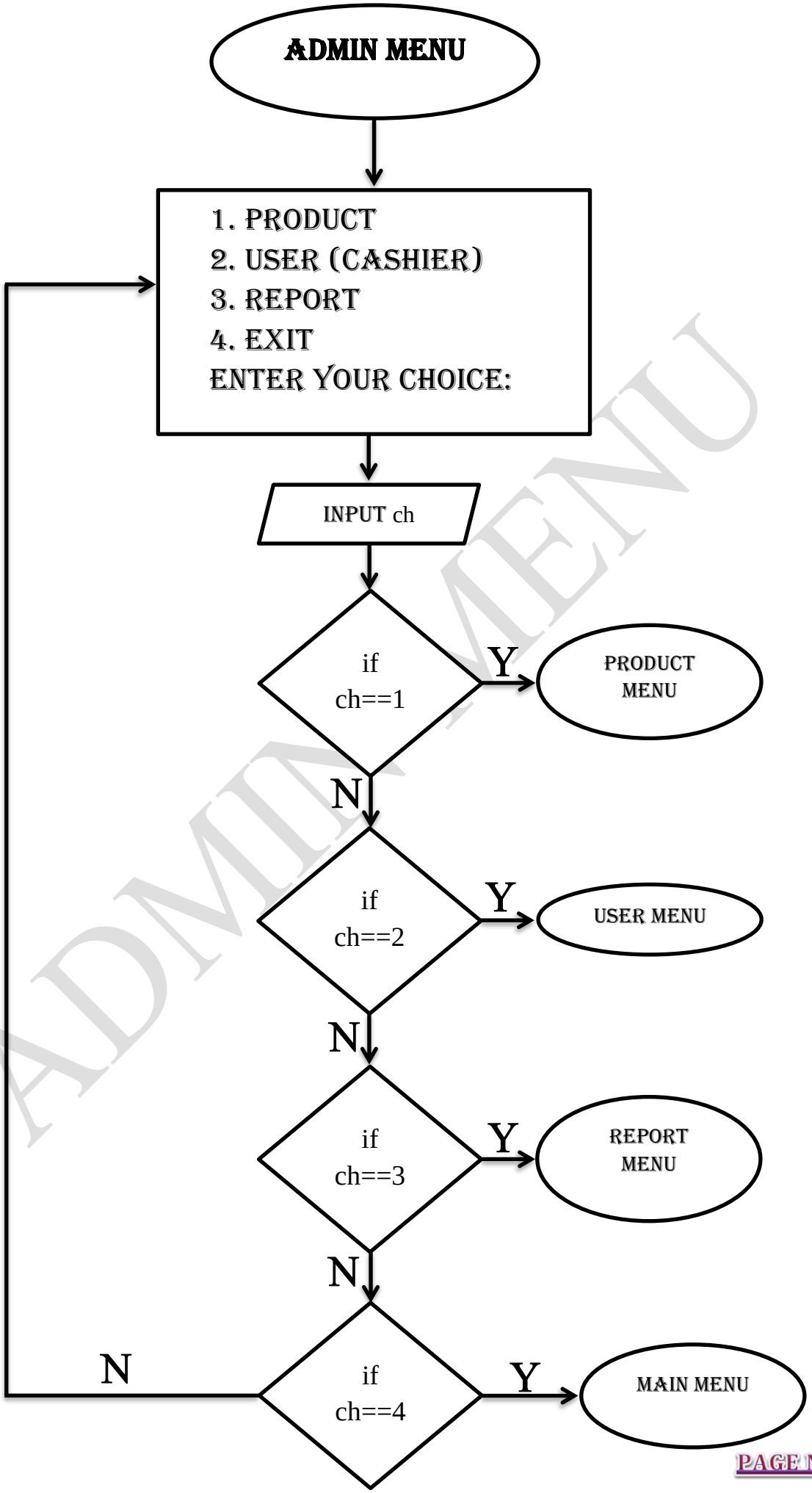

START
WELCOME
1. USER(CASHIER)
2. ADMIN
3. CLOSE SOFTWARE
ENTER YOUR CHOICE:
INPUT ch
if ch==1
Y
1
N
if ch==2
Y
2
N
if ch==3
N
Y
STOP

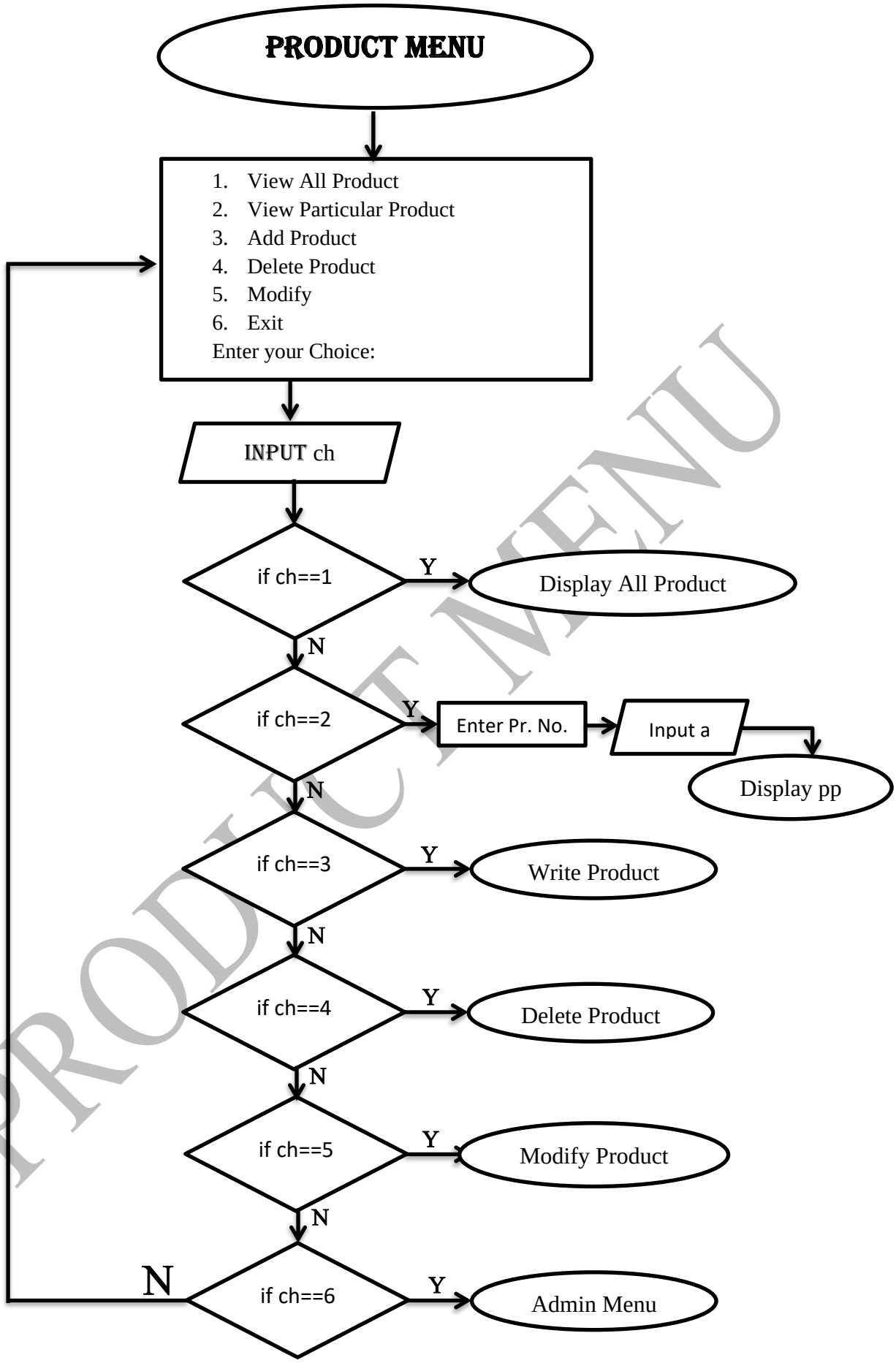


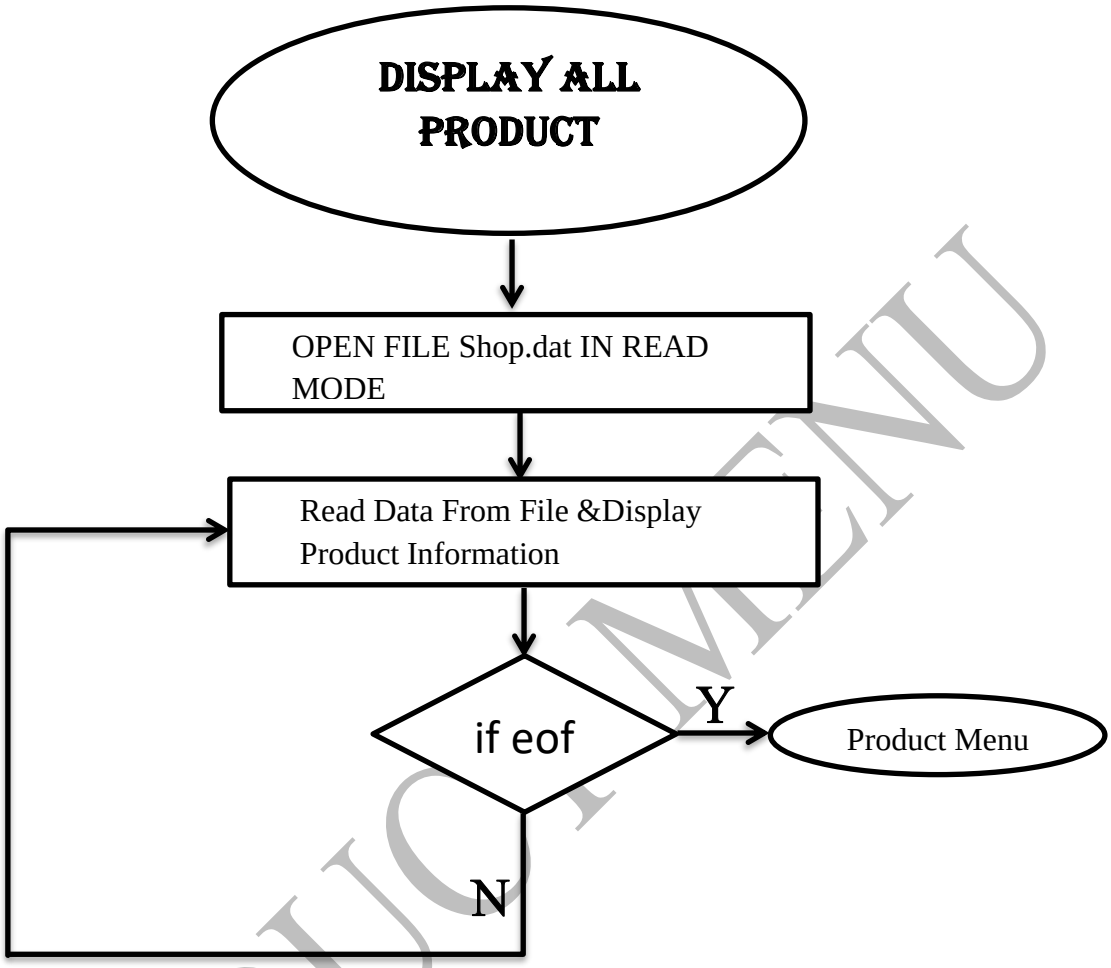

COUNTER
1. PLACE ORDER
2. EXIT
ENTER YOUR CHOICE:
INPUT ch
if
ch==1
Y
PLACE
ORDER
N
if
ch==2
N
Y
MAIN
MENU

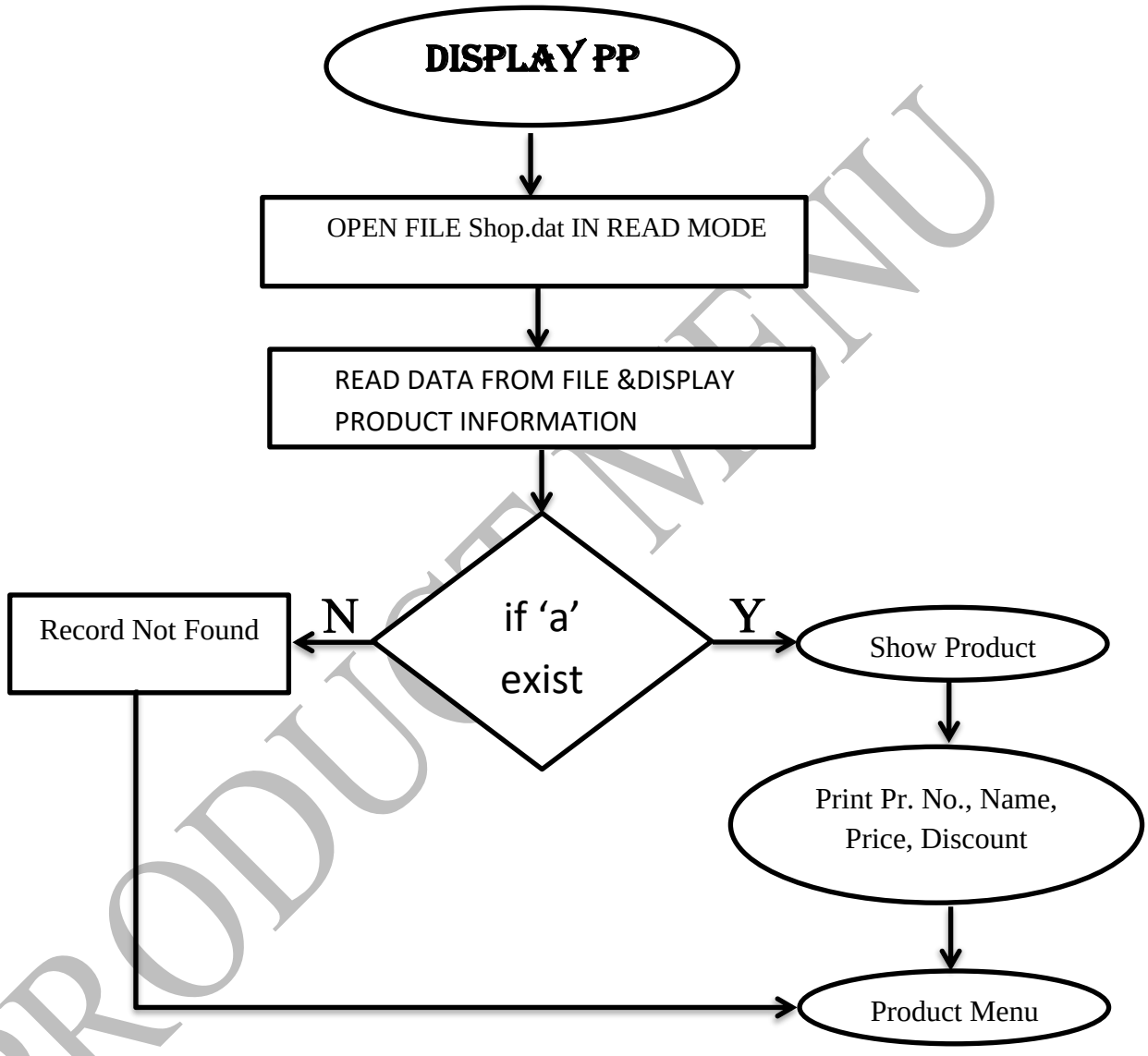

PLACE ORDER
MENU
INPUT PR_NO. & QTY
WANT TO ODER MORE PRODUCT
INPUT ch
Y
if ch=='y' or 'Y'
N
Customer Detail
OPEN Shop.dat IN READ MODE
READ THE PRODUCT INFO. FROM FILE
GENERATE INVOICE
Customer Total amount
COUNTER

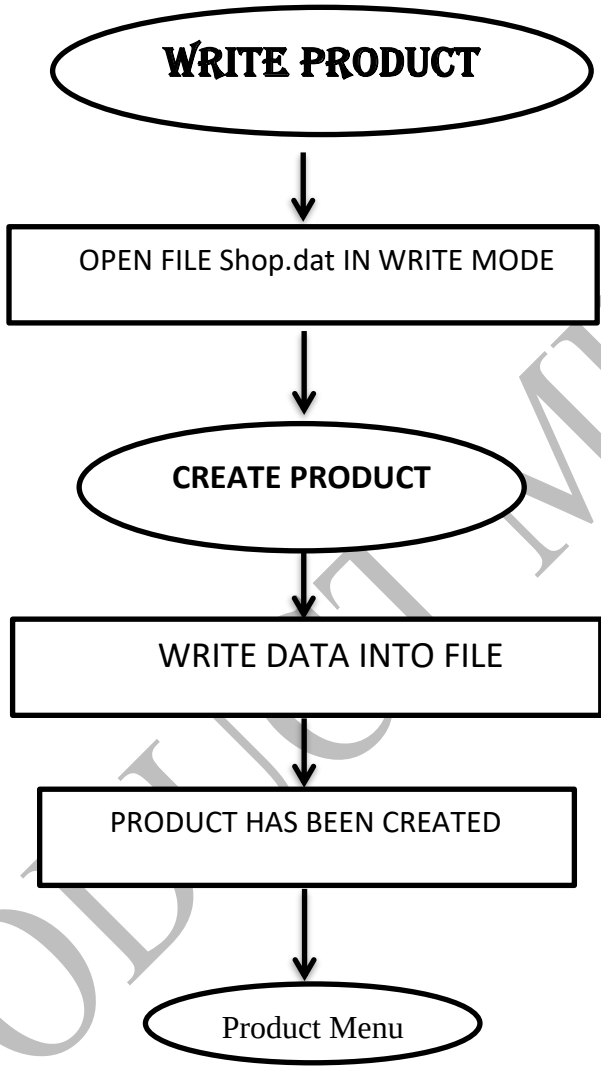


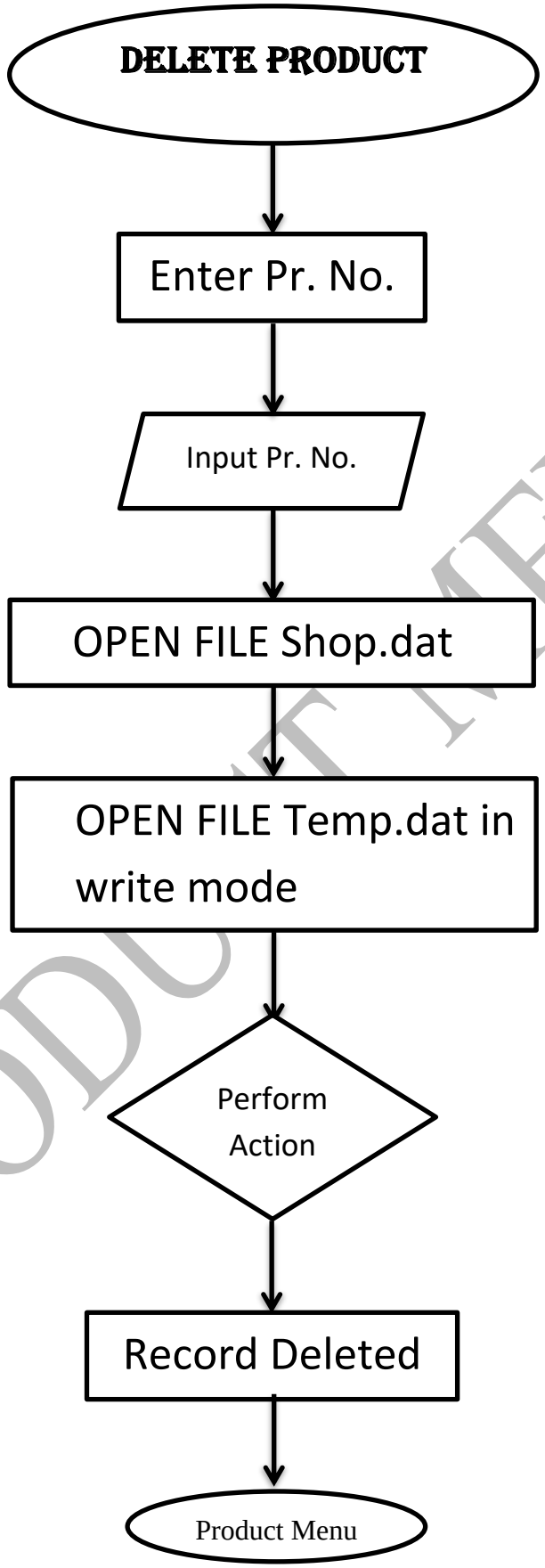

ADMIN MENU
1. PRODUCT
2. USER (CASHIER)
3. REPORT
4. EXIT
ENTER YOUR CHOICE:
INPUT ch
if ch==1
Y
PRODUCT MENU
N
if ch==2
Y
USER MENU
N
if ch==3
Y
REPORT MENU
N
if ch==4
Y
MAIN MENU
N

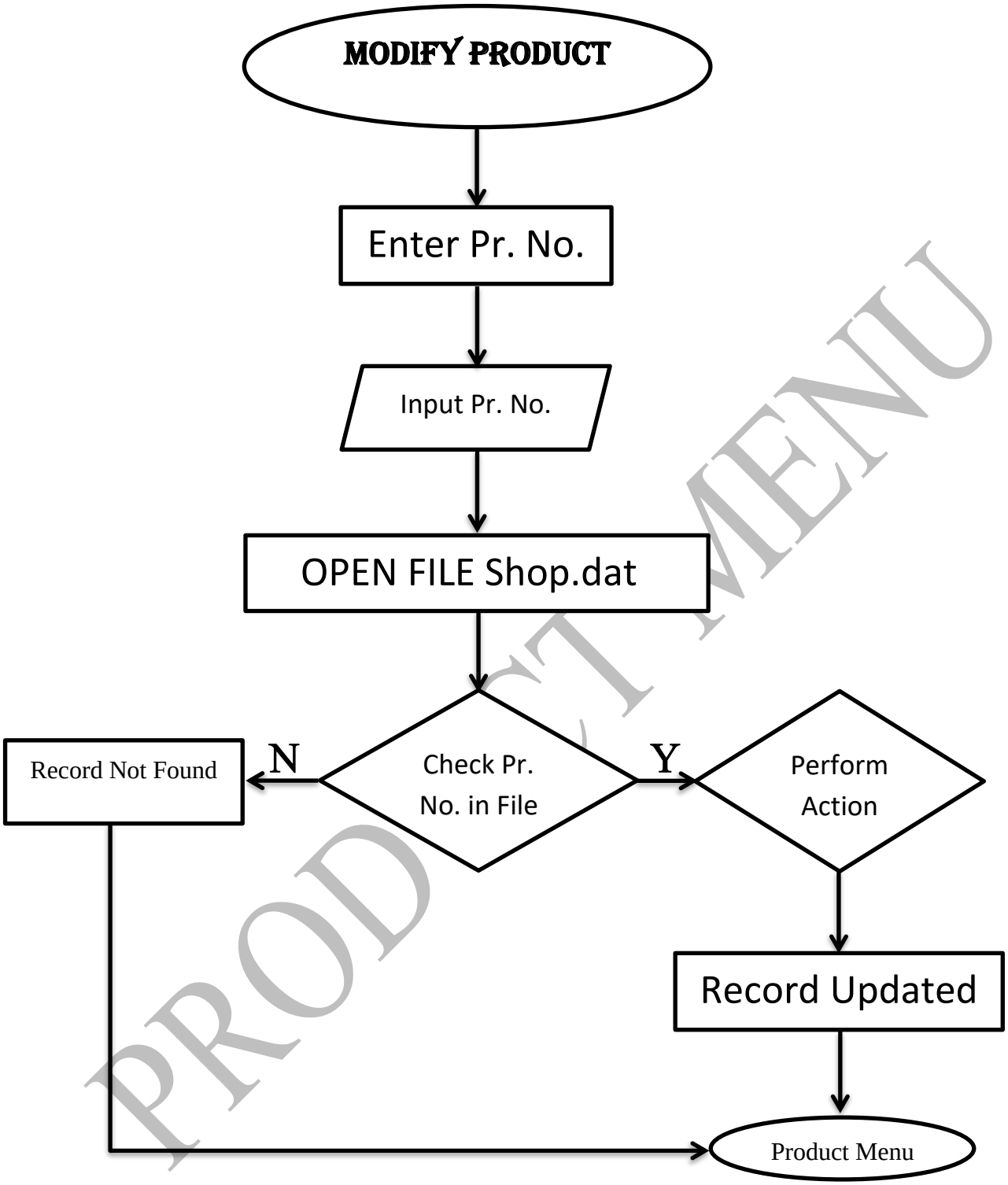


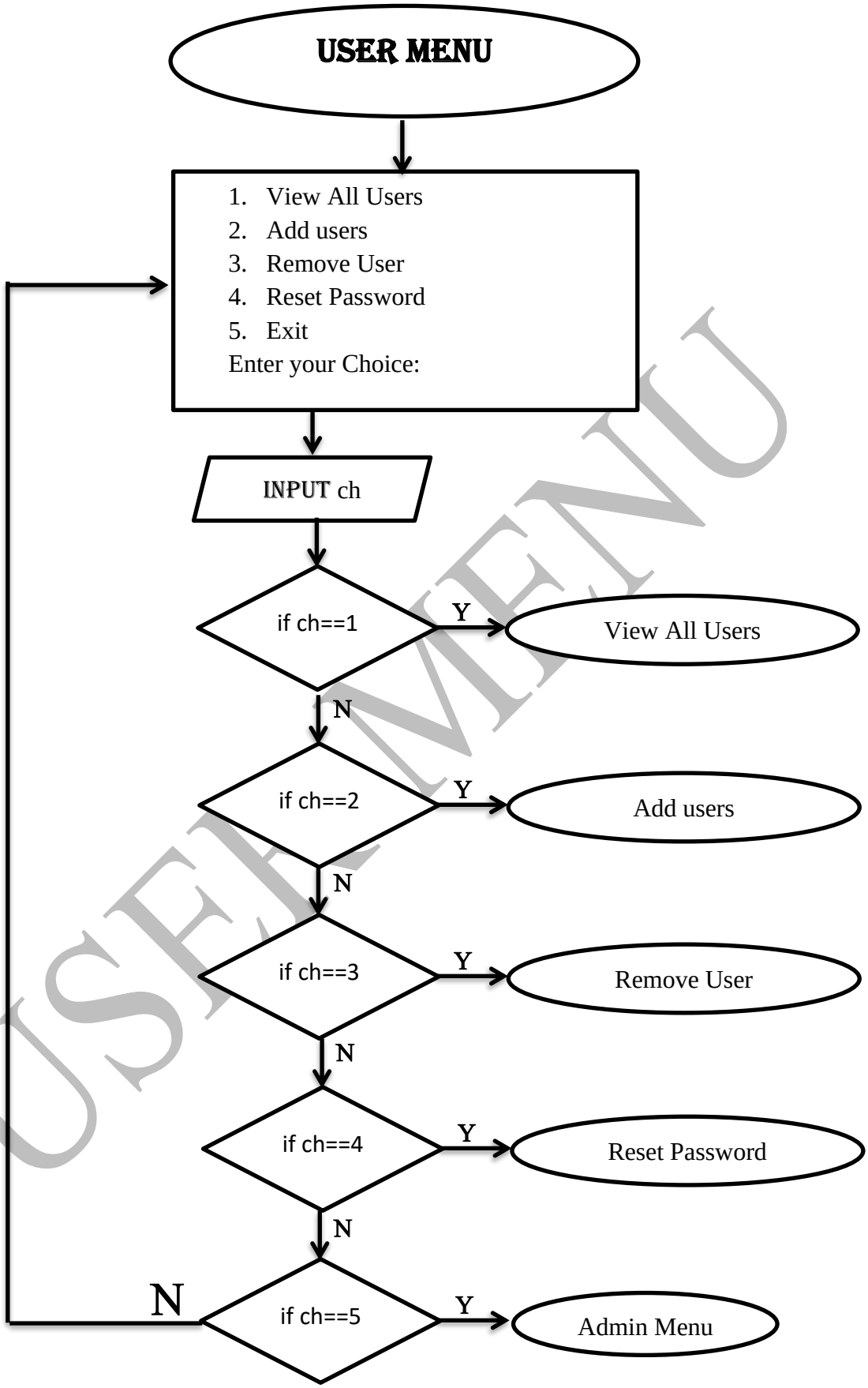


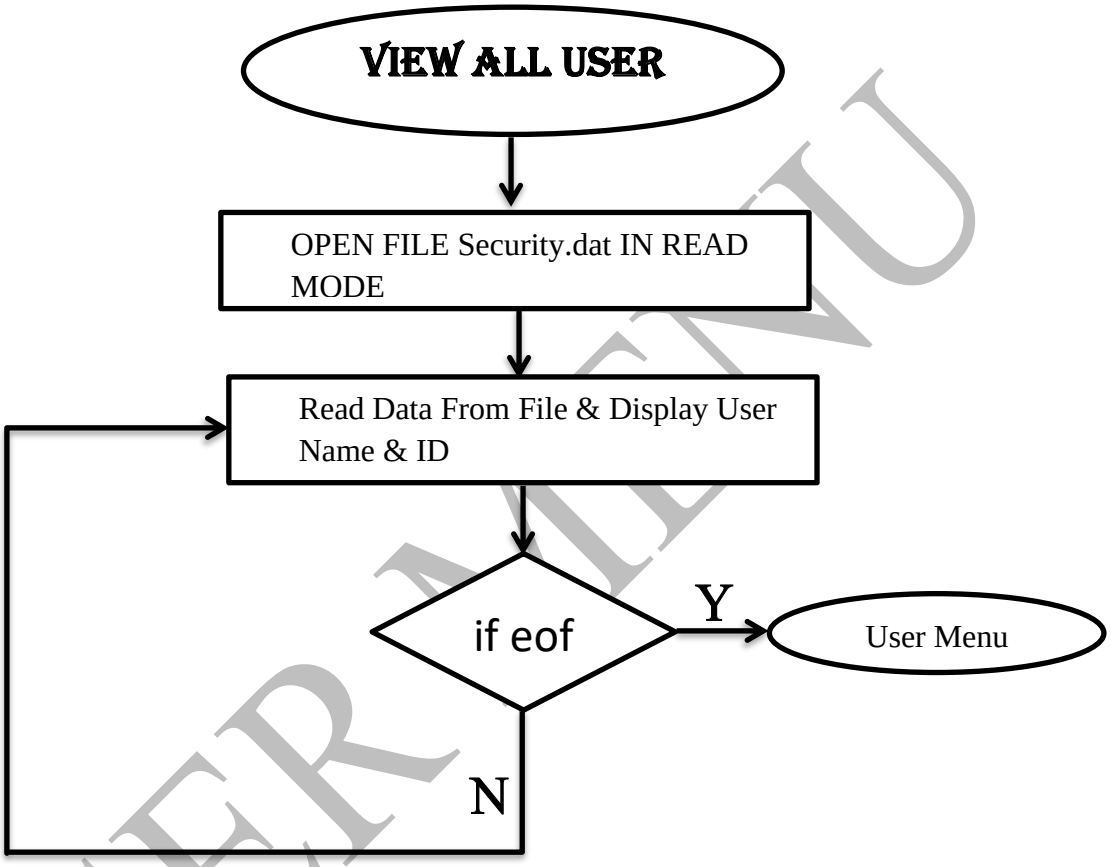


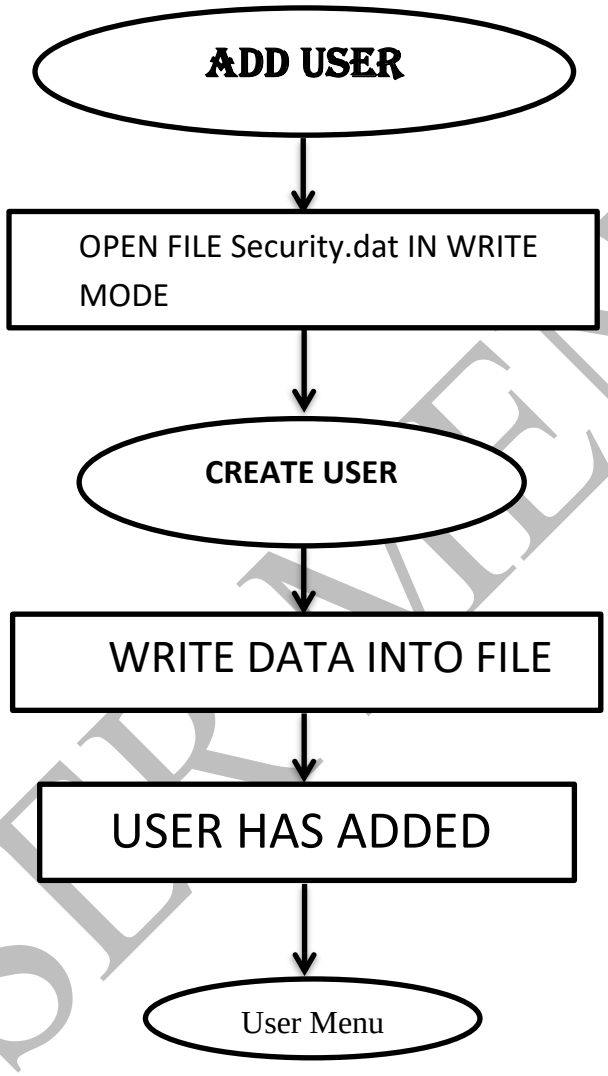


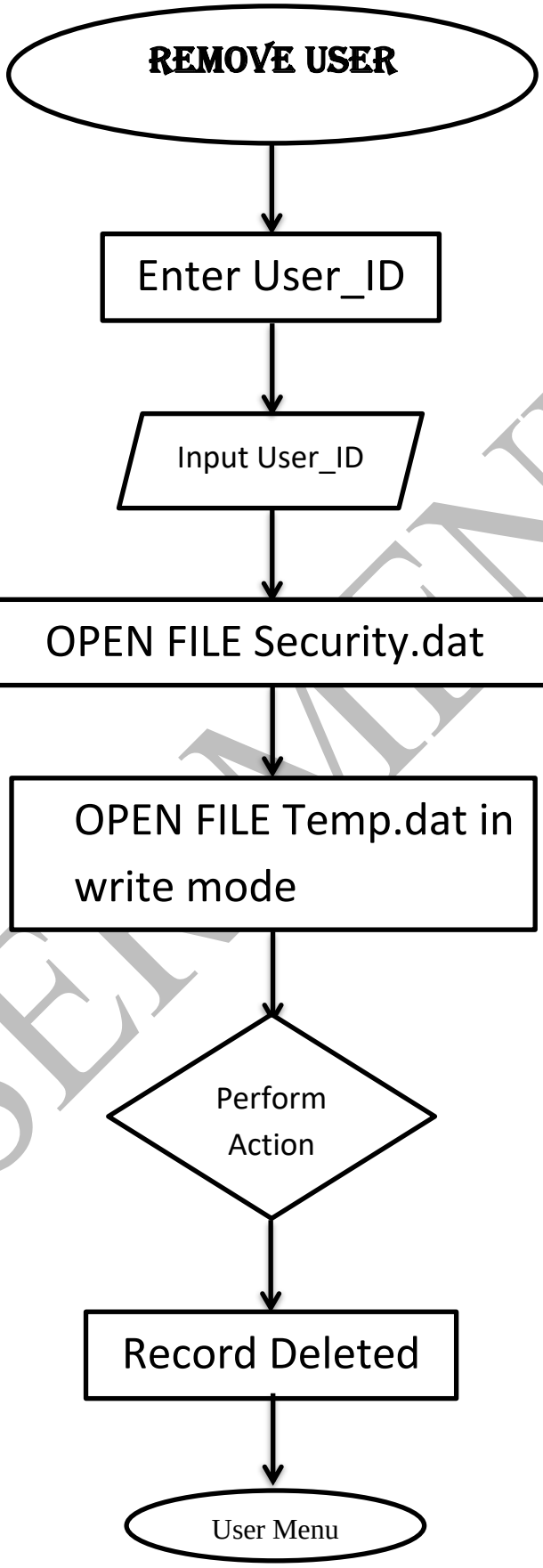


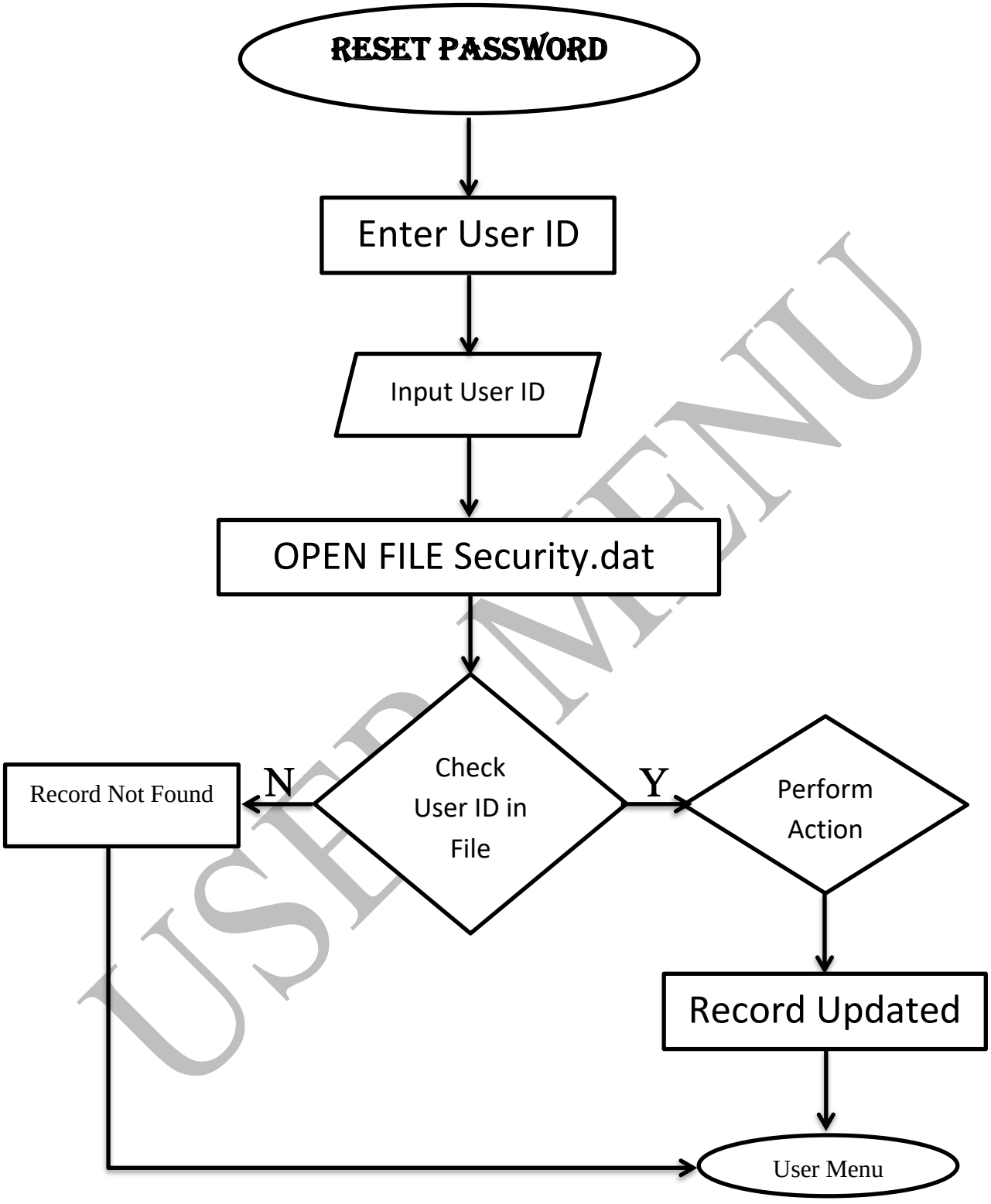


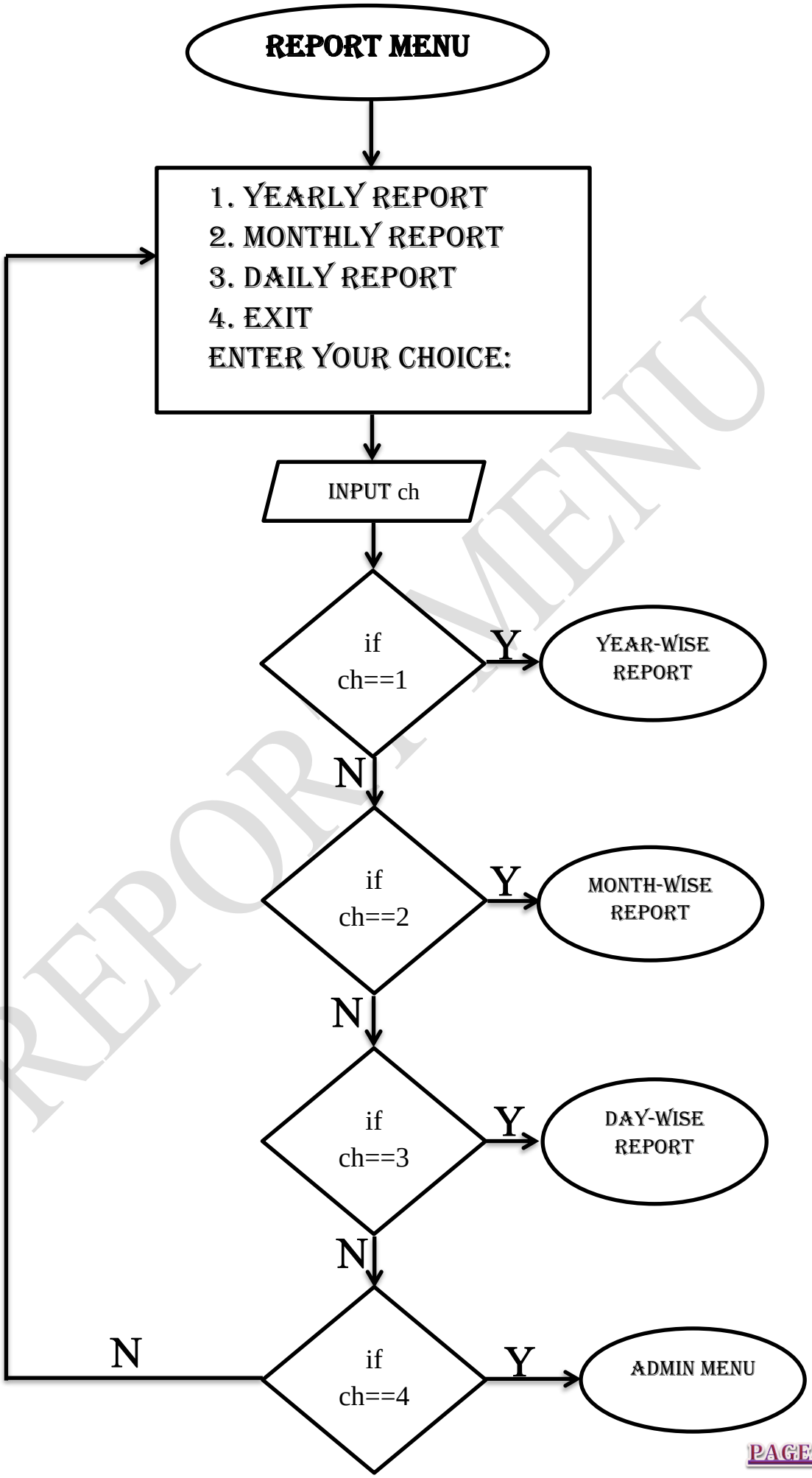

USER MENU
1. View All Users
2. Add users
3. Remove User
4. Reset Password
5. Exit
Enter your Choice:
INPUT ch
if ch==1
Y
View All Users
N
if ch==2
Y
Add users
N
if ch==3
Y
Remove User
N
if ch==4
Y
Reset Password
N
if ch==5
Y
Admin Menu
N

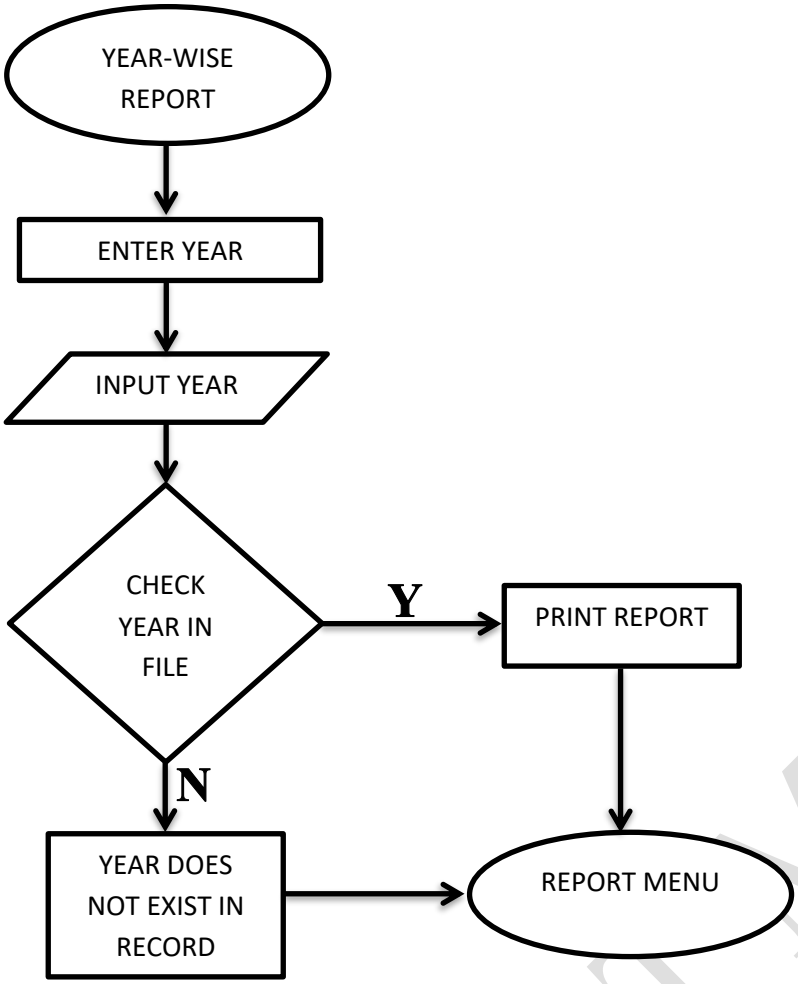

VIEW ALL USER
OPEN FILE Security.dat IN READ MODE
Read Data From File & Display User Name & ID
if eof
Y
User Menu
N

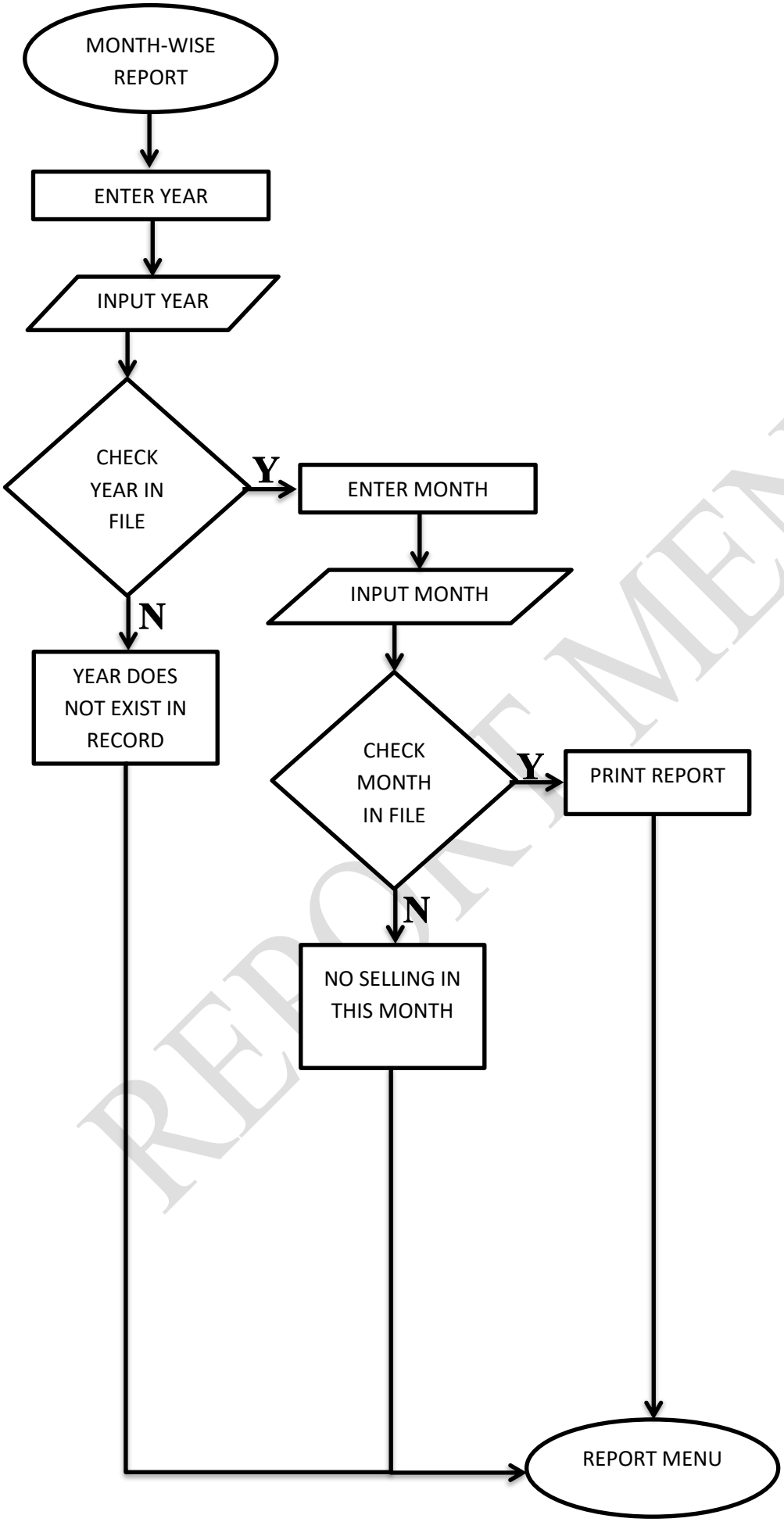

ADD USER
OPEN FILE Security.dat IN WRITE MODE
CREATE USER
WRITE DATA INTO FILE
USER HAS ADDED
User Menu

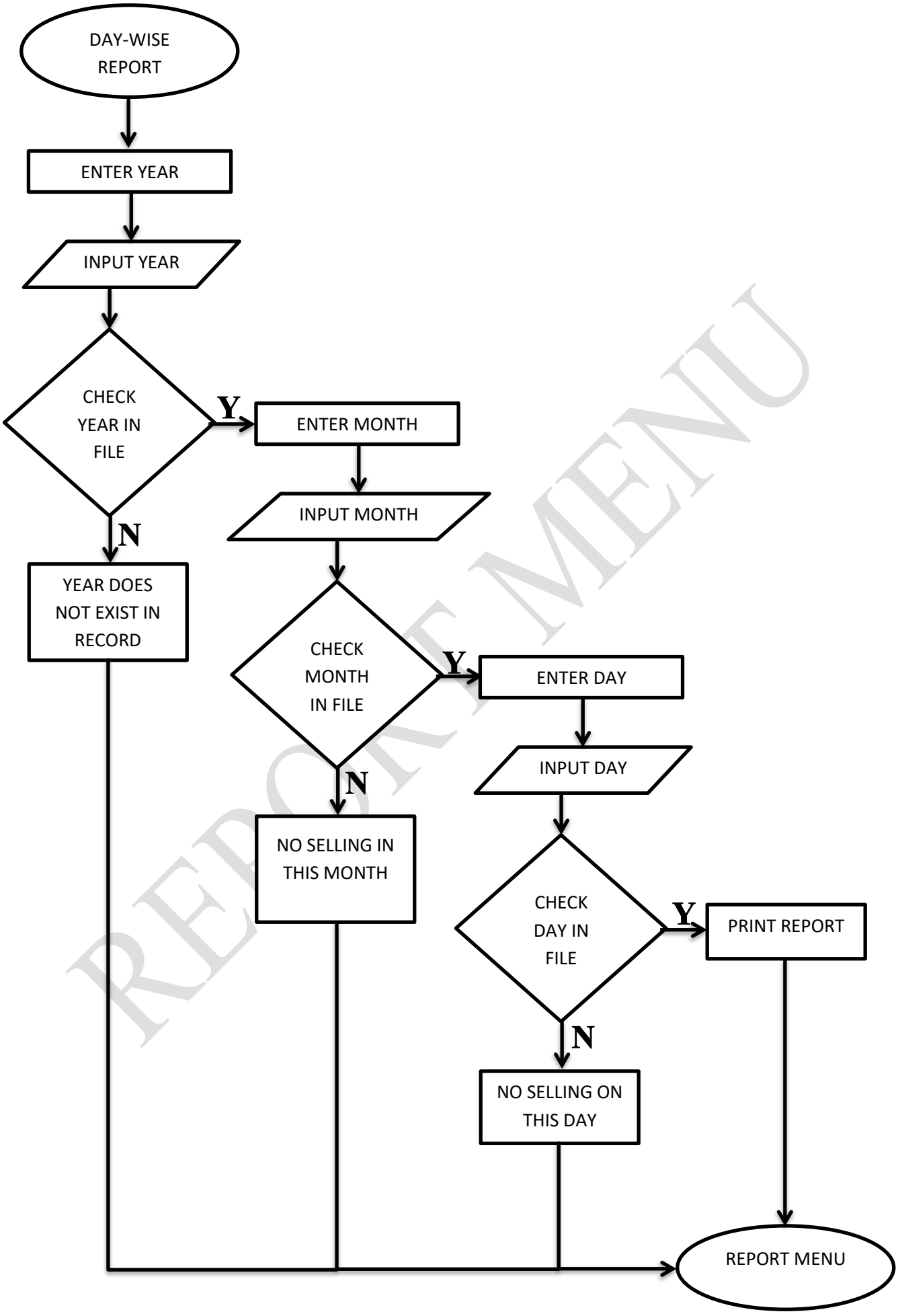

REMOVE USER
Enter User_ID
Input User_ID
OPEN FILE Security.dat
OPEN FILE Temp.dat in write mode
Perform Action
Record Deleted
User Menu


RESET PASSWORD
Enter User ID
Input User ID
OPEN FILE Security.dat
Check User ID in File
N
Y
Record Not Found
Perform Action
Record Updated
User Menu


REPORT MENU
1. YEARLY REPORT
2. MONTHLY REPORT
3. DAILY REPORT
4. EXIT
ENTER YOUR CHOICE:
INPUT ch
if ch==1
Y
YEAR-WISE REPORT
N
if ch==2
Y
MONTH-WISE REPORT
N
if ch==3
Y
DAY-WISE REPORT
N
if ch==4
Y
ADMIN MENU
N


YEAR-WISE REPORT
ENTER YEAR
INPUT YEAR
CHECK YEAR IN FILE
Y
PRINT REPORT
N
YEAR DOES NOT EXIST IN RECORD
REPORT MENU


MONTH-WISE REPORT
ENTER YEAR
INPUT YEAR
CHECK YEAR IN FILE
Y
ENTER MONTH
INPUT MONTH
N
YEAR DOES NOT EXIST IN RECORD
CHECK MONTH IN FILE
Y
PRINT REPORT
N
NO SELLING IN THIS MONTH
REPORT MENU


DAY-WISE REPORT
ENTER YEAR
INPUT YEAR
CHECK YEAR IN FILE
Y
ENTER MONTH
INPUT MONTH
N
YEAR DOES NOT EXIST IN RECORD
CHECK MONTH IN FILE
Y
ENTER DAY
INPUT DAY
N
NO SELLING IN THIS MONTH
CHECK DAY IN FILE
Y
PRINT REPORT
N
NO SELLING ON THIS DAY
REPORT MENU

DATAFLOW DIAGRAM

Data flow diagram is the starting point of the design phase that functionally decomposes the requirements specification. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flow rather than how they are processed, so it does not hardware, software and data structure.

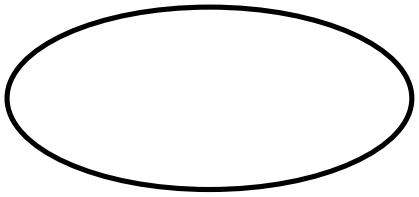
A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

The data flow diagram is a graphical description of a system's data and how to Process transform the data is known as Data Flow Diagram (DFD).

Unlike details flow chart, DFDs don't supply detail descriptions of modules that graphically describe a system's data and how the data interact with the system. Data flow diagram number of symbols and the following symbols are of by demarco.

Symbol:

1. Process



2. data store

3. Source

4. data FLOW

There are seven rules for construct a data flow diagram:

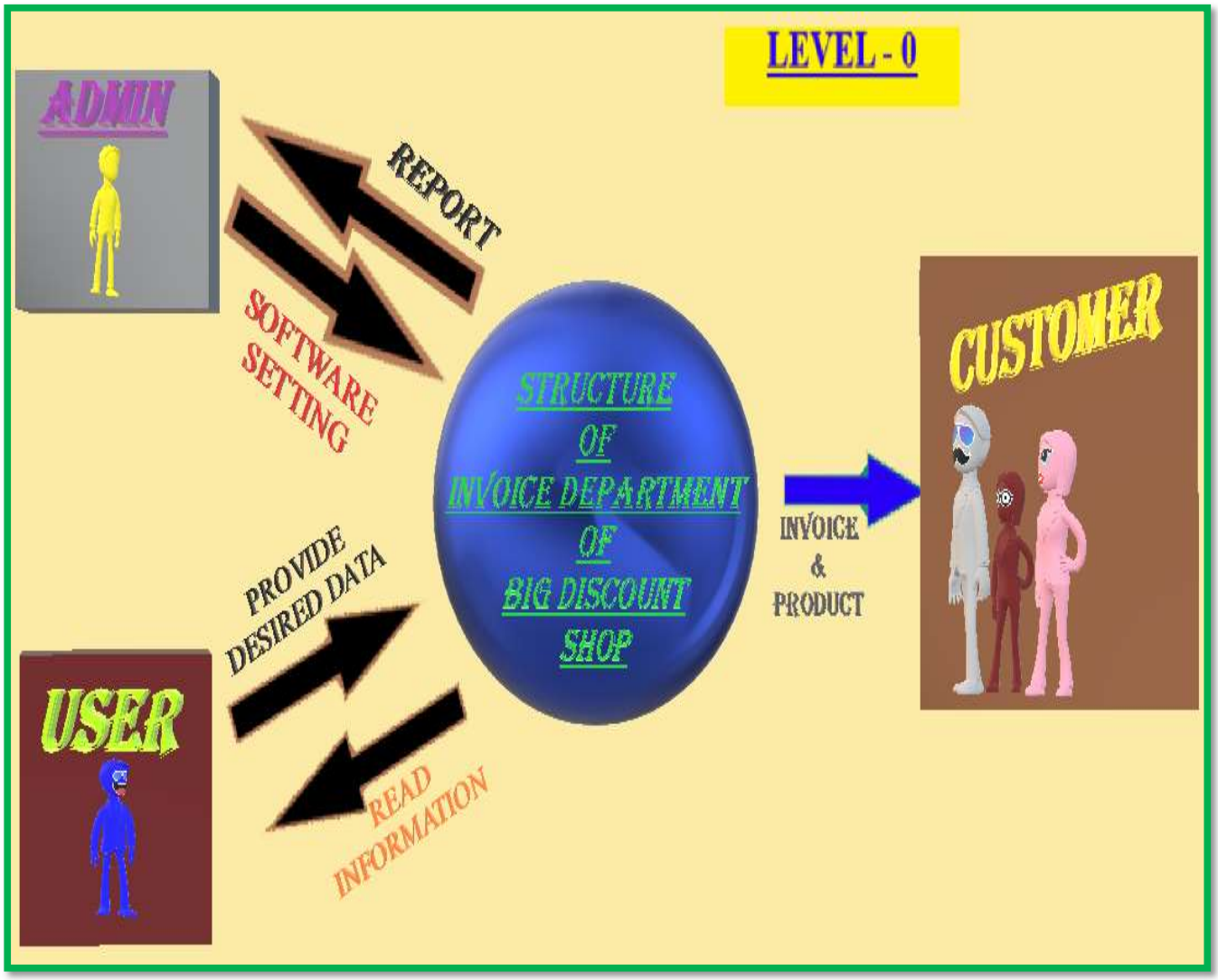
i) Arrows should not cross each other.
ii) Squares, circles and files must wears names.
iii) Decomposed data flows must be balanced.
iv) No two data flows, squares or circles can be the same names.
v) Draw all data flows around the outside of the diagram.
vi) Choose meaningful names for data flows, processes & data stores.
vii) Control information such as record units, password and validation requirements are not penitent to a data flow diagram.

Additionally, a DFD can be utilized to visualize data processing or a structured design.

This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modeled.

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents, which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD'), the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. The level 1 DFD is further spreaded and split into more descriptive and detailed description about the project as level 2 DFD. The level 2 DFD can be a number of data flows which will finally show the entire description of the software project.


LEVEL - 0
ADMIN
REPORT
SOFTWARE SETTING
STRUCTURE OF INVOICE DEPARTMENT OF BIG DISCOUNT SHOP
INVOICE & PRODUCT
CUSTOMER
PROVIDE DESIRED DATA
USER
READ INFORMATION

EXPLANATION OF LEVEL-0 DFD:

In level-0 DFD there are 3 external entities. Named as:

1. Admin
2. User
3. Customer

Work Of "ADMIN" as shown in level-0:

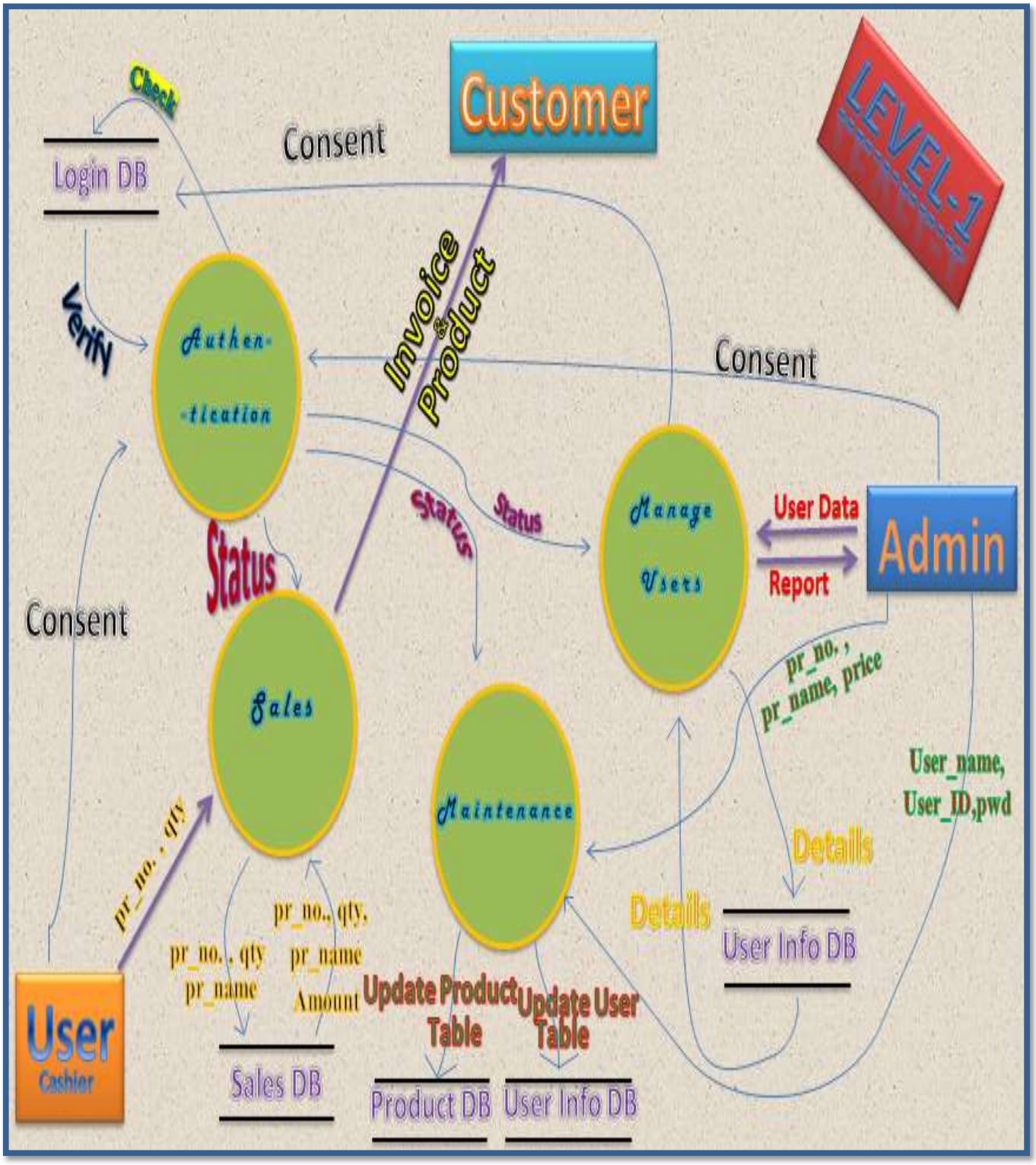
The admin will able to modify the software setting of the system in a way that it can add user, delete user & change the password of user. The admin will also able to fetch the report in the form that how many users are there in database. If any user has forgotten their password then it will contact to admin to change their password. The admin will also able to fetch the product list and can change the information of particular product.

Work Of "USER" as shown in level-0:

Here the cashier is named as user. The work of the user is read the information of that product which customer is buying from store and provides it to the invoice software to create the invoice.

Work Of "CUSTOMER" as shown in level-0:

Here the work of the customer is to put that entire product in the basket which they want to purchase and handover it to the invoice department's user to create the invoice. After getting the invoice, the customer will pay the invoice amount and take their product.


LEVEL-1
Customer
Check
Login DB
Consent
Verify
Authen-tication
Invoice & Product
Consent
Status
Status
Status
Manage Users
User Data
Report
Admin
Consent
Sales
Maintenance
pr_no. , price
pr_name
User_name,
User_ID,pwd
Details
Details
User Info DB
pr_no. , qty
pr_no., qty,
pr_name
Amount
pr_no. , qty
pr_name
Update Product Table
Update User Table
User
Cashier
Sales DB
Product DB
User Info DB

EXPLANATION OF LEVEL-1 DFD:

In level-1 DFD there are 3 external entities with 4 processes and 4 databases. Named as:

External Entity:

1. Admin
2. Users
3. Customer

Process:

1. Authentication
2. Sales
3. Maintenance
4. Manage Users

Database:

1. Login
2. Sales
3. Product
4. Users info

Work of Processes:

1. Work Of "AUTHENTICATION PROCESS" as shown in level-1:

   The authentication process is deal with security of software. It helps the admin to secure the whole system in the way that only admin and that user (cashier) whom the admin has provided user ID & PASSWORD will only use and interact with software.

2. Work Of "SALES PROCESS" as shown in level-1:

   The sales is a process which acts like a mediator. It takes the information from user and sends it to the different database to create invoice to sell the product legally.

3. Work Of "MAINTENANCE PROCESS" as shown in level-1:

   The maintenance is the process which helps the admin to update the product database and user database, i.e. The admin will able to add, delete or modify the data in product database as well as user database.

4. Work Of "MANAGE USERS PROCESS" as shown in level-1:

   The manage users is the process which help the admin to fetch the all information about any particular user (cashier).

Work of Databases:

1. Work Of "LOGIN DATABASE" as shown in level-1:

   In login database there is user ID & PASSWORD of each user's has saved and it's work is to check and verify the information which has entered by the each user in this software including admin also.

2. Work Of "SALES DATABASE" as shown in level-1:

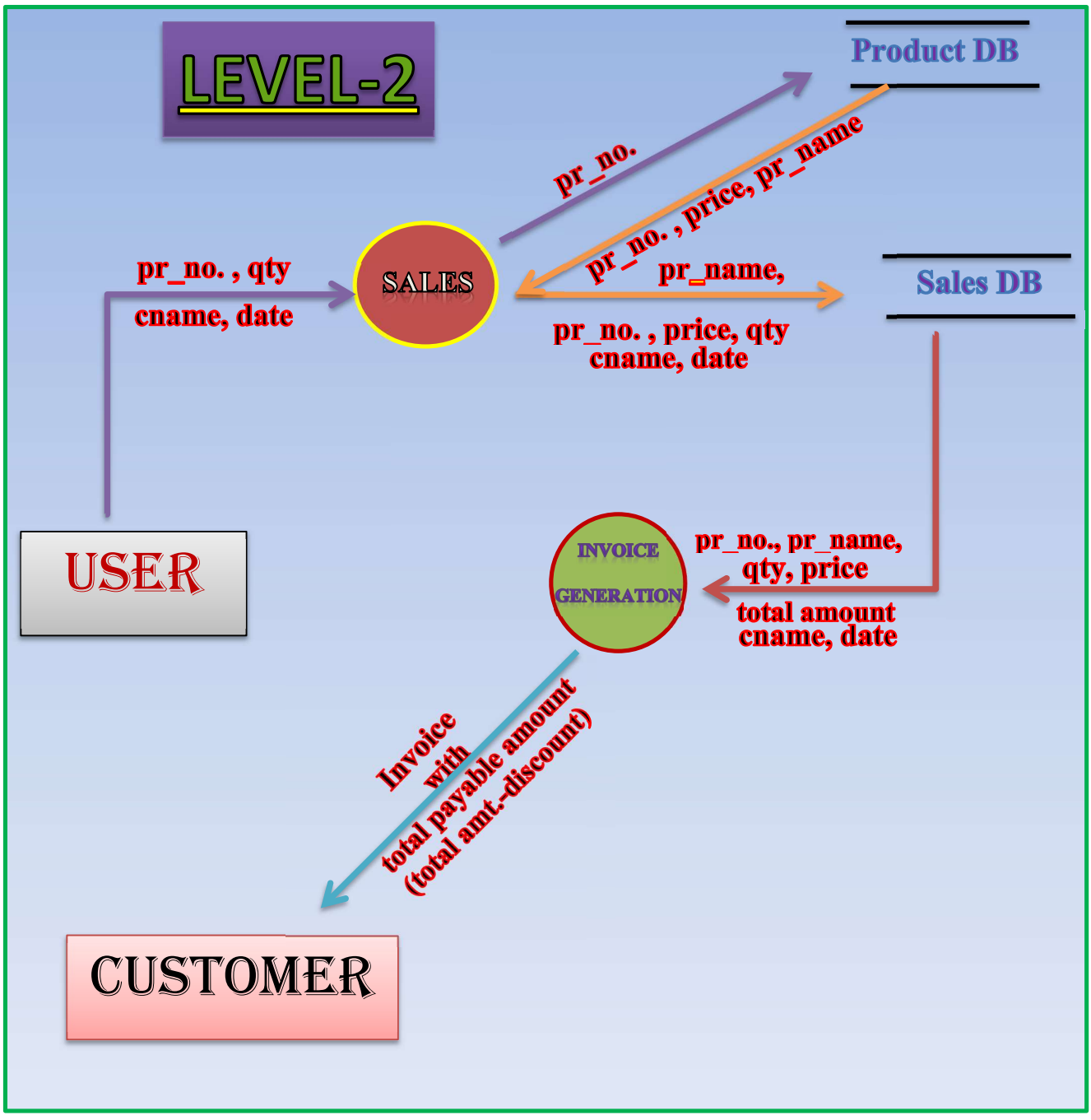
   It contains all the information about sales activity, i.e. how they have to generate the bill, how software have to manage discount in bill. After doing all these activity it gives invoice which show that how much amount the customer have to pay regarding their purchasing product.

3. Work Of "PRODUCT DATABASE" as shown in level-1:

   In product database all the information of each product is stored. When admin want to do any modification in existing database content like add product or delete product or modify product then admin have to contact with maintenance process and then it will contact with product database to make desired changes in it.

4. Work Of "USER INFO DATABASE" as shown in level-1:

   It contains each information like user's name, user ID & PASSWORD of each user (cashier). It helps the admin to fetch all information of each user and also help to modify it through maintenance process.


LEVEL-2
Product DB
pr_no.
pr_no. , price, pr_name
pr_no. , qty
cname, date
SALES
pr_name,
Sales DB
pr_no. , price, qty
cname, date
USER
INVOICE
GENERATION
pr_no., pr_name,
qty, price
total amount
cname, date
Invoice
with
total payable amount
(total amt.-discount)
CUSTOMER

EXPLANATION OF LEVEL-2 DFD:

In level-1 DFD there are 2 external entities with 2 processes and 2 databases. Named as:

External Entity:

4. Users
5. Customers

Process:

5. Sales
6. Bill Generation

Database:

5. Product
6. Sales

Work of External Entities:

1. Work of "USER ENTITY" as shown in level-2:

The work of user entity is to take product number (pr_no.) & quantity (qty) from product's tag and provide it to sales process.

2. Work of "CUSTOMER ENTITY" as shown in level-2:

The work of customer entity is to receive the invoice (bill) and pay the payable amount to the user (cashier) and take the product.

Work of Processes:

1. Work of "SALES PROCESS" as shown in level-2:

The work of sales process is to read data i.e. product number (pr_no.) & quantity (qty) from user and send it to product database and after that again read the data i.e. product number (pr_no.), price & product name (pr_name) from the product database and send these data along with quantity (qty) to the sales database.

2. Work of "BILL GENERATION PROCESS" as shown in level-2:

The work of bill generation process is to fetch the data i.e. product number (pr_no.), product name (pr_name), quantity (qty), price, & total amount from the sale database. Then subtract the discount from total amount to get the total payable amount. After performing above process it will print the invoice.

Work of Databases:

1. Working of "PRODUCT DATABASE" as shown in level-2:

It will fetch the product number (pr_no.) and provide all the information like price & product name (pr_name) of that product number.

2. Working of "Product database" as shown in level-2:

It working is to fetch the data like product number (pr_no.), product name (pr_name), price, quantity (qty) from the sales process and find total amount by multiplying the number quantity (qty) with its price of particular product number.

ENTITY - RELATIONSHIP DIAGRAM

Entity Relationship Diagram (E-R Diagram):

E-R Model is a popular high level conceptual data model. This model and its variations are frequently used for the conceptual design of database application and many database design tools employ its concept.

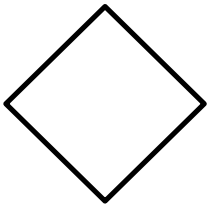
A database that confirms to an E-R diagram can be represented by a collection of tables in the relational system. The mapping of E-R diagram to the entities is:

- Attributes
- Relations
  - Many-to-many
  - Many-to-one
  - One-to-many
  - One-to-one
- Weak entities
- Sub-type and super-type

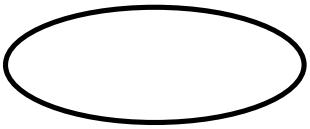
The entities and their relationships between them are shown using the following conventions.

- An entity is shown in rectangle.

- A diamond represents the relationship among number of entities.



- The attributes shown as ovals are connected to the entities or relationship by lines.



- Diamond, oval and relationships are labeled.

- Model is an abstraction process that hides super details while
  highlighting details relation to application at end.

- A data model is a mechanism that provides this abstraction for database application.

- Data modeling is used for representing entities and their relationship in the database.

- Entities are the basic units used in modeling database entities can have concrete existence or constitute ideas or concepts.

- Entity type or entity set is a group of similar objects concern to an organization for which it maintains data.

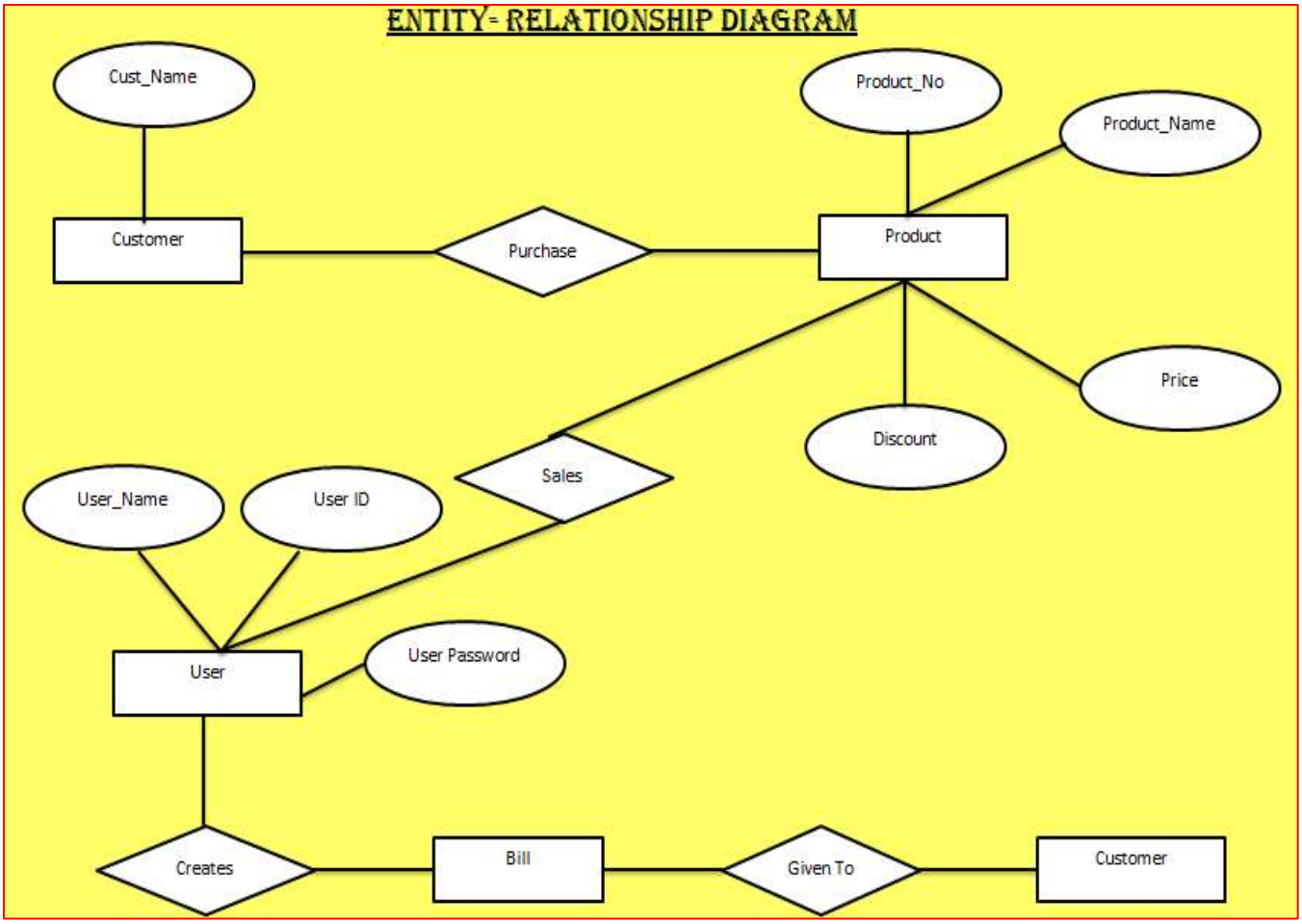
- Properties are characteristics of an entity also called as attributes.

- A key is a single attribute or combination of 2 or more attributes of an entity set is used to identify one or more instances of the set.

- In relational model we represent the entity by a relation and use tuples to represent an instance of the entity.

- Relationship is used in data modeling to represent in association between an entity set.

- An association between two attributes indicates that the values of the associated attributes are independent.


ENTITY- RELATIONSHIP DIAGRAM
Cust_Name
Product_No
Product_Name
Customer
Purchase
Product
Price
Discount
Sales
User_Name
User ID
User
User Password
Creates
Bill
Given To
Customer

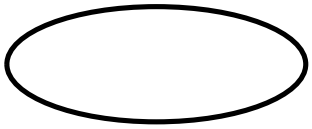
EXPLANATION OF E-R DIAGRAM:

In E-R diagram I have used 3 symbols.

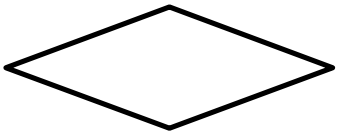
Named as:

1. Entity symbol :

2. Attribute Symbol :



3. Relationship Symbol :



Name used in E-R diagram:

1. In Entity:

    i) Customer
    ii) Product
    iii) User

2. In Attribute:

    i) Cust_Name
    ii) Product_No.
    iii) Product_Name
    iv) Price
    v) Discount
    vi) User_Name
    vii) User ID
    viii) User Password

3. In Relationship:

    i) Purchase
    ii) Sales
    iii) Given to

1. ‘Customer’ is an entity which contain one attribute named as customer name (cust_name) and connect to the ‘product’ entity because of ‘purchase’ relationship.

2. ‘Product’ is an entity which contain four attributes named as product number (Product_No), product name (Product_Name), price, discount and connected with ‘user’ entity because of ‘sales’ relationship.

3. ‘User’ is an entity which contains three attributes named as user name (User_Name), User ID, User Password and connected to the ‘invoice’ entity with the help of ‘creates’ relationship.

4. Finally, the ‘invoice’ entity has no attributes and has connected with ‘customer’ by the relationship of ‘given to’.

FUTURE SCOPE

Every project whether large or small has some limitations no matter however diligently developed. In some cases limitations is small while in other cases they may be broad also. The new system has got some limitations.

Major areas where modifications can be done are as follows:

- Our system is not online so further it can be improved.
- The security is limited so some additional arrangement could be made to provide more security to the system.
- We can add inventory management.
- We can give more advance software for Shop Invoice System including more facilities.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.
- We can manage employee salaries.