

GURU GHASIDAS VISHWAVIDYALAYA

A CENTRAL UNIVERSITY



SESSION : 2019-20

Major Project

On

Billing Management System

Guided By :

Dr. Vikas Pandey

Assistant Professor

Dept. of CSIT

Developed By :

Manikant Mahto

Roll no. 17207232

Dept. of CSIT

GURU GHASIDAS VISHWAVIDYALAYA

A CENTRAL UNIVERSITY



ACKNOWLEDGEMENT

I am extremely grateful and remain indebted to my guide for being a source of inspiration and for his constant support in the Design, Implementation and Evaluation of the project. I am thankful to him for the constant constructive criticism and invaluable suggestions, which benefitted me a lot while developing the “Billing Management System”. He has been a constant source of inspiration and motivation for hard work. He has been very cooperative throughout this project work. Through this column, it would be my utmost pleasure to express my warm thanks to him for encouragement, cooperation and constant help without which I might not be able to accomplish this project. I am also expressing my gratitude towards him for providing me the infrastructure to carry out the project and to all staff members who were directly and indirectly indulged in enabling me to stay committed for project.

Manikant Mahto

GURU GHASIDAS VISHWAVIDYALAYA

A CENTRAL UNIVERSITY



CERTIFICATE BY GUIDE

This is to certify that “Manikant Mahto” of B.Sc. VI Sem, from dept. of Computer Science and Information and Technology, has satisfactorily completed the major project on “Billing Management System” in partial fulfilment of the requirement for the Degree of Bachelor in Computer Science and Information Technology of this university.

I am forwarding this project to the university to partially fulfil the requirements of the degree of B.Sc. VI Semester (C.S.I.T.), during the academic session 2019-20.

APPROVED BY:

Dr. Vikas Pandey

Assistant Professor

Dept. of C.S.I.T.

GURU GHASIDAS VISHWAVIDYALAYA

A CENTRAL UNIVERSITY



CERTIFICATE BY EXAMINER

This certifies that “Manikant Mahto” of B.Sc. VI Semester, from dept. of Computer Science and Information and Technology, Guru Ghasidas University, Bilaspur (C.G.) has submitted the project report on “Billing Management System” in partial fulfilment of the requirement for the Degree of Bachelor in Computer Science and Information Technology of the university during the academic session 2019-20.

INTERNAL EXAMINER

Mr. Prashant Vaishnav

Assistant Professor

Dept. of CSIT, GGU

GURU GHASIDAS VISHWAVIDYALAYA

A CENTRAL UNIVERSITY



DECLARATION

I Manikant Mahto hereby declare that the major project entitled “Billing Management System” is a record research work done by me under the guidance of Dr. Vikas Pandey, Assistant Professor, dept. of C.S.I.T., GGU, Bilaspur and it has not formed the basis of an award or degree or diploma but it is partially fulfilling the requirement for B.Sc. VI Sem.

Manikant Mahto

RollNo: 17207232

B.Sc.VI semester

Dept. of C.S.I.T

ABSTRACT

Electronic billing improves efficiency and saves time while also reducing error rates. While time-saving is a major benefit, the reduction of manual labour on monotonous tasks allows finance employees to focus their time on more strategic activities. The shift from paper-based to electronic bills also inherently creates better organization and results in fewer errors. E-billing adds efficiency to multiple business units, ranging from accounting to customer service.

Traditional billing and payment methods relied heavily on paper and checks. This process made organization a difficult task with no clear system of record (SOR).

Modern finance and accounting applications allow for digital SORs to exist for both the payer and the payee. Billing and invoicing systems often integrate directly with payment platforms, furthering the ability to keep a clean, organized SOR.

INDEX

Content	Page no.
Introduction	8
Description of existing system	8
Description of project	8
Methodology of the project	9
Software requirements specification	12
Software design document	12
System Requirement Analysis	13
Data Source	14
Fact Finding Techniques	14
Feasibility Study	15
Minimum System Requirements	16
Back End Description	17
Design : Flow Chart	20
Design : Data Flow Diagram	23
Design : E-R Diagram	25
Description of Modules	26
Testing	52
Future Scope	53
Conclusion	53
Bibliography	54

Billing Management System

Introduction

The 'Store Management System' is targeted to automate the almost all of the processes mentioned above to reduce the clerical labour of the staff working in Stores both technical and as well as Accounts departments using the software Industry's latest technologies and cost effective tools there by providing the better control to the management by avoiding manual errors etc.,

Description of existing system

The existing system is quite slow and inefficient. Perhaps, it is more prone to errors. This type of system mainly depends on the effectiveness of the employee working which makes this system a slower process.

Drawbacks:

1. Slow
2. Inefficient
3. Prone to errors
4. Ineffective
5. Less secure

Description of project

The project consists of all the required modules to increase the speed and efficiency of the user. It also helps in making the calculations error free. The details of the entire invoice can be seen in a detailed format. The list of all the stocks is also displayed in a tabular form.

Main features:

1. Faster than existing way
2. Error free
3. Records saved in digital format
4. Tabular format
5. High level of security

Objectives of the project:

The main aim of the project is to decrease the human effort to do all the required work such as maintain record of all the product/stocks , billing details , creation of new bill etc.

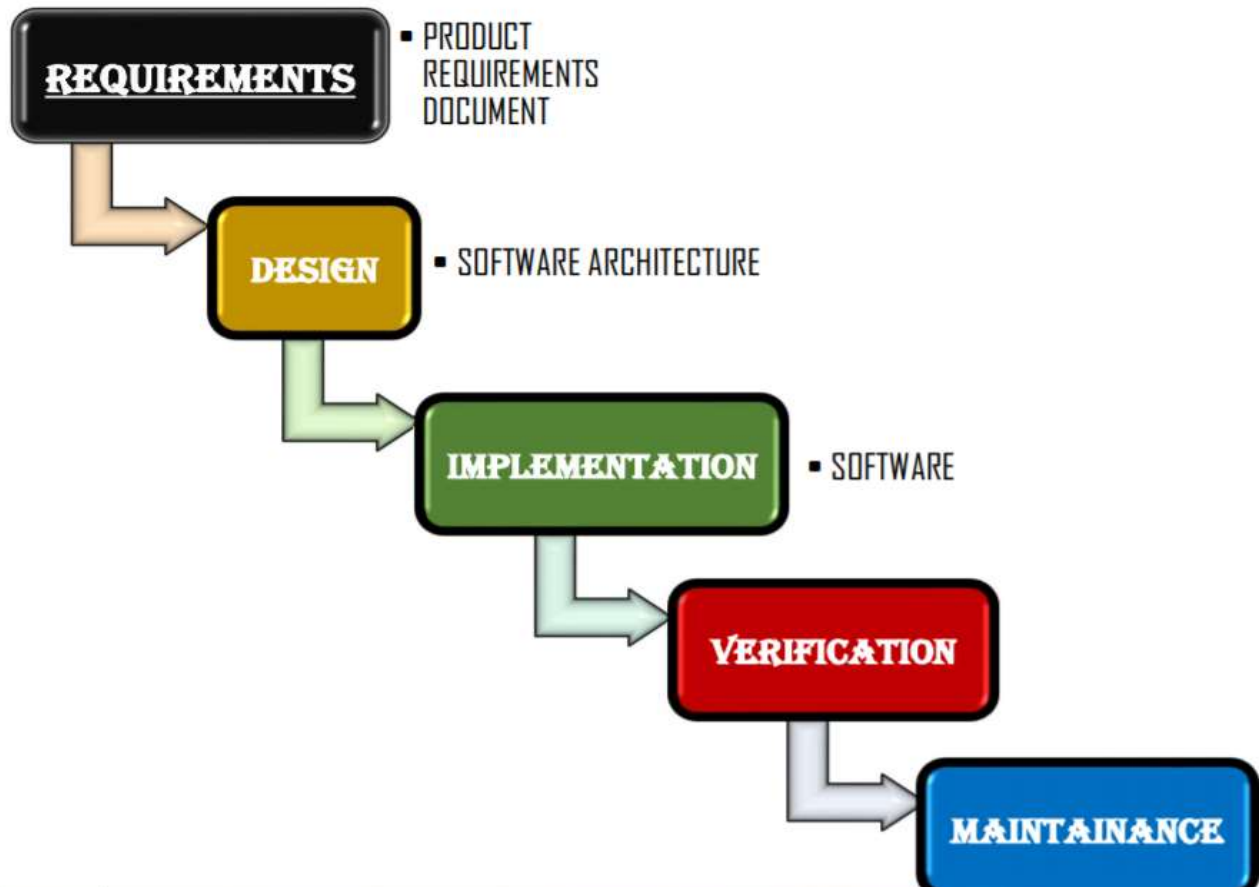
Methodology of the project

The software development life cycle model that we used for developing this project is Waterfall Model.

WATERFALL MODEL

The Waterfall Model was the first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest SDLC approach that was used for software development. In “The Waterfall” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is completed. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a Linear-Sequential Life Cycle Model.

SEQUENTIAL PHASES IN WATERFALL MODEL



Requirements:

The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

System Design:

The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

Implementation:

With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

Integration and Testing (Verification):

All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.

Maintenance:

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name “Waterfall Model”.

Advantages of the Waterfall Model:

1. The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
2. The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
3. It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

4. In this model, phases are processed and completed one at a time and they do not overlap. Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of Waterfall Model:

1. It is difficult to estimate time and cost for each phase of the development process.
2. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage
3. Not a good model for complex and object-oriented projects.
4. Not suitable for the projects where requirements are at a moderate to high risk of changing.

Software requirements specification

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. An SRS describes the functionality the product needs to fulfil all stakeholders (business, users) needs. A typical SRS includes: A purpose of an overall description. It's the output of the Requirement phase.

Software design document

Software design document or SDD; just design document; also, Software Design Specification) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. It's the output of the Design phase.

System Requirement Analysis

In software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Information Gathering

Methods:

In order to determine the requirements of a system, information must be gathered from the customer. Ideally, the information obtained will enable a well-defined, accurate, and complete description of how the business functions as well as the people, functions and data involved. However, this is not always the case, and information is often misinterpreted or omitted entirely. There are many techniques that can be employed when gathering information. The type of information you are trying to obtain, as well as the people providing the information, will determine which techniques you should use.

Traditional Methods of Gathering Information

Traditional methods of gathering information include:

1. Interviews
2. Questioning
3. Questionnaires
4. Observation
5. Study of existing organizational documents, forms and reports

Traditional methods are often used when the overall objective is clear and requirements are well defined. In addition, methods such as questioning and interviewing are generally regarded as the only choice for particular phases such as the initial contact.

Modern Methods of Gathering Information

Modern methods of gathering information include:

1. JAD
2. RAD
3. Prototyping

Modern methods are often used when the requirements and objectives are not clearly understood. Modern methods of gathering information require a greater effort, but often the results are much faster and more accurate than those gathered by traditional methods.

Data Source

The data source of this project is faculty of the store. Internet is used to get all the desired information for the proper development of the system. Paper works done by the institution head for record maintenance.

Fact Finding Techniques

Fact finding is process of collection of data and information based on techniques which contain sampling of existing documents, research, observation, questionnaires, interviews, prototyping and joint requirements planning. System analyst uses suitable fact-finding techniques to develop and implement the current existing system. Collecting required facts are very important to apply tools in System Development Life Cycle because tools cannot be used efficiently and effectively without proper extracting from facts. Fact-finding techniques are used in the early stage of System Development Life Cycle including system analysis phase, design and post implementation review. Facts included in any information system can be tested based on three steps: data- facts used to create useful information, process- functions to perform the objectives and interface- designs to interact with users.

There are seven common fact-finding techniques:

1. Sampling of existing documentation, forms and databases

2. Research and Site visits
3. Observation of the work environment
4. Questionnaires
5. Interviews
6. Prototyping
7. Joint requirements planning

Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institution. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

The acronym TELOS refers to the five areas of feasibility:

Technical: This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project.

Economic: It defines the availability of all the requirements under the desired expenses.

Legal: It determines whether the proposed system conflicts with legal requirements, e.g., a data processing system must comply with the local data

protection regulations and if the proposed venture is acceptable in accordance to the laws of the land.

Operational: Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Scheduling: A time feasibility study will take into account the period in which the project is going to take up to its completion. A project will fail if it takes too long to be completed before it is useful.

Minimum System Requirements

Hardware Configuration:

- A computer (i.e. a laptop or a desktop)
- Processor: Intel Core Processor
- RAM : 2 GB
- Hard Disk: 256 GB
- Printer (for printing bills)

Software Configuration:

- Back End: Microsoft SQL Server Management Studio 18 (SQL Database)
- Environment : Microsoft Visual Studio 2019
- Language : VB.NET
- Operating System : Microsoft Windows 10 (Home/ Professional/ Education/ Enterprise)

Back End Description

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

Various Syntax in SQL

“All the syntax mentioned have been tested with a Microsoft SQL Server Management Studio”

SQL SELECT Statement

SELECT column1, column2....columnN FROM table_name;

SQL DISTINCT Clause

SELECT DISTINCT column1, column2....columnN FROM table_name;

SQL WHERE Clause

SELECT column1, column2....columnN FROM table_name WHERE
CONDITION;

SQL AND/OR Clause

SELECT column1, column2....columnN FROM table_name WHERE
CONDITION-1 {AND|OR} CONDITION-2;

SQL IN Clause

SELECT column1, column2....columnN FROM table_name WHERE
column_name IN (val-1, val-2,...val-N);

SQL BETWEEN Clause

SELECT column1, column2....columnN FROM table_name WHERE
column_name BETWEEN val-1 AND val-2;

SQL LIKE Clause

SELECT column1, column2....columnN FROM table_name WHERE
column_name LIKE { PATTERN };

SQL ORDER BY Clause

SELECT column1, column2....columnN FROM table_name WHERE
CONDITION ORDER BY column_name {ASC|DESC};

SQL GROUP BY Clause

SELECT SUM(column_name) FROM table_name WHERE CONDITION
GROUP BY column_name;

SQL COUNT Clause

SELECT COUNT(column_name) FROM table_name WHERE CONDITION;

SQL HAVING Clause

SELECT SUM(column_name) FROM table_name WHERE CONDITION
GROUP BY column_name HAVING (arithmetic function condition);

SQL CREATE TABLE Statement

CREATE TABLE table_name(column1 datatype, column2 datatype, column3
datatype, columnN datatype, PRIMARY KEY(one or more columns));

SQL DROP TABLE Statement

DROP TABLE table_name; SQL CREATE INDEX Statement CREATE
UNIQUE INDEX index_name ON table_name (column1,
column2,...columnN);

SQL DROP INDEX Statement

ALTER TABLE table_name DROP INDEX index_name;

SQL DESC Statement

DESC table_name;

SQL TRUNCATE TABLE Statement

TRUNCATE TABLE table_name; SQL ALTER TABLE Statement ALTER
TABLE table_name {ADD|DROP|MODIFY} column_name {data_type};

SQL ALTER TABLE Statement (Rename)

ALTER TABLE table_name RENAME TO new_table_name; SQL INSERT INTO Statement INSERT INTO table_name(column1, column2....columnN) VALUES (value1, value2....valueN);

SQL UPDATE Statement

UPDATE table_name SET column1 = value1, column2 = value2....columnN=valueN [WHERE CONDITION];

SQL DELETE Statement

DELETE FROM table_name WHERE {CONDITION}; SQL CREATE DATABASE Statement CREATE DATABASE database_name;

SQL DROP DATABASE Statement

DROP DATABASE database_name; SQL USE Statement USE database_name; SQL COMMIT Statement COMMIT;

SQL ROLLBACK Statement

ROLLBACK;

Data in SQL Server:

1. dbo.Invoice_Details

	INVOICE NO.	INVOICE DATE	COSTUMER NAME	PAYMENT MODE
1	10001	15-06-2020	HARI RAM	DEBIT CARD
2	10002	15-06-2020	VIRAJ SHARMA	INTERNET BANKING
3	10003	15-06-2020	RAGHU RAJ	CASH

2. dbo.Product_Details

	PRODUCT ID	PRODUCT CATEGORY	PRODUCT NAME
1	101	1 TB HDD	TOSHIBA HDD
2	102	1 TB HDD	WESTERN DIGITAL HDD
3	103	1 TB HDD	SEAGATE HDD
4	104	1 TB HDD	SAMSUNG HDD

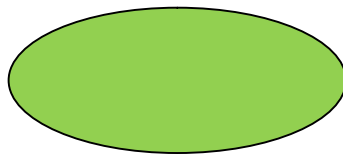
Design

Flowchart

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as “flowcharting”.

Basic Symbols used in Flowchart Designs

1. Terminal: The oval symbol indicates Start, Stop and Halt in a program’s logic flow. A pause/halt is generally used in a program logic under some error conditions. Terminal is the first and last symbols in the flowchart.



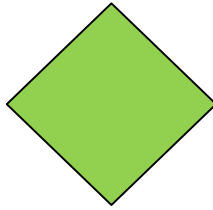
2. Input/output: A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.



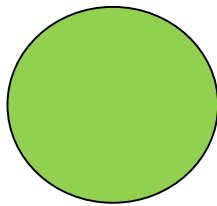
3. Processing: A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.



4. Decision: Diamond symbol represents a decision point. Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.



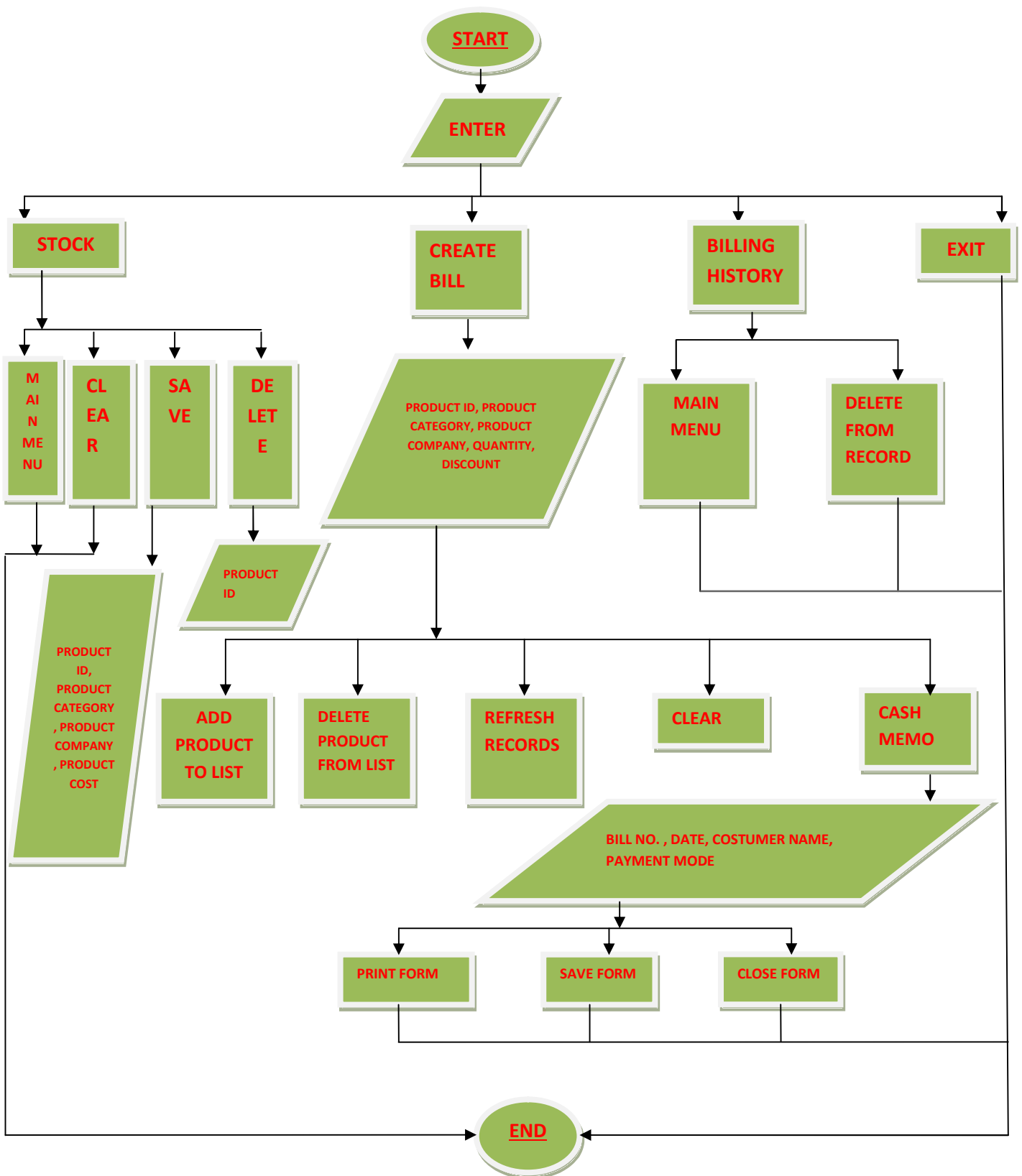
5. Connectors: Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



6. Flow lines: Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.



In the Store Management System, we have used flow chart to design the entire layout of the software.

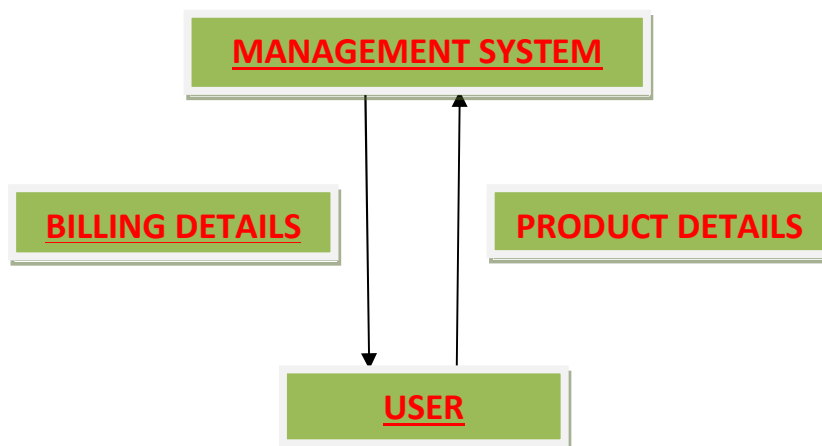


Data Flow Diagram:

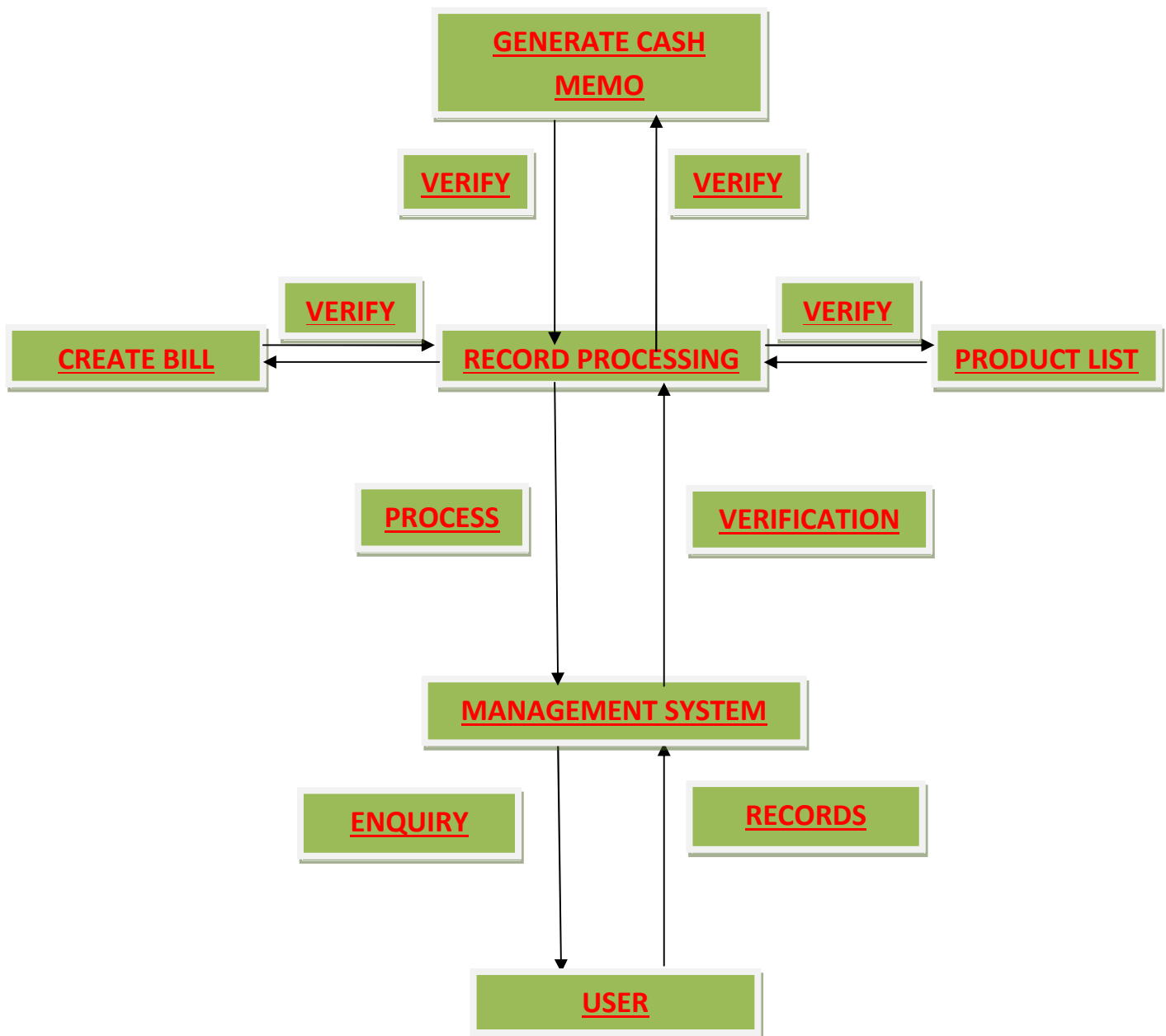
In Software engineering DFD (data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

0-level DFD: It is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

Here, 0-level DFD is shown alongside, used for this project;



1-level DFD: In 1-level DFD, context diagram is decomposed into multiple bubbles/processes. In this level we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.



Entity Relationship Diagram

ER-modelling is a data modelling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER-modelling technique are called Entity Relationship Diagrams, or ER diagrams or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases

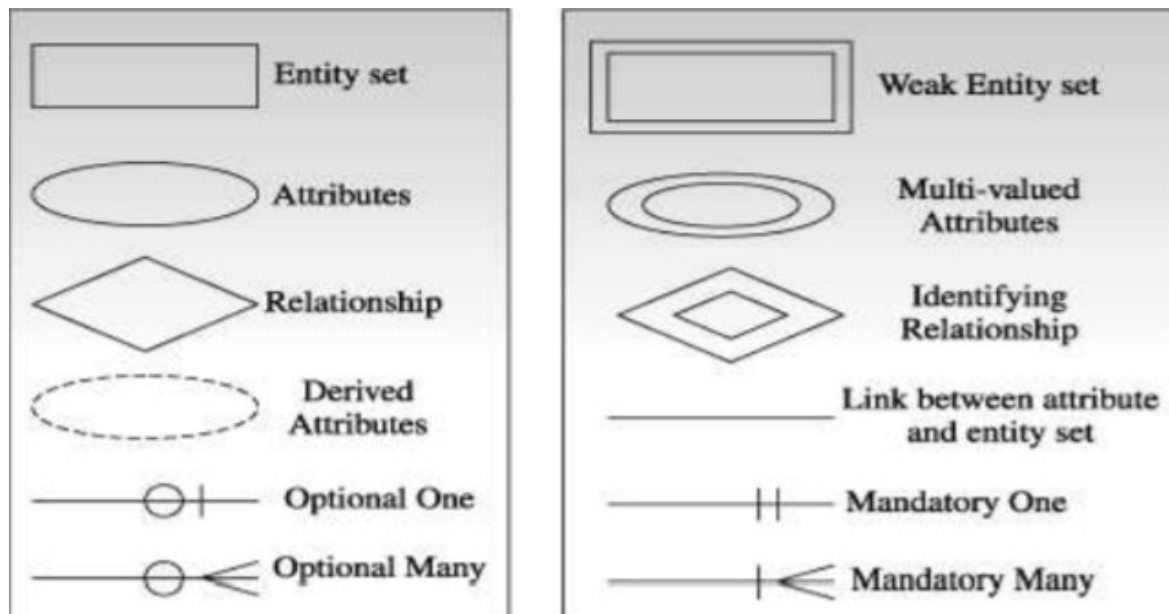
There are three basic elements in ER-Diagrams:

Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.

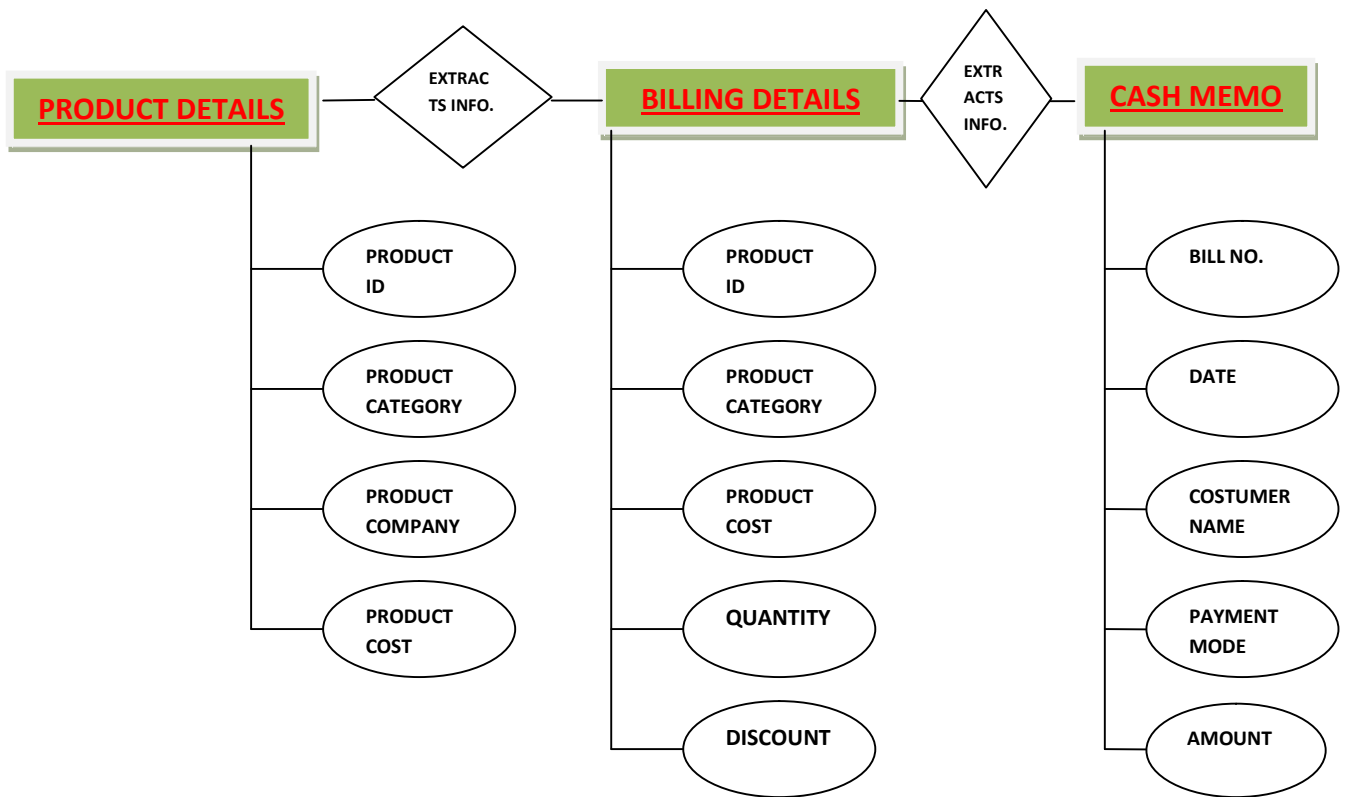
Attributes are the data we want to collect for an entity.

Relationships describe the relations between the entities.

ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.



E-R Diagram for this project is designed as:



Description of the modules

All the modules used in this project along with their source code and output screen are mentioned below:

FORM 1: (SOURCE CODE)

Public Class Form1

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

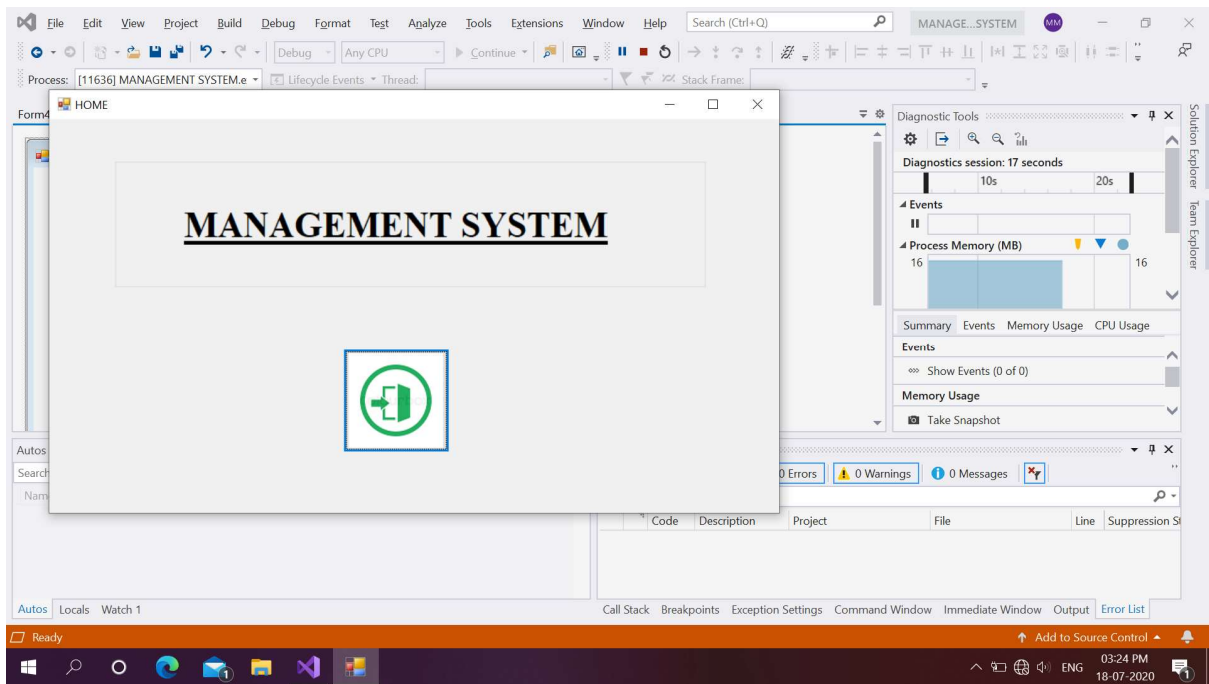
Form2.Show()

Me.Hide()

End Sub

End Class

FORM 1: (OUTPUT)



FORM 2: (SOURCE CODE)

Public Class Form2

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

Me.Hide()

Form3.Show()

End Sub

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles  
Button2.Click
```

```
    Me.Hide()
```

```
    Form4.Show()
```

```
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles  
Button3.Click
```

```
    Me.Hide()
```

```
    Form6.Show()
```

```
End Sub
```

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles  
Button4.Click
```

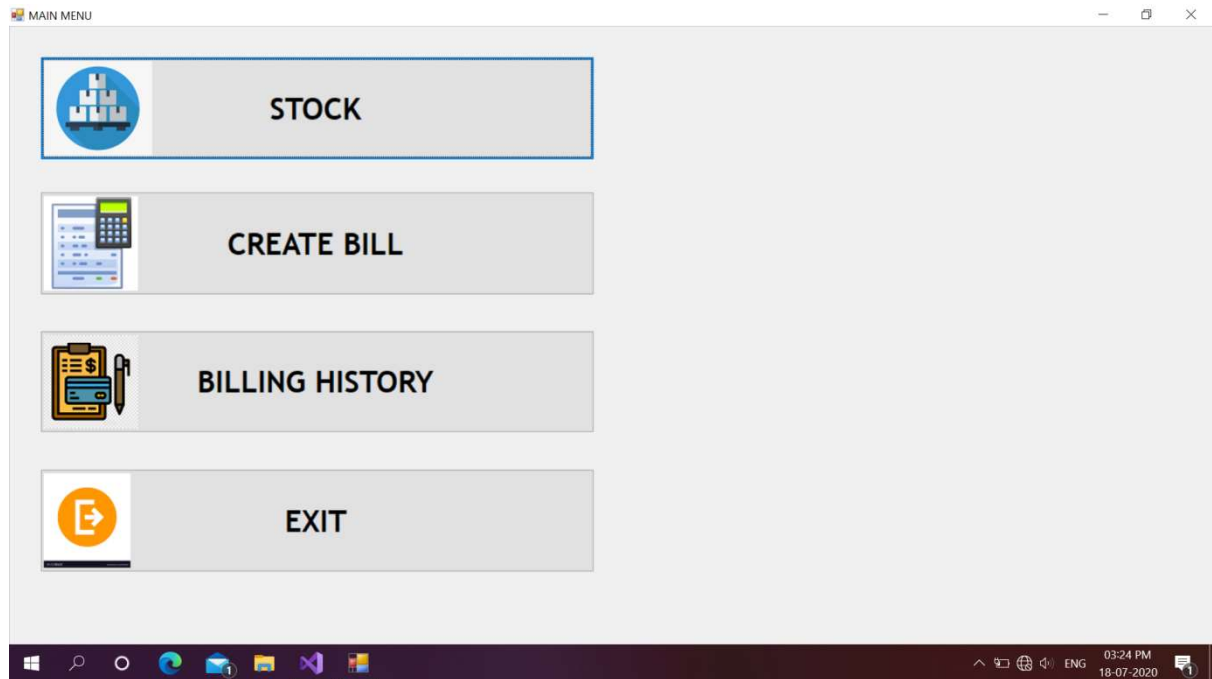
```
    Form1.Close()
```

```
    Me.Close()
```

```
End Sub
```

```
End Class
```

FORM 2: (OUTPUT)



FORM 3: (SOURCE CODE)

Imports System.Data.SqlClient

Imports System.Data

Public Class Form3

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click

 TextBox1.Clear()

 TextBox2.Clear()

 TextBox3.Clear()

 TextBox4.Clear()

 TextBox1.Focus()

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

 If TextBox1.TextLength = 0 Then

 MessageBox.Show("ENTER PRODUCT ID")

 End If

 If TextBox2.TextLength = 0 Then

 MessageBox.Show("ENTER PRODUCT CATEGORY")

 End If

 If TextBox3.TextLength = 0 Then

 MessageBox.Show("ENTER PRODUCT NAME")

 End If

 If TextBox4.TextLength = 0 Then

 MessageBox.Show("ENTER PRODUCT COST")

 End If

 Dim cn As SqlConnection

 Dim cmd As SqlCommand

 cn = New SqlConnection("Data Source=LAPTOP-6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

 cmd = New SqlCommand("insert into Product_Details([PRODUCT ID] , [PRODUCT CATEGORY] , [PRODUCT NAME] , [PRODUCT COST]) values ('" & TextBox1.Text & "' ,'" & TextBox2.Text & "' ,'" & TextBox3.Text & "' ,'" & TextBox4.Text & "')", cn)

 Try

 cn.Open()

 cmd.ExecuteNonQuery()

 cn.Close()

```
MessageBox.Show(" SAVED SUCCESSFULLY ")
```

```
Button4.PerformClick()
```

```
TextBox1.Clear()
```

```
TextBox2.Clear()
```

```
TextBox3.Clear()
```

```
TextBox4.Clear()
```

```
TextBox1.Focus()
```

```
Catch ex As Exception
```

```
    MessageBox.Show("ERROR ...")
```

```
End Try
```

```
End Sub
```

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles  
Button4.Click
```

```
    If TextBox1.TextLength = 0 Then
```

```
        MessageBox.Show("ENTER PRODUCT ID")
```

```
    End If
```

```
    Dim cn1 As SqlConnection
```

```
    Dim cmd1 As SqlCommand
```

```
    cn1 = New SqlConnection("Data Source=LAPTOP-  
6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")
```

```
    cmd1 = New SqlCommand("delete from Product_Details where  
[PRODUCT ID] = " & TextBox1.Text & "", cn1)
```

```
    Try
```

```
        cn1.Open()
```

```
cmd1.ExecuteNonQuery()
```

```
cn1.Close()
```

```
MessageBox.Show(" DELETED SUCCESSFULLY ")
```

```
Button4.PerformClick()
```

```
TextBox1.Clear()
```

```
TextBox2.Clear()
```

```
TextBox3.Clear()
```

```
TextBox4.Clear()
```

```
TextBox1.Focus()
```

```
Catch ex1 As Exception
```

```
    MessageBox.Show("ERROR ...")
```

```
End Try
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles  
Button1.Click
```

```
    Form2.Show()
```

```
    Me.Close()
```

```
End Sub
```

```
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles  
Button5.Click
```

```
    Dim cn2 As SqlConnection
```

```
    Dim cmd2 As SqlCommand
```

```
    Dim da As SqlDataAdapter
```



```

Dim dt As DataTable

cn2 = New SqlConnection("Data Source=LAPTOP-
6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

da = New SqlDataAdapter

dt = New DataTable

cmd2 = New SqlCommand("select * from Product_Details order by
[PRODUCT ID]", cn2)

Try

    cn2.Open()

    da.SelectCommand = cmd2

    da.Fill(dt)

    cn2.Close()

    DataGridView1.DataSource = dt

Catch ex2 As Exception

    MessageBox.Show("ERROR ...")

End Try

End Sub

```

```

Private Sub Form3_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

```

```

    Button5.PerformClick()

```

```

End Sub

```

```

Private Sub DataGridView1_CellMouseClick(sender As Object, e As
DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick

```

```

Try

```

```

    If e.RowIndex >= 0 Then

```

```
        Dim row As DataGridViewRow =  
DataGridView1.Rows(e.RowIndex)  
  
        TextBox1.Text = row.Cells(0).Value.ToString  
  
        TextBox2.Text = row.Cells(1).Value.ToString  
  
        TextBox3.Text = row.Cells(2).Value.ToString  
  
        TextBox4.Text = row.Cells(3).Value.ToString  
  
    End If  
  
    Catch ex3 As Exception  
  
        MessageBox.Show("ERROR ...")  
  
    End Try  
  
End Sub
```

```
Private Sub TextBox1_KeyPress(sender As Object, e As  
KeyPressEventArgs) Handles TextBox1.KeyPress  
  
    If Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Or  
Char.IsPunctuation(e.KeyChar) Then  
  
        e.Handled = True  
  
    Else  
  
        e.Handled = False  
  
    End If  
  
End Sub
```

```
Private Sub TextBox4_KeyPress(sender As Object, e As  
KeyPressEventArgs) Handles TextBox4.KeyPress  
  
    If Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Or  
Char.IsPunctuation(e.KeyChar) Then
```

e.Handled = True

Else

e.Handled = False

End If

End Sub

End Class

FORM 3: (OUTPUT)

The screenshot shows a Windows application window titled "STOCK". Inside the window, there is a section titled "PRODUCT DETAILS" with four input fields: "PRODUCT ID :", "PRODUCT CATEGORY :", "PRODUCT COMPANY :", and "PRODUCT COST :". Below these fields are four buttons: "<< MAIN MENU <<", "CLEAR", "ADD", and "DELETE". Below the buttons is a table with the following data:

	PRODUCT ID	PRODUCT CATEGORY	PRODUCT NAME	PRODUCT COST
▶	101	1 TB HDD	TOSHIBA HDD	3000
	102	1 TB HDD	WESTERN DIGI...	5000
	103	1 TB HDD	SEAGATE HDD	7000
	104	256 GB SSD	SAMSUNG	5500
*				

The Windows taskbar at the bottom shows the time as 05:08 PM on 19-07-2020, along with various system icons and open applications.

The screenshot shows a Windows application window titled "STOCK". Inside the window, there is a section titled "PRODUCT DETAILS" with four input fields: "PRODUCT ID : 104", "PRODUCT CATEGORY : 256 GB SSD", "PRODUCT COMPANY : SAMSUNG", and "PRODUCT COST : 5500". Below these fields are four buttons: "<< MAIN MENU <<", "CLEAR", "ADD", and "DELETE". At the bottom of the window is a table with the following data:

	PRODUCT ID	PRODUCT CATEGORY	PRODUCT NAME	PRODUCT COST
	101	1 TB HDD	TOSHIBA HDD	3000
	102	1 TB HDD	WESTERN DIGI...	5000
	103	1 TB HDD	SEAGATE HDD	7000
▶	104	256 GB SSD	SAMSUNG	5500
*				

The Windows taskbar at the bottom shows the time as 04:57 PM on 19-07-2020.

FORM 4: (SOURCE CODE)

Imports System.Data.SqlClient

Imports System.Data

Public Class Form4

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles
Button6.Click

Dim cn3 As SqlConnection

Dim cmd3 As SqlCommand

Dim da1 As SqlDataAdapter

Dim dt1 As DataTable

cn3 = New SqlConnection("Data Source=LAPTOP-
6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

da1 = New SqlDataAdapter

dt1 = New DataTable

```
cmd3 = New SqlCommand("select * from Product_Details order by  
[PRODUCT ID]", cn3)
```

```
Try
```

```
cn3.Open()
```

```
da1.SelectCommand = cmd3
```

```
da1.Fill(dt1)
```

```
cn3.Close()
```

```
DataGridView1.DataSource = dt1
```

```
Catch ex4 As Exception
```

```
MessageBox.Show("ERROR ...")
```

```
End Try
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles  
Button1.Click
```

```
Form2.Show()
```

```
Me.Close()
```

```
End Sub
```

```
Private Sub DataGridView1_CellMouseClick(sender As Object, e As  
DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick
```

```
Try
```

```
If e.RowIndex >= 0 Then
```

```
Dim row As DataGridViewRow =  
DataGridView1.Rows(e.RowIndex)
```

```
TextBox1.Text = row.Cells(0).Value.ToString
```

```

        TextBox2.Text = row.Cells(1).Value.ToString
        TextBox3.Text = row.Cells(2).Value.ToString
        TextBox4.Text = row.Cells(3).Value.ToString
    End If

    Catch ex5 As Exception

        MessageBox.Show("ERROR ...")

    End Try

End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click

    If TextBox5.TextLength = 0 Then

        MessageBox.Show("ENTER QUANTITY")

    Else

        Try

            TextBox6.Text = Val(TextBox5.Text) * Val(TextBox4.Text)

            ListView1.Items.Add(New ListViewItem(New String()
{TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text,
TextBox6.Text}))

            Dim TotalSum As Double = 0

            Dim TempNode As ListViewItem

            Dim TempDbl As Double

            For Each TempNode In ListView1.Items

                If Double.TryParse(TempNode.SubItems.Item(4).Text, TempDbl)

Then

                    TotalSum += TempDbl

```

End If

Next

Label10.Text = TotalSum

Catch ex6 As Exception

MessageBox.Show("ERROR ...")

End Try

End If

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click

TextBox1.Clear()

TextBox2.Clear()

TextBox3.Clear()

TextBox4.Clear()

TextBox5.Clear()

TextBox6.Clear()

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click

Form5.Show()

Me.Hide()

End Sub

```
Private Sub Form4_Load(sender As Object, e As EventArgs) Handles  
MyBase.Load
```

```
    Button6.PerformClick()
```

```
End Sub
```

```
Private Sub TextBox5_KeyPress(sender As Object, e As  
KeyPressEventArgs) Handles TextBox5.KeyPress
```

```
    If Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Or  
Char.IsPunctuation(e.KeyChar) Then
```

```
        e.Handled = True
```

```
    Else
```

```
        e.Handled = False
```

```
    End If
```

```
End Sub
```

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles  
Button4.Click
```

```
    Try
```

```
        For Each i As ListViewItem In ListView1.SelectedItems
```

```
            ListView1.Items.Remove(i)
```

```
        Next
```

```
        Dim TotalSum As Double = 0
```

```
        Dim TempNode As ListViewItem
```

```
        Dim TempDb1 As Double
```

```
        For Each TempNode In ListView1.Items
```



```
        If Double.TryParse(TempNode.SubItems.Item(4).Text, TempDb1)
Then
        TotalSum += TempDb1
    End If
Next
Label10.Text = TotalSum
TextBox1.Clear()
TextBox2.Clear()
TextBox3.Clear()
TextBox4.Clear()
TextBox5.Clear()
TextBox6.Clear()
Catch ex7 As Exception
    MessageBox.Show("ERROR ...")
End Try
End Sub
End Class
```

FORM 4: (OUTPUT)

BILL CREATOR

PRODUCT DETAILS

PRODUCT ID :

PRODUCT CATEGORY :

PRODUCT COMPANY :

PRODUCT COST :

QUANTITY :

AMOUNT :

PRODUCT ID	PRODUCT CATEGORY	PRODUCT NAME	PRODUCT COST
101	1 TB HDD	TOSHIBA HDD	3000
102	1 TB HDD	WESTERN DIGI...	5000
103	1 TB HDD	SEAGATE HDD	7000
104	256 GB SSD	SAMSUNG	5500
*			

Product Category	Product Company	Product Cost	Quantity	Total Amount
1 TB HDD	WESTERN DIGITAL HDD	5000	2	10000

CLICK TO REFRESH PRODUCT LIST

ADD PRODUCT TO LIST

REMOVE PRODUCT FROM LIST

DISCOUNT : %

TOTAL AMOUNT : 10000

<< MAIN MENU <<

FORM 5: (SOURCE CODE)

Imports System.Data.SqlClient

Imports System.Data

Public Class Form5

Private Sub PrintDocument1_PrintPage(sender As Object, e As
Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

e.Graphics.DrawString("E BILLING SYSTEM", New Font("Arial", 30,
FontStyle.Bold), Brushes.Black, New Point(250, 30))

e.Graphics.DrawString("-----
-----", New Font("Arial", 30), Brushes.Black, New Point(0,
80))

e.Graphics.DrawString("-----
-----", New Font("Arial", 30), Brushes.Black, New Point(0,
90))

e.Graphics.DrawString("NAME :" + TextBox3.Text, New Font("Arial",
20, FontStyle.Bold), Brushes.Black, New Point(20, 150))

```
e.Graphics.DrawString("BILL NO. :" + TextBox1.Text, New Font("Arial",  
20, FontStyle.Bold), Brushes.Black, New Point(20, 210))
```

```
e.Graphics.DrawString("DATE :" + TextBox2.Text, New Font("Arial", 20,  
FontStyle.Bold), Brushes.Black, New Point(20, 270))
```

```
e.Graphics.DrawString("TOTAL AMOUNT :" + Label7.Text, New  
Font("Arial", 20, FontStyle.Bold), Brushes.Black, New Point(20, 330))
```

```
e.Graphics.DrawString("-----  
-----", New Font("Arial", 30), Brushes.Black, New Point(0,  
380))
```

```
e.Graphics.DrawString("-----  
-----", New Font("Arial", 30), Brushes.Black, New Point(0,  
390))
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles  
Button1.Click
```

```
PrintPreviewDialog1.Document = PrintDocument1
```

```
PrintPreviewDialog1.ShowDialog()
```

```
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles  
Button3.Click
```

```
Me.Close()
```

```
Form4.Show()
```

```
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles  
Button2.Click
```

If TextBox1.TextLength = 0 Then

 MessageBox.Show("ENTER INVOICE NO.")

End If

If TextBox2.TextLength = 0 Then

 MessageBox.Show("ENTER INVOICE DATE")

End If

If TextBox3.TextLength = 0 Then

 MessageBox.Show("ENTER COSTUMER NAME")

End If

If ComboBox1.Text.Length = 0 Then

 MessageBox.Show("ENTER PAYMENT MODE")

End If

Dim cn3 As SqlConnection

Dim cmd3 As SqlCommand

cn3 = New SqlConnection("Data Source=LAPTOP-6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

cmd3 = New SqlCommand("insert into Invoice_Details([INVOICE NO.] , [INVOICE DATE] , [COSTUMER NAME] , [PAYMENT MODE] , [AMOUNT]) values ('" & TextBox1.Text & "' ,'" & TextBox2.Text & "' ,'" & TextBox3.Text & "' ,'" & ComboBox1.Text & "' ,'" & Label7.Text & "')", cn3)

Try

 cn3.Open()

 cmd3.ExecuteNonQuery()

 MessageBox.Show(" SAVED SUCCESSFULLY ...")

 cn3.Close()

Catch ex8 As Exception

```
        MessageBox.Show("ERROR ...")
```

```
    End Try
```

```
End Sub
```

```
Private Sub Form5_Load(sender As Object, e As EventArgs) Handles  
MyBase.Load
```

```
    Dim discount As Integer
```

```
    discount = (((Val(Form4.Label10.Text)) * (Val(Form4.TextBox7.Text))) /  
100))
```

```
    Label7.Text = Val(Form4.Label10.Text) - discount
```

```
End Sub
```

```
Private Sub TextBox1_KeyPress(sender As Object, e As  
KeyPressEventArgs) Handles TextBox1.KeyPress
```

```
    If Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Or  
Char.IsPunctuation(e.KeyChar) Then
```

```
        e.Handled = True
```

```
    Else
```

```
        e.Handled = False
```

```
    End If
```

```
End Sub
```

```
Private Sub TextBox2_KeyPress(sender As Object, e As  
KeyPressEventArgs) Handles TextBox2.KeyPress
```

```
    If Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Then
```

```
        e.Handled = True
```

Else

e.Handled = False

End If

End Sub

Private Sub TextBox3_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles TextBox3.KeyPress

If Char.IsDigit(e.KeyChar) Or Char.IsPunctuation(e.KeyChar) Then

e.Handled = True

Else

e.Handled = False

End If

End Sub

Private Sub ComboBox1_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles ComboBox1.KeyPress

If Char.IsDigit(e.KeyChar) Or Char.IsPunctuation(e.KeyChar) Or
Char.IsUpper(e.KeyChar) Or Char.IsLower(e.KeyChar) Then

e.Handled = True

Else

e.Handled = False

End If

End Sub

End Class

FORM 5: (OUTPUT)

CASH MEMO

CASH MEMO

BILLING DETAILS

BILL NO : 10003

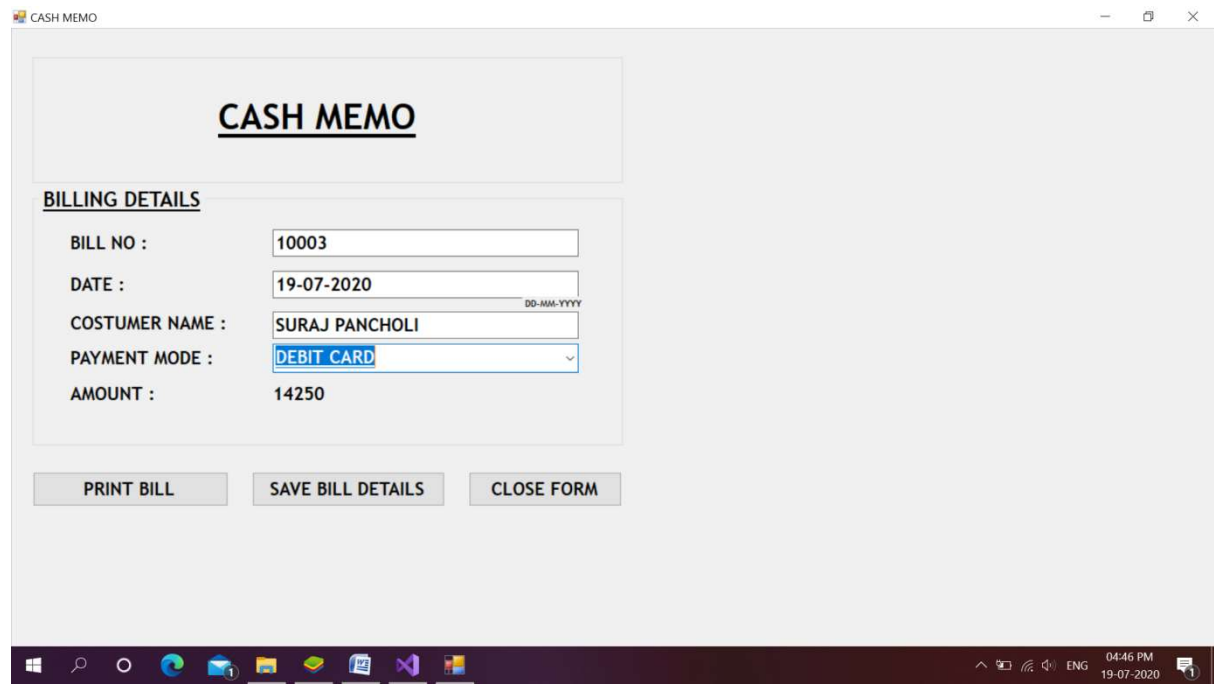
DATE : 19-07-2020 DD-MM-YYYY

COSTUMER NAME : SURAJ PANCHOLI

PAYMENT MODE : DEBIT CARD

AMOUNT : 14250

PRINT BILL SAVE BILL DETAILS CLOSE FORM



CASH MEMO

CASH MEMO

BILLING DETAILS

BILL NO : 10003

DATE : 19-07-2020 DD-MM-YYYY

COSTUMER NAME : SURAJ PANCHOLI

PAYMENT MODE : DEBIT CARD

AMOUNT : 14250

PRINT BILL SAVE BILL DETAILS CLOSE FORM

Print preview

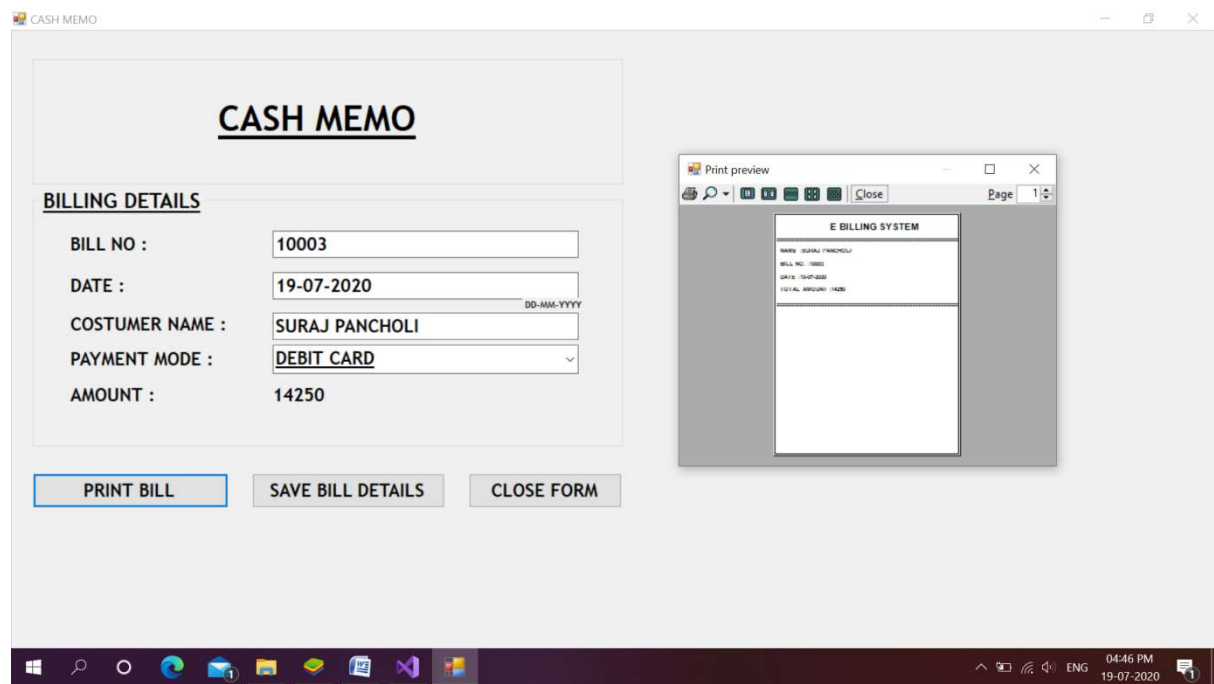
E BILLING SYSTEM

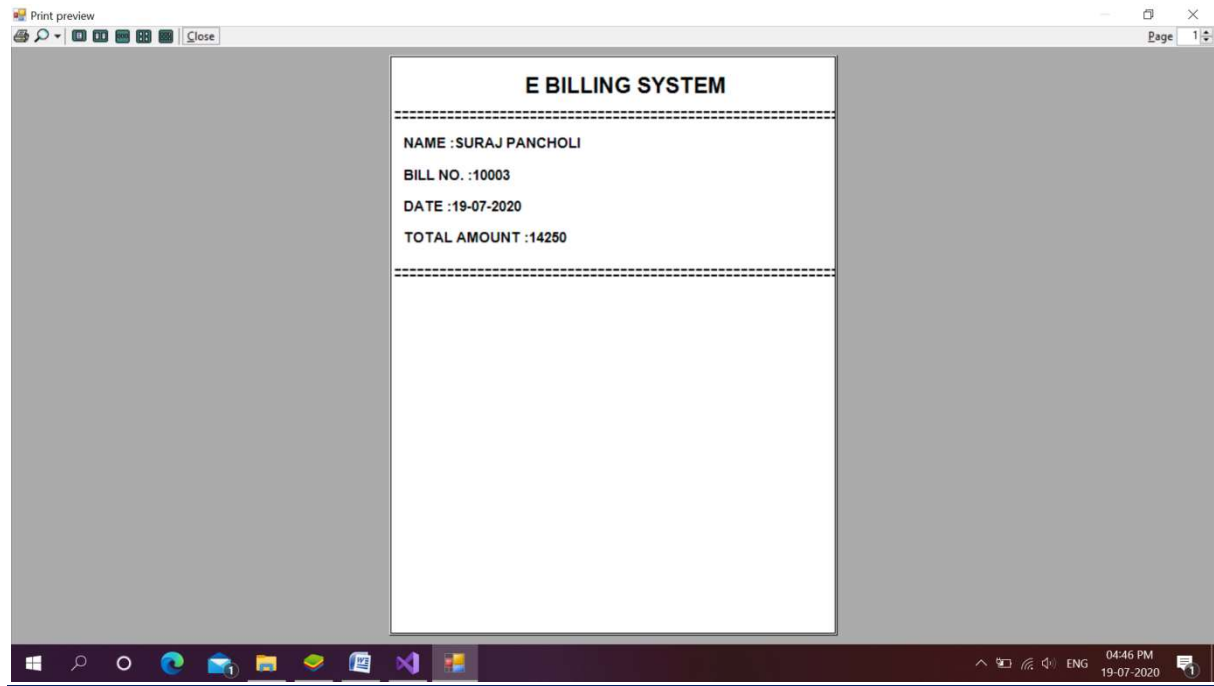
NAME: SURAJ PANCHOLI

BILL NO: 10003

DATE: 19-07-2020

TOTAL AMOUNT: 14250





FORM 6: (SOURCE CODE)

Imports System.Data.SqlClient

Imports System.Data

Public Class Form6

Dim da2 As SqlDataAdapter

Dim dt2 As DataTable

Dim cmd4 As SqlCommand

Dim cn4 As New SqlConnection

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

Me.Close()

Form2.Show()

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click

Try

cn4 = New SqlConnection("Data Source=LAPTOP-
6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

da2 = New SqlDataAdapter

dt2 = New DataTable

cmd4 = New SqlCommand("select * from Invoice_Details order by
[INVOICE NO.]", cn4)

cn4.Open()

da2.SelectCommand = cmd4

da2.Fill(dt2)

cn4.Close()

DataGridView1.DataSource = dt2

DataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect

Catch ex9 As Exception

MessageBox.Show("ERROR ...")

End Try

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click

cn4 = New SqlConnection("Data Source=LAPTOP-
6OHR0KQ7\SQLEXPRESS;Initial Catalog=MYDB;Integrated Security=True")

```
cmd4 = New SqlCommand("delete from Invoice_Details where [INVOICE  
NO.] = " & TextBox1.Text & "", cn4)
```

```
Try
```

```
cn4.Open()
```

```
cmd4.ExecuteNonQuery()
```

```
cn4.Close()
```

```
MessageBox.Show(" DELETED SUCCESSFULLY ")
```

```
Button3.PerformClick()
```

```
TextBox1.Clear()
```

```
Catch ex1 As Exception
```

```
MessageBox.Show("ERROR ...")
```

```
End Try
```

```
End Sub
```

```
Private Sub Form6_Load(sender As Object, e As EventArgs) Handles  
MyBase.Load
```

```
Button3.PerformClick()
```

```
End Sub
```

```
Private Sub DataGridView1_CellMouseClick(sender As Object, e As  
DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick
```

```
Try
```

```
If e.RowIndex >= 0 Then
```

```
Dim row As DataGridViewRow =  
DataGridView1.Rows(e.RowIndex)
```

```
TextBox1.Text = row.Cells(0).Value.ToString
```

End If

Catch ex3 As Exception

MessageBox.Show("ERROR ...")

End Try

End Sub

End Class

FORM 6: (OUTPUT)

The screenshot displays a software application window titled "BILLING HISTORY". At the top left, there is a button labeled "<< MAIN MENU <<". At the top right, there is a button labeled "DELETE". Below these buttons is a table with the following data:

	INVOICE NO.	INVOICE DATE	COSTUMER NAME	PAYMENT MODE	AMOUNT
▶	10001	18-07-2020	VIRAJ SETH	UPI	10250
	10002	18-07-2020	AYUSH GUPTA	INTERNET BAN...	14725
*					

Below the table is a large grey rectangular area. The Windows taskbar is visible at the bottom of the screen, showing the time as 05:30 PM on 18-07-2020.

Testing

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 the quality product.

Software Testing Definition according to ANSI/IEEE 1059 standard – A process of analysing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item.

The types of testing we have used over here are:

1. Black Box testing
2. White Box testing
3. Unit testing
4. Integration testing
5. Interface Testing

Black Box Testing: It is also called as Behavioural/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.

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

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.

Integration Testing: Integration Testing is the process of testing the connectivity or data transfer between a couple of unit tested modules.

Interface Testing: Interface Testing is defined as a software testing type which verifies whether the communication between two different software systems is done correctly. A connection that integrates two components is called interface.

Advantages:

1. Less paper work.
2. Fastest work in a mean time interval.
3. It is reliable for the users.
4. User friendly environment.
5. Conjugate resource allocation of the project.
6. Administrator user interface.

Future Scope

1. Provide options for backup of database
2. Provide options for export of data to excel sheets
3. Provide options for compaction of data
4. Provide options for retrieval of if database crashes down
5. Provide options for keeping data of available stock

Conclusion

It has been a matter of immense pleasure, honour and challenge to have this opportunity to take up this project and to complete it successfully. While developing this project I have learned a lot about academic records. I have also learnt how to make it user friendly by hiding the complicated parts of it.

During the development of this project, I studied and carefully understood the criteria for making software more demanding. I realized the importance of file backups and recovery facilities. I also realized the importance of maintaining a minimal margin for errors.

Bibliography

Websites

- <http://www.geekforgeeks.com>
- <http://www.tutorialspoint.com>
- <http://www.stackoverflow.com>

Books

- Introduction to VB Programming : By V. K. Jain