PROPERTY MANAGEMENT SYSTEM

Submitted By

PREETHI.S (17PCCO13)

Under the Guidance of

Dr. S. JANSI, MCA., Ph.D.,

In partial fulfillment of the requirements for the award of the degree of

Master of Commerce with Computer Applications

DEPARTMENT OF COMMERCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

APRIL-2019

CERTIFICATE

This is to certify that the project work entitled **"PROPERTY MANAGEMENT SYSTEM"** submitted to Department of Commerce, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, in partial fulfillment of the requirements for the award of the **DEGREE OF MASTER OF COMMERCE WITH COMPUTER APPLICATIONS,** is the record of the original project work done by **S.PREETHI**during the period of their study, under my supervision and guidance.

Signature of the Guide

Signature of Head of the Department

Submitted for the viva voce examination held on_____

Internal Examiner External Examiner

DECLARATION

I hereby declare that this project work entitled **"PROPERTY MANAGEMENT SYSTEM"** submitted to Department of commerce, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, in partial fulfillment of the requirements for the award of the **DEGREE OF MASTER OF COMMERCE WITH COMPUTER APPLICATIONS** is the record of original project work done by **S.PREETHI** during the period of my study, under the supervision and guidance of **Dr.S.JANSI, MCA., Ph.D.,** Assistant Professor , Department of Commerce.

Signature of the Candidate

S.PREETHI

ACKNOWLEDGEMENT

First and Foremost, I thank the God Almighty who has been a power of strength towards the successful completion of the project work

I express my gratitude to our **Chancellor Padma Shri. Dr. P.R. Krishnakumar** of Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and kindness.

I express my sincere and heartfelt gratitude to our **Vice Chancellor Dr. (Mrs) Premavathy Vijayan** M.Sc, M.Ed., Dip.spl.Edn., M.Phil., Ph.D., Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore for the resources facilitated for completion of my project.

I express my gratitude to our **Registrar Dr. (Mrs) S.Kowsalya** M.sc.,M.Phil.,Ph.D., Avinashilingam Institute for Home Science and Higher Education for Women, for providing all facilities necessary for the project.

My special thanks to **Dr.(Mrs.) U.Jerinabi** M.com.,Dip.Ed.,M.Phil.,Ph.D., Dean, School of Commerce and Management, Department of Commerce, and **Dr (Mrs.) D. Geetha** M.Com., Dip.Ed., M.Phil., Ph.D., Head and Professor Department of Commerce for her encouragement and for giving necessary help and support for completing the project successfully.

I take this opportunity to express my sincere thanks to my supervisor and guide **Dr. (Mrs.) S.JANSI** M.C.A., Ph.D., Assistant Professor, Department of Commerce for her valuable guidance and for her timely support, suggestions and motivation throughout the project.

Let yet importantly, I would like to thank all the partners in "**BALAJI BUILDERS**" Coimbatore for the support throughout my endeavor and also I thank my parents, friends and well-wisher who have contributed directly or indirectly to the successful completion of the project.

ABSTRACT

The main aim of the project is to develop an application for **"property management system"** for computerization of number of purchase and sales property sites is to keep track on their customer details, sales, property details, installment pay, message alert for payment and billing process.

This system is designed for the real estate as a consultant for distribution of sites. This system maintains the entire process of purchase and sales plots. The administrator enters into the project through the login form. This form contains the login id and password. This software has the interface of adding, updating, deleting the property details, customer details etc. The proposed system will help the admin to make storing of the sales records, installment details and customer information a lot easier. This project is developed with VB.Net as frontend and MY SQL SERVER 2008 as backend.

CHAPTER I

INTRODUCTION

1.1 AN OVERVIEW

The project work entitled as **"Property Management System"** is designed using VB. Net as front end and Microsoft SQL Server 2008 as back end.

The main aim of the project is to develop an application for PROPERTY MANAGEMENT SYSTEM for computerization of number of purchase and sales property sites is to keep track on their customer details, sales, property details, installment pay, message alert for payment and billing process. It involves the processes, systems and manpower required to manage the life cycle of all acquired property including acquisition, control, accountability, maintenance, utilization, and disposition. The conventional method of fixed asset management has a lot of associated problems such as mismanagement, improper maintenance, lack of communication, inability to track the status of a property and tenant dissatisfaction. Computerized property management is an accounting process that seeks to register properties for the purposes of financial accounting and preventive maintenance.

This system is designed for the real estate as a consultant for distribution of sites. This system maintains the entire process of purchase and sales plots. The administrator enters into the project through the login form. This form contains the login id and password. If the id and password are correct, next main form opens and if the password and id are wrong, it will say error message. This software has the interface of adding, updating, deleting the property details, customer details etc. The proposed system will help the admin to make storing of the sales records, installment details and customer information a lot easier.

COMPANY PROFILE

COMPANY NAME	:Balaji builders
NATURE	:properitor
ADDRESS	:132-A, TVR Towers, Thadagam road, GCT Post, Coimbatore-641013, :Tamil Nadu, India.
DIRECTOR NAME	**:**G.prasad
YEAR OF ESTABLISHMENT	:2002
YEAR OF REGISTRATION	:2002
PHONE NO	:97885352277
EMAIL-ID	:tirupatibalaji04@gmail.com
ANNUAL TURNOVER	:50 Lakhs

Balaji Builders is a reputed enterprise focused on developing luxurious gated communities with the best of amenities. Established in the year 2002, the company has developed numerous luxury apartments, villas at prime locations in Coimbatore. Over the years, Balaji Buildershas earned a reputation for quality and innovation. All its projects come with ample greenery, spacious and modern design and the best of amenities and of course, luxury. The company takes particular care in ensuring that its projects feature perfect vaasthu and a clear title so as to give buyers a hassle-free and happy ownership. Today, owning a Balaji Builders Property is associated with prestige as virtually every project the company has developed till date has become a landmark in its respective surroundings.

Some of the popular apartments and villas from Balaji Builders include - Balaji Builders's Hill Top Residency, Balaji Builders 's Swapnalok **,** Balaji Builders 's Hill Paradise **,** Balaji Builders 's Park View and Balaji Builders 's Odyssey to Name a few. Apart from meeting the evolving lifestyles of urban families, all these projects have also demonstrated excellent growth in value, thus giving buyers the twin advantages of a proud address and a good investment. Its no surprise then, that all of Balaji Builders’s projects till date have received overwhelming appreciation, a fact that any of its long list of happy buyers would vouch for.

CHAPTER II

2. SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

PROCESSOR	:	INTEL DUAL CORE
RAM	:	64 GB
MONITOR	:	15” COLOR
HARD DISK	:	4 GB
CDDRIVE	:	LG 52X
KEYBOARD	:	STANDARD 102 KEYS

2.2 SOFTWARE SPECIFICATION

OPERATING SYSTEM	:	WINDOWS XP
FRONT END	:	VISUAL BASIC .NET
BACK END	:	MY SQLSERVER 2008

ABOUT WINDOWS XP

Windows XP is a computer operating system and graphical user interface (GUI), which enables you to work with a wide variety of programs on your computer, often simultaneously. Windows XP is itself a special computer program that communicates your instructions to the actual computer hardware, and displays the results.

KEY FEATURES OF WINDOWS XP

WINDOWS WITHIN WINDOWS

A Window refers to a rectangular area of the screen, within which you may view program folders and files, or display file contents such as documents, spreadsheets, and graphic images. A window can occupy part of the screen, can be maximized to fill the entire screen, or can be minimized so that it is no longer visible but remains active and is easily reaccessed.

THE DESKTOP

The Desktop gives you access to everything you need in Windows XP. It occupies the entire screen, and unlike a window, it can't be reduced in size. The desktop consists of a coloured or patterned background, containing small pictures called Icons that represent programs or data stores. Double-clicking on an icon opens the corresponding program or file inside a window.

THE TASKBAR

The Taskbar lies across the bottom edge of your screen. The Start button on the left provides access to all the programs, data files, and other features available on your computer. When you open a program or file, a corresponding rectangular icon will be displayed on your taskbar - even if the program has been minimized and is no longer visible on your screen. To access that program, you just need to click its icon on the taskbar.

THE START MENU

When you click on the Start button, a set of menu options is displayed. The contents will vary depending on your computer setup and most frequently accessed programs. If you click on the All Programs option, you'll see a list of all the programs installed on your computer – even those that don't have icons on the desktop. Press the [ESC] (escape) key to close the menu.

HELP

One of the menu options displayed when you click the Start button, is labelled Help and Support. If you select this option, a Help window will open. To get help on a specific topic, type a word or phrase in the blank space at the top left of the window and then click the search arrow; alternatively, you can click to browse any of the Help topics listed in the window. Click the X in the top right corner to close the Help window.

ABOUT VISUAL BASIC .NET

Visual Basic .NET (VB.NET) is an object-oriented computer programming language implemented on the .NET Framework. Although it is an evolution of classic Visual Basic language, it is not backwards-compatible with VB6, and any code written in the old version does not compile under VB.NET.

Like all other .NET languages, VB.NET has complete support for object-oriented concepts. Everything in VB.NET is an object, including all of the primitive types (Short, Integer, Long, String, Boolean, etc.) and user-defined types, events, and even assemblies. All objects inherits from the base class Object.

VB.NET is implemented by Microsoft's .NET framework. Therefore, it has full access to all the libraries in the .Net Framework. It's also possible to run VB.NET programs on Mono, the open-source alternative to .NET, not only under Windows, but even Linux or Mac OSX.

FEATURES OF VB.NET

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management

- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading

THE .NET FRAMEWORK

The .Net framework is a revolutionary platform that helps you to write the following types of applications such as,

- Windows applications
- Web applications
- Web services

The .Net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages such as,

- Visual Basic
- C#
- C++
- Jscript and
- COBOL etc.

All these languages can access the framework as well as communicate with each other.

The .Net framework consists of an enormous library of codes used by the client languages like VB.Net. These languages use object-oriented methodology.

Following are some of the components of the .Net framework

- Common Language Runtime (CLR)
- The .Net Framework Class Library
- Common Language Specification

- Common Type System
- Metadata and Assemblies
- Windows Forms
- ASP.Net and ASP.Net AJAX
- ADO.Net
- Windows Workflow Foundation (WF)
- Windows Presentation Foundation
- Windows Communication Foundation (WCF)
- LINQ

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR VB.NET

Microsoft provides the following development tools for VB.Net programming −

- Visual Studio 2010 (VS)
- Visual Basic 2010 Express (VBE)
- Visual Web Developer

The Visual Basic 2010 Express (VBE) and Visual Web Developer tools are used free. Using these tools, you can write all kinds of VB.Net programs from simple command-line applications to more complex applications. Visual Basic Express and Visual Web Developer Express edition are trimmed down versions of Visual Studio and has the same look and feel. They retain most features of Visual Studio.

A VB.Net program basically consists of the following parts −

- Namespace declaration
- A class or module
- One or more procedures
- Variables

- The Main procedure
- Statements & Expressions
- Comments

COMPILE AND EXECUTE VB.NET PROGRAM

If you are using Visual Studio.Net IDE, take the following steps −

- Start Visual Studio.
- On the menu bar, choose File → New → Project.
- Choose Visual Basic from templates
- Choose Console Application.
- Specify a name and location for your project using the Browse button, and then choose the OK button.
- The new project appears in Solution Explorer.
- Write code in the Code Editor.
- Click the Run button or the F5 key to run the project.

SYNTAX

- **OBJECT** − Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors - wagging, barking, eating, etc. An object is an instance of a class.
- **CLASS** − A class can be defined as a template/blueprint that describes the behaviors/states that objects of its type support.
- **METHODS** − A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
- **INSTANCE VARIABLES** − Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

MY SQL SERVER 2008

The purpose of this document is to help you migrate your applications when you are migrating the underlying database from Microsoft SQL server to some other database. Most of the issues encountered when migrating applications to use Microsoft SQLserver database are related to database incompatibility. This paper presents these incompatibilities and provides solutions for many issues. The choices made about how to migrate your application affect how you migrate the underlying database from Microsoft SQL server to some other database.

SQL SERVER SECURITY

- Login Authentication.
- Windows NT Authentication.
- SQL Server Authentication
- Permissions validation on user database.
- T-SQL statements sent to SQL server.
- SQL server checks user permissions on receipt of T-SQL statements

FEATURES OF SQL SERVER

- Created by Microsoft and Sybase in the 80s.
- Is SQL Compliant - Uses ANSI SQL
- Supports SQL – 92 standards - Uses T-SQL
- Stores data in a central location and delivers it to clients on request
- New Server Architecture
- Graphic Administration Tools
- Maintains ANSI standards and 6.x Compatibility
- Data integrity means reliability and accuracy of data.
- Integrity rules keep data consistent.
- Supports Client/Server model.
- Request response dialog.
- workload is split between the client and the server.

- Operating System compatibility.
- Runs on Win 95/98 NT, Netware, UNIX, OS/2, Appletalk, Banyan VINES.
- SQL Server must have Service Pack 4 (SP4) to run on Windows NT 4.0.
- Multiple protocol compatibility.
- SQL Server supports these protocols - Appletalk, TCP/IP.
- SMP Compatibility and Scalability
- Supports multiple processors. SMP leads to scalability.

CHAPTER III

3. SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies.

3.1EXISTING SYSTEM

In Existing system, the property record, document, customer management and controlling the installment payment are recorded by manually. Context and Importance of the system it is critical that any real estate company to control the expenses of the management and tracking the payment of the customer.

DRAWBACKS OF THE EXISTING SYSTEM

- A lot of manual work and maintain track of documents and paper
- Difficulty to maintain old records
- Time delay

3.2 PROPOSED SYSTEM

The drawbacks, which are faced during existing system, can be eradicated by using the PROPERTY MANAGEMENT SYSTEM. It has Customer details, property details and installment details. The main objective of the proposed system is to provide a user-friendly interface. The system, which is proposed, now computerizes all the processes involved in sales record management. Once the details are fed into the computer there is no need for various persons to deal with separate sections. Only a single person is enough to maintain all the reports. The security can also be given as per the requirement of the users.

FEATURES AND BENEFITS OF PROPOSED SYSTEM

- This software will reduce manual work and maintain updates in database from time to time.
- It is easy to handle works related to many apartments at a time without any confusion.
- Data is secured and easy to retrieve old records in a short time
- Efficient recording of details at various stages
- Data redundancy avoided

3.3 FEASABILITY STUDY

An important outcome of the preliminary investigation is the determination that the system requested is feasible. Feasibility study is carried out to select the best system that meets the performance requirements.Feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. It involves preliminary investigation of the project and examines whether the designed system will be useful to the organization. Months or years of effort, thousand for millions of money and untold professional embarrassment can be averted if an in-conceived system is recognized early in the definition phase.

The different types of feasibility are:

- Technical Feasibility
- Behavioral Feasibility
- Economical Feasibility
- Operational Feasibility

TECHNICAL FEASIBILITY

Technical Feasibility deals with the hardware as well as software requirements. Technology is not a constraint to type system development. We have to find out whether the necessary technology, the proposed equipments have the capacity to hold the data, which is used in the project, should be checked to carryout this technical feasibility.

The technical feasibility issues usually raised during the feasibility stage of investigation includes these

- This software is running in windows 2000 Operating System, which can be easily installed.
- The hardware required is Pentium based server.
- The system can be expanded.

BEHAVIORAL FEASIBILITY

This feasibility test asks if the system will work when it is developed and installed.

Operational feasibility in this project:

- The proposed system offers greater level of user-friendliness.
- The proposed system produces best results and gives high performance. It can be implemented easily.

ECONOMICAL FEASIBILITY

Economical Feasibility deals about the economical impact faced by the organization to implement a new system. Financial benefits must equal or exceed the costs. The cost of conducting a full system, including software and hardware cost for the class of application being considered should be evaluated. Economic Feasibility in this project:

- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

OPERATIONAL FEASIBILITY

Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project.

CHAPTER IV

4. SYSTEM DESIGN AND DEVELOPMENT

4 .1 INTRODUCTION

System Design is the most creative and challenging phase in the system life cycle. Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. System design is a solution how to approach the creation of a new system. System design transforms a logic representation of what is required to do into the physical specification. The specification is converted into physical reality during development.

4.2 LOGICAL DESIGN

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes, frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

4.3 PHYSICAL DESIGN

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design Walk through.
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits , costs , conversion date and system constraints

DESIGN/SPECIFICATION ACTIVITIES

- Concept formulation.
- Problem understanding.
- High level requirements proposals.
- Feasibility study.
- Requirements engineering.
- Architectural design.

4.4 INPUT DESIGN

Input Design deals with what data should be given as input, how the data should be arranged or code, the dialog to guide the operating personnel in providing input, methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input

process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

In this project, the input design consists of a log in screen, tab for compression/ decompression, source and destination browsing button, a menu list for selecting the algorithm, Compress/Decompress option, compress/decompress button.

4.5 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. The objective of output design is to convey information about past activities, current status or projections of the future, signal important events, opportunities, problems, or warnings, trigger an action, confirm an action etc. Efficient, intelligible output design should improve the system's relationship with the user and helps in decisions making. In output design the emphasis is on displaying the output on a CRT screen in a predefined format. The primary consideration in design of output is the information requirement and objectives of the end users. The major formation of the output is to convey the information and so its layout and design need a careful consideration.

There is an output display screen for showing the compressed/ decompressed file or folder details (Original file size, Compressed/Decompressed file size, distinct characters)

4.6 DATA FLOW DIAGRAM

The first step is to draw a Data Flow Diagram (DFD). The DFD was first developed by Larry Constantine as a way of expressing system requirements in graphical form.

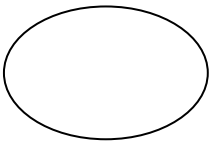
A DFD also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of series of bubbles join by the data flows in the system.

The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers. The diagrams are:

- Graphical, eliminating thousands of words;
- Logical representations, modeling WHAT a system does, rather than physical models showing.
- Hierarchical, showing systems at any level of detail; and
- Jargon less, allowing user understanding and reviewing.

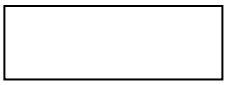
The goal of data flow diagramming is to have a commonly understood model of a system. The diagrams are the basis of structured systems analysis. Data flow diagrams are supported by other techniques of structured systems analysis such as data structure diagrams, data dictionaries, and procedure-representing techniques such as decision tables, decision trees, and structured language.

The symbols appearing in the DFD has been explained below:



- Represents a process

- Which shows data flow

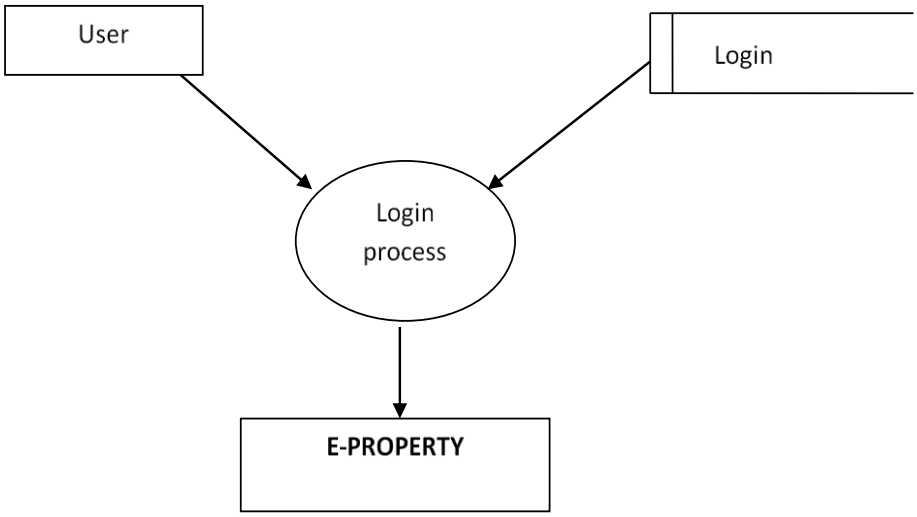


- Designation of the data

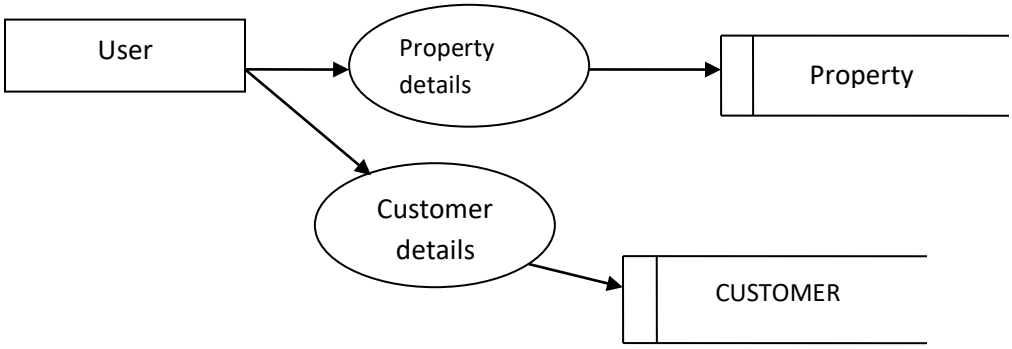
- Shows Data source

DATA FLOW DIAGRAM

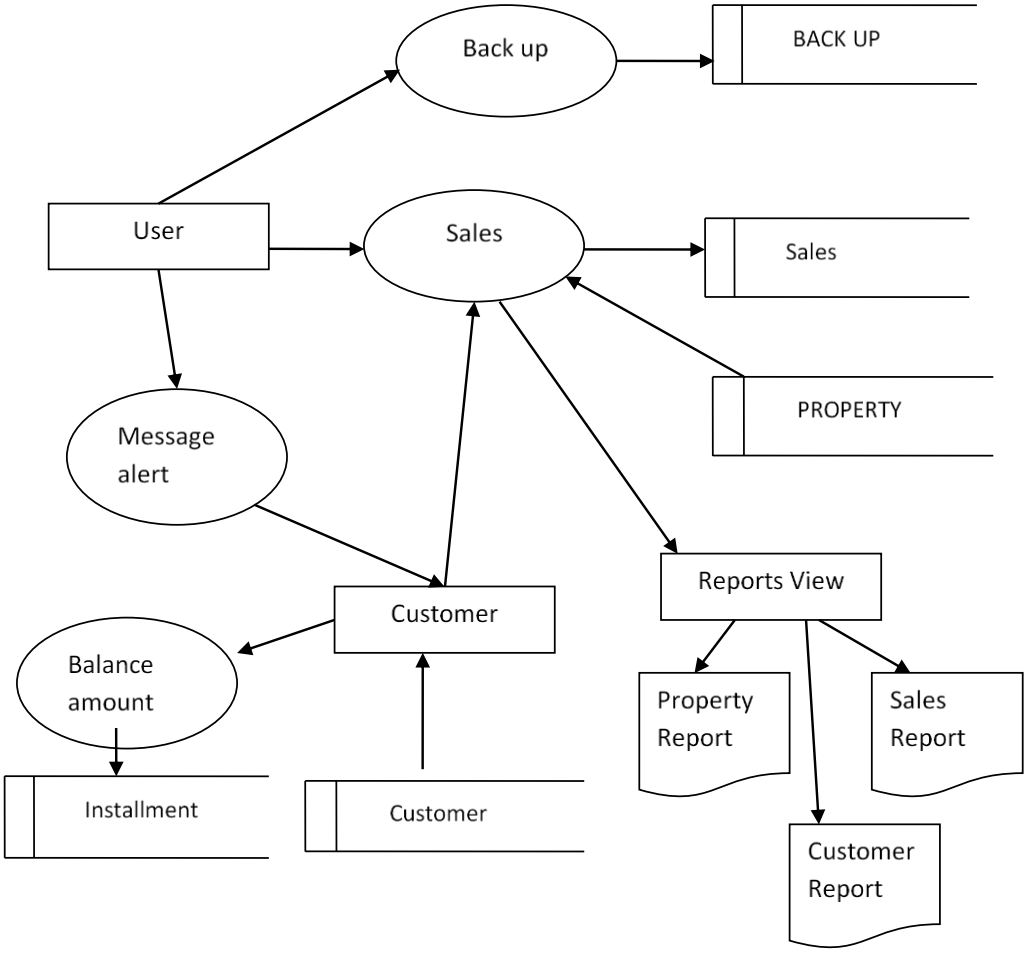
Level-0



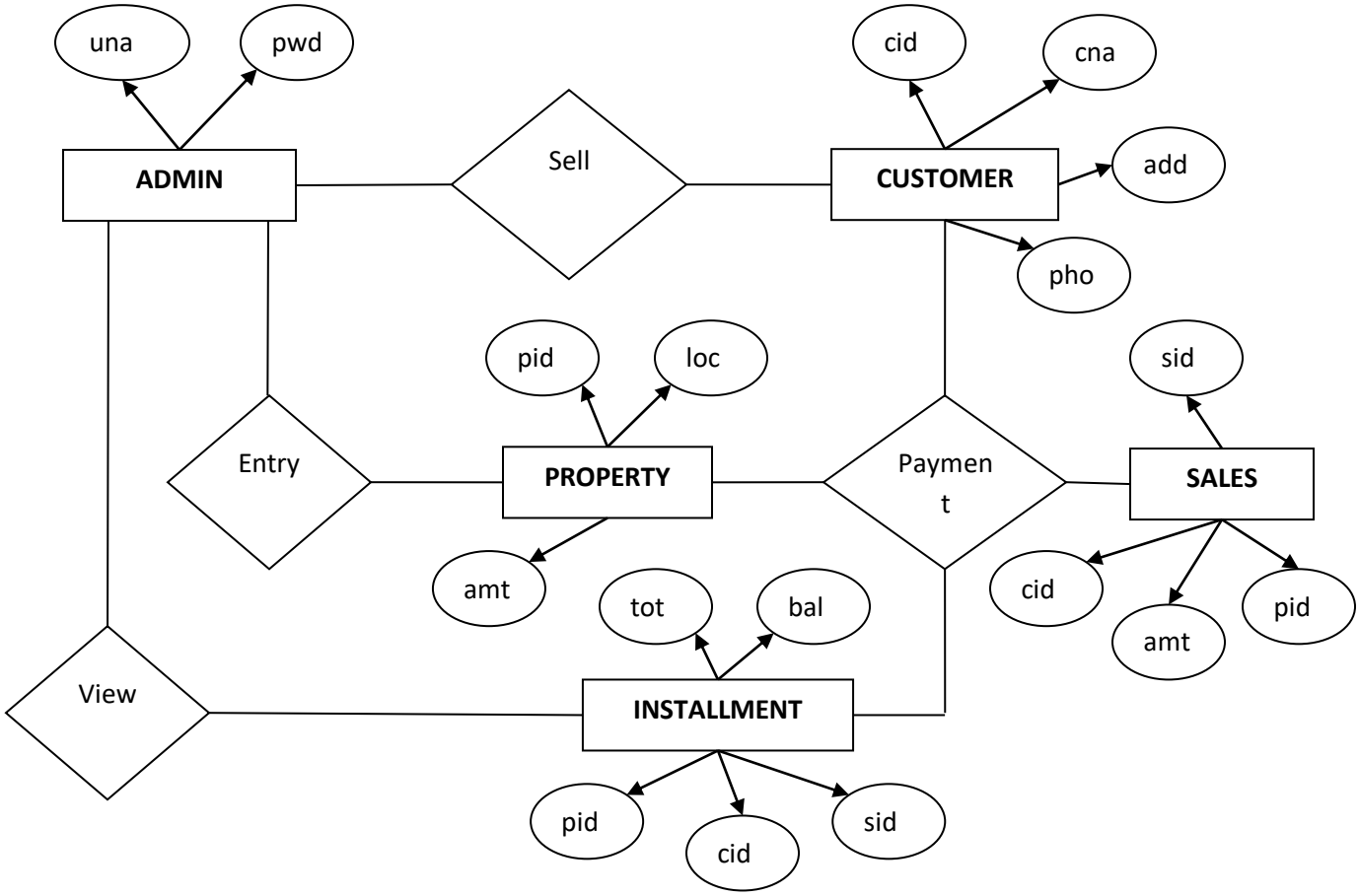
Level 1


User
Property details
Property
Customer details
CUSTOMER

Level 2


Back up
BACK UP
User
Sales
Sales
PROPERTY
Message alert
Reports View
Customer
Balance amount
Property Report
Sales Report
Installment
Customer
Customer Report

ER DIAGRAM


una
pwd
ADMIN
Sell
cid
cna
CUSTOMER
add
pho
pid
loc
Entry
PROPERTY
Paymen
t
sid
SALES
amt
tot
bal
cid
amt
pid
View
INSTALLMENT
pid
cid
sid

4.7 TABLE DESIGN

Table Name: login id

Field Name	**DataType**	**Description**
Uname	Varchar(10)	username
Pass	Varchar(10)	password

Table Name: Property details

Primary key: Property id

Field Name	**DataType**	**Description**
Pid	Varchar(7)	Property id
Pname	Varchar(20)	Property name
Property type	Varchar(20)	Property type
Loc	Varchar(15)	Location
Area(sq)	Varchar(15)	Area(sq)
City	Varchar(15)	City
State	Varchar(15)	State
Tot_sites	Int	Tot_sites
Booked	int	Booked
Registered	int	Registered

Table Name: Customer details

Primary key: Customer id

Field name	**Data Type**	**Description**
Cid	Varchar(10)	Customer id
Cname	Varchar(20)	Customer name
Fname	Varchar(20)	Fname
Dob	datetime	Date of birth
Address	Varchar(30)	Address
City	Varchar(15)	City
State	Varchar(15)	State
Pincode	int	Pincode
Pno	int	Phone no
Alternate_phone no	int	Alternate_pno
Emailed	Varchar(30)	Email-ID
Nominee	Varchar(20)	Nominee
Nominee_relationship	Varchar(10)	Nominee_relationship
Site_place	Varchar(20)	Site_place
Site_num	Varchar(10)	Site_num
Date	datetime	Date

Table Name: Sales details

Foreign key: Sales id

Field name	**Data Type**	**Description**
Sales id	Varchar(10)	Sales id
cid	Varchar(10)	Customer id
Pid	Varchar(10)	Property id
Sno	Varchar(10)	Site Number
Total amount	int	Total amount
Paidby	Varchar(20)	Paid by installment or cash

Table Name: Payment details

Primary key: Insid

Field Name	**DataType**	**Description**
Insid	int	Installment id
Sid	Varchar()	Sales id
cid	Varchar(50)	Customer id
Pid	Varchar(50)	Property id
t_amount	float	Total amount
Date	datetime	Date
Paid	float	Paid
Bal	float	Balance

Table Name: Site details

Field Name	**Data Type**	**Description**
Pid	Varchar(10)	Property id
S_no	Varchar(5)	Site no
Area(sq)	Varchar(10)	Area(sq)
Facing	Varchar(10)	Facing
Amt	int	Amount

4.8 CODE DESIGN

Code is an ordered collection of symbols designed to provide unique identification of an attribute. Codes can be used for various purposes. They can specify object's physical or performance characteristics and they can be used to give operational instructions. They also can show inter relationships and may sometimes used to achieve secrecy or confidentiality. Codes are designed for optimum human-oriented use and machine efficiency.

Codes possess uniqueness, expandability, conciseness, uniform nets, simplicity, versatility, sort ability, meaningfulness and operability.Sufficient effort and time is spent in the preliminary study of the problem to design an efficient code. Activate serve scripting is object oriented. The source code is designed so that it can do transaction efficiently.

It is the code that dose all the updating, modifications, etc. for all object used in the project there exist an associated source code, which explains the work of that object. It also describes the flow of the project.Source code is enhanced by structured coding techniques by good internal comments and features provided by the language. The code design in this project is made modular. The modular behavior enables easy debugging and testing. Inserting comment statement wherever enhances the coding. This is done during the documentation process coding is done in such a way that errors can be trapped easily. Also modifications can easily be appended due to the codes modular behavior

SAMPLE CODING

ADMIN HOME

```
Public Class main
Private Sub main_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Form1.MdiParent = Me
Form1.Show()End Sub
Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
End
```

```
End Sub
Private Sub SiteToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SiteToolStripMenuItem.Click
sitedetails.MdiParent = Me
sitedetails.Show()
End Sub
Private Sub CustomerDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CustomerDetailsToolStripMenuItem.Click
cust.MdiParent = Me
cust.Show()
End Sub

Private Sub SalesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalesToolStripMenuItem.Click
sales.MdiParent = Me
sales.Show()
End Sub

Private Sub InstallmentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles InstallmentToolStripMenuItem.Click
instalment.MdiParent = Me
instalment.Show()
End Sub

Private Sub ChangePasswordToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChangePasswordToolStripMenuItem.Click
password.Show()
End Sub

Private Sub PropertyDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PropertyDetailsToolStripMenuItem.Click
```

```
pro.MdiParent = Me
pro.Show()
End Sub

Private Sub PaymentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PaymentToolStripMenuItem.Click
pay.MdiParent = Me
pay.Show()
End Sub

Private Sub BackupDBToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BackupDBToolStripMenuItem.Click
backup.MdiParent = Me
backup.Show()
End Sub

Private Sub CalculatorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CalculatorToolStripMenuItem.Click
Dim p As New Process()
p.StartInfo = New ProcessStartInfo("calc.exe")
p.Start()
End Sub

Private Sub NotepadToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NotepadToolStripMenuItem.Click
Dim p As New Process()
p.StartInfo = New ProcessStartInfo("notepad.exe")
p.Start()
End Sub
Private Sub CustomerReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CustomerReportToolStripMenuItem.Click
```

```
rep_customer.MdiParent = Me
rep_customer.Show()
End Sub

Private Sub PropertyReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PropertyReportToolStripMenuItem.Click
rep_property.MdiParent = Me
rep_property.Show()
End Sub

Private Sub SalesReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalesReportToolStripMenuItem.Click
rep_sales.MdiParent = Me
rep_sales.Show()
End Sub

Private Sub PaymentReportToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PaymentReportToolStripMenuItem.Click
rep_netpay.MdiParent = Me
rep_netpay.Show()
End Sub

Private Sub MessageAlertToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
End Sub
End Class
```

LOGIN

```
Imports System.Data.SqlClient
Public Class Form1
```

```
Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogin.Click
cn.Open()
Dim cmd As New SqlCommand("select * from tbl_user where user_name='" & txtLogin.Text & "' and password='" & txtPwd.Text & "' ", cn)
Dim dr As SqlDataReader = Nothing
dr = cmd.ExecuteReader()
If dr.Read() Then
main.MenuStrip1.Enabled = True
Me.Hide()
Else
MessageBox.Show("Enter Correctly..!")
End If
dr.Close()
cn.Close()
End Sub
End Class
```

CUSTOMER DETAILS

```
Imports System.Data.SqlClient
Public Class cust
Public Sub fun1()
Try
cn.Open()
Dim da As New SqlDataAdapter("select sid from tbl_site", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
```

```
ComboBox1.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub

Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNew.Click
clr()
End Sub
Public Sub clr()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
TextBox12.Text = ""
End Sub

Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
Me.Close()
End Sub
```

```
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_customer values('" & TextBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox10.Text & "','" & TextBox11.Text & "','" & ComboBox1.Text & "','" & TextBox12.Text & "','" & DateTimePicker1.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Successfully Saved", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
Private Sub fillgrid()
Dim da As New SqlDataAdapter("select * from tbl_customer", cn)
If cn.State = ConnectionState.Closed Then cn.Open()
Dim ds As New DataSet
da.Fill(ds)
DataGridView1.DataSource = ds.Tables(0)
cn.Close()
End Sub
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click
Try
Dim i As Integer = InputBox("Enter customer Id to Delete", "Delete")
If cn.State = ConnectionState.Closed Then cn.Open()
```

```
cmd = New SqlCommand("delete from tbl_customer where cid=" & i & "", cn)
cmd.ExecuteNonQuery()
MsgBox("Deleted Successfully", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
clr()
End Sub
Dim iId As Integer
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellClick
iId = Convert.ToInt32(DataGridView1.Rows(e.RowIndex).Cells(0).Value)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("select * from tbl_customer where cid=" & iId & "", cn)
Dim dr As SqlDataReader
dr = cmd.ExecuteReader
If dr.Read Then
TextBox1.Text = dr(0)
TextBox2.Text = dr(1)
TextBox3.Text = dr(2)
TextBox4.Text = dr(3)
TextBox5.Text = dr(4)
TextBox6.Text = dr(5)
TextBox7.Text = dr(6)
TextBox8.Text = dr(7)
TextBox9.Text = dr(8)
TextBox10.Text = dr(9)
TextBox11.Text = dr(10)
```

```
ComboBox1.Text = dr(11)
TextBox12.Text = dr(12)
DateTimePicker1.Text = dr(13)
End If
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
End Sub
Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("update tbl_customer set name='" & TextBox2.Text & "',addr='" & TextBox3.Text & "',city='" & TextBox4.Text & "',pin='" & TextBox5.Text & "',state='" & TextBox6.Text & "',email='" & TextBox7.Text & "',phone='" & TextBox8.Text & "',mobile='" & TextBox9.Text & "',nominee='" & TextBox10.Text & "',relation='" &TextBox11.Text & "',siteno='" & ComboBox1.Text & "',place='" & TextBox12.Text & "',date='" & DateTimePicker1.Text & "' where cid='" & iId & "'", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub

Private Sub cust_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
fillgrid()
fun1()
End Sub
End Class
```

PAYMENT DETAILS

```
Imports System.Data.SqlClient
Public Class pay
Public Sub fun1()
Try
cn.Open()
Dim da As New SqlDataAdapter("select sal_id from tbl_sales", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox1.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNew.Click
clr()
End Sub
Public Sub clr()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
```

```
TextBox4.Text = ""
End Sub

Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
Me.Close()
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
TextBox7.Text = Val(TextBox5.Text) - Val(TextBox6.Text)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_installment values('" & TextBox1.Text & "','" & ComboBox1.Text & "','" & TextBox2.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','" & DateTimePicker1.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Successfully Saved", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
clr()
End Sub
Private Sub fillgrid()
Dim da As New SqlDataAdapter("select * from tbl_netpay", cn)
If cn.State = ConnectionState.Closed Then cn.Open()
Dim ds As New DataSet
da.Fill(ds)
DataGridView1.DataSource = ds.Tables(0)
```

```
cn.Close()
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click
Try
Dim i As Integer = InputBox("Enter  Id to Delete", "Delete")
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("delete from tbl_netpay where bno=" & i & "", cn)
cmd.ExecuteNonQuery()
MsgBox("Deleted Successfully", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
clr()
End Sub
Dim iId As Integer
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellClick
iId = Convert.ToInt32(DataGridView1.Rows(e.RowIndex).Cells(0).Value)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("select * from tbl_netpay where bno=" & iId & "", cn)
Dim dr As SqlDataReader
dr = cmd.ExecuteReader
If dr.Read Then
TextBox1.Text = dr(0)
ComboBox1.Text = dr(1)
TextBox2.Text = dr(2)
```

```
TextBox5.Text = dr(3)
TextBox6.Text = dr(4)
DateTimePicker1.Text = dr(5)
End If
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
End Sub

Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("update tbl_netpay set sal_id='" & ComboBox1.Text & "',cid='" & TextBox2.Text & "',tot='" & TextBox5.Text & "',payment='" & TextBox6.Text & "' where bno='" & TextBox1.Text & "'", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
Private Sub pay_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
fillgrid()
fun1()
End Sub
```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
cn.Open()
cmd = New SqlCommand("SELECT cid,pid,sno,tot FROM tbl_sales where sal_id='" & ComboBox1.Text & "' ", cn)
Dim dr As SqlDataReader = cmd.ExecuteReader()
If (dr.Read()) Then
TextBox2.Text = dr(0).ToString()
TextBox3.Text = dr(1).ToString()
TextBox4.Text = dr(2).ToString()
TextBox5.Text = dr(3).ToString()
End If
cn.Close()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_netpay values('" & TextBox1.Text & "','" & ComboBox1.Text & "','" & TextBox2.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "','" & DateTimePicker1.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Successfully Saved", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
```

```
clr()
End Sub
End Class

Imports System.Data.SqlClient
Public Class sales
Private Sub sales_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
fillgrid()
fun1()
fun2()
fun3()
End Sub
Public Sub fun1()
Try
cn.Open()
Dim da As New SqlDataAdapter("select cid from tbl_customer", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox1.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Public Sub fun2()
Try
cn.Open()
Dim da As New SqlDataAdapter("select pid from tbl_property", cn)
```

```
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox2.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Public Sub fun3()
Try
cn.Open()
Dim da As New SqlDataAdapter("select sid from tbl_site", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox3.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNew.Click
clr()
End Sub
Public Sub clr()
```

```
TextBox1.Text = ""
TextBox2.Text = ""
End Sub

Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
Me.Close()
End SubPrivate Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_sales values('" & TextBox1.Text & "','" & ComboBox1.Text & "','" & ComboBox2.Text & "','" & ComboBox3.Text & "','" & TextBox2.Text & "','" & ComboBox4.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Successfully Saved", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
Private Sub fillgrid()
Dim da As New SqlDataAdapter("select * from tbl_sales", cn)
If cn.State = ConnectionState.Closed Then cn.Open()
Dim ds As New DataSet
da.Fill(ds)
DataGridView1.DataSource = ds.Tables(0)
cn.Close()
End Sub
```

```
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click
Try
Dim i As Integer = InputBox("Enter customer Id to Delete", "Delete")
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("delete from tbl_sales where sal_id=" & i & "", cn)
cmd.ExecuteNonQuery()
MsgBox("Deleted Successfully", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
clr()
End Sub
Dim iId As Integer
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellClick
iId = Convert.ToInt32(DataGridView1.Rows(e.RowIndex).Cells(0).Value)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("select * from tbl_sales where sal_id=" & iId & "", cn)
Dim dr As SqlDataReader
dr = cmd.ExecuteReader
If dr.Read Then
TextBox1.Text = dr(0)
ComboBox1.Text = dr(1)
ComboBox2.Text = dr(2)
ComboBox3.Text = dr(3)
TextBox2.Text = dr(4)
```

```
ComboBox4.Text = dr(5)
End If
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
End Sub

Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("update tbl_sales set cid='" & ComboBox1.Text & "',pid='" & ComboBox2.Text & "',sno='" & ComboBox3.Text & "',tot='" & TextBox2.Text & "',pay='" & ComboBox4.Text & "' where sal_id='" & iId & "'", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
End Class
```

INSTALLEMENT

```
Imports System.Data.SqlClient
Imports System.Net
Imports System.IO
Public Class instalment
```

```
Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
Me.Close()
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
cn.Open()
cmd = New SqlCommand("SELECT cid,tot,balance FROM tbl_installment where bno='" & ComboBox1.Text & "' ", cn)
Dim dr As SqlDataReader = cmd.ExecuteReader()
If (dr.Read()) Then
TextBox1.Text = dr(0).ToString()
TextBox2.Text = dr(1).ToString()
TextBox3.Text = dr(2).ToString()
End If
cn.Close()
End Sub
Public Sub fun1()
Try
cn.Open()
Dim da As New SqlDataAdapter("select distinct(bno) from tbl_installment", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox1.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
```

```
End Try
End Sub
Public Sub fun2()
Try
cn.Open()
Dim da As New SqlDataAdapter("select cid from tbl_customer", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
ComboBox2.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Private Sub instalment_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
fun1()
fillgrid()
fun2()
End Sub
Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNew.Click
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
End Sub
Private Sub fillgrid()
```

```
Dim da As New SqlDataAdapter("select * from tbl_installment_bill", cn)
If cn.State = ConnectionState.Closed Then cn.Open()
Dim ds As New DataSet
da.Fill(ds)
DataGridView1.DataSource = ds.Tables(0)
cn.Close()
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
TextBox5.Text = Val(TextBox3.Text) - Val(TextBox4.Text)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_installment_bill values('" & ComboBox1.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & DateTimePicker1.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("update tbl_installment set balance='" & TextBox5.Text & "' where bno='" & ComboBox1.Text & "'", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
```

```
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox2.SelectedIndexChanged
cn.Open()
cmd = New SqlCommand("SELECT name,mobile FROM tbl_customer where cid='" & ComboBox1.Text & "' ", cn)
Dim dr As SqlDataReader = cmd.ExecuteReader()
If (dr.Read()) Then
TextBox6.Text = dr(0).ToString()
TextBox7.Text = dr(1).ToString()
End If
cn.Close()
End Sub
Public Function apicall(ByVal url As String) As String
Dim httpreq As HttpWebRequest = DirectCast(WebRequest.Create(url), HttpWebRequest)
Try
Dim httpres As HttpWebResponse = DirectCast(httpreq.GetResponse(), HttpWebResponse)
Dim sr As New StreamReader(httpres.GetResponseStream())
Dim results As String = sr.ReadToEnd()
sr.Close()
Return results
Catch
Return "0"
End Try
End Function
Dim result As String
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
result = apicall("http://smsc.vinuxnetwork.com/httpapi/send?username=ascentztech@gmail.com&password=miracle&sender_id=PROMOTIONAL&route=P&phonenumber=" & TextBox7.Text & "&message=Payment Date And Balance Amount:" & DateTimePicker2.Text & " and " & TextBox3.Text & "")
MsgBox("Msg send")
End Sub
Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label2.Click
End Sub
End Class
```

SITE

```
Imports System.Data.SqlClient
Public Class sitedetails
Private Sub sitedetails_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
fillgrid()
fun1()
End Sub
Public Sub fun1()
Try
cn.Open()
Dim da As New SqlDataAdapter("select pid from tbl_property", cn)
Dim dt As New DataTable
Dim dr As DataRow
da.Fill(dt)
cn.Close()
For Each dr In dt.Rows
```

```
ComboBox1.Items.Add(dr.Item(0))
Next
Catch ex As Exception
MsgBox(ex.ToString())
End Try
End Sub
Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNew.Click
clr()
End Sub
Public Sub clr()
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
End Sub
Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
Me.Close()
End Sub
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("insert into tbl_site values('" & TextBox1.Text & "','" & ComboBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "')", cn)
cmd.ExecuteNonQuery()
MsgBox("Successfully Saved", MsgBoxStyle.Exclamation)
cn.Close()
Catch ex As Exception
```

```
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
Private Sub fillgrid()
Dim da As New SqlDataAdapter("select * from tbl_site", cn)
If cn.State = ConnectionState.Closed Then cn.Open()
Dim ds As New DataSet
da.Fill(ds)
DataGridView1.DataSource = ds.Tables(0)
cn.Close()
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnDelete.Click
Try
Dim i As Integer = InputBox("Enter customer Id to Delete", "Delete")
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("delete from tbl_site where sid=" & i & "", cn)
cmd.ExecuteNonQuery()
MsgBox("Deleted Successfully", MsgBoxStyle.Exclamation)
cn.Close()
fillgrid()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
clr()
End Sub
Dim iId As Integer
```

```
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellClick
iId = Convert.ToInt32(DataGridView1.Rows(e.RowIndex).Cells(0).Value)
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("select * from tbl_site where sid=" & iId & "", cn)
Dim dr As SqlDataReader
dr = cmd.ExecuteReader
If dr.Read Then
TextBox1.Text = dr(0)
ComboBox1.Text = dr(1)
TextBox2.Text = dr(2)
TextBox3.Text = dr(3)
TextBox4.Text = dr(4)
End If
cn.Close()
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
End Sub

Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEdit.Click
Try
If cn.State = ConnectionState.Closed Then cn.Open()
cmd = New SqlCommand("update tbl_site set pid='" & ComboBox1.Text & "',area='" & TextBox2.Text & "',facing='" & TextBox3.Text & "',amt='" & TextBox4.Text & "' where sid='" & TextBox1.Text & "'", cn)
cmd.ExecuteNonQuery()
MsgBox("Data are Successfully Updated", MsgBoxStyle.Exclamation)
cn.Close()
```

```
Catch ex As Exception
MsgBox(ex.Message, MsgBoxStyle.Critical)
End Try
fillgrid()
clr()
End Sub
End Class
```

PROPERTY

```
Imports System.Data.SqlClient

Public Class password

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

If TextBox2.Text = TextBox3.Text Then

cn.Open()

Dim cmd1 As New SqlCommand()

cmd1.CommandText = "update tbl_user set password='" & TextBox3.Text & "'"

cmd1.Connection = cn

cmd1.ExecuteNonQuery()

MessageBox.Show("Password Changed Successfully")

cn.Close()

Else

MessageBox.Show("Mismatch Password...Please Check..!")
```

End If

End Sub

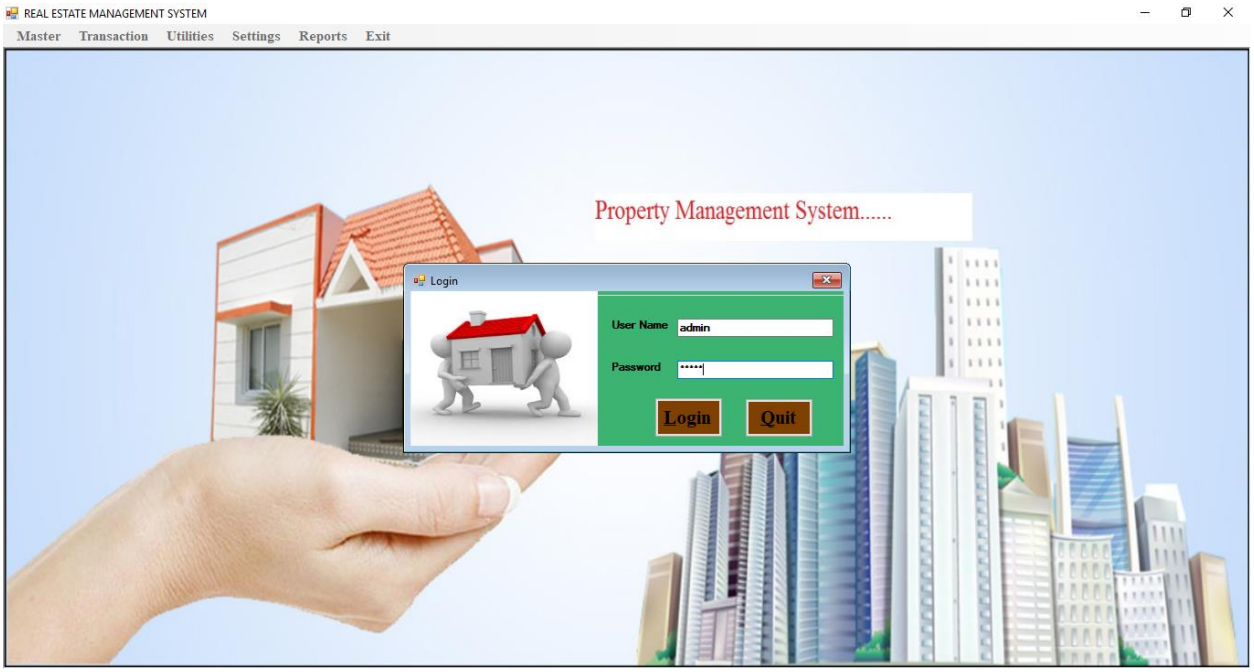
End Class

PASSWORD

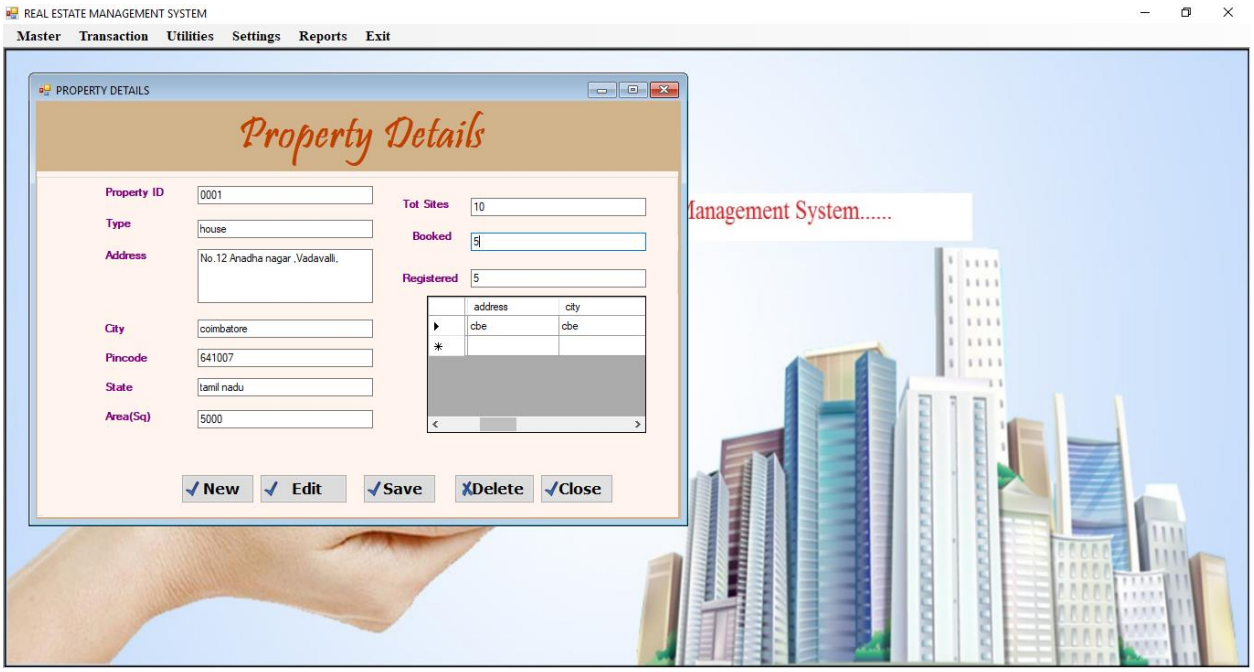
```
Imports System.Data.SqlClient
Public Class password
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
If TextBox2.Text = TextBox3.Text Then
cn.Open()
Dim cmd1 As New SqlCommand()
cmd1.CommandText = "update tbl_user set password='" & TextBox3.Text & "'"
cmd1.Connection = cn
cmd1.ExecuteNonQuery()
MessageBox.Show("Password Changed Successfully")
cn.Close()
Else
MessageBox.Show("Mismatch Password...Please Check..!")
End If
End Sub
End Class
```

4.9 FORM DESIGN

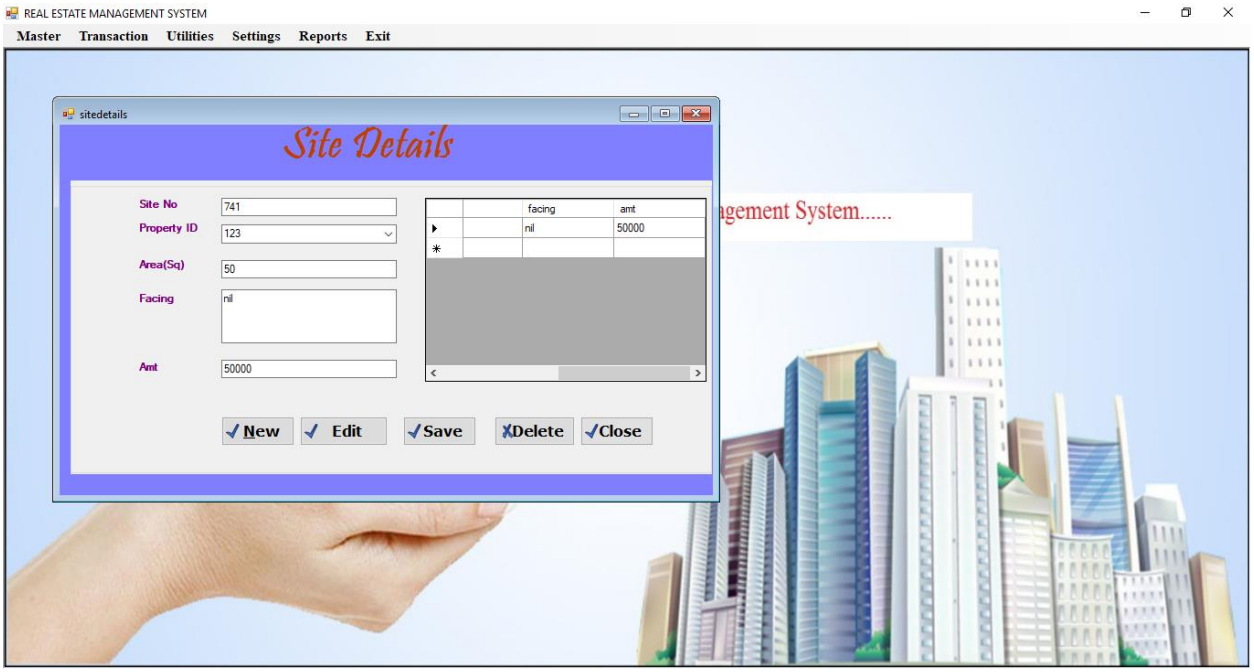
1. LOGIN PAGE



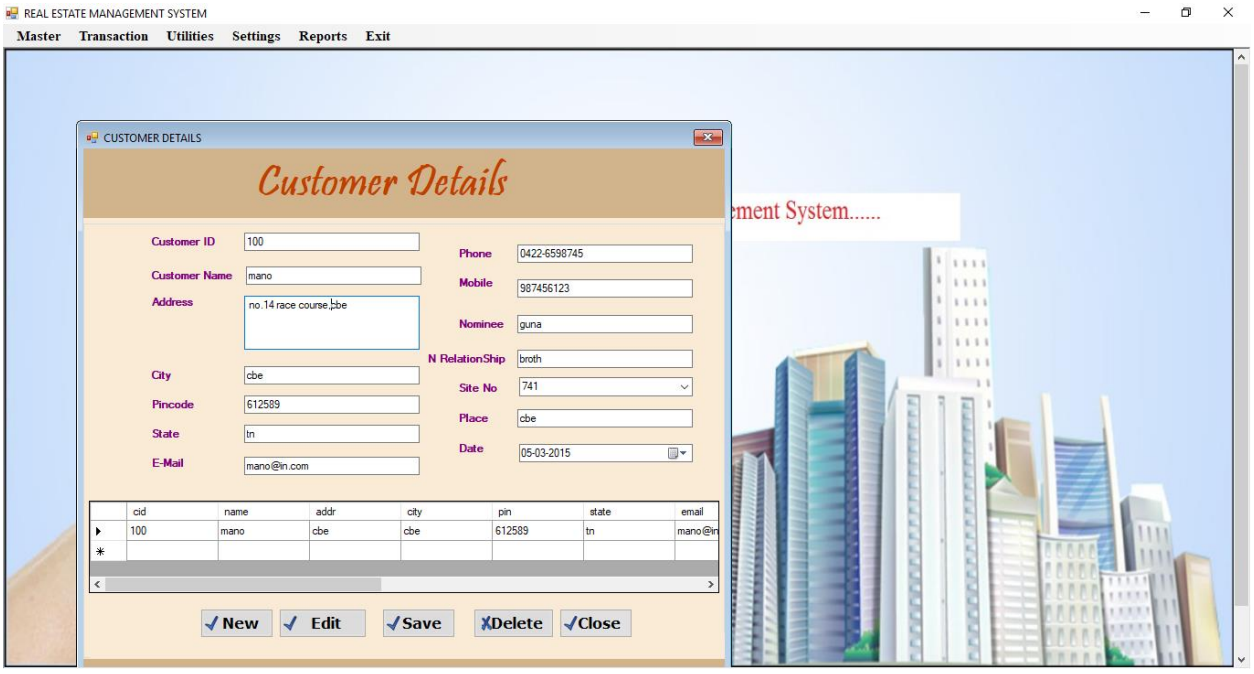
2. PROPERTY DETAILS



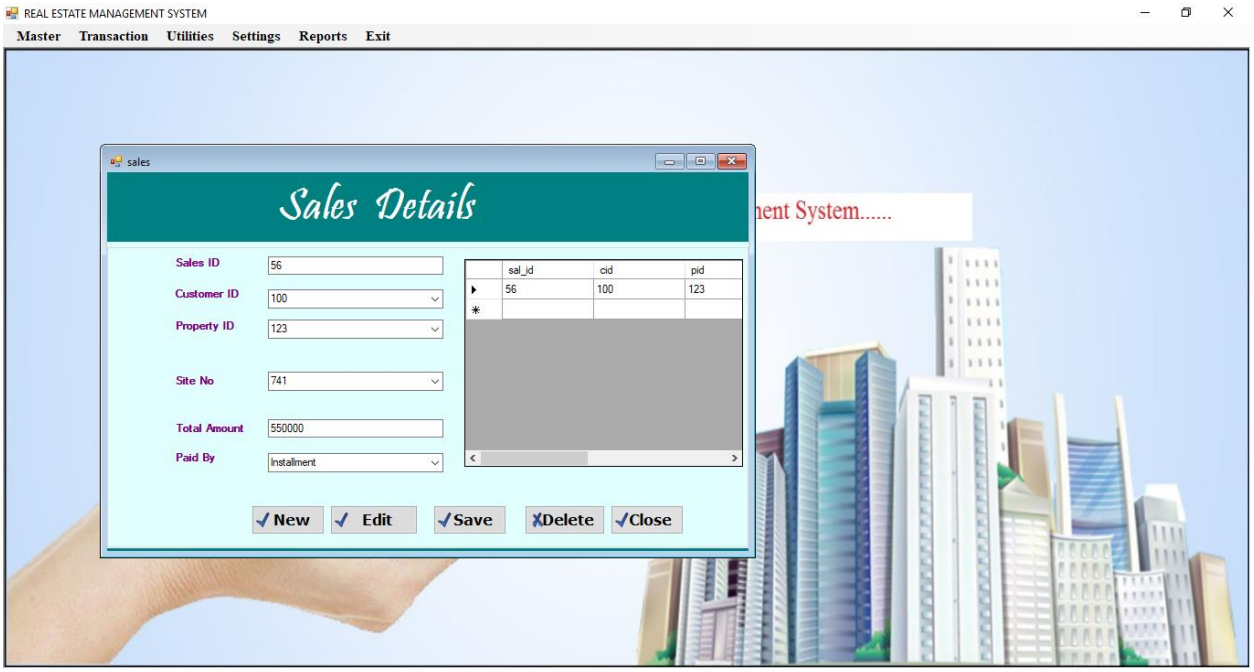
3. SITE DETAILS



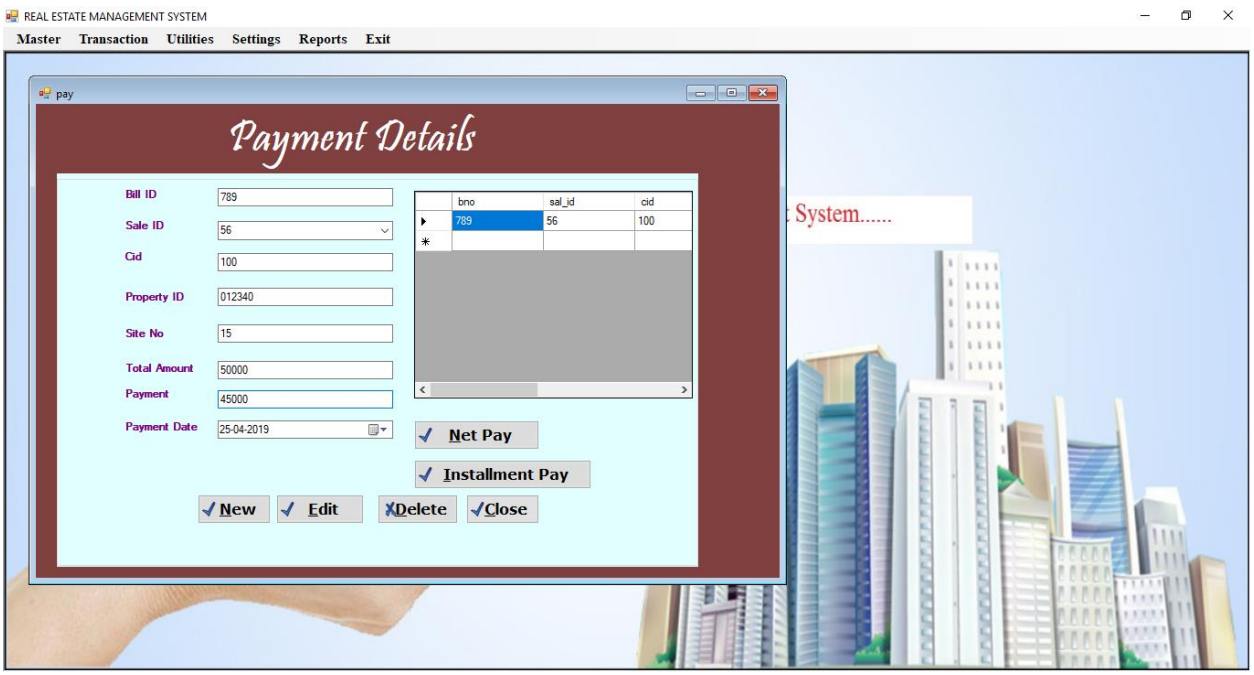
4. CUSTOMER DETAILS



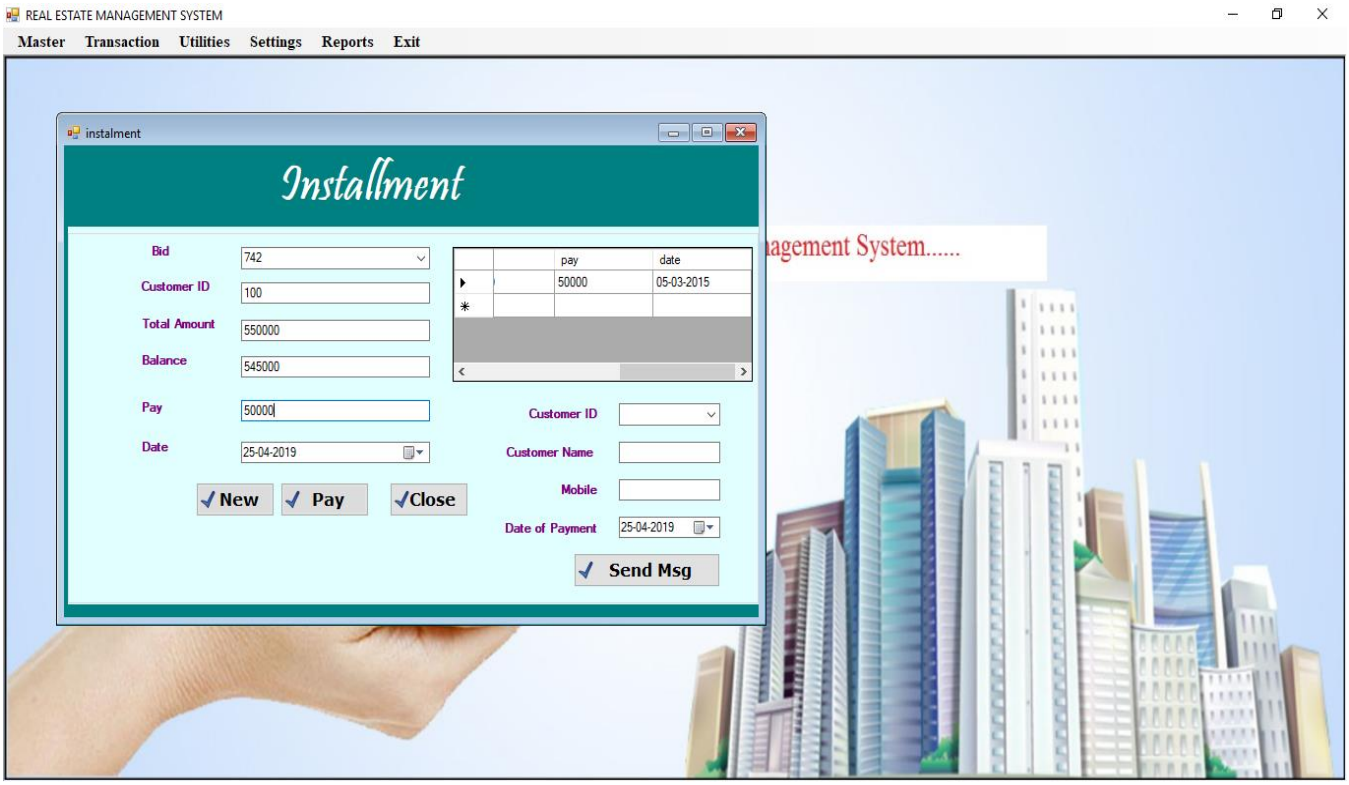
5. SALES DETAILS



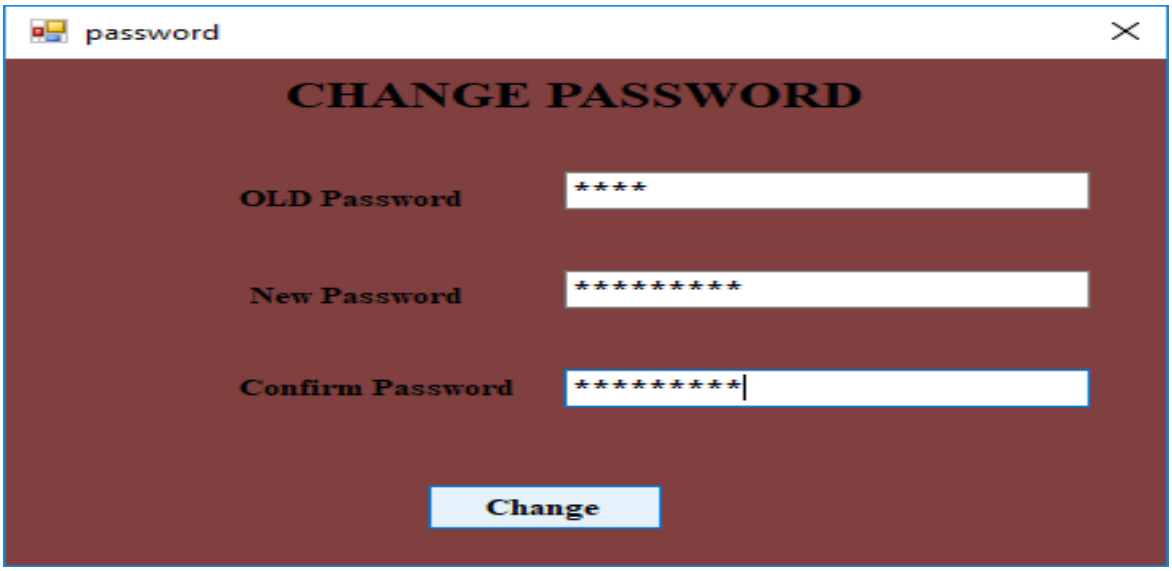
6. PAYMENT DETAILS



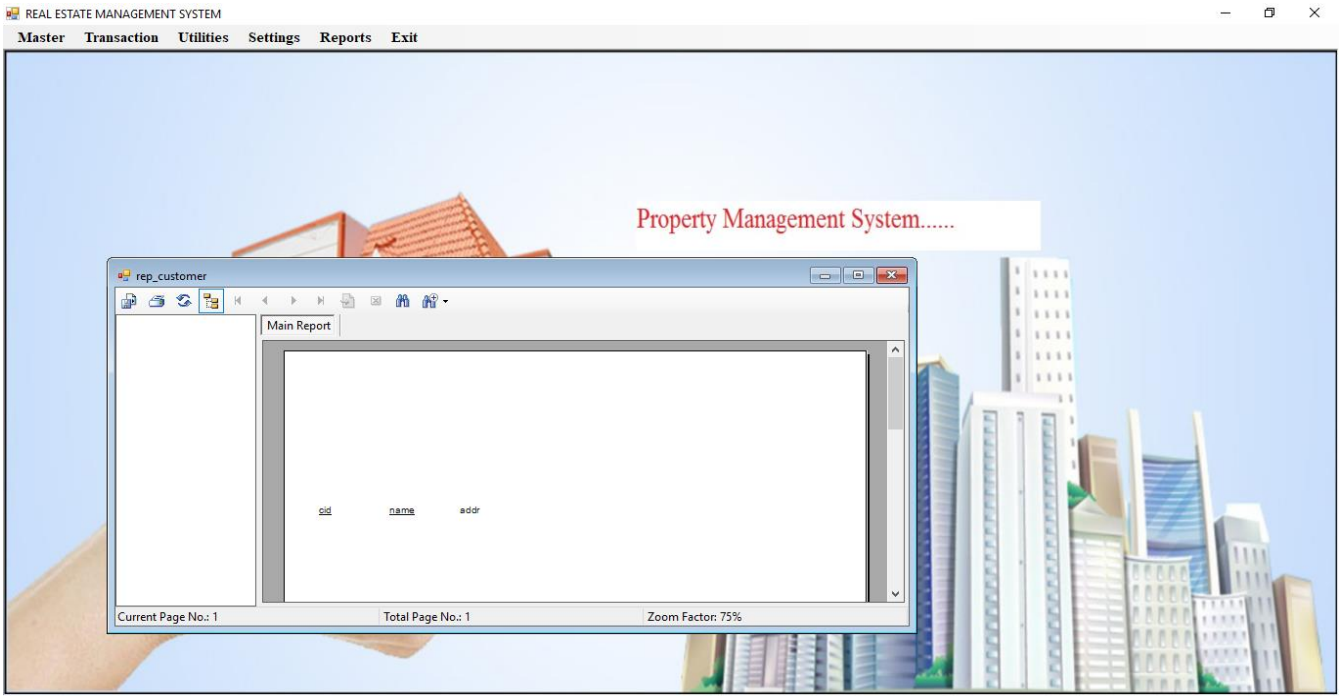
7. INSTALLMENT FORM



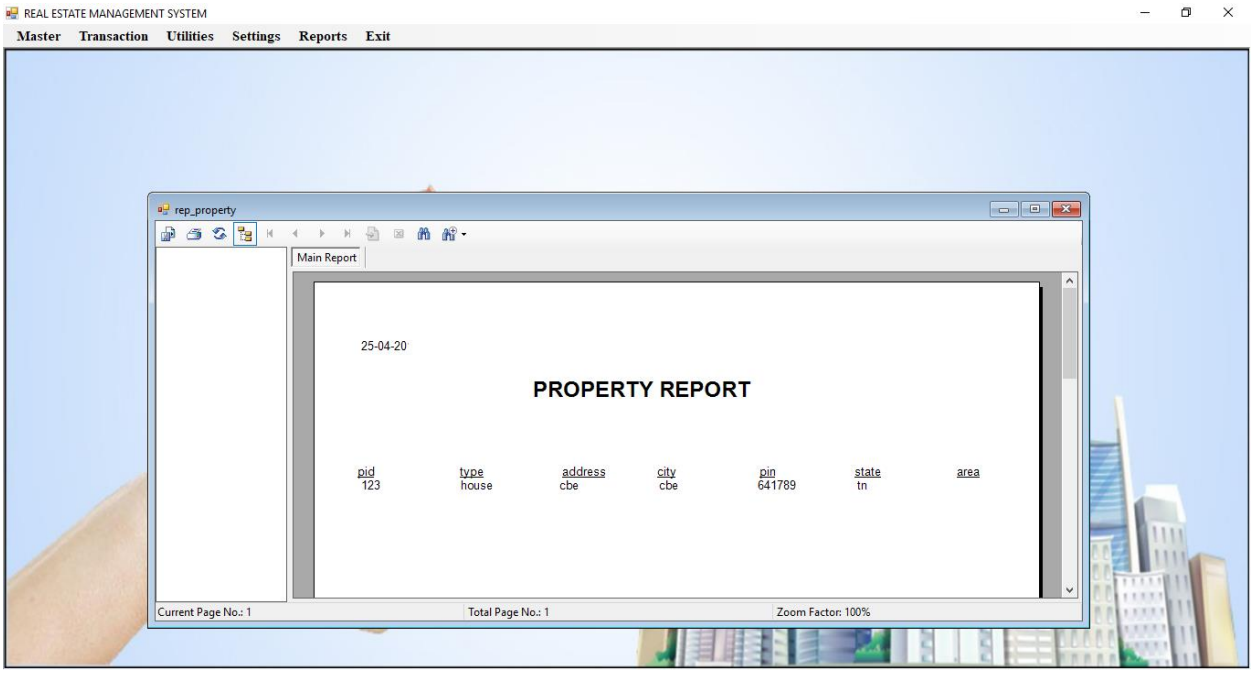
8. CHANGE PASSWORD



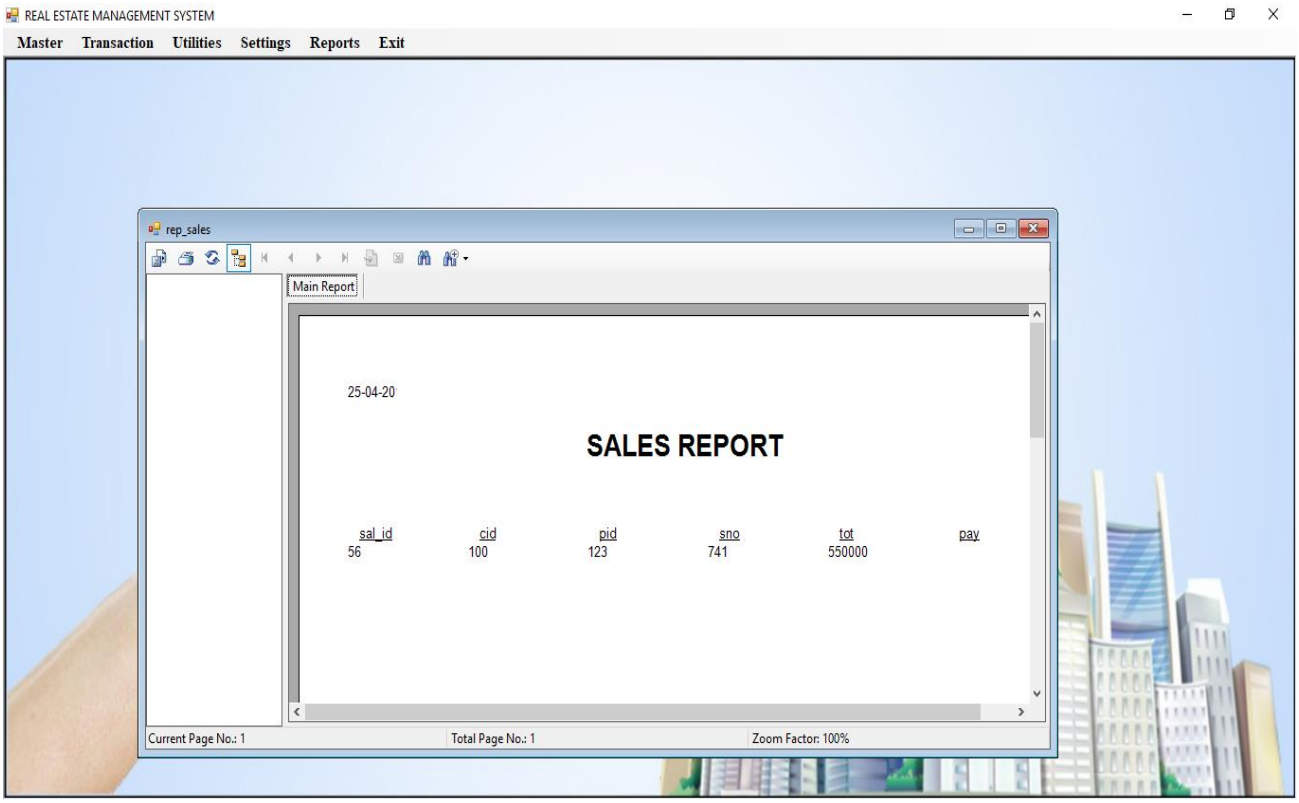
9. CUSTOMER REPORT



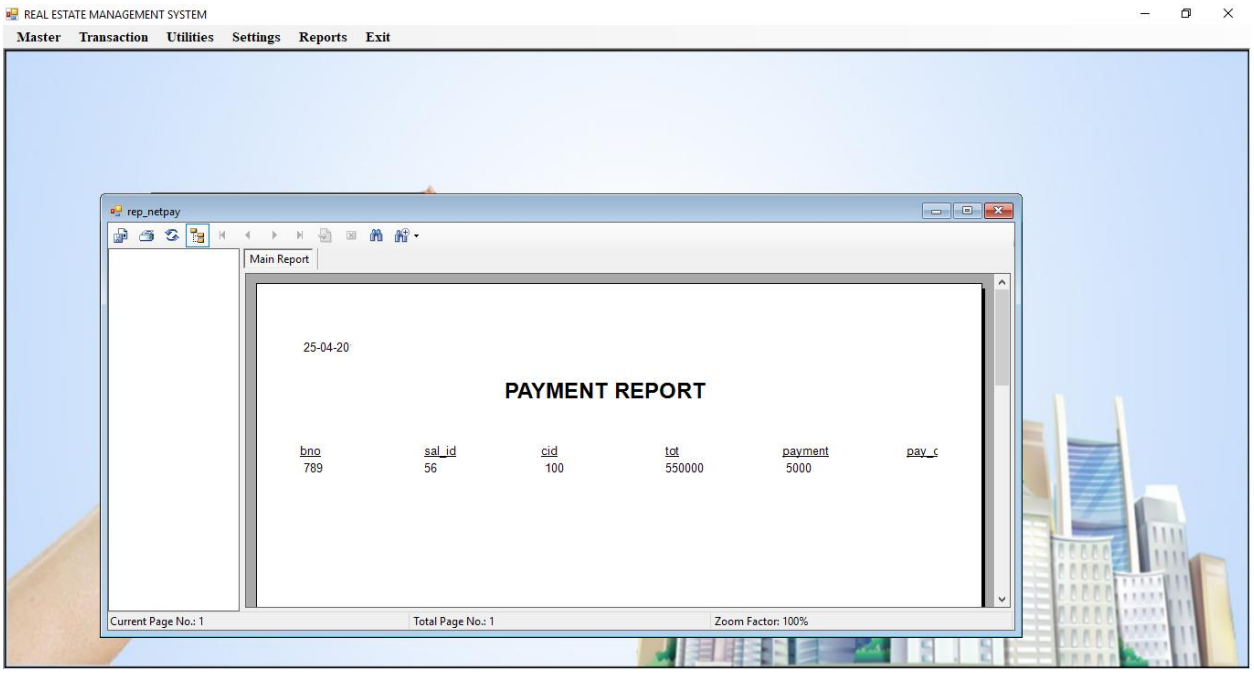
10. PROPERTY REPORT



11. SALES REPORT



12. PAYMENT REPORT



CHAPTER V

5. SYSTEM TESTING

5.1 INTRODUCTION

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

5.2 TYPES OF TESTING

- Unit testing
- Integration testing
- Verification testing
- Validation testing
- White Box testing
- Black box testing
- Test data output

UNIT TESTING

Unit testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed.

A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

INTEGRATION TESTING

The entire project was split into small program; each of this single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructors were used. It give a lot of problem by not functioning is an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. when the frames where given for the test, the end user gave suggestion. Based on their suggestions the frames where modified and put into practice.

VERIFICATION TESTING

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements.

Verification ensures that the product is built according to the requirements and design specifications. The artefacts such as test Plans, requirement specification, design, code and test cases are evaluated.

VALIDATION TESTING

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., Validation succeeds when the software function in a manner that can be reasonably Accepted by the customer.

WHITE BOX TESTING

White box testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester.

The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

BLACK BOX TESTING

Black box testing is done to find out the following information as shown in below:

1. Incorrect or missing functions
2. Interface errors.
3. Errors or database access.
4. Performance error.
5. Termination error.

The mentioned testing is carried out successfully for this application according to the user's requirement specification.

TEST DATA OUTPUT

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

CHAPTER VI

6. SYSTEM IMPLEMETATION

6.1 INTRODUCTION

The purpose of System Implementation can be summarized as making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the Consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system development to a system support and maintenance mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

List of System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. Planning means deciding on the method and the time scale to be adopted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The line managers controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider;

The implications of system environment are

- Self selection and allocation form implementation tasks.
- Consultation with unions and resources available.
- Standby facilities and channels of communication.

The purpose of prepare for system implementation is to take all possible steps to ensure that the upcoming system deployment and transition occurs smoothly, efficiently, and flawlessly. In the implementation of any new system, it is necessary to ensure that the Consumer community is best positioned to utilize the system once deployment efforts have been validated. Therefore, all necessary training activities must be scheduled and coordinated. As this training is often the first exposure to the system for many individuals, it should be conducted as professionally and competently as possible. A positive training experience is a great first step towards Customer acceptance of the system.

During System Implementation it is essential that everyone involved be absolutely synchronized with the deployment plan and with each other. Often the performance of deployment efforts impacts many of the Performing Organization's normal business operations. Examples of these impacts include:

- Consumers may experience a period of time in which the systems that they depend on to perform their jobs are temporarily unavailable to them. They may be asked to maintain detailed manual records or logs of business functions that they perform to be entered into the new system once it is operational.
- Technical Services personnel may be required to assume significant implementation responsibilities while at the same time having to continue current levels of service on other critical business systems.
- Technical Support personnel may experience unusually high volumes of support requests due to the possible disruption of day-to-day processing.

Because of these and other impacts, the communication of planned deployment activities to all parties involved in the project is critical. A smooth deployment requires strong leadership, planning, and communications. By this point in the project lifecycle, the team will have spent countless hours devising and refining the steps to be followed. During this preparation process the Project Manager must verify that all conditions that must be met prior to initiating deployment activities have been met, and that the final 'green light' is on for the team to proceed.

The final process within the System Development Lifecycle is to transition ownership of the system support responsibilities to the Performing Organization. In order for there to be an efficient and effective transition, the Project Manager should make sure that all involved parties are aware of the transition plan, the timing of the various transition activities, and their role in its execution.

Due to the number of project participants in this phase of the SDLC, many of the necessary conditions and activities may be beyond the direct control of the Project Manager. Consequently, all Project Team members with roles in the implementation efforts must understand the plan, acknowledge their responsibilities, recognize the extent to which other implementation efforts are dependent upon them, and confirm their commitment.

6.2 SYSTEM MAINTENANCE

All system is dynamic and subjects to constantly changing requirements. Effort must be devoted to adapting them and design must be flexible specified so that such changes can be easily implemented. This activity is called system maintains. It includes improvement of system functions and correction of errors.

Back up for the entire database files are taken and stored in secondary storage devices like magnetic tapes and disks so that is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintains of the system.

The master file has flags for maintains. After the mentioned period, the rejection suppliers, unused data in the files will be deleted in the master file. This method is the increasing the memory to store the data.

CHAPTER VII
FUTURE SCOPE AND CONCLUSION

In future all transactions will be processed in a secure manner and can find the intruders activity by getting all relevant details. This project provided practical knowledge of not only programming in VB.NET web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with “Property management system”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently. This facilitates to maintain installment details of customer. Through this the total amount and the balance to be paid by the customer are verified. The sales information in the database will automatically generate bills when the customer buys property. The bill history can be retrieved promptly. A message is sent to customer before the payment date. And reports will be generated based on different criteria’s.

BIBLIOGRAPHY

Books Referred:

- Alex Homer , **'Professional VB.NET 1.1'**, 2004 Edition, Wrox Publications
- Clayton crooks II **'Learning Visual Basic .Net Through Applications'**
- Roger S Pressman, **'Software Engineering'**, 2000 Edition, Dreamtech Publications
- Steven Holzner, **'Visual Basic.NET Black Book'**, 2003 Edition, Dreamtech Publications
- Bill Hamilton, "**Programming SQL Server 2005**", O'Reilly Media Publisher, 2006.
- Elias M.Award,"System Analysis and Design", Galgotia Publications, Second Edition.
- Daniel Solis, "Illustrated VB.NET 2008", Apress Publisher, 2008.
- David B. Makofske, Michael J. Donahoo, Kenneth L. Calvert, "TCP/IP Sockets in VB.NET", Academic Press Publishers, 2004.
- Richard Blum, "VB.NET Network Programming", John Wiley & Sons Publishers, 2006.
- Robin Dewson, "**Pro SQL Server 2005",** Apress Publisher.

Websites:

1. www.msdn.microsoft.com
2. www.vbcity.com
3. www.vbdotnetheaven.com
4. www.codeproject.com
5. www.dotnetjohn.com