**KOLHAN UNIVERSITY**

**PROJECT REPORT ON**

**“society MANAGEMENT SYSTEM”**

***Submitted for the partial fulfillment of the requirement to award the degree of:-***

**BACHELOR OF SCIENCE IN COMPUTER APPLICATION**

**IN JAVA 6.01**

**SUBMITTED TO:-**

**GRADUATE SCHOOL COLLEGE FOR WOMEN**

**(A CONSTITUENT UNIT OF KOLHAN UNIVERSITY)**

**Submitted By:**

* **Baby muskan**

**(150505943192)**

**Guided By**

**MUKESH KUMAR**

**CERTIFICATION**

**GRADUATE SCHOOL COLLGE FOR WOMEN,SAKCHI**

**DEPATMENT OF VOCATIONAL COURSES**

****

This is certify that the project report entitled “SOCIETY MANAGEMENT SYSTEM” has been prepared by BABY MUSKAN in partial fulfillment for the award of degree in ‘Bachelor of Science in Computer application’ at GRADUATE SCHOOL COLLEGE FOR WOMEN, SAKCHI and is a record of project work carried out under the proper guidance and supervision.

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are grateful to all the computer teachers who helped us directly and indirectly in the preparation of the given job training report.

We express my sincere gratitude and thanks to (Our Principal) for providing an opportunity to work on the project “SOCIETY MANAGEMENT SYSTEM”.

We are also thankful to MR MUKESH KUMAR for the valuable practical and theoretical guidance and constant encouragement that have been a source of inspiration to my entire project.

Here are many personalities who deserve acknowledgement and gratitude for their contribution to the development and presentation of the project and I take this opportunity to thanks all of them.

Contents

**1. Introduction 12 - 13**

**2. Objective 5**

**3. System Analysis**

3.1. *Working Environment 6*

3.2. *Hardware and Software requirement 7*

**3.3. *Feasibility Study 8-11***

**4. System Design**

4.1. *E-R Diagram 28-29*

**4.2. *Data Flow Diagram 30-34***

4.2.1. Zero level or Context level DFD 32

4.2.2. 1st level DFD for Order Confirmation 33

4.2.3. 2nd level DFD for Bill Generation 34

**4.3. MODULES 14-27**

**4.4. Data Dictionary 35**

**5. SOURCE CODING 41-151**

**6. System Implementation**

6.1. *Implementation 152*

6.3 Testing phase 155

**7. Result and Discussion**

7.1. *Conclusion 156*

7.2. *Future Scope and Limitations 157*

**8.Bibliography 158**

OBJECTIVE

Objectives means the various usages of one system. By clarifying the system objectives one can be much clearer about the processes he/she has to perform.

The objectives of our proposed system are: Keep records of all flats , duplexes , owners and staffs of a particular society. Generates quick Result and booking details of each and every flat and duplex of a society.

Automatic generate stud flat/duplex no for each every flat and duplex as well owner, staff , booking and payment id for each payment made.

We can modify and edit details of flat’s and records of owner, staff, booking, cancellation, payment of each booking or payment receipt. It stores information of empty and occupied flats and duplexes. We can view total details of each module and purchasing of each flat or duplex in month or a day.

It saves the time and make the work easy for desk user. Maintains the whole information about a society. The system can allow us to select unnecessary information and delete them from database. Here administrators have total control on the system.

It will be easily used and the time consuming is decreased.

Working Environment

The entire system has been developed in Java 6.01 as a front-end and Microsoft Access as the back-end.

**Java:** Java is a high level programming language developed by Sun Microsystems in 1991.It took 18 months to develop the first working version. This language is initially called OAK and was renamed JAVA in 1995.

Java has the following features:-

* Java is platform Independent.
* It follows the concept of OOPs.
* Internet based programming language.
* Easy to read, write, debug.

**Microsoft Access:** Microsoft Access is a tool in MS-Office package. It is a database which is used to store data in form of rows and columns. It is used as a container to store data. It provides the user with the facility of inserting, deleting, updating and retrieving data from it. It is used as a backend in combination with a frontend designing software in an application. MS-Access makes use of different queries for different operations to be performed.

Hardware and software requirements

**Hardware requirements:**

* Processor : I3 and above
* Ram : 2 GB RAM and above
* 100 MB Hard Disk
* Monitor 15’
* DVD read write(for backup)
* Printer : Laser(for Printing receipts)

**Software requirements:**

* Operating System: Windows 7 and above
* Frontend : JAVA c 6.0
* Backend : Microsoft Access
* Interface : 32 Bit ODBC interface

FEASIBILITY STUDY

* Prior to stating whether the system we have to develop is feasible or not we believe that we should emphasize on what is implied by the word “Feasibility”. Feasibility is the measure of how beneficial or practical the development of the system will be to an organization. It is a preliminary survey for the system investigation. It aims to provide information to facilitate a later in depth investigation.

**Technical Feasibility**

* Based on the outline design of the system requirements in terms of inputs, outputs, procedures, the technical issues raised during technical feasibility include:
* Does the necessary technology exist to do what is proposed?
* Does the proposed equipment have the technical capacity to hold the data required to use in the new system?
* Adequate responses provided by the proposed system?
* Is the system flexible enough to facilitate expansion?
* Is there any technical guarantee of accuracy, reliability, ease of access and data security?
* The system developer’s task is to view needed capabilities in light of currently available technology. Our application works hand in hand with high technology. A database has to be maintained in order to update and backup data whenever required. To create databases we take Microsoft Access. After taking the above facts into consideration we can state that the new proposed system is technically feasible.

**Behavioral Study**

* A proposed system is beneficial only if it can be turned into an information system that will meet the operational requirements of an organization. A system often fails if it does not fit within existing operations and if users resist the change.
* Important issues a system developer must look into are:
* Will the new system be used if implemented in an organization?
* Are there any major barriers to implementation or is proposed system accepted without destructive resistance?
* The whole system of computerizing the Institute Management System is to handle the work much more accurately and efficiently with less time consumption. There will be additional work to be completed, because now the institute will have to maintain databases of their courses, faculty and student’s and the asset in their store.
* Compared to the semi-computerized system the chances of avoiding errors in a computerized system is much higher because the user need not stress himself unnecessarily resulting in recklessness. Unlike the semi-computerized system there would be backup data for all the information concerning the daily transaction occurred within the organization.

Another important fact to be regarded is the security control, which is handled by the system. Since data falling in wrong hands could make things tough. After taking the above facts into consideration we can state that the operating of the proposed system within the organization is feasible.

In this phase of the feasibility study the following two main topics are:

* Technical Performance Aspect
* Acceptance within the organization

Technical performance aspect is explained in the technical feasibility report and there is new information needed in this to explain it again, but as for the acceptance within the organization the following points are important and those are explained according to the topics.

**Whether the system provides right information to the right place?**

In the current system which is the semi computerized system the information may be last in the process of sending from one place to another. This is mainly due to human interaction in the process of the transferring information from one place to another.

**Whether the new system affects the current users in the system?**

The new proposed system will affect the user in

the following areas:

* Accuracy
* Efficiency
* Productivity
* Robustness
* Lesser time consuming

**Economic Feasibility:**

In making recommendations a study of the economy of the proposed system should be made. Even though finding out the costs of the proposed project is difficult. We assume and estimate the costs and benefits. According to the computerized system we propose, the cost can be broken in two categories.

Costs associated with the development of the system.

Costs associated with operating the system.

The proposed system database has been developed in Microsoft Access which is already installed in the system in Microsoft Office package; therefore the system is economically feasible.

INTRODUCTION

Society management system is a large system which can be used for managing an society’s day by day activities. Society management system allows users to store society’s information electronically, including information of the flats, duplexes,owners,staffs etc. Most importantly, this information can be easily shared with authorized users. Records can be easily searched and reports can be easily generated.

SMS is configurable and can be configured to meet most individual Society’s needs. The first truly scalable society management package with power to revolutionize the way the societies are run. The software is more than just another technology solution. It is a system that will improve the way society is managed.

SMS is policy driven process, which requires customization as per our client’s process. The design is an easy process to understand, handle or customize the product by creating society’s specific rules.

This is an attempt to create a project “Society Management System” through which we will show all the working of an society being automated. This project can make the tedious and mistake prone job easier, accurate and faster. It can be used both in top level and bottom level management’s daily operations.

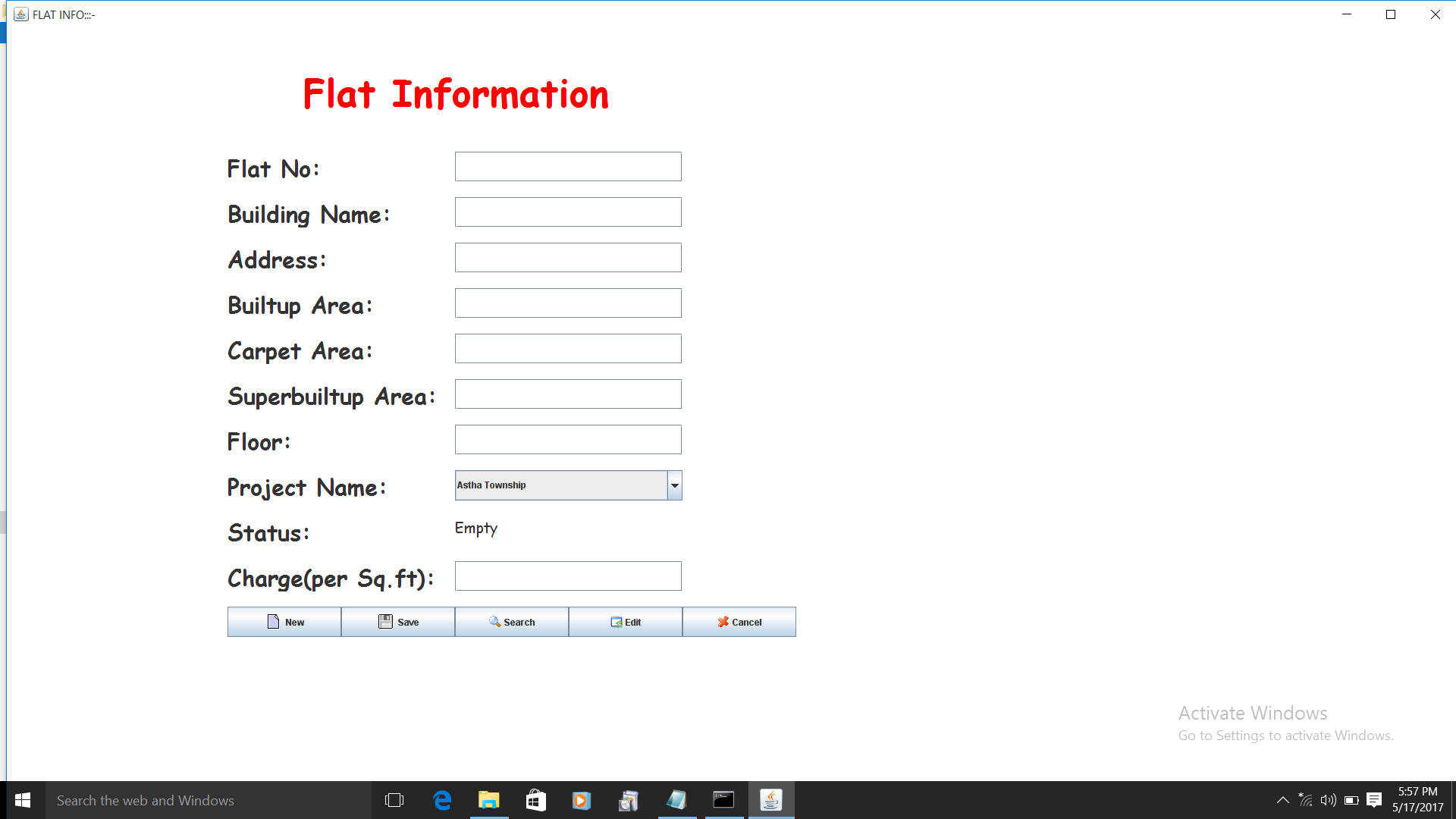
All the facilities like flat/duplex information, owner/staff information, booking, payment management etc will be added, deleted, updated and saved in the project that is when a new customer will come for booking, all the information regarding the customer will be filled like in which flat/duplex he/she will be alloted, payment charts, working staff’s information. We can easily see and maintain the number of vacant flats that are there in the society. The application will be scalable and easily maintained. It could be easily customized to the changing needs of the user.

Society management system is a simple yet powerful one joint integrated platform that connects all modules like flat/duplex details, owner details, payment and many more specialized modules.

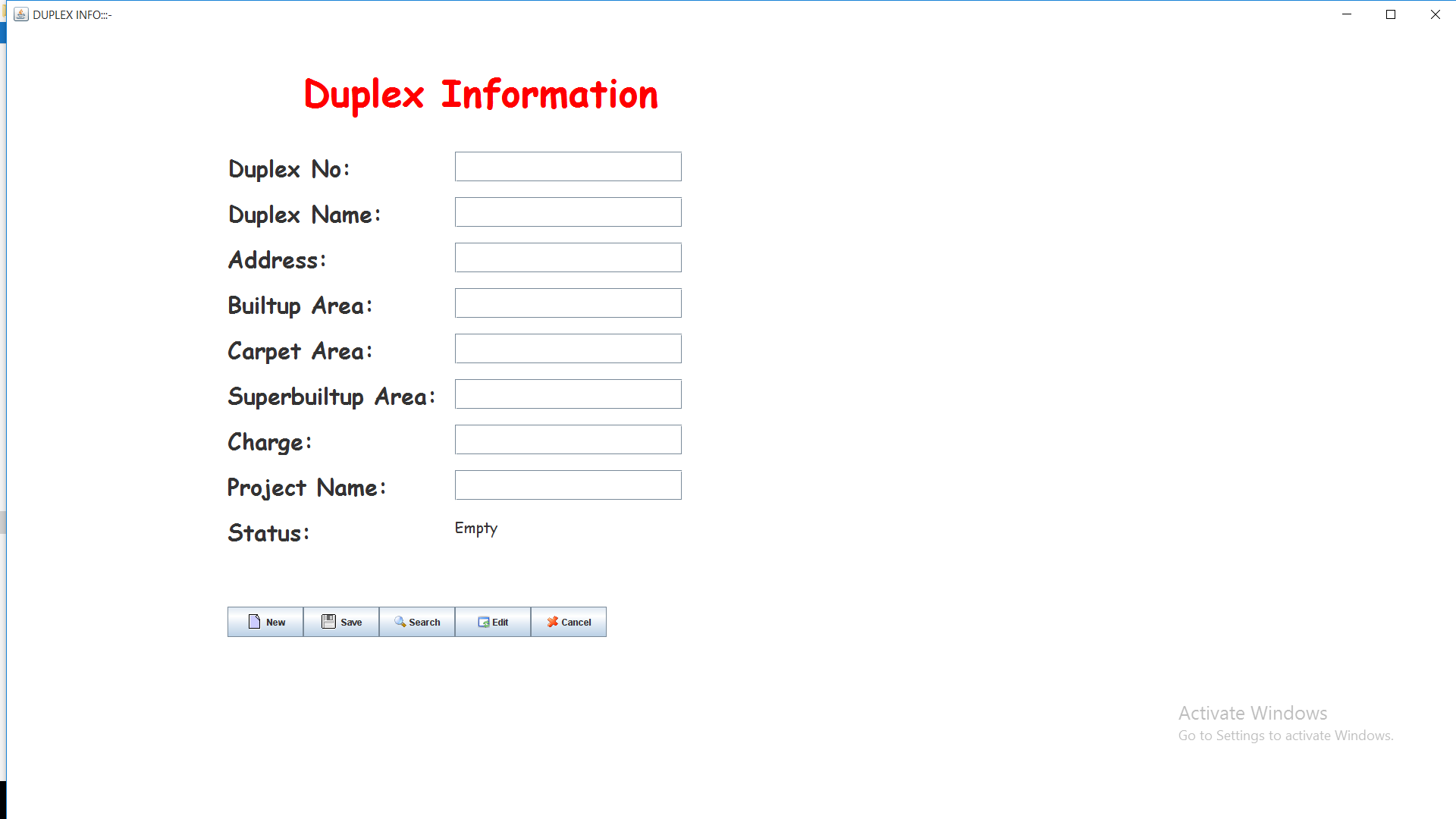
Main aim in developing a society management system is to provide an easy way not only to automate all the functionalities of a society, but also to provide full functional reports to top management of the society with the finest of details about any aspect of society.

**Areas covered in an society management system are:**

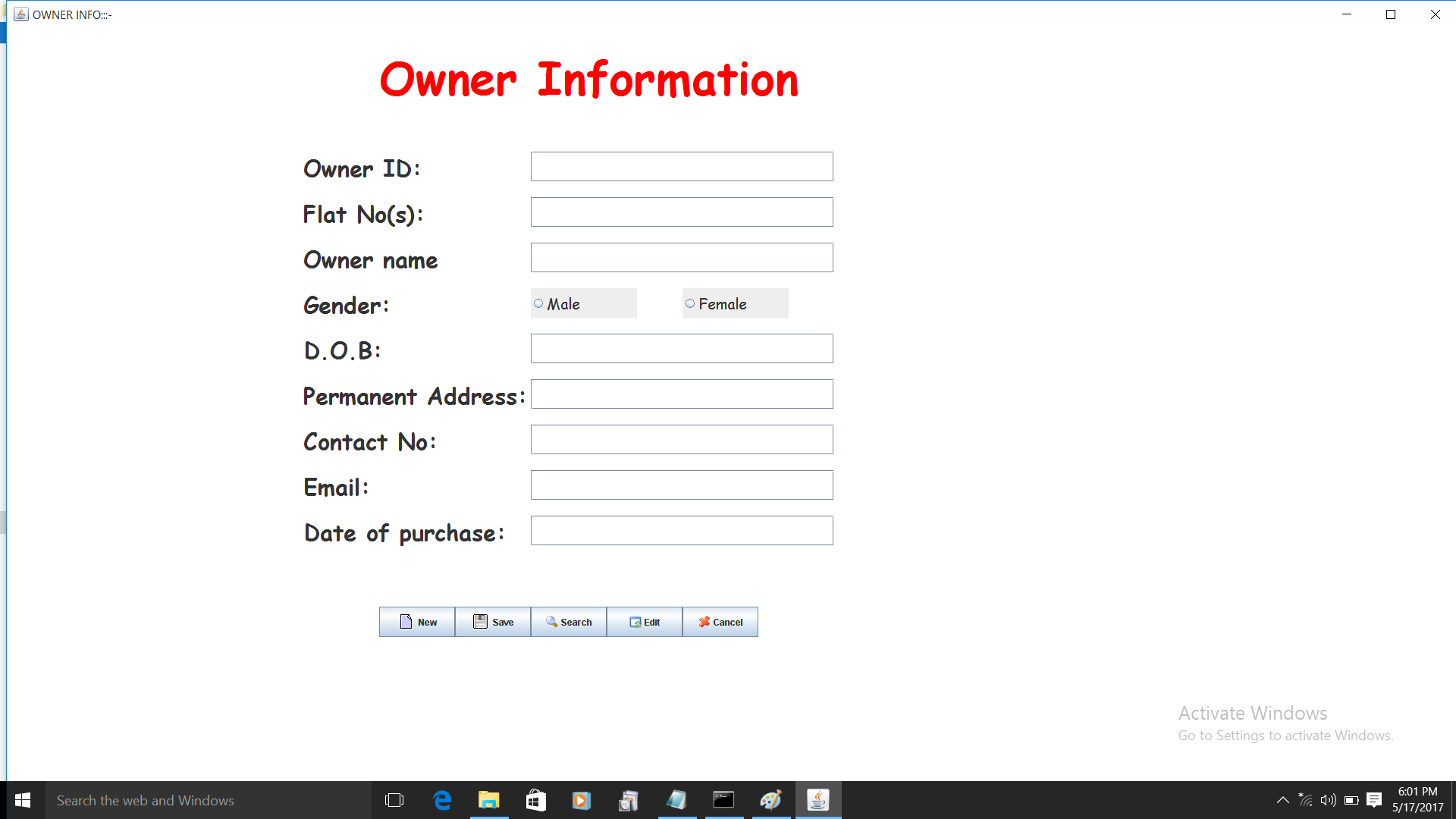
**flat module:**Flat module include all the information regarding to flat that is how many flats are available, flat id for each flat available, flat name, project name, flat price. We can edit and modify the records of flat module like adding a new flat ,changing the flat price according to requirements . In flat module we save our flat record in database. While defining the flat we can also define flat type and their status. At the time of booking, we can easily know the status of the flat.

****

**Duplex management:** This module is a Master module. This module enables us to store data regarding the duplex which can be updated time to time. Here we save duplex information like duplex no., duplex name,, project name, charge, address and status.

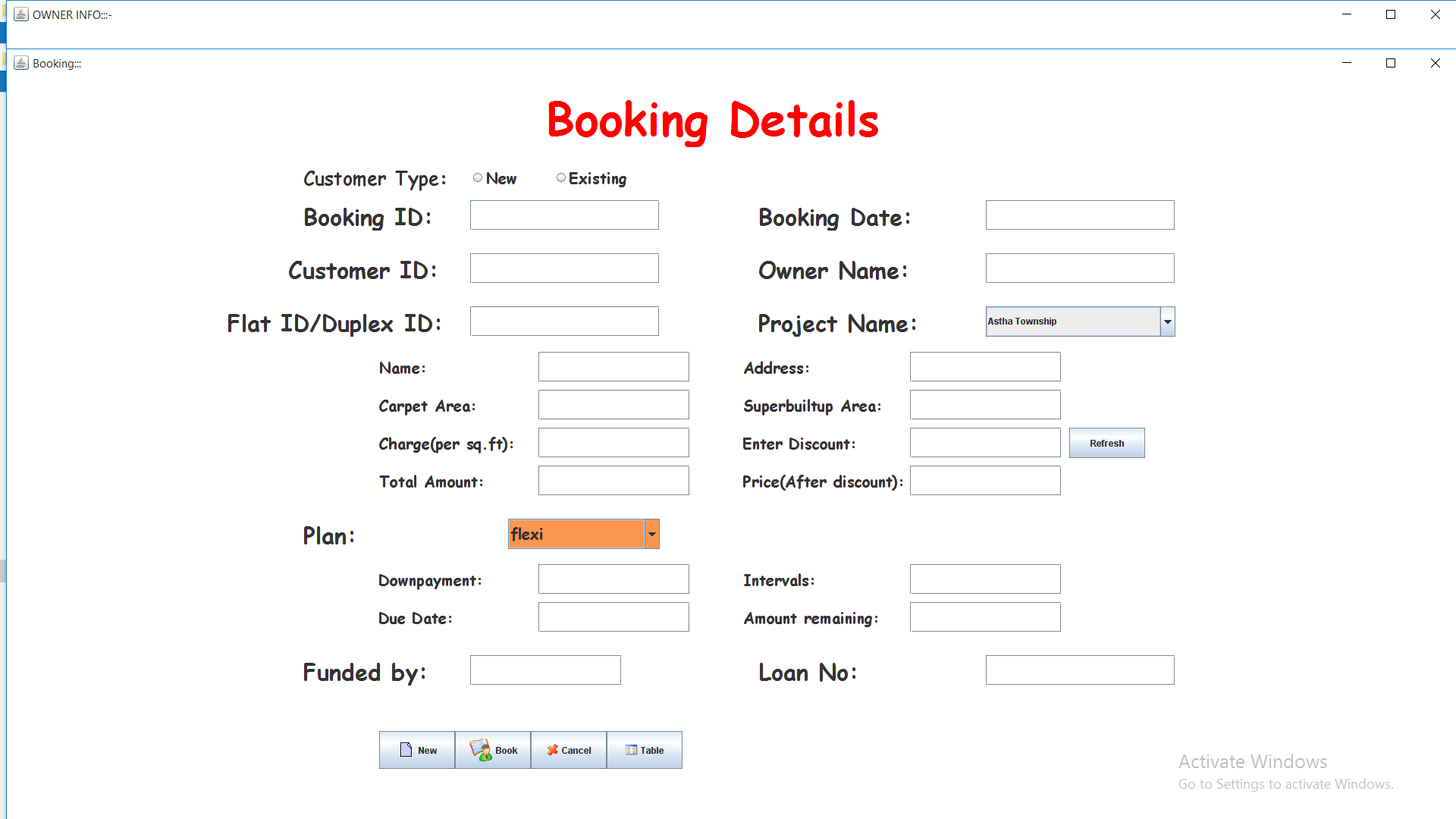


**Owner module:** Owner module helps to feed the details like owner id, flat /duplex owned i.e its no , information about the its area and charge etc. Here we can display the status of the duplex. It integrates with all the related modules like if a particular duplex is purchased or booked by an owner or a customer then the status of the duplex is updated and if the same duplex is again tried to be booked then it displays a message about the booking of the duplex.

****

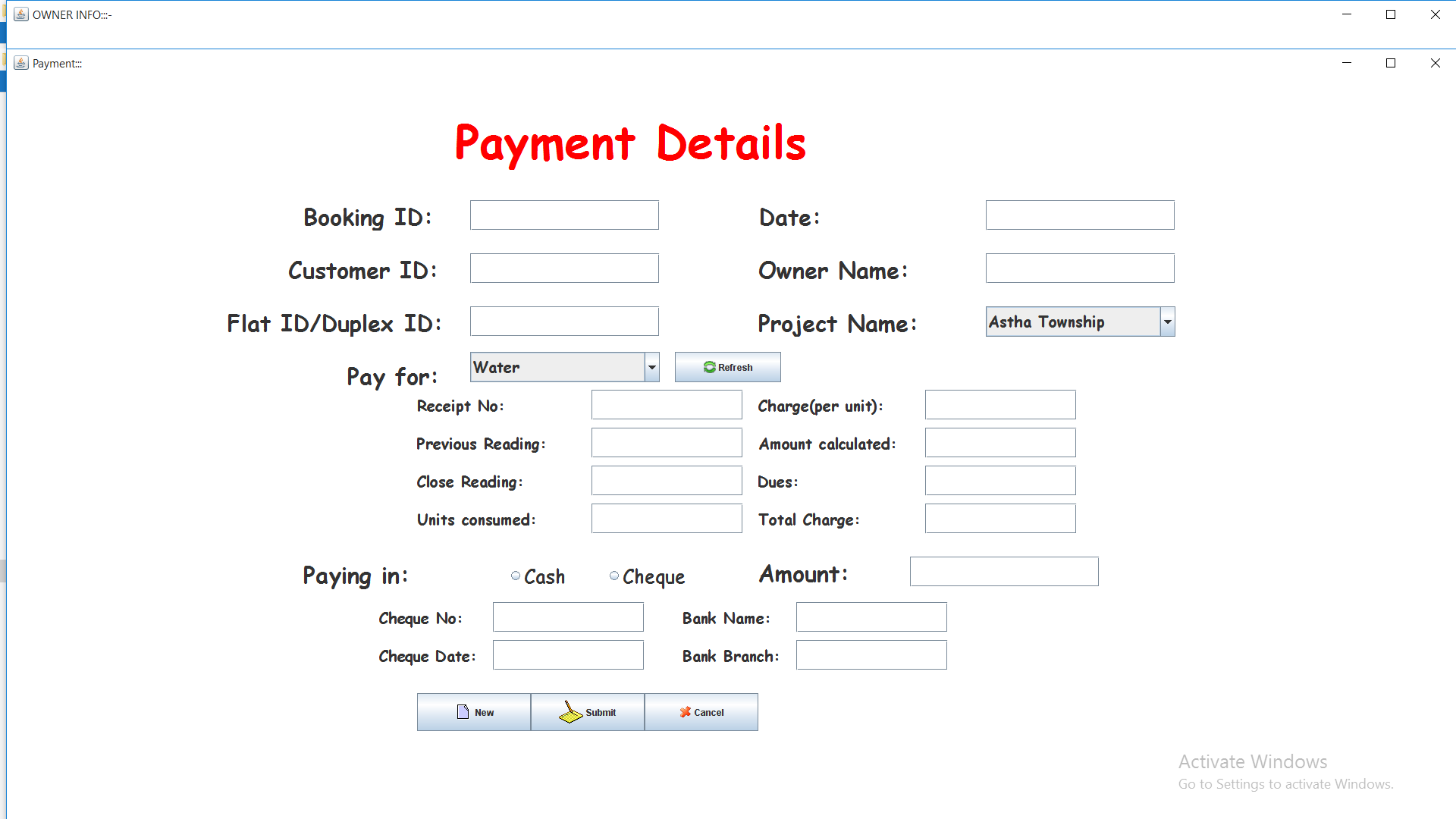
**Staff module:** This module is very much helpful to manage all the staff personal information and information can be updated time to time as per requirement. Here we store the details like staff id, staff name,staff address, staff qualification, gender, staff contact no. etc, we can view staff all details of each staff in a table we can also search each staff details by their staff id.

**Booking module**: Thebooking module is major operation for the society. Also its automates the payment receipts entries in the society accounts eliminating the need of repeated data entries.the booking module has a feature that when a flat/duplex is allotted , its status is checked that whether it is empty or vacant and if its is empty , then only it can be selled or purchased and its status is updated to occupied in other correspondent tables and the booking module also deals with the instalment plans for the payment of the cost.

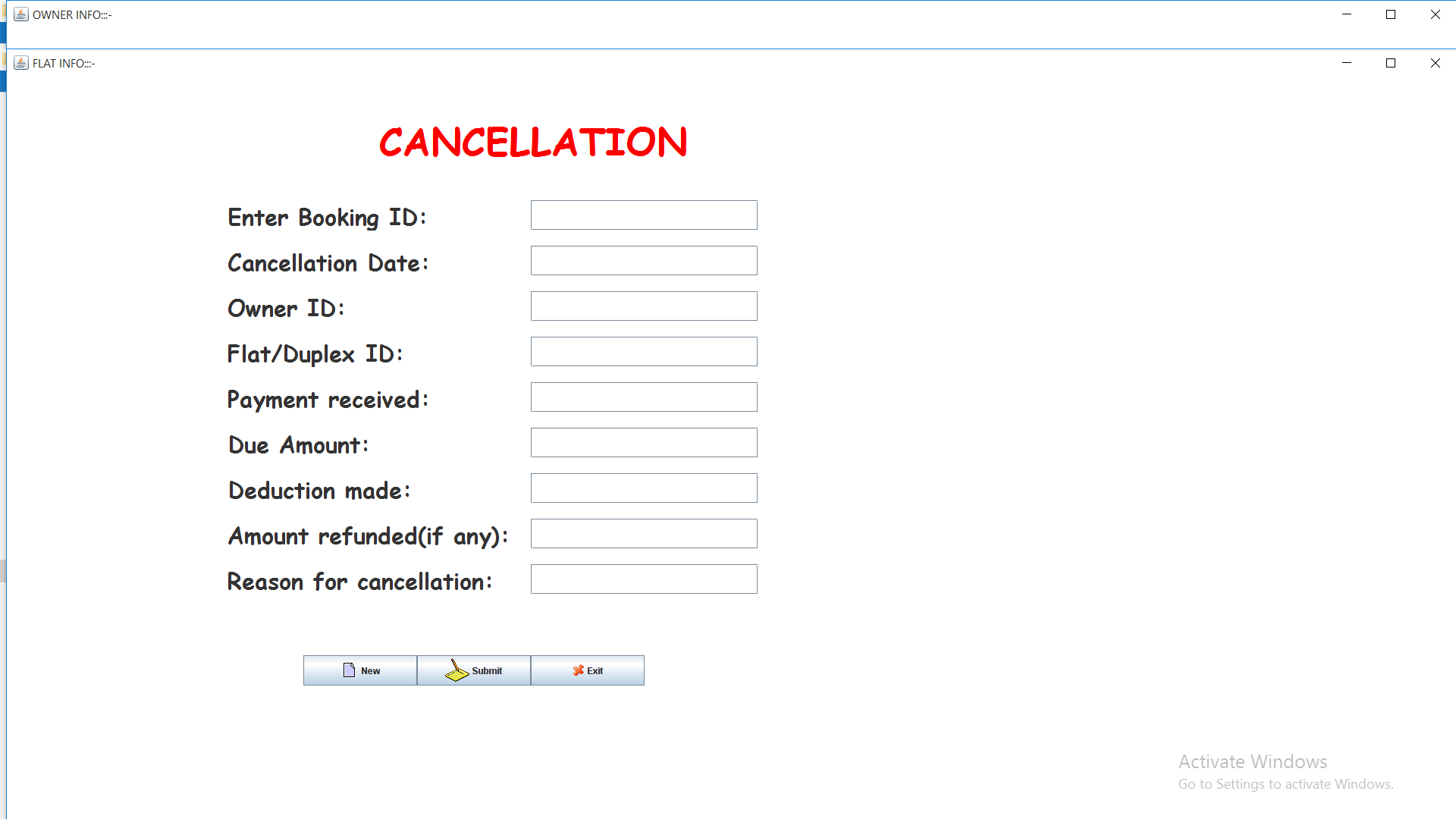
****

**Payment Module**:

Thepayment module is a critical operation for the society. Also it automates the payment receipts entries in the society accounts eliminating the need of repeated data entries.the payment module has several features which facilitate the user and makes the task of payment much aeasier and safe .The payment module mainly deals with the payment of the water ,electricity and maintenance charges of the society which are the basic needs of every society. The payment module also deals with the payment of instalments and generates the payment receipt with a unique receipt no.for each payment made.



**Cancellation Module:** The cancellation is the module which covers the cancellation information of a flat/duplex. In this module , the reason of cancellation is also provided. Every society has a different scheme but most of the time,the amount paid during the purachasing is not fully refunded if the booking is cancelled for some reason . So,may be 30-40% would be deducted and rest of the amount is returned back to the customer . So,this module completely deals with the cacellation of a flat/duplex by a customer which has been booked and the downpayment has been done.



**DRAWBACKS OF MANUAL SYSTEM**

* ***Redundancy of data:***

Due to improper maintenance of data, inconsistency is there which leads to problem like duplication of data. The same data gets repeated over and over since we find it hard to keep track of the documents, information and transactions.

* ***Difficulty in updating the data:***

Problem of updating data in existing system since everything is stored in registers and files. It is very difficult and time consuming to update the data. If there are any changes to be made, the data will have to be entered again. At times the user would forget to make the changes or forget that they had already altered it and might redo it again, it’s again time consuming

* ***Non-Centralized data:***

In the manual system, data records are kept under file management system, due to this system the placement of data is not at one particular place. This results in more confusion and more consumption of time for the proper maintenance of records.

* ***Delay in retrieving information:***

Retrieving information is quite time consuming process because of large volume of data, if the data is required. The information and the details are stored in different parts of the site and so takes a long time to retrieve the data, it takes a long time to find information about a relevant person.

* ***Lack of security:***

Since data is stored in filling cabinets it is freely available to anyone. If the information falls into the wrong hands it can be harmful for the institute. The data can be destroyed permanently from the files.

* ***Inconsistency of data:***

There will be unavailability for future use, since data might get misplaced during the manual filling. So data won’t be preserved properly for future use.

* ***Integrity of data:***

Data in manual systems is typically accessible by more than one person at one time and is vulnerable to being lost or misplaced. If an employee takes a folder of data from the database to his/her desk, the data would no longer available for others.

***ADVANTAGES OF A PROPOSED SYSTEM***

When people live in housing societies, they need to share the responsibility jointly to handle operational stuff such as paying utility bills, maintenance of the premises, security, and legal issues. Since they are time-consuming things, people search for efficient methods such as society management tools. With end-to-end service management and work management, they are ideal for small, medium and large enterprises. When people manage societies using software, they build transparency in the system. They also save time and make the work easy.

Managing the things become simple when you do it using software. Workflow-based systems keep all concerned people aware of the happenings. With sophisticated Email and SMS alerts, online communications, and contact and document management modules; members find it convenient and fast.

**Is it user-friendly?**

Yes, modern software modules are quite user-friendly. Since they are made for multiple things and versatility, the modular structure is followed by the designers.  Most of the things are configurable, and one-time setup is required. Software development companies that bring society management modules help their clients in setting up the base data. With a little bit training, people can use it without any trouble.

Since it is possible to interface the software with external software, society management becomes as simple as ABC.

**Benefits of society management software**

Well, there are several benefits of implementing full-fledged society management software in the housing society.

* Manage accounting and financial transaction.
* Integrated member management system. Reminders through email, SMS and e-receipts make it useful.
* Transactional and MIS reports offer comprehensive, at-a-glance view.
* Reconciliation of bank management accounts.
* Privacy of data.
* Effective user-management, login-based access controls, and privileges.

E-R (ENTITY RELATIONSHIP) DIAGRAM

An ER (entity-relationship) diagram is a specialized graphic that illustrates the relationship between entities in a database.

ER diagram often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to present relationships and Ovals are used to represent attributes.

1. The entities are represented by the Rectangle shape.

2. The relationships are presented by the Diamond shape.

3. The attributes are presented by the Oval shape.

DATA FLOW DIAGRAM (DFD)

Data Flow Diagram is a process modeling tool. It traces the flow of data through different processes in the system. It helps in easy understanding of the system. Data Flow Diagram is a graphical representation of the ‘Flow’ of data through an information system. Often they are preliminary step, used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (Structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to and where the data will be stored. It does not show the information about the timing of the process or information about whether the process will operate in sequence or in parallel. It is a technique; we are benefitted from, particularly before we go through business process. A Data Flow Diagram looks at how data flows through a system. It concerns things like where the data will come from and go, as well as where it will be stored. But we won’t find information about the processing timing.

We usually begin drawing a context diagram, a simple representation of the whole system. To elaborate further from that we drill down to level 1 diagram with additional information about the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on are possible but anything beyond level 3 is not very common.

**DFD Symbols:**

In DFD, there are four symbols used:

**Symbol**  **Meaning**

* **A Rectangle** defines a source

(originator) Destination or

Entity of system data.

* **An** Arrow identifies data flow

(data in motion). It is a

pipeline through which

information flows.

* **A Circle** or **an Oval** Bubblere-

presents a process that trans-

forms incoming data flow(s)

into outgoing data flow(s).

* **An Data** Source is a data store

Data at rest, or temporary

repository of data.

**MASTER TABLES**

* **Table name**: **Flat table**
* **Description: Stores the Flat details**
* **Primary Key: Flat\_id**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Staff\_no | Text |
| Staff\_name | Text |
| Staff\_address | Text |
| Staff\_adhar | Number |
| faculty\_gender | Text |
| Phone\_no | Number |
| Designation | Text |
| PostOffice | Text |
| PinCode | Text |

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Flat\_id | Text |
| Flat\_name | Text |
| Address | Text |
| Carpetarea | Number |
| Builtuparea | Number |
| Superbuiltuparea | Number |
| Floor | Number |
| Projectname | Text |
| Fstatus | Text |
| Charge | Number |

* **Table name: Duplex table**
* **Description: Stores the Detail of duplex**
* **Primary Key: Duplex\_id**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Duplex\_id | Number |
| Duplex\_name | Text |
| Address | Text |
| Carpetarea | Number |
| Builtuparea | Number |
| Superbuiltuparea | Number |
| Fstatus | Text |
| Charge | Number |

* **Table name**: Staff**tbl**
* **Description: Stores the Staff details**
* **Primary Key: Duplex\_id**
* **Table name**:Owner**tbl**
* **Description: Stores the Owner details**
* **Primary Key: Owner\_id**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Owner\_id | Number |
| Owner\_name | Text |
| Address | Text |
| gender | Text |
| Phone\_no | Number |
| email | Text |
| Bank Account No | Number |

**TRANSACTION TABLES**

**Table name**: **Bookingtbl**

* **Description: Stores the booking details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| book\_id | Number |
| Book\_date | Date |
| Customer\_id | Number |
| Flat\_id | Text |
| Plan | Number |
| downpayment | Number |
| Remamt | Number |
| Due\_date | Date |
| Interval | Number |
| Bank\_name | Text |
| Loan\_no | Text |
| Installment | Number |
| Discount | Number |
| Total\_amt | Number |
| Status | Text |

**Table name:Cancellationtbl**

* **Description: stores the details of the cancellation made.**
* **Primary Key: None**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Book\_id | Text |
| Owner\_id | Text |
| Flat\_id | Text |
| Deduction | Number |
| Amt\_refund | Number |
| Reason | Text |

Source code

LOGIN Form:

import javax.swing.\*; import java.awt.\*;

import java.awt.event.\*; import java.applet.\*;

import java.io.\*; import java.util.\*; import java.sql.\*;

public class login extends Frame implements ActionListener

{

JButton btnLogin=new JButton("Login");

JButton btnCancel=new JButton("Cancel");

TextField txtUsername=new TextField(30);

TextField passwordField=new TextField(20);

JLabel lblpic=new JLabel();

public login()

{

setLayout(null);

txtUsername.setBounds(198,144,148,25);

passwordField.setBounds(198,175,148,25);

btnLogin.setBounds(146,220,85,25);

btnCancel.setBounds(244,220,85,25);

add(txtUsername);add(passwordField);

add(btnLogin);add(btnCancel);

btnLogin.addActionListener(this);

btnCancel.addActionListener(this); }

public void actionPerformed(ActionEvent e)

{ if(e.getSource()==btnCancel)

{ System.exit(0); }

if(e.getSource()==btnLogin)

{ try

{ Connection con;

con=DriverManager.getConnection("jdbc:Odbc:imsdsn");

ResultSet res;

Statement st=con.createStatement(); Int Flg=0;

res=st.executeQuery("select \*from logintbl where username='"+txtUsername.getText()+"'And password=’"+passwordField.getText()+”’");

while(res.next())

{ Flg=1; }

If(Flg==1)

{

JOptionPane.showMessageDialog(null,"Welcome To Society Management System");

JMenuD obj=new JMenuD(); }

else

JOptionPane.showMessageDialog(null,"Wrong UserID And Password"); }

catch(Exception X)

{ System.out.println("Exception :"+X); } } }

public static void main(String argv[])throws IOException

{ login obj=new login(); } }

MDI FORM (MEnud)

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import java.sql.\*;

public class JMenud extends JFrame implements ActionListener

{

int flag;

JMenuBar mb=new JMenuBar();

JMenu mnumaster=new JMenu("Master");

JMenu mnureport=new JMenu("Report");

JMenu mnutransaction=new JMenu("Transaction");

JMenu mnuexit=new JMenu("Logout");

JMenuItem mnuflat=new JMenuItem("Flat");

JMenuItem mnuduplex=new JMenuItem("Duplex");

JMenuItem mnuowner=new JMenuItem("Owner");

JMenuItem mnustaff=new JMenuItem("Staff");

JMenuItem mnuex=new JMenuItem("Ok");

JMenuItem mnubooking=new JMenuItem("Booking");

JMenuItem mnucancellation=new JMenuItem("Cancellation");

JMenuItem mnupayment=new JMenuItem("Payment");

JMenuItem mnuavailability=new JMenuItem("Available flats/duplex");

JMenuItem mnubbook=new JMenuItem("Total Booking");

JLabel lbltitle=new JLabel("<html><u>SOCIETY</u></html>");

JLabel lbltitle1=new JLabel("<html><u>MANAGEMENT</u></html>");

JLabel lbltitle2=new JLabel("<html><u>SYSTEM</u></html>");

public JMenud()

{

setLayout(null);

Dimension Screen=Toolkit.getDefaultToolkit().getScreenSize();

setSize(Screen.width,Screen.height);

setVisible(true);

setTitle("Society Management");

lbltitle.setForeground(Color.blue);

lbltitle2.setFont(new Font("Copperplate Gothic Bold",Font.BOLD,35));

lbltitle2.setForeground(Color.blue);

lbltitle.setBounds(580,100,400,40);

mnumaster.add(mnuflat);

mnumaster.add(mnuduplex);

mnumaster.add(mnuowner);

mnumaster.add(mnustaff);

mnutransaction.add(mnubooking);

mnureport.add(mnuavailability);

mnureport.add(mnubbook);

mnuexit.add(mnuex);

add(lbltitle);

add(lbltitle1);

add(lbltitle2);

mnuflat.addActionListener(this);

mnuduplex.addActionListener(this);

mnupayment.addActionListener(this);

mnuavailability.addActionListener(this);

mnubbook.addActionListener(this);

mnuex.addActionListener(this);

this.setJMenuBar(mb);

repaint();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==mnuflat)

{

flat\_info obj=new flat\_info();

}

if(e.getSource()==mnuduplex)

{

duplex\_info obj=new duplex\_info();

}

if(e.getSource()==mnustaff)

{

staff obj=new staff();

}

if(e.getSource()==mnuowner)

{

owner\_info obj=new owner\_info();

}

if(e.getSource()==mnucancellation)

{

cancellation obj=new cancellation();

}

if(e.getSource()==mnuavailability)

{

availability obj=new availability();

}

if(e.getSource()==mnuex)

{

System.exit(0);

}

}

public static void main(String args[])

{

JMenud obj=new JMenud();

}

}

Master Form

Flat form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.awt.font.\*;

public class flat\_info extends Frame implements ActionListener

{

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bsave=new JButton("Save",new ImageIcon("images/save.gif"));

JButton bsearch=new JButton("Search",new ImageIcon("images/search.png"));

JButton bedit=new JButton("Edit",new ImageIcon("images/edit.png"));

JButton bcancel=new JButton("Cancel",new ImageIcon("images/exit.png"));

JTextField txflatno=new JTextField(10);

JTextField txbuildname=new JTextField(10);

JTextField txaddress=new JTextField(10);

JTextField txcharge=new JTextField(10);

String[] pn={"Astha Township","Madhusudan Vihar","Ashok Vatika","Green City","Devnagri Palace"};

JComboBox projectname=new JComboBox(pn);

JLabel lbl=new JLabel("Flat Information");

JLabel lblflatno=new JLabel("Flat No: ");

JLabel lblbuildname=new JLabel("Building Name: ");

JLabel lbladdress=new JLabel("Address: ");

JLabel lblbuiltuparea=new JLabel("Builtup Area: ");

JLabel lblsuperbuiltuparea=new JLabel("Superbuiltup Area: ");

JLabel lblcarpetarea=new JLabel("Carpet Area: ");

JLabel lblfloor=new JLabel("Floor: ");

JLabel lblprojectname=new JLabel("Project Name: ");

JLabel lblfstatus=new JLabel("Status: ");

JLabel lblcharge=new JLabel("Charge(per Sq.ft):");

JLabel lbl1=new JLabel("Empty");

public flat\_info()

{

setLayout(null);

setSize(2100,1500);

setTitle("FLAT INFO:::-");

setVisible(true);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,50));

lbl.setForeground(Color.red);

lbl.setBounds(400,80,500,80);

lblflatno.setBackground(new Color(250,150,80));

lblflatno.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblflatno.setBounds(300,200,200,40);

txflatno.setBounds(600,200,300,40);

lblbuildname.setBackground(new Color(250,150,80));

lblbuildname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbuildname.setBounds(300,260,300,40);

txbuildname.setBounds(600,260,300,40);

lbladdress.setBackground(new Color(250,150,80));

lbladdress.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbladdress.setBounds(300,320,300,40);

txaddress.setBounds(600,320,300,40);

lblbuiltuparea.setBackground(new Color(250,150,80));

lblbuiltuparea.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbuiltuparea.setBounds(300,380,300,40);

txbuiltuparea.setBounds(600,380,300,40);

lblcarpetarea.setBackground(new Color(250,150,80));

lblcarpetarea.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcarpetarea.setBounds(300,440,250,40);

txcarpetarea.setBounds(600,440,300,40);

lbl1.setBackground(new Color(250,200,100));

lbl1.setFont(new Font("Comic Sans MS",Font.PLAIN,20));

lbl1.setBounds(600,680,250,30);

bnew.setBounds(300,800,150,40);

bsave.setBounds(450,800,150,40);

bsearch.setBounds(600,800,150,40);

bedit.setBounds(750,800,150,40);

bcancel.setBounds(900,800,150,40);

add(lbl);

add(lblflatno);

add(lblbuildname);

add(lbladdress);

add(lblbuiltuparea);

add(lblcarpetarea);

add(lblsuperbuiltuparea);

add(lblfloor);

add(lblprojectname);

add(lblcharge);

add(lbl1);

add(txflatno);

add(txbuildname);

add(txaddress);

add(txbuiltuparea);

add(txcarpetarea);

add(txsuperbuiltuparea);

add(txfloor);

add(projectname);

add(txcharge);

add(lblfstatus);

add(bnew);

add(bsave);

add(bsearch);

add(bedit);

add(bcancel);

bnew.addActionListener(this);

bsave.addActionListener(this);

bsearch.addActionListener(this);

bedit.addActionListener(this);

bcancel.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

txflatno.setText("");

txaddress.setText("");

txbuildname.setText("");

txbuiltuparea.setText("");

txcarpetarea.setText("");

txsuperbuiltuparea.setText("");

txfloor.setText("");

lbl1.setText("Empty");

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl"); while(res.next())

{

int x=Integer.parseInt(res.getString(3));

txflatno.setText("F\_"+String.format("%04d",x));

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bsave)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str1=projectname.getSelectedItem().toString();

String str="insert into flat\_info(flat\_no,build\_name,address,builtuparea,carpetarea,superbuiltuparea,floor,projectname,fstatus)values(?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

str="update keytbl set flat\_no=flat\_no+1";

ps=con.prepareStatement(str);

ps.executeUpdate();

txflatno.setText("");

txaddress.setText("");

txbuildname.setText("");

txbuiltuparea.setText("");

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bsearch)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String id=JOptionPane.showInputDialog(null,"Enter Flat ID:");

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bedit)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

String pn=projectname.getSelectedItem().toString();

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("Select \* from flat\_info");

String str="update flat\_info set flat\_no='"+txflatno.getText()+"',build\_name='"+txbuildname.getText()+"',address='"+txaddress.getText()+"',builtuparea='"+txbuiltuparea.getText()+"',carpetarea='"+txcarpetarea.getText()+"',superbuiltuparea='"+txsuperbuiltuparea.getText()+"',floor='"+txfloor.getText()+"',projectname='"+pn+"',fstatus='"+lblfstatus.getText()+"',charge='"+txcharge.getText()+"' where flat\_no='"+txflatno.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Updated successfully");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bcancel)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

flat\_info obj=new flat\_info();

}

}

DUPlex form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.awt.font.\*;

public class duplex\_info extends Frame implements ActionListener

{

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bsave=new JButton("Save",new ImageIcon("images/save.gif"));

JButton bsearch=new JButton("Search",new ImageIcon("images/search.png"));

JButton bedit=new JButton("Edit",new ImageIcon("images/edit.png"));

JButton bcancel=new JButton("Cancel",new ImageIcon("images/exit.png"));

JTextField txduplexno=new JTextField(10);

JTextField txduplexname=new JTextField(10);

JTextField txaddress=new JTextField(10);

JTextField txbuiltuparea=new JTextField(10);

JTextField txsuperbuiltuparea=new JTextField(20);

JTextField txcarpetarea=new JTextField(10);

JTextField txcharge=new JTextField(10);

JTextField txprojectname=new JTextField(10);

JLabel lbl=new JLabel("Duplex Information");

JLabel lblprojectname=new JLabel("Project Name: ");

JLabel lblfstatus=new JLabel("Status: ");

JLabel lbl1=new JLabel("Empty");

public duplex\_info()

{

setLayout(null);

setSize(2100,1500);

setTitle("DUPLEX INFO:::-");

setVisible(true);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,50));

lbl.setForeground(Color.red);

lbl.setBounds(400,80,500,80);

lblprojectname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblprojectname.setBounds(300,620,350,40);

txprojectname.setBounds(600,620,300,40);

lblfstatus.setBackground(new Color(250,150,80));

lblfstatus.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblfstatus.setBounds(300,680,300,40);

lbl1.setBackground(new Color(250,200,100));

lbl1.setFont(new Font("Comic Sans MS",Font.PLAIN,20));

lbl1.setBounds(600,680,250,30);

bnew.setBounds(300,800,100,40);

bsave.setBounds(400,800,100,40);

bsearch.setBounds(500,800,100,40);

bedit.setBounds(600,800,100,40);

bcancel.setBounds(700,800,100,40);

add(lbl);

add(lblduplexno);

add(lblduplexname);

add(lbladdress);

add(lblbuiltuparea);

add(txcharge);

add(txprojectname);

add(lblfstatus);

add(bnew);

add(bsave);

add(bsearch);

add(bedit);

add(bcancel);

bnew.addActionListener(this);

bsave.addActionListener(this);

bsearch.addActionListener(this);

bedit.addActionListener(this);

bcancel.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

txduplexno.setText("");

txaddress.setText("");

txduplexname.setText("");

txbuiltuparea.setText("");

txcarpetarea.setText("");

txsuperbuiltuparea.setText("");

txcharge.setText("");

txprojectname.setText("");

lbl1.setText("Empty");

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl");

while(res.next())

{

int x=Integer.parseInt(res.getString(4));

txduplexno.setText("D\_"+String.format("%04d",x));

}

}

if(e.getSource()==bsave)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str="insert into duplex\_info(duplex\_no,duplex\_name,address,builtuparea,carpetarea,superbuiltuparea,projectname,fstatus,charge)values(?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txduplexno.getText());

ps.setString(2,txduplexname.getText());

ps.setString(3,txaddress.getText());

ps.setString(4,txbuiltuparea.getText());

ps.setString(5,txcarpetarea.getText());

ps.setString(6,txsuperbuiltuparea.getText());

ps.setString(7,txcharge.getText());

ps.setString(8,txprojectname.getText());

ps.setString(9,lbl1.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved Successfully");

ResultSet res=st.executeQuery("update keytbl set duplex\_no=duplex\_no+1");

txcharge.setText("");

txprojectname.setText("");

lbl1.setText("Empty");

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bsearch)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

JOptionPane.showInputDialog(null,"Enter Duplex ID:");

res=st.executeQuery("Select \* from duplex\_info where duplex\_no='"+txduplexno.getText()+"'");

while(res.next())

{

txduplexno.setText(""+res.getString(1));

txduplexname.setText(""+res.getString(2));

txaddress.setText(""+res.getString(3));

txbuiltuparea.setText(""+res.getString(4));

txcarpetarea.setText(""+res.getString(5));

txsuperbuiltuparea.setText(""+res.getString(6));

txprojectname.setText(""+res.getString(7));

lbl1.setText(""+res.getString(8));

txcharge.setText(""+res.getString(9));

flg=1;

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No such record");

}

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bedit)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("Select \* from duplex\_info");

String str="update duplex\_info set duplex\_no='"+txduplexno.getText()+"',duplex\_name='"+txduplexname.getText()+"',address='"+txaddress.getText()+"',builtuparea='"+txbuiltuparea.getText()+"',carpetarea='"+txcarpetarea.getText()+"',superbuiltuparea='"+txsuperbuiltuparea.getText()+"',projectname='"+txprojectname.getText()+"',charge='"+txcharge.getText()+"' where duplex\_no='"+txduplexno.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Updated successfully");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bcancel)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

duplex\_info obj=new duplex\_info();

}

}

OWNER form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

public class owner\_info extends Frame implements ActionListener

{

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bsave=new JButton("Save",new ImageIcon("images/save.gif"));

JButton bsearch=new JButton("Search",new ImageIcon("images/search.png"));

JButton bedit=new JButton("Edit",new ImageIcon("images/edit.png"));

JButton bcancel=new JButton("Cancel",new ImageIcon("images/exit.png"));

JTextField txownerid=new JTextField(10);

JTextField txflatno=new JTextField(10);

JTextField txname=new JTextField(10);

JTextField txdob=new JTextField(10);

JTextField txaddress=new JTextField(10);

JTextField txcontactno=new JTextField(10);

JTextField txemail=new JTextField(10);

JTextField txdop=new JTextField(10);

JLabel lblownerid=new JLabel("Owner ID: ");

JLabel lbl=new JLabel("Owner Information ");

JLabel lblflatno=new JLabel("Flat No(s): ");

JLabel lblname=new JLabel("Owner name ");

JLabel lblgender=new JLabel("Gender: ");

JLabel lbldob=new JLabel("D.O.B: ");

JLabel lbladdress=new JLabel("Permanent Address: ");

JLabel lblcontactno=new JLabel("Contact No: ");

JLabel lblemail=new JLabel("Email: ");

JLabel lbldop=new JLabel("Date of purchase: ");

ButtonGroup cg=new ButtonGroup();

JRadioButton b3=new JRadioButton("Male");

JRadioButton b4=new JRadioButton("Female");

public owner\_info()

{

setLayout(null);

setSize(2100,1200);

setTitle("OWNER INFO:::-");

setVisible(true);

//setBackground(new Color(200,150,80));

lblownerid.setBackground(new Color(250,150,80));

lblownerid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblflatno.setBackground(new Color(250,150,80));

lblflatno.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblname.setBackground(new Color(250,150,80));

lblname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblgender.setBackground(new Color(250,150,80));

lblgender.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbldob.setBackground(new Color(250,150,80));

lbldob.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbladdress.setBackground(new Color(250,150,80));

lbladdress.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcontactno.setBackground(new Color(250,150,80));

lblcontactno.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblemail.setBackground(new Color(250,150,80));

lblemail.setFont(new Font("Comic Sans MS",Font.BOLD,30));

b3.setFont(new Font("Comic Sans MS",Font.PLAIN,20));

b4.setFont(new Font("Comic Sans MS",Font.PLAIN,20));

lbldop.setBackground(new Color(250,150,80));

lbldop.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbl.setBackground(Color.red);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,60));

lbl.setBounds(500,60,600,80);

lbl.setForeground(Color.red);

lblownerid.setBounds(400,200,200,40);

txownerid.setBounds(700,200,400,40);

lblflatno.setBounds(400,260,200,40);

txflatno.setBounds(700,260,400,40);

lblname.setBounds(400,320,300,40);

txname.setBounds(700,320,400,40);

lblgender.setBounds(400,380,200,40);

b3.setBounds(700,380,140,40);

b4.setBounds(900,380,140,40);

lbldob.setBounds(400,440,200,40);

txdob.setBounds(700,440,400,40);

lbladdress.setBounds(400,500,400,40);

txaddress.setBounds(700,500,400,40);

lblcontactno.setBounds(400,560,200,40);

txcontactno.setBounds(700,560,400,40);

lblemail.setBounds(400,620,200,40);

txemail.setBounds(700,620,400,40);

lbldop.setBounds(400,680,300,40);

txdop.setBounds(700,680,400,40);

bnew.setBounds(500,800,100,40);

bsave.setBounds(600,800,100,40);

bsearch.setBounds(700,800,100,40);

bedit.setBounds(800,800,100,40);

bcancel.setBounds(900,800,100,40);

add(lbl);

add(lblflatno);

add(lblownerid);

add(lblname);

add(lblgender);

add(lbldob);

add(lbladdress);

add(lblcontactno);

add(lblemail);

add(lbldop);

add(txflatno);

add(txownerid);

add(txname);

add(txdob);

add(txaddress);

add(txcontactno);

add(txemail);

add(txdop);

add(bnew);

add(bsave);

add(bsearch);

add(bedit);

add(bcancel);

add(b3);

add(b4);

cg.add(b3);

cg.add(b4);

bnew.addActionListener(this);

bsave.addActionListener(this);

bsearch.addActionListener(this);

bedit.addActionListener(this);

bcancel.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

repaint();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

txflatno.setText("");

txemail.setText("");

txdop.setText("");

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl");

while(res.next())

{

int x=Integer.parseInt(res.getString(1));

txownerid.setText("OWN\_"+String.format("%04d",x));

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bsave)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

////////////////////////////////////////

char d1=txdob.getText().charAt(0);

char d2=txdob.getText().charAt(1);

char d3=txdob.getText().charAt(2);

char d4=txdob.getText().charAt(3);

char d5=txdob.getText().charAt(4);

char d6=txdob.getText().charAt(5);

char d7=txdob.getText().charAt(6);

char d8=txdob.getText().charAt(7);

char d9=txdob.getText().charAt(8);

char d10=txdob.getText().charAt(9);

if((d1>='0' && d1<='9')&&(d2>='0' && d2<='9')&&(d4>='0' && d4<='9')&&(d5>='0' && d5<='9')&&(d7>='0' && d7<='9')&&(d8>='0' && d8<='9')&&(d9>='0' && d9<='9')&&(d10>='0' && d10<='9')&&d3=='/' && d6=='/')

////////////////////////////////////////

{

String str="insert into owner\_info(owner\_id,flat\_no,oname,gender,dob,address,contact,email,dop)values(?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txownerid.getText());

ps.setString(2,txflatno.getText());

ps.setString(3,txname.getText());

if(b3.isSelected())

ps.setString(4,"male");

else

ps.setString(4,"female");

ps.setString(5,txdob.getText());

ps.setString(6,txaddress.getText());

ps.setString(7,txcontactno.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved Successfully");

ResultSet res=st.executeQuery("update keytbl set owner\_id=owner\_id+1");

}

else

{

JOptionPane.showMessageDialog(null,"Date Invalid!!!");

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bsearch)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from owner\_info where owner\_id='"+txownerid.getText()+"'");

while(res.next())

{

txflatno.setText(""+res.getString(2));

txname.setText(""+res.getString(3));

if(res.getString(4).equals("male"))

{

b3.setSelected(true);

}

else

{

b4.setSelected(true);

}

txdob.setText(""+res.getString(5));

txaddress.setText(""+res.getString(6));

txcontactno.setText(""+res.getString(7));

txemail.setText(""+res.getString(8));

txdop.setText(""+res.getString(9));

flg=1;

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No such record");

}

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==bedit)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

ResultSet res;

Statement st=con.createStatement();

res=st.executeQuery("Select \* from owner\_info");

String gen;

if(b3.isSelected())

gen="male";

else

gen="female";

String str="update owner\_info set flat\_no='"+txflatno.getText()+"',oname='"+txname.getText()+"',gender='"+gen+"',dob='"+txdob.getText()+"',address='"+txaddress.getText()+"',contact='"+txcontactno.getText()+"',email='"+txemail.getText()+"',dop='"+txdop.getText()+"' where owner\_id='"+txownerid.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Updated successfully");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bcancel)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

owner\_info obj=new owner\_info();

}

}

Staff form

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import javax.swing.\*;

import java.sql.\*;

public class staff extends Frame implements ActionListener

{

JLabel l1=new JLabel("STAFF ID:-");

JLabel l2=new JLabel("FIRST NAME:-");

JLabel l3=new JLabel("LAST NAME:-");

JLabel l4=new JLabel("AGE:-");

JLabel l5=new JLabel("ADDRESS:-");

JLabel l6=new JLabel("CITY:-");

JLabel l7=new JLabel("STATE:-");

JLabel l8=new JLabel("EMAIL-ID:-");

JLabel l9=new JLabel("PHONE:-");

JLabel l10=new JLabel("GENDER:-");

JLabel l11=new JLabel("DESIGNATION:-");

JLabel l12=new JLabel("POST OFFICE:-");

JLabel l13=new JLabel("PIN-CODE:-");

JTextField txstaff\_id=new JTextField (5);

JTextField txfirst\_name=new JTextField (15);

JTextField txlast\_name=new JTextField (15);

JTextField txage=new JTextField (3);

JTextField txaddress=new JTextField (30);

JTextField txcity=new JTextField (15);

JTextField txstate=new JTextField (15);

JTextField txe\_mail=new JTextField (25);

JTextField txphone\_no=new JTextField (10);

JTextField txdesignation=new JTextField (5);

JTextField txpost\_office=new JTextField (15);

JTextField txpin\_code=new JTextField (15);

ButtonGroup cg=new ButtonGroup();

JButton bsave=new JButton("SAVE");

JButton bnew=new JButton("NEW");

JButton bcancel=new JButton("CANCEL");

JRadioButton b3=new JRadioButton("MALE");

JRadioButton b4=new JRadioButton("FEMALE");

public staff()

{

setLayout(null);

setSize(1000,1000);

setTitle("STAFF DETAILS:--");

setVisible(true);

setBackground(new Color(150,190,80));

l1.setBackground(new Color(200,40,124));

l1.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l2.setBackground(new Color(200,40,124));

l2.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l3.setBackground(new Color(200,40,124));

l3.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l4.setBackground(new Color(200,40,124));

l4.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l5.setBackground(new Color(200,40,124));

l5.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l6.setBackground(new Color(200,40,124));

l6.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l7.setBackground(new Color(200,40,124));

l7.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l8.setBackground(new Color(200,40,124));

l8.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l9.setBackground(new Color(200,40,124));

l9.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l10.setBackground(new Color(200,40,124));

l10.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l11.setBackground(new Color(200,40,124));

l11.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l12.setBackground(new Color(200,40,124));

l12.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l13.setBackground(new Color(200,40,124));

l13.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

b3.setBackground(new Color(150,190,80));

b3.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

b4.setBackground(new Color(150,190,80));

b4.setFont(new Font("Monotype Corsiva",Font.BOLD,30));

l1.setBounds(100,40,250,35);

l2.setBounds(100,80,250,35);

l3.setBounds(100,120,250,35);

l4.setBounds(100,160,250,35);

l5.setBounds(100,200,250,35);

l6.setBounds(100,240,250,35);

l7.setBounds(100,280,250,35);

l8.setBounds(100,320,250,35);

l9.setBounds(100,360,250,35);

l10.setBounds(100,400,250,35);

l11.setBounds(100,440,250,35);

l12.setBounds(100,480,250,35);

l13.setBounds(100,520,250,35);

txstaff\_id.setBounds(400,40,250,35);

txfirst\_name.setBounds(400,80,250,35);

txlast\_name.setBounds(400,120,250,35);

txage.setBounds(400,160,250,35);

txaddress.setBounds(400,200,250,35);

txcity.setBounds(400,240,250,35);

txstate.setBounds(400,280,250,35);

txe\_mail.setBounds(400,320,250,35);

txphone\_no.setBounds(400,360,250,35);

txdesignation.setBounds(400,440,250,35);

txpost\_office.setBounds(400,480,250,35);

txpin\_code.setBounds(400,520,250,35);

bsave.setBounds(150,600,100,30);

bnew.setBounds(350,600,100,30);

bcancel.setBounds(550,600,100,30);

b3.setBounds(400,400,120,30);

b4.setBounds(580,400,170,30);

add(l1);

add(l2);

add(l3);

add(l4);

add(l5);

add(l6);

add(l7);

add(l8);

add(l9);

add(l10);

add(l11);

add(l12);

add(l13);

add(txstaff\_id);

add(txfirst\_name);

add(txlast\_name);

add(txage);

add(txaddress);

add(txcity);

add(txstate);

add(txe\_mail);

add(txphone\_no);

add(txdesignation);

add(txpost\_office);

add(txpin\_code);

add(bsave);

add(bnew);

add(bcancel);

cg.add(b3);

cg.add(b4);

add(b3);

add(b4);

bsave.addActionListener(this);

bnew.addActionListener(this);

bcancel.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

txstaff\_id.setText("");

txfirst\_name.setText("");

txlast\_name.setText("");

txage.setText("");

txaddress.setText("");

txcity.setText("");

txstate.setText("");

txe\_mail.setText("");

txphone\_no.setText("");

txdesignation.setText("");

txpost\_office.setText("");

txpin\_code.setText("");

}

if(e.getSource()==bsave)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str="insert into staff(staff\_id,first\_name,last\_name,age,address,city,state,e\_mail,phone\_no,gender,designation,post\_office,pin\_code)values(?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txstaff\_id.getText());

ps.setString(2,txfirst\_name.getText());

ps.setString(3,txlast\_name.getText());

ps.setString(4,txage.getText());

if(b3.isSelected())

ps.setString(10,"Male");

else

ps.setString(10,"Female");

ps.setString(11,txdesignation.getText());

ps.setString(12,txpost\_office.getText());

ps.setString(13,txpin\_code.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Saved successfully");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bcancel)

{

System.exit(0);

}

}

public static void main(String argv[])throws IOException

{

staff obj=new staff();

}

Transaction forms

Booking form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.Calendar;

public class booking extends Frame implements ActionListener

{

String[] plan={"flexi","CLP","Subvention"};

JComboBox planlist=new JComboBox(plan);

String[] pn={"Astha Township","Madhusudan Vihar","Ashok Vatika","Green City","Devnagri Palace"};

JComboBox projectname=new JComboBox(pn);

JLabel lblbookid=new JLabel("Booking ID:");

JLabel lblbookdate=new JLabel("Booking Date:");

JLabel lblcustomerid=new JLabel("Customer ID:");

JLabel lblcustomername=new JLabel("Owner Name:");

JLabel lblflatid=new JLabel("Flat ID/Duplex ID:");

JLabel lbln=new JLabel("Name:");

JLabel lbladdress=new JLabel("Address:");

JLabel lblcarpetarea=new JLabel("Carpet Area:");

JLabel lblsuperbuiltuparea=new JLabel("Superbuiltup Area:");

JLabel lblpn=new JLabel("Project Name:");

JLabel lblcharge=new JLabel("Charge(per sq.ft):");

JLabel lbltotalamount=new JLabel("Total Amount:");

JLabel lbldiscount=new JLabel("Enter Discount:");

JLabel lbldprice=new JLabel("Price(After discount):");

JLabel lbldownpayment=new JLabel("Downpayment:");

JLabel lblnextpayment=new JLabel("Due Date:");

JLabel lblamt=new JLabel("Amount remaining:");

JLabel lblinterval=new JLabel("Intervals:");

JLabel lblbankname=new JLabel("Funded by:");

JLabel lblloanno=new JLabel("Loan No:");

JLabel lblctype=new JLabel("Customer Type:");

JButton brefresh=new JButton("Refresh");

JLabel lbl=new JLabel("Booking Details");

JLabel lblplan=new JLabel("Plan:");

JTextField txbookid=new JTextField(10);

JTextField txbookdate=new JTextField(10);

JTextField txcustomerid=new JTextField(10);

JTextField txcustomername=new JTextField(10);

JTextField txflatid=new JTextField(10);

JTextField txcharge=new JTextField(10);

JTextField txtotalamount=new JTextField(10);

JTextField txdiscount=new JTextField(10);

JTextField txdprice=new JTextField(10);

JTextField txdownpayment=new JTextField(10);

JTextField txnextpayment=new JTextField(10);

JTextField txamt=new JTextField(10);

JTextField txinterval =new JTextField(10);

JTextField txbankname=new JTextField(10);

JTextField txloanno=new JTextField(10);

JTextField txn=new JTextField(10);

JTextField txaddress=new JTextField(10);

JTextField txsuperbuiltuparea=new JTextField(10);

JTextField txcarpetarea=new JTextField(10);

ButtonGroup cg=new ButtonGroup();

JRadioButton b1=new JRadioButton("New");

JRadioButton b2=new JRadioButton("Existing");

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bbook=new JButton("Book",new ImageIcon("images/users.png"));

JButton bcancel=new JButton("Cancel",new ImageIcon("images/exit.png"));

JButton btable=new JButton("Table",new ImageIcon("images/table.png"));

public booking()

{

setLayout(null);

setVisible(true);

setTitle("Booking:::");

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

pack();

setSize(2100,1200);

lbl.setForeground(Color.red);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,60));

lbl.setBounds(720,50,500,80);

lblbookid.setBackground(new Color(250,150,80));

lblbookid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbookdate.setBackground(new Color(250,150,80));

lblbookdate.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcustomerid.setBackground(new Color(250,150,80));

lblcustomerid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcustomername.setBackground(new Color(250,150,80));

lblcustomername.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblflatid.setBackground(new Color(250,150,80));

lblflatid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblpn.setBackground(new Color(250,150,80));

lblpn.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblplan.setBackground(new Color(250,150,80));

lblplan.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbankname.setBackground(new Color(250,150,80));

lblbankname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblloanno.setBackground(new Color(250,150,80));

lblloanno.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcharge.setBackground(new Color(250,150,80));

lblcharge.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblamt.setBackground(new Color(250,150,80));

lblamt.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldiscount.setBackground(new Color(250,150,80));

lbldiscount.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbltotalamount.setBackground(new Color(250,150,80));

lbltotalamount.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldprice.setBackground(new Color(250,150,80));

lbldprice.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldownpayment.setBackground(new Color(250,150,80));

lbldownpayment.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblnextpayment.setBackground(new Color(250,150,80));

lblnextpayment.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblamt.setBackground(new Color(250,150,80));

lblamt.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblinterval.setBackground(new Color(250,150,80));

lblinterval.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbln.setBackground(new Color(250,150,80));

lbln.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbladdress.setBackground(new Color(250,150,80));

lbladdress.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblcarpetarea.setBackground(new Color(250,150,80));

lblcarpetarea.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblsuperbuiltuparea.setBackground(new Color(250,150,80));

lblsuperbuiltuparea.setFont(new Font("Comic Sans MS",Font.BOLD,20));

planlist.setBackground(new Color(250,150,80));

planlist.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblctype.setBackground(new Color(250,150,80));

lblctype.setFont(new Font("Comic Sans MS",Font.BOLD,25));

b1.setBackground(new Color(255,255,255));

b2.setBackground(new Color(255,255,255));

b1.setFont(new Font("Comic Sans MS",Font.BOLD,20));

b2.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblctype.setBounds(400,150,200,40);

b1.setBounds(620,150,100,40);

b2.setBounds(730,150,150,40);

lblbookid.setBounds(400,200,200,40);

txbookid.setBounds(620,200,250,40);

lblbookdate.setBounds(1000,200,300,40);

txbookdate.setBounds(1300,200,250,40);

lblcustomerid.setBounds(380,270,200,40);

txcustomerid.setBounds(620,270,250,40);

lblcustomername.setBounds(1000,270,300,40);

txcustomername.setBounds(1300,270,250,40);

lblflatid.setBounds(300,340,300,40);

txflatid.setBounds(620,340,250,40);

lblpn.setBounds(1000,340,300,40);

projectname.setBounds(1300,340,250,40);

lbln.setBounds(500,400,200,40);

txn.setBounds(710,400,200,40);

lblcarpetarea.setBounds(500,450,200,40);

txcarpetarea.setBounds(710,450,200,40);

lblcharge.setBounds(500,500,200,40);

txcharge.setBounds(710,500,200,40);

lbltotalamount.setBounds(500,550,200,40);

txtotalamount.setBounds(710,550,200,40);

lbladdress.setBounds(980,400,200,40);

txaddress.setBounds(1200,400,200,40);

lblsuperbuiltuparea.setBounds(980,450,200,40);

txsuperbuiltuparea.setBounds(1200,450,200,40);

lbldiscount.setBounds(980,500,200,40);

txdiscount.setBounds(1200,500,200,40);

brefresh.setBounds(1410,500,100,40);

lbldprice.setBounds(980,550,300,40);

txdprice.setBounds(1200,550,200,40);

lblplan.setBounds(400,620,250,40);

planlist.setBounds(670,620,200,40);

lbldownpayment.setBounds(500,680,200,40);

txdownpayment.setBounds(710,680,200,40);

lblinterval.setBounds(980,680,200,40);

txinterval.setBounds(1200,680,200,40);

lblnextpayment.setBounds(500,730,200,40);

txnextpayment.setBounds(710,730,200,40);

lblamt.setBounds(980,730,200,40);

txamt.setBounds(1200,730,200,40);

lblbankname.setBounds(400,800,200,40);

txbankname.setBounds(620,800,200,40);

lblloanno.setBounds(1000,800,250,40);

txloanno.setBounds(1300,800,250,40);

bnew.setBounds(500,900,100,50);

bbook.setBounds(600,900,100,50);

bcancel.setBounds(700,900,100,50);

btable.setBounds(800,900,100,50);

add(lblbookid);

add(txbookid);

add(lblbookdate);

add(txbookdate);

add(lblcustomerid);

add(txcustomerid);

add(lblcustomername);

add(txcustomername);

add(lblflatid);

add(txflatid);

add(lbln);

add(txn);

add(lbladdress);

add(txaddress);

add(lblcarpetarea);

add(txcarpetarea);

add(lblsuperbuiltuparea);

add(txsuperbuiltuparea);

add(projectname);

add(lblpn);

add(lblcharge);

add(txcharge);

add(lbltotalamount);

add(txtotalamount);

add(lbldiscount);

add(txdiscount);

add(lbldprice);

add(txdprice);

add(lblplan);

add(planlist);

add(lbldownpayment);

add(txdownpayment);

add(lblnextpayment);

add(txnextpayment);

add(lblamt);

add(txamt);

add(lblinterval);

add(txinterval);

add(lblbankname);

add(txbankname);

add(lblloanno);

add(txloanno);

add(lblctype);

cg.add(b1);

cg.add(b2);

add(b1);

add(b2);

add(lbl);

add(bnew);

add(bbook);

add(bcancel);

add(brefresh);

add(btable);

bnew.addActionListener(this);

bbook.addActionListener(this);

bcancel.addActionListener(this);

brefresh.addActionListener(this);

btable.addActionListener(this);

b1.addActionListener(this);

b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

try

{

Calendar cal=Calendar.getInstance();

txbookdate.setText(""+cal.get(Calendar.DATE)+" / "+((cal.get(Calendar.MONTH))+1)+" / "+cal.get(Calendar.YEAR));

txbankname.setText("");

txloanno.setText("");

txdiscount.setText("");

txtotalamount.setText("");

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl");

while(res.next())

{

int x=Integer.parseInt(res.getString(5));

txbookid.setText("BOOK\_"+String.format("%04d",x));

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from booking where flat\_id='"+txflatid.getText()+"'");

while(res.next())

{

flg=1;

}

if(flg==1)

{

JOptionPane.showMessageDialog(null,"Already booked");

txcustomerid.setText("");

txcustomername.setText("");

txflatid.setText("");

txn.setText("");

txsuperbuiltuparea.setText("");

txcarpetarea.setText("");

txcharge.setText("");

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0,tflg=0;

ResultSet res;

ResultSet res1;

ResultSet res2;

Double totalcharge;

String str=txflatid.getText();

int k=str.length();

int i=0;

if(str.charAt(i)=='F')

{

res=st.executeQuery("Select \* from flat\_info where flat\_no='"+txflatid.getText()+"'");

while(res.next())

{

txn.setText(""+res.getString(2));

txaddress.setText(""+res.getString(3));

txcarpetarea.setText(""+res.getString(5));

txsuperbuiltuparea.setText(""+res.getString(6));

Double ar=Double.parseDouble(txsuperbuiltuparea.getText());

txcharge.setText(""+res.getString(10));

Double charge=Double.parseDouble(txcharge.getText());

totalcharge=charge\*ar;

txtotalamount.setText(""+totalcharge);

flg=1;

}

}

if(str.charAt(i)=='D')

{

tflg=0;

res=st.executeQuery("Select \* from duplex\_info where duplex\_no='"+txflatid.getText()+"'");

while(res.next())

{

txn.setText(""+res.getString(2));

txaddress.setText(""+res.getString(3));

txcarpetarea.setText(""+res.getString(6));

txsuperbuiltuparea.setText(""+res.getString(5));

Double ar=Double.parseDouble(txsuperbuiltuparea.getText());

txcharge.setText(""+res.getString(9));

Double charge=Double.parseDouble(txcharge.getText());

totalcharge=charge\*ar;

txtotalamount.setText(""+totalcharge);

flg=1;

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No such record");

}

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from owner\_info where owner\_id='"+txcustomerid.getText()+"'");

while(res.next())

{

txcustomername.setText(""+res.getString(3));

flg=1;

}

if(flg==0)

JOptionPane.showMessageDialog(null,"Wrong id");

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==brefresh)

{

try

{

Double downpayment,remamount;

Double discount=Double.parseDouble(txdiscount.getText());

Double totalcharge=Double.parseDouble(txtotalamount.getText());

Double discountcharge=(discount/100)\*totalcharge;

Double finalcharge=totalcharge-discountcharge;

downpayment=0.1\*finalcharge;

remamount=finalcharge-downpayment;

txdprice.setText(""+finalcharge);

if(planlist.getSelectedItem()=="flexi")

{

txdownpayment.setText(""+downpayment) ;

txamt.setText(""+remamount);

Calendar cal=Calendar.getInstance();

txnextpayment.setText(""+cal.get(Calendar.DATE)+" / "+((cal.get(Calendar.MONTH))+2)+" / "+cal.get(Calendar.YEAR));

}

if(planlist.getSelectedItem()=="subvention")

{

txdownpayment.setText("");

}

}

catch(Exception ex)

{

System.out.println("Exception:"+ex);

}

}

if(e.getSource()==b1)

{

owner\_info obj=new owner\_info();

}

if(e.getSource()==b2)

{

JOptionPane.showMessageDialog(null,"Please Provide your Owner ID");

}

if(e.getSource()==bbook)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str="insert into booking(book\_id,book\_date,customer\_id,flat\_id,plan,downpayment,remamount,duedate,interval,bank\_name,loan\_no,installment,discount,total\_amount)values(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txbookid.getText());

ps.setString(2,txbookdate.getText());

ps.setString(3,txcustomerid.getText());

ps.setString(4,txflatid.getText());

ps.setString(5,(String)planlist.getSelectedItem());

ps.setString(6,txdownpayment.getText());

ps.setString(7,txamt.getText());

ps.setString(8,txnextpayment.getText());

ps.setString(9,txinterval.getText());

ps.setString(10,txbankname.getText());

ps.setString(11,txloanno.getText());

ps.setString(12,txnextpayment.getText());

ps.setString(13,txdiscount.getText());

ps.setString(14,txtotalamount.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Booking Successfull!!!");

str="update keytbl set book\_id=book\_id+1";

ps=con.prepareStatement(str);

ps.executeUpdate();

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str1=txflatid.getText();

int k=str1.length();

int i=0;

if(str1.charAt(i)=='F')

{

String str="update flat\_info set fstatus='occupied'"+" where flat\_no='"+txflatid.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

if(str1.charAt(0)=='D')

{

String str="update duplex\_info set fstatus='occupied'"+" where duplex\_no='"+txflatid.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

try

{

txbookid.setText("");

txbookdate.setText("");

txcustomerid.setText("");

txcustomername.setText("");

txflatid.setText("");

txn.setText("");

txaddress.setText("");

txcarpetarea.setText("");

txsuperbuiltuparea.setText("");

txcharge.setText("");

txbankname.setText("");

txloanno.setText("");

txinterval.setText("");

txdownpayment.setText("");

txdprice.setText("");

txnextpayment.setText("");

txdiscount.setText("");

txtotalamount.setText("");

txamt.setText("");

}

catch(Exception ex)

{

System.out.println("Exception:::"+ex);

}

/\*try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str1=txflatid.getText();

int k=str1.length();

int i=0;

//System.out.println("1111111");

if(str1.charAt(i)=='D')

{

System.out.println("22222222");

String str="update duplex\_info set fstatus='occupied'"+" where duplex\_no='"+txflatid.getText()+"'";

System.out.println(str);

PreparedStatement ps=con.prepareStatement(str);

System.out.println("333333333");

ps.executeUpdate();

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}\*/

}

if(e.getSource()==btable)

{

availability obj=new availability();

}

if(e.getSource()==bcancel)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

booking obj=new booking();

}

}

Payment form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.Calendar;

public class booking extends Frame implements ActionListener

{

String[] plan={"flexi","CLP","Subvention"};

JComboBox planlist=new JComboBox(plan);

String[] pn={"Astha Township","Madhusudan Vihar","Ashok Vatika","Green City","Devnagri Palace"};

JComboBox projectname=new JComboBox(pn);

JLabel lblbookid=new JLabel("Booking ID:");

JLabel lblbookdate=new JLabel("Booking Date:");

JLabel lblcustomerid=new JLabel("Customer ID:");

JLabel lblcustomername=new JLabel("Owner Name:");

JLabel lblflatid=new JLabel("Flat ID/Duplex ID:");

JLabel lbln=new JLabel("Name:");

JLabel lbladdress=new JLabel("Address:");

JLabel lblcarpetarea=new JLabel("Carpet Area:");

JLabel lblsuperbuiltuparea=new JLabel("Superbuiltup Area:");

JLabel lblpn=new JLabel("Project Name:");

JLabel lblcharge=new JLabel("Charge(per sq.ft):");

JLabel lbltotalamount=new JLabel("Total Amount:");

JLabel lbldiscount=new JLabel("Enter Discount:");

JLabel lbldprice=new JLabel("Price(After discount):");

JLabel lbldownpayment=new JLabel("Downpayment:");

JLabel lblnextpayment=new JLabel("Due Date:");

JLabel lblamt=new JLabel("Amount remaining:");

JLabel lblinterval=new JLabel("Intervals:");

JLabel lblbankname=new JLabel("Funded by:");

JLabel lblloanno=new JLabel("Loan No:");

JLabel lblctype=new JLabel("Customer Type:");

JButton brefresh=new JButton("Refresh");

JLabel lbl=new JLabel("Booking Details");

JLabel lblplan=new JLabel("Plan:");

JTextField txdownpayment=new JTextField(10);

JTextField txnextpayment=new JTextField(10);

JTextField txamt=new JTextField(10);

JTextField txinterval =new JTextField(10);

JTextField txbankname=new JTextField(10);

JTextField txloanno=new JTextField(10);

JTextField txn=new JTextField(10);

JTextField txaddress=new JTextField(10);

JTextField txsuperbuiltuparea=new JTextField(10);

JTextField txcarpetarea=new JTextField(10);

ButtonGroup cg=new ButtonGroup();

JRadioButton b1=new JRadioButton("New");

JRadioButton b2=new JRadioButton("Existing");

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bbook=new JButton("Book",new ImageIcon("images/users.png"));

JButton bcancel=new JButton("Cancel",new ImageIcon("images/exit.png"));

JButton btable=new JButton("Table",new ImageIcon("images/table.png"));

public booking()

{

setLayout(null);

setVisible(true);

setTitle("Booking:::");

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

pack();

setSize(2100,1200);

lbl.setForeground(Color.red);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,60));

lbl.setBounds(720,50,500,80);

lblbookid.setBackground(new Color(250,150,80));

lblbookid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbookdate.setBackground(new Color(250,150,80));

lblbookdate.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcustomerid.setBackground(new Color(250,150,80));

lblcustomerid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcustomername.setBackground(new Color(250,150,80));

lblcustomername.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblflatid.setBackground(new Color(250,150,80));

lblflatid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblpn.setBackground(new Color(250,150,80));

lblpn.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblplan.setBackground(new Color(250,150,80));

lblplan.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbankname.setBackground(new Color(250,150,80));

lblbankname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblloanno.setBackground(new Color(250,150,80));

lblloanno.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblcharge.setBackground(new Color(250,150,80));

lblcharge.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblamt.setBackground(new Color(250,150,80));

lblamt.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldiscount.setBackground(new Color(250,150,80));

lbldiscount.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbltotalamount.setBackground(new Color(250,150,80));

lbltotalamount.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldprice.setBackground(new Color(250,150,80));

lbldprice.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldownpayment.setBackground(new Color(250,150,80));

lbldownpayment.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblnextpayment.setBackground(new Color(250,150,80));

lblnextpayment.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblamt.setBackground(new Color(250,150,80));

lblamt.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblinterval.setBackground(new Color(250,150,80));

lblinterval.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbln.setBackground(new Color(250,150,80));

lbln.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbladdress.setBackground(new Color(250,150,80));

lbladdress.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblcarpetarea.setBackground(new Color(250,150,80));

lblcarpetarea.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblsuperbuiltuparea.setBackground(new Color(250,150,80));

lblsuperbuiltuparea.setFont(new Font("Comic Sans MS",Font.BOLD,20));

planlist.setBackground(new Color(250,150,80));

planlist.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblctype.setBackground(new Color(250,150,80));

lblctype.setFont(new Font("Comic Sans MS",Font.BOLD,25));

b1.setBackground(new Color(255,255,255));

b2.setBackground(new Color(255,255,255));

b1.setFont(new Font("Comic Sans MS",Font.BOLD,20));

b2.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lblcustomername.setBounds(1000,270,300,40);

txcustomername.setBounds(1300,270,250,40);

lblflatid.setBounds(300,340,300,40);

txflatid.setBounds(620,340,250,40);

lblpn.setBounds(1000,340,300,40);

projectname.setBounds(1300,340,250,40);

lbln.setBounds(500,400,200,40);

txn.setBounds(710,400,200,40);

lblcarpetarea.setBounds(500,450,200,40);

txcarpetarea.setBounds(710,450,200,40);

lblcharge.setBounds(500,500,200,40);

txcharge.setBounds(710,500,200,40);

lbltotalamount.setBounds(500,550,200,40);

txtotalamount.setBounds(710,550,200,40);

lbladdress.setBounds(980,400,200,40);

txaddress.setBounds(1200,400,200,40);

lblsuperbuiltuparea.setBounds(980,450,200,40);

txsuperbuiltuparea.setBounds(1200,450,200,40);

lbldiscount.setBounds(980,500,200,40);

txdiscount.setBounds(1200,500,200,40);

brefresh.setBounds(1410,500,100,40);

lbldprice.setBounds(980,550,300,40);

txdprice.setBounds(1200,550,200,40);

lblplan.setBounds(400,620,250,40);

planlist.setBounds(670,620,200,40);

lbldownpayment.setBounds(500,680,200,40);

txdownpayment.setBounds(710,680,200,40);

lblinterval.setBounds(980,680,200,40);

txinterval.setBounds(1200,680,200,40);

lblnextpayment.setBounds(500,730,200,40);

txnextpayment.setBounds(710,730,200,40);

lblamt.setBounds(980,730,200,40);

txamt.setBounds(1200,730,200,40);

lblbankname.setBounds(400,800,200,40);

txbankname.setBounds(620,800,200,40);

lblloanno.setBounds(1000,800,250,40);

txloanno.setBounds(1300,800,250,40);

bnew.setBounds(500,900,100,50);

bbook.setBounds(600,900,100,50);

bcancel.setBounds(700,900,100,50);

btable.setBounds(800,900,100,50);

add(lblbookid);

add(txbookid);

add(lblbookdate);

add(projectname);

add(lblpn);

add(lblcharge);

add(txcharge);

add(txinterval);

add(lblbankname);

add(txbankname);

add(lblloanno);

add(txloanno);

add(lblctype);

cg.add(b1);

cg.add(b2);

add(b1);

add(b2);

add(lbl);

add(bnew);

add(bbook);

add(bcancel);

add(brefresh);

add(btable);

bnew.addActionListener(this);

bbook.addActionListener(this);

bcancel.addActionListener(this);

brefresh.addActionListener(this);

btable.addActionListener(this);

b1.addActionListener(this);

b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

try

{

Calendar cal=Calendar.getInstance();

txbookdate.setText(""+cal.get(Calendar.DATE)+" / "+((cal.get(Calendar.MONTH))+1)+" / "+cal.get(Calendar.YEAR));

txbankname.setText("");

txloanno.setText("");

txdiscount.setText("");

txtotalamount.setText("");

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("Select \* from keytbl");

while(res.next())

{

int x=Integer.parseInt(res.getString(5));

txbookid.setText("BOOK\_"+String.format("%04d",x));

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from booking where flat\_id='"+txflatid.getText()+"'");

while(res.next())

{

flg=1;

}

if(flg==1)

{

JOptionPane.showMessageDialog(null,"Already booked");

txcustomerid.setText("");

txcustomername.setText("");

txflatid.setText("");

txn.setText("");

txsuperbuiltuparea.setText("");

txcarpetarea.setText("");

txcharge.setText("");

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0,tflg=0;

ResultSet res;

ResultSet res1;

ResultSet res2;

Double totalcharge;

String str=txflatid.getText();

int k=str.length();

int i=0;

if(str.charAt(i)=='F')

{

res=st.executeQuery("Select \* from flat\_info where flat\_no='"+txflatid.getText()+"'");

while(res.next())

{

txn.setText(""+res.getString(2));

txaddress.setText(""+res.getString(3));

txcarpetarea.setText(""+res.getString(5));

txsuperbuiltuparea.setText(""+res.getString(6));

Double ar=Double.parseDouble(txsuperbuiltuparea.getText());

txcharge.setText(""+res.getString(10));

Double charge=Double.parseDouble(txcharge.getText());

totalcharge=charge\*ar;

txtotalamount.setText(""+totalcharge);

flg=1;

}

}

if(str.charAt(i)=='D')

{

tflg=0;

res=st.executeQuery("Select \* from duplex\_info where duplex\_no='"+txflatid.getText()+"'");

while(res.next())

{

txn.setText(""+res.getString(2));

txaddress.setText(""+res.getString(3));

txcarpetarea.setText(""+res.getString(6));

txsuperbuiltuparea.setText(""+res.getString(5));

Double ar=Double.parseDouble(txsuperbuiltuparea.getText());

txcharge.setText(""+res.getString(9));

Double charge=Double.parseDouble(txcharge.getText());

totalcharge=charge\*ar;

txtotalamount.setText(""+totalcharge);

flg=1;

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"No such record");

}

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from owner\_info where owner\_id='"+txcustomerid.getText()+"'");

while(res.next())

{

txcustomername.setText(""+res.getString(3));

flg=1;

}

if(flg==0)

JOptionPane.showMessageDialog(null,"Wrong id");

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

}

if(e.getSource()==brefresh)

{

try

{

Double downpayment,remamount;

Double discount=Double.parseDouble(txdiscount.getText());

Double totalcharge=Double.parseDouble(txtotalamount.getText());

Double discountcharge=(discount/100)\*totalcharge;

Double finalcharge=totalcharge-discountcharge;

downpayment=0.1\*finalcharge;

remamount=finalcharge-downpayment;

txdprice.setText(""+finalcharge);

if(planlist.getSelectedItem()=="flexi")

{

txdownpayment.setText(""+downpayment) ;

txamt.setText(""+remamount);

Calendar cal=Calendar.getInstance();

txnextpayment.setText(""+cal.get(Calendar.DATE)+" / "+((cal.get(Calendar.MONTH))+2)+" / "+cal.get(Calendar.YEAR));

}

if(planlist.getSelectedItem()=="subvention")

{

txdownpayment.setText("");

}

}

catch(Exception ex)

{

System.out.println("Exception:"+ex);

}

}

if(e.getSource()==b1)

{

owner\_info obj=new owner\_info();

}

if(e.getSource()==b2)

{

JOptionPane.showMessageDialog(null,"Please Provide your Owner ID");

}

if(e.getSource()==bbook)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str="insert into booking(book\_id,book\_date,customer\_id,flat\_id,plan,downpayment,remamount,duedate,interval,bank\_name,loan\_no,installment,discount,total\_amount)values(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txbookid.getText());

ps.setString(2,txbookdate.getText());

ps.setString(3,txcustomerid.getText());

ps.setString(4,txflatid.getText());

ps.setString(5,(String)planlist.getSelectedItem());

ps.setString(6,txdownpayment.getText());

ps.setString(7,txamt.getText());

ps.setString(8,txnextpayment.getText());

ps.setString(9,txinterval.getText());

ps.setString(10,txbankname.getText());

ps.setString(11,txloanno.getText());

ps.setString(12,txnextpayment.getText());

ps.setString(13,txdiscount.getText());

ps.setString(14,txtotalamount.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Booking Successfull!!!");

str="update keytbl set book\_id=book\_id+1";

ps=con.prepareStatement(str);

ps.executeUpdate();

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str1=txflatid.getText();

int k=str1.length();

int i=0;

if(str1.charAt(i)=='F')

{

String str="update flat\_info set fstatus='occupied'"+" where flat\_no='"+txflatid.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

if(str1.charAt(0)=='D')

{

String str="update duplex\_info set fstatus='occupied'"+" where duplex\_no='"+txflatid.getText()+"'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

try

{

txbookid.setText("");

txbookdate.setText("");

txcustomerid.setText("");

txcustomername.setText("");

txflatid.setText("");

txn.setText("");

txaddress.setText("");

txcarpetarea.setText("");

txsuperbuiltuparea.setText("");

txcharge.setText("");

txbankname.setText("");

txloanno.setText("");

txinterval.setText("");

txdownpayment.setText("");

txdprice.setText("");

txnextpayment.setText("");

txdiscount.setText("");

txtotalamount.setText("");

txamt.setText("");

}

catch(Exception ex)

{

System.out.println("Exception:::"+ex);

}

}

if(e.getSource()==btable)

{

availability obj=new availability();

}

if(e.getSource()==bcancel)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

booking obj=new booking();

}

}

Cancellation form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.awt.font.\*;

import java.util.Calendar;

public class cancellation extends Frame implements ActionListener

{

JButton bnew=new JButton("New",new ImageIcon("images/new.gif"));

JButton bsubmit=new JButton("Submit",new ImageIcon("images/restore.gif"));

JButton bexit=new JButton("Exit",new ImageIcon("images/exit.png"));

JTextField txbookid=new JTextField(10);

JTextField txdate=new JTextField(10);

JTextField txownerid=new JTextField(10);

JTextField txflatid=new JTextField(10);

JTextField txpaymentrcv=new JTextField(20);

JTextField txdueamount=new JTextField(10);

JTextField txdeductionmade=new JTextField(10);

JTextField txamtrefund=new JTextField(10);

JTextField txreason=new JTextField(10);

JLabel lbl=new JLabel("CANCELLATION");

JLabel lblbookid=new JLabel("Enter Booking ID: ");

JLabel lbldate=new JLabel("Cancellation Date: ");

JLabel lblownerid=new JLabel("Owner ID: ");

JLabel lblflatid=new JLabel("Flat/Duplex ID: ");

JLabel lblpaymentrcv=new JLabel("Payment received: ");

JLabel lbldueamount=new JLabel("Due Amount: ");

JLabel lbldeductionmade=new JLabel("Deduction made: ");

JLabel lblamtrefund=new JLabel("Amount refunded(if any): ");

JLabel lblreason=new JLabel("Reason for cancellation: ");

public cancellation()

{

setLayout(null);

setSize(2100,1500);

setTitle("FLAT INFO:::-");

setVisible(true);

lbl.setFont(new Font("Comic Sans MS",Font.BOLD,50));

lbl.setForeground(Color.red);

lbl.setBounds(500,80,500,80);

lblbookid.setBackground(new Color(250,150,80));

lblbookid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblbookid.setBounds(300,200,400,40);

txbookid.setBounds(700,200,300,40);

lbldate.setBackground(new Color(250,150,80));

lbldate.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbldate.setBounds(300,260,400,40);

txdate.setBounds(700,260,300,40);

lblownerid.setBackground(new Color(250,150,80));

lblownerid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblownerid.setBounds(300,320,400,40);

txownerid.setBounds(700,320,300,40);

lblflatid.setBackground(new Color(250,150,80));

lblflatid.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblflatid.setBounds(300,380,400,40);

txflatid.setBounds(700,380,300,40);

lblpaymentrcv.setBackground(new Color(250,150,80));

lblpaymentrcv.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblpaymentrcv.setBounds(300,440,400,40);

txpaymentrcv.setBounds(700,440,300,40);

lbldueamount.setBackground(new Color(250,150,80));

lbldueamount.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbldueamount.setBounds(300,500,400,40);

txdueamount.setBounds(700,500,300,40);

lbldeductionmade.setBackground(new Color(250,150,80));

lbldeductionmade.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lbldeductionmade.setBounds(300,560,400,40);

txdeductionmade.setBounds(700,560,300,40);

lblamtrefund.setBackground(new Color(250,150,80));

lblamtrefund.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblamtrefund.setBounds(300,620,400,40);

txamtrefund.setBounds(700,620,300,40);

lblreason.setBackground(new Color(250,150,80));

lblreason.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblreason.setBounds(300,680,400,40);

txreason.setBounds(700,680,300,40);

bnew.setBounds(400,800,150,40);

bsubmit.setBounds(550,800,150,40);

bexit.setBounds(700,800,150,40);

add(lbl);

add(lblbookid);

add(lbldate);

add(lblownerid);

add(lblflatid);

add(lblpaymentrcv);

add(lbldueamount);

add(lbldeductionmade);

add(lblamtrefund);

add(lblreason);

add(txbookid);

add(txdate);

add(txownerid);

add(txflatid);

add(txpaymentrcv);

add(txdueamount);

add(txdeductionmade);

add(txamtrefund);

add(txreason);

add(bnew);

add(bsubmit);

add(bexit);

bnew.addActionListener(this);

bsubmit.addActionListener(this);

bexit.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bnew)

{

try

{

Calendar cal=Calendar.getInstance();

txdate.setText(""+cal.get(Calendar.DATE)+" / "+((cal.get(Calendar.MONTH))+1)+" / "+cal.get(Calendar.YEAR));

txdeductionmade.setText("");

txamtrefund.setText("");

txreason.setText("");

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

int flg=0;

ResultSet res;

res=st.executeQuery("Select \* from booking where book\_id='"+txbookid.getText()+"'");

while(res.next())

{

flg=1;

txownerid.setText(""+res.getString(3));

txflatid.setText(""+res.getString(4));

txpaymentrcv.setText(""+res.getString(6));

txdueamount.setText(""+res.getString(7));

}

if(flg==0)

{

JOptionPane.showMessageDialog(null,"Booking ID not found");

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bsubmit)

{

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

String str="insert into cancellation(book\_id,book\_date,owner\_id,flat\_id,deduction,amtrefund,reason)values(?,?,?,?,?,?,?)";

PreparedStatement ps=con.prepareStatement(str);

ps.setString(1,txbookid.getText());

ps.setString(2,txdate.getText());

ps.setString(3,txownerid.getText());

ps.setString(4,txflatid.getText());

ps.setString(5,txdeductionmade.getText());

ps.setString(6,txamtrefund.getText());

ps.setString(7,txreason.getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Cancellation Successfull");

Connection con1;

con1=DriverManager.getConnection("jdbc:odbc:sms");

ResultSet res;

Statement st1=con.createStatement();

res=st1.executeQuery("Select \* from booking");

String s="Cancelled";

String str1="update booking set status='"+s+"'";

PreparedStatement ps1=con.prepareStatement(str1);

ps1.executeUpdate();

}

catch(Exception ex)

{

System.out.println("ERROR:"+ex);

}

try

{

txbookid.setText("");

txdate.setText("");

txownerid.setText("");

txflatid.setText("");

txpaymentrcv.setText("");

txdeductionmade.setText("");

txreason.setText("");

}

catch(Exception ex)

{

System.out.println("Exception:::"+ex);

}

try

{

Connection con;

con=DriverManager.getConnection("jdbc:odbc:sms");

ResultSet res;

Statement st=con.createStatement();

String id=txflatid.getText();

if(id.charAt(0)=='F')

{

String str="update flat\_info set fstatus='empty'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

else if(id.charAt(0)=='D')

{

String str="update duplex\_info set fstatus='empty'";

PreparedStatement ps=con.prepareStatement(str);

ps.executeUpdate();

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bexit)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

cancellation obj=new cancellation();

}

}

Report forms

Availability form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.Calendar;

import java.util.\*;

import javax.swing.table.DefaultTableModel;

public class availability extends Frame implements ActionListener

{

String[] projects={"Astha township","Madhusudan Vihar","Green city","Ashok Vatika","DevNagri Palace"};

JComboBox projectlist=new JComboBox(projects);

JTable table1 =new JTable();

JScrollPane sp;

JButton bsearch=new JButton("Search");

JButton bprint=new JButton("Print");

JButton bclose=new JButton("Close");

JLabel lblhouse=new JLabel("House Type:");

JLabel lblprojectname=new JLabel("Project Name:");

ButtonGroup fd=new ButtonGroup();

JRadioButton b1=new JRadioButton("Flat");

JRadioButton b2=new JRadioButton("Duplex");

String m,n,o,p,q;

JTable table2;

String[] col={"Flat no","Address","Floor","Carpet Area","Charge(per Sq.ft)"};

Object[] [] data={{m,n,o,p,q}};

DefaultTableModel model=new DefaultTableModel(col,0);

public availability()

{

setLayout(null);

setSize(1200,1500);

setVisible(true);

table2=new JTable(data,col);

sp=new JScrollPane(table2);

table2.setModel(model);

setBackground(new Color(234,255,229));

add(sp);

table2.setBounds(450,500,800,200);

sp.setBounds(450,500,800,200);

table2.setFillsViewportHeight(true);

add(bsearch);

add(bprint);

add(bclose);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

projectlist.setFont(new Font("Comic Sans MS",Font.PLAIN,20));

lblhouse.setBackground(new Color(250,150,80));

lblhouse.setFont(new Font("Comic Sans MS",Font.BOLD,30));

b1.setBackground(new Color(255,255,255));

b2.setBackground(new Color(255,255,255));

b1.setFont(new Font("Comic Sans MS",Font.PLAIN,30));

b2.setFont(new Font("Comic Sans MS",Font.PLAIN,30));

lblprojectname.setBackground(new Color(255,255,255));

lblprojectname.setFont(new Font("Comic Sans MS",Font.BOLD,50));

lblprojectname.setBackground(new Color(250,150,80));

lblprojectname.setFont(new Font("Comic Sans MS",Font.BOLD,30));

lblhouse.setBounds(400,200,200,60);

b1.setBounds(700,200,140,60);

b2.setBounds(900,200,140,60);

lblprojectname.setBounds(400,300,350,60);

projectlist.setBounds(700,300,300,50);

bsearch.setBounds(1050,300,100,50);

bprint.setBounds(1150,300,100,50);

bclose.setBounds(1250,300,100,50);

add(bclose);

add(lblprojectname);

add(projectlist);

add(lblhouse);

add(b1);

add(b2);

fd.add(b1);

fd.add(b2);

add(bsearch);

add(bprint);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

b1.addActionListener(this);

b2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bsearch)

{

model.setRowCount(0);

try

{

if(b1.isSelected())

{

Connection con;

ResultSet res;

String[] col={"Flat no","Address","Floor","Carpet Area","Charge(per Sq.ft)"};

DefaultTableModel model=new DefaultTableModel(col,0);

table2.setModel(model);

String pname=projectlist.getSelectedItem().toString();

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from flat\_info where projectname='"+pname+"'");

while(res.next())

{

Vector v=new Vector();

v.add(res.getString(1));

v.add(res.getString(3));

v.add(res.getString(7));

v.add(res.getString(5));

v.add(res.getString(10));

model.addRow(v);

}

}

if(b2.isSelected())

{

Connection con;

ResultSet res;

String[] col={"Duplex no","Address","Carpet Area","Superbuiltup Area","Charge(per Sq.ft)"};

DefaultTableModel model=new DefaultTableModel(col,0);

table2.setModel(model);

String pname=projectlist.getSelectedItem().toString();

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from duplex\_info where projectname='"+pname+"'");

while(res.next())

{

Vector v=new Vector();

v.add(res.getString(1));

v.add(res.getString(3));

v.add(res.getString(6));

v.add(res.getString(5));

v.add(res.getString(9));

model.addRow(v);

}

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bprint)

{

try

{

PrintWriter objpr=new PrintWriter("avail.txt");

objpr.println("Flat/Duplex No \t Address \t Area");

objpr.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

Connection con;

ResultSet res;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

if(b1.isSelected())

{

res=st.executeQuery("Select \* from flat\_info where projectname='"+projectlist.getSelectedItem()+"'");

while(res.next())

{

objpr.println(res.getString(1)+"\t \t"+res.getString(3)+"\t \t"+res.getString(6));

}

}

else

{

res=st.executeQuery("Select \* from flat\_info where projectname='"+projectlist.getSelectedItem()+"'");

while(res.next())

{

objpr.println(res.getString(1)+"\t \t"+res.getString(3)+"\t \t"+res.getString(5));

}

}

objpr.close();

Runtime robj=Runtime.getRuntime();

Process pr=robj.exec("notepad.exe d:\\new\\avail.txt");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bclose)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

availability obj=new availability();

}

}

Total booking form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.Calendar;

import java.util.\*;

import javax.swing.table.DefaultTableModel;

public class bbook extends Frame implements ActionListener

{

JTable table1 =new JTable();

JScrollPane sp;

JButton bsearch=new JButton("Search");

JButton bprint=new JButton("Print");

JButton bclose=new JButton("Close");

JLabel lbldate=new JLabel("Enter Date(Eg:01/01/2017):");

JTextField txdate=new JTextField(10);

String m,n,o;

JTable table2;

String[] col={"Booking ID","Customer ID","Flat/Duplex No"};

Object[] [] data={{m,n,o}};

DefaultTableModel model=new DefaultTableModel(col,0);

public bbook()

{

setLayout(null);

setSize(1200,1500);

setVisible(true);

table2=new JTable(data,col);

sp=new JScrollPane(table2);

table2.setModel(model);

setBackground(new Color(234,255,229));

lbldate.setBackground(new Color(250,150,80));

lbldate.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldate.setBounds(500,300,450,60);

txdate.setBounds(810,300,140,60);

table2.setBounds(500,400,800,200);

sp.setBounds(500,400,800,200);

table2.setFillsViewportHeight(true);

add(sp);

add(lbldate);

add(txdate);

add(bsearch);

add(bprint);

add(bclose);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

bsearch.setBounds(1050,300,100,50);

bprint.setBounds(1150,300,100,50);

bclose.setBounds(1250,300,100,50);

add(bclose);

add(bsearch);

add(bprint);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==bsearch)

{

int flg=0;

model.setRowCount(0);

try

{

char d1=txdate.getText().charAt(0);

char d2=txdate.getText().charAt(1);

char d3=txdate.getText().charAt(2);

char d4=txdate.getText().charAt(3);

char d5=txdate.getText().charAt(4);

char d6=txdate.getText().charAt(5);

char d7=txdate.getText().charAt(6);

char d8=txdate.getText().charAt(7);

char d9=txdate.getText().charAt(8);

char d10=txdate.getText().charAt(9);

if((d1>='0' && d1<='9')&&(d2>='0' && d2<='9')&&(d4>='0' && d4<='9')&&(d5>='0' && d5<='9')&&(d7>='0' && d7<='9')&&(d8>='0' && d8<='9')&&(d9>='0' && d9<='9')&&(d10>='0' && d10<='9')&&d3=='/' && d6=='/')

////////////////////////////////////////

{

Connection con;

ResultSet res;

String[] col={"Booking ID","Customer ID","Flat/Duplex No"};

DefaultTableModel model=new DefaultTableModel(col,0);

table2.setModel(model);

String bdate=txdate.getText();

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from booking where book\_date='"+bdate+"'");

while(res.next())

{

flg=1;

Vector v=new Vector();

v.add(res.getString(1));

v.add(res.getString(3));

v.add(res.getString(4));

model.addRow(v);

}

}

else

JOptionPane.showMessageDialog(null,"Invalid Date!!!!");

if(flg==0)

JOptionPane.showMessageDialog(null,"Date Not Found!!!!");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bprint)

{

try

{

PrintWriter objpr=new PrintWriter("bbook.txt");

objpr.println("Booking ID \t Customer ID \t Flat/Duplex No");

objpr.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

Connection con;

ResultSet res;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from booking where book\_date='"+txdate.getText()+"'");

while(res.next())

{

objpr.println(res.getString(1)+"\t"+res.getString(3)+"\t"+res.getString(4));

}

objpr.close();

Runtime robj=Runtime.getRuntime();

Process pr=robj.exec("notepad.exe d:\\new\\bbook.txt");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

if(e.getSource()==bclose)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

bbook obj=new bbook();

}

}

Total payment form

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.io.\*;

import javax.swing.\*;

import java.sql.\*;

import java.util.Calendar;

import java.util.\*;

import javax.swing.table.DefaultTableModel;

public class dpayment extends Frame implements ActionListener

{

JTable table1 =new JTable();

JScrollPane sp;

JButton bsearch=new JButton("Search");

JButton bprint=new JButton("Print");

JButton bclose=new JButton("Close");

JLabel lbldate=new JLabel("Enter Date(Eg:01/01/2017):");

JTextField txdate=new JTextField(10);

String m,n,o,p;

JTable table2;

String[] col={"Receipt No","Booking ID","Amount paid"};

Object[] [] data={{m,n,o}};

DefaultTableModel model=new DefaultTableModel(col,0);

public dpayment()

{

setLayout(null);

setSize(1200,1500);

setVisible(true);

table2=new JTable(data,col);

sp=new JScrollPane(table2);

table2.setModel(model);

setBackground(new Color(234,255,229));

lbldate.setBackground(new Color(250,150,80));

lbldate.setFont(new Font("Comic Sans MS",Font.BOLD,20));

lbldate.setBounds(500,300,450,60);

txdate.setBounds(810,300,140,60);

table2.setBounds(500,400,800,200);

sp.setBounds(500,400,800,200);

table2.setFillsViewportHeight(true);

add(sp);

add(lbldate);

add(txdate);

add(bsearch);

add(bprint);

add(bclose);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

bsearch.setBounds(1050,300,100,50);

bprint.setBounds(1150,300,100,50);

bclose.setBounds(1250,300,100,50);

add(bclose);

add(bsearch);

add(bprint);

bsearch.addActionListener(this);

bclose.addActionListener(this);

bprint.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

table2.setModel(model);

if(e.getSource()==bsearch)

{

int flg=0;

model.setRowCount(0);

char d1=txdate.getText().charAt(0);

char d2=txdate.getText().charAt(1);

char d3=txdate.getText().charAt(2);

char d4=txdate.getText().charAt(3);

char d5=txdate.getText().charAt(4);

char d6=txdate.getText().charAt(5);

char d7=txdate.getText().charAt(6);

char d8=txdate.getText().charAt(7);

char d9=txdate.getText().charAt(8);

char d10=txdate.getText().charAt(9);

if((d1>='0' && d1<='9')&&(d2>='0' && d2<='9')&&(d4>='0' && d4<='9')&&(d5>='0' && d5<='9')&&(d7>='0' && d7<='9')&&(d8>='0' && d8<='9')&&(d9>='0' && d9<='9')&&(d10>='0' && d10<='9')&&d3=='/' && d6=='/')

////////////////////////////////////////

{

try

{

Connection con;

ResultSet res;

String[] col={"Receipt No","Booking ID","Amount Paid"};

String bdate=txdate.getText();

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from electricity where paymentdate='"+bdate+"'");

while(res.next())

{

flg=1;

Vector v=new Vector();

v.add(res.getString(1));

v.add(res.getString(2));

v.add(res.getString(9));

model.addRow(v);

}

if(flg==0)

JOptionPane.showMessageDialog(null,"Date Not Found!!!!");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

try

{

Connection con;

ResultSet res;

//table2.setModel(model);

String bdate=txdate.getText();

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from water where paymentdate='"+bdate+"'");

while(res.next())

{

flg=1;

Vector v=new Vector();

v.add(res.getString(1));

v.add(res.getString(2));

v.add(res.getString(5));

model.addRow(v);

}

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

}

else

JOptionPane.showMessageDialog(null,"Invalid Date!!!!");

}

if(e.getSource()==bprint)

{

try

{

Connection con;

ResultSet res;

PrintWriter objpr=new PrintWriter("dpayment.txt");

objpr.println("Receipt No \t Booking ID \t Flat/Duplex No");

objpr.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();

res=st.executeQuery("Select \* from electricity where paymentdate='"+txdate.getText()+"'");

while(res.next())

{

objpr.println(res.getString(1)+"\t\t"+res.getString(2)+"\t\t"+res.getString(9));

}

/\*

Connection con;

ResultSet res;

con=DriverManager.getConnection("jdbc:odbc:sms");

Statement st=con.createStatement();\*/

res=st.executeQuery("Select \* from water where paymentdate='"+txdate.getText()+"'");

while(res.next())

{

objpr.println(res.getString(1)+"\t\t"+res.getString(2)+"\t\t"+res.getString(5));

}

objpr.close();

Runtime robj=Runtime.getRuntime();

Process pr=robj.exec("notepad.exe d:\\new\\dpayment.txt");

}

catch(Exception ex)

{

System.out.println("Error:"+ex);

}

/\*objpr.close();

Runtime robj=Runtime.getRuntime();

Process pr=robj.exec("notepad.exe d:\\new\\dpayment.txt");\*/

}

if(e.getSource()==bclose)

{

this.dispose();

}

}

public static void main(String argv[])throws IOException

{

dpayment obj=new dpayment();

}

}

Implementation

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation is education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions are made regarding what equipment’s has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the user confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system.

The major elements of implementation plan are test plan, training plan, equipment installation plan and a conversion plan.

There are three types of implementation:

1. Implementation of a computer system to replace a manual system.
2. Implementation of a new computer system to replace an existing system.
3. Implementation of a modified application to replace an existing one, using the same computer.

Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. It has been observed that even the best system cannot show good result if the analysis managing the implementation do not attend to every important details. This is an area where the system analysis need to work with utmost care.

**Implementation Tools**

1. Training personnel
2. Conversion Procedures
3. Post-Implementation Review

**Training of Personnel involved with system:**

Even well designed system can succeed or fail because of the way they are operated and used. Therefore, the quality of training received by the personnel involved with the system in various capacities helps or hinders and may even prevent the successful implementation of management information system.

**Conversion Procedures**:

A conversion is the process of changing from the old system to the new one. It must be properly planned and executed. Four methods are common in use. They are:

* Parallel System
* Direct Conversion
* Pilot System
* 34`Phase In Method

**Parallel System:**

The most secure method of converting from an old to new system is to run both systems in parallel. This method is the safest on because it ensures that in case of any problem in using new system, the organization can still fall back to the old system without the loss of time and money.

**Direct Conversion:**

This method converts from the old system to new system abruptly, sometimes over a weekend or even overnight. The old system is used until the planned conversion day, when it is replaced by the new system.

**Pilot System:**

Pilot approach is often preferred in the case of the new system which involves new techniques or some drastic changes in organizations performance. In this method, a working version of the system is implemented in one part of the organization, such as a single work area or department.

**Phase In Method:**

This method is used when it is not possible to install new system throughout an organization all at once. The conversion of files, training of personnel or arrival of equipment may force the staging of the implementation over a period of time, ranging from weeks to months.

**Post Implementation Review:**

After the system is implemented and conversion is completed, a review should be conducted to determine whether the system is meeting the expectations and where improvements are needed. A post implemented review measures the systems performance against predefined requirement. It determines how well the system continues to meet the performance specifications.

Testing Phase

The basic goal of the software development process is to produce software that has no errors or very few errors. In an effort to detect errors soon after they are introduced, each phase ends with verification activity such as a review.

As testing is the last phase before the final software is delivered, it has the enormous responsibility of detecting any type of error that may be in the software. Software typically undergoes changes even after it has been delivered. And to validate that a change has not affected some old functionality of the software, so regression testing is performed.

**Levels of Testing:**

The basic levels of testing are:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

The different levels of testing attempts to detect different types of fault.

|  |  |
| --- | --- |
| Client Needs | Acceptance Testing |
| Requirements | System Testing |
| Design | Integration Testing |
| Code | Unit Testing |

Conclusion

This section provides us an opportunity to do self-introspection of what value we have added to our knowledge and skill set and to the project.

**Exposure to different technology**

Working on Visual Basic 6.0 was a very enriching experience. It added to our knowledge other things which are apart from the syllabus. We came to know about the connectivity in details and also other controls along with their properties.

**Database Implementation**

We worked on MS-Access after a long time and learnt many new things about database creation, concept of stored procedure etc. We learnt many things theoretically as part of our curriculum but here we got a chance to apply our knowledge practically and enhance it also at the same time.

Restaurant Management System can prove to be an excellent software for the management of customers’ bills and assets. It uses Visual Basic 6.0 as its front end and MS-Access as its backend. It is economically feasible as it doesn’t require much capital. Overall it can be used for an effective Restaurant Management.

Future scope & limitations

**Future scope of project:**

Completion of the development process will result in a software package that will provide user-friendly environment, which is very easy to work with, even for people with very little knowledge of computer.

Management of various tasks is incorporated in the package and will deliver the required information in a very easy to use and easy to access manner.

Future of this application can be great. The application can be implemented in the organization for better service of itself and also the response of the customers.

This package will provide accuracy, efficiency, speed and easiness to the end user. Since the system is verified with valid as well as invalid data and is run with an insight into the necessary modifications that may be required in the future, it can be maintained successfully without much effort.

**Limitations:**

There are certain limitations of this project which can be improved in near future for better performance.

* Application cannot be used online.
* Deployment of the project on Pocket-PC i.e., Mobile Application.
* Passwords cannot be changed.

BIBLIOGRAPHY

**BOOKS:**

* **Title :** Java 6.0 made easy
* **Author :** Kogent
* **Publisher :** DreamTech
* **Title :** Java 6.0 How to program
* **Author :** E Balagurusamy
* **Publisher :**  Tata McGraw-Hill
* **Author :** Bill Venners
* **Publisher :** Tata McGraw-Hill

**URL’s**

* www.dreamtechpress.com