

***SUBMITTED TO***

**Dr. Pakeeza Akram**

**Object Oriented Programming**

**NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY**

School of Electrical Engineering and Computer Sciences

***SUBMITTED BY***

**M ABDULLAH-265362**

**Masood Tariq-258392**

**Adil Bashir-301995**

Smart Hostel System

Table of Contents

[Abstract 3](https://d.docs.live.net/4e5e2a74c6898616/BSCS/2nd%20Semester%20(Spring-2019)/Introduction%20to%20Management/Assignment/Hostel%20Management%20through%20Online%20App/Idea%20Report.docx#_Toc8646006)

[Introduction : 3](#_Toc8646008)

Method :

I) login 3

[II) Expenses: 4](#_Toc8646010)

[III) Checkout: 5](#_Toc8646011)

I[V) Checkin: 8](#_Toc8646012)

[V) Employee: 10](#_Toc8646013)

[VI) SQL Connection: 12](#_Toc8646014)

[VII) Reservation: 12](#_Toc8646015)

[) Main: 13](#_Toc8646016)

Results 29

Discussion 30

Conclusion 31

# Abstract

A smart Hostel System (SHS) has designed to provide a computerized process that is stress free, reliable and quick through the use of **java** computer programming language and **MySQL** database application to both the students an the staff in charge of the registration and hostel management processes.

This system is designed in favour of the hostel management which helps them to **save the records of the students** about their rooms and other things.

It helps them from the **manual work** from which it is very difficult to find the record of the students and the **mess bills** of the students, and the information of about those ones who had left the hostel years before. This system **gives an idea** about how a student and fee details, room allocation, mess expenditure are maintained in a better way.

The hostel management system will also contain **special features** like how many students are in a room, student’s id and free rooms or space available. The administration has a **unique identity** for each member as well as students details.

So, the Solution of this is that we introduce **an application which will solve all their issues** for finding a good Hostel in any city.

# Introduction

Smart Hostel System is a java application that has solutions for all the hostel management problems. Hostel management by manual way is tedious process, since it involves

work load and time consumption. In this system, we can easily manage the

hostel details, room details, student records, mess expenditure, mess bill

calculation, easy way of room allocation and hostel attendance.

The main feature of this project is easy to allocate for the student and also

easy to calculate mess bill.

# Method

This project is carried out using java as a programming as well as an front end language. MySql has also implemented to give more and more storage to the data as well as access to the specific persons. It can be accessed by the persons selected by the owner. The following OOP concepts have been implemented in the project:

* **Object**
* **Class**
* **Inheritance**
* **Polymorphism**
* **Abstraction**
* **Encapsulation**

The detailed description of the project has explained below:-

## I)Login:

// The login class consist of user id, status, password as variables and stores them in getter and setter methods so that they can be accessed. There is also an constructor to initialize the variables used in the class login.

public class login {

int id;

String user;

String pass;

String status;

public login(int id, String user, String pass, String status)

{

this.id=id;

this.user=user;

this.pass=pass;

this.status=status;

}

public int getid()

{

return id;

}

public String getuser()

{

return user;

}

public String getpass()

{

return pass;

}

public String getstatus()

{

return status;

}

}

## II)Expenses:

// The classExpens class gets the user id, name, expenses, date and stores them in getter and setter methods so that they can be accessed. There is also an constructor to initialize the variables used in the class classExpens.//

public class classExpens {

int id;

String name;

String nature;

float Expens;

String date;

public classExpens(int id, String name, String nature, float Expens, String date){

this.id=id;

this.name=name;

this.nature=nature;

this.Expens=Expens;

this.date=date;

}

public int getid()

{

return id;

}

public String getname(){

return name;

}

public String getnature()

{

return nature;

}

public float getexpens()

{

return Expens;

}

public String getdate()

{

return date;

}

}

## Checkout:

// The checkout class has variables the user id, name, father, address, nic, date, city, country, status and stores them in getter and setter methods so that they can be accessed. There is also an constructor to initialize the variables used in the class Checkout .//

//class chechouthaving details of persons checking out.

public class ClassCheckOut { //class starts from here.

private int id; //private data members.

private String name;

private String father;

private String address;

private long nic;

private String date;

private String odate;

private long phone;

private String country;

private String city;

private int adult;

private int children;

private String roomtype;

private int roomno;

private float roomcost;

private float taxes;

private float total;

//overloaded constructor.

public ClassCheckOut(int id, String name, String father, String address, long nic,String date,String odate, long phone, String country, String city, int adult, int children, String roomtype, int roomno, float roomcost, float taxes, float total)

{

//use of this reference.

this.id=id;

this.name=name;

this.father=father;

this.address=address;

this.nic=nic;

this.phone=phone;

this.date=date;

this.odate=odate;

this.country=country;

this.city=city;

this.adult=adult;

this.children=children;

this.roomtype=roomtype;

this.roomno=roomno;

this.taxes=taxes;

this.total=total;

}

//getter method for id.

public int getid()

{

return id;

}

//getter method for name.

public String getname()

{

return name;

}

//getter method for father.

public String getfather()

{

return father;

}

//getter method for Address.

public String getaddress()

{

return address;

}

//getter method for nic.

public long getnic()

{

return nic;

}

//getter method for phone.

public long getphone()

{

return phone;

}

public String getdate(){

return date;

}

public String getodate()

{

return odate;

}

public String getcountry()

{

return country;

}

public String getcity()

{

return city;

}

public int getadult()

{

return adult;

}

public int getchildren()

{

return children;

}

public String getroomtype()

{

return roomtype;

}

public int getroomno()

{

return roomno;

}

public float getroomcost()

{

return roomcost;

}

public float gettaxes()

{

return taxes;

}

public float gettotal()

{

return total;

}

}

## 

## III)Checkin:

// this class is used for check in purposes.

public class ClassCheckIn { //public class starts from here.

private int id; //private data members.

private String name;

private String father;

private String address;

private long nic;

private String date;

private long phone;

private String country;

private String city;

private int adult;

private int children;

private String roomtype;

private int roomno;

private float roomcost;

private float taxes;

private float total;

//overloaded constructor.

public ClassCheckIn(int id, String name, String father, String address, long nic,String date, long phone, String country, String city, int adult, int children, String roomtype, int roomno, float roomcost, float taxes, float total)

{

//use of this reference.

this.id=id;

this.name=name;

this.father=father;

this.address=address;

this.nic=nic;

this.phone=phone;

this.date=date;

this.country=country;

this.city=city;

this.adult=adult;

this.children=children;

this.roomtype=roomtype;

this.roomno=roomno;

this.taxes=taxes;

this.total=total;

}

//getter method for id.

public int getid()

{

return id;

}

//getter method for name.

public String getname()

{

return name;

}

//getter method for father.

public String getfather()

{

return father;

}

//getter method for address.

public String getaddress()

{

return address;

}

//getter method for nic.

public long getnic()

{

return nic;

}

//getter method for phone.

public long getphone()

{

return phone;

}

//getter method for date.

public String getdate(){

return date;

}

//getter method for country.

public String getcountry()

{

return country;

}

//getter method for city.

public String getcity()

{

return city;

}

//getter method for adult.

public int getadult()

{

return adult;

}

public int getchildren()

{

return children;

}

public String getroomtype()

{

return roomtype;

}

public int getroomno()

{

return roomno;

}

public float getroomcost()

{

return roomcost;

}

public float gettaxes()

{

return taxes;

}

public float gettotal()

{

return total;

}

}

## Employee:

// The Employee class has variables the user id, name, nic, date, phone, salary and stores them in getter and setter methods so that they can be accessed. There is also an constructor to initialize the variables used in the class Employee .//

//class employee .

public class classEmployee {

private int id;

private String name;

private String father;

private long nic;

private long phone;

private String desig;

private float salary;

public classEmployee(int id, String name,String father,long nic, long phone, String desig, float salary)

{

this.id=id;

this.name=name;

this.father=father;

this.nic=nic;

this.phone=phone;

this.desig=desig;

this.salary=salary;

}

public int getid()

{

return id;

}

public String getname()

{

return name;

}

public String getfather()

{

return father;

}

public long getnic()

{

return nic;

}

public long getphone()

{

return phone;

}

public String getdesig(){

return desig;

}

public float getsalary()

{

return salary;

}

}

## SQL Connection:

// the sql connection has implemented by impoting java..sql and declaring connection as null. It then gets the id and password by using the following statement:

**DriverManager.getConnection("jdbc:mysql://localhost/smarthotel","root","12345");**

import java.sql.\*;

import javax.swing.\*;

public class mySqlConnection {

Connection conn = null;

public static Connection ConnectDB(){

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/smarthotel","root","12345");

return conn;

}

catch(ClassNotFoundException | SQLException e){

JOptionPane.showMessageDialog(null, e);

return null;

}

}

}

## Reservation:

public class reserv {

private int id;

private String name;

private long phone;

private String date;

private String roomtype;

private int roomno;

public reserv(int id, String name, long phone, String date, String roomtype, int roomno)

{

this.id=id;

this.name=name;

this.phone=phone;

this.date=date;

this.roomtype=roomtype;

this.roomno=roomno;

}

public int getid()

{

return id;

}

public String getname()

{

return name;

}

public long getphone()

{

return phone;

}

public String getdate(){

return date;

}

public String getroomtype()

{

return roomtype;

}

public int getroomno()

{

return roomno;

}

}

## Main:

import java.awt.print.PrinterException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.MessageFormat;

import java.util.ArrayList;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.RowFilter;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableRowSorter;

public class main extends javax.swing.JFrame {

Connection conn=null;

PreparedStatement pst= null;

ResultSet rs=null;

public main() {

initComponents();

showTable();

showTable2();

}

//----------------------------------------------------------------------------------------filtering tbl\_checkout for search

private void filter(String sql){

DefaultTableModel model=(DefaultTableModel)checkoutTable.getModel();

TableRowSorter<DefaultTableModel>tr=new TableRowSorter

<DefaultTableModel>(model);

checkoutTable.setRowSorter(tr);

tr.setRowFilter(RowFilter.regexFilter(sql));

}

//----------------------------------------------------------------------------------------filtering tbl\_checkin for search

private void filter2(String sql){

DefaultTableModel model=(DefaultTableModel)maintable.getModel();

TableRowSorter<DefaultTableModel>tr=new TableRowSorter

<DefaultTableModel>(model);

maintable.setRowSorter(tr);

tr.setRowFilter(RowFilter.regexFilter(sql));

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Populating Array List Check In Records

public ArrayList<ClassCheckIn> dataList()

{

ArrayList<ClassCheckIn> dataList = new ArrayList<ClassCheckIn>();

conn=mySqlConnection.ConnectDB();

String sql="select \* from guests";

try{

Statement st = conn.createStatement();

rs=st.executeQuery(sql);

ClassCheckIn Data;

while(rs.next()){

Data = new ClassCheckIn(rs.getInt("id"),rs.getString("cus\_name"),rs.getString("cus\_father"),rs.getString("address"),rs.getLong("cus\_nic"),rs.getString("cus\_date"),rs.getLong("phone"),rs.getString("cus\_country"),rs.getString("cus\_city"),rs.getInt("cus\_adult"),rs.getInt("cus\_child"),rs.getString("room\_type"),rs.getInt("room\_no"),rs.getFloat("room\_cost"),rs.getFloat("taxes"),rs.getFloat("total"));

dataList.add(Data);

}

}

catch(Exception ex){

ex.printStackTrace();

}

return dataList;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Display data in checkin table

public void showTable(){

ArrayList<ClassCheckIn> list = dataList();

DefaultTableModel model = (DefaultTableModel)maintable.getModel();

while(maintable.getRowCount()>0)

{

((DefaultTableModel)maintable.getModel()).removeRow(0);

}

Object[]row = new Object[13];

for (int i = 0; i <list.size(); i++) {

row[0]=list.get(i).getid();

row[1]=list.get(i).getname();

row[2]=list.get(i).getfather();

row[3]=list.get(i).getnic();

row[4]=list.get(i).getaddress();

row[5]=list.get(i).getdate();

row[6]=list.get(i).getphone();

row[7]=list.get(i).getcountry();

row[8]=list.get(i).getcity();

row[9]=list.get(i).getadult();

row[10]=list.get(i).getchildren();

row[11]=list.get(i).getroomtype();

row[12]=list.get(i).getroomno();

model.addRow(row);

}

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Populating Array List Check out Records

public ArrayList<ClassCheckOut> dataList2()

{

ArrayList<ClassCheckOut> dataList = new ArrayList<ClassCheckOut>();

conn=mySqlConnection.ConnectDB();

String sql="select \* from checkout";

try{

Statement st = conn.createStatement();

rs=st.executeQuery(sql);

ClassCheckOut Data;

while(rs.next()){

Data = new ClassCheckOut(rs.getInt("id"),rs.getString("cus\_name"),rs.getString("cus\_father"),rs.getString("address"),rs.getLong("cus\_nic"),rs.getString("cus\_date"),rs.getString("out\_date"),rs.getLong("phone"),rs.getString("cus\_country"),rs.getString("cus\_city"),rs.getInt("cus\_adult"),rs.getInt("cus\_child"),rs.getString("room\_type"),rs.getInt("room\_no"),rs.getFloat("room\_cost"),rs.getFloat("taxes"),rs.getFloat("total"));

dataList.add(Data);

}

}

catch(Exception ex){

ex.printStackTrace();

}

return dataList;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Display data in checkin table

public void showTable2(){

ArrayList<ClassCheckOut> list = dataList2();

DefaultTableModel model = (DefaultTableModel)checkoutTable.getModel();

while(checkoutTable.getRowCount()>0)

{

((DefaultTableModel)checkoutTable.getModel()).removeRow(0);

}

Object[]row = new Object[14];

for (int i = 0; i <list.size(); i++) {

row[0]=list.get(i).getid();

row[1]=list.get(i).getname();

row[2]=list.get(i).getfather();

row[3]=list.get(i).getnic();

row[4]=list.get(i).getaddress();

row[5]=list.get(i).getdate();

row[6]=list.get(i).getodate();

row[7]=list.get(i).getphone();

row[8]=list.get(i).getcountry();

row[9]=list.get(i).getcity();

row[10]=list.get(i).getadult();

row[11]=list.get(i).getchildren();

row[12]=list.get(i).getroomtype();

row[13]=list.get(i).getroomno();

model.addRow(row);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

background = new javax.swing.JPanel();

sidebar = new javax.swing.JPanel();

btnReservation = new javax.swing.JButton();

btnCheckIN = new javax.swing.JButton();

btnCheckOUT = new javax.swing.JButton();

btnExpens = new javax.swing.JButton();

btnReport = new javax.swing.JButton();

jButton14 = new javax.swing.JButton();

btnCheckIN.setBackground(new java.awt.Color(33, 55, 116));

btnCheckIN.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

btnCheckIN.setForeground(new java.awt.Color(204, 0, 0));

btnCheckIN.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/vacant.jpg"))); // NOI18N

btnCheckIN.setText(" Check IN");

btnCheckIN.setHorizontalAlignment(javax.swing.SwingConstants.LEADING);

btnCheckIN.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnCheckINActionPerformed(evt);

}

});

btnCheckOUT.setBackground(new java.awt.Color(33, 55, 116));

btnCheckOUT.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

btnCheckOUT.setForeground(new java.awt.Color(204, 0, 0));

btnCheckOUT.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/occupied.jpg"))); // NOI18N

btnCheckOUT.setText(" Check OUT");

btnCheckOUT.setHorizontalAlignment(javax.swing.SwingConstants.LEADING);

btnCheckOUT.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnCheckOUTActionPerformed(evt);

}

});

btnExpens.setBackground(new java.awt.Color(33, 55, 116));

btnExpens.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

btnExpens.setForeground(new java.awt.Color(204, 0, 0));

btnExpens.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/if\_Money-Increase\_379473.png"))); // NOI18N

btnExpens.setText("Expenses");

btnExpens.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

btnExpens.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

btnExpens.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnExpensActionPerformed(evt);

}

});

btnReport.setBackground(new java.awt.Color(33, 55, 116));

btnReport.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

btnReport.setForeground(new java.awt.Color(204, 0, 0));

btnReport.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/planning.png"))); // NOI18N

btnReport.setText("Report");

("/icons/out.jpg"))); // NOI18N

btnExpenses.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

btnExpenses.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnExpensesActionPerformed(evt);

}

});

jButton8.setBackground(new java.awt.Color(33, 55, 116));

jButton8.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

jButton8.setForeground(new java.awt.Color(204, 255, 255));

jButton8.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/history\_icon.jpg"))); // NOI18N

jButton8.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jButton8.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton8ActionPerformed(evt);

}

});

jButton9.setBackground(new java.awt.Color(33, 55, 116));

jButton9.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

jButton9.setForeground(new java.awt.Color(204, 255, 255));

jButton9.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/guest.jpg"))); // NOI18N

jButton9.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jButton9.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton9ActionPerformed(evt);

}

});

jButton11.setBackground(new java.awt.Color(33, 55, 116));

jButton11.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

jButton11.setForeground(new java.awt.Color(204, 255, 255));

jButton11.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/2316703845\_37623b98a9\_m.jpg"))); // NOI18N

jButton11.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jButton11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton11ActionPerformed(evt);

}

});

jButton7.setBackground(new java.awt.Color(33, 55, 116));

jButton7.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

jButton7.setForeground(new java.awt.Color(204, 255, 255));

jButton7.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/room.jpg"))); // NOI18N

jButton7.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jButton7.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton7ActionPerformed(evt);

}

});

refreshAll.setBackground(new java.awt.Color(33, 55, 116));

refreshAll.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

refreshAll.setForeground(new java.awt.Color(204, 255, 255));

refreshAll.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/change.jpg"))); // NOI18N

refreshAll.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

refreshAll.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

refreshAllActionPerformed(evt);

}

});

jButton13.setBackground(new java.awt.Color(33, 55, 116));

jButton13.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

jButton13.setForeground(new java.awt.Color(204, 255, 255));

jButton13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/booking.jpg"))); // NOI18N

jButton13.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

jButton13.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton13ActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N

jLabel4.setForeground(new java.awt.Color(204, 0, 0));

jLabel4.setText("Smart Hostel System");

javax.swing.GroupLayout topLayout = new javax.swing.GroupLayout(top);

top.setLayout(topLayout);

topLayout.setHorizontalGroup(

topLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(topLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jButton7, javax.swing.GroupLayout.PREFERRED\_SIZE, 61, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, topLayout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 56, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

new String [] {

"ID", "Name", "Father", "NIC/Passport", "Address", "IN Date", "OUT Date", "Phone", "Country", "City", "Adult", "Children", "Room Type", "Room No"

}

));

checkoutTable.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

checkoutTableMouseClicked(evt);

}

});

jScrollPane2.setViewportView(checkoutTable);

searchBox1.setBackground(new java.awt.Color(51, 51, 51));

searchBox1.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

searchBox1.setForeground(new java.awt.Color(153, 255, 255));

searchBox1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyReleased(java.awt.event.KeyEvent evt) {

searchBox1KeyReleased(evt);

}

});

btn\_print1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/printer (1).png"))); // NOI18N

btn\_print1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

btn\_print1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btn\_print1ActionPerformed(evt);

}

});

btn\_refresh1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/icons8-refresh-35.png"))); // NOI18N

btn\_refresh1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);

btn\_refresh1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btn\_refresh1ActionPerformed(evt);

}

});

.addComponent(btn\_print, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(searchBox, javax.swing.GroupLayout.PREFERRED\_SIZE, 166, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(113, 113, 113)

.addComponent(jLabel2))

.addGroup(backgroundLayout.createSequentialGroup()

.addComponent(btn\_backward1, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(backgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(backgroundLayout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(jScrollPane2))

.addGroup(backgroundLayout.createSequentialGroup()

.addComponent(btn\_forward1, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(btn\_refresh1, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(btn\_print1, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(searchBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, 166, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(103, 103, 103)

.addComponent(jLabel3)))))

.addContainerGap())))

);

backgroundLayout.setVerticalGroup(

backgroundLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(backgroundLayout.createSequentialGroup()

.addComponent(top, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

(java.awt.event.KeyEvent evt) {

String txt=searchBox1.getText();

filter(txt);

}

private void searchBoxKeyReleased(java.awt.event.KeyEvent evt) {

String txt=searchBox.getText();

filter2(txt);

}

private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {

settings set = new settings();

set.setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(main.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new main().setVisible(true);

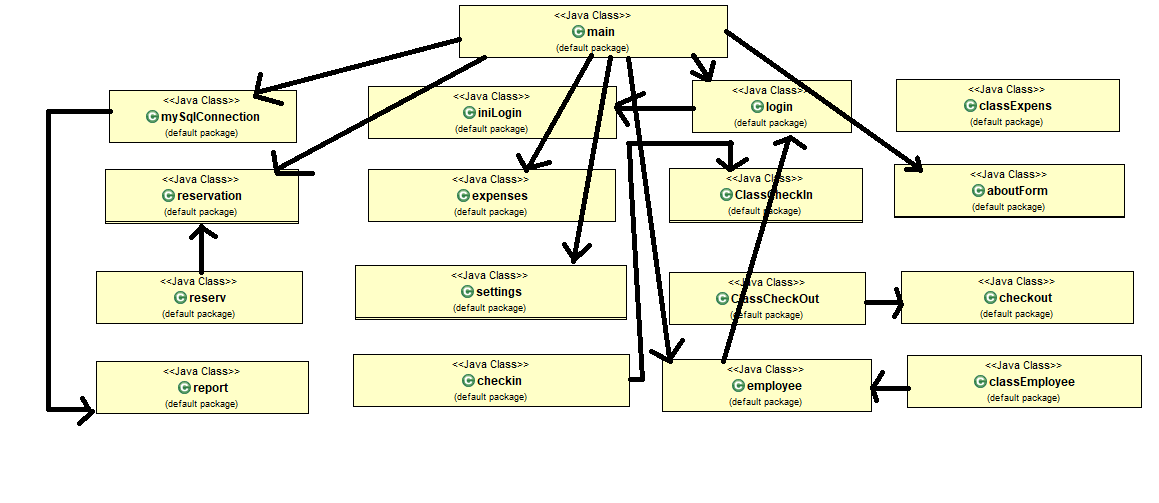
}

});

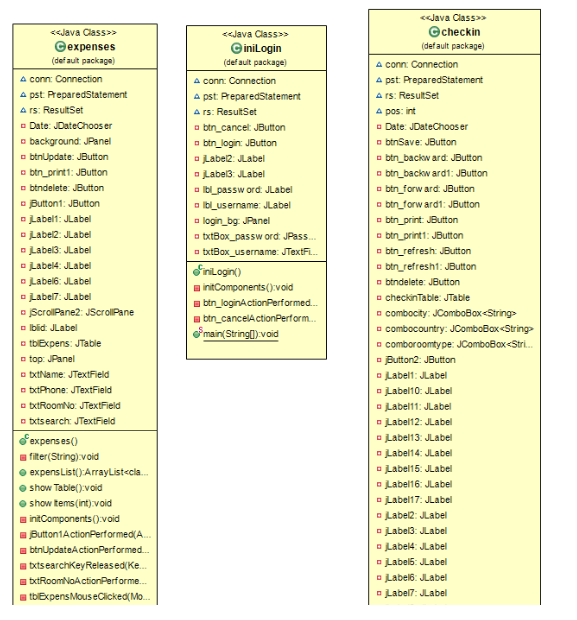
}

// this is the uml diagram for better understanding of the inheritance and polymorphism.

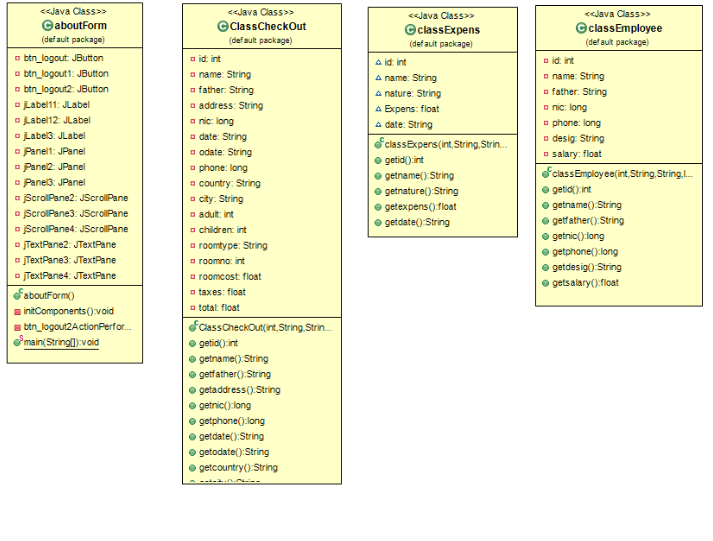
# UML Diagram



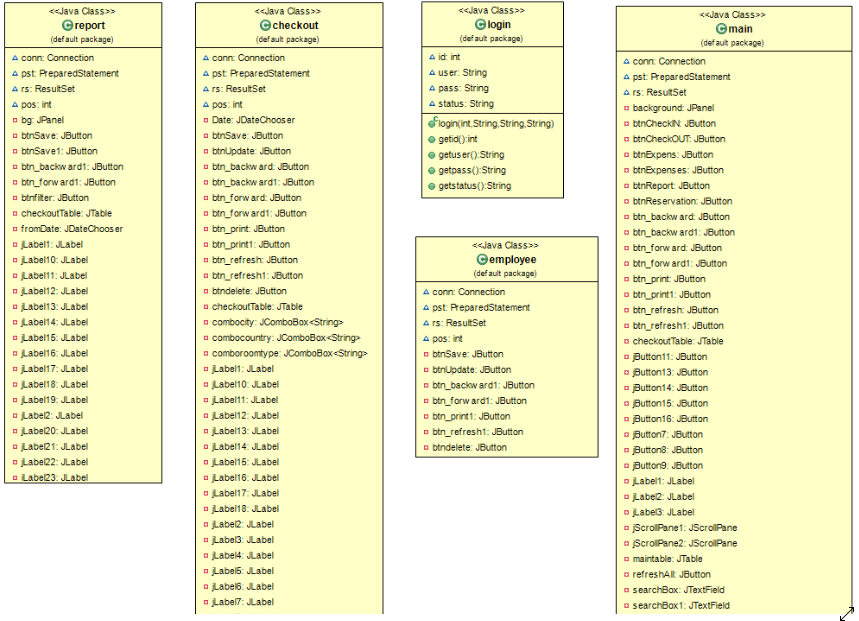
# All Classes



2



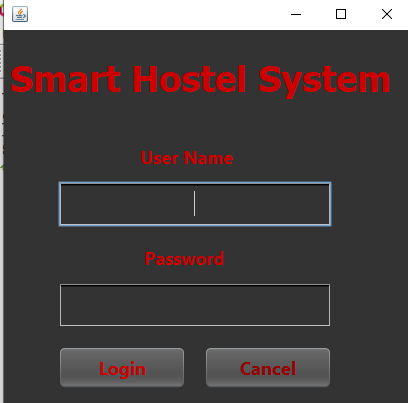
3.

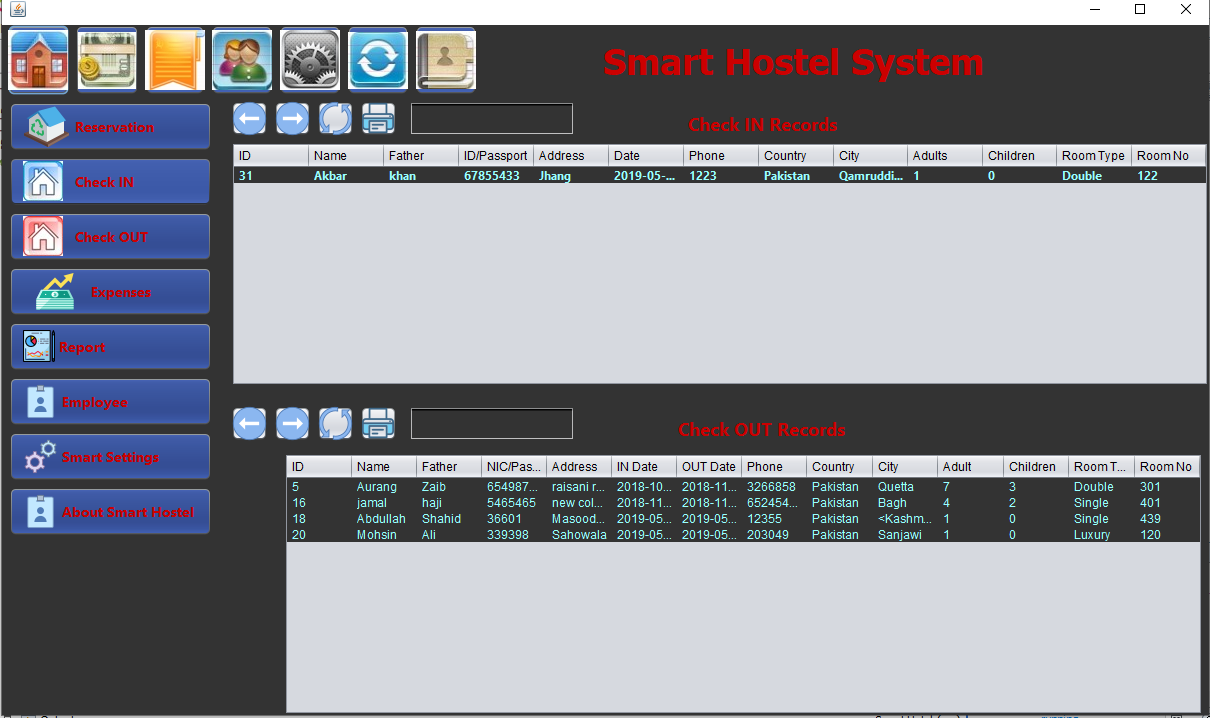


4.

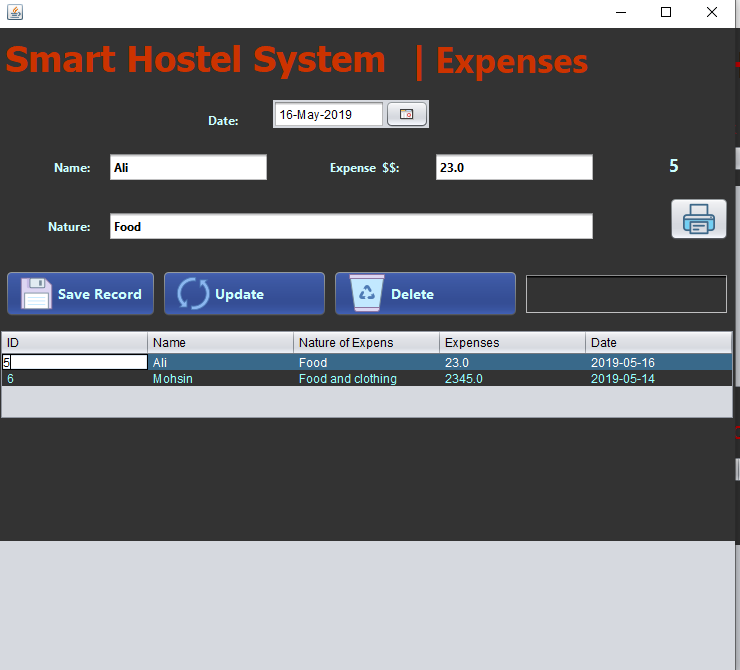
# Results

**Login page**: it takes input from the person and checks whether it matches to the database.



 Main page: This is the main page of the Smart Hostel System. It has all the records of check in,s and check outs. Morever, it has access to all the functions required and embedded in out application like reservations reports, employees, smart settings and about us page.

Expenses page: It saves the expenses of the student. The management can update,delete the records too. Once the required information is added, it can be saved.



# Discussion

### Solution to Students` Issues:

Every year, there are thousands of students who migrate from one city to another city for the sake of higher education in Pakistan. Most of them have to accommodate in hostels. During their stay, they face very difficulties regarding hostel.

The students have to fill out their check in form manually which is actually very time consuming. SHS provides them the facility not to bother with pen and paper rather it just fills out the form by asking their information by the management team.

After tall this, they are **still not happy with their services**. So , these things effect their studies. This process is not good for Students. Because they spend both money and time for finding a good Hostel. It provides them **automatically generated** receipts of their expenses and hence no need to worry about their expenses. It also provides **reservations** to the students coming for **temporary visit** for example any admission test.

### Solution to Management Team`s Issues:

Lets see the second aspect of this picture. In our current era of automated systems with it being either software or hardware, it’s not advisable to be using manual system.

* Hostels without a management system are usually done manually.

It has solved by embedding many function is SHS so that they don’t need to do anything manually.

* Registration forms verification to other data saving processes are done manually and most at times, they are written on paper.

# 

# Conclusion

OOP is great because it encourages thinking in a black box. We used concepts, rather than details. OOP gave us many options how we can implement our solution. With many options come many places where it can go wrong. And so people of FP say OOP adds unnecessary complexity. And that is true, but only for the cases when OOP is done wrong. Proper OOP makes code easier to read, understand and maintain.We learned how to Keep your analysis in the business world, using business words, with encapsulation of related strings and numbers into classes.