

INDIVIDUAL REPORT

Anoj Aryal

2146495

CIS017-1 – Computer Systems Structure

CIS095-1 – Principles of programming 2021-2022

Assignment 1 – Design and Implement a

Desktop Application

**Table of Contents**

[INTRODUCTION 3](#_Toc113411241)

[ASSUMPTIONS 3](#_Toc113411242)

[TASK DESCRIPTION 4](#_Toc113411243)

[PROJECT PLAN 5](#_Toc113411245)

[REQUIREMENT ANALYSIS 6](#_Toc113411246)

[NON-FUNCTIONAL REQUIREMENTS 8](#_Toc113411247)

[USABILITY REQUIREMENTS 8](#_Toc113411248)

[USE CASE DIAGRAM 9](#_Toc113411249)

[ERM 10](#_Toc113411250)

[SKELETON TABLE 11](#_Toc113411251)

[DATA DICTONARY 12](#_Toc113411252)

[USER INTERFACE DESIGN 22](#_Toc113411253)

[IMPLEMENTATION 28](#_Toc113411254)

[TESTING 29](#_Toc113411255)

[DISCUSSION / CRITICAL ANALYSIS /REFLECTION 32](#_Toc113411256)

[CONCLUSION 33](#_Toc113411257)

[REFRENCES 34](#_Toc113411258)

[APPENDIX 34](#_Toc113411259)

# INTRODUCTION

In this project we are basically designing the POP module and desktop application where, a hotel would like to provide an online service for their customers including all the facilities that includes booking rooms and even checking the upcoming trips.

A desktop application needs to be created to store data and information of different categories (Corporate – Non corporate) customer. If the customer is a corporate customer than they can directly login and book the rooms as per required. All the required information of customer like (name, DOB, contact no., email, address) are taken at time of login. If the customer is non-corporate than first of all they should register to the page after the registration sign in process continues and again the details are taken as before. After the sign in process is completed booking is done as per customer requirement. And also all of the hotel booking information's are kept that basically includes start date to end date and what kind of room they are booking (single or double) with all the required facilities included.

After the booking is made receptionist must assign an appropriate room to each booking, matching all the customer demand. At last desktop application must be created to store all information of customer and all of the billing things should be taken care by the receptionist at the time of departure.

# ASSUMPTIONS

* A hotel online booking system is created in POP taking all the customer details.
* A customer is identified weather they are corporate or a non-corporate customer.
* After that registration and login are made to the hotel booking page.
* Credit card information will be verified upon log in and it is possible some registered users may enter invalid information.
* Room rates may vary due to high or low season, according the facilities provided in the room and the room view.
* The hotel may provide services according to the criteria of the customer. Facilities may vary according to the customer choices.
* Booking confirmations would be done quickly upon request to ensure customer satisfaction.
* The hotel will provide WI-FI and therefore Customers will not need to request this service.
* All the payments of customer are made by the receptionist at last at the time of departure if the customer is non-corporate.
* A customer can pay the bills through credit card and change these details or add another card details in the future if needed.
* In case of corporate customer services are provided according to the packages taken by them.

# TASK DESCRIPTION

Luton Hotel is a large independently run hotel located in the town of Luton. It offers a variety of accommodation (single, twin and double rooms) all of which offer a suite facility, a telephone with an outside line and a mini-bar. The hotel has a bar and a restaurant, where guests may take refreshment and charge this to their room bill. Room service is also available. Services not offered by the hotel itself (for example, dry cleaning) can be ordered by guests through the hotel receptionist and charged to their room.

# Non-corporate customers must register online before making a booking. Companies need to set up an account with the hotel before making a corporate booking and they are billed at the end of each month for bookings that have been completed since the last invoice was sent. A typical corporate client is Luton University, and during months such as February and June when examination boards are taking place as many as 30 or 40 guests may be staying at the hotel on the University account. At other times there may be only a few. A discount is negotiated for each corporate client when their account is set up and is reviewed annually.

Non-corporate customers who are not registered yet need to register (providing their address, phone number, email address and credit card information). A registered customer should be able to book a room, i.e. provide the check-in and check-out date and if a double or single room is required. Customers should also be able to view their bookings and change or cancel them if needed. All these operations are only possible after a customer signs in (which means they must be registered). Our system should also check if the requested room (single, twin or double room) is available for the time of stay.

After that customer is provided with the booked room with their choices including all the facilities that they have demanded. According to the criteria and requirements of customers different services are provided. At last at the time of departure all of the billings are made by customer payment using credit cards and so on. All the billings are made by receptionist at last.

After that the given task are performed by the students they will have the clear vision and knowledge about development of basic desktop application. They can also acknowledge themselves into having the basic idea that how the business runs. Basic hotel services their online registration process and different things related to the project will be known to the developers.

# PROJECT PLAN

|  |  |
| --- | --- |
| WEEK | TASKS COMPLETED |
| Week 1 | Introduction, Description part completed. |
| Week 2 | Requirements Analysis  Overview of Functional, Technical (Non-Functional) and Usability Requirements are completed |
| Week 3 | User interface design |
| Week 4 | Started GUI designing |
| Week 5 | Data dictionary and GUI completed |
| Week 6 | Coding part was done including backend part |
| Week 7 | Final testing of the program and completion of the report |

# REQUIREMENT ANALYSIS

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirement | Priority\* |
| 1 | The main page containing hotel information should be visible to all viewers of the page. | MUST |
| 2 | Guest users are not allowed to book the rooms | MUST |
| 3 | Attempting to Book the room without registration and login is not allowed. Customer should login to the page to make bookings to the app. | COULD |
| 4 | The menu button will include all the shortcut pages | SHOULD |
| 5 | Place the requirement page and login page next to one other for easy access to the site | SHOULD |
| 6 | Profile page will only display after login is completed | SHOULD |
| 7 | After the login is done menu bar should add the booking shortcuts | SHOULD |
| 8 | Clicking on the logo will direct the user to the home page | SHOULD |
| 9 | The menu bar , logo and book now button must be visible throughout all the pages of the application | MUST |
| 10 | Guest will be able to visit all the rooms and service pages without logging | MUST |
| 11 | Every room and service need their separate page | MUST |
| 12 | A page with discount offer to the users must be created | MUST |
| 13 | During the time of booking , a reservation form should be produced which should fill all the essential information of the customers | MUST |
| 14 | Credit card information also needs to be added to the registration form | SHOULD |
| 15 | Users should be allowed to book their preferred room on the registration form | SHOULD |
| 16 | A users upcoming reservation data should be generated by their user profile | SHOULD |
| 17 | User should be allowed to edit or cancel their reservation form before the check-in date to the hotel | SHOULD |
| 18 | The daily bill of the user must be created from the check-in day | COULD |
| 19 | External services bill must be added to overall bill on a daily basis | SHOULD |
| 20 | Total bill including each day bills must be shown in the user profile | SHOULD |
| 21 | All the reservation data must be preserved as history in user profile after the check-out | SHOULD |
| 22 | After the check-out a review and ratings of the user should be allowed | SHOULD |

# NON-FUNCTIONAL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirements | Priority\* |
| 1 | The processing time of the desktop application should be quick |  |
| 2 | The desktop program must be device-screen adaptive |  |
| 3 | The future changes to the desktop application must be adaptable |  |
| 4 | The system should be operational and no bugs should appear at execution time after the development |  |

# USABILITY REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| Requirement No. | Requirements | Priority\* |
| 1 | The desktop application should be designed with the user in mind |  |
| 2 | The user need must be met in order to obtain sufficient information regarding booking and lodging system |  |
| 3 | Simple forms and reservation system must be created |  |
| 4 | Simple and straight forward interface application must be designed and created |  |

# USE CASE DIAGRAM

**( SmartDraw, LLC, 1994-2022)**“A use case diagram is a dynamic or behaviour diagram in [UML](https://www.smartdraw.com/uml-diagram/). Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.”

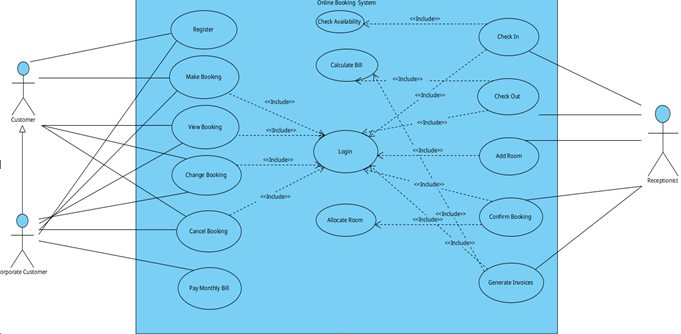


Figure 1 : Use case diagram

# ERM

**(Peterson, 2022)** “ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.”

HOTEL ONLINE BOOKING SYSTEM

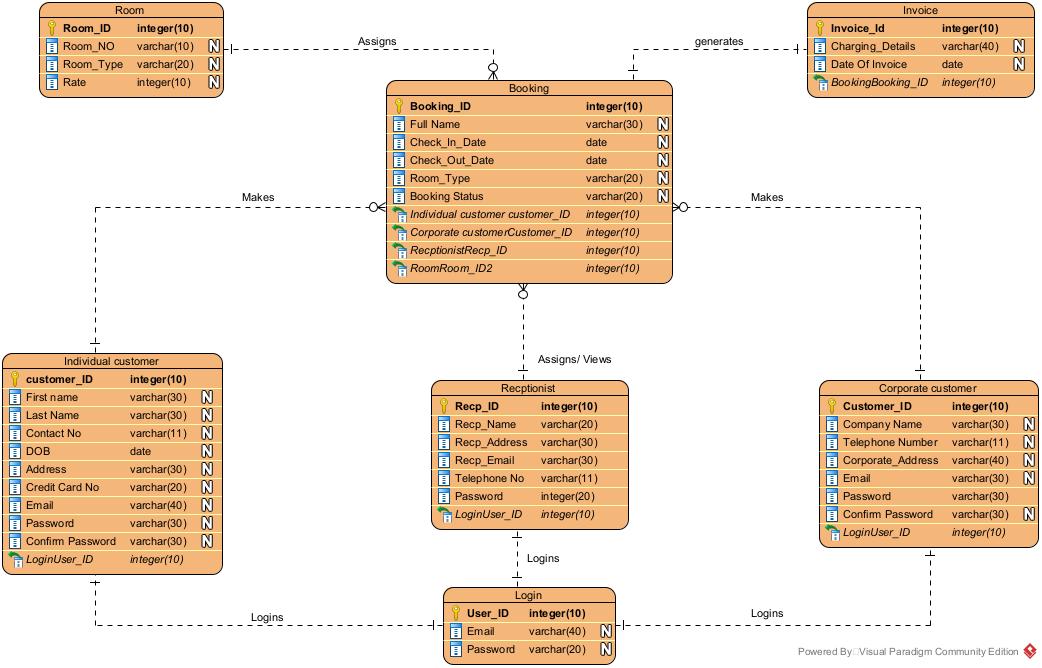
****

Figure 2 : ERM Diagram

# SKELETON TABLE

* Individual Customer (Customer ID, First Name, last name, Address, credit card no, Customer, Email, Password, Confirm Password)
* Corporate Customer (Customer ID, Company Name, Contact No, Email, Password, Password, Confirm Password)
* Receptionist (Receptionist ID, Name, Address, Telephone No, Email, Password, Login ID\*)
* Login (User Id, Email, Password)
* Booking (Booking ID, Check-In-Date, Check-Out-Date, Booking Status, Room Required, Individual CustomerID\*, Corporate CustomerID\*, Receptionist ID\*, Room ID\*)
* Room (Room ID, RoomNo, Room Type, Room Rate)
* Invoice (Invoice ID, Charging Details, Date of Invoice, BookingID\*)

# DATA DICTONARY

**Table 1: Data Dictionary for Individual Customer**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Individual Customer** | | | | | | | |
| **Description: Customer details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Customer ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented to uniquely identifies every customer |
| **First Name** | varchar(30) | 30 |  | No |  |  | First Name of the customer |
| **Last Name** | varchar(30) | 30 |  | No |  |  | Last Name of the customer |
| **Address** | varchar(40) | 40 |  | No |  |  | Address Of The Customer |
| **Credit card No** | Varchar (20) | 20 |  | No |  |  | Card Number of the customer should be unique of everyone |
| **Email** | Varchar(40) | 40 |  |  |  | Must contain an @ and a ‘.’ | Email should be unique from each customer |
| **Password** | varchar(20) | 20 |  | No |  | Must contain at least one capital letter and Special Character | Customer Password |
| **Confirm Password** | Varchar(20) | 20 |  | No |  | Retype the password | Customer Password h |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Customer\_ID | No |

**Table 2: Data Dictionary for Corporate Customer**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Corporate Customer** | | | | | | | |
| **Description: Customer details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Customer ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented to uniquely identifies every customer |
| **Company Name** | varchar(30) | 30 |  | No |  |  | First Name of the customer |
| **Contact No** | varchar(11) | 11 |  | No |  |  | Contact number |
| **Email** | varchar(30) | 30 |  |  |  | Must contain an @ and a ‘.’ | Email should be unique from each customer |
| **Password** | varchar(30) | 30 |  | No |  | Must contain at least one capital letter and Special Character | Customer Password |
| **Confirm Password** | Varchar(30) | 30 |  | No |  | Retype the password | Customer Password |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Customer­\_ID | No |

**Table 3: Data Dictionary for Receptionist**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Receptionist** | | | | | | | |
| **Description: Receptionist details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Receptionist ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented to uniquely identifies every customer |
| **Name** | varchar(40) | 30 |  | No |  |  | Name of the Receptionist |
| **Address** | varchar(30) | 30 |  | No |  |  | Address Of the Receptionist |
| **Telephone No** | varchar(11) | 11 |  | No |  |  | Telephone No of the Receptionist |
| **Email** | Varchar (40) | 40 |  | No |  | Must contain an @ and a ‘.’ | Email should be unique |
| **Password** | varchar(30) | 30 |  | No |  | Must contain at least one capital letter and Special Character | Receptionist Password |
| **LoginID** | int(10) | 10 | FK | No |  | Retype the password | Customer Password h |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Receptionist ID | No |
| FOREIGN | BTREE | Yes | LoginID | No |

**Table 4: Data Dictionary for Login**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Login** | | | | | | | |
| **Description: Login details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Customer ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented to uniquely identifies every customer |
| **Email** | varchar(30) | 30 |  | No |  |  | Email of the customer should be different of each customer |
| **Password** | varchar(20) | 20 |  | No |  |  | Password of the customer |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Customer\_ID | No |

**Table 5: Data Dictionary for Booking**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Booking** | | | | | | | |
| **Description: Booking details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Booking ID (Primary)** | integer | 15 | PK | No |  |  | Auto-incremented to uniquely identifies every customer |
| **Check-In-Date** | Date |  |  | No |  |  | Arrival date of the customer |
| **Check-out-Date** | Date |  |  | No |  |  | Departure date of the customer |
| **Booking Status** | varchar(20) | 20 |  | No |  |  | Pending status of the Booking |
| **Room Required** | Varchar (20) | 20 |  | No |  |  | No of room required for the customer |
| **Individual CustomerID** | Int | 15 | FK | No |  |  | Auto-incremented to uniquely identifies every individual  customer |
| **Corporate**  **CustomerID** | Int | 15 | FK | No |  |  | Auto-incremented to uniquely identifies every corporate  customer |
| **Receptionist ID** | Int | 15 | FK | No |  |  | Auto-incremented to uniquely identifies Receptionist |
| **Room ID** | Int | 15 | FK | No |  |  | Room ID assigned at the time of booking |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Customer\_ID | No |
| FOREIGN | BTREE | Yes | Individual CustomerID | No |
| FOREIGN | BTREE | Yes | Corporate CustomerID | No |
| FOREIGN | BTREE | Yes | Receptionist­\_ID | No |
| FOREIGN | BTREE | Yes | Room\_ID | No |

**Table 6: Data Dictionary for Room**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Room** | | | | | | | |
| **Description: Room details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Room ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented |
| **Room No** | varchar(10) | 10 |  | No |  |  | Room Number |
| **Room Type** | varchar(20) | 20 |  | No |  |  | Preferred room type of the customer |
| **Room Rate** | Varchar(10) | 10 |  | No |  |  | Cost of the room selected by the customer |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Room\_ID | No |

**Table 7: Data Dictionary for Invoice**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table: Invoice** | | | | | | | |
| **Description: Invoice details** | | | | | | | |
| **Field Name** | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| **Invoice ID**  **(Primary)** | integer | 10 | PK | No |  |  | Auto-incremented to uniquely identifies every invoice |
| **Charging Details** | varchar(40) | 40 |  | No |  |  | Details of the payment that is made by the customer |
| **Date of Invoice** | Date |  |  | No |  |  | Payment date of the customer |
| **Booking ID** | Int | 10 | FK | No |  |  | Auto-increment unique from each booking |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | Invoice\_ID | No |
| FOREIGN | BTREE | Yes | Booking\_ID | No |

# USER INTERFACE DESIGN

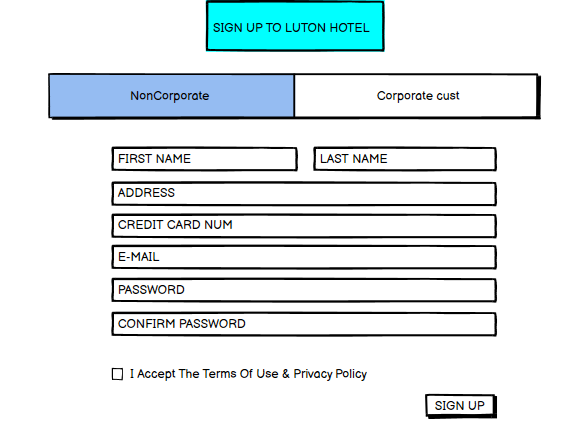


Figure 3 : Non-corporate registration prototype

Above figure is the prototype design for the registration page of the app of the Non corporate customer. Before signing in if the user is not registered than first they should resister to the page filling valid details with their first and last name including a valid email and a password.

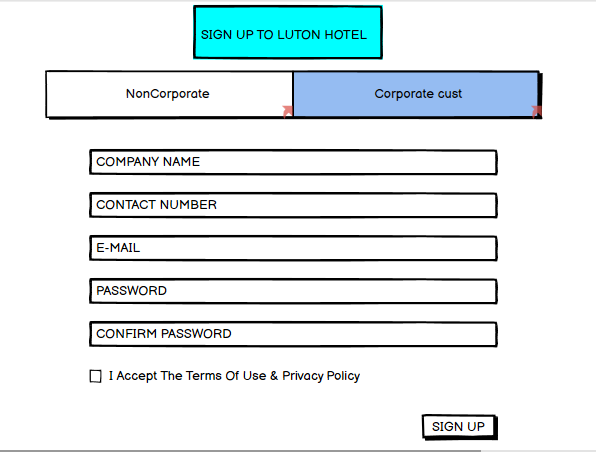


Figure 4: corporate registration prototype

Above figure is the prototype design for the registration page of the app of the corporate customer. Before signing in if the user is not registered than first they should resister to the page filling valid details.

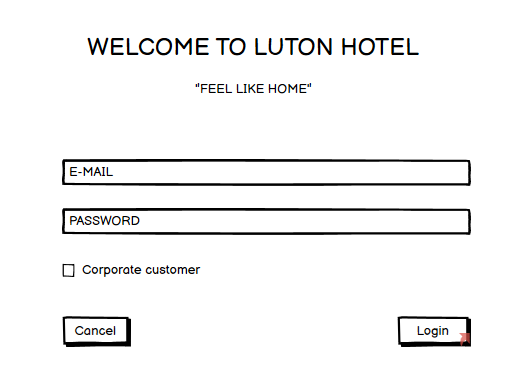


Figure 5: sign in page prototype

Above figure is the prototype design for the login page to the app. User must fil the login page with their registered email and password to login to the page.

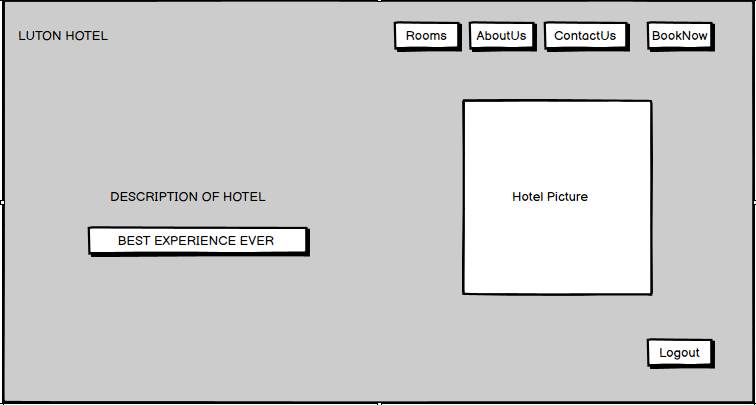


Figure 6: Home page prototype

Above prototype is the home page design of the application. After the login is completed the customer will be directed to the home page from where they can view all the services and facilities of the hotel .

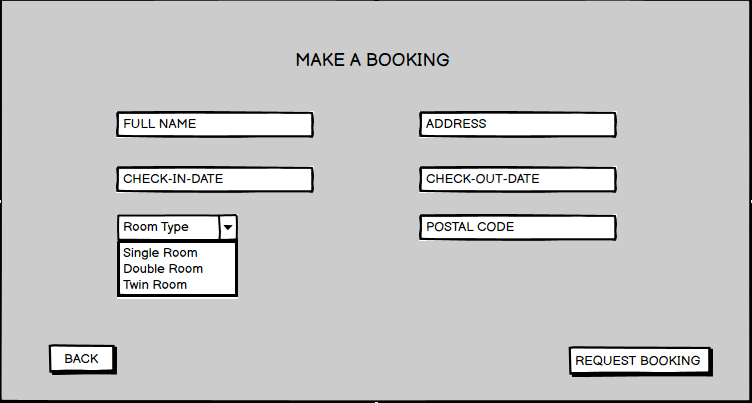


Figure 7: booking page prototype

Above prototype is the booking page design of the application. After the selection and viewings of the room customer can make a booking.

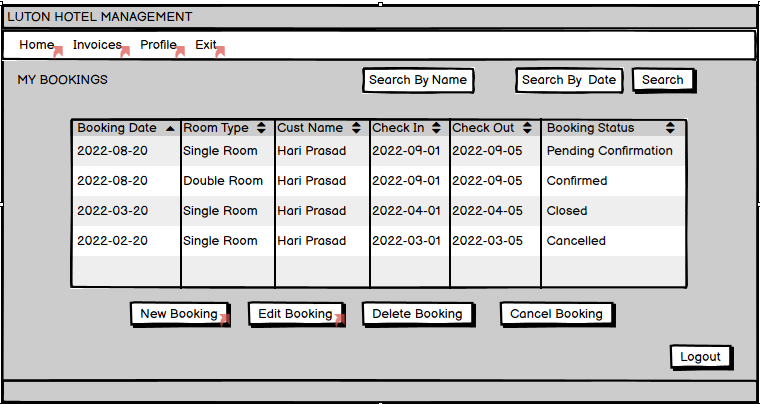


Figure 8: customer booking table prototype

Above figure is the customer booking table prototype from where customer can review the bookings made by them. They can add, edit or cancel the bookings from this page.

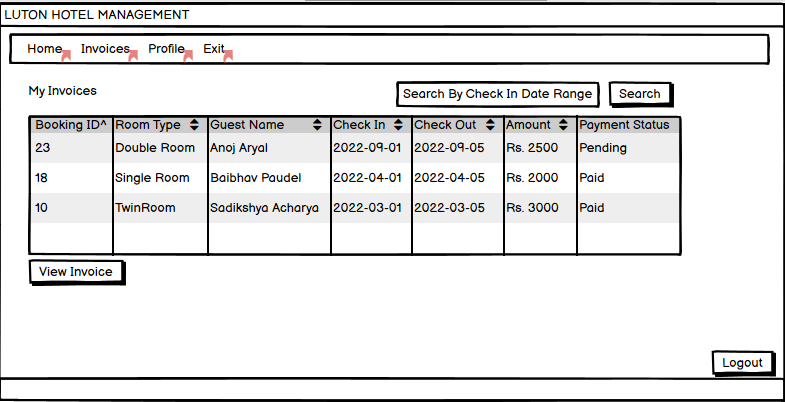


Figure 9 : invoices prototype

Above Figure is the invoice page design prototype from this page billings of the customer is generated at the time of check out.

# IMPLEMENTATION

First of all, at the starting of the project we decided to have Eclipse as our IDE as is very user reliable. It can be partially difficult to learn all the things on it but it has a lots of shortcuts and features that makes programming easier and efficient to perform the given task.

The required application was to be designed and created into GUI and Manager classes as the software was split up into same type which made coding really easy.

Application was designed into different layers using different classes. First of all, GUI was created using JFrames. After that different layers were created to validate the program. Different models and packages were used. Validation was done different statements and queries were used. All the data were connected to the database server. The required statements were retrieved from the database layers.

Many challenges were arrived during the execution as many syntax errors and run time errors were occurred to the application. Administrative part was really very much confusing and lots of error were arised at the time of execution of the application.

Minor mistakes were made in coding the queries which became the cause of the error. But After the proper feedback and several time debugging the program was executed successfully. In this way facing many challenges the application was completed.

# TESTING

1. **User cannot leave the field empty.**

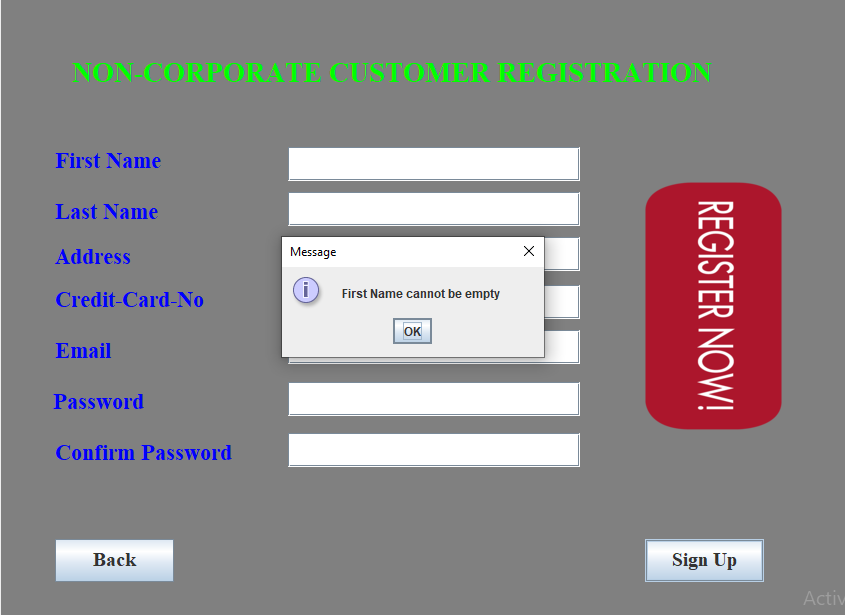
****

Figure 10 : Non-corporate page testing design

Attempting to sign in without registration or without entering the correct details will create the error. Each and every filed must be filled with correct information.

1. **Customer will be able to login after entering the correct details in every filed at the time of registration.**

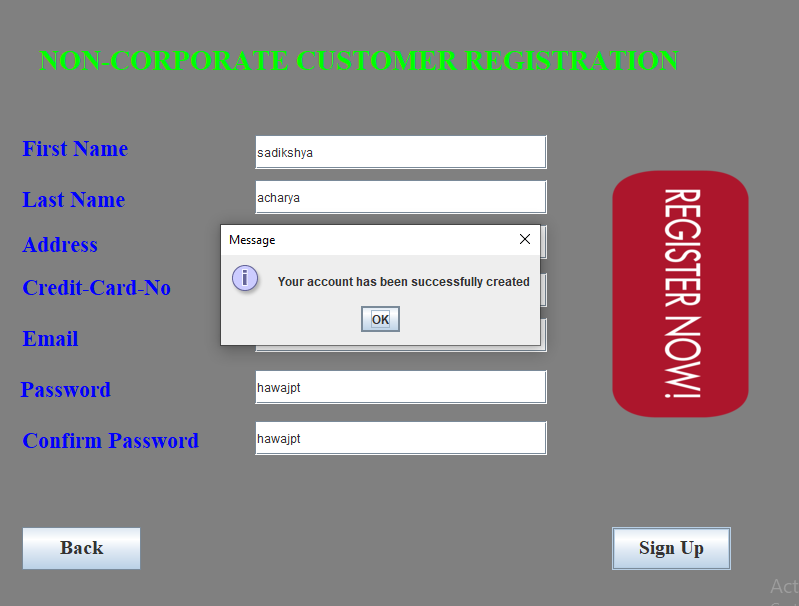


Figure 11: Non-corporate registration successful design

Customer should fill up the form with proper and valid information to each and every field. After that only they will be directed to the login page.

1. **Must enter the registered and valid email and password**

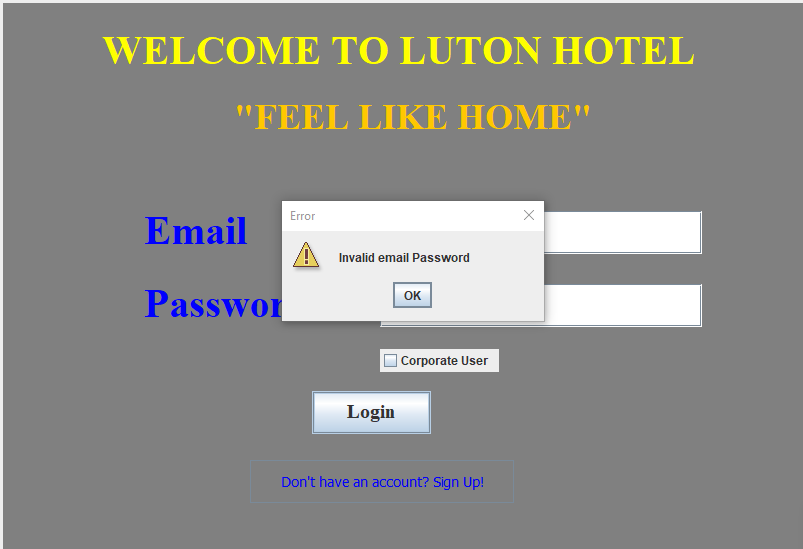
****

Figure 12 : Login page validation

Here to login to the app user must enter the valid email and password if the password is invalid, if the user is not registered then he/she will not be able to login.

1. **After the login is done with proper valid details. Customer is guided to home page.**

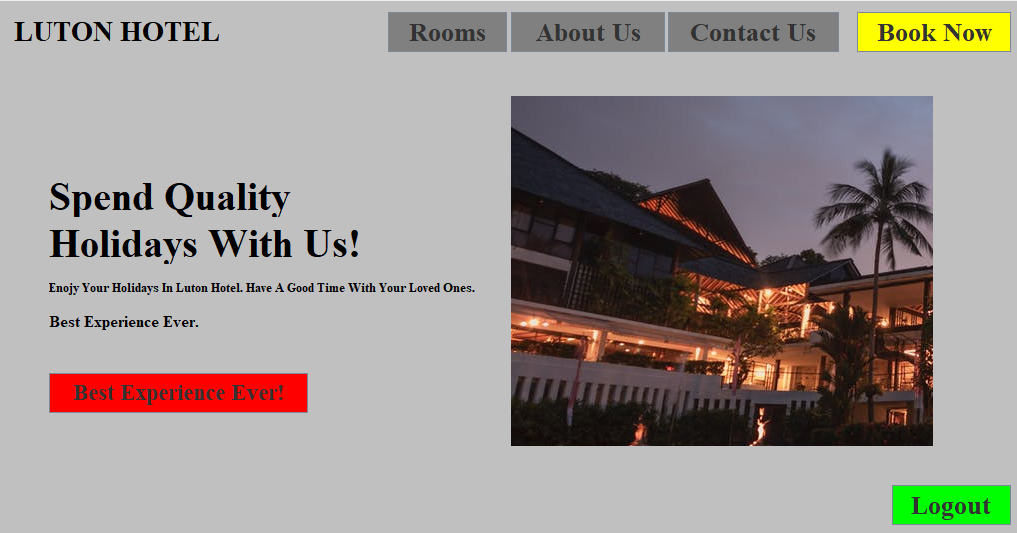
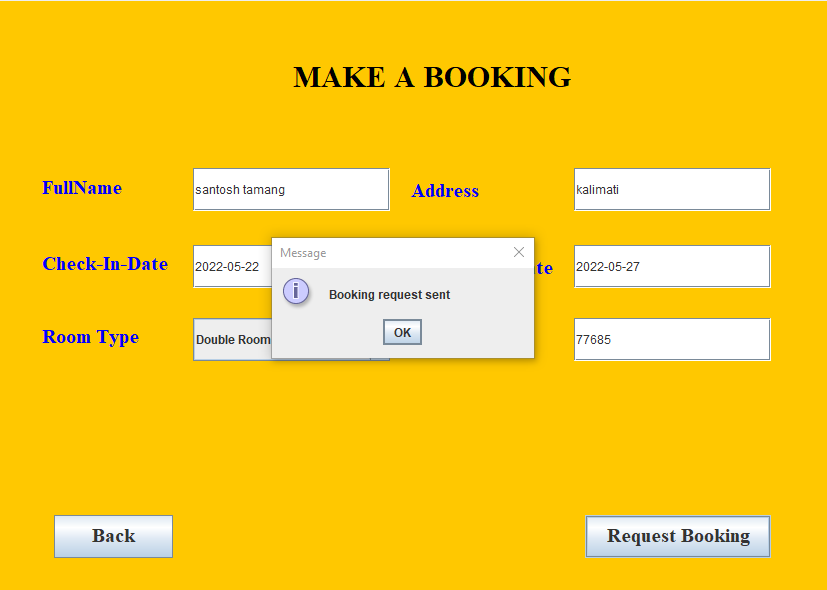


Figure 13: Home page testing

The customer is directed to the home page after the login is made with proper and valid details. Customer can now book a room. Look for the available room with the services in it. Customer can make the payment at the check-out time and so on.

If the customer is corporate customer than he/she will make the payment according to the packages taken by them at the time of check-out with mentioned discount at the time of booking.

1. **Customer can make a booking after login.**



After the login process is completed customer can finally make a bookings entering all the valid details required for bookings.

# DISCUSSION / CRITICAL ANALYSIS /REFLECTION

All the requirements for the assignment is fulfilled. First of all, the customer must register and sign in to the page to make booking. Non-Corporate customer must provide all the required information at the time of registration to the app. Whereas the corporate customer can book the room with the company name too.

The allotted time was sufficient to complete the assignment prior to the due date of the submission.

Throughout the completion time of the assignment ERM was developed using visual paradgyam. Data dictionary were created and the complete coding was done in java using eclipse.

No, the assignment was not that much harder than my expectation. There were many small and logical errors which partially affected the performance. Administrative part was quite harder as compared to others.

But after the proper feedback and guidance of the instructor the accomplishment of assignment became easier at the end. And such mistakes will not be repeated in further assignments in future.

# CONCLUSION

The given assignment was all about creating the desktop application for the hotel online booking system that will provide booking, cancelling and viewing the room facilities to the customer after the registration and signup.

After the booking is made receptionist will view and confirm the booking. Payment will be done by the customer at the time of checkout.

The given assignment was completed and performed on time. Coding part was done with java using eclipse and all the mentioned requirements are fulfilled.

# REFRENCES

SmartDraw, LLC, 1994-2022. *Smartdraw.* [Online]   
Available at: https://www.smartdraw.com/use-case-diagram/  
[Accessed 4 sep 2022].

Peterson, R., 2022. *Guru99.* [Online]   
Available at: https://www.guru99.com/er-diagram-tutorial-dbms.html  
[Accessed 4 sep 2022].

# APPENDIX

**BUSINESS LAYER**

BLBOOKING

package BusinessLayer;

import Models.AllModel;

import Models.Booking;

import Models.Corporate;

import javax.swing.JOptionPane;

import DatabaseLayer.DLBooking;

import DatabaseLayer.DLCorporate;

import Helper.InputException;

public class BLBooking {

Booking booking;

AllModel allmodel;

public BLBooking(){

this.booking = new Booking();

}

public Booking getBooking(){

return booking;

}

public void setBooking(Booking booking) throws InputException{

try{

if(this.validateBooking(booking))

this.booking = booking;

} catch (InputException ex){

throw ex;

}

}

public boolean validateBooking(Booking booking) throws InputException{

if(booking.getFullName() == null || booking.getFullName().length() == 0)

throw new InputException(" FullName cannot be empty");

if(booking.getAddress() == null || booking.getAddress().length() == 0)

throw new InputException(" Address cannot be empty");

if(booking.getCheck\_in\_date() == null || booking.getCheck\_in\_date().length() == 0)

throw new InputException(" Check-IN-Date cannot be empty");

if(booking.getCheck\_out\_date() == null || booking.getCheck\_out\_date().length() == 0)

throw new InputException(" Check-Out-Date cannot be empty");

if(booking.getPostal\_code() == null || booking.getPostal\_code().length() == 0)

throw new InputException(" Postal Code Cannot be empty");

return true;

}

public Booking bookinginf(Booking booking) throws Exception {

//this fuction get the corporateatt detail to log in

try {

DLBooking dlbooking = new DLBooking(booking);

return dlbooking.save();

} catch (Exception e) {

throw e;

}

}

public AllModel assignRoom(AllModel allmodel) throws Exception {

//this fuction get the corporateatt detail to log in

try {

DLBooking dlbooking = new DLBooking(allmodel);

return dlbooking.updateRoom();

} catch (Exception e) {

throw e;

}

}

}

BLBookingDetails

package BusinessLayer;

import Models.AllModel;

import Models.Corporate;

import java.util.ArrayList;

import DatabaseLayer.DLBookingTable;

import DatabaseLayer.DLCorporate;

import DatabaseLayer.RoomDL;

import Helper.InputException;

public class BLBookingDetails {

static AllModel model;

public BLBookingDetails(){

this.model = new AllModel();

}

public AllModel getAllModel(){

return model;

}

public static ArrayList<AllModel> Booking() throws Exception {

try {

DLBookingTable dlbookingtable = new DLBookingTable(model);

return dlbookingtable.getBooking();

}catch(Exception e) {

throw e;

}

}

}

BLBookingTable

package BusinessLayer;

import Models.AllModel;

import Models.Corporate;

import java.util.ArrayList;

import DatabaseLayer.DLBookingTable;

import DatabaseLayer.DLCorporate;

import DatabaseLayer.RoomDL;

import Helper.InputException;

public class BLBookingTable {

static AllModel model;

public BLBookingTable(){

this.model = new AllModel();

}

public AllModel getAllModel(){

return model;

}

public static ArrayList<AllModel> Booking() throws Exception {

try {

DLBookingTable dlbookingtable = new DLBookingTable(model);

return dlbookingtable.getBooking();

}catch(Exception e) {

throw e;

}

}

}

BLCorporate

package BusinessLayer;

import Models.Corporate;

import DatabaseLayer.DLCorporate;

import Helper.InputException;

public class BLCorporate {

Corporate corporate;

public BLCorporate(){

this.corporate = new Corporate();

}

public Corporate getCorporate(){

return corporate;

}

public void setCorporate(Corporate corporate) throws InputException{

try{

if(this.validateCorporate(corporate))

this.corporate = corporate;

} catch (InputException ex){

throw ex;

}

}

public boolean validateCorporate(Corporate corporate) throws InputException{

if(corporate.getCompany\_Name() == null || corporate.getCompany\_Name().length() == 0)

throw new InputException(" Company Name cannot be empty");

if(corporate.getContact() == null || corporate.getContact().length() == 0)

throw new InputException(" Contact no cannot be empty");

if(corporate.getEmail() == null || corporate.getEmail().length() == 0)

throw new InputException(" Email cannot be empty");

if(corporate.getPassword() == null || corporate.getPassword().length() == 0)

throw new InputException(" Password cannot be empty");

if(corporate.getConfirm\_Password() == null || corporate.getConfirm\_Password().length() == 0)

throw new InputException(" Retype the password");

return true;

}

public Corporate corporateinf(Corporate corporate) throws Exception {

//this fuction get the corporateatt detail to log in

try {

DLCorporate dlcorporate = new DLCorporate(corporate);

return dlcorporate.save();

} catch (Exception e) {

throw e;

}

}

public boolean validateLogin(Corporate corporateLogin) throws InputException{

if(corporateLogin.getEmail() == null || corporateLogin.getEmail().length() == 0)

throw new InputException(" Email cannot be empty");

if(corporateLogin.getPassword() == null || corporateLogin.getPassword().length() == 0)

throw new InputException(" Password cannot be empty");

return true;

}

public Corporate logininf(Corporate corporateLogin) throws Exception {

//this fuction get the loginatt detail to log in

try {

DLCorporate dlloginatt = new DLCorporate(corporateLogin);

return dlloginatt.corporate();

} catch (Exception e) {

throw e;

}

}

}

BLNonCorporate

package BusinessLayer;

import Models.AllModel;

import Models.Corporate;

import Models.NonCorporate;

import DatabaseLayer.DLCorporate;

import DatabaseLayer.DLNonCorporate;

import Helper.InputException;

public class BLNonCorporate {

NonCorporate noncorporate;

public BLNonCorporate(){

this.noncorporate = new NonCorporate();

}

public NonCorporate getNonCorporate(){

return noncorporate;

}

public void setNonCorporate(NonCorporate noncorporate) throws InputException{

try{

if(this.validateNonCorporate(noncorporate))

this.noncorporate = noncorporate;

} catch (InputException ex){

throw ex;

}

}

public boolean validateNonCorporate(NonCorporate noncorporate) throws InputException{

if(noncorporate.getFirst\_Name() == null || noncorporate.getFirst\_Name().length() == 0)

throw new InputException(" First Name cannot be empty");

if(noncorporate.getLast\_Name() == null || noncorporate.getLast\_Name().length() == 0)

throw new InputException(" Last Name cannot be empty");

if(noncorporate.getAddress() == null || noncorporate.getAddress().length() == 0)

throw new InputException(" Address cannot be empty");

if(noncorporate.getEmail() == null || noncorporate.getEmail().length() == 0)

throw new InputException(" Email cannot be empty");

if(noncorporate.getPassword() == null || noncorporate.getPassword().length() == 0)

throw new InputException(" Password cannot be empty");

if(noncorporate.getCredit\_Card\_No() == null || noncorporate.getCredit\_Card\_No().length() == 0)

throw new InputException(" credit card no cannot be empty");

if(noncorporate.getConfirm\_Password() == null || noncorporate.getConfirm\_Password().length() == 0)

throw new InputException(" Retype the password");

return true;

}

public NonCorporate noncorporateinf(NonCorporate noncorporate) throws Exception {

//this fuction get the corporateatt detail to log in

try {

DLNonCorporate dlnoncorporate = new DLNonCorporate(noncorporate);

return dlnoncorporate.save();

} catch (Exception e) {

throw e;

}

}

public boolean validateLogin(AllModel model) throws InputException{

if(model.getEmail() == null || model.getEmail().length() == 0)

throw new InputException(" Email cannot be empty");

if(model.getPassword() == null || model.getPassword().length() == 0)

throw new InputException(" Password cannot be empty");

return true;

}

public AllModel logininf(AllModel model) throws Exception {

//this fuction get the loginatt detail to log in

try {

DLNonCorporate dllogin = new DLNonCorporate(model);

return dllogin.model();

} catch (Exception e) {

throw e;

}

}

}

RoomBL

package BusinessLayer;

import Models.AllModel;

import Models.Corporate;

import java.util.ArrayList;

import DatabaseLayer.DLCorporate;

import DatabaseLayer.RoomDL;

import Helper.InputException;

public class RoomBL {

AllModel model;

public RoomBL(){

this.model = new AllModel();

}

public AllModel getAllModel(){

return model;

}

public ArrayList<AllModel> getRoom() throws Exception {

try {

RoomDL dlRoom = new RoomDL(this.model);

return dlRoom.getRoom();

}catch(Exception e) {

throw e;

}

}

}

**DATABASE LAYER**

DLBillings

package DatabaseLayer;

import Models.AllModel;

import Helper.DatabaseConnector;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import Frontend.Admin;

public class DLBillings {

private AllModel admin;

private DatabaseConnector db;

private Connection connection;

public DLBillings() throws Exception {

this.admin = new AllModel();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLBillings(AllModel admin) throws Exception {

this.admin = admin;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public AllModel getAdmin() {

return admin;

}

public void setAdmin(AllModel admin) {

this.admin = admin;

}

public AllModel save() throws Exception {

try {

// prepare for the data to be returned in case of insert

String generatedColumns[] = {"FullName","Address","Check-in-date","Chek-out-date"};

// create the statement

String query = "INSERT INTO admin(FullName, Check\_in\_date,Check\_out\_date,Address) VALUES(?,?,?,?)";

PreparedStatement statement = this.connection.prepareStatement(query, generatedColumns);

statement.setString(1, this.admin.getFullName());

statement.setString(2, this.admin.getAddress());

statement.setString(3, this.admin.getCheck\_in\_date());

statement.setString(4, this.admin.getCheck\_out\_date());

// execute the query

int noOfUpdate = statement.executeUpdate();

if(noOfUpdate>0) {

ResultSet rs = statement.getGeneratedKeys();

if(rs.next()) {

int id = rs.getInt(1);

String name = rs.getString(2);

this.admin.setBooking\_ID(id);

}

}

return this.admin;

}catch(Exception ex) {

throw ex;

}

}

}

DLBooking

package DatabaseLayer;

import Models.AllModel;

import Models.Booking;

import Models.Corporate;

import Helper.DatabaseConnector;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.JOptionPane;

import Frontend.Home;

public class DLBooking {

private Booking booking;

private DatabaseConnector db;

private Connection connection;

private AllModel allmodel;

public DLBooking() throws Exception {

this.booking = new Booking();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLBooking(Booking booking) throws Exception {

this.booking = booking;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLBooking(AllModel allmodel) throws Exception {

this.allmodel = allmodel;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public Booking getUser() {

return booking;

}

public void setUser(Booking booking) {

this.booking = booking;

}

public AllModel getAllmodel() {

return allmodel;

}

public void setAllmodel(AllModel allmodel) {

this.allmodel = allmodel;

}

public Booking save() throws Exception {

try {

// create the statement

String query = "INSERT INTO booking(FullName, Address, Check\_in\_date ,Check\_out\_date,Room\_Type,postal\_Code) VALUES(?,?,?,?,?,?)";

PreparedStatement statement = this.connection.prepareStatement(query);

statement.setString(1, this.booking.getFullName());

statement.setString(2, this.booking.getAddress());

statement.setString(3, this.booking.getCheck\_in\_date());

statement.setString(4, this.booking.getCheck\_out\_date());

statement.setObject(5, this.booking.getRoom\_Type());

statement.setString(6, this.booking.getPostal\_code());

// execute the query

if (statement.executeUpdate()>0) {

JOptionPane.showMessageDialog(null,"Booking request sent");

}

else {

JOptionPane.showMessageDialog(null,"Invalid information");

}

return this.booking;

}catch(Exception ex) {

throw ex;

}

}

public AllModel updateRoom() throws Exception {

PreparedStatement updateRoom;

ResultSet urs;

try {

String inserRoomNoQuery = "update booking b set b.RoomNo = ? where b.Booking\_ID = ?";

updateRoom = this.connection.prepareStatement(inserRoomNoQuery);

updateRoom.setInt(1, this.allmodel.getRoomNo());

updateRoom.setInt(2, this.allmodel.getBooking\_ID());

try {

updateRoom.executeUpdate();

JOptionPane.showMessageDialog(null, "Booking Accepted", "Complete", 2);

} catch (Exception ex) {

throw ex;

}

} catch (Exception ex) {

throw ex;

}

return allmodel;

}

}

DLBookingDetails

package DatabaseLayer;

import Helper.DatabaseConnector;

import Models.AllModel;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class DLBookingDetails {

private AllModel model;

private DatabaseConnector db;

private Connection connection;

public DLBookingDetails() throws Exception {

this.model = new AllModel();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLBookingDetails(AllModel model) throws Exception {

this.model = model;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public AllModel getModel() {

return model;

}

public void setUser(AllModel model) {

this.model = model;

}

public ArrayList<AllModel> getBooking() throws Exception {

try {

ArrayList<AllModel> model = new ArrayList<AllModel>();

String query = "SELECT FullName, Address, Check\_in\_date, Check\_out\_date, Room\_Type, Booking\_status, Booking\_ID FROM booking";

Statement statement = this.connection.createStatement();

ResultSet rs = statement.executeQuery(query);

while(rs.next()) {

AllModel allmodel = new AllModel();

allmodel.setFullName(rs.getString("FullName"));

allmodel.setAddress(rs.getString("Address"));

allmodel.setCheck\_in\_date(rs.getString("Check\_in\_date"));

allmodel.setCheck\_out\_date(rs.getString("Check\_out\_date"));

allmodel.setRoom\_Type(rs.getString("Room\_Type"));

allmodel.setBooking\_status(rs.getString("Booking\_status"));

allmodel.setBooking\_ID(rs.getInt("Booking\_ID"));

model.add(allmodel);

}

return model;

}

catch(Exception ex) {

throw ex;

}

}

}

DLBookingTable

package DatabaseLayer;

import Helper.DatabaseConnector;

import Models.AllModel;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class DLBookingTable {

private AllModel model;

private DatabaseConnector db;

private Connection connection;

public DLBookingTable() throws Exception {

this.model = new AllModel();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLBookingTable(AllModel model) throws Exception {

this.model = model;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public AllModel getModel() {

return model;

}

public void setUser(AllModel model) {

this.model = model;

}

public ArrayList<AllModel> getBooking() throws Exception {

try {

ArrayList<AllModel> model = new ArrayList<AllModel>();

String query = "SELECT FullName, Address, Check\_in\_date, Check\_out\_date, Room\_Type, Booking\_status, Booking\_ID FROM booking";

Statement statement = this.connection.createStatement();

ResultSet rs = statement.executeQuery(query);

while(rs.next()) {

AllModel allmodel = new AllModel();

allmodel.setFullName(rs.getString("FullName"));

allmodel.setAddress(rs.getString("Address"));

allmodel.setCheck\_in\_date(rs.getString("Check\_in\_date"));

allmodel.setCheck\_out\_date(rs.getString("Check\_out\_date"));

allmodel.setRoom\_Type(rs.getString("Room\_Type"));

allmodel.setBooking\_status(rs.getString("Booking\_status"));

allmodel.setBooking\_ID(rs.getInt("Booking\_ID"));

model.add(allmodel);

}

return model;

}

catch(Exception ex) {

throw ex;

}

}

}

**DLCorporate**

package DatabaseLayer;

import Models.Corporate;

import Models.Login;

import Helper.DatabaseConnector;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import Frontend.NewCorporate;

import Frontend.Home;

import Frontend.NewLogin;

public class DLCorporate{

private Corporate corporate;

private DatabaseConnector db;

private Connection connection;

public DLCorporate() throws Exception {

this.corporate = new Corporate();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLCorporate(Corporate corporate) throws Exception {

this.corporate = corporate;try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public Corporate getCorporateatt() {

return corporate;

}

public void setCorporateatt(Corporate corporate) {

this.corporate = corporate;

}

public Corporate save() throws Exception {

try {

String query = "INSERT INTO Corporate (Company\_Name, Contact, Email, Password, Confirm\_Password) VALUES(?,?,?,?,?)";

PreparedStatement statement = this.connection.prepareStatement(query);

statement.setString(1, this.corporate.getCompany\_Name());

statement.setString(2, this.corporate.getContact());

statement.setString(3, this.corporate.getEmail());

statement.setString(4, this.corporate.getPassword());

statement.setString(5, this.corporate.getConfirm\_Password());

// execute the query

try {

if(statement.executeUpdate() !=0) {

JOptionPane.showMessageDialog(null,"Your account has been successfully created");

NewLogin log = new NewLogin();

log.setVisible(true);

log.setLocationRelativeTo(null);

NewCorporate reg = new NewCorporate();

reg.dispose();

}

else {

JOptionPane.showMessageDialog(null,"Please check your details");

}

}

catch (Exception ex) {

throw ex;

}

return this.corporate;

} catch (Exception ex) {

throw ex;

}

}

public Corporate corporate() throws Exception {

PreparedStatement customers;

ResultSet cust;

String query = "SELECT \* FROM Corporate WHERE Email = ? AND Password = ? ";

try {

customers = this.connection.prepareStatement(query);

customers.setString(1, this.corporate.getEmail());

customers.setString(2, this.corporate.getPassword());

cust = customers.executeQuery();

//condition if the username and password match

if (cust.next())

{

//take user to user Login page

Home log = new Home();

log.setVisible(true);

log.setLocationRelativeTo(null);

}

else {

JOptionPane.showMessageDialog(null, "Invalid Username Password", "Error", 2);

}

}catch(Exception ex) {

throw ex;

}

return corporate;

}

}

DLNonCorporate

package DatabaseLayer;

import Models.AllModel;

import Models.Corporate;

import Models.Login;

import Models.NonCorporate;

import Helper.DatabaseConnector;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

import Frontend.NewNonCorporate;

import Frontend.Admin;

import Frontend.Home;

import Frontend.NewLogin;

public class DLNonCorporate {

private NonCorporate noncorporate;

private AllModel model;

private DatabaseConnector db;

private Connection connection;

public DLNonCorporate() throws Exception {

this.noncorporate = new NonCorporate();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLNonCorporate(NonCorporate noncorporate) throws Exception {

this.noncorporate = noncorporate;try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public DLNonCorporate(AllModel model) throws Exception {

this.model = model;try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public NonCorporate getNonCorporate() {

return noncorporate;

}

public void setNonCorporate( NonCorporate noncorporate) {

this.noncorporate = noncorporate;

}

public NonCorporate save() throws Exception {

try {

String query = "INSERT INTO non\_Corporate (First\_Name, Last\_Name, Address,Credit\_Card\_No, Email, Password, Confirm\_Password) VALUES(?,?,?,?,?,?,?)";

PreparedStatement statement = this.connection.prepareStatement(query);

statement.setString(1, this.noncorporate.getFirst\_Name());

statement.setString(2, this.noncorporate.getLast\_Name());

statement.setString(3, this.noncorporate.getAddress());

statement.setString(4, this.noncorporate.getCredit\_Card\_No());

statement.setString(5, this.noncorporate.getEmail());

statement.setString(6, this.noncorporate.getPassword());

statement.setString(7, this.noncorporate.getConfirm\_Password());

// execute the query

try {

if(statement.executeUpdate() !=0) {

JOptionPane.showMessageDialog(null,"Your account has been successfully created");

NewLogin log = new NewLogin();

log.setVisible(true);

log.setLocationRelativeTo(null);

NewNonCorporate reg = new NewNonCorporate();

reg.dispose();

}

else {

JOptionPane.showMessageDialog(null,"Please check your details");

}

}

catch (Exception ex) {

throw ex;

}

return this.noncorporate;

} catch (Exception ex) {

throw ex;

}

}

public AllModel model() throws Exception {

PreparedStatement customers;

PreparedStatement reception;

ResultSet cust;

ResultSet recep;

String query = "SELECT \* FROM non\_Corporate WHERE Email = ? AND Password = ? ";

String adminQuery = "SELECT \* FROM ADMIN WHERE Email = ? AND Password = ? AND UserType = 'admin'";

try {

customers = this.connection.prepareStatement(query);

customers.setString(1, this.model.getEmail());

customers.setString(2, this.model.getPassword());

cust = customers.executeQuery();

reception = this.connection.prepareStatement(adminQuery);

reception.setString(1, this.model.getEmail());

reception.setString(2, this.model.getPassword());

cust = customers.executeQuery();

recep = reception.executeQuery();

//condition if the username and password match

if (cust.next())

{

//take user to user Login page

Home log = new Home();

log.setVisible(true);

log.setLocationRelativeTo(null);

}

else if (recep.next())

{

//take user to user Login page

Admin admin = new Admin();

admin.setVisible(true);

admin.setLocationRelativeTo(null);

}

else {

JOptionPane.showMessageDialog(null, "Invalid email Password", "Error", 2);

}

}catch(Exception ex) {

throw ex;

}

return model;

}

}

RoomDL

package DatabaseLayer;

import Helper.DatabaseConnector;

import Models.AllModel;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class RoomDL {

private AllModel model;

private DatabaseConnector db;

private Connection connection;

public RoomDL() throws Exception {

this.model = new AllModel();

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public RoomDL(AllModel model) throws Exception {

this.model = model;

try {

this.db = DatabaseConnector.getInstance();

this.connection = this.db.getConnection();

} catch (Exception ex) {

throw ex;

}

}

public AllModel getModel() {

return model;

}

public void setUser(AllModel model) {

this.model = model;

}

public ArrayList<AllModel> getRoom() throws Exception {

try {

ArrayList<AllModel> model = new ArrayList<AllModel>();

String query = "SELECT \* FROM room";

Statement statement = this.connection.createStatement();

ResultSet rs = statement.executeQuery(query);

while(rs.next()) {

AllModel allmodel = new AllModel();

allmodel.setRoom\_No(rs.getString("room\_number"));

allmodel.setRoom\_Type(rs.getString("room\_type"));

allmodel.setRoom\_Rate(rs.getString("room\_rate"));

allmodel.setRoom\_Status(rs.getString("room\_Status"));

model.add(allmodel);

}

return model;

}

catch(Exception ex) {

throw ex;

}

}

}

FRONTENED

AboutUs

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import java.awt.Toolkit;

public class AboutUs extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

AboutUs frame = new AboutUs();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public AboutUs() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\aboutt.jpg"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(159, 86, 1033, 554);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblSpendQuality = new JLabel();

lblSpendQuality.setText("INNOVATIVE BUSINESS");

lblSpendQuality.setForeground(Color.BLACK);

lblSpendQuality.setFont(new Font("Times New Roman", Font.PLAIN, 16));

lblSpendQuality.setBackground(Color.BLACK);

lblSpendQuality.setBounds(519, 157, 271, 40);

desktopPane.add(lblSpendQuality);

JLabel lblHolidaysWithUs = new JLabel();

lblHolidaysWithUs.setText("About Us!");

lblHolidaysWithUs.setForeground(Color.BLACK);

lblHolidaysWithUs.setFont(new Font("Times New Roman", Font.BOLD, 50));

lblHolidaysWithUs.setBackground(Color.BLACK);

lblHolidaysWithUs.setBounds(504, 193, 326, 40);

desktopPane.add(lblHolidaysWithUs);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JLabel lblEnojyYourHolidays = new JLabel();

lblEnojyYourHolidays.setText("Luton Hotel is a large independently run hotel located in the town of Luton. ");

lblEnojyYourHolidays.setForeground(Color.BLACK);

lblEnojyYourHolidays.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblEnojyYourHolidays.setBackground(Color.BLACK);

lblEnojyYourHolidays.setBounds(510, 267, 492, 32);

desktopPane.add(lblEnojyYourHolidays);

JLabel lblBestExperienceEver = new JLabel();

lblBestExperienceEver.setText("It offers a variety of accommodation all of which offer a suite facilities.");

lblBestExperienceEver.setForeground(Color.BLACK);

lblBestExperienceEver.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblBestExperienceEver.setBackground(Color.BLACK);

lblBestExperienceEver.setBounds(510, 299, 465, 32);

desktopPane.add(lblBestExperienceEver);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Home signIn = new Home();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(870, 478, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\aboutus.jpg"));

lblNewLabel.setBounds(24, 136, 465, 315);

desktopPane.add(lblNewLabel);

JLabel lblBestExperienceEver\_1 = new JLabel();

lblBestExperienceEver\_1.setText("Best Experience Ever.\r\n");

lblBestExperienceEver\_1.setForeground(Color.BLACK);

lblBestExperienceEver\_1.setFont(new Font("Times New Roman", Font.BOLD, 18));

lblBestExperienceEver\_1.setBackground(Color.BLACK);

lblBestExperienceEver\_1.setBounds(643, 371, 176, 32);

desktopPane.add(lblBestExperienceEver\_1);

}

}

Admin

package Frontend;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.ImageIcon;

import javax.swing.JDesktopPane;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.awt.event.ActionEvent;

import javax.swing.border.LineBorder;

import javax.swing.table.DefaultTableModel;

import BusinessLayer.BLBillings;

import BusinessLayer.BLBooking;

import BusinessLayer.BLBookingTable;

import BusinessLayer.BLCorporate;

import BusinessLayer.RoomBL;

import Helper.InputException;

import Models.AllModel;

import Models.Booking;

import Models.Corporate;

import Models.Login;

import javax.swing.JTable;

import javax.swing.JScrollPane;

import java.awt.Toolkit;

import javax.swing.border.CompoundBorder;

import javax.swing.border.BevelBorder;

import javax.swing.JInternalFrame;

import javax.swing.JTextField;

public class Admin extends JFrame {

private JPanel contentPane;

private DefaultTableModel model;

private DefaultTableModel bookingmodel;

private JTable bookingTable;

private JTable roomTable;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

private JTextField textField\_3;

private Object billing;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

Admin rep = new Admin();

rep.setVisible(true);

}

/\*\*

\* Create the frame.

\*/

public Admin() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\icon.png"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1800, 900);

contentPane = new JPanel();

contentPane.setBackground(Color.LIGHT\_GRAY);

contentPane.setForeground(new Color(190, 190, 190));

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel Head = new JLabel();

Head.setText("LUTON HOTEL ADMINISTRATION ");

Head.setForeground(new Color(64, 0, 64));

Head.setFont(new Font("Times New Roman", Font.BOLD, 40));

Head.setBackground(new Color(255, 0, 0));

Head.setBounds(334, 34, 700, 55);

contentPane.add(Head);

JDesktopPane repField = new JDesktopPane();

repField.setBorder(new LineBorder(Color.BLACK, 2, true));

repField.setBackground(Color.ORANGE);

repField.setForeground(new Color(153, 204, 255));

repField.setBounds(33, 100, 796, 512);

contentPane.add(repField);

JButton addRoomButton = new JButton("Add Room");

addRoomButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(bookingTable.getSelectionModel().isSelectionEmpty()) {

return;

}

int row = bookingTable.getSelectedRow();

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

int bookingId = Integer.parseInt(model.getValueAt(row, 6).toString());

String roomNo = JOptionPane.showInputDialog("Enter the room no.");

AllModel allmodel = new AllModel();

allmodel.setRoomNo(Integer.parseInt(roomNo));

allmodel.setBooking\_ID(bookingId);

BLBooking blBooking = new BLBooking();

try {

blBooking.assignRoom(allmodel);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

addRoomButton.setForeground(Color.BLACK);

addRoomButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

addRoomButton.setBackground(Color.CYAN);

addRoomButton.setBounds(481, 38, 138, 30);

repField.add(addRoomButton);

JButton roomButton = new JButton("Room");

roomButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

loadAllRoom();

}

});

roomButton.setForeground(Color.BLACK);

roomButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

roomButton.setBackground(Color.CYAN);

roomButton.setBounds(216, 38, 107, 30);

repField.add(roomButton);

JButton bookingButton = new JButton("Booking");

bookingButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

loadAllBooking();

}

});

bookingButton.setForeground(Color.BLACK);

bookingButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

bookingButton.setBackground(Color.CYAN);

bookingButton.setBounds(333, 38, 138, 30);

repField.add(bookingButton);

JButton logOutButton = new JButton("Log Out");

logOutButton.setBounds(1198, 598, 125, 39);

contentPane.add(logOutButton);

logOutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewLogin login = new NewLogin();

login.setVisible(true);

dispose();

}

});

repField.setLayer(logOutButton, 2);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(57, 100, 689, 187);

repField.add(desktopPane);

desktopPane.setLayout(null);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(10, 11, 669, 165);

desktopPane.add(scrollPane);

bookingTable = new JTable();

scrollPane.setViewportView(bookingTable);

JDesktopPane desktopPane\_1 = new JDesktopPane();

desktopPane\_1.setBounds(57, 298, 689, 179);

repField.add(desktopPane\_1);

JScrollPane scrollPane\_1 = new JScrollPane();

scrollPane\_1.setBounds(10, 11, 669, 165);

desktopPane\_1.add(scrollPane\_1);

roomTable = new JTable();

scrollPane\_1.setViewportView(roomTable);

logOutButton.setForeground(Color.BLACK);

logOutButton.setFont(new Font("Times New Roman", Font.BOLD, 22));

logOutButton.setBackground(Color.GREEN);

JDesktopPane desktopPane\_2 = new JDesktopPane();

desktopPane\_2.setBounds(884, 145, 439, 369);

contentPane.add(desktopPane\_2);

JButton btnCheckout = new JButton("Check-Out");

btnCheckout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

userbillings();

}

});

btnCheckout.setForeground(Color.BLACK);

btnCheckout.setFont(new Font("Times New Roman", Font.BOLD, 20));

btnCheckout.setBackground(Color.CYAN);

btnCheckout.setBounds(293, 314, 136, 33);

desktopPane\_2.add(btnCheckout);

JLabel lblFullnmae = new JLabel();

lblFullnmae.setText("FullName");

lblFullnmae.setForeground(Color.BLUE);

lblFullnmae.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblFullnmae.setBackground(Color.BLACK);

lblFullnmae.setBounds(31, 43, 141, 40);

desktopPane\_2.add(lblFullnmae);

JLabel lblCheckindate = new JLabel();

lblCheckindate.setText("Check-In-Date");

lblCheckindate.setForeground(Color.BLUE);

lblCheckindate.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblCheckindate.setBackground(Color.BLACK);

lblCheckindate.setBounds(31, 107, 141, 40);

desktopPane\_2.add(lblCheckindate);

JLabel lblCheckoutdate = new JLabel();

lblCheckoutdate.setText("Check-Out-Date");

lblCheckoutdate.setForeground(Color.BLUE);

lblCheckoutdate.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblCheckoutdate.setBackground(Color.BLACK);

lblCheckoutdate.setBounds(31, 170, 154, 40);

desktopPane\_2.add(lblCheckoutdate);

JLabel lblBookingId = new JLabel();

lblBookingId.setText("Address");

lblBookingId.setForeground(Color.BLUE);

lblBookingId.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblBookingId.setBackground(Color.BLACK);

lblBookingId.setBounds(31, 236, 141, 40);

desktopPane\_2.add(lblBookingId);

textField = new JTextField();

textField.setColumns(10);

textField.setBackground(Color.LIGHT\_GRAY);

textField.setBounds(232, 44, 197, 43);

desktopPane\_2.add(textField);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_1.setBackground(Color.LIGHT\_GRAY);

textField\_1.setBounds(232, 104, 197, 43);

desktopPane\_2.add(textField\_1);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_2.setBackground(Color.LIGHT\_GRAY);

textField\_2.setBounds(232, 167, 197, 43);

desktopPane\_2.add(textField\_2);

textField\_3 = new JTextField();

textField\_3.setColumns(10);

textField\_3.setBackground(Color.LIGHT\_GRAY);

textField\_3.setBounds(232, 233, 197, 43);

desktopPane\_2.add(textField\_3);

JLabel lblBillings = new JLabel();

lblBillings.setText("BILLINGS");

lblBillings.setForeground(Color.BLUE);

lblBillings.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblBillings.setBackground(Color.BLACK);

lblBillings.setBounds(1049, 100, 141, 40);

contentPane.add(lblBillings);

model = new DefaultTableModel();

Object[] columnsName = new Object[4];

columnsName[0] = "ROOM NUMBER";

columnsName[1] = "ROOM TYPE";

columnsName[2] = "ROOM RATE";

columnsName[3] = "ROOM STATUS";

model.setColumnCount(0);

model.setColumnIdentifiers(columnsName);

}

private void loadAllRoom() {

try {

RoomBL roomBL = new RoomBL();

ArrayList<AllModel> model = roomBL.getRoom();

setTableData(model);

} catch (Exception e1) {

JOptionPane.showMessageDialog(null, e1.getMessage());

}

}

private void setTableData(ArrayList<AllModel> models) {

// Create the object array from arraylist and add to the table row

Object[] rowData = new Object[4];

// set the number of rows in table model to zero

model.setRowCount(0);

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

rowData[0] = models.get(i).getRoom\_No();

rowData[1] = models.get(i).getRoom\_Type();

rowData[2] = models.get(i).getRoom\_Rate();

rowData[3] = models.get(i).getRoom\_Status();

model.addRow(rowData);

}

roomTable.setModel(model);

model = new DefaultTableModel();

Object[] columnsName = new Object[7];

columnsName[0] = "FULL NAME";

columnsName[1] = "ADDRESS";

columnsName[2] = "CHECK IN DATE";

columnsName[3] = "CHECK OUT DATE";

columnsName[4] = "ROOM TYPE";

columnsName[5] = "BOOKING STATUS";

columnsName[6] = "BOOKING ID";

model.setColumnCount(0);

model.setColumnIdentifiers(columnsName);

}

private void loadAllBooking() {

try {

BLBookingTable blbookingtable = new BLBookingTable();

ArrayList<AllModel> model = BLBookingTable.Booking();

setTableData1(model);

} catch (Exception e1) {

JOptionPane.showMessageDialog(null, e1.getMessage());

}

}

private void setTableData1(ArrayList<AllModel> models) {

// Create the object array from arraylist and add to the table row

Object[] rowData = new Object[7];

// set the number of rows in table model to zero

model.setRowCount(0);

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

rowData[0] = models.get(i).getFullName();

rowData[1] = models.get(i).getAddress();

rowData[2] = models.get(i).getCheck\_in\_date();

rowData[3] = models.get(i).getCheck\_out\_date();

rowData[4] = models.get(i).getRoom\_Type();

rowData[5] = models.get(i).getBooking\_status();

rowData[6] = models.get(i).getBooking\_ID();

model.addRow(rowData);

}

bookingTable.setModel(model);

}

private void userbillings() {

// On click of the log In button

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

AllModel adminn = new AllModel();

adminn.setFullName(textField.getText());

adminn.setCheck\_in\_date(textField\_1.getText());

adminn.setCheck\_out\_date(textField\_2.getText());

adminn.setAddress(textField\_3.getText());

BLBillings<AllModel> billingsBL = new BLBillings<AllModel>();

billingsBL.validateUser(adminn);

billingsBL.billinginf(adminn);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

public Object getBilling() {

return billing;

}

public void setBilling(Object billing) {

this.billing = billing;

}

}

BookingDetails

package Frontend;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.ImageIcon;

import javax.swing.JDesktopPane;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.awt.event.ActionEvent;

import javax.swing.border.LineBorder;

import javax.swing.table.DefaultTableModel;

import BusinessLayer.BLBookingDetails;

import BusinessLayer.RoomBL;

import Models.AllModel;

import Models.Booking;

import Models.Login;

import javax.swing.JTable;

import javax.swing.JScrollPane;

import java.awt.Toolkit;

import javax.swing.border.CompoundBorder;

import javax.swing.border.BevelBorder;

public class BookingDetails extends JFrame {

private JPanel contentPane;

private DefaultTableModel demo;

private DefaultTableModel model;

private JTable table;

private JTable bookingTable;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

BookingDetails rep = new BookingDetails();

rep.setVisible(true);

}

/\*\*

\* Create the frame.

\*/

public BookingDetails() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\icon.png"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1800, 900);

contentPane = new JPanel();

contentPane.setBackground(Color.GRAY);

contentPane.setForeground(new Color(190, 190, 190));

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel Head = new JLabel();

Head.setText("BOOKING DETAILS");

Head.setForeground(Color.RED);

Head.setFont(new Font("Times New Roman", Font.BOLD, 40));

Head.setBackground(new Color(255, 0, 0));

Head.setBounds(523, 23, 412, 55);

contentPane.add(Head);

JDesktopPane repField = new JDesktopPane();

repField.setBorder(new LineBorder(Color.BLACK, 2, true));

repField.setBackground(Color.ORANGE);

repField.setForeground(new Color(153, 204, 255));

repField.setBounds(281, 89, 845, 573);

contentPane.add(repField);

JButton addRoomButton = new JButton("Cancell Booking ");

addRoomButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms room = new Rooms();

contentPane.add(room);

room.setVisible(true);

}

});

addRoomButton.setForeground(Color.BLACK);

addRoomButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

addRoomButton.setBackground(Color.CYAN);

addRoomButton.setBounds(1013, 514, 187, 30);

repField.add(addRoomButton);

JButton roomButton = new JButton("View Booking ");

roomButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

loadAllBookings();

}

}

);

roomButton.setForeground(Color.BLACK);

roomButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

roomButton.setBackground(Color.CYAN);

roomButton.setBounds(370, 36, 167, 30);

repField.add(roomButton);

JButton bookingButton = new JButton("Edit Bookings");

bookingButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

bookingButton.setForeground(Color.BLACK);

bookingButton.setFont(new Font("Times New Roman", Font.BOLD, 20));

bookingButton.setBackground(Color.CYAN);

bookingButton.setBounds(359, 305, 167, 30);

repField.add(bookingButton);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(32, 102, 789, 178);

repField.add(scrollPane);

bookingTable = new JTable();

scrollPane.setViewportView(bookingTable);

JDesktopPane desktopPane = new JDesktopPane();

scrollPane.setColumnHeaderView(desktopPane);

desktopPane.setLayout(null);

JScrollPane scrollPane\_1 = new JScrollPane();

scrollPane\_1.setBounds(10, 11, 669, 165);

desktopPane.add(scrollPane\_1);

model = new DefaultTableModel();

Object[] columnsName = new Object[7];

columnsName[0] = "FULL NAME";

columnsName[1] = "ADDRESS";

columnsName[2] = "CHECK IN DATE";

columnsName[3] = "CHECK OUT DATE";

columnsName[4] = "ROOM TYPE";

columnsName[5] = "BOOKING STATUS";

columnsName[6] = "BOOKING ID";

model.setColumnCount(0);

model.setColumnIdentifiers(columnsName);

}

private void loadAllBookings() {

try {

BLBookingDetails blbookingdetails = new BLBookingDetails();

ArrayList<AllModel> model = blbookingdetails.Booking();

setTableData(model);

} catch (Exception e1) {

JOptionPane.showMessageDialog(null, e1.getMessage());

}

}

private void setTableData(ArrayList<AllModel> models) {

// Create the object array from arraylist and add to the table row

Object[] rowData = new Object[7];

// set the number of rows in table model to zero

model.setRowCount(0);

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

rowData[0] = models.get(i).getFullName();

rowData[1] = models.get(i).getAddress();

rowData[2] = models.get(i).getCheck\_in\_date();

rowData[3] = models.get(i).getCheck\_out\_date();

rowData[4] = models.get(i).getRoom\_Type();

rowData[5] = models.get(i).getBooking\_status();

rowData[6] = models.get(i).getBooking\_ID();

model.addRow(rowData);

}

bookingTable.setModel(model);

}

}

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

public class ContactUs extends JFrame {

private JPanel contentPane;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ContactUs frame = new ContactUs();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public ContactUs() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(159, 86, 1033, 554);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblHolidaysWithUs = new JLabel();

lblHolidaysWithUs.setText("Contact Us!");

lblHolidaysWithUs.setForeground(Color.BLACK);

lblHolidaysWithUs.setFont(new Font("Times New Roman", Font.BOLD, 50));

lblHolidaysWithUs.setBackground(Color.BLACK);

lblHolidaysWithUs.setBounds(480, 148, 326, 40);

desktopPane.add(lblHolidaysWithUs);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Home signIn = new Home();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(874, 481, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\contactus.jpg"));

lblNewLabel.setBounds(66, 109, 350, 400);

desktopPane.add(lblNewLabel);

JLabel lblBestExperienceEver\_1 = new JLabel();

lblBestExperienceEver\_1.setText("Best Experience Ever\r\n");

lblBestExperienceEver\_1.setForeground(Color.BLACK);

lblBestExperienceEver\_1.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblBestExperienceEver\_1.setBackground(Color.BLACK);

lblBestExperienceEver\_1.setBounds(53, 43, 163, 32);

desktopPane.add(lblBestExperienceEver\_1);

textField = new JTextField();

textField.setColumns(10);

textField.setBackground(Color.WHITE);

textField.setBounds(667, 217, 326, 43);

desktopPane.add(textField);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_1.setBackground(Color.WHITE);

textField\_1.setBounds(667, 270, 326, 43);

desktopPane.add(textField\_1);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_2.setBackground(Color.WHITE);

textField\_2.setBounds(667, 324, 326, 59);

desktopPane.add(textField\_2);

JLabel label\_2\_1 = new JLabel();

label\_2\_1.setText("Enter Your Name");

label\_2\_1.setForeground(Color.BLUE);

label\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 18));

label\_2\_1.setBackground(Color.BLACK);

label\_2\_1.setBounds(438, 220, 194, 40);

desktopPane.add(label\_2\_1);

JLabel label\_2\_1\_1 = new JLabel();

label\_2\_1\_1.setText("Enter A Valid Email Address");

label\_2\_1\_1.setForeground(Color.BLUE);

label\_2\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 16));

label\_2\_1\_1.setBackground(Color.BLACK);

label\_2\_1\_1.setBounds(438, 270, 212, 40);

desktopPane.add(label\_2\_1\_1);

JLabel label\_2\_1\_2 = new JLabel();

label\_2\_1\_2.setText("Type Your Message Here");

label\_2\_1\_2.setForeground(Color.BLUE);

label\_2\_1\_2.setFont(new Font("Times New Roman", Font.BOLD, 18));

label\_2\_1\_2.setBackground(Color.BLACK);

label\_2\_1\_2.setBounds(438, 341, 206, 40);

desktopPane.add(label\_2\_1\_2);

JButton btnNewButton\_1\_1 = new JButton("Submit");

btnNewButton\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 22));

btnNewButton\_1\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_1.setBounds(625, 415, 113, 40);

desktopPane.add(btnNewButton\_1\_1);

}

}

ContactUs

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JTextField;

public class ContactUs extends JFrame {

private JPanel contentPane;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

ContactUs frame = new ContactUs();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public ContactUs() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(159, 86, 1033, 554);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblHolidaysWithUs = new JLabel();

lblHolidaysWithUs.setText("Contact Us!");

lblHolidaysWithUs.setForeground(Color.BLACK);

lblHolidaysWithUs.setFont(new Font("Times New Roman", Font.BOLD, 50));

lblHolidaysWithUs.setBackground(Color.BLACK);

lblHolidaysWithUs.setBounds(480, 148, 326, 40);

desktopPane.add(lblHolidaysWithUs);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Home signIn = new Home();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(874, 481, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\contactus.jpg"));

lblNewLabel.setBounds(66, 109, 350, 400);

desktopPane.add(lblNewLabel);

JLabel lblBestExperienceEver\_1 = new JLabel();

lblBestExperienceEver\_1.setText("Best Experience Ever\r\n");

lblBestExperienceEver\_1.setForeground(Color.BLACK);

lblBestExperienceEver\_1.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblBestExperienceEver\_1.setBackground(Color.BLACK);

lblBestExperienceEver\_1.setBounds(53, 43, 163, 32);

desktopPane.add(lblBestExperienceEver\_1);

textField = new JTextField();

textField.setColumns(10);

textField.setBackground(Color.WHITE);

textField.setBounds(667, 217, 326, 43);

desktopPane.add(textField);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_1.setBackground(Color.WHITE);

textField\_1.setBounds(667, 270, 326, 43);

desktopPane.add(textField\_1);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_2.setBackground(Color.WHITE);

textField\_2.setBounds(667, 324, 326, 59);

desktopPane.add(textField\_2);

JLabel label\_2\_1 = new JLabel();

label\_2\_1.setText("Enter Your Name");

label\_2\_1.setForeground(Color.BLUE);

label\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 18));

label\_2\_1.setBackground(Color.BLACK);

label\_2\_1.setBounds(438, 220, 194, 40);

desktopPane.add(label\_2\_1);

JLabel label\_2\_1\_1 = new JLabel();

label\_2\_1\_1.setText("Enter A Valid Email Address");

label\_2\_1\_1.setForeground(Color.BLUE);

label\_2\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 16));

label\_2\_1\_1.setBackground(Color.BLACK);

label\_2\_1\_1.setBounds(438, 270, 212, 40);

desktopPane.add(label\_2\_1\_1);

JLabel label\_2\_1\_2 = new JLabel();

label\_2\_1\_2.setText("Type Your Message Here");

label\_2\_1\_2.setForeground(Color.BLUE);

label\_2\_1\_2.setFont(new Font("Times New Roman", Font.BOLD, 18));

label\_2\_1\_2.setBackground(Color.BLACK);

label\_2\_1\_2.setBounds(438, 341, 206, 40);

desktopPane.add(label\_2\_1\_2);

JButton btnNewButton\_1\_1 = new JButton("Submit");

btnNewButton\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 22));

btnNewButton\_1\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_1.setBounds(625, 415, 113, 40);

desktopPane.add(btnNewButton\_1\_1);

}

}

DoubleRoom

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

public class DoubleRoom extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

DoubleRoom frame = new DoubleRoom();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public DoubleRoom() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(168, 52, 1033, 611);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_1 = new JButton("Double Rooms");

btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

btnNewButton\_1\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2\_1.setBounds(395, 65, 205, 40);

desktopPane.add(btnNewButton\_1\_2\_1);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms signIn = new Rooms();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(891, 549, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblWithSofaTelevision = new JLabel();

lblWithSofaTelevision.setText("Room facilities: Furnished balcony overlooking the pool or the street, telephone, flat screen TV with satellite channels");

lblWithSofaTelevision.setForeground(Color.BLACK);

lblWithSofaTelevision.setFont(new Font("Times New Roman", Font.BOLD, 15));

lblWithSofaTelevision.setBackground(Color.BLACK);

lblWithSofaTelevision.setBounds(121, 398, 777, 23);

desktopPane.add(lblWithSofaTelevision);

JLabel lblCostPer = new JLabel();

lblCostPer.setText("COST: 350$ PER DAY ");

lblCostPer.setForeground(Color.BLUE);

lblCostPer.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblCostPer.setBackground(new Color(0, 0, 255));

lblCostPer.setBounds(382, 479, 242, 40);

desktopPane.add(lblCostPer);

JLabel lblRoomFacilitiesFurnished = new JLabel();

lblRoomFacilitiesFurnished.setText(" hot/cold air conditioned, desk, tiled floor, wardrobe with safe for rent, mini fridge. Hairdryer, free toiletries");

lblRoomFacilitiesFurnished.setForeground(Color.BLACK);

lblRoomFacilitiesFurnished.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblRoomFacilitiesFurnished.setBackground(Color.BLACK);

lblRoomFacilitiesFurnished.setBounds(117, 420, 712, 23);

desktopPane.add(lblRoomFacilitiesFurnished);

JLabel lblGelShampooDispenser = new JLabel();

lblGelShampooDispenser.setText(" gel /shampoo dispenser and a bath or shower.");

lblGelShampooDispenser.setForeground(Color.BLACK);

lblGelShampooDispenser.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblGelShampooDispenser.setBackground(Color.BLACK);

lblGelShampooDispenser.setBounds(117, 437, 742, 31);

desktopPane.add(lblGelShampooDispenser);

JLabel lblNewLabel\_1\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\double2.jpg"));

lblNewLabel\_1\_1\_1.setBounds(528, 136, 393, 254);

desktopPane.add(lblNewLabel\_1\_1\_1);

JLabel lblNewLabel\_1\_1\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\double1.jpg"));

lblNewLabel\_1\_1\_1\_1.setBounds(86, 136, 393, 254);

desktopPane.add(lblNewLabel\_1\_1\_1\_1);

}

}

Home

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JDesktopPane;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.ImageIcon;

import java.awt.Toolkit;

public class Home extends JFrame {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Home frame = new Home();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Home() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\homee.jpg"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(159, 86, 1033, 554);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblSpendQuality = new JLabel();

lblSpendQuality.setText("Spend Quality");

lblSpendQuality.setForeground(Color.BLACK);

lblSpendQuality.setFont(new Font("Times New Roman", Font.BOLD, 40));

lblSpendQuality.setBackground(Color.BLACK);

lblSpendQuality.setBounds(55, 176, 271, 40);

desktopPane.add(lblSpendQuality);

JLabel lblHolidaysWithUs = new JLabel();

lblHolidaysWithUs.setText("Holidays With Us!");

lblHolidaysWithUs.setForeground(Color.BLACK);

lblHolidaysWithUs.setFont(new Font("Times New Roman", Font.BOLD, 40));

lblHolidaysWithUs.setBackground(Color.BLACK);

lblHolidaysWithUs.setBounds(55, 223, 326, 40);

desktopPane.add(lblHolidaysWithUs);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(863, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JButton btnNewButton\_1\_2 = new JButton("Rooms");

btnNewButton\_1\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms signIn = new Rooms();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2.setBounds(394, 11, 119, 40);

desktopPane.add(btnNewButton\_1\_2);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JLabel lblEnojyYourHolidays = new JLabel();

lblEnojyYourHolidays.setText("Enojy Your Holidays In Luton Hotel. Have A Good Time With Your Loved Ones.\r\n");

lblEnojyYourHolidays.setForeground(Color.BLACK);

lblEnojyYourHolidays.setFont(new Font("Times New Roman", Font.BOLD, 13));

lblEnojyYourHolidays.setBackground(Color.BLACK);

lblEnojyYourHolidays.setBounds(55, 271, 495, 32);

desktopPane.add(lblEnojyYourHolidays);

JButton btnNewButton\_1\_1 = new JButton("Best Experience Ever!");

btnNewButton\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 22));

btnNewButton\_1\_1.setBackground(Color.RED);

btnNewButton\_1\_1.setBounds(55, 372, 259, 40);

desktopPane.add(btnNewButton\_1\_1);

JLabel lblBestExperienceEver = new JLabel();

lblBestExperienceEver.setText("Best Experience Ever.\r\n");

lblBestExperienceEver.setForeground(Color.BLACK);

lblBestExperienceEver.setFont(new Font("Times New Roman", Font.BOLD, 16));

lblBestExperienceEver.setBackground(Color.BLACK);

lblBestExperienceEver.setBounds(55, 305, 154, 32);

desktopPane.add(lblBestExperienceEver);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\homepic.jpeg"));

lblNewLabel.setBounds(517, 95, 422, 350);

desktopPane.add(lblNewLabel);

JButton btnNewButton\_1\_2\_1 = new JButton("About Us");

btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

AboutUs signIn = new AboutUs();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2\_1.setBounds(517, 11, 154, 40);

desktopPane.add(btnNewButton\_1\_2\_1);

JButton btnNewButton\_1\_2\_2 = new JButton("Logout");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewLogin signIn = new NewLogin();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(898, 484, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JButton btnNewButton\_1\_2\_1\_1 = new JButton("Contact Us");

btnNewButton\_1\_2\_1\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

ContactUs Contact = new ContactUs();

Contact.setVisible(true);

}

});

btnNewButton\_1\_2\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1\_1.setBackground(Color.GRAY);

btnNewButton\_1\_2\_1\_1.setBounds(674, 11, 171, 40);

desktopPane.add(btnNewButton\_1\_2\_1\_1);

}

}

NewBookings

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

import BusinessLayer.BLBooking;

import BusinessLayer.BLCorporate;

import Helper.InputException;

import Models.Booking;

import Models.Corporate;

import javax.swing.JScrollPane;

import javax.swing.JComboBox;

import java.awt.Toolkit;

public class NewBooking extends JFrame {

private JPanel contentPane;

private JTextField tfpcode;

private JTextField tfcodate;

private JTextField tffname;

private JTextField tfadd;

private JTextField tfciddate;

private JComboBox combobox12;/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

NewBooking frame = new NewBooking();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public NewBooking() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\booook.jpg"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(269, 33, 839, 594);

desktopPane.setBackground(Color.ORANGE);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblLoginToLuton =new JLabel();

lblLoginToLuton.setBounds(296, 49, 283, 55);

lblLoginToLuton.setForeground(Color.BLACK);

lblLoginToLuton.setBackground(Color.BLACK);

lblLoginToLuton.setText("MAKE A BOOKING");

lblLoginToLuton.setFont(new Font("Times New Roman", Font.BOLD, 30));

desktopPane.add(lblLoginToLuton);

JButton btnNewButton\_1 = new JButton("Request Booking");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

saveBookinginf();

}

});

btnNewButton\_1.setBounds(588, 514, 186, 43);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 20));

desktopPane.add(btnNewButton\_1);

JLabel lblCheckindate = new JLabel();

lblCheckindate.setText("Check-In-Date");

lblCheckindate.setForeground(Color.BLUE);

lblCheckindate.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblCheckindate.setBackground(Color.BLACK);

lblCheckindate.setBounds(45, 243, 141, 40);

desktopPane.add(lblCheckindate);

tfciddate = new JTextField();

tfciddate.setColumns(10);

tfciddate.setBackground(Color.WHITE);

tfciddate.setBounds(196, 244, 197, 43);

desktopPane.add(tfciddate);

JLabel lblCheckoutdate = new JLabel();

lblCheckoutdate.setText("Check-Out-Date");

lblCheckoutdate.setForeground(Color.BLUE);

lblCheckoutdate.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblCheckoutdate.setBackground(Color.BLACK);

lblCheckoutdate.setBounds(414, 247, 162, 40);

desktopPane.add(lblCheckoutdate);

JButton btnNewButton\_1\_1 = new JButton("Back\r\n");

btnNewButton\_1\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms signUp = new Rooms();

signUp.setVisible(true);

}

});

btnNewButton\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 20));

btnNewButton\_1\_1.setBounds(57, 514, 119, 43);

desktopPane.add(btnNewButton\_1\_1);

JLabel lblRoomType = new JLabel();

lblRoomType.setText("Room Type\r\n");

lblRoomType.setForeground(Color.BLUE);

lblRoomType.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblRoomType.setBackground(Color.BLACK);

lblRoomType.setBounds(45, 316, 119, 40);

desktopPane.add(lblRoomType);

tfpcode = new JTextField();

tfpcode.setColumns(10);

tfpcode.setBackground(Color.WHITE);

tfpcode.setBounds(577, 317, 197, 43);

desktopPane.add(tfpcode);

tfcodate = new JTextField();

tfcodate.setColumns(10);

tfcodate.setBackground(Color.WHITE);

tfcodate.setBounds(577, 244, 197, 43);

desktopPane.add(tfcodate);

JLabel lblPostalCode = new JLabel();

lblPostalCode.setText("Postal Code");

lblPostalCode.setForeground(Color.BLUE);

lblPostalCode.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblPostalCode.setBackground(Color.BLACK);

lblPostalCode.setBounds(417, 316, 119, 40);

desktopPane.add(lblPostalCode);

String [] room = {"Single Room", "Double Room", "Twin Room"};

combobox12 = new JComboBox(room);

combobox12.setBounds(196, 317, 197, 43);

desktopPane.add(combobox12);

tffname = new JTextField();

tffname.setColumns(10);

tffname.setBackground(Color.WHITE);

tffname.setBounds(196, 167, 197, 43);

desktopPane.add(tffname);

tfadd = new JTextField();

tfadd.setColumns(10);

tfadd.setBackground(Color.WHITE);

tfadd.setBounds(577, 167, 197, 43);

desktopPane.add(tfadd);

JLabel lblFullnmae = new JLabel();

lblFullnmae.setText("FullName");

lblFullnmae.setForeground(Color.BLUE);

lblFullnmae.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblFullnmae.setBackground(Color.BLACK);

lblFullnmae.setBounds(45, 167, 141, 40);

desktopPane.add(lblFullnmae);

JLabel lblAddress = new JLabel();

lblAddress.setText("Address");

lblAddress.setForeground(Color.BLUE);

lblAddress.setFont(new Font("Times New Roman", Font.BOLD, 20));

lblAddress.setBackground(Color.BLACK);

lblAddress.setBounds(414, 170, 141, 40);

desktopPane.add(lblAddress);

JPanel panel = new JPanel();

panel.setBounds(346, 239, -260, -41);

contentPane.add(panel);

}

private void saveBookinginf() {

// On click of the log In button

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

Booking booking = new Booking();

booking.setFullName(tffname.getText());

booking.setAddress(tfadd.getText());

booking.setCheck\_in\_date(tfciddate.getText());

booking.setCheck\_out\_date(tfcodate.getText());

booking.setRoom\_Type(combobox12.getSelectedItem().toString());

booking.setPostal\_code(tfpcode.getText());

BLBooking bookingBL = new BLBooking();

bookingBL.validateBooking(booking);

bookingBL.bookinginf(booking);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

NewCoporate

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

import BusinessLayer.BLCorporate;

import Helper.InputException;

import Models.Corporate;

import Models.Login;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class NewCorporate extends JFrame {

private JTextField tfcname;

private JTextField tfccontact;

private JTextField tfcemail;

private JTextField tfcpass;

private JTextField tfcconfirmpass;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

NewCorporate frame = new NewCorporate();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public NewCorporate() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(279, 44, 849, 627);

desktopPane.setBackground(Color.GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

tfcname = new JTextField();

tfcname.setBackground(Color.WHITE);

tfcname.setBounds(310, 138, 410, 43);

desktopPane.add(tfcname);

tfcname.setColumns(10);

JLabel lblLoginToLuton =new JLabel();

lblLoginToLuton.setBounds(193, 45, 656, 55);

lblLoginToLuton.setForeground(Color.GREEN);

lblLoginToLuton.setBackground(Color.BLACK);

lblLoginToLuton.setText("CORPORATE CUSTOMER REGISTRATION");

lblLoginToLuton.setFont(new Font("Times New Roman", Font.BOLD, 28));

desktopPane.add(lblLoginToLuton);

tfcemail = new JTextField();

tfcemail.setBackground(Color.WHITE);

tfcemail.setBounds(310, 285, 410, 43);

tfcemail.setColumns(10);

desktopPane.add(tfcemail);

JLabel lblcompanyname = new JLabel();

lblcompanyname.setText("Company Name");

lblcompanyname.setForeground(Color.BLUE);

lblcompanyname.setFont(new Font("Times New Roman", Font.BOLD, 26));

lblcompanyname.setBackground(Color.BLACK);

lblcompanyname.setBounds(54, 141, 194, 40);

desktopPane.add(lblcompanyname);

JLabel lblEmail = new JLabel();

lblEmail.setText("Email");

lblEmail.setForeground(Color.BLUE);

lblEmail.setFont(new Font("Times New Roman", Font.BOLD, 26));

lblEmail.setBackground(Color.BLACK);

lblEmail.setBounds(54, 288, 144, 40);

desktopPane.add(lblEmail);

JButton btncompanysignup = new JButton("Sign Up");

btncompanysignup.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

userCorporate();

}

});

btncompanysignup.setBounds(601, 539, 119, 43);

btncompanysignup.setFont(new Font("Times New Roman", Font.BOLD, 20));

desktopPane.add(btncompanysignup);

tfccontact = new JTextField();

tfccontact.setColumns(10);

tfccontact.setBackground(Color.WHITE);

tfccontact.setBounds(310, 212, 410, 43);

desktopPane.add(tfccontact);

JLabel lblContactNo = new JLabel();

lblContactNo.setText("Contact ");

lblContactNo.setForeground(Color.BLUE);

lblContactNo.setFont(new Font("Times New Roman", Font.BOLD, 26));

lblContactNo.setBackground(Color.BLACK);

lblContactNo.setBounds(54, 215, 144, 40);

desktopPane.add(lblContactNo);

JLabel lblcomppass = new JLabel();

lblcomppass.setText("Password");

lblcomppass.setForeground(Color.BLUE);

lblcomppass.setFont(new Font("Times New Roman", Font.BOLD, 26));

lblcomppass.setBackground(Color.BLACK);

lblcomppass.setBounds(52, 353, 146, 40);

desktopPane.add(lblcomppass);

tfcpass = new JTextField();

tfcpass.setColumns(10);

tfcpass.setBackground(Color.WHITE);

tfcpass.setBounds(310, 350, 410, 43);

desktopPane.add(tfcpass);

JLabel lblcompconpass = new JLabel();

lblcompconpass.setText("Confirm Password ");

lblcompconpass.setForeground(Color.BLUE);

lblcompconpass.setFont(new Font("Times New Roman", Font.BOLD, 26));

lblcompconpass.setBackground(Color.BLACK);

lblcompconpass.setBounds(54, 426, 239, 40);

desktopPane.add(lblcompconpass);

tfcconfirmpass = new JTextField();

tfcconfirmpass.setColumns(10);

tfcconfirmpass.setBackground(Color.WHITE);

tfcconfirmpass.setBounds(310, 423, 410, 43);

desktopPane.add(tfcconfirmpass);

JButton btnbackc = new JButton("Back");

btnbackc.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Register signUp = new Register();

signUp.setVisible(true);

}

});

btnbackc.setFont(new Font("Times New Roman", Font.BOLD, 20));

btnbackc.setBounds(54, 539, 119, 43);

desktopPane.add(btnbackc);

}

private void userCorporate() {

// On click of the log In button

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

Corporate corporate = new Corporate();

corporate.setCompany\_Name(tfcname.getText());

corporate.setContact(tfccontact.getText());

corporate.setEmail(tfcemail.getText());

corporate.setPassword(tfcpass.getText());

corporate.setConfirm\_Password(tfcconfirmpass.getText());

BLCorporate userSL = new BLCorporate();

userSL.validateCorporate(corporate);

userSL.corporateinf(corporate);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

NewLogin

package Frontend;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BusinessLayer.BLCorporate;

import BusinessLayer.BLNonCorporate;

import Helper.InputException;

import Models.AllModel;

import Models.Corporate;

import Models.NonCorporate;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JDesktopPane;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JTextField;

import javax.swing.JCheckBox;

public class NewLogin extends JFrame {

private JPanel contentPane;

private JTextField tfemail;

private JTextField tfpassword;

private JCheckBox userChoiseBox;

/\*\*

\* Launch the application.

\*/

public static void main (String[] args) {

try {

NewLogin window =new NewLogin();

window.setVisible(true);

}catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the frame.

\*/

public NewLogin() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(265, 72, 800, 546);

desktopPane.setBackground(Color.GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

tfemail = new JTextField();

tfemail.setBackground(Color.WHITE);

tfemail.setBounds(377, 208, 322, 43);

desktopPane.add(tfemail);

tfemail.setColumns(10);

JLabel lblLoginToLuton =new JLabel();

lblLoginToLuton.setBounds(99, 21, 599, 55);

lblLoginToLuton.setForeground(Color.YELLOW);

lblLoginToLuton.setBackground(Color.BLACK);

lblLoginToLuton.setText("WELCOME TO LUTON HOTEL");

lblLoginToLuton.setFont(new Font ("Times New Roman", Font.BOLD, 40));

desktopPane.add(lblLoginToLuton);

tfpassword = new JTextField();

tfpassword.setBackground(Color.WHITE);

tfpassword.setBounds(377, 281, 322, 43);

tfpassword.setColumns(10);

desktopPane.add(tfpassword);

JButton clickheretosignupbtn = new JButton("Don't have an account? Sign Up!");

clickheretosignupbtn.setFont(new Font("Tahoma", Font.PLAIN, 14));

clickheretosignupbtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Register signUp = new Register();

signUp.setVisible(true);

}

});

clickheretosignupbtn.setBackground(Color.GRAY);

clickheretosignupbtn.setForeground(Color.BLUE);

clickheretosignupbtn.setBounds(247, 457, 264, 43);

desktopPane.add(clickheretosignupbtn);

JLabel labelemail = new JLabel();

labelemail.setText("Email");

labelemail.setForeground(Color.BLUE);

labelemail.setFont(new Font("Times New Roman", Font.BOLD, 40));

labelemail.setBackground(Color.BLACK);

labelemail.setBounds(141, 208, 194, 40);

desktopPane.add(labelemail);

JLabel labelpassword = new JLabel();

labelpassword.setText("Password");

labelpassword.setForeground(Color.BLUE);

labelpassword.setFont(new Font("Times New Roman", Font.BOLD, 40));

labelpassword.setBackground(Color.BLACK);

labelpassword.setBounds(141, 281, 194, 40);

desktopPane.add(labelpassword);

userChoiseBox = new JCheckBox("Corporate User");

userChoiseBox.setBounds(377, 346, 119, 23);

desktopPane.add(userChoiseBox);

JButton loginbutton = new JButton("Login");

loginbutton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(userChoiseBox.isSelected()) {

corporateinf();

}

else {

nonCorporateinf();

}

}

});

loginbutton.setBounds(309, 388, 119, 43);

loginbutton.setFont(new Font("Times New Roman", Font.BOLD, 20));

desktopPane.add(loginbutton);

JLabel lblfeelLikeHome = new JLabel();

lblfeelLikeHome.setText("\"FEEL LIKE HOME\"");

lblfeelLikeHome.setForeground(Color.ORANGE);

lblfeelLikeHome.setFont(new Font("Times New Roman", Font.BOLD, 36));

lblfeelLikeHome.setBackground(Color.BLACK);

lblfeelLikeHome.setBounds(231, 87, 407, 55);

desktopPane.add(lblfeelLikeHome);

}

private void corporateinf() {

// On click of the log In buttond

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

Corporate corporateLogin = new Corporate();

corporateLogin.setEmail(tfemail.getText());

corporateLogin.setPassword(String.valueOf(tfpassword.getText()));

BLCorporate corporateLoginBL = new BLCorporate();

corporateLoginBL.validateLogin(corporateLogin);

corporateLoginBL.logininf(corporateLogin);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

private void nonCorporateinf() {

// On click of the log In button

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

AllModel model = new AllModel();

model.setEmail(tfemail.getText());

model.setPassword(String.valueOf(tfpassword.getText()));

BLNonCorporate noncorporateLoginBL = new BLNonCorporate();

noncorporateLoginBL.validateLogin(model);

noncorporateLoginBL.logininf(model);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

NewNonCorporate

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

import BusinessLayer.BLCorporate;

import BusinessLayer.BLNonCorporate;

import Helper.InputException;

import Models.Corporate;

import Models.Login;

import Models.NonCorporate;

import javax.swing.JCheckBox;

import javax.swing.ImageIcon;

public class NewNonCorporate extends JFrame {

private JPanel contentPane;

private JTextField textFieldconfirmpass2;

private JTextField tffname1;

private JTextField tflname1;

private JTextField tfadd2;

private JTextField tfccno;

private JTextField tfemail2;

private JTextField tfcpass2;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

NewNonCorporate frame = new NewNonCorporate();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public NewNonCorporate() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(263, 33, 849, 627);

desktopPane.setBackground(Color.GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JLabel lblLoginToLuton =new JLabel();

lblLoginToLuton.setBounds(71, 46, 656, 55);

lblLoginToLuton.setForeground(Color.GREEN);

lblLoginToLuton.setBackground(Color.BLACK);

lblLoginToLuton.setText("NON-CORPORATE CUSTOMER REGISTRATION");

lblLoginToLuton.setFont(new Font("Times New Roman", Font.BOLD, 28));

desktopPane.add(lblLoginToLuton);

JLabel lblFirstName = new JLabel();

lblFirstName.setText("First Name");

lblFirstName.setForeground(Color.BLUE);

lblFirstName.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblFirstName.setBackground(Color.BLACK);

lblFirstName.setBounds(54, 141, 144, 40);

desktopPane.add(lblFirstName);

JLabel lblEmail = new JLabel();

lblEmail.setText("Email");

lblEmail.setForeground(Color.BLUE);

lblEmail.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblEmail.setBackground(Color.BLACK);

lblEmail.setBounds(54, 331, 144, 40);

desktopPane.add(lblEmail);

JButton btnsignup = new JButton("Sign Up");

btnsignup.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

nonCorporate();

}

});

btnsignup.setBounds(644, 539, 119, 43);

btnsignup.setFont(new Font("Times New Roman", Font.BOLD, 20));

desktopPane.add(btnsignup);

JLabel lbllastname = new JLabel();

lbllastname.setText("Last Name ");

lbllastname.setForeground(Color.BLUE);

lbllastname.setFont(new Font("Times New Roman", Font.BOLD, 22));

lbllastname.setBackground(Color.BLACK);

lbllastname.setBounds(54, 192, 144, 40);

desktopPane.add(lbllastname);

JLabel lblpassword = new JLabel();

lblpassword.setText("Password");

lblpassword.setForeground(Color.BLUE);

lblpassword.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblpassword.setBackground(Color.BLACK);

lblpassword.setBounds(52, 382, 146, 40);

desktopPane.add(lblpassword);

JLabel lblverifypass = new JLabel();

lblverifypass.setText("Confirm Password ");

lblverifypass.setForeground(Color.BLUE);

lblverifypass.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblverifypass.setBackground(Color.BLACK);

lblverifypass.setBounds(54, 433, 239, 40);

desktopPane.add(lblverifypass);

textFieldconfirmpass2 = new JTextField();

textFieldconfirmpass2.setColumns(10);

textFieldconfirmpass2.setBackground(Color.WHITE);

textFieldconfirmpass2.setBounds(287, 433, 292, 34);

desktopPane.add(textFieldconfirmpass2);

JButton btnback = new JButton("Back");

btnback.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Register signUp = new Register();

signUp.setVisible(true);

}

});

btnback.setFont(new Font("Times New Roman", Font.BOLD, 20));

btnback.setBounds(54, 539, 119, 43);

desktopPane.add(btnback);

JLabel lblAddress = new JLabel();

lblAddress.setText("Address");

lblAddress.setForeground(Color.BLUE);

lblAddress.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblAddress.setBackground(Color.BLACK);

lblAddress.setBounds(54, 237, 144, 40);

desktopPane.add(lblAddress);

tffname1 = new JTextField();

tffname1.setColumns(10);

tffname1.setBackground(Color.WHITE);

tffname1.setBounds(287, 147, 292, 34);

desktopPane.add(tffname1);

tflname1 = new JTextField();

tflname1.setColumns(10);

tflname1.setBackground(Color.WHITE);

tflname1.setBounds(287, 192, 292, 34);

desktopPane.add(tflname1);

tfadd2 = new JTextField();

tfadd2.setColumns(10);

tfadd2.setBackground(Color.WHITE);

tfadd2.setBounds(287, 237, 292, 34);

desktopPane.add(tfadd2);

tfccno = new JTextField();

tfccno.setColumns(10);

tfccno.setBackground(Color.WHITE);

tfccno.setBounds(287, 285, 292, 34);

desktopPane.add(tfccno);

tfemail2 = new JTextField();

tfemail2.setColumns(10);

tfemail2.setBackground(Color.WHITE);

tfemail2.setBounds(287, 330, 292, 34);

desktopPane.add(tfemail2);

tfcpass2 = new JTextField();

tfcpass2.setColumns(10);

tfcpass2.setBackground(Color.WHITE);

tfcpass2.setBounds(287, 382, 292, 34);

desktopPane.add(tfcpass2);

JLabel lblCreditcardno = new JLabel();

lblCreditcardno.setText("Credit-Card-No");

lblCreditcardno.setForeground(Color.BLUE);

lblCreditcardno.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblCreditcardno.setBackground(Color.BLACK);

lblCreditcardno.setBounds(54, 280, 171, 40);

desktopPane.add(lblCreditcardno);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\rn.png"));

lblNewLabel.setBackground(Color.ORANGE);

lblNewLabel.setBounds(623, 165, 179, 282);

desktopPane.add(lblNewLabel);

}

private void nonCorporate() {

// On click of the log In button

// It fetch the data from the field

// Create an object of Service Layer and pass the model to Service layer

// Perform the required action from the Service layer.

try {

NonCorporate noncorporate = new NonCorporate();

noncorporate.setFirst\_Name(tffname1.getText());

noncorporate.setLast\_Name(tflname1.getText());

noncorporate.setAddress(tfadd2.getText());

noncorporate.setCredit\_Card\_No(tfccno.getText());

noncorporate.setEmail(tfemail2.getText());

noncorporate.setPassword(tfcpass2.getText());

noncorporate.setConfirm\_Password(textFieldconfirmpass2.getText());

BLNonCorporate userSL = new BLNonCorporate();

userSL.validateNonCorporate(noncorporate);

userSL.noncorporateinf(noncorporate);

}

catch(InputException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

catch(Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}

Register

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

import Models.Login;

import javax.swing.ImageIcon;

import java.awt.Toolkit;

public class Register extends JFrame {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Register frame = new Register();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Register() {

setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\icon.png"));

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(307, 117, 745, 519);

desktopPane.setBackground(Color.ORANGE);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JButton btnNewButton\_1\_1 = new JButton("Already have Account ! Login?");

btnNewButton\_1\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewLogin signUp = new NewLogin();

signUp.setVisible(true);

}

});

btnNewButton\_1\_1.setBackground(Color.GRAY);

btnNewButton\_1\_1.setForeground(Color.BLUE);

btnNewButton\_1\_1.setBounds(260, 448, 214, 31);

desktopPane.add(btnNewButton\_1\_1);

JButton btnNewButton\_1 = new JButton("NON-CORPORATE ");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewNonCorporate Signup = new NewNonCorporate();

Signup.setVisible(true);

}

});

btnNewButton\_1.setBounds(441, 226, 278, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 20));

desktopPane.add(btnNewButton\_1);

JLabel lblfeelLikeHome = new JLabel();

lblfeelLikeHome.setText("SIGN UP TO LUTON HOTEL");

lblfeelLikeHome.setForeground(Color.BLACK);

lblfeelLikeHome.setFont(new Font("Times New Roman", Font.BOLD, 38));

lblfeelLikeHome.setBackground(Color.BLACK);

lblfeelLikeHome.setBounds(177, 52, 525, 56);

desktopPane.add(lblfeelLikeHome);

JButton btnNewButton\_1\_2 = new JButton("CORPORATE ");

btnNewButton\_1\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewCorporate signUp = new NewCorporate();

signUp.setVisible(true);

}

});

btnNewButton\_1\_2.setFont(new Font("Times New Roman", Font.BOLD, 20));

btnNewButton\_1\_2.setBounds(441, 305, 278, 40);

desktopPane.add(btnNewButton\_1\_2);

JLabel lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\1.jpg"));

lblNewLabel.setBounds(46, 141, 349, 266);

desktopPane.add(lblNewLabel);

}

}

Rooms

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

public class Rooms extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Rooms frame = new Rooms();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Rooms() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(176, 74, 1033, 554);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_1 = new JButton("Single Room ");

btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

SingleRoom signIn = new SingleRoom();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2\_1.setBounds(86, 402, 193, 40);

desktopPane.add(btnNewButton\_1\_2\_1);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Home signIn = new Home();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(891, 490, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JButton btnNewButton\_1\_2\_1\_1 = new JButton("Double Room");

btnNewButton\_1\_2\_1\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

DoubleRoom signIn = new DoubleRoom();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_1\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1\_1.setBackground(Color.GRAY);

btnNewButton\_1\_2\_1\_1.setBounds(419, 402, 211, 40);

desktopPane.add(btnNewButton\_1\_2\_1\_1);

JButton btnNewButton\_1\_2\_1\_2 = new JButton("Twin Room");

btnNewButton\_1\_2\_1\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

TwinRoom signIn = new TwinRoom();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_1\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1\_2.setBackground(Color.GRAY);

btnNewButton\_1\_2\_1\_2.setBounds(792, 402, 171, 40);

desktopPane.add(btnNewButton\_1\_2\_1\_2);

JLabel lblNewLabel = new JLabel("\r\n");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\single.jpg"));

lblNewLabel.setBounds(20, 99, 311, 286);

desktopPane.add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("\r\n");

lblNewLabel\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\d3.jpg"));

lblNewLabel\_1.setBounds(356, 99, 332, 286);

desktopPane.add(lblNewLabel\_1);

JLabel lblNewLabel\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\twin.png"));

lblNewLabel\_1\_1.setBounds(715, 99, 295, 286);

desktopPane.add(lblNewLabel\_1\_1);

}

}

SingleRoom

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

public class SingleRoom extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

SingleRoom frame = new SingleRoom();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public SingleRoom() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(168, 52, 1033, 611);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_1 = new JButton("Single Rooms");

btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

btnNewButton\_1\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2\_1.setBounds(395, 65, 193, 40);

desktopPane.add(btnNewButton\_1\_2\_1);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms signIn = new Rooms();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(891, 549, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblNewLabel\_1 = new JLabel("\r\n");

lblNewLabel\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\single2.jpg"));

lblNewLabel\_1.setBounds(536, 139, 393, 248);

desktopPane.add(lblNewLabel\_1);

JLabel lblNewLabel\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\single 3.png"));

lblNewLabel\_1\_1.setBounds(98, 136, 393, 254);

desktopPane.add(lblNewLabel\_1\_1);

JLabel lblTheseRoomsAre = new JLabel();

lblTheseRoomsAre.setText("These rooms are assigned to one person or a couple. It contains only one bed ");

lblTheseRoomsAre.setForeground(Color.BLACK);

lblTheseRoomsAre.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblTheseRoomsAre.setBackground(Color.BLACK);

lblTheseRoomsAre.setBounds(126, 398, 737, 40);

desktopPane.add(lblTheseRoomsAre);

JLabel lblWithSofaTelevision = new JLabel();

lblWithSofaTelevision.setText("with sofa, television and a drawyer.");

lblWithSofaTelevision.setForeground(Color.BLACK);

lblWithSofaTelevision.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblWithSofaTelevision.setBackground(Color.BLACK);

lblWithSofaTelevision.setBounds(126, 432, 352, 40);

desktopPane.add(lblWithSofaTelevision);

JLabel lblCostPer = new JLabel();

lblCostPer.setText("COST: 250$ PER DAY ");

lblCostPer.setForeground(Color.BLUE);

lblCostPer.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblCostPer.setBackground(new Color(0, 0, 255));

lblCostPer.setBounds(376, 487, 242, 40);

desktopPane.add(lblCostPer);

}

}

TwinRoom

package Frontend;

import java.awt.Color;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDesktopPane;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

public class TwinRoom extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

TwinRoom frame = new TwinRoom();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public TwinRoom() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(0, 0, 1920, 1080);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JDesktopPane desktopPane = new JDesktopPane();

desktopPane.setBounds(167, 25, 1033, 613);

desktopPane.setBackground(Color.LIGHT\_GRAY);

contentPane.add(desktopPane);

desktopPane.setLayout(null);

desktopPane.setOpaque(true);

JButton btnNewButton\_1 = new JButton("Book Now");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

NewBooking signIn = new NewBooking();

signIn.setVisible(true);

}

});

btnNewButton\_1.setBackground(Color.YELLOW);

btnNewButton\_1.setBounds(856, 11, 154, 40);

btnNewButton\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

desktopPane.add(btnNewButton\_1);

JLabel lblLutonHotel = new JLabel();

lblLutonHotel.setText("LUTON HOTEL");

lblLutonHotel.setForeground(Color.BLACK);

lblLutonHotel.setFont(new Font("Times New Roman", Font.BOLD, 28));

lblLutonHotel.setBackground(Color.BLACK);

lblLutonHotel.setBounds(20, 11, 235, 40);

desktopPane.add(lblLutonHotel);

JButton btnNewButton\_1\_2\_1 = new JButton("Twin Rooms");

btnNewButton\_1\_2\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

btnNewButton\_1\_2\_1.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_1.setBackground(new Color(128, 128, 128));

btnNewButton\_1\_2\_1.setBounds(414, 78, 205, 40);

desktopPane.add(btnNewButton\_1\_2\_1);

JButton btnNewButton\_1\_2\_2 = new JButton("Back");

btnNewButton\_1\_2\_2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Rooms signIn = new Rooms();

signIn.setVisible(true);

}

});

btnNewButton\_1\_2\_2.setFont(new Font("Times New Roman", Font.BOLD, 26));

btnNewButton\_1\_2\_2.setBackground(Color.GREEN);

btnNewButton\_1\_2\_2.setBounds(891, 549, 119, 40);

desktopPane.add(btnNewButton\_1\_2\_2);

JLabel lblCostPer = new JLabel();

lblCostPer.setText("COST: 400$ PER DAY ");

lblCostPer.setForeground(Color.BLUE);

lblCostPer.setFont(new Font("Times New Roman", Font.BOLD, 22));

lblCostPer.setBackground(new Color(0, 0, 255));

lblCostPer.setBounds(404, 487, 242, 40);

desktopPane.add(lblCostPer);

JLabel lblNewLabel\_1\_1\_1\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1\_1\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\twin2.jpg"));

lblNewLabel\_1\_1\_1\_1\_1.setBounds(552, 144, 393, 254);

desktopPane.add(lblNewLabel\_1\_1\_1\_1\_1);

JLabel lblNewLabel\_1\_1\_1\_1\_1\_1 = new JLabel("\r\n");

lblNewLabel\_1\_1\_1\_1\_1\_1.setIcon(new ImageIcon("C:\\Users\\home\\OneDrive\\Desktop\\Images for pop\\twin1.jpg"));

lblNewLabel\_1\_1\_1\_1\_1\_1.setBounds(103, 144, 393, 254);

desktopPane.add(lblNewLabel\_1\_1\_1\_1\_1\_1);

JLabel lblRoomFacilitiesFurnished = new JLabel();

lblRoomFacilitiesFurnished.setText("Most of the rooms are with two separate beds, but there are also a few rooms with a double bed. Room size appr. 12-14 sqm. The");

lblRoomFacilitiesFurnished.setForeground(Color.BLACK);

lblRoomFacilitiesFurnished.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblRoomFacilitiesFurnished.setBackground(Color.BLACK);

lblRoomFacilitiesFurnished.setBounds(123, 409, 797, 23);

desktopPane.add(lblRoomFacilitiesFurnished);

JLabel lblRoomSizeAppr = new JLabel();

lblRoomSizeAppr.setText(" room is fully equipped with private bath room with shower, WC, hairdryer, umbrella, shoe spoon, telephone, TV, safe box");

lblRoomSizeAppr.setForeground(Color.BLACK);

lblRoomSizeAppr.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblRoomSizeAppr.setBackground(Color.BLACK);

lblRoomSizeAppr.setBounds(123, 432, 809, 23);

desktopPane.add(lblRoomSizeAppr);

JLabel lblMinibarWorkingDesk = new JLabel();

lblMinibarWorkingDesk.setText(" minibar, working desk and free WiFi. Max. occupancy 2 persons.\r\n\r\n");

lblMinibarWorkingDesk.setForeground(Color.BLACK);

lblMinibarWorkingDesk.setFont(new Font("Times New Roman", Font.BOLD, 14));

lblMinibarWorkingDesk.setBackground(Color.BLACK);

lblMinibarWorkingDesk.setBounds(123, 453, 809, 23);

desktopPane.add(lblMinibarWorkingDesk);

}

}

PACKAGE MODELS

AllModel

package Models;

public class AllModel {

private int Booking\_ID;

private int Customer\_ID;

private int user\_ID;

private String Company\_Name;

private String Contact;

private String Email;

private String First\_Name ;

private String Last\_Name;

private String Address;

private String Credit\_Card\_No;

private String Password;

private String FullName;

private String Check\_in\_date;

private String Check\_out\_date;

private String Room\_Type;

private String booking\_status;

private String Room\_No;

private String Room\_Rate;

private String Room\_Status;

private String Postal\_code;

private int roomNo;

private int totalbill;

public AllModel() {

this.Booking\_ID = 0;

this.Customer\_ID = 0;

this.user\_ID = 0;

this.Company\_Name = "";

this.Contact = "";

this.Email = "";

this.First\_Name = "";

this.Last\_Name = "";

this.Address = "";

this.Credit\_Card\_No = "";

this.Password = "";

this.FullName = "";

this.Check\_in\_date = "";

this.Check\_out\_date = "";

this.Room\_Type = "";

this.Room\_No = "";

this.Room\_Rate = "";

this.Room\_Status = "";

this.booking\_status= "";

this.Postal\_code = "";

this.roomNo = 0;

this.totalbill=0;

}

public AllModel(int Booking\_ID,int Customer\_ID, int user\_ID, String Company\_Name, String Contact, String Email, String First\_Name, String booking\_status,

String Last\_Name, String Address , String Password, String FullName, String Check\_in\_date, String Check\_out\_date , String Room\_Type, String Postal\_code, String Room\_No, String Room\_Rate, String Room\_Status) {

super();

this.Booking\_ID=Booking\_ID;

this.Customer\_ID = Customer\_ID;

this.user\_ID = user\_ID;

this.Company\_Name = Company\_Name;

this.Contact = Contact;

this.Email = Email;

this.First\_Name= First\_Name;

this.Last\_Name = Last\_Name;

this.Address= Address;

this.Password= Password;

this.FullName =FullName;

this.Check\_in\_date= Check\_in\_date;

this.Room\_Type= Room\_Type;

this.Room\_No= Room\_No;

this.Room\_Rate= Room\_Rate;

this. booking\_status = booking\_status;

this.Room\_Status= Room\_Status;

this.Postal\_code=Postal\_code;

}

public int getBooking\_ID() {

return Booking\_ID;

}

public void setBooking\_ID(int booking\_ID) {

Booking\_ID = booking\_ID;

}

public int getCustomer\_ID() {

return Customer\_ID;

}

public void setCustomer\_ID(int customer\_ID) {

Customer\_ID = customer\_ID;

}

public int getUser\_ID() {

return user\_ID;

}

public void setUser\_ID(int user\_ID) {

this.user\_ID = user\_ID;

}

public String getCompany\_Name() {

return Company\_Name;

}

public void setCompany\_Name(String company\_Name) {

Company\_Name = company\_Name;

}

public String getContact() {

return Contact;

}

public void setContact(String contact) {

Contact = contact;

}

public String getEmail() {

return Email;

}

public void setEmail(String email) {

Email = email;

}

public String getFirst\_Name() {

return First\_Name;

}

public void setFirst\_Name(String first\_Name) {

First\_Name = first\_Name;

}

public String getLast\_Name() {

return Last\_Name;

}

public void setLast\_Name(String last\_Name) {

Last\_Name = last\_Name;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

public String getCredit\_Card\_No() {

return Credit\_Card\_No;

}

public void setCredit\_Card\_No(String credit\_Card\_No) {

Credit\_Card\_No = credit\_Card\_No;

}

public String getPassword() {

return Password;

}

public void setPassword(String password) {

Password = password;

}

public String getFullName() {

return FullName;

}

public void setFullName(String fullName) {

FullName = fullName;

}

public String getCheck\_in\_date() {

return Check\_in\_date;

}

public void setCheck\_in\_date(String check\_in\_date) {

Check\_in\_date = check\_in\_date;

}

public String getCheck\_out\_date() {

return Check\_out\_date;

}

public void setCheck\_out\_date(String check\_out\_date) {

Check\_out\_date = check\_out\_date;

}

public String getRoom\_Type() {

return Room\_Type;

}

public String getRoom\_No() {

return Room\_No;

}

public void setRoom\_No(String room\_No) {

Room\_No = room\_No;

}

public String getRoom\_Rate() {

return Room\_Rate;

}

public void setRoom\_Rate(String room\_Rate) {

Room\_Rate = room\_Rate;

}

public String getRoom\_Status() {

return Room\_Status;

}

public void setRoom\_Status(String room\_Status) {

Room\_Status = room\_Status;

}

public void setRoom\_Type(String room\_Type) {

Room\_Type = room\_Type;

}

public String getPostal\_code() {

return Postal\_code;

}

public void setPostal\_code(String postal\_code) {

Postal\_code = postal\_code;

}

public String getBooking\_status() {

return booking\_status;

}

public void setBooking\_status(String booking\_status) {

this.booking\_status = booking\_status;

}

public int getRoomNo() {

return roomNo;

}

public void setRoomNo(int roomNo) {

this.roomNo = roomNo;

}

public int getTotalbill() {

return totalbill;

}

public void setTotalbill(int totalbill) {

this.totalbill = totalbill;

}

}

MODEL: Booking

package Models;

public class Booking {

private int Booking\_ID;

private String FullName;

private String Address;

private String Check\_in\_date;

private String Check\_out\_date;

private String Room\_Type;

private String Postal\_code;

private int roomNo;

public Booking() {

this.Booking\_ID = 0;

this.FullName ="";

this.Address ="";

this.Check\_in\_date = "";

this.Check\_out\_date= "";

this.Room\_Type= "";

this.Postal\_code = "";

this.roomNo = 0;

}

public Booking(int booking\_ID,String fullname,String address, String check\_in\_date, String check\_out\_date,String Room\_Type, String postal\_code, int roomNo) {

this.Booking\_ID = booking\_ID;

this.FullName = fullname;

this.Address = address;

this.Check\_in\_date = check\_in\_date;

this.Check\_out\_date = check\_out\_date;

this.Room\_Type = Room\_Type;

this.Postal\_code = postal\_code;

this.roomNo = roomNo;

}

public int getBooking\_ID() {

return Booking\_ID;

}

public void setBooking\_ID(int booking\_ID) {

Booking\_ID = booking\_ID;

}

public String getFullName() {

return FullName;

}

public void setFullName(String fullName) {

FullName = fullName;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

public String getCheck\_in\_date() {

return Check\_in\_date;

}

public void setCheck\_in\_date(String check\_in\_date) {

Check\_in\_date = check\_in\_date;

}

public String getCheck\_out\_date() {

return Check\_out\_date;

}

public void setCheck\_out\_date(String check\_out\_date) {

Check\_out\_date = check\_out\_date;

}

public String getRoom\_Type() {

return Room\_Type;

}

public void setRoom\_Type(String room\_Type) {

Room\_Type = room\_Type;

}

public String getPostal\_code() {

return Postal\_code;

}

public void setPostal\_code(String postal\_code) {

Postal\_code = postal\_code;

}

public int getRoomNo() {

return roomNo;

}

public void setRoomNo(int roomNo) {

this.roomNo = roomNo;

}

}

MODEL: BookingDetails

package Models;

public class BookingDetails {

private int Booking\_ID;

private String Address;

private String FullName;

private String Check\_in\_date;

private String Check\_out\_date;

private String booking\_status;

private String Room\_Rate;

private String Room\_Type;

/\*\* for(int i=0; i<models.size(); i++) {

rowData[0] = models.get(i).getFullName();

rowData[1] = models.get(i).getAddress();

rowData[2] = models.get(i).getCheck\_in\_date();

rowData[3] = models.get(i).getCheck\_out\_date();

rowData[4] = models.get(i).getRoom\_Type();

rowData[5] = models.get(i).getBooking\_status();

rowData[6] = models.get(i).getBooking\_ID();

\*\*/

public BookingDetails() {

this.Booking\_ID = 0;

this.Address = "";

this.FullName = "";

this.Check\_in\_date = "";

this.Check\_out\_date = "";

this.Room\_Rate = "";

this.Room\_Type = "";

this.booking\_status= "";

}

public BookingDetails(int Booking\_ID, String booking\_status,

String Address , String FullName, String Check\_in\_date, String Check\_out\_date , String Room\_No, String Room\_Rate) {

super();

this.Booking\_ID=Booking\_ID;

this.FullName =FullName;

this.Address =Address;

this.Check\_in\_date= Check\_in\_date;

this.Room\_Rate= Room\_Rate;

this.Room\_Type= Room\_Type;

this.booking\_status = booking\_status;

}

public int getBooking\_ID() {

return Booking\_ID;

}

public void setBooking\_ID(int booking\_ID) {

Booking\_ID = booking\_ID;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

public String getFullName() {

return FullName;

}

public void setFullName(String fullName) {

FullName = fullName;

}

public String getCheck\_in\_date() {

return Check\_in\_date;

}

public void setCheck\_in\_date(String check\_in\_date) {

Check\_in\_date = check\_in\_date;

}

public String getCheck\_out\_date() {

return Check\_out\_date;

}

public void setCheck\_out\_date(String check\_out\_date) {

Check\_out\_date = check\_out\_date;

}

public String getBooking\_status() {

return booking\_status;

}

public void setBooking\_status(String booking\_status) {

this.booking\_status = booking\_status;

}

public String getRoom\_Rate() {

return Room\_Rate;

}

public void setRoom\_Rate(String room\_Rate) {

Room\_Rate = room\_Rate;

}

public String getRoom\_Type() {

return Room\_Type;

}

public void setRoom\_Type(String room\_Type) {

Room\_Type = room\_Type;

}

}

MODEL: Corporate

package Models;

public class Corporate {

private int Customer\_id;

private String Company\_Name;

private String Contact;

private String Email;

private String Password;

private String Confirm\_Password;

public Corporate() {

this.Customer\_id = 0;

this.Company\_Name = "";

this.Contact= "";

this.Email= "";

this.Password = "";

this.Confirm\_Password = "";

}

public Corporate(int Customer\_id , String Company\_Name , String Contact , String Email, String Password, String Confirm\_Password) {

this.Customer\_id =Customer\_id ;

this.Company\_Name = Company\_Name;

this.Contact= Contact;

this.Email= Email;

this.Password = Password;

this.Confirm\_Password = Confirm\_Password;

}

public int getCustomer\_id() {

return Customer\_id;

}

public void setCustomer\_id(int customer\_id) {

Customer\_id = customer\_id;

}

public String getCompany\_Name() {

return Company\_Name;

}

public void setCompany\_Name(String company\_Name) {

Company\_Name = company\_Name;

}

public String getContact() {

return Contact;

}

public void setContact(String contact) {

Contact = contact;

}

public String getEmail() {

return Email;

}

public void setEmail(String email) {

Email = email;

}

public String getPassword() {

return Password;

}

public void setPassword(String password) {

Password = password;

}

public String getConfirm\_Password() {

return Confirm\_Password;

}

public void setConfirm\_Password(String confirm\_Password) {

Confirm\_Password = confirm\_Password;

}

}

MODEL: Login

package Models;

public class Login {

private int user\_ID;

private String Email;

private String Password;

public Login() {

this.user\_ID = 0;

this.Email = "";

this.Password = "";

}

public Login(int id, String name, String address) {

this.user\_ID = id;

this.Email = name;

this.Password = address;

}

public int getUser\_ID() {

return user\_ID;

}

public void setUser\_ID(int user\_ID) {

this.user\_ID = user\_ID;

}

public String getEmail() {

return Email;

}

public void setEmail(String email) {

Email = email;

}

public String getPassword() {

return Password;

}

public void setPassword(String password) {

Password = password;

}

}

MODEL: NonCorporate

package Models;

public class NonCorporate {

private int Customer\_ID ;

private String First\_Name ;

private String Last\_Name;

private String Address;

private String Credit\_Card\_No;

private String Email;

private String Password;

private String Confirm\_Password;

private String Contact;

public NonCorporate() {

this.Customer\_ID = 0;

this.First\_Name = "";

this.Last\_Name = "";

this.Credit\_Card\_No = "";

this.Address = "";

this.Email = "";

this.Password = "";

this.Confirm\_Password = "";

this.Contact = "";

}

public NonCorporate(int customer\_ID, String first\_Name, String last\_Name, String CreditCardNo, String address, String email,

String password, String contact) {

super();

Customer\_ID = customer\_ID;

First\_Name = first\_Name;

Last\_Name = last\_Name;

Credit\_Card\_No = CreditCardNo;

Address = address;

Email = email;

Password= password;

Contact = contact;

}

public int getCustomer\_ID() {

return Customer\_ID;

}

public void setCustomer\_ID(int customer\_ID) {

Customer\_ID = customer\_ID;

}

public String getFirst\_Name() {

return First\_Name;

}

public void setFirst\_Name(String first\_Name) {

First\_Name = first\_Name;

}

public String getLast\_Name() {

return Last\_Name;

}

public void setLast\_Name(String last\_Name) {

Last\_Name = last\_Name;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

public String getCredit\_Card\_No() {

return Credit\_Card\_No;

}

public void setCredit\_Card\_No(String credit\_Card\_No) {

Credit\_Card\_No = credit\_Card\_No;

}

public String getEmail() {

return Email;

}

public void setEmail(String email) {

Email = email;

}

public String getPassword() {

return Password;

}

public void setPassword(String password) {

Password = password;

}

public String getConfirm\_Password() {

return Confirm\_Password;

}

public void setConfirm\_Password(String confirm\_Password) {

Confirm\_Password = confirm\_Password;

}

public String getContact() {

return Contact;

}

public void setContact(String contact) {

Contact = contact;

}

}