hoteLGuru : A Hotel Booking System

GROUP MEMBERS

11010117 – Chetan Anand
11010120 – Harsha Tirumalla
11010122 - Hitesh Arora
11010142 - Mopuru Sanketh
11010145 - Nagelli Kartheek
11010170 - Vishal Anand

Department of Computer Science and Engineering

Technical Documentation

Introduction:

This involves the technical aspects and descriptions of the hotel registration software that we have developed : “hoteLGuru”.

Scope of Document:

This document will give a bird’s eye view as to what this software is capable of performing. It contains:

- ✓ The process flow, i.e. the illustrating flowcharts.
- ✓ The description of different modules, forms and the corresponding functions, et. al.
- ✓ Data modeling.
- ✓ Limitations
- ✓ Scope of future applications.
- ✓ Hardware and software requirements

Objective:

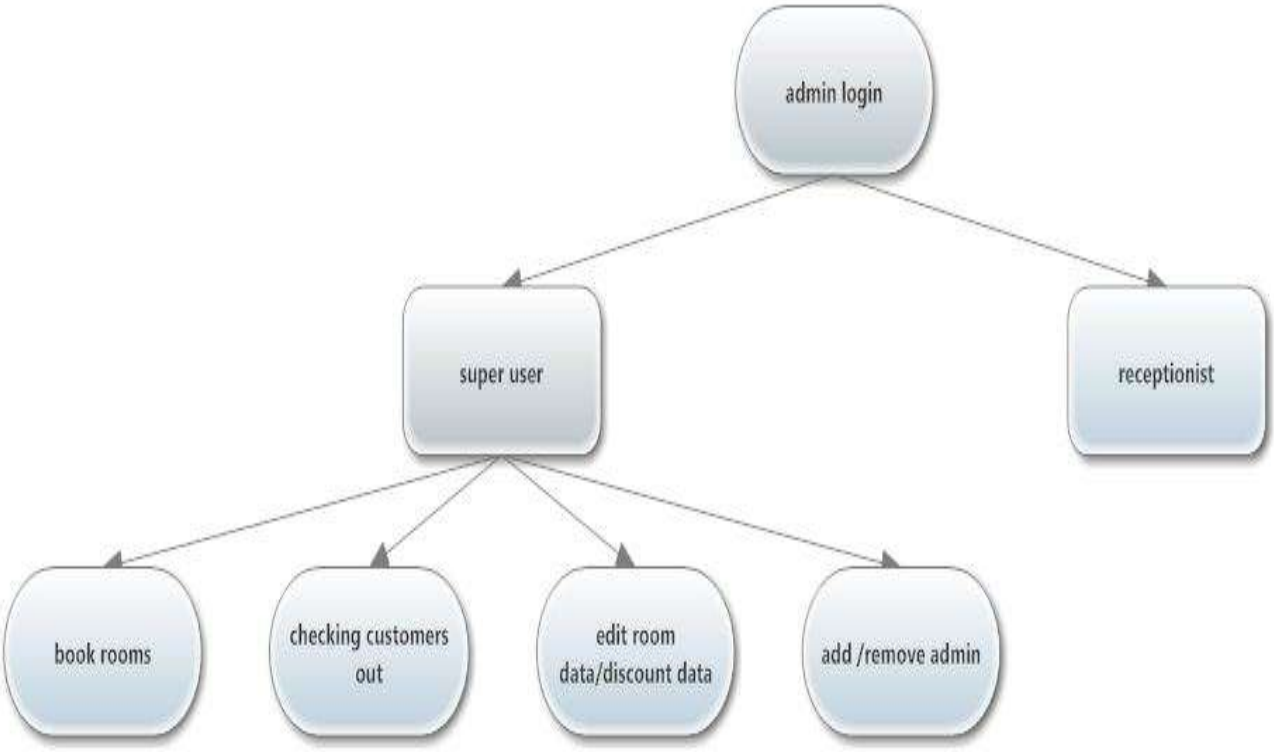
This project aims at to enable the super-users and the hoteliers for the matter to book the rooms on the spot and also enable them to book them in advance with the provision of discounts being awarded for the loyal customers. The Relational Database Management System in SQL using ODBC driver would ensure easier and proper use of the data.

Requirements:

Windows OS, .NET framework, MySQL server,

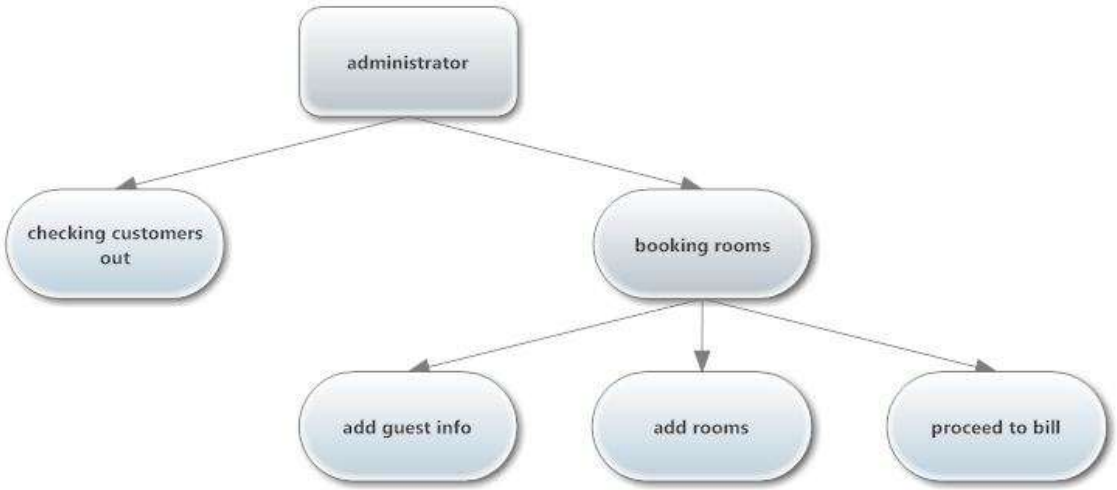
Flowcharts:

- This denotes the logic flow with regards to the super-user/receptionist administrators' log-in.

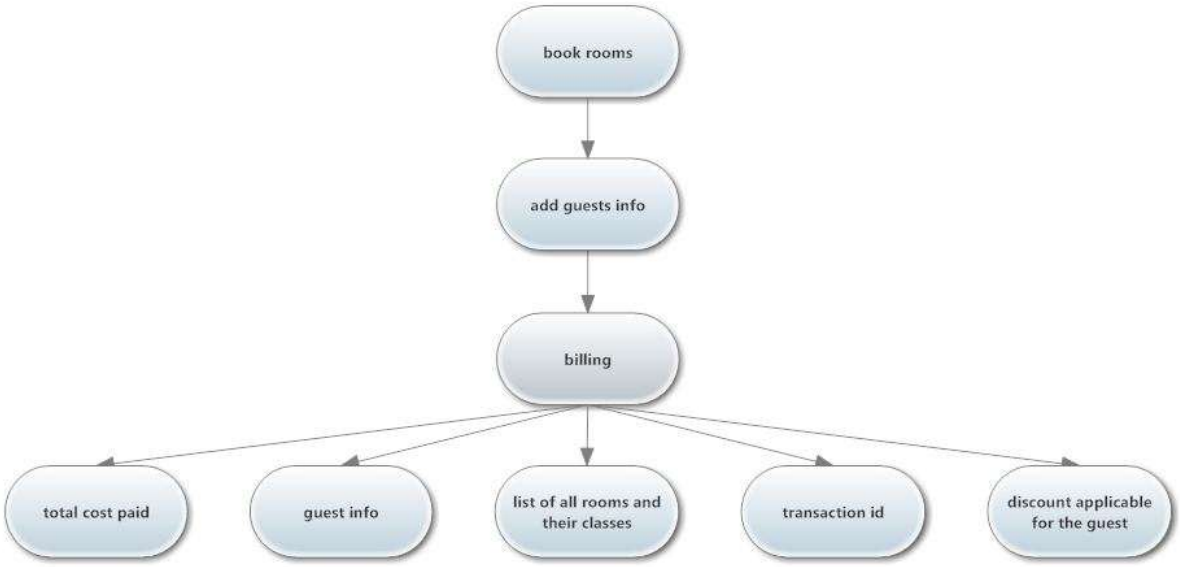


The logic flow with regards to the super-user/receptionist administrators' log-in

- This flow-chart depicts the transfer of logic about the authority of the administrator. The super-user has the additional right about “add new class of rooms“.

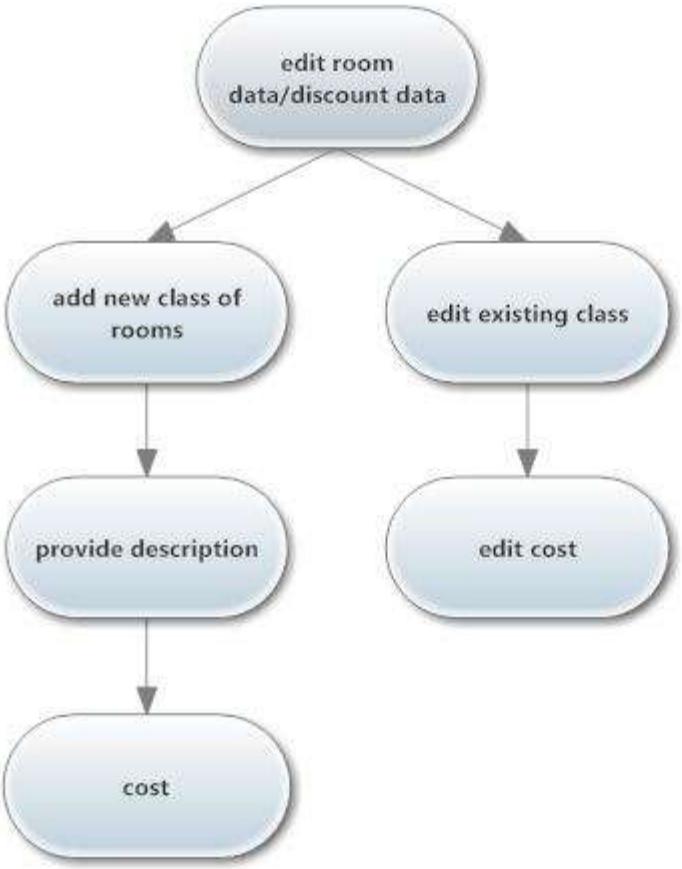


- A schematic on the room(s)-booking procedure.



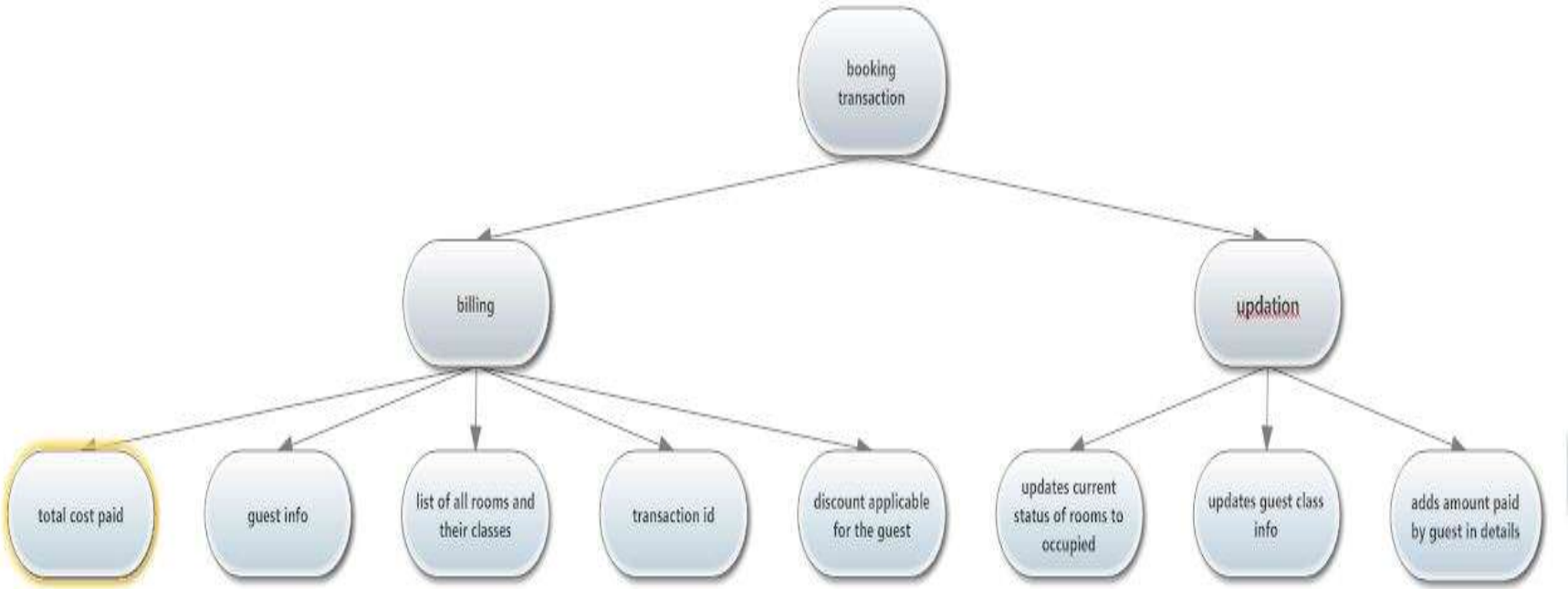
Booking of rooms

- The super-user's ability to edit the rooms and the corresponding process flow is hereby depicted.



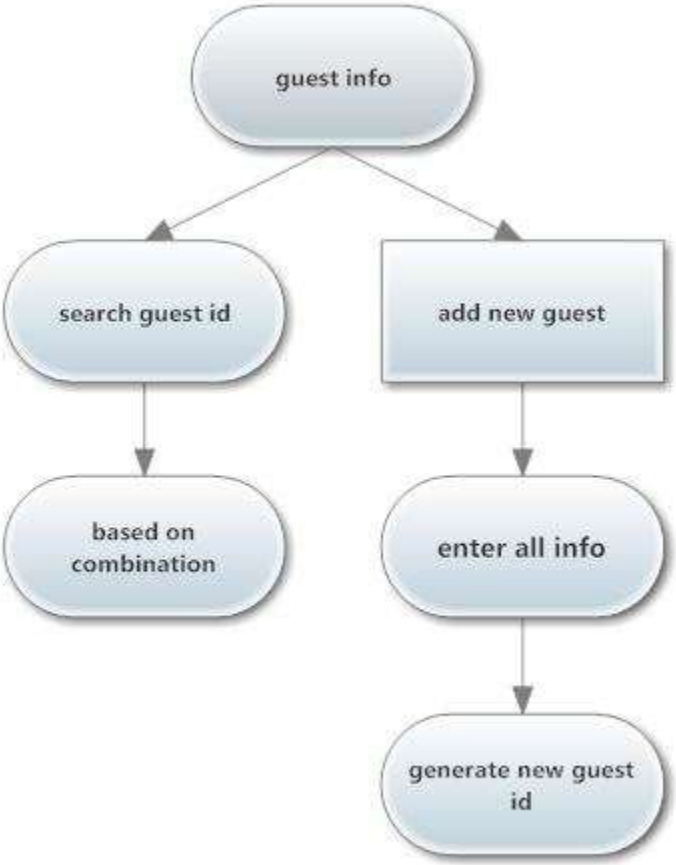
Rooms-info-edit procedures

P.T.O.

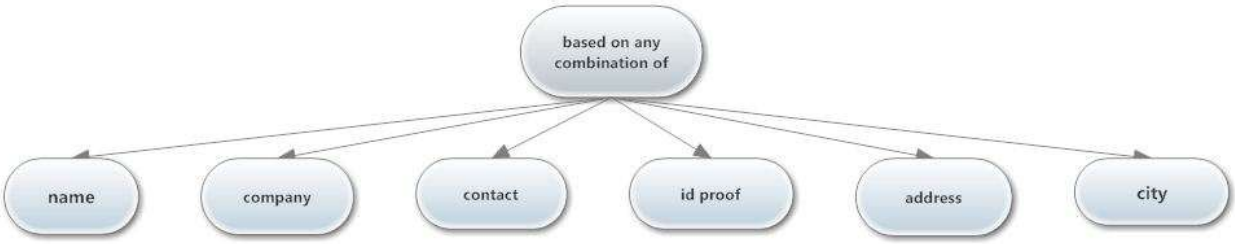


The booking and transaction flow of the algorithm to perform hotel registrations.

- Addition of the guests/previous customers.

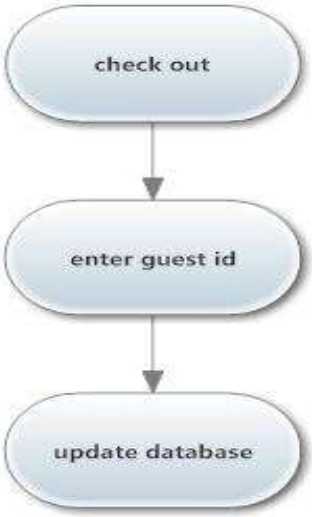


- Searches for the User Information from the database

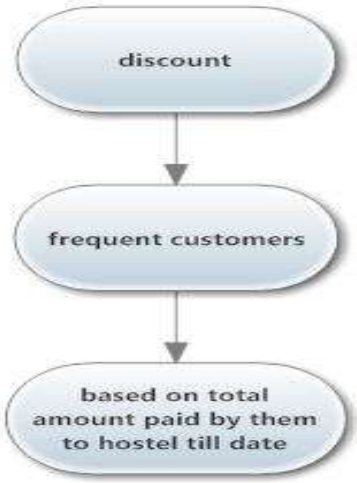


User-Searches

➢



➢ Discount awarding algorithm to the loyal users.



➢

➢ Forms :

After connecting to the database schema : hotel, and with the specifications of root as the 'username', and password 'hotel' at the port number 3306, we proceed into each of the tabular data access.

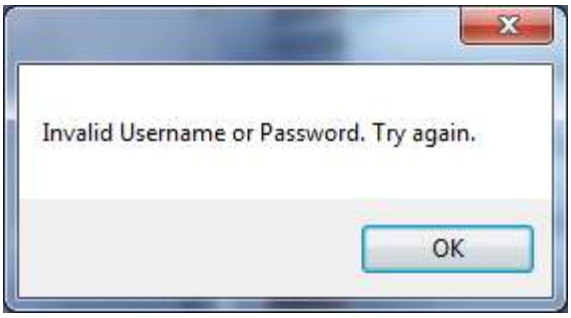
Log-In Window:

The main window that comes up would welcome the guest to the hotel registration system.

The existing user would then be entering the username and the password.

It is Form1.h -- the main form that allows a user to login using textbox as input fields and sends query to database and if they match, now depending on permissions of the person i.e. Super-user or a receptionist and hence accordingly their forms are opened. This uses a table named user with the columns username and password.

If the passwords and/or the username do not match with the records, the program comes with an error prompt.

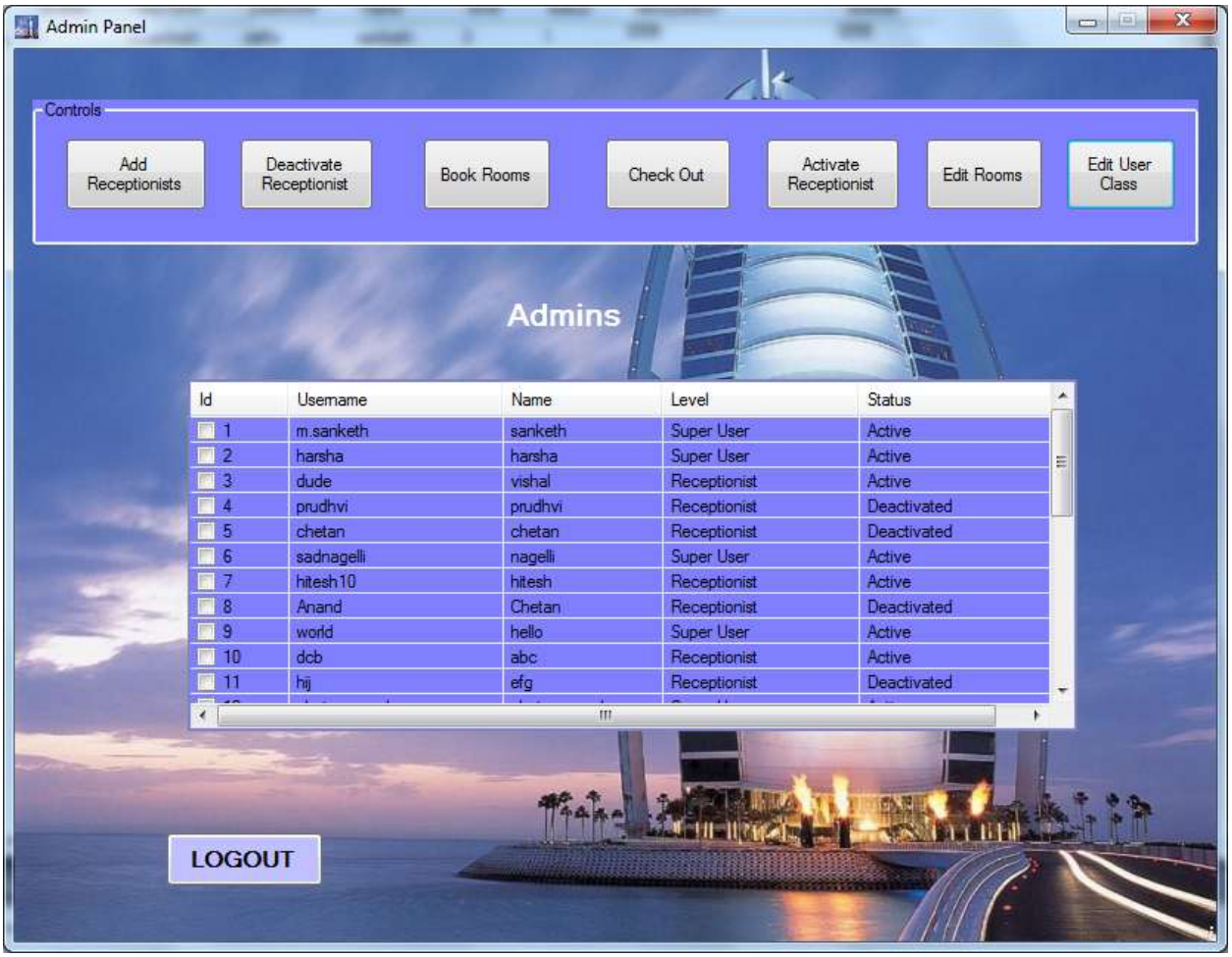


The user on seeing the error would be entering the correct data, and only then he/she would be given the permission to access the next page [Admin Panel].

There is also a “Forgot password” feature is there, where a user needs to answer the security question and then can set a new password. This also accesses the user named database, and it accesses the variables named “secquestions” and “answers” to validate the changing of the password of the user.

The Admin Panel:

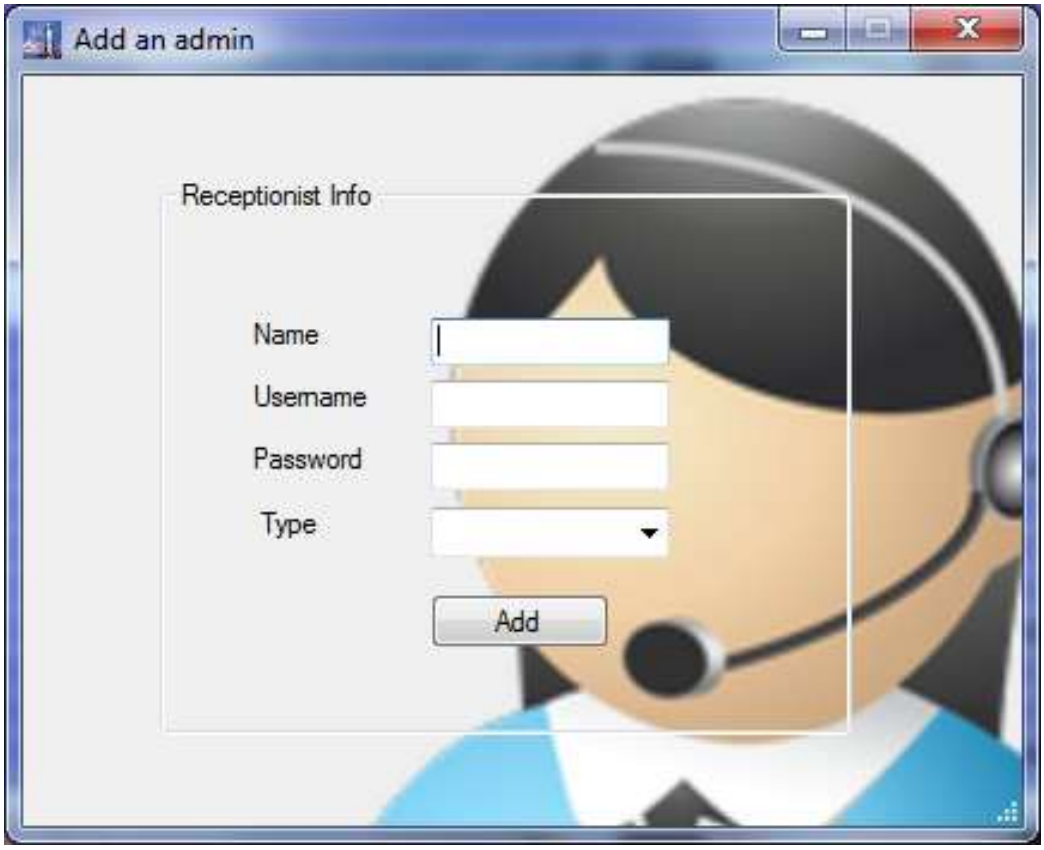
This page would then come up for the authenticated user :



The window coming up brings up the data fetched using an SQL query to the table user.

This page has the following features :

1. “Add administrators” : Add administrators [receptionists/super-users]



The administrators have to feed in the data of the new user to be added like :
“Name”, “Username”, “Password”, “Type”, “Security Question” and the “Security Key”.

These contact the table named user using SQL queries and feeds then into the table by the INSERT commands.

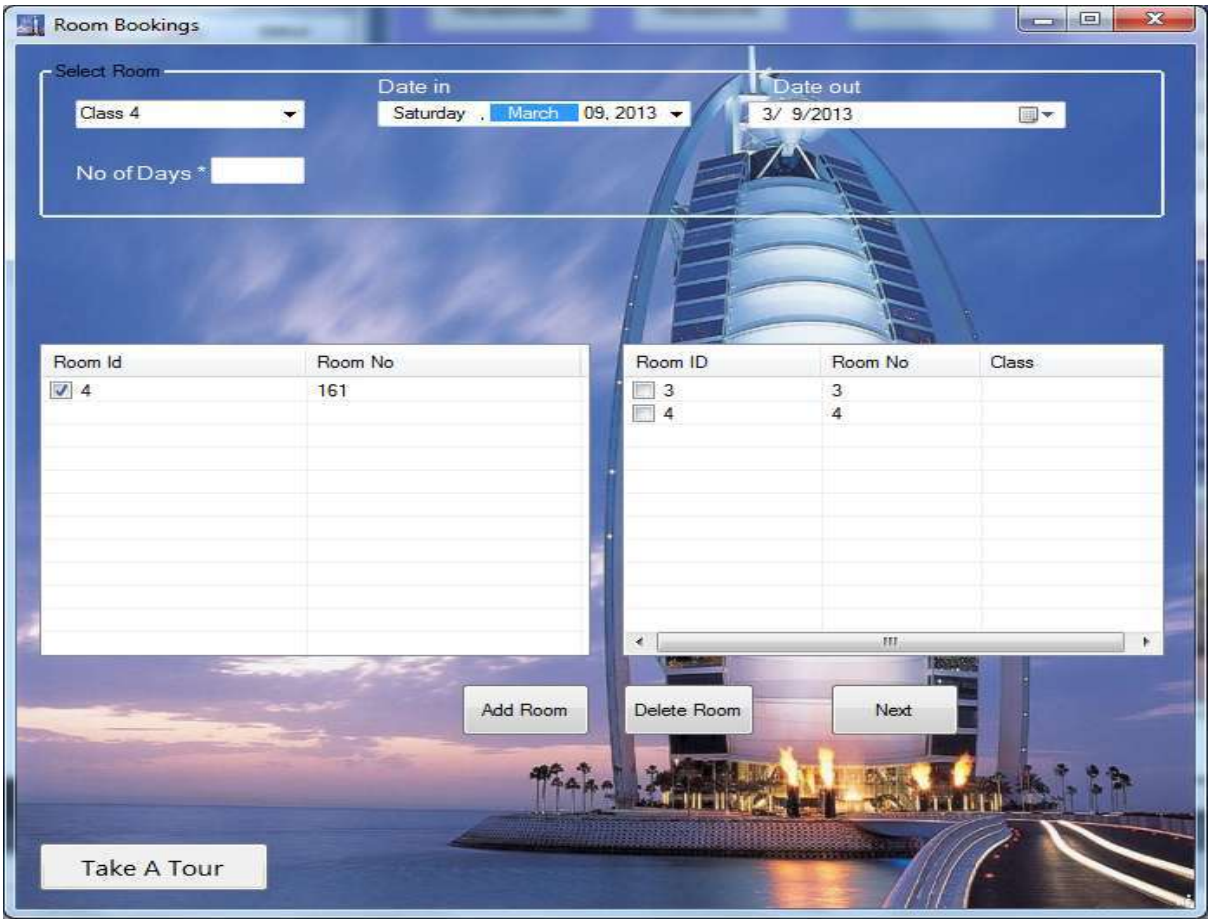
2. “Deactivate administrators”: Remove the authority of the current administrators. This would reflect the ability of the super-user to allot leaves to the employees, so that those users cannot make changes to the database of adding or manipulating the user details. In other words the other users cannot compromise and log into their accounts when they are not present on the premise. Also, when an admin is dismissed or retires, they are struck off the authorities, but the records contain their information for the back-tracking of the information corresponding to them.
But, the super user cannot be deleted by the super-user or the normal administrator, i.e. the receptionist [only by the coder, as the same power holding persons cannot eliminate their counterpart].

In this part of the software, the table would access and change the status named column from 1(active) to 0(de-activated).

3. “Book Rooms”: This would allow for the receptionist(s) and the super-user(s) to register a customer into the available rooms. This would display the information about the different classes of the rooms available and then, the user might ask the receptionist to select the rooms that they want to book, and then on clicking on the next button, the ensuing forms would come up. These are elaborated later on in the document.

Booking Rooms:

When a user arrives to the booking counter, the super-user would have the authority to book room(s)/suite(s) for the customer. The user would see the available the rooms under the varied classes, and would select those rooms. If, the user would by mistake put in some more rooms, or has entered the wrong room to be booked, the user would select the room in the right-box which would be fed into the system. Then, the user can remove them from the list of to-be-booked and billed rooms.



Features:

(i). Selects a class from all available suites/rooms

(ii). Selects the date in and the date out, and the corresponding number of days of stay is fed by the receptionist or the super-user. Then, list of all available rooms of that class are automatically displayed on the left hand pane (more specifically: list-view).

(iii). Clicking on the button for “Add room(s)” comes up with the rooms on the right pane, after we check the check-boxes for the rooms of the users’ choice.

(iv). In case of erroneous entries into the panes, that comes up for the erroneous clicking on the check-boxes, we have a provision of deleting the rooms from the selections.

Limitations:

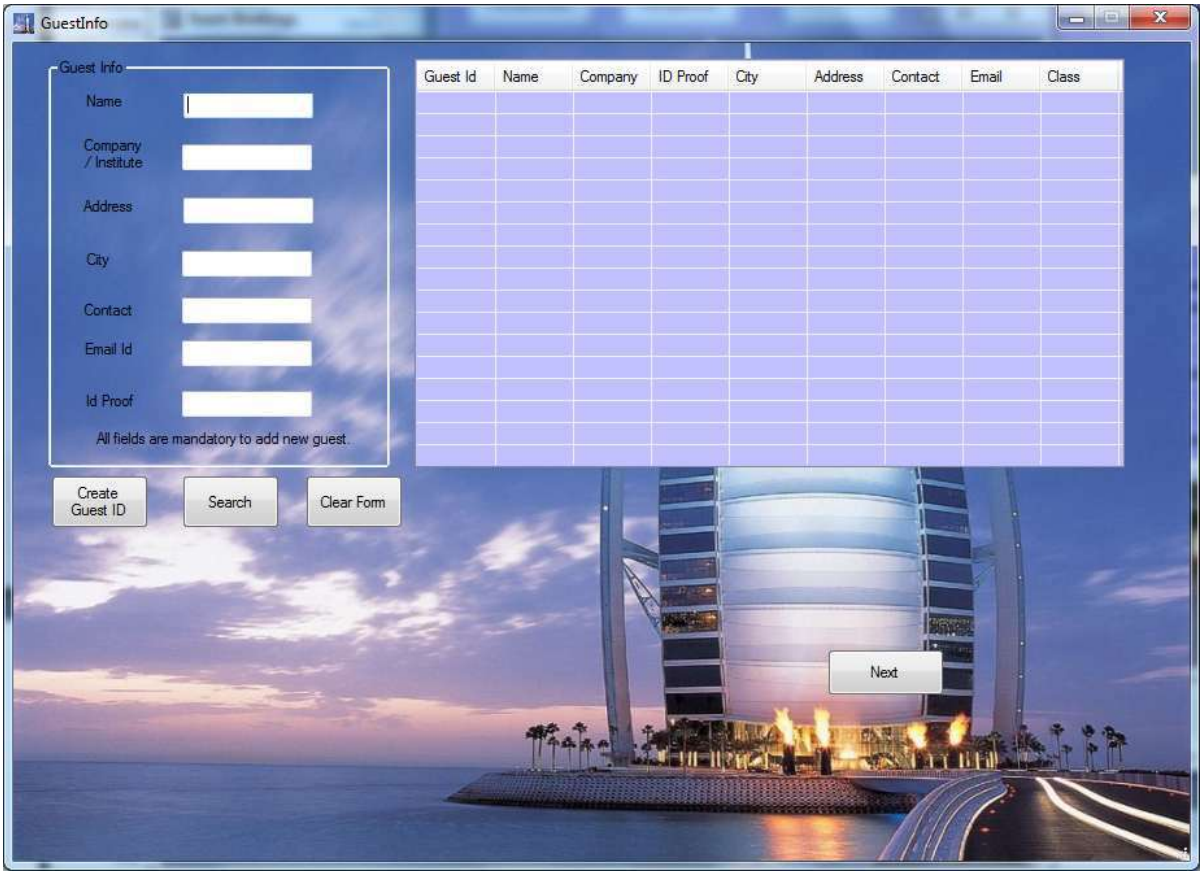
(i). As of now, we have not yet added the checking of the dates that are entered by the end-on user
(ii). The number of days of stay is not automatically calculated based on the entries by the user.

In case the user does not select a room and attempts to perform some action, there would be a prompt to select a room first and only then would he/she would be allowed to proceed.

Continued....

Guest Information Entry:

As soon as the user finally decides the rooms he/she would like to book; then, on pressing the next button, the following window pops-up.



(I). In case the user has visited the hotel beforehand, he/she can click on the “Search” button and all the users would get displayed on the right-placed list-view table.
It has some additional features:
1. If the user just clicks on the search after putting in some data in the “Name” field, then all the entries having the entries with that substring would come up in the table.
2. In case the user puts in some additional information in the “Address” and/or “City” and/or “Contact” and/or “Email Id” and/or “Id Proof”, all the entries having some substring of the Address, and the exact value of the City, Contact, Email Id, Id Proof would come up in the display box alongside.

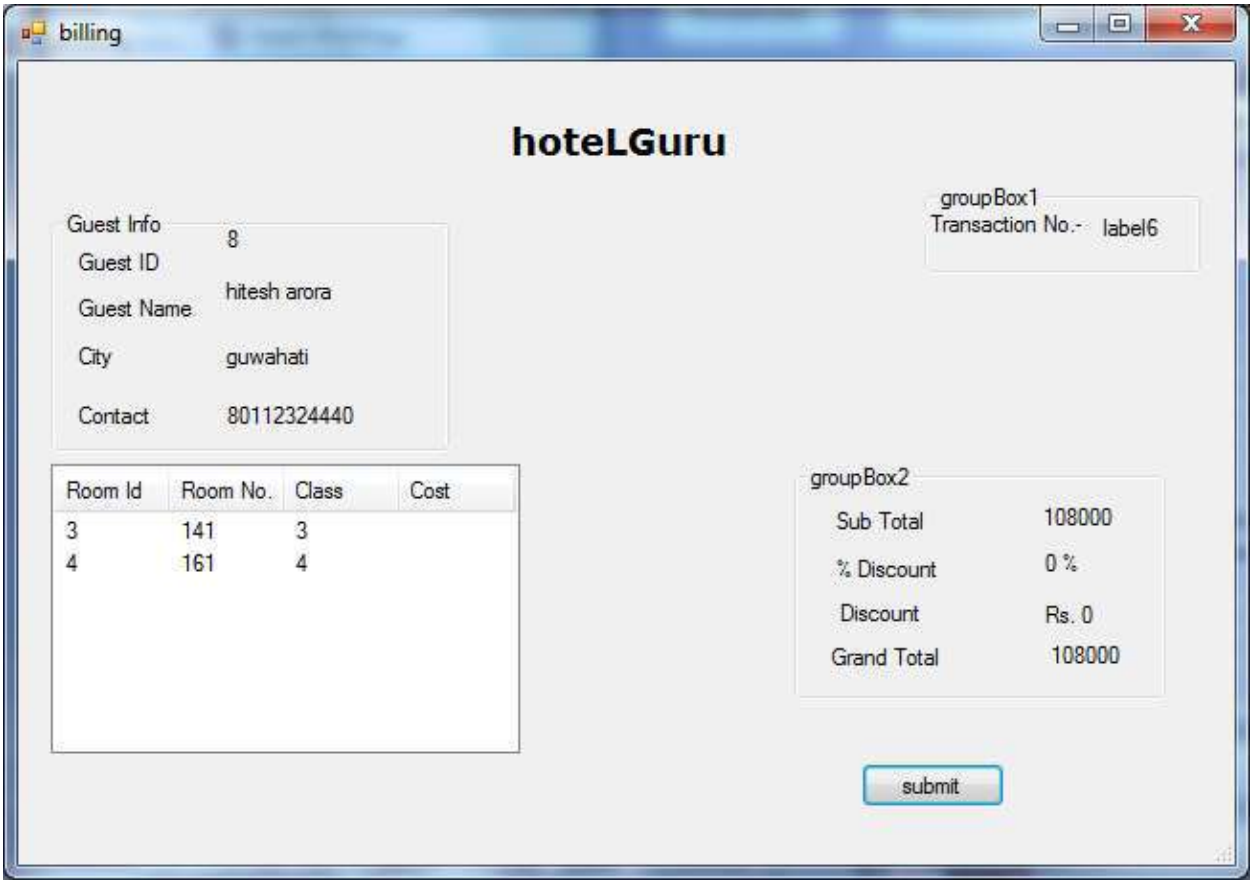
Now, the user can select on any of the user’s details and the data corresponding to the user would get filled into the text-boxes and then the user can proceed further into the check-in system.

(II). In case the user has never visited the hotel earlier, then he/she can click on the “Create Guest ID” after entering the information about the self, viz. “Name”, “Company/Institute”, “Address”, “City”,

“Contact”, “Contact”, “Email Id” and the “Id Proof”. The form ensures that no text-input is left empty before submission; else there comes an error-prompt.

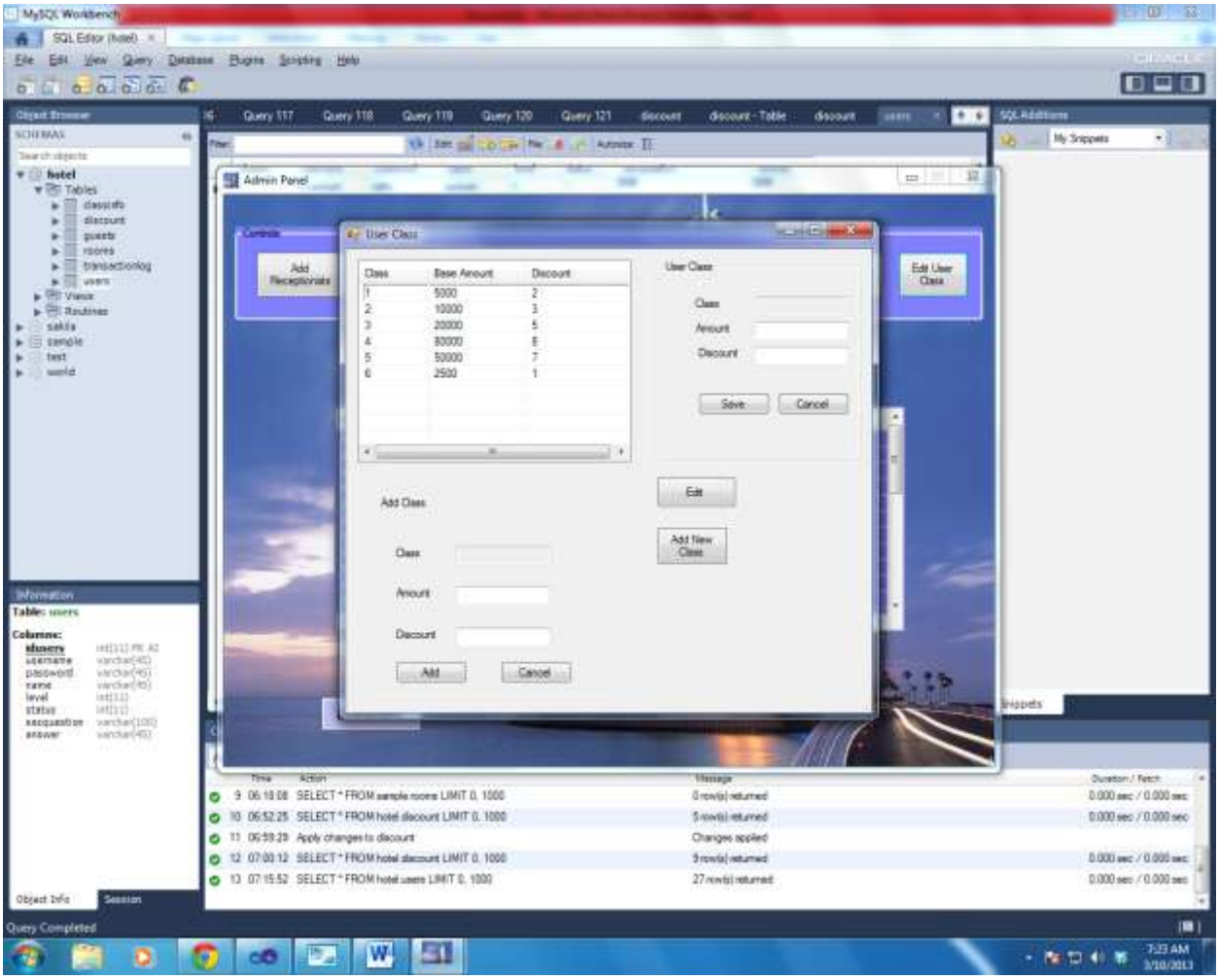
Billing:

This form accesses the table discount table. And only after the user clicks on submit, does the software put into the database this collected information of the user.



Discount Calculation:

This is a very robust feature of the software, as it does not allow the super-user to make non-realistic distribution of the discounts over its loyal users. This form works on these guidelines:



1. The admin can only effect upon a change, if the value of the class's discount is in proportionate view of the lesser and greater classes, i.e. the class with more value of money, cannot have lesser discount and vice-versa.
2. When the user finally comes up with a valid update, the cost of the class and the discount corresponding to it would be displayed after refreshing the list-view.

Database System & Hierarchy:

The information would be divided into access hierarchy and the hotel-owner would have the supreme right to see the data and would be given complete control after the software would be deployed for their usage. The receptionist would be given the freedom to just key in the new user's data and see the

history of the user to ensure proper discount system's implementation. The end-on user would be only able to enter the new information. These will be the three levels of database access specifications.

Database Schema (Data Modelling)

The basic structure/schema of our database has been shown pictorially.

TABLES:

1.users

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idusers	INT(11)	☑	☑	☑	☐	☐	☐	☑	
username	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
password	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
name	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
level	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
status	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
secquestion	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	NULL
answer	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	

idusers	username	password	name	level	status	secquestion	answer
1	m.sanketh	daffa	sanketh	0	1	NULL	NULL
2	harsha	3	harsha	0	1	Who am I?	me
3	dude	j	vishal	1	1	NULL	NULL
4	prudhvi	prudhvi	prudhvi	1	1	NULL	NULL
5	chetan	daffa	chetan	1	0	NULL	NULL
6	sadnagelli	kojja	nagelli	0	1	NULL	NULL
7	hitesh10	ghissmaxx	hitesh	1	1	NULL	NULL
8	Anand	2	Chetan	1	0	NULL	NULL
9	world	1	hello	0	1	NULL	NULL
10	dcb	123	abc	1	1	NULL	NULL
11	hij	123	efg	1	0	NULL	NULL
12	chetananand	rock	chetananand	0	1	NULL	NULL
13	daffa	jaffa	jaffa	1	0	NULL	NULL
14	chetan123	jaffa	chetan	0	1	NULL	NULL
15	hunny1	1	hunny	0	1	NULL	NULL
16	cholad	bhargav	chola	1	1	NULL	NULL
17	gsad	dasjg	kartik	1	1	NULL	NULL
18	sads	sdad	hisd	1	1	NULL	NULL
19	daffa123	123	jaffa	1	1	What is your favourite color?	black
20	harsha1	12	harsha	0	1	What is your favourite color?	blue

2.Rooms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idrooms	INT(11)	☑	☑	☑	☐	☐	☐	☑	
roomno	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
class	INT(11)	☐	☑	☐	☐	☐	☐	☐	
status	INT(11)	☐	☑	☐	☐	☐	☐	☐	
datein	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
dateout	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
guestid	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	

idrooms	roomno	class	status	datein	dateout	guestid
1	101	1	0	Sunday, March 10, 2013	Sunday, March 10, 2013	7
2	102	1	0	Sunday, March 10, 2013	Sunday, March 10, 2013	7
3	103	1	0	Sunday, March 10, 2013	Monday, March 11, 2013	1
4	104	1	0	Sunday, March 10, 2013	Monday, March 11, 2013	1

3.Guests

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idguests	INT(11)	☑	☑	☑	☐	☐	☐	☑	
name	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	NULL
contact	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	NULL
idproof	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
company	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	'none'
city	VARCHAR(50)	☐	☐	☐	☐	☐	☐	☐	NULL
email	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
address	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	NULL
class	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
amountpaid	INT(11)	☐	☐	☐	☐	☐	☐	☐	'0'
		☐	☐	☐	☐	☐	☐	☐	

idguests	name	contact	idproof	company	city	email	address	class	amountpaid
1	hitesh	8011232440	122	iitg	guwahati	hites@iitg.emet.in	manas	1	4000
2	chetan	9985654566	118	iitg	assam	anand@ya.com	kapili	1	0
3	sanketh	4565456	156	iitb	guwa	mopuru@fb.com	barak	NULL	1000
4	vishal	565665	186	mit	delhi	anad@ds.com	dihing	NULL	0
5	kartik	990198266	856	google	califomia	k@gmail.com	dihing	NULL	8000
6	harsha	99985623	232	fb	us	hh@fb.com	umiam	NULL	5000
7	hitesh arora	9990198244	H56	stanford	gurgaon	hunny@yo.com	manas	NULL	10000
8	chetan anand	9963256456	4545	steel	gurgaon	ana@hk.com	dihing	NULL	0

4.Discount

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
class	INT(11)	☐	☑	☐	☐	☐	☐	☐	
classamount	INT(11)	☐	☑	☐	☐	☐	☐	☐	
discount	INT(11)	☐	☑	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	

id	class	classamount	discount
1	1	5000	2
2	2	10000	3
3	3	20000	5
4	4	80000	8
5	5	50000	7
10	6	2500	1

5.Classinfo

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idclassinfo	INT(11)	☑	☑	☑	☐	☐	☐	☑	
type	INT(11)	☐	☑	☐	☐	☐	☐	☐	
cost	INT(11)	☐	☑	☐	☐	☐	☐	☐	
description	VARCHAR(100)	☐	☑	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	

idclassinfo	type	cost	description
1	1	1000	standard single
2	2	2000	deluxe single
3	3	3000	executive single
4	4	6000	standard double
5	5	5000	deluxe double
6	6	8000	executive double
7	7	10000	superior deluxe
8	8	12000	superior double
9	9	15000	ashiana

6.transaction log table.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
transactionid	INT(11)	☑	☑	☐	☐	☐	☐	☑	
date_booked	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
time	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
roomid	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
guestid	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
date_out	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
totalpaidbyguest	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
employeeid	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	

transactionid	date_booked	time	roomid	guestid	date_out	totalpaidbyguest	emplc
1	Wednesday, March 20, 2013		1	3	3/19/2013	19600	
2	Wednesday, March 20, 2013		7	3	3/19/2013	19600	
3	Wednesday, March 20, 2013		3	3	3/19/2013	19600	
4	Tuesday, March 12, 2013		7	15	3/20/2013	40000	
5	Tuesday, March 12, 2013		3	15	3/20/2013	40000	
6	Saturday, March 09, 2013		8	10	3/9/2013	12000	2
7	Saturday, March 09, 2013	1362892126	5	9	Saturday, March 09, 2013	5000	2
8	Tuesday, March 12, 2013	1362912412	1	1	Thursday, March 14, 2013	37200	2
9	Tuesday, March 12, 2013	1362912412	7	1	Thursday, March 14, 2013	37200	2
10	Tuesday, March 12, 2013	1362912412	4	1	Thursday, March 14, 2013	37200	2
11	Sunday, March 10, 2013	1362912090	1	7	Sunday, March 10, 2013	3880	2
12	Sunday, March 10, 2013	1362912090	2	7	Sunday, March 10, 2013	3880	2
13	Sunday, March 10, 2013	1362912264	21	2	Wednesday, March 13, 2013	18000	2

Special Features:

First Run:

We do not require any setup file or installer.
The executable file does everything. We have written code for creating a database tables and are providing CSV files too for easier checking of

software. We have created basic structure of hotel in our CSV files.You may import it the database or you could make ur CSV files depending on your HOTEL rooms structure. We have given this flexibility so that it can be run in any hotel.

Search_Users:

This is a very strong feature of the software that we have developed: “hoteLGuru”. It allows a lot of flexibility on part of the user to search for the customers as per each of the different attributes of the customer information.

Limitations and Bugs:

1. The text boxes that are supposed to have only the integer as inputs, are actually taking texts as inputs as well.
2. Display the current status of all the rooms.
3. We cannot back-up the data in case of a power failure.
4. We have not enabled the safeguards against SQL injections.
5. MD5 hashing was planned to be done, but could not do.
6. Not taking information of multiple guests for same booking since we have assumed that booking will be done by head of family or a HR head.
7. We did not provide specific categories for ID proof.
8. Rooms cannot be added by GUI,since we made a very simple assumption that room will be fixed in a hotel and only their class or cost or description may change, that we provided.
   The admin can initially only add basic structure of rooms through CSV files only since it will be difficult to add all rooms manually.
9. Any Date from datepicker can be selected,we did add validation check.
10. Number of days for hotel stay is to be added manually by receptionist.

References/Links:

MSDN Website
Youtube VC++ tutorials