

Guest House Management System

Technical Documentation

A Graphical Utility for management of Guest Houses

CONTENTS

About the project	4
Introduction	4
Definitions, Acronyms and Abbreviations	4
Overview	5
Intended Users	5
Miscellaneous Details	6
Hardware	6
Software	6
Userbase	6
Limitations	6
Startup Page	7
Log In	7
Sign-Up	7
Admin	9
Access	9
1. Guest House Details	9
2. Staff Info	9
3. User Approval	10
4. Housekeeping	10
5. Add Room	11
6. Room Status	12
7. Leave Approval	12
8. Guest House Facilities	13
9. Customer Feedbacks	15
10. Generate bill	16

11. Booking Approval	17
12. Display Booking	18
13. Floor Map	18
Customer	19
Access	19
1. New Booking	19
2. Guest House Facilities	20
3. My Bookings	21
4. House Keeping	22
5. Feedback	22
6. Logout	23
Staff	24
1. Register	24
2. Leave Request	24
3. Edit Records	26
4. Planner	27

ABOUT THE PROJECT

Introduction

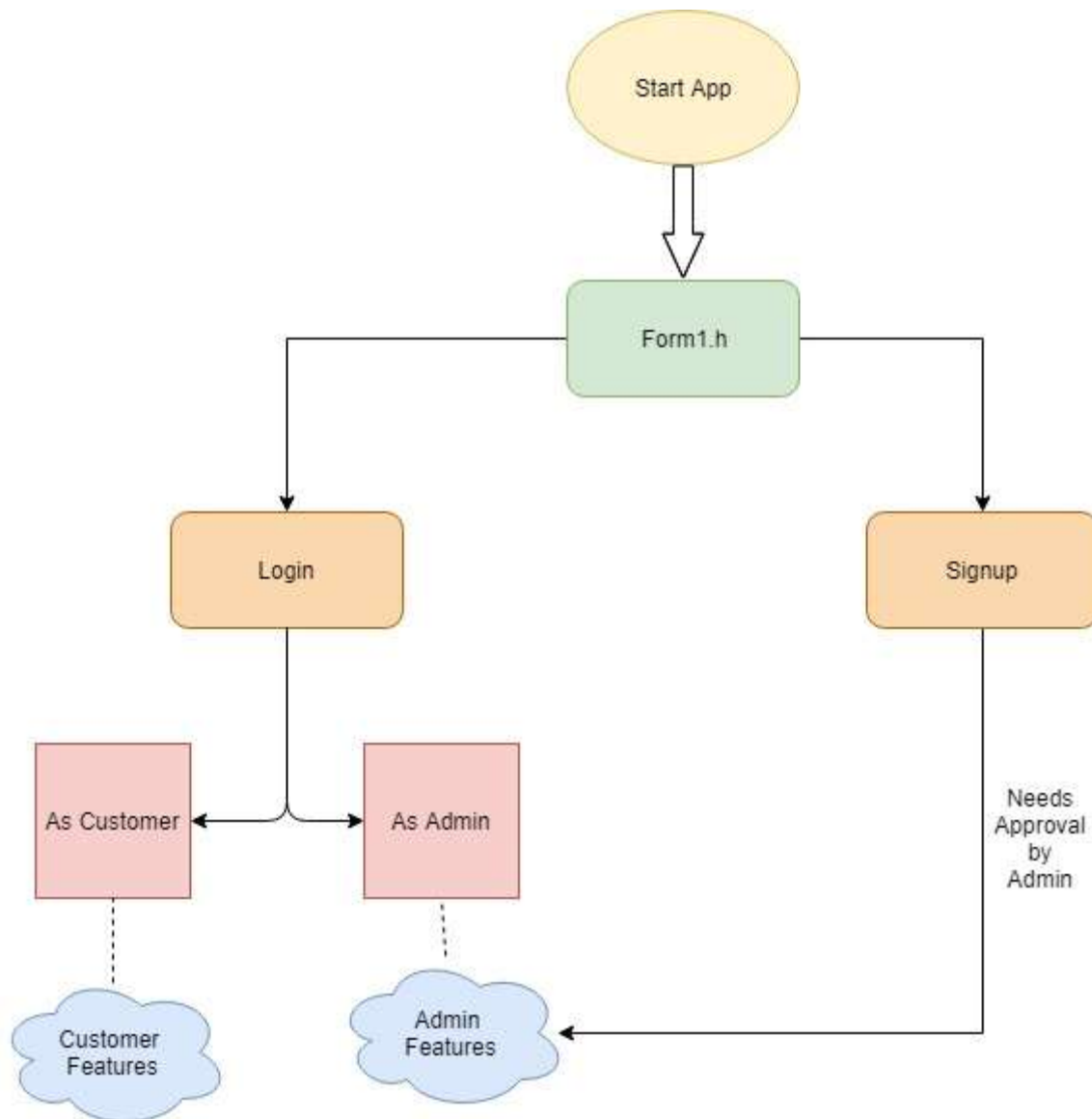
This project was developed as a part of CS243 System Programming Lab to demonstrate the working of a Guest House management system.

The Project is made in Visual C++ with the aid of Visual Studio 2010. A Microsoft Access Database is used, and MS ACE OLEDB engine used to access this database. The program is divided into sections, each of which contains a series of 'User Controls'.

Definitions, Acronyms and Abbreviations

- App/Application/Project/Programme and any synonym refers to the application developed in this Assignment and all statements talk in the context of the same
- Features(s) refer to the functions/forms added to the application
- The application was developed using Visual C++ and keywords and terms relating to it carry the same meaning here
- Database talks in reference to the MS Access database that is used in the application and words/statements relating to MS Access carry the same meaning here
- Knowledge of computer related terms like Login, Sign-up, administrator etc is assumed

Overview



Intended Users

Anyone willing to Study, Evaluate, Use, Modify, Deploy or work with the application in any way is free to use this document.

MISCALLENOUS DETAILS

The use and development of the application needs some requirements to be fulfilled

Hardware

- To run the built application:

Any machine with 500 MB of storage, any modern CPU, a display with resolution of 1280 x 800 for output, keyboard and a pointing device (or touchscreen) for input

- For development:

Any PC that satisfies the system requirements of Visual Studio 2010

Software

- To run the built application:

A PC with windows 7 or later capable of running C++ compiled binary files

- For development:

Visual Studio 2010 or later with its software requirements satisfied

Userbase

This application is intended for use by small scale guest houses. The admin panel is for the receptionist/booking desk to access all user requests. The users can access their own account through work PCs in the guest house.

Limitations

The current app is intended for small scale use hence it lacks features like accessing and booking over internet or sending confirmation mails. This can be fixed by making proper improvements

STARTUP PAGE

Header file: Form1.h

This allows the user to move on to login or Sign-Up

Function Descriptions:

Name	Arguments	Description	Output
Btn_Login_Click	System Arguments	Open the login form	Void
Btn_Signup_Click	System Arguments	Opens the signup form	Void

Log In

Header file: UserControl_Login.h

This allows to sign in as a particular user, the user type is taken from the database

Function Descriptions:

Name	Arguments	Description	Output
Btn_Forget_Click	System Arguments	To be used to recover the account	Void
Btn_submit_Click	System Arguments	Accesses the DB to validate the user info and results in appropriate action	Void

Sign-Up

Header file: UserControl_Signup.h

This functionality allows to send a signup application to the admin

It requires the applicant to fill out the following data

1. First Name

2. Last Name
3. Email ID
4. Password
5. Password confirmation
6. Username
7. Category (Selected from the drop-down menu)

Function Descriptions:

Name	Arguments	Description	Output
Btn_Create_Click	System Arguments	Validate the inputs and sends application to admin	Void

NOTE:

For ease of use as well as testing, user accounts of each type has been preloaded into the database. Each of the following sections contain those username and passwords

ADMIN

Access

Header File: `Form_Reception.h`

The admin page can be accessed by using the following credentials in the login form

Username: admin

Password: admin

The admin page comes with the following functionalities, each of which are accessible through corresponding buttons

1. Guest House Details

Section for admin to change Guest House info related to – Name, Address, Contact, Email, User Types, Room types

- Section for admin to change Guest House info related to – Name, Address, Contact, Email, User Types, Room types
- Header file: `UserControl_Guest_House_info.h`
- Contains:
 - 4 x textBoxes
 - 3 x buttons
 - User Types – Opens a User Control with a Textbox to add new user types
 - Room types – Opens a User Control with a Textbox to add new room type (AC/ Non-AC, etc)
 - Submit – Commits all data to table 'Guest_House_Info' in DB
- Details taken - Name, Address, Contact and Email of Guest House

2. Staff Info

Go to page 21

3. User Approval

This section is for Admin to approve Valid User registrations.

- Header file: `UserControl_Approve_Users.h`
- Contains:
 - Dynamically allocated textboxes
 - Dynamically allocated buttons to approve or disapprove.
 - Access 'User_Table' in DB and fetch user details of entries where 'Approved' column is "NO"

Function Description:

Name	Arguments	Description	Output
<code>UserControl_Approve_Users_Load</code>	System Args	Access 'User_Table' In DB and dynamically allocate textboxes and buttons.	Void

4. Housekeeping

This section contains the housekeeping requests made by the customer

- Header file:
`UserControl_Staff_Housekeeping_Approval.h`
- Contains:
 - Text Box + Button pairs for every request

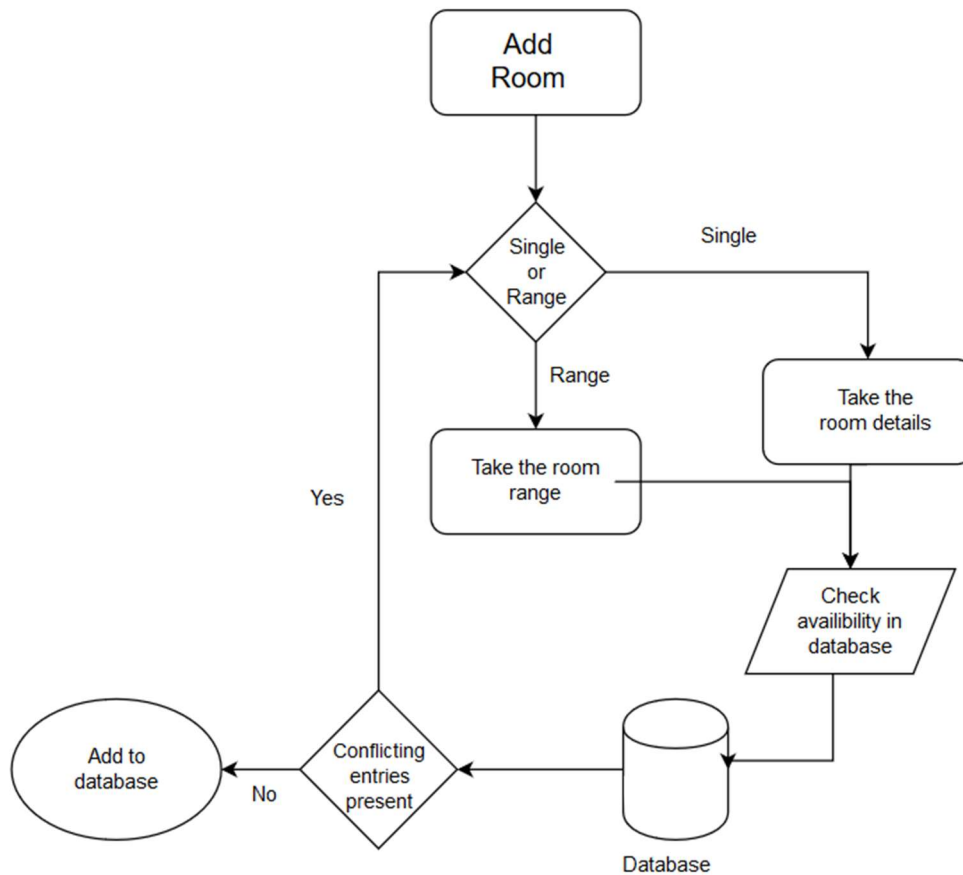
Function Description:

Name	Arguments	Description	Output
<code>UserControl_Staff_Housekeeping_Approval_Load</code>	System Arguments	Loads the textbox+button pair for every housekeeping request	Void
<code>Finish_Button_Click</code>	System Arguments	Marks the housekeeping request as finished	Void

5. Add Room

This allows the admin to add rooms to the guest house record

- Header file: UserControl_Add_Room.h
- Contains:
 - 3 x textboxes
 - Floor
 - Room starting/Room No.
 - Room ending (only when adding range)
 - 1 x combobox
 - 3x buttons
 - Individual Add
 - Add Range
 - Add to database



Function Description:

Name	Arguments	Description	Output
Btn_Ind_Click	System Args	Gives option to add individual room to	Void
Btn_Range_Click	System Args	Helps to add a room range to the db	Void
Btn_Add_Room_Click	System Args	Adds the data in form to the db	Void

6. Room Status

Allows the admin to change the status of any room

- Header file
- Contains
 - 3 x Combobox
 - Floor
 - Room No.
 - Status
 - 1 x button
- Function Descriptions:

Name	Arguments	Description	Output
Btn_Status_Change_Click	System Args	Changes the room status in the db	Void

○

7. Leave Approval

This section allows the admin to approve or disapprove pending Leave requests by staff members.

- Header file: `UserControl_Staff_Leave_Approval.h`
- Contains:
 - 1 x datagridView
 - 7 x columns
 - 1 x Reload button.

Function Descriptions:

Name	Arguments	Description	Output
<code>loadDataGrid</code>	None	Access Db and get necessary data.	Void
<code>dataGridView1_CellContentClick</code>	System Arguments	Opens a message Box for approval or disapproval	Void

8. Guest House Facilities

This section is for admin to add information about the Guest House that will be viewed by customers.

- Header file: `UserControl_GuestHouseDetails.h`
- Contains:
 - 4 buttons –
 1. Add Cabs-
 - Header file: `UserControl_Add_Cab_Details.h`
 - 4 Textboxes and 1 multiline textbox
 - Validate data and save into 'Cab_details' in DB
 - Details taken:
 - Driver Name
 - Vehicle Name
 - Vehicle Number
 - Vehicle Description
 2. Add Restaurant-
 - Header file: `UserControl_Add_Resturant.h`

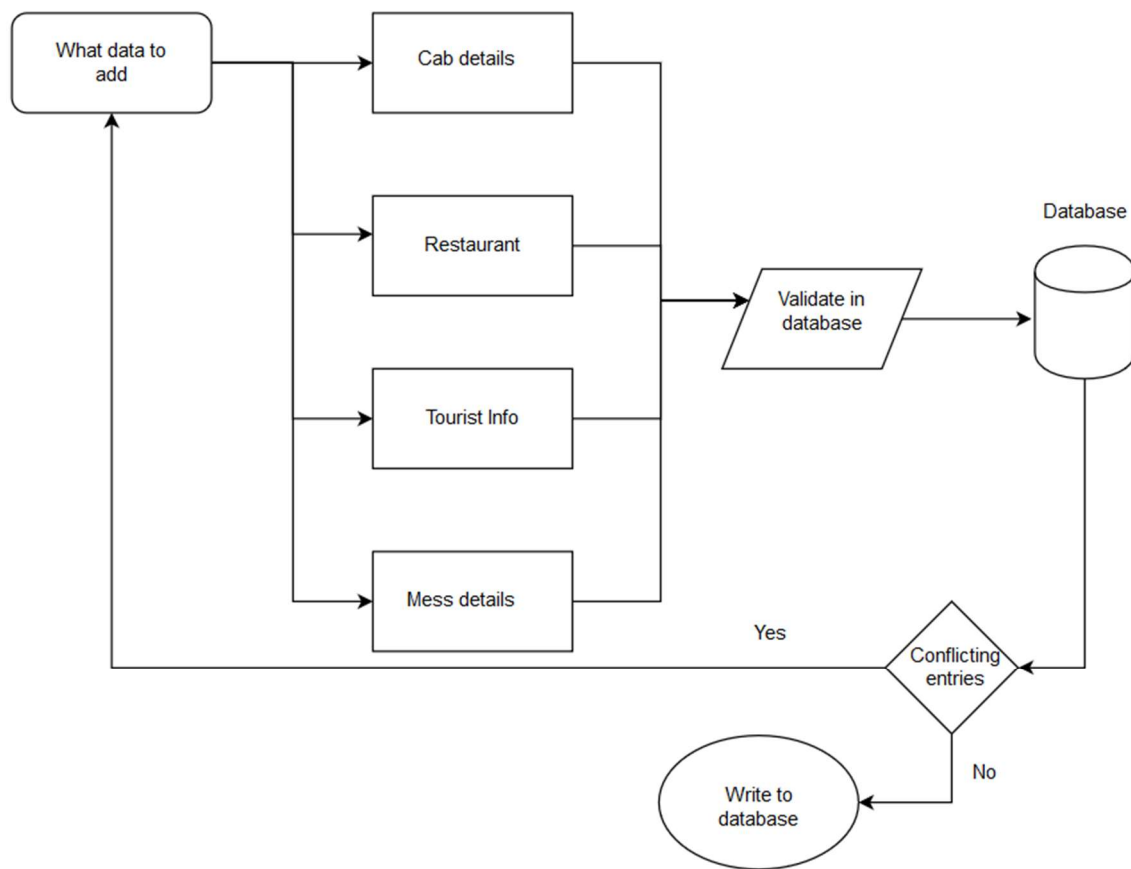
- Here admin adds the restaurants trusted but the Guest House for the users to see. This contains
 - o Restaurant name
 - o Restaurant Address
 - o Delivery facilities to the guest house
 - o Contact number

3. Add Hotspot/Tourist Place –

- Header file: `UserControl_Add_Cab_Places.h`
- Here admin adds various tourist places in the vicinity of the Guest House which will be suggested to the customers
 - o Place name
 - o Distance from guest house
 - o Mode of transportation
 - o Important details
- Data saved to 'Places_To_Visit' in the db

4. Add Mess Info –

- Header file: `UserControl_Add_Mess.h`
- 3 buttons
- One to add mess picture to pictureBox and then to image directory as mess.jpeg
- Another to open 3 TextBoxes for entries of Mess food rates of Breakfast, Lunch, Dinner, saved into `Mess_Rates` table of db.



9. Customer Feedbacks

This section views reviews given by Guests.

- Header file: `UserControl_Customer_Feedback.h`
- Contains-
 - 3 Viewing options (buttons) – All, Unread, Read
 - Dynamically allocated Textboxes with Feedback details

Function Description:

Name	Arguments	Description	Output
<code>markAsRead_button_click</code>	System Args	Update all checked Feedback status to 'Read' on 'Booking_Request' table in db	Void

Btn_Read_Feedback_Click	System Args	Dynamically allocate textboxes to view feedbacks with status = Read	Void
Btn_Unread_Feedback_Click	System Args	Dynamically allocate textboxes to view feedbacks with status = Unread	Void
Btn_All_Feedback_Click	System Args	View all Feedbacks in 'Booking_Request' table in db	Void

10. Generate bill

View all rooms and their current status.

- Header file: UserControl_Floor_Map.h
- Contains:
 - Combobox to select floor
 - Dynamically allocated labels with room status

Function Description:

Name	Arguments	Description	Output
UserControl_Floor_Map_Load	System Args	Access 'Room_No' table and fill Cobobox , Txt_Floor with valid floor numbers. Make a vector of all Rooms that are booked on system date.	Void
Txt_Floor_SelectedIndexChanged	System Args	Access 'Room_No' table and view rooms of selected floor with current status.	Void

bookingDetailsOfRoom	System Args	View details of selected 'Booked' room by accessing 'Booking_Request' table in db	Void
----------------------	----------------	--	------

11. Booking Approval

This section provides admin with all pending booking requests

- Header file: `UserControl_Booking_Approval.h`
- Contains:
 - ComboBox of Room type
 - Dynamically allocate

Function Description:

Name	Arguments	Description	Output
<code>approve_button_click</code>	System Args	Allocate rooms from 'Room_No' table and update bookings table accordingly	Void
<code>cancel_button_click</code>	System Args	Selected Booking deleted from 'Booking_Request' in db	Void
<code>UserControl_Booking_Approval_Load</code>	System Args	Load valid entries into Txt_Room_Type ComboBox	Void
<code>Txt_Room_Type_SelectedIndexChanged</code>	System Args	View all Not approved requests from 'Booking_request' table from db and dynamically	Void

allocate textboxes and
Approve/Decline Buttons

12. Display Booking

Shows all the bookings

Header file: `UserControl_Staff_Booking_Display.h`

Contains:

- Variable number of textboxes showing bookings

Function Descriptions:

Name	Arguments	Description	Output
<code>UserControl_Staff_Booking_Display_Load</code>	System Args	Loads all bookings from the database into the display	Void

13. Floor Map

This shows a visual status of the selected floor as recorded in the database

- Header file: `UserControl_Staff_Booking_Display.h`
- Contains:
 - 1 combo box
 - 4 buttons (appear only when a floor is selected)
 - Variable Number of buttons (equal to the rooms on that floor)

Function Descriptions:

Name	Arguments	Description	Output
<code>UserControl_Floor_Map_Load</code>	System Arguments	Fetches and stores floors in the combobox	Void
<code>Txt_Floor_SelectedIndexChanged</code>	System Arguments	Shows the recorded status of the selected floor	Void

CUSTOMER

Access

Header File: `Form_Reception.h`

A test customer homepage can be accessed with following credentials

Username: q

Password: qqq

The section provides the following facilities to the customer

1. New Booking

This section is used by customer to request for a booking. These requests have to be approved by admin.

- Header File: `UserControl_Customer_Booking.h`
- Contains:
 - 4 Textboxes
 - 3 ComboBoxes
 - 2 dateTimePickers
 - 2 buttons
- Data Validation:
 1. Date of 'Booking From' is set as system date.
 2. Min Date of 'Booking Upto' is set as value of Booking From so that valid durations can only be filled
 3. Username of current user and User types are fetched from db.

Function Descriptions:

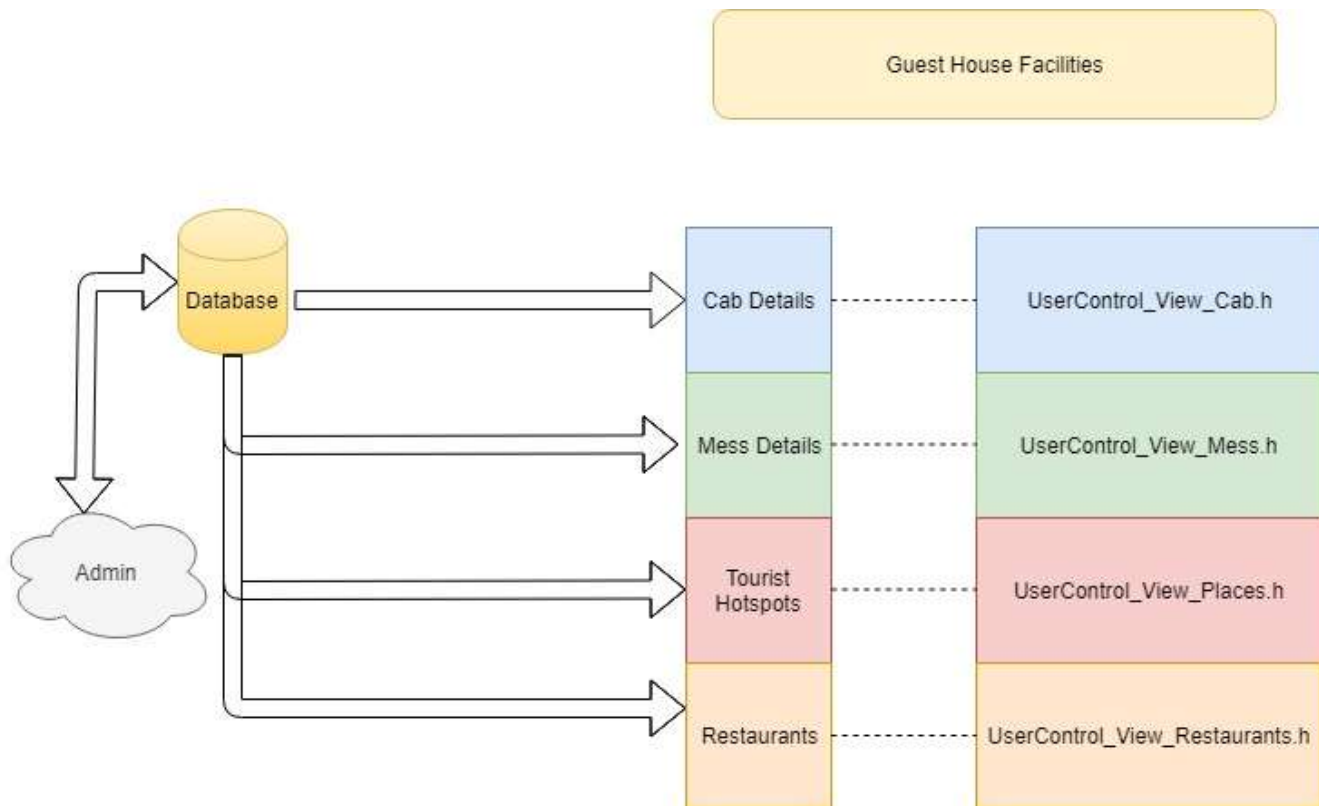
Name	Arguments	Description	Output
<code>UserControl_Customer_Booking_Load</code>	System Args	Access Db and get necessary data for Username, Room type, User types, contact and email	Void

<code>button1_Click</code>	System Arguments	Write user inputs into 'Booking_Request' table for approval by admin.	Void
<code>Btn_Room_Add_Click</code>	System Arguments	Add Textboxes on screen with different types of rooms and their counts selected by user.	Void

2. Guest House Facilities

The customer can use this section to view Details of cabs, Guest House Mess, Tourist Hotspots and restaurants nearby made available by admin in a section with the same name in Admin controls.

- Header File: `UserControl_In_Stay_Facilities.h`
- Contains:
 - 4 Buttons –
 1. Cab details – opening '`UserControl_View_Cab.h`'
 2. Mess Details – opening '`UserControl_View_Mess.h`'
 3. Tourist Hotspots– opening '`UserControl_View_Places.h`'
 4. Restaurants – opening '`UserControl_View_Restaurants.h`'



Function Description:

Name	Arguments	Description	Output
Button_Guest_House_Facilities_Click	System Args	Access the DB and show the guest house facilities	Void

3. My Bookings

This section helps customer view all past, current and future bookings.

- Header file: `UserControl_Customer_Book_Edit.h`
- Contains:
 - 4 x columns of buttons that lead to appropriate form.
- Pressing a past booking opens '`Form_Feedback.h`'
- Pressing current booking opens a form with bookings details.
- Pressing a future booking opens option to delete it.

Function Description:

Name	Arguments	Description	Output
Button_My_Bookings_Click	System Args	Access the DB and show the bookings made by user	Void

-

4. House Keeping

This section provides a number of housekeeping facilities to the customer.

- Header file: UserControl_Customer_Housekeeping.h
- Contains:
 - 1 x Button – Submit button to submit the form
 - 6 x checkboxes – They allow the customer to choose one or more housekeeping service-
 1. Bedsheets
 2. Room Cleaning
 3. Water Issues-Plumbing
 4. Laundry
 5. Electrical Issue
 6. Other Requests
 - 2 x textboxes –
 1. Room Number – Customer enters the room number for availing the housekeeping service
 2. Other Requests – (Hidden until the corresponding checkbox is selected) Customer can request a service not covered by the checkboxes

Function Descriptions

Name	Arguments	Description	Output
button1_Click	System Arguments	Sends the housekeeping request to the database	Void

5. Feedback

This provides a form for the user to give feedback.

- Header file: `Form_Feedback.h`
- Contains:
 - 15 x Picture box, to rate a service on a scale on 1 to 5 (for 3 services)–
 - Mess Rating
 - Cleanliness rating
 - Service rating
 - 1 x Textbox – for writing comments
 - 1 x button – to submit the feedback to the Guesthouse

Function Description:

Name	Arguments	Description	Output
<code>Btn_Create_Click</code>	System Arguments	Writes the feedback to the database	Void

6. Logout

Provides a way to move back to the initial form (The Login/Signup page)

Name	Arguments	Description	Output
<code>Button_Logout_Click</code>	System Arguments	Closes the current form and restarts the application	Void

STAFF

The admin has controls that can aid the management and working of staff in the guest house. It provides the admin with the following features

1. Register

New staff members can be registered in this section.

- Header file: `UserControl_Staff_Register.h`
- Contains:
 - 5 x textboxes
 - 2 x comboboxes
 - 2 x dateTimePickers
- All fields except email are compulsory.
- Data is written into table 'Staff_Register' in GuestHouse.accdb.
- Data Validation:
 - No field can be left empty (except email).
 - Employee name can contain only English alphabets.
 - Length of all fields limited according to db¹ limitations to avoid errors.
 - 'Contact Number' must contain only digits of length 10.
 - 'Date of birth' set according to minimum age of employment as 18yo.
 - 'Date of Joining' is fixed as system date.
- Using OleDb connection to access DB.

2. Leave Request

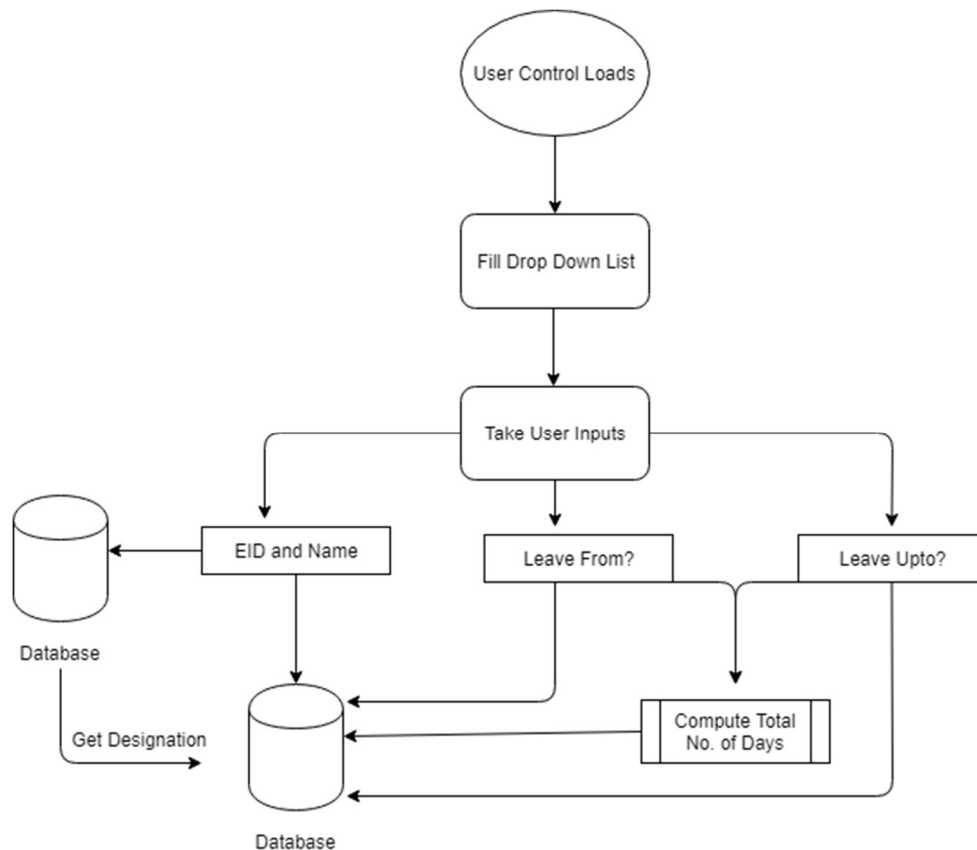
Staff members can apply for leave using this section.

- Header file: `UserControl_Staff_Leave.h`
- Contains:
 - A drop-down list with unique EID² and Employee Name
 - Two dateTimePicker controls

¹ Data Base, here 'GuestHouse.accdb'

² EID = Employee ID given at the time of registration.

- Leave requests are stored in 'Staff_Leave' table in GuestHouse.accdb database.
- Data Validation:
 - ➔ DropDown contains only valid EIDs and Names.
 - ➔ dateTimePickers have minDate as system's date so only future requests can be made.
 - ➔ dateTimePicker2 has minDate as that selected in 1.



Function Descriptions:

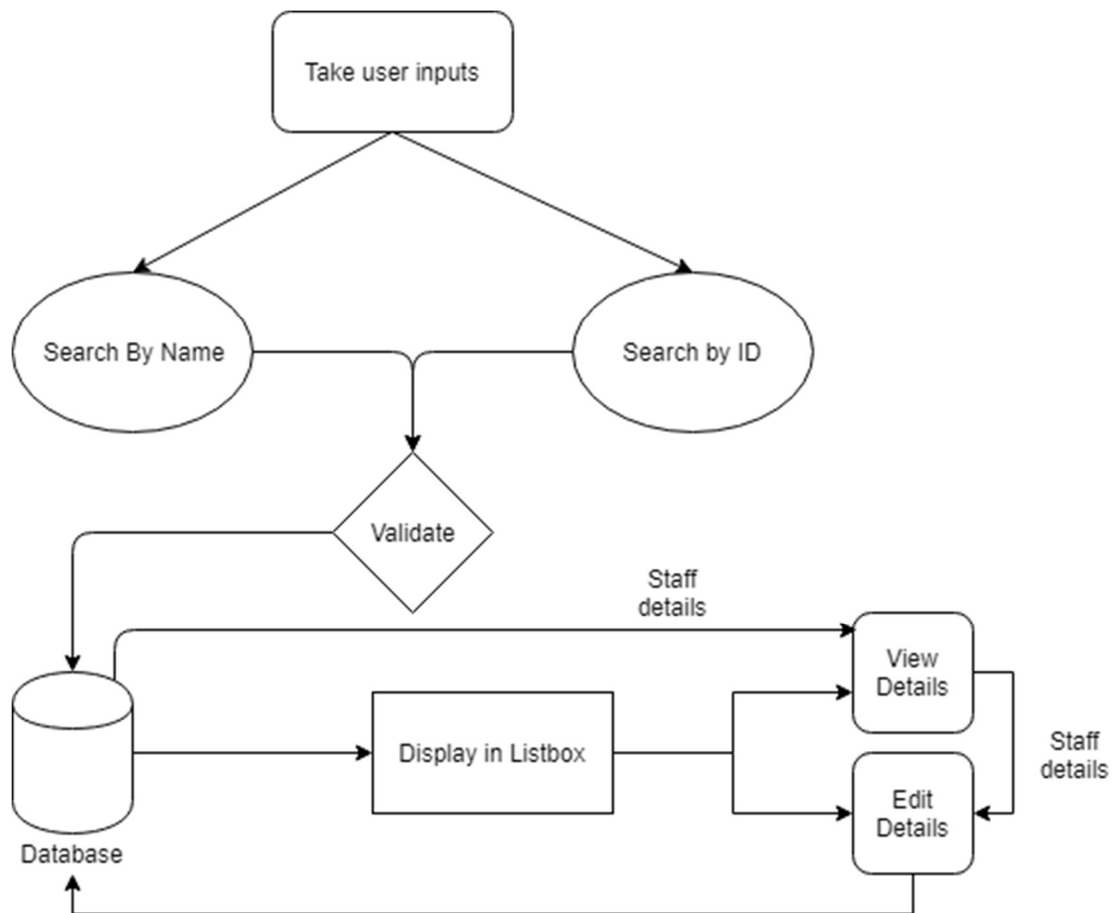
Name	Arguments	Description	Output
Fillcombo	None	Access Db for valid EID and Names	Void
vailidDates	None	Returns true if date entries are correct.	Boolean
UserControl_Staff_Leave_Load	System Arguments	Call Fillcombo and set minDate to systemDate	Void
btn_submit_Click	System Arguments	Access db and get designation using EID. Calculate totalDays and write a new entry to db table Staff_Leave.	Void

<code>dateTimePicker1_ValueChanged</code>	System Arguments	Set minDate of 'Leave Upto' to that selected in 'Leave from'	Void
---	---------------------	---	------

3. Edit Records

Admin can view or change Staff member's details in this section.

- Header file: `UserControl_Staff_Edit.h`
- Contains:
 - A textbox
 - A combobox
 - A button
 - A listbox.
- Textbox used to enter search string (name or EID).
- Combobox to select type of search (By name, by EID)
- Listbox gives result(s) of search.
- Selecting an option from Listbox redirects to `Form_Edit_Staff_Details.h`
- Records picked from `Staff_Register` table from `GueastHouse.accdb`.
- Data Structures Used:
 - Integer array 'joke'
- Editable details:
 - Designation
 - Identity proof
 - Address
 - Contact Number
 - Email
- Data Validation:
 - ➔ Error message on empty search string.
 - ➔ Only numerical allowed in search by ID



Function Description:

Name	Arguments	Description	Output
Button_Search_Click	System Arguments	Search db Staff_Register by name or ID and fill Listbox with valid results	Void
List_Search_SelectedIndexChanged	System Arguments	Open Form_Edit_Staff_Details for selected option.	Void

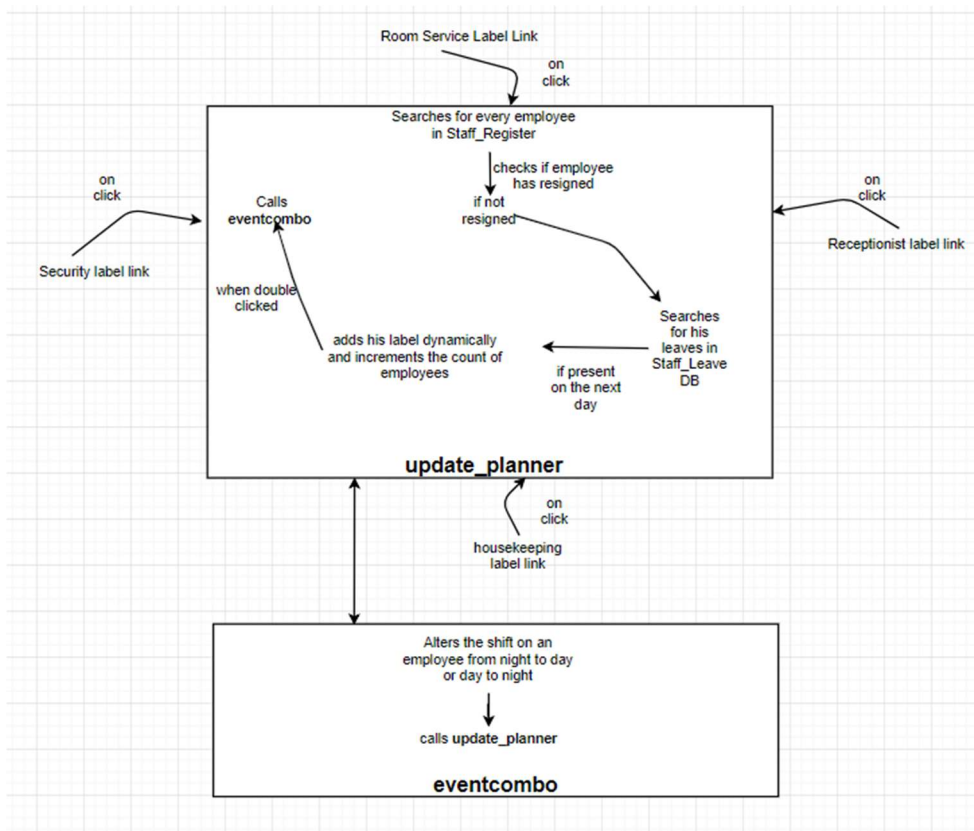
4. Planner

This feature is used by admin to plan his employee allotment into day and night shift for the next day.

- Header file: `UserControl_Staff_Planner.h`
- Contains:
 - A DateTimePicker

- Labels for Employee category
- Link labels to choose the employee category
- Dynamically added labels for employees
- DateTimePicker is disabled and only displays tomorrow's date
- Link Labels lets the admin view employees available in a particular category by clicking on it
- The dynamically added labels for employees can be double clicked to change their shift from day to night or vice versa

Flowchart:



Function Description:

Name	Arguments	Description	Output
Update planner	An integer	Updates the planner by adding dynamically labels of each employee of a user category (decided by the argument)	Void
Event_combo	System Arguments	Prompted when dynamically added labels for employees are double_clicked. Alters the shift of the employee and calls update planner	Void