Design

Design Data Dictionary

In order to function properly the system needs to make use of various data files which will store crucial information such as information about bookings or customers. This part of the design will aim to describe the files that will be necessary and what type of information they will store.

Initial Setup

Initially some files will contain a fixed number of “dummy” or blank records. These files will be created immediately after the system setup has been completed and the user has specified the file path of the system resource folder that will store all system related files. Certain files do not need to be filled with dummy records as they may not be necessary in certain situations. For example the extra requests file will only be created if the customer has a special request that they need to specify. Likewise the Customer Order file may not need to be created if the customer has chosen not to make use of the catering services. The files mentioned are non-essential and whether or not they exist depends entirely on the customer’s choice, if needed, a unique file will be created.

Below is a list of files and the number of dummy records they should contain initially.

Booking File – 1000 records

Customer File – 1000 records

Catering File – 200 records

Room File – 200 Records

User File – 100 records

Once the system generates all of the required files, then all of the control files containing the “Max” and “Current” fields should be updated so that the “Max” field is set to the number of dummy records that has been assigned to that file and the “Current” field is set to 0. This is done so that the number of records currently present in the file is known, and since this number will be 0 then this indicates that there are no records in the file. Should the maximum limit be reached then the system should copy the contents of the file into another file of the same name but with an extra 1000 dummy records added onto it. After this is done the old file should be archived or deleted. The newly created file should make use of the old control file, however the “Max” field should be amended to take into account an extra 1000 dummy records being added on.

Addition and Deletion of data

The addition of new data to the files is one of the most important functions of the system and must be handled with care. All files will add data by over-writing the next free “dummy” record in the file with a piece of data. Deletion of data poses a great problem for the system as it is a complicated process which involves moving large

amounts of data around. The bigger the file the slower this process will be. In order to alleviate this problem the system will make use of a special field within some files. This is the “Active” field which indicates whether or not the record is still available or if it has been deleted. This special field can be found within the User, Customer, Catering, Room and Booking files. Small files used by the system do not require an active field and their data can be deleted if necessary.

Naming Convention for generated files

Some of the files used by the system need to be generated on request, these files are the Customer order file, Room Setup file and the Extra requests file. Due to there being multiple such files they require an appropriate naming convention.

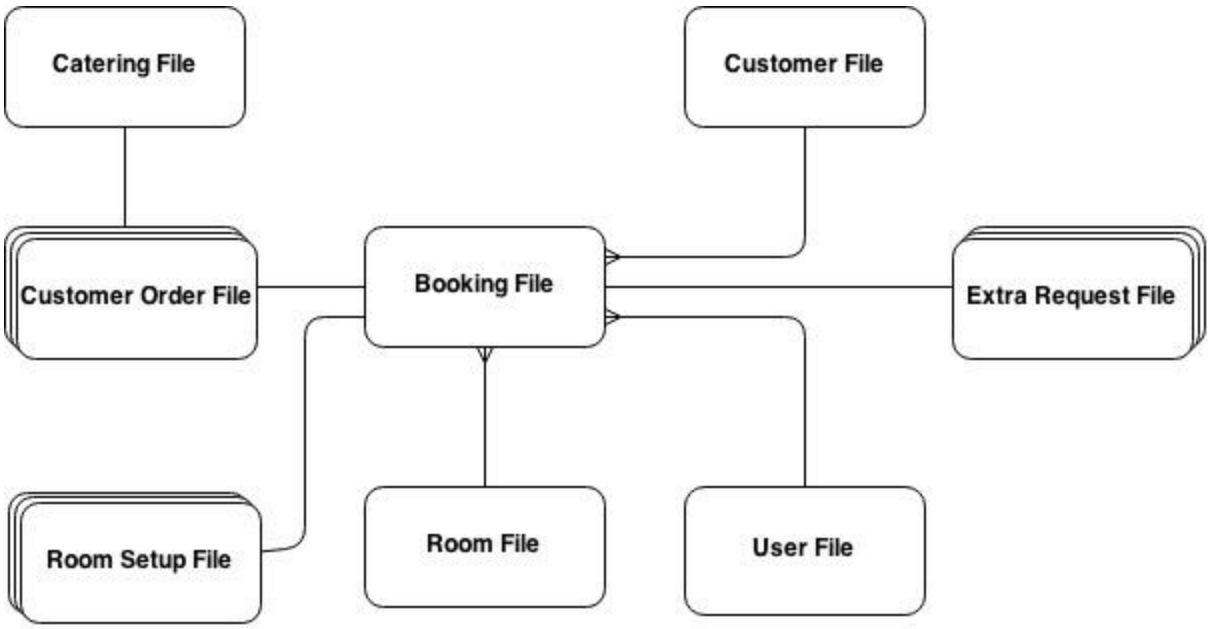
The naming convention for the customer order file is: Order_**BookingID**.dat

While the naming convention for the Extra requests file is: Request _**BookingID**.dat

Finally the naming convention for the Room setup file is: Setup_**BookingID**.dat

The Booking ID is an actual value from the Booking file used by the system.

Entity Relationship Diagram



User File

Name: User.dat Type: Random Access File

Name	Type	Size in Bytes	Validation	Description
User ID	Integer	4	Must be >0	Unique identifier for each user.

				Autonumber e.g. 1,2,3
User Name	String	50	Cannot be longer than 50 characters	Stores the username of a particular user. Max length is 50 characters
Password	String	50	Cannot be longer than 50 characters	This field stores the users password and has a maximum length of 50 characters
Access Level	Integer	4	Must be >=1 or <=3 1 = Administrator access 2 = Read and write access 3= Access to booking calendar	The user's access level dictates what they user can or cannot have access to in the system.
Active	Boolean	1	Value must be True of False	This field indicates whether or not the record is still available to the system.

Example record

User ID	**Username**	**Password**	**Access Level**	**Active**
8	"John"	"Smith"	2	True

Customer Data File

Name: Customer.dat Type: Random Access File

Name	**Type**	**Size**	**Validation**	**Description**
Customer ID	Integer	4	Value must be >0	Unique identifier for each customer. Customer IDs start from 1 and go up by 1 for each customer. Autonumber e.g. 1,2,3
Customer Name	String	50	Cannot be longer than 50 characters	Stores the name of the client making a booking
Telephone Number	String	11	Must be exactly 11 characters	The client's telephone number
Mobile Phone Number	String	11	Cannot be longer than 11 characters	Client's mobile phone number

Email Address	String	320	64 characters are assigned to the account name and a further 255 characters are available for the domain name. Must contain the “@” symbol.	Email address of the client
Postal Address	String	50	Must not be longer than 50 characters. Must support multiple postal address formats	Client’s postal address
Post Code	String	7	Post code cannot be longer than 7 characters. Must allow multiple postcode formats	Client’s Post code
Invoicing Contact Name	String	50	Cannot be longer than 50 characters	The name of the invoicing contact for a booking
Invoicing Telephone Number	String	11	Must be exactly 11 characters	The telephone number of the invoicing contact
Invoicing Mobile Phone Number	String	11	Must be exactly 11 characters	The mobile telephone number of the invoicing contact
Invoicing Email Address	String	320	64 characters are assigned to the account name and a further 255. Must contain the “@” symbol.	The invoicing contact’s email address
Invoicing Postal Address	String	50	Must not be longer than 50 characters. Must support multiple postal address formats	The postal address of the invoicing contact
Invoicing Post Code	String	7	Post code cannot be longer than 7 characters. Must allow multiple postcode formats	Invoicing contact’s Post code
Oldest Booking Pointer	Integer	4	Must be >0	Points to the position in the booking list file which holds the ID of the oldest booking made by the customer.
Active	Boolean	1	Value must be True of False	This field indicates whether or not the record is still available to the system.

Example

Customer Name	Telephone Number	Mobile Phone Number
“Adam Lawder”	07850456554	07850348809

Customer ID	Email Address	Postal Address	Post Code
580	AdamL@gmail.com	4 Bloomefield Avenue	BT78AU

Invoicing Contact Name	Invoicing Telephone Number	Invoicing Mobile Phone Number
Michael Johnston	0785048805	0785028513

Invoicing Email Address	Invoicing Postal Address	Invoicing Post Code
Michael78@hotmail.co.uk	34 Cregah Park	BT99DZ

Index	Active
33	True

Room File

Name: Room.dat Type: Random Access File

Name	Type	Size	Validation	Description
Room ID	Integer	4	Must be >0	Unique Identifier for each room. Room IDs start from 1 and go up by 1 for each room. Autonumber e.g. 1,2,3
Room Name	String	50	Must not be longer than 50 characters.	Stores the name of each room
Extra Details	String	100	Must not be longer than 100 characters	The extra details field can store the details regarding the availability of the room
Active	Boolean	1	Value must be True of False	This field indicates whether or not the record is still available to the system.

Example

Room ID	Room Name	Extra Details	Active
3	“Community Hall”	“Only available on Mondays”	True

Catering File

Name: Catering.dat Type: Random Access File

Name	Type	Size	Validation	Description
Item ID	Integer	4	Must be >0	Unique Identifier for each food or drink item. Autonumber e.g. 1,2,3
Item Name	String	50	Must not be longer than 50 characters.	Name of the food item
Price	Real(Single)	4	Must be >0	Price of the item in Pounds
Active	Boolean	1	Value must be True of False	This field indicates whether or not the record is still available to the system.

Example

Item ID	Item Name	Price
45	“Twix”	0.95

Customer Order File

Name: Naming convention for generated files can be found at the top

Type: Serial File

Name	Type	Size	Validation	Description
Item ID	Integer	4	Must be >0	Foreign Key Identifier for each food or drink item.
Quantity	Integer	4	Must be >=0	The quantity of each item
Delivery Time	Date	8	Must be a valid date.	Stores the time at which the specified item should arrive during the booking

Example

Item ID	Quantity	Delivery Time
18	9	17:00

Extra Requests File

Name: Naming convention for generated files can be found at the top

Type: Serial File

Name	Type	Size	Validation	Description
Extra Requests	String	1000	Must not exceed 1000 characters	Stores any extra requests regarding the catering service. For example this may include the preparation of items not present in the Catering resource

Example

Extra Requests
"I would like to request that all food ordered is prepared using gluten free ingredients."

Room Setup File

Name: Naming convention for generated files can be found at the top

Type: Serial File

Name	Type	Size	Validation	Description
Room Layout	String	1000	Must not be longer than 1000 characters	This is the layout of the room. Customer may choose one of the standard layouts or propose their own
Microphones	Integer	4	Must be >=0 and <=10	The number of microphones required for the booking.
Technical support	String	1000	Must not be longer than 1000 characters	An outline of the technical support that is necessary for the booking. Should also contain the number of hours of technical support needed

Example

Room Layout	Microphones	Technical Support
"The room should contain 15 chairs all placed in a circle. No tables or other kinds of furniture should be present."	1	"No technical support needed"

Booking File

Name: Booking.dat Type: Random Access File

Name	Type	Size	Validation	Description
Booking ID	Integer	4	Must be >0	The Booking ID is a unique

				identifier for each booking that is requested by a customer. Autonumber e.g. 1,2,3
Customer ID	Integer	4	Must be >0	Foreign Key Identifier for each customer.
Room ID	Integer	4	Must be >0	Foreign Key for each room
User ID	Integer	4	Must be >0	Foreign Key for each user
Reference	String	20	Cannot be longer than 0 characters	A reference for each booking. Can be entered in manually by the user.
Date	Date	8	Must be a valid date/time	This field stores the date of the booking. The data type “Date” stores the date in the format MM/DD/YYYY
Start Time	Date	8	Must be a valid date/time	Stores the time at which the booking should begin.
End Time	Date	8	Must be a valid date/time	Stores the time at which the booking should end.
Customer Order File	String	300	Must be a valid file path. Cannot be longer than 300 characters.	Stores the file path of the Customer order file. If there are no orders then the file will be left blank.
Room Setup File	String	300	Must be a valid file path. Cannot be longer than 300 characters.	Stores the file path of the Room Setup file. If room setup is not chosen then the file will be left blank.
Extra Requests File	String	300	Must be a valid file path. Cannot be longer than 300 characters.	Stores the file path of the Catering Requests file. If there are no extra requests then the file will be left blank
Active	Boolean	1	Value must be True of False	This field indicates whether or not the record is still available to the system.
NextBooking	Integer	4	Must be >=0	This is a pointer to the next node in the list which contains a booking ID. If its value is 0 then the customer has no other bookings.

Example

Booking ID	**Customer ID**	**Room ID**	**Date**	**Customer Order File**	**Room Setup File**	**Extra Requests File**
43	34	3	09/26/2014	C:\Common Folder\Booking System\Order_43.dat	C:\Common Folder\Booking System\ Setup_43.dat	C:\Common Folder\Booking System\Request_43.dat

Start Time	End Time	User ID	Active	Next Booking
10:00	15:30	24	True	433

System Setup File

Name: SystemSetup.dat Type: Serial File

Name	Type	Size	Validation	Description
Company Name	String	50	Must not be longer than 50 characters	This field stores the name of the company making use of the room booking program
Resource Folder	String	300	Must be a valid file path. Cannot be longer than 300 characters.	This is the file path of the resource folder containing all of the files used by the bookings system.

Example

Company Name	Resource Folder
East Belfast Mission	C:\Common Folder\Booking System\

Control Files

In order to function correctly the system also needs a number of control files which store data which concerns other files used by the system, such as the number of records or the maximum number of records.

System Setup Control File

Name: SetupControl.dat

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.

Example

InUse
True

Customer Control File

Name: CustomerControl.dat Type: Serial File

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.
Max	Integer	4	Must be >0	Indicates the maximum allowable number of records in the file
Current	Integer	4	Must be >=0	Indicates the current number of records in the file.

Room Control File

Name: RoomControl.dat Type: Serial File

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.
Max	Integer	4	Must be >0	Indicates the maximum allowable number of records in the file
Current	Integer	4	Must be >=0	Indicates the current number of records in the file.

Booking Control File

Name: BookingControl.dat Type: Serial File

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.
Max	Integer	4	Must be >0	Indicates the maximum allowable number of records in the file
Current	Integer	4	Must be >=0	Indicates the current number of records in the file.

CateringControl File

Name: CateringControl.dat Type: Serial File

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.
Max	Integer	4	Must be >0	Indicates the maximum allowable

				number of records in the file
Current	Integer	4	Must be >=0	Indicates the current number of records in the file.

User Control File

Name: UserControl.dat Type: Serial File

Name	Type	Size	Validation	Description
InUse	Boolean	1	True or False	Indicates whether the file is currently being accessed by a user.
Max	Integer	4	Must be >0	Indicates the maximum allowable number of records in the file
Current	Integer	4	Must be >=0	Indicates the current number of records in the file.

Example

InUse	Max	Current
False	1000	56

Integrated Example

The diagram below shows the Customer File which contains all of the personal information about a customer whose name is “Jordan Earle”. The field “Oldest Booking Pointer” acts as a link from that particular record to a position in the Booking file which stores the oldest booking made by the customer. The field contains the number 200 and hence is linked to the record stored at position 200 in the Booking file.

The record in the Booking List file contains the Booking ID for the customer and the file position, in the Booking List file, of the next Booking created by the customer. From the diagram we can see that the oldest booking requested by the customer has the Booking ID of 200 and that the position, in the Booking List file, of the next Booking ID for that customer is 201. The key field called Booking ID acts as a link to the Booking File as it also acts as the position in the Booking file which holds the information about that particular booking.

The record at position 200 in the Booking file contains the details for the booking such as the ID of the customer who made the booking, the date of the booking and the ID of the room in which the booking should take place. The room ID acts as a link to the Room File and is shown to contain the value of 2. In the Room file we can see that the record at position 2 in the file stores the details about the room such as its name. In the diagram the name of room 2 is “Dance Hall”.

The Booking File also shows two fields called “Room Setup File” and “Extra Requests File”. These two fields store the pathnames for the Room setup file which contains the information regarding the layout of the room and notes about the technical support, and for the Extra Requests File which stores the details of any extra catering requests that the customer may have.

From the diagram we can see that the Booking File contains a field called “Customer Order File” which holds the path name of the Order File. The Customer Order File stores the ID and quantity of every item requested by the customer. The Item ID shown in the diagram is 19 and is linked with the Item ID in the Catering file which is held at position 19 in said file. At this position in the Catering file we can find the Item with an ID of 19 and its details such as the fact that its name is “Twix”. In the diagram we can only see one item being shown but in reality the file will contain a number of items that have been ordered by the customer.

Finally the Booking File is linked with the User File via the User ID field within the file. The User file is shown to store the information about the User who processed the booking, which in this case is the User with ID 9.

Customer File	
File Position	32
Customer ID	32
Customer Name	Jordan Earle
Telephone Number	12345654321
Mobile Phone Number	09876567890
Email Address	jearle@me.com
Postal Address	78 Jersey Street
Post Code	BT56DF
Invoicing Contact Name	Matt Smith
Invoicing Telephone Number	64788923453
Invoicing Mobile Phone Number	23642587342
Invoicing Email address	MSmith@gmail.com
Invoicing Postal address	45 Houston Park
Invoicing Post Code	DB67HH
Oldest Booking Pointer	200
Active	True

Booking File	
File Position	200
Booking ID	200
Customer ID	32
Room ID	2
User ID	9
Reference	DA11112014
Date	11/11/2014
Start Time	10:30
End Time	12:00
Customer Order File	C:\Common Folder\Booking System\Order_200.dat
Room Setup File	C:\Common Folder\Booking System\Setup_200.dat
Extra Requests File	C:\Common Folder\Booking System\Request_200.dat
Active	True
Next Booking	201

Room File	
File Position	2
Room ID	2
Room Name	Dance Hall
Extra Details	Only available on Tuesdays
Active	True

User File	
File Position	9
User ID	9
User Name	Chris Purdy
Password	Code monkey
Access Level	1
Active	True

Room Setup File	
Room Layout	Room Should be cleared of all furniture
Microphones	0
Technical Support	No technical support needed

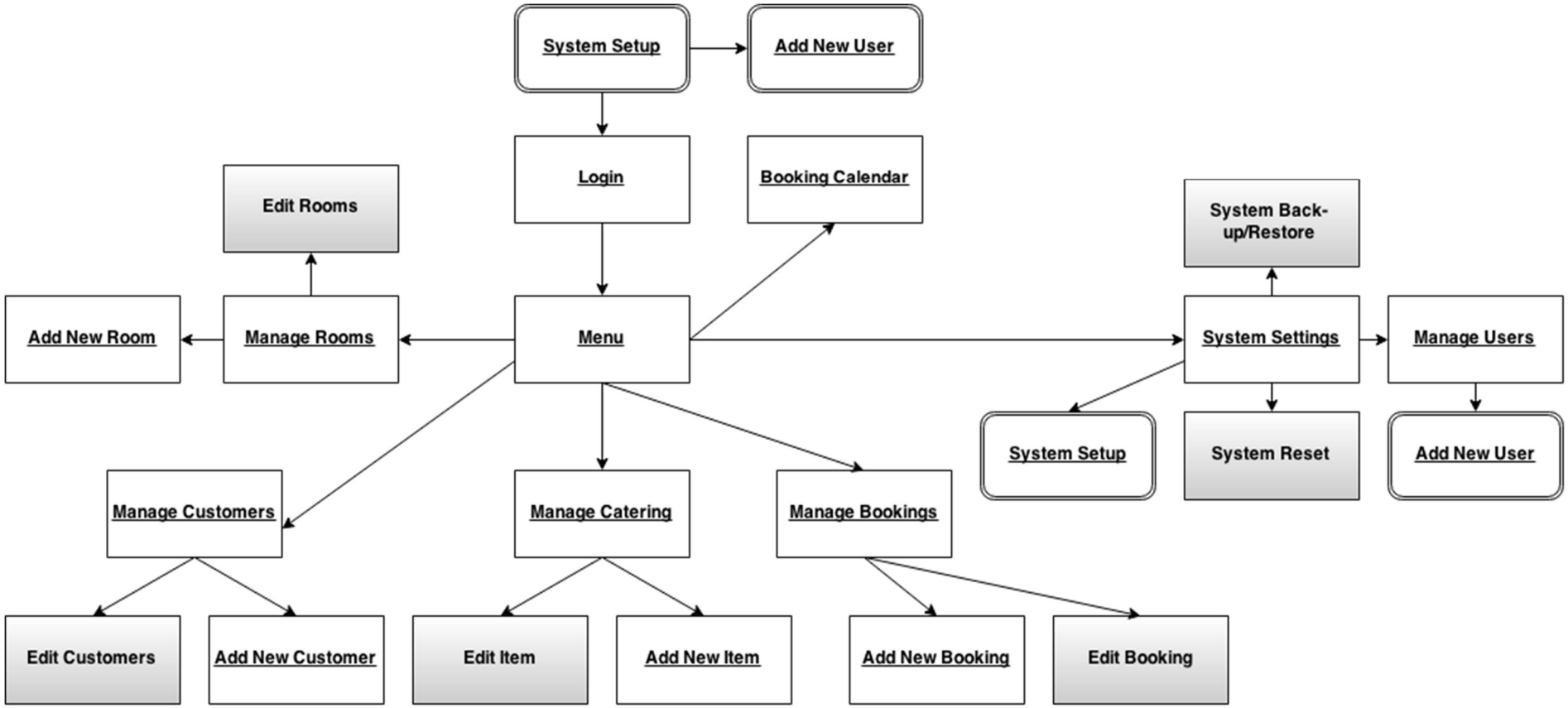
Customer Order File	
Item ID	19
Quantity	15
Delivery Time	11:00

Catering File	
File Position	19
Item ID	19
Item Name	Twix
Price	0.95
Active	True

Extra Requests File	
Extra Requests	I would like to request that all food ordered is made using gluten free ingredients

Hierarchy Chart

The chart below details the forms that will be used in the new system and their links with other forms. From the chart we can see that the “System setup” and “Add User” forms are both highlighted and also appear twice in this chart. The reason behind this is that when the user opens the program for the first time they will be asked to setup the system, thus meaning that the setup form must be shown before the log-in form. Once the system is set-up the user setting up the entire system for the first time will have to make themselves an administrator of the system, hence the “Add User” form needs to be shown. It should be noted that if the system has had an initial setup and all of the system files exist then the “Add User” form will not be shown and the administrator will have to add that particular individual to the list of allowable users. Any name which is underlined signifies a form while any box with a double outline is a form which can be accessed via more than one route.


System Setup
Add New User
Login
Booking Calendar
Edit Rooms
System Back-up/Restore
Add New Room
Manage Rooms
Menu
System Settings
Manage Users
System Setup
System Reset
Add New User
Manage Customers
Manage Catering
Manage Bookings
Edit Customers
Add New Customer
Edit Item
Add New Item
Add New Booking
Edit Booking

Component Acronyms

Each component name begins with a three letter acronym which can be used to identify the type of component. Below is a list of these acronyms and their descriptions

“txt” – A text box which enables user to enter text

“btn” – A simple button which can be pressed to trigger an event. Can contain text

“lbl” – A label which displays text on the screen

“cmb” – A Combo box which displays an editable text box with a drop down list of allowable values

“dgv” – Data grid view which displays rows and columns of data in a customizable grid

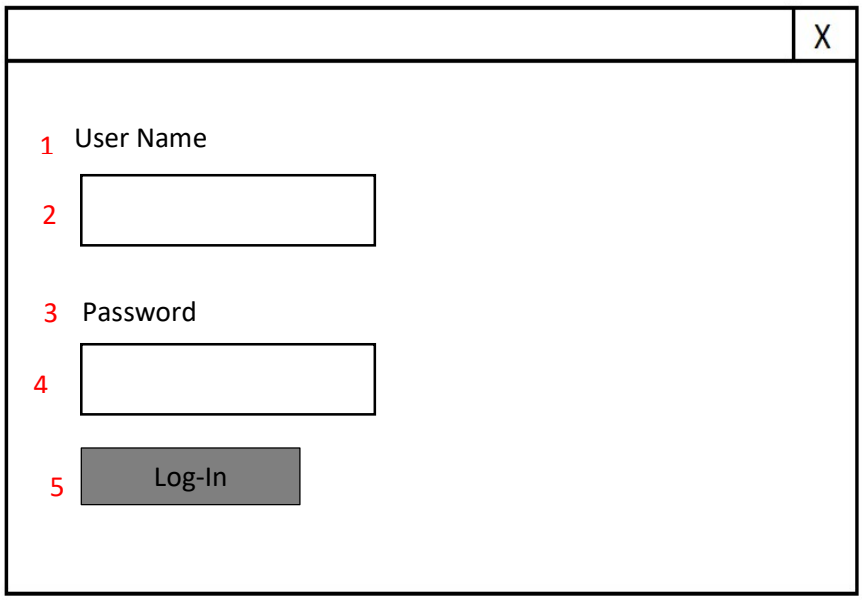
“cbx” – Check box which allows the user to select an associated option

“mtb” – Masked text box allows for the concealment of characters entered in to it

“pnl” – Panel holding a group of related components

“frm” – A form which contains components and carries out a specific function

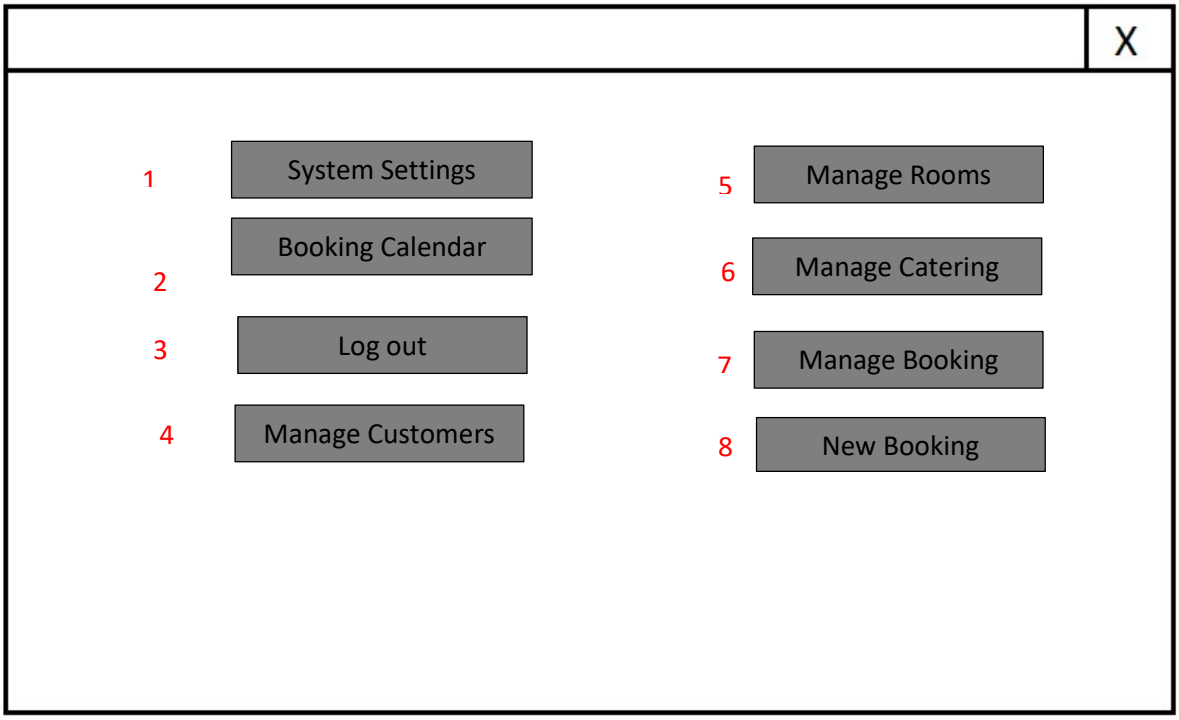
Login Form (frmLogin)



This Log-in form will be displayed every time the program is opened, provided that the initial system setup has taken place. The form's purpose is to prevent unauthorised access to the system and its resources.

1. lblUserName – A label with the text "User Name"
2. txtUserName – A text box for entering in the user name
3. lblPassword – A label with the text "Password"
4. mtbPassword – Masked text box for entering a password. Characters in this masked text box come up as the "*" character
5. btnLogIn – A button which allows the user to log in to the system. Once this button is pressed the system checks if the username and password belong to a valid user. The program goes through the entire User file to see if a user with the username and password that has been entered into the text boxes exists. If the user does exist the menu form will be shown, otherwise a message box should be shown telling the user to try again. The text for this button is "Log-In".

Menu Form (frmMenu)

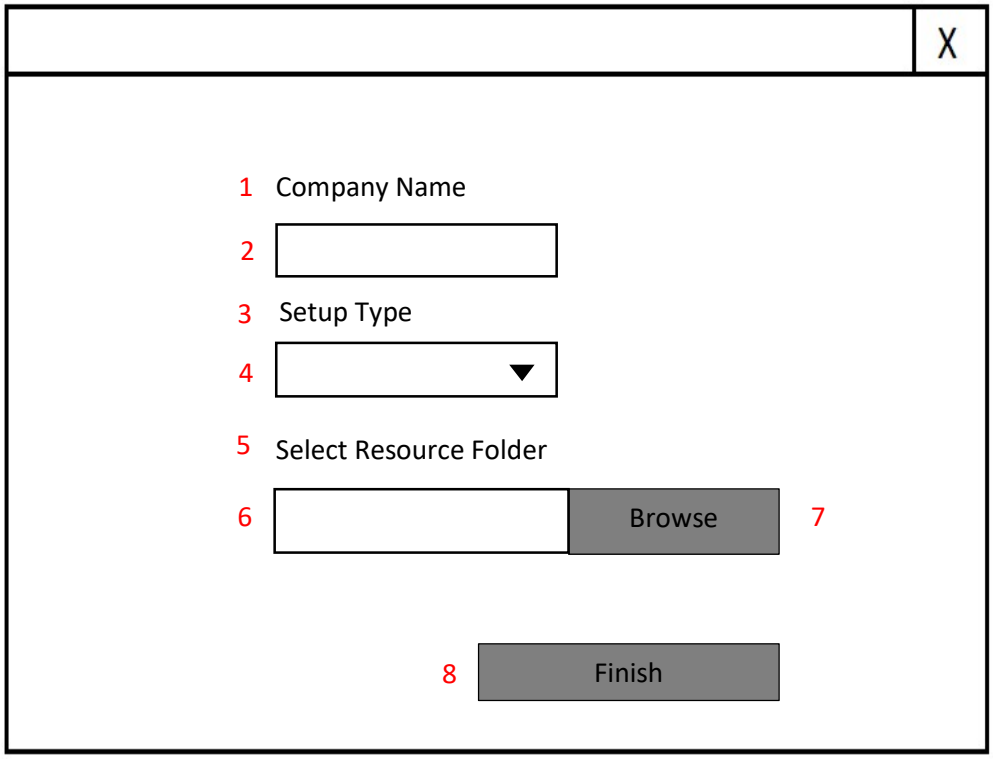


The menu form contains buttons which allow for navigating around the system.

Once a button is pressed a form should be shown on the screen. This form should also contain a button which allows the user to log out of the system.

1. btnSettings – Button with the text “System Settings”. Once the button is pressed the system settings form should be displayed.
2. btnBookingCalendar – Button with the text “Booking Calendar”. Once the button is pressed the Booking Calendar form should be displayed.
3. btnLogOut – Button with the text “Log out”. Once the button is pressed the Log-in form should be displayed.
4. btnManageCustomers – Button with the text “Manage Customers”. Once the button is pressed the Manage Customers form should be displayed.
5. btnManageRooms – Button with the text “Manage Rooms”. Once the button is pressed the ManageRooms form should be displayed.
6. btnManageCatering – Button with the text “Manage Catering”. Once the button is pressed the ManageCatering form should be displayed.
7. btnManageBookings – Button with the text “Manage Bookings”. Once the button is pressed the Manage Bookings form should be displayed.
8. btnNewBooking – Button with the text “Create a New Booking”. Once the button is pressed the NewBooking form should be displayed.

System Setup Form (frmSetup)

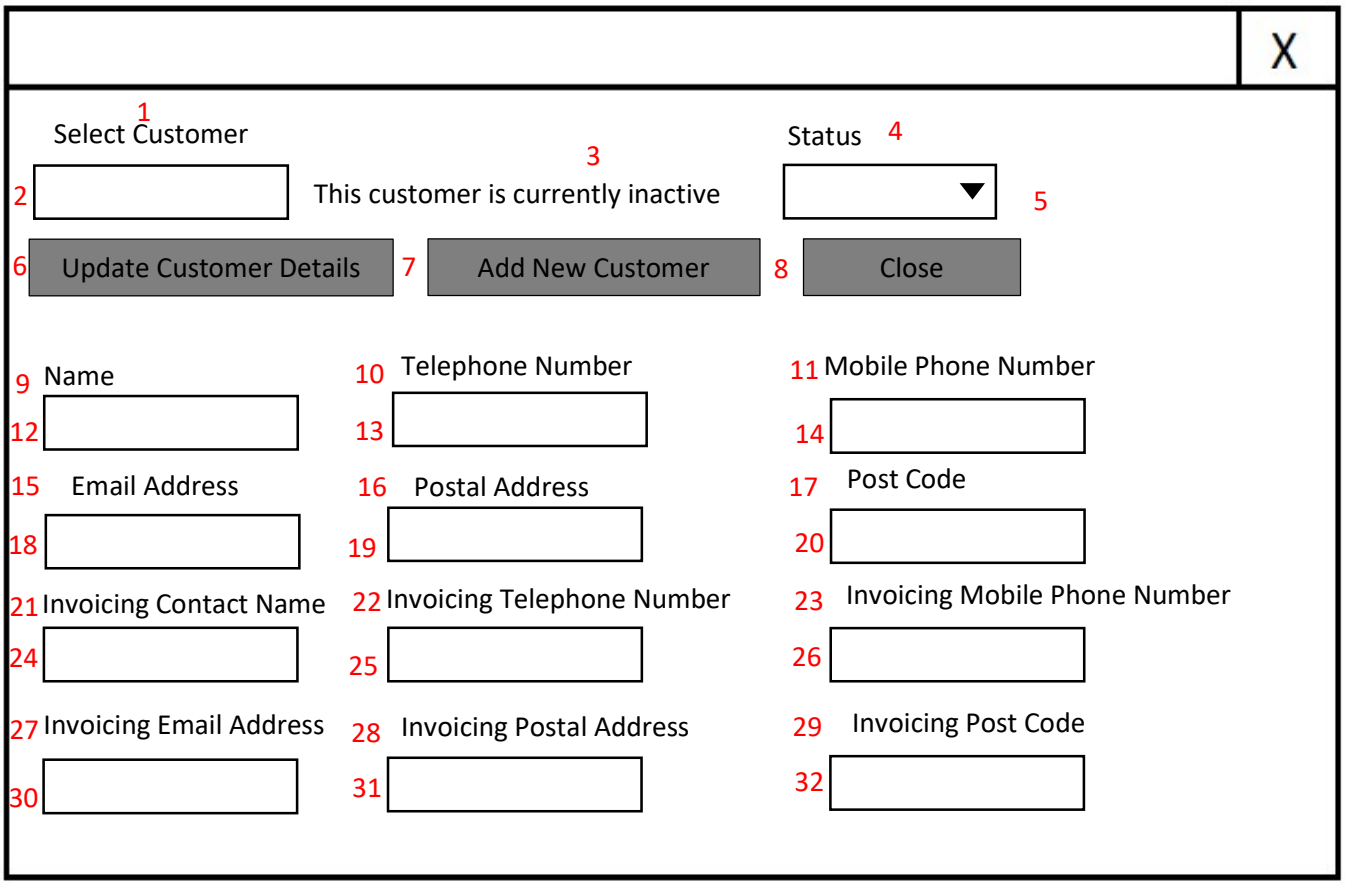


The system setup form allows the user to setup the system. This form asks for the name of the company that will use system, the location in which the system files should be created or the location of the existing system files, and type of setup. The two types of setup that are available are the system setup and the user setup. The system setup should be chosen if the system has not been setup before, while the User setup should be used to setup a copy of the system on a user's machine given that the system files have already been created.

1. lblCompanyName – A label with the text "Company Name"
2. txtCompanyName – Text Box for entering in the name of the company making use of the system
3. lblSetupType – Label with the text "Setup Type"
4. cmbSetupType – A combo box which allows the user to choose what kind of setup they wouyld like to perform. The combo box should contain two options; these are "System Setup" and "User Setup".
5. lblResourceFolder – Label the text "Select Resource Folder"
6. txtResourceFolder – This text box allows the user to type in the file path of where the new resource folder should be created or the location of the existing resource folder containing the system files.
7. btnBrowse – A button with the text "Browse". Once pressed by a user the button causes a folder browser dialog to be shown. This provides an alternative to entering in the file path of the resource folder into txtResourceFolder.

8. btnFinish – A button with the text “Finish”. Closes the form and sets up the system. If the system setup is chosen then all of the system files will be created and fill with 50 dummy records.

Manage Customers Form (frmManageCustomers)

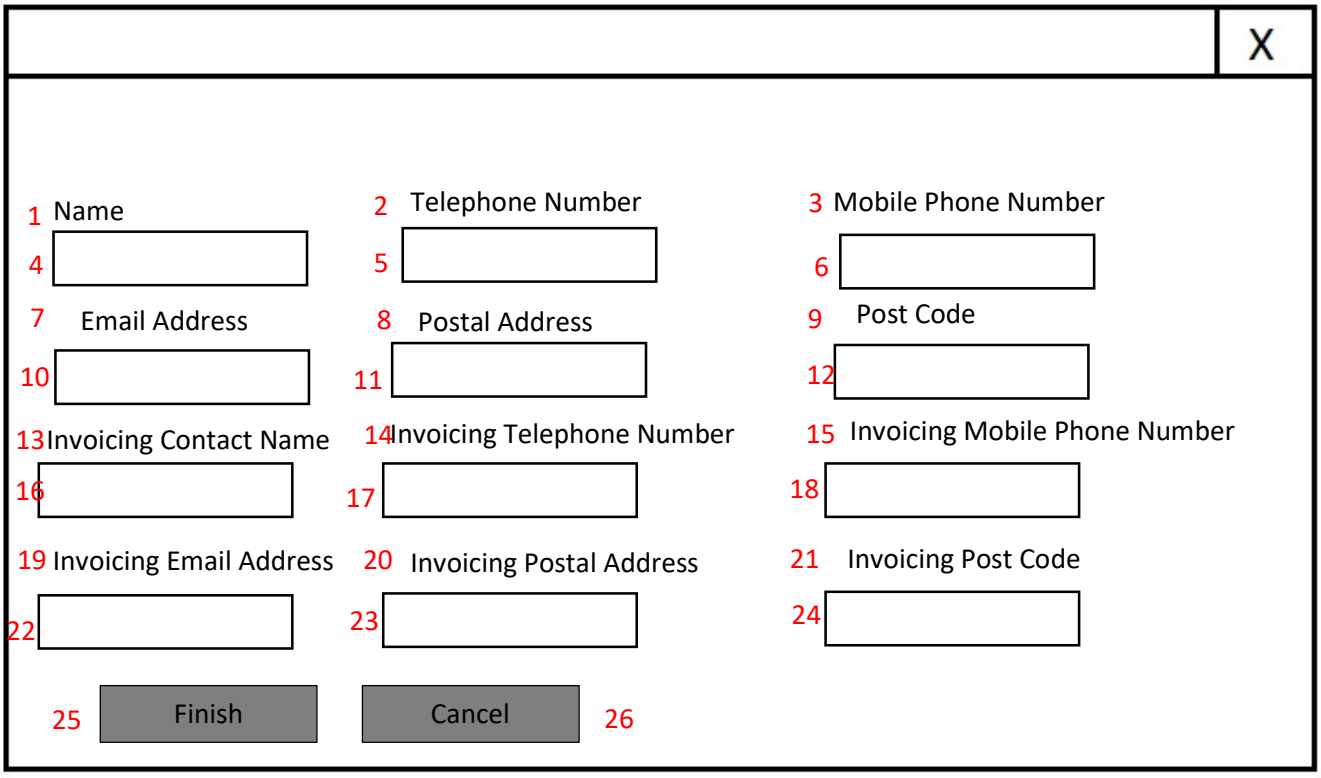


The Manage Customers form has two main uses, the most important of which is that it allows the user to update customer data which has already been entered into the system. The second purpose of the form is that it contains a link to the Add New Customer form.

1. lblCustomer – A label containing the text “Select Customer”
2. lstCustomer – List box containing the names of customers that are stored in the system’s database
3. lblWarning – A label contining the text “This customer is currently inactive”. If the customer chosen in lstCustomer is active then the label should be hidden, otherwise if the customer chosen is not active then the label should become visible
4. lblStatus – A label containing the text “Status”
5. cmbStatus – A combo box which will contain the status of the currently selected customer
6. btnUpdate – A button with the text “Update Customer Details”. Once pressed, the information which has been updated will be written to the customer file at the position given by the ID of the customer.

7. btnAddNewCustomer – A button with the text “Add New Customer”. When pressed this button shows the Add New Customer form
8. btnClose – A button with the text “Close”. Closes the form
9. lblName – A label with the text “Name”
10. lblTelephone – A label with the text “Telephone Number”
11. lblMobile – A label with the text “Mobile Phone Number”
12. txtName – Text box containing the name of the customer
13. txtTelephone – Text box containing the telephone of the customer
14. txtMobile – Text box for containing the mobile phone of the customer
15. lblEmail – A label with the text “Email Address”
16. lblAddress – A label with the text “Postal Address”
17. lblPostcode – A label with the text “Postcode”
18. txtEmail – Text box containing the email address of the customer
19. txtAddress – Text box containing the postal address of the customer
20. txtPostcode – Text box containing the postcode of the customer
21. lblInvoiceName – A label with the text “Invoicing Contact Name”
22. lblInvoiceTelephone – A label with the text “Invoicing Telephone Number”
23. lblInvoiceMobile – A label with the text “Invoicing Mobile Phone Number”
24. txtInvoiceName – Text box containing the name of the customer
25. txtInvoiceTelephone – Text box containing the telephone of the customer
26. txtInvoiceMobile – Text box for containing the mobile phone of the customer
27. lblInvoiceEmail – A label with the text “Invoicing Email Address”
28. lblInvoiceAddress – A label with the text “Invoicing Postal Address”
29. lblInvoicePostcode – A label with the text “Invoicing Postcode”
30. txtInvoiceEmail – Text box containing the email address of the customer
31. txtInvoiceAddress – Text box containing the postal address of the customer
32. txtInvoicePostcode – Text box containing the postcode of the customer

Add New Customer Form (frmNewCustomer)

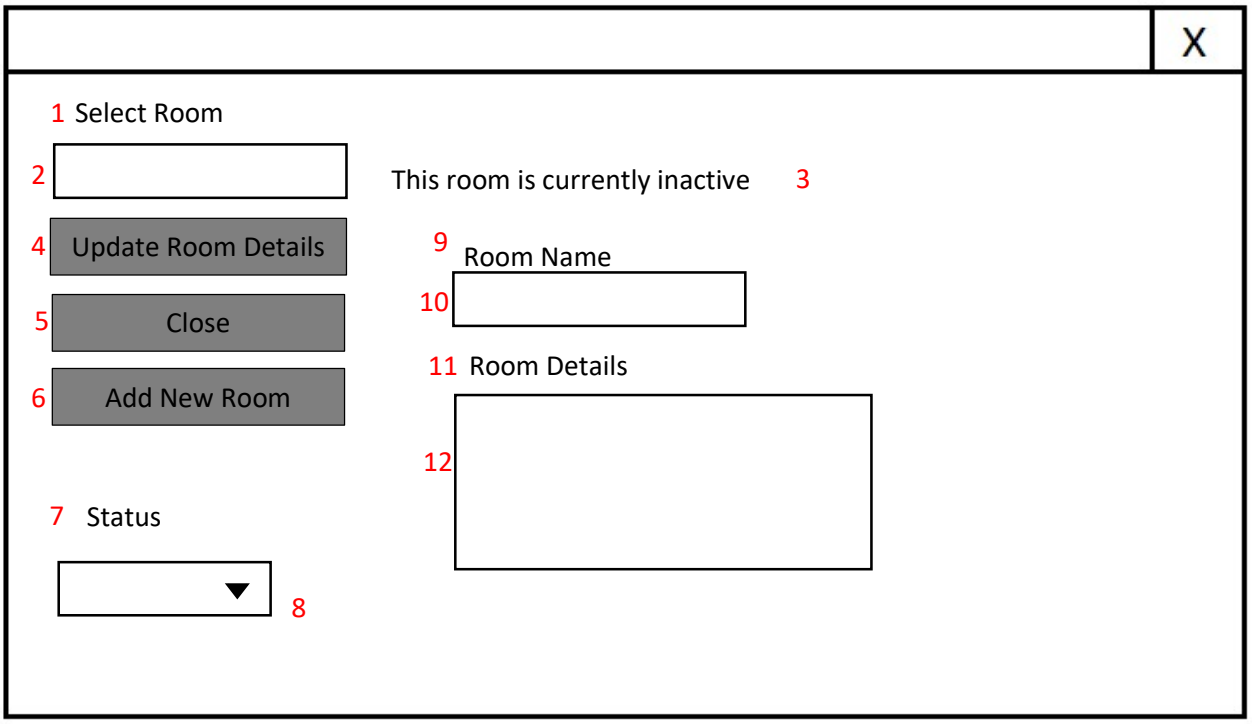


The Add New Customer form allows the user to add a new user to the system.

1. lblName – A label with the text “Name”
2. lblTelephone – A label with the text “Telephone Number”
3. lblMobile – A label with the text “Mobile Phone Number”
4. txtName – Text box containing the name of the customer
5. txtTelephone – Text box containing the telephone of the customer
6. txtMobile – Text box for containing the mobile phone of the customer
7. lblEmail – A label with the text “Email Address”
8. lblAddress – A label with the text “Postal Address”
9. lblPostcode – A label with the text “Postcode”
10. txtEmail – Text box containing the email address of the customer
11. txtAddress – Text box containing the postal address of the customer
12. txtPostcode – Text box containing the postcode of the customer
13. lblInvoiceName – A label with the text “Invoicing Contact Name”
14. lblInvoiceTelephone – A label with the text “Invoicing Telephone Number”
15. lblInvoiceMobile – A label with the text “Invoicing Mobile Phone Number”
16. txtInvoiceName – Text box containing the name of the customer
17. txtInvoiceTelephone – Text box containing the telephone of the customer
18. txtInvoiceMobile – Text box for containing the mobile phone of the customer
19. lblInvoiceEmail – A label with the text “Invoicing Email Address”
20. lblInvoiceAddress – A label with the text “Invoicing Postal Address”
21. lblInvoicePostcode – A label with the text “Invoicing Postcode”
22. txtInvoiceEmail – Text box containing the email address of the customer
23. txtInvoiceAddress – Text box containing the postal address of the customer

24. txtInvoicePostcode – Text box containing the postcode of the customer
25. btnFinish – A button with the text “Finish”. Closes form and adds the new information to the file at the next free position in the customer file.
26. btnClose – A button with the text “Close”. Closes the form

Manage Rooms Form (frmManageRooms)

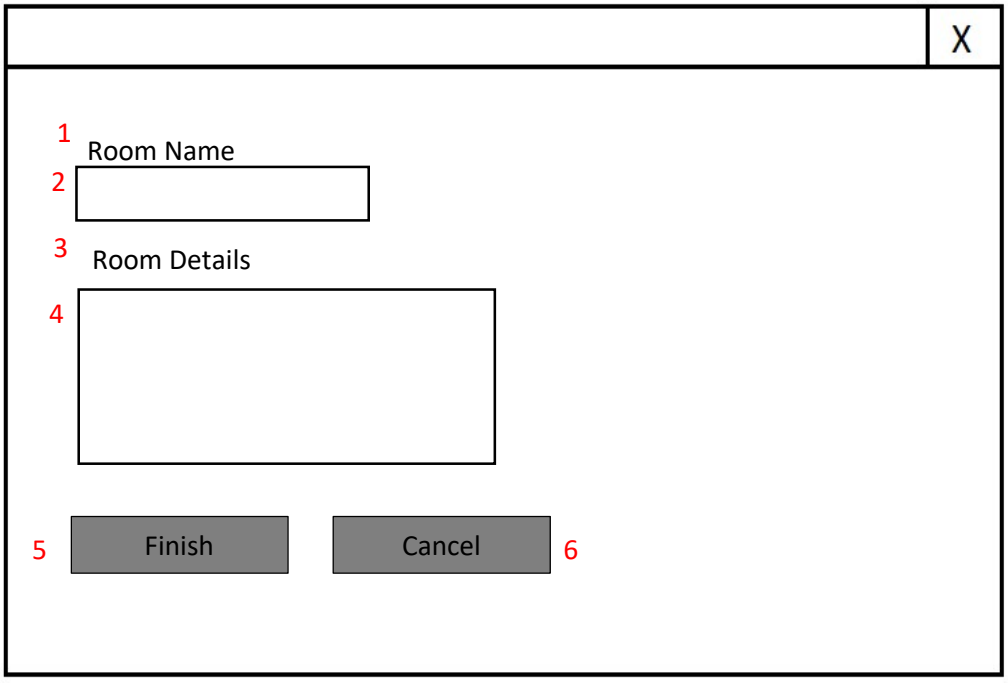


The Manage Rooms form allows the user to update the information regarding the rooms which can be hired out.

1. lblRoom – A label containing the text “Select Room”
2. lstRoom – Combo box containing the names of customers that are stored in the system’s database
3. lblWarning – A label containing the text “This Room is currently inactive”. If the Room chosen in lstRoom is active then the label should be hidden, otherwise if the Room is not active then the label should become visible
4. btnUpdate – A button with the text “Update Room Details”. Once pressed the edited details of the room will be written to the room file at the position given by the ID of the room currently being processed.
5. btnClose – A button with the text “Close”. Closes the form
6. btnAddNewRoom – A button with the text “Add New Room”. When pressed this button shows the Add New Room form
7. lblStatus – A label containing the text “Status”
8. cmbStatus – A combo box which will contain the status of the currently selected room
9. lblName – A label with the text “Room Name”
10. txtName – Text box containing the name of the currently selected room
11. lblDetails – A label containing with the text “Room Details”

12. txtDetails – Text box containing the details concerning the currently selected room

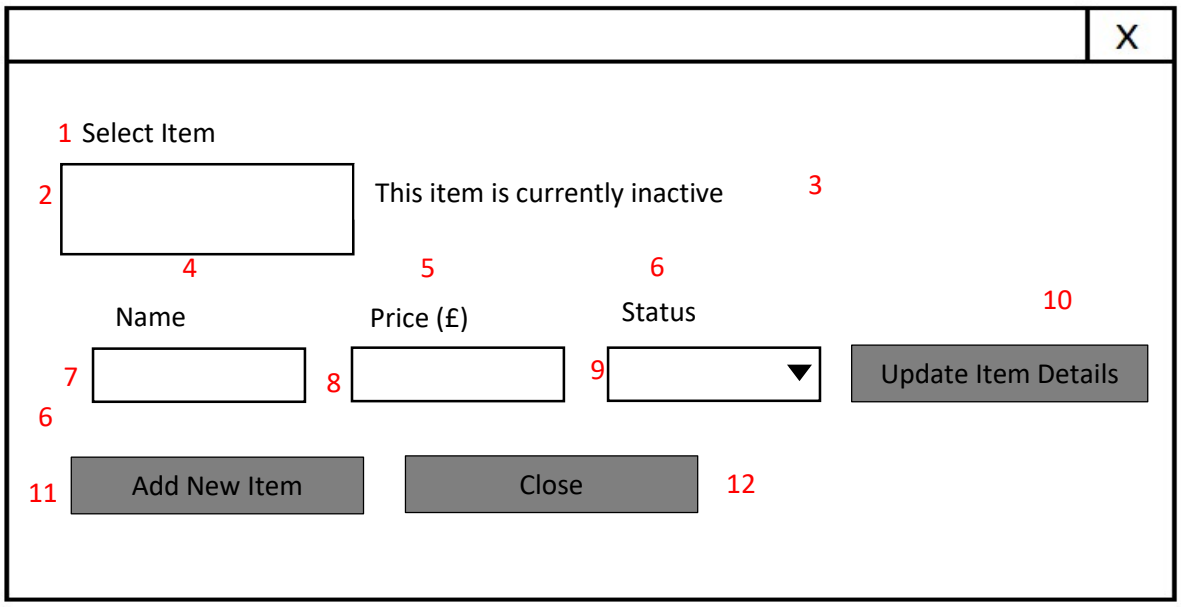
Add New Room Form (frmNewRoom)



This form allows the user to create a new room.

1. lblName – A label with the text “Room Name”
2. txtName – Text box containing the name of the currently selected room
3. lblDetails – A label containing with the text “Room Details”
4. txtDetails – Text box containing the details concerning the currently selected room
5. btnFinish – A button with the text “Finish”. Closes form and adds the new information to the room file at the next free space.
6. btnClose – A button with the text “Close”. Closes the form

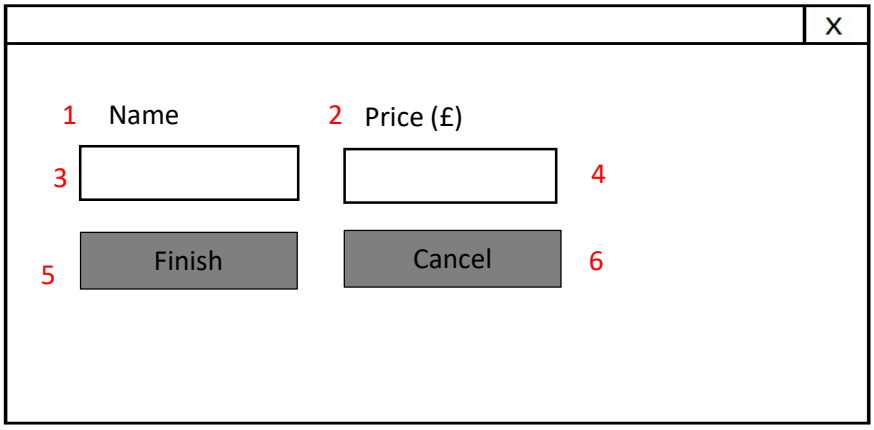
Manage Catering Form (frmManageCatering)



The manage customers form allows the user to edit the information concerning an item.

1. lblItem – A label containing with the text “Select Item”
2. lstItem – Combo box containing the names of already existing items
3. lblWarning – A label with the text “This item is currently inactive”. If the Itemchosen in lstItem is active then the label should be hidden, otherwise if the Item is not active then the label should become visible
4. lblName – A label with the text “Name”
5. lblPrice – A label with the text “Price (£)”
6. lblStatus – A label with the text “Status”
7. txtName – Text box containing the name of the currently selected item
8. txtPrice – Text box containing the price of the selected item
9. cmbStatus – Combo box holding the status of the current item
10. btnUpdate – A button with the text “Update Item Details”. Once pressed the updated item information is written to the file at the position given by the ID of the currently processed item.
11. btnAddNewItem – A button with the text “Add New Item”. Shows the Add New Item form
12. btnClose – A button with the text “Close”. Closes the form

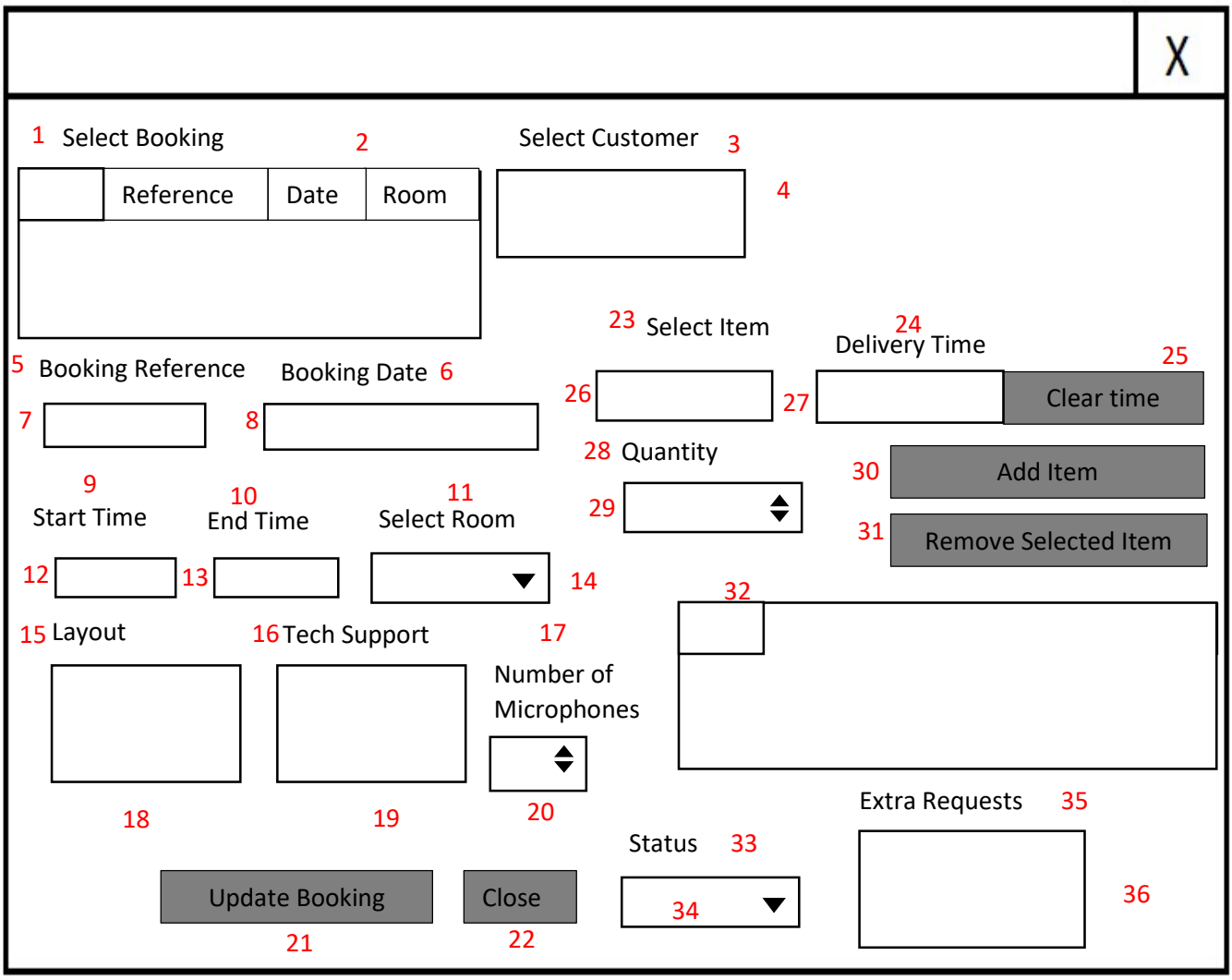
Add New Item Form (frmNewItem)



This form allows for the addition of new items

1. lblName – A label with the text “Name”
2. lblPrice – A label with the text “Price (£)”
3. txtName – Text box containing the name of the currently selected item
4. txtPrice – Text box containing the price of the selected item
5. btnFinish – A button with the text “Finish”. Closes form and adds the new item information to the Catering file at the next free position.
6. btnClose – A button with the text “Close”. Closes the form

Manage Bookings Form (frmManageBookings)



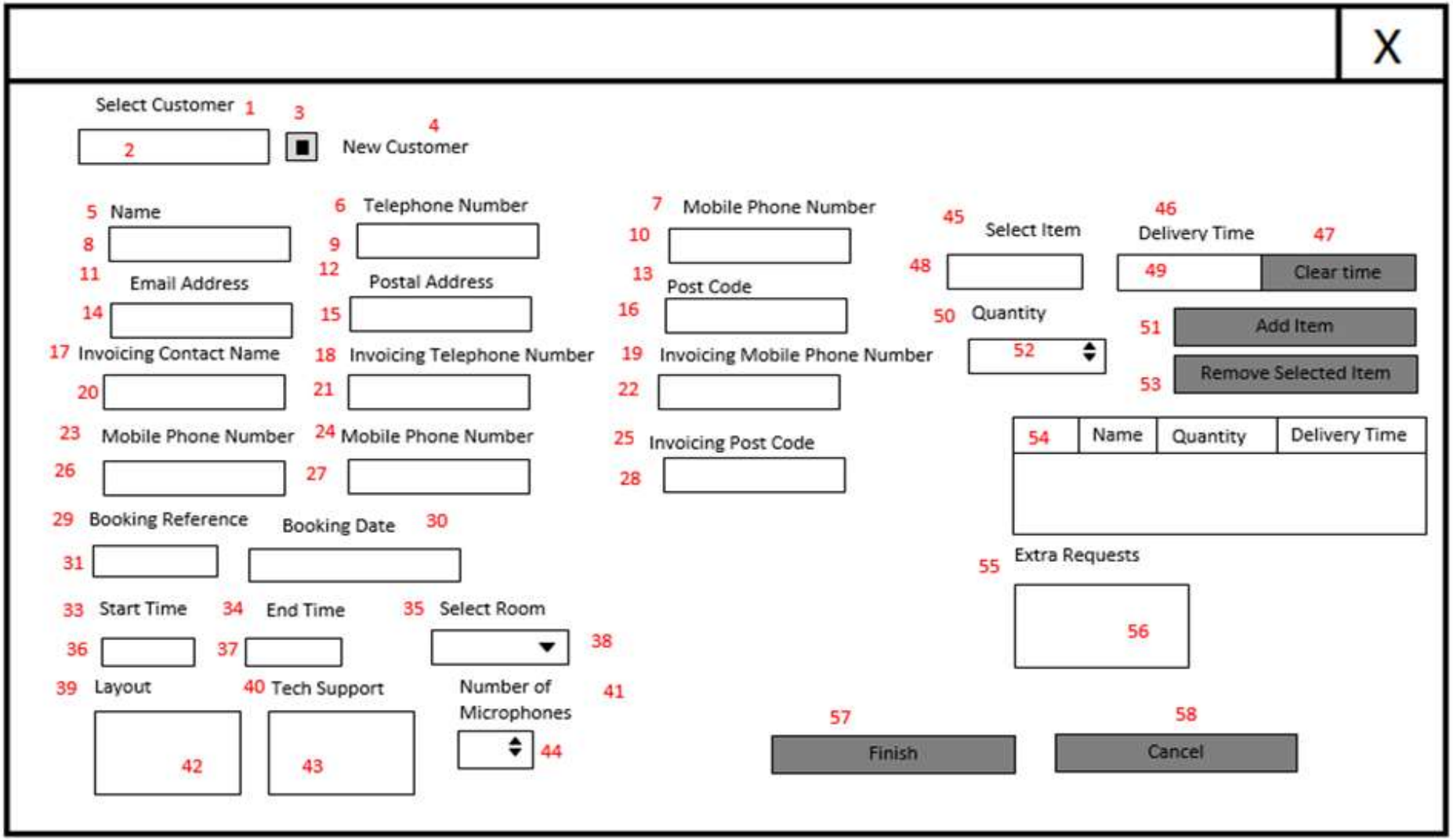
The Manage bookings form provides the user with a facility to update existing bookings.

1. lblBooking – A label with the text “Select Booking
2. dgvBooking - Data grid view with the following fields: “Reference”, ”Date”, ”Room” and “Booking ID”. All of the fields apart from the Booking ID field are visible. This Data grid view is used to display the bookings for a particular customer.
3. lblCustomer – A label with the text “Select Customer”
4. lstCustomer – List box containing the names of customers
5. lblReference – A label with the text “Booking Reference”
6. lblDate – A label with the text “Booking Date”
7. txtReference – Text box containing the booking reference fo a particular booking
8. dtpDate – Date-Time Picker containing the date of the booking
9. lblStartTime – Label containing the text “Start Time”

10. lblEndTime – Label containing the text “End Time”
11. lblRoom – Label containing the text “Select Room”
12. dtpStartTime – Date Time Picker containing the start time of the booking
13. dtpEndTime – Date Time Picker containing the end time of the booking
14. cmbRoom – Combo box containing the names of rooms
15. lblLayout – Label containing the text “Layout”
16. lblTechSupport – Label containing the text “Tech Support”
17. lblMicrophone – Label containing the text “Number of Microphones”
18. txtLayout – Text box storing the layout details of the room
19. txtTechSupport – Text box containing the details of any technical support required for the booking
20. nudMicrophone – Numeric-Up-Down storing the number of microphones requested for the booking
21. btnUpdate – Button with the text “Update Booking”. The updated information is written to a position in the booking file that is given by the ID of that particular booking.
22. btnClose- Button with the text “Close”. Closes the form once pressed
23. lblItem – Label containing the text “Select Item”
24. lblDeliveryTIme – Label containing the text “Delivery Time”
25. btnClearTime – ButtonIst with the text “Clear Time”. Clears the time stored in dtpDelivery
26. lstItem – List box containing the names of items which could be ordered
27. dtpDelivery – Date-time picker storing the delivery time of a particular item
28. lblQuantity – Label with the text “Quantity”
29. nudQuantity – Numeric-Up-Down storing the quantity of an item
30. btnAddItem – Button with the text “Add Item”. Gets the ID of the item selected in lstItem from an array which holds the ID’s of every single item in lstItem. The ID is then used to get the name of the item from the catering file at a position equal to the ID of the selected item. This name is then placed in dgvItem, along with the delivery time and quantity given by dtpDelivery and nudQuantity respectively.

31. btnRemoveSelected – Button with the text “Remove Selected”. Removes an item from dgvItem by deleting the row that has been selected by the user.
32. dgvItem – Data grid view with the following fields : “Name”, ”Quantity”, ”Delivery Time”. This component displays the items that have been ordered
33. lblStatus – Label with the text “Status”
34. txtStatus – Text box containing the status of the booking
35. lblExtraRequest – Label containing the text “Extra Requests”
36. txtExtra Request – Text box containing any extra requests concering the catering

New Booking Form (frmNewBooking)



This form allows the user to create a new booking for an existing customer or for a new customer.

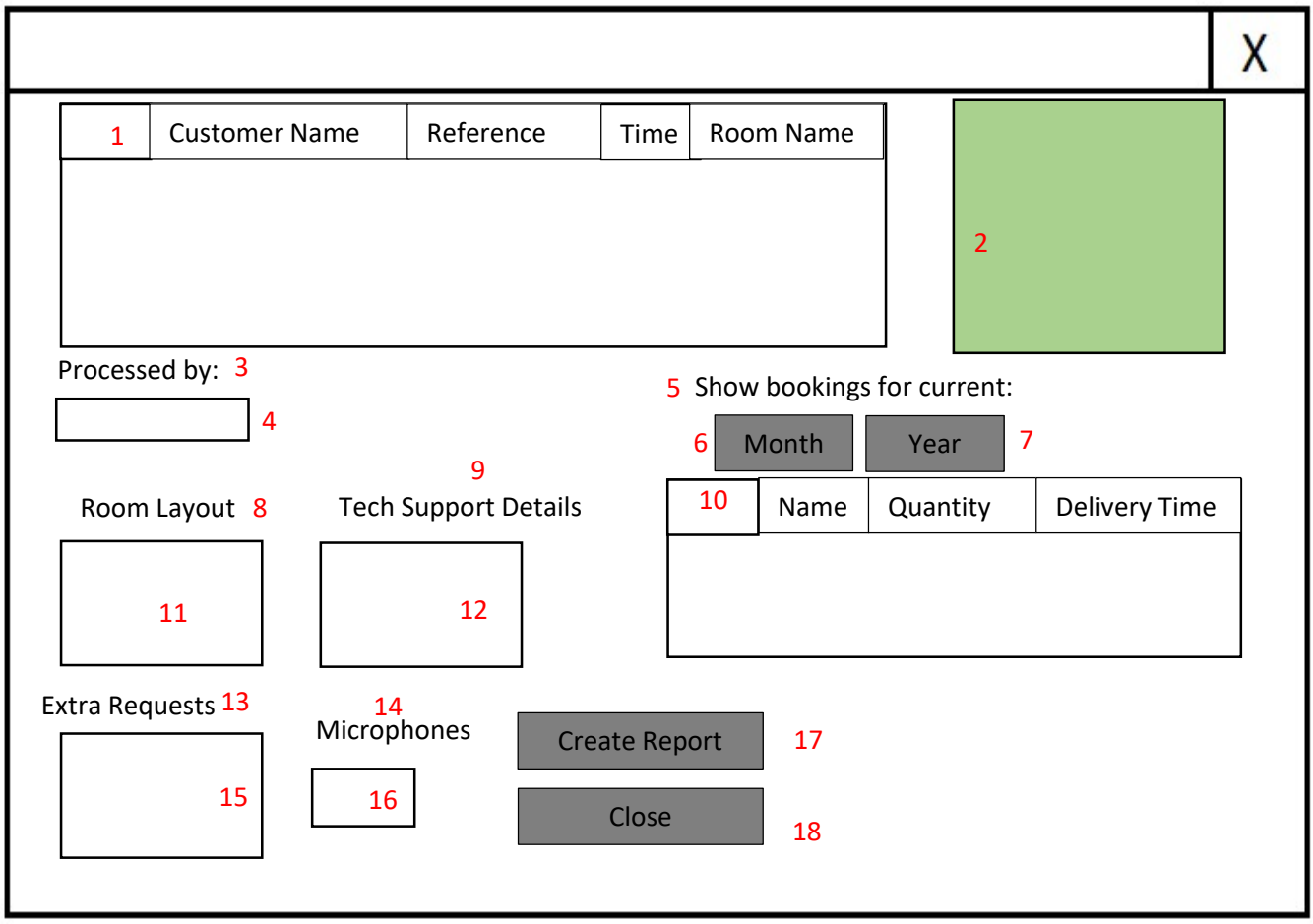
1. lblCustomer – A label containing the text “Select Customer”
2. lstCustomer – List box containing the names of customers
3. cbxNewCustomer – Check box which indicates whether or not the customer whose booking is being processed is new to the system
4. lblNewCustomer – Label containing the text “New Customer”
5. lblName – A label with the text “Name”
6. lblTelephone – A label with the text “Telephone Number”
7. lblMobile – A label with the text “Mobile Phone Number”
8. txtName – Text box containing the name of the customer
9. txtTelephone – Text box containing the telephone of the customer
10. txtMobile – Text box for containing the mobile phone of the customer
11. lblEmail – A label with the text “Email Address”
12. lblAddress – A label with the text “Postal Address”
13. lblPostcode – A label with the text “Postcode”
14. txtEmail – Text box containing the email address of the customer
15. txtAddress – Text box containing the postal address of the customer
16. txtPostcode – Text box containing the postcode of the customer
17. lblInvoiceName – A label with the text “Invoicing Contact Name”
18. lblInvoiceTelephone – A label with the text “Invoicing Telephone Number”
19. lblInvoiceMobile – A label with the text “Invoicing Mobile Phone Number”
20. txtInvoiceName – Text box containing the name of the customer
21. txtInvoiceTelephone – Text box containing the telephone of the customer
22. txtInvoiceMobile – Text box for containing the mobile phone of the customer
23. lblInvoiceEmail – A label with the text “Invoicing Email Address”
24. lblInvoiceAddress – A label with the text “Invoicing Postal Address”
25. lblInvoicePostcode – A label with the text “Invoicing Postcode”
26. txtInvoiceEmail – Text box containing the email address of the customer
27. txtInvoiceAddress – Text box containing the postal address of the customer
28. txtInvoicePostcode – Text box containing the postcode of the customer
29. lblReference – A label with the text “Booking Reference”
30. lblDate – A label with the text “Booking Date”
31. txtReference – Text box containing the booking reference fo a particular booking
32. dtpDate – Date-Time Picker containing the date of the booking
33. lblStartTime – Label containing the text “Start Time”
34. lblEndTime – Label containing the text “End Time”
35. lblRoom – Label containing the text “Select Room”
36. dtpStartTime – Date Time Picker containing the start time of the booking
37. dtpEndTime – Date Time Picker containing the end time of the booking
38. cmbRoom – Combo box containing the names of rooms
39. lblLayout – Label containing the text “Layout”
40. lblTechSupport – Label containing the text “Tech Support”
41. lblMicrophone – Label containing the text “Number of Microphones”
42. txtLayout – Text box storing the layout details of the room

43. txtTechSupport – Text box containing the details of any technical support required for the booking
44. nudMicrophone – Numeric-Up-Down storing the number of microphones requested for the booking
45. lblItem – Label containing the text “Select Item”
46. lblDeliveryTIme – Label containing the text “Delivery Time”
47. btnClearTime – Button with the text “Clear Time”. Clears the time stored in dtpDelivery
48. lstItem – List box containing the names of items which could be ordered
49. dtpDelivery – Date-time picker storing the delivery time of a particular item
50. lblQuantity – Label with the text “Quantity”
51. btnAddItem – Button with the text “Add Item”. Gets the ID of the item selected in lstItem from an array which holds the ID’s of every single item in lstItem. The ID is then used to get the name of the item from the catering file at a position equal to the ID of the selected item. This name is then placed in dgvItem, along with the delivery time and quantity given by dtpDelivery and nudQuantity respectively.

52. nudQuantity – Numeric-Up-Down storing the quantity of an item
53. btnRemoveSelected – Button with the text “Remove Selected”. Removes an item from dgvItem by deleting the row that has been selected by the user.
54. dgvItem – Data grid view with the following fields : “Name”, ”Quantity”, ”Delivery Time”. This component displays the items that have been ordered
55. lblExtraRequest – Label containing the text “Extra Requests”
56. txtExtra Request – Text box containing any extra requests concerning the catering
57. btnFinish – Button with the text “Finish”. Creates a new booking
58. btnCancel – Button with the text “Cancel”. Cancels the operation and closes the form

It should be noted that components 5 through to 28 are all contained within a panel called “pnlNewCustomer” which is not shown in the diagram above. The panel is used to group together related components.

Booking Calendar Form (frmBookingCalendar)



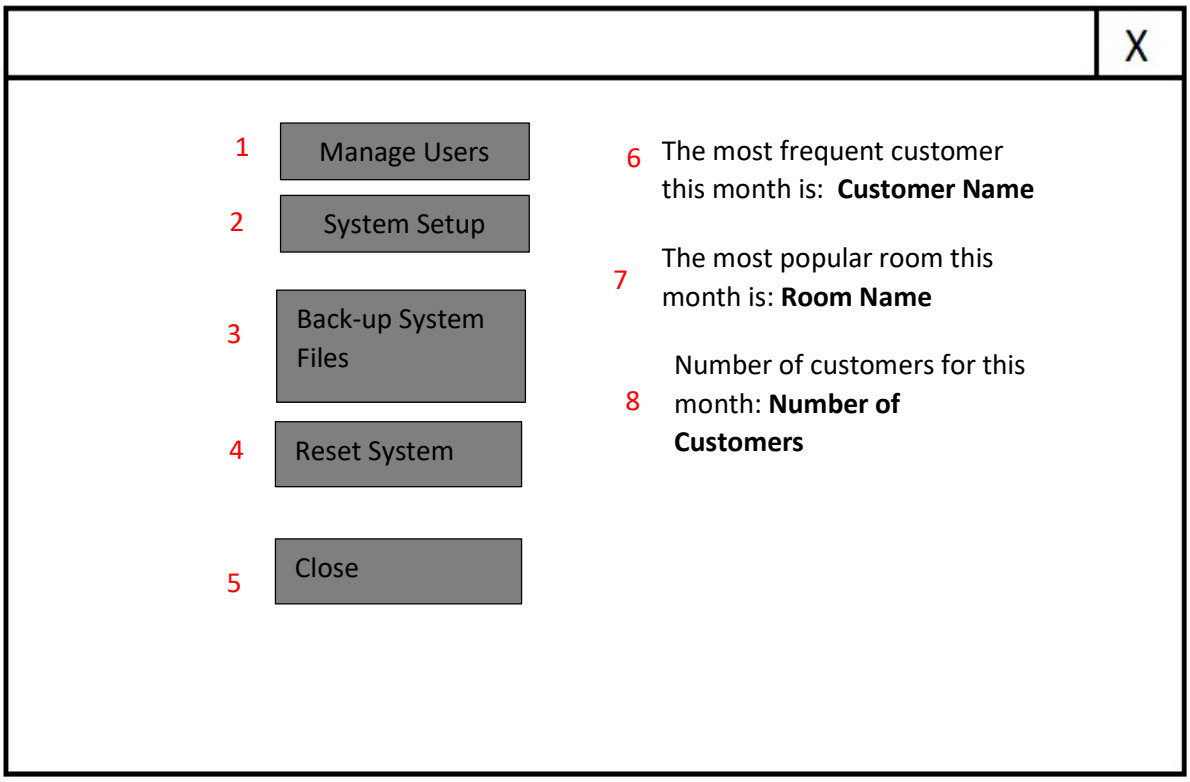
Along with being able to show bookings occurring on different dates the booking calendar can also create reports for that booking in the form of an editable Microsoft Word document.

1. dgvBookingResults – Data grid view storing the information about the bookings taking place on a certain date. The fields in the data grid are: ”Customer Name”, ”Reference”, ”Time”, ”Room Name”, and “BookingID”.The Booking ID field should be hidden
2. calBooking – Month Calendar which displays dates that can be selected using the mouse
3. lblUser – Label containing the text “Processed by:”
4. txtUser – Text box showing the name of the user who processed the booking
5. lblShowBooking – Label with the text “Show bookings for current :”
6. btnMonth – Button with the text “Month”. When pressed the button forces dgvBokingResults to show bookings which have occurred in the currently selected month. The program will search through the booking file to find any bookings which happen in the currently selected month in calBooking. If such a booking is found it will be added to dgvBookingResults
7. btnYear – Button with the text “Year”. When pressed the button forces dgvBokingResults to show bookings which have occurred in the currently

selected year. The program will search through the booking file to find any bookings which happen in the currently selected year in calBooking. If such a booking is found it will be added to dgvBookingResults

8. lblLayout – Label with the text “Room Layout”
9. lblTechSupport – Label with the text “Technical Support”
10. dgvOrder – Data grid view storing the order information for a booking. Contains the fields: “Name”, “Quantity”, “Delivery Time”
11. txtLayout – Text box storing the room layout information
12. txtTechSupport – Text box storing the technical support information
13. lblRequest – Label with the text “Extra Requests”
14. lblMicrophone – Label with the text “Microphones”
15. txtRequest – Text box containing the extra catering request information
16. txtMicrophone – Text box storing the number of microphones requested for the booking
17. btnCreateReport – Button with the text “Create Report”. Creates a Microsoft Word document containing all of the information about the selected booking. The program will take the information of the booking which has been selected in dgvBookingResults, along with the information in the text boxes and dgvOrder and add them to the Microsoft Word Document
18. btnClose – Button with the text

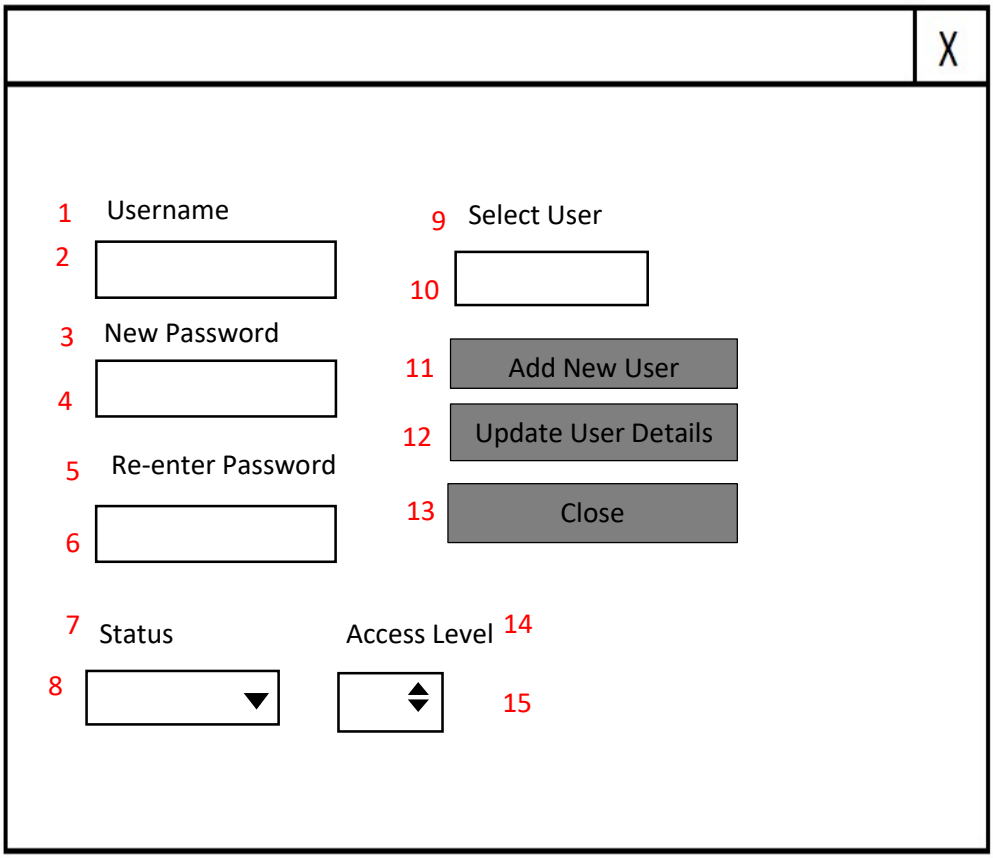
System Settings Form (frmSystemSettings)



This form provides system administrators with the facilities they require to manage the system

1. btnManageUsers – Button containing the text “Manage Users”. Displays the Manage Users Form
2. btnSetup – Button containing the text “System Setup”. Displays the System Setup form
3. btnBackUp – Button containing the text “Back-up System Files”. Shows a dialog which allows the user to select the location where the backed up file should be placed
4. btnReset – Button with the text “Reset System”. Deletes system files and shows the System Setup Form.
5. btnClose – Button with the text “Close”. Closes the form when pressed.
6. lblCustomer – Label containing the text “The most frequent customer this month is:” followed by the name of the most frequent customer in that month. The program shall search through the booking file and keep a count of all the times a particular customer ID occurs in the current month. The customer file is then accessed, at the position given by the ID with the highest number of counts in a month, in order to retrieve the name of the most frequent customer.
7. lblRoom – Label containing the text “The most popular room this month is: “ followed by the name of the most popular room. The program shall search through the booking file and keep a count of all the times a particular room ID occurs in the current month. The room file is then accessed, at the position given by the ID with the highest number of counts in a month, in order to retrieve the name of the most commonly used room.
8. lblCustomerNumber – Label containing the text “Number of customers for this month: “ followed by the number of customers who have made a booking in that month. The program will search through the booking file and keep count of all the customer who have made a booking in the current month.

Manage Users Form(frmManageUsers)

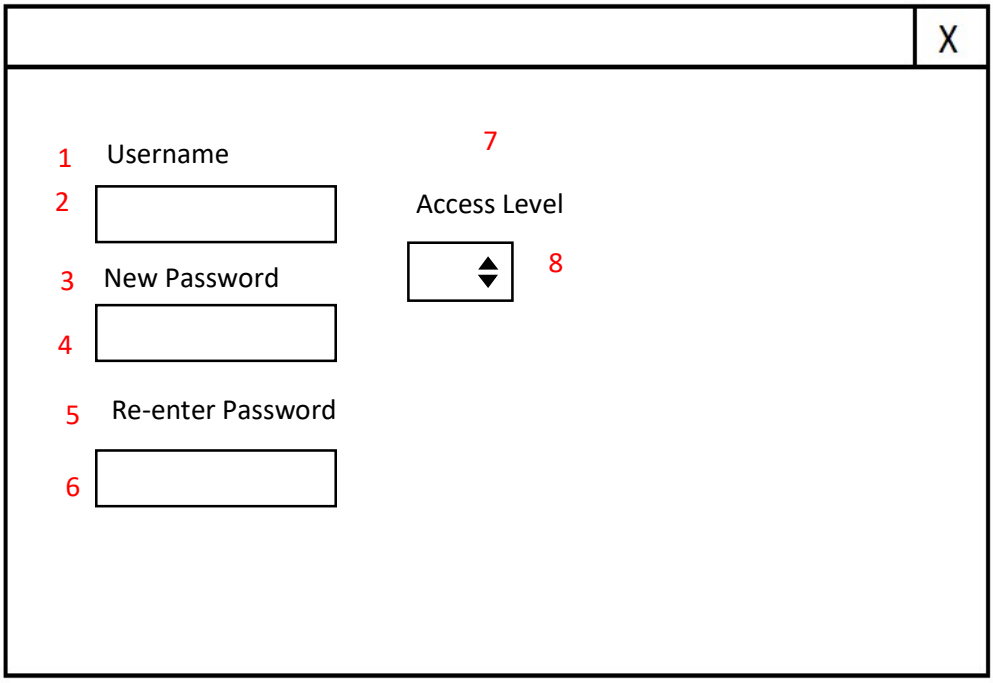


The Manage Users form allows the system administrator to make changes to a user's access level or their username. It also allows the users password to be changed.

1. lblUsername – Label with the text "Username"
2. txtUsername – Text box storing the username of the user
3. lblNewPassword – Label with the text "New Password"
4. mtbNewPassword – Masked text box for inputting the new password for a user
5. lblReEnterPass – Label with the text "Re-Enter Password"
6. mtbReEnterPass – Masked text box for inputting the new password. Used for verifying that the passwords entered in are identical
7. lblStatus – Label with the text "Status"
8. cmbStatus – Combo box which indicates whether or not the user is active
9. lblUser – Label with the text "Select User"
10. lstUser – List box containing the names of users
11. btnAddNewUser – Button containing the text "Add New User". Displays the Add New User Form
12. btnUpdate – Button containing the text "Update User Details". The updated user information is written to the user file at the position given by the ID of the currently selected user.

13. btnClose – Button containing the text “Close”. Closes the form
14. lblAccess – Label containing the text “Access Level”
15. nudAccess – Numeric-up-down storing the access level of the user

Add New User(frmNewUser)



This form allows the system administrator to add a new user to the system

1. lblUsername – Label with the text “Username”
2. txtUsername – Text box storing the username of the user
3. lblNewPassword – Label with the text “New Password”
4. mtbNewPassword – Masked text box for inputting the new password for a user
5. lblReEnterPass – Label with the text “Re-Enter Password”
6. mtbReEnterPass – Masked text box for inputting the new password. Used for verifying that the passwords entered in are identical
7. lblAccess – Label containing the text “Access Level”
8. nudAccess – Numeric-up-down storing the access level of the user
9. btnFinish – Button containing the text “Finish”. Writes the new information to the User file at the next free space.
10. btnClose – Button with the text “Close”. Closes the form

Pseudo code of Complex Algorithms

Log-In algorithm

This algorithm is used to decide whether or not to allow the user access to the system

```
Open User File
While User Not found And  i <= Total number of users Do
        Retrieve each user from the file from at position i in the user file
        Compare the username and password with input data
        If username <> input_username or Password <> input_password then
                User not found
        Else if the username and password do not match the input data
                User found
                Set Access level = User.AccessLevel
                Set UserID = User.UserID
        End if
i = i+1
End While
Close User File
If User found then
        Set reset text box values to default
        Hide Login Form
        Display Menu form
Else if User not found
        Display Error Message telling the User to try again
End if
```

Linking Bookings

The algorithm below describes the process of linking new bookings with older bookings using a linked list.

```
If Oldest Booking Pointer  = 0 then
```

Set Customer.BookingPointer = Number of bookings +1

Else Oldest Booking Pointer <> 0 then

Open Booking File

i = Oldest Booking Pointer

While i<>0 do

Retrieve Booking from file at position i

LastBookingPointer = i

i = NextBookingPointer

End While

Set NextBookingPointer = Number of bookings +1

Update booking file at position LastBookingPointer with new NextBookingPointer

End If

Date Changed Event

This algorithm describes the process that takes place when the selected date, in the booking calendar is changed.

Sub Date_Changed Event

Clear Information in the data grid

For i = 1 to Number of Bookings do

Retrieve Booking at position i in the file

If the Booking date is within the selected date range then

Open Room File

Get Room Name for the Room specified in the booking

Close Room File

Open Customer File

Get Customer name for the specified booking from customer file

Add the Customer Name, Booking reference, Start time, End Time and End Time to data grid Close Customer File

End if

End For

End Sub

Check For Clashes Function

This function checks if any booking is clashing with another one. The parameters for this function are RoomID, Booking Date, Start Time, End Time and the resource folder location. If a clash is found the function shall return all of the information about the booking, otherwise it will return an empty record with no information

Public Function CheckForClashes(ByVal BookingID As Integer, ByVal RoomID As Integer,
ByVal BookingDate As Date,
ByVal StartTime As Date, ByVal EndTime As Date,
ByVal ResourceFolder As String,
ByVal BookingControl As BookingControlData) As BookingData
Open Booking File

For i = 1 to NumberOfBookings

Retrieve Booking Information from position i in the file

If the Room ID =Current Room ID and the Booking date = Current Booking Date then

NumberOf Clashes = NumberOf Clashes +1

ClashingBooking(NumberOfClashes -1) = Booking Information

End If

End For

Close booking file

i = 0

While(Number Of clashes) <>0 and (Clash <>True) and (i<>Number of Clashes) do

If the time period for ClashingBooking(i) lies within that of the Current Booking then

Clash = True

End If

i=i+1

End While

Booking = Nothing

If Clash = True then

Return Clashing Booking(i-1)

Else Return Booking

Lock File Procedure

The lock file procedure ensures that other users cannot access the file during their use. This procedure should be used in conjunction with the unlock file procedure. Five variations of this procedure exist, each of which is used to lock a particular data file e.g Customer.dat. The procedure takes the parameters: Resource folder location and FileControl

Public Sub Lock_File(ByVal ResourceFolder As String, ByRef Control As ControlData)

Repeat

Sleep for 10 miliseconds

Open Control file

Retrieve Control data

Set File Control = Control data

Close File

Until FileControl.InUse = False

Set FileControl.InUse = True

Open Control File

Update File with FileControl

Close File

Unlock File Procedure

The Unlock File procedure has does the opposite of the Lock File procedure. This procedure ensures that once a user has finished using a file it is made available for other users. The parameters for this procedure are exactly the same as the lock file procedure. Five definitions of this procedure should exist in the actual system i.e. one for each data file.

Public Sub Unlock_File(ByVal ResourceFolder As String, ByRef Control As ControlData)

Set FileControl.InUse = False

Open Control File

Update File with FileControl

Close File

Hardware Specifications

In order for the system to run quickly and effectively the minimum hardware requirements must be met by the user.

The new room booking system requires the user to possess a keyboard, a mouse or trackpad and a monitor with a minimum resolution of 800 x 600 pixels. The system will support a monochrome display if necessary. It is recommended that the computer running the system has, at the very least, a 1GHz single-core processor and is running windows XP or higher as its operating system. It is also recommended that the computer has at least 1 GB of RAM. A magnetic hard drive or another form of non-volatile storage device, containing at least 100MB of free space, is necessary for the system to run. External hard drives and flash memory may be used in order to contain any back-ups.

System Security

The new system needs to be secure and prevent unauthorised access. As a means of achieving security the system will utilise two different security measures. The first way of preventing unauthorised access is to issue every user with a unique username and password which will then be used to log into the system. This means that an individual cannot make use of the system unless they were to steal the password. The system administrators will be in charge of issuing users with their usernames and passwords and they will also be in charge of changing passwords if the need arises.

The second security measure is the use of access levels to prevent users from causing too much damage to the system. Each genuine user is given a certain access level by the system admin and can range from 1 to 3, where 1 is the least restricted and 3 is the most restricted.

Overview of user access levels

Level 1:

Level 1 access is granted to system administrators only. This access level gives the ability to manage users i.e. add or remove users and reset their passwords. Level 1 grants the user access to all of the system facilities and all files used by the system.

Level 2:

The second highest access level allows the users access to system files meaning that they can add, remove or change their contents. Grants access to all non-administrative areas of the system.

Level 3:

This is the lowest and most restricted access level. Users with this access level cannot modify any files and only have access to the Booking file and basic details

regarding rooms such as the required setup. This access level is mostly suited towards the facilities staff that need to know what is happening in each room.

System Back Up and Restore

In order to lessen the effects of data loss the new system will allow the administrators to make a back-up of the existing system files. This will involve copying the files over to a new location, preferably an external hard drive. This backup function does not take into account settings files as those can easily be restored. If there is a problem with the system the administrators may use the system restore function to replace the faulty existing system files with the backed up files.

Testing Strategy

To ensure that the new booking system is fully functional and does not contain any bugs I will have to perform very thorough tests, the results of which will determine the overall effectiveness of the system. To do this I will need to have a suitable testing strategy which will allow me to carry out in-depth tests on the new system.

Below is an outline of the tests which will be carried out

- **Navigation -** The system relies on a number of forms which must be navigated easily and without any issues. The navigation tests will be used to ensure that the users can navigate between different forms, which means testing all of the buttons which link different forms, and also make sure that the relevant forms are loaded at the user's request.
- **Validation -** Validation is a very important part of the system as it ensures that all of the data entered into the system is correct and will not cause any conflicts within the system. Due to the importance of validation, the system must be tested to see if the appropriate type of validation is applied to the input data. The handling of incorrect input data must also be checked, this will involve making sure that the error messages provided by the system inform the user about what may be wrong with the input data so that they may be able to correct it.
- **Module/Functionality -** Module testing will involve testing each of the forms in order to determine whether they are fit to use. This will also help ensure that the code meets the design and behaves as intended.
- **Integration -** Integration testing will be used to ensure that all of the forms work well together and can pass data to each other without any errors.
- **Acceptance -** The last test involves handing over a copy of the system to the end user so that they may see whether or not they are happy with the system. These tests should be carried out last.