

Design Folio

EventSolutions Catalogue System



Mitch Blaser

May 2019

Table of Contents

EventSolutions Catalogue System

Page 1	Front Cover
Page 2	Table of Contents
Page 4	Design One
Page 4	Description
Page 5	Interface Design
Page 6	Object Descriptions
Page 8	Pseudo-Code
Page 9	Design Two
Page 9	Description
Page 10	Interface Design
Page 11	Object Descriptions
Page 12	Pseudo-Code
Page 13	Evaluation
Page 14	Login Window Design
Page 14	Description
Page 15	Interface Design
Page 16	Object Descriptions
Page 17	Data Dictionary
Page 18	Pseudo-Code
Page 19	Main Menu
Page 19	Description
Page 20	Interface Design
Page 21	Object Descriptions
Page 22	Data Dictionary
Page 23	Pseudo-Code
Page 24	Data Entry Form
Page 24	Description
Page 25	Interface Design
Page 26	Object Descriptions

Page 29	Data Dictionary
Page 31	Pseudo-Code
Page 34	Data View
Page 34	Description
Page 35	Interface Design
Page 36	Object Descriptions
Page 38	Data Dictionary
Page 39	Pseudo-Code
Page 40	Ribbon
Page 40	Menu Structure

Design One

Description

Design one provides a colourfully designed interface, with a ribbon to save space. Using this design allows for one form per category of data, but it could be lessened with the use of panels. The design doesn't use any elements that are too small, so in future if a tablet version is made it can be easily ported.

Different sections of the menu can be accessed by clicking on the tabs in the ribbon up the top of the form, and a record can be saved by clicking the save button, which is the default tab to show when the form is started for efficient access. Information is entered to the solution below the ribbon, with the use of either text boxes, drop-down list boxes or buttons. Information about items that have been added to the order are shown on the left.

Design One

Interface Design

EVCA - Design Option 1 - Interface Design - Hire Application Form (frmHire)

Font: Gill Sans (Light), Black

The screenshot shows a Windows application window titled "eventSolutions catalogue system". The title bar has a blue background with white text. On the right side of the title bar is a red "X" button. Below the title bar is a dark blue ribbon bar with the word "File" and two buttons labeled "Search/Sort" and "View Options". The main content area has a dark grey background. At the top left, it says "Record #008 Client Information". There are four text input fields: "First Name" and "Last Name" in a row, followed by "Email Address" and "Phone Number" in another row. Below this is a section titled "Equipment Information" with a dropdown menu for "Equipment Name" and a "Add" button. At the bottom left are dropdown menus for "Check-out Date" and "Return Date". To the right of these input fields is a table with two columns: "Hired Item" and "Price". A "Remove Item" button is located at the bottom of this table. The entire application window is enclosed in a black border.

Color: #1976d2

Color: #373737

Color: #004ba0

Color: #373737

Resolution: 1366x768, Scalable

All text in the solution will use the font “Gill Sans”, with a weight of “light”. The size of the text varies. For titles, a font size of 28 is used. For subtitles, a font size of 12 is used. Any other text is a font size of 8. The font colour is #000000. The title bar uses the EventSolutions logo in a picture box, and a text box next to it for the “catalogue system” part of the title with the same font (Gill Sans). It is held in a panel, which uses the colour #1976d2.

The ribbon uses a font size of 12 and is also stored in a panel. This panel’s colour is #004ba0, which is a slightly darker colour to the title bar. Colours for this design have been chosen from the Material Design Guidelines.

Mostly text boxes and/or dropdown lists are used for data entry, and use the same fill colour as the background panel - #373737, but has a white border, which is 2px thick and has the colour #ffffff.

Design One

Object Descriptions

Object Name	Object Type	Object Description
pnlTitle	Panel	Panel to hold the contents of the title bar
picLogo	Picture Box	Holds the company logo
lblTitle	Label	Holds the title text next to the company logo
picClose	Picture Box	A clickable picture box. Holds the close button so it can be styled to fit the theme of the application a bit better.
pnlRibbon	Panel	Holds the contents of the ribbon
btnFile	Button	Button to activate the “File” Subpanel
btnSearchSort	Button	Button to activate the “Search/Sort” Subpanel
btnView	Button	Button to activate the “View Options” Subpanel
pnlFile	Panel	Holds the contents of the File section of the ribbon
btnSave	Button	Saves the record to the data store
btnNew	Button	Creates a new record. Destroys all data already in the form if it’s not saved.
btnLoad	Button	Loads an existing record for editing.
pnlSearchSort	Panel	Holds the contents of the Search/Sort section of the ribbon
btnSortAsc	Button	Sort any lists on screen by ascending order.
btnSortDesc	Button	Sort any lists on screen by descending order.
btnSearchRecords	Button	Search for a record. Display a message box if more than one match is found. After a match is found, load information into form.
pnlViewOptions	Panel	Holds the contents of the File Section of the ribbon.
btnCalendar	Button	Selects the “Calendar” view option when available. Not available on this form because there’s no information to show.
btnList	Button	Selects the “List” view option when available. Not available on this form because there’s no information to show.
lblRecordNumber	Label	Shows the currently selected record number.
lblClientInfo	Label	Label for client information group box
grpClientInfo	Group Box	Container for the client information items. Keeps the form clean and organised.
lblFirstName	Label	Label for first name text box

Object Name	Object Type	Object Description
txtFirstName	Text Box	Holds the client's first name
lblLastName	Label	Label for the last name text box
txtLastName	Text Box	Holds the client's last name
lblEmailAddress	Label	Label for the email address text box
txtEmailAddresss	Text Box	Holds the client's email address
lblPhoneNumber	Label	Label for the phone number text box
txtPhoneNumber	Text Box	Holds the client's phone number
lblEquipmentInfo	Label	Label for the equipment information group box
grpEquipmentInfo	Group Box	Container for the equipment information items. Keeps the form clean and organised.
lblEquipment	Label	Label for the equipment combo box
cmbEquipment	Combo Box	Holds the currently selected piece of information
btnAddEquipment	Button	Adds equipment to the list of selected equipment and clears the box ready for the next entry.
lblCheckOut	Label	Label for the Check Out DateTimePicker
dtpCheckOut	DateTimePicker	Holds the time and date the order will be collected from the company
lblReturn	Label	Label for the Return DateTimePicker
dtpReturn	DateTimePicker	Holds the time and date the order will be returned to the company
lblItems	Label	Label for the items list box
lstItems	List Box	Holds a list of items that the client has hired.
lblPrice	Label	Label for the item prices list box
lstPrice	List Box	Holds a list of prices for the items that the client has hired. Lines up with lstItems.
btnRemoveItem	Button	Removes the selected item from the list box. Removes from both the price and equipment list box automatically.

Design One

Pseudo-Code

Pseudo-Code for Form Data Entry and Saving

nDays <— 0

When “New Record” is clicked:

 Clear all variables, text-boxes, date-time-pickers, drop-down-lists and list-boxes.

When “Save Record” is clicked:

 Check data for errors

 If data has errors:

 Show message box with error.

 Else:

 Write information to data structure ‘hireInformation’

End If

When “btnAddEquipment” is clicked:

 New item in ‘lstItem’ <— cmbEquipment.selectedItem

 priceTotal <— Look up price from selected item and multiply by nDays

 New item in ‘lstPrice’ <— priceTotal

When “dtpCheckOut” or “dtpReturn” is changed:

 nDays <— calculate how many days is included in the selected range

When “btnRemoveItem” is clicked:

 selectedIndex <— which item is selected in the list box?

 delete ‘lstItem[selectedIndex]’

 delete ‘lstPrice[selectedIndex]’

Design Two

Description

Design two uses an interface which is slightly less visually appealing, but makes the solution's available functions easier to find and use in less time, but the location and functionality of those buttons could vary more often based on the form that is active unlike the ribbon which is static.

This Interface Design is for the hire data entry screen, so though it isn't visible the different forms will be accessed from a main menu. This means that slightly more clicks can be expected to do the same task, but is still much more efficient than the previous solution. Loading/Saving different records of the form can be done on the bottom right, and all search/sort data is entered through pop-up message boxes. Data is entered on the left in text boxes and drop-down lists, and will not save until the Save button is pressed. Data is then processed and added to the list box which is central to the form.

Design Two

Interface Design

EVCA - Design Option 1 - Interface Design - Hire Application Form (frmHire)

The screenshot shows a Windows application window titled "event**solutions** catalogue system". The title bar is blue with white text. In the top right corner is a red "X" button. The main interface has a dark grey background. On the left side, there are three input fields: "Client Name" with a "Search" button and an "Add New" button below it; "Equipment Name" with a "Search" button and an "Add" button below it; and two date pickers: "Check-Out Date" and "Return Date". On the right side, there is a vertical column of buttons: "Load #", "Search", and "Save". At the bottom left, it says "Resolution: 1366x768, Scalable".

All text in the solution will use the font “Gill Sans”, with a weight of “light”. The size of the text varies. For titles, a font size of 26 is used. For buttons, a font size of 16 is used. Any other text is a font size of 10. The font colour is #000000.

The title bar uses the EventSolutions logo in a picture box, and a text box next to it for the “catalogue system” part of the title with the same font (Gill Sans). It is held in a panel, which uses the colour #1976d2.

Colours for this design have been chosen from the Material Design Guidelines. Mostly text boxes and/or dropdown lists are used for data entry, with some date-time-pickers, and use the same fill colour as the background panel - #373737, but has a white border, which is 2px thick and has the colour #ffffff.

Design Two

Object Descriptions

Object Name	Object Type	Object Description
pnTitle	Panel	Panel to hold the contents of the title bar
picLogo	Picture Box	Holds the company logo
lblTitle	Label	Holds the title text next to the company logo
picClose	Picture Box	A clickable picture box. Holds the close button so it can be styled to fit the theme of the application a bit better.
lblClientName	Label	Holds the currently selected client name
btnSearchClient	Button	Opens a message box and prompts for a search term. Once found, it puts the result into the Client Name label and into the record's data.
btnAddClient	Button	Opens a message box and prompts for new info for a client. Automatically adds to the form and record's data once it has been made.
lblEquipmentName	Label	Holds the currently selected equipment name
btnSearchEquipment	Button	Opens a message box and prompts for a search term. Once found, it puts the result into the Equipment Name label.
btnAddEquipment	Button	Opens a message box and prompts for new info for equipment. Once it has been made the data will be automatically inserted into the form.
btnAddEquipmentToList	Button	Adds the currently selected equipment item to the hire list. (lstItems)
dtpCheckOut	DateTimePicker	Holds the time the equipment will be collected from the company
dtpReturn	DateTimePicker	Holds the time the equipment will be returned to the company.
lstItems	ListBox	Holds all the items to be hired and their prices, along with a total as the last item in the list.
btnRemoveSelectedItem	Button	Remove the currently selected item in the list box.
btnLoadRecord	Button	Load a specific record ID. Data is entered into a message box.
btnSearchRecord	Button	Search for and load a specific record. Data is entered into a message box.
btnSaveRecord	Button	Save the contents of the form into a record and write out to the file.

Design Two

Pseudo-Code

Pseudo-Code for Saving data to a File

Imports System.IO

fileLines <— 4 'The number of lines to write out for each record
 '1. Client Data, 2. Equipment Array, 3. Check-Out Date, 4. Return Date.

Path <— “C:\EVCS\Data\HireList” 'Path to store file.

HireIO <— New StreamWriter(Path)

If btnSaveRecord is clicked:

If data is correct:

 'Lookup client ID by name
 HireIO.WriteLine <— Clients[FullName][lblClientName.Text]
 'Write entirety of “selected equipment” array to file
 HireIO.WriteLine <— EquipmentID[]
 'Write selected date to file
 HireIO.WriteLine <— dtpCheckOut.SelectedDate
 'Write selected date to file
 HireIO.WriteLine <— dtpReturn.SelectedDate

Else:

 MsgBox(error)

HireIO.CloseFile

Evaluation

Design 1 vs Design 2

Criteria	Area	Requirement	Design 1 x/5	Design 2 x/5
Time - No more than three clicks to get to any of the application's functions.	Efficiency	NFR	5	3
Clarity - Is the solution easily readable? Is the font a good size?	Effectiveness	NFR	5	4
Clarity - How easy is it to locate all of the solution's functions?	Efficiency	NFR	4	5
Robustness - Are all inputs validated?	Effectiveness	FR	5	5
Portability - Is the form able to scale correctly?	Effectiveness	FR	5	5
Robustness - Are all outputs correct?	Efficiency	FR		
Usability - Is the solution easy to pick up and use for the first time?	Efficiency	NFR	3	4
Time - The project must be completed by the 21st August 2019	Efficiency	NFR	Can not be determined by design.	Can not be determined by design.
Attractiveness - The solution must look clean and modern	Effectiveness	NFR	5	3
Usability - Quickly and easily read, edit and write the required data, with the most efficient way of entering the data taking priority.	Efficiency	FR	5	4
TOTAL:			37	33

Design One has been chosen over Design Two, because it is more visually appealing and is easier to access all of the functions of the solution without it feeling too cluttered. All inputs are validated when they are entered, and data is quickly readable for the end user.

Login Window

Description

The Login Window is a relatively simple form. It's job is to get a username and password from the user, and look up the user and it's permissions from a secure data store inside the program's files.

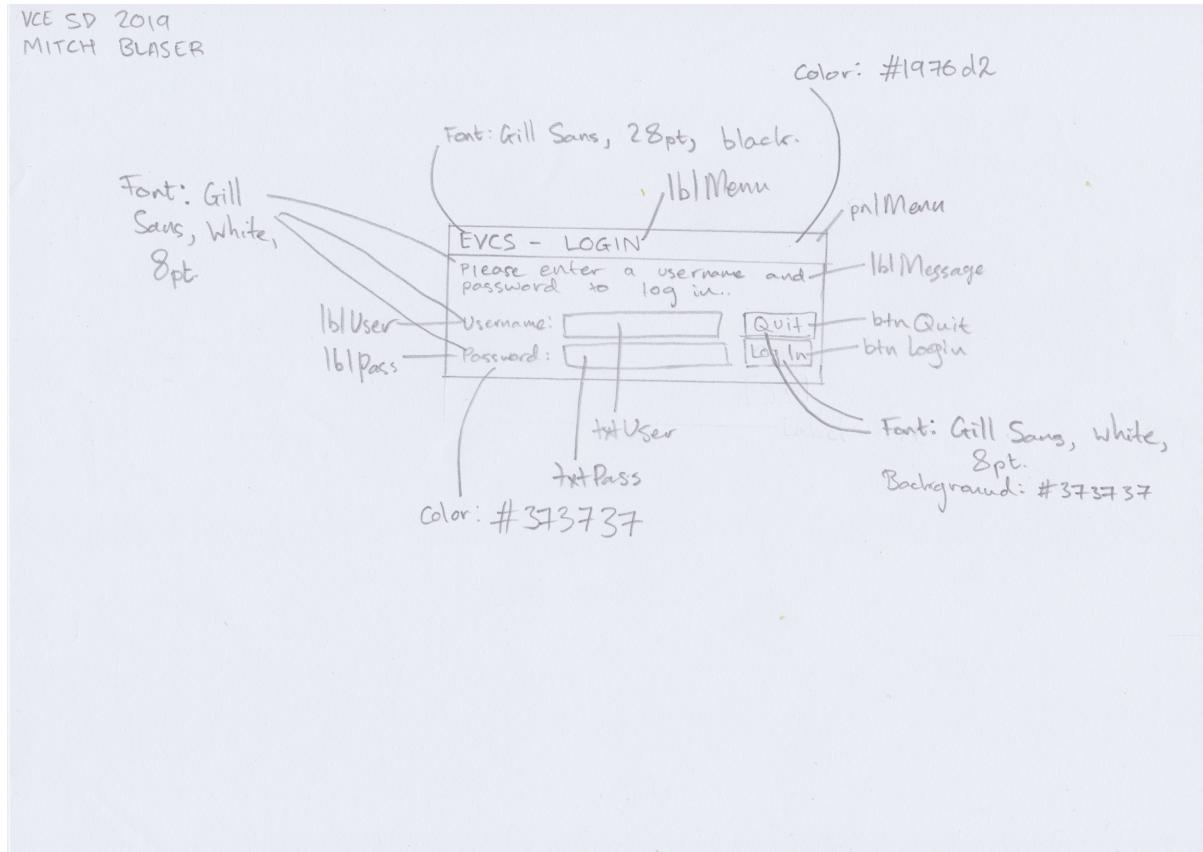
If the user account doesn't exist or a password is typed incorrectly, two different error messages will be displayed. Respectively, the first one will say "Error: User account does not exist!" and the other will say "Error: Incorrect password. Please try again."

Once the data the user entered into the form correctly matches one of the entries (it will be matched with something similar to a linear search routine), the Main Menu form will be shown and the Login form will be closed, with it's variables and text boxes completely cleared, ready for the next time a user wants to log in. If the user account does not match, after the error is shown, the password field is cleared and the user is prompted to try to log in again.

Login Window

Interface Design

VCE SD 2019
MITCH BLASER



Login Window

Object Descriptions

Object Name	Object Type	Object Description
frmLogin	Form	The form that the user uses to log in to the rest of the solution. Color: #373737
pnlMenu	Panel	A panel which holds the top menu bar items
lblMenu	Label	The title of the form. Font: Gill Sans, White, 28pt.
lblMessage	Label	The message that prompts the user to log in to the solution. Font: Gill Sans, White, 8pt.
lblUser	Label	The message that indicates which text box is used for the username.
lblPass	Label	The message that indicates which text box is used for the password.
txtUser	Text Box	The text box which allows the user to enter their username.
txtPass	Text Box	The text box which allows the user to enter their password.
btnLogin_Click	Button (Click Event)	When this button is clicked, the solution checks if the username and password match an entry in the solution's data store.
btnQuit_Click	Button (Click Event)	The button which quits the entire solution when clicked.

Login Window

Data Dictionary

Variable Name	Variable Type	Access Level	Description
username	String	Private	Holds the value of txtUser.Text when btnLogin is clicked.
password	String	Private	Holds the value of txtPass.Text when btnLogin is clicked.
userKeys	StreamReader	Private	The file which stores all usernames and passwords.
userInfo	Structure	Public	Holds records for each user which is allowed to log in, their password, and their status.
usernames()	String()	Public	Holds each record's username
passwords()	String()	Public	Holds each record's password
permissions()	String()	Public	Holds each record's permissions (either "admin" or "standard")
UserEntries	Integer	Private	Holds the amount of users in the file to avoid looping 50 times every form load.
ItemCounter	Integer	Private	Iterator for the linear search algorithm. Represents the current user record that the solution is searching.
authenticatedUser	Integer	Public	If the linear search algorithm finds a match, authenticatedUser holds the index of the user in userInfo.
i	Integer	Private	Iterator for the file read loop.

Login Window

Pseudo-Code

```
Declare username, password as String
Open file "userkeys.evdb" As userKeys
Declare Public Structure UserInfo
    Declare usernames(50) As String
    Declare passwords(50) As String
    Declare permissions(50) As String
End Structure
Declare UserEntries As Integer
Declare ItemCounter As Integer
Declare Public authenticatedUser As Integer

frmLogin_Load_Event
UserEntries = 0
Loop i until end of file:
    usernames(i) = userKeys.ReadLine()
    Increment i by 1
    passwords(i) = userKeys.ReadLine()
    Increment i by 1
    permissions(i) = userKeys.ReadLine()
    Increment I by 1
    Increment UserEntries by 1
End Loop
userKeys.Close()

btnLogin_Click_Event
username = txtUser.Text
password = txtPass.Text
ItemCounter = 0
authenticatedUser = -1
While ItemCounter <= UserEntries And authenticatedUser == -1
    If usernames(itemCounter).ToUpper == username.ToUpper() And passwords(itemCounter).ToUpper == password.ToUpper()
        authenticatedUser = ItemCounter
    End If
    Increment ItemCounter by 1
End While
If authenticatedUser == -1
    MessageBox("Username and/or Password Incorrect.")
Else
    txtUser.Text = ""
    txtPass.Text = ""

    frmMainMenu.Show()
    Me.Close()
End If
```

Main Menu

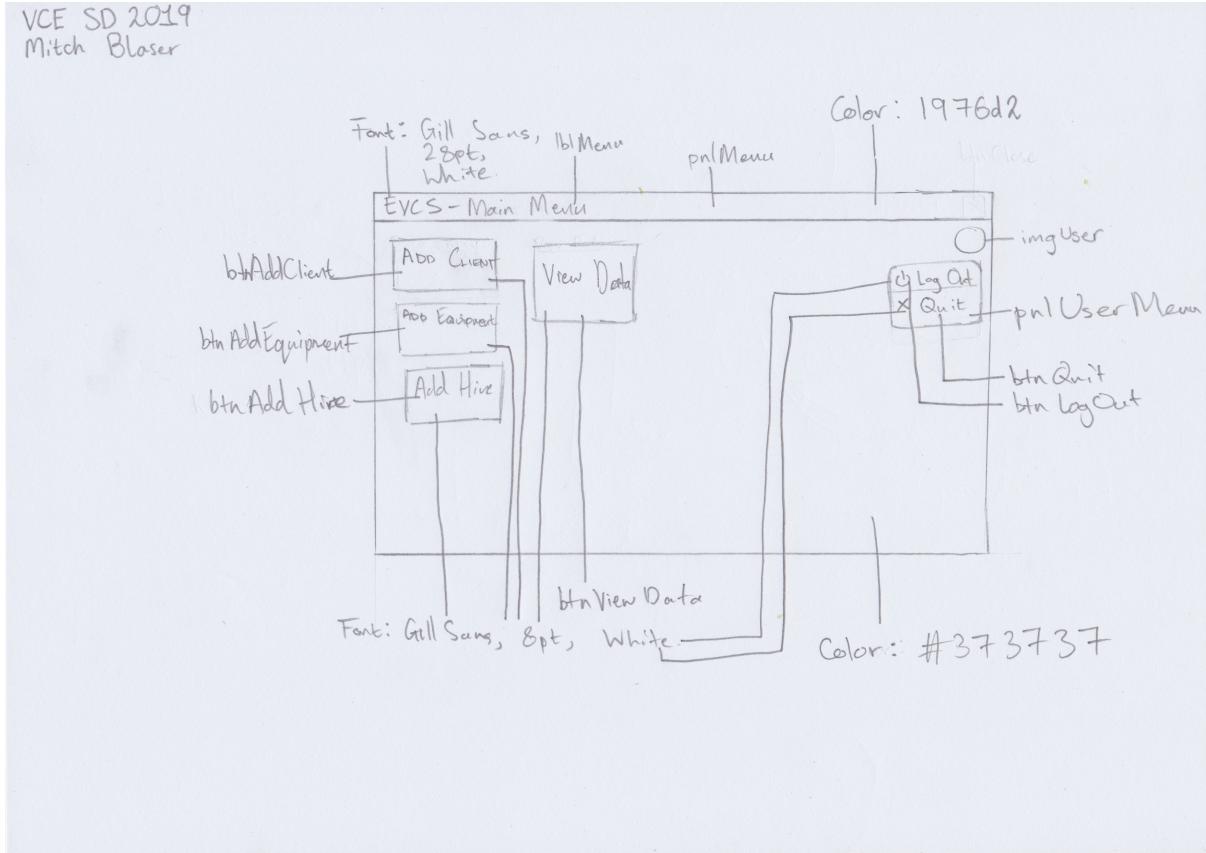
Description

The Main Menu appears after the user has successfully logged in. It's job is to allow the user to select which mode the user wants to access - Data Entry mode or Data View mode.

There aren't many objects in this form, as it just acts as a menu to launch into the main functions of the application. It consists of some buttons, which allow the user to select the part of the solution that they want to use. Once a button has been clicked, the form that links to the specific button is shown (either the Data Entry or Data View forms), but the Main Menu form isn't ever closed or hidden. It stays open so when the user is finished with their task and they close the window, they see the Main Menu again and can either log out or choose another task. It also has an image box, which is up in the top right corner. This displays the currently logged in user's profile picture, and is also clickable to show a drop-down menu with the option to log out. Once the user has opted to log out, the Main Menu form is finally closed, and the login window appears once again.

Main Menu

Interface Design



Main Menu

Object Descriptions

Object Name	Object Type	Object Description
frmMainMenu	Form	The form that the user uses to launch the rest of the solution and its features. Color: #373737
pnlMenu	Panel	A panel which holds the top menu bar items
lblMenu	Label	The title of the form. Font: Gill Sans, White, 28pt.
imgUser_Click	ImageBox (Click Event)	Holds the picture of the user that is currently logged in. When clicked, pnlUserMenu is set to be visible.
pnlUserMenu	Panel	The container for the drop-down menu under imgUser.
btnLogOut	Button (Click Event)	Logs out user when clicked. Closes the main menu form and goes back to the formLogin form.
btnQuit	Button (Click Event)	Quits the entire solution when the button is clicked.
btnAddClient_Click	Button (Click Event)	Opens the Data Entry Form (frmEnterData) and sets the global variable to “Client Mode”.
btnAddEquipment_Click	Button (Click Event)	Opens the Data Entry Form (frmEnterData) and sets the global variable to “Equipment Mode”.
btnAddHire_Click	Button (Click Event)	Opens the Data Entry Form (frmEnterData) and sets the global variable to “Hire Mode”.
btnViewData_Click	Button (Click Event)	Opens the Data Viewer Form (frm ViewData).

Main Menu

Data Dictionary

Variable Name	Variable Type	Access Level	Description
dataEntryMode	String	Public	Tells the data entry form (frmEnterData) which panel to display. Logic code for this is inside of the other form.
popupMenu	Boolean	Private	Holds the state of the drop-down panel (open = true, closed = false)

Main Menu

Pseudo-Code

```
Declare Public dataEntryMode As String
Declare popupMenu As Boolean = False

frmMainMenu_Load_Event
If file "userFiles/user_" & frmLogin.authenticatedUser & ".jpeg" exists
    imgUser.Image = Image.FromFile("userFiles/user_" & frmLogin.authenticatedUser & ".jpeg")
Else
    imgUser.Image = Image.FromFile("userFiles/defaultImage.jpeg")
End If


```

Data Entry Form

Description

The Data Entry form's job is to allow the user to enter data into the solution's various data stores. The correct data store is chosen from which button is clicked in the Main Menu form. The button sets a global variable, and that is what dictates which set of elements (buttons, text boxes, etc.) are shown to the end user. These elements are stored inside different panels, and when the form is loaded, there is code which decides which panel to show based off the global variable from the main menu's value.

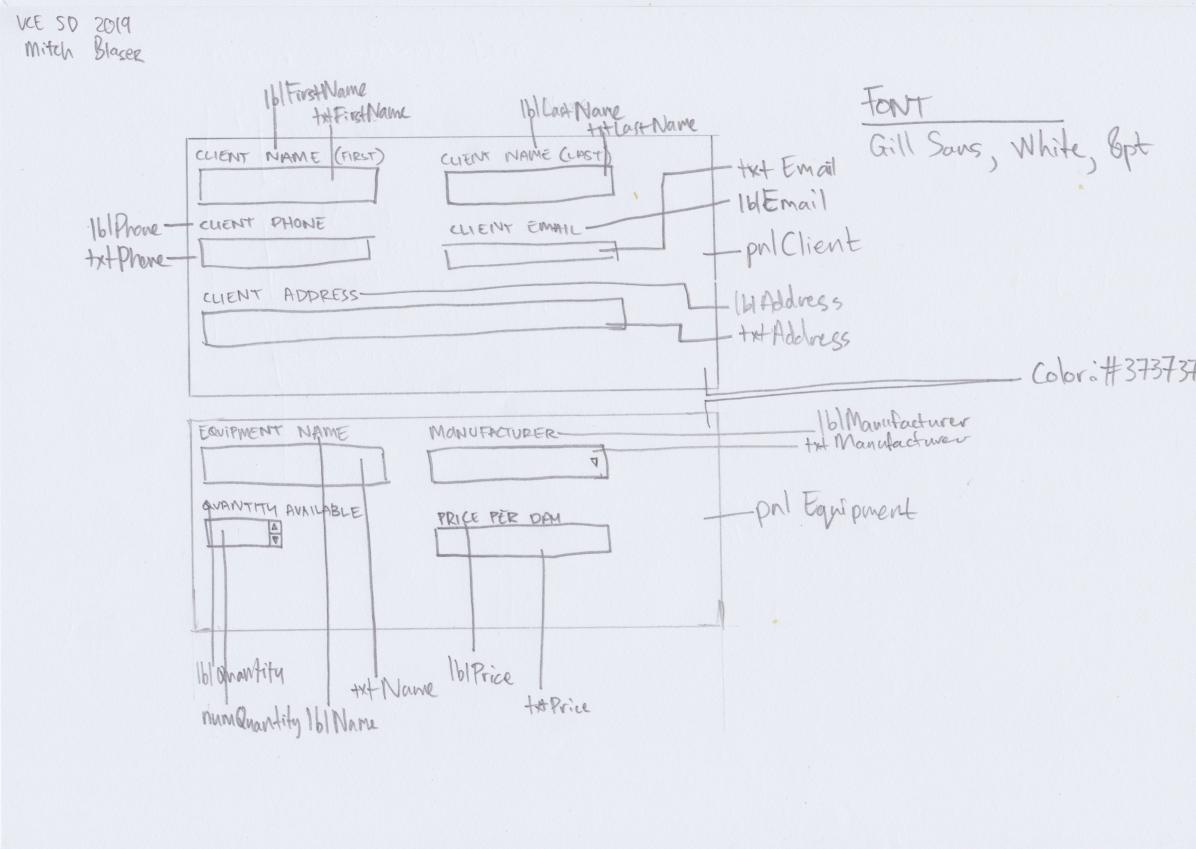
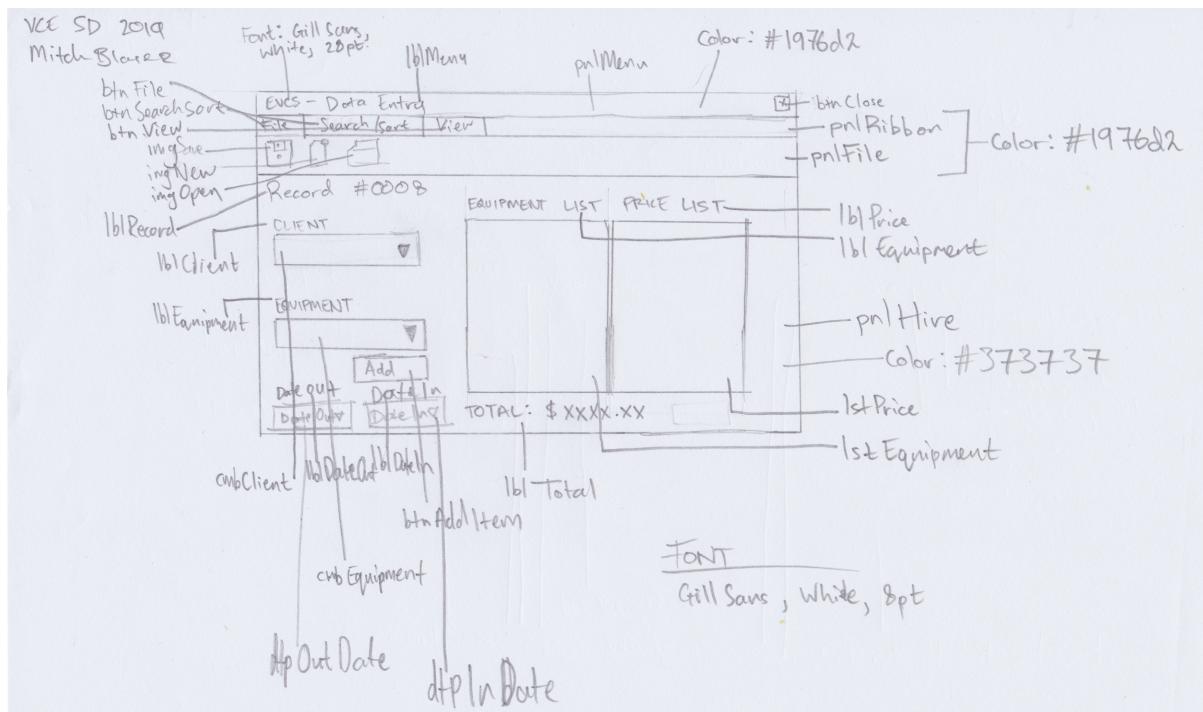
Once the correct panel has been shown, the user can enter the correct information into the text boxes and drop down lists in the form. They then press the save button to finalise the record, and write it to the relevant data store.

Through the Search/Sort tab in the ribbon, the user can also opt to search for a record that has already been created and saved, and load it back into memory and edit the record.

Some options in the ribbon are not accessible to this form, and are only used in the Data View form, such as “sorting method”.

Data Entry Form

Interface Design



Data Entry Form

Object Descriptions

Object Name	Object Type	Object Description
frmEnterData	Form	The form that the user uses to enter data into the solution. Color: #373737
pnlMenu	Panel	The panel which holds the top menu bar items.
lblMenu	Label	The title of the form.
btnClose_Click	Button (Click Event)	Closes the form when the button is clicked.
pnlRibbon	Panel	Holds the ribbon items. Color: #1976d2
btnFile_Click	Button (Click Event)	Shows the File menu panel in the ribbon.
btnSearchSort_Click	Button (Click Event)	Shows the Search/Sort menu panel in the ribbon
btnView_Click	Button (Click Event)	Shows the View menu panel in the ribbon
pnlFile	Panel	Holds the buttons inside of the File menu
imgSave_Click	Image Box (Click Event)	Saves the record and writes out to the data file
imgNew_Click	Image Box (Click Event)	Creates a new record and clears all form elements.
imgOpen_Click	Image Box (Click Event)	Opens an existing record based on a record index number. Input number is inputted from a Message Box.
pnlSearchSort	Panel	Holds the buttons inside of the Search/Sort menu.
imgSearch_Click	Image Box (Click Event)	Prompts user for a search term via an Input Box, and launches a linear search algorithm. Once a result is found, the relevant record appears in the form's elements.
imgSortByDate	Image Box (Click Event)	If list data is displayed on screen, this button will launch an selection sort algorithm that rearranges items inside of all list boxes to be in order of newest to oldest date.
imgSortAlphabetical	Image Box (Click Event)	If list data is displayed on screen, this button will launch an selection sort algorithm that rearranges items inside of all list boxes to be in order of a to z.
pnlView	Panel	Holds the buttons inside of the View menu.
imgListView	Image Box (Click Event)	If the active form is the Data View form, using this button will rearrange all data points to be shown in a list form, instead of calendar.
imgCalendar	Image Box (Click Event)	If the active form is the Data View form, using this button will rearrange all data points to be shown in calendar form, instead of a list.

Object Name	Object Type	Object Description
pnlHire	Panel	Holds the entirety of the Data Entry - Hire section.
lblRecord	Label	Gets updated with the current record's ID each time a new record is loaded as active.
lblClient	Label	Header label to signify what the combo box under it is used for.
lblEquipment	Label	Header label to signify what the combo box under it is used for.
cmbClient	Combo Box	On form load, the combo box gets populated with every client in the system. It allows the user to select the required client easily, without typing in their name every time.
cmbEquipment	Combo Box	On form load, the combo box gets populated with every piece of equipment in the system. Once a user selects an item, they can use the button below it (btnAddItem) to add it to the list.
btnAddItem_Click	Button (Click Event)	When the user clicks this button, it adds the selected piece of equipment to the array and list box of equipment.
lblDateOut	Label	Header label to signify what the datetimepicker under it is used for.
lblDateIn	Label	Header label to signify what the datetimepicker under it is used for.
dtpDateOut	DateTimePicker	Specifies the date that the equipment is checked out and marked as unavailable in the system. Is also used to calculate price per day, along with dtpDateIn.
dtpDateIn	DateTimePicker	Specifies the date that the equipment is checked back in and marked as available for hire in the system. Is also used to calculate price per day, along with dtpDateOut.
lblEquipment	Label	Header label to signify what the list box under it is used for.
lblPrice	Label	Header label to signify what the list box under it is used for.
lstPrice	List Box	A list of prices of equipment that has been hired, relative to lstEquipment.
lstEquipment	List Box	A list of each piece of equipment that has been hired.
lblTotal	Label	Holds the total cost for the total period of days. This gets updated each time a new piece of equipment is added to the list.

Object Name	Object Type	Object Description
pnlClient	Panel	Holds the entirety of the Data Entry - Client section.
lblFirstName	Label	Header label to signify what the text box under it is used for.
lblLastName	Label	Header label to signify what the text box under it is used for.
lblPhone	Label	Header label to signify what the text box under it is used for.
lblEmail	Label	Header label to signify what the text box under it is used for.
lblAddress	Label	Header label to signify what the text box under it is used for.
txtFirstName	Text Box	Stores the current record's first name.
txtLastName	Text Box	Stores the current record's last name.
txtPhone	Text Box	Stores the current record's phone number.
txtEmail	Text Box	Stores the current record's email address.
txtAddress	Text Box	Stores the current record's house address.

Object Name	Object Type	Object Description
pnlEquipment	Panel	Holds the entirety of the Data Entry - Equipment section.
lblName	Label	Header label to signify what the text box under it is used for.
lblManufacturer	Label	Header label to signify what the text box under it is used for.
lblQuantity	Label	Header label to signify what the numericupdown under it is used for.
lblPrice	Label	Header label to signify what the text box under it is used for.
txtName	Text Box	Holds the name of the piece of equipment.
txtManufacturer	Text Box	Holds the name of the equipment's manufacturer.
numQuantity	NumericUpDown	Holds the amount of that piece of equipment that are owned by the company.
txtPrice	Text Box	Holds the price per day of the item.

Data Entry Form

Data Dictionary

Variable Name	Variable Type	Access Level	Description
clientStructure	Structure	Public	Structure to hold client information.
clientFirstName	String	Public	Holds the client's first name
clientLastName	String	Public	Holds the client's last name
clientPhone	String	Public	Holds the client's phone number
clientEmail	String	Public	Holds the client's email address
clientAddress	String	Public	Holds the client's home address
equipmentStructure	Structure	Public	Structure to hold equipment information
equipmentName	String	Public	Holds the name of the specific piece of equipment.
equipmentManufacturer	String	Public	Holds the name of the manufacturer of the specific piece of equipment.
equipmentQuantity	Integer	Public	Holds the amount of equipment that's available from the company to hire.
equipmentPrice	Double	Public	Holds the amount per day to hire one piece of the equipment.
hireStructure	Structure	Public	Structure to hold hire information.
hireClient	Integer	Public	The client who has hired the equipment. Links back to the client data structure.
hireOut	Date	Public	Check out date.
hireIn	Date	Public	Check in date. Is used along with the check out date to calculate the amount of days that the client is hiring the equipment for, so we can charge the correct amount of money.
equipmentLength	Integer	Public	Amount of items being hired. This needs to be there for reading in the files.
equipment()	String()	Public	Array of equipment that has been hired in each specific record.
clientFile	StreamWriter & StreamReader	Private	Handles reading and writing to the client data file.
equipmentFile	StreamWriter & StreamReader	Private	Handles reading and writing to the equipment data file.
hireFile	StreamWriter & StreamReader	Private	Handles reading and writing to the hire data file.
clientLength	Integer	Private	The amount of entries in the client data file

Variable Name	Variable Type	Access Level	Description
equipmentLength	Integer	Private	The amount of entries in the equipment data file.
hireLength	Integer	Private	The amount of entries in the hire data file.
i	Integer	Private	Outer loop iterator. Private to itself.
f	Integer	Private	Inner loop iterator. Private to itself.
selectedEquipment	String	Private	Holds the item that's selected inside the combo box. This is used when it adds an item from the combo box into the list box.
days	Integer	Private	The difference between the check out and in dates. Used to calculate the amount to charge the client.

Data Entry Form

Pseudo-Code

```

Declare Public Structure clientStructure
    Declare clientFirstName As String
    Declare clientLastName As String
    Declare clientPhone As String
    Declare clientEmail As String
    Declare clientAddress As String
End Structure

Declare Public Structure equipmentStructure
    Declare equipmentName As String
    Declare equipmentManufacturer As String
    Declare equipmentQuantity As Integer
    Declare equipmentPrice As Double
End Structure

Declare Public Structure hireStructure
    Declare hireClient As Integer
    Declare hireOut As Date
    Declare hireIn As Date
    Declare equipmentLength As Integer
    Declare equipment(60) As String
End Structure

Declare Public clientLength, equipmentLength, hireLength As Integer = 0

frmEnterData_Load_Event:
If frmMainMenu.dataEntryMode == "Client"
    pnlClient.Visible = True
    pnlEquipment.Visible = False
    pnlHire.Visible = False
Else If frmMainMenu.dataEntryMode == "Equipment"
    pnlClient.Visible = False
    pnlEquipment.Visible = True
    pnlHire.Visible = False
Else If frmMainMenu.dataEntryMode == "Hire"
    pnlClient.Visible = False
    pnlEquipment.Visible = False
    pnlHire.Visible = True
End If

Open File "dataStore/clientList.evdb" As clientFile
Open File "dataStore/equipmentList.evdb" As equipmentFile
Open File "dataStore/hireList.evdb" As hireFile

Loop i until end of clientFile
    clientStructure(i).clientFirstName = clientFile.ReadLine()
    clientStructure(i).clientLastName = clientFile.ReadLine()
    clientStructure(i).clientPhone = clientFile.ReadLine()
    clientStructure(i).clientEmail = clientFile.ReadLine()
    clientStructure(i).clientAddress = clientFile.ReadLine()
    Increment i by 1
End Loop
clientLength = i
clientFile.Close()

Loop i until end of equipmentFile
    equipmentStructure(i).equipmentName = equipmentFile.ReadLine()
    equipmentStructure(i).equipmentManufacturer = equipmentFile.ReadLine()
    equipmentStructure(i).equipmentQuantity = equipmentFile.ReadLine()
    equipmentStructure(i).equipmentPrice = equipmentFile.ReadLine()
    Increment i by 1
End Loop
equipmentLength = I
equipmentFile.Close()

Loop i until end of hireFile
    hireStructure(i).hireClient = hireFile.ReadLine()
    hireStructure(i).hireOut = hireFile.ReadLine()
    hireStructure(i).hireIn = hireFile.ReadLine()
    hireStructure(i).equipmentLength = hireFile.ReadLine()

    Loop while f <= equipmentLength
        hireStructure(i).equipment(f) = hireFile.ReadLine()
        Increment f by 1
    End Loop
End Loop
hireLength = I
hireFile.Close()

```

```



```

```

        subFindIndex = i
    End If
End While
End If

If subFindIndex != -1
    Load data from relevant structure and fill the required elements (text boxes, lists, etc.)
End If

btnAddItem_Click_Event
selectedEquipment = cmbEquipment.SelectedItem

Declare days as Integer = DateDiff(DateInterval.Day, dtpDateOut.Date, dtpDateIn.Date)

lstEquipment.Items.Add(equipmentStructure(selectedEquipment).equipmentManufacturer & " " &
equipmentStructure(selectedEquipment).equipmentName
lstPrice.Items.Add(equipmentStructure(selectedEquipment).equipmentPrice * days)

```

Data View Form

Description

The Data View form is responsible for reading in every piece of data from the data stores, and displaying them in a human-readable format.

The choices for viewing the data are as follows:

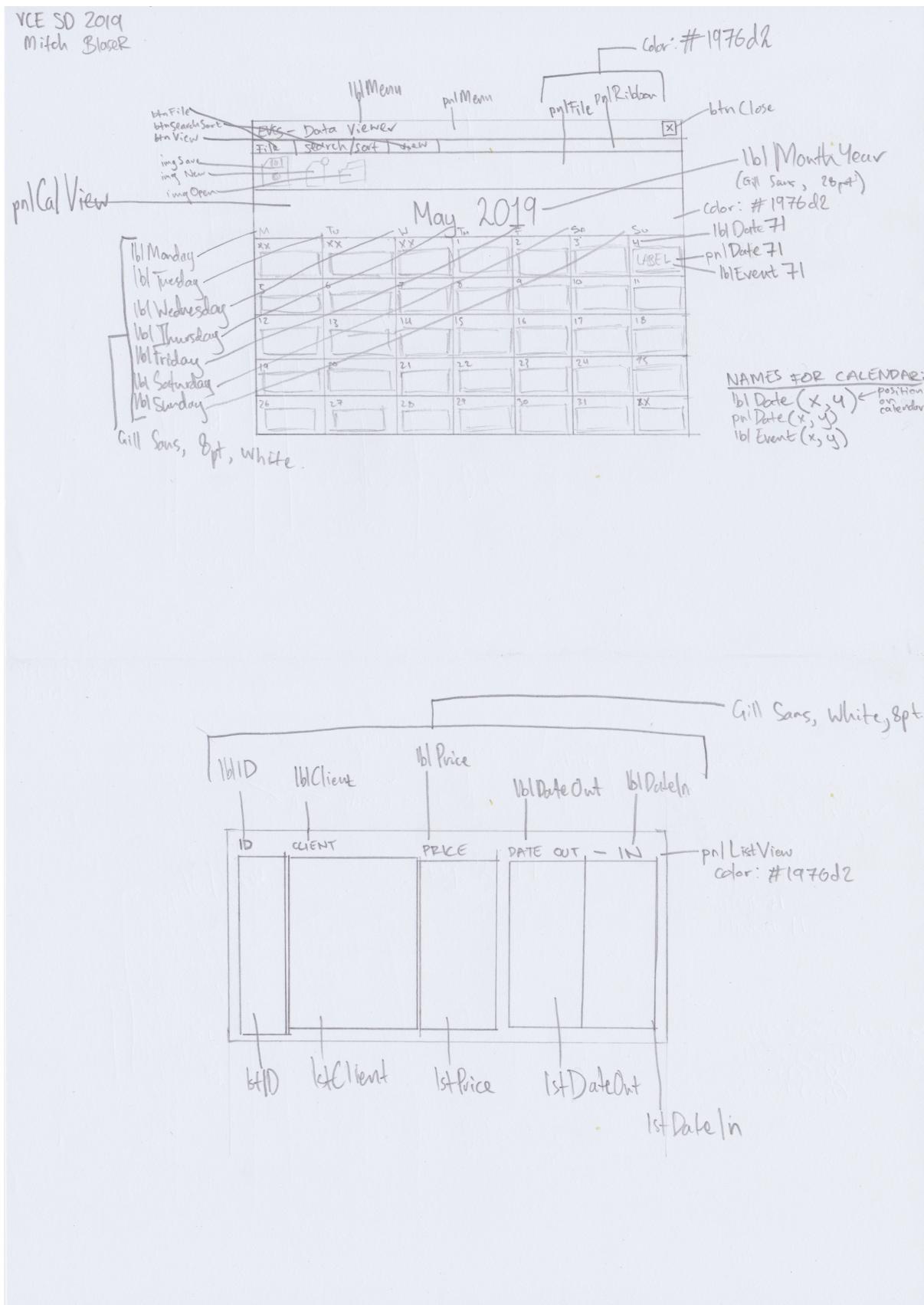
- Calendar
- List (Date)
- List (Alphabetical)
- List (Search)

Each view has its own button inside of the ribbon, which selects each view. All of the list-based views use one panel, and the calendar view uses one other, different panel. This is because the list based views all use the same elements to view everything.

The specific panel to be shown, along with the correct sorting algorithm, is selected based on which button in the ribbon is clicked. Unlike transitioning from the main menu to the data entry form, no private or public variables are used and everything is done from the relevant button's click event.

Data View Form

Interface Design



Data View Form

Object Descriptions

Object Name	Object Type	Object Description
frmEnterData	Form	The form that the user uses to enter data into the solution. Color: #373737
pnlMenu	Panel	The panel which holds the top menu bar items.
lblMenu	Label	The title of the form.
btnClose_Click	Button (Click Event)	Closes the form when the button is clicked.
pnlRibbon	Panel	Holds the ribbon items. Color: #1976d2
btnFile_Click	Button (Click Event)	Shows the File menu panel in the ribbon.
btnSearchSort_Click	Button (Click Event)	Shows the Search/Sort menu panel in the ribbon
btnView_Click	Button (Click Event)	Shows the View menu panel in the ribbon
pnlFile	Panel	Holds the buttons inside of the File menu
imgSave_Click	Image Box (Click Event)	Saves the record and writes out to the data file
imgNew_Click	Image Box (Click Event)	Creates a new record and clears all form elements.
imgOpen_Click	Image Box (Click Event)	Opens an existing record based on a record index number. Input number is inputted from a Message Box.
pnlSearchSort	Panel	Holds the buttons inside of the Search/Sort menu.
imgSearch_Click	Image Box (Click Event)	Prompts user for a search term via an Input Box, and launches a linear search algorithm. Once a result is found, the relevant record appears in the form's elements.
imgSortByDate	Image Box (Click Event)	If list data is displayed on screen, this button will launch an selection sort algorithm that rearranges items inside of all list boxes to be in order of newest to oldest date.
imgSortAlphabetical	Image Box (Click Event)	If list data is displayed on screen, this button will launch an selection sort algorithm that rearranges items inside of all list boxes to be in order of a to z.
pnlView	Panel	Holds the buttons inside of the View menu.
imgListView	Image Box (Click Event)	If the active form is the Data View form, using this button will rearrange all data points to be shown in a list form, instead of calendar.
imgCalendar	Image Box (Click Event)	If the active form is the Data View form, using this button will rearrange all data points to be shown in calendar form, instead of a list.

Object Name	Object Type	Object Description
pnlCalView	Panel	Holds all of the elements of the calendar view.
lblMonthYear	Label	Bigger label than the others, displays the month and year that the calendar is currently showing.
lblMonday	Label	Indicates which column of the calendar view is for Monday
lblTuesday	Label	Indicates which column of the calendar view is for Tuesday
lblWednesday	Label	Indicates which column of the calendar view is for Wednesday
lblThursday	Label	Indicates which column of the calendar view is for Thursday
lblFriday	Label	Indicates which column of the calendar view is for Friday
lblSaturday	Label	Indicates which column of the calendar view is for Saturday
lblSunday	Label	Indicates which column of the calendar view is for Sunday
lblDate11	Label	Name comes from the X and Y axis.
pnlDate11	Panel	Name comes from the X and Y axis.
lblEvent11	Label	Name comes from the X and Y axis.
lblDate77	Label	Name comes from the X and Y axis.
pnlDate77	Panel	Name comes from the X and Y axis.
lblEvent77	Label	Name comes from the X and Y axis.
pnlListView	Panel	Holds all of the elements of the list view.
lblID	Label	Indicates which column of the list view is for ID
lblClient	Label	Indicates which column of the list view is for Client
lblPrice	Label	Indicates which column of the list view is for Price
lblDateOut	Label	Indicates which column of the list view is for Date Out
lblDateIn	Label	Indicates which column of the list view is for Date In
lstID	List Box	A list of all hires. This list box shows ID.
lstClient	List Box	A list of all hires. This list box shows Client.
lstPrice	List Box	A list of all hires. This list box shows Price.
lstDateOut	List Box	A list of all hires. This list box shows Date Out.
lstDateIn	List Box	A list of all hires. This list box shows Date In.

Data View Form

Data Dictionary

Variable Name	Variable Type	Access Level	Description
i	Integer	Private	Iterator for loops. Private variable.
f	Integer	Private	Iterator for internal loops. Private variable.
Structure variables (clientStructure, equipmentStructure, hireStructure)	Structure	Private	Used to access data from the data store files.

Data View Form

Pseudo-Code

```
**I've had to condense this down quite a bit to meet the submission so it won't be as detailed as I'd like it to be.**  
Form_Load_Event  
pnlCalView.Show()  
  
Open files and put them into their structures.  
  
Find hire dates  
  
Mark start and end dates on calendar array  
  
Fill contents of calendar array into date labels (lblDateXX)  
  
imgListView_Click_Event  
lblEvent* = ""  
                                'Clear every label. It'd be too long to write everything here.  
  
Loop for length of structure array  
    lstID.Items.Add(i)  
    lstClient.Items.Add(clientStructure(hireStructure(i).hireClient).clientFirstName & " " &  
        clientStructure(hireStructure(i).hireClient).clientLastName)  
    Loop for f = 0 to hireStructure(i).equipmentLength  
        totalCost = totalCost + equipmentStructure(hireStructure(i).equipment(f)).equipmentPrice  
        f = f + 1  
    End Loop  
    lstPrice.Items.Add(f)  
    lstDateOut.Items.Add(hireStructure(i).hireOut)  
    lstDateIn.Items.Add(hireStructure(i).hireIn)  
End Loop  
  
imgCalendar_Click_Event  
lstID.Items.Clear()  
lstClient.Items.Clear()  
lstPrice.Items.Clear()  
lstDateOut.Items.Clear()  
lstDateIn.Items.Clear()  
  
pnlCalView.Show()  
  
Find hire dates  
  
Mark start and end dates on calendar array  
  
Fill contents of calendar array into date labels (lblDateXX)
```

Ribbon

Menu Structure

