

# Implementation

## Forms

Form Name	Notes/Features used
AddCust	Declares a data structure for customers and a variable with that data structure. Generates a customer ID based on the customer's name and their phone number, if ID already exists then it alters the end number. Saving performs data validations to ensure data integrity, then appends the data into the Customers text file. The text file is then sorted alphabetically using a bubblesort.
AddEquipment	Declares a data structure for equipment and a variable with that data structure. Generates an ID based on how many pieces of equipment are within the text file, if ID already exists then the ID is incremented. Saving performs data validations to ensure data integrity, then appends the data into the Equipment text file. The text file is then sorted alphabetically using a bubblesort.
AddStaff	Declares two data structure (one for staff and the other for passwords) then declares two variables variable with those data structures. Generates a staff ID based on the customer's name and their phone number, if ID already exists then it alters the end number. Generates a password ID based on the number of staff, if the ID is the same as another then it is incremented. Saving performs data validations to ensure data integrity, then appends the data appropriately into the Staff and Password text files. The staff text file is then sorted alphabetically using a bubblesort (the password file is not sorted).
AdminMenu	Opens forms accordingly. For equipment test checks and stock checks, the form goes through the equipment textfile and compares the dates (for test check) and amount (for stock check) to see if the specific markers have been passed and displays warning messages showing which equipment needs to be tested/reordered.
CalculatingPayroll	Declares data structures for Monthly Payroll and Yearly Payroll then a variable for each of the data structures. Once staff ID, month, and pay rate have been entered then the form calculates the gross pay, NI, income tax, pension, and net pay by loading in the number of hours that staff member has worked in that month. The form then checks if a monthly payroll slip with that payroll ID exists, if not then the Yearly

	Payroll Masterfile is updated with the monthly data and a Monthly Payroll text file is created.
CreatingInvoice	Declares a data structure for Invoices and creates a variable using that data structure. Once booking ID has been entered, the invoice ID is generated based on the number of existing invoices and it is incremented accordingly. The data is then appended to the Invoice Masterfile and an individual printable invoice is also created.
CustomerLogs	The form loads the customer IDs into the drop-down list. After the customer ID has been entered, the user can also select a date range. If no data is entered into either of the textboxes, then the value for that textbox is taken as the equivalent to infinity. The bookings text file is loaded and searched through to display the bookings for that customer and within the date range.
Login	Compares the staff ID and password entered to the password that is stored and associated with that staff ID. If the password is correct, the appropriate menu is opened depending on the staff type.
MyBookingsToday	Displays the bookings for the next 3 days of the logged-on staff member.
PayForInvoice	Once an invoice ID is entered, form displays the invoice's details and allows the user to confirm whether or not the customer has paid.
PlaceBooking	Declares a data structure and variable for bookings. Generates an ID depending on the date of booking and the number of other bookings on that day. Has a function and array for holding the tools for that job. Validates entered data and saves the data into the booking text file.
QuickStockUpdate	The user enters a number of equipment names into the drop-down lists and is able to change the stock amounts and last/next test dates. Once the save button is clicked the form validates the data using range checks then amends the changes into the Equipment text file.
SearchBooking	Once a booking ID has been selected, the details of that booking, the customer, and the staff member are loaded into the textboxes. The user can then edit the data and click "Save" which then validates the data and amends it within the bookings text file. The user can also cancel any changes that they made which loads back up the original data or, alternatively, delete the booking from the booking text file.
SearchCust	Once a customer ID has been selected, the details of the staff member are loaded into the textboxes. The user can then edit the data and click "Save" which then validates the data and amends it within the staff text file. The user can also cancel any changes that they made which loads back up the original data or, alternatively, delete the staff member from the staff text file.

SearchEquipment	Once an equipment ID has been selected, the details of the equipment are loaded into the textboxes. The user can then edit the data and click "Save" which then validates the data and amends it within the equipment text file. The user can also cancel any changes that they made which loads back up the original data or, alternatively, delete that equipment from the equipment text file.
SearchMonthlyPayroll	Once a staff ID, month, and year have been selected, and the "Open" button has been clicked, the form searches the text files for one which be named after the selected year, month and staff ID. If no text file is found then an error message is displayed, if it is found then the text file is opened.
SearchStaff	Once a staff ID has been selected, the details of the staff member and their password data are loaded into the textboxes. The user can then edit the data and click "Save" which then validates the data and amends it within the staff and password text file. The user can also cancel any changes that they made which loads back up the original data or, alternatively, delete that staff member and their password data from the staff and password text files.
SearchYearlyPayroll	Once a staff ID and year is selected, the form loads the data of that staff member from the Payroll Masterfile for that year into the textboxes.
StaffMenu	Opens forms accordingly. For equipment stock checks, the form goes through the equipment textfile and compares whether the amount has passed the reorder stock level and displays a message for which pieces of equipment has.
UninvoicedBookings	The form loads all the un-invoiced bookings into the rich textbox as soon as you open the form. The user can then select a date range using the calendars. The bookings text file is loaded and searched through to display the bookings with HasBeenInvoiced = False and within the date ranges.
UnpaidInvoices	The form loads all the unpaid invoices into the rich textbox as soon as you open the form. The user can then select a date range using the calendars. The invoices text file is loaded and searched through to display the invoices with Paid = False and within the date ranges.
WorkerLogs	The form loads the staff IDs into the drop-down list. After the staff ID has been entered, the user can also select a date range. If no data is entered into either of the textboxes, then the value for that textbox is taken as the equivalent to infinity. The bookings text file is loaded and searched through to display the bookings for that staff member and within the date range.

--	--

**I have used the following features:**

Feature	Description/Use	Page Num.
Linear Search	Looks at the lines and elements within a text file or array in a sequential manner. This is used within all the search functions.	
Bubble Sort	Goes through the array a certain number of times, comparing two times side by side to see which is out of order then switching them around if they are.	
Insertion Sort	Goes through the array and compares each value in turn, starting with the second value in the array. If the value is less than the value to its left, then the data items are swapped. The sort then goes onto the third bit and repeats the process until the end of the array is reached. This is used for sorting bookings in my program.	
Range Check	Compares whether a value is within a range (e.g. If amount in stock is < 1000).	
List Check	Compares whether a value is equal to at least one value from a given list.	
Presence Check	Checks whether something has, for example, been entered into a textbox.	
Format Check	Compares whether a value is in the correct format (i.e. date of birth is like **/**/****).	
Length Check	Compares whether a value is a certain number of characters.	
Type Check	Compares whether the entered data is of a certain type. Used often for checking if a value is an integer within the program.	
Error Catching	Confirmation used for all the saving/editing/deleting functions. Prevention of data duplication using searches to check if, for example, an ID already exists. Check whether notes and tools list contain commas as this would cause errors with loading saved data.	

## Aim and Objectives Coded

Original aim/objective	Successfully implemented (Y/N)
Log in screen	Y
Forgot password	Y
Adding new customers	Y
Adding new staff members	Y
Adding new equipment	Y
Editing customers	Y
Editing Staff	Y
Editing booking details	Y
Editing equipment	Y
Searching customers	Y
Searching staff	Y
Searching bookings/invoices	Y
Searching unpaid bookings/invoices	Y
Searching workers logs	Y
Deleting customers	Y
Deleting staff	Y
Cancelling bookings	Y
Placing bookings by searching for available slots	Y
Customers can pay for bookings/invoices	Y
Calculating payroll	Y
Showing when stock is nearing reorder levels	Y
Logging out	Y

## Implementation of Relational Database

The databases that have been created are entirely dependent on the primary key meaning that the program has been implemented with 3NF relational databases. I used these primary keys to include data from one table in other forms via the use of foreign keys.

## Data Structures

Array Name	Description of use
AllCustomers	Holds an individual customer's data. Uses the customer ID as the primary key.

AllStaff	Holds a staff member's data. Uses the staff ID as the primary key.
AllPasswords	Holds a staff member's login data (i.e. password, password clue, and staff type). Uses the password ID as the primary key and the staff ID as a foreign key.
AllEquipment	Holds a piece of equipment's data. Uses the equipment ID as the primary key.
AllBookings	Holds the data for a booking. Uses the booking ID as the primary key and staff and customer IDs as the foreign keys.
AllInvoices	Holds the data for an invoice of a booking. Uses the invoice ID as the primary key and the booking ID as a foreign key.
AllPayroll	Holds the data for the payroll data of a staff member. It uses the payroll ID as the primary key and the staff ID as a foreign key.

### User Defined routines

Routine Name	Function/Description
Load_Customers	Loads customer data from the "Customers" text file into the AllCustomer array
Bubble_Sort_Customers	Uses a bubble sort to sort the customers within the AllCustomers array in alphabetical order of surname
Save_All_Customers	Saves the data of all the customers to the customers text file (i.e. in case of an edit)
Load_Equipment	Loads equipment data from the "Equipment" text file into the AllEquipment array
Bubble_Sort_Equipment	Uses a bubble sort to sort the equipment within the AllEquipment array in order of equipment ID
Save_All_Equipment	Saves the data of all the equipment to the equipment text file(i.e. in case of an edit)
Load_Staff	Loads staff data from the "Staff" text file into the AllStaff array
Bubble_Sort_Staff	Uses a bubble sort to sort the staff within the AllStaff array in alphabetical order of surname
Save_All_Staff	Saves the data of all the staff to the staff text file (i.e. in case of an edit)
Load_Passwords	Loads password data from the "Passwords" text file into the AllPasswords array
Save_All_Passwords	Saves the data of all the passwords to the password text file (i.e. in case of an edit)

Load_Bookings	Loads booking data from the "Bookings" text file into the AllCustomer array
Insertion_Sort_Bookings	Uses an insertion sort to sort the bookings within the AllBookings array in order of booking ID
Save_All_Bookings	Saves the data of all the bookings to the booking text file (i.e. in case of an edit)
Load_Invoices	Loads invoice data from the "Invoice_Masterfile" text file into the AllInvoices array
Save_All_Invoices	Saves the data of all the invoices to the invoice Masterfile (i.e. for payments)
Load_Payroll	Loads payroll data from the "Payroll_Masterfile" text file into the AllPayroll array
Save_All_Payroll	Saves the data of all the payroll to the payroll Masterfile (i.e. For new monthly payrolls)

### Variable List

Module/Form Name	Variable name	Description
AddCust	NewCustomer	Variable that holds the customer data structure
	IDGenClick	Variable to check if the "Generate ID" button has been clicked
	TempNum	Variable for the number part of the customer ID from the phone number
	ChangedID	Variable to check whether the customer ID has been altered
	EmailFormat	Variable used to check the format of the email textbox
AddEquipment	NewEquipment	Variable that holds the equipment data structure
	IDGenClick	Variable to check if the "Generate ID" button has been clicked
	ChangedID	Variable to check whether the equipment ID has been altered
AddStaff	NewStaff	Variable to hold the staff data structure
	NewPassword	Variable to hold the password data structure
	IDGenClick	Used to check whether the "Generate ID" button has been clicked

	EmailFormat	Used to check the format of the email text box
	ChangedID	Variable to check whether the staff/password ID has been changed
	ArrayPasswordCommaCheck	Used to check if there is a comma within the password textbox
	ArrayClueCommaCheck	Used to check if there is a comma within the clue textbox
	TempNum	Variable for the number part of the staff ID
AdminMenu	TestingFound	Variable to hold whether there is a piece of equipment that requires testing
	NeedsReorder	Variable to hold the items that need testing
	AmountMissing	Variable to hold the amount of a type of item that is missing
CalculatingPayroll	NewMonthPayroll	Variable to hold the monthly payroll data structure
	NewYearPayroll	Variable to hold the yearly payroll data structure
	NotBookedPay	Variable to hold the amount of pay for the hours that a worker was not booked
CreatingInvoice	NewInvoice	Variable with the invoice data structure
	Position	Variable to hold position where a booking was found
	JobNotesLength	Variable to hold job notes length
	Index	Variable to hold which part of job notes the WriteLine() function is taking
CustomerLogs	AllocatedHoursString	Variable to allow "Allocated hours" to be displayed as a string
	Found	Variable to check the customers data has been found
	DateFormatCheck	Used to check if the range date is in the correct format
Login	FoundID	Variable to be able to find the staff ID within the password file
	AttemptCount	Variable so that the clue appears if the user fails to logon 3 times
MyBookingsToday	AllocatedHoursString	Variable to allow "Allocated hours" to be displayed as a string
	Found	Variable to check data has been found
Password_Module	TempCust (1 to 11)	Temporary variables for the customer bubble sort

	NumCustomers	Holds the number of customers in the customer array
	file	Stream reader for the computer system
	line	"Readline" variable for files
	parts	Splits the line at the declared index (i.e. ",")
	SaveAllCustomers	Stream writer of customers for the computer system
	TempEquip (1 to 8)	Temporary variables for the equipment bubble sort
	NumEquipment	Holds the number of equipment in the equipment array
	SaveAllEquipment	Stream writer of equipment for the computer system
	TempStaff (1 to 12)	Temporary variables for the staff bubble sort
	NumStaff	Holds the number of staff in the staff array
	SaveAllStaff	Stream writer of staff for the computer system
	LoggedOnStaffType	Holds the staff type of the logged on user
	LoggedOnStaffID	Holds the staff ID of the logged on user
	TempPass (1 to 5)	Temporary variables for the password bubble sort
	NumPasswords	Holds the number of passwords in the password array
	SaveAllPasswords	Stream writer of passwords for the computer system
	NumBookings	Holds the number of bookings in the bookings array
	Element (1 to 10)	Temporary variables for the bookings insertion sort
	Position	Holds the position for the bookings insertion sort
	SaveAllBookings	Stream writer of bookings for the computer system
	NumInvoices	Holds the number of invoices in the invoices array
	SaveAllInvoices	Stream writer of invoices for the computer system
	NumPayroll	Holds the number of payrolls in the payroll array
	SaveAllPayroll	Stream writer of payroll for the computer system
PayForInvoices	Price	Holds the price of a booking

	VAT	Holds VAT of a booking
	PayCheck	Used to check if the customer has paid
PlaceBooking	NewBooking	Variable with the Booking data structure
	SameBookingDateCount	Variable to count how many bookings there are with the same date
	StaffFound	Variable to check if a staff member within the job type has been found
	ToolPosition	Variable to hold which position in the tool array is being used
	StringDate	Variable to hold the booking date as a string
	PriceCommaCheck	Variable used to check if there is a comma in the price of a booking
	NotesCommaCheck	Variable used to check if there is a comma in the notes of a booking
QuickStockUpdate	Position (1 to 3)	Variables to hold the position of items from the array
	ChangedItem (1 to 3)	Variables to hold whether the drop down lists have been changed
SearchBooking	BookingIDChanged	Variable to say if the booking ID has been changed
	Position	Holds the position of an item within the booking array
	StaffFound	Variable to check if a staff member within the job type has been found
	SameBookingDateCount	Variable to count how many bookings there are with the same date
	ToolPosition	Holds the position of the tool within the tool array
	TempBookingDate	Variable to temporarily hold the booking date so can use it after the booking date ddl has been cleared
	TempCustomerID	Variable to temporarily hold the customerID as to not call "ddlCustomerID_SelectedIndexChanged()" until after loop
	BookingTimeCheck	Variable to hold the max time for a booking to finish
	FinishTime	Variables that's calculated to hold the finish time of the booking

	PriceCommaCheck	Variable used to check if there is a comma in the price of a booking
	NotesCommaCheck	Variable used to check if there is a comma in the notes of a booking
SearchCust	Position	Used to hold the position of the customer within the customer array
	NumFound	Number of customers found with a certain surname
	EmailFormat	Used to check the format of the email
SearchEquipment	Position	Used to hold the position of the equipment within the equipment array
SearchMonthlyPayroll	SearchPayID	Holds the generated monthly payroll ID associated with the entered month, year, and staff ID
	file	Stream reader for the computer system
SearchStaff	PositionPassword	Position of the password data within the password array
	PositionStaff	Position of the staff data within the staff array
	NumFound	Number of staff found with a certain surname
	PasswordFound	Holds whether the password data has been found
	EmailFormat	Checks the format of the entered email
	ArrayPasswordCommaCheck	Variable used to check if there is a comma in the password
	ArrayClueCommaCheck	Variable used to check if there is a comma in the clue
SearchYearlyPayroll	file	Stream reader for the computer system
StaffMenu	NeedsReorder	Variable to hold the items that need testing
	AmountMissing	Variable to hold the amount of a type of item that is missing
UninvoicedBookings	NumDays	Variable to hold "Days until booking"
	BookingFound	Used to check if the booking has been found
UnpaidInvoices	NumDays	Variable to hold "Days since booking"
	InvoiceFound	Used to check if the invoice data has been found
WorkerLogs	AllocatedHoursString	Variable to allow "Allocated hours" to be displayed as a string
	Found	Used to check if the worker logs data has been found
	DateFormatCheck	Used to check the format of the date

## **Interface Suitability**

Visual basic allows the use of interactable items, such as button and menus, for the user to use to manipulate the program. This allows for a flexibility in ability so that even the most amateur users are able to quickly understand how it works and prevent the need for excessive IT training.

**Form Name:**

## Adding Customer

Customer ID	<input type="text"/>	<input type="button" value="Generate ID"/>
Title	<input type="text"/>	
First name	<input type="text"/>	
Sumame	<input type="text"/>	
1st Address Line	<input type="text"/>	
2nd Address Line	<input type="text"/>	
Postcode	<input type="text"/>	
Town/City	<input type="text"/>	
Phone Number	<input type="text"/>	
Email	<input type="text"/>	
DoB	<input type="text" value="01/01/1900"/>	<input type="button" value="📅"/>

**Code Listing:**

```

1 Imports System.IO
2 Public Class AddCust
3
4     'Declaring a data structure for customers
5     Public Structure Customer
6         Public CustomerID As String
7         Public Title As String
8         Public FName As String
9         Public Sname As String
10        Public FAddress As String
11        Public SAddress As String
12        Public Postcode As String
13        Public Town As String
14        Public Phone As String
15        Public Email As String
16        Public DoB As String
17    End Structure
18
19    'Declaring a variable with the customer data structure
20    Public NewCustomer As Customer
21
22    Public IDGenClick As Boolean = False 'Variable to check if the "Generate ID" button has been clicked
23
24    Private Sub butGenID_Click(sender As Object, e As EventArgs) Handles butGenID.Click
25
26        'Checking that all the textboxes that are used in the ID have been entered
27        If txtFname.Text = "" Or txtSname.Text = "" Or txtNum.Text = "" Then
28            MsgBox("Please fill out the rest of the form before generating ID")
29            Exit Sub
30        End If
31
32        'Generating Customer ID
33        NewCustomer.CustomerID = txtSname.Text.Substring(0, 2).ToUpper & txtFname.Text.Substring(1, 2).ToUpper & txtNum.Text.Substring(4, 2)
34        'Showing the generated customer ID in the textbox
35        txtCustID.Text = NewCustomer.CustomerID
36
37        'Loading the customer textfile
38        Load_Customers()
39
40        'Looping through the AllCustomers array to check if the new customer ID already exists
41        Dim TempNum = Int(txtNum.Text.Substring(4, 2)) 'Declaring a variable for the number part of the ID from the phone number
42        Dim ChangedID As Boolean = True 'Declaring a variable to check whether the ID has been changed
43
44        While ChangedID = True
45            ChangedID = False
46            For x = 0 To NumCustomers - 1
47                'Check if that customer ID already exists
48                If AllCustomers(x).CustomerID = NewCustomer.CustomerID Then
49                    TempNum = TempNum + 10
50
51                'Check if TempNum is >99 as the ID would then be 7 characters long rather than 6
52                If TempNum > 99 Then
53                    TempNum = TempNum - 89 'Alters TempNum so its >10 and <100 and has a different last digit
54                End If
55
56                'Changes generated ID to include the "TempNum" variable on the end instead
57                NewCustomer.CustomerID = txtSname.Text.Substring(0, 2).ToUpper & txtFname.Text.Substring(1, 2).ToUpper & TempNum
58                txtCustID.Text = NewCustomer.CustomerID
59                'Change variable so program loops through the array again And compares all IDs with the New generated ID
60                ChangedID = True
61            End If
62        Next
63    End While
64
65    'Changing boolean variable to say that the generate ID button has been clicked
66    IDGenClick = True
67
68 End Sub
69
70 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
71
72    'Checking if generate ID button has been clicked
73    If IDGenClick = False Then
74        MsgBox("Please generate an ID before saving")
75        Exit Sub
76    End If

```

```

77
78 'Entering textbox value into Title
79 NewCustomer.Title = txtTitle.Text
80 'List check on title
81 If NewCustomer.Title <> "Mr" And NewCustomer.Title <> "Mrs" And NewCustomer.Title <> "Miss" And NewCustomer.Title <> "Ms" And NewCustomer.Title <> "Mx" And
82     MsgBox("Incorrect title entered, correct titles; Mr, Mrs, Miss, Ms, Mx, Sir, Dr, Cllr, Lord, Lady. Please try again")
83     Exit Sub
84 End If
85
86 'Correct capitalisation on first name and surname
87 NewCustomer.Fname = txtFname.Text.Substring(0, 1).ToUpper & txtFname.Text.Substring(1, txtFname.Text.Length - 1).ToLower
88 NewCustomer.Sname = txtSname.Text.Substring(0, 1).ToUpper & txtSname.Text.Substring(1, txtSname.Text.Length - 1).ToLower
89
90 'Entering textbox values into FAddress, SAddress, Postcode and Town
91 NewCustomer.FAddress = txtFAddress.Text
92 NewCustomer.SAddress = txtSAddress.Text
93 NewCustomer.Postcode = txtPost.Text
94 NewCustomer.Town = txtTown.Text
95
96 'Entering textbox value into Phone
97 NewCustomer.Phone = txtNum.Text
98 'Length check on phone number
99 If Len(NewCustomer.Phone) <> 11 Then
100     MsgBox("Incorrect phone number entered, must be 11 characters long, please try again")
101     Exit Sub
102 End If
103
104 'Entering textbox value into email
105 NewCustomer.Email = txtEmail.Text
106 'Format check on email
107 Dim EmailFormat As Boolean = NewCustomer.Email Like "*@*.*"
108 If EmailFormat = False Then
109     MsgBox("Incorrect email format detected, must be like ___@__._, please try again")
110     Exit Sub
111 End If
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
```

```

131 ewCustomer.FAddress & ", " & NewCustomer.SAddress & ", " & NewCustomer.Postcode & ", " & NewCustomer.Town & ", " & NewCustomer.Phone & ", " & NewCustomer.Email & ", " & NewCustomer.DoB)
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150 Private Sub butClear_Click() Handles butClear.Click
151
152     'Clearing all the textboxes
153     txtCustID.Text = ""
154     txtTitle.Text = ""
155     txtFname.Text = ""
156     txtSname.Text = ""
157     txtFAddress.Text = ""
158     txtSAddress.Text = ""
159     txtPost.Text = ""
160     txtTown.Text = ""
161     txtNum.Text = ""
162     txtEmail.Text = ""
163     calendarDoB.Value = "01/01/1900"
164
165     'Resets IDGenClick
166     IDGenClick = False
167
168 End Sub
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

```

---

```

1 reference
Private Sub butClear_Click() Handles butClear.Click
    'Clearing all the textboxes
    txtCustID.Text = ""
    txtTitle.Text = ""
    txtFname.Text = ""
    txtSname.Text = ""
    txtFAddress.Text = ""
    txtSAddress.Text = ""
    txtPost.Text = ""
    txtTown.Text = ""
    txtNum.Text = ""
    txtEmail.Text = ""
    calendarDoB.Value = "01/01/1900"
    'Resets IDGenClick
    IDGenClick = False
End Sub

```

---

```

0 references
Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
    'Closing the form
    Me.Close()
End Sub

```

---

```

0 references
Private Sub txtFname_TextChanged(sender As Object, e As EventArgs) Handles txtFname.TextChanged
    'Resets IDGenClick as this variable is used within the ID generation
    IDGenClick = False
    txtCustID.Text = ""
End Sub

```

---

```

0 references
Private Sub txtSname_TextChanged(sender As Object, e As EventArgs) Handles txtSname.TextChanged
    'Resets IDGenClick as this variable is used within the ID generation
    IDGenClick = False
    txtCustID.Text = ""
End Sub

```

---

```

0 references
Private Sub txtNum_TextChanged(sender As Object, e As EventArgs) Handles txtNum.TextChanged
    'Resets IDGenClick as this variable is used within the ID generation
    IDGenClick = False
    txtCustID.Text = ""
End Sub
End Class

```

Form Name:

## Adding Equipment

Equipment ID	<input type="text"/>	<input type="button" value="Generate ID"/>
Item Name	<input type="text"/>	
Amount in stock	<input type="text"/>	
Reorder stock level	<input type="text"/>	
Reorder quantity	<input type="text"/>	
Last test date	<input type="text" value="01/01/2020"/>	<input type="button" value="Calendar"/>
Next test date	<input type="text" value="01/01/2020"/>	<input type="button" value="Calendar"/>
Tested by	<input type="text"/>	

Code Listing:

```

1 Imports System.IO
  1 reference
2 Public Class AddEquipment
3
4     'Declaring a data structure for equipment
  1 reference
5     Public Structure Equipment
6         Public EquipmentID As Integer
7         Public ItemName As String
8         Public AmountInStock As Integer
9         Public ReorderLevel As Integer
10        Public ReorderAmount As Integer
11        Public LTestDate As String
12        Public NTestDate As String
13        Public TestedBy As String
14    End Structure
15
16    'Declaring a variable with the Equipment data structure
17    Public NewEquipment As Equipment
18
19    'Declaring a variable to check if the "Generate ID" button has been clicked
20    Public IDGenClick As Boolean = False
21
  0 references
22    Private Sub buGenID_Click(sender As Object, e As EventArgs) Handles buGenID.Click
23
24        'Generating Equipment ID
25        NewEquipment.EquipmentID = 1
26
27        'Showing the generated customer ID in the textbox
28        txtEquipID.Text = NewEquipment.EquipmentID
29
30        'Loading the customer textfile
31        Load_Equipment()
32
33        'Looping through the AllCustomers array to check if the new customer ID already exists
34        Dim ChangedID As Boolean = True 'Declaring a variable to check whether the ID has been changed
35
36        While ChangedID = True
37            ChangedID = False
38            For x = 0 To NumEquipment - 1
39                If AllEquipment(x).EquipmentID = NewEquipment.EquipmentID Then 'Loops through AllEquipment
40                    'Increments the Equipment ID

```

```

41         NewEquipment.EquipmentID = NewEquipment.EquipmentID + 1
42         txtEquipID.Text = NewEquipment.EquipmentID 'Display new ID
43         'Change variable so program loops through the array again And compares all IDs with the New generated ID
44         ChangedID = True
45     End If
46 Next
47 End While
48
49 'Changing boolean variable to say that the generate ID button has been clicked
50 IDGenClick = True
51
52 End Sub
53
54 0 references
55 Private Sub calendarLTestDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarLTestDate.ValueChanged
56
57     'Sets Next Test Date equal to the Last Test Date value if N < L
58     If calendarNTestDate.Value < calendarLTestDate.Value Then
59         calendarNTestDate.Value = calendarLTestDate.Value
60     End If
61 End Sub
62
63 0 references
64 Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
65
66     'Checking if generate ID button has been clicked
67     If IDGenClick = False Then
68         MsgBox("Please generate an ID before saving")
69         Exit Sub
70     End If
71
72     'Assign item name and change the first letter to a capital
73     NewEquipment.ItemName = txtName.Text.Substring(0, 1).ToUpper & txtName.Text.Substring(1, txtName.Text.Length - 1)
74
75     'Check if item exists in the Equipment textfile already
76     For x = 0 To NumEquipment - 1
77         If txtName.Text = AllEquipment(x).ItemName Then
78             MsgBox("This item already exists, please try again")
79             Exit Sub
80         End If

```

```

80     Next
81
82     'Check that amount in stock is an integer
83     Try
84         NewEquipment.AmountInStock = txtAmount.Text
85     Catch
86         MsgBox("Amount in stock must be an integer")
87     Exit Sub
88 End Try
89
90     'Range check on Amount in Stock
91     If NewEquipment.AmountInStock > 1000 Or NewEquipment.AmountInStock < 0 Then
92         MsgBox("The amount for this item is out of range, must be <1000 and >0, please try again")
93     Exit Sub
94 End If
95
96     'Integer check on reorder level and quantity
97     Try
98         NewEquipment.ReorderLevel = txtReorderLevel.Text
99         NewEquipment.ReorderAmount = txtReorderQuan.Text
100    Catch
101        MsgBox("Reorder level and quantity must be integer values")
102    Exit Sub
103 End Try
104
105    'Check if Last Test Date has been selected to be before or on today
106    If calendarLTestDate.Value = "01/01/2020" Or calendarLTestDate.Value > Now Then
107        MsgBox("Please select an appropriate last test date (i.e. not after today)")
108    Exit Sub
109 End If
110    NewEquipment.LTestDate = calendarLTestDate.Value 'By using ".value" the data is taken as a Date value
111
112    'Check if Next Test Date has been selected to be on or after today
113    If calendarNTestDate.Value < Now Or calendarNTestDate.Value = calendarLTestDate.Value Then
114        MsgBox("Please select an appropriate next test date (i.e. not today or before)")
115    Exit Sub
116 End If
117    NewEquipment.NTestDate = calendarNTestDate.Value 'By using ".value" the data is taken as a Date value
118
119    NewEquipment.TestedBy = txtTestedBy.Text
120
121
122    'Confirmation that the details are correct
123    If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
124        Exit Sub 'Exit if user clicks cancel
125    End If
126
127    'Telling the user that the generated ID is what will be used
128    MsgBox("Please note that if you have manually changed the ID then it will not be saved and the previously generated ID will be saved")
129    MsgBox("The ID that was generated is: " & NewEquipment.EquipmentID)
130
131    'Saving data to a textfile called "Equipment"
132    Dim SaveNewEquipment As System.IO.StreamWriter
133    SaveNewEquipment = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Equipment.txt"), True)
134    SaveNewEquipment.WriteLine(NewEquipment.EquipmentID & "," & NewEquipment.ItemName & "," & NewEquipment.AmountInStock & "," & NewEquipment.ReorderLevel & "," & N
135    SaveNewEquipment.Close()
136
137    Load_Equipment() 'Loads the newly saved equipment into the equipment array
138    Bubble_Sort_Equipment() 'Sorts the equipment array alphabetically
139    Save_All_Equipment() 'Saves the sorted equipment array to the equipment textfile
140
141    'Msgbox with a summary of the data saved
142    MsgBox("Data has been saved")
143
144    'Resets IDGenClick
145    IDGenClick = False
146
147    'Clear text boxes
148    butClear_Click()
149 End Sub
150
151
152
153 t.AmountInStock & "," & NewEquipment.ReorderLevel & "," & NewEquipment.ReorderAmount & "," & NewEquipment.LTestDate & "," & NewEquipment.NTestDate & "," & NewEquipment.TestedBy)
154

```

```
151 1 reference
152 Private Sub butClear_Click() Handles butClear.Click
153     'Clearing all the text boxes
154     txtEquipID.Text = ""
155     txtName.Text = ""
156     txtAmount.Text = ""
157     txtReorderLevel.Text = ""
158     txtReorderQuan.Text = ""
159     calendarLTestDate.Text = "01/01/2020"
160     calendarNTestDate.Text = "01/01/2020"
161     txtTestedBy.Text = ""
162
163     'Resets IDGenClick
164     IDGenClick = False
165
166 End Sub
167
168 0 references
169 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
170     Me.Close()
171 End Sub
172 End Class
173
```

**Form Name:**

## Adding Staff

Staff ID	<input type="text"/>	<input type="button" value="Generate ID"/>
Staff Type	<input type="text"/>	
Speciality	<input type="text" value="v"/>	
Title	<input type="text"/>	
First Name	<input type="text"/>	
Surname	<input type="text"/>	
1st Address Line	<input type="text"/>	
2nd Address Line	<input type="text"/>	
Postcode	<input type="text"/>	
Town/City	<input type="text"/>	
Phone Number	<input type="text"/>	
Email	<input type="text"/>	
DoB	<input type="text" value="01/01/1900"/> <input type="button" value="calendar"/>	
Password	<input type="text"/>	
Clue	<input type="text"/>	

**Code Listing:**

```

1 reference
1 Public Class AddStaff
2
3     'Declaring a structure for staff members
4     1 reference
5     Public Structure Staff
6         Public StaffID As String
7         Public Title As String
8         Public FName As String
9         Public Sname As String
10        Public FAddress As String
11        Public SAddress As String
12        Public Postcode As String
13        Public Town As String
14        Public Phone As String
15        Public Email As String
16        Public DoB As String
17        Public Speciality As String
18    End Structure
19
20    'Declaring a structure for password data
21    1 reference
22    Public Structure Password
23        Public PasswordID As Integer
24        Public StaffID As String
25        Public StaffType As String
26        Public Password As String
27        Public Clue As String
28    End Structure
29
30    'Declaring a variable with the Staff data structure and one with the Password data structure
31    Public NewStaff As Staff
32    Public NewPassword As Password
33
34    'Declaring a variable to check if the "Generate ID" button has been clicked
35    Public IDGenClick As Boolean = False
36
37    - references
38    Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
39
40        '*****STAFF FILE
41
42        'Checking if generate ID button has been clicked

```

```

39 'Checking if generate ID button has been clicked
40 If IDGenClick = False Then
41     MsgBox("Please generate an ID before saving")
42     Exit Sub
43 End If
44
45 NewStaff.Title = txtTitle.Text
46 'List check on title
47 If NewStaff.Title <> "Mr" And NewStaff.Title <> "Mrs" And NewStaff.Title <> "Miss" And NewStaff.Title <> "Ms" And NewStaff.Title <> "Mx" And NewStaff
48     MsgBox("Incorrect title entered, correct titles; Mr, Mrs, Miss, Ms, Mx, Sir, Dr, Cllr, Lord, Lady. Please try again")
49     Exit Sub
50 End If
51
52 'Correct capitalisation on first name and surname
53 NewStaff.Fname = txtFname.Text.Substring(0, 1).ToUpper & txtFname.Text.Substring(1, txtFname.Text.Length - 1).ToLower
54 NewStaff.Sname = txtSname.Text.Substring(0, 1).ToUpper & txtSname.Text.Substring(1, txtSname.Text.Length - 1).ToLower
55
56 'Entering textbox values into FAddress, SAddress, Postcode and Town
57 NewStaff.FAddress = txtFAddress.Text
58 NewStaff.SAddress = txtSAddress.Text
59 NewStaff.Postcode = txtPost.Text
60 NewStaff.Town = txtTown.Text
61
62 NewStaff.Phone = txtNum.Text
63 'Length check on phone number
64 If Len(NewStaff.Phone) <> 11 Then
65     MsgBox("Incorrect phone number entered, must be 11 characters long, please try again")
66     Exit Sub
67 End If
68
69 NewStaff.Email = txtEmail.Text
70 'Format check on email
71 Dim EmailFormat As Boolean = NewStaff.Email Like "*@*.*"
72 If EmailFormat = False Then
73     MsgBox("Incorrect email format detected, must be like ___@___.__, please try again")
74     Exit Sub
75 End If
76
77 'Check if DoB has been selected
78 If calendarDoB.Value = "01/01/1900" Then
79     MsgBox("Please select a date of birth")
80
81     Exit Sub
82 End If
83 NewStaff.DoB = calendarDoB.Text
84
85 NewStaff.Speciality = ddlSpecial.Text
86
87 '*****PASSWORD FILE
88 NewPassword.PasswordID = 1 'Generating Password ID
89
90 'Loading the password textfile
91 Load_Password()
92
93 'Looping through the AllPasswords array to check if the new password ID already exists
94 Dim ChangedID As Boolean = True 'Declaring a variable to check whether the ID has been changed
95
96 While ChangedID = True
97     ChangedID = False
98     For x = 0 To NumPasswords - 1
99         If AllPasswords(x).PasswordID = NewPassword.PasswordID Then 'Loops through AllEquipment
100             'Increments the Password ID
101             NewPassword.PasswordID = NewPassword.PasswordID + 1
102             'Change variable so program loops through the array again And compares all IDs with the New generated ID
103             ChangedID = True
104         End If
105     Next
106 End While
107
108 NewPassword.StaffID = NewStaff.StaffID 'Foreign Key
109
110 'List check on staff type
111 If txtStaffType.Text <> "Worker" And txtStaffType.Text <> "Admin" Then
112     MsgBox("Incorrect staff type, must be 'Worker' or 'Admin', please try again")
113     Exit Sub
114 End If
115 NewPassword.StaffType = txtStaffType.Text
116
117 NewPassword.Password = txtPassword.Text
118 NewPassword.Clue = txtClue.Text
119
120 End NewStaff.Title <> "Mx" And NewStaff.Title <> "Sir" And NewStaff.Title <> "Dr" And NewStaff.Title <> "Cllr" And NewStaff.Title <> "Lady" And NewStaff.Title <> "Lord" Then
121     MsgBox("Please try again")
122     Exit Sub
123 End If
124
125 Exit Sub
126 End Sub

```

```

118
119 'Checking that the password and clue do not include ","
120 Dim ArrayPasswordCommaCheck() As Char = txtPassword.Text.ToCharArray
121 Dim ArrayClueCommaCheck() As Char = txtClue.Text.ToCharArray
122 For x = 0 To Len(txtPassword.Text) - 1
123     If ArrayPasswordCommaCheck(x) = "," Then
124         MsgBox("Your password must not include a comma, please try again")
125         Exit Sub
126     End If
127 Next
128 For x = 0 To Len(txtClue.Text) - 1
129     If ArrayClueCommaCheck(x) = "," Then
130         MsgBox("Your clue must not include a comma, please try again")
131         Exit Sub
132     End If
133 Next
134
135 MsgBox("Please remember your password: " & NewPassword.Password)
136
137 *****SAVING DATA
138
139 'Confirmation that the details are correct
140 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
141     Exit Sub 'Exit if user clicks cancel
142 End If
143
144 'Saving relative data to a textfile called "Staff"
145 Dim SaveNewStaff As System.IO.StreamWriter
146 SaveNewStaff = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Staff.txt"), True)
147 SaveNewStaff.WriteLine(NewStaff.StaffID & "," & NewStaff.Title & "," & NewStaff.Fname & "," & NewStaff.Sname & "," & NewStaff.FAddress & "," & NewStaff.SAddress & ","
148 SaveNewStaff.Close()
149
150 'Saving relative data to a textfile called "Passwords"
151 Dim SaveNewPassword As System.IO.StreamWriter
152 SaveNewPassword = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Passwords.txt"), True)
153 SaveNewPassword.WriteLine(NewPassword.PasswordID & "," & NewPassword.StaffID & "," & NewPassword.StaffType & "," & NewPassword.Password & "," & NewPassword.Clue)
154 SaveNewPassword.Close()
155
156 'Telling the user that the generated ID is what will be used
157 MsgBox("Please note that if you have manually changed your ID then it will not be saved and the previously generated ID will be saved" + vbCrLf + "The ID that was gen
158
444
147 Address & "," & NewStaff.SAddress & "," & NewStaff.Postcode & "," & NewStaff.Town & "," & NewStaff.Phone & "," & NewStaff.Email & "," & NewStaff.DoB & "," & NewStaff.Speciality)
148
149
150
151
152
153 Password & "," & NewPassword.Clue)
154
155
156
157 saved" + vbCrLf + "The ID that was generated is: " & NewStaff.StaffID)
158

```

```
158
159 Load_Staff() 'Load the new staff into the array
160 Bubble_Sort_Staff() 'Sorts the staff within the staff array alphabetically, according to ID
161 Save_All_Staff() 'Saves the newly sorted staff array to the staff file
162
163 Load_Password() 'Load the new password into the array
164
165 'Msgbox with a summary of the data saved
166 MsgBox("Data has been saved")
167
168 'Resets IDGenClick
169 IDGenClick = False
170
171 'Clear.click
172 butClear_Click()
173
174 End Sub
175
```

0 references

```
176 Private Sub AddStaff_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
177
178 'Loading items into "Speciality" drop down list
179 ddlSpecial.Items.Add("Welding")
180 ddlSpecial.Items.Add("Woodwork")
181 ddlSpecial.Items.Add("Plumbing")
182 ddlSpecial.Items.Add("Carpentry")
183 ddlSpecial.Items.Add("Electrician")
184 ddlSpecial.Items.Add("Gardening")
185 ddlSpecial.Items.Add("Roofing")
186 ddlSpecial.Items.Add("Other")
187
188 End Sub
189
```

```
176 0 references
177 Private Sub AddStaff_Load(sender As Object, e As EventArgs) Handles MyBase.Load
178     'Loading items into "Speciality" drop down list
179     ddlSpecial.Items.Add("Welding")
180     ddlSpecial.Items.Add("Woodwork")
181     ddlSpecial.Items.Add("Plumbing")
182     ddlSpecial.Items.Add("Carpentry")
183     ddlSpecial.Items.Add("Electrician")
184     ddlSpecial.Items.Add("Gardening")
185     ddlSpecial.Items.Add("Roofing")
186     ddlSpecial.Items.Add("Other")
187
188 End Sub
189
```

```
190 1 reference
191 Private Sub butClear_Click() Handles butClear.Click
192     'Clearing all the textboxes
193     txtStaffID.Text = ""
194     txtStaffType.Text = ""
195     txtTitle.Text = ""
196     txtFname.Text = ""
197     txtSname.Text = ""
198     txtFAddress.Text = ""
199     txtSAddress.Text = ""
200     txtPost.Text = ""
201     txtTown.Text = ""
202     txtNum.Text = ""
203     txtEmail.Text = ""
204     calendarDoB.Value = "01/01/1900"
205     txtPassword.Text = ""
206     txtClue.Text = ""
207     ddlSpecial.Text = ""
208
209     'Resets IDGenClick
210     IDGenClick = False
211
212 End Sub
213
```

```

214 0 references
215 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
216     'Closing the form
217     Me.Close()
218
219 End Sub
220
221 0 references
222 Private Sub butGenID_Click(sender As Object, e As EventArgs) Handles butGenID.Click
223
224     'Checking that all the textboxes that are used in the ID have been entered
225     If txtFname.Text = "" Or txtSname.Text = "" Or txtNum.Text = "" Then
226         MsgBox("Please fill out the rest of the form before generating ID")
227         Exit Sub
228     End If
229
230     'Generating Staff ID
231     NewStaff.StaffID = txtSname.Text.Substring(0, 2).ToUpper & txtFname.Text.Substring(1, 2).ToUpper & txtNum.Text.Substring(8, 2)
232     'Showing the generated staff ID in the textbox
233     txtStaffID.Text = NewStaff.StaffID
234
235     'Loading the staff textfile
236     Load_Staff()
237
238     'Looping through the AllStaff array to check if the new staff ID already exists
239     Dim TempNum = Int(txtNum.Text.Substring(8, 2)) 'Declaring a variable for the number part of the ID
240     Dim ChangedID As Boolean = True 'Declaring a variable to check whether the ID has been changed
241
242     While ChangedID = True
243         ChangedID = False
244         For x = 0 To NumStaff - 1
245             If AllStaff(x).StaffID = NewStaff.StaffID Then
246                 TempNum = TempNum + 10
247                 'Changes generated ID to include the "TempNum" variable on the end instead
248                 NewStaff.StaffID = txtSname.Text.Substring(0, 2).ToUpper & txtFname.Text.Substring(1, 2).ToUpper & TempNum
249                 txtStaffID.Text = NewStaff.StaffID
250                 'Change variable so program loops through the array again And compares all IDs with the New generated ID
251                 ChangedID = True
252             End If
253         Next
254     End While

```

```
254
255     'Changing boolean variable to say that the generate ID button has been clicked
256     IDGenClick = True
257
258 End Sub
259
260 0 references
261 Private Sub txtFname_TextChanged(sender As Object, e As EventArgs) Handles txtFname.TextChanged
262
263     'Resets IDGenClick as this variable is used within the ID generation
264     IDGenClick = False
265     txtStaffID.Text = ""
266
267 End Sub
268
269 0 references
270 Private Sub txtSname_TextChanged(sender As Object, e As EventArgs) Handles txtSname.TextChanged
271
272     'Resets IDGenClick as this variable is used within the ID generation
273     IDGenClick = False
274     txtStaffID.Text = ""
275
276 End Sub
277
278 0 references
279 Private Sub txtNum_TextChanged(sender As Object, e As EventArgs) Handles txtNum.TextChanged
280
281     'Resets IDGenClick as this variable is used within the ID generation
282     IDGenClick = False
283     txtStaffID.Text = ""
284
285 End Sub
286
287 End Class
```

Form Name:

**Admin Menu** Logout

**Customers**

Create Customer Search Customer Show Customer Logs

**Equipment**

Create Equipment Search Equipment Stock Check Quick Stock Update Equipment Test Check

**Payroll**

Generating Payroll Search Monthly Payroll Search Yearly Payroll

**Staff**

Create Staff Search Staff Show Workers Logs

**Bookings**

Place a Booking Search Booking Show Un-invoiced Bookings Near Bookings

**Invoices**

Create Invoice Textfile Search Invoices Pay for Invoice Show Unpaid Invoices

Code Listing:

```
1 1 reference
2  Public Class AdminMenu
3     0 references
4     Private Sub butCreateCust_Click(sender As Object, e As EventArgs) Handles butCreateCust.Click
5         'Open form for adding customers
6         AddCust.Show()
7     End Sub
8
9     0 references
10    Private Sub butCreateStaff_Click(sender As Object, e As EventArgs) Handles butCreateStaff.Click
11        'Open for for adding staff
12        AddStaff.Show()
13    End Sub
14
15    0 references
16    Private Sub butCreateEquip_Click(sender As Object, e As EventArgs) Handles butCreateEquip.Click
17        'Open for for adding equipment
18        AddEquipment.Show()
19    End Sub
20
21    0 references
22    Private Sub butPlaceBook_Click(sender As Object, e As EventArgs) Handles butPlaceBook.Click
23        'Open for for placing a booking
24        PlaceBooking.Show()
25    End Sub
26
27    0 references
28    Private Sub butPayBook_Click(sender As Object, e As EventArgs) Handles butPayInvoice.Click
29        'Open for for paying for an invoice
30        PayForInvoice.Show()
31    End Sub
32
33
34
35
36
```

```
37 0 references
38 Private Sub butShowCustLogs_Click(sender As Object, e As EventArgs) Handles butShowCustLogs.Click
39     'Open for for showing customer logs
40     CustomerLogs.Show()
41
42 End Sub
43
44 0 references
45 Private Sub butPayroll_Click(sender As Object, e As EventArgs) Handles butPayroll.Click
46     'Open for for calculating payroll
47     CalculatingPayroll.Show()
48
49 End Sub
50
51 0 references
52 Private Sub butSearchCust_Click(sender As Object, e As EventArgs) Handles butSearchCust.Click
53     'Open for for searching customers
54     SearchCust.Show()
55
56 End Sub
57
58 0 references
59 Private Sub butSearchStaff_Click(sender As Object, e As EventArgs) Handles butSearchStaff.Click
60     'Open for for searching staff
61     SearchStaff.Show()
62
63 End Sub
64
65 0 references
66 Private Sub butSearchEquip_Click(sender As Object, e As EventArgs) Handles butSearchEquip.Click
67     'Open for for searching equipment
68     SearchEquipment.Show()
69
70 End Sub
71
```

```

72 0 references
73 Private Sub butSearchBookID_Click(sender As Object, e As EventArgs) Handles butSearchBooking.Click
74     'Open for for searching bookings
75     SearchBooking.Show()
76
77 End Sub
78
79 0 references
80 Private Sub butSearchUnpaidBooks_Click(sender As Object, e As EventArgs) Handles butSearchUnpaidInvoice.Click
81     'Open for for showing unpaid invoices
82     UnpaidInvoices.Show()
83
84 End Sub
85
86 0 references
87 Private Sub butWorkLogs_Click(sender As Object, e As EventArgs) Handles butWorkLogs.Click
88     'Open for for showing workers logs
89     WorkerLogs.Show()
90
91 End Sub
92
93 0 references
94 Private Sub butCreateInvoice_Click(sender As Object, e As EventArgs) Handles butCreateInvoice.Click
95     'Open for for creating invoices
96     CreatingInvoice.Show()
97
98 End Sub
99
100 0 references
101 Private Sub butUpdateStock_Click(sender As Object, e As EventArgs) Handles butUpdateStock.Click
102     'Open for for updating stock
103     QuickStockUpdate.Show()
104
105 End Sub
106
107 0 references
108 Private Sub butEquipTestCheck_Click(sender As Object, e As EventArgs) Handles butEquipTestCheck.Click
109     'Load equipment from textfile into an array
110     Load_Equipment()
111
112     Dim TestingFound As Boolean = False 'Declare a variable to check if no items require testing
113
114     'Loop through array to check if Next Test Date has been passed
115     For x = 0 To (NumEquipment - 1)
116         If AllEquipment(x).NTestDate < Today.AddDays(7) Then
117             MsgBox("The " & AllEquipment(x).ItemName & " item will/has required testing since " & AllEquipment(x).NTestDate & ", please provide testing")
118             TestingFound = True
119         End If
120     Next
121
122     'Message if there are no items that require testing
123     If TestingFound = False Then
124         MsgBox("No items that require testing have been found")
125     End If
126
127 End Sub
128
129 0 references
130 Private Sub butStockCheck_Click(sender As Object, e As EventArgs) Handles butStockCheck.Click
131     Dim NeedsReorder As String = "" 'Variable to hold the items that need testing
132     Dim AmountMissing As Integer 'Variable to hold the amount of a type of item that is missing
133
134     'Load equipment from textfile into array
135     Load_Equipment()
136
137     'Search through all equipment to find which items are below the reorder level
138     For x = 0 To (NumEquipment - 1)
139         If AllEquipment(x).AmountInStock < AllEquipment(x).ReorderLevel + 1 Then
140
141             AmountMissing = AllEquipment(x).ReorderLevel - AllEquipment(x).AmountInStock 'Calculate how much an item is below the stock level
142             AmountMissing += AllEquipment(x).ReorderAmount 'Add on the reorder amount to get the stock amount past the reorder stock level
143         End If
144     Next
145
146 End Sub

```

```

143
144     'Put items how many of an item are required into
145     If AmountMissing = 1 Then
146         'Grammatical correctness so that if 1 item is missing it won't add an "s"
147         NeedsReorder = NeedsReorder & AmountMissing & " more " & AllEquipment(x).ItemName & " required, "
148     Else
149         NeedsReorder = NeedsReorder & AmountMissing & " more " & AllEquipment(x).ItemName & "s required, "
150     End If
151
152     End If
153 Next
154
155     'Removes the comma at the end
156     NeedsReorder = NeedsReorder.Remove(Len(NeedsReorder) - 2)
157
158     'Print variable
159     MsgBox(NeedsReorder)
160
161 End Sub
162
163 0 references
164 Private Sub butSearchInvoices_Click(sender As Object, e As EventArgs) Handles butSearchInvoices.Click
165     'Open form for searching invoices
166     SearchInvoice.Show()
167
168 End Sub
169
170 0 references
171 Private Sub butUninvoicedBookings_Click(sender As Object, e As EventArgs) Handles butUninvoicedBookings.Click
172     'Open form for showing uninvoiced bookings
173     UninvoicedBookings.Show()
174
175 End Sub
176
177

```

```

176
177 0 references
178 Private Sub butLogout_Click(sender As Object, e As EventArgs) Handles butLogout.Click
179     'Close form
180     Me.Close()
181
182 End Sub
183
184 0 references
185 Private Sub butSearchMonthPay_Click(sender As Object, e As EventArgs) Handles butSearchMonthPay.Click
186     'Open form for searching monthly payroll slips
187     SearchMonthlyPayroll.Show()
188
189 End Sub
190
191 0 references
192 Private Sub butSearchYearPay_Click(sender As Object, e As EventArgs) Handles butSearchYearPay.Click
193     'Open form from searching yearly payroll slips
194     SearchYearlyPayroll.Show()
195
196 End Sub
197
198 0 references
199 Private Sub butTodayBookings_Click(sender As Object, e As EventArgs) Handles butTodayBookings.Click
200     'Opens MyBookingsToday form
201     MyBookingsToday.Show()
202
203 End Sub
204 End Class

```

Form Name:

## Calculating Payroll

**Please fill out the options below**

Staff ID

Month

Rate of Pay (£)

Payroll ID	<input type="text" value=""/>
Hours Worked	<input type="text" value=""/>
Gross Pay (£)	<input type="text" value=""/>
NI (£)	<input type="text" value=""/>
Tax (£)	<input type="text" value=""/>
Pension (£)	<input type="text" value=""/>
Net Pay (£)	<input type="text" value=""/>

Code Listing:

```

1 1 reference
2  Public Class CalculatingPayroll
3      'Declaring a data structure for monthly payroll
4      1 reference
5      Public Structure MonthPayroll
6          Public MonthPayrollID As String
7          Public StaffID As String 'Foreign key
8          Public RateOfPay As Double
9          Public HoursWorked As Double
10         Public GrossPay As Double
11         Public NI As Double
12         Public Tax As Double
13         Public Pension As Double
14         Public NetPay As Double
15     End Structure
16
17     'Declaring a data structure for yearly payroll
18     1 reference
19     Public Structure YearPayroll
20         Public StaffID As String 'Foreign key
21         Public RateOfPay As Double
22         Public HoursWorked As Double
23         Public GrossPay As Double
24         Public NI As Double
25         Public Tax As Double
26         Public Pension As Double
27         Public NetPay As Double
28     End Structure
29
30     'Declaring a variable with the month payroll data structure
31     Public NewMonthPayroll As MonthPayroll
32
33     'Declaring a variable with the year payroll data structure
34     Public NewYearPayroll As YearPayroll
35
36     0 references
37     Private Sub CalculatingPayroll_Load(sender As Object, e As EventArgs) Handles MyBase.Load
38
39         'Load appropriate textfiles
40         Load_Staff()
41         Load_Bookings()
42         Load_Payroll()
43     End Sub
44 End Class

```

```
40
41 'Clear ddl incase anything was loaded previously
42 ddlStaffID.Items.Clear()
43 ddlMonth.Items.Clear()
44
45 'Load staff IDs into ddl
46 For x = 0 To NumStaff - 1
47     ddlStaffID.Items.Add(AllStaff(x).StaffID)
48 Next
49
50 'Load months into ddl
51 ddlMonth.Items.Add("01")
52 ddlMonth.Items.Add("02")
53 ddlMonth.Items.Add("03")
54 ddlMonth.Items.Add("04")
55 ddlMonth.Items.Add("05")
56 ddlMonth.Items.Add("06")
57 ddlMonth.Items.Add("07")
58 ddlMonth.Items.Add("08")
59 ddlMonth.Items.Add("09")
60 ddlMonth.Items.Add("10")
61 ddlMonth.Items.Add("11")
62 ddlMonth.Items.Add("12")
63
64 End Sub
65
66
67 0 references
68 Private Sub butSearchStaff_Click(sender As Object, e As EventArgs) Handles butSearchStaff.Click
69     'Open form for searching staff
70     SearchStaff.Show()
71
72 End Sub
73
74 1 reference
75 Private Sub ddlStaffID_SelectedIndexChanged() Handles ddlStaffID.SelectedIndexChanged
76     'Message if month is empty
77     If ddlMonth.Text = "" Then
78         MsgBox("Please select a month")
79     Exit Sub
80 End If
```

```

81
82 'Payroll ID will consist of the year, month and staff ID
83 txtPayID.Text = Str(Today.Year).Substring(3, 2) & ddlMonth.Text & ddlStaffID.Text
84 NewMonthPayroll.MonthPayrollID = txtPayID.Text
85
86 'Staff ID
87 NewMonthPayroll.StaffID = ddlStaffID.Text
88
89 'Clear pay rate
90 txtRateOfPay.Text = ""
91
92 'User must enter the staff members rate of pay manually
93 MsgBox("Please enter the staff member's rate of pay")
94
95 End Sub
96
97 0 references
98 Private Sub ddlMonth_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlMonth.SelectedIndexChanged
99
100 'Message if staff ID is empty
101 If ddlStaffID.Text = "" Then
102     MsgBox("Please select a staff ID")
103 End If
104
105 ddlStaffID_SelectedIndexChanged() 'Calls the ddlStaffID_SelectedIndexChanged subroutine
106 End Sub
107
108 0 references
109 Private Sub txtRateOfPay_TextChanged(sender As Object, e As EventArgs) Handles txtRateOfPay.TextChanged
110
111 'Check if month and staff ID have been entered
112 If ddlMonth.Text = "" Or ddlStaffID.Text = "" Then
113     txtRateOfPay.Text = ""
114     MsgBox("Please enter the month and staff ID first")
115     Exit Sub
116 End If
117
118 'Reset variables in case something was entered beforehand
119 NewMonthPayroll.GrossPay = 0
120 NewMonthPayroll.NI = 0
121 NewMonthPayroll.Tax = 0
122 NewMonthPayroll.Pension = 0
123
124
125 NewMonthPayroll.NetPay = 0
126
127 *****Calculate gross pay, NI, tax, pension and net pay for the worker*****
128
129 'Gross pay
130 For x = 0 To NumBookings - 1
131     'Compare entered month and staff ID to that of the bookings
132     If AllBookings(x).BookingID.Substring(1, 4) = txtPayID.Text.Substring(0, 4) And AllBookings(x).StaffID = ddlStaffID.Text Then
133         'Adds up the hours of all the bookings from the selected month for the staff member
134         NewMonthPayroll.GrossPay = NewMonthPayroll.GrossPay + AllBookings(x).AllocatedTime
135     End If
136 Next
137
138 'Adding 30% of the rate of pay for the hours where the workers were not booked
139 Dim NotBookedPay As Double
140 NotBookedPay = (30 * 10) - NewMonthPayroll.GrossPay 'Potential hours worked in month (30 days * 10 hours) - Hours booked
141
142 'Hours worked
143 NewMonthPayroll.HoursWorked = NewMonthPayroll.GrossPay 'Gross pay is holding the num hours worked when it gets to this line
144 txtHoursWorked.Text = NewMonthPayroll.HoursWorked
145
146 'Integer check on rate of pay
147 Try
148     '(Hours booked * rate of pay) + (Hours not booked * rate of pay * 30%)
149     NewMonthPayroll.GrossPay = (NewMonthPayroll.GrossPay * txtRateOfPay.Text) + (NotBookedPay * txtRateOfPay.Text * 0.3)
150     txtGross.Text = NewMonthPayroll.GrossPay 'Gross income
151     NewMonthPayroll.RateOfPay = txtRateOfPay.Text 'Rate of pay
152 Catch
153     MsgBox("Rate of pay must be an integer")
154     Exit Sub
155 End Try
156
157 'National insurance
158 If NewMonthPayroll.GrossPay > 792 Then
159     NewMonthPayroll.NI = Math.Round(((NewMonthPayroll.GrossPay - 792) * 0.135), 2)
160 Else
161     NewMonthPayroll.NI = 0
162 End If
163 txtNI.Text = NewMonthPayroll.NI
164
165

```

```

160
161 'Income Tax
162 If NewMonthPayroll.GrossPay > 1048 Then
163     NewMonthPayroll.Tax = Math.Round((NewMonthPayroll.GrossPay - 1048) * 0.2, 2)
164 Else
165     NewMonthPayroll.Tax = 0
166 End If
167 txtTax.Text = NewMonthPayroll.Tax
168
169 'Pension
170 If NewMonthPayroll.GrossPay > 520 Then
171     NewMonthPayroll.Pension = Math.Round((NewMonthPayroll.GrossPay - 520) * 0.2, 2)
172 Else
173     NewMonthPayroll.Pension = 0
174 End If
175 txtPension.Text = NewMonthPayroll.Pension
176
177 'Net pay
178 NewMonthPayroll.NetPay = NewMonthPayroll.GrossPay - (NewMonthPayroll.NI + NewMonthPayroll.Tax + NewMonthPayroll.Pension)
179 txtNetPay.Text = NewMonthPayroll.NetPay
180
181 End Sub
182

```

```

183 0 references
184 Private Sub butGenerate_Click(sender As Object, e As EventArgs) Handles butGenerate.Click
185
186     'Check if all textboxes have values
187     If ddlStaffID.Text = "" Or ddlMonth.Text = "" Or txtRateOfPay.Text = "" Then
188         MsgBox("ERROR: StaffID, month, and rate of pay must all have entered values")
189         Exit Sub
190     End If
191
192     'Check if this monthly payroll already exists
193     Try 'Try loading file named as the monthly payroll
194         Dim file As System.IO.StreamReader
195         file = My.Computer.FileSystem.OpenTextFileReader(Dir$(NewMonthPayroll.MonthPayrollID & ".txt"))
196         MsgBox("ERROR: Monthly payroll file already exists. Masterfile has not been altered")
197     End Try
198
199     'Ask user if they wish to open payroll textfile if it already exists
200     If MsgBox("Click OK if you wish to open the monthly payroll slip", 1) = vbOK Then
201         Process.Start(Dir(NewMonthPayroll.MonthPayrollID & ".txt")) 'Opening the printable payroll slip
202     End If
203

```

```

202     Exit Sub
203 End If
204
205 Catch 'If doesn't work then individual payroll textfile must not exist
206     Exit Try
207 End Try
208
209 'Confirmation that the details are correct
210 If MsgBox("Please confirm that the details are correct", 1, "Confirmation") = vbCancel Then
211     Exit Sub
212 End If
213
214 For x = 0 To NumPayroll - 1
215     If NewMonthPayroll.StaffID = AllPayroll(x).StaffID Then 'Check if staff member is already in the payroll masterfile
216         'Check if pay rate is different within the masterfile compared to the entered value
217         If NewMonthPayroll.RateOfPay <> AllPayroll(x).RateOfPay Then
218             If MsgBox("The entered rate of pay is different to the one stored in the master file. Please select OK if you wish to overwrite the stored rate of pay" + vbCrLf, 1, "Confirmation") = vbOK Then
219                 Exit Sub 'If user selects "Cancel" then exit sub so they can change it
220             End If
221         End If
222     End If
223
224     'Add the new payroll data from the selected month onto the existing payroll data for the year from the masterfile
225     NewMonthPayroll.GrossPay = NewMonthPayroll.GrossPay + AllPayroll(x).GrossPay
226     NewMonthPayroll.NI = NewMonthPayroll.NI + AllPayroll(x).NI
227     NewMonthPayroll.Tax = NewMonthPayroll.Tax + AllPayroll(x).Tax
228     NewMonthPayroll.Pension = NewMonthPayroll.Pension + AllPayroll(x).Pension
229     NewMonthPayroll.NetPay = NewMonthPayroll.NetPay + AllPayroll(x).NetPay
230
231     Save_All_Payroll() 'Updates the payroll masterfile
232
233     MsgBox("Data within the payroll masterfile has been updated")
234     Exit For
235 End For
236
237 Else
238     'Saving data to a master textfile called "Payroll_Masterfile_2022" that holds the data of all the payrolls
239     Dim SaveNewYearlyPayroll As System.IO.StreamWriter
240     SaveNewYearlyPayroll = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Payroll_Masterfile_2022.txt"), True)
241     SaveNewYearlyPayroll.WriteLine(NewMonthPayroll.StaffID & ", " & NewMonthPayroll.RateOfPay & ", " & NewMonthPayroll.HoursWorked & ", " & NewMonthPayroll.GrossPay & ", " & NewMonthPayroll.NI & ", " & NewMonthPayroll.Tax & ", " & NewMonthPayroll.Pension & ", " & NewMonthPayroll.NetPay)
242     SaveNewYearlyPayroll.Close()
243
244     MsgBox("New data saved to the payroll masterfile")
245

```

```

218 vbCrLf + vbCrLf + "Entered rate of pay(£): " & NewMonthPayroll.RateOfPay + vbCrLf + "Masterfile rate of pay(£): " & AllPayroll(x).RateOfPay, 1, "Please confirm") = vbCancel Then
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239 & "," & NewMonthPayroll.NI & "," & NewMonthPayroll.Tax & "," & NewMonthPayroll.Pension & "," & NewMonthPayroll.NetPay)
...

244     End If
245 Next
246
247
248 'Make individual textfile for the month and the staff member
249 Dim SaveIndivMonthTextfile As System.IO.StreamWriter
250 SaveIndivMonthTextfile = My.Computer.FileSystem.OpenTextFileWriter(NewMonthPayroll.MonthPayrollID & ".txt", True)
251 SaveIndivMonthTextfile.WriteLine("***** HANDY MAN HIRING ***** ")
252 SaveIndivMonthTextfile.WriteLine()
253 SaveIndivMonthTextfile.WriteLine(LSet("Month:", 15) & NewMonthPayroll.MonthPayrollID.Substring(2, 2))
254 SaveIndivMonthTextfile.WriteLine(LSet("Payroll ID:", 15) & NewMonthPayroll.MonthPayrollID)
255 SaveIndivMonthTextfile.WriteLine(LSet("Staff ID:", 15) & NewMonthPayroll.StaffID)
256 SaveIndivMonthTextfile.WriteLine()
257 SaveIndivMonthTextfile.WriteLine(LSet("Rate of pay (£):", 20) & NewMonthPayroll.RateOfPay)
258 SaveIndivMonthTextfile.WriteLine(LSet("Hours worked:", 20) & NewMonthPayroll.HoursWorked)
259 SaveIndivMonthTextfile.WriteLine()
260 SaveIndivMonthTextfile.WriteLine(LSet("Gross Income:", 25) & NewMonthPayroll.GrossPay)
261 SaveIndivMonthTextfile.WriteLine(LSet("National Insurance:", 25) & NewMonthPayroll.NI)
262 SaveIndivMonthTextfile.WriteLine(LSet("Income Tax:", 25) & NewMonthPayroll.Tax)
263 SaveIndivMonthTextfile.WriteLine(LSet("Pension:", 25) & NewMonthPayroll.Pension)
264 SaveIndivMonthTextfile.WriteLine(LSet("Net Income:", 25) & NewMonthPayroll.NetPay)
265 SaveIndivMonthTextfile.Close()
266
267 MsgBox("Payroll slip created, please print out" + vbCrLf + "Payroll ID: " & NewMonthPayroll.MonthPayrollID + vbCrLf + vbCrLf + "Opening payroll slip...")
268 Process.Start(Dir(NewMonthPayroll.MonthPayrollID & ".txt")) 'Opening the printable payroll slip
269
270
271 End Sub
272
273 0 references
274 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
275     Me.Close()
276
277 End Sub
278 End Class

```

**Form Name:**

## Create an Invoice

Booking ID

Date

Customer Name

Address

Job Notes

**Code Listing:**

```

1 Public Class CreatingInvoice
2
3     'Declare a structure for invoices
4     1 reference
5     Public Structure Invoice
6         Public InvoiceID As Integer
7         Public BookingID As String
8         Public BookingDate As String
9         Public Price As Double
10        Public Paid As Boolean
11    End Structure
12
13    'Declare a variable with the Invoice data structure
14    Public NewInvoice As Invoice
15
16    Public Position As Integer 'Variable to hold position where booking was found
17
18    'Variables to be used when creating the printable invoice
19    Public CustomerID As String 'Customer details
20    Public CustName As String
21    Public Address As String
22    Public StaffID As String 'Staff details
23    Public StaffName As String
24    Public JobType As String 'Booking details
25    Public BookingDate As String
26    Public AllocatedTime As Double
27    Public Price As String
28    Public JobNotes As String
29
30    1 reference
31    Private Sub CreatingInvoice_Load() Handles MyBase.Load
32
33        'Load relevant files
34        Load_Bookings()
35        Load_Customers()
36        Load_Staff()
37
38        'Loading bookings into the booking ddl
39        ddlBookingID.Items.Clear()
40        For x = 0 To NumBookings - 1
41            'Check that booking was today or before and that it hasn't had an invoice created previously
42            If AllBookings(x).BookingDate < Today.AddDays(1) And AllBookings(x).HasBeenInvoiced = False Then

```

```

41         ddlBookingID.Items.Add(AllBookings(x).BookingID)
42     End If
43 Next
44 End Sub
45
46
47 0 references
48 Private Sub ddlBookingID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlBookingID.SelectedIndexChanged
49     'Load the bookings data into the textboxes
50     For x = 0 To NumBookings - 1
51         'Check if booking ID in textfiles is the same as the one entered
52         If AllBookings(x).BookingID = ddlBookingID.Text Then
53             txtDate.Text = AllBookings(x).BookingDate
54             rtxtJobNotes.Text = AllBookings(x).JobNotes
55             Position = x
56             For y = 0 To NumCustomers - 1 'Compare booking's customer ID to IDs in textfile
57                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
58                     txtCustName.Text = AllCustomers(y).Fname & " " & AllCustomers(y).Sname
59                     rtxtAddress.Text = AllCustomers(y).FAddress + vbCrLf
60                     rtxtAddress.Text = rtxtAddress.Text + AllCustomers(y).SAddress + vbCrLf
61                     rtxtAddress.Text = rtxtAddress.Text + AllCustomers(y).Postcode + vbCrLf
62                     rtxtAddress.Text = rtxtAddress.Text + AllCustomers(y).Town
63                 End If
64             Next
65         End If
66     Next
67 End Sub
68
69
70 0 references
71 Private Sub btnGenInvoice_Click(sender As Object, e As EventArgs) Handles btnGenInvoice.Click
72     'Check that ID has been entered
73     If ddlBookingID.Text = "" Then
74         MsgBox("Invalid booking ID, please try again")
75     Exit Sub
76 End If
77
78     'Confirmation that the details are correct
79     If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
80     Exit Sub 'Exit if user clicks cancel]

```

```

81     End If
82
83     NewInvoice.InvoiceID = 1
84     'Check if there is an invoice with that ID already
85     Load_Invoices()
86     For x = 0 To NumInvoices - 1
87         If NewInvoice.InvoiceID < AllInvoices(x).InvoiceID Then
88             'Sets InvoiceID as the highest InvoiceID within the textfile
89             NewInvoice.InvoiceID = AllInvoices(x).InvoiceID
90         End If
91     Next
92     NewInvoice.InvoiceID = NewInvoice.InvoiceID + 1 'Increment InvoiceID
93
94     'Load data into NewInvoice ID
95     NewInvoice.BookingID = ddlBookingID.Text
96     NewInvoice.BookingDate = txtDate.Text
97     NewInvoice.Price = AllBookings(Position).Price
98     NewInvoice.Paid = False
99
100    'Saving data to a textfile called "Invoices_Masterfile" that holds the data of all the invoices
101    Dim SaveNewInvoice As System.IO.StreamWriter
102    SaveNewInvoice = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Invoices_Masterfile.txt"), True)
103    SaveNewInvoice.WriteLine(NewInvoice.InvoiceID & "," & NewInvoice.BookingID & "," & NewInvoice.BookingDate & "," & NewInvoice.Price & "," & NewInvoice.Paid)
104    SaveNewInvoice.Close()
105
106    '*****Saving a printable unique invoice*****
107
108    'Search customer and staff details for the booking
109    For x = 0 To NumBookings - 1
110        For y = 0 To NumCustomers - 1
111            For z = 0 To NumStaff - 1
112                If AllBookings(x).BookingID = ddlBookingID.Text Then 'Finding booking details
113                    JobType = AllBookings(x).JobType
114                    BookingDate = AllBookings(x).BookingDate
115                    AllocatedTime = AllBookings(x).AllocatedTime
116                    Price = AllBookings(x).Price
117                    JobNotes = AllBookings(x).JobNotes
118                    If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then 'Finding customer details
119                        CustomerID = AllCustomers(y).CustomerID
120                        CustName = txtCustName.Text 'Customer name already found in ddlBookingID_SelectedIndexChanged()
121                        Address = rtxtAddress.Text 'Address already found in ddlBookingID_SelectedIndexChanged()
122                    End If
123                    If AllBookings(x).StaffID = AllStaff(z).StaffID Then 'Finding staff details
124                        StaffID = AllStaff(z).StaffID
125                        StaffName = AllStaff(z).Fname & " " & AllStaff(z).Sname
126                    End If
127                End If
128            Next z
129        Next y
130    Next x
131
132    'Creating and saving the invoice
133    Dim SaveInvoiceIndividualTextfile As System.IO.StreamWriter
134    SaveInvoiceIndividualTextfile = My.Computer.FileSystem.OpenTextFileWriter(NewInvoice.InvoiceID & ".txt", True)
135    SaveInvoiceIndividualTextfile.WriteLine("***** HANDY MAN HIRING *****")
136    SaveInvoiceIndividualTextfile.WriteLine("                Bryncoch, Neath, SA10 7PD")
137    SaveInvoiceIndividualTextfile.WriteLine("                " & Now())
138    SaveInvoiceIndividualTextfile.WriteLine()
139    SaveInvoiceIndividualTextfile.WriteLine()
140    SaveInvoiceIndividualTextfile.WriteLine("Customer:")
141    SaveInvoiceIndividualTextfile.WriteLine(CustName)
142    SaveInvoiceIndividualTextfile.WriteLine(Address)
143    SaveInvoiceIndividualTextfile.WriteLine()
144    SaveInvoiceIndividualTextfile.WriteLine()
145    SaveInvoiceIndividualTextfile.WriteLine(LSet("Invoice ID", 15) & LSet("Booking ID", 15))
146    SaveInvoiceIndividualTextfile.WriteLine(LSet(NewInvoice.InvoiceID, 15) & LSet(NewInvoice.BookingID, 15))
147    SaveInvoiceIndividualTextfile.WriteLine()
148    SaveInvoiceIndividualTextfile.WriteLine(LSet("Booking Date", 24) & LSet("Job Type", 15) & LSet("Staff ID", 15) & LSet("Staff Name", 25))
149    SaveInvoiceIndividualTextfile.WriteLine(LSet(BookingDate, 24) & LSet(JobType, 15) & LSet(StaffID, 11) & LSet(StaffName, 25))
150    SaveInvoiceIndividualTextfile.WriteLine()
151    SaveInvoiceIndividualTextfile.WriteLine("Job Notes:")
152
153    Dim JobNotesLength As String = JobNotes.Length 'Variable to hold job notes length
154    Dim Index As Integer = 0 'Variable to hold which part of job notes the WriteLine() function is taking
155
156    'Go through job notes and ensure they stay within the borders of the invoice
157    While JobNotesLength > 0
158
159        If JobNotesLength < 60 Then
160            'Do this so the substring is within the job notes index
161            SaveInvoiceIndividualTextfile.WriteLine(JobNotes.Substring(Index, JobNotesLength))

```

```

162         Exit While
163     End If
164
165     'First time WriteLine does 0 to 59, second time 60 to 119, etc.
166     SaveInvoiceIndividualTextfile.WriteLine(JobNotes.Substring(Index, 60))
167
168     JobNotesLength = JobNotesLength - 60 'Decrements how many characters are left
169     Index = Index + 60 'Increments the index at which the WriteLine is reading
170 End While
171
172 SaveInvoiceIndividualTextfile.WriteLine()
173 SaveInvoiceIndividualTextfile.WriteLine(LSet("", 75) & "Price: " & Price)
174 SaveInvoiceIndividualTextfile.WriteLine()
175 SaveInvoiceIndividualTextfile.WriteLine("                                Thank you for you Order ")
176 SaveInvoiceIndividualTextfile.WriteLine("Website: HandyManHiringBrynoch.co.uk  email: info@HHBryn.co.uk  Phone:01790 182934")
177 SaveInvoiceIndividualTextfile.Close()
178
179 'Changing variable HasBeenInvoiced to "True" within the booking array for this booking
180 For x = 0 To NumBookings - 1
181     If NewInvoice.BookingID = AllBookings(x).BookingID Then
182         AllBookings(x).HasBeenInvoiced = True
183         Save_All_Bookings() 'Save the edited data
184     End If
185 Next
186
187 MsgBox("Invoice created, please print and hand to customer" + vbCrLf + "Invoice ID: " & Str(NewInvoice.InvoiceID) + vbCrLf + vbCrLf + "Opening invoice...")
188 Process.Start(Dir(NewInvoice.InvoiceID & ".txt")) 'Opening the printable invoice
189
190 ddlBookingID.Text = "" 'Clears textboxes
191 txtCustName.Text = ""
192 txtDate.Text = ""
193 rtxtAddress.Text = ""
194
195 CreatingInvoice_Load() 'Reload the form so the booking that was just invoiced is removed from the ddl
196
197 End Sub
198
199 0 references
200 Private Sub butSearchBookings_Click(sender As Object, e As EventArgs) Handles butSearchBookings.Click
201     'Open form for searching bookings
202     SearchBooking.Show()
203
204 End Sub
205
206 0 references
207 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
208     Me.Close()
209
210 End Sub
211
212 End Class

```

**Form Name:**

## Customer Logs

Customer ID

Dates from  to

**Code Listing:**

```
1 reference
1 Public Class CustomerLogs
2
3     Public AllocatedHoursString As String 'Variable to allow "Allocated hours" to be displayed as a string
4     Public Found As Boolean = False 'Variable to check data has been found
5
6     Private Sub CustomerLogs_Load(sender As Object, e As EventArgs) Handles MyBase.Load
7
8         'Load relevant textfiles
9         Load_Customers()
10        Load_Bookings()
11        Load_Staff()
12        Load_Invoices()
13
14        'Clear ddl incase there was something there previously
15        ddlCustID.Items.Clear()
16
17        For x = 0 To NumCustomers - 1
18            'Load customer IDs into the ddl
19            ddlCustID.Items.Add(AllCustomers(x).CustomerID)
20        Next
21
22    End Sub
23
24    Private Sub butSearch_Click(sender As Object, e As EventArgs) Handles butSearch.Click
25
26        'Presence check on customer ID
27        If ddlCustID.Text = "" Then
28            rtxtLogs.Text = "No bookings found as no customer ID has been entered"
29            Exit Sub
30        End If
31
32        rtxtLogs.Text = "" 'Clear rich textbox incase something was loaded in before
33        Found = False 'Resets "Found"
34
35        'Format check on dates
36        If txtMinDate.Text <> "" Then
37            Dim DateFormatCheck As Boolean = txtMinDate.Text Like "***/**/*****"
38            If DateFormatCheck = False Or Len(txtMinDate.Text) <> 10 Then
39                MsgBox("Please ensure the dates are in the format '**/**/*****")
40            Exit Sub
41        End If
42    End Sub
43 End Class
```

```

40         Exit Sub
41     End If
42 End If
43 If txtMaxDate.Text <> "" Then
44     Dim DateFormatCheck As Boolean = txtMaxDate.Text Like "***/**/*****"
45     If DateFormatCheck = False Or Len(txtMaxDate.Text) <> 10 Then
46         MsgBox("Please ensure the dates are in the format '**/**/*****")
47     Exit Sub
48 End If
49 End If
50
51 *****IF NO DATES ARE ENTERED*****
52
53 If txtMinDate.Text = "" And txtMaxDate.Text = "" Then
54     'Load all the customers bookings and their details
55     For x = 0 To NumBookings - 1
56         'Add the details of each booking to the rich textbox
57         If ddlCustID.Text = AllBookings(x).CustomerID Then
58             rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
59             rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
60             rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
61             rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
62             rtxtLogs.Text = rtxtLogs.Text + "Staff ID: " & AllBookings(x).StaffID + vbCrLf
63
64             'Load staff members name from staff textfile
65             For y = 0 To NumStaff - 1
66                 If AllBookings(x).StaffID = AllStaff(y).StaffID Then
67                     rtxtLogs.Text = rtxtLogs.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
68                 End If
69             Next
70
71             rtxtLogs.Text = rtxtLogs.Text + vbCrLf
72             AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
73             rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
74
75             rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
76             rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
77             rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
78             rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
79
80             If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
81                 For y = 0 To NumInvoices - 1
82                     If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
83                         rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
84                     End If
85                 Next
86
87             ElseIf AllBookings(x).HasBeenInvoiced = False Then
88                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
89             End If
90
91             Found = True
92         End If
93     Next
94
95 ElseIf txtMinDate.Text <> "" Then *****IF MIN DATE ENTERED *****
96     If txtMaxDate.Text <> "" Then
97         For x = 0 To NumBookings
98             If AllBookings(x).CustomerID = ddlCustID.Text And AllBookings(x).BookingDate > txtMinDate.Text And AllBookings(x).BookingDate < txtMaxDate.Text Then
99                 rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
100                rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
101                rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
102                rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
103                rtxtLogs.Text = rtxtLogs.Text + "Staff ID: " & AllBookings(x).StaffID + vbCrLf
104
105                'Load staff members name from staff textfile
106                For y = 0 To NumStaff - 1
107                    If AllBookings(x).StaffID = AllStaff(y).StaffID Then
108                        rtxtLogs.Text = rtxtLogs.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
109                    End If
110                Next
111
112                rtxtLogs.Text = rtxtLogs.Text + vbCrLf
113                AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
114                rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
115
116                rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
117                rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
118                rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
119                rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
120

```

```

122     If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
123         For y = 0 To NumInvoices - 1
124             If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
125                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
126             End If
127         Next
128
129     ElseIf AllBookings(x).HasBeenInvoiced = False Then
130         rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
131     End If
132
133     Found = True
134 End If
135 Next
136
137
138 Else ***** IF ONLY MIN DATE ENTERED*****
139 For x = 0 To NumBookings
140     If AllBookings(x).CustomerID = ddlCustID.Text And AllBookings(x).BookingDate > txtMinDate.Text Then
141         rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
142         rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
143         rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
144         rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
145         rtxtLogs.Text = rtxtLogs.Text + "Staff ID: " & AllBookings(x).StaffID + vbCrLf
146
147         'Load staff members name from staff textfile
148         For y = 0 To NumStaff - 1
149             If AllBookings(x).StaffID = AllStaff(y).StaffID Then
150                 rtxtLogs.Text = rtxtLogs.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
151             End If
152         Next
153
154         rtxtLogs.Text = rtxtLogs.Text + vbCrLf
155         AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
156         rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
157
158         rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
159         rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
160         rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
161         rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
162     End If
163
164     If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
165         For y = 0 To NumInvoices - 1
166             If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
167                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
168             End If
169         Next
170
171     ElseIf AllBookings(x).HasBeenInvoiced = False Then
172         rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
173     End If
174
175     Found = True
176 End If
177 Next
178 End If
179
180 ElseIf txtMaxDate.Text <> "" Then *****IF MAX DATE ENTERED*****
181     If txtMinDate.Text <> "" Then
182         For x = 0 To NumBookings
183             If AllBookings(x).CustomerID = ddlCustID.Text And AllBookings(x).BookingDate > txtMinDate.Text And AllBookings(x).BookingDate < txtMaxDate.Text Then
184                 rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
185                 rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
186                 rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
187                 rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
188                 rtxtLogs.Text = rtxtLogs.Text + "Staff ID: " & AllBookings(x).StaffID + vbCrLf
189
190                 'Load staff members name from staff textfile
191                 For y = 0 To NumStaff - 1
192                     If AllBookings(x).StaffID = AllStaff(y).StaffID Then
193                         rtxtLogs.Text = rtxtLogs.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
194                     End If
195                 Next
196
197                 rtxtLogs.Text = rtxtLogs.Text + vbCrLf
198                 AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
199                 rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
200
201                 rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
202                 rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
203                 rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf

```

```

204 rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
205
206     If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
207         For y = 0 To NumInvoices - 1
208             If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
209                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
210             End If
211         Next
212
213     ElseIf AllBookings(x).HasBeenInvoiced = False Then
214         rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
215     End If
216
217     Found = True
218 End If
219 Next
220
221
222 Else '*****IF ONLY MAX DATE ENTERED*****
223 For x = 0 To NumBookings
224     If AllBookings(x).CustomerID = ddlCustID.Text And AllBookings(x).BookingDate < txtMaxDate.Text Then
225         rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
226         rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
227         rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
228         rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
229         rtxtLogs.Text = rtxtLogs.Text + "Staff ID: " & AllBookings(x).StaffID + vbCrLf
230
231         'Load staff members name from staff textfile
232         For y = 0 To NumStaff - 1
233             If AllBookings(x).StaffID = AllStaff(y).StaffID Then
234                 rtxtLogs.Text = rtxtLogs.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
235             End If
236         Next
237
238         rtxtLogs.Text = rtxtLogs.Text + vbCrLf
239         AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
240         rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
241
242         rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
243         rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
244         rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
245
246         rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
247
248         If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
249             For y = 0 To NumInvoices - 1
250                 If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
251                     rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
252                 End If
253             Next
254
255         ElseIf AllBookings(x).HasBeenInvoiced = False Then
256             rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
257         End If
258
259         Found = True
260     End If
261 Next
262
263 End If
264
265 'Message if no data was found
266 If Found = False Then
267     rtxtLogs.Text = "No bookings found"
268 End If
269
270 End Sub
271
272 0 references
273 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
274     Me.Close()
275 End Sub
276
277 0 references
278 Private Sub butSearchCustomers_Click(sender As Object, e As EventArgs) Handles butSearchCustomers.Click
279     'Open form for searching customers
280     SearchCust.Show()
281 End Sub
282
283 End Class

```

**Form Name:**

The image shows a login form with a light blue background. At the top center, the word "Login" is written in a large, bold, black font. Below this, there are three input fields arranged vertically. The first field is labeled "ID", the second "Password", and the third "Clue". Each label is positioned to the left of its corresponding white input box. At the bottom of the form, there are three buttons: "Login" on the left, "Forgot Password" in the center, and "Cancel" on the right. All buttons have a light blue background and a thin black border.

**Code Listing:**

```

1 reference
1 Public Class Login
2
3     'Declare a variable to be able to find the staff ID within the password file
4     Public FoundID As String
5
6     'Variable so that the clue appears if the user fails to logon 3 times
7     Public AttemptCount As Integer = 0
8
9     0 references
10    Private Sub Login_Load(sender As Object, e As EventArgs) Handles MyBase.Load
11
12        'Load all the staff and passwords
13        Load_Staff()
14        Load_Password()
15    End Sub
16
17    0 references
18    Private Sub txtID_TextChanged(sender As Object, e As EventArgs) Handles txtID.TextChanged
19
20        'Loop through staff file to find staff ID whenever the textbox value is changed
21        For x = 0 To (NumStaff - 1)
22            If txtID.Text = AllStaff(x).StaffID Then 'Comparing if the staff ID exists
23                FoundID = AllStaff(x).StaffID
24                Exit For
25            End If
26        Next
27    End Sub
28
29    0 references
30    Private Sub butLogin_Click(sender As Object, e As EventArgs) Handles butLogin.Click
31
32        'Loop through password file to find password with according staff ID and compare with entered password
33        For x = 0 To (NumStaff - 1)
34            If AllPasswords(x).StaffID = FoundID And txtPassword.Text = AllPasswords(x).Password Then
35                MsgBox("Welcome!!")
36
37                'Changes global variables to hold the staff type and ID
38                LoggedOnStaffID = AllPasswords(x).StaffID
39                LoggedOnStaffType = AllPasswords(x).StaffType
40
41                'Clear text boxes

```

```

40         txtID.Text = ""
41         txtPassword.Text = ""
42         txtClue.Text = ""
43
44         If AllPasswords(x).StaffType = "Worker" Then 'Opening menu depending on staff type
45             'Opens worker staff menu
46             StaffMenu.Show()
47             Exit Sub
48         Else
49             'Opens admin menu if not a worker
50             AdminMenu.Show()
51             Exit Sub
52         End If
53     End If
54 Next
55
56 'Message if they could not logon
57 MsgBox("ID was not found or password did not match (Please note that both fields are case sensitive)")
58 AttemptCount = AttemptCount + 1 'Increment attempt count
59
60 'Show clue if number of attempts is 3 or more
61 If AttemptCount > 2 Then
62     For x = 0 To (NumStaff - 1)
63         If AllPasswords(x).StaffID = FoundID Then
64             txtClue.Text = AllPasswords(x).Clue
65         End If
66     Next
67 End If
68
69 End Sub

```

```

70
71 0 references
72 Private Sub butForgPass_Click(sender As Object, e As EventArgs) Handles butForgPass.Click
73
74     'Finding clue according to the entered staff ID
75     For x = 0 To (NumStaff - 1)
76         If AllPasswords(x).StaffID = FoundID Then
77             txtClue.Text = AllPasswords(x).Clue
78             Exit Sub 'Exit subroutine once found
79         End If
80     Next

```

```

81     MsgBox("This staff ID was not found so no clue can be shown")
82
83 End Sub

```

```

84
85 0 references
86 Private Sub butCancel_Click(sender As Object, e As EventArgs) Handles butCancel.Click
87
88     Me.Close()
89
90 End Sub
End Class

```

**Form Name:**

## My Near Bookings

Close

**Code Listing:**

```

1  Public Class MyBookingsToday
2
3      Public AllocatedHoursString As String 'Variable to allow "Allocated hours" to be displayed as a string
4      Public Found As Boolean = False 'Variable to check data has been found
5
6  #region 0 references
7      Private Sub MyBookingsToday_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8
9          'Clears rtxt incase something was there before
10         rtxtBookingDetails.Text = ""
11
12         'Load bookings and customers from textfiles
13         Load_Bookings()
14         Load_Customers()
15
16         '*****TODAY*****
17         'Displays "Today" text
18         rtxtBookingDetails.Text = "TODAY - - - TODAY - - - TODAY - - - TODAY - - - TODAY" + vbCrLf + vbCrLf
19
20         'Goes through bookings textfile
21         For x = 0 To NumBookings - 1
22             'Check if the booking is today and if the logged on staff member is the one with that booking
23             If AllBookings(x).BookingDate.Substring(0, 10) = Today And AllBookings(x).StaffID = LoggedOnStaffID Then
24                 'Add the details of each booking to the rich textbox
25                 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "- - - - -" + vbCrLf
26                 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
27                 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
28                 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
29
30                 'Search through customers textfile, add address of customer to rich textbox
31                 For y = 0 To NumCustomers - 1
32                     If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
33                         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
34                         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllC
35                     Exit For
36                 End If
37             Next
38
39             AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
40             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
41             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Start time: " & AllBookings(x).BookingDate.Substring(11, 5) + vbCrLf
42             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
43
44             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf + vbCrLf
45             Found = True
46         End If
47     Next
48
49     '*****TOMORROW*****
50     'Displays "Tomorrow" text
51     rtxtBookingDetails.Text = "TOMORROW - - - TOMORROW - - - TOMORROW - - - TOMORROW - - - TOMORROW" + vbCrLf + vbCrLf
52
53     'Goes through bookings textfile
54     For x = 0 To NumBookings - 1
55         'Check if the booking is today and if the logged on staff member is the one with that booking
56         If AllBookings(x).BookingDate.Substring(0, 10) = Today.AddDays(1) And AllBookings(x).StaffID = LoggedOnStaffID Then
57             'Add the details of each booking to the rich textbox
58             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "- - - - -" + vbCrLf
59             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
60             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf
61             rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
62
63             'Search through customers textfile, add address of customer to rich textbox
64             For y = 0 To NumCustomerID - 1
65                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
66                     rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
67                     rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).P
68                 Exit For
69             End If
70         Next
71
72         AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
73         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
74         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Start time: " & AllBookings(x).BookingDate.Substring(11, 5) + vbCrLf
75         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
76         rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf + vbCrLf
77         Found = True
78     End If
79 Next
80
81 '*****IN 2 DAYS*****
82 'Displays "Tomorrow" text
83 rtxtBookingDetails.Text = "IN 2 DAYS - - - IN 2 DAYS - - - IN 2 DAYS - - - IN 2 DAYS - - - IN 2 DAYS" + vbCrLf + vbCrLf

```

```

64 Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
65 Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf + vbCrLf
66

83 'Goes through bookings textfile
84 For x = 0 To NumBookings - 1
85 'Check if the booking is today and if the logged on staff member is the one with that booking
86 If AllBookings(x).BookingDate.Substring(0, 10) = Today.AddDays(2) And AllBookings(x).StaffID = LoggedOnStaffID Then
87 'Add the details of each booking to the rich textbox
88 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "-----" + vbCrLf
89 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
90 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
91 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
92
93 'Search through customers textfile, add address of customer to rich textbox
94 For y = 0 To NumCustomers - 1
95 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
96 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
97 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf + vbCrLf
98 Exit For
99 End If
100 Next
101
102 AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
103 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
104 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Start time: " & AllBookings(x).BookingDate.Substring(11, 5) + vbCrLf
105 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
106 rtxtBookingDetails.Text = rtxtBookingDetails.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf + vbCrLf
107 Found = True
108 End If
109 Next
110
111 'Message if no data was found
112 If Found = False Then
113 rtxtBookingDetails.Text = "No bookings were found for the next 3 days"
114 End If
115
116 End Sub
117
118 0 references
119 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
120 Me.Close()
121 End Sub
122 End Class

96 Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
97 Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf + vbCrLf
98

```

**Form Name:**

## Paying for a Booking

**Please enter Invoice ID**

Invoice ID

Booking ID

Customer Name

Date

Job Notes

Price (£)

VAT (£)

Gross Price (£)

**Code Listing:**

```

1 1 reference
2  Public Class PayForInvoice
3  0 references
4  Private Sub PayForInvoice_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5      'Load textfiles
6      Load_Invoices()
7      Load_Bookings()
8      Load_Customers()
9
10     'Load invoices into invoice ddl and clear first incase anything was loaded beforehand
11     ddlInvoiceID.Items.Clear()
12     For x = 0 To NumInvoices - 1
13         If AllInvoices(x).Paid = False Then 'Check if the invoice has not been paid for
14             ddlInvoiceID.Items.Add(AllInvoices(x).InvoiceID)
15         End If
16     Next
17
18 End Sub
19
20 0 references
21 Private Sub ddlInvoiceID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlInvoiceID.SelectedIndexChanged
22
23     Dim Price As Double
24     Dim VAT As Double
25
26     'Load the booking details of the invoice into the textboxes
27     For x = 0 To NumInvoices - 1
28         For y = 0 To NumBookings - 1
29             'Compare the booking ID from the invoice textfile to those of the Bookings textfile
30             If AllInvoices(x).InvoiceID = ddlInvoiceID.Text And AllInvoices(x).BookingID = AllBookings(y).BookingID Then
31                 'Loads booking details from textfile
32                 txtBookingID.Text = AllBookings(y).BookingID
33                 txtDate.Text = AllBookings(y).BookingDate
34                 rtxtJobNotes.Text = AllBookings(y).JobNotes
35                 txtPrice.Text = AllBookings(y).Price.Substring(1, Len(AllBookings(y).Price) - 1)
36                 Price = txtPrice.Text
37                 VAT = Price * 0.2 'Calculating VAT and Total
38                 txtVAT.Text = VAT
39                 txtGross.Text = Price + VAT
40             End If
41         Next y
42     Next x
43
44     'Loads customer name from textfile

```

```

41     For z = 0 To NumCustomers - 1
42         If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
43             txtCustName.Text = AllCustomers(z).Fname & " " & AllCustomers(z).Sname
44         End If
45     Next
46 End If
47 Next
48 Next
49
50 End Sub

```

```

51
52 0 references
53 Private Sub butConfirmPayment_Click(sender As Object, e As EventArgs) Handles butConfirmPayment.Click
54     'Check that an invoice has been selected
55     If ddlInvoiceID.Text = "" Then
56         MsgBox("No invoice ID selected, please try again")
57     Exit Sub
58 End If
59
60     'Confirmation that customer has paid
61     Dim PayCheck As String
62     PayCheck = InputBox("Please confirm, has customer provided payment? (Y/N)")
63
64     While PayCheck <> "N" And PayCheck <> "Y" 'Insisting an answer of "Y" or "N"
65         PayCheck = InputBox("Must answer with 'Y' (meaning yes) or 'N' (meaning no)" + vbCrLf + "Please confirm, has customer provided payment? (Y/N)")
66     End While
67
68     If PayCheck = "N" Then 'If payment has not been made
69         MsgBox("Please tell the customer to provide payment")
70     Exit Sub
71     ElseIf PayCheck = "Y" Then 'IF payment has been made
72         MsgBox("Thank you for your payment, data saved")
73     End If
74
75     'Sets the "Paid" variable for this invoice in the Invoice textfile to "True"
76     For x = 0 To NumInvoices - 1
77         If AllInvoices(x).InvoiceID = ddlInvoiceID.Text Then
78             AllInvoices(x).Paid = True
79         End If
80     Next

```

```

81
82     'Save "Paid = True" into the Invoice textfile
83     Save_All_Invoices()
84
85     'Closes form once invoice has been paid for
86     Me.Close()
87
88 End Sub

```

```

89
90 0 references
91 Private Sub butCancelPay_Click(sender As Object, e As EventArgs) Handles butCancelPay.Click
92     Me.Close()
93
94 End Sub

```

```

95
96 0 references
97 Private Sub butSearchInvoices_Click(sender As Object, e As EventArgs) Handles butSearchInvoices.Click
98     'Open form for searching invoices
99     SearchInvoice.Show()
100
101 End Sub
102 End Class

```

**Form Name:**

## Placing a Booking

Booking ID	<input type="text"/>	<input type="button" value="Generate ID"/>	
Job Type	<input type="text"/>		(Data automatically filled once customer ID selected)
Date	<input type="text" value="27/03/2022 - 12:00 AM"/>		
Customer ID	<input type="text"/>	<input type="button" value="Search Customers"/>	
Staff ID	<input type="text"/>		
Allocated Time (hrs)	<input type="text"/>		
Price (£)	<input type="text"/>		
	(This does not including VAT)		
Tools for job	<input type="text"/>	<input type="button" value="Add Tool"/> <input type="button" value="Remove Tool"/>	(Data automatically filled once staff ID selected)
	<input type="text"/>		
Job Specific Notes	<input type="text"/>		
	<input type="text"/>		
	<input type="button" value="Save"/> <input type="button" value="Clear"/> <input type="button" value="Quit"/>		

### Customer Details (not editable)

First Name	<input type="text"/>
Surname	<input type="text"/>
1st Address Line	<input type="text"/>
2nd Address Line	<input type="text"/>
Postcode	<input type="text"/>
Town/City	<input type="text"/>

### Staff Details (not editable)

First Name	<input type="text"/>
Surname	<input type="text"/>

**Code Listing:**

```
1 reference
1 Public Class PlaceBooking
2
3     'Declare a structure for bookings
4     1 reference
5     Public Structure Booking
6         Public BookingID As String
7         Public JobType As String
8         Public BookingDate As String
9         Public CustomerID As String
10        Public StaffID As String
11        Public AllocatedTime As Double 'Double is a decimal number
12        Public Price As Double
13        Public ToolsList As String
14        Public JobNotes As String
15        Public HasBeenInvoiced As Boolean
16    End Structure
17
18    Public NewBooking As Booking 'Declare a variable with the Booking data structure
19
20    Public SameBookingDateCount As Integer 'Variable to count how many bookings there are with the same date
21    Public Stafffound As Boolean 'Variable to check if a staff member within the job type has been found
22
23    Public ToolArray(0 To 99) 'Array for the job tools to be kept
24    Public ToolPosition As Integer = 0
25
26    0 references
27    Private Sub PlaceBooking_Load(sender As Object, e As EventArgs) Handles MyBase.Load
28
29        'Load textfiles
30        Load_Customers()
31        Load_Staff()
32        Load_Equipment()
33        Load_Password()
34
35        'Loading items into "Job Type" drop down list
36        ddlJobType.Items.Clear()
37        ddlJobType.Items.Add("Welding")
38        ddlJobType.Items.Add("Woodwork")
39        ddlJobType.Items.Add("Plumbing")
40        ddlJobType.Items.Add("Carpentry")
41        ddlJobType.Items.Add("Electrician")
42        ddlJobType.Items.Add("Gardening")
43    End Sub
44 End Class
```

```

41     ddlJobType.Items.Add("Roofing")
42     ddlJobType.Items.Add("Other")
43
44     'Loading customers into the customer ddl
45     ddlCustID.Items.Clear()
46     For x = 0 To NumCustomers - 1
47         ddlCustID.Items.Add(AllCustomers(x).CustomerID)
48     Next
49
50     'Loading equipment into the tools ddl
51     ddlTools.Items.Clear()
52     For x = 0 To NumEquipment - 1
53         ddlTools.Items.Add(AllEquipment(x).ItemName)
54     Next
55
56     'Staff is loaded into the ddl when the job type is selected
57
58     'Sets the booking date automatically to today
59     calendarBookDate.Value = Today
60
61 End Sub

```

1 reference

```

63 Private Sub butGenID_Click() Handles butGenID.Click
64
65     'Generating part of the new booking ID
66     Dim StringDate As String = calendarBookDate.Value
67     NewBooking.BookingID = "0" & StringDate.Substring(8, 2) & StringDate.Substring(3, 2) & StringDate.Substring(0, 2)
68
69     Load_Bookings() 'Load bookings textfile to calculate the number to be added on the end
70     SameBookingDateCount = 1
71     For x = 0 To NumBookings - 1
72         'Compares the date of saved bookings to that of the booking being placed
73         If AllBookings(x).BookingDate.Substring(0, 9) = StringDate.Substring(0, 9) Then
74             SameBookingDateCount = SameBookingDateCount + 1 'Incremented for every booking with the same date
75         End If
76     Next
77
78     'Final new booking ID
79     NewBooking.BookingID = NewBooking.BookingID & SameBookingDateCount
80

```

```

81     txtBookingID.Text = NewBooking.BookingID 'Show generated ID in the textbox
82
83 End Sub

```

1 reference

```

85 Private Sub ddlJobType_SelectedIndexChanged() Handles ddlJobType.SelectedIndexChanged
86
87
88     'If job type changed, clear staff ID and name
89     ddlStaffID.Text = ""
90     txtFnameStaff.Text = ""
91     txtSnameStaff.Text = ""
92
93     StaffFound = False 'Variable to check if staff member has been found with appropriate speciality
94
95     'Clear all the staff beforehand so there is no data duplication
96     ddlStaffID.Items.Clear()
97
98     'Load in all the staff members with the same speciality as the job type
99     For x = 0 To NumStaff - 1
100         If ddlJobType.Text = AllStaff(x).Speciality Then
101             ddlStaffID.Items.Add(AllStaff(x).StaffID)
102             StaffFound = True
103         End If
104     Next
105
106     'If no staff members have that speciality or "Other" is selected, it loads all the staff members
107     If StaffFound = False Or ddlJobType.Text = "Other" Then
108         ddlStaffID.Items.Clear()
109         For x = 0 To NumStaff - 1
110             ddlStaffID.Items.Add(AllStaff(x).StaffID)
111         Next
112     End If
113
114     'Doesn't remove admin users as they also have a speciality
115
116     'Checking if the loaded staff members already have a booking during the selected time slot
117     For x = 0 To NumBookings - 1
118         'Compares the date part of the booking ID to the booking IDs in the textfile (i.e. from 02203211 to takes 220321 which means 21/03/2022)
119         If AllBookings(x).BookingID.Substring(1, 6) = txtBookingID.Text.Substring(1, 6) Then
120             ddlStaffID.Items.Remove(AllBookings(x).StaffID) 'Removes staff ID if that staff member has a booking on that day
121         End If

```

```

124 'If there is nothing in the staff ID ddl, it reloads all the staff and removes the unavailable staff again
125 'It does this because the first time its the specialist staff loaded, this time it will be all the staff
126 If ddlStaffID.Items.Count = 0 Then
127     'Load all staff
128     For x = 0 To NumStaff - 1
129         ddlStaffID.Items.Add(AllStaff(x).StaffID)
130     Next
131
132     'Remove unavailable staff
133     For x = 0 To NumBookings - 1
134         'Compares the date part of the booking ID to the booking IDs in the textfile (i.e. from 02203211 to takes 220321 which means 21/03/2022)
135         If AllBookings(x).BookingID.Substring(1, 6) = txtBookingID.Text.Substring(1, 6) Then
136             ddlStaffID.Items.Remove(AllBookings(x).StaffID) 'Removes staff ID if that staff member has a booking on that day
137         End If
138     Next
139
140 End If
141
142 End Sub
143
144 0 references
145 Private Sub calendarBookDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarBookDate.ValueChanged
146
147     'Clicks the generate ID button
148     butGenID_Click()
149
150     'Calls the ddlJobType_SelectedIndexChanged() subroutine as staff member availability may be different for the newly selected day
151     ddlJobType_SelectedIndexChanged()
152 End Sub
153
154 0 references
155 Private Sub ddlCustID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlCustID.SelectedIndexChanged
156
157     'Showing customer's address and names in textboxes
158     'This data is not saved in the booking textfile
159     For x = 0 To NumCustomers - 1
160         If ddlCustID.Text = AllCustomers(x).CustomerID Then
161             txtFnameCust.Text = AllCustomers(x).Fname
162             txtSnameCust.Text = AllCustomers(x).Sname
163             txtFAddress.Text = AllCustomers(x).FAddress
164             txtSAddress.Text = AllCustomers(x).SAddress

```

```

164         txtPostcode.Text = AllCustomers(x).Postcode
165         txtTown.Text = AllCustomers(x).Town
166     End If
167 Next
168
169 End Sub
170
171 0 references
172 Private Sub ddlStaffID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlStaffID.SelectedIndexChanged
173     'Showing staff's name in textboxes
174     'This data is not saved in the booking textfile
175     For x = 0 To NumStaff - 1
176         If ddlStaffID.Text = AllStaff(x).StaffID Then
177             txtFnameStaff.Text = AllStaff(x).Fname
178             txtSnameStaff.Text = AllStaff(x).Sname
179         End If
180     Next
181
182 End Sub
183
184 0 references
185 Private Sub butAddTool_Click(sender As Object, e As EventArgs) Handles butAddTool.Click
186     'Checking if tools ddl is empty
187     If ddlTools.Text = "" Then
188         MsgBox("Please add a tool from the drop down list")
189         Exit Sub
190     End If
191
192     'Check if the first tool has been added
193     If ToolPosition = 0 Then
194         'Saving new tool to tools textbox without the "+" at the start
195         NewBooking.ToolsList = NewBooking.ToolsList & ddlTools.Text 'Adds tool in ddl to NewBooking array
196         rtxtTools.Text = NewBooking.ToolsList 'Display in textbox
197         ToolArray(ToolPosition) = ddlTools.Text 'Enters tool into first slot of the ToolArray
198         ToolPosition = ToolPosition + 1
199         ddlTools.Text = "" 'Clear tool ddl
200         Exit Sub
201     End If
202
203     'Saving new tool to tools variable with the "+" at the start
204     NewBooking.ToolsList = NewBooking.ToolsList & " + " & ddlTools.Text

```

```

205     rtxtTools.Text = NewBooking.ToolsList 'Display in textbox
206     ToolArray(ToolPosition) = " " & ddlTools.Text 'Enters tool into next slot of the array
207     ToolPosition = ToolPosition + 1 'Increments tool position
208     ddlTools.Text = "" 'Clear tool ddl
209
210 End Sub
211

```

0 references

```

212 Private Sub butRemoveTool_Click(sender As Object, e As EventArgs) Handles butRemoveTool.Click
213
214     'Checks if nothing is in the textbox
215     If ToolPosition < 0 Then
216         ToolPosition = 0 'Resets tool position
217     End If
218
219     'Check if a tool has been entered first
220     If ToolPosition = 0 Then
221         MsgBox("Please add a tool from the drop down list")
222         Exit Sub
223     End If
224
225     NewBooking.ToolsList = "" 'Clears tools list from NewBooking array
226
227     For x = 0 To ToolPosition - 2 'Goes to "position - 2" because position is set as 1 ahead of the last item
228         NewBooking.ToolsList = NewBooking.ToolsList & ToolArray(x) 'Load tools from ToolArray into tools list in NewBooking array
229     Next
230
231     ToolPosition = ToolPosition - 1 'Decrements tool position as one has been removed
232     rtxtTools.Text = NewBooking.ToolsList 'Display in tools textbox
233
234 End Sub
235

```

0 references

```

236 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
237
238     NewBooking.JobType = ddlJobType.Text
239
240     'Checking that the booking date is either today or onwards
241     If calendarBookDate.Value < Now.AddHours(2) Then
242         MsgBox("Please select a booking date at least 2 hours from now")
243         Exit Sub
244     End If

```

```

246     'Checking that the allocated hours and start time allow the booking to finish at 6pm at the latest
247     Dim FinishTime = calendarBookDate.Value.Hour + Int(txtAllocatedHours.Text) 'Calculating what time job finishes
248     If FinishTime > 18 Then
249         MsgBox("Please either select a different time or reduce the amount of hours allocated (latest a worker will finish is 6pm)")
250         Exit Sub
251     End If
252
253     'Checking that job doesn't start too early
254     If calendarBookDate.Value.Hour < 8 Then
255         MsgBox("Job cannot start before 8 am, please try again")
256         Exit Sub
257     End If
258
259     NewBooking.BookingDate = calendarBookDate.Value
260
261     'Presence check of customer and staff ID
262     If ddlCustID.Text = "" Or ddlStaffID.Text = "" Then
263         MsgBox("Please enter a staff and customer ID for the booking")
264         Exit Sub
265     End If
266
267     NewBooking.CustomerID = ddlCustID.Text
268     NewBooking.StaffID = ddlStaffID.Text
269
270     'Range check on time
271     If txtAllocatedHours.Text < 0.5 Or txtAllocatedHours.Text > 10 Then
272         MsgBox("Allocated time must be between 0.5 and 10 hrs, please try again")
273         Exit Sub
274     End If
275
276     'Integer/double check on allocated time
277     Try
278         NewBooking.AllocatedTime = txtAllocatedHours.Text
279     Catch
280         MsgBox("Allocated time must be an integer/decimal point value")
281         Exit Sub
282     End Try
283
284     'Checking that the price does not include "," or "£"
285     Dim PriceCommaCheck() As Char = txtPrice.Text.ToCharArray
286     For x = 0 To Len(txtPrice.Text) - 1
287         If PriceCommaCheck(x) = "," Or PriceCommaCheck(x) = "£" Then

```

```

288         MsgBox("The price must not include a comma or '£', please try again")
289         Exit Sub
290     End If
291 Next
292
293 'Integer (double) check on price
294 Try
295     NewBooking.Price = "£" & txtPrice.Text
296 Catch
297     MsgBox("Price must be an integer/decimal point value")
298     Exit Sub
299 End Try
300
301 'If nothing entered into tools list, check that this is correct
302 If rtxtTools.Text = "" Then
303     'Confirm with user whether tools list is meant to be empty
304     If MsgBox("Nothing was entered into tools." + vbCrLf + "Click 'OK' if this is correct.", 1, "Tools error") = vbCancel Then
305         Exit Sub
306     End If
307 End If
308
309 NewBooking.ToolsList = rtxtTools.Text
310
311 'Presence check on job notes
312 If rtxtNotes.Text = "" Then
313     MsgBox("Notes must be entered for the job")
314     Exit Sub
315 End If
316
317 'Checking that the notes do not include ","
318 Dim NotesCommaCheck() As Char = rtxtNotes.Text.ToCharArray
319 For x = 0 To Len(rtxtNotes.Text) - 1
320     If NotesCommaCheck(x) = "," Then
321         MsgBox("Your notes must not include a comma, please try again")
322         Exit Sub
323     End If
324 Next
325 NewBooking.JobNotes = rtxtNotes.Text
326
327
328 'Has been invoiced" will be false as the booking would not have been completed
329 NewBooking.HasBeenInvoiced = False
330
331 'Confirmation that the details are correct
332 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
333     Exit Sub 'Exit if user clicks cancel
334 End If
335
336 'Saving data to a textfile called "Bookings"
337 Dim SaveNewBooking As System.IO.StreamWriter
338 SaveNewBooking = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Bookings.txt"), True)
339 SaveNewBooking.WriteLine(NewBooking.BookingID & "," & NewBooking.Jobtype & "," & NewBooking.BookingDate & "," & NewBooking.CustomerID & "," & NewBooking.StaffID & "," & NewBooking.AllocatedTime & "," & NewBooking.Price & "," & NewBooking.ToolsList & "," & NewBooking.JobNotes & "," & NewBooking.HasBeenInvoiced)
340
341 'Telling the user that the generated ID is what will be used
342 MsgBox("Please note that if you have manually changed the booking ID then it will not be saved and the previously generated ID will be saved" + vbCrLf + vbCrLf + "Also note that ID may have been altered automatically")
343 MsgBox("The ID that was generated is: " & NewBooking.BookingID + vbCrLf + "Booking has been created")
344
345 Load_Bookings() 'Load new booking into AllBookings array
346 Insertion_Sort_Bookings() 'Sort the bookings according to date
347 Save_All_Bookings() 'Save sorted into the bookings textfile
348
349 butClear_Click()
350
351
352 End Sub

```

353

```
1 reference
354 Private Sub butClear_Click() Handles butClear.Click
355
356     'Clearing dropdown lists and resetting textboxes
357     txtBookingID.Text = ""
358     ddlJobType.Text = ""
359     calendarBookDate.Value = Today
360     ddlCustID.Text = ""
361     ddlStaffID.Text = ""
362     txtAllocatedHours.Text = ""
363     txtPrice.Text = ""
364     rtxtNotes.Text = ""
365     rtxtTools.Text = ""
366     txtFnameCust.Text = ""
367     txtSnameCust.Text = ""
368     txtFAddress.Text = ""
369     txtSAddress.Text = ""
370     txtPostcode.Text = ""
371     txtTown.Text = ""
372     txtFnameStaff.Text = ""
373     txtSnameStaff.Text = ""
374
375     'Loading customers into the customer ddl
376     ddlCustID.Items.Clear()
377     For x = 0 To NumCustomers - 1
378         ddlCustID.Items.Add(AllCustomers(x).CustomerID)
379     Next
380
381     'Loading equipment into the tools ddl
382     ddlTools.Items.Clear()
383     For x = 0 To NumEquipment - 1
384         ddlTools.Items.Add(AllEquipment(x).ItemName)
385     Next
386
387     'Only need to clear staff ddl as they're loaded when job type is selected
388     ddlStaffID.Items.Clear()
---
```

```
389
390     'Reset variables
391     ToolPosition = 0
392     NewBooking.ToolsList = ""
393
394     'Clears Tool array
395     For x = 0 To ToolPosition
396         ToolArray(x) = ""
397     Next
398
399 End Sub
400
```

```
0 references
401 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
402
403     Me.Close()
404
405 End Sub
406
```

```
0 references
407 Private Sub butSearchCustomers_Click(sender As Object, e As EventArgs) Handles butSearchCustomers.Click
408
409     'Open form for searching customers
410     SearchCust.Show()
411
412 End Sub
413
414 End Class
```

Form Name:

## Quick Stock Update

Item Name	Equipment ID	Amount in stock	Last Test Date	Next Test Date
<input type="text"/>	<input type="text"/>	<input type="text"/>	23 April 2022 <input type="text"/>	23 April 2022 <input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	23 April 2022 <input type="text"/>	23 April 2022 <input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	23 April 2022 <input type="text"/>	23 April 2022 <input type="text"/>

Code Listing:

```
1 reference
1 Public Class QuickStockUpdate
2
3     'Variables to hold the position of items from the array
4     Public Position1 As Integer
5     Public Position2 As Integer
6     Public Position3 As Integer
7
8     'Variables to hold whether the ddls have been changed
9     Public ChangedItem1 As Boolean = False
10    Public ChangedItem2 As Boolean = False
11    Public ChangedItem3 As Boolean = False
12
13    0 references
14    Private Sub QuickStockUpdate_Load(sender As Object, e As EventArgs) Handles MyBase.Load
15
16        'Load equipment
17        Load_Equipment()
18
19        'Clears dropdown lists incase there was anything in there before
20        ddlName1.Items.Clear()
21        ddlName2.Items.Clear()
22        ddlName3.Items.Clear()
23
24        'Load the IDs from the AllEquipment array into the ID drop down lists
25        For x = 0 To NumEquipment - 1
26            ddlName1.Items.Add(AllEquipment(x).ItemName)
27            ddlName2.Items.Add(AllEquipment(x).ItemName)
28            ddlName3.Items.Add(AllEquipment(x).ItemName)
29        Next
30
31        'Set the calendars as today's date for the purpose of continuity
32        calendarLast1.Value = Today
33        calendarLast2.Value = Today
34        calendarLast3.Value = Today
35        calendarNext1.Value = Today
36        calendarNext2.Value = Today
37        calendarNext3.Value = Today
38    End Sub
```

```

0 references
40 Private Sub ddlName1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlName1.SelectedIndexChanged
41
42     'Checks if ddlItem1 has been changed before, if so then the item entered would've been removed from the other ddls
43     'This code puts the removed item back, as ddlItem1 has been changed again
44     If ChangedItem1 = True Then
45         ddlName2.Items.Clear()
46         ddlName3.Items.Clear()
47         For x = 0 To NumEquipment - 1
48             ddlName2.Items.Add(AllEquipment(x).ItemName)
49             ddlName3.Items.Add(AllEquipment(x).ItemName)
50         Next
51     End If
52
53     'Code to check if Item 2 has been entered, if so then remove it from ddlItem3
54     If ChangedItem2 = True Then
55         ddlName3.Items.Remove(AllEquipment(Position2).ItemName)
56     End If
57
58     'Code to check if Item 3 has been entered, if so then remove it from ddlItem2
59     If ChangedItem3 = True Then
60         ddlName2.Items.Remove(AllEquipment(Position3).ItemName)
61     End If
62
63     For x = 0 To NumEquipment - 1
64         'Goes through the AllEquipment array and compares the IDs with the one entered within drop down list 1
65         If AllEquipment(x).ItemName = ddlName1.Text Then
66             'If IDs are the same then load the data for that ID into the textboxes
67             txtID1.Text = AllEquipment(x).EquipmentID
68             txtAmount1.Text = AllEquipment(x).AmountInStock
69             calendarLast1.Value = AllEquipment(x).LTestDate
70             calendarNext1.Value = AllEquipment(x).NTestDate
71             Position1 = x
72             ddlName2.Items.Remove(AllEquipment(x).ItemName) 'Removes the item in dd11 from the other ddls
73             ddlName3.Items.Remove(AllEquipment(x).ItemName)
74             ChangedItem1 = True 'Says that the item 1 in the ddls has been removed
75             Exit Sub 'Exits subroutine once equipment has been found
76         End If
77     Next
78
79 End Sub

```

```

80
0 references
81 Private Sub ddlName2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlName2.SelectedIndexChanged
82
83     'Checks if ddlItem2 has been changed before, if so then the item entered would've been removed from the other ddls
84     'This code puts the removed item back, as ddlItem2 has been changed again
85     If ChangedItem2 = True Then
86         ddlName1.Items.Clear()
87         ddlName3.Items.Clear()
88         For x = 0 To NumEquipment - 1
89             ddlName1.Items.Add(AllEquipment(x).ItemName)
90             ddlName3.Items.Add(AllEquipment(x).ItemName)
91         Next
92     End If
93
94     'Code to check if Item 1 has been entered, if so then remove it from ddlItem3
95     If ChangedItem1 = True Then
96         ddlName3.Items.Remove(AllEquipment(Position1).ItemName)
97     End If
98
99     'Code to check if Item 3 has been entered, if so then remove it from ddlItem1
100    If ChangedItem3 = True Then
101        ddlName1.Items.Remove(AllEquipment(Position3).ItemName)
102    End If
103
104    For x = 0 To NumEquipment - 1
105        'Goes through the AllEquipment array and compares the IDs with the one entered within drop down list 2
106        If AllEquipment(x).ItemName = ddlName2.Text Then
107            'If IDs are the same then load the data for that ID into the textboxes
108            txtID2.Text = AllEquipment(x).EquipmentID
109            txtAmount2.Text = AllEquipment(x).AmountInStock
110            calendarLast2.Value = AllEquipment(x).LTestDate
111            calendarNext2.Value = AllEquipment(x).NTestDate
112            Position2 = x
113            ddlName1.Items.Remove(AllEquipment(x).ItemName) 'Removes the item in dd12 from the other ddls
114            ddlName3.Items.Remove(AllEquipment(x).ItemName)
115            ChangedItem2 = True 'Says that the item 2 in the ddls has been removed
116            Exit Sub 'Exits subroutine once equipment has been found
117        End If
118    Next

```

```

119
120 End Sub
121
122 0 references
123 Private Sub ddlName3_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlName3.SelectedIndexChanged
124     'Checks if ddlItem3 has been changed before, if so then the item entered would've been removed from the other ddls
125     'This code puts the removed item back, as ddlItem3 has been changed again
126     If ChangedItem3 = True Then
127         ddlName1.Items.Clear()
128         ddlName2.Items.Clear()
129         For x = 0 To NumEquipment - 1
130             ddlName1.Items.Add(AllEquipment(x).ItemName)
131             ddlName2.Items.Add(AllEquipment(x).ItemName)
132         Next
133     End If
134
135     'Code to check if Item 1 has been entered, if so then remove it from ddlItem2
136     If ChangedItem1 = True Then
137         ddlName2.Items.Remove(AllEquipment(Position1).ItemName)
138     End If
139
140     'Code to check if Item 2 has been entered, if so then remove it from ddlItem1
141     If ChangedItem2 = True Then
142         ddlName1.Items.Remove(AllEquipment(Position2).ItemName)
143     End If
144
145     For x = 0 To NumEquipment - 1
146         'Goes through the AllEquipment array and compares the IDs with the one entered within drop down list 3
147         If AllEquipment(x).ItemName = ddlName3.Text Then
148             'If IDs are the same then load the data for that ID into the textboxes
149             txtID3.Text = AllEquipment(x).EquipmentID
150             txtAmount3.Text = AllEquipment(x).AmountInStock
151             calendarLast3.Value = AllEquipment(x).LTestDate
152             calendarNext3.Value = AllEquipment(x).NTestDate
153             Position3 = x
154             ddlName1.Items.Remove(AllEquipment(x).ItemName) 'Removes the item in dd3 from the other ddls
155             ddlName2.Items.Remove(AllEquipment(x).ItemName)
156             ChangedItem3 = True 'Says that the item 3 in the ddls has been removed
157             Exit Sub 'Exits subroutine once equipment has been found
158         End If
159     Next

```

```

160 End Sub
161
162 0 references
163 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
164     '*****ITEM 1*****
165
166     If ddlName1.Text <> "" Then 'If dd11 does not equal nothing
167
168         If txtAmount1.Text <> "" Then 'Check that amount1 doesn't equal nothing
169             Try 'Tries to do the statement below
170                 AllEquipment(Position1).AmountInStock = txtAmount1.Text
171             Catch 'If that does not work, it displays message and exits sub
172                 MsgBox("Values in 'Amount in stock' must be integers")
173                 Exit Sub
174             End Try
175         End If
176     End If
177
178     If calendarLast1.Value > Today Then 'Check that last test date is less than or equal to today
179         MsgBox("Last Test Date 1 must be less than or equal to today")
180         Exit Sub
181     End If
182
183     If calendarNext1.Value < Today.AddDays(1) Then 'Check that next date is more than today
184         MsgBox("Next Test Date 1 must be after today")
185         Exit Sub
186     End If
187
188     'Sets the entered data from the textboxes into the array for the equipment item in dd11
189     AllEquipment(Position1).AmountInStock = txtAmount1.Text
190     AllEquipment(Position1).LTestDate = calendarLast1.Value
191     AllEquipment(Position1).NTestDate = calendarNext1.Value
192
193 End If
194
195     '*****ITEM 2*****
196
197     If ddlName2.Text <> "" Then 'If dd12 does not equal nothing

```

```

198
199     If txtAmount2.Text <> "" Then 'Check that amount2 doesn't equal nothing
200         Try
201             AllEquipment(Position2).AmountInStock = txtAmount2.Text
202         Catch
203             MsgBox("Values in 'Amount in stock' must be integers")
204             Exit Sub
205         End Try
206     End If
207
208     If calendarLast2.Value > Today Then 'Check that last test date is less than or equal to today
209         MsgBox("Last Test Date 2 must be less than or equal to today")
210         Exit Sub
211     End If
212
213     If calendarNext2.Value < Today.AddDays(1) Then 'Check that next date is more than today
214         MsgBox("Next Test Date 2 must be after today")
215         Exit Sub
216     End If
217
218     'Sets the entered data from the textboxes into the array for the equipment item in ddl1
219     AllEquipment(Position2).AmountInStock = txtAmount3.Text
220     AllEquipment(Position2).LTestDate = calendarLast3.Value
221     AllEquipment(Position2).NTestDate = calendarNext3.Value
222
223 End If
224
225 *****ITEM 3*****
226
227 If ddlName3.Text <> "" Then 'If ddl2 does not equal nothing
228
229     If txtAmount3.Text <> "" Then 'Check that amount3 doesn't equal nothing
230         Try
231             AllEquipment(Position3).AmountInStock = txtAmount3.Text
232         Catch
233             MsgBox("Values in 'Amount in stock' must be integers")
234             Exit Sub
235         End Try
236     End If
237
238     If calendarLast3.Value > Today Then 'Check that last test date is less than or equal to today
239         MsgBox("Last Test Date 3 must be less than or equal to today")
240         Exit Sub
241     End If
242
243     If calendarNext3.Value < Today.AddDays(1) Then 'Check that next date is more than today
244         MsgBox("Next Test Date 3 must be after today")
245         Exit Sub
246     End If
247
248     'Sets the entered data from the textboxes into the array for the equipment item in ddl1
249     AllEquipment(Position3).AmountInStock = txtAmount3.Text
250     AllEquipment(Position3).LTestDate = calendarLast3.Value
251     AllEquipment(Position3).NTestDate = calendarNext3.Value
252
253 End If
254
255 'Confirmation that the details are correct
256 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
257     Exit Sub 'Exit if user clicks cancel
258 End If
259
260 MsgBox("All amounts and test dates saved (Please note that any changes to the ID will not be saved)")
261 Save_All_Equipment() 'Save the new data to textfile
262
263 End Sub
264
265 0 references
266 Private Sub butCancel_Click(sender As Object, e As EventArgs) Handles butCancel.Click
267     'Closes form
268     Me.Close()
269 End Sub
270 End Class
271

```



Form Name:

### Searching a Booking

**Please fill these options to load the corresponding data**

Booking ID

Customer ID

Booking Date

**Customer Details (cannot edit these here)**

First Name

Surname

Address

**Staff Details (cannot edit these here)**

First Name

Surname

**Booking Details (editable)**

Job Type

Date

Staff ID

Allocated Time (hrs)

Price (£)

Invoiced?  (This cannot be edited)

Tools for job

Job Specific Notes

Code Listing:

```

1 1 reference
2  Public Class SearchBooking
3
4      Public BookingIDChanged As Boolean = False 'Variable to say if the booking ID has been changed
5      Public Position As Integer
6      Public StaffFound As Boolean 'Variable to check if a staff member within the job type has been found
7      Public SameBookingDateCount As Integer 'Variable to count how many bookings there are with the same date
8
9      Public ToolArray(0 To 99) 'Array for the job tools to be kept
10     Public ToolPosition As Integer = 0
11
12 1 reference
13  Private Sub SearchBooking_Load() Handles MyBase.Load
14
15     'Load corresponding textfiles
16     Load_Bookings()
17     Load_Customers()
18     Load_Staff()
19     Load_Equipment()
20
21     'Clear ddl incase there was something there before
22     ddlBookingID.Items.Clear()
23     ddlCustomerID.Items.Clear()
24     ddlBookingDate.Items.Clear()
25     ddlJobType.Items.Clear()
26     ddlStaffID.Items.Clear()
27     ddlTools.Items.Clear()
28
29     'Loads all the bookings IDs into the ddl
30     For x = 0 To NumBookings - 1
31         ddlBookingID.Items.Add(AllBookings(x).BookingID)
32     Next
33
34     'Loads all the customers IDs into the ddl
35     'Doesn't only load customers who have placed a booking as the program will simply display no bookings for a customer with no bookings
36     For x = 0 To NumCustomers - 1
37         ddlCustomerID.Items.Add(AllCustomers(x).CustomerID)
38     Next
39
40     'Loads job types into ddl
41     ddlJobType.Items.Add("Welding")
42     ddlJobType.Items.Add("Woodwork")
43     ddlJobType.Items.Add("Plumbing")

```



```

80
81 'Clears the non-editable staff details
82 txtFnameStaff.Text = ""
83 txtSnameStaff.Text = ""
84
85 End Function
86
2 references
87 Private Sub ddlBookingID_SelectedIndexChanged() Handles ddlBookingID.SelectedIndexChanged
88
89 'Variable to temporarily hold the booking date so can use it after the booking date ddl has been cleared
90 Dim TempBookingDate As Date
91
92 'Variable to temporarily hold the customerID as to not call "ddlCustomerID_SelectedIndexChanged()" until after loop
93 Dim TempCustomerID As String
94
95 'Loop through booking file and load all the booking data with the ddl ID
96 For x = 0 To NumBookings - 1
97     If AllBookings(x).BookingID = ddlBookingID.Text Then
98
99         BookingIDChanged = True 'Say that the booking ID has been changed so msgbox("enter booking date") doesn't display
100
101         'Loads data into the editable textboxes
102         ddlJobType.Text = AllBookings(x).JobType
103         calendarBookDate.Value = AllBookings(x).BookingDate
104         TempBookingDate = AllBookings(x).BookingDate
105         TempCustomerID = AllBookings(x).CustomerID
106         ddlStaffID.Text = AllBookings(x).StaffID
107         txtAllocatedHours.Text = AllBookings(x).AllocatedTime
108         txtPrice.Text = AllBookings(x).Price.Substring(1, Len(AllBookings(x).Price) - 1) 'Removes £ from price
109         rtxtTools.Text = AllBookings(x).ToolsList
110         rtxtNotes.Text = AllBookings(x).JobNotes
111         txtHasBeenInvoiced.Text = AllBookings(x).HasBeenInvoiced
112         Position = x
113
114     End If
115 Next
116
117 'Sets the ddl texts as the temporarily held variable
118 ddlCustomerID.Text = TempCustomerID
119 ddlBookingDate.Text = TempBookingDate

```

```

120
121 'Loads data into the non-editable customer details
122 For y = 0 To NumCustomers - 1
123     If ddlCustomerID.Text = AllCustomers(y).CustomerID Then
124         txtFnameCust.Text = AllCustomers(y).Fname
125         txtSnameCust.Text = AllCustomers(y).Sname
126         rtxtAddressCust.Text = AllCustomers(y).FAddress + vbCrLf
127         rtxtAddressCust.Text = rtxtAddressCust.Text + AllCustomers(y).SAddress + vbCrLf
128         rtxtAddressCust.Text = rtxtAddressCust.Text + AllCustomers(y).Postcode + vbCrLf
129         rtxtAddressCust.Text = rtxtAddressCust.Text + AllCustomers(y).Town + vbCrLf
130     End If
131 Next
132
133 'Loads data into the non-editable staff details
134 For y = 0 To NumStaff - 1
135     If ddlCustomerID.Text = AllCustomers(y).CustomerID Then
136         txtFnameStaff.Text = AllStaff(y).Fname
137         txtSnameStaff.Text = AllStaff(y).Sname
138     End If
139 Next
140
141 End Sub
142
0 references
143 Private Sub butSearchCust_Click(sender As Object, e As EventArgs) Handles butSearchCust.Click
144
145     'Show searching customers form
146     SearchCust.Show()
147
148 End Sub
149
0 references
150 Private Sub ddlCustomerID_SelectedIndexChanged() Handles ddlCustomerID.SelectedIndexChanged
151
152     '*****If BookingIDChanged = False *****
153     If BookingIDChanged = False Then
154
155         'Clear booking date ddl
156         ddlBookingDate.Items.Clear()
157         ddlBookingDate.Text = ""

```

```

158
159     ddlBookingID.Text = "" 'Clear booking ID textbox
160     Clear_All_Textboxes() 'Clear all textboxes
161
162     'Loop through booking file and load all the booking data with the customer ID
163     For x = 0 To NumBookings - 1
164         If AllBookings(x).CustomerID = ddlCustomerID.Text Then
165             'Load the possible booking dates into booking date ddl
166             ddlBookingDate.Items.Add(AllBookings(x).BookingDate)
167         End If
168     Next
169
170     MsgBox("Please select a booking date in the dropdown list below")
171     Exit Sub
172 End If
173
174
175 *****If BookingIDChanged = True *****
176
177 'Clear booking date ddl (without clearing the text)
178 ddlBookingDate.Items.Clear()
179
180 'Loop through booking file and load all the booking data with the customer ID
181 For x = 0 To NumBookings - 1
182     If AllBookings(x).CustomerID = ddlCustomerID.Text Then
183         'Load the possible booking dates into booking date ddl
184         ddlBookingDate.Items.Add(AllBookings(x).BookingDate)
185     End If
186 Next
187
188 End Sub
189
190
191 0 references
192 Private Sub ddlBookingDate_SelectedIndexChanged() Handles ddlBookingDate.SelectedIndexChanged
193     'If BookingIDChanged = True then there's no point in searching for the BookingID
194     If BookingIDChanged = False Then
195         'Loop through booking file and load all the booking data with the customer ID
196

```

```

197     For x = 0 To NumBookings - 1
198     If AllBookings(x).BookingDate = ddlBookingDate.Text Then
199
200         'Set booking ID ddl to corresponding booking ID
201         ddlBookingID.Text = AllBookings(x).BookingID
202
203         ddlBookingID_SelectedIndexChanged() 'Sets off the called private sub
204
205     End If
206 Next
207 End If
208
209 'Resets BookingIDChanged
210 BookingIDChanged = False
211
212 End Sub
213

```

0 references

```

214 Private Sub ddlJobType_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlJobType.SelectedIndexChanged
215
216     'If job type changed, clear staff ID and name
217     ddlStaffID.Text = ""
218     txtFnameStaff.Text = ""
219     txtSnameStaff.Text = ""
220
221     StaffFound = False 'Variable to check if staff member has been found with appropriate speciality
222
223     'Clear all the staff beforehand so there is no data duplication
224     ddlStaffID.Items.Clear()
225
226     'Load in all the staff members with the same speciality as the job type
227     For x = 0 To NumStaff - 1
228     If ddlJobType.Text = AllStaff(x).Speciality Then
229         ddlStaffID.Items.Add(AllStaff(x).StaffID)
230         StaffFound = True
231     End If
232 Next
233
234     'Leave all staff members available if "Other" job type selected
235     If ddlJobType.Text = "Other" Then
236         ddlStaffID.Items.Clear()

```

```

237     For x = 0 To NumStaff - 1
238         ddlStaffID.Items.Add(AllStaff(x).StaffID)
239     Next
240 End If
241
242 'If no staff members have that speciality, it loads all the staff members
243 If StaffFound = False Then
244     ddlStaffID.Items.Clear()
245     For x = 0 To NumStaff - 1
246         ddlStaffID.Items.Add(AllStaff(x).StaffID)
247     Next
248 End If
249
250 'Doesn't remove admin users as they also have a speciality
251
252 End Sub
253

```

```

0 references
254 Private Sub ddlStaffID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlStaffID.SelectedIndexChanged
255
256     'Showing staff's name in textboxes
257     'This data is not saved in the booking textfile
258     For x = 0 To NumStaff - 1
259         If ddlStaffID.Text = AllStaff(x).StaffID Then
260             txtFnameStaff.Text = AllStaff(x).Fname
261             txtSnameStaff.Text = AllStaff(x).Sname
262         End If
263     Next
264
265 End Sub
266

```

```

0 references
267 Private Sub butAddTool_Click(sender As Object, e As EventArgs) Handles butAddTool.Click
268
269     'Checking if tools ddl is empty
270     If ddlTools.Text = "" Then
271         MsgBox("Please add a tool from the drop down list")
272     Exit Sub
273 End If
274

```

```

275 'Check if the first tool has been added
276 If ToolPosition = 0 Then
277     'Saving new tool to tools textbox without the "+" at the start
278     AllBookings(Position).ToolsList = AllBookings(Position).ToolsList & ddlTools.Text 'Adds tool in ddl to NewBooking array
279     rtxtTools.Text = AllBookings(Position).ToolsList 'Display in textbox
280     ToolArray(ToolPosition) = ddlTools.Text 'Enters tool into first slot of the ToolArray
281     ToolPosition = ToolPosition + 1
282     ddlTools.Text = "" 'Clear tool ddl
283 Exit Sub
284 End If
285
286 'Saving new tool to tools variable with the "+" at the start
287 AllBookings(Position).ToolsList = AllBookings(Position).ToolsList & " + " & ddlTools.Text
288 rtxtTools.Text = AllBookings(Position).ToolsList 'Display in textbox
289 ToolArray(ToolPosition) = " + " & ddlTools.Text 'Enters tool into next slot of the array
290 ToolPosition = ToolPosition + 1 'Increments tool position
291 ddlTools.Text = "" 'Clear tool ddl
292
293 End Sub
294

```

```

0 references
295 Private Sub butRemoveTool_Click(sender As Object, e As EventArgs) Handles butRemoveTool.Click
296
297     'Checks if nothing is in the textbox
298     If ToolPosition < 0 Then
299         ToolPosition = 0 'Resets tool position
300     End If
301
302     'Check if a tool has been entered first
303     If ToolPosition = 0 Then
304         MsgBox("Please add a tool from the drop down list")
305     Exit Sub
306 End If
307
308 AllBookings(Position).ToolsList = "" 'Clears tools list from NewBooking array
309
310 For x = 0 To ToolPosition - 2 'Goes to "position - 2" because position is set as 1 ahead of the last item
311     AllBookings(Position).ToolsList = AllBookings(Position).ToolsList & ToolArray(x) 'Load tools from ToolArray into tools list in NewBooking array
312 Next
313

```

```
313
314 ToolPosition = ToolPosition - 1 'Decrements tool position as one has been removed
315 rtxtTools.Text = AllBookings(Position).ToolsList 'Display in tools textbox
316
317 End Sub
318
```

0 references

```
319 Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
320
321 'If booking has been invoiced and therefore completed, the details cannot be edited
322 If txtHasBeenInvoiced.Text = "True" Then
323     MsgBox("As booking has been invoiced and therefore completed, its details cannot be edited")
324     Exit Sub
325 End If
326
327 'Booking ID cannot be edited
328
329 AllBookings(Position).JobType = ddlJobType.Text
330
331 'Checking that the booking date is either today or onwards
332 If calendarBookDate.Value < Now.AddHours(2) Then
333     MsgBox("Please select a booking date at least 2 hours from now")
334     Exit Sub
335 End If
336
337 'Checking that the allocated hours and start time allow the booking to finish at 6pm at the latest
338 Dim BookingTimeCheck As Integer = 18
339 Dim FinishTime = calendarBookDate.Value.Hour + Int(txtAllocatedHours.Text) 'Calculating what time job finishes
340 If FinishTime > BookingTimeCheck Then
341     MsgBox("Please either select a different time or reduce the amount of hours allocated (latest a worker will finish is 6pm)")
342     Exit Sub
343 End If
344
345 'Checking that job doesn't start too early
346 If calendarBookDate.Value.Hour < 8 Then
347     MsgBox("Job cannot start before 8 am, please try again")
348     Exit Sub
349 End If
350
351 AllBookings(Position).BookingDate = calendarBookDate.Value
352
```

```

353 'Presence check of customer and staff ID
354 If ddlStaffID.Text = "" Then
355     MsgBox("Please enter a staff for the booking")
356     Exit Sub
357 End If
358
359 AllBookings(Position).StaffID = ddlStaffID.Text
360
361 'Range check on time
362 If txtAllocatedHours.Text < 0.5 Or txtAllocatedHours.Text > 10 Then
363     MsgBox("Allocated time must be between 0.5 and 10 hrs, please try again")
364     Exit Sub
365 End If
366
367 'Integer/double check on allocated time
368 Try
369     AllBookings(Position).AllocatedTime = txtAllocatedHours.Text
370 Catch
371     MsgBox("Allocated time must be an integer/decimal point value")
372     Exit Sub
373 End Try
374
375 'Checking that the price does not include ",", or "£"
376 Dim PriceCommaCheck() As Char = txtPrice.Text.ToCharArray
377 For x = 0 To Len(txtPrice.Text) - 1
378     If PriceCommaCheck(x) = "," Or PriceCommaCheck(x) = "£" Then
379         MsgBox("The price must not include a comma or '£', please try again")
380         Exit Sub
381     End If
382 Next
383
384 'Integer/double check on price
385 Try
386     AllBookings(Position).Price = "£" & txtPrice.Text
387 Catch
388     MsgBox("Price must be an integer/decimal point value")
389     Exit Sub
390 End Try
391
392 'If nothing entered into tools list, check that this is correct
393 If rtxtTools.Text = "" Then
394
395     'Confirm with user whether tools list is meant to be empty
396     If MsgBox("Nothing was entered into tools." + vbCrLf + "Click 'OK' if this is correct.", 1, "Tools error") = vbCancel Then
397         Exit Sub
398     End If
399     AllBookings(Position).ToolsList = rtxtTools.Text
400
401 'Checking that the notes do not include ","
402 Dim NotesCommaCheck() As Char = rtxtNotes.Text.ToCharArray
403 For x = 0 To Len(rtxtNotes.Text) - 1
404     If NotesCommaCheck(x) = "," Then
405         MsgBox("Your notes must not include a comma, please try again")
406         Exit Sub
407     End If
408 Next
409 AllBookings(Position).JobNotes = rtxtNotes.Text
410
411 ''Has been invoiced" will be false as the booking would not have been completed
412 AllBookings(Position).HasBeenInvoiced = False
413
414 'Confirmation that the details are correct
415 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
416     Exit Sub 'Exit if user clicks cancel
417 End If
418
419 'Insertion_sort_bookings() 'Sort bookings into date order
420 Save_All_Bookings() 'Saves the updated booking data
421
422 SearchBooking_Load() 'If booking ID changed, then it is loaded as an option into the booking ID ddl
423
424 MsgBox("Booking data has been edited" + vbCrLf + "Your booking ID is: " & AllBookings(Position).BookingID)
425
426 'If booking ID then load it into booking ID ddl
427 ddlBookingID.Text = AllBookings(Position).BookingID
428 ddlBookingID_SelectedIndexChanged() 'This will load the corresponding data to this booking
429
430 End Sub
431

```

```

432 0 references
433 Private Sub ...butCancelBooking_Click(sender As Object, e As EventArgs) Handles butCancelBooking.Click
434
435     'Confirmation that the details are correct
436     If MsgBox("Please confirm that you wish to cancel this booking by clicking OK", 1, "Confirmation") = vbCancel Then
437         Exit Sub "Exit if user clicks cancel"
438     End If
439
440     'Take the position and replace the data with the data in the next position until end of array reached
441     For x = Position To NumBookings - 2 '-2 because we cannot replace the last variable with a non-existing one
442         AllBookings(x).BookingID = AllBookings(x + 1).BookingID
443         AllBookings(x).JobType = AllBookings(x + 1).JobType
444         AllBookings(x).BookingDate = AllBookings(x + 1).BookingDate
445         AllBookings(x).CustomerID = AllBookings(x + 1).CustomerID
446         AllBookings(x).StaffID = AllBookings(x + 1).StaffID
447         AllBookings(x).AllocatedTime = AllBookings(x + 1).AllocatedTime
448         AllBookings(x).Price = AllBookings(x + 1).Price
449         AllBookings(x).ToolsList = AllBookings(x + 1).ToolsList
450         AllBookings(x).JobNotes = AllBookings(x + 1).JobNotes
451         AllBookings(x).HasBeenInvoiced = AllBookings(x + 1).HasBeenInvoiced
452     Next
453
454     'Decremate the number of bookings
455     'This is so that when SaveAllBookings is called, the bottom line is ignored
456     NumBookings = NumBookings - 1
457
458     'Saves the changes
459     Save_All_Bookings()
460     MsgBox("Booking cancelled")
461
462     'Close the form as this solves many problems
463     butQuit_Click()
464 End Sub
465
466 0 references
467 Private Sub butCancelEdit_Click(sender As Object, e As EventArgs) Handles butCancelEdit.Click
468
469     MsgBox("Changes have been reset")
470     butQuit_click()
471
472 End Sub
473
474 2 references
475 Private Sub ...butQuit_Click() Handles butQuit.Click
476
477     Me.Close()
478 End Sub
479 End Class

```

Form Name:

### Searching a Customer

**Please fill the drop down lists below to load the corresponding data**

Surname

Postcode

Customer ID	<input type="text"/>
Title	<input type="text"/>
First name	<input type="text"/>
Surname	<input type="text"/>
1st Address Line	<input type="text"/>
2nd Address Line	<input type="text"/>
Postcode	<input type="text"/>
Town/City	<input type="text"/>
Phone number	<input type="text"/>
Email	<input type="text"/>
DoB	<input type="text" value="01/01/1900"/>

Save EditCancel

Delete UserQuit

Code Listing:

```

1 reference
1 Public Class SearchCust
2
3     Public Position As Integer
4     Public NumFound As Integer = 0 'Variable for number of found customers
5
6     1 reference
7     Private Sub SearchCust_Load() Handles MyBase.Load
8         'Load all the customers
9         Load_Customers()
10
11         'Clears the drop down list in case the list was loaded before
12         ddlSname.Items.Clear()
13         ddlPostcode.Items.Clear()
14
15         'Load the surnames from the AllCustomers array into the surname drop down list
16         For x = 0 To NumCustomers - 1
17             ddlSname.Items.Add(AllCustomers(x).Sname)
18         Next
19
20     End Sub
21
22     0 references
23     Private Sub ddlSname_SelectedIndexChanged_1(sender As Object, e As EventArgs) Handles ddlSname.SelectedIndexChanged
24         ddlPostcode.Items.Clear() 'Clear postcode drop down list
25         NumFound = 0 'Resets NumFound
26
27         For x = 0 To NumCustomers - 1
28             'Goes through the AllCustomers array and compares the surnames with the one entered within the drop down list
29             If ddlSname.Text = AllCustomers(x).Sname Then
30                 'If surnames are the same then load the postcode for that surname into the postcode drop down list
31                 ddlPostcode.Items.Add(AllCustomers(x).Postcode)
32                 NumFound = NumFound + 1
33                 Position = x
34             End If
35         Next
36
37         If NumFound > 1 Then 'If more than 1 customer found, make user select from the postcode list below
38             'Clear textboxes so other data isn't displayed in the textboxes
39             txtCustID.Text = ""
40             txtTitle.Text = ""

```

```

41     txtFname.Text = ""
42     txtSname.Text = ""
43     txtFAddress.Text = ""
44     txtSAddress.Text = ""
45     txtPost.Text = ""
46     txtTown.Text = ""
47     txtNum.Text = ""
48     txtEmail.Text = ""
49     calendarDoB.Text = "01/01/1900"
50     MsgBox("Please select your postcode from the drop down list below")
51
52     ElseIf NumFound = 1 Then 'If only 1 customer found then there can only be 1 option
53         'Load all the data in for that one customer into textboxes
54         ddlPostcode.Text = AllCustomers(Position).Postcode
55         txtCustID.Text = AllCustomers(Position).CustomerID
56         txtTitle.Text = AllCustomers(Position).Title
57         txtFname.Text = AllCustomers(Position).Fname
58         txtSname.Text = AllCustomers(Position).Sname
59         txtFAddress.Text = AllCustomers(Position).FAddress
60         txtSAddress.Text = AllCustomers(Position).SAddress
61         txtPost.Text = AllCustomers(Position).Postcode
62         txtTown.Text = AllCustomers(Position).Town
63         txtNum.Text = AllCustomers(Position).Phone
64         txtEmail.Text = AllCustomers(Position).Email
65         calendarDoB.Text = AllCustomers(Position).DoB
66
67     End If
68
69 End Sub
70

```

```

71 0 references
72 Private Sub ddlPostcode_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlPostcode.SelectedIndexChanged
73
74     'Compares value within ddlPostcode to postcodes within the customers textfile
75     For x = 0 To NumCustomers - 1
76         If ddlPostcode.Text = AllCustomers(x).Postcode Then
77             txtCustID.Text = AllCustomers(x).CustomerID
78             txtTitle.Text = AllCustomers(x).Title
79             txtFname.Text = AllCustomers(x).Fname
80             txtSname.Text = AllCustomers(x).Sname

```

```

81             txtFAddress.Text = AllCustomers(x).FAddress
82             txtSAddress.Text = AllCustomers(x).SAddress
83             txtPost.Text = AllCustomers(x).Postcode
84             txtTown.Text = AllCustomers(x).Town
85             txtNum.Text = AllCustomers(x).Phone
86             txtEmail.Text = AllCustomers(x).Email
87             calendarDoB.Text = AllCustomers(x).DoB
88             Position = x
89         End If
90     Next
91 End Sub
92

```

```

93 0 references
94 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
95
96     "'Position" is from the ddlSname code
97     'CustomerID will not be changed
98
99     'Checks the drop down list isn't empty
100    If ddlSname.Text = "" Then
101        MsgBox("Please select an ID")
102        Exit Sub
103    End If
104
105    AllCustomers(Position).Title = txtTitle.Text
106    'List check on title
107    If AllCustomers(Position).Title <> "Mr" And AllCustomers(Position).Title <> "Mrs" And AllCustomers(Position).Title <> "Miss" And AllCustomers(Position).Title <> "Ms" A
108        MsgBox("Incorrect title entered, correct titles; Mr, Mrs, Miss, Ms, Mx, Sir, Dr, Cllr, Lord, Lady. Please try again")
109        Exit Sub
110    End If
111
112    'Correct capitalisation on first name and surname
113    AllCustomers(Position).Fname = txtFname.Text.Substring(0, 1).ToUpper & txtFname.Text.Substring(1, txtFname.Text.Length - 1).ToLower
114    AllCustomers(Position).Sname = txtsname.Text.Substring(0, 1).ToUpper & txtsname.Text.Substring(1, txtsname.Text.Length - 1).ToLower
115
116    'Entering textbox values into FAddress, SAddress, Postcode and Town
117    AllCustomers(Position).FAddress = txtFAddress.Text
118    AllCustomers(Position).SAddress = txtSAddress.Text
119    AllCustomers(Position).Postcode = txtPost.Text
120    AllCustomers(Position).Town = txtTown.Text

```

```

106 title <> "Ms" And AllCustomers(Position).Title <> "Mr" And AllCustomers(Position).Title <> "Sir" And AllCustomers(Position).Title <> "Dr" A
107
---
106 n).Title <> "Dr" And AllCustomers(Position).Title <> "Cllr" And AllCustomers(Position).Title <> "Lady" And AllCustomers(Position).Title <> "Lord" Then
107

119 AllCustomers(Position).Town = txtTown.Text
120
121 'Entering textbox value into Phone
122 AllCustomers(Position).Phone = txtNum.Text
123 'Length check on phone number
124 If Len(AllCustomers(Position).Phone) <> 11 Then
125     MsgBox("Incorrect phone number entered, must be 11 characters long, please try again")
126     Exit Sub
127 End If
128
129 'Entering textbox value into email
130 AllCustomers(Position).Email = txtEmail.Text
131 'Format check on email
132 Dim EmailFormat As Boolean = AllCustomers(Position).Email Like "*@*.*"
133 If EmailFormat = False Then
134     MsgBox("Incorrect email format detected, must be like __@__._ , please try again")
135     Exit Sub
136 End If
137
138 'Entering date picker value into Date of Birth
139 AllCustomers(Position).DoB = calendarDoB.Text
140
141 'Confirmation that the details are correct
142 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
143     Exit Sub 'Exit if user clicks cancel
144 End If
145
146 Save_All_Customers() 'Calls function to save the new customer data
147 SearchCust_Load() 'Loads form again so updated surname is in the surname ddl
148 ddlSurname.Text = txtSurname.Text 'If surname changed, puts new surname into the ddl text
149
150 MsgBox("Edited data has been saved (Please note you cannot edit the ID)")
151
152 End Sub
153
0 references
154 Private Sub butCancel_Click(sender As Object, e As EventArgs) Handles butCancel.Click
155
156     MsgBox("Changes have been reset")
157     butQuit_Click()
158

```

```

158
159 End Sub
160
161 Private Sub butDelete_Click(sender As Object, e As EventArgs) Handles butDelete.Click
162
163     'Confirmation that the details are correct
164     If MsgBox("Please confirm that you wish to delete this user by clicking OK", 1, "Confirmation") = vbCancel Then
165         Exit Sub 'Exit if user clicks cancel
166     End If
167
168     'Take the position and replace the data with the data in the next position until end of array reached
169     For x = Position To NumCustomers - 2 '-2 because we cannot replace the last variable with a non-existing one
170         AllCustomers(x).CustomerID = AllCustomers(x + 1).CustomerID
171         AllCustomers(x).Title = AllCustomers(x + 1).Title
172         AllCustomers(x).Fname = AllCustomers(x + 1).Fname
173         AllCustomers(x).Sname = AllCustomers(x + 1).Sname
174         AllCustomers(x).FAddress = AllCustomers(x + 1).FAddress
175         AllCustomers(x).SAddress = AllCustomers(x + 1).SAddress
176         AllCustomers(x).Postcode = AllCustomers(x + 1).Postcode
177         AllCustomers(x).Town = AllCustomers(x + 1).Town
178         AllCustomers(x).Phone = AllCustomers(x + 1).Phone
179         AllCustomers(x).Email = AllCustomers(x + 1).Email
180         AllCustomers(x).DoB = AllCustomers(x + 1).DoB
181     Next
182
183     'Decremate the number of customers
184     'This is so that when SaveAllCustomers is called, the bottom line is ignored
185     NumCustomers = NumCustomers - 1
186
187     'Saves the changes
188     Save_All_Customers()
189     MsgBox("Customer deleted")
190
191     'Close the form as this solves many problems
192     butQuit_Click()
193
194 End Sub
195

```

```

2 references
195
196 Private Sub butQuit_Click() Handles butQuit.Click
197
198     Me.Close()
199
200 End Sub
201 End Class

```

**Form Name:**

## Searching Equipment

Item Name	<input type="text"/>
Equipment ID	<input type="text"/>
Amount in stock	<input type="text"/>
Reorder stock level	<input type="text"/>
Reorder quantity	<input type="text"/>
Last test date	<input type="text" value="01/01/2020"/>
Next test date	<input type="text" value="01/01/2020"/>
Tested by	<input type="text"/>

Save EditCancelDeleteQuit

**Code Listing:**

```

1 reference
1 Public Class SearchEquipment
2
3     Public Position As Integer
4     Private Sub SearchEquipment_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5
6         'Load the equipment textfile
7         Load_Equipment()
8
9         'Clears the drop down list in case the list was loaded before
10        ddlItemName.Items.Clear()
11
12        'Load the IDs from the AllEquipment array into the ID drop down list
13        For x = 0 To NumEquipment - 1
14            ddlItemName.Items.Add(AllEquipment(x).ItemName)
15        Next
16
17    End Sub
18
19    Private Sub ddlEquipID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlItemName.SelectedIndexChanged
20
21        For x = 0 To NumEquipment - 1
22            'Goes through the AllEquipment array and compares the IDs with the one entered within the drop down list
23            If AllEquipment(x).ItemName = ddlItemName.Text Then
24                'If IDs are the same then load the data for that ID into the textboxes
25                txtID.Text = AllEquipment(x).EquipmentID
26                txtAmount.Text = AllEquipment(x).AmountInStock
27                txtReorderLevel.Text = AllEquipment(x).ReorderLevel
28                txtReorderQuan.Text = AllEquipment(x).ReorderAmount
29                calendarLTestDate.Value = AllEquipment(x).LTestDate
30                calendarNTestDate.Value = AllEquipment(x).NTestDate
31                txtTestedBy.Text = AllEquipment(x).TestedBy
32                Position = x
33                Exit Sub 'Exits subroutine once equipment has been found
34            End If
35        Next
36
37    End Sub

```

```

39 0 references
40 Private Sub calendarLTestDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarLTestDate.ValueChanged
41     'Sets Next Test Date equal to the Last Test Date value if N < L
42     If calendarNTestDate.Value < calendarLTestDate.Value Then
43         calendarNTestDate.Value = calendarLTestDate.Value
44     End If
45
46 End Sub
47
48 0 references
49 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
50     'Position is from ddlEquipID
51     'EquipmentID will not be changed
52     'Item name will not be changed
53
54     If ddlItemName.Text = "" Then
55         MsgBox("Please select an item")
56         Exit Sub
57     End If
58
59     'Integer check on amount in stock
60     Try
61         AllEquipment(Position).AmountInStock = txtAmount.Text
62     Catch
63         MsgBox("Amount in stock must be an integer value")
64         Exit Sub
65     End Try
66
67     'Range check on Amount in Stock
68     If AllEquipment(Position).AmountInStock > 1000 Or AllEquipment(Position).AmountInStock < 0 Then
69         MsgBox("The amount for this item is out of range, must be <1000 and >0, please try again")
70         Exit Sub
71     End If
72
73     'Integer check on reorder level and quantity
74     Try
75         AllEquipment(Position).ReorderLevel = txtReorderLevel.Text
76         AllEquipment(Position).ReorderAmount = txtReorderQuan.Text
77     Catch
78         MsgBox("Reorder level and quantity must be integer values")

```

```

79     Exit Sub
80 End Try
81
82 'Check that last test date is before or on today
83 If calendarLTestDate.Value > Now Then
84     MsgBox("Please select an appropriate last test date (i.e. not after today)")
85     Exit Sub
86 End If
87 AllEquipment(Position).LTestDate = calendarLTestDate.Value
88
89 'Check that next test date is on or after today and not equal to the last test date
90 If calendarNTestDate.Value < Now Or calendarNTestDate.Value = calendarLTestDate.Value Then
91     MsgBox("Please select an appropriate next test date (i.e. not today or before)")
92     Exit Sub
93 End If
94 AllEquipment(Position).NTestDate = calendarNTestDate.Value
95
96 AllEquipment(Position).TestedBy = txtTestedBy.Text
97
98 'Confirmation that the details are correct
99 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
100     Exit Sub 'Exit if user clicks cancel
101 End If
102
103 Save_All_Equipment() 'Calls function to save the new equipment data
104
105 MsgBox("Edited data has been saved (Please note you cannot edit the ID)")
106 MsgBox("The ID for the item is " & AllEquipment(Position).EquipmentID)
107
108 End Sub
109

```

```

110 0 references
111 Private Sub butCancel_Click(sender As Object, e As EventArgs) Handles butCancel.Click
112     If ddlItemName.Text = "" Then
113         MsgBox("Please select an item")
114         Exit Sub
115     End If
116

```

```

117 For x = 0 To NumEquipment - 1
118     'Goes through the AllCustomers array and compares the IDs with the one entered within the drop down list
119     If AllEquipment(x).ItemName = ddlItemName.Text Then
120         'If IDs are the same then load the data for that ID into the textboxes
121         txtID.Text = AllEquipment(x).EquipmentID
122         txtAmount.Text = AllEquipment(x).AmountInStock
123         txtReorderLevel.Text = AllEquipment(x).ReorderLevel
124         txtReorderQuan.Text = AllEquipment(x).ReorderAmount
125         calendarLTestDate.Value = AllEquipment(x).LTestDate
126         calendarNTestDate.Value = AllEquipment(x).NTestDate
127         txtTestedBy.Text = AllEquipment(x).TestedBy
128         Position = x
129         MsgBox("Changes have been reset")
130         Exit Sub 'Exits subroutine once customer has been found
131     End If
132 Next
133
134 End Sub
135

```

```

136 0 references
137 Private Sub butDelete_Click(sender As Object, e As EventArgs) Handles butDelete.Click
138     'Confirmation that the details are correct
139     If MsgBox("Please confirm that you wish to delete this piece of equipment by clicking OK", 1, "Confirmation") = vbCancel Then
140         Exit Sub 'Exit if user clicks cancel
141     End If
142
143     'Take the position and replace the data with the data in the next position until end of array reached
144     For x = Position To NumEquipment - 2 '-2 because we cannot replace the last variable with a non-existing one
145         AllEquipment(x).EquipmentID = AllEquipment(x + 1).EquipmentID
146         AllEquipment(x).ItemName = AllEquipment(x + 1).ItemName
147         AllEquipment(x).AmountInStock = AllEquipment(x + 1).AmountInStock
148         AllEquipment(x).ReorderLevel = AllEquipment(x + 1).ReorderLevel
149         AllEquipment(x).ReorderAmount = AllEquipment(x + 1).ReorderAmount
150         AllEquipment(x).LTestDate = AllEquipment(x + 1).LTestDate
151         AllEquipment(x).NTestDate = AllEquipment(x + 1).NTestDate
152         AllEquipment(x).TestedBy = AllEquipment(x + 1).TestedBy
153     Next
154
155     'Decrement the number of customers as one has been deleted
156     'This is so that when SaveAllEquipment is called, the bottom line is ignored
157     NumEquipment = NumEquipment - 1

```

```
158
159     'Saves the changes
160     Save_All_Equipment()
161     MsgBox("Equipment deleted")
162
163     'Close the form as this solves many problems
164     butQuit_Click()
165
166 End Sub
167
168 1 reference
169 Private Sub butQuit_Click() Handles butQuit.Click
170     Me.Close()
171
172 End Sub
173
174 End Class
```

**Form Name:**

### Searching an Invoice

**Please fill these options to load the corresponding data**

Invoice ID

Booking ID

**Customer Data**

Customer Name

Customer Address

**Invoice Data**

Invoice ID

Booking ID

Booking Date

Price

Paid

**Booking Data**

Job Notes

**Code Listing:**

```

1 reference
1 Public Class SearchInvoice
2
3 Private Sub SearchInvoice_Load(sender As Object, e As EventArgs) Handles MyBase.Load
4
5     'Load textfiles
6     Load_Customers()
7     Load_Invoices()
8     Load_Bookings()
9
10    'Clear dropdown lists incase there was something in them beforehand
11    ddlInvoiceID.Items.Clear()
12    ddlBookingID.Items.Clear()
13
14    'Load invoice IDs into ddl
15    For x = 0 To NumInvoices - 1
16        ddlInvoiceID.Items.Add(AllInvoices(x).InvoiceID)
17    Next
18
19    'Load booking IDs into ddl
20    For x = 0 To NumBookings - 1
21        ddlBookingID.Items.Add(AllBookings(x).BookingID)
22    Next
23
24 End Sub
25
26 Private Sub ddlInvoiceID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlInvoiceID.SelectedIndexChanged
27
28    'Loading invoice data into textboxes
29    For x = 0 To NumInvoices - 1
30        If AllInvoices(x).InvoiceID = ddlInvoiceID.Text Then
31            'Setting BookingID ddl to the bookingID corresponding to this invoice
32            ddlBookingID.Text = AllInvoices(x).BookingID
33
34            'Loading Invoice Details
35            txtInvoiceID.Text = AllInvoices(x).InvoiceID
36            txtBookingID.Text = AllInvoices(x).BookingID
37            txtBookingDate.Text = AllInvoices(x).BookingDate
38            txtPrice.Text = AllInvoices(x).Price
39            txtPaid.Text = AllInvoices(x).Paid
40

```

```

41 'Loading Booking Details
42 For y = 0 To NumBookings - 1
43     If AllInvoices(x).BookingID = AllBookings(y).BookingID Then
44         rtxtJobNotes.Text = AllBookings(y).JobNotes 'Displaying job notes
45
46         'Loading Customer Details
47         For z = 0 To NumCustomers - 1
48             If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
49                 txtCustName.Text = AllCustomers(z).Fname & " " & AllCustomers(z).Sname 'Displaying customer name
50
51                 'Displaying customer address
52                 rtxtCustAddress.Text = AllCustomers(z).FAddress + vbCrLf
53                 rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).SAddress + vbCrLf
54                 rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).Postcode + vbCrLf
55                 rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).Town + vbCrLf
56
57             End If
58         Next
59     End If
60 Next
61 End If
62 Next
63 End If
64 Next
65 End Sub
66
67

```

0 references

```

68 Private Sub ddlBookingID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlBookingID.SelectedIndexChanged
69
70     'Loading invoice data into textboxes
71     For x = 0 To NumInvoices - 1
72         If AllInvoices(x).BookingID = ddlBookingID.Text Then
73             'Setting BookingID ddl to the bookingID corresponding to this invoice
74             ddlInvoiceID.Text = AllInvoices(x).InvoiceID
75
76             'Loading Invoice Details
77             txtInvoiceID.Text = AllInvoices(x).InvoiceID
78             txtBookingID.Text = AllInvoices(x).BookingID
79             txtBookingDate.Text = AllInvoices(x).BookingDate
80
81         End If
82     Next
83 End Sub

```

```

80      txtPrice.Text = AllInvoices(x).Price
81      txtPaid.Text = AllInvoices(x).Paid
82
83      'Loading Booking Details
84      For y = 0 To NumBookings - 1
85          If AllInvoices(x).BookingID = AllBookings(y).BookingID Then
86              rtxtJobNotes.Text = AllBookings(y).JobNotes 'Displaying job notes
87
88              'Loading Customer Details
89              For z = 0 To NumCustomers - 1
90                  If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
91                      txtCustName.Text = AllCustomers(z).Fname & " " & AllCustomers(z).Sname 'Displaying customer name
92
93                      'Displaying customer address
94                      rtxtCustAddress.Text = AllCustomers(z).FAddress + vbCrLf
95                      rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).SAddress + vbCrLf
96                      rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).Postcode + vbCrLf
97                      rtxtCustAddress.Text = rtxtCustAddress.Text + AllCustomers(z).Town + vbCrLf
98
99                  End If
100             Next
101
102             End If
103         Next
104
105     End If
106 Next
107
108 End Sub
109
110 0 references
111 Private Sub butShowInvoice_Click(sender As Object, e As EventArgs) Handles butShowInvoice.Click
112     MsgBox("Opening invoice...")
113     Process.Start(Dir(txtInvoiceID.Text & ".txt")) 'Opening the printable invoice
114
115 End Sub
116
117 0 references
118 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
119     Me.Close()
120
121 End Sub
122
123 0 references
124 Private Sub butSearchBookings_Click(sender As Object, e As EventArgs) Handles butSearchBookings.Click
125     'Open form for searching bookings
126     SearchBooking.Show()
127
128 End Sub
129 End Class

```

Form Name:

## Searching a Monthly Pay Slip

**Please fill these options then click OPEN**

Staff ID	<input type="text"/>	<input type="button" value="Search Staff"/>
Month	<input type="text"/>	
Year	<input type="text"/>	

**Staff Details (cannot edit these here)**

First Name	<input type="text"/>	<input type="button" value="Open"/>
Surname	<input type="text"/>	<input type="button" value="Quit"/>

Code Listing:

```
1 reference
1 Public Class SearchMonthlyPayroll
2
3     Public SearchPayID As String 'Variable to hold the search for payroll ID
4
5     0 references
6     Private Sub SearchMonthlyPayroll_Load(sender As Object, e As EventArgs) Handles MyBase.Load
7
8         'Load appropriate textfiles
9         Load_Staff()
10
11         'Clear ddl incase something was loaded in previously
12         ddlStaffID.Items.Clear()
13         ddlMonth.Items.Clear()
14         ddlYear.Items.Clear()
15
16         'Load staff into ddl
17         For x = 0 To NumStaff - 1
18             ddlStaffID.Items.Add(AllStaff(x).StaffID)
19         Next
20
21         'Load months and years into ddls
22         ddlMonth.Items.Add("01")
23         ddlMonth.Items.Add("02")
24         ddlMonth.Items.Add("03")
25         ddlMonth.Items.Add("04")
26         ddlMonth.Items.Add("05")
27         ddlMonth.Items.Add("06")
28         ddlMonth.Items.Add("07")
29         ddlMonth.Items.Add("08")
30         ddlMonth.Items.Add("09")
31         ddlMonth.Items.Add("10")
32         ddlMonth.Items.Add("11")
33         ddlMonth.Items.Add("12")
34         ddlYear.Items.Add("2021")
35         ddlYear.Items.Add("2022")
36         ddlYear.Items.Add("2023")
37         ddlYear.Items.Add("2024")
38         ddlYear.Items.Add("2025")
39         ddlYear.Items.Add("2026")
40     End Sub
```

```

42 0 references
43 Private Sub butSearchStaff_Click(sender As Object, e As EventArgs) Handles butSearchStaff.Click
44     'Open form to search staff
45     SearchStaff.Show()
46
47 End Sub
48
49 0 references
50 Private Sub ddlStaffID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlStaffID.SelectedIndexChanged
51     'Load data from staff textfile into textboxes
52     For x = 0 To NumStaff - 1
53         If AllStaff(x).StaffID = ddlStaffID.Text Then
54             txtFnameStaff.Text = AllStaff(x).Fname
55             txtSnameStaff.Text = AllStaff(x).Sname
56         End If
57     Next
58
59 End Sub
60
61 0 references
62 Private Sub butOpen_Click(sender As Object, e As EventArgs) Handles butOpen.Click
63     'Presence check on ddl values
64     If ddlStaffID.Text = "" Or ddlMonth.Text = "" Or ddlYear.Text = "" Then
65         MsgBox("Please enter staffID, month, and year for the payroll slip you wish to search")
66         Exit Sub
67     End If
68
69     'Set search term (using ddl values) into the same format as monthly payroll slip IDs are formed
70     SearchPayID = ddlYear.Text.Substring(2, 2) & ddlMonth.Text & ddlStaffID.Text
71
72     Try 'Try loading file named as the monthly payroll
73         Dim file As System.IO.StreamReader
74         file = My.Computer.FileSystem.OpenTextFileReader(Dir$(SearchPayID & ".txt"))
75         MsgBox("Monthly payroll slip found" + vbCrLf + "Opening textfile...")
76         Process.Start(Dir(SearchPayID & ".txt")) 'Opening the printable payroll slip
77
78     Catch 'If doesn't work then individual payroll textfile must not exist
79         MsgBox("Monthly payroll textfile not found" + vbCrLf + "Please ensure search values are correct")
80     End Try
81
82 End Sub
83
84 0 references
85 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
86     'Close form
87     Me.Close()
88
89 End Sub
90 End Class

```

**Form Name:**

### Searching a Staff Member

**Please fill the drop down lists below to load the corresponding data**

Surname

Postcode

Staff ID <input type="text"/>	1st Address Line <input type="text"/>
Staff Type <input type="text"/>	2nd Address Line <input type="text"/>
Specialty <input type="text"/>	Postcode <input type="text"/>
Title <input type="text"/>	Town/city <input type="text"/>
First name <input type="text"/>	Phone number <input type="text"/>
Surname <input type="text"/>	Email <input type="text"/>
Password <input type="text"/>	DoB <input type="text" value="01/01/1900"/>
Clue <input type="text"/>	

**Code Listing:**

```
1 reference
1 Public Class SearchStaff
2
3     Public PositionPassword As Integer
4     Public PositionStaff As Integer
5
6     Public NumFound As Integer = 0
7
0 references
7 Private Sub SearchStaff_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8
9     Load_Staff() 'Load all the staff
10    Load_Password() 'Load all the passwords
11
12    'Clears the drop down list in case the list was loaded before
13    ddlSname.Items.Clear()
14    ddlPostcode.Items.Clear()
15
16    'Load the IDs from the AllStaff array into the ID drop down list
17    For x = 0 To NumStaff - 1
18        ddlSname.Items.Add(AllStaff(x).Sname)
19    Next
20
21    'Clears the speciality drop down list in case the form was loaded before
22    ddlSpecial.Items.Clear()
23
24    'Loading items into "Speciality" drop down list
25    ddlSpecial.Items.Add("Welding")
26    ddlSpecial.Items.Add("Woodwork")
27    ddlSpecial.Items.Add("Plumbing")
28    ddlSpecial.Items.Add("Carpentry")
29    ddlSpecial.Items.Add("Electrician")
30    ddlSpecial.Items.Add("Gardening")
31    ddlSpecial.Items.Add("Roofing")
32    ddlSpecial.Items.Add("Other")
33
34 End Sub
35
0 references
36 Private Sub ddlSname_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlSname.SelectedIndexChanged
37
38     Dim PasswordFound As Boolean = False 'Used when searching for a staff members password
39
40     ddlPostcode.Items.Clear() 'Clear postcode drop down list
```

```

41 NumFound = 0 'Resets NumFound
42
43 For x = 0 To NumStaff - 1
44     'Goes through the AllStaff array and compares the surnames with the one entered within the drop down list
45     If ddlSname.Text = AllStaff(x).Sname Then
46         'If surnames are the same then load the postcode for that surname into the postcode drop down list
47         ddlPostcode.Items.Add(AllStaff(x).Postcode)
48         NumFound = NumFound + 1
49         PositionStaff = x
50
51         'Find password data
52         For y = 0 To NumPasswords - 1
53             If AllStaff(x).StaffID = AllPasswords(y).StaffID Then
54                 PositionPassword = y
55                 PasswordFound = True
56             End If
57         Next
58
59     End If
60 Next
61
62 'Check if password was found
63 If PasswordFound = False Then
64     MsgBox("ERROR: Password data has not been found")
65 End If
66
67 If NumFound > 1 Then 'If more than 1 staff member found, make user select from the postcode list below
68     'Clear textboxes so other data isn't displayed in the textboxes
69     txtStaffID.Text = ""
70     txtStaffType.Text = ""
71     txtTitle.Text = ""
72     txtFname.Text = ""
73     txtSname.Text = ""
74     txtFAddress.Text = ""
75     txtSAddress.Text = ""
76     txtPost.Text = ""
77     txtTown.Text = ""
78     txtNum.Text = ""
79     txtEmail.Text = ""
80     calendarDoB.Text = "01/01/1900"

```

```

81     ddlSpecial.Text = ""
82     txtPassword.Text = ""
83     txtClue.Text = ""
84     ddlPostcode.Text = ""
85     MsgBox("Please select your postcode from the drop down list below")
86
87     ElseIf NumFound = 1 Then 'If only 1 customer found then there can only be 1 option
88
89         'Load all the data in for that one customer into textboxes
90         ddlPostcode.Text = AllStaff(PositionStaff).Postcode
91         txtStaffID.Text = AllStaff(PositionStaff).StaffID
92         txtTitle.Text = AllStaff(PositionStaff).Title
93         txtFname.Text = AllStaff(PositionStaff).Fname
94         txtSname.Text = AllStaff(PositionStaff).Sname
95         txtFAddress.Text = AllStaff(PositionStaff).FAddress
96         txtSAddress.Text = AllStaff(PositionStaff).SAddress
97         txtPost.Text = AllStaff(PositionStaff).Postcode
98         txtTown.Text = AllStaff(PositionStaff).Town
99         txtNum.Text = AllStaff(PositionStaff).Phone
100        txtEmail.Text = AllStaff(PositionStaff).Email
101        calendarDoB.Text = AllStaff(PositionStaff).DoB
102        ddlSpecial.Text = AllStaff(PositionStaff).Speciality
103
104        txtPassword.Text = AllPasswords(PositionPassword).Password
105        txtClue.Text = AllPasswords(PositionPassword).Clue
106        txtStaffType.Text = AllPasswords(PositionPassword).StaffType
107
108    End If
109
110 End Sub

```

```

111
112 1 reference
113 Private Sub ddlPostcode_SelectedIndexChanged() Handles ddlPostcode.SelectedIndexChanged
114
115     'Compares value within ddlPostcode to postcodes within the staff textfile
116     For x = 0 To NumStaff - 1
117         If ddlPostcode.Text = AllStaff(x).Postcode Then
118             txtStaffID.Text = AllStaff(x).StaffID
119             txtTitle.Text = AllStaff(x).Title
120             txtFname.Text = AllStaff(x).Fname
121             txtSname.Text = AllStaff(x).Sname
122             txtFAddress.Text = AllStaff(x).FAddress
123             txtSAddress.Text = AllStaff(x).SAddress
124             txtPost.Text = AllStaff(x).Postcode
125             txtTown.Text = AllStaff(x).Town
126             txtNum.Text = AllStaff(x).Phone
127             txtEmail.Text = AllStaff(x).Email
128             calendarDoB.Text = AllStaff(x).DoB
129             ddlSpecial.Text = AllStaff(x).Speciality
130
131             txtPassword.Text = AllPasswords(PositionPassword).Password
132             txtClue.Text = AllPasswords(PositionPassword).Clue
133             txtStaffType.Text = AllPasswords(PositionPassword).StaffType
134
135         End If
136     Next x
137 End Sub

```

```

120         txtSname.Text = AllStaff(x).Sname
121         txtFAddress.Text = AllStaff(x).FAddress
122         txtSAddress.Text = AllStaff(x).SAddress
123         txtPost.Text = AllStaff(x).Postcode
124         txtTown.Text = AllStaff(x).Town
125         txtNum.Text = AllStaff(x).Phone
126         txtEmail.Text = AllStaff(x).Email
127         calendarDoB.Text = AllStaff(x).DoB
128         ddlSpecial.Text = AllStaff(x).Speciality
129         PositionStaff = x
130
131         'Compare staff ID from staff textfile to the password textfile
132         For y = 0 To NumPasswords - 1
133             If AllStaff(x).StaffID = AllPasswords(y).StaffID Then
134                 txtPassword.Text = AllPasswords(y).Password
135                 txtClue.Text = AllPasswords(y).Clue
136                 txtStaffType.Text = AllPasswords(y).StaffType
137                 PositionStaff = x
138             End If
139         Next
140     End If
141 Next
142
143
144
145 End Sub
146

```

```

D references
147 Private Sub butSave_Click(sender As Object, e As EventArgs) Handles butSave.Click
148
149     "Position" is from the ddlStaffID code
150     'StaffID and PasswordID will not be changed
151
152     '*****STAFF FILE
153
154     'Checking if generate ID button has been clicked
155     If ddlSname.Text = "" Or ddlPostcode.Text = "" Then
156         MsgBox("Please select a staff member")
157         Exit Sub
158     End If

```

```

160     AllStaff(PositionStaff).Title = txtTitle.Text
161     'List check on title
162     If AllStaff(PositionStaff).Title <> "Mr" And AllStaff(PositionStaff).Title <> "Mrs" And AllStaff(PositionStaff).Title <> "Miss" And AllStaff(PositionStaff).Title <> "
163         MsgBox("Incorrect title entered, correct titles; Mr, Mrs, Miss, Ms, Mx, Sir, Dr, Cllr, Lord, Lady. Please try again")
164         Exit Sub
165     End If
166
167     'Correct capitalisation on first name and surname
168     AllStaff(PositionStaff).Fname = txtFname.Text.Substring(0, 1).ToUpper & txtFname.Text.Substring(1, txtFname.Text.Length - 1).ToLower
169     AllStaff(PositionStaff).Sname = txtSname.Text.Substring(0, 1).ToUpper & txtSname.Text.Substring(1, txtSname.Text.Length - 1).ToLower
170
171     'Entering textbox values into FAddress, SAddress, Postcode and Town
172     AllStaff(PositionStaff).FAddress = txtFAddress.Text
173     AllStaff(PositionStaff).SAddress = txtSAddress.Text
174     AllStaff(PositionStaff).Postcode = txtPost.Text
175     AllStaff(PositionStaff).Town = txtTown.Text
176
177     AllStaff(PositionStaff).Phone = txtNum.Text
178     'Length check on phone number
179     If Len(AllStaff(PositionStaff).Phone) <> 11 Then
180         MsgBox("Incorrect phone number entered, must be 11 characters long, please try again")
181         Exit Sub
182     End If
183
184     AllStaff(PositionStaff).Email = txtEmail.Text
185     'Format check on email
186     Dim EmailFormat As Boolean = AllStaff(PositionStaff).Email Like "*@*.*"
187     If EmailFormat = False Then
188         MsgBox("Incorrect email format detected, must be like __@__._, please try again")
189         Exit Sub
190     End If
191
192     'Entering date picker value into Date of Birth
193     AllStaff(PositionStaff).DoB = calendarDoB.Text
194
195     AllStaff(PositionStaff).Speciality = ddlSpecial.Text
196

```

```

162     Title <> "Ms" And AllStaff(PositionStaff).Title <> "Mx" And AllStaff(PositionStaff).Title <> "Sir" And AllStaff(PositionStaff).Title <> "Dr"
163

```

```

162 title <> "Dr" And AllStaff(PositionStaff).Title <> "Cllr" And AllStaff(PositionStaff).Title <> "Lady" And AllStaff(PositionStaff).Title <> "Lord" Then
163
197 .....PASSWORD FILE
198
199 'The password textfile is already loaded
200 'The PasswordID and StaffID will remain the same
201
202 Dim PasswordFound As Boolean = False 'Used when searching for a staff members password
203
204 'Find password data
205 For y = 0 To NumPasswords - 1
206     If txtStaffID.Text = AllPasswords(y).StaffID Then
207         PositionPassword = y
208         PasswordFound = True
209     End If
210 Next
211
212 'Check if password was found
213 If PasswordFound = False Then
214     MsgBox("ERROR: Password data has not been found")
215 End If
216
217 'List check on staff type
218 If txtStaffType.Text <> "Worker" And txtStaffType.Text <> "Admin" Then
219     MsgBox("Incorrect staff type, must be 'Worker' or 'Admin', please try again")
220     Exit Sub
221 End If
222 AllPasswords(PositionPassword).StaffType = txtStaffType.Text
223
224 AllPasswords(PositionPassword).Password = txtPassword.Text
225 AllPasswords(PositionPassword).Clue = txtClue.Text
226
227 'Checking that the password and clue do not include ","
228 Dim ArrayPasswordCommaCheck() As Char = txtPassword.Text.ToCharArray
229 Dim ArrayClueCommaCheck() As Char = txtClue.Text.ToCharArray
230 For x = 0 To Len(txtPassword.Text) - 1
231     If ArrayPasswordCommaCheck(x) = "," Then
232         MsgBox("Your password must not include a comma, please try again")
233         Exit Sub
234     End If
235 Next
236 For x = 0 To Len(txtClue.Text) - 1
237     If ArrayClueCommaCheck(x) = "," Then

```

```

238     MsgBox("Your clue must not include a comma, please try again")
239     Exit Sub
240 End If
241 Next
242
243 'Confirmation that the details are correct
244 If MsgBox("Please confirm these details by clicking OK", 1, "Confirmation") = vbCancel Then
245     Exit Sub 'Exit if user clicks cancel
246 End If
247
248 MsgBox("Please remember your password: " & AllPasswords(PositionPassword).Password)
249
250 Save_All_Staff() 'Saves the edits to the staff textfile
251 Save_All_Passwords() 'Saves the edits to the password textfile
252
253 MsgBox("Edited data has been saved (Please note you cannot edit the ID)")
254
255 End Sub
256
257 0 references
258 Private Sub butCancel_Click(sender As Object, e As EventArgs) Handles butCancel.Click
259
260     'Calls Postcode_changed subroutine to load the staff member with the surname and postcode from the ddls
261     ddlPostcode_SelectedIndexChanged()
262     MsgBox("Changes have been reset")
263
264 End Sub
265
266 0 references
267 Private Sub butDelete_Click(sender As Object, e As EventArgs) Handles butDelete.Click
268
269     'Take the position and replace the data with the data in the next position until end of array reached
270     For x = PositionStaff To NumStaff - 2 '-2 because we cannot replace the last variable with a non-existing one
271         AllStaff(x).StaffID = AllStaff(x + 1).StaffID
272         AllStaff(x).Title = AllStaff(x + 1).Title
273         AllStaff(x).Fname = AllStaff(x + 1).Fname
274         AllStaff(x).Sname = AllStaff(x + 1).Sname
275         AllStaff(x).FAddress = AllStaff(x + 1).FAddress
276         AllStaff(x).SAddress = AllStaff(x + 1).SAddress
277         AllStaff(x).Postcode = AllStaff(x + 1).Postcode
278         AllStaff(x).Town = AllStaff(x + 1).Town
279
280         AllStaff(x).Phone = AllStaff(x + 1).Phone
281         AllStaff(x).Email = AllStaff(x + 1).Email
282         AllStaff(x).DoB = AllStaff(x + 1).DoB
283         AllStaff(x).Speciality = AllStaff(x + 1).Speciality
284     Next
285
286     'Take the position and replace the data with the data in the next position until end of array reached
287     For x = PositionPassword To NumPasswords - 2 '-2 because we cannot replace the last variable with a non-existing one
288         AllPasswords(x).PasswordID = AllPasswords(x + 1).PasswordID
289         AllPasswords(x).StaffID = AllPasswords(x + 1).StaffID
290         AllPasswords(x).Password = AllPasswords(x + 1).Password
291         AllPasswords(x).Clue = AllPasswords(x + 1).Clue
292     Next
293
294     'Decremate the number of staff and passwords as one of each has been deleted
295     'This is so that when SaveAllStaff and SaveAllPasswords is called, the bottom line is ignored
296     NumStaff = NumStaff - 1
297     NumPasswords = NumPasswords - 1
298
299     'Saves the changes
300     Save_All_Staff()
301     Save_All_Passwords()
302     MsgBox("Staff member deleted")
303
304     'Close the form as this solves many problems
305     butQuit_Click()
306
307 End Sub
308
309 1 reference
310 Private Sub butQuit_Click() Handles butQuit.Click
311
312     Me.Close()
313
314 End Sub
315
316 End Class

```



## Form Name:

### Searching a Yearly Pay Slip

**Please fill these options to load corresponding data**

Staff ID

Year

Hours Worked

Gross Pay (£)

NI (£)

Tax (£)

Pension (£)

Net Pay (£)

**Staff Details (cannot edit these here)**

First Name

Surname

## Code Listing:

```
1 Public Class SearchYearlyPayroll
2
3     0 references
4     Private Sub SearchYearlyPayroll_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5         'Load appropriate textfiles
6         Load_Staff()
7         Load_Payroll()
8
9         'Clear ddls incase something was previously loaded
10        ddlStaffID.Items.Clear()
11        ddlYear.Items.Clear()
12
13        'Load staff into ddl
14        For x = 0 To NumStaff - 1
15            ddlStaffID.Items.Add(AllStaff(x).StaffID)
16        Next
17
18        'Load years into ddl
19        ddlYear.Items.Add("2021")
20        ddlYear.Items.Add("2022")
21        ddlYear.Items.Add("2023")
22        ddlYear.Items.Add("2024")
23        ddlYear.Items.Add("2025")
24        ddlYear.Items.Add("2026")
25
26    End Sub
27
28    0 references
29    Private Sub butSearchStaff_Click(sender As Object, e As EventArgs) Handles butSearchStaff.Click
30        'Open form to search staff
31        SearchStaff.Show()
32
33    End Sub
34
35    1 reference
36    Private Sub ddlStaffID_SelectedIndexChanged() Handles ddlStaffID.SelectedIndexChanged
37        'Load staff name into textboxes
38        For x = 0 To NumStaff - 1
39            If ddlStaffID.Text = AllStaff(x).StaffID Then
```

```

40     txtFnameStaff.Text = AllStaff(x).Fname
41     txtSnameStaff.Text = AllStaff(x).Sname
42     End If
43 Next
44
45 'Presence check on year
46 If ddlYear.Text = "" Then
47     MsgBox("Please enter year")
48     Exit Sub
49 End If
50
51 'Check if the selected year has an existing masterfile
52 Try 'Try loading masterfile
53     Dim file As System.IO.StreamReader
54     file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Payroll_Masterfile_" & ddlYear.Text & ".txt"))
55     Exit Try 'If masterfile found then continue with sub
56
57 Catch 'If doesn't work then masterfile doesn't exist
58     MsgBox("Masterfile for " & ddlYear.Text & " doesn't exist")
59     Exit Sub
60 End Try
61
62 'Check that this staff members data is within the masterfile
63 For x = 0 To NumPayroll - 1
64     If AllPayroll(x).StaffID = ddlStaffID.Text Then
65         txtHoursWorked.Text = AllPayroll(x).HoursWorked
66         txtGross.Text = AllPayroll(x).GrossPay
67         txtNI.Text = AllPayroll(x).NI
68         txtTax.Text = AllPayroll(x).Tax
69         txtPension.Text = AllPayroll(x).Pension
70         txtNetPay.Text = AllPayroll(x).NetPay
71         Exit Sub 'Exit subroutine once staff data loaded
72     End If
73 Next
74
75 'If program gets to this line then the staff member must not exist within the payroll textfile
76 MsgBox("Staff member has not been found within the masterfile")
77
78 End Sub
79

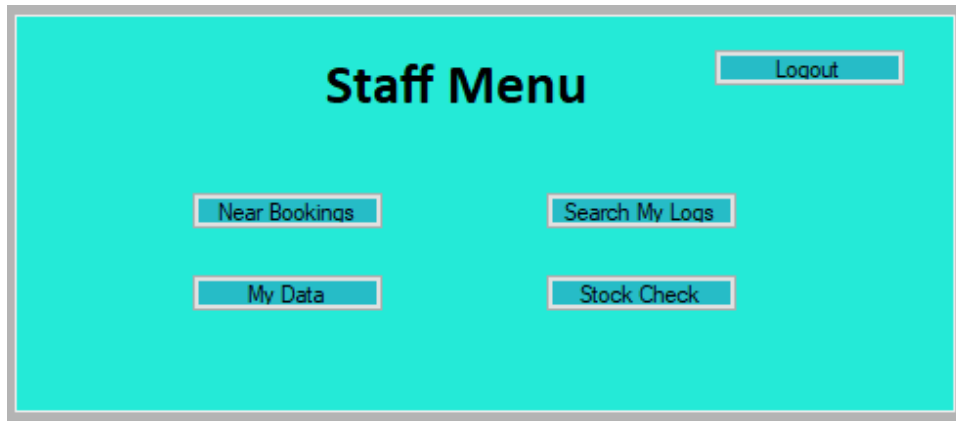
```

```

79
80 0 references
81 Private Sub ddlYear_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ddlYear.SelectedIndexChanged
82
83     'Presence check on staff ID
84     If ddlStaffID.Text = "" Then
85         MsgBox("Please enter staff ID")
86         Exit Sub
87     End If
88
89     'Open ddlStaffID_SelectedIndexChanged() sub as it loads the data into the textboxes
90     ddlStaffID_SelectedIndexChanged()
91 End Sub
92
93 0 references
94 Private Sub butQuit_Click(sender As Object, e As EventArgs) Handles butQuit.Click
95
96     'Close form
97     Me.Close()
98 End Sub
99 End Class

```

Form Name:



Code Listing:

```
1 reference
1 Public Class StaffMenu
2     0 references
3     Private Sub butTodayBookings_Click(sender As Object, e As EventArgs) Handles butTodayBookings.Click
4         'Open form for showing a workers bookings for today
5         MyBookingsToday.Show()
6     End Sub
7
8
9     0 references
10    Private Sub butSearchLogs_Click(sender As Object, e As EventArgs) Handles butSearchLogs.Click
11        'Open form for showing the worker's logs
12        WorkerLogs.Show()
13    End Sub
14
15
16    0 references
17    Private Sub butStockCheck_Click(sender As Object, e As EventArgs) Handles butStockCheck.Click
18        Dim NeedsReorder As String = "" 'Variable to hold the items that need testing
19        Dim AmountMissing As Integer 'Variable to hold the amount of a type of item that is missing
20
21        'Load equipment from textfile into array
22        Load_Equipment()
23
24        'Search through all equipment to find which items are below the reorder level
25        For x = 0 To (NumEquipment - 1)
26            If AllEquipment(x).AmountInStock < AllEquipment(x).ReorderLevel + 1 Then
27
28                AmountMissing = AllEquipment(x).ReorderLevel - AllEquipment(x).AmountInStock 'Calculate how much an item is below the stock level
29                AmountMissing = AmountMissing + AllEquipment(x).ReorderAmount 'Add on the reorder amount to get the stock amount past the reorder stock level
30
31                'Put items how many of an item are required into
32                If AmountMissing = 1 Then
33                    'Grammatical correctness so that if 1 item is missing it won't add an "s"
34                    NeedsReorder = NeedsReorder & AmountMissing & " more " & AllEquipment(x).ItemName & " required, "
35                Else
36                    NeedsReorder = NeedsReorder & AmountMissing & " more " & AllEquipment(x).ItemName & "s required, "
37                End If
38            End If
39        End For
40    End Sub
41 End Class
```

```

40     Next
41
42     'Removes the comma at the end
43     NeedsReorder = NeedsReorder.Remove(Len(NeedsReorder) - 2)
44
45     'Print variable
46     MsgBox(NeedsReorder)
47
48 End Sub
49
50 0 references
51 Private Sub butMyData_Click(sender As Object, e As EventArgs) Handles butMyData.Click
52     'Load staff from textfile into array
53     Load_Staff()
54
55     'Goes through staff array
56     For x = 0 To NumStaff - 1
57         If AllStaff(x).StaffID = LoggedOnStaffID Then
58             'Displays staff data in a multuple line message box
59             MsgBox("Staff ID: " & AllStaff(x).StaffID + vbCrLf + "Full name: " & AllStaff(x).Title & " " & AllStaff(x).Fname & " " & AllStaff(x).Sname + vbCrLf + "Address: "
60             End If
61         Next
62
63 End Sub
64
65 0 references
66 Private Sub butLogout_Click(sender As Object, e As EventArgs) Handles butLogout.Click
67     'Close form
68     Me.Close()
69
70 End Sub
71
72 End Class

```

---

```

59     + "Address: " & AllStaff(x).FAddress & " " & AllStaff(x).SAddress & " " & AllStaff(x).Postcode & " " & AllStaff(x).Town + vbCrLf + "Phone numb
60

```

---

```

59     f + "Phone number: " & AllStaff(x).Phone + vbCrLf + "Email: " & AllStaff(x).Email + vbCrLf + "Date of birth: " & AllStaff(x).DoB + vbCrLf + "Sp
60

```

---

```

59     f(x).Phone + vbCrLf + "Email: " & AllStaff(x).Email + vbCrLf + "Date of birth: " & AllStaff(x).DoB + vbCrLf + "Speciality: " & AllStaff(x).Speciality)
60

```

Form Name:

### Searching Uninvoiced Bookings

**From:**   **Until:**

Code Listing:

```

1 1 reference
2  Public Class UninvoicedBookings
3
4      Public NumDays As Integer 'Variable to hold "Days until booking"
5      Public BookingFound As Boolean = False
6
7  0 references
8  Private Sub UninvoicedBookings_Load() Handles MyBase.Load
9
10     'Load appropriate textfiles
11     Load_Bookings()
12     Load_Invoices()
13     Load_Customers()
14
15     'Clear richtext box in case anything is loaded in
16     rtxtUninvoicedBookings.Text = ""
17
18     'Loop through bookings
19     For x = 0 To NumBookings - 1
20         If AllBookings(x).HasBeenInvoiced = False Then 'Load invoice data for all invoices that haven't been paid for
21             BookingFound = True
22
23             'Load booking data
24             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text & "- - - - -" + vbCrLf
25             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
26             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Date: " & AllBookings(x).BookingDate + vbCrLf
27             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
28
29             'Load staff details
30             For y = 0 To NumStaff - 1
31                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
32                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff ID: " & AllStaff(y).StaffID + vbCrLf + vbCrLf
33             Next
34
35             'Load customer details
36             For y = 0 To NumCustomers - 1
37                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
38                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
39                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer ID: " & AllCustomers(y).CustomerID + vbCrLf
40                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Address: " & AllCustomers(y).FAddress & " " &
41                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).SAddress & " " &
42                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Postcode & " " &
43                     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Town + vbCrLf

```

```

44     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Phone number: " & AllCustomers(x).Phone + vbCrLf
45     rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Email: " & AllCustomers(x).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
46     End If
47     Next
48 End If
49 Next
50
51 'Message if no uninvoiced bookings were found
52 If BookingFound = False Then
53     rtxtUninvoicedBookings.Text = "No uninvoiced bookings found"
54 End If
55 End Sub

```

```

56 0 references
57 Private Sub calendarMinDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarMinDate.ValueChanged
58
59     'Clear rich textbox and reset BookingFound
60     rtxtUninvoicedBookings.Text = ""
61     BookingFound = False
62
63     'Loop through bookings
64     For x = 0 To NumBookings - 1
65
66         'Check if booking invoiced for and that the booking date is within the selected date values
67         If AllBookings(x).HasBeenInvoiced = False And AllBookings(x).BookingDate > calendarMinDate.Value And AllBookings(x).BookingDate < calendarMaxDate.Value.AddDays(1) Then
68             BookingFound = True
69
70             'Load booking data
71             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text & "- - - - -" + vbCrLf
72             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
73             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Date: " & AllBookings(x).BookingDate + vbCrLf
74             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
75
76             'Load staff details
77             For y = 0 To NumStaff - 1
78                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
79                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff ID: " & AllStaff(y).StaffID + vbCrLf + vbCrLf
80             Next
81
82             'Load customer details
83             For y = 0 To NumCustomers - 1

```

```

83     If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
84         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
85         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer ID: " & AllCustomers(y).CustomerID + vbCrLf
86         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Address: " & AllCustomers(y).FAddress & " , "
87         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).SAddress & " , "
88         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Postcode & " , "
89         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Town + vbCrLf
90         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Phone number: " & AllCustomers(x).Phone + vbCrLf
91         rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Email: " & AllCustomers(x).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
92     End If
93     Next
94 End If
95 Next
96
97 'Message if no uninvoiced bookings were found
98 If BookingFound = False Then
99     rtxtUninvoicedBookings.Text = "No uninvoiced bookings found"
100 End If
101
102 End Sub
103

```

```

104 0 references
105 Private Sub calendarMaxDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarMaxDate.ValueChanged
106
107     'Clear rich textbox and reset BookingFound
108     rtxtUninvoicedBookings.Text = ""
109     BookingFound = False
110
111     'Loop through bookings
112     For x = 0 To NumBookings - 1
113
114         'Check if booking invoiced for and that the booking date is within the selected date values
115         If AllBookings(x).HasBeenInvoiced = False And AllBookings(x).BookingDate > calendarMinDate.Value And AllBookings(x).BookingDate < calendarMaxDate.Value.AddDays(1) Then
116             BookingFound = True
117
118             'Load booking data
119             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text & "- - - - -" + vbCrLf
120             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
121             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Date: " & AllBookings(x).BookingDate + vbCrLf
122             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Booking Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
123

```

```

123         'Load staff details
124         For y = 0 To NumStaff - 1
125             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff name: " & AllStaff(y).Fname & " " & AllStaff(y).Sname + vbCrLf
126             rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Staff ID: " & AllStaff(y).StaffID + vbCrLf + vbCrLf
127         Next
128
129         'Load customer details
130         For y = 0 To NumCustomers - 1
131             If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
132                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
133                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Customer ID: " & AllCustomers(y).CustomerID + vbCrLf
134                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Address: " & AllCustomers(y).FAddress & " , "
135                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).SAddress & " , "
136                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Postcode & " , "
137                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + AllCustomers(y).Town + vbCrLf
138                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Phone number: " & AllCustomers(x).Phone + vbCrLf
139                 rtxtUninvoicedBookings.Text = rtxtUninvoicedBookings.Text + "Email: " & AllCustomers(x).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
140             End If
141         Next
142     End If
143 Next
144
145 'Message if no uninvoiced bookings were found
146 If BookingFound = False Then
147     rtxtUninvoicedBookings.Text = "No uninvoiced bookings found"
148 End If
149
150 End Sub
151

```

```

152 0 references
153 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
154
155     Me.Close()
156
157 End Sub
158 End Class

```

**Form Name:**

## Searching Unpaid Invoices

**From:**   **Until:**

**Code Listing:**

```

1 reference
1 Public Class UnpaidInvoices
2
3     Public NumDays As Integer 'Variable to hold "Days since booking"
4     Public InvoiceFound As Boolean = False
5
6 0 references
7 Private Sub UnpaidInvoices_Load() Handles MyBase.Load
8
9     'Load appropriate textfiles
10    Load_Bookings()
11    Load_Invoices()
12    Load_Customers()
13
14    'Clear richtext box in case anything is loaded in
15    rtxtUnpaidBooks.Text = ""
16
17    'Loop through invoices
18    For x = 0 To NumInvoices - 1
19        If AllInvoices(x).Paid = False Then 'Load invoice data for all invoices that haven't been paid for
20            InvoiceFound = True
21            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text & "-----" + vbCrLf
22            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Invoice ID: " & Str(AllInvoices(x).InvoiceID) + vbCrLf
23            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking ID: " & AllInvoices(x).BookingID + vbCrLf
24            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Date: " & AllInvoices(x).BookingDate + vbCrLf
25
26            'Loop through booking details
27            For y = 0 To NumBookings - 1
28                If AllInvoices(x).BookingID = AllBookings(y).BookingID Then
29                    rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Notes: " & AllBookings(y).JobNotes + vbCrLf + vbCrLf
30
31                    'Load customer details
32                    For z = 0 To NumCustomers - 1
33                        If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
34                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer name: " & AllCustomers(z).Fname & " " & AllCustomers(z).Sname + vbCrLf
35                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer ID: " & AllCustomers(z).CustomerID + vbCrLf
36                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Address: " & AllCustomers(z).FAddress & ", "
37                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).SAddress & ", "
38                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Postcode & ", "
39                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Town + vbCrLf
40                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Phone number: " & AllCustomers(x).Phone + vbCrLf
41                            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Email: " & AllCustomers(x).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
42                        End If
43                    Next
44                End If
45            Next
46        End If
47    Next
48
49    'Message if no unpaid invoices were found
50    If InvoiceFound = False Then
51        rtxtUnpaidBooks.Text = "No unpaid invoices found"
52    End If
53 End Sub

```

```

0 references
55 Private Sub calendarMinDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarMinDate.ValueChanged
56
57    'Clear rich textbox and reset InvoiceFound
58    rtxtUnpaidBooks.Text = ""
59    InvoiceFound = False
60
61    'Loop through invoices
62    For x = 0 To NumInvoices - 1
63
64        'Check if invoice paid for and that the booking date is within the selected date values
65        If AllInvoices(x).Paid = False And AllInvoices(x).BookingDate > calendarMinDate.Value And AllInvoices(x).BookingDate < calendarMaxDate.Value.AddDays(1) Then
66            InvoiceFound = True
67
68            'Load invoice data for all invoices that haven't been paid for
69            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text & "-----" + vbCrLf
70            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Invoice ID: " & Str(AllInvoices(x).InvoiceID) + vbCrLf
71            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking ID: " & AllInvoices(x).BookingID + vbCrLf
72            rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Date: " & AllInvoices(x).BookingDate + vbCrLf
73
74            'Loop through booking details
75            For y = 0 To NumBookings - 1
76                If AllInvoices(x).BookingID = AllBookings(y).BookingID Then
77                    rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Notes: " & AllBookings(y).JobNotes + vbCrLf + vbCrLf
78
79                    'Load customer details
80                    For z = 0 To NumCustomers - 1

```

```

81         If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
82             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer name: " & AllCustomers(z).Fname & " " & AllCustomers(z).Sname + vbCrLf
83             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer ID: " & AllCustomers(z).CustomerID + vbCrLf
84             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Address: " & AllCustomers(z).FAddress & " , "
85             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).SAddress & " , "
86             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Postcode & " , "
87             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Town + vbCrLf
88             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Phone number: " & AllCustomers(z).Phone + vbCrLf
89             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Email: " & AllCustomers(z).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
90         End If
91     Next
92 End If
93 Next
94 End If
95 Next
96
97 'Message if no unpaid invoices were found
98 If InvoiceFound = False Then
99     rtxtUnpaidBooks.Text = "No unpaid invoices found"
100 End If
101
102 End Sub
103
104 0 references
105 Private Sub calendarMaxDate_ValueChanged(sender As Object, e As EventArgs) Handles calendarMaxDate.ValueChanged
106
107     'Clear rich textbox and reset InvoiceFound
108     rtxtUnpaidBooks.Text = ""
109     InvoiceFound = False
110
111     'Loop through invoices
112     For x = 0 To NumInvoices - 1
113
114         'Check if invoice paid for and that the booking date is within the selected date values
115         If AllInvoices(x).Paid = False And AllInvoices(x).BookingDate > calendarMinDate.Value And AllInvoices(x).BookingDate < calendarMaxDate.Value.AddDays(1) Then
116             InvoiceFound = True
117
118             'Load invoice data for all invoices that haven't been paid for
119             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text & "-----" + vbCrLf
120             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Invoice ID: " & Str(AllInvoices(x).InvoiceID) + vbCrLf
121
122             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking ID: " & AllInvoices(x).BookingID + vbCrLf
123             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Date: " & AllInvoices(x).BookingDate + vbCrLf
124
125             'Loop through booking details
126             For y = 0 To NumBookings - 1
127                 If AllInvoices(x).BookingID = AllBookings(y).BookingID Then
128                     rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Booking Notes: " & AllBookings(y).JobNotes + vbCrLf + vbCrLf
129
130                     'Load customer details
131                     For z = 0 To NumCustomers - 1
132                         If AllBookings(y).CustomerID = AllCustomers(z).CustomerID Then
133                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer name: " & AllCustomers(z).Fname & " " & AllCustomers(z).Sname + vbCrLf
134                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Customer ID: " & AllCustomers(z).CustomerID + vbCrLf
135                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Address: " & AllCustomers(z).FAddress & " , "
136                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).SAddress & " , "
137                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Postcode & " , "
138                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + AllCustomers(z).Town + vbCrLf
139                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Phone number: " & AllCustomers(z).Phone + vbCrLf
140                             rtxtUnpaidBooks.Text = rtxtUnpaidBooks.Text + "Email: " & AllCustomers(z).Email + vbCrLf + vbCrLf + vbCrLf 'Leave plenty white space
141                         End If
142                     Next
143                 End If
144             Next
145
146             'Message if no unpaid invoices were found
147             If InvoiceFound = False Then
148                 rtxtUnpaidBooks.Text = "No unpaid invoices found"
149             End If
150
151         End Sub
152
153 0 references
154 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
155
156     Me.Close()
157
158 End Sub
159 End Class

```

**Form Name:**

## Worker Logs

Worker ID

Dates from  to

**Code Listing:**

```

1 reference
1 Public Class WorkerLogs
2
3     Public AllocatedHoursString As String 'Variable to allow "Allocated hours" to be displayed as a string
4     Public Found As Boolean = False 'Variable to check data has been found
5
6     0 references
7     Private Sub WorkerLogs_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8
9         'Load relevant textfiles
10        Load_Customers()
11        Load_Bookings()
12        Load_Staff()
13
14        'If a worker account accesses this form, doesn't load the other staff members
15        If LoggedOnStaffType = "Worker" Then
16            ddlStaffID.Items.Add(LoggedOnStaffID)
17            ddlStaffID.Text = LoggedOnStaffID
18            butSearch_Click() 'Clicks the search button
19            Exit Sub
20        End If
21
22        'Clear ddl incase there was something there previously
23        ddlStaffID.Items.Clear()
24
25        For x = 0 To NumStaff - 1
26            'Load customer IDs into the ddl
27            ddlStaffID.Items.Add(AllStaff(x).StaffID)
28        Next
29    End Sub
30
31    1 reference
32    Private Sub butSearch_Click() Handles butSearch.Click
33
34        'Presence check on staff ID
35        If ddlStaffID.Text = "" Then
36            rtxtLogs.Text = "No bookings found as no staff ID has been entered"
37            Exit Sub
38        End If
39
40        rtxtLogs.Text = "" 'Clear rich textbox incase something was loaded in before
41        Found = False 'Resets "Found"

```

```

41
42
43 'Format check on dates
44 If txtMinDate.Text <> "" Then
45     Dim DateFormatCheck As Boolean = txtMinDate.Text Like "***/**/*****"
46     If DateFormatCheck = False Or Len(txtMinDate.Text) <> 10 Then
47         MsgBox("Please ensure the dates are in the format ***/**/*****")
48         Exit Sub
49     End If
50 End If
51 If txtMaxDate.Text <> "" Then
52     Dim DateFormatCheck As Boolean = txtMaxDate.Text Like "***/**/*****"
53     If DateFormatCheck = False Or Len(txtMaxDate.Text) <> 10 Then
54         MsgBox("Please ensure the dates are in the format ***/**/*****")
55         Exit Sub
56     End If
57 End If
58
59 *****IF NO DATES ARE ENTERED*****
60
61 If txtMinDate.Text = "" And txtMaxDate.Text = "" Then
62     'Load all the customers bookings and their details
63     For x = 0 To NumBookings - 1
64         'Add the details of each booking to the rich textbox
65         If ddlStaffID.Text = AllBookings(x).StaffID Then
66             'Add the details of each booking to the rich textbox
67             rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
68             rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
69             rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
70             rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
71             rtxtLogs.Text = rtxtLogs.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
72
73             'Search through customers textfile, add address of customer to rich textbox
74             For y = 0 To NumCustomers - 1
75                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
76                     rtxtLogs.Text = rtxtLogs.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
77                     rtxtLogs.Text = rtxtLogs.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
78                     Exit For
79                 End If
80             Next
81             rtxtLogs.Text = rtxtLogs.Text + vbCrLf

```

```

76 t + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
77

```

```

82
83 AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
84 rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
85 rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
86 rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolList + vbCrLf
87 rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
88 rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
89
90 If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
91     For y = 0 To NumInvoices - 1
92         If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
93             rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
94         End If
95     Next
96
97 ElseIf AllBookings(x).HasBeenInvoiced = False Then 'If booking hasn't been invoiced
98     rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
99 End If
100
101 Found = True
102 End If
103
104 Next
105
106
107 ElseIf txtMinDate.Text <> "" Then *****IF MIN DATE ENTERED *****
108 If txtMaxDate.Text <> "" Then
109     For x = 0 To NumBookings
110         If ddlStaffID.Text = AllBookings(x).StaffID And AllBookings(x).BookingDate > txtMinDate.Text And AllBookings(x).BookingDate < txtMaxDate.Text Then
111             'Add the details of each booking to the rich textbox
112             rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
113             rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
114             rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
115             rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
116             rtxtLogs.Text = rtxtLogs.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
117
118             'Search through customers textfile, add address of customer to rich textbox
119             For y = 0 To NumCustomers - 1
120                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
121                     rtxtLogs.Text = rtxtLogs.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
122                     rtxtLogs.Text = rtxtLogs.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
123                     Exit For

```

```

122 .Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
123

```

```

124         End If
125     Next
126
127     rtxtLogs.Text = rtxtLogs.Text + vbCrLf
128
129     AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
130     rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
131     rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
132     rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
133     rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
134     rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
135
136     If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
137         For y = 0 To NumInvoices - 1
138             If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
139                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
140             End If
141         Next
142
143     ElseIf AllBookings(x).HasBeenInvoiced = False Then 'If booking hasn't been invoiced
144         rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
145     End If
146
147     Found = True
148 End If
149 Next
150
151
152 Else '***** IF ONLY MIN DATE ENTERED*****
153     For x = 0 To NumBookings
154         If ddlStaffID.Text = AllBookings(x).StaffID And AllBookings(x).BookingDate > txtMinDate.Text Then
155             'Add the details of each booking to the rich textbox
156             rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
157             rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
158             rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
159             rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
160             rtxtLogs.Text = rtxtLogs.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
161
162             'Search through customers textfile, add address of customer to rich textbox
163             For y = 0 To NumCustomers - 1

```

```

164                 If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
165                     rtxtLogs.Text = rtxtLogs.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
166                     rtxtLogs.Text = rtxtLogs.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " &
167                     Exit For
168                 End If
169             Next
170
171             rtxtLogs.Text = rtxtLogs.Text + vbCrLf
172
173             AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
174             rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
175             rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
176             rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
177             rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
178             rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
179
180             If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
181                 For y = 0 To NumInvoices - 1
182                     If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
183                         rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
184                     End If
185                 Next
186
187             ElseIf AllBookings(x).HasBeenInvoiced = False Then 'If booking hasn't been invoiced
188                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
189             End If
190
191             Found = True
192         End If
193     Next
194
195 End If
196
197 ElseIf txtMaxDate.Text <> "" Then '*****IF MAX DATE ENTERED*****
198     If txtMinDate.Text <> "" Then
199         For x = 0 To NumBookings
200             If ddlStaffID.Text = AllBookings(x).StaffID And AllBookings(x).BookingDate > txtMinDate.Text And AllBookings(x).BookingDate < txtMaxDate.Text Then
201                 'Add the details of each booking to the rich textbox
202                 rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
203                 rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf

```

```

166         .Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf

```

```

284 rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
285 rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
286 rtxtLogs.Text = rtxtLogs.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
287
288 'Search through customers textfile, add address of customer to rich textbox
289 For y = 0 To NumCustomers - 1
290     If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
291         rtxtLogs.Text = rtxtLogs.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
292         rtxtLogs.Text = rtxtLogs.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " "
293         Exit For
294     End If
295 Next
296
297 rtxtLogs.Text = rtxtLogs.Text + vbCrLf
298
299 AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
300 rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
301 rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
302 rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
303 rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
304 rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
305
306 If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
307     For y = 0 To NumInvoices - 1
308         If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
309             rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
310         End If
311     Next
312
313 ElseIf AllBookings(x).HasBeenInvoiced = False Then 'If booking hasn't been invoiced
314     rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
315 End If
316
317 Found = True
318 End If
319 Next
320
321 Else *****IF ONLY MAX DATE ENTERED*****
322 For x = 0 To NumBookings

```

```

212 .Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
213

```

```

244 If ddlStaffID.Text = AllBookings(x).StaffID And AllBookings(x).BookingDate < txtMaxDate.Text Then
245     'Add the details of each booking to the rich textbox
246     rtxtLogs.Text = rtxtLogs.Text + "-----" + vbCrLf
247     rtxtLogs.Text = rtxtLogs.Text + "Booking ID: " & AllBookings(x).BookingID + vbCrLf
248     rtxtLogs.Text = rtxtLogs.Text + "Date: " & AllBookings(x).BookingDate + vbCrLf
249     rtxtLogs.Text = rtxtLogs.Text + "Job Type: " & AllBookings(x).JobType + vbCrLf + vbCrLf
250     rtxtLogs.Text = rtxtLogs.Text + "Customer ID: " & AllBookings(x).CustomerID + vbCrLf
251
252     'Search through customers textfile, add address of customer to rich textbox
253     For y = 0 To NumCustomers - 1
254         If AllBookings(x).CustomerID = AllCustomers(y).CustomerID Then
255             rtxtLogs.Text = rtxtLogs.Text + "Customer Name: " & AllCustomers(y).Fname & " " & AllCustomers(y).Sname + vbCrLf
256             rtxtLogs.Text = rtxtLogs.Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Pos
257             Exit For
258         End If
259     Next
260
261     rtxtLogs.Text = rtxtLogs.Text + vbCrLf
262
263     AllocatedHoursString = CStr(AllBookings(x).AllocatedTime) 'Convert to string data type
264     rtxtLogs.Text = rtxtLogs.Text + "Hours allocated: " & AllocatedHoursString + vbCrLf
265     rtxtLogs.Text = rtxtLogs.Text + "Price: " & AllBookings(x).Price + vbCrLf
266     rtxtLogs.Text = rtxtLogs.Text + "Tools: " & AllBookings(x).ToolsList + vbCrLf
267     rtxtLogs.Text = rtxtLogs.Text + "Notes: " & AllBookings(x).JobNotes + vbCrLf + vbCrLf
268     rtxtLogs.Text = rtxtLogs.Text + "Completed/Invoiced?: " & Str(AllBookings(x).HasBeenInvoiced) + vbCrLf
269
270     If AllBookings(x).HasBeenInvoiced = True Then 'If booking has been invoiced
271         For y = 0 To NumInvoices - 1
272             If AllBookings(x).BookingID = AllInvoices(y).BookingID Then
273                 rtxtLogs.Text = rtxtLogs.Text + "Paid?: " & Str(AllInvoices(y).Paid) + vbCrLf + vbCrLf + vbCrLf
274             End If
275         Next
276
277     ElseIf AllBookings(x).HasBeenInvoiced = False Then 'If booking hasn't been invoiced
278         rtxtLogs.Text = rtxtLogs.Text + "Paid?: False (Booking has not been completed/invoiced yet)" + vbCrLf + vbCrLf + vbCrLf
279     End If
280
281     Found = True
282 End If
283 Next

```

```

256 .Text + "Address: " & AllCustomers(y).FAddress & " " & AllCustomers(y).SAddress & " " & AllCustomers(y).Postcode & " " & AllCustomers(y).Town + vbCrLf
257

```

```
282         End If
283     Next
284
285     End If
286 End If
287
288 'Message if no data was found
289 If Found = False Then
290     rtxtLogs.Text = "No bookings found"
291 End If
292
293 End Sub
294
295 0 references
296 Private Sub butClose_Click(sender As Object, e As EventArgs) Handles butClose.Click
297     Me.Close()
298 End Sub
299
300 0 references
301 Private Sub butSearchStaff_Click(sender As Object, e As EventArgs) Handles butSearchStaff.Click
302
303     'Open form for searching staff
304     SearchStaff.Show()
305 End Sub
306 End Class
307
```

## Module Name:

▶ VB Password\_Module.vb

## Code Listing:

```
0 references
1 Module Password_Module
2
3 .....CUSTOMERS.....
4 'Declaring temporary variables for bubble sort
5 Public TempCust1 As String
6 Public TempCust2 As String
7 Public TempCust3 As String
8 Public TempCust4 As String
9 Public TempCust5 As String
10 Public TempCust6 As String
11 Public TempCust7 As String
12 Public TempCust8 As String
13 Public TempCust9 As String
14 Public TempCust10 As String
15 Public TempCust11 As String
16
17 'Declaring a data structure for customers
1 reference
18 Public Structure Customer
19     Public CustomerID As String
20     Public Title As String
21     Public FName As String
22     Public Sname As String
23     Public FAddress As String
24     Public SAddress As String
25     Public Postcode As String
26     Public Town As String
27     Public Phone As String
28     Public Email As String
29     Public DoB As String
30 End Structure
31
32 'Declaring an array for customers
33 Public AllCustomers(0 To 99999) As Customer
34 Public NumCustomers As Integer = 0
35
13 references
36 Public Function Load_Customers()
37
38     NumCustomers = 0
39     Dim file As System.IO.StreamReader
40     Dim line As String
```

```

41 Dim parts(0 To 10) As String
42
43 'Loading the customer textfile
44 file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Customers.txt"))
45
46 Do
47     line = file.ReadLine()
48
49     'Reads each readline and splits the data at each "," and puts it into the AllCustomers array
50     parts = line.Split(",")
51     AllCustomers(NumCustomers).CustomerID = parts(0)
52     AllCustomers(NumCustomers).Title = parts(1)
53     AllCustomers(NumCustomers).Fname = parts(2)
54     AllCustomers(NumCustomers).Sname = parts(3)
55     AllCustomers(NumCustomers).FAddress = parts(4)
56     AllCustomers(NumCustomers).SAddress = parts(5)
57     AllCustomers(NumCustomers).Postcode = parts(6)
58     AllCustomers(NumCustomers).Town = parts(7)
59     AllCustomers(NumCustomers).Phone = parts(8)
60     AllCustomers(NumCustomers).Email = parts(9)
61     AllCustomers(NumCustomers).DoB = parts(10)
62
63     'Goes to the next customers data
64     NumCustomers = NumCustomers + 1
65 Loop Until (file.EndOfStream)
66
67 file.Close()
68
69 End Function

```

```

70
71 1 reference
72 Public Function Bubble_Sort_Customers()
73
74     'Bubble sort by customer ID
75     For x = 0 To NumCustomers - 2
76         For y = 0 To NumCustomers - 2
77             If AllCustomers(y).CustomerID > AllCustomers(y + 1).CustomerID Then
78                 TempCust1 = AllCustomers(y).CustomerID
79                 TempCust2 = AllCustomers(y).Title
80                 TempCust3 = AllCustomers(y).Fname

```

```

81                 TempCust4 = AllCustomers(y).Sname
82                 TempCust5 = AllCustomers(y).FAddress
83                 TempCust6 = AllCustomers(y).SAddress
84                 TempCust7 = AllCustomers(y).Postcode
85                 TempCust8 = AllCustomers(y).Town
86                 TempCust9 = AllCustomers(y).Phone
87                 TempCust10 = AllCustomers(y).Email
88                 TempCust11 = AllCustomers(y).DoB
89                 AllCustomers(y).CustomerID = AllCustomers(y + 1).CustomerID
90                 AllCustomers(y).Title = AllCustomers(y + 1).Title
91                 AllCustomers(y).Fname = AllCustomers(y + 1).Fname
92                 AllCustomers(y).Sname = AllCustomers(y + 1).Sname
93                 AllCustomers(y).FAddress = AllCustomers(y + 1).FAddress
94                 AllCustomers(y).SAddress = AllCustomers(y + 1).SAddress
95                 AllCustomers(y).Postcode = AllCustomers(y + 1).Postcode
96                 AllCustomers(y).Town = AllCustomers(y + 1).Town
97                 AllCustomers(y).Phone = AllCustomers(y + 1).Phone
98                 AllCustomers(y).Email = AllCustomers(y + 1).Email
99                 AllCustomers(y).DoB = AllCustomers(y + 1).DoB
100                AllCustomers(y + 1).CustomerID = TempCust1
101                AllCustomers(y + 1).Title = TempCust2
102                AllCustomers(y + 1).Fname = TempCust3
103                AllCustomers(y + 1).Sname = TempCust4
104                AllCustomers(y + 1).FAddress = TempCust5
105                AllCustomers(y + 1).SAddress = TempCust6
106                AllCustomers(y + 1).Postcode = TempCust7
107                AllCustomers(y + 1).Town = TempCust8
108                AllCustomers(y + 1).Phone = TempCust9
109                AllCustomers(y + 1).Email = TempCust10
110                AllCustomers(y + 1).DoB = TempCust11
111            End If
112        Next
113    Next
114 End Function

```

```

115
116 3 references
117 Public Function Save_All_Customers()

```

```

117
118 'Saving all the customers (i.e. for edits)
119 Dim SaveAllCustomers As System.IO.StreamWriter
120 SaveAllCustomers = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Customers.txt"), False)
121 For y = 0 To (NumCustomers - 1)
122     SaveAllCustomers.WriteLine(AllCustomers(y).CustomerID & "," & AllCustomers(y).Title & "," & AllCustomers(y).Fname & "," & AllCustomers(y).Sname & "," & AllCustomer
123 Next y
124 SaveAllCustomers.Close()
125
126 End Function
127
128 *****EQUIPMENT*****
129
130 'Declaring temporary variables for equipment bubble sort
131 Public TempEquip1 As String
132 Public TempEquip2 As String
133 Public TempEquip3 As String
134 Public TempEquip4 As String
135 Public TempEquip5 As String
136 Public TempEquip6 As String
137 Public TempEquip7 As String
138 Public TempEquip8 As String
139
140 'Declaring a data structure for equipment
141 1 reference
142 Public Structure Equipment
143     Public EquipmentID As String
144     Public ItemName As String
145     Public AmountInStock As Integer
146     Public ReorderLevel As Integer
147     Public ReorderAmount As Integer
148     Public LTestDate As String
149     Public NTestDate As String
150     Public TestedBy As String
151 End Structure
152
153 'Declaring an array for equipment
154 Public AllEquipment(0 To 999) As Equipment
155
122 customers(y).Sname & "," & AllCustomers(y).FAddress & "," & AllCustomers(y).SAddress & "," & AllCustomers(y).Postcode & "," & AllCustomers(y).Town & ",
123
---
122 & "," & AllCustomers(y).Postcode & "," & AllCustomers(y).Town & "," & AllCustomers(y).Phone & "," & AllCustomers(y).Email & "," & AllCustomers(y).DoB)
123
---
```

```

154 Public NumEquipment As Integer = 0
155
156 Public Function Load_Equipment()
157
158     NumEquipment = 0
159     Dim file As System.IO.StreamReader
160     Dim line As String
161     Dim parts(0 To 7) As String
162
163     'Loading the equipment textfile
164     file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Equipment.txt"))
165
166     Do
167         line = file.ReadLine()
168
169         'Reads each readline and splits the data at each "," and puts it into the AllEquipment array
170         parts = line.Split(",")
171         AllEquipment(NumEquipment).EquipmentID = parts(0)
172         AllEquipment(NumEquipment).ItemName = parts(1)
173         AllEquipment(NumEquipment).AmountInStock = parts(2)
174         AllEquipment(NumEquipment).ReorderLevel = parts(3)
175         AllEquipment(NumEquipment).ReorderAmount = parts(4)
176         AllEquipment(NumEquipment).LTestDate = parts(5)
177         AllEquipment(NumEquipment).NTestDate = parts(6)
178         AllEquipment(NumEquipment).TestedBy = parts(7)
179
180         'Goes to the next equipment data
181         NumEquipment = NumEquipment + 1
182     Loop Until (file.EndOfStream)
183
184     file.Close()
185
186 End Function
187
188 Public Function Bubble_Sort_Equipment()
189
190     'Bubble sort by equipment ID
191     For x = 0 To NumEquipment - 2

```

```

192     For y = 0 To NumEquipment - 2
193         If AllEquipment(y).ItemName > AllEquipment(y + 1).ItemName Then
194             TempEquip1 = AllEquipment(y).EquipmentID
195             TempEquip2 = AllEquipment(y).ItemName
196             TempEquip3 = AllEquipment(y).AmountInStock
197             TempEquip4 = AllEquipment(y).ReorderLevel
198             TempEquip5 = AllEquipment(y).ReorderAmount
199             TempEquip6 = AllEquipment(y).LTestDate
200             TempEquip7 = AllEquipment(y).NTestDate
201             TempEquip8 = AllEquipment(y).TestedBy
202             AllEquipment(y).EquipmentID = AllEquipment(y + 1).EquipmentID
203             AllEquipment(y).ItemName = AllEquipment(y + 1).ItemName
204             AllEquipment(y).AmountInStock = AllEquipment(y + 1).AmountInStock
205             AllEquipment(y).ReorderLevel = AllEquipment(y + 1).ReorderLevel
206             AllEquipment(y).ReorderAmount = AllEquipment(y + 1).ReorderAmount
207             AllEquipment(y).LTestDate = AllEquipment(y + 1).LTestDate
208             AllEquipment(y).NTestDate = AllEquipment(y + 1).NTestDate
209             AllEquipment(y).TestedBy = AllEquipment(y + 1).TestedBy
210             AllEquipment(y + 1).EquipmentID = TempEquip1
211             AllEquipment(y + 1).ItemName = TempEquip2
212             AllEquipment(y + 1).AmountInStock = TempEquip3
213             AllEquipment(y + 1).ReorderLevel = TempEquip4
214             AllEquipment(y + 1).ReorderAmount = TempEquip5
215             AllEquipment(y + 1).LTestDate = TempEquip6
216             AllEquipment(y + 1).NTestDate = TempEquip7
217             AllEquipment(y + 1).TestedBy = TempEquip8
218         End If
219     Next
220 Next
221
222 End Function
223
224 4 references
225 Public Function Save_All_Equipment()
226     Dim SaveAllEquipment As System.IO.StreamWriter
227     SaveAllEquipment = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Equipment.txt"), False)
228     For y = 0 To (NumEquipment - 1)
229         SaveAllEquipment.WriteLine(AllEquipment(y).EquipmentID & "," & AllEquipment(y).ItemName & "," & AllEquipment(y).AmountInStock & "," & AllEquipment(y).Reorde
230     Next y
231
232     SaveAllEquipment.Close()
233 End Function
234
235 *****STAFF*****
236
237 'Declaring temporary variables for the bubble sort
238 Public TempStaff1 As String
239 Public TempStaff2 As String
240 Public TempStaff3 As String
241 Public TempStaff4 As String
242 Public TempStaff5 As String
243 Public TempStaff6 As String
244 Public TempStaff7 As String
245 Public TempStaff8 As String
246 Public TempStaff9 As String
247 Public TempStaff10 As String
248 Public TempStaff11 As String
249 Public TempStaff12 As String
250
251 'Declaring a structure for staff
252 1 reference
253 Public Structure Staff
254     Public StaffID As String
255     Public Title As String
256     Public Fname As String
257     Public Sname As String
258     Public FAddress As String
259     Public SAddress As String
260     Public Postcode As String
261     Public Town As String
262     Public Phone As String
263     Public Email As String
264     Public DoB As String
265     Public Speciality As String
266 End Structure
267
268 'Declaring an array for staff
269 Public AllStaff(0 To 99) As Staff
270 Public NumStaff As Integer = 0

```

270

13 references

Public Function Load\_Staff()

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

Do

line = file.ReadLine()

'Reads each readline and splits the data at each "," and puts it into the AllStaff array

parts = line.Split(",")

AllStaff(NumStaff).StaffID = parts(0)

AllStaff(NumStaff).Title = parts(1)

AllStaff(NumStaff).Fname = parts(2)

AllStaff(NumStaff).Sname = parts(3)

AllStaff(NumStaff).FAddress = parts(4)

AllStaff(NumStaff).SAddress = parts(5)

AllStaff(NumStaff).Postcode = parts(6)

AllStaff(NumStaff).Town = parts(7)

AllStaff(NumStaff).Phone = parts(8)

AllStaff(NumStaff).Email = parts(9)

AllStaff(NumStaff).DoB = parts(10)

AllStaff(NumStaff).Speciality = parts(11)

'Goes to the next equipment data

NumStaff = NumStaff + 1

Loop Until (file.EndOfStream)

file.Close()

End Function

307

308

1 reference

Public Function Bubble\_Sort\_Staff()

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

'Bubble sort by staff ID

For x = 0 To NumStaff - 2

For y = 0 To NumStaff - 2

If AllStaff(y).StaffID > AllStaff(y + 1).StaffID Then

TempStaff1 = AllStaff(y).StaffID

TempStaff2 = AllStaff(y).Title

TempStaff3 = AllStaff(y).Fname

TempStaff4 = AllStaff(y).Sname

TempStaff5 = AllStaff(y).FAddress

TempStaff6 = AllStaff(y).SAddress

TempStaff7 = AllStaff(y).Postcode

TempStaff8 = AllStaff(y).Town

TempStaff9 = AllStaff(y).Phone

TempStaff10 = AllStaff(y).Email

TempStaff11 = AllStaff(y).DoB

TempStaff12 = AllStaff(y).Speciality

AllStaff(y).StaffID = AllStaff(y + 1).StaffID

AllStaff(y).Title = AllStaff(y + 1).Title

AllStaff(y).Fname = AllStaff(y + 1).Fname

AllStaff(y).Sname = AllStaff(y + 1).Sname

AllStaff(y).FAddress = AllStaff(y + 1).FAddress

AllStaff(y).SAddress = AllStaff(y + 1).SAddress

AllStaff(y).Postcode = AllStaff(y + 1).Postcode

AllStaff(y).Town = AllStaff(y + 1).Town

AllStaff(y).Phone = AllStaff(y + 1).Phone

AllStaff(y).Email = AllStaff(y + 1).Email

AllStaff(y).DoB = AllStaff(y + 1).DoB

AllStaff(y).Speciality = AllStaff(y + 1).Speciality

AllStaff(y + 1).StaffID = TempStaff1

AllStaff(y + 1).Title = TempStaff2

AllStaff(y + 1).Fname = TempStaff3

AllStaff(y + 1).Sname = TempStaff4

AllStaff(y + 1).FAddress = TempStaff5

AllStaff(y + 1).SAddress = TempStaff6

AllStaff(y + 1).Postcode = TempStaff7

AllStaff(y + 1).Town = TempStaff8

AllStaff(y + 1).Phone = TempStaff9

AllStaff(y + 1).Email = TempStaff10

AllStaff(y + 1).DoB = TempStaff11

AllStaff(y + 1).Speciality = TempStaff12

End If

```

350     Next
351     Next
352
353     End Function
354
355     3 references
356     Public Function Save_All_Staff()
357         'Saving all the staff (i.e. for edits)
358         Dim SaveAllStaff As System.IO.StreamWriter
359         SaveAllStaff = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Staff.txt"), False)
360         For y = 0 To (NumStaff - 1)
361             SaveAllStaff.WriteLine(AllStaff(y).StaffID & "," & AllStaff(y).Title & "," & AllStaff(y).Fname & "," & AllStaff(y).Sname & "," & AllStaff(y).FAddre
362         Next y
363         SaveAllStaff.Close()
364
365     End Function
366
367     *****PASSWORDS*****
368
369     'Variable for program to know what staff type and ID of the logged on user
370     Public LoggedOnStaffType As String
371     Public LoggedOnStaffID As String
372
373     'Declaring temporary variables for passwords
374     Public TempPass1 As Integer
375     Public TempPass2 As String
376     Public TempPass3 As String
377     Public TempPass4 As String
378     Public TempPass5 As String
379
380     'Declaring a structure for password data
381     1 reference
382     Public Structure Password
383         Public PasswordID As Integer
384         Public StaffID As String
385         Public StaffType As String
386         Public Password As String
387         Public Clue As String
388
389     End Structure
390
391     'Declaring an array for passwords
392     Public AllPasswords(0 To 999) As Password
393     Public NumPasswords As Integer = 0
394
395     5 references
396     Public Function Load_Password()
397         NumPasswords = 0
398         Dim file As System.IO.StreamReader
399         Dim line As String
400         Dim parts(0 To 4) As String
401         'Loading the passwords textfile
402         file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Passwords.txt"))
403
404         Do
405             line = file.ReadLine()
406
407             'Reads each readline and splits the data at each "," and puts it into the AllPasswords array
408             parts = line.Split(",")
409             AllPasswords(NumPasswords).PasswordID = parts(0)
410             AllPasswords(NumPasswords).StaffID = parts(1)
411             AllPasswords(NumPasswords).StaffType = parts(2)
412             AllPasswords(NumPasswords).Password = parts(3)
413             AllPasswords(NumPasswords).Clue = parts(4)
414
415             'Goes to the next equipment data
416             NumPasswords = NumPasswords + 1
417         Loop Until (file.EndOfStream)
418
419         file.Close()
420
421     End Function
422
423     'No need to bubble sort passwords
424
425     2 references
426     Public Function Save_All_Passwords()

```

```

426
427 'Saving all the passwords (i.e) for new edits
428 Dim SaveAllPasswords As System.IO.StreamWriter
429 SaveAllPasswords = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Passwords.txt"), False)
430 For y = 0 To (NumPasswords - 1)
431     SaveAllPasswords.WriteLine(AllPasswords(y).PasswordID & "," & AllPasswords(y).StaffID & "," & AllPasswords(y).StaffType & "," & AllPasswords(y).Pas
432     Next y
433 SaveAllPasswords.Close()
434
435 End Function
-----
436
437
438 .....BOOKINGS.....
439
440 'Declare a structure for bookings
441 1 reference
442 Public Structure Booking
443     Public BookingID As String
444     Public JobType As String
445     Public BookingDate As String
446     Public CustomerID As String
447     Public StaffID As String
448     Public AllocatedTime As Double 'Double is a decimal number
449     Public Price As String
450     Public ToolsList As String
451     Public JobNotes As String
452     Public HasBeenInvoiced As Boolean
453 End Structure
-----
454 'Declaring an array for bookings
455 Public AllBookings(0 To 99999) As Booking
456 Public NumBookings As Integer = 0
457
458 11 references
459 Public Function Load_Bookings()
460
461     NumBookings = 0
462     Dim file As System.IO.StreamReader
463     Dim line As String
464     Dim parts(0 To 9) As String
-----
431 "& AllPasswords(y).StaffType & "," & AllPasswords(y).Password & "," & AllPasswords(y).Clue)
432
433
-----
465 'Loading the bookings textfile
466 file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Bookings.txt"))
467
468 Do
469     line = file.ReadLine()
470
471     'Reads each readline and splits the data at each "," and puts it into the AllBookings array
472     parts = line.Split(",")
473     AllBookings(NumBookings).BookingID = parts(0)
474     AllBookings(NumBookings).JobType = parts(1)
475     AllBookings(NumBookings).BookingDate = parts(2)
476     AllBookings(NumBookings).CustomerID = parts(3)
477     AllBookings(NumBookings).StaffID = parts(4)
478     AllBookings(NumBookings).AllocatedTime = parts(5)
479     AllBookings(NumBookings).Price = parts(6)
480     AllBookings(NumBookings).ToolsList = parts(7)
481     AllBookings(NumBookings).JobNotes = parts(8)
482     AllBookings(NumBookings).HasBeenInvoiced = parts(9)
483
484     'Goes to the next booking data
485     NumBookings = NumBookings + 1
486
487 Loop Until (file.EndOfStream)
488
489 file.Close()
490
491 End Function
-----
492
493 0 references
494 Public Function Insertion_Sort_Bookings() 'This is an insertion sort for bookings
495
496 Dim Element1, Element2, Element3, Element4, Element5, Element6, Element7, Element8, Element9, Element10 As String 'Variable to hold the "i" value of the "For" loop
497 Dim Position As Integer 'Variable to hold the value of the position within the array
498
499 For i = 1 To NumBookings - 1 'Must be "1 to" as we're initially comparing the second value to the first
500     'i is the pivot point
501     Element1 = AllBookings(i).BookingID
502     Element2 = AllBookings(i).JobType
503     Element3 = AllBookings(i).CustomerID
504     Element4 = AllBookings(i).StaffID
505     Element5 = AllBookings(i).AllocatedTime

```

```

505 Element6 = AllBookings(i).Price
506 Element7 = AllBookings(i).ToolsList
507 Element8 = AllBookings(i).JobNotes
508 Element9 = AllBookings(i).BookingDate
509 Element10 = AllBookings(i).HasBeenInvoiced
510 Position = i
511
512 While (Position > 0) And (AllBookings(Position - 1).BookingID > Element1)
513     'Swaps data of the two positions in the AllBookings array
514     AllBookings(Position).BookingID = AllBookings(Position - 1).BookingID
515     AllBookings(Position).BookingDate = AllBookings(Position - 1).BookingDate
516     AllBookings(Position).JobType = AllBookings(Position - 1).JobType
517     AllBookings(Position).CustomerID = AllBookings(Position - 1).CustomerID
518     AllBookings(Position).StaffID = AllBookings(Position - 1).StaffID
519     AllBookings(Position).AllocatedTime = AllBookings(Position - 1).AllocatedTime
520     AllBookings(Position).Price = AllBookings(Position - 1).Price
521     AllBookings(Position).ToolsList = AllBookings(Position - 1).ToolsList
522     AllBookings(Position).JobNotes = AllBookings(Position - 1).JobNotes
523     AllBookings(Position).HasBeenInvoiced = AllBookings(Position - 1).HasBeenInvoiced
524     Position = Position - 1 'Decrements Position
525
526     If Position = 0 Then
527         Exit While
528     End If
529
530 End While
531
532 'Insert the pivot point data back into the array
533 AllBookings(i).BookingID = Element1
534 AllBookings(i).JobType = Element2
535 AllBookings(i).CustomerID = Element3
536 AllBookings(i).StaffID = Element4
537 AllBookings(i).AllocatedTime = Element5
538 AllBookings(i).Price = Element6
539 AllBookings(i).ToolsList = Element7
540 AllBookings(i).JobNotes = Element8
541 AllBookings(i).BookingDate = Element9
542 AllBookings(i).HasBeenInvoiced = Element10
543
544 Next

```

```

545
546
547 End Function
548
3 references
549 Public Function Save_All_Bookings()
550
551     'Saving all the bookings (i.e) for edits
552     Dim SaveAllBookings As System.IO.StreamWriter
553     SaveAllBookings = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Bookings.txt"), False)
554     For y = 0 To (NumBookings - 1)
555         SaveAllBookings.WriteLine(AllBookings(y).BookingID & "," & AllBookings(y).JobType & "," & AllBookings(y).BookingDate & "," & AllBookings(y).Custo
556     Next y
557     SaveAllBookings.Close()
558
559 End Function
560
561 *****INVOICES*****
562
563 'Declare a structure for invoices
564 1 reference
565 Public Structure Invoice
566     Public InvoiceID As Integer
567     Public BookingID As String
568     Public BookingDate As String
569     Public Price As Double
570     Public Paid As Boolean
571 End Structure
572
573 'Declaring an array for invoices
574 Public AllInvoices(0 To 99999) As Invoice
575 Public NumInvoices As Integer = 0
576
6 references
577 Public Function Load_Invoices()
578
579     NumInvoices = 0
580     Dim file As System.IO.StreamReader
581     Dim line As String
582     Dim parts(0 To 4) As String
583
584     .CustomerID & "," & AllBookings(y).StaffID & "," & AllBookings(y).AllocatedTime & "," & AllBookings(y).Price & "," & AllBookings(y).ToolsLi
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

621 1 reference
622 Public Structure Payroll
623     Public StaffID As String 'Foreign key
624     Public RateOfPay As Double
625     Public HoursWorked As Double
626     Public GrossPay As Double
627     Public NI As Double
628     Public Tax As Double
629     Public Pension As Double
630     Public NetPay As Double
631 End Structure
632
633 'Declaring an array for payroll
634 Public AllPayroll(0 To 99) As Payroll 'Same as number of staff
635 Public NumPayroll As Integer = 0
636
637 2 references
638 Public Function Load_Payroll()
639
640     NumPayroll = 0
641     Dim file As System.IO.StreamReader
642     Dim line As String
643     Dim parts(0 To 7) As String
644
645     'Loading the bookings textfile
646     file = My.Computer.FileSystem.OpenTextFileReader(Dir$("Payroll_Masterfile_2022.txt"))
647
648     Do
649         line = file.ReadLine()
650
651         'Reads each readline and splits the data at each "," and puts it into the AllPayroll array
652         parts = line.Split(",")
653         AllPayroll(NumPayroll).StaffID = parts(0)
654         AllPayroll(NumPayroll).RateOfPay = parts(1)
655         AllPayroll(NumPayroll).HoursWorked = parts(2)
656         AllPayroll(NumPayroll).GrossPay = parts(3)
657         AllPayroll(NumPayroll).NI = parts(4)
658         AllPayroll(NumPayroll).Tax = parts(5)
659         AllPayroll(NumPayroll).Pension = parts(6)
660         AllPayroll(NumPayroll).NetPay = parts(7)
661
662         'Goes to the next payroll data
663         NumPayroll = NumPayroll + 1
664
665     Loop Until (file.EndOfStream)
666
667     file.Close()
668 End Function
669
670 1 reference
671 Public Function Save_All_Payroll()
672
673     'Saving all the payroll to the masterfile (i.e) for new monthly payrolls
674     Dim SaveAllPayroll As System.IO.StreamWriter
675     SaveAllPayroll = My.Computer.FileSystem.OpenTextFileWriter(Dir$("Payroll_Masterfile_2022.txt"), False)
676     For y = 0 To (NumPayroll - 1)
677         SaveAllPayroll.WriteLine(AllPayroll(y).StaffID & "," & AllPayroll(y).RateOfPay & "," & AllPayroll(y).HoursWorked & "," & AllPayroll(y).GrossPay & "," & AllPayr
678     Next y
679     SaveAllPayroll.Close()
680 End Function
681 End Module
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900

```