

Implementation

Forms

Form Name	Notes/Features used	Page Number
Welcome Menu	This is the first form that shows up when the system is ran. It displays a welcome message to the user, and then a button which is pressed to open the log in screen.	20
Log In	This form allows the user to enter their username and password in order to gain access to either the admin menu or staff menu, depending on their level of access. It checks that both the username and password are correct before allowing them to access the menus. There is a button on this form that the user can press if they have forgotten their password, and they will need to answer their personal security question correctly, and then they will be told their password. If they have forgotten their username, then they can press a button which tells them how the usernames are formatted, so they can then work theirs out.	21
Admin Menu	This is the form that is opened when an admin logs into the system. It contains buttons which lead to every function that they have access to. They will have access to more of the system's functions, as they have more authority than regular staff. They can add, edit, search, and delete data stored within the text files.	23
Staff Menu	This is the form that opens when a normal member of staff signs in to the system. The form contains many buttons that lead to the forms that the staff have access to, which is less forms than the admins can access.	26
Add Parent	This form allows the user to enter details for a new parent being added to the system. It validates the data being entered using a range of checks such as presence checks, length checks, and format checks. This data is then saved within a text file.	28
Edit/Delete Parent	This form will allow the user to make changes to the details saved for parents in the text file. They can use the 'previous' and 'next' buttons to view data for different parents, and they can use the save button to save the changes they make, after the new data has passed the necessary validation checks. There is also a 'Delete Parent' button on the bottom of the	31

	form which allows the user to delete the record of the Parent being displayed from the text file.	
View All Parents	This form simply allows the user to view a list of every parent saved within the text file using a rich text box. It lists all details of every parent in order of A-Z with their Surnames.	35
Search Parents	This form allows the user to input the First Name and Surname of a Parent saved within the text file, and if they match with someone in the file, it will display all of their details in a message box. If it does not match a parent in the text file, then a message will be displayed to the user indicating that a parent was not found.	34
Add Coach	This form allows the user to enter details for a new coach being added to the system. It validates the data being entered using a range of checks such as presence checks, length checks, and format checks. This data is then saved within a text file.	36
Edit/Delete Coach	This form will allow the user to make changes to the details saved for coaches in the text file. They can use the 'previous' and 'next' buttons to view data for different coaches, and they can use the save button to save the changes they make, after the new data has passed the necessary validation checks. There's also a 'Delete Coach' button on the bottom which allows the user to delete the record of a coach from the text file.	39
View All Coaches	This form simply allows the user to view a list of every coach saved within the text file using a rich text box. It lists all details of every coach in order of A-Z with their Surnames.	43
Search Coaches	This form allows the user to input the First Name and Surname of a Coach saved within the text file, and if they match with someone in the file, it will display all of their details in a message box. If it does not match a coach in the text file, then a message will be displayed to the user indicating that a coach was not found.	42
Add Players	This form allows the user to enter details for a new player being added to the system. It validates the data being entered using a range of checks such as presence checks, length checks, and format checks. This data is then saved within a text file.	44
Edit/Delete Players	This form will allow the user to make changes to the details saved for players in the text file. They can use the 'previous' and 'next' buttons to view data for different players, and they can use the save button to save the changes they	47

	make, after the new data has passed the necessary validation checks. There's also a 'Delete Player' button on the bottom of the form that allows the user to delete the record of a Player within the system.	
View All Players	This form simply allows the user to view a list of every player saved within the text file using a rich text box. It lists all details of every player in order of A-Z with their Surnames.	51
Search Players	This form allows the user to input the First Name and Surname of a Player saved within the text file, and if they match with someone in the file, it will display all of their details in a message box. If it does not match a player in the text file, then a message will be displayed to the user indicating that a player was not found.	50
Add Stock	This form allows the user to enter details for new stock being added to the system. It validates the data being entered using presence checks, and then stores it within a text file.	52
Edit/Delete Stock	This form will allow the user to make changes to the details saved for stock in the text file. They can use the 'previous' and 'next' buttons to view data for different items, and they can use the save button to save the changes they make, after the new data has passed the necessary validation checks. There's also a 'Delete Item' button on the bottom of the form, that allows the user to delete the record of any item saved within the text file.	54
View All Stock	This form simply allows the user to view a list of every stock item saved within the text file using a rich text box. It lists all details of every item in order of A-Z with their item description.	57
Search Stock	This form allows the user to input the description of an item saved within the text file, and if they match with an item in the file, it will display all of its details in a message box. If it does not match an item in the text file, then a message will be displayed to the user indicating that an item was not found.	56
Create New User	This form allows a system admin to enter details for a new user who can access the system. It validates the data being entered using a range of checks such as presence checks, length checks, and format checks. This data is then saved within a text file, and a username is generated for the user.	58
Edit/Delete User	This form will allow the user to make changes to the details saved for the users in the text file. They can use the 'previous' and 'next' buttons	60

	to view data for different users, and they can use the save button to save the changes they make, after the new data has passed the necessary validation checks. There is also the 'Delete User' button which lets the user delete the record of another user from the text file.	
View Age Group	This form allows the user to select an Age Group from a dropdown list, and it then displays the name, position, and their parents name and phone number of every player in that age group using a rich text box.	63
View Position	This form allows the user to select a Position from a dropdown list, and it then displays the name, position, and their parents name and phone number of every player in that age group using a rich text box.	65
View Player Contact Details	This form allows the user to enter the first name and surname of a player stored within the system, and it then displays the name, phone number and email of their parent using a message box.	67
View Subs Paid and Add Payments	This form allows the user to select an age group from the drop down list, and then displays the name of the players, their player id, the amount that they have paid, and the amount they have left to pay. Each year, the players have to pay £75 to play for the club, so this is used to calculate the amount left to pay. As well as this, the form also allows the user to add new payments for players. They must select an age group and then a player ID from the dropdown lists, and then the amount paid. This payment is then added to the system, and the amount paid and amount left to pay for this player will change accordingly.	68
View An Age Group's Coach	This form will allow the user to view the records held within the system for a coach who takes charge of a certain age group. They select an age group from the dropdown list at the top of the form, and then the data for that group's coach is displayed in a text box.	72

I have used the following features:

Feature	Description/Use	Page Number
Linear Search	I used a linear search on my forms which allows the user to search for the data stored for a specific name of a player, parent, coach or stock item. It asked them to enter the name of the person/item, and it then searches through the array one by one until it finds that record. If the person/item is not found, then an error message is displayed.	34, 42, 50, 56, 67
Bubble Sort	I used a bubble sort when storing the data of parents, players, coaches and stock items into their own text files. This let me store them in alphabetical order using the surnames of the parents players and coaches, and using the name of the stock item.	Implemented – 30, 38, 45, 53 Module Code – 92, 93, 94, 95, 96, 97
Length Check	I used a length check when a user inputted a postcode, password or phone number into the system. If the postcode was not at least 7 characters, the password was not at least 8 and the phone number was not exactly 11 digits, then this presence check would stop that data from being saved into the text files.	29, 37, 59, 62
Presence Check	I used a presence check on most data inputted into the system such as first names and surnames, street names, towns and cities etc. This stopped users from leaving textboxes blank.	21, 29, 37, 45, 52, 53, 59, 61, 62
Format Check	I used a format check when a user inputted either an email or a date of birth into the system. If the email didn't follow the format *@*.*, then this presence check would output an error message, along with the DoB format **/**/****.	29, 37, 45, 59, 62
Error Catching	To deal with the wrong data types being added to the system, I used the error catching code. I ran into an issue when a date of birth was entered into the system which did not follow the format of a date, so the system crashed. To stop this, I used 'Try' and 'Catch' code, and this problem was solved.	28, 36, 44, 58, 60, 61

Aim and Objectives Coded

Original aim/objective	Successfully implemented? (Yes/No)
To allow the user to store details of parents, players, coaches and subs fees with ease	Yes
To ensure that data stored within the system can be easily edited and records can be deleted	Yes
Implement a good search system which allows users to quickly access the record of an individual using a key search term	Yes
Keep the data secure, with different levels of access for different users	Yes
Ensure that the system's interface is easy to maneuver around, and that the font, text size and color scheme does not distract	Yes
The user must be able to quickly access contact details for a player's parent, as their child may be in an accident or have an injury	Yes
Allow the user to view all players from a certain position, as group training sessions are often organized, focusing on certain fractions of the field	Yes
Allow the user to view all players from a certain age group	Yes

Implementation of Relational Database

In my system, I implemented a 3rd Normalised Form Relational Database when I did my Players Data. I allowed the user to link a player to their parent using the primary key in the parents database, which is Parent ID. Doing so made Parent ID a Foreign Key in the Players database.

The 'Players' Database is:

- PlayerID (Primary Key)
- Firstname
- Surname
- DoB
- AgeGroup
- Position
- ParentID (Foreign Key)



All_Players - Notepad

File Edit Format View Help

```
PLY5,Adam,Adams,09/10/2004,U18,Midfielder,PNT5  
PLY12,Alex,Adams,12/09/2005,U17,Forward,PNT5  
PLY26,Alex,Alan,28/07/2010,U12,Goalkeeper,PNT26  
PLY23,Neil,Amond,18/11/2011,U11,Defender,PNT23  
PLY24,Benny,Brown,14/09/2011,U11,Forward,PNT24  
PLY31,Adrian,Chiles,12/11/2009,U13,Midfielder,PNT31  
PLY28,Yacob,Clark,16/11/2013,U9,Goalkeeper,PNT28  
PLY3,Sean,Davies,09/03/2008,U14,Goalkeeper,PNT3
```

Data Structures

Structure Name	Description of use
AllParents	Holds a parents data. Primary key is ParentID
AllCoaches	Holds a coaches data. Primary key is CoachID
AllPlayers	Holds a players data. Primary key is PlayerID
AllStock	Holds the stock data. Primary key is StockID
AllUsers	Holds the users data. Primary key is username
AllSubs	Holds the subs data for every player

User Defined routines

Routine Name	Function/Description
LoadSubPayments	Loads all subs payments from AllSubs array
SaveSubs	Saves subs records into SubsRecord text file
LoadUsers	Loads all user records from AllUsers array
SaveAllUsers	Saves user records into All_Users textfile
LoadParents	Loads all parent records from AllParents array
LoadPlayers	Loads all player records from AllPlayers array
LoadCoaches	Loads all coach records from AllCoaches array
LoadStock	Loads all stock records from AllStock array
SaveAllParents	Saves parents records into All_Parents textfile
SaveAllCoaches	Saves coaches records into All_Coaches textfile
SaveAllPlayers	Saves players records into All_Players textfile
SaveAllStock	Saves stock records into All_Stock Textfile
BubblesortCoach	Used to sort all data within All_Coaches textfile into alphabetical order depending on surname
BubblesortParents	Used to sort all data within All_Parents textfile into alphabetical order depending on surname
BubblesortPlayers	Used to sort all data within All_Players textfile into alphabetical order depending on surname
BubblesortStock	Used to sort all data within All_Stock textfile into alphabetical order depending on surname

Variable List

Module/Form Name	Variable name	Description
Welcome Screen	openLogIn	Used to link the log in button on the welcome screen to the log in form
Log In	moveFormAdmin	Used to direct an admin to the admin menu when they sign in
Log In	moveFormStaff	Used to direct a staff member to the staff menu when they sign in
Log In	Username	Variable which is linked to the username textbox on the log in form
Log In	Password	Variable linked to the password textbox on the log in form
Log In	Username2	Variable linked to the enter username input box shown when forgot password is pressed
Log In	Answer	Variable linked to the security question answer given when forgot password is pressed
Admin Menu	admAddParent	Used to open add parent form when the button is pressed on the admin menu
Admin Menu	admEditParent	Used to open edit parent form when the button is pressed on the admin menu
Admin Menu	admSearchParent	Used to open search parent form when the button is pressed on the admin menu
Admin Menu	admViewAllParents	Used to view all parent form when the button is pressed on the admin menu
Admin Menu	admAddCoach	Used to open add coach form when the button is pressed on the admin menu
Admin Menu	admSearchCoach	Used to open search coach form when the button is pressed on the admin menu
Admin Menu	admEditCoach	Used to open edit coach form when the button is pressed on the admin menu
Admin Menu	admViewAllCoaches	Used to open view all coaches form when the button is pressed on the admin menu

Admin Menu	admAddPlayer	Used to open add player form when the button is pressed on the admin menu
Admin Menu	admEditPlayer	Used to open edit player form when the button is pressed on the admin menu
Admin Menu	admSearchPlayer	Used to open search player form when the button is pressed on the admin menu
Admin Menu	admViewAllPlayers	Used to open view all players form when the button is pressed on the admin menu
Admin Menu	admAddStock	Used to open add stock form when the button is pressed on the admin menu
Admin Menu	admEditStock	Used to open edit stock form when the button is pressed on the admin menu
Admin Menu	admSearchStock	Used to open search stock form when the button is pressed on the admin menu
Admin Menu	admViewAllStock	Used to open view all stock form when the button is pressed on the admin menu
Admin Menu	admViewPosition	Used to open view position form when the button is pressed on the admin menu
Admin Menu	admViewContact	Used to open view player's contact details form when the button is pressed on the admin menu
Admin Menu	admAddUser	Used to open create new user form when the button is pressed on the admin menu
Admin Menu	admViewAgeGroup	Used to open view age group form when the button is pressed on the admin menu
Admin Menu	admViewAddSubs	Used to open view subs/add subs form when the button is pressed on the admin menu
Admin Menu	admEditUser	Used to open edit user form when the button is pressed on the admin menu
Admin Menu	admViewAgeGroupCoach	Used to open view an age group's coach form when the button is pressed on the admin menu

Staff Menu	stfSearchParent	Used to open search parent form when the button is pressed on the staff menu
Staff Menu	stfAddParent	Used to open add parent form when the button is pressed on the staff menu
Staff Menu	stfViewAddSubs	Used to open view subs/add subs form when the button is pressed on the staff menu
Staff Menu	stfSearchStock	Used to open search stock form when the button is pressed on the staff menu
Staff Menu	stfAddStock	Used to open add stock form when the button is pressed on the staff menu
Staff Menu	stfSearchPlayer	Used to open search player form when the button is pressed on the staff menu
Staff Menu	stfAddPlayer	Used to open add player form when the button is pressed on the staff menu
Staff Menu	stfSearchCoach	Used to open search coach form when the button is pressed on the staff menu
Staff Menu	stfAddCoach	Used to open add coach form when the button is pressed on the staff menu
Staff Menu	stfViewAgeGroup	Used to open view age group form when the button is pressed on the staff menu
Staff Menu	stfViewContact	Used to open view player contact details form when the button is pressed on the staff menu
Staff Menu	stfViewPosition	Used to open view position form when the button is pressed on the staff menu
Staff Menu	stfViewAgeGroupCoach	Used to open view an age group's coach form when the button is pressed on the staff menu
Add Parent	ParentID	Variable linked to parentID textbox on add parent form
Add Parent	FirstName	Variable linked to firstname textbox on add parent form
Add Parent	Surname	Variable linked to surname textbox on add parent form
Add Parent	DoB	Variable linked to dob textbox on add parent form

Add Parent	Postcode	Variable linked to postcode textbox on add parent form
Add Parent	Street	Variable linked to street textbox on add parent form
Add Parent	Town	Variable linked to town textbox on add parent form
Add Parent	City	Variable linked to city textbox on add parent form
Add Parent	PhoneNum	Variable linked to phone number textbox on add parent form
Add Parent	Email	Variable linked to email textbox on add parent form
Add Parent	Correct	Variable used during format check on date of birth on add parent form
Add Parent	Correct2	Variable used during format check on email on add parent form
Add Parent	NewParent	Used when saving data entered on add parent form to the text file
Search Parent	File	Linked to the text file all parents in search parents form
Search Parent	Parts	Linked to the commas in-between fields in the text file all parents
Search Parent	Line	Linked to the different lines of records in the text file all parents
Search Parent	SearchParentFName	Linked to enter parent first name textbox on search parent form
Search Parent	SearchParentSurname	Linked to enter parent surname textbox on search parent form
Add Player	PlayerID	Variable linked to playerid textbox on add player form
Add Player	FirstName	Variable linked to firstname textbox on add player form
Add Player	Surname	Variable linked to surname textbox on add player form
Add Player	DoB	Variable linked to dob textbox on add player form
Add Player	AgeGroup	Variable linked to agegroup textbox on add player form
Add Player	Position	Variable linked to position textbox on add player form
Add Player	ParentID	Variable linked to parentid textbox on add player form
Add Player	NewPlayer	Variable used when saving data from add player form to text file
Add Player	Correct	Variable used during format check on date of birth of player

Add Player	NewPlayerSubs	Variable used to add player to the SubsPaid file
Search Player	File	Linked to the text file all players in search parents form
Search Player	Parts	Linked to the commas in-between fields in the text file all players
Search Player	Line	Linked to the different lines of records in the text file all players
Search Player	SearchPlayerFName	Linked to enter player first name textbox on search player form
Search Player	SearchPlayerSurname	Linked to enter player surname textbox on search player form
Add Coach	CoachID	Variable linked to coachid textbox on add coach form
Add Coach	FirstName	Variable linked to firstname textbox on add coach form
Add Coach	Surname	Variable linked to surname textbox on add coach form
Add Coach	DoB	Variable linked to dob textbox on add coach form
Add Coach	Postcode	Variable linked to postcode textbox on add coach form
Add Coach	Street	Variable linked to street textbox on add coach form
Add Coach	Town	Variable linked to town textbox on add coach form
Add Coach	City	Variable linked to city textbox on add coach form
Add Coach	PhoneNum	Variable linked to phone number textbox on add coach form
Add Coach	Email	Variable linked to email textbox on add coach form
Add Coach	AgeGroupCoached	Variable linked to age group coached textbox on add coach form
Add Coach	Correct	Variable used during format check of date of birth on add coach form
Add Coach	Correct2	Variable used during format check of email on add coach form
Add Coach	NewCoach	Variable used when saving data from add coach form to text file
Search Coach	File	Linked to the text file all coaches in search coach form
Search Coach	Parts	Linked to the commas in-between fields in the text file all coaches
Search Coach	Line	Linked to the different lines of records in the text file all coaches

Search Coach	SearchCoachFName	Linked to enter coach first name textbox on search coach form
Search Coach	SearchCoachSurname	Linked to enter coach surname textbox on search coach form
Add Stock	Description	Variable linked to description textbox on add stock form
Add Stock	Quantity	Variable linked to quantity textbox on add stock form
Add Stock	ExtraNotes	Variable linked to extranotes textbox on add stock form
Add Stock	NewStock	Variable used when saving data from add stock form to text file
Search Stock	File	Linked to the text file all stock in search stock form
Search Stock	Parts	Linked to the commas in-between fields in the text file all stock
Search Stock	Line	Linked to the different lines of records in the text file all stock
Search Stock	SearchStockName	Linked to enter item name textbox on search stock form
Create New User	Correct	Used during format check on date of birth in create new user form
Create New User	AddUser2	Used when adding new user entered into text file
View Subs Paid and Add Payments	File9	Used to open subsdueperyear textfile in view/add subs form
View Subs Paid and Add Payments	File	Used to open subsrecords textfile in view/add subs form
View Subs Paid and Add Payments	Line	Linked to the commas in-between fields in the text file subsrecords
View Subs Paid and Add Payments	Parts	Linked to the different lines of records in the text file subsrecords
View Subs Paid and Add Payments	PlayerId	Linked to the player id's of all players when finding subs in that age group and loading them into foundsubsinfo
View Subs Paid and Add Payments	AmountPAID	Linked to the amount of subs that a player has paid so far
View Subs Paid and Add Payments	FoundSubsInfo	Linked to the textfile foundsubsinfo

View Subs Paid and Add Payments	FoundSubs	Linked to the textfile foundsubsinfo when an age group has been selected
View Subs Paid and Add Payments	Playerid	Linked to the player id selected when making a payment
View Subs Paid and Add Payments	AmountPaid	Linked to the amount paid when making a payment
View Age Group	SelectedAgeGroup	linked to the age group that the user selects from the dropdown list
View Age Group	FoundPlayers	Linked to the players from the age group that the user selected
View An Age Group's Coach	SelectedAgeGroupCoach	Linked to the age group the user chose from the dropdown list
View An Age Group's Coach	FoundCoach	Linked to the coach that was found who coaches the age group chosen
View Player Contact Details	SearchPlayerFName	Linked to enter player first name textbox on view player contact details form
View Player Contact Details	SearchPlayerSurname	Linked to enter player surname textbox on view player contact details form
View Player Contact Details	SearchParentID	Linked to message box displaying error message when player isn't found
View Player Contact Details	PlayerName	Used to link the first name and surname of the player
View Position	SelectedPosition	Linked to the position that the user selects from the dropdown list
View Position	FoundPositionPlayers	Linked to the players from the position the user selected
Module	File	Used to open subsrecords textfile in LoadSubPayments() function
Module	Line	Linked to the commas in-between fields in the text file subsrecords in LoadSubPayments() function
Module	Parts	Linked to the different lines of records in the text file subsrecords in the LoadSubPayments() function
Module	SaveSubPaymentFile	Linked to the SubsRecords textfile in SaveSubs() function
Module	File	Used to open allusers textfile in loadusers() function

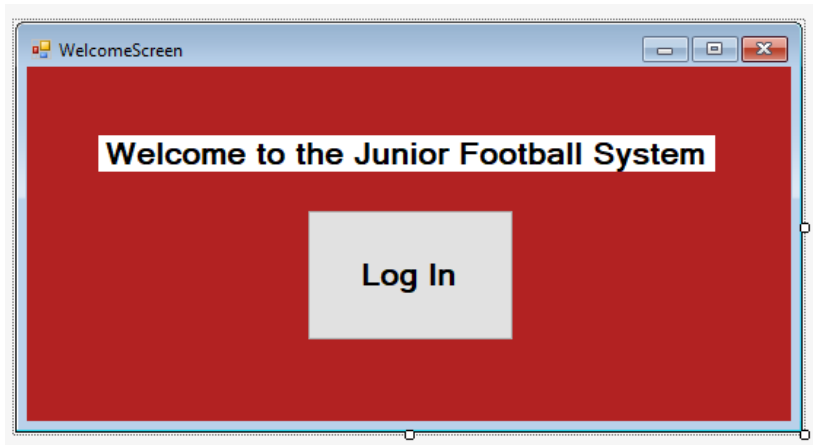
Module	Line	Linked to the commas in-between fields in the text file allusers in loadusers() function
Module	Parts	Linked to the different lines of records in the text file allusers in the loadusers() function
Module	SaveUserFile	Linked to the allusers textfile in saveallusers() function
Module	File	Used to open allparents textfile in loadparents() function
Module	Line	Linked to the commas in-between fields in the text file allparents in loadparents() function
Module	Parts	Linked to the different lines of records in the text file allparents in the loadparents() function
Module	Position	Used to define every field in the textfile allparents in loadparents()
Module	File2	Used to open allplayers textfile in loadplayers() function
Module	Line2	Linked to the commas in-between fields in the text file allplayers in loadplayers() function
Module	Parts2	Linked to the different lines of records in the text file allplayers in the loadplayers() function
Module	Position2	Used to define every field in the textfile allplayers in loadplayers()
Module	File3	Used to open allcoaches textfile in loadcoaches() function
Module	Line3	Linked to the commas in-between fields in the text file allcoaches in loadcoaches() function
Module	Parts3	Linked to the different lines of records in the text file allcoaches in the loadcoaches() function
Module	Position3	Used to define every field in the textfile allcoaches in loadcoaches()
Module	File4	Used to open allstock textfile in loadstock() function
Module	Line4	Linked to the commas in-between fields in the text file allstock in loadstock() function
Module	Parts4	Linked to the different lines of records in the text file allstock in the loadstock() function

Module	Position4	Used to define every field in the textfile allstock in loadstock()
Module	SaveParentFile	Linked to the allparents textfile in the SaveAllParents() function
Module	SaveCoachFile	Linked to the allcoaches textfile in the SaveAllCoaches() function
Module	SavePlayerFile	Linked to the allplayers textfile in the SaveAllPlayers() function
Module	SaveStockFile	Linked to the allstock textfile in SaveAllStock() function
Module	Temp1	Linked to CoachID in BubblesortCoach()
Module	Temp2	Linked to firstname in BubblesortCoach()
Module	Temp3	Linked to surname in BubblesortCoach()
Module	Temp4	Linked to Dob in BubblesortCoach()
Module	Temp5	Linked to postcode in BubblesortCoach()
Module	Temp6	Linked to street in BubblesortCoach()
Module	Temp7	Linked to town in BubblesortCoach()
Module	Temp8	Linked to city in BubblesortCoach()
Module	Temp9	Linked to phoneno in BubblesortCoach()
Module	Temp10	Linked to email in BubblesortCoach()
Module	Temp11	Linked to agegroupcoached in BubblesortCoach()
Module	Temp1	Linked to ParentID in BubblesortParents()
Module	Temp2	Linked to firstname in BubblesortParents()
Module	Temp3	Linked to surname in BubblesortParents()
Module	Temp4	Linked to dob in BubblesortParents()
Module	Temp5	Linked to postcode in BubblesortParents()
Module	Temp6	Linked to street in BubblesortParents()
Module	Temp7	Linked to town in BubblesortParents()
Module	Temp8	Linked to city in BubblesortParents()

Module	Temp9	Linked to phonenum in BubblesortParents()
Module	Temp10	Linked to email in BubblesortParents()
Module	Temp1	Linked to playerid in BubblesortPlayers()
Module	Temp2	Linked to firstname in BubblesortPlayers()
Module	Temp3	Linked to surname in BubblesortPlayers()
Module	Temp4	Linked to dob in BubblesortPlayers()
Module	Temp5	Linked to agegroup in BubblesortPlayers()
Module	Temp6	Linked to position in BubblesortPlayers()
Module	Temp7	Linked to parentid in BubblesortPlayers()
Module	Temp1	Linked to StockID in BubblesortStock()
Module	Temp2	Linked to description in BubblesortStock()
Module	Temp3	Linked to quantity in BubblesortStock()
Module	Temp4	Linked to extranotes in BubblesortStock()

All of my forms are suitable for the user to operate because they are very easy to follow due to the well-labelled buttons, simple colour scheme which is easy on the eye, and also suitable font sizes.

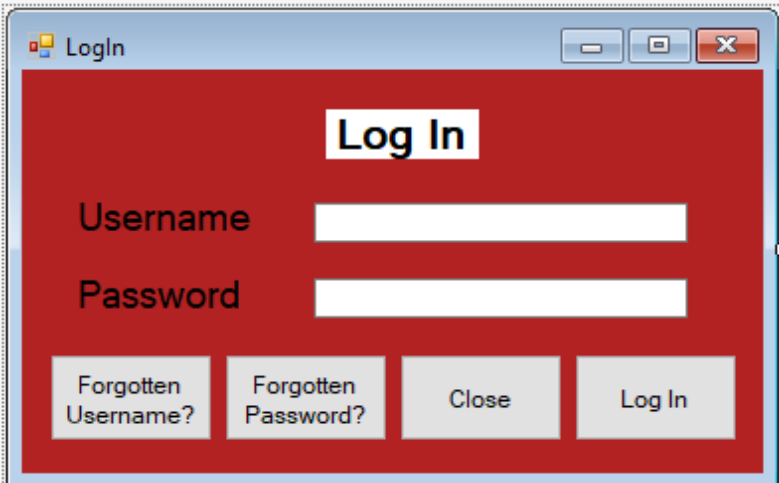
Form Name: Welcome Screen



Code Listing:

```
1 Public Class WelcomeScreen
2     Private Sub butWelcomeLogIn_Click(sender As Object, e As EventArgs) Handles butWelcomeLogIn.Click
3
4         Dim openLogIn As New LogIn
5         openLogIn.Show()
6
7     End Sub
8 End Class
```

Form Name: Log In



The screenshot shows a Windows Form titled "Log In". The form has a red background and a white title bar. It contains a "Log In" label at the top center, followed by "Username" and "Password" labels with corresponding text input fields. At the bottom, there are four buttons: "Forgotten Username?", "Forgotten Password?", "Close", and "Log In".

Code Listing:

```
LogIn.vb* x LogIn.vb [Design]* Coursework System
Coursework System butLogInClose Click
- references
1 Public Class LogIn
2 Private Sub butLogInLogIn_Click(sender As Object, e As EventArgs) Handles butLogInLogIn.Click
3
4 Dim moveFormAdmin As New AdminMenu
5 Dim moveFormStaff As New StaffMenu
6
7 LoadUsers()
8
9 Dim Username As String = txtLogInUsername.Text
10 Dim Password As String = txtLogInPassword.Text
11
12 'Presence Checks
13
14 If Username = "" Then
15     MsgBox("You must enter a username, please try again.")
16 End If
17
18 If Password = "" Then
19     MsgBox("You must enter a password, please try again.")
20 End If
21
22 For x = 0 To NumUsers - 1
23     If Username = AllUsers(x).Username And Password = AllUsers(x).Password Then
24         MsgBox("welcome to the Junior Football System")
25         If AllUsers(x).AccessLevel = "Admin" Then
26             moveFormAdmin.Show()
27         Else
28             moveFormStaff.Show()
29         End If

```

```

LogIn.vb*  X LogIn.vb [Design]*  Coursework System
Coursework System  butLogInClose  Click
30      Exit Sub
31      Else
32          MsgBox("Username and/or Password is incorrect, please try again.")
33          Exit Sub
34      End If
35
36
37
38
39      Next
40
41  End Sub
42
References
43  Private Sub butLogInForgotPassword_Click(sender As Object, e As EventArgs) Handles butLogInForgotPassword
44
45      LoadUsers()
46
47      Dim Username2 As String = InputBox("Enter Username")
48
49      For x = 0 To NumUsers - 1
50          If Username2 = AllUsers(x).Username Then
51              Dim Answer As String = InputBox(AllUsers(x).SecurityQuestion & " " & "(not case sensitive)")
52              If Answer.ToUpper = AllUsers(x).SecurityAnswer.ToUpper Then
53                  MsgBox("Your Password is :" & AllUsers(x).Password)
54              Else
55                  MsgBox("Your Answer to the Security Question is incorrect")
56              End If
57          End If
58      Next
59  End Sub
60
References
61  Private Sub butLogInForgotUsername_Click(sender As Object, e As EventArgs) Handles butLogInForgotUsername
62      MsgBox("Your Username is generated using your Surname, the first letter of your firstname, and the fi
63  End Sub
64  Private Sub butLogInClose_Click(sender As Object, e As EventArgs) Handles butLogInClose.Click
65      Me.Close()
66  End Sub
67  End Class

```

Form Name: Admin Menu



Code Listing:

```
AdminMenu.vb*  AdminMenu.vb [Design]*  Module1.vb*  LogIn.vb  LogIn.vb [Design]  WelcomeScreen.vb  WelcomeScreen.vb [Design]
Coursework System  (AdminMenu Events)  Load

2 references
1  Public Class AdminMenu
2  0 references
3  Private Sub butAdminMenuAddParent_Click(sender As Object, e As EventArgs) Handles butAdminMenuAddParent.Click
4  Dim admAddParent As New Form1
5  admAddParent.Show()
6  End Sub

7  0 references
8  Private Sub butAdminMenuEditParent_Click(sender As Object, e As EventArgs) Handles butAdminMenuEditParent.Click
9  Dim admEditParent As New EditParents
10 admEditParent.Show()
11 End Sub

12 0 references
13 Private Sub butAdminMenuSearchParent_Click(sender As Object, e As EventArgs) Handles butAdminMenuSearchParent.Click
14 Dim admSearchParent As New SearchParents
15 admSearchParent.Show()
16 End Sub

17 0 references
18 Private Sub butAdminMenuViewAllParents_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAllParents.Click
19 Dim admViewAllParents As New ViewAllParents
20 admViewAllParents.Show()
21 End Sub

22 0 references
23 Private Sub butAdminMenuAddCoach_Click(sender As Object, e As EventArgs) Handles butAdminMenuAddCoach.Click
24 Dim admAddCoach As New AddCoach
25 admAddCoach.Show()
26 End Sub
```

```

AdminMenu.vb* -> X AdminMenu.vb [Design]* Module1.vb* LogIn.vb LogIn.vb [Design] WelcomeScreen.vb WelcomeScreen.vb [Design]
Coursework System (AdminMenu Events) Load
27 Private Sub butAdminMenuEditCoach_Click(sender As Object, e As EventArgs) Handles butAdminMenuEditCoach.Click
28 Dim admEditCoach As New EditCoach
29 admEditCoach.Show()
30 End Sub
31
32 Private Sub butAdminMenuSearchCoach_Click(sender As Object, e As EventArgs) Handles butAdminMenuSearchCoach.Click
33 Dim admSearchCoach As New SearchCoaches
34 admSearchCoach.Show()
35 End Sub
36
37 Private Sub butAdminMenuViewAllCoaches_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAllCoaches.Click
38 Dim admViewAllCoaches As New ViewAllCoaches
39 admViewAllCoaches.Show()
40 End Sub
41
42 Private Sub butAdminMenuAddPlayer_Click(sender As Object, e As EventArgs) Handles butAdminMenuAddPlayer.Click
43 Dim admAddPlayer As New AddPlayer
44 admAddPlayer.Show()
45 End Sub
46
47 Private Sub butAdminMenuEditPlayer_Click(sender As Object, e As EventArgs) Handles butAdminMenuEditPlayer.Click
48 Dim admEditPlayer As New EditPlayer
49 admEditPlayer.Show()
50 End Sub
51

```

```

AdminMenu.vb* -> X AdminMenu.vb [Design]* Module1.vb* LogIn.vb LogIn.vb [Design] WelcomeScreen.vb WelcomeScreen.vb [Design]
Coursework System (AdminMenu Events) Load
52 Private Sub butAdminMenuSearchPlayer_Click(sender As Object, e As EventArgs) Handles butAdminMenuSearchPlayer.Click
53 Dim admSearchPlayer As New SearchPlayers
54 admSearchPlayer.Show()
55 End Sub
56
57 Private Sub butAdminViewAllPlayers_Click(sender As Object, e As EventArgs) Handles butAdminViewAllPlayers.Click
58 Dim admViewAllPlayers As New ViewAllPlayers
59 admViewAllPlayers.Show()
60 End Sub
61
62 Private Sub butAdminMenuAddStock_Click(sender As Object, e As EventArgs) Handles butAdminMenuAddStock.Click
63 Dim admAddStock As New AddStock
64 admAddStock.Show()
65 End Sub
66
67 Private Sub butAdminMenuEditStock_Click(sender As Object, e As EventArgs) Handles butAdminMenuEditStock.Click
68 Dim admEditStock As New EditStock
69 admEditStock.Show()
70 End Sub
71
72 Private Sub butAdminMenuSearchStock_Click(sender As Object, e As EventArgs) Handles butAdminMenuSearchStock.Click
73 Dim admSearchStock As New SearchStock
74 admSearchStock.Show()
75 End Sub
76

```

```

AdminMenu.vb* -> X AdminMenu.vb [Design]* Module1.vb* LogIn.vb LogIn.vb [Design] WelcomeScreen.vb WelcomeScreen.vb [Design]
Coursework System (AdminMenu Events) Load
77 Private Sub butAdminMenuViewAllStock_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAllStock.Click
78 Dim admViewAllStock As New ViewAllStock
79 admViewAllStock.Show()
80 End Sub
81
82 Private Sub butAdminMenuViewPosition_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewPosition.Click
83 Dim admViewPosition As New ViewPosition
84 admViewPosition.Show()
85 End Sub
86
87 Private Sub butAdminMenuViewPlayerContactDetails_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewPlayerContactDet
88 Dim admViewContact As New ViewPlayerContactDetails
89 admViewContact.Show()
90 End Sub
91
92 Private Sub butAdminMenuCreateUser_Click(sender As Object, e As EventArgs) Handles butAdminMenuCreateUser.Click
93 Dim admAddUser As New CreateNewUser
94 admAddUser.Show()
95 End Sub
96
97 Private Sub butAdminMenuViewAgeGroup_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAgeGroup.Click
98 Dim admViewAgeGroup As New ViewAgeGroup
99 admViewAgeGroup.Show()
100 End Sub
101

```

```
0 references
3 Private Sub butAdminMenuViewAddSubs_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAddSubs.Click
  Dim admViewAddSubs As New View_subs_Paid_and_Add_Payments
  admViewAddSubs.Show()
End Sub

0 references
3 Private Sub butAdminMenuEditUser_Click(sender As Object, e As EventArgs) Handles butAdminMenuEditUser.Click
  Dim admEditUser As New EditUser
  admEditUser.Show()
End Sub

0 references
3 Private Sub butAdminMenuViewAgeGroupCoach_Click(sender As Object, e As EventArgs) Handles butAdminMenuViewAgeGroupCoach.Click
  Dim admViewAgeGroupCoach As New ViewAgeGroupCoach
  admViewAgeGroupCoach.Show()
End Sub
```

Form Name: Staff Menu



Code Listing:

```
StaffMenu.vb x StaffMenu.vb [Design] AdminMenu.vb* AdminMenu.vb [Design]* Module1.vb* LogIn.vb LogIn.vb [Design]
VB Coursework System (StaffMenu Events) Load
2 references
1 Public Class StaffMenu
0 references
2 Private Sub butStaffMenuSearchParent_Click(sender As Object, e As EventArgs) Handles butStaffMenuSearchParent.Click
3 Dim stfSearchParent As New SearchParents
4 stfSearchParent.Show()
5 End Sub
6
0 references
7 Private Sub butStaffMenuAddParent_Click(sender As Object, e As EventArgs) Handles butStaffMenuAddParent.Click
8 Dim stfAddParent As New Form1
9 stfAddParent.Show()
10 End Sub
11
0 references
12 Private Sub butStaffMenuViewAddSubs_Click(sender As Object, e As EventArgs) Handles butStaffMenuViewAddSubs.Click
13 Dim stfViewAddSubs As New View_subs_Paid_and_Add_Payments
14 stfViewAddSubs.Show()
15 End Sub
16
0 references
17 Private Sub butStaffMenuSearchCoach_Click(sender As Object, e As EventArgs) Handles butStaffMenuSearchCoach.Click
18 Dim stfSearchCoach As New SearchCoaches
19 stfSearchCoach.Show()
20 End Sub
21
0 references
22 Private Sub butStaffMenuAddCoach_Click(sender As Object, e As EventArgs) Handles butStaffMenuAddCoach.Click
23 Dim stfAddCoach As New AddCoach
24 stfAddCoach.Show()
25 End Sub
26
```

```

StaffMenu.vb [Design] | AdminMenu.vb* | AdminMenu.vb [Design]* | Module1.vb* | LogIn.vb | LogIn.vb [Design]
Coursework System | (StaffMenu Events) | Load
26
27 | 0 references
28 | Private Sub butStaffMenuViewAgeGroup_Click(sender As Object, e As EventArgs) Handles butStaffMenuViewAgeGroup.Click
29 |     Dim stfViewAgeGroup As New ViewAgeGroup
30 |     stfViewAgeGroup.Show()
31 | End Sub
32 | 0 references
33 | Private Sub butStaffMenuSearchPlayer_Click(sender As Object, e As EventArgs) Handles butStaffMenuSearchPlayer.Click
34 |     Dim stfSearchPlayer As New SearchPlayers
35 |     stfSearchPlayer.Show()
36 | End Sub
37 | 0 references
38 | Private Sub butStaffMenuAddPlayer_Click(sender As Object, e As EventArgs) Handles butStaffMenuAddPlayer.Click
39 |     Dim stfAddPlayer As New AddPlayer
40 |     stfAddPlayer.Show()
41 | End Sub
42 | 0 references
43 | Private Sub butStaffMenuViewPlayerContactDetails_Click(sender As Object, e As EventArgs) Handles butStaffMenuViewPlayerContactDetails.Click
44 |     Dim stfViewContact As New ViewPlayerContactDetails
45 |     stfViewContact.Show()
46 | End Sub
47 | 0 references
48 | Private Sub butStaffMenuSearchStock_Click(sender As Object, e As EventArgs) Handles butStaffMenuSearchStock.Click
49 |     Dim stfSearchStock As New SearchStock
50 |     stfSearchStock.Show()
51 | End Sub

0 references
Private Sub butStaffMenuAddStock_Click(sender As Object, e As EventArgs) Handles butStaffMenuAddStock.Click
    Dim stfAddStock As New AddStock
    stfAddStock.Show()
End Sub

0 references
Private Sub butStaffMenuViewPosition_Click(sender As Object, e As EventArgs) Handles butStaffMenuViewPosition.Click
    Dim stfViewPosition As New ViewPosition
    stfViewPosition.Show()
End Sub

0 references
Private Sub butStaffMenuViewAgeGroupCoach_Click(sender As Object, e As EventArgs) Handles butStaffMenuViewAgeGroupCoach.Click
    Dim stfViewAgeGroupCoach As New ViewAgeGroupCoach
    stfViewAgeGroupCoach.Show()
End Sub
End Class

```

Form Name: Add Parent



The image shows a Windows form window titled "Form1" with a red background. At the top center, there is a white box containing the text "Add Parent". Below this, there are nine text input fields, each with a label to its left: "First Name", "Surname", "Date of Birth", "Postcode", "Street", "Town", "City", "Phone Num", and "Email". At the bottom of the form, there are two buttons: "Save" on the left and "Close" on the right. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Code Listing:

```
Form1.vb*  Form1.vb [Design]*  StaffMenu.vb*  StaffMenu.vb [Design]*  AdminMenu.vb*  AdminMenu.vb [Design]*  Module1.vb*
Coursework System  butAddParentClose  Click
1 Imports System.IO
2 Public Class Form1
3     Private Sub butAddParentSave_Click(sender As Object, e As EventArgs) Handles butAddParentSave.Click
4
5         LoadParents()
6
7         'Declaring the Parent Data and it's variables
8         Dim ParentID As String
9         Dim FirstName As String
10        Dim Surname As String
11        Dim DoB As String
12        Dim Postcode As String
13        Dim Street As String
14        Dim Town As String
15        Dim City As String
16        Dim PhoneNum As String
17        Dim Email As String
18
19        'Inputting Data from the form'
20        ParentID = "PNT" & (NumParents + 1)
21        FirstName = txtAddParentFName.Text
22        Surname = txtAddParentSurname.Text
23        Try
24            DoB = txtAddParentDoB.Text
25        Catch
26            MsgBox("DoB must be **/**/*****")
27        End Try
28        Postcode = txtAddParentPostcode.Text
29        Street = txtAddParentStreet.Text
```

```

Town = txtAddParentTown.Text
City = txtAddParentCity.Text
PhoneNum = txtAddParentPhoneNum.Text
Email = txtAddParentEmail.Text

'Validation for first name'
If FirstName = "" Then
    MsgBox("You must enter a First Name, data not saved. Please try again")
    Exit Sub
End If

'Validation for surname'
If Surname = "" Then
    MsgBox("You must enter a Surname, data not saved. Please try again")
    Exit Sub
End If

'Validation for DoB'
Dim Correct As Boolean = DoB Like "**/**/**"
If Correct = False Or Len(DoB) <> 10 Then
    MsgBox("Date of birth must follow **/**/** format, data not saved. Please try again")
    Exit Sub
End If

'Validation for Postcode'
If Len(Postcode) > 7 Then
    MsgBox("Postcode must be 7 or less characters (do not leave spaces) data not saved. Please try again")
    Exit Sub
End If

'Validation for Street'
If Street = "" Then
    MsgBox("You must enter a Street, data not saved. Please try again")
    Exit Sub
End If

'Validation for Town'
If Town = "" Then
    MsgBox("You must enter a Town, data not saved. Please try again")
    Exit Sub
End If

'Validation for City/County'
If City = "" Then
    MsgBox("You must enter a City/County, data not saved. Please try again")
    Exit Sub
End If

'Validation for Phone Number'
If Len(PhoneNum) <> 11 Then
    MsgBox("Phone number must be 11 characters long, data not saved. Please try again")
    Exit Sub
End If

'Validation for Email'
Dim Correct2 As Boolean = Email Like "*@*.*"
If Correct2 = False Then
    MsgBox("Email must follow *@*.* format, data not saved. Please try again")
    Exit Sub
End If

```

```

'Clearing the textboxes'
txtAddParentFName.Text = ""
txtAddParentSurname.Text = ""
txtAddParentDoB.Text = ""
txtAddParentPostcode.Text = ""
txtAddParentStreet.Text = ""
txtAddParentTown.Text = ""
txtAddParentCity.Text = ""
txtAddParentPhoneNum.Text = ""
txtAddParentEmail.Text = ""

'Saving the data to the text file'
Dim NewParent As System.IO.StreamWriter
NewParent = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Parents.txt"), True)
NewParent.WriteLine(ParentID & "," & FirstName & "," & Surname & "," & DoB & "," & Postcode & "," & Street & "," & Town & "," & City & ",")
NewParent.Close()

LoadParents()
BubblesortParents()
SaveAllParents()

'End message to user, closing form
MsgBox("Your Parents data has been saved")
End Sub

```

References

```
Private Sub butAddParentClose_Click(sender As Object, e As EventArgs) Handles butAddParentClose.Click
```

```
'Operating close button
```

```
Me.Close()
```

Form Name: Edit/Delete Parent

The form is titled "Edit/Delete Parent" and is contained within a window titled "EditParents". It features a red background. The form contains the following fields and buttons:

- Parent ID
- First Name
- Surname
- Date of Birth
- Postcode
- Street
- Town
- City
- Phone Num
- Email
- Save
- Close
- Previous
- Next
- Delete Parent

Code Listing:

```
1 Public Class EditParents
2
3     Public Place As Integer = 0
4     Private Sub EditParents_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5
6         'Loading in all parents data (declared in module)
7         LoadParents()
8
9         'Linking the textboxes and data saved
10        txtEditParentID.Text = AllParents(Place).ParentID
11        txtEditParentFName.Text = AllParents(Place).FirstName
12        txtEditParentSurname.Text = AllParents(Place).Surname
13        txtEditParentDoB.Text = AllParents(Place).DoB
14        txtEditParentPostcode.Text = AllParents(Place).Postcode
15        txtEditParentStreet.Text = AllParents(Place).Street
16        txtEditParentTown.Text = AllParents(Place).Town
17        txtEditParentCity.Text = AllParents(Place).City
18        txtEditParentPhoneNum.Text = AllParents(Place).PhoneNum
19        txtEditParentEmail.Text = AllParents(Place).Email
20
21    End Sub
22
23    Private Sub butEditParentNext_Click(sender As Object, e As EventArgs) Handles butEditParentNext.Click
24
25        'Operating the next button to move to next record
26        Place = Place + 1
27        If Place = NumParents Then
28            Place = 0
29        End If
30    End Sub
31 End Class
```

```
End If
```

```
'Linking the textboxes and data saved
```

```
txtEditParentID.Text = AllParents(Place).ParentID  
txtEditParentFName.Text = AllParents(Place).FirstName  
txtEditParentSurname.Text = AllParents(Place).Surname  
txtEditParentDoB.Text = AllParents(Place).DoB  
txtEditParentPostcode.Text = AllParents(Place).Postcode  
txtEditParentStreet.Text = AllParents(Place).Street  
txtEditParentTown.Text = AllParents(Place).Town  
txtEditParentCity.Text = AllParents(Place).City  
txtEditParentPhoneNum.Text = AllParents(Place).PhoneNum  
txtEditParentEmail.Text = AllParents(Place).Email
```

```
End Sub
```

```
0 references
```

```
Private Sub butEditParentPrevious_Click(sender As Object, e As EventArgs) Handles butEditParentPrevious.Click
```

```
'Operating the previous button to move to previous record
```

```
Place = Place - 1  
If Place = -1 Then  
    Place = NumParents - 1  
End If
```

```
'Linking the textboxes and data saved
```

```
txtEditParentID.Text = AllParents(Place).ParentID  
txtEditParentFName.Text = AllParents(Place).FirstName  
txtEditParentSurname.Text = AllParents(Place).Surname  
txtEditParentDoB.Text = AllParents(Place).DoB  
txtEditParentPostcode.Text = AllParents(Place).Postcode  
txtEditParentStreet.Text = AllParents(Place).Street  
txtEditParentTown.Text = AllParents(Place).Town  
txtEditParentCity.Text = AllParents(Place).City  
txtEditParentPhoneNum.Text = AllParents(Place).PhoneNum  
txtEditParentEmail.Text = AllParents(Place).Email
```

```
End Sub
```

```
0 references
```

```
Private Sub butEditParentSave_Click(sender As Object, e As EventArgs) Handles butEditParentSave.Click
```

```
'Linking the textboxes and data saved
```

```
AllParents(Place).ParentID = txtEditParentID.Text  
AllParents(Place).FirstName = txtEditParentFName.Text  
AllParents(Place).Surname = txtEditParentSurname.Text  
AllParents(Place).DoB = txtEditParentDoB.Text  
AllParents(Place).Postcode = txtEditParentPostcode.Text  
AllParents(Place).Street = txtEditParentStreet.Text  
AllParents(Place).Town = txtEditParentTown.Text  
AllParents(Place).City = txtEditParentCity.Text  
AllParents(Place).PhoneNum = txtEditParentPhoneNum.Text  
AllParents(Place).Email = txtEditParentEmail.Text
```

```
'Saving changes made to parents (declared in module), end message to user and closing of form
```

```
SaveAllParents()  
MsgBox("Your changes have been saved")  
Close()
```

```
End Sub
```

References

```
Private Sub butEditParentClose_Click(sender As Object, e As EventArgs) Handles butEditParentClose.Click
```

```
    'Operating closing form button  
    Me.Close()
```

```
End Sub
```

References

```
Private Sub butEditParentDelete_Click(sender As Object, e As EventArgs) Handles butEditParentDelete.Click
```

```
    For x = Place To NumParents - 1  
        AllParents(x).ParentID = AllParents(x + 1).ParentID  
        AllParents(x).FirstName = AllParents(x + 1).FirstName  
        AllParents(x).Surname = AllParents(x + 1).Surname  
        AllParents(x).DoB = AllParents(x + 1).DoB  
        AllParents(x).Postcode = AllParents(x + 1).Postcode  
        AllParents(x).Street = AllParents(x + 1).Street  
        AllParents(x).Town = AllParents(x + 1).Town  
        AllParents(x).City = AllParents(x + 1).City  
        AllParents(x).PhoneNum = AllParents(x + 1).PhoneNum  
        AllParents(x).Email = AllParents(x + 1).Email
```

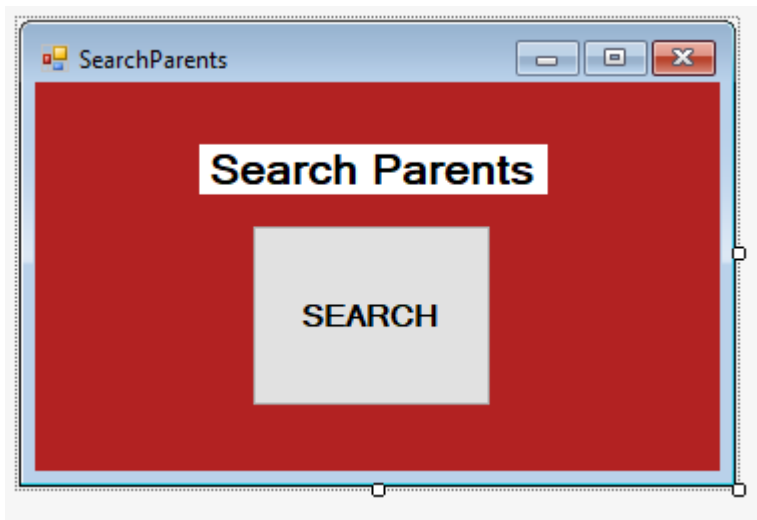
```
    Next
```

```
    NumParents = NumParents - 1
```

```
    SaveAllParents()  
    MsgBox("Parent deleted")
```

```
    Me.Close()
```

Form Name:



Code Listing:

```
Public Class SearchParents
    Private Sub butSearchParents_Click(sender As Object, e As EventArgs) Handles butSearchParents.Click

        Dim file As System.IO.StreamReader
        Dim parts(0 To 9) As String
        Dim line As String

        Dim SearchParentFName As String
        Dim SearchParentSurname As String

        SearchParentFName = InputBox("Enter Parent First Name:")
        SearchParentSurname = InputBox("Enter Parent Surname:")

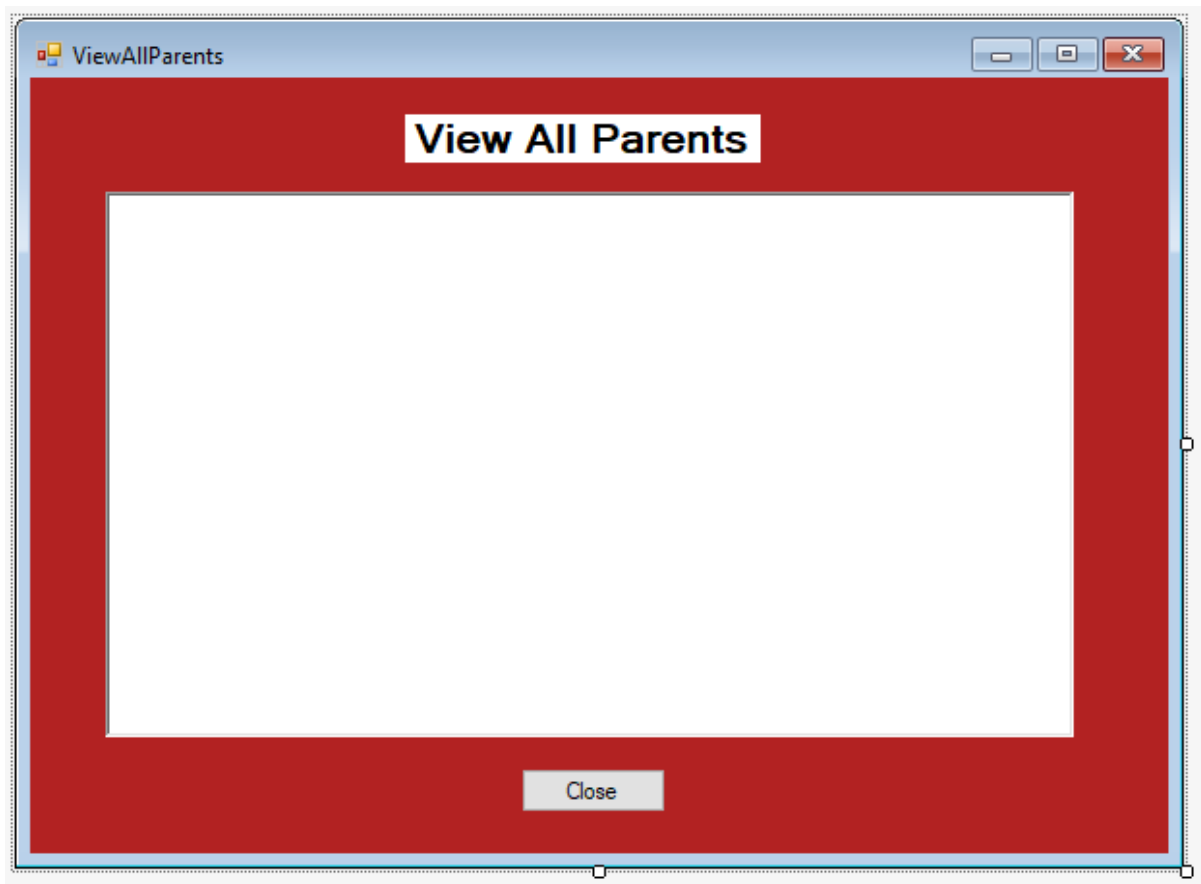
        file = My.Computer.FileSystem.OpenTextFileReader(Dir("All_Parents.txt"))

        Do
            line = file.ReadLine()
            parts = line.Split(",")

            If parts(1).ToUpper = SearchParentFName.ToUpper And parts(2).ToUpper = SearchParentSurname.ToUpper Then
                MsgBox("*****RECORD FOUND*****" + vbCrLf + vbCrLf + "Parent ID: " & parts(0) + vbCrLf + "First Name: " & parts(1) + vbCrLf)
                Exit Sub
            End If
        Loop Until (file.EndOfStream)

        MsgBox("No Record Found that matches")
    End Sub
End Class
```

Form Name: View All Parents



Code Listing:

```
2 references
:Public Class ViewAllParents
    0 references
3 | Private Sub butViewAllParentsClose_Click(sender As Object, e As EventArgs) Handles butViewAllParentsClose.Click
    |     Me.Close()
    | End Sub
-----
    0 references
3 | Private Sub ViewAllParents_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    |     rtbViewAllParents.LoadFile(Dir$("All_Parents.txt"), RichTextBoxStreamType.PlainText)
    | End Sub
End Class
```

Form Name: All Coach

The image shows a Windows form titled "AddCoach" with a red background. The form contains the following fields and controls:

- First Name: Text box
- Surname: Text box
- Date of Birth: Text box
- Postcode: Text box
- Street: Text box
- Town: Text box
- City: Text box
- Phone Num: Text box
- Email: Text box
- Age Group: Dropdown menu
- Save: Button
- Close: Button

Code Listing:

```
AddCoach.vb | ViewAllParents.vb* | ViewAllParents.vb [Design]* | SearchParents.vb | SearchParents.vb [Design] | EditParents.vb [Design]
Coursework System | (AddCoach Events) | Load
1 Imports System.IO
2 Public Class AddCoach
3 Private Sub butAddCoachSave_Click(sender As Object, e As EventArgs) Handles butAddCoachSave.Click
4
5     'Loading in all coaches (declared in module)'
6     LoadCoaches()
7
8     'Declaring Coach Data Variables'
9     Dim CoachID As String
10    Dim FirstName As String
11    Dim Surname As String
12    Dim DoB As String
13    Dim Postcode As String
14    Dim Street As String
15    Dim Town As String
16    Dim City As String
17    Dim PhoneNo As String
18    Dim Email As String
19    Dim AgeGroupCoached As String
20
21    'Inputting Data from the Form'
22    CoachID = "COA" & (NumCoaches + 1)
23    FirstName = txtAddCoachFName.Text
24    Surname = txtAddCoachSurname.Text
25    Try
26        DoB = txtAddCoachDoB.Text
27    Catch
28        MsgBox("DoB must be **/**/****")
29    End Try
```

```

AddCoach.vb  ViewAllParents.vb*  ViewAllParents.vb [Design]*  SearchParents.vb  SearchParents.vb [Design]  EditParents.vb [Design]
Coursework System  (AddCoach Events)  Load
29      End Try
30      Postcode = txtAddCoachPostcode.Text
31      Street = txtAddCoachStreet.Text
32      Town = txtAddCoachTown.Text
33      City = txtAddCoachCity.Text
34      PhoneNo = txtAddCoachPhoneNum.Text
35      Email = txtAddCoachEmail.Text
36      AgeGroupCoached = comAddCoachAgeGroupCoached.Text
37
38      'Validation for first name'
39      If FirstName = "" Then
40          MsgBox("You must enter a First Name, data not saved. Please try again")
41          Exit Sub
42      End If
43
44      'Validation for surname'
45      If Surname = "" Then
46          MsgBox("You must enter a Surname, data not saved. Please try again")
47          Exit Sub
48      End If
49
50      'Validation for DoB'
51      Dim Correct As Boolean = DoB Like "**/**/**"
52      If Correct = False Or Len(DoB) <> 10 Then
53          MsgBox("Date of birth must follow **/**/**** format, data not saved. Please try again")
54          Exit Sub
55      End If
56
57      'Validation for Postcode'
58      If Len(Postcode) > 7 Then
59          MsgBox("Postcode must be 7 or less characters (do not leave spaces) data not saved. Please try again")
60          Exit Sub
61      End If
62
63      'Validation for Street'
64      If Street = "" Then
65          MsgBox("You must enter a Street, data not saved. Please try again")
66          Exit Sub
67      End If
68
69      'Validation for Town'
70      If Town = "" Then
71          MsgBox("You must enter a Town, data not saved. Please try again")
72          Exit Sub
73      End If
74
75      'Validation for City/County'
76      If City = "" Then
77          MsgBox("You must enter a City/County, data not saved. Please try again")
78          Exit Sub
79      End If
80
81      'Validation for Phone Number'
82      If Len(PhoneNo) <> 11 Then
83          MsgBox("Phone number must be 11 characters long, data not saved. Please try again")
84          Exit Sub
85      End If
86
87      'Validation for Email'
88      Dim Correct2 As Boolean = Email Like "**@*.**"
89      If Correct2 = False Then

```

```

AddCoach.vb  ViewAllParents.vb*  ViewAllParents.vb [Design]*  SearchParents.vb  SearchParents.vb [Design]  EditParents.vb [Design]
Coursework System  (AddCoach Events)  Load
89  If Correct2 = False Then
90      MsgBox("Email must follow *@*. * format, data not saved. Please try again")
91      Exit Sub
92  End If
93
94  'Clearing Textboxes'
95  txtAddCoachFName.Text = ""
96  txtAddCoachSurname.Text = ""
97  txtAddCoachDoB.Text = ""
98  txtAddCoachPostcode.Text = ""
99  txtAddCoachStreet.Text = ""
100 txtAddCoachTown.Text = ""
101 txtAddCoachCity.Text = ""
102 txtAddCoachPhoneNum.Text = ""
103 txtAddCoachEmail.Text = ""
104 comAddCoachAgeGroupCoached.Text = ""
105
106 'Saving the data to the text file'
107 Dim NewCoach As System.IO.StreamWriter
108 NewCoach = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Coaches.txt"), True)
109 NewCoach.WriteLine(CoachID & ", " & FirstName & ", " & Surname & ", " & DoB & ", " & Postcode & ", " & Street & ", " & Town & ", " &
110 NewCoach.Close()
111
112 LoadCoaches() 'Load in the newly saved coach
113 BubblesortCoach() 'Sort coaches
114 SaveAllCoaches() 'Save sorted coaches to textfile
115
116 'End Message to user and closing form
117 MsgBox("Your Coaches data has been saved")
118
Private Sub AddCoach_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    comAddCoachAgeGroupCoached.Items.Add("U8")
    comAddCoachAgeGroupCoached.Items.Add("U9")
    comAddCoachAgeGroupCoached.Items.Add("U10")
    comAddCoachAgeGroupCoached.Items.Add("U11")
    comAddCoachAgeGroupCoached.Items.Add("U12")
    comAddCoachAgeGroupCoached.Items.Add("U13")
    comAddCoachAgeGroupCoached.Items.Add("U14")
    comAddCoachAgeGroupCoached.Items.Add("U15")
    comAddCoachAgeGroupCoached.Items.Add("U16")
    comAddCoachAgeGroupCoached.Items.Add("U17")
    comAddCoachAgeGroupCoached.Items.Add("U18")
    comAddCoachAgeGroupCoached.Items.Add("U19")
    comAddCoachAgeGroupCoached.Items.Add("U21")
    comAddCoachAgeGroupCoached.Items.Add("U23")

End Sub

0 references
Private Sub butAddCoachClose_Click(sender As Object, e As EventArgs) Handles butAddCoachClose.Click
    Me.Close()
End Sub
Class

```

Form Name: Edit/Delete Coach



The image shows a web form window titled "EditCoach" with a red background. The form is titled "Edit/Delete Coach" and contains the following fields and buttons:

- Coach ID
- First Name
- Surname
- Date of Birth
- Postcode
- Street
- Town
- City
- Phone Num
- Email
- Age Group (dropdown menu)
- Save
- Close
- Previous
- Next
- Delete Coach

Code Listing:

```

EditCoach.vb -> X EditCoach.vb [Design] AddCoach.vb ViewAllParents.vb* ViewAllParents.vb [Design]* SearchParents.vb
VB Coursework System (EditCoach Events) Load
1 Public Class EditCoach
2
3     Public Place As Integer
4     Private Sub EditCoach_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5
6         'Loading all coaches in (declared in module)'
7         LoadCoaches()
8
9         'Linking the textboxes to the data saved
10        txtEditCoachID.Text = AllCoaches(Place).CoachID
11        txtEditCoachFName.Text = AllCoaches(Place).FirstName
12        txtEditCoachSurname.Text = AllCoaches(Place).Surname
13        txtEditCoachDoB.Text = AllCoaches(Place).DoB
14        txtEditCoachPostcode.Text = AllCoaches(Place).Postcode
15        txtEditCoachStreet.Text = AllCoaches(Place).Street
16        txtEditCoachTown.Text = AllCoaches(Place).Town
17        txtEditCoachCity.Text = AllCoaches(Place).City
18        txtEditCoachPhoneNum.Text = AllCoaches(Place).PhoneNo
19        txtEditCoachEmail.Text = AllCoaches(Place).Email
20        comEditCoachAgeGroupCoached.Text = AllCoaches(Place).AgeGroupCoached
21
22    End Sub
23    Private Sub butEditCoachNext_Click(sender As Object, e As EventArgs) Handles butEditCoachNext.Click
24
25        'Operating the next button to move to the next record
26        Place = Place + 1
27        If Place = NumCoaches Then
28            Place = 0
29        End If
30
31        'Linking the text boxes to the data saved
32        txtEditCoachID.Text = AllCoaches(Place).CoachID
33        txtEditCoachFName.Text = AllCoaches(Place).FirstName
34        txtEditCoachSurname.Text = AllCoaches(Place).Surname
35        txtEditCoachDoB.Text = AllCoaches(Place).DoB
36        txtEditCoachPostcode.Text = AllCoaches(Place).Postcode
37        txtEditCoachStreet.Text = AllCoaches(Place).Street
38        txtEditCoachTown.Text = AllCoaches(Place).Town
39        txtEditCoachCity.Text = AllCoaches(Place).City
40        txtEditCoachPhoneNum.Text = AllCoaches(Place).PhoneNo
41        txtEditCoachEmail.Text = AllCoaches(Place).Email
42        comEditCoachAgeGroupCoached.Text = AllCoaches(Place).AgeGroupCoached
43
44    End Sub
45
46    Private Sub butEditCoachPrevious_Click(sender As Object, e As EventArgs) Handles butEditCoachPrevious.Click
47
48        'Operating the previous button to move to the previous record
49        Place = Place - 1
50        If Place = -1 Then
51            Place = NumCoaches - 1
52        End If
53
54        'Linking the text boxes to the data saved
55        txtEditCoachID.Text = AllCoaches(Place).CoachID
56        txtEditCoachFName.Text = AllCoaches(Place).FirstName
57        txtEditCoachSurname.Text = AllCoaches(Place).Surname
58        txtEditCoachDoB.Text = AllCoaches(Place).DoB
59        txtEditCoachPostcode.Text = AllCoaches(Place).Postcode

```

```

EditCoach.vb -p x EditCoach.vb [Design] AddCoach.vb ViewAllParents.vb* ViewAllParents.vb [Design]* SearchParents.vb
VB Coursework System (EditCoach Events) Load
59 txtEditCoachPostcode.Text = AllCoaches(Place).Postcode
60 txtEditCoachStreet.Text = AllCoaches(Place).Street
61 txtEditCoachTown.Text = AllCoaches(Place).Town
62 txtEditCoachCity.Text = AllCoaches(Place).City
63 txtEditCoachPhoneNum.Text = AllCoaches(Place).PhoneNo
64 txtEditCoachEmail.Text = AllCoaches(Place).Email
65 comEditCoachAgeGroupCoached.Text = AllCoaches(Place).AgeGroupCoached
66
67 End Sub
68
0 references
69 Private Sub butEditCoachSave_Click(sender As Object, e As EventArgs) Handles butEditCoachSave.Click
70
71
72 'Linking the data entered with the text boxes
73 AllCoaches(Place).CoachID = txtEditCoachID.Text
74 AllCoaches(Place).FirstName = txtEditCoachFName.Text
75 AllCoaches(Place).Surname = txtEditCoachSurname.Text
76 AllCoaches(Place).DoB = txtEditCoachDoB.Text
77 AllCoaches(Place).Postcode = txtEditCoachPostcode.Text
78 AllCoaches(Place).Street = txtEditCoachStreet.Text
79 AllCoaches(Place).Town = txtEditCoachTown.Text
80 AllCoaches(Place).City = txtEditCoachCity.Text
81 AllCoaches(Place).PhoneNo = txtEditCoachPhoneNum.Text
82 AllCoaches(Place).Email = txtEditCoachEmail.Text
83 AllCoaches(Place).AgeGroupCoached = comEditCoachAgeGroupCoached.Text
84
85 'Saving all the changes to coach data (declared in module, end message to user and closing form
86 SaveAllCoaches()
87 MsgBox("Your changes have been saved")
88 Close()

Private Sub butEditCoachClose_Click(sender As Object, e As EventArgs) Handles butEditCoachClose.Click
    Me.Close()
End Sub

0 references
Private Sub butEditCoachDelete_Click(sender As Object, e As EventArgs) Handles butEditCoachDelete.Click

    For x = Place To NumCoaches - 1
        AllCoaches(x).CoachID = AllCoaches(x + 1).CoachID
        AllCoaches(x).FirstName = AllCoaches(x + 1).FirstName
        AllCoaches(x).Surname = AllCoaches(x + 1).Surname
        AllCoaches(x).DoB = AllCoaches(x + 1).DoB
        AllCoaches(x).Postcode = AllCoaches(x + 1).Postcode
        AllCoaches(x).Street = AllCoaches(x + 1).Street
        AllCoaches(x).Town = AllCoaches(x + 1).Town
        AllCoaches(x).City = AllCoaches(x + 1).City
        AllCoaches(x).PhoneNo = AllCoaches(x + 1).PhoneNo
        AllCoaches(x).Email = AllCoaches(x + 1).Email
        AllCoaches(x).AgeGroupCoached = AllCoaches(x + 1).AgeGroupCoached
    Next

    NumCoaches = NumCoaches - 1

    SaveAllCoaches()
    MsgBox("Coach deleted")

    Me.Close()
End Sub
d Class

```

Form Name: Search Coach



Code Listing:

```
Public Class SearchCoaches
    References
    Private Sub butSearchCoaches_Click(sender As Object, e As EventArgs) Handles butSearchCoaches.Click
        Dim file As System.IO.StreamReader
        Dim parts(0 To 9) As String
        Dim line As String

        Dim SearchCoachFName As String
        Dim SearchCoachSurname As String
        SearchCoachFName = InputBox("Enter Coach First Name:")
        SearchCoachSurname = InputBox("Enter Coach Surname:")

        file = My.Computer.FileSystem.OpenTextFileReader(Dir("All_Coaches.txt"))

        Do
            line = file.ReadLine()
            parts = line.Split(",")

            If parts(1).ToUpper = SearchCoachFName.ToUpper And parts(2).ToUpper = SearchCoachSurname.ToUpper Then
                MsgBox("*****RECORD FOUND*****" + vbCrLf + vbCrLf + "Coach ID: " & parts(0) + vbCrLf + "First Name: " & parts(1) + vbC
                Exit Sub
            End If
        Loop Until (file.EndOfStream)

        MsgBox("No Record Found that matches")
    End Sub
End Class
```


Form Name: View All Coaches



Code Listing:

```
ViewAllCoaches.vb*  X ViewAllCoaches.vb [Design]* SearchCoaches.vb SearchCoaches.vb [Design] EditCoach.vb EditCoach.vb [Design]
Coursework System butViewAllCoachesClose Click
1 Public Class ViewAllCoaches
  0 references
2 Private Sub ViewAllCoaches_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3     rtbViewAllCoaches.LoadFile(Dir$("All_Coaches.txt"), RichTextBoxStreamType.PlainText)
4 End Sub
5
6 Private Sub butViewAllCoachesClose_Click(sender As Object, e As EventArgs) Handles butViewAllCoachesClose.Click
7     Me.Close()
8 End Sub
9
```

Form Name: Add Player



The screenshot shows a Windows form titled "AddPlayer" with a red background. The form contains the following elements:

- First Name: Text input field
- Surname: Text input field
- Date of Birth: Text input field
- Age Group: Dropdown menu
- Position: Dropdown menu
- Parent ID: Dropdown menu
- Save: Button
- Close: Button

Code Listing:

```
1 Imports System.IO
2 Public Class AddPlayer
3     Private Sub butAddPlayerSave_Click(sender As Object, e As EventArgs) Handles butAddPlayerSave.Click
4
5         'Loading in players (declared in module)
6
7         LoadPlayers()
8
9         'Declaring Player Data Variables'
10        Dim PlayerID As String
11        Dim FirstName As String
12        Dim Surname As String
13        Dim DoB As String
14        Dim AgeGroup As String
15        Dim Position As String
16        Dim ParentID As String
17
18        'Inputting Data from the form'
19        PlayerID = "PLY" & (NumPlayers + 1)
20        FirstName = txtAddPlayerFName.Text
21        Surname = txtAddPlayerSurname.Text
22        Try
23            DoB = txtAddPlayerDoB.Text
24        Catch
25            MsgBox("DoB must be **/**/*****")
26        End Try
27        AgeGroup = comAddPlayerAgeGroup.Text
28        Position = comAddPlayerPosition.Text
29        ParentID = lblParentID.Text
```

```

AddPlayer.vb [Design] | ViewAllCoaches.vb* | ViewAllCoaches.vb [Design]* | SearchCoaches.vb | SearchCoaches.vb [Design]
Coursework System | (AddPlayer Events) | Load
30
31 'Validation for first name'
32 If FirstName = "" Then
33     MsgBox("You must enter a First Name, data not saved. Please try again")
34     Exit Sub
35 End If
36
37 'Validation for surname'
38 If Surname = "" Then
39     MsgBox("You must enter a Surname, data not saved. Please try again")
40     Exit Sub
41 End If
42
43 'Validation for DoB'
44 Dim Correct As Boolean = DoB Like "**/**/"
45 If Correct = False Or Len(DoB) <> 10 Then
46     MsgBox("Date of birth must follow **/**/**** format, data not saved. Please try again")
47     Exit Sub
48 End If
49
50 'Validation for Age Group'
51 If AgeGroup = "" Then
52     MsgBox("You must select an age group from the drop down list, data not saved. Please try again")
53     Exit Sub
54 End If
55
56 'Validation for Position'
57 If Position = "" Then
58     MsgBox("You must select a position from the drop down list, data not saved. Please try again")
59     Exit Sub
60 End If
61
62 'Validation for ParentID'
63 If ParentID = "" Then
64     MsgBox("You must select a ParentID from the drop down list, data not saved. Please try again")
65     Exit Sub
66 End If
67
68 'Clearing the textboxes'
69 txtAddPlayerFName.Text = ""
70 txtAddPlayerSurname.Text = ""
71 txtAddPlayerDoB.Text = ""
72 comAddPlayerAgeGroup.Text = ""
73 comAddPlayerPosition.Text = ""
74 comAddPlayerParentID.Text = ""
75 lblParentID.Text = ""
76
77 'Saving the data to the text file'
78 Dim NewPlayer As System.IO.StreamWriter
79 NewPlayer = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Players.txt"), True)
80 NewPlayer.WriteLine(PlayerID & "," & FirstName & "," & Surname & "," & DoB & "," & AgeGroup & "," & Position & "," & ParentID)
81 NewPlayer.Close()
82
83 LoadPlayers()
84 BubblesortPlayers()
85 SaveAllPlayers()
86
87 'Adding player to SubPaid file
88 Dim NewPlayerSubs As System.IO.StreamWriter
89 NewPlayerSubs = My.Computer.FileSystem.OpenTextFileWriter(Dir$("SubsRecords.txt"), True)
90 NewPlayerSubs.WriteLine(PlayerID & "," & 0)
91 NewPlayerSubs.Close()
92

```

```

AddPlayer.vb [Design] | ViewAllCoaches.vb* | ViewAllCoaches.vb [Design]* | SearchCoaches.vb | SearchCoaches.vb [Design]
Coursework System | (AddPlayer Events) | Load
92
93 'End message to user and closing form
94 MsgBox("Your Players data has been saved")
95 End Sub
96
97 O references
98 Private Sub AddPlayer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
99 'Adding items to my Position, Age Group and ParentID combo boxes'
100
101 LoadParents()
102 comAddPlayerPosition.Items.Add("Goalkeeper")
103 comAddPlayerPosition.Items.Add("Defender")
104 comAddPlayerPosition.Items.Add("Midfielder")
105 comAddPlayerPosition.Items.Add("Forward")
106
107 comAddPlayerAgeGroup.Items.Add("U8")
108 comAddPlayerAgeGroup.Items.Add("U9")
109 comAddPlayerAgeGroup.Items.Add("U10")
110 comAddPlayerAgeGroup.Items.Add("U11")
111 comAddPlayerAgeGroup.Items.Add("U12")
112 comAddPlayerAgeGroup.Items.Add("U13")
113 comAddPlayerAgeGroup.Items.Add("U14")
114 comAddPlayerAgeGroup.Items.Add("U15")
115 comAddPlayerAgeGroup.Items.Add("U16")
116 comAddPlayerAgeGroup.Items.Add("U17")
117 comAddPlayerAgeGroup.Items.Add("U18")
118 comAddPlayerAgeGroup.Items.Add("U19")
119 comAddPlayerAgeGroup.Items.Add("U21")
120 comAddPlayerAgeGroup.Items.Add("U23")
121
122 For x = 0 To NumParents - 1
123     comAddPlayerParentID.Items.Add(AllParents(x).FirstName & " " & AllParents(x).Surname)
124 Next
125 End Sub
126
127 O references
128 Private Sub comAddPlayerParentID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comAddPlayerParentID.SelectedIndex
129
130 'Linking the ParentID combobox name to the ParentID label
131 lblParentID.Text = AllParents(comAddPlayerParentID.SelectedIndex).ParentID
132
133 End Sub
134
135 O references
136 Private Sub butAddPlayerClose_Click(sender As Object, e As EventArgs) Handles butAddPlayerClose.Click
137
138 'Closing the form when close button is clicked
139 Close()
140
141 End Sub
142 End Class

```

Form Name: Edit/Delete Player

The form is titled "Edit Player" and contains the following fields and controls:

- Player ID: Text box
- First Name: Text box
- Surname: Text box
- Date of Birth: Text box
- Age Group: Dropdown menu
- Position: Dropdown menu
- Parent ID: Dropdown menu
- Buttons: Save, Close, Previous, Next, Delete Player

Code Listing:

```
1 Public Class EditPlayer
2
3     Public Place As Integer = 0
4     Private Sub EditPlayer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5
6         'Loading in both players and parents data (declared in array)
7         LoadPlayers()
8         LoadParents()
9
10        'Linking the textboxes/label to the data saved
11        txtEditPlayerID.Text = AllPlayers(Place).PlayerID
12        txtEditPlayerFName.Text = AllPlayers(Place).FirstName
13        txtEditPlayerSurname.Text = AllPlayers(Place).Surname
14        txtEditPlayerDoB.Text = AllPlayers(Place).DoB
15        comEditPlayerPosition.Text = AllPlayers(Place).Position
16        comEditPlayerAgeGroup.Text = AllPlayers(Place).AgeGroup
17        lblEditParentID.Text = AllPlayers(Place).ParentID
18        For x = 0 To NumParents - 1
19            If AllPlayers(Place).ParentID = AllParents(x).ParentID Then
20                comEditPlayerParentID.Text = AllParents(x).FirstName & " " & AllParents(x).Surname
21            End If
22        Next
23
24        'Linking the items to the comboboxes for Position, AgeGroup and ParentID
25        comEditPlayerPosition.Items.Add("Goalkeeper")
26        comEditPlayerPosition.Items.Add("Defender")
27        comEditPlayerPosition.Items.Add("Midfielder")
28        comEditPlayerPosition.Items.Add("Forward")
29
30        comEditPlayerAgeGroup.Items.Add("U8")
```

```

EditPlayer.vb [Design] | AddPlayer.vb | AddPlayer.vb [Design] | ViewAllCoaches.vb* | ViewAllCoaches.vb [Design]*
VB Coursework System (EditPlayer Events) Load
30 comEditPlayerAgeGroup.Items.Add("U8")
31 comEditPlayerAgeGroup.Items.Add("U9")
32 comEditPlayerAgeGroup.Items.Add("U10")
33 comEditPlayerAgeGroup.Items.Add("U11")
34 comEditPlayerAgeGroup.Items.Add("U12")
35 comEditPlayerAgeGroup.Items.Add("U13")
36 comEditPlayerAgeGroup.Items.Add("U14")
37 comEditPlayerAgeGroup.Items.Add("U15")
38 comEditPlayerAgeGroup.Items.Add("U16")
39 comEditPlayerAgeGroup.Items.Add("U17")
40 comEditPlayerAgeGroup.Items.Add("U18")
41 comEditPlayerAgeGroup.Items.Add("U19")
42 comEditPlayerAgeGroup.Items.Add("U21")
43 comEditPlayerAgeGroup.Items.Add("U23")
44
45 For x = 0 To NumParents - 1
46     comEditPlayerParentID.Items.Add(AllParents(x).FirstName & " " & AllParents(x).Surname)
47 Next
48
49
50 End Sub
51
52 0 references
53 Private Sub butEditPlayerNext_Click(sender As Object, e As EventArgs) Handles butEditPlayerNext.Click
54     'Operating the next button to move to next record
55     Place = Place + 1
56     If Place = NumPlayers Then
57         Place = 0
58     End If
59
60 'Linking textboxes to data saved
61 txtEditPlayerID.Text = AllPlayers(Place).PlayerID
62 txtEditPlayerFName.Text = AllPlayers(Place).FirstName
63 txtEditPlayerSurname.Text = AllPlayers(Place).Surname
64 txtEditPlayerDoB.Text = AllPlayers(Place).DoB
65 comEditPlayerPosition.Text = AllPlayers(Place).Position
66 comEditPlayerAgeGroup.Text = AllPlayers(Place).AgeGroup
67 lblEditParentID.Text = AllPlayers(Place).ParentID
68 For x = 0 To NumParents - 1
69     If AllPlayers(Place).ParentID = AllParents(x).ParentID Then
70         comEditPlayerParentID.Text = AllParents(x).FirstName & " " & AllParents(x).Surname
71     End If
72 Next
73
74 End Sub
75
76 0 references
77 Private Sub butEditPlayerPrevious_Click(sender As Object, e As EventArgs) Handles butEditPlayerPrevious.Click
78     'Operating the previous button to move to the previous record
79     Place = Place - 1
80     If Place = -1 Then
81         Place = NumPlayers - 1
82     End If
83
84 'Linking textboxes to data saved
85 txtEditPlayerID.Text = AllPlayers(Place).PlayerID
86 txtEditPlayerFName.Text = AllPlayers(Place).FirstName
87 txtEditPlayerSurname.Text = AllPlayers(Place).Surname
88 txtEditPlayerDoB.Text = AllPlayers(Place).DoB

```

```

EditPlayer.vb [Design] | AddPlayer.vb | AddPlayer.vb [Design] | ViewAllCoaches.vb* | ViewAllCoaches.vb [Design]*
Coursework System | (EditPlayer Events) | Load
88      txtEditPlayerDoB.Text = AllPlayers(Place).DoB
89      comEditPlayerPosition.Text = AllPlayers(Place).Position
90      comEditPlayerAgeGroup.Text = AllPlayers(Place).AgeGroup
91      lblEditParentID.Text = AllPlayers(Place).ParentID
92      For x = 0 To NumParents - 1
93      If AllPlayers(Place).ParentID = AllParents(x).ParentID Then
94          comEditPlayerParentID.Text = AllParents(x).FirstName & " " & AllParents(x).Surname
95      End If
96      Next
97      End Sub
98
0 references
99      Private Sub btnEditPlayerSave_Click(sender As Object, e As EventArgs) Handles btnEditPlayerSave.Click
100
101          'Linking data saved to the textboxes
102          AllPlayers(Place).PlayerID = txtEditPlayerID.Text
103          AllPlayers(Place).FirstName = txtEditPlayerFName.Text
104          AllPlayers(Place).Surname = txtEditPlayerSurname.Text
105          AllPlayers(Place).DoB = txtEditPlayerDoB.Text
106          AllPlayers(Place).Position = comEditPlayerPosition.Text
107          AllPlayers(Place).AgeGroup = comEditPlayerAgeGroup.Text
108          AllPlayers(Place).ParentID = lblEditParentID.Text
109
110          'Saving all changes to players (declared in module), end message to users and closing the form
111          SaveAllPlayers()
112          MsgBox("Your changes have been saved")
113          Close()
114
115      End Sub
116
117
EditPlayer.vb [Design] | AddPlayer.vb | AddPlayer.vb [Design] | ViewAllCoaches.vb* | ViewAllCoaches.vb [Design]*
Coursework System | (EditPlayer Events) | Load
118      Private Sub comEditPlayerParentID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comEditPlayerParentID.SelectedIn
119
120          For x = 0 To NumParents - 1
121          If comEditPlayerParentID.Text = AllParents(x).FirstName & " " & AllParents(x).Surname Then
122              lblEditParentID.Text = AllParents(x).ParentID
123          End If
124          Next
125
126      End Sub
127
0 references
128      Private Sub btnEditPlayerClose_Click(sender As Object, e As EventArgs) Handles btnEditPlayerClose.Click
129          Me.Close()
130      End Sub
131
0 references
132      Private Sub btnEditPlayerDelete_Click(sender As Object, e As EventArgs) Handles btnEditPlayerDelete.Click
133
134          For x = Place To NumPlayers - 1
135          AllPlayers(x).PlayerID = AllPlayers(x + 1).PlayerID
136          AllPlayers(x).FirstName = AllPlayers(x + 1).FirstName
137          AllPlayers(x).Surname = AllPlayers(x + 1).Surname
138          AllPlayers(x).DoB = AllPlayers(x + 1).DoB
139          AllPlayers(x).AgeGroup = AllPlayers(x + 1).AgeGroup
140          AllPlayers(x).Position = AllPlayers(x + 1).Position
141          AllPlayers(x).ParentID = AllPlayers(x + 1).ParentID
142          Next
143
144          NumPlayers = NumPlayers - 1
145
146          SaveAllPlayers()
147          MsgBox("Player deleted")
148
149          Me.Close()
150
151      End Sub
152      End Class

```

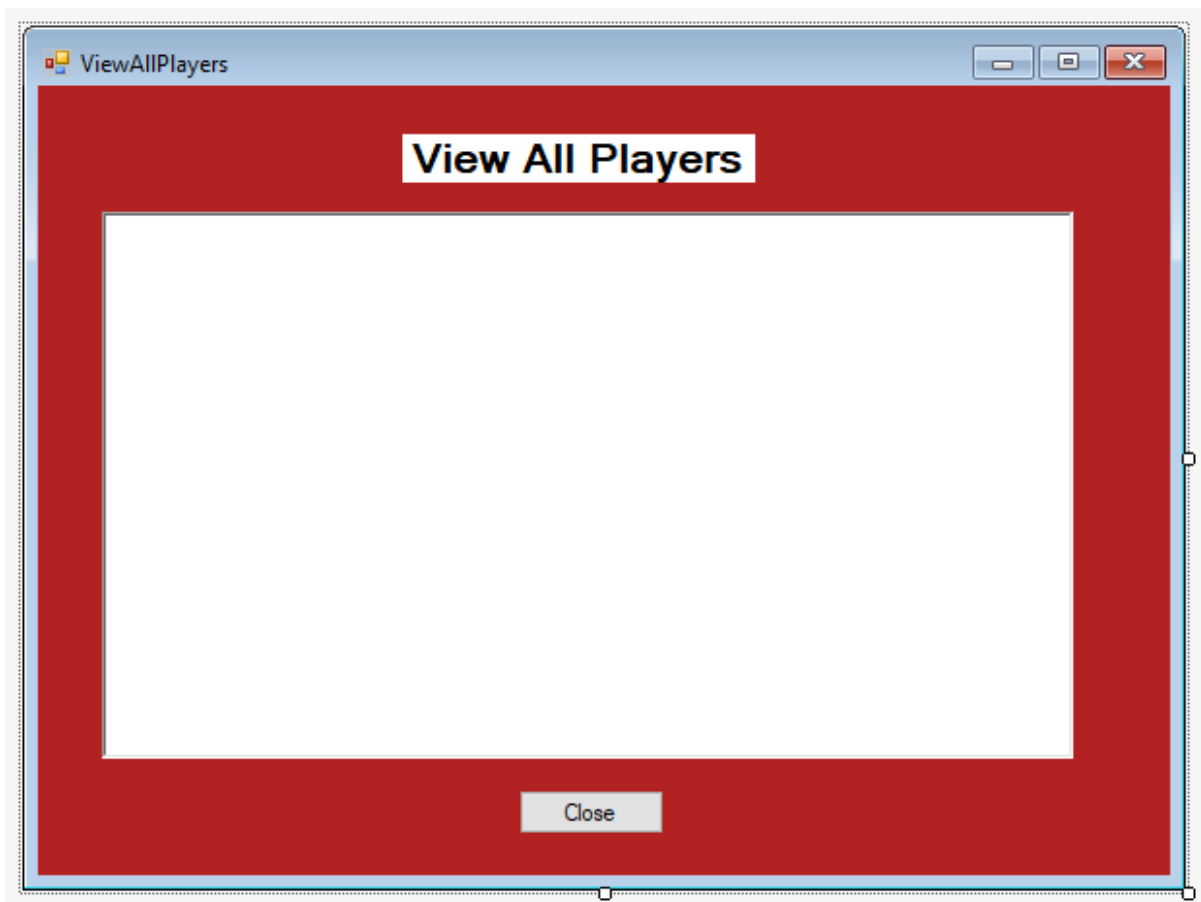
Form Name: Search Player



Code Listing:

```
SearchPlayers.vb x SearchPlayers.vb [Design] EditPlayer.vb EditPlayer.vb [Design] AddPlayer.vb AddPlayer.vb [Design]
Coursework System (SearchPlayers Events) Load
3 references
1 Public Class SearchPlayers
0 references
2 Private Sub butSearchPlayers_Click(sender As Object, e As EventArgs) Handles butSearchPlayers.Click
3
4     Dim file As System.IO.StreamReader
5     Dim parts(0 To 6) As String
6     Dim line As String
7
8     Dim SearchPlayerFName As String
9     Dim SearchPlayerSurname As String
10    SearchPlayerFName = InputBox("Enter Player's First Name:")
11    SearchPlayerSurname = InputBox("Enter Player's Surname:")
12
13    'Loading the players textfile
14    file = My.Computer.FileSystem.OpenTextFileReader(Dir("All_Players.txt"))
15
16
17    'Reads each line and splits the data at the ","
18    Do
19        line = file.ReadLine()
20        parts = line.Split(",")
21
22        If parts(1).ToUpper = SearchPlayerFName.ToUpper And parts(2).ToUpper = SearchPlayerSurname.ToUpper Then
23            MsgBox("*****RECORD FOUND*****" + vbCrLf + vbCrLf + "Player ID: " & parts(0) + vbCrLf + "First Name: " & parts(1) +
24                vbCrLf + "Surname: " & parts(2))
25            Exit Sub
26        End If
27    Loop Until (file.EndOfStream)
28    MsgBox("No Record Found that matches")
29 End Sub
```

Form Name: View All Players



Code Listing:

```
ViewAllPlayers.vb*  X ViewAllPlayers.vb [Design]* SearchPlayers.vb SearchPlayers.vb [Design] EditPlayer.vb EditPlayer.vb [Design]
Coursework System  butViewAllPlayersClose Click
1 Public Class ViewAllPlayers
2     Private Sub ViewAllPlayers_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3         rtbViewAllPlayers.LoadFile(Dir$("All_players.txt"), RichTextBoxStreamType.PlainText)
4     End Sub
5
6     Private Sub butViewAllPlayersClose_Click(sender As Object, e As EventArgs) Handles butViewAllPlayersClose.Click
7         Me.Close()
8     End Sub
9 End Class
```

Form Name: Add Stock



Code Listing:

```
AddStock.vb  AddStock.vb [Design]  ViewAllPlayers.vb*  ViewAllPlayers.vb [Design]*  SearchPlayers.vb  SearchPlayers.vb [Design]
Coursework System  (AddStock Events)  Load
1 Imports System.IO
2 Public Class AddStock
3     Private Sub butAddStockSave_Click(sender As Object, e As EventArgs) Handles butAddStockSave.Click
4
5
6         'Loading in all Stock (declared in module)
7         LoadStock()
8
9         'Declaring Stock Data Variables
10        Dim StockID As String
11        Dim Description As String
12        Dim Quantity As String
13        Dim ExtraNotes As String
14
15        'Inputting Data from the form
16        StockID = "STK" & (NumStock + 1)
17        Description = txtAddStockDescription.Text
18        Quantity = txtAddStockQuantity.Text
19        ExtraNotes = txtAddStockExtraInfo.Text
20
21        'Validation for Description
22        If Description = "" Then
23            MsgBox("You must enter a description, data not saved. Please try again")
24            Exit Sub
25        End If
26
27        'Validation for Quantity
28        If Quantity = "" Then
29            MsgBox("You must enter a quantity, data not saved. Please try again")
```

```

AddStock.vb*  AddStock.vb [Design]*  ViewAllPlayers.vb*  ViewAllPlayers.vb [Design]*  SearchPlayers.vb  SearchPlayers.vb [Design]
Coursework System  butAddStockSave  Click
28  If Quantity = "" Then
29      MsgBox("You must enter a quantity, data not saved. Please try again")
30      Exit Sub
31  End If
32
33  'Validation for Extra Notes
34  If ExtraNotes = "" Then
35      MsgBox("If this item does not have any extra notes, enter 'N/A', data not saved. Please try again")
36      Exit Sub
37  End If
38
39  'Clearing Textboxes'
40  txtAddStockDescription.Text = ""
41  txtAddStockQuantity.Text = ""
42  txtAddStockExtraInfo.Text = ""
43
44
45  'Saving the data to the text file'
46  Dim NewStock As System.IO.StreamWriter
47  NewStock = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Stock.txt"), True)
48  NewStock.WriteLine(StockID & "," & Description & "," & Quantity & "," & ExtraNotes)
49  NewStock.Close()
50
51  LoadStock()
52  BubblesortStock()
53  SaveAllStock()
54
55  End Sub
56

```

References

Private Sub butAddStockClose_Click(sender As Object, e As EventArgs) Handles butAddStockClose.Click

```

'Closing the form
Me.Close()

```

Form Name: Edit/Delete Stock

The screenshot shows a Windows form titled "EditStock" with a red background. At the top center, there is a white box with the text "Edit Stock". Below this, there are four text boxes labeled "Stock ID", "Description", "Quantity", and "Extra Info". At the bottom of the form, there are five buttons: "Save", "Close", "Previous", "Next", and "Delete Item".

Code Listing:

```
1 Public Class EditStock
2
3     Public Place As Integer = 0
4     Private Sub EditStock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5         'Loading in all stock data
6         LoadStock()
7
8         'Linking textboxes to the data saved
9         txtEditStockID.Text = AllStock(Place).StockID
10        txtEditStockDescription.Text = AllStock(Place).Description
11        txtEditStockQuantity.Text = AllStock(Place).Quantity
12        txtEditStockExtraInfo.Text = AllStock(Place).ExtraNotes
13    End Sub
14
15
16
17    Private Sub butEditStockNext_Click(sender As Object, e As EventArgs) Handles butEditStockNext.Click
18
19        'Operating next button to move to next record
20        Place = Place + 1
21        If Place = NumStock Then
22            Place = 0
23        End If
24
25        'Linking textboxes to the data saved
26        txtEditStockID.Text = AllStock(Place).StockID
27        txtEditStockDescription.Text = AllStock(Place).Description
28        txtEditStockQuantity.Text = AllStock(Place).Quantity
29        txtEditStockExtraInfo.Text = AllStock(Place).ExtraNotes
```

```

EditStock.vb [Design] | AddStock.vb* | AddStock.vb [Design]* | ViewAllPlayers.vb* | ViewAllPlayers.vb [Design]*
VB Coursework System | (EditStock Events) | Load

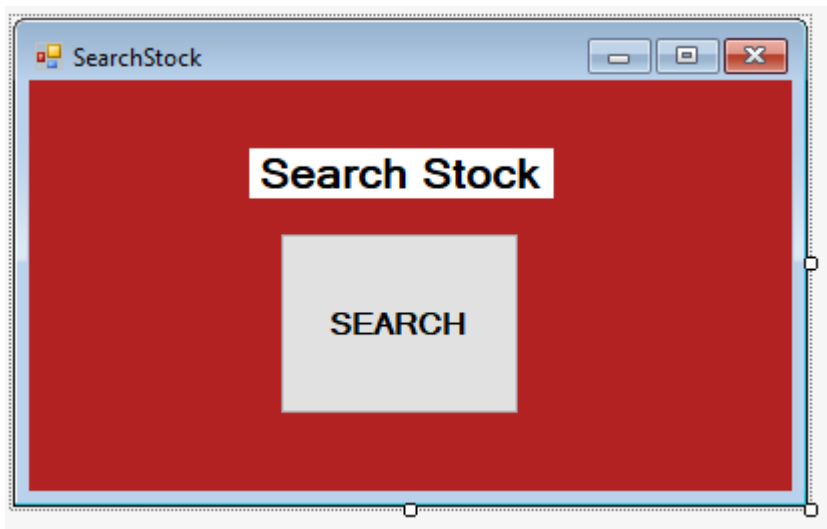
30 | End Sub
31 |
32 |
33 | Private Sub butEditStockPrevious_Click(sender As Object, e As EventArgs) Handles butEditStockPrevious.Click
34 |
35 |     'Operating previous button to move to previous record
36 |     Place = Place - 1
37 |     If Place = -1 Then
38 |         Place = NumStock - 1
39 |     End If
40 |
41 |     'Linking textboxes to the data saved
42 |     txtEditStockID.Text = AllStock(Place).StockID
43 |     txtEditStockDescription.Text = AllStock(Place).Description
44 |     txtEditStockQuantity.Text = AllStock(Place).Quantity
45 |     txtEditStockExtraInfo.Text = AllStock(Place).ExtraNotes
46 |
47 | End Sub
48 |
49 | Private Sub butEditStockSave_Click(sender As Object, e As EventArgs) Handles butEditStockSave.Click
50 |
51 |     'Linking data saved to the textboxes
52 |     AllStock(Place).StockID = txtEditStockID.Text
53 |     AllStock(Place).Description = txtEditStockDescription.Text
54 |     AllStock(Place).Quantity = txtEditStockQuantity.Text
55 |     AllStock(Place).ExtraNotes = txtEditStockExtraInfo.Text
56 |
57 |     'Saving all changes to data, end message displayed to user and close form
58 |     SaveAllStock()

EditStock.vb [Design] | AddStock.vb* | AddStock.vb [Design]* | ViewAllPlayers.vb* | ViewAllPlayers.vb [Design]*
VB Coursework System | (EditStock Events) | Load

56 |     'Saving all changes to data, end message displayed to user and close form
57 |     SaveAllStock()
58 |     MsgBox("Your changes have been saved")
59 |     Close()
60 |
61 | End Sub
62 |
63 |
64 | Private Sub butEditStockClose_Click(sender As Object, e As EventArgs) Handles butEditStockClose.Click
65 |     Me.Close()
66 | End Sub
67 |
68 | Private Sub butEditStockDelete_Click(sender As Object, e As EventArgs) Handles butEditStockDelete.Click
69 |
70 |     For x = Place To NumStock - 1
71 |         AllStock(x).Description = AllStock(x + 1).Description
72 |         AllStock(x).Quantity = AllStock(x + 1).Quantity
73 |         AllStock(x).ExtraNotes = AllStock(x + 1).ExtraNotes
74 |     Next
75 |
76 |     NumStock = NumStock - 1
77 |
78 |     SaveAllStock()
79 |     MsgBox("Item deleted")
80 |
81 |     Me.Close()
82 |
83 | End Sub
84 | End Class

```

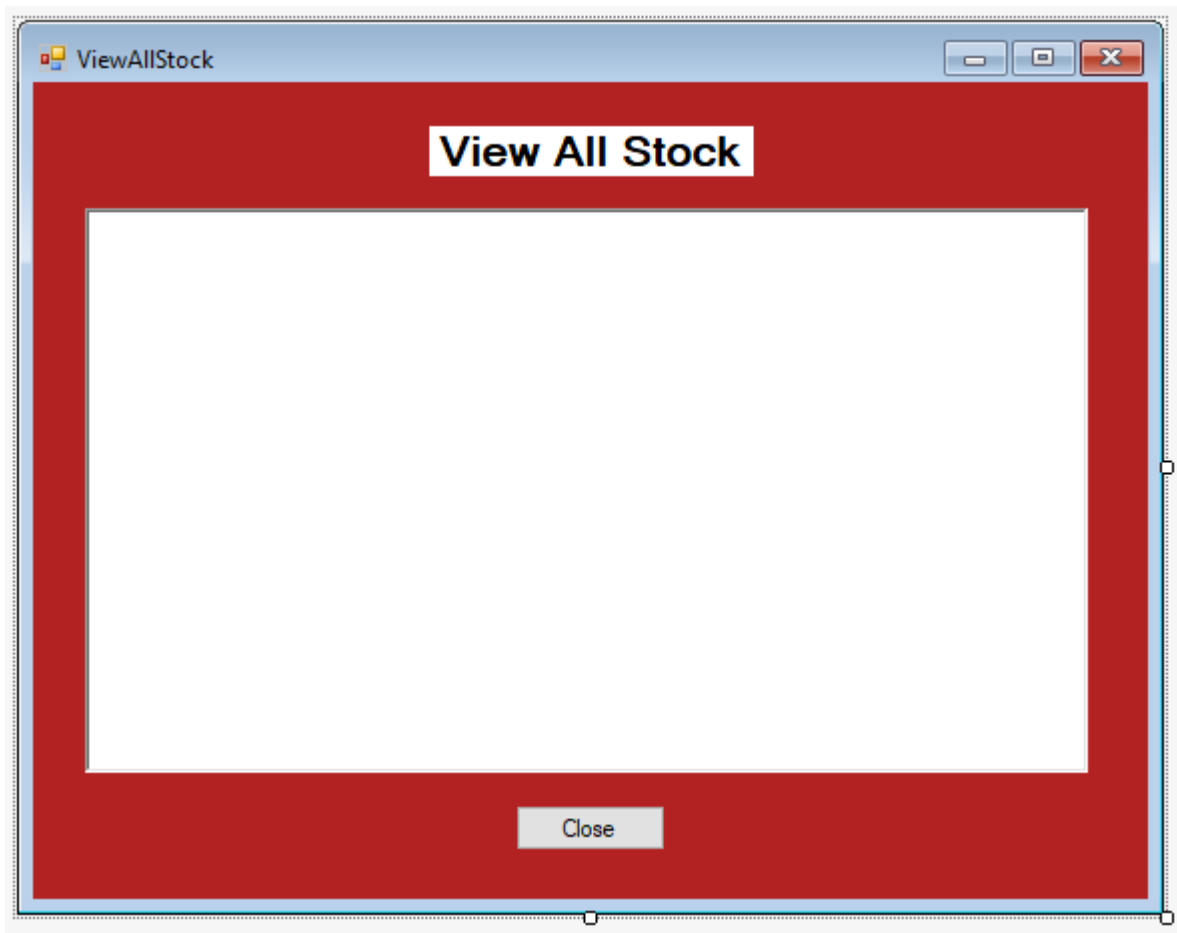
Form Name: Search Stock



Code Listing:

```
SearchStock.vb* x SearchStock.vb [Design]* EditStock.vb EditStock.vb [Design] AddStock.vb* AddStock.vb [Design]*
VB Coursework System butSearchStock Click
3 references
1 Public Class SearchStock
0 references
2 Private Sub butSearchStock_Click(sender As Object, e As EventArgs) Handles butSearchStock.Click
3
4 Dim file As System.IO.StreamReader
5 Dim parts(0 To 3) As String
6 Dim line As String
7
8 Dim SearchStockName As String
9 SearchStockName = InputBox("Enter Stock Item:")
10
11 file = My.Computer.FileSystem.OpenTextFileReader(Dir("All_Stock.txt"))
12
13 Do
14 line = file.ReadLine()
15 parts = line.Split(",")
16
17 If parts(1).ToUpper = SearchStockName.ToUpper Then
18 MsgBox("*****RECORD FOUND*****" + vbCrLf + vbCrLf + "Stock ID: " & parts(0) + vbCrLf + "Description: " & parts(1) +
19 Exit Sub
20 End If
21 Loop Until (file.EndOfStream)
22
23 MsgBox("No Record Found that matches")
24
25 End Sub
26 End Class
```

Form Name: View All Stock



Code Listing:

```
ViewAllStock.vb*  ViewAllStock.vb [Design]*  SearchStock.vb*  SearchStock.vb [Design]*  EditStock.vb  EditStock.vb [Design]
VB Coursework System  ViewAllStock  InitializeComponent
1  Public Class ViewAllStock
2      Private Sub ViewAllStock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3          rtbViewAllStock.LoadFile(Dir$("All_Stock.txt"), RichTextBoxStreamType.PlainText)
4      End Sub
5
6      Private Sub butViewAllStockClose_Click(sender As Object, e As EventArgs) Handles butViewAllStockClose.Click
7          Me.Close()
8      End Sub
9  End Class
```

Form Name: Create New User

The image shows a Windows-style window titled "CreateNewUser". The window has a red background and a white title bar with standard minimize, maximize, and close buttons. The main content area is titled "Create New User" in a large, bold, black font. Below the title, there are several input fields: a dropdown menu for "Access Level", and text boxes for "First Name", "Surname", "Date of Birth", "Password", "Re-Enter Password", "Security Question", and "Security Answer". At the bottom of the form, there are two buttons: "Save" and "Close".

Code Listing:

```
Coursework System -> CreateNewUser.vb [Design] | ViewAllStock.vb* | ViewAllStock.vb [Design]* | SearchStock.vb* | SearchStock.vb [Design]*
- Load
(CreateNewUser Events)

1 Imports System.IO
2 Public Class CreateNewUser
3     Public Structure User
4         Public AccessLevel As String
5         Public Username As String
6         Public FirstName As String
7         Public Surname As String
8         Public DoB As String
9         Public Password As String
10        Public RePassword As String
11        Public SecurityQuestion As String
12        Public SecurityAnswer As String
13    End Structure
14
15    Public AddUser As User
16    Private Sub butCreateUserSave_Click(sender As Object, e As EventArgs) Handles butCreateUserSave.Click
17
18        AddUser.AccessLevel = comCreateUserAccessLevel1.Text
19        Try
20            AddUser.Username = txtCreateUserSurname.Text & txtCreateUserFName.Text(0) & txtCreateUserDoB.Text(0) & txtCreateUserDoB.
21        Catch
22            MsgBox("You must enter a First Name and Surname. Please try again")
23            Exit Sub
24        End Try
25        AddUser.FirstName = txtCreateUserFName.Text
26        AddUser.Surname = txtCreateUserSurname.Text
27        Try
28            AddUser.DoB = txtCreateUserDoB.Text
29        Catch
30            Exit Sub
31        End Try
32    End Sub
33 End Class
```

```

CreateNewUser.vb -> X CreateNewUser.vb [Design] ViewAllStock.vb* ViewAllStock.vb [Design]* SearchStock.vb* SearchStock.vb [Design]*
VB Coursework System (CreateNewUser Events) Load
27 Try
28     AddUser.DoB = txtCreateUserDoB.Text
29 Catch
30     MsgBox("DoB must be **/**/****")
31 End Try
32 AddUser.Password = txtCreateUserPassword.Text
33 AddUser.RePassword = txtCreateUserRePassword.Text
34 AddUser.SecurityQuestion = txtCreateUserSecurityQ.Text
35 AddUser.SecurityAnswer = txtCreateUserSecurityAns.Text
36
37 'Validation for Access Level'
38 If AddUser.AccessLevel = "" Then
39     MsgBox("You must select an access level, data not saved. Please try again")
40     Exit Sub
41 End If
42
43 'Validation for Surname'
44 If AddUser.Surname = "" Then
45     MsgBox("You must enter a surname, data not saved. Please try again")
46     Exit Sub
47 End If
48
49 'Validation for DoB'
50 Dim Correct As Boolean = AddUser.DoB Like "**/**/****"
51 If Correct = False Or Len(AddUser.DoB) <> 10 Then
52     MsgBox("Date of birth must follow **/**/**** format, data not saved. Please try again")
53     Exit Sub
54 End If
55
56 'Validation for Security Question'
57 If AddUser.SecurityQuestion = "" Then
58     MsgBox("You must enter a security question, data not saved. Please try again")
59     Exit Sub
60 End If
61
62 'Validation for Security Answer'
63 If AddUser.SecurityAnswer = "" Then
64     MsgBox("You must enter an answer to your security question, data not saved. Please try again")
65     Exit Sub
66 End If
67
68 'Validation for Password'
69 If Len(AddUser.Password) < 8 Then
70     MsgBox("Your password must be a minimum of 8 characters, please try again")
71     Exit Sub
72 End If
73
74 'Double entry check on password & conf password'
75 If AddUser.Password <> AddUser.RePassword Then
76     MsgBox("Your Passwords entered do not match, please try again")
77     Exit Sub
78 End If
79
80
81
82 Dim AddUser2 As System.IO.StreamWriter
83 AddUser2 = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Users.txt"), True)
84 AddUser2.WriteLine(AddUser.Username & ", " & AddUser.AccessLevel & ", " & AddUser.FirstName & ", " & AddUser.Surname & ", " & Add
85 AddUser2.Close()
86
87 'End message to user and closing form
88 MsgBox("Your Username is: " & AddUser.Username)
89 MsgBox("Your User data has been saved")
90 Close()
91
92 End Sub
93
94 0 references
95 Private Sub butCreateUserClose_Click(sender As Object, e As EventArgs) Handles butCreateUserClose.Click
96     Me.Close()
97 End Sub
98
99 0 references
100 Private Sub CreateNewUser_Load(sender As Object, e As EventArgs) Handles MyBase.Load
101     comCreateUserAccessLevel.Items.Add("Admin")
102     comCreateUserAccessLevel.Items.Add("Staff")
103 End Sub
104 End Class

```

Form Name: Edit/Delete User

The form is titled "Edit User Details" and is contained within a window titled "EditUser". It features a red background. The fields are arranged vertically on the left side of the form:

- Username: text input field
- Access Level: dropdown menu
- First Name: text input field
- Surname: text input field
- Date of Birth: text input field
- Password: text input field
- Security Question: text input field
- Security Answer: text input field

At the bottom of the form, there are five buttons:

- Save
- Close
- Previous
- Next
- Delete User

Code Listing:

```
1 Public Class EditUser
2
3     Public Place As Integer = 0
4     Private Sub EditUser_Load(sender As Object, e As EventArgs) Handles MyBase.Load
5
6         LoadUsers()
7
8         txtEditUserUsername.Text = AllUsers(Place).Username
9         comEditUserAccessLevel.Text = AllUsers(Place).AccessLevel
10        txtEditUserFName.Text = AllUsers(Place).FirstName
11        txtEditUserSurname.Text = AllUsers(Place).Surname
12        txtEditUserDoB.Text = AllUsers(Place).DoB
13        Try
14            txtEditUserDoB.Text = AllUsers(Place).DoB
15        Catch
16            MsgBox("DoB must be **/**/****")
17        End Try
18        txtEditUserPassword.Text = AllUsers(Place).Password
19        txtEditUserSecurityQ.Text = AllUsers(Place).SecurityQuestion
20        txtEditUserSecurityAns.Text = AllUsers(Place).SecurityAnswer
21
22        comEditUserAccessLevel.Items.Add("Admin")
23        comEditUserAccessLevel.Items.Add("Staff")
24
25    End Sub
26
27    Private Sub butEditUserNext_Click(sender As Object, e As EventArgs) Handles butEditUserNext.Click
28
29        Place = Place + 1
```

```

EditUser.vb [Design] | CreateNewUser.vb* | CreateNewUser.vb [Design]* | ViewAllStock.vb* | ViewAllStock.vb [Design]*
Coursework System (EditUser Events) Load
29 Place = Place + 1
30 If Place = NumUsers Then
31     Place = 0
32 End If
33
34 txtEditUserUsername.Text = AllUsers(Place).Username
35 comEditUserAccessLevel.Text = AllUsers(Place).AccessLevel
36 txtEditUserFName.Text = AllUsers(Place).FirstName
37 txtEditUserSurname.Text = AllUsers(Place).Surname
38
39 Try
40     txtEditUserDoB.Text = AllUsers(Place).DoB
41 Catch
42     MsgBox("DoB must be **/**/****")
43 End Try
44 txtEditUserPassword.Text = AllUsers(Place).Password
45 txtEditUserSecurityQ.Text = AllUsers(Place).SecurityQuestion
46 txtEditUserSecurityAns.Text = AllUsers(Place).SecurityAnswer
47 End Sub
48
49 Private Sub butEditUserPrevious_Click(sender As Object, e As EventArgs) Handles butEditUserPrevious.Click
50
51     Place = Place - 1
52     If Place = -1 Then
53         Place = NumUsers - 1
54     End If
55
56     txtEditUserUsername.Text = AllUsers(Place).Username
57     comEditUserAccessLevel.Text = AllUsers(Place).AccessLevel
58     txtEditUserFName.Text = AllUsers(Place).FirstName
59     txtEditUserSurname.Text = AllUsers(Place).Surname
60
61     Try
62         txtEditUserDoB.Text = AllUsers(Place).DoB
63     Catch
64         MsgBox("DoB must be **/**/****")
65     End Try
66     txtEditUserPassword.Text = AllUsers(Place).Password
67     txtEditUserSecurityQ.Text = AllUsers(Place).SecurityQuestion
68     txtEditUserSecurityAns.Text = AllUsers(Place).SecurityAnswer
69 End Sub
70 Private Sub butEditUserSave_Click(sender As Object, e As EventArgs) Handles butEditUserSave.Click
71
72     'Validation for Access Level'
73     If AllUsers(Place).AccessLevel = "" Then
74         MsgBox("You must select an access level, data not saved. Please try again")
75         Exit Sub
76     End If
77
78     'Validation for First Name'
79     If AllUsers(Place).FirstName = "" Then
80         MsgBox("You must enter a first name, data not saved. Please try again")
81         Exit Sub
82     End If
83
84     'Validation for Surname'
85     If AllUsers(Place).Surname = "" Then
86         MsgBox("You must enter a surname, data not saved. Please try again")

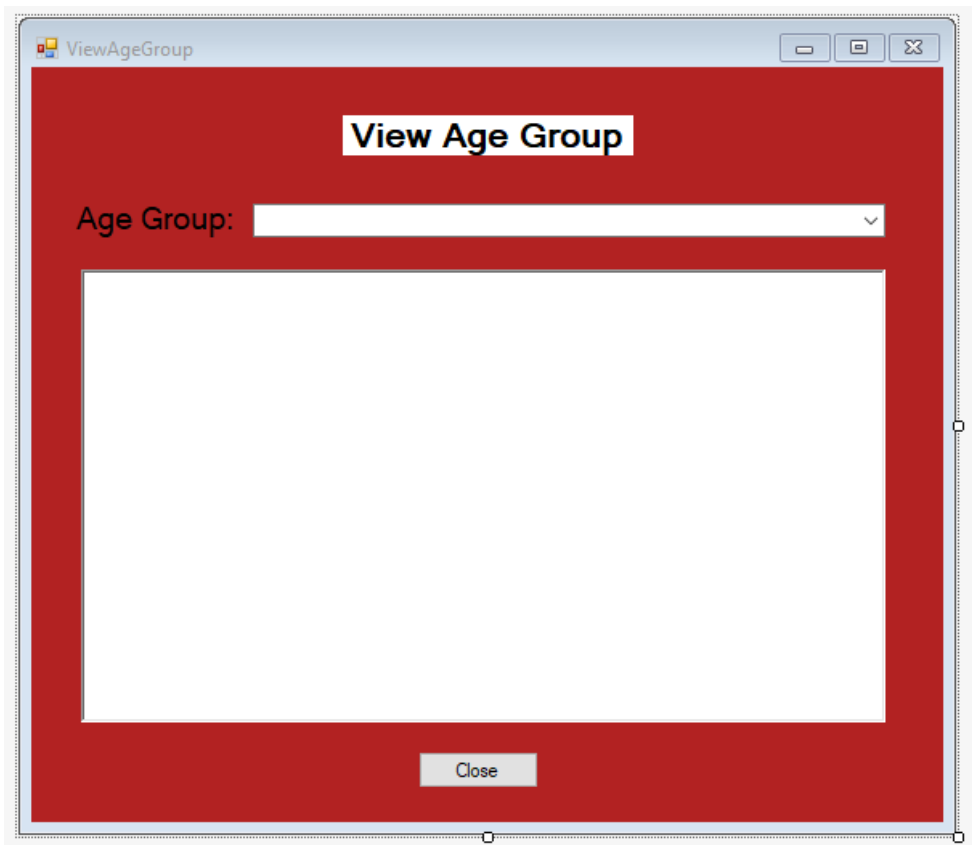
```

```

EditUser.vb [Design] | CreateNewUser.vb* | CreateNewUser.vb [Design]* | ViewAllStock.vb* | ViewAllStock.vb [Design]*
Coursework System (EditUser Events) Load
85     If AllUsers(Place).Surname = "" Then
86         MsgBox("You must enter a surname, data not saved. Please try again")
87         Exit Sub
88     End If
89
90     'Validation for DoB'
91     Dim Correct As Boolean = AllUsers(Place).DoB Like "**/**/"
92     If Correct = False Or Len(AllUsers(Place).DoB) <> 10 Then
93         MsgBox("Date of birth must follow **/**/**** format, data not saved. Please try again")
94         Exit Sub
95     End If
96
97     'Validation for Security Question'
98     If AllUsers(Place).SecurityQuestion = "" Then
99         MsgBox("You must enter a security question, data not saved. Please try again")
100        Exit Sub
101    End If
102
103    'Validation for Security Answer'
104    If AllUsers(Place).SecurityAnswer = "" Then
105        MsgBox("You must enter an answer to your security question, data not saved. Please try again")
106        Exit Sub
107    End If
108
109    'Validation for Password
110    If Len(AllUsers(Place).Password) < 8 Then
111        MsgBox("Your password must be a minimum of 8 characters, please try again")
112        Exit Sub
113    End If
114
115    AllUsers(Place).Username = txtEditUserUsername.Text
116    AllUsers(Place).AccessLevel = comEditUserAccessLevel.Text
117    AllUsers(Place).FirstName = txtEditUserFName.Text
118    AllUsers(Place).Surname = txtEditUserSurname.Text
119    Try
120        AllUsers(Place).DoB = txtEditUserDoB.Text
121    Catch
122        MsgBox("DoB must be **/**/****")
123    End Try
124    AllUsers(Place).Password = txtEditUserPassword.Text
125    AllUsers(Place).SecurityQuestion = txtEditUserSecurityQ.Text
126    AllUsers(Place).SecurityAnswer = txtEditUserSecurityAns.Text
127
128    SaveAllUsers()
129    MsgBox("Your changes have been saved")
130
131 End Sub
132
133 Private Sub btnEditUserClose_Click(sender As Object, e As EventArgs) Handles btnEditUserClose.Click
134     Me.Close()
135 End Sub
136
137 Private Sub btnEditUserDelete_Click(sender As Object, e As EventArgs) Handles btnEditUserDelete.Click
138
139     For x = Place To NumUsers - 1
140         AllUsers(x).Username = AllUsers(x + 1).Username
141         AllUsers(x).FirstName = AllUsers(x + 1).FirstName
142         AllUsers(x).Surname = AllUsers(x + 1).Surname
143         AllUsers(x).DoB = AllUsers(x + 1).DoB
144         AllUsers(x).Password = AllUsers(x + 1).Password
145         AllUsers(x).SecurityQuestion = AllUsers(x + 1).SecurityQuestion
146         AllUsers(x).SecurityAnswer = AllUsers(x + 1).SecurityAnswer
147     Next
148
149     NumUsers = NumUsers - 1
150
151     SaveAllUsers()
152     MsgBox("User deleted")
153
154     Me.Close()
155 End Sub
Class

```

Form Name: View Age Group

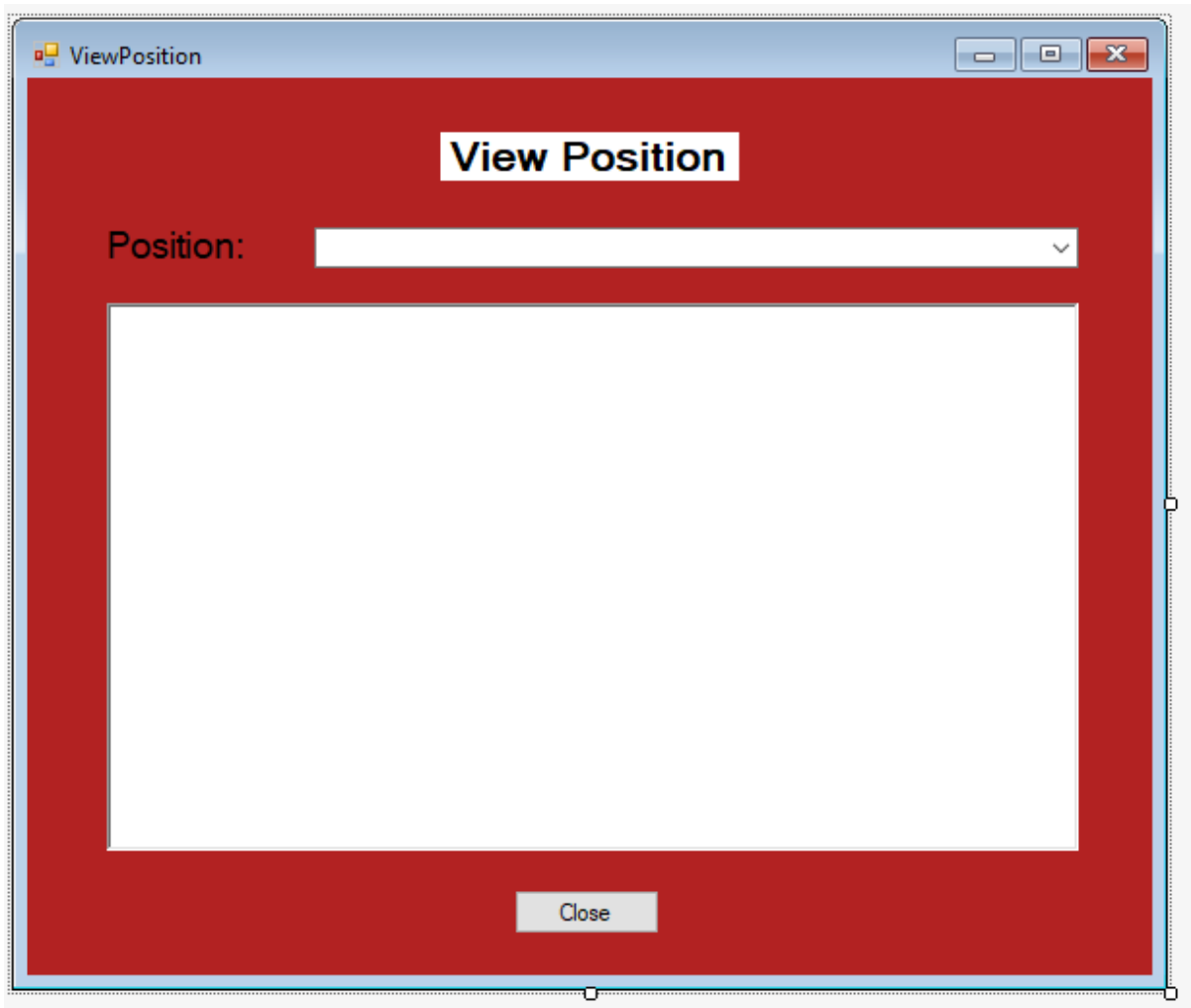


Code Listing:

```
ViewAgeGroup.vb  X ViewAgeGroup.vb [Design]  EditUser.vb  EditUser.vb [Design]  CreateNewUser.vb*  CreateNewUser.vb [Design]*
Coursework System  (ViewAgeGroup Events)  Load
1  Public Class ViewAgeGroup
2      References
3      Private Sub ViewAgeGroup_Load(sender As Object, e As EventArgs) Handles MyBase.Load
4          'Loading in Players and Parents, and adding age groups to the combobox
5          LoadPlayers()
6          LoadParents()
7          comSelectAgeGroup.Items.Add("U8")
8          comSelectAgeGroup.Items.Add("U9")
9          comSelectAgeGroup.Items.Add("U10")
10         comSelectAgeGroup.Items.Add("U11")
11         comSelectAgeGroup.Items.Add("U12")
12         comSelectAgeGroup.Items.Add("U13")
13         comSelectAgeGroup.Items.Add("U14")
14         comSelectAgeGroup.Items.Add("U15")
15         comSelectAgeGroup.Items.Add("U16")
16         comSelectAgeGroup.Items.Add("U17")
17         comSelectAgeGroup.Items.Add("U18")
18         comSelectAgeGroup.Items.Add("U19")
19         comSelectAgeGroup.Items.Add("U21")
20         comSelectAgeGroup.Items.Add("U23")
21     End Sub
22
23     References
24     Private Sub comSelectAgeGroup_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comSelectAgeGroup.SelectedIndexChang
25         'Loading items into the rich text box
26         Dim SelectedAgeGroup As String = comSelectAgeGroup.Text
27         Dim FoundPlayers As System.IO.StreamWriter
28
29         FoundPlayers = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundPlayers.txt"), True)
```

```
ViewAgeGroup.vb* x ViewAgeGroup.vb [Design]* EditUser.vb EditUser.vb [Design] CreateNewUser.vb* CreateNewUser.vb [Design]*
Coursework System butViewAgeGroupClose Click
27 Dim FoundPlayers As System.IO.StreamWriter
28
29 FoundPlayers = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundPlayers.txt"), True)
30
31 For x = 0 To NumPlayers - 1
32     If AllPlayers(x).AgeGroup = SelectedAgeGroup Then
33         For y = 0 To NumParents - 1
34             If AllParents(y).ParentID = AllPlayers(x).ParentID Then
35                 FoundPlayers.WriteLine(AllPlayers(x).FirstName & " " & AllPlayers(x).Surname & ", " & AllPlayers(x).Position)
36             End If
37         Next
38     End If
39 Next
40
41 FoundPlayers.Close()
42
43 'Saving to textfile
44 rtbViewAgeGroup.LoadFile(Dir$("FoundPlayers.txt"), RichTextBoxStreamType.PlainText)
45 FoundPlayers = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundPlayers.txt"), False)
46 FoundPlayers.Close()
47
48 End Sub
49
50
51
52
53 0 references
54 Private Sub butViewAgeGroupClose_Click(sender As Object, e As EventArgs) Handles butViewAgeGroupClose.Click
55     Me.Close()
56 End Sub
```

Form Name: View Position



Code Listing:

```
ViewPosition.vb x ViewPosition.vb [Design] ViewAgeGroup.vb* ViewAgeGroup.vb [Design]* EditUser.vb EditUser.vb [Design]
Coursework System (ViewPosition Events) Load
1 Public Class ViewPosition
2 Private Sub ViewPosition_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3
4 'Loading in all players and parents, and adding positions to combobox
5 LoadPlayers()
6 LoadParents()
7 comSelectPosition.Items.Add("Goalkeeper")
8 comSelectPosition.Items.Add("Defender")
9 comSelectPosition.Items.Add("Midfielder")
10 comSelectPosition.Items.Add("Forward")
11
12 End Sub
13
14 Private Sub comSelectPosition_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comSelectPosition.SelectedIndexChang
15
16 'Loading items to rich text box
17 Dim SelectedPosition As String = comSelectPosition.Text
18 Dim FoundPositionPlayers As System.IO.StreamWriter
19
20 FoundPositionPlayers = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundPosition.txt"), True)
21
22 For x = 0 To NumPlayers - 1
23 If AllPlayers(x).Position = SelectedPosition Then
24 For y = 0 To NumParents - 1
25 If AllParents(y).ParentID = AllPlayers(x).ParentID Then
26 FoundPositionPlayers.WriteLine(AllPlayers(x).FirstName & " " & AllPlayers(x).Surname & ", " & AllPlayers(x).P
27 End If
28 Next
29
```

```
ViewPosition.vb*  ViewPosition.vb [Design]*  ViewAgeGroup.vb*  ViewAgeGroup.vb [Design]*  EditUser.vb  EditUser.vb [Design]
Coursework System  ViewPosition  InitializeComponent
26      FoundPositionPlayers.WriteLine(AllPlayers(x).FirstName & " " & AllPlayers(x).Surname & ", " & AllPlayers(x).I
27      End If
28      Next
29
30      End If
31
32      Next
33
34      FoundPositionPlayers.Close()
35
36      'Saving into textfile
37      rtbViewPosition.LoadFile(Dir$("FoundPosition.txt"), RichTextBoxStreamType.PlainText)
38      FoundPositionPlayers = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundPosition.txt"), False)
39      FoundPositionPlayers.Close()
40  End Sub
41
42  O references
43  Private Sub butViewPositionClose_Click(sender As Object, e As EventArgs) Handles butViewPositionClose.Click
44      Me.Close()
45  End Sub
End Class
```

Form Name: View Player Contact Details



Code Listing:

```
ViewPlayerContactDetails.vb x ViewPlayerContactDetails.vb [Design] ViewPosition.vb* ViewPosition.vb [Design]* ViewAgeGroup.vb*
Coursework System (ViewPlayerContactDetails Events) Load

3 references
Public Class ViewPlayerContactDetails
0 references
Private Sub butSearchPlayerContactInfo_Click(sender As Object, e As EventArgs) Handles butSearchPlayerContactInfo.Click
3
4 'Loading players and parents into form
5 LoadPlayers()
6 LoadParents()
7
8 'declaring variables and displaying input boxes
9 Dim SearchPlayerFName As String
10 Dim SearchPlayerSurname As String
11 Dim SearchParentID As String = "-1"
12 Dim PlayerName As String = ""
13 SearchPlayerFName = InputBox("Enter Player's First Name:")
14 SearchPlayerSurname = InputBox("Enter Player's Surname:")
15
16
17 For x = 0 To NumPlayers - 1
18     If AllPlayers(x).FirstName.ToUpper = SearchPlayerFName.ToUpper And AllPlayers(x).Surname.ToUpper = SearchPlayerSurname.
19         SearchParentID = AllPlayers(x).ParentID
20         PlayerName = AllPlayers(x).FirstName & " " & AllPlayers(x).Surname
21         Exit For
22     End If
23 Next
24
25 'Displaying record in message box
26 If SearchParentID = "-1" Then
27     MsgBox("Player Not Found")
28 Else
29     For y = 0 To NumParents - 1
30         If AllParents(y).ParentID = SearchParentID Then
31             MsgBox("*****RECORD FOUND*****" + vbCrLf + vbCrLf + PlayerName + vbCrLf + vbCrLf + "Parent Name: " & AllParents
32         End If
33     Next
34 End If
35 End Sub
36
```

Form Name: View Subs/Add Payments

View Subs Paid and Add Payments

Select Team

Make a Payment

Year Group: Amount Paid: Make Payment

PlayerID: Payent for Player: Close

Code Listing:

```
View subs Paid and Add Payments.vb* - X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Design]*
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
1 Public Class View_subs_Paid_and_Add_Payments
2
3     Public NumFoundinGroup As Integer = 0
4     Public subsDue As Double
5     Public AgeGroup As String
6
7     Private Sub View_subs_Paid_and_Add_Payments_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8         comPlayerAgeGroup.Items.Add("U8")
9         comPlayerAgeGroup.Items.Add("U9")
10        comPlayerAgeGroup.Items.Add("U10")
11        comPlayerAgeGroup.Items.Add("U11")
12        comPlayerAgeGroup.Items.Add("U12")
13        comPlayerAgeGroup.Items.Add("U13")
14        comPlayerAgeGroup.Items.Add("U14")
15        comPlayerAgeGroup.Items.Add("U15")
16        comPlayerAgeGroup.Items.Add("U16")
17        comPlayerAgeGroup.Items.Add("U17")
18        comPlayerAgeGroup.Items.Add("U18")
19        comPlayerAgeGroup.Items.Add("U19")
20        comPlayerAgeGroup.Items.Add("U21")
21        comPlayerAgeGroup.Items.Add("U23")
22
23        comPaymentGroup.Items.Add("U8")
24        comPaymentGroup.Items.Add("U9")
25        comPaymentGroup.Items.Add("U10")
26        comPaymentGroup.Items.Add("U11")
27        comPaymentGroup.Items.Add("U12")
28        comPaymentGroup.Items.Add("U13")
29        comPaymentGroup.Items.Add("U14")
30        comPaymentGroup.Items.Add("U15")
```

```

View subs Paid and Add Payments.vb* - X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Design]*
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
28 comPaymentGroup.Items.Add("U13")
29 comPaymentGroup.Items.Add("U14")
30 comPaymentGroup.Items.Add("U15")
31 comPaymentGroup.Items.Add("U16")
32 comPaymentGroup.Items.Add("U17")
33 comPaymentGroup.Items.Add("U18")
34 comPaymentGroup.Items.Add("U19")
35 comPaymentGroup.Items.Add("U21")
36 comPaymentGroup.Items.Add("U23")
37
38
39 Dim file9 As System.IO.StreamReader
40 file9 = My.Computer.FileSystem.OpenTextFileReader(Dir$("SubsDuePerYear.txt"))
41 SubsDue = file9.ReadLine()
42 file9.Close()
43
44 End Sub
45
3 references
46 Private Function DisplaySubs()
47     LoadPlayers()
48
49     Dim file As System.IO.StreamReader
50     Dim line As String
51     Dim parts(0 To 1) As String
52     Dim PlayerId As String
53     Dim AmountPAID As Double
54
55     'add header to text file and delete previous contents
56
57     Dim FoundSubsInfo As System.IO.StreamWriter
58
View subs Paid and Add Payments.vb* - X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Design]*
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
56
57 Dim FoundSubsInfo As System.IO.StreamWriter
58 FoundSubsInfo = My.Computer.FileSystem.OpenTextFileWriter(Dir$("FoundSubsInfo.txt"), False)
59 FoundSubsInfo.WriteLine("***** Subs Data for " & comPlayerAgeGroup.Text & " **")
60 FoundSubsInfo.Close()
61
62
63 file = My.Computer.FileSystem.OpenTextFileReader(Dir$("SubsRecords.txt"))
64 Do
65     line = file.ReadLine()
66     parts = line.Split(",")
67     PlayerId = parts(0)
68     AmountPAID = parts(1)
69
70
71 For x = 0 To NumPlayers
72     If PlayerId = AllPlayers(x).PlayerID Then
73         If AllPlayers(x).AgeGroup = AgeGroup Then
74             Dim FoundSubs As System.IO.StreamWriter
75             FoundSubs = My.Computer.FileSystem.OpenTextFileWriter(Dir$("FoundSubsInfo.txt"), True)
76             FoundSubs.WriteLine(PlayerId.PadRight(6) & " " & AllPlayers(x).FirstName.PadRight(15) & " " & AllPlaye
77             FoundSubs.Close()
78         End If
79     End If
80 End For
81 Next
82
83 Loop Until (file.EndOfStream)
84 file.Close()
85

```

```

View subs Paid and Add Payments.vb* -> X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Design]*
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
83 Loop Until (file.EndOfStream)
84 file.Close()
85
86 rtbSubsInfo.LoadFile(Dir$("FoundSubsInfo.txt"), RichTextBoxStreamType.PlainText)
87 End Function
88
References
89 Private Sub comPlayerAgeGroup_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comPlayerAgeGroup.SelectedIndexChanged
90 AgeGroup = comPlayerAgeGroup.Text
91 DisplaySubs()
92 comPlayerID.Items.Clear()
93 comPaymentGroup.Text = comPlayerAgeGroup.Text
94 End Sub
95
References
96 Private Sub comPaymentGroup_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comPaymentGroup.SelectedIndexChanged
97 AgeGroup = comPaymentGroup.Text
98 comPlayerAgeGroup.Text = comPaymentGroup.Text
99 DisplaySubs()
100 LoadPlayers()
101 comPlayerID.Items.Clear()
102
103 For x = 0 To NumPlayers - 1
104 If AllPlayers(x).AgeGroup = AgeGroup Then
105 comPlayerID.Items.Add(AllPlayers(x).PlayerID)
106 End If
107 Next
108 End Sub
109

```

```

View subs Paid and Add Payments.vb* -> X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Des
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
109
References
110 Private Sub cmdMakePayment_Click(sender As Object, e As EventArgs) Handles cmdMakePayment.Click
111 If comPlayerID.Text = "" Then
112 MsgBox("Select PlayerId first, paymnet not processed")
113 Exit Sub
114 End If
115 If txtAmount.Text = "" Or txtAmount.Text < 1 Then
116 MsgBox("Please enter an amount of £1 minimum before clicking Make Payment")
117 Exit Sub
118 End If
119
120 Dim Playerid As String = comPlayerID.Text
121 Dim AmountPaid As Double = txtAmount.Text
122 LoadSubPayments()
123 For x = 0 To NumSubPays - 1
124 If AllSubs(x).PlyerId = Playerid Then
125 AllSubs(x).TotalPaid = AllSubs(x).TotalPaid + AmountPaid
126 SaveSubs()
127 DisplaySubs()
128 MsgBox("Payment Processed")
129 lblPaymentForPlayer.Text = "Payent for Player: "
130 comPlayerID.Text = ""
131 txtAmount.Text = ""
132
133 Exit Sub
134 End If
135 Next
136 End Sub
137

```

```

View subs Paid and Add Payments.vb* -> X View subs Paid and...yments.vb [Design]* ViewPlayerContactDetails.vb* ViewPlayerContactDetails.vb [Design]*
Coursework System (View_subs_Paid_and_Add_Payments Events) Load
133 Exit Sub
134 End If
135 Next
136 End Sub
137
References
139 Private Sub comPlayerID_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comPlayerID.SelectedIndexChanged
140 LoadPlayers()
141 Dim Playerid As String = comPlayerID.Text
142
143 For x = 0 To NumPlayers - 1
144 If Playerid = AllPlayers(x).PlayerID Then
145 lblPaymentForPlayer.Text = "Payent for Player: " & AllPlayers(x).FirstName & " " & AllPlayers(x).Surname
146 End If
147 Next
148 End Sub
149
References
150 Private Sub butViewSubsClose_Click(sender As Object, e As EventArgs) Handles butViewSubsClose.Click
151 Me.Close()
152 End Sub
153 End Class

```


Form Name: View An Age Group's Coach



Code Listing:

```
ViewAgeGroupCoach.vb  ViewAgeGroupCoach.vb [Design]  View subs Paid and Add Payments.vb*  View subs Paid and...yments.vb [Design]*
Coursework System  (ViewAgeGroupCoach Events)  Load
1  Public Class ViewAgeGroupCoach
2      Private Sub ViewAgeGroupCoach_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3          LoadCoaches()
4
5          comSelectAgeGroupCoach.Items.Add("U8")
6          comSelectAgeGroupCoach.Items.Add("U9")
7          comSelectAgeGroupCoach.Items.Add("U10")
8          comSelectAgeGroupCoach.Items.Add("U11")
9          comSelectAgeGroupCoach.Items.Add("U12")
10         comSelectAgeGroupCoach.Items.Add("U13")
11         comSelectAgeGroupCoach.Items.Add("U14")
12         comSelectAgeGroupCoach.Items.Add("U15")
13         comSelectAgeGroupCoach.Items.Add("U16")
14         comSelectAgeGroupCoach.Items.Add("U17")
15         comSelectAgeGroupCoach.Items.Add("U18")
16         comSelectAgeGroupCoach.Items.Add("U19")
17         comSelectAgeGroupCoach.Items.Add("U21")
18         comSelectAgeGroupCoach.Items.Add("U23")
19
20     End Sub
21
22
23     Private Sub comSelectAgeGroupCoach_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comSelectAgeGroupCoach.Selecte
24
25         Dim SelectedAgeGroupCoach As String = comSelectAgeGroupCoach.Text
26         Dim FoundCoach As System.IO.StreamWriter
27
28         FoundCoach = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundCoach.txt"), True)
29
```

```
ViewAgeGroupCoach.vb* -> X ViewAgeGroupCoach.vb [Design]* View subs Paid and Add Payments.vb* View subs Paid and...yments.vb [Design]*
Coursework System ViewAgeGroupCoach InitializeComponent
21 End Sub
22
23 Private Sub comSelectAgeGroupCoach_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comSelectAgeGroupCoach.Selecte
24
25     Dim SelectedAgeGroupCoach As String = comSelectAgeGroupCoach.Text
26     Dim FoundCoach As System.IO.StreamWriter
27
28     FoundCoach = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundCoach.txt"), True)
29
30     For x = 0 To NumCoaches - 1
31         If AllCoaches(x).AgeGroupCoached = SelectedAgeGroupCoach Then
32             FoundCoach.WriteLine(AllCoaches(x).CoachID & ", " & AllCoaches(x).FirstName & " " & AllCoaches(x).Surname & ", " & All
33         End If
34     Next
35
36     FoundCoach.Close()
37
38     rtbViewAgeGroupCoach.LoadFile(Dir$("FoundCoach.txt"), RichTextBoxStreamType.PlainText)
39     FoundCoach = My.Computer.FileSystem.OpenTextFileWriter(Dir("FoundCoach.txt"), False)
40     FoundCoach.Close()
41 End Sub
42
43
44 Private Sub butViewAgeGroupCoachClose_Click(sender As Object, e As EventArgs) Handles butViewAgeGroupCoachClose.Click
45     Me.Close()
46 End Sub
47 End Class
```

Module Code: Public Parents

```
Public Structure Parents
```

```
Public ParentID As String  
Public FirstName As String  
Public Surname As String  
Public DoB As Date  
Public Postcode As String  
Public Street As String  
Public Town As String  
Public City As String  
Public PhoneNum As String  
Public Email As String
```

```
End Structure
```

```
Public AllParents(0 To 999) As Parents
```

```
Public NumParents As Integer = 0
```

Module Code: Public Coaches

```
Public Structure Coaches
```

```
    Public CoachID As String  
    Public FirstName As String  
    Public Surname As String  
    Public DoB As Date  
    Public Postcode As String  
    Public Street As String  
    Public Town As String  
    Public City As String  
    Public PhoneNo As String  
    Public Email As String  
    Public AgeGroupCoached As String
```

```
End Structure
```

```
Public AllCoaches(0 To 999) As Coaches
```

```
Public NumCoaches As Integer = 0
```

Module Code: Public Players

2 references

Public Structure Players

```
Public PlayerID As String
Public FirstName As String
Public Surname As String
Public DoB As Date
Public AgeGroup As String
Public Position As String
Public ParentID As String
```

End Structure

Public AllPlayers(0 To 999) As Players

Public FoundGroup(0 To 30) As Players

Public NumPlayers As Integer = 0

Module Code: Public Stock

1 reference

```
Public Structure Stock
```

```
    Public StockID As String  
    Public Description As String  
    Public Quantity As String  
    Public ExtraNotes As String
```

```
End Structure
```

```
Public AllStock(0 To 999) As Stock
```

```
Public NumStock As Integer = 0
```

Module Code: Public SubPayments

1 reference

```
Public Structure SubPayments
  Public PlyerId As String
  Public TotalPaid As Double
End Structure
Public AllSubs(0 To 999) As SubPayments
Public NumSubPays As Integer = 0
```

Module Code: LoadSubPayments()

```
1 Reference
Public Function LoadSubPayments()
    Dim file As System.IO.StreamReader
    Dim line As String
    Dim parts(0 To 1) As String

    file = My.Computer.FileSystem.OpenTextFileReader(Dir$("SubsRecords.txt"))

    Do
        line = file.ReadLine()
        parts = line.Split(",")

        AllSubs(NumSubPays).PlyerId = parts(0)
        AllSubs(NumSubPays).TotalPaid = parts(1)

        NumSubPays = NumSubPays + 1
    Loop Until (file.EndOfStream)
    file.Close()
End Function
```

Module Code: SaveSubs()

```
1reference  
Public Function SaveSubs()  
    Dim SaveSubPaymentFile As System.IO.StreamWriter  
    SaveSubPaymentFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("SubsRecords.txt"), False)  
    For y = 0 To (NumSubPays - 1)  
        SaveSubPaymentFile.WriteLine(AllSubs(y).PlyerId & "," & AllSubs(y).TotalPaid)  
    Next y  
    SaveSubPaymentFile.Close()  
End Function
```

Module Code: LoadUsers()

```
Public Function LoadUsers()  
  
    Dim file As System.IO.StreamReader  
    Dim line As String  
    Dim parts(0 To 7) As String  
  
    file = My.Computer.FileSystem.OpenTextFileReader(Dir$("All_Users.txt"))  
  
    Do  
        line = file.ReadLine()  
  
        'splits each readline into parts and puts in array  
  
        parts = line.Split(",")  
  
        AllUsers(NumUsers).Username = parts(0)  
        AllUsers(NumUsers).AccessLevel = parts(1)  
        AllUsers(NumUsers).FirstName = parts(2)  
        AllUsers(NumUsers).Surname = parts(3)  
        AllUsers(NumUsers).DoB = parts(4)  
        AllUsers(NumUsers).Password = parts(5)  
        AllUsers(NumUsers).SecurityQuestion = parts(6)  
        AllUsers(NumUsers).SecurityAnswer = parts(7)  
  
        NumUsers = NumUsers + 1  
  
    Loop Until (file.EndOfStream)  
    file.Close()  
  
End Function
```

Module Code: SaveAllUsers()

2 references

```
Public Function SaveAllUsers()  
  
    Dim SaveUserFile As System.IO.StreamWriter  
    SaveUserFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Users.txt"), False)  
    For y = 0 To (NumUsers - 1)  
        SaveUserFile.WriteLine(AllUsers(y).Username & "," & AllUsers(y).AccessLevel & "," & AllUsers(y).FirstName & "," & AllUsers(y).LastName)  
    Next y  
    SaveUserFile.Close()  
  
End Function
```

Module Code: LoadParents()

```
Public Function LoadParents()  
  
    Dim file As System.IO.StreamReader  
    Dim line As String  
    Dim parts(0 To 9) As String  
    Dim position As Integer = 0  
    file = My.Computer.FileSystem.OpenTextFileReader(Dir$("All_Parents.txt"))  
    NumParents = 0  
    Do  
        line = file.ReadLine()  
        parts = line.Split(",")  
        AllParents(position).ParentID = parts(0)  
        AllParents(position).FirstName = parts(1)  
        AllParents(position).Surname = parts(2)  
        AllParents(position).DoB = parts(3)  
        AllParents(position).Postcode = parts(4)  
        AllParents(position).Street = parts(5)  
        AllParents(position).Town = parts(6)  
        AllParents(position).City = parts(7)  
        AllParents(position).PhoneNum = parts(8)  
        AllParents(position).Email = parts(9)  
  
        position = position + 1  
  
        NumParents = NumParents + 1  
    Loop Until (file.EndOfStream)  
  
    file.Close()  
End Function
```

Module Code: LoadPlayers()

> REFERENCES

```
Public Function LoadPlayers()  
  
    NumPlayers = 0  
    Dim file2 As System.IO.StreamReader  
    Dim line2 As String  
    Dim parts2(0 To 6) As String  
    Dim position2 As Integer = 0  
    file2 = My.Computer.FileSystem.OpenTextFileReader(Dir$("All_Players.txt"))  
  
    Do  
        line2 = file2.ReadLine()  
        parts2 = line2.Split(",")  
        AllPlayers(position2).PlayerID = parts2(0)  
        AllPlayers(position2).FirstName = parts2(1)  
        AllPlayers(position2).Surname = parts2(2)  
        AllPlayers(position2).DoB = parts2(3)  
        AllPlayers(position2).AgeGroup = parts2(4)  
        AllPlayers(position2).Position = parts2(5)  
        AllPlayers(position2).ParentID = parts2(6)  
  
        position2 = position2 + 1  
  
        NumPlayers = NumPlayers + 1  
    Loop Until (file2.EndOfStream)  
  
    file2.Close()  
End Function
```

Module Code: LoadCoaches()

```
Public Function LoadCoaches()  
    NumCoaches = 0  
    Dim file3 As System.IO.StreamReader  
    Dim line3 As String  
    Dim parts3(0 To 10) As String  
    Dim position3 As Integer = 0  
    file3 = My.Computer.FileSystem.OpenTextFileReader(Dir$("All_Coaches.txt"))  
  
    Do  
        line3 = file3.ReadLine()  
        parts3 = line3.Split(",")  
        AllCoaches(position3).CoachID = parts3(0)  
        AllCoaches(position3).FirstName = parts3(1)  
        AllCoaches(position3).Surname = parts3(2)  
        AllCoaches(position3).DoB = parts3(3)  
        AllCoaches(position3).Postcode = parts3(4)  
        AllCoaches(position3).Street = parts3(5)  
        AllCoaches(position3).Town = parts3(6)  
        AllCoaches(position3).City = parts3(7)  
        AllCoaches(position3).PhoneNo = parts3(8)  
        AllCoaches(position3).Email = parts3(9)  
        AllCoaches(position3).AgeGroupCoached = parts3(10)  
  
        position3 = position3 + 1  
  
        NumCoaches = NumCoaches + 1  
    Loop Until (file3.EndOfStream)  
  
    file3.Close()  
End Function
```

Module Code: Public User

1 reference

```
Public Structure User
    Public AccessLevel As String
    Public Username As String
    Public FirstName As String
    Public Surname As String
    Public DoB As String
    Public Password As String
    Public SecurityQuestion As String
    Public SecurityAnswer As String
End Structure
```

```
Public AllUsers(0 To 999) As User
```

```
Public NumUsers As Integer = 0
```

Module Name: LoadStock()

Public Function

LoadStock()

```
NumStock = 0
Dim file4 As System.IO.StreamReader
Dim line4 As String
Dim parts4(0 To 9) As String
Dim position4 As Integer = 0
file4 = My.Computer.FileSystem.OpenTextFileReader(Dir$("All_Stock.txt"))

Do
    line4 = file4.ReadLine()
    parts4 = line4.Split(",")
    AllStock(position4).StockID = parts4(0)
    AllStock(position4).Description = parts4(1)
    AllStock(position4).Quantity = parts4(2)
    AllStock(position4).ExtraNotes = parts4(3)

    position4 = position4 + 1

    NumStock = NumStock + 1
Loop Until (file4.EndOfStream)

file4.Close()
```

End Function

Module Code: SaveAllParents()

```
3 references
Public Function SaveAllParents()

    Dim SaveParentFile As System.IO.StreamWriter
    SaveParentFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Parents.txt"), False)
    For y = 0 To (NumParents - 1)
        SaveParentFile.WriteLine(AllParents(y).ParentID & "," & AllParents(y).FirstName & "," & AllParents(y).Surname & "," & AllPa
    Next y
    SaveParentFile.Close()

End Function
```

Module Code: SaveAllCoaches()

```
Public Function SaveAllCoaches()  
    Dim SaveCoachFile As System.IO.StreamWriter  
    SaveCoachFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Coaches.txt"), False)  
    For y = 0 To (NumCoaches - 1)  
        SaveCoachFile.WriteLine(AllCoaches(y).CoachID & "," & AllCoaches(y).FirstName & "," & AllCoaches(y).Surname & "," & AllCoaches(y).Address)  
    Next y  
    SaveCoachFile.Close()  
End Function
```

Module Code: SaveAllPlayers()

```
Public Function SaveAllPlayers()  
    Dim SavePlayerFile As System.IO.StreamWriter  
    SavePlayerFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Players.txt"), False)  
    For y = 0 To (NumPlayers - 1)  
        SavePlayerFile.WriteLine(AllPlayers(y).PlayerID & "," & AllPlayers(y).FirstName & "," & AllPlayers(y).Surname & "," & All  
    Next y  
    SavePlayerFile.Close()  
End Function
```

Module Code: SaveAllStock()

3 references

```
Public Function SaveAllStock()  
  
    Dim SaveStockFile As System.IO.StreamWriter  
    SaveStockFile = My.Computer.FileSystem.OpenTextFileWriter(Dir$("All_Stock.txt"), False)  
    For y = 0 To (NumStock - 1)  
        SaveStockFile.WriteLine(AllStock(y).StockID & "," & AllStock(y).Description & "," & AllStock(y).Quantity & "," & AllStoc  
    Next y  
    SaveStockFile.Close()  
  
End Function
```

Module Code: BubblesortCoach()

```
Public Function BubblesortCoach()  
  
    Dim Temp1 As String  
    Dim Temp2 As String  
    Dim Temp3 As String  
    Dim Temp4 As String  
    Dim Temp5 As String  
    Dim Temp6 As String  
    Dim Temp7 As String  
    Dim Temp8 As String  
    Dim Temp9 As String  
    Dim Temp10 As String  
    Dim Temp11 As String  
  
    'Bubble Sort data by Name  
  
    For x = 0 To NumCoaches - 2  
        For y = 0 To NumCoaches - 2  
            If AllCoaches(y).Surname > AllCoaches(y + 1).Surname Then  
                Temp1 = AllCoaches(y).CoachID  
                Temp2 = AllCoaches(y).FirstName  
                Temp3 = AllCoaches(y).Surname  
                Temp4 = AllCoaches(y).DoB  
                Temp5 = AllCoaches(y).Postcode  
                Temp6 = AllCoaches(y).Street  
                Temp7 = AllCoaches(y).Town  
                Temp8 = AllCoaches(y).City  
                Temp9 = AllCoaches(y).PhoneNo  
                Temp10 = AllCoaches(y).Email  
                Temp11 = AllCoaches(y).AgeGroupCoached  
  
                AllCoaches(y).CoachID = AllCoaches(y + 1).CoachID  
                AllCoaches(y).FirstName = AllCoaches(y + 1).FirstName  
                AllCoaches(y).Surname = AllCoaches(y + 1).Surname  
                AllCoaches(y).DoB = AllCoaches(y + 1).DoB  
                AllCoaches(y).Postcode = AllCoaches(y + 1).Postcode  
                AllCoaches(y).Street = AllCoaches(y + 1).Street  
                AllCoaches(y).Town = AllCoaches(y + 1).Town  
                AllCoaches(y).City = AllCoaches(y + 1).City  
                AllCoaches(y).PhoneNo = AllCoaches(y + 1).PhoneNo  
                AllCoaches(y).Email = AllCoaches(y + 1).Email  
                AllCoaches(y).AgeGroupCoached = AllCoaches(y + 1).AgeGroupCoached  
            End If  
        Next y  
    Next x  
End Function
```

```
AllCoaches(y + 1).CoachID = Temp1
AllCoaches(y + 1).FirstName = Temp2
AllCoaches(y + 1).Surname = Temp3
AllCoaches(y + 1).DoB = Temp4
AllCoaches(y + 1).Postcode = Temp5
AllCoaches(y + 1).Street = Temp6
AllCoaches(y + 1).Town = Temp7
AllCoaches(y + 1).City = Temp8
AllCoaches(y + 1).PhoneNo = Temp9
AllCoaches(y + 1).Email = Temp10
AllCoaches(y + 1).AgeGroupCoached = Temp11
```

```
End If
```

```
Next
```

```
Next
```

```
End Function
```

Module Code: BubblesortParents()

```
Public Function BubblesortParents()  
  
    Dim Temp1 As String  
    Dim Temp2 As String  
    Dim Temp3 As String  
    Dim Temp4 As String  
    Dim Temp5 As String  
    Dim Temp6 As String  
    Dim Temp7 As String  
    Dim Temp8 As String  
    Dim Temp9 As String  
    Dim Temp10 As String  
  
    'Bubble Sort data by Name  
  
    For x = 0 To NumParents - 2  
        For y = 0 To NumParents - 2  
  
            If AllParents(y).Surname > AllParents(y + 1).Surname Then  
                Temp1 = AllParents(y).ParentID  
                Temp2 = AllParents(y).FirstName  
                Temp3 = AllParents(y).Surname  
                Temp4 = AllParents(y).DoB  
                Temp5 = AllParents(y).Postcode  
                Temp6 = AllParents(y).Street  
                Temp7 = AllParents(y).Town  
                Temp8 = AllParents(y).City  
                Temp9 = AllParents(y).PhoneNum  
                Temp10 = AllParents(y).Email  
  
                AllParents(y).ParentID = AllParents(y + 1).ParentID  
                AllParents(y).FirstName = AllParents(y + 1).FirstName  
                AllParents(y).Surname = AllParents(y + 1).Surname  
                AllParents(y).DoB = AllParents(y + 1).DoB  
                AllParents(y).Postcode = AllParents(y + 1).Postcode  
                AllParents(y).Street = AllParents(y + 1).Street  
                AllParents(y).Town = AllParents(y + 1).Town  
                AllParents(y).City = AllParents(y + 1).City  
                AllParents(y).PhoneNum = AllParents(y + 1).PhoneNum  
                AllParents(y).Email = AllParents(y + 1).Email  
  
            End If  
  
        Next y  
    Next x  
End Function
```

```
AllParents(y + 1).ParentID = Temp1
AllParents(y + 1).FirstName = Temp2
AllParents(y + 1).Surname = Temp3
AllParents(y + 1).DoB = Temp4
AllParents(y + 1).Postcode = Temp5
AllParents(y + 1).Street = Temp6
AllParents(y + 1).Town = Temp7
AllParents(y + 1).City = Temp8
AllParents(y + 1).PhoneNum = Temp9
AllParents(y + 1).Email = Temp10
End If
Next
Next
End Function
```

Module Code: BubblesortPlayers()

1 reference

```
Public Function BubblesortPlayers()
```

```
Dim Temp1 As String  
Dim Temp2 As String  
Dim Temp3 As String  
Dim Temp4 As String  
Dim Temp5 As String  
Dim Temp6 As String  
Dim Temp7 As String
```

```
'Bubble Sort data by Name
```

```
For x = 0 To NumPlayers - 2
```

```
For y = 0 To NumPlayers - 2
```

```
    If AllPlayers(y).Surname > AllPlayers(y + 1).Surname Then
```

```
        Temp1 = AllPlayers(y).PlayerID  
        Temp2 = AllPlayers(y).FirstName  
        Temp3 = AllPlayers(y).Surname  
        Temp4 = AllPlayers(y).DoB  
        Temp5 = AllPlayers(y).AgeGroup  
        Temp6 = AllPlayers(y).Position  
        Temp7 = AllPlayers(y).ParentID
```

```
        AllPlayers(y).PlayerID = AllPlayers(y + 1).PlayerID  
        AllPlayers(y).FirstName = AllPlayers(y + 1).FirstName  
        AllPlayers(y).Surname = AllPlayers(y + 1).Surname  
        AllPlayers(y).DoB = AllPlayers(y + 1).DoB  
        AllPlayers(y).AgeGroup = AllPlayers(y + 1).AgeGroup  
        AllPlayers(y).Position = AllPlayers(y + 1).Position  
        AllPlayers(y).ParentID = AllPlayers(y + 1).ParentID
```

```
        AllPlayers(y + 1).PlayerID = Temp1  
        AllPlayers(y + 1).FirstName = Temp2  
        AllPlayers(y + 1).Surname = Temp3  
        AllPlayers(y + 1).DoB = Temp4  
        AllPlayers(y + 1).AgeGroup = Temp5  
        AllPlayers(y + 1).Position = Temp6  
        AllPlayers(y + 1).ParentID = Temp7
```

```
    End If
```

```
Next
```

```
Next
```

```
End Function
```

Module Code: BubblesortStock()

```
Public Function BubblesortStock()  
  
    Dim Temp1 As String  
    Dim Temp2 As String  
    Dim Temp3 As String  
    Dim Temp4 As String  
  
    'Bubble Sort data by Name  
  
    For x = 0 To NumStock - 2  
        For y = 0 To NumStock - 2  
            If AllStock(y).Description > AllStock(y + 1).Description Then  
                Temp1 = AllStock(y).StockID  
                Temp2 = AllStock(y).Description  
                Temp3 = AllStock(y).Quantity  
                Temp4 = AllStock(y).ExtraNotes  
  
                AllStock(y).StockID = AllStock(y + 1).StockID  
                AllStock(y).Description = AllStock(y + 1).Description  
                AllStock(y).Quantity = AllStock(y + 1).Quantity  
                AllStock(y).ExtraNotes = AllStock(y + 1).ExtraNotes  
  
                AllStock(y + 1).StockID = Temp1  
                AllStock(y + 1).Description = Temp2  
                AllStock(y + 1).Quantity = Temp3  
                AllStock(y + 1).ExtraNotes = Temp4  
            End If  
        Next  
    Next  
End Function
```