

**Happy Earth**

DELLESHAN REDDY

12PR

13399

**Scenario:**

The challenges we face as the new generation are all created by the ignorant old generation, too greedy to stop. It is our job as the new generation to put an end to their problems that they keep creating. One of the biggest problems that they continue to ignore is the destruction of the environment and the overall increase in the earth’s temperature.

**Task:**

Design an application social management and distribution of information regarding scheduling of marine environmental clean-ups efforts and overall help save the world from environmental issues.

**Purpose:**

The application I will design is planned to slow down the destruction of our environment by getting the users to be more aware and live better. This will be done by giving the user the option to participate in beach clean-ups with the rewarding being for participating is , obviously a cleaner environment but as well as providing the user with credits, that can only be attained in this way, that they can use to purchase partnered brands’ coupons.

**Solution:**

The app will:

Inform the user with facts and articles about the harmful effects of human activity on the environment and how they can better their life.

Notify the user about upcoming events, such as beach clean-ups, as well as provide credits when attending these events

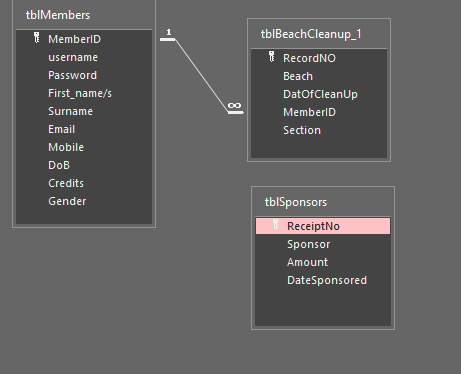
Advertise eco-friendly brands as alternatives to power supplies, food, cars, food

This will all be done using, the programming app, Delphi 2010 IDE. The app will include but not limited to:

* a relational and normalised database with 3 or more linked tables with adequate field sizes and formats and allows admin users for CRUD functions.
* a multi-functional GUI with minimum 3 forms that is user friendly and both visually appeasing and contains the relevant components to perform the correct input /output functions and follows sound HCI principles
* a textfile to output reports
* objects to allow for the reusing of code throughout the program and make the program more efficient. The object will contain advanced O.O.P concepts like overloaded methods

|  |  |  |  |
| --- | --- | --- | --- |
| USER | ROLE | ACTIVITY | LIMITATIONS |
| ADMIN | The overall purpose is to help ensure the smooth running off the app and to assist with any problems or errors | * Alter guests’ credit balances * Delete and edit guests’ accounts * Print reports * Login/logout account * View changes made in the database | * Transfer /add/remove large amounts of credits * Change event dates * Take down adverts |
| GUEST | To utilise the app to educate themselves, whether it be about the environment, alternate solutions or information about recycling centres and events | * Create/login to account * Read educational facts and articles about environmental issues * Learn new recipes * Buy coupons * Check credit balance * View event dates * Click links to partnered brands and app’s social media * Logout account | * Not be able to do the actions of admin and event managers |

**Database design**



|  |  |  |  |
| --- | --- | --- | --- |
| **Tblbeach\_clean-up** | | | |
| **Field Name** | **Data Type** | **Size** | **Format** |
| RecordNO | Number | Long Integer | **-** |
| Beach | Short Text | Long Integer | **-** |
| DateOfCleanUp | Date/Time | - | dd/mm/yyyy |
| MemberID | Short Text | 6 | **-** |
| section | Short text | 25 | **-** |

|  |  |  |  |
| --- | --- | --- | --- |
| **tblSponsors** | | | |
| **Field Name** | **Data Type** | **Size** | **Format** |
| ReceiptNO | AutoNumber | Long Integer | - |
| Sponsor | Short Text | 25 | - |
| Amount | Currency | - | Currency |
| DateSponsored | Date/Time | - | dd/mm/yyyy |

|  |  |  |  |
| --- | --- | --- | --- |
| **TblUsers** | | | |
| **Field Name** | **Data Type** | **Size** | **Format** |
| MemberID | Short Text | 6 | **-** |
| Username | Short Text | 25 | **-** |
| password | Short Text | 10 | **-** |
| First name | Short Text | 25 | **-** |
| Surname | Short Text | 25 | **-** |
| Email | Short Text | 30 | **-** |
| Mobile | Short Text | 10 | **-** |
| DoB | Date/Time | N/A | dd/mm/yyyy |
| Credits | Number | Long Integer | General Number |
| Gender | Short Text | 1 | **-** |

**Data dictionary:**

**OBJmember**

**Attributes:**

**Private:**

* fUsername : **String**;
* fPassword : **String**;
* fBalance : **integer**;
* fID:  **integer**;

**methods:**

**public:**

* **constructor** create(sUsername: **string**; iBalance, iID: **integer);**
* **function** getbalance: **integer;**
* **function** getusername: **string;**
* **function** getID: **integer;**
* **procedure** updatebalance(coupon: **integer);**
* **function** SufficientFunds(coupon: **integer): Boolean;**
* **function** ErrorMsg: **string;**
* **function** tostring(sFname, sSname, sgender: **string): string;**
* **procedure** reward;

**OBJvalid**

**Attributes:**

**Private:**

* fPass1: **string**;
* fpass2: **string**;
* fMobile: **string**;
* fEmail: **string;**
* fDOB: **tdate**;

**methods:**

**public:**

* **constructor** create(sPass1, sPass2, sMobile, sEmail: **string**; dob**: tdate);**
* **function** passmatch: **Boolean;**
* **function** ismobile: **Boolean;**
* **function** emailvalid: **Boolean;**
* **function** ErrorMsg: **string;**
* **function** allvalid: **Boolean;**
* **function** oldenough: **Boolean;**
* **function** correctformat(sname: **string**; checkFor: **string = '?><-()\*&^%$#!;=')**

**: Boolean;**

The class **objguest** will be used when admin wants to get details about the guest or update their account details. It will also be used when the user performs actions like buying coupons, it will check if they can buy it and then update their balance if they can.

The class **objValid** will be used when guests are registering and will be performing checks on the data they enter to prevent invalid data from being entered into the system and will return a suitable error message if invalid data is entered

**Array:**

A 2-d array is used to show the admin the amount of trash collected per month and how much was collected based on the temperature.

**Textfile:**

* admin login
* Tracks admin login, logs admin login, time and date
* Generates a report example:

Username: Pearl\_A\_G

First Name: Graham

Last Name: Pearl

Gender: Male

Balance: 200

**Advanced programming concepts:**

I have simulated a loading bar by using a gauge and timer as well as a gif image for my opening screen. I have also limited input into some edit boxes so it only allows numbers to be inputted.

**NAVIGATION FLOW DIAGRAM**

True

Member login

Is password correct?

False

Home page

Events

Shop

Sponsors

Member Register

Is data valid

False

True

True

Admin login

Is password correct?

False

Admin Main UI

View members/Beach Cleanup

Insert new event

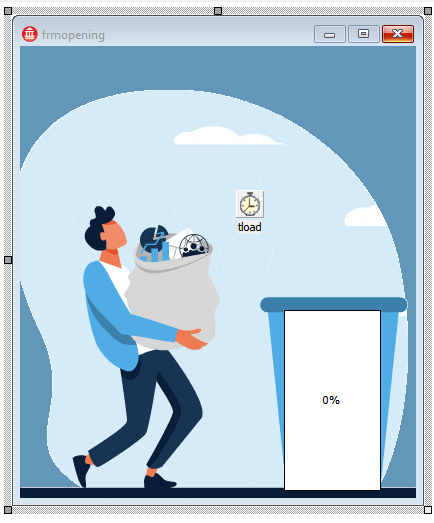
Update member

Delete member

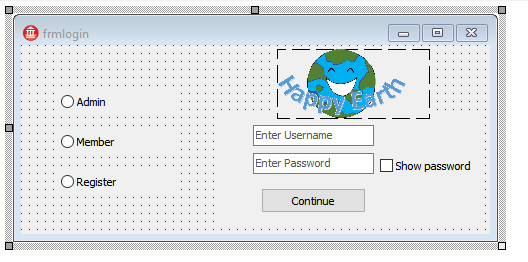
Search

Report

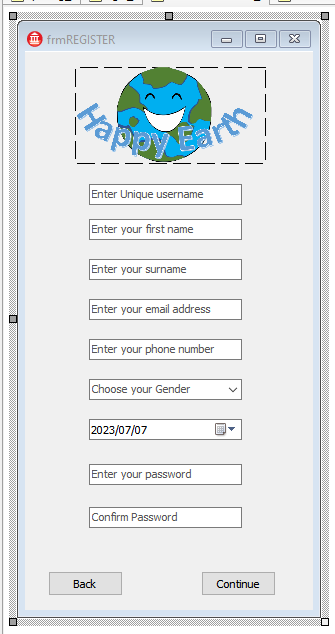
Other



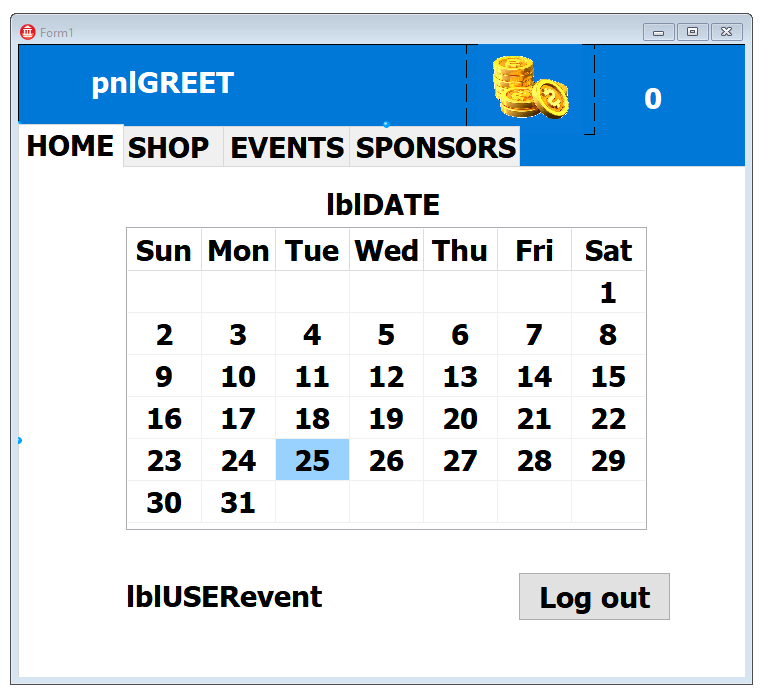
|  |
| --- |
| Gif image with gauge |



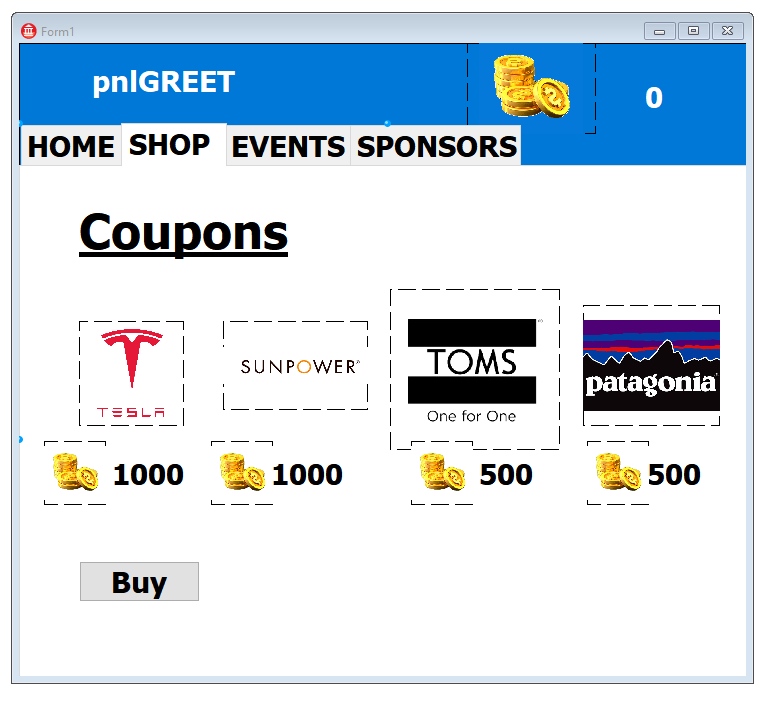
|  |
| --- |
| Radio buttons to choose where you want to login/register,  edit boxes to enter username and password(passwordchar is used here) |
| Timage to display logo and button used to navigate to next form |



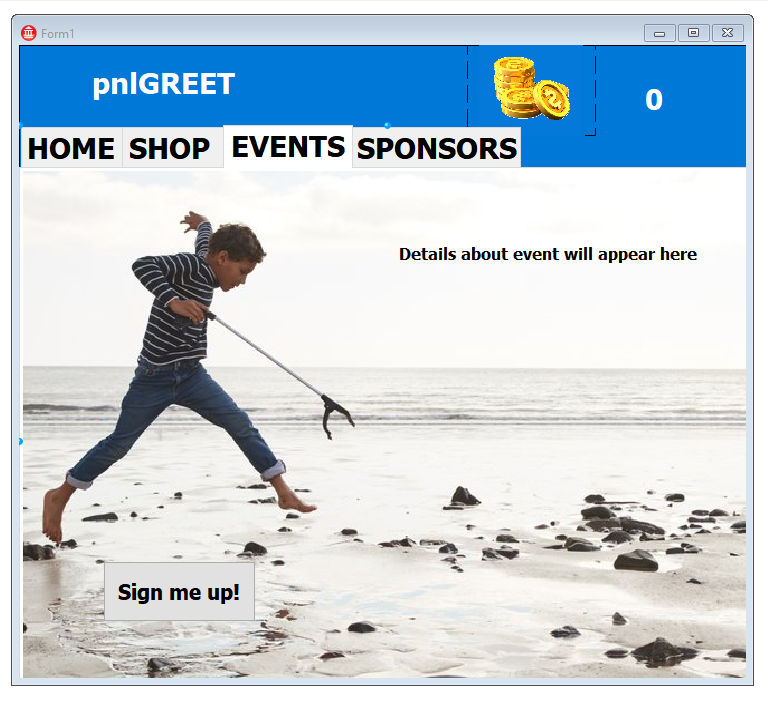
|  |
| --- |
| Timage to display logo  Edit boxes to enter:username,firstname,surname,email,phone,passwords  Combobox to choose gender  Date time picker to enter your date of birth |



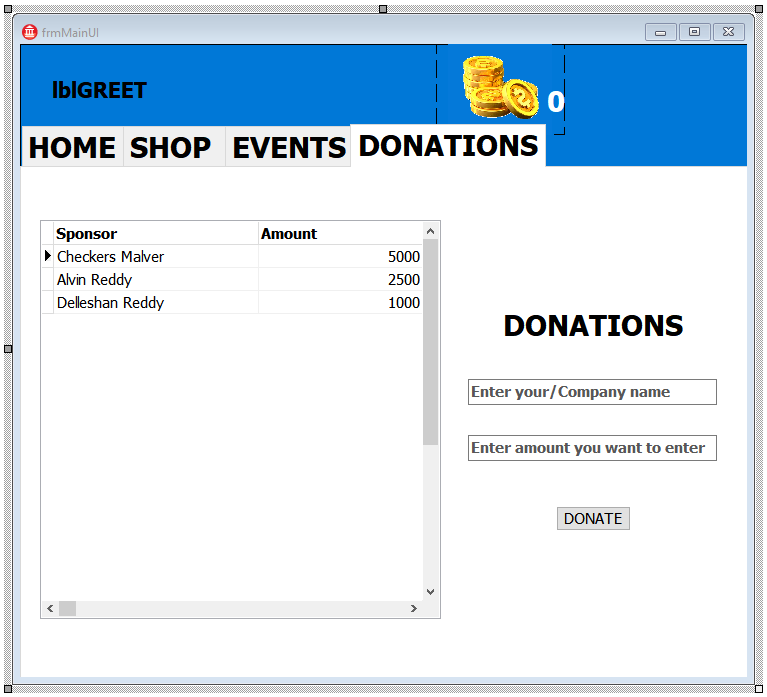
|  |
| --- |
| Panel that greets on every tab  Credit image on every tab to show how many credits  Label to display current month, and what beach cleanup you are signed up for  Tcalendar to show what day it is  Button to log out |



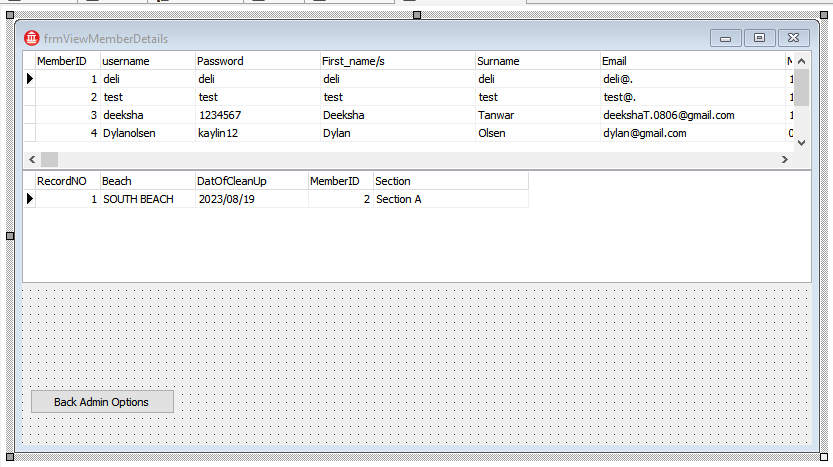
|  |
| --- |
| Timages display sponsor logo’s  Button to validate the amount of credits and update your balance |



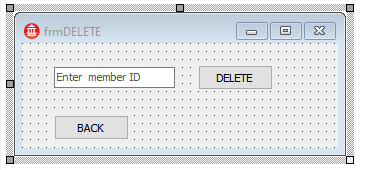
|  |
| --- |
| **Label to display details of the event**  **Button to add credits to balance for attending and to add data to database saying when and which beach you cleaned up at.** |



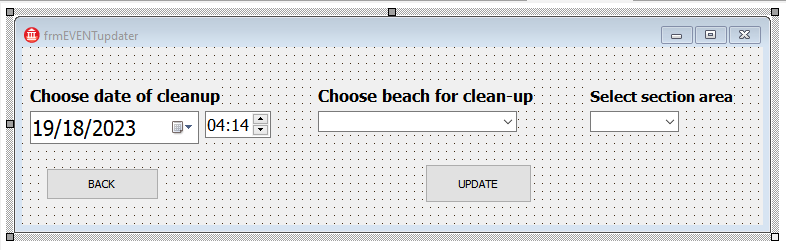
|  |
| --- |
| Dbgrid displaying sponsors |



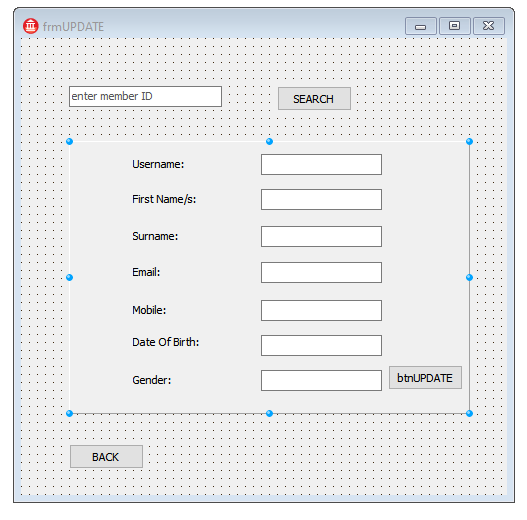
|  |
| --- |
| Dbgrids showing details about members and cleanup duties |



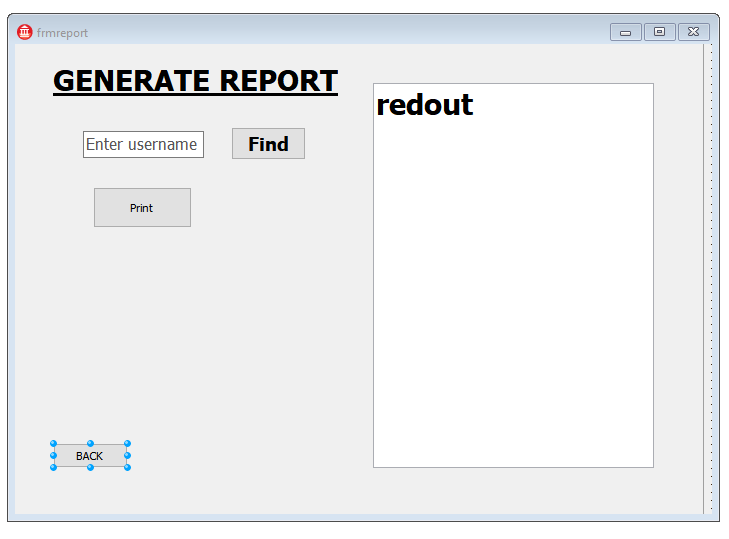
|  |
| --- |
| Edit box to enter member id  Button to search for member id and delete the member as well as navigation to main form |



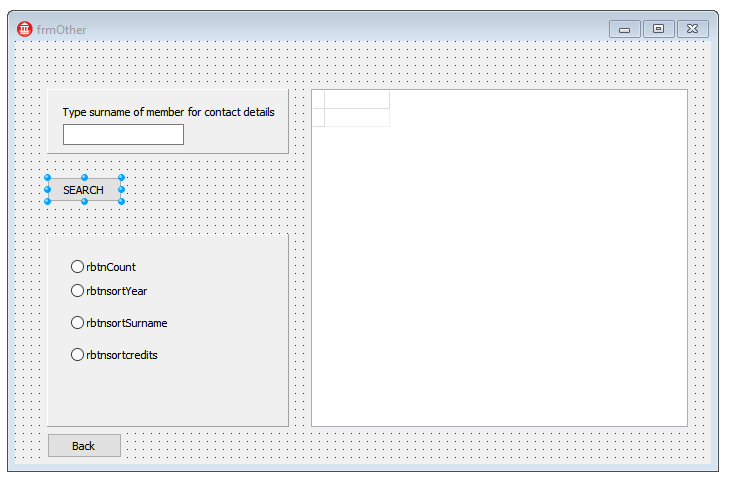
|  |
| --- |
| Labels displaying what type of data that will be displayed in the edit boxes  Edit box to enter member ID  Buttons to search for member ID and navigation to main form |



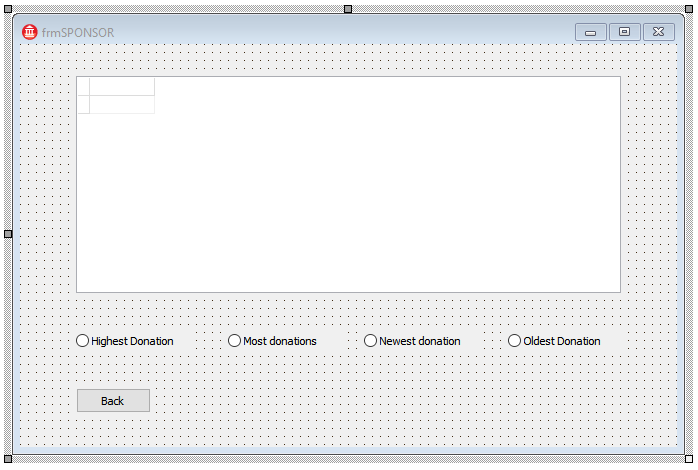
|  |
| --- |
| Labels displaying what type of data that will be displayed in the edit boxes  Edit box to enter member ID and to edit and details of the member  Buttons to search for member ID and navigation to main form as well as update the member’s details |



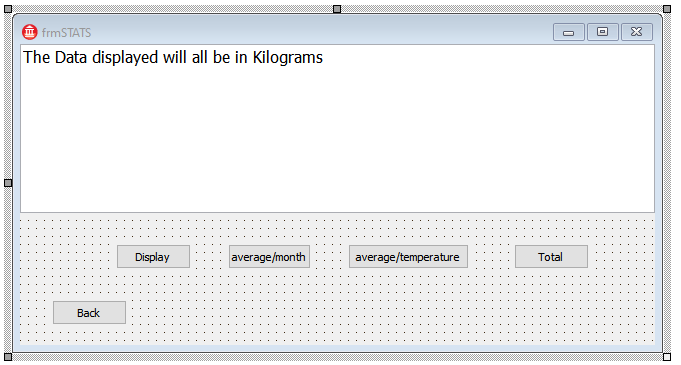
|  |
| --- |
| Edit box to enter username of member  Button to search for member username and for navigation to Main form  Rich edit to preview what will be in textfile report |



|  |
| --- |
| Dbgrid to display details and calculations  Editbox to enter member surname  Radio buttons to choose categories  Buttons to search for member surname and navigation to the main form |



|  |
| --- |
| Radio buttons to choose the options of calculations  Dbgrid to display data  Button to navigate back to admin main |



|  |
| --- |
| Richedit to display 2d array data  Buttons to display data and calculations as well as navigate |

Ipo

**Data input:**

Register form

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Source** | **Data Type** | **requirements** |
| TDateTimePicker | Mouse | DateTime | DD-MM-YYYY |
| TEdit | Keyboard | String | Email: Requires ‘@’and ‘.’ |
| TComboBox | Mouse | String | Gender(male/female) |
| Teditbox | keyboard | string | South African phone number : length 10 |
| Teditbox | keyboard | string | Username :N/A |
| Teditbox | keyboard | String | Password: Matches the confirmed password |
| Teditbox | keyboard | string | Firstname: N/A |
| Teditbox | keyboard | string | Lastname: N/A |

Admin login :

|  |  |  |  |
| --- | --- | --- | --- |
| **component** | **source** | **Data Type** | **Format** |
| Editbox | keyboard | string | Password must match data in admin password textfile |

**Input validation**

Register form

|  |  |  |
| --- | --- | --- |
| **Component** | **Data validation** | **Error message** |
| TDateTimePicker(DOB) | Check if there is a date picked and the years between the date and current date is greater than 10 | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| TEditbox(Email) | Check if there is a ‘@.’ Present | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| TComboBox(gender) | Check if a value was selected | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| Teditbox(mobile) | Check if length equals 10 | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| Teditbox(username) | Check if a value was entered | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| Teditbox (password) | Check if passwords match and that the length is greater than equal to 10 | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| Teditbox (firstname) | Check if a value was entered | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |
| Teditbox (lastname) | Check if a value was entered | data missing, please fill in all text boxes  OR  Data does not contain only numbers and letters, please correct this |

Admin login :

|  |  |  |
| --- | --- | --- |
| **component** | **Data validation** | **Error message** |
| Editbox(code) | Checks if the code matches one of the codes in the textfile | Invalid code |

**Data processing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Components** | **Input** | **Process** | **Output** |
| btnRegister | Button is clicked and all data in components is captured | Data is then validated | If data passes validation, it gets stored in database and the guest is then shown the main GUI otherwise a suitable error message will be shown |
| dtpDateOfBirth | Date is selected | Selected date is checked to see if its not the current date | Date will be stored in database otherwise a suitable error message is shown |
| edtFirstName | Data inputted and captured when guest clicks register | Checks if the data is not null | Data gets stored in database otherwise a suitable error message is shown |
| edtLastName | Data inputted and captured when guest clicks register | Checks if the data is not null | Data gets stored in database otherwise a suitable error message is shown |
| Edtusername(login) | Data inputted and captured when login is clicked | Checks if the data matches ones of the usernames in the database | User shown main GUI  , Otherwise, suitable error message shown |
| Edtpassword(login) | Data inputted and captured when login is clicked | Checks if the data matches ones of the passwords in the database related with the username | User shown main GUI  , Otherwise, a suitable error message is shown |
| Edtusername(report) | Data inputted and captured when FIND is clicked | Checks if the data matches ones of the usernames in the database | User shown the inputted username’s detail in the richedit adjacent |
| Edtusername(register) | Data inputted and captured when register is clicked | Checks if the data is not null | Adds data to database and shows user main GUI otherwise shown suitable error message |

**How will data be processed**

1. When user has attended an event:  
   A preset value Is added to the current balance and the user’s balance will be updated and the updated balance will be displayed

Fbalance:= fbalance + 100;

Lblbalance.caption:= objguest.getbalance;

1. Buy coupon:

The selected coupon when buy is clicked has a value associated with its name that will be brought from the database and will be subtracted from the user’s balance and the updated balance will be displayed

Fbalance:= fbalance – rCost;

Lblbalance.caption:= objguest.getbalance;

1. Edtlastname and edtfirstname:

The program checks if the names entered has a length of greater than 1

Var valid:boolean;

If length(edtfirstname.text)<1 and length( edtlastname.text )<1 then

Valid:= false;

Showmessage(‘please enter first/lastname’)

Else

Valid:= true;

1. Password:

The program checks if the data entered has a length greater than equal to 10 as well as checks if it contains 1 integer

If length(edtpassword.text)< 10

Then

Valid:= false;

Showmessage(‘password weak’)

Else

Valid:= true

End;

1. Date of birth:

Date entered will be checked to see if their age is greater than 15 years of age

Var bValid :Boolean;

Begin

If yearsbetween(now, DoB) > 15 then

Begin

bValid:true;

end

else

begin

bvalid:= false;

showmessage(‘too young’);

end;

1. Gender entered:

Program checks if a gender has been selected from the combo box

Var bvalid:Boolean;

Begin

If cmbBox.value = null;

Then begin

Bvalid:= false;

Objvalid.errormsg;

End

Else

Bvalid:= true;

End;

1. email

var bvalid:Boolean;

begin

if pos(‘@’,edtemail.text) <>1

then begin

bvalid:=false;

objvalid.errormsg;

end

else

bvalid:= true;

end;

1. username

var bvalid:Boolean;

begin

if length(edtusername) <5

then begin

bvalid:= false;

showmessage(‘username to short’);

end

else

bvalid:= true;

end;

**Data output:**

Report:

Once the username matches with one in the database, the data will be displayed using oop method tostring to format and display the data in the adjacent richedit

Main GUI:

Once data is captured and added to the database after registering or logging in, the user’s first name will be displayed in a label along with a friendly greeting

Coupons:

Once user purchases a coupon, their updated balance will be displayed in a label

Event registered:

Once a the user registers for an event, the date of the event the signed up for will be displayed in a label