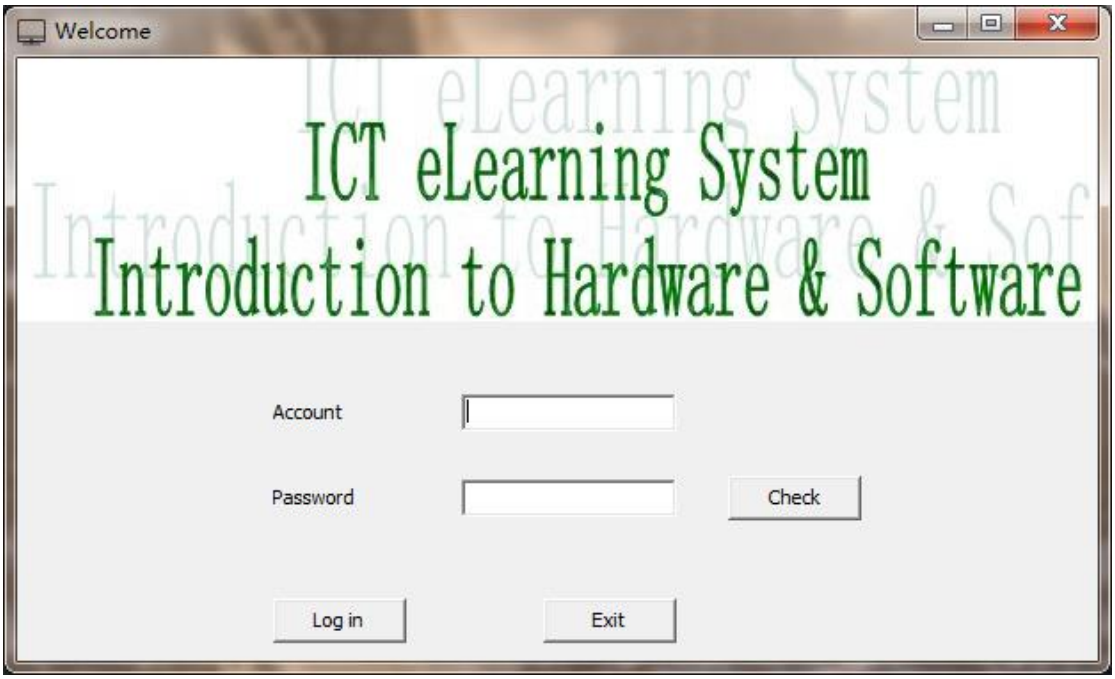
HKDSE 2015

ICT SBA Report



School:

SKH Lui Ming Choi Secondary School

Name:

Lai Ho Wo

Class (Class Number):

6H (13)

Admission Number:

A6916

Instructor:

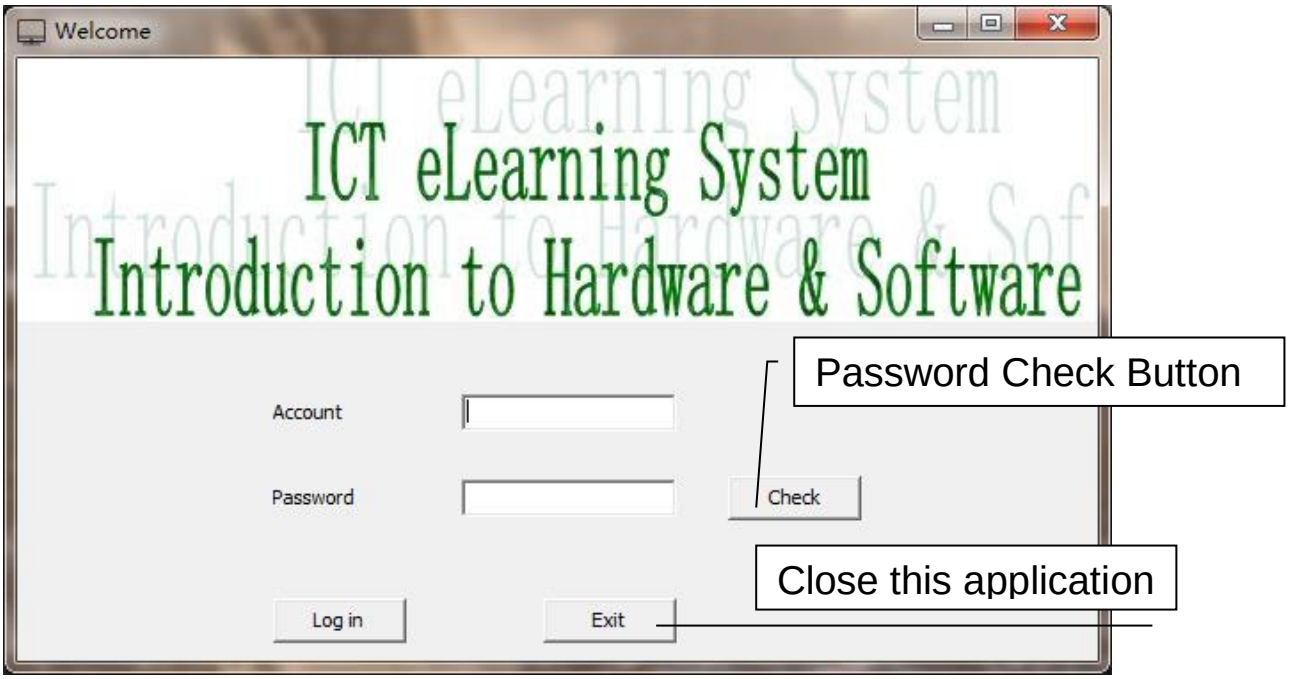
Mr Wong Shing Bun

Content

Design

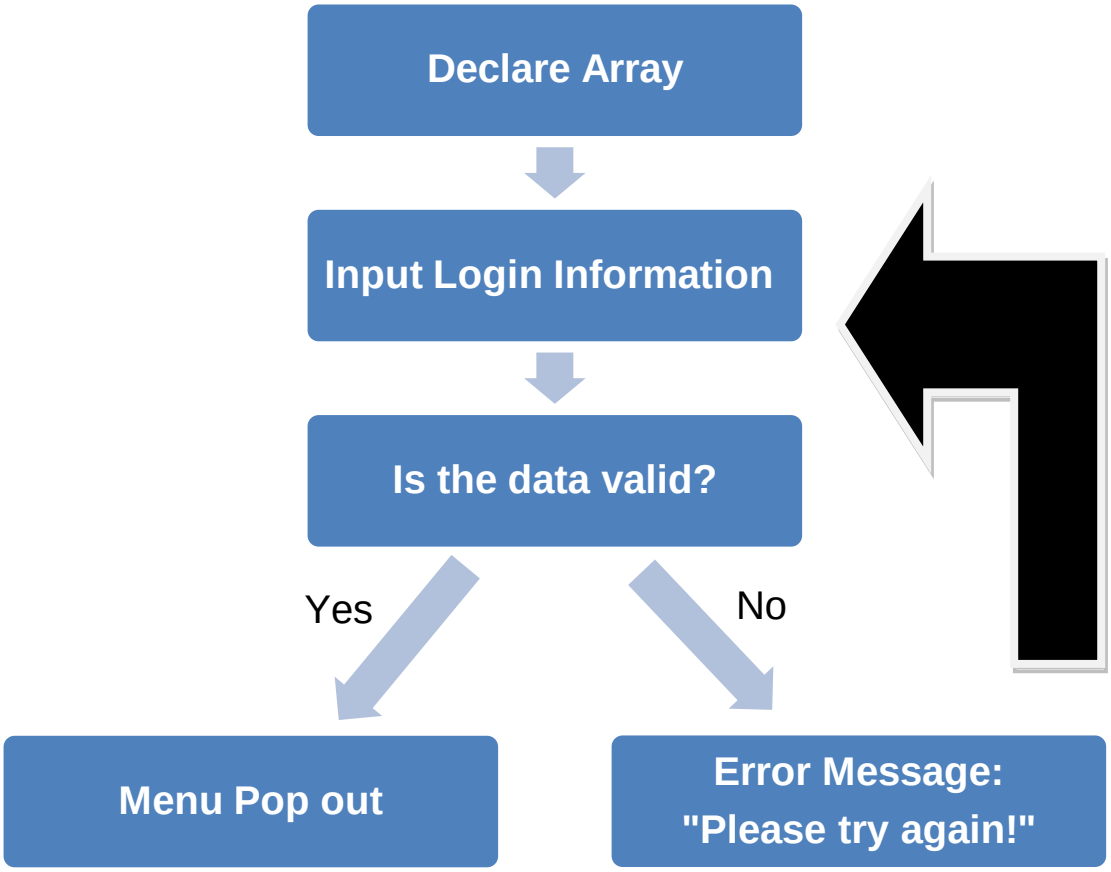
Description of the design

Login System



Since the whole system will be divided as 2 versions, Student Version and Administrator Version, a login system will be set to verify their identities before entering the system.

The Password Check Button next to the Password Textbox enables the password that users have typed to be temporarily displayed on the screen so that users can check for typing errors.



First, I will declare 2 arrays for storing the usernames and the passwords.

Next, users will be allowed to enter their usernames and passwords in the login page.

Then, when they press Enter or “Log in” button, system will check if the data input is valid. If the data is valid, the user menu will be shown while inputted data will be cleared for safety reason.

Data Storage

I will use both database and array for storing data.

For the Login Information Form(accessed by administrator only), I will connect the system to a database which contains all user usernames and passwords accepted by the system.

Database Structure

Field Name	Data Type	Field Length
Account	Text	255
Password	Text	255

"Login_Info" Table

Besides the database, I will also use array for storing the login information of the system by using procedures. It is safer than using external database because it can prevent users from modifying the usernames and passwords directly by using any Database managing software like Microsoft Access.

Moreover, because of the array, the system can store a large amount of data without using too many variables.

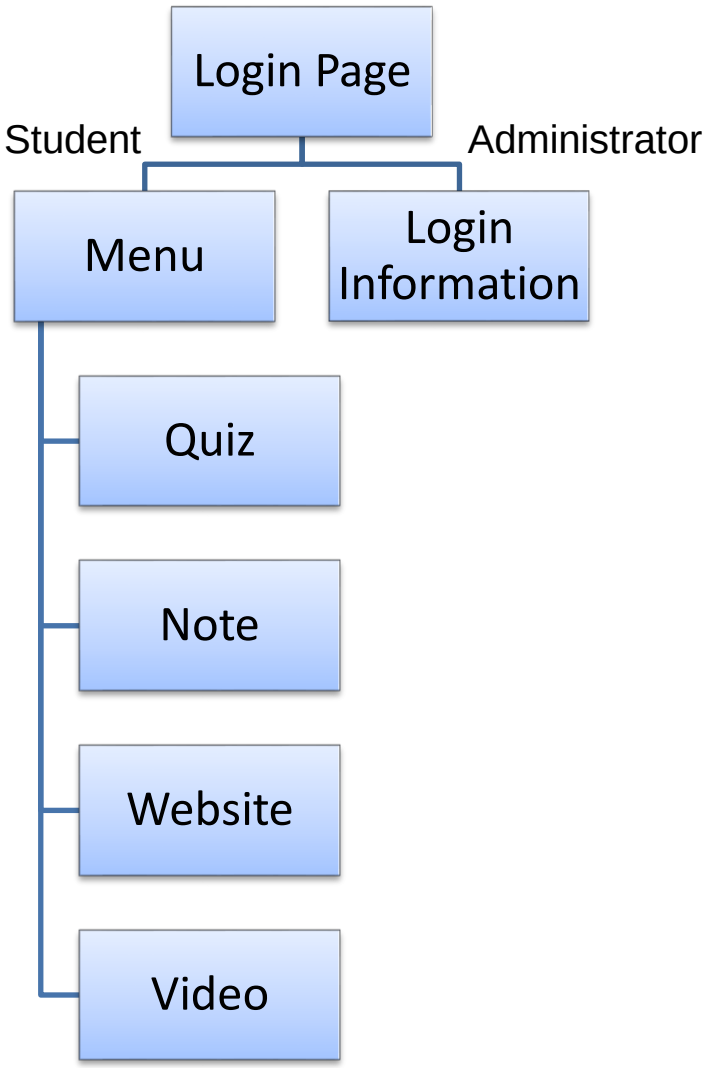
Array Structure

```
ac[1]:= 'admin';
ac[2]:= 'std';
ac[3]:= 'benny';
ac[4]:= 'cyrus';
ac[5]:= 'lusan';

pw[1]:= 'admin';
pw[2]:= 'std';
pw[3]:= 'benny';
pw[4]:= 'cyrus';
pw[5]:= 'lusan';
```

1D Array for storing the login information

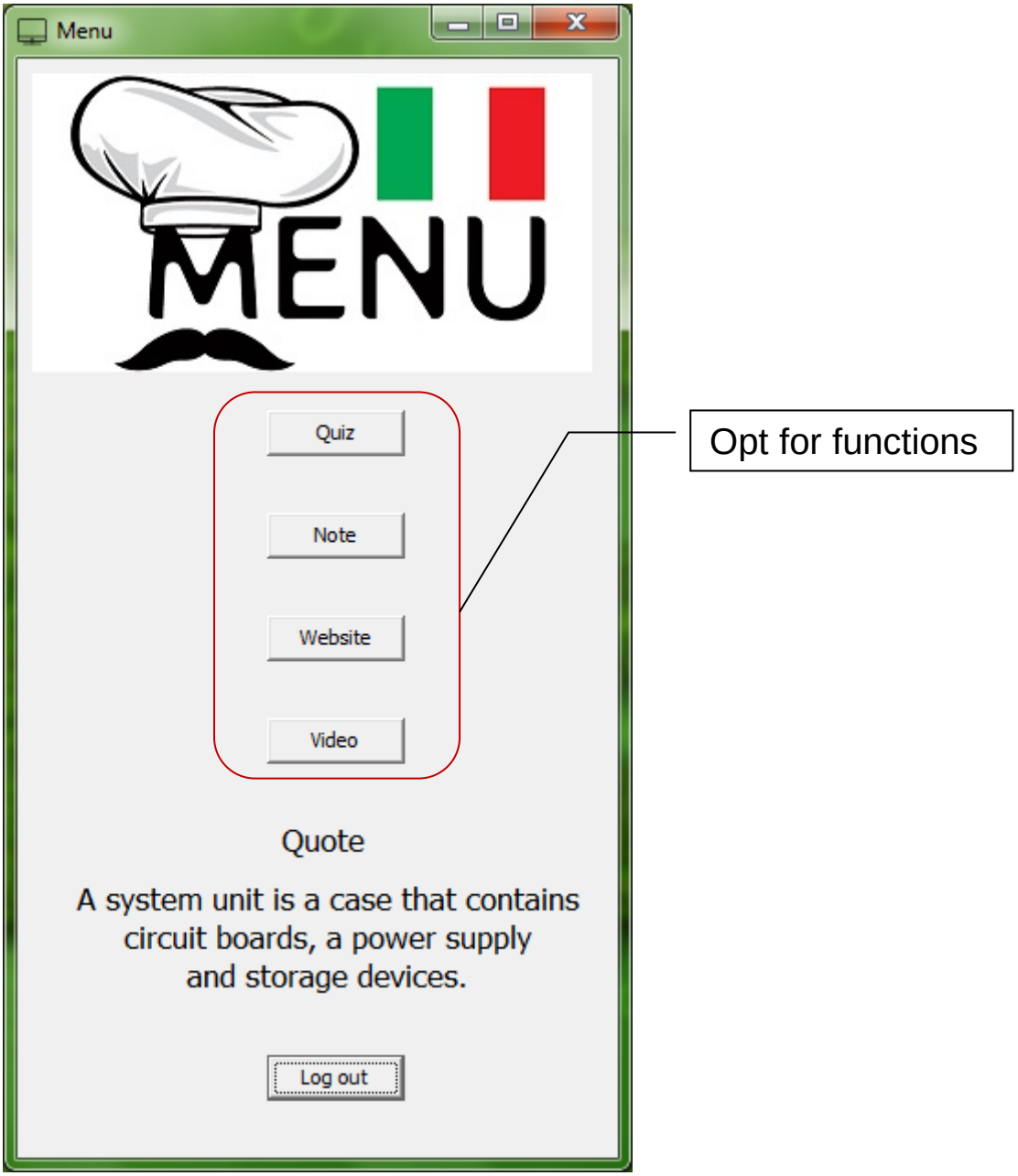
Basic structure of the program



First, when users start to run the program, the Login Page will be shown.

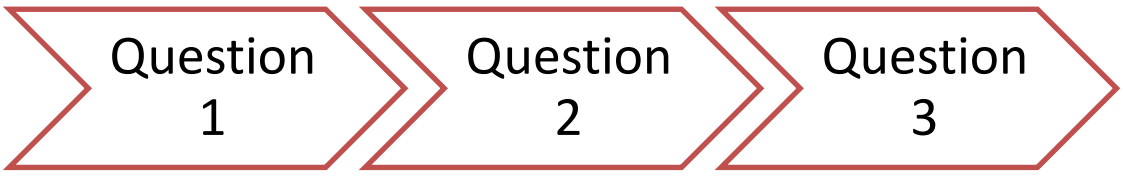
Second, users are required to login. If they login as student, a menu will be popped out and they can choose different functions(“Quiz”, “Note”, “Website”, “Video”) from it; if they login as administrator, a login information form will be shown. The details of the functions will be explained later.

System's Functions



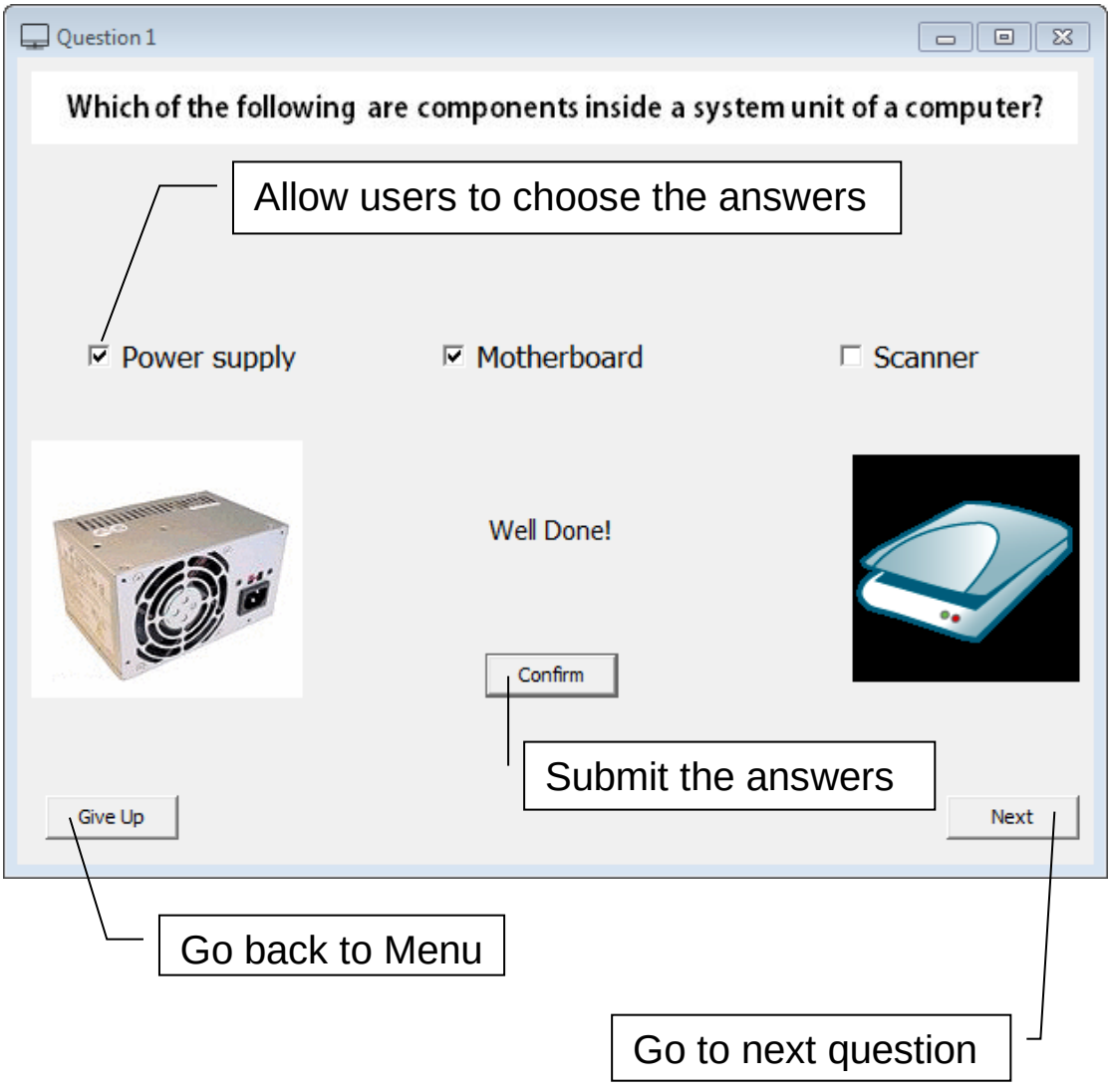
Student users can choose their wanted functions by simply clicking the buttons on the menu, after they login the system.

1 Quiz

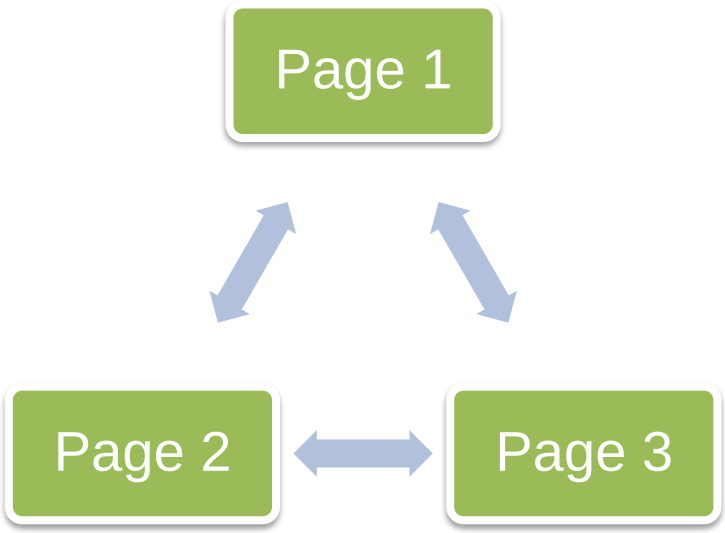


The above diagram shows the flow chart of Quiz. Users must finish the first question before they can answer the second question. Skipping of questions is not allowed.

The following is the design of layout:

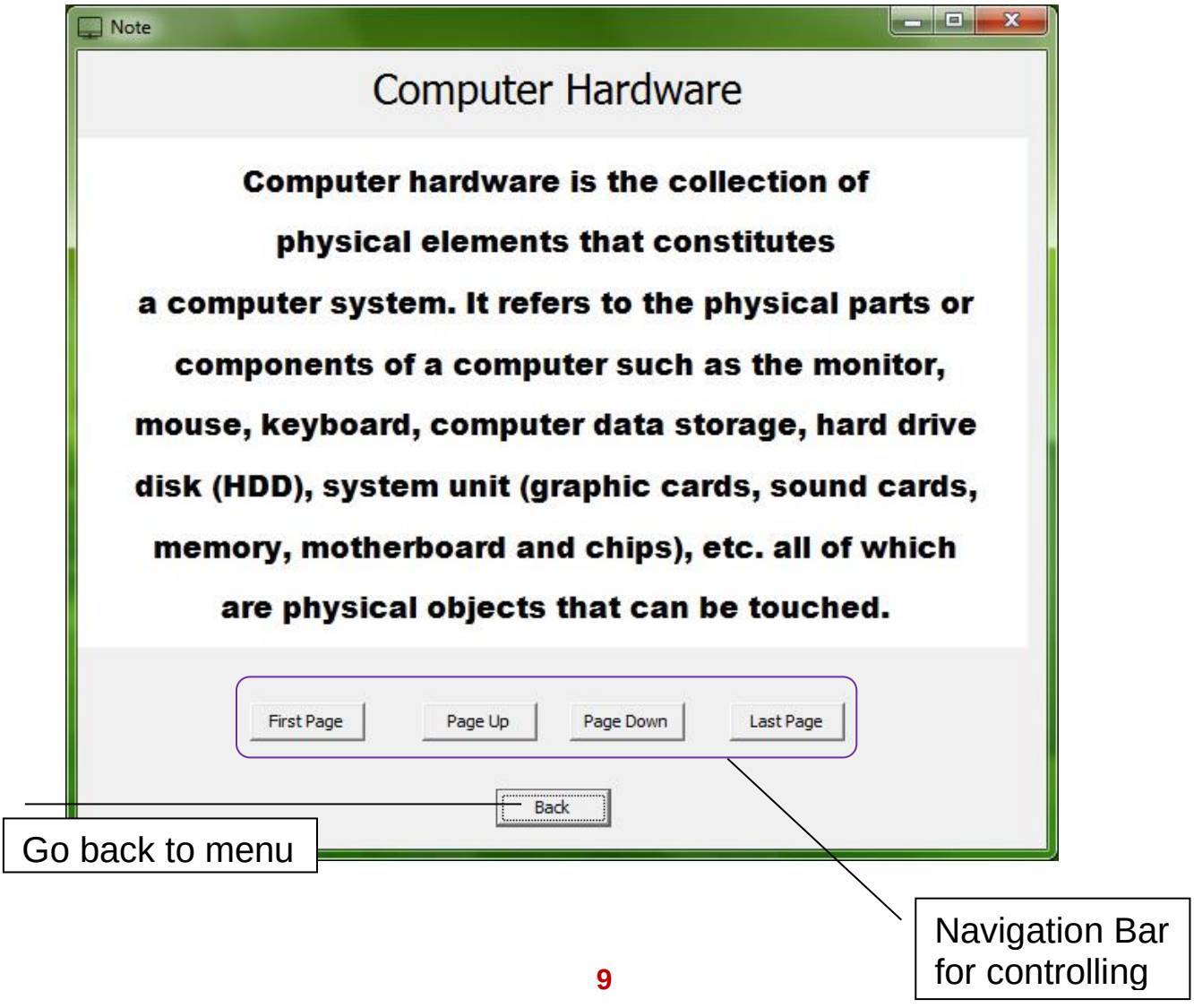


2 Note

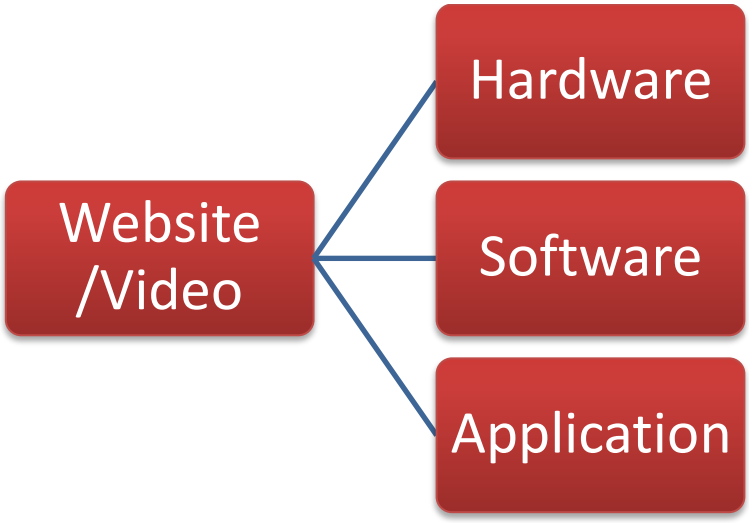


This is the flow chart of Note. Users are free to jump to the page they want.

Here is the design of layout:

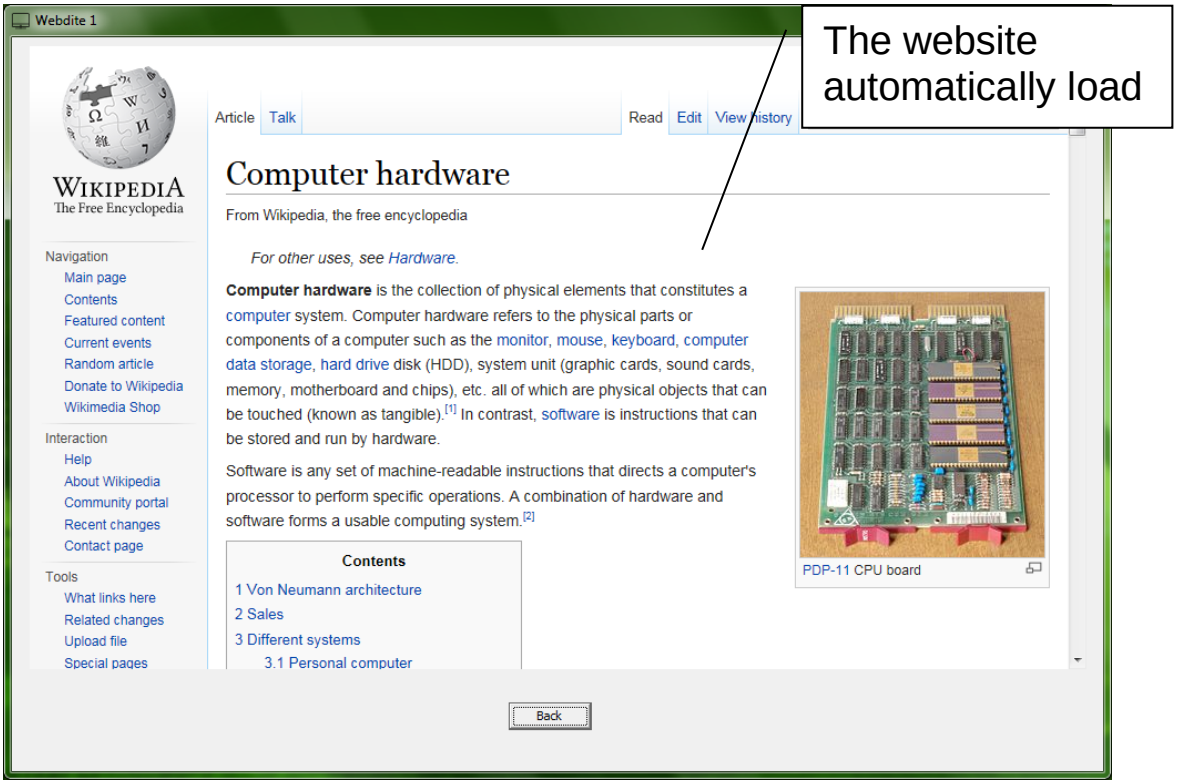


3 Website and Video



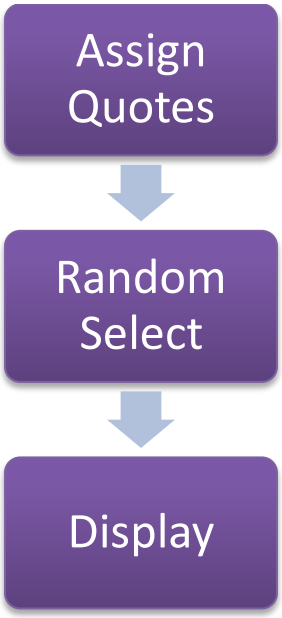
This is the structure of Website and Video. Users are required to connect their computers to the Internet before they can use this function properly.

The layout design is not complicated, as below:




Video 1
Computer hardware basics tutorials for beginner...
Computer
Hardware
Basics
0:00 / 9:05
Back
Play the video

4 Quote



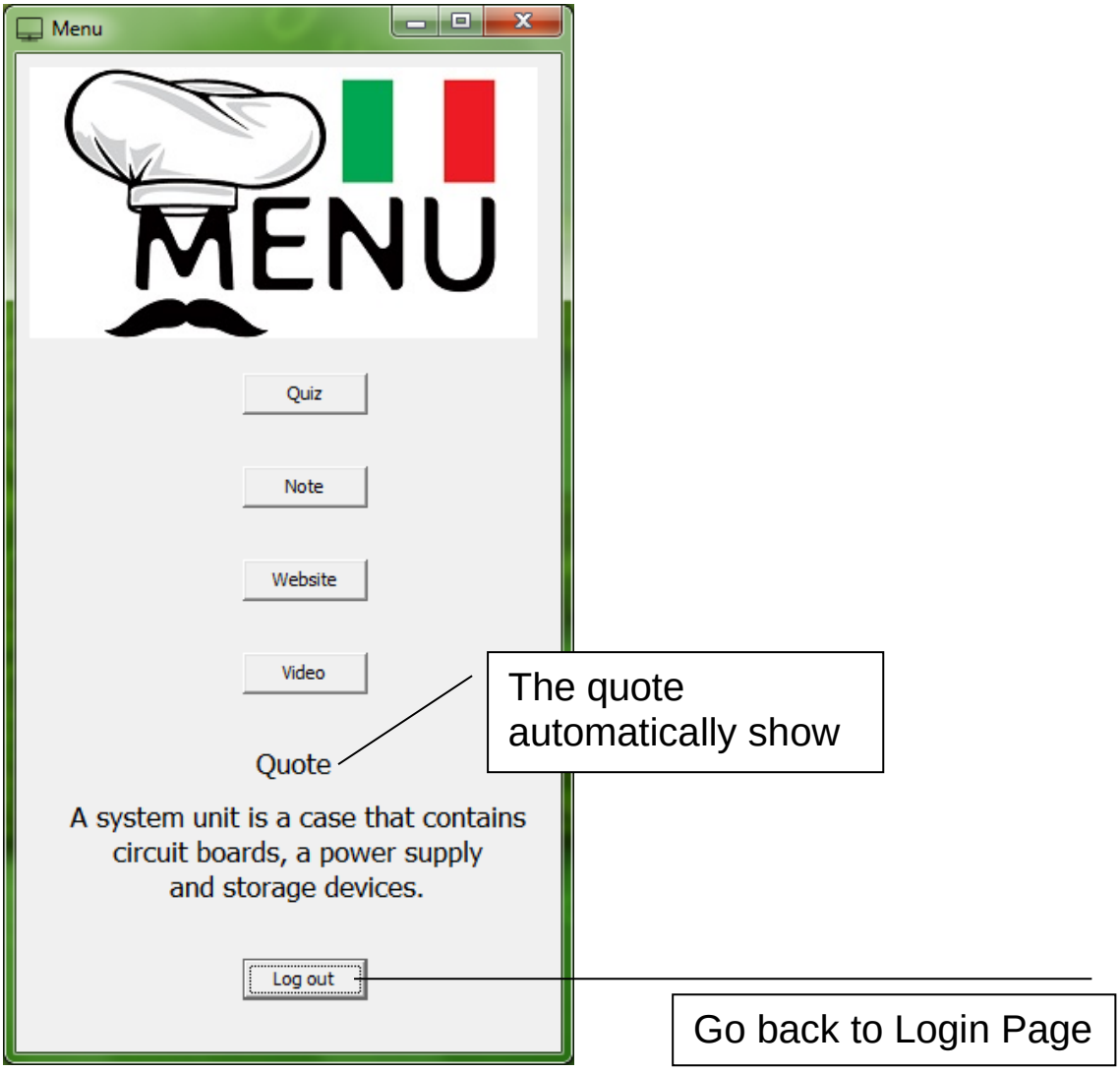
This is the procedure of Quote.

First, the program assigns a number to each quote.

Next, it selects a number randomly.

Finally, it displays the quote with the number selected.

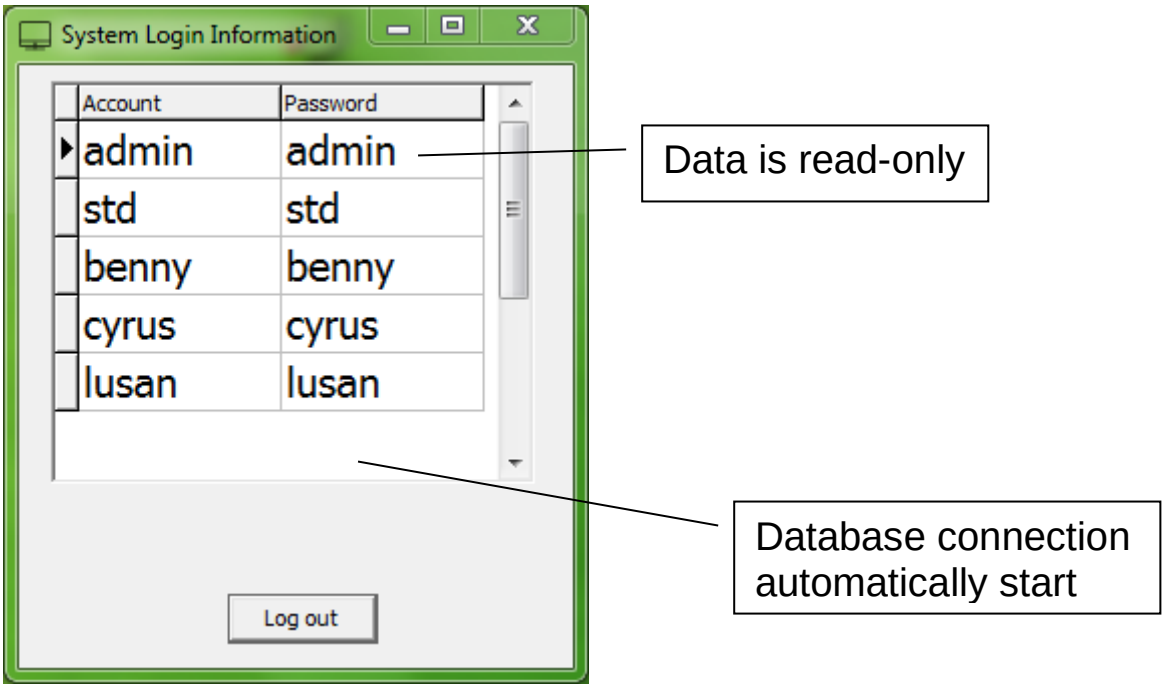
The design of interface is shown below:



5 Login Information

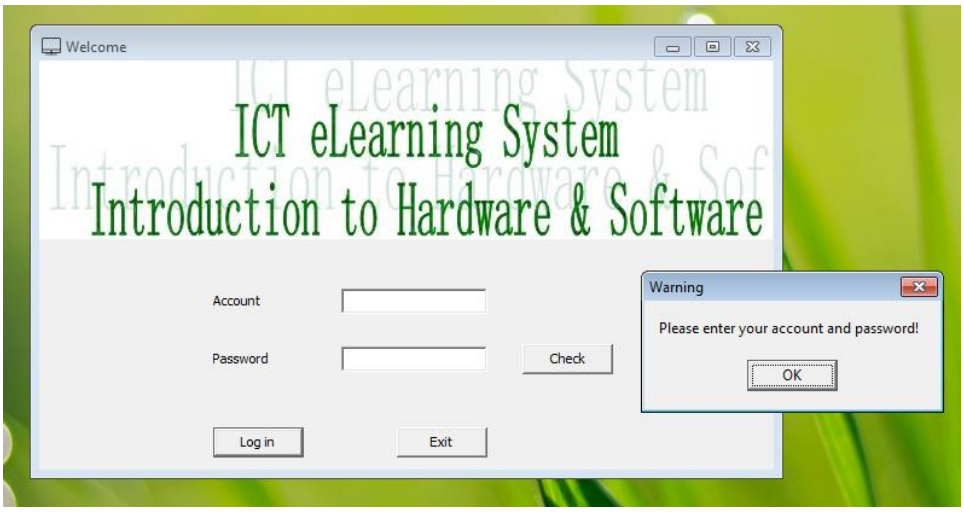
This function allows the administrator to check for the accounts and passwords in case the students forget them.

The interface will be designed as:



Data Validation

In order to ensure that the necessary fields are present, field presence check is used for login process. If the account or password textbox is blank, an alert message will pop out.



Rules users should follow

What are the rules?

In order to ensure the program can be run accurately, users should follow the rules listed below.

- ✓ **Users should keep their passwords secret**

Since passwords cannot be changed by using the build-in function in the system, users should remember the passwords and keep it secret. Furthermore, they need to make sure nobody is looking over their shoulder when they press the Password Check Button.

- ✓ **Users should double check whether they have connected audio output devices to their computers**

As the program will play sound, users ought to connect a speaker or a pair of earphones to their computers so that they can hear the sound and avoid any error messages during running of the program.

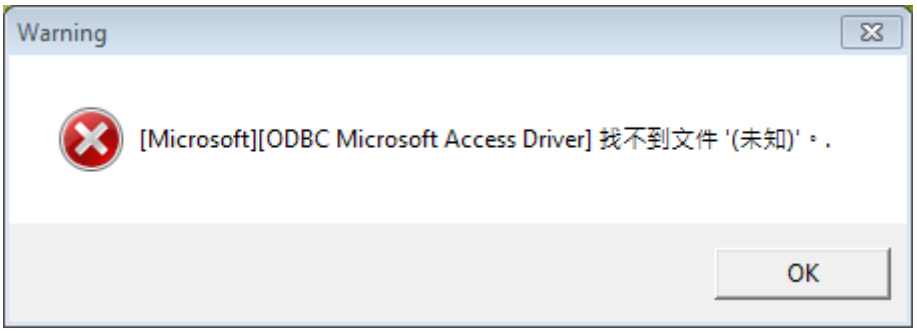
- ✓ **Users should put all the files related to the system in the same directory**

In order to reduce the size of the system, I choose to connect the system with the files by linking instead of embedding (except pictures). So, the path of the files is preset to the directories of the system. Users should not try to move the files out of the directories.

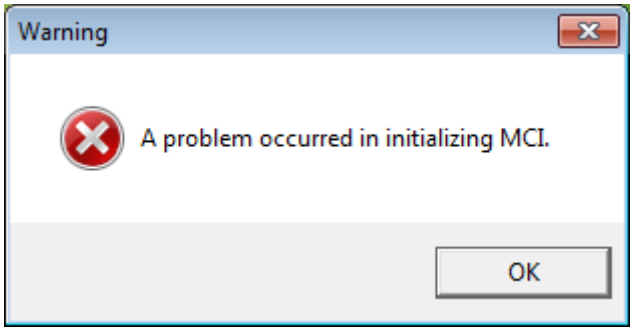
Methods used to implement the rules?

Pop-out Messages

If the system cannot connect with the external database, a pop-out message will be shown:



If the system cannot find any audio output devices, another pop-out message will be shown:



When users saw those error messages, they may not use the functions effectively and efficiently. For database error, they should put the related files into program's directory before using the program. For sound playing error, they ought to connect audio output devices to their computers before using the program.

Implementation

Hardware & Software required

To use the ICT eLearning System, users are required to have the hardware and software below.

Minimum system requirement:

Hardware & Software

Conponent	Item/Description	Quantity
Hardware Platform	● Dual-Core Intel® Pentium® CPU ● 100 MB RAM ● 128 MB external storage device	1
Software	Operating System: ● 16-bit Windows Vista Application Program: ● Microsoft Access 2003 ● Internet Explorer with Flash Player add-ons	1
Input Devices	● Mouse ● Keyboard	1
Output Devices	● LCD/LED Monitor ● Speaker/Headphone	1

Settings

Setting	Basic Requirement
Colour	16-bit colour
Resolution	800 x 600 pixels
Internet Access	Necessary

Recommended system requirement:

Hardware & Software

Conponent	Item/Description	Quantity
Hardware Platform	● Dual-Core Intel® Pentium® CPU ● 1 GB RAM ● 100 GB external storage device	1
Software	Operating System: ● 32-bit Windows 7 Application Program: ● Microsoft Access 2007 ● Internet Explorer with Flash Player add-ons	1
Input Devices	● Mouse ● Keyboard	1
Output Devices	● LCD/LED Monitor ● Speaker/Headphone	1

Settings

Setting	Basic Requirement
Colour	32-bit colour
Resolution	1024 x 768 pixels
Internet Access	Necessary

Skills/Knowledge required

In order to use the system, users should be required to have the following skills and knowledge.

- **Basic Computer Skills**

Users should be able to use a computer so that they know the ways to run the ICT eLearing System. For example, they need to use a mouse and keyboard to control the flow of program. If the users do not know how to use computer, they cannot use the system correctly. As a result, the program may not be run or even crashed during initialization.

- **Basic Computer Knowledge**

The system introduces computer hardware and software. It is better for students to have some background information of computer before enjoying this application.In this way, they will understand the learning materials more easily.

- **Simple English**

Since the program is written in English, users should be able to read simple English. Thus, they can follow the instructions and reduce the errors. Moreover, they are able to read the content of the quiz and notes if they can read simple English.

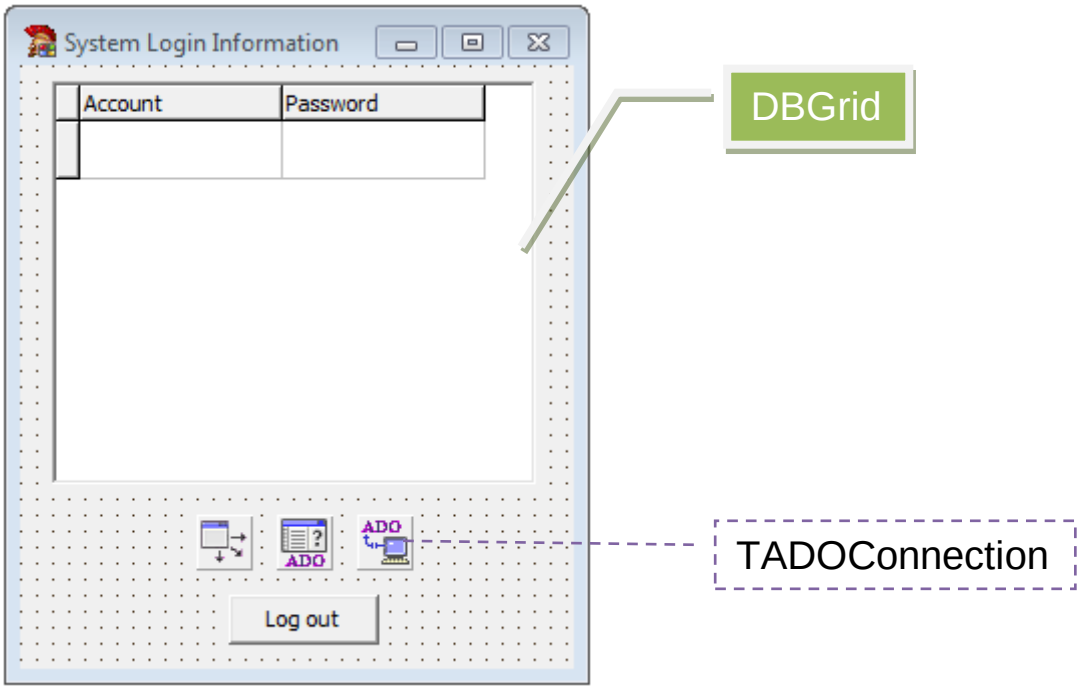
Use of Library

```
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB, ADODB;
```

Library can be used to produce a Graphical User Interface layout of the program. In the program, I commonly used “Windows”, “Messages”, “SysUtils”, “Variants”, “Classes”, “Graphics”, “Controls”, “Forms” and “Dialogs”, in order to produce an user-friendly interface.

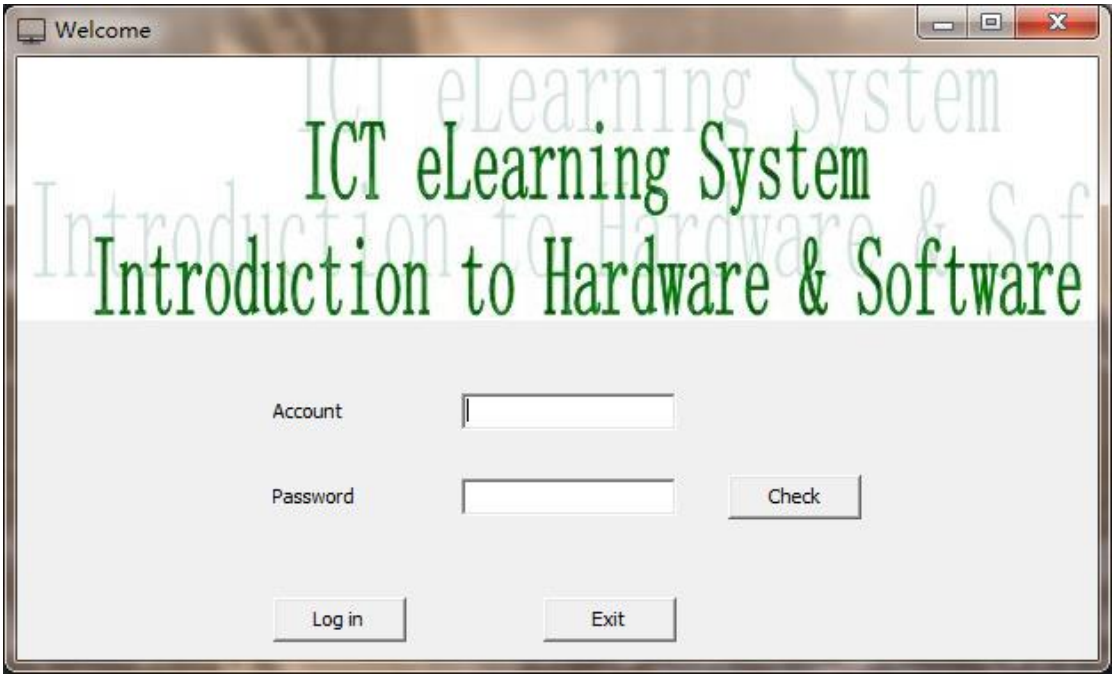
The libraries “DB”, “Grids”, “DBGrids” and “ADODB” are used for the database. “DB”, “Grids” and “DBGrids” are supposed to construct a user-friendly interface to show the tables in database. In the library “ADODB”, it allows me to use “TADOConnection” to connect the database by using a connection string.



Demonstration of the program

Starting the program

Once the system starts to run, a login page shows.



When the login page showed, procedure FormShow will run for initialization.

```
procedure TForm0.FormShow(Sender: TObject);
  begin
    edit1.Text:= '';
    edit2.Text:= '';
    edit1.SetFocus;

    ac[1]:= 'admin';
    ac[2]:= 'std';
    ac[3]:= 'benny';
    ac[4]:= 'cyrus';
    ac[5]:= 'lusan';

    pw[1]:= 'admin';
    pw[2]:= 'std';
    pw[3]:= 'benny';
    pw[4]:= 'cyrus';
    pw[5]:= 'lusan';

    acpass:= false;
    pwpass:= false;
end;
```

Initialization

Assigning user names and passwords*

Initialization

*ac and pw are the arrays for storing the user names and passwords respectively.

```
var
  Form0: TForm0;
  ac: array[1..n] of string;
  pw: array [1..n] of string;
  acpass, pwpass: boolean;
```

If the “Log in” button is pressed, procedure Button1Click will be run and validate whether the account and password are match.

```
procedure TForm0.Button1Click(Sender: TObject);
var
  i: integer;
begin
      if (edit1.Text = '') or (edit2.Text = '') then
        showmessage('Please enter your account and password!')
      else
      begin
      if (edit1.Text = ac[1]) and (edit2.Text = pw[1]) then
      begin
        aform1.Show;
        form0.Hide;
        acpass:= true;
        pwpass:= true;
      end;

      for i:= 2 to n do
      begin
        if (edit1.Text = ac[i]) and (edit2.Text= pw[i])  then
        begin
          sform1.Show;
          form0.Hide;
          acpass:= true;
          pwpass:= true;
        end;
      end;

      if not(acpass) or not(pwpass) then
        showmessage('Please try again!');
      end;
end;
```

For the consideration of the convenience of users, a procedure Edit2KeyPress is added so that users can press Enter key to login after they enter their accounts and passwords.

```
procedure TForm0.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if Ord(Key) = VK_RETURN then
      button1.Click;
end;
```

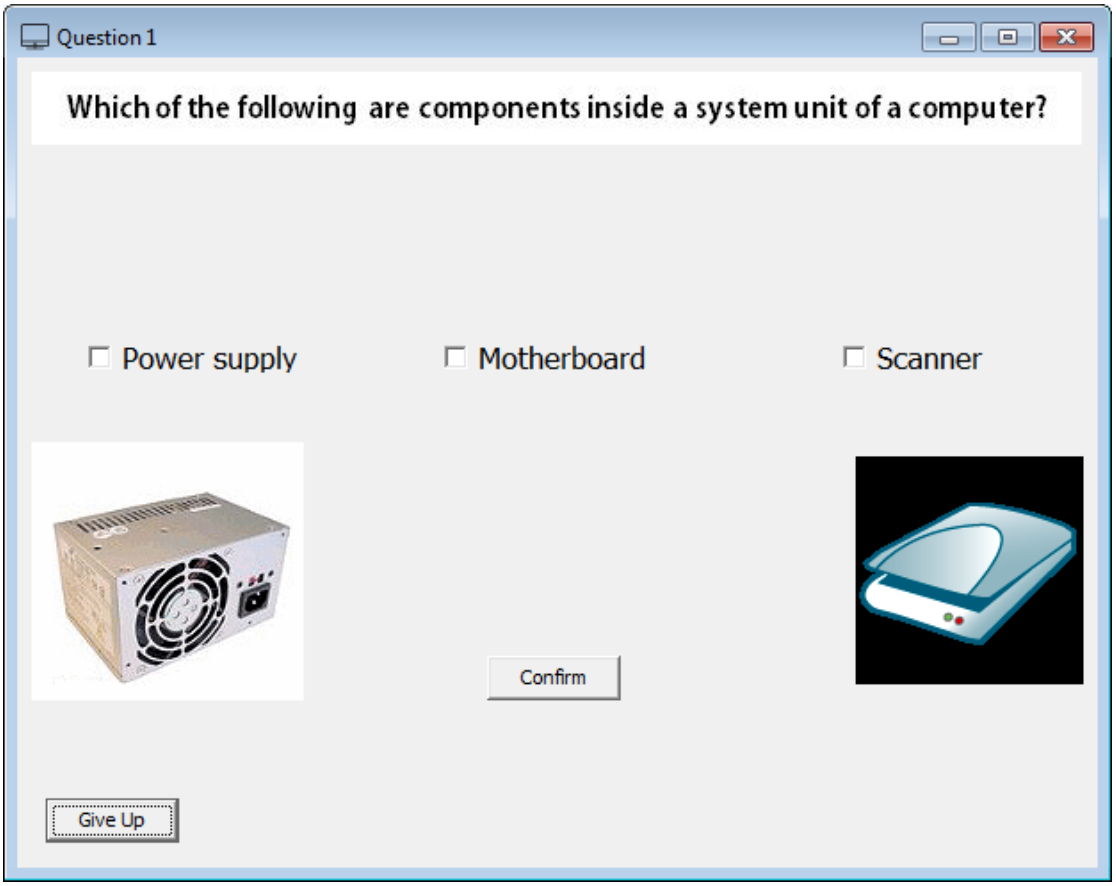
To let users check for typing mistake of their passwords, procedure Button3MouseUp and Button3MouseDown are used. Pressing the “Check” button while holding it will let users see their passwords without any masks; releasing it will make the mask of passwords appear again.

```
procedure TForm0.Button3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  edit2.PasswordChar:= #0;
end;

procedure TForm0.Button3MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  edit2.PasswordChar:= '*';
end;
```

Quiz

If students login and press the “Quiz” button, the following page will be shown.

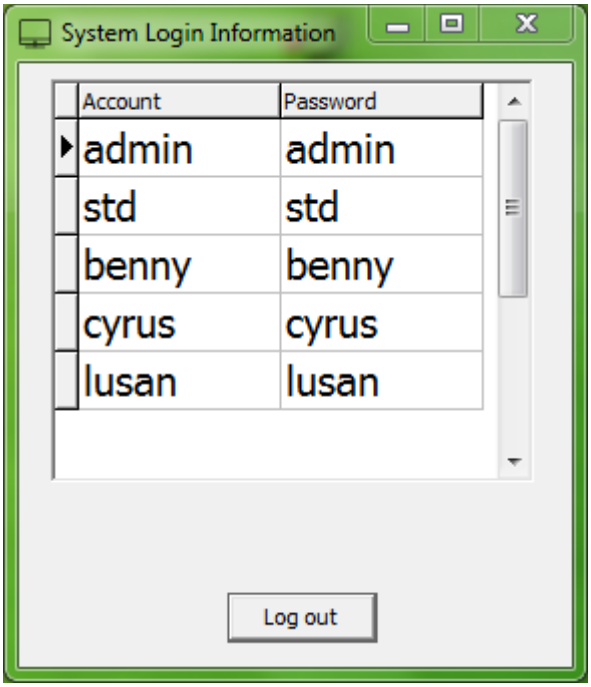


If the “Confirm” button is pressed, procedure Button2Click will be run and validate whether the inputted answers and the built-in answers are match.

```
procedure TsForm2.Button2Click(Sender: TObject);
begin
  label1.Caption:= 'You have got the wrong answer!';
  mediaplayer1.FileName := 'wrong.mp3';
  mediaplayer1.Open;
  mediaplayer1.Play;
  if (checkbox1.Checked and checkbox2.Checked) and (checkbox3.Checked = false)
  then
  begin
    label1.Caption:= 'Well Done!';
    button3.Show;
    mediaplayer1.FileName := 'correct.mp3';
    mediaplayer1.Open;
    mediaplayer1.Play;
  end;
end;
```

Login Information Page

If administrator login, the following page will be shown.



The procedure for displaying the above table and its content:

```
procedure TaForm1.FormCreate(Sender: TObject);
begin
  path:= extractfilepath(application.exename);
  aForm1.ADOConnection1.ConnectionString:='Provider=MSDASQL.1;Persist Security
  aForm1.ADOConnection1.Open;
  aForm1.ADOQuery1.Active:=true;
end;
```

Load the path for database files

Update the connection

Start the connection

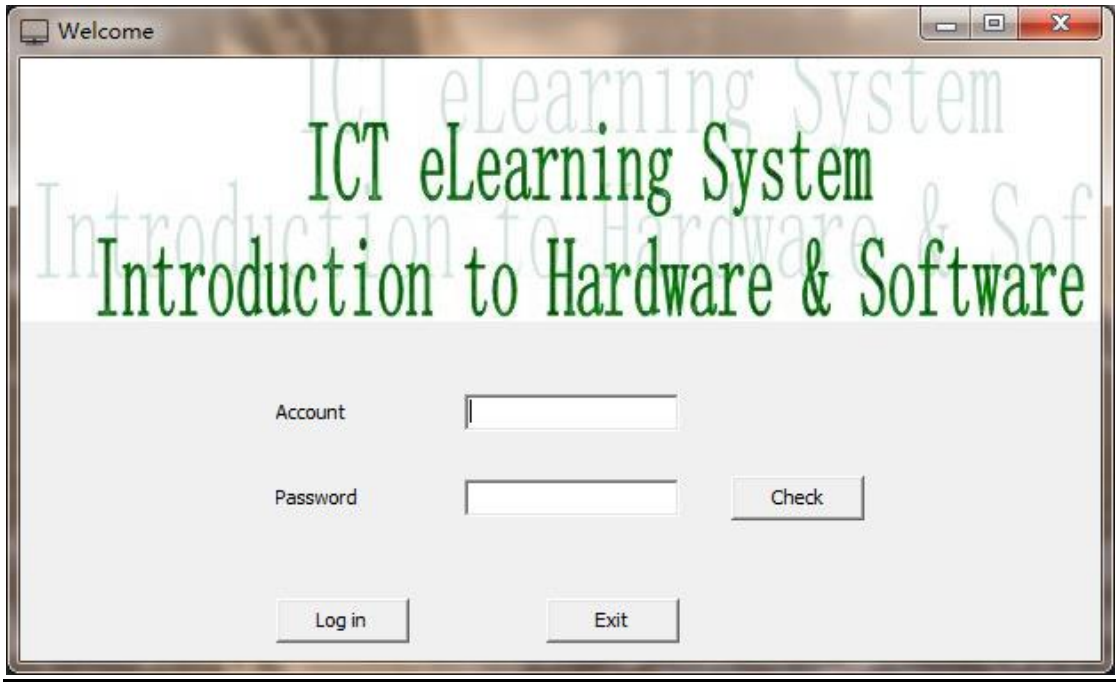
Testing

Testing Plan

- *Login Page*
  - login with valid username and password
  - press “Check” button to display the password
  - press “Log in” button or Enter key to login
  - press “Exit” button to exit the program
- *Login Information Form*
  - press “Log out” button to go back to the Login Page
- *Menu*
  - press the 4 function buttons to open the represented functions’ dialogs
  - press “Log out” button to go back to the Login Page
- *Quiz*
  - choose the correct answers and then press “Confirm” button
  - choose the wrong answers and then press “Confirm” button
  - press “Next” button to go to the next question
  - press “Give Up” button to go back to the Menu
- *Note*
  - press the 4 navigation buttons to open the represented pages
  - press “Back” button to go back to the Menu

Testing of the Program

Login Page



For administrator login,

Account	Password	Expected Result	Actual Result	Results Mached
adm	adm	A new form shows	A new form shows	Yes
adm	123	Warning shows	Warning shows	Yes
123	adm	Warning shows	Warning shows	Yes
123	123	Warning shows	Warning shows	Yes
		Warning shows	Warning shows	Yes

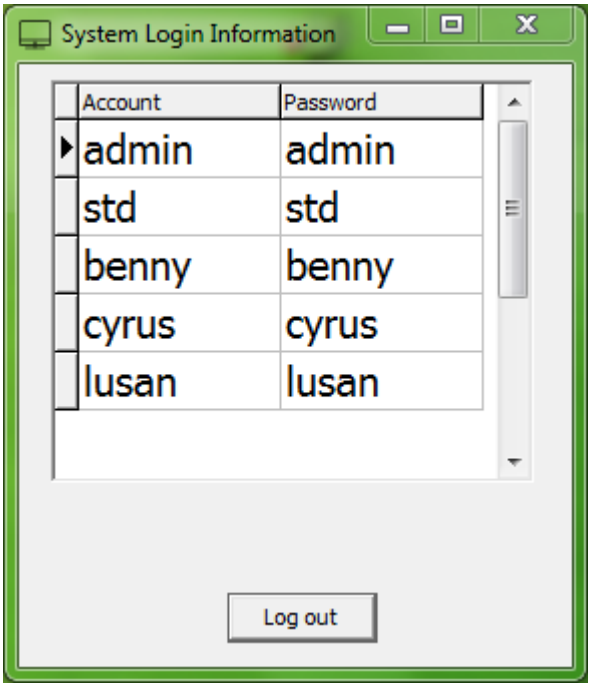
For student “std” login, the password is “std”.

Account	Password	Expected Result	Actual Result	Results Mached
std	std	A new form shows	A new form shows	Yes
std	123	Warning shows	Warning shows	Yes
123	std	Warning shows	Warning shows	Yes
123	123	Warning shows	Warning shows	Yes
		Warning shows	Warning shows	Yes

For buttons,

Button	Expected Result	Actual Result	Results Mached
“Check” is clicked	Password shows	Password shows	Yes
“Log in” is clicked with correct data inputted	A new form shows	A new form shows	Yes
“Log in” is clicked with wrong data inputted	Warning shows	Warning shows	Yes
“Exit” is clicked	Program closes	Program closes	Yes
Pressing Enter key with correct data inputted	A new form shows	A new form shows	Yes
Pressing Enter key with wrong data inputted	Warning shows	Warning shows	Yes

Login Information Form



For buttons,

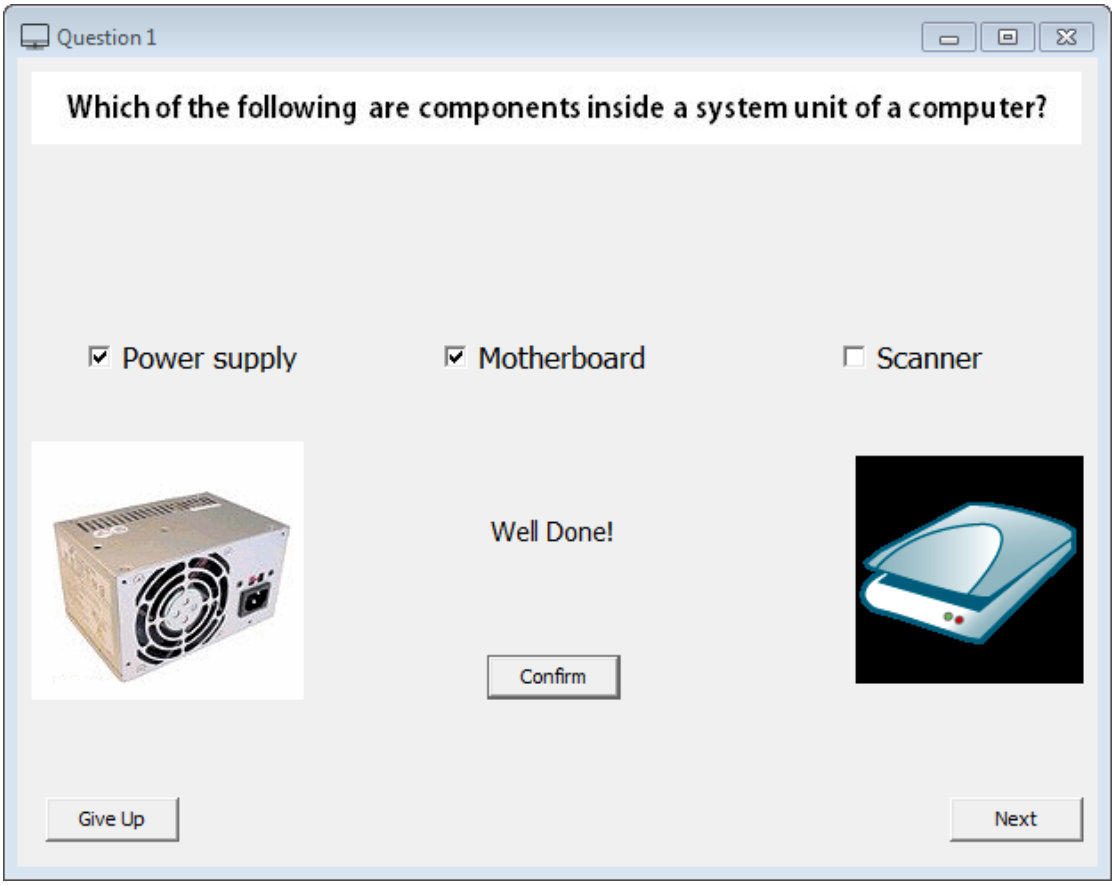
Button	Expected Result	Actual Result	Results Mached
“Log out” is clicked	Back to the Login Page	Back to the Login Page	Yes

Menu

For buttons,

Button	Expected Result	Actual Result	Results Mached
“Quiz” is clicked	A new form shows	A new form shows	Yes
“Note” is clicked	A new form shows	A new form shows	Yes
“Website” is clicked	A new form shows	A new form shows	Yes
“Video” is clicked	A new form shows	A new form shows	Yes
“Log out” is clicked	Back to the Login Page	Back to the Login Page	Yes

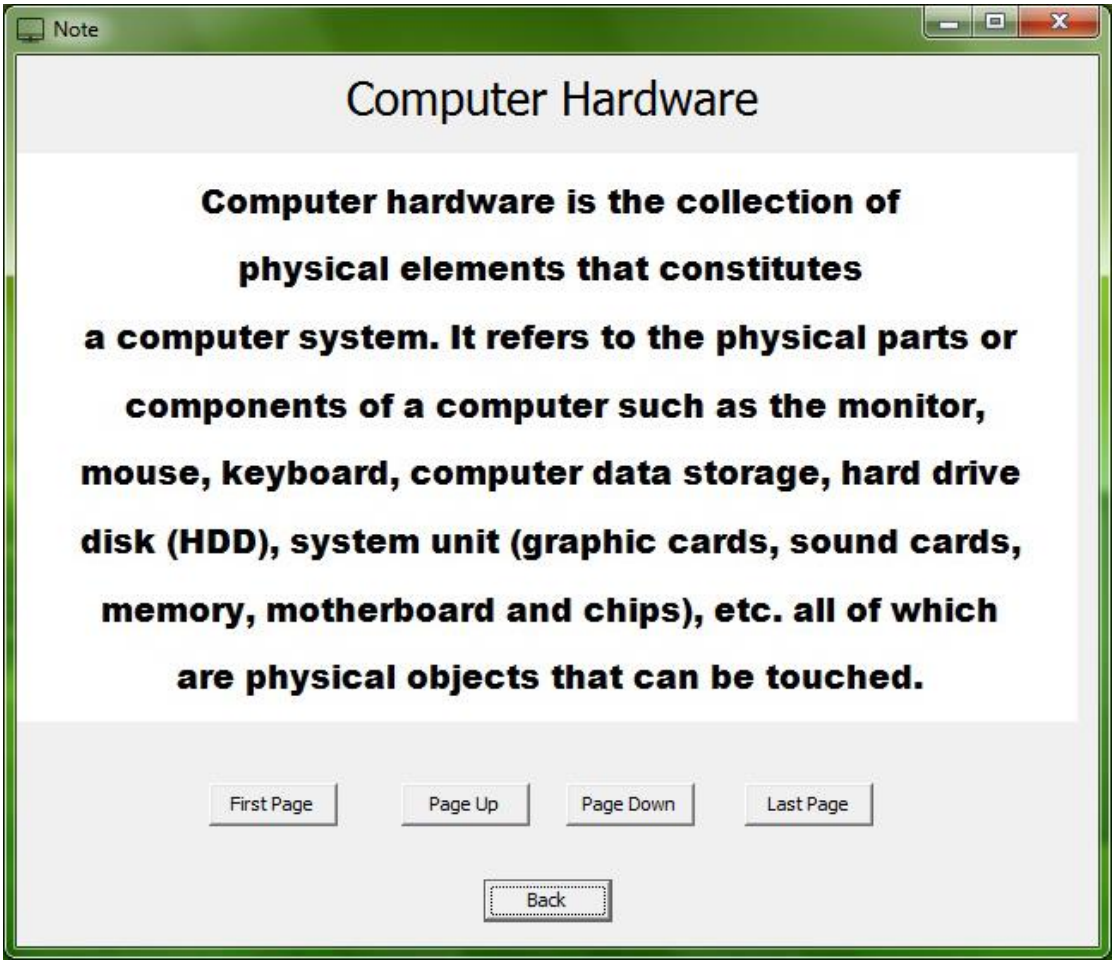
Quiz



For buttons,

Button	Expected Result	Actual Result	Results Mached
“Confirm” is clicked with correct answers	“Well Done!” shows	“Well Done!” shows	Yes
“Confirm” is clicked with wrong answers	“You have got the wrong answer!” shows	“You have got the wrong answer!” shows	Yes
“Next” is clicked	A new form shows	A new form shows	Yes
“Give Up” is clicked	Back to the Menu	Back to the Menu	Yes

Note



For buttons,

Button	Expected Result	Actual Result	Results Mached
“First Page” is clicked	First picture shows	First picture shows	Yes
“Page Up” is clicked	Previous picture shows	Previous picture shows	Yes
“Page Down” is clicked	Next picture shows	Next picture shows	Yes
“Last Page” is clicked	Last picture shows	Last picture shows	Yes
“Back” is clicked	Back to the Menu	Back to the Menu	Yes

Evaluation

Weak Points of my Design

1) Local database

The login information is stored in a database file locally. The security level is low as users can edit the record in the database by simply using Microsoft Access.

2) Difficult to change passwords

Since the login information is not stored in a database or a text files, users are not able to change passwords by themselves. The code should be altered and the program is required to rebuild if users want to change their passwords.

3) No marking function

The system will not calculate and save the marks after users submitted their answers. As a result, no ranking is provided. Teachers may find difficult to follow students' learning progress and students may think that the quiz is boring.

4) Persisted questions

The questions in the quiz will not change after the program was built up. Users may not want to do them again as there are no new questions.

Improvements

✧ Online database

The login information can be stored in an online database instead of a local database in order to increase the security. Moreover, this can allow users to change their passwords by linking the system to the online database

✧ Marking function

The marking function can be added to the system so that students can know their performance among their classmates and this increase their learning interest. Furthermore, it is easy for teachers

to distinguish between the talented students and the normal students.

✧ Random questions

A question bank can be inserted to the system. Thus, the system can choose different questions randomly each time when users do the quiz.

Difficulties encountered

- I had to learn to use Delphi. This was the first time that I tried to use Delphi to build up a system. Although I have learnt some basic Pascal language to construct a command line interface for a program in the school, it was still quite difficult for me to use Delphi to construct a graphical user interface for the system.
- It is difficult for me to add a specific function that I want to the system. This is because I have limited knowledge about built-in reserved words, functions, procedures and libraries in Delphi. It is also hard to find instructions about how to code the specific function that I want.

Conclusion

E-learning is an approach in which computer and information technology is applied to support learning. Nowadays, an increasing number of schools are adopting this approach to enrich the learning and teaching environment.

It usually include a graphical user interface for browsing through informative content and assessment tools for evaluating learning progress. It is believed that e-learning allows students to learn at their own pace.

My target is developing an e-learning package for my schoolmates.

I learnt a lot in the process of developing this e-learning package, mainly on programming. When I first used Delphi, I did not know where I should start. However, I paid a lot of effort to learn programming by online tutorials.

Now, I am able to finish this project. This is quite hard to archieve and very time-consuming. Nevertheless, I can still find the joy of doing this meaningful matter.

Reference

Book

Longman New Senior Secondary Information and Communication Technology Compulsory Workbook Volume 2

http://nss.ict.ilongman.com/index.php?section=10

Website

Wikipedia

http://en.wikipedia.org

愛瘋誌

http://www.myapp.com.tw

YouTube

https://www.youtube.com

the Password Reveal (Eye) button

http://answers.microsoft.com/en-us/ie/wiki/ie11-iewindows8_1/the-use-of-the-password-reveal-eye-button-in/19a9dee2-fb0c-4c26-a6bc-ac02cf98d80e

Picture

Microsoft Office Clip Art

http://office.microsoft.com/en-gb/word-help/add-graphics-and-keep-them-where-you-want-them-RZ001026465.aspx?section=9

Menu

http://www.braunhousehold.com/Global/Recipe-Images/Multiquick-3-Recipes/Main-Course/the-italian-menu-620x325.jpg

Power

http://www.powersourceonline.com/buy-equipment/hp_parts-338296B21-cy-en.jsa

Scanner

http://www.iconseeker.com/search-icon/tulliana-2/scanner-5.html

Trackball

http://www.ldlc.com/informatique/peripherique-pc/trackball/c4616

Touchpad

http://www.i2clipart.com/clipart-sign-language-d-finger-pointing-8684

Correct & Wrong

https://code.google.com/p/correctwrong

Program Icon

https://www.iconfinder.com/icons/185024/computer_imac_icon#size=128

Appendix

Program code

Main program

```
program Project1;


uses
  Forms,
  aUnit1 in 'aUnit1.pas' {aForm1},
  sUnit1 in 'sUnit1.pas' {sForm1},
  sUnit2 in 'sUnit2.pas' {sForm2},
  sUnit2_1 in 'sUnit2_1.pas' {sForm2_1},
  sUnit2_2 in 'sUnit2_2.pas' {sForm2_2},
  sUnit3 in 'sUnit3.pas' {sForm3},
  sUnit6 in 'sUnit6.pas' {sForm6},
  sUnit6_1 in 'sUnit6_1.pas' {sForm6_1},
  sUnit6_2 in 'sUnit6_2.pas' {sForm6_2},
  sUnit6_3 in 'sUnit6_3.pas' {sForm6_3},
  sUnit7 in 'sUnit7.pas' {sForm7},
  sUnit7_1 in 'sUnit7_1.pas' {sForm7_1},
  sUnit7_2 in 'sUnit7_2.pas' {sForm7_2},
  sUnit7_3 in 'sUnit7_3.pas' {sForm7_3},
  Unit0 in 'Unit0.pas' {Form0};


{$R *.res}
```

```
begin
  Application.Initialize;
  Application.Title := 'Warning';
  Application.CreateForm(TForm0, Form0);
  Application.CreateForm(TaForm1, aForm1);
  Application.CreateForm(TsForm2, sForm2);
  Application.CreateForm(TsForm2_1, sForm2_1);
  Application.CreateForm(TsForm6_1, sForm6_1);
  Application.CreateForm(TsForm2_2, sForm2_2);
  Application.CreateForm(TsForm1, sForm1);
  Application.CreateForm(TsForm3, sForm3);
  Application.CreateForm(TsForm6, sForm6);
  Application.CreateForm(TsForm6_2, sForm6_2);
  Application.CreateForm(TsForm6_3, sForm6_3);
  Application.CreateForm(TsForm7, sForm7);
  Application.CreateForm(TsForm7_1, sForm7_1);
  Application.CreateForm(TsForm7_2, sForm7_2);
  Application.CreateForm(TsForm7_3, sForm7_3);
  form0.Position:= forms.poScreenCenter;
  aform1.Position:= forms.poScreenCenter;
  sform1.Position:= forms.poScreenCenter;
  sform2.Position:= forms.poScreenCenter;
  sform2_1.Position:= forms.poScreenCenter;
  sform2_2.Position:= forms.poScreenCenter;
  sform3.Position:= forms.poScreenCenter;
  sform6.Position:= forms.poScreenCenter;
```

```
  sform6_1.Position:= forms.poScreenCenter;
  sform6_2.Position:= forms.poScreenCenter;
  sform6_3.Position:= forms.poScreenCenter;
  sform7.Position:= forms.poScreenCenter;
  sform7_1.Position:= forms.poScreenCenter;
  sform7_2.Position:= forms.poScreenCenter;
  sform7_3.Position:= forms.poScreenCenter;
  Application.Run;
end.
```

Unit: Login Page

```
unit Unit0;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls, DB, ADODB;

type
  TForm0 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
```

```
    Label2: TLabel;
    Button3: TButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Button3MouseDown(Sender: TObject; Button:
TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button3MouseUp(Sender: TObject; Button:
TMouseButton;
      Shift: TShiftState; X, Y: Integer);


  private
    { Private declarations }
  public
    { Public declarations }
  end;


const
  n = 5;


var
  Form0: TForm0;
  ac: array[1..n] of string;
```

```
  pw: array [1..n] of string;
  acpass, pwpass: boolean;

implementation

uses aUnit1, sUnit1;

{$R *.dfm}

procedure TForm0.Button1Click(Sender: TObject);
var
  i: integer;
begin
      if (edit1.Text = '') or (edit2.Text = '') then
        showmessage('Please enter your account and password!')
      else
      begin
      if (edit1.Text = ac[1]) and (edit2.Text = pw[1]) then
      begin
        aform1.Show;
        form0.Hide;
        acpass:= true;
        pwpass:= true;
      end;

      for i:= 2 to n do
```

```
      begin
        if (edit1.Text = ac[i]) and (edit2.Text= pw[i])  then
        begin
          sform1.Show;
          form0.Hide;
          acpass:= true;
          pwpass:= true;
        end;
      end;

      if not(acpass) or not(pwpass) then
        showmessage('Please try again!');
      end;
end;

procedure TForm0.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Ord(Key) = VK_RETURN then
    edit2.SetFocus;
end;

procedure TForm0.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if Ord(Key) = VK_RETURN then
      button1.Click;
end;
```

```
procedure TForm0.FormShow(Sender: TObject);
begin
  edit1.Text:= '';
  edit2.Text:= '';
  edit1.SetFocus;

  ac[1]:= 'admin';
  ac[2]:= 'std';
  ac[3]:= 'benny';
  ac[4]:= 'cyrus';
  ac[5]:= 'lusan';

  pw[1]:= 'admin';
  pw[2]:= 'std';
  pw[3]:= 'benny';
  pw[4]:= 'cyrus';
  pw[5]:= 'lusan';

  acpass:= false;
  pwpass:= false;
end;

procedure TForm0.Button2Click(Sender: TObject);
begin
  application.Terminate;
```

```
end;


procedure TForm0.Button3MouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  edit2.PasswordChar:= #0;
end;


procedure TForm0.Button3MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  edit2.PasswordChar:= '*';
end;


end.
```

Unit: Login Information Form

```
unit aUnit1;


  interface


  uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB, ADODB;


type
  TaForm1 = class(TForm)
    Button1: TButton;
    DataSource1: TDataSource;
    ADOQuery1: TADOQuery;
    ADOConnection1: TADOConnection;
    DBGrid1: TDBGrid;
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;


var
```

```
  aForm1: TaForm1;
  path: string;

implementation

uses Unit0;

{$R *.dfm}

procedure TaForm1.Button1Click(Sender: TObject);
begin
  if MessageBox(0, 'Do you really want to logout?', 'Confirmation',
MB_YesNo+MB_ICONINFORMATION) = idYes then
    begin
      aform1.Hide;
      form0.show;
    end;
end;

procedure TaForm1.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  ADOQuery1.Close;
end;

procedure TaForm1.FormCreate(Sender: TObject);
begin
```

```
  path:= extractfilepath(application.exename);

aForm1.ADOConnection1.ConnectionString:='Provider=MSDASQL.
1;Persist Security Info=False;Extended Properties="DSN=MS
Access
Database;DBQ='+path+'login.mdb;DefaultDir='+path+';DriverId=25;
FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;UID=admin;"';

  aForm1.ADOConnection1.Open;

  aForm1.ADOQuery1.Active:=true;

end;


end.
```

Unit: Menu

```
unit sUnit1;


interface


uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,

  Dialogs, StdCtrls, jpeg, ExtCtrls;


type

  TsForm1 = class(TForm)

    Button1: TButton;

    Label1: TLabel;

    Label2: TLabel;

    Button7: TButton;
```

```
    Button6: TButton;
    Button2: TButton;
    Button3: TButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  procedure quote(var a: string);

var
  sForm1: TsForm1;

implementation

uses Unit0, sUnit3, sUnit7, sUnit6, sUnit2;

{$R *.dfm}
```

```
procedure TsForm1.Button1Click(Sender: TObject);
begin
  if MessageBox(0, 'Do you really want to logout?', 'Confirmation',
MB_YesNo+MB_ICONINFORMATION) = idYes then
  begin
    sform1.Hide;
    form0.show;
  end;
end;

procedure TsForm1.Button7Click(Sender: TObject);
begin
    sform1.Hide;
    sform7.show;
end;

procedure TsForm1.Button2Click(Sender: TObject);
begin
  sform1.Hide;
  sform2.Show;
end;

procedure TsForm1.Button3Click(Sender: TObject);
begin
  sform1.Hide;
  sform3.Show;
end;
```

```
procedure TsForm1.Button6Click(Sender: TObject);
begin
    sform1.Hide;
    sform6.show;
end;

procedure TsForm1.FormCreate(Sender: TObject);
var
  b: string;
begin
 quote(b);
 label2.Caption:= b;
end;

procedure quote(var a: string);
const
  s= 3;
var
  n: integer;
begin
  randomize;
  n:= random(s);
  case n of
  0: a:= 'A system unit is a case that contains'#13#10'circuit boards,
a power supply'#13#10'and storage devices.';
```

```
  1: a:= 'Computer software is a sequence of'#13#10'instructions
which instructs'#13#10'a computer to perform specific tasks.';

  2: a:= 'Microdrive has become one of'#13#10'the essential storage
components'#13#10'for handy electronic devices.';

  end;

end;


end.
```

Unit: Quiz 1

```
unit sUnit2;


interface


uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,

  Dialogs, StdCtrls, Buttons, jpeg, ExtCtrls, MPlayer;


type

  TsForm2 = class(TForm)

    Button1: TButton;

    Image1: TImage;

    CheckBox1: TCheckBox;

    CheckBox2: TCheckBox;

    CheckBox3: TCheckBox;

    Button2: TButton;
```

```
    Button3: TButton;
    Label1: TLabel;
    Image2: TImage;
    Image3: TImage;
    MediaPlayer1: TMediaPlayer;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormHide(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm2: TsForm2;

implementation

uses sUnit1, sUnit2_1;

{$R *.dfm}

procedure TsForm2.Button1Click(Sender: TObject);
begin
```

```
  sform2.Hide;
  sform1.show;
end;

procedure TsForm2.Button2Click(Sender: TObject);
begin
  label1.Caption:= 'You have got the wrong answer!';
  mediaplayer1.FileName := 'wrong.mp3';
  mediaplayer1.Open;
  mediaplayer1.Play;
  if (checkbox1.Checked and checkbox2.Checked) and
(checkbox3.Checked = false) then
  begin
    label1.Caption:= 'Well Done!';
    button3.Show;
    mediaplayer1.FileName := 'correct.mp3';
    mediaplayer1.Open;
    mediaplayer1.Play;
  end;
end;
procedure TsForm2.Button3Click(Sender: TObject);
begin
  sform2.Hide;
  sform2_1.Show;
end;

procedure TsForm2.FormHide(Sender: TObject);
```

```
begin
  checkbox1.Checked:= false;
  checkbox2.Checked:= false;
  button3.Hide;
  label1.Caption:='';
end;

end.
```

Unit: Quiz 2

```
unit sUnit2_1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls, MPlayer;

type
  TsForm2_1 = class(TForm)
    Button3: TButton;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    RadioButton3: TRadioButton;
    RadioButton1: TRadioButton;
```

```
    RadioButton2: TRadioButton;
    RadioButton4: TRadioButton;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    MediaPlayer1: TMediaPlayer;
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormHide(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm2_1: TsForm2_1;

implementation

uses sUnit1, sUnit2_2;

{$R *.dfm}

procedure TsForm2_1.Button1Click(Sender: TObject);
```

```
begin
  sform2_1.Hide;
  sform1.show;
end;

procedure TsForm2_1.Button2Click(Sender: TObject);
begin
  label1.Caption:= 'You have got the wrong answer!';
  mediaplayer1.FileName := 'wrong2.mp3';
  mediaplayer1.Open;
  mediaplayer1.Play;
  if radiobutton4.Checked then
  begin
    label1.Caption:= 'Well Done!';
    button3.Show;
    mediaplayer1.FileName := 'correct2.mp3';
    mediaplayer1.Open;
    mediaplayer1.Play;
  end;
end;

procedure TsForm2_1.Button3Click(Sender: TObject);
begin
  sform2_1.Hide;
  sform2_2.Show;
end;
```

```
procedure TsForm2_1.FormHide(Sender: TObject);
begin
  radiobutton1.Checked:= false;
  radiobutton2.Checked:= false;
  radiobutton3.Checked:= false;
  radiobutton4.Checked:= false;
  button3.Hide;
  label1.Caption:='';
end;

end.
```

Unit: Quiz 3

```
unit sUnit2_2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls, MPlayer;

type
  TsForm2_2 = class(TForm)
    Button3: TButton;
    Button1: TButton;
```

```
    Button2: TButton;
    Label1: TLabel;
    ComboBox1: TComboBox;
    Edit1: TEdit;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    MediaPlayer1: TMediaPlayer;
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Edit1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormHide(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm2_2: TsForm2_2;

implementation

uses sUnit1;
```

```
{$R *.dfm}

procedure TsForm2_2.Button1Click(Sender: TObject);
begin
  sform2_2.Hide;
  sform1.show;
end;

procedure TsForm2_2.Button2Click(Sender: TObject);
begin
  label1.Caption:= 'You have got the wrong answer!';
  image3.Show;
  image2.Hide;
  mediaplayer1.FileName := 'wrong3.mp3';
  mediaplayer1.Open;
  mediaplayer1.Play;
  if (edit1.Text = 'minimum') and (combobox1.Text = 'hard disk') then
  begin
    label1.Caption:= 'Well Done!';
    button3.Show;
    image2.Show;
    image3.Hide;
    mediaplayer1.FileName := 'correct3.mp3';
    mediaplayer1.Open;
    mediaplayer1.Play;
  end;
```

```
end;

procedure TsForm2_2.Button3Click(Sender: TObject);
begin
  sform2_2.Hide;
  sform1.show;
end;

procedure TsForm2_2.Edit1Click(Sender: TObject);
begin
  edit1.Clear;
end;

procedure TsForm2_2.FormHide(Sender: TObject);
begin
  edit1.Text:='Please enter your answer in lower case!';
  combobox1.Text:='Please choose the answer from the list!';
  image2.Hide;
  image3.Hide;
  button3.Hide;
  label1.Caption:='';
end;

end.
```

Unit: Note

```
unit sUnit3;


interface


uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg;


type
  TsForm3 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm3: TsForm3;
  i: integer;

const
  a = 'Computer Hardware';
  b = 'Computer Software';
  c = 'Mainframe Computer';

implementation

uses sUnit1;

{$R *.dfm}

procedure TsForm3.Button1Click(Sender: TObject);
begin
  sform3.Hide;
  sform1.show;
end;
```

```
procedure TsForm3.Button2Click(Sender: TObject);
begin
  case i of
    1: showmessage('This is the first page already!');
    2: begin
         image1.Show;
         image2.Hide;
         label1.Caption:= a;
       end;
    3: begin
         image2.Show;
         image3.Hide;
         label1.Caption:= b;
       end;
  end;
  if i>1 then
    i:= i-1;
end;

procedure TsForm3.Button3Click(Sender: TObject);
begin
  case i of
    1: begin
         image2.Show;
         image1.Hide;
```

```
        label1.Caption:= b;
       end;
    2: begin
        image3.Show;
        image2.Hide;
        label1.Caption:= c;
       end;
    3: showmessage('This is the last page already!');
  end;
  if i<3 then
    i:= i+1;
end;

procedure TsForm3.Button4Click(Sender: TObject);
begin
  i:= 1;
  label1.Caption:= a;
  image1.Show;
  image2.Hide;
  image3.Hide;
end;

procedure TsForm3.Button5Click(Sender: TObject);
begin
  i:= 3;
  label1.Caption:= c;
```

```
  image3.Show;
  image1.Hide;
  image2.Hide;
end;

procedure TsForm3.FormShow(Sender: TObject);
begin
  i:= 1;
  label1.Caption:= a;
  image1.Show;
  image2.Hide;
  image3.Hide;
end;

end.
```

Unit: Website 0

```
unit sUnit6;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls;

type
```

```
  TsForm6 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm6: TsForm6;

implementation

uses sUnit1, sUnit6_1, sUnit6_2, sUnit6_3;

{$R *.dfm}

procedure TsForm6.Button1Click(Sender: TObject);
```

```
begin
  sform6.Hide;
  sform1.show;
end;

procedure TsForm6.Button2Click(Sender: TObject);
begin
  sform6.Hide;
  sform6_1.show;
end;

procedure TsForm6.Button3Click(Sender: TObject);
begin
  sform6.Hide;
  sform6_2.show;
end;

procedure TsForm6.Button4Click(Sender: TObject);
begin
  sform6.Hide;
  sform6_3.show;
end;

end.
```

Unit: Website 1

```
unit sUnit6_1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, OleCtrls, SHDocVw;

type
  TsForm6_1 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm6_1: TsForm6_1;

implementation
```

```
uses sUnit6;

{$R *.dfm}

procedure TsForm6_1.Button1Click(Sender: TObject);
begin
  sForm6_1.Hide;
  sform6.show;
end;

procedure TsForm6_1.FormShow(Sender: TObject);
begin

webbrowser1.Navigate('http://en.wikipedia.org/wiki/Computer_hard
ware');
end;

end.
```

Unit: Website 2

```
unit sUnit6_2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
```

```
  Dialogs, StdCtrls, OleCtrls, SHDocVw;

type
  TsForm6_2 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm6_2: TsForm6_2;

implementation

uses sUnit6;

{$R *.dfm}

procedure TsForm6_2.Button1Click(Sender: TObject);
begin
  sForm6_2.Hide;
```

```
  sform6.show;
end;

procedure TsForm6_2.FormShow(Sender: TObject);
begin
  webbrowser1.Navigate('http://en.wikipedia.org/wiki/Software');
end;

end.
```

Unit: Website 3

```
unit sUnit6_3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, OleCtrls, SHDocVw;

type
  TsForm6_3 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
```

```
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm6_3: TsForm6_3;

implementation

uses sUnit6;

{$R *.dfm}

procedure TsForm6_3.Button1Click(Sender: TObject);
begin
  sForm6_3.Hide;
  sform6.show;
end;

procedure TsForm6_3.FormShow(Sender: TObject);
begin
  webbrowser1.Navigate('http://www.myapp.com.tw/app-
%E5%88%B0%E5%BA%95%E6%98%AF%E4%BB%80%E9%BA
%BC%EF%BC%9F%E5%9F%BA%E6%9C%AC%E5%B8%B8%E8
%AD%98-qa/');
end;
```

```
end.

Unit: Video 0
unit sUnit7;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls;

type
  TsForm7 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
```

```
    { Public declarations }
  end;

var
  sForm7: TsForm7;

implementation

uses sUnit1, sUnit7_1, sUnit7_2, sUnit7_3;

{$R *.dfm}

procedure TsForm7.Button1Click(Sender: TObject);
begin
   sform7.Hide;
   sform1.show;
end;

procedure TsForm7.Button2Click(Sender: TObject);
begin
  sform7.Hide;
  sform7_1.show;
end;

procedure TsForm7.Button3Click(Sender: TObject);
begin
```

```
  sform7.Hide;
  sform7_2.show;
end;

procedure TsForm7.Button4Click(Sender: TObject);
begin
  sform7.Hide;
  sform7_3.show;
end;

end.
```

Unit: Video 1

```
unit sUnit7_1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, OleCtrls, SHDocVw, StdCtrls, MSHtml;

type
  TsForm7_1 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
```

```
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  sForm7_1: TsForm7_1;

implementation

uses sUnit7;

{$R *.dfm}

procedure TsForm7_1.Button1Click(Sender: TObject);
begin
  sform7_1.hide;
  sform7.show;
end;

procedure TsForm7_1.FormShow(Sender: TObject);
begin

webbrowser1.Navigate('http://www.youtube.com/embed/6YwUf3qV
uCE');
```

```
end;


end.
```

Unit: Video 2

```
unit sUnit7_2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, OleCtrls, SHDocVw, StdCtrls, MSHtml;

type
  TsForm7_2 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

```
  sForm7_2: TsForm7_2;

implementation

uses sUnit7;

{$R *.dfm}

procedure TsForm7_2.Button1Click(Sender: TObject);
begin
  sform7_2.hide;
  sform7.show;
end;

procedure TsForm7_2.FormShow(Sender: TObject);
begin

webbrowser1.Navigate('https://www.youtube.com/embed/rNl9ol_rPMs');
end;

end.
```

Unit: Video 3

```
unit sUnit7_3;

interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, OleCtrls, SHDocVw, StdCtrls, MSHtml;


type
  TsForm7_3 = class(TForm)
    Button1: TButton;
    WebBrowser1: TWebBrowser;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;


var
  sForm7_3: TsForm7_3;


implementation


uses sUnit7;


{$R *.dfm}
```

```
procedure TsForm7_3.Button1Click(Sender: TObject);
begin
  sform7_3.hide;
  sform7.show;
end;

procedure TsForm7_3.FormShow(Sender: TObject);
begin

webbrowser1.Navigate('http://www.youtube.com/embed/BiWP_C7CX78');
end;

end.
```