

AS LEVEL

FACT FILES

Software Systems Development

AS2 Event Driven Programming

(iii) USING MULTIPLE FORMS

For first teaching from September 2013

For first AS award in Summer 2014

For first A Level award in Summer 2015



software
systems
development

FACT FILE

software systems development

AS2 Event Driven Programming (ii) UNDERSTANDING EVENTS



Learning Outcomes

Students should be able to:

- Understand the use and application of multiple forms in an event driven application;
- Design and implement multiple forms for an event driven application;
- Implement appropriate navigation:
 - Menus;
 - Toolbars;
 - Buttons.



Course Content

Real Systems require many processes to maintain the data records in an up to date state. In an event driven application each process could be managed on a separate form or a number of related processes could be managed on one form.

The startup form is instantiated in the program.cs code.

E.g. FrmSplash is first form of application

```
Program.cs* x ProjInitial FrmIntro.cs FrmSplash.cs FrmSplash.cs [Design] FrmIntro.cs [Design]
ProjInitial.Program
  using System;
  using System.Collections.Generic;
  using System.Linq;
  using System.Windows.Forms;

  namespace ProjInitial
  {
    static class Program
    {
      /// <summary>
      /// The main entry point for the application.
      /// </summary>
      [STAThread]
      static void Main()
      {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new FrmSplash());
      }
    }
  }
```

First form of application
(Initially set to first form designed)

doubleClick to show code window
(This can be changed to the form being currently tested and, finally the appropriate first form)

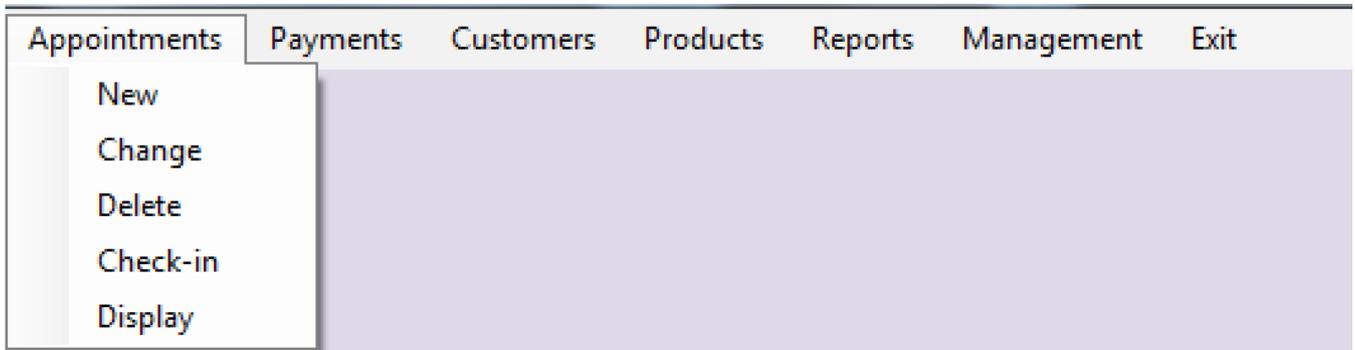
Navigation of Multiple Forms

Menus and Toolbars are a common means of navigation through the system options which are structured through the multiple forms.

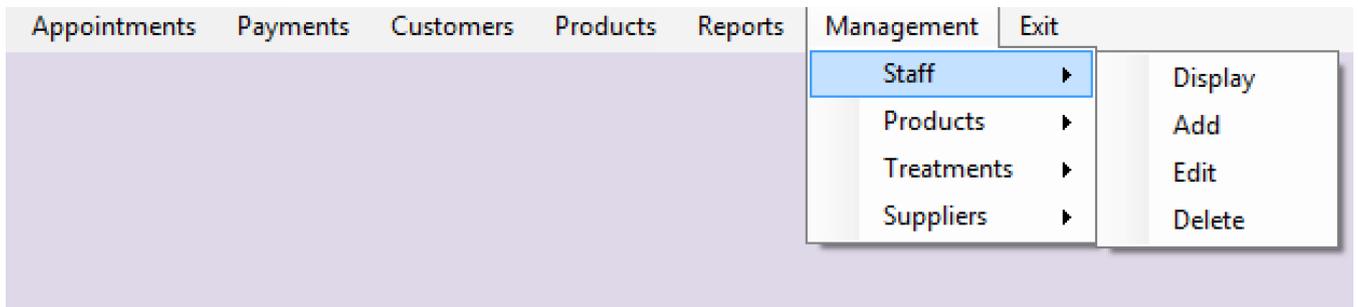
Menus support a hierarchical design of the related options.

Related options are grouped together with the most frequent options in the system ranging from left to right on the menu.

The components provided for building the Navigation are **MenuStrip and ToolStrip**



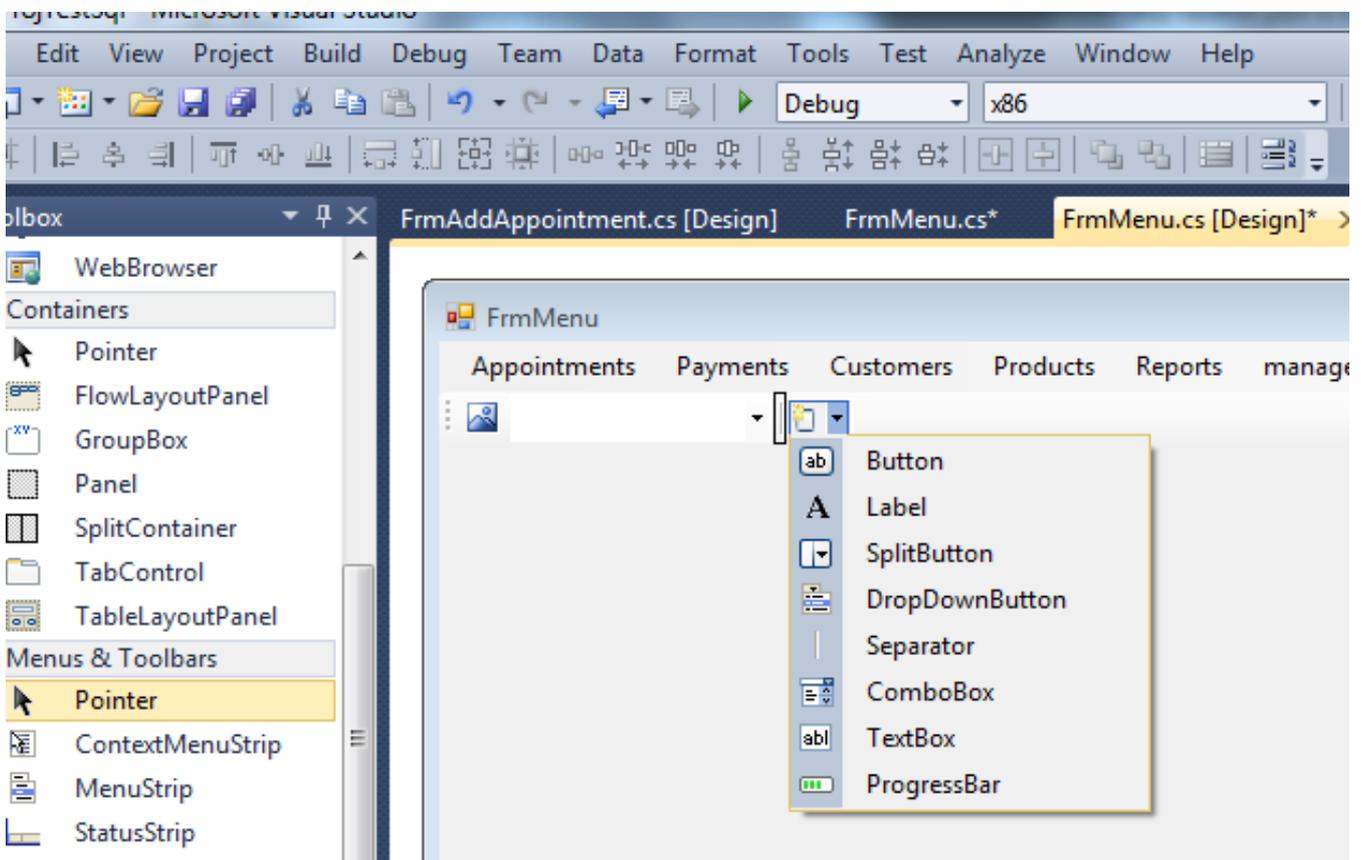
The less frequently occurring supporting table options are generally grouped to the right of the menu under System or a similar heading – e.g. Management as shown below



It is recommended that the **Exit menu** should have a drop down menuitem 'Exit System' to ensure that the exit option is not clicked by accident. An alternative is to use a MessageBox to confirm the exit.

Toolbars, also commonly used in conjunction with the system menu, hold a subset of the most frequently used options. These show icons /text depicting the associated activity and use a tooltip for further information.

Below is an example of a ToolStrip with a Button, ComboBox and Separator component chosen for the ToolStrip. The design options available are shown in the dropdown.



Buttons are also a common means of navigation through forms and are commonly presented in a panel, in a consistent area within each form. This gives a unique feel to the system.

Event - Click

The code required to open and show a form from a MenuItem is as follows:

```
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    FormType FrmAddAppointment frm = new FormType FrmAddAppointment();
    frm.ShowDialog();
}
```

The form is shown as a modal dialogue box – i.e. Only by closing the AddAppointments form can the menu form regain focus.

Parent – Child refers to the modal opening of a form using the `.ShowDialog`.

Example of click event for confirmation of exit menu from system

```
private void exitSystemToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult result;

    // Displays the Confirm Exit MessageBox.
    // You can also set a default button and/or align the buttons
    result = MessageBox.Show(this, "You are exiting system - Click OK to continue or
CANCEL to abort", "Confirm Exit", MessageBoxButtons.OKCancel);

    // Closes the parent form.
    if(result == DialogResult.Yes)
        this.Close();
}
```

keyboard shortcuts

Menu items can be selected quickly using the keyboard shortcuts. These are key sequences which include pressing a key whilst holding down the Ctrl, Alt or Shift key.

Example:- set the Exit menu item to use Alt-X

Set the **ShortcutKeys** property of the menuItem object -

A panel will appear with the option to select Ctrl, Shift or Alt and also the key to be pressed. Select Alt, and X

Care should be taken not to use conventional combinations where possible

QUESTIONS

Q1 Design a menu for a simple game which allows the player to:-

- enter their details and chose a 'player character';
- select a game from 3 levels of expertise;
- view the top scores to date;
- exit the system.

Q2 The designer wants to ensure that a player cannot accidentally choose the New Game option from the Menu. What measures could you include to achieve this.

Q3 Consider whether a Toolbar be applicable for the Game Application and if so what components would you include on it.

