

MÔN : KIỂM THỬ PHẦN MỀM

Bài thực hành số 10.1 : Quản lý tự động kết quả kiểm thử

I. Mục tiêu :

- Giúp SV làm quen với qui trình điển hình để quản lý các bảng dữ liệu miêu tả kết quả kiểm thử, cách viết tiện ích tạo tự động các bảng dữ liệu quản lý kiểm thử.

II. Nội dung :

- Tạo và duy trì các bảng dữ liệu miêu tả kết quả kiểm thử.
- Lập trình tiện ích tạo tự động các bảng dữ liệu miêu tả kết quả kiểm thử.

III. Chuẩn đầu ra :

- Sinh viên nắm vững và thực hiện thành thạo qui trình quản lý các bảng dữ liệu miêu tả kết quả kiểm thử, cách viết tiện ích tạo tự động các bảng dữ liệu quản lý kiểm thử..

IV. Qui trình :

IV.1 Tạo lịch kiểm thử và trạng thái kiểm thử ở 1 thời điểm xác định.


- Chạy Excel, soạn thảo sheet Sheet1 để miêu tả lịch kiểm thử và trạng thái kiểm thử ở tuần đầu như sau, rename sheet Sheet1 thành TestScheduling :

<i>Test case execution schedule</i>					
Date	Test case ID	Date attempted	Outcome	Severity	Date corrected
Week 1	FT-001	5 Jul	Successful		
	FT-002	6 Jul	Successful		
	FT-003	6 Jul	Successful		
	FT-004	6 Jul	Unsuccessful	1	
	FT-005	7 Jul	Successful		
	FT-006	7 Jul	Successful		
	FT-007	7 Jul	Unsuccessful	2	
	FT-008	7 Jul	Unsuccessful	1	
	FT-009	7 Jul	Unsuccessful	1	
	FT-010	8 Jul	Successful		
Week 2	FT-011				
	FT-012				
	FT-013				
	FT-014				
...	...				
Week 14	FT-095				
	FT-096				
	FT-097				
	FT-098				
	FT-099				
	FT-100				

- Soạn Sheet2 chứa bảng theo dõi lỗi như sau, rename nó thành TestTracking :

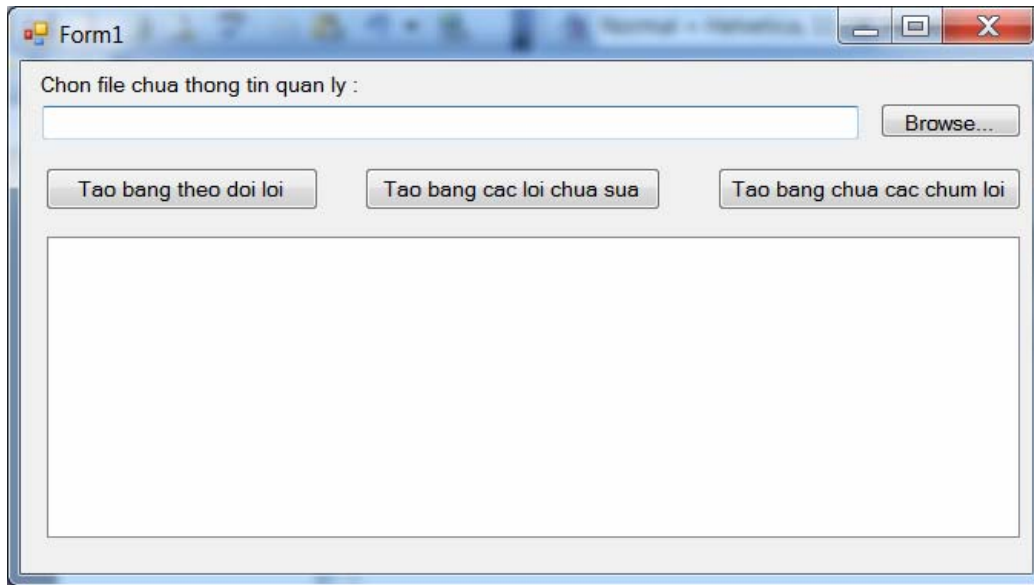
Test case execution schedule								
Date	Test case ID	Date attempted	Outcome	Severity	Date corrected			
Week 1	FT 001	5 Jul	Successful					
	FT 002	6 Jul	Successful					
	FT 003	6 Jul	Successful					
	FT 004	6 Jul	Unsuccessful	1				
	FT 005	7 Jul	Successful					
	FT 006	7 Jul	Successful					
	FT 007	7 Jul	Unsuccessful	2				
	FT 008	7 Jul	Unsuccessful	1				
	FT 009	7 Jul	Unsuccessful	1				
	FT 010	8 Jul	Successful					
Week 2	FT 011							
	FT 012		Defect tracking log					
			Defect ID	Date discovered	Test case ID	Severity	Date assigned for correction	Date corrected
			SD-0001	6 Jul	FT-004	1		
			SD-0002	7 Jul	FT-007	2		
			SD-0003	7 Jul	FT-007	2		
			SD-0004	7 Jul	FT-007	3		
			SD-0005	7 Jul	FT-007	2		
			SD-0006	7 Jul	FT-008	1		
			SD-0007	7 Jul	FT-009	1		

IV.2 Xây dựng tiện ích tạo tự động các bảng theo dõi lỗi, bảng các lỗi chưa sửa,...

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Window, chọn icon "Windows Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. TestManage), click button OK để tạo Project theo các thông số đã khai báo.
3. Form đầu tiên của ứng dụng đã hiển thị trong cửa sổ thiết kế, việc thiết kế form là quá trình lập 4 thao tác tạo mới/xóa/hiệu chỉnh thuộc tính/tạo hàm xử lý sự kiện cho từng đối tượng cần dùng trong form.
4. Nếu cửa sổ ToolBox chưa hiển thị chi tiết, chọn menu View.Toolbox để hiển thị nó (thường nằm ở bên trái màn hình). Click chuột vào button  (Auto Hide) nằm ở góc trên phải của cửa sổ ToolBox để chuyển nó về chế độ hiển thị thường trực. Duyệt tìm phần tử Label (trong nhóm Common Controls), chọn nó, dời chuột về vị trí thích hợp trong form và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính Text = "File thông tin về lịch kiểm thử". Nếu cần, hãy thay đổi vị trí và kích thước của Label và của Form.
5. Duyệt tìm phần tử TextBox (trong nhóm Common Controls), chọn nó, dời chuột về vị trí bên phải Label vừa vẽ và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = txtPath. Nếu cần, hãy thay đổi vị trí và kích thước của TextBox.
6. Duyệt tìm phần tử Button (trong nhóm Common Controls), chọn nó, dời chuột về vị trí bên phải TextBox vừa vẽ và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = btnBrowse, thuộc tính Text="Browse...". Nếu cần, hãy thay đổi vị trí và kích thước của Button.
7. Lập bước 6 ba lần để tạo 3 button chức năng, có (Name, thuộc tính) = (btnTestAnalyse, "Tạo bảng Test Analyse"), (btnTestTracking, "Tạo bảng Test Tracking"), (btnTestBacklog, "Tạo bảng Test Backlog").

8. Duyệt tìm phần tử ListBox (trong nhóm Common Controls), chọn nó, dờ chuột về vị trí bên dưới Label được vẽ ở bước 4 và vẽ nó với kích thước mong muốn. Hiệu chỉnh thuộc tính (Name) = lbOutput. Nếu cần, hãy thay đổi vị trí và kích thước của ListBox.

Sau khi thiết kế xong, Form có dạng sau :



9. Chọn menu Project.Add Reference, chọn tab COM, duyệt tìm thư viện Microsoft Excel 14.0 Library và add thư viện này vào Project.
10. Dờ chuột về button "Browse...", ấn kép chuột vào nó để tạo hàm xử lý sự kiện Click chuột cho button, cửa sổ mã nguồn sẽ hiển thị để ta bắt đầu viết code cho hàm. Lưu ý rằng để tạo hàm xử lý sự kiện bất kỳ cho đối tượng 1 cách chính quy, ta phải hiển thị cửa sổ thuộc tính của đối tượng, rồi hiển thị danh sách các sự kiện rồi mới định nghĩa hàm xử lý sự kiện mong muốn.
11. Viết code cho hàm btnBrowse_Click() như sau :
- ```
private void btnBrowse_Click(object sender, EventArgs e)
{
 //tạo form duyệt chọn file
 OpenFileDialog dlg = new OpenFileDialog();
 //hiển thị form duyệt chọn file để người dùng duyệt chọn file làm việc
 DialogResult ret = dlg.ShowDialog();
 //kiểm tra quyết định của người dùng, nếu người dùng chọn OK thì hiển thị pathname
 if (ret == DialogResult.OK)
 {
 txtPath.Text = dlg.FileName;
 //Chạy Excel.
 app = new Microsoft.Office.Interop.Excel.Application();
 //Mở file
 workbook = app.Workbooks.Open(txtPath.Text);
 }
}
```
12. Lặp lại bước 10,11 để tạo hàm xử lý sự kiện Click chuột trên button btnTestBacklog và viết code cho hàm này như sau :
- ```
private void btnBackLog_Click(object sender, EventArgs e)
{

```

```

//Chọn wroksheet chứa bảng chuyển.
Worksheet ws1 = workbook.Worksheets["TestTracking"];
//thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
Range rg1 = ws1.Range["A1","Z10000"];
//Chọn wroksheet chứa bảng backlog.
Worksheet ws2;
try
{
    ws2 = workbook.Worksheets["TestBacklog"];
}
catch (Exception ex)
{
    //tạo worksheet mới
    app.Sheets.Add();
    //đặt tên cho worksheet mới
    app.ActiveSheet.Name = "TestBacklog";
    ws2 = app.ActiveSheet;
}
//thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
Range rg2 = ws2.Range["A1","Z10000"];
int row1 = 3;
int row2 = 3;
int col;

/*
//thiết lập vùng cell cần xử lý
Range rg2 = app.ActiveSheet.Range("A1:Z10000"); */
//xuất hàng tiêu đề
rg2.Item[1,1].Value = "Test Backlog";
//xuất hàng header
for (col = 1; col <= 6; col++) rg2.Item[2,col].Value = rg1.Item[2,col].Value;
//tìm chuỗi miêu tả lỗi
buf1 = rg1.Item[row1, 1].Value;
while (buf1 != null)
{
    //tìm chuỗi miêu tả ngày dự định sửa lỗi
    if (rg1.Item[row1, 5].Value == null)
    { //lỗi chưa sửa
        for (col = 1; col <= 6; col++) rg2.Item[row2, col].Value = rg1.Item[row1,
col].Value;
        row2++;
    }
    row1++;
    //tìm chuỗi miêu tả lỗi
    buf1 = rg1.Item[row1, 1].Value;
}
lbOutput.Items.Add("Đã tạo bảng TestBacklog");
workbook.Save();
}

```

13. Lặp lại bước 10,11 để tạo hàm xử lý sự kiện FormClosed của Form và viết code cho hàm này như sau :

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    app.Quit();
}
```

14. Lặp lại bước 10,11 để tạo hàm xử lý sự kiện Click chuột trên button btnTestTracking và viết code cho hàm này như sau :

```
private void btnTestTracking_Click(object sender, FormClosedEventArgs e)
{
    //Chọn wroksheet chứa bảng chuyển.
    Worksheet ws1 = workbook.Worksheets["TestSchedule"];
    //thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
    Range rg1 = ws1.Range["A1", "Z10000"];
    //Chọn wroksheet chứa bảng backlog.
    Worksheet ws2;
    try
    {
        ws2 = workbook.Worksheets["TestAnalyse"];
    }
    catch (Exception ex)
    {
        //tạo worksheet mới
        app.Sheets.Add();
        //đặt tên cho worksheet mới
        app.ActiveSheet.Name = "TestAnalyse";
        ws2 = app.ActiveSheet;
    }
    //thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
    Range rg2 = ws2.Range["A1", "Z10000"];
    int row1 = 3, rmax;
    int row2 = 3;
    int sumt, attt, cnts, cnte, cnt;
    int i;
    //1. Xuất tiêu đề bảng tổng kết tình hình thực hiện kiểm thử
    rg2.Item[1, 1].Value = "Bảng tổng kết tình hình thực hiện kiểm thử";
    //2. tính và xuất tổng số testcase trong kế hoạch
    //tìm mã ID của testcase
    sumt = 0;
    while (rg1.Item[row1++, 2].Value != null) sumt++;
    rmax = row1;
    rg2.Item[row2, 2].Value = "Testcases cần kiểm thử trong kế hoạch";
    rg2.Item[row2, 1].Value = sumt;
    row2++;
    //2. tính và xuất số testcase đã kiểm thử
    attt = 0;
    for (row1=3; row1<rmax; row1++) if (rg1.Item[row1, 4].Value != null) attt++;
    rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được";
    rg2.Item[row2, 1].Value = attt;
    row2++;
    //3. xuất tỉ lệ testcase đã kiểm thử
    rg2.Item[row2, 2].Value = "Tỉ lệ phần % testcase đã kiểm thử được";
```

```

rg2.Item[row2, 1].Value = attt*100/sumt + "%";
row2 = row2 + 2;
//4. xuất số testcase đã kiểm thử thành công
cnts = 0;
for (row1=3; row1<rmax; row1++) if (rg1.Item[row1, 4].Value == "Successful") cnts++;
rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được thành công";
rg2.Item[row2, 1].Value = cnts;
row2++;
rg2.Item[row2, 2].Value = "Tỉ lệ % testcases đã kiểm thử được thành công";
rg2.Item[row2, 1].Value = cnts * 100 / attt + "%"; ;
row2 = row2 + 2;
cnte = attt - cnts;
//5. xuất số testcase đã kiểm thử thất bại
rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được nhưng thất bại";
rg2.Item[row2, 1].Value = cnte;
row2++;
rg2.Item[row2, 2].Value = "Tỉ lệ % testcases đã kiểm thử nhưng thất bại";
rg2.Item[row2, 1].Value = cnte * 100 / attt + "%";
row2++;
//6. xuất số bảng phân loại mức độ tầm trọng của các testcase bị lỗi
for (i = 1; i <= 4; i++)
{
    cnt = 0;
    for (row1 = 3; row1 < rmax; row1++) if (rg1.Item[row1, 5].Value == i) cnt++;
    rg2.Item[row2 + i, 2].Value = "Severity " + i + "s";
    rg2.Item[row2 + i, 3].Value = cnt;
    rg2.Item[row2 + i, 4].Value = cnt* 100 / cnte + "%";
}
lbOutput.Items.Add("Da tao bang TestAnalyse");
workbook.Save();
}

```

15. Lặp lại bước 10,11 để tạo hàm xử lý sự kiện Click chuột trên button btnTestAnalyse và viết code cho hàm này như sau :

```

private void btnTestAnalyse_Click(object sender, FormClosedEventArgs e)
{
    //Chọn wroksheet chứa bảng chuyển.
    Worksheet ws1 = workbook.Worksheets["TestSchedule"];
    //thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
    Range rg1 = ws1.Range["A1", "Z10000"];
    //Chọn wroksheet chứa bảng backlog.
    Worksheet ws2;
    try
    {
        ws2 = workbook.Worksheets["TestAnalyse"];
    }
    catch (Exception ex)
    {
        //tạo worksheet mới
        app.Sheets.Add();
        //đặt tên cho worksheet mới
    }
}

```

```

app.ActiveSheet.Name = "TestAnalyse";
ws2 = app.ActiveSheet;
}
//thiết lập vùng xử lý = ma trận nhiều hàng, cột chứa thông tin bảng chuyển
Range rg2 = ws2.Range["A1", "Z10000"];
int row1 = 3, rmax;
int row2 = 3;
int sumt, attt, cnts, cnte, cnt;
int i;
//1. Xuất tiêu đề bảng tổng kết tình hình thực hiện kiểm thử
rg2.Item[1, 1].Value = "Bảng tổng kết tình hình thực hiện kiểm thử";
//2. tính và xuất tổng số testcase trong kế hoạch
//tìm mã ID của testcase
sumt = 0;
while (rg1.Item[row1++, 2].Value != null) sumt++;
rmax = row1;
rg2.Item[row2, 2].Value = "Testcases cần kiểm thử trong kế hoạch";
rg2.Item[row2, 1].Value = sumt;
row2++;
//2. tính và xuất số testcase đã kiểm thử
attt = 0;
for (row1=3; row1<rmax; row1++) if (rg1.Item[row1, 4].Value != null) attt++;
rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được";
rg2.Item[row2, 1].Value = attt;
row2++;
//3. xuất tỉ lệ testcase đã kiểm thử
rg2.Item[row2, 2].Value = "Tỉ lệ phần % testcase đã kiểm thử được";
rg2.Item[row2, 1].Value = attt*100/sumt + "%";
row2 = row2 + 2;
//4. xuất số testcase đã kiểm thử thành công
cnts = 0;
for (row1=3; row1<rmax; row1++) if (rg1.Item[row1, 4].Value == "Successful") cnts++;
rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được thành công";
rg2.Item[row2, 1].Value = cnts;
row2++;
rg2.Item[row2, 2].Value = "Tỉ lệ % testcases đã kiểm thử được thành công";
rg2.Item[row2, 1].Value = cnts * 100 / attt + "%"; ;
row2 = row2 + 2;
cnte = attt - cnts;
//5. xuất số testcase đã kiểm thử thất bại
rg2.Item[row2, 2].Value = "Testcases đã kiểm thử được nhưng thất bại";
rg2.Item[row2, 1].Value = cnte;
row2++;
rg2.Item[row2, 2].Value = "Tỉ lệ % testcases đã kiểm thử nhưng thất bại";
rg2.Item[row2, 1].Value = cnte * 100 / attt + "%";
row2++;
//6. xuất số bảng phân loại mức độ tầm trọng của các testcase bị lỗi
for (i = 1; i <= 4; i++)
{
    cnt = 0;

```

```

for (row1 = 3; row1 < rmax; row1++) if (rg1.Item[row1, 5].Value == i) cnt++;
rg2.Item[row2 + i, 2].Value = "Severity " + i + "s";
rg2.Item[row2 + i, 3].Value = cnt;
rg2.Item[row2 + i, 4].Value = cnt* 100 / cnte + "%";
}
lbOutput.Items.Add("Đã tạo bảng TestAnalyse");
workbook.Save();
}

```

16. Dời cursor về đầu class Form1 và thêm đoạn code sau để định nghĩa các thuộc tính dữ liệu và các tác vụ chức năng cần dùng :

```

Microsoft.Office.Interop.Excel.Application app;
Workbook workbook;
Worksheet worksheet;
String buf1, buf2;

```

17. Dời cursor về đầu file mã nguồn và thêm đoạn code sau để import các namespace cần dùng :

```

using Microsoft.Office.Core;
using Microsoft.Office.Interop.Excel;

```

18. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Hãy thử chọn file Excel chứa lịch kiểm thử, chọn button "Tạo TestBacklog" để máy tạo tự động bảng các lỗi chưa sửa từ bảng theo dõi lỗi.

19. Dừng ứng dụng, chạy Excel, mở file Excel kết quả và xem nội dung bảng BackLog có đúng theo nội dung của bảng theo dõi lỗi không.