**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



***MÔN:*** **ĐỒ ÁN MÔN HỌC CHUYÊN NGÀNH**

***ĐỒ ÁN:* NGHIÊN CỨU EXCEL OBJECT MODEL ĐỂ TỰ ĐỘNG TRÌNH BÀY SỐ LIỆU TRÊN MICROSOFT EXCEL**

*Lớp:**SE112.I11*

*Giảng viên hướng dẫn: Đỗ Thị Thanh Tuyền*

*Sinh viên thực hiện:*

*12520338 – Phạm Thanh Quân*

*T.P Hồ Chí Minh, tháng 12/2017*

**LỜI CẢM ƠN**

Để có thể hoàn thành tốt đồ án này, em xin gửi lời cám ơn chân thành tới các Thầy (Cô) đã giảng dạy môn ***Phân Tích Thiết Kế Hệ Thống Thông Tin*** và môn ***Nhập Môn Công Nghệ Phần Mềm***. Đặc biệt hơn, em xin gửi lời cảm ơn đến Cô ***Đỗ Thị Thanh Tuyền*** đã trực tiếp hướng dẫn giúp em hoàn thiện đồ án này!

Dù đã cố gắng làm việc miệt mài và nghiêm túc, song do thời gian hạn hẹp và kiến thức còn hạn chế nên không tránh khỏi sai sót, mong Cô góp ý để đồ án được hoàn thiện hơn!

**NHẬN XÉT CỦA GIẢNG VIÊN**

**MỤC LỤC**

[**LỜI CẢM ƠN** 2](#_Toc502785268)

[**NHẬN XÉT CỦA GIẢNG VIÊN** 3](#_Toc502785269)

[**MỤC LỤC** 4](#_Toc502785270)

[**I.** **GIỚI THIỆU ĐỀ TÀI** 7](#_Toc502785271)

[**1.** **Tổng quan về Excel Object Model** 7](#_Toc502785272)

[**2.** **Lý do chọn đề tài** 7](#_Toc502785273)

[**3.** **Yêu cầu của đề tài** 7](#_Toc502785274)

[**II.** **TÌM HIỂU HỆ THỐNG CÁC ĐỐI TƯỢNG TRONG EXCEL OBJECT MODEL** 8](#_Toc502785275)

[**1.** **Cách sử dụng Excel Object Model trong Visual Studio** 8](#_Toc502785276)

[**2.** **Giới thiệu hệ thống các đối tượng chính trong Excel Object Model** 9](#_Toc502785277)

[***2.1.*** ***Application Object*** 9](#_Toc502785278)

[***2.2.*** ***Workbook Object*** 9](#_Toc502785279)

[***2.3.*** ***Worksheet Object*** 10](#_Toc502785280)

[***2.4.*** ***Range Object*** 10](#_Toc502785281)

[**3.** **Làm việc với các đối tượng chính trong Excel Object Model** 10](#_Toc502785282)

[***3.1.*** ***Làm việc với Workbooks*** 10](#_Toc502785283)

[***3.2.*** ***Làm việc với Worksheets*** 11](#_Toc502785284)

[***3.3.*** ***Làm việc với Ranges*** 13](#_Toc502785285)

[**III.** **HỆ THỐNG PHẦN MỀM QUẢN LÝ ỨNG DỤNG EXCEL OBJECT MODEL** 14](#_Toc502785286)

[**1.** **Giới thiệu phần mềm Quản lý học sinh THPT** 14](#_Toc502785287)

[**1.1.** **Kế thừa** 15](#_Toc502785288)

[**1.2.** **Phát triển thêm** 15](#_Toc502785289)

[**2.** **Các chức năng chính của chương trình** 15](#_Toc502785290)

[**3.** **Danh sách các yêu cầu** 17](#_Toc502785291)

[**4.** **Danh sách các biểu mẫu và quy định** 17](#_Toc502785292)

[**4.1.** **Biểu mẫu 1: Tiếp nhận học sinh** 17](#_Toc502785293)

[**4.2.** **Biểu mẫu 2: Tra cứu học sinh** 18](#_Toc502785294)

[**4.3.** **Biểu mẫu 3: Lập danh sách lớp** 18](#_Toc502785295)

[**4.4.** **Biễu mẫu 4: Quản lý giáo viên** 18](#_Toc502785296)

[**4.5.** **Biểu mẫu 5: Quản lý môn học** 18](#_Toc502785297)

[**4.6.** **Biểu mẫu 6: Xếp loại hạnh kiểm** 18](#_Toc502785298)

[**4.7.** **Biểu mẫu 7: Tiếp nhận thông tin học phí** 18](#_Toc502785299)

[**4.8.** **Biểu mẫu 8 : Quản lý BHYT** 19](#_Toc502785300)

[**4.9.** **Biểu mẫu 9 : Tiếp nhận bảng điểm** 19](#_Toc502785301)

[**4.10.** **Biểu mẫu 10: Thời khóa biểu** 19](#_Toc502785302)

[**4.11.** **Biểu mẫu 11: Báo cáo tổng kết học kỳ** 19](#_Toc502785303)

[**4.12.** **Biểu mẫu 12: Báo cáo tổng kết năm học** 20](#_Toc502785304)

[**4.13.** **Biểu mẫu 13: Báo cáo tổng kết môn học** 20](#_Toc502785305)

[**5.** **Phân tích** 20](#_Toc502785306)

[**5.1.** **DFD mức 0** 20](#_Toc502785307)

[**5.2.** **DFD mức 1** 22](#_Toc502785308)

[**5.3.** **DFD mức 2** 23](#_Toc502785309)

[**5.3.1.** **Tiếp Nhận Hồ Sơ Học Sinh** 23](#_Toc502785310)

[**5.3.2.** **Cập Nhật Hồ Sơ Học Sinh** 23](#_Toc502785311)

[**5.3.3.** **Hủy Bỏ Hồ Sơ Học Sinh** 23](#_Toc502785312)

[**5.3.4.** **Tiếp Nhận Hồ Sơ Giáo Viên** 24](#_Toc502785313)

[**5.3.5.** **Cập Nhật Hồ Sơ Giáo Viên** 24](#_Toc502785314)

[**5.3.6.** **Hủy Bỏ Hồ Sơ Giáo Viên** 25](#_Toc502785315)

[**5.3.7.** **Lập Thời Khóa Biểu Học Kỳ Năm Học** 25](#_Toc502785316)

[**5.3.8.** **Tiếp Nhận Danh Sách Xếp Loại Hạnh Kiểm** 26](#_Toc502785317)

[**5.3.9.** **Tiếp Nhận Bảng Điểm Môn Học** 27](#_Toc502785318)

[**5.3.10.** **Lập Báo Cáo Kết Quả Học Tập Cuối Kỳ** 28](#_Toc502785319)

[**5.3.11.** **Lập Danh Sách Lớp** 29](#_Toc502785320)

[**5.3.12.** **Lập Danh Sách Môn Học** 29](#_Toc502785321)

[**5.3.13.** **Xếp Lớp** 30](#_Toc502785322)

[**5.3.14.** **Tiếp Nhận Thông Tin Học Phí** 30](#_Toc502785323)

[**5.3.15.** **Tiếp Nhận Hóa Đơn Đóng Học Phí** 31](#_Toc502785324)

[**5.3.16.** **Tiếp Nhận Thông Tin Bảo Hiểm Y Tế** 32](#_Toc502785325)

[**5.3.17.** **Tiếp Nhận Hóa Đơn Đóng Bảo Hiểm Y Tế** 33](#_Toc502785326)

[**5.4.** **Sơ đồ dữ liệu quan hệ (ERD)** 34](#_Toc502785327)

[**6.** **Thiết kế** 35](#_Toc502785328)

[**6.1.** **Thiết kế hệ thống** 35](#_Toc502785329)

[**6.1.1.** **Cấu trúc chức năng phần mềm** 35](#_Toc502785330)

[**6.1.2.** **Thiết kế chức năng phần mềm** 36](#_Toc502785331)

[**6.1.2.1.** **Tiếp nhận học sinh** 36](#_Toc502785332)

[**6.1.2.2.** **Tiếp nhận thông tin học phí** 37](#_Toc502785333)

[**6.1.2.3.** **Tiếp nhận thông tin Bảo hiểm Y tế** 38](#_Toc502785334)

[**6.1.2.4.** **Tiếp nhận hóa đơn đóng Học phí** 39](#_Toc502785335)

[**6.1.2.5.** **Tiếp nhận, cập nhật, hủy bỏ hồ sơ giáo viên** 40](#_Toc502785336)

[**6.1.2.6.** **Lập danh sách giáo viên** 41](#_Toc502785337)

[**6.1.2.7.** **Xếp lớp** 42](#_Toc502785338)

[**6.1.2.8.** **Lập thời khóa biểu** 43](#_Toc502785339)

[**6.1.2.9.** **Tiếp nhận, cập nhật, hủy bỏ Bảng điểm Môn học** 44](#_Toc502785340)

[**6.1.2.10.** **Tiếp nhận, cập nhật, hủy bỏ thông tin Môn học** 45](#_Toc502785341)

[**6.1.2.11.** **Tra cứu** 46](#_Toc502785342)

[**6.2.** **Thiết kế giao diện** 47](#_Toc502785343)

[**6.2.1.** **Danh sách các màn hình** 47](#_Toc502785344)

[**6.2.2.** **Sơ đồ liên kết các màn hình** 48](#_Toc502785345)

[**7.** **Ứng dụng Excel Object Model để nhập dữ liệu từ file Excel** 49](#_Toc502785346)

[**8.** **Ứng dụng Excel Object Model để kết xuất báo cáo** 52](#_Toc502785347)

[**8.1.** **Trình tự chung khi kết xuất báo cáo** 52](#_Toc502785348)

[**8.2.** **Chi tiết** 58](#_Toc502785349)

[**8.2.1.** **Báo cáo không biểu đồ** 58](#_Toc502785350)

[**8.2.2.** **Báo cáo có biểu đồ** 58](#_Toc502785351)

[**IV.** **CÀI ĐẶT VÀ VẬN HÀNH HỆ THỐNG** 58](#_Toc502785352)

[**1.** **Hướng dẫn cài đặt** 58](#_Toc502785353)

[**2.** **Vận hành** 59](#_Toc502785354)

[**V.** **KẾT LUẬN** 61](#_Toc502785355)

[**1.** **Kết quả đạt được** 61](#_Toc502785356)

[**2.** **Những hạn chế** 61](#_Toc502785357)

[**VI.** **TÀI LIỆU THAM KHẢO** 61](#_Toc502785358)

1. **GIỚI THIỆU ĐỀ TÀI**
2. **Tổng quan về Excel Object Model**

Excel Object Model (Mô hình đối tượng Excel) là một hệ thống các lớp của ứng dụng Excel, cho phép tương tác với Excel bằng chương trình thay vì tương tác thủ công như những người sử dụng Excel thường dùng. Excel Object Model có thể được sử dụng để nhập, kẻ bảng, chèn công thức, canh chỉnh lề, đặt font chữ và size chữ, vẽ biểu đồ, …

1. **Lý do chọn đề tài**

* Cung cấp một cái nhìn ngắn gọn về Mô hình đối tượng Excel.
* Khi kết xuất các báo cáo quản lý, trong nhiều trường hợp, ta cần những báo cáo nhanh không có mẫu cố định mà trong quá trình thiết kế thường không được đề cập đến. Mô hình đối tượng Excel tỏ ra khá hiệu quả trong những trường hợp này.
* Làm việc trực tiếp với các đối tượng cấu trúc Excel không thông qua các Opensoure giúp ta nắm rõ cấu trúc của file Excel.

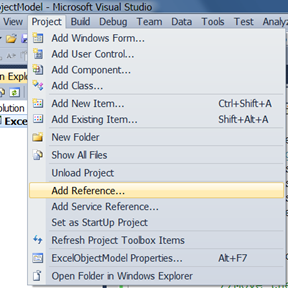
1. **Yêu cầu của đề tài**

* Tìm hiểu Excel Object Model với các lớp được thiết kế để quản lý các đối tượng trong Excel và mối liên hệ giữa chúng để tạo một file Excel như mong muốn.
* Ứng dụng Excel Object Model vào chức năng báo cáo của hệ thống phần mềm quản lý và kết xuất thông tin các báo cáo ra file Excel kết hợp với các thao tác định dạng (canh chỉnh lề, đặt font chữ, …), vẽ biểu đồ, …

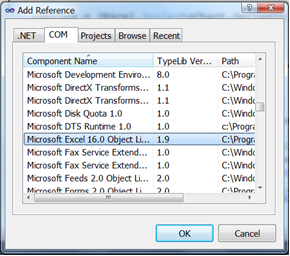
1. **TÌM HIỂU HỆ THỐNG CÁC ĐỐI TƯỢNG TRONG EXCEL OBJECT MODEL**
2. **Cách sử dụng Excel Object Model trong Visual Studio**

Để tạo một ứng dụng C# gọi ứng dụng Excel, Project cần phải tham chiếu đến thư viện COM của Excel.

* Sau khi tạo project C#, mở menu Project, chọn Add Reference:



* Chọn tab COM, tìm đến dòng Microsoft Excel x, trong đó x là số hiệu phiên bản của Excel tương ứng với bộ Office đang được cài đặt trên máy. Chọn OK.



* Thêm chỉ thị sử dụng thư viện Excel vào những class nào cần điều khiển Excel:

using Excel = Microsoft.Office.Interop.Excel;

using System.Reflection;

1. **Giới thiệu hệ thống các đối tượng chính trong Excel Object Model**

Bởi vì dữ liệu trong một tài liệu Excel có cấu trúc cao nên Mô hình đối tượng Excel có tính phân cấp và đơn giản. Excel cung cấp hàng trăm đối tượng mà bạn có thể tương tác, bạn có thể có được một khởi đầu tốt với Mô hình đối tượng Excel bằng cách tập trung vào một tập hợp con của các đối tượng có sẵn này. Các đối tượng này bao gồm 4 loại:

* Application
* Workbook
* Worksheet
* Range

Phần lớn công việc được thực hiện với Excel tập trung quanh bốn đối tượng này và các thành viên của chúng

* 1. ***Application Object***

Đối tượng Excel Application đại diện cho chính ứng dụng Excel. Đối tượng Application chứa nhiều thông tin về ứng dụng đang chạy, các tùy chọn được áp dụng cho thể hiện đó, và các đối tượng mà người dùng đang mở trong thể hiện này.

* 1. ***Workbook Object***

Đối tượng Microsoft.Office.Interop.Excel.Workbook đại diện cho một bảng tính duy nhất trong ứng dụng Excel.

Các công cụ phát triển Office trong Visual Studio mở rộng đối tượng Microsoft.Office.Interop.Excel.Workbook bằng cách cung cấp kiểu Microsoft.Office.Tools.Excel.Workbook. Kiểu này cho phép bạn truy cập vào tất cả các tính năng của một đối tượng Microsoft.Office.Interop.Excel.Workbook.

* 1. ***Worksheet Object***

Đối tượng Microsoft.Office.Interop.Excel.Worksheet là một thành viên của tập các Worksheets. Nhiều thuộc tính, phương thức và sự kiện của Microsoft.Office.Interop.Excel.Worksheet giống hoặc tương tự như các thành viên được cung cấp bởi Application hoặc các đối tượng Microsoft.Office.Interop.Excel.Workbook.

Excel cung cấp một tập các Sheets như một thuộc tính của đối tượng Microsoft.Office.Interop.Excel.Workbook. Mỗi thành viên của tập các Sheets là một Microsoft.Office.Interop.Excel.Worksheet hoặc một đối tượng Microsoft.Office.Interop.Excel.Chart.

Các công cụ phát triển Office trong Visual Studio mở rộng đối tượng Microsoft.Office.Interop.Excel.Worksheet bằng cách cung cấp kiểu Microsoft.Office.Tools.Excel.Worksheet. Kiểu này cho phép bạn truy cập vào tất cả các tính năng của đối tượng Microsoft.Office.Interop.Excel.Worksheet cũng như các tính năng mới như khả năng lưu trữ các điều khiển được quản lý và xử lý các sự kiện mới.

* 1. ***Range Object***

Đối tượng Microsoft.Office.Interop.Excel.Range là đối tượng ta sẽ sử dụng nhiều nhất trong các ứng dụng Excel. Trước khi có thể thao tác với bất kỳ khu vực nào trong Excel, ta phải thể hiện nó như một đối tượng Range và làm việc với các phương thức và thuộc tính của phạm vi đó. Một đối tượng Range đại diện cho một ô, một hàng, một cột, một sự lựa chọn các ô có chứa một hoặc nhiều khối ô, có thể có hoặc có thể không tiếp giáp, hoặc thậm chí là một nhóm các ô trên nhiều trang tính khác nhau.

1. **Làm việc với các đối tượng chính trong Excel Object Model**
   1. ***Làm việc với Workbooks***

Lớp Workbook và Workbooks chứa các phương thức và thuộc tính mà ta sử dụng để thực hiện các tác vụ với các workbooks.

|  |  |
| --- | --- |
| **Tác vụ** | **Thủ tục** |
| Tạo một workbook mới | Excel.Workbook newWorkbook = this.Application.Workbooks.Add(); |
| Mở một workbook được chỉ định. | this.Application.Workbooks.Open(@"C:\Test\YourWorkbook.xlsx"); |
| Đóng một workbook chỉ định và lưu lại những thay đổi. | Globals.ThisWorkbook.Close(false);  this.Application.ActiveWorkbook.Close(false, missing, missing);  this.Application.ActiveWorkbook.Close(true, missing, missing); |
| Lưu workbook với định dạng Excel hoặc định dạng XML. | this.Save();  this.Application.ActiveWorkbook.Save(); |
| Lấy một tập hợp các file workbook mới được sử dụng gần đây. | var collection = this.Application.RecentFiles; |
| Mở các tệp txt dưới dạng các workbook. | this.Application.Workbooks.OpenText(@"C:\Test.txt",  missing, 3,  Excel.XlTextParsingType.xlDelimited,  Excel.XlTextQualifier.xlTextQualifierNone,  missing, missing, missing, true, missing, missing, missing,  missing, missing, missing, missing, missing, missing); |

Ngoài ra, các thao tác được thực hiện với các thuộc tính, phương thức và sự kiện của Workbook hay Workbooks được trình bày rõ hơn tại: [*https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.workbook\_members.aspx*](https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.workbook_members.aspx)

* 1. ***Làm việc với Worksheets***

Các lớp Worksheet và Worksheet chứa các phương thức và thuộc tính ta sử dụng để thực hiện các tác vụ với các worksheets.

|  |  |
| --- | --- |
| **Tác vụ** | **Thủ tục** |
| Thêm một worksheet mới vào một workbook. | Excel.Worksheet newWorksheet = (Excel.Worksheet)Globals.ThisWorkbook.Worksheets.Add(); |
| Tạo một bản sao của một worksheet ở một vị trí xác định trong workbook. | Globals.Sheet1.Copy(Globals.ThisWorkbook.Sheets[3]); |
| Xóa một worksheet xác định. | Globals.Sheet1.Delete(); |
| Di chuyển lựa chọn của người dùng đến một worksheet xác định. | Globals.Sheet1.Select(); |
| Thay đổi khả năng hiển thị của một hoặc nhiều worksheet. | Globals.Sheet1.Visible = Excel.XlSheetVisibility.xlSheetHidden; |
| Khóa tất cả hoặc một phần của worksheet để không thể chỉnh sửa. | Globals.Sheet1.Protect(getPasswordFromUser, true); |
| Loại bỏ khóa khỏi một worksheet để nó có thể được chỉnh sửa. | Globals.Sheet1.Unprotect(getPasswordFromUser); |
| Thêm và xóa comment | this.dateComment.AddComment("Comment”);  this.dateComment.Comment.Delete(); |
| Hiển thị hoặc ẩn tất cả các comment | private void ShowOrHideComments(bool show)  {  for (int i = 1; i <= this.Comments.Count; i++)  {  this.Comments[i].Visible = show;  }  } |
| Tạo nhóm trong worksheet | this.data2001.Group(); |
| Làm đậm 1 hàng khi ô chứa nó được chọn | currentCell.EntireRow.Font.Bold = true; |
| Sao chép dữ liệu và định dạng trên các worksheet. | this.Application.ActiveWorkbook.Sheets.FillAcrossSheets(  this.Application.get\_Range("rangeData"),  Excel.XlFillWith.xlFillWithAll); |
| Kiểm tra chính tả trong worksheet. | Globals.Sheet1.CheckSpelling(); |

Ngoài ra, các thao tác được thực hiện với các thuộc tính, phương thức và sự kiện của Worksheet hay Worksheets được trình bày rõ hơn tại: [*https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.worksheet\_members.aspx*](https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.worksheet_members.aspx)

* 1. ***Làm việc với Ranges***

Các lớp NamedRange và Range bao gồm các phương thức và thuộc tính ta sử dụng để thực hiện các tác vụ với phạm vi range.

|  |  |
| --- | --- |
| **Tác vụ** | **Thủ tục** |
| Truy cập một phạm vi và đặt giá trị. | Excel.Range rng = this.Application.get\_Range("A1");  rng.Value2 = "Range value"; |
| Chọn một dải phạm vi dựa trên giá trị ban đầu. | Excel.Range rng = this.Application.get\_Range("B1");  rng.AutoFill(this.Application.get\_Range("B1","B5"),  Excel.XlAutoFillType.xlFillWeekdays); |
| Lấy và đặt giá trị ngày tháng trong ô. | Excel.Range rng = this.Application.get\_Range("A1");  DateTime dt = DateTime.Now;  rng.Value2 = dt; |
| Áp dụng các kiểu định dạng cho các ô trong một phạm vi. | Excel.Range rng = this.Application.get\_Range("A1");  rng.Style = "Normal"; |
| Tìm kiếm văn bản trong các phạm vi của worksheet. | Excel.Range currentFind = null;  currentFind = Fruits.Find("apples", missing, Excel.XlFindLookIn.xlValues, Excel.XlLookAt.xlPart, Excel.XlSearchOrder.xlByRows, Excel.XlSearchDirection.xlNext, false, missing, missing); |
| Áp dụng màu cho văn bản trong một dải các ô. | Excel.Range rng2 = this.Application.get\_Range("A1");  rng2.Font.Color = System.Drawing.ColorTranslator.ToOle(System.Drawing.Color.Red); |
| Chạy chương trình tính toán của Excel | Excel.Range rng = this.Application.get\_Range("A1");  rng.Calculate(); |
| Hiển thị một chuỗi trong một ô. | Excel.Range rng = Globals.Sheet1.Range["A1"];  rng.Value2 = "Hello world"; |

Ngoài ra, các thao tác được thực hiện với các thuộc tính, phương thức và sự kiện của NamedRange hay Range được trình bày rõ hơn tại: [*https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.range\_members.aspx*](https://msdn.microsoft.com/en-us/library/microsoft.office.interop.excel.range_members.aspx)

1. **HỆ THỐNG PHẦN MỀM QUẢN LÝ ỨNG DỤNG EXCEL OBJECT MODEL**
   1. **Giới thiệu phần mềm Quản lý học sinh THPT**

* Hiện nay trên thị trường dù đã có nhiều phần mềm về quản lý các công việc trong nhà trường nhưng hoặc quá phức tạp khiến phần lớn giáo viên không thể sử dụng được hoặc quá đơn giản không đáp ứng được các yêu cầu trong công tác quản lý nhà trường. Vì thế cho nên các trường học vẫn còn cộng điểm bằng tay, tiếp nhận sinh viên theo cách thủ công, thống kê kết quả học tập bằng cách đếm từng học sinh, hệ thống báo cáo đơn giản ... Cách làm này vừa không chính xác vừa lãng phí khối lượng rất lớn thời gian và công sức của giáo viên. Phần mềm quản lý Học Sinh Trung Học Phổ Thông được xây dựng nhằm đáp ứng được hai yêu cầu tiên quyết:
* Đơn giản, dễ sử dụng. Phần mềm cho phép mọi giáo viên dù khả năng về tin học chỉ ở mức biết soạn thảo văn bản cũng có thể sử dụng dễ dàng.
* Hiệu quả. Phần mềm giải quyết được một cách đầy đủ các yêu cầu thiết yếu trong công tác quản lý nhà trường.
* Đối tượng sử dụng phần mềm Nhân viên quản lý giáo vụ, Nhân viên phòng ban, Ban giám hiệu và kể cả Giáo viên.
* Phạm vi ứng dụng:
* Áp dụng cho các trường học có chương trình đào tạo theo Bộ giáo dục với quy mô vừa và nhỏ.
* Hệ thống trường tư với chương trình đào tạo chuẩn quốc gia.
* Người dùng cuối:
* Nhân viên phòng Tài vụ.
* Nhân viên phòng Giáo vụ.
* Ban Giám Hiệu.

***Đặc biệt:*** Phần mềm Quản lý học sinh THPT có ứng dụng ***Excel Object Model*** vào việc đọc dữ liệu từ file Excel và kết xuất các báo cáo tổng kết ra file Excel, định dạng các thông tin, vẽ biểu đồ cho dữ liệu trong một số báo cáo.

Phần mềm Quản lý học sinh THPT có ứng dụng ***Excel Object Model*** được phát triển dựa trên mã nguồn Đồ án môn học ***Phát triển, vận hành và bảo trì phần mềm*** của nhóm em ở học kỳ trước, được thực hiện có sự kế thừa và phát triển như sau:

* 1. **Kế thừa**
* Toàn bộ cơ sở dữ liệu cũ
* Hệ thống giao diện cũ
* Toàn bộ cấu trúc project, cấu trúc các lớp cũ
  1. **Phát triển thêm**
* Xây dựng lại hệ thống Đọc dữ liệu từ file Excel và Kết xuất báo cáo ra file Excel sử dụng ***Excel Object Model*** thay vì sử dụng ***Object Linking Embedding Database (OLE DB)*** như trước đó.
* Hoàn thiện thêm tính logic của các yêu cầu nghiệp có trong chương trình.
  1. **Các chức năng chính của chương trình**
* Quản lý Học sinh:

- Tiếp nhận, cập nhật, hủy bỏ hồ sơ học sinh.

- Tra cứu thông tin học sinh.

- Tra cứu kết quả học tập của học sinh.

- Xếp lớp.

- Quản lý học phí: tiếp nhận, cập nhật thông tin học phí, hóa đơn đóng học phí của học sinh.

- Quản lý bảo hiểm y tế: tiếp nhận, cập nhật thông tin bảo hiểm y tế, hóa đơn đóng bảo hiểm y tế của học sinh.

- Quản lý hạnh kiểm, xếp loại hạnh kiểm cho học sinh.

- Tra cúu thời khóa biểu cho học sinh.

* Quản lý Điểm:

- Tiếp nhận, cập nhật, hủy bỏ thông tin môn học.

- Tiếp nhận bảng điểm môn học.

- Các xử lý tính toán điểm.

* Quản lý Giáo viên:

- Tiếp nhận, cập nhật, hủy bỏ hồ sơ giáo viên.

- Tra cứu thông tin giáo viên.

- Tra cứu thời khóa biểu cho giáo viên.

- Danh sách các yêu cầu

* Quản lý lớp học:

- Tiếp nhận hồ sơ lớp học theo từng học kỳ - năm học, giáo viên chủ nhiệm, danh sách học sinh.

- Tra cứu kết quả học tập, danh sách học sinh thuộc lớp.

* Thời khóa biểu:

- Lập thời khóa biẻu tự động.

- Tra cứu thời khóa biểu theo: học sinh, giáo viên, lớp.

* Báo cáo, tổng kết:

- Báo cáo môn học theo học kỳ.

- Báo cáo tổng kết lớp theo học kỳ.

- Báo cáo nhân sự theo học kỳ.

- Báo cáo thông tin học phí theo học kỳ.

- Báo cáo thông tin BHYT theo học kỳ.

* Sao lưu và phục hồi hệ thống.
* Phân quyền chức năng sử dụng hệ thống
  1. **Danh sách các yêu cầu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên yêu cầu** | **Biểu mẫu** | **Quy định** | **Ghi chú** |
| 1 | Tiếp nhận học sinh | BM1 | QĐ1 |  |
| 2 | Tra cứu học sinh | BM2 |  |  |
| 3 | Lập danh sách lớp | BM3 |  |  |
| 4 | Quản lý giáo viên | BM4 |  |  |
| 5 | Quản lý môn học | BM5 |  |  |
| 6 | Xếp loại hạnh kiểm | BM6 |  |  |
| 7 | Quản lý thông tin học phí | BM7 |  |  |
| 8 | Quản lý BHYT | BM8 | QĐ2 |  |
| 9 | Tiếp nhận bảng điểm | BM9 | QĐ3 |  |
| 10 | Thời khóa biểu | BM10 |  |  |
| 11 | Báo cáo thống kê | BM11, BM12, BM13 | QĐ4 |  |
| 12 | Thay đổi quy định |  | QĐ5 |  |

* 1. **Danh sách các biểu mẫu và quy định**

1. **Biểu mẫu 1: Tiếp nhận học sinh**

|  |  |  |
| --- | --- | --- |
| **BM1:** | **Tiếp nhận học sinh** | |
| **Mã học sinh:** | | **Họ tên:** |
| **Ngày sinh:** | | **Giới tính:** |
| **Địa chỉ:** | | **Email:** |
| **Ảnh đại diện:** | | **Ngày nhập học:** |
| **Số điện thoại phụ huynh:** | | **Ghi chú:** |
| **Trạng thái theo học:** | |

Quy định 1:

|  |
| --- |
| **QĐ1: Tuổi học sinh từ 15-20 tuối.** |

1. **Biểu mẫu 2: Tra cứu học sinh**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BM2:** | **Danh sách học sinh** | | | | | | | |
| **STT** | **Mã học sinh** | **Họ tên** | **Ngày sinh** | **Giới tính** | **Địa chỉ** | **Số điện thoại** | **Ghi chú** | **Trung bình** |
| **1** |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |  |

1. **Biểu mẫu 3: Lập danh sách lớp**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BM3:** | | | **Danh sách lớp** | |
| **Mã lớp:** | | **Sĩ số:** | | **Học kì năm học:** |
| **STT** | **Họ tên** | **Ngày sinh** | **Giới tính** | **Địa chỉ** |
| **1** |  |  |  |  |
| **2** |  |  |  |  |

1. **Biễu mẫu 4: Quản lý giáo viên**

|  |  |
| --- | --- |
| **BM4:** | **Quản lý giáo viên** |
| **Mã giáo viên:** | **Họ tên:** |
| **Ngày sinh:** | **Địa chỉ:** |
| **Email** | **Số điện thoại** |
| **Ảnh đại diện** | **Giới tính** |
| **Mã chuyên môn** | **Ngày vào làm** |
| **Ghi chú** | **Tình trạng công tác** |

1. **Biểu mẫu 5: Quản lý môn học**

|  |  |  |  |
| --- | --- | --- | --- |
| **BM5:** | | **Quản lý môn học** | |
| **STT** | **Mã môn học** | **Tên môn học** | **Hệ số** |
| **1** |  |  |  |
| **2** |  |  |  |

1. **Biểu mẫu 6: Xếp loại hạnh kiểm**

|  |  |  |
| --- | --- | --- |
| **BM6:** | **Xếp loại hạnh kiểm** | |
| **Học kỳ :** | | |
| **STT** | **Tên học sinh** | **Xếp loại** |
| **1** |  |  |
| **2** |  |  |

1. **Biểu mẫu 7: Tiếp nhận thông tin học phí**

|  |  |
| --- | --- |
| **BM7:** | **Tiếp nhận thông tin học phí** |
| **Mã học sinh:** | **Học kỳ năm học:** |
| **Tổng học phí:** | **Học phí đã đóng:** |
| **Số dư:** | **Ngày đóng:** |

1. **Biểu mẫu 8 : Quản lý BHYT**

|  |  |
| --- | --- |
| **BM8:** | **Quản lý BHYT** |
| **Mã học sinh:** | **Học kỳ năm học** |
| **Phí BHYT:** | **Trạng thái:** |
| **Ngày đóng:** | **Số dư:** |
| **Số tiền đóng:** |  |

Quy định 2:

|  |
| --- |
| **QĐ2: Phí BHYT phải theo quy định của nhà nước.** |

1. **Biểu mẫu 9 : Tiếp nhận bảng điểm**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BM9:** | | **Tiếp nhận bảng điểm** | | | | | |
| **Mã học sinh:** | | | | | **Tên học sinh:** | | |
| **Mã lớp:** | | | | | **Mã môn học:** | | |
| **Học kỳ năm học:** | | | | |  | | |
| **STT** | **Họ tên** | | **Điểm miệng** | **Điểm 15 phút** | **Điểm 45 phút** | **Điểm học kỳ** | **Điểm trung bình** |
| **1** |  | |  |  |  |  |  |
| **2** |  | |  |  |  |  |  |

Quy định 3:

|  |
| --- |
| **QĐ3: Điểm của học sinh : 0 <= Điểm <= 10.** |

1. **Biểu mẫu 10: Thời khóa biểu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BM10:** | | | **Thời khóa biểu** | |
| **Lớp :** | | | **Học kỳ :** | |
| **STT** | **Thứ** | **Tiết** | | **Môn học** |
| **1** |  |  | |  |
| **2** |  |  | |  |

Quy định 4:

|  |
| --- |
| **QĐ4 : Chủ nhật học sinh được nghỉ.** |

1. **Biểu mẫu 11: Báo cáo tổng kết học kỳ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BM11:** | | **Báo cáo tổng kết học kỳ** | | | |
| **Học kỳ:** | | | | | |
| **STT** | **Lớp** | | **Sĩ số** | **Số lượng đạt** | **Tỉ lệ** |
| **1** |  | |  |  |  |
| **2** |  | |  |  |  |

Quy định 5:

|  |
| --- |
| **QĐ5: Học sinh đạt có điềm >=5.0 .** |

1. **Biểu mẫu 12: Báo cáo tổng kết năm học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BM12:** | | **Báo cáo tổng kết năm học** | | |
| **Năm học:** | | | | |
| **STT** | **Lớp** | **Sĩ số** | **Số lượng đạt** | **Tỉ lệ** |
| **1** |  |  |  |  |
| **2** |  |  |  |  |

1. **Biểu mẫu 13: Báo cáo tổng kết môn học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BM13:** | | **Báo cáo tổng kết môn học** | | |
| **Môn học:** | | **Học kỳ:** | | |
| **STT** | **Lớp** | **Sĩ số** | **Số lượng đạt** | **Tỉ lệ** |
| **1** |  |  |  |  |
| **2** |  |  |  |  |

* 1. **Phân tích**

1. **DFD mức 0**



1. **DFD mức 1**



1. **DFD mức 2**
2. **Tiếp Nhận Hồ Sơ Học Sinh**



1. **Cập Nhật Hồ Sơ Học Sinh**



1. **Hủy Bỏ Hồ Sơ Học Sinh**



1. **Tiếp Nhận Hồ Sơ Giáo Viên**



1. **Cập Nhật Hồ Sơ Giáo Viên**



1. **Hủy Bỏ Hồ Sơ Giáo Viên**



1. **Lập Thời Khóa Biểu Học Kỳ Năm Học**



1. **Tiếp Nhận Danh Sách Xếp Loại Hạnh Kiểm**



1. **Tiếp Nhận Bảng Điểm Môn Học**



1. **Lập Báo Cáo Kết Quả Học Tập Cuối Kỳ**



1. **Lập Danh Sách Lớp**



1. **Lập Danh Sách Môn Học**



1. **Xếp Lớp**



1. **Tiếp Nhận Thông Tin Học Phí**



1. **Tiếp Nhận Hóa Đơn Đóng Học Phí**



1. **Tiếp Nhận Thông Tin Bảo Hiểm Y Tế**



1. **Tiếp Nhận Hóa Đơn Đóng Bảo Hiểm Y Tế**



1. **Sơ đồ dữ liệu quan hệ (ERD)**



* 1. **Thiết kế**

1. **Thiết kế hệ thống**
2. **Cấu trúc chức năng phần mềm**



1. **Thiết kế chức năng phần mềm**
2. **Tiếp nhận học sinh**



1. **Tiếp nhận thông tin học phí**



1. **Tiếp nhận thông tin Bảo hiểm Y tế**



1. **Tiếp nhận hóa đơn đóng Học phí**



1. **Tiếp nhận, cập nhật, hủy bỏ hồ sơ giáo viên**



1. **Lập danh sách giáo viên**



1. **Xếp lớp**



1. **Lập thời khóa biểu**



1. **Tiếp nhận, cập nhật, hủy bỏ Bảng điểm Môn học**



1. **Tiếp nhận, cập nhật, hủy bỏ thông tin Môn học**



1. **Tra cứu**



1. **Thiết kế giao diện**
2. **Danh sách các màn hình**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Mô tả chức năng** |
| 1 | Đăng nhập | Đăng nhập vào hệ thống |
| 2 | Tiếp nhận học sinh | Thực hiện chức năng Nhập, xóa, sửa học sinh vào danh sách |
| 3 | Tra cứu học sinh | Tra cứu học sinh trong danh sách đã có |
| 4 | Tiếp nhận học phí | Thực hiện chức năng thêm và cập nhật học sinh đóng học phí |
| 5 | Xếp loại hạnh kiểm | Xếp hạnh kiểm cho học sinh |
| 6 | Xếp lớp | Tiến hành xếp lớp hoặc chuyển lớp cho học sinh |
| 7 | Tiếp nhận bảo hiểm | Tiếp nhận bảo hiểm của học sinh |
| 8 | Tiếp nhận hóa đơn đóng bảo hiểm | Tiếp nhận hóa đơn của học sinh đóng bảo hiểm |
| 9 | Tiếp nhận hóa đơn đóng học phí | Tiếp nhận hóa đơn của học sinh đóng học phí |
| 10 | Tiếp nhận giáo viên | Thực hiện chức năng Nhập, xóa, sửa giáo viên vào danh sách |
| 11 | Tra cứu thông tin giáo viên | Tra cứu giáo viên trong danh sách đã có |
| 12 | Tiếp nhận bảng điểm môn học | Tiến hành nhập hoặc cập nhật bảng điểm của học sinh |
| 13 | Lập danh sách giáo viên | Lập ra danh sách chứa thông tin các giáo viên và tình trạng công tác |
| 14 | Lập thời khóa biểu | Lập thời khóa biểu |
| 15 | Tiếp nhận lớp học | Tiếp hành nhận thời khóa biểu cho giáo viên |
| 16 | Tra cứu thời khóa biểu giáo viên | Tiến hành in và xuất file TKB của lớp giáo viên |
| 17 | Tra cứu thời khóa biểu lớp | Tiến hành in và xuất file TKB của lớp học |
| 18 | Xuất ra file Excel | Tiến hành xuất các thông tin ra file Excel |
| 19 | Tùy chọn in | Các tùy chọn cài đặt để in |
| 20 | Báo cáo nhân sự | Lập bảng báo cáo về nhân sự |
| 21 | Báo cáo tổng kết học kỳ | Lập bảng báo cáo về tổng kết năm học |
| 22 | Báo cáo tổng kết học kỳ chi tiết theo lớp | Lập bảng báo cáo về tổng kết năm học chi tiết hơn theo từng lớp |
| 23 | Báo cáo tổng kết môn học | Lập bảng báo cáo về tổng kết môn học |

1. **Sơ đồ liên kết các màn hình**



* 1. **Ứng dụng Excel Object Model để nhập dữ liệu từ file Excel**
* Trong thư mục **DuLieuMau** có chứa các file Excel để người sử dụng có thể nhập dữ liệu số lượng lớn một cách tự động vào hệ thống.
* Với mỗi loại dữ liệu kiểu danh sách tương ứng với chức năng mà hệ thống cần sẽ có 1 file Excel kèm theo.
* Tên các file Excel không cần phải đặt theo một định dạng nhất định nhưng tên Sheet và cấu trúc dữ liệu trong từng Sheet phải theo định dạng mẫu.
* Cấu trúc dữ liệu mẫu trong các file Excel là do nhóm tự liên hệ với thực tế đã gặp và yêu cầu dữ liệu của chương trình để xây dựng. Không tuân theo một mẫu có sẵn nào.
* Dữ liệu đọc từ file Excel sẽ qua các hàm xử lý trung gian ở các tầng trên của hệ thống trước khi lưu xuống CSDL SQL Server.
* Khi người dùng sử dụng chức năng **Nhập Liệu Từ File**, hệ thống sẽ gọi tới hàm: (DataSet) DAL.ExcelFile.readExcelFile(filePath). Hàm này sẽ thực hiện việc lấy thông tin từ tất cả các Sheet có trong file Excel đã chọn tại địa chỉ filePath. Thông tin và cấu trúc của mỗi Sheet sẽ được lưu vào 1 DataTable. Một DataSet sẽ lưu trữ dữ liệu của tất cả các DataTable này và trả về cho hệ thống thông qua hàm ở trên để xử lý những bước tiếp theo.
* Chi tiết hàm DAL.ExcelFile.readExcelFile(filePath) như sau:

public static DataSet readExcelFile(string filePath)

{

//Khai báo các đối tượng Excel Object Model

Excel.Application xlApplication = null;

Excel.Workbook xlWorkbook = null;

Excel.Range xlRange = null;

try

{

//Khởi tạo đối tượng trả về của hàm

DataSet ds = new DataSet();

xlApplication = new Excel.Application();

xlApplication.Visible = false;

xlApplication.ScreenUpdating = false;

xlApplication.DisplayAlerts = false;

//Mở file Excel tại địa chỉ @filePath để đọc dữ liệu

xlWorkbook = xlApplication.Workbooks.Open(@filePath,

Missing.Value, Missing.Value, Missing.Value,

Missing.Value, Missing.Value, Missing.Value,

Missing.Value, Missing.Value, Missing.Value,

Missing.Value, Missing.Value, Missing.Value,

Missing.Value, Missing.Value);

//Nếu số lượng Sheet trong file Excel khác 2 sẽ báo lỗi

if (xlWorkbook.Worksheets.Count != 2)

throw new Exception("Cấu trúc dữ liệu của file Excel không đúng định dạng!");

//Khai báo biến số lượng cột và dòng của sheet

int iTotalColumns, iTotalRows;

//Thực hiện vòng lặp duyệt qua từng sheet trong file Excel

foreach (Excel.Worksheet xlWorksheet in xlWorkbook.Worksheets)

{

//Khởi tạo đối tượng DataTable để lưu trữ dữ liệu của sheet

DataTable dt = new DataTable();

//Lấy tên của sheet làm tên cho DataTable

dt.TableName = xlWorksheet.Name;

//Xóa tất cả các định dạng có sẵn trên các cột và dòng của sheet

//để việc lấy số cột và dòng không lỗi

xlWorksheet.Columns.ClearFormats();

xlWorksheet.Rows.ClearFormats();

//Lấy số cột và dòng của sheet

iTotalColumns = xlWorksheet.UsedRange.Columns.Count;

iTotalRows = xlWorksheet.UsedRange.Rows.Count;

//Tạo một vùng chọn là dòng đầu tiên của sheet

xlRange = xlWorksheet.get\_Range("A1", letters[iTotalColumns - 1] +

iTotalRows.ToString());

//Lấy tên các cột của sheet lưu vào DataTable

for (int col = 1; col <= iTotalColumns; col++)

{

if (xlRange.Cells[1, col].Value2 != null)

dt.Columns.Add(xlRange.Cells[1, col].Value2);

else

dt.Columns.Add(col.ToString());

}

//Lấy dữ liệu từ tất cả các dòng còn lại của sheet và lưu vào DataTable

for (int row = 2; row <= iTotalRows; row++)

{

DataRow dr = dt.NewRow();

for (int col = 1; col <= iTotalColumns; col++)

{

dr[dt.Columns[col - 1].ColumnName] =

xlRange.Cells[row, col].Value2;

}

dt.Rows.Add(dr);

}

//Thêm DataTable vào DataSet

ds.Tables.Add(dt);

}

//Giải phóng các đối tượng Excel Object Model

if (xlRange != null)

Marshal.ReleaseComObject(xlRange);

if (xlWorkbook != null)

{

xlWorkbook.Close(false, Missing.Value, Missing.Value);

Marshal.ReleaseComObject(xlWorkbook);

}

if (xlApplication != null)

{

xlApplication.Quit();

Marshal.ReleaseComObject(xlApplication);

}

xlRange = null;

xlWorkbook = null;

xlApplication = null;

return ds;

}

catch (Exception ex)

{

if (xlRange != null)

Marshal.ReleaseComObject(xlRange);

if (xlWorkbook != null)

{

xlWorkbook.Close(false, Missing.Value, Missing.Value);

Marshal.ReleaseComObject(xlWorkbook);

}

if (xlApplication != null)

{

xlApplication.Quit();

Marshal.ReleaseComObject(xlApplication);

}

xlRange = null;

xlWorkbook = null;

xlApplication = null;

throw ex;

}

}

* 1. **Ứng dụng Excel Object Model để kết xuất báo cáo**

1. **Trình tự chung khi kết xuất báo cáo**

* Khi thực hiện kết xuất báo cáo, các thông tin: Tên file, Người lập là do người dùng nhập vào. Các thông tin: Năm học, Học kỳ, Lớp, Môn học là các thông tin tùy chọn. Người dùng sẽ tự thêm vào các thông tin này (bằng cách check vào các CheckBox tương ứng trên giao diện kết xuất) sao cho phù hợp với từng loại báo cáo cần kết xuất.
* Cấu trúc dữ liệu trong mỗi loại báo cáo kết xuất là do nhóm tự liên hệ với thực tế đã gặp và yêu cầu chương trình để xây dựng. Không tuân theo một mẫu có sẵn nào.
* Mỗi báo cáo kết xuất sẽ được lưu lại thành 1 file Excel với dữ liệu tương ứng. **Báo Cáo Tổng Kết Giáo Viên** sẽ được lưu thành 1 file Excel với 1 sheet dữ liệu. Các báo cáo còn lại: **Báo Cáo Tổng Kết Lớp**, **Báo Cáo Tổng Kết Môn Học**, **Báo Cáo Thông Tin Học Phí** và **Báo Cáo Thông Tin BHYT** sẽ được lưu thành 1 file Excel với 2 sheet (1 sheet ***Thông tin chi tiết*** chứa chi tiết thông tin cần kết xuất, 1 sheet ***Biểu đồ*** gồm dữ liệu biểu đồ kèm với biểu đồ hình tròn thể hiện cho dữ liệu đó).
* Sheet ***Thông tin chi tiết*** được kết xuất trên cơ sở dữ liệu lấy từ SQL Server đã qua xử lý trung gian tùy theo từng loại báo cáo. Sheet ***Biểu đồ*** được kết xuất từ 1 dữ liệu cụ thể theo từng loại báo cáo (tỉ lệ học sinh đạt (chưa đạt), tỉ lệ học sinh đã hoàn thành (chưa hoàn thành) bảo hiểm hay học phí.
* Khi người dùng thực hiện việc kết xuất báo cáo bất kỳ với các thông tin như trên, hệ thống sẽ gọi đến hàm kết xuất dữ liệu ra file Excel:

public static bool writeExcelFile(string filePath, string reportTitle, DataTable infoTable, DataTable detailTable, DataTable tableChart, string chartTitle)

{

//Khai báo cáo đối tượng Excel Object Model

Excel.Application xlApplication = null;

Excel.Workbook xlWorkbook = null;

Excel.Worksheet xlWorksheet1 = null;

Excel.Worksheet xlWorksheet2 = null;

Excel.Range xlRange = null;

try

{

xlApplication = new Excel.Application();

xlApplication.DisplayAlerts = false;

//Tạo 1 workbook mới

xlWorkbook = xlApplication.Workbooks.Add();

//Lấy dữ liệu từ DataTable tableChart để vẽ biểu đồ

//ở sheet thứ 1

#region Vẽ biểu đồ

if (tableChart != null)

{

xlWorksheet2 = xlWorkbook.ActiveSheet;

xlWorksheet2.Name = "Biểu đồ";

//Tạo 1 vùng chọn là tất cả các ô trong sheet

//và định dạng dữ liệu cho vùng chọn đó

xlRange = xlWorksheet2.get\_Range("A1").EntireRow.EntireColumn;

xlRange.Font.Name = "Times New Roman";

xlRange.RowHeight = 20;

xlRange.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Lấy số cột và dòng của tableChart

int col = tableChart.Columns.Count;

int row = tableChart.Rows.Count;

//Tạo 1 vùng chọn là dòng đầu tiên

//để xuất thông tin các cột của tableChart

xlRange = xlWorksheet2.get\_Range("A1", letters[col - 1] + "1");

xlRange.Font.Bold = true;

xlRange.RowHeight = 30;

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Xuất thông tin các cột

for (int i = 0; i < col; i++)

{

if (i == 0)

continue;

xlRange.Cells[1, i + 1] = tableChart.Columns[i].ColumnName;

}

//Xuất thông tin các dòng

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

xlWorksheet2.Cells[i + 2, j + 1] = tableChart.Rows[i][j];

}

}

//Tạo 1 vùng chọn bao quanh tiêu đề của bảng

//để định dạng cho nó

xlRange = xlWorksheet2.get\_Range("A1", "A" + (row + 1).ToString());

xlRange.Font.Bold = true;

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Tạo 1 vùng chọn bao quanh bảng để căn chỉnh dữ liệu các cột

//cho vừa khít

xlRange = xlWorksheet2.get\_Range("A1", letters[col - 1] + (row + 1).ToString());

xlRange.Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

xlRange.EntireColumn.AutoFit();

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

for (int i = 1; i <= xlRange.Columns.Count; i++)

{

xlWorksheet2.Columns[i].ColumnWidth += 5;

}

//Vẽ biểu đồ với dữ liệu vùng chọn (A1, C2)

Excel.ChartObjects xlCharts = (Excel.ChartObjects)xlWorksheet2.ChartObjects(Type.Missing);

Excel.ChartObject myChart = (Excel.ChartObject)xlCharts.Add(20, 90, 300, 250);

Excel.Chart chartPage = myChart.Chart;

Excel.Range chartRange = xlWorksheet2.get\_Range("A1", "C2");

chartPage.SetSourceData(chartRange, Missing.Value);

chartPage.ChartType = Excel.XlChartType.xl3DPie;

chartPage.ChartWizard(Source: chartRange, Title: chartTitle + " (%)");

}

#endregion

//Xuất thông tin báo cáo chi tiết ở sheet thứ 2

#region Xuất thông tin báo cáo

if (tableChart != null)

xlWorksheet1 = xlApplication.Worksheets.Add();

else

xlWorksheet1 = xlWorkbook.ActiveSheet;

xlWorksheet1.Name = "Thông tin chi tiết";

//Tạo 1 vùng chọn là tất cả các ô trong sheet

//và định dạng cho nó

xlRange = xlWorksheet1.get\_Range("A1").EntireRow.EntireColumn;

xlRange.Font.Name = "Times New Roman";

xlRange.RowHeight = 20;

xlRange.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Lấy số cột của detailTable

int numberColumnDetailTable = detailTable.Columns.Count;

//Tạo 1 vùng chọn để xuất tiêu đề của bảng

xlRange = xlWorksheet1.get\_Range("A1", letters[numberColumnDetailTable - 1] + "1");

xlRange.Merge(false);

xlRange.Cells[1, 1] = reportTitle;

xlRange.RowHeight = 40;

xlRange.Font.Bold = true;

xlRange.Font.Size = 20;

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

int numberColumnInformationTable = infoTable.Columns.Count;

int h = 2, k = 2;

//Tạo 1 vùng chọn để xuất thông tin tùy chọn từ infoTable

xlRange = xlWorksheet1.get\_Range("A2", "D4");

xlRange.Font.Bold = true;

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

for (int i = 0; i < numberColumnInformationTable; i++)

{

if (i >= 3)

{

xlWorksheet1.Cells[k, 3] = infoTable.Columns[i].ColumnName + ":";

xlWorksheet1.Cells[k, 4] = infoTable.Rows[0][i];

k++;

}

else

{

xlWorksheet1.Cells[h, 1] = infoTable.Columns[i].ColumnName + ":";

xlWorksheet1.Cells[h, 2] = infoTable.Rows[0][i];

h++;

}

}

//Tạo 1 vùng chọn để xuất tên các cột của detailTable

xlRange = xlWorksheet1.get\_Range("A6", letters[numberColumnDetailTable - 1] + "6");

xlRange.Font.Bold = true;

xlRange.RowHeight = 30;

xlRange.Interior.Color = System.Drawing.Color.FromArgb(204, 255, 255);

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

for (int i = 0; i < numberColumnDetailTable; i++)

{

if (i == 0)

{

xlRange.Cells[1, i + 1] = "STT";

}

else

{

xlRange.Cells[1, i + 1] = detailTable.Columns[i].ColumnName;

}

}

//Xuất thông tin các dòng của detailTable

int numberRowDetailTable = detailTable.Rows.Count;

String[,] data = new String[numberRowDetailTable, numberColumnDetailTable];

for (int r = 0; r < numberRowDetailTable; r++)

{

for (int c = 0; c < numberColumnDetailTable; c++)

{

if (c == 0)

{

data[r, c] = (r + 1).ToString();

}

else

{

data[r, c] = detailTable.Rows[r][c].ToString();

}

}

}

String newPosition = (6 + detailTable.Rows.Count).ToString();

xlWorksheet1.get\_Range("A7", letters[numberColumnDetailTable - 1] + newPosition).Value2 = data;

//Tạo 1 vùng chọn bao quanh cả bảng để định dạng

xlRange = xlWorksheet1.get\_Range("A6", letters[numberColumnDetailTable - 1] + newPosition);

xlRange.Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

xlRange.EntireColumn.AutoFit();

xlRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

for (int i = 2; i <= xlRange.Columns.Count; i++)

{

xlWorksheet1.Columns[i].ColumnWidth += 10;

}

#endregion

//Kết xuất ra file Excel với định dạng xlExcel8

xlWorkbook.SaveAs(filePath, FileFormat: Excel.XlFileFormat.xlExcel8, AccessMode: Excel.XlSaveAsAccessMode.xlExclusive,

ConflictResolution: Excel.XlSaveConflictResolution.xlLocalSessionChanges);

//Giải phóng các đối tượng Excel Object Model

if (xlRange != null)

Marshal.ReleaseComObject(xlRange);

if (xlWorksheet1 != null)

Marshal.ReleaseComObject(xlWorksheet1);

if (xlWorksheet2 != null)

Marshal.ReleaseComObject(xlWorksheet2);

if (xlWorkbook != null)

{

xlWorkbook.Close(true, Missing.Value, Missing.Value);

Marshal.ReleaseComObject(xlWorkbook);

}

if (xlApplication != null)

{

xlApplication.Quit();

Marshal.ReleaseComObject(xlApplication);

}

xlRange = null;

xlWorksheet1 = null;

xlWorksheet2 = null;

xlWorkbook = null;

xlApplication = null;

return true;

}

catch (Exception err)

{

if (xlRange != null)

Marshal.ReleaseComObject(xlRange);

if (xlWorksheet1 != null)

Marshal.ReleaseComObject(xlWorksheet1);

if (xlWorksheet2 != null)

Marshal.ReleaseComObject(xlWorksheet2);

if (xlWorkbook != null)

{

xlWorkbook.Close(true, Missing.Value, Missing.Value);

Marshal.ReleaseComObject(xlWorkbook);

}

if (xlApplication != null)

{

xlApplication.Quit();

Marshal.ReleaseComObject(xlApplication);

}

xlRange = null;

xlWorksheet1 = null;

xlWorksheet2 = null;

xlWorkbook = null;

xlApplication = null;

throw err;

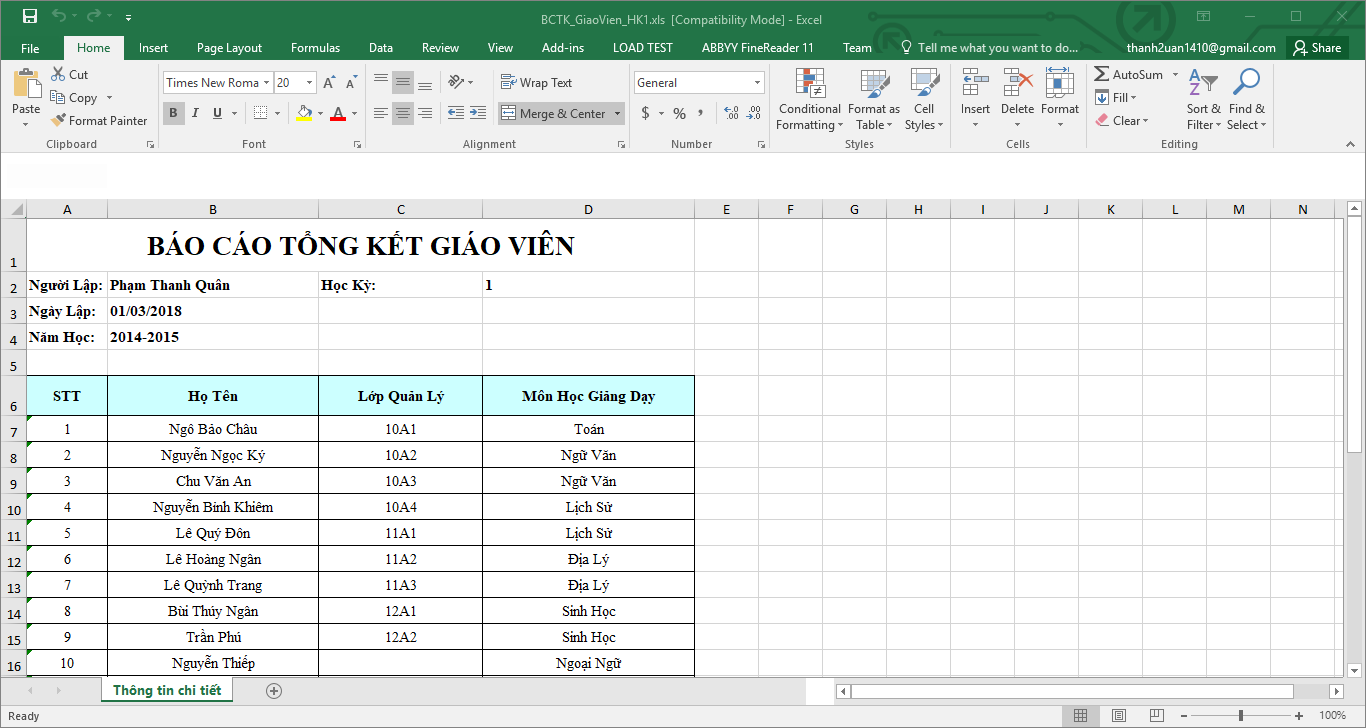
}

}

|  |  |  |
| --- | --- | --- |
| **Tên tham số** | **Kiểu dữ liệu** | **Mô tả** |
| filePath | string | Chứa đường dẫn vị trí kết xuất báo cáo |
| reportTitle | string | Tiêu đề của báo cáo, nằm ở sheet ***Thông tin chi tiết*** |
| infoTable | DataTable | Là bảng chứa các thông tin tùy chọn của báo cáo mà người dùng chọn từ giao diện kết xuất |
| detailTable | DataTable | Là bảng chứa thông tin chi tiết sẽ kết xuất, nằm ở sheet ***Thông tin chi tiết*** |
| tableChart | DataTable | Là bảng chứa dữ liệu để vẽ biểu đồ, nằm ở sheet ***Biểu đồ*** |
| chartTitle | string | Tiêu đề của biểu đồ, nằm ở sheet ***Biểu đồ*** |

1. **Chi tiết**
2. **Báo cáo không biểu đồ**

* Báo cáo không biểu đồ gồm có: **Báo Cáo Tổng Kết Giáo Viên**
* Khi thực hiện kết xuất, từ màn hình Báo cáo tương ứng chương trình sẽ gọi một hàm: XuatRaThietBi OutpuFileExcel(DataTable tb1, DataTable tb2, string reportTitle, string chartTitle = null) và truyền vào các tham số lần lượt là: XuatRaThietBi.OutpuFileExcel(data, null, "BÁO CÁO TỔNG KẾT GIÁO VIÊN"). Đây là hàm khởi tạo của một form trung gian để thêm bớt các tùy chọn khi kết xuất báo cáo
* Với hàm này chỉ truyền vào giá trị DataTable data chứa dữ liệu chi tiết để kết xuất báo cáo, vì báo cáo không có dữ liệu để vẽ biểu đồ nên tham số thứ 2 ta truyền vào giá trị null, tham số thứ 3 sẽ là tiêu đề của báo cáo, tham số thứ 4 mặc định là null vì ta không có biểu đồ
* Từ form trung gian này, chương trình sẽ gọi tới hàm DAL.ExcelFile.writeExcelFile() cùng với các tham số tương ứng như ở phần 1 để thực hiện việc kết xuất báo cáo



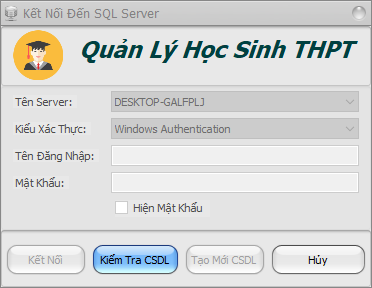
1. **Báo cáo có biểu đồ**

* Báo cáo có biểu đồ gồm có: **Báo Cáo Tổng Kết Lớp, Báo Cáo Tổng Kết Môn Học, Báo Cáo Thông Tin Học Phí và Báo Cáo Thông Tin BHYT**
* Cũng tương tự như Báo cáo không có biểu đồ, các báo cáo có biểu đồ sẽ truyền vào hàm khởi tạo của form trung gian nhưng với đầy đủ các giá trị, không khuyết cũng không null
* Dữ liệu của biểu đồ sẽ được tính toán trước khi gửi sang form trung gian

1. **CÀI ĐẶT VÀ VẬN HÀNH HỆ THỐNG**
2. **Hướng dẫn cài đặt**

Tiến hành cài đặt chương trình theo trình tự sau:

* Cài đặt môi trường: Windows OS (7, 8, 8.1, 10).
* Cài đặt hệ quản trị cơ sở dữ liệu: SQL Server 2012 hoặc cao hơn.
* Cài đặt công cụ: Visual studio 2013 hoặc cao hơn, SQL Server Management 2012 hoặc cao hơn, Dev Express 14.2.
* Cài đặt Framework: .Net framework 4.5 hoặc cao hơn.
* Tiến hành chạy project để khởi tạo cơ sở dữ liệu cho hệ thống



* Tiến hành thêm dữ liệu mẫu về Lớp, Học Kỳ, Học Kỳ - Năm Học từ file *QUANLYHOCSINH\_DuLieuMau.sql* trong thư mục **DuLieuMau\File\_SQLScript**bằng cách thực thi nó trongSQL Server Management.
* Chạy project và bắt đầu thực hiện các quy trình nghiệp vụ của phần mềm
* **Chú ý:** Chương trình đã được phân quyền sử dụng với 3 loại người dùng như sau

|  |  |  |
| --- | --- | --- |
| **Tên đăng nhập** | **Mật khẩu** | **Quyền** |
| admin | quantrihethong | Quản trị toàn bộ hệ thống |
| giaovu | giaovu | Sử dụng các chức năng của phòng Giáo vụ |
| taivu | taivu | Sử dụng các chức năng của phòng Tài vụ |

1. **Vận hành**

Sau khi đã thực hiện thành công các bước ở **Hướng dẫn cài đặt**, ta cần thực hiện thêm dữ liệu cho chương trình theo đúng quy trình sau:

* Đăng nhập vào tài khoản ***giaovu*** để nhập những dữ liệu liên quan dến công tác **Giáo vụ**
* Chọn chức năng **Thêm Học Sinh** ở tab **Hồ Sơ** để Tiếp nhận hồ sơ học sinh mới (Thêm dữ liệu mẫu từ file *DanhSachHocSinh.xlsx* hoặc nhập thủ công)
* Chọn chức năng **Thêm Giáo Viên** ở tab **Hồ Sơ** để Tiếp nhận hồ sơ giáo viên mới (Thêm dữ liệu mẫu từ file *DanhSachGiaoVien.xlsx* hoặc nhập thủ công)
* Chọn chức năng **Lập Danh Sách Giáo Viên** ở tab **Giáo Vụ** để thêm giáo viên trong danh sách mới tiếp nhận vào Học Kỳ - Năm Học được phép giảng dạy, kèm với Môn học mà giáo viên đó sẽ giảng dạy
* Chọn chức năng **Lập Danh Sách Lớp** ở tab **Giáo Vụ** đề phân công giáo viên chủ nhiệm cho các lớp
* Chọn chức năng **Xếp Lớp** ở tab **Giáo Vụ** để xếp lớp cho những học sinh đã tiếp nhận (Xếp lớp theo dữ liệu mẫu từ file *GUI\_XepLop\_12A1\_HK1.xlsx* hoặc xếp thủ công)
* Chọn chức năng **Xếp Hạnh Kiểm** ở tab **Giáo Vụ** để xếp loại hạnh kiểm cho học sinh trong từng lớp (Xếp loại hạnh kiểm theo dữ liệu mẫu từ file *GUI\_XepLoaiHanhKiem\_12A1\_HK1.xlsx* hoặc xếp thủ công)
* Chọn chức năng **Bảng Điểm Môn Học** ở tab **Giáo Vụ** để Tiếp nhận bảng điểm theo môn học cho từng học sinh (Nhập điểm theo dữ liệu mẫu từ file *GUI\_TiepNhanBangDiemMonHoc\_12A1\_HK1\_Toan.xlsx* hoặc nhập thủ công)
* Đăng nhập vào tài khoản ***taivu*** để nhập những thông tin liên quan đến công tác **Tài Vụ.**
* Chọn chức năng **Thông Tin Học Phí** ở tab **Tài Vụ** để Tiếp nhận thông tin học phí cho từng học sinh (Tiếp nhận thông tin học phí theo dữ liệu mẫu từ file *GUI\_HocPhi\_HK1.xlsx* hoặc tiếp nhận thủ công)
* Chọn chức năng **Thông Tin Bảo Hiểm Y Tế** ở tab **Tài Vụ** để Tiếp nhận thông tin bảo hiểm cho từng học sinh (Tiếp nhận thông tin bảo hiểm theo dữ liệu mẫu từ file *GUI\_BaoHiem\_HK1.xlsx* hoặc tiếp nhận thủ công)
* Đăng nhập vào tài khoản ***admin*** để thực hiện các chức năng quản trị hệ thống: **Thêm/Cập Nhật/Xóa Người Dùng**, **Tạo Sao Lưu** hay **Khôi Phục CSDL Từ Bản Sao Lưu**, **Xem Nhật Ký Hoạt Động**.
* Như vậy, làm theo trình tự trên sẽ giúp ta thêm những dữ liệu cần thiết để sử dụng toàn bộ các chức năng của hệ thống phần mềm.
* **Lưu ý:** Chức năng liên quan đến **Thời Khóa Biểu** còn đang trong quá trình hoàn thiện.

1. **KẾT LUẬN**
2. **Kết quả đạt được**
   * + - Hoàn thành được toàn bộ các yêu cầu đặt ra ban đầu khi bắt đầu phát triển phần mềm.
       - Phần mềm đáp ứng được tương đối đầy đủ các yêu cầu nghiệp vụ quản lý học sinh trong các trường THPT hiện nay.
       - Phần mềm thân thiện, dễ sử dụng.
       - Học thêm được nhiều kiến thức bổ ích về lập trình C# với Dev Express
       - Đã ứng dụng thành công **Excel Object Model** vào việc đọc dữ liệu, kết xuất báo cáo ra file Excel, định dạng file và vẽ biểu đồ cho thông tin của báo cáo.
3. **Những hạn chế**
   * + - Thời gian còn hạn chế nên việc kiểm tra để phát hiện hết tất cả các lỗi của phần mềm còn chưa thực hiện được.
4. **TÀI LIỆU THAM KHẢO**
5. <https://msdn.microsoft.com/en-us/library/wss56bz7.aspx>