**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

NGUYỄN KHOA TUYÊN MSSV:N15CCN100 Phần mềm Quản lí Học viên Lớp:C15CQCP01 2015-2018

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP CAO ĐẲNG**

***Đề tài*: “PHẦN MỀM QUẢN LÍ HỌC VIÊN”**

**Người hướng dẫn : TRƯƠNG BÁ THÁI**

**Sinh viên thực hiện : NGUYỄN KHOA TUYÊN**

**Mã số sinh viên : N15CCCN100**

**Lớp : C15CQCP01**

**Khoá** **: 2015 - 2018**

**Hệ** **: CAO ĐẲNG**

TP. HCM

2018

**TP.HCM, tháng 05/2018**

**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC/ CAO ĐẲNG**

***Đề tài*: “PHẦN MỀM QUẢN LÍ HỌC VIÊN”**

**Người hướng dẫn : TRƯƠNG BÁ THÁI**

**Sinh viên thực hiện : NGUYỄN KHOA TUYÊN**

**Mã số sinh viên : N15CCCN100**

**Lớp : C15CQCP01**

**Khoá** **: 2015 - 2018**

**Hệ** **: CAO ĐẲNG**

**TP.HCM, tháng 05/2018**

# **LỜI MỞ ĐẦU**

Sự phát triển của CNTT trong những năm gần đây đã đem lại những thành tựu to lớn trong việc phát triển kinh tế. Những chương trình tin học ứng dụng ngày càng nhiều, rất nhiều công việc thủ công trước đây nay đã được xử lí bằng các phần mềm chuyên dụng, giúp giảm đáng kể công sức, hoàn thành công việc nhanh chóng và chính xác. Để có một phần mềm ứng dụng đáp ứng được yêu cầu công việc đặt ra thì những người làm tin học phải biết phân tích thiết kế hệ thống làm việc của chương trình để từ đó xây dựng nên một phần mềm ứng dụng quản lí chương trình đó. Những chương trình, phần mềm ứng dụng đóng vai trò quan trọng trong lĩnh vực quản lý, nó là những công cụ hỗ trợ đắc lực và hầu như không thể thiếu nhằm đáp ứng đầy đủ nhất, thuận lợi nhất những đòi hỏi của công việc quản lý nhờ những công cụ trợ giúp có sẵn.

Chương trình quản lí học viên của một trung tâm tin học là một chương trình được xây dựng nhằm đáp ứng những đòi hỏi đặt ra của quá trình quản lý như ghi danh, nhập điểm, lưu kết quả thi, tìm kiếm thông tin học viên, giáo viên … và rất nhiều công việc khác một cách nhanh chóng và thuận tiện, chính xác một cách hệ thống.

# **LỜI CÁM ƠN**

Lời đầu tiên, em xin bày tỏ sự biết ơn chân thành đến Khoa Công nghệ thông tin II, Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại Thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi cho em được học tập tốt và thực hiện Đồ án thực tập tốt nghiệp: “Xây dựng Phần mềm Quản lí học viên.” này.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy giáo Trương Bá Thái đã quan tâm, tận tình hướng dẫn em thực hiện đề tài trong suốt thời gian hoàn thành Đồ án thực tập tốt nghiệp.

Cuối cùng, em xin chân thành cảm ơn quý thầy cô trong khoa đã luôn nhiệt tình giảng dạy trang bị những kiến thức quý báu, đồng hành cùng sinh viên trong suốt hơn 4 năm học vừa qua.

Trong quá trình thực hiện đề tài, em vẫn luôn cố gắng nỗ lực hết sức mình để có thể hoàn thành tốt nhất nhiệm vụ thầy cô đã giao.

Tuy nhiên, chắc hẳn sẽ còn nhiều thiếu sót do hạn chế về khả năng, kinh nghiệm cũng như quỹ thời gian. Kính mong quý thầy cô thông cảm và góp ý thêm để em có thể hoàn thiện tốt hơn Đồ án thực tập tốt nghiệp: “Xây dựng Phần mềm Quản lí học viên.” của mình và phát triển đồ án hoàn chỉnh hơn.

Em xin cảm ơn!

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập- Tự do- Hạnh phúc

TP. Hồ Chí Minh, ngày tháng năm 20……

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

THỰC TẬP TỐT NGHIỆP ĐẠI HỌC

1. Tên đề tài: **Xây dựng phần mềm Quản lí học viên**

2. Sinh viên: Nguyễn Khoa Tuyên Lớp: C15CQCP01

3. Giáo viên hướng dẫn: Trương Bá Thái

4. Nơi công tác: Khoa Công Nghệ Thông Tin 2, Học Viện Công Nghệ Bưu Chính Viễn Thông.

NỘI DUNG NHẬN XÉT

1. Đánh giá chung:

…………………………………………………………………………………………

…………………………………………………………………………………………

2. Đánh giá chi tiết:

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

3. Nhận xét về tinh thần, thái độ làm việc:

…………………………………………………………………………………………

…………………………………………………………………………………………

4. Kết luận:

…………………………………………………………………………………………

…………………………………………………………………………………………

5. Điểm hướng dẫn (): GIẢNG VIÊN HƯỚNG DẪN

(Ký, ghi rõ họ tên)

# **MỤC LỤC**

[**LỜI MỞ ĐẦU** i](#_Toc513036308)

[**LỜI CÁM ƠN** ii](#_Toc513036309)

[**MỤC LỤC** iv](#_Toc513036310)

[**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI** vi](#_Toc513036311)

[**1.1** **TỔNG QUAN ĐỀ TÀI:** vi](#_Toc513036312)

[1.1.1 BÀI TOÁN: vi](#_Toc513036313)

[1.1.2 MỤC TIÊU ĐỀ TÀI: vi](#_Toc513036314)

[1.1.2.1 Lý thuyết: vi](#_Toc513036315)

[1.1.2.2 Thực hành: vi](#_Toc513036316)

[1.1.2.3 Đối tượng sử dụng: vi](#_Toc513036317)

[**CHƯƠNG 2: NỀN TẢNG CÔNG NGHỆ** vii](#_Toc513036318)

[**2.1 NGÔN NGỮ WPF:** vii](#_Toc513036319)

[**2.2 XAML:** viii](#_Toc513036320)

[**2.3 LINQ TO SQL:** ix](#_Toc513036321)

[**CHƯƠNG 3: KHẢO SÁT HỆ THỐNG CỦA ĐỀ TÀI** xiv](#_Toc513036322)

[**3.1. LÍ DO CHỌN ĐỀ TÀI:** xiv](#_Toc513036323)

[**3.2 MỤC ĐÍCH CỦA ĐỀ TÀI:** xiv](#_Toc513036324)

[**3.3 ĐỐI TƯỢNG PHẠM VI NGHIÊN CỨU CÁC ỨNG DỤNG CỦA ĐỀ TÀI:** xiv](#_Toc513036325)

[**3. 4 Ý NGHĨA KHOA HỌC THỰC TIỄN CỦA ĐỀ TÀI:** xiv](#_Toc513036326)

[**CHƯƠNG 4: PHÂN TÍCH – TRIỂN KHAI** xv](#_Toc513036327)

[**4.1 PHÁT HIỆN THỰC THỂ:** xv](#_Toc513036328)

[**4.1.1. Biên lai (BIEN\_LAI):** xv](#_Toc513036329)

[**4.1.2. Ca học (CA\_HOC):** xv](#_Toc513036330)

[**4.1.3. Giáo viên (GIAO\_VIEN):** xv](#_Toc513036331)

[**4.1.4. Học viên (HOC\_VIEN):** xv](#_Toc513036332)

[**4.1.5. Khoá học (KHOA\_HOC):** xv](#_Toc513036333)

[**4.1.6. Kì thi (KI\_THI):** xv](#_Toc513036334)

[**4.1.7. Đề thi (DE\_THI):** xv](#_Toc513036335)

[**4.1.8. Lớp học (LOP\_HOC):** xv](#_Toc513036336)

[**4.1.9. Câu hỏi (CAU\_HOI):** xv](#_Toc513036337)

[**4.1.10 Tài khoản (TAI\_KHOAN):** xv](#_Toc513036338)

[**4.2 MÔ TẢ CHI TIẾT THỰC THỂ:** xvi](#_Toc513036339)

[***4.2.1. Biên lai*** xvi](#_Toc513036340)

[***4.2.2. Ca học:*** xvi](#_Toc513036341)

[***4.2.3. Giáo viên:*** xvi](#_Toc513036342)

[***4.2.4 Học viên:*** xvi](#_Toc513036343)

[***4.1.5 Khoá học:*** xvii](#_Toc513036344)

[***4.1.6 Kì thi:*** xvii](#_Toc513036345)

[***4.1.7 Đề thi:*** xvii](#_Toc513036346)

[***4.1.8 Lớp học:*** xviii](#_Toc513036347)

[***4.1.9 Câu hỏi:*** xviii](#_Toc513036348)

[***4.1.10 Tài khoản:*** xix](#_Toc513036349)

[**4.3 MÔ HÌNH THỰC THỂ LIÊN KẾT BAO GỒM THUỘC TÍNH:** xix](#_Toc513036350)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN VÀ KIỂM THỬ** xxi](#_Toc513036351)

[***Hình 5.1. Giao diện Đăng nhập.*** xxi](#_Toc513036352)

[***Hình 5.2. Giao diện menu chính (admin)***. xxii](#_Toc513036353)

[***Hình 5.3. Giao diện Đăng kí học viên mới.*** xxiii](#_Toc513036354)

[***Hình 5.4. Giao diện Quản lí thông tin học viên.*** xxiv](#_Toc513036355)

[***Hình 5.5. Giao diện Quản lí thông tin giáo viên.*** xxv](#_Toc513036356)

[***Hình 5.6. Giao diện Quản lí lớp học.*** xxvi](#_Toc513036357)

[***Hình 5.7. Giao diện quản lí khoá học.*** xxvii](#_Toc513036358)

[***Hình 5.8. Giao diện phân công giáo viên giảng dạy.*** xxviii](#_Toc513036359)

[***Hình 5.9. Giao diện quản lí tài khoản.*** xxix](#_Toc513036360)

[***Hình 5.10. Giao diện quản lí kì thi, đề thi, câu hỏi thi trắc nghiệm.*** xxx](#_Toc513036361)

[***Hình 5.11. Giao diện thi trắc nghiệm (học viên).*** xxxii](#_Toc513036362)

[***Hình 5.12. Giao diện xuất biên lai.*** xxxiii](#_Toc513036363)

[***Hình 5.13. Giao diện Kết quả thi.*** xxxiv](#_Toc513036364)

[**CHƯƠNG 6: TÀI LIỆU THAM KHẢO** xxxv](#_Toc513036365)

# **CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI**

* 1. **TỔNG QUAN ĐỀ TÀI:**

### 1.1.1 BÀI TOÁN:

- Đề tài Phần mềm Quản lí học viên được dựa trên nhu cầu thực tế khi các trung tâm học tập ngày càng phát triển. Việc quản lí học viên nhanh chóng, thuận lợi, tiết kiệm chi phí và nhân công là vấn đề cần thiết.

- Bài toán quản lí học viên không còn mới lạ với các trung tâm, hay các trường học giảng dạy. Việc quản lí luôn cần phải chính xác, dễ dàng, nhanh chóng, tiết kiệm được nguồn lực đến tối đa. Tại sao phải quản lí? Quản lí để dễ dàng đưa ra các phương thức hoạt động một cách nhanh gọn và dễ dàng nhất.

### 1.1.2 MỤC TIÊU ĐỀ TÀI:

### 1.1.2.1 Lý thuyết:

- Tìm hiểu ngôn ngữ lập trình Windows Presentation Foundation (WPF).

- Tìm hiểu về việc ứng dụng LinQ to SQL.

### 1.1.2.2 Thực hành:

- Xây dựng phần mềm Quản lí học viên cho phép các trung tâm quản lí được danh sách học viên, lớp, khoá học, giáo viên, tổ chức các kì thi trắc nghiệm, điểm thi…

### 1.1.2.3 Đối tượng sử dụng:

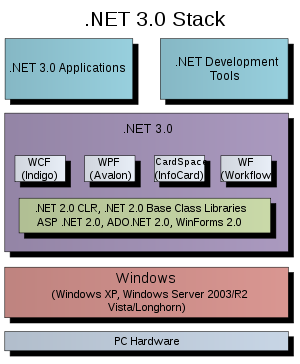
- Các trung tâm giảng dạy vừa và nhỏ.

# **CHƯƠNG 2: NỀN TẢNG CÔNG NGHỆ**

Trong đề tài này, ngôn ngữ được sử dụng là WPF, ngôn ngữ định nghĩa giao diện xaml và phương thức kết nối LinQ to SQL.

## **2.1 NGÔN NGỮ WPF:**

- **Windows Presentation Foundation** (viết tắt là WPF) do Microsoft phát triển, là công nghệ kế tiếp Windows Form dùng để xây dựng các ứng dụng dành cho máy trạm chạy hệ điều hành Windows. WPF được giới thiệu từ năm 2006 trong.NET Framework 3.0 (dưới tên gọi Avalon), công nghệ này nhận được sự quan tâm của cộng đồng lập trình viên bởi nhiều điểm đổi mới trong lập trình ứng dụng và khả năng xây dựng giao diện thân thiện, sinh động. Tại Việt Nam, WPF thực sự chưa phát triển so với nhánh khác là Silverlight (WPF/E).

[](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:DotNet3.0.svg)

[.NET Framework 3.0](https://vi.wikipedia.org/w/index.php?title=.NET_Framework_3.0&action=edit&redlink=1)

WPF sử dụng 2 thư viện lõi là PresentationCore và PresentationFramework để xử lý các điều hướng, ràng buộc dữ liệu, sự kiện và quản lý giao diện. WPF dựa trên nền tảng đồ họa là DirectX, xử lý vector, hỗ trợ gam màu rộng, cho phép tùy biến giá trị opacity hay tạo gradient một cách dễ dàng, cho phép tạo ảnh không gian 2 chiều hoặc 3 chiều. Thư viện thực thi của WPF tự động tính toán và tận dụng tài nguyên của hệ thống một cách tối ưu để giảm tải cho CPU.

Ngoài ra, WPF hỗ trợ tốt hơn Winform trong việc xử lý hình ảnh, âm thanh, video, quản lý phông chữ, quản lý hiển thị và chỉnh sửa văn bản. Các control trong WPF có thể được lồng ghép linh động để tạo ra giao diện do được viết bằng XAML. Một ứng dụng WPF có thể được xây dựng để chạy độc lập dưới dạng mở rộng EXE hoặc đóng gói với phần mở rộng là XBAP để có thể tích hợp lên website.

Thư viện thực thi WPF được tích hợp trong tất cả các hệ điều hành kể từ Windows Vista và Windows Server 2008.

Cho đến thời điểm hiện tại, WPF có 5 phiên bản: WPF 3.0 (11-2006), WPF 3.5 (11-2007), WPF 3.5sp1 (8-2008), WPF 4 (4-2010), và WPF 4.5 (8-2012).

- Thành phần:

+ Direct3D: Đồ họa và giao diện dựa trên Direct3D cho phép hiển thị các xử lý điểm - ảnh phức tạp hơn GDI's. Công nghệ này cho phép chia sẻ tác vụ xử lý cho GPU và giảm tải CPU.

+ Data Binding (Liên kết dữ liệu): *Data binding* là điểm nổi bật của WPF trong việc liên kết dữ liệu của ứng dụng. Tùy theo mục đích của chương trình, lập trình viên có thể xử lý liên kết theo ba loại:

one time: Chỉ tải dữ liệu 1 lần khi ứng dụng được thực thi.

one way: Ứng dụng chỉ có nhiệm vụ hiển thị dữ liệu, mọi cập nhật từ phía người dùng sẽ không được lưu trữ.

two way: Dữ liệu có thể được lấy và cập nhật từ phía người dùng.

one way to source: Trái ngược với OneTime, dữ liệu sẽ chỉ được nhận từ Interface.

+ Đa phương tiện: WPF cho phép xây dựng chương trình với sự hỗ trợ vector, xử lý hình ảnh, âm thanh, video, đồ họa 2D, 3D.

Đồ họa 2D: brushes, pens, geometries, và transforms.

Đồ họa 3D

Hình ảnh: hỗ trợ các định dạng: BMP, JPEG, PNG, TIFF, Windows Media Photo, GIF và ICON.

Video: hỗ trợ WMV, MPEG và AVI. Xử lý tốt hơn nếu Windows Media Player đã được cài và tích hợp codec.

+ Templates: Dữ liệu trong WPF có thể được định nghĩa để hiển thị theo yêu cầu thông qua các thuộc tính (properties) và việc định nghĩa template, style cho các control.

HierarchicalDataTemplate

ItemsPanelTemplate

ControlTemplate

DataTemplate

+ Hiệu ứng: WPF hỗ trợ hiệu ứng dựa trên thời gian thực thi của chương trình, thay vì frame-based. Các hiệu ứng đơn giản có thể xử lý bằng việc quản lý thời gian chạy, còn các xử lý phức tạp hơn cần đến sự hỗ trợ của lớp Animation.

Tất cả các thuộc tính của đối tượng trong WPF đều có thể được xử lý để trở nên sinh động hơn, miễn là nó thuộc loại Dependency Property.

Các lớp quản lý hiệu ứng tùy theo loại của thuộc tính được xử lý. Ví dụ, việc thay đổi màu sắc của control sẽ do lớp ColorAnimation hay thay đổi chiều dài hoặc chiều rộng do lớp DoubleAnimation đảm nhiệm.

Các hiệu ứng có thể được gom nhóm vào Storyboards và gán cho đối tượng và lớp này điều khiển việc bắt đầu hoặc dừng các hiệu ứng thông qua các phương thức tương ứng (Start, Stop, Pause...). Hiệu ứng cho ứng dụng có thể được viết trên giao diện bằng XAML hoặc định nghĩa bằng C#.

**2.2 XAML:**

XAML – Extensible Application Markup Language (phát âm là “zammel”) là một ngôn ngữ đánh dấu dùng để thể hiện các đối tượng trong .NET. Mặc dù XAML là một công nghệ có thể dùng để giải quyết nhiều loại vấn đề khác nhau, nhưng vai trò chính của nó là dùng để xây dựng giao diện người dùng WPF. Nói cách khác, XAML documents sẽ định nghĩa cách sắp xếp, thể hiện các control, buttons trong cửa sổ của một chương trình WPF.

Understanding XAML

Các nhà phát triển phần mềm từ lâu đã biết rằng, cách hiệu quả nhất để giảm bớt tính phức tạp, nâng cao chất lượng giao diện người dùng trong việc xây dựng ứng dụng chính là cần phải tách rời việc design ra khỏi mã nguồn chương trình. Bằng cách đó, các nhà thiết kế có thể tạo ra giao diện đẹp hơn, thể hiện được tính nghệ thuật, còn programmer có thể phát triển code thoải mái hơn.

Graphic User Interfaces before WPF

Trước đây, không có cách nào đơn giản để có thể tách rời giữa các nội dung về đồ họa ra khỏi code. Vấn đề trong tâm trong các ứng dụng sử dụng Windows Form đó là tất cả các form bạn tạo ra đều được định nghĩa lại toàn bộ bằng C#. Khi bạn tạo ra một control trên giao diện, Visual Studio sẽ tự động tạo ra code tương ứng từ các class có sẵn. Đáng tiếc là các nhà thiết kế giao diện lại không có bất kì công cụ nào có thể làm việc với mã nguồn C#. Để giải quyết vấn đề đau đầu này, Microsoft đã sáng tạo ra công nghệ WPF cùng với XAML. Khi bạn thiết kế một ứng dụng WPF trong Visual Studio, cửa sổ mà bạn đang thiết kế sẽ không chuyển thành code C# như Windows Form. Thay vào đó, giao diện này sẽ được định nghĩa bằng tập hợp các tag XAML. Khi chạy ứng dụng, các thẻ này được dùng để phát sinh ra các đối tượng dùng trong user interface.

**2.3 LINQ TO SQL:**

LINQ to SQL là một phiên bản hiện thực hóa của O/RM (object relational mapping) có bên trong .NET Framework bản “Orcas” (nay là .NET 3.5), nó cho phép bạn mô hình hóa một cơ sở dữ liệu dùng các lớp .NET. Sau đó bạn có thể truy vấn cơ sở dữ liệu (CSDL) dùng LINQ, cũng như cập nhật/thêm/xóa dữ liệu từ đó.

LINQ to SQL hỗ trợ đầy đủ transaction, view và các stored procedure (SP). Nó cũng cung cấp một cách dễ dàng để thêm khả năng kiểm tra tính hợp lệ của dữ liệu và các quy tắc vào trong mô hình dữ liệu của bạn.

Một số câu truy vấn cơ bản trong LINQ:

**Lấy một nguồn dữ liệu**  
Trong truy vấn LINQ, bước đầu tiên là đặc tả nguồn dữ liệu. Mệnh đề from phải có đầu tiên để mô tả nguồn dữ liệu (customers) và biến phạm vi (cust).

|  |  |
| --- | --- |
| 1  2  3 | //queryAllCustomers là 1 biến có kiểu dữ liệu IEnumerable<Customer> - dammio.com  var queryAllCustomers = from cust in customers                          select cust; |

Để lấy dữ liệu 1 trường (ví dụ tên khách hàng FullName) trong danh sách khách hàng (Customer) thì làm như sau:

|  |  |
| --- | --- |
| 1  2  3 | // Lấy tên đầy đủ FullName  var queryAllCustomers = from cust.FullName in customers                          select cust; |

**Biến phạm vi** (range variable) giống như biến lặp trong vòng lặp foreach ngoại trừ không có vòng lặp nào xảy ra trong 1 diễn giải truy vấn. Khi truy vấn được thực thi, biến phạm vi sẽ đóng vai trò là 1 tham chiếu với mỗi phần tử tiếp theo trong **customers**. Bởi vì trình biên dịch có thể suy luận dạng của **cust**, vì vậy chúng ta không cần phải đặc tả kiểu dữ liệu cho cust rõ ràng. Thêm nữa các biến phạm vi có thể được giới thiệu bằng mệnh đề let. Bạn có thể tìm hiểu thêm về mệnh đề let ở https://msdn.microsoft.com/en-us/library/bb383976.aspx.

**Lọc dữ liệu** (Filter)  
Lọc dữ liệu là câu lệnh truy vấn phổ biến ở dạng diễn giải Boolean (đúng hoặc sai). Câu truy vấn chỉ trả về các phần tử nếu diễn giải là đúng (true). Để lọc dữ liệu, chúng ta dùng mệnh đề **where**, trong đó mô tả các điều kiện lọc.

|  |  |
| --- | --- |
| 1  2  3 | var queryLondonCustomers = from cust in customers                             where cust.City == "Đà Lạt"                             select cust; |

Trong câu truy vấn ở ví dụ trên, chúng ta tìm những khách hàng (customers) ở thành phố ở “Đà Lạt”. Bạn có dùng các toán tử AND và OR trong C# để áp dụng các diễn giải lọc cần thiết trong mệnh đề where. Ví dụ tiếp theo, để tìm các khách hàng ở thành phố Đà Lạt và có tên là “Dammio” chúng ta có thể viết đoạn mã sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | //Tìm các khách hàng ở Đà Lạt và có tên là Dammio  var queryLondonCustomers = from cust in customers                           where cust.City == "Đà Lạt" && cust.Name == "Dammio"                           select cust;    //Tìm các khách hàng ở Đà Lạt hoặc có tên là Dammio  var queryLondonCustomers1 = from cust in customers                           where cust.City == "Đà Lạt" || cust.City == "Dammio"                           select cust; |

**Sắp xếp** (order)  
Mệnh đề **orderby**cho phép sắp xếp các phần tử theo thứ tự nào đó trong dữ liệu trả về. Để sắp xếp trường **Name** (tên) theo thứ tự alphabet với các khách hàng ở Đà Lạt, chúng ta có thể làm như ví dụ sau:

|  |  |
| --- | --- |
| 1  2  3  4  5 | var queryLondonCustomers3 =      from cust in customers      where cust.City == "Đà Lạt"      orderby cust.Name ascending      select cust; |

Để sắp xếp ngược lại, bạn có thể dùng mệnh đề **descending**. Ví dụ tiếp theo mô tả cách sắp xếp tăng theo trường Name (tên), và sau đó sắp giảm theo trường Birthday (ngày sinh) của các khách hàng ở Đà Lạt.

|  |  |
| --- | --- |
| 1  2  3  4  5 | var queryLondonCustomers3 =      from cust in customers      where cust.City == "Đà Lạt"      orderby cust.Name ascending, cust.Birthday descending      select cust; |

**Gom nhóm** (group)  
Mệnh đề **group** cho phép gom nhóm kết quả dựa trên 1 khóa được mô tả. Ví dụ, chúng ta muốn gom nhóm các khách hàng từ Đà Lạt theo thành phố (City), trong trường hợp này cust.City được gọi là khóa.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | // queryCustomersByCity là 1 IEnumerable<IGrouping<string, Customer>>  var queryCustomersByCity = from cust in customers                             group cust by cust.City; // gom theo thành phố    // customerGroup là 1 IGrouping<string, Customer>  foreach (var customerGroup in queryCustomersByCity)  {       Console.WriteLine(customerGroup.Key);       foreach (Customer customer in customerGroup)       {              Console.WriteLine("    {0}", customer.Name);       }  } |

Nếu thực hiện 1 câu truy vấn trong 1 mệnh đề group, **kết quả là dạng 1 danh sách lồng 1 danh sách**. Mỗi phần trong danh sách là 1 phần tử có 1 thành phần khóa và 1 danh sách các phần tử được gom nhóm theo khóa đó. Chúng ta có thể dùng vòng lặp foreach để lấy các nhóm khách hàng theo thành phố, trong mỗi nhóm đó lại lấy xuất ra tên của từng khách hàng như ví dụ trên. Ví dụ trên có kết quả như dưới đây:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Đà Lạt          Nguyễn Văn Anh          Trịnh Thị Chung          Hoàng Phan  Nha Trang          Phan Văn Tiến          Lê Công Minh  Đà Nẵng          Dammio Ta          John Adam |

Ví dụ tiếp theo hiển thị tên các nhóm thành phố nếu chứa hơn 2 khách hàng.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // custQuery is an IEnumerable<IGrouping<string, Customer>>  var custQuery =     from cust in customers                      group cust by cust.City into custGroup                      where custGroup.Count() > 2                      orderby custGroup.Key                      select custGroup; |

**Kết hợp** (join)  
Tương tự như SQL, kết hợp (join) dữ liệu xảy ra giữa các tập đối tượng dữ liệu mà chưa được mô hình rõ ràng trong nguồn dữ liệu. Ví dụ, chúng ta tìm tất cả khách hàng (customers) và các nhà phân phối (distributors) ở cùng thành phố. Mệnh đề **join**trong LINQ cho phép kết hợp dữ liệu trên các tập đối tượng theo vì dùng bảng cơ sở dữ liệu trực tiếp.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var innerJoinQuery = from cust in customers                       join dist in distributors on cust.City equals dist.City                       select new {                          CustomerName = cust.Name,                          DistributorName = dist.Name                       }; |

Trong ví dụ trên, chúng ta join 2 tập dữ liệu customers và distributors trên trường City sau đó xuất ra tập dữ liệu chứa CustomerName và DistributorName.

Trong LINQ, chúng ta không cần phải dùng mệnh đề join thường xuyên như trong SQL bởi vì các khóa ngoại trong LINQ thông thường được mô tả trong mô hình đối tượng (object model) như là các thuôc tính liên kết 1 tập các phần tử. Ví dụ, một đối tượng Customer chứa 1 tập các đối tượng Order. Thay vì dùng join, chúng ta có thể truy xuất danh sách orders bằng dấu chấm.

|  |  |
| --- | --- |
| 1 | from order in Customer.Orders ... |

**Chọn các trường dữ liệu (select)**  
Mệnh đề select dùng để chọn các dạng giá trị làm kết quả trả về trong 1 truy vấn LINQ.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | //Tạo nguồn dữ liệu  List<int> Scores = new List<int>() { 9, 92, 811, 44, 55 };    // Tạo câu truy vấn  IEnumerable<int> queryHighScores =      from score in Scores      where score > 80      select score;    // Thực hiện truy vấn  foreach (int i in queryHighScores)  {      Console.Write(i + " ");  } |

Trong ví dụ trên, chúng ta thấy mệnh đề select được dùng để chọn các điểm (score) với điều kiện score > 80 và hiển thị kết quả trên màn hình Console. Để chọn hết tất cả thuộc tính dữ liệu của đối tượng trong dữ liệu nguồn, chúng ta có thể làm như ví dụ sau:

|  |  |
| --- | --- |
| 1  2  3  4 | var studentQuery =                  from s in Students                  where s.ID > 111                  select s; |

Trong ví dụ trên chúng ta sử dụng **select s** với s là 1 phần tử trong danh sách Students, khi đó câu lệnh sẽ chọn hết tất cả thuộc tính của 1 sinh viên bao gồm s.ID, s.Name, s.Birthday, … và câu lệnh này tương tự như **select \*** trong SQL.

Ngoài ra để chọn 1 vài thuộc tính dữ liệu của các đối tượng nhưng không chọn hết tất cả các thuộc tính chúng ta có thể sử dụng **select new**.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var studentQuery =                  from s in Students                  where s.ID > 111                  select new ScoreInfo                  {                      Average = student.Scores.Average(),                      ID = s.ID                  }; |

Trong ví dụ, chúng ta chọn những sinh viên có ID > 111 và sau đó xuất ra thông tin điểm của từng sinh viên ở đối tượng **ScoreInfo** gồm 2 thuộc tính mới: Average (điểm trung bình của sinh viên) và ID của sinh viên đó.

**CHƯƠNG 3: KHẢO SÁT HỆ THỐNG CỦA ĐỀ TÀI**

**3.1. LÍ DO CHỌN ĐỀ TÀI:**

Như chúng ta đã thấy, hiện nay các trung tâm tin học, trung tâm ngoại ngữ đang ngày càng phát triển, cùng với đó là nhu cầu quản lí một cách hiệu quả, nhanh chóng, tiện lợi.

Việc quản lí bằng giấy tờ, sổ sách rất tốn thời gian và nhân lực. Bởi lẽ đó, họ rất cần những phần mềm quản lí đáp ứng đầy đủ những tính năng cơ bản như quản lí danh sách học viên, tổ chức các lớp học, quản lí điểm… Nhận thấy nhu cầu đó nên em chọn đề tài “ Xây dựng Phần mềm quản lí học viên”.

**3.2 MỤC ĐÍCH CỦA ĐỀ TÀI:**

Chương trình phải đáp ứng được các mục tiêu sau:

\* Cho trung tâm:

- Cập nhật và quản lí thông tin học viên: thông tin cá nhân, lớp học đăng kí, điểm thi.

- Quản lí, cập nhật danh sách các khoá học.

- Danh sách lớp học, số lượng học viên trong lớp.

- Quản lí thông tin giáo viên.

- Phân công giáo viên giảng dạy cho từng lớp.

- Tạo đề thi trắc nghiệm, tổ chức thi cho học viên và lưu lại kết quả.

- Tra cứu nhanh thông tin của học viên theo tên.

- Xuất biên lai thu học phí.

\* Cho học viên:

- Thi trắc nghiệm ngay trên phần mềm và biết kết quả ngay sau khi thi.

**3.3 ĐỐI TƯỢNG PHẠM VI NGHIÊN CỨU CÁC ỨNG DỤNG CỦA ĐỀ TÀI:**

- Đối tượng: Tất cả các trung tâm có nhu cầu quản lí học viên thông qua kĩ thuật số.

- Phạm vi nghiên cứu:

+ Tìm hiểu kĩ thuật lập trình, cách thức hoạt động và các đối tượng trong WPF.

+ Hiểu được cách sử dụng LinQ to SQL để kết nối tới cơ sở dữ liệu.

**3. 4 Ý NGHĨA KHOA HỌC THỰC TIỄN CỦA ĐỀ TÀI:**

Trong thời đại công nghiệp hoá, hiện đại hoá, việc ứng dụng vào hoạt động của phần mềm quản lí đã mang lại nhiều ý nghĩa như:

- Tiết kiệm thời gian, công sức và cả tiền bạc.

- Giúp dễ dàng quản lí, lưu trữ và bảo mật thông tin.

**CHƯƠNG 4: PHÂN TÍCH – TRIỂN KHAI**

**4.1 PHÁT HIỆN THỰC THỂ:**

**4.1.1. Biên lai (BIEN\_LAI):**

- Mỗi thực thể tượng trưng cho một biên lai thu học phí.

- Thuộc tính: MaBL, NgayBL, NoiDung, SoTien.

**4.1.2. Ca học (CA\_HOC):**

- Mỗi thực thể tượng trưng cho một ca học (buổi, giờ).

- Thuộc tính: MaCaHoc, Buoi, Gio.

**4.1.3. Giáo viên (GIAO\_VIEN):**

- Mỗi thực thể tượng trưng cho một giáo viên.

- Thuộc tính: MaGiaoVien, HoGiaoVien, TenGiaoVien, NgaySinh, Diachi, Email.

**4.1.4. Học viên (HOC\_VIEN):**

- Mỗi thực thể tượng trưng cho một học viên.

- Thuộc tính: MaHocVien, HoHocVien, TenHocVien, NgaySinh, DiaChi.

SoDT, NgheNghiep.

**4.1.5. Khoá học (KHOA\_HOC):**

- Mỗi thực thể tượng trưng cho một khoá học.

- Thuộc tính: MaKhoaHoc, TenKhoaHoc, NgayBD, NgayKT.

**4.1.6. Kì thi (KI\_THI):**

- Mỗi thực thể tượng trưng cho một kì thi.

- Thuộc tính: MaKiThi, TenKiThi.

**4.1.7. Đề thi (DE\_THI):**

- Mỗi thực thể tượng trưng cho một đề thi.

- Thuộc tính: MaDeThi.

**4.1.8. Lớp học (LOP\_HOC):**

- Mỗi thực thể tượng trưng cho một lớp học.

- Thuộc tính: MaLopHoc, TenLopHoc,SiSoDK, HocPhi.

**4.1.9. Câu hỏi (CAU\_HOI):**

- Mỗi thực thể tượng trưng cho một câu hỏi.

- Thuộc tính: MaCauHoi, CauHoi, DapAnA, DapAnB, DapAnC, DapAnD, TraLoi.

**4.1.10 Tài khoản (TAI\_KHOAN):**

- Mỗi thực thể tượng trưng cho một tài khoản.

- Thuộc tính: TaiKhoan, MatKhau, Loai.

**4.2 MÔ TẢ CHI TIẾT THỰC THỂ:**

***4.2.1. Biên lai***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaBL | int | Mã biên lai | x |  |
| 2 | NgayBL | date | Ngày xuất biên lai |  |  |
| 3 | NoiDung | nvarchar(500) | Nội dung biên lai |  |  |
| 4 | SoTien | int | Số tiền thu |  |  |

***4.2.2. Ca học:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaCaHoc | int | Mã ca học | x |  |
| 2 | Buoi | nvarchar(50) | Buổi học |  |  |
| 3 | Giờ | nvarchar(50) | Giờ học |  |  |

***4.2.3. Giáo viên:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaGiaoVien | nvarchar(7) | Mã giáo viên | x |  |
| 2 | HoTenGiaoVien | nvarchar(50) | Họ tên giáo viên |  |  |
| 3 | Ngày sinh | date | Ngày sinh của giáo viên |  |  |
| 4 | Địa chỉ | nvarchar(50) | Địa chỉ của giáo viên |  |  |
| 5 | Email | nvarchar(50) | Địa chỉ email của giáo viên |  |  |

***4.2.4 Học viên:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaHocVien | nvarchar(7) | Mã học viên | x |  |
| 2 | HoTenHocVien | nvarchar(50) | Họ và tên của học viên |  |  |
| 3 | NgaySinh | date | Ngày sinh của học viên |  |  |
| 4 | DiaChi | nvarchar(50) | Địa chỉ của học viên |  |  |
| 5 | SoDT | nvarchar(50) | Số điện thoại liên hệ |  |  |
| 6 | NgheNghiep | nvarchar(50) | Nghề nghiệp của học viên |  |  |
| 7 | Email | nvarchar(50) | Email của học viên |  |  |
| 8 | GioiTinh | nvarchar(10) | Giới tính của học viên |  |  |
| 9 | TaiKhoan | nvarchar(50) | Tài khoản được cấp phát để đăng nhập |  |  |

***4.1.5 Khoá học:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaKhoaHoc | nvarchar(7) | Mã Khoá học | x |  |
| 2 | TenKhoaHoc | nvarchar(50) | Tên của khoá học |  |  |
| 3 | NgayBD | date | Ngày bắt đầu khoá học |  |  |
| 4 | NgayKT | date | Ngày kết thúc khoá học |  |  |

***4.1.6 Kì thi:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaKiThi | nvarchar(7) | Mã kì thi | x |  |
| 2 | TenKiThi | nvarchar(50) | Tên kì thi |  |  |

***4.1.7 Đề thi:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaDeThi | int | Mã đề thi | x |  |
| 2 | MaKiThi | nvarchar(7) | Mã kì thi |  | x |

***4.1.8 Lớp học:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaLopHoc | nvarchar(7) | Mã lớp học | x |  |
| 2 | TenLopHoc | nvarchar(50) | Tên lớp học |  |  |
| 3 | SiSoDk | int | Số học viên đã đăng kí |  |  |
| 4 | HocPhi | int | Học phí |  |  |
| 5 | MaKhoaHoc | nvarchar(7) | Mã khoá học |  | x |

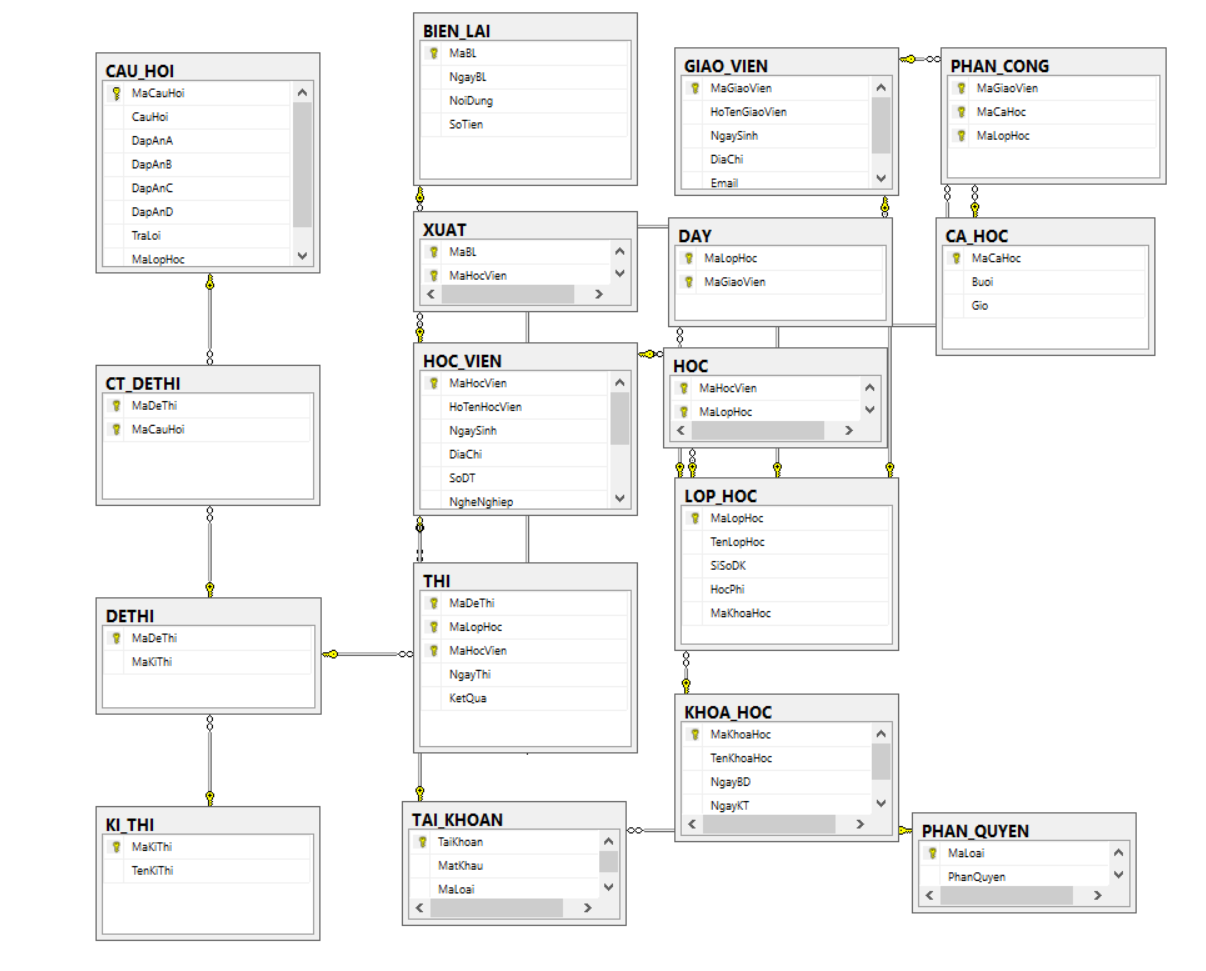
***4.1.9 Câu hỏi:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | MaCauHoi | int | Mã câu hỏi | x |  |
| 2 | CauHoi | nvarchar(500) | Nội dung câu hỏi |  |  |
| 3 | DapAnA | nvarchar(500) | Đáp án A |  |  |
| 4 | DapAnB | nvarchar(500) | Đáp án B |  |  |
| 5 | DapAnC | nvarchar(500) | Đáp án C |  |  |
| 6 | DapAnD | nvarchar(500) | Đáp án D |  |  |
| 7 | TraLoi | nvarchar(5) | Câu trả lời |  |  |

***4.1.10 Tài khoản:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu** | **Diễn giải** | **Khóa chính** | **Khóa**  **ngoại** |
| 1 | TaiKhoan | nvarchar(50) | Tên tài khoản | x |  |
| 2 | MatKhau | nvarchar(50) | Mật khẩu |  |  |
| 3 | MaLoai | int | Mã loại để phân quyền |  |  |

**4.3 MÔ HÌNH THỰC THỂ LIÊN KẾT BAO GỒM THUỘC TÍNH:**



***Hình 1. Mô hình thực thể liên kết có thuộc tính.***

**4.4 MÔ HÌNH QUAN HỆ:**

1. BIEN\_LAI(MaBL, NgayBL,NoiDung,SoTien)

2. CA\_HOC(MaCaHoc, Buoi, Gio)

3. HOC\_VIEN(MaHocVien, HoTenHocVien, NgaySinh, DiaChi, SoDT, NgheNghiep, GioiTinh, Email, TaiKhoan)

4. GIAO\_VIEN(MaGiaoVien, HoTenGiaoVien, NgaySinh, Email, DiaChi)

5. LOP\_HOC(MaLopHoc,MaKhoaHoc, TenLopHoc, SiSoDK, HocPhi)

6. DAY(MaLopHoc, MaGiaoVien)

7. HOC (MaLopHoc, MaHocVien)

8. XUAT(MaBL,MaHocVien)

9. KHOA\_HOC(MaKhoaHoc, TenKhoaHoc, NgayBD, NgayKT)

10. KI\_THI (MaKiThi, TenKiThi)

11. PHANCONG(MaGiaoVien, MaCaHoc, MaLopHoc)

12. DE\_THI(MaDeThi, MaKiThi)

13. CT\_DETHI ( MaDeThi, MaCauHoi)

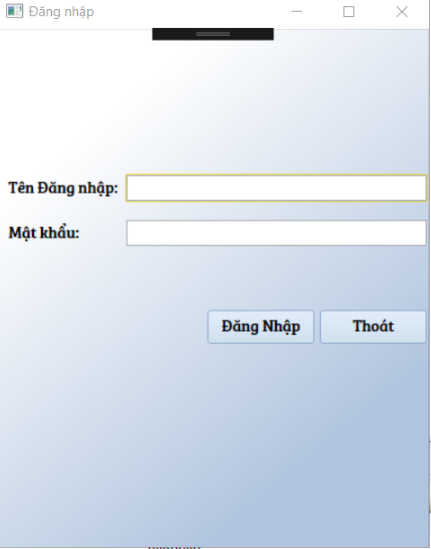
14. CAU\_HOI(MaCauHoi, CauHoi, DapAnA, DapAnB, DapAnC, DapAnD, CauTraLoi, MaLopHoc)

15. THI(MaDeThi, MaLopHoc, MaHocVien, NgayThi, KetQua)

16. TAI\_KHOAN(TaiKhoan, MatKhau, MaLoai)

17. PHAN\_QUYEN(MaLoai,PhanQuyen)

**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN VÀ KIỂM THỬ**

******

***Hình 5.1. Giao diện Đăng nhập.***

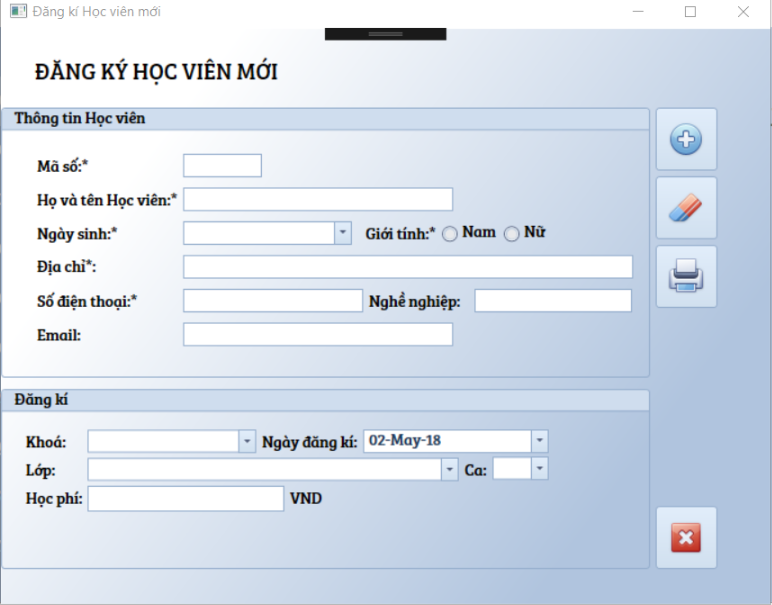
- Tiến hành đăng nhập vào hệ thống với tài khoản được tạo sẵn: tên tài khoản: admin, mật khẩu: admin. Nếu đăng nhập thành công sẽ mở form Main.

- Hoặc đăng nhập với tài khoản học viên đã được tạo sẵn. Nếu đăng nhập thành công sẽ chuyển tới form Thi trắc nghiệm.

******

***Hình 5.2. Giao diện menu chính (admin)***.

- Lựa chọn các chức năng ở thanh menu bên tay trái để vào các phần tương ứng

******

***Hình 5.3. Giao diện Đăng kí học viên mới.***

- Giao diện gồm có:

+ Các textbox tượng trưng cho các thông tin tương ứng như: Mã số, Họ tên học viên, Ngày sinh, Địa chỉ, Số điện thoại, Nghề nghiệp, Email, Học phí. Radion button Giới tính.

+ Các combobox tượng trưng cho các thông tin về Khoá học, Lớp học, Ngày đăng kí.

- Tiến hành nhập đầy đủ thông tin học viên, thông tin khoá học.

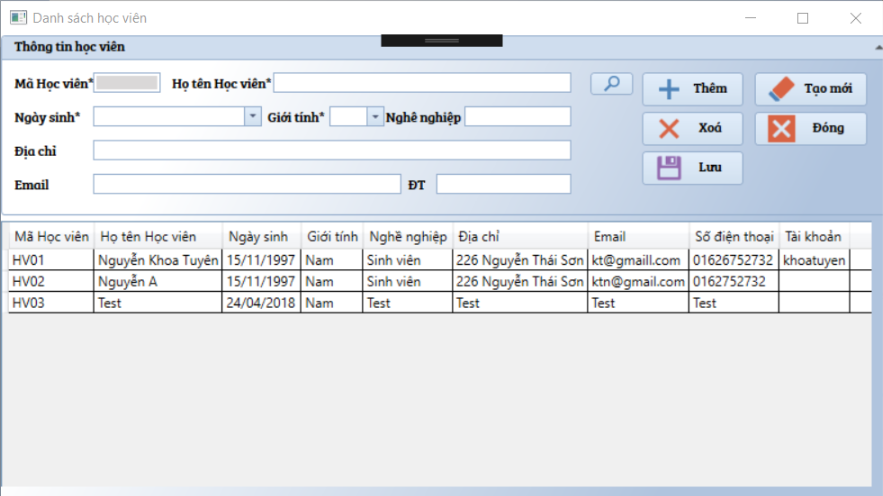
- Lưu ý: Khi chọn Khoá học, combobox Lớp học mới được hiển thị. Khi chọn Lớp học, Học phí sẽ được tự động hiển thị theo Lớp học tương ứng.

- Nhấn nút  để đăng kí học viên mới.

- Nhấn nút  để tạo mới các textbox.

- Nhấn nút  để chuyển tới form Xuất biên lai.

- Nhấn nút  để đóng.

******

***Hình 5.4. Giao diện Quản lí thông tin học viên.***

**-** Giao diện gồm có:

+ 1 datagrid hiển thị các thông tin của học viên.

+ Các text box và combobox hiển thị các thông tin tương ứng như: Mã học viên, Họ tên học viên, Ngày sinh, Giới tính, Nghề nghiệp, Địa chỉ, Email, ĐT( điện thoại).

+ Các nút: Tìm kiếm, Thêm, Xoá, Lưu, Tạo mới, Đóng.

- Tìm kiếm học viên:

**+** Nhập tên học viên muốn tìm kiếm vào ô Họ tên Học viên.

+ Nhấn nút  để tìm kiếm thông tin học viên.

- Thêm học viên mới: Nhấn nút  để mở ra form Đăng kí học viên mới.

- Xoá học viên:

+ Chọn học viên muốn xoá ở datagrid.

+ Nhấn nút  để xoá học viên muốn xoá.

+ Học viên đã được xoá khỏi cơ sở dữ liệu.

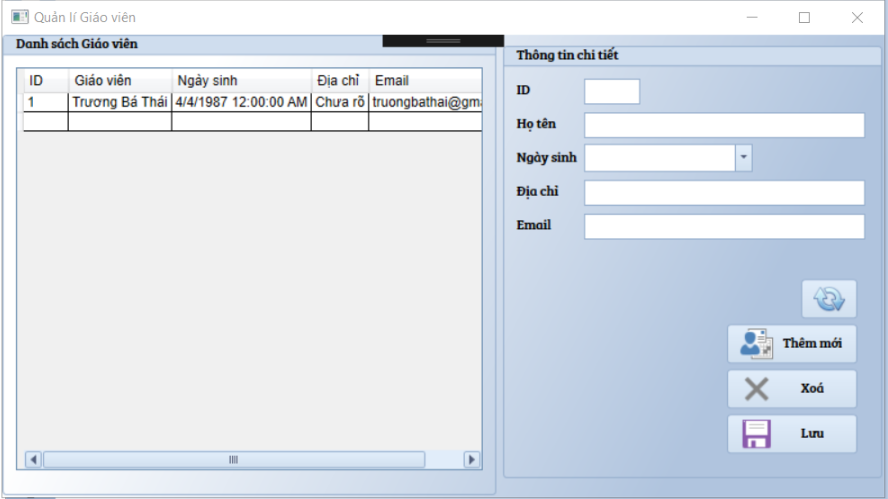
- Cập nhật thông tin học viên:  
+ Chọn học viên cần cập nhật thông tin.  
+ Các thông tin của học viên sẽ được hiển thị lên textbox tương ứng.

+ Chỉnh sửa các thông tin cần thiết.

+ Nhấn nút  để lưu lại thông tin vào cơ sở dữ liệu.

- Nhấn nút  để tạo mới các ô thông tin.

- Nhấn nút  để đóng cửa sổ.

******

***Hình 5.5. Giao diện Quản lí thông tin giáo viên.***

- Giao diện gồm có:

+ 1 datagrid hiển thị các thông tin của giáo viên.

+ Các textbox và combobox hiển thị các thông tin tương ứng: ID, Họ tên giáo viên, Ngày sinh, Địa chỉ, Email.

+ Các nút nhấn: Tạo mới, Thêm mới, Xoá, Lưu.

- Thêm mới giáo viên:

+ Nhập đầy đủ thông tin của giáo viên.

+ Nhấn nút  để thêm mới một giáo viên vào cơ sở dữ liệu.

- Xoá giáo viên đã tồn tại:

+ Chọn giáo viên cần xoá ở datagrid.

+ Thông tin giáo viên được chọn sẽ hiển thị lên các textbox và combobox tương ứng.

+ Nhấn nút  để xoá bỏ giáo viên đó khỏi cơ sở dữ liệu.

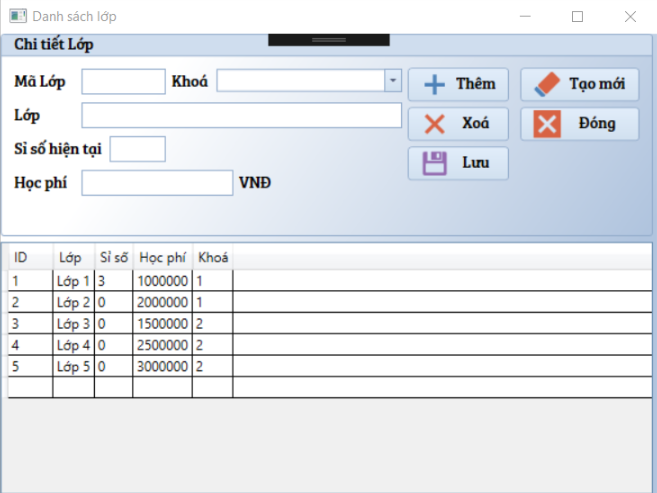
- Chỉnh sửa thông tin:

+ Chọn giáo viên cần chỉnh sửa ở datagrid.

+ Thông tin giáo viên được chọn sẽ hiển thị lên các textbox và combobox tương ứng.

+ Chỉnh sửa các thông tin trong các textbox và combobox cần thiết.

+ Nhấn nút  để lưu lại thông tin của giáo viên vào cơ sở dữ liệu.

******

***Hình 5.6. Giao diện Quản lí lớp học.***

- Giao diện gồm có:

+ 1 datagrid hiển thị thông tin Lớp học.

+ Các textbox và combobox hiển thị các thông tin tương ứng: Mã lớp, Khoá, Lớp, Sỉ số, Học phí.

+ Các nút nhấn: Thêm, Xoá, Lưu, Tạo mới, Đóng.

- Thêm mới lớp học:

+ Nhập đầy đủ thông tin cần thiết của lớp học.

+ Nhấn nút  để thêm mới lớp học vào cơ sở dữ liệu.

- Xoá lớp học đã tồn tại:

+ Chọn lớp học muốn xoá ở datagrid.

+ Nhấn nút  để xoá bỏ lớp học đã chọn khỏi cơ sở dữ liệu.

- Chỉnh sửa thông tin một lớp học đã tồn tại:

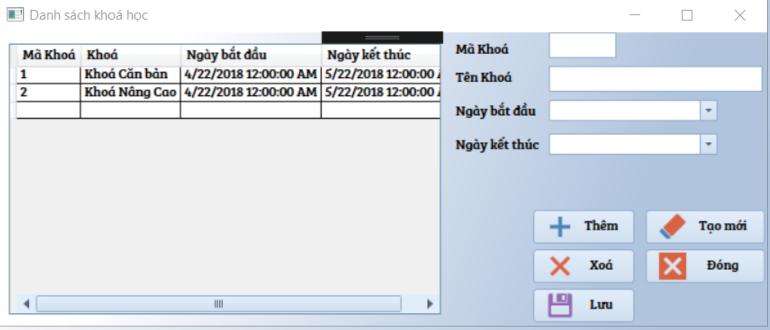
+ Chọn lớp học muốn chỉnh sửa tại datagrid.

+ Chỉnh sửa thông tin ở các ô tương ứng.

+ Nhấn nút  để lưu lại thông tin vào cơ sở dữ liệu.

- Nhấn nút  để tạo mới các ô thông tin.

- Nhấn nút  để đóng cửa sổ.

******

***Hình 5.7. Giao diện quản lí khoá học.***

- Giao diện gồm có:

+ 1 datagrid hiển thị các thông tin khoá học.

+ Các textbox và combobox hiển thị các thông tin tương ứng: Mã khoá, Tên khoá, Ngày bắt đầu khoá học, Ngày kết thúc khoá học.

+ Các nút nhấn: Thêm, Xoá, Lưu, Tạo mới, Đóng.

- Thêm mới khoá học:  
+ Nhập đầy đủ các thông tin cần thiết của khoá học.

+ Nhấn nút  để thêm khoá học mới vào cơ sở dữ liệu.

- Xoá khoá học đã tồn tại:   
+ Chọn khoá học đã tồn tại ở datagrid.

+ Nhấn nút  để xoá bỏ khoá học khỏi cơ sở dữ liệu.

- Chỉnh sửa khoá học đã tồn tại:

+ Chọn khoá học đã tồn tại ở datagrid.

+ Chỉnh sửa các thông tin cần thiết ở các ô tương ứng.

+ Nhấn nút  để lưu lại thông tin vào cơ sở dữ liệu.

- Nhấn nút  để tạo mới các ô thông tin.

- Nhấn nút  để đóng cửa sổ.

******

***Hình 5.8. Giao diện phân công giáo viên giảng dạy.***

- Giao diện gồm có:

+ 1 datagrid thể hiện lịch phân công giảng dạy.

+ Các combobox thể hiện các thông tin tương ứng: Tên giáo viên, Lớp, Ca học.

- Phân công giảng dạy:

+ Chọn giáo viên để phân công ở combobox Giáo viên.

+ Chọn lớp cần phân công ở combobox Lớp.

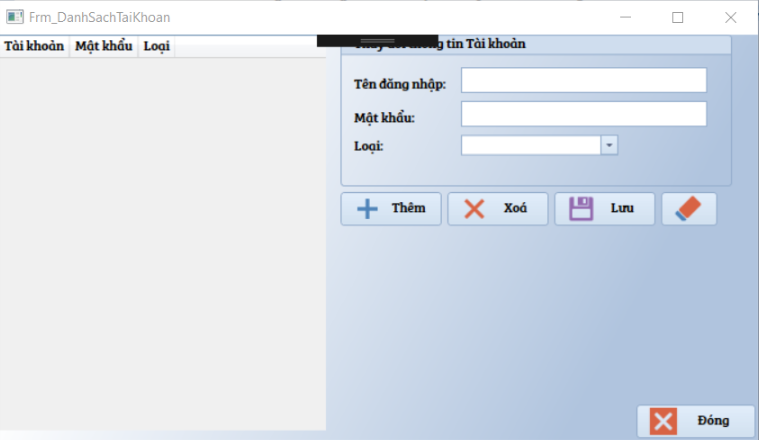
+ Chọn ca học ở combobox Ca học để có buổi và giờ tương ứng với mỗi ca.

+ Nhấn nút Phân công để lưu vào cơ sở dữ liệu.

- Huỷ bỏ lịch phân công đã tồn tại:

+ Chọn lịch phân công đã tồn tại ở datagrid.

+ Nhấn nút Huỷ để huỷ bỏ lịch phân công.

******

***Hình 5.9. Giao diện quản lí tài khoản.***

- Giao diện gồm có:

+ 1 datagrid hiển thị các tài khoản đã tồn tại trong cơ sở dữ liệu.

+ Các textbox và combobox hiển thị các thông tin tương ứng: Tên đăng nhập, Mật khẩu, Loại.

+ Các nút nhấn: Thêm, Xoá, Lưu, Tạo mới, Đóng.

- Tạo tài khoản mới:

+ Nhập đầy đủ thông tin tài khoản cần thiết: Tên đăng nhập, Mật khẩu.

+ Lựa chọn Loại tài khoản cần tạo: Admin hoặc Học viên.

+ Nhấn nút  để thêm tài khoản mới vào cơ sở dữ liệu.

- Xoá tài khoản đã tồn tại:   
+ Chọn tài khoản đã tồn tại ở datagrid.

+ Nhấn nút  để xoá bỏ tài khoản khỏi cơ sở dữ liệu.

- Chỉnh sửa mật khẩu của tài khoản đã tồn tại:

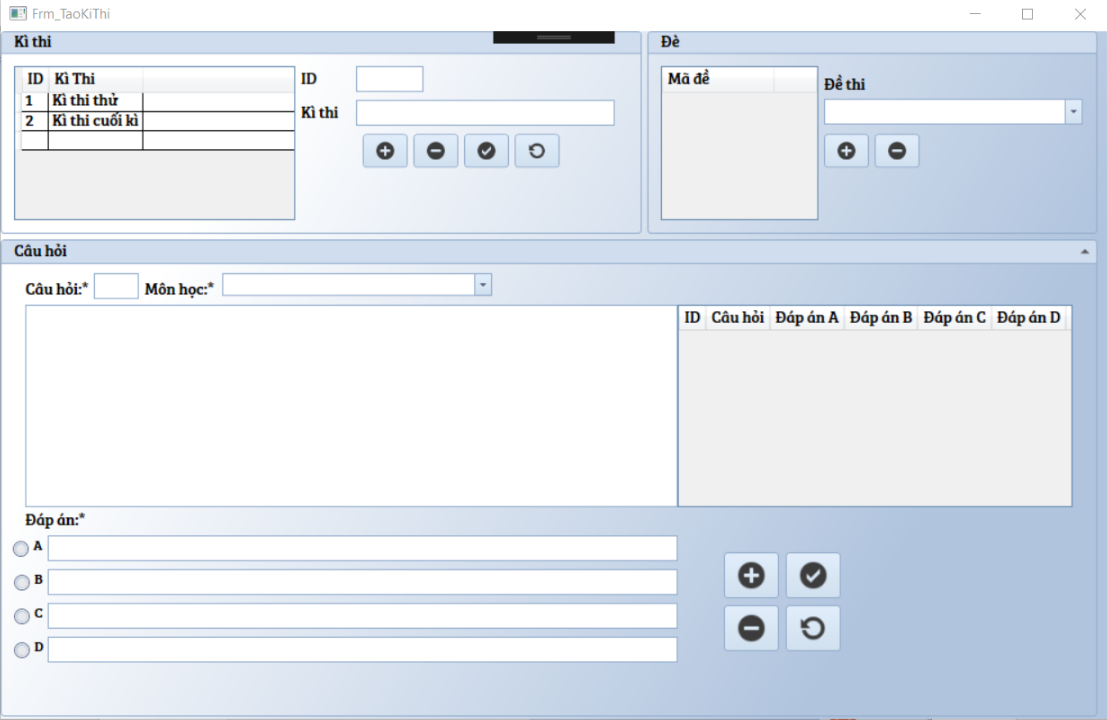
+ Chọn tài khoản đã tồn tại ở datagrid.

+ Chỉnh sửa mật khẩu.

+ Nhấn nút  để lưu lại thông tin vào cơ sở dữ liệu.

- Nhấn nút  để tạo mới các ô thông tin.

- Nhấn nút  để đóng cửa sổ.

******

***Hình 5.10. Giao diện quản lí kì thi, đề thi, câu hỏi thi trắc nghiệm.***

- Giao diện gồm có:

+ Groupbox Kì thi: Gồm có 1 datagrid hiển thị thông tin các kì thi đã tồn tại trong cơ sở dữ liệu; Các textbox hiển thị các thông tin tương ứng ID (mã kì thi), Kì thi. Các nút nhấn: Thêm, Xoá, Lưu, Tạo mới.

+ Groupbox Đề thi: Gồm có 1 datagrid hiển thị thông tin các đề thi đã tồn tại trong cơ sở dữ liệu; Combobox hiển thị thông tin đề thi. Các nút nhấn: Thêm, Xoá.

+ Groupbox Câu hỏi: Gồm có 1 datagrid hiển thị các câu hỏi đã tồn tại trong cơ sở dữ liệu; Các combobox, textbox hiển thị các thông tin tương ứng: Mã câu hỏi, Môn học, Nội dung câu hỏi, các đáp án. Radion Button đáp án A, B, C, D. Các nút nhấn: Thêm, Xoá, Lưu, Tạo mới.

- Tạo kì thi mới:  
+ Nhập đầy đủ thông tin cần thiết của kì thi.

+ Nhấn nút  để thêm mới kì thi vào cơ sở dữ liệu.

- Xoá kì thi mới:

+ Chọn kì thi muốn xoá trong datagrid.

+ Nhấn nút  để xoá bỏ kì thi khỏi cơ sở dữ liệu.

- Chỉnh sửa kì thi đã tồn tại:  
+ Chọn kì thi cần chỉnh sửa trong datagrid.

+ Chỉnh sửa các thông tin.

+ Nhấn nút  để ưu lại thông tin vào cơ sở dữ liệu.

- Nhấn nút  để làm mới các ô thông tin.

- Tạo đề thi mới:

+ Chọn kì thi chứa đề thi.

+ Nhập đầy đủ thông tin cần thiết của đề thi.

+ Nhấn nút  để thêm mới đề thi vào cơ sở dữ liệu.

- Xoá đề thi đã tồn tại:

+ Chọn đề thi đã tồn tại trên datagrid.

+ Nhấn nút  để xoá đề thi ra khỏi cơ sở dữ liệu.

- Tạo mới câu hỏi:

+ Chọn đề thi chứa câu hỏi ở combobox Đề thi.

+ Nhập các thông tin cần thiết của câu hỏi.

+ Nhấn nút  để thêm câu hỏi vào cơ sở dữ liệu.

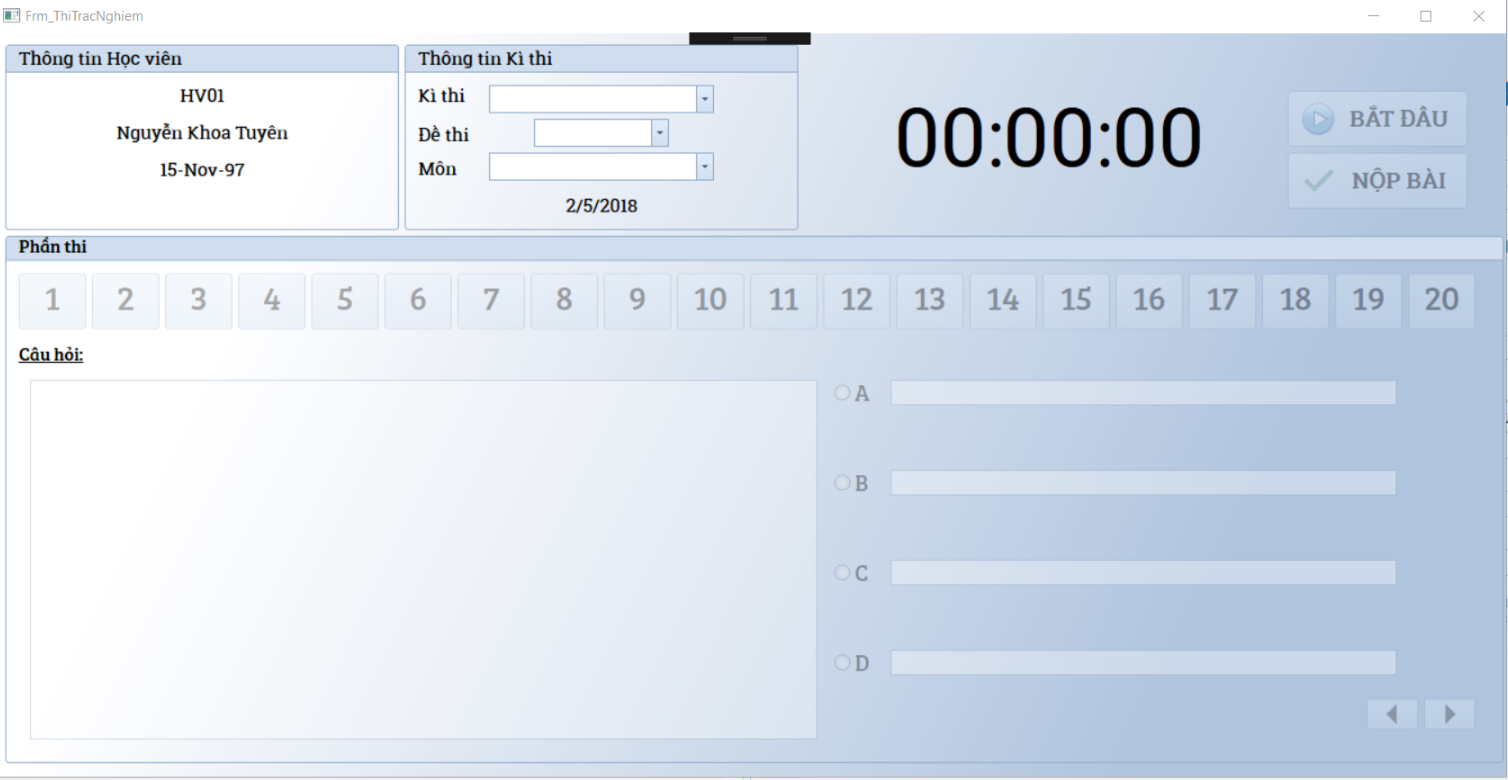
- Xoá câu hỏi đã tồn tại trong cơ sở dữ liệu:

+ Chọn câu hỏi cần xoá trên datagrid.

+ Nhấn nút  để xoá câu hỏi ra khỏi cơ sở dữ liệu.

- Chọn câu hỏi cần chỉnh sửa trên datagrid:  
+ Chỉnh sửa các thông tin cần thiết ở các ô textbox tương ứng.

+ Nhấn nút  để lưu lại thông tin vào cơ sở dữ liệu.

******

***Hình 5.11. Giao diện thi trắc nghiệm (học viên).***

- Khi đăng nhập bằng tài khoản được phân quyền là học viên sẽ được đưa tới form này.

- Giao diện gồm có:  
+ Groupbox Thông tin học viên: hiển thị các thông tin học viên tham gia thi trắc nghiệm.

+ Groupbox Thông tin Kì thi: hiển thị các thông tin kì thi, đề thi, môn, ngày thi.

+ Groupbox Phần thi: hiển thị thông tin các câu hỏi, đáp án và các nút nhấn chuyển câu hỏi.

+ Textblock hiển thị thời gian đếm ngược và hiển thị điểm.

+ Các nút nhấn: Bắt đầu, Nộp bài.

- Tiến hành thi:

+ Chọn lần lượt Kì thi, Đề thi, Môn ở những combobox tương ứng.

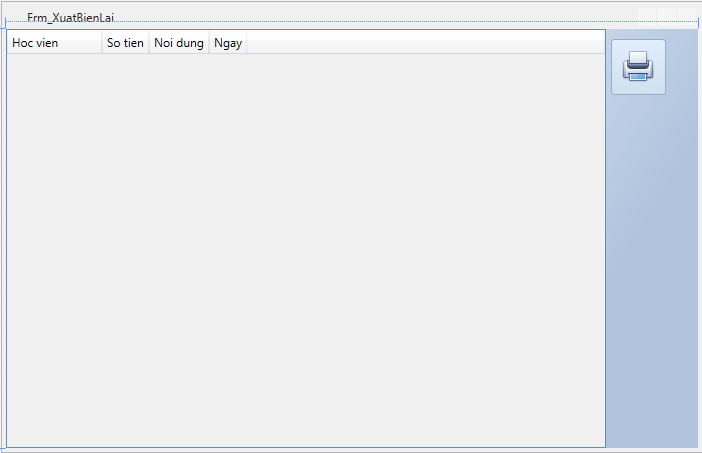
+ Nhấn nút Bắt đầu.

+ Đọc câu hỏi, chọn đáp án ở các radionbutton tương ứng.

+ Nhấn 2 nút  để chuyển câu hỏi tiếp theo, hoặc trở lại câu hỏi trước. Hoặc nhấn các nút số để chuyển tới câu hỏi tương ứng.

+ Nhấn nút Nộp bài sau khi hoàn thành bài thi.

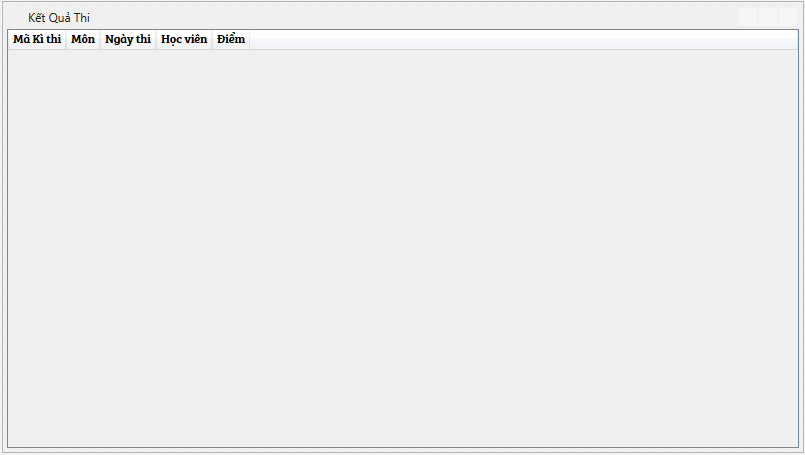
+ Điểm sẽ được hiển thị trực tiếp trên textblock đồng hồ đếm ngược.



***Hình 5.12. Giao diện xuất biên lai.***

- Giao diện gồm có: 1 datagrid hiển thị thông tin Học viên, số tiền, nội dung của biên lai, ngày thu biên lai.

- Nhấn nút  để xuất ra file excel.



***Hình 5.13. Giao diện Kết quả thi.***

- Giao diện gồm có 1 datagrid hiển thị các thông tin Kì thi, môn thi, ngày thi, tên học viên, điểm.

**CHƯƠNG 6: TÀI LIỆU THAM KHẢO**

1. Wikimedia.

2. CsharpNotesForProfessionals.

3. DotNETFrameworkNotesForProfessionals.