 BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TPHCM

KHOA CÔNG NGHỆ THÔNG TIN

**ĐỒ ÁN**

**CÔNG CỤ VÀ MÔI TRƯỜNG**

**PHÁT TRIỂN PHẦN MỀM**

**“QUẢN LÝ CHUYẾN BAY”**

**GVHD:** Dương Hữu Thành

**SVTH:** Nguyễn Thị Bé Thảo - 1411060290

Nguyễn Phan Thanh Tuyền - 1512360111

TPHCM, Tháng 1 năm 2018

**MỤC LỤC**

Lời Cảm Ơn 3

**CHƯƠNG 1: Tổng quan về đồ án** 4

**CHƯƠNG 2: Giới thiệu về Visual Stidio** 5

2.1 Giới thiệu về Visual Studio 5

2.2 Giao diện làm việc 6

2.3 Các chức năng 8

2.4 Sử dụng 12

**CHƯƠNG III: Giới thiệu về SQL** 13

**CHƯƠNG IV: Thông tin về đồ án** 16

4.1 Giao diện phần mềm 17

**LỜI NÓI ĐẦU**

Ngày nay việc sử dụng tin học trong tất cả các công việc đã không còn xa lạ, Đối với ngành hàng không, việc cần có một phần mềm giúp hỗ trợ cho các hoạt động quản lý là không thể thiếu. “Phần Mềm Quản Lý Chuyến Bay” là phần mềm giúp các nhân viên bán vé có thể thực hiện tốt công việc của mình một cách nhanh chóng và chính xác.

Chúng em cảm ơn thầy Dương Hữu Thành đã nhiệt tính giúp đỡ và cho chúng em nhiều kiến thức quý giá về môn Phân tích thiết kế hệ thống thông tin để chúng em có thể hoàn thành đồ án một cách tốt nhất. Tuy nhiên đồ án của chúng em chắc chắc còn nhiều sai sót mong thầy thông cảm và góp ý thêm cho chúng em.

**CHƯƠNG I: TỔNG QUAN VỀ ĐỒ ÁN**

1.1 Giới thiệu về đồ án

Ngày nay thương mại hàng không ngày càng trở nên rộng rãi do những nhu cầu của khách hàng muốn sử dụng dịch vụ để di chuyển đến điểm đến tới những mối làm ăn ,gia đình , hoặc có thể là đi du lịch.Những hãng hàng không ra đời để phục vụ các yêu cáu đó của khách hàng.Các hãng hàng không lớn như là ViệtNamAirLine , VietJet… Nhưng đi kèm theo nó phải cần có một hệ thống giám sát như một bên thứ 3 kiểm chứng để khách hàng có thể lên máy bay thông qua bên thứ ba ấy. Đó chính là một phần mềm nhỏ quản lý chuyến bay, vé đã đặt cho khách ,thông tin khách hàng ,.. Thể nên nhóm chúng em viết một phần mềm nhỏ nhằm phục vụ việc quản lý dễ hơn cho các nhân viên .Đó là phần mềm quản lý bán vé máy bay.

1.2 Mục tiêu về tiện ích

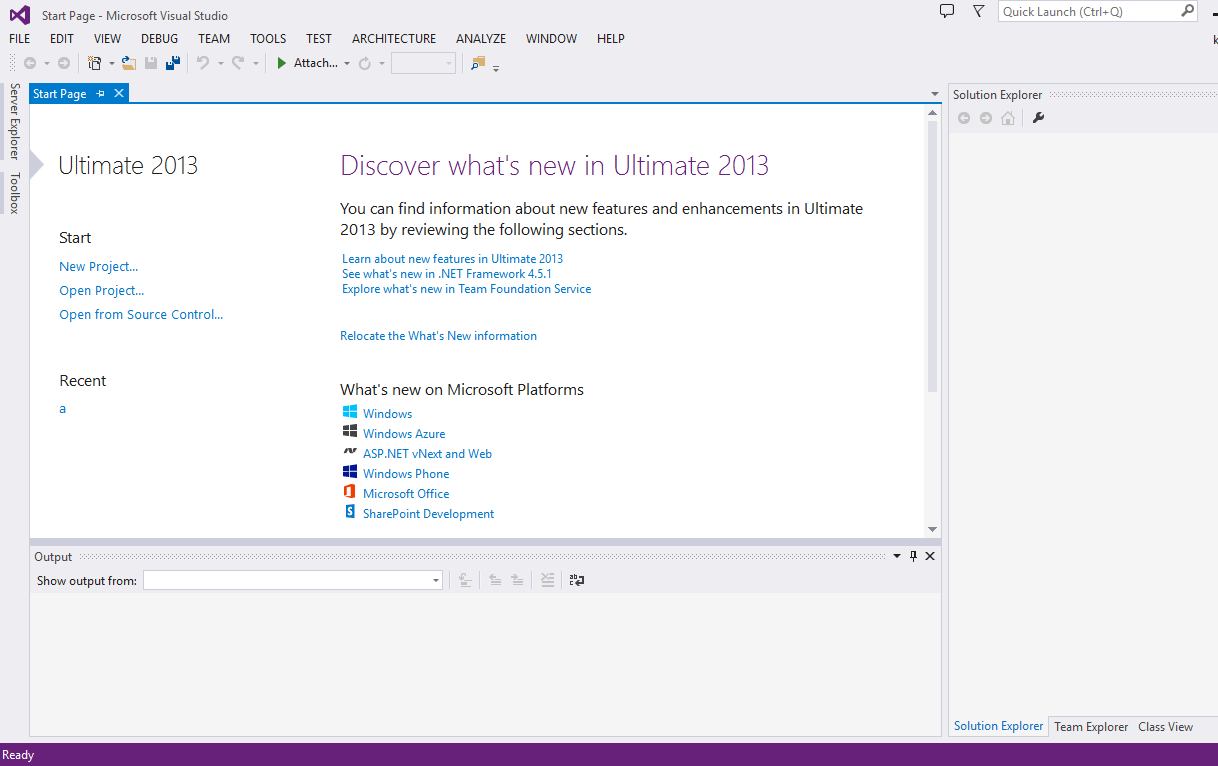
Giúp cho nhân viên quản lý dễ hơn về các khoản đặt vé từ hành khách . Quản lý chuyến bay cũng như tìm kiếm tên hóa đơn và các thông tin các chuyến bay và khách hàng .

**CHƯƠNG II: GIỚI THIỆU VỀ VISUAL STDIO**

2.1 Giới thiệu về Visual stdio

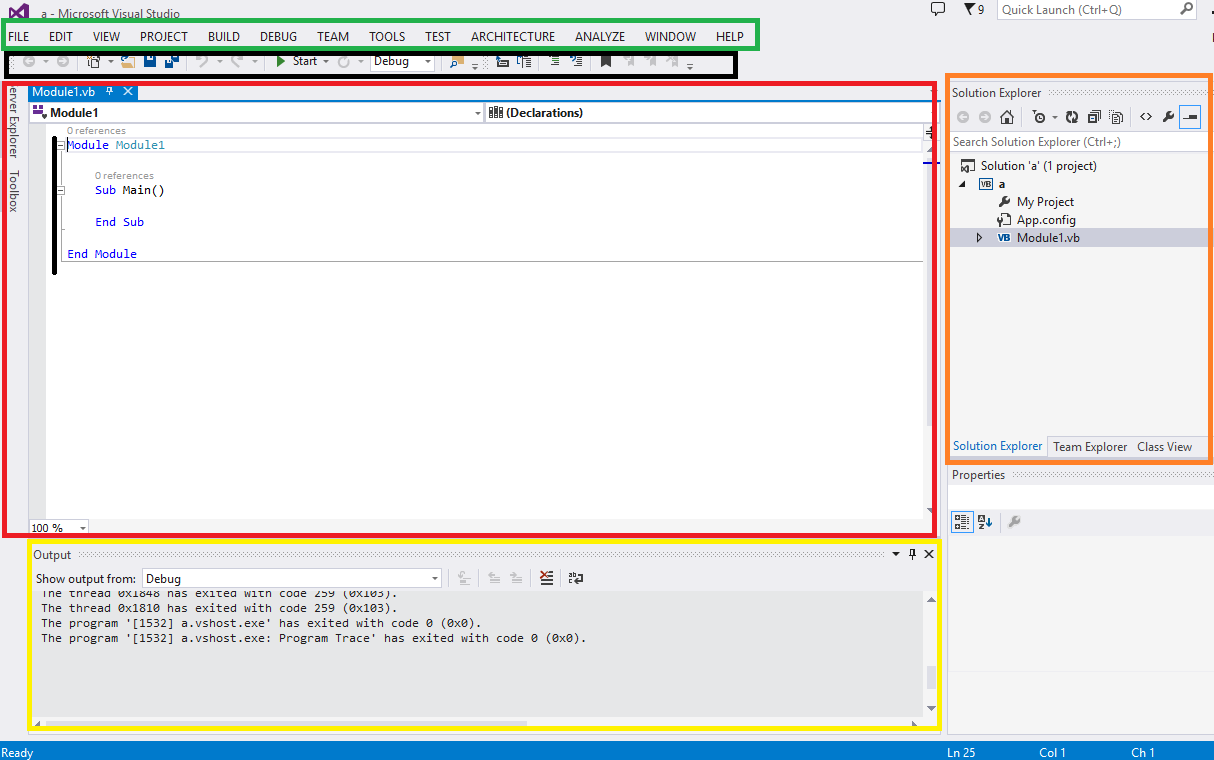
[***Microsoft Visual Studio***](https://www.visualstudio.com/) là một môi trường phát triển tích hợp từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của [Microsoft](https://www.microsoft.com/vi-vn/) như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.



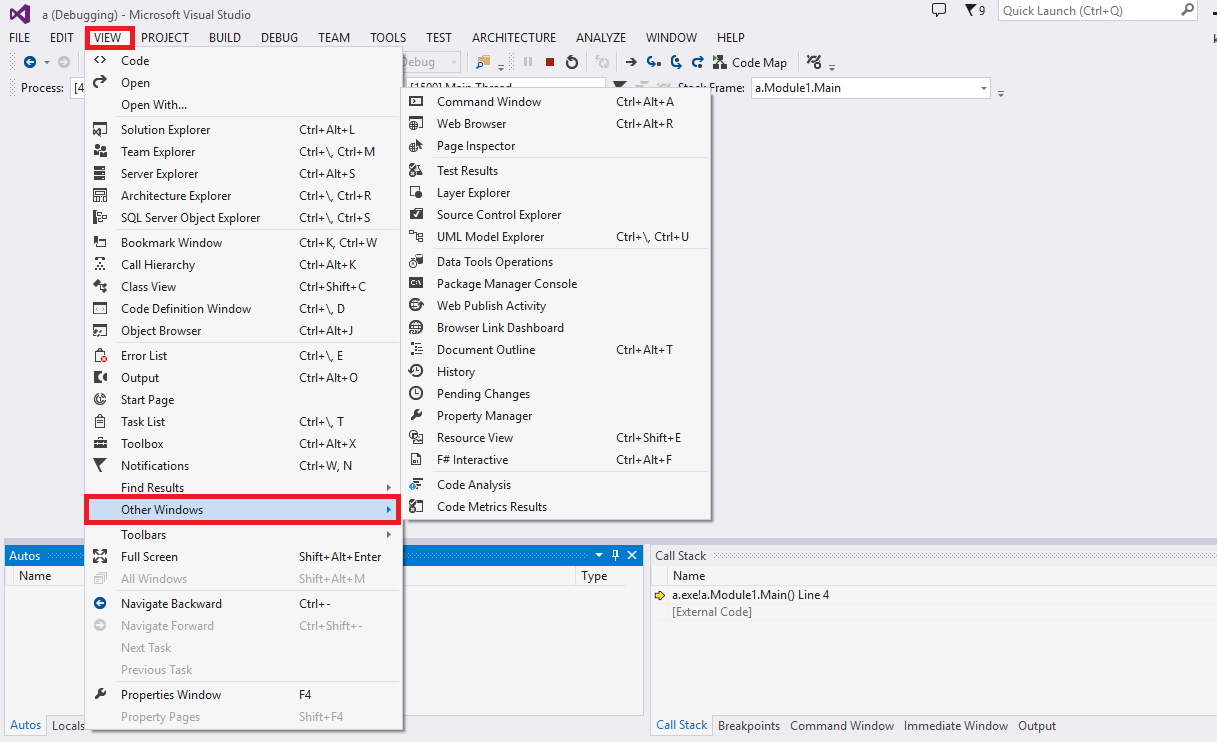
Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C,[1] C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C thăng (thông qua Visual C#) và F thăng (như của Visual Studio 2010[2]). Hỗ trợ cho các ngôn ngữ khác như J++/J thăng, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

2.2 Giao diện làm việc



* **Màu cam**:(Solution Explorer**)**là cửa sổ hiển thị Solution, các Project và các tập tin trong project.
* **Màu đỏ:**đây là khu vực để lập trình viên viết mã nguồn cho chương trình. Cửa sổ lập trình cho một tập tin trong Project sẽ hiển thị khi người dùng nháy đúp chuột lên tập tin đó trong cửa sổ  Solution Explorer.
* **Màu vàng**:(Output)đây là cửa sổ hiển thị các thông tin, trạng thái của Solution khi build hoặc của chương trình khi debug.
* **Màu đen**(Toolbar) với các công cụ hỗ trợ người dùng trong việc viết mã và debug (các công cụ trên thanh có thể thay đổi khi bắt đầu debug).
* **Màu xanh lá cây**(Thanh menu) với đầy đủ các danh mục chứa các chức năng của VS. Khi người dùng cài thêm những trình cắm hỗ trợ VS (ví dụ như Visual Assist), thanh menu này sẽ cập nhật thêm menu của các trình cắm (nếu có).

**Một số cửa sổ bị ẩn bạn có thể tìm ở trong** :**View -> Other Windows**, các thanh công cụ được đặt trong **View -> Toolbars**.



2.3 Các tính năng

* **Biên tập mã**

Giống như bất kỳ [IDE](https://vi.wikipedia.org/wiki/IDE) khác, nó bao gồm một [trình soạn thảo mã](https://vi.wikipedia.org/w/index.php?title=Tr%C3%ACnh_so%E1%BA%A1n_th%E1%BA%A3o_m%C3%A3&action=edit&redlink=1) hỗ trợ [tô sáng cú pháp](https://vi.wikipedia.org/w/index.php?title=T%C3%B4_s%C3%A1ng_c%C3%BA_ph%C3%A1p&action=edit&redlink=1) và [hoàn thiện mã](https://vi.wikipedia.org/w/index.php?title=Ho%C3%A0n_thi%E1%BB%87n_m%C3%A3&action=edit&redlink=1) bằng cách sử dụng [IntelliSense](https://vi.wikipedia.org/w/index.php?title=IntelliSense&action=edit&redlink=1) không chỉ cho các [biến](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_(l%E1%BA%ADp_tr%C3%ACnh)), [hàm](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_con)và các [phương pháp](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_th%E1%BB%A9c_l%E1%BA%ADp_tr%C3%ACnh) mà còn các cấu trúc ngôn ngữ như [vòng điều khiển](https://vi.wikipedia.org/w/index.php?title=Ki%E1%BB%83m_so%C3%A1t_l%C6%B0u_l%C6%B0%E1%BB%A3ng&action=edit&redlink=1) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1). IntelliSense được hỗ trợ kèm theo cho các ngôn ngữ như [XML](https://vi.wikipedia.org/wiki/XML), [Cascading Style Sheets](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript) khi phát triển các [trang web](https://vi.wikipedia.org/wiki/Trang_web) và các [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web). Các đề xuất tự động hoàn chỉnh được xuất hiện trong một hộp danh sách phủ lên trên đỉnh của trình biên tập mã. Trong Visual Studio 2008 trở đi, nó có thể được tạm thời bán trong suốt để xem mã che khuất bởi nó.  Các trình biên tập mã được sử dụng cho tất cả các ngôn ngữ được hỗ trợ.

Các trình biên tập mã Visual Studio cũng hỗ trợ cài đặt dấu trang trong mã để điều hướng nhanh chóng. Hỗ trợ điều hướng khác bao gồm [thu hẹp các khối mã lệnh](https://vi.wikipedia.org/w/index.php?title=Thu_h%E1%BA%B9p_c%C3%A1c_kh%E1%BB%91i_m%C3%A3_l%E1%BB%87nh&action=edit&redlink=1) và [tìm kiếm gia tăng](https://vi.wikipedia.org/w/index.php?title=T%C3%ACm_ki%E1%BA%BFm_gia_t%C4%83ng&action=edit&redlink=1), ngoài việc tìm kiếm văn bản thông thường và tìm kiếm [Biểu thức chính quy](https://vi.wikipedia.org/wiki/Bi%E1%BB%83u_th%E1%BB%A9c_ch%C3%ADnh_quy). Các trình biên tập mã cũng bao gồm một bìa kẹp đa mục và một danh sách công việc. Các trình biên tập mã hỗ trợ lưu lại các đoạn mã được lặp đi lặp lại nhằm để chèn vào mã nguồn sử dụng về sau. Một công cụ quản lý cho đoạn mã được xây dựng là tốt. Những công cụ này nổi lên như các cửa sổ trôi nổi có thể được thiết lập để tự động ẩn khi không sử dụng hoặc neo đậu đến các cạnh của màn hình. Các trình biên tập mã Visual Studio cũng hỗ trợ [cải tiến mã nguồn](https://vi.wikipedia.org/wiki/C%E1%BA%A3i_ti%E1%BA%BFn_m%C3%A3_ngu%E1%BB%93n) bao gồm tham số sắp xếp lại, biến và phương pháp đổi tên, khai thác và đóng gói [giao diện](https://vi.wikipedia.org/w/index.php?title=Giao_di%E1%BB%87n&action=edit&redlink=1) các [lớp](https://vi.wikipedia.org/wiki/L%E1%BB%9Bp_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) thành viên bên trong những trạng thái giữa những thứ khác.

Visual Studio có tính năng biên dịch nền (còn gọi là biên dịch gia tăng). Như mã đang được viết, Visual Studio biên dịch nó trong nền để cung cấp thông tin phản hồi về cú pháp và biên dịch lỗi, được đánh dấu bằng một gạch dưới gợn sóng màu đỏ. Biên dịch nền không tạo ra mã thực thi, vì nó đòi hỏi một trình biên dịch khác hơn là để sử dụng tạo ra mã thực thi. Biên dịch nền ban đầu được giới thiệu với [Microsoft Visual Basic](https://vi.wikipedia.org/wiki/Microsoft_Visual_Basic) nhưng bây giờ đã được mở rộng cho tất cả các ngôn ngữ.

* **Trình gỡ lỗi**

Visual Studio có một trình gỡ lỗi hoạt động vừa là một trình gỡ lỗi cấp mã nguồn và là một trình gỡ lỗi cấp máy. Nó hoạt động với cả hai mã quản lý cũng như [ngôn ngữ máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) và có thể được sử dụng để gỡ lỗi các ứng dụng được viết bằng các ngôn ngữ được hỗ trợ bởi Visual Studio. Ngoài ra, nó cũng có thể đính kèm theo quy trình hoạt động và theo dõi và gỡ lỗi những quy trình. Nếu mã nguồn cho quá trình hoạt động có sẵn, nó sẽ hiển thị các mã như nó đang được chạy. Nếu mã nguồn không có sẵn, nó có thể hiển thị các tháo gỡ. Các *Visual Studio debugger* cũng có thể tạo [bãi bộ nhớ](https://vi.wikipedia.org/w/index.php?title=B%C3%A3i_b%E1%BB%99_nh%E1%BB%9B&action=edit&redlink=1) cũng như tải chúng sau để gỡ lỗi. Các chương trình đa luồng cao cấp cũng được hỗ trợ. Trình gỡ lỗi có thể được cấu hình sẽ được đưa ra khi một ứng dụng đang chạy ngoài Visual Studio bị treo môi trường.

Trình gỡ lỗi cho phép thiết lập các [*breakpoint*](https://vi.wikipedia.org/w/index.php?title=Breakpoint&action=edit&redlink=1) (mà cho phép thực thi được tạm thời dừng lại tại một vị trí nhất định) và *watch* (trong đó giám sát các giá trị của biến là việc thực hiện tiến bộ). *Breakpoint* có thể có điều kiện, nghĩa là chúng được kích hoạt khi điều kiện được đáp ứng. Mã có thể được biểu diễn, tức là chạy một dòng (của mã nguồn) tại một thời điểm. Nó có hoặc là *bước sang* các chức năng để gỡ lỗi bên trong nó, hoặc là *nhảy qua nó*, tức là, việc thực hiện các chức năng không có sẵn để kiểm tra thủ công. Trình gỡ lỗi hỗ trợ *Edit and Continue*, nghĩa là, nó cho phép mã được chỉnh sửa khi nó đang được sửa lỗi (chỉ có 32 bit, không được hỗ trợ trong 64 bit). Khi gỡ lỗi, nếu con trỏ chuột di chuyển lên bất kỳ biến, giá trị hiện tại của nó được hiển thị trong phần chú giải ("chú thích dữ liệu"), nơi mà nó cũng có thể được thay đổi nếu muốn. Trong quá trình viết mã, các trình gỡ lỗi của Visual Studio cho phép một số chức năng được gọi ra bằng tay từ cửa sổ công cụ Immediate. Các thông số cho phương thức được cung cấp tại các cửa sổ *Immediate*.

* **Thiết kế**

*Windows Forms Designer*: được sử dụng để xây dựng [GUI](https://vi.wikipedia.org/wiki/GUI) sử dụng [Windows Forms](https://vi.wikipedia.org/w/index.php?title=Windows_Forms&action=edit&redlink=1); bố trí có thể được xây dựng bằng các nút điều khiển bên trong hoặc khóa chúng vào bên cạnh mẫu. Điều khiển trình bày dữ liệu (như hộp văn bản, hộp danh sách, vv) có thể được liên kết với các nguồn dữ liệu như [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) hoặc [truy vấn](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1). Các điều khiển dữ liệu ràng buộc có thể được tạo ra bằng cách rê các mục từ cửa sổ nguồn dữ liệu lên bề mặt thiết kế. Các giao diện người dùng được liên kết với mã sử dụng một mô hình [lập trình hướng sự kiện](https://vi.wikipedia.org/w/index.php?title=L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_s%E1%BB%B1_ki%E1%BB%87n&action=edit&redlink=1). Nhà thiết kế tạo ra bằng [C thăng](https://vi.wikipedia.org/wiki/C_th%C4%83ng) hay [VB.NET](https://vi.wikipedia.org/wiki/VB.NET) cho ứng dụng.

*WPF Designer*: có tên mã là *Cider*, được giới thiệu trong Visual Studio 2008. Giống như *Windows Forms Designer*, hỗ trợ kéo và thả ẩn dụ. Sử dụng [tương tác người-máy](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%C3%A1c_ng%C6%B0%E1%BB%9Di-m%C3%A1y) nhắm mục tiêu theo [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation). Nó hỗ trợ các chức năng WPF bao gồm [kết nối dữ liệu](https://vi.wikipedia.org/w/index.php?title=K%E1%BA%BFt_n%E1%BB%91i_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1) và tự động hóa bố trí quản lý. Nó tạo ra mã [XAML](https://vi.wikipedia.org/w/index.php?title=XAML&action=edit&redlink=1) cho giao diện người dùng. Các tập tin *XAML* được tạo ra là tương thích với [Microsoft Expression Design](https://vi.wikipedia.org/w/index.php?title=Microsoft_Expression_Design&action=edit&redlink=1), sản phẩm thiết kế theo định hướng. Các mã *XAML* được liên kết với mã đang sử dụng một mô hình [code-behind](https://vi.wikipedia.org/wiki/Code-behind).

*Web designer/development*: Visual Studio cũng bao gồm một trình soạn thảo và thiết kế trang web cho phép các trang web được thiết kế bằng cách kéo và thả các đối tượng. Nó được sử dụng để phát triển các ứng dụng [ASP.NET](https://vi.wikipedia.org/wiki/ASP.NET) và hỗ trợ [HTML](https://vi.wikipedia.org/wiki/HTML), [CSS](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Nó sử dụng mô hình [code-behind](https://vi.wikipedia.org/wiki/Code-behind) để liên kết với mã ASP.NET. Từ Visual Studio 2008 trở đi, công cụ bố trí được sử dụng bởi các nhà thiết kế web được chia sẻ với [Microsoft Expression Web](https://vi.wikipedia.org/w/index.php?title=Microsoft_Expression_Web&action=edit&redlink=1). Ngoài ra [ASP.NET MVC Framework](https://vi.wikipedia.org/w/index.php?title=ASP.NET_MVC_Framework&action=edit&redlink=1) hỗ trợ cho công nghệ [MVC](https://vi.wikipedia.org/wiki/MVC) là tải xuống riêng biệt và dự án [ASP.NET Dynamic Data](https://vi.wikipedia.org/w/index.php?title=ASP.NET_Dynamic_Data&action=edit&redlink=1) có sẵn từ Microsoft.

*Class designer*: Các *lớp thiết kế* được dùng để biên soạn và chỉnh sửa các lớp (bao gồm cả các thành viên và truy cập của chúng) sử dụng mô hình [UML](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%B4_h%C3%ACnh_h%C3%B3a_th%E1%BB%91ng_nh%E1%BA%A5t). Các lớp thiết kế có thể tạo ra mã phác thảo [C thăng](https://vi.wikipedia.org/wiki/C_th%C4%83ng) và [VB.NET](https://vi.wikipedia.org/wiki/VB.NET) cho các lớp và cá phương thức. Nó cũng có thể tạo ra sơ đồ lớp từ các lớp viết tay.

*Data designer*: *Thiết kế dữ liệu* có thể được sử dụng để chỉnh sửa đồ họa [giản đồ cơ sở dữ liệu](https://vi.wikipedia.org/w/index.php?title=Gi%E1%BA%A3n_%C4%91%E1%BB%93_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1) bao gồm các bảng, khóa chính, khóa ngoại và các rằng buộc. Nó cũng có thể được sử dụng để thiết kế các truy vấn từ các giao diện đồ họa.

*Mapping designer*: Từ Visual Studio 2008 trở đi, *thiết kế ánh xạ* được dùng bởi [Language Integrated Query](https://vi.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1) để thiết kế các ánh xạ giữa các giản đồ cơ sở dữ liệu và các lớp để đóng gói dữ liệu. Các giải pháp mới từ cách tiếp cận [*ORM*](https://vi.wikipedia.org/w/index.php?title=ORM&action=edit&redlink=1), [*ADO.NET Entity Framework*](https://vi.wikipedia.org/w/index.php?title=ADO.NET_Entity_Framework&action=edit&redlink=1) sẽ thay thế và cải thiện các công nghệ cũ.

* **Các công cụ khác**

*Open Tabs Browser*: được sử dụng để liệt kê tất cả thẻ đang mở và chuyển đổi giữa chúng. Được viện dẫn bằng cách sử dụng CTRL+TAB.

*Properties Editor*: được sử dụng để chỉnh sửa các thuộc tính trong một cửa sổ giao diện bên trong Visual Studio. Nó liệt kê tất cả các thuộc tính có sẵn (gồm chỉ đọc và những thuộc tính có thể được thiết lập) cho tất cả các đối tượng bao gồm các lớp, biểu mẫu, trang web và các hạng mục khác.

*Object Browser*: là một [không gian tên](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian_t%C3%AAn) và trình duyệt [lớp thư viện](https://vi.wikipedia.org/w/index.php?title=L%E1%BB%9Bp_th%C6%B0_vi%E1%BB%87n&action=edit&redlink=1) cho [Microsoft NET](https://vi.wikipedia.org/w/index.php?title=Microsoft_NET&action=edit&redlink=1). Nó có thể được sử dụng để duyệt các không gian tên (được sắp xếp theo thứ bậc) trong [Assembly (CLI)](https://vi.wikipedia.org/w/index.php?title=Assembly_(CLI)&action=edit&redlink=1). Các hệ thống phân cấp có thể hoặc không có thể phản ánh các tổ chức trong hệ thống tập tin.

*Solution Explorer*: theo cách nói trong Visual Studio, một giải pháp là một tập hợp các tập tin mã và các nguồn khác được sử dụng để xây dựng một ứng dụng. Các tập tin trong một giải pháp được sắp xếp theo thứ bậc, mà có thể có hoặc không thể phản ánh các tổ chức trong hệ thống tập tin. *Solution Explorer* được sử dụng để quản lý và duyệt các tập tin trong một giải pháp.

*Team Explorer*: được sử dụng để tích hợp các khả năng của [Team Foundation Server](https://vi.wikipedia.org/w/index.php?title=Team_Foundation_Server&action=edit&redlink=1), [Revision Control System](https://vi.wikipedia.org/w/index.php?title=Revision_Control_System&action=edit&redlink=1) và là cơ sở cho môi trường [CodePlex](https://vi.wikipedia.org/wiki/CodePlex) đối với dự án mã nguồn mở. Ngoài việc kiểm soát nguồn nó cung cấp khả năng xem và quản lý các công việc riêng lẻ (bao gồm cả lỗi, nhiệm vụ và các tài liệu khác) và để duyệt thống kê [TFS](https://vi.wikipedia.org/w/index.php?title=Team_Foundation_Server&action=edit&redlink=1). Nó được bao gồm như là một phần của một cài đặt TFS và cũng có sẵn để tải xuống cho Visual Studio. *Team Explorer* cũng có sẵn như là một môi trường độc lập duy nhất để truy cập các dịch vụ TFS.

*Data Explorer*: được sử dụng để quản lý cơ sở dữ liệu trên [Microsoft SQL Server](https://vi.wikipedia.org/w/index.php?title=Microsoft_SQL_Server&action=edit&redlink=1). Nó cho phép tạo ra và sửa đổi các bảng cơ sở dữ liệu (hoặc bằng cách ban hành các lệnh [T-SQL](https://vi.wikipedia.org/w/index.php?title=T-SQL&action=edit&redlink=1) hoặc bằng cách sử dụng các thiết kế dữ liệu). Nó cũng có thể được sử dụng để tạo các truy vấn và các [thủ tục lưu trữ](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%A7_t%E1%BB%A5c_l%C6%B0u_tr%E1%BB%AF&action=edit&redlink=1) trong *T-SQL* hoặc trong [Managed code](https://vi.wikipedia.org/w/index.php?title=Managed_code&action=edit&redlink=1) thông qua [SQL CLR](https://vi.wikipedia.org/w/index.php?title=SQL_CLR&action=edit&redlink=1). Có sẵn *gỡ lỗi* và hỗ trợ [IntelliSense](https://vi.wikipedia.org/w/index.php?title=IntelliSense&action=edit&redlink=1).

*Server Explorer*: công cụ được sử dụng để quản lý các kết nối cơ sở dữ liệu trên một máy tính truy cập được. Nó cũng được sử dụng để duyệt chạy [Windows Services](https://vi.wikipedia.org/w/index.php?title=Windows_Services&action=edit&redlink=1), quầy thực hiện, [Windows Event Log](https://vi.wikipedia.org/w/index.php?title=Windows_Event_Log&action=edit&redlink=1) và [hàng đợi tin nhắn](https://vi.wikipedia.org/w/index.php?title=H%C3%A0ng_%C4%91%E1%BB%A3i_tin_nh%E1%BA%AFn&action=edit&redlink=1) và sử dụng chúng như một nguồn dữ liệu.

*Dotfuscator Software Services Community Edition*: Visual Studio bao gồm một phiên bản *light* của sản phẩm [PreEmptive Solutions](https://vi.wikipedia.org/w/index.php?title=PreEmptive_Solutions&action=edit&redlink=1)' [Dotfuscator](https://vi.wikipedia.org/wiki/Dotfuscator) cho mã gây rối và giảm kích thước ứng dụng. Khởi đầu với Visual Studio 2010, phiên bản này của Dotfuscator sẽ bao gồm khả năng [Runtime Intelligence](https://vi.wikipedia.org/w/index.php?title=Runtime_Intelligence&action=edit&redlink=1) cho phép tác giả thu thập cách sử dụng của người dùng cuối, hiệu suất, tính ổn định và các thông tin từ các ứng dụng của họ chạy trong sản xuất.

*Text Generation Framework*: Visual Studio bao gồm một khung tạo văn bản đầy đủ được gọi là [Text Template Transformation Toolkit T4](https://vi.wikipedia.org/w/index.php?title=Text_Template_Transformation_Toolkit_T4&action=edit&redlink=1) cho phép Visual Studio tạo ra tập tin văn bản từ các mẫu hoặc trong *IDE* hoặc thông qua mã.

*ASP.NET Web Site Administration Tool*: [công cụ quản trị trang web ASP.NET](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_c%E1%BB%A5_qu%E1%BA%A3n_tr%E1%BB%8B_trang_web_ASP.NET&action=edit&redlink=1) cho phép cấu hình các trang web [ASP.NET](https://vi.wikipedia.org/wiki/ASP.NET).

*Visual Studio Tools for Office*: [Công cụ Visual Studio cho Ofice](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_c%E1%BB%A5_Visual_Studio_cho_Ofice&action=edit&redlink=1) là một *SDK* và một add-in cho Visual Studio bao gồm các công cụ để phát triển cho các bộ [Microsoft Office](https://vi.wikipedia.org/wiki/Microsoft_Office). Trước đây (với Visual Studio.NET 2003 và Visual Studio 2005) đó là một [*SKU*](https://vi.wikipedia.org/wiki/SKU) riêng biệt mà chỉ hỗ trợ [Visual C#](https://vi.wikipedia.org/w/index.php?title=Visual_C&action=edit&redlink=1) [Visual Basic.NET](https://vi.wikipedia.org/wiki/Visual_Basic.NET) hoặc đã được đưa vào *Team Suite*. Với Visual Studio 2008, nó không còn là một *SKU* riêng biệt nhưng lại kèm trong các phiên bản chuyên nghiệp và cao hơn. Một thời gian chạy riêng biệt được yêu cầu khi triển khai các giải pháp *VSTO*.

* **Khả năng mở rộng**

Visual Studio cho phép các nhà phát triển viết các phần mở rộng cho Visual Studio để mở rộng tính năng của nó. Những phần mở rộng "cắm vào" Visual Studio và mở rộng tính năng của nó. Các phần mở rộng đến ở dạng *macro*, *add-in* và *các gói*. Các macro đại diện cho các nhiệm vụ lặp đi lặp lại và hành động mà các nhà phát triển có thể ghi lại theo chương trình để tiết kiệm, phát lại, và phân phối. Tuy nhiên, không thể thực hiện các lệnh mới hoặc tạo ra các cửa sổ công cụ. Chúng được viết bằng [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic) và không được biên dịch. Các trình bổ sung cung cấp quyền truy cập vào mô hình đối tượng Visual Studio và có thể tương tác với các công cụ *IDE*. Các trình bổ sung có thể được sử dụng để thực hiện các chức năng mới và có thể thêm các cửa sổ công cụ mới. Các trình bổ sung được cắm vào *IDE* thông qua *COM* và có thể được tạo ra trong bất kỳ ngôn ngữ COM-tuân thủ. Gói được tạo ra bằng cách sử dụng *Visual Studio SDK* và cung cấp mức cao nhất của khả năng mở rộng. Chúng có thể tạo ra các thiết kế và các công cụ khác, cũng như tích hợp các ngôn ngữ lập trình khác. Visual Studio SDK cung cấp các [API](https://vi.wikipedia.org/wiki/API) không được quản lý cũng như một API quản lý để thực hiện các nhiệm vụ này. Tuy nhiên, các API quản lý không phải là toàn diện như một không được quản lý. Các phần mở rộng được hỗ trợ trong phiên bản Standard và cao hơn của phiên bản Visual Studio 2005. [Phiên bản Express](https://vi.wikipedia.org/w/index.php?title=Visual_Studio_Express&action=edit&redlink=1) không hỗ trợ phần mở rộng.

Visual Studio 2008 giới thiệu *Visual Studio Shell* cho phép phát triển một phiên bản tùy biến của *IDE*. *Visual Studio Shell* định nghĩa một tập hợp các *VSPackages* cung cấp các chức năng cần thiết trong bất kỳ *IDE*. Trên hết, các gói khác có thể được thêm vào tùy chỉnh các cài đặt. Các chế độ biệt lập của vỏ tạo ra một *AppID* mới, nơi các gói được cài đặt. Những thứ này được bắt đầu với một thực thi khác nhau. Nó nhằm mục đích cho sự phát triển của môi trường phát triển tùy chỉnh, hoặc cho một ngôn ngữ cụ thể hoặc một kịch bản cụ thể. Các chế độ tích hợp cài đặt các gói vào AppID của các phiên bản Professional / Standard / Team System, do đó các công cụ tích hợp vào các phiên bản. *Visual Studio Shell* là miễn phí tải về.

Sau khi phát hành Visual Studio 2008, Microsoft đã tạo ra *Visual Studio Gallery*. Nó phục vụ như vị trí trung tâm cho đăng tải thông tin về phần mở rộng cho Visual Studio. Phát triển cộng đồng cũng như phát triển thương mại có thể tải lên thông tin về các phần mở rộng của họ đến *Visual Studio.NET 2002* thông qua Visual Studio 2010. Người sử dụng trang web có thể đánh giá và xem lại các phần mở rộng để giúp đánh giá chất lượng các phần mở rộng được đăng. *RSS feed* thông báo cho người dùng trên bản cập nhật tới trang web và các tính năng gắn thẻ cũng được lên kế hoạch.

2.4 Sử dụng

* Người kiểm thử chủ yếu thao tác trên Microsoft Test Manager(MTM).
* Microsoft cung cấp thêm một ứng dụng Visual Studio Online giúp người dử dụng có thể thao tác trên nền Web và tương tác với MTM được cài đặt ở máy.
* Người kiểm thử sẽ Tạo các Test Plan trong dự án.
* Kết nối MTM với Team Project Collection
* Người kiểm thử chọn project chuẩn bị kiểm thử.
* Một project được lập ra một số TestPlan. Sau khi truy nhập vào project. Chọn TestPlan muốn test.
* Sau khi kết nối xong. MTM được chia làm hai phần đó là Testing Center và Lab Center.

**CHƯƠNG III: GIỚI THIỆU VỀ SQL**

**SQL là một ngôn ngữ theo chuẩn ANSI để truy xuất các cơ sở dữ liệu.**

* **SQL là gì?**
* SQL là **S**tructured **Q**uery **L**anguage – Ngôn ngữ Truy vấn có Cấu trúc
* SQL cho phép bạn truy xuất một cơ sở dữ liệu
* SQL là một ngữ theo chuẩn ANSI
* SQL có thể thực hiện các truy vấn đến một cơ sở dữ liệu
* SQL có thể truy tìm dữ liệu từ một cơ sở dữ liệu
* SQL có thể chèn các mẩu tin mới vào trong một cơ sở dữ liệu
* SQL có thể xóa các mẩu tin trong một cơ sở dữ liệu
* SQL có thể cập nhật các mẩu tin trong một cơ sở dữ liệu
* SQL rất dễ học

#### SQL là một chuẩn

SQL là một chuẩn ANSI (American National Standards Institute - Viện Tiêu chuẩn Quốc gia Mỹ) cho các hệ thống truy xuất cơ sở dữ liệu. Các phát biểu SQL dùng để truy tìm và cập nhật  dữ liệu trong một cơ sở dữ liệu.

SQL làm việc với các trình quản lý cơ sở dữ liệu như Access, DB2, Informix, Microsoft SQL Server, Oracle, Sybase, và nhiều trình khác (đáng tiếc là đa số trong chúng có các phần mở rộng ngôn ngữ SQL riêng).

#### Các bảng cơ sở dữ liệu

Cơ sở dữ liệu chứa các đối tượng gọi là các **Bảng** (Tables).

Các **Mẩu tin** (Records) lưu trong các bảng này. Các bảng được gọi theo tên bảng (như "Persons", "Orders", "Suppliers"). Các bảng chứa các **Cột** (Columns) và các **Dòng** (Rows) dữ liệu. Dòng chứa các mẩu tin (như mẩu tin về một người). Cột chứa dữ liệu (như First Name, Last Name, Address, và City). Một ví dụ là bảng "Persons" sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **LastName** | **FirstName** | **Address** | **City** |
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Pettersen | Kari | Storgt 20 | Stavanger |
|  |  |  |  |

LastName, FirstName, Address, và City là các cột của bảng. Các dòng chứa ba mẩu tin của 3 người.

#### Các truy vấn SQL

Với SQL, chúng ta có thể **truy vấn**một cơ sở dữ liệu và nhận được một **kết quả** trả về với dạng bảng.  Một truy vấn giống như sau:

SELECT LastName FROM Persons

Sẽ trả về một kết quả giống như sau:

|  |
| --- |
| **LastName** |
| Hansen |
| Svendson |
| Pettersen |

* **Chú ý:**Vài hệ cơ sở dữ liệu cần một dấu “;” ở cuối phát biểu SQL. Chúng ta không dùng dấu “;” trong bài viết này.

#### Thao tác dữ liệu SQL

SQL là một cú pháp để thực hiện các truy vấn. Nhưng ngôn ngữ SQL cũng chứa các cú pháp cập nhật các mẩu tin (record), chèn các mẩu tin mới và xóa các mẩu tin đang tồn tại.

Các lệnh truy vấn và cập nhật này thuộc dạng Ngôn ngữ Thao tác Dữ liệu (Data Manipulation Language - DML) một phần của SQL:

* SELECT – trích dữ liệu từ một cơ sở dữ liệu
* UPDATE – cập nhật dữ liệu trong một cơ sở dữ liệu
* DELETE – xóa dữ liệu từ một cơ sở dữ liệu
* INSERT – chèn dữ liệu mới vào trong một cơ sở dữ liệu

#### Định nghĩa dữ liệu SQL

Ngôn ngữ Định nghĩa Dữ liệu (Data Definition Language - DDL) một phần của SQL, cho phép tạo hay xóa các bảng cơ sở dữ liệu. Chúng ta cũng có thể định nghĩa các chỉ mục (các khóa - key), chỉ định liên kết giữa các bảng, và ràng buột giữa các bảng cơ sở dữ liệu.

Các phát biểu DDL quan trọng nhất trong SQL là:

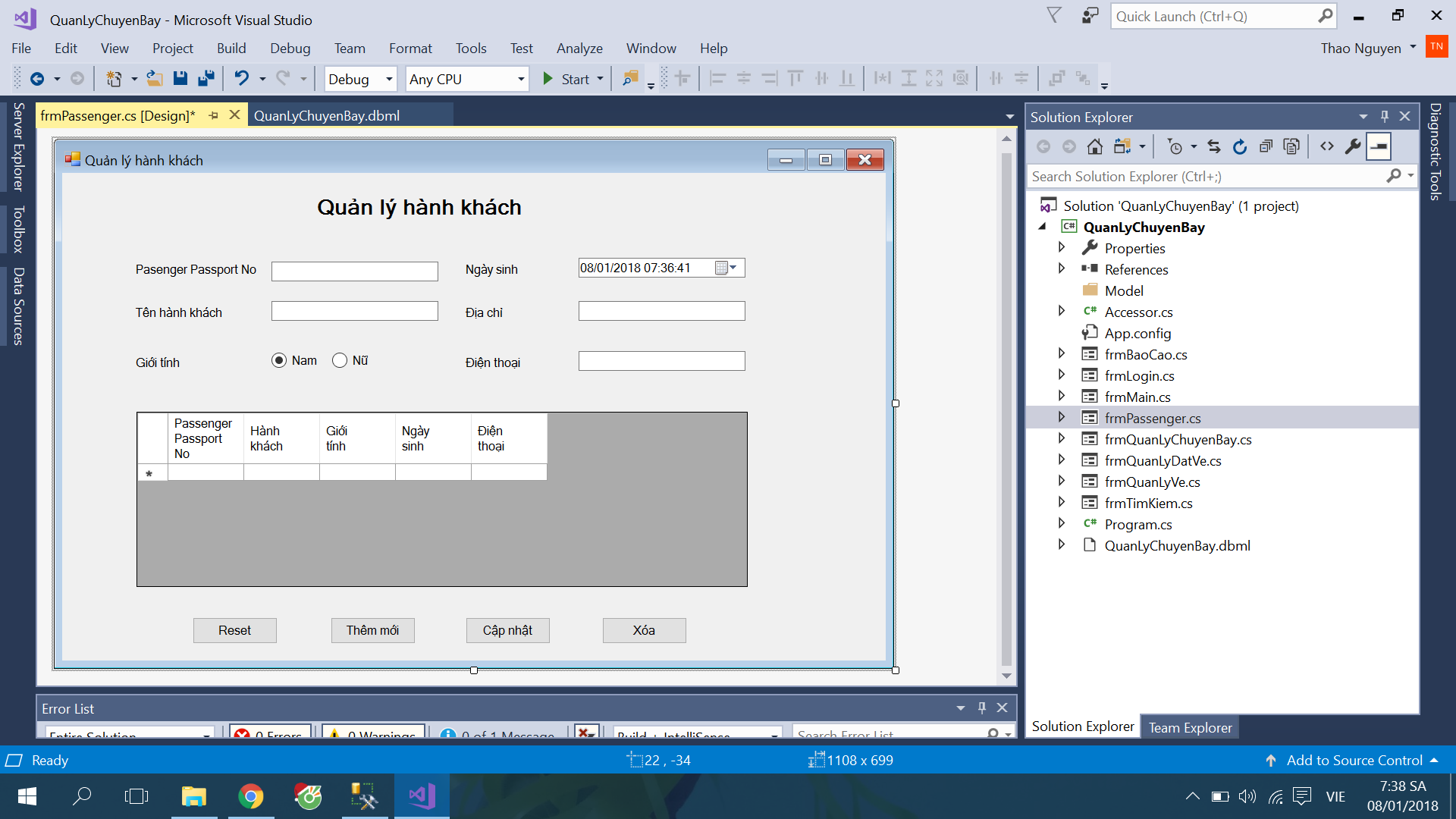
* CREATE TABLE – tạo một bảng cơ sở dữ liệu mới
* ALTER TABLE – thay đổi (alters) một bảng cơ sở dữ liệu
* DROP TABLE – xóa một bảng cơ sở dữ liệu
* CREATE INDEX – tạo một chỉ mục (khóa tìm kiếm)
* DROP INDEX – xoá một chỉ mục

#### SQL và ASP

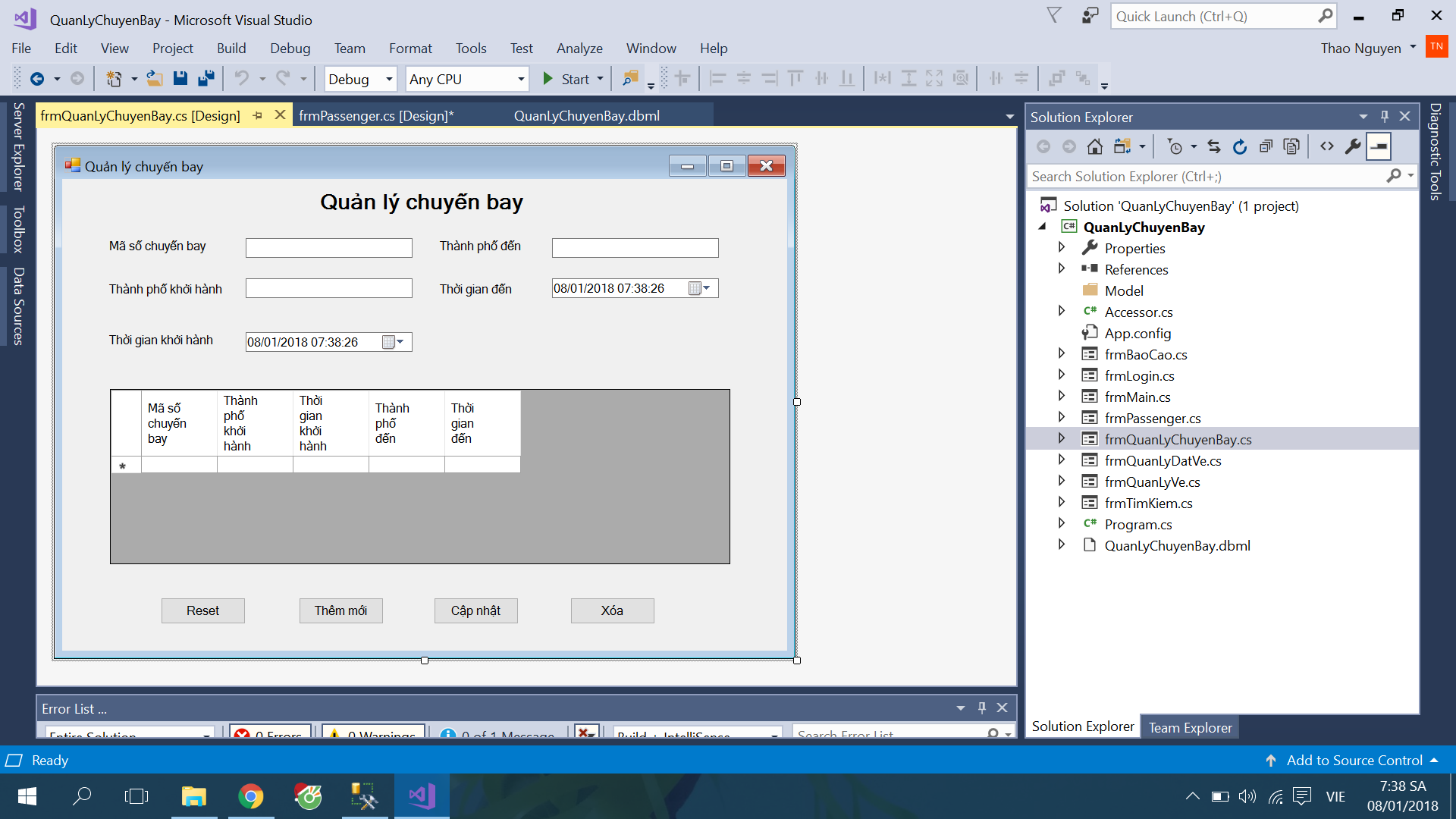
SQL là một phần quan trọng của ASP (Active Server Pages), vì ADO (Active Data Object) được dùng trong ASP để truy xuất cơ  sở dữ liệu, ADO dựa trên SQL để truy xuất dữ liệu.

**CHƯƠNG IV: THÔNG TIN VỀ ĐỒ ÁN**

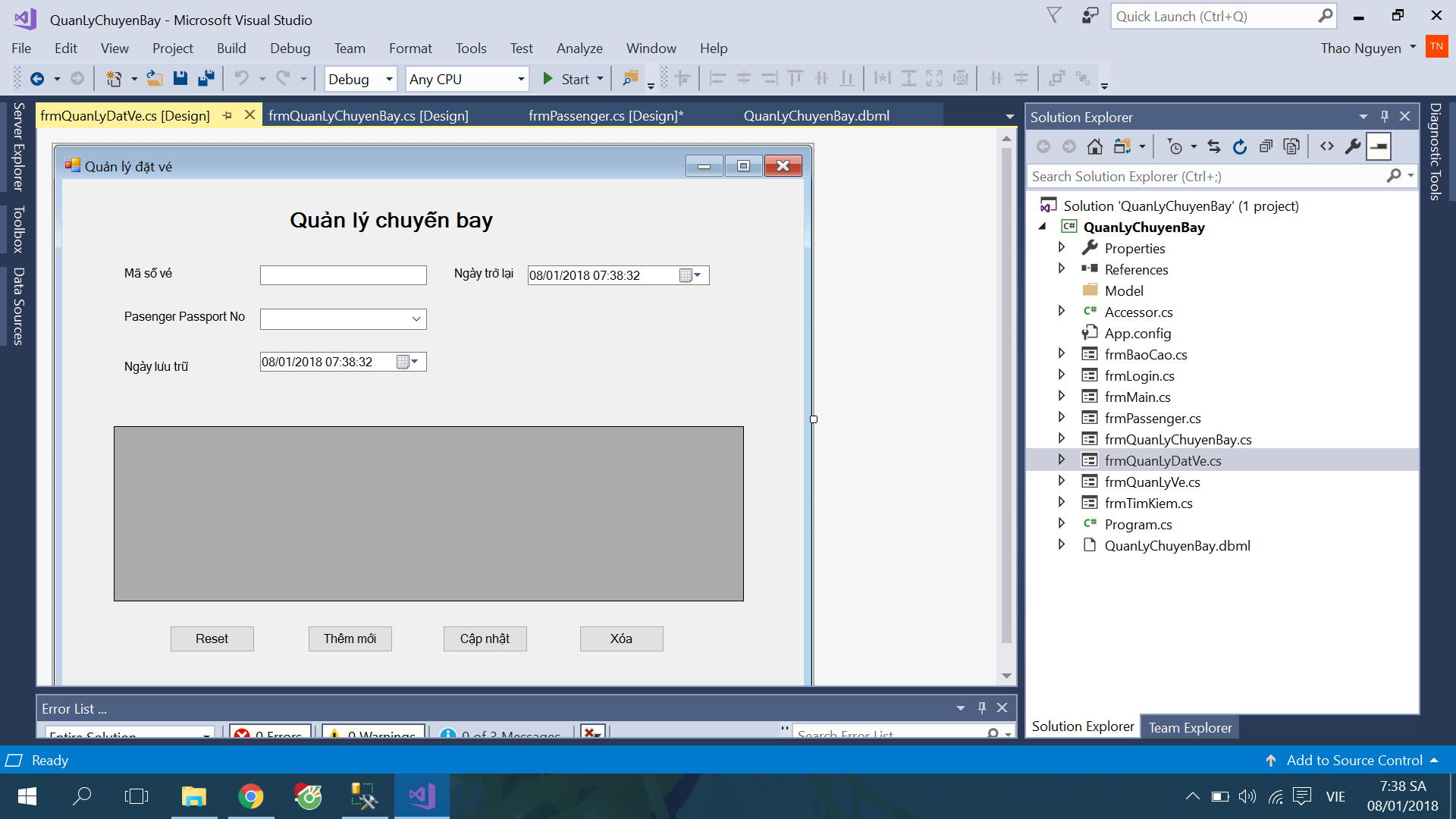
4.1 Giao diện phần mềm



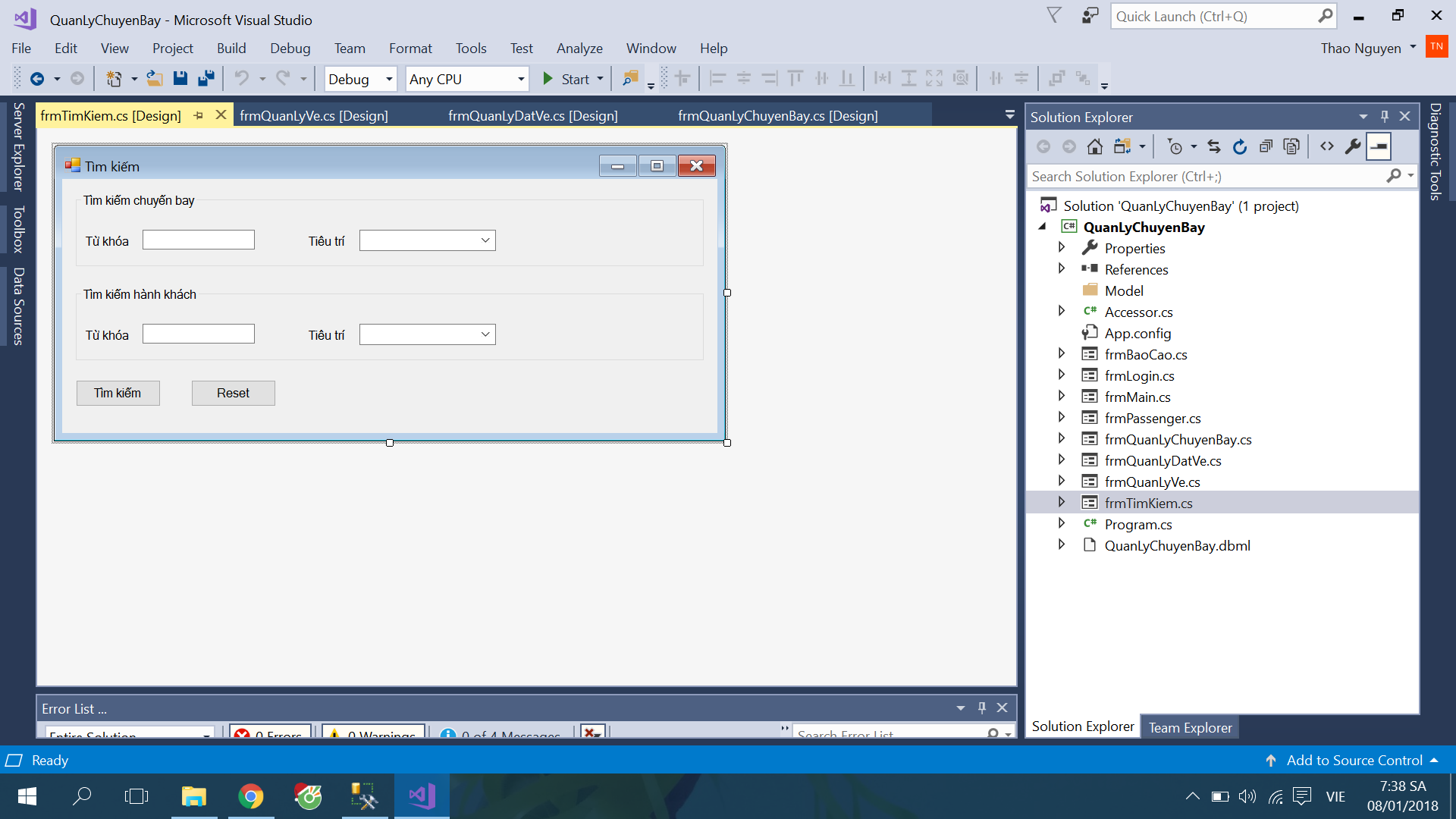
**Giao diện Quản lý hành khách**



**Giao diện Quản lý chuyến bay**



**Giao diện Quản lý đặt vé**



**Giao diện Tìm kiếm**