**PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN**

**CỔNG THÔNG TIN SỰ KIỆN TẠI ĐẠI HỌC CÔNG NGHỆ HUTECH**

1. **GIỚI THIỆU**

Trang Website BigEvent cung cấp các thông tin hội thảo, sự kiện của trường Đại học Công Nghệ Hutech đến những sinh viên của trường một cách đầy đủ - nhanh chóng - chính xác.

1. **ĐẶC TẢ ĐỀ TÀI**

Sự kiện của Website được chia thành 2 loại : sự kiện trong trường và sự kiện ngoài trường.

Các thông tin của sự kiện cần quản lý bao gồm : mã sự kiện, tên sự kiện, loại sự kiện, tên địa điểm, địa chỉ, phòng, thời gian bắt đầu, thời gian kết thúc, ngày bắt đầu, ngày kết thúc, thông tin sự kiện, ảnh sự kiện, ngày tạo sự kiện, tên cơ sở diễn ra, tên bộ phận tổ chức, tên quận/huyện diễn ra.

Một sự kiện được tổ chức bởi một và nhiều bộ phận hoặc không bộ phận nào. Một bộ phận tổ chức được một và nhiều sự kiện hoặc không sự kiện nào.

Một sự kiện được diễn ra ở một cơ sở. Một cơ sở sẽ diễn ra nhiều sự kiện.

Một sự kiện sẽ được diễn ra ở một quận huyện. Một quận huyện sẽ diễn ra nhiều sự kiện.

Admin là tác nhân duy nhất thực hiện được hành vi và bắt buộc phải đăng nhập thực hiện được chức năng.

Dưới đây là một số mô tả về chức năng của Website :

* 1. *Quản lý sự kiện*

Gồm 4 chức năng chính: tạo sự kiện, xem sự kiện, sửa sự kiện, xóa sự kiện.

Xem sự kiện có chức năng tìm kiếm : tên sự kiện, loại sự kiện, tên cơ sở, tên bộ phận, tên quận huyện, thời gian (hôm nay, ngày mai, tuần này)

Xóa sự kiện có hành vi : xác nhận xóa.

Tạo sự kiện :

* Mã sự kiện : thuộc tính bị ẩn. Được tự động tạo khi tạo sự kiện thành công. Thuộc tính bắt buộc.
* Tải ảnh bìa: kích thước tối ưu 1560 - 600px và dung lượng không lớn hơn 1 MB nếu không hợp lệ thì thông báo mời nhập lại. Thuộc tính không bắt buộc nhập.
* Tên sự kiện: tối đa 128 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính bắt buộc nhập.
* Loại sự kiện : chọn 1 trong 2 là “Trong trường” và “Ngoài trường”, nếu chưa chọn thì thông báo mời chọn. Thuộc tính bắt buộc chọn.
* Tên địa điểm : tối đa 50 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính bị ẩn và chỉ hiện khi check ô “Địa chỉ tùy chọn” trên giao diện. Thuộc tính không bắt buộc nhập.
* Địa điểm : tối đa 128 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính bị ẩn và chỉ hiện khi check ô “Địa chỉ tùy chọn” trên giao diện. Thuộc tính không bắt buộc nhập.
* Tên quận huyện : chỉ được chọn một quận huyện. Thuộc tính bị ẩn và chỉ hiện khi check ô “Địa chỉ tùy chọn” trên giao diện. Thuộc tính không bắt buộc nhập.
* Tên cơ sở : chỉ được chọn một cơ sở nếu không chọn thì thông báo mời chọn. Thuộc tính bắt buộc chọn.
* Phòng : tối đa 20 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính không bắt buộc nhập.
* Tên bộ phận : chọn được một và nhiều hoặc không chọn bộ phận. Thuộc tính không bắt buộc chọn.
* Thông tin sự kiện : Ràng buộc chỉ tối đa 4000 kí tự có thể chỉnh sửa, định dạng văn bản nếu vượt quá thì thông báo mời nhập lại. Thuộc tính không bắt buộc nhập
* Ngày bắt đầu - ngày kết thúc : Hiện một danh sách lịch, tháng - năm trên đầu và danh sách ngày ở dưới. Những tháng - năm và ngày đã cũ thì sẽ bị mờ có ý nghĩa là không chọn được. Còn từ tháng - năm, ngày hiện tại và tương lai thì được chọn nhưng chỉ được chọn một cho ngày bắt đầu và ngày kết thúc. Ràng buộc ngày kết thúc chỉ được bằng và lớn hơn ngày bắt đầu nếu sai thông báo mời chọn lại. Định dạng hiển thị dd/mm/yyyy. Thuộc tính bắt buộc nhập
* Thời gian bắt đầu - thời gian kết thúc : Hiện một danh sách giờ, hệ 12 giờ từ 12:00 AM đến 11:45 PM. Chỉ được chọn một cho thuộc tính thời gian bắt đầu và thuộc tính thời gian kết thúc. Ràng buộc giờ kết thúc phải lớn hơn giờ bắt đầu nếu sai thông báo mời chọn lại. Thuộc tính bắt buộc chọn
* Ngày tạo sự kiện : thuộc tính bị ẩn trên giao diện. Tự động lấy ngày hiện tại khi tạo sự kiện thành công. Thuộc tính bắt buộc
  1. *Quản lý bộ phận*

- Gồm 4 chức năng : tạo bộ phận, xem bộ phận, sửa bộ phận, xóa bộ phận.

- Hutech hiện tại 2019 có 11 phòng ban, 13 khoa, 8 viện, 7 trung tâm.

- Các thông tin của bộ phận cần quản lý bao gồm : mã bộ phận, tên bộ phận, logo bộ phận.

- Admin phải đăng nhập để thực hiện chức năng này

- Xóa bộ phận có hành vi : xác nhận xóa.

- Tạo bộ phận :

+ Mã bộ phận : Thuộc tính bị ẩn trên giao diện. Được tự động tạo khi tạo bộ phận thành công. Thuộc tính bắt buộc

+ Tên bộ phận : tối đa 128 kí tự nếu vượt quá thì thông báo nhập lại. Thuộc tính bắt buộc nhập

+ Logo bộ phận :Kích thước tối ưu 130 - 130px nếu không hợp lệ thì mời nhập lại. Thuộc tính không bắt buộc nhập

* 1. *Quản lý cơ sở*

- Gồm 4 chức năng : tạo cơ sở, xem cơ sở, sửa cơ sở, xóa cơ sở

- Admin phải đăng nhập để thực hiện chức năng này.

- Hutech hiện tại 2019 có 5 cơ sở là :

A : 475A Điện Biên Phủ, P25, Q.Bình Thạnh, Tp.HCM

B : 475A Điện Biên Phủ, P25, Q.Bình Thạnh, Tp.HCM

U : 31/36 Ưng Văn Khiêm, P.25, Q.Bình Thạnh, Tp.HCM

E :  Lô E1, Phân khu đào tạo E1, Khu Công Nghệ cao TP.HCM, Phường Hiệp Phú, Quận 9, TP.HCM.

R : Viện Công nghệ Cao Hutech, Lô E2B4, đường D1, Phường Long Thạnh Mỹ, khu Công Nghệ Cao, Quận 9, TP.HCM.

- Các thông tin của cơ sở cần quản lý bao gồm : mã cơ sở, tên cơ sở, địa chỉ.

- Xóa cơ sở có chức năng xác nhận xóa.

- Tạo cơ sở:

+ Mã cơ sở : Thuộc tính bị ẩn trên giao diện. Được tự động tạo khi tạo cơ sở thành công. Thuộc tính bắt buộc

+ Tên cơ sở : tối đa 20 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính bắt buộc nhập

+ Địa chỉ : tối đa 256 kí tự nếu vượt quá thì thông báo mời nhập lại. Thuộc tính không bắt buộc nhập

* 1. *Đăng nhập*

- Các thông tin tài khoản cần quản lý bao gồm : tên đăng nhập, mật khẩu.

* 1. *Danh mục quận huyện*

- Admin phải đăng nhập để thực hiện hành vi này.

- Gồm 4 chức năng chính : tạo quận huyện, xem quận huyện, sửa quận huyện, xóa quận huyện

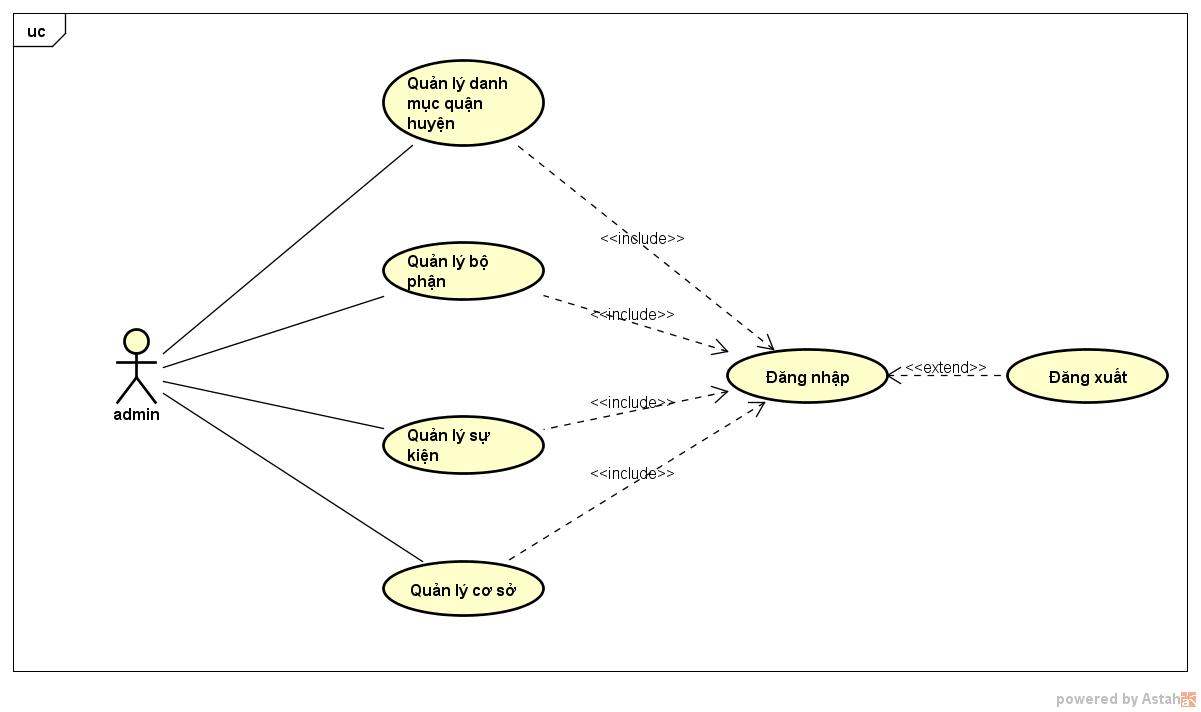
- Các thông tin quận huyện bao gồm : mã quận huyện, tên quận huyện.

- Tạo quận huyện :

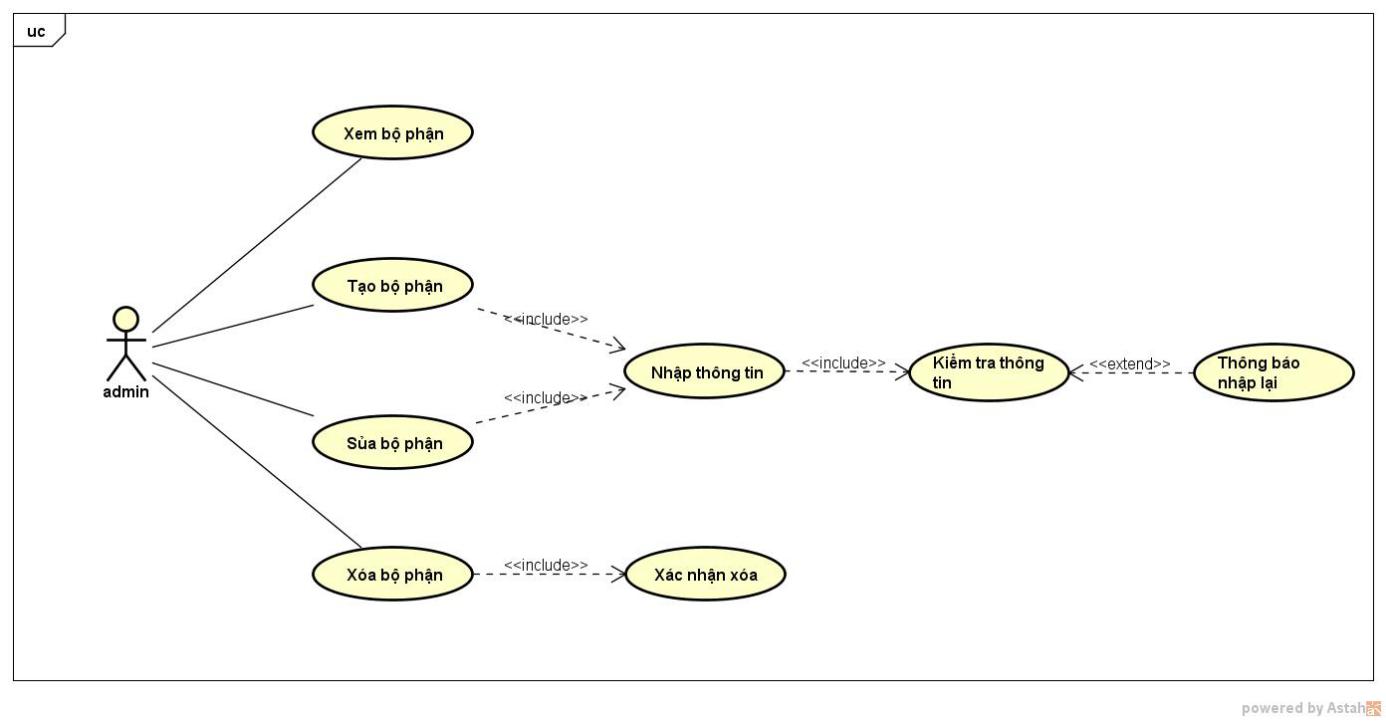
+ Mã quận huyện : thuộc tính bị ẩn trên giao diện. Được tự động tạo khi tạo quận huyện thành công. Thuộc tính bắt buộc

+ Tên quận huyện : tối đa 128 kí tự nếu vượt quá thì thông báo nhập lại. Thuộc tính bắt buộc nhập

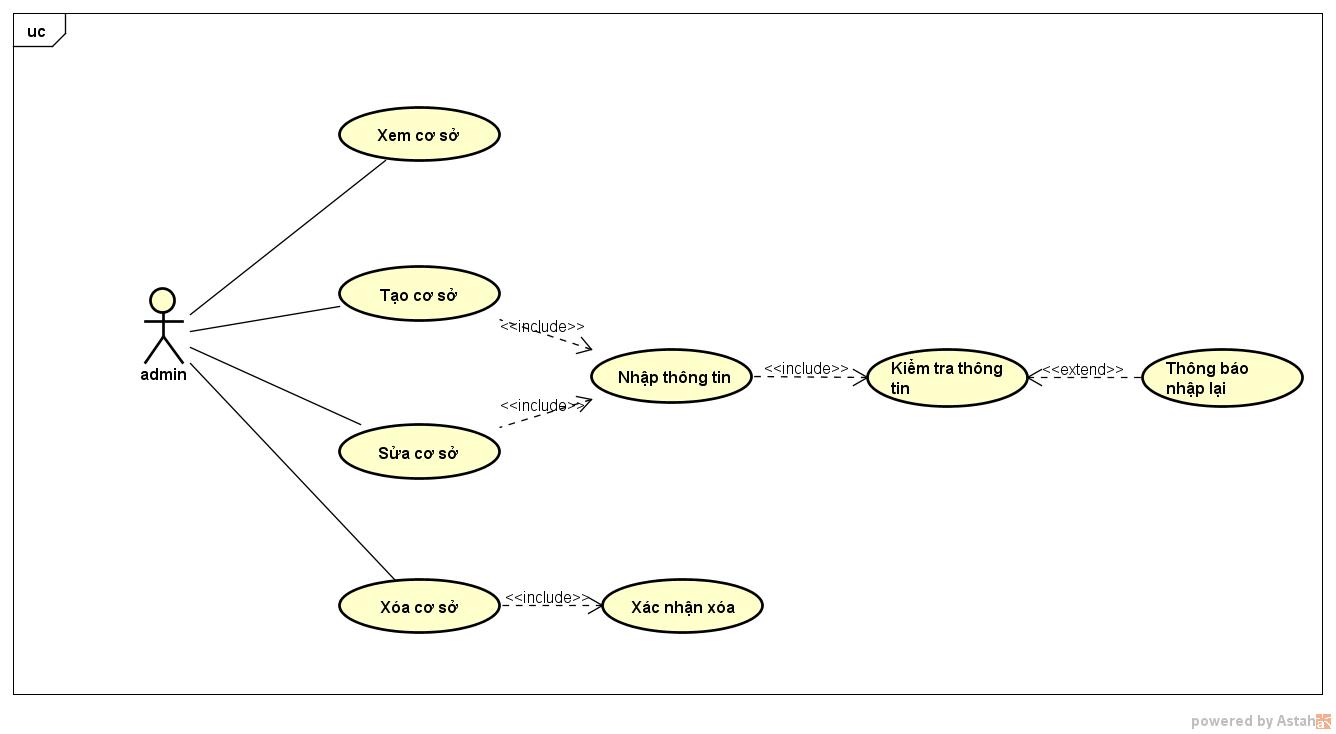
1. **Mô hình**
   1. *Usecase Diagram*
2. Usecase Tổng Thể



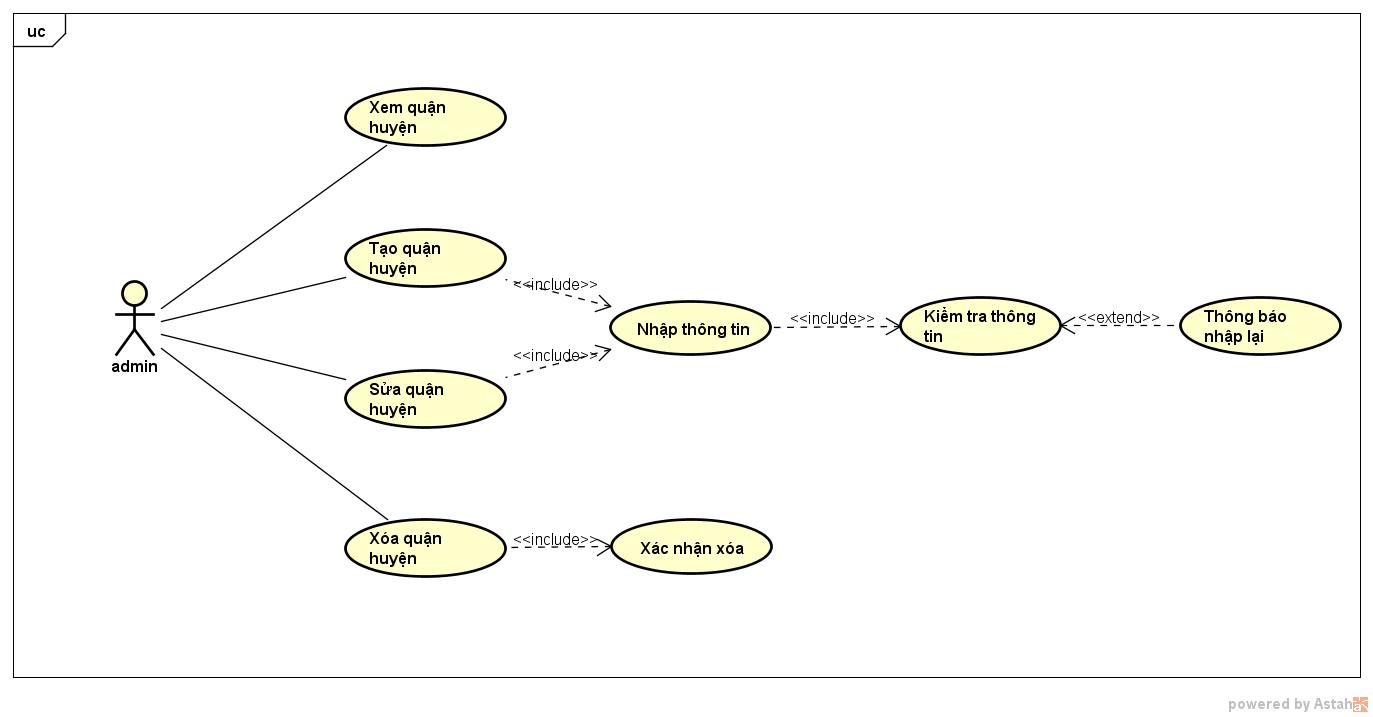
1. Usecase Quản Lý Bộ Phận



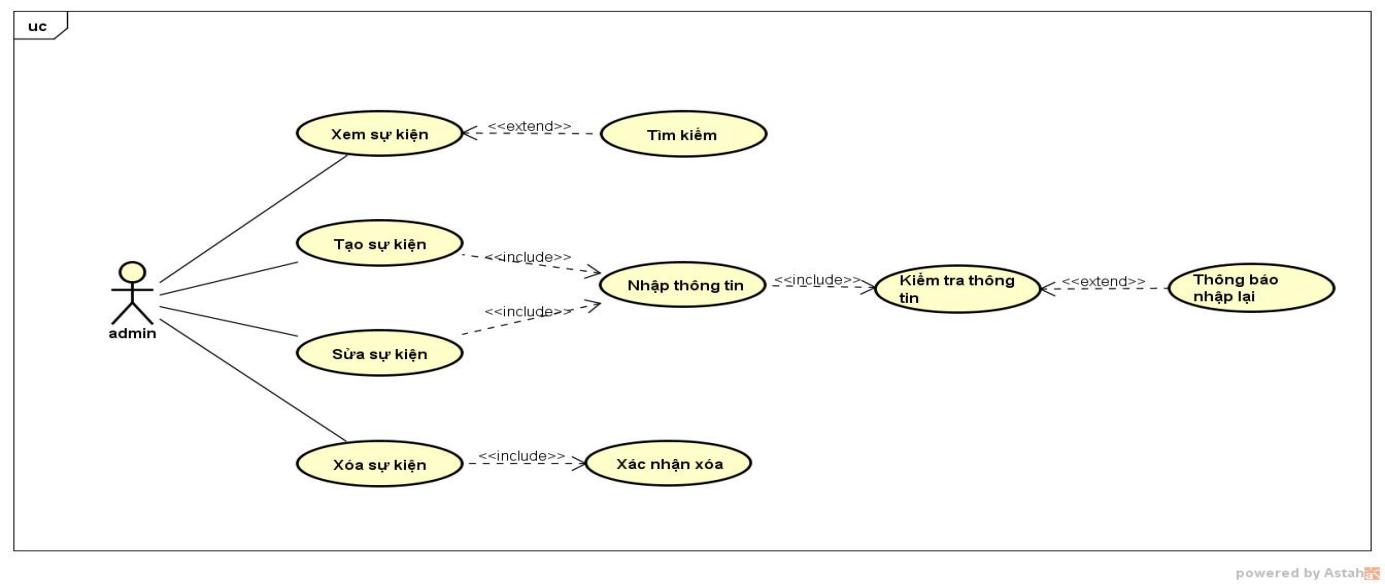
1. Usecase Quản Lý Cơ Sở



1. Usecase Quản Lý Danh Mục Quận Huyện



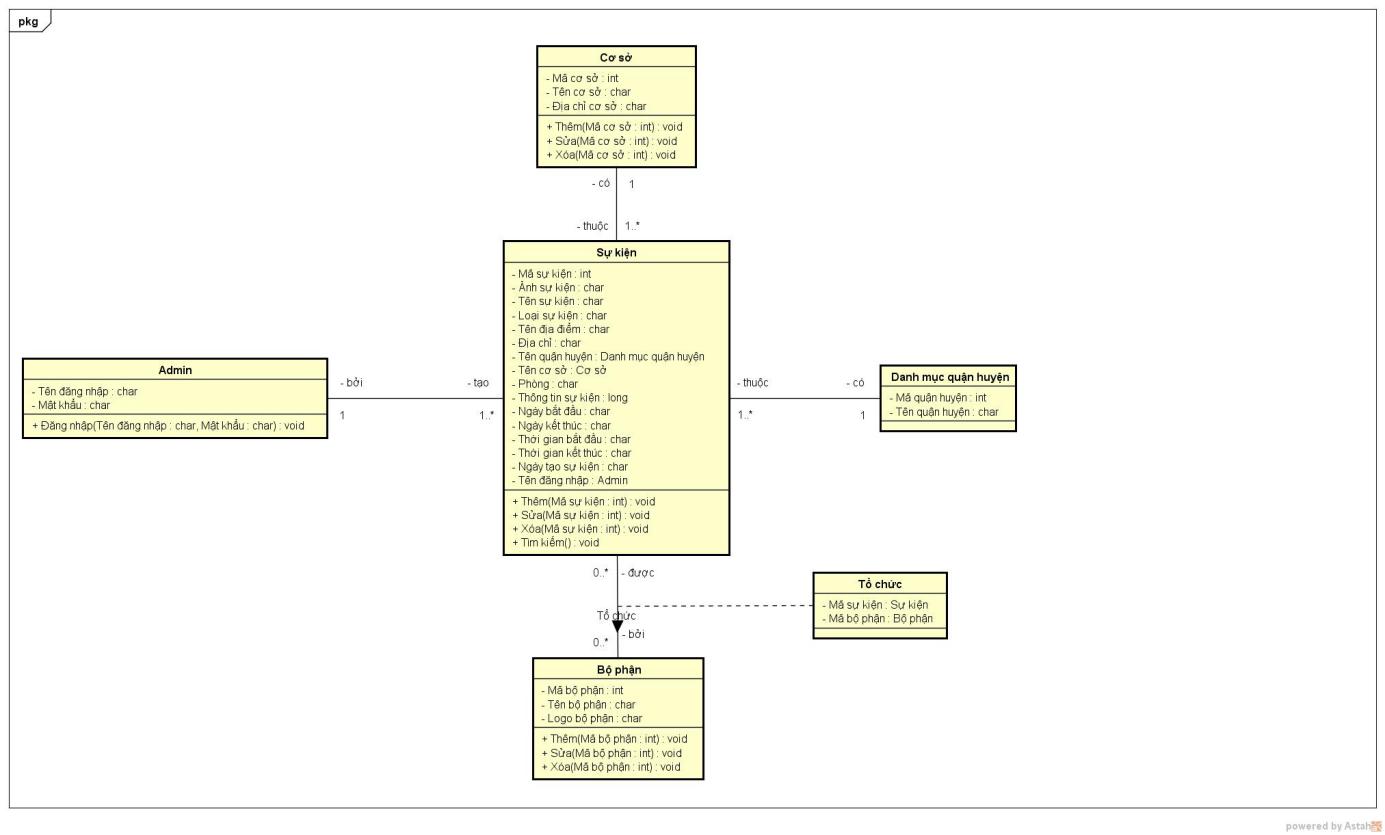
1. Usecase Quản Lý Sự Kiện



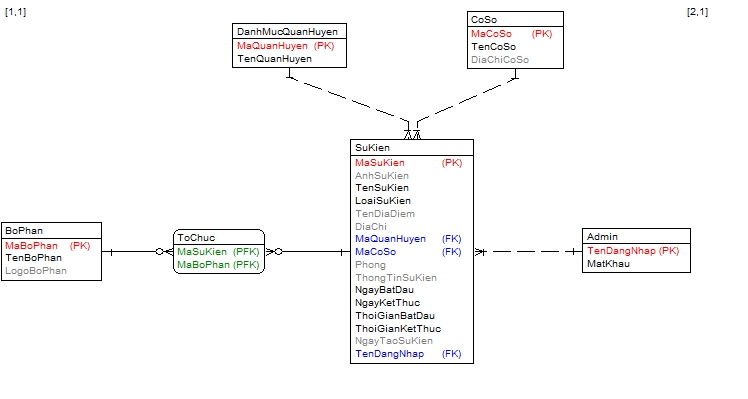
Đặc tả Usecase tạo sự kiện :

|  |  |  |
| --- | --- | --- |
| Tên Use Case: Tạo sự kiện | ID: | Mức quan trọng: Cao |
| Tác nhân chính: Quản trị viên | Loại Use Case: Chi tiết | |
| Người liên quan và công việc quan tâm: Quản trị viên tạo sự kiện | | |
| Mô tả tóm tắt: Use Case này mô tả xử lý tạo sự kiện của quản trị viên | | |
| Ràng buộc: Quản trị viên phải đăng nhập để thực hiện hành vi này.  Loại: | | |
| Các mối quan hệ:   1. Kết hợp: Bộ phận, Cơ sở, Danh mục quận huyện 2. Bao hàm : Tổ chức 3. Mở rộng: Không có 4. Tổng quát hóa : Sự kiện | | |
| Các dòng sự kiện tổng quát chính   1. Hệ thống hiển thị màn hình đăng nhập và quản trị viên chọn chức năng đăng nhập. 2. Quản trị viên nhập thông tin đăng nhập thì thực hiện xử lý đăng nhập 3. Hệ thống hiển thị màn hình quản lý sự kiện và quản trị viên chọn chức năng tạo sự kiện:   - Nếu quản trị viên chọn tạo sự kiện 3.1: Xử lý tạo sự kiện | | |
| Các dòng sự kiện chi tiết chính  3.1: Xử lý tạo sự kiện  3.1.1 Nhập thông tin  3.1.2 Kiểm tra thông tin đã nhập  3.1.3 Hiển thị thông báo tạo sự kiện thành công | | |
| Các dòng sự kiện ngoại lệ  3.1.2 Xử lý nhập thông tin sai tạo sự kiện  3.1.2.1 Hiển thị thông báo lỗi và yêu cầu nhập lại  3.1.2.2 Quay lại bước 3.1.1 | | |

1. *Class Diagram*

**

1. *ERD*

**

1. **DỰ ĐỊNH LÀM TUẦN 2**

Tâm : Design giao diện trang quản trị Admin (Xem sự kiện), giao diện đăng nhập.

Đệ : Thiết kế lại kiểu dữ liệu thuộc tính phần thời gian, viết đặc tả mô hình ERD, test dữ liệu bằng lệnh SQL, cài đặt hệ thống lên VisualStudio 2017, code chức năng đăng nhập.

**TUẦN 2**

1. **Tạo CSDL và các bảng trong CSDL BigEvent**
2. *Tạo CSDL*

|  |
| --- |
| CREATE DATABASE [BigEvent7]  ON  PRIMARY (NAME = BigEvent7\_DATA,  FILENAME = 'D:\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Data\BigEvent7.MDF' ,  SIZE = 50MB ,  MAXSIZE = 200MB ,  FILEGROWTH = 10MB)  LOG ON (NAME = BigEvent7\_LOG ,  FILENAME = 'D:\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Data\BigEvent7.LDF' ,  SIZE = 10MB ,  FILEGROWTH = 5MB)  GO  --BACKUP DATABASE  BACKUP DATABASE [BigEvent7]  TO DISK = 'D:\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup\BigEvent7.BAK'  DROP TABLE [CoSo]  DROP TABLE [SuKien]  DROP TABLE [Admin]  DROP TABLE [BoPhan]  DROP TABLE [ToChuc]  DROP TABLE [DanhMucQuanHuyen]  DROP DATABASE [BigEvent7]  USE [BigEvent7]  GO |

1. *Bảng Sự Kiện : SuKien*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |  |
| **MaSuKien** | BigInt(Auto) | NotNull |
| AnhSuKien | Image |  |
| TenSuKien | NVarchar(128) | NotNull |
| LoaiSuKien | NVarchar(20) | NotNull |
| TenDiaDiem | NVarchar(50) |  |
| DiaChi | NVarchar(128) |  |
| *MaQuanHuyen* | TinyInt(Auto) | NotNull |
| *MaCoSo* | TinyInt(Auto) | NotNull |
| Phong | NVarchar(20) |  |
| ThongTinSuKien | NVarchar(4000) |  |
| NgayBatDau | Date | NotNull |
| NgayKetThuc | Date | NotNull |
| ThoiGianBatDau | DateTime | NotNull |
| ThoiGianKetThuc | DateTime | NotNull |
| NgayTaoSuKien | DateTime | NotNull |
| *TenDangNhap* | Nvarchar(128) | NotNull |
| Create table [SuKien]  (  [MaSuKien] Bigint Identity(1,1) NOT NULL,  [AnhSuKien] Image NULL,  [TenSuKien] Nvarchar(128) NOT NULL,  [LoaiSuKien] Nvarchar(20) NOT NULL,  [TenDiaDiem] Nvarchar(50) NULL,  [DiaChi] Nvarchar(128) NULL,  [MaQuanHuyen] Tinyint NOT NULL,  [MaCoSo] Tinyint NOT NULL,  [Phong] Nvarchar(20) NULL,  [ThongTinSuKien] Nvarchar(max) NULL,  [NgayBatDau] Date NOT NULL,  [NgayKetThuc] Date NOT NULL,  [ThoiGianBatDau] Datetime NOT NULL,  [ThoiGianKetThuc] Datetime NOT NULL,  [NgayTaoSuKien] Datetime NOT NULL,  [TenDangNhap] Nvarchar(20) NOT NULL,  Primary Key ([MaSuKien])  )  Go  --Tạo ràng buộc bảng sự kiện  ALTER TABLE [SuKien] ADD  CONSTRAINT CK\_NBD CHECK([NgayBatDau] >= GETDATE()),  CONSTRAINT CK\_TG CHECK([ThoiGianKetThuc] > [ThoiGianBatDau]),  CONSTRAINT CK\_N CHECK([NgayKetThuc] >= [NgayBatDau]),  CONSTRAINT CK\_LSK CHECK([LoaiSuKien] IN (N'Trong trường',N'Ngoài trường')),  DEFAULT GETDATE() FOR [NgayTaoSuKien],  DEFAULT (1) FOR [MaCoSo],  DEFAULT (1) FOR [MaQuanHuyen]  GO  --Tạo khóa ngoại  Alter table [SuKien] add foreign key([MaCoSo]) references [CoSo] ([MaCoSo]) on update cascade on delete set default  go  Alter table [SuKien] add foreign key([TenDangNhap]) references [Admin] ([TenDangNhap]) on update no action on delete no action  go  Alter table [SuKien] add foreign key([MaQuanHuyen]) references [DanhMucQuanHuyen] ([MaQuanHuyen]) on update cascade on delete set default  go | | | |

1. *Bảng Cơ Sở : CoSo*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |  |
| **MaCoSo** | TinyInt(Auto) | NotNull |
| TenCoSo | NVarchar(20) | NotNull |
| DiaChiCoSo | NVarchar(256) |  |
| Create table [CoSo]  (  [MaCoSo] Tinyint Identity(1,1) NOT NULL,  [TenCoSo] Nvarchar(20) NOT NULL,  [DiaChiCoSo] Nvarchar(256) NULL,  Primary Key ([MaCoSo])  )  go | | | |

1. *Bảng Bộ Phận : BoPhan*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |  |
| **MaBoPhan** | TinyInt(Auto) | NotNull |
| TenBoPhan | NVarchar(256) | NotNull |
| LogoBoPhan | Image |  |  |
| Create table [BoPhan]  (  [MaBoPhan] Tinyint Identity(1,1) NOT NULL,  [TenBoPhan] Nvarchar(128) NOT NULL,  [LogoBoPhan] Image NULL,  Primary Key ([MaBoPhan])  )  Go | | | |

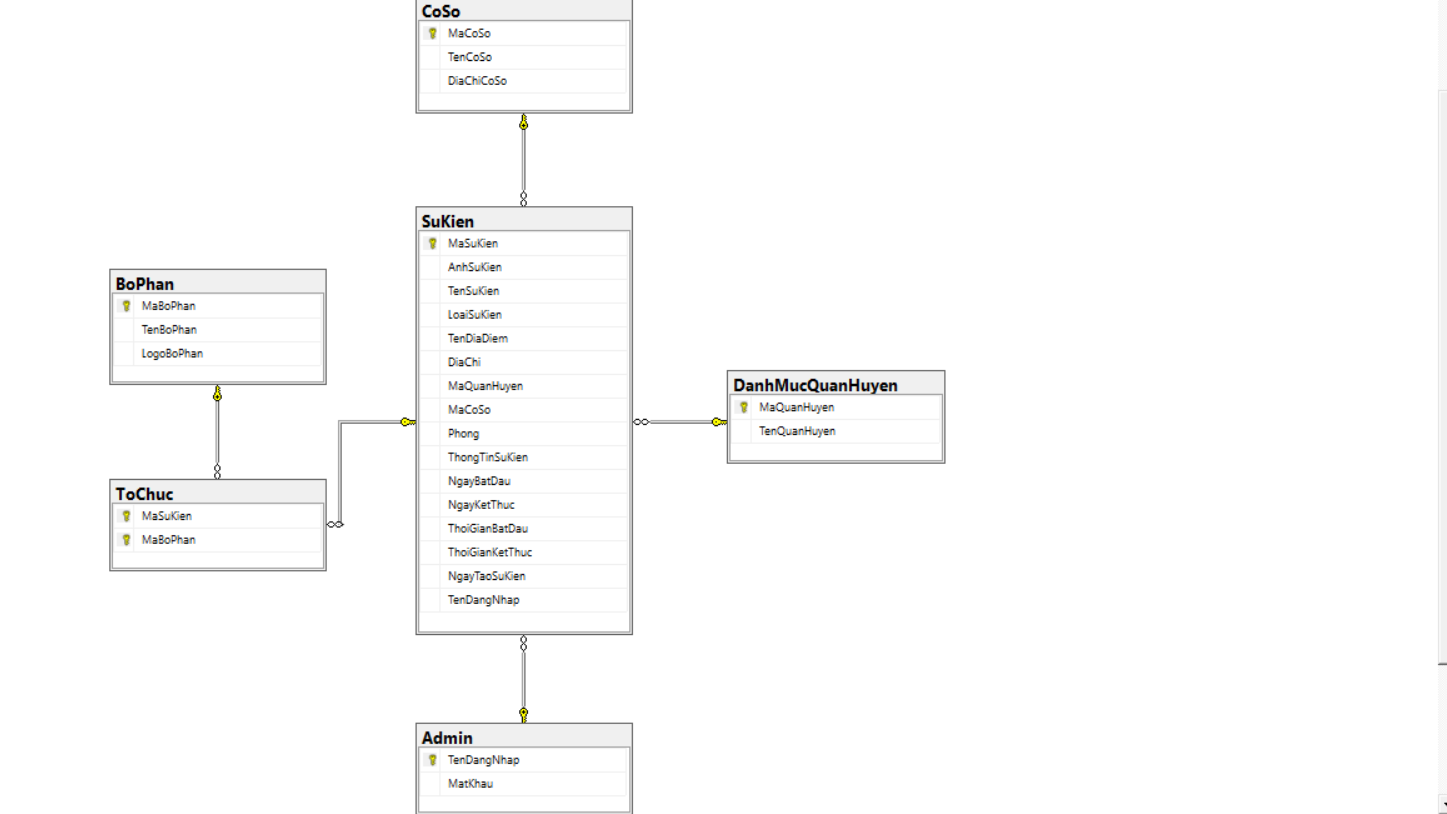
1. *Bảng Tổ Chức tạo ra bởi bảng Bộ Phận và Sự Kiện : ToChuc*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |  |
| *MaSuKien* | BigInt(Auto) | NotNull |
| *MaBoPhan* | TinyInt(Auto) | NotNull |
| Create table [ToChuc]  (  [MaSuKien] Bigint NOT NULL,  [MaBoPhan] Tinyint NOT NULL,  Primary Key ([MaSuKien],[MaBoPhan])  )  Go  --Tạo ràng buộc bảng tổ chức  ALTER TABLE [ToChuc] ADD  DEFAULT (1) FOR [MaBoPhan]  GO  --Tạo khóa ngoại  Alter table [ToChuc] add foreign key([MaSuKien]) references [SuKien] ([MaSuKien]) on update cascade on delete cascade  go  Alter table [ToChuc] add foreign key([MaBoPhan]) references [BoPhan] ([MaBoPhan]) on update cascade on delete set default  go | | | |

1. *Bảng Admin : Admin*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ràng buộc |  |
| **TenDangNhap** | Nvarchar(128) | NotNull |
| MatKhau | Nvarchar(128) | NotNull |
| Create table [Admin]  (  [TenDangNhap] Nvarchar(20) NOT NULL,  [MatKhau] Nvarchar(50) NOT NULL,  Primary Key ([TenDangNhap])  )  go | | | |

1. *ERD Diagram của dữ liệu BigEvent*

**

1. *Lê Khôi Đệ*

Bài toán 1 : Khi nạp dữ liệu bảng cha Sự Kiện thì có thể nạp được ở bảng con Trong Trường và Ngoài Trường cùng một mã sự kiện. Làm sao để phân biệt được mã sự kiện này chỉ nằm trong 1 loại là Trong Trường hoặc là Ngoài Trường. Hướng giải quyết viết ràng buộc

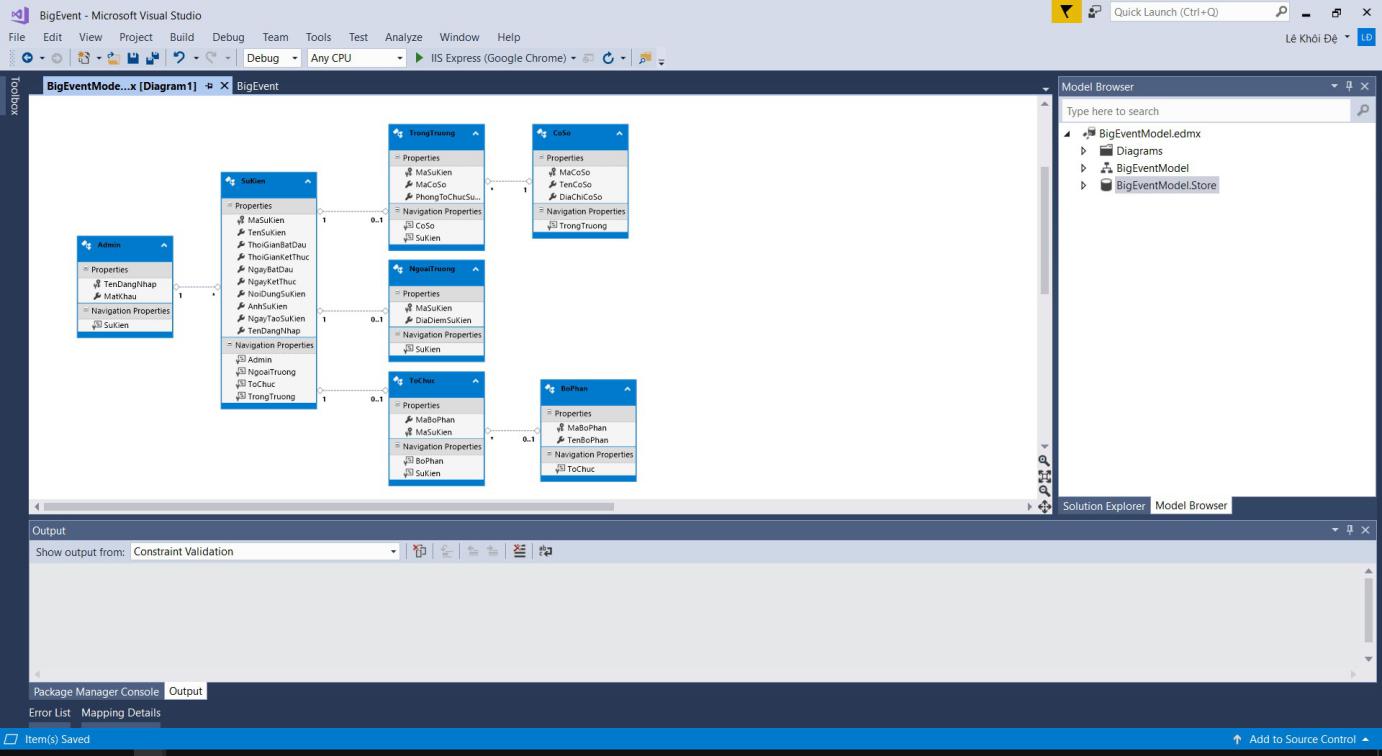
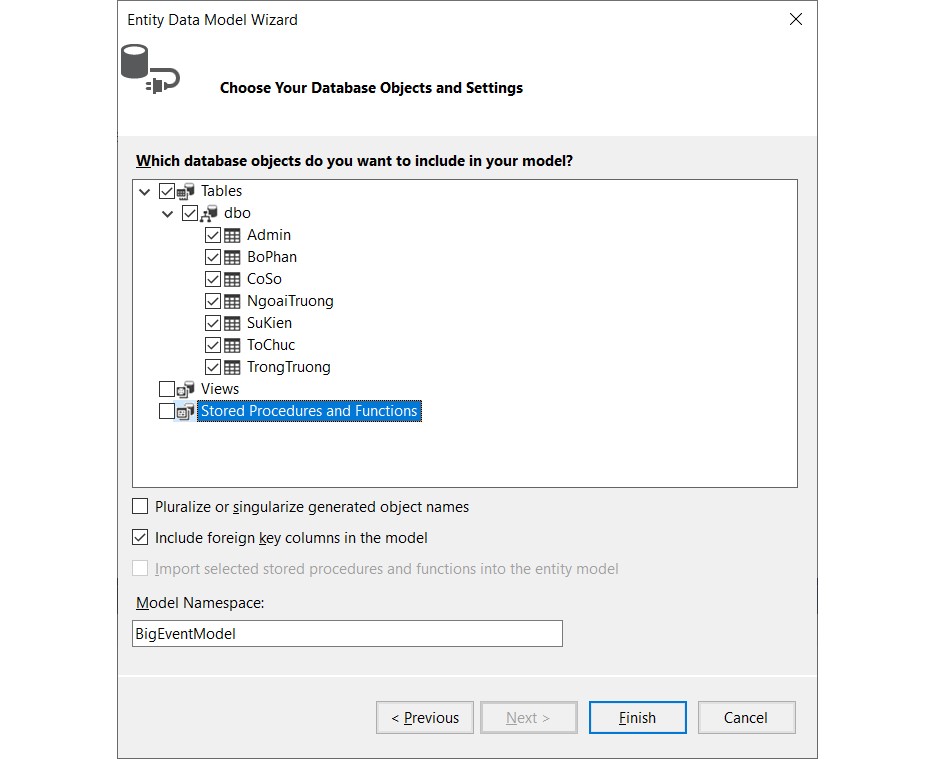
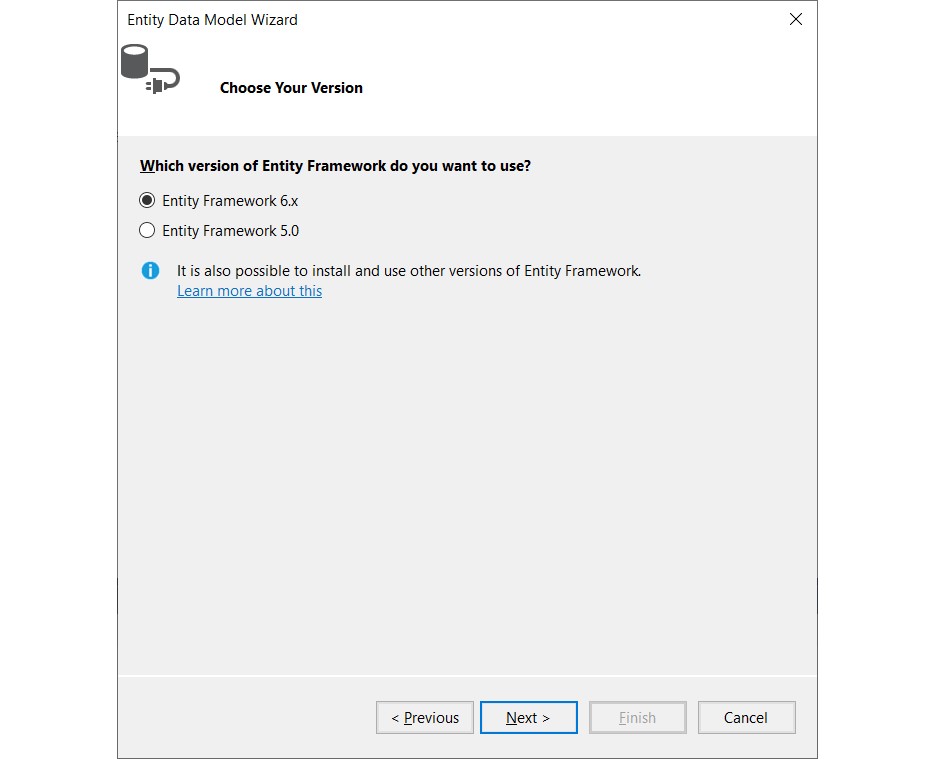
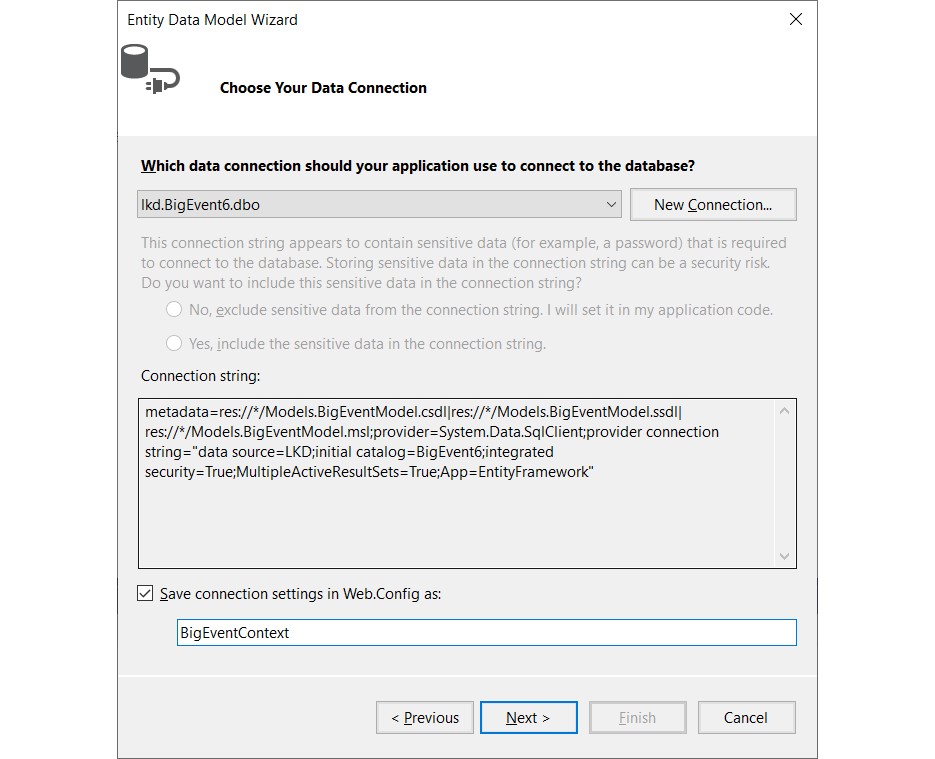
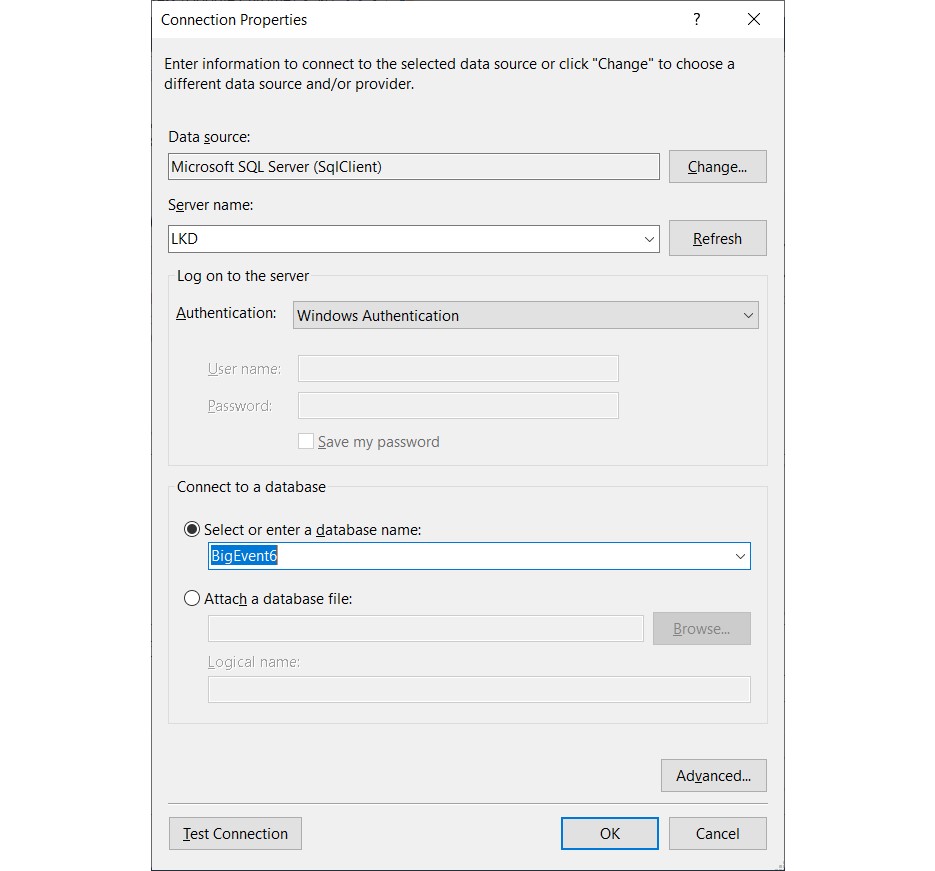
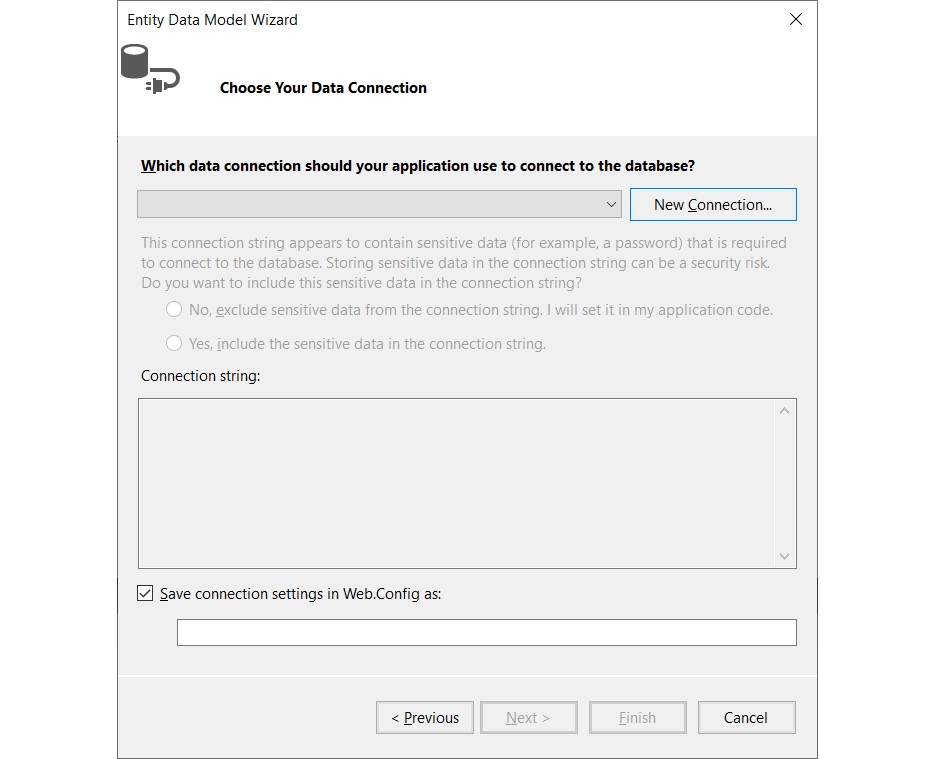
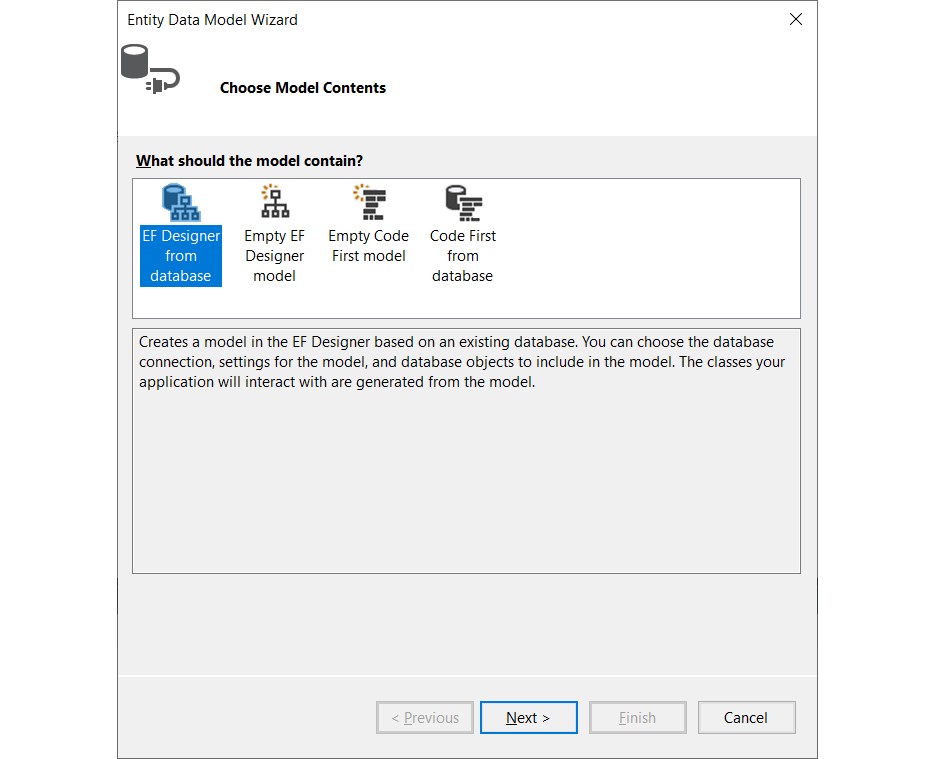
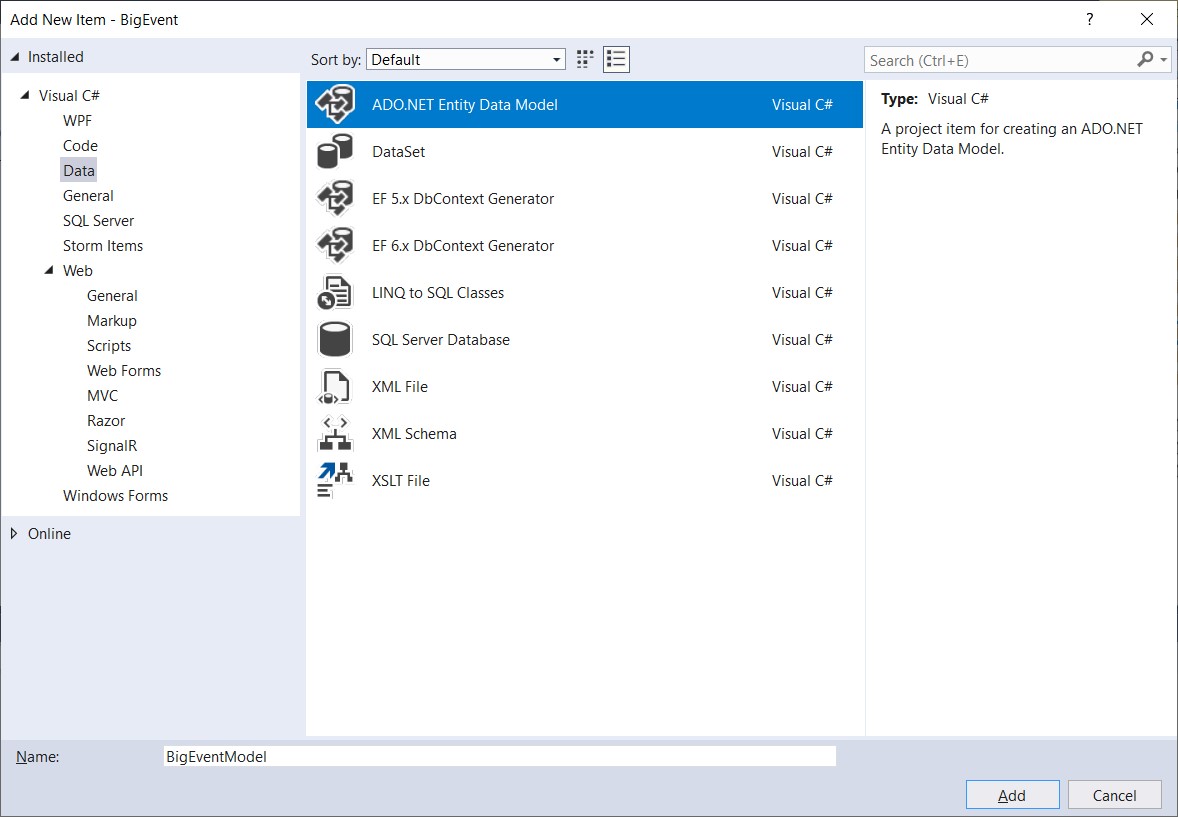
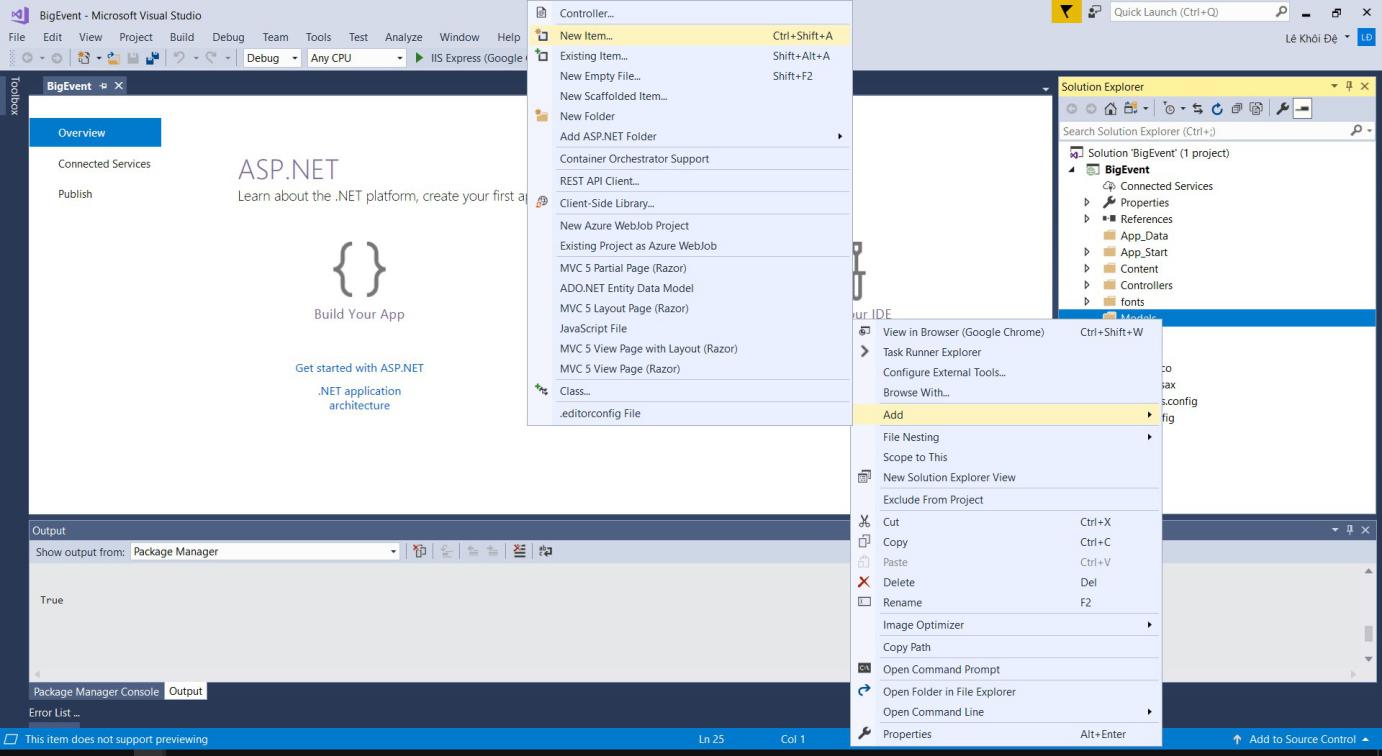
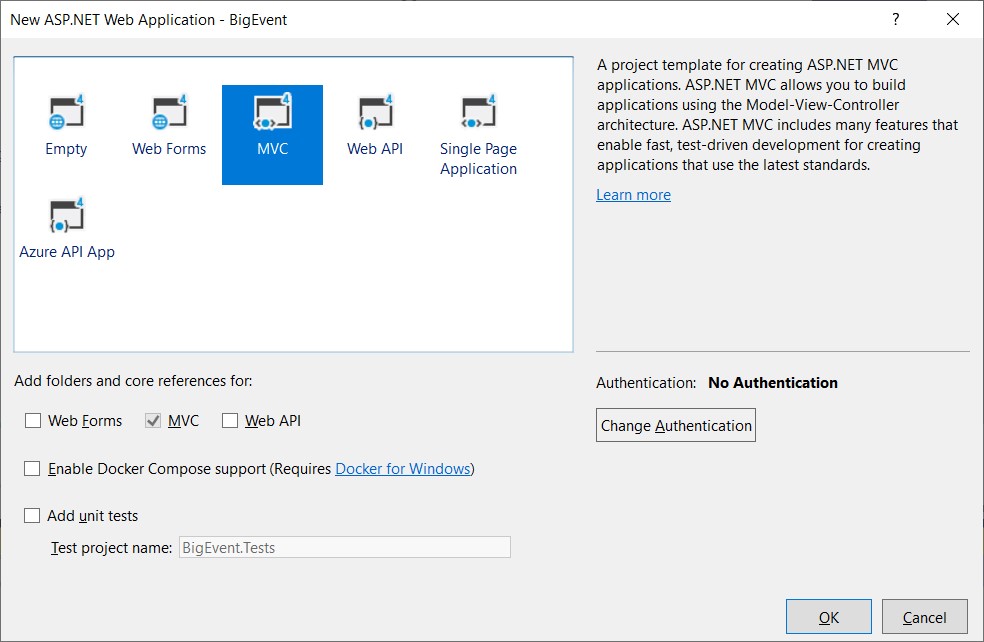
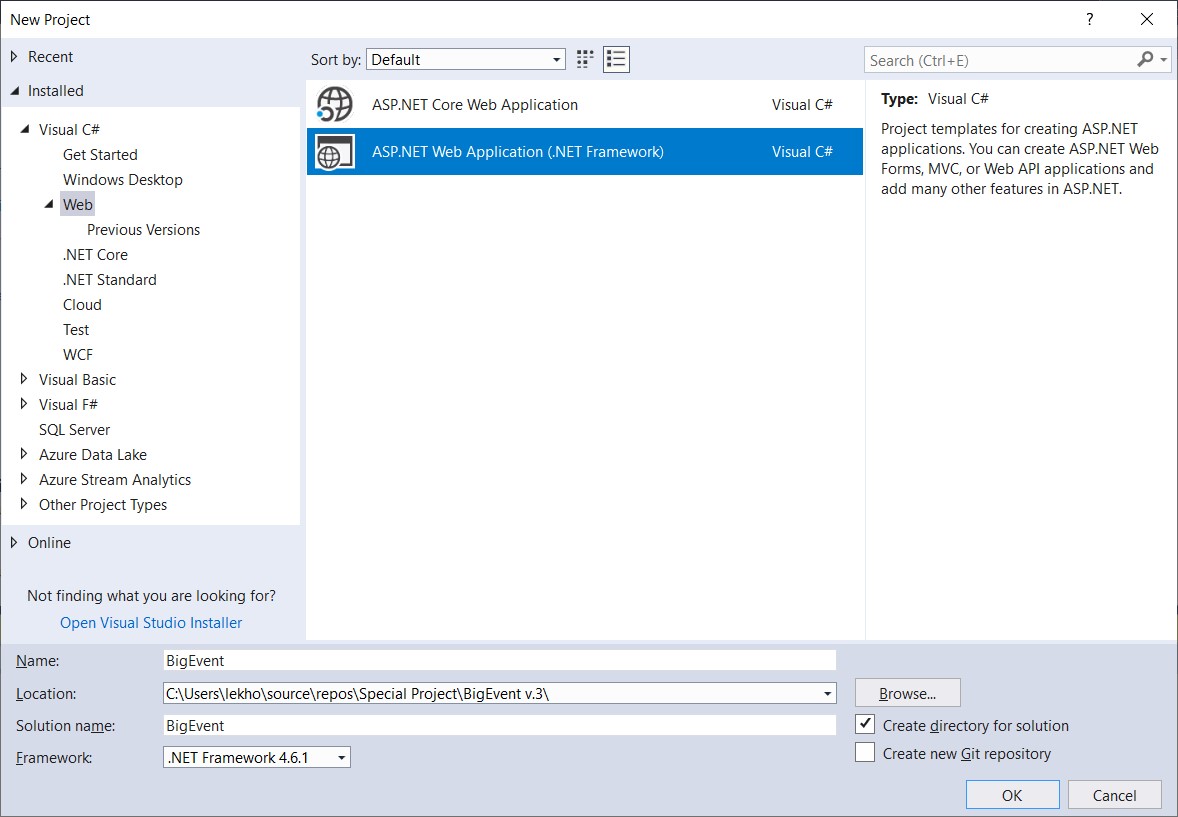
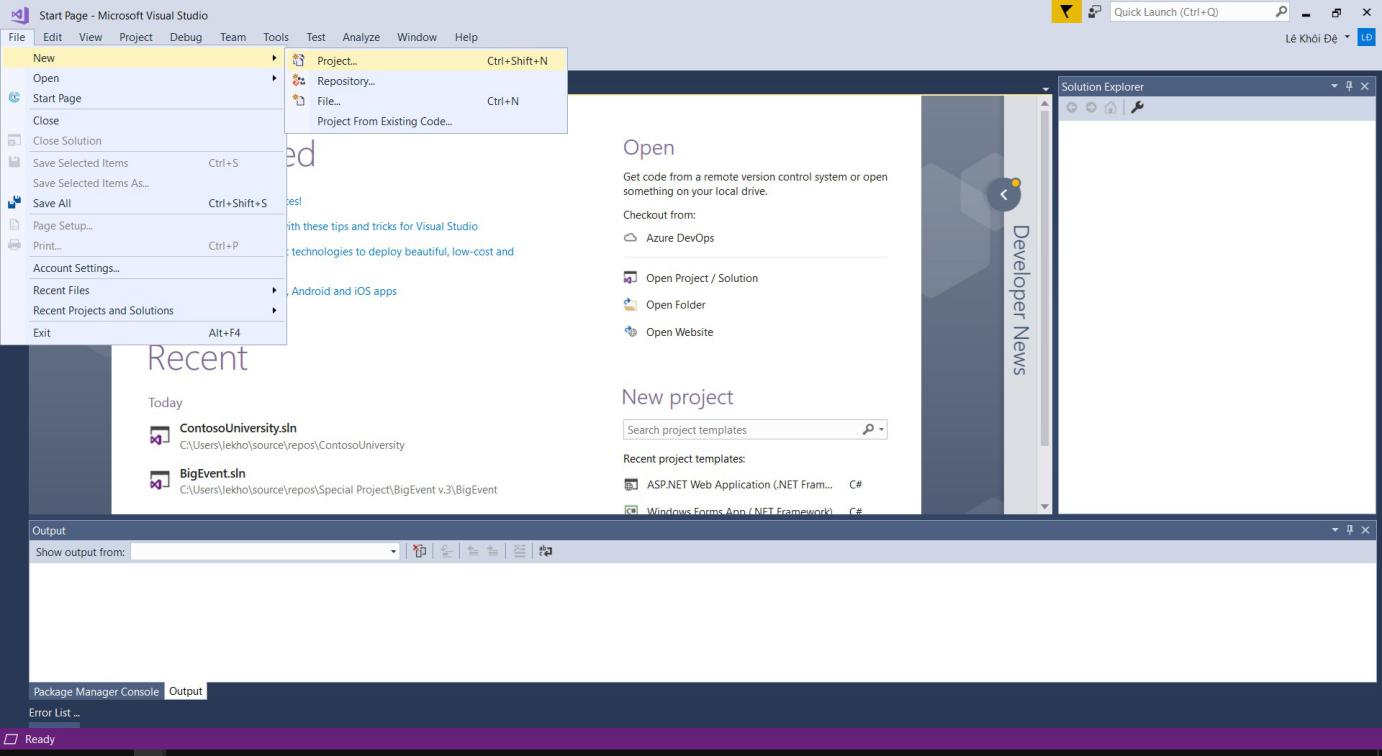
Bài toán 2 : Thuộc tính Ngày Tạo Sự Kiện trong bảng Sự Kiện khi nạp một dữ liệu làm sao để tự động chọn ngày và giờ hiện tại. Hướng giải quyết tìm hiểu SQL

To do list

1. Chưa test các chức năng Update và Delete bằng câu lệnh SQL
2. Chưa đưa lên hệ thống Visual Studio
3. Chưa code chức năng đăng nhập

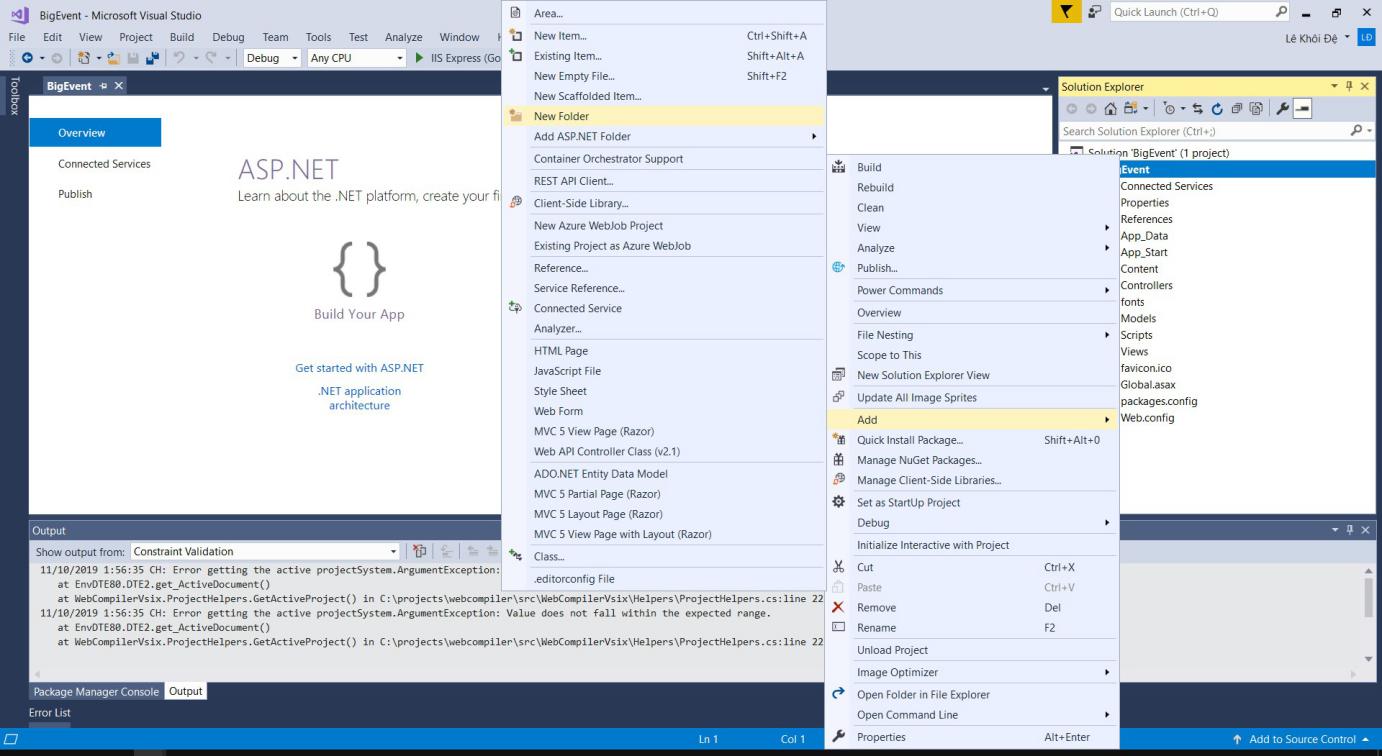
**TUẦN 3**

1. **CÀI ĐẶT HỆ THỐNG**

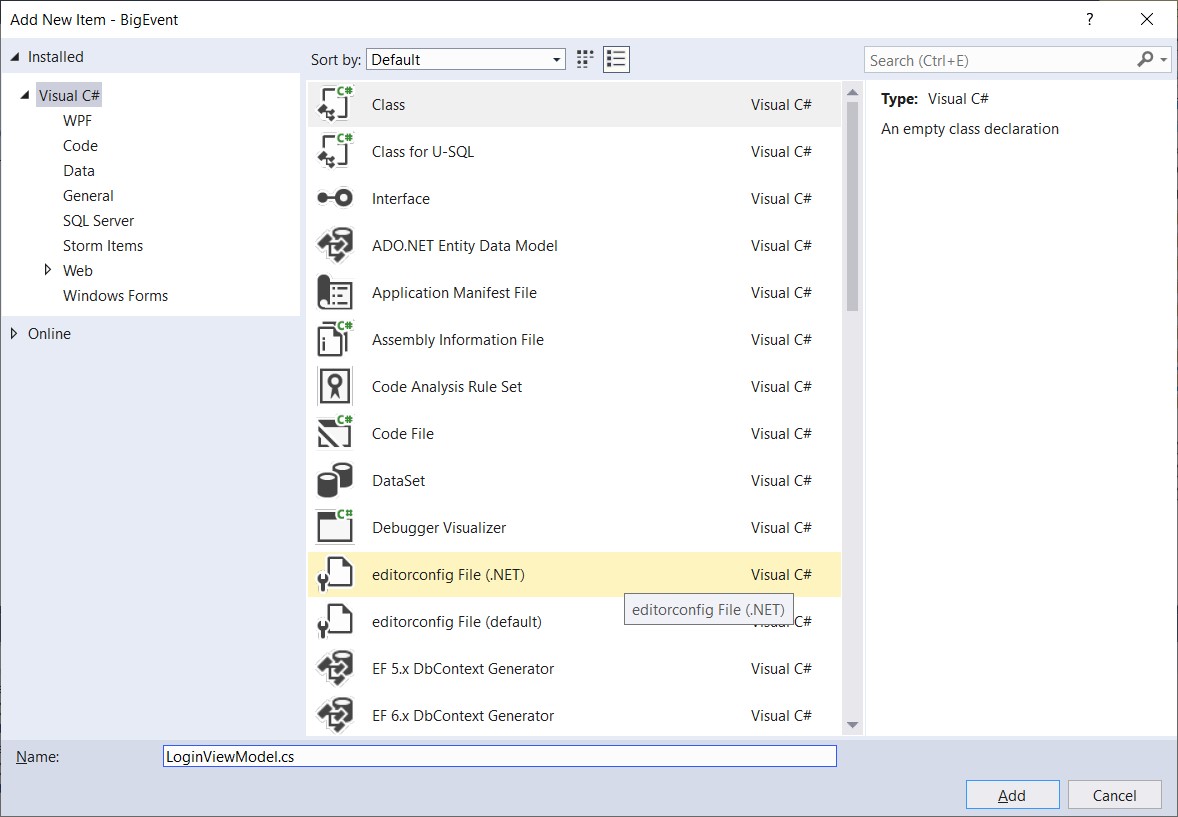


1. **HỒ SƠ THIẾT KẾ**
   1. *Chức năng đăng nhập Admin*

- Tạo Folder ViewModel



- Tạo lớp “LoginViewModel.cs”

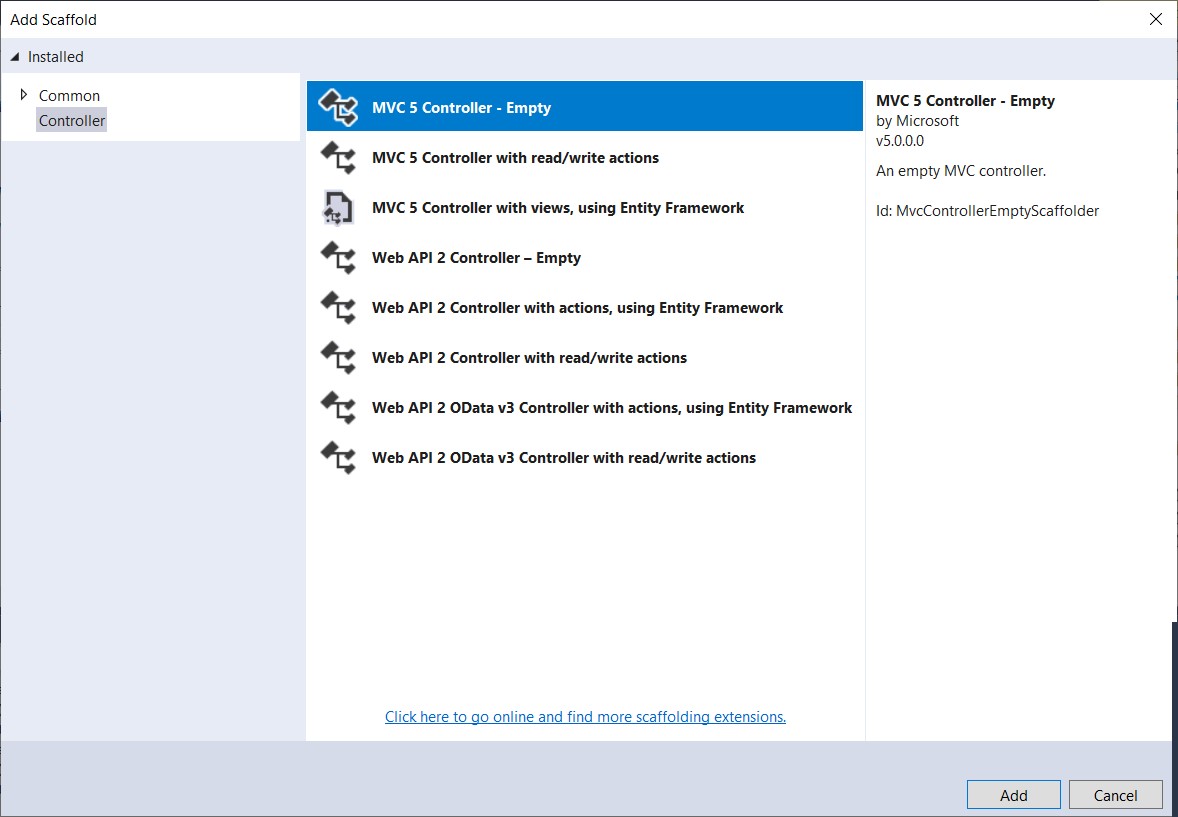
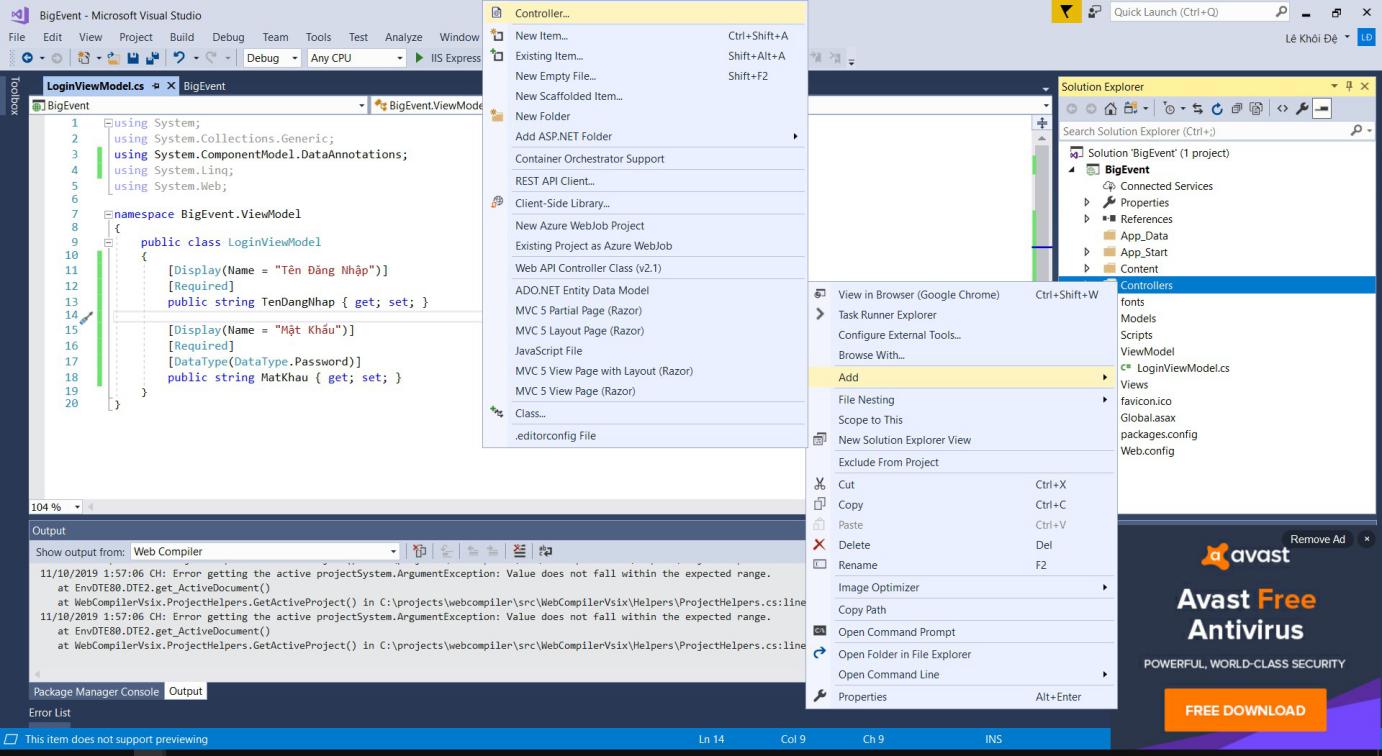


- Code của lớp “LoginViewModel.cs”

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  namespace BigEvent.ViewModel  {      public class LoginViewModel      {          [Display(Name = "Tên Đăng Nhập")]          [Required]          public string TenDangNhap { get; set; }            [Display(Name = "Mật Khẩu")]          [Required]          [DataType(DataType.Password)]          public string MatKhau { get; set; }      }  } |

Sử dụng mô hình kiến trúc MVVM (Model - View - ViewModel) để dễ dàng thực hiện data binding, command.

- Tạo Controller “AdminController”



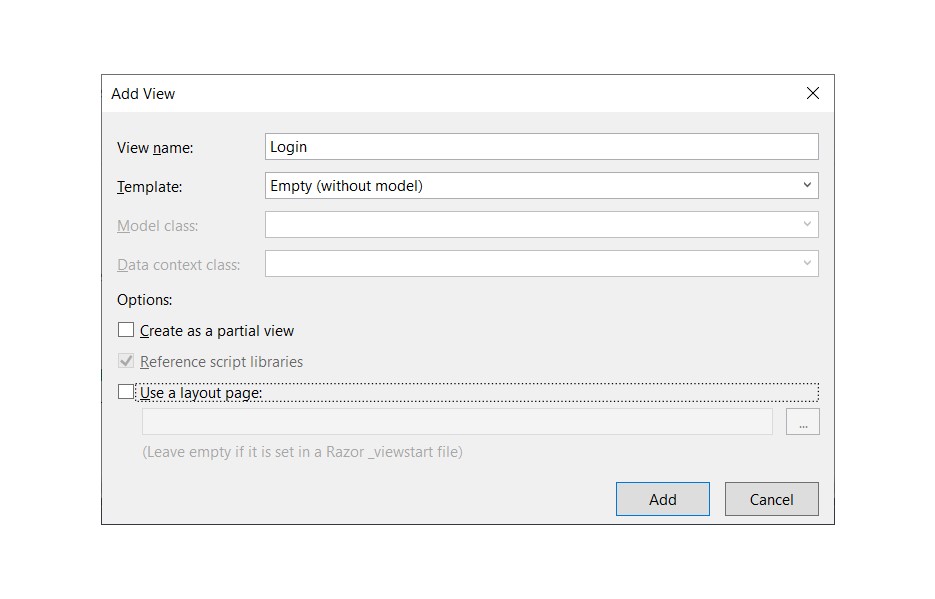
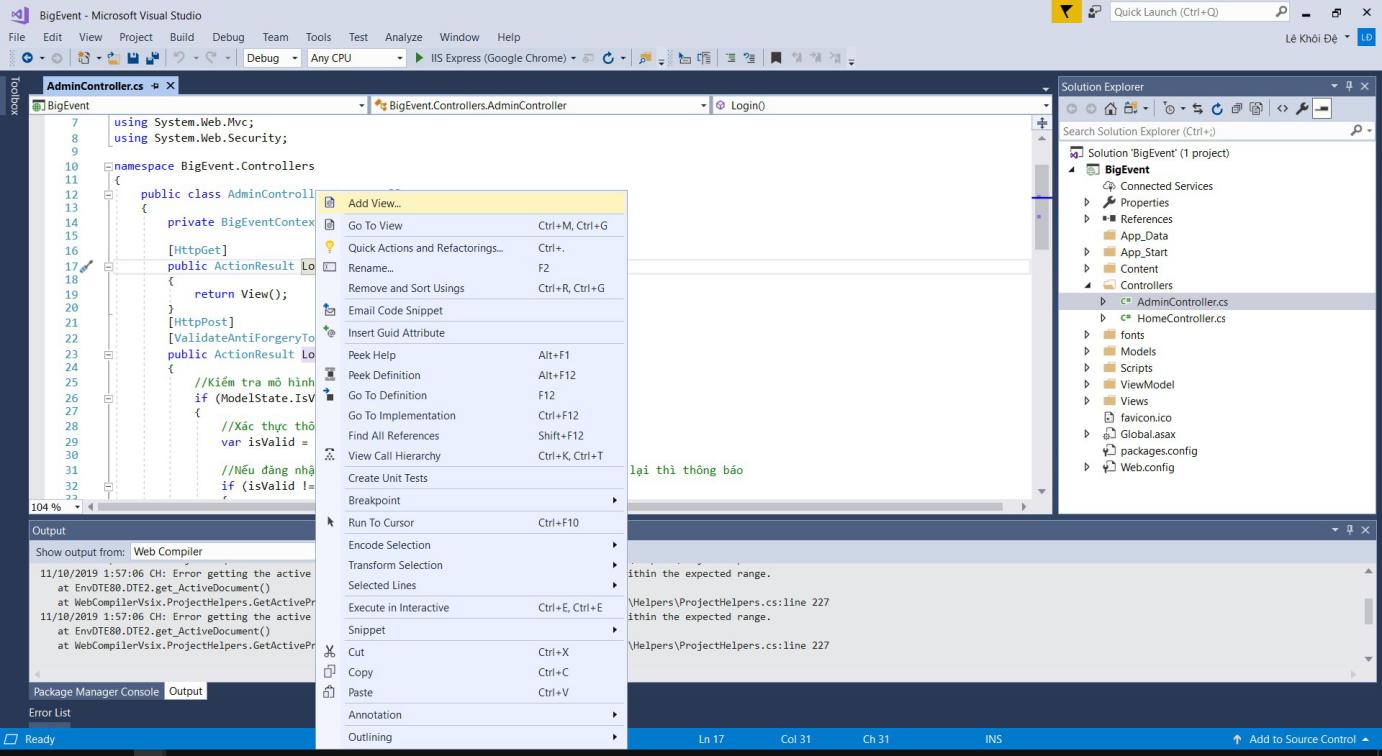
- Code của “AdminController”

|  |
| --- |
| using BigEvent.Models;  using BigEvent.ViewModel;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  using System.Web.Security;  namespace BigEvent.Controllers  {      public class AdminController : Controller      {          private BigEventContext db = new BigEventContext();            [HttpGet]          public ActionResult Login()          {              return View();          }          [HttpPost]          [ValidateAntiForgeryToken]          public ActionResult Login(LoginViewModel model)          {              //Kiểm tra mô hình dữ liệu              if (ModelState.IsValid)              {                  //Xác thực thông tin có phải admin                  var isValid = IsValidUser(model);                    //Nếu đăng nhập thành công thì chuyển tới trang chủ và ngược lại thì thông báo                  if (isValid != null)                  {                      FormsAuthentication.SetAuthCookie(model.TenDangNhap, false);                      return RedirectToAction("Index", "Home");                  }                  else                  {                      //Nếu thông tin đăng nhập không chính xác thì thông báo lỗi                      ModelState.AddModelError("Mời nhập lại", "Tên đăng nhập hoặc mật khẩu không chính xác !");                      return View();                  }              }              else              {                  //Nếu mô hình dữ liệu không hợp lệ, thì thông báo lỗi                  return View(model);              }          }              public ActionResult Logout()          {              FormsAuthentication.SignOut();              Session.Abandon();              return RedirectToAction("Index", "Home");          }            //Phương thức kiểm tra đăng nhập người dùng có phải là admin          public Admin IsValidUser(LoginViewModel model)          {              using (db)              {                  Admin admin = db.Admin.Where                      (                          query =>                          query.TenDangNhap.Equals(model.TenDangNhap)                          &&                          query.MatKhau.Equals(model.MatKhau)                      ).SingleOrDefault();                    //Nếu thông tin nhập không có trong hệ thống thì trả về Null                  if (admin == null)                      return null;                  else                      return admin;              }            }      }  } |

- Vào Web.config chỉnh sửa thêm đoạn code để kích hoạt xác thực mẫu.

|  |
| --- |
| <system.web>      <compilation debug="true" targetFramework="4.6.1" />  <httpRuntime targetFramework="4.6.1" />        <authentication mode="Forms">        <forms loginUrl="~/Admin/Login" timeout="2880" />      </authentication>    </system.web> |

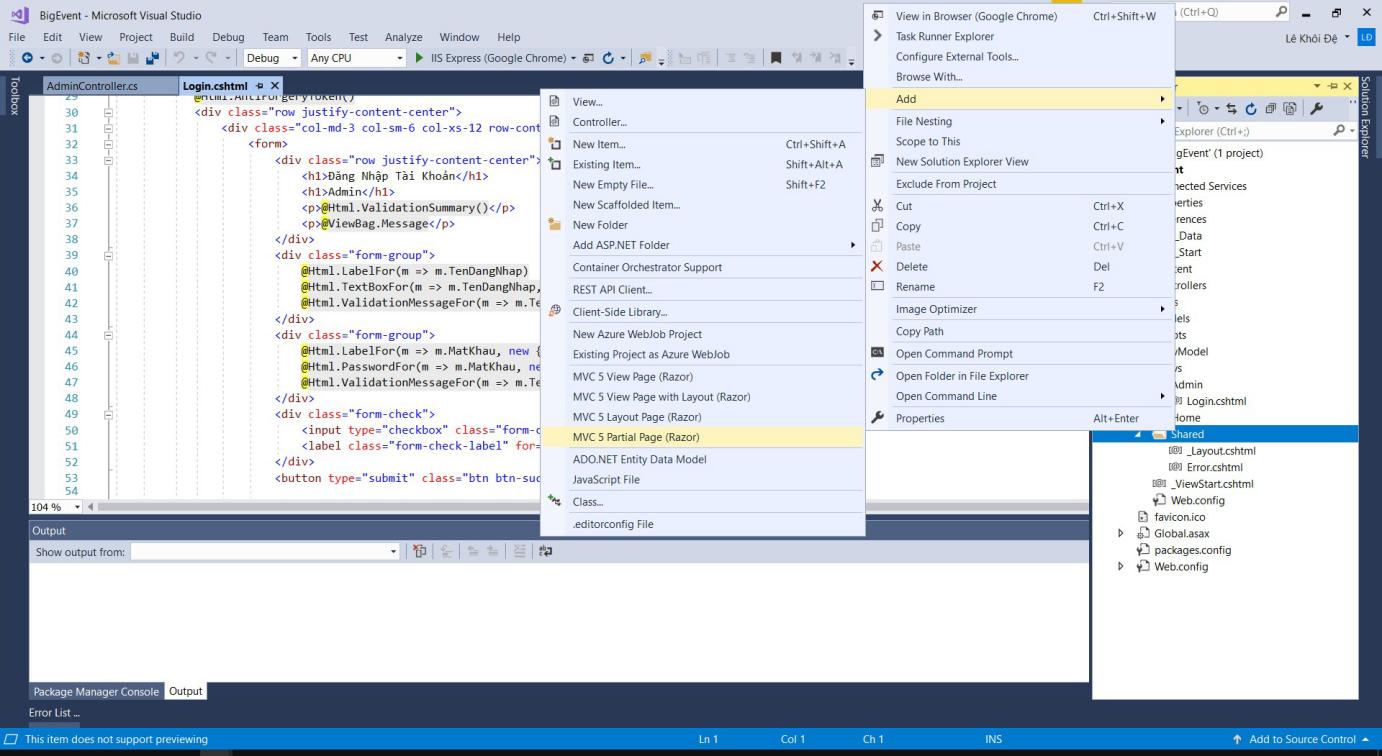
- Sau đó ta tạo View



- Code giao diện “Login.cshtml”

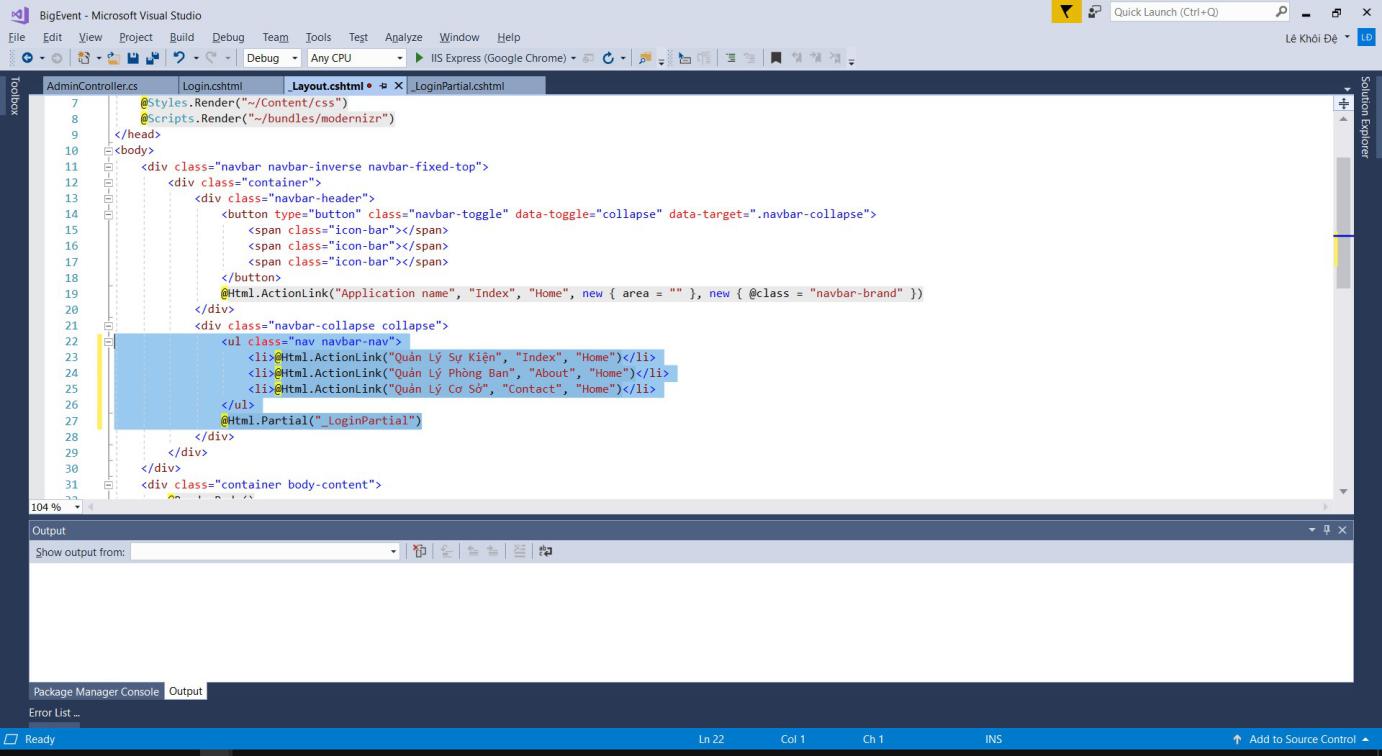
|  |
| --- |
| @model BigEvent.ViewModel.LoginViewModel@{      Layout = null;}  <!DOCTYPE html><html><head>      <meta charset="utf-8">      <meta name="viewport" content="width=device-width, initial-scale=1">      <title>Đăng Nhập</title>      <!--CSS-->      <link href="~/Content/Login/Login.css" rel="stylesheet">      <!--Script-->      <script type="text/javascript" src="~/Content/Login/jquery.js"></script>      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">      </script>      <link rel="stylesheet"            href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js">        </script>  </head><body>      <div class="container-fluid bg">          @using (Html.BeginForm("Login", "Admin", FormMethod.Post))          {              @Html.AntiForgeryToken()              <div class="row justify-content-center">                  <div class="col-md-3 col-sm-6 col-xs-12 row-container">                      <form>                          <div class="row justify-content-center">                              <h1 style="color:white">Đăng Nhập Tài Khoản</h1>                              <h1 style="color:white">Admin</h1>                          </div>                          <div class="row justify-content-center" , style="color:red">                              <h5>@Html.ValidationSummary()</h5>                              @ViewBag.Message                          </div>                          <div class="form-group">                              @Html.LabelFor(m => m.TenDangNhap)                              @Html.TextBoxFor(m => m.TenDangNhap, new { @class = "form-control", @placeholder = "Mời nhập tên đăng nhập", @required = "required" })                              @Html.ValidationMessageFor(m => m.TenDangNhap, "", new { @class = "text-danger" })                          </div>                          <div class="form-group">                              @Html.LabelFor(m => m.MatKhau, new { @class = "label", @for = "password" })                              @Html.PasswordFor(m => m.MatKhau, new { @class = "form-control", @id = "password", @placeholder = "Mời nhập mật khẩu", @required = "required" })                              @Html.ValidationMessageFor(m => m.TenDangNhap, "", new { @class = "text-danger" })                          </div>                          <div class="form-check">                              <input type="checkbox" class="form-check-input" id="rememberMe">                              <label class="form-check-label" for="rememberMe">Lưu mật khẩu</label>                          </div>                          <button type="submit" class="btn btn-success btn-block my-3">Đăng Nhập</button>                        </form>                  </div>              </div>          }      </div></body></html>  @section Scripts {      @Scripts.Render("~/bundles/jqueryval")} |

Ta tạo “\_LoginPartial.cshtml” để khi admin đăng nhập nếu thông tin chính xác thì trả về giao diện admin, nếu sai trả về giao diện đăng nhập

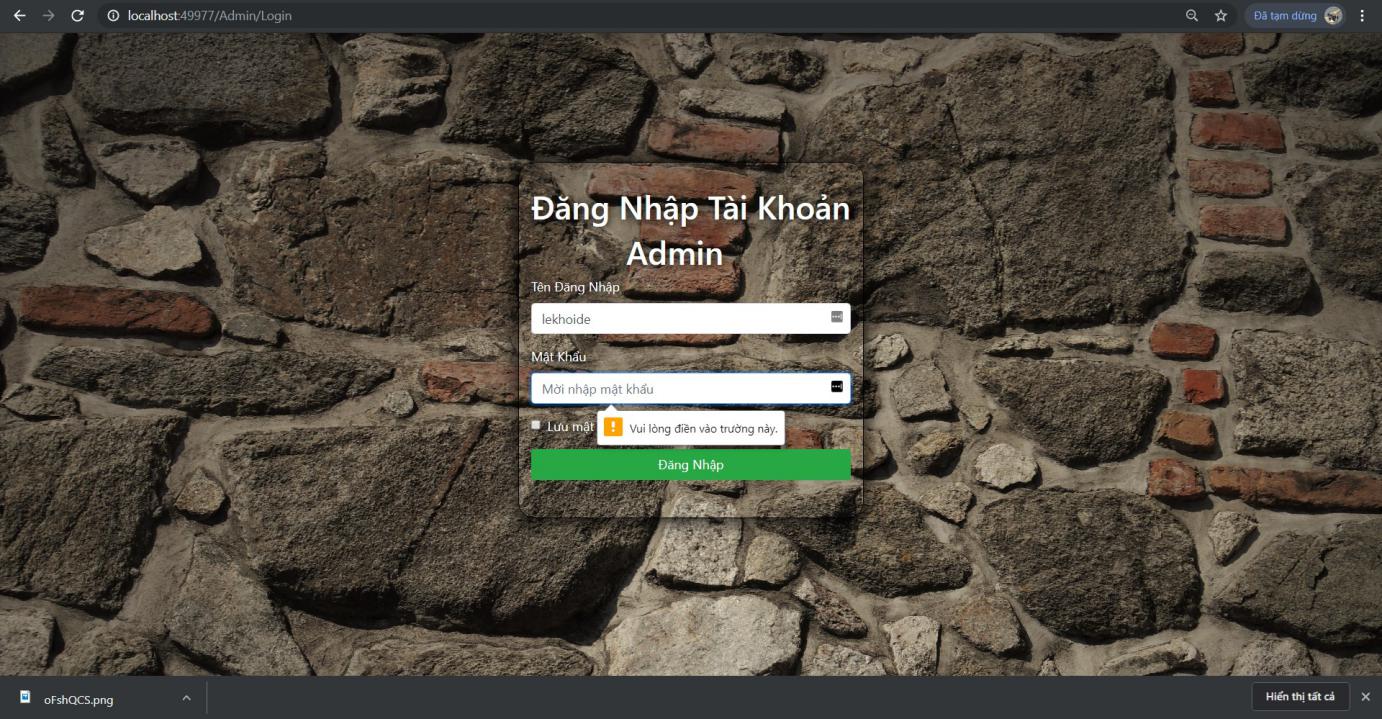
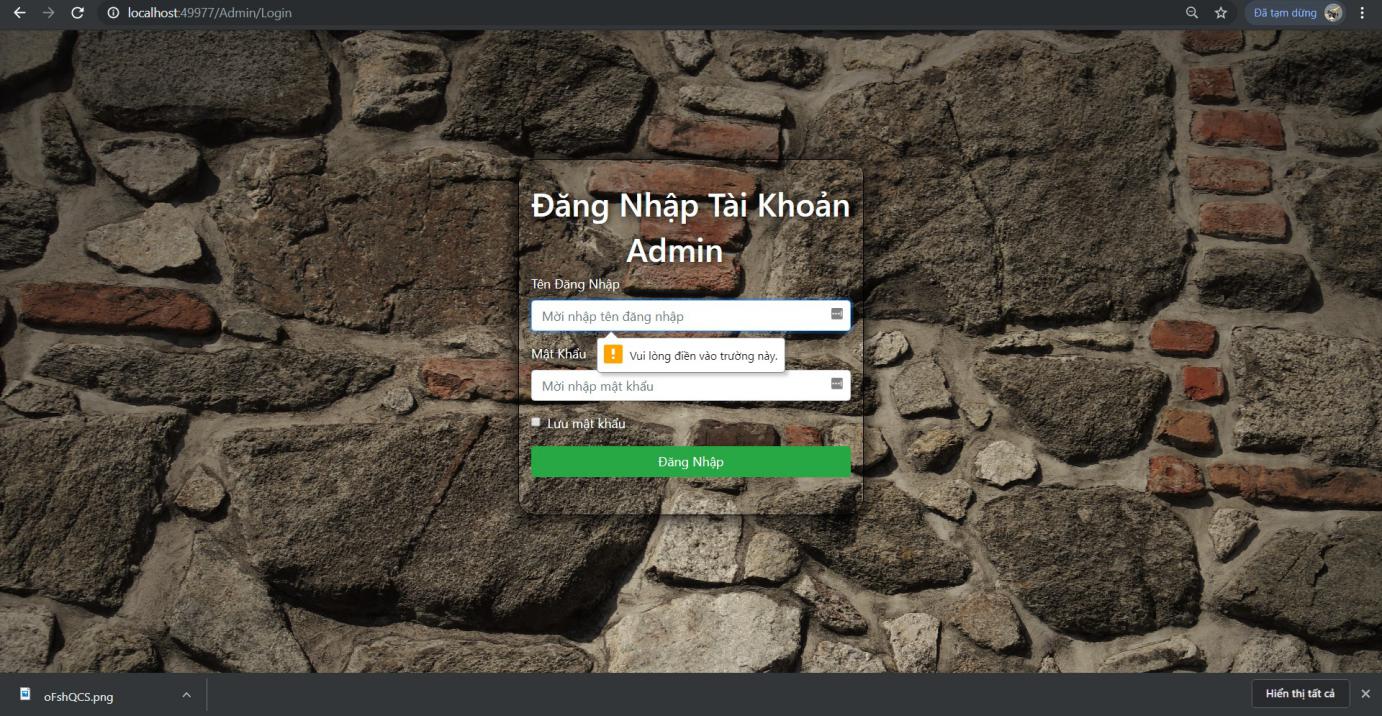


|  |
| --- |
| @if (Request.IsAuthenticated)  {      using (Html.BeginForm("Logout", "Admin", FormMethod.Post, new { id = "logoutForm", @class = "navbar-right" }))      {          @Html.AntiForgeryToken()            <ul class="nav navbar-nav navbar-right">              <li>                  @Html.ActionLink("Hello " + User.Identity.Name + " !", "", "", routeValues: null, htmlAttributes: new {})              </li>              <li><a href="javascript:document.getElementById('logoutForm').submit()">Log off</a></li>          </ul>      }  }else  {      <ul class="nav navbar-nav navbar-right">          <li>@Html.ActionLink("Đăng Nhập", "Login", "Admin", routeValues: null, htmlAttributes: new { id = "loginLink" })</li>      </ul>  } |

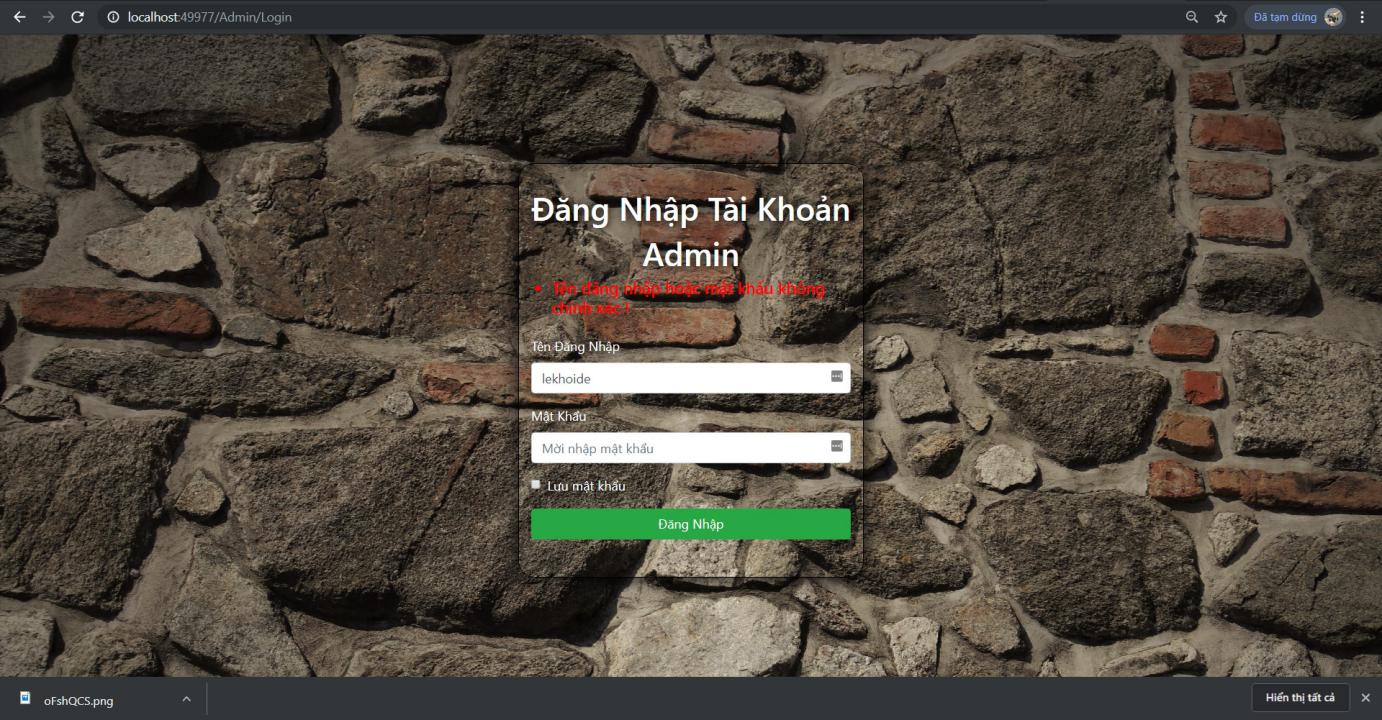
- Ta chỉnh sửa ở trang “Layout.cshtml”



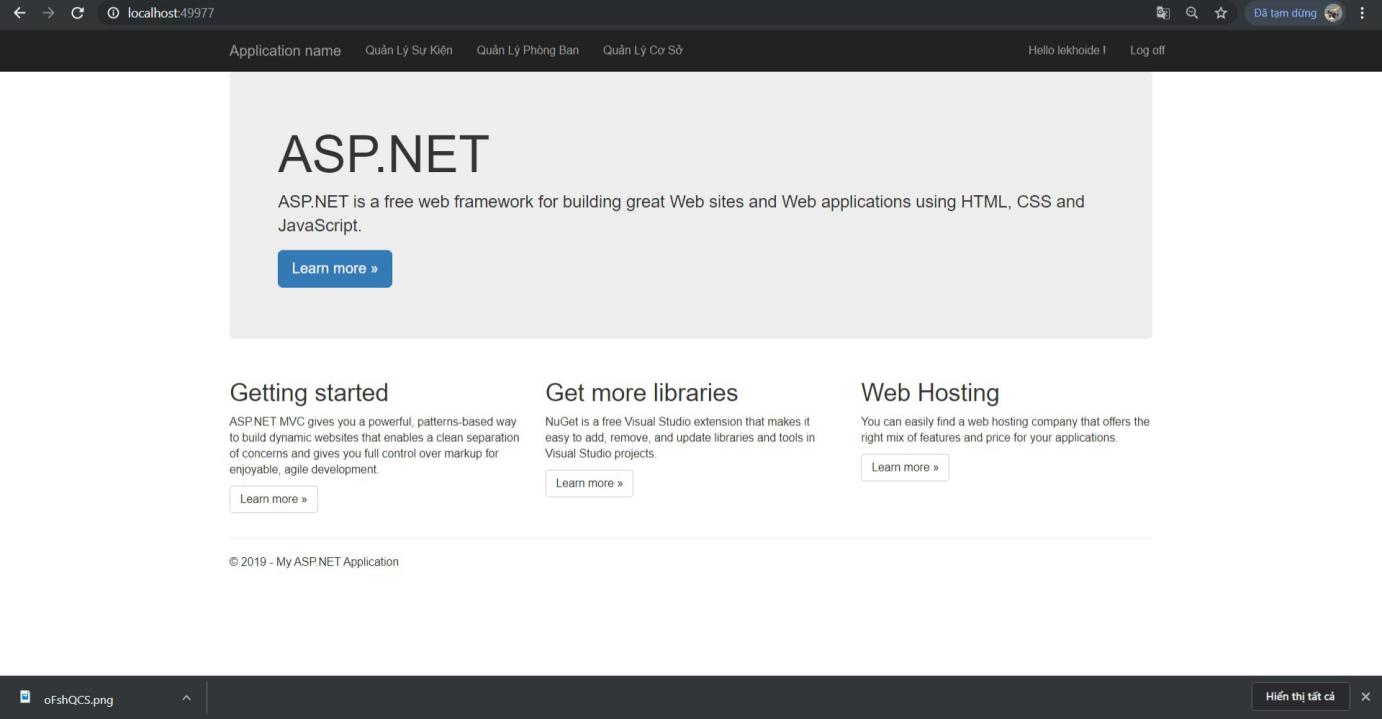
- Kết quả trả về khi chưa nhập thông tin



-Kết quả trả về khi nhập thông tin không chính xác



- Kết quả trả về khi đăng nhập thành công



1. **LÊ KHÔI ĐỆ**
   1. Hoàn thành

Cài đặt hệ thống lên VisualStudio 2017

Code chức năng Đăng Nhập của Admin

Cập nhật lại đặc tả đề tài và ERD

Viết hồ sơ thiết kế chức năng đăng nhập của Admin

Bài toán 2 (Tuần 2) : Thuộc tính Ngày Tạo Sự Kiện trong bảng Sự Kiện khi nạp một dữ liệu làm sao để tự động chọn ngày và giờ hiện tại. Hướng giải quyết tìm hiểu SQL

|  |
| --- |
| ALTER TABLE [SuKien] ADD  DEFAULT GETDATE() FOR [NgayTaoSuKien]  GO |

* 1. Công việc tuần 4

Code chức năng xem, tạo, xóa, sửa Quản Lý Phòng Ban

Code chức năng xem, tạo, xóa, sửa Quản Lý Cơ Sở

Code chức năng xem, tạo xóa, sửa Quản Lý Sự Kiện

Chuẩn bị bài thuyết trình họp cô Thơ Tuần 4

1. **NGUYỄN BÁ TÂM**