**TRƯỜNG ĐẠI HỌC SƯ PHẠM**

**KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**



**MÔN BẢO MẬT THÔNG TIN**

**GIẢI PHÁP CHO CÔNG TY MagicECert**

**\* Giáo viên hướng dẫn: Lương Vĩ Minh**

**\* Nhóm FinalC-M:**

**Nguyễn Văn Phi Tước – 14110434**

Contents

[**I.** **Phần mở đầu** 4](#_Toc484290687)

[**1.** **Lời nói đầu** 4](#_Toc484290688)

[**2.** **Đặt vấn đề** 4](#_Toc484290689)

[II. Nội dung 4](#_Toc484290690)

[**1.** **Hướng giải quyết vấn đề** 4](#_Toc484290691)

[**2.** **Giải pháp** 4](#_Toc484290692)

[**3.** **Các công nghệ cần sử dụng** 5](#_Toc484290693)

[Về phía Client 5](#_Toc484290694)

[Về phía Server 5](#_Toc484290695)

[Về phía DataCenter 6](#_Toc484290696)

[**4.** **Kiến thức sử dụng** 6](#_Toc484290697)

[a) Mã hóa bất đối xứng 6](#_Toc484290698)

[b) Chữ ký điện tử 6](#_Toc484290699)

[c) Giấy chứng nhận điện tử 6](#_Toc484290700)

[d) Giao thức SSL 6](#_Toc484290701)

[5. Khuyết điểm và giới hạn của phần mềm 6](#_Toc484290702)

[6. Hướng dẫn cài đặt 8](#_Toc484290703)

[B1: Deploy web to IIS Server 8](#_Toc484290704)

[B1: Tạo CA root và mở kênh truyền 11](#_Toc484290705)

[B3: Binding port 15](#_Toc484290706)

[B4: Restore Database 16](#_Toc484290707)

[III. Use-Case diagram 19](#_Toc484290708)

[1. Usecase Diagram 19](#_Toc484290709)

[2. Actors 20](#_Toc484290710)

[3. Use-case 21](#_Toc484290711)

[4. Use-case name 21](#_Toc484290712)

[4.1 Use-case ID 21](#_Toc484290713)

[4.2 Use-case Name 21](#_Toc484290714)

1. **Phần mở đầu**
2. **Lời nói đầu**

Sự phát triển nhanh chóng của khoa công nghệ, đặc biệt là sự phát triển của công nghệ máy tính và kỹ thuật máy tính toán đã làm thay đổi rất nhiều các hoạt động xã hội trên thế giới. Thương mai điện tử đã ra đời trong bối cảnh đó, tính hiệu quả và thuận lợi của hoạt động thương mại điện tử ngày càng được khẳng định. Nhờ thế mà các hoạt động thương mại điện tử ngày càng được mở rộng và dần phổ biến ở nhiều nới trên thế giới. Nền tảng công nghệ của thương mại điện tử chính là mạng toàn cầu – Internet. Song, Internet cũng chính là mối đe dọa đến an toàn của hoạt động thương mại điện tử. Trong đó vấn đề toàn vẹn và chống từ chối trách nhiệm là vô cùng quan trọng, vì thế giấy chứng nhận điện tử ra đời. Bài viết này, sẽ nêu lên vấn đề và các giải pháp một bài toán thực tế thường gặp ngày nay.

1. **Đặt vấn đề**

Công ty MagicECert đang có kế hoạch triển khai dịch vụ cung cấp quản lý giấy chứng nhận số (Certificate Authority) miễn phí cho mọi đối tượng. Đồng thời, giúp khách hàng quản lý luôn các khóa mật mã có sử dụng trong hệ thống.

Để giảm chi phí triển khai phần cứng, công ty quyết định thuê một dịch vụ Data Center trên mạng để chứa dữ liệu thông tin của chứng nhận điện tử và các thông tin khác liên quan đến hệ thống.

Tuy nhiên, do đặc thù sản phẩm dịch vụ này có các thông tin nhạy cảm cần bảo vệ, dữ liệu lại lưu trữ trên môt Data Center không đảm bảo tính an toàn và bảo mật về nội dung, công ty mong muốn có được một hệ thống đảm bảo an toàn về mặt dữ liệu lưu trữ và dữ luyện truyền nhận giữa ứng dụng và kho dữ liệu.

# Nội dung

1. **Hướng giải quyết vấn đề**

Phía người dùng: Xây dựng Winform Application để cho người dùng tương tác với phía server.  
Phía server: Xây dựng ta có Service API nhận request từ phía người dùng và sau đấy thực hiện trên databasecenter, sau đó trả kết quả về cho người dùng

Phía dataCenter: dữ liệu sẽ được mã hóa và lưu trên này, cặp khóa sẽ do admin giữ (trong project thì cặp khỏa do chính server giữ)

1. **Giải pháp**

* Phía client: là một Window form cho phép người dùng tương tác với hệ thống. Do đặc thù về tính bảo mật, riêng tư cũng như toàn vẹn dữ liệu thì việc truyền bằng một kênh truyền về phía server là vô cùng quan trọng. Ở phía client người dùng có thể thực hiện các chức năng như:
* Đăng kí.
* Đăng nhập.
* Yêu cầu cấp giấy chứng nhận điện tử.
* Quản lí các giấy chứng nhân điện tử hiện có.
* Kiểm tra giấy chứng nhận điện tử
* Xuất giấy chứng nhận
* Tìm kiếm mật khẩu
* Thây đổi giấy chứng nhận
* Phía server: là một service API viết bằng C# để có thể giao tiếp với phía client. Phía client sẽ gửi request lên server, sau khi tiếp nhận thành công qua kênh truyền an toàn thì phía server tiến hành xử lí và mã hóa dữ liệu để truyền tải tới database, để đảm bảo dữ liệu được an toàn. Ở phía server thực hiện các chức năng sau:
* Cấp tài khoản cho người dùng.
* Cấp giấy chứng nhận điện tử.
* Quản lý giấy chứng nhận điện tử của các user (bao gồm chức năng CRUD).
* Đảm báo bảo mật thông tin người dùng.
* Giao tiếp với database.
* Phía Data Center: Lưu trữ thông người dùng và giấy chứng nhận điện tử đã được mã hóa, quản lý cặp khóa của giấy chứng nhận.

1. **Các công nghệ cần sử dụng**

### Về phía Client

* Winform Application
* Socket, newtonsoft json, web request
* Môi trường lập trình chính trên visual studio 2017

### Về phía Server

* Web Service MVC Restful API
* Library Bouncy Castle
* Socket
* Entity framework 6.x
* Môi trường lập trình chính trên visual studio 2017
* IIS10.0 Server

### Về phía DataCenter

* SQL server 2008R2

1. **Kiến thức sử dụng**

### Mã hóa bất đối xứng

### Chữ ký điện tử

### Giấy chứng nhận điện tử

### Giao thức SSL

## Khuyết điểm và giới hạn của phần mềm

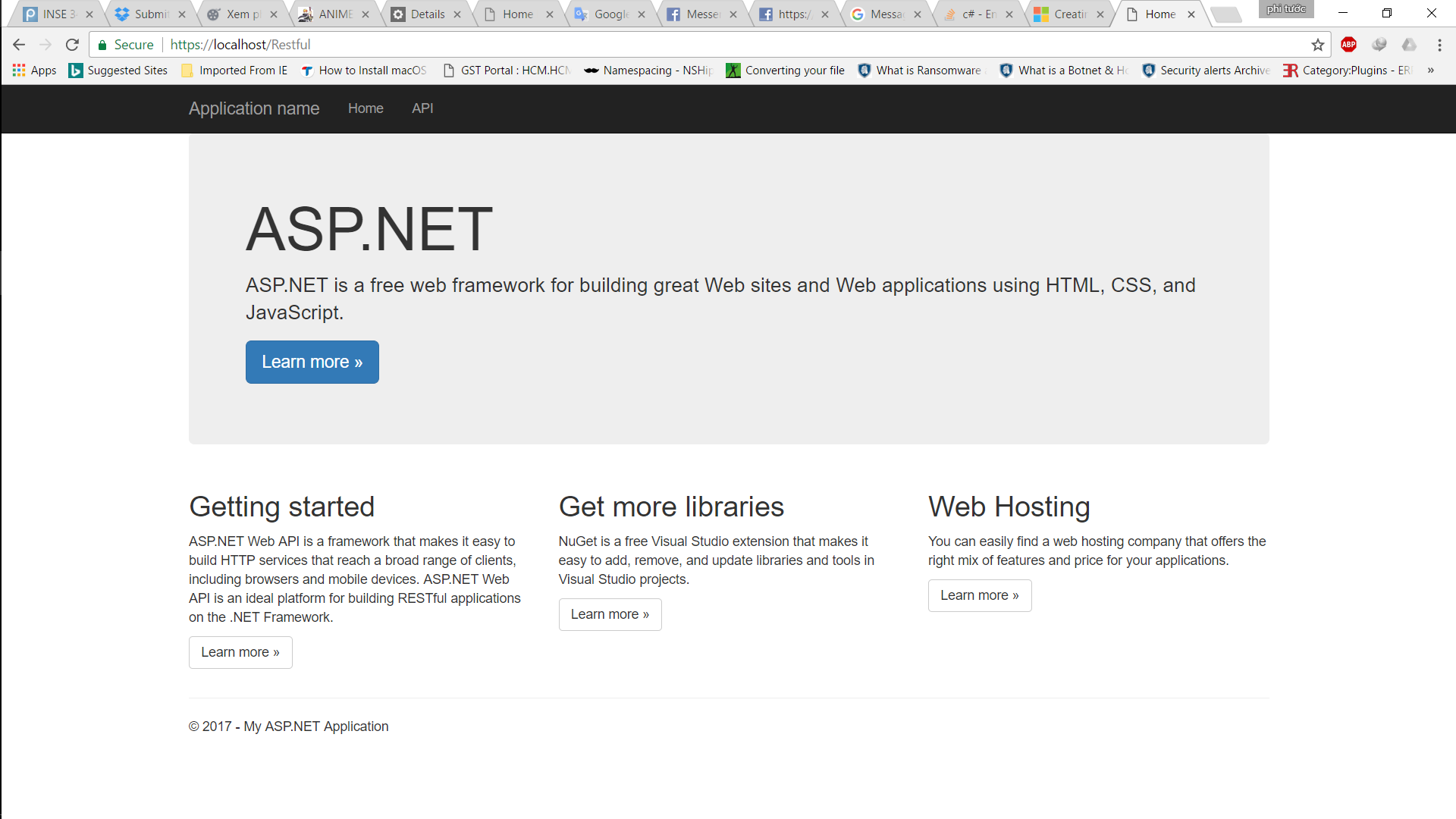
Do trình độ kỹ thuật chưa cao, nên phần mềm chỉ được viết dưới sự hiểu biết của học sinh về các kiến thức cơ bản của môn bảo mật thông tin.

Khuyết điểm:

* Một số chức năng chưa hoàn thành (xuất giấy chứng nhận, tìm lại mật khẩu, update giấy chứng nhận(ở phía client))
* Ơ

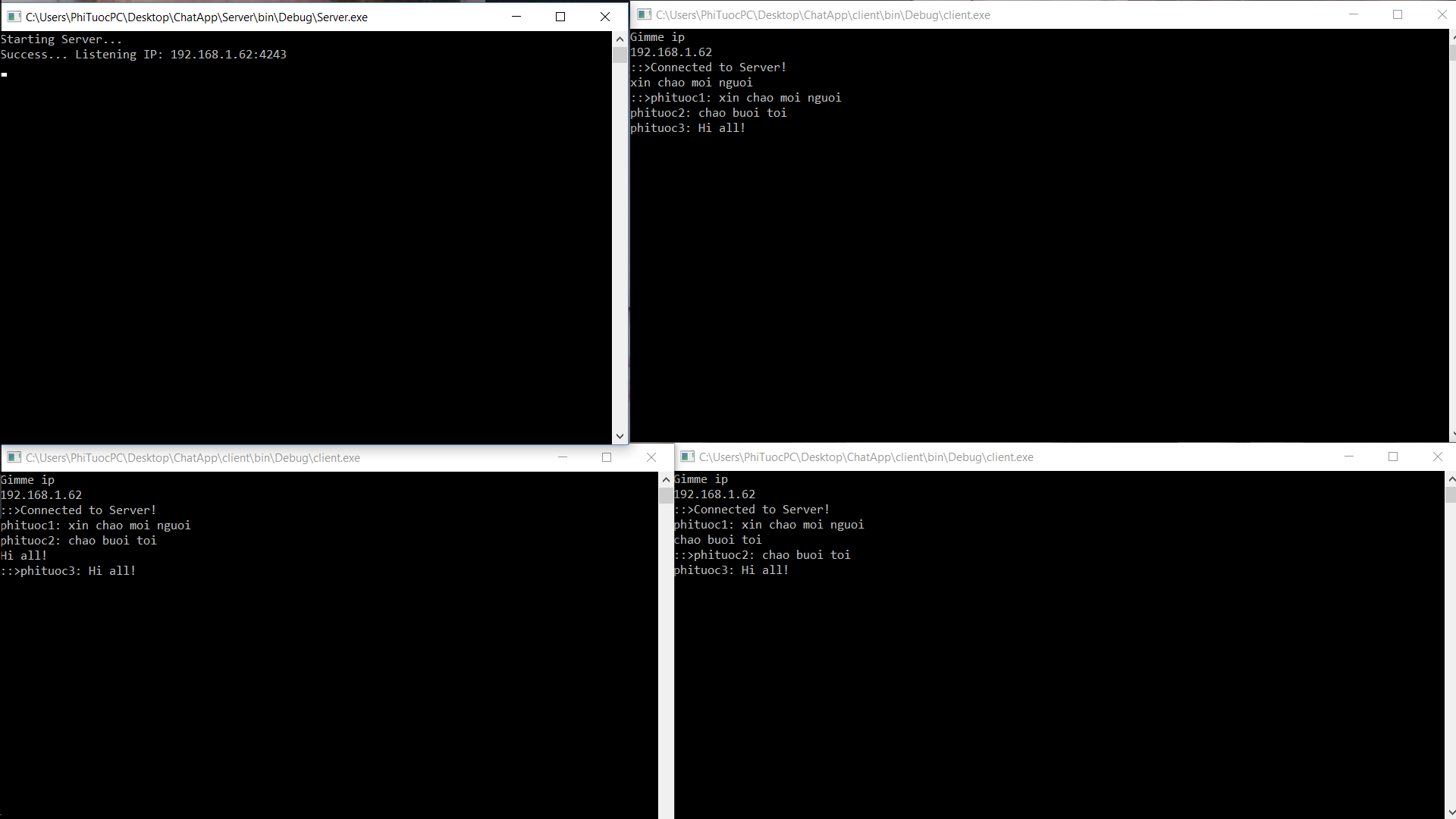
Giới hạn:

* Do chưa nắm vững kiến thức ssl nên chỉ mới mở socket cho server ở máy tính cục bộ bằng cách tạo một chứng nhận tự ký lưu trên máy và sau đó mở kênh truyền an toàn qua địa chỉ sau: <https://localhost/Restful> là website được deploy lên IIS10.0



Khi deploy web online : qua đường dẫn <http://finalc-m.gear.host/> thì không còn kênh truyền an toàn nữa, ý tưởng ở đây sẽ viết một lớp SSL để tạo ra kênh truyền an toàn.

* Dựa trên chat app đã tự viết:



Ý tưởng cơ bản như sau: ta sẽ cho

B1: server mở socket ở port x tự chọn (listener)

B2: Khi client gửi request đến server, server sẽ tùy chọn cho request nào được thực hiện (ơ chatapp thì tất cả mọi request điều được) (client socket accept)

B3: Sau đấy server sẽ gửi giấy chứng nhận và public key của mình cho client.

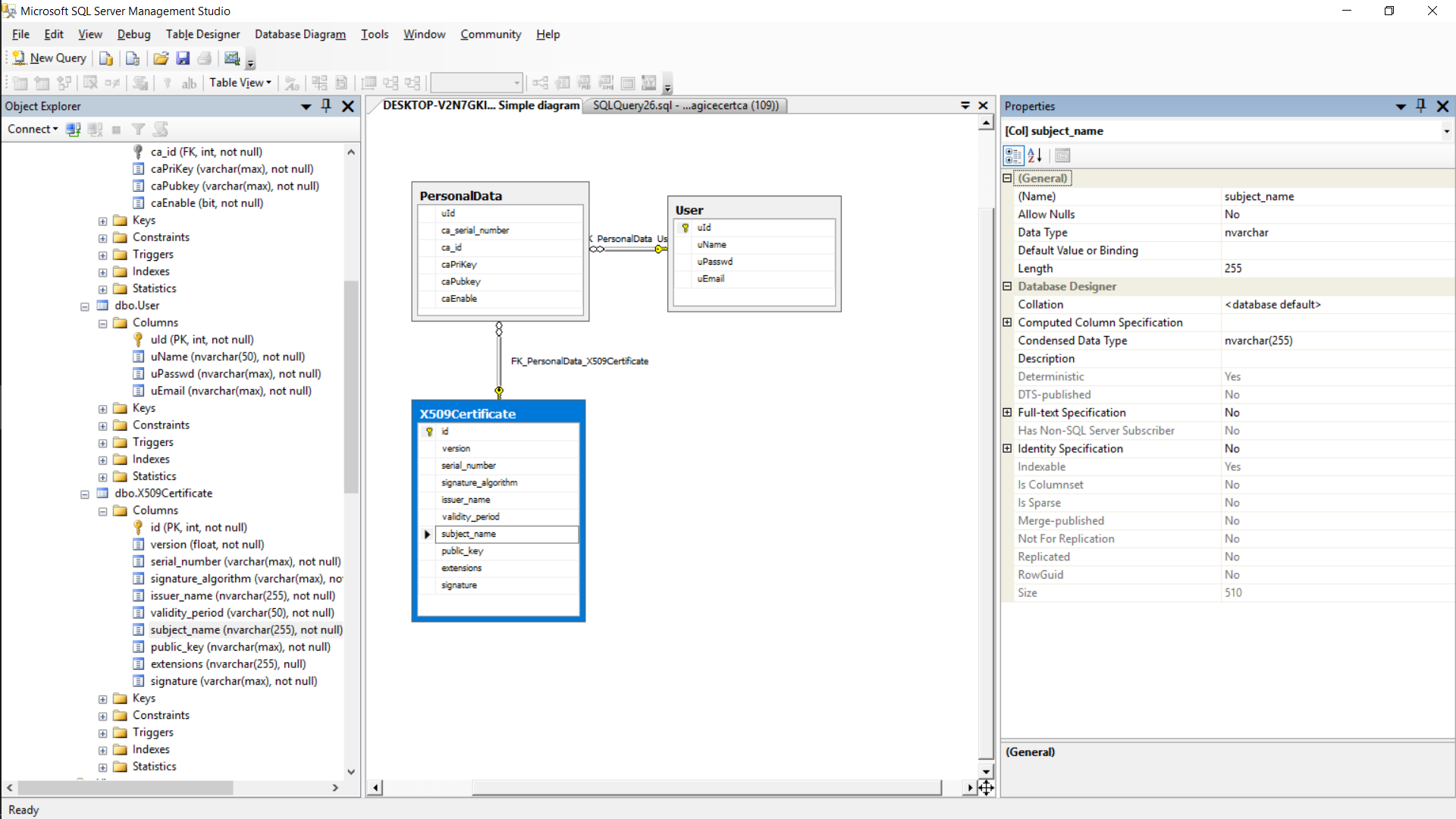
B4: Client sẽ kiểm tra giấy chứng nhận, nếu đúng thì client sẽ dùng publickey mã hóa dữ liệu có chứa publicKeyClient gửi lại cho Sever

B5: Server sẽ giải mã bằng privateKeyServer của mình sau đó tạo ra cặp key pairs dùng cho session của client và gửi cho client publicKeyServer mã hóa bằng publicKeyClient

B6: Client sẽ giải mã bằng privateKeyClient và nhận publicKeyServer

B7: Hai bên kết thúc và tiến hành giao dịch

* Do thời gian và kiến thức còn hạn chế, và chưa tiếp cận kỹ thuật này kịch bản phần này chưa hoàn thành. Ở báo cáo này sẽ nói về kịch bản dùng CA root để mở https request ở localhost
* Ở phần database, bị lỗi không lưu được subject name quá 50 ký tự mặc dù database đã thiết kế có dung lượng lớn hơn:



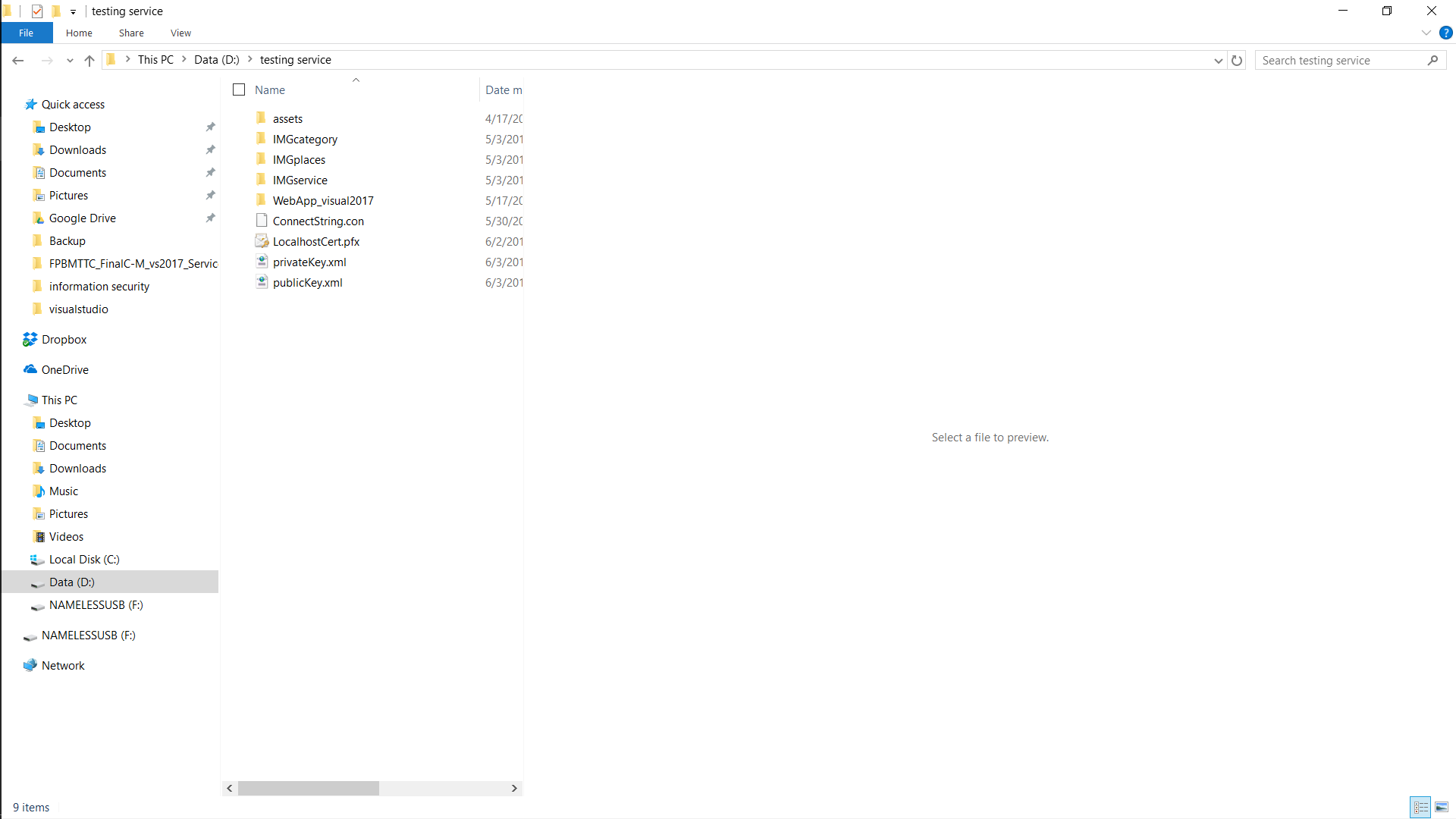
## Hướng dẫn cài đặt

Về phía Client: winform application chỉ cần mở file .exe và sử dụng

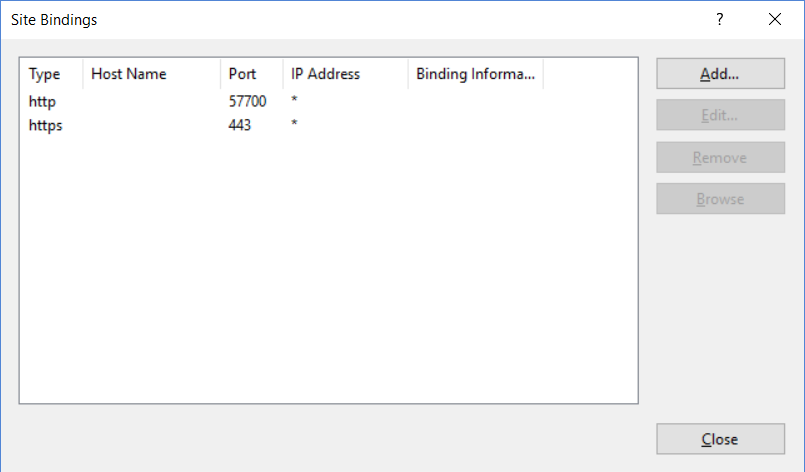
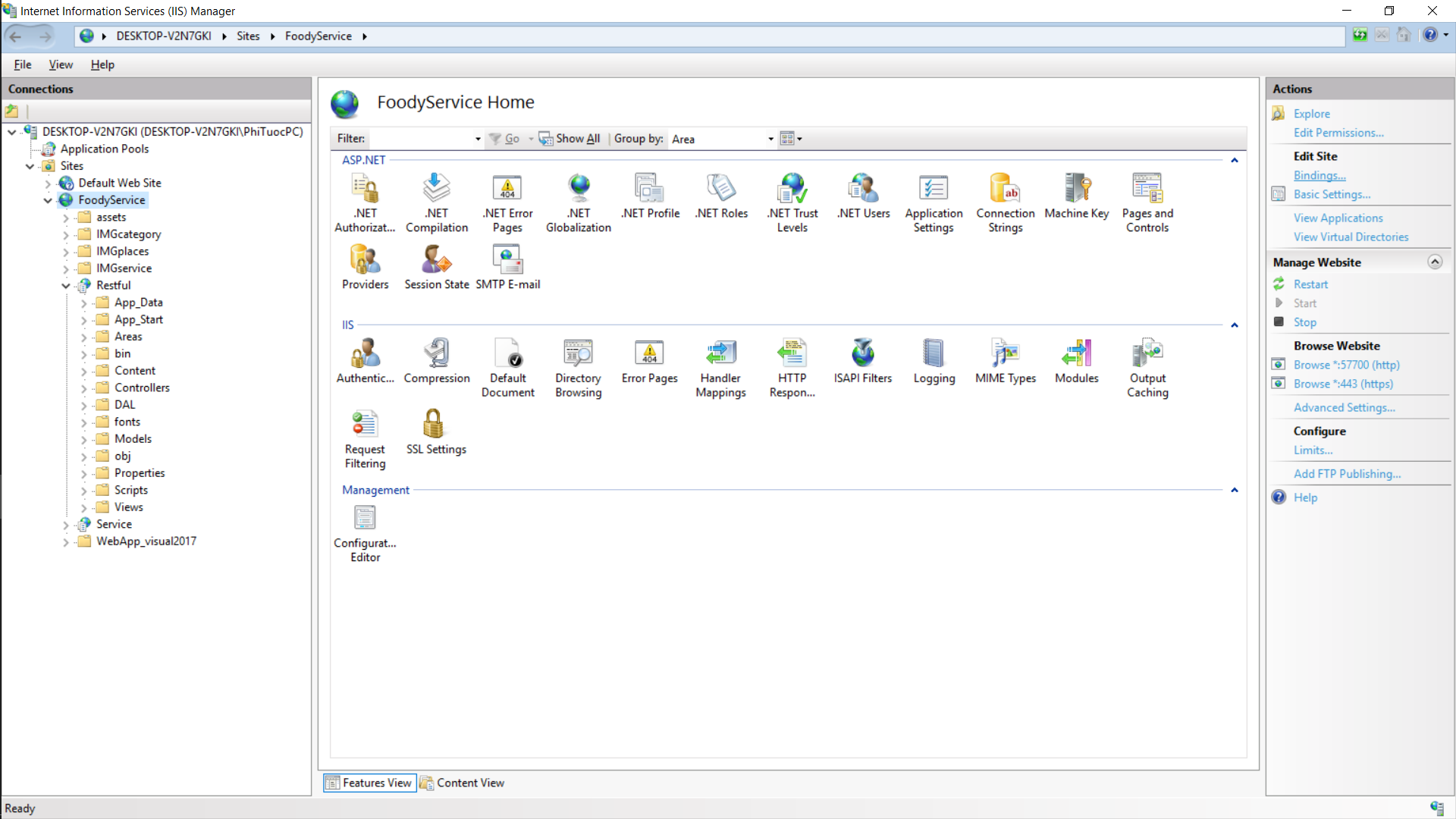
Về phía Server:

### B1: Deploy web to IIS Server

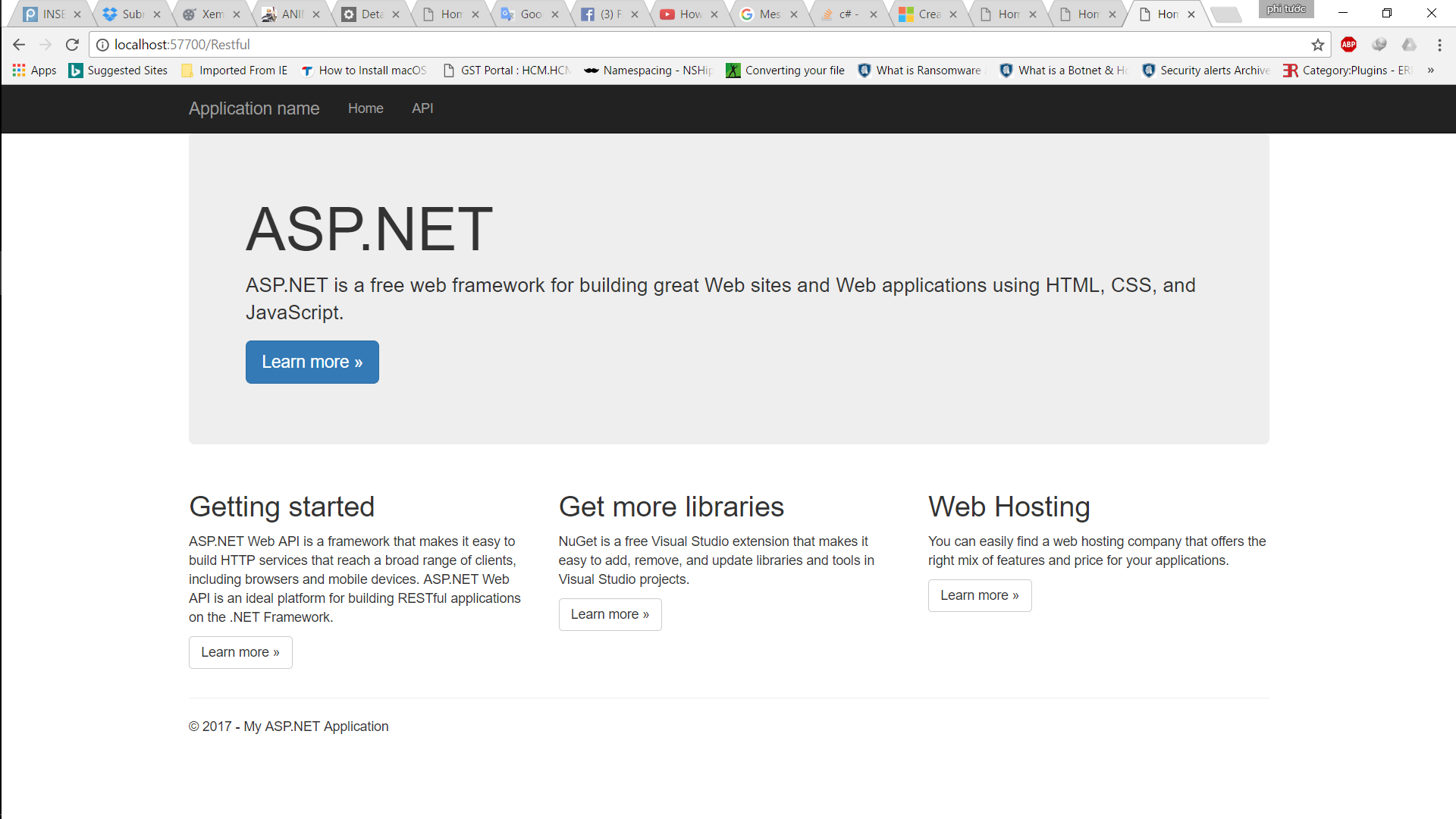
* Ta mở iis manager => chuột phải vào sites và chọn Add Website => chọn sites name, nên nhớ là để defaultAppPool do trong ứng dụng sử dụng nhiều phiên bản .net framework cấp cao => chọn physiscal path là một thư mục nào đấy, làm server root trong bài là testing service



* Chuột phải vào sites vừa tạo chọn Add Application ở đây chọn alias là Restful, sau đó chọn physiscal path đển web application
* Sau đó ở site vừa tạo ta chọn bindings để edit port

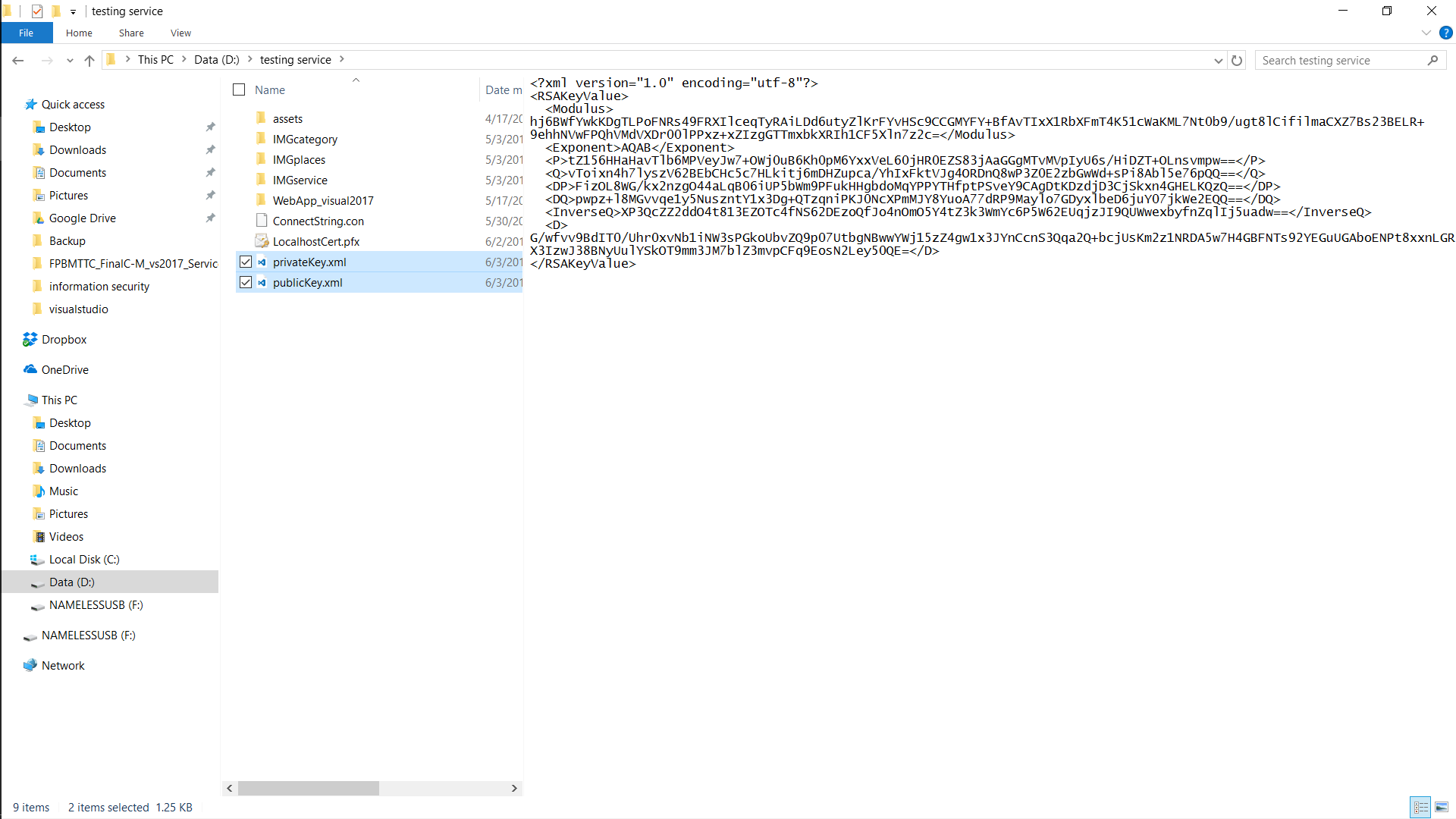


Sau đấy chạy thử



* Sau đấy đặt 2 file publickey.xml và privatekey.xml vào đường dẫn root của server:

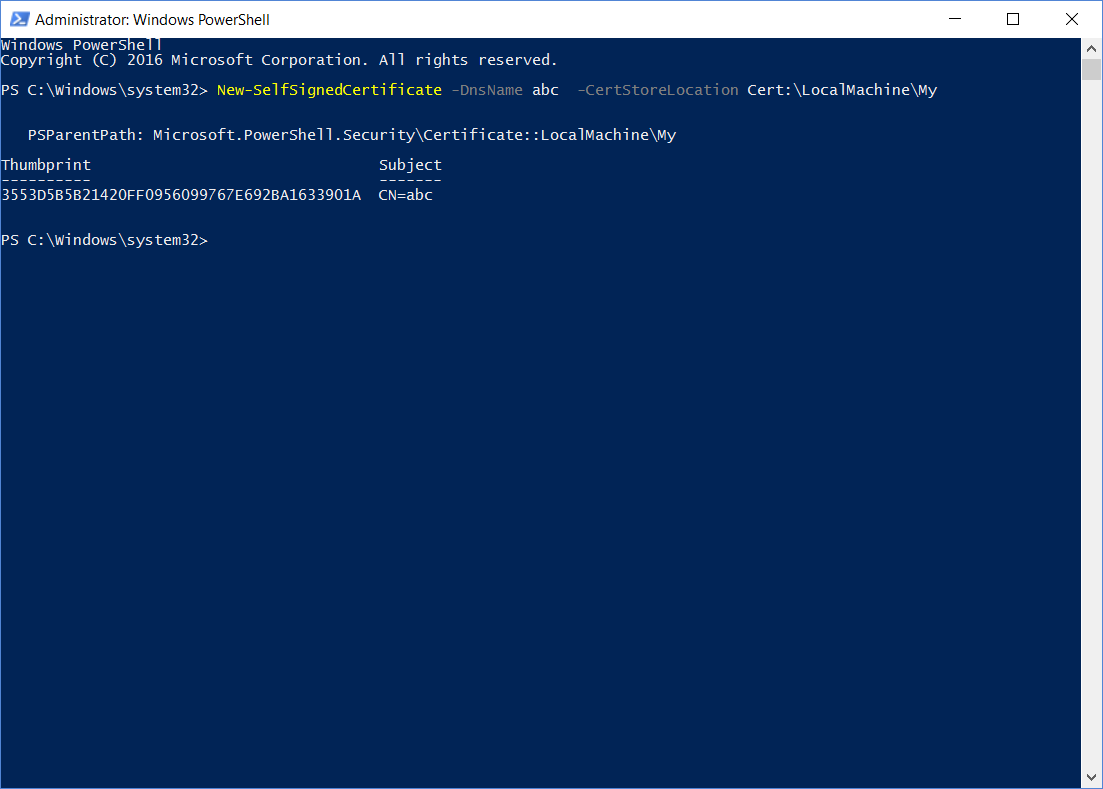
Đấy là cặp key rsa 128bytes



### B1: Tạo CA root và mở kênh truyền

* Tạo CA root bằng windows 10, ta mở power shell bằng quyền administrator lên
* Chạy câu lênh sau để mở self-sign:

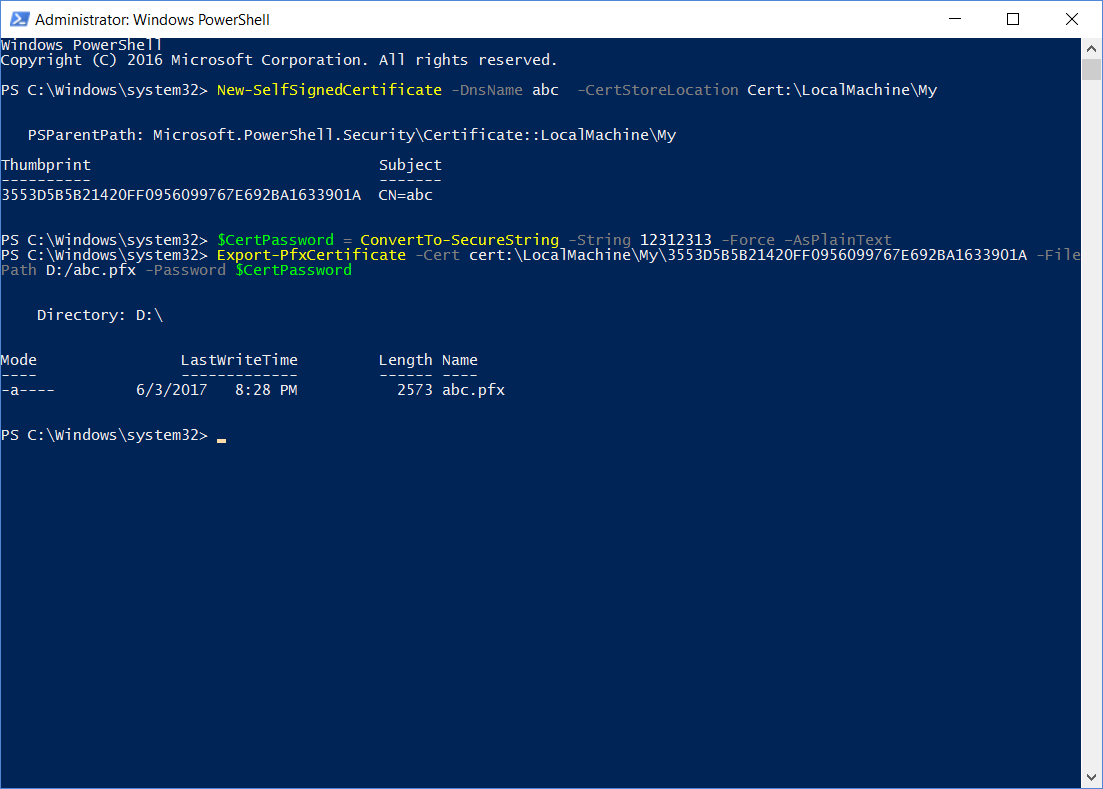
New-SelfSignedCertificate -DnsName {Your Subject} -CertStoreLocation Cert:\LocalMachine\My

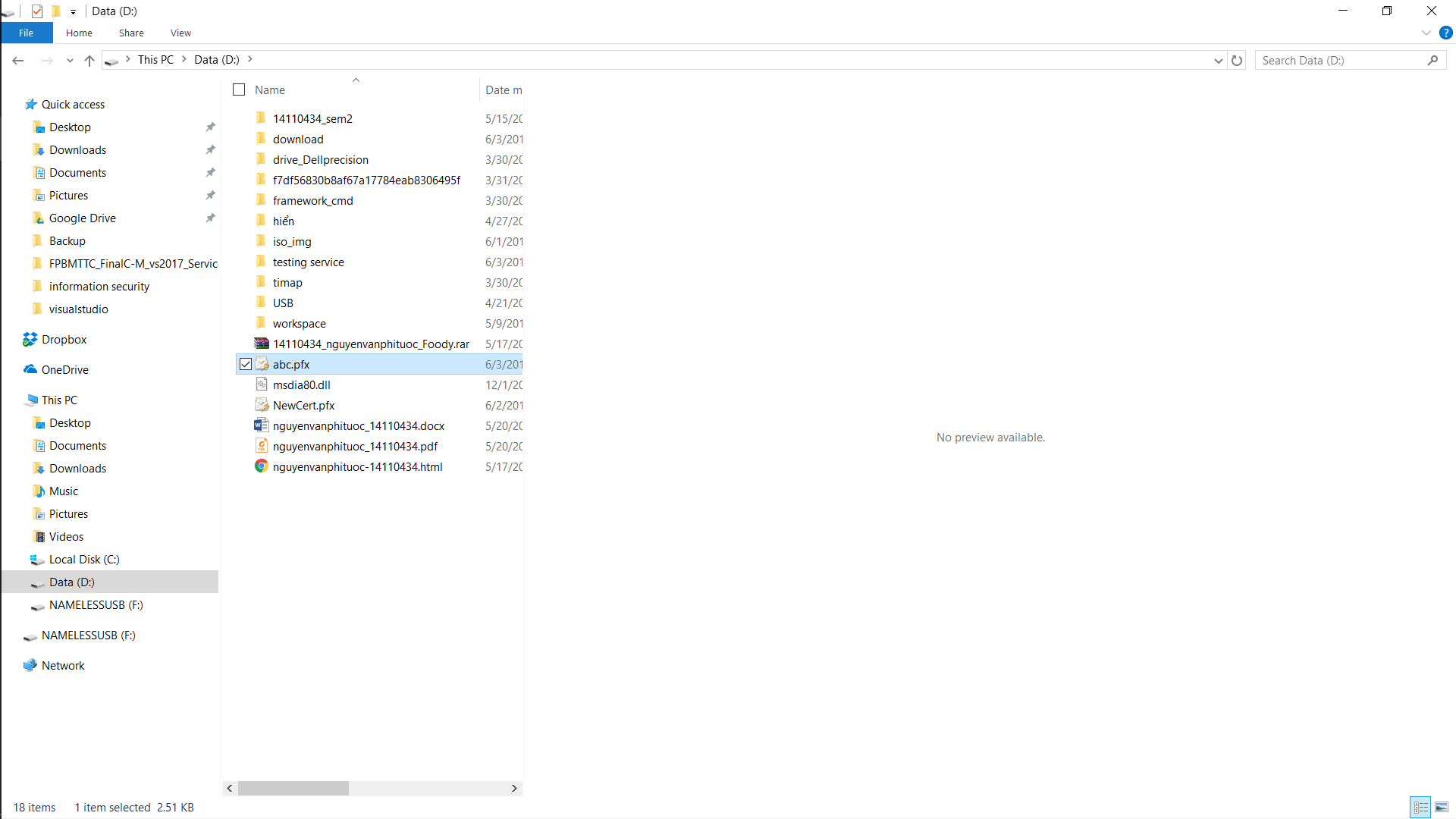


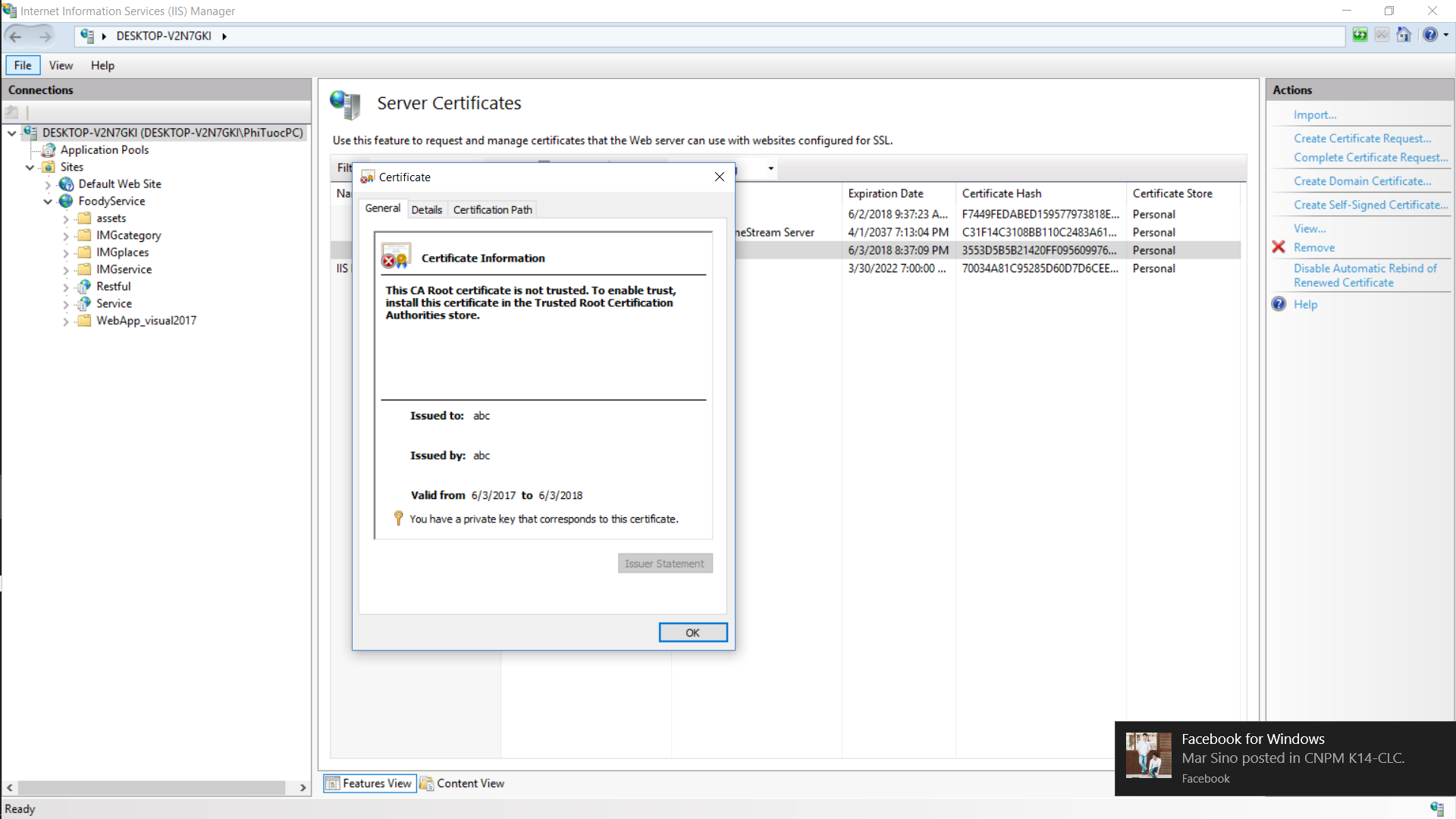
* Chạy câu lệnh sau để chọn passwd sign:

$CertPassword = ConvertTo-SecureString -String {your password} -Force –AsPlainText

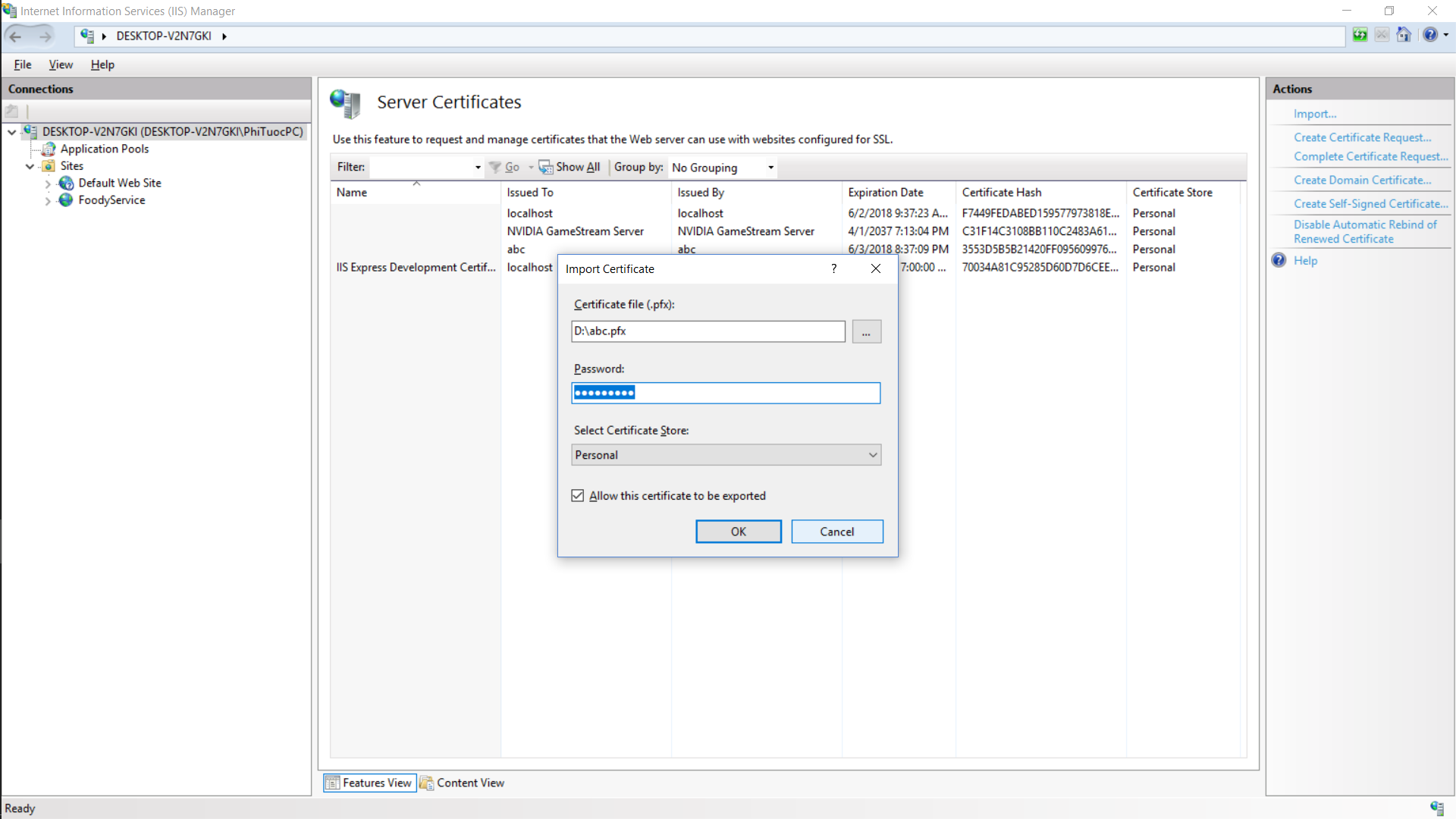
* Thay thế "Thumbprint" với Thumbprint và "path-to-pfx" là đường dẫn mà nơi PFX cert được lưu.  
  Export-PfxCertificate -Cert cert:\LocalMachine\My\{Thumbprint} -FilePath {path-to.pfx} -Password $CertPassword



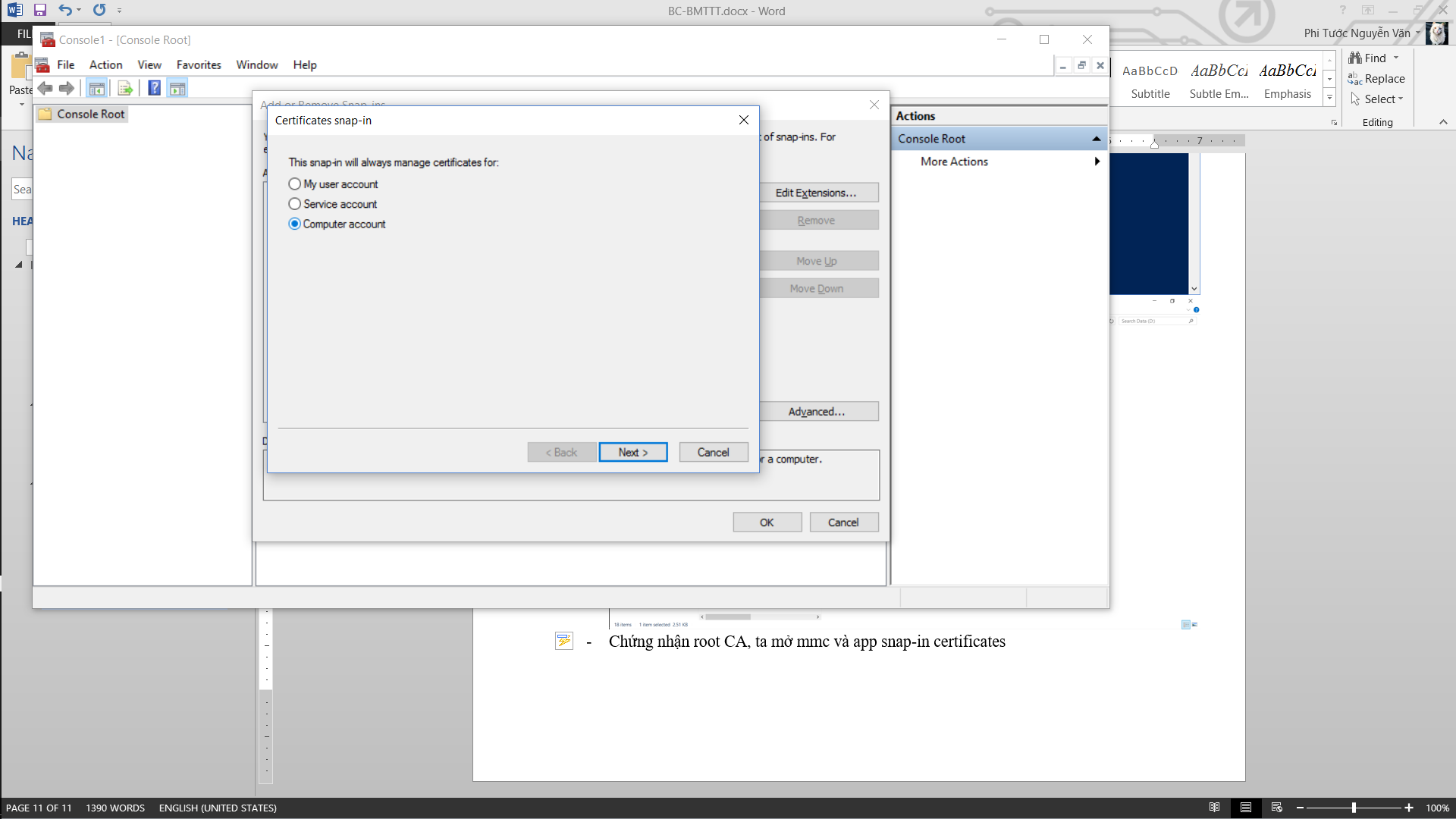




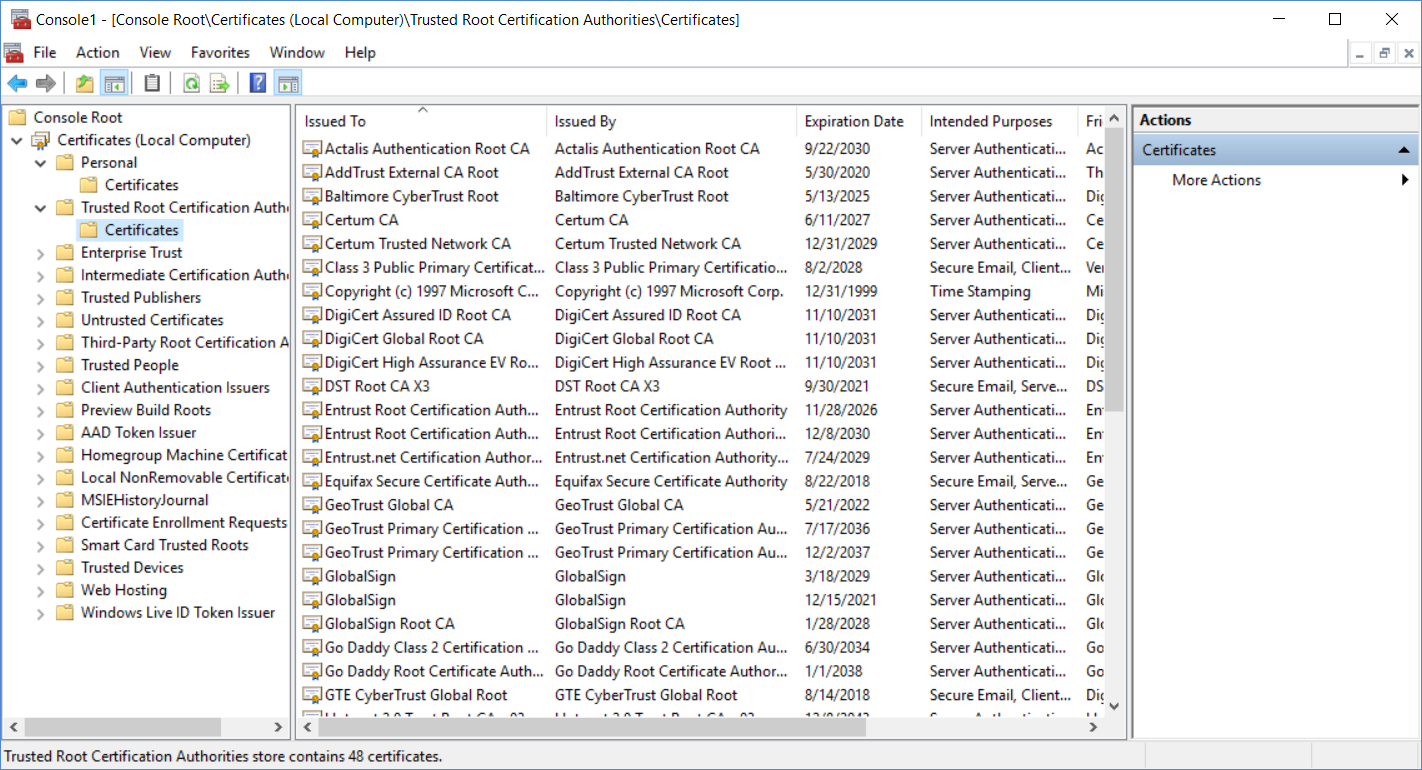
* Import CA vừa tạo, ta vào iis manager chọn server là tên máy tính cục bộ và mở Server Certificates



* Chứng nhận root CA, ta mở mmc và app snap-in certificates

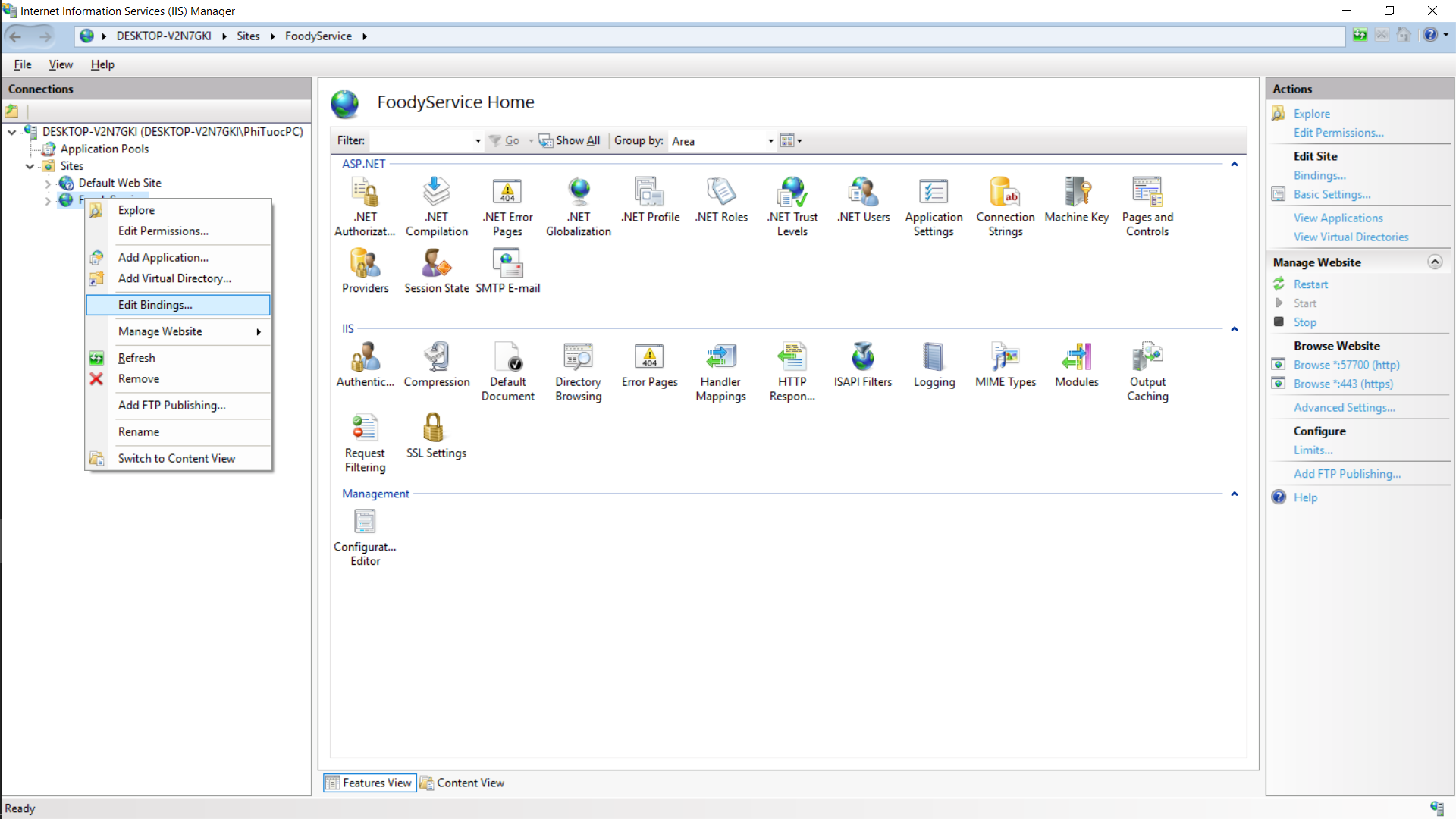


* Sau đó mở personal certificates copy cái chứng nhận có subject name mình vừa tạo, sau đó mở trusted root CA và pasted vào certificates

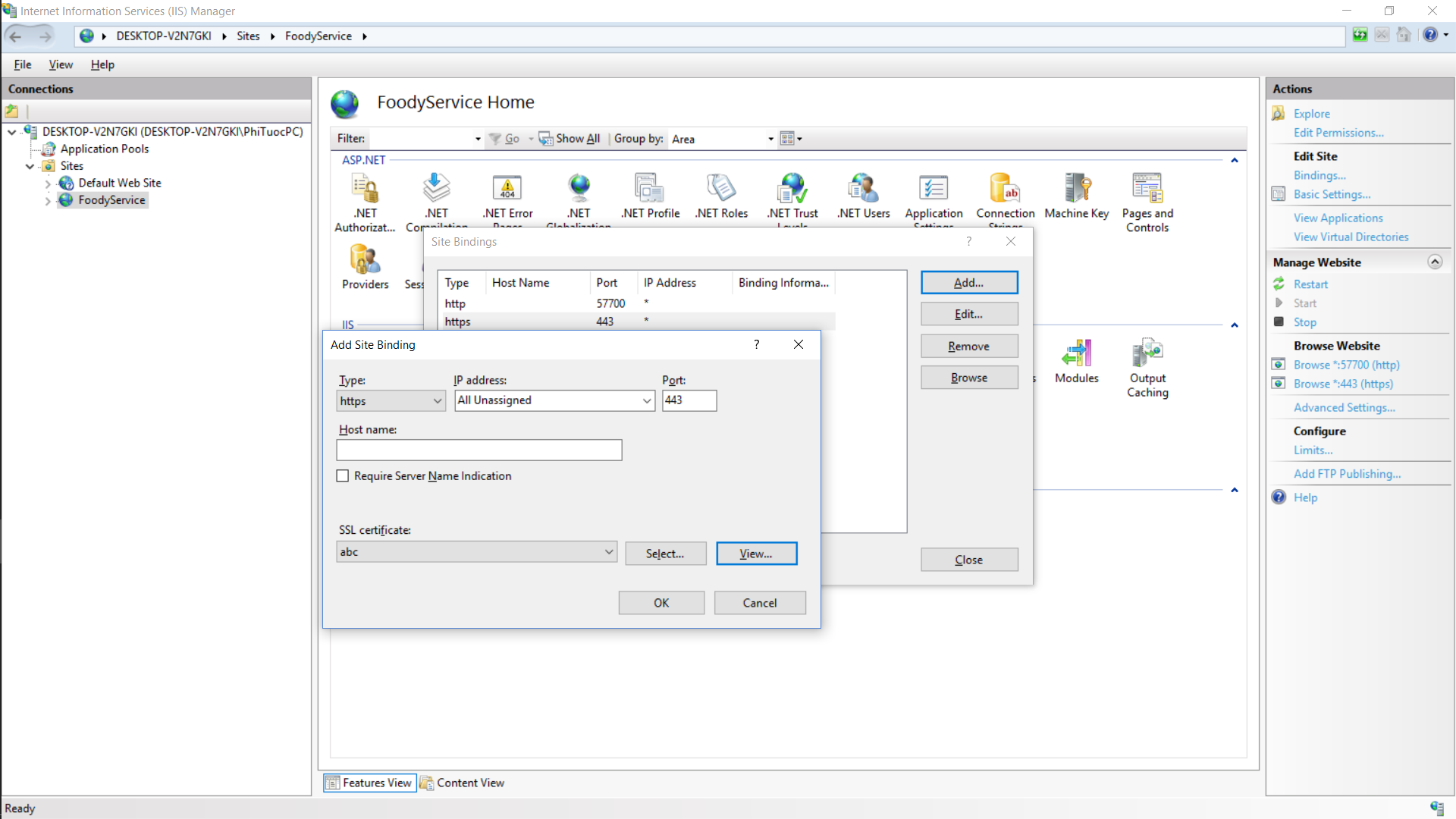


### B3: Binding port

* Mở sites vừa tạo và chọn edit bindings



* Chọn chứng nhận đã đăng ký thành công và chọn port chọn kiểu kênh truyền là https



### B4: Restore Database

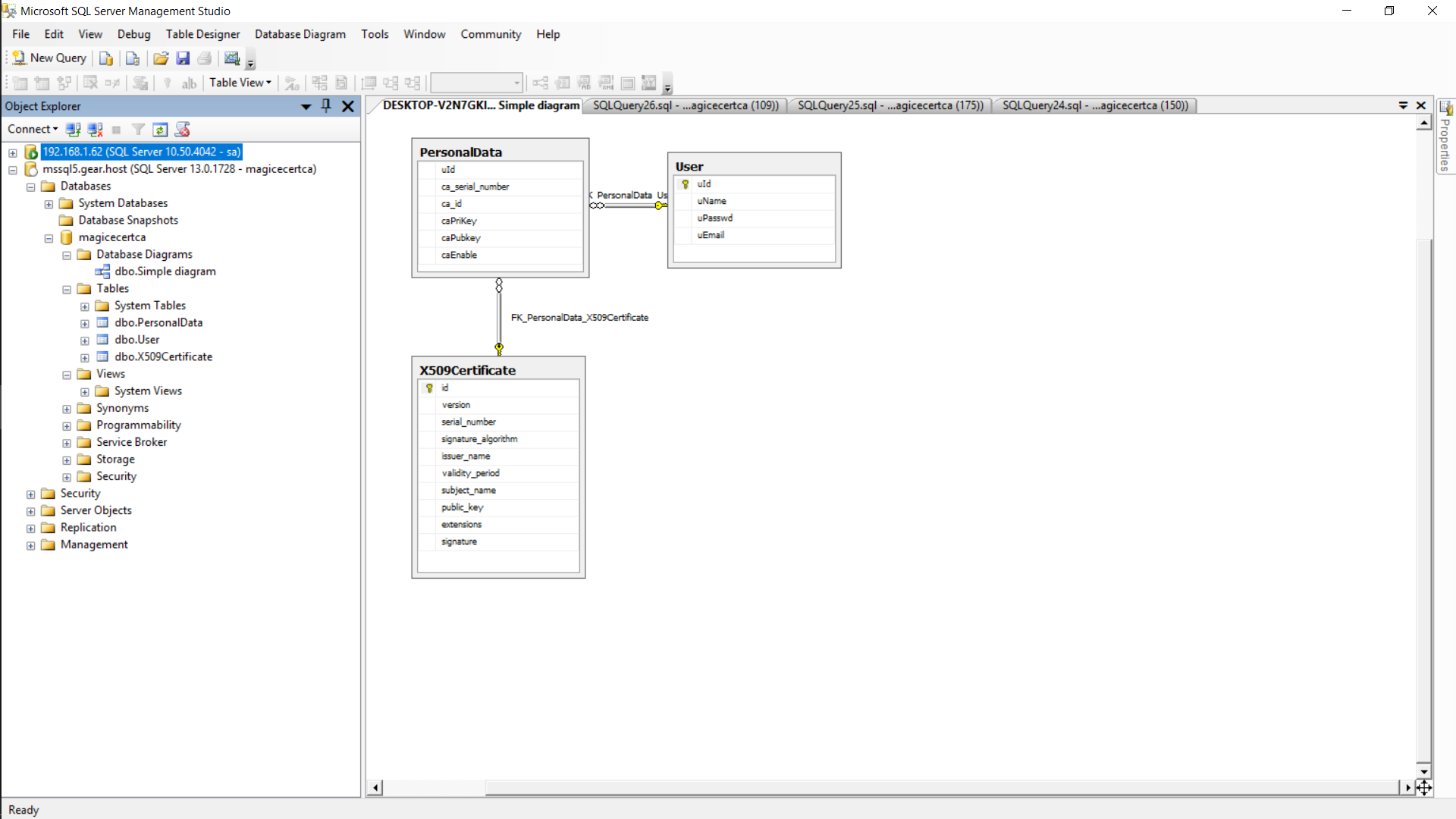
* Mở sql server và chọn file MagicECertCA.bak để restored vào database
* Kết nối tới databasecenter:
* Đối với <https://www.gearhost.com/>

Server type: database engine

Server name: mssql5.gear.host

Login: magicecertca

Passwd: Cw07j~mb7\_3J



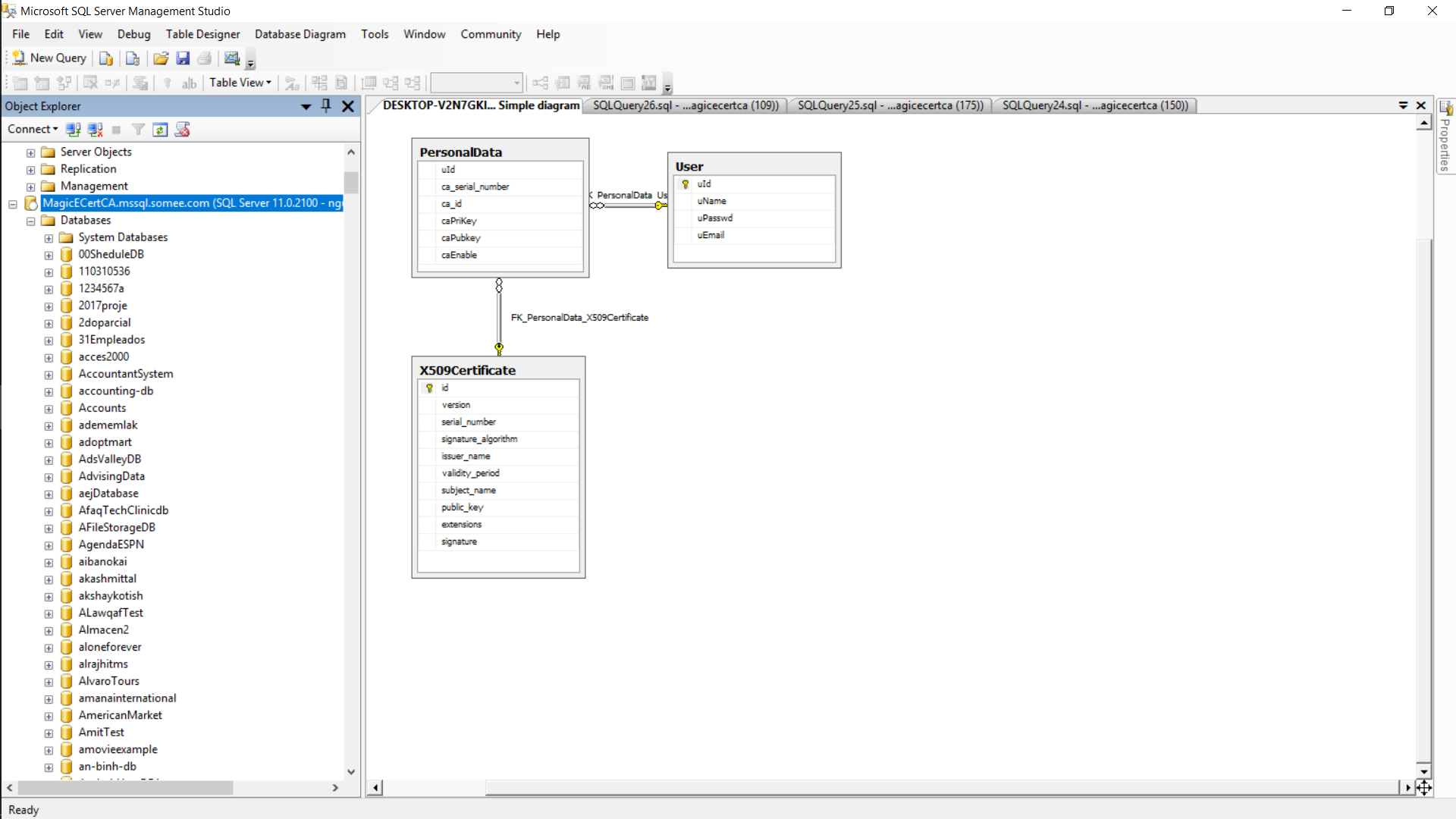
* Đối với soome.com

Server type: database engine

Server name: MagicECertCA.mssql.somee.com

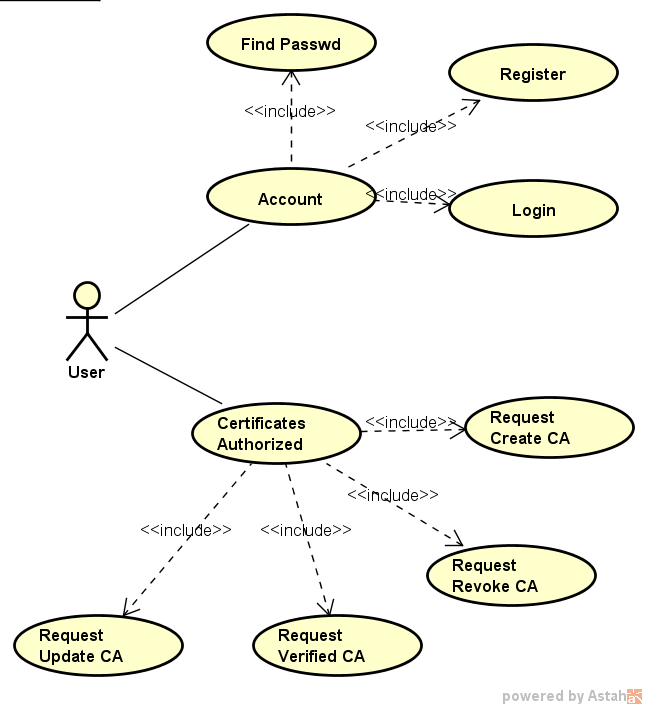
Login: nguyenvanphituoc\_SQLLogin\_1

Passwd: ixl3bpvla6



# Use-Case diagram

## Usecase Diagram



## Actors

|  |  |
| --- | --- |
| Đối tượng | Tên yêu cầu |
| User | * Đăng nhập * Đăng ký * Tìm kiếm mật khẩu * Yêu cầu cấp chứng nhận số * Yêu cầu hủy chứng nhận số * Yêu cầu cập nhật chứng nhận số * Yếu cầu tìm kiếm chứng nhận số * Yêu cầu kiểm tra chứng nhận số |
| Admin | * Backup database * Giữ keypairs |

## Use-case

|  |  |  |
| --- | --- | --- |
| NO | Use-case name | Yêu cầu/ ghi chú |
| U01 | Login | Đăng nhập |
| U02 | Register | Đăng ký |
| U03 | FindPasswd | Tìm kiếm mật khẩu |
| U04 | CreateCA | Tạo chứng nhận số |
| U05 | RevokeCA | Hủy chứng nhận số |
| U06 | UpdateCA | Cập nhật chứng nhận số |
| U07 | VerifiedCA | Kiểm tra chứng nhận số |
| U08 | SearchCA | Tìm chứng nhận số |
| A01 | BackupDatabase | Lưu trữ database |
| A02 | Manage Keypairs | Quản lý cặp khóa |

## Use-case name

### Use-case ID

|  |  |  |
| --- | --- | --- |
| Actor    Use Case | Admin | User |
| U01-U03 | x | x |
| U04-U08 |  | x |
| A01-A02 | x |  |

### Use-case Name

#### Login

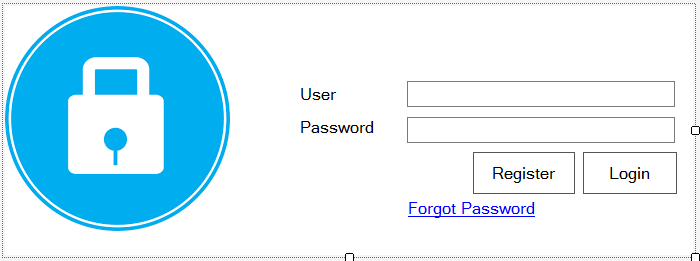
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Login | **Code** | U01 |
| **Description** | Cho phép actor đăng nhập vào hệ thống | | |
| **Actor** | Admin/User | **Trigger** | Actor bấm nút login/Actor bấm nút thoát/Actor bấm nút quên mật khẩu |
| **Pre-condition** |  | | |
| **Post condition** | Thông báo đã đăng nhập thành công | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập tên đăng nhập/ mật khẩu và click vào button Login trên trang đăng nhập (SC01). |  |  |
|  |  | 2 | Kiểm tra email đăng nhập/ mật khẩu là chính xác sau đó thông báo cho người dùng biết |

Design Screen



#### Register

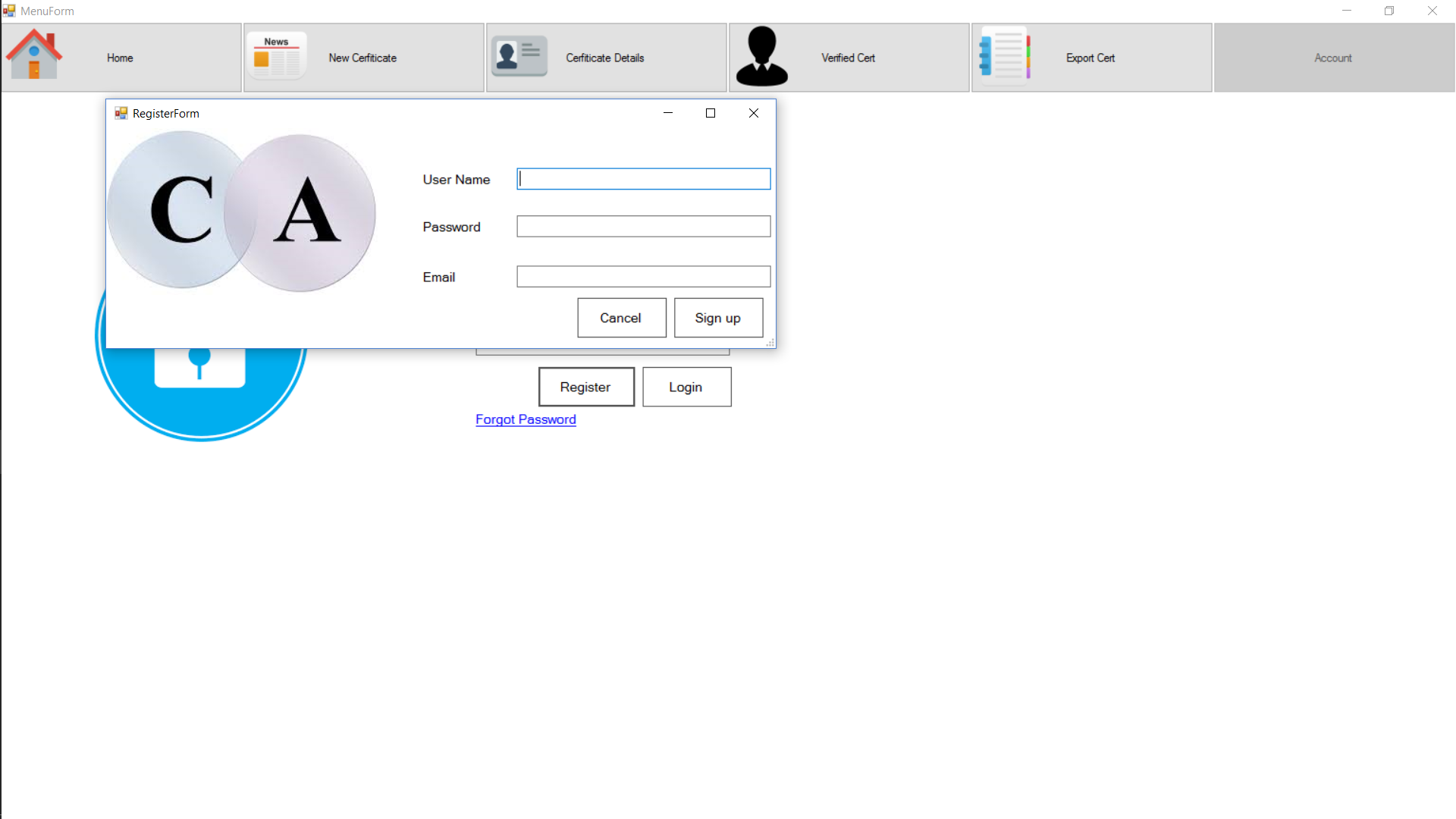
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Register | **Code** | U02 |
| **Description** | Cho phép actor đăng ký vào hệ thống | | |
| **Actor** | Admin/User | **Trigger** | Actor nhấn nút đồng ký |
| **Pre-condition** | User nhấn nút đăng ký | | |
| **Post condition** | Thông báo đã đăng ký thành công | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập vào tên người dùng, mật khẩu và email |  |  |
|  |  | 2 | Kiểm tra tên người dùng và email nếu thỏa thì trả kết quả đăng ký thành công |

Design Screen



#### CreateCA

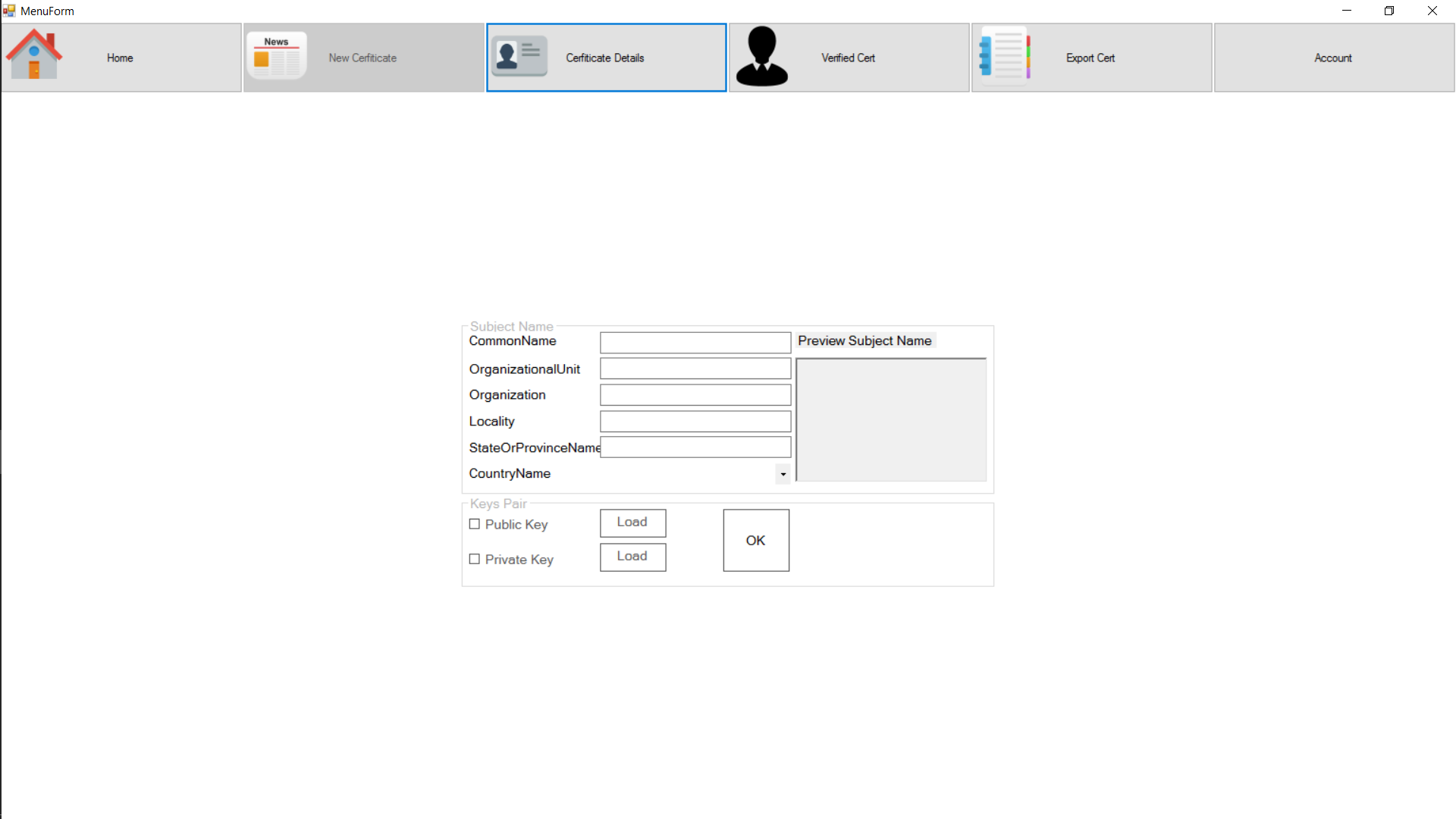
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | CreateCA | **Code** | U04 |
| **Description** | Cho phép actor tạo một chứng nhận điện tử | | |
| **Actor** | User | **Trigger** | User nhấn nút đồng ký tạo CA |
| **Pre-condition** | User Đã đăng nhập thành công vào hệ thống | | |
| **Post condition** | Thông báo đã tạo CA thành công | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor điền các trường tên để tạo ra tên riêng cho CA của mình |  |  |
|  |  | 2 | Tạo CA và quản lý khóa của nó trả kết quả thành công về cho Actor |

Design Screen



#### VerifiedCA

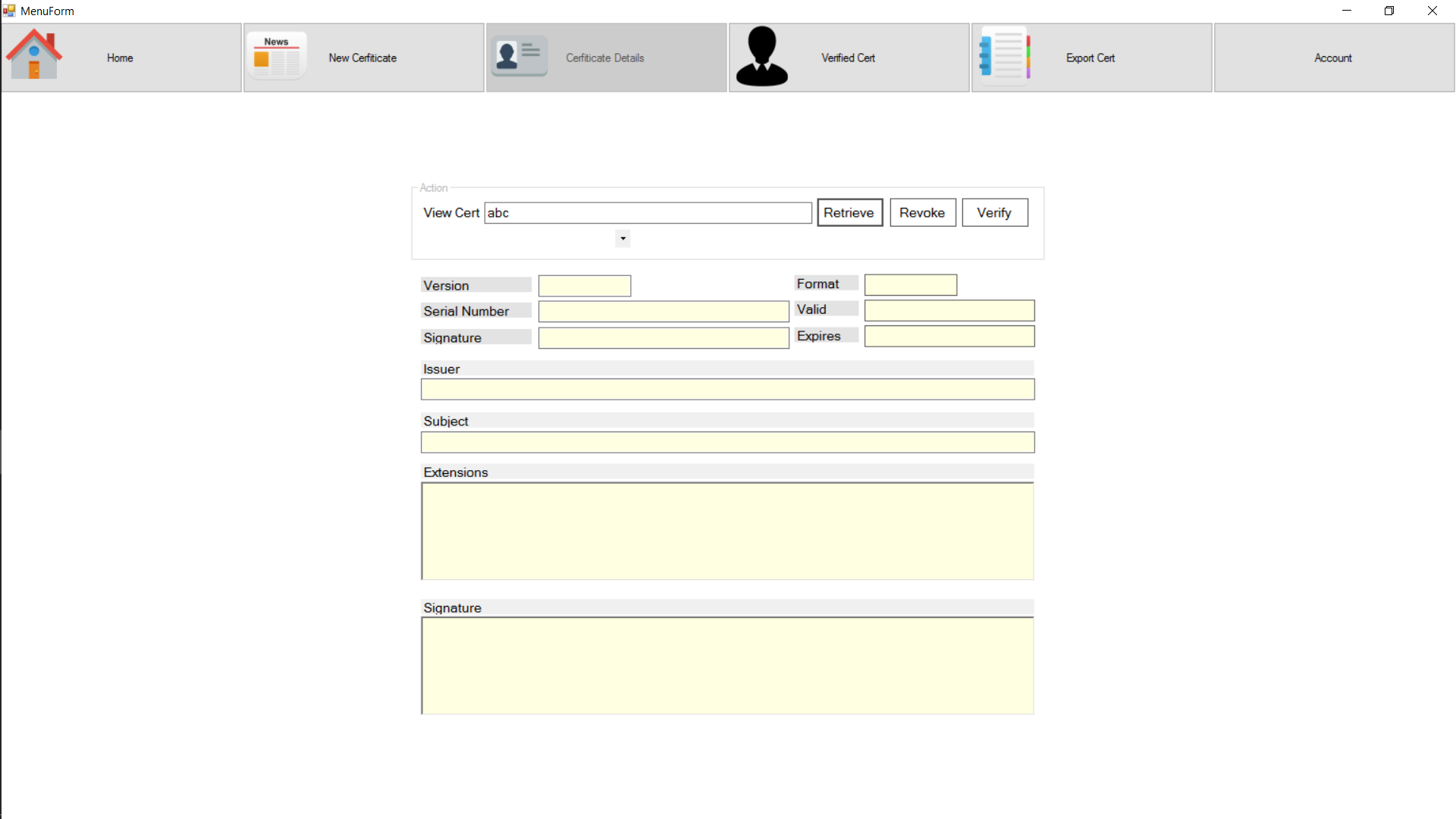
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | VerifiedCA | **Code** | U07 |
| **Description** | Cho phép actor kiểm tra một chứng nhận điện tử | | |
| **Actor** | User | **Trigger** | Actor bấm nút kiểm tra |
| **Pre-condition** | User đã đăng nhập vào hệ thống | | |
| **Post condition** | Thông báo kết quả kiểm tra | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor tìm kiếm và chọn CA muốn kiểm tra |  |  |
|  |  | 2 | Kiểm tra CA đó đúng với chữ ký bằng privateKey trên datacenter |

Design Screen



#### SearchCA

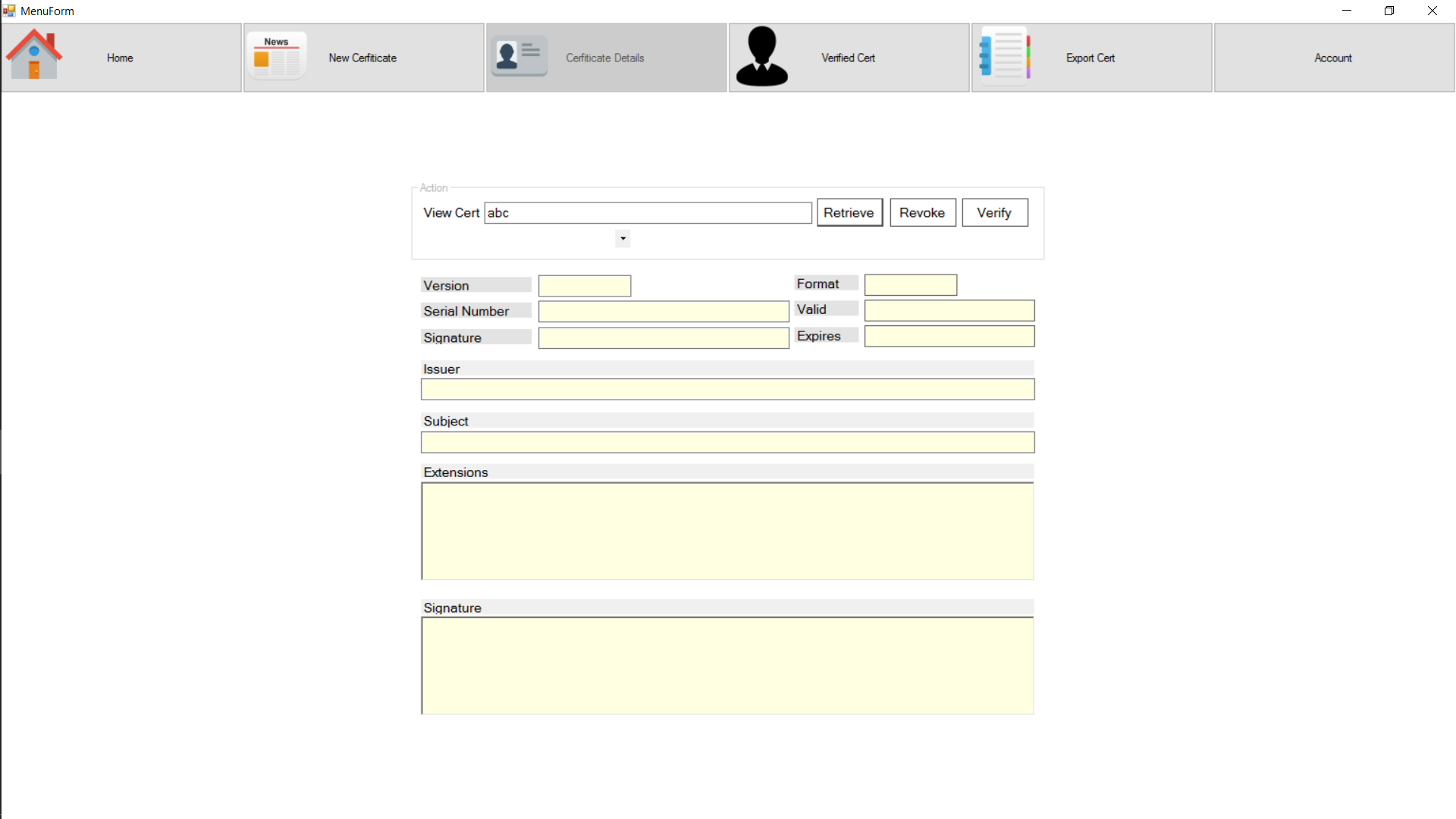
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | SearchCA | **Code** | U08 |
| **Description** | Cho phép actor tiềm kiếm CA trên hệ thống | | |
| **Actor** | User | **Trigger** | Actor bấm nút Retrieve |
| **Pre-condition** | User đã đăng nhập vào hệ thống | | |
| **Post condition** | Xuất ra list CA tìm được cho User xem thông tin | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập tên subjectName mà mình muốn tìm CA |  |  |
|  |  | 2 | Tìm kiếm CA dựa trên subjectName |

Design Screen



#### RevokeCA

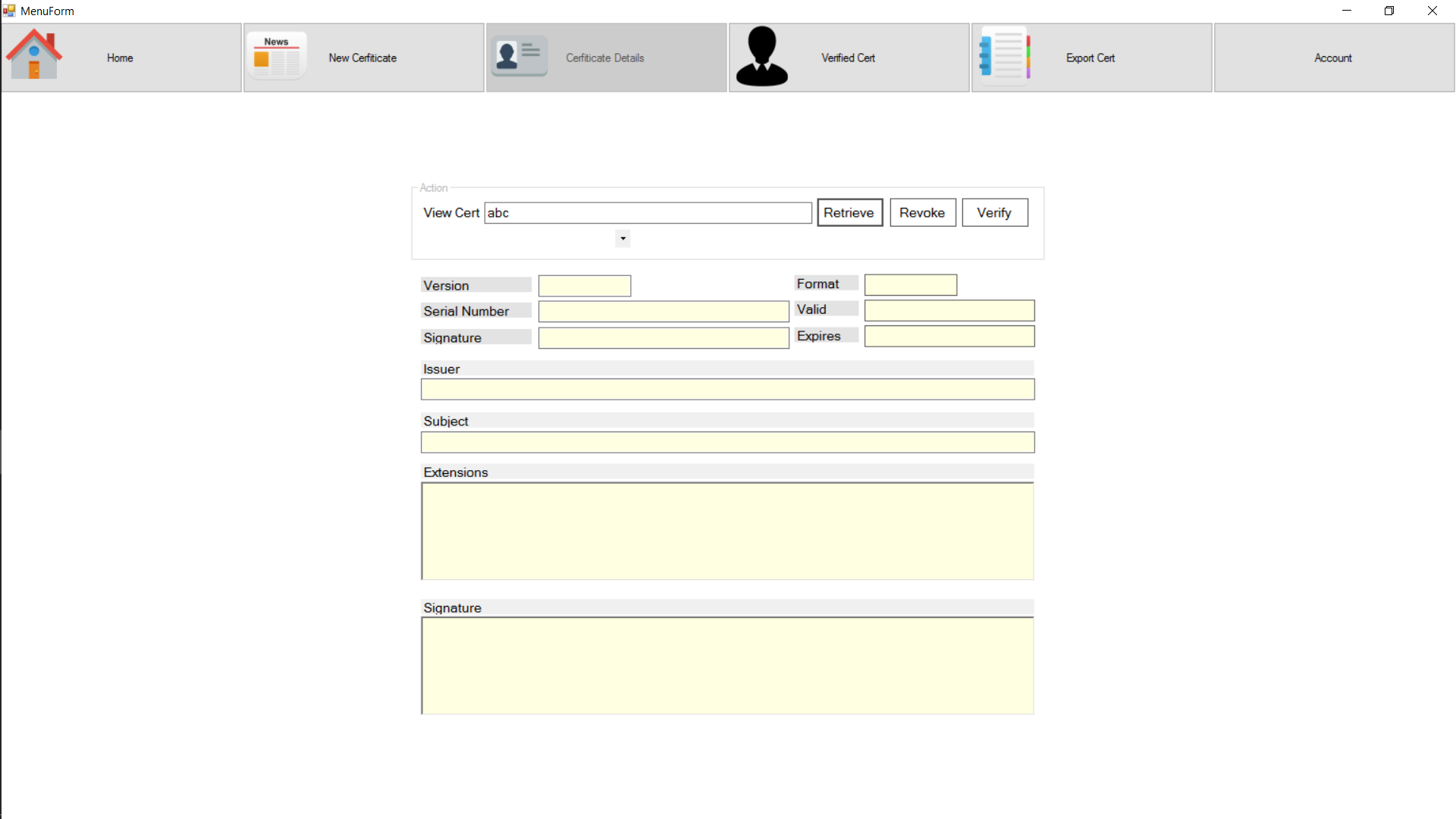
Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | RevokeCA | **Code** | U05 |
| **Description** | Cho phép actor hủy một chứng nhận tạm thời | | |
| **Actor** | Admin/User | **Trigger** | Actor bấm nút revoke |
| **Pre-condition** | User đã đăng nhập vào hệ thống | | |
| **Post condition** | Thông báo kết quả hủy chứng nhận | | |

Activities

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor chọn một chứng nhận của mình và hủy nó |  |  |
|  |  | 2 | Kiểm tra chứng nhận và hủy |

Design Screen



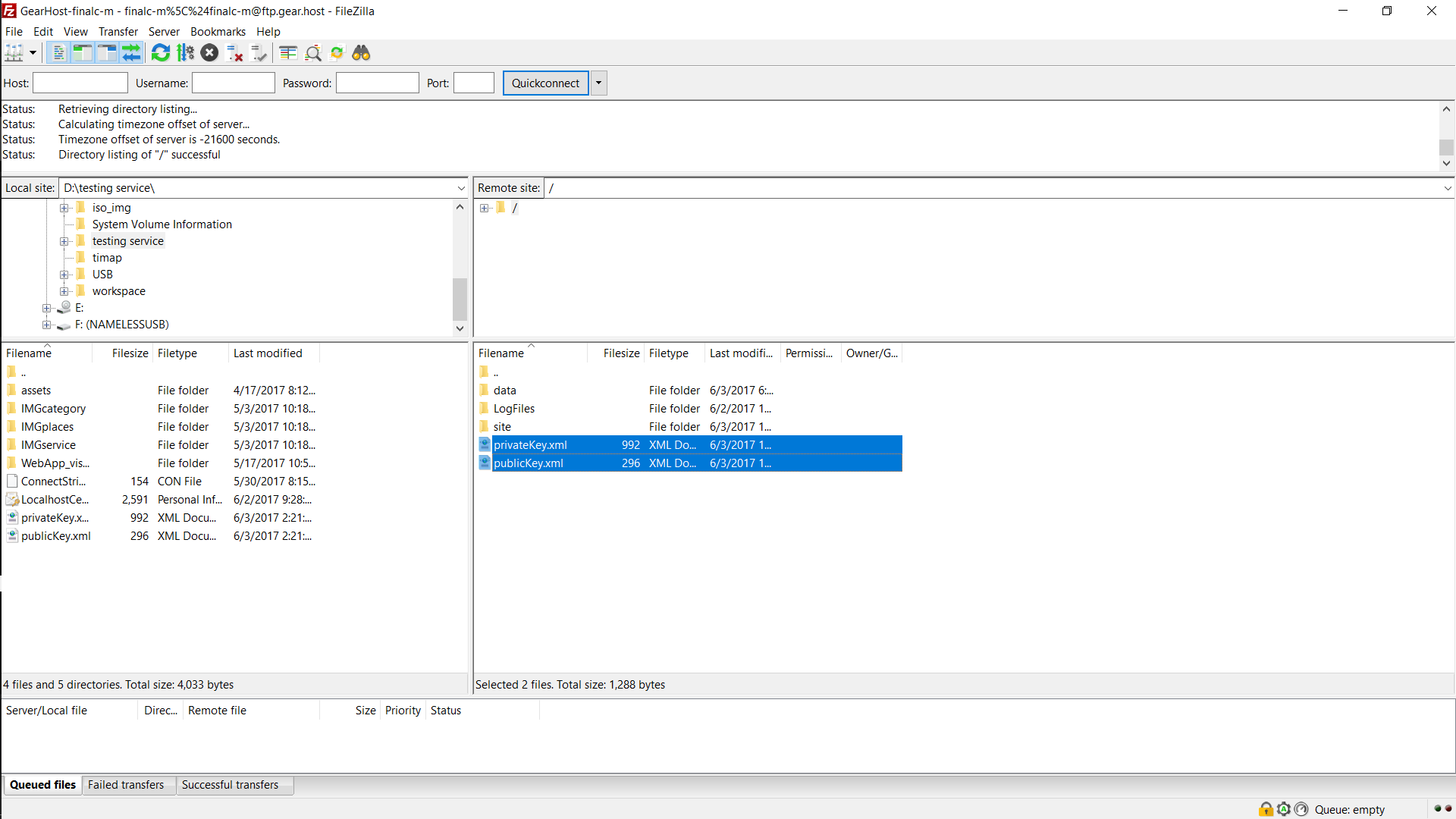
# Thiết kế hệ thống



SSL

Phía client: sẽ gửi request qua sever qua một trong hai đường dẫn sau :

(đường dẫn trên web vẫn chưa hoàn thiện, do chưa tìm được chỗ để cặp keypair, mặc dù đã đặt nó trên webroot)



static private string prefixOnWeb = "http://finalc-m.gear.host/FinalC-M/Values/";

. Nên ta sẽ chọn đường dẫn ở local

static private string prefixOnLocal = "https://localhost/Restful/FinalC-M/Values/";

Phía Client sẽ gửi đường dẫn đến Server thông qua https, để mở kênh truyền an toàn.

Phía Client sẽ nhận giá trị trả về từ Server thông qua chuỗi JSON.

Phía Sever sẽ nhận request và thực hiện nó trên datacenter, trong lúc này sẽ có sự tham gia của các thao toán mã giúp cho dữ liệu trên datacenter an toàn dù bị đánh cắp.

Ở đây Server sẽ dùng thuật toán mã RSA keyLengh là 128bytes được lưu trong 2 file privateKey.xml:

<?xml version="1.0" encoding="utf-8"?>

<RSAKeyValue>

<Modulus>hj6BWfYwkKDgTLPoFNRs49FRXIlceqTyRAiLDd6utyZlKrFYvHSc9CCGMYFY+BfAvTIxX1RbXFmT4K51cWaKML7Nt0b9/ugt8lCifilmaCXZ7Bs23BELR+9ehhNVwFPQhVMdVXDr00lPPxz+xZIzgGTTmxbkXRIh1CF5Xln7z2c=</Modulus>

<Exponent>AQAB</Exponent>

<P>tZ156HHaHavTlb6MPVeyJw7+OWj0uB6Kh0pM6YxxVeL60jHR0EZS83jAaGGgMTvMVpIyU6s/HiDZT+OLnsvmpw==</P>

<Q>vToixn4h7lyszV62BEbCHc5c7HLkitj6mDHZupca/YhIxFktVJg4ORDnQ8wP3Z0E2zbGwWd+sPi8Abl5e76pQQ==</Q>

<DP>FizOL8WG/kx2nzgO44aLqB06iUP5bWm9PFukHHgbdoMqYPPYTHfptPSveY9CAgDtKDzdjD3CjSkxn4GHELKQzQ==</DP>

<DQ>pwpz+l8MGvvqe1y5NuszntY1x3Dg+QTzqniPKJ0NcXPmMJY8YuoA77dRP9Maylo7GDyxlbeD6juY07jkWe2EQQ==</DQ>

<InverseQ>XP3QcZZ2ddO4t813EZOTc4fNS62DEzoQfJo4nOmO5Y4tZ3k3WmYc6P5W62EUqjzJI9QUWwexbyfnZqlIj5uadw==</InverseQ>

<D>G/wfvv9BdIT0/Uhr0xvNb1iNW3sPGkoUbvZQ9p07UtbgNBwwYWj15zZ4gw1x3JYnCcnS3Qqa2Q+bcjUsKm2z1NRDA5w7H4GBFNTs92YEGuUGAboENPt8xxnLGRX3IzwJ38BNyUulYSkOT9mm3JM7blZ3mvpCFq9EosN2Ley50QE=</D>

</RSAKeyValue>

và publicKey.xml:

<?xml version="1.0" encoding="utf-8"?>

<RSAKeyValue>

<Modulus>hj6BWfYwkKDgTLPoFNRs49FRXIlceqTyRAiLDd6utyZlKrFYvHSc9CCGMYFY+BfAvTIxX1RbXFmT4K51cWaKML7Nt0b9/ugt8lCifilmaCXZ7Bs23BELR+9ehhNVwFPQhVMdVXDr00lPPxz+xZIzgGTTmxbkXRIh1CF5Xln7z2c=</Modulus>

<Exponent>AQAB</Exponent>

</RSAKeyValue>

Connection String phía Sever:

Data Source=mssql5.gear.host;Initial Catalog=MagicECertCA; UserID=magicecertca; Password=Cw07j~mb7\_3J; connect timeout=60; MultipleActiveResultSets=True; App=EntityFramework

(connect đến datacenter gearhost)

workstation id=MagicECertCA.mssql.somee.com;packet size=4096;user id=nguyenvanphituoc\_SQLLogin\_1;pwd=ixl3bpvla6;data source=MagicECertCA.mssql.somee.com;persist security info=False;initial catalog=MagicECertCA;connect timeout=60;MultipleActiveResultSets=True;App=EntityFramework

(connect đến datacenter soome host)

Phía datacenter, có một số trường dữ liệu được mã hóa(passwd, và email):



Thông tin của chứng nhận cũng được mã hóa một số trường(serialnumber).

# Kết luận

* Mặc dù hệ thống còn chưa hoạt động tốt trên online, tuy nhiện hệ thống vẫn chạy tạm ổn trên local với các chức năng đơn giản.
* Người thực hiện đề tài đã nắm được căn bản cách thức bảo mật và giúp an toàn trên mạng, sử dụng được những kiến thức đã học
* Hệ thống đảm bảo tính bảo mật từ yêu cầu bài toán: liên lạc từ client đến server trên kênh truyền ssl, dữ liệu lưu trên server được mã hóa, cặp keypairs mã hóa được cất ở nơi bí mất(do administrator nắm giữ).
* Hệ thống cần được nghiên cứu kỹ càng hơn để chạy tốt trên online, cần cải thiện vấn để mở socket ssl.

# Thông tin thêm

## Tài liệu tham khảo

<https://youtu.be/zTlc5Z59FpA> (Link cấu hình ssl cho IIS Server)

<https://www.codeproject.com/Articles/12893/TCP-IP-Chat-Application-Using-C>

(Tham khảo ứng dụng mở socket)

<http://finalc-m.gear.host/> (Server host online)

<https://drive.google.com/open?id=0B5ZjEXnn17D5Z3pSY2gxZVc3MEk> (Tài nguyên về host: pushlish.xml, [ftp.xml](ftp://ftp.xml), cặp key pairs)

<https://www.gearhost.com/documentation/how-to-publish-your-app-with-ftp> (cách để pushlish file)

<https://www.gearhost.com/documentation/how-to-publish-your-app-from-visual-studio> (cách pushlish app từ visual studio)