



## CAPSTONE PROJECT FINAL REPORT



VTCHD Team	
<b>Group Members</b>	SE03669 – Nguyễn Sơn Vũ SE03706 – Tạ Quang Huy SE03817 – Đỗ Ngọc Trinh SE03806 – Hà Phan Anh Công SE03875 – Bùi Vinh Đạt SE03880 – Vũ Trí Trung
<b>Supervisor</b>	Lecturer: Nguyễn Tất Trung
<b>Capstone Project code</b>	MD

Hanoi, August 21<sup>th</sup>, 2017

## Table of contents

<b>Table of contents.....</b>	<b>2</b>
<b>Acknowledgements.....</b>	<b>5</b>
<b>Definitions and Acronyms .....</b>	<b>6</b>
<b>Chapter 1 : Introduction .....</b>	<b>7</b>
1.1    Purpose.....	7
1.2    Project Information .....	7
1.3    The people.....	7
1.3.1    Supervisors .....	7
1.3.2    Team members .....	7
1.4    Background.....	7
1.4.1    Problems.....	7
1.4.2    The current system .....	10
1.5    The proposal system .....	12
1.5.1    System functions .....	12
1.5.2    Technical approach .....	13
1.5.3    Out of scope function .....	13
<b>Chapter 2 : Project Plan .....</b>	<b>14</b>
2.1    Project Organization .....	14
2.1.1    Purpose .....	14
2.1.2    Software Process Model.....	14
2.1.3    Roles and Responsibilities .....	15
2.1.4    Organization Structure .....	15
2.1.5    Project Team Member.....	15
2.1.6    Tools and Techniques .....	15
2.2    Project Management Plan .....	16
2.2.1    Tasks.....	16
2.2.2    Meeting Minutes .....	16
2.2.3    Coding Conventions.....	18
2.2.4    Risk Management Plan .....	18
2.2.5    Communication Plan .....	18
<b>Chapter 3 : Software Requirement Specification .....</b>	<b>20</b>
3.1    Purpose.....	20
3.2    Functional Requirements .....	20
3.2.1    Use Case Diagram.....	20
3.2.2    Business Rules .....	22

3.2.3	Use Cases .....	23
3.3	Non-functional Requirement .....	99
3.3.1	Security.....	99
3.3.2	Maintainability & Extensibility.....	101
3.3.3	Availability and Scalability.....	101
3.3.4	Performance .....	101
3.3.5	Usability .....	102
3.4	Entity Relationship Diagram.....	102
<b>Chapter 4 : Software Design .....</b>	<b>103</b>	
4.1	Purpose.....	103
4.2	Architecture Overview .....	103
4.2.1	System Architecture .....	103
4.2.2	System Architecture Explanation.....	104
4.3	Design of MindDoctor System .....	107
4.3.1	Architecture Layers Design.....	107
4.3.2	Database Design.....	109
4.3.3	Common Design.....	119
4.3.4	Detail Design.....	125
<b>Chapter 5 : Software Testing Documentation.....</b>	<b>310</b>	
5.1	Introduction .....	310
5.1.1	Purpose .....	310
5.1.2	Scope of testing .....	310
5.2	Test plan .....	311
5.2.1	Testing tools and environment .....	311
5.2.2	Resources and responsibilities .....	312
5.2.3	Test strategy .....	313
5.2.4	Features to be tested .....	315
5.3	Test Case .....	317
5.3.1	Automation test Unit testing .....	317
5.3.2	System testing .....	319
5.3.3	Defect Log.....	319
5.4	Test Report .....	320
5.4.1	Automation test case report.....	320
5.4.2	Automation test report.....	321
5.4.3	System test case report .....	321
5.4.4	System test report .....	328

5.4.5	Benchmarks .....	329
5.4.6	Usability in mobile .....	330
5.4.7	Stress test.....	331
<b>Chapter 6 : User manual .....</b>	<b>332</b>	
6.1	Deployment guidelines .....	332
6.1.1	Microsoft Azure .....	332
6.1.2	SQL Server 2012 Enterprise .....	334
6.2	User guidelines.....	335
6.2.1	Sign in MindDoctor system .....	335
6.2.2	Update profile.....	336
6.2.3	Interact with post.....	337
6.2.4	View Advisors.....	340
6.2.5	Set up free time (Only for Advisor account).....	342
6.2.6	Make an appointment with Advisor.....	344
6.2.7	Make a conversation with advisor.....	345

## Acknowledgements

*This project was completed thanks to the support from many people; we would like to sincerely thank all of them.*

*Foremost, we wish to express our extreme gratitude to Mr. Nguyen Tat Trung, our supervisor. If it was not for his continuous support to instruct and rally the team, we could not have done this. He had always pointed us to the right direction and given us technological and methodological insights that tremendously helped the team.*

*We would like to thanks to all teachers, educators at FPT University and FPT University as a whole. The knowledge you had taught us is critical to the completion of this project. We hope that this project reflects what we are taught over the years at FPT University.*

*There are also many unnamed people that contribute to the project: Our friends and our families... Thanks to all of you from the bottom of our heart.*

## Definitions and Acronyms

Acronym	Definition	Note
FU	FPT University	
MD	MindDoctor	
SRS	Software Requirement Specification	
API	Application Programming Interface	
PM	Project manager	
BA	Business Analysis	
UC	Use Case	
N/A	Not Available	
URL	Uniform Resource Locator	
HTTP	Hypertext Transfer Protocol	
GUI	Graphical User Interface	
UI	User Interface	
UX	User experiences	
GPC	Gold per conversation	
MD	MindDoctor	
OWIN	Open Web Interface for .NET	
No.	Number of	
VM	Virtual Machine	
JSON	Javascript Object Notation	
SMTP	Simple Mail Transfer Protocol	
IIS	Internet Information Service	
RAM	Random Access Memory	

# Chapter 1 : Introduction

## 1.1 Purpose

This chapter provides an overview of the project include background information, a literature review of the existing system and raising a proposal for ideas of improvement.

## 1.2 Project Information

- Project name: **MindDoctor**.
- Project code: **MD**.
- Description: A website provides services for easier connection and expertise between people and experts in many other fields.
- Project group name: **VTCHD**.
- Product type: **Web application**.
- Timeline: **From May 8<sup>th</sup>, 2017 to August 21<sup>th</sup>, 2017**.

## 1.3 The people

### 1.3.1 Supervisors

	Full name	Phone	E-Mail	Title
Supervisor	Nguyễn Tất Trung	090 439 9139	<a href="mailto:trungnt@fpt.edu.vn">trungnt@fpt.edu.vn</a>	Lecturer

Table 1-1: Supervisor's information

### 1.3.2 Team members

	Full name	Code	Phone	E-mail	Role
1	Bùi Vinh Đạt	SE03875	0166 360 5557	<a href="mailto:datbvse03875@fpt.edu.vn">datbvse03875@fpt.edu.vn</a>	Leader
2	Hà Phan Anh Công	SE03806	098 937 6395	<a href="mailto:conghpase03806@fpt.edu.vn">conghpase03806@fpt.edu.vn</a>	Member
3	Vũ Trí Trung	SE03880	098 744 5154	<a href="mailto:trungvtse03880@fpt.edu.vn">trungvtse03880@fpt.edu.vn</a>	Member
4	Tạ Quang Huy	SE03706	096 623 6917	<a href="mailto:huytqse03706@fpt.edu.vn">huytqse03706@fpt.edu.vn</a>	Member
5	Nguyễn Sơn Vũ	SE03669	0166 323 4748	<a href="mailto:vunsse03669@fpt.edu.vn">vunsse03669@fpt.edu.vn</a>	Member
6	Đỗ Ngọc Trinh	SE03817	098 974 0361	<a href="mailto:trinhdnse03817@fpt.edu.vn">trinhdnse03817@fpt.edu.vn</a>	Member

Table 1-2: Team member's information

## 1.4 Background

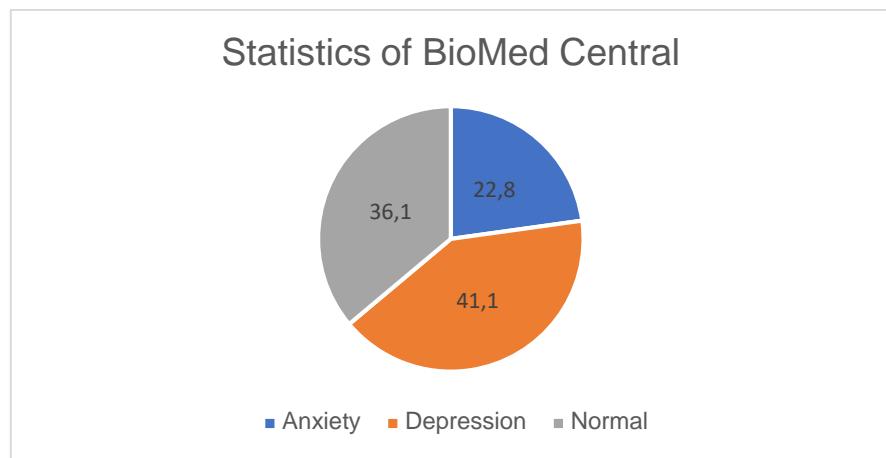
### 1.4.1 Problems

#### 1.4.1.1 Psychology

Viet Nam is a country with high rate of people suffering from depression in the world. According to World Health Organization in 2015, Viet Nam has 3.6 million depression people, it is about 4.0% of Viet Nam population.

In particular, the rate of adolescents and young people are increasing by the time. According to researching of BioMed Central, in 1161 secondary students of Can Tho, there are 22.8% anxiety, 41.1% depression. On the other hand, suicide had been seriously considered by 26.3% of the students, while 12.9% had made a suicide plan and 3.8% had

attempted suicide. Major risk factors related to anxiety and depression were physical or emotional abuse by the family and high educational stress.

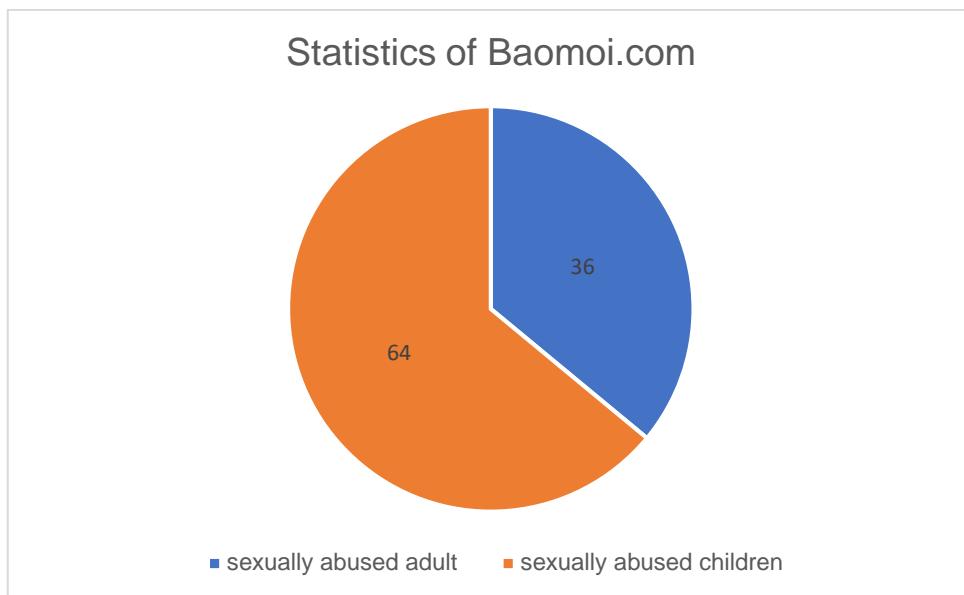


*Figure 1-1: Provided by BioMed Central  
(<https://bmcpublichealth.biomedcentral.com/articles/10.1186/1471-2458-13-1195>)*

However, the care services and supports for depression has not been heightened because the human resources in psychological domain is not much and some current systems is not really meet the needs of customers.

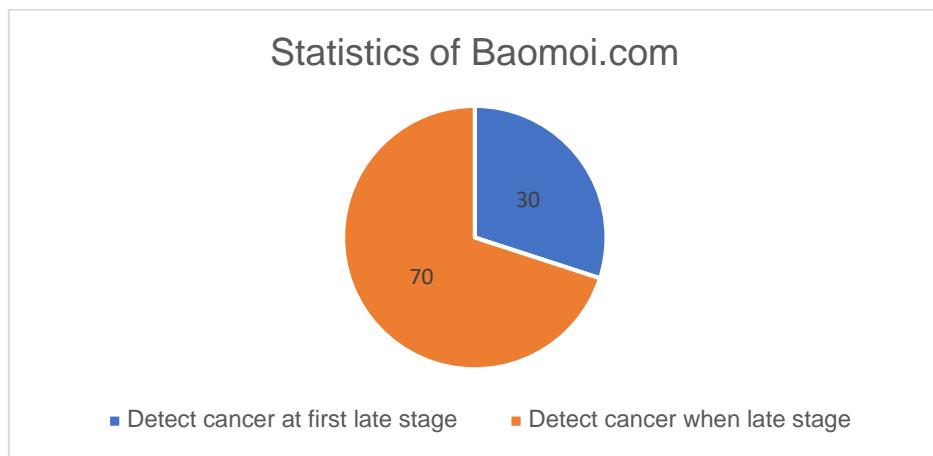
#### 1.4.1.2 Health

Viet Nam is a country with high rate sexually abused children. According to data from Baomoi magazine, in 5 years (2011-2015) Viet Nam has 5300 sexually abused children case in 8200 sexually abused case. Statistics show the lack of knowledge about sex education of Vietnamese student. It easy to understand because sex education is not an official subject in school.



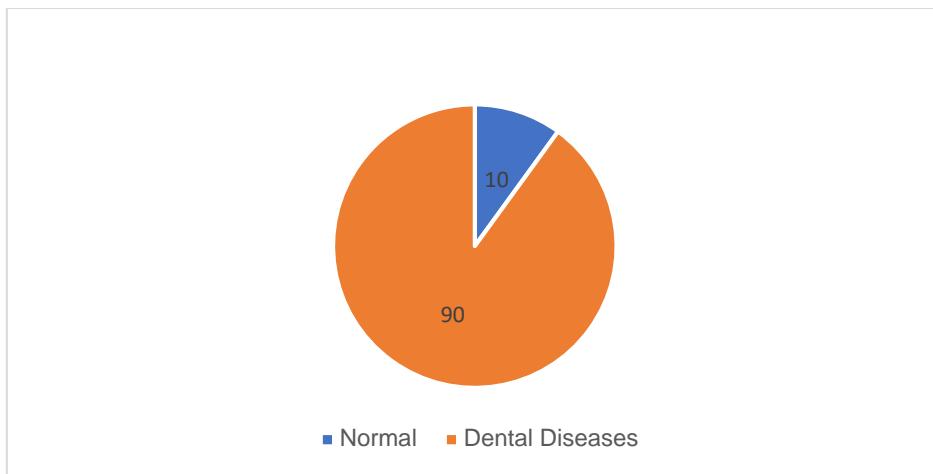
*Figure 1-2: Data provided by Baomoi magazine  
(<http://www.baomoi.com/giao-duc-gioi-tinh-trong-nha-truong-nen-day-gi-de-tranh-ve-duong-cho-huou-chay/c/2171995.epi>)*

70% cancer people detect their disease when late stage (data from survey of national cancer program)



*Figure 1-3: Data provided by survey of national cancer program  
(<http://www.baomoi.com/70-benh-nhan-ung-thu-phat-hien-o-giai-doan-muon/c/4976405.epi>)*

and 90% Vietnamese people have dental diseases (according of the central orthopedic dental - facial hospital).



*Figure 1-4: Data provided by the central orthopedic dental-facial hospital  
(<http://suckhoe.vnexpress.net/tin-tuc/suc-khoe/90-nguo-viet-mac-benh-rang-mieng-2432736.html>)*

The main reason is Vietnamese people do not have right knowledge about health, do not get regular health care and do not have time to see doctor.

#### 1.4.1.3 Traditional consulting process

The current consulting process can be visualized as the below figure. This process causes many problems for both customer and consultant.

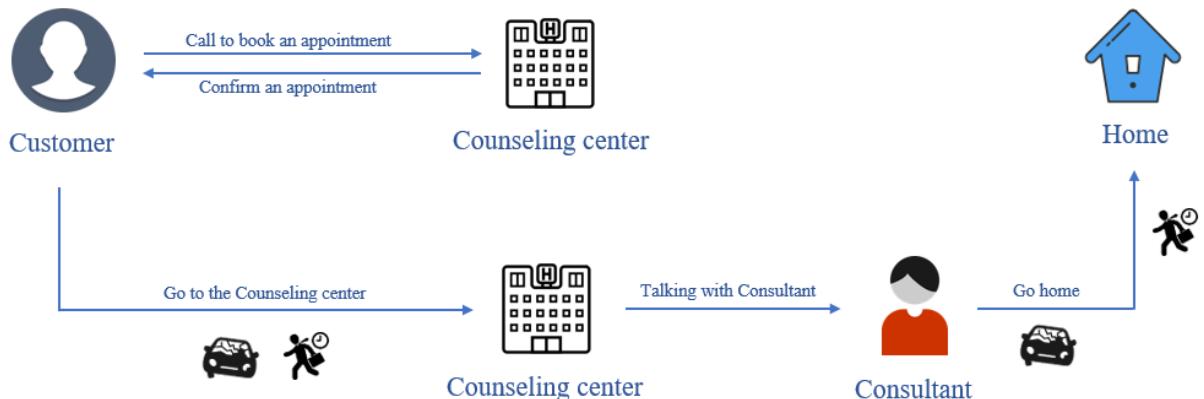


Figure 1-5: Traditional consulting process

#### 1.4.1.4 Conclusion

Psychology and health are the biggest asset for everyone. Having good psychology and good health, person's life will be better. A powerful country is a powerful people.

Therefore, MindDoctor was built to improve the poor psychology and poor health of Vietnamese people by provide the trustable information, service for easier connection and expertise with experts in other fields. When using MindDoctor system, user will reduce waste time, travel fee and avoid crash traffic. Consultant can do other things instead of waiting customer.

### 1.4.2 The current system

#### 1.4.2.1 Overview of current system

Opened in November 2016, Ecomedic is the first social network about health care in Viet Nam and provides health service to people of any ages, anytime, anywhere. It allows searching clinics, doctors and also ask question about heath, diseases with the professional doctors team of Ecomedic.



Figure 1-6: Snap shot of ecomedic.vn

### 1.4.2.2 Problem of current system

#### 1.4.2.2.1 Do not have search questions

Ecomedic is a large web site with many question in many different domains. However, it does not have a search function, so that user needs to spend too much time for finding the right question to solve their problems in thousands of questions.

Bệnh amiđan | 15:46 11/08/2017 | Tai mũi họng

cần giải đáp về bệnh liên quan da liễu | 09:32 10/08/2017 | Da liễu

xương tay | 09:27 04/08/2017 | Cơ Xương Khớp

Xét nghiệm Tiểu đường sau sinh | 08:38 04/08/2017 | Nội tiết

Biện pháp trong phòng và tránh thai | 21:15 31/07/2017 | Sản phụ khoa

Viêm tai giữa | 10:33 27/07/2017 | Tai mũi họng

Răng khôn và các vấn đề liên quan | 00:34 26/07/2017 | Răng hàm mặt

Đau dưới lòng bàn chân | 08:03 25/07/2017 | Da liễu

Kham thai 13 tuan | 16:27 22/07/2017 | Nhi

Hiển thị 10 trong tổng số 368 kết quả tìm kiếm

1 2 3 4 5 6 7 8 9 10 11 Trang sau

Figure 1-7: Snap shot of questions in ecomedic.vn

#### 1.4.2.2.2 Wait for responding

Ecomedic can not answer question from the user immediately or in a short time, because it does not have a doctor team always online and ready to answer questions. So it can not solve the problems of user in emergencies case.

Nguyễn | ngày gửi : 21:43 15/08/2017 | Cơ Xương Khớp

Chào bác sĩ.  
Tôi năm nay 30 tuổi. Cách đây hơn 2 năm tôi có bị đau vùng thắt lưng vào tầm 3h sáng trở đi,lúc này tôi không ngủ được,trần trác trở mình thi chí đỡ được một lúc rồi lại đau. Đến sáng dậy tôi đi lại thì cơ đau dần biến mất và ban ngày thi thường như không cảm thấy đau đớn gì,tôi vẫn làm việc như bình thường và không hề có cảm giác đau. Tôi có đi khám rất lần trên nhiều bệnh viện ở Hà Nội,chụp X quang,siêu âm ổ bụng,xét nghiệm máu,thử nước tiểu...Có lần bác sĩ nói tôi bị viêm khớp,có lần nói đang thoái hóa cột sống thắt lưng,rồi có lần nghỉ ngơi chức năng thận.Gần đây tôi có đi chụp X quang lại tại bệnh viện Bạch Mai thi bác sĩ nói cốt sống bình thường. 1 năm trở lại đây tôi cũng chỉ làm việc bình thường nhẹ nhàng chứ không làm việc nặng nhọc. Tôi cũng đã điều trị thuốc theo chỉ dẫn của bác sĩ thăm khám nhưng không thấy có hiệu quả gì. Và hiện nay các cơn đau vẫn hành hạ tôi mỗi khi gần về sáng. Giờ đây tôi rất hoang mang và không biết nên khám chữa như thế nào,khám ở đâu? Rất mong các bác sĩ tư vấn giúp tôi!

Tôi xin chân thành cảm ơn!

Like 0 Share

#### Bình luận

Hỗ trợ Ecomedic | ngày gửi : 21:43 15/08/2017  
Cảm ơn bạn đã sử dụng dịch vụ của Ecomedic. Câu hỏi của bạn đã được gửi tới Đội ngũ Bác sĩ Ecomedic và sẽ sớm được phản hồi lại

Figure 1-8: Snap shot of question in ecomedic.vn

## 1.5 The proposal system



Figure 1-9: MindDoctor's proposal system

### 1.5.1 System functions

The list below summarizes the main functions of MindDoctor and what is in the scope of the project. Detailed actions and its implementations will be listed in chapter 4:

- Allow basic user function: Register / Login / Log out / Modify Password and Profile.
- Allow to login by social account: Facebook.
- For guests/visitors:
  - Full-text search functionality on multiple fields: post title, post content, post introduction...
  - Register to become an Advisor.
  - See Advisors online list.
- For customers have all functionalities of guest and:
  - Making an conversation with Advisor via chat text and video call.
  - Like, unlike and comment post which is interested.
  - Booking an appointment with Advisor.
  - Insert money for tipping or consulting.
  - View conversation history, inserted money history.
- For advisors have all functionalities of guest and:
  - Making an conversation with Advisor via chat text and video call.
  - Assess, like, unlike or comment post.
  - Set free time for schedule.

- View list of customers booking.
- Admin have functionalities:
  - Manage posts, domains, subdomains, finances...
  - Manage advisor and customer, conversation, pay and repay money.
- On home page, posts will be displayed base on what customer interested in (like, search) and nearly consulted.
- On home page, online advisors are displayed order by number of star, number of conversation and GPC.

### **1.5.2 Technical approach**

- Using MVC design pattern.
- Recommend posts base on customer behavior.
- Website is responsive that is not broken to the degree of unused at different popular screen sizes.
- Realtime using Firebase.
- Full-text search with SQL Server.
- Windows services to manage appointment between customer and advisor.
- Incorporate multiple thirds party services to improve user experiences, notably Mailboxlayer, currencylayer, numverify, Paypal, OpenTok...
- Caching data on client to increase performance.

### **1.5.3 Out of scope function**

The list below defines the boundary of current iteration of MindDoctor. These functions will not be included in this version, but not permanently excluded in future versions.

- Application on mobile phone.
- Super recommendation system for posts and advisors based on conversation history and behavior.
- Transfer money via Bank automatically.
- Chat bot.

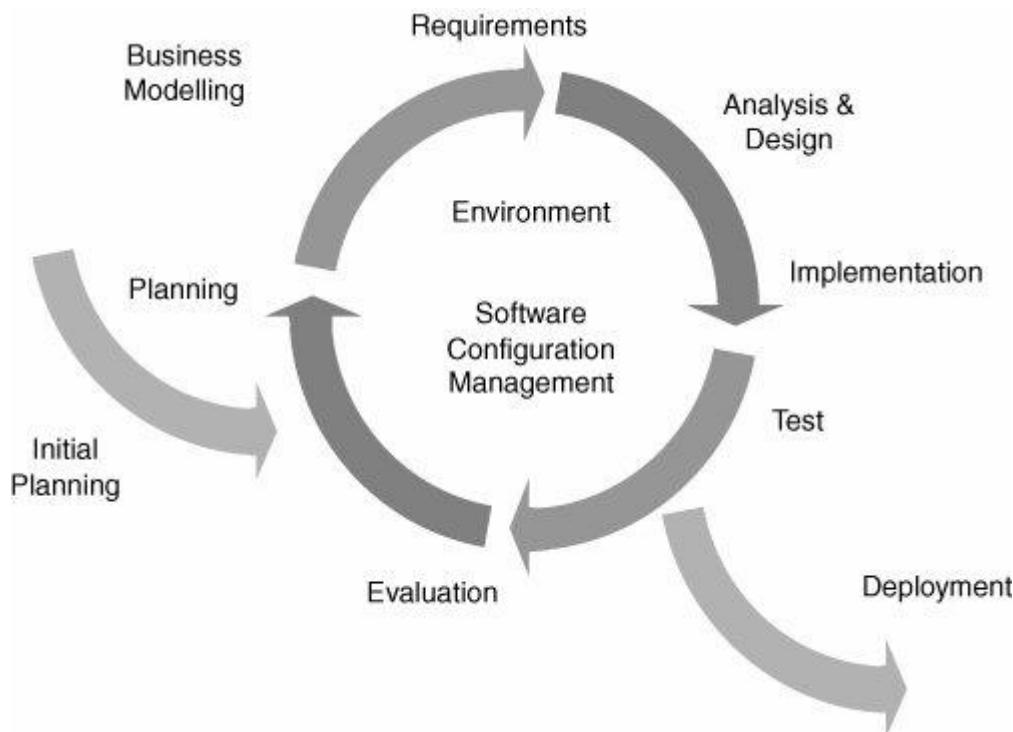
# Chapter 2 : Project Plan

## 2.1 Project Organization

### 2.1.1 Purpose

This chapter provides an overview of the project plan includes project organization and project management plan.

### 2.1.2 Software Process Model



*Figure 2-1: Iterative and Incremental Software Process Model*

This figure above describes the Iterative and Incremental Software Process Model that is used by MindDoctor.

The Iterative and Incremental Software Process Model is a method of software development that is modeled around a gradual increase in feature additions and a cyclical release and upgrade pattern. In incremental development, different parts of the system are developed at various times or rates are integrated based on their completion. In iterative development, MindDoctor team can revisit parts of the system in order to revise and improve them. Tester's feedback is consulted to modify the targets for successive deliverables.

### 2.1.3 Roles and Responsibilities

#### 2.1.4 Organization Structure

Role	Responsibility
Project Manager	Planning, developing schedules, coordinating communication, generally responsible for keeping the team's focus on the main goal.
Technical Leader	Responsible for choosing and deciding what technologies should be used, as well as for overseeing the work being done by other developers.
Quality Assurance Manager	Ensuring the product meets the certain standards of quality from requirements.
Test Leader	Responsible for test execution, including test set-up and test run, evaluation of test run and error recovery, defect logging and test results recording.
Developer	Involve coding the product and reviewing code of other developers.
Tester	Involve testing the product.
Business Analyst	Analyzes an organization or business domain and documents its business, processes, or systems.

Table 2-1: Project Structure

#### 2.1.5 Project Team Member

Team Member	Role
DatBV	Project Manager, Technical Leader, BA, Developer
TrungVT	Developer
CongHPA	Developer, Tester, BA
HuyTQ	Developer
VuNS	Developer, Tester Leader
TrinhDN	Developer

Table 2-2: Project Team Member

#### 2.1.6 Tools and Techniques

Programming languages	C#, JavaScript
Framework	.NET 4.5, Entity Framework 6.1.3, Bootstrap CSS, AngularJS v 1.0, jQuery 3.1.1
Software architecture	MVC 5
Version control	SVN 1.9.5
IDEs/Editors	Visual Studio 2015 Enterprise Edition
UML tools	Visio 2016

<b>Browser</b>	Google Chrome 60.0
<b>Web server</b>	IIS 10.0
<b>DBMS</b>	SQL Management Studio 2012 Enterprise Edition
<b>Deployment server</b>	Microsoft Azure
<b>Project management tool</b>	Trello.com
<b>Process model</b>	Iterative and incremental Software Process Model
<b>Development process</b>	Test-driven development

Table 2-3: Project Tools and Techniques

## 2.2 Project Management Plan

### 2.2.1 Tasks

Refers to “MindDoctor\_Project\_Management.mpp” file.

✓	★	▫ Development	72 days	Mon 15/05/17	Tue 22/08/17	11	
✓	★	▫ Phase 1	42 days	Mon 15/05/17	Tue 11/07/17		
⌚✓	⌚	▷ Team Weekly Meeting	40,25 days	Mon 15/05/17	Mon 10/07/17		
⌚✓	⌚	▷ Meeting with supervisor	40,25 days	Mon 15/05/17	Mon 10/07/17		
✓	⌚	▫ Planning	1 day	Mon 15/05/17	Mon 15/05/17	11	CongHPA,DatBV
✓	⌚	Planning Resource, Requirement, Trainning	1 day	Mon 15/05/17	Mon 15/05/17	11	DatBV
✓	⌚	Planning Test, Host	1 day	Mon 15/05/17	Mon 15/05/17		CongHPA
✓	⌚	Project Overall Design and design for phase 1	1 day	Tue 16/05/17	Wed 17/05/17	35	TrungVT
✓	★	▷ Create report 2	1 day	Tue 16/05/17	Tue 16/05/17	35	HuyTQ,TrungVT
✓	⌚	▷ Create report 3	1 day?	Tue 16/05/17	Tue 16/05/17	34	TrinhDN,VuNS
✓	★	▫ Create Test plan and setup testing environment	1 day	Thu 18/05/17	Thu 18/05/17	37	CongHPA,VuNS
✓	⌚	Create plan	1 day	Thu 18/05/17	Thu 18/05/17	37	CongHPA
✓	⌚	Prepare test enviroment	1 day?	Thu 18/05/17	Thu 18/05/17	37	VuNS
✓	⌚	▫ Create test case	11 days	Fri 19/05/17	Fri 02/06/17	44	CongHPA,VuNS
✓	⌚	GUI, Admin	11 days	Fri 19/05/17	Fri 02/06/17		CongHPA
✓	⌚	System, Guest, Customer	11 days	Fri 19/05/17	Fri 02/06/17		VuNS
✓	★	▷ Create report 4	2 days	Fri 19/05/17	Mon 22/05/17	37	TrinhDN[50%],TrungVT[50%]
✓	★	▷ Implementation	34 days	Fri 19/05/17	Wed 05/07/17	37	
✓	⌚	Review Test plan and update	1 day	Fri 09/06/17	Mon 12/06/17	47	CongHPA
✓	★	▷ Test phase 1	16,5 days	Mon 12/06/17	Tue 04/07/17	53	VuNS,CongHPA
✓	★	Summary and evaluation system for phase 1	1 day	Wed 05/07/17	Wed 05/07/17	54	DatBV
✓	⌚	Finish phase 1	0 days	Thu 06/07/17	Thu 06/07/17	116	
✓	★	▫ Phase 2	32 days	Mon 10/07/17	Tue 22/08/17		
⌚✓	★	▷ Team Weekly Meeting	31 days	Mon 10/07/17	Mon 21/08/17		

Figure 2-2: MindDoctor Project Management file

### 2.2.2 Meeting Minutes

All meeting minutes will be written following this template:

<b>Meeting/Project Name:</b>	MindDoctor		
<b>Date of Meeting:</b>	08/05/2017	<b>Time: (Type)</b>	2 hours (Face-to-face)
<b>Meeting Called by:</b>	DatBV	<b>Location:</b>	FPT University's 501R
<b>Note Taker:</b>	TrinhDN	<b>Time Keeper:</b>	VuNS
1. Meeting Objective			
- Choose names, ideas for project			

<b>Meeting/Project Name:</b>	<i>MindDoctor</i>		
<b>Date of Meeting:</b>	08/05/2017	<b>Time: (Type)</b>	2 hours (Face-to-face)
<b>Meeting Called by:</b>	DatBV	<b>Location:</b>	FPT University's 50IR
<b>Note Taker:</b>	TrinhDN	<b>Time Keeper:</b>	VuNS
2. Attendance			
<b>Name</b>	<b>Roles</b>	<b>E-mail</b>	<b>Phone</b>
Bùi Vinh Đạt	PM, Technical Leader, BA	<a href="mailto:datbvse03875@fpt.edu.vn">datbvse03875@fpt.edu.vn</a>	0166 360 5557
Vũ Trí Trung	Developer, Designer	<a href="mailto:trungvtse03880@fpt.edu.vn">trungvtse03880@fpt.edu.vn</a>	098 744 5154
Hà Phan Anh Công	Developer, Tester, BA	<a href="mailto:conghpase03806@fpt.edu.vn">conghpase03806@fpt.edu.vn</a>	098 937 6395
Tạ Quang Huy	Developer, Designer	<a href="mailto:huytqse03706@fpt.edu.vn">huytqse03706@fpt.edu.vn</a>	096 623 6917
Nguyễn Sơn Vũ	Developer, Tester Leader	<a href="mailto:vunsse03669@fpt.edu.vn">vunsse03669@fpt.edu.vn</a>	0166 323 4748
Đỗ Ngọc Trình	Developer, Designer	<a href="mailto:trinhdnse03817@fpt.edu.vn">trinhdnse03817@fpt.edu.vn</a>	098 974 0361
3. Content			
-			
4. Note			
-			

Table 2-4: Meeting Minutes Template

### 2.2.3 Coding Conventions

We strictly follow C# Convention Style Guide.

Please refer to the official website at <https://msdn.microsoft.com/en-us/library/ff926074.aspx>

### 2.2.4 Risk Management Plan

No	Description	Avoidance plan	Contingency plan	Status
R1	<b>Illness or absence of team members</b>	Member has to notice to the team about absence period and the plan of how to keep up with the work process.	Ensure that the absence of a member will not affect others and always have plans to deal with this problem.	Closed
R2	<b>Business problem</b>	Any ideas are welcome but members have to discuss with others and always focus on the reality and possibility.	Make sure the business logic of any ideas is carefully analyzed.	Closed
R3	<b>Change management overload</b>	A large number of change requests dramatically raises the complexity of the project and distracts key resources.	If there is a “must be changed” requirement, all team members must join the meeting to decide whether it should be implemented or not.	Closed
R4	<b>New technology</b>	Choosing technology based on member’s qualification. All team members must nurture by self-study.	When someone chooses a new technology, he/she has to explain to all team members about the decision.	Closed
R5	<b>Meeting issues and missing deadline</b>	When having issues or missing deadline, member has to raise to technical leader or pm. Technical leader and pm will find the way to solve issues.	Pm notice to supervisor to get support. In case missing deadline pm will allocate others to support or make member work over time.	Closed

Table 2-5: Risk Management

### 2.2.5 Communication Plan

*Weekly meeting schedule:* We use Iterative and Incremental Process Model, then we divide the system into functions, each function is divided into a bunch of small tasks. Each task is recorded to “Project Management Plan” file then estimated depending on difficulty and the amount of work by the whole team, after that the Team Leader will assign the task to team members and depending on difficulty, the Technical Leader will assign deadlines for each task. We will have a meeting every Monday to inform to all team about what each member finished last week, the status (fast, on time or slow), the issues met and how to solve them. If any member raises any issue, the whole team will help to find out a solution together. After that, the team will define detailed stories for next week tasks and estimate how long it takes to finish them.

*Daily discussing schedule:* All member will have a stand-up meeting to inform to others: “What did I do yesterday?”, “What will I do today?”, “Are there any impediments in

my ways?” and “Do you have any issue?”. By focusing on what each person accomplished yesterday and will accomplish today, the team gains an excellent understanding of what work has been done and what work remains.

*Unscheduled meeting:* If someone has an important problem that he wants to solve immediately, we will have a meeting for discussion, usually via some online channel: Slack, Skype, or Phone.

*Communication channel:* Our main communication channels are Slack and Messenger. On the other hand, we used face-to-face meeting, Email, Messenger and comment on “Project Management Plan” file. However, we sometimes make a phone call or instant message if someone has a problem.



Figure 2-3: Communication via Slack

# **Chapter 3 : Software Requirement Specification**

## **3.1 Purpose**

This chapter outlines functional and non-functional requirements of our website. It also provides some format constraints in common requirements and project success criteria. All members will work (design, code, test) based on the information provided in this chapter.

## **3.2 Functional Requirements**

### **3.2.1 Use Case Diagram**

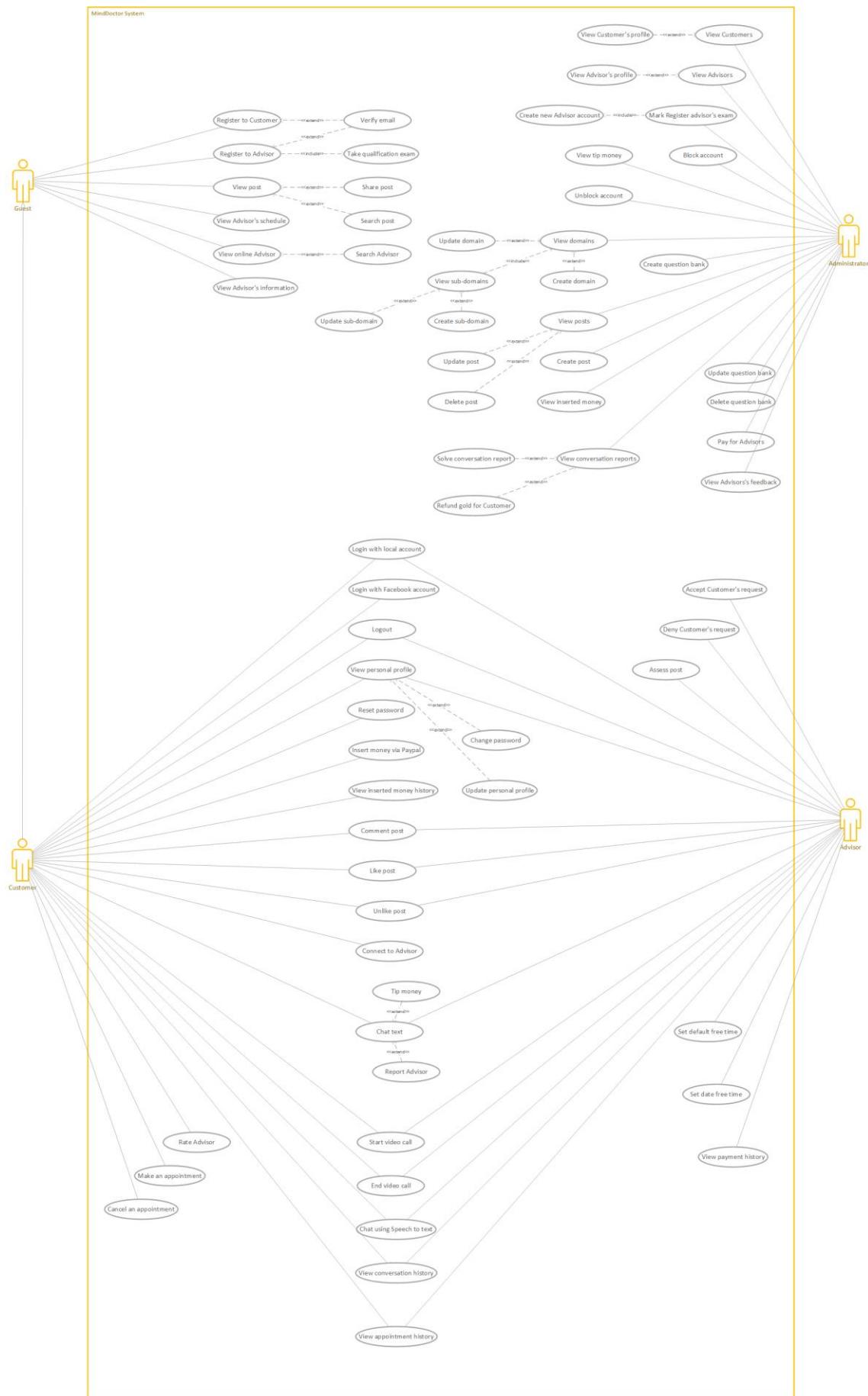


Figure 3.1 : Use case diagram of MindDoctor

### 3.2.2 Business Rules

No	Description
<b>B1</b>	The email address must not be empty.
<b>B2</b>	The email must be valid and real email.
<b>B3</b>	The email must be not same others account's email.
<b>B4</b>	Uploaded image must be less or equal than 4MB.
<b>B5</b>	The password must not be empty.
<b>B7</b>	The password length must be great or equal than 8 characters.
<b>B8</b>	The phone number must be real.
<b>B9</b>	Post's title and content must be not empty.
<b>B10</b>	The confirm password must be matched with the new password.
<b>B11</b>	The advisor is able to: manage profile, accept request from customer, chat with customer, add, like, comment, share, search post, set free time for schedule...
<b>B12</b>	The customer is able to: manage profile, connect to advisor, add, like, comment share search post, booking schedule with advisor, insert money, tip money, rate advisor, report advisor...
<b>B13</b>	The administrator is able to: create advisor account, manage advisor and customer information, manage domain, subdomain, block, unblock account, manage post, pay money for advisor....
<b>B14</b>	The guest is not able to: connect to advisor, chat, like, comment post ...
<b>B15</b>	Customer cannot connect with advisor being busy.
<b>B16</b>	Customer cannot connect with advisor offline.
<b>B17</b>	Customer cannot tip amount of money that great than remained money in account.
<b>B18</b>	Customer only connect or book an appointment with Advisor when remained money in account is equal or larger than GPC of Advisor.
<b>B19</b>	Customer cannot make appointment with advisor in busy time of that advisor.
<b>B20</b>	Guest must fill all information when register to become advisor.
<b>B21</b>	Advisor cannot cancel free time that customer book schedule.
<b>B22</b>	Customer can cancel appointment before one day only.
<b>B23</b>	User can only update his/her profile.
<b>B24</b>	Admin can not update profile of everyone.
<b>B25</b>	Customer will be gifted 10% money if inserted equal or larger than \$10.
<b>B26</b>	Advisor do not have the right to cancel appointment.
<b>B27</b>	If advisor does not participate in conversation which booked before, advisor will be minus 0.1 rating. Customer will receive all deposited money.
<b>B28</b>	If customer does not participate in conversation which booked before, advisor and customer will be received half of deposited money.
<b>B29</b>	Inserted money must be integer (\$) and larger or equal than \$1.

<b>B30</b>	Appointment will be canceled if not have any response from Customer or Advisor after 2 minutes.
------------	---

*Table 3-1: Business rules*

### 3.2.3 Use Cases

<b>Actor</b>	<b>Description</b>
Guest	User does not have account and accesses into the websites.
Customer	User has account as a customer and accesses into MindDoctor Website.
Advisor	User has account as an advisor and accesses into MindDoctor Website.
Administrator	A person who has the highest permission level and has responsibility for managing the website.

*Table 3-2: Actor description*

<b>ID</b>	<b>Primary Actor</b>	<b>Name</b>	<b>Secondary Actor</b>	<b>Assign To</b>
UC-1	Guest	Register to become Customer		DatBV
UC-2	Guest	Verify email address	Customer Advisor	DatBV
UC-3	Guest	Register to become Advisor	Customer	DatBV
UC-4	Guest	Take qualification exam	Customer	DatBV
UC-5	Guest	View post	Customer Advisor Administrator	DatBV
UC-6	Guest	Share post on Facebook	Customer Advisor Administrator	DatBV
UC-7	Guest	Search post	Customer Advisor Administrator	VuNS
UC-8	Guest	View Advisors's schedule	Customer Advisor Administrator	DatBV
UC-9	Guest	View online Advisors	Customer Advisor Administrator	TrungVT
UC-10	Guest	View online Advisors by searching Advisor's information	Customer Advisor Administrator	TrungVT

UC-11	Guest	View Advisor's information	Customer Advisor Administrator	TrungVT
UC-12	Customer	Login with local account	Advisor Administrator	DatBV
UC-13	Customer	Login with Facebook account		DatBV
UC-14	Customer	Logout	Advisor Administrator	DatBV
UC-15	Customer	Reset password	Advisor Administrator	DatBV
UC-16	Customer	Change password	Advisor Administrator	DatBV
UC-17	Customer	View personal profile	Advisor	TrungVT
UC-18	Customer	Update personal profile	Advisor	TrungVT
UC-19	Customer	Insert money via Paypal		TrungVT
UC-20	Customer	View inserted money history		DatBV
UC-21	Customer	Comment post	Advisor Administrator	DatBV
UC-22	Customer	Like post	Advisor Administrator	DatBV
UC-23	Customer	Unlike post	Advisor Administrator	DatBV
UC-24	Customer	Connect to Advisor		TrungVT
UC-25	Customer	Chat text	Advisor	TrungVT
UC-26	Customer	Start video call	Advisor	TrinhDN
UC-27	Customer	End video call	Advisor	TrinhDN
UC-28	Customer	Chat using Speech to Text		TrinhDN
UC-29	Customer	Tip money for Advisor		TrungVT
UC-30	Customer	Rate Advisor after finishing conversation		TrungVT
UC-31	Customer	Report Advisor while chatting		TrungVT
UC-32	Customer	View conversation history		TrungVT
UC-33	Customer	Make an appointment with Advisor		DatBV
UC-34	Customer	Cancel an appointment with Advisor		DatBV
UC-35	Customer	View appointment history	Advisor	TrungVT

UC-36	Advisor	Accept Customer's chat request		TrungVT
UC-37	Advisor	Deny Customer's chat request		TrungVT
UC-38	Advisor	Set default free time on week		DatBV
UC-39	Advisor	Set free time by date		DatBV
UC-40	Advisor	View payment history		DatBV
UC-41	Advisor	Assess Post		DatBV
UC-42	Administrator	View Customers		VuNS
UC-43	Administrator	View Customer's profile		VuNS
UC-44	Administrator	View Advisors		VuNS
UC-45	Administrator	View Advisor's profile		VuNS
UC-46	Administrator	Create new Advisor account		HuyTQ
UC-47	Administrator	Mark exam for Register Advisor		HuyTQ
UC-48	Administrator	Block account		HuyTQ
UC-49	Administrator	Unblock account		HuyTQ
UC-50	Administrator	View domain		CongHPA
UC-51	Administrator	Update domain		CongHPA
UC-52	Administrator	Create new domain		CongHPA
UC-53	Administrator	View sub-domain		CongHPA
UC-54	Administrator	Update sub-domain		CongHPA
UC-55	Administrator	Create new sub-domain		CongHPA
UC-56	Administrator	Create new post		TrinhDN
UC-57	Administrator	Edit post		TrinhDN
UC-58	Administrator	Delete post		TrinhDN
UC-59	Administrator	View conversation report		HuyTQ
UC-60	Administrator	Solve conversation report		HuyTQ
UC-61	Administrator	Refund gold for customer		HuyTQ
UC-62	Administrator	View inserted money		HuyTQ
UC-63	Administrator	View tip money		HuyTQ
UC-64	Administrator	Create questions bank		HuyTQ
UC-65	Administrator	Update questions bank		HuyTQ
UC-66	Administrator	Delete questions bank		HuyTQ
UC-67	Administrator	View Advisor's feedback		HuyTQ
UC-68	Administrator	Pay money for Advisor (assume via Bank)		DatBV

UC-69		Appointment helper		DatBV
-------	--	--------------------	--	-------

Table 3-3: Use Case list

### 3.2.3.1 Guest

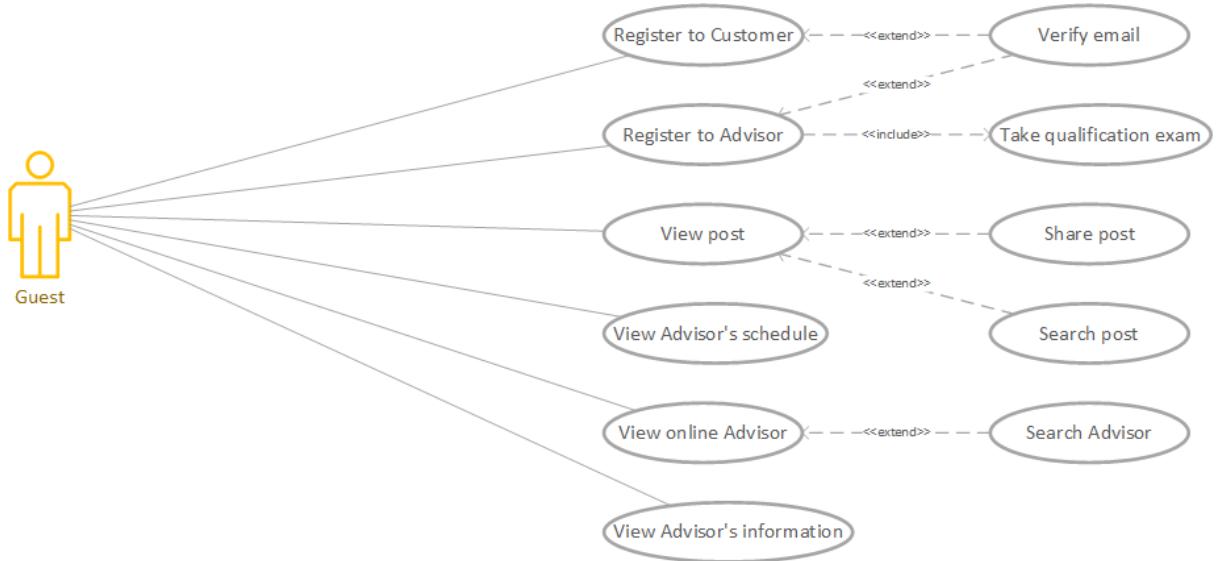


Figure 3.2 :Use case diagram of Guest actor

### **3.2.3.1.1 Register to become Customer**

*Use Case Specification*

UC ID and Name:	<b>UC-1.0: Register to become Customer</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Guest who wants to create an Customer's account.		
Preconditions:	<b>PRE-1.1:</b> Guest who has not an Customer's account.		
Post conditions:	<b>POST-1.1:</b> Registered user is logged into system successfully. <b>POST-1.2:</b> Browser displays in page which Guest clicked on register. <b>POST-1.3:</b> System stores Registered user access token to cookie.		
Normal Flow:	1. Guest visits website. 2. Browser displays home page. 3. Guest user clicks on button “Đăng nhập” on the navigation bar. 4. Page shows “Đăng nhập” dialog. 5. Guest clicks “Đăng ký” on the bottom of dialog. 6. Page shows “Đăng ký” dialog. 7. Guest enters the input: “Email”, “Mật khẩu”, “Xác nhận mật khẩu”. 8. Guest clicks on button “Đăng ký”.		
Alternative Flows:	N/A		
Exceptions:	1. Input wrong email. 2. Input invalid password.		
Priority:	Low		
Frequency of Use:	High		
Business Rules:	<b>B1, B2, B3, B5, B7, B10</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.2 Verify email address

*Use Case Specification*

UC ID and Name:	<b>UC-2.0: Verify email address</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor</b>
Trigger:	N/A		
Description:	Send verify email to Registered user after clicks on button “Đăng ký”.		
Preconditions:	<b>PRE-2.1:</b> Guest who registered an Customer’s account successfully. <b>PRE-2.2:</b> Guest checks email and click on verify link.		
Post conditions:	<b>POST-2.1:</b> Guest’s email is verified successfully.		
Normal Flow:	1. Guest opens mail and clicks on “Xác thực email” link. 2. Browser displays “Xác thực email thành công” page.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.3 *Register to become Advisor*

#### *Use Case Specification*

UC ID and Name:	<b>UC-3.0: Register to become Advisor</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer</b>
Trigger:	User clicks on button “Đăng ký”.		
Description:	User who wants to create an Advisor’s account.		
Preconditions:	N/A		
Post conditions:	<p><b>POST-3.1:</b> Browser displays “TakeExam” page.</p> <p><b>POST-3.2:</b> User must take qualification exam.</p> <p><b>POST-3.3:</b> System stores Exam access token to cookie.</p>		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits website.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Hỗ trợ” on the navigation bar.</li> <li>4. Browser shows drop down.</li> <li>5. User clicks on “Đăng ký Tư vấn viên”.</li> <li>6. Browser displays “Đăng ký Tư vấn viên” page.</li> <li>7. User enters all the inputs and selects.</li> <li>8. User clicks on button “Đăng ký”.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:	<ol style="list-style-type: none"> <li>1. Not input all field.</li> <li>2. Input wrong email.</li> <li>3. Input wrong phone number.</li> </ol>		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	<b>B2, B3, B8, B20</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.4 *Take qualification exam*

*Use Case Specification*

UC ID and Name:	<b>UC-4.0: Take qualification exam</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 22<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Guest</b>
Trigger:	User clicks on button “Xác nhận” in “Đăng ký Tư vấn viên” page.		
Description:	User who wants to create an Advisor's account.		
Preconditions:	<b>PRE-4.1:</b> User who clicks on button “Đăng ký” on “Đăng ký Tư vấn viên” page successfully.		
Post conditions:	<b>POST-4.1:</b> System saves User's exam to database and send email to user.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Browser shows Confirm dialog.</li> <li>2. User clicks on button “Xác nhận”.</li> <li>3. Browser displays “Bài kiểm tra năng lực” page.</li> <li>4. User does this exam on the preset time.</li> <li>5. User clicks on “Hoàn thành” button or system auto “Hoàn thành” if the time is finished.</li> <li>6. System saves User's exam to database and send email to user.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.5 *View post*

*Use Case Specification*

UC ID and Name:	<b>UC-5.0: View post</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 22<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	User can view list of posts.		
Preconditions:	<b>PRE-5.1:</b> User of MindDoctor system.		
Post conditions:	<b>POST-5.1:</b> System shows posts.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tin tức” in the navigation bar.</li> <li>4. Browser shows drop down.</li> <li>5. User chooses category in the drop down of “Tin tức”.</li> <li>6. Browser displays all post of this category.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.6 Share post on Facebook

#### Use Case Specification

UC ID and Name:	<b>UC-6.0: Share post on Facebook</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 23<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	User clicks on button “Chia sẻ” at the comments area in the bottom of post.		
Description:	User can share a post's link on Facebook.		
Preconditions:	<b>PRE-6.1:</b> User of MindDoctor. <b>PRE-6.2:</b> User is on post detail page.		
Post conditions:	N/A		
Normal Flow:	1. User clicks on “Chia sẻ” button at the comments area in the bottom of post. 2. Facebook's popup will be shown, user clicks on “Share”.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Low		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.7 *Search post*

#### *Use Case Specification*

UC ID and Name:	<b>UC-7.0: Search post</b>		
Created by:	<b>VuNS</b>	Created date:	<b>Jun 14<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	User who wants to search a post by approximate title, description, content, domain or subdomain of this post.		
Preconditions:	N/A		
Post conditions:	<p><b>POST-7.1:</b> List all candidate posts from search result into search page.</p> <p><b>POST-7.2:</b> Candidate posts have been sorted by decreasing accuracy of input text from search bar and text in post's content.</p>		
Normal Flow:	<ol style="list-style-type: none"> <li>9. User visits website.</li> <li>10. Browser displays home page.</li> <li>11. User hovers on “Tin tức” on navigation bar.</li> <li>12. Browser shows drop down.</li> <li>13. User clicks on any sub-domain like.</li> <li>14. Redirect to Posts page.</li> <li>15. User clicks on search button on navigation bar.</li> <li>16. User fills text into search bar.</li> <li>17. User presses on enter button from keyboard.</li> <li>18. Browser redirects to search result.</li> </ol>		
Alternative Flows:	<p>Search all post:</p> <ol style="list-style-type: none"> <li>1. From search page, user fills text into search bar.</li> <li>2. User clicks on search button or presses on enter button from keyboard.</li> <li>3. All search result about post will be showed.</li> </ol> <p>Search post in domain:</p> <ol style="list-style-type: none"> <li>1. From search page, user selects domain in “Lĩnh vực” drop down list.</li> <li>2. User fills text into search bar.</li> <li>3. User presses on enter button from keyboard.</li> <li>4. Browser redirects to search result.</li> </ol>		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.8 View Advisors's schedule

*Use Case Specification*

UC ID and Name:	<b>UC-8.0: View Advisor's schedule</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 23<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	User can view Advisor's free time schedule.		
Preconditions:	N/A		
Post conditions:	<b>POST-8.1:</b> System displays Advisor's schedule.  1. User visits home page. 2. Browser displays home page. 3. User clicks on “Đặt lịch hẹn” on the navigation bar. 4. Browser displays “Đặt lịch hẹn” page. 5. User can enter conditions to view data which comfortability. 6. User clicks on “Đặt lịch” button. 7. System displays Advisors's matched conditions.		
Normal Flow:	1. User visits home page. 2. Browser displays home page. 3. User scrolls down and click on button “Đặt lịch hẹn”. 4. Browser display “Đặt lịch hẹn” page. 5. User can enter conditions to view data which comfortability. 6. User clicks on “Đặt lịch” button. System displays Advisor's matched conditions.		
Alternative Flows:	1. User visits home page. 2. Browser displays home page. 3. User scrolls down and click on button “Đặt lịch hẹn”. 4. Browser display “Đặt lịch hẹn” page. 5. User can enter conditions to view data which comfortability. 6. User clicks on “Đặt lịch” button. System displays Advisor's matched conditions.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running. Firebase is stable running.		

### 3.2.3.1.9 *View online Advisors*

#### *Use Case Specification*

UC ID and Name:	<b>UC-9.0: View online Advisors</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	User can view online advisor who is online.		
Preconditions:	<b>PRE-9.1:</b> Advisor has to log into the system.		
Post conditions:	<b>POST-9.1:</b> Browser displays online advisor.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tư vấn” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Redirect to Advising domain page.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.1.10      View online Advisors by searching Advisor's information**

*Use Case Specification*

UC ID and Name:	<b>UC-10.0: View online Advisors by searching Advisor's information</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 22<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	User can search online Advisors by input field or select domain.		
Preconditions:	<b>PRE-10.1:</b> Go to domain detail page. <b>PRE-10.2:</b> Advisors have to log into the system.		
Post conditions:	<b>POST-10.1:</b> Advisors will appear with information you search or subdomain you choose.		
Normal Flow:	1. User visits home page. 2. User hovers on “Tu vấn” on the navigation bar. 3. Page shows drop down. 4. User clicks on domain which want to be advised. 5. Redirect to advising domain page. 6. Type text to input field or choose domain in drop down list.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.1.11 View Advisor's information

Use Case Specification

UC ID and Name:	<b>UC-11.0: View Advisor's information</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Guest</b>	Secondary Actor:	<b>Customer Advisor Administrator</b>
Trigger:	N/A		
Description:	View information of advisor, such as: name, rate, workplace, introduce about advisor, fee each advice conversation, degree, domain and subdomain which advisor can consult.		
Preconditions:	<b>PRE-11.1:</b> The system must be had at least one advisor online.		
Post conditions:	<b>POST-11.1:</b> A popup will be appeared with information of advisor.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tu vấn” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on domain which wants to be advised.</li> <li>6. User clicks on advisor's avatar want to view information.</li> <li>7. A dialog will appear with advisor's information.</li> </ol>		
Alternative Flows:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Hỗ trợ” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on “Tất cả tư vấn viên” on the drop down.</li> <li>6. Redirect to “Tất cả tư vấn viên” page.</li> <li>7. User clicks on advisor's avatar want to view information.</li> <li>8. A dialog will appear with advisor's information.</li> </ol>		
Exceptions:			
Priority:	Medium		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2 Customer

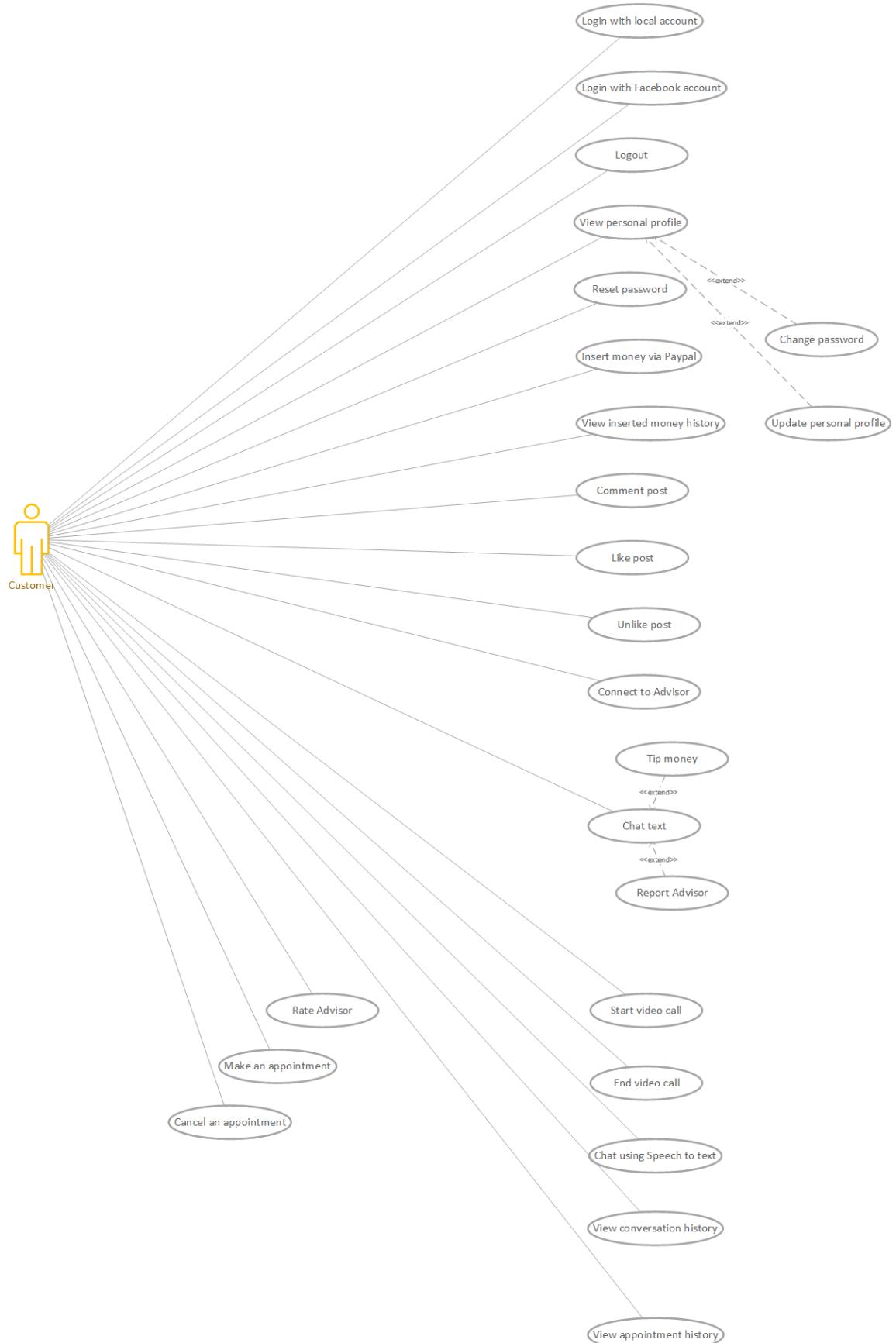


Figure 3.3 : Use case diagram of Registered user actor

### **3.2.3.2.1      Login with local account**

*Use Case Specification*

UC ID and Name:	<b>UC-12.0: Login with local account</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 24<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A.		
Description:	Internal login with MindDoctor account.		
Preconditions:	<b>PRE-12.1:</b> Registered user who has MindDoctor account.		
Post conditions:	<b>POST-12.1:</b> Registered user is logged into system successfully. <b>POST-12.2:</b> Browser displays previous page which user login. <b>POST-12.3:</b> System stores Registered user access token to cookie.		
Normal Flow:	1. Guest visits home page. 2. Browser displays home page. 3. Guest clicks on button “Đăng nhập” on the navigation bar. 4. Page shows “Đăng nhập” dialog. 5. Guest enters account which has been registered. 6. Guest clicks on button “Đăng nhập”. 7. Register user logs into system successfully. 8. Browser displays page which guest clicks on button “Đăng nhập”.		
Alternative Flows:	1. Guest enters link “Đăng nhập” page on the navigation bar. 2. Browser displays “Đăng nhập” page. 3. Guest enters account which has been registered. 4. Guest clicks on button “Đăng nhập”. 5. Register user logs into system successfully. 6. Browser displays page which guest clicks on button “Đăng nhập”.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	High		
Business Rules:	<b>B1, B2, B5, B7</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.2 Login with Facebook account**

*Use Case Specification*

UC ID and Name:	<b>UC-13.0: Login with Facebook account</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 24<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	External login with Facebook account.		
Preconditions:	<b>PRE-13.1:</b> User has Facebook account.		
Post conditions:	<b>POST-13.1:</b> Registered user is logged into system successfully. <b>POST-13.2:</b> Browser displays previous page which user login. <b>POST-13.3:</b> System stores Registered user access token to cookie.		
Normal Flow:	1. Guest visits home page. 2. Browser displays home page. 3. Guest clicks on button “Đăng nhập” on the navigation bar. 4. Page shows “Đăng nhập” dialog. 5. Guest clicks on button “Đăng nhập bằng Facebook”. 6. Register user log into system successfully. 7. Browser displays page which guest clicks on button “Đăng nhập”.		
Alternative Flows:	1. Guest enters link “Đăng nhập” page on the url bar. 2. Browser displays “Đăng nhập” page. 3. Guest clicks on button “Đăng nhập bằng Facebook”. 4. Register users log into system successfully. 5. Browser displays page which guest clicks on button “Đăng nhập”.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.3      ***Logout***

*Use Case Specification*

UC ID and Name:	<b>UC-14.0: Logout</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 25<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A		
Description:	Log out the system.		
Preconditions:	<b>PRE-14.1:</b> User of MindDoctor who is logged into the system.		
Post conditions:	<b>POST-14.1:</b> Registered user log out of the system successfully. <b>POST-14.2:</b> Browser displays home page. <b>POST-14.3:</b> System deletes Registered user access token to cookie.		
Normal Flow:	1. User clicks on button “Đăng xuất” on the navigation bar. 2. User logs out successfully.		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.4 Reset password

*Use Case Specification*

UC ID and Name:	<b>UC-15.0: Reset password</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 26<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A		
Description:	Change Registered user password.		
Preconditions:	<b>PRE-15.1:</b> User of MindDoctor who verified email successfully.		
Post conditions:	<b>POST-15.1:</b> Registered user changes password successfully.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User clicks on button “Đăng nhập” on the navigation bar.</li> <li>4. Page shows “Đăng nhập” dialog.</li> <li>5. User clicks on “Quên mật khẩu”.</li> <li>6. Page shows “Quên mật khẩu” dialog.</li> <li>7. User enters “Email” field.</li> <li>8. User clicks on “Lấy lại mật khẩu” button.</li> <li>9. Server sends a mail to confirm that user wants to reset a password.</li> <li>10. User checks email and click to reset password link.</li> <li>11. User enters “Email”, “Mật khẩu”, “Quên mật khẩu” field.</li> <li>12. User clicks on “Lấy lại mật khẩu” button.</li> <li>13. Browser shows successful dialog.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:	<ol style="list-style-type: none"> <li>1. Email has not verified yet.</li> </ol>		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	<b>B1, B2</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.5 *Change password*

*Use Case Specification*

UC ID and Name:	<b>UC-16.0: Change password</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 29<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A		
Description:	Change Registered user's password.		
Preconditions:	<b>PRE-16.1:</b> Registered user is logged into the system.		
Post conditions:	<b>POST-16.1:</b> Registered user changes password successfully.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User hovers on display name on the navigation bar.</li> <li>2. Page shows dropdown.</li> <li>3. User clicks on “Thay đổi mật khẩu”.</li> <li>4. Browser displays “Thay đổi mật khẩu” page.</li> <li>5. User enters input: “Mật khẩu hiện tại”, “Mật khẩu mới”, “Nhập lại mật khẩu”.</li> <li>6. User clicks on button “Cập nhật”.</li> <li>7. Page shows successful dialog.</li> <li>8. User changes password successfully.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:	<ol style="list-style-type: none"> <li>1. Current password is not true.</li> </ol>		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	<b>B5, B7, B10</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.6 View personal profile**

*Use Case Specification*

UC ID and Name:	<b>UC-17.0: View personal profile</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 01<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	View information of himself/herself, such as: full name, date of birth, gender, email, province city, country, avatar.		
Preconditions:	<b>PRE-17.1:</b> User has to log into the system.		
Post conditions:	<b>POST-17.1:</b> The Profile page is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on “Hồ sơ cá nhân”.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	<b>B11, B12</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.7      Update personal profile**

*Use Case Specification*

UC ID and Name:	<b>UC-18.0: Update personal profile</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 25<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	Update information of himself/herself, such as: full name, date of birth, gender, province city, country, avatar.		
Preconditions:	<b>PRE-18.1:</b> User have to logged into the system.		
Post conditions:	<b>POST-18.1.</b> The Profile page is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks to “Hồ sơ cá nhân”.</li> <li>6. Type text into input field which you want to change.</li> <li>7. User clicks to “Cập nhật” button to save new information.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	<b>B4, B8, B11, B12, B23</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.8 Insert money via Paypal

Use Case Specification

UC ID and Name:	<b>UC-19.0: Insert money via Paypal</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 25<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Customer can insert money via Paypal.		
Preconditions:	<b>PRE-18.1:</b> User has to logged into the system.		
Post conditions:	<b>POST-18.1.</b> User's money is increase.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Hỗ trợ” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on “Nạp tiền” button.</li> <li>6. Page shows “Nạp tiền” dialog.</li> <li>7. User enters amount of money wants to insert.</li> <li>8. User clicks on button “Nạp tiền”.</li> <li>9. Page redirects to Paypal.</li> <li>10. User checks information and click Accept to accept order.</li> <li>11. Page redirects to “Xác nhận thanh toán”.</li> <li>12. User clicks on “Xác nhận thanh toán” to finish this transaction.</li> <li>13. User's gold will be added corresponding with inserted gold.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	Medium		
Business Rules:	<b>B29</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.9 View inserted money history

*Use Case Specification*

UC ID and Name:	<b>UC-20.0: View inserted money history</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 30<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Customer can view inserted money history.		
Preconditions:	<b>PRE-20.1:</b> Registered user is logged into the system.		
Post conditions:	<b>POST-20.1:</b> System display inserted money history table.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on “Lịch sử nạp tiền”.</li> <li>6. Redirect to “Lịch sử nạp tiền” page.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.10 Comment post**

*Use Case Specification*

UC ID and Name:	<b>UC-21.0: Comment post</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 26<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A		
Description:	Registered user can comment on the post.		
Preconditions:	<b>PRE-21.1:</b> Registered user is logged into the system. <b>PRE-21.2:</b> Registered user is in post view.		
Post conditions:	<b>POST-21.1:</b> Registered user comment on post successfully.		
Normal Flow:	1. User scrolls down to comments area at bottom of page. 2. Enter content into text field. 3. Press “Enter” on the keyboard. 4. Browser displays user’s comment.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.11    *Like post***

*User Case Specification*

UC ID and Name:	<b>UC-22.0: Like post</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 29<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A		
Description:	Registered user can like the post which user interested in.		
Preconditions:	<b>PRE-22.1:</b> Registered user is logged into the system. <b>PRE-22.2:</b> Registered user is in post view.		
Post conditions:	<b>POST-22.1:</b> Registered user like the post successfully.		
Normal Flow:	1. User scrolls down to comments area at bottom of page. 2. Click on button “Thích”.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.12 Unlike post

User Case Specification

UC ID and Name:	<b>UC-23.0: Unlike post</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 30<sup>th</sup>, 2017</b>
Primary Actor:	Customer	Secondary Actor:	<b>Advisor Administrator</b>
Trigger:	N/A.		
Description:	Registered user can unlike the post which user has not interested in yet.		
Preconditions:	<b>PRE-23.1:</b> Registered user is logged into the system. <b>PRE-23.2:</b> Registered user is in liked post view.		
Post conditions:	<b>POST-23.1.</b> Registered user like the post successfully.		
Normal Flow:	1. User scrolls down to comments area at bottom of page. 2. Click on button “Bỏ thích”.		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.13 Connect to Advisor

#### User Case Specification

UC ID and Name:	<b>UC-24.0: Connect to Advisor</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 24<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Connect to advisor who you want to be advised.		
Preconditions:	<b>PRE-24.1:</b> Customer has to log into the system.		
Post conditions:	<b>POST-24.1.</b> The connecting dialog is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tư vấn” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Click to “Tư vấn ngay” button of advisor wants to be advised.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B12, B15, B16</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.14 Chat text

#### User Case Specification

UC ID and Name:	<b>UC-25.0: Chat text</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 30<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	Send text message to each other.		
Preconditions:	<p><b>PRE-25.1:</b> User has to log into the system.</p> <p><b>PRE-25.2:</b> Customer has to connect successful with one advisor and the advisor accepts to start a conversation.</p>		
Post conditions:	<p><b>POST-25.1.</b> The text message which is sent will appear in the right hand of text area.</p>		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tu vân” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Click to “Tu vân ngay” button of advisor wants to be advised.</li> <li>7. When advisor starts a conversation, the website will redirect to Page conversation.</li> <li>8. Type text to input field then press “Enter” keyboard to send.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B11, B12</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.15 Start Video Call

User Case Specification

UC ID and Name:	<b>UC-26.0: Create Video Call</b>		
Created by:	<b>TrinhDN</b>	Date Created:	<b>May 25<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A.		
Description:	Customer creates a video conference with Advisor.		
Preconditions:	<b>PRE-26.1:</b> User has to log into the system. <b>PRE-26.2:</b> Customer has to connect successful with one advisor and the advisor accepts to start a conversation.		
Post conditions:	<b>POST-26.1:</b> Video Conference between Advisor and Customer is created.		
Normal Flow:	<ol style="list-style-type: none"> <li><b>1. Advisor accepts the call</b> <ol style="list-style-type: none"> <li>1.1 Click on “Bắt đầu gọi video” button.</li> <li>1.2 Page displays “Cuộc gọi đến” dialog with Advisor.</li> <li>1.3 Advisor accepts the call.</li> <li>1.4 Page displays “Video Call” area, hide information area and resize chat area.</li> </ol> </li>   <li><b>2. Advisor cancels the call</b> <ol style="list-style-type: none"> <li>2.1 Click on “Bắt đầu gọi video” button.</li> <li>2.2 Page displays “Cuộc gọi đến” dialog with Advisor.</li> <li>2.3 Advisor does not accept the call.</li> <li>2.4 Page displays “Cuộc gọi bị huỷ” dialog.</li> </ol> </li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B15, B16</b>		
Other Information:	N/A.		
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.16 End Video Call

*Use Case Specification*

UC ID and Name:	<b>UC-27.0: End Video Call</b>		
Created by:	<b>TrinhDN</b>	Date Created:	<b>May 31<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A.		
Description:	Customer ends the video conference.		
Preconditions:	<b>PRE-27.1:</b> User has to log into the system. <b>PRE-27.2:</b> Customer has to connect successful with one advisor and the advisor accepts to start a conversation. <b>PRE-27.3:</b> Customer is in video conference with Advisor.		
Post conditions:	<b>POST-27.1:</b> Video Conference between Advisor and Customer is end.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Click on “Huỷ cuộc gọi” button.</li> <li>2. Page ends the current video conference of customer, hide video call area, displays information area and resize chat box.</li> <li>3. Page displays “Cuộc gọi bị huỷ” dialog.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B15, B16</b>		
Other Information:	N/A.		
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.17 Chat using Speech to Text

*Use Case Specification*

UC ID and Name:	<b>UC-28.0: Chat using Speech to Text</b>		
Created by:	<b>TrinhDN</b>	Date Created:	<b>May 31<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	Customer can use voice to text into conversation.		
Preconditions:	<b>PRE-28.1:</b> User has to log into the system. <b>PRE-28.2:</b> Customer has to connect successful with one advisor and the advisor accepts to start a conversation.		
Post conditions:	<b>POST-28.1:</b> Page show text content from customer's voice.		
Normal Flow:	1. Click on “Speak to text” button. 2. Customer speaks. 3. Page transforms customer’s voice to text and show in chat box.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	<b>B15, B16</b>		
Other Information:	N/A		
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.2.18 Tip money for Advisor

*Use Case Specification*

UC ID and Name:	<b>UC-29.0: Tip money for Advisor</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>May 26<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	User can tip money for advisor when they want.		
Preconditions:	<b>PRE-29.1:</b> User must be in conversation with advisor.		
Post conditions:	<b>POST-29.1.</b> The success alert is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tư vấn” on the navigation bar.</li> <li>4. Page shows drop down.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Click on “Tư vấn ngay” button of advisor wants to be advised.</li> <li>7. When advisor starts a conversation, the website will redirect to Page conversation.</li> <li>8. Click on setting icon at the top right.</li> <li>9. Page shows drop down.</li> <li>10. Click on “Tip tiền cho Tư vấn viên”.</li> <li>11. User types number of money wants to tip to input field.</li> <li>12. Click on “Tip” button.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:	<ol style="list-style-type: none"> <li>1. Customer tips much more than the money they have.</li> </ol>		
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B12, B17</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.19      Rate Advisor after finishing conversation**

*Use Case Specification*

UC ID and Name:	<b>UC-30.0: Rate Advisor after finishing conversation</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 02<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	User can rate advisor and send feedback to improve the quality of the system and advisor.		
Preconditions:	<b>PRE-30.1:</b> User must be in conversation with advisor.		
Post conditions:	<b>POST-30.1.</b> The success alert is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tu vấn” on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Click on “Tu vấn ngay” button of advisor you want to be advised.</li> <li>7. When advisor starts a conversation, the website will redirect to Page conversation.</li> <li>8. Click on the exit icon in the top of the right hand.</li> <li>9. Click on “Đồng ý” when alert is appeared.</li> <li>10. Choose star in “Đánh giá” and user can type feedback to input field if they want.</li> <li>11. Click on “Gửi” button.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:	<ol style="list-style-type: none"> <li>1. User did not rate star before click to “Gửi” button.</li> </ol>		
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B12</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.20 Report Advisor while chatting**

*Use Case Specification*

UC ID and Name:	<b>UC-31.0: Report Advisor while chatting</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 01<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	
Trigger:	N/A		
Description:	User can report advisor when they feel unsatisfied.		
Preconditions:	<b>PRE-31.1:</b> User must be in conversation with advisor.		
Post conditions:	<b>POST-31.1.</b> The success alert is appeared.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on “Tu vấn” on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on domain which want to be advised.</li> <li>6. Click on “Tu vấn ngay” button of advisor want to be advised.</li> <li>7. When advisor starts a conversation, the website will redirect to Page conversation.</li> <li>8. Click on setting icon in the top right.</li> <li>9. Click on “Báo cáo Tu vấn viên”.</li> <li>10. Type report content to input field.</li> <li>11. Click on “Chọn 1 ảnh” button.</li> <li>12. Choose a image which is proof of advisor.</li> <li>13. Click on “Báo cáo” button.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:	<ol style="list-style-type: none"> <li>1. Customer did not fill full information, such as: report content or image before report.</li> </ol>		
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:	<b>B12</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.21 View conversation history**

*Use Case Specification*

UC ID and Name:	<b>UC-32.0: View conversation history</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 02<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	User can view all chat history which is finished.		
Preconditions:	<b>PRE-32.1:</b> User has to log into the system.		
Post conditions:	<b>POST-32.1.</b> The website will open history conversation page in a new tab.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on “Lịch sử tư vấn”.</li> <li>6. Choose conversation in the left to display conversation history in the right.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.22 Make an appointment with Advisor**

*Use Case Specification*

UC ID and Name:	<b>UC-33.0: Make an appointment with Advisor</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jun 01<sup>th</sup>, 2017</b>
Primary Actor:	Customer	Secondary Actor:	
Trigger:	N/A		
Description:	Customer can book an appointment with Advisor.		
Preconditions:	<b>PRE-33.1:</b> Customer is logged into the system. <b>PRE-33.2:</b> Customer is on Appointment page. <b>PRE-33.2:</b> Has advisor ready in time.		
Post conditions:	<b>POST-33.1:</b> System sends confirm email to Customer and Advisor.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User inputs field and clicks on button “Tìm kiếm”.</li> <li>2. Browser displays all Advisor matching conditions.</li> <li>3. User clicks on any Advisor.</li> <li>4. Browser shows Advisor’ information dialog.</li> <li>5. User clicks on button “Đặt lịch”.</li> <li>6. Browser shows successful dialog.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B18, B19, B26, B27, B28</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.23 Cancel an appointment with Advisor**

*Use Case Specification*

UC ID and Name:	<b>UC-34.0: Cancel an appointment with Advisor</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jun 01<sup>th</sup>, 2017</b>
Primary Actor:	Customer	Secondary Actor:	
Trigger:	N/A		
Description:	Customer can cancel an appointment with Advisor.		
Preconditions:	<b>PRE-34.2:</b> Customer is on Appointment's history page. <b>PRE-34.1:</b> Customer has a waiting appointment with Advisor.		
Post conditions:	<b>POST-34.1:</b> Customer cancels an appointment successfully.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User clicks on button “Hủy lịch hẹn” on row which want to cancel an appointment.</li> <li>2. System shows confirm dialog.</li> <li>3. User clicks on button “Đồng ý”.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	Low		
Business Rules:	<b>B22</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.2.24 View appointment history**

*Use Case Specification*

UC ID and Name:	<b>UC-35.0: View appointment history</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 02<sup>th</sup>, 2017</b>
Primary Actor:	<b>Customer</b>	Secondary Actor:	<b>Advisor</b>
Trigger:	N/A		
Description:	User can view all appointment history.		
Preconditions:	<b>PRE-35.1:</b> User has to log into the system.		
Post conditions:	<b>POST-35.1:</b> The website will redirect to history conversation page.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on “Lịch hẹn cá nhân”.</li> <li>6. Browser shows “Lịch hẹn cá nhân” page.</li> </ol>		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.3 Advisor

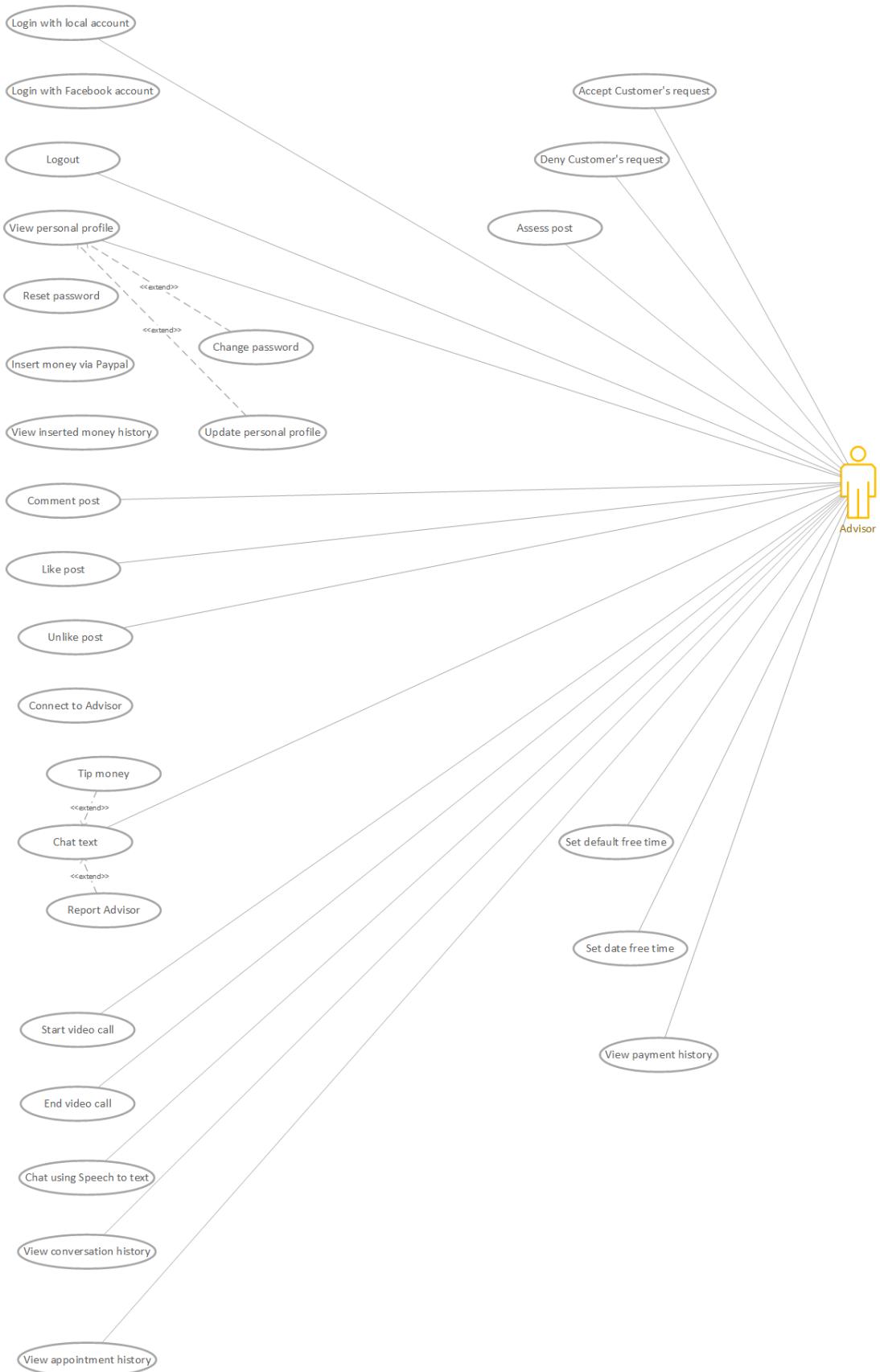


Figure 3.4 : Use case diagram of Registered user actor

### **3.2.3.3.1      Accept Customer's chat request**

*Use Case Specification*

UC ID and Name:	<b>UC-36.0: Accept Customer's chat request</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 06<sup>th</sup>, 2017</b>
Primary Actor:	<b>Advisor</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Accept request when customer connect to advisor.		
Preconditions:	<b>PRE-36.1:</b> Advisor has to log into the system and is not in any conversation. <b>PRE-36.2:</b> Customer sends request to advisor.		
Post conditions:	<b>POST-36.1:</b> The website will redirect to Page conversation.		
Normal Flow:	1. User visits home page. 2. When customer sends request, browser displays a popup. 3. User clicks “Bắt đầu tư vấn” to accept this request.		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B11</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.3.2 Deny Customer's chat request**

*Use Case Specification*

UC ID and Name:	<b>UC-37.0: Deny Customer's chat request</b>		
Created by:	<b>TrungVT</b>	Created date:	<b>Jun 05<sup>th</sup>, 2017</b>
Primary Actor:	<b>Advisor</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Deny request when customer connects to advisor.		
Preconditions:	<b>PRE-37.1:</b> Advisor has to log into the system and is not in any conversation. <b>PRE-37.2:</b> Customer sends request to advisor.		
Post conditions:	<b>POST-37.1:</b> The popup is disappeared.		
Normal Flow:	1. User visits home page. 2. Browser displays home page. 3. When customer sends request, browser displays a popup. 4. User clicks on “Tù chối” to deny this request.		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.3.3 Set default free time on week**

*Use Case Specification*

UC ID and Name:	<b>UC-38.0: Set default free time on week</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jun 02<sup>th</sup>, 2017</b>
Primary Actor:	<b>Advisor</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Advisor can set default free time on week from Monday to Saturday. This is not set free time for each day. This is like preferences for set free time each day.		
Preconditions:	<b>PRE-38.1:</b> Advisor is logged into the system.		
Post conditions:	<b>POST-38.1:</b> Advisor sets default free time successfully.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on button “Cài đặt lịch hẹn mặc định”.</li> <li>6. Browser displays “Cài đặt lịch hẹn mặc định” page.</li> <li>7. User clicks on time of each day to set free time.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	Low		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.3.4 Set free time by date

Use Case Specification

UC ID and Name:	<b>UC-39.0: Set free time by date</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jun 02<sup>th</sup>, 2017</b>
Primary Actor:	Advisor	Secondary Actor:	
Trigger:	N/A		
Description:	Advisor can set free time by date which can viewed by customer.		
Preconditions:	<b>PRE-39.1:</b> Advisor is logged into the system.		
Post conditions:	<b>POST-39.1:</b> Advisor sets free time by date successfully. <b>POST-39.2:</b> System pushes free time to firebase.		
Normal Flow:	1. User visits home page. 2. Browser displays home page. 3. User hovers on display name on the navigation bar. 4. Page shows dropdown. 5. User clicks on button “Cài đặt lịch hẹn”. 6. Browser displays “Cài đặt lịch hẹn” page. 7. User chooses day which want to setting and clicks on button “Cài đặt”. 8. Page displays time block of this day. 9. User clicks on time block which ready on this day.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:	<b>B21</b>		
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.3.5 View payment history**

*Use Case Specification*

UC ID and Name:	<b>UC-40.0: View payment history</b>		
Created by:	<b>DatBV</b>	Created date:	<b>May 25<sup>th</sup>, 2017</b>
Primary Actor:	Advisor	Secondary Actor:	
Trigger:	N/A		
Description:	Advisor can view payment history.		
Preconditions:	<b>PRE-40.1:</b> Advisor is logged into the system.		
Post conditions:	<b>POST-40.1:</b> Browser displays View payment history page.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. User visits home page.</li> <li>2. Browser displays home page.</li> <li>3. User hovers on display name on the navigation bar.</li> <li>4. Page shows dropdown.</li> <li>5. User clicks on button “Xem lịch sử thanh toán”.</li> <li>6. Browser displays “Lịch sử thanh toán” page.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.3.6 Assess Post

*Use Case Specification*

UC ID and Name:	<b>UC-41.0: Assess Post</b>		
Created by:	<b>DatBV</b>	Created by:	<b>May 26<sup>th</sup>, 2017</b>
Primary Actor:	Advisor	Secondary Actor:	
Trigger:	N/A.		
Description:	Advisor participates in assess post with system		
Preconditions:	<b>PRE-41.1:</b> Administrator creates post successful. <b>PRE-41.2:</b> Advisor is in the same sub-domain with post.		
Post conditions:	N/A		
Normal Flow:	1. User clicks on “Xem chi tiết” on email which sent by system to Assess post. 2. Browser displays this post. 3. User assesses post by click on button “Nội dung không phù hợp” or “Nội dung phù hợp”.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4 Administrator

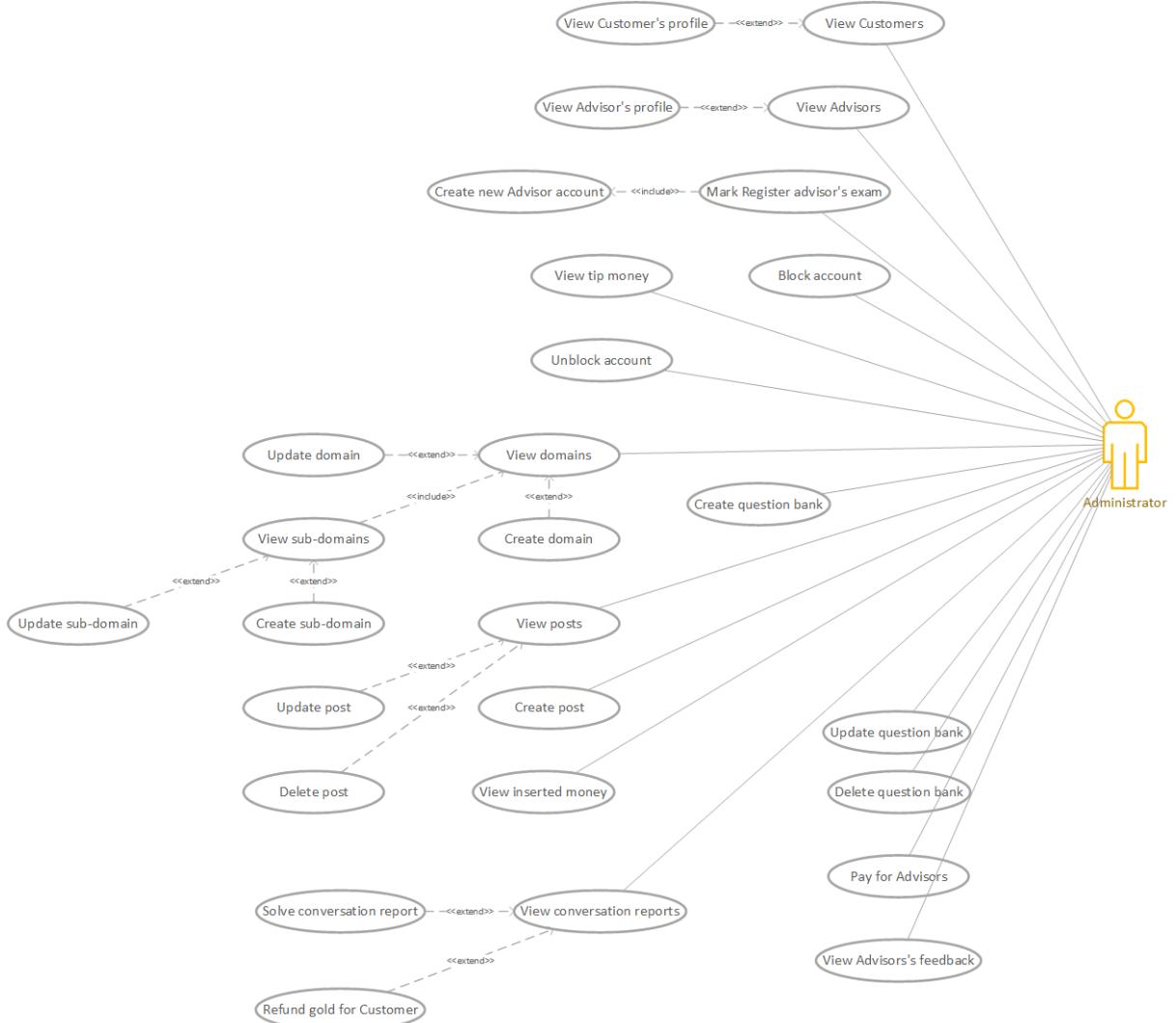


Figure 3.5 : Use case diagram of Administrator actor

### 3.2.3.4.1 View Customers

*Use Case Specification*

UC ID and Name:	<b>UC-42.0: View customers</b>		
Created by:	<b>VuNS</b>	Created date:	<b>Jun 08<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view customer's information.		
Preconditions:	<b>PRE-42.1:</b> Login by Administrator account. <b>PRE-42.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-42.1:</b> System displays customer's data table.		
Normal Flow:	1. Admin clicks on “Quản lý khách hàng” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách khách hàng”. 4. Page displays customer list data table.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.2 View Customer's profile**

*Use Case Specification*

UC ID and Name:	<b>UC-43.0: View customer's profile</b>		
Created by:	<b>VuNS</b>	Created date:	<b>Jul 04<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view a customer's detail information.		
Preconditions:	<b>PRE-43.1:</b> Login by Administrator account. <b>PRE-43.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-43.1:</b> System displays dialog include customer's detail information.		
Normal Flow:	1. Admin clicks on “Quản lý khách hàng” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách khách hàng”. 4. Page displays customer list data table. 5. Admin clicks on customer name to display dialog include customer's detail information.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.3 View Advisors

*Use Case Specification*

UC ID and Name:	<b>UC-44.0: View advisors</b>		
Created by:	<b>VuNS</b>	Created date:	<b>Jun 09<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view advisor's information.		
Preconditions:	<b>PRE-44.1:</b> Login by Administrator account. <b>PRE-44.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-44.1:</b> System displays advisor's data table.		
Normal Flow:	1. Admin clicks on “Quản lý Tư vấn viên” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách Tư vấn viên”. 4. Page displays advisors list data table.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.4 View Advisor's profile

Use Case Specification

UC ID and Name:	<b>UC-45.0: View advisor's profile</b>		
Created by:	<b>VuNS</b>	Created date:	<b>Jun 12<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view an advisor's detail information.		
Preconditions:	<b>PRE-45.1:</b> Login by Administrator account. <b>PRE-45.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-45.1:</b> System displays dialog include advisor's detail information.		
Normal Flow:	1. Admin clicks on “Quản lý Tư vấn viên” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách Tư vấn viên”. 4. Page displays advisors list data table. 5. Admin clicks on advisor name to display dialog include advisor's detail information.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.5 Create new Advisor account

Use Case Specification

UC ID and Name:	<b>UC-46.0: Create new advisor account</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can create a new advisor.		
Preconditions:	<b>PRE-46.1:</b> Login by Administrator account. <b>PRE-46.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-46.1:</b> System displays dialog to inform that created advisor account successfully. <b>POST-46.2:</b> System sends email to inform that he/she becomes advisor of system and confirms email.		
Normal Flow:	1. Admin clicks on “Quản lý Tư vấn viên” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Đăng ký Tư vấn viên”. 4. Admin clicks on “Chấm bài” action. 5. Admin clicks on canvas circle to mark and click “Tính điểm” button. 6. Admin clicks on “Đăng ký Tư vấn viên” button.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.6      *Mark exam for Register Advisor***

*Use Case Specification*

UC ID and Name:	<b>UC-47.0: Mark exam for register advisor</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can calculate mark for advisor test.		
Preconditions:	<b>PRE-47.1:</b> Login by Administrator account. <b>PRE-47.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-47.1:</b> System marks the test.		
Normal Flow:	1. Admin clicks on “Quản lý Tư vấn viên” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Đăng ký Tư vấn viên”. 4. Admin clicks on “Chấm bài” action. 5. Admin clicks on canvas circle to mark and click “Tính điểm” button.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.7 *Block account*

*Use Case Specification*

UC ID and Name:	<b>UC-48.0: Block account</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can block an advisor or customer account.		
Preconditions:	<b>PRE-48.1:</b> Login by Administrator account. <b>PRE-48.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-48.1:</b> System blocks that account. <b>POST-48.2:</b> System displays alert to notify that block successful. <b>POST-48.3:</b> System sends email to user to inform that this account was blocked.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Admin clicks on “Quản lý Tư vấn viên”/ “Quản lý khách hàng” at the side bar.</li> <li>2. Page displays drop down.</li> <li>3. Admin clicks on “Danh sách Tư vấn viên”/ “Danh sách khách hàng”.</li> <li>4. Page displays advisors/customers list data table.</li> <li>5. Admin clicks on “Hoạt động” in status column to display alert to confirm block account or not.</li> <li>6. Admin clicks on “Đồng ý” button to block account.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.8      ***Unblock account***

*Use Case Specification*

UC ID and Name:	<b>UC-49.0: Unblock account</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>May 22<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can unblock an advisor or customer account.		
Preconditions:	<b>PRE-49.1:</b> Login by Administrator account. <b>PRE-49.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-49.1:</b> System unblocks that account. <b>POST-49.2:</b> System displays alert to notify that unblock successful. <b>POST-49.3:</b> System sends email to user to inform that this account was unblocked.		
Normal Flow:	<ol style="list-style-type: none"> <li>1. Admin clicks on “Quản lý Tư vấn viên”/ “Quản lý khách hàng” at the side bar.</li> <li>2. Page displays drop down.</li> <li>3. Admin clicks on “Danh sách Tư vấn viên”/ “Danh sách khách hàng”.</li> <li>4. Page displays advisors/customers list data table.</li> <li>5. Admin clicks on “Đã bị khóa” in status column to display alert to confirm unblock account or not.</li> <li>6. Admin clicks on “Đồng ý” button to unblock account.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.9 View Domains

*Use Case Specification*

UC ID and Name:	<b>UC-50.0: View domains</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 16<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view domains.		
Preconditions:	<b>PRE-50.1:</b> Login by Administrator account. <b>PRE-50.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-50.1:</b> System displays a list of domains.		
Normal Flow:	1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Page displays a list of domains.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.10      Update Domain**

*Use Case Specification*

UC ID and Name:	<b>UC-51.0: Update domain</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 12<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can edit information in a domain.		
Preconditions:	<b>PRE-51.1:</b> Login by Administrator account. <b>PRE-51.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-51.1:</b> System saves the updated domain information.		
Normal Flow:	1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Admin clicks on “pliers” symbol in top-right slick carousel of a domain and click on “Chỉnh sửa”. 5. Input new information and click “Xác nhận” button.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.11 Create New Domain

*Use Case Specification*

UC ID and Name:	<b>UC-52.0: Create new domain</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 15<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can create new domain.		
Preconditions:	<b>PRE-52.1:</b> Login by Administrator account. <b>PRE-52.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-52.1:</b> System creates new domain.  Normal Flow: 1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Admin clicks on “Tạo lĩnh vực mới” button in top-right slick of web page. 5. Admin inputs information and click “Xác nhận” button.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.12 View Sub-domain

*Use Case Specification*

UC ID and Name:	<b>UC-53.0: View sub-domain</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 16<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view sub-domain.		
Preconditions:	<b>PRE-53.1:</b> Login by Administrator account. <b>PRE-53.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-53.1:</b> System creates new domain.  1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Admin clicks on “pliers” symbol in top-right slick carousel of a domain and clicks on “Xem chuyên khoa”. 5. Page displays a list of sub-domains.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.13 Update Sub-domain

*Use Case Specification*

UC ID and Name:	<b>UC-54.0: Update sub-domain</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 16<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can edit new information for a sub-domain.		
Preconditions:	<b>PRE-54.1:</b> Login by Administrator account. <b>PRE-54.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-54.1:</b> System creates new domain.  1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Admin clicks on “pliers” symbol in top-right slick carousel of a domain and clicks on “Xem chuyên khoa”. 5. Click on “Sửa” action of any sub-domain. 6. Input new information and click on “Xác nhận” button.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.14 Create New Sub-domain

*Use Case Specification*

UC ID and Name:	<b>UC-55.0: Create new sub-domain</b>		
Created by:	<b>CongHPA</b>	Created date:	<b>Jun 15<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can create new sub-domain.		
Preconditions:	<b>PRE-55.1:</b> Login by Administrator account. <b>PRE-55.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-55.1:</b> System saves new sub-domain.		
Normal Flow:	1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách lĩnh vực”. 4. Admin clicks on “pliers” symbol in top-right slick carousel of a domain and clicks on “Xem chuyên khoa”. 5. Admin clicks on “Tạo chuyên khoa mới” on top-right of sub-domain data table. 6. Admin inputs sub-domain name and clicks on “Xác nhận” button.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.15 Create New Post

*Use Case Specification*

UC ID and Name:	<b>UC-56.0: Create new post</b>		
Created by:	TrinhDN	Created date:	May 19 <sup>th</sup> , 2017
Primary Actor:	Administrator	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can create new post.		
Preconditions:	<b>PRE-56.1:</b> Login by Administrator account. <b>PRE-56.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-56.1:</b> System saves new post. <b>POST-56.2:</b> System sends email to advisors to assess this post.		
Normal Flow:	1. Admin clicks on “Quản lý bài đăng” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Tạo bài đăng mới”. 4. Admin inputs data and clicks on “Đăng bài” button”.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.16 Edit Post

*Use Case Specification*

UC ID and Name:	<b>UC-57.0: Edit post</b>		
Created by:	<b>TrinhDN</b>	Created date:	<b>May 19<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can edit post.		
Preconditions:	<b>PRE-57.1:</b> Login by Administrator account. <b>PRE-57.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-57.1:</b> System saves new post's data.  1. Admin clicks on “Quản lý bài đăng” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Tạo bài đăng mới”. 4. Admin clicks on “Sửa” action corresponding to the post to edit. 5. Admin inputs new data and clicks on “Lưu thay đổi” button.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.17 Delete post

*Use Case Specification*

UC ID and Name:	<b>UC-58.0: Delete post</b>		
Created by:	TrinhDN	Created date:	May 22 <sup>th</sup> , 2017
Primary Actor:	Administrator	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can change status of post to delete.		
Preconditions:	<b>PRE-58.1:</b> Login by Administrator account. <b>PRE-58.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-58.1:</b> Post status is changed to "Delete".  1. Admin clicks on "Quản lý bài đăng" at the side bar. 2. Page displays drop down. 3. Admin clicks on "Tạo bài đăng mới". 4. Admin clicks on "Xóa" action corresponding to the post to delete. 5. Admin clicks on "Xác nhận xóa".		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.18 View conversation report

Use Case Specification

UC ID and Name:	<b>UC-59.0: View conversation report</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 14<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view list of conversation report.		
Preconditions:	<b>PRE-59.1:</b> Login by Administrator account. <b>PRE-59.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-59.1:</b> System displays list of conversation report.		
Normal Flow:	1. Admin clicks on “Quản lý tư vấn” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách báo cáo”. 4. System display all conversation report in table.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.19      Solve conversation report**

*Use Case Specification*

UC ID and Name:	<b>UC-60.0: Solve conversation report</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 26<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can solve report in pending process.		
Preconditions:	<b>PRE-60.1:</b> Login by Administrator account. <b>PRE-60.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-60.1:</b> Report has been resolved.  1. Admin clicks on “Quản lý tư vấn” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách báo cáo”. 4. Admin clicks on “Giải quyết” action corresponding to the conversation to resolve. 5. Admin inputs description and clicks on “Xác nhận”.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.20 Refund gold for customer**

*Use Case Specification*

UC ID and Name:	<b>UC-61.0: Refund gold for customer</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 18<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	When customer's report is true. Gold in that conversation will return to customer, then if report is wrong gold will be pay to advisor.		
Preconditions:	<b>PRE-61.1:</b> Login by Administrator account. <b>PRE-61.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-61.1:</b> System returns gold to customer. <b>POST-61.2:</b> Customer will receive gold back.		
Normal Flow:	1. Admin clicks on “Quản lý tư vấn” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách báo cáo”. 4. Admin clicks on “Hoàn trả KH/Trả tiền TVV” action corresponding to the conversation. 5. Admin clicks on “Xác nhận”.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.21 View inserted money

*Use Case Specification*

UC ID and Name:	<b>UC-62.0: View inserted money</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 21<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view list of customer's inserted money.		
Preconditions:	<b>PRE-62.1:</b> Login by Administrator account. <b>PRE-62.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-62.1:</b> System displays list of customer's inserted money.		
Normal Flow:	1. Admin clicks on “Quản lý tài chính” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách nạp tiền”. 4. System displays list of customer's inserted money.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.4.22 View tip money

*Use Case Specification*

UC ID and Name:	<b>UC-63.0: View tip money</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 24<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view list of tip money.		
Preconditions:	<b>PRE-63.1:</b> Login by Administrator account. <b>PRE-63.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-63.1:</b> System displays list of tip money.		
Normal Flow:	1. Admin clicks on “Quản lý tài chính” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách Tip”. 4. System displays list of tip money.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.23      Create question bank**

*Use Case Specification*

UC ID and Name:	<b>UC-64.0: Create question bank</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 14<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can create question corresponding to subdomain.		
Preconditions:	<b>PRE-64.1:</b> Login by Administrator account. <b>PRE-64.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-64.1:</b> System saves new question.  1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Ngân hàng câu hỏi”. 4. System saves new question.		
Normal Flow:	N/A		
Alternative Flows:			
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.24      Update question bank**

*Use Case Specification*

UC ID and Name:	<b>UC-65.0: Update question bank</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 17<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can edit question in question bank.		
Preconditions:	<b>PRE-65.1:</b> Login by Administrator account. <b>PRE-65.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-65.1:</b> System saves updated question's data.  1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Ngân hàng câu hỏi”. 4. Admin clicks on “Sửa” action corresponding to question. 5. Admin edits data and “Cập nhật” button.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.25 Delete question bank**

*Use Case Specification*

UC ID and Name:	<b>UC-66.0: Delete question bank</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 20<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can delete question in question bank.		
Preconditions:	<b>PRE-66.1:</b> Login by Administrator account. <b>PRE-66.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-66.1:</b> Question is deleted.  1. Admin clicks on “Quản lý lĩnh vực” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Ngân hàng câu hỏi”. 4. Admin clicks on “Xóa” action corresponding to question. 5. Admin clicks on “Xác nhận” button.		
Normal Flow:			
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.26 View advisor's feedback**

*Use Case Specification*

UC ID and Name:	<b>UC-67.0: View advisor's feedback</b>		
Created by:	<b>HuyTQ</b>	Created date:	<b>Jul 25<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can view feedback of customer in conversation.		
Preconditions:	<b>PRE-67.1:</b> Login by Administrator account. <b>PRE-67.2:</b> Go to the Administrator's management area.		
Post conditions:	<b>POST-67.1:</b> System displays feedback data table.		
Normal Flow:	1. Admin clicks on “Quản lý Tư vấn viên” at the side bar. 2. Page displays drop down. 3. Admin clicks on “Danh sách Tư vấn viên”. 4. Admin clicks on “Xem” action corresponding to advisor. 5. Page displays list of feedbacks.		
Alternative Flows:	N/A		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### **3.2.3.4.27 Pay money for Advisor (assume via Bank)**

*Use Case Specification*

UC ID and Name:	<b>UC-68.0: Pay money for Advisor (assume via Bank)</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jul 26<sup>th</sup>, 2017</b>
Primary Actor:	<b>Administrator</b>	Secondary Actor:	
Trigger:	N/A		
Description:	Administrator can pay money for Advisor via Bank account.		
Preconditions:	<b>PRE-68.1:</b> Administrator is logged into the system. <b>PRE-68.2:</b> Administrator is on Dashboard page.		
Post conditions:	<b>POST-68.1:</b> Browser shows successful dialog.		
Normal Flow:	1. Click on “Quản lý Tư vấn viên” at the side bar. 2. Click on “Danh sách Tư vấn viên”. 3. Browser displays “Danh sách Tư vấn viên” page. 4. Admin clicks on “Thanh toán cho tất cả” to pay all advisor, or can click on “Thanh toán” on row which want to payment to payment.		
Alternative Flows:	N/A.		
Exceptions:			
Priority:	Medium		
Frequency of Use:	Medium		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.2.3.5 System

#### 3.2.3.5.1 Appointment helper

*Use Case Specification*

UC ID and Name:	<b>UC-69.0: Appointment helper</b>		
Created by:	<b>DatBV</b>	Created date:	<b>Jul 14<sup>th</sup>, 2017</b>
Primary Actor:		Secondary Actor:	
Trigger:	N/A		
Description:	This is Windows service supports system about management appointment. It changes appointment status if it is not occurred. It sends email to customer and advisor to remind that the appointment is coming soon.		
Preconditions:	N/A		
Post conditions:	N/A		
Normal Flow:	<p><b>Send email to Advisor and Customer</b></p> <ol style="list-style-type: none"> <li>1. System checks now time.</li> <li>2. If now time close to appointment time, continue. Else, back to step 1.</li> <li>3. System checks appointment. If have appointment, continue. Else, back to step 1.</li> <li>4. System sends email to all customer and advisor in all appointment found in step 3.</li> </ol> <p><b>Change appointment status</b></p> <ol style="list-style-type: none"> <li>1. System checks now time.</li> <li>2. If now time is over appointment time more than 2 minutes, continue. Else, back to step 1.</li> <li>3. System checks pending appointment. If have appointment, continue. Else, back to step 1.</li> <li>4. System changes appointment status to not occurred of all appointment found in step 3.</li> </ol>		
Alternative Flows:	N/A		
Exceptions:			
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:			
Assumptions:	Server is stable running. SQL Server is stable running.		

### 3.3 Non-functional Requirement

#### 3.3.1 Security

- MindDoctor is built from Entity Framework, so the potential risk of SQL Injection is mitigated to the lowest.
- MindDoctor used ASP Identity system for authorizing and authenticating.
- Applied Anti Forgery Token of ASP.NET to prevent cross-site request.
- The security matrix is as the following table:

Function	Guest	Customer	Advisor	Administrator
Register to become Customer	✓			
Verify email address	✓	✓	✓	
Register to become Advisor	✓	✓		
Take qualification exam	✓	✓		
View post	✓	✓	✓	✓
Share post on Facebook	✓	✓	✓	✓
Search post	✓	✓	✓	✓
View Advisors's schedule	✓	✓	✓	✓
View online Advisors	✓	✓	✓	✓
View online Advisors by searching Advisor's information	✓	✓	✓	✓
View Advisor's information	✓	✓	✓	✓
Login with local account		✓	✓	✓
Login with Facebook account		✓		
Logout		✓	✓	✓
Reset password		✓	✓	✓
Change password		✓	✓	✓
View personal profile		✓	✓	
Update personal profile		✓	✓	
Insert money via Paypal		✓		
View inserted money history		✓		
Comment post		✓	✓	✓
Like post		✓	✓	✓
Unlike post		✓	✓	✓
Connect to Advisor		✓		
Chat text		✓	✓	

Start video call		✓	✓	
End video call		✓	✓	
Chat using Speech to Text		✓	✓	
Tip money for Advisor		✓		
Rate Advisor after finishing conversation		✓		
Report Advisor while chatting		✓		
View conversation history		✓		
Make an appointment with Advisor		✓		
Cancel an appointment with Advisor		✓		
View appointment history		✓	✓	
Accept Customer's chat request			✓	
Deny Customer's chat request			✓	
Set default free time on week			✓	
Set free time by date			✓	
View payment history			✓	
Assess Post			✓	
View Customers				✓
View Customer's profile				✓
View Advisors				✓
View Advisor's profile				✓
Create new Advisor account				✓
Mark exam for Register Advisor				✓
Block account				✓
Unblock account				✓
View domain				✓
Update domain				✓
Create new domain				✓
View sub-domain				✓
Update sub-domain				✓
Create new sub-domain				✓
Create new post				✓
Edit post				✓
Delete post				✓
View conversation report				✓
Solve conversation report				✓

Refund gold for customer				✓
View inserted money				✓
View tip money				✓
Create questions bank				✓
Update questions bank				✓
Delete questions bank				✓
View Advisor's feedback				✓
Pay money for Advisor (assume via Bank)				✓
Appointment helper				✓
Register to become Customer				✓

*Table 3-4: Security matrix*

### 3.3.2 Maintainability & Extensibility

- Apply MVC model for system.
- We wrote automation test including Unit Tests by using NUnit.
- Strictly follow the coding convention to help improve readability of source code and make the website more maintainable.
- Develop website based on Object Oriented Programming paradigm in orders to increase maintainability and extendibility.

### 3.3.3 Availability and Scalability

- Using Azure virtual cloud service for automatic horizontal scalability.
- Apply some third-party services for reducing development time:
  - Mailboxlayer for verifying email.
  - Firebase for pushing realtime interaction.
  - Windows service for handle appointment between customer and advisor.
  - PaypalAPI for online payment.
  - Numverify for verify phone number.
  - Currencylayer to exchange rate and currency conversion.

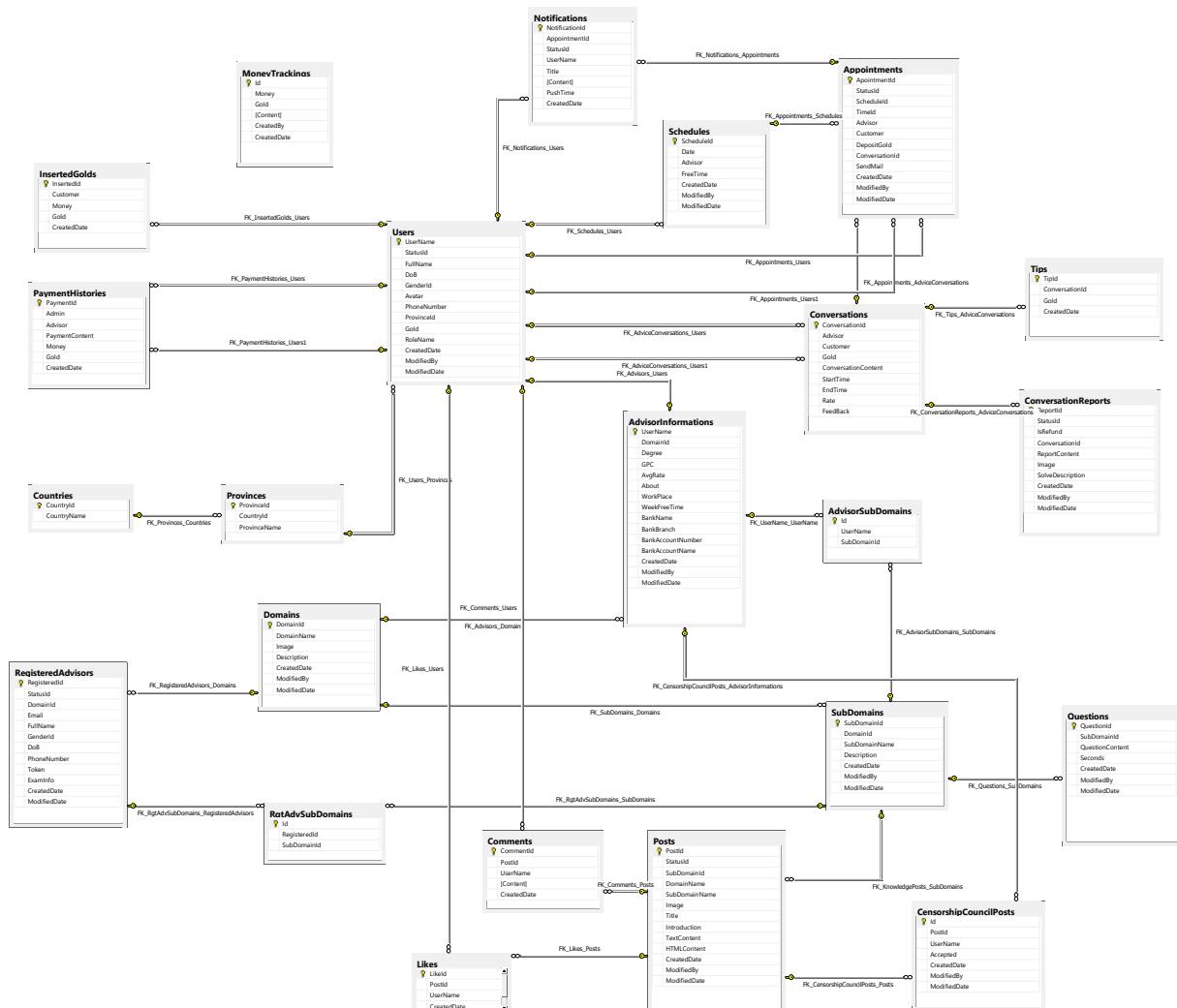
### 3.3.4 Performance

- Web front-end uses AngularJS that uses several clever techniques to minimize the number of costly DOM operations required to update the UI.
- Add indexes to SQL server, which make query incurring filtering and sorting run much faster.
- Using Entity Framework, which is loaded once only, from second time data is save in cache. This help to increase performance when loads data from database.
- Using Entity Framework, which use lazy loading, means delaying the loading of related data, until you specifically request for it.
- Caching data on client to decrease time of load.

### 3.3.5 Usability

- ✓ All of screen should be elegant and simple.
  - ✓ Many pages are single page loading.
  - ✓ Using flat-design.
  - ✓ Mobile responsiveness – website can handle different screen sizes and slow connections.
  - ✓ Simplicity – focus on what's important. Display necessary information, what customer want them to do.
  - ✓ Applied Nielsen principles.
  - ✓ List advisors online, booking an appointment are pushed realtime.
  - ✓ All attributes are provided for non-text elements, such as images, description.
  - ✓ Search box is wide enough, so that users can see what they have typed.
  - ✓ Search is available on every page, not just the homepage.
  - ✓ Advisor online list is displayed base on ranking, number of conversation of advisor that make customer easier to choose advisor.
  - ✓ Post is displayed base on customer behaviors as what they search, like...

### 3.4 Entity Relationship Diagram



*Figure 3.6 : ERD of MindDoctor system*

# Chapter 4 : Software Design

## 4.1 Purpose

This chapter is to give the developer team of what the system's architecture is, and how they should be implemented. This chapter consists of:

- Architecture overview
- Component diagram
- Detailed design
- Detailed description of components
- Database design

## 4.2 Architecture Overview

### 4.2.1 System Architecture

#### 4.2.1.1 Diagram

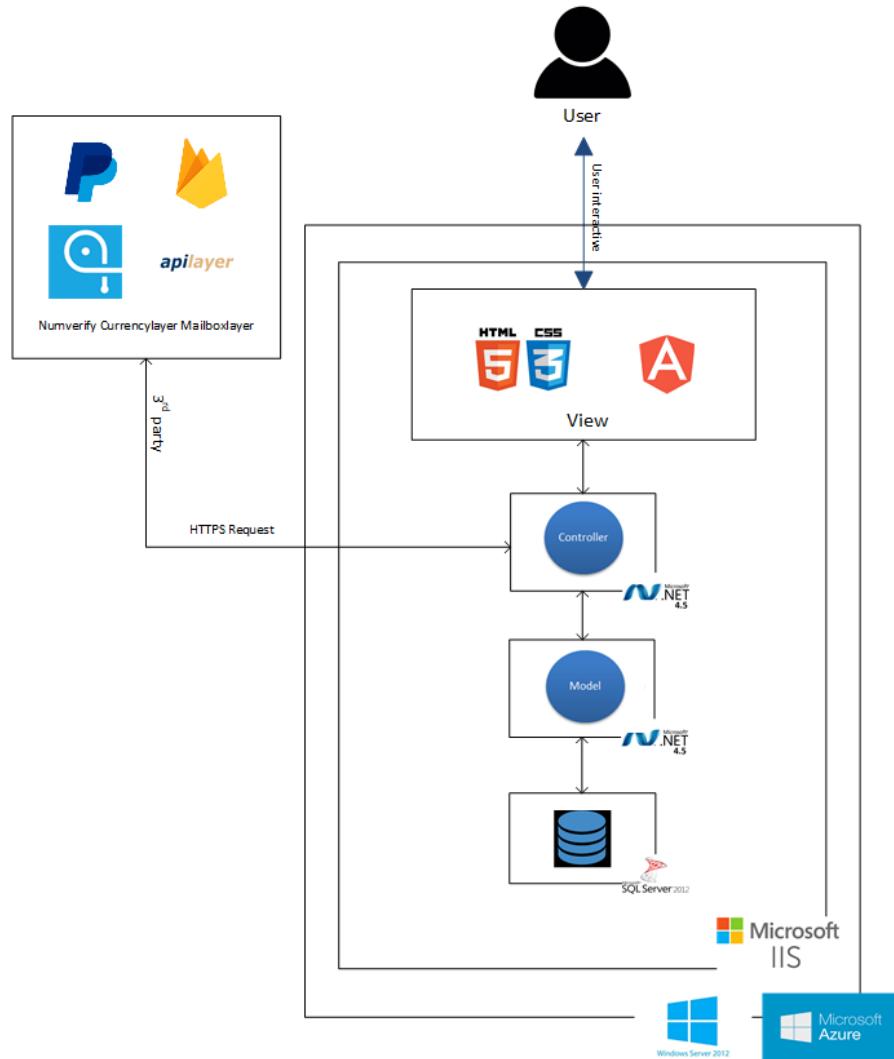


Figure 4-1: MindDoctor's system architecture

#### **4.2.1.2 Explanation**

#### **4.2.2 System Architecture Explanation**

The entire project will be deployed on Microsoft Azure. We aim at delivering a secured, responsive and highly available system. In the following section, we will explain the function and mechanism of each unit in the system architecture design.

##### **4.2.2.1 Microsoft Azure**



*Figure 4-2: Microsoft Azure*

**Microsoft Azure** is an open, flexible, enterprise-grade cloud computing platform. Move faster, do more, and save money with Infrastructure as a Service (IaaS) + Platform as a service (PaaS).

##### **4.2.2.2 IIS**



*Figure 4-3: IIS 10*

**IIS** is a web server, run on Windows OS. It is the most common solution for web hosting Microsoft-based web application. IIS is feature-rich with different management tools and add-on like URL Rewrite module, Authentication module.

##### **4.2.2.3 Windows Server**



*Figure 4-4: Windows Server 2012*

**Windows Server 2012** is an operating system provided by Microsoft. Windows server 2012 provide easy-to-access and configure environment, including built-in Firewall, native support for .NET framework, IIS and Sql Server. **MindDoctor will be deployed on Windows Server VPS and is dependent on .NET framework installed on Windows Server 2012.**

#### 4.2.2.4 Microsoft .NET Framework 4.5



Figure 4-5: .NET Framework 4.5

The .NET Framework provides a comprehensive programming model for building all kinds of applications on Windows. Specifically, MindDoctor's back end is built using the .NET Framework 4.5.

#### 4.2.2.5 Microsoft SQL Server 2012 Enterprise



Figure 4-6: Sql Server 2012 Enterprise

Microsoft SQL Server 2012 Enterprise is a powerful and reliable free data management system that delivers a rich and reliable data store for lightweight Web Sites and desktop applications. With including full-text search component, SQL Server 2012 power MindDoctor's database management system needs.

#### 4.2.2.6 AngularJS



Figure 4-7: AngularJS

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax. Powered by Google, AngularJS is a proven framework for Single Page Application. In MindDoctor, AngularJS is used to create the front-end, which delivers smooth experience, and reduce server's workload.

#### 4.2.2.7 PaypalApi



Figure 4-8: PayPalApi

PayPal is the faster, safer way to send money, make an online payment, receive money or set up a merchant account. We used it to insert money for MindDoctor's customer.

#### 4.2.2.8 Mailboxlayer



Figure 4-9: Mailboxlayer

**Mailboxlayer** is simple and powerful email verification JSON API using SMTP, typo checks, syntax validation, and free and disposable provider filtering.

#### 4.2.2.9 Numverify



Figure 4-10: Numverify

**Numverify** is free, powerful JSON API offering international and national phone number validation and carrier, location and line type lookup for developers and businesses.

#### 4.2.2.10 Currencylayer



Figure 4-11: Currency Layer

**Cunrrencylayer** is free and startup-friendly currency converter JSON API for real-time and historical exchange rates - reliable and accurate, supporting 168 world currencies. We used it for exchange rate from USD to VND of insert money function.

#### 4.2.2.11 OpenTok



Figure 4-12: OpenTok

OpenTok is web RTC Platform for video, voice and messaging from TokBox. MindDoctor used it to build live interactive video and voice into website.

#### 4.2.2.12 Firebase



Figure 4-13: Firebase

**Firebase** provides a realtime database and backend as a service. The service provides application developers an API that allow application data to be synchronized across clients and stored on Firebase's cloud. MindDoctor used it to realtime connection for chatting and making an appointment.

## 4.3 Design of MindDoctor System

### 4.3.1 Architecture Layers Design

#### 4.3.1.1 MindDoctor – MVC Design

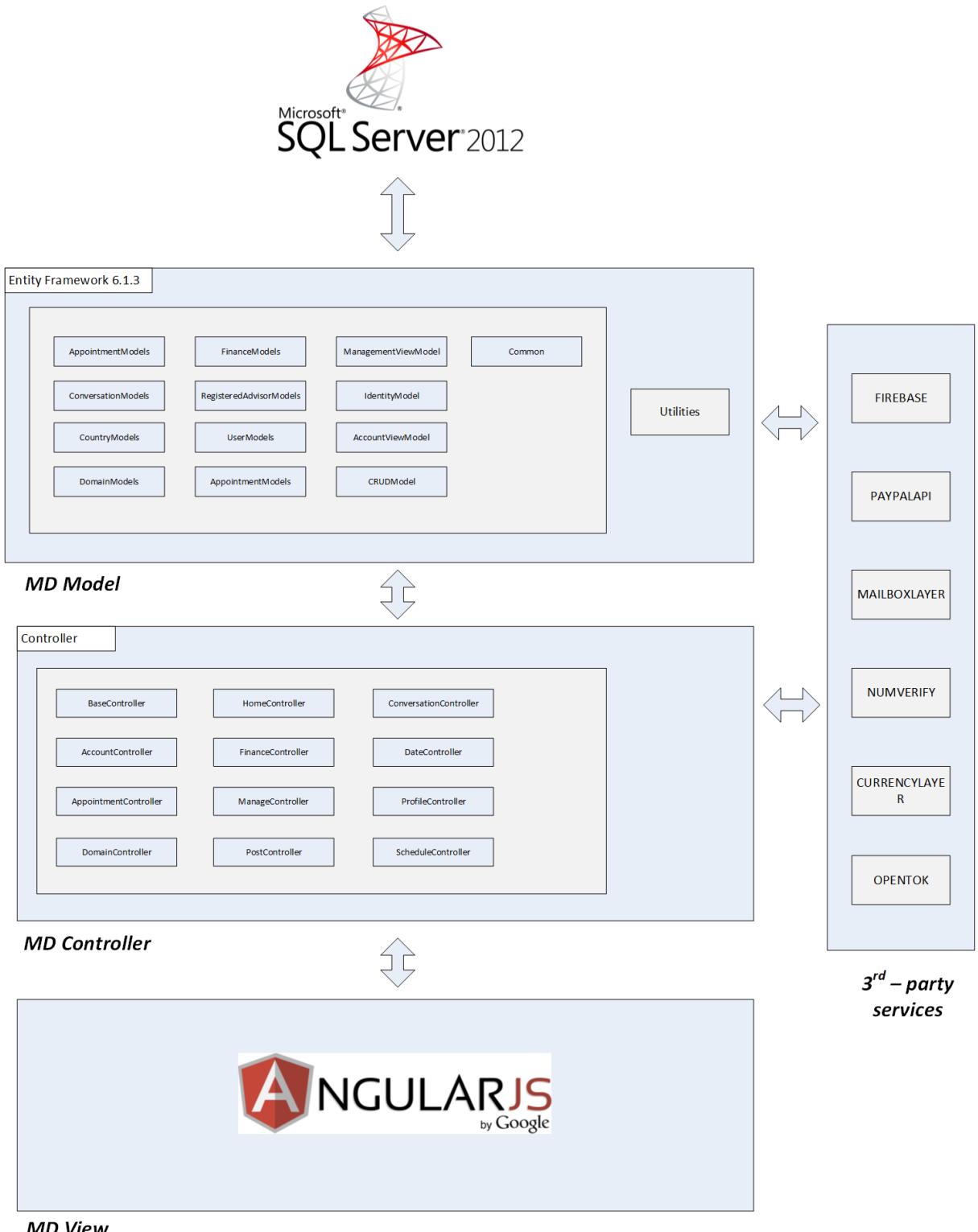


Figure 4-14: MindDoctor – Architecture design

No	Directory/File	Description
1	MindDoctor/	This folder contains everything of the application.
2	MindDoctor/App_Data	This folder stores application related storage files like DB files, XML files etcetera or we can say the physical store for data happens here.
3	MindDoctor/App_Start	This folder contains a set of static classes that are used as a set of configuration logic in Global.asax before.
4	MindDoctor/Area	Areas are logical grouping of Controller, Models and Views and other related folders for a module in MVC applications.
5	MindDoctor/bin	This folder contains all libraries built.
6	MindDoctor/Common	This folder contains all class shared all project.
7	MindDoctor/Content	This folder contains a default folder named "themes" that holds various jQuery version files, even included minified versions. It also stores files, like images, cascading style sheets etcetera. We can store any static files here.
8	MindDoctor/Controllers	This folder contains controllers of project.
9	MindDoctor/EntityFramework	This folder contains all class mapping with table in database generated.
10	MindDoctor/fonts	This file contains all font use in project.
11	MindDoctor/Models	This folder contain all models used in project.
12	MindDoctor/obj	This folder contains everything of debug or release after project run.
13	MindDoctor/Scripts	This folder contains all javascript files.
14	MindDoctor/Views	This folder contains all view .cshtml of project.
15	MindDoctor/Areas/Admin	This folder contains grouping of Controller, Models and Views and other related folders for admin module.
16	MindDoctor/Areas/Advisor	This folder contains grouping of Controller, Models and Views and other related folders for advisor module.
17	MindDoctor/Areas/Customer	This folder contains grouping of Controller, Models and Views and other related folders for customer module.

## 4.3.2 Database Design

### 4.3.2.1 Database Diagram

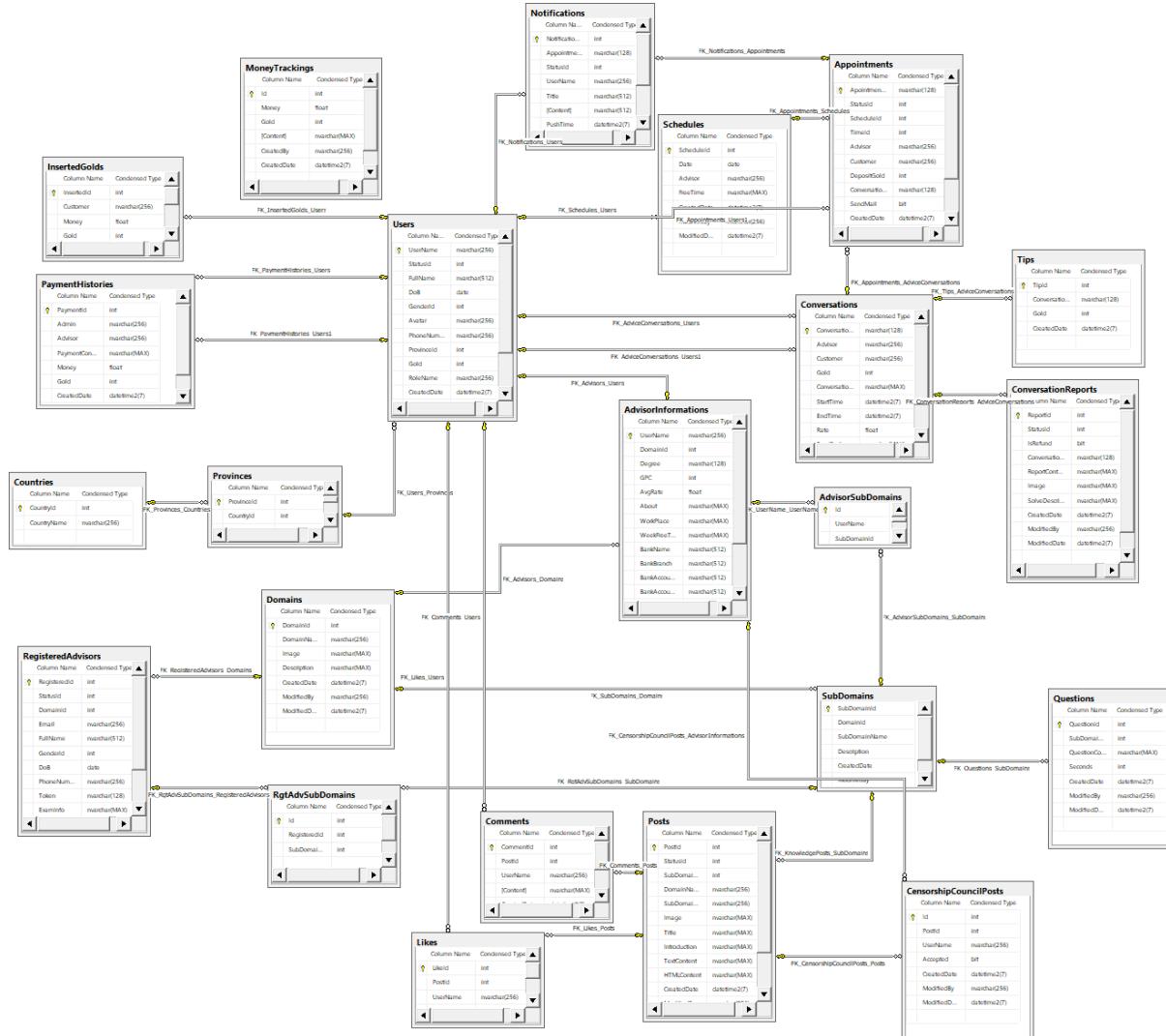


Figure 4-15: MindDoctor's Database diagram

#### 4.3.2.2 Data Dictionary

##### 4.3.2.2.1 *Entities*

No	Entity Name	Description	Occurrence
1.	User	The registered user of MindDoctor system.	Have 1 Province Link 1 User to 1 AdvisorInformation Contain many PaymentHistory, InsertedGold, Schedule Have many Comment, Like, Notification, Appointment, Conversation
2.	Country	All countries in the world	Have many Province
3.	Province	All provinces belong to country.	Belong to Country
4.	AdvisorInformation	Information of advisors.	Link 1 User to 1 AdvisorInformation Belong to Domain Have many SubDomain Have many CensorshipCouncilPost
5.	Domain	All domain of MindDoctor system.	Have many SubDomain, RegisteredAdvisor Contain many AdvisorInformation
6.	SubDomain	All sub-domains belong to domain.	Belong to Domain Have many RgtAdvSubDomain, Post, Question
7.	AdvisorSubDomain	Junction table of AdvisorInformation and SubDomain	Link 1 AdvisorInformation to 1 SubDomain
8.	RegisteredAdvisor	All registered user for Advisor position.	Belong to Domain Have many SubDomain
9.	RgtAdvSubDomain	Junction table of RegisteredAdvisor and SubDomain.	Link 1 RegisteredAdvisor to 1 SubDomain
10.	Post	All posts of MindDoctor system.	Belong to SubDomain Have many CensorshipCouncilPost Contain many Like,

			Comment
11.	Comment	All comments of post.	Belong to User, Post
12.	Like	All likes of post.	Belong to User, Post
13.	CensorshipCouncilPost	All advisor to assess the post can be accepted.	Belong to Post, AdvisorInfomation
14.	Question	All questions are exam for user who wants to become advisor.	Belong to SubDomain
15.	InsertedGold	All inserted money records of customer.	Belong to User
16.	PaymentHistory	Information of payment transaction between system and advisors.	Belong to User
17.	Schedule	Information of advisor's date free time.	Belong to User Have many Appointment
18.	Appointment	Appointment records between customer and advisor.	Belong to Schedule Contain many Notification Have 1 Conversation
19.	Notification	Notifications for customer and advisor who have an appointment in the future.	Belong to Appointment, User
20.	Conversation	Conversation between customer and advisor.	Belong to User Contain many Tip Have Conversation Report
21.	Tip	All tips which customer tips to advisor.	Belong to Conversation
22.	ConversationReport	Report advisor which is reported by customer.	Belong to Conversation
23.	MoneyTracking	Record all cash flow in or out system.	

#### 4.3.2.2.2 Attributes

Countries		
Field Name	Description	Data Type
CountryId	Unique identifier.	int
CountryName	Country name.	nvarchar(256)

Provinces		
Field Name	Description	Data Type
ProvinceId	Unique identifier.	int
CountryId	Reference to Country belongs to.	int
ProvinceName	Province name.	nvarchar(256)

Domains		
Field Name	Description	Data Type
DomainId	Unique identifier.	int
DomainName	Domain name.	nvarchar(256)
Image	Image of domain.	nvarchar(MAX)
Description	Additional Information for domain.	nvarchar(MAX)
CreatedDate	The date that created domain.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that domain was modified.	datetime2(7)

SubDomains		
Field Name	Description	Data Type
SubDomainId	Unique identifier.	int
DomainId	Reference to Domain that is parent of sub-domain.	nvarchar(256)
SubDomainName	Sub-domain name.	nvarchar(MAX)
CreatedDate	The date that create sub-domain.	datetime2(7)
Description	Additional Information for sub-domain.	nvarchar(MAX)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that sub-domain was modified.	datetime2(7)

Posts		
Field Name	Description	Data Type
PostId	Unique identifier.	int
StatusId	Post status is accepted, denied or pending.	int
SubDomainId	Reference to SubDomains that Post content is in that SubDomain.	int

DomainName	Domain name (Serve for Full-text search).	nvarchar(256)
SubDomainName	Sub-domain name (Serve for Full-text search).	nvarchar(256)
Image	Image is related to the Post.	nvarchar(MAX)
Title	Title of the Post.	nvarchar(MAX)
Introduction	A brief introduction introduces the post.	nvarchar(MAX)
TextContent	Detailed content of the post in the text form.	nvarchar(MAX)
HTMLContent	Detailed content of the post in the html form.	nvarchar(MAX)
CreatedDate	The date that create post.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that Post was modified.	datetime2(7)

RegisteredAdvisors		
Field Name	Description	Data Type
RegisteredId	Unique identifier.	int
StatusId	This field indicates that user has finished or cancel the exam, accepted or denied to advisor.	int
DomainId	Reference to Domains that Advisor register.	int
Email	Email of user.	nvarchar(256)
FullName	Full name of user.	nvarchar(512)
GenderId	Gender.	int
DoB	Date of birth.	date
PhoneNumber	Phone number.	nvarchar(256)
Token	Key to determine the advisor registration exam.	nvarchar(128)
ExamInfo	Detail information of user exam.	nvarchar(MAX)
CreatedDate	The date that registered advisor.	datetime2(7)
ModifiedDate	The date time that record was modified.	datetime2(7)

RgtAdvSubDomains		
Field Name	Description	Data Type
Id	Unique identifier.	int
RegisteredId	Reference to RegisteredAdvisors that register advisor.	int
SubDomainId	Reference to SubDomains that Advisor register.	int

AdvisorSubDomains		
Field Name	Description	Data Type
Id	Unique identifier.	int
UserName	User name of advisor.	nvarchar(256)
SubDomainId	Reference to SubDomains that Advisor is belong to that SubDomain.	int

ConversationReports		
Field Name	Description	Data Type
ReportId	Unique identifier	int
StatusId	Report status was solved or pending.	int
ConversationId	Reference to AdviceConversation that blocked that conversation.	int
IsRefund	Refund all gold in conversation for customer or not.	bit
ReportContent	Detailed content for conversation report.	nvarchar(MAX)
SolveDescription	This report has solved or pending.	nvarchar(MAX)
Image	Image related.	nvarchar(MAX)
CreatedDate	The date that report was created.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that record was modified.	datetime2(7)

Comments		
Field Name	Description	Data Type
CommentId	Unique identifier.	int
PostId	Reference to Posts which user likes this post.	int
UserName	User name.	nvarchar(256)
Content	Content of comment.	nvarchar(MAX)
CreatedDate	The date comment that post.	datetime2(7)

Questions		
Field Name	Description	Data Type
QuestionId	Unique identifier.	int
SubDomainId	Reference to SubDomains that question belongs to.	int
QuestionContent	Question content.	nvarchar(MAX)
Seconds	Time to answer question.	nvarchar(256)
CreatedDate	The date created question.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that record was modified.	datetime2(7)

Likes		
Field Name	Description	Data Type
LikeId	Unique identifier.	int
PostId	Reference to Posts that user likes post.	int
UserName	User name.	nvarchar(256)
CreatedDate	The date like post.	datetime2(7)

User		
Field Name	Description	Data Type
UserName	Unique identifier.	nvarchar(256)
StatusId	Status for user is online or not.	int
FullName	Full name of user.	nvarchar(512)
DoB	Date of birth.	date
GenderId	Gender.	int
Avatar	Image of user.	nvarchar(256)
PhoneNumber	Phone number.	nvarchar(256)
ProvinceId	Reference to the Province that belong users address.	int
Gold	Current gold of user.	int
RoleName	Authorize the user.	nvarchar(256)
CreatedDate	The date that user register account.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that this was modified.	datetime2(7)

AdvisorInformations		
Field Name	Description	Data Type
UserName	Username of advisor.	nvarchar(256)
DomainId	Advisor major.	int
Degree	Degree of advisor.	nvarchar(128)
GPC	Gold per conversation.	int
AvgRate	Average rate of advisor evaluated by customer.	float
About	Information of advisor.	nvarchar(MAX)
WorkPlace	Work place of advisor.	nvarchar(MAX)
WeekFreeTime	Default free time of advisor in week.	nvarchar(MAX)
BankName	Bank name.	nvarchar(512)
BankBranch	Branch of Bank.	nvarchar(512)
BankAccountNumber	Bank account number.	nvarchar(512)
BankAccountName	Bank account name.	nvarchar(512)
CreatedDate	The date that information was created.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that record was modified.	datetime2(7)

Tips		
Field Name	Description	Data Type
TipId	Unique identifier.	nvarchar(256)
ConversationId	Reference to Conversation that tip received come from that conversation.	nvarchar(512)
Gold	Gold received.	int
CreatedDate	The date that perform this created.	datetime2(7)

CensorshipCouncilPosts		
Field Name	Description	Data Type
Id	Unique identifier.	nvarchar(256)
PostId	Reference to Posts that post was created.	nvarchar(512)
UserName	The person who created this post.	nvarchar(256)
Accepted	This post was accepted or not.	bit
CreatedDate	The date that perform this action.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(512)
ModifiedDate	The date time that record was modified.	datetime2(7)

Conversations		
Field Name	Description	Data Type
ConversationId	Unique identifier.	int
Advisor	Username of advisor.	nvarchar(256)
Customer	Username of customer.	nvarchar(256)
Gold	Gold in conversation that customer has to pay.	int
ConversationContent	Detailed content of conversation.	nvarchar(MAX)
StartTime	Start time of conversation.	datetime2(7)
EndTime	End time of conversation.	datetime2(7)
Rate	Evaluate the conversation.	float
Feedback	Feedback of customer about conversation.	nvarchar(max)

Schedules		
Field Name	Description	Data Type
ScheduleId	Unique identifier.	int
Date	Schedule date.	date
Advisor	Username of advisor.	nvarchar(256)
FreeTime	Free time of advisor.	nvarchar(MAX)
CreatedDate	The date that schedule was created.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that record was modified.	datetime2(7)

Appointments		
Field Name	Description	Data Type
AppointmentId	Unique identifier.	int
StatusId	Status for user is online or not.	int
ScheduleId	Reference to the Schedule to choose date and advisor.	int
TimeId	Reference to the Notifications to set time to inform to user.	int
Advisor	Advisor participated in the conversation.	nvarchar(256)
Customer	Customer participated in the conversation.	nvarchar(256)
DepositGold	Deposit gold.	int
ConversationId	Reference to the Conversation.	int
SendMail	Send mail appointments time to user or not.	bit
CreatedDate	The date that appointment was created.	datetime2(7)
ModifiedBy	The latest person who modified this record.	nvarchar(256)
ModifiedDate	The date time that this was modified.	datetime2(7)

PaymentHistories		
Field Name	Description	Data Type
PaymentId	Unique identifier.	int
Admin	Who paid money.	nvarchar(256)
Advisor	Who receives money.	nvarchar(256)
PaymentContent	Content of the payment.	nvarchar(MAX)
Money	Value in money.	float
Gold	Gold value.	int
CreatedDate	The date that payment was happened.	datetime2(7)

Notifications		
Field Name	Description	Data Type
NotificationId	Unique identifier.	int
AppointmentId	Reference to the Appointments that inform the appointment time.	nvarchar(128)
StatusId	Status was notified or not yet.	int
UserName	User name.	nvarchar(256)
Title	Title of notification.	nvarchar(512)
Content	Content of notification.	nvarchar(512)
PushTime	Time to notify for user.	datetime2(7)
CreatedDate	The date that appointment was created.	datetime2(7)

MoneyTrackings		
Field Name	Description	Data Type
Id	Unique identifier.	int

Money	Value in money.	float
Gold	Gold value.	int
Content	Content of money tracking.	nvarchar(MAX)
CreatedBy	The person who created this record.	nvarchar(256)
CreatedDate	The date created this record.	datetime2(7)

InsertedGolds		
Field Name	Description	Data Type
InsertedId	Unique identifier.	int
Customer	Username of customer who insert card.	nvarchar(256)
Money	Money that inserted.	float
Gold	Gold value.	int
CreatedDate	The date that appointment was created.	datetime2(7)

### 4.3.3 Common Design

#### 4.3.3.1 Database – first approach with Entity Framework

Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. It **eliminates the need for most of the data-access code that developers usually need to write.**

The Database First Approach provides an alternative to the Code First and Model First approaches to the Entity Data Model. It creates model codes (classes, properties, DbContext, etc.) from the database in the project and those classes become the link between the database and controller.

The pros and cons of using Entity Framework with Database first:

- ✓ Development in agile team.
- ✓ Can use an existing database and create your tables and associations in there.
- ✓ Suitable for team without exceptionally good Database administrator.
- ✓ Data access code is independent and can be used for multiple type of Database management system.
- ✓ Better code management, and overall OOP process.
- ✓ Faster development pace in general.
- Extra overhead to learn if developers are not familiar with Entity Framework.
- Entity Framework will be slower than well-constructed SQL query. Overall, this is negligible.

#### 4.3.3.2 ASP.NET Identity System

ASP.NET Identity System is designed for authorizing and authenticating. ASP.NET Identity can be used with all of the ASP.NET Frameworks such as ASP.NET MVC, Web API... ASP.NET Identity can be used when building web, phone, store or hybrid applications.

The pros of using ASP.NET Identity System:

- ✓ By default, the ASP.NET Identity System stores all the user information in a database. ASP.NET Identity use Entity Framework Code First to implement all of its persistence mechanism.
- ✓ ASP.NET Identity makes the web application more unit testable. We can write unit tests for the parts of application that use ASP.NET Identity.
- ✓ ASP.NET Identity supports claims-based authentication, where the user's identity is represented as a set of claims.
- ✓ We can easily add social log-ins such as Microsoft Account, Facebook, Twitter, Google and others to application and store the user-specific data in our application.
- ✓ ASP.NET authentication is now based on OWIN middleware that can be used on any OWIN-based host. ASP.NET Identity does not have any dependency on System.Web. It is fully compliant OWIN framework and can be used in any OWIN hosted application.

## Implement ASP.NET Identity System:

1. Create an ASP.NET MVC application with Individual Accounts.

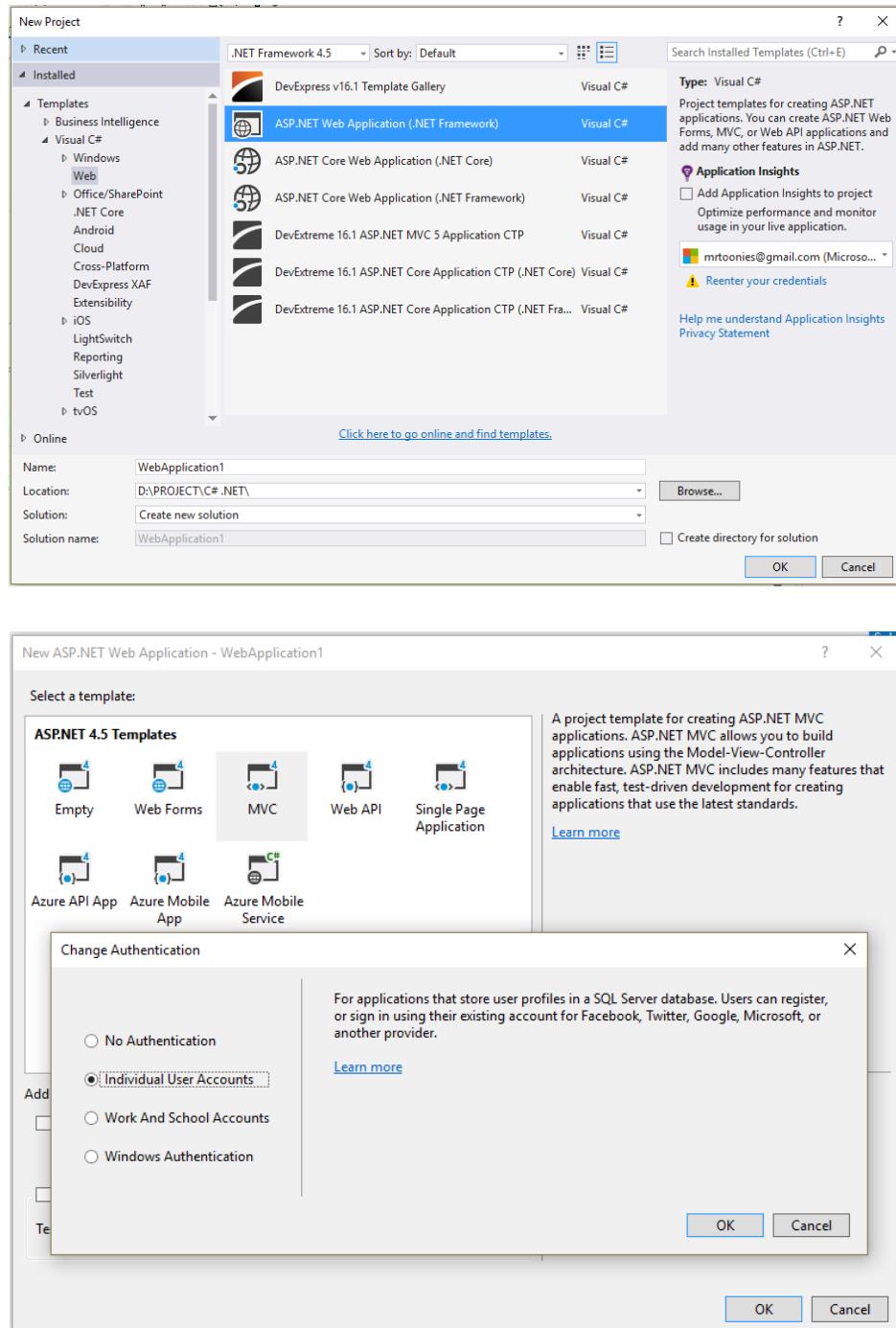


Figure 4-16: Implement ASP.NET Identity System

2. The created project contains the following three packages for ASP.NET Identity
  - a. **Microsoft.AspNet.Identity.EntityFramework**: this package has the Entity Framework implementation of ASP.NET Identity which will persist the ASP.NET Identity data and schema to SQL Server.
  - b. **Microsoft.AspNet.Identity.Core**: this package has the core interfaces for ASP.NET Identity. This package can be used to write an implementation for ASP.NET Identity that targets different persistence stores such as Azure Table Storage.

- c. **Microsoft.AspNet.Identity.OWIN**: this package contains functionality that is used to plug in OWIN authentication with ASP.NET Identity in ASP.NET applications. This is used when add login functionality to application and call into OWIN Cookie Authentication middleware to generate a cookie.

#### 4.3.3.3 Applied Full-text search from SQL Server

##### 4.3.3.3.1 Overview

Full-Text Search in SQL Server lets users and applications run full-text queries against character-based data in SQL Server tables. Before you can run full-text queries on a table, the database administrator must create a full-text index on the table. The full-text index includes one or more character-based columns in the table. Each full-text index indexes one or more columns from the table, and each column can use a specific language.

##### 4.3.3.3.2 Full-text search queries

After columns have been added to a full-text index, users and applications can run full-text queries on the text in the columns. These queries can search for any of the following:

- One or more specific words or phrases (*simple term*).
- A word or a phrase where the words begin with specified text (*prefix term*).
- Inflectional forms of a specific word (*generation term*).
- A word or phrase close to another word or phrase (*proximity term*).
- Synonymous forms of a specific word (*thesaurus*).
- Words or phrases using weighted values (*weighted term*).

Full-text queries are not case-sensitive. For example, searching for “MindDoctor” or “minddoctor” returns the same results.

Full-text queries use a small set of Transact-SQL predicates (CONTAINS and FREETEXT) and functions (CONTAINSTABLE and FREETEXTTABLE). However, the search goals of a given business scenario influence the structure of the full-text queries.

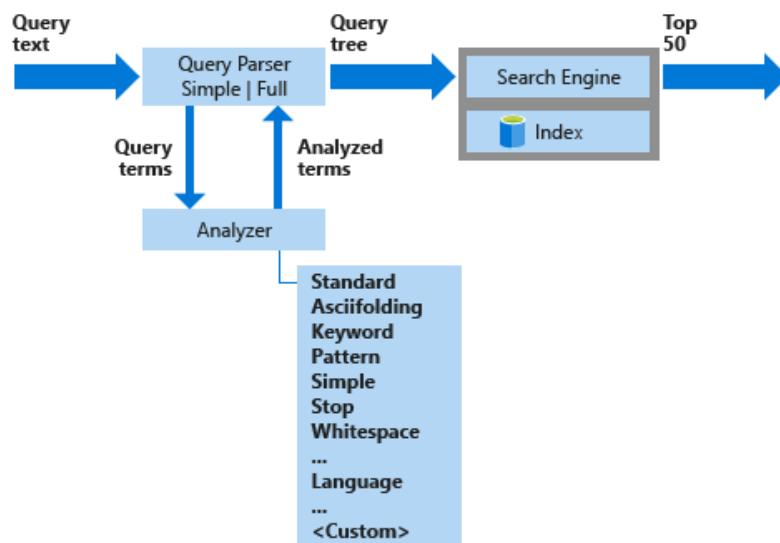


Figure 4-17: Full-text search queries

#### 4.3.3.3.3 Full-text indexing

Full text index is a type of inverted index that long string documents are indexed through “terms” and their respective location in documents which greatly improve the searching process. A simple illustration of full text index is provided below:

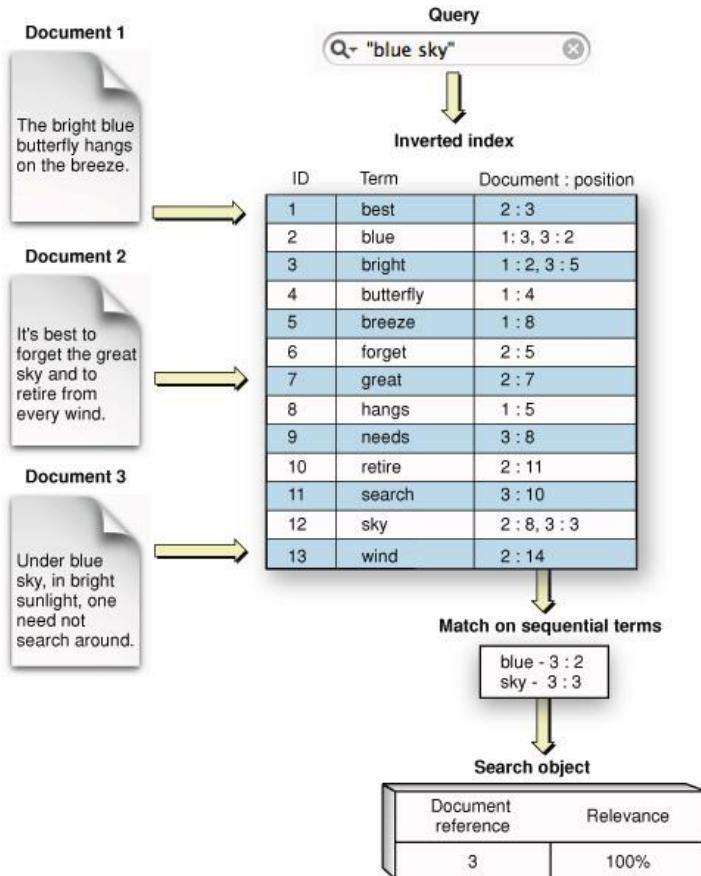


Figure 4-68: Full-text indexing

#### 4.3.3.4 Implementation

Pseudo Code:

```
Select (Post) Join FREETEXTTABLE (Post, (Post.Title, Post.Introduction,
Post.TextContent, Post.DomainName, Post.SubDomainName), text) ON Post.PostId =
FREETEXTTABLE.Key.
```

#### 4.3.3.4 Controller with API Service

Ajax (Asynchronous JavaScript and XML) is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, applications can send data to and retrieve from a server asynchronously without interfering with the display and behavior of the existing page.

MindDoctor built Controller API Service with Ajax for easier updating a part of website without reload. It helps increase user experiences.



Figure 4-79: Ajax

#### 4.3.3.5 Applied content based recommendation system

Recommendation system is a subclass of information filtering system that seeks to predict the “rating” or “preference” that a user would give to an item. Recommendation system are active information filtering systems which personalize the information coming to a user based on interests, relevance of the information etc.

From this perspective, MindDoctor applied recommendation system to our system to suggest relation posts which user may like.

MindDoctor applied content based recommendation system because our system is small and focus on individual. A based recommendation system works with data the user provides, either explicitly or implicitly. Based on data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendation, the engine becomes more and more accurate.

We collect user behaviors data to represent it on matrix

		Items					
		1	2	...	i	...	m
Users	1	5	3		1	2	
	2		2				4
	:			5			
	u	3	4	?	2	1	
	:					4	
	n			3	2		

Figure 4-20: Matrix between user and items

In matrix, we have boxes with number that represent the level of interested on item. Other, empty boxes represent that user has not approached item yet which is based on information that is known in the past by the user. The recommendation system will give the user information that user does not know but may like. That is predict the values in the empty boxes in the matrix then arrange in decreasing order to preference to suggest to the user.

	It1	It2	It3	It4	It5	It6	It7	It8	It9	...
U1	0	2	4							0
U2	1			3	5	2				
U3	2	3	3	2						
U4	2	1	2	1	3					
U5	2				4					
U6	2									
U7	1			2	2					
U8	3	3		5	1					
U9	3				1					
U10	2	1	3	2	1					
U11	1	1	3		3	1				
...										

	It1	It2	It3	It4	It5	It6	It7	It8	It9	...
U1	0	0	2	0	4	0	0	0	0	0
U2	1	0	0	0	0	3	0	5	2	
U3	0	2	3	0	3	2	0	0	0	
U4	0	2	1	0	2	1	0	3	0	
U5	0	0	2	0	0	0	4	0	0	
U6	0	0	2	0	0	0	0	0	0	
U7	1	0	0	0	0	2	0	2	0	
U8	0	3	0	3	0	0	5	0	1	
U9	0	0	3	0	0	0	0	0	1	
U10	2	1	3	0	2	0	1	0	0	
U11	0	1	0	1	0	3	0	3	1	
...										

Figure 4-21: Normalizing data

After normalizing data, we used concepts of Term Frequency and Inverse Document Frequency applied on our recommendation. This equation is applied for calculating MindDoctor's posts.

$$w_{t,d} = \begin{cases} 1 + \log tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Figure 4-22: Equation

We used Vector Space Model to compute the proximity based on the angle between the vector to determine which item user like more.

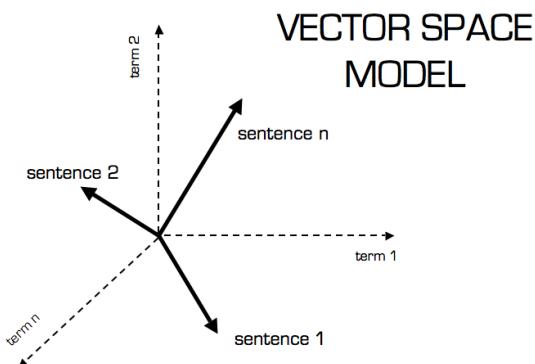


Figure 4-23: Vector space model.

Pseudo code:

```
// this array contains sub domain id
int[] conversations = GetRecentlyConversations();
int[] searchs = GetRecentlySearch();
int[] reads = GetRecentlyRead();
int[] likes = GetRecentlyLike();

// put it into matrix
int[][] = CreateMatrix(int[] conversations, int[] searchs, int[] reads, int[] likes);

// normalizing matrix
int[][] = NormalizingMatrix(int[][] matrix)

// calculate grade of each sub domain id
// put it into new matrix with sub domain id and this grade
// applied vector space model, because the axis is coincidence with sub domain id, so that we must find
// the max grade of matrix to find the most interested of user.
```

<https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/>

#### 4.3.4 Detail Design

##### 4.3.4.1 Register to become Customer

###### Screen Design



Figure 4-24: Register to become Customer screen design

###### Class Diagram

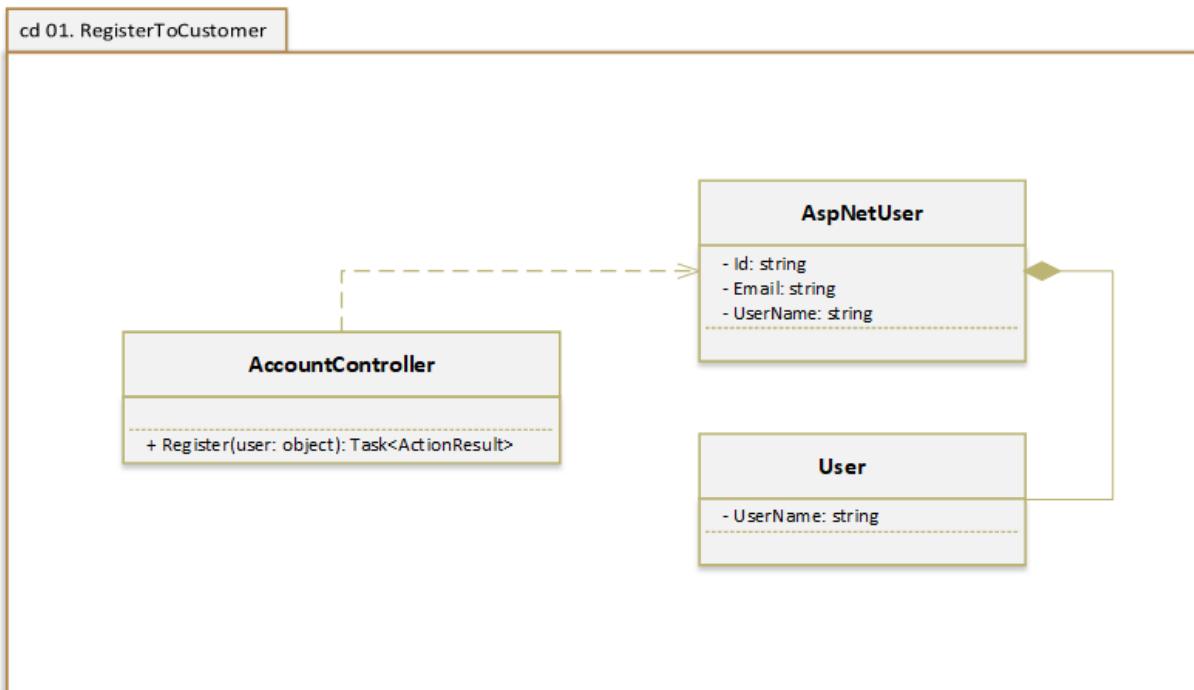


Figure 4-25: Register to become Customer class diagram

## Class Specification

AccountController			
Physical address	MindDoctor / Controllers / AccountController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Register			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	User	object	Object contains user name and password.

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	
2.	Email	string	
3.	UserName	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
Operation			

## Sequence Diagram

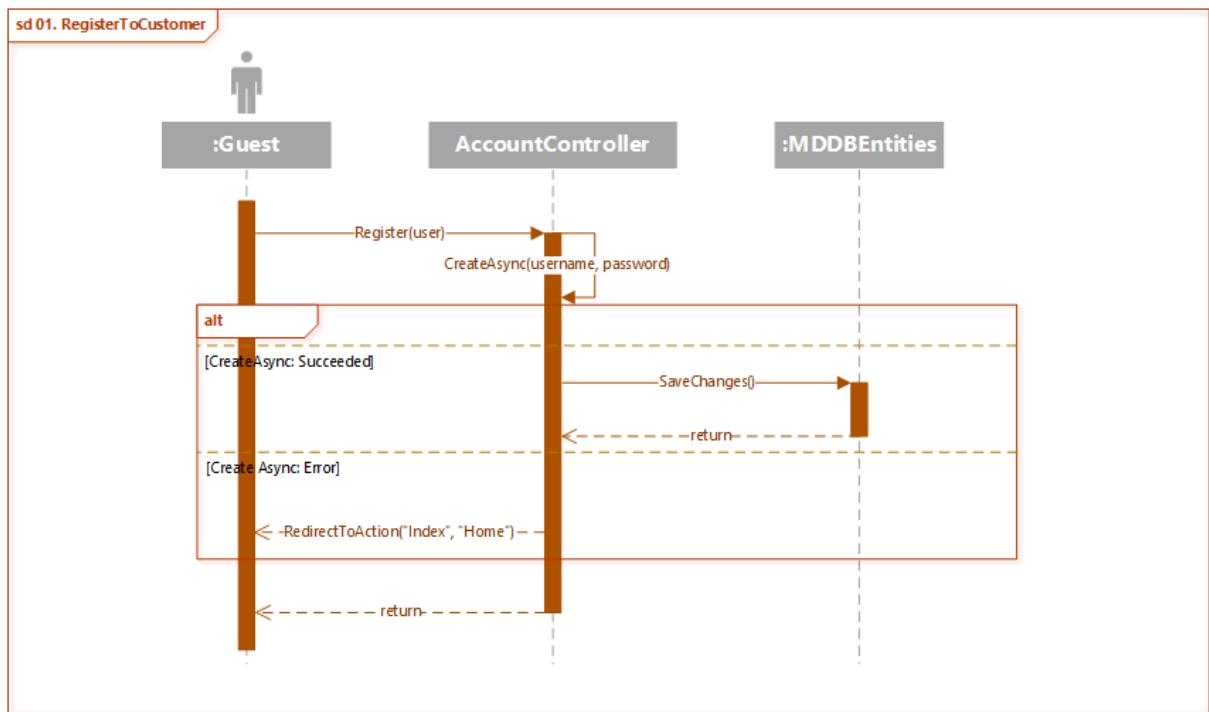


Figure 4-26: Register to become Customer Sequence diagram

#### 4.3.4.2 Verify email address

##### Screen Design



Figure 4-27: Verify email Screen design

##### Class Diagram

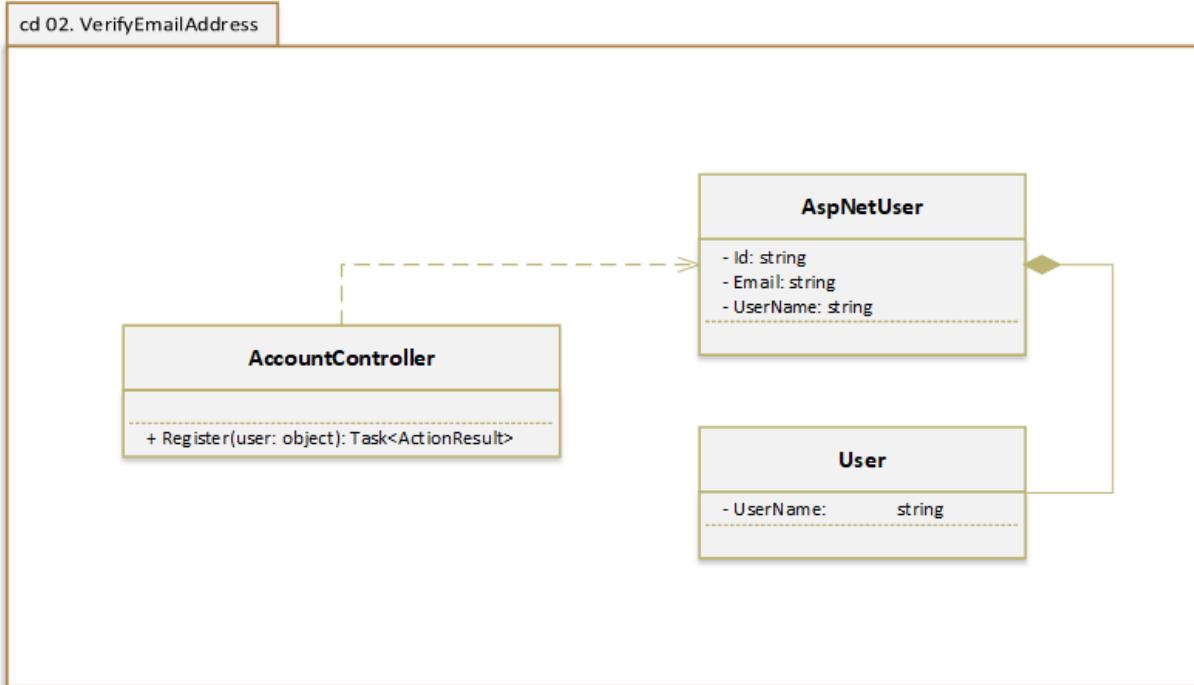


Figure 4-28: Verify email Class diagram

##### Class Specification

AccountController			
Physical address	MindDoctor / Controllers / AccountController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Register			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	User	object	Object contains user

			name and password.
--	--	--	--------------------

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	
2.	Email	string	
3.	UserName	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
Operation			

## Sequence Diagram

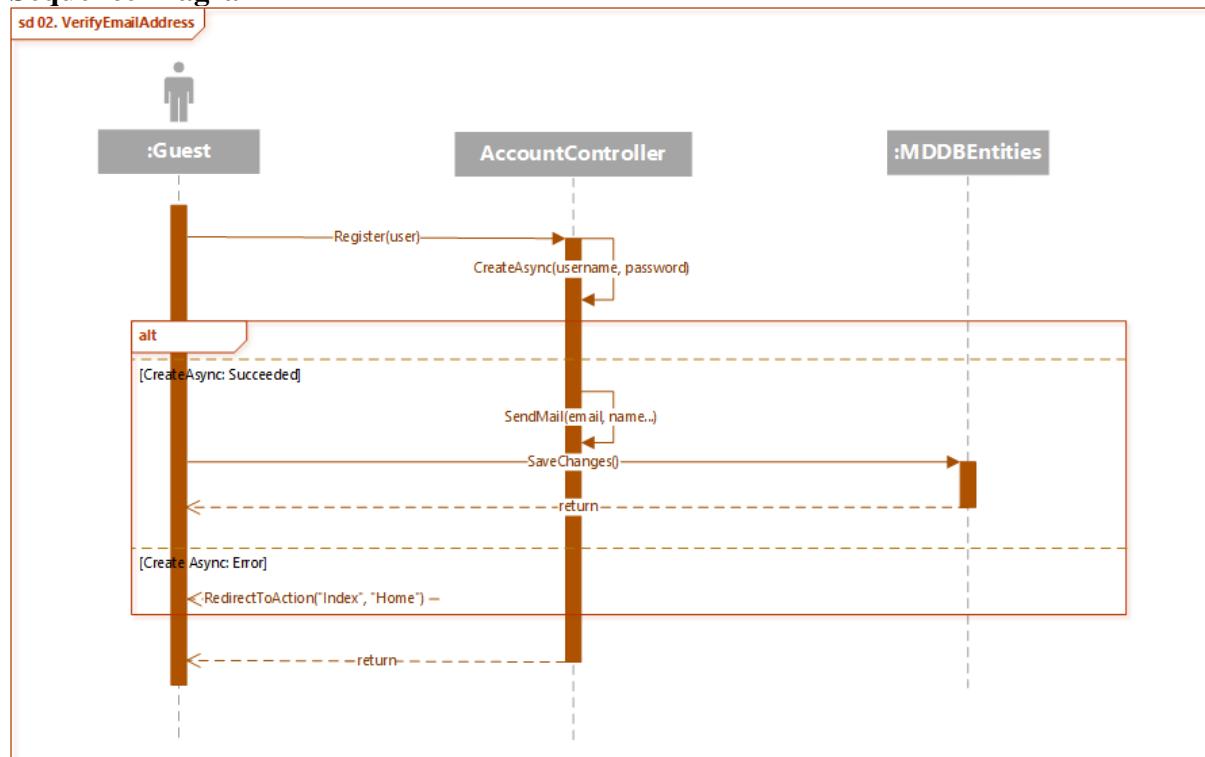


Figure 4-89: Verify email Sequence diagram

#### 4.3.4.3 Register to become Advisor

##### Screen Design

**THÔNG TIN CÁ NHÂN**

---

Họ và tên:	Tạ Quang Huy	Lĩnh vực tư vấn:	Tâm lý
Email:	huytqse03875@fpt.edu.vn	Chuyên mục tư vấn:	<input checked="" type="checkbox"/> Tâm lý học đường <input type="checkbox"/> Hôn nhân gia đình <input type="checkbox"/> Tuổi dậy thi
Ngày sinh:	10/05/1995		
Giới tính:	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ		
Số điện thoại:	0166 360 5555		
<input type="button" value="Đăng ký"/>			

Figure 4-90: Register to become Advisor Screen design

##### Class Diagram

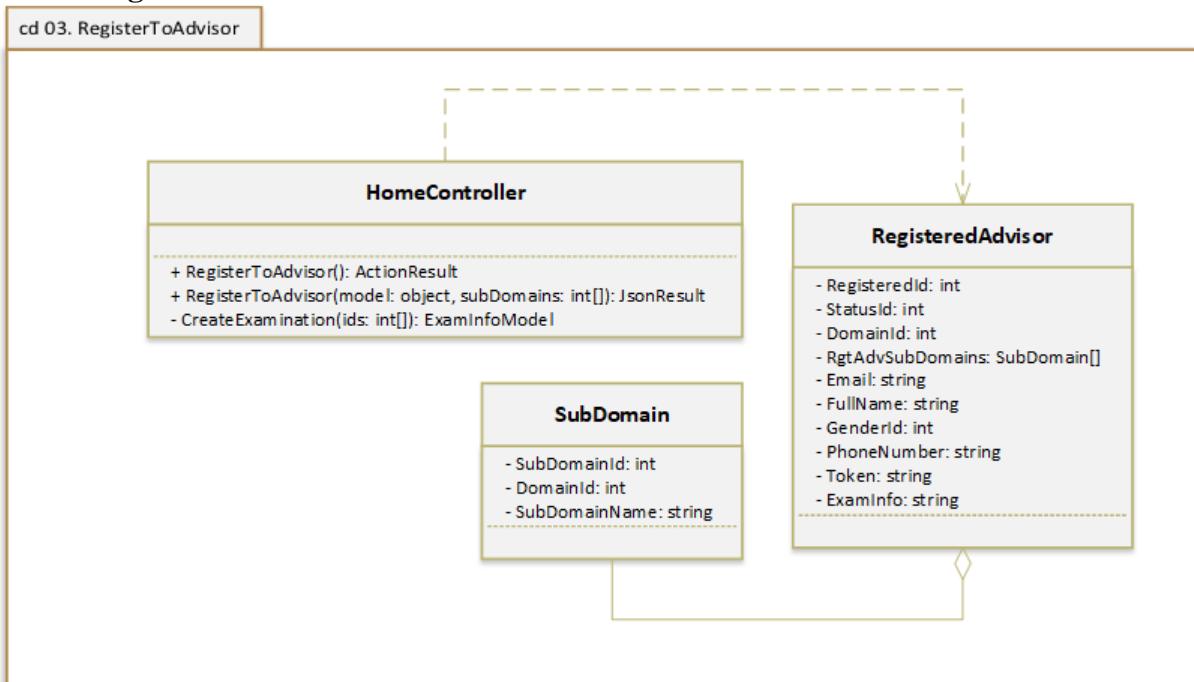


Figure 4-31: Register to become Advisor Class diagram

##### Class Specification

HomeController			
Physical address	MindDoctor / Controllers / HomeController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
<b>RegisterToAdvisor</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
<b>RegisterToAdvisor</b>			

Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Object contains registered information of user.
	subDomains	int[]	List sub-domain id of registered information of user.
<b>CreateExamination</b>			
Return Type	ExamInfoModel		
Parameters	Name	Type	Description
	ids	int[]	List sub-domain id of registered information of user.

<b>RegisteredAdvisor</b>			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / RegisteredAdvisor.cs		
Base class			
<b>Attributes</b>			
No	Name	Type	Description
1.	RegisteredId	int	
2.	StatusId	int	
3.	DomainId	int	
4.	RgtAdvSubDomains	Collection	List of sub-domain is registered by user.
5.	Email	string	
6.	FullName	string	
7.	GenderId	int	
8.	PhoneNumber	string	
9.	Token	string	
10.	ExamInfo	string	JSON string to save the exam information of user.
<b>Operation</b>			

<b>SubDomain</b>			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / SubDomain.cs		
Base class			
<b>Attributes</b>			
No	Name	Type	Description
1.	SubDomainId	int	

2.	DomainId	int	
3.	SubDomainName	string	
Operation			

## Sequence Diagram

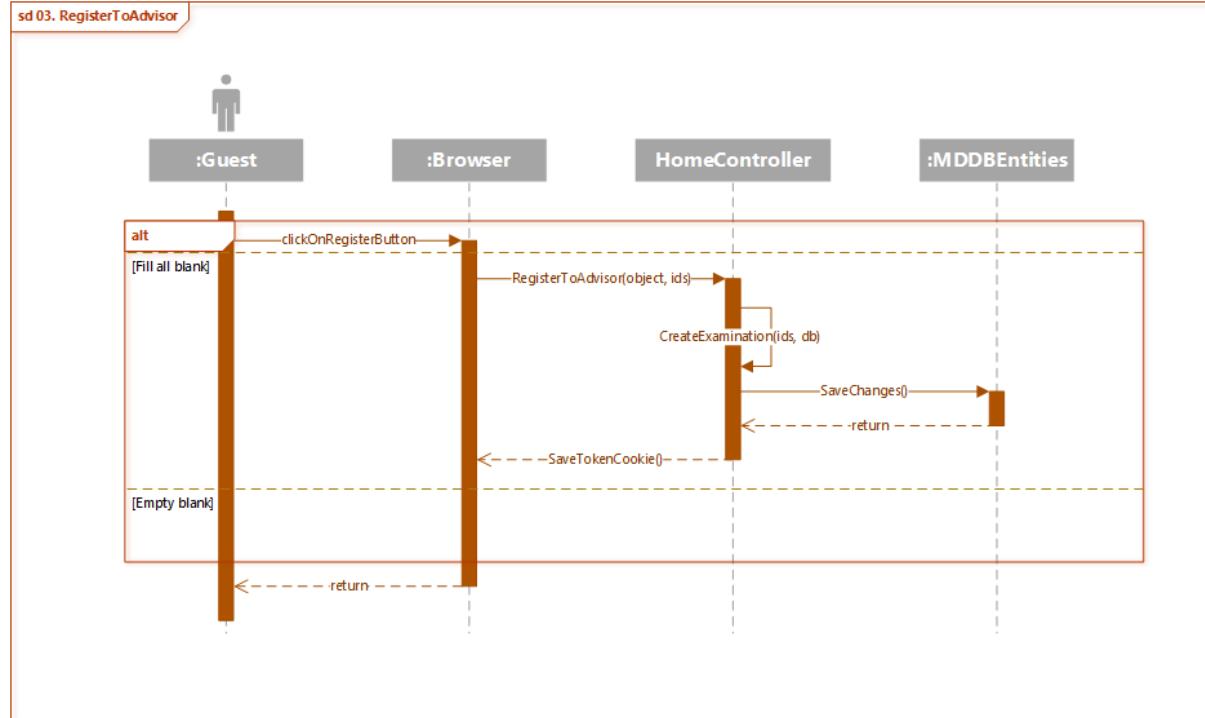


Figure 4-32: Register to become Advisor Sequence diagram

#### 4.3.4.4 Take qualification exam

##### Screen Design

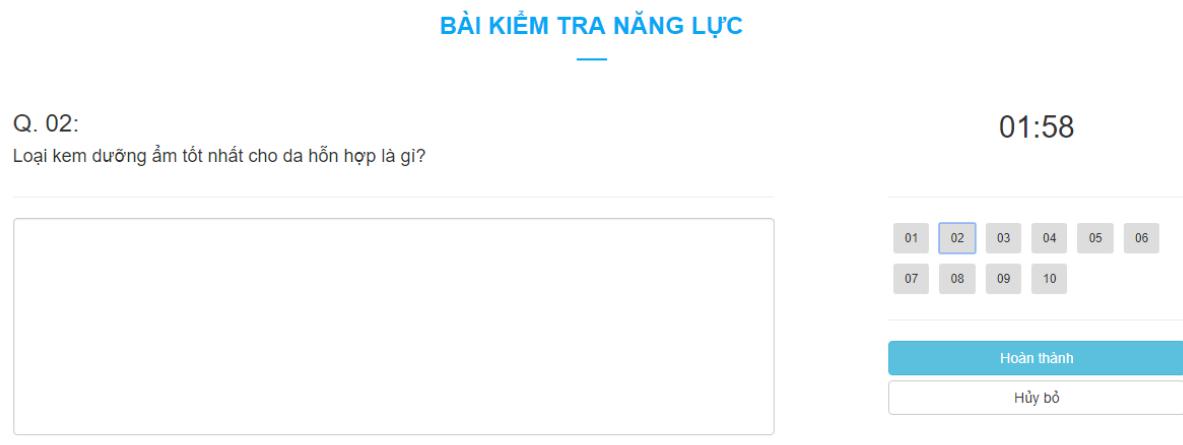


Figure 4-33: Take qualification exam Screen design

##### Class Diagram

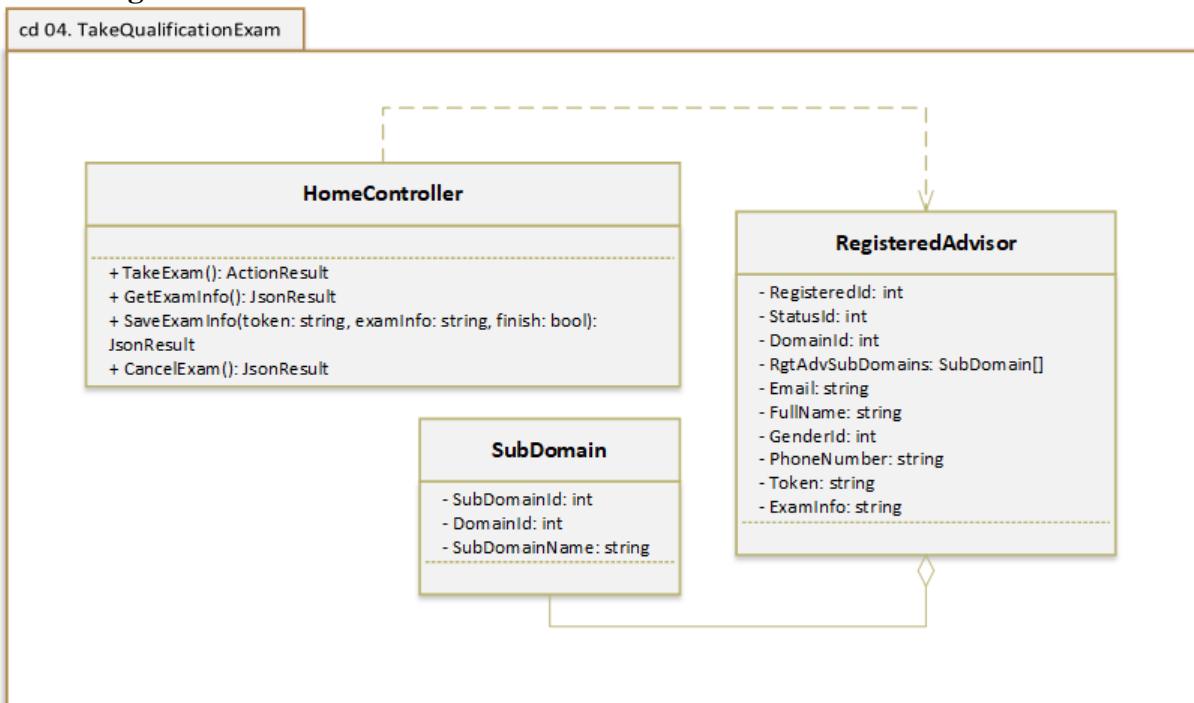


Figure 4-34: Take qualification exam Class diagram

##### Class Specification

HomeController			
Physical address	MindDoctor / Controllers / HomeController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
TakeExam			

Return Type	ActionResult		
Parameters	Name	Type	Description
<b>GetExamInfo</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
<b>SaveExamInfo</b>			
Return Type	ExamInfoModel		
Parameters	Name	Type	Description
	token	string	This is a token save into cookie on client can help user can reconnect to exam if disconnect or close browser.
	examInfo	string	JSON string to save the exam information of user.
	finish	bool	

<b>RegisteredAdvisor</b>			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / RegisteredAdvisor.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	RegisteredId	int	
2.	StatusId	int	
3.	DomainId	int	
4.	RgtAdvSubDomains	Collection	List of sub-domain is registered by user.
5.	Email	string	
6.	FullName	string	
7.	GenderId	int	
8.	PhoneNumber	string	
9.	Token	string	
10.	ExamInfo	string	JSON string to save the exam information of user.
Operation			

<b>SubDomain</b>	
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / SubDomain.cs

Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	

Operation			
-----------	--	--	--

## Sequence Diagram

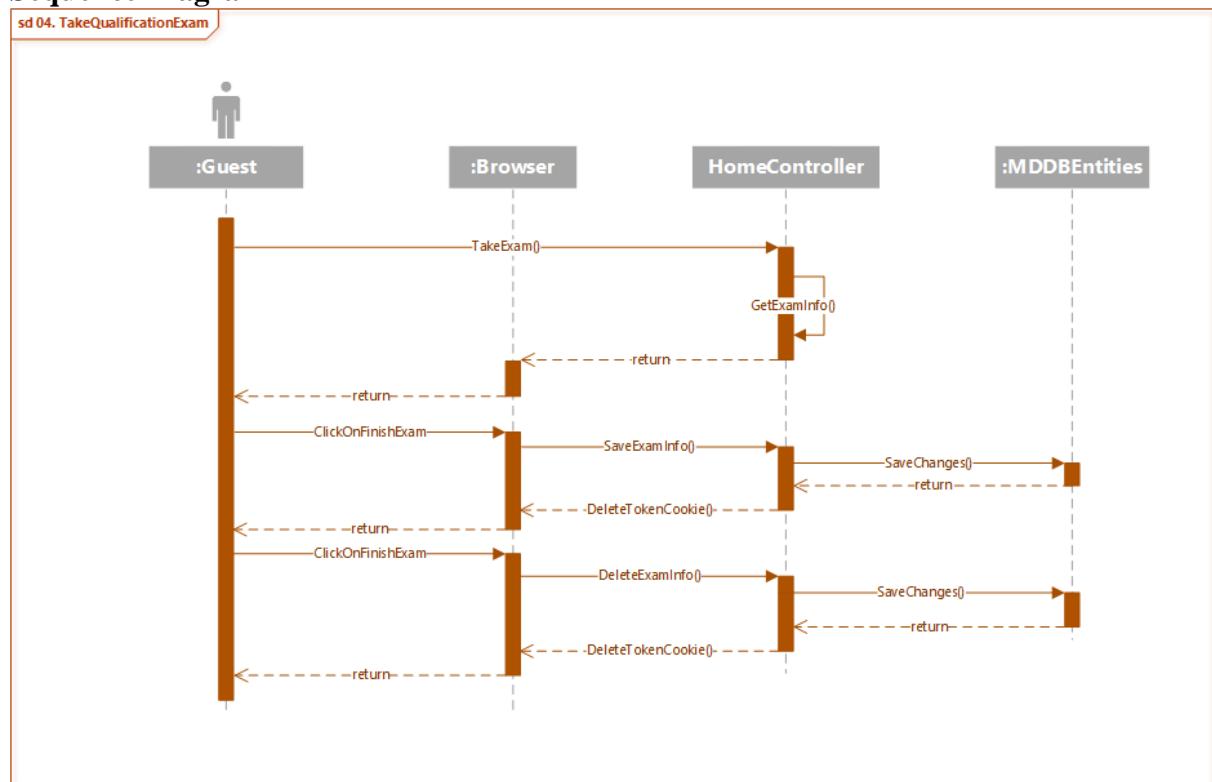


Figure 4-35: Take qualification exam Sequence diagram

#### 4.3.4.5 View post

##### Screen Design

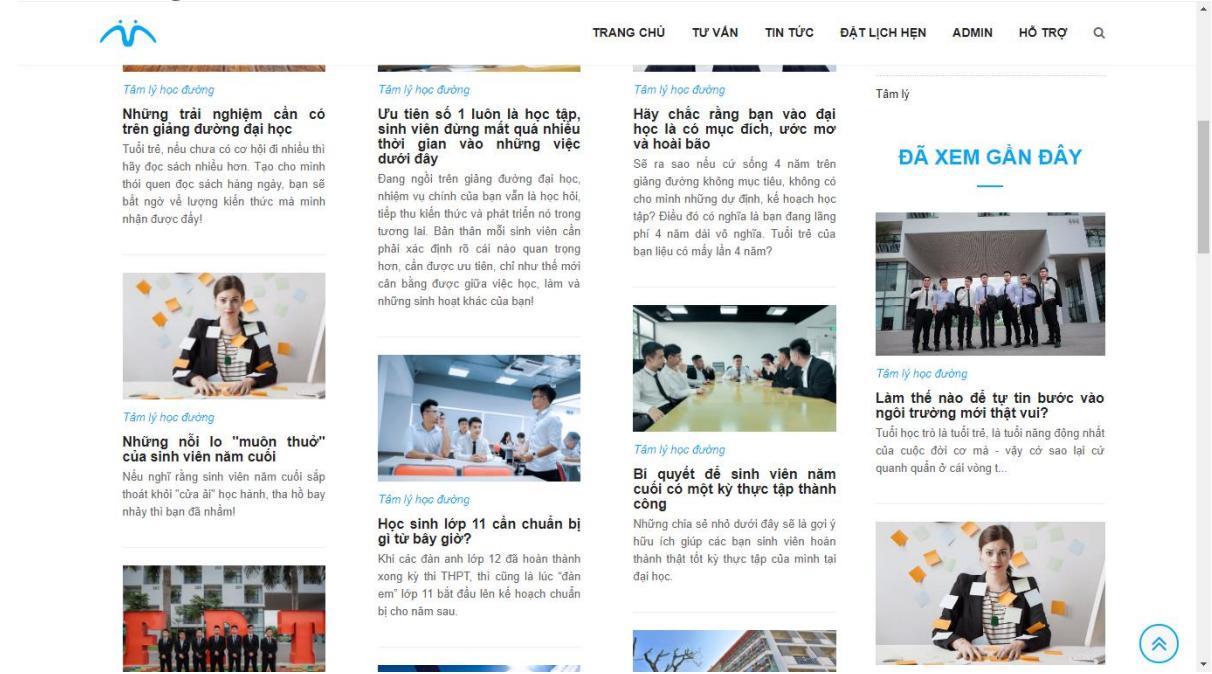


Figure 4-36: View post Screen design

##### Class Diagram

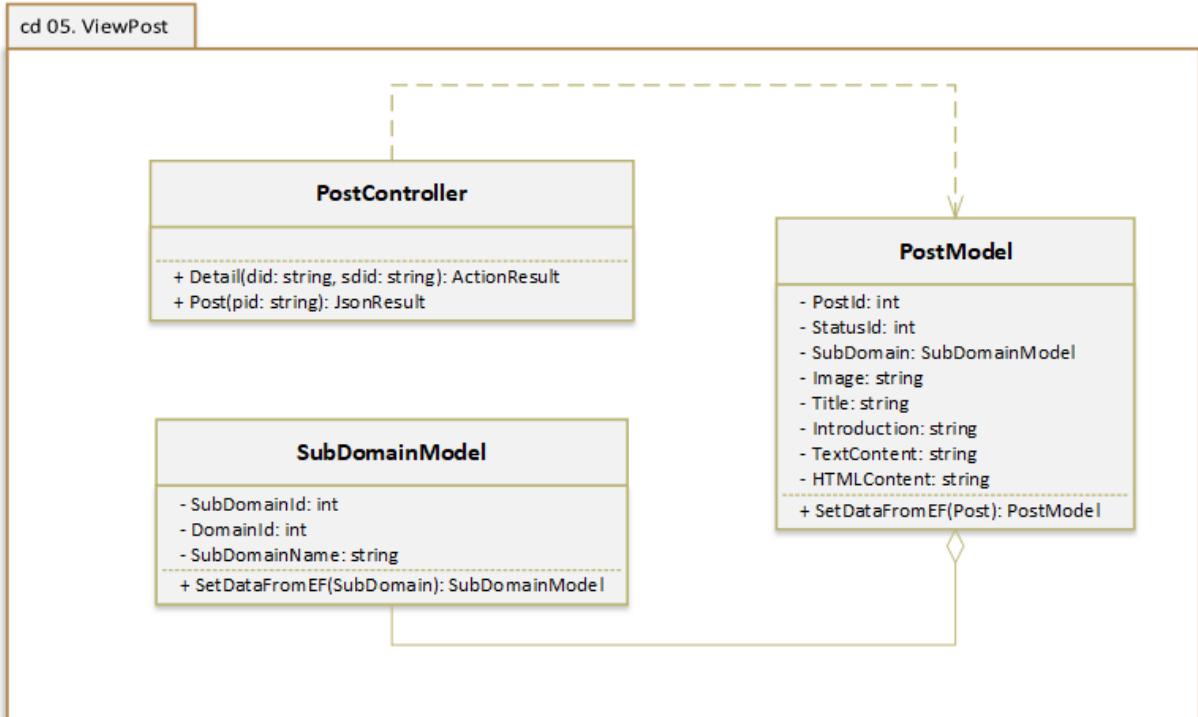


Figure 4-37: View post Class diagram

##### Class Specification

PostController	
Physical address	MindDoctor / Controllers / PostController.cs
Base class	BaseController

Attributes			
No	Name	Type	Description
Operation			
Detail			
Return Type	ActionResult		
Parameters	Name	Type	Description
1.	did	string	DomainId.
2.	sdid	string	Sub-domainId.
Post			
Return Type	ActionResult		
Parameters	Name	Type	Description
	postId	string	PostId.

PostModel			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Image	string	
5.	Title	string	
6.	Introduction	string	
7.	TextContent	string	
8.	HTMLContent	string	
Operation			
SetDataFromEF			
Return Type	PostModel		
Parameters	Name	Type	Description
	post	Post	

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			

SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	

## Sequence Diagram

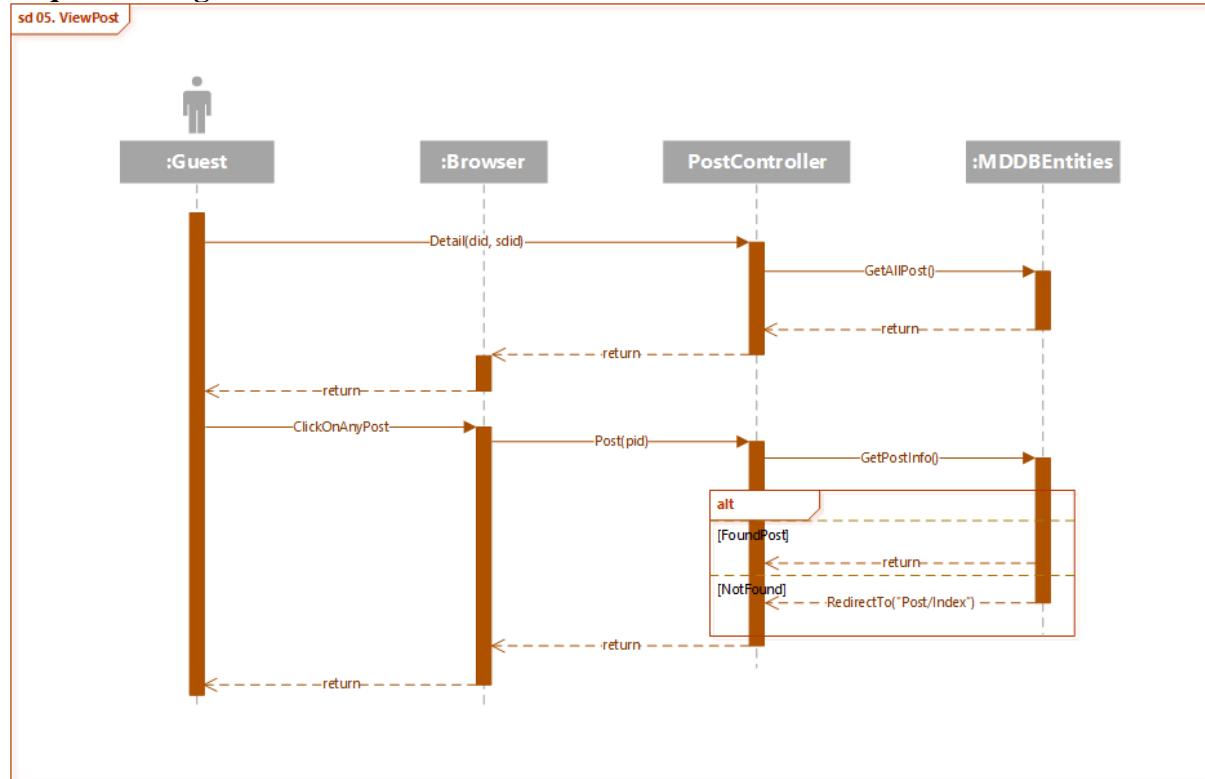


Figure 4-38: View post Sequence diagram

#### 4.3.4.6 Share post on Facebook

##### Screen Design

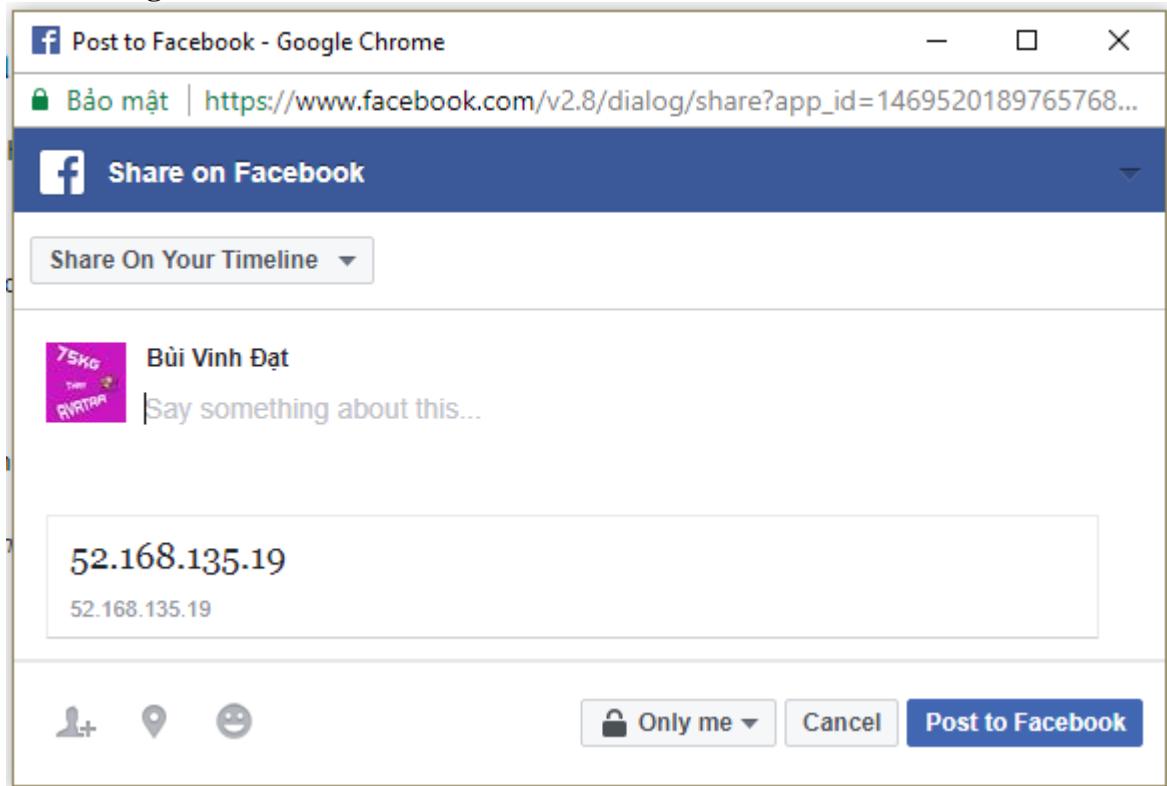


Figure 4-39: Share post on Facebook Screen design

##### Class Diagram

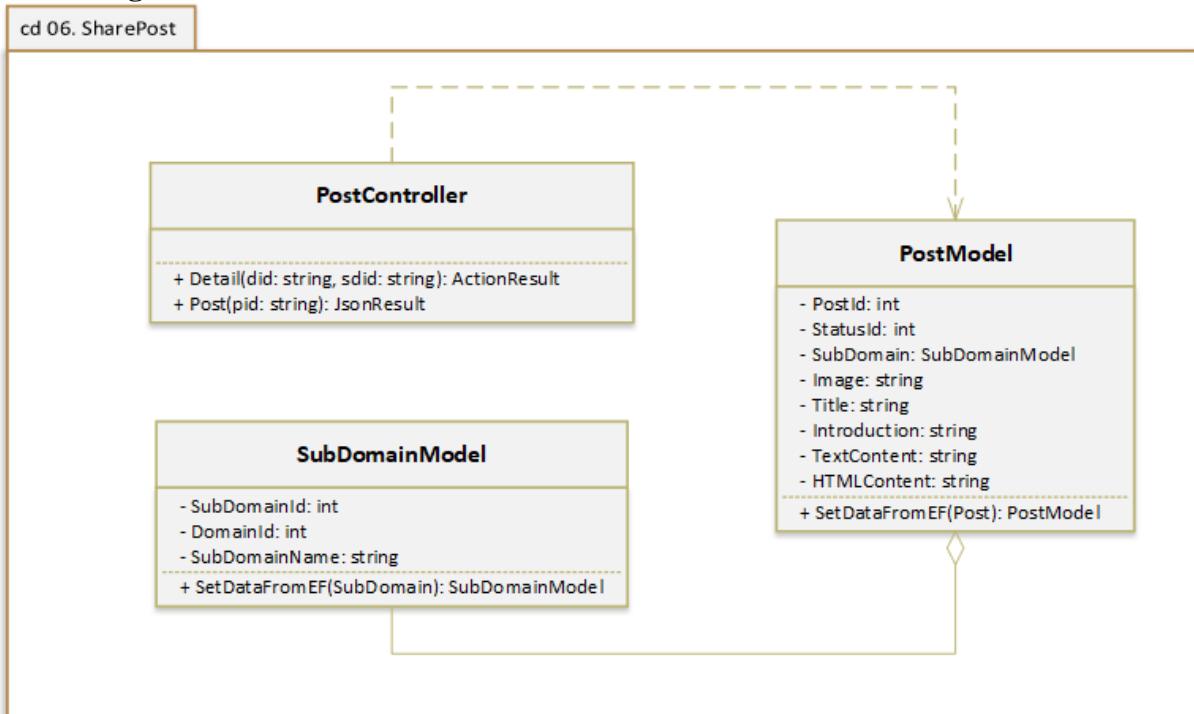


Figure 4-40: Share post on Facebook Class diagram

##### Class Specification

PostController

Physical address	MindDoctor / Controllers / PostController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Detail			
Return Type	ActionResult		
Parameters	Name	Type	Description
1.	did	string	DomainId.
2.	sdid	string	Sub-domainId.
Post			
Return Type	ActionResult		
Parameters	Name	Type	Description
	pid	string	PostId.

PostModel			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Image	string	
5.	Title	string	
6.	Introduction	string	
7.	TextContent	string	
8.	HTMLContent	string	
Operation			
SetDataFromEF			
Return Type	PostModel		
Parameters	Name	Type	Description
	post	Post	

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	

2.	DomainId	int	
3.	SubDomainName	string	
Operation			
SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	

## Sequence Diagram

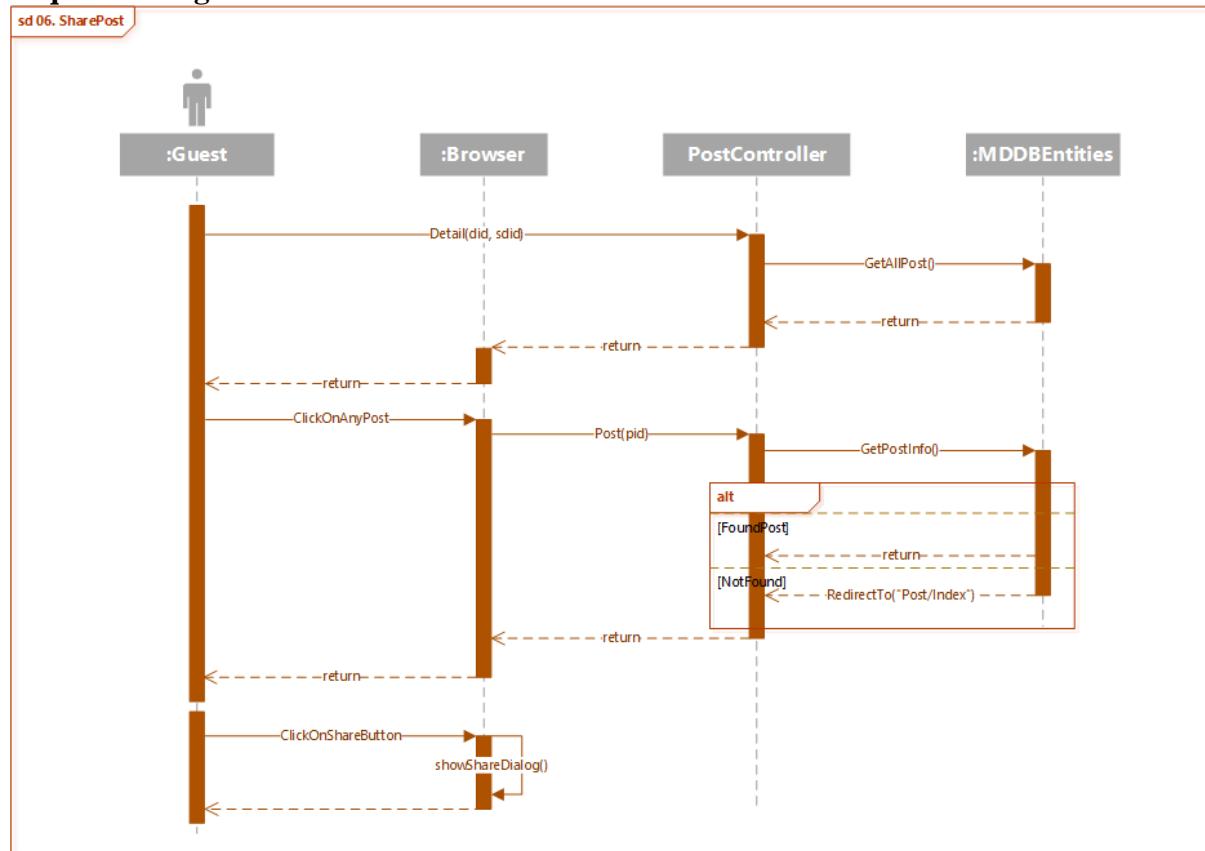


Figure 4-41: Share post on Facebook Sequence diagram

#### 4.3.4.7 Search post

##### Screen Design

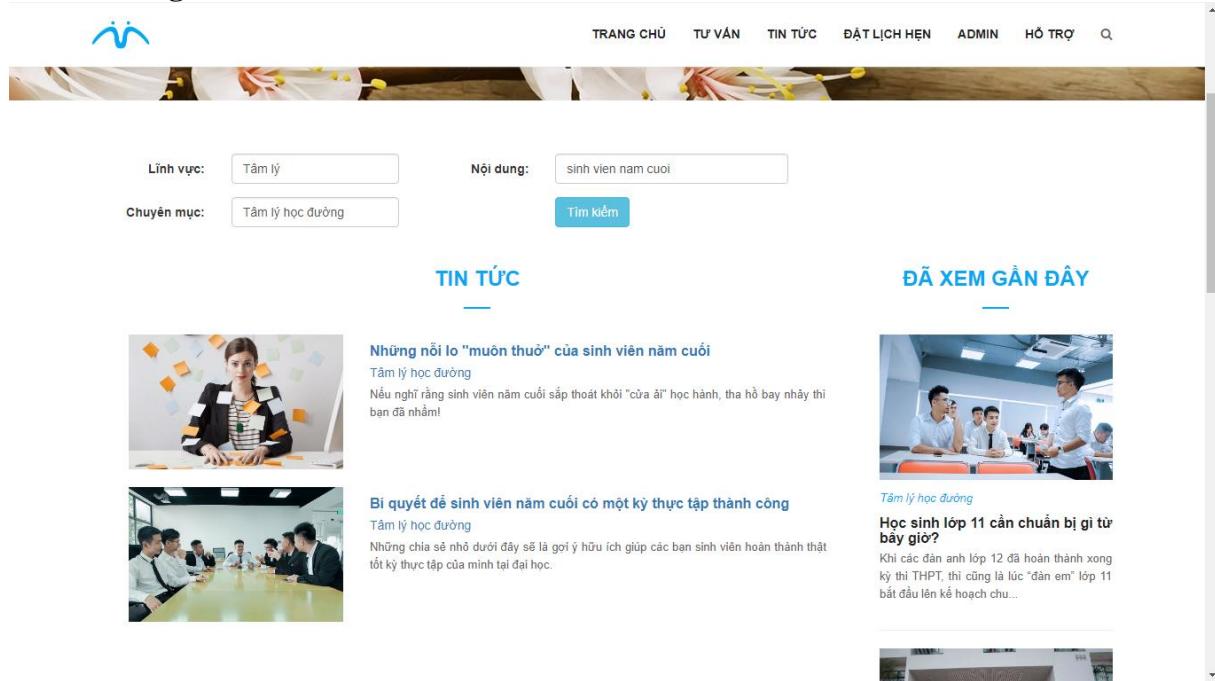


Figure 4-42: Search post Screen design

##### Class Diagram

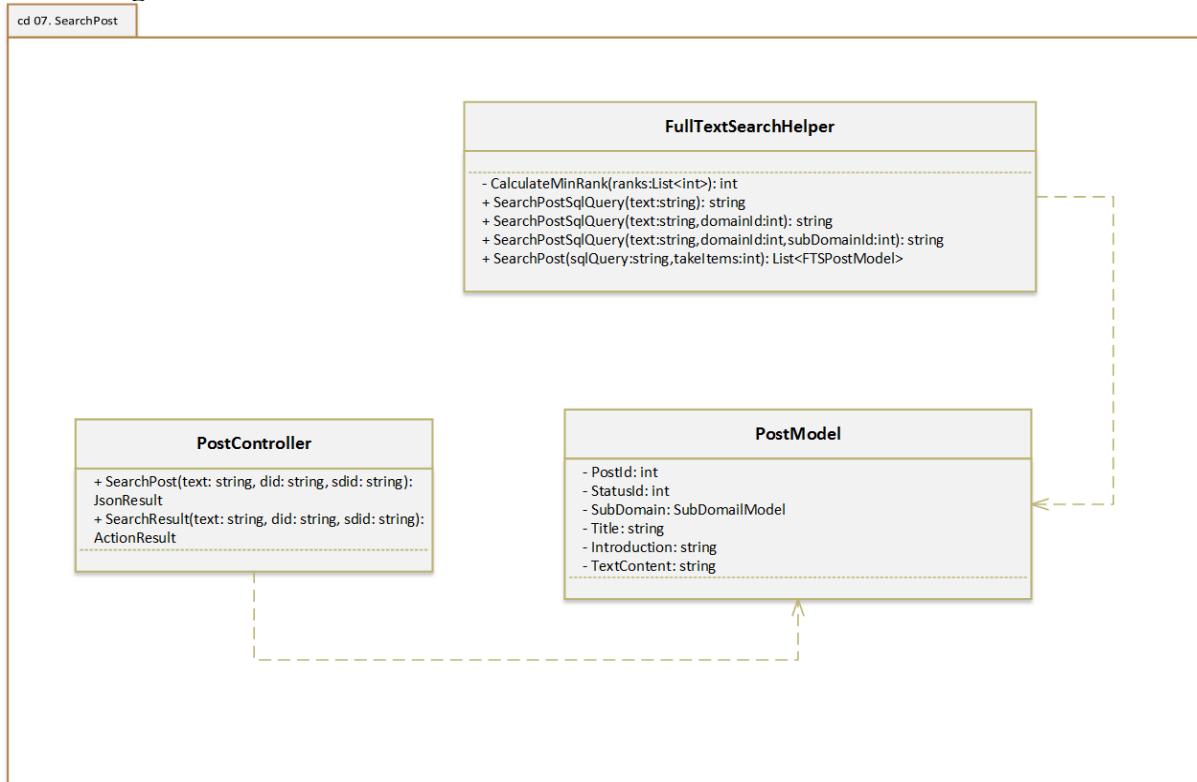


Figure 4-43: Search post Class diagram

##### Class Specification

**PostModel**

Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Title	string	
5.	Introduction	string	
6.	TextContent	string	
Operation			

PostController			
Physical address	MindDoctor / Controllers / PostController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
SearchPost			
Return Type	JsonResult		
Parameters	Name	Type	Description
	text	string	Text from search bar.
	did	string	DomainId.
	sdid	string	SubDomainId.
SearchResult			
Return Type	ActionResult		
Parameters	Name	Type	Description
	text	string	Text from search bar.
	did	string	DomainId.
	sdid	string	SubDomainId.

FullTextSearchHelper			
Physical address	MindDoctor / Common / FullTextSearchHelper.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
CaculateMinRank			
Return Type	int		
Parameters	Name	Type	Description

	ranks	List<int>	Return a ranking list from search results.
<b>SearchPostSqlQuery</b>			
Return Type	string		
Parameters	Name	Type	Description
	text	string	Text from search bar.
<b>SearchPostSqlQuery</b>			
Return Type	string		
Parameters	Name	Type	Description
	text	string	Text from search bar.
	domainId	int	DomainId.
<b>SearchPostSqlQuery</b>			
Return Type	string		
Parameters	Name	Type	Description
	text	string	Text from search bar.
	domainId	int	DomainId.
	subDomainId	int	SubDomainId.
<b>SearchPost</b>			
Return Type	List<FTSPostModel>		
Parameters	Name	Type	Description
	sqlQuery	string	Query string to search post.
	takeItems	int	Number of items that SearchPost method will return.

## Sequence Diagram

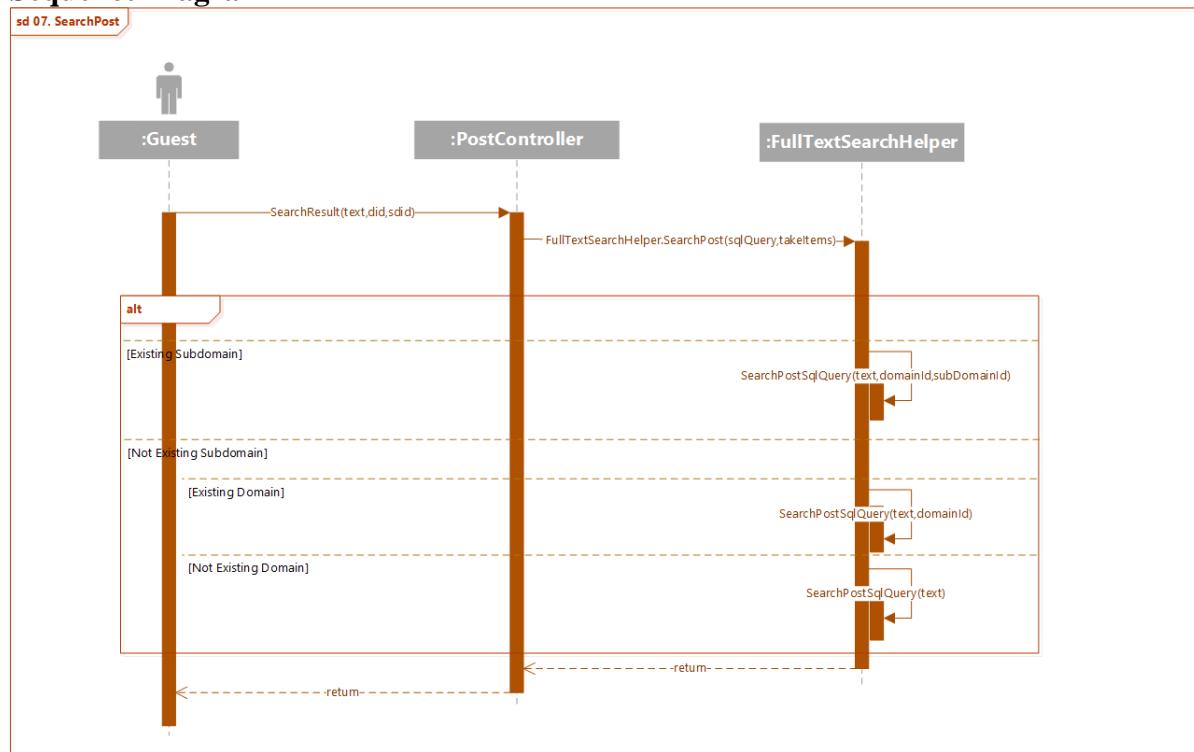


Figure 4-44: Search post Sequence diagram

## Implementaion

Pseudo Code: SearchPostSqlQuery(text)

1. TB1 = **Select (Post) Join**  
**FREETEXTTABLE**(Post,(Post.Title,Post.Introduction,Post.TextContent, Post.DomainName, Post.SubDomainName),text) **ON** Post.PostId =  
**FREETEXTTABLE**.Key
2. TB2 = **Select (SubDomain)**
3. **Select TB1 Join TB2 ON** TB1.SubDomainId = TB2.SubDomainId **WHERE** TB1.Rank > 0 **AND** TB1.StatusId = 1

Pseudo Code: SearchPostSqlQuery(text, domainId)

1. TB1 = **Select (Post) Join**  
**FREETEXTTABLE**(Post,(Post.Title,Post.Introduction,Post.TextContent, Post.DomainName, Post.SubDomainName),text) **ON** Post.PostId =  
**FREETEXTTABLE**.Key
2. TB2 = **Select (SubDomain)**
3. **Select TB1 Join TB2 ON** TB1.SubDomainId = TB2.SubDomainId **WHERE** TB2.DomainId = domainId **AND** TB1.Rank > 0 **AND** TB1.StatusId = 1

Pseudo Code: SearchPostSqlQuery(text, domainId, subDomainId)

- if subDomainId = 0
- return SearchPostSqlQuery(text, domainId)
- else
1. TB1 = **Select (Post) Join**  
**FREETEXTTABLE**(Post,(Post.Title,Post.Introduction,Post.TextContent, Post.DomainName, Post.SubDomainName),text) **ON** Post.PostId =  
**FREETEXTTABLE**.Key
  2. TB2 = **Select (SubDomain)**
  3. **Select TB1 Join TB2 ON** TB1.SubDomainId = TB2.SubDomainId **WHERE** TB2.SubDomainId = subDomainId **AND** TB1.Rank > 0 **AND** TB1.StatusId = 1

### 4.3.4.8 View Advisors's schedule

## Screen Design



Figure 4-45: View Advisors's schedule Screen design

## Class Diagram

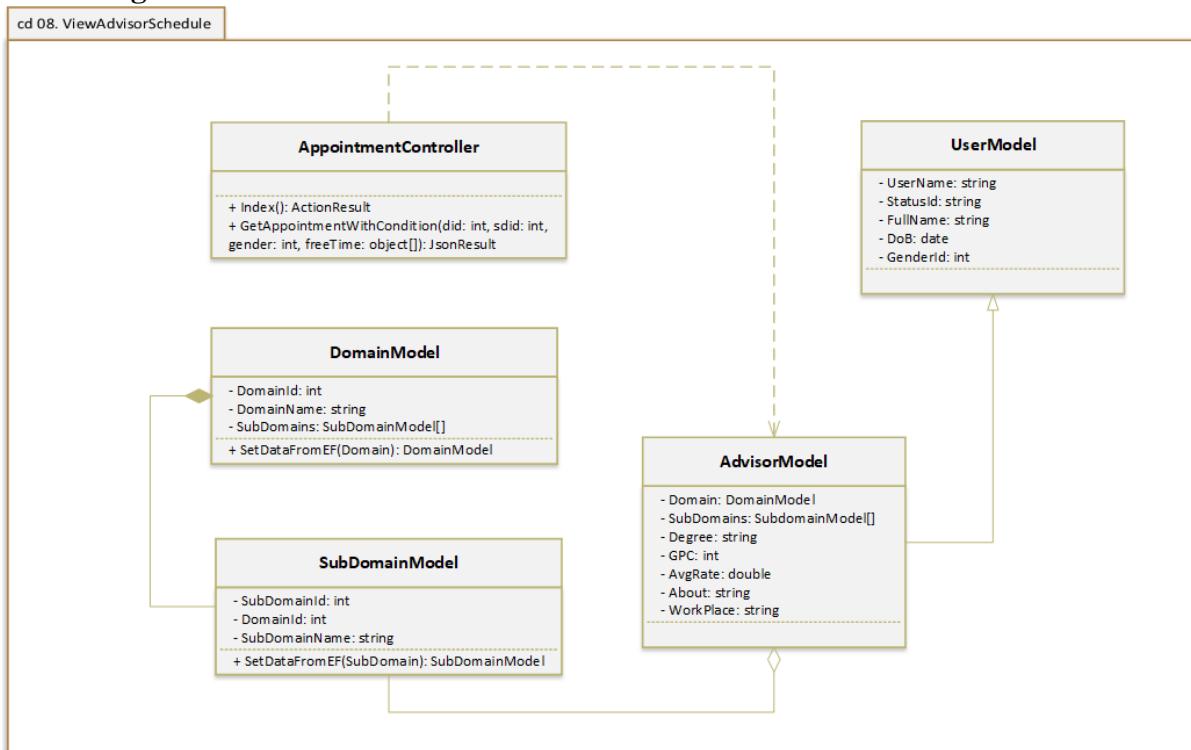


Figure 4-46: View Advisors's schedule Class diagram

## Class Specification

AppointmentController			
Physical address	MindDoctor / Controllers / AppointmentController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		

Parameters	Name	Type	Description
<b>GetAppointmentWithCondition</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
	did	string	DomainId.
	sdid	string	SubDomainId.
	gender	int	
	freeTime	object[]	Object array from Firebase which contains all advisors are ready on this day.

<b>UserModel</b>			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
<b>Attributes</b>			
No	Name	Type	Description
1.	Domain	DomainModel	
2.	SubDomains	SubDomainModel[]	
3.	Degree	string	
4.	GPC	int	
5.	AvgRate	double	
6.	About	string	
7.	Workplace	string	
<b>Operation</b>			

<b>AdvisorModel</b>			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
<b>Attributes</b>			
No	Name	Type	Description
1.	Domain	DomainModel	
2.	SubDomains	SubDomainModel[]	
3.	Degree	string	
4.	GPC	int	
5.	AvgRate	double	
6.	About	string	
7.	Workplace	string	
<b>Operation</b>			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			
SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	
DomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
3.	SubDomains	SubDomainModel[]	
Operation			
SetDataFromEF			
Return Type	DomainModel		
Parameters	Name	Type	Description
	domain	Domain	

## Sequence Diagram

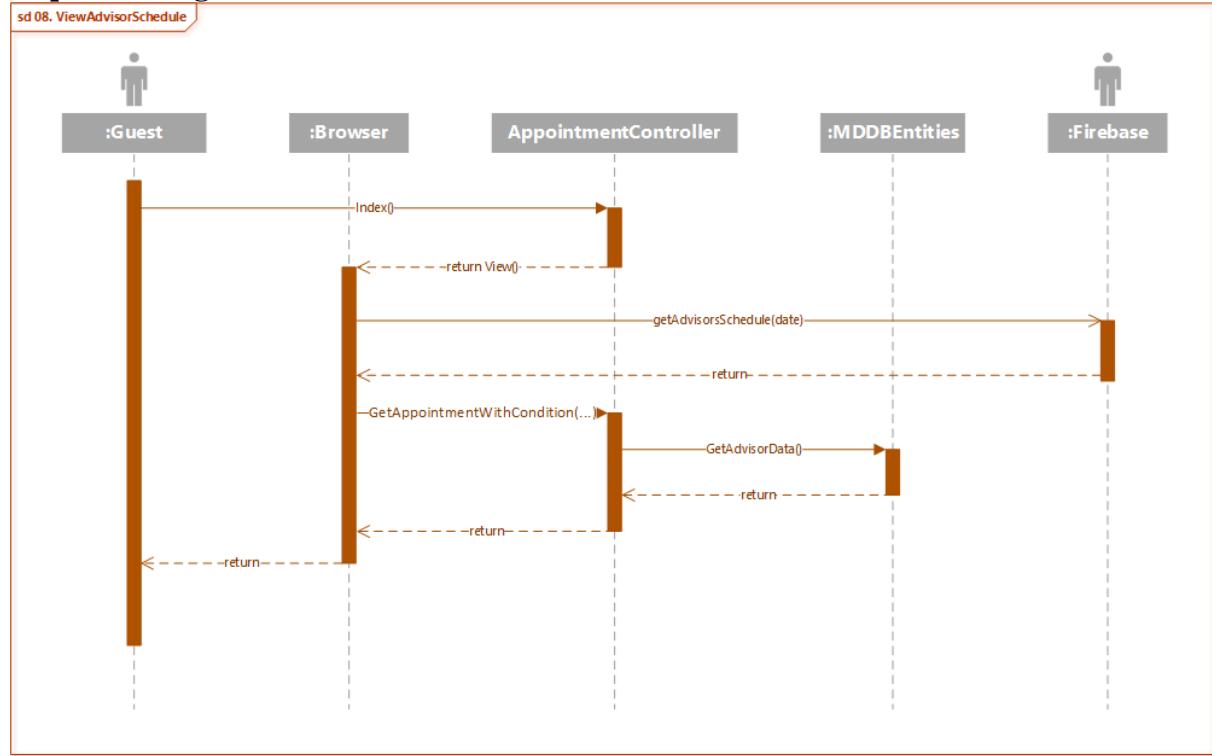


Figure 4-47: View Advisors's schedule Sequence diagram

#### 4.3.4.9 View online Advisors

##### Screen Design



Figure 4-48: View online Advisors Screen design

##### Class Diagram

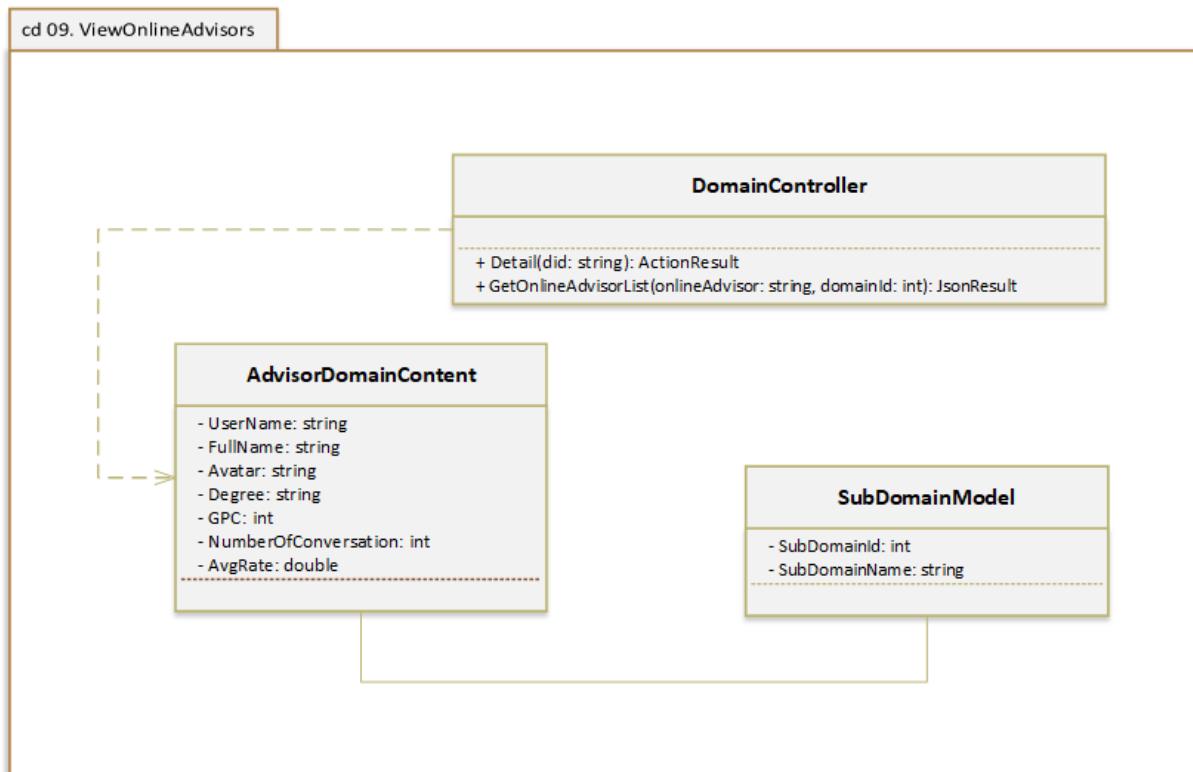


Figure 4-49: View online Advisor Class diagram

##### Class Specification

DomainController			
Physical address	MindDoctor / Controllers / DomainController.cs		
Attributes			
No	Name	Type	Description
Operation			

GetOnlineAdvisorList			
Return Type	JsonResult		
Parameters	Name	Type	Description
	onlineAdvisor	string	List advisor was converted to Json string
	domainId	int	Current domain of detail page

AdvisorDomainContent			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	Avatar	string	
4.	Degree	string	
5.	GPC	int	
6.	NumberOfConversation	int	
7.	AvgRate	double	
Operation			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

## Sequence Diagram

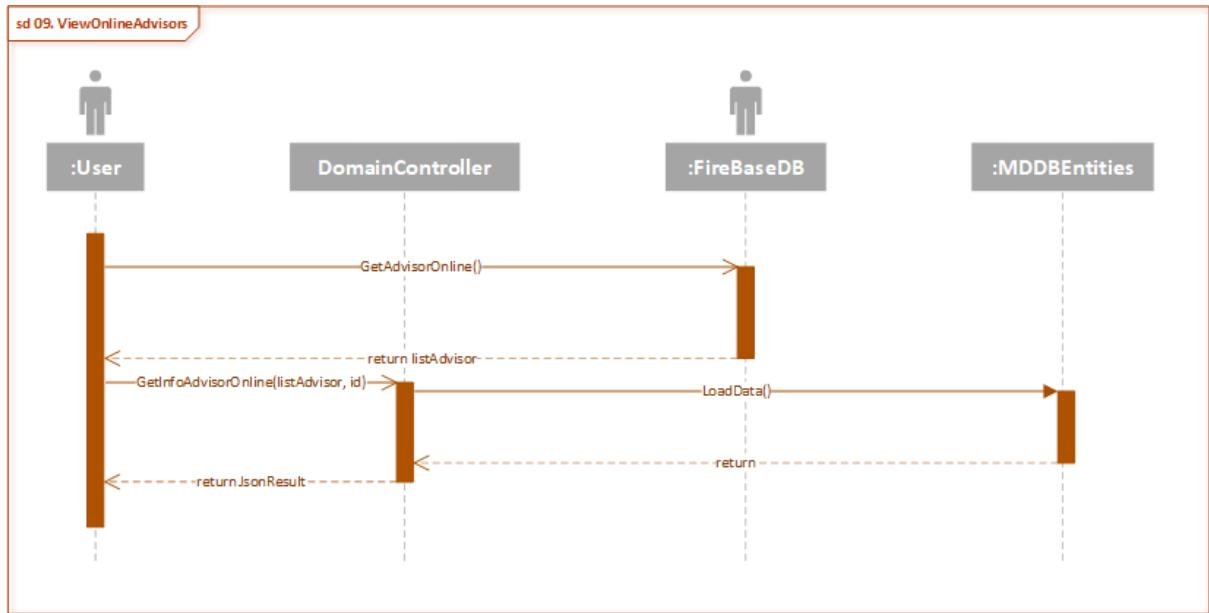
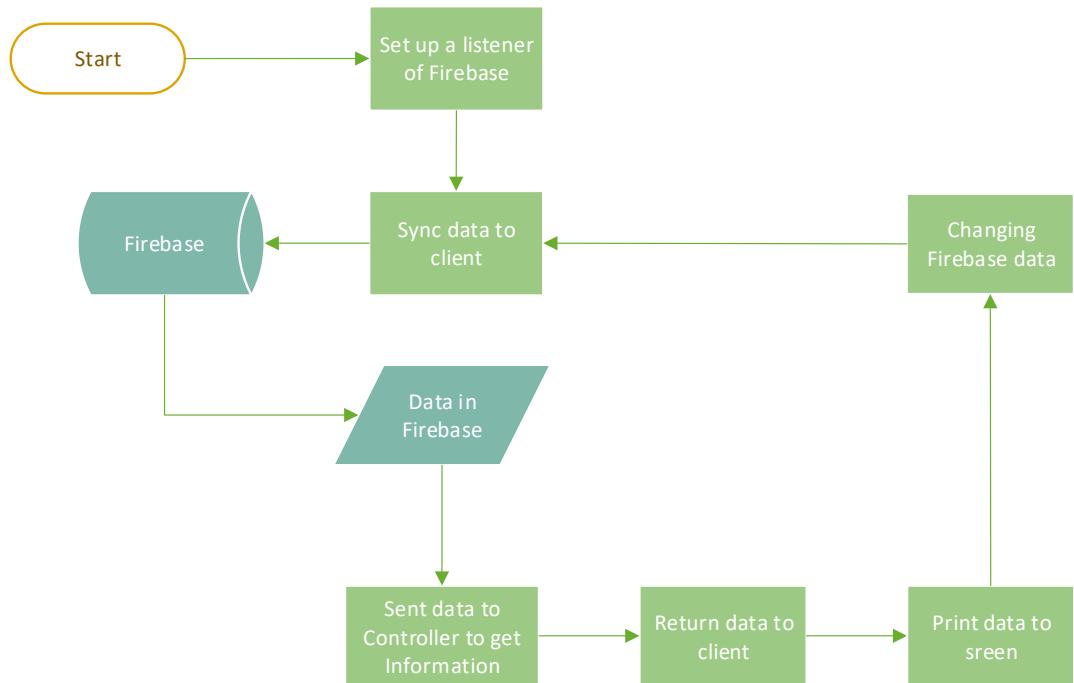


Figure 4-50: View online Advisor Sequence diagram

## Implementation

When user goes to DomainDetail page, all advisor online from Firebase will send to client, then client sends it to DomainController to get information of Advisor who are online. If any status of advisor become offline or online, the database in Firebase will change, Firebase will send online advisor data to client, then client send data to DomainController and so on.



#### 4.3.4.10 View online Advisors by searching Advisor's information

##### Screen Design

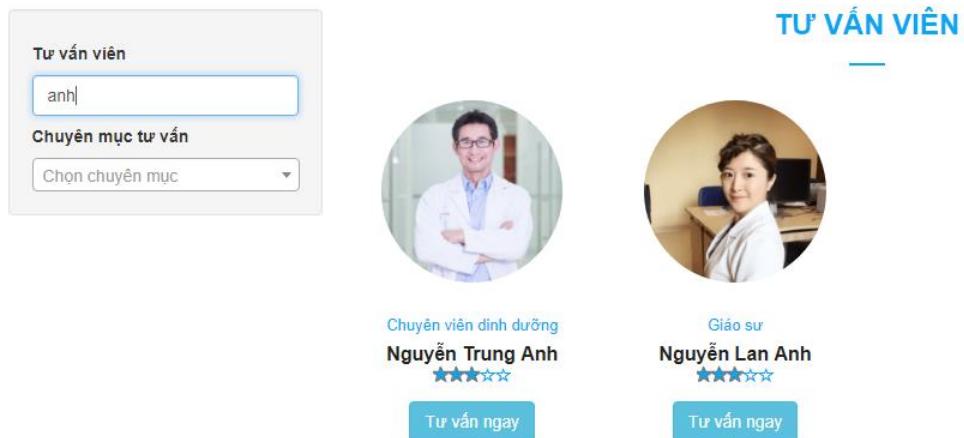


Figure 4-51: View online Advisors by searching Advisor's information Screen design

##### Class Diagram

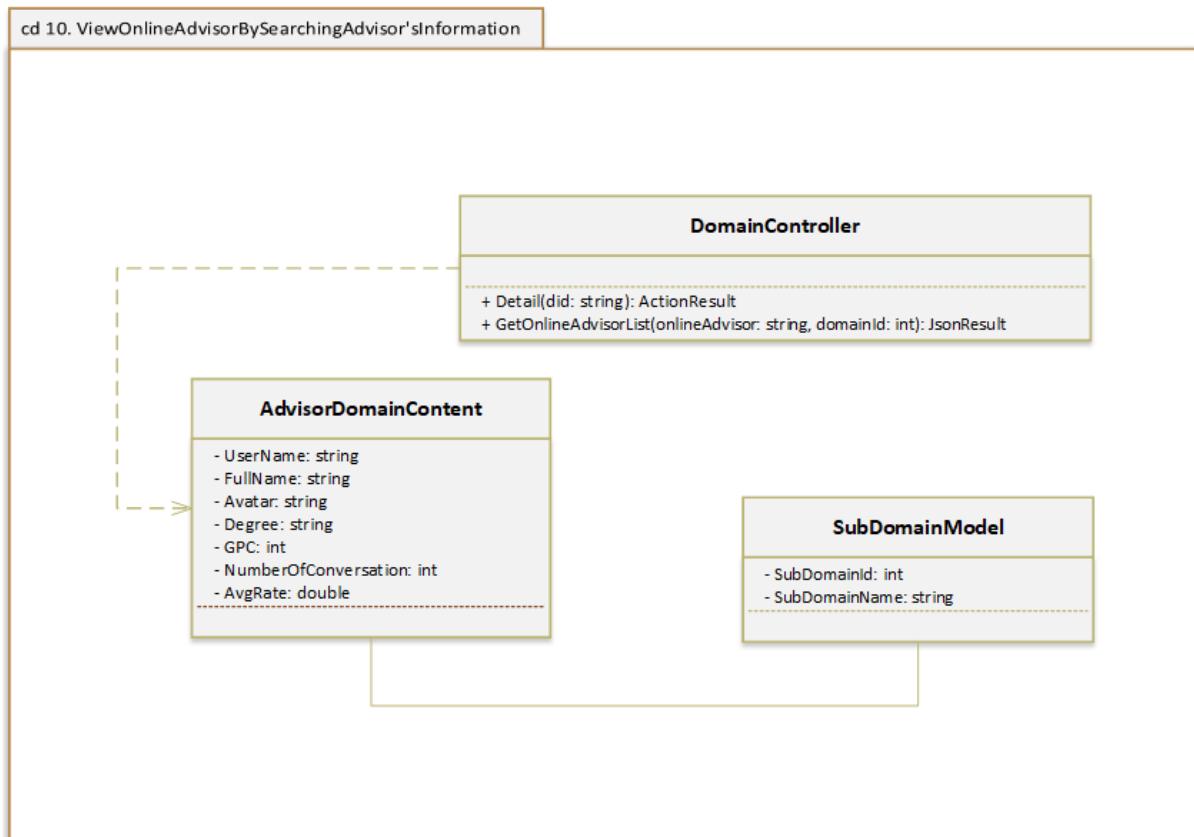


Figure 4-52: View online Advisors by searching Advisor's information Class diagram

##### Class Specification

DomainController	
Physical address	MindDoctor / Controllers / DomainController.cs
Base class	BaseController

Attributes			
No	Name	Type	Description
Operation			
<b>GetOnlineAdvisorList</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	onlineAdvisor	string	List advisor was converted to Json string
	domainId	int	Current domain of detail page

AdvisorDomainContent			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	Avatar	string	
4.	Degree	string	
5.	GPC	int	
6.	NumberOfConversation	int	
7.	AvgRate	double	
Operation			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

## Sequence Diagram

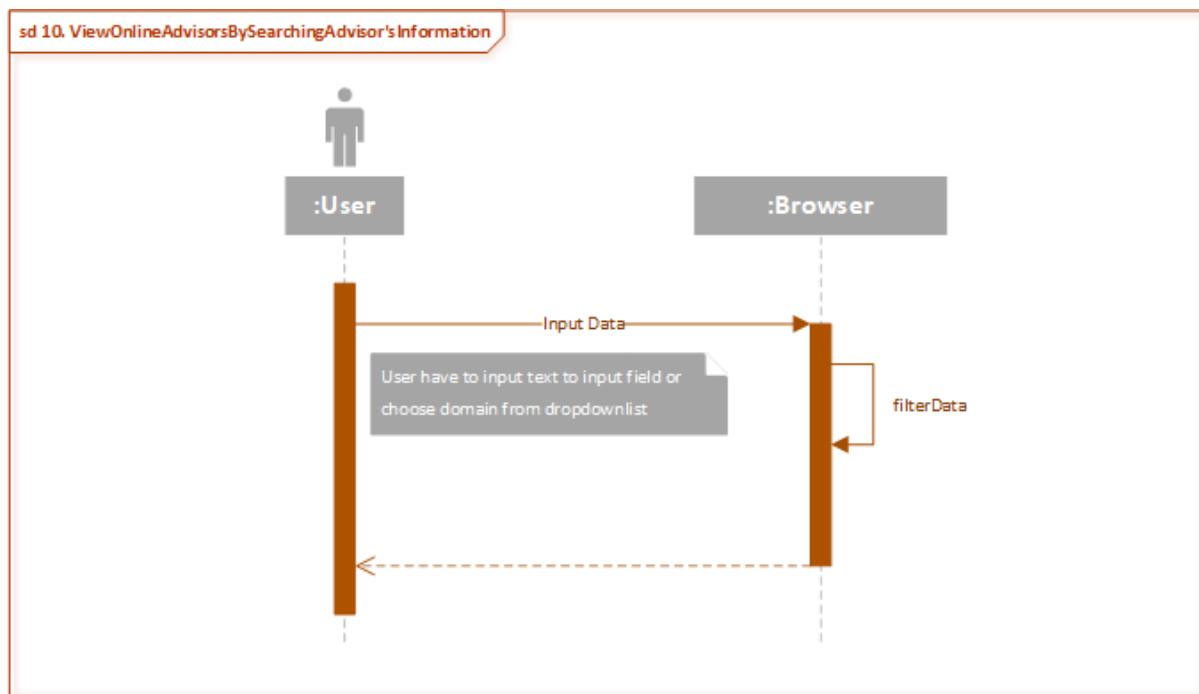


Figure 4-53: View online Advisors by searching Advisor's information Sequence diagram

#### 4.3.4.11 View Advisor's information

##### Screen Design

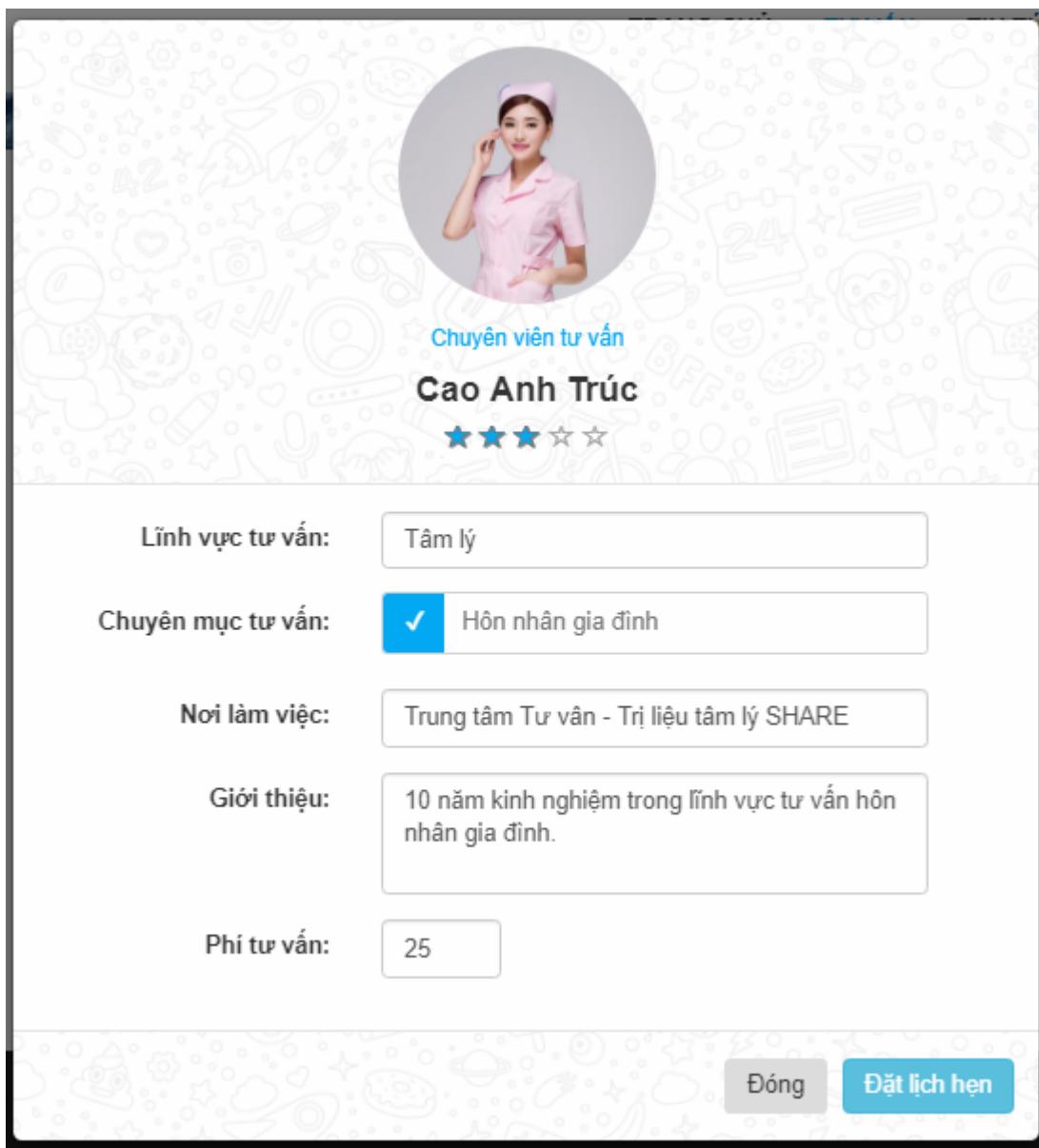


Figure 4-54: View Advisor's information Screen design

## Class Diagram

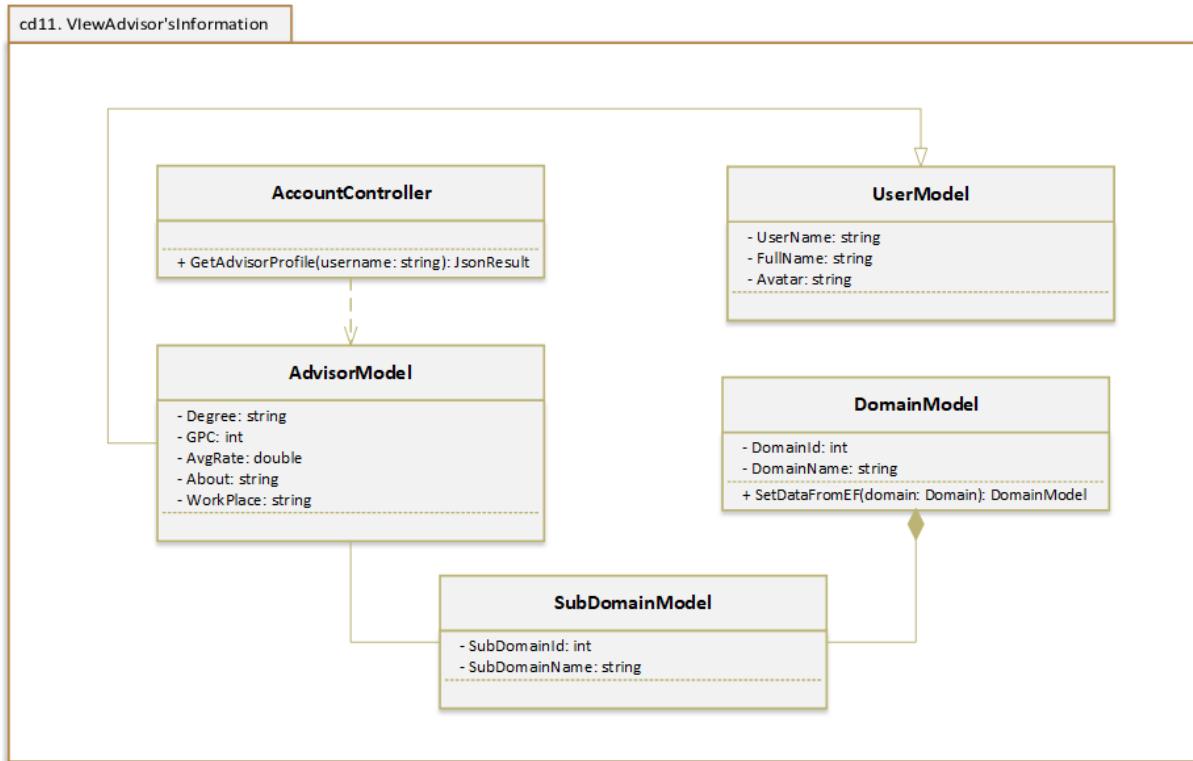


Figure 4-55: View Advisor's information Class diagram

### Class Specification

AccountController			
Physical address	MindDoctor / Controllers / AccountController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
GetAdvisorProfile			
Return Type	JsonResult		
Parameters	Name	Type	Description
	username	string	

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	Degree	string	
2.	GPC	int	
3.	AvgRate	double	
4.	About	string	

5.	WorkPlace	string	
Operation			

<b>UserModel</b>			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	Avatar	string	
Operation			

<b>DomainModel</b>			
Physical address	MindDoctor / Models / DomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
Operation			
SetDataFromEF			
Return Type	DomainModel		
Parameters	Name	Type	Description
	domain	Domain	This class is Domain class from entity framework

<b>SubDomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

## Sequence Diagram

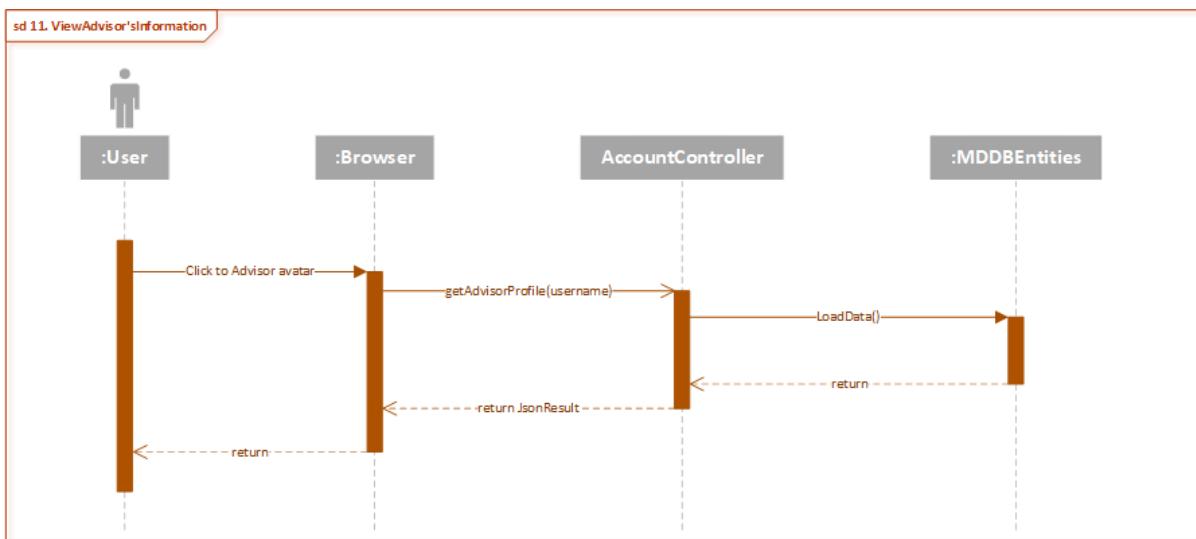


Figure 4-56: View Advisor's information Sequence diagram

#### 4.3.4.12 Login with local account

##### Screen Design

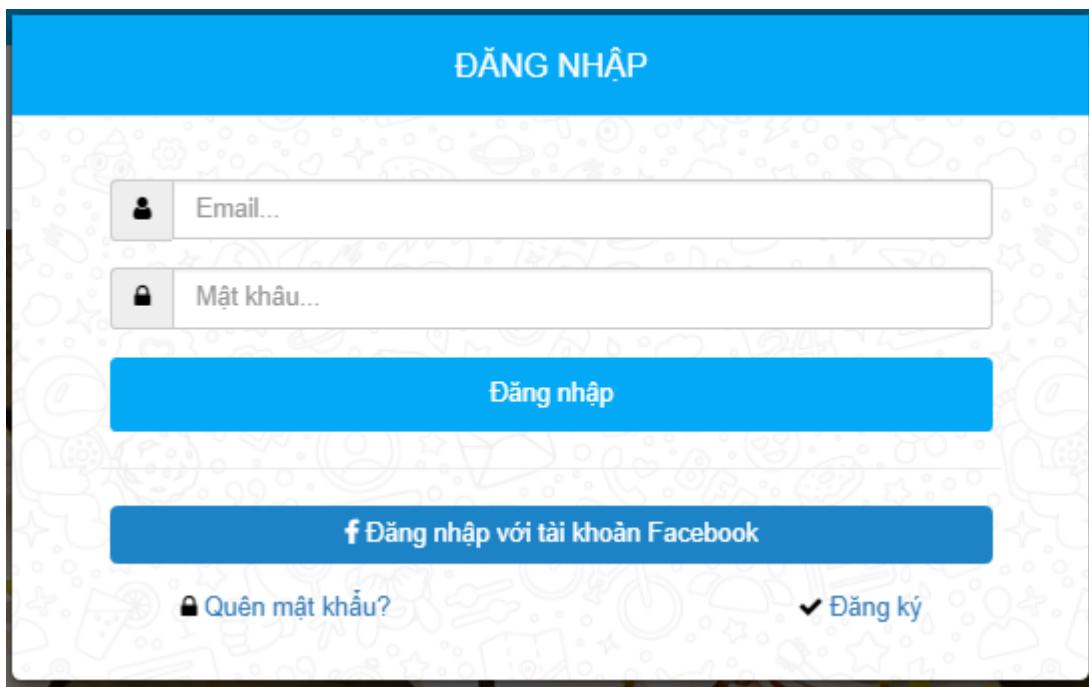


Figure 4-57: Login with local account Screen design

##### Class Diagram

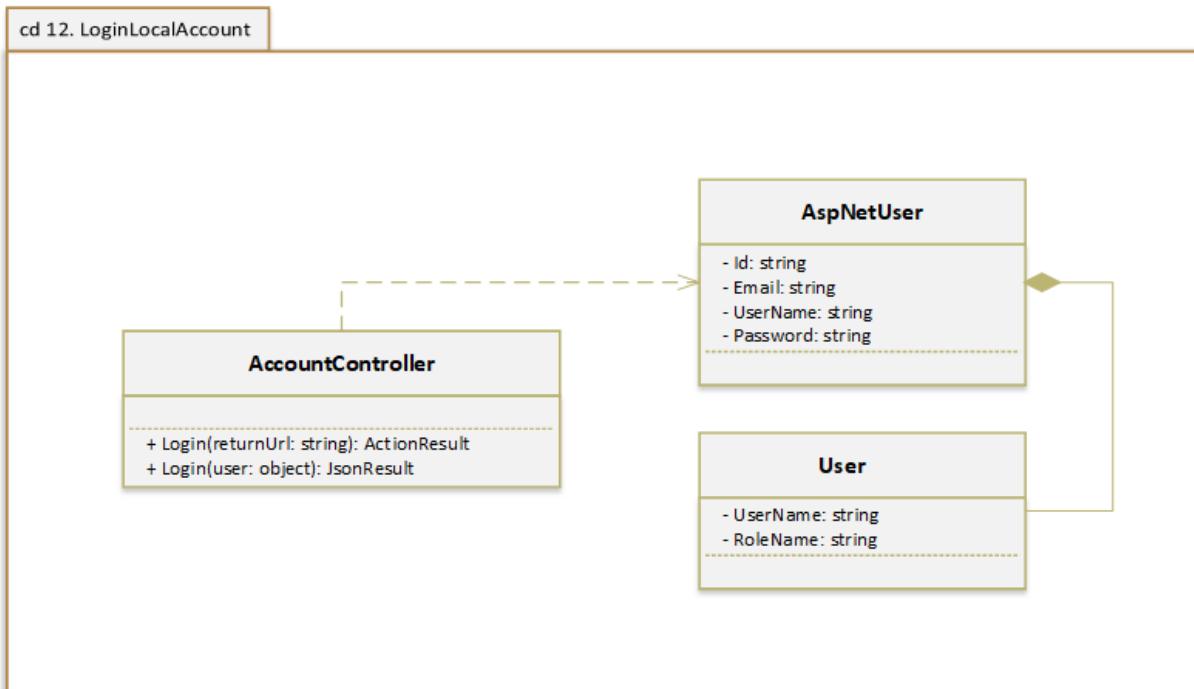


Figure 4-58: Login with local account Class diagram

##### Class Specification

AccountController	
Physical address	MindDoctor / Controllers / AccountController.cs
Base class	

Attributes			
No	Name	Type	Description
Operation			
<b>Login</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
	returnUrl	string	
<b>Login</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	user	object	Object contains user name and password.

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	
2.	Email	string	
3.	UserName	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	RoleName	string	
Operation			

## Sequence Diagram

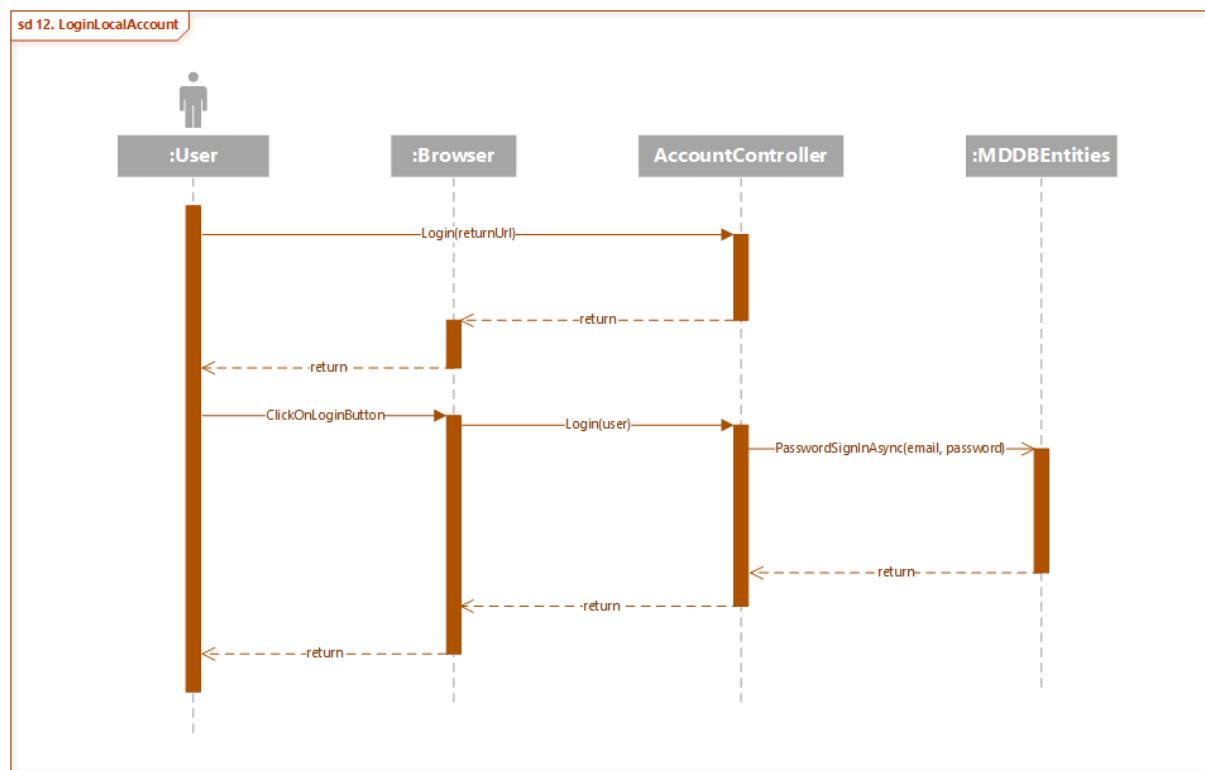


Figure 4-59: Login with local account Sequence diagram

#### 4.3.4.13 Login with Facebook account

##### Screen Design



Figure 4-60: Login with Facebook account Screen design

##### Class Diagram

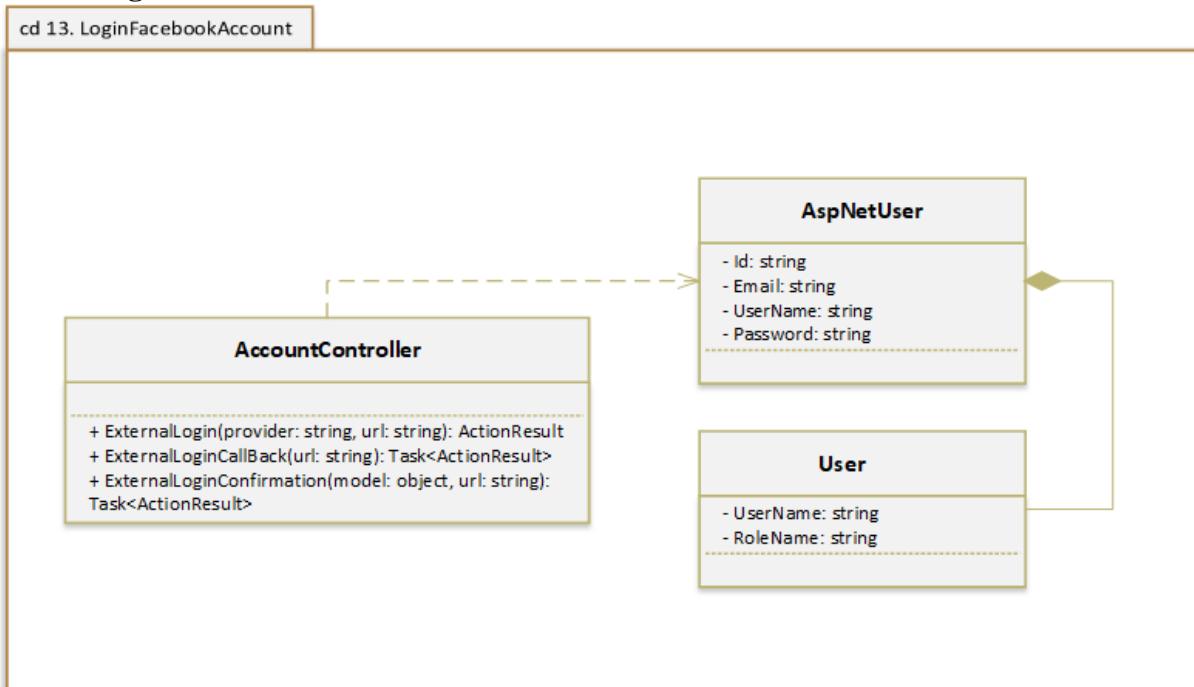


Figure 4-61: Login with Facebook account Class diagram

##### Class Specification

AccountController	
Physical address	MindDoctor / Controllers / AccountController.cs
Base class	

Attributes			
No	Name	Type	Description
Operation			
<b>ExternalLogin</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
	provider	string	Name of 3 <sup>rd</sup> party (Facebook, Google...).
	url	string	
<b>ExternalLoginCallBack</b>			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	url	string	
<b>ExternalLoginConfirmation</b>			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	model	object	Contain user information from 3 <sup>rd</sup> party.
	url	string	

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	
2.	Email	string	
3.	UserName	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	RoleName	string	
Operation			

## Sequence Diagram

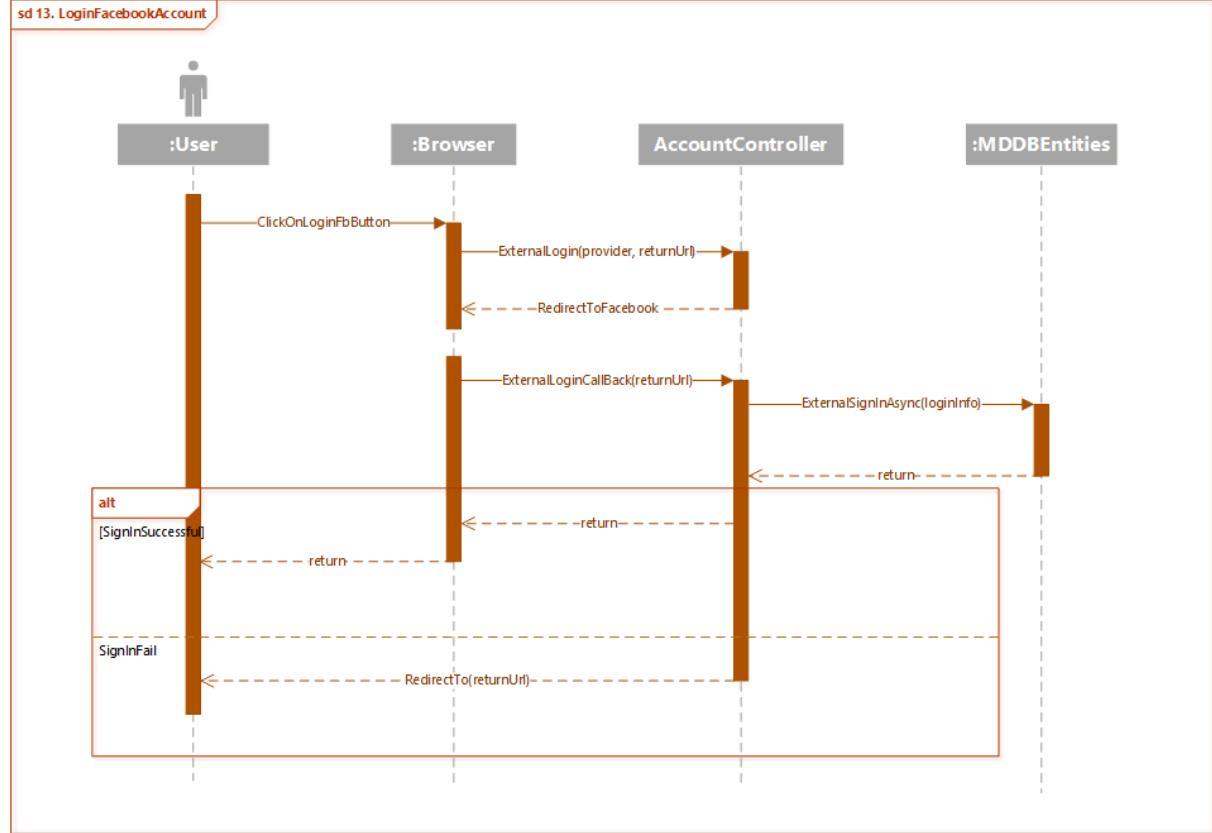


Figure 4-62: Login with Facebook account Sequence diagram

#### 4.3.4.14 Logout

##### Screen Design

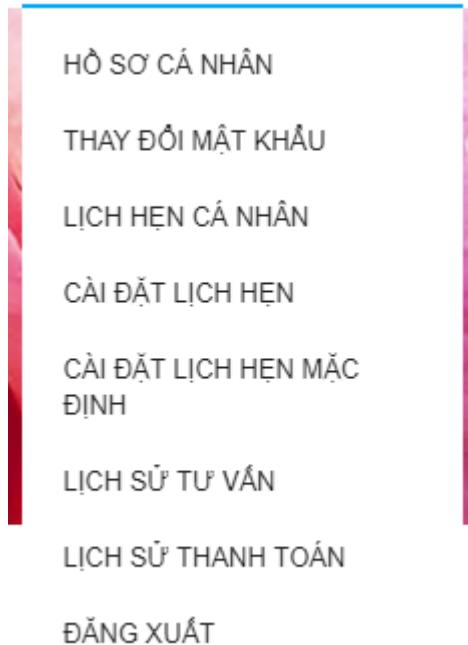


Figure 4-63: Logout Screen design

##### Class Diagram

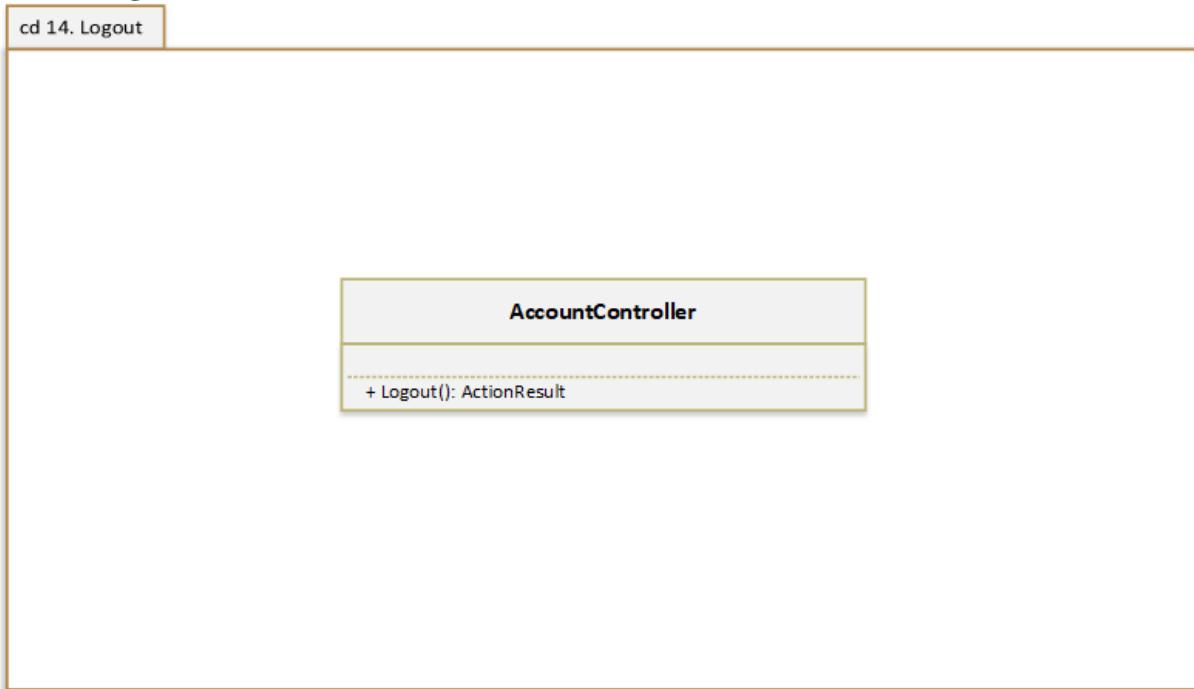


Figure 4-64: Logout Class diagram

##### Class Specification

AccountController	
Physical address	MindDoctor / Controllers / AccountController.cs

Base class			
Attributes			
No	Name	Type	Description
Operation			
Logout			
Return Type	ActionResult		
Parameters	Name	Type	Description

## Sequence Diagram

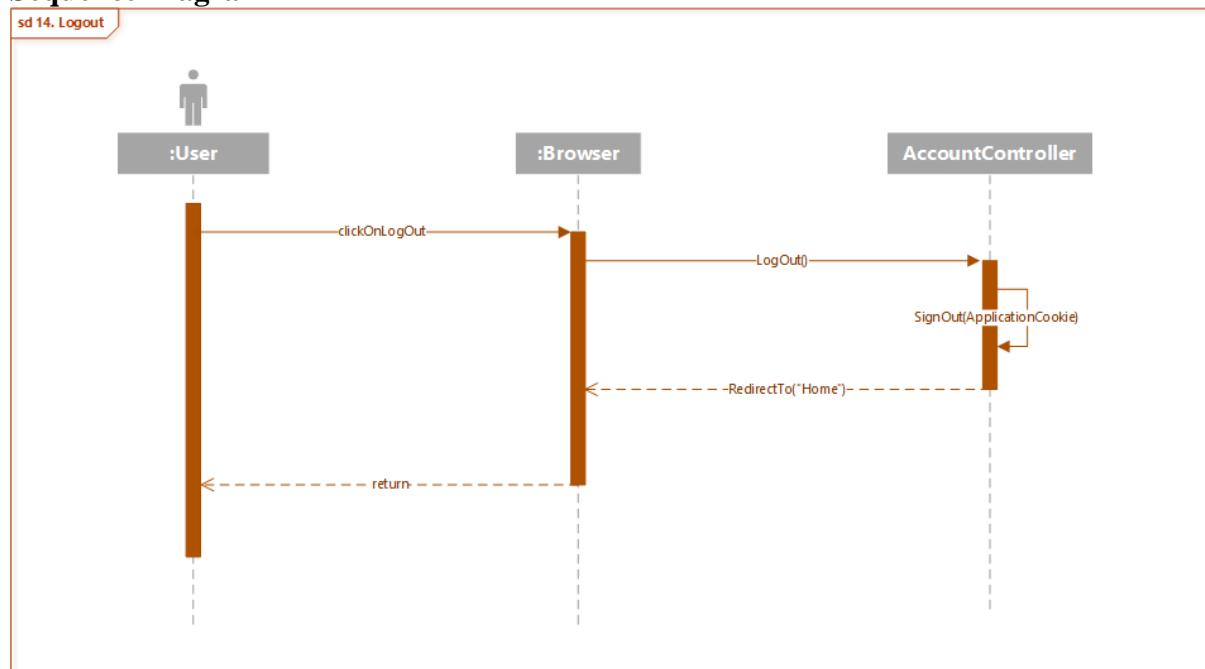


Figure 4-65: Logout Sequence diagram

#### 4.3.4.15 Reset password

##### Screen Design

The image displays two versions of a password reset screen design. The top version is a wireframe or low-fidelity mockup, while the bottom version is a high-fidelity design.

**Top Version (Wireframe):**

- Header:** QUÊN MẬT KHẨU
- Input Field:** Email... (with '@' icon)
- Primary Button:** Lấy lại mật khẩu
- Secondary Buttons:** Đăng nhập and Đăng ký

**Bottom Version (High-Fidelity):**

- Header:** LẤY LẠI MẬT KHẨU
- Input Fields:** Email, Mật khẩu, Xác nhận mật khẩu
- Primary Button:** Lưu

Figure 4-66: Reset password Screen design

## Class Diagram

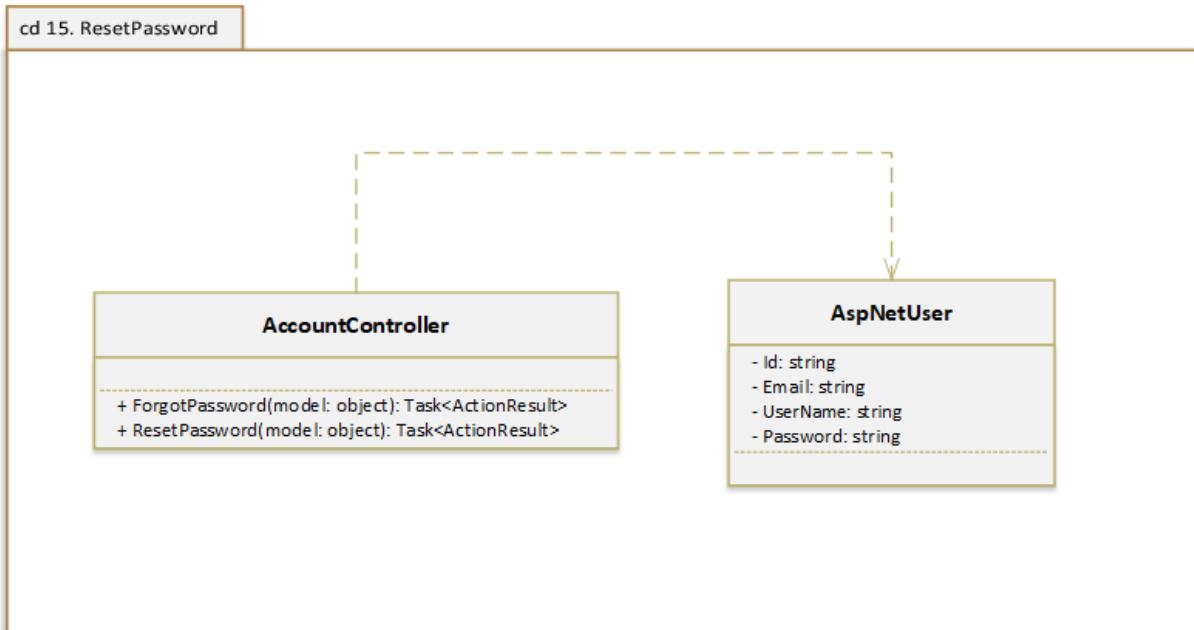


Figure 4-67: Reset password Class diagram

## Class Specification

AccountController			
Physical address	MindDoctor / Controllers / AccountController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
ForgotPassword			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	model	object	Contain user information.
ResetPassword			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	model	object	Contain user information.

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	

2.	Email	string	
3.	UserName	string	
4.	Password	string	
Operation			

## Sequence Diagram

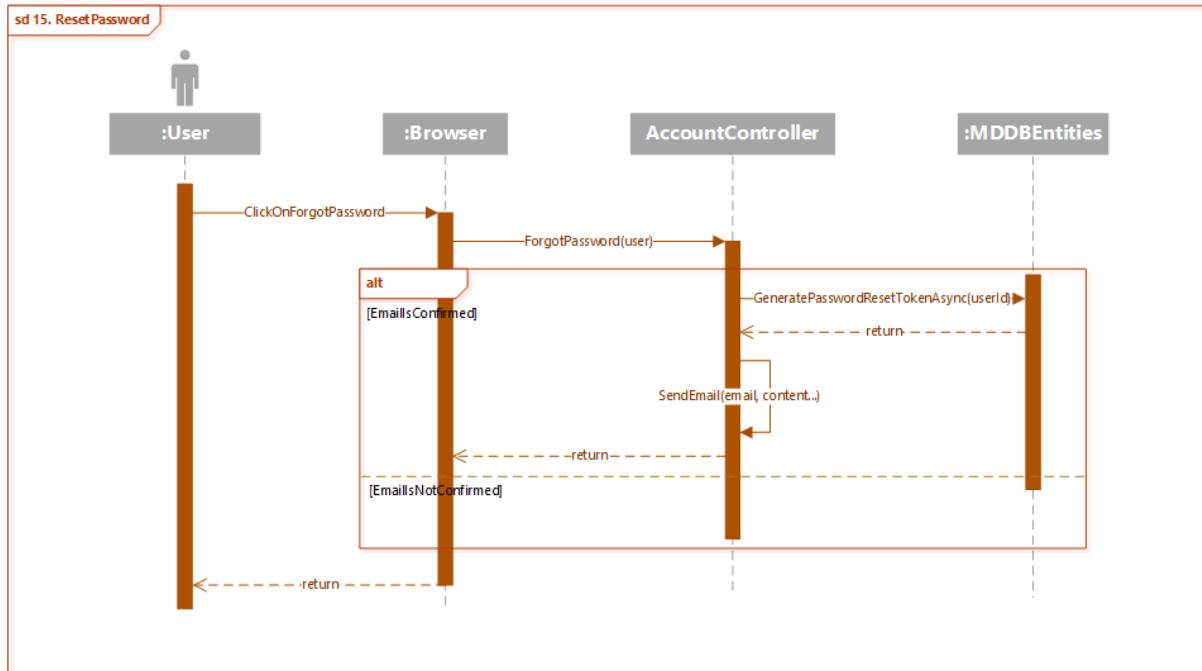


Figure 4-68: Reset password Sequence diagram

#### 4.3.4.16 Change password

##### Screen Design

**THAY ĐỔI MẬT KHẨU**

---

Mật khẩu hiện tại

Mật khẩu mới

Xác nhận mật khẩu

Thay đổi mật khẩu

Figure 4-69: Change password Screen design

##### Class Diagram

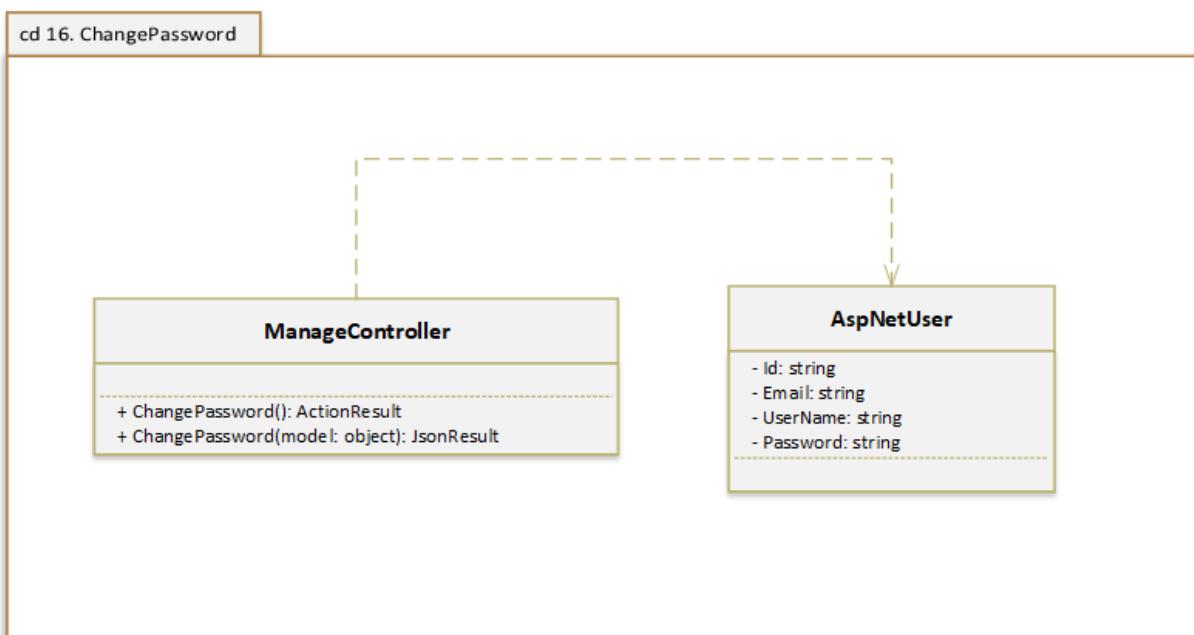


Figure 4-70: Change password Class diagram

##### Class Specification

ManageController	
Physical address	MindDoctor / Controllers / ManageController.cs

Base class			
Attributes			
No	Name	Type	Description
Operation			
<b>ChangePassword</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
<b>ChangePassword</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Contain user information.

AspNetUser			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AspNetUser.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	string	
2.	Email	string	
3.	UserName	string	
4.	Password	string	
Operation			

## Sequence Diagram

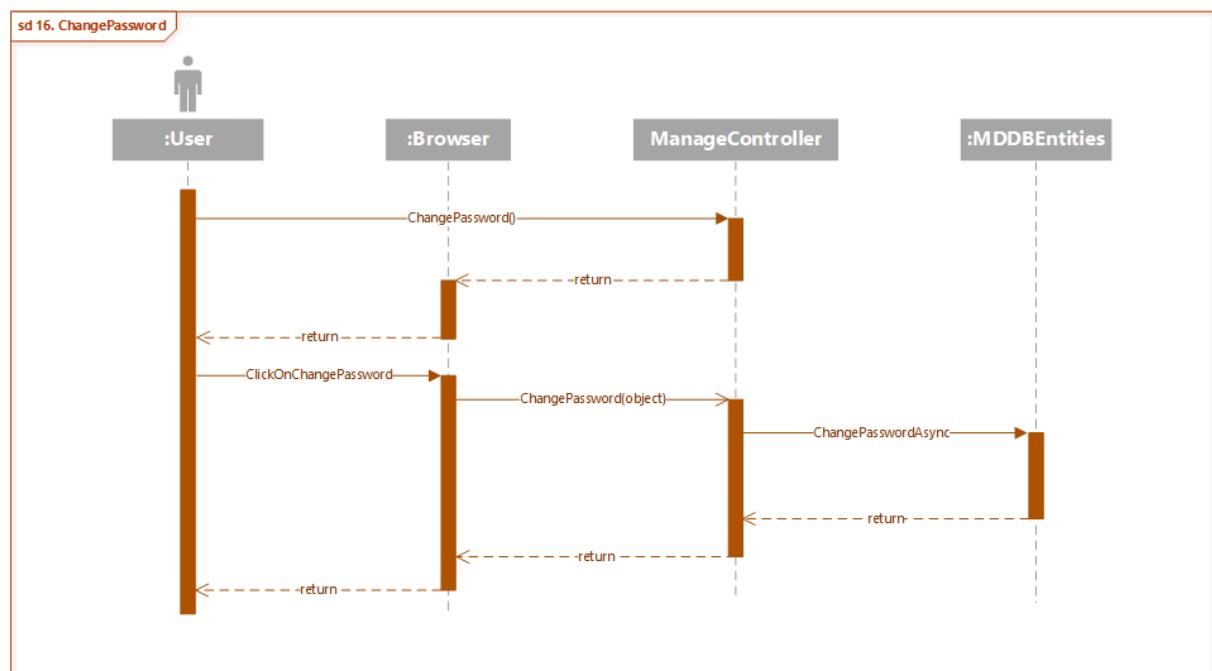


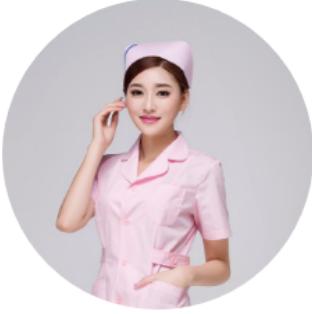
Figure 4-71: Change password Sequence diagram

#### 4.3.4.17 View personal profile

##### Screen Design

**HỒ SƠ CÁ NHÂN**

---



Cập nhật ảnh đại diện

0

**Thông tin cá nhân**

Trình độ:	Chuyên viên tư vấn
Họ và tên:	Cao Anh Trúc
Ngày sinh:	18 Tháng Năm 1983
Giới tính:	Nữ
Email:	mds.adv20@mailnesia.com <span style="color: green;">✓</span>
Quốc gia:	Việt Nam <span style="color: red;">x v</span>
Tỉnh/Thành phố:	Đắk Lăk <span style="color: red;">x v</span>
Số điện thoại:	0166 147 8523
Nơi làm việc:	Trung tâm Tư vấn - Trị liệu tâm lý SHARE
Giới thiệu:	10 năm kinh nghiệm trong lĩnh vực tư vấn hôn nhân gia đình.
GPC:	25 <span style="border: 1px solid #ccc; padding: 2px;">^</span> <span style="border: 1px solid #ccc; padding: 2px;">v</span>

---

**Thông tin tài khoản ngân hàng**

Ngân hàng:	Vietcombank
Chi nhánh:	Hà Nội
Số tài khoản:	4458 5887 44585
Tên tài khoản:	CAO ANH TRUC

**Cập nhật**

---

**Lĩnh vực tư vấn**

Figure 4-72: View personal profile Screen design

## Class Diagram

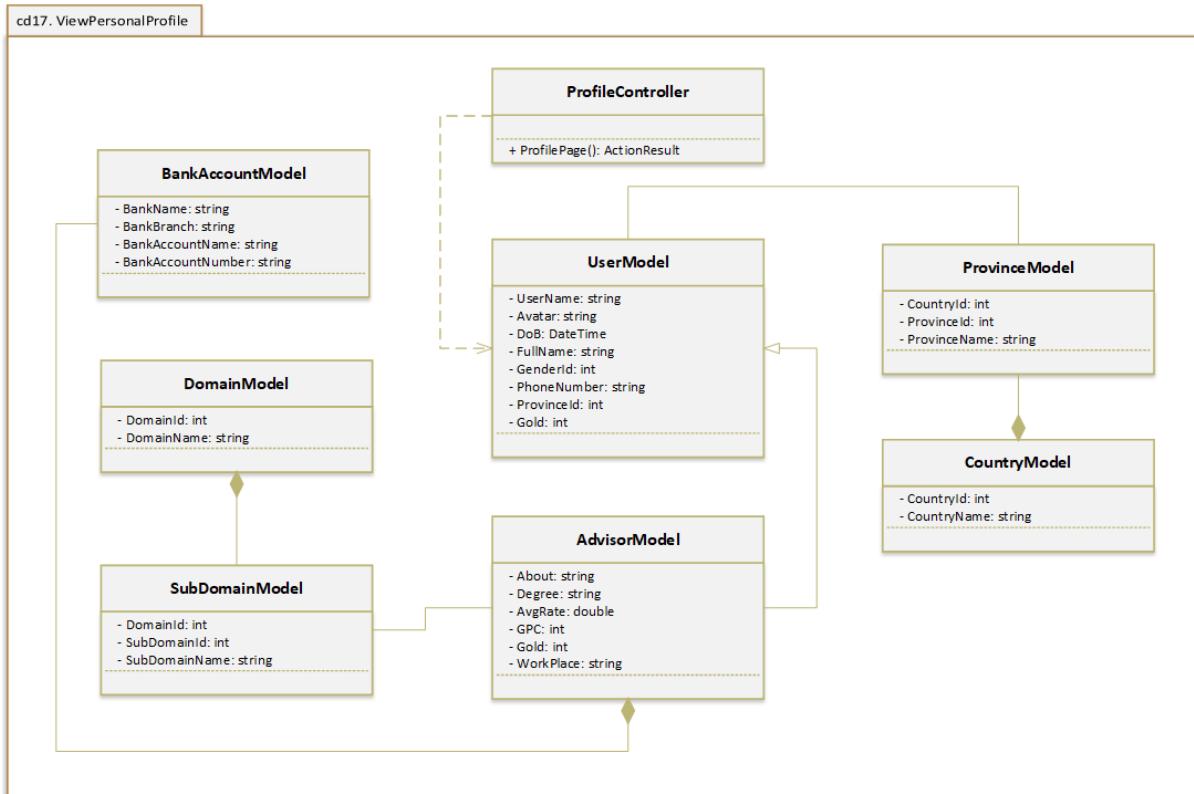


Figure 4-73: View personal profile Class diagram

## Class Specification

ProfileController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ProfileController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
ProfilePage			
Return Type	ActionResult		
Parameters	Name	Type	Description

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	Avatar	string	
3.	DoB	DateTime	
4.	FullName	string	

5.	GenderId	int	
6.	PhoneNumber	string	
7.	ProvinceId	int	
8.	Gold	int	
Operation			

CountryModel			
Physical address	MindDoctor / Models / CountryModels / CountryModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	CountryId	int	
2.	CountryName	string	
Operation			

ProvinceModel			
Physical address	MindDoctor / Models / CountryModels / CountryModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	CountryId	int	
2.	ProvinceId	int	
3.	ProvinceName	string	
Operation			

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	About	string	
2.	Degree	string	
3.	AvgRate	double	
4.	GPC	int	
5.	Gold	int	
6.	WorkPlace	string	
Operation			

DomainModel			
-------------	--	--	--

Physical address	MindDoctor / Models / DomainModels / DomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
Operation			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	SubDomainId	int	
3.	SubDomainName	string	
Operation			

BankAccountModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	BankName	string	
2.	BankBranch	string	
3.	BankAccountName	string	
4.	BankAccountNumber	string	
Operation			

## Sequence Diagram

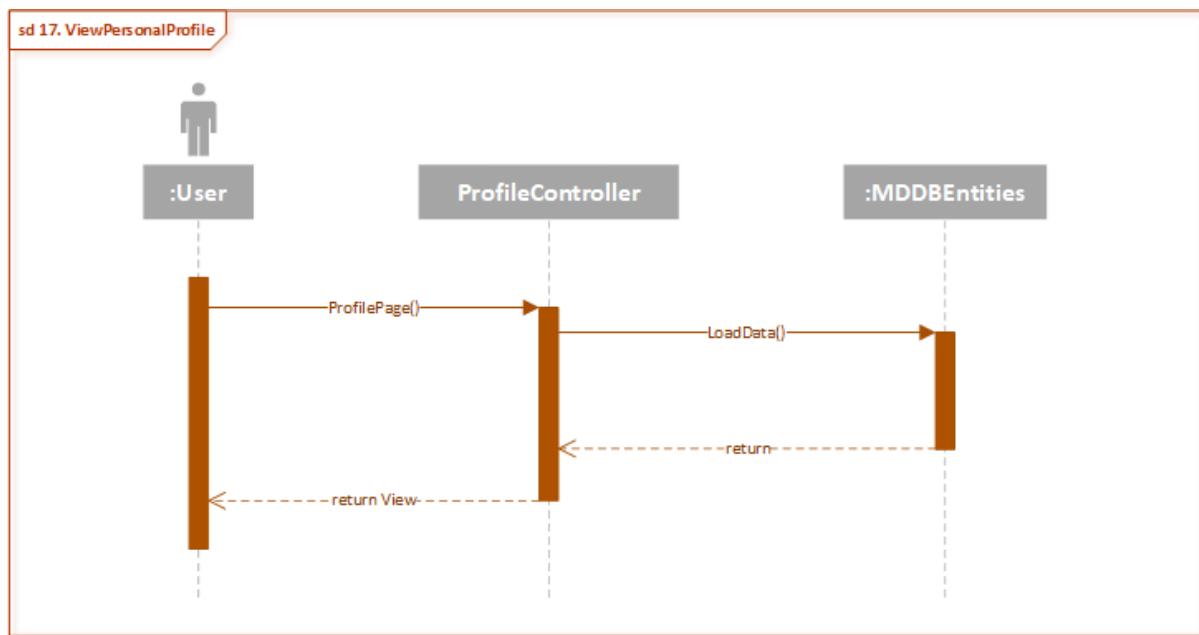


Figure 4-74: View personal profile Sequence diagram

#### 4.3.4.18 Update personal profile

##### Screen Design

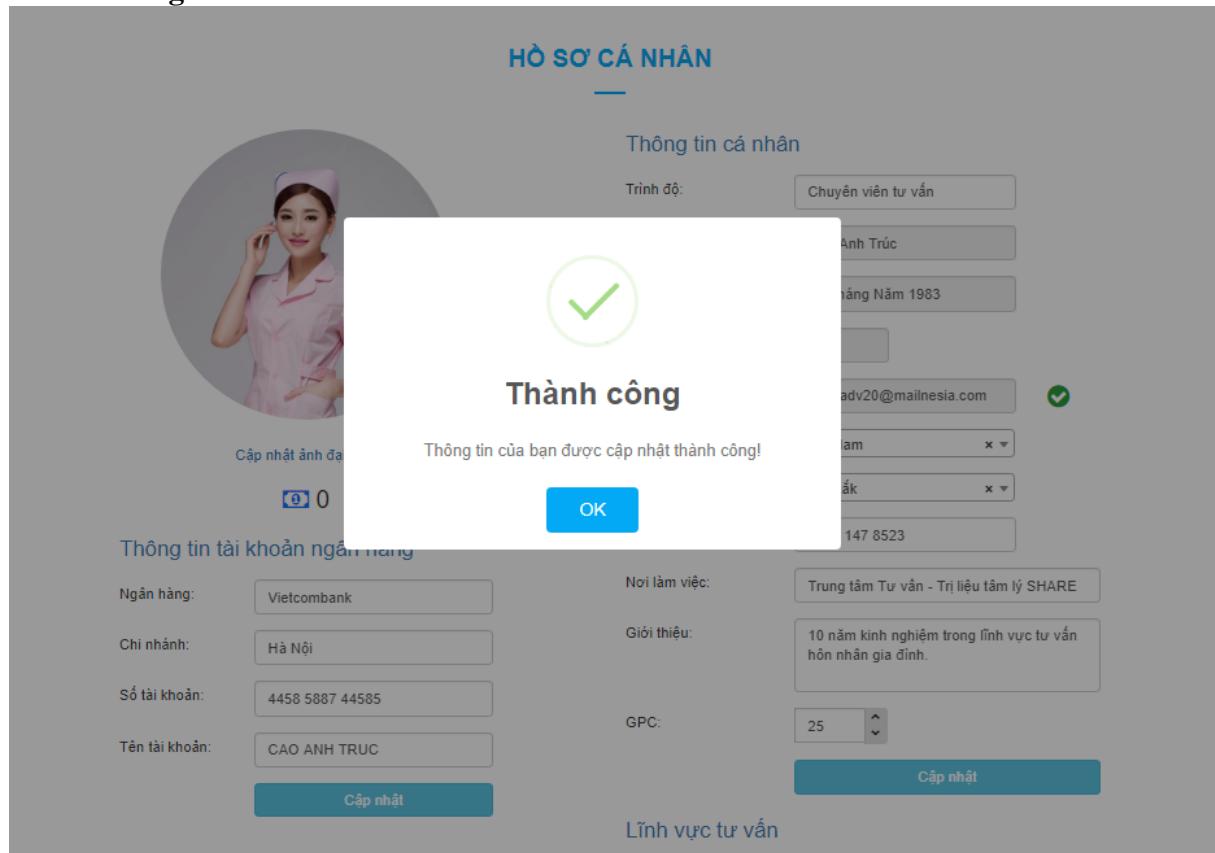


Figure 4-75: Update personal profile Screen design

##### Class Diagram

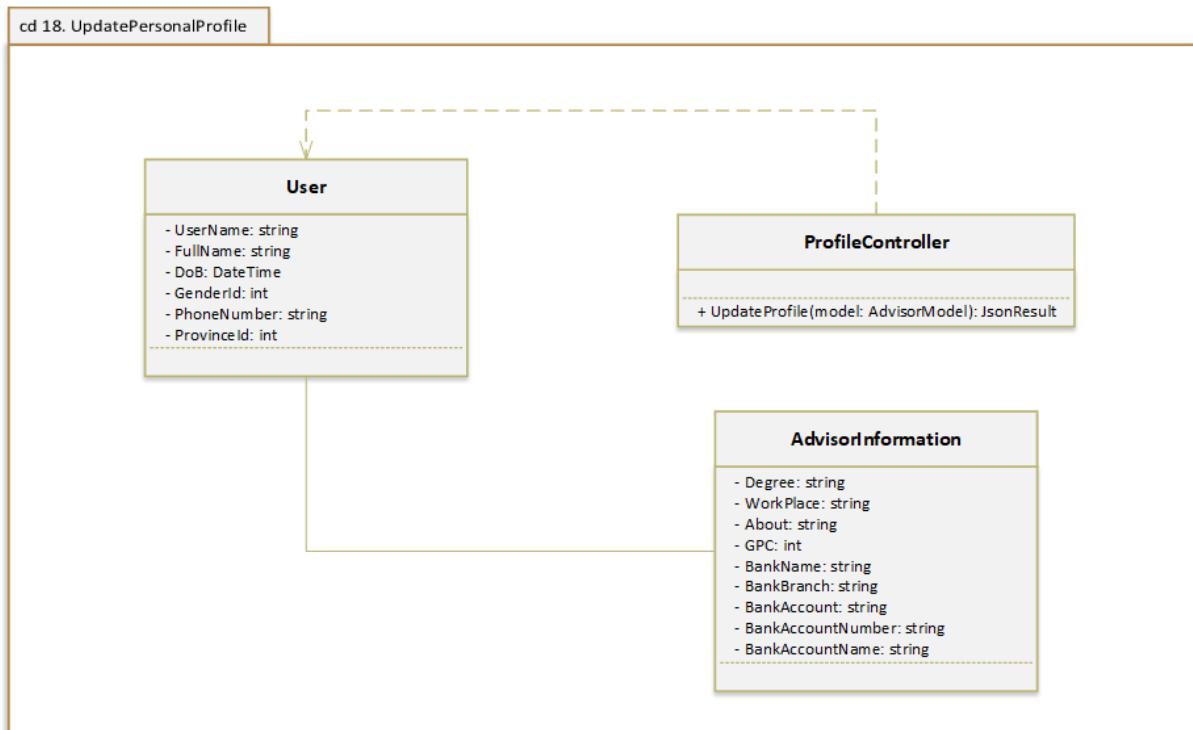


Figure 4-76: Update personal profile Class diagram

## Class Specification

ProfileController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ProfileController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
UpdateProfile			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	AdvisorModel	

AdvisorInformation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AdvisorInformation.cs		
Base class	UserModel.cs		
Attributes			
No	Name	Type	Description
1.	Degree	string	
2.	WorkPlace	string	
3.	About	string	
4.	GPC	int	
5.	BankName	string	
6.	BankBranch	string	
7.	BankAccount	string	
8.	BankAccountNumber	string	
9.	BankAccountName	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	DoB	DateTime	
4.	GenderId	int	
5.	PhoneNumber	string	
6.	ProvinceId	int	
Operation			

## Sequence Diagram

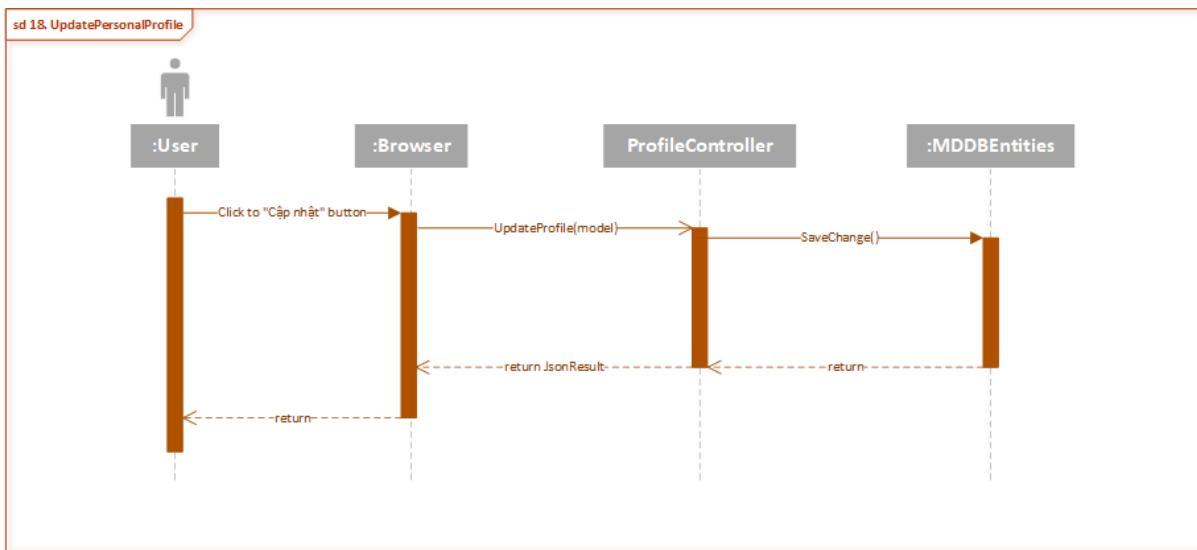


Figure 4-77: Update personal profile Sequence diagram

#### 4.3.4.19 Insert money via Paypal

##### Screen Design

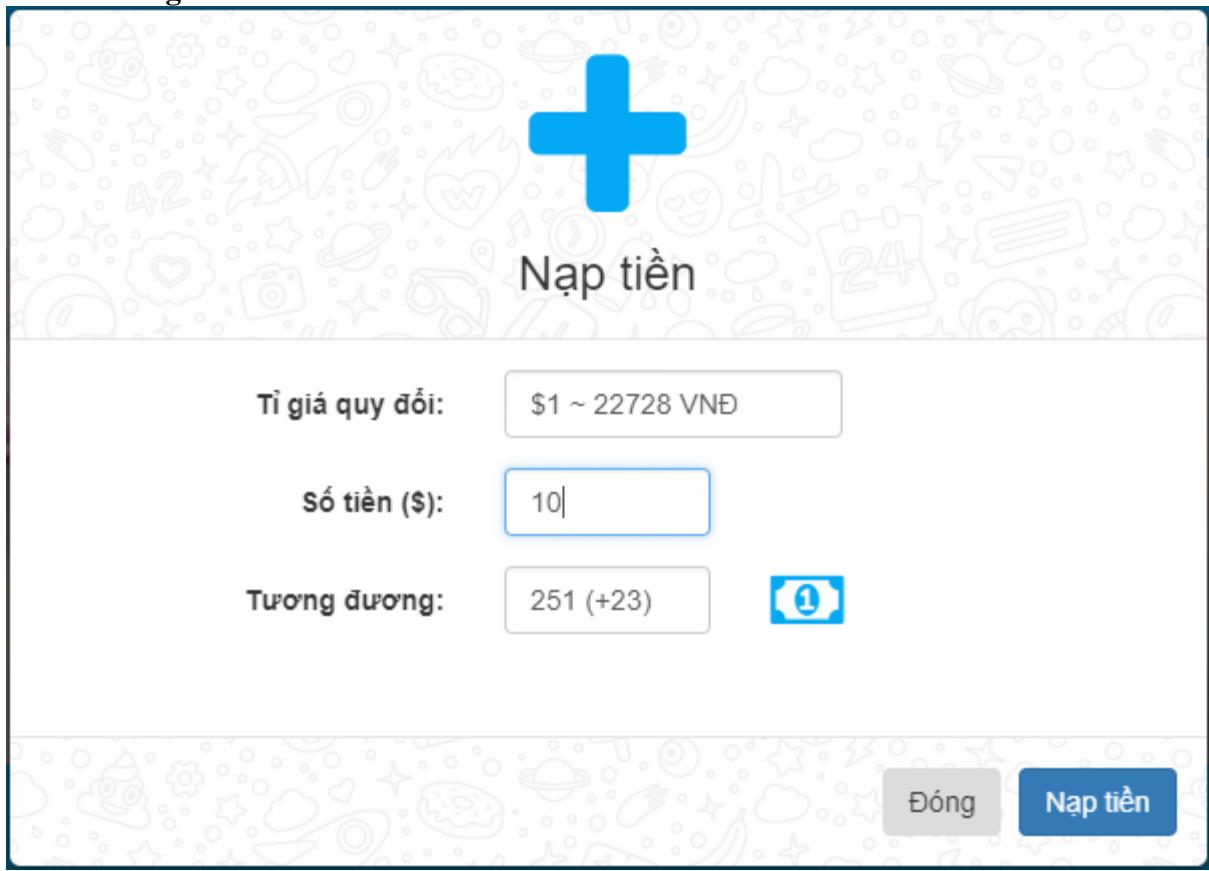


Figure 4-78: Insert money via Paypal Screen design

##### Class Diagram

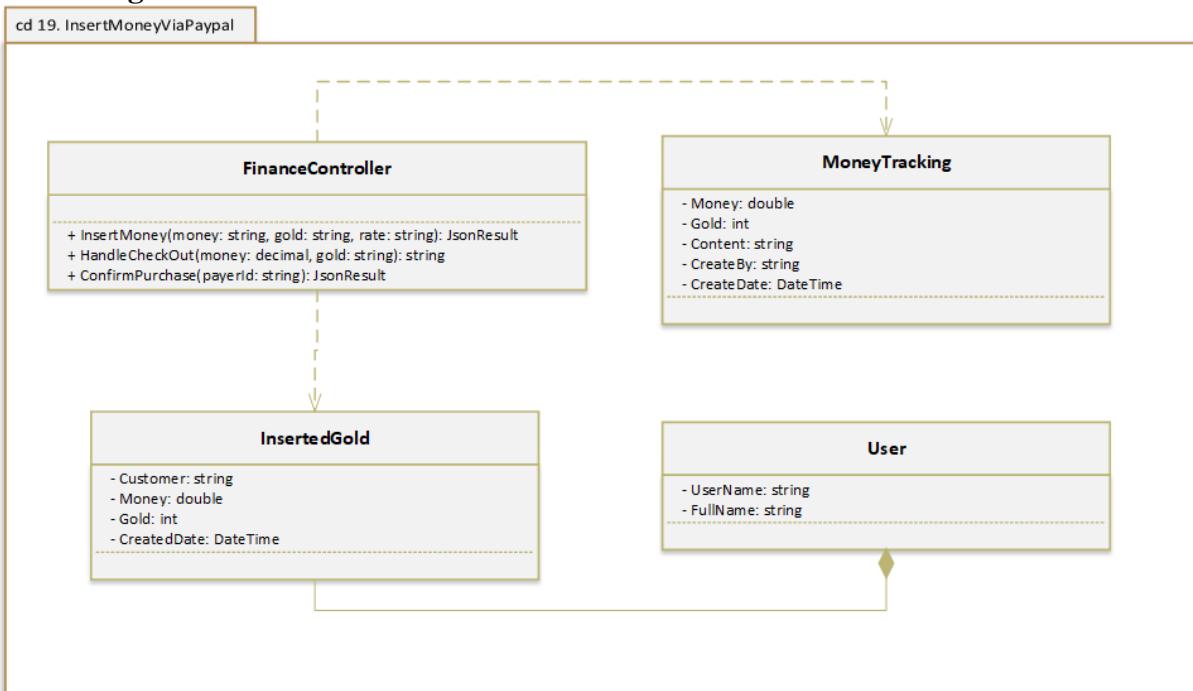


Figure 4-79: Insert money via Paypal Class diagram

## Class Specification

FinanceController			
Physical address	MindDoctor / Controllers / FinanceModel.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
InsertMoney			
Return Type	JsonResult		
Parameters	Name	Type	Description
	money	string	
	gold	int	
	gold	string	
HandleCheckOut			
Return Type	string		
Parameters	Name	Type	Description
	money	decimal	
	gold	string	
ConfirmPurchase			
Return Type	JsonResult		
Parameters	Name	Type	Description
	payerId	string	

## MoneyTracking

Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / MoneyTracking.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Money	double	
2.	Gold	int	
3.	Content	string	
4.	CreateBy	string	
5.	CreateDate	DateTime	
Operation			

## InsertedGold

Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / InsertedGold.cs		
Base class			
Attributes			
No	Name	Type	Description

1.	Customer	string	
2.	Money	double	
3.	Gold	int	
4.	CreatedDate	DateTime	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
Operation			

## Sequence Diagram

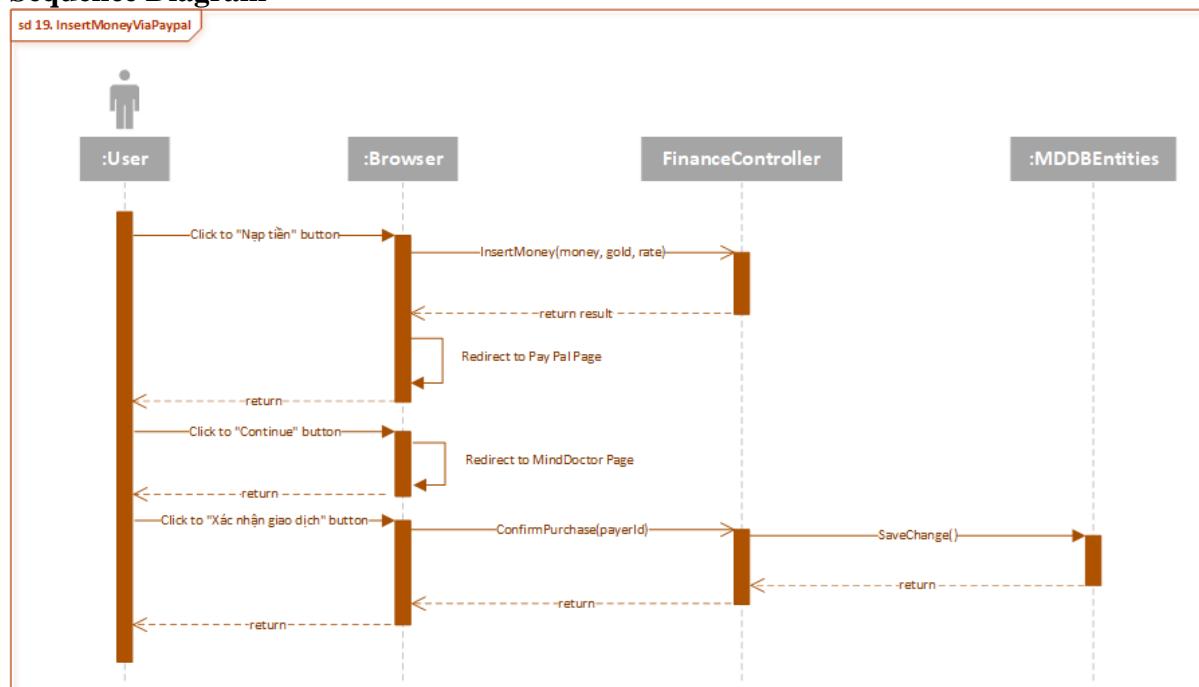


Figure 4-80: Insert money via Paypal Sequence diagram

#### 4.3.4.20 View inserted money history

##### Screen Design

LỊCH SỬ NẠP TIỀN			
HÌNH THỨC	MÊNH GIÁ (VND)	STT	NGÀY
Paypal	113.630	114	18:16 - 13 Tháng Tám 2017
Paypal	227.260	250	18:15 - 13 Tháng Tám 2017
<input type="button" value="«"/> <input type="button" value="&lt;"/> <input type="button" value="1"/> <input type="button" value="&gt;"/> <input type="button" value="»"/>			

Figure 4-81: View inserted money history Screen design

##### Class Diagram

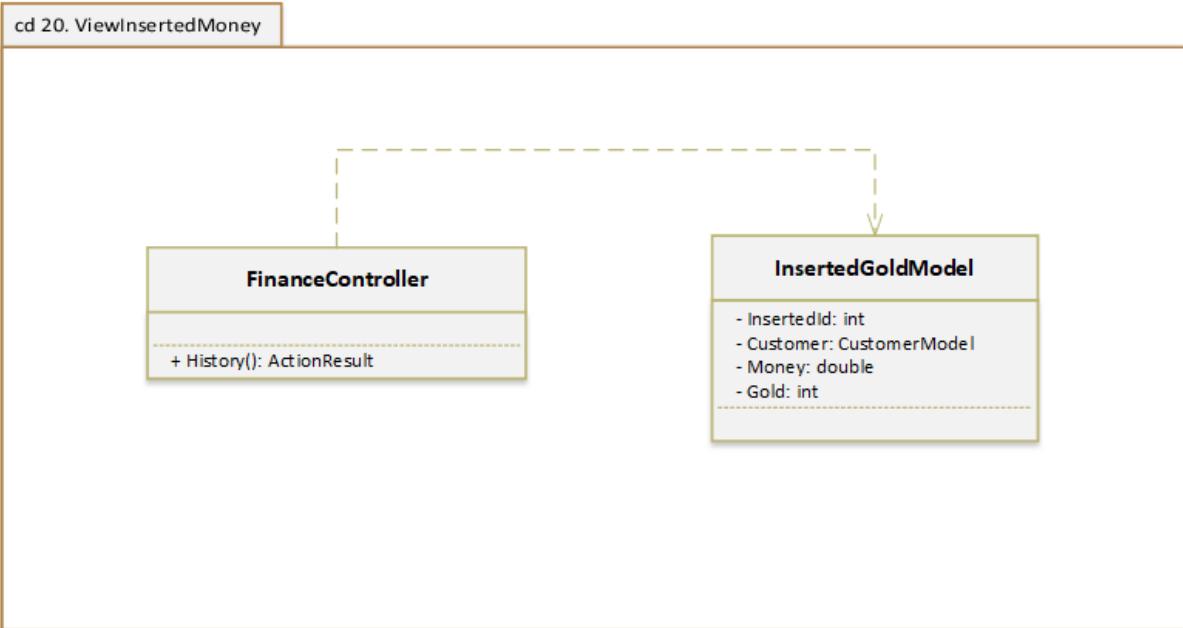


Figure 4-82: View inserted money history Class diagram

##### Class Specification

FinanceController			
Physical address	MindDoctor / Controllers / FinanceController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
History			
Return Type	ActionResult		
Parameters	Name	Type	Description

InsertedGoldModel			
-------------------	--	--	--

Physical address	MindDoctor / Models / FinanceModels / FinanceModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	InsertedId	int	
2.	Customer	CustomerModel	
3.	Money	double	
4.	Gold	int	
Operation			

## Sequence Diagram

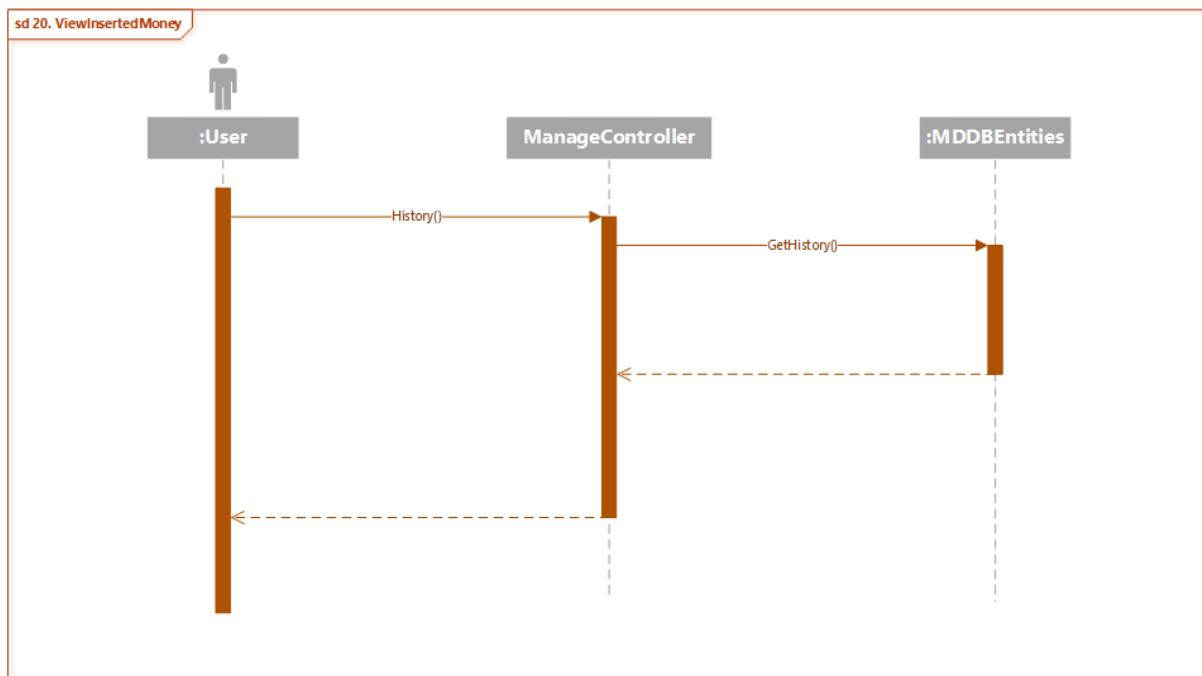


Figure 4-83: View inserted money history Sequence diagram

#### 4.3.4.21 Comment post

##### Screen Design

Thích Bình luận Chia sẻ  
589 người thích nội dung này

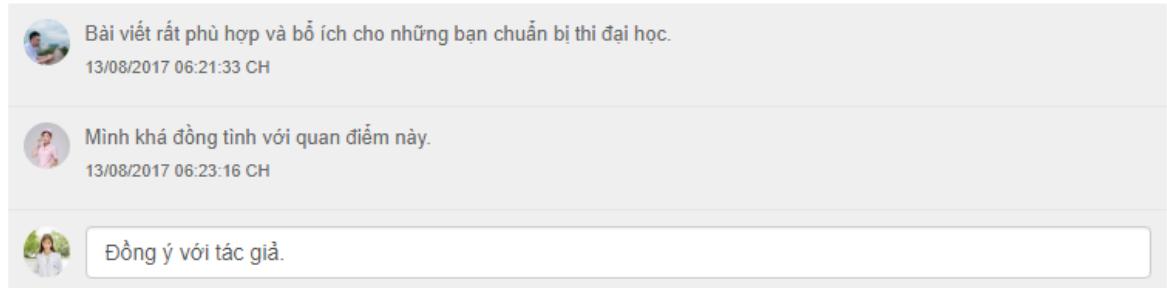


Figure 4-84: Comment post Screen design

##### Class Diagram

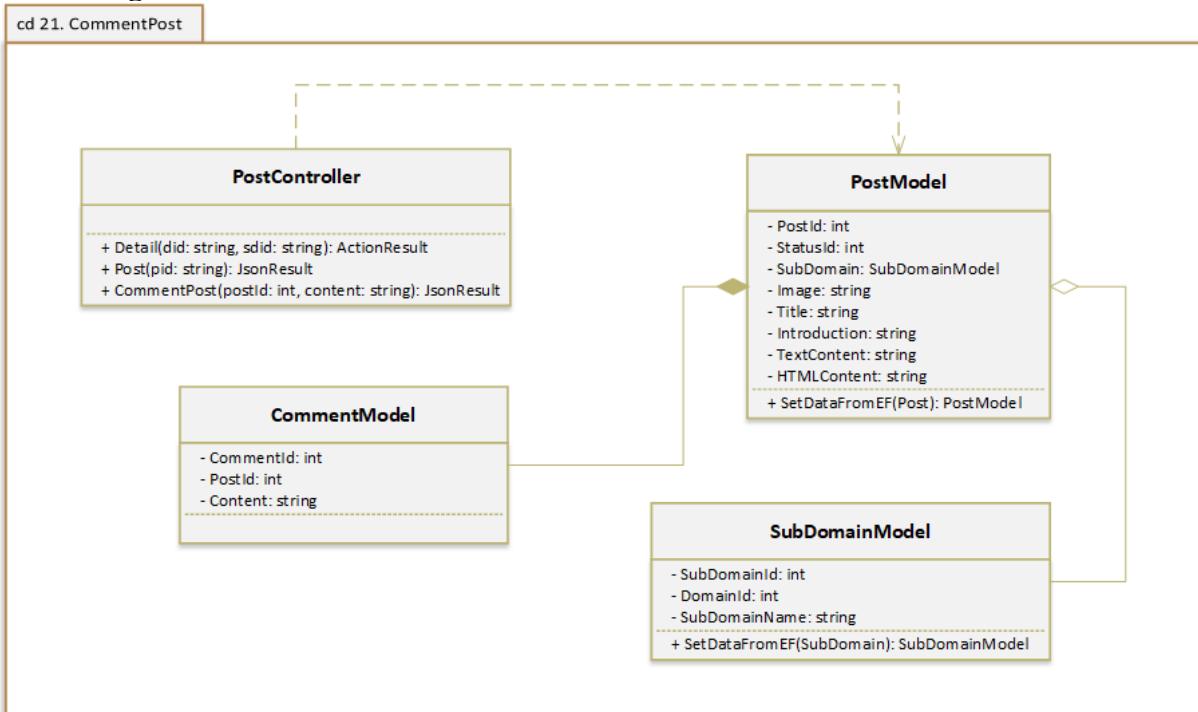


Figure 4-85: Comment post Class diagram

##### Class Specification

PostController			
Physical address	MindDoctor / Controllers / PostController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Detail			
Return Type	ActionResult		

Parameters	Name	Type	Description
	did	string	DomainId.
	sdid	string	Sub-domainId.
Post			
Return Type	ActionResult		
Parameters	Name	Type	Description
	pid	string	PostId.

<b>PostModel</b>			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Image	string	
5.	Title	string	
6.	Introduction	string	
7.	TextContent	string	
8.	HTMLContent	string	
Operation			
SetDataFromEF			
Return Type	PostModel		
Parameters	Name	Type	Description
	post	Post	

<b>SubDomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			
SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	

CommentModel			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	CommentId	int	
2.	PostId	int	
3.	Content	string	
Operation			

## Sequence Diagram

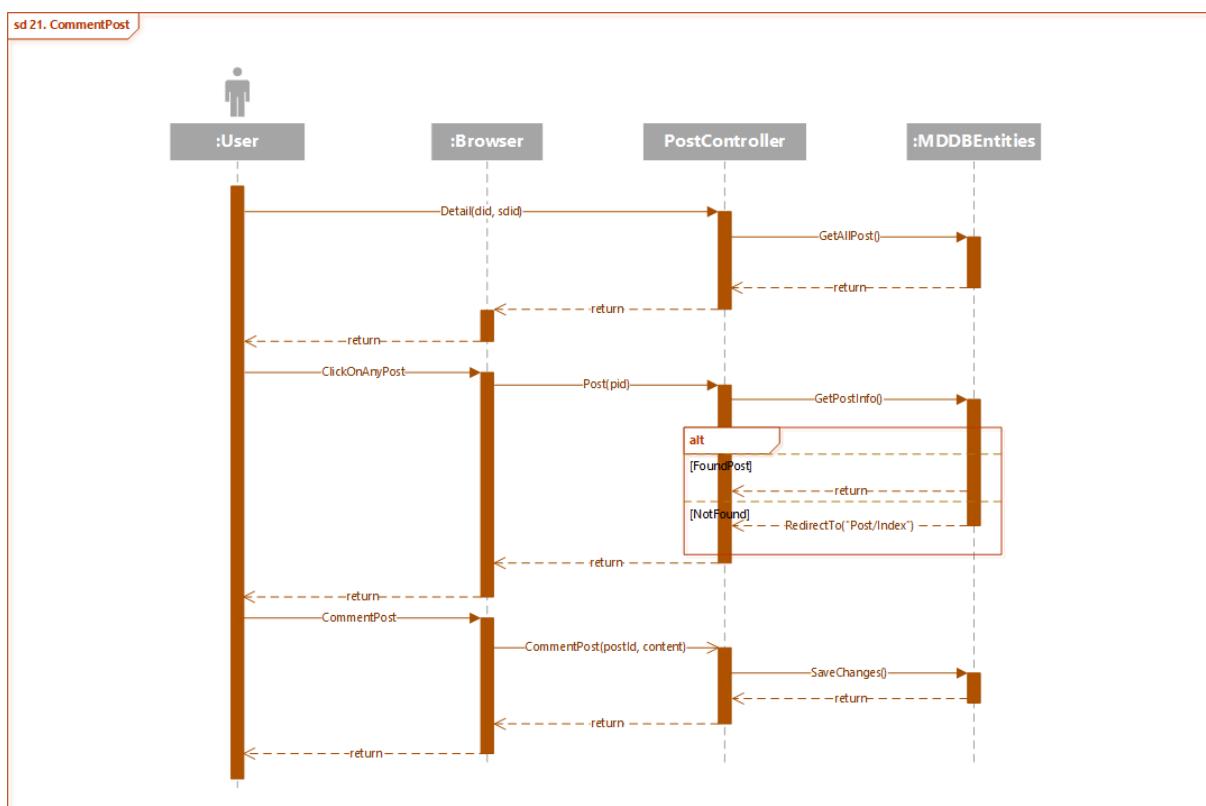


Figure 4-86: Comment post Sequence diagram

#### 4.3.4.22 Like post

##### Screen Design

Thích Bình luận Chia sẻ  
589 người thích nội dung này

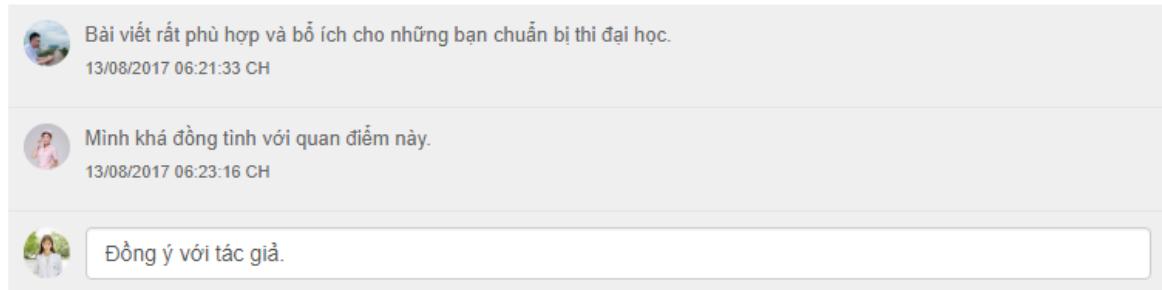


Figure 4-87: Like post Screen design

##### Class Diagram

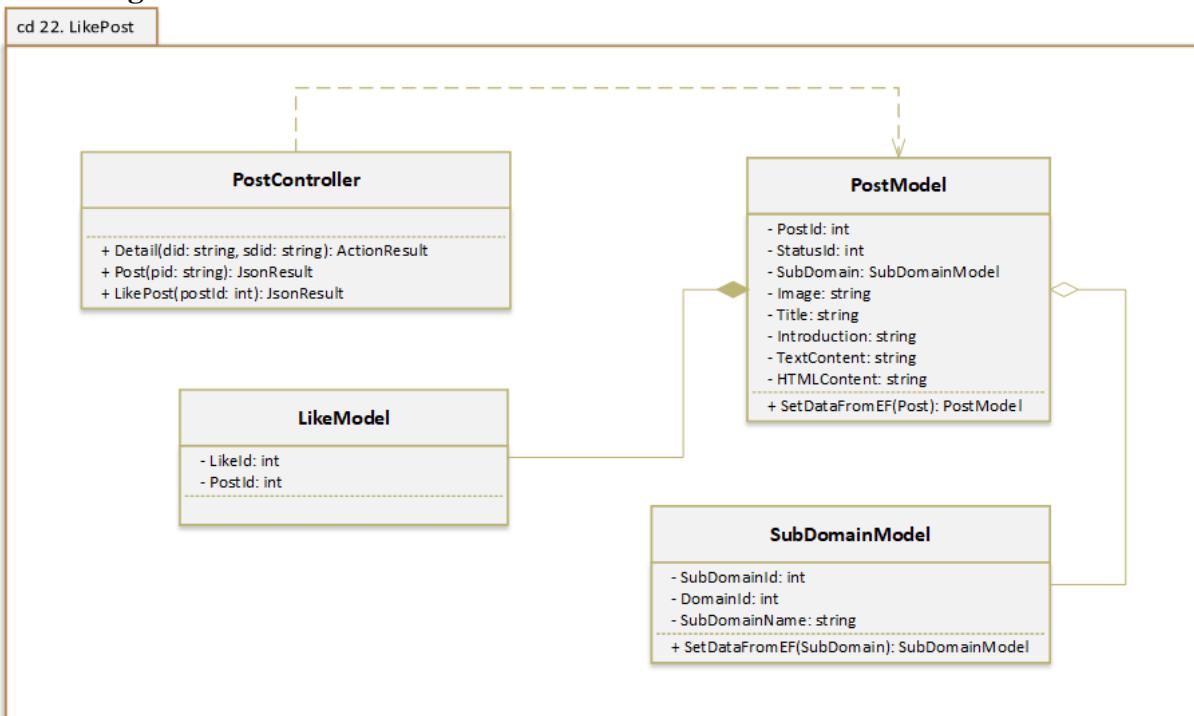


Figure 4-88: Like post Class diagram

##### Class Specification

PostController			
Physical address	MindDoctor / Controllers / PostController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Detail			
Return Type	ActionResult		
Parameters	Name	Type	Description

	did	string	DomainId.
	sdid	string	Sub-domainId.
Post			
Return Type	ActionResult		
Parameters	Name	Type	Description
	pid	string	PostId.

<b>PostModel</b>			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Image	string	
5.	Title	string	
6.	Introduction	string	
7.	TextContent	string	
8.	HTMLContent	string	
Operation			
SetDataFromEF			
Return Type	PostModel		
Parameters	Name	Type	Description
	post	Post	

<b>SubDomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			
SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	

LikeModel			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	LikeId	int	
2.	PostId	int	
Operation			

### Sequence Diagram

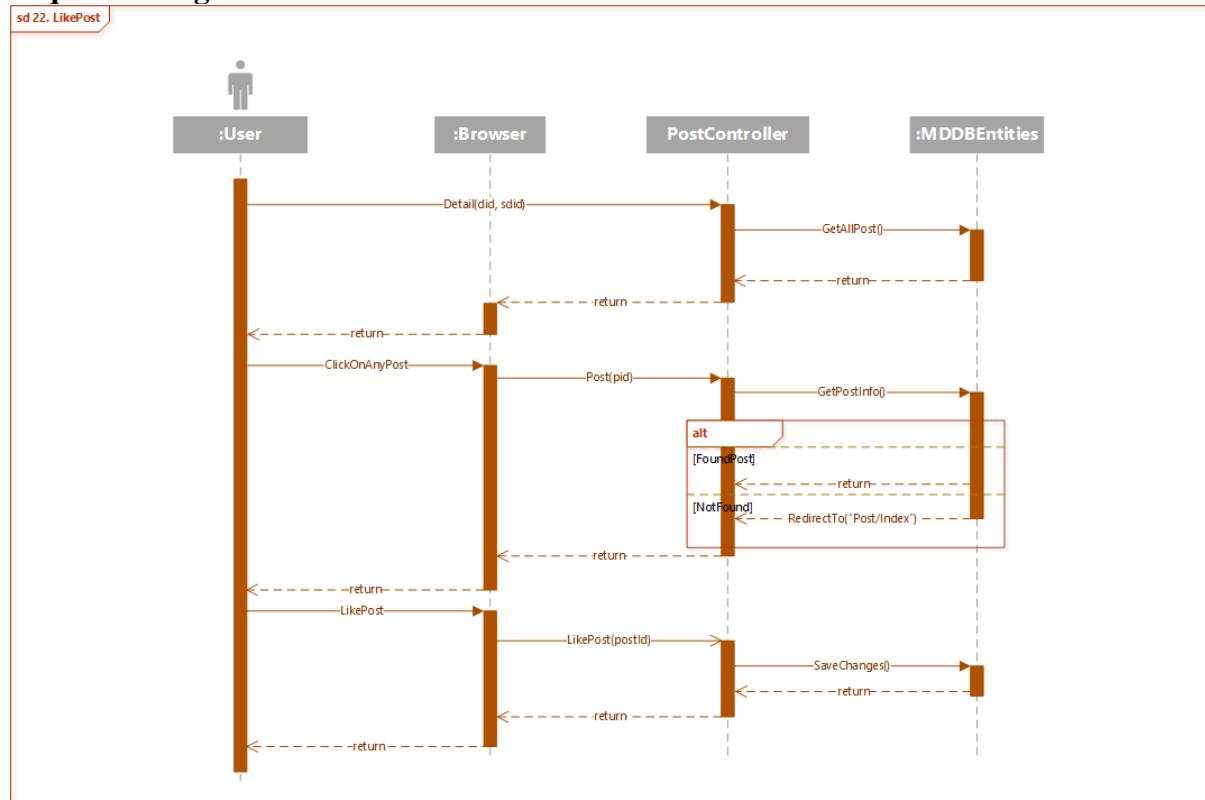


Figure 4-89: Like post Sequence diagram

#### 4.3.4.23 Unlike post

##### Screen Design

Thích Bình luận Chia sẻ  
 983 người thích nội dung này

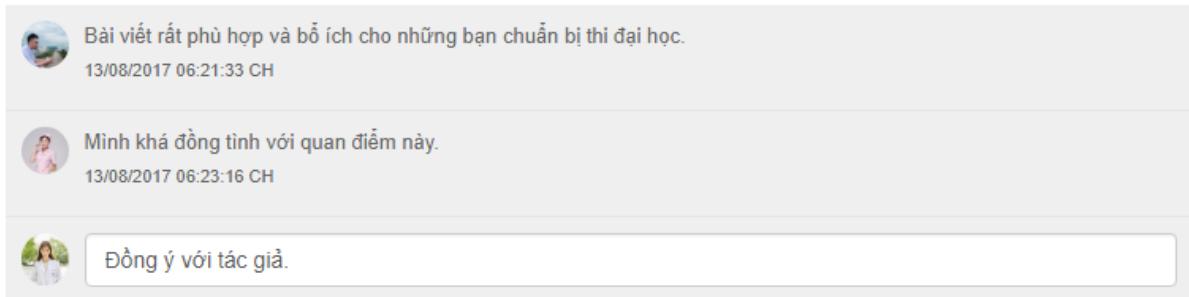


Figure 4-90: Unlike post Screen design

##### Class Diagram

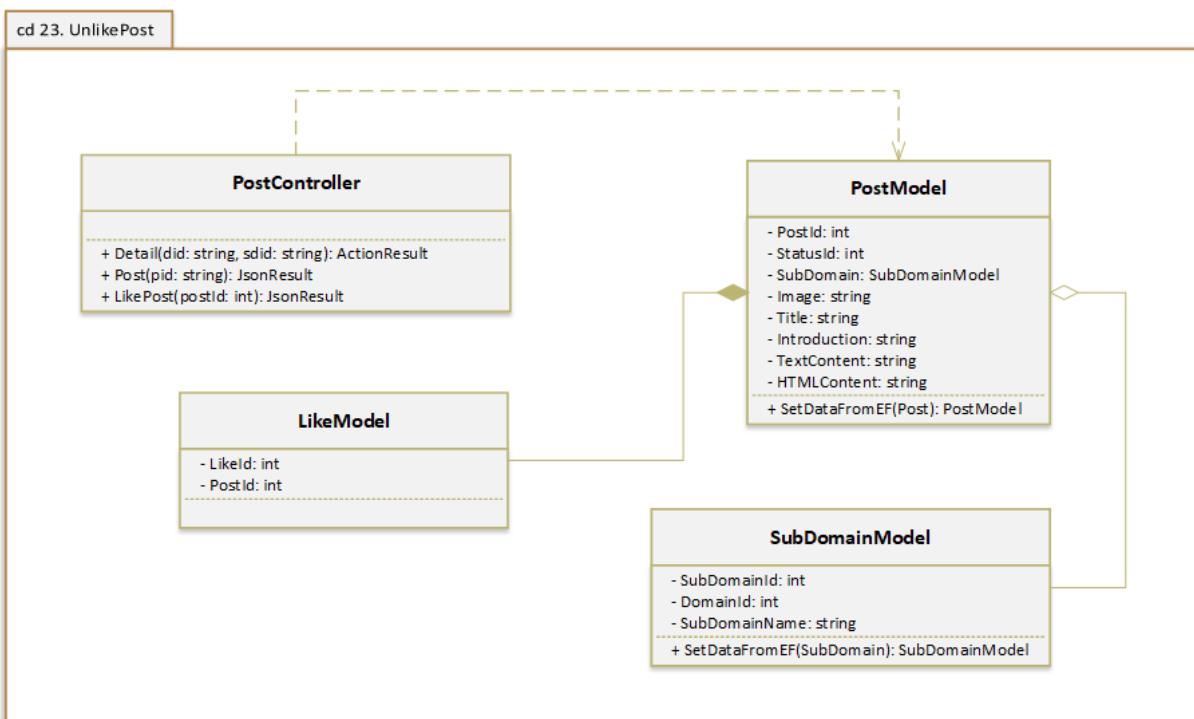


Figure 4-91: Unlike post Class diagram

##### Class Specification

PostController			
Physical address	MindDoctor / Controllers / PostController.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Detail			
Return Type	ActionResult		
Parameters	Name	Type	Description

	did	string	DomainId.
	sdid	string	Sub-domainId.
Post			
Return Type	ActionResult		
Parameters	Name	Type	Description
	postId	string	PostId.

<b>PostModel</b>			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	StatusId	int	
3.	SubDomain	SubDomainModel	
4.	Image	string	
5.	Title	string	
6.	Introduction	string	
7.	TextContent	string	
8.	HTMLContent	string	
Operation			
SetDataFromEF			
Return Type	PostModel		
Parameters	Name	Type	Description
	post	Post	

<b>SubDomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			
SetDataFromEF			
Return Type	SubDomainModel		
Parameters	Name	Type	Description
	subDomain	SubDomain	

LikeModel			
Physical address	MindDoctor / Models / DomainModels / PostModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	LikeId	int	
2.	PostId	int	
Operation			

## Sequence Diagram

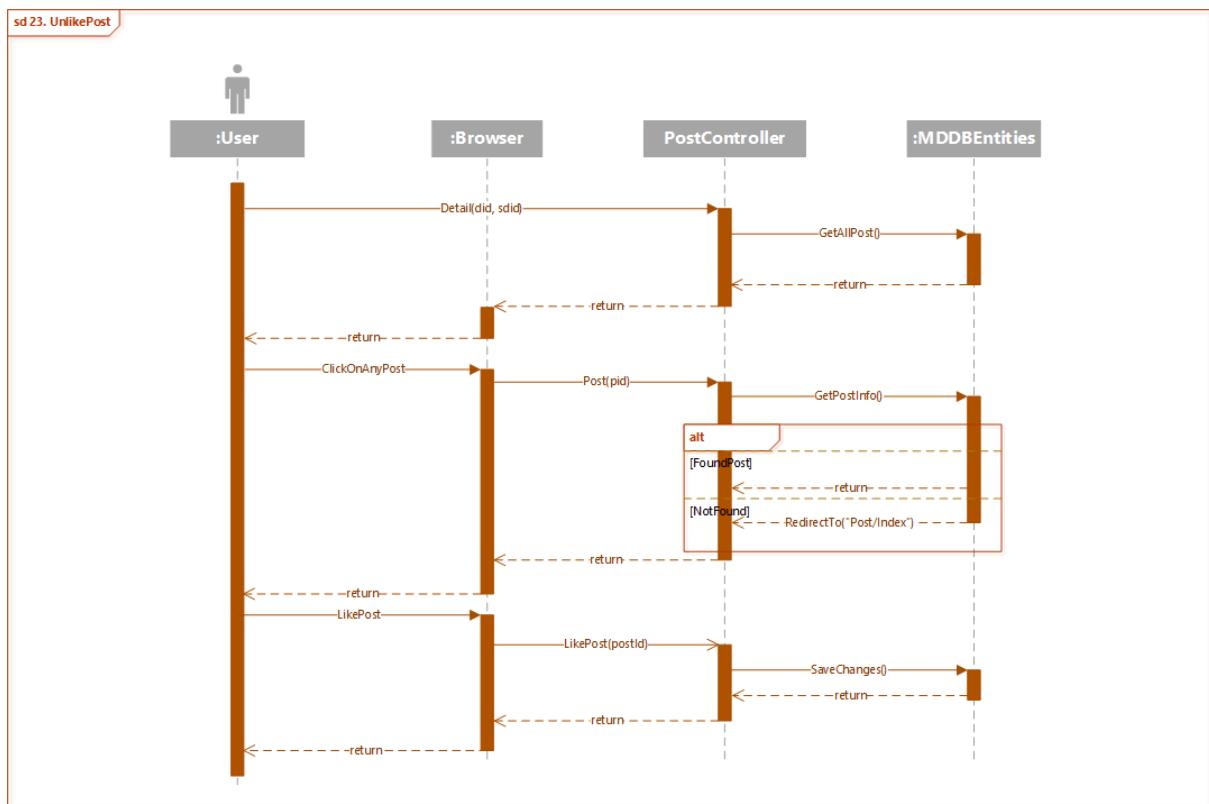


Figure 4-92: Unlike post Sequence diagram

#### 4.3.4.24 Connect to Advisor

##### Screen Design

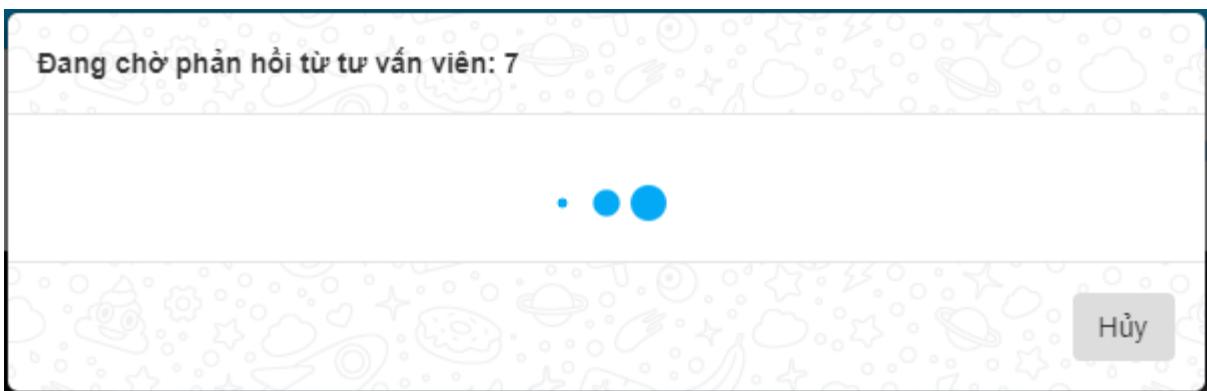


Figure 4-93: Connect to Advisor Screen design

##### Class Diagram

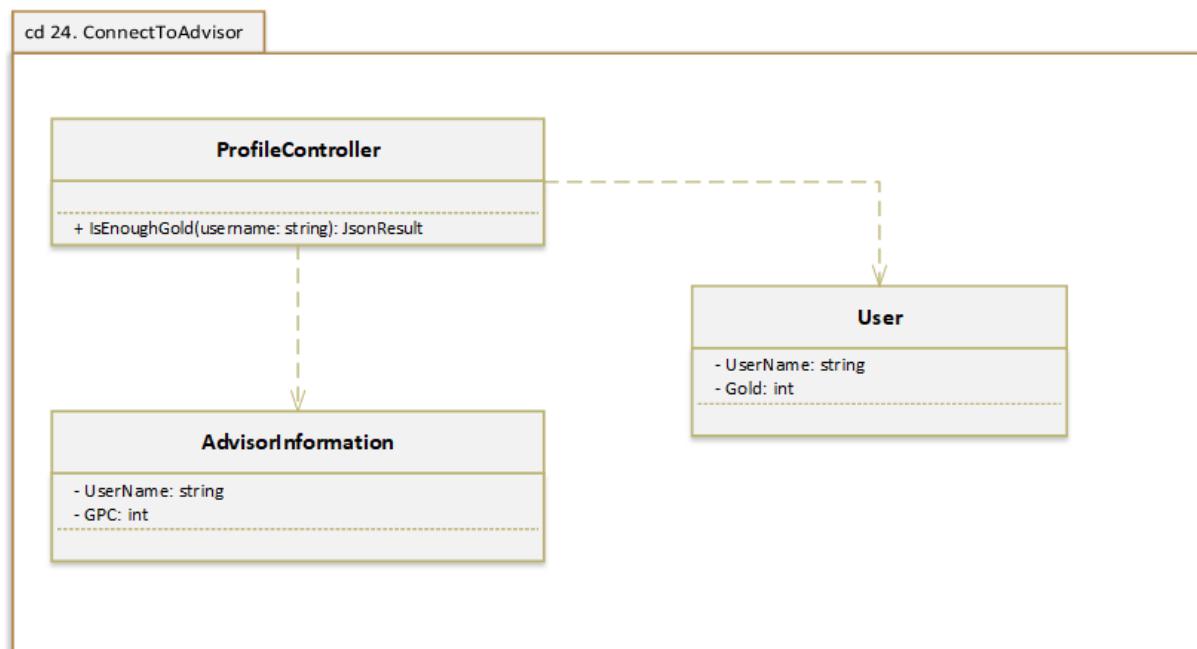


Figure 4-94: Connect to Advisor Class diagram

##### Class Specification

ProfileController			
Physical address	MindDoctor / Areas / Customer / Controllers / ProfileController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
IsEnoughGold			
Return Type	JsonResult		
Parameters	Name	Type	Description

	username	string	
--	----------	--------	--

AdvisorInformation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AdvisorInformation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	GPC	int	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	Gold	int	
Operation			

## Sequence Diagram

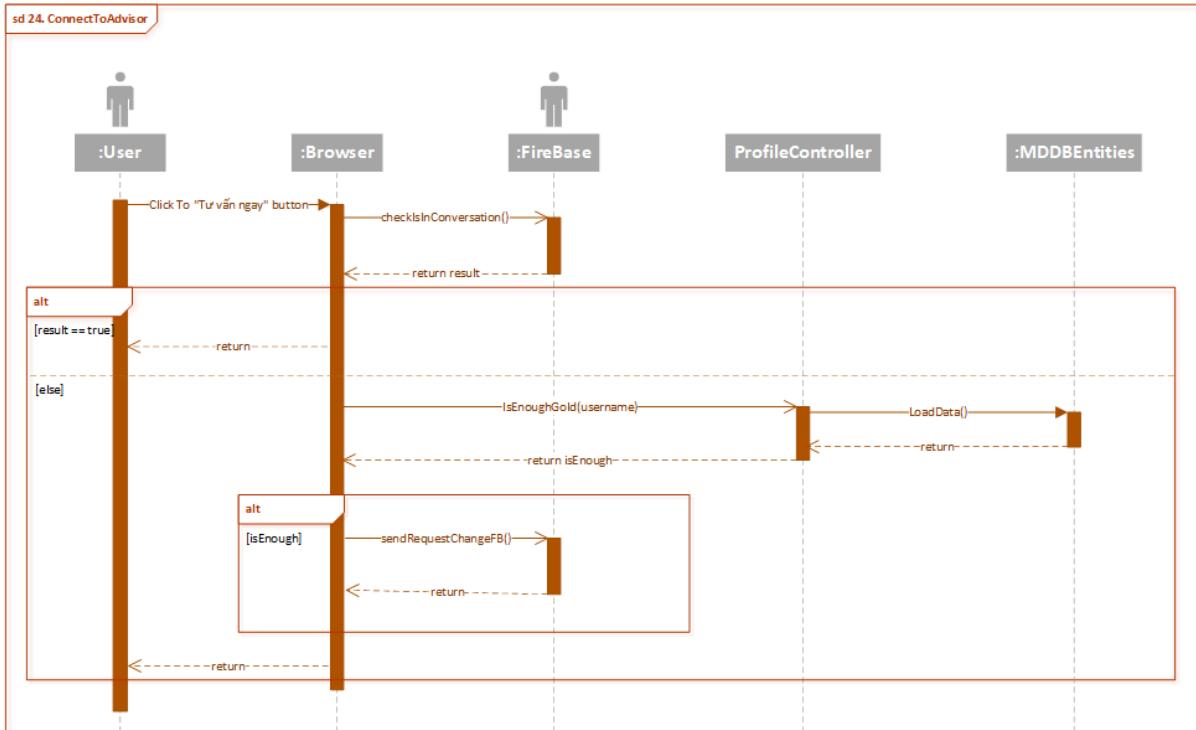


Figure 4-95: Connect to Advisor Sequence diagram

#### 4.3.4.25 Chat text

##### Screen Design

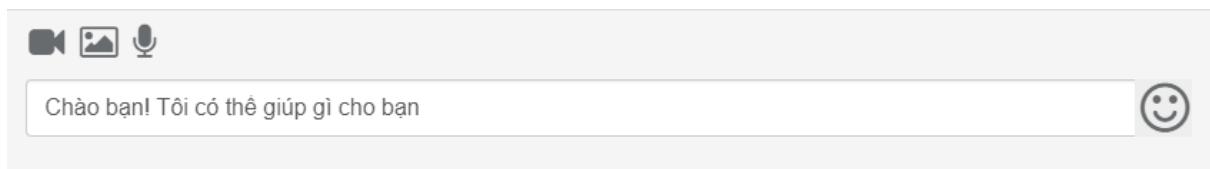


Figure 4-96: Chat text Screen design

##### Class Diagram

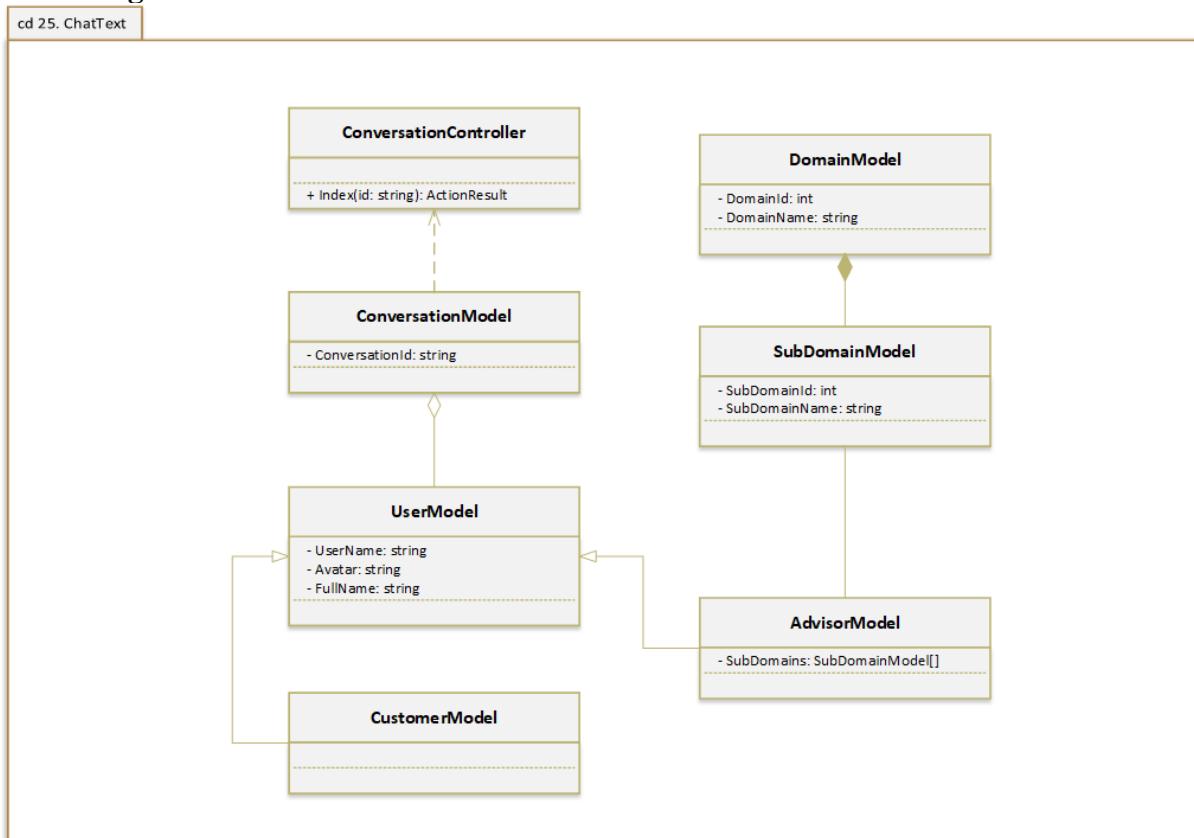


Figure 4-97: Chat text Class diagram

##### Class Specification

ConversationController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		
Parameters	Name	Type	Description
	id	string	

ConversationModel			
Physical address	MindDoctor / Models / ConversationModels / ConversationModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	string	
Operation			

CustomerModel			
Physical address	MindDoctor / Models / UserModels / CustomerModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	Avatar	string	
3.	FullName	string	
Operation			

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	About	string	
4.	Degree	string	
5.	AvgRate	double	
6.	Avatar	string	
7.	WorkPlace	string	
Operation			

DomainModel			
Physical address	MindDoctor / Models / DomainModels / DomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainName	string	

## Operation

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

## Sequence Diagram

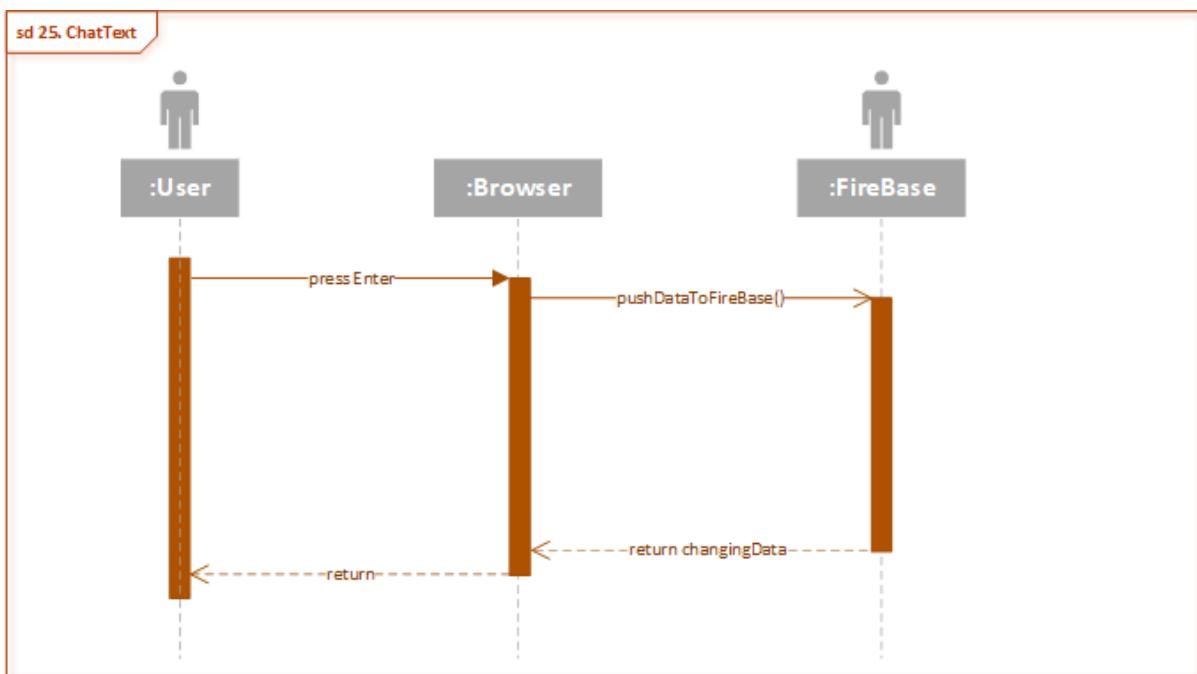


Figure 4-98: Chat text Sequence diagram

#### 4.3.4.26 Start video call

##### Screen Design

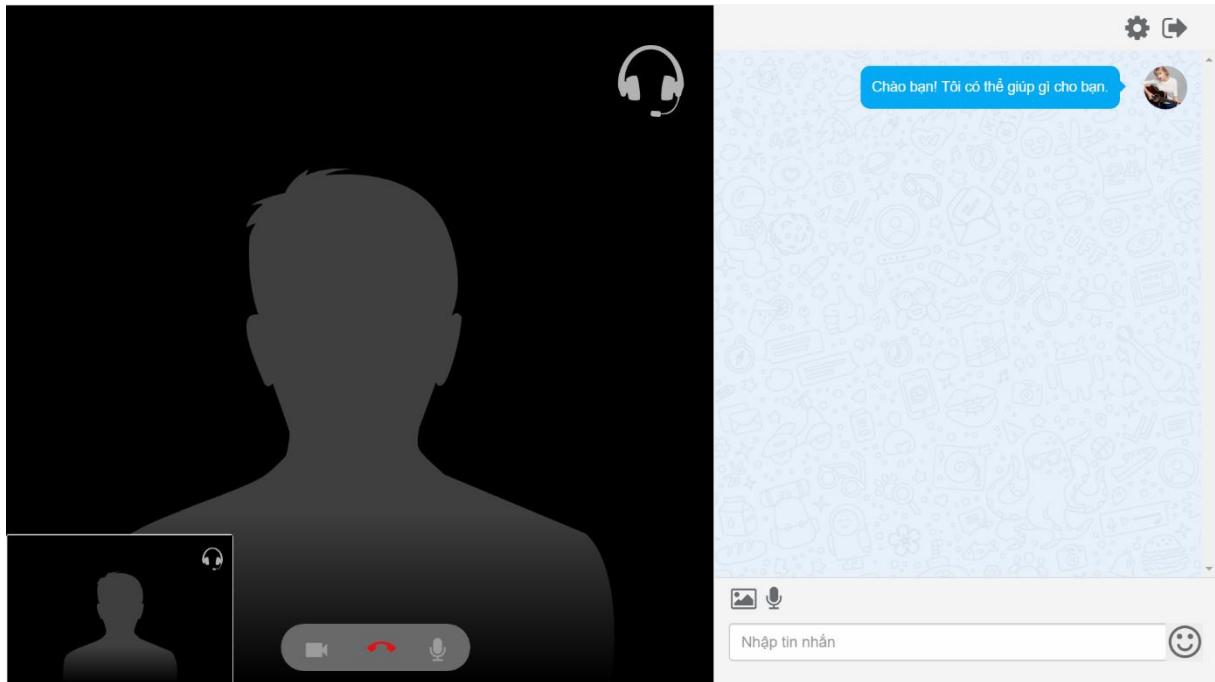


Figure 4-99: Start video call Screen design

##### Class Diagram

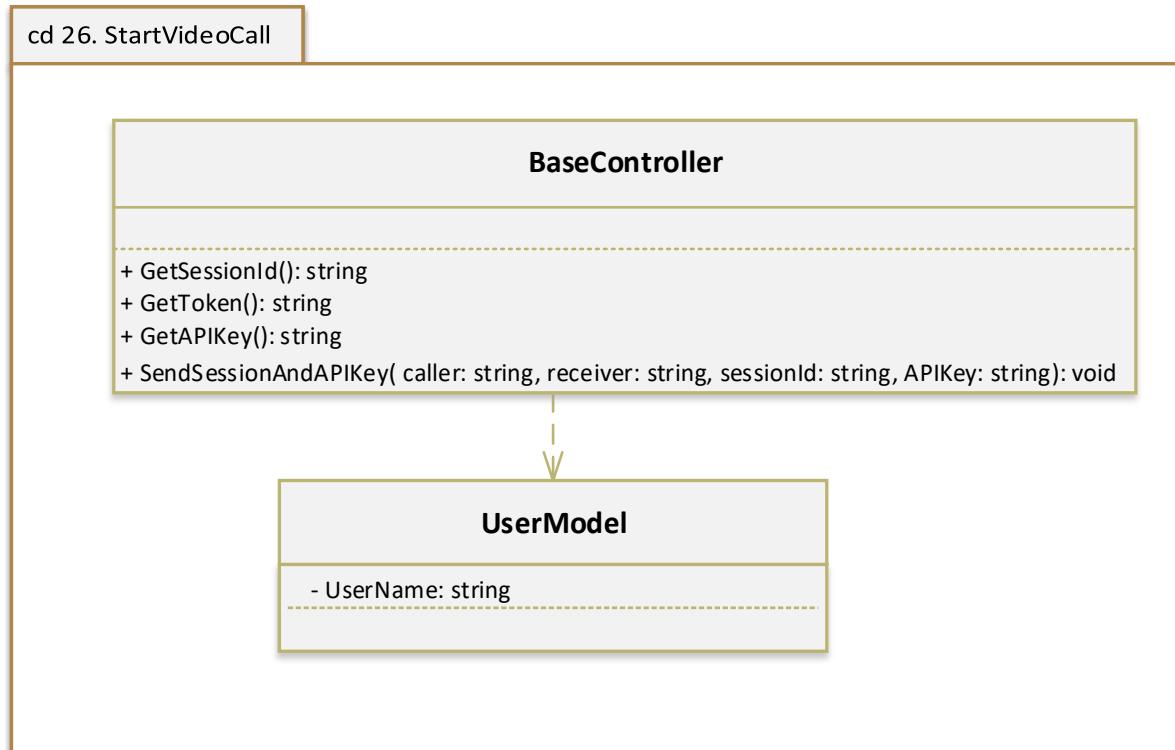


Figure 4-100: Start video call Class diagram

##### Class Specification

BaseController

Physical address	MindDoctor / Controllers / BaseController.cs		
Base class			
Attributes			
	Name	Type	Description
Operation			
<b>GetSessionId</b>			
Return Type	string		
Parameters	Name	Type	Description
<b>GetToken</b>			
Return Type	string		
Parameters	Name	Type	Description
<b>GetAPIKey</b>			
Return Type	int		
Parameters	Name	Type	Description
<b>SendSessionAndAPIKey</b>			
Return Type	void		
Parameters	Name	Type	Description
	caller	string	Username of caller who start video call
	receiver	string	Username of receiver who receive the call
	sessionId	string	Id of session for video call
	APIKey	int	APIKey of Opentok framework

<b>UserModel</b>			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
Operation			

## Sequence Diagram

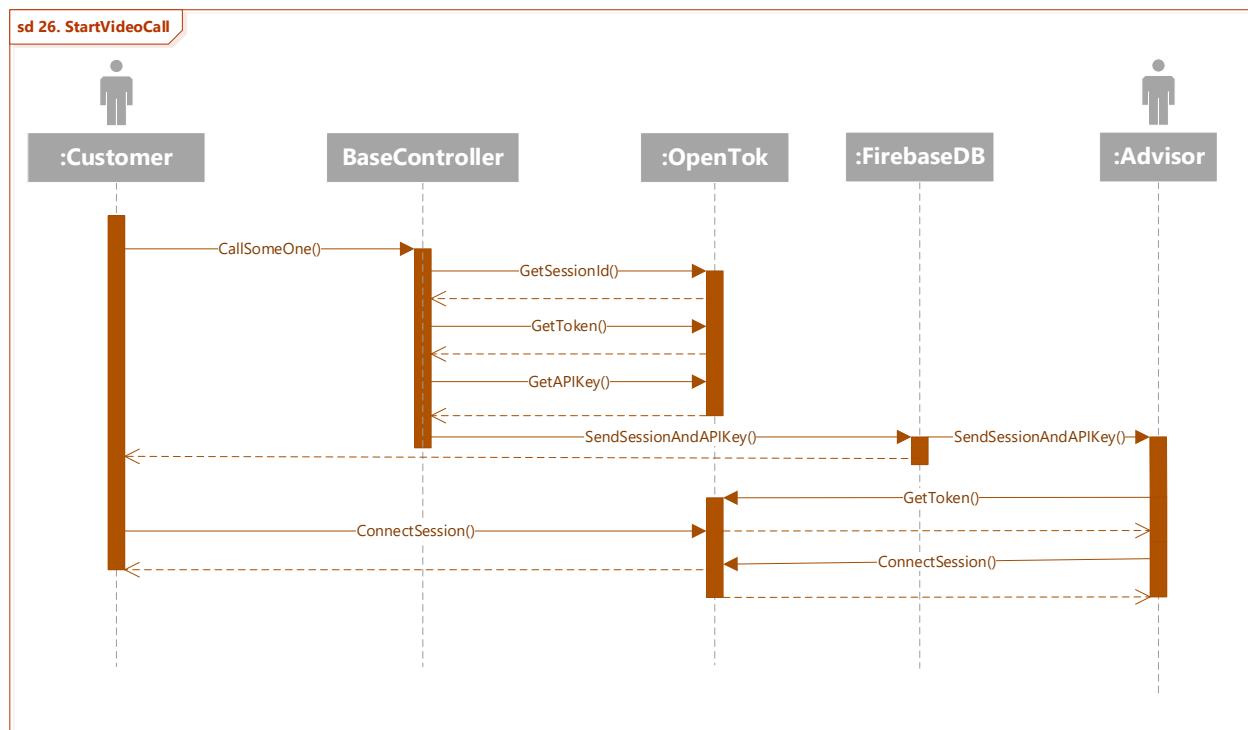


Figure 4-101: Start video call Sequence diagram

### Implementation

Using Opentok framework to create session, get token for session. To implement, using the sample code from Opentok guide (<https://tokbox.com/developer/sdks/dot-net/>) follow steps:

+ Server side:

- Step 1: Initializing:

```

String ApiKey = "Your Api Key"
String ApiSecret = "Your Api Secret"
OpenTok = new OpenTok(ApiKey, ApiSecret)

```

- Step 2: To get sessionId:

```

// Create a session that will attempt to transmit streams directly
Session session = OpenTok.CreateSession()
// Store this sessionId to the database for later use
String sessionId = session.Id

```

- Step 3: To get token for session:

```

// Generate a token from a sessionId
String token = OpenTok.GenerateToken(sessionId)

```

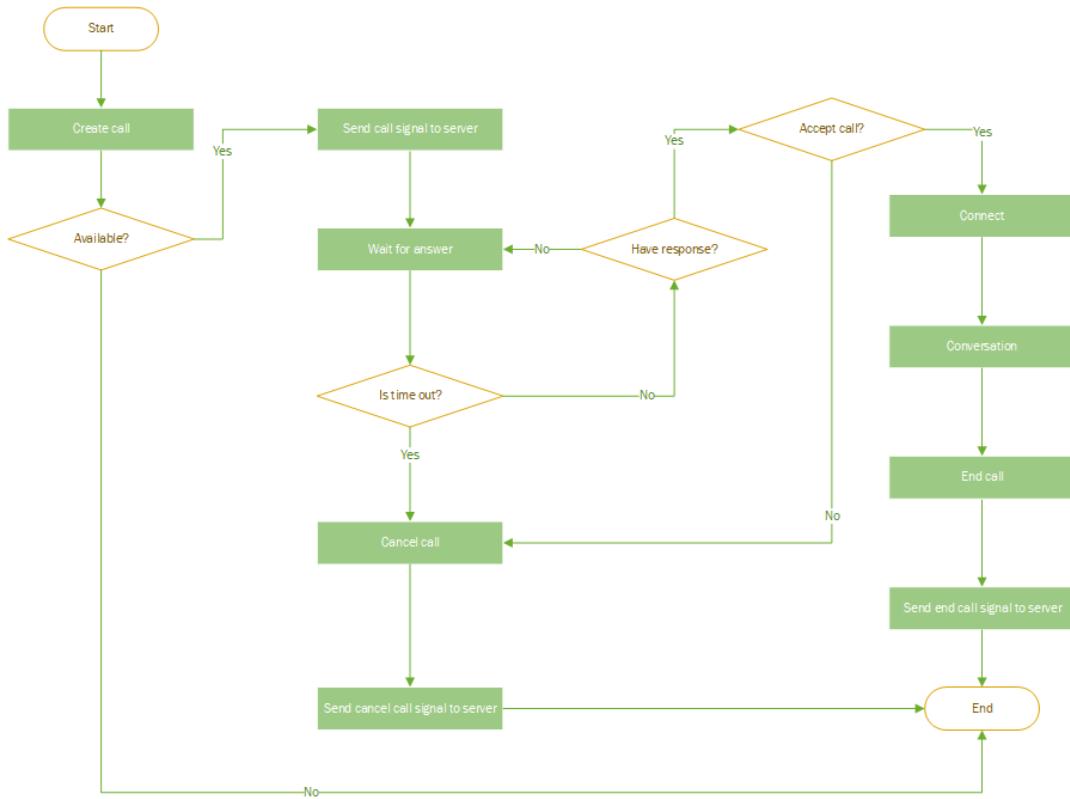
+ Client side:

Using javascript to init session and connect session after have sessionId and token for session:

- Step 1: Import Opentok

```
<script src="https://static.opentok.com/v2/js/opentok.min.js"></script>
```

- Step 2: Create session, create stream and connect session  
‘subscriber’ , ‘publisher’ are id of video holder in front end



#### 4.3.4.27 End video call

##### Screen Design

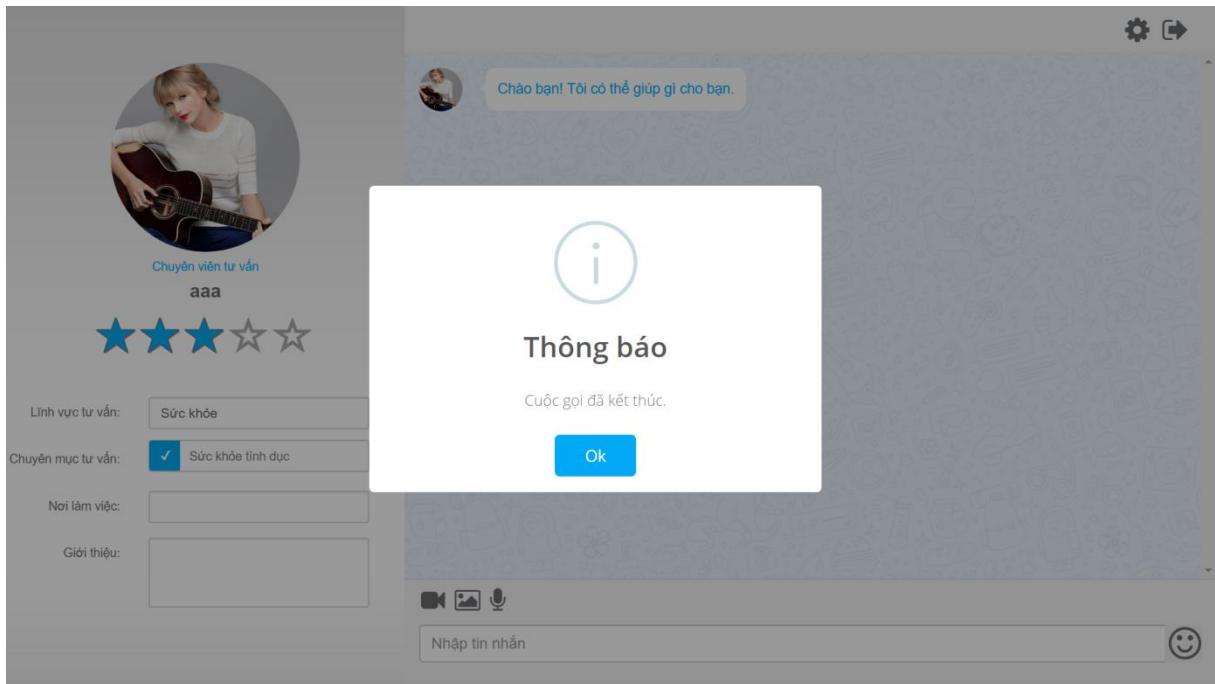


Figure 4-102: End video call Screen design

##### Class Diagram

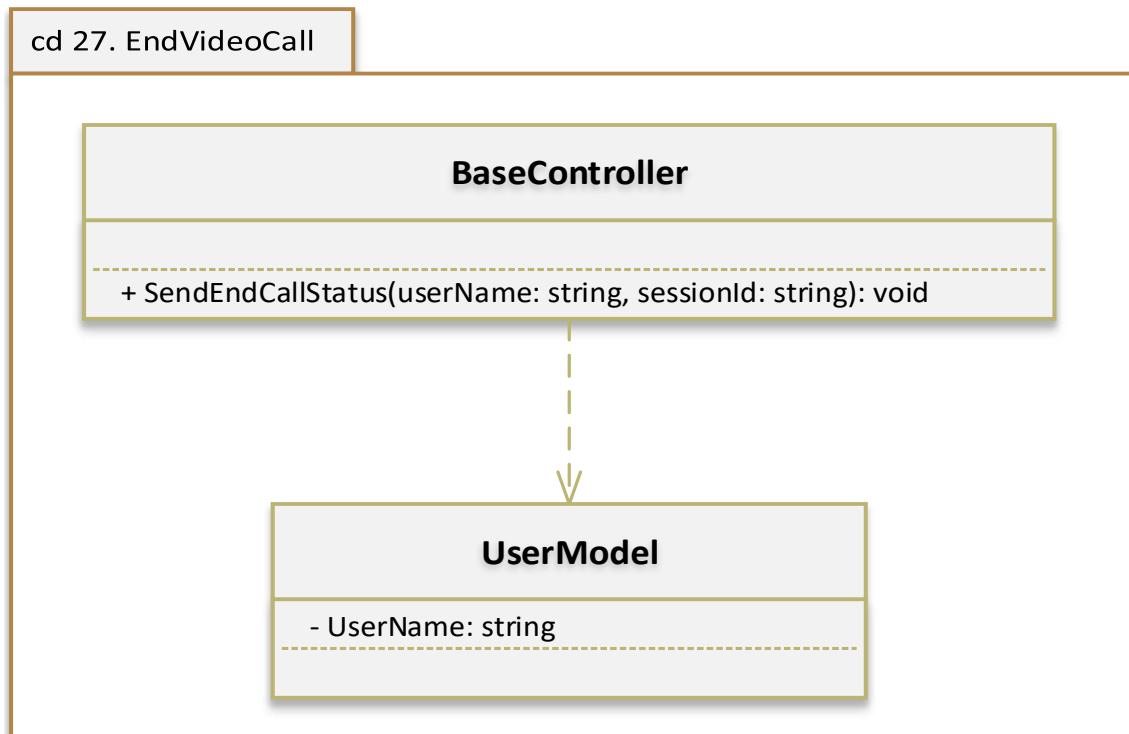


Figure 4-103: End video call Class diagram

## Class Specification

BaseController			
Physical address	MindDoctor / Controllers / BaseController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
GetOnlineAdvisorList			
Return Type	void		
Parameters	Name	Type	Description
	username	string	Username of client who end the call
	sessionId	string	Id of the video call session

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
Operation			

## Sequence Diagram

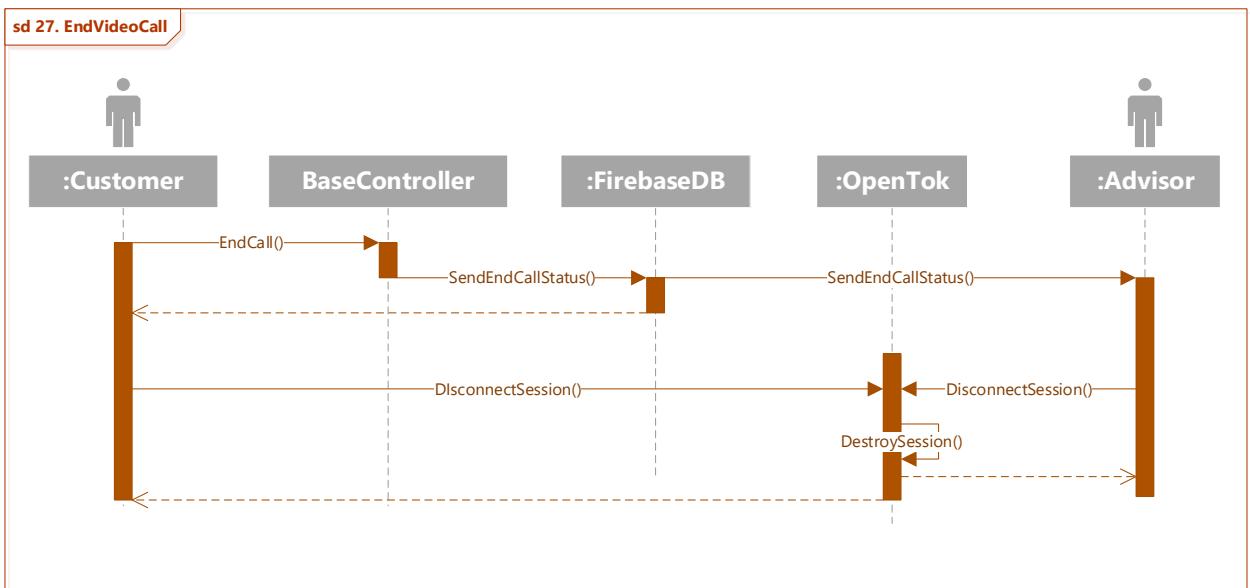
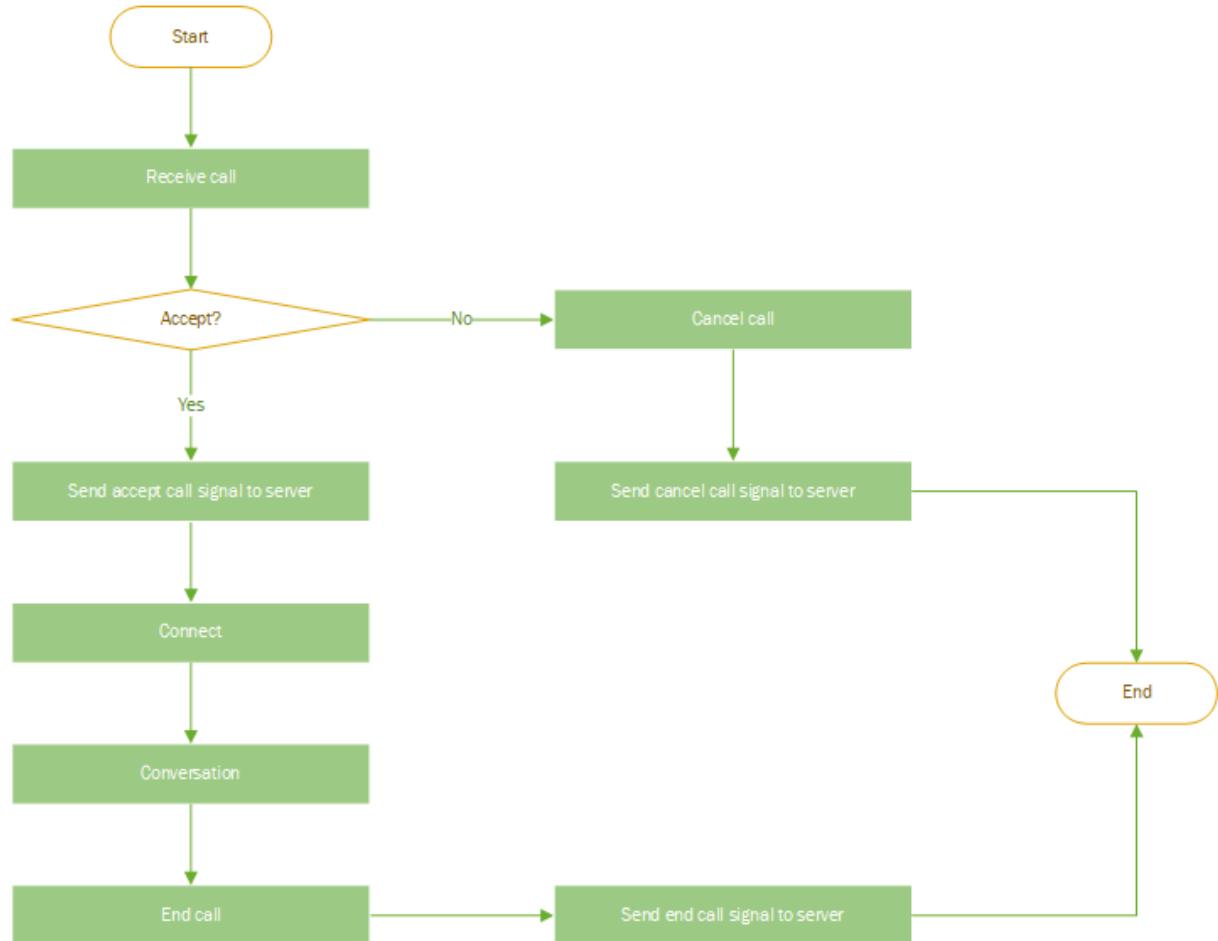


Figure 4-104: End video call Sequence diagram

## Implementation

Using Opentok in Client side to disconnect session to end video call. Opentok provide disconnect() function to disconnect session.

```
session.disconnect();
```



#### 4.3.4.28 Chat using speech to text

##### Screen Design

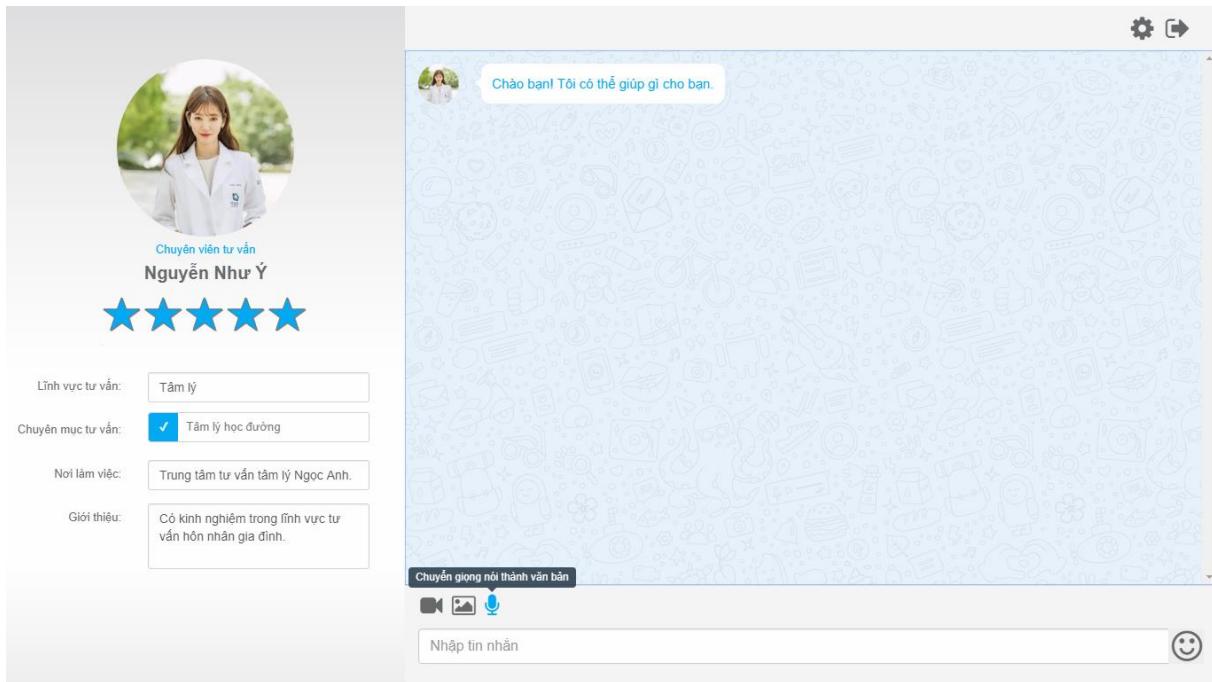


Figure 4-105: Chat using speech to text Screen design

##### Sequence Diagram

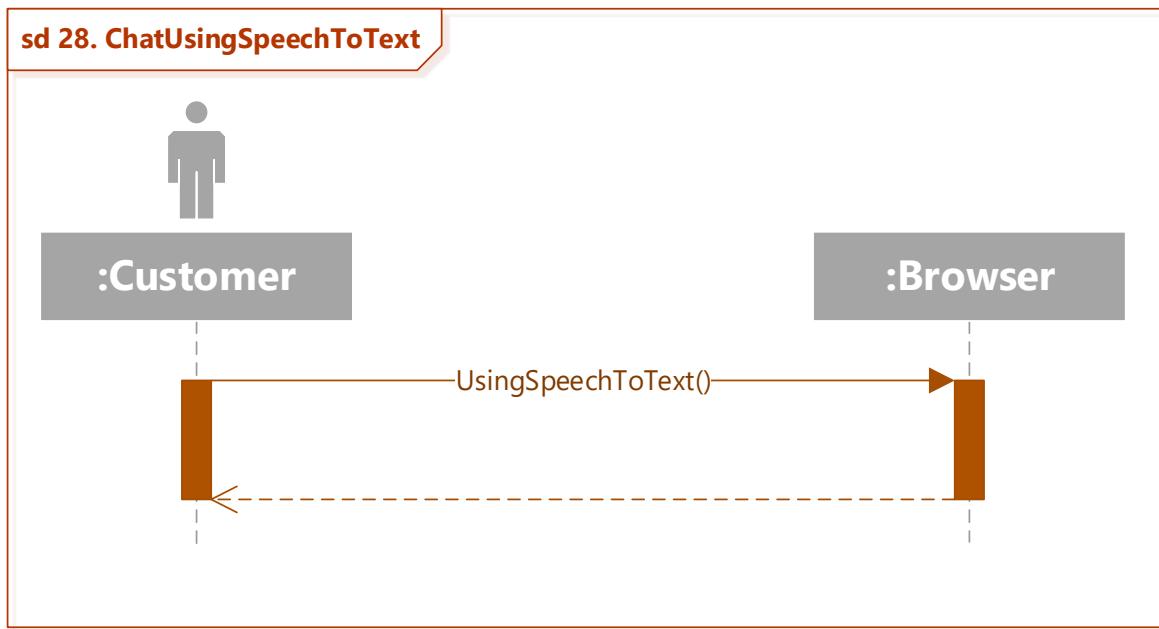


Figure 4-106: Chat using speech to text Sequence diagram

##### Implementation

Using javascript to access to browser's audio stream with **WebKitSpeechRecognition**.

#### 4.3.4.29 Tip money for Advisor

##### Screen Design

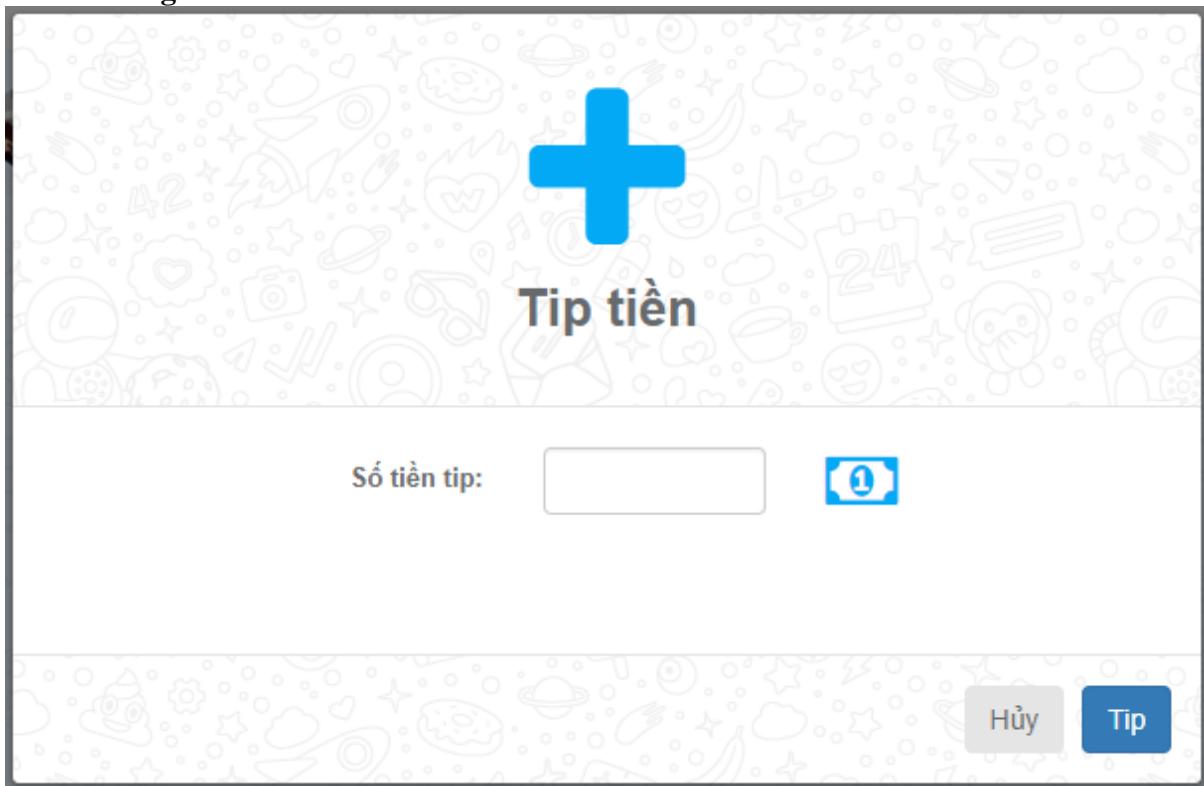


Figure 4-107: Tip money for Advisor Screen design

##### Class Diagram

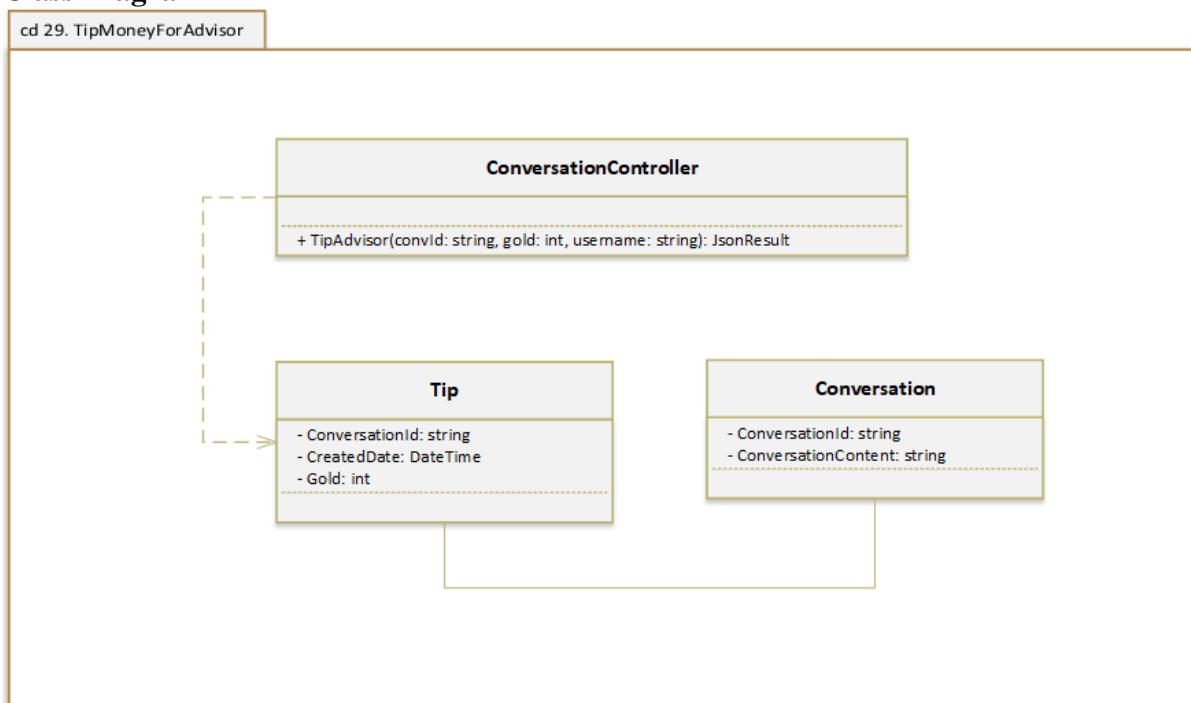


Figure 4-108: Tip money for Advisor Class diagram

##### Class Specification

ConversationController	
Physical	MindDoctor / Areas / Customer / Controllers /

address	ConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
TipAdvisor			
Return Type	JsonResult		
Parameters	Name	Type	Description
	convId	string	
	gold	int	
	username	string	

Tip			
Physical address	MindDoctor / EntityFramework / Tip.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	string	
2.	CreatedDate	DateTime	
3.	Gold	int	
Operation			

Conversation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Conversation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	string	
2.	ConversationContent	string	
Operation			

## Sequence Diagram

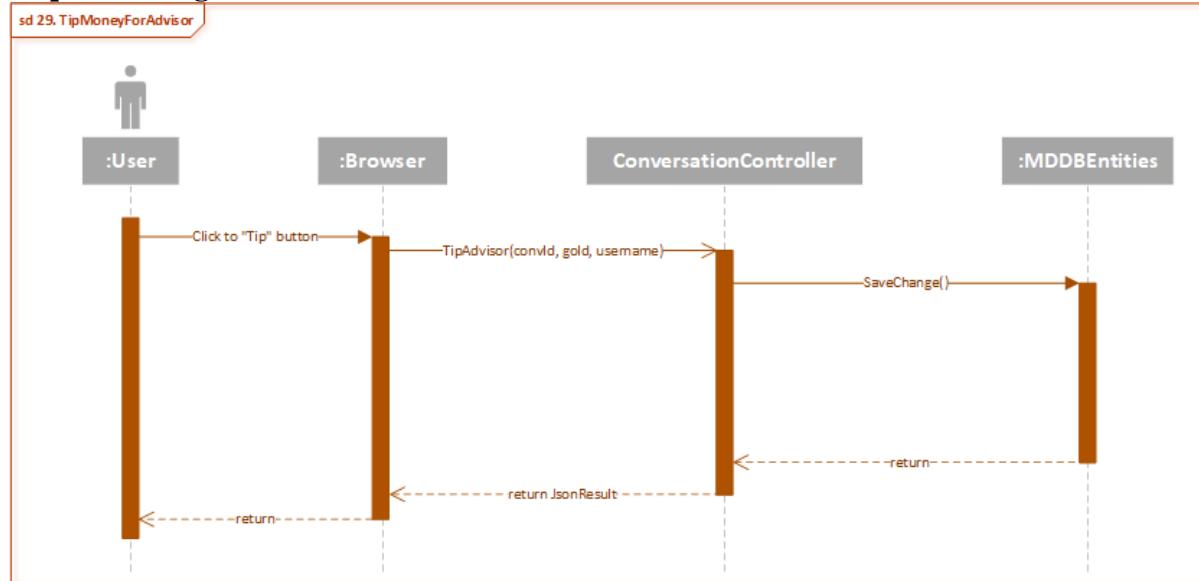


Figure 4-109: Tip money for Advisor Sequence diagram

#### 4.3.4.30 Rate Advisor after finishing conversation

##### Screen Design



Figure 4-110: Rate Advisor after finishing conversation Screen design

##### Class Diagram

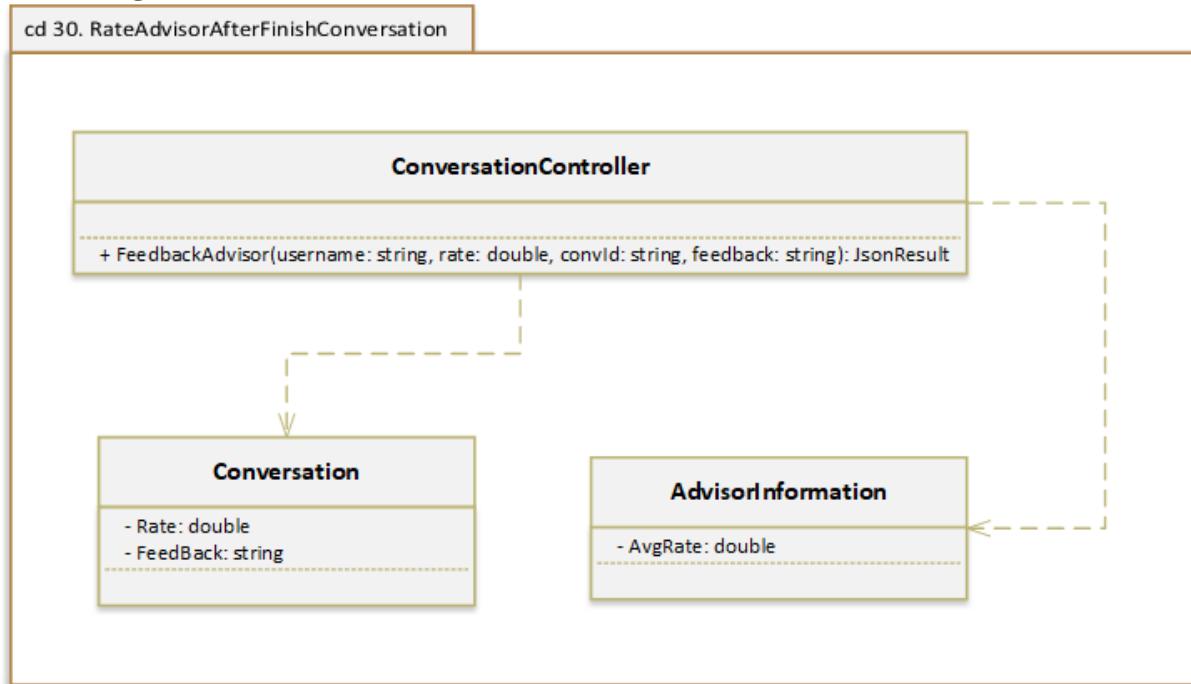


Figure 4-111: Rate Advisor after finishing conversation Class diagram

##### Class Specification

ConversationController	
Physical	MindDoctor / Areas / Customer / Controllers /

address	ConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
<b>FeedbackAdvisor</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	username	string	
	rate	double	
	convId	string	
	feedback	string	

Conversation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Conversation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Rate	double	
2.	FeedBack	string	
Operation			

AdvisorInformation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AdvisorInformation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	AvgRate	double	
Operation			

## Sequence Diagram

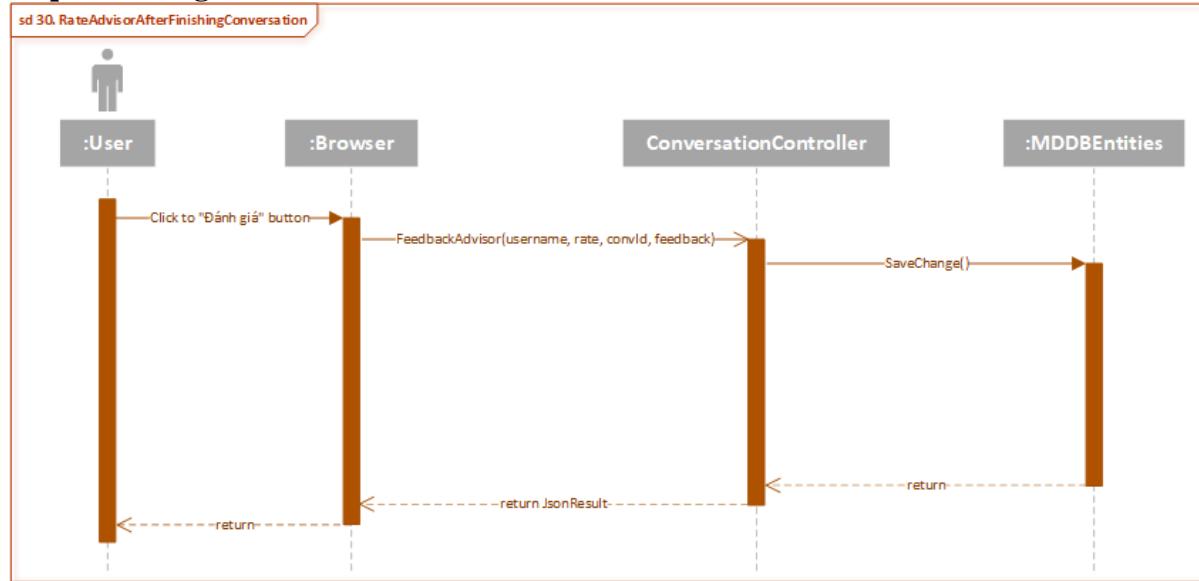


Figure 4-112: Rate Advisor after finishing conversation Sequence diagram

#### 4.3.4.31 Report Advisor while chatting

##### Screen Design

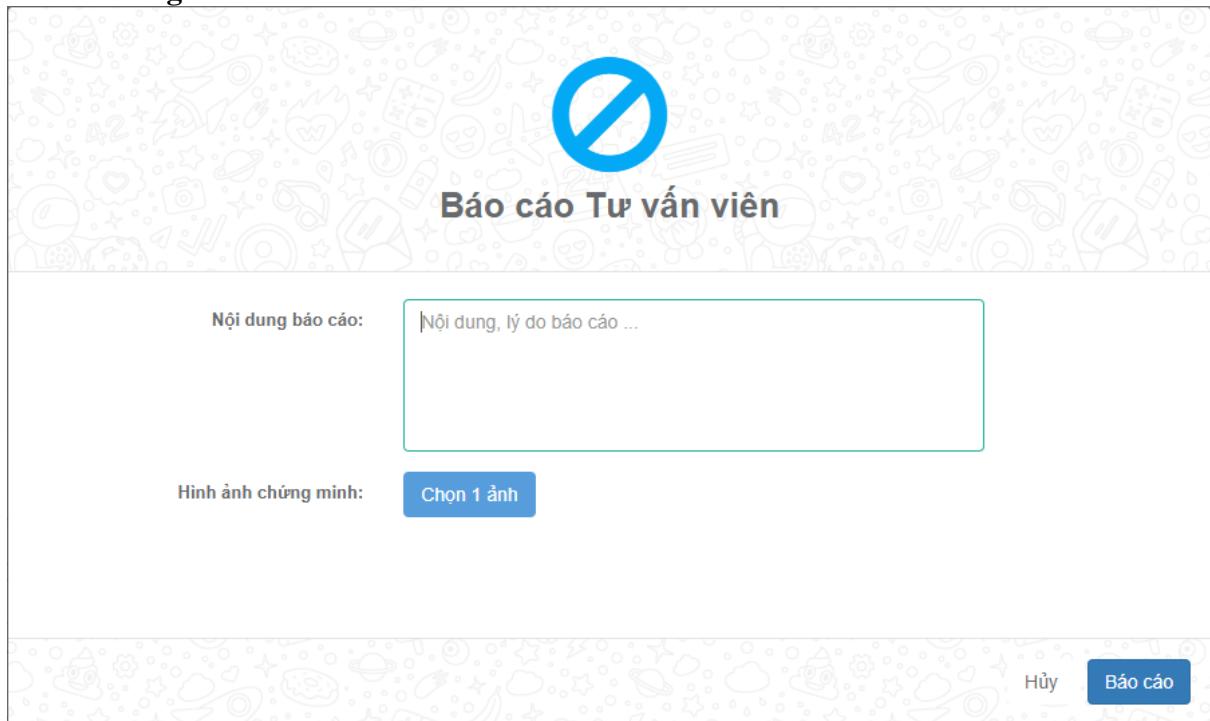


Figure 4-113: Report Advisor Screen design

##### Class Diagram

cd 31. ReportAdvisorWhileChatting

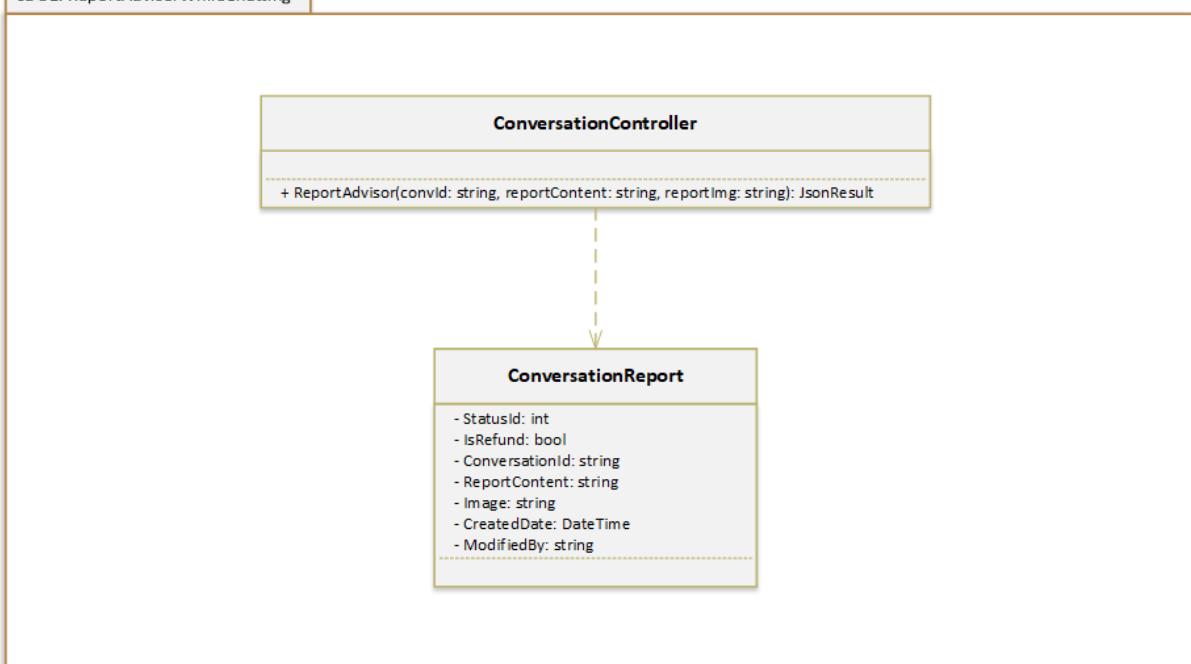


Figure 4-114: Report Advisor Class diagram

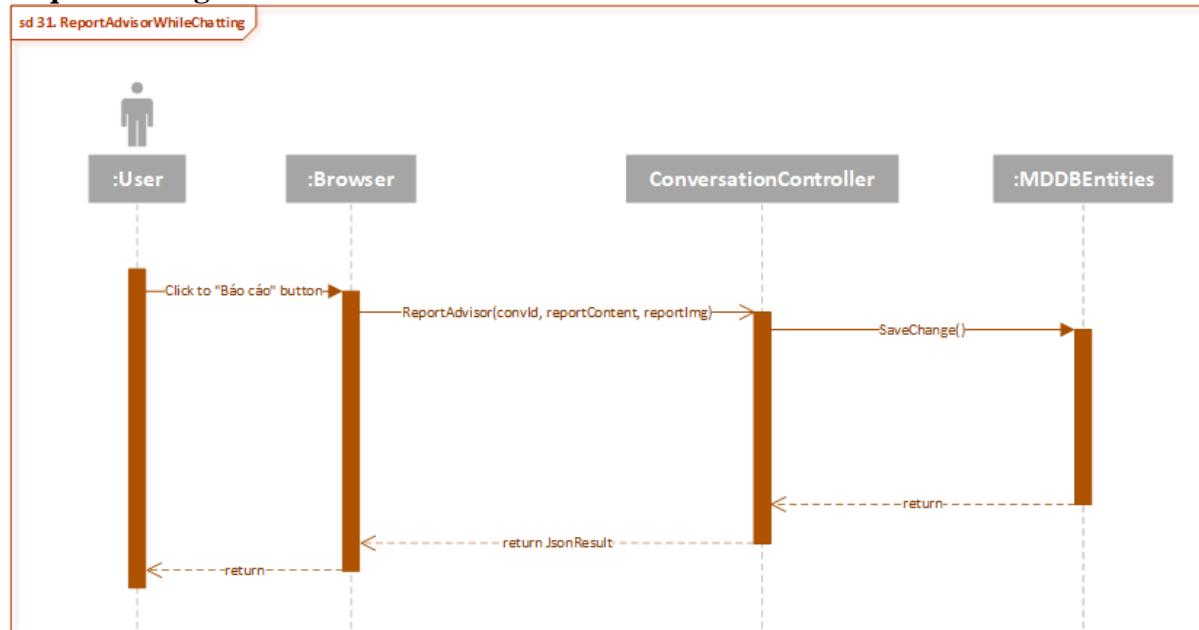
##### Class Specification

ConversationController	
Physical	MindDoctor / Areas / Customer / Controllers /

address	ConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
<b>ReportAdvisor</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	convId	string	
	reportContent	string	
	reportImg	string	

ConversationReport			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / ConversationReport.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	StatusId	int	
2.	IsRefund	bool	
3.	ConversationId	string	
4.	ReportContent	string	
5.	Image	string	
6.	CreatedDate	DateTime	
7.	ModifiedBy	string	
Operation			

## Sequence Diagram



*Figure 4-115: Report Advisor Sequence diagram*

#### 4.3.4.32 View conversation history

##### Screen Design

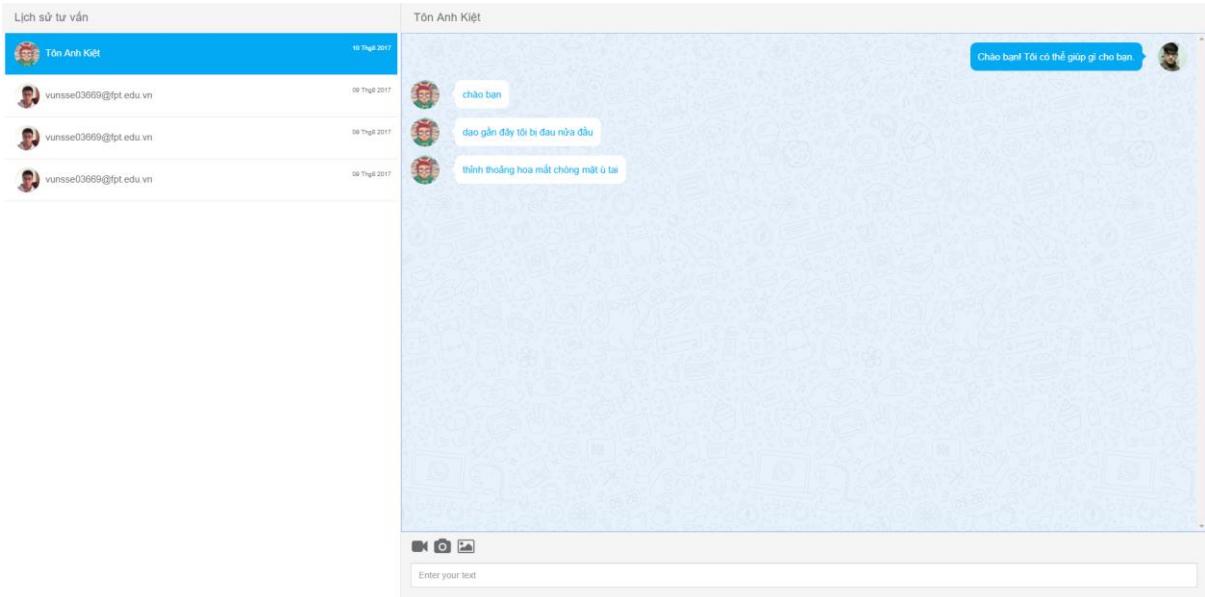


Figure 4-116: View conversation history Design screen

##### Class Diagram

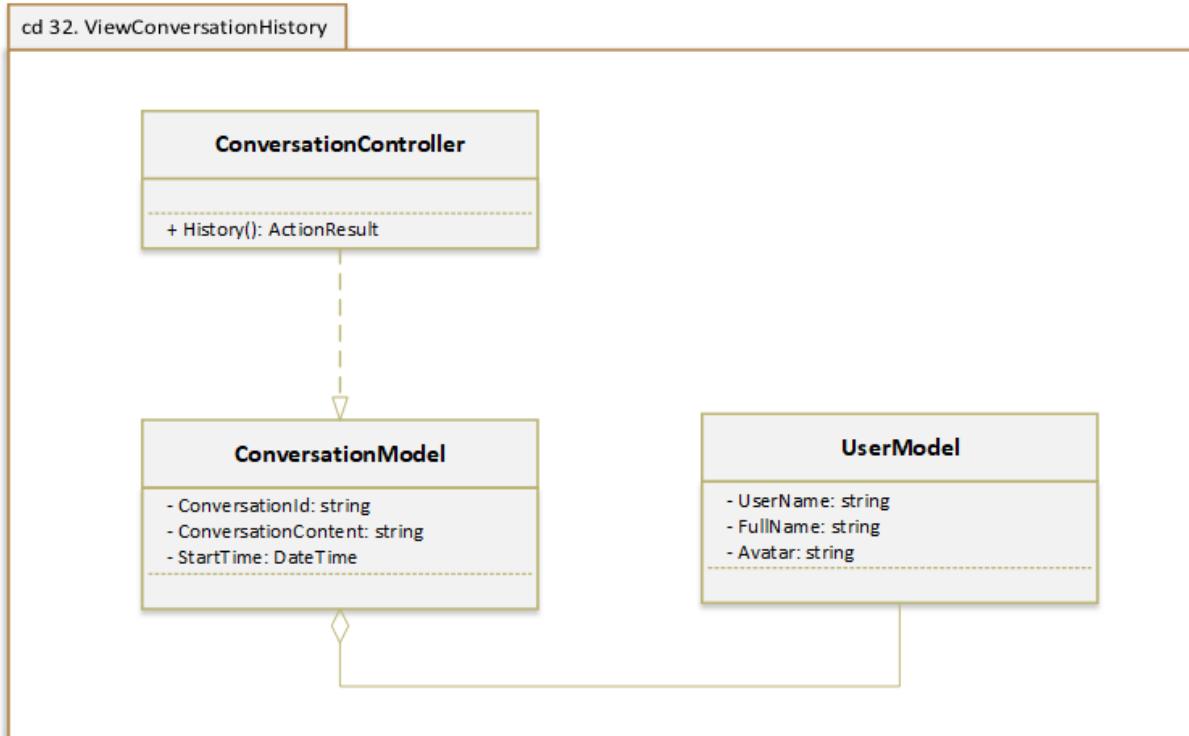


Figure 4-117: View conversation history Class diagram

##### Class Specification

ConversationController	
Physical address	MindDoctor / Areas / Customer / Controllers / ConversationController.cs

Base class			
Attributes			
No	Name	Type	Description
Operation			
History			
Return Type	ActionResult		
Parameters	Name	Type	Description

ConversationModel			
Physical address	MindDoctor / Models / ConversationModels / ConversationModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	string	
2.	StartTime	DateTime	
Operation			

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	Avatar	string	
Operation			

## Sequence Diagram

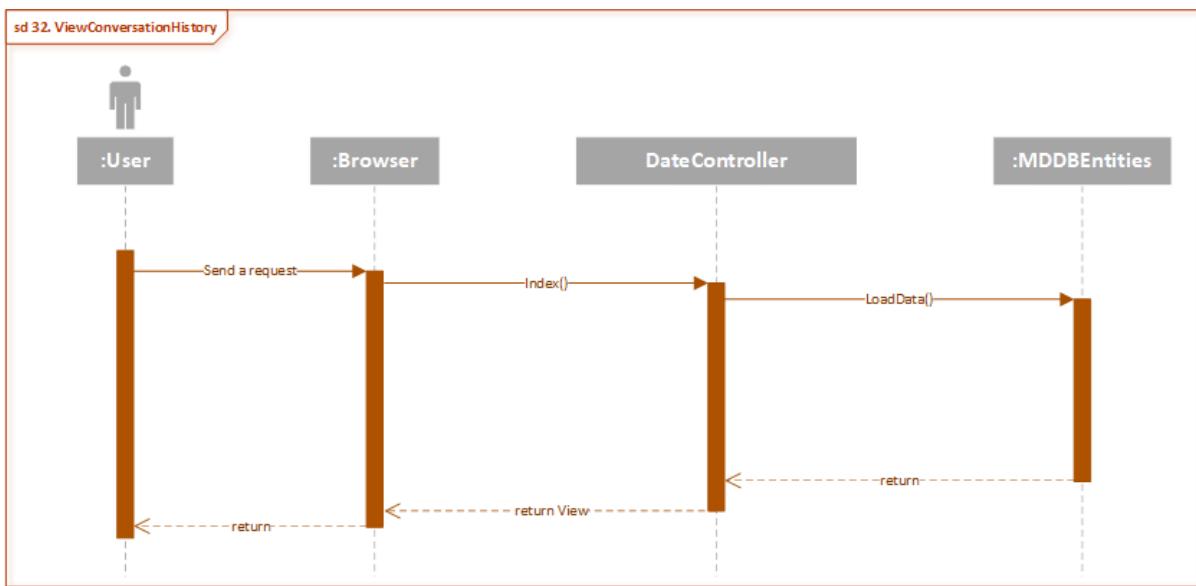


Figure 4-118: View conversation history Sequences diagram

#### 4.3.4.33 Make an appointment with Advisor

##### Screen Design

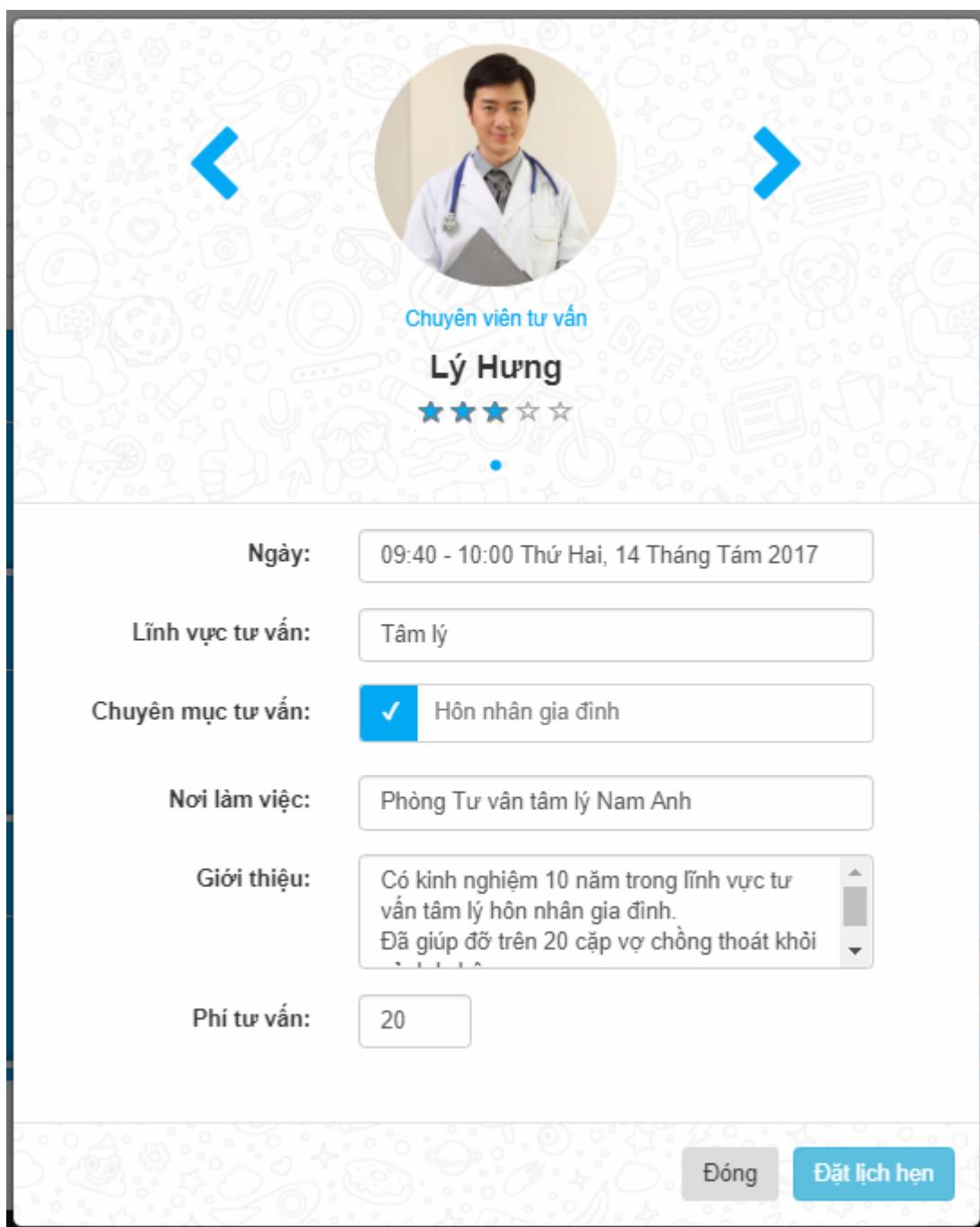


Figure 4-119: Make an appointment Screen design

## Class Diagram

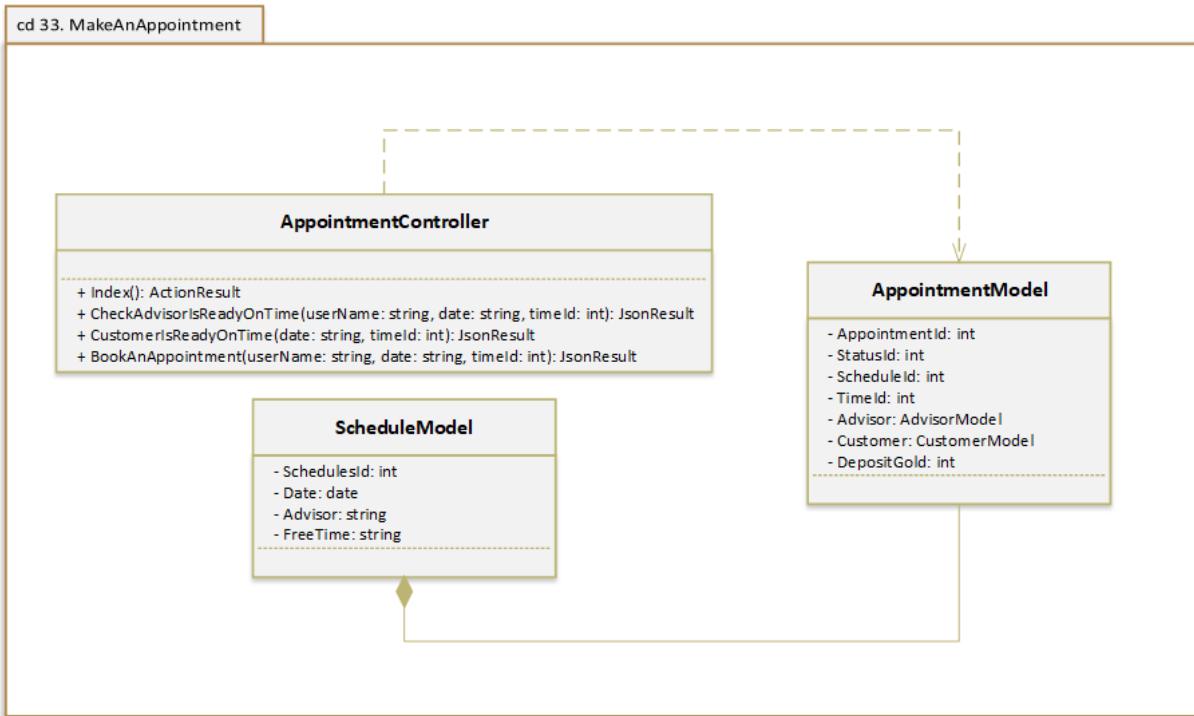


Figure 4-120: Make an appointment Class diagram

## Class Specification

AppointmentController			
Physical address	MindDoctor / Controllers / Appointment.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		
Parameters	Name	Type	Description
CheckAdvisorIsReadyOnTime			
Return Type	JsonResult		
Parameters	Name	Type	Description
	userName	string	Username of advisor.
	date	string	
	timeId	int	
BookAnAppointment			
Return Type	JsonResult		
Parameters	Name	Type	Description
	userName	string	Username of advisor.
	date	string	
	timeId	int	
CustomerIsReadyOnTime			
Return Type	JsonResult		

Parameters	Name	Type	Description
	date	string	
	timeId	int	

<b>AppointmentModel</b>			
Physical address	MindDoctor / Models / AppointmentModels / AppointmentModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	AppointmentId	int	
2.	StatusId	int	
3.	ScheduleId	int	
4.	Date	Date	
5.	TimeId	int	
6.	Advisor	AdvisorModel	
7.	Customer	CustomerModel	
8.	DepositGold	int	
Operation			

<b>ScheduleModel</b>			
Physical address	MindDoctor / Models / AppointmentModels / ScheduleModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ScheduleId	int	
2.	Advisor	string	
3.	Date	Date	
4.	FreeTime	string	
Operation			

## Sequence Diagram

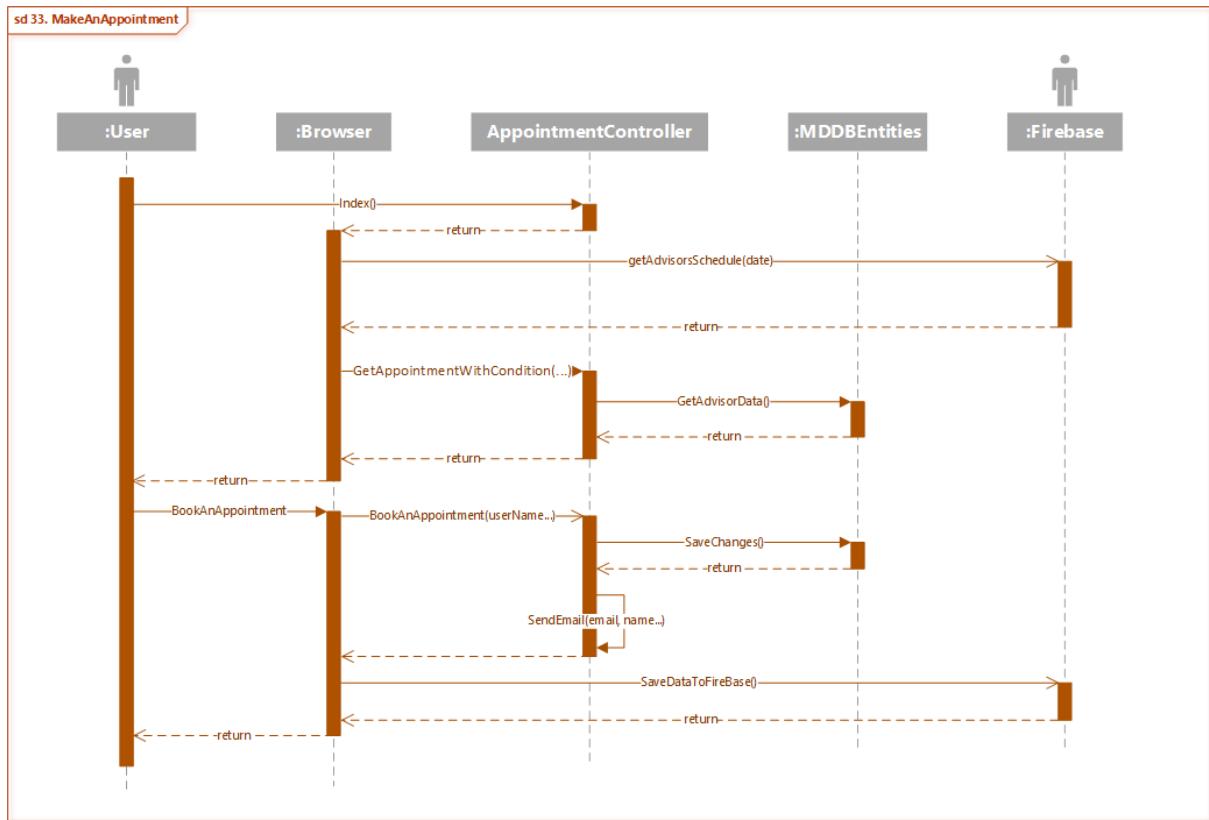


Figure 4-121: Make an appointment Sequence diagram

## Implementation

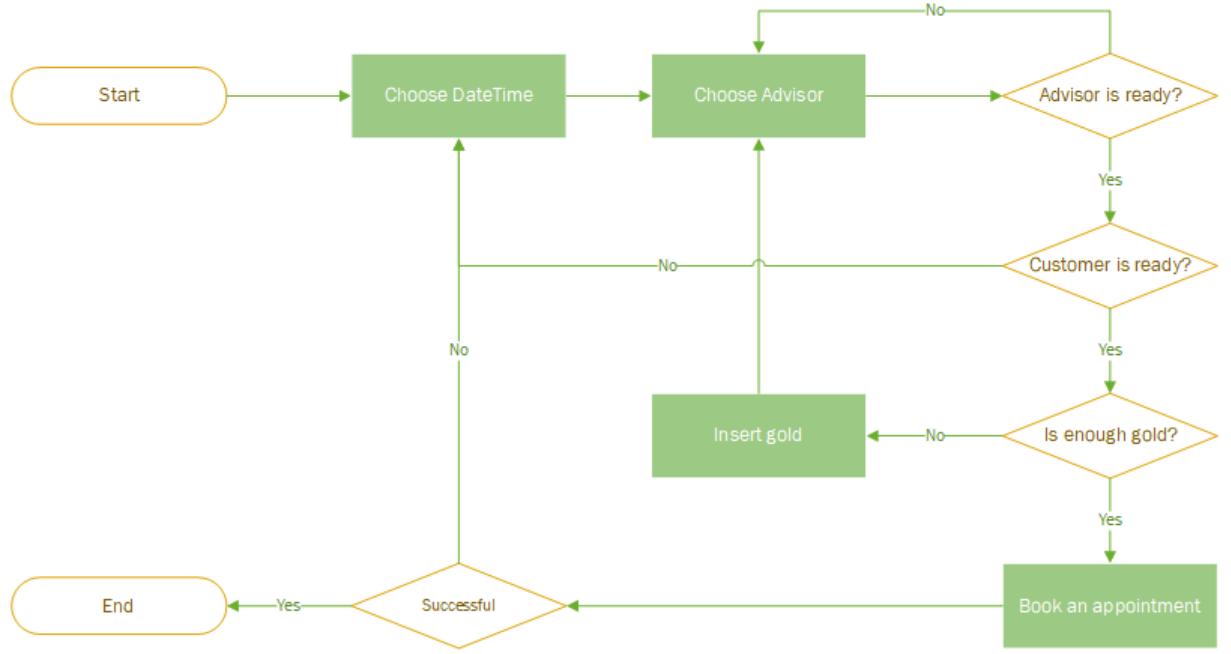


Figure 4-108: Make an appointment Flow chart

It is realtime so that needs to check carefully with 4 conditions to makes an appointment successful.

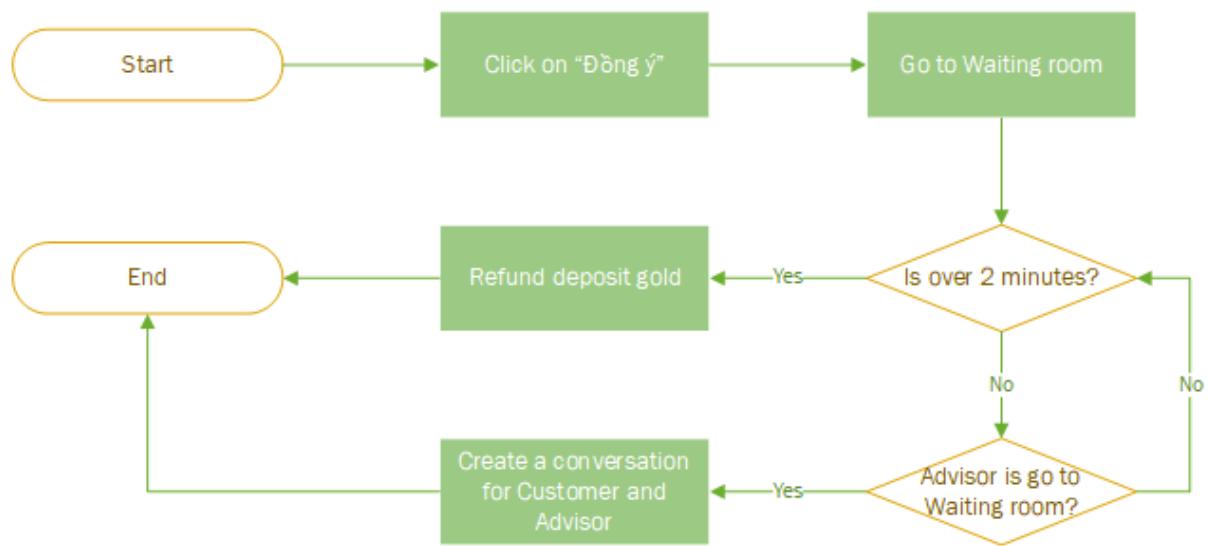


Figure 4-119: Go to conversation after making an appointment

#### 4.3.4.34 Cancel an appointment with Advisor

##### Screen Design

TRẠNG THÁI	NGÀY	GIỜ	TƯ VẤN VIÊN		
Hủy bỏ	Thứ Hai, 14 Tháng Tám 2017	08:00 - 08:20	Nguyễn Như Ý	④ 10	
Bang chờ	Thứ Hai, 14 Tháng Tám 2017	08:20 - 08:40	Đỗ Ngọc Trinh	④ 20	Hủy bỏ
Bang chờ	Thứ Hai, 14 Tháng Tám 2017	08:40 - 09:00	Đỗ Ngọc Trinh	④ 20	Hủy bỏ
Bang chờ	Thứ Hai, 14 Tháng Tám 2017	09:00 - 09:20	Nguyễn Như Ý	④ 10	Hủy bỏ
Bang chờ	Thứ Hai, 14 Tháng Tám 2017	10:20 - 10:40	Cao Anh Trúc	④ 15	Hủy bỏ
Bang chờ	Thứ Hai, 14 Tháng Tám 2017	10:40 - 11:00	Vương Tuệ Văn	④ 0	Hủy bỏ

Figure 4-122: Cancel an appointment Screen design

##### Class Diagram

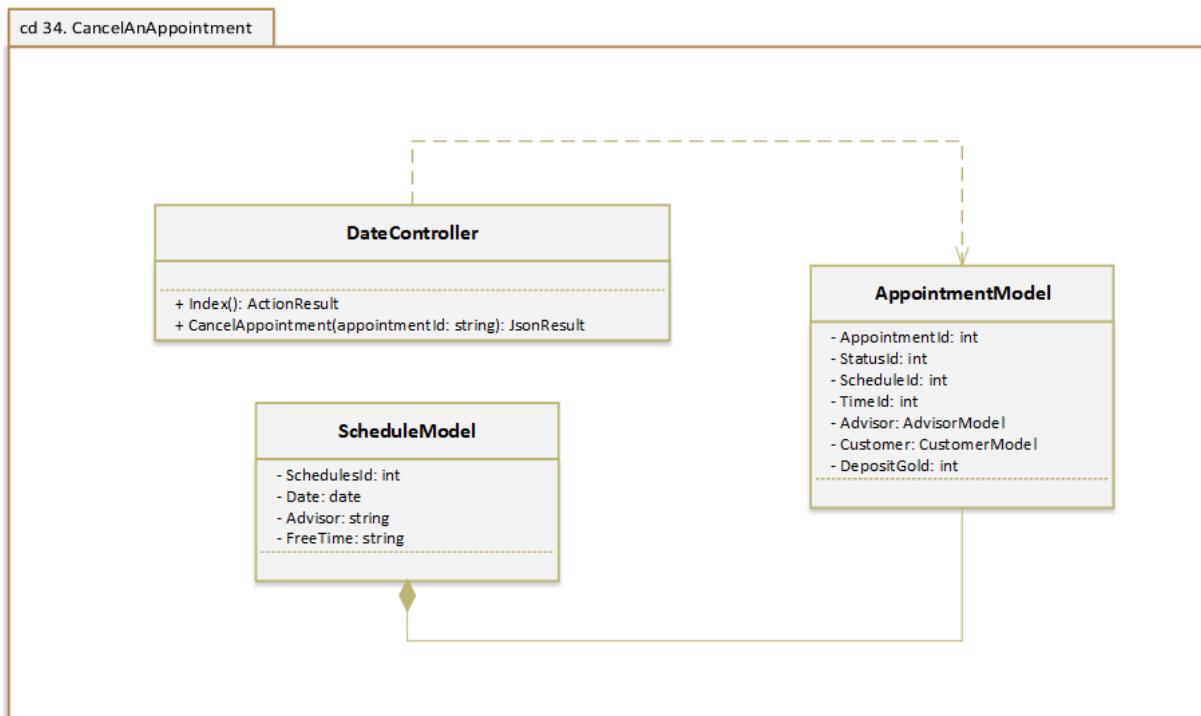


Figure 4-123: Cancel an appointment Class diagram

##### Class Specification

AppointmentController			
Physical address	MindDoctor / Controllers / Appointment.cs		
Base class	BaseController		
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		
Parameters	Name	Type	Description
CancelAppointment			

Return Type	JsonResult		
Parameters	Name	Type	Description
	AppointmentId	string	

<b>AppointmentModel</b>			
Physical address	MindDoctor / Models / AppointmentModels / AppointmentModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	AppointmentId	int	
2.	StatusId	int	
3.	ScheduleId	int	
4.	Date	Date	
5.	TimeId	int	
6.	Advisor	AdvisorModel	
7.	Customer	CustomerModel	
8.	DepositGold	int	
Operation			

<b>ScheduleModel</b>			
Physical address	MindDoctor / Models / AppointmentModels / ScheduleModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ScheduleId	int	
2.	Advisor	string	
3.	Date	Date	
4.	FreeTime	string	
Operation			

## Sequence Diagram

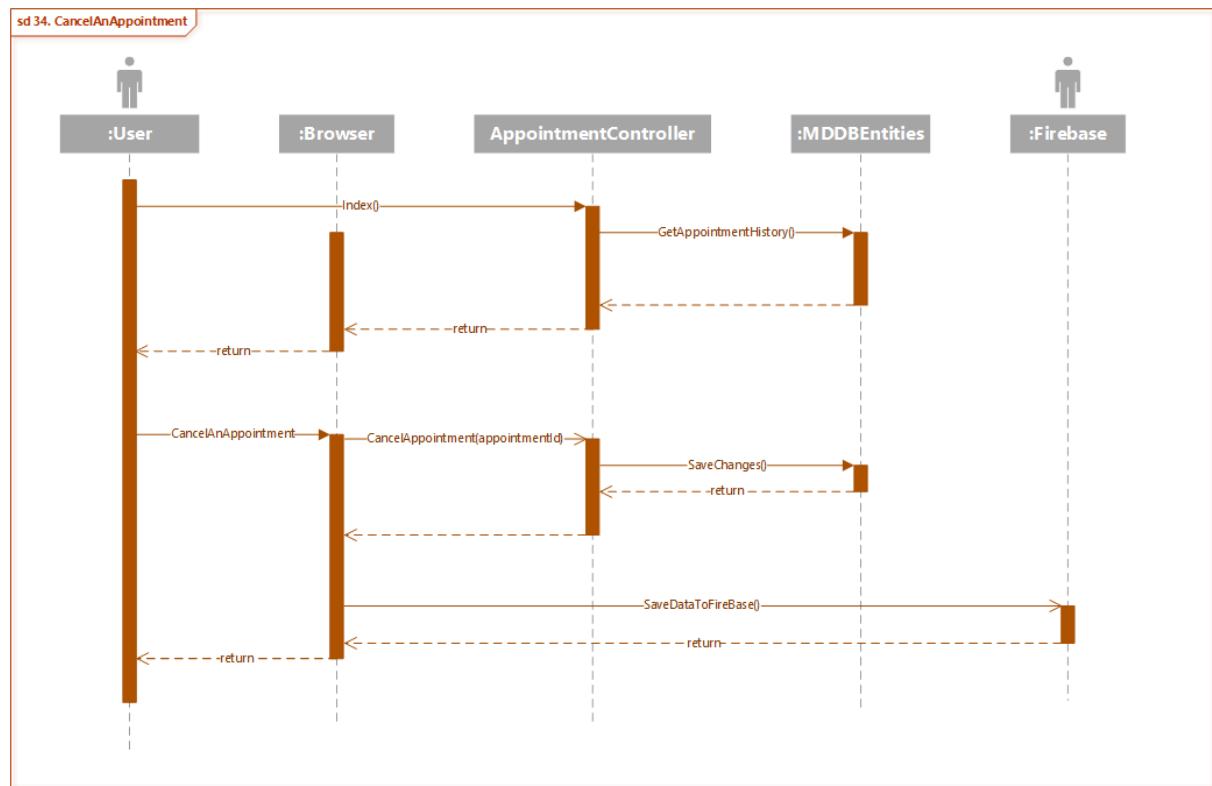


Figure 4-124: Cancel an appointment Sequence diagram

#### 4.3.4.35 View appointment history

##### Screen Design

TRẠNG THÁI	NGÀY	GIỜ	TƯ VẤN VIÊN		
Hủy bỏ	Thứ Năm, 03 Tháng Tám 2017	08:40 - 09:00	Phạm Hoàng Sơn	20	

Figure 4-125: View appointment history Screen design

##### Class Diagram

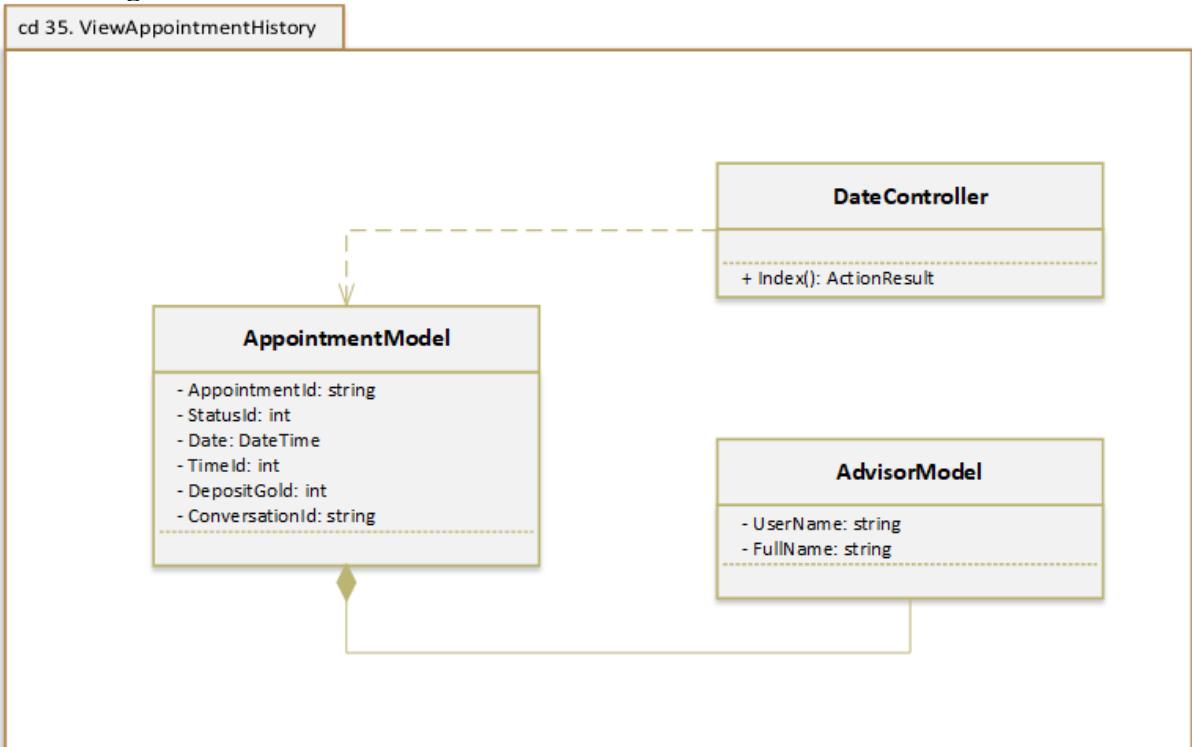


Figure 4-126: View appointment history Class diagram

##### Class Specification

DateController			
Physical address	MindDoctor / Areas / Advisor / Controllers / DateController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		
Parameters	Name	Type	Description

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
Operation			

AppointmentModel			
Physical address	MindDoctor / Models / AppointmentModels / AppointmentModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	AppointmentId	string	
2.	StatusId	int	
3.	Date	DateTime	
4.	TimeId	int	
5.	DepositGold	int	
6.	ConversationId	string	
Operation			

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel.cs		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
Operation			

## Sequence Diagram

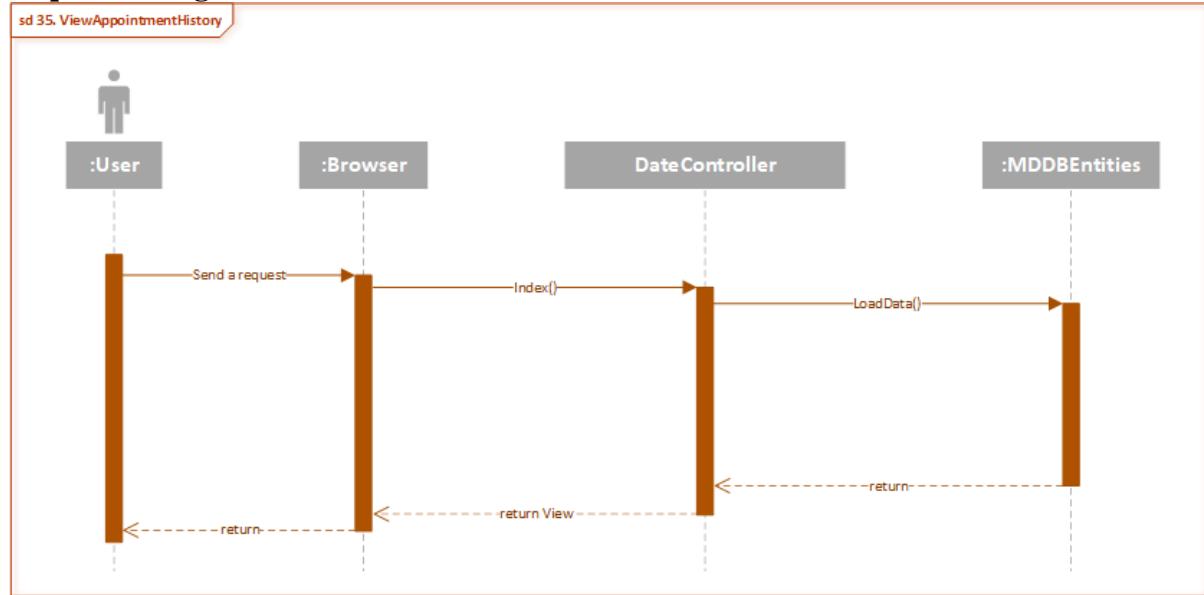


Figure 4-127: View appointment history Sequence diagram

#### 4.3.4.36 Accept Customer's chat request

##### Screen Design

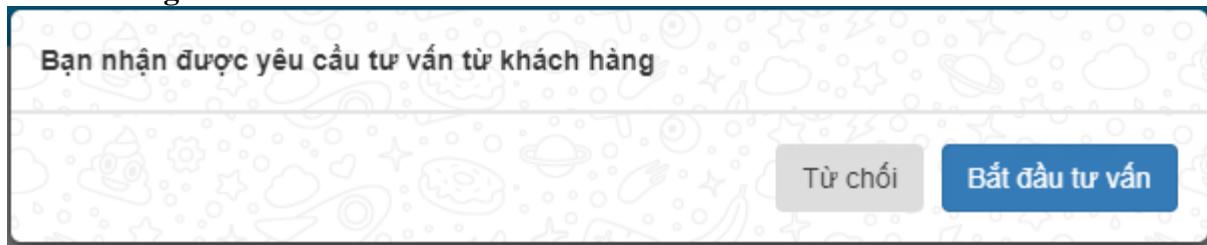


Figure 4-128: Accept Customer's chat request Screen design

##### Class Diagram

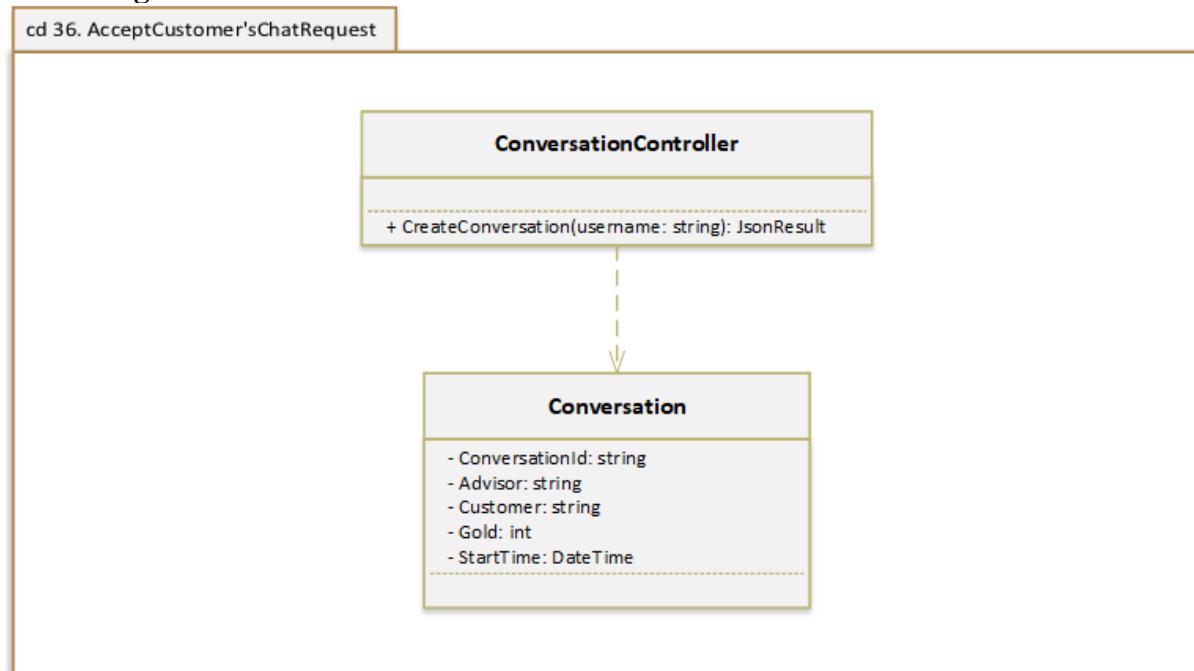


Figure 4-129: Accept Customer's chat request Class diagram

##### Class Specification

ConversationController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
CreateConversation			
Return Type	JsonResult		
Parameters	Name	Type	Description
	username	string	

Conversation	
Physical	MindDoctor / EntityFramework / Conversation.cs

address			
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	string	
2.	Advisor	string	
3.	Customer	string	
4.	Gold	int	
5.	StartTime	DateTime	
Operation			

### Sequence Diagram

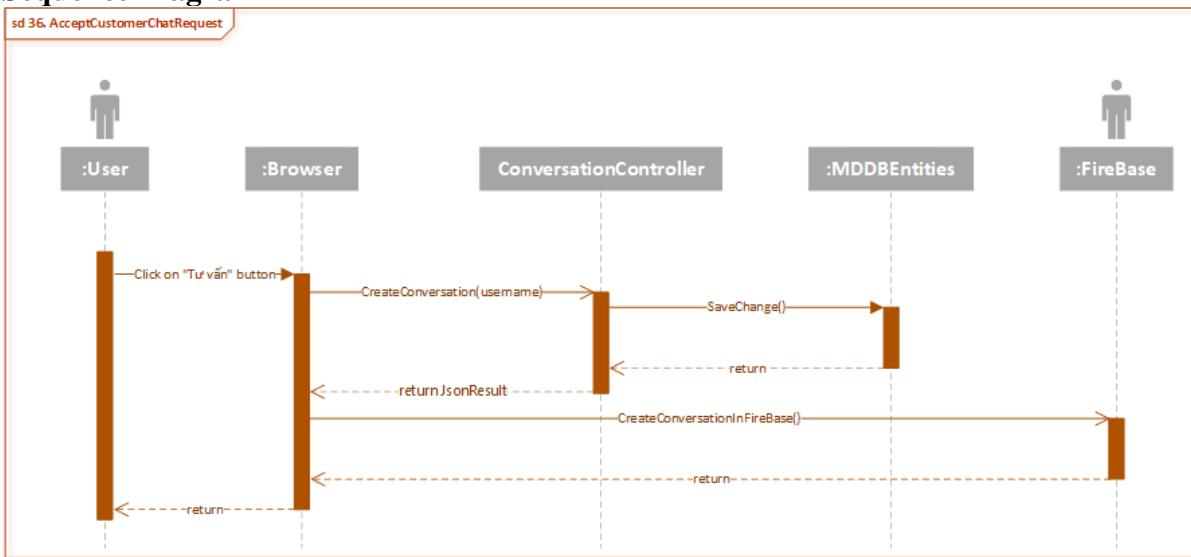


Figure 4-130: Accept Customer's chat request Sequence diagram

#### 4.3.4.37 Deny Customer's chat request

##### Screen Design

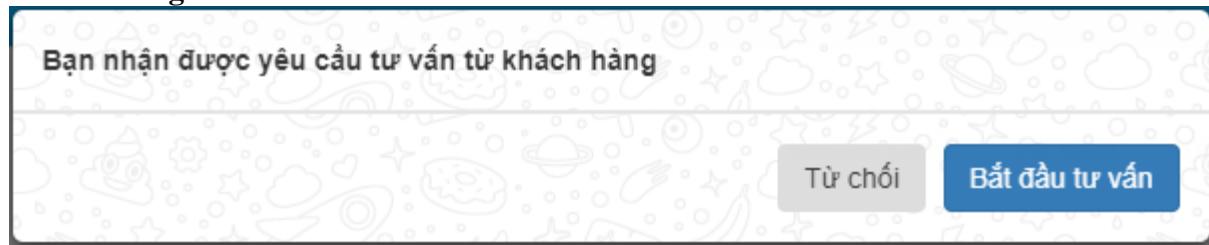


Figure 4-131: Deny Customer's chat request Screen design

##### Class Diagram

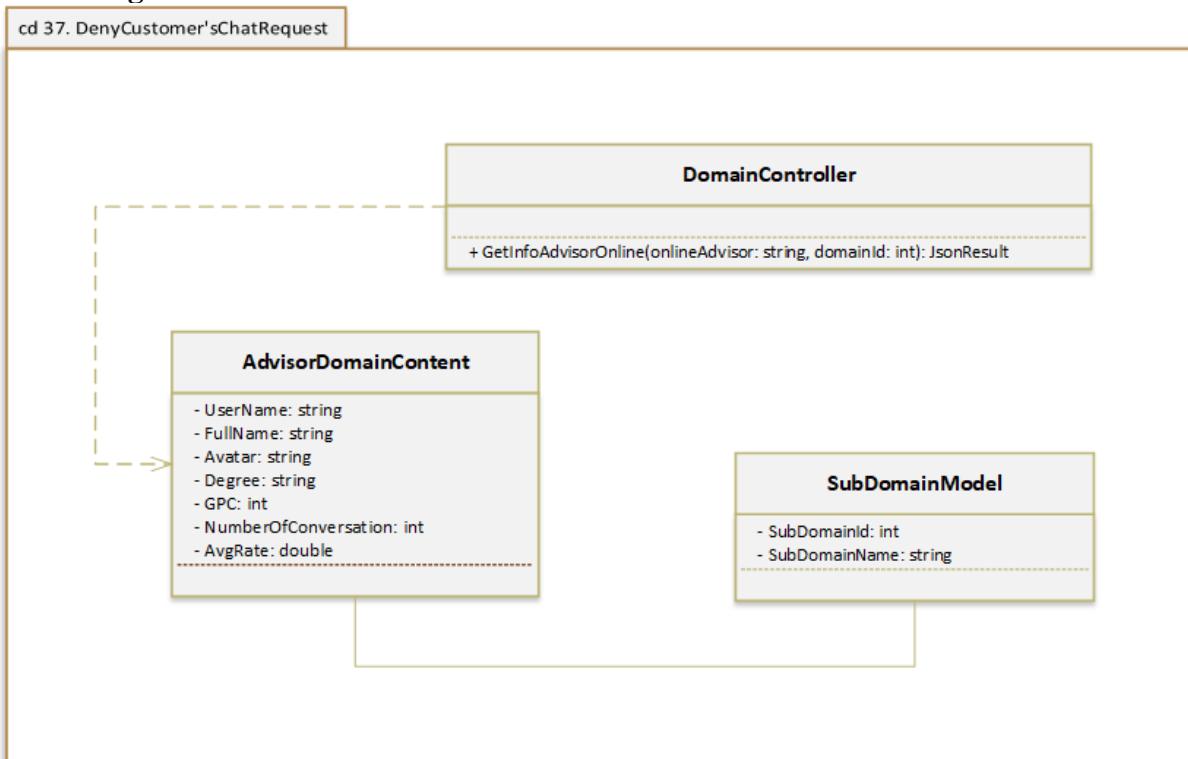


Figure 4-132: Deny Customer's chat request Class diagram

##### Class Specification

DomainController			
Physical address	MindDoctor / Controllers / DomainController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
GetOnlineAdvisorList			
Return Type	JsonResult		
Parameters	Name	Type	Description
	onlineAdvisor	string	List advisor was converted to Json

			string
	domainId	int	Current domain of detail page

AdvisorDomainContent			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	Avatar	string	
4.	Degree	string	
5.	GPC	int	
6.	NumberOfConversation	int	
7.	AvgRate	double	
Operation			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

## Sequence Diagram

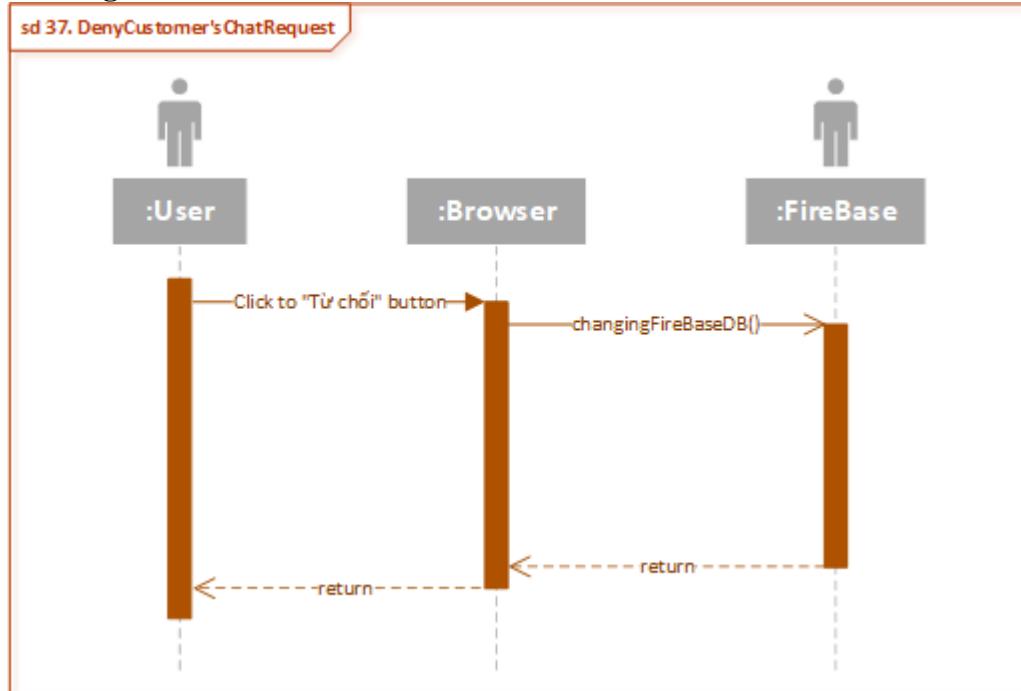


Figure 4-133: Deny Customer's chat request Sequence diagram

#### 4.3.4.38 Set default free time on week

##### Screen Design



Figure 4-134: Set default free time on week Screen design

##### Class Diagram

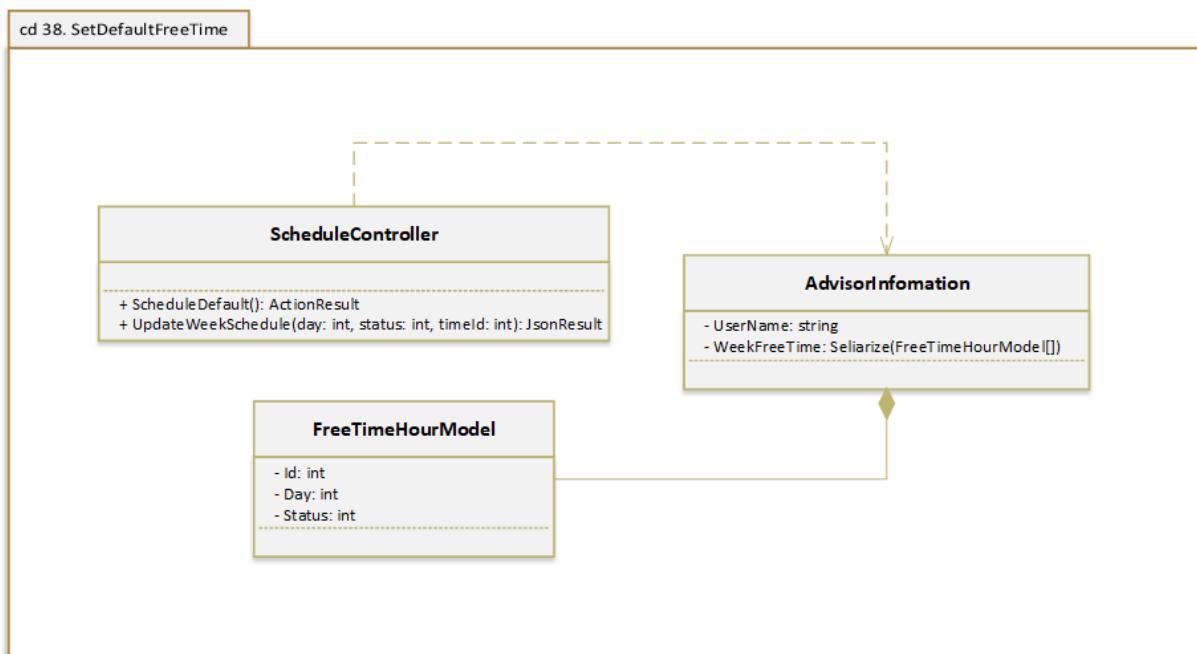


Figure 4-135: Set default free time on week Class diagram

##### Class Specification

ScheduleController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ScheduleController.cs		
Base class			
Attributes			
No	Name	Type	Description

Operation			
<b>ScheduleDefault</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description
<b>UpdateWeekSchedule</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	day	int	Date of week.
	status	int	
	timeId	int	

AdvisorInformation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AdvisorInformation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	WeekFreeTime	string	JSON string to save free time on week.
Operation			

FreeTimeHourModel			
Physical address	MindDoctor / Models / AppointmentModels / FreeTimeModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Id	int	
2.	Day	int	
3.	Status	int	
Operation			

## Sequence Diagram

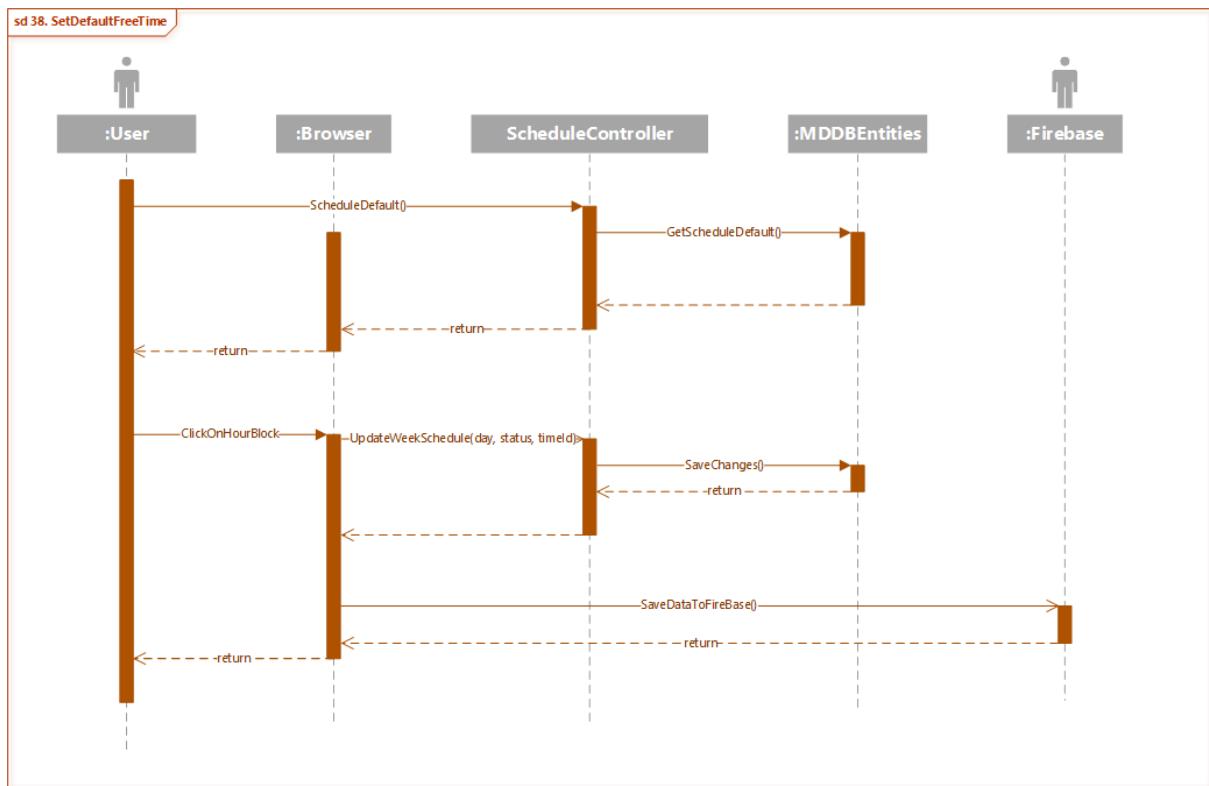


Figure 4-1236: Set default free time on week Sequence diagram

#### 4.3.4.39 Set free time by date

##### Screen Design

08:00 - 08:20	08:20 - 08:40	08:40 - 09:00
09:00 - 09:20	09:20 - 09:40	09:40 - 10:00
10:00 - 10:20	10:20 - 10:40	10:40 - 11:00
11:00 - 11:20	11:20 - 11:40	11:40 - 12:00
12:00 - 12:20	12:20 - 12:40	12:40 - 13:00

Figure 4-1337: Set free time by date Screen design

##### Class Diagram

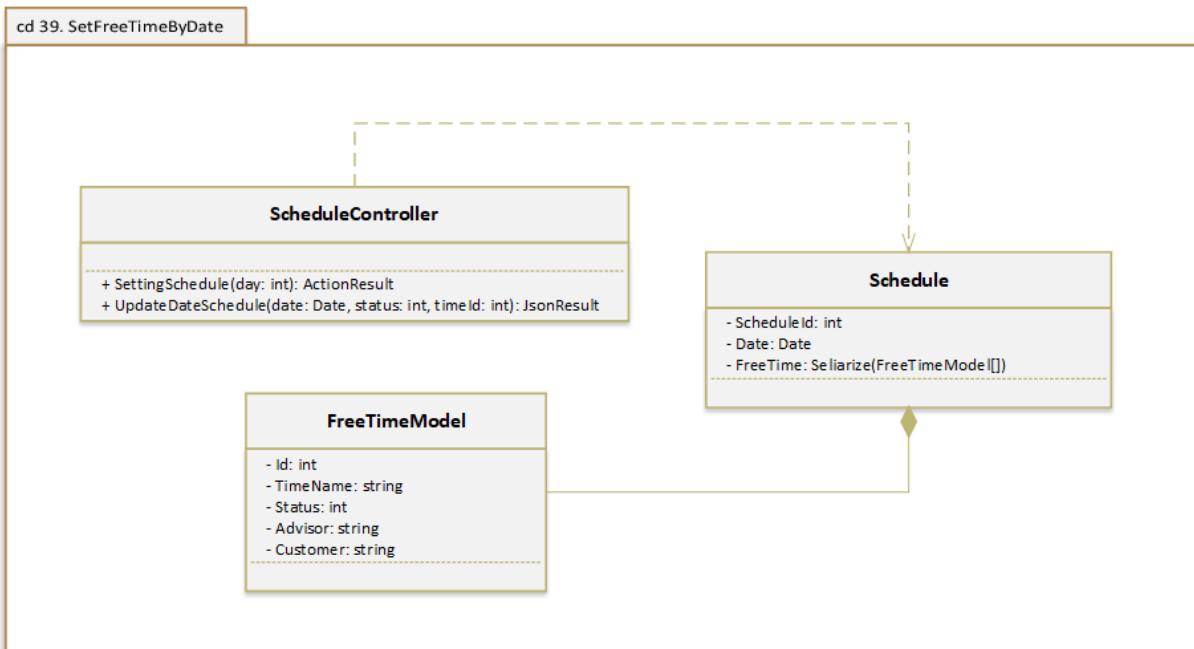


Figure 4-138: Set free time by date Class diagram

##### Class Specification

ScheduleController			
Physical address	MindDoctor / Areas / Advisor / Controllers / ScheduleController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
SettingSchedule			

Return Type	ActionResult		
Parameters	Name	Type	Description
	day	int	
<b>UpdateDateSchedule</b>			
Return Type	JsonResult		
Parameters	Name	Type	Description
	date	Date	
	status	int	
	timeId	int	

<b>Schedule</b>			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Schedule.cs		
Base class			
<b>Attributes</b>			
No	Name	Type	Description
1.	ScheduleId	int	
2.	Date	Date	
3.	FreeTime	string	
<b>Operation</b>			

<b>FreeTimeModel</b>			
Physical address	MindDoctor / Models / AppointmentModels / FreeTimeModel.cs		
Base class			
<b>Attributes</b>			
No	Name	Type	Description
1.	Id	int	
2.	TimeName	string	
3.	Status	int	
4.	Advisor	string	
5.	Customer	string	
<b>Operation</b>			

## Sequence Diagram

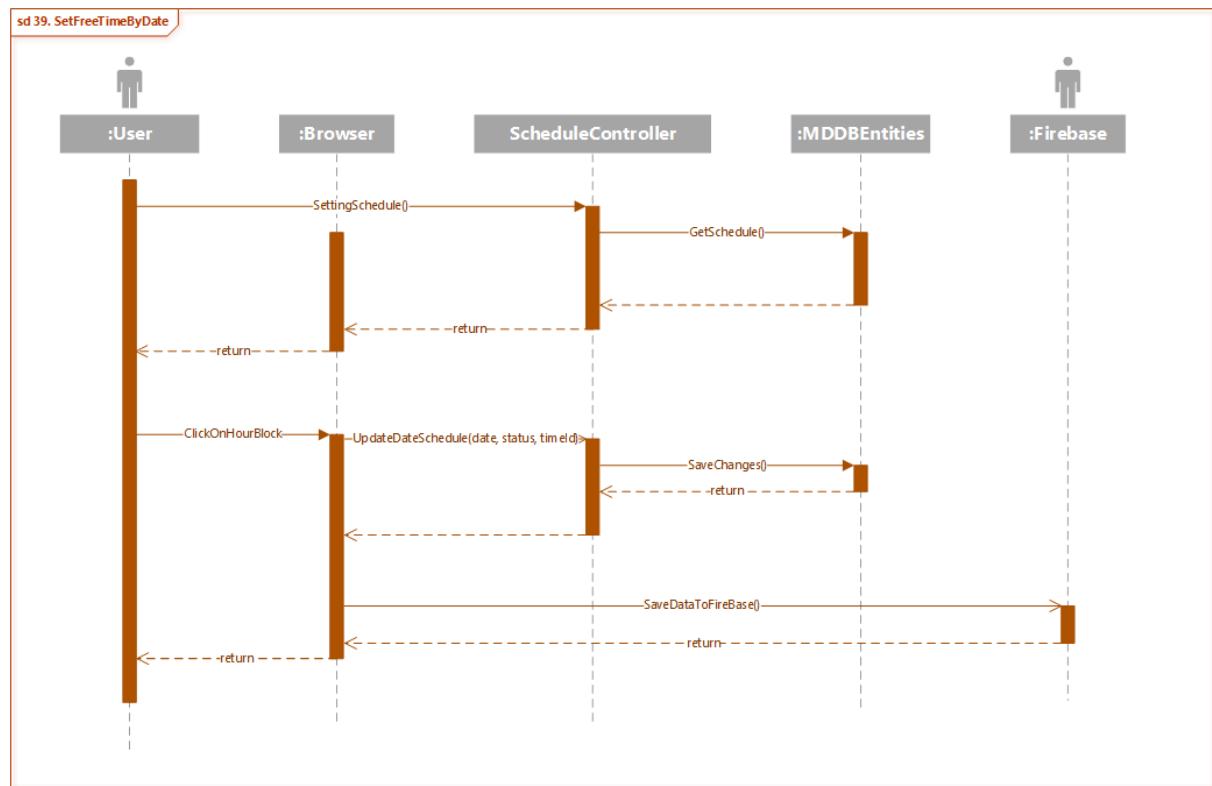


Figure 4-139: Set free time by date Sequence diagram

#### 4.3.4.40 View payment history

##### Screen Design

**LỊCH SỬ THANH TOÁN**

---

GIÁ TRỊ (VND) ♦	@ ♦	NỘI DUNG ♦	NGÀY ♦
32.500	50	Trả lương cho Tư vấn viên: Nguyễn Như Ý. Số tiền: 32.500 VNĐ	22:36, 13 Tháng Tám 2017
<span style="border: 1px solid #ccc; padding: 2px;">«</span> <span style="border: 1px solid #ccc; padding: 2px;">&lt;</span> <span style="border: 1px solid #0070C0; background-color: #0070C0; color: white; padding: 2px;">1</span> <span style="border: 1px solid #ccc; padding: 2px;">&gt;</span> <span style="border: 1px solid #ccc; padding: 2px;">»</span>			

Figure 4-140: View payment history Screen design

##### Class Diagram

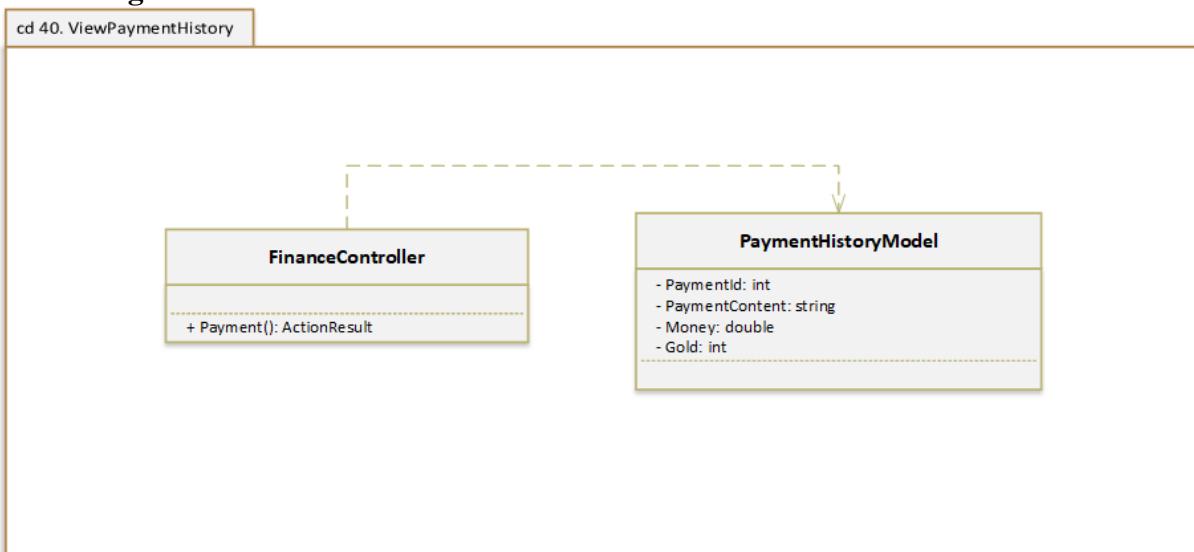


Figure 4-1441: View payment history Class diagram

##### Class Specification

FinanceController			
Physical address	MindDoctor / Controllers / FinanceController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
<b>Payment</b>			
Return Type	ActionResult		
Parameters	Name	Type	Description

##### PaymentHistoryModel

PaymentHistoryModel	
Physical address	MindDoctor / Models / FinanceModels / FinanceModel.cs
Base class	

Attributes			
No	Name	Type	Description
1.	PaymentId	string	
2.	PaymentContent	string	
3.	Money	double	
4.	Gold	int	
Operation			

### Sequence Diagram

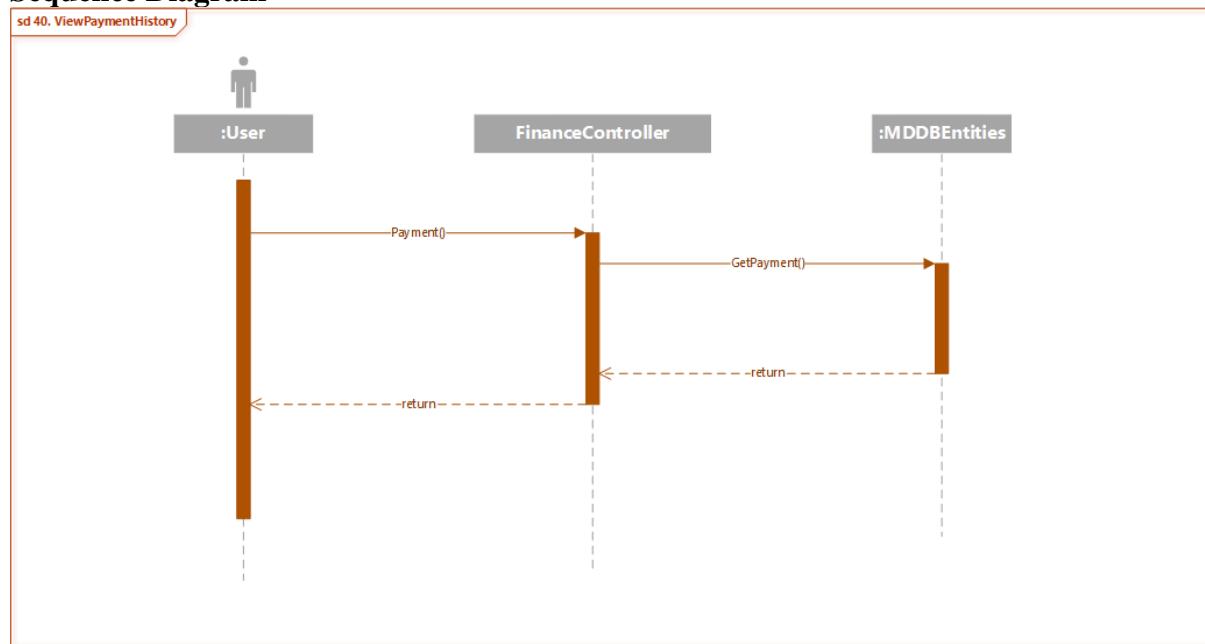


Figure 4-142: View payment history Sequence diagram

#### 4.3.4.41 Assess post

##### Screen Design

**Chúc mừng bạn**

Chúc mừng bạn đã trở thành thành viên ban kiểm duyệt bài đăng của hệ thống. Chúng tôi cần sự giúp đỡ của bạn trong việc phê duyệt nội dung bài đăng này có đúng và phù hợp với hệ thống của chúng ta hay không. Vui lòng click vào link bên dưới để xem bài đăng. Cảm ơn bạn đã hợp tác cùng chúc tôi.

[Xét duyệt bài đăng](#)



Không chỉ là được chủ động làm những điều mình thích mà bạn còn phải chịu trách nhiệm với những quyết định, việc làm của bản thân. Khi đối diện với những vấn đề vướng mắc trong học tập, hay cuộc sống hàng ngày bạn phải đổi mới và tìm cách giải quyết, chứ không quá dựa dẫm vào ai. Sự tự do ở trường đại học gắn liền với việc bạn phải sống có trách nhiệm với chính bản thân mình và học cách nói "không" khi cần thiết.

Nội dung không phù hợp

Nội dung phù hợp

Figure 4-1543: Assess post Screen design

## Class Diagram

cd 41. AssessPost

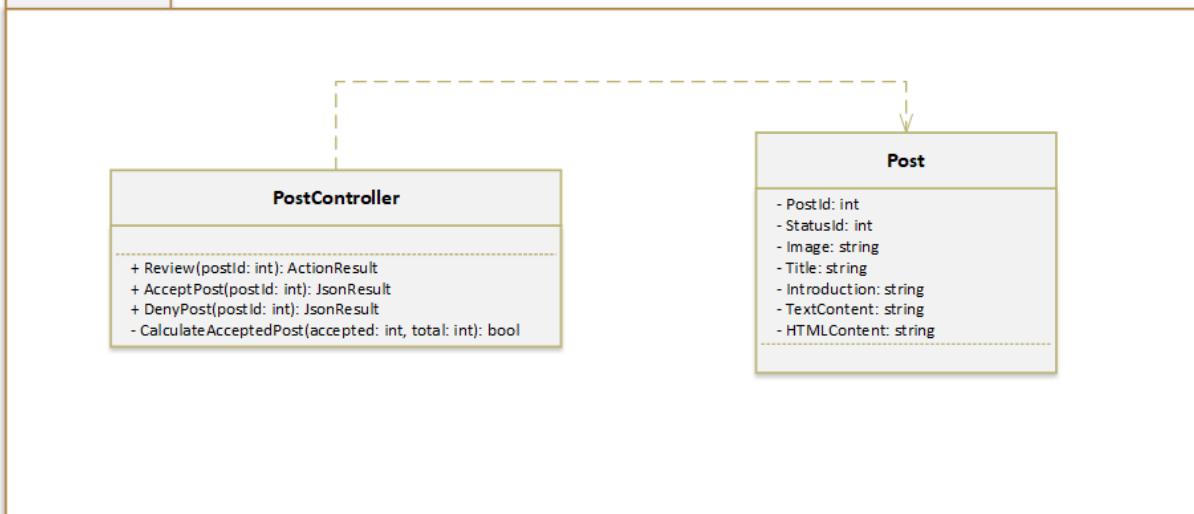


Figure 4-1644: Assess post Class diagram

## Class Specification

PostController			
Physical address	MindDoctor / Controllers / PostController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Review			
Return Type	ActionResult		
Parameters	Name	Type	Description
	postId	int	
AcceptPost			
Return Type	JsonResult		
Parameters	Name	Type	Description
	postId	int	
DenyPost			
Return Type	JsonResult		
Parameters	Name	Type	Description
	postId	int	
CalculateAcceptedPost			
Return Type	bool		
Parameters	Name	Type	Description
	accepted	int	
	total	int	

## Sequence Diagram

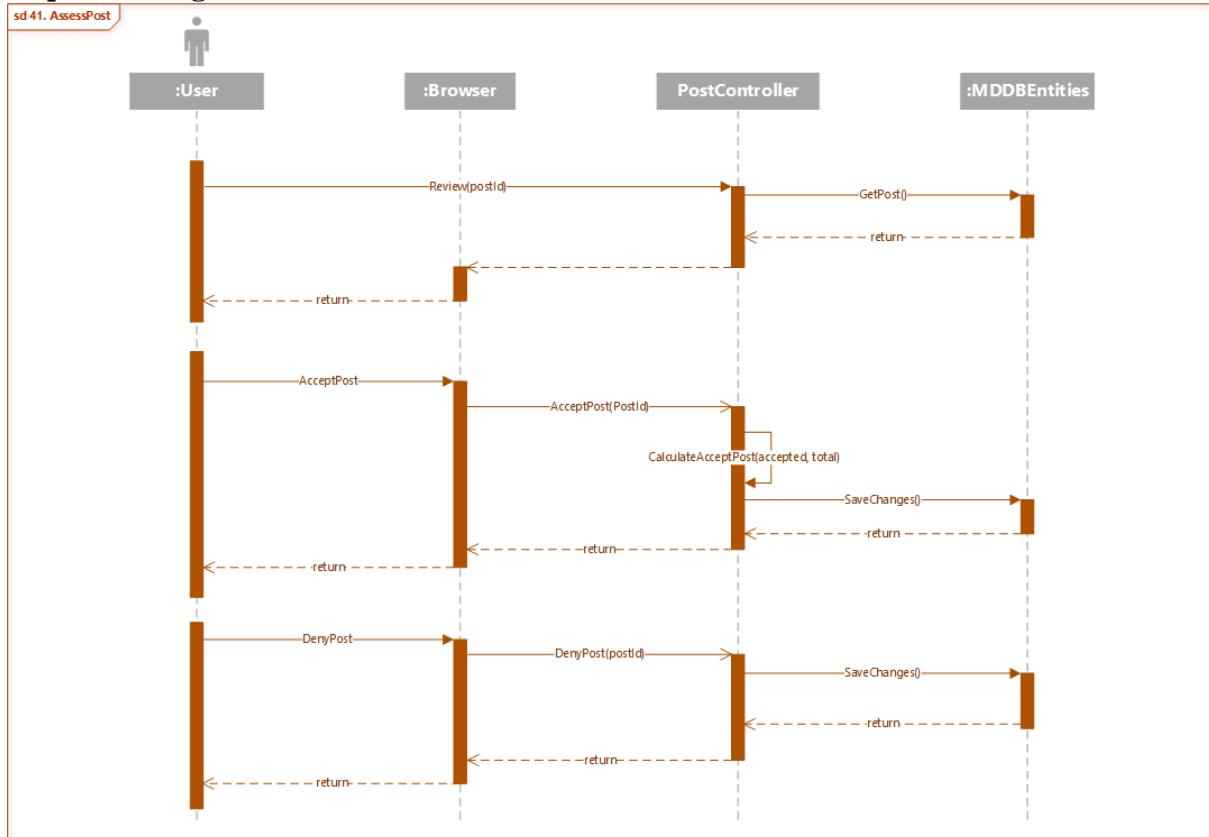


Figure 4-1745: Assess post Sequence diagram

#### 4.3.4.42 View customers

##### Screen Design

Bảng dữ liệu Danh sách khách hàng						
Show		10	▼ entries	Search:	Copy CSV Excel PDF Print	
Showing 1 to 4 of 4 entries						
Trạng thái	Họ và tên	Email	Giới tính	Gold	Số cuộc trò chuyện	
Hoạt động	Bùi Vinh Đạt	datbv95@gmail.com	Khác	970	13	
Hoạt động	Đào Bá Anh	md.cus01@mailnesia.com	Nam	0	0	
Hoạt động	Tôn Anh Kiệt	md.cus02@mailnesia.com	Nam	0	0	
Hoạt động	Đào Minh Hà	md.cus03@mailnesia.com	Nữ	1000	3	

Previous 1 Next

Figure 4-1846: View customers Screen design

##### Class Diagram

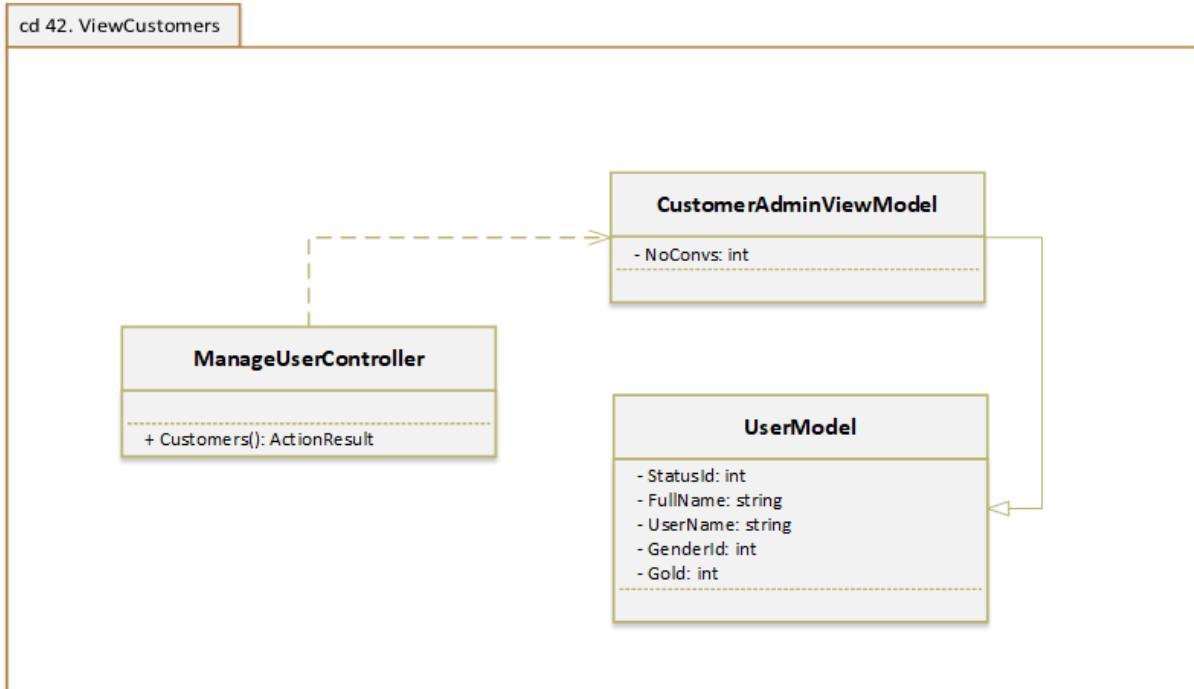


Figure 4-19: View customers Class diagram

##### Class Specification

ManageUserController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Customers			
Return Type	<code>ActionResult</code>		
Parameters	Name	Type	Description

CustomerAdminViewModel			
Physical address	MindDoctor / Models / UserModels / CustomerModel.cs		
Base class	CustomerModel		
Attributes			
No	Name	Type	Description
1.	NoConvs	int	
Operation			

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	NoConvs	int	
2.	FullName	string	
3.	UserName	string	
4.	GenderId	int	
5.	Gold	Int	
Operation			

### Sequence Diagram

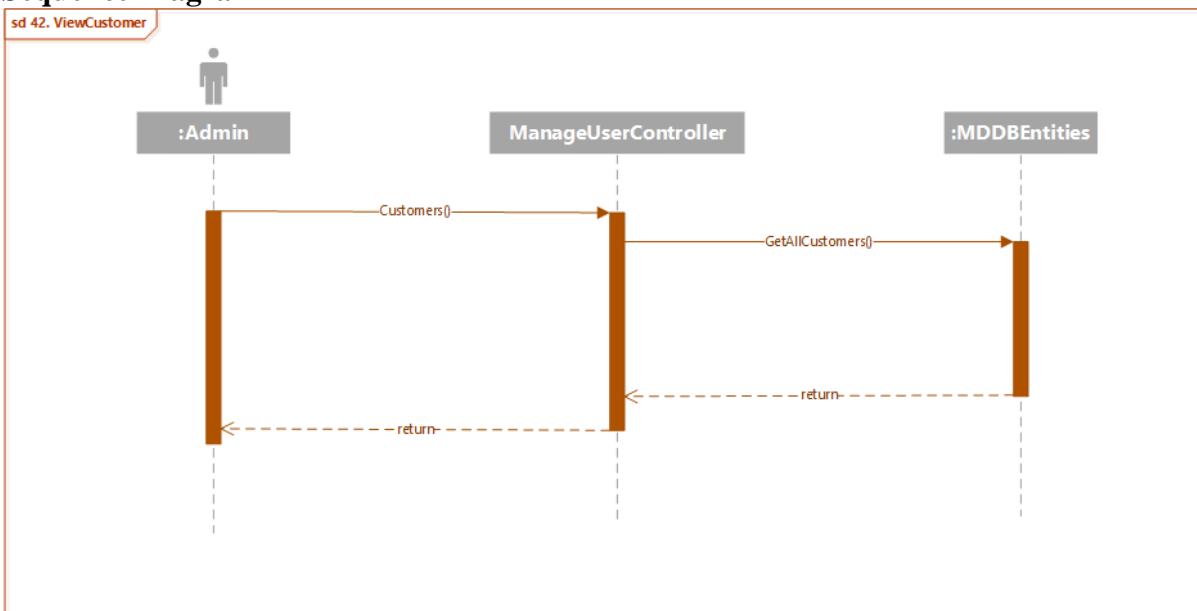


Figure 4-2048: View customers Sequence diagram

#### 4.3.4.43 View customer's profile

##### Screen Design

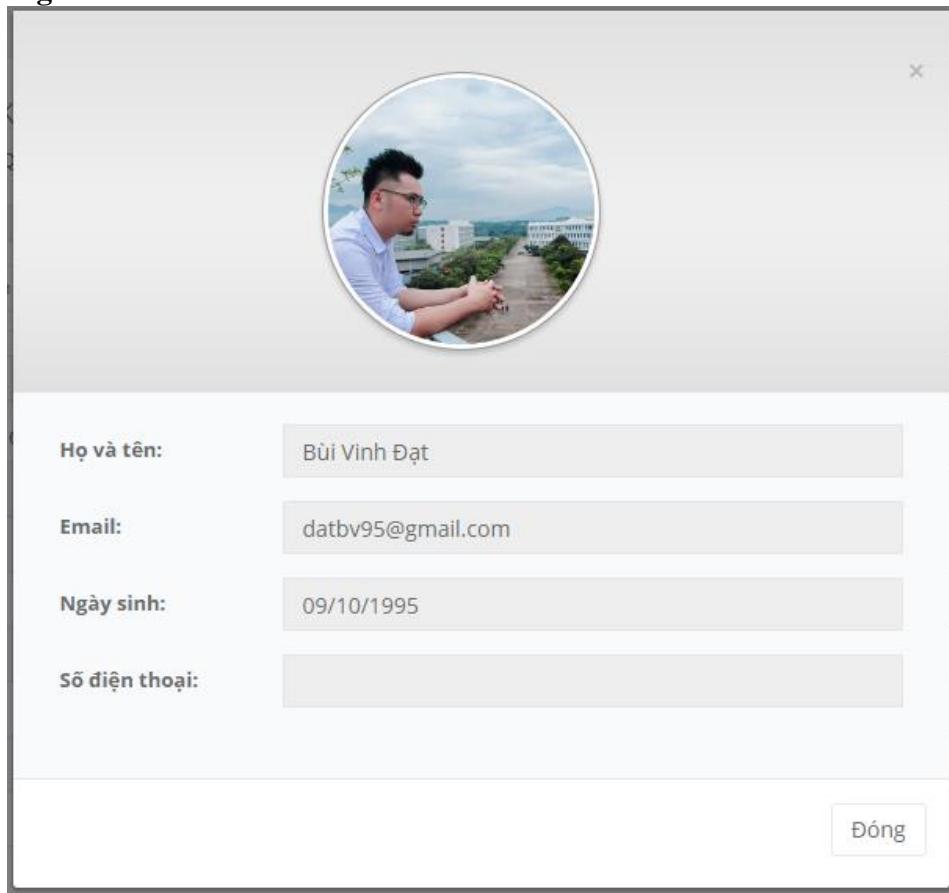


Figure 4-2149: View customer's profile Screen design

##### Class Diagram

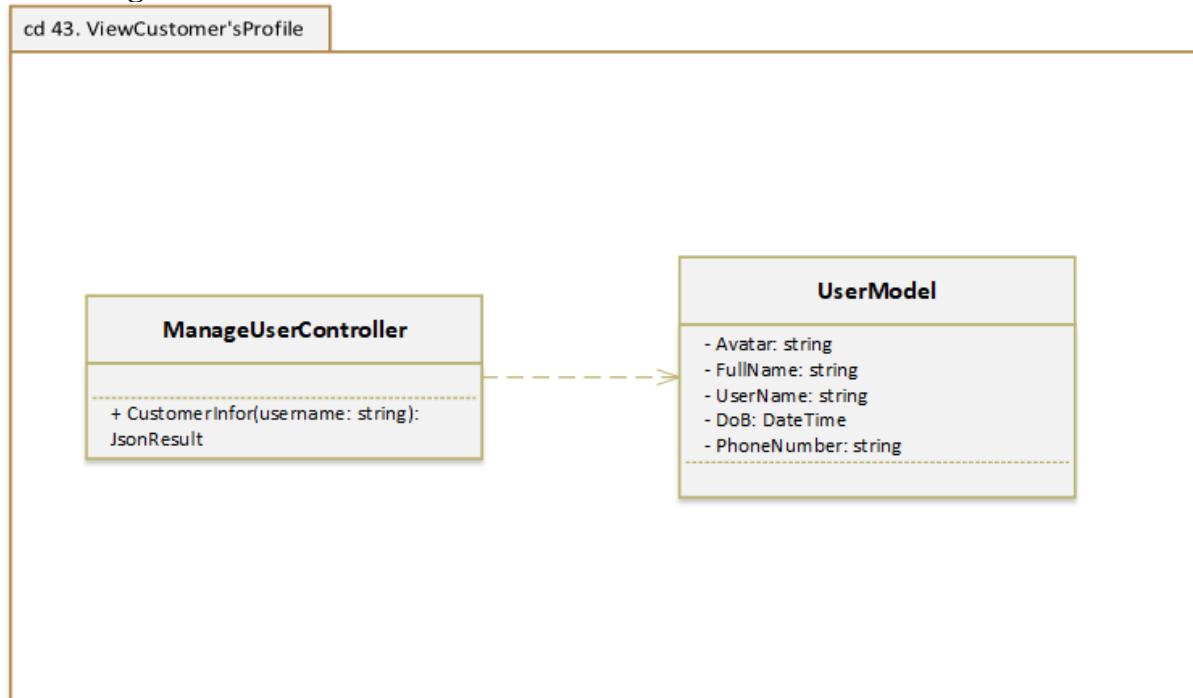


Figure 4-2250: View customer's profile Class diagram

## Class Specification

ManageUserController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
CustomerInfor			
Return Type	JsonResult		
Parameters	Name	Type	Description
	username	string	Username of customer.

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Avatar	string	
2.	FullName	string	
3.	UserName	string	
4.	DoB	DateTime	
5.	PhoneNumber	string	
Operation			

## Sequence Diagram

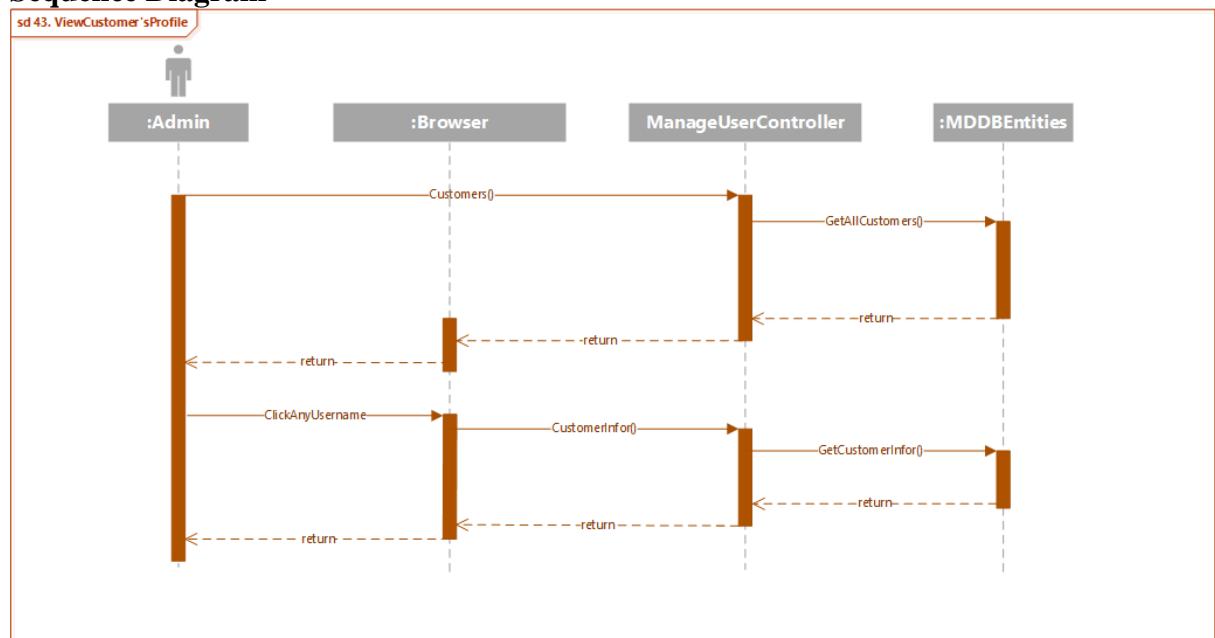


Figure 4-2351: View customer's profile Sequence diagram

#### 4.3.4.44 View advisors

##### Screen Design

Bảng dữ liệu Danh sách Tư vấn viên								
Trả lương cho toàn bộ Tư vấn viên								
Show	10	entries	Search: <input type="text"/>					
Showing 1 to 10 of 20 entries								
Trạng thái	Họ và tên	Giới tính	Lĩnh vực	Trình độ	Số cuộc trò chuyện	Điểm đánh giá	Hành động	
Hoạt động	Nguyễn Trung Anh	Nam	Sức khỏe	Nutritionist	0	3	Xem phản hồi   Trả lương	
Hoạt động	Đỗ Chí Nguyên	Nam	Sức khỏe	Dr	0	3	Xem phản hồi   Trả lương	
Hoạt động	Lê Bảo Đặng	Nam	Sức khỏe	Doctor	0	3	Xem phản hồi   Trả lương	
Hoạt động	Lê Thảo Nguyên	Nữ	Sức khỏe	Masters	0	3	Xem phản hồi   Trả lương	
Hoạt động	Hoàng Như Quỳnh	Nữ	Sức khỏe	Doctor	0	3	Xem phản hồi   Trả lương	
Hoạt động	Trần Yến Quỳnh	Nữ	Sức khỏe	Professor	0	3	Xem phản hồi   Trả lương	
Hoạt động	Đỗ Yến Nhi	Nữ	Sức khỏe	Masters	0	3	Xem phản hồi   Trả lương	
Hoạt động	Nguyễn Lan Anh	Nữ	Sức khỏe	Professor	0	3	Xem phản hồi   Trả lương	
Hoạt động	Trần Tuyết Ngọc	Nữ	Sức khỏe	Consultant	0	3	Xem phản hồi   Trả lương	
Hoạt động	Lý Văn Anh	Nữ	Sức khỏe	Doctor	0	3	Xem phản hồi   Trả lương	

Figure 4-2452: View advisors Screen design

##### Class Diagram

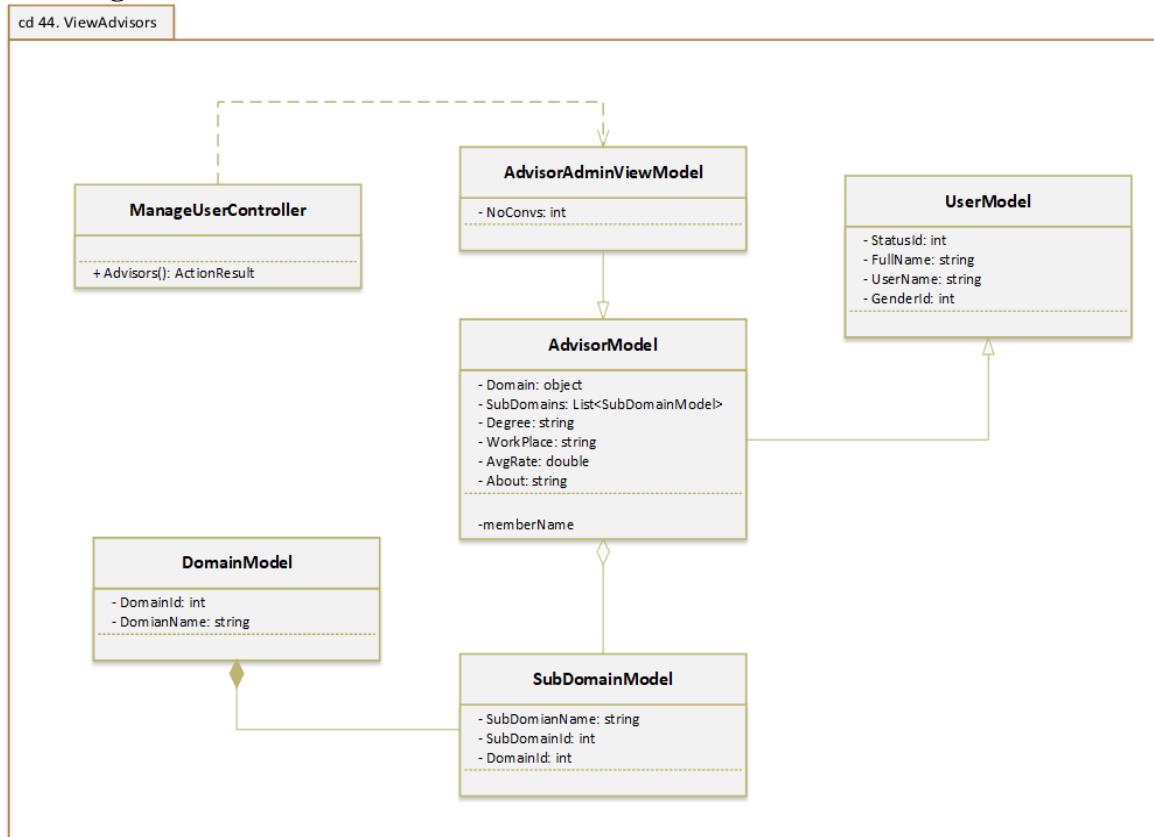


Figure 4-2553: View advisors Class diagram

## Class Specification

ManageUserController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Advisors			
Return Type	ActionResult		
Parameters	Name	Type	Description

AdvisorViewAdminModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	AdvisorModel		
Attributes			
No	Name	Type	Description
1.	NoConvs	int	
Operation			

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	Domain	object	Object contains domain id and domain name.
2.	SubDomains	List<SubDomainModel>	List object contains subdomain name.
3.	Degree	string	
4.	WorkPlace	string	
5.	AvgRate	double	
6.	About	string	
Operation			

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			

No	Name	Type	Description
1.	StatusId	int	
2.	FullName	string	
3.	UserName	string	
4.	GenderId	int	
Operation			

<b>DomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / DomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
Operation			

<b>SubDomainModel</b>			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainName	string	
Operation			

## Sequence Diagram

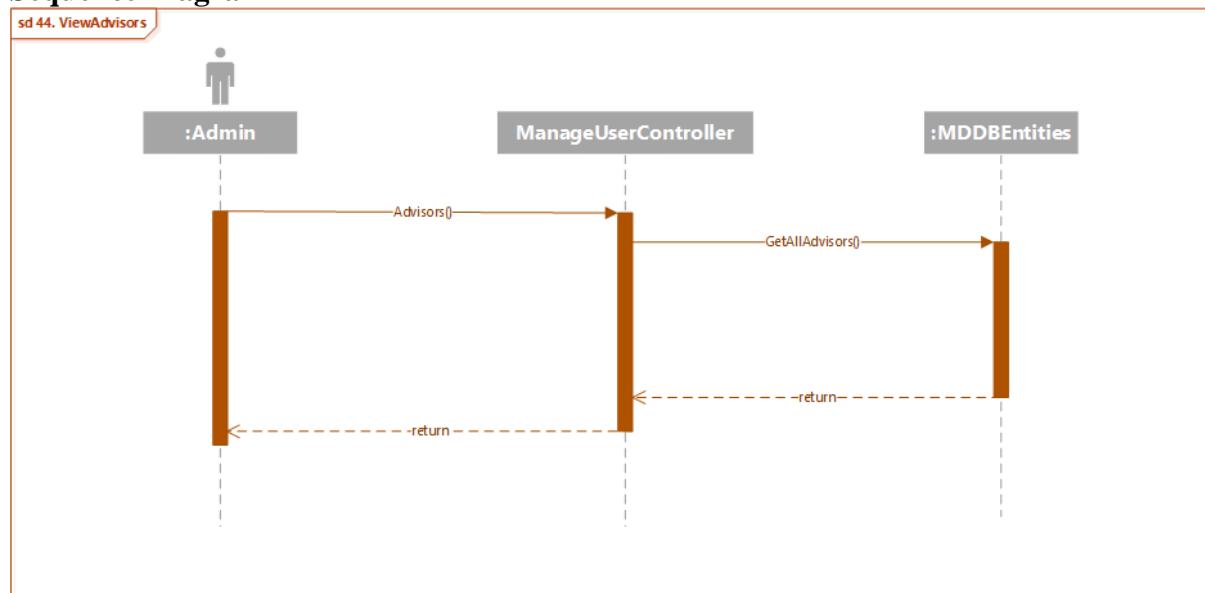


Figure 4-2654: View advisors Sequence diagram

#### 4.3.4.45 View advisor's profile

##### Screen Design

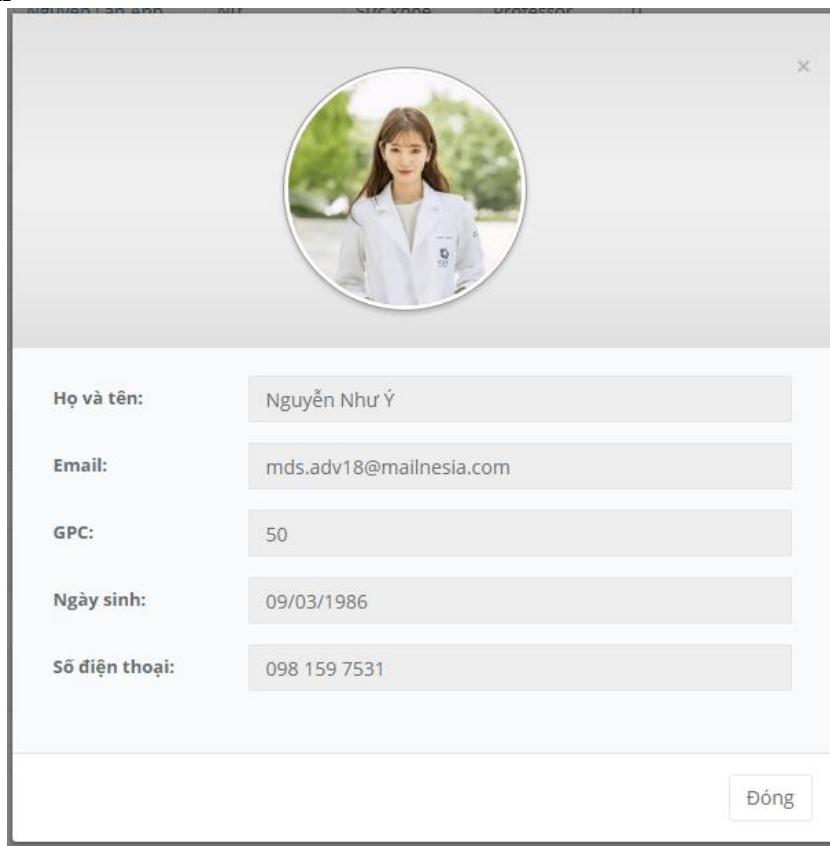


Figure 4-2755: View advisor's profile Screen design

##### Class Diagram

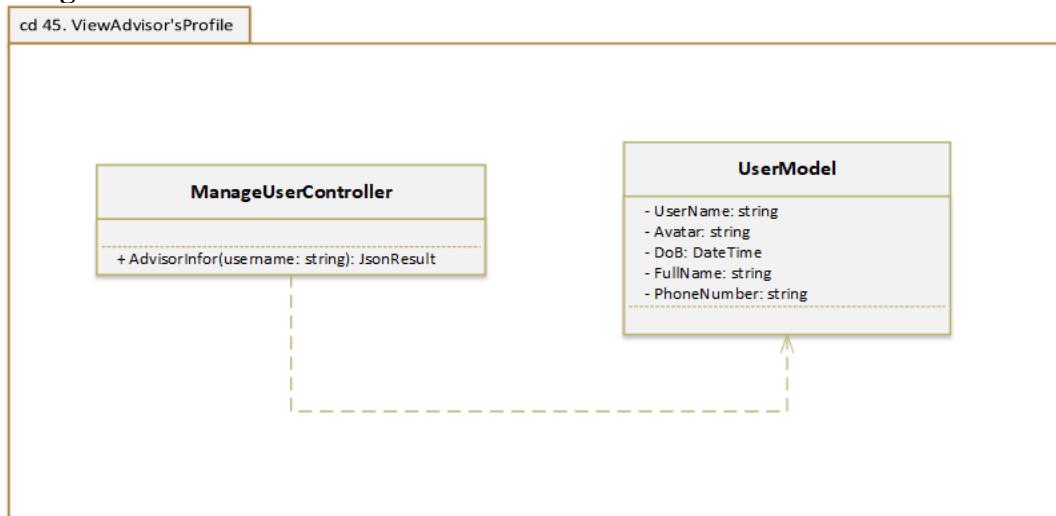


Figure 4-2856: View advisor's profile Class diagram

##### Class Specification

ManageUserController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs
Base class	
Attributes	

No	Name	Type	Description
Operation			
AdvisorInfor			
Return Type	ActionResult		
Parameters	Name	Type	Description
	username	string	Username of advisor.

UserModel			
Physical address	MindDoctor / Models / Models / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Avatar	string	
2.	FullName	string	
3.	UserName	string	
4.	DoB	DateTime	
5.	PhoneNumber	string	
Operation			

## Sequence Diagram

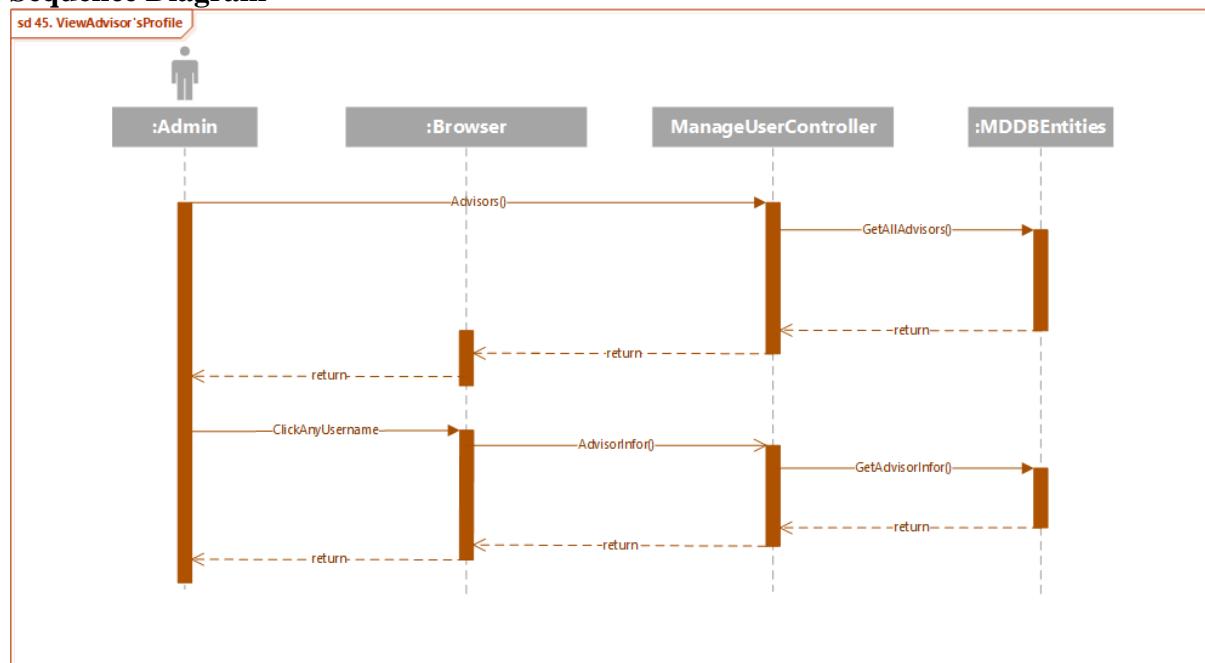


Figure 4-2957: View advisor's profile Sequence diagram

#### 4.3.4.46 Create new Advisor account

##### Screen Design

The screenshot shows a user interface for creating a new Advisor account. At the top left is a search bar labeled 'Bài kiểm tra năng lực'. Below it is an orange button labeled 'TÌM KIẾM'. The main form contains the following fields:

- Ngày sinh:** 07 Tháng Tám 1999
- Giới tính:** Nữ
- Email:** md.adv22@mailnesia.com
- Số điện thoại:** 096 623 6917
- Lĩnh vực:** Tâm lý
- Chuyên mục:** Hôn nhân gia đình

At the bottom right is a teal button labeled 'Đăng ký Tư vấn viên'.

Figure 4-3058: Create new Advisor account Screen design

##### Class Diagram

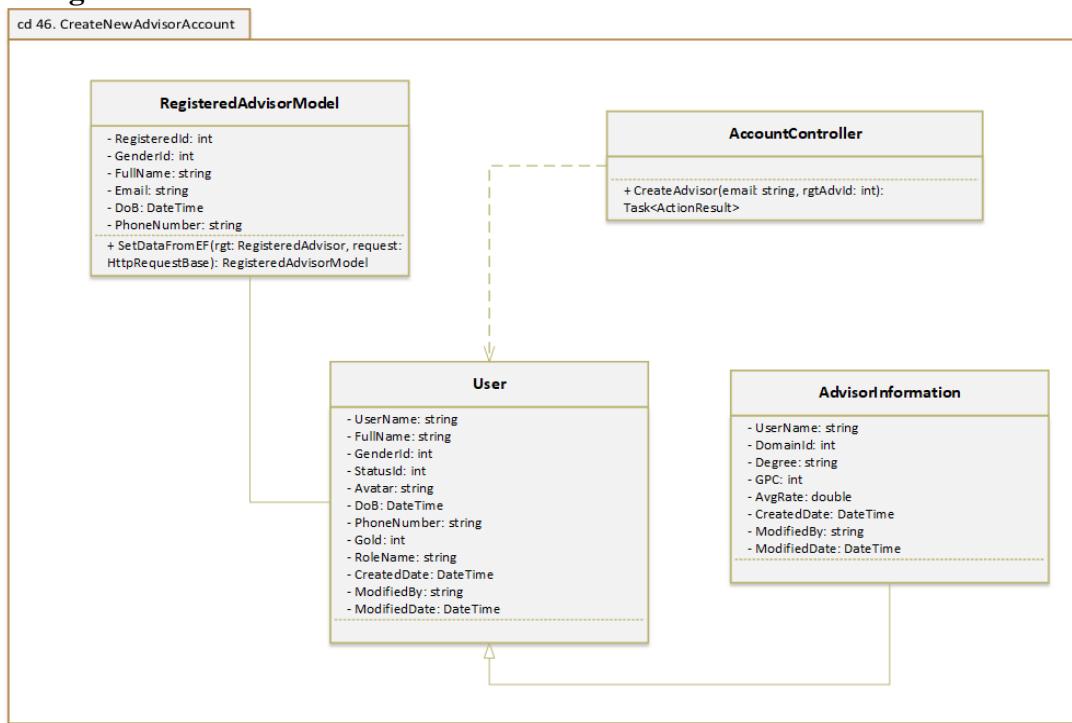


Figure 4-31: Create new Advisor account Class diagram

##### Class Specification

AccountController			
Physical address	MindDoctor / Controllers / AccountController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			

CreateAdvisor			
Return Type	Task<ActionResult>		
Parameters	Name	Type	Description
	email	string	Email of registered user.
	rgtAdvId	int	Registered user id.

RegisteredAdvisorModel			
Physical address	MindDoctor / Models / RegisteredAdvisorModels / RegisteredAdvisorModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	RegisterdId	int	
2.	Domain	object	Object contains domain id and domain name
3.	Username	string	
4.	FullName	string	
5.	Email	string	
6.	DoB	DateTime	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	GenderId	int	
4.	StatusId	int	
5.	Avatar	string	
6.	DoB	DateTime	
7.	PhoneNumber	string	
8.	Gold	int	
9.	RoleName	string	
10.	CreatedDate	DateTime	
11.	ModifiedBy	string	
12.	ModifiedDate	DateTime	
Operation			

AdvisorInformation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / AdvisorInformation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	DomainId	int	
3.	Degree	string	
4.	GPC	int	
5.	AvgRate	double	
6.	CreatedDate	DateTime	
7.	ModifiedBy	string	
8.	ModifiedDate	DateTime	
Operation			

## Sequence Diagram

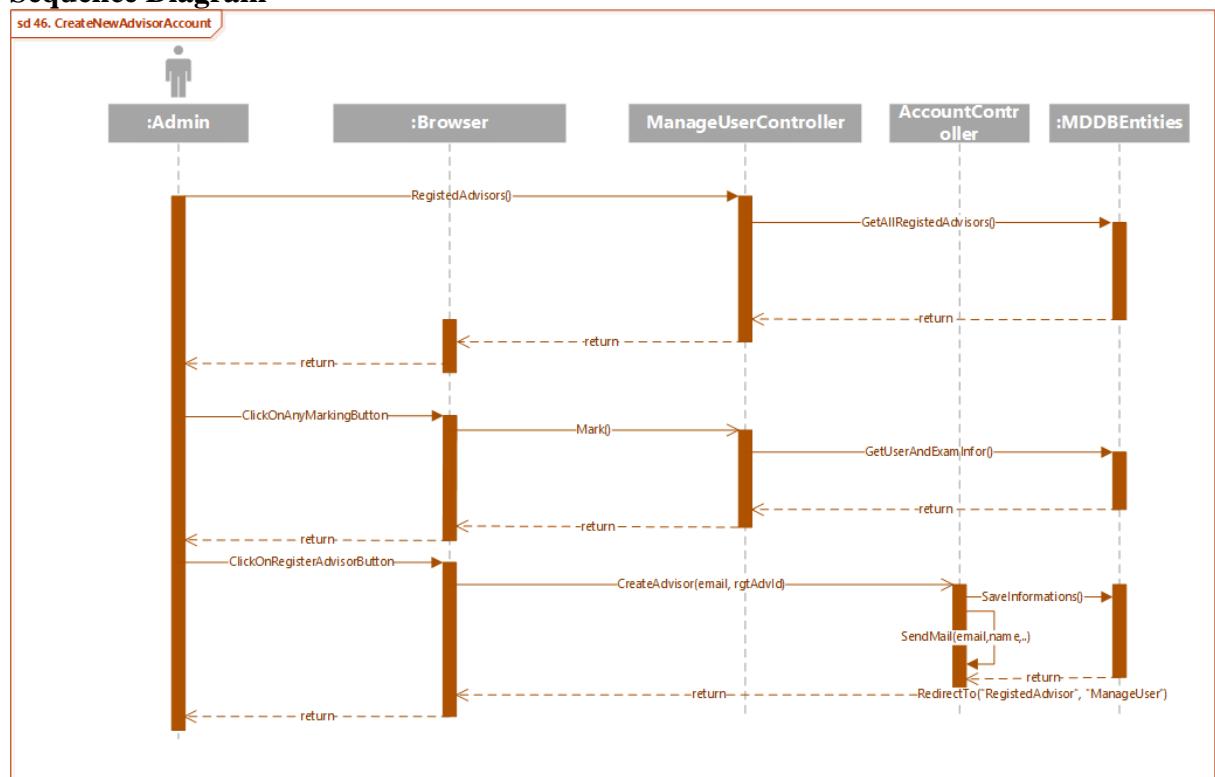


Figure 4-3260: Create new Advisor account Sequence diagram

#### 4.3.4.47 Mark exam for Register advisor

##### Screen Design

Bài kiểm tra năng lực

Câu 1/1      Liệt kê một số cách để làm cho hết nấc cụt?



Uống nước bằng những ngụm nhỏ,  
Bịt chặt lỗ tai trong 3 phút sau đó uống vài ngụm nước lạnh

**Tính điểm**

**Xóa bài**

Họ và tên	Bành Thị Thu Yến
Ngày sinh	07 Tháng Tám 1999
Giới tính	Nữ
Email	md.adv22@mailnesia.com

Figure 4-3361: Mark exam for Register advisor Screen design

##### Class Diagram

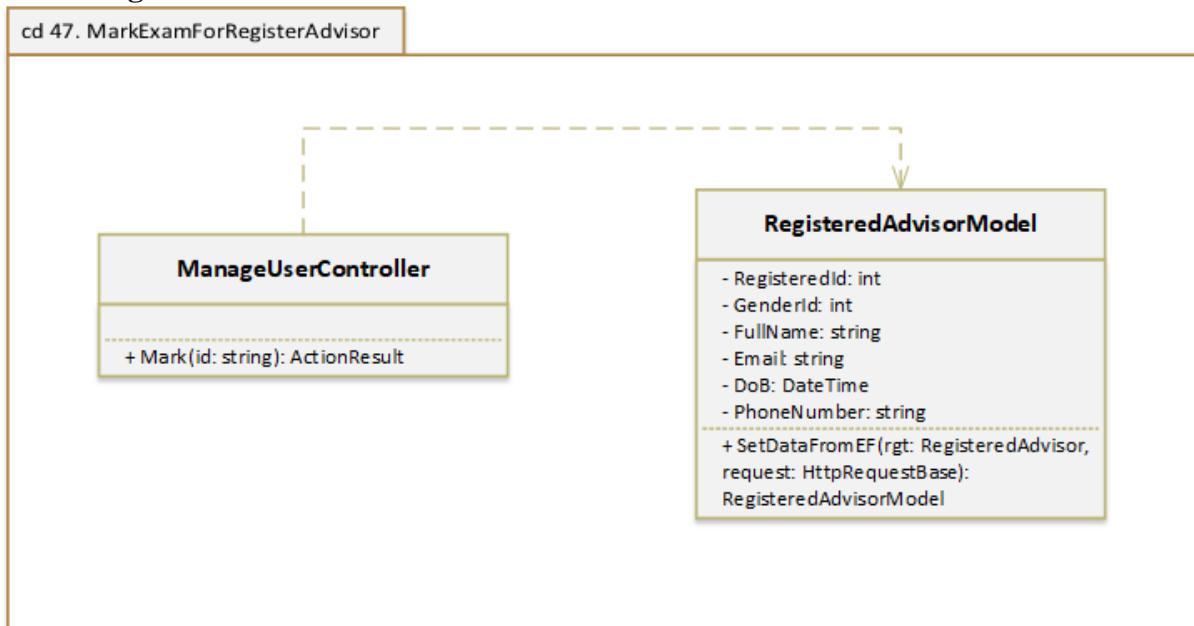


Figure 4-3462: Mark exam for Register advisor Class diagram

##### Class Specification

ManageUserController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs
Base class	
Attributes	

No	Name	Type	Description
Operation			
RegisteredAdvisors			
Return Type	ActionResult		
Parameters	Name	Type	Description
	email	string	Email of registered user.
Mark			
Return Type	ActionResult		
Parameters	Name	Type	Description
	id	string	Registered user id.

RegisteredAdvisorModel			
Physical address	MindDoctor / Models / RegisteredAdvisorModels / RegisteredAdvisorModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	rgtAdv	object	
2.	request	object	
Operation			

## Sequence Diagram

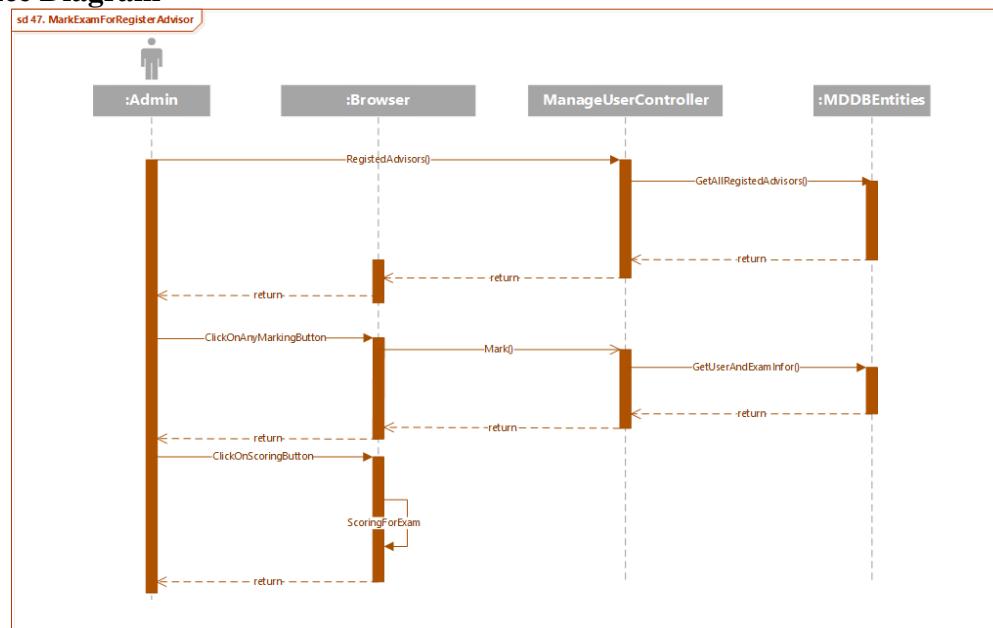


Figure 4-3563: Mark exam for Register advisor Sequence diagram

#### 4.3.4.48 Block account

##### Screen Design

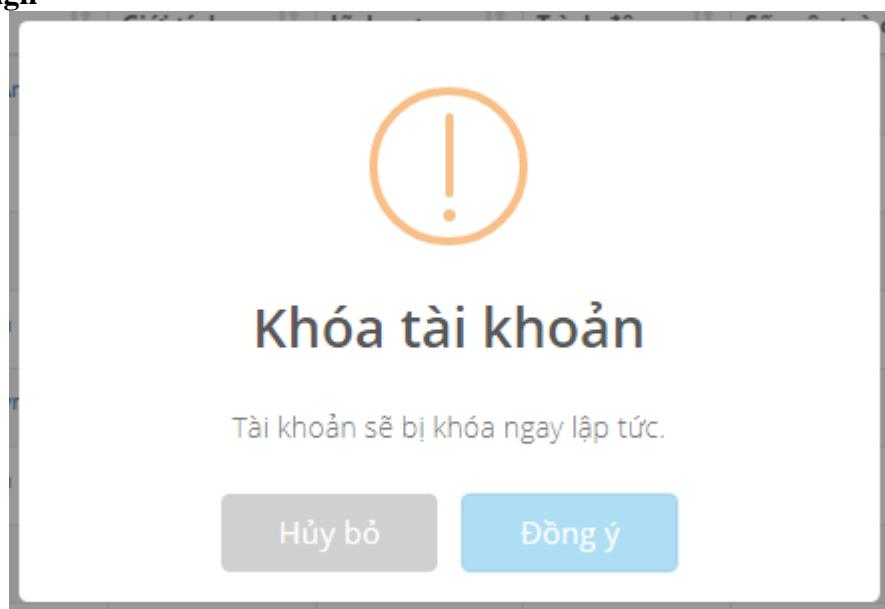


Figure 4-3664: Block account Screen design

##### Class Diagram

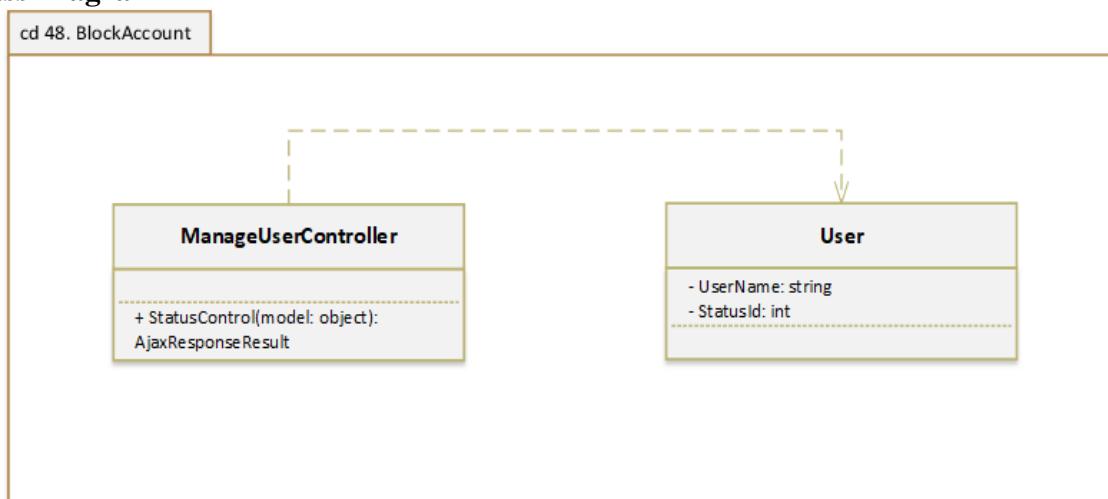


Figure 4-3765: Block account Class diagram

##### Class Specification

ManageUserController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
StatusControl			
Return Type	JsonResult		
Parameters	Name	Type	Description

	model	object	Object contains id and user name.
--	-------	--------	-----------------------------------

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	StatusId	int	
Operation			

## Sequence Diagram

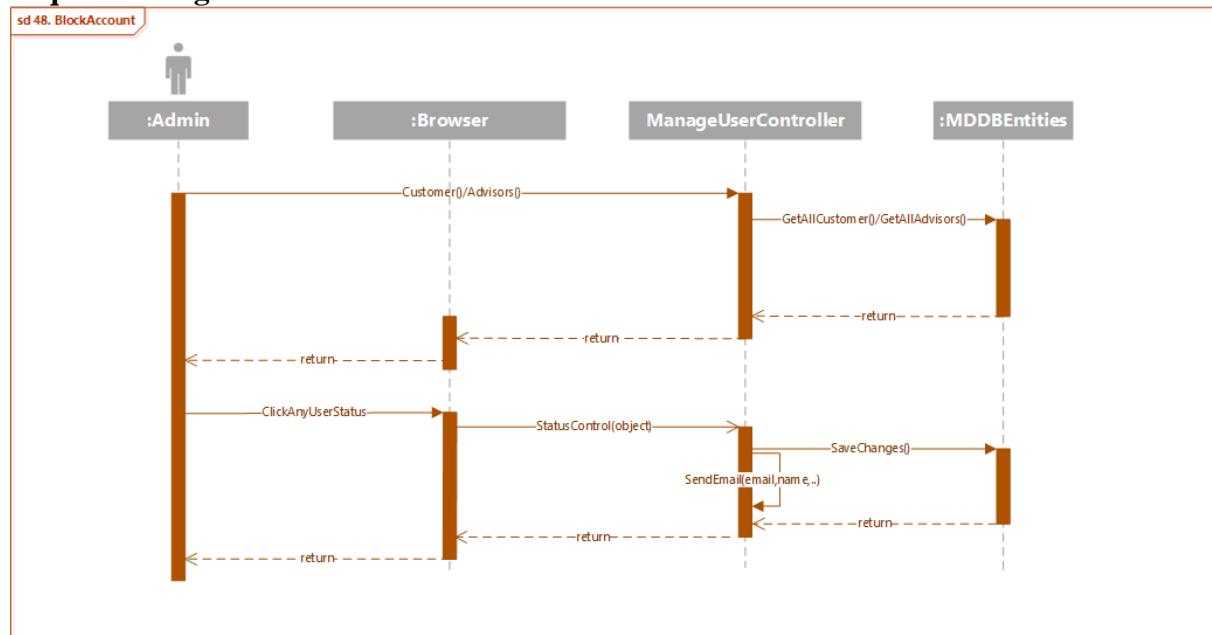


Figure 4-3866: Block account Sequence diagram

#### 4.3.4.49 Unblock account

##### Screen Design

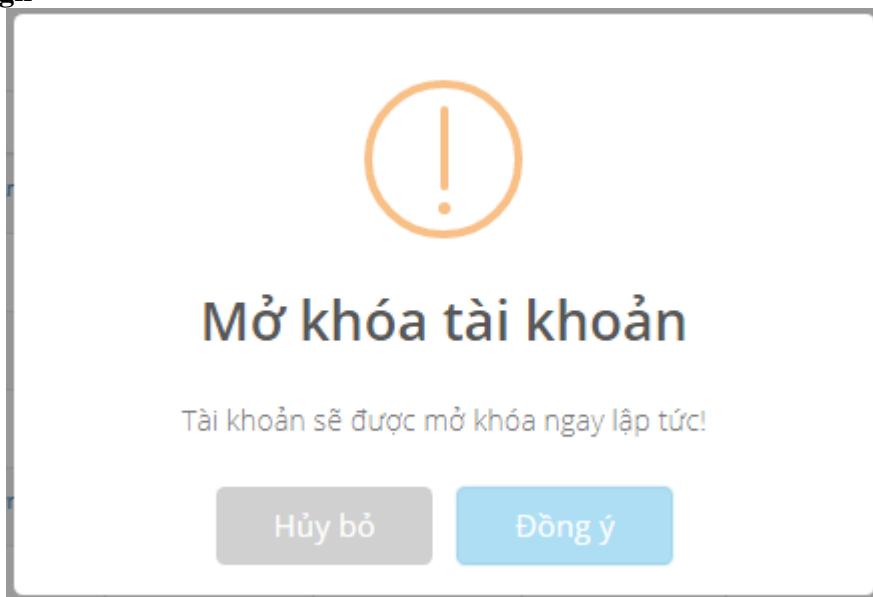


Figure 4-3967: Unblock account Screen design

##### Class Diagram

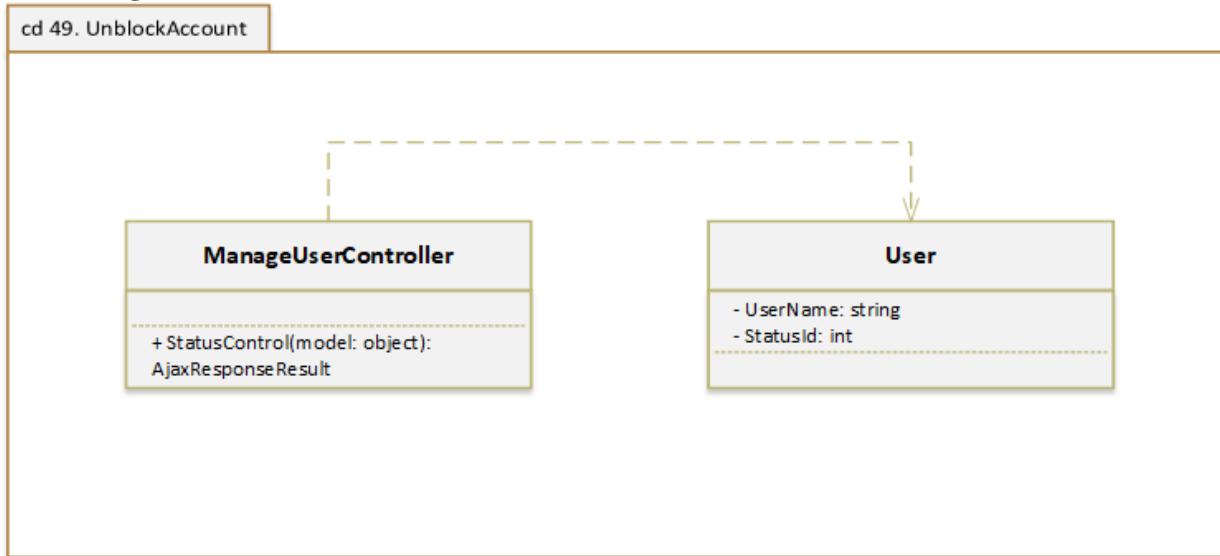


Figure 4-4068: Unblock account Class diagram

##### Class Specification

ManageUserController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
StatusControl			
Return Type	JsonResult		

Parameters	Name	Type	Description
	model	object	Object contains id and user name.

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	StatusId	int	
Operation			

## Sequence Diagram

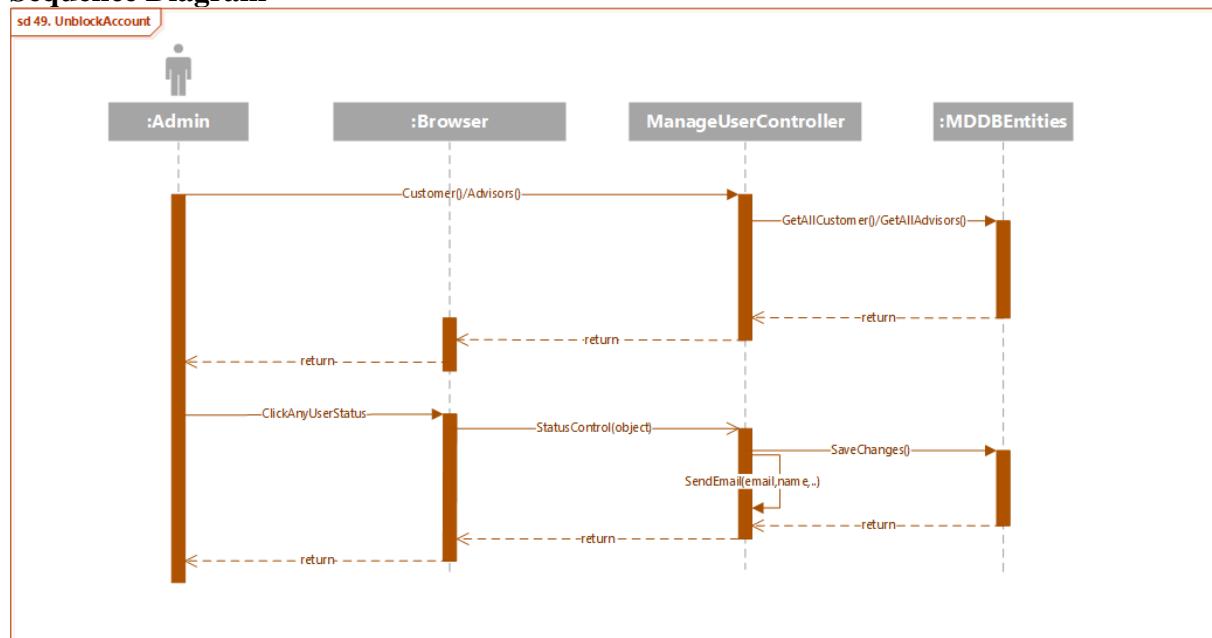


Figure 4-169: Unblock account Sequence diagram

#### 4.3.4.50 View domain

##### Screen Design



Figure 4-170: View domain Screen design

##### Class Diagram

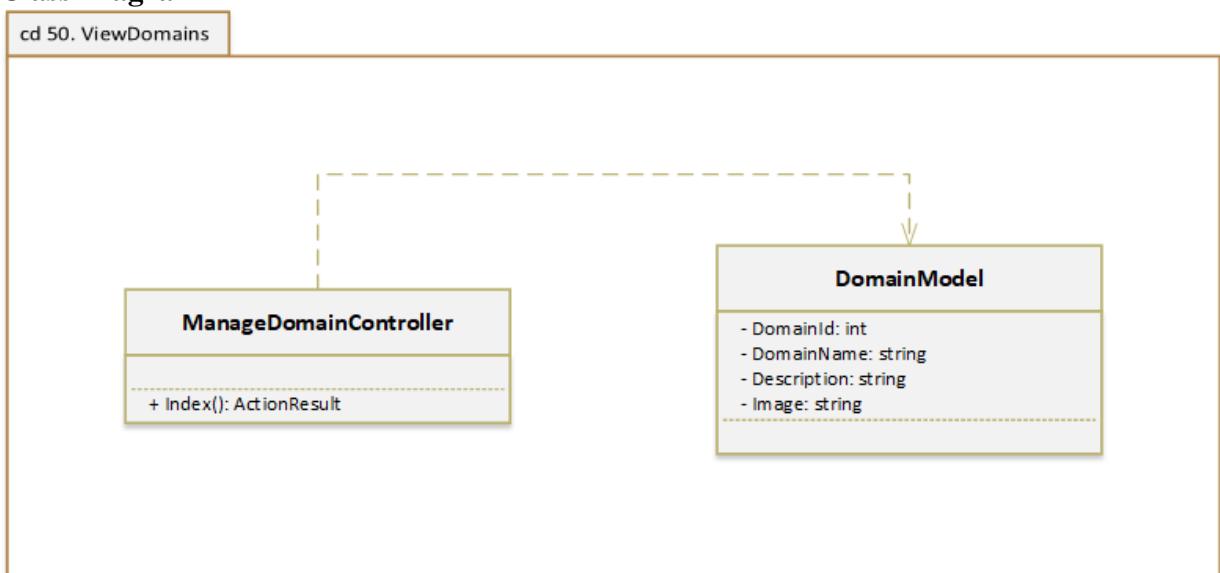


Figure 4-4171: View domain Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Index			
Return Type	ActionResult		

Parameters	Name	Type	Description
------------	------	------	-------------

DomainModel			
Physical address	MindDoctor / Models / DomainModels / DomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
3.	Description	string	
4.	Image	string	
Operation			

### Sequence Diagram

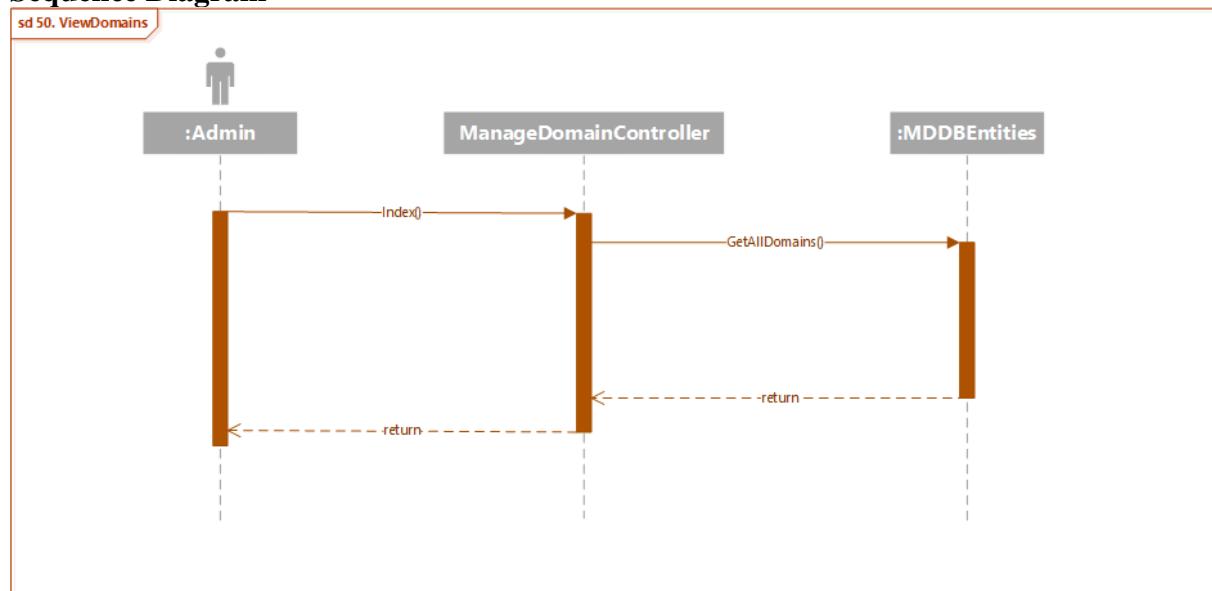


Figure 4-4272: View domain Sequence diagram

#### 4.3.4.51 Update domain

##### Screen Design

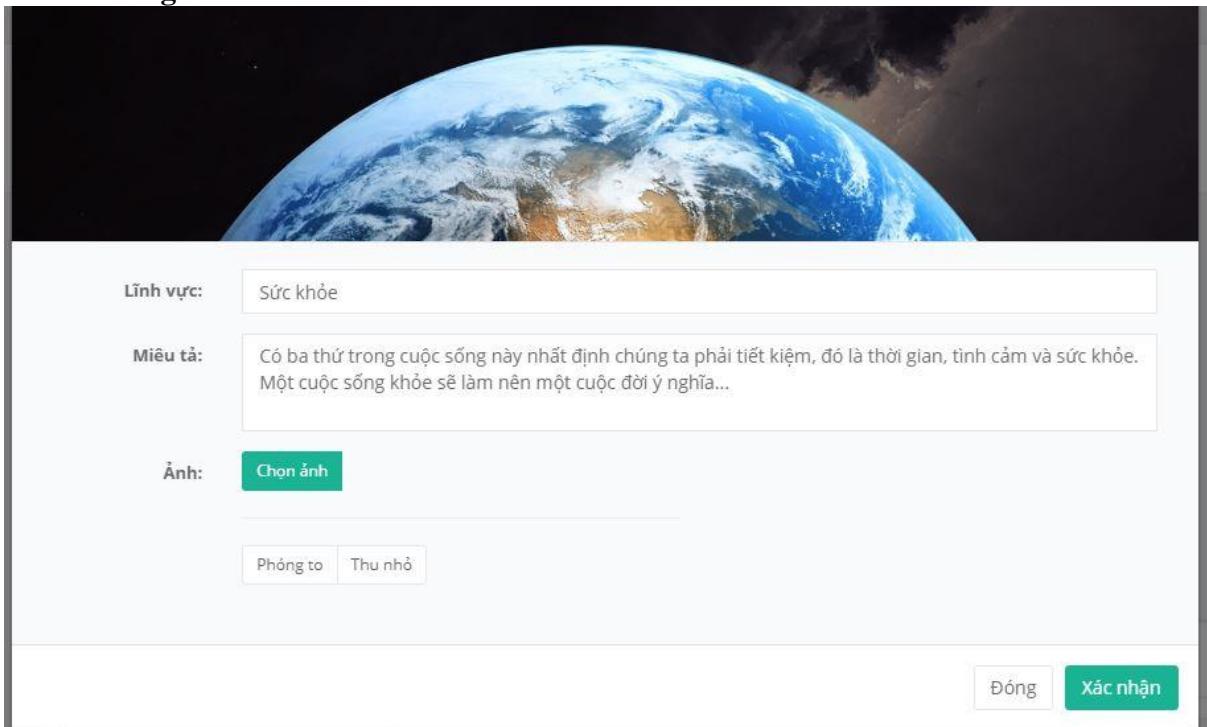


Figure 4-4373: Update domain Screen design

##### Class Diagram

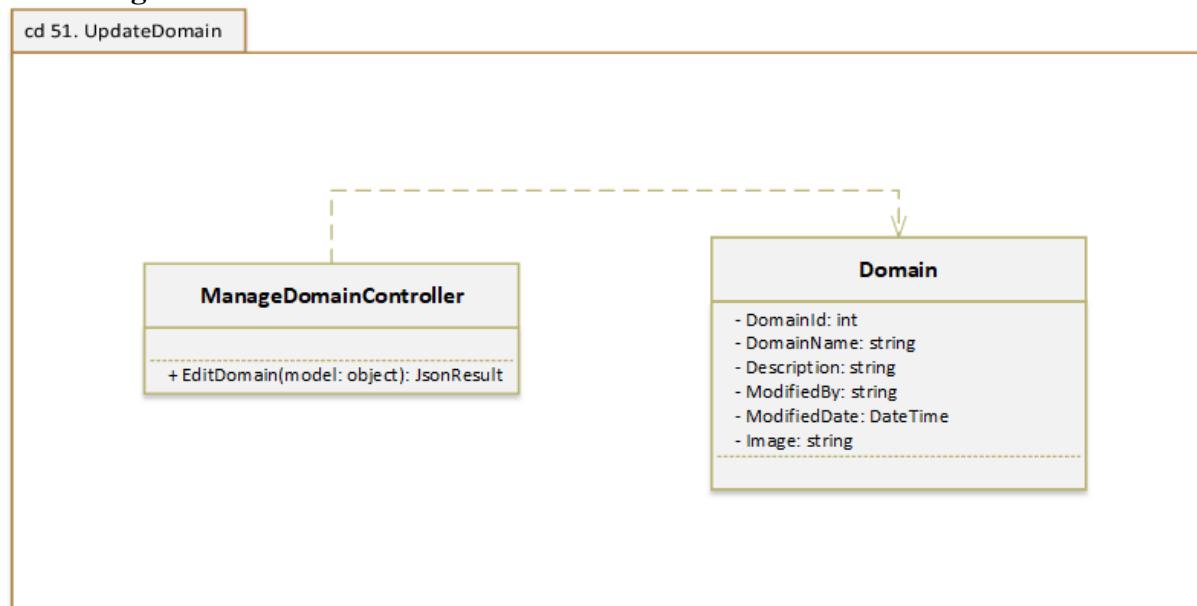


Figure 4-4474: Update domain Class diagram

##### Class Specification

ManageDomainController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs
Base class	
Attributes	

No	Name	Type	Description
Operation			
EditDomain			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Object contains domain name, description,....

Domain			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Domain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainName	string	
2.	Description	string	
3.	ModifiedBy	string	
4.	ModifiedDate	DateTime	
5.	Image	string	
Operation			

## Sequence Diagram

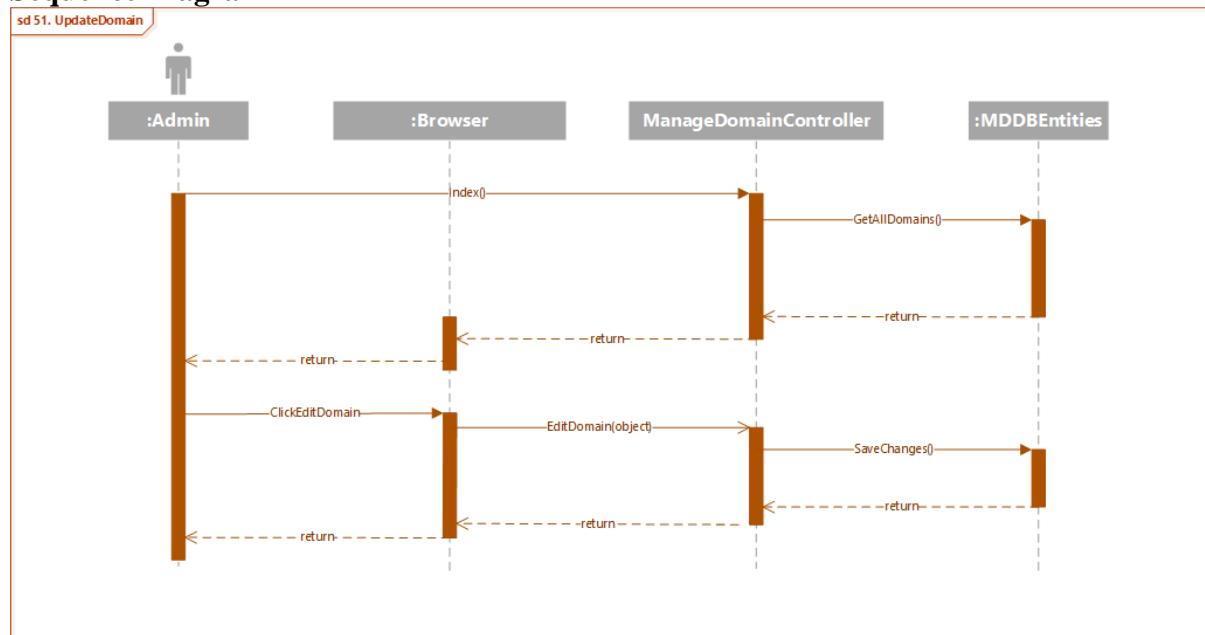
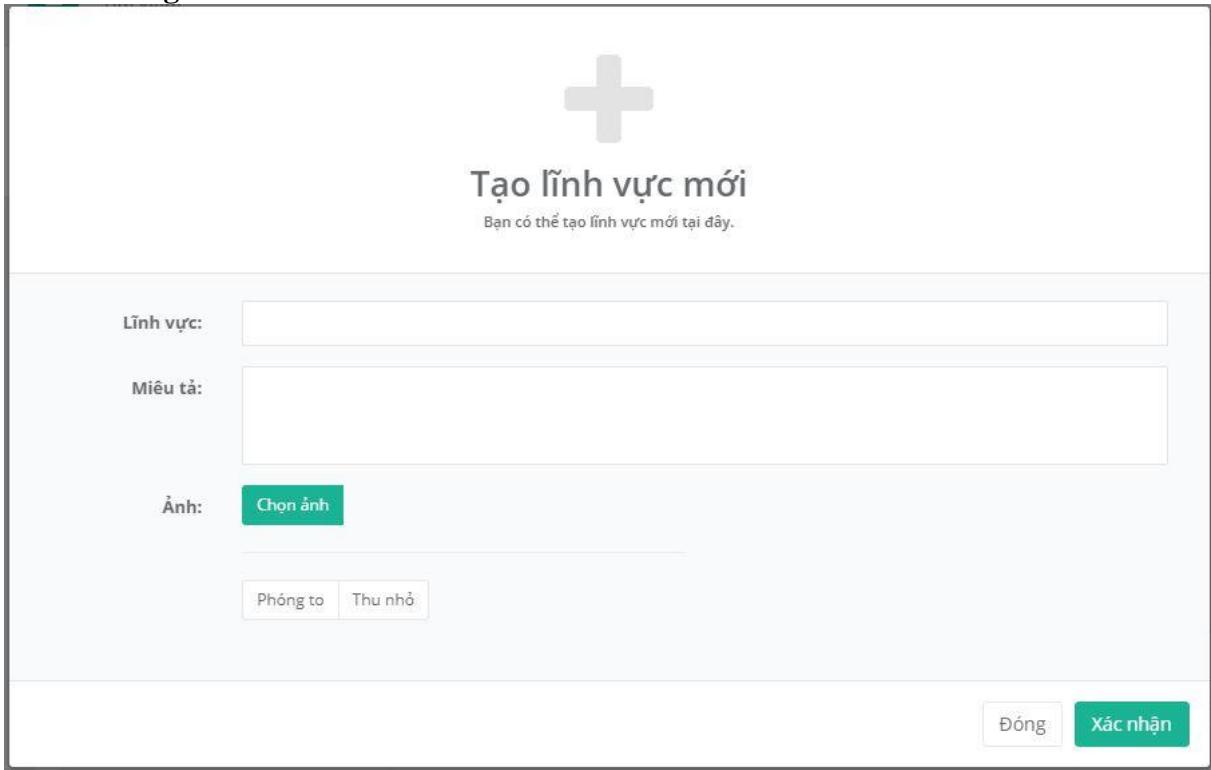


Figure 4-4575: Update domain Sequence diagram

#### 4.3.4.52 Create new domain

##### Screen Design



The screenshot shows a user interface for creating a new domain. At the top center is a large gray plus sign icon. Below it, the text "Tạo lĩnh vực mới" (Create new category) is displayed in bold. Underneath that, a smaller text says "Bạn có thể tạo lĩnh vực mới tại đây." (You can create new categories here). There are two input fields: one labeled "Lĩnh vực:" (Category:) with a placeholder text field, and another labeled "Miêu tả:" (Description:) with a larger text area. Below these fields is a section for "Ảnh:" (Image) with a "Chọn ảnh" (Select image) button. At the bottom left are two buttons: "Phóng to" (Zoom in) and "Thu nhỏ" (Zoom out). At the bottom right are two buttons: "Đóng" (Close) and a green "Xác nhận" (Confirm) button.

Figure 4-4676: Create new domain Screen design

##### Class Diagram

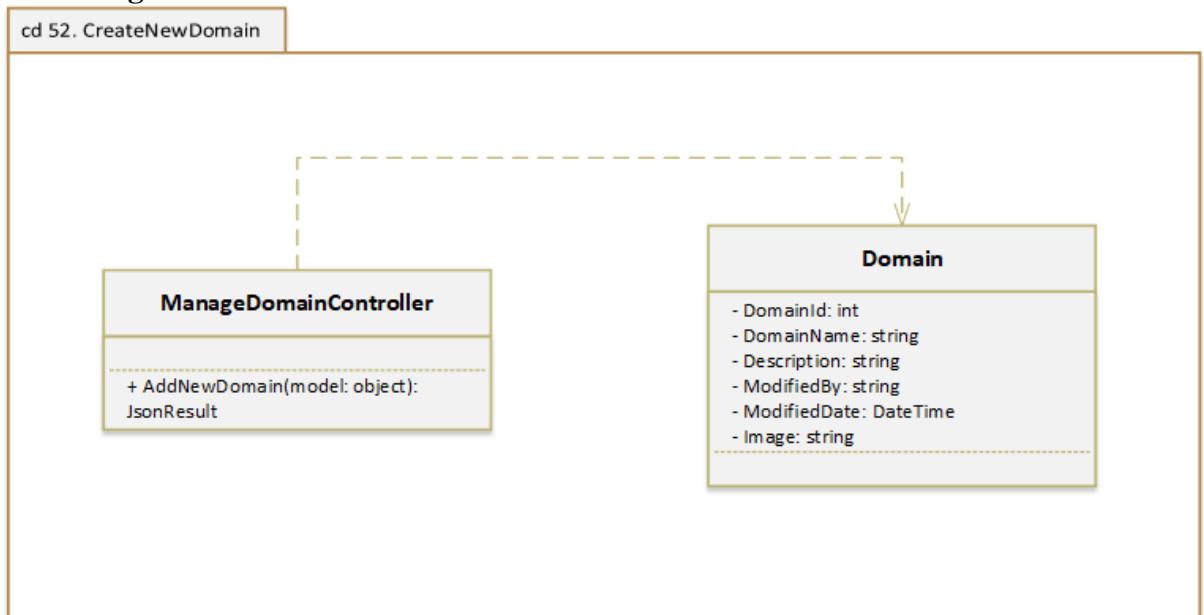


Figure 4-4777: Create new domain Class diagram

##### Class Specification

ManageDomainController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs
Base class	

Attributes			
No	Name	Type	Description
Operation			
AddNewDomain			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	model	object	Object contains domain name, description,....

Domain			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Domain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainName	string	
2.	Description	string	
3.	ModifiedBy	string	
4.	ModifiedDate	DateTime	
5.	Image	string	
Operation			

## Sequence Diagram

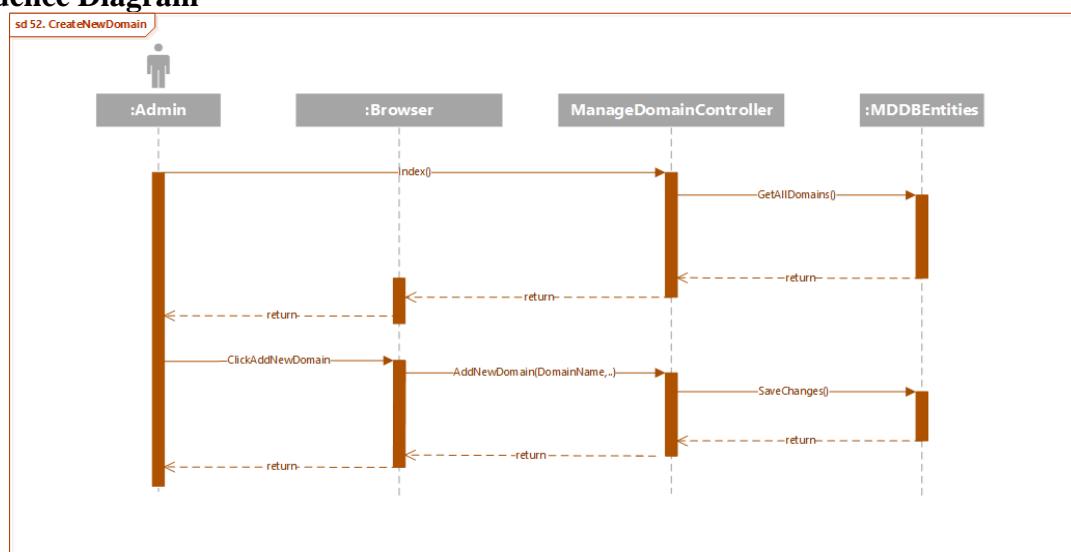


Figure 4-4878: Create new domain Sequence diagram

#### 4.3.4.53 View sub-domain

##### Screen Design

Bảng dữ liệu lĩnh vực "Sức khỏe"		Tạo một chuyên khoa mới							
Show	10	entries	Search:		Copy	CSV	Excel	PDF	Print
Showing 1 to 1 of 1 entries									
Chuyên khoa		Số lượng tư vấn viên		Hành động					
Sức khỏe giới tính		8		Sửa					
Chuyên khoa		Số lượng tư vấn viên		Hành động					

Previous | 1 | Next

Figure 4-4979: View sub-domain Screen design

##### Class Diagram

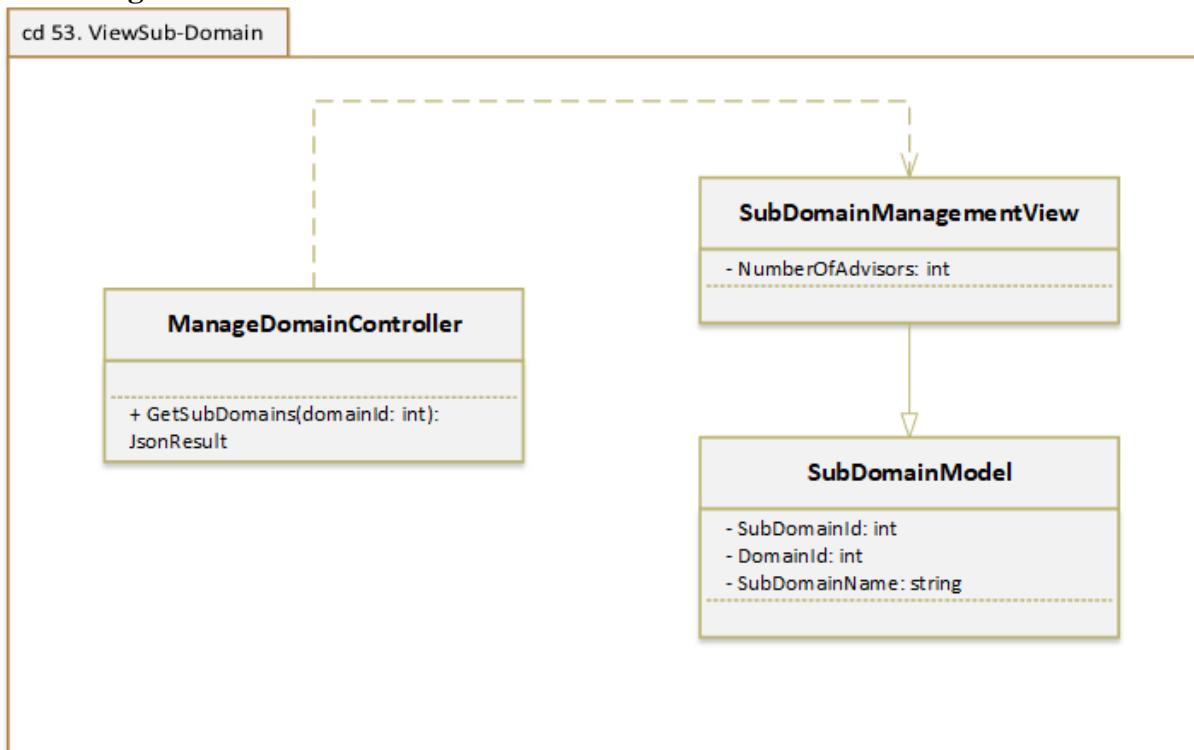


Figure 4-5080: View sub-domain Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
GetSubDomains			
Return Type	JsonResult		
Parameters	Name	Type	Description
	domainId	int	This field to find all

			sub-domain corresponding to domain.
--	--	--	-------------------------------------

SubDomainManagementView			
Physical address	MindDoctor / Areas / Admin / Models / SubDomainManagementView.cs		
Base class	SubDomainModel		
Attributes			
No	Name	Type	Description
1.	NumberOfAdvisors	int	
Operation			

SubDomainModel			
Physical address	MindDoctor / Models / DomainModels / SubDomainModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	DomainId	int	
3.	SubDomainName	string	
Operation			

## Sequence Diagram

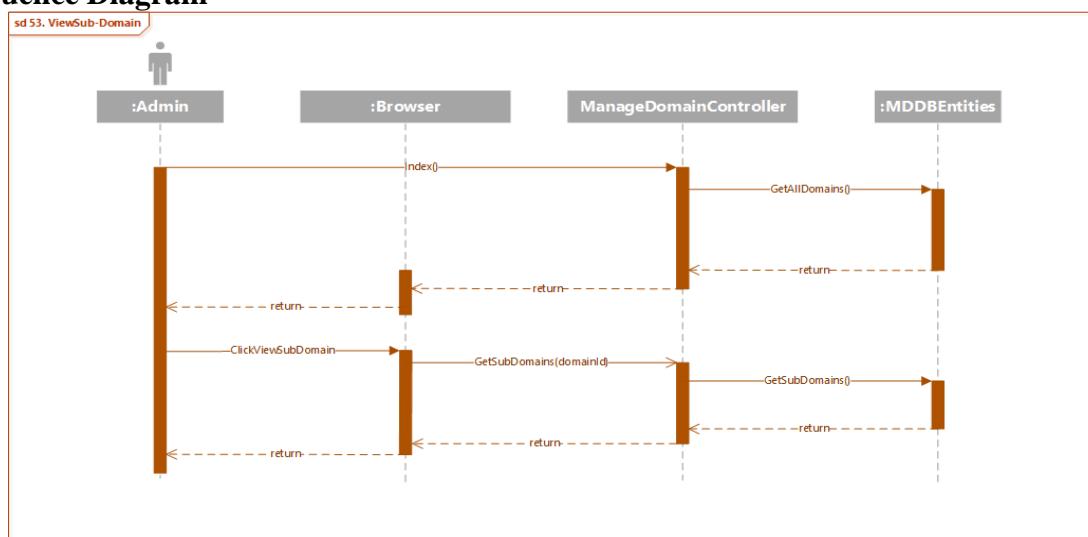


Figure 4-5181: View sub-domain Sequence diagram

#### 4.3.4.54 Update sub-domain

##### Screen Design



Figure 4-5282: Update sub-domain Screen design

##### Class Diagram

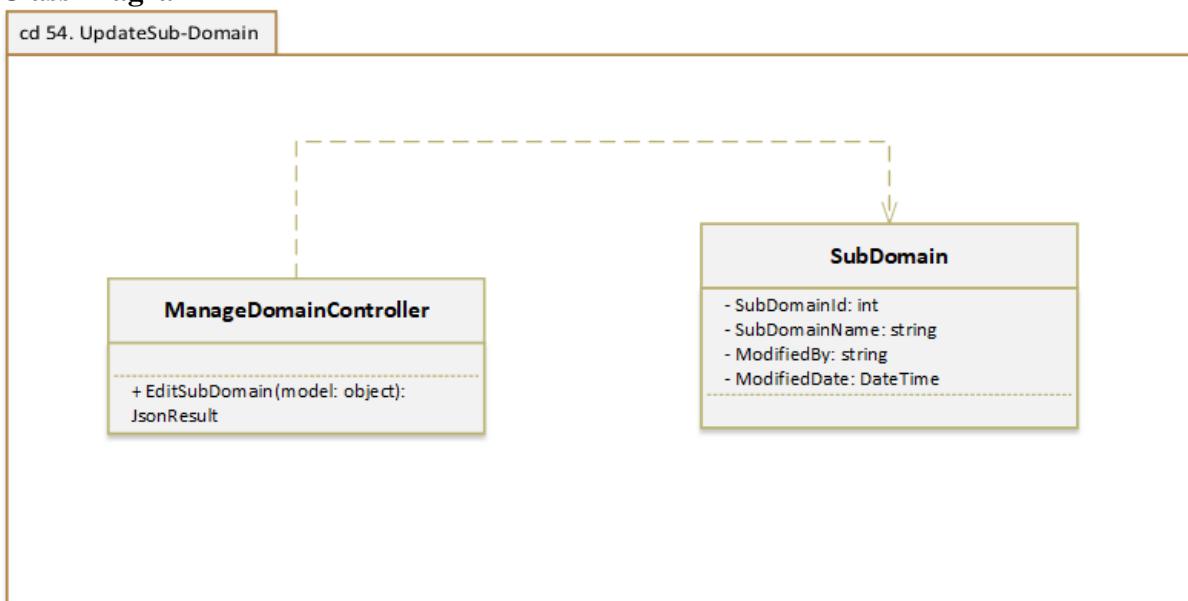


Figure 4-5383: Update sub-domain Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			

EditSubDomain			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Object contains sub domain id and sub domain name.

SubDomain			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / SubDomain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
3.	ModifiedBy	string	
4.	ModifiedDate	DateTime	
Operation			

### Sequence Diagram

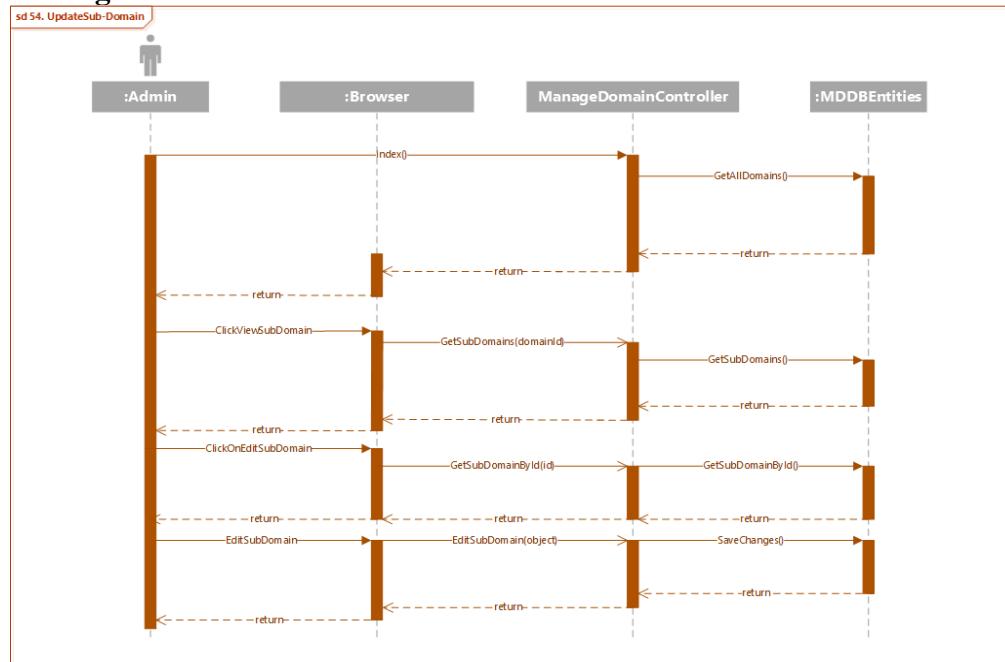


Figure 4-5484: Update sub-domain Sequence diagram

#### 4.3.4.55 Create new sub-domain

##### Screen Design

Tạo chuyên khoa mới cho "Tâm lý"

Bạn có thể tạo chuyên khoa mới tại đây.

Chuyên khoa:

Đóng Xác nhận

Figure 4-5585: Create new sub-domain Screen design

##### Class Diagram

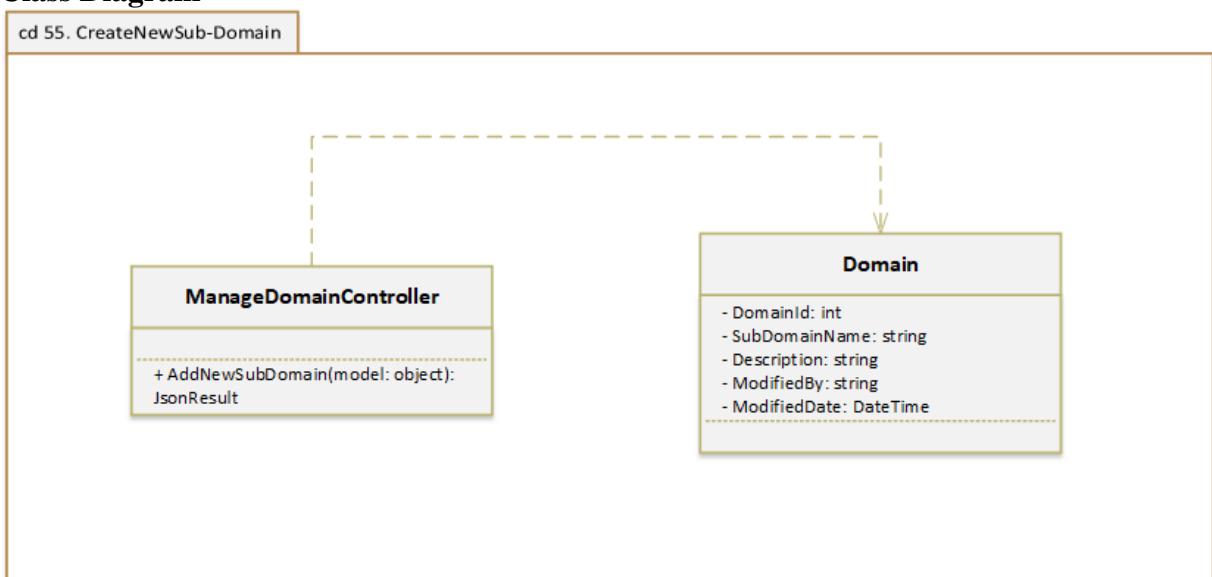


Figure 4-5686: Create new sub-domain Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
No	Name	Type	Description
No	Name	Type	Description

AddNewSubDomain			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	model	object	Object contains domain id, sub-domain name,...

Domain			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Domain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	SubDomainName	string	
3.	Description	string	
4.	ModifiedBy	string	
5.	ModifiedDate	DateTime	
Operation			

## Sequence Diagram

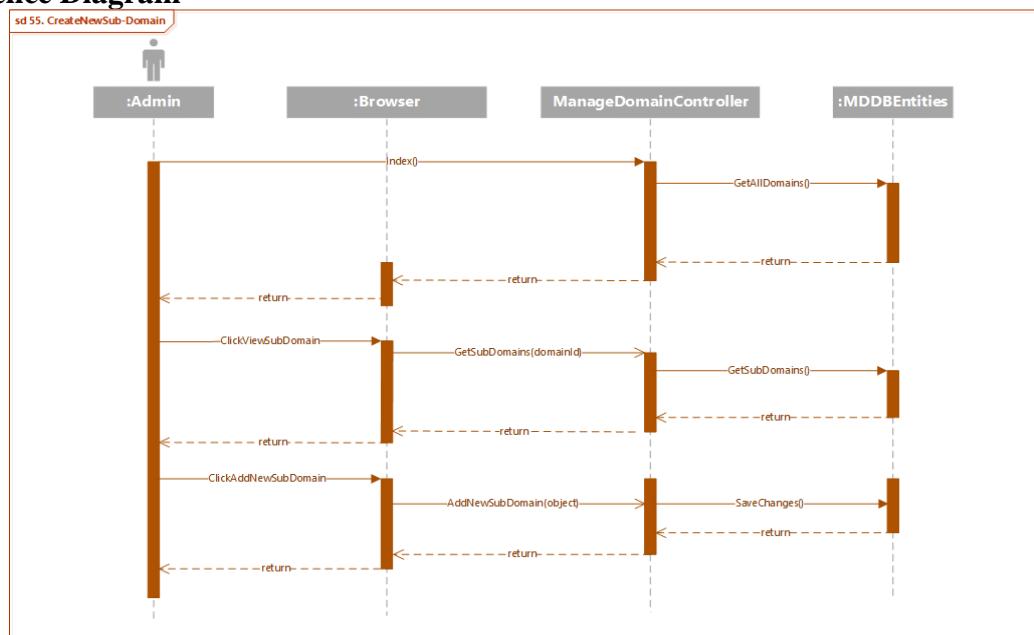


Figure 4-5787: Create new sub-domain Sequence diagram

#### 4.3.4.56 Create new post

##### Screen Design

Tạo bài đăng mới

Lĩnh vực: Sức khỏe

Chuyên mục: Sức khỏe giới tính

Ngôn ngữ: Vietnamese

Tiêu đề:

Giới thiệu:

Nội dung:

Ảnh: Phương pháp thông thường

Tải ảnh lên

Tùy chọn khác

Phóng to Thu nhỏ

Đăng bài

Figure 4-5888: Create new post Screen design

##### Class Diagram

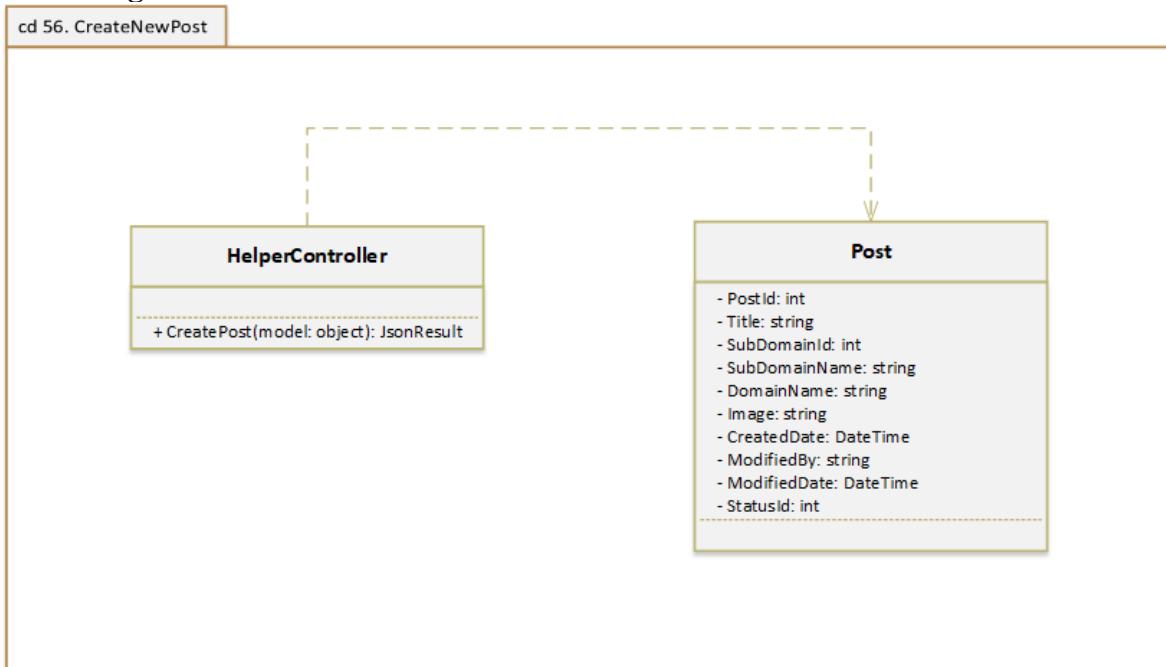


Figure 4-5989: Create new post Class diagram

##### Class Specification

HelperController	
Physical address	MindDoctor / Areas / Admin / Controllers / HelperController.cs
Base class	
Attributes	

No	Name	Type	Description
Operation			
CreatePost			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Object contains sub-domain name, domain name,....

Post			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Post.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	PostId	int	
2.	Title	string	
3.	SubDomainId	int	
4.	SubDomainName	string	
5.	DomainName	string	
6.	Image	string	
7.	CreatedDate	DateTime	
8.	ModifiedBy	string	
9.	ModifiedDate	DateTime	
10.	StatusId	int	
Operation			

## Sequence Diagram

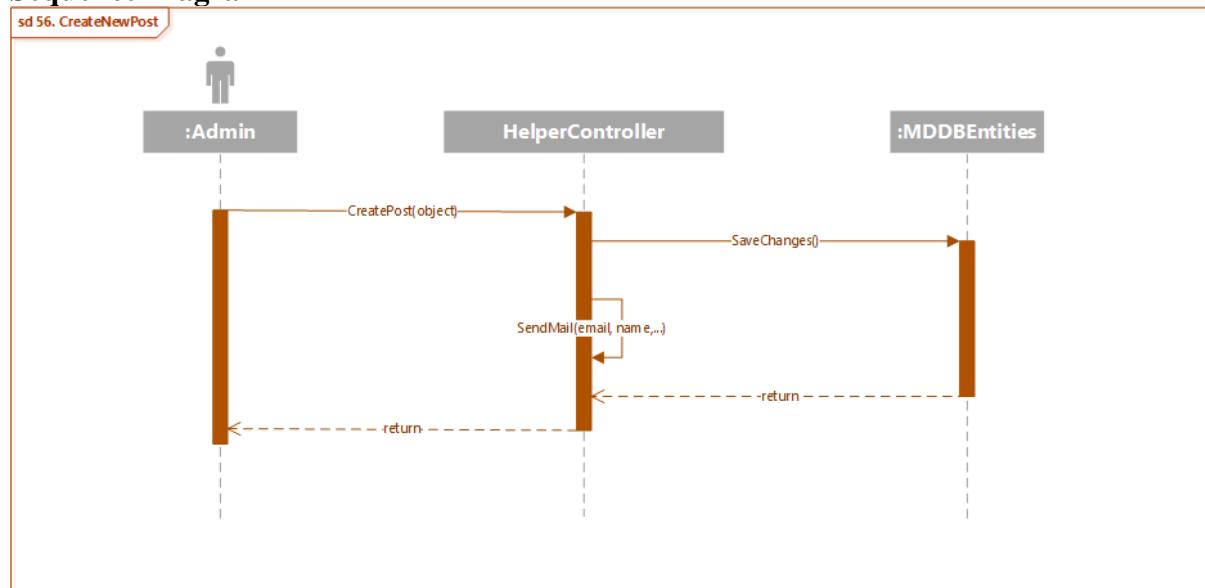


Figure 4-6090: Create new post Sequence diagram

#### 4.3.4.57 Edit post

##### Screen Design

Chỉnh sửa bài đăng

Tiêu đề:	Học sinh lớp 11 cần chuẩn bị gì từ bây giờ?
Giới thiệu:	Khi các đàn anh lớp 12 đã hoàn thành xong kỳ thi THPT, thì cũng là lúc "đàn em" lớp 11 bắt đầu lên kế hoạch chuẩn bị cho năm sau.
Nội dung:	<p><b>Tranh thủ học hè để lập những "lỗ hổng" kiến thức</b></p> <p>Nếu như những ngày lớp 10 lên lớp 11, học sinh vẫn có thể dành thời gian trong hè cho những kế hoạch vui chơi, nhảy múa tung bừng thì hè này thời gian không còn được thoải mái như vậy. Đa phần các bạn học sinh lớp 11 chuẩn bị lên lớp 12 đều có những dự định học tập cho riêng mình nhằm chuẩn bị cho kì thi THPT Quốc gia 2018 đã cận kề. Vậy nên, hè này hãy tranh thủ học hành để lập những lỗ hổng kiến thức nhé các teen 11. Vì bước vào đầu năm học chúng ta sẽ có rất nhiều việc cần chuẩn bị nữa đó!</p> <p><b>Tìm hiểu về các trường đại học</b></p>

Figure 4-6191: Edit post Screen design

##### Class Diagram

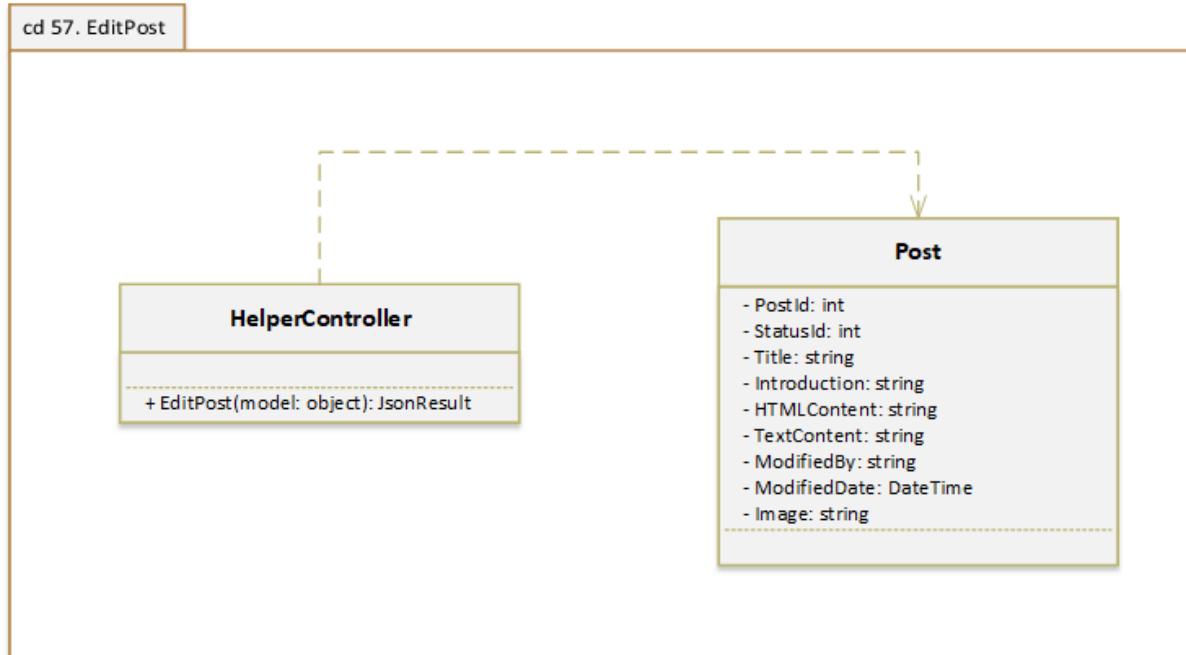


Figure 4-6292: Edit post Class diagram

##### Class Specification

HelperController			
Physical address	MindDoctor / Areas / Admin / Controllers / HelperController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			

EditPost			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	model	object	Object contains title, description...

Post			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Post.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Title	string	
2.	Introduction	string	
3.	HTMLContent	string	
4.	TextContent	string	
6.	Image	string	
8.	ModifiedBy	string	
9.	ModifiedDate	DateTime	
Operation			

## Sequence Diagram

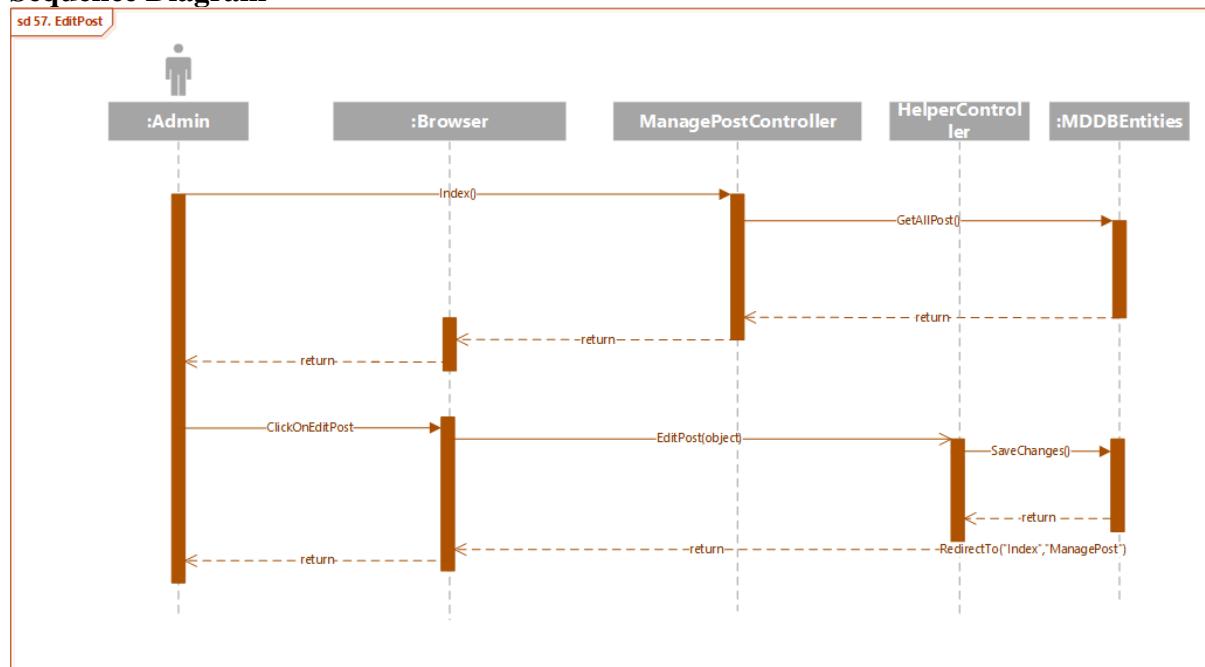


Figure 4-6393: Edit post Sequence diagram

#### 4.3.4.58 Delete post

##### Screen Design

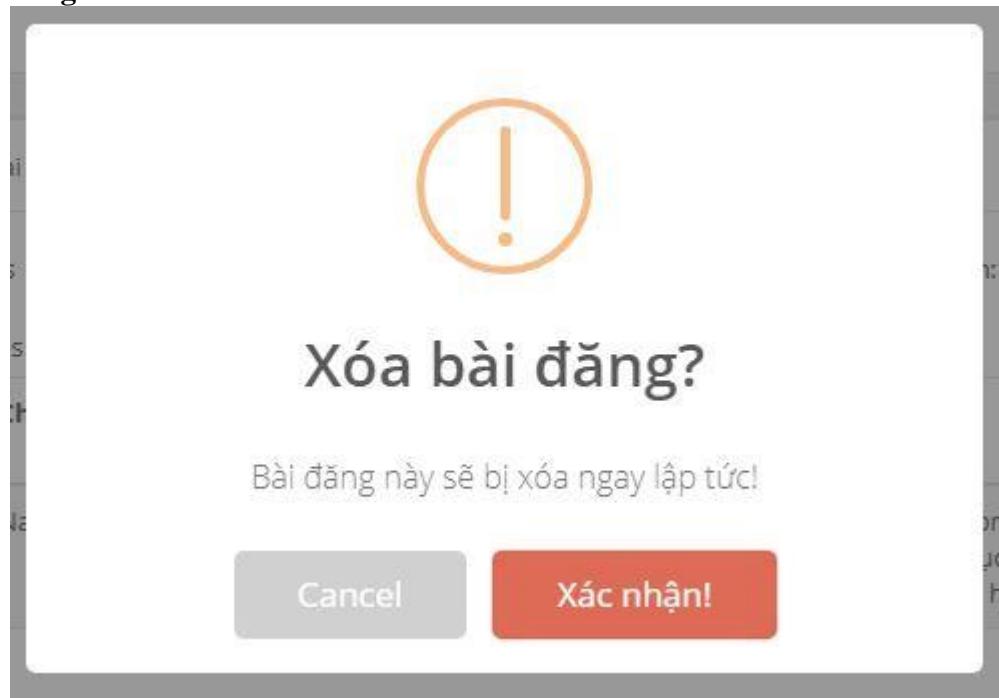


Figure 4-6494: Delete post Screen design

##### Class Diagram

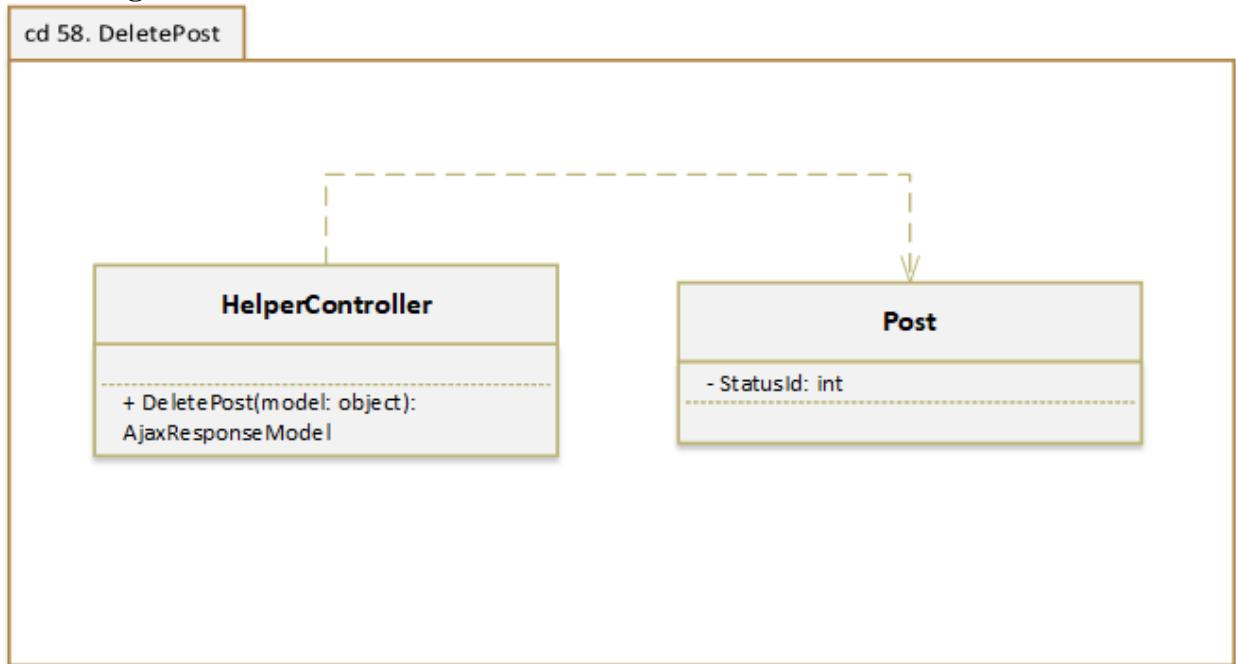


Figure 4-6595: Delete post Class diagram

##### Class Specification

HelperController	
Physical address	MindDoctor / Areas / Admin / Controllers / HelperController.cs
Base class	

Attributes			
No	Name	Type	Description
Operation			
DeletePost			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	model	object	Object contains post id.

Post			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Post.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	StatusId	int	
Operation			

### Sequence Diagram

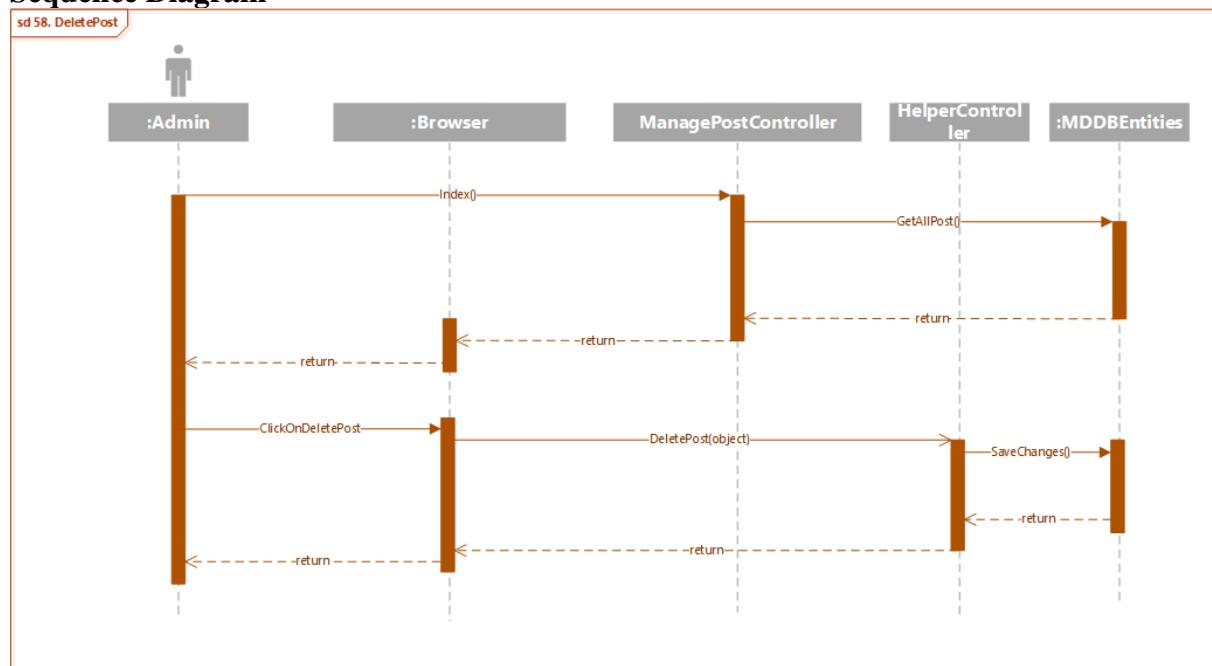


Figure 4-6696: Delete post Sequence diagram

#### 4.3.4.59 View conversation report

##### Screen Design

Figure 4-6797: View conversation report Screen design

##### Class Diagram

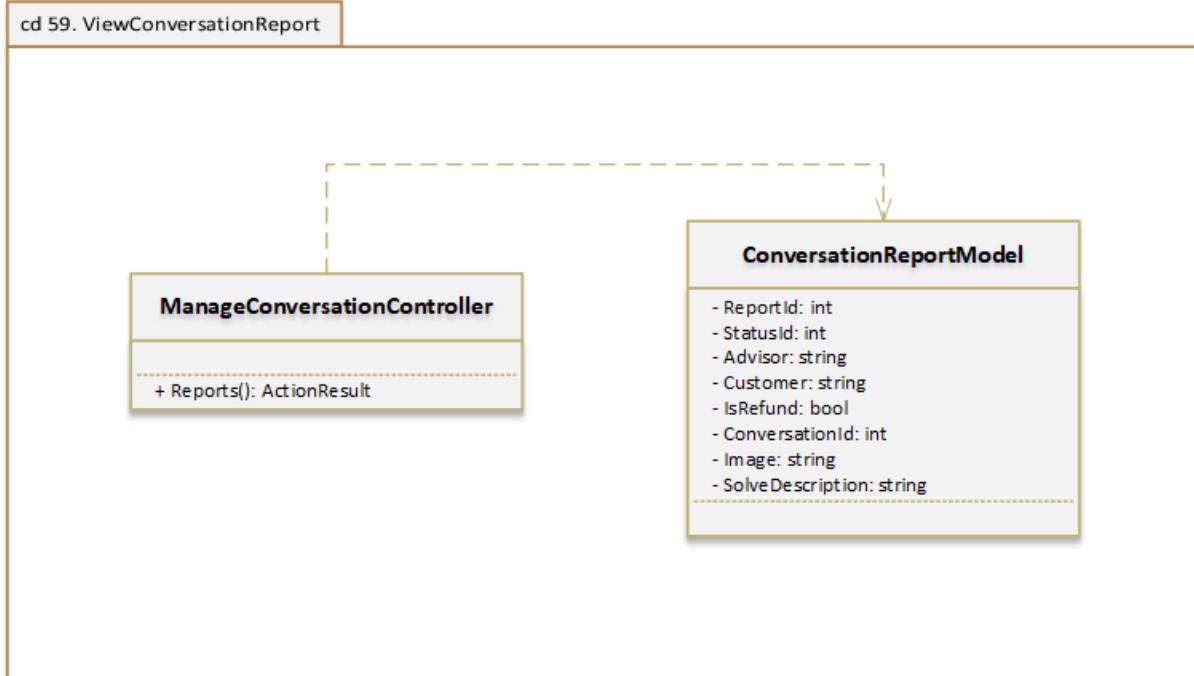


Figure 4-6898: View conversation report Class diagram

##### Class Specification

ManageConversationController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageConversationController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Report			
Return Type	ActionResult		
Parameters	Name	Type	Description

ConversationReportModel	
Physical address	MindDoctor / Models / ConversationModels / ConversationModel.cs

Base class	Attributes		
No	Name	Type	Description
1.	ReportId	int	
2.	StatusId	int	
3.	Advisor	string	
4.	Customer	string	
5.	IsRefund	bool	
6.	ConversationId	int	
7.	Image	string	
8.	SolveDescription	string	

Operation			
-----------	--	--	--

### Sequence Diagram

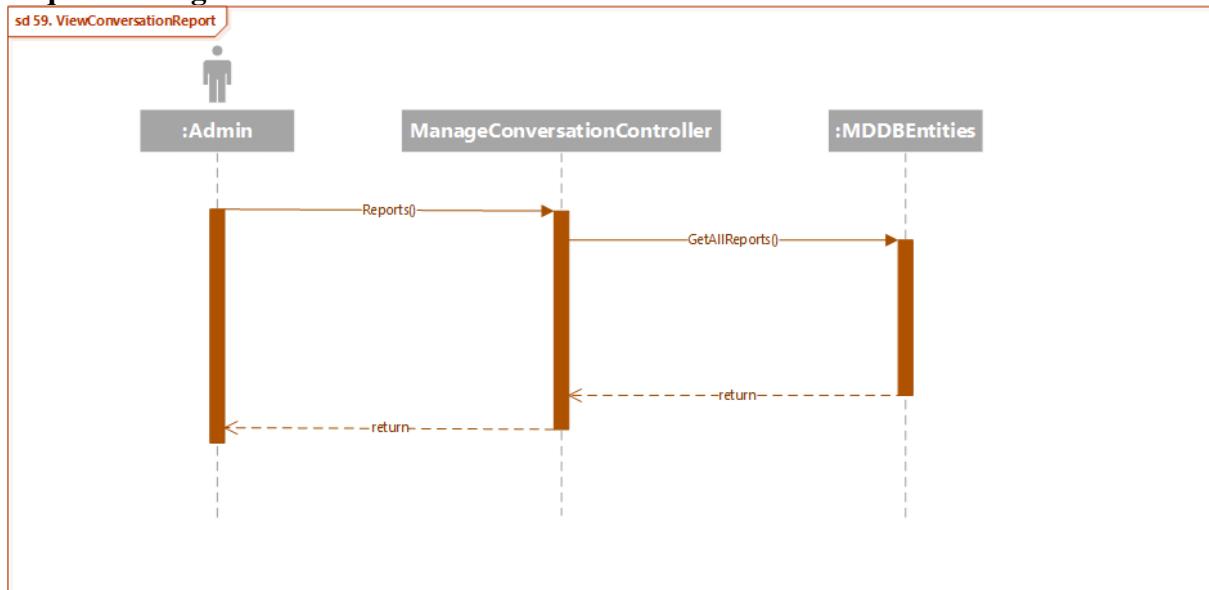


Figure 4-6999: View conversation report Sequence diagram

#### 4.3.4.60 Solve conversation report

##### Screen Design



Figure 4-200: Solve conversation report Screen design

##### Class Diagram

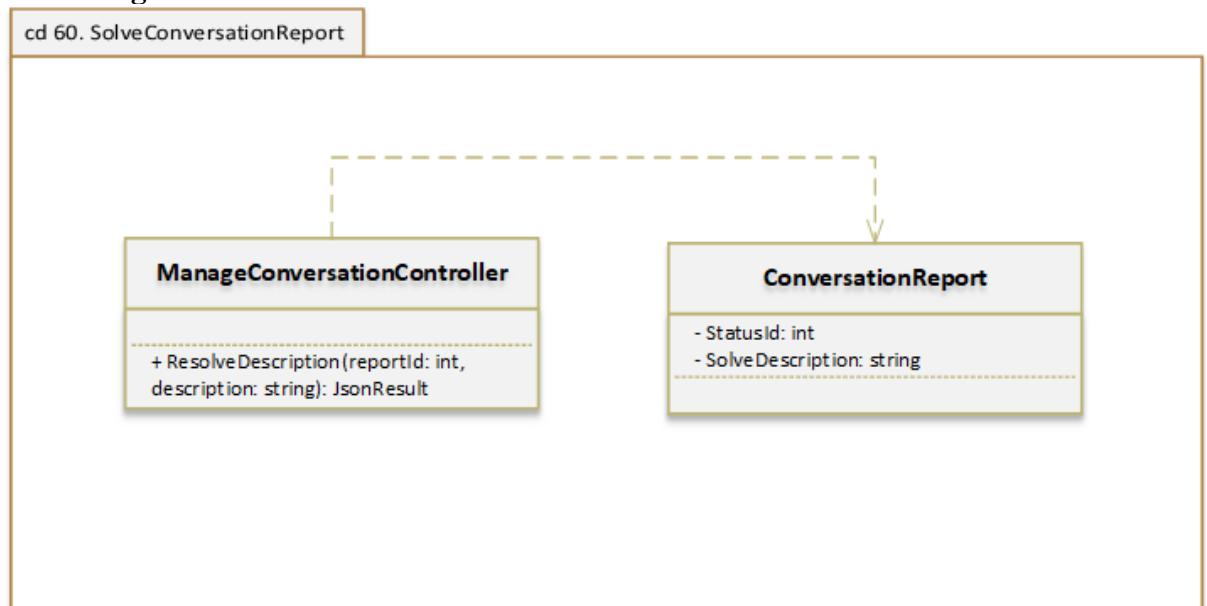


Figure 4-201: Solve conversation report Class diagram

##### Class Specification

ManageConversationController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageConversationController.cs
Base class	
Attributes	

No	Name	Type	Description
Operation			
ResolveConversation			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	reportId	int	Report id represents conversation to be report.
2.	description	string	Resolve description.

ConversationReport			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / ConversationReport.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	StatusId	int	
2.	SolveDescription	string	
Operation			

## Sequence Diagram

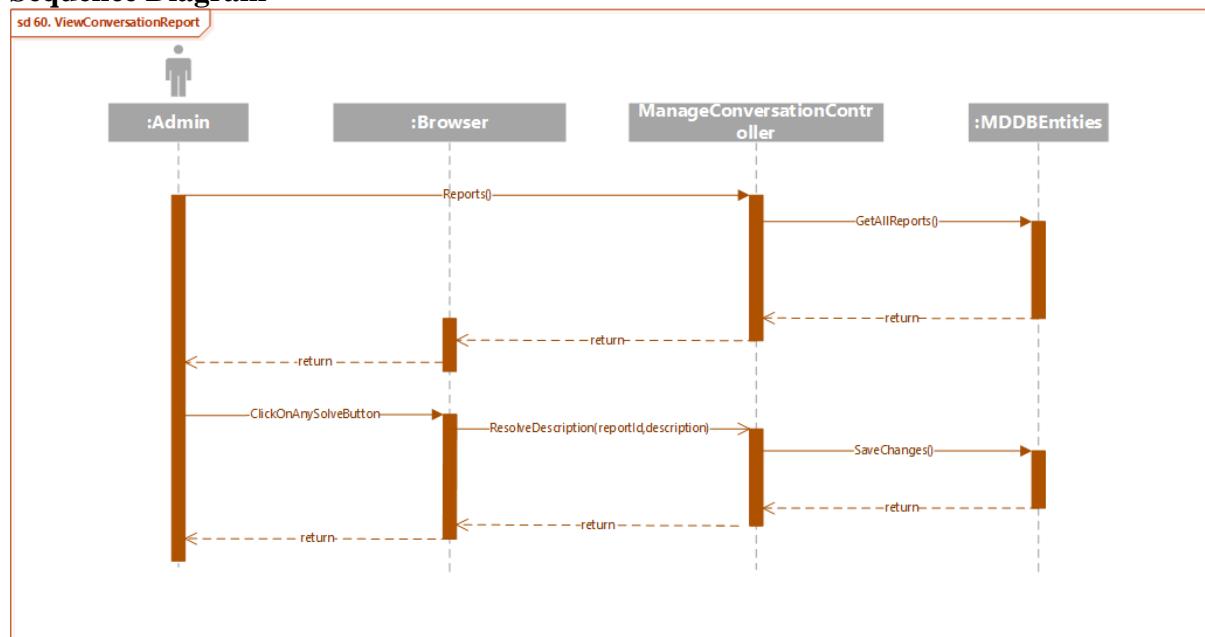


Figure 4-202: Solve conversation report Sequence diagram

#### 4.3.4.61 Refund gold for customer

##### Screen Design



Figure 4-203: Refund gold for customer Screen design

##### Class Diagram

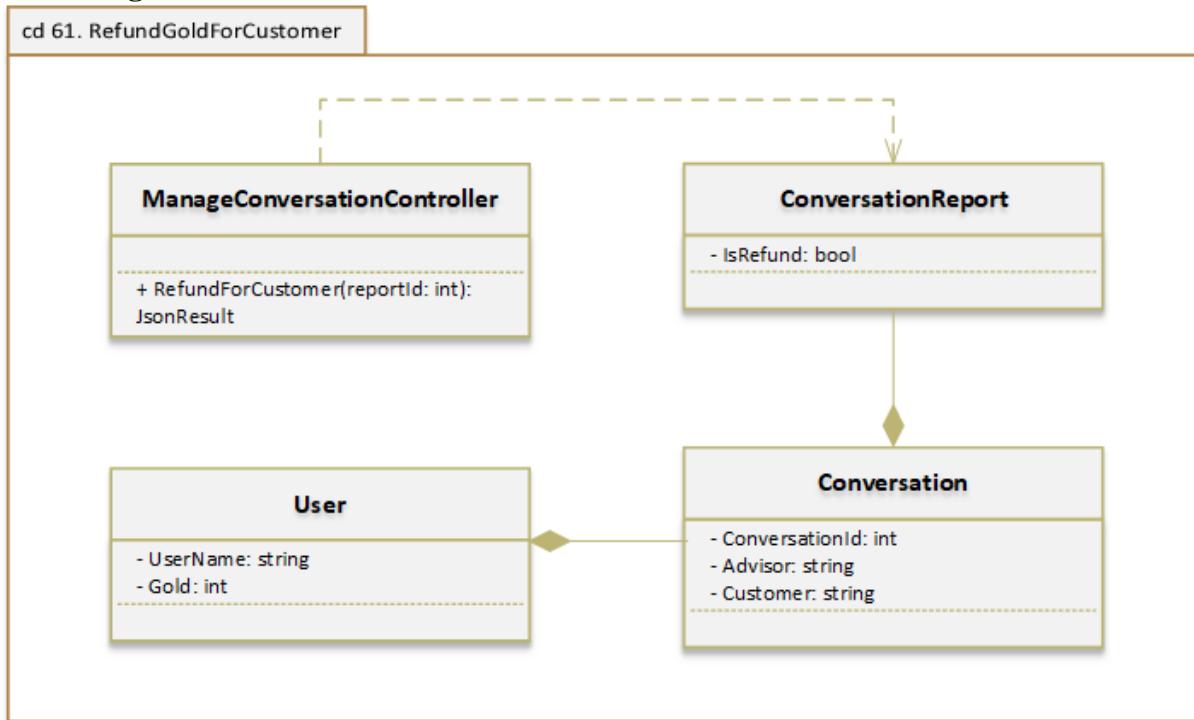


Figure 4-204: Refund gold for customer Class diagram

##### Class Specification

ManageConversationController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageConversationController.cs

Base class			
Attributes			
No	Name	Type	Description
Operation			
RefundForCustomer			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	reportId	int	Report id represents reported conversation.

ConversationReport			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / ConversationReport.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	IsRefund	bool	
Operation			

Conversation			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Conversation.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	int	
2.	Advisor	string	
3.	Customer	string	
Operation			

User			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / User.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	Gold	int	
Operation			

## Sequence Diagram

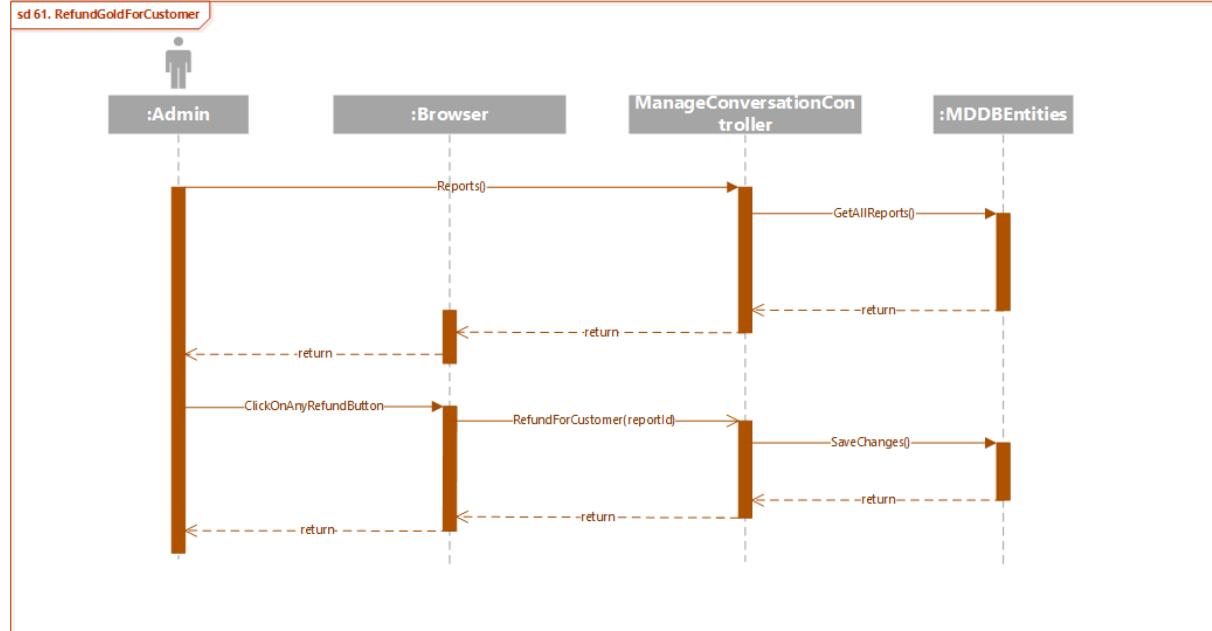


Figure 4-205: Refund gold for customer Sequence diagram

#### 4.3.4.62 View inserted money

##### Screen Design

Bảng dữ liệu danh sách nạp tiền			
Show	10	entries	Search: <input type="text"/>
Showing 1 to 2 of 2 entries			
Hình thức nạp tiền	Khách hàng	Giá trị nạp vào	Quy đổi thành Gold
Paypal	datbv95@gmail.com	227.260	250
Paypal	datbv95@gmail.com	113.630	114

Previous 1 Next

Figure 4-206: View inserted money Screen design

##### Class Diagram

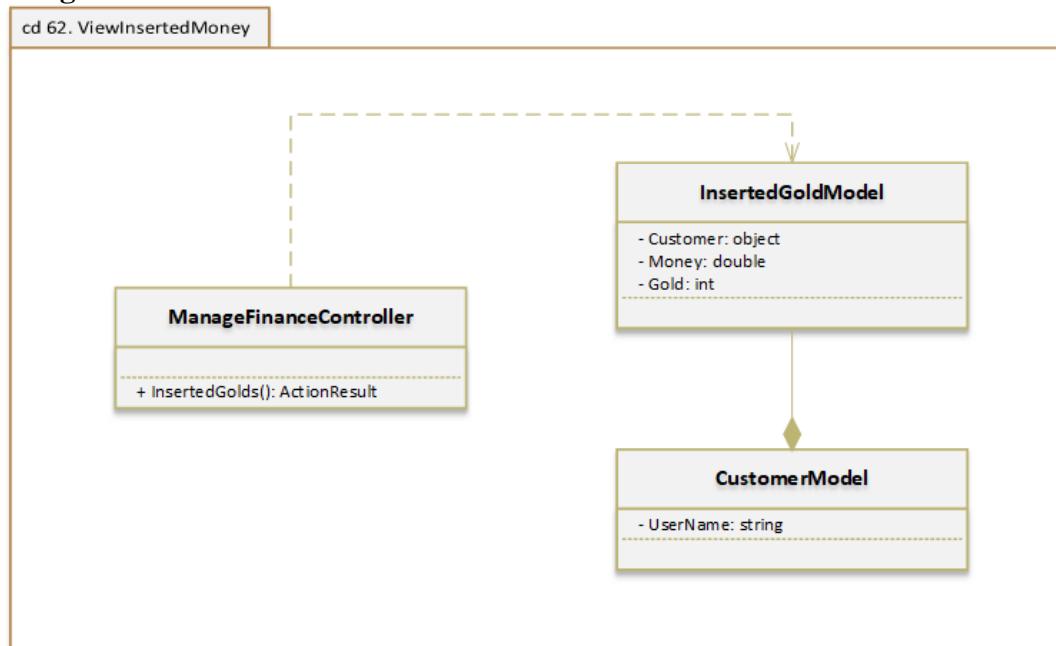


Figure 4-207: View inserted money Class diagram

##### Class Specification

ManageFinanceController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageFinanceController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
InsertedGoldHistory			
Return Type	ActionResult		
Parameters	Name	Type	Description

InsertedGoldModel			
Physical address	MindDoctor / Models / FinanceModels / FinanceModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Customer	object	
2.	Money	double	
3.	Gold	int	
Operation			

CustomerModel			
Physical address	MindDoctor / Models / UserModels / CustomerModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	UserName	string	
Operation			

### Sequence Diagram

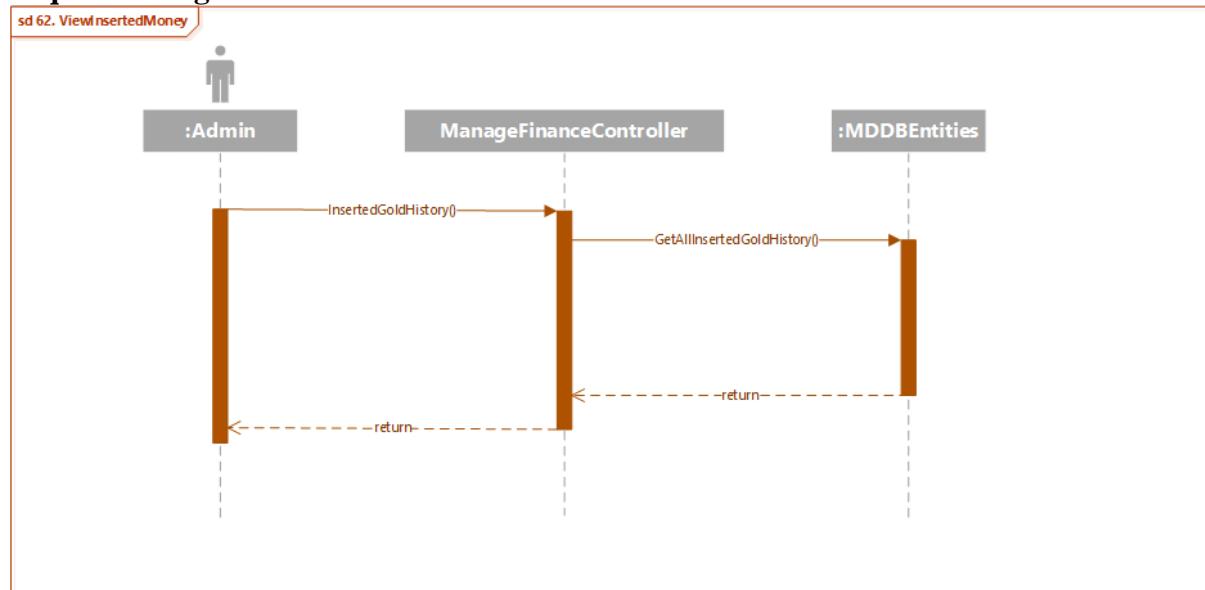


Figure 4-208: View inserted money Sequence diagram

#### 4.3.4.63 View tip money

##### Screen Design

Bảng dữ liệu Danh sách Tip				
Show		10	▼ entries	
Showing 1 to 10 of 11 entries				
Tư vấn viên	Khách hàng	Số Gold	Ngày tạo	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	12	11/07/2017 1:32:16 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	1	11/07/2017 1:33:42 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	1	11/07/2017 2:07:36 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	6	11/07/2017 2:08:02 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	1	11/07/2017 2:09:34 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	1	11/07/2017 2:10:27 CH	
md.adv10@mailnesia.com	md.cus03@mailnesia.com	1	11/07/2017 2:10:41 CH	

Figure 4-209: View tip money Screen design

##### Class Diagram

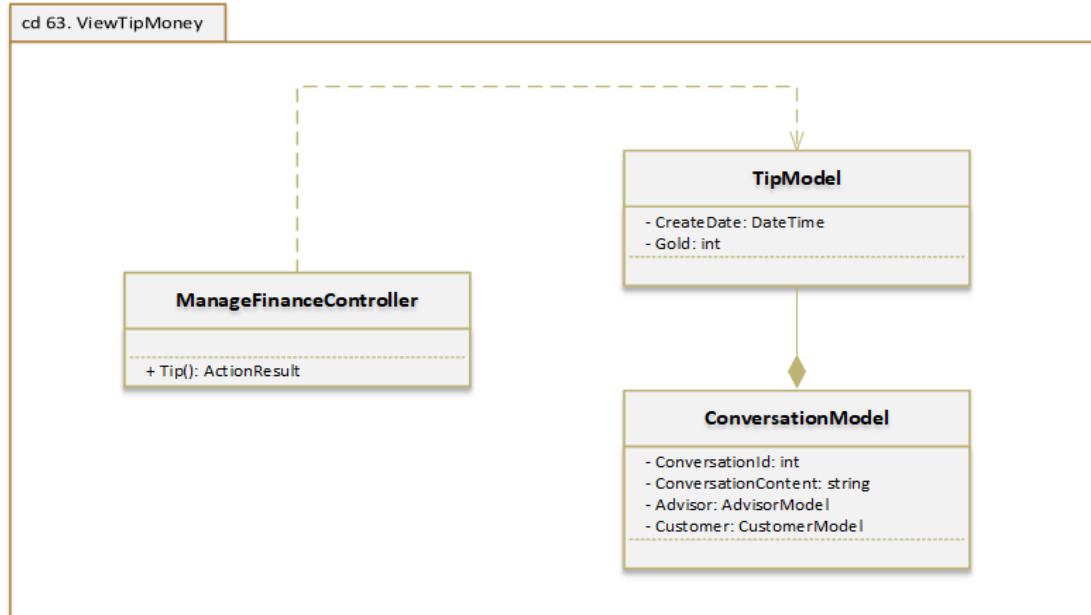


Figure 4-210: View tip money Class diagram

##### Class Specification

ManageFinanceController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageFinanceController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
Tip			
Return Type	JsonResult		
Parameters	Name	Type	Description

TipModel			
Physical address	MindDoctor / Models / FinanceModels / FinanceModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	CreateDate	DateTime	
2.	Gold	int	
Operation			

ConversationModel			
Physical address	MindDoctor / Models / ConversationModels / ConversationModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	ConversationId	int	
2.	ConversationContent	string	
3.	Customer	CustomerModel	
4.	Advisor	AdvisorModel	
Operation			

## Sequence Diagram

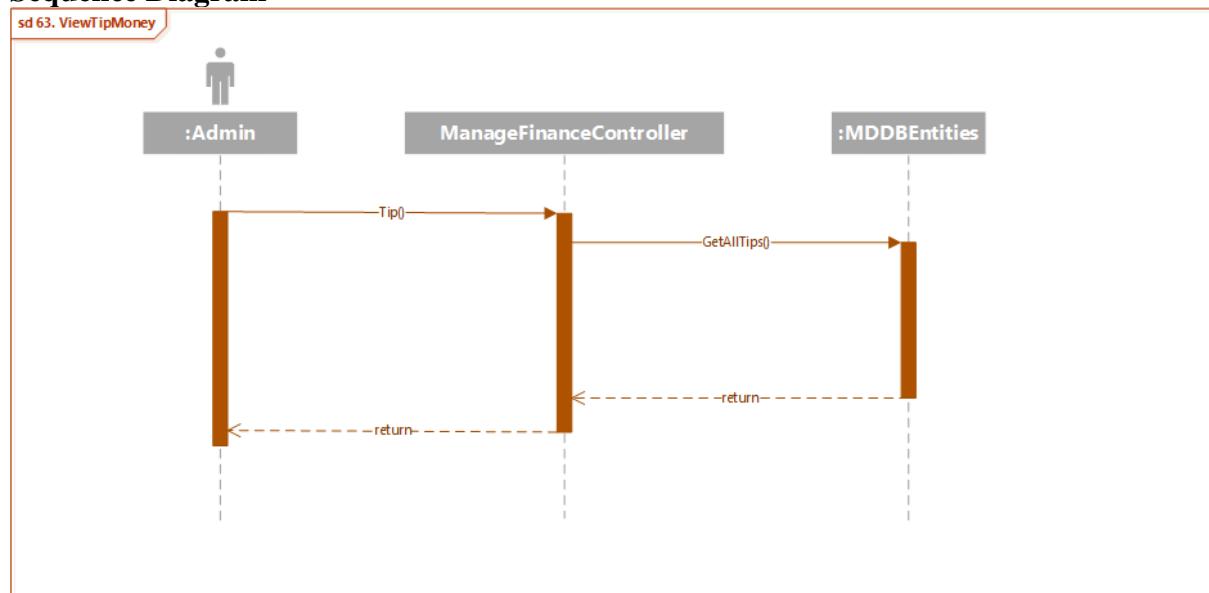


Figure 4-211: View tip money Sequence diagram

#### 4.3.4.64 Create question bank

##### Screen Design

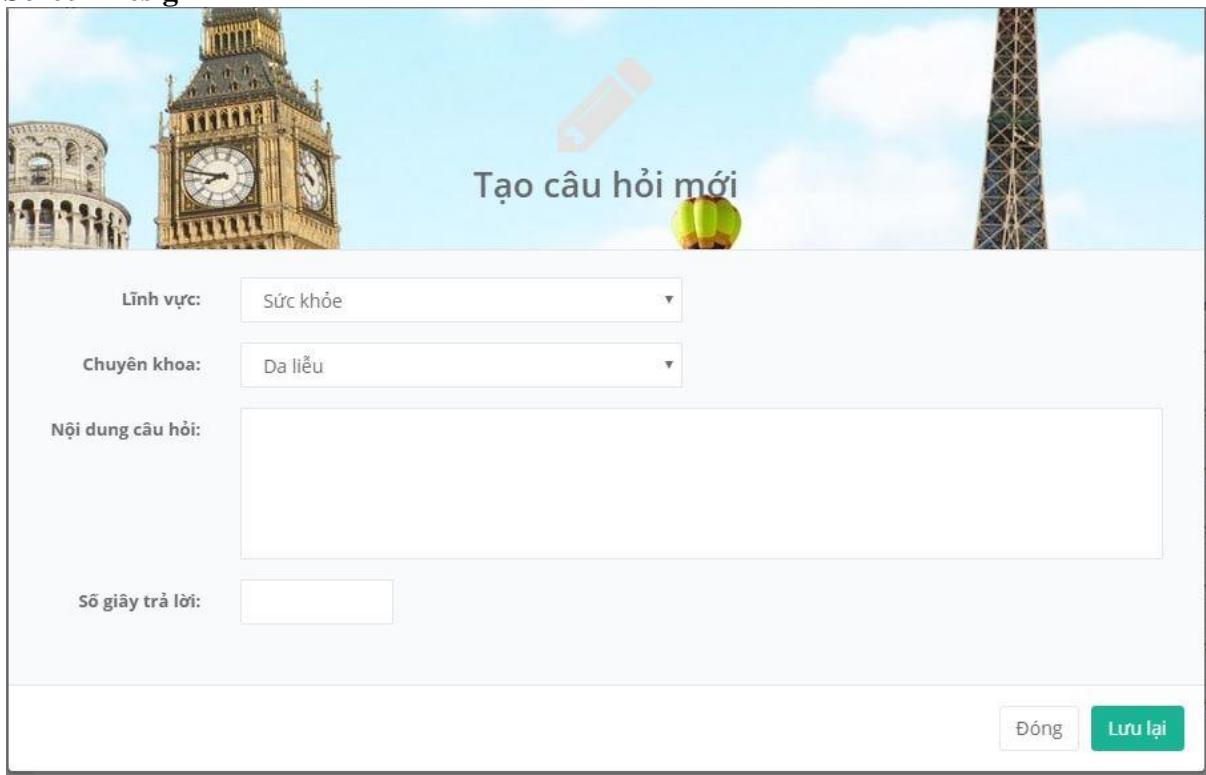


Figure 4-212 Create question bank Screen design

##### Class Diagram

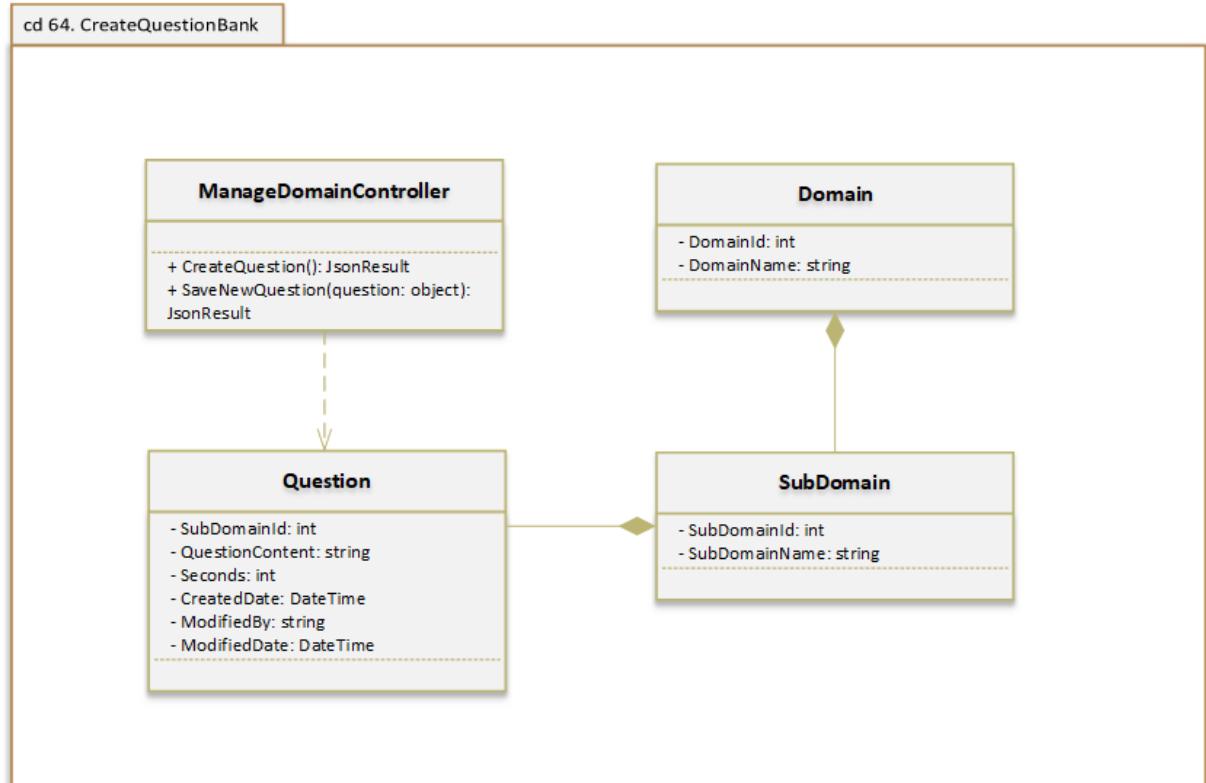


Figure 4-213: Create question bank Class diagram

## Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
Attributes			
No	Name	Type	Description
Operation			
CreateQuestion			
Return Type	JsonResult		
Parameters	Name	Type	Description
SaveNewQuestion			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	question	object	Object contains question id, question content,..

Question			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Question.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	QuestionContent	string	
3.	Seconds	int	
4.	CreatedDate	DateTime	
5.	ModifiedBy	string	
6.	ModifiedDate	DateTime	
Operation			

Domain			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Domain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	DomainId	int	
2.	DomainName	string	
Operation			

## SubDomain

Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / SubDomain.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	SubDomainId	int	
2.	SubDomainName	string	
Operation			

### Sequence Diagram

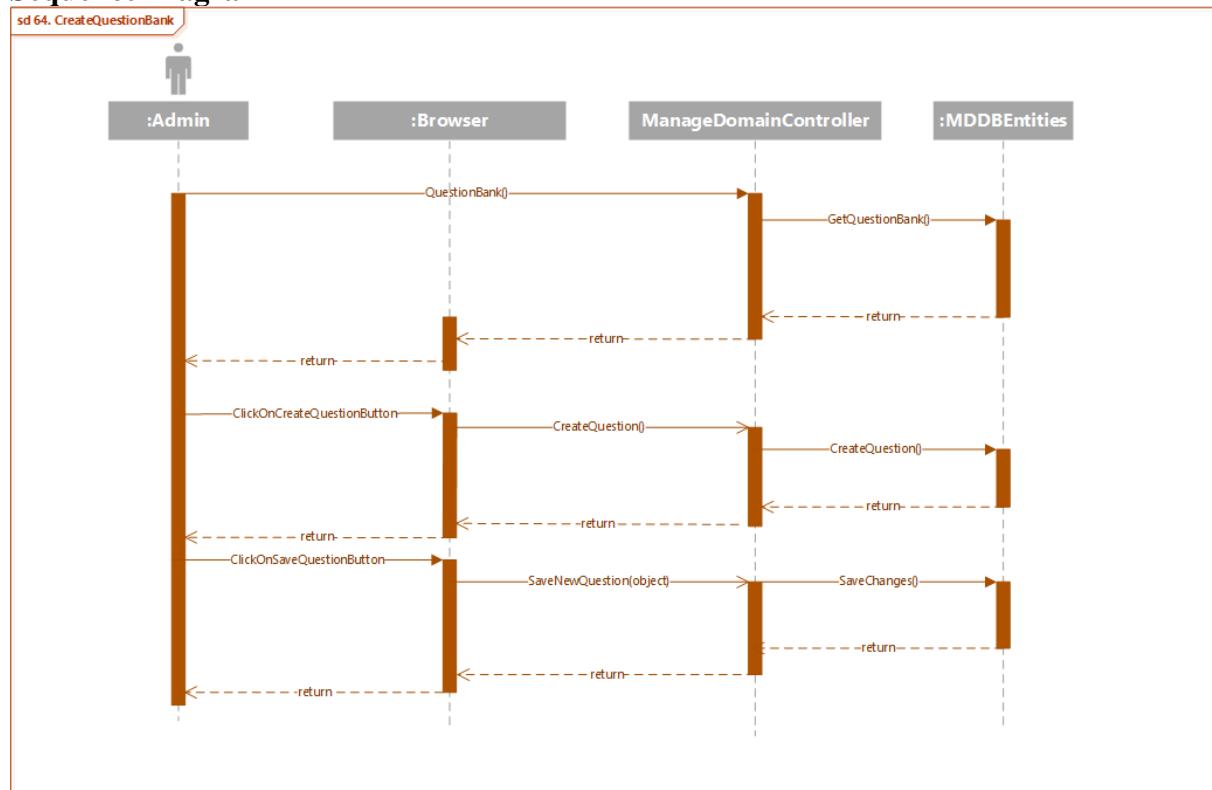


Figure 4-214 Create question bank Sequence diagram

#### 4.3.4.65 Update question bank

##### Screen Design

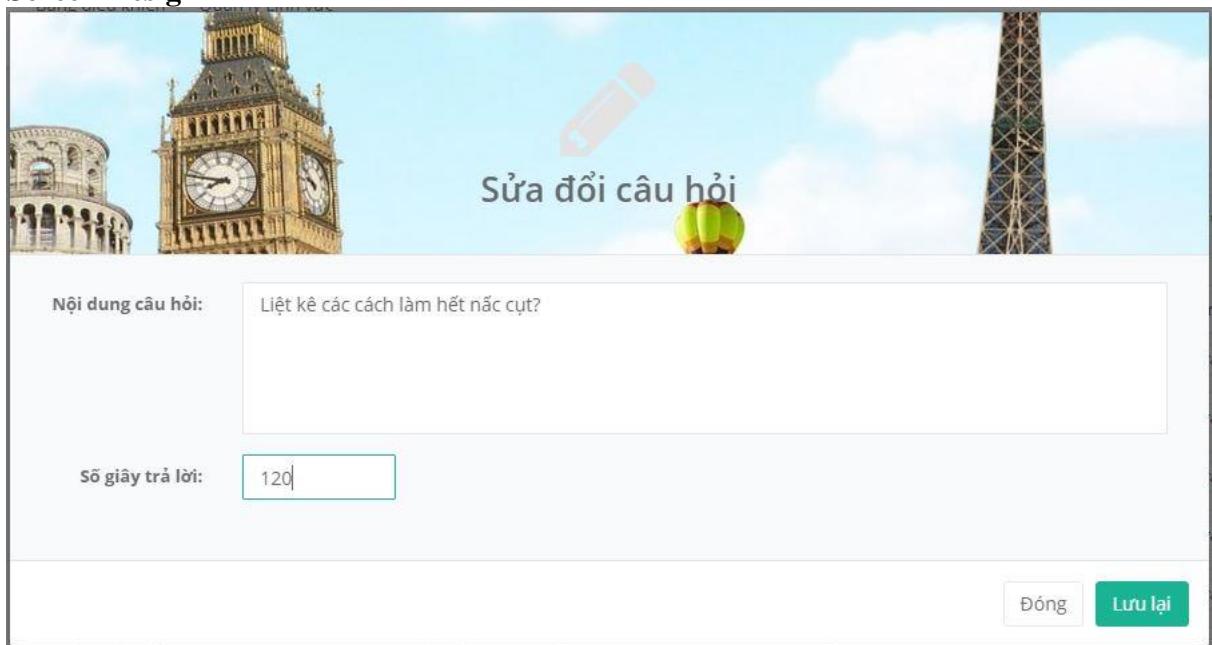


Figure 4-215: Update question bank Screen design

##### Class Diagram

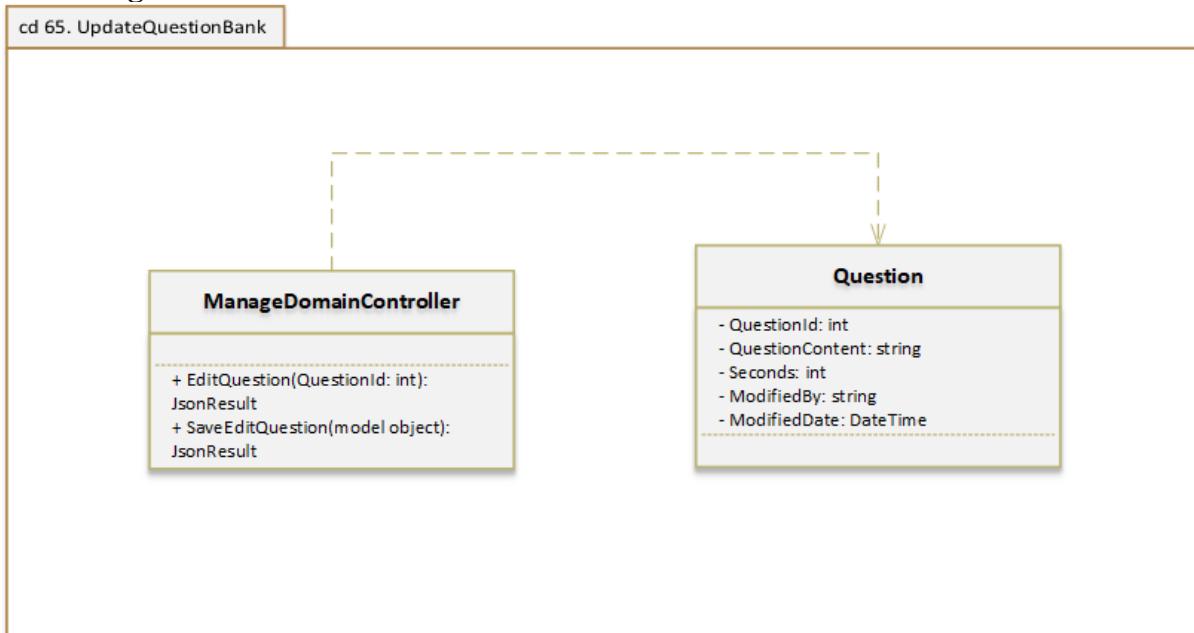


Figure 4-216: Update question bank Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
Attributes			
No	Name	Type	Description

Operation			
EditQuestion			
Return Type	JsonResult		
Parameters	Name	Type	Description
	QuestionId	int	Question id represents question to be update.

Question			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Question.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	QuestionContent	string	
2.	Seconds	int	
3.	ModifiedBy	string	
4.	ModifiedDate	DateTime	
Operation			

### Sequence Diagram

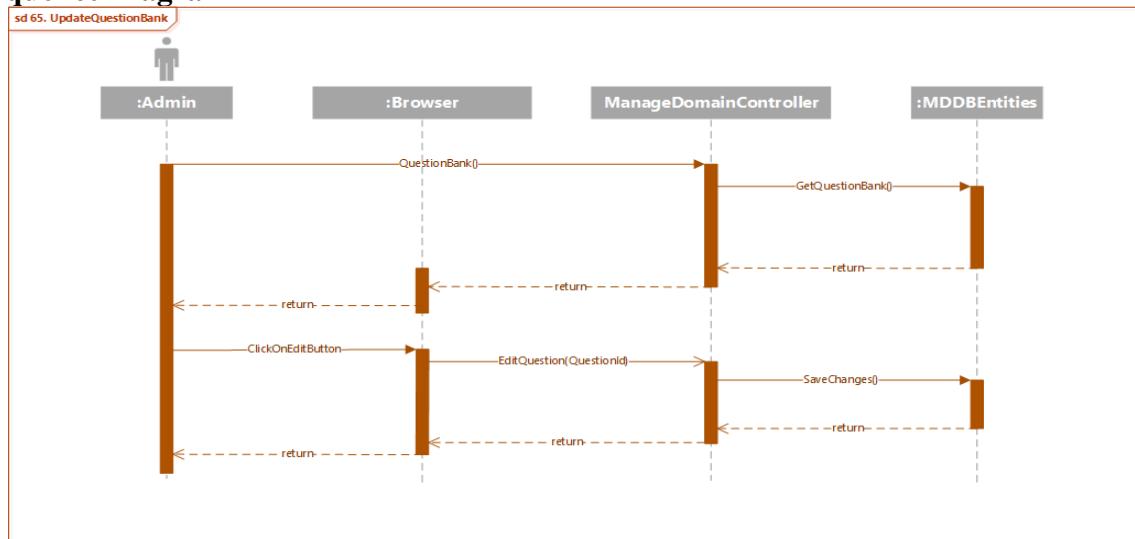


Figure 4-217: Update question bank Sequence diagram

#### 4.3.4.66 Delete question bank

##### Screen Design

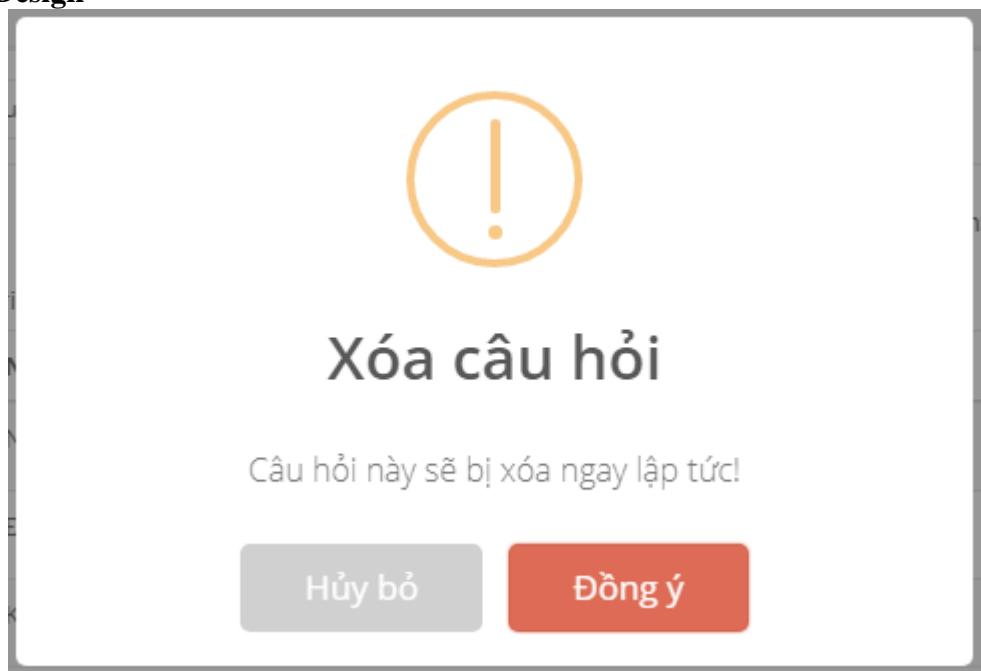


Figure 4-218: Delete question bank Screen design

##### Class Diagram

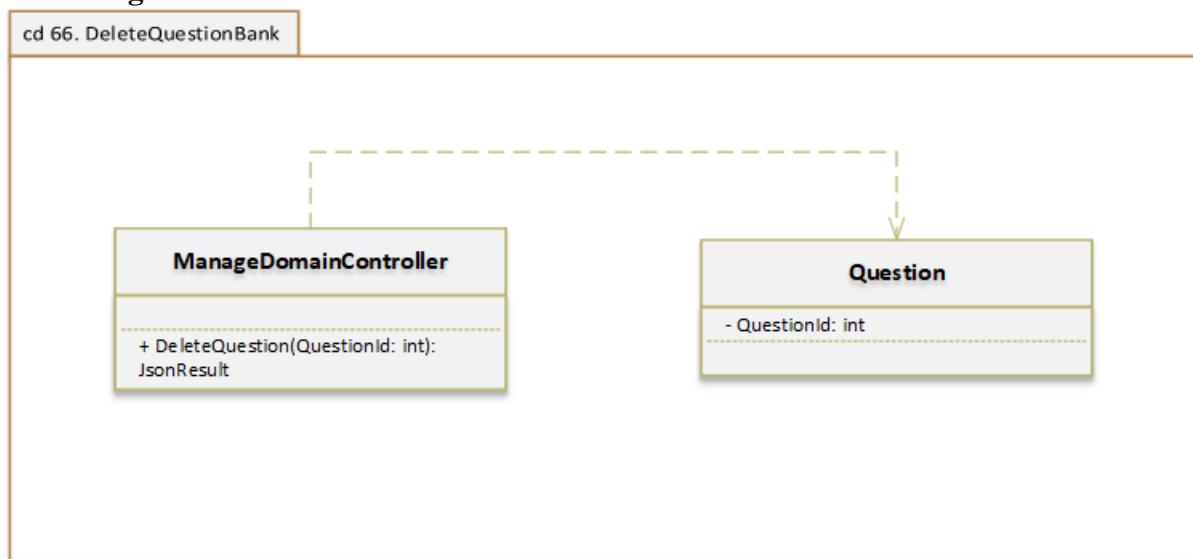


Figure 4-219: Delete question bank Class diagram

##### Class Specification

ManageDomainController			
Physical address	MindDoctor / Areas / Admin / Controllers / ManageDomainController.cs		
Base class			
No	Name	Type	Description
No	Name	Type	Description

DeleteQuestion			
Return Type	JsonResult		
Parameters	Name	Type	Description
	model	object	Object contains question id.

Question			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Question.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	QuestionId	int	
Operation			

### Sequence Diagram

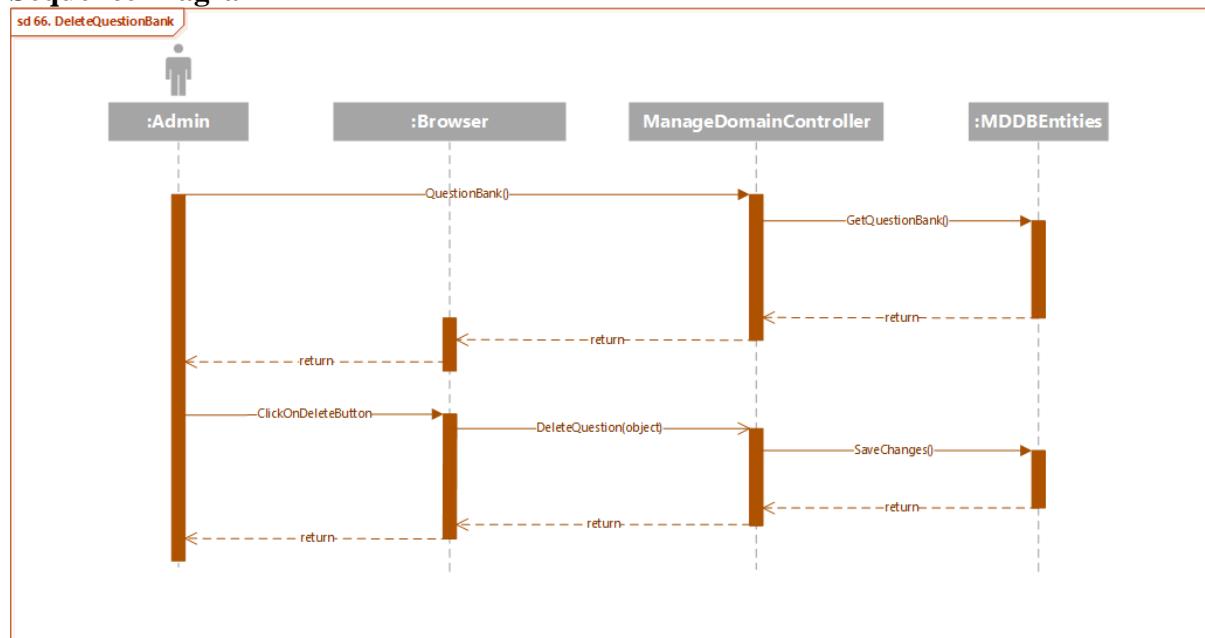


Figure 4-220: Delete question bank Sequence diagram

#### 4.3.4.67 View Advisor's feedback

##### Screen Design



Figure 4-221: View advisor's feedback Screen design

##### Class Diagram

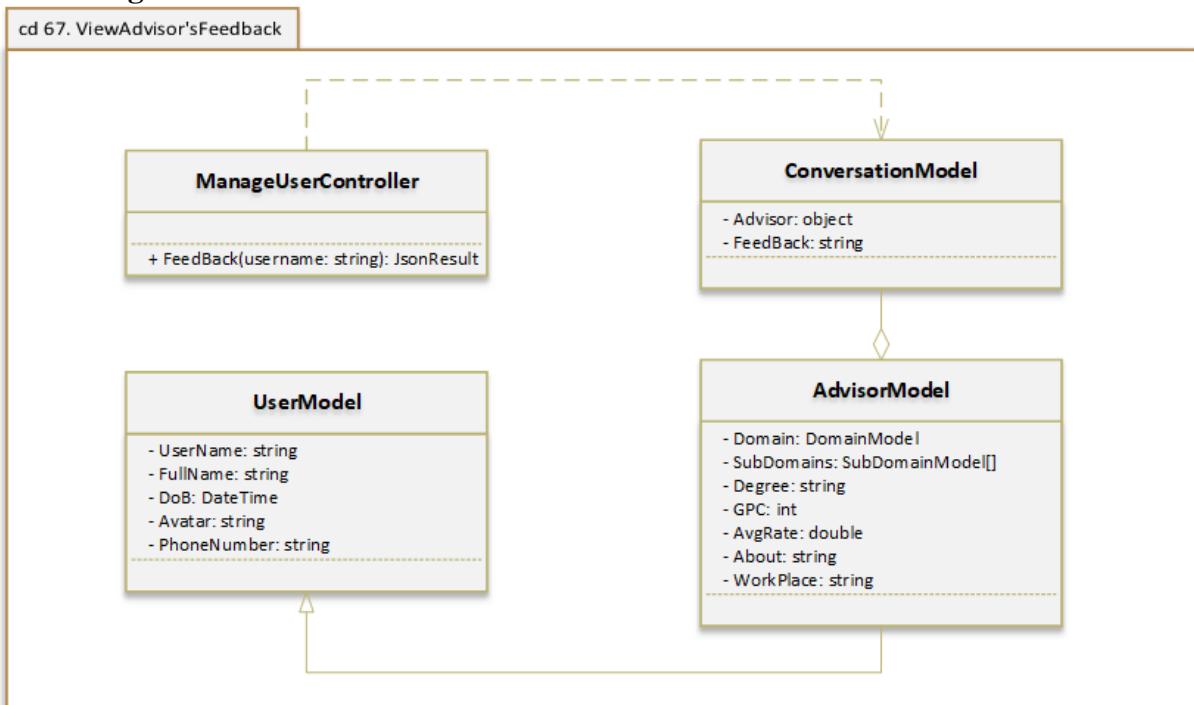


Figure 4-222: View advisor's feedback Class diagram

##### Class Specification

ManageUserController	
Physical address	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs
Base class	

Attributes			
No	Name	Type	Description
Operation			
FeedBack			
Return Type	JsonResult		
Parameters	Name	Type	Description
1.	username	string	Username of advisor.

ConversationModel			
Physical address	MindDoctor / Models / ConversationModels / ConversationModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Advisor	object	
2.	FeedBack	string	
Operation			

AdvisorModel			
Physical address	MindDoctor / Models / UserModels / AdvisorModel.cs		
Base class	UserModel		
Attributes			
No	Name	Type	Description
1.	Domain	DomainModel	
2.	SubDomains	SubDomainModel[]	
3.	Degree	string	
4.	GPC	int	
5.	AvgRate	double	
6.	About	string	
7.	WorkPlace	string	
Operation			

UserModel			
Physical address	MindDoctor / Models / UserModels / UserModel.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	UserName	string	
2.	FullName	string	
3.	DoB	DateTime	

4.	Avatar	string	
5.	PhoneNumber	string	
Operation			

## Sequence Diagram

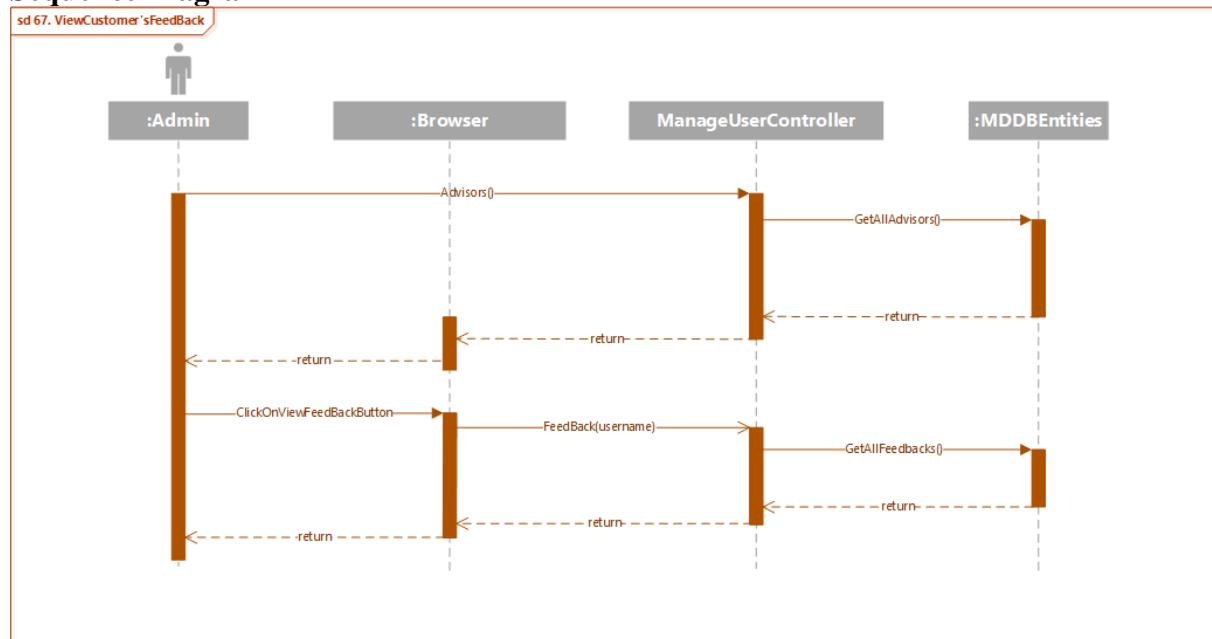


Figure 4-223: View advisor's feedback Sequence diagram

#### 4.3.4.68 Pay money for advisor (assum via Bank)

##### Screen Design

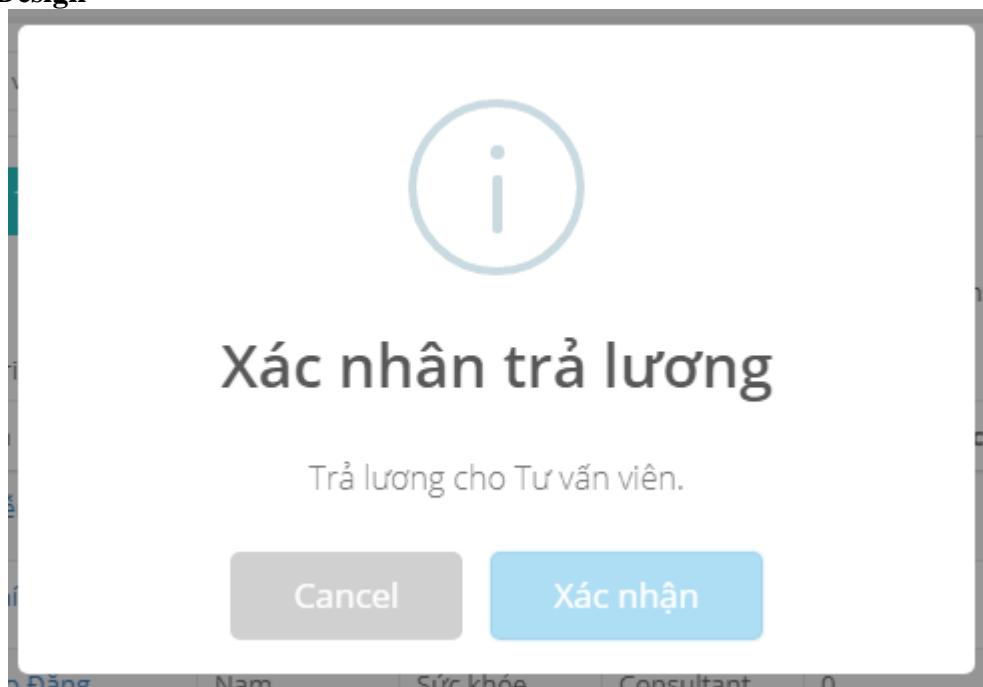


Figure 4-224: Pay money for advisor Screen design

##### Class Diagram

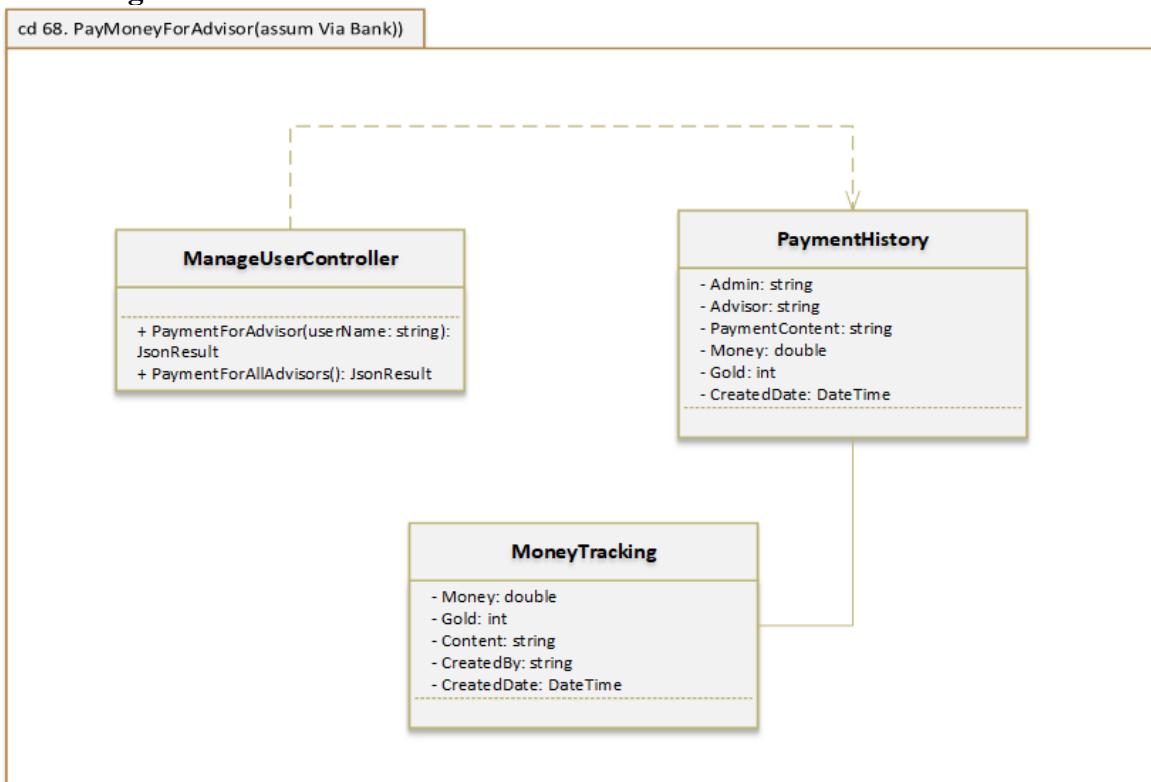


Figure 4-225: Pay money for advisor Class diagram

##### Class Specification

ManageUserController	
Physical	MindDoctor / Areas / Admin / Controllers / ManageUserController.cs

address			
Base class			
Attributes			
No	Name	Type	Description
Operation			
PaymentForAdvisor			
Return Type	JsonResult		
Parameters	Name	Type	Description
	userName	string	Username of advisor.

PaymentHistory			
Physical address	MindDoctor / EntityFramework / PaymentHistory.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Admin	string	
2.	Advisor	string	
3.	PaymentContent	string	
4.	Money	double	
5.	Gold	int	
6.	CreatedDate	DateTime	
Operation			

MoneyTracking			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / MoneyTracking.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	Money	double	
2.	Gold	int	
3.	Content	string	
4.	CreatedBy	string	
5.	CreatedDate	DateTime	
Operation			

## Sequence Diagram

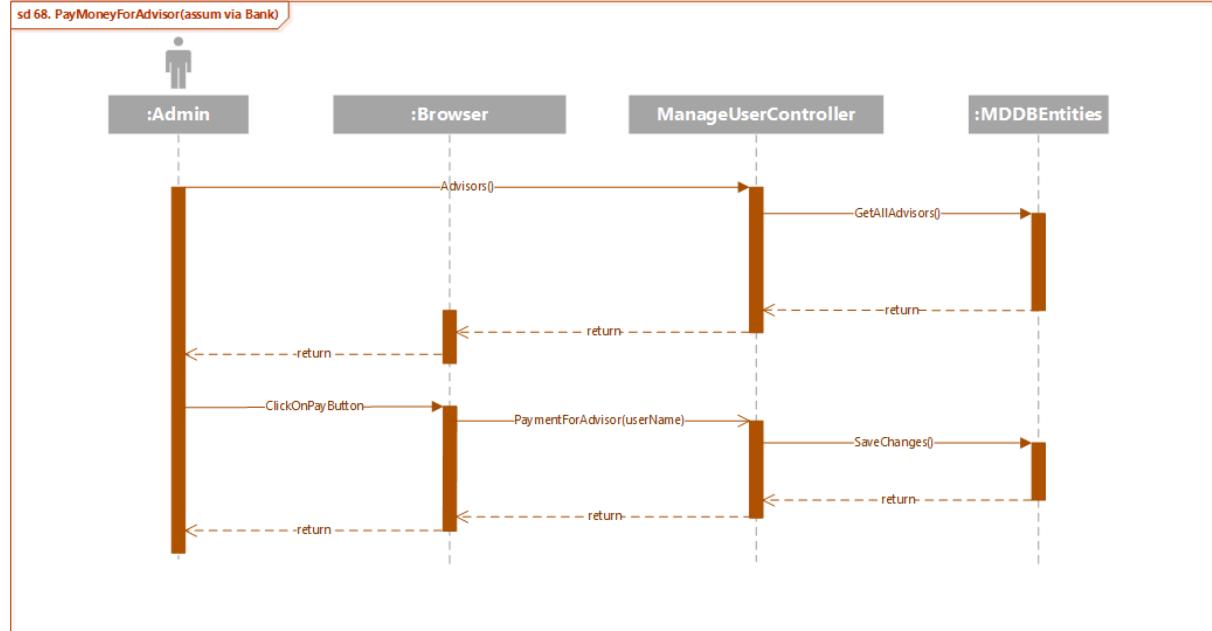


Figure 4-226: Pay money for advisor Sequence diagram

#### 4.3.4.69 Appointment helper

##### Screen Design

Microsoft Software Shadow Copy Provider	Manages software-based volum...	Running	Manual	Local Syste...
Microsoft Storage Spaces SMP	Host service for the Microsoft St...	Manual	Network S...	
MindDoctorAppointmentHandleService	This is help MindDoctor system t...	Running	Manual	Local Syste...
Net.Tcp Port Sharing Service	Provides ability to share TCP por...	Disabled	Local Service	
Netlogon	Maintains a secure channel betw...	Manual	Local Syste...	
Network Connection Broker	Brokers connections that allow ...	Running	Manual (Trig...	Local Syste...

Figure 4-227: Appointment helper Screen design

##### Class Diagram



Figure 4-228: Appointment helper Class diagram

##### Class Specification

Appointment			
Physical address	MindDoctor / EntityFramework / Model.edmx / Model.tt / Appointment.cs		
Base class			
Attributes			
No	Name	Type	Description
1.	AppointmentId	int	
2.	StatusId	int	
3.	ScheduleId	int	
4.	TimeId	int	
5.	DepositGold	int	
Operation			

## Sequence Diagram

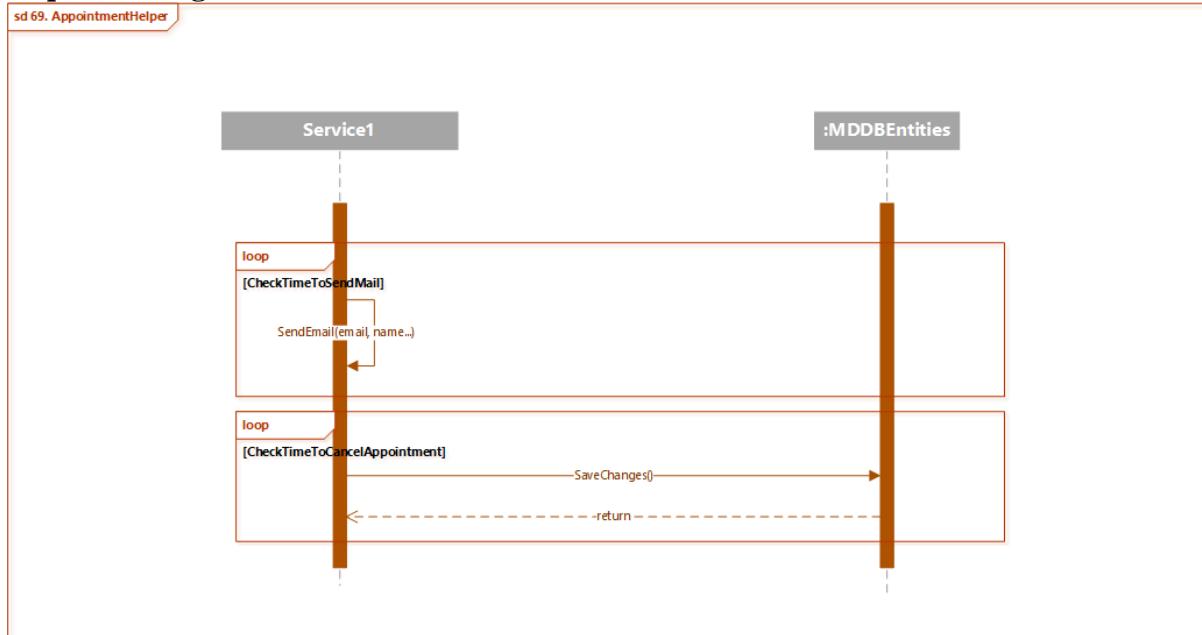


Figure 4-229: Appointment helper Sequence diagram

## Implementation

Microsoft Windows services enable to create long-running executable applications that run in their own Windows sessions. The services can be automatically started when the computers boots, can paused and restarted, and do not show any user interface. The features make services ideal for use on a server or whenever you need long-running functionality.

Therefore, MindDoctor built a Windows service to manage and handle appointment between Customer and Advisor.

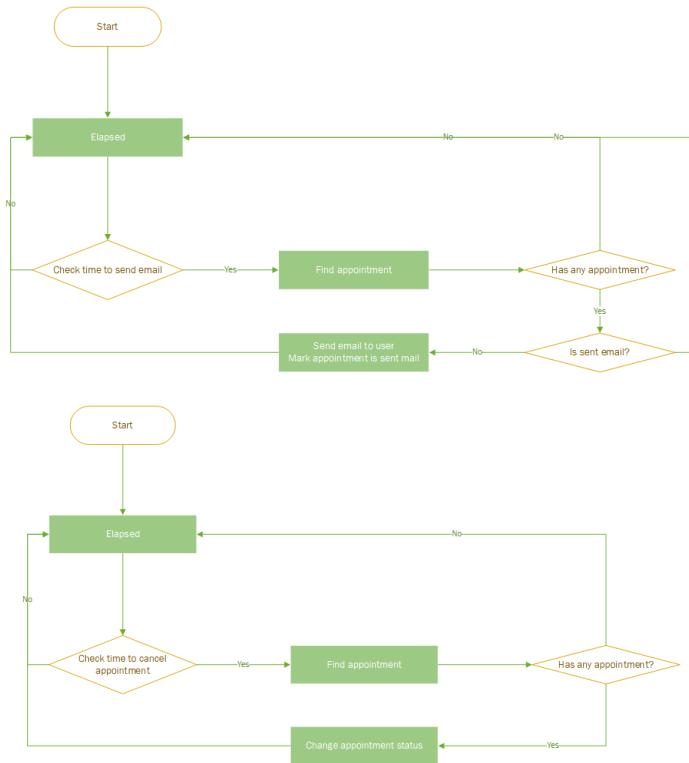
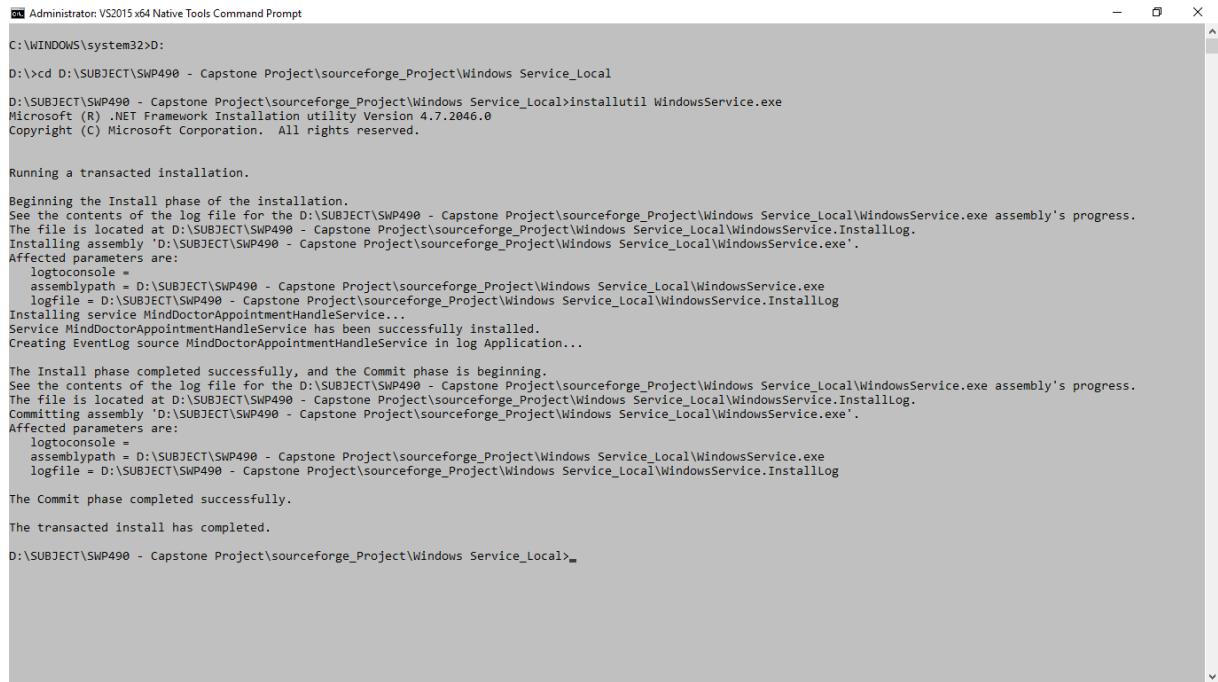


Figure 4-230: Flow chart of send email and mark appointment

After release Windows service, we need to use VS15 Native Tools Command Prompt and run as Administrator to install.

Install code:           installutil WindowsService.exe

Uninstall code:        installutil /u WindowsService.exe



The screenshot shows a Windows Command Prompt window titled "Administrator: VS2015 x64 Native Tools Command Prompt". The command entered is "installutil WindowsService.exe". The output log shows the installation process:

```
C:\>cd D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local  
D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local>installutil WindowsService.exe  
Microsoft (R) .NET Framework Installation utility Version 4.7.2046.0  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Running a transacted installation.  
Beginning the Install phase of the installation.  
See the contents of the log file for the D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog.  
The file is located at D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog.  
Installing assembly 'D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.exe'.  
Affected parameters:  
  logtoconsole =  
    assemblypath = D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.exe  
    logfile = D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog  
Installing service MindDoctorAppointmentHandleService...  
Service MindDoctorAppointmentHandleService has been successfully installed.  
Creating Eventlog source MindDoctorAppointmentHandleService in log Application...  
  
The Install phase completed successfully, and the Commit phase is beginning.  
See the contents of the log file for the D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog.  
The file is located at D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog.  
Committing assembly 'D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.exe'.  
Affected parameters:  
  logtoconsole =  
    assemblypath = D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.exe  
    logfile = D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local\WindowsService.InstallLog  
  
The Commit phase completed successfully.  
The transacted install has completed.  
D:\SUBJECT\SWP490 - Capstone Project\sourceforge_Project\Windows Service_Local>=
```

Figure 4-231: Install Windows service

# Chapter 5 : Software Testing Documentation

## 5.1 Introduction

### 5.1.1 Purpose

The primary purpose of this chapter is to detect and prevent defects, which may be created by developers while developing the software and this may lead to software failures. On the other hand, another objective of this chapter is to provide information about the level of quality and to make sure that the end result meets the business and user requirements. It contains the following sections:

- Scope of Testing.
- Testing Tool and Environment.
- Resources and responsibilities.
- Test strategy: Test approach, test stages.
- Test schedule.
- Feature to be tested.
- Feature not to be tested.
- Defect Log.
- Test report.

### 5.1.2 Scope of testing

#### ➤ Stages of testing:

There are 3 phases in Testing Process: Unit testing, Integration testing and System testing.

ID	Test Stages	Description
1	<b>Unit testing</b>	The main aim of this endeavor is to determine whether the application functions as designed. In this phase, a unit can refer to a function, individual program or even a procedure, and a White-box Testing method is usually used to get the job done.
2	<b>Integration testing</b>	Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/ functions. This is particularly beneficial because it determines how efficiently the units are running together.
3	<b>System testing</b>	System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards.

➤ **Type of testing**

The test team has to test the following type on Google Chrome

- GUI test
- Function test
- Performance test
- Regression test
- Acceptance test
- Unit test
- Load test

➤ **Range of testing**

Team performs all functions defined in the SRS based on the approved version.

## 5.2 Test plan

### 5.2.1 Testing tools and environment

#### 5.2.1.1 Testing tools

##### 5.2.1.1.1 *MindDoctor Front-end and Project testing*

- **Chrome DevTools:** To view logs, inspect elements.



Figure 5-1: Chrome Developer Tools 60.0

- **Microsoft Excel:** To manage test cases.



Figure 5-2: Microsoft Excel

- **Trello:** To manage bugs: Show all bugs and life cycle of process to resolve bug.



Figure 5-3: Trello.com

- **Nunit:** To test individual implemented methods, functions of class.



*Figure 5-4: Nunit 3.7.1*

- **JMeter:** To load test functional behavior and mean sure performance.



*Figure 5-5: JMeter 2.13*

### 5.2.1.2 Testing environment

Type of testing	Software	Hardware
System test	- Chrome Version 60.0.3112.101	Personal computer for developing with the minimum configuration: <ul style="list-style-type: none"> <li>- Windows 10 Education 64-bit</li> <li>- Intel® Core™ i7 6500U</li> <li>- Installed memory (RAM): 8.00GB</li> </ul>
Unit test	Visual Studio 2015 .NET Framework 4.5	Personal computer for run testing with minimum configuration: <ul style="list-style-type: none"> <li>- Windows 10 Education 64-bit</li> <li>- Intel® Core™ i7 6500U</li> <li>- Installed memory (RAM): 8.00GB</li> </ul>

### 5.2.2 Resources and responsibilities

ID	Resources	Responsibilities
1	Project Manager	<ul style="list-style-type: none"> <li>• Review Test case and report.</li> </ul>
2	Tester	<ul style="list-style-type: none"> <li>• Performing the actual system testing.</li> <li>• Manage test resource and assign test tasks.</li> <li>• Create Test Plan.</li> <li>• Create Test Cases.</li> <li>• Create Test Report.</li> <li>• Execute Test.</li> <li>• Test Log report.</li> </ul>

3	Developer	<ul style="list-style-type: none"> <li>• Create unit test and integration test scripts.</li> <li>• Fix bugs.</li> </ul>
---	-----------	---

### 5.2.3 Test strategy

#### 5.2.3.1 Test model

Overall, MindDoctor project uses the Iterative and Incremental Software Process Model. In each phase of process, we use specific process for each sub-system team to fit the requirement, the characteristic and the human resource of the team.

MindDoctor applies Test-Driven Development (TDD) and Behavior Driven Development (BDD) process, which covers source code by Unit. In the development time, whenever we add a new feature or change the old features, we will add/modify the tests first, and then write code to make the test pass then refactor the code and refactor the test at the last.

Instead of waiting for project is completed and tested. Tester parcticipate software development process with the role of analyzing and constructing a test script. At the same time, tester help developer in explaining and offerring planing for practical coding.

MindDoctor GUI testing works mostly with UX, responsive in mobile web. So that MindDoctor also excute test case GUI as system test.

#### 5.2.3.2 Test types

Testing MindDoctor Project will be carried out in each release package as defined in project plan and will depend on internal delivery by development side. Critical tests will be defined as any new or modified tests for the MindDoctor system. Project Manager will decide which closed defects the test team cannot determine.

Each type of MindDoctor Project Testing will be designated to cover a level of project, from each model level, project level with Unit testing combined with Database testing, System testing. On the other hand, MindDoctor will use several of testing type to prevent defects from code modifications and latent bugs.

The different types of testing that will be carried out this project are:

- Unit test:
  - Testing individual implemented methods, functions of model class or library class.
  - Unit test also includes database testing to verify constraint, transaction, default value, data types, data format, and check null and junk characters which are mentioned in database design and software requirement.
  - Test case will have to cover all logic branch that function or method could execute with difference data input.
  - Implemented function's error message and database error message will be included in this test.
- UI testing

- User Interface testing verifies a user's interaction with the software. The goal of GUI testing is to ensure that the GUI provides the user with an appropriate access and navigation through the functions of the target-of-test. In addition, GUI testing ensures that the objects within the GUI function as expected and conform to requirement.
- GUI test will be performed fully on all screens.
- This test targets to cover the verification of the overall look and feel of the MindDoctor system including initial position, font, text size, color, focus, initial button, tab order, label, screen sizes and sentences width.
- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, it must be able to provide inputs to the input fields.
  - Check if Error Messages are displayed correctly.
  - Check if Font used in application is readable.
  - Check if the alignment of the text is proper.
  - Check if images are set first in question block.
  - Check the positioning of GUI elements for different screen resolution.
- Performance testing
  - The testing is to confirm to cover for response time for MindDoctor System with different network conditions with the throttling tool of Google Chrome.
- Regression testing
  - The testing is to confirm that the bug was removed including the extent of the impact, when developers fix bug, developers and testers will confirm with each other what is the impacts of fix bug modification, after that all impact unit or function will be retested by developers then testers after that.

### 5.2.3.3 Test stage

Table below are the stages in which common tests are executed:

Type of test	Stage of test		
	Unit test	Integration test	System test
Unit Tests	x	x	x
UI Tests			x
Regression Tests	x		x
Performance Tests			x
Load Tests			x

Table 5-1: Test stage

#### 5.2.3.4 Test schedule

Table below is the Test Schedule for MindDoctor Project:

Test Schedule	Start Date	End Date
<i>Phase 1: Authentication and features</i>	17/05/2017	11/07/2017
Unit Testing	19/05/2017	05/07/2017
User Interface Testing	30/05/2017	06/07/2017
System Testing	12/06/2017	03/07/2017
Performance Tests	03/07/2017	04/07/2017
Regression Tests	03/07/2017	04/07/2017
<i>Phase 2: Related features and optimization</i>	11/07/2017	08/08/2017
Unit Testing	11/07/2017	02/08/2017
User Interface Testing	30/07/2017	03/08/2017
System Testing	26/07/2017	20/08/2017
Performance Tests	20/08/2017	21/08/2017
Regression Tests	20/08/2017	21/08/2017

Table 5-2: Test schedule

#### 5.2.3.5 Deliverables

Table below is the Deliverables for MindDoctor Project:

Deliverables	Responsibilities	Complete date
Test Plan	Tester	18/05/2017
Test Cases	Tester	13/07/2017
Test case review	Tester + PM	14/07/2017
Test Data	Tester	15/07/2017
Defect report	All members	21/08/2017
Final test Summary report	PM	22/08/2017

#### 5.2.4 Features to be tested

Actor	Name
Guest	Register to become Customer
	Verify email address
	Register to become Advisor
	Take qualification exam
	View post
	Share post on Facebook
	Search post
	View Advisor's schedule
	View online Advisors
	View online Advisors by searching Advisor's information
Customer	View Advisor's information
	Login with local account

	Login with Facebook account
	Logout
	Reset password
	Change password
	View personal profile
	Update personal profile
	Insert money via Paypal
	View inserted money history
	Comment post
	Like post
	Unlike post
	Connect to Advisor
	Chat text
	Start video call
	End video call
	Chat using Speech to Text
	Tip money for Advisor
	Rate Advisor after finishing conversation
	Report Advisor while chatting
	View conversation history
	Make an appointment with Advisor
	Cancel an appointment with Advisor
	View appointment history
Advisor	Deny Customer's chat request
	Set default free time on week
	Set free time by date
	View payment history
	Assess Post
	Deny Customer's chat request
	Set default free time on week
	Set free time by date
	View payment history
	Assess Post
Admin	View Customers
	View Customer's profile

	View Advisors
	View Advisor's profile
	Create new Advisor account
	Mark exam for Register Advisor
	Block account
	Unblock account
	View domain
	Update domain
	Create new domain
	View sub-domain
	Update sub-domain
	Create new sub-domain
	Create new post
	Edit post
	Delete post
	View conversation report
	Solve conversation report
	Refund gold for customer
	View inserted money
	View tip money
	Create questions bank
	Update questions bank
	Delete questions bank
	View Advisor's feedback
	Pay money for Advisor (assume via Bank)
	Appointment helper

*Table 5-3: Features to be tested*

### 5.3 Test Case

#### 5.3.1 Automation test Unit testing

Unit testing will be done by the developers and approved by the development team leader.

The MindDoctor development team embrace this feature to gain the following advantages:

- Reduce the level of bugs in production code.
- Save development time.

- Automation tests can be run as frequently as required.
- Make it easier to change and refactor code by improving the design of code especially with Test-Driven Development.
- Can easily form a document from the tests.
- Reduce cost of resource to corresponding Front end testing.

### 5.3.1.1 Automation testing framework

All automation test scripts will be created manually and saved to **MindDoctorTest** directory of project.

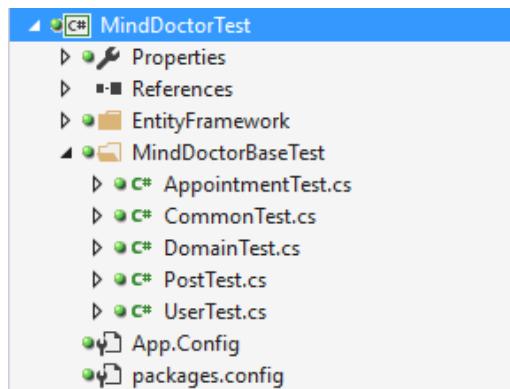


Figure 5-6: Test directory structure

- Unit tests focus on individual functions in class and will be created simply as below:

```
[Test]
public void MakeAnAppointmentTest ()
{
    var db = new MindDoctorDatabaseEntities();
    var appointment = new Appointment()
    {
        AppointmentId = Guid.NewGuid().ToString(),
        StatusId = 0,
        ScheduleId = 1,
        TimeId = 1,
        Advisor = "mds.adv01@mailnesia.com",
        Customer = "datbv95@gmail.com",
        DepositGold = 20,
        SendMail = false,
        CreatedDate = DateTime.Now,
        ModifiedBy = "Admin",
        ModifiedDate = DateTime.Now
    };
    db.Appointments.Add(appointment);
    db.SaveChanges();
}
```

Figure 5-7: Unit test case sample

- Test case are executed automatically by using Nunit as well as by Visual Studio.

✓ CalculateLikesSubDomainTest	22 ms
✓ CancelAnAppointmentTest	481 ms
✓ CreateAdvisorTest	4 ms
✓ CreateDomainTest	3 ms
✓ CreateFreeTimeInScheduleTableTest	33 ms
✓ CreateListFreeTimeModelForSchedul...	1 ms
✓ CreateListFreeTimeViewModelFire...	200 ms
✓ CreatePostTest	3 ms
✓ CreateSubDomainTest	4 ms
✓ CreateUserTest	10 ms
✓ GetAllAdvisorsTest	27 ms
✓ GetAllAppointments	6 ms
✓ GetAllDomainsTest	23 ms
✓ GetAllPostsTest	48 ms
✓ GetAllSubDomainsTest	23 ms

Figure 5-8: Run tests in IDE Visual Studio 2015

### 5.3.2 System testing

Detailed Test cases will be described in **MindDoctor\_TestCase\_Final.xls** file.

As a standard definition, MindDoctor Project defines that a test case is:

- A set of test data and test programs (test scripts) and their expected results. A test case validates one or more system requirements and generates a pass or fail.
- A good test case should follow two basic aspects, the Contents and the Style. Test cases for functional testing are derived from the target of test's use cases. Test cases should be developed for each use case scenario. The use case scenarios are identified by describing the paths through the use case that traverse the basic flow and alternate flows start to finish through the use case.
- By using good automation test and using, MindDoctor Project System testing will not focus on common logic of system like length of text but focus on behavior of website and aims to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution.

### 5.3.3 Defect Log

MindDoctor project used <http://www.trello.com/> to manager bugs and defects.

Every member of VTCHD project created an account in Trello to take part in activities: control bugs, fix bugs, re-test bugs and close bug. Bug will be log by tester or developer in development progress.

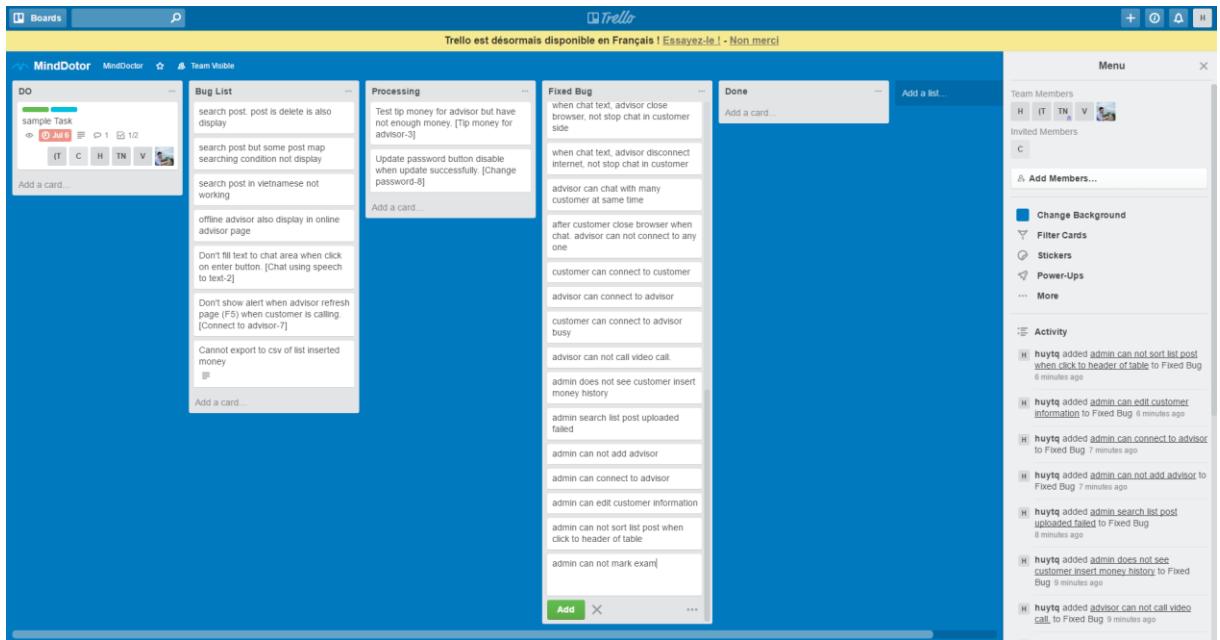


Figure 5-9: Control bug with Trello

## 5.4 Test Report

### 5.4.1 Automation test case report

The contents of the Automation Test Case Report are shown in the table below:

Type of test case	Test Case	Pass	Fail	Not available	Number of Test Case
Unit test - Model	Guest	3	0	0	3
	Customer	9	0	0	9
	Advisor	7	0	0	7
	Admin	4	0	0	4
	Appointment helper	1	0	0	1
<b>Total of Test Case</b>		<b>24</b>	<b>0</b>	<b>0</b>	<b>24</b>

Table 5-4: Automation test case report

#### 5.4.2 Automation test report

Automation test is an integral part of development process. So that, there are 2 phases as 2 phases of Iterative and Incremental Software Process Model.

Test case	Phase 1		Phase 2		Final
	Pass	Fail	Pass	Fail	
Unit test – Model	10	0	14	0	24
Total all test case	10	0	14	0	24

Table 5-5: Automation test report

#### 5.4.3 System test case report

The contents of the System Test Case Report are shown in the table below:

Client	Test Case	Pass	Fail	N / A	No. Test Case
Guest	Register to become Customer	6	0	0	6
	Verify email address	1	0	0	1
	Register to become Advisor	6	0	0	6
	Take qualification exam	5	0	0	5
	View post	5	0	0	5
	Share post on Facebook	4	0	0	4
	Search post	10	0	0	10
	View Advisors's schedule	15	0	0	15
	View online Advisors	9	0	0	9
	View online Advisors by searching Advisor's information	6	0	0	6
Customer	View Advisor's information	1	0	0	1
	Login with local account	6	0	0	6
	Login with Facebook account	3	0	0	3
	Logout	1	0	0	1

Reset password	5	0	0	5
Change password	8	0	0	8
View personal profile	2	0	0	2
Update personal profile	12	0	0	12
Insert money via Paypal	7	0	0	7
View inserted money history	6	0	0	6
Comment post	4	0	0	4
Like post	2	0	0	2
Unlike post	2	0	0	2
Connect to Advisor	7	0	0	7
Chat text	11	0	0	11
Start video call	22	0	0	22
End video call	12	0	0	12
Chat using Speech to Text	4	0	0	4
Tip money for Advisor	5	0	0	5
Rate Advisor after finishing conversation	4	0	0	4
Report Advisor while chatting	3	0	0	3
View conversation history	4	0	0	4
Make an appointment with Advisor	6	0	0	6
Cancel an appointment with Advisor	2	0	0	2

	View appointment history	7	0	0	7
	Deny Customer's chat request	6	0	0	6
	Set default free time on week	6	0	0	6
	Set free time by date	3	0	0	3
	View payment history	4	0	0	4
Advisor	Assess Post	3	0	0	3
	Deny Customer's chat request	5	0	0	5
	Set default free time on week	20	0	0	20
	Set free time by date	2	0	0	2
	View payment history	28	0	0	28
	Assess Post	2	0	0	2
Admin	View Customers	4	0	0	4
	View Customer's profile	1	0	0	1
	View Advisors	2	0	0	2
	View Advisor's profile	2	0	0	2
	Create new Advisor account	4	0	0	4
	Mark exam for Register Advisor	3	0	0	3
	Block account	3	0	0	3
	Unblock account	3	0	0	3
	View domain	3	0	0	3

	Update domain	2	0	0	2
	Create new domain	3	0	0	3
	View sub-domain	3	0	0	3
	Update sub-domain	3	0	0	3
	Create new sub-domain	7	0	0	7
	Create new post	3	0	0	3
	Edit post	3	0	0	3
	Delete post	13	0	0	13
	View conversation report	15	0	0	15
	Solve conversation report	5	0	0	5
	Refund gold for customer	5	0	0	5
	View inserted money	3	0	0	3
	View tip money	16	0	0	16
	Create questions bank	2	0	0	2
	Update questions bank	1	0	0	1
	Delete questions bank	6	0	0	6
	View Advisor's feedback	1	0	0	1
	Pay money for Advisor (assume via Bank)	6	0	0	6
	Appointment helper	5	0	0	5
UI	UI	25	0	0	25
Total of Test Case		426	0	0	426

Table 5-6: System test case report

**Test case list:**

[Register to become customer-1]	Test register to become a customer of Minddoctor system	<ol style="list-style-type: none"> <li>Click on button "Đăng nhập".</li> <li>Click on button "Đăng ký" in login dialog.</li> <li>Fill an existing email into "Email" field, fill at least 8 characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog.</li> <li>Click on button "Đăng ký" in register dialog.</li> </ol>	Register into Minddoctor system successfully and auto login into system	Pass	
[Register to become customer-2]	Test register to become a customer fail. Fill not enough information into required fields	<ol style="list-style-type: none"> <li>Click on button "Đăng nhập".</li> <li>Click on button "Đăng ký" in login dialog.</li> <li>Fill an existing email into "Email" field, fill at least 8 characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog.</li> <li>Click on button "Đăng ký" in register dialog.</li> </ol>	Show a warning in register dialog "Nhập đầy đủ thông tin"	Pass	
[Register to become customer-3]	Test register to become a customer fail. Input have an email that format is wrong	<ol style="list-style-type: none"> <li>Click on button "Đăng nhập".</li> <li>Click on button "Đăng ký" in login dialog.</li> <li>Fill an existing email into "Email" field, fill at least 8 characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog.</li> <li>Click on button "Đăng ký" in register dialog.</li> </ol>	Show a warning in register dialog "Email không đúng định dạng"	Pass	
[Register to become customer-4]	Test register to become a customer fail. Input have an email that not existing	<ol style="list-style-type: none"> <li>Click on button "Đăng nhập".</li> <li>Click on button "Đăng ký" in login dialog.</li> <li>Fill an existing email into "Email" field, fill at least 8 characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog.</li> <li>Click on button "Đăng ký" in register dialog.</li> </ol>	Show a warning in register dialog "Địa chỉ email không tồn tại!"	Pass	
[Register to become customer-5]	Test register to become a customer fail. Input have a password that not validate	<ol style="list-style-type: none"> <li>Click on button "Đăng nhập".</li> <li>Click on button "Đăng ký" in login dialog.</li> <li>Fill an existing email into "Email" field, fill at least 8</li> </ol>	Show a warning in register dialog "Mật khẩu phải lớn hơn hoặc bằng 8 ký tự!"	Pass	

		characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog. 4. Click on button "Đăng ký" in register dialog.		
[Register to become customer-6]	Test register to become a customer fail. Input have a confirm password that not validate	1. Click on button "Đăng nhập". 2. Click on button "Đăng ký" in login dialog. 3. Fill an existing email into "Email" field, fill at least 8 characters into "Mật khẩu" field and fill the same characters with "Mật khẩu" field into "Xác nhận mật khẩu" field in register dialog. 4. Click on button "Đăng ký" in register dialog.	Show a warning in register dialog "Mật khẩu không trùng hợp!"	Pass
[Verify email-1]	Test verify email address	1. Go to mail site. 2. Login by email it's registered to Midoctor system successfully. 3. Open "Xác nhận địa chỉ email" mail from MindDoctor Team. 4. Click on "Xác nhận địa chỉ email" link.	Browser redirect to verify email page of Minddoctor system and show alert that verify email successfully	Pass
[Register to become advisor-1]	Test register to become an advisor successfully	1. Hover on "HỒ TRỌNG" from navigation bar. 2. Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list. 3. Fill enough information for all required fields. 4. Click on "Đăng ký" button in register to advisor page in register to advisor page.	Show a alert that have a test about subdomain knowledge	Pass
[Register to become advisor-2]	Test register to become an advisor fail because not fill all required fields	1. Hover on "HỒ TRỌNG" from navigation bar. 2. Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list. 3. Click on "Đăng ký" button in register to advisor page.	Show an alert: "Vui lòng điền đầy đủ thông tin"	Pass
[Register to become advisor-3]	Test register to become an advisor fail because not fill all required fields	1. Hover on "HỒ TRỌNG" from navigation bar. 2. Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list 3. Fill your name into "Họ và tên" field in register to advisor page but not fill others required field. 4. Click on "Đăng ký" button in register to advisor page in register to advisor page.	Show a warning in register dialog "Email không đúng định dạng"	Pass

[Register to become advisor-4]	Test register to become an advisor fail because not fill all required fields	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> <li>Fill your information into "Họ và tên" field and "Số điện thoại" field in register to advisor page but not fill others required field.</li> <li>Click on "Đăng ký" button in register to advisor page.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	
[Register to become advisor-5]	Test register to become an advisor fail because not fill all required fields	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> <li>Fill your information into "Họ và tên" field, "Số điện thoại" field and select "Lĩnh vực tư vấn" in register to advisor page but not select "Chuyên mục tư vấn".</li> <li>Click on "Đăng ký" button in register to advisor.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	
[Register to become advisor-6]	Test register to become an advisor fail because not fill all required fields	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> <li>Fill your information into "Họ và tên" field, "Số điện thoại" field, select "Lĩnh vực tư vấn" and select "Chuyên mục tư vấn" in register to advisor page but remove information of email field or "Ngày sinh" field.</li> <li>Click on "Đăng ký" button in register to advisor.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	
[Register to become advisor-7]	Test register to become an advisor fail because email not existing	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> <li>Fill into all required fields and email is not existing.</li> <li>Click on "Đăng ký" button in register to advisor.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	
[Register to become advisor-8]	Test register to become an advisor fail because email not right format	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> <li>Fill into all required fields and email is wrong format.</li> <li>Click on "Đăng ký" button in register to advisor.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	
[Register to become advisor-9]	Test register to become an advisor fail because email has been	<ol style="list-style-type: none"> <li>Hover on "HỒ TRỌNG" from navigation bar.</li> <li>Select "ĐĂNG KÝ LÀM TƯ VẤN VIÊN" from "HỒ TRỌNG" dropdown list.</li> </ol>	Show an alert: "Vui lòng điền đầy đủ thông tin" in register to advisor page	<b>Pass</b>	

	registered in MindDoctor system	3. Fill into all required fields and email has been registered in MindDoctor system. 4. Click on "Đăng ký" button in register to advisor.			
[Login-1]	Test login into Minddoctor system successfully	1. Click on button "ĐĂNG NHẬP". 2. Fill the right information of both email and password in login dialog. 3. Click on button "Đăng nhập" in login dialog.	login into Minddoctor system successfully	Pass	
[Login-2]	Test login into Minddoctor system fail. Fill not enough information into all required fields	1. Click on button "ĐĂNG NHẬP". 2. Fill information of email but not for password or fill information of password but not for email or not fill both of them. 3. Click on button "Đăng nhập" in login dialog.	Show a warning in login dialog: "Nhập email và mật khẩu!"	Pass	
[Login-3]	Test login into Minddoctor system that fill a wrong password	1. Click on button "ĐĂNG NHẬP". 2. Fill an existing email and wrong password in login dialog. 3. Click on button "Đăng nhập" in login dialog.	Show a warning in login dialog: "Mật khẩu không đúng!"	Pass	
[Login-4]	Test login into Minddoctor system that fill a not existing email in Minddoctor	1. Click on button "ĐĂNG NHẬP". 2. Fill an email that not existing in Minddoctor system and any password in login dialog. 3. Click on button "Đăng nhập" in login dialog.	Show a warning in login dialog: "email không tồn tại!"	Pass	
[Login-5]	Test login into Minddoctor system that fill a not right email's format	1. Click on button "ĐĂNG NHẬP". 2. Fill an email that not right format and any password in login dialog. 3. Click on button "Đăng nhập" in login dialog.	Show a warning in login dialog: "email không đúng định dạng!"	Pass	
[Login-6]	Test login into Minddoctor system fail, because the internet was disconnected	1. Click on button "ĐĂNG NHẬP". 2. Fill the right information of both email and password in login dialog. 3. Click on button "Đăng nhập" in login dialog. 4. When waiting login, disconnect the internet.	Login into Minddoctor fail	Pass	

For more details about all test suites, please look at file “**MindDoctor\_TestCase\_Final.xls**”

#### 5.4.4 System test report

We execute test with 2 stages with 2 phases of process model, to finish project.

The contents of the Test Report are shown in the table below:

Test Case		Stage 1		Stage 2-2		Final		Test Case Pass
		Pass	Fail	Pass	Fail	Pass	Fail	
Guest	68	30	0	38	0	68	0	68
Customer	145	61	0	84	0	145	0	145
Advisor	42	30		12	0	42	0	42
Admin	145	91	0	54	0	145	0	145
System	1	0	0	1	0	1	0	1
GUI Test Case		10	0	15	0	15	0	15
<b>Total all Test Case</b>		<b>202</b>	<b>0</b>	<b>226</b>	<b>0</b>	<b>426</b>	<b>0</b>	<b>426</b>

Table 5-7: System test report

#### 5.4.5 Benchmarks

##### 5.4.5.1.1 Benchmark Tool



Figure 5-10 : <https://tools.pingdom.com/>

We use **Pingdom** to do our benchmark. **Pingdom** is a Website Speed Test. It's a tool to help webmasters and web developers everywhere optimize the performance of their websites. It supports: **HTML, JavaScript, CSS, images**, etc.

##### 5.4.5.1.2 Benchmark implementation

With **Pingdom** we can write a benchmark script that simulates realistic user behavior with scenarios.

- We will configure the **Pingdom** as following:
  - Run on address: <https://52.168.135.19/>
  - Then we run the benchmark with **Pingdom** tool:

The form shows the URL as https://52.168.135.19/ and the location as Dallas, Texas, USA. A green 'START TEST' button is visible.

##### 5.4.5.1.3 Benchmark result

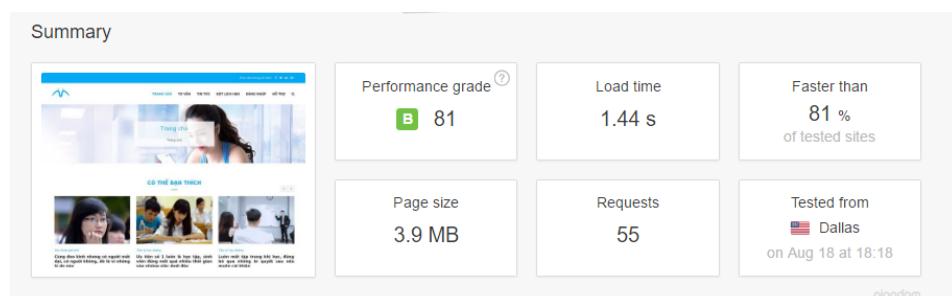


Figure 5-11: Overall performance

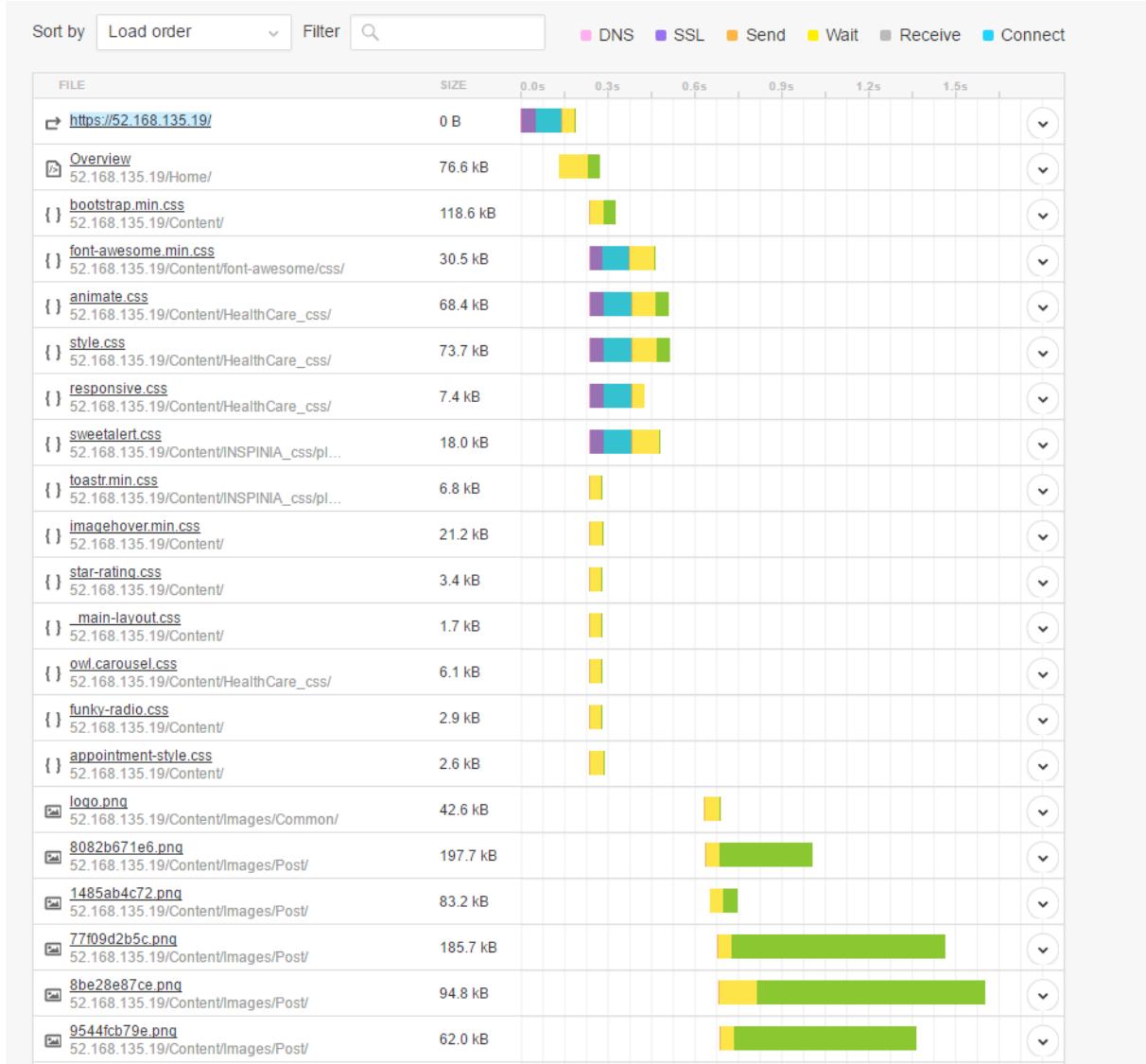


Figure 5-12: Loading time

The benchmark shows that our system works as expected under the high load we defined in the first place:

- The maximum response time is 276ms (only some first requests)
- The median response time is 1.44s Benchmarks

#### 5.4.6 Usability in mobile

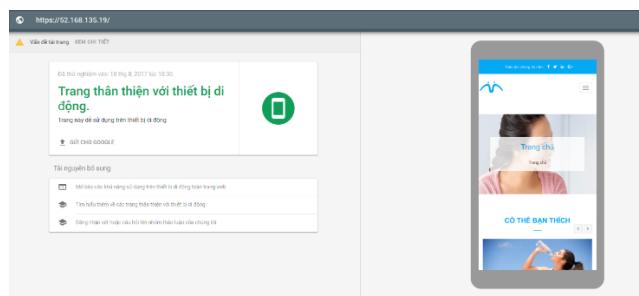


Figure 5-13: <https://search.google.com/test/mobile-friendly>

MindDoctor design with Bootstrap, AngularJS, jQuery so UI is responsive, easy to use in mobile web.

### 5.4.7 Stress test

- MindDoctor use JMeter tool to execute stress test.
- Stress testing is a form of testing that is used to determine the stability of a given system or entity.



Figure 5-14: JMeter 2.13

- MindDoctor execute test case 150 user access to website in the same time with result:  
The maximum response time is 3114ms.

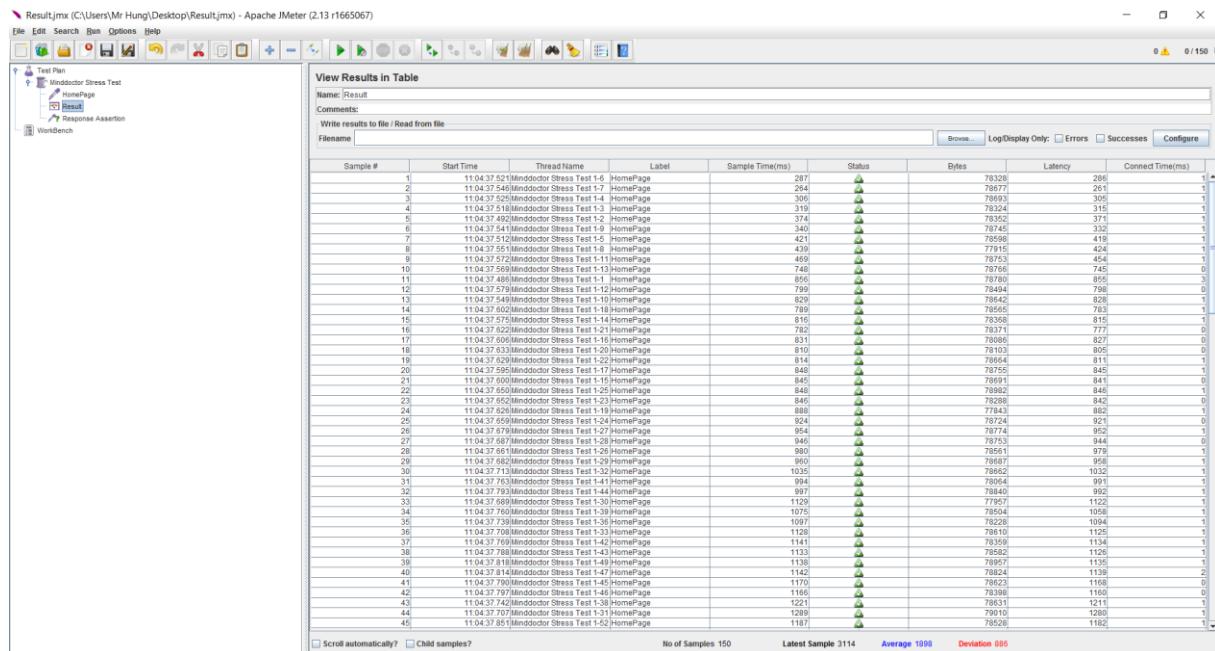


Figure 5-15: Test result

# Chapter 6 : User manual

## 6.1 Deployment guidelines

### 6.1.1 Microsoft Azure

- Go to <https://portal.azure.com/#create/hub>
- Create virtual machine with configuration is recommended
  - Click the **New** button found on the upper left-hand corner of the Azure portal.
  - Select **Compute**, and then select **Windows Server 2016 Datacenter**.
  - Enter the virtual machine information. The user name and password entered here is used to log in to the virtual machine. When complete, click **OK**.

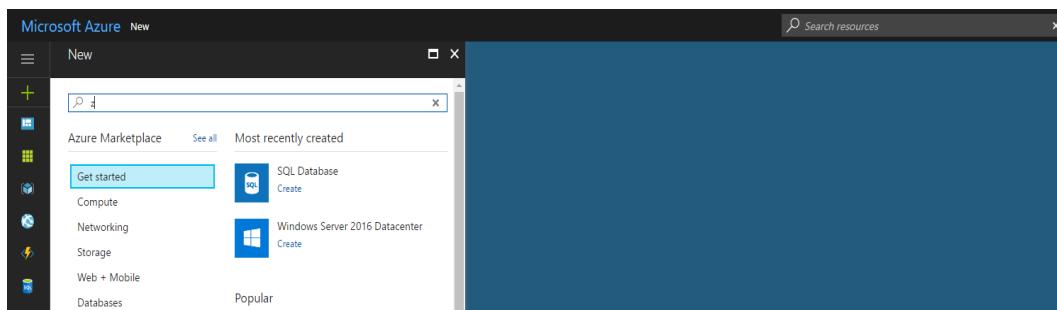


Figure 6-1: Microsoft Azure

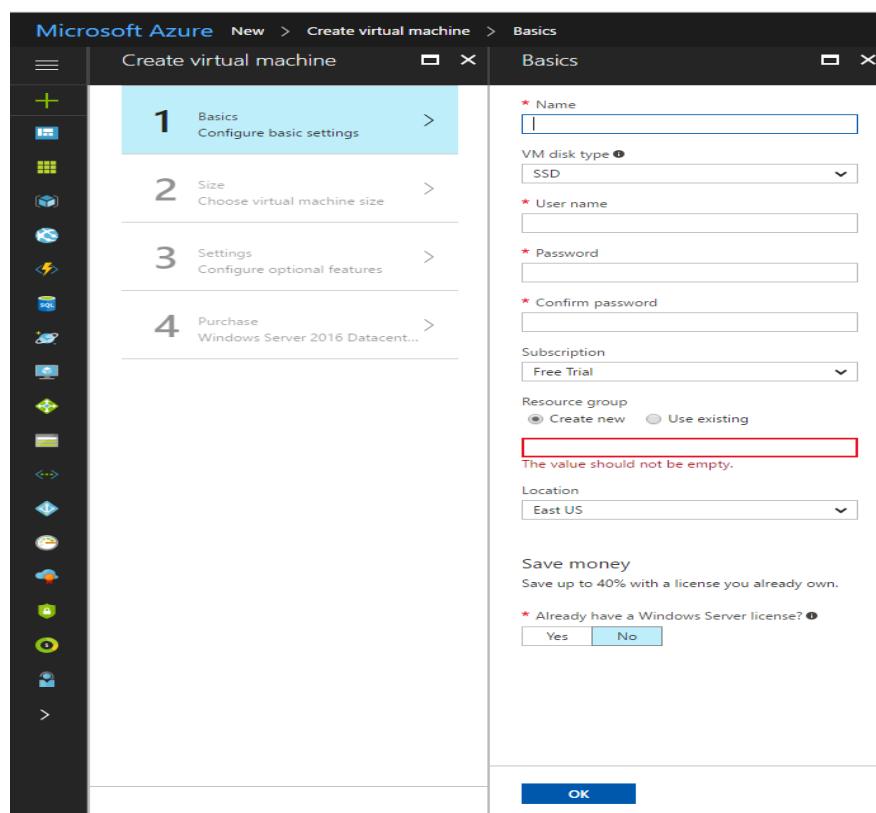
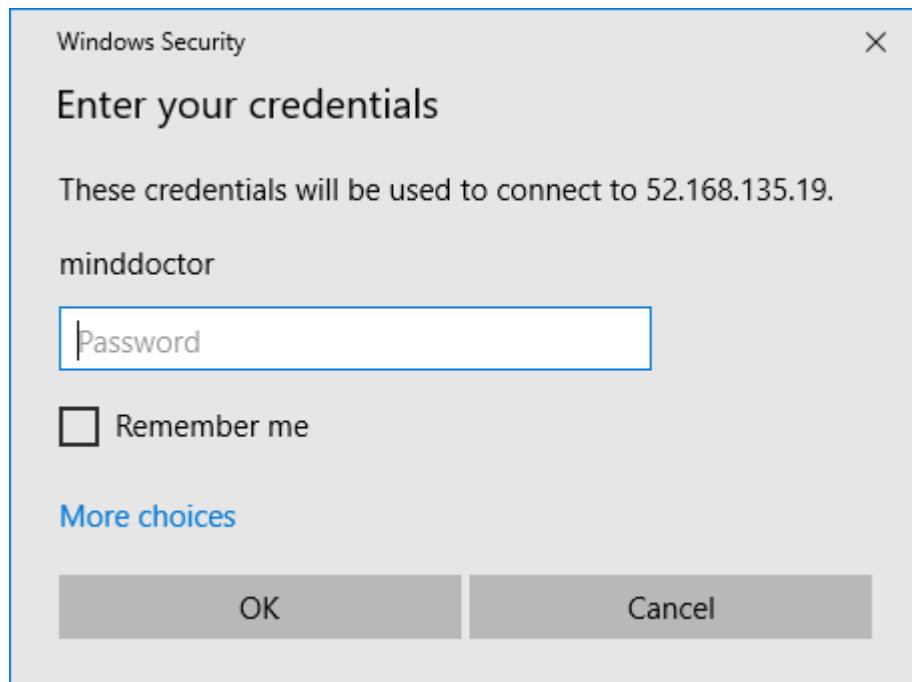


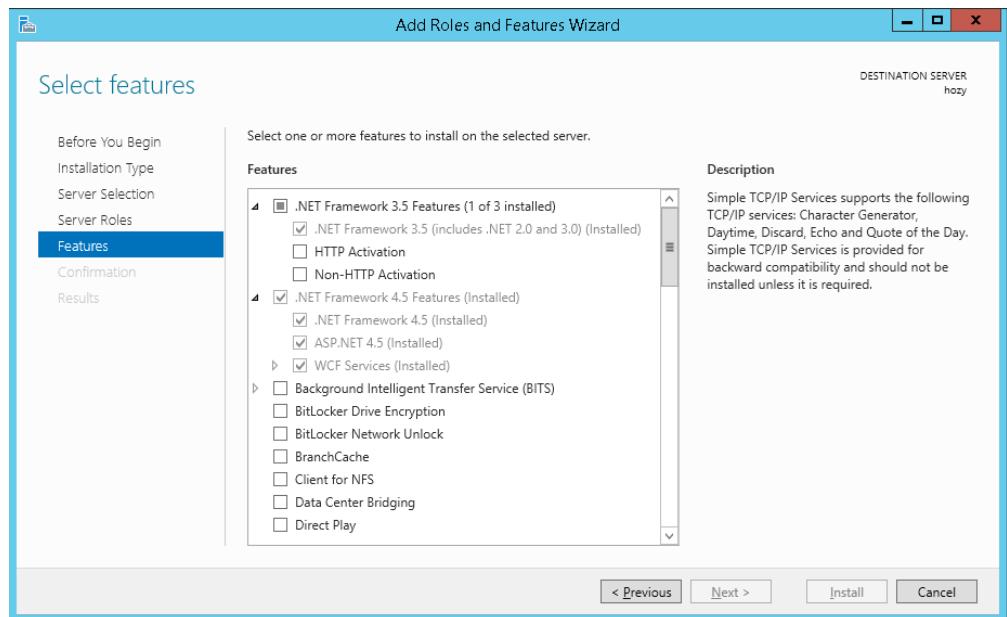
Figure 6-2: Microsoft Azure

- Connect to VM instance with provided Account and Password



*Figure 6-3: Connect to VM*

- Install corresponding services : IIS and .NET services from Server Manager



*Figure 6-4: Install correspoding services*

- Configure IIS 10

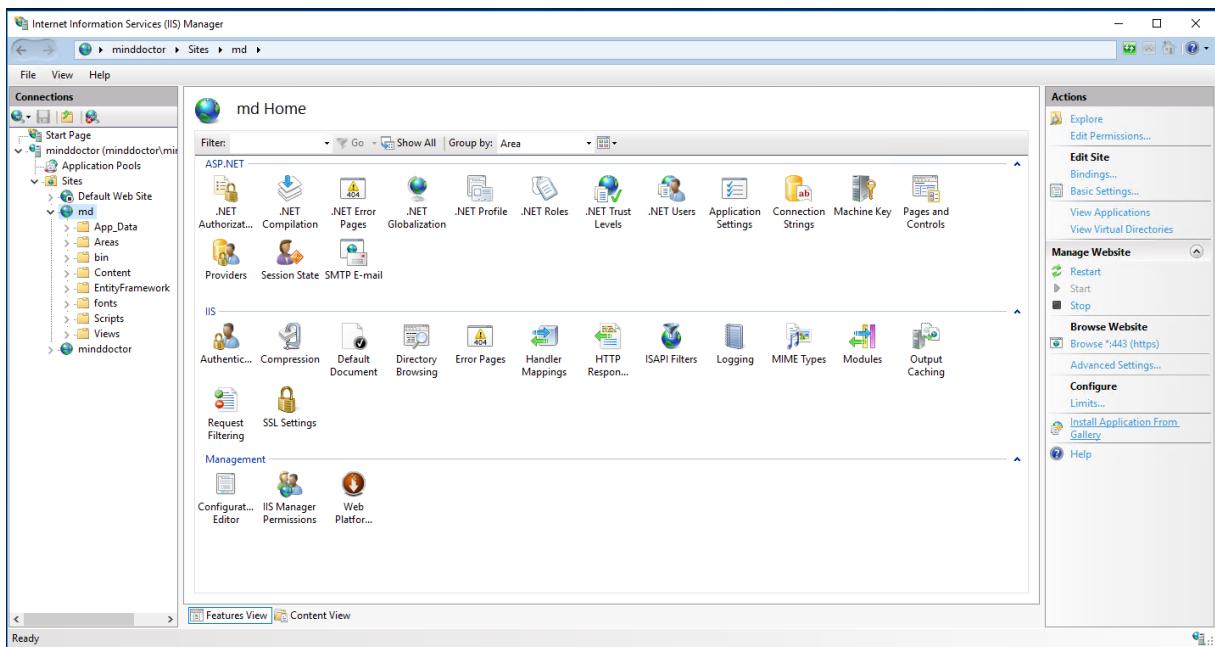


Figure 6-5: Configure IIS

### 6.1.2 SQL Server 2012 Enterprise

Get SQL Server 2012 Enterprise Edition and Install on previously created VM.

Student of FPT University get free access for SQL Server 2012 Enterprise after logging in on DreamSpark for FPT.

Full installation including SQL Management Studio is recommended.

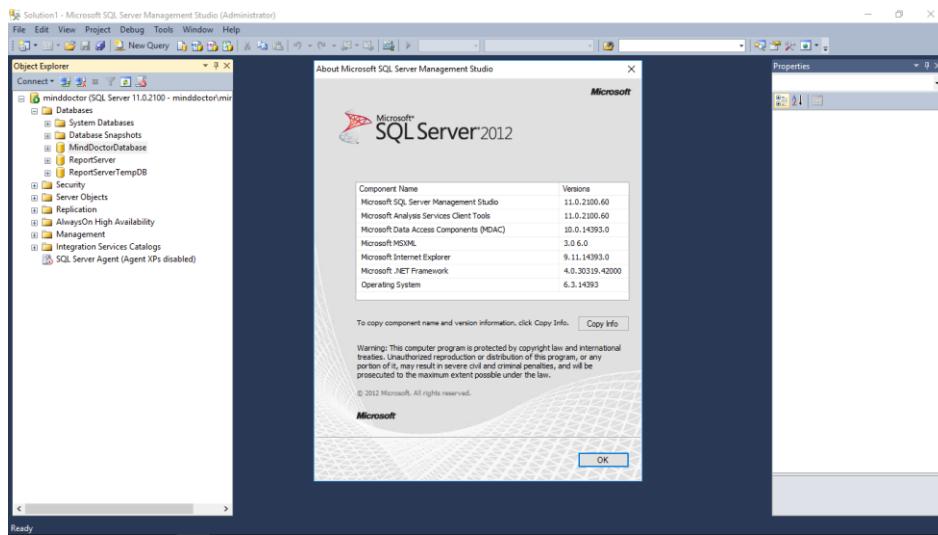


Figure 6-6: SQL Server management

## 6.2 User guidelines

### 6.2.1 Sign in MindDoctor system

1. The first time visiting website, welcome page will be appeared, user clicks on “Tham gia ngay” to join with us.

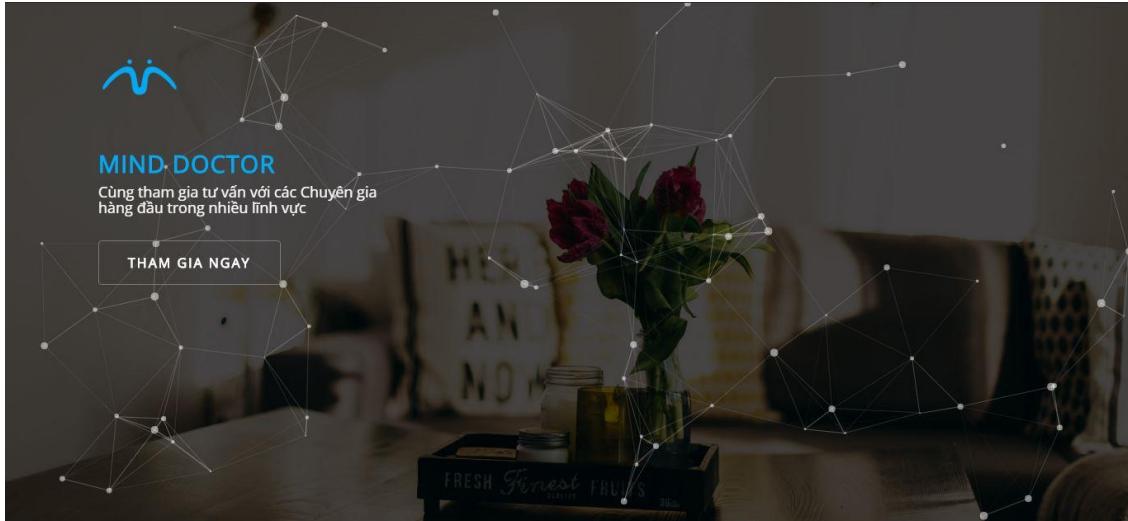


Figure 6-7: Welcome page

2. User clicks on “Đăng nhập” in the navigation bar to login.

ĐẶT LỊCH HẸN      ĐĂNG NHẬP      HỖ TRỢ

Figure 6-8: Login button

3. Browser displays login dialog.

The image shows a login dialog box titled "ĐĂNG NHẬP" (Login) in a blue header. The form contains two input fields: one for "Email..." with a user icon and another for "Mật khẩu..." with a lock icon. Below the inputs is a large blue "Đăng nhập" (Login) button. Underneath the button is a blue bar with the text "f Đăng nhập với tài khoản Facebook" (Facebook login). At the bottom of the dialog, there are two links: "Quên mật khẩu?" (Forgot password?) and "Đăng ký" (Sign up) preceded by a checkmark.

Figure 6-9: Login dialog

4. User sign in with local account or Facebook account. If user does not have any account yet, they can sign up with local account or Facebook account by clicking on “Đăng ký”.



Figure 6-10: Register dialog

### 6.2.2 Update profile

1. User hovers on display name on the navigation bar and clicks on “Hồ sơ cá nhân” in dropdown menu.

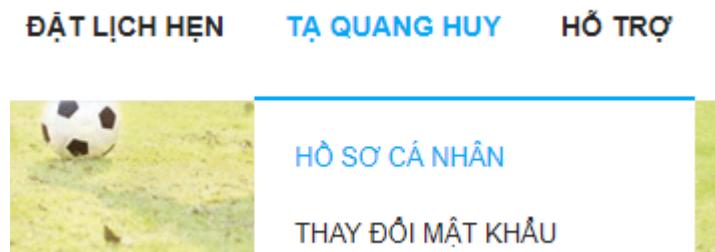


Figure 6-11: Dropdown menu

2. User updates avatar by click on “Cập nhật ảnh đại diện”. Besides, user also can update personal information on the right.

**HỒ SƠ CÁ NHÂN**



Cập nhật ảnh đại diện

0

**Thông tin cá nhân**

Họ và tên:	Ta Quang Huy
Ngày sinh:	06/25/1995
Giới tính:	Nam
Email:	huytq2506@gmail.com
Quốc gia:	Việt Nam
Tỉnh/Thành phố:	Nghệ An

**Cập nhật**

Figure 6-12: Personal profile

### 6.2.3 Interact with post

- User hovers on “Tin tức” on the navigation bar and select domain which wants to see.

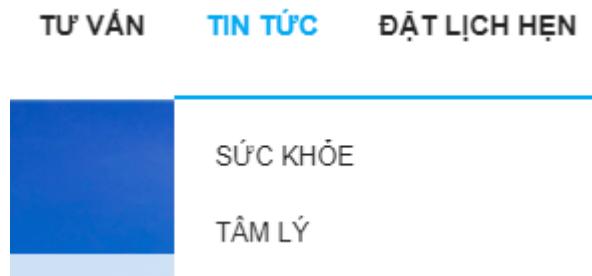


Figure 6-13: Select domain

- Browser displays posts of domain which user chooses.

**TIN TỨC**



Sức khỏe giới tính

Uống nước ngay sau khi ăn: thói quen khiến hệ tiêu hóa của bạn sẽ gặp đủ vấn đề

Thói quen này rất nhiều người mắc phải khiến hệ tiêu hóa lẫn sức khỏe ngày càng yếu đi.



Sức khỏe giới tính

Chuyên gia chỉ ra con số chính xác thịt, cá, rau... cần ăn mỗi ngày để cơ thể luôn khỏe mạnh

Người lớn cần ăn bao nhiêu là đủ chất dinh dưỡng? Chuyên gia sẽ bắt mí lượng thực phẩm chính xác để bạn có thể an tâm và áp dụng nó trong thực đơn hàng ngày.



Sức khỏe giới tính

Bóp ngay những thói quen xấu buổi sáng khiến bạn tăng cân vèo vèo

Thói quen xấu buổi sáng cũng vô hình là tác nhân làm cho bạn mãi chẳng giảm được cân nào.

**CHUYÊN MỤC**

- Sức khỏe giới tính

**LĨNH VỰC**

- Sức khỏe
- Tâm lý

**ĐÃ XEM GẦN ĐÂY**



Sức khỏe giới tính

Uống nước ngay sau khi ăn: thói quen khiến hệ tiêu hóa của bạn sẽ gặp đủ vấn đề

Thói quen này rất nhiều người mắc phải khiến hệ tiêu hóa lẫn sức khỏe ngày càng yếu đi.



Sức khỏe giới tính

Đừng quên chải lưỡi khi vệ sinh răng miệng nếu không muốn mắc bệnh nguy hiểm sau

Nếu không có thói quen làm sạch lưỡi mỗi ngày, bạn sẽ rất dễ mắc phải các

Figure 6-14: Posts

3. User can search post by clicking on search icon on the navigation bar.



Figure 6-15: Search bar

Lĩnh vực:

Sức khỏe

Chuyên mục:

-----

Tìm kiếm

**TIN TỨC**



Hãy áp dụng những cách làm trắng răng này: vừa rẻ mà vừa hiệu quả  
Sức khỏe giới tính  
Đừng để hàm răng vàng ố làm bạn mất tự tin. Sử dụng ngay 1 trong các nguyên liệu sau bao đảm răng trắng sáng như ý.



Uống nước rồi mà vẫn thấy khát, có thể bạn đã mắc phải 1 trong 5 bệnh nguy hiểm sau  
Sức khỏe giới tính  
Khát nước bất thường không đơn giản chỉ là thiếu nước bạn nhé, bởi đôi khi đó là triệu chứng của 1 loại bệnh đấy.



Ăn uống cũng phải chọn kỹ nếu không muốn tình trạng hôi miệng làm bạn khó chịu  
Sức khỏe giới tính  
Không ngờ, nguyên nhân gây hôi miệng lại nằm từ chính những thứ mà chúng ta ăn hàng ngày.



Sức khỏe giới tính  
Uống nước ngay sau khi ăn: thói quen khiến hệ tiêu hóa của bạn sẽ gặp đủ vấn đề  
Thói quen này rất nhiều người mắc phải khiến hệ tiêu hóa lẫn sức khỏe ngày càng yếu đi.



Figure 6-16: Search post

4. User also can search with filter in search result page.

Lĩnh vực:

Sức khỏe

Chuyên mục:

-----

Tìm kiếm

Figure 6-17: Search area

5. User clicks on post to view detail.

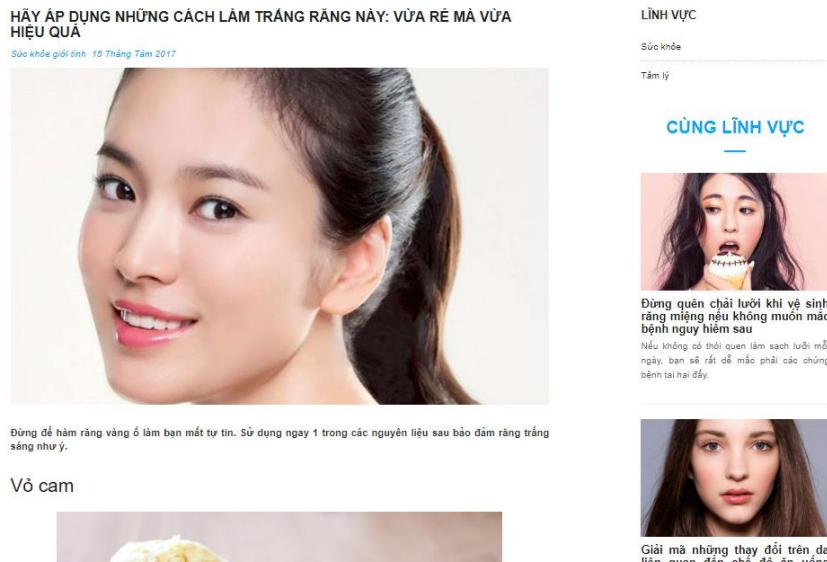


Figure 6-18: Post detail

6. User can share, like and comment post.



Figure 6-19: Comment area

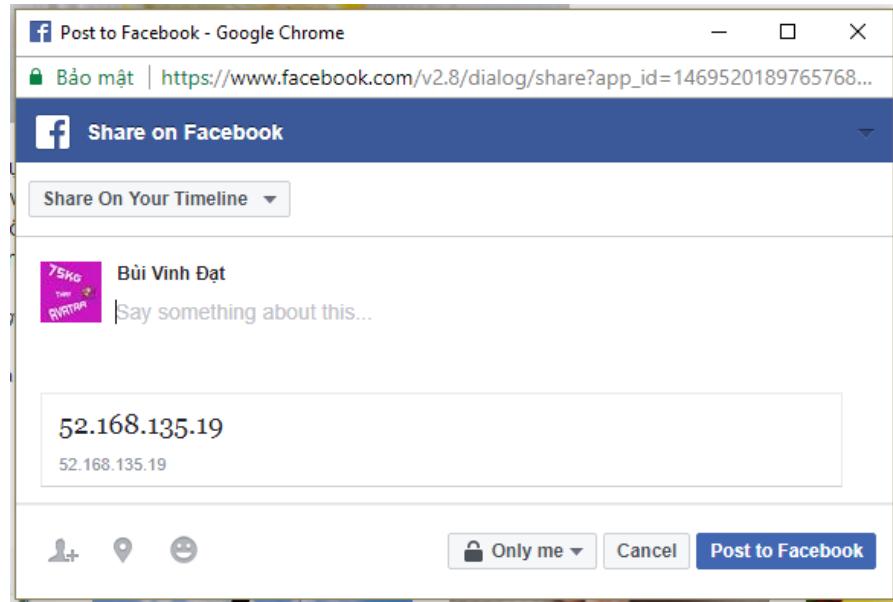


Figure 6-20: Share post

#### 6.2.4 View Advisors

- User hovers on “Hỗ trợ” and clicks on “Danh sách Tư vấn viên” to view all Advisors off MindDoctor system.

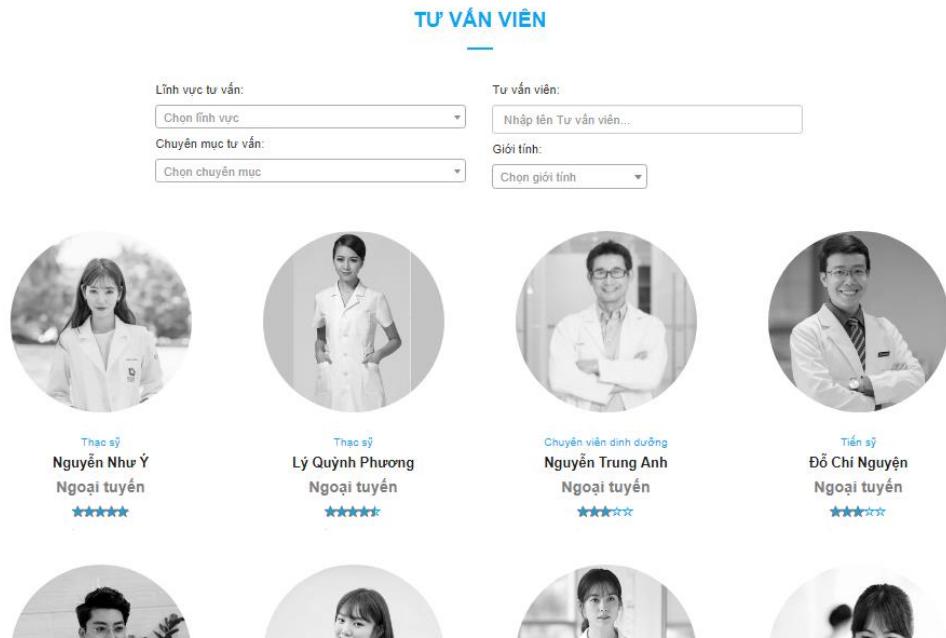


Figure 6-21: Advisors

- User hovers on “Tư vấn” and clicks on which domain wants to be consulted. Browser displays all online Advisors who can consult right now.

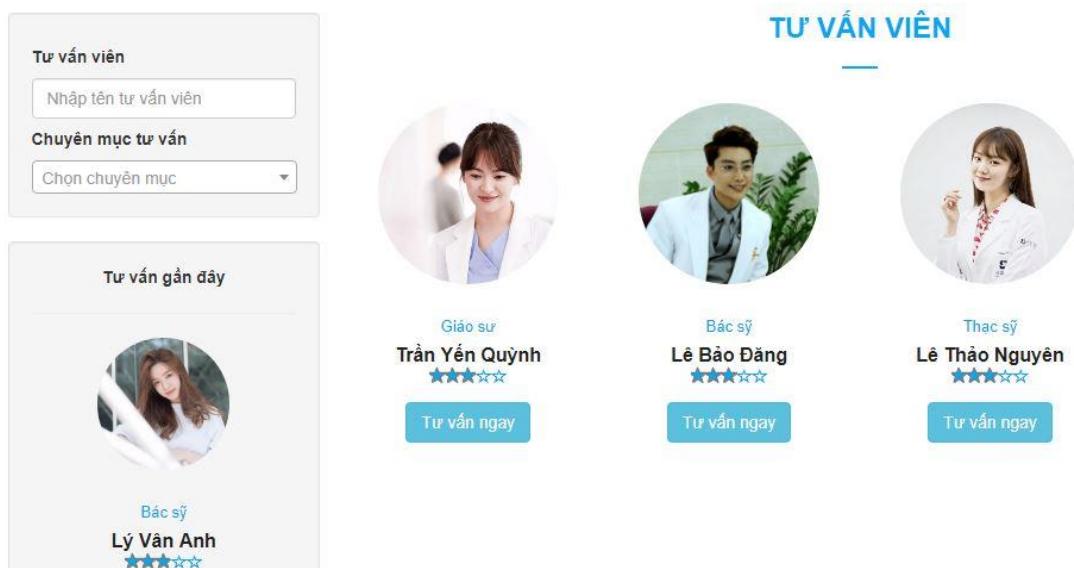


Figure 6-22: Online advisors

3. User also search advisor with advisor's information.

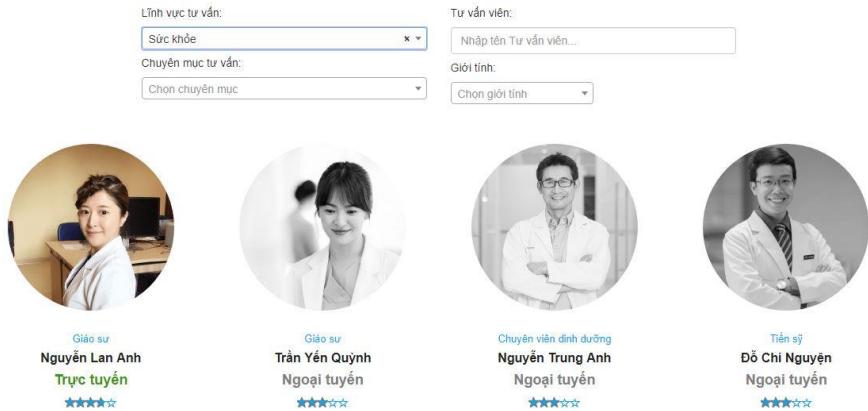


Figure 6-23: Welcome page

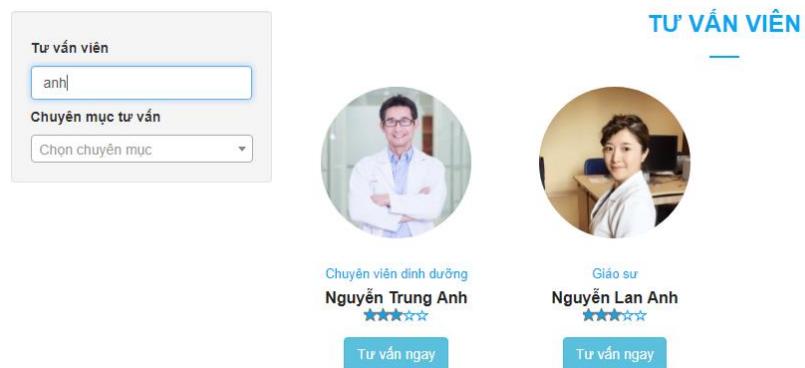


Figure 6-24: Search advisor

4. User clicks on advisor's avatar to see more information.

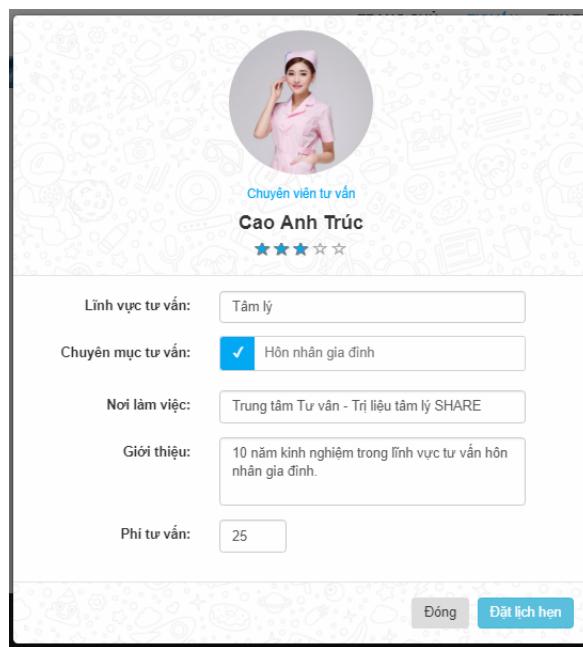


Figure 6-25: Advisor information

### 6.2.5 Set up free time (Only for Advisor account)

1. User log into system with advisor's account. User hovers on display name on the navigation bar and click on "Cài đặt lịch hẹn mặc định". This function is support for advisor on setting free time to show to customer.

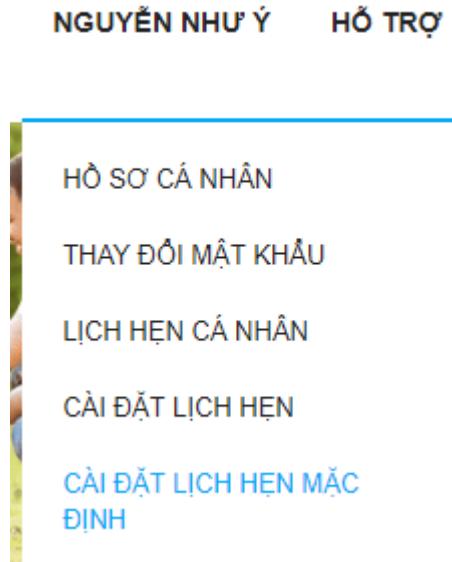


Figure 6-26: Dropdown menu

CÀI ĐẶT LỊCH HẸN MẶC ĐỊNH					
Thứ hai	Thứ ba	Thứ tư	Thứ năm	Thứ sáu	Thứ bảy
08:00 - 08:20	08:00 - 08:20	08:00 - 08:20	08:00 - 08:20	08:00 - 08:20	08:00 - 08:20
08:20 - 08:40	08:20 - 08:40	08:20 - 08:40	08:20 - 08:40	08:20 - 08:40	08:20 - 08:40
08:40 - 09:00	08:40 - 09:00	08:40 - 09:00	08:40 - 09:00	08:40 - 09:00	08:40 - 09:00
09:00 - 09:20	09:00 - 09:20	09:00 - 09:20	09:00 - 09:20	09:00 - 09:20	09:00 - 09:20
09:20 - 09:40	09:20 - 09:40	09:20 - 09:40	09:20 - 09:40	09:20 - 09:40	09:20 - 09:40
09:40 - 10:00	09:40 - 10:00	09:40 - 10:00	09:40 - 10:00	09:40 - 10:00	09:40 - 10:00

Figure 6-27: Setting default schedule

2. User hovers on display name on the navigation bar and click on “Cài đặt lịch hẹn”. It will be shown for customer.



**CÀI ĐẶT LỊCH HẸN**

Thứ Bảy, 19 Tháng Tám 2017

Cài đặt

Figure 6-28: Setting schedule

**CÀI ĐẶT LỊCH HẸN**

Thứ Bảy, 19 Tháng Tám 2017

Cài đặt

08:00 - 08:20	08:20 - 08:40	08:40 - 09:00
09:00 - 09:20	09:20 - 09:40	09:40 - 10:00
10:00 - 10:20	10:20 - 10:40	10:40 - 11:00
11:00 - 11:20	11:20 - 11:40	11:40 - 12:00
12:00 - 12:20	12:20 - 12:40	12:40 - 13:00

Figure 6-29: Setting schedule

### 6.2.6 Make an appointment with Advisor

- User clicks on “Đặt lịch hẹn” on the navigation bar. User can view free time of advisor by searching with filter and text.

**ĐẶT LỊCH HẸN**

Lĩnh vực tư vấn: \_\_\_\_\_ Ngày: Chủ Nhật, 20 Tháng Tám 2017

Chuyên mục tư vấn: \_\_\_\_\_ Giới tính: \_\_\_\_\_



Figure 6-30: Search appointment area

08:00 - 08:20 	08:20 - 08:40 	08:40 - 09:00 
09:00 - 09:20 	09:20 - 09:40 	09:40 - 10:00 
10:00 - 10:20 	10:20 - 10:40 	10:40 - 11:00 

Figure 6-31: Appointment result

- User clicks on advisor's avatar to view more information. After that, user clicks on “Đặt lịch hẹn” to make an appointment with advisor.

  
Chuyên viên tư vấn  
**Lý Hưng**  


Ngày: 09:40 - 10:00 Thứ Hai, 14 Tháng Tám 2017  
Lĩnh vực tư vấn: Tâm lý  
Chuyên mục tư vấn:  Hôn nhân gia đình  
Nơi làm việc: Phòng Tư vấn tâm lý Nam Anh  
Giới thiệu:  
Có kinh nghiệm 10 năm trong lĩnh vực tư vấn tâm lý hôn nhân gia đình.  
Đã giúp đỡ trên 20 cặp vợ chồng thoát khỏi  
Phí tư vấn: 20

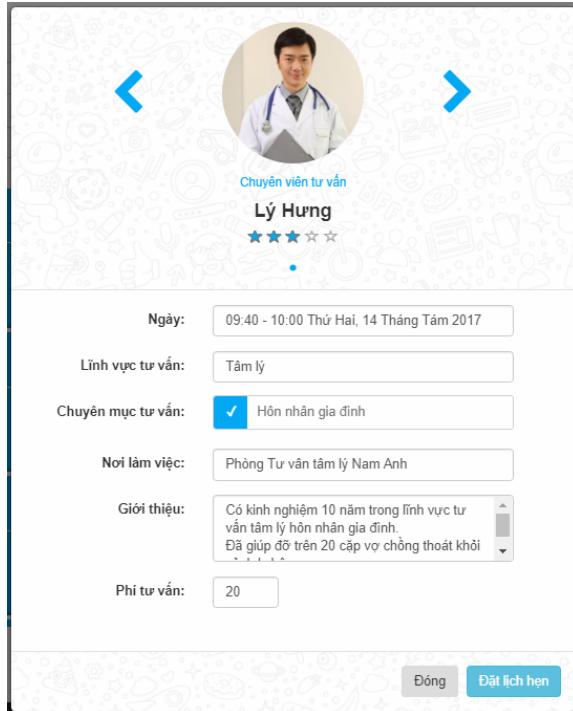


Figure 6-32: Advisor information

3. User hovers on display name on the navigation bar and click on “Lịch hẹn cá nhân” on dropdown menu to view booked appointment.

Figure 6-33: Appointment history

### 6.2.7 Make a conversation with advisor

1. User clicks on “Tư vấn” and select domain which wants to be consulted to view online advisors. After that, user selects an advisor to send consultant request.

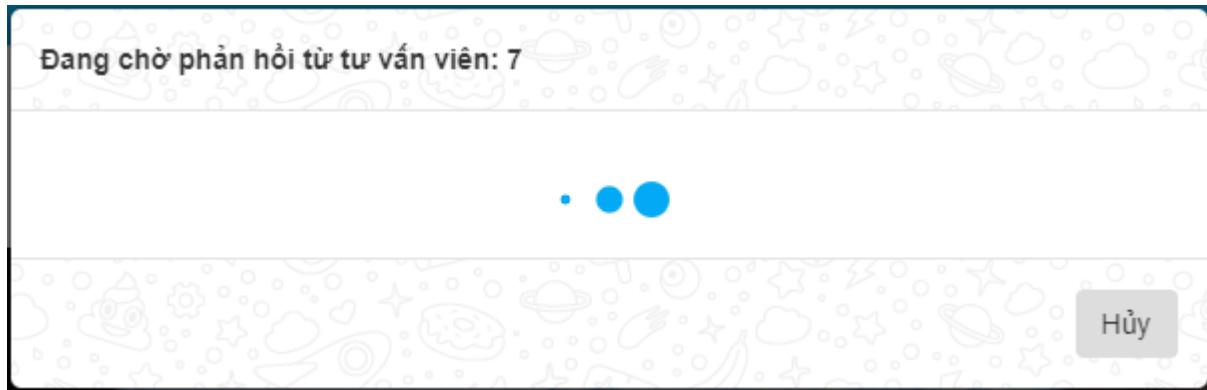


Figure 6-34: Connect to advisor

2. If advisor accepts user's request, they will be redirected to chat page to start conversation.

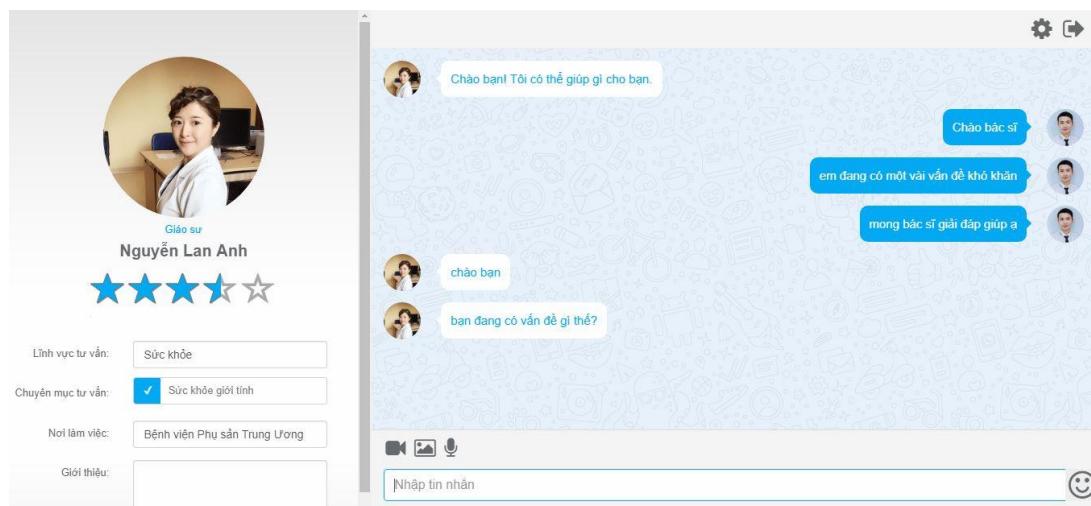


Figure 6-35: Conversation

3. In other case, if customer booked an appointment with advisor they will receive email to notice that appointment will be soon. Besides, if they are online, notifications will be pushed.

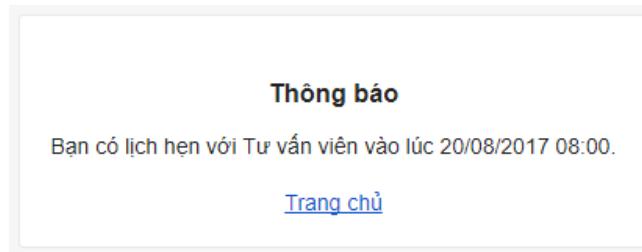


Figure 6-36: Email notification

4. Notification is pushed to user.

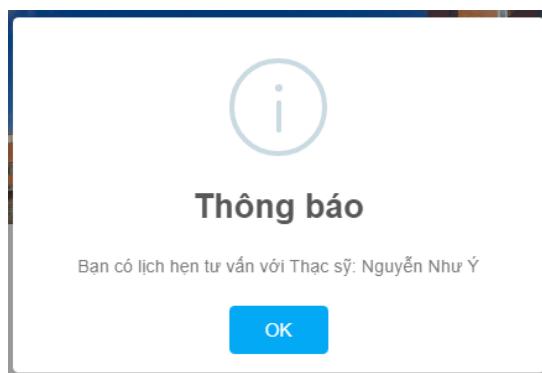


Figure 6-37: Notification

5. If users accept request, they will go to “Waiting room”.



## PHÒNG CHỜ



Figure 6-38: Waiting room