

California State University, San Bernardino

**CSUSB ScholarWorks**

---

Theses Digitization Project

John M. Pfau Library

---

2004

## Traditional Chinese medical clinic system

Chaomei Liu

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Liu, Chaomei, "Traditional Chinese medical clinic system" (2004). *Theses Digitization Project*. 2517.  
<https://scholarworks.lib.csusb.edu/etd-project/2517>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

TRADITIONAL CHINESE MEDICAL CLINIC SYSTEM

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science

---

by  
Chaomei Liu  
December 2004

TRADITIONAL CHINESE MEDICAL CLINIC SYSTEM

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

by  
Chaomei Liu  
December 2004

Approved by:



Dr. David A. Turner, Chair, Computer  
Science



Dr. Richard Botting



Dr. Ernesto Gomez

*Oct 15, 2004*

Date

## ABSTRACT

Traditional Chinese Medical Clinic System (TCMCS) is designed for a small acupuncture office. The software will be used by assistants and acupuncturists in the clinic to schedule appointments, and record patient diagnoses and case histories. With the Clinic System database, assistants and acupuncturists can search and keep track of patient information effectively. Furthermore, an acupuncturist can treat different patients in different rooms, and record their case histories in each room through the intranet, which will save time and improve the working efficiency.

## ACKNOWLEDGMENTS

I would like to convey my gratitude for all the faculty of Computer Science Department in Computer Science at California State University, San Bernardino.

I would like also to express my sincere appreciation to my graduate advisor, Dr. David Turner and other committee members, who are Dr. Richard Botting and Dr. Ernesto Gomez for their valuable input.

And finally, I would like to thank my husband, who is the most important person in my life. Without his support, I would not be able to study and get my Master degree.

The support of the National Science Foundation under the award 9810708 is gratefully acknowledged.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
CHAPTER ONE: SOFTWARE REQUIREMENTS SPECIFICATION	
1.1 Introduction .....	1
1.2 Purpose of the Project .....	2
1.3 Context of the Problem .....	2
1.4 Significance of the Project .....	3
1.5 Assumptions .....	4
1.6 Limitations .....	4
1.7 Definition of Terms .....	5
1.8 Organization of the Thesis .....	7
CHAPTER TWO: SOFTWARE DESIGN	
2.1 Introduction .....	8
2.1 System Architecture and Design .....	8
2.2.1 Class Diagram .....	11
2.3 Database Design .....	12
2.3.1 Database Schema Conceptual Model - ER Diagram .....	16
2.3.2 Data Type and Details .....	16
CHAPTER THREE: SOFTWARE QUALITY ASSURANCE	
3.1 Introduction .....	22
3.2 Unit Test Plan .....	22

3.3 System Test Plan .....	36
CHAPTER FOUR: USERS MANUAL	
4.1 Introduction .....	37
4.2 Traditional Chinese Medical Clinic System Graphical User Interface Design .....	38
4.2.1 The Traditional Chinese Medical Clinic System and Traditional Chinese Medical Clinic System Page .....	39
4.2.2 Staff's Page .....	41
4.2.3 Schedule an Appointment Page .....	43
4.2.4 View or Edit Appointments Page .....	44
4.2.5 Create New Patient .....	45
4.2.6 View or Modify Patient's Information .....	46
4.2.7 The Acupuncturist's Page .....	47
4.2.8 View Appointment Page .....	48
4.2.9 Patient Search Page .....	49
4.2.10 Manage User's Accounts Page .....	50
4.2.11 Case History Page .....	52
4.2.12 Inspection Page .....	53
4.2.13 Inquiry Page .....	54
4.2.14 Auscultation and Olfaction Page .....	55
4.2.15 Palpation and Pulse-Taking Page .....	56
4.2.16 Selected Items Page .....	57
4.2.17 Fourteen Meridian Page .....	58
4.2.18 Lung Channel of Hand Tai-Yin .....	59

4.2.19 Prescription Page .....	60
4.2.20 The Herb's Quantity's Page.....	61
4.2.21 Diagnosis Page .....	62
4.2.22 Edit Current Case History Page .....	63
4.2.23 Print Current Case History Report Page .....	64
4.2.24 The Print Prescription Page .....	65
4.2.25 View Patient's Previous Case History Page .....	66
4.2.26 The Medical History Page .....	67
4.2.27 Patient Information .....	68

CHAPTER FIVE: MAINTENANCE

5.1 Introduction .....	70
5.2 Software Installation .....	70
5.2.1 Internet Information Service Installation .....	70
5.2.2 MS SQL Server 2000 .....	71
5.2.3 Grand Access the ASPNET User .....	73
5.3 Traditional Chinese Medical Clinic System Installation .....	74
5.4 Backup and Restore .....	75
5.4.1 System Backup .....	75
5.4.2 Database Backup .....	76
5.4.3 System Restore .....	77
5.4.4 Database Restore .....	77

CHAPTER SIX: CONCLUSION AND FUTURE DIRECTIONS

6.1 Conclusion .....	79
----------------------	----



6.2 Future Directions .....	80
APPENDIX: SOURCE CODE PRINTOUT .....	81
REFERENCES .....	181

LIST OF TABLES

Table 1.	Definition, Acronyms and Abbreviations .....	5
Table 2.	Structure of Table CaseHistory .....	17
Table 3.	Structure of Table DiagnosisList1 .....	17
Table 4.	Structure of Table HerbsList .....	17
Table 5.	Structure of Table Inquiry .....	18
Table 6.	Structure of Table ListenSmell .....	18
Table 7.	Structure of Table Look .....	18
Table 8.	Structure of Table Palpation .....	19
Table 9.	Structure of Table TypesOfPulse .....	19
Table 10.	Structure of Table MedicalHistory .....	20
Table 11.	Structure of Table NewAppointments .....	20
Table 12.	Structure of Table Patients .....	21
Table 13.	Structure of Table StaffAccounts .....	21
Table 14.	The Acupuncturist Side's Test Plan .....	22
Table 15.	Staff Side Test Plan .....	34
Table 16.	System Test Results .....	36

## LIST OF FIGURES

Figure 1.	3-Tier Client/Server .....	9
Figure 2.	Class Diagram without Attributes and Operations .....	11
Figure 3.	ER Diagram .....	16
Figure 4.	Traditional Chinese Medical Clinic System Use Case Diagram .....	38
Figure 5.	Traditional Chinese Medical Clinic System Page .....	40
Figure 6.	Chinese Medical Clinic System Page with Login Error Message .....	41
Figure 7.	The Staff's Page .....	42
Figure 8.	Schedule an Appointment Page .....	44
Figure 9.	View or Edit Appointments Page .....	45
Figure 10.	Create New Patients Page .....	46
Figure 11.	View or Modify Patient's Information Page .....	47
Figure 12.	The Acupuncturist's Page .....	48
Figure 13.	View Appointment's Page .....	49
Figure 14.	Patient Search Page .....	50
Figure 15.	Manage Users' Accounts Page .....	52
Figure 16.	Case History Page .....	53
Figure 17.	Inspection Page .....	54
Figure 18.	Inquiry Page .....	55
Figure 19.	Auscultation and Olfaction Page .....	56
Figure 20.	Palpation and Pulse-Taking Page .....	57
Figure 21.	Selected Items Page .....	58

Figure 22. Fourteen Meridians Page .....	59
Figure 23. Lung Channel of Hand Tai-Yin .....	60
Figure 24. Prescription Page .....	61
Figure 25. Herb's Quantity Page .....	62
Figure 26. Diagnosis Page .....	63
Figure 27. Edit Current Case History Page .....	64
Figure 28. Print Current Case History Report Page .....	65
Figure 29. Print Prescription Page .....	66
Figure 30. View patient's Previous Case History Page .....	67
Figure 31. View Patient's Medical History Page .....	68
Figure 32. Patient' Information Page .....	69

CHAPTER ONE  
SOFTWARE REQUIREMENTS SPECIFICATION

1.1 Introduction

In small acupuncture offices, staff members are used to scheduling appointments and recording patient diagnoses by either writing or typing. Most of these records do not have any connections with each other. As the data becomes bigger, information will become more difficult to search and manage. For example, folders may be put in wrong places, causing assistants to spend more time looking for them. In addition, bad handwriting can cause mistakes. In order to help acupuncturists and assistants, the Chinese Medical Clinic System can maintain a schedule of appointments and provide a view of patient information, including histories of diagnoses and treatments. The staff and acupuncturist will be the users. The system will allow the staff to input patient personal information and their medical histories, schedule appointments, view or edit appointments, and view or modify patient information. The acupuncturist can view appointments, search patient records, record case histories, and manage user accounts. From the View Appointment page, the acupuncturist can connect with the patient's case history page, which allows

the doctor to select symptoms, Chinese herbs and acupuncture points' icons from a pop out window. The flexibility of functions will make the procedure much easier.

### 1.2 Purpose of the Project

The Chinese Medical Clinic System is designed to help acupuncturists and assistants record and store information. This system can maintain and schedule appointments, view patient diagnoses effectively. The system will be implemented in desktop PC to facilitate the acupuncturists record of information. The acupuncturist can select symptoms and acupuncture points simply by clicking buttons and checking checkboxes, and save the data into the database easily. The flexibility of functions will make the database entry a much easier task.

### 1.3 Context of the Problem

For the US acupuncture offices, the appointments, patients' history, and patients' information are recorded in English, while Chinese medicine terminologies are still vastly untranslated for English database. As a result, most of the Chinese medicine clinics in US still rely on hand writing to record information. Handwriting method has

several deficiencies. It is tedious, error prone, unreliable, and difficult to retrieve information.

#### 1.4 Significance of the Project

The Chinese Medical Clinic System provides an organized management for the processes of treatments in a small clinic. The system will handle not only patients' and employees' information, but also provide an easy way for doctors to record diagnoses and prescriptions.

The project Chinese Medical Clinic System provides some advantages as follows:

1. The acupuncturist can easily enter symptoms, diagnoses, and the Chinese herbs treatments into a computer organized database. The system provides a list of commonly used Chinese herbs, symptoms of various diseases, and the acupuncture points for treatments. By consulting the available information from the computerized database, the acupuncturist can accurately and promptly diagnose the symptoms. He can retrieve the previous notes and records of the patient and make the appropriate treatments.

2. The users' accounts can be added or deleted by the acupuncturist. The user name and password also can be changed.
3. An assistant can input patient's personal information into a form and save the data into the database effectively, besides searching patients' information in the database easily.
4. The Chinese Medical Clinic System is able to organize the schedule, view, edit, or modify appointments.

#### 1.5 Assumptions

Regarding to the project, I make the following assumptions.

1. The operating systems for the desktop computers are Windows XP Professional with IIS Web Server.
2. The knowledge of the database requires MS SQL Server.
3. The language for the project requires Visual Studio.NET, C# language, ASP.NET, HTML, and JavaScript.

#### 1.6 Limitations

The traditional Chinese medicament and acupuncture contain very broad field of knowledge. During the



development of the project, a number of limitations were noted. These limitations are presented here:

1. In the acupuncture points, some acupuncturist might focus on acupuncture point at ears or hands, which does not contain in the database of the project.
2. In the herbs' prescription, there are lots of special prescription of herbs, which has already been used very often. Each of these prescriptions has its special treatment. In the project, the prescription can only allow acupuncturists to input the herbs and quantities.

### 1.7 Definition of Terms

Table 1. Definition, Acronyms and Abbreviations

HTML	HyperText Markup Language. A language that describes the formatting of text inside a browser.
ADO.NET	ADO.NET does connect to the database to retrieve data, and connects again to update data when you have made changes. ADO.NET provides a disconnected subset of the data for your use while reading and displaying.
ASP.NET	ASP.NET is the name Microsoft has given to the combination of its two web development technology: Web Forms and web service.
JavaScript	A JavaScript script is a programming language that you can use to add interactivity to your web pages

Browser (Internet Explorer 5.0)	A program capable of retrieving HTML documents that include references to images and Java bit code and rendering it into a user-readable document.
GUI	Graphical User Interface. The graphical representation of physical or pseudo-physical objects (such as buttons, trees, and lists) that allow the user to direct the flow of the program through the use of a mouse or other pointing device.
MSDE Or MS SQL Server 2000	<p>1. Microsoft SQL Server Desktop Engine (MSDE), which is a specialized version of SQL server 2000. MSDE is entirely compatible with SQL Server, Which is truly an enterprise-class database server.</p> <p>2. SQL Server is the easiest database system to use: in addition to the well-known user interface, Microdoft offers several different tools to help you create database objects, tune your database application, and manage system administration's tasks.</p>
Visual C#.NET	Microsoft Visual C#.NET is a powerful but simple language aimed primarily at developers creating applications for the Microsft.NET platform. It inherits many of the best features of C++ and Microsoft Visual Basic but with some of the inconsistencies and anachronisms removed, resulting in a cleaner and more logical language. C# also contains a variety of useful features that accelerate application development, especially when used in conjunction with Microsoft Visual Studio.NET.
IIS 5.1	Internet Information Services (IIS) is designed to provide secure, scalable solution for creating and managing World Wide Web sites and Servers.
Visual Studio.NET	It is an integrated development environment (IDE) from Microsoft in which a programmer uses a graphical user interface (GUI) to choose and modify preselected sections of code written in the choice programming language.

Paint	The software can be used to create .bmp file for backgrounds ,Image Buttons and etc.
PhotoShop 7.0	Adobe Photoshop is one of most powerful visual communication tools ever appear in the desktop. The program has expanded the visual vocabulary of designers and illustrators to include color photo imagery, making photos the "raw material" for creative expression.

### 1.8 Organization of the Thesis

The thesis portion of the project was divided into six chapters. The chapter one contains software requirements specification, purpose of the project, the context of the problem, significance of the project, assumptions, limitation, definitions of terms and organization of the thesis. The Chapter two consists of the System Architecture and Design and Database Design: The unit test plan and system test plan will be described in the chapter three. The content of chapter four presents TCMCS graphical user interface design. The chapter five presents the maintenance, which includes Software installation, TCMCS Installation, and backup and restore. The Chapter six is the conclusion of the project.

CHAPTER TWO  
SOFTWARE DESIGN

2.1 Introduction

The software design includes the System architecture and design, class diagram, Database Design, Database Schema Conceptual Model - ER Diagram and Data Type and Details.

2.1 System Architecture and Design

The Chinese Medical Clinic System is designed through the concept of 3-tier client/server systems, which are more scalable, robust, and flexible (figure 1). It provides better security by not exposing the data schema to the client and by enabling authorization on the server. In addition, the client/server architecture is particularly well-suited for the LAN-based single server establishment. The Chinese Medical Clinic System consists of multiple clients talking to a local server, and is suitable for a multi-user environment. The foundation client server technologies of the Chinese Medical Clinic System are how to connect the web client and server, and how they play together. First, a web client/server interaction starts when the user specifies a target URL within the web browser. Second, the browser takes the URL

the user specified, embeds it inside of HTTP request, and then sends it to the target server. Third, on the receiving side, the HTTP server spins in a loop, waiting for requests to arrive on its well-known port. The server receives the client's message, accesses the database, generates an ASPX file with information from the database, ships it back to the client along with some status information, and then closes the connection.

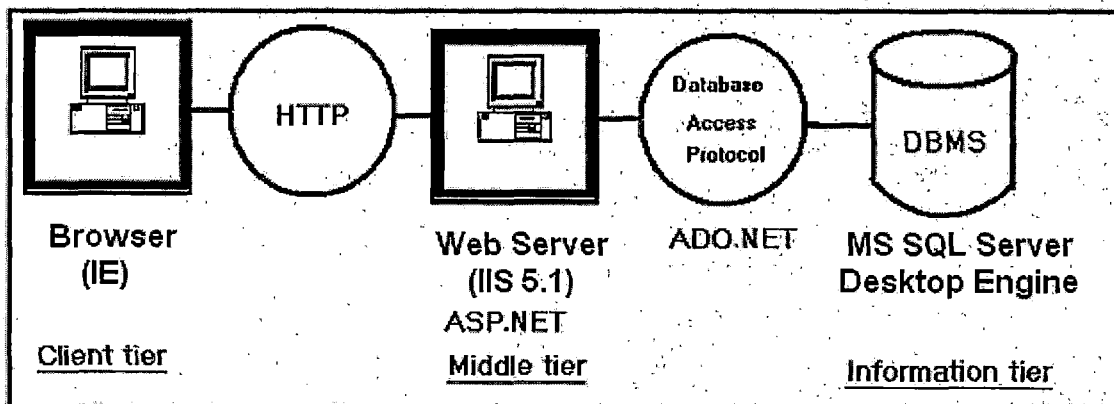


Figure 1. 3-Tier Client/Server

Web Interfaces will be used to access all system functionality.

1. Web Interface for Browsing presents the user with list boxes, check boxes, buttons and other controls that lead the user through the formulation of a request.

2. Most GUIs use a pointing device, such as a mouse, to pick certain parts of the displayed image button, image link or check box.
3. Users often have a small set of operations that they must perform repeatedly. In the system, this includes a list of symptoms, list of Chinese herbs, and a list acupuncture points. These lists will allow the acupuncturist or assistant to perform the sets of operations repeatedly.
4. The acupuncturist is also an administrator. The Manage Users' Accounts Interface links to the Manage Users' Accounts page, in which the administrator could create new users and granted account authorization, change user name or password, or update the user's information.

## 2.2.1 Class Diagram

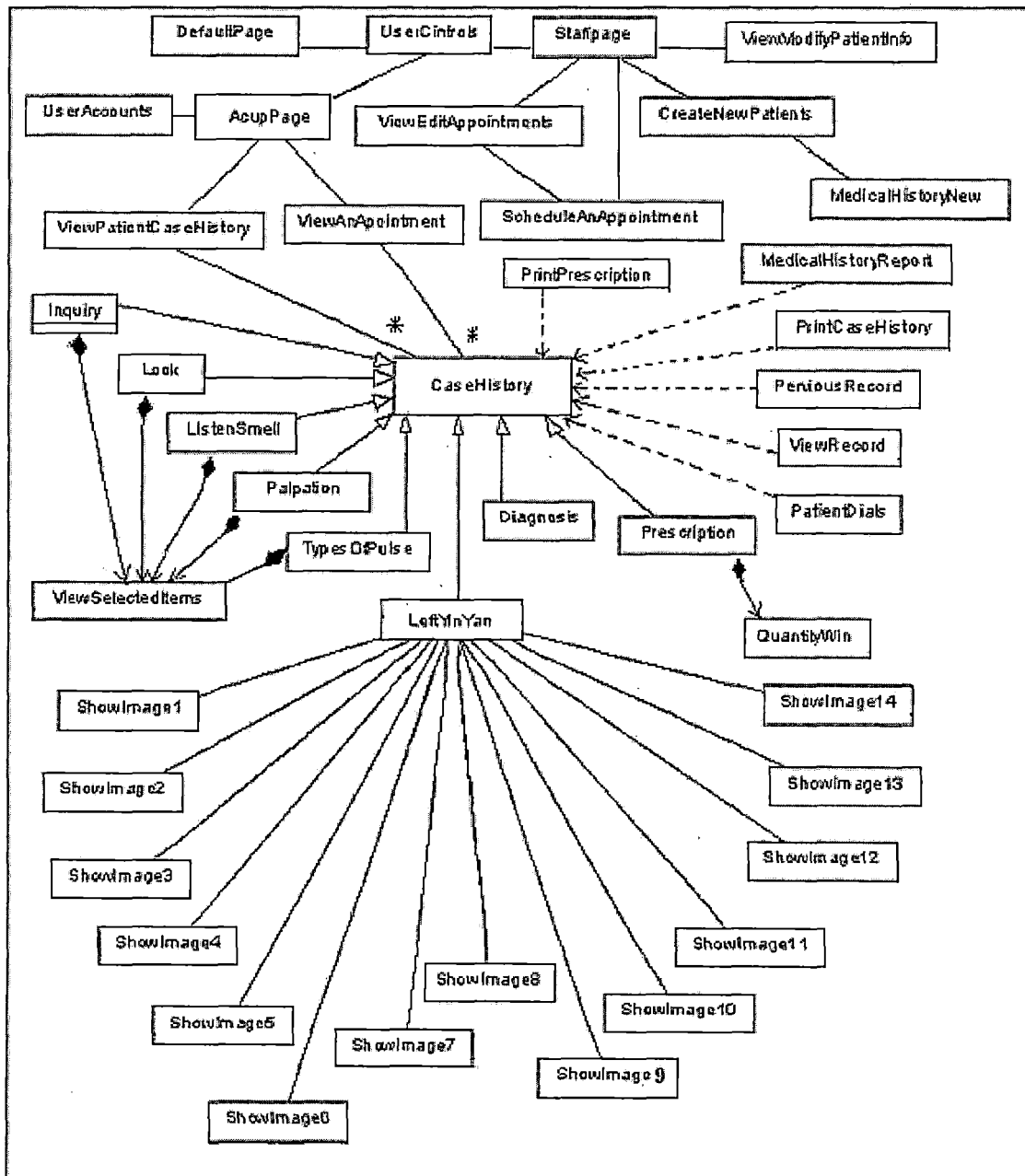


Figure 2. Class Diagram without Attributes and Operations

### 2.3 Database Design

Database in this system contains twelve tables that are bound together through the database relations. The Patients, MedicalHistory, CaseHistory and NewAppointments tables are built for recording patients' information, medical histories, case histories and appointment's schedules. The StaffAccount table records the staffs and acupuncturist's names, phones, user names and passwords. The other seven tables are DaignosisList1, HerbsList, inquiry, ListenSmell, Look, Palpation, and TypesOfPulse. These tables contain a symptom library, a diagnosis library, and a Chinese herbs library for reference.

The system is designed for two types of users: acupuncturist and staff.

First, acupuncturists are able to create user accounts for staff, view daily appointments, create new case histories, prepare for prescriptions, record diagnoses and track patients' information, case histories and medical histories.

The acupuncturist also plays the role of the administrator. In the Manage Users' account page, the administrator can add new user, update user information, and delete a user. The users' information is saved in the StaffAccount table.



According to the date, the acupuncturist can receive the daily schedule from the NewAppointments table in the View Appointment page.

The Case History page contains many GUIs, which links to the four diagnostic methods, a diagnosis list, the fourteen meridians of human, and the Chinese herb prescription. And also, it combines with several buttons, linking to the other pages, which can edit current record, track pervious records, print the report and the prescription, and connect with the medical history page, patient's personal information page, view appointment pages and search patient page.

When you click the Inspection, Listen and Smell, Inquiring, or Pulse-Taking Palpation interfaces, the screen will pop up a window, which contains each method's symptoms and suggestion of disorders from Look, ListenSmell, inquiry, Palpation, and TypesOfPulse tables separately.

The diagnosis interface in the Case History page links to the Diagnosis page, displaying a diagnostic list, which retrieve from the DiagnosisList1 table.

The fourteen channels (meridians) of human are displayed graphically in the separate pages. Each page of the channel displays acupuncture points, related to each

different meridian with checkboxes. The acupuncture diagnosis is recorded by checking the appropriate boxes associated with the acupuncture points. The recorded acupuncture diagnosis is included as a part of patient's case history.

The HerbsList table is displayed in the prescription page and it allows the user to select and add into prescription instead of typing.

The patient's Info button in the Case History page will connect with the Patient Information page, which will show the patient's personal information through retrieving the data from the Patient table.

The Medical History button in the Case History page will connect with the Medical History page, which will display the patient's medical history by retrieving from the MedicalHistory table.

The Edit Report button and Print Report button links to the current case history pages, which retrieve the data from CaseHistory table according to the Date and CaseNo (primary key).

The Print Prescription button in the Case History page links to the Prescription page, which list herbs' name and quantity from the CaseHiatory table according to the Date, and CaseNo (primary key).

The previous Records button in the Case History links to the Previous Case History page, which contain the patient's case histories from the CaseHistory table, and the user can track the data according to the Date.

Second, the system allows the staff to record and update patient information. After a patient complete the registration, the staff can log in the system and click "Create New Patients" interface to enter patient information and medical history. If staff needs to modify or view the patient information, staff can click "View/Modify patients' information" interface to view or modify the data. The staff can also schedule appointments for the acupuncturist. If a patient request to cancel or change the appointment, the staff could reschedule the appointment for him through clicking the View/Edit Appointments Link Button.

2.3.1 Database Schema Conceptual Model - ER Diagram

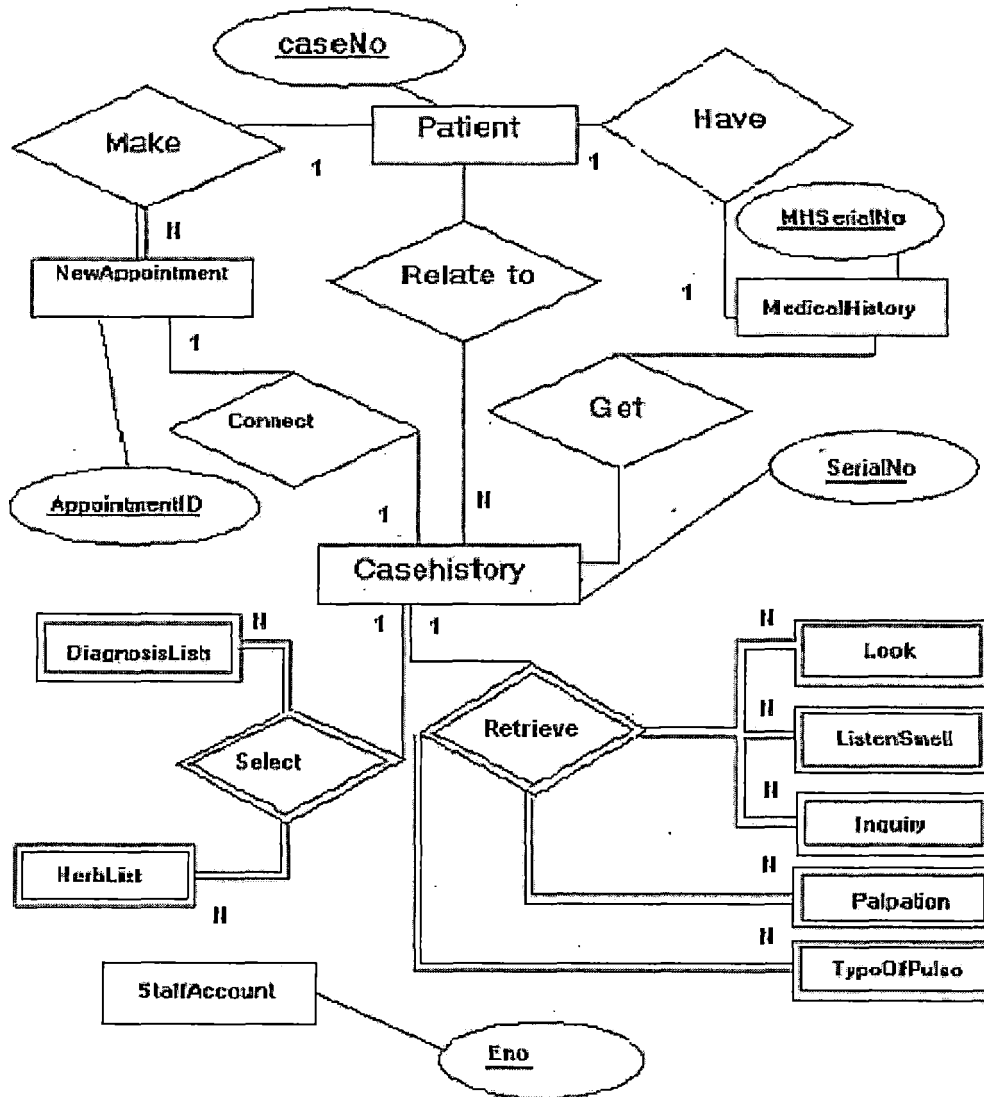


Figure 3. ER Diagram

2.3.2 Data Type and Details

The ClinicSystem1 database is designed for the Chinese Medical Clinic System. It contains twelve tables. The table designs are described as follows.

Table 2. Structure of Table CaseHistory

Field	Data Type	Length	Null	Key
SerialNo	int	4		Primary Key
CaseNo	int	4		
MainType	char	200	Yes	
SubType	char	500	Yes	
SuggestionOfDisorders	char	500	Yes	
Herbs	char	500	Yes	
Acupoint	char	500	Yes	
Diagnosis	char	500		
Date	nvarchar	20		

Table 3. Structure of Table DiagnosisList1

Field	Data Type	Length	Null	Key
Keyno	int	4		Primary Key
ICD-9	float	8		
Area	char	150		
Diagnosis	char	200		

Table 4. Structure of Table HerbsList

Field	Data Type	Length	Null	Key
SerialNo	int	4		Primary Key
Pinyin	char	50		
ChineseHerbs	char	50		
Strokes	int	4		
FirstChar	char	10		

Table 5. Structure of Table Inquiry

Field	Data Type	Length	Null	Key
InquiryNo	int	4		Primary Key
MainType	char	100	Yes	
SubType	char	300		
Symptoms	char	500	Yes	
SuggestionOfDisorders	Char	500	Yes	

Table 6. Structure of Table ListenSmell

Field	Data Type	Length	Null	Key
ListenSmellNo	int	4		Primary Key
mainType	char	100		
SubType	char	200		
Symptoms	char	500	Yes	
SuggestionOfDisorders	char	500	Yes	

Table 7. Structure of Table Look

Field	Data Type	Length	Null	Key
LookNo	int	4		Primary Key
MainType	char	100		
SubType	char	200		
Symptoms	char	500	Yes	
SuggestionOfDisorders	char	500	Yes	

Table 8. Structure of Table Palpation

Field	Data Type	Length	Null	Key
PNo	int	4		Primary Key
MainType	char	100		
SubType	char	100		
Symptoms	char	500		
SuggestionOfDisorders	char	500		

Table 9. Structure of Table TypesOfPulse

Field	Data Type	Length	Null	Key
PNo	int	4		Primary Key
MainType	char	50		
SubType	char	100		
Symptoms	char	500		
SuggestionOfDisorders	char	500		

Table 10. Structure of Table MedicalHistory

Field	Data Type	Length	Null	Key
MHSerialNo	int	4		Primary key
CaseNo	int	4		
RecordDate	nvarchar	10		
FirstName	char	50		
LastName	char	50		
MInit	char	2		
Sex	char	1		
SSN	nvarchar	11		
Q1	char	3		
Q1 5	char	500	Yes	
Q2	char	3		
Q2 5	char	800	Yes	
Q3	char	500	Yes	
Q3 5	char	500	Yes	
Q4	char	500	Yes	
Q5	char	3		
Q6	char	3	Yes	
Q7	char	3		
Q8	char	3		
Q8 5	cahr	500	Yes	

Table 11. Structure of Table NewAppointments

Field	Data Type	Length	Null	Key
AppointmentID	int	4		Primary key
CaseNum	int	4	yes	
FirstName	char	50		
LastName	char	10		
Phone	nvarchar	24		
AppDate	nvarchar	10		
AppTime	nvarchar	5		



Table 12. Structure of Table Patients

Field	Data Type	Length	Null	Key
CaseNo	int	4		Primary key
RegDate	Nvarchar	15		
LastName	Char	20		
Minit	Char	10	Yes	
FirstName	Char	20		
Birthday	nvarchar	15		
Sex	Char	10		
SSN	nvarchar	50		
DriverLicense	nvarchar	20	Yes	
MartialStatus	char	10	Yes	
Phone	nvarchar	50		
E mail	nvarchar	50	Yes	
Address	nvarchar	150	Yes	
Occupation	char	30	Yes	
BusinessPhone	nvarchar	50	Yes	
ContactPerson	char	50	Yes	
Relation	char	10	Yes	
EmergencyPhone	nvarchar	50		
InsuranceCo	char	20	Yes	
PolicyNo	nvarchar	50	Yes	
BillingAddress	nvarchar	200	Yes	
InsurancePhone	nvarchar	50	Yes	

Table 13. Structure of Table StaffAccounts

Field	Data Type	Length	Null	Key
Eno	int	4		Primary key
FirstName	char	20		
MInit	char	20	Yes	
LastName	char	20		
Phone	nvarchar	50		
UserName	nvarchar	50		
Password	nvarchar	50		

## CHAPTER THREE

### SOFTWARE QUALITY ASSURANCE

#### 3.1 Introduction

In order to perform effectively, the unit test plan and system test plan will play a very important role. The TCMCS has only two users and each of them have different works, so there is two unit test plans. The purpose of the system test is to provide a high degree of assurance. Both of the test plans can guarantee the system performance and reliability.

#### 3.2 Unit Test Plan

According to the project's functions, there are two major users and each user can perform their own functions. The unit Test plan includes the acupuncturist side test and Staff side's test.

Table 14. The Acupuncturist Side's Test Plan

Forms	Tests performed	Results
TCMCS Page	<ul style="list-style-type: none"><li>This page only contains a global variable, which might be used in the Chinese medical clinic system page.</li></ul>	Pass
Chinese Medical Clinic System page	<ul style="list-style-type: none"><li>When fill in the user name and password, and clicks the login button, the page will link to the Acupuncturist's page.</li><li>The Clear button can erase the User name and password in the TextBoxes.</li></ul>	Pass

Forms	Tests performed	Results
Acupuncturist's Page	<ul style="list-style-type: none"> <li>• The My Appointment button links to View Appointment page</li> <li>• The Patient Search &amp; Case History button in the Acupuncturist' page connects to the Search Patient page</li> <li>• The Manage Users' Account button in the Acupuncturist's page connects to the Manage Users' Accounts page</li> <li>• The Logout button links to the Chinese Medical Clinic System.</li> </ul>	Pass
View Appointments Page	<ul style="list-style-type: none"> <li>• After select a date from calendar, the TextBox displays the date. When the user clicks the search button, all appointments at that day are displayed in the DataGrid, which contains patients' first name, last name, phone number, and time.</li> <li>• When selects a date and there is no appointment at that day, the message displays "no appointment at that day."</li> <li>• When select a day, which is before or after today, the user cannot link to the Case History page. After clicking the button in front of the first name, the message display "You cannot create a case history for the future or past appointment."</li> <li>• When click the button in front of first name, the button link to the Case History page.</li> <li>• The Exit button links to the Acupuncturist's page.</li> </ul>	Pass
The Patient's Case History Page	<ul style="list-style-type: none"> <li>• The View Appointment page links to the Case History page, which will automatically display the patient's first name, last name, case number, social security number and gender. The patient's information is passed from the View Appointment page.</li> </ul> <p>Image Button:</p> <ul style="list-style-type: none"> <li>• When the user clicks the Inspection image button, the screen will pop out a window, which is Inspection page.</li> <li>• Clicking the Listen/Smell image button, the Auscultation and Olfaction page will be in a pop out window.</li> <li>• Clicking the Inquiring image button, the Inquiry page will appear in a pop out window.</li> <li>• The Pulse Taking and Palpation page will appear in a pop out window after the user clicks Pulse-Taking Palpation image button</li> </ul>	Pass

Forms	Tests performed	Results
	<ul style="list-style-type: none"> <li>• When the user clicks the Diagnosis image button, The Diagnoses page will be in a pop out window.</li> <li>• Clicking the Acupuncture image button, Fourteen Meridian page will be in a pop out window.</li> <li>• Click the Prescription image button the Prescription page will be in a pop out window</li> </ul> <p>Button:</p> <ul style="list-style-type: none"> <li>• The Edit Record button links to the Edit Current Case History in a jumping out window.</li> <li>• The print Report button links to the Print Current Case History page in a pop out window</li> <li>• The Print Prescription button links to the Print Prescription page in a pop out window.</li> <li>• The pervious Records button links to the View Patient's Pervious Case History page in a pop out window</li> <li>• The Medical History button links to the View Patients Medical History page in a pop out window</li> <li>• The Patient's Info button links to the Patient Information page in a pop out window.</li> <li>• The Appointment button links to the View Appointment page.</li> <li>• The Search Patient button links to the Patient Search page.</li> </ul>	
Inspection Page	<ul style="list-style-type: none"> <li>• The list box in the Inspection page displays all main types of Inspection. When select any one in the ListBox, the DataGrid displays the subtype of the Inspection, symptoms and suggestion of disorder. Each row has an image button in front of the row. Once the user clicks the button in the select column, the subtype, symptom, and suggestion of disorder can be added into the database in the CaseHistory table.</li> <li>• The View Selects button can links to the Selected Items page.</li> <li>• The Close button can close the window.</li> </ul>	Pass

Forms	Tests performed	Results
Auscultation and Olfaction Page	<ul style="list-style-type: none"> <li>• The list box in the Auscultation and Olfaction page displays all main types of Auscultation and Olfaction. When select any one in the list box, the DataGrid displays the subtype of the Auscultation and Olfaction, symptoms and suggestion of disorder. Each row has an image button. Once click the button in the select column, the subtype, symptom, and suggestion of disorder can be added into the database in the CaseHistory table.</li> <li>• The View Selects button can links to the Selected Items page.</li> <li>• The Close button can close the window.</li> </ul>	Pass
Palpation and Pulse-Taking Page	<ul style="list-style-type: none"> <li>• The list box in the Palpation and Pulse-Taking page displays all main types of. When the select any one in the list box, the DataGrid displays the subtype of the Palpation and Pulse-Taking, symptoms and suggestion of disorder. Each row has a button. Once click the image button in the selected row, the subtype, symptom, and suggestion of disorder can be added into the database in the CaseHistory table.</li> <li>• The View Selects button can link to the Selected Items page.</li> <li>• When select Palpation and click the View Selects button, the Palpation and Pulse-Taking page links to the Selected Items page which displays the type of Palpation and Suggestion of disorders.</li> <li>• When select Pulse-Taking and click the View Select button, the Palpation and Pulse-Taking page links to the Selected Items page which displays the type of Pulse-Taking and suggestion of disorders.</li> <li>• The Close button can close the window.</li> </ul>	Pass
Inquiry Page	<ul style="list-style-type: none"> <li>• The list box in the Inquiry page displays all main types of Inquiry. When the user selects any one in the list box, the DataGrid will display the subtype of the Inquiry, symptoms and suggestion of disorder. Each row has a button. Once click the image button in the row, the subtype, symptom, and suggestion of disorder can be added into the database in the CaseHistory table.</li> <li>• The View Selects button can link to the Selected Items page.</li> <li>• The Close button can close the window.</li> </ul>	Pass

Forms	Tests performed	Results
Selected Items Page	<ul style="list-style-type: none"> <li>• The selected Item page displays date and case number under the title.</li> <li>• The delete function in the each line. The user can click it to delete the item.</li> <li>• The Back button links to the previous page which is Palpation and pulse-Taking, Inspection, Auscultation and Olfaction, or Inquiry page.</li> </ul>	Pass
Diagnoses Page	<ul style="list-style-type: none"> <li>• The Diagnoses page is a jump out window.</li> <li>• The list box contains all diagnoses</li> <li>• When select any diagnosis from the ListBox in the left side and click the Add button, the right side of the text box will display the selected diagnosis.</li> <li>• When clicks the Save button and the right side of the text box contains at least one item, the diagnoses will be saved in the CaseHistory table.</li> <li>• While click the Save button and the right side of the text box contain nothing, the message will display "The multiline contains nothing. Please add diagnosis into multiline."</li> <li>• The Close button can close the window.</li> </ul>	Pass
Prescription Page	<ul style="list-style-type: none"> <li>• The Prescription page is in a pop out window.</li> <li>• While input first alphabet in the text box and click the search button, the left side of ListBox displays an Herbs list.</li> <li>• When input stroke of the first letter, a Chinese herbs' the list displays in the left side of the list box.</li> <li>• While Select any herb item and click the Add button, the right side of the text box will display the selected item. When Click the Quantity button, a small window pop out. The window allows input the quality for each herb.</li> <li>• After Select an herb item and click the save button, the selected herbs and their quantities will be saved into CaseHistory table.</li> <li>• The Close button can close the window.</li> </ul>	Pass

Forms	Tests performed	Results
Edit Current Case History	<ul style="list-style-type: none"> <li>• After Click Edit Record in the Case History page, the Edit Current Case History page is in the pop out window.</li> <li>• The patient's information in the Edit current Case History page contains first name, last name, gender, birthday and today's date, which are transferred from the Case History page.</li> <li>• In the Edit Current Case History page, there are Seven DataGrids, which are Inquiry, Inspection, Auscultation and Olfaction, Pulse-Talking and Palpation, Acupunturist-Therapy, and Diagnosis</li> <li>• Each DataGrid contains many items, which selected by the user.</li> <li>• The Edit link button allows changing the content of each text box.</li> <li>• The Delete link button allows deleting the row.</li> <li>• The Close button can close the window.</li> </ul>	Pass
Print Current Case History	<ul style="list-style-type: none"> <li>• The Print Current Case History page is in a jump out window.</li> <li>• The page includes record date, first name, last name, birthday, middle initial, sex, case number, diagnosis, Acupuncture points, Prescription, and four DataGrids, which are Inquiry, Inspection, Auscultation and Olfaction ,and Pulse-Taking and Palpation. Each DataGrid contains two columns such as type and suggestion of disorders.</li> <li>• After Click the Print button, the button will display"Prepared by Dr. XXX," the printer will print the Case history, and the window will be closed automatically.</li> </ul>	Pass
Print Prescription	<ul style="list-style-type: none"> <li>• In the pop out window, the Prescription page contains date, first name, last name middle initial, birthday, gender, case number and prescription.</li> <li>• After Click the Print button, the button will display"Prepared by Dr. XXX," the printer will print the Case history, and the window will be closed automatically.</li> </ul>	Pass

Forms	Tests performed	Results
Herbs' Quantity	<ul style="list-style-type: none"> <li>• In the pop out window, there are ten digits', gram, 錢 , rest, sent, and Close buttons and a TextBox. After clicking the digit's and 'g' or '錢' button, the text box will display the selected data. The reset button will clear the TextBox. The send button will sent the content of TextBox into the right side of the text box in the Prescription page.</li> </ul>	Pass
View Patient's Previous Case History	<ul style="list-style-type: none"> <li>• In the pop out window, the View Patient's Previous Case History page contains date, case number, first Name, last Name, middle initial, Sex, and birth date, which transfers from the Case History page.</li> <li>• The Previous Case History contains seven DataGrids such as Inquiry, Inspection, Auscultation and Olfaction, Acupuncture-therapy, Prescription(Herbs), and Diagnosis.</li> <li>• There is DrawDownList in the page. The draw down list contains previous visit dates. When selects a day from the draw down list, the seven DataGrids will display the record of treatment.</li> <li>• A Close button can close the window.</li> </ul>	Pass
View Patient's Medical History	<ul style="list-style-type: none"> <li>• The View Patient's Medical History page is in a pop out window.</li> <li>• The page contains patient medical history, which is recorded in his/her first visit.</li> <li>• There is a print button in the button of the page. When click the button, the printer will print the medical history and close the window.</li> <li>• The window will be closed automatically after 25 second.</li> </ul>	Pass
Patient's Information	<ul style="list-style-type: none"> <li>• The Patient's Information page in the pop out window contains patient's information.</li> <li>• The print will print the patient's information and close the window.</li> <li>• The window will be closed automatically after 20000 ns.</li> </ul>	Pass



Forms	Tests performed	Results
Parent Search Page	<ul style="list-style-type: none"> <li>• While input the patient's phone in the text box and click search button, the DataGrid will display all patients who have the same phone number.</li> <li>• When input the last name in the text box and click the Search button, the DataGrid displays all patients who have the same last name.</li> <li>• When input the phone and last name, the DataGrid will display all patients who have the same phone and last name.</li> <li>• The Clear button will clean two text boxes.</li> <li>• The Exit Button allows the user to go back to the Acupuncturist's Page.</li> </ul>	Pass
Fourteen Meridians Page	<ul style="list-style-type: none"> <li>• The Fourteen Meridian page is in a pop out window.</li> <li>• There is a draw down list, which contains fourteen Meridians. After the user selects any item from the draw down list, the Fourteen Meridian page will be changed to the one of fourteen meridians.</li> <li>• The Close button can close the window.</li> </ul>	pass
The Lung Channel of Hand-Tai yin Page	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Lung Channel of Hand-Tai yin Page.</li> <li>• When select the check boxes in the page and click the Save button to save them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After the selected check boxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Large Intestine Channel of hand-Yang Ming	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Large Intestine Channel of hand-Yang Ming.</li> <li>• When select the CheckBoxes in the page and click the Save button, the selected items will save them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After the selected CheckBoxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass

Forms	Tests performed	Results
The stomach Channel of Foot-Yang Ming	<ul style="list-style-type: none"> <li>• The page contains DrawDownList, which can change the page to the stomach Channel of Foot-Yang Ming Page.</li> <li>• When select the check boxes in the page and click the Save button, the selected items will be saved them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After selected items are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	pass
The Spleen Channel of Foot-Tai yin	<ul style="list-style-type: none"> <li>• The Page contains DrawDownList, which can change the page to the Spleen Channel of Foot-Tai yin Page.</li> <li>• When select the CheckBoxes in the page and click the Save button, the selected items will save them into the CaseHistory table.</li> <li>• The Close button will close the window.</li> <li>• After the selected CheckBoxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Heart Channel of Hand-Shao yin jing	<ul style="list-style-type: none"> <li>• The Page contains DrawDownList, which can change the page to the Heart Channel of Hand-Shao yin jing Page.</li> <li>• When select the CheckBoxes in the page and click the Save button, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button will close the window.</li> <li>• After selected items are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Smell Intestine Channel of Hand-tai yang	<ul style="list-style-type: none"> <li>• The Page contains DrawDownList, which can change the page to the Smell Intestine Channel of Hand-tai yang Page.</li> <li>• When select the CheckBoxes in the page and click the Save button, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button will close the window.</li> <li>• After selected items are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass

Forms	Tests performed	Results
The Bladder Channel of Foot-Tai yang page	<ul style="list-style-type: none"> <li>• The Page contains DrawDownList, which can change the page to the Bladder Channel of Foot-Tai yang Page.</li> <li>• After select the check boxes in the page and click the Save button in the Auscultation and Olfaction page, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After selected items are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Kidney Channel of Foot-Jue Yin page	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Kidney Channel of Foot-Jue Yin Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button will close the window.</li> <li>• After selected items are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The San Jiao (Triple Warmer) Channel of Hand-Shao yang	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the San Jiao (Triple Warmer) Channel of Hand-Shao yang Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After the selected CheckBoxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The San Jiao (Triple Warmer) Channel of Hand-Shao yang	<ul style="list-style-type: none"> <li>• The Page contains DrawDownList, which can change the page to the San Jiao (Triple Warmer) Channel of Hand-Shao yang Page.</li> <li>• After select the CheckBoxes in the page and click the Save button, the selected items will be saved them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• If the selected check boxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass

Forms	Tests performed	Results
The Gallblader Channel of Foot-Shao-yang	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Gallblader Channel of Foot-Shao-yang Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will be saved into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After selected check boxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Liver Channel of Foot-Jue yin	<ul style="list-style-type: none"> <li>• The page contains draw down list, which can change the page to the Liver Channel of Foot-Jue yin Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will be save them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• If the selected check boxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Du Channel	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Du Channel Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will be saved them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• After selected CheckBoxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass
The Ren Channel	<ul style="list-style-type: none"> <li>• The Page contains draw down list, which can change the page to the Ren Channel Page.</li> <li>• After select the check boxes in the page and click the Save button, the selected items will save them into the CaseHistory table.</li> <li>• The Close button can close the window.</li> <li>• If the selected check boxes are saved successfully, the message will display "The Selected items are saved successfully!"</li> </ul>	Pass

Forms	Tests performed	Results
Manage Users' Accounts	<ul style="list-style-type: none"> <li>• In the Manage Users' accounts, the list box contains employees' first name such as staff and acupuncturists.</li> <li>• Selecting the first name from the list box, the right side of TextBoxes will display first name, Middle Initial, last name, Employee number, phone number, user name, and password.</li> <li>• While all TextBoxes in the right side are empty, the user can input name staff information. After fill in all text boxes, and click the Add New button the data will be saved into the StaffsAccount table and the table will generate a Primary key, which is employee number (ENo). The Employee number will display in the right side of the ENo label.</li> <li>• When the text boxes in the right side are contains data, and the user clicks the Add New button, the message will display "please fill the data in the TextBoxes."</li> <li>• When the text boxes have already contained exiting staff's information and the user clicks the Add new button, the message will display "Wrong button"</li> <li>• After modify existing staff's information in the text boxes, and click the Add New button, the message will display "Wrong button."</li> <li>• After modify existing staff's information and click the Update button, the modified data will save into the database.</li> <li>• After select a first name in the list box, and click the Delete button, the staff will delete from the StaffsAccounts.</li> <li>• The clear button can erase the content of the text boxes.</li> <li>• The Exit button is used to go back to the Acupuncturist's page.</li> </ul>	Pass

Table 15. Staff Side Test Plan

Forms	Tests performed	Result
TCMCS Page	<ul style="list-style-type: none"> <li>• It is the first page in the Chinese Medical Clinic System. It will automatically links to the Chinese Medical Clinic System page. The user will not see the page except the user has failed to login for five times.</li> </ul>	Pass
Chinese Medical Clinic System page	<ul style="list-style-type: none"> <li>• After Failing to login for five times, the Chinese Medical Clinic System page will jump to the TCMCS page and cannot go back to the Chinese Medical Clinic System page at that day. The user can only try again next day.</li> <li>• When filling in the user name and password, and clicking the login button, the page will link to the Acupuncturist's page.</li> <li>• Test the Clear button function</li> </ul>	pass
Staff's page	<ul style="list-style-type: none"> <li>• After Clicking Schedule An Appointment button, Staff's page will link to the Schedule An Appointment page</li> <li>• The View/Modify Patients' Information button will connect to the View or Modify patient's Information page</li> <li>• The View/Edit Appointments button links to the View or Edit Appointments page</li> <li>• The Create New Patients button connects to the Create New Patients page</li> </ul>	pass
Create A New Patient page	<ul style="list-style-type: none"> <li>• The Clear button will clear all columns.</li> <li>• The Save button will save all information, which user had input</li> <li>• While some important columns are empty, the warning and the error message will show up and the data cannot be saved until these columns are filled in the data.</li> <li>• The Exit button links to the Staff's page</li> </ul>	Pass

Forms	Tests performed	Result
Schedule An Appointment page	<ul style="list-style-type: none"> <li>• Click the date from calendar and the date will appear in the date column.</li> <li>• The draw down list will display the time to be selected.</li> <li>• The Save button can save the Appointment if the date and time are not taken.</li> <li>• The message label will display different messages, while the data and time is not available, any column is empty, the patient's name and phone cannot found in the Patient table. The data will not be saved until all problems above are solved.</li> <li>• Check the Clear button function</li> <li>• The View/Edit button links to the View or Edit Appointments</li> <li>• The Exit button links to the Staff's page.</li> </ul>	pass
View or Edit Appointment page	<ul style="list-style-type: none"> <li>• Select a date from calendar and the selected date will appear in the date column.</li> <li>• When only select a date and click the search, the DataGrid will display the all patients, who have appointments in that date. If the patient selects a date and inputs a patient's phone, the DataGrid will display only the patient, who has the phone and the appointment at the selected date.</li> <li>• The Edit link button or Delete link button are allowed to edit, the date and time or delete the appointment. While select the date and time, which are taken, the message display a message- the appointment is taken.</li> <li>• The clear button is to erase the content of the TextBoxes.</li> <li>• The Appointment button links to the Schedule An Appointment page</li> <li>• The Exit button links to the Staff's page</li> </ul>	pass
View or Modify Patient's Information page	<ul style="list-style-type: none"> <li>• After input the patient's last name, the list box will list all patients' first name. While the select a first name, all TextBoxes will display patients' information.</li> <li>• To make some changes in the text boxes expect the last name and first name. After making any change and clicking the update Button, the changed data will be saved in the Patient table.</li> <li>• Exit button links to the Staff's page</li> </ul>	pass

Forms	Tests performed	Result
TCMCS Page	<ul style="list-style-type: none"> <li>This page only contains a global variable, which might be used in the Chinese medical clinic system page.</li> </ul>	pass

### 3.3 System Test Plan

The system should be tested, before using the TCMCS. System testing of TCMCS begins with the following steps:

Table 16. System Test Results

System Testing	Results
1. Install TCMCS into server.	Pass
2. Start up all servers such as InternetInformation server, ASP.NET Server and The Microsoft SQL Server Desktop Engine (A Small SQL Server). And also grand the privilege to the user for using ASP.NET.	Pass
3. Running and testing by using real data on all forms.	Pass



## CHAPTER FOUR

### USERS MANUAL

#### 4.1 Introduction

The Traditional Chinese Medical Clinic System is designed for two major users such as staff and Acupuncturists. Each of them will perform different functions. Staff creates appointment tables and, input patient information. Acupuncturists can view the Appointment table and link to the each patient's case history. From the patient's case history, the acupuncturist can view the patient's information and medical history.

The following figure is the Use Case Diagram of the Traditional Chinese Medical Clinic System.

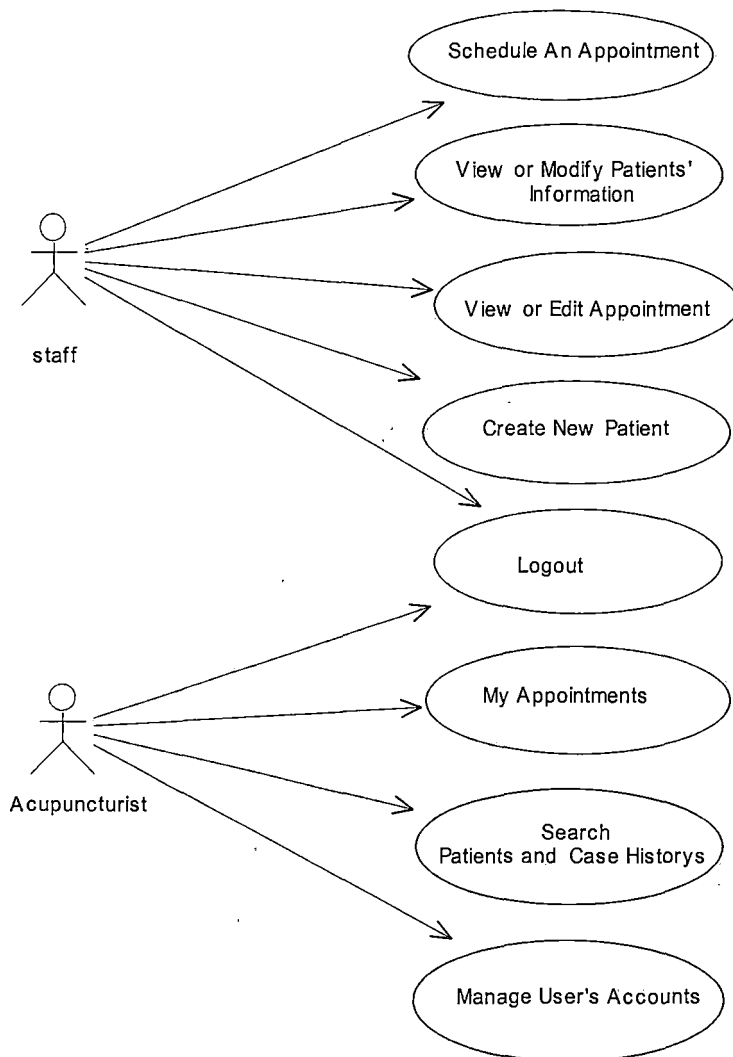


Figure 4. Traditional Chinese Medical Clinic System Use Case Diagram

#### 4.2 Traditional Chinese Medical Clinic System Graphical User Interface Design

The GUI is written with ASP.NET and Java Script. The TCMCS is designed through creating Microsoft ASP.NET pages, and building application that can be accessed from

a Web browser. TCMCS use forms-based authentication to secure access to a web application. The web application is built by many web forms that present, update, and delete data from the ClinicSystem1 database using DataGrid control. These web forms can also save data into the ClinicSystem1 database. The Web Forms application has the ability to navigate from one form to another by clicking an interface or button. In addition to moving between forms, the application can pass information between forms.

#### 4.2.1 The Traditional Chinese Medical Clinic System and Traditional Chinese Medical Clinic System Page

The Traditional Chinese Medical Clinic System page is a login page created for two major users such as staff and acupuncturist. After the user inputs the user name and password, the data will compare the user name and password in the StaffAccounts table. If the user name and password does not match with the user name and password in the StaffAccounts table, the error message displays "Login attempt failed, invalid username or password." If the user have failed to login for five times, the Traditional Chinese Medical Clinic System page will jump to the Home page. The Traditional Chinese Medical Clinic System page set a variable equal to today's day and transfer the variable to the Home page. When the user enters the Home

page, the variable will compare today's date. If the variable is equal to today's date, the Home page cannot automatically link to the Traditional Chinese Medical Clinic System.



Figure 5. Traditional Chinese Medical Clinic System Page

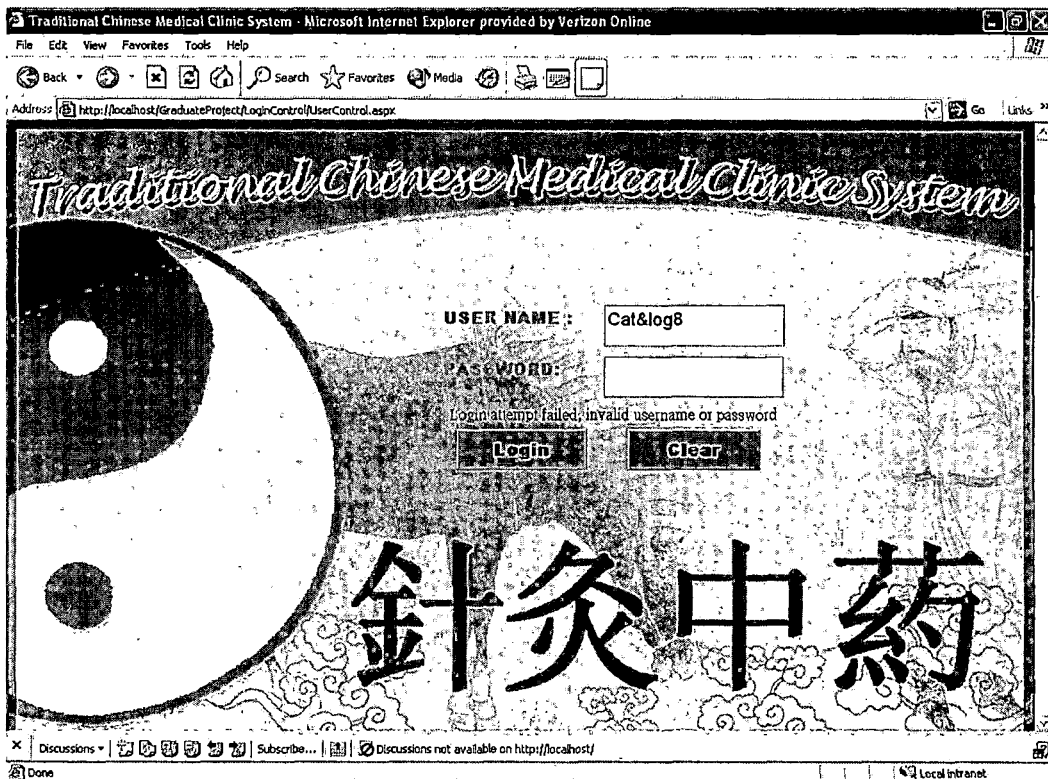


Figure 6. Chinese Medical Clinic System Page with Login Error Message

#### 4.2.2 Staff's Page

After the user input the user name and password, and they match with the user name and password in the StaffAccounts, the Chinese Medical Clinic System will link to the staff's page or Acupuncturist's page depending on the user type. If the user type is staff, the page will be Staff's page, which can link to five different pages and perform different jobs.

1. The Schedule an Appointment link button will connect to The Schedule an Appointment page.

2. The View/Modify Patient's Information LinkButton will Link to the View or Modify Patient's Information.
3. The View/Edit Appointments LinkButton will jump to the View or Edit Appointments page.
4. The Create new patient LinkButton will link to the Create new patients.
5. The logout LinkButton will go back the traditional Chinese Medical Clinic System.

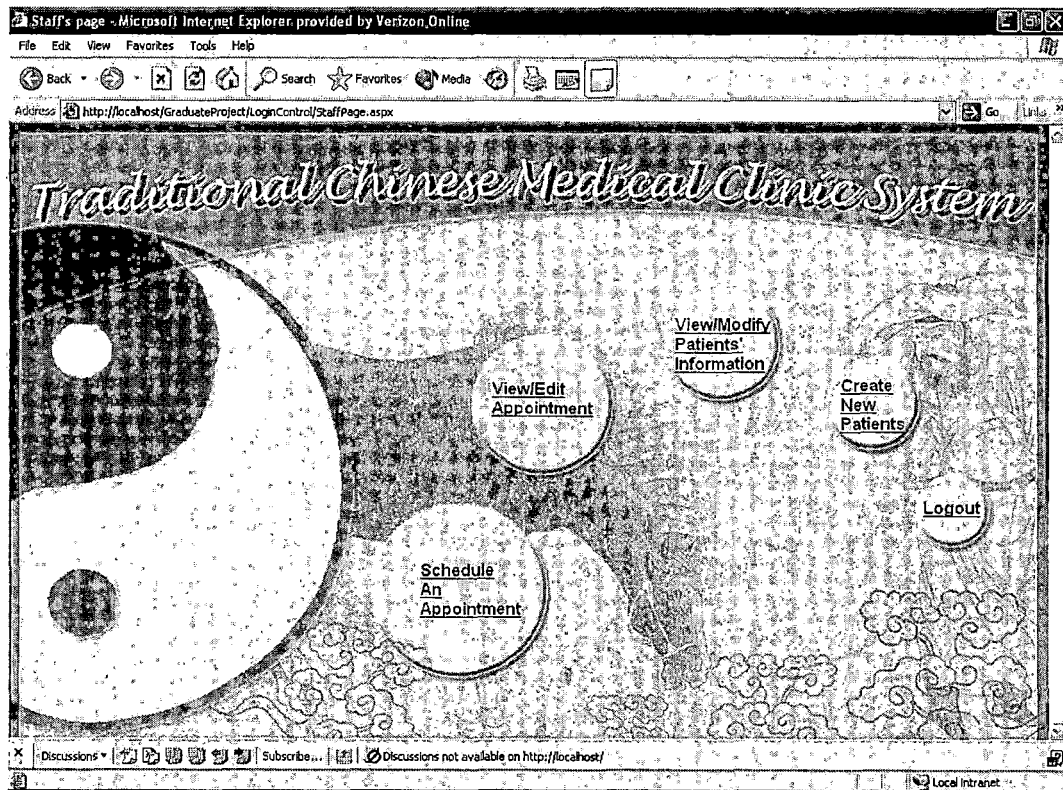


Figure 7. The Staff's Page

### 4.2.3 Schedule an Appointment Page

The staff can click the date from calendar and the date will appear in the date column. The list box will display the time to be selected. After the user selected the date and time and filled in the last name, first name, date ,and phone number, the user clicks the Save button, which will save the date if the appointment is not taken. If the appointment is taken, the message will tell the user the data and time is not available. If any column is empty, the message will tell the user to input date into the column. If the patient's name and phone cannot found in the Patient table, the message will tell the user the patient information cannot be found in the database. The data will not be saved until all problems above are resolved. The clear button will clear all columns. The View/Edit button can link to the View or Edit Appointment page. The Exit button will go back to the Staff's page.

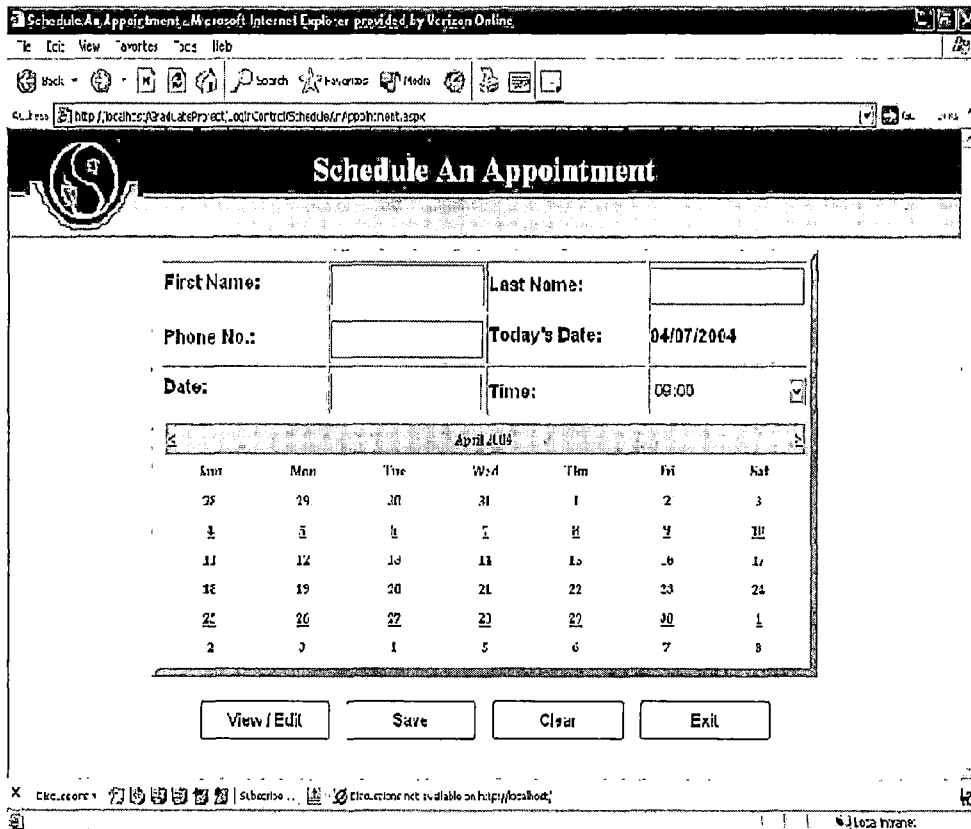


Figure 8. Schedule an Appointment Page

#### 4.2.4 View or Edit Appointments Page

When the user selects a date from the calendar and clicks the Search button, the DataGrid will display the all appointments in the selected date. If the user selects a date from the calendar and input phone number, and then click save button, the search will narrow down the search. The user is allowed to delete or edit the any appointment in the DataGrid. When the user clicks Edit link button, the date and time columns become text boxes, where the



user can make any change in the text boxes or cancel the job.

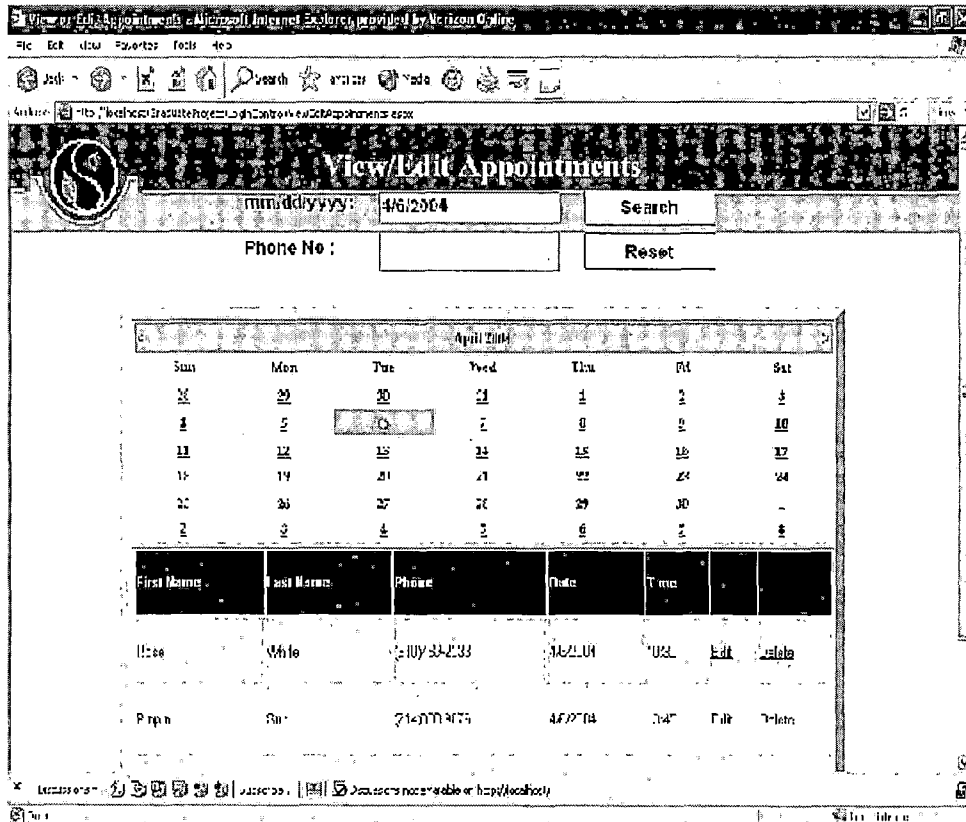


Figure 9. View or Edit Appointments Page

#### 4.2.5 Create New Patient

The Create New Patient page allows the user to fill in patient's information. If the major information has not filled in, the red star and a warning message will pop up.

The clear button is used to clean the text boxes. The Exit button will go back to the Staff's page.

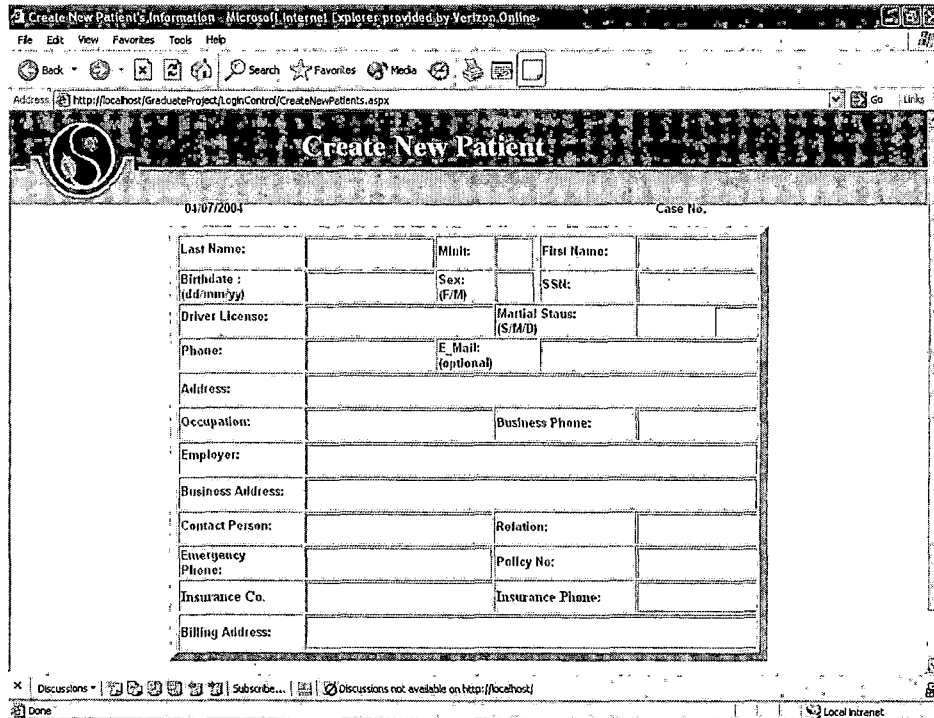


Figure 10. Create New Patients Page

#### 4.2.6 View or Modify Patient's Information

When the user inputs the last name and click the Search button, the patient's first name will display in the ListBox. If the user selects any first name in the list box, the patient's personal information will display in the right side of the columns. The SSN, Phone number and business phone number has their format, if the format is incorrect, the data cannot be save. Error will display under the title. Last name and first name cannot be changed. If the user tries to modify the first name or last name, the error message will display under the title.



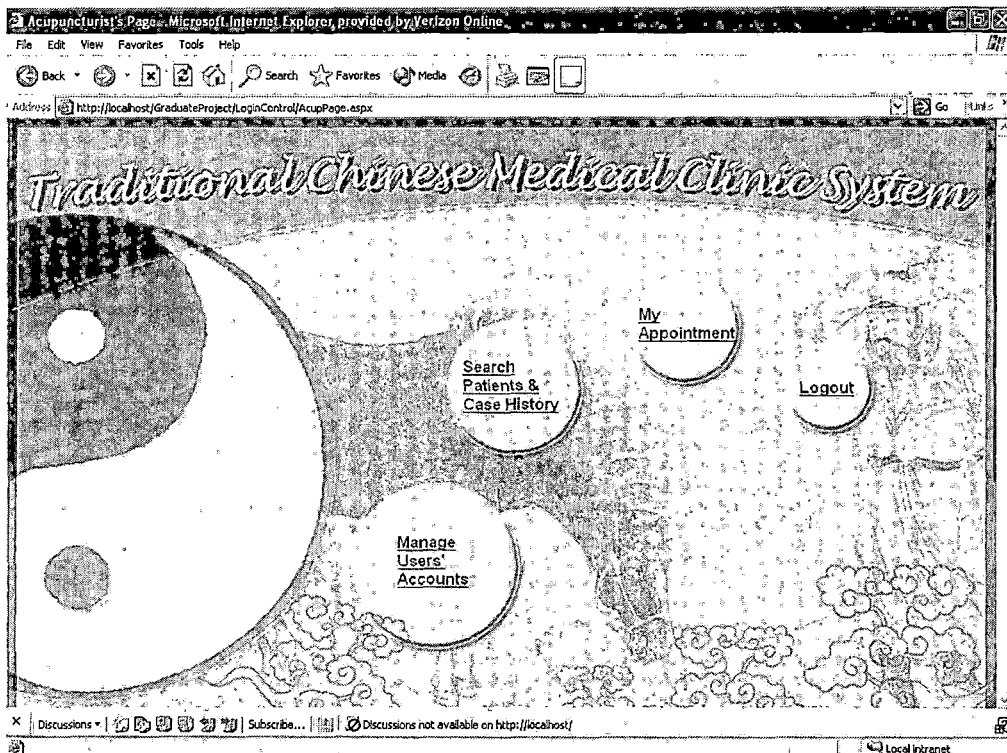


Figure 12. The Acupuncturist's Page

#### 4.2.8 View Appointment Page

When the user clicks the View My Appointments in the Acupuncturist page, the page connects to the View Appointment. While the acupuncturist selects a date in the calendar, the text box will display the date. After the user clicks the search button and search the date in the NewAppointments table, the DataGrid will display all appointments in that date. The DataGrid contains sex row in each page. The user can click the number at the low right corner in order to enter the page. The ImageButton in each row will links to the CaseHistory page and

transfer the patient's case number. The Exit button can link to the Acupuncturist's page.

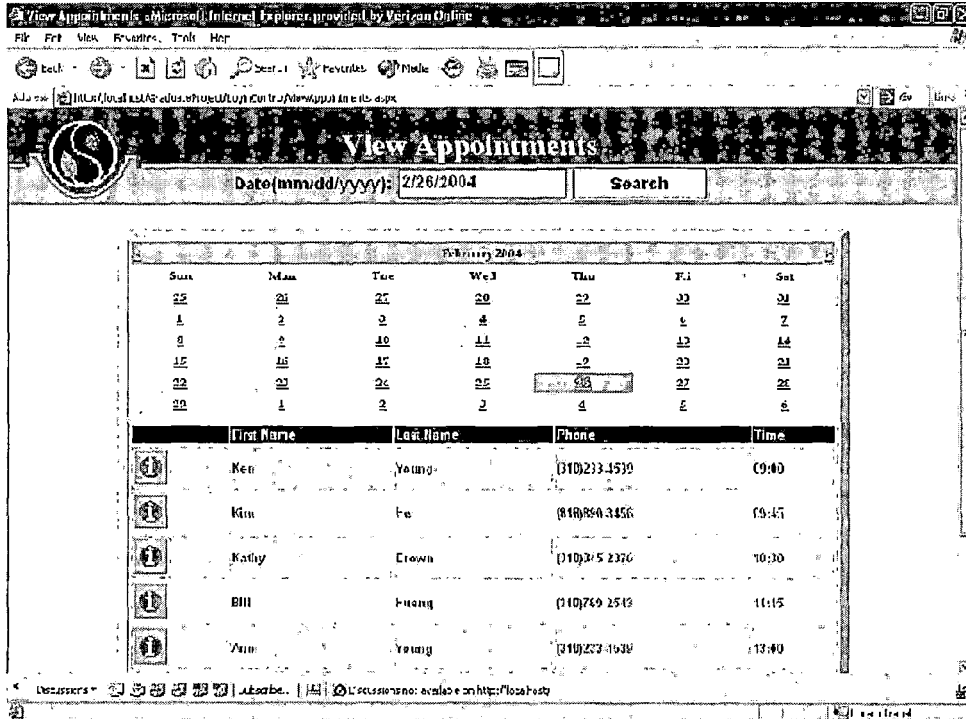


Figure 13. View Appointment's Page

#### 4.2.9 Patient Search Page

After clicks the Search Patient & Case History button in the Acupuncturist's page, the page connects to the Patient Search page. If the user inputs the Phone only and clicks the Search button, the DataGrid will display the patients who have the phone number. If the user inputs the last name only and click the Search button, the DataGrid will display all patients who have the same last name. If the user fill in phone number and last name and clicks the

Search button, the DataGrid will display all patients who match the phone number and last name. The Exit button allows the user back to the Acupuncturist's page.

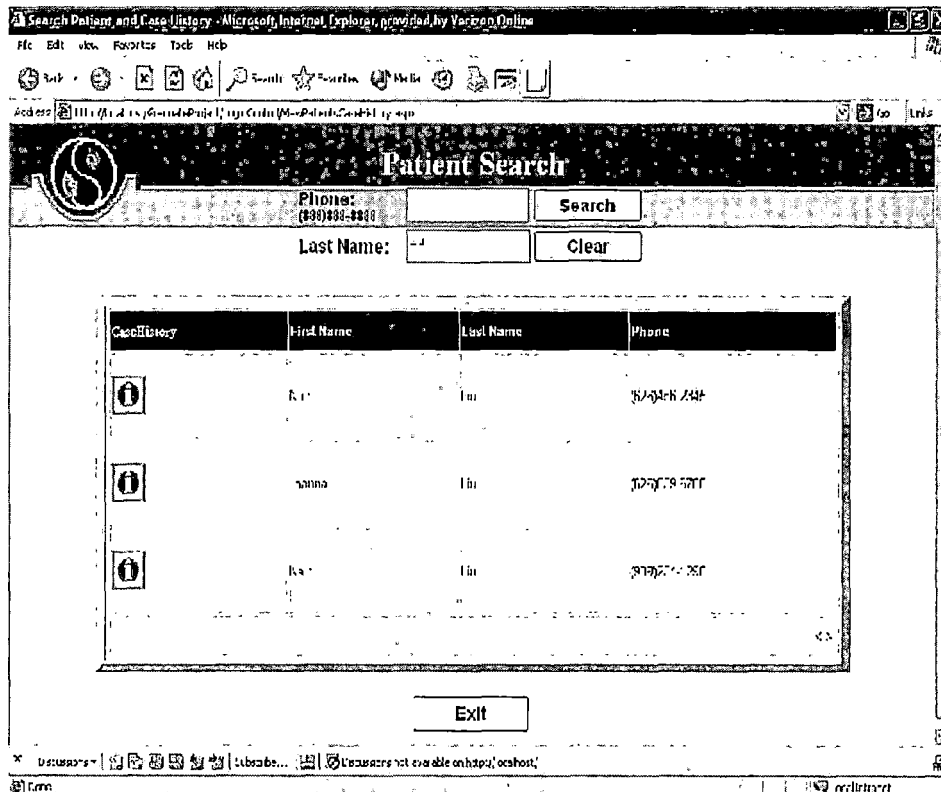


Figure 14. Patient Search Page

#### 4.2.10 Manage User's Accounts Page

The ListBox in the left side contains the acupuncturist's and staffs' first name. When the administrator selects a name from the list box, the personal information will display in the text boxes. The administrator can only modify the user's phone number, user name, and password. The clear button will clean all

text boxes. If the administrator fills in the staff's name, phone, user name and password, and clicks the Add New button, the staff's first will appear in the ListBox. The Delete button will delete the staff from the StaffAccounts table in the database. If the text boxes contain staff's data, and the administrator click Add New button, the message will display "Wrong button." After the administrator updates or adds new staff in the StaffAccounts table, the message will display "you have updated successfully" or "You have added a new staff successfully." If the text boxes are empty, and click the Add New button, Update button, or Delete button, the error message will appear. The Exit button allow the user go back to the Acupuncturist's page.





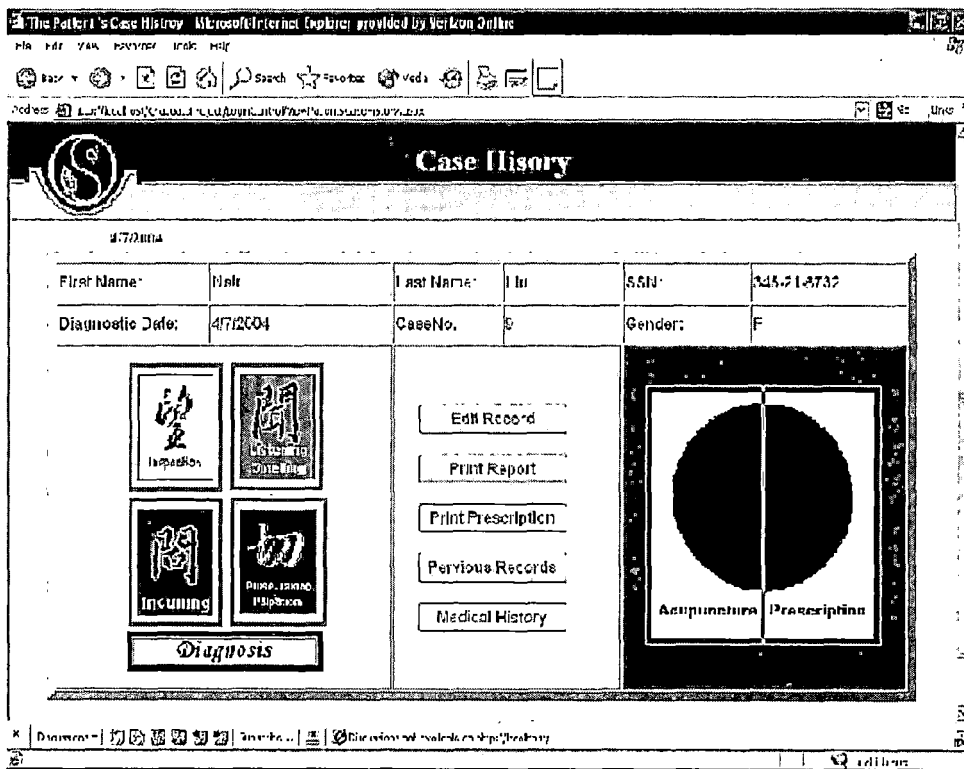


Figure 16. Case History Page

#### 4.2.12 Inspection Page

After the acupuncturist inspect the different areas of the patient's body or spirit, he can select the main types in the ListBox. After selects the main type from the main types, the DataGridView will display the subtype, symptoms and suggestion of disorders. The DataGridView contains four rows in the each page. The page number is displayed in the bottom of the DataGridView. Each row has an ImageButton in front of each row. While click the ImageButton of the row, the subtype and suggestion of disorders will be recorded in to the patient's case

history. The View Selected Item button allow the user to take a look these selected items. The Close button is used to close the window.

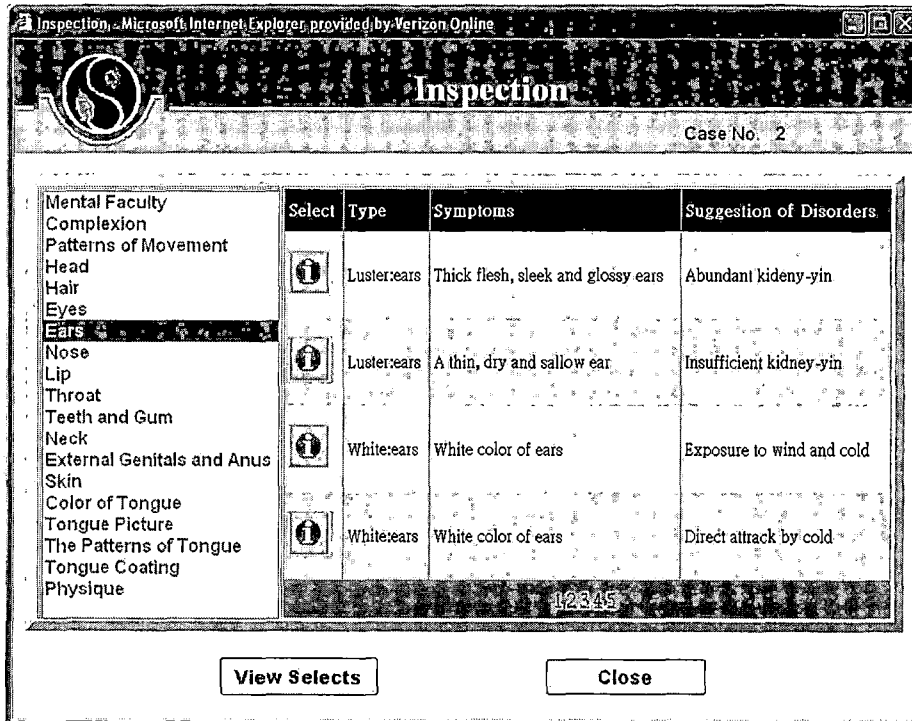


Figure 17. Inspection Page

#### 4.2.13 Inquiry Page

The inquiry page contains many main types of inquiry in the list box. Actually, the functions are the same as the Inspection.

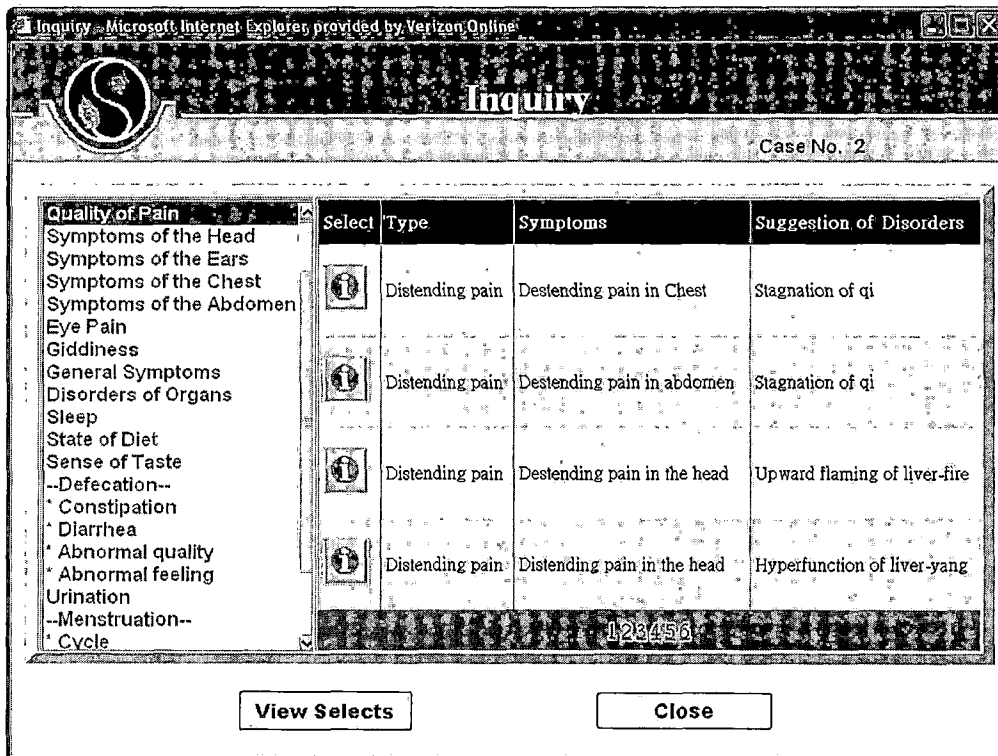


Figure 18. Inquiry Page

#### 4.2.14 Auscultation and Olfaction Page

The Auscultation and Olfaction page contains many main types of Auscultation and Olfaction page in the list box. Actually, the functions are as same as the Inspection.

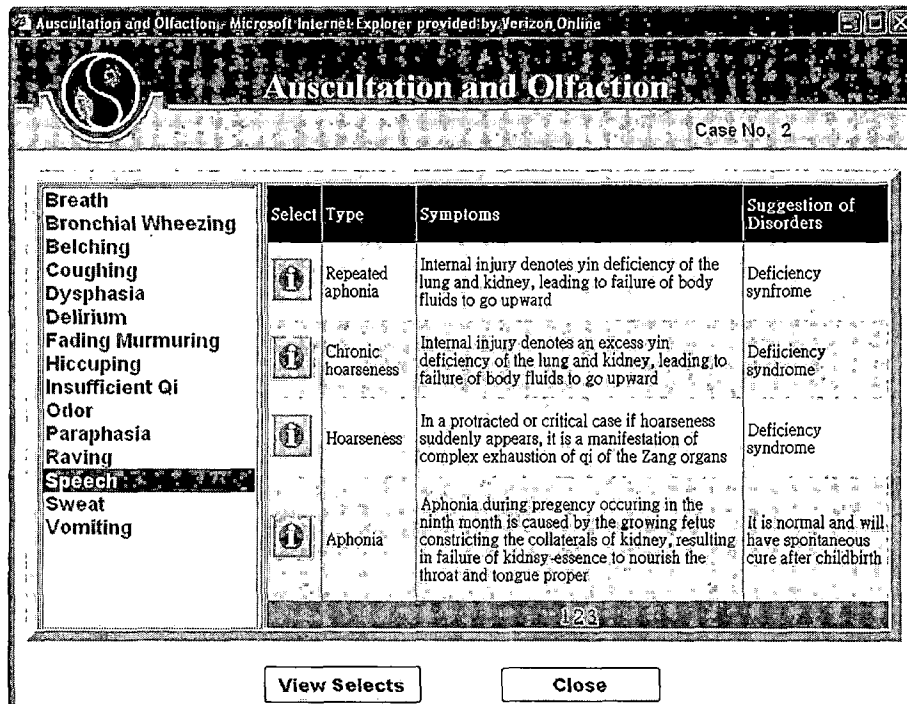


Figure 19. Auscultation and Olfaction Page

#### 4.2.15 Palpation and Pulse-Taking Page

The Palpation and Pulse-Taking page contains many main types of Palpation and Pulse-Taking in the list box. Actually, the functions are as same as the Inspection. If the acupuncturist would like to look at the selected main types of Palpation, he could select --Palpation-- in the ListBox and click the View Selects button. If the acupuncturist would like to look at the selected main types of Pulse-Taking, he could select -- Pulse-Taking -- in the list box and click the View Selects button.

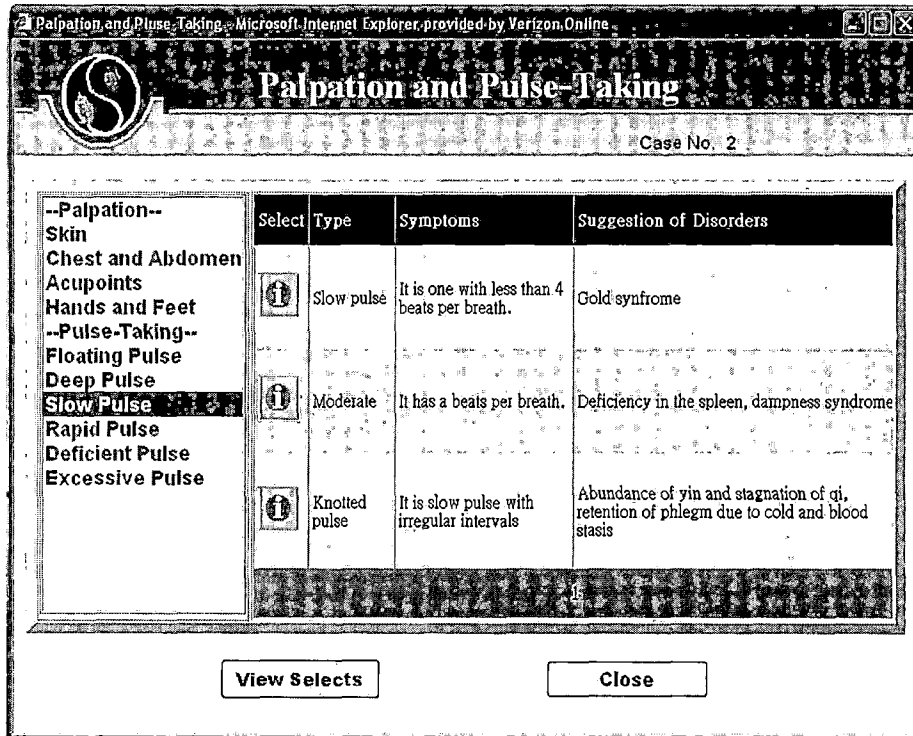


Figure 20. Palpation and Pulse-Taking Page

#### 4.2.16 Selected Items Page

The Delete LinkButton allows the user to delete the item. The Back button allows the user to go back to the previous page.

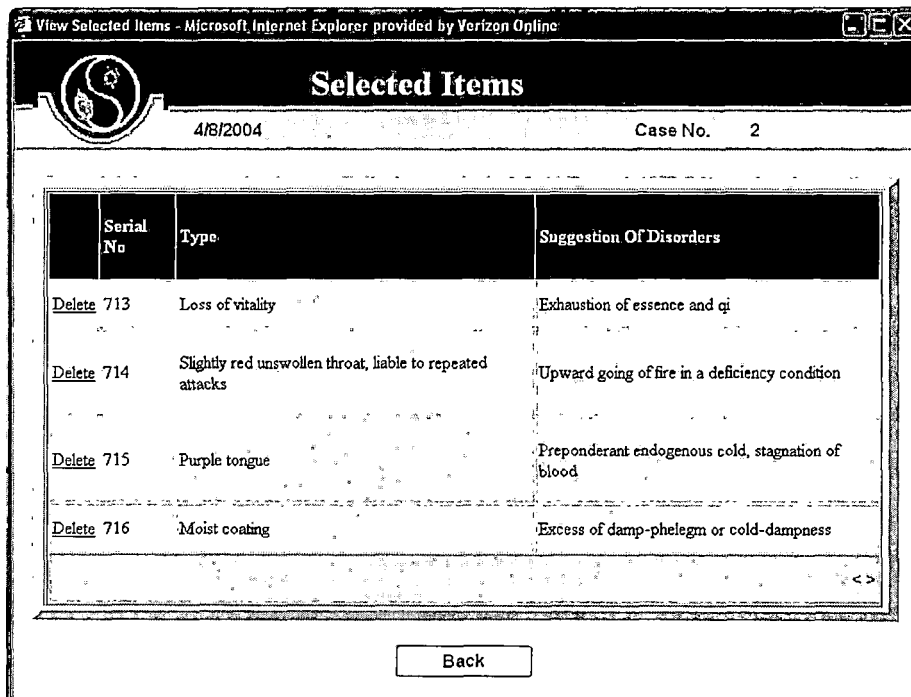


Figure 21. Selected Items Page

#### 4.2.17 Fourteen Meridian Page

In the Fourteen Meridian page, the DrawDownList includes fourteen meridians. The acupuncturist can select any meridian in the draw down list and the page will connect to one of fourteen meridian pages.

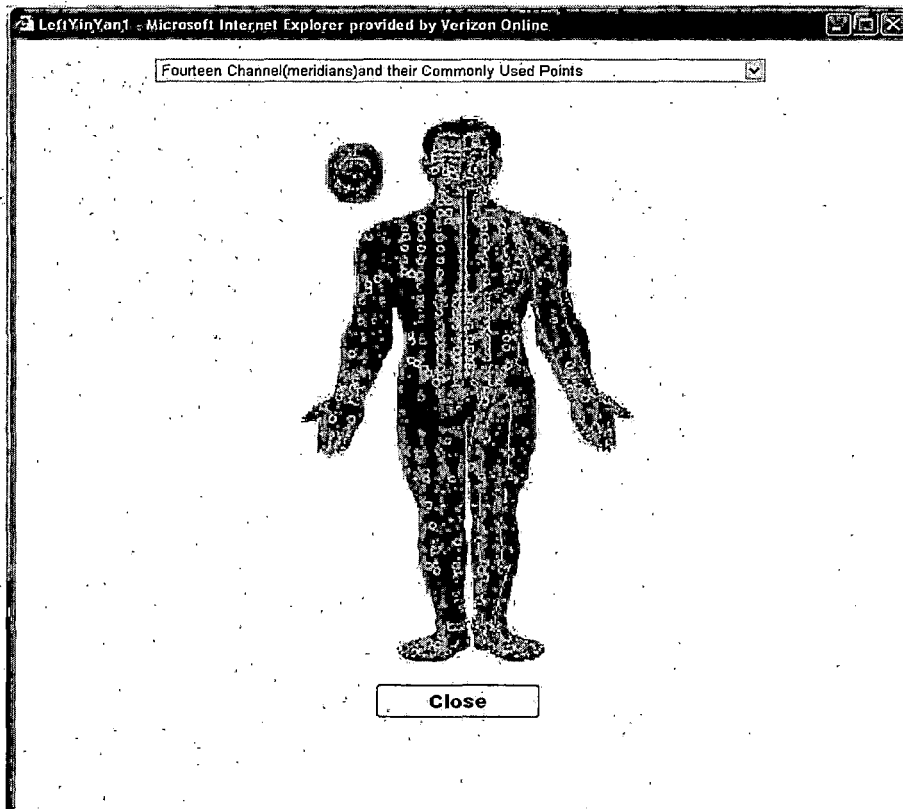


Figure 22. Fourteen Meridians Page

#### 4.2.18 Lung Channel of Hand Tai-Yin

There are total fourteen Channels, so there have fourteen pages. The Lung Channel of Hand Tai-Yin is one of examples. Each check box indicates each acupuncture point. While the acupuncturist checks these check boxes and clicks the Save button, the selected items will be saved into the CaseHistory table. The Close button will close the window.

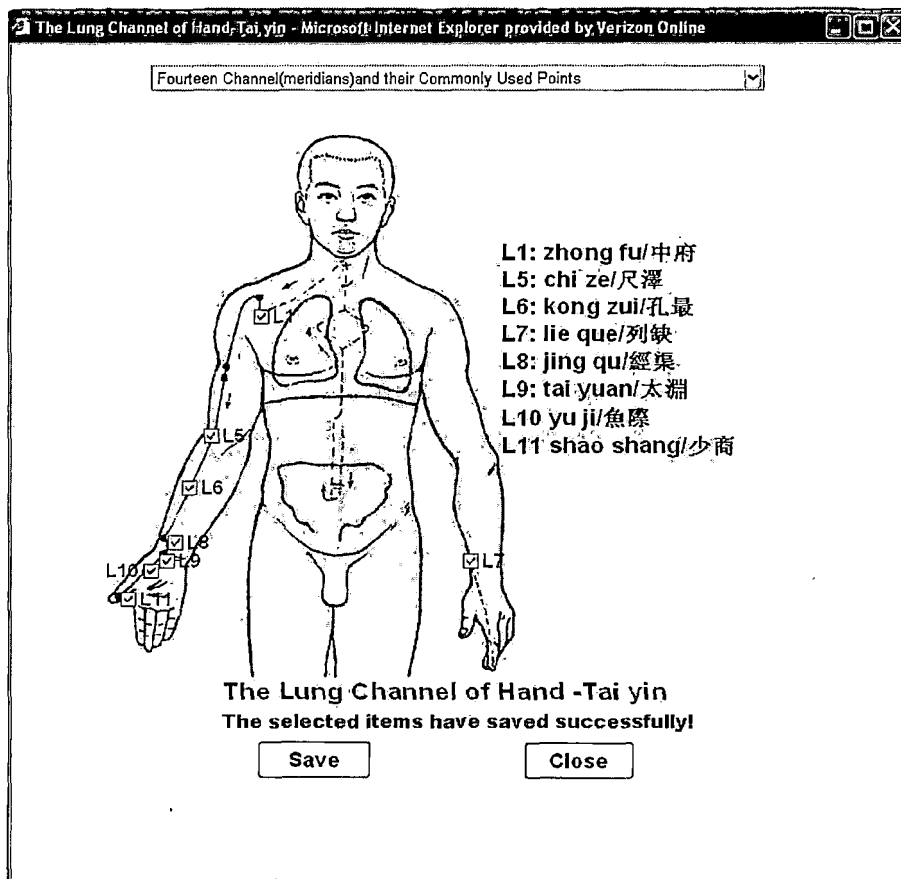


Figure 23. Lung Channel of Hand Tai-Yin

#### 4.2.19 Prescription Page

If the acupuncturist inputs the stroke in the text box and clicks the Search button, the list box will appear Chinese herb's name. If the Acupuncturist inputs first alphabet in the TextBox and clicks the Search button, the list box will display the herbs through syllables (PinYin). While the acupuncturist can select from the list box and click the Add button, the MultiLine-TextBox will display the selected items. When click the Quantity



button, a pop up window will allow the user to add the quantity. After the user complete the selection, clicking the Save button will save all selected items and their quantities. The Exit button will close the window.

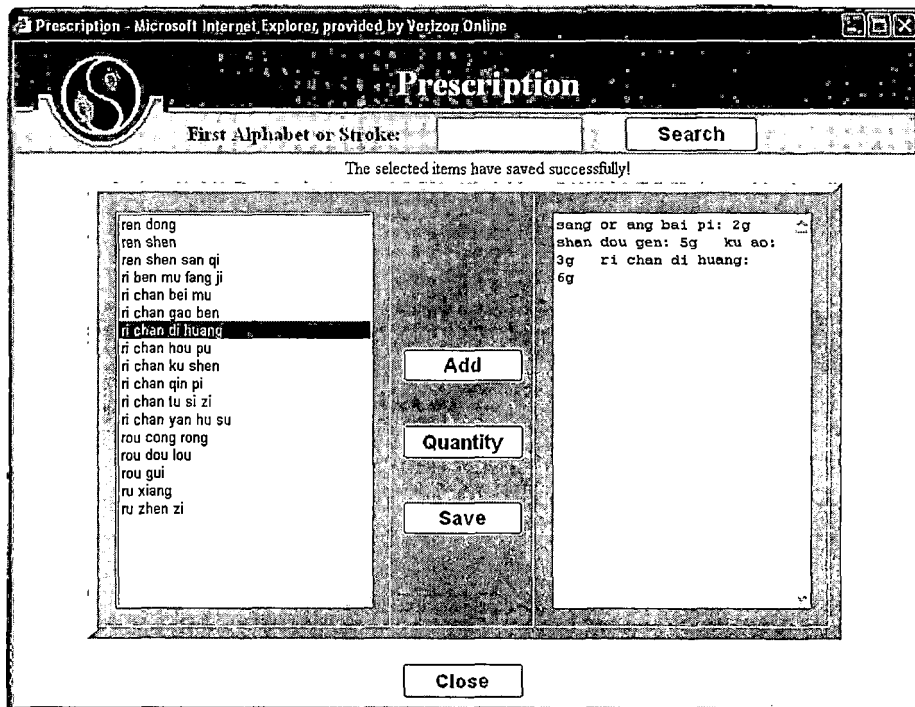


Figure 24. Prescription Page

#### 4.2.20 The Herb's Quantity's Page

If the acupuncturist inputs the quantity in the text box and then clicks the send in the Herb's Quantity's page, then data will send to the right side of the text box in the Prescription page.

The reset button will clean the text box and the Close button will close the Herbs' Quantity page.

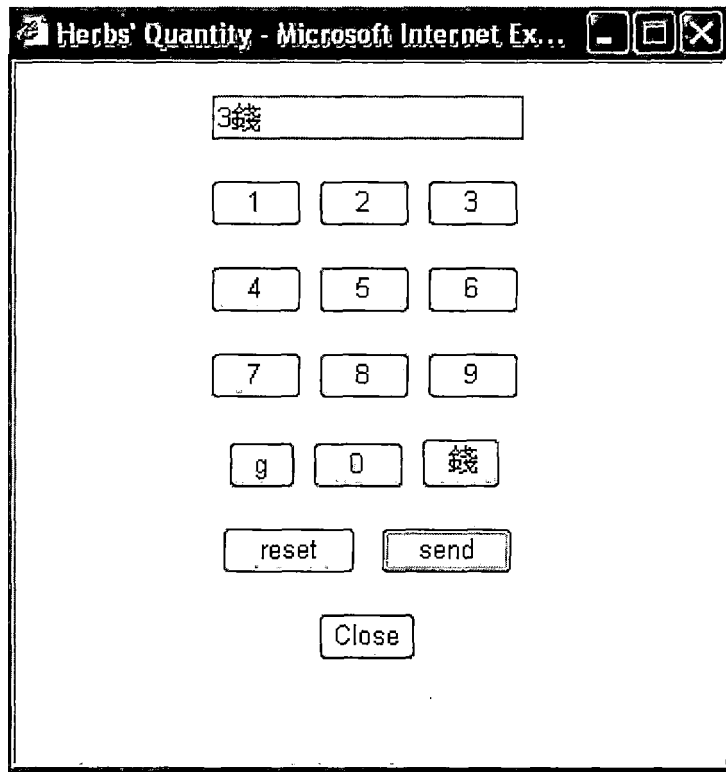


Figure 25. Herb's Quantity Page

#### 4.2.21 Diagnosis Page

There is a list of diagnoses in the list box. While the acupuncturist can select the diagnosis and click Add button, the diagnosis will add into the TextBox at the right hand side of the Diagnosis page. The Save button will save the diagnoses in the TextBox. The Close button will close the window.

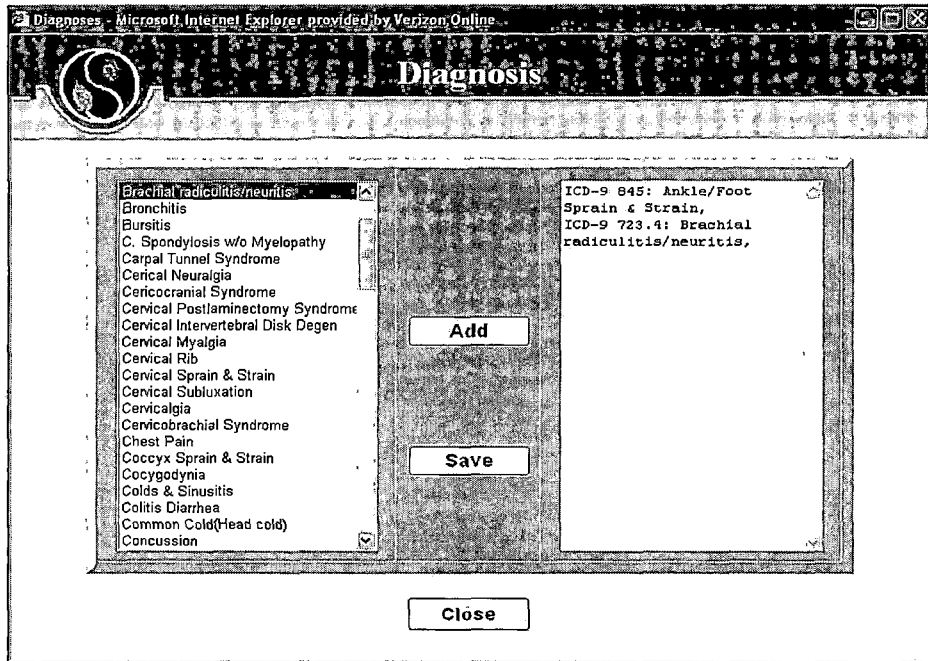


Figure 26. Diagnosis Page

#### 4.2.22 Edit Current Case History Page

The Edit Current Case History includes the patient's information, and seven DataGrids such as Inquiry, Inspection, Auscultation and Olfaction page, Palpation and Pulse-Taking, Diagnosis, Acupuncture points, and prescription. Each row of the DataGrid can be edited and deleted. The Close button in the button of the page is used to close the window.

2 Edit Current Case History - Microsoft Internet Explorer provided by Verizon Online

**Case History**

4/19/2004 Case No. 1

First Name: Peter M.D., S. Last Name: Danitz Sex: M Birth Date: 12/6/1967

**Inquiry:**

Type	Suggestion Of Disorders	Edit	Delete
Tenesmus	Colibacter of dem-pact and qia-cpation in intestines	Edit	Delete
Loose stools associated with poor appetite, low abdominal distension and a dull pain in the abd		Edit	Delete
Reddened conjunct eyes	Symptoms of acute conjunctivitis	Edit	Delete
<>			

**Inspection:**

Type	Suggestion Of Disorders	Edit	Delete
Quiescent and Dynamic Movement	Deficiency syndrome of jin-qi	Edit	Delete
Quiescent and Dynamic Movement	Vin Syndrome	Edit	Delete
A swollen and translucent serosa of unfixed dia	inguinal hernia	Edit	Delete
Developed of uterus	Overstrain of the childbed	Edit	Delete
<>			

**Accumulation and Olfaction:**

Type	Suggestion Of Disorders	Edit	Delete
Leitium	An excess syndrome due to the zinc disturbed by heat	Edit	Delete
Insufficient qi	A deficiency syndrome and weak physique	Edit	Delete
Tas odor from the mouth	A filthy mouth	Edit	Delete
Tas odor from the mouth	Undigestion	Edit	Delete

Figure 27. Edit Current Case History Page

#### 4.2.23 Print Current Case History Report Page

The Print Current Case History page is the similar to the Edit Current page. There is the Print Button under the title. When the acupuncturist click the Print button, the button will display the name of the acupuncturist, print out the case history, and then close the window.

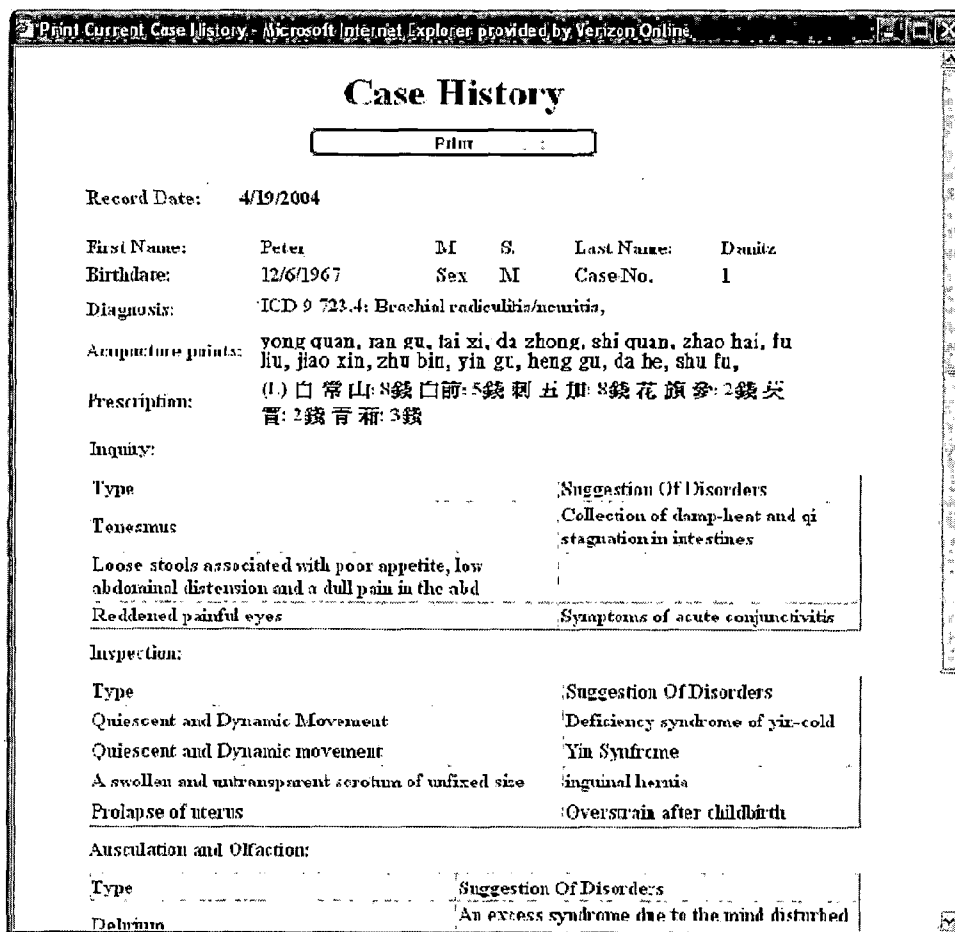


Figure 28. Print Current Case History Report Page

#### 4.2.24 The Print Prescription Page

The Print Prescription page will display the Chinese herbs as the prescription. After the acupuncturist click the Print button, the prescription will print out and also display the name of acupuncturist.

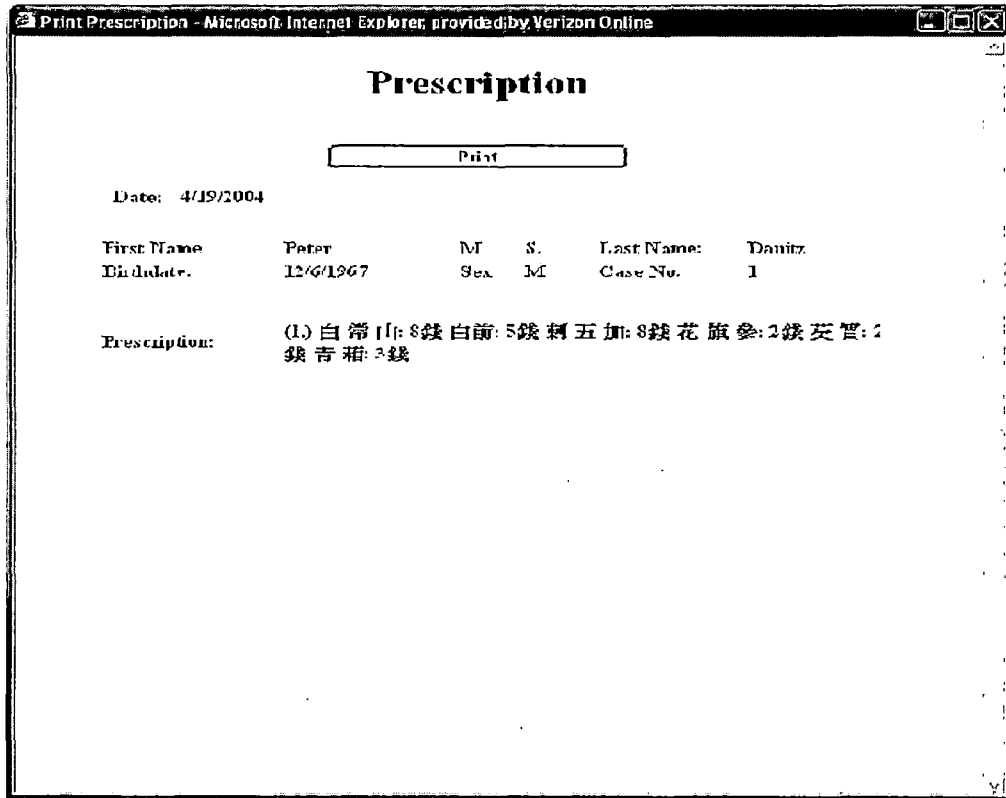


Figure 29. Print Prescription Page

#### 4.2.25 View Patient's Previous Case History Page

The acupuncturist can trace the previous case history through selecting the date from the DrawDownList.

View Patient's Previous Case History Microsoft Internet Explorer provided by Verizon Online

## Previous Case History

4/19/2004      2/6/2004      Case No. 1

First Name: Peter	M Init: S	Last Name: Danitz	Sex: M	Birthdate: 12/6/1967
-------------------	-----------	-------------------	--------	----------------------

**Inquiry:**

Type	Suggestion Of Disorders
Headache	Excess syndrome
Headache	Excess syndrome
Night sweats	Yin deficiency
Swelling	Interior deficiency syndrome

<>

**Inspection:**

Type	Suggestion Of Disorders
Loss of visibility	Asthenia syndrome
Red sclera	Fireing lung-fire
Yellow nose	Interior accumulation of damp-heat
Red nose	Heat in the spleen and lung meridians

<>

**Auscultation and Olfaction:**

Type	Suggestion Of Disorders
Coughing with a low, weak, multiple voice	Deficiency syndrome
Essential wheezing	It is protracted disease, difficult to cure
Strachasia	Insufficient heart-g and c disturbed mind

<>

Figure 30. View patient's Previous Case History Page

#### 4.2.26 The Medical History Page

The Print button in the Medical History will print the page and also close the window. If the Acupuncturist does not print the page, the page will close automatically after 25000 ms.

View Patient's Medical History - Microsoft Internet Explorer provided by Verizon Online

## Medical History

Case No.	<input type="text" value="1"/>	Sex	<input type="text" value="M"/>	DOB	<input type="text" value="4/8/2004"/>	SSN	<input type="text" value="345-67-6789"/>
First Name:	<input type="text" value="Pater"/>	MInitial	<input type="text" value="S"/>	Last Name:	<input type="text" value="Danitz"/>		

1. Are you under a physician's care now?

If so, Please give the reason for treatment.

2. Are You take any Medication at this time?

If so, what medication are you currently taking?

Medication	Dose	Frequency
<input type="text" value="Flonase"/>	<input type="text" value="1"/>	<input type="text" value="2"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

3. The illness the patient have ever had:

Allergies, Seasonal Allergies,

Figure 31. View Patient's Medical History Page

#### 4.2.27 Patient Information

After click the print button, the page will be print and the window will close automatically. If the user does not click print button, the window will close automatically after 20 seconds.



Microsoft Internet Explorer provided by Verizon Online

## Patient Information

Case No. 2

Last Name:	Wei	Minit:	S.	First Name:	Nancy
Birthdate : (dd/mm/yy)	6/6/1990	Sex: (F/M)	F	SSN:	678-78-7345
Driver License:		Marital Status: (S/M/D)			M
Phone:	(626)831-5678	E-Mail: (optional)		Lily345@yahoo.com	
Address:	528 E. Myrtle. Ave. Glendora, CA 91741				
Occupation:		Business Phone:		(626)434-3498	
Employer:					
Business Address:					
Contact Person:	Lily Wei	Relation:		Mother	
Emergency Phone:	(626)831-5678	Policy No:		35777422-02	
Insurance Co.	cigna	Insurance Phone:		P.O Box 182223.	
Billing Address:					

**Print**

Figure 32. Patient' Information Page

## CHAPTER FIVE

### MAINTENANCE

#### 5.1 Introduction

The maintenance manual contains the information about the setup, backup, and restore. It helps the user to know how to do it step by step. There are four major parts of processes, which will be described in the following sections.

#### 5.2 Software Installation

The TCMCS must be installed in the Window XP environments with Internet Explorer. And also it requires install Microsoft SQL Server 2000, Internet Information Service 5.1, and grant access to the ASP.NET service.

##### 5.2.1 Internet Information Service Installation

In order to install the IIS 5.1, follow the three major steps as following. You might use the reference book for the details about the installation of the IIS.

First, you need to put the Window XP CD into CD Rom Drive and select "Install Optional Windows Components." There is a pop up window called Windows Components Wizard. There are many check boxes in the Windows Components Wizard. Check the Internet Explorer and Internet Information Service, and then click Next button.

Second, Click Start button in the button of left side, select Control panel and click the icon of Administrative Tools. After clicking the Internet Information Services shortcut in the Administrative Tools wizard, Internet Information Services Wizard pops up. Click the local computer name in left side of the Internet Information Services Wizard, click the Web Site folder, and right click Default Web Site where you can create a new virtual directory-GraduateProject for the TCMCS.

Third, open the Internet Explorer, click the Tools in the menu, and select Internet Options. In the General tab, type" `http://localhost/GraduateProject/LoginControl/UserControl.aspx`" in the address text Box, click Apply button, and then click Ok button.

Forth, please reference the Microsoft IIS 5.0 for the details about configuring a Site's home directory, ports, IP Addresses, and host name etc.

#### 5.2.2 MS SQL Server 2000

To begin the installation, insert the SQL Server 2000 CD. After that the system shows to you the SQL Server Automenu, you can install SQL Server 2000 Components (either Standard or Desktop Edition) and then continue the following steps. To Start the installation process of the Standard Edition, click the option with the same name. In

the computer name wizard, enter the name of the computer on which you want to create a new installation.

1. Install the database system on your local computer that belongs to the network. (Both the local and remote computer must be running MS Window XP.)  
Select local computer in the Computer Name wizard, and then click the next button.
2. In the Installation Selection wizard, select "Create a new instance of SQL Server" and then click Next button.
3. Accept Software License Agreement.
4. Accept default instance name and click Next button
5. Select the Typical installation and click the Next button
6. Select "Use the same account for each service. Auto start SQL Server Service," select "Use a Domain User Account" in the service setting, input username, password, domain and click the Next button.
7. Configuring SQL Server after installation.
  - Create server groups and register the server using SQL Server Enterprise Manager
  - Set Server option

8. Create Server groups and register the server
  - Run Enterprise manager for the first time, it automatically registers the local SQL Server.
  - When you register a server, you must specify the server name, the security type (if you use SQL server security, you must provide the SQL Server login ID and password of the server you are registering) and the name of the group where you want the server to be listed after it is registered.
9. Set server Options: Many server configuration options Have been simplified and streamlined so that, in most Case you do not have to set server options.
10. Download security patches from Microsoft website and follow the instruction to install.

For the detail of the installation of MS SQL Server 2000, you might need to consult some reference books.

### 5.2.3 Grant Access the ASPNET User

Replace MACHINENAME with the name of your computer, and run the following commands from the command line.

Note. The path to the osql.exe command must be located in your PATH environment variable.

```
Osql -E -S(local)\NetSDK -Q "sp_grantlogin 'MACHINENAME  
\ASPNET'"
```

```
Osql -E -S(local)\NetSDK -d ClinicSystem1 -Q  
"sp_grantdbaccess 'MACHINENAME\ASPNET'"
```

```
Osql -E -S(local)\NetSDK -d ClinicSystem1 -Q  
"ap_addrolemember 'db_owner', 'MANCHINENAME\ASPNET'"
```

### 5.3 Traditional Chinese Medical Clinic System Installation

There are four major steps to install TSMCS as following:

1. All TCMCS file are store in the directory  
C:\Inetpub\wwwroot\GraduateProject>LoginControl.
2. The Vistual Studio Solution LoginControl is  
stored in the directory C:\Documents and  
Settings\prophecy\My Documents\Visual Studio  
Projects>LoginControl.
3. The ClinicSystem1 database should be stored in  
the Directory C:\Program Files\Microsoft SQL  
Server\MSSQL\$NetSDK\Data. If the Database  
ClinicSystem1 is in a backup file, I will use  
Enterprise Manager to restore the ClinicSystem1.  
To store ChinicSystem1 database, expand the  
server, choose Databases, and Right-click the  
database. After pointing to all Tasks choose

Restore Database. In the Restore Database dialog box, choose Database if you want to install from a database backup file. After that, on the General tab, click From device and enter ClinicSystem1 in Restore as database. Click Select devices. In the Choose Restore Devices dialog box, Check Restore from Disk, click Add and enter the name of the backup device (C:\Program Files\Microsoft Server\MSSQL\$NetSDK\Data\ClinicSystem1.MDF).

4. Open the internet Explorer, click Tools in the menu, select internet Options, input <http://localhost/GraduateProject/LoginControl/Default Page.aspx> into Address text box.

#### 5.4 Backup and Restore

Backup and Restore are two important tasks to prevent data loss.

##### 5.4.1 System Backup

All files of TCMCS are located in the directory C:\Inetpub\wwwroot\GraduateProject>LoginControl and the Visual studio solution file LoginControl is location in the directory C:\Documents and Settings\prophecy\My Documents\Visual Studio Projects>LoginControl. I will

suggest that put the solution file in the same location with all files of TCMCS and use the WinZip to compress the folder.

#### 5.4.2 Database Backup

In the database backup, expand the server, click management, right-click Backup, and choose New Backup Device. In the Backup Device Properties dialog box, either enters the name of the disk device (if you clicked file name) or the name of the tape device (if you clicked Type drive name). After you specify backup devices, a database backup can be down. Expand the server; choose Databases, and right-click the database. After pointing to All Tasks, choose Backup Database. The SQL Server Backup dialog box appears.

- In the general tab of the dialog box, enter the backup set name in the Name field and, in the Description field, a description of this set. For the full database backup, click Database-complete.
- In the Destination frame, select an existing device by clicking the Add button.
- To append to an existing backup on the selected device, check the Append to media check box under the Destination frame. To overwrite existing backup in the selected backup device.



- In the schedule frame, you can optionally determine at what time the backup operation will be down for later or periodic execution
- In the options tab of the SQL Server Backup dialog box and then click Verify backup upon completion. You can also initialize backup media by clicking initialize and label media and then enter the media set name and description.

#### 5.4.3 System Restore

There are two steps to restore system as following.

1. To extract the backup system file by using WinZip and puts it in the directory C:\Inetpub\wwwroot\.
2. Copy visual studio solution file LoginControl from the backup file and put it in the directory C:\Documents and Settings\prophecy\My Documents\Visual Studio Projects>LoginControl.

Through doing the steps above, all the TCMCS files will restore into the directory C:\Inetpub\wwwroot\ and complete the restore system process.

#### 5.4.4 Database Restore

To restore ClinicSystem1 database from a full database backup, expand the server, choose Database and right click the database. After pointing to All Tasks,

choose Restore Database. The Restore database dialog box appears. In the Restore database dialog box, there are some steps in the following to be completed.

1. On the General tab in the Restore frame, click Database. Select the ClinicSystem1 database backup you want to restore in the Restore as database list.
2. To select the appropriate restore options, choose the Options tab of the Restore database dialog box. In the Recovery completion state frame, select Leave database optional. No additional transaction logs can be restored, which instructs SQL Server to roll forward any committed transaction and to roll back any uncommitted transaction. After applying this option, the database is in a consistent state and is ready for use.

## CHAPTER SIX

### CONCLUSION AND FUTURE DIRECTIONS

#### 6.1 Conclusion

The Traditional Chinese Medical Clinic System is designed for acupuncture clinics. The system can help staff to schedule and edit appointments, and allow acupuncturist to record and edit patients' information and assist in recording case history and managing user accounts. The system can raise working efficiency, improve service quality and is easy to use. The TCMCS is definitely a helpful commercial software system, which will help small acupuncture clinics to provide quality service. The system can have some more improvements at prescription library and diagnoses methods. There exists more than thousands of herbs' prescriptions for different treatments. If the database contains a prescription library in the system, the acupuncturist can save more time on selecting single herb from the herbs list. Many acupuncturists' hope that diagnosis can come out automatically after the user selected some symptoms and suggestion of disorders. If we can collect the case histories from experience acupuncturists in different specialties, and get help from them, the system might be

able to evaluate all symptoms and suggest the diagnosis. Through the system, the acupuncturist can compare his own diagnosis with the system, which might help acupuncturist not only diagnose accurately, but also able to give a right treatment.

## 6.2 Future Directions

The TCMCS only focuses on the clinic in the US, since it is based on English. The system can be also developed to more fully support the Chinese language in order to fit the Chinese environment.

Additionally, if the system contains more tables in the database, the system can contain more acupuncture points and herbal prescriptions, which may help it to become widely accepted commercial software.

APPENDIX  
SOURCE CODE PRINTOUT

## File :UserControl.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class UserControl : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label userNameLabel;
        protected System.Web.UI.WebControls.Label passwordLabel;
        protected System.Web.UI.WebControls.TextBox userNameText;
        protected System.Web.UI.WebControls.Button Button2;
        protected System.Web.UI.WebControls.Button loginBtr;
        protected System.Web.UI.WebControls.Label errorMessage;
        protected System.Web.UI.WebControls.TextBox passwordText;
        protected static int timeCt;
        public static string loginDate;
        public static string AcupName;
        public static string staffName;
        private string userSQL;
        public static string staffPwd;
        public static bool uCount;
        private void Page_Load(object sender, System.EventArgs e)
        {
            staffName="none";
            AcupName="none";
            staffPwd="none";
        }
        private void loginClick(object sender, System.EventArgs e)
        {
            timeCt+=1;
            loginDate=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
            DateTime.Now.Day,DateTime.Now.Year);
            string newConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
            string str1, str2;
            str1=userNameText.Text;
            str2=passwordText.Text;
            SqlConnection objConn=new SqlConnection(newConn);
            //chack the input sring from the StaffAccount table
            userSQL="SELECT Eno FROM StaffAccountsNew WHERE UserName='"+str1+"' and
            Password='"+str2+"'";
            SqlCommand obComm1=new SqlCommand(userSQL, objConn);
            objConn.Open();
            SqlDataReader strReader1=objComm1.ExecuteReader();
            if(strReader1.Read().ToString().Equals("True"))
            {
                objConn.Close();
                SqlConnection objConn_UC=new SqlConnection(newConn);
                objConn_UC.Open();
                SqlDataAdapter objAdapter_UC= new SqlDataAdapter(userSQL, objConn_UC);
                DataSet objDataSet_UC=new DataSet("dtSelectedData");
                objAdapter_UC.Fill(objDataSet_UC, "dtSelectedData");
                string employeeNo=Convert.ToString(objDataSet_UC.Tables["dtSelectedData"].Rows[0]["Eno"]);
                objConn_UC.Close();
                if(employeeNo.Equals("1"))
                {
                    AcupName=str1.ToString();
                    Response.Redirect("AcupPage.aspx");
                }
            }
        }
    }
}
```

```

    }
    else
    {
        staffName=str1.ToString();
        if(DefaultPage.btnEnterPermission)
            this.Response.Redirect("Staffpage.aspx");
        else
            errorMessage.Text="You cannot login at this moment.";
    }
}
else
{
    objConn.Close();
    errorMessage.Text="Login attempt failed, invalid username or password";
}
}
private void clearClick(object sender, System.EventArgs e)
{
    userNameText.Text="";
    passwordText.Text="";
    errorMessage.Text="";
}
}
}
}

```

### File: UserAccount.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class UserAccount : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.ListBox FNListBox;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.TextBox FNTxtBox;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.TextBox MInitTxtBox;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.TextBox LNameTxtBox;
        protected System.Web.UI.WebControls.Label Label4;
        protected System.Web.UI.WebControls.Label EnoLabel;
        protected System.Web.UI.WebControls.Label Label5;
        protected System.Web.UI.WebControls.TextBox PhoneTxtBox;
        protected System.Web.UI.WebControls.Label Label6;
        protected System.Web.UI.WebControls.TextBox UserTxtBox;
        protected System.Web.UI.WebControls.Label Label7;
        protected System.Web.UI.WebControls.TextBox PwdTxtBox;
        protected System.Web.UI.WebControls.Button NewUserBtn;
        protected System.Web.UI.WebControls.Button saveBtn;
        protected System.Web.UI.WebControls.Button deleteBtn;
        protected System.Web.UI.WebControls.Button clearBtn;
        protected System.Web.UI.WebControls.Button exitBtn;
        private string newConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        private string strQuery1="SELECT * FROM StaffAccountsNew;";
        protected System.Web.UI.WebControls.Label titleLabel;
        protected System.Web.UI.WebControls.Label MessageLabel;
        protected System.Web.UI.WebControls.TextBox ConfirmTxtbox;
    }
}

```

```

protected System.Web.UI.WebControls.Label ConfirmLabel;
private SqlConnection objConn;
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
    if(! IsPostBack)
        listFirstName();
}
private void listFirstName()
{
    objConn = new SqlConnection(newConn);
    //DataAdapter set up
    SqlDataAdapter objAdapter = new SqlDataAdapter(strQuery1, newConn);
    //Dataset & adapter & Table
    DataSet objDataSet=new DataSet("dtStaffAccounts");
    objAdapter.Fill(objDataSet, "dtStaffAccounts");
    FNListBox.DataSource=objDataSet;
    FNListBox.DataTextField="FirstName";
    FNListBox.DataBind();
}
private void exitClick(object sender, System.EventArgs e)
{
    Response.Redirect("AcupPage.aspx");
}
private void clearClick(object sender, System.EventArgs e)
{
    MessageLabel.Text="";
    FNTxtBox.Text="";
    MInitTxtBox.Text="";
    LNameTxtBox.Text="";
    EnoLabel.Text="";
    PhoneTxtBox.Text="";
    UserTxtBox.Text="";
    PwdTxtBox.Text="";
}
private void deleteClick(object sender, System.EventArgs e)
{
    if(!EnoLabel.Text.Equals(""))
    {
        int ENO;
        ENO=Convert.ToInt32(EnoLabel.Text);
        objConn =new SqlConnection(newConn);
        objConn.Open();
        SqlDataAdapter objAdapter = new SqlDataAdapter(strQuery1, objConn);
        DataSet dsObj = new DataSet();
        objAdapter.Fill(dsObj, "dtStaffAccounts");
        objConn.Close();
        DataTable dtTable = dsObj.Tables["dtStaffAccounts"];
        dtTable.PrimaryKey = new DataColumn[] {dtTable.Columns["Eno"]};
        DataRow row = dtTable.Rows.Find(ENO);
        if(!ENO.Equals(1))
        {
            row.Delete();
            // Reconnect the DataSet and delete the record from the database
            SqlCommandBuilder cb = new SqlCommandBuilder(objAdapter);
            objConn.Open();
            objAdapter.Update(dsObj, "dtStaffAccounts");
            objConn.Close();
            FNTxtBox.Text="";
            LNameTxtBox.Text="";
            MInitTxtBox.Text="";
            EnoLabel.Text="";
            PhoneTxtBox.Text="";
        }
    }
}

```



```

FNTxtBox.Text=row["FirstName"].ToString();
MinitTxtBox.Text=row["Minit"].ToString();
LNameTxtBox.Text=row["LastName"].ToString();
if((phoneStr!=PhoneTxtBox.Text)||((userIDStr!=UserTxtBox.Text)&&
(ch.Equals(false))||((pwdStr!=PwdTxtBox.Text)&&(!PwdTxtBox.Text.Equals(""))&&
(!ConfirmTxtbox.Text.Equals(""))))
    MessageLabel.Text="You have updated successfully!";
else
{ if(ch.Equals(true))
    {MessageLabel.Text="Please choose different username!";
    else if(!PwdTxtBox.Text.Equals(""))&&(ConfirmTxtbox.Text.Equals(""))
    {MessageLabel.Text="You need to fill in both password and confirm password
columns.";}
    else if((PwdTxtBox.Text.Equals(""))&&(!ConfirmTxtbox.Text.Equals(""))
    {MessageLabel.Text="You need to fill in both password and confirm password
columns.";}
    else if((PwdTxtBox.Text.Equals(""))&&(ConfirmTxtbox.Text.Equals(""))
    {MessageLabel.Text="You did not make any change.";}
    }
}
}
else
{
    if((FNTxtBox.Text.Equals(""))||(LNameTxtBox.Text.Equals(""))|(PhoneTxtBox.Text.Equals(""))|
    (UserTxtBox.Text.Equals(""))|(PwdTxtBox.Equals(""))
    {
        MessageLabel.Text="Please enter the data in the TextBoxes and push 'Add New' button!";
    }
    else
    {
        MessageLabel.Text="Wrong button! Please push 'Add New' button!";
    }
}
}
}
private bool usernameCheck(string user1)
{
    objConn =new SqlConnection(newConn);
    string strQ="SELECT UserName FROM StaffAccountsNew where UserName="+user1+"";
    SqlCommand obComm=new SqlCommand(strQ, objConn);
    objConn.Open();
    SqlDataReader strRr=obComm.ExecuteReader();
    if(strRr.Read().ToString().Equals("True"))
    {
        objConn.Close();
        return true;
    }
    else
    {
        objConn.Close();
        return false;
    }
}
private void AddNewStaff(object sender, System.EventArgs e)
{
    string name1, name2, name3, phn, user, pwd;
    if(EnoLabel.Text.Equals(""))
    {
        if((!FNTxtBox.Text.Equals(""))&&(!LNameTxtBox.Text.Equals(""))&&
        (!PhoneTxtBox.Text.Equals(""))&&(!UserTxtBox.Text.Equals(""))&&
        (!PwdTxtBox.Text.Equals(""))&&(!ConfirmTxtbox.Text.Equals(""))
        {
            if((FNTxtBox.Text.Length>15)|(LNameTxtBox.Text.Length>15)|
            (MinitTxtBox.Text.Length>15)|(PhoneTxtBox.Text.Length!=13)|
            (UserTxtBox.Text.Length>10)|(PwdTxtBox.Text.Length>10))
            {
                if(FNTxtBox.Text.Length>15)
                    MessageLabel.Text="The length of first name must be less or equal to fifty.";
            }
        }
    }
}

```

```

else if(LNameTextBox.Text.Length>15)
    MessageLabel.Text="The length of last name must be less or equal to fifty.";
else if(MInitTextBox.Text.Length>15)
    MessageLabel.Text="The length of middle name must be less or equal to fifty.";
else if(PhoneTextBox.Text.Length!=13)
    MessageLabel.Text="The telephone format is incorrect";
else if(UserTextBox.Text.Length>10)
    MessageLabel.Text="The length of user name must be less or equal to ten.";
else if(PwdTextBox.Text.Length>10)
    MessageLabel.Text="The length of password must be less or equal to ten.";
}
else
{
    MessageLabel.Text="";
    name1=FNTxtBox.Text;
    name2=MInitTextBox.Text;
    name3=LNameTextBox.Text;
    phn=PhoneTextBox.Text;
    user=UserTextBox.Text;
    if(PwdTextBox.Text.Equals(ConfirmTxtbox.Text))
    {
        pwd=PwdTextBox.Text;
        SqlConnection objConn1=new SqlConnection(newConn);
        string strQy1="SELECT * FROM StaffAccountsNew WHERE
        UserName='"+user+"'";
        SqlCommand obComm1=new SqlCommand(strQy1, objConn1);
        objConn1.Open();
        SqlDataReader strReader=obComm1.ExecuteReader();
        if(strReader.Read().ToString().Equals("True"))
            MessageLabel.Text="Please change the user name.";
        else
        {
            objConn=new SqlConnection(newConn);
            string InputStr="INSERT INTO StaffAccountsNew(firstName, MInit, lastName,
            Phone, UserName, Password) VALUES('"+name1+"', '"+name2+"',
            '"+name3+"', '"+phn+"', '"+user+"', '"+pwd+"'); SELECT @@IDENTITY As
            'Identity'";
            SqlCommand dbComm=new SqlCommand(InputStr, objConn);
            int iID=0;
            try
            {
                objConn.Open();
                iID = Convert.ToInt32( dbComm.ExecuteScalar());
                MessageLabel.Text="You have added a new staff successfully!";
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
                Response.End();
            }
            finally
            {
                if (objConn.State == ConnectionState.Open)
                {
                    objConn.Close();
                }
            }
            EnoLabel.Text=iID.ToString();
            listFirstName();
        }
    }
    //end else
    objConn1.Close();
}
//end of if
else if(!PwdTextBox.Text.Equals(ConfirmTxtbox.Text))
{MessageLabel.Text="Please check the password and retype again!";}
// end of else
}
// end of if
else if((FNTxtBox.Text.Equals(""))||(LNameTextBox.Text.Equals(""))||

```



```

private void ScheduleAnAppClick(object sender, System.EventArgs e)
{
    Response.Redirect("ScheduleAnAppointment.aspx");
}
private void VEAppClick(object sender, System.EventArgs e)
{
    Response.Redirect("ViewEditAppointments.aspx");
}
private void VMPInfoClick(object sender, System.EventArgs e)
{
    Response.Redirect("ViewModifyPatientInfo.aspx");
}
private void NPatientClick(object sender, System.EventArgs e)
{
    Response.Redirect("CreateNewPatients.aspx");
}
private void LogoutClick(object sender, System.EventArgs e)
{
    DefaultPage.blnEnterPermission=true;
    Response.Redirect("UserControl.aspx");
}
}
}
}

```

### File: PreviousRecord.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
namespace LoginControl
{
    public class Staffpage : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.LinkButton VEAppointmentBtn;
        protected System.Web.UI.WebControls.LinkButton VMPInformation;
        protected System.Web.UI.WebControls.LinkButton CreateNewPatient;
        protected System.Web.UI.WebControls.LinkButton logoutBtn;
        protected System.Web.UI.WebControls.LinkButton ScheduleBtn;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.staffName);
            if(UserName1.Equals("none"))
                Response.Redirect("UserControl.aspx");
            DefaultPage.blnEnterPermission=false;
        }
        public override void Dispose()
        {
            base.Dispose();
        }
        private void ScheduleAnAppClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ScheduleAnAppointment.aspx");
        }
        private void VEAppClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ViewEditAppointments.aspx");
        }
        private void VMPInfoClick(object sender, System.EventArgs e)

```

```

    {
        Response.Redirect("ViewModifyPatientInfo.aspx");
    }
    private void NPatientClick(object sender, System.EventArgs e)
    {
        Response.Redirect("CreateNewPatients.aspx");
    }
    private void LogoutClick(object sender, System.EventArgs e)
    {
        DefaultPage.blnEnterPermission=true;
        Response.Redirect("UserControl.aspx");
    }
}
}
}

```

### File: AcupPage.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class AcupPage : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.LinkButton MyAppointBtn;
        protected System.Web.UI.WebControls.LinkButton SPCHBtn;
        protected System.Web.UI.WebControls.LinkButton logoutBtn;
        protected System.Web.UI.WebControls.LinkButton MUAccountClick;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
        }
        private void MyAppointmentClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ViewAppointments.aspx");
        }
        private void SearchPatientClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ViewPatientsCaseHistory.aspx");
        }
        private void MUAccountClick(object sender, System.EventArgs e)
        {
            Response.Redirect("UserAccounts.aspx");
        }
        private void LogoutClick(object sender, System.EventArgs e)
        {
            Response.Redirect("UserControl.aspx");
        }
    }
}
}

```

### File: CAsEHisory.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class AcupPage : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.LinkButton MyAppointBtn;
        protected System.Web.UI.WebControls.LinkButton SPCHBtn;
        protected System.Web.UI.WebControls.LinkButton logoutBtn;
        protected System.Web.UI.WebControls.LinkButton MUAccountClick;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
        }
        private void MyAppointmentClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ViewAppointments.aspx");
        }
        private void SearchPatientClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ViewPatientsCaseHistory.aspx");
        }
        private void MUAccountClick(object sender, System.EventArgs e)
        {
            Response.Redirect("UserAccounts.aspx");
        }
        private void LogoutClick(object sender, System.EventArgs e)
        {
            Response.Redirect("UserControl.aspx");
        }
    }
}
```

### File: CreateNewPatients.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class CreateNewPatients : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label titleLabel;
```

```

protected System.Web.UI.WebControls.Label LastName;
protected System.Web.UI.WebControls.TextBox lastNameText;
protected System.Web.UI.WebControls.Label MInit;
protected System.Web.UI.WebControls.TextBox mInitText;
protected System.Web.UI.WebControls.Label FirstNameLabel;
protected System.Web.UI.WebControls.TextBox firstNameText;
protected System.Web.UI.WebControls.Label Birthdate;
protected System.Web.UI.WebControls.TextBox birthText;
protected System.Web.UI.WebControls.Label Sex;
protected System.Web.UI.WebControls.TextBox sexText;
protected System.Web.UI.WebControls.Label SSN;
protected System.Web.UI.WebControls.Label DriverLicense;
protected System.Web.UI.WebControls.TextBox driverLicenseText;
protected System.Web.UI.WebControls.Label MartialStatus;
protected System.Web.UI.WebControls.TextBox martialText;
protected System.Web.UI.WebControls.Label Phone;
protected System.Web.UI.WebControls.TextBox phoneText;
protected System.Web.UI.WebControls.Label E_mail;
protected System.Web.UI.WebControls.TextBox e_mailText;
protected System.Web.UI.WebControls.Label Address;
protected System.Web.UI.WebControls.TextBox addressText1;
protected System.Web.UI.WebControls.Label Occupation;
protected System.Web.UI.WebControls.TextBox occupationText;
protected System.Web.UI.WebControls.Label BusinessPhone;
protected System.Web.UI.WebControls.TextBox businessPhoneText;
protected System.Web.UI.WebControls.Label Employer;
protected System.Web.UI.WebControls.TextBox employerText;
protected System.Web.UI.WebControls.Label BusinessAddress;
protected System.Web.UI.WebControls.TextBox businessAddressText;
protected System.Web.UI.WebControls.Label Contact;
protected System.Web.UI.WebControls.TextBox contactText;
protected System.Web.UI.WebControls.Label Relation;
protected System.Web.UI.WebControls.TextBox relationText;
protected System.Web.UI.WebControls.Button exitButton;
protected System.Web.UI.WebControls.Button submitButton;
protected System.Web.UI.WebControls.Label EmergencyPhone;
protected System.Web.UI.WebControls.TextBox emergencyPhoneText;
protected System.Web.UI.WebControls.Label PolicyNo;
protected System.Web.UI.WebControls.TextBox policyText;
protected System.Web.UI.WebControls.TextBox insuranceCoText;
protected System.Web.UI.WebControls.TextBox IPhoneTxtBox;
protected System.Web.UI.WebControls.Label billAddLabel;
protected System.Web.UI.WebControls.TextBox billingAddress;
protected System.Web.UI.WebControls.Label InsuranceCoLabel;
protected System.Web.UI.WebControls.Label M1;
protected System.Web.UI.WebControls.Label M3;
protected System.Web.UI.WebControls.Label M6;
protected System.Web.UI.WebControls.Label M2;
protected System.Web.UI.WebControls.Label M5;
protected System.Web.UI.WebControls.Label M4;
protected System.Web.UI.WebControls.Label M7;
protected System.Web.UI.WebControls.TextBox ssnText;
protected System.Web.UI.WebControls.Label M8;
protected System.Web.UI.WebControls.Label M9;
protected System.Web.UI.WebControls.Button clearButton;
private string ConnStr="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
private SqlConnection objConn;
string str1, str2, str3, str4, str5, str6, str7, str8, str9, str10, str11, str12, str13, str14, str15, str16,
str17, str18, str19, str20, str21, str22, str23;
protected System.Web.UI.WebControls.Label InsPhoneLabel;
protected System.Web.UI.WebControls.Label Date;
protected System.Web.UI.WebControls.Label caseNoLabel;
protected System.Web.UI.WebControls.Label No;
protected System.Web.UI.WebControls.Label SuccMessage;
protected System.Web.UI.WebControls.Label M10;
public static string lastName_NMH,sex_NMH,SSN_NMH,firstName_NMH,caseNo_NMH,MInit_NMH;
protected System.Web.UI.WebControls.Label Label1;

```

```

private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.staffName);
    if(UserName1.Equals("none"))
        Response.Redirect("UserControl.aspx");
    Date.Text=String.Format("{0:D2}/{1:D2}/{2:D2}",
        DateTime.Now.Month, DateTime.Now.Day,DateTime.Now.Year);
}
private void exClick1(object sender, System.EventArgs e)
{
    Response.Redirect("StaffPage.aspx");
}
private void clearClick(object sender, System.EventArgs e)
{
    lastNameText.Text="";
    mInitText.Text="";
    firstNameText.Text="";
    birthText.Text="";
    sexText.Text="";
    ssnText.Text="";
    driverLicenseText.Text="";
    martialText.Text="";
    phoneText.Text="";
    e_mailText.Text="";
    addressText1.Text="";
    occuptionText.Text="";
    businessPhoneText.Text="";
    businessAddressText.Text="";
    employerText.Text="";
    emergencyPhoneText.Text="";
    contactText.Text="";
    policyText.Text="";
    mInitText.Text="";
    InsuranceCoText.Text="";
    relationText.Text="";
    IPhoneTxtBox.Text="";
    billingAddress.Text="";
    M1.Text="";
    M2.Text="";
    M3.Text="";
    M4.Text="";
    M5.Text="";
    M6.Text="";
    M7.Text="";
    M8.Text="";
    M9.Text="";
    M10.Text="";
    No.Text="";
    SuccMessage.Text="";
}
private void saveClick(object sender, System.EventArgs e)
{
    //Clean the warning mark
    M1.Text="";
    M2.Text="";
    M3.Text="";
    M4.Text="";
    M5.Text="";
    M6.Text="";
    M7.Text="";
    M8.Text="";
    M9.Text="";
    M10.Text="";
    No.Text="";
    SuccMessage.Text="";
}

```



```

if((lastNameText.Text.Equals("")||firstNameText.Text.Equals("")||birthText.Text.Equals("")
||ssnText.Text.Equals("")||sexText.Text.Equals("")||phoneText.Text.Equals("")||
addressText1.Text.Equals("")||maritalText.Text.Equals("")||contactText.Text.Equals("")||
emergencyPhoneText.Text.Equals("")||maritalText.Text.Equals(""))
{
    if(lastNameText.Text.Equals(""))
        M1.Text="***";
    if(firstNameText.Text.Equals(""))
        M2.Text="***";
    if(birthText.Text.Equals(""))
        M3.Text="***";
    if(ssnText.Text.Equals(""))
        M5.Text="***";
    if(sexText.Text.Equals(""))
        M4.Text="***";
    if(phoneText.Text.Equals(""))
        M6.Text="***";
    if(addressText1.Text.Equals(""))
        M7.Text="***";
    if(contactText.Text.Equals(""))
        M8.Text="***";
    if(emergencyPhoneText.Text.Equals(""))
        M9.Text="***";
    if(maritalText.Text.Equals(""))
        M10.Text="***";
    SuccMessage.Text="Please full in the coulums, before save them.";
}
else if(((!sexText.Text.Equals("F"))&&!sexText.Text.Equals("f"))&&!sexText.Text.Equals("M"))
&&!sexText.Text.Equals("m"))||(!maritalText.Text.Equals("S"))&&
(!maritalText.Text.Equals("M"))&&!maritalText.Text.Equals("D"))&&
(!maritalText.Text.Equals("s"))&&!maritalText.Text.Equals("m"))&&!maritalText.Text.Equals("d"))))
{
    string message1="";
    string message2="";
    if(!sexText.Text.Equals("F"))&&!sexText.Text.Equals("f"))&&
(!sexText.Text.Equals("M"))&&
(!sexText.Text.Equals("m"))
    {
        M4.Text="***";
        sexText.Text="";
        message1="sex column";
    }
    if(!maritalText.Text.Equals("S"))&&!maritalText.Text.Equals("M"))&&
(!maritalText.Text.Equals("D"))&&!maritalText.Text.Equals("s"))&&!maritalText.Text.Equals("m"))
&&!maritalText.Text.Equals("d"))
    {
        M10.Text="***";
        maritalText.Text="";
        message2="marital status coulumn";
    }
    if(!message2.ToString().Equals(""))&&!message1.ToString().Equals(""))
    {SuccMessage.Text=" Incorrect input in the "+message1.ToString()+ " and "
+message2.ToString()+ " !";}
    else if((message2.ToString().Equals(""))&&!message1.ToString().Equals(""))
    {SuccMessage.Text=" Incorrect input in the "+message1.ToString();}
    else if(!message2.ToString().Equals(""))&&(message1.ToString().Equals(""))
    {SuccMessage.Text=" Incorrect input in the "+message2.ToString();}
}
else if(!ssnText.Text.Length.Equals(11) ||(!ssnText.Text.Substring(3,1).Equals("-")
||(!ssnText.Text.Substring(6,1).Equals("-"))))
{SuccMessage.Text="Incorrect SSN format! ex 123-45-6789";}
else if(!phoneText.Text.Substring(0,1).Equals("(")||(!phoneText.Text.Substring(4,1).Equals(")"))
||(!phoneText.Text.Substring(8,1).Equals("-"))||(!phoneText.Text.Length.Equals(13)))
{ SuccMessage.Text=" Inorrect telephone formate! ex. (888)888-8888"; }
elseif(!emergencyPhoneText.Text.Substring(0,1).Equals("(")||
(!emergencyPhoneText.Text.Substring(4,1).Equals(")"))||
(!emergencyPhoneText.Text.Substring(8,1).Equals("-")

```

```

    "))))(emergencyPhoneText.Text.Length.Equals(13)))
{SuccMessage.Text=" Inorrect telephone formate! ex.(888)888-8888";}
else
{
    string str7_Check=ssnText.Text;
    //Check whether the patient information in the database or not
    //If the new patient's SSN has existed in the database, the user should check
    //whether the patient's information has already in the database or the patient provide
    // the incorreect SSN
    objConn= new SqlConnection(ConnStr);
    string ssnCheck="SELECT SSN FROM Patients WHERE SSN='"+str7_Check+"'";
    SqlCommand obComm=new SqlCommand(ssnCheck, objConn);
    Connect();
    SqlDataReader readSSN=obComm.ExecuteReader();
    try
    {
        if(readSSN.Read().ToString().Equals("True"))
        {
            Disconnection();
            readSSN.Close();
            SuccMessage.Text=" Incorrect input! The SSN has already existed in the
            database.";
            ssnText.Text="";
        }
        else
        {
            InsertIntoTable();
            Response.Redirect("MedicalHistoryNew.aspx");
        }
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
        Response.End();
    }
    finally
    {
        if (objConn.State == ConnectionState.Open)
        {
            Disconnection();
        }
        readSSN.Close();
    }
}
}
private void InsertIntoTable()
{
    str1=Date.Text;
    str2=lastNameText.Text;
    str3=mInitText.Text;
    str4=firstNameText.Text;
    str5=birthText.Text;
    str6=sexText.Text;
    str7=ssnText.Text;
    str8=driverLicenseText.Text;
    str9=maritalText.Text;
    str10=phoneText.Text;
    str11=e_mailText.Text;
    str12= addressText1.Text;
    str13=occupationText.Text;
    str14=businessPhoneText.Text;
    str15=employerText.Text;
    str16=businessAddressText.Text;
    str17=contactText.Text;
    str18=relationText.Text;
    str19=emergencyPhoneText.Text;
    str20=InsuranceCoText.Text;
}

```

```

str21=policyText.Text;
str22=IPhoneTxtBox.Text;
str23=billingAddress.Text;
// Change lower character to upper
if(martialText.Text=="s")
    martialText.Text="S";
if(martialText.Text=="d")
    martialText.Text="D";
if(martialText.Text=="m")
    martialText.Text="M";
if(sexText.Text=="f")
    sexText.Text="F";
if(sexText.Text=="m")
    sexText.Text="M";
objConn=new SqlConnection(ConnStr);
string insertStr="INSERT INTO Patients(RegDate, LastName, MInit, FirstName, Birthday, Sex, SSN,
DriverLicense, MartialStatus, Phone, E_mail, Address, Occupation, BusinessPhone,
Employer, BusinessAddress, ContactPerson, Relation, EmergencyPhone, InsuranceCo,
PolicyNo, InsurancePhone, BillingAddress) VALUES(""+str1+"",""+str2+"",""+str3+"",""+
str4+"",""+str5+"",""+str6+"",""+str7+"",""+str8+"",""+str9+"",""+str10+"",""+str11+"",""+str12+
",""+str13+"",""+str14+"",""+str15+"",""+str16+"",""+str17+"",""+str18+"",""+str19+"",""+str20+
",""+str21+"",""+str22+"",""+str23+""); SELECT @@IDENTITY As 'Identity';
SqlCommand CommObj=new SqlCommand(insertStr,objConn);
int iID=0;
try
{
    Connect();
    iID=Convert.ToInt32(CommObj.ExecuteScalar());
    No.Text=Convert.ToString(iID);
    lastName_NMH=str2;
    sex_NMH=str6;
    SSN_NMH=str7;
    firstName_NMH=str4;
    caseNo_NMH=Convert.ToString(iID);
    MInit_NMH=str3;
}
catch(Exception ex)
{
    Response.Write(ex.Message);
    Response.End();
}
finally
{
    if(objConn.State==ConnectionState.Open)
    {
        Disconnection();
    }
}
}
private void Connect()
{
    if(objConn==null)
        objConn= new SqlConnection(ConnStr);
    if(objConn.State==ConnectionState.Closed)
        objConn.Open();
}
private void Disconnection()
{
    objConn.Close();
}
}
}
}

```

## File: Diagnosis.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class Diagnosis : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Button addBtn;
        protected System.Web.UI.WebControls.Button saveBtn;
        protected System.Web.UI.WebControls.Label errMessage;
        protected System.Web.UI.WebControls.TextBox msgLine;
        protected static int caseID;
        protected string strConn1="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        protected System.Web.UI.WebControls.ListBox DiagnosisList;
        string str9, str7, ICDStr, str8;
        int k;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            caseID=Convert.ToInt32(LoginControl.CaseHistory.CaseHistoryNo);
            if(!IsPostBack)
            {
                SqlConnection objConn8=new SqlConnection(strConn1);
                string strSQL8="SELECT Keyno, Diagnosis FROM DiagnosisList1 order by Diagnosis";
                SqlCommand objCommandD=new SqlCommand(strSQL8, objConn8);
                objConn8.Open();
                DiagnosisList.DataSource=objCommandD.ExecuteReader();
                DiagnosisList.DataTextField="Diagnosis";
                DiagnosisList.DataValueField="Keyno";
                DiagnosisList.DataBind();
                objConn8.Close();
            }
        }
        private void addClick(object sender, System.EventArgs e)
        {
            str8="";
            if(DiagnosisList.SelectedIndex.ToString().Equals("-1"))
            {
                errMessage.Text="Please select one item from the ListBox before you click add button.{};
            }
            else
            {
                str8=DiagnosisList.SelectedItem.ToString();
                for(int i=0; i<str8.Length;i++)
                {
                    k=i+1;
                    str7=str8.Substring(i, 1);
                    if(!str7.ToString().Equals(" "))
                    {str9+=str7;}
                    else if(str7.ToString().Equals(" "))
                    {
                        if(str8.Substring(k,1).Equals(" "))
                        {break;}
                        else
                    }
                }
            }
        }
    }
}
```

```

        { str9+=" ";
    }
}
SqlConnection objConn9=new SqlConnection(strConn1);
string strSQL9="SELECT* FROM DiagnosisList1 WHERE Diagnosis='"+str9+"'";
objConn9.Open();
SqlDataAdapter ad=new SqlDataAdapter(strSQL9, objConn9);
DataSet objDset=new DataSet();
ad.Fill(objDset, "dtDiagnosis");
ICDStr=objDset.Tables["dtDiagnosis"].Rows[0][1].ToString();
msgLine.Text+="(ICD-9 "+ICDStr+": " +str9+", "+"\n");
}
}
private void saveClick(object sender, System.EventArgs e)
{
    int len;
    string str6,finalStr;
    if(msgLine.Text.Equals(""))
    {
        errMsg.Text="The multiline contains nothing. Please add diagnosis into multiline.";
    }
    else
    {
        string today=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
        DateTime.Now.Day,DateTime.Now.Year);
        SqlConnection objConn_Dg =new SqlConnection(strConn1);
        string Date_Dg="SELECT * FROM CaseHistory WHERE MainType='Diagnosis' and
        Date='"+today+"'and caseNo='"+caseID+"' ";
        SqlCommand obComm_Dg=new SqlCommand(Date_Dg, objConn_Dg);
        objConn_Dg.Open();
        SqlDataReader reader_Dg=objComm_Dg.ExecuteReader();
        try
        {
            if(reader_Dg.Read().ToString().Equals("True"))
            {
                objConn_Dg.Close();
                reader_Dg.Close();
                errMsg.Text="The diagnosis has existed already! To modify or edit it by click \'
                View Record \' button";
            }
            else
            {
                str6=msgLine.Text;
                len=str6.Length-1;
                if(str6.Substring(len,1).Equals(","))
                    finalStr=str6.Substring(0, len+1);
                else
                    finalStr=str6;
                SqlConnection obj8=new SqlConnection(strConn1);
                string todaydate=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
                DateTime.Now.Day,DateTime.Now.Year);
                string strSQL_D="INSERT INTO CaseHistory(CaseNo., Date, MainType,
                Diagnosis)VALUES('"+caseID+"', '"+todaydate+"', 'Diagnosis', '"+finalStr+"'"); SELECT
                @@IDENTITY AS 'identity';
                SqlCommand dbComm8=new SqlCommand(strSQL_D, obj8);
                if(!msgLine.Equals(""))
                {
                    int iID=0;
                    try
                    {
                        obj8.Open();
                        iID=Convert.ToInt32(dbComm8.ExecuteScalar());
                        errMsg.Text=" The selected items have saved successfully!";
                    }
                    catch(Exception ex)
                    {
                        Response.Write(ex.Message);
                        Response.End();
                    }
                }
            }
        }
    }
}

```



```

        if(UserName1.Equals("none"))
            Response.Redirect("DefaultPage.aspx");
        CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
        Label2.Text=CaseNum;
        CheckNo=Convert.ToInt32(CaseNum);
    }
    private void SelectedItemChange(object sender, System.EventArgs e)
    {
        InquiryDataGrid.CurrentPageIndex=Convert.ToInt32(0);
        if(TypeListBox.SelectedItem.Text.Equals("Chills and Fever"))
        {
            colStr="Chills and Fever";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Perspiration"))
        {
            colStr="Perspiration";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Pain Site"))
        {
            colStr="Pain Site";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Quality of Pain"))
        {
            colStr="Quality of Pain";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Symptoms of the Head"))
        {
            colStr="Symptoms of the Head-Dizziness";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Symptoms of the Ears"))
        {
            colStr="Symptoms of the Ears";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Eye Pain"))
        {
            colStr="Eye Pain";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Giddiness"))
        {
            colStr="Giddiness";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("General Symptoms"))
        {
            colStr="General Symptoms";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Symptoms of the Chest"))
        {
            colStr="Symptoms of Chest";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Symptoms of the Abdomen"))
        {
            colStr="Symptoms of Abdomen";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Disorders of Organs"))
        {
            colStr="Disorders of Organs";
        }
    }
}

```

```

        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("Sleep"))
    {
        colStr="Sleep";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("State of Diet"))
    {
        colStr="State of Diet";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("Sense of Taste"))
    {
        colStr="Sense of Taste";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Constipation"))
    {
        colStr="Defecation-Constipation";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Diarrhea"))
    {
        colStr="Defecation-Diarrhea";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Abnormal quality"))
    {
        colStr="Defecation-Abnormal quality";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Abnormal feeling"))
    {
        colStr="Defecation-Abnormal feeling";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("Urination"))
    {
        colStr="Urination";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Cycle"))
    {
        colStr="Menstruation-Cycle";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Amount"))
    {
        colStr="Menstruation-Amount";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Color"))
    {
        colStr="Menstruation-Color";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("** Adominal pain"))
    {
        colStr="Mensturation-Adominal pain";
        BindGrid();
    }
    else if(TypeListBox.SelectedItem.Text.Equals("Vaginal Discharge"))
    {
        colStr="Vaginal Discharge";
        BindGrid();
    }
}

```



```

else if(TypeListBox.SelectedItem.Text.Equals("Pregnancy and Childbirth"))
{
    colStr="Pregnancy and Childbirth";
    BindGrid();
}
}
private void BindGrid()
{
    objConn1=new SqlConnection(newConn1);
    strSQL1="SELECT InquiryNo, SubType, Symptoms, SuggestionOfDisorders FROM inquiry WHERE
    MainType='"+colStr+"'";
    //DataAdapter setup
    SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL1,objConn1);
    //Dataset & Adapter & Table
    DataSet objDataSet=new DataSet();
    objAdapter.Fill(objDataSet, "dtInquiry");
    InquiryDataGrid.DataSource=objDataSet.Tables["dtInquiry"];
    InquiryDataGrid.DataBind();
}
private void ChangePage(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    InquiryDataGrid.CurrentPageIndex=e.NewPageIndex;
    BindGrid();
}
private void ViewClick(object sender, System.EventArgs e)
{
    method2="Inquiry";
    Server.Transfer("ViewSelectedItems.aspx?CaseNo="+ CheckNo);
}
private void ProcessCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    string str1, str2, str3, str4;
    str1= e.CommandArgument.ToString();
    if (e.CommandName.Equals("GetProductDetails"))
    {
        strSQL5="SELECT InquiryNo,SubType, Symptoms, SuggestionOfDisorders FROM inquiry
        WHERE MainType='"+colStr+"' and InquiryNo='"+str1+"'";
        SqlConnection objConn5 =new SqlConnection(newConn1);
        objConn5.Open();
        SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL5, objConn5);
        DataSet objDataSet1=new DataSet("dtNewListen");
        objAdapter1.Fill(objDataSet1, "dtNewInquiry");
        DataTable tbl1=objDataSet1.Tables["dtNewInquiry"];
        tbl1.PrimaryKey=new DataColumn[] {tbl1.Columns["InquiryNo"]};
        index=Convert.ToInt32(str1);
        DataRow row=tbl1.Rows.Find(index);
        str2=row["SubType"].ToString();
        str3=row["SuggestionOfDisorders"].ToString();
        str4=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
        DateTime.Now.Day,DateTime.Now.Year);
        SqlConnection objConn6=new SqlConnection(newConn1);
        string InputStr1="INSERT INTO CaseHistory(CaseNo,MainType ,SubType,
        SuggestionOfDisorders, Date) VALUES('"+CaseNum+"','Inquiry','"+str2+"', '"+str3+"', '"+str4+"')";
        SELECT @@IDENTITY As 'Identity';
        SqlCommand dbComm1=new SqlCommand(InputStr1, objConn6);
        int iID=0;
        try
        {
            objConn6.Open();
            iID = Convert.ToInt32( dbComm1.ExecuteScalar());
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
            Response.End();
        }
    }
    finally
}

```



```

else if(str1.Equals("Triple warmer, sanjiao channel of hand shaoyang (手少陽三焦經)"))
    Response.Redirect("ShowImage10.aspx");
else if(str1.Equals("Gallbladder channel of foot shaoyang (足少陽膽經)"))
    Response.Redirect("ShowImage11.aspx");
else if(str1.Equals("Liver channel of foot jueyin(足厥陰肝經)"))
    Response.Redirect("ShowImage12.aspx");
else if(str1.Equals("Du channel(督脈)"))
    Response.Redirect("ShowImage13.aspx");
else if(str1.Equals("Ren channel(任脈)"))
    Response.Redirect("ShowImage14.aspx");
    }
}
}

```

### File:ListenSmell.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class ListenSmell : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.Label caseNo;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.ListBox TypeListBox;
        protected System.Web.UI.WebControls.Button ViewSelectBtn;
        protected System.Web.UI.HtmlControls.HtmlForm UserAccount;
        protected System.Web.UI.WebControls.DataGrid AODataGrid;
        protected static string colStr;
        protected static string CaseNum;
        protected static int CheckNo;
        private string newConn1="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        private SqlConnection objConn1;
        string strSQL1;
        string strSQL5;
        public static string method3="";
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
            Label2.Text=CaseNum;
            CheckNo=Convert.ToInt32(CaseNum);
        }
        private void SelectedItemChange(object sender, System.EventArgs e)
        {
            AODataGrid.CurrentPageIndex=Convert.ToInt32(0);
            if(TypeListBox.SelectedItem.Text.Equals("Breath"))
            {
                colStr="Breath";
                BindGrid();
            }
        }
    }
}

```

```

}
else if(TypeListBox.SelectedItem.Text.Equals("Bronchial Wheezing"))
{
    colStr="Bronchial Wheezing";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Belching"))
{
    colStr="Belching";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Coughing"))
{
    colStr="Coughing";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Dysphasia"))
{
    colStr="Dysphasia";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Delirium"))
{
    colStr="Delirium";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Fading Murmuring"))
{
    colStr="Fading Murmuring";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Hiccuping"))
{
    colStr="Hiccuping";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Insufficient Qi"))
{
    colStr="Insufficient Qi";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Odor"))
{
    colStr="odor";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Paraphasia"))
{
    colStr="Paraphasia";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Raving"))
{
    colStr="Raving";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Speech"))
{
    colStr="Speech";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Sweat"))
{
    colStr="Sweat";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Vomiting"))

```

```

    {
        colStr="Vomiting";
        BindGrid();
    }
}
private void BindGrid()
{
    objConn1=new SqlConnection(newConn1);
    strSQL1="SELECT ListenSmellNo, SubType, Symptoms, SuggestionOfDisorders FROM ListenSmall
    WHERE MainType='"+colStr+"'";
    //DataAdapter setup
    SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL1,objConn1);
    //Dataset & Adapter & Table
    DataSet objDataSet=new DataSet();
    objAdapter.Fill(objDataSet, "dtListenSmell");
    AODataGrid.DataSource=objDataSet.Tables["dtListenSmell"];
    AODataGrid.DataBind();
}
private void ChangePage(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    AODataGrid.CurrentPageIndex=e.NewPageIndex;
    BindGrid();
}
private void ViewClick(object sender, System.EventArgs e)
{
    method3="ListenSmell";
    Server.Transfer("ViewSelectedItems.aspx?CaseNo=" + CheckNo);
}
private void ProcessCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    string str1, str2, str3, str4;
    int index;
    str1= e.CommandArgument.ToString();
    if (e.CommandName.Equals("GetListenDetails"))
    {
        strSQL5="SELECT ListenSmellNo, SubType, Symptoms, SuggestionOfDisorders FROM
        ListenSmall WHERE MainType='"+colStr+"' and ListenSmellNo='"+str1+"'";
        SqlConnection objConn5 =new SqlConnection(newConn1);
        objConn5.Open();
        SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL5, objConn5);
        DataSet objDataSet1=new DataSet("dtNewListen");
        objAdapter1.Fill(objDataSet1, "dtNewListen");
        DataTable tbl1=objDataSet1.Tables["dtNewListen"];
        tbl1.PrimaryKey=new DataColumn[] {tbl1.Columns["ListenSmellNo"]};
        index=Convert.ToInt32(str1);
        DataRow row=tbl1.Rows.Find(index);
        str2=row["SubType"].ToString();
        str3=row["SuggestionOfDisorders"].ToString();
        str4=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
        DateTime.Now.Day,DateTime.Now.Year);
        SqlConnection objConn6=new SqlConnection(newConn1);
        string InputStr1="INSERT INTO CaseHistory(CaseNo,MainType ,SubType,
        SuggestionOfDisorders, Date) VALUES('"+CaseNum+"','Auscultation and Olfaction',"
        +str2+"', '"+str3+"', '"+str4+"'); SELECT @@IDENTITY As 'Identity'";
        SqlCommand dbComm1=new SqlCommand(InputStr1, objConn6);
        int iID=0;
        try
        {
            objConn6.Open();
            iID = Convert.ToInt32( dbComm1.ExecuteScalar());
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
            Response.End();
        }
    }
    finally
}

```



```

}
else if(TypeListBox.SelectedItem.Text.Equals("Physique"))
{
    colStr="Physique";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Patterns of Movement"))
{
    colStr="Patterns of Movement";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Head"))
{
    colStr="Head";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Hair"))
{
    colStr="Hair";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Eyes"))
{
    colStr="Eyes";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Ears"))
{
    colStr="Ears";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Nose"))
{
    colStr="Nose";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Lip"))
{
    colStr="Lip";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Throat"))
{
    colStr="Throat";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Teeth and Gum"))
{
    colStr="Teeth and Gum";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Neck"))
{
    colStr="Neck";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("External Genitals and Anus"))
{
    colStr="External Genitals and Anus";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Skin"))
{
    colStr="Skin";
    BindGrid();
}
else if(TypeListBox.SelectedItem.Text.Equals("Color of Tongue"))

```

```

        {
            colStr="Color of Tongue";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Tongue Picture"))
        {
            colStr="Tongue Picture";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("The Patterns of Tongue"))
        {
            colStr="The Patterns of Tongue";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Tongue Coating"))
        {
            colStr="Tongue Coating";
            BindGrid();
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Physique"))
        {
            colStr="Physique";
            BindGrid();
        }
    }
}
private void BindGrid()
{
    objConn1=new SqlConnection(newConn1);
    strSQL1="SELECT LookNo, SubType, Symptoms, SuggestionOfDisorders FROM Look WHERE
    MainType='"+colStr+"'";
    //DataAdapter setup
    SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL1,objConn1);
    //Dataset & Adapter & Table
    DataSet objDataSet=new DataSet();
    objAdapter.Fill(objDataSet, "dtLook");
    LookDataGrid.DataSource=objDataSet.Tables["dtLook"];
    LookDataGrid.DataBind();
}
private void ChangePage(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    LookDataGrid.CurrentPageIndex=e.NewPageIndex;
    BindGrid();
}
private void ViewClick(object sender, System.EventArgs e)
{
    method1="Look";
    Server.Transfer("ViewSelectedItems.aspx?CaseNo=" + CheckNo);
}
private void ProcessCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    string str1, str2, str3, str4;
    int index;
    str1= e.CommandArgument.ToString();
    if (e.CommandName.Equals("GetLookDetails"))
    {
        strSQL5="SELECT LookNo,SubType, Symptoms, SuggestionOfDisorders FROM Look WHERE
        MainType='"+colStr+"' and LookNo='"+str1+"'";
        SqlConnection objConn5 =new SqlConnection(newConn1);
        objConn5.Open();
        SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL5, objConn5);
        DataSet objDataSet1=new DataSet("dtNewLook");
        objAdapter1.Fill(objDataSet1, "dtNewLook");
        DataTable tbl1=objDataSet1.Tables["dtNewLook"];
        tbl1.PrimaryKey=new DataColumn[] {tbl1.Columns["LookNo"]};
        index=Convert.ToInt32(str1);
        DataRow row=tbl1.Rows.Find(index);
        str2=row["SubType"].ToString();
    }
}

```





```

protected System.Web.UI.WebControls.Label Q2Label;
protected System.Web.UI.WebControls.Label Label4;
protected System.Web.UI.WebControls.Label Label1;
protected System.Web.UI.WebControls.TextBox M1Txt;
protected System.Web.UI.WebControls.Label dose1;
protected System.Web.UI.WebControls.TextBox D1Txt;
protected System.Web.UI.WebControls.Label Label5;
protected System.Web.UI.WebControls.TextBox F1Txt;
protected System.Web.UI.WebControls.Label Label7;
protected System.Web.UI.WebControls.TextBox M2Txt;
protected System.Web.UI.WebControls.Label Label8;
protected System.Web.UI.WebControls.TextBox D2Txt;
protected System.Web.UI.WebControls.Label Label9;
protected System.Web.UI.WebControls.TextBox F2Txt;
protected System.Web.UI.WebControls.Label Label10;
protected System.Web.UI.WebControls.TextBox M3Txt;
protected System.Web.UI.WebControls.Label Label11;
protected System.Web.UI.WebControls.TextBox D3Txt;
protected System.Web.UI.WebControls.Label Label12;
protected System.Web.UI.WebControls.TextBox F3Txt;
protected System.Web.UI.WebControls.Label Label13;
protected System.Web.UI.WebControls.TextBox M4Txt;
protected System.Web.UI.WebControls.Label Label14;
protected System.Web.UI.WebControls.TextBox D4Txt;
protected System.Web.UI.WebControls.Label Label15;
protected System.Web.UI.WebControls.TextBox F4Txt;
protected System.Web.UI.WebControls.Label Label16;
protected System.Web.UI.WebControls.TextBox M5Txt;
protected System.Web.UI.WebControls.Label Label17;
protected System.Web.UI.WebControls.TextBox D5Txt;
protected System.Web.UI.WebControls.Label Label18;
protected System.Web.UI.WebControls.TextBox F5Txt;
protected System.Web.UI.WebControls.Label Label19;
protected System.Web.UI.WebControls.TextBox M6Txt;
protected System.Web.UI.WebControls.Label Label20;
protected System.Web.UI.WebControls.TextBox D6Txt;
protected System.Web.UI.WebControls.Label Label21;
protected System.Web.UI.WebControls.TextBox F6Txt;
protected System.Web.UI.WebControls.Label Label6;
protected System.Web.UI.WebControls.Label IllnessList;
protected System.Web.UI.WebControls.Label Label23;
protected System.Web.UI.WebControls.TextBox SergeryTxt;
protected System.Web.UI.WebControls.Label Label24;
protected System.Web.UI.WebControls.Label Q5Label;
protected System.Web.UI.WebControls.Label Label25;
protected System.Web.UI.WebControls.Label Q6Label;
protected System.Web.UI.WebControls.Label Label26;
protected System.Web.UI.WebControls.Label Q7Label;
protected System.Web.UI.WebControls.Label Label27;
protected System.Web.UI.WebControls.Label Q8Label;
protected System.Web.UI.WebControls.Label Label28;
protected System.Web.UI.WebControls.TextBox PainTxt;
protected System.Web.UI.WebControls.Button exitButton;
private void Page_Load(object sender, System.EventArgs e)
{
    string chTest1="1";
    string chTest2="2" ;
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    chTest1=Convert.ToString(LoginControl.UserControl.AcupName);
    chTest2=Convert.ToString(LoginControl.UserControl.staffName);
    if(!chTest1.Equals("1"))
        UserName1=chTest1;
    if(!chTest2.Equals("2"))
        UserName1=chTest2;
    if(UserName1.Equals("none"))
        Response.Redirect("UseContral.aspx");
}

```

## File:MedicalHistoryReport.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class MedicalHistoryReport : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label CaseNo;
        protected System.Web.UI.WebControls.TextBox casenoTxt;
        protected System.Web.UI.WebControls.Label Gender;
        protected System.Web.UI.WebControls.TextBox sexTxt;
        protected System.Web.UI.WebControls.Label Date;
        protected System.Web.UI.WebControls.Label ssnLabel;
        protected System.Web.UI.WebControls.TextBox ssnTxt;
        protected System.Web.UI.WebControls.Label fnameLabel;
        protected System.Web.UI.WebControls.TextBox FNameTxt;
        protected System.Web.UI.WebControls.Label Mlnit;
        protected System.Web.UI.WebControls.TextBox MTxt;
        protected System.Web.UI.WebControls.Label lnameLabel;
        protected System.Web.UI.WebControls.TextBox LNameTxt;
        string strConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        private SqlConnection objConn_MHR;
        string q1, q2, q5, q6, q7, q8;
        string MedStr;
        private string MedTable="dtMedicalHistory";
        protected System.Web.UI.WebControls.Label MHLabel1;
        protected System.Web.UI.WebControls.Label Q1Label;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.TextBox ReasonTxt;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.Label Q2Label;
        protected System.Web.UI.WebControls.Label Label4;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.TextBox M1Txt;
        protected System.Web.UI.WebControls.Label dose1;
        protected System.Web.UI.WebControls.TextBox D1Txt;
        protected System.Web.UI.WebControls.Label Label5;
        protected System.Web.UI.WebControls.TextBox F1Txt;
        protected System.Web.UI.WebControls.Label Label7;
        protected System.Web.UI.WebControls.TextBox M2Txt;
        protected System.Web.UI.WebControls.Label Label8;
        protected System.Web.UI.WebControls.TextBox D2Txt;
        protected System.Web.UI.WebControls.Label Label9;
        protected System.Web.UI.WebControls.TextBox F2Txt;
        protected System.Web.UI.WebControls.Label Label10;
        protected System.Web.UI.WebControls.TextBox M3Txt;
        protected System.Web.UI.WebControls.Label Label11;
        protected System.Web.UI.WebControls.TextBox D3Txt;
        protected System.Web.UI.WebControls.Label Label12;
        protected System.Web.UI.WebControls.TextBox F3Txt;
        protected System.Web.UI.WebControls.Label Label13;
        protected System.Web.UI.WebControls.TextBox M4Txt;
        protected System.Web.UI.WebControls.Label Label14;
        protected System.Web.UI.WebControls.TextBox D4Txt;
        protected System.Web.UI.WebControls.Label Label15;
        protected System.Web.UI.WebControls.TextBox F4Txt;
        protected System.Web.UI.WebControls.Label Label16;
```

```

protected System.Web.UI.WebControls.TextBox M5Txt;
protected System.Web.UI.WebControls.Label Label17;
protected System.Web.UI.WebControls.TextBox D5Txt;
protected System.Web.UI.WebControls.Label Label18;
protected System.Web.UI.WebControls.TextBox F5Txt;
protected System.Web.UI.WebControls.Label Label19;
protected System.Web.UI.WebControls.TextBox M6Txt;
protected System.Web.UI.WebControls.Label Label20;
protected System.Web.UI.WebControls.TextBox D6Txt;
protected System.Web.UI.WebControls.Label Label21;
protected System.Web.UI.WebControls.TextBox F6Txt;
protected System.Web.UI.WebControls.Label Label23;
protected System.Web.UI.WebControls.TextBox SurgeryTxt;
protected System.Web.UI.WebControls.Label Label24;
protected System.Web.UI.WebControls.Label Q5Label;
protected System.Web.UI.WebControls.Label Label25;
protected System.Web.UI.WebControls.Label Q6Label;
protected System.Web.UI.WebControls.Label Label26;
protected System.Web.UI.WebControls.Label Q7Label;
protected System.Web.UI.WebControls.Label Label27;
protected System.Web.UI.WebControls.Label Q8Label;
protected System.Web.UI.WebControls.Label Label28;
protected System.Web.UI.WebControls.TextBox PainTxt;
protected System.Web.UI.WebControls.Label Label6;
protected System.Web.UI.WebControls.Label IllnessList;
protected System.Web.UI.WebControls.Label EMsg;
protected System.Web.UI.WebControls.Label titleLabel;
private string mdfStr="";
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
    getPatientMedicalHistory();
}
private void getPatientMedicalHistory()
{
    string caseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
    int case1=Convert.ToInt32(caseNum);
    SqlConnection objConn_Rd =new SqlConnection(strConn);
    string strQuery_MHR="SELECT * FROM MedicalHistory WHERE CaseNo='"+caseNum+"'";
    SqlCommand obComm_MHR=new SqlCommand(strQuery_MHR, objConn_Rd);
    objConn_Rd.Open();
    SqlDataReader Medical_rd=obComm_MHR.ExecuteReader();
    if(Medical_rd.Read().ToString().Equals("False"))
    {
        objConn_Rd.Close();
        casenoTxt.Text=caseNum;
        EMsg.Text="Patient's Medical History does not exist.";
    }
    else
    {
        objConn_Rd.Close();
        objConn_MHR =new SqlConnection(strConn);
        objConn_MHR.Open();
        SqlDataAdapter objAdapter_MHR=new SqlDataAdapter(strQuery_MHR, objConn_MHR);
        DataSet objDataSet_MHR=new DataSet();
        objAdapter_MHR.Fill(objDataSet_MHR, MedTable);
        objConn_MHR.Close();
        //From the tabel to get patient's medical history information
        DataTable tbl_MHR=new DataTable(MedTable);
        casenoTxt.Text=caseNum;
        Date.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["RecordDate"].ToString();
        FNameTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["FirstName"].ToString();
        sexTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["Sex"].ToString();
    }
}

```

```

MTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["Minit"].ToString();
ssnTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["SSN"].ToString();
LNameTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["LastName"].ToString();
ReasonTxt.Text=objDataSet_MHR.Tables[MedTable].Rows[0]["Q1_5"].ToString();
q1=objDataSet_MHR.Tables[MedTable].Rows[0]["Q1"].ToString();
q2=objDataSet_MHR.Tables[MedTable].Rows[0]["Q2"].ToString();
q5=objDataSet_MHR.Tables[MedTable].Rows[0]["Q5"].ToString();
q6=objDataSet_MHR.Tables[MedTable].Rows[0]["Q6"].ToString();
q7=objDataSet_MHR.Tables[MedTable].Rows[0]["Q7"].ToString();
q8=objDataSet_MHR.Tables[MedTable].Rows[0]["Q8"].ToString();
MedStr=objDataSet_MHR.Tables[MedTable].Rows[0]["Q2_5"].ToString();
Q1Label.Text=q1;
Q2Label.Text=q2;
Q5Label.Text=q5;
Q6Label.Text=q6;
Q7Label.Text=q7;
Q8Label.Text=q8;
string[] mStr={,,,,,,,,,};
string[] dStr={,,,,,,,,,};
string[] fStr={,,,,,,,,,};
int i;
int y=0;
int z=0;
int w=0;
int j=0;
int k=0;
int t=0;
char u='Y';
for( i=0; i<MedStr.Length; i++)
{
    mdfStr+=MedStr.Substring(i,1);
    if((MedStr.Substring(i,1).Equals(".")))
    {
        if(u=='Y')
        {
            mStr[0]=MedStr.Substring(0, i);// get Medication
            y=i+1;
            u='N';
        }
        if(u=='N')
        {
            mStr[j]=MedStr.Substring(w, i-w);// get Medication
            j++;
            y=i+1;
        }
        mdfStr="";
    }
    else if(MedStr.Substring(i,1).Equals(","))
    {
        dStr[k]=MedStr.Substring(y, i-y);
        k++;
        z=i+1;
        mdfStr="";
    }
    else if(MedStr.Substring(i, 1).Equals(";"))
    {
        fStr[t]=MedStr.Substring(z, i-z);
        t++;
        w=i+1;
        mdfStr="";
    }
}
M1Txt.Text=mStr[0];
M2Txt.Text=mStr[1];
M3Txt.Text=mStr[2];
M4Txt.Text=mStr[3];
M5Txt.Text=mStr[4];

```



```

        Response.Redirect("DefaultPage.aspx");
        CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
        Label2.Text=CaseNum;
        CheckNo=Convert.ToInt32(CaseNum);
    }
    private void SelectedItemChange(object sender, System.EventArgs e)
    {
        PTDataGrid.CurrentPageIndex=Convert.ToInt32(0);
        if(TypeListBox.SelectedItem.Text.Equals("Skin"))
        {
            test=0;
            colStr="Skin";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Chest and Abdomen"))
        {
            test=0;
            colStr="Chest and Abdomen";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Acupoints"))
        {
            test=0;
            colStr="Acupoints";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Hands and Feet"))
        {
            test=0;
            colStr="Hands and Feet";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Floating Pulse"))
        {
            test=1;
            colStr1="Floating Pulse";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Deep Pulse"))
        {
            test=1;
            colStr1="Deep Pulse";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Slow Pulse"))
        {
            test=1;
            colStr1="Slow Pulse";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Rapid Pulse"))
        {
            test=1;
            colStr1="Rapid Pulse";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Deficient Pulse"))
        {
            test=1;
            colStr1="Deficient Pulse";
            BindGrid(test);
        }
        else if(TypeListBox.SelectedItem.Text.Equals("Excessive Pulse"))
        {
            test=1;
            colStr1="Excessive Pulse";
            BindGrid(test);
        }
    }

```

```

    }
    else if(TypeListBox.SelectedItem.Text.Equals("--Palpation--"))
    {
        test=2;
        colStr2="Palpation";
    }
    else if(TypeListBox.SelectedItem.Text.Equals("--Pulse-Taking--"))
    {
        test=3;
        colStr2="Pulse-Taking";
    }
}
private void BindGrid(int T)
{
    objConn1=new SqlConnection(newConn1);
    if(T==0)
    {
        strSQL1="SELECT PNo, SubType, Symptoms, SuggestionOfDisorders FROM Palpation WHERE
        MainType='"+colStr+''";
        //DataAdapter setup
        SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL1,objConn1);
        //Dataset & Adapter & Table
        DataSet objDataSet=new DataSet();
        objAdapter.Fill(objDataSet, "dtPalpation");
        PTDataGrid.DataSource=objDataSet.Tables["dtPalpation"];
        PTDataGrid.DataBind();
    }
    else if(T==1)
    {
        strSQL2="SELECT PNo, SubType, Symptoms, SuggestionOfDisorders FROM TypesOfPulse
        WHERE MainType='"+colStr+''";
        //DataAdapter setup
        SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL2,objConn1);
        //Dataset & Adapter & Table
        DataSet objDataSet1=new DataSet();
        objAdapter1.Fill(objDataSet1, "dtPulse");
        PTDataGrid.DataSource=objDataSet1.Tables["dtPulse"];
        PTDataGrid.DataBind();
    }
}
private void ChangePage(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    PTDataGrid.CurrentPageIndex=e.NewPageIndex;
    BindGrid(test);
}
private void ViewClick(object sender, System.EventArgs e)
{
    if((test==2)||test==3)
    { ckStr=colStr2;}
    method4="Palpation";
    Server.Transfer("ViewSelectedItems.aspx?CaseNo=" + CheckNo);
}
private void ProcessCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    string str1, str2, str3, str4;
    int index;
    str1 = e.CommandArgument.ToString();
    if ((e.CommandName.Equals("GetPalpationDetails"))&&(test==0))
    {
        strSQL5="SELECT PNo, SubType, Symptoms, SuggestionOfDisorders FROM Palpation WHERE
        MainType='"+colStr+" and PNo='"+str1+''";
        SqlConnection objConn5 =new SqlConnection(newConn1);
        objConn5.Open();
        SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL5, objConn5);
        DataSet objDataSet1=new DataSet("dtPalpation");
        objAdapter1.Fill(objDataSet1, "dtPalpation");
        DataTable tbl1=objDataSet1.Tables["dtPalpation"];
    }
}

```



```

tbl1.PrimaryKey=new DataColumn[] {tbl1.Columns["PNo"]};
index=Convert.ToInt32(str1);
DataRow row=tbl1.Rows.Find(index);
str2=row["SubType"].ToString();
str3=row["SuggestionOfDisorders"].ToString();
str4=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
    DateTime.Now.Day,DateTime.Now.Year);
SqlConnection objConn6=new SqlConnection(newConn1);
string InputStr1="INSERT INTO CaseHistory(CaseNo,MainType ,SubType,
    SuggestionOfDisorders, Date) VALUES('"+CaseNum+"','Palpation','"+str2+"', '"+
    str3+"', '"+str4+"'); SELECT @@IDENTITY As 'Identity'";
SqlCommand dbComm1=new SqlCommand(InputStr1, objConn6);
int iID=0;
try
{
    objConn6.Open();
    iID = Convert.ToInt32( dbComm1.ExecuteScalar());
    ckStr="Palpation";
}
catch (Exception ex)
{
    Response.Write(ex.Message);
    Response.End();
}
finally
{
    if (objConn6.State == ConnectionState.Open)
        objConn6.Close();
    if(objConn5.State == ConnectionState.Open)
        objConn5.Close();
}
}
else if((e.CommandName.Equals("GetPalpationDetails"))&&(test==1))
{
    strSQL6="SELECT PNo, SubType, Symptoms, SuggestionOfDisorders FROM TypesOfPulse
    WHERE MainType='"+colStr1+"' and PNo='"+str1+"'";
    SqlConnection objConn7 =new SqlConnection(newConn1);
    objConn7.Open();
    SqlDataAdapter objAdapter2=new SqlDataAdapter(strSQL6, objConn7);
    DataSet objDataSet2=new DataSet("dtPulse");
    objAdapter2.Fill(objDataSet2, "dtPulse");
    DataTable tbl2=objDataSet2.Tables["dtPulse"];
    tbl2.PrimaryKey=new DataColumn[] {tbl2.Columns["PNo"]};
    index=Convert.ToInt32(str1);
    DataRow row=tbl2.Rows.Find(index);
    str2=row["SubType"].ToString();
    str3=row["SuggestionOfDisorders"].ToString();
    str4=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
        DateTime.Now.Day,DateTime.Now.Year);
    SqlConnection objConn8=new SqlConnection(newConn1);
    string InputStr2="INSERT INTO CaseHistory(CaseNo,MainType ,SubType,
        SuggestionOfDisorders, Date) VALUES('"+CaseNum+"','Pulse-Taking','"+str2+"', '"+str3+"',
        '"+str4+"'); SELECT @@IDENTITY As 'Identity'";
    SqlCommand dbComm2=new SqlCommand(InputStr2, objConn8);
    int iID=0;
    try
    {
        objConn8.Open();
        iID = Convert.ToInt32( dbComm2.ExecuteScalar());
        ckStr="Pulse-Taking";
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
        Response.End();
    }
    finally

```

```

        {
            if (objConn8.State == ConnectionState.Open)
                objConn8.Close();
            if(objConn7.State == ConnectionState.Open)
                objConn7.Close();
        }
    }
}
}
}
}

```

### File: PatientDetail.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class PatientDetails : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label caseNoLabel;
        protected System.Web.UI.WebControls.Label No;
        protected System.Web.UI.WebControls.Label LastName;
        protected System.Web.UI.WebControls.TextBox lastNameText;
        protected System.Web.UI.WebControls.Label Minit;
        protected System.Web.UI.WebControls.TextBox mInitText;
        protected System.Web.UI.WebControls.Label FirstNameLabel;
        protected System.Web.UI.WebControls.TextBox firstNameText;
        protected System.Web.UI.WebControls.Label Birthdate;
        protected System.Web.UI.WebControls.TextBox birthText;
        protected System.Web.UI.WebControls.Label Sex;
        protected System.Web.UI.WebControls.TextBox sexText;
        protected System.Web.UI.WebControls.Label SSN;
        protected System.Web.UI.WebControls.TextBox ssnText;
        protected System.Web.UI.WebControls.Label DriverLicense;
        protected System.Web.UI.WebControls.TextBox driverLicenseText;
        protected System.Web.UI.WebControls.Label MartialStatus;
        protected System.Web.UI.WebControls.TextBox martialText;
        protected System.Web.UI.WebControls.Label Phone;
        protected System.Web.UI.WebControls.TextBox phoneText;
        protected System.Web.UI.WebControls.Label E_mail;
        protected System.Web.UI.WebControls.TextBox e_mailText;
        protected System.Web.UI.WebControls.Label Address;
        protected System.Web.UI.WebControls.TextBox addressText1;
        protected System.Web.UI.WebControls.Label Occupation;
        protected System.Web.UI.WebControls.TextBox occupationText;
        protected System.Web.UI.WebControls.Label BusinessPhone;
        protected System.Web.UI.WebControls.TextBox businessPhoneText;
        protected System.Web.UI.WebControls.Label Employer;
        protected System.Web.UI.WebControls.TextBox employerText;
        protected System.Web.UI.WebControls.Label BusinessAddress;
        protected System.Web.UI.WebControls.TextBox businessAddressText;
        protected System.Web.UI.WebControls.Label Contact;
        protected System.Web.UI.WebControls.TextBox contactText;
        protected System.Web.UI.WebControls.Label Relation;
        protected System.Web.UI.WebControls.TextBox relationText;
        protected System.Web.UI.WebControls.Label EmergencyPhone;
        protected System.Web.UI.WebControls.TextBox emergencyPhoneText;
    }
}

```

```

protected System.Web.UI.WebControls.Label PolicyNo;
protected System.Web.UI.WebControls.TextBox policyText;
protected System.Web.UI.WebControls.Label InsuranceCoLabel;
protected System.Web.UI.WebControls.TextBox InsuranceCoText;
protected System.Web.UI.WebControls.Label InsPhoneLabel;
protected System.Web.UI.WebControls.TextBox IPhone TxtBox;
protected System.Web.UI.WebControls.Label billAddLabel;
protected System.Web.UI.WebControls.TextBox billingAddress;
protected System.Web.UI.WebControls.Label PatientInflLabel;
private string strConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
string CaseNum;
int CaseID;
string FStr, LStr, SSNStr,sexStr,miniStr, BirthStr, driverLicenseStr, martialStr, phoneStr,EmailStr;
string addressStr, occupationStr, businessPhoneStr, employerStr, businessAddressStr;
string emergencyPhoneStr,contactStr, relationStr, insuranceCoStr, policyStr, billingAddressStr,
    InsurancePhoneStr;
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
    CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
    CaseID=LoginControl.CaseHistory.CaseHistoryNo;
    SqlConnection objConn_P =new SqlConnection(strConn);
    string PInfo="SELECT * FROM Patients WHERE caseNo='"+CaseNum+"'";
    objConn_P.Open();
    SqlDataAdapter objAdapter_P=new SqlDataAdapter(PInfo, objConn_P);
    DataSet objDataSet_P=new DataSet();
    objAdapter_P.Fill(objDataSet_P, "dtSelectedPatientInfo");
    objConn_P.Close();
    //From the tabel to get patient information
    DataTable tbl_P=objDataSet_P.Tables["dtSelectedPatientInfo"];
    tbl_P.PrimaryKey=new DataColumn[]{tbl_P.Columns["CaseNo"]};
    DataRow row_P=tbl_P.Rows.Find(CaseID);
    FStr=row_P["FirstName"].ToString();
    LStr=row_P["LastName"].ToString();
    SSNStr=row_P["SSN"].ToString();
    sexStr=row_P["Sex"].ToString();
    miniStr=row_P["Minit"].ToString();
    BirthStr=row_P["Birthday"].ToString();
    driverLicenseStr=row_P["DriverLicense"].ToString();
    martialStr=row_P["MartialStatus"].ToString();
    phoneStr=row_P["Phone"].ToString();
    EmailStr=row_P["E_mail"].ToString();
    addressStr=row_P["Address"].ToString();
    occupationStr=row_P["Occupation"].ToString();
    businessPhoneStr=row_P["BusinessPhone"].ToString();
    employerStr=row_P["Employer"].ToString();
    businessAddressStr=row_P["BusinessAddress"].ToString();
    emergencyPhoneStr=row_P["EmergencyPhone"].ToString();
    contactStr=row_P["ContactPerson"].ToString();
    relationStr=row_P["Relation"].ToString();
    insuranceCoStr=row_P["InsuranceCo"].ToString();
    policyStr=row_P["PolicyNo"].ToString();
    billingAddressStr=row_P["BillingAddress"].ToString();
    InsurancePhoneStr=row_P["InsurancePhone"].ToString();
    //Put patient information into textBox
    lastNameText.Text=LStr;
    firstNameText.Text=FStr;
    ssnText.Text=SSNStr;
    No.Text=CaseNum;
    mInitText.Text=miniStr;
    birthText.Text=BirthStr;
    sexText.Text=sexStr;
    driverLicenseText.Text=driverLicenseStr;

```

```

        martialText.Text=martialStr;
        phoneText.Text=phoneStr;
        e_mailText.Text=EmailStr;
        addressText1.Text=addressStr;
        occuptionText.Text=occuptionStr;
        businessPhoneText.Text=businessPhoneStr;
        employerText.Text=employerStr;
        businessAddressText.Text=businessAddressStr;
        emergencyPhoneText.Text=emergencyPhoneStr;
        contactText.Text=contactStr;
        relationText.Text=relationStr;
        InsuranceCoText.Text=insuranceCoStr;
        policyText.Text=policyStr;
        IPhoneTxtBox.Text=billingAdressStr;
        billingAddress.Text=InsurancePhoneStr;
    }
}

```

### File:Prescription.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class Prescription : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.TextBox searchTextBox;
        protected System.Web.UI.WebControls.Button searchBtn;
        protected System.Web.UI.WebControls.Label instruLabel;
        protected System.Web.UI.WebControls.ListBox HerbList;
        protected System.Web.UI.WebControls.Button addBtn;
        protected System.Web.UI.WebControls.Button saveBtn;
        protected System.Web.UI.WebControls.TextBox msgLine;
        protected System.Web.UI.WebControls.Label errMessage;
        private int strLength;
        private string str1,str2, str3, str4, str5,str6;
        string caseID;
        private string strConn="server=(local)\NetSDK; database=ClinicSystem1; integrated security=true";
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            caseID=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
        }
        private void searchClick(object sender, System.EventArgs e)
        {
            errMessage.Text="";
            if(searchTextBox.Text=="")
            {
                errMessage.Text="Please enters a letter of the alphabet or the stroke of first Chinese character!"
            }
            else
            {
                string strConnection="server=(local)\NetSDK; database=ClinicSystem1; integrated security=true";

```

```

SqlConnection objConnection=new SqlConnection(strConnection);
str2=searchTextBox.Text;
str1=searchTextBox.Text.Substring(0,1);
strLength=str2.Length;
if(str1.Equals("0")||str1.Equals("1")|| str1.Equals("2")||
   str1.Equals("3")||str1.Equals("4")||str1.Equals("5")||str1.Equals("6")||str1.Equals("7")||
   str1.Equals("8")|| str1.Equals("9"))
{
    if(str1.Equals("1"))
    {
        if(strLength==1)
        {
            string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
            WHERE Strokes="+str2+"";
            SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
            objConnection);
            objConnection.Open();
            HerbList.DataSource=objCommandU.ExecuteReader();
            HerbList.DataTextField="ChineseHerbs";
            HerbList.DataValueField="SerialNo";
            HerbList.DataBind();
            errMessage.Text="There is no data in the herbs' list through stroke "+ str2;
        }
        else if (strLength==2)
        {
            if(searchTextBox.Text.Substring(0,2).Equals("10"))
            {
                string strSQLforListboxI= "SELECT SerialNo, ChineseHerbs FROM
                HerbsList WHERE Strokes=10";
                SqlCommand objCommandI =new SqlCommand(strSQLforListboxI,
                objConnection);
                objConnection.Open();
                HerbList.DataSource=objCommandI.ExecuteReader();
                HerbList.DataTextField="ChineseHerbs";
                HerbList.DataValueField="SerialNo";
                HerbList.DataBind();
            }
            else if(searchTextBox.Text.Substring(0,2).Equals("11"))
            {
                string strSQLforListboxJ= "SELECT SerialNo, ChineseHerbs FROM
                HerbsList WHERE Strokes=11";
                SqlCommand objCommandJ =new SqlCommand(strSQLforListboxJ,
                objConnection);
                objConnection.Open();
                HerbList.DataSource=objCommandJ.ExecuteReader();
                HerbList.DataTextField="ChineseHerbs";
                HerbList.DataValueField="SerialNo";
                HerbList.DataBind();
            }
        }
        else if(searchTextBox.Text.Substring(0,2).Equals("12"))
        {
            string strSQLforListboxK= "SELECT SerialNo, ChineseHerbs FROM
            HerbsList WHERE Strokes=12";
            SqlCommand objCommandK =new SqlCommand(strSQLforListboxK
            , objConnection);
            objConnection.Open();
            HerbList.DataSource=objCommandK.ExecuteReader();
            HerbList.DataTextField="ChineseHerbs";
            HerbList.DataValueField="SerialNo";
            HerbList.DataBind();
        }
        else if(searchTextBox.Text.Substring(0,2).Equals("13"))
        {
            string strSQLforListboxL= "SELECT SerialNo, ChineseHerbs FROM
            HerbsList WHERE Strokes=13";
            SqlCommand objCommandL =new SqlCommand(strSQLforListboxL,
            objConnection);

```

```

        objConnection.Open();
        HerbList.DataSource=objCommandL.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("14"))
    {
        string strSQLforListboxM= "SELECT SerialNo, ChineseHerbs FROM
        HerbsList WHERE Strokes=14";
        SqlCommand objCommandM =new SqlCommand(strSQLforListboxM,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandM.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("15"))
    {
        string strSQLforListboxN= "SELECT SerialNo, ChineseHerbs FROM
        HerbsList WHERE Strokes=15";
        SqlCommand objCommandN =new SqlCommand(strSQLforListboxN,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandN.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("16"))
    {
        string strSQLforListboxO= "SELECT SerialNo, ChineseHerbs FROM
        HerbsList WHERE Strokes=16";
        SqlCommand objCommandO =new SqlCommand(strSQLforListboxO,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandO.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("17"))
    {
        string strSQLforListboxP= "SELECT SerialNo, ChineseHerbs FROM
        HerbsList WHERE Strokes=17";
        SqlCommand objCommandP =new SqlCommand(strSQLforListboxP,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandP.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("18"))
    {
        string strSQLforListboxQ= "SELECT SerialNo, ChineseHerbs FROM
        HerbsList WHERE Strokes=9";
        SqlCommand objCommandQ =new SqlCommand(strSQLforListboxQ
        , objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandQ.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(searchTextBox.Text.Substring(0,2).Equals("19"))

```

```

    {
        string strSQLforListboxR= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=19";
SqlCommand objCommandR =new SqlCommand(strSQLforListboxR,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandR.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
else
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes="+str2+"";
SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandU.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
errMessage.Text="There is no data in the herbs' list through stroke "+ str2;
}
}
else if(strLength>=3)
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes="+str2+"";
SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandU.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
errMessage.Text="There is no data in the herbs' list through stroke "+ str2;
}
}
else if(str1.Equals("2"))
{
    if(strLength==1)
    {
        string strSQLforListboxA= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes=2";
SqlCommand objCommandA =new SqlCommand(strSQLforListboxA,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandA.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
else if(strLength==2)
{
    if(searchTxtBox.Text.Substring(0,2).Equals("20"))
    {
        string strSQLforListboxS= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=20";
SqlCommand objCommandS =new SqlCommand(strSQLforListboxS,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandS.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
}
}

```

```

else if(searchTextBox.Text.Substring(0,2).Equals("21"))
{
    string strSQLforListboxT= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=21";
SqlCommand objCommandT =new SqlCommand(strSQLforListboxT,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandT.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
}
else if(searchTextBox.Text.Substring(0,2).Equals("23"))
{
    string strSQLforListboxV= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=23";
SqlCommand objCommandV =new SqlCommand(strSQLforListboxV,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandV.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
}
else if(searchTextBox.Text.Substring(0,2).Equals("25"))
{
    string strSQLforListboxX= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=25";
SqlCommand objCommandX =new SqlCommand(strSQLforListboxX,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandX.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
}
else if(searchTextBox.Text.Substring(0,2).Equals("29"))
{
    string strSQLforListboxZ= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes=29";
SqlCommand objCommandZ =new SqlCommand(strSQLforListboxZ,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandZ.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
}
else
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM
HerbsList WHERE Strokes='"+str2+"'";
SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
objConnection);
objConnection.Open();
HerbList.DataSource=objCommandU.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    errorMessage.Text="There is no data in the herbs' list through stroke "+ str2;
}
}
else if(strLength>=3)
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes='"+str2+"'";
SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,

```



```

        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else if(str1.Equals("3"))
{
    if(strLength==1)
    {
        string strSQLforListboxB= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes=3";
        SqlCommand objCommandB =new SqlCommand(strSQLforListboxB,
objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandB.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes="+str2+"";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else if(str1.Equals("4"))
{
    if(strLength==1)
    {
        string strSQLforListboxC= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes=4";
        SqlCommand objCommandC =new SqlCommand(strSQLforListboxC,
objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandC.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes="+str2+"";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else if(str1.Equals("5"))
{

```

```

if(strLength==1)
{
    string strSQLforListboxD= "SELECT SerialNo, ChineseHerbs FROM HerbsList
    WHERE Strokes=5";
    SqlCommand objCommandD =new SqlCommand(strSQLforListboxD,
    objConnection);
    objConnection.Open();
    HerbList.DataSource=objCommandD.ExecuteReader();
    HerbList.DataTextField="ChineseHerbs";
    HerbList.DataValueField="SerialNo";
    HerbList.DataBind();
}
else
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
    WHERE Strokes="+str2+"";
    SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
    objConnection);
    objConnection.Open();
    HerbList.DataSource=objCommandU.ExecuteReader();
    HerbList.DataTextField="ChineseHerbs";
    HerbList.DataValueField="SerialNo";
    HerbList.DataBind();
    errMessage.Text="There is no data in the herbs' list through stroke "+ str2;
}
}
else if(str1.Equals("6"))
{
    if(strLength==1)
    {
        string strSQLforListboxE= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes=6";
        SqlCommand objCommandE =new SqlCommand(strSQLforListboxE,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandE.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes="+str2+"";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMessage.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
}
else if(str1.Equals("7"))
{
    if(strLength==1)
    {
        string strSQLforListboxF= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes=7";
        SqlCommand objCommandF =new SqlCommand(strSQLforListboxF,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandF.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
}
}

```

```

    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes='"+str2+"'";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else if(str1.Equals("8"))
{
    if(strLength==1)
    {
        string strSQLforListboxG= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes=8";
        SqlCommand objCommandG =new SqlCommand(strSQLforListboxG,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandG.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes='"+str2+"'";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else if(str1.Equals("9"))
{
    if(strLength==1)
    {
        string strSQLforListboxH= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes=9";
        SqlCommand objCommandH =new SqlCommand(strSQLforListboxH,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandH.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else
    {
        string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
        WHERE Strokes='"+str2+"'";
        SqlCommand objCommandU =new SqlCommand(strSQLforListboxU,
        objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommandU.ExecuteReader();
        HerbList.DataTextField="ChineseHerbs";
        HerbList.DataValueField="SerialNo";
    }
}

```

```

        HerbList.DataBind();
        errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
    }
}
else
{
    string strSQLforListboxU= "SELECT SerialNo, ChineseHerbs FROM HerbsList
WHERE Strokes='"+str2+"'";
SqlCommand objCommandU =new SqlCommand(strSQLforListboxU, objConnection);
objConnection.Open();
HerbList.DataSource=objCommandU.ExecuteReader();
HerbList.DataTextField="ChineseHerbs";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
errMsg.Text="There is no data in the herbs' list through stroke "+ str2;
}
}
else if(str1.Equals("A")||str1.Equals("a")||str1.Equals("B")||str1.Equals("b")||
str1.Equals("C")||str1.Equals("c")||str1.Equals("D")||str1.Equals("d")||
str1.Equals("E")||str1.Equals("e")||str1.Equals("F")||str1.Equals("f")||
str1.Equals("G")||str1.Equals("g")||str1.Equals("H")||str1.Equals("h")||
str1.Equals("I")||str1.Equals("i")||str1.Equals("J")||str1.Equals("j")||
str1.Equals("K")||str1.Equals("k")||str1.Equals("L")||str1.Equals("l")||
str1.Equals("M")||str1.Equals("m")||str1.Equals("N")||str1.Equals("n")||
str1.Equals("O")||str1.Equals("o")||str1.Equals("P")||str1.Equals("p")||
str1.Equals("Q")||str1.Equals("q")||str1.Equals("R")||str1.Equals("r")||
str1.Equals("S")||str1.Equals("s")||str1.Equals("T")||str1.Equals("t")||
str1.Equals("U")||str1.Equals("u")||str1.Equals("V")||str1.Equals("v")||
str1.Equals("W")||str1.Equals("w")||str1.Equals("X")||str1.Equals("x")||
str1.Equals("Y")||str1.Equals("y")||str1.Equals("Z")||str1.Equals("z"))
{
    if(str1.Equals("a")||str1.Equals("A"))
    {
        string strSQLforListbox= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
serialNo>=1 and SerialNo<=9";
SqlCommand objCommand =new SqlCommand(strSQLforListbox, objConnection);
objConnection.Open();
HerbList.DataSource=objCommand.ExecuteReader();
HerbList.DataTextField="Pinyin";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
    else if(str1.Equals("b")||str1.Equals("B"))
    {
        string strSQLforListbox= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
serialNo>=10 and SerialNo<=66";
SqlCommand objCommand2 =new SqlCommand(strSQLforListbox, objConnection);
objConnection.Open();
HerbList.DataSource=objCommand2.ExecuteReader();
HerbList.DataTextField="Pinyin";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
    else if(str1.Equals("c")||str1.Equals("C"))
    {
        string strSQLforListbox3= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
serialNo>=67 and SerialNo<=99";
SqlCommand objCommand3 =new SqlCommand(strSQLforListbox3, objConnection);
objConnection.Open();
HerbList.DataSource=objCommand3.ExecuteReader();
HerbList.DataTextField="Pinyin";
HerbList.DataValueField="SerialNo";
HerbList.DataBind();
    }
    else if(str1.Equals("d")||str1.Equals("D"))
    {
        string strSQLforListbox4= "SELECT SerialNo, Pinyin FROM HerbsList WHERE

```

```

        serialNo>=100 and SerialNo<=127";
        SqlCommand objCommand4 =new SqlCommand(strSQLforListbox4, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand4.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("e")||str1.Equals("E"))
    {
        string strSQLforListbox5= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=128 and SerialNo<=129";
        SqlCommand objCommand5 =new SqlCommand(strSQLforListbox5, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand5.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("f")||str1.Equals("F"))
    {
        string strSQLforListbox6= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=130 and SerialNo<=141";
        SqlCommand objCommand6 =new SqlCommand(strSQLforListbox6, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand6.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("g")||str1.Equals("G"))
    {
        string strSQLforListbox7= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=142 and SerialNo<=160";
        SqlCommand objCommand7 =new SqlCommand(strSQLforListbox7, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand7.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("h")||str1.Equals("H"))
    {
        string strSQLforListbox8= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=161 and SerialNo<=195";
        SqlCommand objCommand8 =new SqlCommand(strSQLforListbox8, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand8.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("i")||str1.Equals("I"))
    {
        string strSQLforListbox28= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo=0";
        SqlCommand objCommand28 =new SqlCommand(strSQLforListbox28, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand28.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no first alphabet of i or I in the herbs list.";
    }
    else if(str1.Equals("j")||str1.Equals("J"))
    {
        string strSQLforListbox9= "SELECT SerialNo, Pinyin FROM HerbsList WHERE

```

```

        serialNo>=196 and SerialNo<=216";
        SqlCommand objCommand9 =new SqlCommand(strSQLforListbox9, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand9.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("K")||str1.Equals("k"))
    {
        string strSQLforListbox10= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=217 and SerialNo<=226";
        SqlCommand objCommand10 =new SqlCommand(strSQLforListbox10, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand10.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("")||str1.Equals("L"))
    {
        string strSQLforListbox11= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=227 and SerialNo<=241";
        SqlCommand objCommand11 =new SqlCommand(strSQLforListbox11, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand11.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("m")||str1.Equals("M"))
    {
        string strSQLforListbox12= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=242 and SerialNo<=269";
        SqlCommand objCommand12 =new SqlCommand(strSQLforListbox12, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand12.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("n")||str1.Equals("N"))
    {
        string strSQLforListbox13= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=270 and SerialNo<=274";
        SqlCommand objCommand13 =new SqlCommand(strSQLforListbox13, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand13.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("o")||str1.Equals("O"))
    {
        string strSQLforListbox14= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=275 and SerialNo<=276";
        SqlCommand objCommand14 =new SqlCommand(strSQLforListbox14, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand14.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("p")||str1.Equals("P"))
    {
        string strSQLforListbox15= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=277 and SerialNo<=280";

```

```

        SqlCommand objCommand15 =new SqlCommand(strSQLforListbox15, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand15.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("q")||str1.Equals("Q"))
    {
        string strSQLforListbox17= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=281 and SerialNo<=297";
        SqlCommand objCommand17 =new SqlCommand(strSQLforListbox17, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand17.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("r")||str1.Equals("R"))
    {
        string strSQLforListbox18= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=298 and SerialNo<=314";
        SqlCommand objCommand18 =new SqlCommand(strSQLforListbox18, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand18.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("s")||str1.Equals("S"))
    {
        string strSQLforListbox19= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=315 and SerialNo<=354";
        SqlCommand objCommand19 =new SqlCommand(strSQLforListbox19, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand19.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("t")||str1.Equals("T"))
    {
        string strSQLforListbox20= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=355 and SerialNo<=370";
        SqlCommand objCommand20 =new SqlCommand(strSQLforListbox20, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand20.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("u")||str1.Equals("U"))
    {
        string strSQLforListbox26= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo=0";
        SqlCommand objCommand26 =new SqlCommand(strSQLforListbox26, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand26.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMessage.Text="There is no first alphabet of u or U in the herbs list.";
    }
    else if(str1.Equals("v")||str1.Equals("V"))
    {
        string strSQLforListbox27= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo=0";
    }

```

```

        SqlCommand objCommand27 =new SqlCommand(strSQLforListbox27, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand27.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
        errMsg.Text="There is no first alphabet of v or V in the herbs list.";
    }
    else if(str1.Equals("w")||str1.Equals("W"))
    {
        string strSQLforListbox21= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=371 and SerialNo<=382";
        SqlCommand objCommand21 =new SqlCommand(strSQLforListbox21, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand21.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("x")||str1.Equals("X"))
    {
        string strSQLforListbox22= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=383 and SerialNo<=410";
        SqlCommand objCommand22 =new SqlCommand(strSQLforListbox22, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand22.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("y")||str1.Equals("Y"))
    {
        string strSQLforListbox23= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=411 and SerialNo<=435";
        SqlCommand objCommand23 =new SqlCommand(strSQLforListbox23, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand23.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    else if(str1.Equals("z")||str1.Equals("Z"))
    {
        string strSQLforListbox24= "SELECT SerialNo, Pinyin FROM HerbsList WHERE
        serialNo>=436 and SerialNo<=456";
        SqlCommand objCommand24 =new SqlCommand(strSQLforListbox24, objConnection);
        objConnection.Open();
        HerbList.DataSource=objCommand24.ExecuteReader();
        HerbList.DataTextField="Pinyin";
        HerbList.DataValueField="SerialNo";
        HerbList.DataBind();
    }
    }
    else
    {
        errMsg.Text="The first alphabet of " + str2 + " cannot find in the herbs' list. Please
        enter again!";
    }
    searchTextBox.Text="";
    } // end of else
}
private void addClick(object sender, System.EventArgs e)
{
    int k;
    str3="";
    if(HerbList.SelectedIndex.ToString().Equals("-1"))
    {errMsg.Text=" Please select one item from ListBox before you click add button.";}
}

```





## File:PrintCaseHistory.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class PrintCaseHistory : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.Label Init;
        protected System.Web.UI.WebControls.Label Label10;
        protected System.Web.UI.WebControls.Label Sex;
        protected System.Web.UI.WebControls.Label Label6;
        protected System.Web.UI.WebControls.Label recordDate;
        protected System.Web.UI.WebControls.Label today;
        protected System.Web.UI.WebControls.Label Firstname;
        protected System.Web.UI.WebControls.Label Lastname;
        protected System.Web.UI.WebControls.Label Birthlabel;
        protected System.Web.UI.WebControls.Label BDate;
        protected System.Web.UI.WebControls.Label CaseNO;
        protected System.Web.UI.WebControls.Label FLabel;
        protected System.Web.UI.WebControls.Label diagnosis;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.Label Label4;
        protected System.Web.UI.WebControls.Label Label5;
        protected System.Web.UI.WebControls.Label Label7;
        protected System.Web.UI.WebControls.Label Label8;
        protected System.Web.UI.WebControls.Label LLabel;
        protected System.Web.UI.WebControls.Label Label9;
        protected System.Web.UI.WebControls.DataGrid DataGrid1;
        protected System.Web.UI.WebControls.Label Label11;
        protected System.Web.UI.WebControls.DataGrid DataGrid2;
        protected System.Web.UI.WebControls.Label Label12;
        protected System.Web.UI.WebControls.DataGrid DataGrid3;
        protected System.Web.UI.WebControls.Label Label13;
        protected System.Web.UI.WebControls.DataGrid DataGrid4;
        protected static string dateStr;
        protected static string CaseNum;
        string sexStr, FStr, LStr, miniStr,BirthStr;
        protected static string datStr;
        protected static int CaseID;
        protected string strConn1="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        string conStr, AcpStr;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            dateStr=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
            DateTime.Now.Day,DateTime.Now.Year);
            today.Text=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
            DateTime.Now.Day,DateTime.Now.Year);
            CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
            CaseID=LoginControl.CaseHistory.CaseHistoryNo;
            //Get patient information form Patients' table
        }
    }
}
```

```

SqlConnection objConn_P =new SqlConnection(strConn1);
string PInfo="SELECT * FROM Patients WHERE caseNo='"+CaseNum+"'";
objConn_P.Open();
SqlDataAdapter objAdapter_P=new SqlDataAdapter(PInfo, objConn_P);
DataSet objDataSet_P=new DataSet();
objAdapter_P.Fill(objDataSet_P, "dtPInfo");
objConn_P.Close();
//From the tabel to get patient information
DataTable tbl_P=objDataSet_P.Tables["dtPInfo"];
tbl_P.PrimaryKey=new DataColumn[]{tbl_P.Columns["CaseNo"]};
DataRow row_P=tbl_P.Rows.Find(CaseID);
FStr=row_P["FirstName"].ToString();
LStr=row_P["LastName"].ToString();
sexStr=row_P["Sex"].ToString();
miniStr=row_P["MInit"].ToString();
BirthStr=row_P["Birthday"].ToString();
FLabel.Text=FStr;
Llabel.Text=LStr;
Sex.Text=sexStr;
Init.Text=miniStr;
BDate.Text=BirthStr;
CaseNO.Text=CaseNum;
if(!IsPostBack)
{
    getDiagnosis();
    getAcupuncturePoints();
    getPrescription();
    getInquiry();
    getInspection();
    getListenSmell();
    getPlusePalpation();
}
}
private void getDiagnosis()
{
    SqlConnection objConn_1=new SqlConnection(strConn1);
    string strSQL_1="SELECT Diagnosis FROM CaseHistory WHERE MainType='Diagnosis'and
    CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
    SqlCommand obComm1=new SqlCommand(strSQL_1, objConn_1);
    objConn_1.Open();
    SqlDataReader reader1=obComm1.ExecuteReader();
    if(reader1.Read().ToString().Equals("True"))
    {
        objConn_1.Close();
        reader1.Close();
        objConn_1.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter_1=new SqlDataAdapter(strSQL_1,objConn_1);
        //Dataset & Adapter & Table
        DataSet objDataSet_1=new DataSet("dtDiagnosis");
        objAdapter_1.Fill(objDataSet_1, "dtDiagnosis");
        Label8.Text=Convert.ToString(objDataSet_1.Tables["dtDiagnosis"].Rows[0]["Diagnosis"]);
        objConn_1.Close();
    }
    else
    {
        objConn_1.Close();
        reader1.Close();
        Label8.Text="";
    }
}
}
private void getAcupuncturePoints()
{
    SqlConnection objConn_2=new SqlConnection(strConn1);
    string strSQL_2="SELECT * FROM CaseHistory WHERE MainType='Acupuncture-therapy'and
    CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
    SqlCommand obComm2=new SqlCommand(strSQL_2, objConn_2);

```

```

objConn_2.Open();
SqlDataReader reader2=obComm2.ExecuteReader();
if(reader2.Read().ToString().Equals("True"))
{
    objConn_2.Close();
    reader2.Close();
    objConn_2.Open();
    //DataAdapter setup
    SqlDataAdapter objAdapter_2=new SqlDataAdapter(strSQL_2, objConn_2);
    //Dataset & Adapter & Table
    DataSet objDataSet_2=new DataSet("dtAcupuncture");
    objAdapter_2.Fill(objDataSet_2, "dtAcupuncture");
    int rowCount= objDataSet_2.Tables["dtAcupuncture"].Rows.Count;
    for(int k=0; k<rowCount; k++)
    AcpStr+=Convert.ToString(objDataSet_2.Tables["dtAcupuncture"].Rows[k]["Acupoint"]);
    objConn_2.Close();
    Label5.Text=AcpStr;
}
else
{
    objConn_2.Close();
    reader2.Close();
    Label5.Text="";
}
}
private void getPrescription()
{
    SqlConnection objConn_3=new SqlConnection(strConn1);
    string strSQL_3="SELECT Herbs FROM CaseHistory WHERE MainType='Prescription' and
    CaseNo='"+CaseID+"' and Date='"+dateStr+""";
    SqlCommand obComm3=new SqlCommand(strSQL_3, objConn_3);
    objConn_3.Open();
    SqlDataReader reader3=obComm3.ExecuteReader();
    if(reader3.Read().ToString().Equals("True"))
    {
        objConn_3.Close();
        reader3.Close();
        objConn_3.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter_3=new SqlDataAdapter(strSQL_3,objConn_3);
        //Dataset & Adapter & Table
        DataSet objDataSet_3=new DataSet("dtPrescription");
        objAdapter_3.Fill(objDataSet_3, "dtPrescription");
        //connect all different meridians' acupuncture points together
        int rowCt= objDataSet_3.Tables["dtPrescription"].Rows.Count;
        for(int k=0; k<rowCt; k++)
        {
            conStr+=(" "+(k+1)+".) "+Convert.ToString(objDataSet_3.Tables
            ["dtPrescription"].Rows[k]["Herbs"]);
            objConn_3.Close();
            Label7.Text=conStr;
        }
    }
    else
    {
        objConn_3.Close();
        reader3.Close();
        Label7.Text="";
    }
}
private void getInquiry()
{
    SqlConnection objConn_4=new SqlConnection(strConn1);
    string strSQL_4="SELECT SubType, SuggestionOfDisorders FROM CaseHistory WHERE
    MainType='Inquiry'and CaseNo='"+CaseID+"' and Date='"+dateStr+""";
    SqlCommand obComm4=new SqlCommand(strSQL_4, objConn_4);
    objConn_4.Open();
    SqlDataReader reader4=obComm4.ExecuteReader();

```

```

if(reader4.Read().ToString().Equals("True"))
{
    objConn_4.Close();
    reader4.Close();
    objConn_4.Open();
    //DataAdapter setup
    SqlDataAdapter objAdapter_4=new SqlDataAdapter(strSQL_4,objConn_4);
    //Dataset & Adapter & Table
    DataSet ds_4=new DataSet();
    objAdapter_4.Fill(ds_4,"dtInquiry");
    objConn_4.Close();
    DataGrid1.DataSource=ds_4.Tables["dtInquiry"];
    DataGrid1.DataBind();
}
else
{
    objConn_4.Close();
    reader4.Close();
}
}
private void getInspection()
{
    SqlConnection objConn_5=new SqlConnection(strConn1);
    string strSQL_5="SELECT SubType, SuggestionOfDisorders FROM CaseHistory WHERE
    MainType='Look' and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
    SqlCommand obComm5=new SqlCommand(strSQL_5, objConn_5);
    objConn_5.Open();
    SqlDataReader reader5=obComm5.ExecuteReader();
    if(reader5.Read().ToString().Equals("True"))
    {
        objConn_5.Close();
        reader5.Close();
        objConn_5.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter_5=new SqlDataAdapter(strSQL_5,objConn_5);
        //Dataset & Adapter & Table
        DataSet ds_5=new DataSet();
        objAdapter_5.Fill(ds_5,"dtInspection");
        objConn_5.Close();
        DataGrid2.DataSource=ds_5.Tables["dtInspection"];
        DataGrid2.DataBind();
    }
    else
    {
        objConn_5.Close();
        reader5.Close();
    }
}
private void getListenSmell()
{
    SqlConnection objConn_6=new SqlConnection(strConn1);
    string strSQL_6="SELECT SerialNo, SubType, SuggestionOfDisorders FROM CaseHistory WHERE
    MainType='Auscultation and Olfaction' and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
    SqlCommand obComm6=new SqlCommand(strSQL_6, objConn_6);
    objConn_6.Open();
    SqlDataReader reader6=obComm6.ExecuteReader();
    if(reader6.Read().ToString().Equals("True"))
    {
        objConn_6.Close();
        reader6.Close();
        objConn_6.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter_6=new SqlDataAdapter(strSQL_6,objConn_6);
        //Dataset & Adapter & Table
        DataSet objDataSet_6=new DataSet();
        objAdapter_6.Fill(objDataSet_6, "dtListenSmell");
        DataGrid3.DataSource=objDataSet_6.Tables["dtListenSmell"];
    }
}

```



```

protected System.Web.UI.WebControls.Label Birthlabel;
protected System.Web.UI.WebControls.Label BDate;
protected System.Web.UI.WebControls.Label Label10;
protected System.Web.UI.WebControls.Label Sex;
protected System.Web.UI.WebControls.Label Label6;
protected System.Web.UI.WebControls.Label CaseNO;
protected System.Web.UI.WebControls.Label Label4;
protected System.Web.UI.WebControls.Label Label7;
protected static string dateStr;
protected static string CaseNum;
string sexStr, FStr, LStr, miniStr,BirthStr;
protected static string datStr;
protected static int CaseID;
protected string strConn1="server=(local)\\NetSdk; database=ClinicSystem1; integrated security=true";
string connectStr;
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
    dateStr=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
    DateTime.Now.Day,DateTime.Now.Year);
    today.Text=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
    DateTime.Now.Day,DateTime.Now.Year);
    CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
    CaseID=LoginControl.CaseHistory.CaseHistoryNo;
    //Get patient information form Patients' table
    SqlConnection objConn_P=new SqlConnection(strConn1);
    string PInfo="SELECT * FROM Patients WHERE caseNo='"+CaseNum+""";
    objConn_P.Open();
    SqlDataAdapter objAdapter_P=new SqlDataAdapter(PInfo, objConn_P);
    DataSet objDataSet_P=new DataSet();
    objAdapter_P.Fill(objDataSet_P, "dtPInfo");
    objConn_P.Close();
    //From the tabel to get patient information
    DataTable tbl_P=objDataSet_P.Tables["dtPInfo"];
    tbl_P.PrimaryKey=new DataColumn[]{tbl_P.Columns["CaseNo"]};
    DataRow row_P=tbl_P.Rows.Find(CaseID);
    FStr=row_P["FirstName"].ToString();
    LStr=row_P["LastName"].ToString();
    sexStr=row_P["Sex"].ToString();
    miniStr=row_P["Mnit"].ToString();
    BirthStr=row_P["Birthday"].ToString();
    FLabel.Text=FStr;
    LLabel.Text=LStr;
    Sex.Text=sexStr;
    Init.Text=miniStr;
    BDate.Text=BirthStr;
    CaseNO.Text=CaseNum;
    if(!IsPostBack)
    {
        getPrescription();
    }
}
private void getPrescription()
{
    SqlConnection objConn_3=new SqlConnection(strConn1);
    string strSQL_3="SELECT Herbs FROM CaseHistory WHERE MainType='Prescription' and
    CaseNo='"+CaseID+"' and Date='"+dateStr+""";
    SqlCommand obComm3=new SqlCommand(strSQL_3, objConn_3);
    objConn_3.Open();
    SqlDataReader reader3=obComm3.ExecuteReader();
    if(reader3.Read().ToString().Equals("True"))
    {
        objConn_3.Close();
    }
}

```





## File:RecordView.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class RecordView : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.Label Today;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.Label FN;
        protected System.Web.UI.WebControls.Label FNLabel;
        protected System.Web.UI.WebControls.Label Label5;
        protected System.Web.UI.WebControls.Label M;
        protected System.Web.UI.WebControls.Label LN;
        protected System.Web.UI.WebControls.Label LNLabel;
        protected System.Web.UI.WebControls.Label sex;
        protected System.Web.UI.WebControls.Label x;
        protected System.Web.UI.WebControls.Label BDate;
        protected System.Web.UI.WebControls.Label Date;
        protected System.Web.UI.WebControls.DataGrid DataGrid2;
        protected System.Web.UI.WebControls.DataGrid DataGrid3;
        protected System.Web.UI.WebControls.DataGrid DataGrid4;
        protected System.Web.UI.WebControls.DataGrid DataGrid1;
        string FStr, LStr, miniStr,sexStr, BirthStr, CaseNum;
        protected static int CaseID;
        protected string strConn1="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        protected System.Web.UI.WebControls.DataGrid DataGrid5;
        protected System.Web.UI.WebControls.DataGrid DataGrid6;
        protected static string dateStr;
        protected SqlConnection objConn1;
        protected static string strSQL1;
        protected SqlConnection objConn2;
        protected static string strSQL2;
        protected SqlConnection objConn3;
        protected static string strSQL3;
        protected SqlConnection objConn4;
        protected static string strSQL4;
        protected SqlConnection objConn5;
        protected static string strSQL5;
        protected SqlConnection objConn6;
        protected static string strSQL6;
        protected SqlConnection objConn7;
        protected static string strSQL7;
        protected string ItemsTableName="dtInquiry_CH";
        protected string ItemsTableName2="dtLook_CH";
        protected string ItemsTableName3="dtListen_CH";
        protected string ItemsTableName4="dtpalpation_CH";
        protected string ItemsTableName5="dtacupuncture_CH";
        protected string ItemsTableName6="dtherbs_CH";
        protected string ItemsTableName7="dtDiagnosis_CH";
        protected System.Web.UI.WebControls.DataGrid DataGrid7;
        private void Page_Load(object sender, System.EventArgs e)
        {
            // Do not allow loading the page if the user did not login from default page
            string UserName1="none";
        }
    }
}
```

```

// Do not allow loading the page if the user did not login from default page
UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
if(UserName1.Equals("none"))
    Response.Redirect("DefaultPage.aspx");
dateStr=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
DateTime.Now.Day,DateTime.Now.Year);
Today.Text=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
DateTime.Now.Day,DateTime.Now.Year);
CaseNum=Convert.ToString(LoginControl.CaseHistory.CaseHistoryNo);
Label3.Text=CaseNum;
CaseID=LoginControl.CaseHistory.CaseHistoryNo;
//Get patient information form Patients' table
SqlConnection objConn_R =new SqlConnection(strConn1);
string PInfo="SELECT * FROM Patients WHERE caseNo='"+CaseNum+"'";
objConn_R.Open();
SqlDataAdapter objAdapter_R=new SqlDataAdapter(PInfo, objConn_R);
DataSet objDataSet_R=new DataSet();
objAdapter_R.Fill(objDataSet_R, "dtPatientInfo");
objConn_R.Close();
//From the tabel to get patient information
DataTable tbl_R=objDataSet_R.Tables["dtPatientInfo"];
tbl_R.PrimaryKey=new DataColumn[]{tbl_R.Columns["CaseNo"]};
DataRow row_R=tbl_R.Rows.Find(CaseID);
FStr=row_R["FirstName"].ToString();
LStr=row_R["LastName"].ToString();
sexStr=row_R["Sex"].ToString();
miniStr=row_R["MInit"].ToString();
BirthStr=row_R["Birthday"].ToString();
FNLabel.Text=FStr;
LNLabel.Text=LStr;
x.Text=sexStr;
M.Text=miniStr;
Date.Text=BirthStr;
//Get Inquiry information from CaseHistory table
if(!IsPostBack)
{
    BindGrid1();
    BindGrid2();
    BindGrid3();
    BindGrid4();
    BindGrid5();
    BindGrid6();
    BindGrid7();
}
}
private void BindGrid1()
{
    objConn1=new SqlConnection(strConn1);
    strSQL1="SELECT SerialNo , SubType, SuggestionOfDisorders FROM CaseHistory WHERE
MainType='Inquiry'and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
objConn1.Open();
//DataAdapter setup
SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL1,objConn1);
//Dataset & Adapter & Table
DataSet ds1=new DataSet();
objAdapter1.Fill(ds1,ItemsTableName);
objConn1.Close();
DataGrid1.DataSource=ds1.Tables[ItemsTableName];
DataGrid1.DataBind();
}
private void BindGrid2()
{
    objConn2=new SqlConnection(strConn1);
    strSQL2="SELECT SerialNo, SubType , SuggestionOfDisorders FROM CaseHistory WHERE
MainType='Look' and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
objConn2.Open();
//DataAdapter setup

```

```

        SqlDataAdapter objAdapter2=new SqlDataAdapter(strSQL2,objConn2);
        //Dataset & Adapter & Table
        DataSet ds2=new DataSet();
        objAdapter2.Fill(ds2, ItemsTableName2);
        objConn2.Close();
        DataGrid2.DataSource=ds2.Tables[ItemsTableName2];
        DataGrid2.DataBind();
    }
    private void BindGrid3()
    {
        objConn3=new SqlConnection(strConn1);
        strSQL3="SELECT SerialNo, SubType, SuggestionOfDisorders FROM CaseHistory WHERE
        MainType='Auscultation and Olfaction' and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
        objConn3.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter3=new SqlDataAdapter(strSQL3,objConn3);
        //Dataset & Adapter & Table
        DataSet objDataSet3=new DataSet();
        objAdapter3.Fill(objDataSet3, "ItemsTableName3");
        DataGrid3.DataSource=objDataSet3.Tables["ItemsTableName3"];
        DataGrid3.DataBind();
        objConn3.Close();
    }
    private void BindGrid4()
    {
        objConn4=new SqlConnection(strConn1);
        strSQL4="SELECT SerialNo, SubType , SuggestionOfDisorders FROM CaseHistory WHERE
        (MainType='Palpation' or MainType='Pulse-Taking')and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
        objConn4.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter4=new SqlDataAdapter(strSQL4,objConn4);
        //Dataset & Adapter & Table
        DataSet objDataSet4=new DataSet();
        objAdapter4.Fill(objDataSet4, "dtpalpation_CH");
        DataGrid4.DataSource=objDataSet4.Tables["dtpalpation_CH"];
        DataGrid4.DataBind();
        objConn4.Close();
    }
    private void BindGrid5()
    {
        objConn5=new SqlConnection(strConn1);
        strSQL5="SELECT SerialNo, SubType, Acupoint FROM CaseHistory WHERE
        MainType='Acupuncture-therapy' and CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
        objConn5.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter5=new SqlDataAdapter(strSQL5,objConn5);
        //Dataset & Adapter & Table
        DataSet objDataSet5=new DataSet();
        objAdapter5.Fill(objDataSet5, ItemsTableName5 );
        DataGrid5.DataSource=objDataSet5.Tables[ItemsTableName5];
        DataGrid5.DataBind();
        objConn5.Close();
    }
    private void BindGrid6()
    {
        objConn6=new SqlConnection(strConn1);
        strSQL6="SELECT Serialno, Herbs FROM CaseHistory WHERE MainType='Prescription'and
        CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
        objConn6.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter6=new SqlDataAdapter(strSQL6,objConn6);
        //Dataset & Adapter & Table
        DataSet objDataSet6=new DataSet();
        objAdapter6.Fill(objDataSet6, "ItemsTableName6");
        DataGrid6.DataSource=objDataSet6.Tables["ItemsTableName6"];
        DataGrid6.DataBind();
        objConn6.Close();
    }

```

```

}
private void BindGrid7()
{
    objConn7=new SqlConnection(strConn1);
    strSQL7="SELECT Serialno, Diagnosis FROM CaseHistory WHERE MainType='Diagnosis'and
    CaseNo='"+CaseID+"' and Date='"+dateStr+"'";
    objConn7.Open();
    //DataAdapter setup
    SqlDataAdapter objAdapter7=new SqlDataAdapter(strSQL7,objConn7);
    //Dataset & Adapter & Table
    DataSet objDataSet7=new DataSet();
    objAdapter7.Fill(objDataSet7, "ItemsTableName7");
    DataGrid7.DataSource=objDataSet7.Tables["ItemsTableName7"];
    DataGrid7.DataBind();
    objConn7.Close();
}
private void ChangPages(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid1.CurrentPageIndex=e.NewPageIndex;
    BindGrid1();
}
private void changePage2(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid2.CurrentPageIndex=e.NewPageIndex;
    BindGrid2();
}
private void changePage3(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid3.CurrentPageIndex=e.NewPageIndex;
    BindGrid3();
}
private void changePage4(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid4.CurrentPageIndex=e.NewPageIndex;
    BindGrid4();
}
private void changePage5(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid5.CurrentPageIndex=e.NewPageIndex;
    BindGrid5();
}
private void changePage6(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGrid6.CurrentPageIndex=e.NewPageIndex;
    BindGrid6();
}
private void EditRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataGrid1.EditItemIndex=e.Item.ItemIndex;
    BindGrid1();
}
private void updateRow(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    //Retrieve the field values in the edited row
    int SerialID=Convert.ToInt32(e.Item.Cells[0].Text);
    TextBox subtypeTxtBox=(TextBox)e.Item.Cells[1].Controls[0];
    string subtype=Convert.ToString(subtypeTxtBox.Text);
    TextBox disorderTxtBox=(TextBox)e.Item.Cells[2].Controls[0];
    string disorder=Convert.ToString(disorderTxtBox.Text);
    DataGrid1.EditItemIndex=-1;
    UpdateItems(SerialID, subtype, disorder);
    objConn1=new SqlConnection(strConn1);
    objConn1.Open();
    //DataAdapter setup
    SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL1,objConn1);
    DataSet ds1=new DataSet();
    objAdapter1.Fill(ds1,ItemsTableName);
}

```

```

        objConn1.Close();
        DataGrid1.DataSource=ds1.Tables[ItemsTableName];
        DataGrid1.DataBind();
    }
    private void UpdateItems(int SerialID,string sub, string disorders)
    {
        objConn1=new SqlConnection(strConn1);
        objConn1.Open();
        SqlDataAdapter adapter1=new SqlDataAdapter(strSQL1, objConn1);
        DataSet ds1=new DataSet();
        adapter1.Fill(ds1, ItemsTableName);
        objConn1.Close();
        //Modify the in-memory records in the DataSet
        DataTable table1=ds1.Tables[ItemsTableName];
        table1.PrimaryKey=new DataColumn[] {table1.Columns["SerialNo"]};
        DataRow r1=table1.Rows.Find(SerialID);
        r1["SubType"]=sub;
        r1["SuggestionOfDisorders"]=disorders;
        //Reconnect the DataSet and update the database
        SqlCommandBuilder cb1=new SqlCommandBuilder(adapter1);
        objConn1.Open();
        adapter1.Update(ds1, ItemsTableName);
        objConn1.Close();
    }
    private void DeleteRecord1(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //selcet the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid1.Items.Count; //Cout the number of items in each page of DataGrid1
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid1.CurrentPageIndex.Equals(0))
            DataGrid1.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid1.CurrentPageIndex-=1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid1.EditItemIndex = -1;
        deleteSelectedRow1(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds_1 = new DataSet();
        DataGrid1.DataSource = ds_1.Tables["dtSelectedItems1"];
        DataGrid1.DataBind();
        BindGrid1();
    }
    private void deleteSelectedRow1(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        SqlConnection objConn_1= new SqlConnection(strConn1);
        objConn_1.Open();
        SqlDataAdapter apt_1 = new SqlDataAdapter(strSQL1, objConn_1);
        DataSet ds_1 = new DataSet();
        apt_1.Fill(ds_1, "dtSelectedItems1");
        objConn_1.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl_1 = ds_1.Tables["dtSelectedItems1"];
        tbl_1.PrimaryKey = new DataColumn[] {tbl_1.Columns["SerialNo"]};
        DataRow row_1 = tbl_1.Rows.Find(SerialID);
        row_1.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb_1 = new SqlCommandBuilder(apt_1);
        objConn_1 = new SqlConnection(strConn1);
        objConn_1.Open();
        apt_1.Update(ds_1,"dtSelectedItems1");
        objConn_1.Close();
    }
    private void cancelRow(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {

```

```

        DataGrid1.EditItemIndex = -1;
        BindGrid1();
    }
    private void EditRecord2(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid2.EditItemIndex=e.Item.ItemIndex;
        BindGrid2();
    }
    private void updateRow2(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        //Retrieve the field values in the edited row
        int SerialID2=Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox subtypeTextBox2=(TextBox)e.Item.Cells[1].Controls[0];
        string subtype2=Convert.ToString(subtypeTextBox2.Text);
        TextBox disorderTextBox2=(TextBox)e.Item.Cells[2].Controls[0];
        string disorder2=Convert.ToString(disorderTextBox2.Text);
        DataGrid2.EditItemIndex=-1;
        UpdateItems2(SerialID2, subtype2, disorder2);
        objConn2=new SqlConnection(strConn1);
        objConn2.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter2=new SqlDataAdapter(strSQL2,objConn2);
        DataSet ds2=new DataSet();
        objAdapter2.Fill(ds2,ItemsTableName2);
        objConn2.Close();
        DataGrid2.DataSource=ds2.Tables[ItemsTableName2];
        DataGrid2.DataBind();
    }
    private void UpdateItems2(int SerialID,string sub, string disorders)
    {
        objConn2=new SqlConnection(strConn1);
        objConn2.Open();
        SqlDataAdapter adapter2=new SqlDataAdapter(strSQL2, objConn2);
        DataSet ds2=new DataSet();
        adapter2.Fill(ds2, ItemsTableName2);
        objConn2.Close();
        //Modify the in-memory records in the DataSet
        DataTable table2=ds2.Tables[ItemsTableName2];
        table2.PrimaryKey=new DataColumn[] {table2.Columns["SerialNo"]};
        DataRow r2=table2.Rows.Find(SerialID);
        r2["Sub Type"]=sub;
        r2["SuggestionOfDisorders"]=disorders;
        //Reconnect the DataSet and update the database
        SqlCommandBuilder cb2=new SqlCommandBuilder(adapter2);
        objConn2.Open();
        adapter2.Update(ds2, ItemsTableName2);
        objConn2.Close();
    }
    private void DeleteRecord2(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //selcet the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid2.Items.Count; //Cout the number of items in each page of DataGrid2
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid2.CurrentPageIndex.Equals(0))
            DataGrid2.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid2.CurrentPageIndex=-1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid2.EditItemIndex = -1;
        deleteSelectedRow2(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds_2 = new DataSet();
        DataGrid2.DataSource = ds_2.Tables["dtSelectedItems2"];
        DataGrid2.DataBind();
        BindGrid2();
    }

```

```

}
private void deleteSelectedRow2(int SerialID)
{
    // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
    SqlConnection objConn_2= new SqlConnection(strConn1);
    objConn_2.Open();
    SqlDataAdapter apt_2 = new SqlDataAdapter(strSQL2, objConn_2);
    DataSet ds_2 = new DataSet();
    apt_2.Fill(ds_2, "dtSelectedItems2");
    objConn_2.Close();
    // Mark the appointment as Deleted in the DataSet
    DataTable tbl_2 = ds_2.Tables["dtSelectedItems2"];
    tbl_2.PrimaryKey = new DataColumn[] {tbl_2.Columns["SerialNo"]};
    DataRow row_2 = tbl_2.Rows.Find(SerialID);
    row_2.Delete();
    // Reconnect the DataSet and delete the record from the database
    SqlCommandBuilder cb_2 = new SqlCommandBuilder(apt_2);
    objConn_2 = new SqlConnection(strConn1);
    objConn_2.Open();
    apt_2.Update(ds_2,"dtSelectedItems2");
    objConn_2.Close();
}
private void cancelRow2(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataGrid2.EditItemIndex = -1;
    BindGrid2();
}
private void EditRecord3(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    DataGrid3.EditItemIndex=e.Item.ItemIndex;
    BindGrid3();
}
private void updateRow3(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    //Retrieve the field values in the edited row
    int SerialID3=Convert.ToInt32(e.Item.Cells[0].Text);
    TextBox subtypeTxtBox3=(TextBox)e.Item.Cells[1].Controls[0];
    string subtype3=Convert.ToString(subtypeTxtBox3.Text);
    TextBox disorderTxtBox3=(TextBox)e.Item.Cells[2].Controls[0];
    string disorder3=Convert.ToString(disorderTxtBox3.Text);
    DataGrid3.EditItemIndex=-1;
    UpdateItems3(SerialID3, subtype3, disorder3);
    objConn3=new SqlConnection(strConn1);
    objConn3.Open();
    //DataAdapter setup
    SqlDataAdapter objAdapter3=new SqlDataAdapter(strSQL3,objConn3);
    DataSet ds3=new DataSet();
    objAdapter3.Fill(ds3,ItemsTableName3);
    objConn3.Close();
    DataGrid3.DataSource=ds3.Tables[ItemsTableName3];
    DataGrid3.DataBind();
}
private void UpdateItems3(int SerialID,string sub, string disorders)
{
    objConn3=new SqlConnection(strConn1);
    objConn3.Open();
    SqlDataAdapter adapter3=new SqlDataAdapter(strSQL3, objConn3);
    DataSet ds3=new DataSet();
    adapter3.Fill(ds3, ItemsTableName3);
    objConn3.Close();
    //Modify the in-memory records in the DataSet
    DataTable table3=ds3.Tables[ItemsTableName3];
    table3.PrimaryKey=new DataColumn[] {table3.Columns["SerialNo"]};
    DataRow r3=table3.Rows.Find(SerialID);
    r3["SubType"]=sub;
    r3["SuggestionOfDisorders"]=disorders;
    //Reconnect the DataSet and update the database
}

```

```

        SqlCommandBuilder cb3=new SqlCommandBuilder(adapter3);
        objConn3.Open();
        adapter3.Update(ds3, ItemsTableName3);
        objConn3.Close();
    }
    private void DeleteRecord3(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //select the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid3.Items.Count; //Cout the number of items in each page of DataGrid3
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid3.CurrentPageIndex.Equals(0))
            DataGrid3.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid3.CurrentPageIndex-=1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid3.EditItemIndex = -1;
        deleteSelectedRow3(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds_3 = new DataSet();
        DataGrid3.DataSource = ds_3.Tables["dtSelectedItems3"];
        DataGrid3.DataBind();
        BindGrid3();
    }
    private void deleteSelectedRow3(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        SqlConnection objConn_3= new SqlConnection(strConn1);
        objConn_3.Open();
        SqlDataAdapter apt_3 = new SqlDataAdapter(strSQL3, objConn_3);
        DataSet ds_3 = new DataSet();
        apt_3.Fill(ds_3, "dtSelectedItems3");
        objConn_3.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl_3 = ds_3.Tables["dtSelectedItems3"];
        tbl_3.PrimaryKey = new DataColumn[] {tbl_3.Columns["SerialNo"]};
        DataRow row_3 = tbl_3.Rows.Find(SerialID);
        row_3.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb_3 = new SqlCommandBuilder(apt_3);
        objConn_3 = new SqlConnection(strConn1);
        objConn_3.Open();
        apt_3.Update(ds_3,"dtSelectedItems3");
        objConn_3.Close();
    }
    private void cancelRow3(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid3.EditItemIndex = -1;
        BindGrid3();
    }
    private void EditRecord4(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid4.EditItemIndex=e.Item.ItemIndex;
        BindGrid4();
    }
    private void updateRow4(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        //Retrieve the field values in the edited row
        int SerialID4=Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox subtypeTextBox4=(TextBox)e.Item.Cells[1].Controls[0];
        string subtype4=Convert.ToString(subtypeTextBox4.Text);
        TextBox disorderTextBox4=(TextBox)e.Item.Cells[2].Controls[0];
        string disorder4=Convert.ToString(disorderTextBox4.Text);
        DataGrid4.EditItemIndex=-1;
        UpdateItems4(SerialID4, subtype4, disorder4);
        objConn4=new SqlConnection(strConn1);
    }

```



```

        objConn4.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter4=new SqlDataAdapter(strSQL4,objConn4);
        DataSet ds4=new DataSet();
        objAdapter4.Fill(ds4,ItemsTableName4);
        objConn4.Close();
        DataGrid4.DataSource=ds4.Tables[ItemsTableName4];
        DataGrid4.DataBind();
    }
    private void UpdateItems4(int SerialID,string sub, string disorders)
    {
        objConn4=new SqlConnection(strConn1);
        objConn4.Open();
        SqlDataAdapter adapter4=new SqlDataAdapter(strSQL4, objConn4);
        DataSet ds4=new DataSet();
        adapter4.Fill(ds4, ItemsTableName4);
        objConn4.Close();
        //Modify the in-memory records in the DataSet
        DataTable table4=ds4.Tables[ItemsTableName4];
        table4.PrimaryKey=new DataColumn[] {table4.Columns["SerialNo"]};
        DataRow r4=table4.Rows.Find(SerialID);
        r4["SubType"]=sub;
        r4["SuggestionOfDisorders"]=disorders;
        //Reconnect the DataSet and update the database
        SqlCommandBuilder cb4=new SqlCommandBuilder(adapter4);
        objConn4.Open();
        adapter4.Update(ds4, ItemsTableName4);
        objConn4.Close();
    }
    private void DeleteRecord4(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //select the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid4.Items.Count; //Cout the number of items in each page of DataGrid4
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid4.CurrentPageIndex.Equals(0))
            DataGrid4.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid4.CurrentPageIndex-=1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid4.EditItemIndex = -1;
        deleteSelectedRow4(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds_4 = new DataSet();
        DataGrid4.DataSource = ds_4.Tables["dtSelectedItems4"];
        DataGrid4.DataBind();
        BindGrid4();
    }
    private void deleteSelectedRow4(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        objConn4= new SqlConnection(strConn1);
        objConn4.Open();
        SqlDataAdapter apt_4 = new SqlDataAdapter(strSQL4, objConn4);
        DataSet ds_4 = new DataSet();
        apt_4.Fill(ds_4, "dtSelectedItems4");
        objConn4.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl_4 = ds_4.Tables["dtSelectedItems4"];
        tbl_4.PrimaryKey = new DataColumn[] {tbl_4.Columns["SerialNo"]};
        DataRow row_4 = tbl_4.Rows.Find(SerialID);
        row_4.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb_4 = new SqlCommandBuilder(apt_4);
        objConn4 = new SqlConnection(strConn1);
        objConn4.Open();
    }

```

```

        apt_4.Update(ds_4,"dtSelectedItems4");
        objConn4.Close();
    }
    private void cancelRow4(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid4.EditItemIndex = -1;
        BindGrid4();
    }
    private void EditRecord5(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid5.EditItemIndex=e.Item.ItemIndex;
        BindGrid5();
    }
    private void updateRow5(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        //Retrieve the field values in the edited row
        int SerialID5=Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox subtypeTxtBox5=(TextBox)e.Item.Cells[1].Controls[0];
        string subtype5=Convert.ToString(subtypeTxtBox5.Text);
        TextBox disorderTxtBox5=(TextBox)e.Item.Cells[2].Controls[0];
        string disorder5=Convert.ToString(disorderTxtBox5.Text);
        DataGrid5.EditItemIndex=-1;
        UpdateItems5(SerialID5, subtype5, disorder5);
        objConn5=new SqlConnection(strConn1);
        objConn5.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter5=new SqlDataAdapter(strSQL5,objConn5);
        DataSet ds5=new DataSet();
        objAdapter5.Fill(ds5,ItemsTableName5);
        objConn5.Close();
        DataGrid5.DataSource=ds5.Tables[ItemsTableName5];
        DataGrid5.DataBind();
    }
    private void UpdateItems5(int SerialID,string sub, string accupoints)
    {
        objConn5=new SqlConnection(strConn1);
        objConn5.Open();
        SqlDataAdapter adapter5=new SqlDataAdapter(strSQL5, objConn5);
        DataSet ds5=new DataSet();
        adapter5.Fill(ds5, ItemsTableName5);
        objConn5.Close();
        //Modify the in-memory records in the DataSet
        DataTable table5=ds5.Tables[ItemsTableName5];
        table5.PrimaryKey=new DataColumn[] {table5.Columns["SerialNo"]};
        DataRow r5=table5.Rows.Find(SerialID);
        r5["SubType"]=sub;
        r5["Accupoint"]=accupoints;
        //Reconnect the DataSet and update the database
        SqlCommandBuilder cb5=new SqlCommandBuilder(adapter5);
        objConn5.Open();
        adapter5.Update(ds5, ItemsTableName5);
        objConn5.Close();
    }
    private void DeleteRecord5(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //selcet the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid5.Items.Count; //Cout the number of items in each page of DataGrid5
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid5.CurrentPageIndex.Equals(0))
            DataGrid5.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid5.CurrentPageIndex-=1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid5.EditItemIndex = -1;
        deleteSelectedRow5(SerialID);
    }

```

```

        // Display the remaining items in the DataGrid
        DataSet ds_5 = new DataSet();
        DataGrid5.DataSource = ds_5.Tables["dtSelectedItems5"];
        DataGrid5.DataBind();
        BindGrid5();
    }
    private void deleteSelectedRow5(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        objConn5= new SqlConnection(strConn1);
        objConn5.Open();
        SqlDataAdapter apt_5 = new SqlDataAdapter(strSQL5, objConn5);
        DataSet ds_5 = new DataSet();
        apt_5.Fill(ds_5, "dtSelectedItems5");
        objConn5.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl_5 = ds_5.Tables["dtSelectedItems5"];
        tbl_5.PrimaryKey = new DataColumn[] {tbl_5.Columns["SerialNo"]};
        DataRow row_5 = tbl_5.Rows.Find(SerialID);
        row_5.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb_5 = new SqlCommandBuilder(apt_5);
        objConn5 = new SqlConnection(strConn1);
        objConn5.Open();
        apt_5.Update(ds_5,"dtSelectedItems5");
        objConn5.Close();
    }
    private void cancelRow5(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid5.EditItemIndex = -1;
        BindGrid5();
    }
    private void EditRecord6(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid6.EditItemIndex=e.Item.ItemIndex;
        BindGrid6();
    }
    private void updateRow6(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        //Retrieve the field values in the edited row
        int SerialID6=Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox herbTextBox6=(TextBox)e.Item.Cells[1].Controls[0];
        string herb6=Convert.ToString(herbTextBox6.Text);
        DataGrid6.EditItemIndex=-1;
        UpdateItems6(SerialID6,herb6);
        objConn6=new SqlConnection(strConn1);
        objConn6.Open();
        //DataAdapter setup
        SqlDataAdapter objAdapter6=new SqlDataAdapter(strSQL6,objConn6);
        DataSet ds6=new DataSet();
        objAdapter6.Fill(ds6,ItemsTableName6);
        objConn6.Close();
        DataGrid6.DataSource=ds6.Tables[ItemsTableName6];
        DataGrid6.DataBind();
    }
    private void UpdateItems6(int SerialID, string herbs)
    {
        objConn6=new SqlConnection(strConn1);
        objConn6.Open();
        SqlDataAdapter adapter6=new SqlDataAdapter(strSQL6, objConn6);
        DataSet ds6=new DataSet();
        adapter6.Fill(ds6, ItemsTableName6);
        objConn6.Close();
        //Modify the in-memory records in the DataSet
        DataTable table6=ds6.Tables[ItemsTableName6];
        table6.PrimaryKey=new DataColumn[] {table6.Columns["SerialNo"]};
        DataRow r6=table6.Rows.Find(SerialID);
    }

```

```

        r6["Herbs"]=herbs;
        //Reconnect the DataSet and update the database
        SqlCommandBuilder cb6=new SqlCommandBuilder(adapter6);
        objConn6.Open();
        adapter6.Update(ds6, ItemsTableName6);
        objConn6.Close();
    }
    private void DeleteRecord6(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialNo to be deleted
        //select the right column position, which contain primary key and put its content into the variable
        int itemCt=DataGrid6.Items.Count; //Cout the number of items in each page of DataGrid6
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //If the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && DataGrid6.CurrentPageIndex.Equals(0))
            DataGrid6.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            DataGrid6.CurrentPageIndex=-1;
        int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
        DataGrid6.EditItemIndex = -1;
        deleteSelectedRow6(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds_6 = new DataSet();
        DataGrid6.DataSource = ds_6.Tables["dtSelectedItems6"];
        DataGrid6.DataBind();
        BindGrid6();
    }
    private void deleteSelectedRow6(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        objConn6= new SqlConnection(strConn1);
        objConn6.Open();
        SqlDataAdapter apt_6 = new SqlDataAdapter(strSQL6, objConn6);
        DataSet ds_6 = new DataSet();
        apt_6.Fill(ds_6, "dtSelectedItems6");
        objConn6.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl_6 = ds_6.Tables["dtSelectedItems6"];
        tbl_6.PrimaryKey = new DataColumn[] {tbl_6.Columns["SerialNo"]};
        DataRow row_6 = tbl_6.Rows.Find(SerialID);
        row_6.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb_6 = new SqlCommandBuilder(apt_6);
        objConn6 = new SqlConnection(strConn1);
        objConn6.Open();
        apt_6.Update(ds_6,"dtSelectedItems6");
        objConn6.Close();
    }
    private void cancelRow6(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid6.EditItemIndex = -1;
        BindGrid6();
    }
    private void EditRecord7(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid7.EditItemIndex=e.Item.ItemIndex;
        BindGrid7();
    }
    private void updateRow7(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        //Retrieve the field values in the edited row
        int SerialID7=Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox diagnosisTxtBox7=(TextBox)e.Item.Cells[1].Controls[0];
        string diagnosis7=Convert.ToString(diagnosisTxtBox7.Text);
        DataGrid7.EditItemIndex=-1;
        UpdateItems7(SerialID7,diagnosis7);
        objConn7=new SqlConnection(strConn1);
    }

```

```

objConn7.Open();
//DataAdapter setup
SqlDataAdapter objAdapter7=new SqlDataAdapter(strSQL7,objConn7);
DataSet ds7=new DataSet();
objAdapter7.Fill(ds7,ItemsTableName7);
objConn7.Close();
DataGrid7.DataSource=ds7.Tables[ItemsTableName7];
DataGrid7.DataBind();
}
private void UpdateItems7(int SerialID, string dia)
{
objConn7=new SqlConnection(strConn1);
objConn7.Open();
SqlDataAdapter adapter7=new SqlDataAdapter(strSQL7, objConn7);
DataSet ds7=new DataSet();
adapter7.Fill(ds7, ItemsTableName7);
objConn7.Close();
//Modify the in-memory records in the DataSet
DataTable table7=ds7.Tables[ItemsTableName7];
table7.PrimaryKey=new DataColumn[] {table7.Columns["SerialNo"]};
DataRow r7=table7.Rows.Find(SerialID);
r7["Diagnosis"]=dia;
//Reconnect the DataSet and update the database
SqlCommandBuilder cb7=new SqlCommandBuilder(adapter7);
objConn7.Open();
adapter7.Update(ds7, ItemsTableName7);
objConn7.Close();
}
private void DeleteRecord7(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
// Retrieve the SerialNo to be deleted
//select the right column position, which contain primary key and put its content into the variable
int itemCt=DataGrid7.Items.Count; //Cout the number of items in each page of DataGrid7
//If the page only contain one items, after delte the item, the page will jump to the previous page
//If the current is zero, keep the page in the zero after delete the last item in page zero
if(itemCt.Equals(1) && DataGrid7.CurrentPageIndex.Equals(0))
DataGrid7.CurrentPageIndex=0;
else if(itemCt.Equals(1))
DataGrid7.CurrentPageIndex=-1;
int SerialID = Convert.ToInt32(e.Item.Cells[0].Text);
DataGrid7.EditItemIndex = -1;
deleteSelectedRow7(SerialID);
// Display the remaining items in the DataGrid
DataSet ds_7 = new DataSet();
DataGrid7.DataSource = ds_7.Tables["dtSelectedItems7"];
DataGrid7.DataBind();
BindGrid7();
}
private void deleteSelectedRow7(int SerialID)
{
// Create and load a DataSet with records from ClinicSystem1's CaseHistory table
objConn7= new SqlConnection(strConn1);
objConn7.Open();
SqlDataAdapter apt_7 = new SqlDataAdapter(strSQL7, objConn7);
DataSet ds_7 = new DataSet();
apt_7.Fill(ds_7, "dtSelectedItems7");
objConn7.Close();
// Mark the appointment as Deleted in the DataSet
DataTable tbl_7 = ds_7.Tables["dtSelectedItems7"];
tbl_7.PrimaryKey = new DataColumn[] {tbl_7.Columns["SerialNo"]};
DataRow row_7 = tbl_7.Rows.Find(SerialID);
row_7.Delete();
// Reconnect the DataSet and delete the record from the database
SqlCommandBuilder cb_7 = new SqlCommandBuilder(apt_7);
objConn7 = new SqlConnection(strConn1);
objConn7.Open();
apt_7.Update(ds_7,"dtSelectedItems7");
}

```

```

        objConn7.Close();
    }
    private void cancelRow7(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        DataGrid7.EditItemIndex = -1;
        BindGrid7();
    }
}
}

```

### File: ScheduleAnAppointment.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class ScheduleAnAppointment : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Button exitButton;
        protected System.Web.UI.WebControls.Button viewScheduleButton;
        protected System.Web.UI.WebControls.Button SaveButton;
        protected System.Web.UI.WebControls.Button cancelButton;
        protected System.Web.UI.WebControls.Label phoneLabel1;
        protected System.Web.UI.WebControls.Label dateLabel;
        protected System.Web.UI.WebControls.Label firstNameLabel1;
        protected System.Web.UI.WebControls.Label lastNameLabel1;
        protected System.Web.UI.WebControls.Label datelabel1;
        protected System.Web.UI.WebControls.Label timeLabel1;
        protected System.Web.UI.WebControls.Label todayLabel1;
        protected System.Web.UI.WebControls.Label ScheduleLabel;
        protected System.Web.UI.WebControls.TextBox phoneTextBox;
        protected System.Web.UI.WebControls.TextBox selectionDateTextBox;
        protected System.Web.UI.WebControls.TextBox firstNameTextBox;
        protected System.Web.UI.WebControls.TextBox lastNameTextBox;
        protected System.Web.UI.WebControls.DropDownList timeFilter;
        protected System.Web.UI.WebControls.Calendar calAppoint;
        private string firstStr, lastStr, phoneStr, dateStr, timeStr;
        protected System.Web.UI.WebControls.Label eMessage;
        string strConn="server=(local)\NetSDK; database=ClinicSystem1; integrated security=true";
        private SqlConnection objConn;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.staffName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            dateLabel.Text=String.Format("{0:D2}/{1:D2}/{2:D2}",
                DateTime.Now.Month, DateTime.Now.Day,DateTime.Now.Year);
        }
        private void CalendarChanged(object sender, System.EventArgs e)
        {
            selectionDateTextBox.Text=calAppoint.SelectedDate.ToShortDateString();
        }
        // end of CalendarChanged
        private void candelClick(object sender, System.EventArgs e)
        {
            firstNameTextBox.Text="";
        }
    }
}

```

```

        lastNameTextBox.Text="";
        phoneTextBox.Text="";
        timeFilter.SelectedItem.Text="9:00AM";
        selectionDateTextBox.Text="";
        eMessage.Text="";
    } // end of CancelClick
    private void exitClick(object sender, System.EventArgs e)
    {
        Response.Redirect("StaffPage.aspx");
    } // end of exitClick
    private void ViewScheduleClick(object sender, System.EventArgs e)
    {
        Response.Redirect("ViewEditAppointments.aspx");
    }
    private void Connect()
    {
        if(objConn==null)
            objConn= new SqlConnection(strConn);
        if(objConn.State==ConnectionState.Closed)
            objConn.Open();
    }
    private void Disconnection()
    {
        objConn.Close();
    }
    private void saveClick(object sender, System.EventArgs e)
    {
        // If there are one person take an appointment at the same time already,
        // it will show the error message in the form and will not save the input.
        dateStr=selectionDateTextBox.Text;
        firstStr=firstNameTextBox.Text;
        lastStr= lastNameTextBox.Text;
        phoneStr=phoneTextBox.Text;
        timeStr=timeFilter.SelectedItem.Text;
        if((firstNameTextBox.Text.Equals(""))||
            (lastNameTextBox.Text.Equals(""))||(phoneTextBox.Text.Equals(""))||
            (selectionDateTextBox.Text.Equals("")))
        {
            eMessage.Text="      Please enter data in any empty column! ";
        }
        else
        {
            if(!phoneStr.Length.Equals(13))
            {eMessage.Text=" Please input the phone number as the following format. Ex. (888)888-8888";}
            else
            {
                //Check the phone is the right format or not. Ex.(888)888-8888
                if(!phoneStr.Substring(0,1).Equals("(")&&!phoneStr.Substring(0,4).Equals(")")&&
                    (!phoneStr.Substring(0,8).Equals("-")))
                {
                    eMessage.Text=" Please input the phone number as the following format.
                    Ex. (888)888-8888";
                }
                else
                {
                    //No matter uper case or lower case which you had input,the first caaracter will be
                    uppercase and the other part will be lowercase
                    string f1=firstStr.Substring(0,1).ToUpper();
                    string l1=lastStr.Substring(0,1).ToUpper();
                    string f2=firstStr.Substring(1, firstStr.Length-1).ToLower();
                    string l2=lastStr.Substring(1, lastStr.Length-1).ToLower();
                    firstStr=f1 +f2;
                    lastStr=l1+l2;
                    //Check weather the patient information has existed in the database or not
                    objConn =new SqlConnection(strConn);
                    string CheckStr="SELECT * FROM Patients WHERE FirstName='"+firstStr+"' and
                    LastName='"+lastStr+"' and Phone='"+phoneStr+"'";
                }
            }
        }
    }

```

```

SqlCommand obComm_P=new SqlCommand(CheckStr, objConn);
Connect();
SqlDataReader rd_P=obComm_P.ExecuteReader();
if(rd_P.Read().ToString().Equals("False"))
{
    Disconnection();
    rd_P.Close();
    eMessage.Text="The patient information cannot be found in the database.
Please check name and phone again or create a new patient first!";
}
else
{
    string caseNumStr;
    rd_P.Close();
    Disconnection();
    //Connection set up
    objConn =new SqlConnection(strConn);
    string strQuery="SELECT AppDate, AppTime FROM NewAppointments WHERE
AppDate='"+dateStr+"' and AppTime='"+timeStr+"'";
    SqlCommand obComm=new SqlCommand(strQuery, objConn);
    Connect();
    SqlDataReader reader=obComm.ExecuteReader();
    if(reader.Read().ToString().Equals("True"))
    {
        Disconnection();
        reader.Close();
        eMessage.Text=timeFilter.SelectedItem.ToString()+"
"+calAppoint.SelectedDate.ToLongDateString()+" is not available";
    }
    else
    {
        reader.Close();
        Disconnection();
        objConn =new SqlConnection(strConn);
        string caseNoStr="SELECT CaseNo FROM Patients WHERE
FirstName='"+firstStr+"' and LastName='"+lastStr+"' and
Phone='"+phoneStr+"'";
        Connect();
        SqlDataAdapter objAdapter_SA=new SqlDataAdapter(caseNoStr, objConn);
        DataSet objDataSet_SA=new DataSet("myTable");
        objAdapter_SA.Fill(objDataSet_SA, "myTable");
        caseNumStr=objDataSet_SA.Tables["myTable"].Rows[0
][["CaseNo"].ToString()];
        Disconnection();
        objConn =new SqlConnection(strConn);
        Connect();
        string strSQL="INSERT INTO NewAppointments (CaseNum, FirstName,
LastName, Phone, AppDate, AppTime) VALUES('"+caseNumStr+",
"+firstStr+", "+lastStr+", "+phoneStr+", "+dateStr+", "+timeStr+");
SELECT @@IDENTITY As 'Identity'";
        SqlCommand objComm=new SqlCommand(strSQL, objConn);
        int iID=0;
        try
        {
            Connect();
            iID = Convert.ToInt32(objComm.ExecuteScalar());
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
            Response.End();
        }
        finally
        {
            if (objConn.State == ConnectionState.Open)
            {
                Disconnection();
            }
        }
    }
}
}

```





```

else if(str1.Equals("Spleen channel of foot taiyin (足太陰脾經)"))
    Response.Redirect("ShowImage4.aspx");
else if(str1.Equals("Heart channel of hand shaoyin (手少陰心經)"))
    Response.Redirect("ShowImage5.aspx");
else if(str1.Equals("Small intestine channel of hand taiyang (手太陽小腸經)"))
    Response.Redirect("ShowImage6.aspx");
else if(str1.Equals("Bladder channel of foot taiyang(足太陽膀胱經)"))
    Response.Redirect("ShowImage7.aspx");
else if(str1.Equals("Kidney channel of foot shaoyin (足少陰腎經)"))
    Response.Redirect("ShowImage8.aspx");
else if(str1.Equals("Pericardium channel of hand jueyin(手厥陰心包經)"))
    Response.Redirect("ShowImage9.aspx");
else if(str1.Equals("Triple warmer, sanjiao channel of hand shaoyang (手少陽三焦經)"))
    Response.Redirect("ShowImage10.aspx");
else if(str1.Equals("Gallbladder channel of foot shaoyang (足少陽膽經)"))
    Response.Redirect("ShowImage11.aspx");
else if(str1.Equals("Liver channel of foot jueyin(足厥陰肝經)"))
    Response.Redirect("ShowImage12.aspx");
else if(str1.Equals("Du channel(督脈)"))
    Response.Redirect("ShowImage13.aspx");
else if(str1.Equals("Ren channel(任脈)"))
    Response.Redirect("ShowImage14.aspx");
}
private void saveClick(object sender, System.EventArgs e)
{
    if(L1.Checked)
    {
        colStr+="zhong fu";
        colStr+=",";
    }
    if(L5.Checked)
    {
        colStr+="chi ze";
        colStr+=",";
    }
    if(L6.Checked)
    {
        colStr+="kong zui";
        colStr+=",";
    }
    if(L8.Checked)
    {
        colStr+="lie que";
        colStr+=",";
    }
    if(L9.Checked)
    {
        colStr+="tai yuan";
        colStr+=",";
    }
    if(L10.Checked)
    {
        colStr+="yu ji";
        colStr+=",";
    }
    if(L11.Checked)
    {
        colStr+="shao shang";
        colStr+=",";
    }
}
caseID=Convert.ToInt32(LoginControl.CaseHistory.CaseHistoryNo);

```

```

SqlConnection objConnCH=new SqlConnection(strConn);
string todaydate=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
DateTime.Now.Day,DateTime.Now.Year);
string strSQL1="INSERT INTO CaseHistory(CaseNo, Date, MainType, SubType,
Acupoint)VALUES('"+caseID+"', '"+todaydate+"','Acupuncture-therapy','Lung channel of hand-tai
yin' , '"+colStr+"'); SELECT @@IDENTITY AS 'Identity'";
SqlCommand dbComm1=new SqlCommand(strSQL1, objConnCH);
if(!colStr.Equals(""))
{
    int iID=0;
    try
    {
        objConnCH.Open();
        iID=Convert.ToInt32(dbComm1.ExecuteScalar());
        message1.Text=" The selected items have saved successfully!";
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
        Response.End();
    }
    finally
    {
        if(objConnCH.State==ConnectionState.Open)
        {
            objConnCH.Close();
        }
    }
}
else
{ message1.Text="You did not select any item!";}
}
}
}
}

```

### File:ViewAppointment.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class ViewAppointments : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Label dateLabel;
        protected System.Web.UI.WebControls.TextBox dateTextBox;
        protected System.Web.UI.WebControls.Button SearchButton;
        protected System.Web.UI.WebControls.Label ErrorMessage;
        protected System.Web.UI.WebControls.Calendar VMCalendar;
        protected System.Web.UI.WebControls.DataGrid AppDataGrid;
        protected System.Web.UI.WebControls.Button ExitButton1;
        string strConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
        protected static string dateStr;
        protected static string selectedDate;
        protected static string strQuery9;
        protected static string selectDate;
        protected static int AppID, patientID;
    }
}

```

```

private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
}
private void searchButton(object sender, System.EventArgs e)
{
    //Since each section contains different pages, you might need to go to the other section in the lower page.
    //You need to initialize the begin page to zero, while you begin to a new section
    AppDataGrid.CurrentPageIndex=0;
    dateStr=dateTextBox.Text;
    //Connection set up
    SqlConnection objConn =new SqlConnection(strConn);
    string strQuery="SELECT * FROM NewAppointments WHERE AppDate='"+dateStr+"'";
    SqlCommand obComm=new SqlCommand(strQuery, objConn);
    objConn.Open();
    SqlDataReader reader=obComm.ExecuteReader();
    try
    {
        if(reader.Read().ToString().Equals("True"))
        {
            objConn.Close();
            reader.Close();
            bindGrid();
        }
        else
        {
            ErrorMessage.Text="          There is no appointment on the date.";
            bindGrid();
        }
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
        Response.End();
    }
    finally
    {
        if (objConn.State == ConnectionState.Open)
        {
            objConn.Close();
        }
        reader.Close();
    }
}
private void bindGrid()
{
    string dateStr;
    dateStr=dateTextBox.Text;
    SqlConnection objConnVP =new SqlConnection(strConn);
    string strSQL="SELECT * FROM NewAppointments WHERE AppDate='"+dateStr+"' ORDER by
AppTime";
    objConnVP.Open();
    SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL, objConnVP);
    //Dataset & Adapter & table
    DataSet objDataSet1=new DataSet();
    objAdapter1.Fill(objDataSet1, "dtTodayAppointments");
    objConnVP.Close();
    AppDataGrid.DataSource=objDataSet1.Tables["dtTodayAppointments"];
    //Bind data
    AppDataGrid.DataBind();
}
private void PageIndexChange(object source,
System.Web.UI.WebControls.DataGridPageChangedEventArgs e)

```

```

    {
        AppDataGrid.CurrentPageIndex=e.NewPageIndex;
        bindGrid();
    }
    private void ExitClick(object sender, System.EventArgs e)
    {
        Response.Redirect("AcupPage.aspx");
    }
    private void calendarChange1(object sender, System.EventArgs e)
    {
        ErrorMessage.Text="";
        dateTextBox.Text=VMCalendar.SelectedDate.ToShortDateString();
    }
    private void processCommand(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        string appIDStr=e.CommandArgument.ToString();
        string todayDate=String.Format("{0:D}/{1:D}/{2:D2}",DateTime.Now.Month,
        DateTime.Now.Day,DateTime.Now.Year);
        if(e.CommandName.Equals("GetViewAppointments"))
        {
            PreRegisterPatients(appIDStr);
            if(selectedDate.Equals(todayDate))
            {
                Server.Transfer("CaseHistory.aspx?patientNo=" + patientID);//Transfer to CaseHistory.aspx
            }
            else
            {
                ErrorMessage.Text=" You cannot create the case hisetory page for the future or past
                appointment.";
            }
        }
    }
    private void PreRegisterPatients( string appIDStr)
    {
        AppID=Convert.ToInt32(appIDStr);
        SqlConnection objConn_PP =new SqlConnection(strConn);
        string strQy="SELECT AppointmentID, CaseNum, AppDate FROM NewAppointments WHERE
        AppointmentID='"+appIDStr+"'";
        //From AppointmentID to get patient's name, phone number and the date of appointment.
        objConn_PP.Open();
        SqlDataAdapter objAdapter2=new SqlDataAdapter(strQy, objConn_PP);
        DataSet objDataSet2=new DataSet();
        objAdapter2.Fill(objDataSet2, "dtNewAppoints");
        objConn_PP.Close();
        //Modify the in-memory records in the DataSet
        DataTable tbl2=objDataSet2.Tables["dtNewAppoints"];
        tbl2.PrimaryKey=new DataColumn[]{tbl2.Columns["AppointmentID"]};
        DataRow row2=tbl2.Rows.Find(AppID);
        patientID=Convert.ToInt32(row2["CaseNum"].ToString());
        selectedDate=row2["AppDate"].ToString();
    }
}
}
}

```

#### File: ViewEditAppointment.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

```

```

using System.Data.SqlClient;
namespace LoginControl
{
    public class ViewEditAppointments : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label TitleLabel;
        protected System.Web.UI.WebControls.Button ExitButton1;
        protected System.Web.UI.WebControls.Label dateLabel1;
        protected System.Web.UI.WebControls.Button SearchButton;
        protected System.Web.UI.WebControls.DataGrid AppDataGrid;
        protected System.Web.UI.WebControls.Label phoneLabel;
        protected System.Web.UI.WebControls.TextBox phoneTextBox;
        protected System.Web.UI.WebControls.TextBox dateTextBox;
        protected System.Web.UI.WebControls.Button resetBtn;
        protected System.Web.UI.WebControls.Calendar VMCalendar;
        private string dateStr;
        private string phoneStr;
        private string strSQL;
        private string strConnection="server=(local)\NetSDK; database=ClinicSystem1; integrated security=true";
        private string strSelect="SELECT * FROM NewAppointments";
        protected System.Web.UI.WebControls.Button Button1;
        protected System.Web.UI.WebControls.Label EMessage;
        protected SqlConnection objConnection;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.staffName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
        }
        private void ResetClick(object sender, System.EventArgs e)
        {
            EMessage.Text="";
            phoneTextBox.Text="";
            dateTextBox.Text="";
        }
        private void ExitClick(object sender, System.EventArgs e)
        {
            Response.Redirect("StaffPage.aspx");
        }
        private void AppointmentClick(object sender, System.EventArgs e)
        {
            Response.Redirect("ScheduleAnAppointment.aspx");
        }
        private void SelectedDateChanged(object sender, System.EventArgs e)
        {
            EMessage.Text="";
            dateTextBox.Text=VMCalendar.SelectedDate.ToShortDateString();
        }
        private void SearchClick(object sender, System.EventArgs e)
        {
            phoneStr=phoneTextBox.Text;
            dateStr=dateTextBox.Text;
            string testStr1, testStr2;
            //Connection set up
            string strConn="server=(local)\NetSDK; database=ClinicSystem1; integrated security=true";
            SqlConnection objConn =new SqlConnection(strConn);
            string strQuery1="SELECT AppDate FROM NewAppointments WHERE AppDate="+dateStr+"";
            string strQuery2="SELECT Phone FROM NewAppointments WHERE Phone="+phoneStr+"";
            SqlCommand obComm1=new SqlCommand(strQuery1, objConn);
            SqlCommand obComm2=new SqlCommand(strQuery2, objConn);
            objConn.Open();
            SqlDataReader reader1=obComm1.ExecuteReader();
            if(reader1.Read().ToString().Equals("True"))
                testStr1="YES";
            else
        }
    }
}

```

```

        testStr1="NO";
reader1.Close();
SqlDataReader reader2=obComm2.ExecuteReader();
if(reader2.Read().ToString().Equals("True"))
    testStr2="YES";
else
    testStr2="NO";
reader2.Close();
try
{
    if((testStr1.ToString()=="YES")||(testStr2.ToString()=="YES"))
    {
        objConn.Close();
        bindGrid();
    }
    else
    {
        objConn.Close();
        bindGrid();
    }
}
catch (Exception ex)
{
    Response.Write(ex.Message);
    Response.End();
}
finally
{
    if (objConn.State == ConnectionState.Open)
    {
        objConn.Close();
    }
}
}
private void bindGrid()
{
    dateStr=dateTextBox.Text;
    phoneStr=phoneTextBox.Text;
    objConnection =new SqlConnection(strConnection);
    if(!dateStr.ToString().Equals("") && !phoneStr.ToString().Equals(""))
    {
        strSQL="SELECT AppointmentID, FirstName, LastName, Phone, AppDate, AppTime FROM
NewAppointments WHERE AppDate='"+dateStr+"' and Phone='"+phoneStr+"' ORDER by
AppDate desc";
        SqlCommand obCom=new SqlCommand(strSQL, objConnection);
        Connect();
        SqlDataReader rder1=obCom.ExecuteReader();
        if(rder1.Read().ToString().Equals("False"))
        {
            EMessage.Text="It cannot be found in the database!";
        }
        rder1.Close();
        SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL, objConnection);
        //Dataset & Adapter & table
        DataSet objDataSet=new DataSet();
        objAdapter.Fill(objDataSet, "dtNewAppointments");
        objConnection.Close();
        AppDataGrid.DataSource=objDataSet.Tables["dtNewAppointments"];
        //Bind data
        AppDataGrid.DataBind();
        Disconnection();
    }
    else if(!dateStr.ToString().Equals(""))
    {
        strSQL="SELECT AppointmentID, FirstName, LastName, Phone, AppDate, AppTime FROM
NewAppointments WHERE AppDate='"+dateStr+"' ORDER by Apptime";
        SqlCommand obCom=new SqlCommand(strSQL, objConnection);
    }
}

```

```

Connect();
SqlDataReader rder1=obCom.ExecuteReader();
if(rder1.Read().ToString().Equals("False"))
{
    EMessage.Text="It cannot be found in the database!";
}
rder1.Close();
SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL, objConnection);
//Dataset & Adapter & table
DataSet objDataSet=new DataSet();
objAdapter.Fill(objDataSet, "dtNewAppointments");
AppDataGrid.DataSource=objDataSet.Tables["dtNewAppointments"];
//Bind data
AppDataGrid.DataBind();
Disconnection();
}
else if(!phoneStr.ToString().Equals(""))
{
    strSQL="SELECT AppointmentID, FirstName, LastName, Phone, AppDate, AppTime FROM
NewAppointments WHERE Phone='"+phoneStr+"' ORDER by AppDate desc";
SqlCommand obCom=new SqlCommand(strSQL, objConnection);
Connect();
SqlDataReader rder1=obCom.ExecuteReader();
if(rder1.Read().ToString().Equals("False"))
{
    EMessage.Text="It cannot be found in the database!";
}
rder1.Close();
SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL, objConnection);
//Dataset & Adapter & table
DataSet objDataSet=new DataSet();
objAdapter.Fill(objDataSet, "dtNewAppointments");
AppDataGrid.DataSource=objDataSet.Tables["dtNewAppointments"];
//Bind data
AppDataGrid.DataBind();
Disconnection();
}
else
{EMessage.Text="Please reenter again"; }
}
private void Connect()
{
    if(objConnection==null)
        objConnection= new SqlConnection(strConnection);
    if(objConnection.State==ConnectionState.Closed)
        objConnection.Open();
}
private void Disconnection()
{
    objConnection.Close();
}
private void AppDataGrid_PageIndexChanged(object source,
System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    AppDataGrid.CurrentPageIndex=e.NewPageIndex;
    bindGrid();
}
private void DeletedRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
{
    // Retrieve the ID of the Appointment to be deleted
    int AppointmentID = Convert.ToInt32(e.Item.Cells[0].Text);
    AppDataGrid.EditItemIndex = -1;
    DeleteAppointments(AppointmentID);
    // Display the remaining items in the DataGrid
    DataSet ds = new DataSet();
    AppDataGrid.DataSource = ds.Tables["dtNewAppointments"];
    AppDataGrid.DataBind();
}

```



```

        bindGrid();
        EMessage.Text="The appointment has been deleted!";
    }
    private void DeleteAppointments( int AppointmentID)
    {
        // Create and load a DataSet with records from ClinicSystem1's NewAppointments table
        objConnection = new SqlConnection(strConnection);
        objConnection.Open();
        SqlDataAdapter adapter = new SqlDataAdapter(strSelect, objConnection);
        DataSet ds = new DataSet();
        adapter.Fill(ds, "dtNewAppointments");
        objConnection.Close();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl = ds.Tables["dtNewAppointments"];
        tbl.PrimaryKey = new DataColumn[] {tbl.Columns["AppointmentID"]};
        DataRow row = tbl.Rows.Find(AppointmentID);
        row.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb = new SqlCommandBuilder(adapter);
        objConnection = new SqlConnection(strConnection);
        objConnection.Open();
        adapter.Update(ds,"dtNewAppointments");
        objConnection.Close();
    }
    private void EditRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        AppDataGrid.EditItemIndex=e.Item.ItemIndex;
        bindGrid();
    }
    private void CancelRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        AppDataGrid.EditItemIndex=-1;
        bindGrid();
    }
    private void UpdateRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the field values in the edited row
        int AppointmentID = Convert.ToInt32(e.Item.Cells[0].Text);
        TextBox DateTextBox= (TextBox)e.Item.Cells[4].Controls[0];
        TextBox TimeTextBox= (TextBox)e.Item.Cells[5].Controls[0];
        string date1=DateTextBox.Text;
        string time1=TimeTextBox.Text;
        //If the appointment was taken, you cannot update it. You can only update it while the appointment
        // is not taken.
        string strCk="SELECT * FROM NewAppointments WHERE AppDate='"+date1+"' and
        AppTime='"+time1+"' ORDER by AppTime";
        objConnection =new SqlConnection(strConnection);
        SqlCommand obCommCk=new SqlCommand(strCk, objConnection);
        objConnection.Open();
        SqlDataReader readerCK=obCommCk.ExecuteReader();
        if(readerCK.Read().ToString().Equals("False"))
        {
            objConnection.Close();
            readerCK.Close();
            AppDataGrid.EditItemIndex = -1;
            UpdateAppointment(AppointmentID, date1, time1);
            DataSet ds = new DataSet();
            AppDataGrid.DataSource = ds.Tables["dtNewAppointments"];
            AppDataGrid.DataBind();
            bindGrid();
        }
        else
        {
            objConnection.Close();
            readerCK.Close();
            EMessage.Text="                The appointment was taken, please change the other time!";
        }
    }
}

```

```

    }
    private void UpdateAppointment(int AppointmentID, string date1, string time1)
    {
        // Create and load a DataSet with records from Northwind.Products table
        objConnection = new SqlConnection(strConnection);
        objConnection.Open();
        SqlDataAdapter adapter = new SqlDataAdapter(strSelect, objConnection);
        DataSet ds = new DataSet();
        adapter.Fill(ds, "dtNewAppointments");
        objConnection.Close();
        // Modify the in-memory records in the DataSet
        DataTable tbl = ds.Tables["dtNewAppointments"];
        tbl.PrimaryKey = new DataColumn[] {tbl.Columns["AppointmentID"]};
        DataRow row = tbl.Rows.Find(AppointmentID);
        row["AppDate"]=date1;
        row["Apptime"]=time1;
        EMessage.Text="          The changed data has been saved into the database.";
        // Reconnect the DataSet and update the database
        SqlCommandBuilder cb = new SqlCommandBuilder(adapter);
        objConnection.Open();
        adapter.Update(ds, "dtNewAppointments");
        objConnection.Close();
    }
}
}
}

```

### File:ViewModifyPatientInfo.aspx.cs

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class ViewModifyPatientInfo1 : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.TextBox LNameTxtBox1;
        protected System.Web.UI.WebControls.ListBox FNListBox;
        protected System.Web.UI.WebControls.Label caseNoLabel1;
        protected System.Web.UI.WebControls.Label CaseNoLabel;
        protected System.Web.UI.WebControls.Label dateLabel1;
        protected System.Web.UI.WebControls.Label DateLabel2;
        protected System.Web.UI.WebControls.Label LNLabel;
        protected System.Web.UI.WebControls.TextBox LNTxtBox;
        protected System.Web.UI.WebControls.Label MInitLabel;
        protected System.Web.UI.WebControls.TextBox MInitTxtBox;
        protected System.Web.UI.WebControls.Label birthLabel;
        protected System.Web.UI.WebControls.TextBox BirthTxtBox;
        protected System.Web.UI.WebControls.Label FNLabel;
        protected System.Web.UI.WebControls.TextBox FNTxtBox;
        protected System.Web.UI.WebControls.Label SexLabel;
        protected System.Web.UI.WebControls.TextBox SexTxtBox;
        protected System.Web.UI.WebControls.Label SSNLabel;
        protected System.Web.UI.WebControls.TextBox SSNTxtBox;
        protected System.Web.UI.WebControls.Label DLICENSELabel;
        protected System.Web.UI.WebControls.TextBox DLTxtBox;
        protected System.Web.UI.WebControls.Label MartialLabel;
        protected System.Web.UI.WebControls.TextBox MSTxtBox1;
        protected System.Web.UI.WebControls.Label phoneLabel;
    }
}

```

```

protected System.Web.UI.WebControls.TextBox PhoneTxtBox;
protected System.Web.UI.WebControls.Label Label15;
protected System.Web.UI.WebControls.TextBox E_MailTxtBox;
protected System.Web.UI.WebControls.Label addressLabel;
protected System.Web.UI.WebControls.TextBox AddressTxtBox;
protected System.Web.UI.WebControls.Label OccupLabel;
protected System.Web.UI.WebControls.TextBox OccupTxtBox;
protected System.Web.UI.WebControls.Label BPhoneLabel;
protected System.Web.UI.WebControls.TextBox BPhoneTxtBox;
protected System.Web.UI.WebControls.Label employerLabel;
protected System.Web.UI.WebControls.TextBox EmployerTxtBox;
protected System.Web.UI.WebControls.Label BAddressLabel;
protected System.Web.UI.WebControls.TextBox BAddressTxtBox;
protected System.Web.UI.WebControls.Label contactLabel;
protected System.Web.UI.WebControls.TextBox ContactTxtBox;
protected System.Web.UI.WebControls.Label relationLabel;
protected System.Web.UI.WebControls.TextBox RelationTxtBox1;
protected System.Web.UI.WebControls.Label EPhoneLabel;
protected System.Web.UI.WebControls.TextBox EPhoneTxtBox1;
protected System.Web.UI.WebControls.Label InsuranceLabel;
protected System.Web.UI.WebControls.TextBox InsuranceTxtBox1;
protected System.Web.UI.WebControls.Label PolicyLabel;
protected System.Web.UI.WebControls.TextBox PolicyTxtBox;
protected System.Web.UI.WebControls.Label Label8;
protected System.Web.UI.WebControls.Button exitBtn;
private string newConn="server=(local)\\NetSDK; database=ClinicSystem1; integrated security=true";
protected System.Web.UI.WebControls.Label TitleLabel;
protected System.Web.UI.WebControls.Label LNLabel1;
protected System.Web.UI.WebControls.Button SearchBtn;
protected System.Web.UI.WebControls.Button updateBtn;
protected System.Web.UI.WebControls.Label BillingAddress;
protected System.Web.UI.WebControls.TextBox BATextBox;
protected System.Web.UI.WebControls.Label IPhoneLabel;
protected System.Web.UI.WebControls.TextBox IPhoneTxtBox;
private string firstname, lastname;
private string lName, fName, mName, bDate,sex,ssn, dLicense, Mstatus, phone, email,addr,
occup,employer,bPhone, BAddr,conact, relation, ePhone, InsCo, Insphone, InsAddr, policyNo;
protected static string No;
protected static string FirstN, LastN;
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.staffName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
}
private void searchClick(object sender, System.EventArgs e)
{
    Label8.Text="";
    listFirstName();
}
private void listFirstName()
{
    lastname= LNameTxtBox1.Text;
    if(!lastname.Equals(""))
    {
        string strQuery6="SELECT FirstName, CaseNo FROM Patients WHERE
        LastName='"+lastname+"'";
        SqlConnection objConn6 = new SqlConnection(newConn);
        SqlCommand objCommand=new SqlCommand(strQuery6, objConn6);
        objConn6.Open();
        FNListBox.DataSource=objCommand.ExecuteReader();
        FNListBox.DataTextField="FirstName";
        FNListBox.DataValueField="CaseNo";
        FNListBox.DataBind();
        objConn6.Close();
    }
}

```

```

    }
}
private void updataClick(object sender, System.EventArgs e)
{
    Label8.Text="";
    int caseNo=Convert.ToInt32(No);
    if(!No.Equals(""))
    {
        string strPatient="SELECT * FROM Patients WHERE CaseNo="+caseNo+"";
        SqlConnection objConnP =new SqlConnection(newConn);
        objConnP.Open();
        SqlDataAdapter objAdapterP=new SqlDataAdapter(strPatient, objConnP);
        DataSet objDataSetP=new DataSet("dtNewPatients");
        objAdapterP.Fill(objDataSetP, "dtNewPatients");
        objConnP.Close();
        //Modify the in-memory records in the DataSet
        DataTable tbl=objDataSetP.Tables["dtnewPatients"];
        tbl.PrimaryKey=new DataColumn[]{tbl.Columns["CaseNo"]};
        DataRow row=tbl.Rows.Find(caseNo);
        // Retrive data from patient table and put them into each TextBoxes
        lName=row["LastName"].ToString();
        fName=row["FirstName"].ToString();
        mName=row["MInit"].ToString();
        bDate=row["BirthDay"].ToString();
        sex=row["Sex"].ToString();
        ssn=row["SSN"].ToString();
        dLicense=row["DriverLicense"].ToString();
        Mstatus=row["MaritalStatus"].ToString();
        phone=row["Phone"].ToString();
        email=row["E_mail"].ToString();
        addr=row["Address"].ToString();
        occup=row["Occupation"].ToString();
        employer=row["Employer"].ToString();
        bPhone=row["BusinessPhone"].ToString();
        BAddr=row["BusinessAddress"].ToString();
        conact=row["ContactPerson"].ToString();
        relation=row["Relation"].ToString();
        ePhone=row["EmergencyPhone"].ToString();
        InsCo=row["InsuranceCo"].ToString();
        policyNo=row["PolicyNo"].ToString();
        Insphone=row["InsurancePhone"].ToString();
        InsAddr=row["BillingAddress"].ToString();
        if((!lNTxtBox.Text.Equals(lName))||(!fNTxtBox.Text.Equals(fName))||
        (!mInitTxtBox.Text.Equals(mName))|| (!BirthTxtBox.Text.Equals(bDate))||
        (!SexTxtBox.Text.Equals(sex))||(!SSNTextBox.Text.Equals(ssn))||
        (!DLTxtBox.Text.Equals(dLicense))||(!MSTxtBox1.Text.Equals(Mstatus))||
        (!PhoneTxtBox.Text.Equals(phone))||(!E_MailTxtBox.Text.Equals(email))||
        (!AddressTxtBox.Text.Equals(addr))||(!OccupTxtBox.Text.Equals(occup))||
        (!EmployerTxtBox.Text.Equals(employer))|| (!BPhoneTxtBox.Text.Equals(bPhone))||
        (!BAddressTxtBox.Text.Equals(BAddr))||(!ContactTxtBox.Text.Equals(conact))||
        (!RelationTxtBox1.Text.Equals(relation))|| (!EPhoneTxtBox1.Text.Equals(ePhone))||
        (!InsuranceTxtBox1.Text.Equals(InsCo))||(!PolicyTxtBox.Text.Equals(policyNo))||
        (!IPhoneTxtBox.Text.Equals(Insphone))||(!BATextBox.Text.Equals(InsAddr))
        {
            if((!SexTxtBox.Text.Equals(""))&&!ContactTxtBox.Text.Equals(""))&&
            (!BirthTxtBox.Text.Equals(""))&&!SSNTextBox.Text.Equals(""))&&
            (!PhoneTxtBox.Text.Equals(""))&&!EPhoneTxtBox1.Text.Equals(""))
            {
                if(SSNTextBox.Text.Length.Equals(11) && SSNTextBox.Text.Substring(3,1).Equals("-")
                && SSNTextBox.Text.Substring(6,1).Equals("-"))
                {
                    if(PhoneTxtBox.Text.Length.Equals(13) &&
                    PhoneTxtBox.Text.Substring(0,1).Equals("(") &&
                    PhoneTxtBox.Text.Substring(4,1).Equals(")") &&
                    PhoneTxtBox.Text.Substring(8,1).Equals("-") &&
                    EPhoneTxtBox1.Text.Length.Equals(13) &&
                    EPhoneTxtBox1.Text.Substring(0,1).Equals("(") &&

```

```

EPhoneTextBox1.Text.Substring(4,1).Equals("") &&
EPhoneTextBox1.Text.Substring(8,1).Equals("-")
{
    if(((!BPhoneTextBox.Equals("") && ( BPhoneTextBox.Text.Length.Equals(13))
        && (BPhoneTextBox.Text.Substring(0,1).Equals("")) &&
        (BPhoneTextBox.Text.Substring(4,1).Equals("")) &&
        (BPhoneTextBox.Text.Substring(8,1).Equals("-"))))|
        (BPhoneTextBox.Equals("")))
    {
        //Cannot modify the first name and last name
        // The each content in the TextBox will be write into the Patients' table
        // The last name is used to find each person in the table, so it will not
        //allow to modify last nam
        row["MInit"]=MInitTextBox.Text;
        row["Birthday"]=BirthTextBox.Text;
        row["Sex"]=SexTextBox.Text;
        row["SSN"]=SSNTextBox.Text;
        row["DriverLicense"]=DLTextBox.Text;
        row["MaritalStatus"]=MSTxtBox1.Text;
        row["Phone"]=PhoneTextBox.Text;
        row["E_mail"]=E_MailTextBox.Text;
        row["Address"]=AddressTextBox.Text;
        row["Occupation"]=OccupTextBox.Text;
        row["Employer"]=EmployerTextBox.Text;
        row["BusinessPhone"]=BPhoneTextBox.Text;
        row["BusinessAddress"]=BAddressTextBox.Text;
        row["ContactPerson"]=ContactTextBox.Text;
        row["Relation"]=RelationTextBox1.Text;
        row["EmergencyPhone"]=EPhoneTextBox1.Text;
        row["InsuranceCo"]=InsuranceTextBox1.Text;
        row["PolicyNo"]=PolicyTextBox.Text;
        row["InsurancePhone"]=IPhoneTextBox.Text;
        row["BillingAddress"]= BATextBox.Text;
        //Reconnect the Dataset and update the database
        SqlCommandBuilder cbP=new SqlCommandBuilder(objAdapterP);
        objConnP.Open();
        objAdapterP.Update(objDataSetP, "dtNewPatients");
        objConnP.Close();
        MinitTextBox.Text=row["MInit"].ToString();
        BirthTextBox.Text=row["Birthday"].ToString();
        SexTextBox.Text=row["Sex"].ToString();
        SSNTextBox.Text=row["SSN"].ToString();
        DLTextBox.Text=row["DriverLicense"].ToString();
        MSTxtBox1.Text=row["MaritalStatus"].ToString();
        PhoneTextBox.Text=row["Phone"].ToString();
        E_MailTextBox.Text=row["E_mail"].ToString();
        AddressTextBox.Text=row["Address"].ToString();
        OccupTextBox.Text=row["Occupation"].ToString();
        EmployerTextBox.Text=row["Employer"].ToString();
        BPhoneTextBox.Text=row["BusinessPhone"].ToString();
        BAddressTextBox.Text=row["BusinessAddress"].ToString();
        ContactTextBox.Text=row["ContactPerson"].ToString();
        RelationTextBox1.Text=row["Relation"].ToString();
        EPhoneTextBox1.Text=row["EmergencyPhone"].ToString();
        InsuranceTextBox1.Text=row["InsuranceCo"].ToString();
        PolicyTextBox.Text=row["policyNo"].ToString();
        IPhoneTextBox.Text=row["InsurancePhone"].ToString();
        BATextBox.Text=row["BillingAddress"].ToString();
        if(!FNTxtBox.Text.Equals(fName))|(!LNTxtBox.Text.Equals(lName))
        {
            Label8.Text="you cannot change either the last name or first
            name.";
            LNTxtBox.Text=row["LastName"].ToString();
            FNTxtBox.Text=row["FirstName"].ToString();
        }
        else if((row["FirstName"].ToString().Equals(
        FNTxtBox.Text))|(lName.Equals(LNTxtBox.Text)))
    }
}

```

```

    {
        //Update the data successfully
        Label8.Text=" You have updated the data successfully";
    }
}
else if(BPhoneTextBox.Text.Equals(""))
{
    // The each content in the TextBox will be write into the Patients' table
    // The last name is used to find each person in the table, so it will not
    // allow to modify last name
    //Cannot modify the first name and last name
    row["MInit"]=MInitTextBox.Text;
    row["Birthday"]=BirthTextBox.Text;
    row["Sex"]=SexTextBox.Text;
    row["SSN"]=SSNTextBox.Text;
    row["DriverLicense"]=DLTextBox.Text;
    row["MaritalStatus"]=MSTxtBox1.Text;
    row["Phone"]=PhoneTextBox.Text;
    row["E_mail"]=E_MailTextBox.Text;
    row["Address"]=AddressTextBox.Text;
    row["Occupation"]=OccupTextBox.Text;
    row["Employer"]=EmployerTextBox.Text;
    row["BusinessPhone"]=BPhoneTextBox.Text;
    row["BusinessAddress"]=BAddressTextBox.Text;
    row["ContactPerson"]=ContactTextBox.Text;
    row["Relation"]=RelationTextBox1.Text;
    row["EmergencyPhone"]=EPhoneTextBox1.Text;
    row["InsuranceCo"]=InsuranceTextBox1.Text;
    row["PolicyNo"]=PolicyTextBox.Text;
    row["InsurancePhone"]=IPhoneTextBox.Text;
    row["BillingAddress"]=BATextBox.Text;
    //Reconnect the Dataset and update the database
    SqlCommandBuilder cbP=new SqlCommandBuilder(objAdapterP);
    objConnP.Open();
    objAdapterP.Update(objDataSetP, "dtNewPatients");
    objConnP.Close();
    MInitTextBox.Text=row["MInit"].ToString();
    BirthTextBox.Text=row["Birthday"].ToString();
    SexTextBox.Text=row["Sex"].ToString();
    SSNTextBox.Text=row["SSN"].ToString();
    DLTextBox.Text=row["DriverLicense"].ToString();
    MSTxtBox1.Text=row["MaritalStatus"].ToString();
    PhoneTextBox.Text=row["Phone"].ToString();
    E_MailTextBox.Text=row["E_mail"].ToString();
    AddressTextBox.Text=row["Address"].ToString();
    OccupTextBox.Text=row["Occupation"].ToString();
    EmployerTextBox.Text=row["Employer"].ToString();
    BPhoneTextBox.Text=row["BusinessPhone"].ToString();
    BAddressTextBox.Text=row["BusinessAddress"].ToString();
    ContactTextBox.Text=row["ContactPerson"].ToString();
    RelationTextBox1.Text=row["Relation"].ToString();
    EPhoneTextBox1.Text=row["EmergencyPhone"].ToString();
    InsuranceTextBox1.Text=row["InsuranceCo"].ToString();
    PolicyTextBox.Text=row["policyNo"].ToString();
    IPhoneTextBox.Text=row["InsurancePhone"].ToString();
    BATextBox.Text=row["BillingAddress"].ToString();
    if(!FNTxtBox.Text.Equals(fName))(!(LNTxtBox.Text.Equals
    (lName)))
    {
        Label8.Text="You cannot change either the last name or
        first name.";
        LNTxtBox.Text=row["LastName"].ToString();
        FNTxtBox.Text=row["FirstName"].ToString();
    }
}
else if((row["FirstName"].ToString().Equals(FNTxtBox.Text))
||(!Name.Equals(LNTxtBox.Text)))
{

```

```

        //Update the data successfully
        Label8.Text=" You have updated the data successfully";
    }
}
else if(!BPhoneTextBox.Text.ToString().Equals(""))
{
    BPhoneTextBox.Text=row["BusinessPhone"].ToString();
    Label8.Text="Incorrect phone format. Please use the following phone
    format. Ex.(000)000-0000";
}
}
else
{
    PhoneTextBox.Text=row["Phone"].ToString();
    Label8.Text="Incorrect phone format. Please use the following phone format.
    Ex.(888)888-8888";
}
}
else
{
    Label8.Text="The Social security number should use as following format. Ex.000-
    00-0000";
    SSNTextBox.Text=ssn;
}
}
else
{
    SexTextBox.Text=sex;
    SSNTextBox.Text=ssn;
    PhoneTextBox.Text=phone;
    BirthTextBox.Text=bDate;
    EPhoneTextBox1.Text=ePhone;
    ContactTextBox.Text=conact;
    LNTxtBox.Text=lName;
    FNTxtBox.Text=fName;
    Label8.Text=" First name, last name, SSN, sex, birday, phone, contact person
    and emergency phone number TextBoxes cannot be empty.The last name or first
    name cannot be modified.";
}
}
else
{
    Label8.Text="You do not make any change!";
}
}
}
private void selectedItemChanged(object sender, System.EventArgs e)
{
    Label8.Text="";
    lastname=LNameTextBox1.Text;
    firstname=FNListBox.SelectedItem.Text;
    string strQuery7="SELECT * FROM Patients WHERE LastName='"+lastname+"'";
    SqlConnection objConn7 =new SqlConnection(newConn);
    SqlDataAdapter objAdapter=new SqlDataAdapter(strQuery7, objConn7);
    DataSet objDataSet=new DataSet("dtPatients");
    objAdapter.Fill(objDataSet, "dtPatients");
    CaseNoLabel.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]
    ["CaseNo"].ToString();
    No=CaseNoLabel.Text;
    DateLabel2.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]
    ["RegDate"].ToString();
    LNTxtBox.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]
    ["LastName"].ToString();
    FNTxtBox.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]
    ["FirstName"].ToString();
    MinitTextBox.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]["Minit"].ToString();
    BirthTextBox.Text=objDataSet.Tables["dtPatients"].Rows[FNListBox.SelectedIndex]

```





```

protected System.Web.UI.WebControls.Label errorMsg;
string strConn="server=(local)\\NetSdk; database=ClinicSystem1; integrated security=true";
protected SqlConnection objConn;
protected System.Web.UI.WebControls.Label phoneLabel;
protected System.Web.UI.WebControls.TextBox phoneTextBox;
protected System.Web.UI.WebControls.Button searchButton;
protected System.Web.UI.WebControls.Button ClearButton;
protected System.Web.UI.WebControls.Label LastNameLabel;
protected System.Web.UI.WebControls.TextBox LNameTextBox;
protected System.Web.UI.WebControls.Label Label1;
protected static string strSQL_VM;
private void Page_Load(object sender, System.EventArgs e)
{
    string UserName1="none";
    // Do not allow loading the page if the user did not login from default page
    UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
    if(UserName1.Equals("none"))
        Response.Redirect("DefaultPage.aspx");
}
private void ChangePage(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
{
    DataGridVM.CurrentPageIndex=e.NewPageIndex;
    bindGrid();
}
private void ClearClick(object sender, System.EventArgs e)
{
    phoneTextBox.Text="";
    LNameTextBox.Text="";
    errorMsg.Text="";
}
private void exitClick(object sender, System.EventArgs e)
{
    Response.Redirect("AcupPage.aspx");
}
private void searchClick(object sender, System.EventArgs e)
{
    errorMsg.Text="";
    string Lname, phoneNo;
    Lname=LNameTextBox.Text;
    phoneNo=phoneTextBox.Text;
    if(!Lname.Equals("") && (!phoneNo.Equals("")))
    {
        objConn =new SqlConnection(strConn);
        if(phoneNo.Length.Equals(13))
        {
            if((phoneNo.Substring(0,1).Equals("")) && (phoneNo.Substring(4,1).Equals("")) &&
            (phoneNo.Substring(8,1).Equals("-")))
            {
                strSQL_VM="SELECT CaseNo, LastName, FirstName, Phone FROM Patients
                WHERE Phone="+phoneNo+" and LastName="+Lname+"";
                SqlCommand obComm=new SqlCommand(strSQL_VM, objConn);
                objConn.Open();
                SqlDataReader reader=obComm.ExecuteReader();
                try
                {
                    if(reader.Read().ToString().Equals("True"))
                    {
                        objConn.Close();
                        reader.Close();
                        bindGrid();
                    }
                    else
                    {
                        errorMsg.Text="Incorrect phone number or last name! Try again or
                        input either one.";
                    }
                }
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            Response.Write(ex.Message);
            Response.End();
        }
        finally
        {
            if (objConn.State == ConnectionState.Open)
            {
                objConn.Close();
            }
            reader.Close();
        }
    }
    else
    {errorMsg.Text="Incorrect format in the phone number!";}
}
else
{errorMsg.Text="Incorrect phone number!! Please input again.";}
}
else if((!Lname.Equals("")) || (!phoneNo.Equals("")))
{
    objConn =new SqlConnection(strConn);
    if(!Lname.Equals(""))
    {
        strSQL_VM="SELECT CaseNo, LastName, FirstName, Phone FROM Patients WHERE
        LastName='"+Lname+"'";
        SqlCommand obComm=new SqlCommand(strSQL_VM, objConn);
        objConn.Open();
        SqlDataReader reader=obComm.ExecuteReader();
        try
        {
            if(reader.Read().ToString().Equals("True"))
            {
                objConn.Close();
                reader.Close();
                bindGrid();
            }
            else
            {
                errorMsg.Text="The last name cannot found in the database.!\n\n" Please input
                again or input phone number instead of last name.";
            }
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
            Response.End();
        }
        finally
        {
            if (objConn.State == ConnectionState.Open)
            {
                objConn.Close();
            }
            reader.Close();
        }
    }
}
if(!phoneNo.Equals(""))
{
    if(phoneNo.Length.Equals(13))
    {
        if(phoneNo.Substring(0,1).Equals("(") && phoneNo.Substring(4,1).Equals(")") &&
        phoneNo.Substring(8,1).Equals("-"))
        {
            strSQL_VM="SELECT CaseNo, LastName, FirstName, Phone FROM Patients
            WHERE Phone='"+phoneNo+"'";

```

```

SqlCommand obComm=new SqlCommand(strSQL_VM, objConn);
objConn.Open();
SqlDataReader reader=obComm.ExecuteReader();
try
{
    if(reader.Read().ToString().Equals("True"))
    {
        objConn.Close();
        reader.Close();
        bindGrid();
    }
    else
    {
        errorMsg.Text="The last name cannot found in the database!"+"\n"+"
Please input again or input phone number instead of last name.";
    }
}
catch (Exception ex)
{
    Response.Write(ex.Message);
    Response.End();
}
finally
{
    if (objConn.State == ConnectionState.Open)
    {
        objConn.Close();
    }
    reader.Close();
}
}
else
{errorMsg.Text="Wrong format in the phone number! Please input again.";}
}
else
{errorMsg.Text="Wrong format in the phone number! Please input again.";}
}
}
else if(Lname.Equals("") && phoneNo.Equals(""))
{errorMsg.Text="Please Input Last Name, Phone number or both of them!";}
}
private void bindGrid()
{
    objConn =new SqlConnection(strConn);
    objConn.Open();
    SqlDataAdapter objAdapter1=new SqlDataAdapter(strSQL_VM, objConn);
    //Dataset & Adapter & table
    DataSet objDataSet1=new DataSet();
    objAdapter1.Fill(objDataSet1, "dtPatients");
    objConn.Close();
    DataGridVM.DataSource=objDataSet1.Tables["dtPatients"];
    //Bind data
    DataGridVM.DataBind();
}
private void ProcessCommand(object source, System.Web.UI.WebControls
.DataGridCommandEventArgs e)
{
    string caseID=e.CommandArgument.ToString();
    if(e.CommandName.Equals("GetPatientInfo"))
    {
        Server.Transfer("CaseHistory.aspx?patientNo=" +caseID);
    }
}
}
}
}

```

## File:ViewSelectedItems.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
namespace LoginControl
{
    public class ViewSelectedItems : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.Label SelectedItemsLabel;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.DataGrid SIDataGrid;
        protected System.Web.UI.WebControls.Button Button1;
        protected static string newConn7="server=(local)\\NetSDK; database=ClinicSystem1; integrated
security=true";
        protected static string strSQL7;
        protected SqlConnection objConn7;
        int CaseID;
        protected string dateStr;
        private int itemCt;
        string caseno;
        protected static string type;
        private void Page_Load(object sender, System.EventArgs e)
        {
            string UserName1="none";
            // Do not allow loading the page if the user did not login from default page
            UserName1=Convert.ToString(LoginControl.UserControl.AcupName);
            if(UserName1.Equals("none"))
                Response.Redirect("DefaultPage.aspx");
            if(LoginControl.Inquiry.method2.Equals("Inquiry"))
            { type="Inquiry";}
            if(LoginControl.Look.method1.Equals("Look"))
            {type="Look";}
            if(LoginControl.ListenSmell.method3.Equals("ListenSmell"))
            {type="Auscultation and Olfaction";}
            if(LoginControl.Palpation.method4.Equals("Palpation"))
            {type=Convert.ToString(LoginControl.Palpation.ckStr); }
            CaseID=Int32.Parse(Request.Params["CaseNo"]);
            caseno=Convert.ToString(CaseID);
            Label3.Text=caseno;
            dateStr=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
            DateTime.Now.Day,DateTime.Now.Year);
            Label1.Text=String.Format("{0:D}{1:D}/{2:D2}",DateTime.Now.Month,
            DateTime.Now.Day,DateTime.Now.Year);
            if(!IsPostBack)
                bindGrid();
        }
        private void bindGrid()
        {
            objConn7= new SqlConnection(newConn7);
            strSQL7="SELECT SerialNo, SubType, SuggestionOfDisorders FROM CaseHistory WHERE
            MainType="+type+" and CaseNo="+CaseID+" and Date="+dateStr+"";
            Connect();
            //DataAdapter & Adapter setup
            SqlDataAdapter objAdapter=new SqlDataAdapter(strSQL7,objConn7);
```

```

        DataSet objDataSet=new DataSet();
        objAdapter.Fill(objDataSet, "dtSelectedItems1");
        Disconnection();
        SIDataGrid.DataSource=objDataSet.Tables["dtSelectedItems1"];
        //Bind data
        SIDataGrid.DataBind();
    }
    private void Connect()
    {
        if(objConn7==null)
            objConn7= new SqlConnection(newConn7);
        if(objConn7.State==ConnectionState.Closed)
            objConn7.Open();
    }
    private void Disconnection()
    {
        objConn7.Close();
    }
    private void ChangePages(object source, System.Web.UI.WebControls.DataGridPageChangedEventArgs e)
    {
        SIDataGrid.CurrentPageIndex=e.NewPageIndex;
        bindGrid();
    }
    private void DeleteRecord(object source, System.Web.UI.WebControls.DataGridCommandEventArgs e)
    {
        // Retrieve the SerialID to be deleted
        //selcet the right column position, which contain primary key and put its content into the variable
        itemCt=SIDataGrid.Items.Count; //Cout the number of items in each page of SIDataGrid
        //If the page only contain one items, after delte the item, the page will jump to the previous page
        //if the current is zero, keep the page in the zero after delete the last item in page zero
        if(itemCt.Equals(1) && SIDataGrid.CurrentPageIndex.Equals(0))
            SIDataGrid.CurrentPageIndex=0;
        else if(itemCt.Equals(1))
            SIDataGrid.CurrentPageIndex-=1;
        int SerialID = Convert.ToInt32(e.Item.Cells[1].Text);
        SIDataGrid.EditItemIndex = -1;
        deleteSelectedRow(SerialID);
        // Display the remaining items in the DataGrid
        DataSet ds = new DataSet();
        SIDataGrid.DataSource = ds.Tables["dtSelectedItems1"];
        SIDataGrid.DataBind();
        bindGrid();
    }
    private void deleteSelectedRow(int SerialID)
    {
        // Create and load a DataSet with records from ClinicSystem1's CaseHistory table
        objConn7= new SqlConnection(newConn7);
        Connect();
        SqlDataAdapter adapter = new SqlDataAdapter(strSQL7, objConn7);
        DataSet ds = new DataSet();
        adapter.Fill(ds, "dtSelectedItems1");
        Disconnection();
        // Mark the appointment as Deleted in the DataSet
        DataTable tbl = ds.Tables["dtSelectedItems1"];
        tbl.PrimaryKey = new DataColumn[] {tbl.Columns["SerialNo"]};
        DataRow row = tbl.Rows.Find(SerialID);
        row.Delete();
        // Reconnect the DataSet and delete the record from the database
        SqlCommandBuilder cb = new SqlCommandBuilder(adapter);
        objConn7 = new SqlConnection(newConn7);
        Connect();
        adapter.Update(ds,"dtSelectedItems1");
        Disconnection();
    }
    private void backClick(object sender, System.EventArgs e)
    {
        if(LoginControl.Inquiry.method2.Equals("Inquiry"))

```

```
{
    LoginControl.Inquiry.method2="";
    Response.Redirect("Inquiry.aspx");
}
if(LoginControl.Look.method1.Equals("Look"))
{
    LoginControl.Look.method1="";
    Response.Redirect("Look.aspx");
}
if(LoginControl.ListenSmell.method3.Equals("ListenSmell"))
{
    LoginControl.ListenSmell.method3="";
    Response.Redirect("ListenSmell.aspx");
}
if(LoginControl.Palpation.method4.Equals("Palpation"))
{
    LoginControl.Palpation.method4="";
    Response.Redirect("Palpation.aspx");
}
}
}
```

## REFERENCES

- [1] John K., et al.; Beginning ASP .NET Database using C#.; Wrox Press, July, 2002.
- [2] Ramez E. and Shamkant B. Navathe; Fundamentals of Database Systems Third Edition; Addison-Wesley; 2000.
- [3] Harvey M. Deitel, et al.; C# How to Program; Sams; December 14, 2001
- [4] John S. and Jon J.; Visual C#.NET step by step; Microsoft Press; 2003.
- [5] William R. S.; Microsoft IIS 6.0; Microsoft Press; 2003.
- [6] Dusan Petkovic; SQL SERVER 2000 A Beginner's Guide; McGraw-Hill; 2000
- [7] Xuufeb Wu; Traditional Chinese Diagnostics; People's Medical Publishing House; December, 2000
- [8] Tom N. et al.; JAVASCRIPT FOR THE WORLD WIDE WEB 3<sup>rd</sup> Edition; Peachpit Press; 1999.
- [9] <http://alternativehealing.org/Fourteen%20Channels.htm>