NEUROLOGICAL PATIENT INFORMATION MANAGEMENT SYSTEM

NIKOM SUWANKAMOL

PATCHARAPORN MUNCHUPAIBOON

PALAT CHANTAWANNOP

A SPECIAL PROJECT SUMMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE (INTERNATIONAL PROGRAM)

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACEDEMIC YEAR 2011

Thesis Title	Neurological Patient Information Management System		
Students	Mr. Nikom	Suwankamol	Student ID: 51051182
	Ms. Patcharaporn	Munchupaiboon	Student ID: 51051184
	Mr. Palat	Chantawannop	Student ID: 51051205
Degree	Bachelor of Science		
Program	Computer Science (International Program)		
Year	2011		
Thesis Advisor	Dr. Warangkhana Kimpan		

ABSTRACT

Neurological Patient Information Management System has been developed for solving the problems in Neurological Department of Ramathibodi Hospital. Since, this department has found many problems with current system such as, the current interface which still provides hard operations. The system uses less technology and many functions of manual process are outdated. Developing this system will help this department get a new easy system.

In order to provide the easy system, it is one good way in considering the use of web application which provides the connection from different places as well as different platforms. Furthermore, allowing the use of tablets in this new system also represents the progression of developing new technology in order to help doctors operate the information easily.

This system uses Entity-Relationship Diagram (ER-Diagram) to verify relation of database, Data Flow Diagram (DFD) to define the steps to go through and other tools such as, PHP language, Cascading Style Sheets (CSS) to create and develop this system.

ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to my advisor Dr. Warangkhana Kimpan for the continuous support and guidance. We could not have imagined having a better advisor and mentor for our special problem.

Besides my advisor, we would like to thank the rest of our this special problem committees: Assoc. Prof. Dr. Jeeraporn Werapun and Asst. Prof. Dr. Korakot Prachumrak, for their encouragement, insightful comments, and hard questions.

We also thank you doctor Taweesak Chanwitayanuchit from Ramathibodi Hospital for his advice about Neurological Observation.

We thank our friends for suggestion, and for all the fun we have had in all four years.

Last but not least, we would like to thank our family, for supporting us for this special problem.

Nikom Suwankamol

Patcharaporn Munchupaiboon

Palat Chantawannop

April 2012

Content

Content (Continue)

Content (Continue)

Content (Continue)

List of Tables

List of Tables (Continue)

List of Figures

List of Figures (Continue)

List of Figures (Continue)

List of Figures (Continue)

Chapter 1

Introduction

1.1 Importance and Cause of Problem

Information and Communication Technology is very important in daily life. Especially, in the organizations that communicate through each department via computer network such as in a hospital. The centralized database system in the hospital has caused troubles to the users to access the system because it has the complexity and the user cannot access it all the time. This special problem's aim is to develop a new application to solve all those problems. The proposed system will be developed on a web application which is used to record and collect the patients' information in the ease study of the neurological department which does not depend on the centralized database of the hospital. The system focuses on the important functions that are necessary for the users. For example, the function that is used to record the details of the patient symptoms. The web application provides Graphical User Interface (GUI) which is easy to use. In addition, the user can be able to use the web application through the web browser and also through the handheld devices such as mobile phones or tablets which are easy and quickly in retrieving and recording the patient information.

1.2 Objectives

1) To develop the fast and easy web application to search and retrieve the patients' information.

2) To retrieve some information from centralized database which it needs to be included in a new application instead of creating a whole system in order to reduce the time of development.

3) To develop the necessary functions which are needed to be kept in the database.

1.3 Scope of Project

1) To manage the patient's information: add, delete, update, and record the details of the patients' symptoms.

2) To develop the open-source on the platform that can be transferred to other systems from the front-end and back-end with existing data and new data.

3) To store the different type of information in the database such as, texts or pictures to record the details of the patient.

4) To represent some data with a graph.

5) To provide touch and gesture systems in some parts of the patient information management system (Draft version).

1.4 The Benefits of the Project

1) Graphical User Interface can easily be used to enter the patient's information into the system.

2) Necessary data for doctors can be retrieved conveniently and correctly within the shorter time.

3) New technology (Touch and Gesture system) is implemented.

1.5 Implementation Procedures

1) Study Open Source language, tools and software that will be used.

2) Get the requirement from the users.

3) Design the Graphical user interface (GUI) of the system.

4) Develop the system as the requirements

5) Test the software and improve the software qualities.

6) Write the document to describe how to use the software.

7) Write the final report of the special problem.

1.6 The Equipment to be used for the Special Problem.

1) Hardware: PC, Laptop or Notebook, and Tablets.

2) Programming Language and Software: PHP language, JavaScript language, Asynchronous JavaScript and XML (AJAX), jQuery, HTML5, Highcharts, Integrated Development Environment (IDE), phpMyAdmin, Adobe Dreamweaver, Microsoft Windows, Microsoft Visio, Microsoft Project.

Chapter 2

Background

2.1 HTML (Hyper Text Markup Language) and HTML5 [1]

HTML (Hyper Text Markup Language) is the markup language for web pages. HTML is not a programming language; it is a markup language to create a webpage. HTML is written in the form of HTML elements consisting of tags and enclosed in angle brackets (like <html>) within the web page content. HTML tags come in pairs such as <h1> and </h1>. Although some tags known as "empty elements". Empty elements are closed in the start tag" such as <img>. The first tag in a pair is the start tag. The second tag is the end tag. In between these tags web designers can add text, tags, comments, and other types of text-based content. The purpose of a web browser (such as Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the HTML tags but uses the tags to interpret the content of the page.

2.1.1 Editing HTML

HTML can be written and edited by using a simple plain text editor like many different editors like Dreamweaver and Visual Studio.

2.1.2 HTML Elements

An HTML element is composed of everything from the start tag to the end tag.

Table 2.1 HTML elements

Start Tag	Element Content	End Tag
	This is a paragraph	
	This is a link	

1) Example of HTML Elements

```
<html>
<body>
<p>This is a paragraph.</p>
</body>
</html>
```

a) The Element

```
<p>This is a paragraph.</p>
```

- The element defines a paragraph in the HTML document.
- The text between and is displayed as a paragraph.
- The element content is: This is a paragraph.

b) The <body> Element

```
<body>
<p>This is my first paragraph.</p>
</body>
```

- The <body> element defines the body of the HTML document.
- The text between <body> and </body> is the visible page content.

- The element content is a p element.

b) The <html> Element

```
<html>
<body>
        <p>This is my first paragraph.</p>
</body>
</html>
```

- The <html> element defines the whole HTML document.
- The text between <html> and </html> describes the web page.
- The element content is the body element.

2.1.3 Empty HTML Elements

HTML elements with no content are called empty elements. For example,
 is an empty element without a closing tag which uses to define a line break. In XHTML, all elements must be closed. It requires to add a slash inside the start tag, like
which is the proper way of closing empty elements in XHTML (and XML).

2.1.4 Use Lowercase Tags

HTML tags are not case sensitive. For example, <P> means the same as <p>.

2.1.5 HTML Comments

Comments can be inserted into the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed.

1) Example of Comment

```
<!-- This is a comment -->
```

Table 2.2 HTML tags

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines an anchor
 	Defines a single line break
<col />	Defines attribute values for one or more columns in a table
	Defines a section in a document
<form>	Defines an HTML form for user input
<h1> to <h6>	Defines HTML headings
<h1> to <h6>	Defines HTML headings
<img />	Defines an image
<table>	Defines a table
<title>	Defines the title of a document

2.1.6 HTML5 [2]

HTML5 is a language for structuring and presenting content for the World Wide Web, and is a core technology of the Internet originally proposed by Opera Software. It is the fifth revision of the HTML standard (created in 1990 and standardized as HTML4 as of 1997) and as of March 2012 is still under development. Its core aims to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understand by computers and devices (web browsers, parsers, etc.). HTML5 is also a potential candidate for cross-platform mobile applications. Many features of HTML5

have been built with the consideration of being able to run on low-powered devices such as smartphones and tablets.

2.1.7 HTML5 Features

HTML5 introduces a number of new elements and attributes that reflect typical usage on modern websites. Some of them are semantic replacements for common uses of generic block ( ) and inline () elements, for example <nav> (website navigation block), <footer> (usually referring to bottom of web page or to last lines of HTML code), or <audio> and <video> instead of <object>. Some deprecated elements from HTML 4.01 have been dropped, including purely presentational elements such as <font> and , whose effects are achieved using Cascading Style Sheets. There is also a renewed emphasis on the importance of DOM scripting (e.g., JavaScript) in Web behavior.

The HTML5 syntax is no longer based on SGML despite the similarity of its markup. It has, however, been designed to be backward compatible with common parsing of older versions of HTML. It comes with a new introductory line that looks like an SGML document type declaration, <!DOCTYPE html>, which triggers the standards-compliant rendering mode. As of 5 January 2009, HTML5 also includes *Web Forms 2.0*, a previously separate WHATWG specification.

Table 2.3 HTML5 elements

Tag	Description
<article>	Defines an article
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<figcaption>	Defines a caption for a <figure> element
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<section>	Defines a section in a document
<time>	Defines a date/time

Table 2.4 HTML5 new media elements

Tag	Description
<audio>	Defines sound content
<video>	Defines a video or movie
<source>	Defines multiple media resources for <video> and <audio>
<embed>	Defines a container for an external application or interactive content (a plug-in)
<track>	Defines text tracks for <video> and <audio>

Table 2.5 HTML5 the new <canvas> element

Tag	Description
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)

2.1.8 Removed Elements

The following HTML 4.01 elements are removed from HTML5:

- <acronym>
- <applet>
- <basefont>
- <big>
-
- <dir>
- <font>
- <frame>
- <frameset>
- <noframes>
- <strike>
- <tt>
- <u>

2.2 PHP (PHP: Hypertext Preprocessor) [3]

PHP is an Open Source server-side scripting language designed for web development to produce dynamic web pages which can be embedded into HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. PHP is an open source software and free to use.

2.2.1 PHP Code Syntax

PHP code is executed on the server and HTML result is sent to the browser. A PHP scripting block always starts with <?php and ends with ?> or shorthand enabled start a scripting block with <? and end with ?>. Each code line in PHP must end with a semicolon(;). The semicolon is a separator and is used to distinguish one set of instructions from another. There are two statements to output text with PHP: echo and print. The file must have a .php extension.

2.2.2 PHP Statement

In the example below, PHP script sends the text "Hello World" to the browser and used the echo statement to output the text "Hello World".

1) Example of PHP Statement

```
<html>
        <body>
                <?php
                echo "Hello World";
                ?>
        </body>
</html>
```

2.2.3 Variables

Variables are used to store values such as text strings, numbers or arrays. The variables in PHP start with a $ sign symbol. The correct way of declaring a variable in PHP is $var_name = value;

1) Example of Declaring Variables

```
<?php
$txt="Hello World!";
$x=16;
?>
```

In the example, it creates a variable containing a string and a variable containing a number.

2) Naming Rules for Variables

- A variable name must start with a letter or an underscore "_"
- A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z, 0-9, and _)
- A variable name should not contain spaces. If a variable name is more than one word, it should be separated with an underscore such as $my_string , or with capitalization such as $myString.

2.2.4 Operators

Table 2.6 Arithmetic operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4	20

Operator	Description	Example	Result
		x*5	
/	Division	15/5	3
		5/2	2.5
%	Modulus (Division Remainder)	5%2	1
		10%8	2
		10%2	0

Table 2.7 Assignment operators

Operator	Example	Is the same as
=	x=y	x=y
+=	x+= y	x=x+y
-=	x-= y	x=x-y
*=	x*= y	x=x*y
/=	x/= y	x=x/y
.=	x.= y	x=x.y
%=	x%= y	x=x%y

Table 2.8 Comparison operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
<>	is not equal	5<>8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true

Operator	Description	Example
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

2.2.5 PHP Built-In Functions

- echo(strings) such as echo("Hello, World")
- print(strings) such as print("Hello, World")
- date(format,timestamp) such as date("Y-m-d H:i:s")
- substr(string,start,length) such as substr("Hello world!",6,5)
- strlen(string) such as strlen("Hello world!")
- strcmp(string1,string2) such as strcmp("Hello world!","Hello world!")
- trim(string,charlist) such as trim(("Hello world! ")
- explode(separator,string,limit) such as explode (" ", "Hello world. It's a beautiful day.")
- strrpos(string,find,start) such as strrpos("Hello world!","wo")

2.2.6 Comments in PHP

In PHP, it uses double slash (//) to make a single-line comment or /* and */ to make a large comment block.

1) Example of Comments in PHP

```
<?php
$a = 5;// set $a to 5
$b = 16; // set $b to 16
$b += $a; /* add $a to $b */
echo $b, " ";
?>
```

2.3 JavaScript [4]

JavaScript is the scripting language within the web browser that usually embedded directly into HTML pages. It was designed to add interactivity to HTML pages. JavaScript is used in the form of client-side and implemented as part of a Web browser in order to provide enhanced user interfaces and dynamic websites. JavaScript can use in applications outside webpages. For example in PDF documents, site-specific browsers, and desktop widgets.

JavaScript uses syntax from C language. It copies many names and naming conventions from Java, but the two languages are unrelated and have very different semantics.

2.3.1 The Difference between JavaScript and Java

The two languages are unrelated. The main difference is that Java can stand on its own while JavaScript must be placed inside an HTML document to function. Java is developed under the Sun Microsystems brand. It is an object-oriented programming language. It can be used to create standalone applications and a special type of mini application, called an applet. Applets are downloaded as separate files to your browser alongside an HTML document. JavaScript is developed by Netscape. It is a scripting language that does not create applets or standalone applications. JavaScript embedded inside HTML documents to provide the dynamics and interactive features of the webpage.

2.3.2 Use in Webpages

The primary use of JavaScript is to write functions that are embedded in or included from HTML pages and that interact with the Document Object Model (DOM) of the page. Some examples of this usage are:

- Opening or popping up a new window with programmatic control over the size, position, and attributes of the new window (e.g. whether the menus, toolbars, etc., are visible).
- Validating input values of a web form to make sure that they are

acceptable before being submitted to the server.

- Changing images as the mouse cursor moves over them. This effect is often used to draw the user's attention to important links displayed as graphical elements.

Because JavaScript code can run in a user's browser, the browser can respond to user actions quickly, making an application more responsive. Furthermore, JavaScript code can detect user actions which HTML alone cannot, such as individual keystrokes. The most popular applications such as Gmail uses JavaScript to write user-interface logic and JavaScript requests for information (such as the content of an e- mail message) to the server.

JavaScript is the only language that the most popular browsers share support for. It has become a target language for many frameworks in other languages, even though JavaScript was never intended to be such a language.

2.3.3 Related Languages and Features

jQuery and Prototype are popular JavaScript libraries designed to simplify DOM -oriented client-side HTML scripting.

2.3.4 Insert JavaScript into a HTML Page

In order to insert a JavaScript into an HTML page, the <script> tag will be used. Inside the <script> tag use the type attribute to define the scripting language. The <script> and </script> tell where the JavaScript starts and ends. JavaScript is placed between tags starting with <script type = text/javascript> and ending with</script>.

1) Example of Syntax in JavaScript

```
<html>
<body>
<script type="text/javascript">
  ... //JavaScript
        </script>
</body>
</html>
```

2.3.5 Write Output to a Page

The JavaScript command used for writing output to a page is document.write. The document.write command takes the argument as the text required for producing output.

1) Example of JavaScript Command inside an HTML Page

```
<html>
<body>
<script type="text/javascript">document.write("This is  JavaScript")
</script>
</body>
</html>
```

2) Output of the Script above Produced in a HTML Page

```
This is JavaScript
```

2.3.6 Different Places where JavaScript can be placed in HTML

JavaScript can be placed in various locations in HTML such as:

- JavaScript in HEAD section
- JavaScript in BODY section
- JavaScript in both HEAD and BODY section
- JavaScript in External File

The placing of JavaScript in the above location differs in the timing of their execution. JavaScript placed in the HEAD section of HTML will be executed when called whereas, JavaScript placed in the BODY section of HTML will be executed only when the page is loaded.

1) The General Structure for Placing a JavaScript in the HEAD Section

```
<html>
<head>
<script type="text/javascript">
..... //JavaScript written in HEAD Section

</script>
</head>
<body>
</body>
</html>
```

2) The General Structure for placing a JavaScript in the BODY Section

```
<html>
      <head>
      </head>
<body>
<script type="text/javascript">
.....

..... //JavaScript written in BODY Section

</script>
</body>
```

If a programmer wants to execute JavaScript when called, or when an event is triggered, then JavaScript is placed in the HEAD section. JavaScript is placed in the HEAD section because scripts get loaded first. When a programmer wants to execute JavaScript when the page loads then JavaScript should be placed in the BODY section. In addition to these, JavaScripts can be inserted in both HEAD and BODY sections of an HTML page.

3) The General Structure of JavaScript for Placement in both HEAD and BODY Sections

```
<html>
<head>
<script type="text/javascript">
//JavaScript written in HEAD Section
 document.write("JavaScript placed in HEAD Section")
</script>
</head>
<body>
<script type="text/javascript">
 //JavaScript written in BODY Section
  document.write("JavaScript placed in BODY Section")
</script>
</body>
</html>
```

4) The Output of the Script above Produced in a HTML Page

```
JavaScript placed in HEAD Section
JavaScript placed in BODY Section
```

2.3.7 JavaScript in External File

There may be scenarios in which the same functionality of script needs to be executed in several places in program. For handling this scenario instead of writing the same JavaScript in several places which would cause poor optimization of code, one can place the JavaScript in external file.

1) Place the JavaScript in External File

This process is very simple. The code or the JavaScript which needs to be executed in several places in the program is written separately in a file and is saved with extension as .js for the file.

2) Execute JavaScript placed in External File

This is done by using the attribute named as src for the <script> tag.

a) The General Syntax to refer JavaScript in External File

```
<html>
 <head>
   <script src="filename">
   </script>
</head>
<body>
</html>
```

It is important to note that external JavaScript file cannot contain HTML tags.

b) Example of placing JavaScript in External File

Suppose if the JavaScript placed in external file is name as helloJS.js, then it is executed by placing it as:

```
<html>
<head>
<script src="helloJS.js"></script>
</head>
<body>
        //////// JavaScript
</body>
</html>
```

2.4 CSS (Cascading Style Sheets) [5]

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. CSS language works with HTML to define the contents of a website with ease and flexibility in a straightforward process. It is designed to enable adjusting layouts and style sheets such as layout, colors, and fonts. CSS saves programmer and designer's time in adjusting changes to the development of a website by placing the code directly into the HTML document head, and/or in a separate sheet. Single CSS file can control the appearance of multiple HTML documents. In order to makes a change to all documents. There is no need to make the change in every document, just make it in the CSS file and it will be reflected on all documents that are linked to it. CSS has the additional benefit of reducing the file size of HTML documents. CSS provides the programmer with secure style sheets that can be linked to a single document to ensure that it produce the correct presentation on different media including browsers, handheld devices, projectors, and printers.

2.4.1 CSS Code Syntax

CSS has a simple syntax and uses English keywords to specify the names of style properties. A style sheet consists of a list of rules.

2.4.2 CSS Rules

A rule or rule set tells the browser how to render an element. The rule set consists of the following:

- **The selector**: represents the HTML element to be affected by the rule.
- **The declaration block**: represents the effect to be applied to the element(s), and it contains one or more property value pairs.

A declaration-block consists of a list of declarations in braces. Each declaration itself consists of a property, a colon (:), a value. If there are multiple declarations in a block, a semi-colon (;) must be inserted to separate each declaration. Selectors are used to declare which of the markup elements a style applies to. Selectors may apply to all elements of a specific type, or only those elements that match a certain attribute.

2.4.3 Example of Selector

```
p
{
text-align: right;
color: red;
}
```

The selector in the example is the HTML element < p >. The declaration block has to start with an opening curly brace "{", and ends with a closing curly brace "}", every property/value pair has to be ended with a semi-colon";" The property and the value are separated with a colon ":", spaces and line may be added for better readability, it does not affect the CSS validity.

2.4.4 Combining Selectors

When many selectors have the same declarations, they can be grouped. This will reduce the CSS file size, and make it easier to update the style.

1) Example of Combining Selectors

In the example, all the elements h1, h2, and h3 will have center-aligned text, and red color, and only h2 font style will be italic.

```
h1, h2, h3
{
text-align: center;
color: red;
}
h2
{
font-style: italic;
}
```

2.4.5 The Class Selector

Class selector uses to define the same HTML element with different style.

1) Example of the Class Selector in CSS

```
header {
font-weight: bold;
font-variant: small-caps;
text-align: center;
}
normal {
text-align: left;
}
```

2) Example in HTML Document

```
<table>
        <tr>
                <td class="header"> ID </td>
                <tdclass="header"> Department </td>
        </tr>
        <tr>
                <td> 1 </td>
                <td> Web Design </td>
        </tr>
</table>
```

2.4.6 The Id Selector

The id selector uses to define the same style for different HTML elements.

1) Example in CSS

```
#red
{
color: red;
}
```

2) Example in HTML Document

In this way only the <p> element with id of red <p id="red"> will be styled using this rule.

```
<h1 id="red">This is a red header </h1>
<p id="red"> This is a red paragraph </p>
```

2.4.7 Comments

CSS comments can be placed anywhere in the CSS, it will be ignored by the browser. The comment has to start with "/*" and ends with "*/".

1) Example of Comments

```
/* this is a comment */
p
{
text-align: right;  /* this is another comment */
color: red
}
```

2.4.8 Sources

There are different types for specifying styles to be used with HTML documents, that can be specified in an external CSS file, inside the < head > element of the HTML document, inline to be specific to a single HTML element and there is the browser default style. These styles will be cascaded in a single HTML documents at the following priority:

- Inline style (the highest priority)
- Internal < head > style.
- External CSS file.
- Browser default(the lowest priority)

1) External Style Sheets

The external style sheet can be applied to many HTML documents, a .css file has to be created, and link to it from the HTML document using < link > tag.

a) Example of linking CSS File to the HTML Document

```
<head>
<link rel="stylesheet" type="text/css" href="cssfile.css" />
</head>
```

2) Internal Styles

The internal style will be applied only to the document that declares it.

a) Example of Internal Styles

```
<head>
<style type="text/css">
body {
background-color: #00CCFF;
} h1 {
color: #0000FF;
} p {
font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
```

3) Inline Styles

The inline style is only applied to the HTML element that declares it.

a) Example of Inline Styles

```
<p style="color: red;" >This is a red paragraph.< /p >
```

2.4.9 Some Examples of CSS

Figure 2.1 CSS image map

Figure 2.2 CSS shadow

Figure 2.3 Create a star rater

Name	Thread pitch (mm)	Major thread diameter	Nominal diameter (mm)	Head shape	Price per 50 screws	Available at factory outlet?	Number in stock	Flat or Phillips head?
M4	0.7	4g	4	Pan	$10.06	Yes	276	Flat
M5	0.8	4g	5	Round	$13.59	Yes	183	Both
M6	1	5g	6	Button	$10.42	Yes	1043	Flat
M8	1.25	5g	8	Pan	$11.93	No	298	Phillips
M10	1.5	6g	10	Round	$16.74	Yes	488	Phillips
M12	1.75	7g	12	Pan	$18.26	No	998	Flat
M14	2	7g	14	Round	$21.19	No	235	Phillips
M16	2	8g	16	Button	$23.57	Yes	292	Both
M18	2.1	8g	18	Button	$25.87	No	664	Both
M20	2.4	8g	20	Pan	$29.09	Yes	486	Both
M24	2.55	9g	24	Round	$33.01	Yes	982	Phillips
M28	2.7	10g	28	Button	$35.66	No	1067	Phillips
M36	3.2	12g	36	Pan	$41.32	No	434	Both
M50	4.5	15g	50	Pan	$44.72	No	740	Flat

Figure 2.4 Zebra table (Zebra striping) with JavaScript and CSS

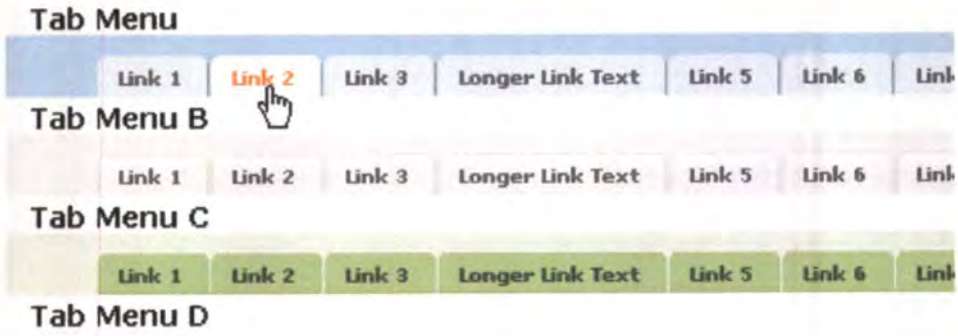


Figure 2.5 CSS tabs

2.5 jQuery and jQuery Mobile Framework [6]

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. jQuery is free and open source software. jQuery library contains a lot of features such as HTML element selections, HTML element manipulation, CSS manipulation, HTML event functions, JavaScript Effects and animations, HTML DOM traversal and modification, AJAX and Utilities. jQuery's syntax is designed to make it easier to navigate a document. For example, select DOM elements, create animations, handle events, and develop Ajax applications. The jQuery framework allows the creation of powerful and dynamic web pages and web applications.

2.5.1 Features

jQuery contains the following features:

- DOM element selections using the cross-browser open source selector
- DOM traversal and modification (including support for CSS 1-3)
- Even
- CSS manipulation
- Effects and animations
- Ajax
- Extensibility through plug-ins

- Utilities - such as browser version and the each function.
- Cross-browser support

2.5.2 Including the Library

The jQuery library is a single JavaScript file, containing all of its common DOM, event, effects, and Ajax functions. It can be included within a web page by linking to a local copy, or to one of the many copies available from public servers.

1) **Example of jQuery File**

```
<script type="text/javascript" src="jquery.js"></script>
        $(document).ready(function() {
        // jquery goes here
});
```

2.5.3 jQuery Syntax

The jQuery syntax is made for selecting HTML elements and performing some action on the element(s). The basic syntax is $(selector).action()

- A dollar sign to define jQuery
- A (selector) to "query (or find)" HTML elements
- A jQuery action() to be performed on the element(s)

1) **Examples of $(selector).action()**

- $(this).hide() - hides current element
- $("p").hide() - hides all paragraphs
- $("p.test").hide() - hides all paragraphs with class="test"
- $("#test").hide() - hides the element with id="test"

2.5.4 The Document Ready Function

Normally, all jQuerymethods are inside a document.ready() function in order to prevent any jQuery code from running before the document is finished loading (is ready).

2.5.5 jQuery Selectors

jQuery selectors allow to select HTML elements (or groups of elements) by element name, attribute name or by content.

2.5.6 jQuery Element Selectors

jQuery uses CSS selectors to select HTML elements.

1) Example of Element Selectors

- $("p") selects all <p> elements.
- $("p.intro") selects all <p> elements with class="intro".
- $("p#demo") selects all <p> elements with id="demo".

2.5.7 jQuery CSS Selectors

jQuery CSS selectors can be used to change CSS properties for HTML elements.

Table 2.9 Examples of CSS selectors

Syntax	Description
$(this)	Current HTML element
$("p")	All <p> elements
$("p.intro")	All <p> elements with class="intro"
$("p#intro")	All <p> elements with id="intro"
$("p#intro:first")	The first <p> element with id="intro"
$(".intro")	All elements with class="intro"
$("#intro")	The first element with id="intro"
$("ulli:first")	The first element of each
$("[href$='.jpg']")	All elements with an href attribute that ends with ".jpg"
$("div#intro .head")	All elements with class="head" inside a element with id="intro"

2.5.8 jQuery Events

Table 2.10 Some examples of event methods in jQuery

Event Method	Description
$(document).ready(function)	Binds a function to the ready event of a document (when the document is finished loading)
$(*selector*).click(function)	Triggers, or binds a function to the click event of selected elements
$(*selector*).dblclick(function)	Triggers, or binds a function to the double click event of selected elements
$(*selector*).focus(function)	Triggers, or binds a function to the focus event of selected elements

Event Method	Description
$(*selector*).mouseover(function)	Triggers, or binds a function to the mouse over event of selected elements

2.5.9 Some Examples of Websites using jQuery

Figure 2.6 Navigation menu

Figure 2.7 Tab menu style 1

Figure 2.8 Tab menu style 2

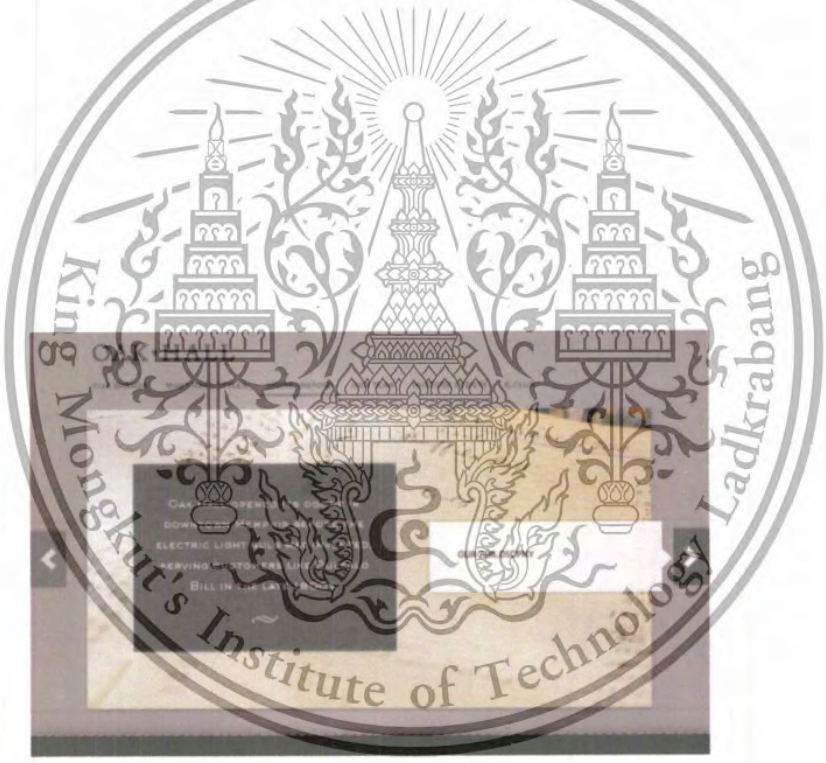


Figure 2.9 Slide show style 1

Figure 2.10 Slide show style 2

Figure 2.11 Form validation style 1

Figure 2.12 Form validation style 2

2.5.10 jQuery Mobile Framework [7]

jQuery Mobile is a touch-optimized web framework (additionally known as a JavaScript library or a mobile framework) currently being developed by the jQuery project team. The development focuses on creating a framework compatible with a wide variety of Smartphone and tablet computers made necessary by the growing but heterogeneous tablet and smartphone market.

1) Basic Example

The first step is to link to the jqm libraries and stylesheet (they can also be downloaded and hosted locally but it is recommended to link to the files hosted on the jQuery CDN). In the head, a meta viewport tag sets the screen width to the pixel width of the device and references to jQuery, jQuery Mobile and the mobile theme stylesheet from the CDN add all the styles and scripts. jQuery Mobile 1.1 works with both 1.6.4 and 1.7.1 versions of jQuery core.

In the body, a div with a data-role of page is the wrapper used to delineate a page, and the header bar (data-role="header") and content region (data-role="content") are added inside to create a basic page (these are both optional). These data- attributes are HTML5 attributes are used throughout jQuery Mobile to transform basic markup into an enhanced and styled widget.

```
<!DOCTYPE html>
<html>
	<head>
	<title>My Page</title>
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0-rc.1/jquery.mobile-1.1.0-rc.1.min.css" />
	<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
	<script src="http://code.jquery.com/mobile/1.1.0-rc.1/jquery.mobile-1.1.0-rc.1.min.js"></script>
</head>
<body>
<div data-role="page">
	<div data-role="header">
		<h1>My Title</h1>
	</div><!-- /header -->
	<div data-role="content">
		<p>Hello world</p>
	</div><!-- /content -->
</div><!-- /page -->
</body>
</html>
```

2) Result of Example

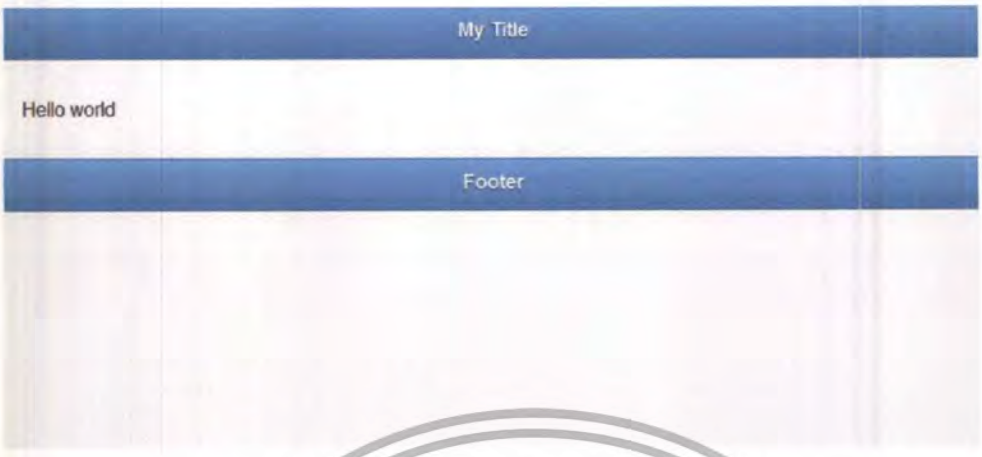


Figure 2.13 Example of jQuery mobile interface

2.6 AJAX (Asynchronous JavaScript and XML) [8]

Asynchronous JavaScript and XML (AJAX) is a technique for creating fast and dynamic web pages. It allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. AJAX is a group of interrelated web development methods used on the client-side to create interactive web applications. Ajax is not one technology, but a group of technologies. Ajax uses a combination of HTML and CSS to mark up and style information.

AJAX is a programming language that has gained high popularity within a short span of time. AJAX became popular in the 2005 by Google. Most of the people think that AJAX is a new programming language but it is not. However, it has introduced a new way of using existing standards. One of the main advantages of using AJAX is that it helps in creating faster, better, dynamic and interactive web applications. Another advantage of using AJAX is that it enables the JavaScript to communicate directly with the server without requiring the user to refresh the page. That is the reason why AJAX is also called as a browser technology which is independent of web server software.

2.6.1 Technologies

AJAX is the combination of several technologies used to implement a web application that communicates with a server in the background without interfering with the current state of the page.

- HTML (or XHTML) and CSS for presentation
- The Document Object Model (DOM) for dynamic display of and interaction with data
- XML for the interchange of data, and XSLT for its manipulation
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

2.6.2 Web Standard

AJAX is basically based on the following web standards:

- XML
- HTML
- JavaScript
- CSS

The main reason why AJAX is supported by all web browsers today is the fact that most of the browsers today support the web standards. AJAX applications are browser and platform independent.

2.6.3 How Does AJAX Works?

AJAX is used for allowing the client side of an application to communicate directly with the server side of the application. Before AJAX had introduced, there was no way for the client side of a web application to communicate directly with the server. Instead,

you would have to use page loads. With AJAX, the JavaScript communicates directly with the server, through the JavaScript XMLHttpRequest object. With an HTTP request, a web page can make a request to, and get a response from a web server, without reloading the page. The user will stay on the same page, and not notice that scripts request pages, or send data to a server in the background.

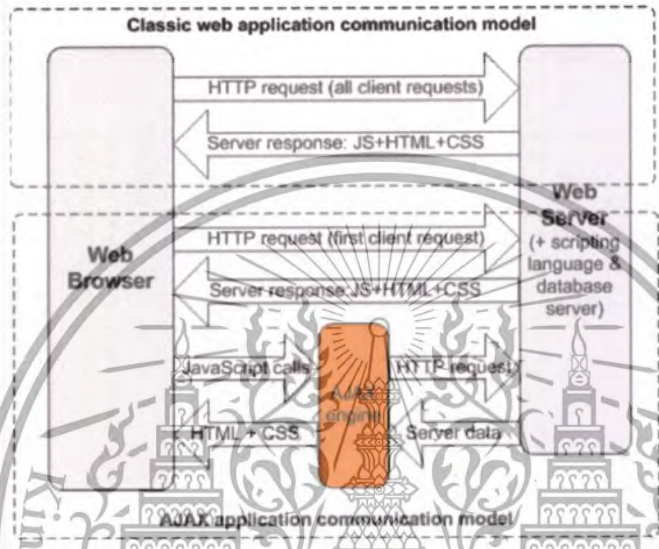


Figure 2.14 Classic and AJAX communication models

1. Some action triggers the event. For example, the user clicking a button.
2. The AJAX call and sends a request to a server-side script, using XML.
3. The server-side script (PHP, ASP, or etc.) takes the input from JavaScript, can access the database if it needs to, and processes the data.
4. Using XML again, the script sends the data back to the original client-side page that made the request.
5. A second JavaScript function, called a callback function, catches the data, and updates the web page.

2.6.4 CreateXMLhttpRequest

Different browsers use different methods to create the XMLHttpRequest object. Internet Explorer uses an ActiveXObject, while other browsers use the built-in JavaScript object called XMLHttpRequest.

```
functionloadXMLDoc(){
        varxmlhttp;
        if (window.XMLHttpRequest {// code for IE7+, Firefox, Chrome,
Opera, Safari
xmlhttp=new XMLHttpRequest();
         }else{// code for IE6, IE5
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
                xmlhttp.onreadystatechange=function()
 {
        if (xmlhttp.readyState==4 &&xmlhttp.status==200){
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
 }
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
}
```

2.6.5 Send a Request To a Server

To send a request to a server, uses the open() and send() methods of the XMLHttpRequest object:

1) Example of Send a Request to a Server

```
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
```

Table 2.11 Methods of the XMLHttpRequest object

Method	Description
open (*method,url,async*)	Specifies the type of request, the URL, and if the request should be handled asynchronously or not. *method*: the type of request: GET or POST *url*: the location of the file on the server *async*: true (asynchronous) or false (synchronous)
send(*string*)	Sends the request off to the server. *string*: Only used for POST requests

2.6.6 GET or POST

GET is simpler and faster than POST, and can be used in most cases. However, always use POST requests when:

- A cached file is not an option (update a file or database on the server)
- Sending a large amount of data to the server (POST has no size limitations)
- Sending user input (which can contain unknown characters), POST is more robust and secure than GET

1) Example of GET Request

```
xmlhttp.open("GET","example_get.php",true);
xmlhttp.send();
```

2) Example of POST Request

```
xmlhttp.open("POST","example_post.php",true);
xmlhttp.send();
```

Chapter 3

Design and Implementation Overview

The design and implementation use the knowledge background from chapter 2 to design and implement the system. This chapter will cover the overview and explain in detail of the several parts in the system.

3.1 System Overview

Neurological Patient Information Management System is used by the doctor to record the patient's symptoms. This system provides an easy way to input the data by representing many pictures as the symptoms. The user can perform actions to the system by just clicking to the button then the system will record the data into a database immediately.

3.2 Data Flow Diagram

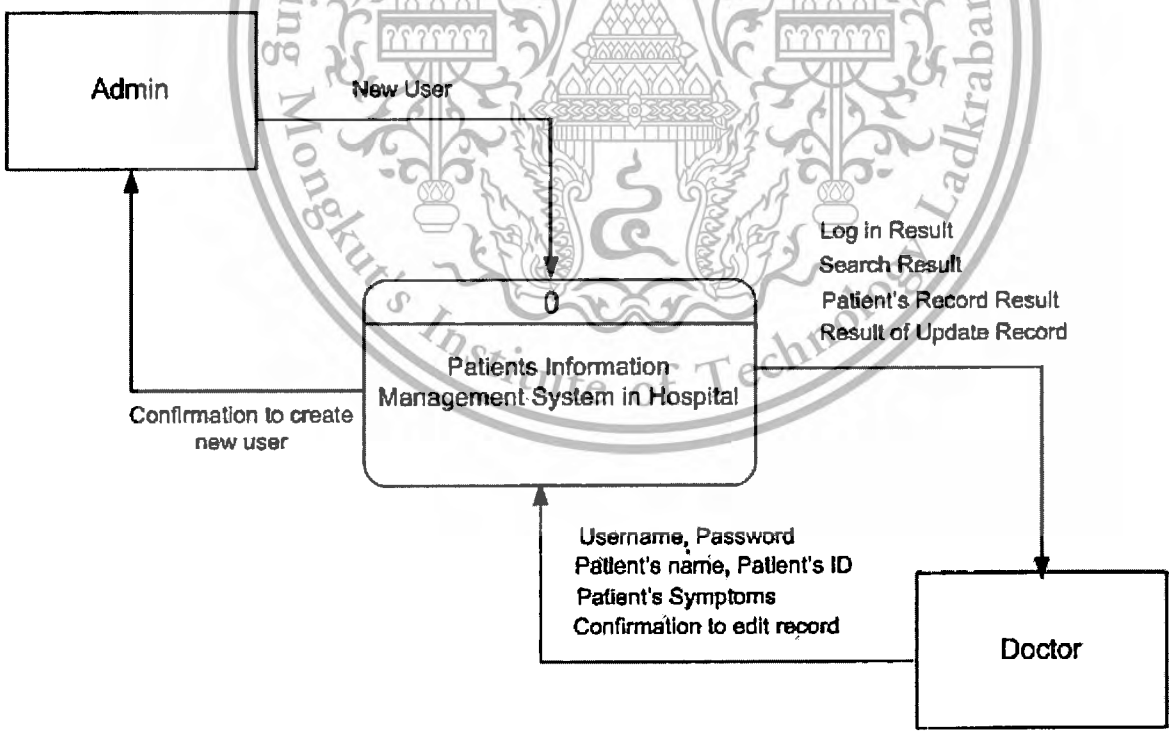


Figure 3.1 Shows context diagram in the overview of the system

The system is divided into two parts, administrator and user. In the administrator section, the administrator is able to add new user into the system. In the doctor (user) section, the doctor can insert, update, delete and also search the patient's symptom record.

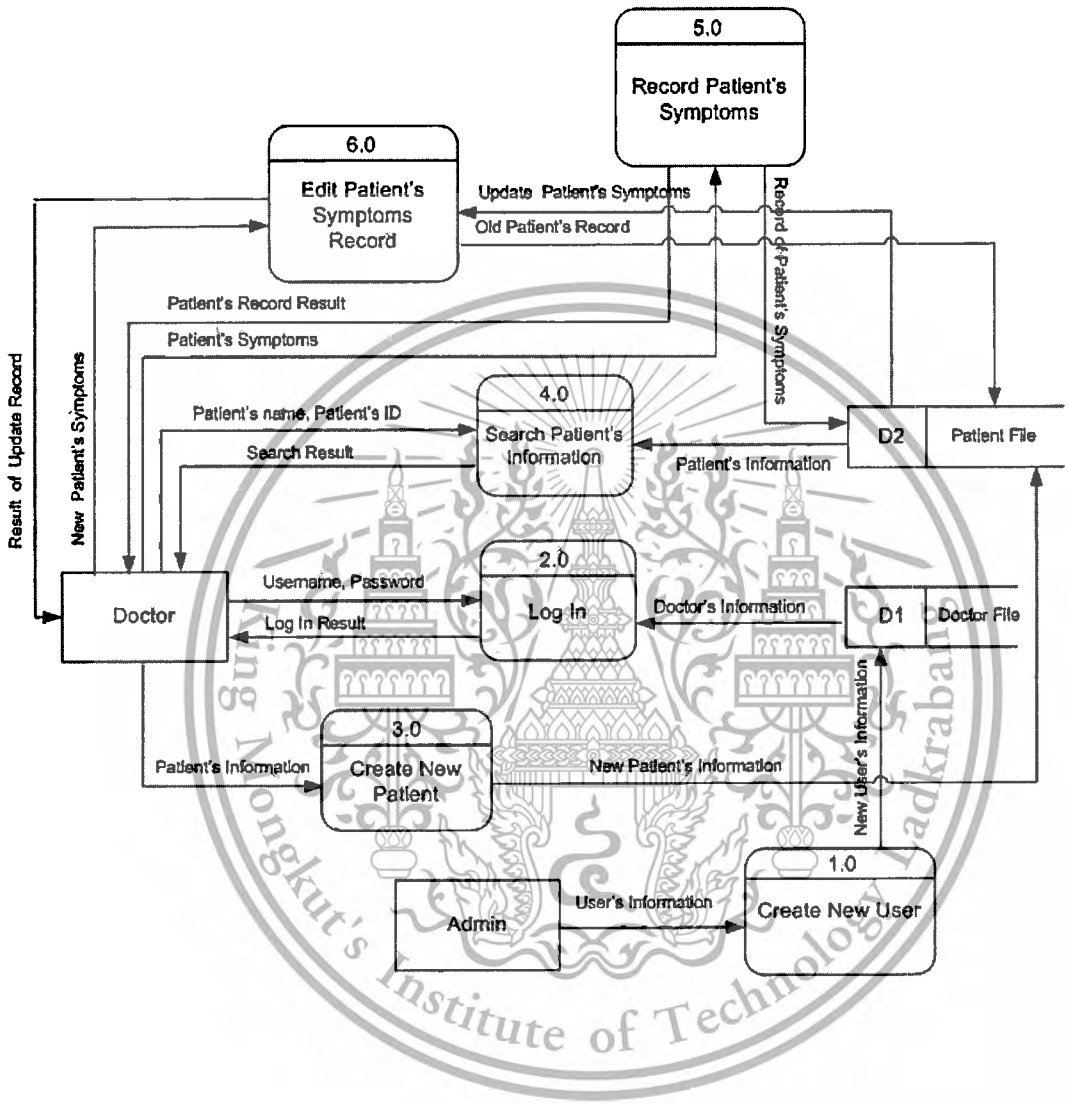


Figure 3.2 Level-0 diagram shows all the processes in the system

Figure 3.2 shows the data flow diagram of the system and the various processes and the relationship among them. The first part is the management of database in the system. The administrator must add the new user. The second part is the doctor who directly uses and interacts directly with the system. The doctor needs to login to the system before recording the patient's information. The doctor is able to inserts, updates, delete or search the patient's

information. When the doctor inserts or updates the patient's information, those records will be kept in the database.

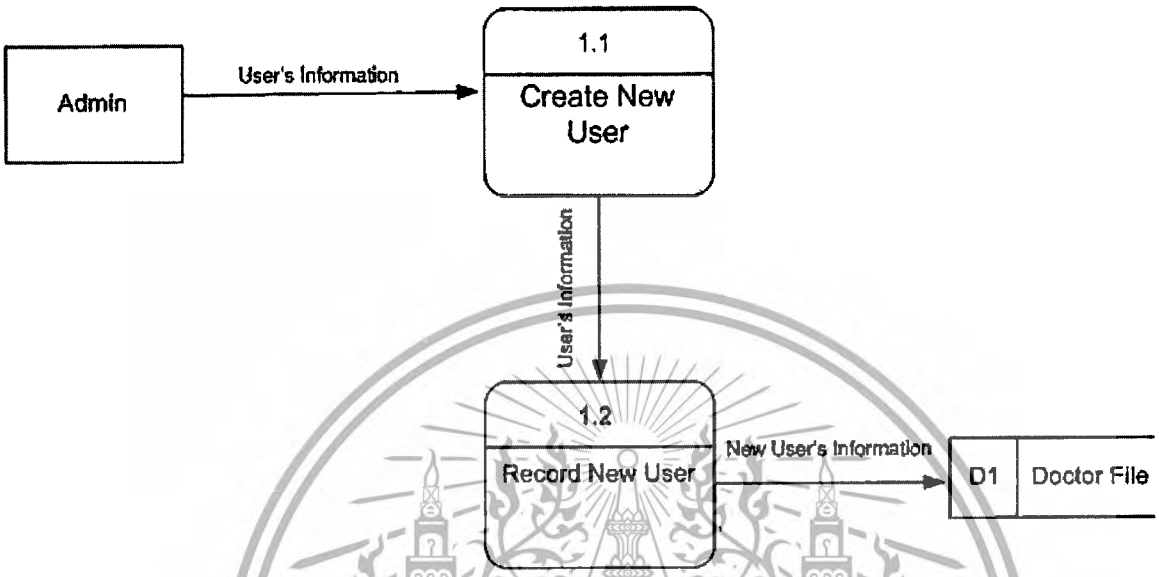


Figure 3.3 Diagram 1 shows process to add new user

Figure 3.3 shows diagram of creating new doctor. The administrator has to create the doctor's information, username and password. Then, the system will record the doctor's information into the database.

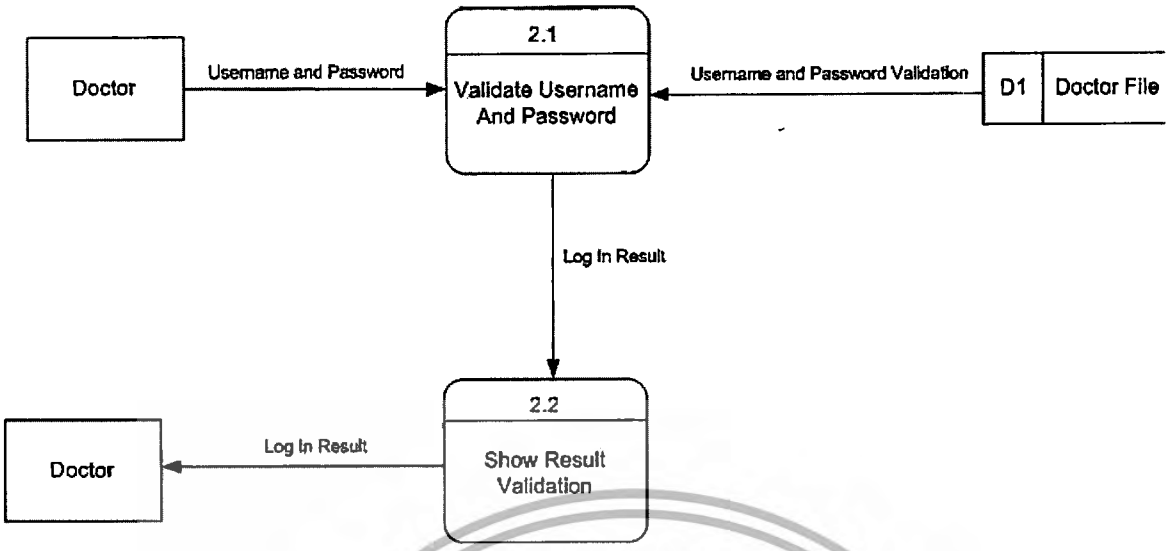


Figure 3.4 Diagram 2 shows process to log in to the system

Figure 3.4 shows diagram of the log in process of the system. The doctor logs into the system by using the username and password. The system will validate the user's account with username and password in the database. Then, the system shows the result of validation to the doctor.

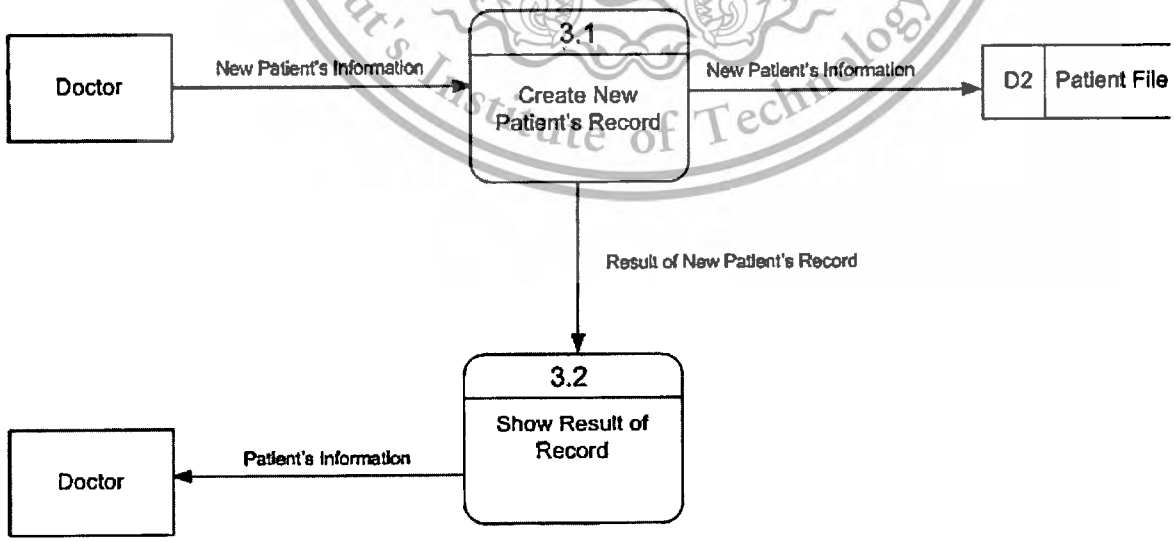


Figure 3.5 Diagram 3 shows process to create the new patient's record

Figure 3.5 shows diagram of creating the new patient's record. After the doctor logs into the system, he or she can create a new patient record by enter the new patient's information. The system receives the new patient information and stores into the patient file. Finally, the system shows the result of new patient's information to the user.

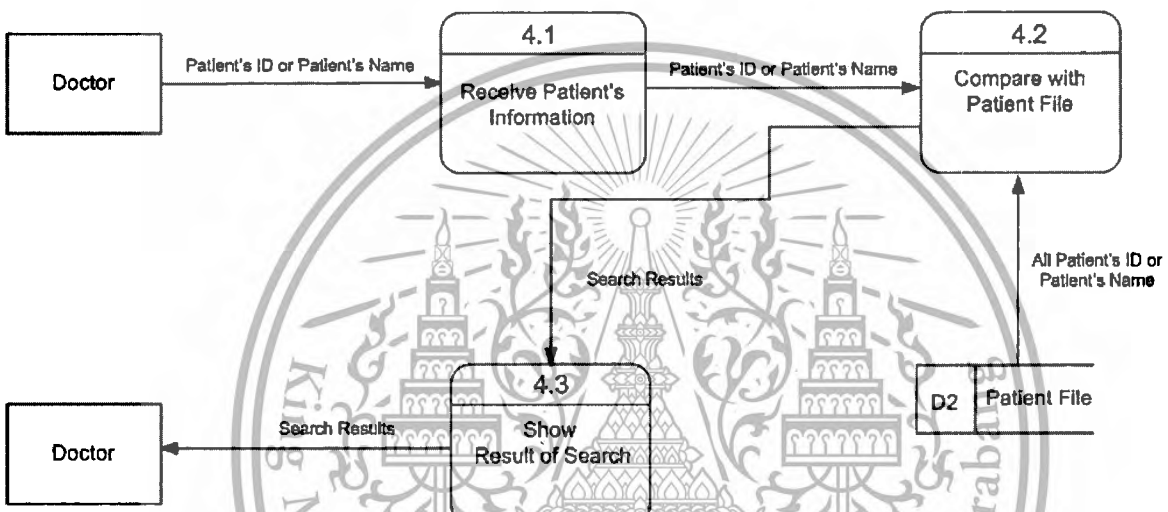


Figure 3.6 Diagram 4 shows process to search the patient's information

Figure 3.6 shows diagram of searching the patient's information. The doctor can search the patient's information by the patient's ID or the patient's name. The system receives the patient's information and compares with the patient file in the database. Then, the system will show the search results to the doctor.

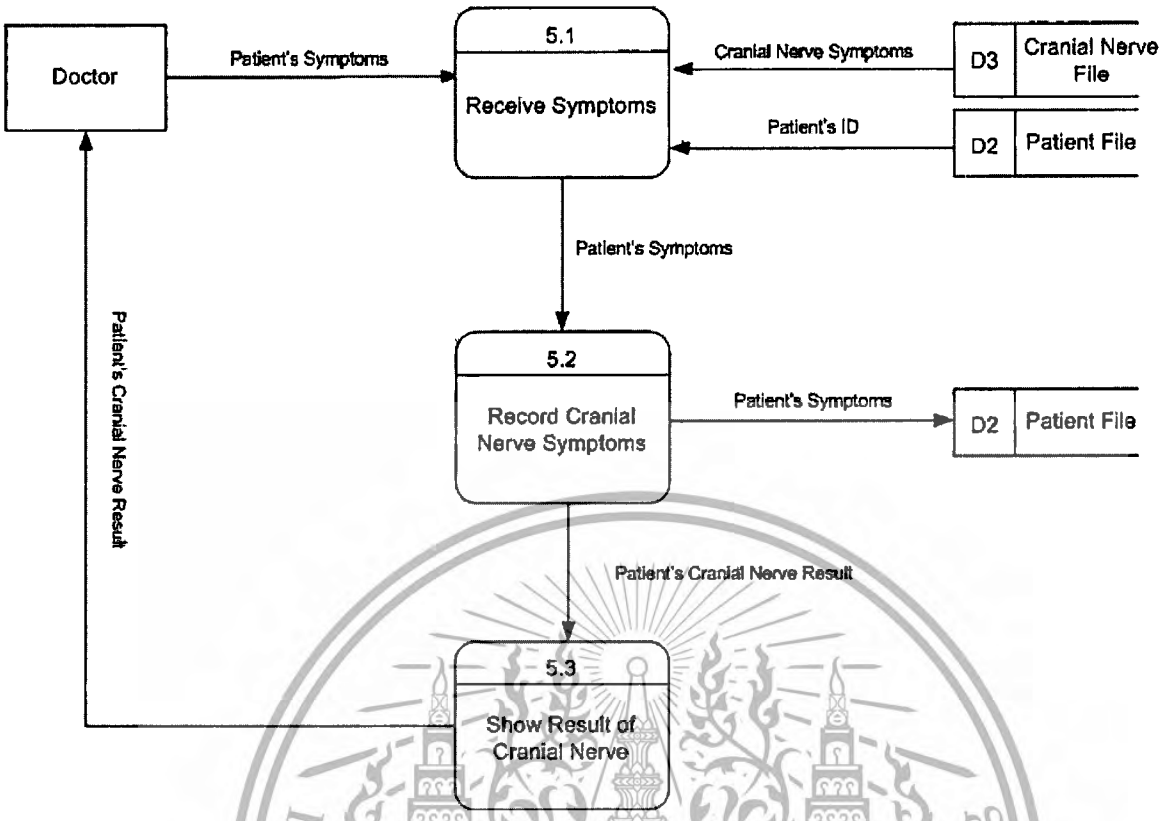


Figure 3.7 Diagram 5 shows process to record cranial nerve

Figure 3.7 shows diagram of recording the patient's symptoms in the cranial nerve section. After selecting the patient to record the patient's information, the doctor selects the patient's symptom which is retrieved from the cranial nerve file. Then, the system will receive the patient's symptom and record into the patient file and shows its result to the doctor.

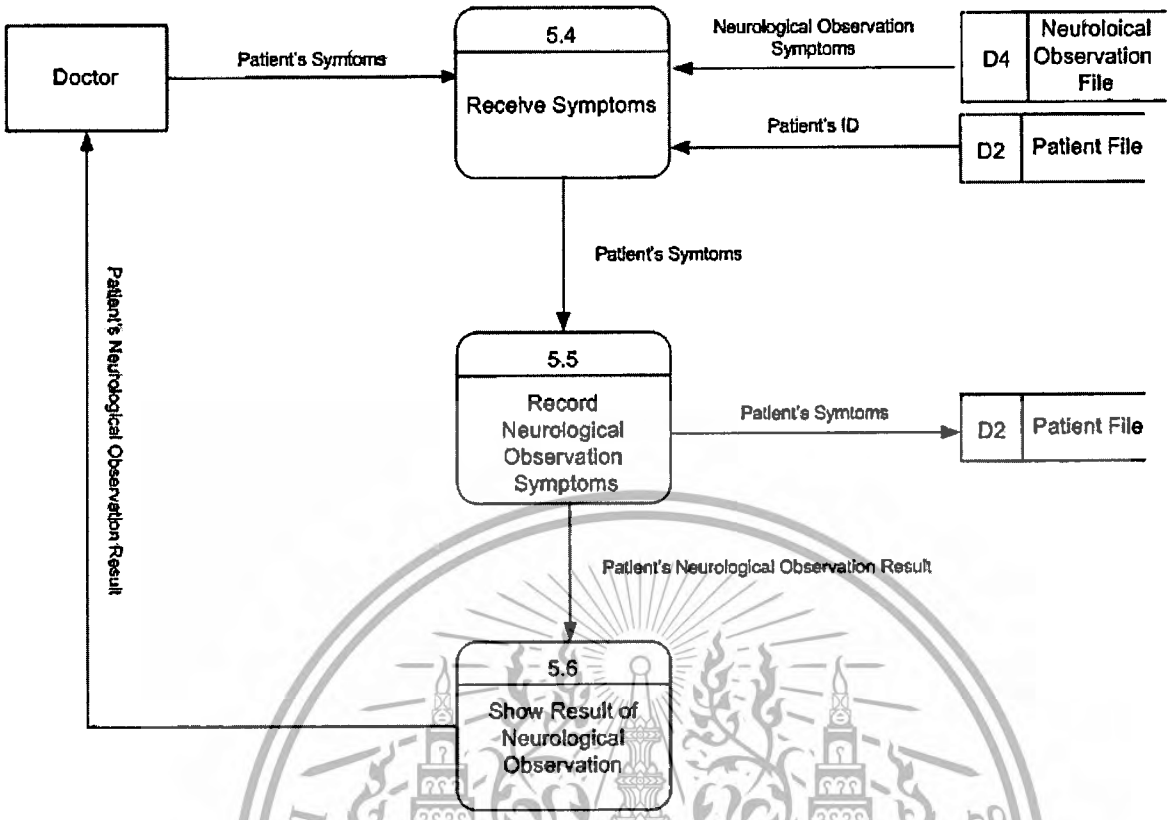


Figure 3.8 Diagram 5 shows process to record neurological observation

Figure 3.8 shows diagram of recording the patient's symptoms in the neurological observation section. After selecting the patient to record the patient's information, the doctor selects the patient's symptom which is retrieved from the neurological observation file. Then, the system will receive the patient's symptom and record into the patient file. Finally, it shows the result to the doctor.

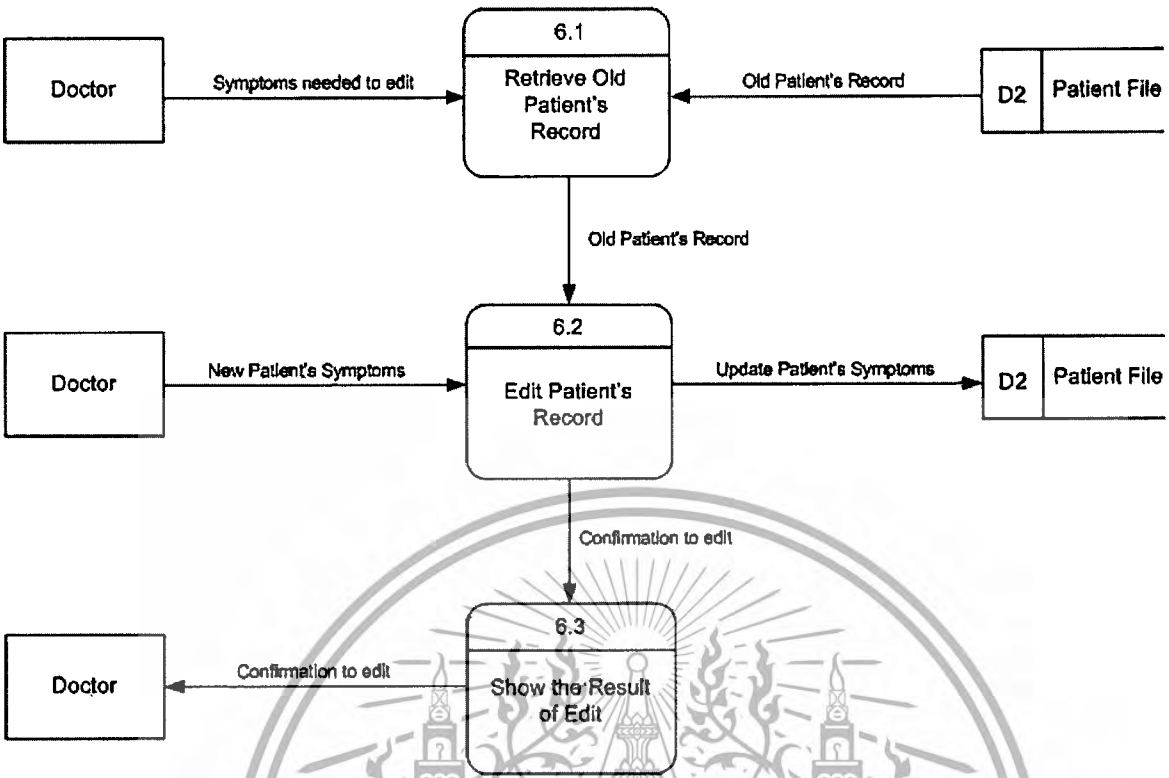


Figure 3.9 Diagram 6 shows process to edit patient's record

Figure 3.9 shows diagram of editing the patient's information. The doctor searches the patient's record which need to edited and go to the section to edit. The system will call an old record in order to edit the patient's record. After that, the system will save the edited record into the patient file and show the result to the doctor.

3.3 ER-Diagram

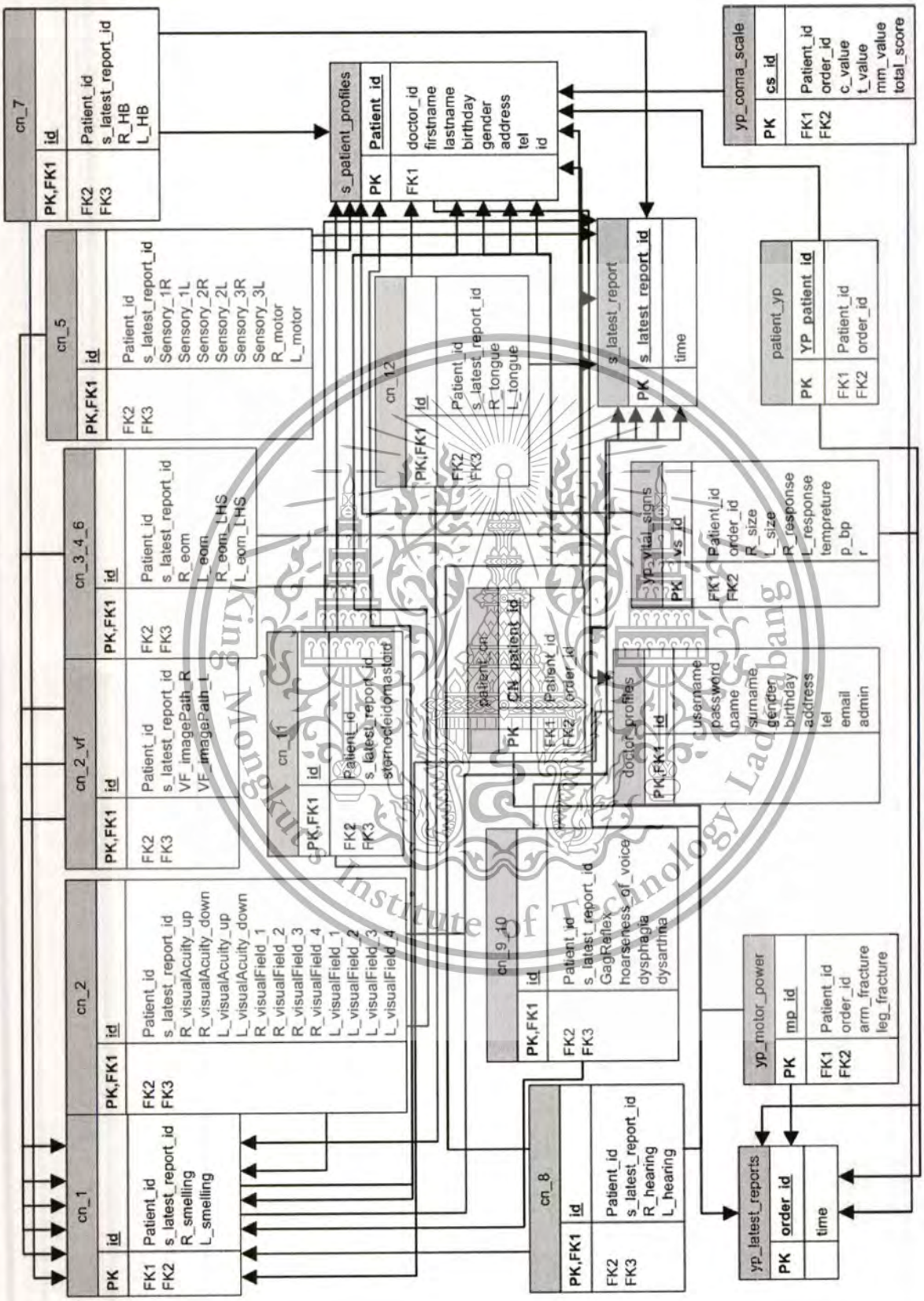


Figure 3.10 ER-diagram of the relationship among the processes in the system

3.4 Database Table

Table 3.1 cn_1

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_smelling	varchar(20)	Anosmia, Good	Anosmia	
L_smelling	varchar(20)	Anosmia, Good	Good	

Table 3.2 cn_2

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_visualAcuity_up	tinyint(4)	Snellen's Chart	20	
R_visualAcuity_down	tinyint(4)	Snellen's Chart	40	
L_visualAcuity_up	tinyint(4)	Snellen's Chart	20	
L_visualAcuity_down	tinyint(4)	Snellen's Chart	40	
R_visualField_1	tinyint(4)	0 = Not Blind, 1 = Blind	0	
R_visualField_2	tinyint(4)	0 = Not Blind, 1 = Blind	0	
R_visualField_3	tinyint(4)	0 = Not Blind, 1 = Blind	1	
R_visualField_4	tinyint(4)	0 = Not Blind, 1 = Blind	0	
L_visualField_1	tinyint(4)	0 = Not Blind, 1 = Blind	1	

Table 3.2 cn_2 (Continue)

Attribute Name	Data type	Description	Example	Key
L_visualField_2	tinyint(4)	0 = Not Blind, 1 = Blind	0	
L_visualField_3	tinyint(4)	0 = Not Blind, 1 = Blind	0	
L_visualField_4	tinyint(4)	0 = Not Blind, 1 = Blind	0	

Table 3.3 cn_2_vf

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
VF_imagePath_R	varchar(100)	Image Address	CN2_1_1_1334822820_R	
VF_imagePath_L	varchar(100)	Image Address	CN2_1_1_1334822820_L	

Table 3.4 cn_3_4_6

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_eom_LHS	tinyint(4)	Right Eye of Movement	-50	
R_eom_RHS	tinyint(4)	Right Eye of Movement	50	
L_eom_LHS	tinyint(4)	Left Eye of Movement	-50	
L_eom_RHS	tinyint(4)	Left Eye of Movement	50	

Table 3.5 cn_5

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_reports_id	int(11)	Latest Treatment	5	FK3
Sensory_1R	tinyint(4)	0 = Good, 1 = Weak	0	
Sensory_1L	tinyint(4)	0 = Good, 1 = Weak	0	
Sensory_2R	tinyint(4)	0 = Good, 1 = Weak	1	
Sensory_2L	tinyint(4)	0 = Good, 1 = Weak	1	
Sensory_3R	tinyint(4)	0 = Good, 1 = Weak	0	
Sensory_3L	tinyint(4)	0 = Good, 1 = Weak	0	
R_motor	varchar(20)	Good, Weak	Good	
L_motor	varchar(20)	Good, Weak	Good	

Table 3.6 cn_7

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_HB	varchar(20)	Facial Response	Weak Less	
L_HB	varchar(20)	Facial Response	No Movement	

Table 3.7 cn_8

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_hearing	varchar(20)	Hearing Response	Serviceable	
L_hearing	varchar(20)	Hearing Response	Non-Serviceable	

Table 3.8 cn_9_10

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
GagReflex	varchar(20)	Good, No Gag	Good	
hoarseness_of_voice	varchar(20)	Good, Weak	Good	
dysphagia	varchar(20)	Good, Weak	Weak	
dysarthria	varchar(20)	Good, Weak	Good	

Table 3.9 cn_11

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
sternocleidomastoid	varchar(20)	Good, Weak	Good	

Table 3.10 cn_12

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK, FK1
Patient_id	int(11)	Patient Code	250	FK2
s_latest_report_id	int(11)	Latest Treatment	5	FK3
R_tongue	varchar(20)	Good, Atrophy	Atrophy	
L_tongue	varchar(20)	Good, Atrophy	Good	

Table 3.11 doctor_profiles

Attribute Name	Data type	Description	Example	Key
id	int(11)	Treatment Order	100	PK
username	varchar(20)	Doctor's Username	twsjwtynch	
password	varchar(20)	Doctor's Password	********	
name	varchar(50)	Doctor's Name	Taweesak	
surname	varchar(50)	Doctor's Surname	Janwitayanuchit	
gender	char(1)	M, F	M	
birthday	date	Y/M/D	1980-08-08	
address	text	Doctor's Address	Ramkhamhaeng	
tel	varchar(20)	Doctor's Telephone	081-234-5678	
email	varchar(50)	Doctor's Email	taweesak@rm.com	
admin	int(1)	0 = Not, 1 = Admin	1	

Table 3.12 patient_cn

Attribute Name	Data type	Description	Example	Key
CN_patient_id	int(11)	Cranial Nerve Order	10	PK
Patient_id	int(11)	Patient Code	250	FK1
order_id	int(11)	Latest Neurological Observation Order	20	FK2

Table 3.13 patient_yp

Attribute Name	Data type	Description	Example	Key
YP_patient_id	int(11)	Neurological Observation Order	50	PK
Patient_id	int(11)	Patient Code	250	FK1
order_id	int(11)	Latest Neurological Observation Order	20	FK2

Table 3.14 s_latest_report

Attribute Name	Data type	Description	Example	Key
s_latest_report_id	int(11)	Latest Treatment	5	PK
time	timestamp	CURRENT_TIMESTAMP	2012-03-01 13:58:00	

Table 3.15 s_patient_profiles

Attribute Name	Data type	Description	Example	Key
Patient_id	int(11)	Patient Code	250	PK
doctor_id	int(11)	Doctor Code	5	FK1
firstname	varchar(20)	Patient's Name	Eakawat	

Table 3.15 s_patient_profiles (Continue)

Attribute Name	Data type	Description	Example	Key
lastname	varchar(20)	Patient's Surname	Tantamjarik	
birthday	date	Y/M/D	1990/12/19	
gender	char(1)	M, F	M	
address	text	Patient's Address	Ramhkamhaeng	
tel	varchar(20)	Patient's Telephone	082-345-6789	

Table 3.16 yp_coma_scale

Attribute Name	Data type	Description	Example	Key
cs_id	int(11)	Coma Scale Order	5	PK
Patient_id	int(11)	Patient Code	250	FK1
order_id	int(11)	Latest Neurological Treatment	20	FK2
c_value	tinyint(4)	Coma Scale Score (1-5)	5	
t_value	tinyint(4)	Tube Score (1-5)	4	
mm_value	tinyint(4)	Movement Score (1-5)	5	
total_score	tinyint(4)	Sum of Three Scores	14	

Table 3.17 yp_latest_reports

Attribute Name	Type	Description	Example	Key
order_id	int(11)	Latest Neurological Observation Order	20	PK
time	timestamp	CURRENT_TIMESTAMP	2012-03-04 11:57:00	

Table 3.18 yp_motor_power

Attribute Name	Data Type	Description	Example	Key
mp_id	int(20)	Motor Power Order	10	PK
Patient_id	int(11)	Patient Code	250	FK1
order_id	int(11)	Latest Neurological Observation Order	20	FK2
arm_fracture	varchar(20)	Ability of Arm Movement	Ab. flexion	
leg_fracture	varchar(20)	Ability of Leg Movement	Ab. extension	

Table 3.19 yp_vital_signs

Attribute Name	Data type	Description	Example	Key
vs_id	int(11)	Vital Sign Order	10	PK
Patient_id	int(11)	Patient Code	250	FK1
order_id	int(11)	Latest Neurological Observation Order	20	FK2
R_size	tinyint(4)	Right Pupil's Size (1-8)	8	
L_size	tinyint(4)	Left Pupil's Size (1-8)	8	
R_response	varchar(20)	Right Pupil's Response	Good	
L_response	varchar(20)	Left Pupil's Response	Medium	
tempreture	decimal(3,1)	Celsius	35.0	
p_bp	decimal(3,1)	Blood Pressure	120	
r	tinyint(4)	Respiration Rate	75	

Chapter 4

Implementation

4.1 Web Application

The development of Neurological Patients Information Management System is used by doctors to keep and manage each patient's information in order to help the doctors make decisions before diagnosis or surgery. The doctors can use the system through the web browser via tablets.

4.2 User Interface

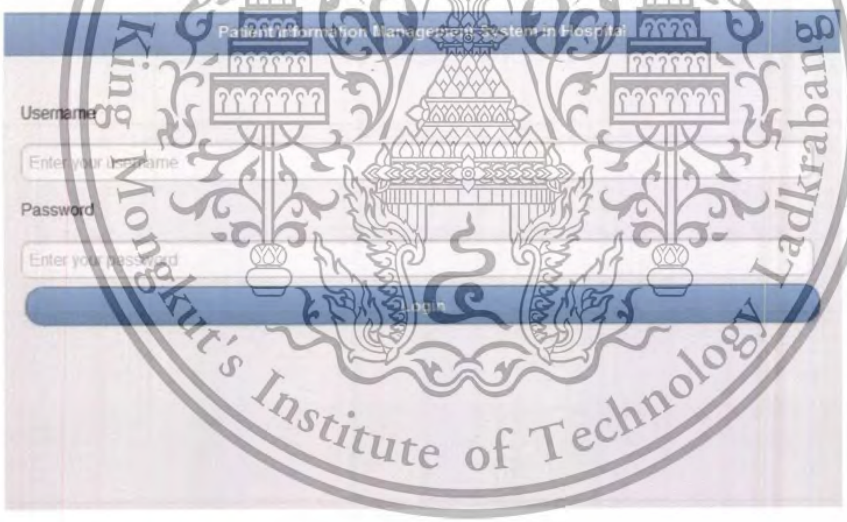


Figure 4.1 The first page of the web application

Figure 4.1 shows the first page of the web application. The doctor needs to log into the system by using a username and a password. If the username and the password are matched, the doctor is able to access to the system shown in Figure 4.2.

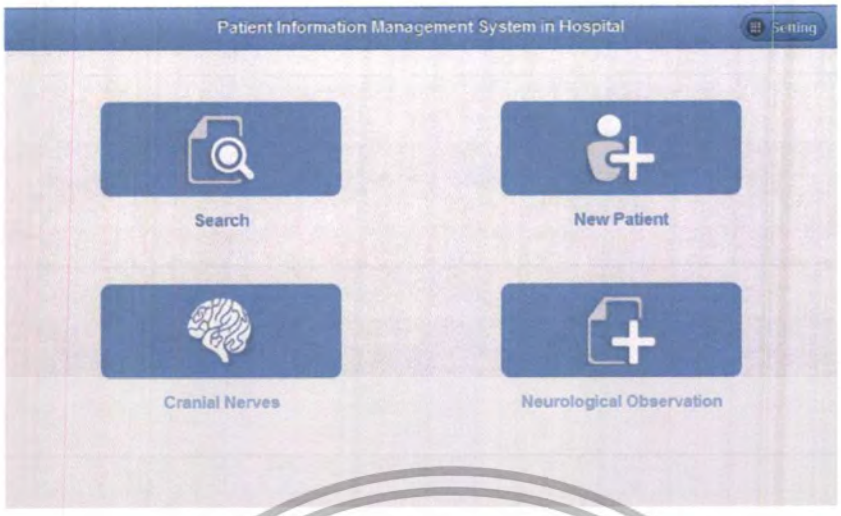


Figure 4.2 The index page of the web application

The index page shows the main functions of the web application which consists of search, new patient, cranial nerves and neurological observation.

Figure 4.3 Setting

Figure 4.3 shows the setting 

on the navigation bar which provides the link to view the doctor's profile, the patient's profile and also log out from the system.

Figure 4.4 Doctor's profile page

Figure 4.4 shows the doctor's profile page which enables the doctor to edit his/her profile by pressing on Edit and Save buttons in order to save the edited information.

Figure 4.5 Search page

Figure 4.5 shows the search page where the doctor can search the patient's profile in order to view a report or create a new record of cranial nerves or neurological observation. If the patient's profile is not stored in the system, the doctor needs to create a new patient's

profile before creating a new record of cranial nerves or neurological observation as shown in Figure 4.6.

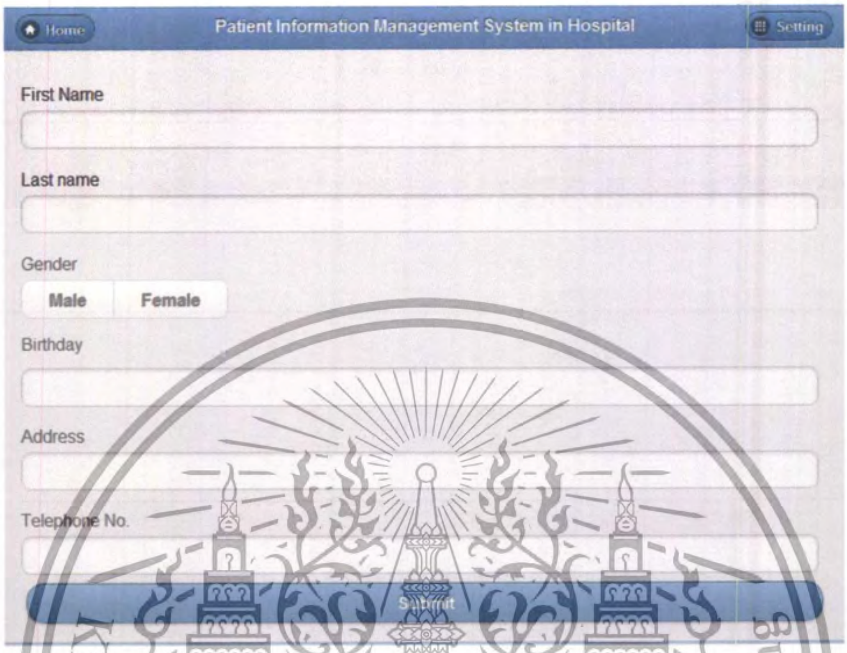


Figure 4.6 Create the new patient's profile

Figure 4.7 Patient information page

Figure 4.7 shows the patient's information page which enables the doctor to edit the patient's profile and create the new record of cranial nerves and neurological observation. The doctor can create a new record of cranial nerves by pressing on New Cranial Nerves button or New Neurological Observation button in order to create a new record of neurological observation. The table below the patient's profile shows the latest record of cranial nerves and neurological observation. The doctor can press Edit button to edit the old records, Report button to view the report, graph button in order to view the graph, and View button in order to view the history of records.

Figure 4.8 Cranial nerves report

Figure 4.8 shows the cranial nerves report of the selected time and date.

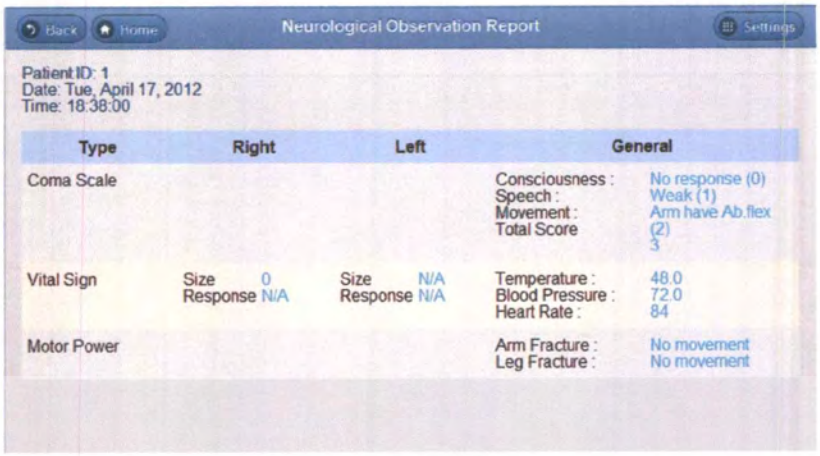


Figure 4.9 Neurological observation report

Figure 4.9 shows the neurological observation report of the selected time and date.

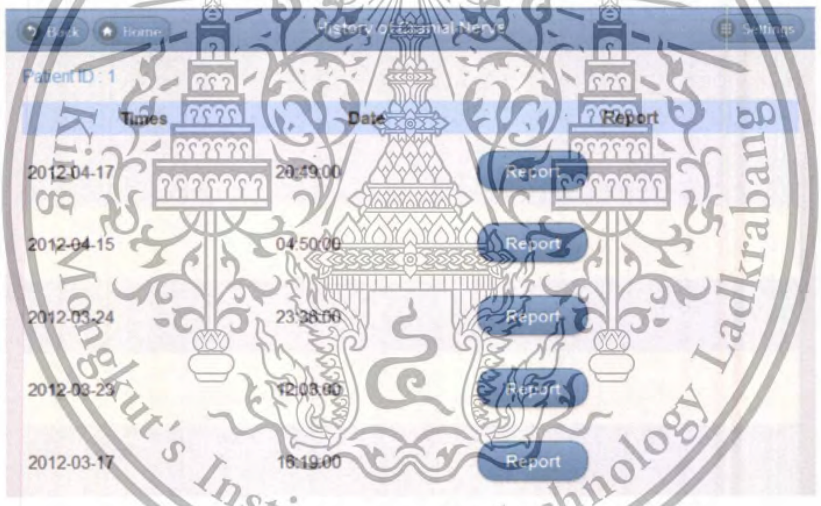
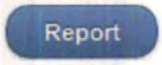


Figure 4.10 History of cranial nerves

Figure 4.10 shows the history of cranial nerves which the doctor can press the



button to view the report.

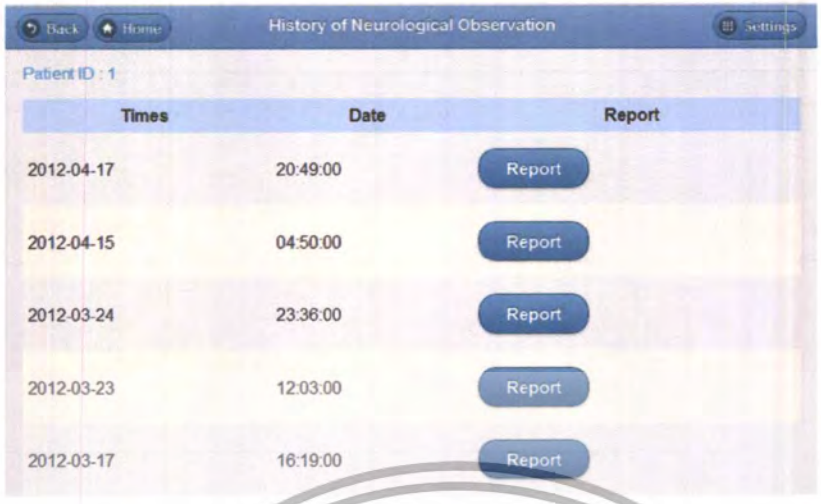


Figure 4.11 The history of neurological observation

Figure 4.11 shows the history of neurological observation which the doctor can press

Report

button to view the report as graphs shown in Figure 4.12 and Figure 4.13.

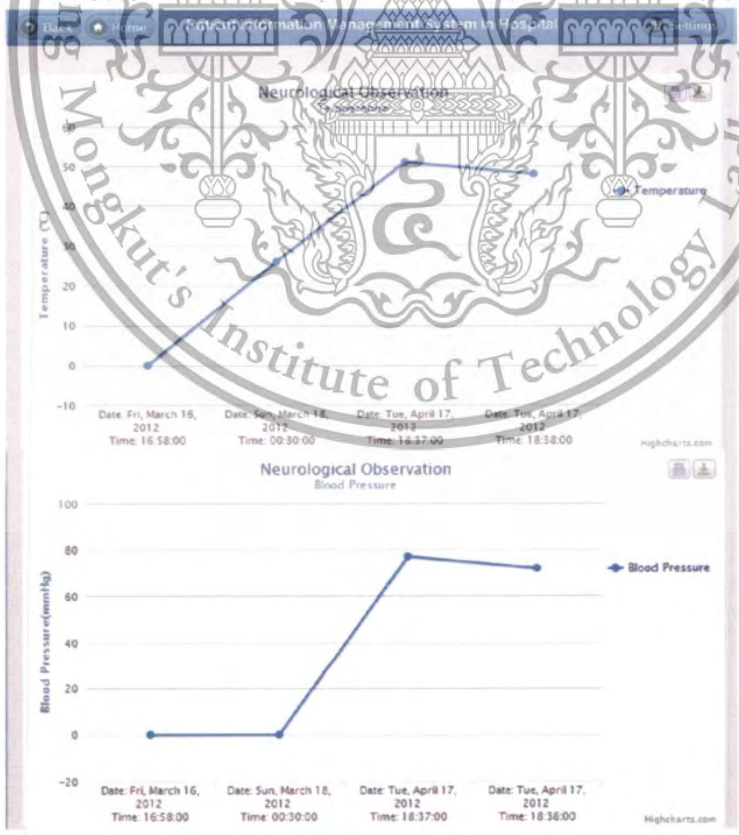


Figure 4.12 Temperature and blood pressure graphs

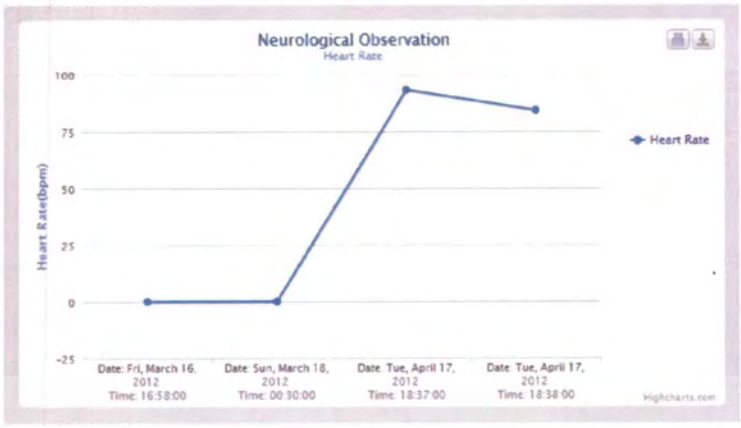


Figure 4.13 Heart rate graph

Figure 4.14 Main page of cranial nerves

Figure 4.14 shows the main functions of the web application which are used in recording the symptoms of twelve pairs of cranial nerves. The buttons on the man's face indicate the positions of the cranial nerves where the doctor can link to the pages in order to record cranial nerves symptoms. The doctor needs to select date and time before recording the symptoms.

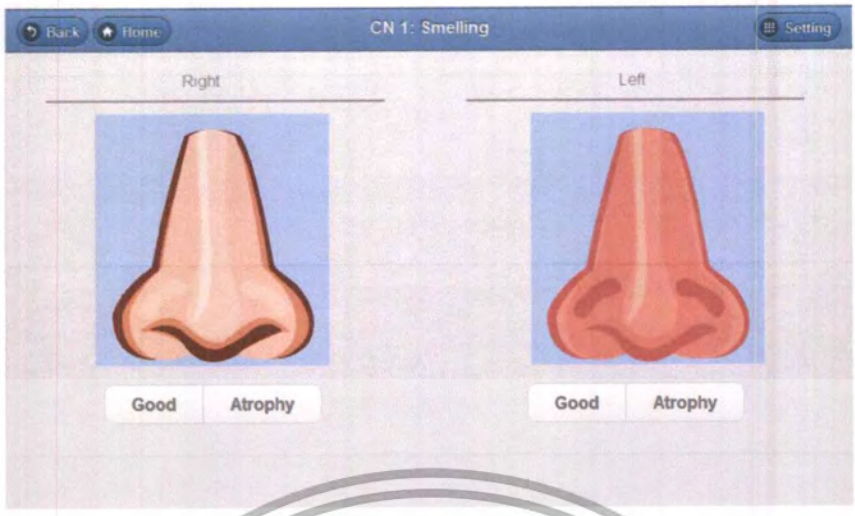


Figure 4.15 CN 1: Smelling

CN 1 in Figure 4.15 is shown about the sense of which is divided into left and right sides. The doctor can select one of the provided choices.

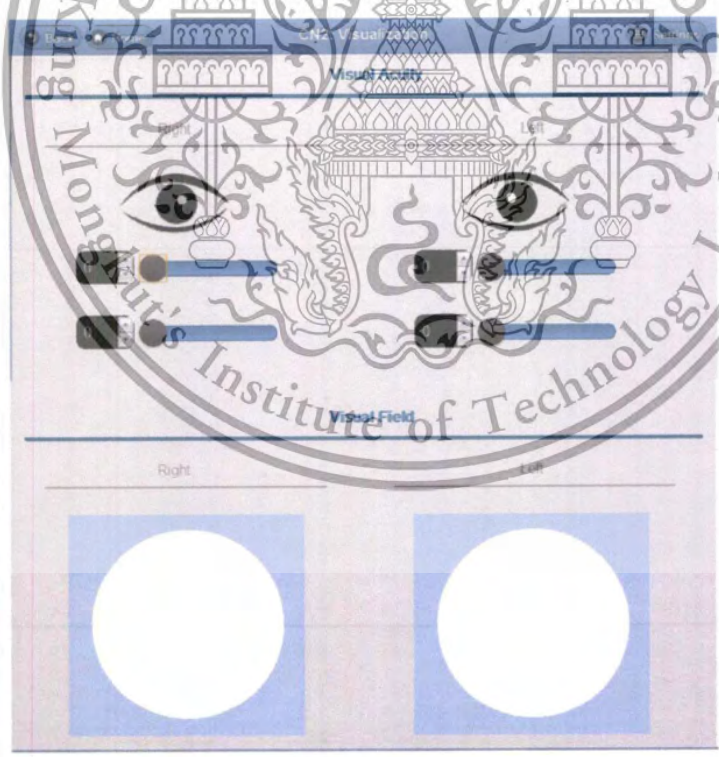


Figure 4.16 CN 2:1 Visualization

CN 2 in Figure 4.16 and Figure 4.17 are the transmission of visual signals from the retina of the eyes to the brain. It consists of two sections including visual acuity and visual field. In the visual acuity section, the doctor can slide the slider bars  to fill in the values. In the visual field section, the doctor can press on the quarter of the circle where the abnormal occurred.

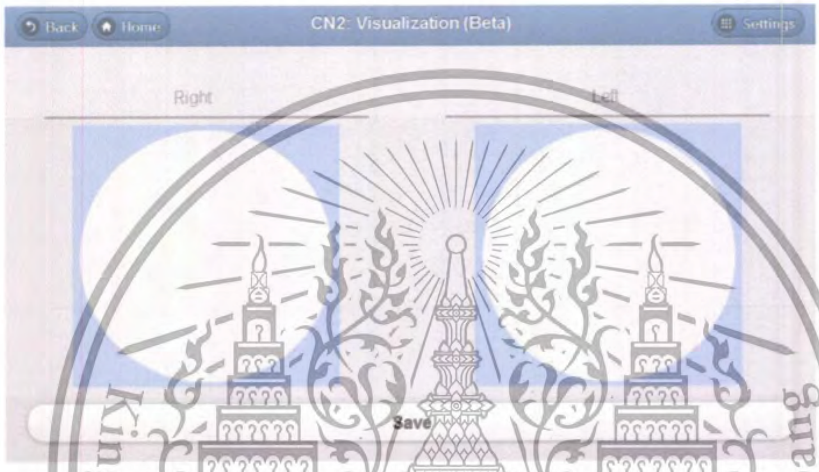


Figure 4.17 CN 2:2 Visualization

CN 2:2 in Figure 4.17 is the same as visual field in CN 2:1 but it allows the doctor to draw the blind position of the eyes.

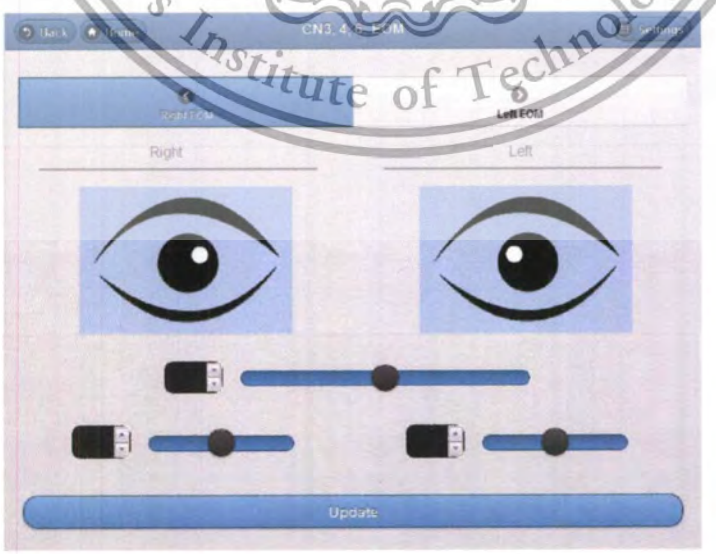


Figure 4.18 CN 3, 4, 6: EOM

CN 3, 4, 6 in Figure 4.18 is the performance of the eye movement of both sides. The doctor can slide the top slider bar  to fill in the same values of both eyes or the slider bars on the right and left to fill in the different values of eye movement on each side.

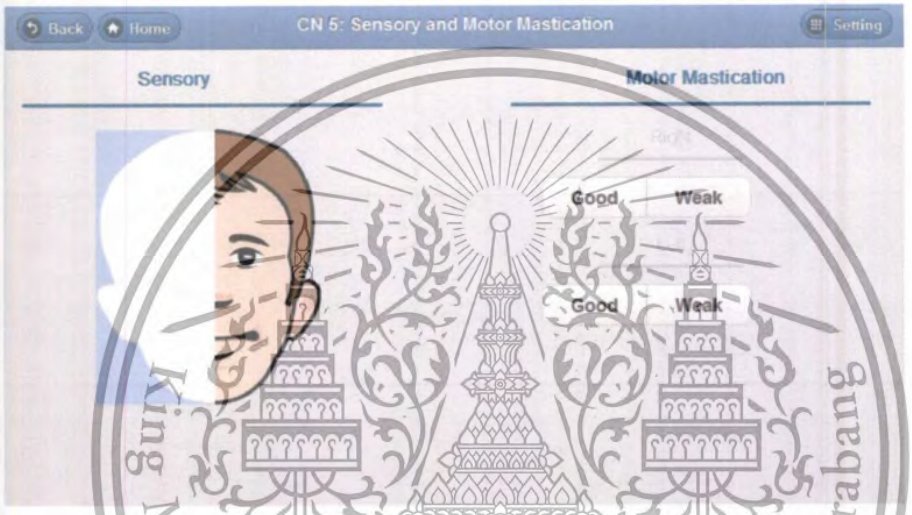


Figure 4.19 CN 5: Sensory and motor mastication

CN 5 in Figure 4.19 shows the sensation of the face and innervates the muscles of mastication. In the sensory section, the doctor can select the abnormal position of the face. In the motor mastication, the doctor can choose the patient's symptom in both sides.

Figure 4.20 CN 7: Facial (Brackman scale)

CN 7 in Figure 4.20 is the motor innervation to the muscles of facial expression. The doctor can choose the patient's symptom in both sides.

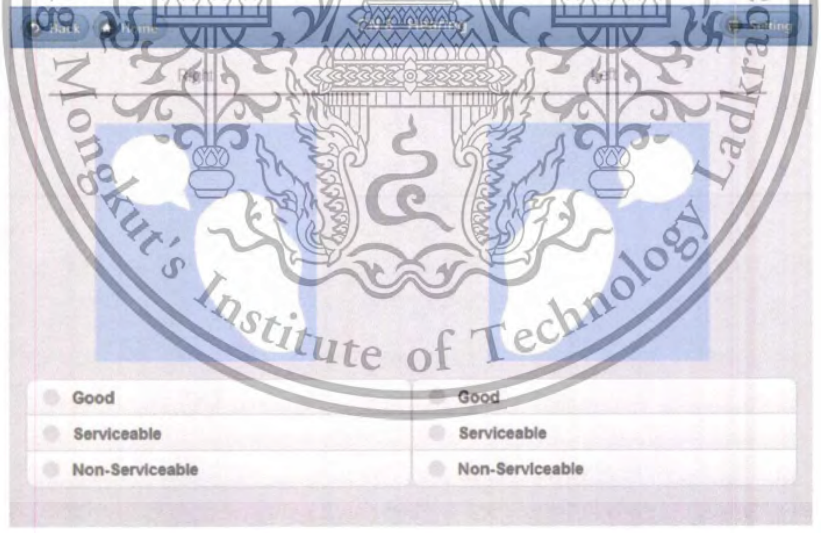


Figure 4.21 CN 8: Hearing

CN 8 in Figure 4.21 is about the senses of sound, rotation, and gravity. The doctor can choose the patient's symptom in both sides.

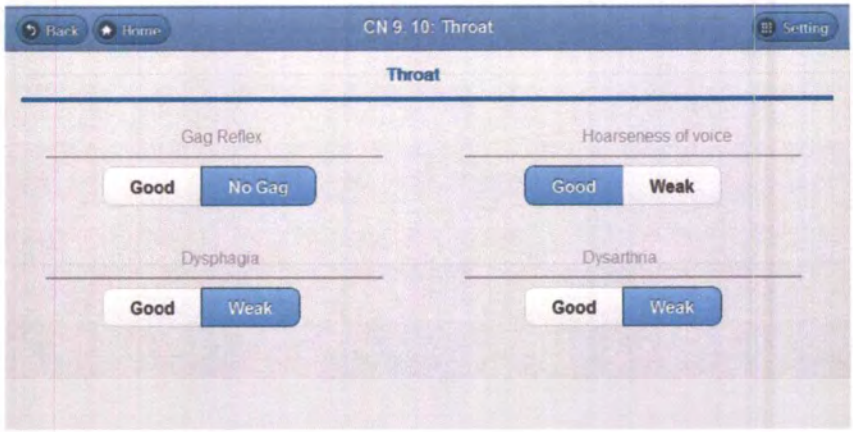


Figure 4.22 CN 9, 10: Throat

CN 9, 10 in Figure 4.22 is the muscle control of voice, resonance, and the soft palate. It consists of four sections. The doctor can choose the symptom from the given choices.

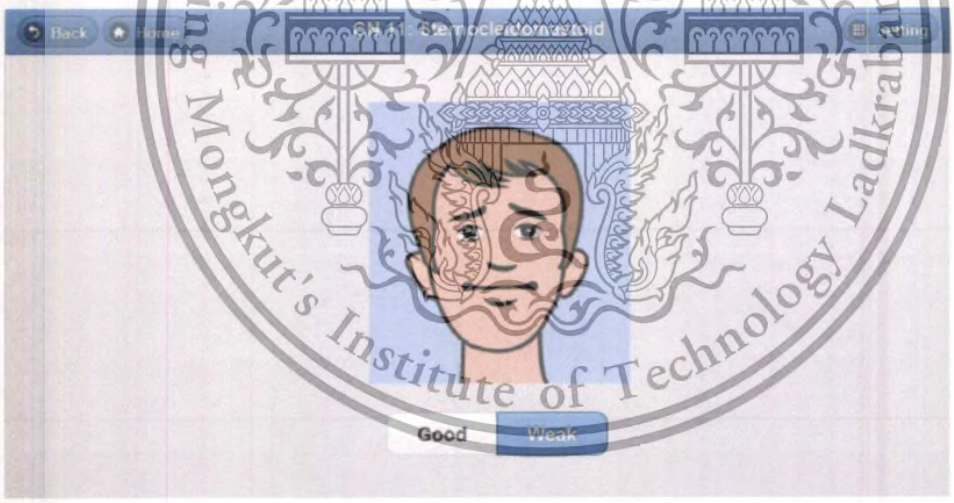


Figure 4.23 CN 11: Sternocleidomastoid

CN 11 in Figure 4.23 is about the control of sternocleidomastoid and trapezius muscles. The doctor can choose either good or weak symptom from the given choices.

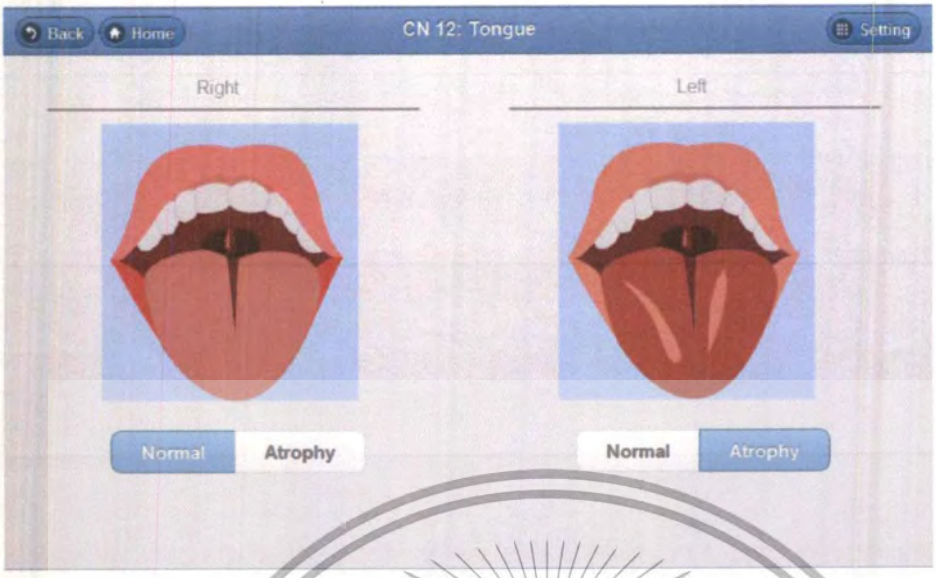


Figure 4.24 CN 12: Tongue

CN 12 in Figure 4.24 is about the motor innervation to the muscles of the tongue. The doctor can select either normal or atrophy symptom from the given choices.

Figure 4.25 Main page of neurological observation

Figure 4.25 shows the other function of the web application which is used in recording the neurological observation symptoms. It consists of three main functions which are coma

scale, vital signs, and motor power. The doctor needs to select date and time before recording the symptoms.

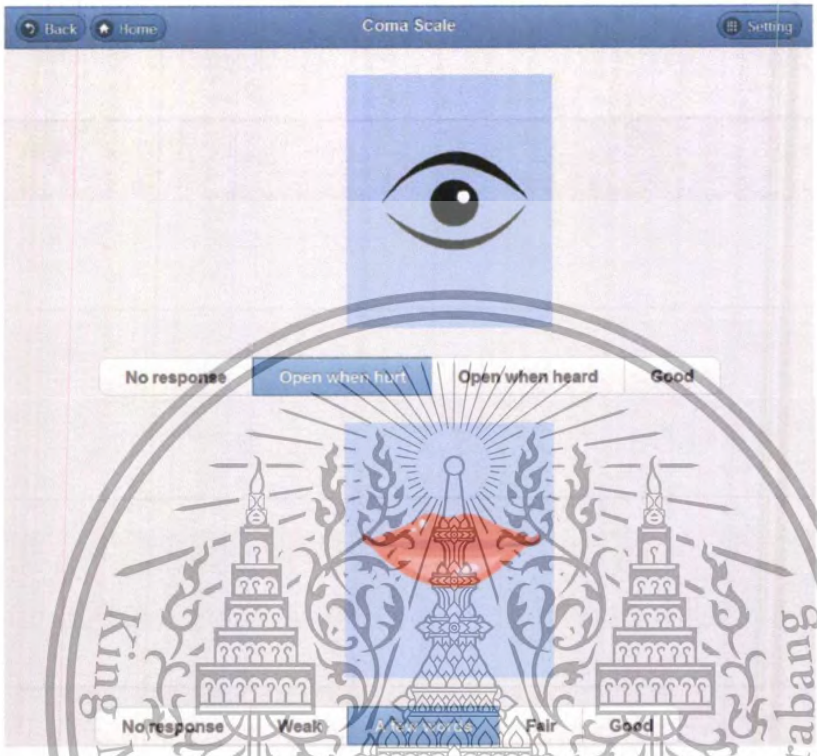


Figure 4.26 Coma scale

Coma scale in Figure 4.26 is used in recording the conscious state of a person from the beginning to subsequent assessment. The doctor can choose the given choices of patient's symptom.

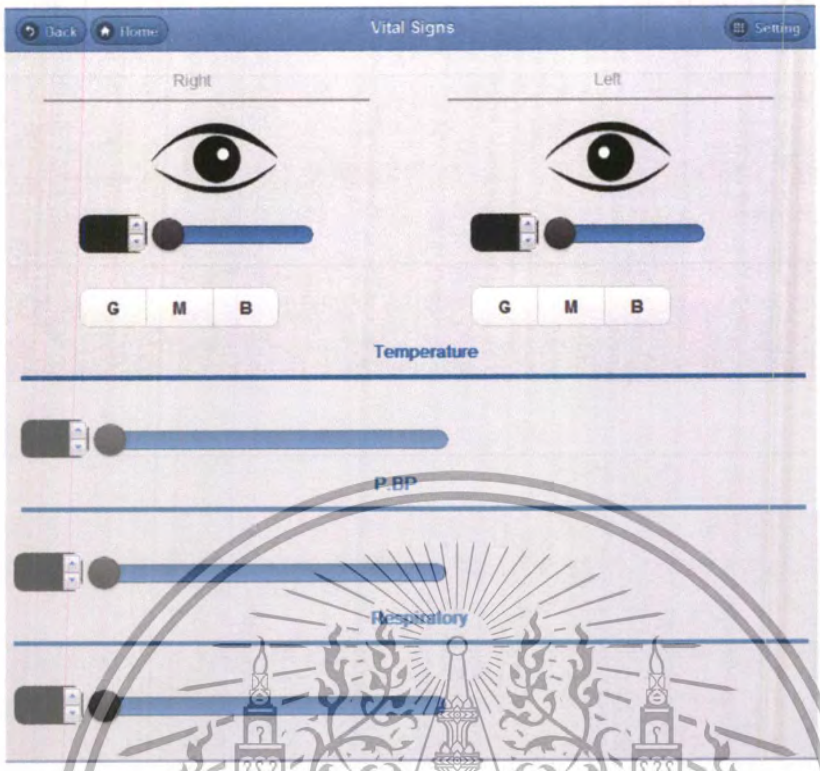


Figure 4.27 Vital signs

Vital signs in Figure 4.27 is used in recording the body temperature, pulse rate (or heart rate), blood pressure, and respiratory rate. The doctor can slide the slider bars  and select radio buttons to record the values.

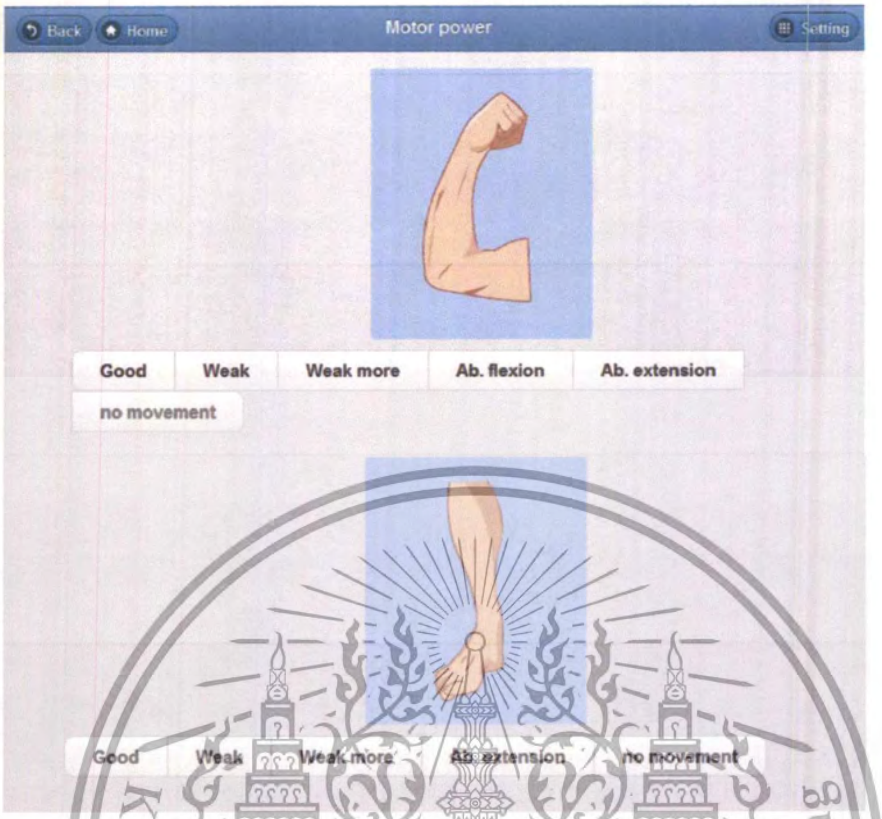


Figure 4.28 Motor power

Motor power in Figure 4.28 is used in recording the movement of arms and legs. The doctor can select the patient's symptom from the given choices.

4.3 Administrator Interface

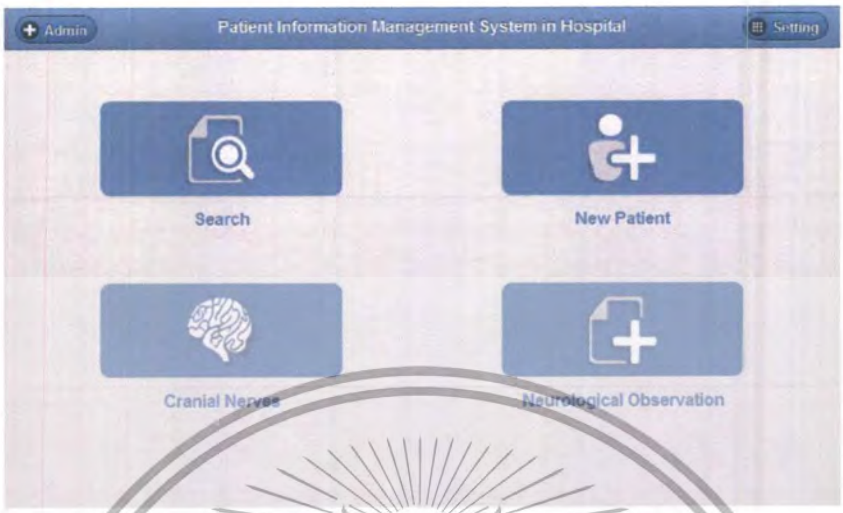


Figure 4.29 Administrator setting

Figure 4.29 shows the index page of the web application when the doctor logs into the system as the administrator, the admin button appears on the navigation bar which enables the administrator to add new users (doctors) into the system. It also shows the main functions of the web application which are search, new patient, cranial nerves, neurological observation, and user setting button on the right side of menu bar. When the administrator presses on an admin button on the menu bar, the system will link to the search doctor's page where the administrator can search for a doctor or add a new doctor into the system shown in Figure 4.30.

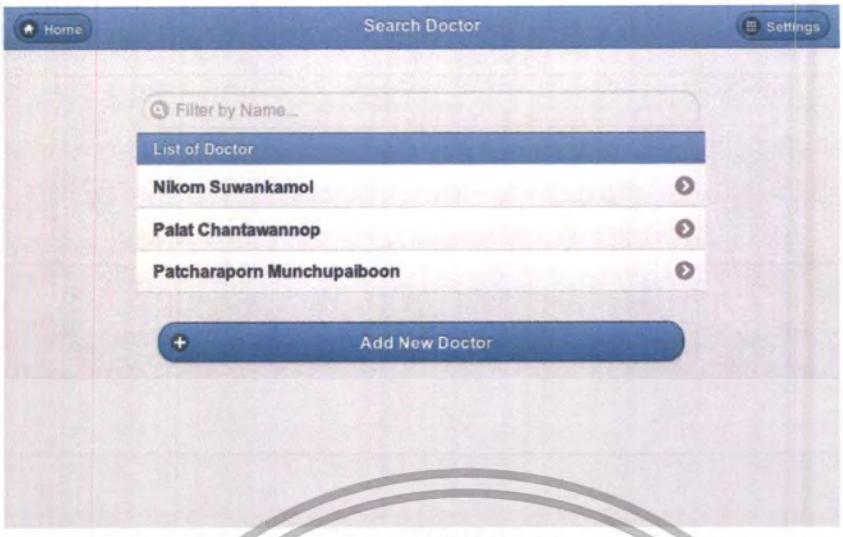


Figure 4.30 View of doctors list

Figure 4.30 shows the lists of doctors who can use the system. The administrator can press the link in order to show the doctors' profiles as shown in Figure 4.31. In this page, the administrator can add new doctors to the lists by pressing on Add New Doctor

button.

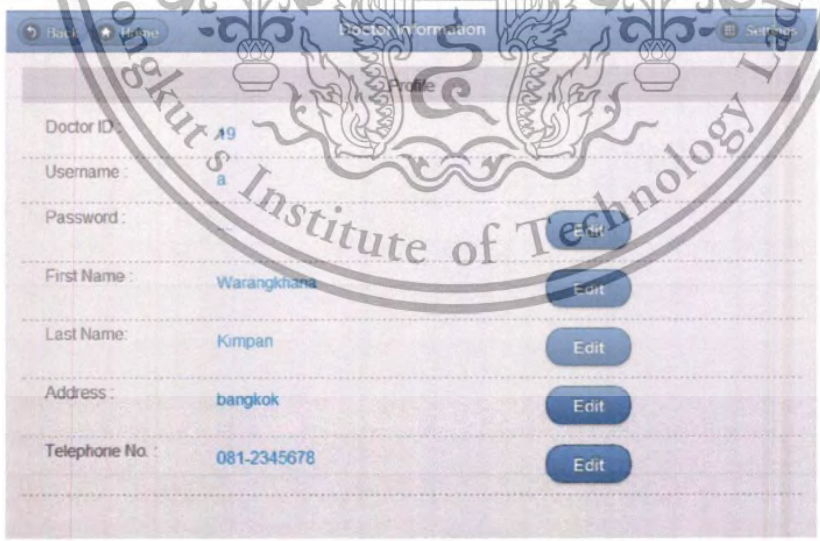


Figure 4.31 Doctor's information page

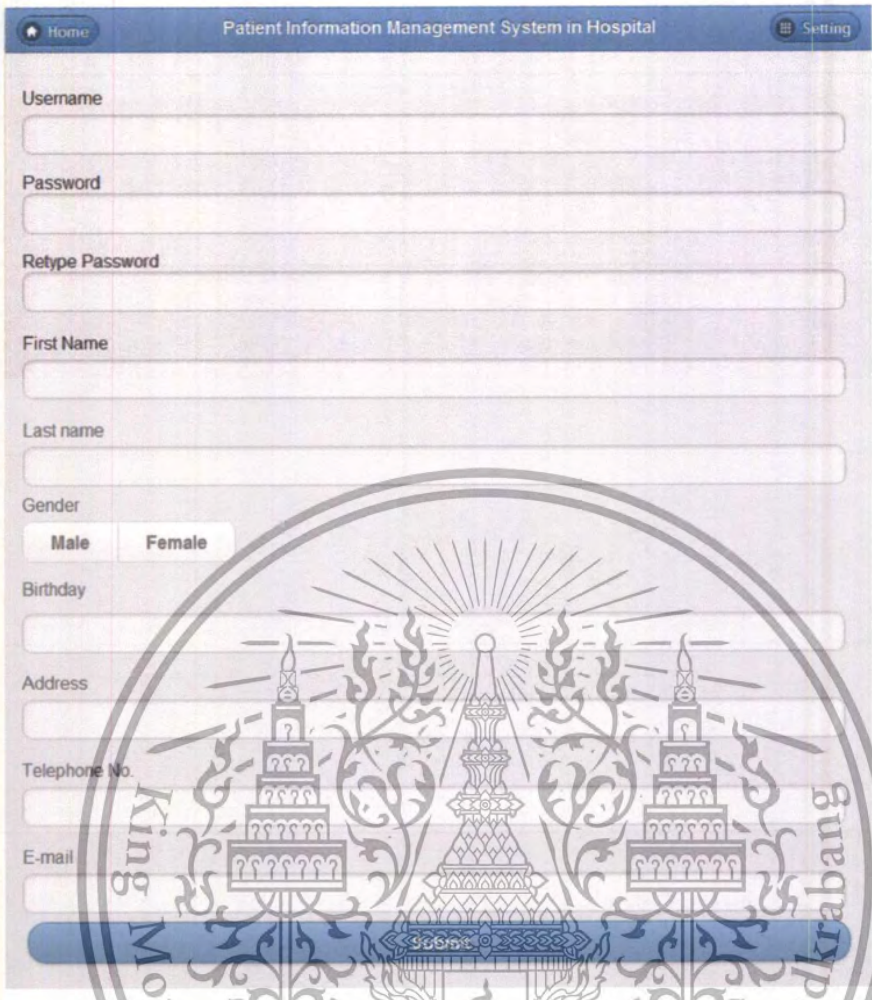


Figure 4.32 Create the new doctor's profile

Figure 4.32 shows a form to create a new doctor's profile.

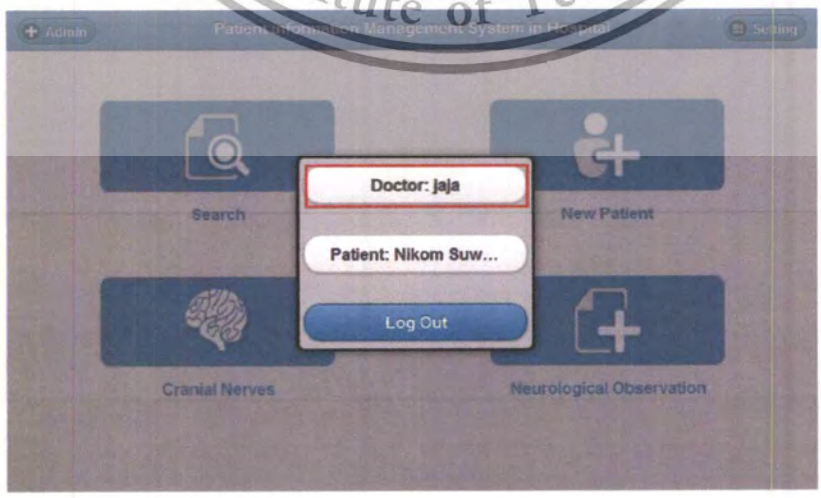


Figure 4.33 Setting

The setting button on the menu bar in Figure 4.33 shows the doctor's name, patient's name, and log out button to log out from the system. The administrator can edit his or her profile as shown in Figure 4.34.

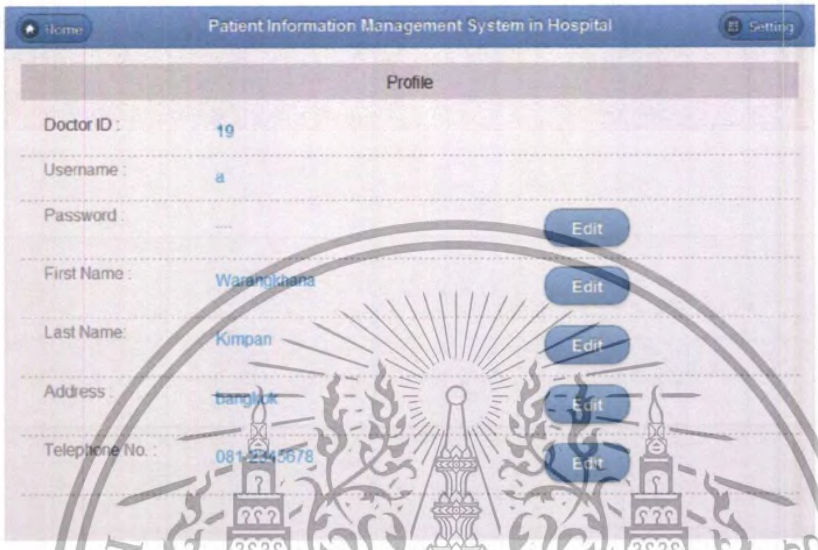


Figure 4.34 View of administrator's information

Chapter 5

Conclusion

5.1 Conclusion

This special problem proposed the Neurological Patient Information Management System. The system focuses on the important functions that are necessary for the users or doctors, for example, the functions that are used to record the symptoms of twelve pairs of cranial nerves and the symptoms of neurological observation. The web application was developed in order to make it easy for the users to use the system with tablet PCs. The system can create the new record, edit the record and view the report in the form of graphs of cranial nerves and neurological observation. The doctors can use the report to compare the patient's symptoms at the different time in order to make a decision before diagnosis or surgery.

5.2 Problems and Solutions

1. The problem occurred when using some functions in jQuery Mobile framework in combination with JavaScripts. For example, the transition effects in jQuery Mobile Framework was not compatible with the OnClick JavaScript function.

2. The problem occurred when exploring the website through the different screen resolutions. The solution was the separation of the CSS files from the different size of the screen.

3. The operation of the web application was slow because of the loading of a lot of codes from the server. The solution was to optimize the codes and use only the necessary functions in order to make the web application faster in loading.

4. The canvas function did not suitable when using on the tablet PCs because it required a lot of resources, and the drawing position was not as accurate as using on the desktop PCs.

5.3 Limitations of This Special Problem

1. The position of layout was suitable only when opening in landscape mode.

2. The application was suitable only with MySQL database.

3. The application can only run on MySQL UNIX socket.

4. The picture drawing by canvas could not be saved and the drawing was not so smooth.

5. The sessions had a time limitation.

5.4 Future Works

1. Add more functions in the application.

2. Design the CSS in the multiple sizes of the screen resolution.

3. Decorate a more beautiful user's interface.

References

[1][Online] http://www.w3schools.com/html/html_intro.asp

[2][Online] http://www.html5rocks.com/en/

http://slides.html5rocks.com/#landing-slide

[3][Online] http://en.wikipedia.org/wiki/PHP

http://www.w3schools.com/PHP/php_syntax.asp

http://www.w3schools.com/php/php_variables.asp

http://www.w3schools.com/php/php_operators.asp

http://www.w3schools.com/php/default.asp

[4][Online] http://www.w3schools.com/js/js_intro.asp

http://en.wikipedia.org/wiki/JavaScript

http://www.exforsys.com/tutorials/javascript/using-javascript-in-html-page.html

[5][Online] http://en.wikipedia.org/wiki/Cascading_Style_Sheets

http://www.exforsys.com/tutorials/css/css-applying-css.html

[6][Online] http://en.wikipedia.org/wiki/JQuery

http://www.w3schools.com/jquery/jquery_syntax.asp

[7][Online] http://jquerymobile.com/demos/1.1.0/

http://www.jqmgallery.com/jquery-mobile-tutorials/

[8][Online] http://en.wikipedia.org/wiki/Ajax_%28programming%29

http://www.noupe.com/ajax/how-ajax-works.html

http://www.w3schools.com/ajax/ajax_xmlhttprequest_send.asp

References (Continue)

http://www.w3schools.com/wsdl/wsdl_uddi.asp

http://www.w3schools.com/wsdl/wsdl_documents.asp

Appendices

Appendix A.

User Manual of Patient Information Management System

User Manual

1. When the doctor opens the web site, he/she needs to fill in the username and the password in order to access into the system.

Figure A.1 The first page of the web application

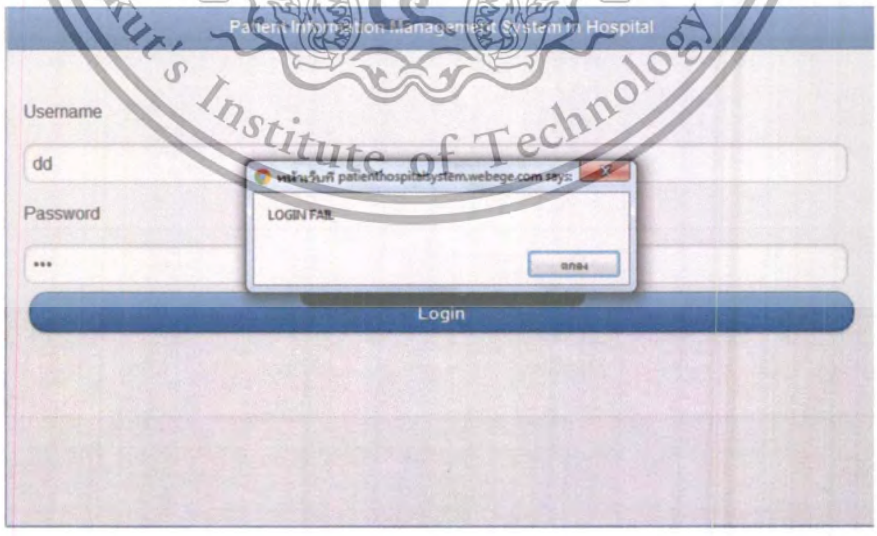


Figure A.2 When the username and password are incorrect

2. When the doctor logs into the system. The system redirect to the index page which shows the four main functions of the web application including search, new patient, cranial nerves, and neurological observation.

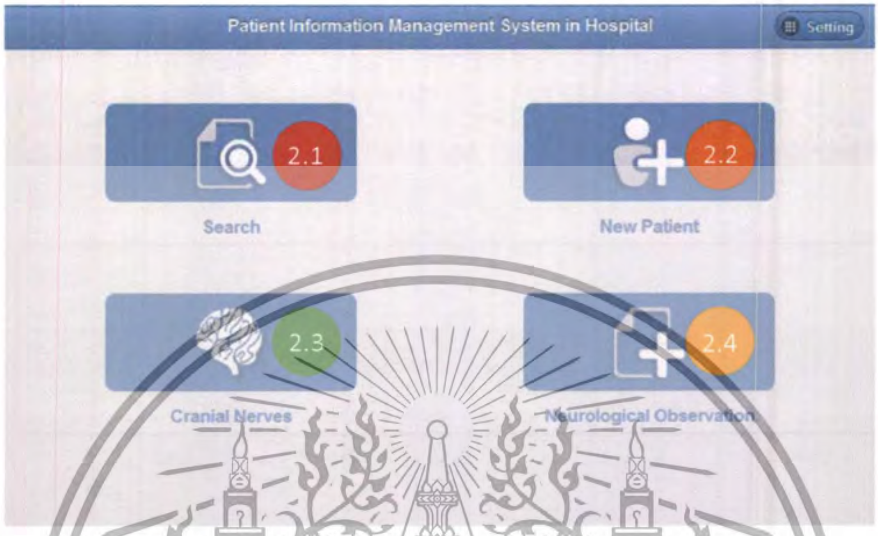


Figure A.3 The index page of the web application

2.1  (Search Function) allows the doctor to search the patient's profile in order to view a report or create a new record of cranial nerves or neurological observation. If the patient's profile is not stored in the system, the doctor needs to create a new patient's profile before creating a new record of cranial nerves or neurological observation by pressing on Add New Patient 

.

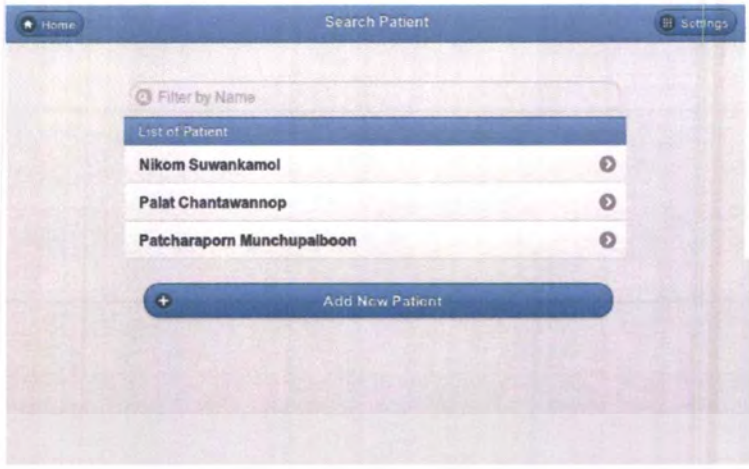
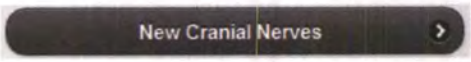


Figure A.4 Search page

Figure A.5 New patient page to create the new patient's profile

When the doctor selects the patient's name on the search page. The system **will redirect to the patient information page which enable the doctor to edit patient's profile and create the new record of cranial nerves and neurological observation. The table below the patient's profile shows the latest record of cranial nerves and neurological observation. The doctor can create a new record of cranial nerves or neurological**

observation from this page by pressing on 

button or 

button.

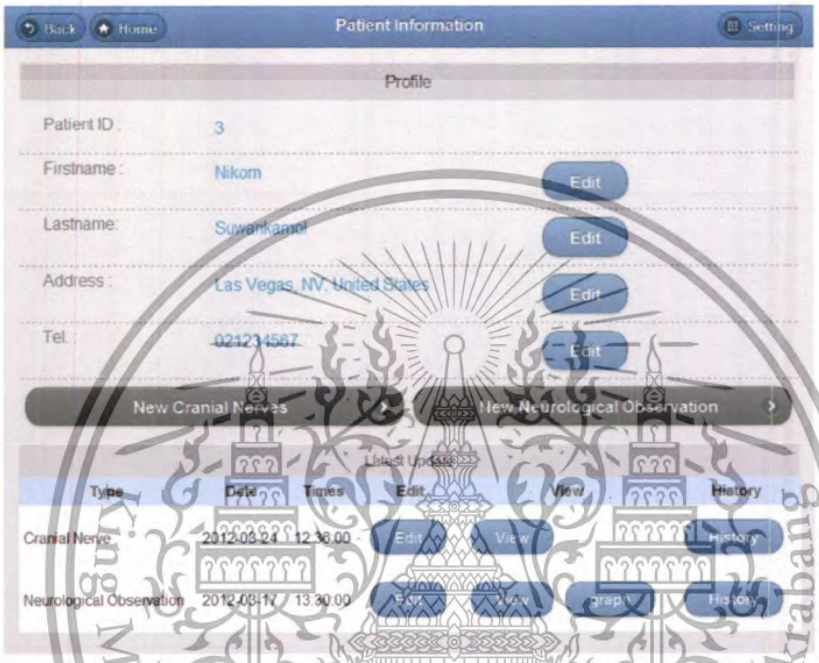


Figure A.6 Patient information page

The doctor enables to edit the patient information by pressing on 

button and 

button in order to save the edited information. The doctor can go back to the previous page(search page) by pressing on 

button, 

button to back the main menu and 

button to see the doctor's profile, the patient's profile or log out from the system.

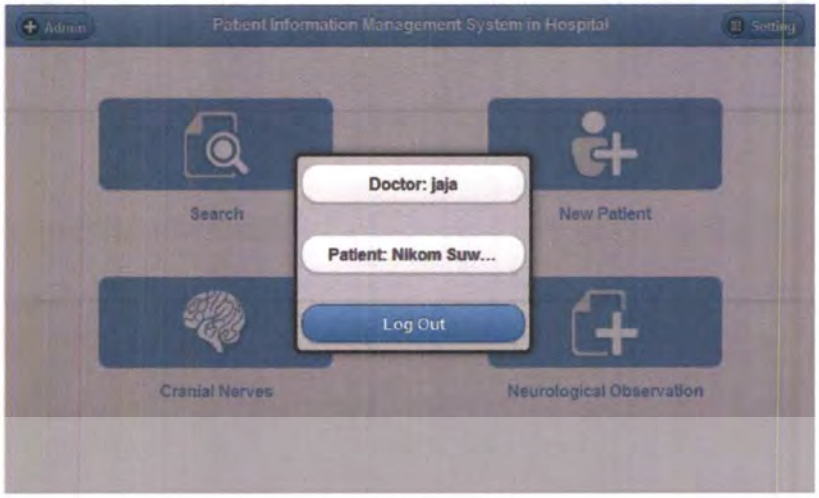


Figure A.7 Setting button shows the link to the doctor's profile, the patient's profile and log out from the system

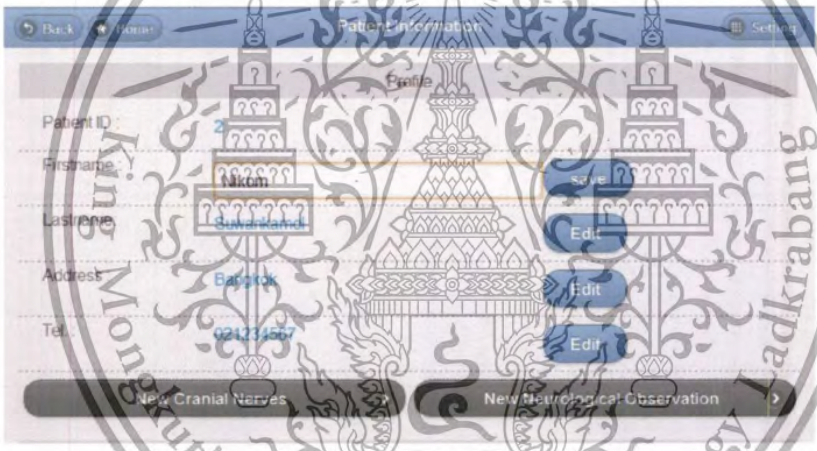


Figure A.8 When the doctor edits the patient's profile

The latest update shows the latest record of cranial nerves and neurological observation. The doctor enables to edit the latest of cranial nerves or neurological observation record by pressing on Edit button, View button to view the latest report, graph button to view the graph, and History button to view all the patient reports. In history page, the doctor can press on Report button in order to view the report of the selected time.

Figure A.9 Latest update of cranial nerves and neurological observation

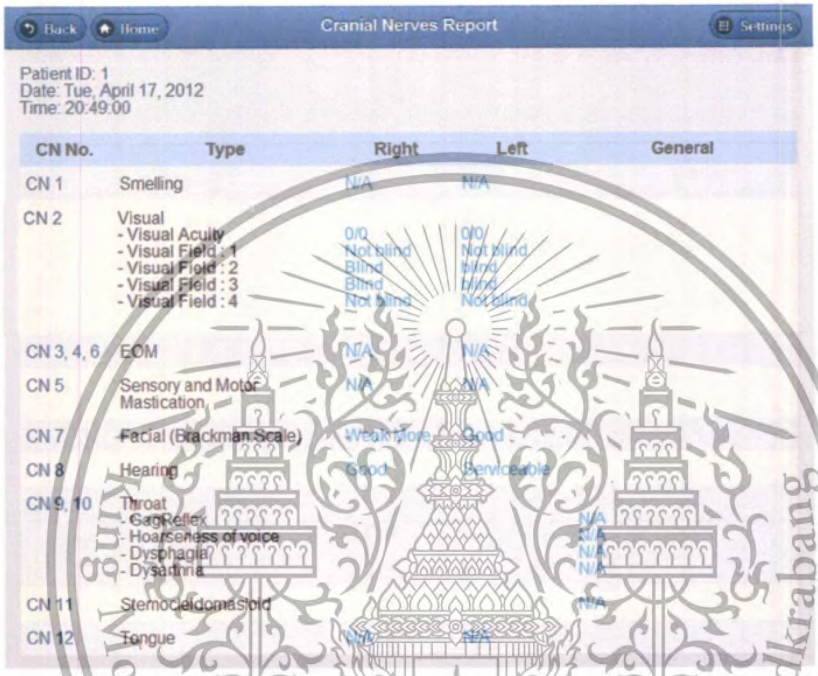


Figure A.10 View of cranial nerves report

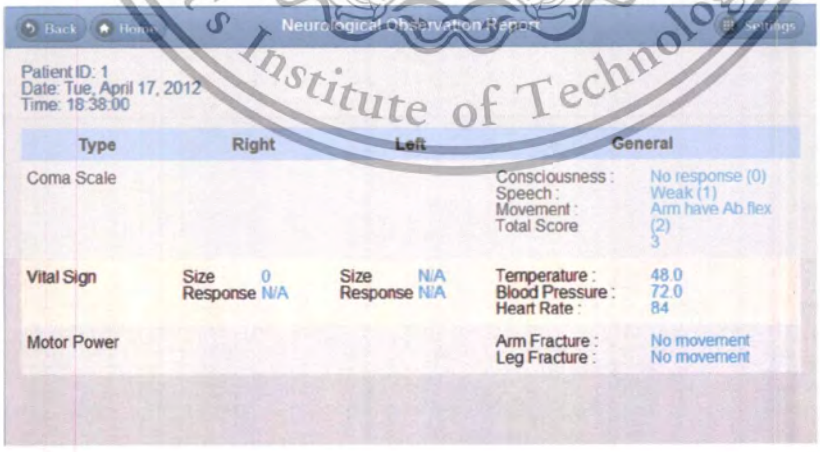


Figure A.11 View of neurological observation report

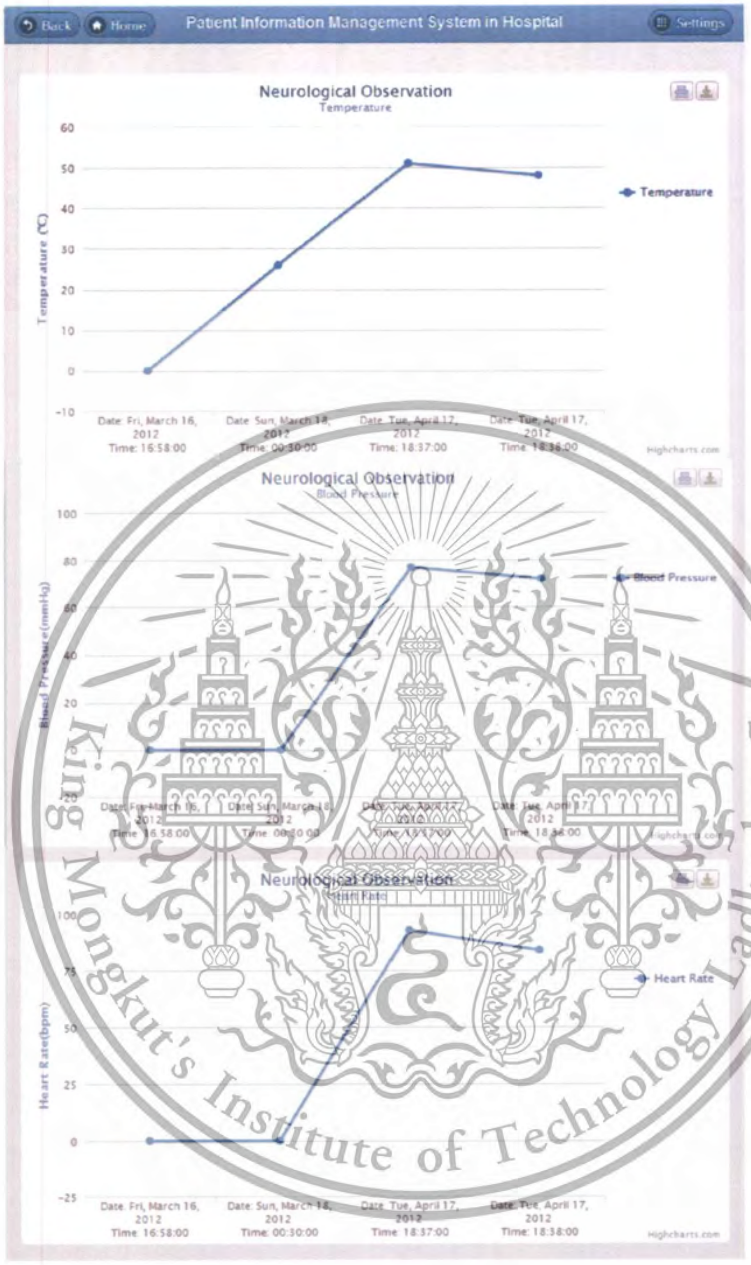


Figure A.12 shows a temperature graph, blood pressure graph, and heart rate graph of neurological observation

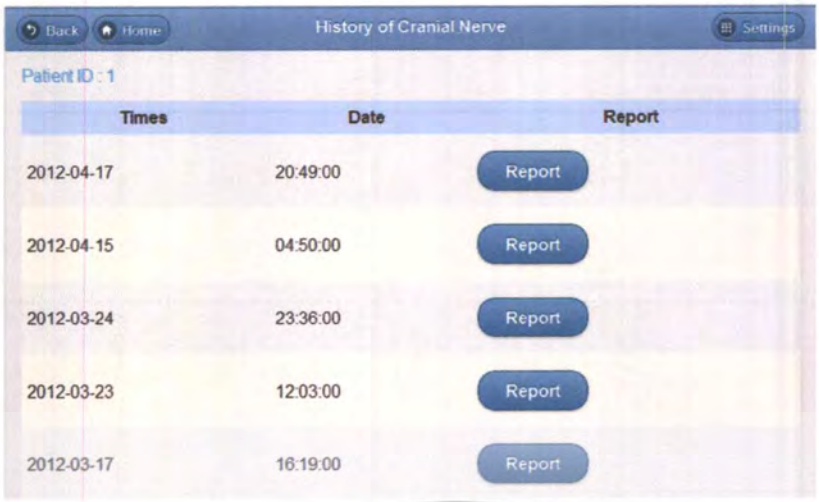


Figure A.13 Show the history of the cranial nerves report

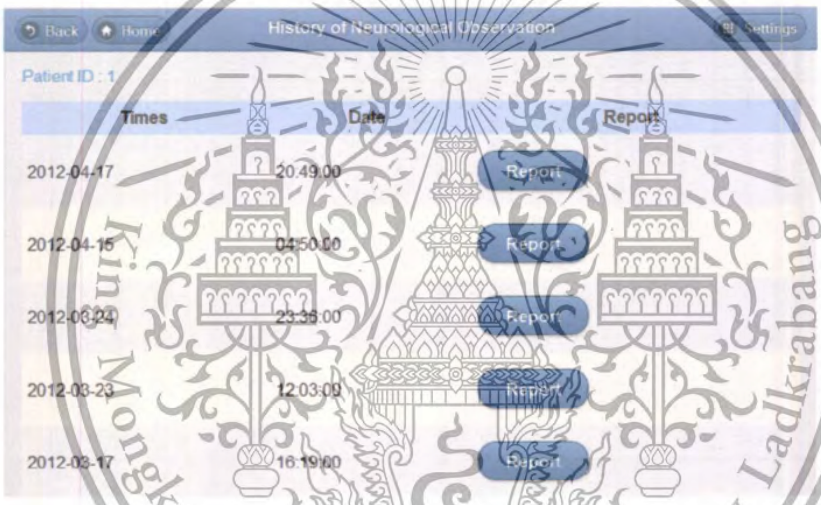


Figure A.14 Show the history of the neurological observation report

2.2  (Cranial Nerves Function) is the main function of the web application which used to record the symptoms twelve pairs of cranial nerves. The buttons on the man's face indicate the positions of the cranial nerves where the doctor can press to record the symptoms. The doctor needs to select the date and time before recording the cranial nerve's symptom. The doctor can press on 

button in order to back the main menu or 

button to view the doctor's profile, the patient's profile, and log out from the system.

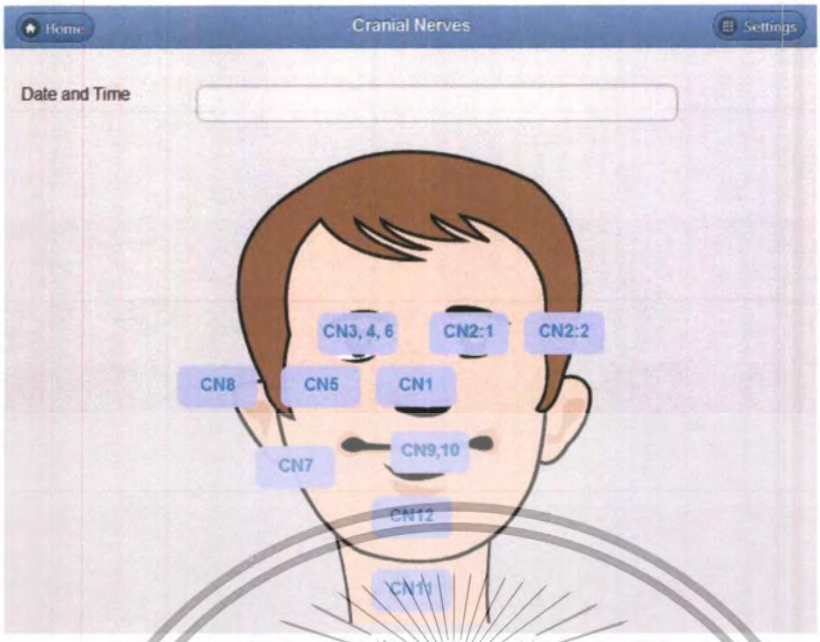


Figure A.15 Main page of cranial nerves function

1) CN 1 in Figure 4.15 is shown about the sense of which is divided into left and right sides. The doctor can select one of the provided choices. After finished recording the symptoms, the doctor needs to press on Back button in order to go back to the main cranial nerves and record the other cranial nerves symptoms. Home button is used to go back to the main menu, and Setting button to view the doctor's profile, the patient's profile or log out from the system.

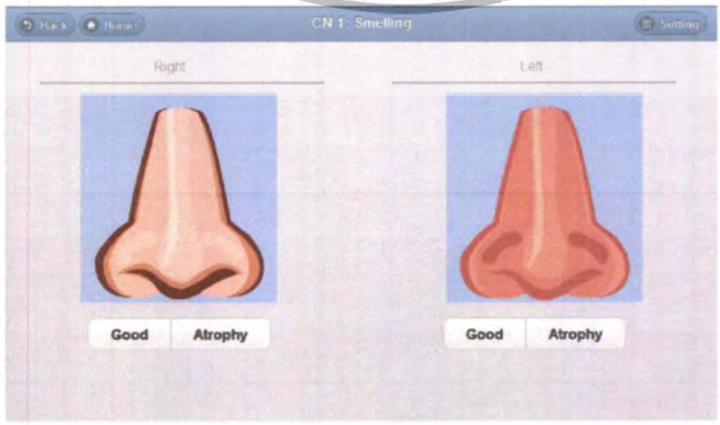


Figure A.16 CN 1: Smelling

2) CN 2 is the transmission of visual signals from the retina of the eyes to the brain. It consists of the two sections including visual acuity and visual field. In the visual acuity section, the doctor can slide the  slider bars to fill in the values. In the visual field section, the doctor can press on the quarter of the picture where the abnormal occurred.

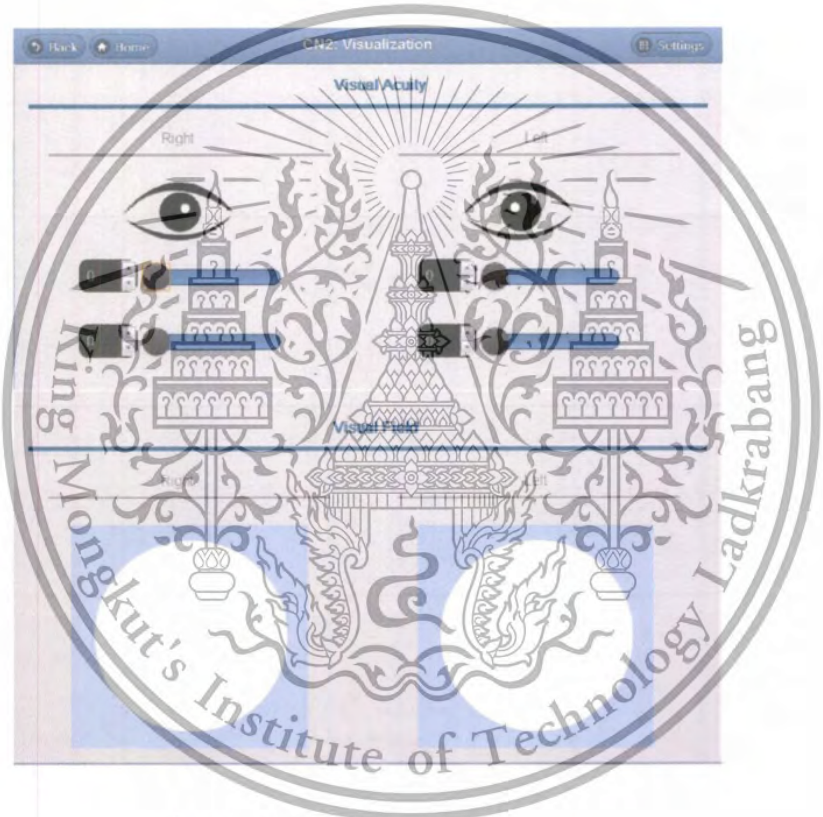


Figure A.17 CN 2:1 Visualization

3) CN 2:2 in Figure 4.17 is the same as visual field in CN 2:1 but it allows the doctor to draw the blind position of the eyes.

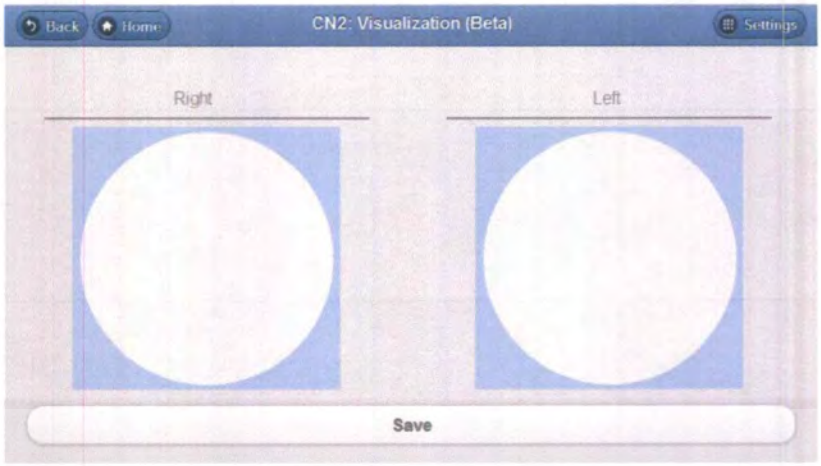


Figure A.18 CN 2:2 Visualization

4) CN 3, 4, 6 in Figure 4.18 is the performance of the eye movement in both sides. The navigation bar indicates the positions of eye movement on the left and the right sides. The doctor can slide the top slider bar to fill in the same values of both eyes or the slider bars on the right and the left sides to fill in the differences values of eye movement on each side.

Figure A.19 CN 3, 4, 6: EOM

5) CN 5 is the sensation of the face and innervates the muscles of mastication. In the sensory section, the doctor can select the abnormal position of the face. In the motor mastication, the doctor can choose the patient's symptom in both sides.

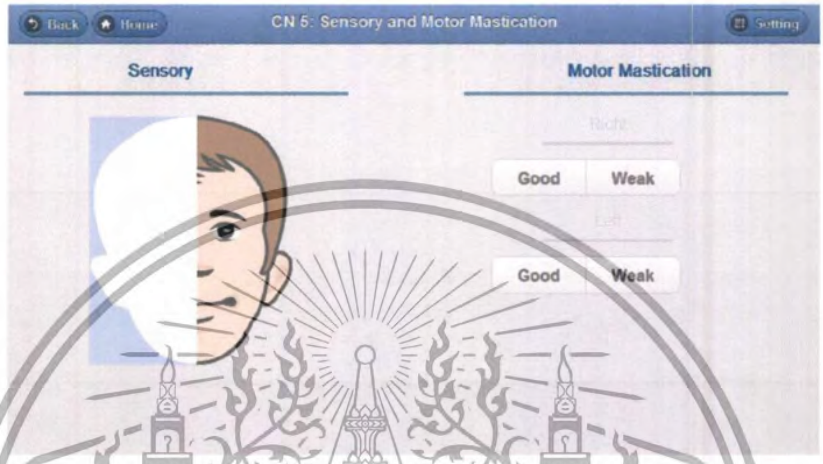


Figure A.20 CN 5: Sensory and motor mastication

6) CN 7 is the motor innervation to the muscles of facial expression. The doctor can choose the patient's symptom in both sides.

Figure A.21 CN 7: Facial (Brackman Scale)

7) CN 8 is about the senses sound, rotation, and gravity. The doctor can choose the patient's symptom in both sides.

Figure A.22 CN 8: Hearing

8) CN 9, 10 is the muscle control of voice, resonance, and the soft palate. It consists of four sections. The doctor can choose the symptom from the given choices.

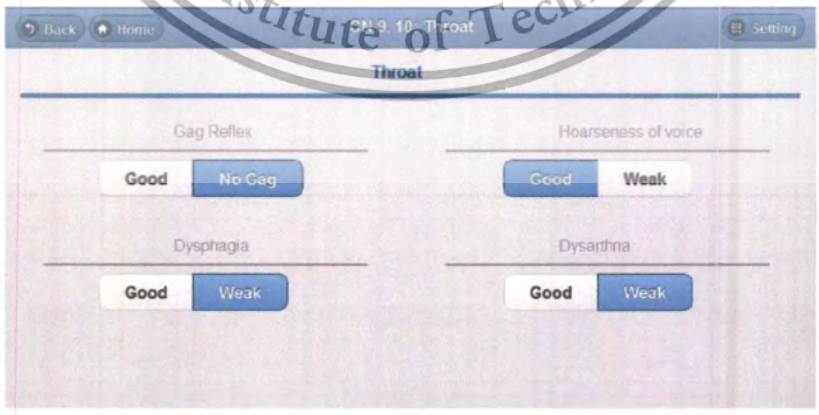


Figure A.23 CN 9, 10: Throat

9) CN 11 is about the controls of sternocleidomastoid and trapezius muscles. The doctor can select the symptom either good or weak from the given choices.

Figure A.24 CN 11: Sternocleidomastoid

10) CN 12 is about the motor innervation to the muscles of the tongue. The doctor can select either normal or atrophy symptom from the given choices.

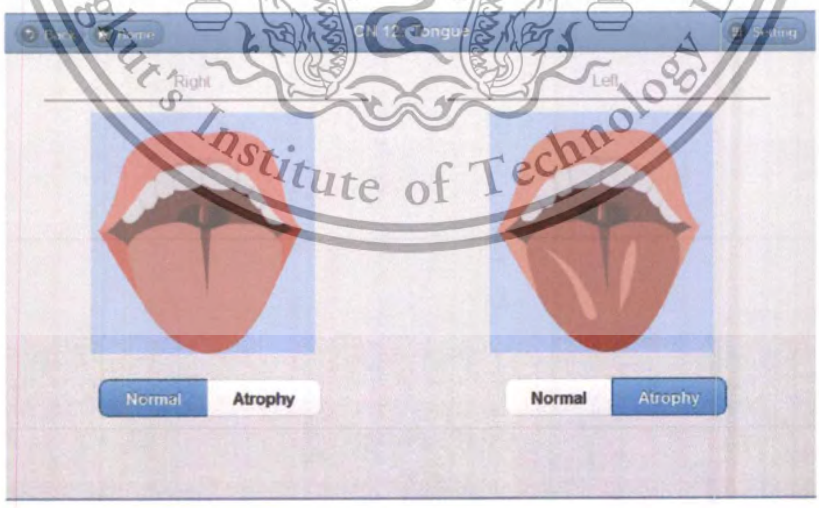


Figure A.25 CN 12: Tongue

2.3  (New Patient Function) is the form to insert a new patient's profile into the system.

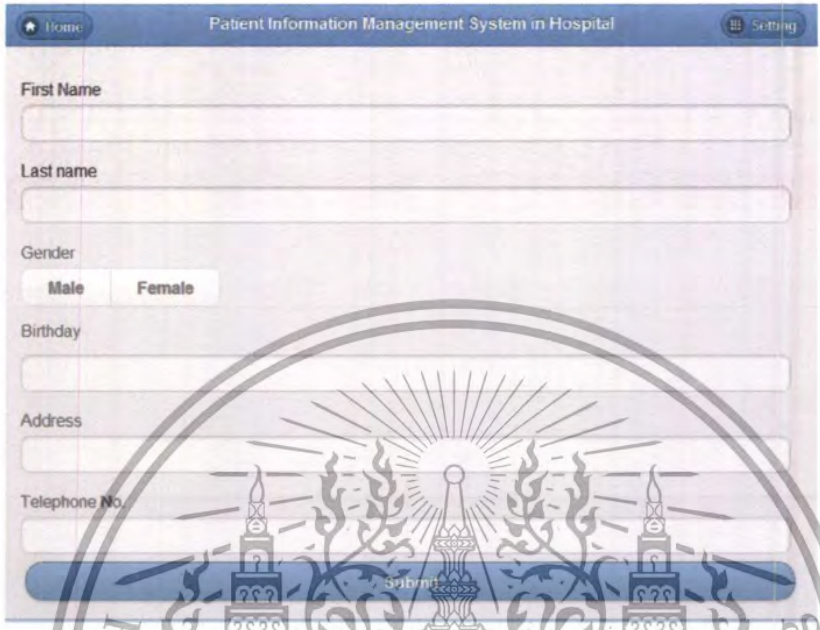


Figure A.26 Create the new patient's profile

2.4  (Neurological Observation Function) is the main function of the web application which used to record the neurological observation symptoms. It consists of three main functions including coma scale, vital signs, and motor power. The neurological observation page shows the three main functions of neurological observation. The doctor needs to select date and time before recording the symptoms on each function. The doctor can go back to the main menu by pressing on  button or  button to view the doctor's profile, the patient's profile or log out from the system.

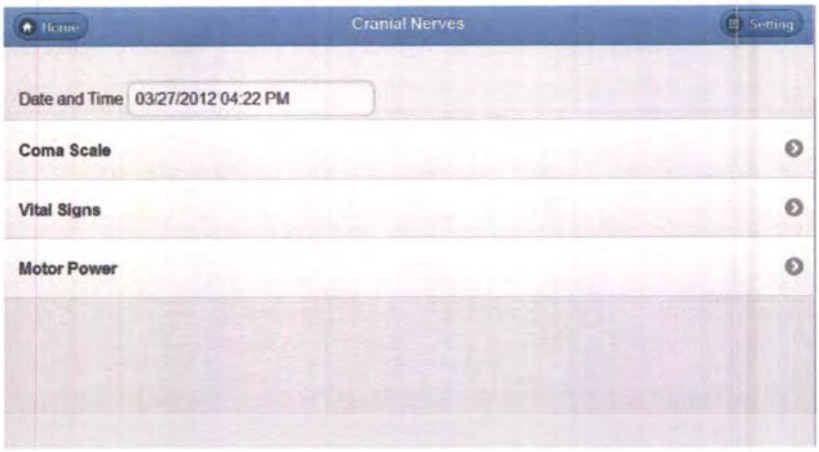


Figure A.27 Main page of neurological observation

1) Coma Scale is used to record the conscious state of a person for initial as well as subsequent assessment. The doctor can select the given choices of the patient's symptom. After finished recording the symptoms, the doctor needs to select on Back button in order to back the main neurological observation and record the other symptoms. Home button is used to back the main menu and Setting button to view the doctor's profile, the patient's profile or log out from the system.

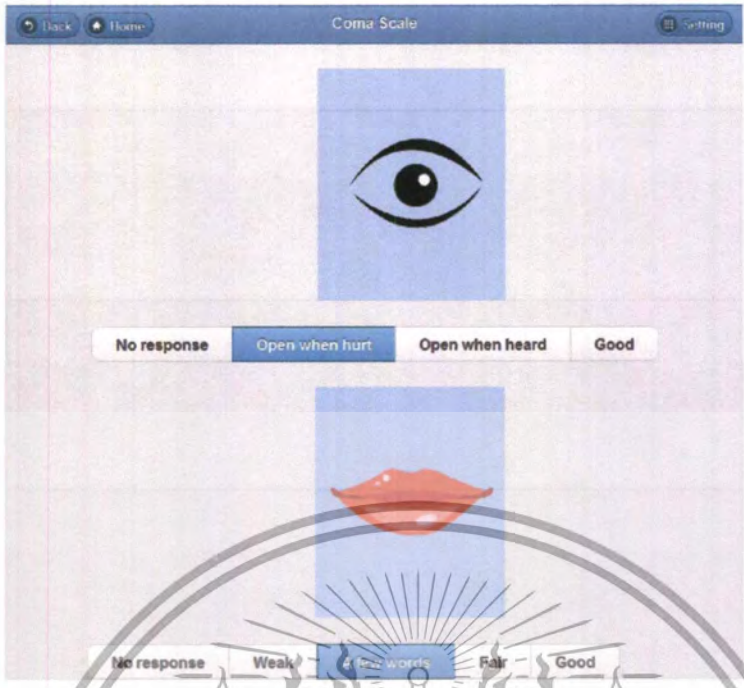


Figure A.28 Coma scale

2) Vital signs is used to record the body temperature, pulse rate (or heart rate), blood pressure, and respiratory rate. The doctor can slide the slider bars or select the choice to record the values.

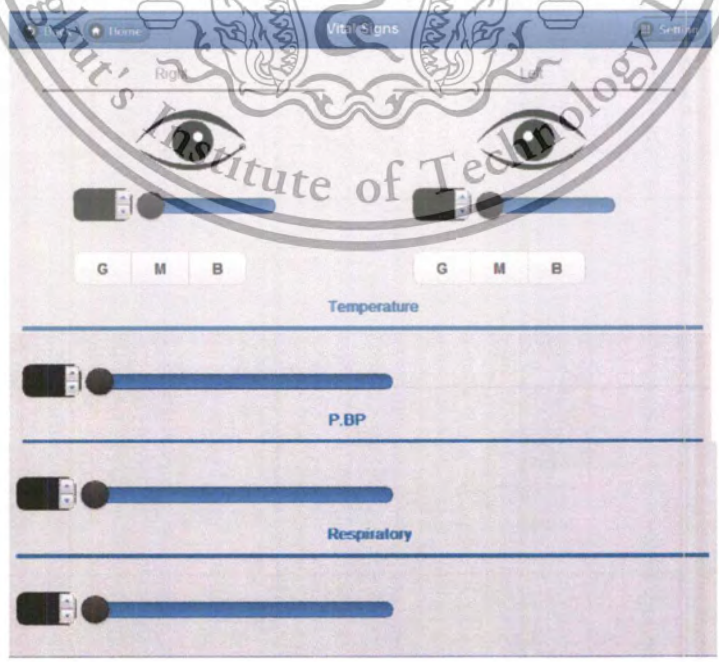


Figure A.29 Vital signs

3) Motor Power is used to record the movement of arms and legs. The doctor can select the patient's symptom from the given choices.

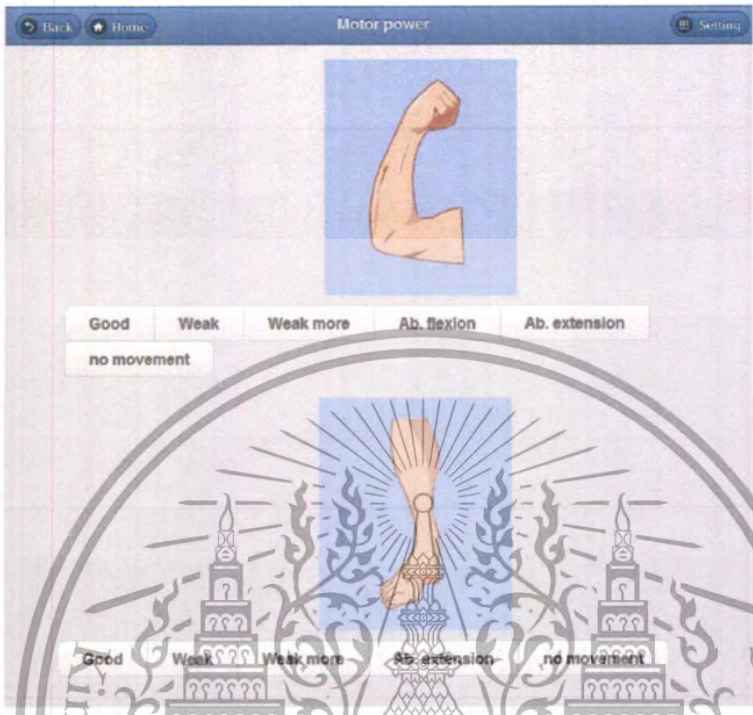


Figure A.30 Motor power

Appendix B.

Installing XAMPP on Windows

Installing XAMPP on Windows

1. **Download the software from**: http://www.apachefriends.org/en/xampp-windows.html#641

   Select the **Installer** option under the Basic Package. You may be taken to a page that presents you with a different download locations. Just click one of the download buttons, and then save the file to your desktop. Once downloaded, the installer works like most Windows installers.

2. **Double-click the .exe file you downloaded.**

   A window opens, asks you to select the language you would like to use.

3. **Choose a language from the menu, and then click OK.**

   A Setup Wizard window appears, ready to step you through the setup process.

4. **Click the Next button.**

   The installer suggests to put the application on your main drive at C:\XAMPP. You can install it anywhere.

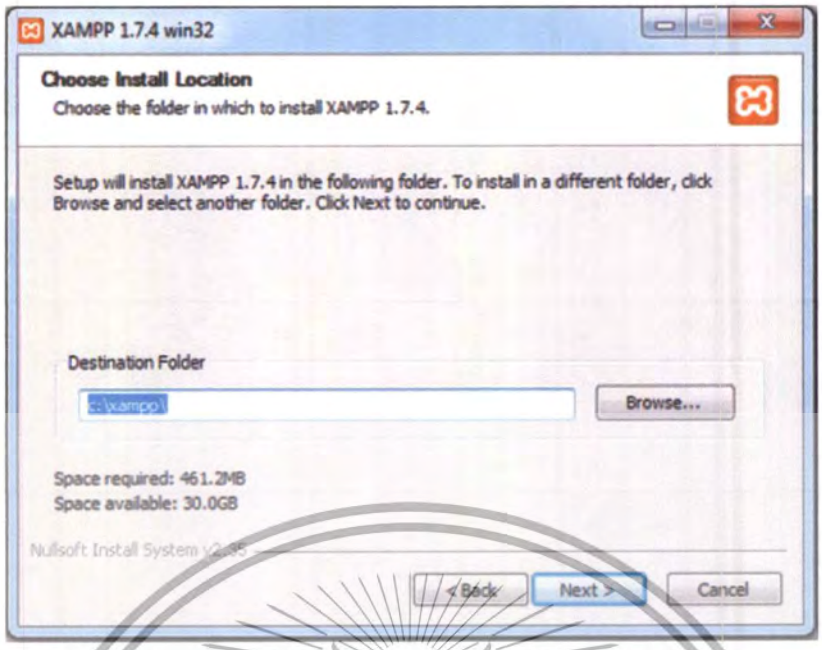


Figure B.1 Select the destination folder to install the application

5. **Click the Next button once again.**

The XAMPP Options window appears. In most cases, it is fine to leave all the window's checkboxes just as you see.

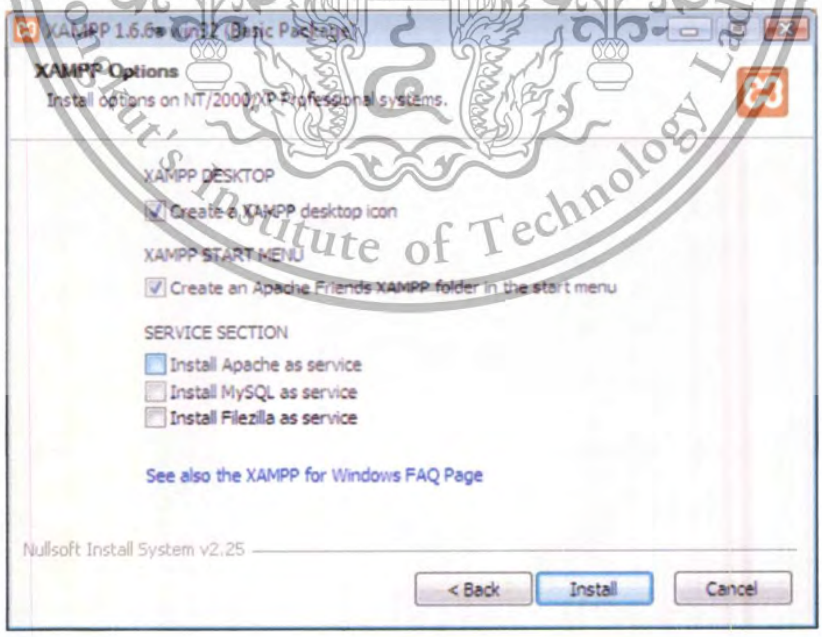


Figure B.2 Select the option to install

If you plan on doing a lot of development, you might want to check on the "Install Apache as service" and "Install MySQL as service" checkboxes. A service starts up every time you turn on your computer, so Apache, PHP, and MySQL are always running. However, if you won't be building database sites frequently, or you do not have a lot of RAM in your computer, do not turn on these boxes (you will just have to manually start the servers when you wish to build dynamic pages, using the XAMPP control panel).

6. **Click Install.**

The installer places all the files in your system. This process takes a while, since a lot of programs and files are being installed.

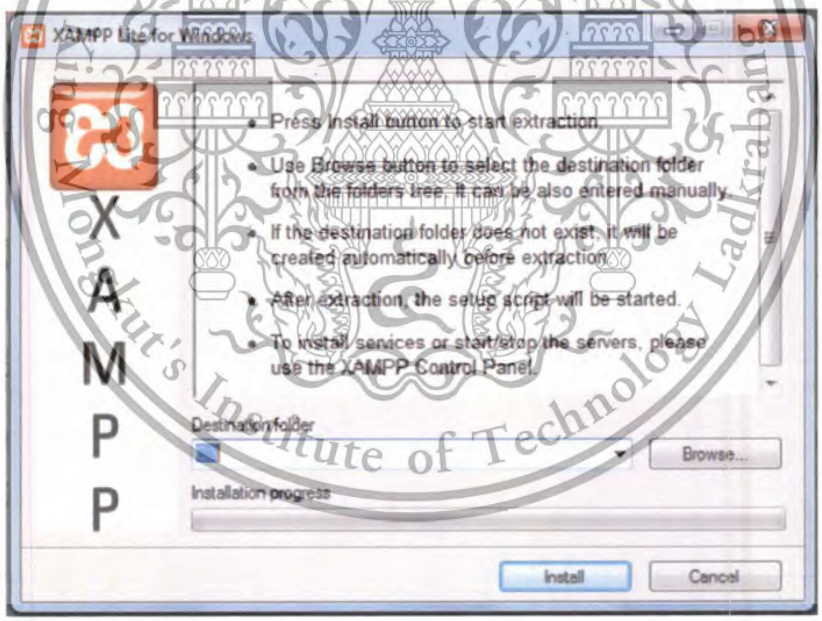


Figure B.3 The destination folder to install and installation progress bar

7. **Finally, click the Finish button.**

A window appears "congratulating" to you (way to double-click the installer program!), and asking whether you wish to start the XAMPP Control panel.

8. Click Yes, to open the XAMPP Control Panel.

The XAMPP Control Panel lets you start and stop the Apache Web server and MySQL database server.

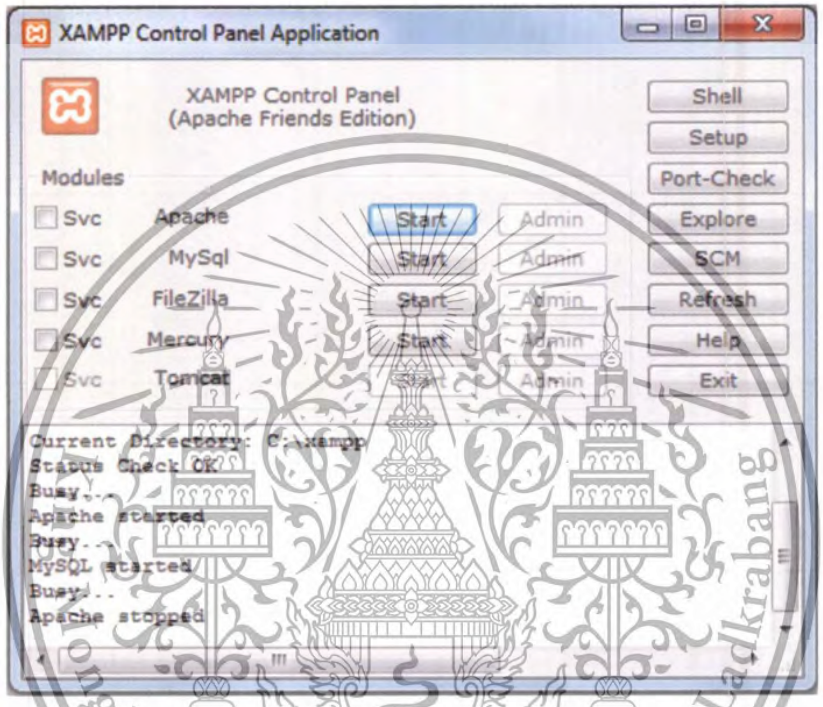


Figure B.4 The XAMPP control panel

In this figure, both Apache and MySQL are currently NOT running, as indicated by the word Start to the right of their names. Click the Start buttons to turn the servers on. You can open the Control Panel by clicking the XAMPP Control Panel shortcut on your desktop.

9. If the buttons on the right of Apache and MySQL say Start, click them to start the Web server and the MySQL database server.

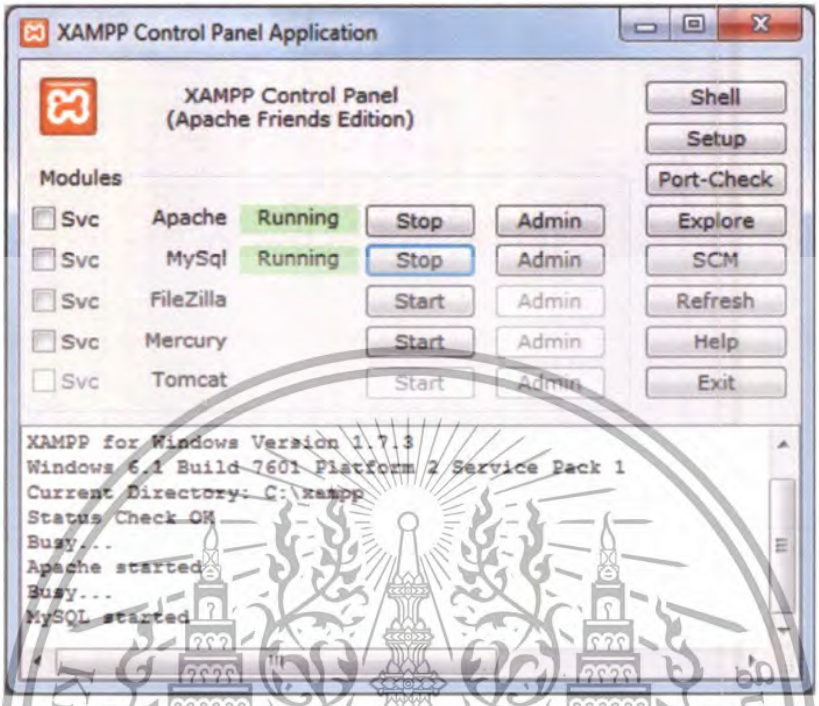


Figure B.5 The running of apache and MySQl on the XAMPP control panel

You probably get a Windows security alert about both MySQL and Apache; click the Unblock button in both cases. This action allows the two servers to run, and tells the Windows firewall protection service that everything is OK. If Apache and MySQL are already running, these buttons say Stop. (Clicking them turns off the Web server and MySQL.) Whenever you start Apache, PHP automatically starts as well. At this point, you should have a complete testing server running on your machine. You just need to make sure it is working.

10. To launch a Web browser in the Location bar type http://localhost/.

You encounter a page that lists a bunch of languages; click the language you prefer, and you are taken to a kind of Web-based control panel for XAMPP.

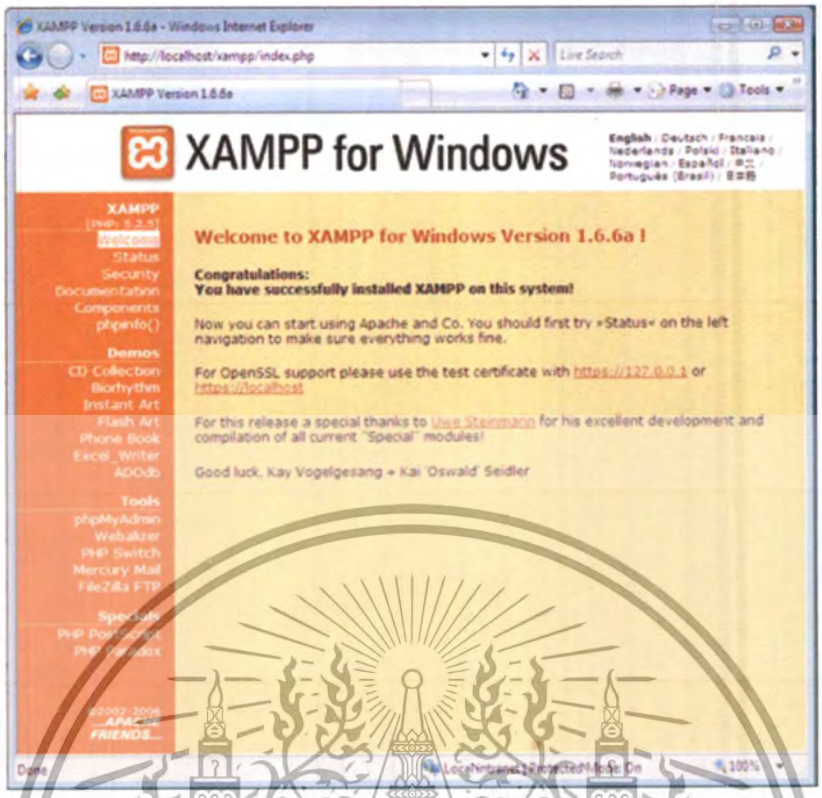


Figure B.6 Launch the XAMPP on web browser

Once you have installed XAMPP, you will see a shortcut called XAMPP Control Panel on your desktop. Double-click this icon to control the servers you have just installed—you can turn the servers off and on, as well as turn them into services (which launch each time you start up your computer).