

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**ระบบจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา**  
**MOBILE COMPUTING APPLICATION FOR HOSPITAL**

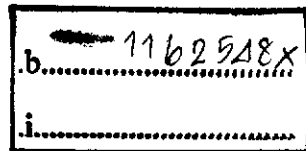
นายชยาคมน์ ภิญญศักดิ์  
นายวารินทร์ ศรีบุญเรือง

ร.ร.

ร. 194 ร

2548

เลขหมู่.....  
เลขทะเบียน..... 62454  
วัน,เดือน,ปี 18 ส.ค. 2549



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2548

**ระบบจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา**  
**MOBILE COMPUTING APPLICATION FOR HOSPITAL**

โดย

**นายชยากรณ์ ภิณญศักดิ์**  
**นายวารินทร์ ศรีบุญเรือง**

**อาจารย์ที่ปรึกษา**

**ผศ.ดร. ศักดิ์ชัย ทิพย์จักรนุรัตน์**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์**

**คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**พ.ศ.2548**

ปริญญาานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

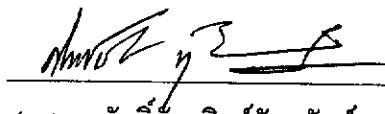
เรื่อง ระบบจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา

MOBILE COMPUTING APPLICATION FOR HOSPITAL

ผู้จัดทำ

1. นายชชาคมน์ ภิญญศักดิ์ รหัสนักศึกษา 46015342

2. นายวารินทร์ ศรีบุญเรือง รหัสนักศึกษา 46015372

  
(ผศ.ดร.ศักดิ์ชัย ทิพย์จักรรัตน์)

อาจารย์ที่ปรึกษา

## ระบบจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา

นายชยากรณ์ ภิญญศักดิ์ 46015342  
นายวารินทร์ ศรีบุญเรือง 46015372  
ผศ.ดร. ศักดิ์ชัย ทิพย์จักษ์รัตน อาจารย์ที่ปรึกษา  
ปีการศึกษา 2548

### บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้เสนอวิธีการพัฒนาแอปพลิเคชัน สำหรับโรงพยาบาลเพื่อช่วยอำนวยความสะดวกในการรักษาผู้ป่วยของทางโรงพยาบาลแผนกผู้ป่วยใน โดยจากที่แต่ก่อนเราใช้การทำงานที่เรียกว่ายังเป็นแบบ Manual อยู่ มาเป็นการทำงานแบบดิจิทัลที่ทันสมัยยิ่งขึ้น โดยระบบของเราจะแบ่งการทำงานออกเป็น 2 ส่วนด้วยกัน ซึ่งประกอบไปด้วย ส่วนแรกเป็นการพัฒนาแอปพลิเคชันบนอุปกรณ์ประมวลผลแบบพกพาที่เรียกว่า Pocket PC โดยใช้ C#.NET เป็นภาษาในการพัฒนา ร่วมกับหลักการของ Web Service ซึ่งใช้งานได้ง่าย และในส่วนที่สองคือ Web Application ใช้ภาษา PHP ในการพัฒนา โดยทั้งสองส่วนจะอ้างอิงฐานข้อมูลเดียวกันคือจากเซิร์ฟเวอร์ (Server) ที่เขียนด้วยภาษา SQL โดย Pocket PC สามารถที่จะดึงข้อมูลคนไข้ เช่น เพียงระบุหมายเลขคนไข้บนตัว Pocket PC ก็สามารถรู้ประวัติการรักษา อาการ ข้อมูลการเจ็บป่วย และสามารถส่งยาผ่านทางหน้าจอเพื่อแจ้งไปยังคอมพิวเตอร์ที่ดูแลโดยพยาบาล เพื่อให้พยาบาลนำจ่ายยาตามที่หมอได้สั่งไว้ตามลำดับ โดยสามารถพกพาตัว Pocket PC ไปที่ไหนก็ได้ที่สามารถรับสัญญาณ Wireless ได้ ซึ่งการทำงานในลักษณะนี้จะช่วยให้การเข้าถึงการรักษาได้เร็ว และยังช่วยลดเวลา และประหยัดเอกสารในการบันทึกผล ส่งผลให้ท้ายที่สุดระบบสามารถที่จะสร้างคุณภาพในการรักษาที่ดีขึ้นกว่าเดิม

# MOBILE COMPUTING APPLICATION FOR HOSPITAL

Mr. Chayakom Pinyasak 46015342

Mr. Warin Sriboonruang 46015372

Asst.Prof.Dr. Sakchai Thipchaksurat Advisor

Academic Year 2005

## ABSTRACT

This project proposes an application development called Mobile Computing Application for Hospital. It provides a convenient for treat of in-patient field in hospital. In early the hospital work by using manual (hand writing) but now it is digital application. The system is divided into two parts. In the first part, we develop the application on the pocket PC by using C#.NET language to work with the Web Service. The second part is Web Application developed by using PHP language. Both of them use SQL language to produce the database system. Pocket PC used for retrieving patients information such as, physicians or medicine' order at anywhere or anytime. It enable doctors to provide point-of care by reducing time-consuming, redundant paperwork. Finally, the system can improve the quality of care.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร. ศักดิ์ชัย ทัพย์จักรภูรัตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาานิพนธ์ ข้าพเจ้ารู้สึกทราบซึ่งในความอนุเคราะห์จาก ท่านอาจารย์และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือ ในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแด่ผู้มีพระคุณทุก ท่าน

นายชยากรณ์ ภิญญศักดิ์  
นายวารินทร์ ศรีบุญเรือง

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ส่วนประกอบของปริญาานิพนธ์.....	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการวิจัย และระบบโอเอฟดีเอ็ม.....	4
2.1 เทคนิคการมัลติเพล็กซ์.....	4
2.1.1 การมัลติเพล็กซ์แบบแบ่งช่วงความถี่.....	4
2.1.2 การมัลติเพล็กซ์แบบแบ่งช่วงเวลา.....	5
2.1.3 การมัลติเพล็กซ์แบบแบ่งรหัส.....	5
2.1.4 การมัลติเพล็กซ์แบบโอเอฟดีเอ็ม.....	6
2.2 คุณลักษณะของช่องสัญญาณแบบมัลติพาธเฟดดิ้ง.....	6
2.3 พื้นฐานของระบบโอเอฟดีเอ็ม.....	15
2.3.1 คุณสมบัติของการออโธโกนอล.....	16
2.3.2 คุณสมบัติของการออโธโกนอลในแกนความถี่.....	20
2.3.3 เครื่องรับส่งโอเอฟดีเอ็ม.....	22
2.3.4 การมอดูเลตคลื่นพาย้อย.....	23
2.3.5 การแปลงสัญญาณในแกนความถี่ให้อยู่ในแกนเวลา.....	24

## สารบัญ (ต่อ)

	หน้า
2.4 เปรียบเทียบระหว่าง Web Application และ Web Services .....	26
<b>บทที่ 3 การออกแบบโปรแกรม.....</b>	<b>28</b>
3.1 ความต้องการของระบบ.....	28
3.2 อุปกรณ์ที่ใช้ในการพัฒนาโครงการ.....	28
3.3 การออกแบบตารางฐานข้อมูล.....	29
3.4 โปรแกรมบน Pocket PC (Mobile Computing Application).....	43
3.4.1 ขอบเขตการทำงานของโปรแกรม.....	43
3.4.2 การออกแบบโปรแกรม.....	43
3.5 โปรแกรม Web Application.....	46
3.5.1 ขอบเขตการทำงานของโปรแกรม .....	46
3.5.2 การออกแบบโปรแกรม.....	46
3.5.2.1 ภาพรวมของระบบ Web Application.....	47
3.5.2.2 การรวมระบบย่อยให้เป็นเว็บไซต์.....	52
<b>บทที่ 4 การทดลองและผลการทดลอง.....</b>	<b>61</b>
4.1 สรุปความสามารถที่โปรแกรมทำได้.....	61
4.1.1 โปรแกรมบน Pocket PC (Mobile Computing Application).....	61
4.1.2 โปรแกรม Web Application.....	61
4.2 ผลการทำงานของโปรแกรม.....	61
4.2.1 โปรแกรมบน Pocket PC (Mobile Computing Application).....	61
4.2.2 โปรแกรม Web Application.....	68
<b>บทที่ 5 บทวิจารณ์และสรุป.....</b>	<b>80</b>
5.1 บทสรุป.....	80
5.2 วิจารณ์สิ่งที่ได้จากโครงการ.....	80
5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....	81
5.4 แนวทางการพัฒนาต่อ.....	81
<b>บรรณานุกรม.....</b>	<b>82</b>

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงการเปรียบเทียบระหว่าง Web Services กับ Web Application .....	27

# สารบัญรูป

รูปที่	หน้า
2.1 แสดงส่วนประกอบของ .Net Framework.....	5
2.2 แสดงการเชื่อมต่อ Database Server กับ Web Server .....	9
2.3 XML message using SOAP .....	15
2.4 โครงสร้างของเอกสาร SOAP.....	16
2.5 แสดง Web Services with SOAP, UDDI และ WSDL.....	24
3.1 แสดงความสัมพันธ์ของฐานข้อมูล.....	29
3.2 แสดงขั้นตอนการทำงานของ PocketPC.....	45
3.3 แสดงฟังก์ชันการติดต่อกับฐานข้อมูลทั้งหมด ทั้งการดึงข้อมูลและการบันทึกข้อมูล.....	46
3.4 แสดงโครงสร้างของเว็บไซต์.....	47
3.5 แสดงขั้นตอนการตรวจสอบผู้ใช้.....	48
3.6 แสดงการตรวจสอบการเป็นสมาชิกทั่วไป.....	49
3.7 แสดงการทำงานของผู้ใช้ที่เป็นพนักงาน โรงพยาบาล.....	49
3.8 แสดงการทำงานในส่วนของผู้ดูแลระบบ.....	50
3.9 แสดงขั้นตอนการสมัครสมาชิกใหม่.....	51
3.10 แสดงการทำงานของสมาชิกทั่วไป.....	51
3.11 แสดงการทำงานกรณีลืมหีสผ่าน.....	52
3.12 แสดงเมนูในส่วนของหน้าแรกของเว็บไซต์.....	53
3.13 แสดงหน้าลืออกอื่น.....	53
3.14 แสดงหน้าตาของเว็บบอร์ด.....	54
3.15 แสดงหน้าตาในส่วนเมนูของพนักงาน.....	55
3.16 แสดงเมนูหน้าค้นหาข้อมูล.....	56
3.17 แสดงฟอร์มในการค้นหาข้อมูลพนักงาน.....	56
3.18 แสดงผลลัพธ์ในการค้นหาข้อมูลพนักงาน.....	57
3.19 แสดงหน้าตาในส่วนของข่าวประชาสัมพันธ์.....	58
3.20 แสดงหน้าตาในส่วนของการเพิ่มบุคลากรใหม่.....	58
3.21 แสดงรายชื่อบุคลากรที่ผู้ดูแลระบบสามารถแก้ไขหรือลบได้.....	59
3.22 แสดงหน้าตาในส่วนของการแก้ไขรายละเอียดบุคลากร.....	59
4.1 แสดงหน้า Login ของ Pocket PC .....	61
4.2 แสดงหน้า Login ในกรณีที่เกิดการผิดพลาดในการ Login .....	62

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 หน้า Main ก่อนที่จะทำการป้อนหมายเลขห้อง.....	62
4.4 แสดงหน้า Main เมื่อทำการป้อนหมายเลขห้องเรียบร้อยแล้ว และกดปุ่ม OK ก็จะ ได้ข้อมูลการเจ็บป่วยของผู้ป่วย.....	63
4.5 แสดงข้อความเตือนกรณีที่ ไม่ได้กรอกหมายเลขห้อง แล้วคลิก OK หรือหมายเลขห้องไม่มีอยู่จริง.....	63
4.6 แสดงข้อความเตือนกรณีที่ ไม่ได้กรอกหมายเลขห้อง แล้วคลิกปุ่ม History หรือ Biography...	64
4.7 เมื่อกดปุ่ม History ที่หน้า Main ก็จะแสดงหน้าข้อมูลการรักษาของผู้ป่วย.....	64
4.8 เมื่อกดปุ่ม Biography ที่หน้า Main ก็จะแสดงหน้าข้อมูลส่วนตัวของผู้ป่วย.....	65
4.9 หน้าแรกบันทึก วันที่ตรวจรักษา ,เวลาที่ตรวจ ,อุณหภูมิร่างกาย ,ชีพจร ,ความดันโลหิต.....	65
4.10 หน้าที่ 2 บันทึกการหายใจ ,อัตราการบีบของหัวใจ ,อัตราการพองของหัวใจ , ตรวจอุจจาระ ,ตรวจปัสสาวะ.....	66
4.11 หน้าที่ 3 บันทึกอาการของผู้ป่วย ,คำวินิจฉัย ,ของที่ห้ามรับประทาน.....	66
4.12 หน้าที่ 4 บันทึกการสั่งเวชภัณฑ์ทางการแพทย์.....	67
4.13 หน้าที่ 5 ใส่รหัสของผู้ที่ทำการตรวจรักษา.....	67
4.14 หน้าโฮมเพจ.....	68
4.15 หน้า User Login.....	69
4.16 หน้าสำหรับตอบคำถามเพื่อใช้ในการเปลี่ยนรหัส.....	69
4.17 หน้าสำหรับเปลี่ยนรหัสผ่านกรณีลืมรหัสผ่าน.....	70
4.18 แสดงความผิดพลาดกรณีความยาวของรหัสผ่าน ไม่อยู่ในช่วงที่กำหนด.....	70
4.19 แสดงความผิดพลาดกรณีรหัสผ่าน 2 ช่องไม่ตรงกัน.....	70
4.20 แสดงหน้าลงทะเบียนสมัครสมาชิกทั่วไป.....	71
4.21 แสดงผลลัพธ์การลงทะเบียนที่เสร็จสมบูรณ์.....	71
4.22 หน้าแสดงกระทู้ทั้งหมดของเว็บบอร์ด.....	71
4.23 หน้าแสดงการตอบกระทู้.....	72
4.24 หน้าฟอร์มตั้งกระทู้ใหม่.....	72
4.25 หน้าหลักของพนักงาน โรงพยาบาล.....	73
4.26 หน้าค้นหาข้อมูล.....	73
4.27 หน้าป้อนคำที่ต้องการค้นหา.....	74
4.28 ผลลัพธ์ที่แสดงออกมาจากการค้นหา.....	74

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.29 หน้าแสดงข้อความใหม่ .....	74
4.30 หน้าข่าวประชาสัมพันธ์.....	75
4.31 หน้าล็อกอินของข่าวประชาสัมพันธ์.....	75
4.32 สร้างหัวข้อข่าวใหม่.....	75
4.33 แสดงการเพิ่มพนักงานใหม่.....	76
4.34 แสดงการแจ้งผลลัพธ์หลังจากที่เพิ่มข้อมูลเรียบร้อยแล้ว.....	76
4.35 แสดงรายชื่อของพนักงานที่มีในฐานข้อมูล.....	77
4.36 แสดงหน้าฟอร์มหลังจากที่ผู้ดูแลระบบทำการเลือกรายชื่อ.....	77
4.37 แสดงหน้าฟอร์มการลบรายชื่อพนักงาน.....	78
4.38 แสดงหน้าจัดการเกี่ยวกับผู้ใช้งานระบบ.....	78
4.39 แสดงการแก้ไขข้อมูลผู้ใช้ของผู้ดูแลระบบ.....	79

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

ปัจจุบันเป็นยุคแห่งข้อมูลข่าวสารที่มีการเปลี่ยนแปลงเกี่ยวกับเทคโนโลยีใหม่ๆ มากมาย ส่งผลให้กิจการต่างๆ ต้องนำเทคโนโลยีเหล่านั้นมาใช้ภายในกิจการให้มากขึ้นเพื่ออำนวยความสะดวกให้กับลูกค้าและพนักงาน บางครั้งในการดำเนินกิจการก็อาจมีปัญหาในการจัดการได้เช่นกัน อาทิเช่น การบริหารงานในระบบโรงพยาบาลแบบเก่า การบันทึกประวัติการรักษา การส่งยา จะอาศัยการบันทึกโดยการเขียนเป็นส่วนใหญ่ ซึ่งบางที่ผู้บันทึกเองอาจบันทึกข้อมูลที่ผิดพลาด ทำให้ข้อมูลไม่ตรงกับความเป็นจริงแล้วยังทำให้เสียเวลาในการทำงานอีกด้วย

เหตุผลของการสร้างระบบจัดการ โรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพาคือ แก้ไขปัญหาของระบบบริหารแบบเดิม โดยสามารถดึงข้อมูลประวัติคนไข้ ข้อมูลการรักษา และสิ่งที่เกี่ยวข้อง เพื่อสร้างการบริการ รวมไปถึงให้ความสะดวกและรวดเร็วแก่ผู้รับบริการที่เข้ามา รับการรักษา และต่อแพทย์ผู้ให้การรักษา ซึ่งการวางแผนการดำเนินงานจะต้องอาศัยข้อมูลข่าวสารเหล่านี้

### 1.2 วัตถุประสงค์ของโครงการ

ปริญญานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาการพัฒนา Application บนอุปกรณ์ประมวลผลแบบพกพา และ Web Application ทำงานร่วมกับฐานข้อมูล ซึ่งกำลังเป็นที่นิยมในปัจจุบันซึ่งเป็นยุคของข้อมูลข่าวสาร รวมไปถึงสามารถนำไปโปรแกรมที่ได้พัฒนานี้ไปใช้งานกับหน่วยงานของโรงพยาบาลได้จริง เพื่ออำนวยความสะดวกและจัดการข้อมูลได้อย่างถูกต้องและมีประสิทธิภาพ

### 1.3 ขอบเขตของโครงการ

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอใน 2 ส่วนคือ ส่วนแรกเป็นการสร้าง Application บนพ็อกเก็ตพีซี (Pocket PC) ที่พัฒนาด้วยเทคโนโลยี .NET framework โดยใช้ภาษา C# เพื่อใช้ในส่วน of แผนกผู้ป่วยใน ภายในโรงพยาบาล โดยสามารถที่จะเพิ่ม ดึงข้อมูลที่เกี่ยวข้องกับการรักษาได้ รวมไปถึงการทำงานที่สอดคล้องกันกับส่วนที่สอง คือส่วนของการสร้าง Web Application ที่ใช้ภาษา PHP โดยในส่วนนี้จะมีลักษณะการทำงานบนเว็บเพจ โดยการทำงานจะแยกออกเป็นสอง

ส่วนคือ ส่วนของผู้ดูแลระบบ และส่วนของผู้ใช้ที่เป็นผู้ใช้ทั่วไป และที่เป็นพนักงานของโรงพยาบาล โดยในแต่ละส่วนจะสามารถเข้าถึงข้อมูลได้ต่างกัน ส่วนฟังก์ชันการทำงานส่วนใหญ่จะเน้นไปในเรื่องของ การแสดงข้อมูลตามที่ใช้ต้องการค้นหา และรองรับคำสั่งจากตัว Pocket PC ถ้าจะกล่าวโดยรวมแล้วทั้งในส่วนของ Mobile Application และ Web Application จะทำงานภายใต้ฐานข้อมูลร่วมกัน เพื่อให้ได้ข้อมูลที่ถูกต้องและตรงตามความต้องการ

#### 1.4 วิธีการดำเนินการ

1. ศึกษาเกี่ยวกับระบบงานของโรงพยาบาล โดยพุ่งประเด็นไปในส่วนของแผนกผู้ป่วยในเป็นหลัก เพื่อนำมาใช้ในการสร้างฐานข้อมูล
2. ศึกษาการสร้างโปรแกรมด้วยภาษา C#.NET รวมไปถึงการ Simulation ด้วย Pocket PC Emulator ก่อนที่จะนำไปใช้จริง
3. ศึกษาการสร้าง Web Application ด้วยภาษา PHP และ MySQL
4. จัดหาวัสดุอุปกรณ์ที่จำเป็นต้องใช้ในการพัฒนา
5. วิเคราะห์ และออกแบบระบบ
6. พัฒนาโปรแกรมที่ใช้ทั้งในส่วนของ Pocket PC และ Web Application รวมทั้งโปรแกรมที่ใช้ในการติดต่อระหว่าง 2 Application
7. วิเคราะห์ผลของระบบที่ได้ทำการพัฒนาขึ้น และแก้ไขส่วนที่ผิดพลาดเพื่อให้สามารถใช้งานได้อย่างถูกต้องที่สุด

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการทำงานของโรงพยาบาลแผนกผู้ป่วยใน
2. ได้รับความรู้ความเข้าใจเกี่ยวกับการพัฒนา Application บน Pocket PC และ Web Application
3. ระบบการจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพาที่สามารถค้นหา ปรับปรุง แก้ไขข้อมูล ได้ถูกต้องและใช้งานง่าย

#### 1.6 ส่วนประกอบของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการทำโครงการ

บทที่ 3 กล่าวถึงชิ้นงานของโครงการนี้ ส่วนที่ได้พัฒนาขึ้น การทำงานของระบบหรือชิ้นงานบรรยายโดยละเอียด

บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง การหาค่าสมรรถนะของระบบ การวัดประสิทธิภาพของระบบ พารามิเตอร์ที่ใช้และผลที่ได้จากการจำลองระบบ ผลการทดลองหรือผลการทำงานทั้งหมด

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

## บทที่ 2

# ทฤษฎีพื้นฐานที่ใช้ในโครงการงาน

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการวิจัย และพื้นฐานของระบบการจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา ซึ่งเนื้อหาในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบด้วย .NET Framework ทฤษฎีที่เกี่ยวข้องกับ Web Application และอื่นๆ ที่เกี่ยวข้องกับระบบ ซึ่งเนื้อหาทั้งหมดในบทนี้จำเป็นสำหรับการศึกษา เพื่อที่จะนำไปพัฒนาระบบต่อไปได้

### 2.1 สถาปัตยกรรม .NET Framework

#### 2.1.1 .Net Framework คืออะไร

.Net Framework คือ โครงสร้างการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อให้อำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ ที่ใช้งานในระบบเครือข่าย (Internet, Intranet, Mobile Devices, ฯลฯ) Bill Gates และ Steve Ballmer ได้บรรยายสรุปวิสัยทัศน์ ที่เกี่ยวกับ .Net เอาไว้ 3 ข้อหลัก ๆ ได้แก่

1. การพัฒนาโปรแกรมในรูปแบบของ Web Service จะเป็นหัวใจหลักในการขับเคลื่อนโปรแกรมต่าง ๆ ที่ใช้งานบน Internet. Web Service จะช่วยให้การติดต่อสื่อสารระหว่าง application บน Internet นั้นง่ายขึ้น และเป็นระบบมากยิ่งขึ้น

2. Web Service ขึ้นพื้นฐานเช่น การตรวจสอบ user ที่ log in เข้าสู่ระบบ จะถูกพัฒนาให้เป็นมาตรฐาน และสามารถนำไปใช้ได้ทั่วไปบน Internet

3. PC (desktop, notebook) และ Mobile Device ที่ต่อเชื่อมกับ Internet ได้ เช่น PDA และ โทรศัพท์มือถือ จะมีบทบาทและประโยชน์มากขึ้นไปอีก เมื่อสามารถติดต่อใช้งานโปรแกรมต่าง ๆ บน Internet ได้

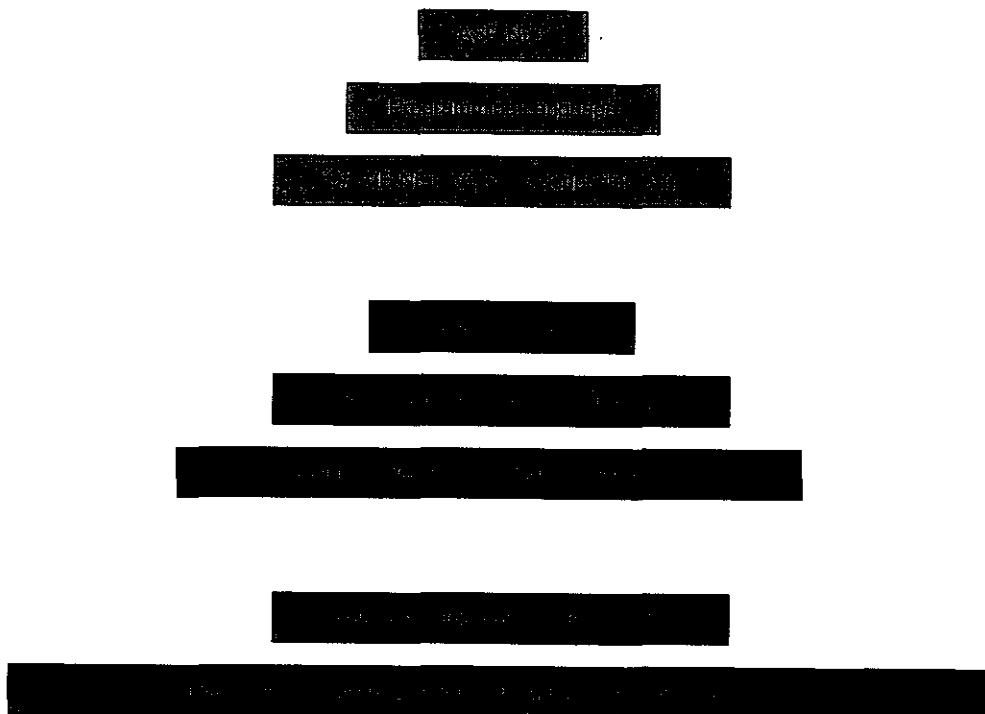
ยกตัวอย่างง่าย ๆ ว่าหากเรามีระบบตรวจสอบลายนิ้วมือ แล้วเราเอานิ้วไปประทับไปที่หน้าจอในรถแท็กซี่ Web Service ก็เกิดขึ้นในทันที เริ่มจาก ระบบตรวจสอบลายนิ้วมือซึ่งให้บริการตรวจสอบว่า ลายนิ้วมือของผู้โดยสารนั้นเป็นใคร (User-Authentication Web Service) พอทราบแล้วว่าเป็นใคร ระบบก็จะส่งข้อมูลไปยังบริษัทที่ให้บริการรถแท็กซี่ ซึ่งจะทำการคิดคำนวณค่าบริการ และส่ง request ไปยังธนาคารที่ผู้โดยสารมีบัญชีอยู่ เพื่อทำการหักค่าใช้จ่ายจากบัญชีของผู้โดยสาร ไปเข้าบัญชีของบริษัทรถแท็กซี่ ส่วนที่พนักงานขับรถจะได้จากการบริการ ก็จะถูกบันทึกไว้ในระบบข้อมูลพนักงานคนนั้น ๆ จะเห็นได้ว่าจากสถานการณ์ที่ได้ยกตัวอย่างไปนี้ จำเป็นที่จะต้องใช้ข้อมูล และการประมวลผลจากเครื่องคอมพิวเตอร์ และโปรแกรมต่าง ๆ อยู่หลายที่ ซึ่งแต่ละโปรแกรมนั้นก็อาจจะทำงานอยู่บนระบบที่แตกต่างกันไป

เช่น Windows, Linux, Mainframe, ฯลฯ ภาษาคอมพิวเตอร์ที่ใช้พัฒนาโปรแกรมเหล่านั้นก็อาจจะแตกต่างกันออกไป ระบบฐานข้อมูลก็ไม่เหมือนกัน การที่จะทำให้ระบบหลาย ๆ ระบบทำงานต่อเชื่อมกันได้อย่างราบรื่นนั้นไม่ง่ายเลย

Microsoft จึงได้พัฒนารูปแบบการพัฒนาระบบคอมพิวเตอร์ขึ้นมา ซึ่งเรียกว่า .Net Framework นั้นเอง (อันที่จริงแล้ว Microsoft ไม่ได้เป็นผู้คิดค้นเรื่องพวกนี้ขึ้นมาแต่เพียงผู้เดียว ซึ่งสิ่งต่างๆ เหล่านี้มีผู้คิดค้นพัฒนาจากหลายบริษัท หลายหน่วยงาน ยกตัวอย่างเช่น Sun Microsystems, IBM, ฯลฯ หากแต่ว่า Microsoft นำแนวคิดเหล่านั้นมาออกแบบให้อยู่ในรูปแบบที่สามารถต่อเชื่อมกันได้ง่ายขึ้น เป็นระบบมากขึ้น) เพราะฉะนั้น ถ้าจะให้นิยามคำว่า .Net คงไม่สามารถชี้เฉพาะได้ว่า .Net คืออะไร เพราะจริง ๆ แล้ว .Net ประกอบไปด้วยส่วนประกอบต่าง ๆ หลายส่วนด้วยกัน ส่วนประกอบเหล่านี้ถูกออกแบบมาเพื่อให้ทำงานได้เข้าหากันได้ดียิ่งขึ้น

### 2.1.2 .Net Framework นั้นประกอบไปด้วยอะไรบ้าง

ส่วนประกอบหลัก ๆ ของ .Net Framework แบ่งเป็นชั้น ๆ ได้ดัง diagram ต่อไปนี้



รูปที่ 2.1 แสดงส่วนประกอบของ .Net Framework

1. **Common Language Runtime (CLR)** เป็นส่วนพื้นฐานที่ติดต่อกับระบบปฏิบัติการ Windows ทำหน้าที่เป็น run-time environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้บน .Net CLR มีส่วนของ compiler ทั้งที่เป็นแบบปกติ (compile ก่อนที่จะนำโปรแกรมไปใช้) และแบบ Just-In-Time (compile เมื่อจะใช้โปรแกรมนั้น ๆ) มีส่วนของ Memory Management ที่เอาไว้สำหรับจัดสรรหน่วยความจำของเครื่องให้กับโปรแกรม รวมไปถึงการทำ Garbage Collection (การเรียกคืนหน่วยความจำที่ไม่ได้ใช้อีกต่อไป) ส่วนของ Common Type Systems (CTS) ทำให้ภาษาต่าง ๆ ที่เขียนขึ้นบน .Net สามารถทำงานร่วมกันได้ เพราะขนาด และรูปแบบของข้อมูลที่เก็บไว้นั้นเป็นรูปแบบเดียวกัน

2. **Base Classes** เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ เช่น การติดต่อบริการฐานข้อมูล (ADO.Net), การติดต่อกับ file system ของ server (IO), ฯลฯ

3. **Programming Languages** เป็นเซตของ ภาษาคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อการเขียนโปรแกรมบน .Net Framework ไมโครซอฟท์นั้นเน้นไปที่ 3-4 ภาษาหลัก ๆ ได้แก่ VB.Net ซึ่งเป็นตัวที่พัฒนาต่อมาจาก VB, C# ซึ่งเป็นภาษาใหม่ที่มี syntax ใกล้เคียงกับ Java และ C++, Visual C++, และ JScript.Net ส่วนภาษาอื่น ๆ นั้น มีบริษัท หรือหน่วยงานอื่น ๆ เป็นผู้พัฒนาขึ้น ซึ่งคาดว่าจะมีเป็นสิบ ๆ ภาษา สำหรับ .Net Framework นั้นไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม Compiler ใน CLR ก็จะ compile โปรแกรมนั้นให้อยู่ในรูปแบบของ Intermediate Language (IL) ซึ่งจะถูกนำไปแปลเป็นภาษาเครื่อง (Native Code) อีกทีเมื่อตอนที่นำไปใช้

4. **ASP.Net** เป็นภาษา script ที่พัฒนาต่อมาจาก ASP ตัวเก่า เพื่อให้โปรแกรมเมอร์สามารถพัฒนา web application ให้ใช้ .Net ได้สะดวกขึ้น ASP.Net นี้ถึงแม้จะอ้างอิงมาจาก ASP ตัวเก่า แต่ก็มี syntax หลายส่วนที่เปลี่ยนแปลงไป อย่างไรก็ตาม ผู้ที่เคยเขียน ASP มาก่อนไม่น่าจะมีปัญหาในการอ่าน และเขียน ASP มากนัก

หมายเหตุ : สำหรับผู้อ่านที่คุ้นเคยกับ Java ของ Sun มาก่อน จะเห็นได้ว่า .Net Framework กับ Sun's J2EE นั้นคล้ายกันมากทีเดียว สรุปโดยคร่าว ๆ ได้ดังนี้ CLR = JVM (Java Virtual Machine), IL = Java Bytecode, .Net base classes = Java Class Library, .Net Programming Languages = Java Language, ASP.Net = Java Server Page (JSP). ข้อแตกต่างในด้าน Architecture หลัก ๆ ก็คือ IL ของ .Net นั้นต้อง run บน Windows เท่านั้น (ไมโครซอฟท์บอกว่า run บน OS ใหนก็ได้)

แต่อย่างที่เราทราบกันดี คงยากที่จะได้เห็นในอนาคตอันใกล้ หรือไม่ก็คงไม่ดีเท่ากับ run บน Windows) แต่ Java Bytecode นั้นสามารถ run บน OS ไหนก็ได้ที่มี JVM. ส่วนในด้าน performance นั้น ไมโครซอฟท์ ได้ทำการเปรียบเทียบโดยพัฒนาโปรแกรม Pet Shop ด้วย .Net โปรแกรม Pet Shop นี้เป็น reference application ที่พัฒนาบน J2EE โดย Sun ไมโครซอฟท์แสดงให้เห็นว่า .Net นั้นทำให้ code สั้นลงหลายเท่าตัว ทำให้โปรแกรมทำงานเร็วขึ้นหลายเท่า เป็นต้น ในส่วนนี้ก็คงต้องรอดูกันต่อไปว่า Sun จะพัฒนา Java อย่างไร เพื่อแก้เกมส์ในจุดนี้

### 2.1.3 .Net Framework มีดีตรงไหน

1) ทำให้พัฒนาโปรแกรมได้เร็วขึ้น - มีโปรแกรมพื้นฐานส่วนมากไว้ให้ใช้เรียบร้อยแล้ว (base classes) โปรแกรมที่ถูกพัฒนาขึ้นมาใหม่สามารถนำไป compile เพื่อให้โปรแกรมอื่นๆ ได้ใช้อีก (reusable)

2) โปรแกรม reliable ขึ้น - เนื่องจากการเขียนโปรแกรมบางรูปแบบ บน .Net นั้นไม่สามารถกระทำได้ด้วยภาษาที่มีให้บน .Net ยกตัวอย่างเช่น การใช้ Pointer ในภาษา C ทำให้ลดโอกาสที่โปรแกรมจะทำอะไรผิดพลาดจนทำให้ระบบไม่สามารถทำงานต่อไปได้

3) ปลอดภัยมากขึ้น - เพราะว่ามี .Net Framework ควบคุมโปรแกรมว่า อะไรทำได้ อะไรทำไม่ได้

4) การนำโปรแกรมที่พัฒนาขึ้นไปใช้บน server จริงง่ายขึ้น - .Net Framework อนุญาตให้โปรแกรมที่พัฒนาขึ้นนั้นระบุข้อมูลต่าง ๆ ไว้กับ code เลข ทำให้ไม่ต้องนำไป register เหมือนที่ component ต้องทำ

### 2.1.4 คำศัพท์อื่น ๆ ที่น่าสนใจ

- **Web Forms** เป็นส่วนที่ใช้สร้าง interface ต่าง ๆ บน web browser ของผู้เข้าชม เว็บไซต์ web form จะทำการตรวจสอบ browser ของผู้ใช้ และสร้าง HTML ที่ถูกต้องสำหรับ browser นั้น ๆ
- **Server Controls** เป็นส่วนต่าง ๆ ที่ โปรแกรมเมอร์สามารถนำมาประกอบกันเพื่อใช้งานบนเว็บไซต์ได้ server control นั้นมีหลายประเภทได้แก่
  - **HTML Server Controls** ได้แก่ HTML Tag ทั่ว ๆ ไป เช่น input, table, ฯลฯ
  - **Web Server Controls** เป็น control ชนิดใหม่ที่ .Net มีให้ใช้ เช่น CheckBoxList, RadioButtonList, DataGrid, Calendar, ฯลฯ ข้อดีของ Server Control ก็คือ มันสามารถเก็บค่าต่าง ๆ บน form ไว้ได้โดยอัตโนมัติ ซึ่งทำให้โปรแกรมเมอร์ไม่ต้องเสียเวลาสร้าง hidden field หรือส่งค่าผ่านทาง URL.

- **Validator Controls** เป็น control ชนิดใหม่อีกเช่นกัน มีไว้ใช้สำหรับการตรวจสอบข้อมูลที่ใช้กรอกลงในฟอร์ม ว่าถูกต้องหรือไม่ก่อนที่จะถูกส่งไปประมวลผลที่ server ยกตัวอย่างเช่น การตรวจสอบว่า e-mail address ที่ผู้กรอกมานั้นอยู่ในรูปแบบที่ถูกต้องหรือไม่

(username@somedomain.com)

Server Control มีส่วนอย่างมากในการทำให้เขียนโปรแกรมได้ง่าย และเร็วขึ้น

- **Compiled Code และ Code Behind** - Compiled code นั้นเทียบได้กับ Component ใน ASP แบบเก่า ใน .Net Framework นั้น compiled code มีชื่อเรียกว่า **Assembly (.dll file)** ซึ่งเราสามารถนำไปไว้ใน directory ที่ชื่อ bin .Net จะตรวจเช็คโดยอัตโนมัติ โดยที่เราไม่ต้องทำการ register assembly เหล่านั้นเหมือนอย่างที่เคยทำกับ component. Code behind คือการแยกส่วนของเนื้อหาและส่วนของโปรแกรม ในหน้าเว็บออกจากกันเป็นคนละไฟล์ ส่วนนี้ทำให้การเขียนโปรแกรม และแก้ไขโปรแกรมทำได้ง่ายมากขึ้นไปอีก ต่างจาก ASP ในแบบเก่า ที่ทุกอย่างผสมปนเปกันไปในหน้าเดียวกัน สำหรับรูปแบบใหม่ web page จะมีนามสกุลเป็น .aspx ส่วน code behind จะมีนามสกุลเป็น .aspx.cs (ถ้าใช้ C#) หรือ .aspx.vb (ถ้าใช้ VB.Net)

สถาปัตยกรรม .NET Framework เป็นกรอบการทำงานของการเขียนโปรแกรมที่ไม่โครซอฟท์คิดขึ้นมาเพื่อรองรับการติดต่อสื่อสาร เพื่อแลกเปลี่ยนข้อมูล ( Exchange Data ) ระหว่างกัน หรือแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์ม ( Platform ) ให้มีความสมบูรณ์ยิ่งขึ้น โดยอาศัยภาษา XML ( Extensible Markup Language ) ทำหน้าที่เป็นตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างแพลตฟอร์มไฟล์ของข้อมูล

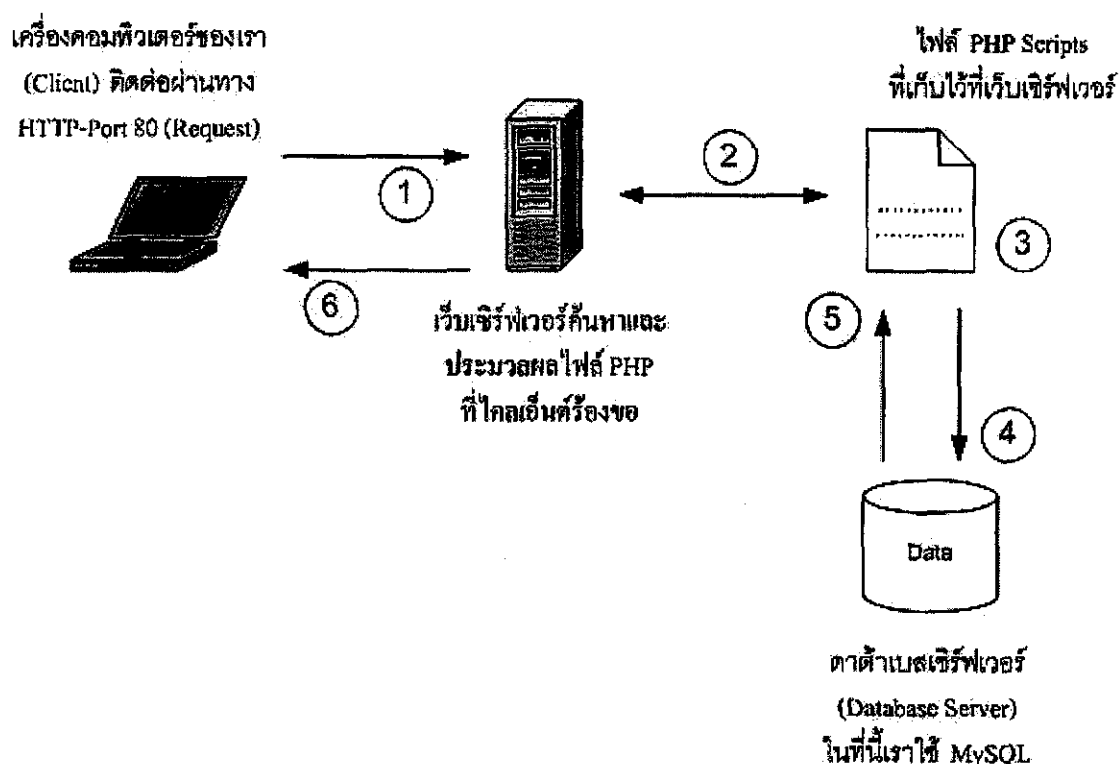
Visual Studio .NET ถือเป็นเครื่องมือพัฒนาแอปพลิเคชันตัวแรกๆ ที่ออกมารองรับการทำงานตามแนวความคิดของสถาปัตยกรรม .NET Framework ซึ่งประกอบไปด้วยเครื่องมือต่างๆ ดังนี้

- Visual Basic .NET
- Visual C++ .NET
- Visual C# .NET
- ASP .NET

ภาษา Visual C# ถือเป็นภาษาที่เกิดขึ้นมาพร้อมกับแนวความคิดของการเขียนโปรแกรมในยุค .NET อาจจะกล่าวได้ว่า ภาษา Visual C# คือภาษาดั้งเดิมของการเขียนโปรแกรมใน .NET นั่นเอง โดยที่ภาษาอื่นๆ ที่เกิดขึ้นมาก่อนหน้านี้ จะต้องปรับตัวเข้าหา .NET ทั้งหมด ซึ่งสามารถสังเกตได้ว่าไวยากรณ์การใช้งานของแต่ละภาษานั้น ล้วนแล้วแต่ถูกปรับเปลี่ยนไปจากเวอร์ชันก่อนหน้าอย่างสิ้นเชิง

## 2.2 Web Application

Web application คือ program ที่อยู่ใน web server ที่คอยให้บริการสิ่งที่ร้องขอ (request) จากทาง client ผ่าน protocol HTTP ซึ่งจะแสดงผลที่ร้องขอในรูปแบบของ HTML page ผ่านทางเบราว์เซอร์ ซึ่งก็คือเว็บไซต์ต่างๆ ที่เราใช้บริการอยู่นั่นเอง Web Application สามารถตอบสนองความคิด Distributed Processing ได้ในระดับหนึ่งซึ่งก็คือ การแบ่งการประมวลผลไว้ที่ฝั่ง Client และฝั่ง Server และมักจะมีการใช้ฐานข้อมูล (database) ควบคู่กับการทำ Web Application ไปด้วยตามความต้องการในการทำ E-Business และ E-Commerce ที่กำลังเป็นที่นิยมในปัจจุบัน แต่ปัญหาที่ตามมาคือ เรื่องของการจ่ายเงินหรือที่เรียกว่า E-payment หรือ Payment-Gateway ซึ่ง Web Application ที่ทำ E-Commerce ต้องใช้บริการจากธนาคาร Online ในการจัดเก็บเงินกับลูกค้า ด้วยเทคโนโลยีนี้ การใช้บริการเก็บเงินจากธนาคารออนไลน์ ซึ่งมีความจำเป็นที่ผู้ค้าจะต้องไปทำการตกลงกับธนาคารและเขียนโปรแกรมให้ตรงตามมาตรฐานที่ธนาคาร Online กำหนดไว้



รูปที่ 2.2 แสดงการเชื่อมต่อ Database Server กับ Web Server

การทำ Web Application ก่อนอื่นต้องติดตั้งเว็บเซิร์ฟเวอร์ขึ้นมา และมี DBMS (Database Management System) ทำหน้าที่จัดการข้อมูลเกี่ยวกับฐานข้อมูลทั้งหมด เครื่องมือ (Tools) ที่นำมาใช้เป็นเว็บเซิร์ฟเวอร์ มีให้เลือกได้หลายตัว ได้แก่

- IIS (Internet Information Server)
- Apache บน LINUX, UNIX

- Oracle

จากรูปจะเห็นการทำงานของ Web Application ซึ่งประกอบไปด้วยส่วนประกอบต่างๆ ซึ่งอาจจะมากกว่านี้ก็ได้แล้วแต่ Application ซึ่งขั้นตอนคร่าวๆจะสรุปเป็นขั้นตอนต่างๆ ดังนี้

ขั้นตอนที่ 1 ฟังก์ชันไคลเอ็นต์ (Client) จะทำการร้องขอหรือเรียกใช้งานไฟล์ PHP ที่เก็บในเครื่องเซิร์ฟเวอร์ (Server)

ขั้นตอนที่ 2 ฟังก์ชันเซิร์ฟเวอร์จะทำการค้นหาไฟล์ PHP แล้วทำการประมวลผลไฟล์ PHP ตามที่ไคลเอ็นต์ทำการร้องขอมา

ขั้นตอนที่ 3 ทำการประมวลผลไฟล์ PHP

ขั้นตอนที่ 4 และ 5 เป็นการติดต่อกับฐานข้อมูล และนำข้อมูลในฐานข้อมูลมาใช้ร่วมกับการประมวลผล

ขั้นตอนที่ 6 ส่งผลลัพธ์จากการประมวลผลไปให้เครื่องไคลเอ็นต์

เครื่องมือที่ใช้เป็น DBMS ก็มีให้เลือกหลายตัวเช่น SQLServer, ORACLE, INFORMIX,

ODBC, SYBASE, PHPMyAdmin และ DB2 เป็นต้น แต่ในที่นี้ใช้ MySQL ควบคู่ไปกับ phpMyAdmin

**MySQL :** MySQL เป็น Database Server ที่เหมาะกับองค์กรขนาดกลางที่มีข้อมูลไม่มากนัก และเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) ซึ่งเป็นพีแรว์ทางด้านฐานข้อมูล จึงได้รับความนิยมอย่างมากในปัจจุบัน โดยสามารถดาวน์โหลดซอร์สโค้ด (Source Code) ได้จากอินเทอร์เน็ตโดยไม่ต้องเสียค่าใช้จ่าย และสามารถแก้ไขได้ตามความต้องการ พร้อมทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการ ตัวอย่างเช่น Unix, Mac และ Windows นอกจากนี้ยังทำงานร่วมกับ Java, C, C++, PHP, ASP หรือ Perl ได้

**phpMyAdmin :** การทำงานของ MySQL นั้นอาจจะยุ่งยากในการคีย์ข้อมูล เพราะทำงานบน DOS ตัว phpMyAdmin ซึ่งเป็นโปรแกรม mysql client แบบ open source ที่ใช้จัดการ MySQL ได้ผ่านหน้า web browser โดยโปรแกรมนี้เขียนด้วย PHP

### 2.3 Web Service

ปัจจุบันท่ามกลางสถานะเศรษฐกิจที่เปลี่ยนแปลง ประกอบกับการแข่งขันทางธุรกิจที่สูงขึ้น ปัจจัยหลายๆ อย่างเข้ามามีอิทธิพลต่อรูปแบบการดำเนินธุรกิจ ทว่าแนวโน้มที่เห็นได้ชัดเจนสำหรับธุรกิจในยุคใหม่ คือ การทำธุรกิจในลักษณะอิเล็กทรอนิกส์หรือที่เรียกว่า พาณิชยกรรมอิเล็กทรอนิกส์ (E-Business) ซึ่งเป็นวิธีการทำธุรกิจที่น่าสนใจและกำลังแพร่หลายในหลายๆ ธุรกิจ รวมทั้งยังเป็นปัจจัยหนึ่งที่มีส่วนทำให้องค์กรทางธุรกิจนั้นๆ ยังยืนหยัดอยู่ได้ องค์กรทาง

ธุรกิจต่างๆ ได้เรียนรู้ที่จะนำข้อมูลและทรัพยากรด้านเทคโนโลยีสารสนเทศ มาใช้เป็นเครื่องมือในการพัฒนาธุรกิจของตนมากยิ่งขึ้น เพื่อเป็นฐานที่มั่นคงและการเติบโตทางเศรษฐกิจในอนาคต

องค์กรทางธุรกิจต่างๆ เริ่มมองเห็นวิธีที่ดีที่สุดสำหรับใช้ในการพัฒนาประสิทธิภาพของการปฏิบัติงาน ลดค่าใช้จ่าย และสานสัมพันธ์ที่มีกับลูกค้ารวมถึงซัพพลายเออร์รายสำคัญให้แน่นแฟ้นยิ่งขึ้น โดย สร้างวิธีการดำเนินธุรกิจแบบใหม่ที่ใช้เทคโนโลยีผสมผสานกับเครือข่ายอินเทอร์เน็ตที่แพร่หลายในปัจจุบัน ซึ่งเทคโนโลยีที่กำลังได้รับความสนใจและยังเป็นเครื่องมือที่สนับสนุนความต้องการดังกล่าวในธุรกิจยุคใหม่ก็คือ **Web Services**

เป็นที่ยอมรับกันว่าเทคโนโลยีสารสนเทศและเครือข่ายอินเทอร์เน็ต เป็นเครื่องมือธุรกิจที่จำเป็นในการดำเนินธุรกิจยุคใหม่ไปแล้ว ซึ่ง Web Application เป็นพระเอกหรือเป็นเครื่องมือสำคัญในการทำธุรกิจบนโลกของอินเทอร์เน็ต แต่ปัจจุบัน Web Application กำลังจะกลายเป็นเพียงตัวประกอบ ด้วยการที่องค์กรธุรกิจเริ่มหันมาสนใจเทคโนโลยีใหม่อย่าง Web Services ซึ่งเข้ามาช่วยเพิ่มประสิทธิภาพให้กับระบบการดำเนินธุรกิจขององค์กร และยังสามารถทำงานร่วมกับ Web Application เดิมได้เป็นอย่างดี เพื่อให้องค์กรธุรกิจต่างๆ ได้เห็นแนวทางในการนำ Web Services ไปใช้ในธุรกิจของตน แต่ละองค์กรจำเป็นต้องเข้าใจความหมายของ Web Application และ Web Services เพื่อที่จะได้เห็นความแตกต่างและนำไปประยุกต์ใช้งานได้อย่างถูกต้องและมีประสิทธิภาพมากที่สุด

### 2.3.1 Web Service คืออะไร

หากเรามองย้อนกลับไปซักไม่กี่ปีที่ผ่านมาใครจะไปคาดคิดว่าเว็บมันจะเติบโตและได้รับความนิยม สูงมากขนาดนี้ ทุกวันนี้หลายๆ คนคงขาดเว็บไม่ได้ เหตุผลที่เว็บประสบความสำเร็จก็คงเป็นเพราะเหตุผลเพียงไม่กี่อย่างคือ ความสะดวก และใช้งานง่าย ในฝั่งผู้ให้บริการ (ผ่านเว็บ) ก็คงมองว่าถ้ามีเว็บเซอร์เวอร์ ก็ขายสินค้าได้ทั่วโลก ในฝั่งผู้ใช้งาน ขอให้คุณเลื่อนเมาส์กับใช้ keyboard เป็น คุณก็ติดต่อ ค้นหา ชื่อของ ได้ทั่วโลก ในมุมมองของ Software เว็บก็ทำหน้าที่อยู่ 3 อย่างคือ GET POST และ ก็ PUT ในเรื่องของ Web Service ก็คือการใช้ Web ที่ไม่เพียงแต่เกี่ยวกับข้อมูลอย่างเดียว แต่หมายถึงการบริการด้วย

คำว่า Service ไม่ได้หมายถึงอะไรที่เด่นชัดอย่าง Promool.com Pantip.com แต่หมายถึงส่วนประกอบที่คนอื่นฯ นำไปใช้ในการทำบริการที่กว้างกว่านี้ด้วย ตัวอย่างเช่น Microsoft Passport ที่ให้บริการตรวจสอบความเป็นตัวตนจริง (Authentication) ผ่านเว็บ ทำให้การบริการข่าวของ Bangkok Post ไม่ต้องตรวจสอบการเข้าสู่ระบบเอง แต่ยกให้ Passport เป็นตัวจัดการแทนหรืออย่าง Dynamic services whitepaper ของ Oracle ก็มีหน้าที่ให้บริการ แปลงค่าเงิน แปลงภาษา การส่งของ กระบวนการเคลมสินค้า เป็นต้น ส่วนความหมายอย่างเป็นทางการของ Web Service ก็คงเป็นของ IBM ที่กล่าวว่า

เว็บเซอวิส คือ Web Application ยุคใหม่ ที่ประกอบด้วยส่วนย่อยๆ มีความสมบูรณ์ในตัวเอง สามารถติดตั้ง ค้นหา เริ่มทำงานได้ผ่านเว็บ Web Service สามารถทำอะไรก็ได้ตั้งแต่งานง่ายๆ เช่นดึงข้อมูล จนถึงกระบวนการทางธุรกิจที่ซับซ้อน เมื่อ Web Service ตัวใดตัวหนึ่งเริ่มทำงาน Web Service ตัวอื่นก็สามารถรับรู้และเริ่มทำงานได้อีกด้วย

หลายคนอาจจะถามว่าทำไมต้องเป็น Web เพราะเรามี Middle Ware อื่นๆ มากมายเช่น RMI Jini CORBA DCOM ฯลฯ แม้ Middle Ware เหล่านี้จะสามารถรองรับได้ แต่ไม่มีตัวใดตัวหนึ่งที่เด่นจริง แต่ในเมื่อ Web มีจุดเด่นในเรื่องของการให้บริการข้อมูลที่สะดวก ใช้งานง่าย จึงกลายเป็นตัวประสาน Middle Ware ต่างๆ เข้าด้วยกันซึ่งจะให้คุยกันเองคงยาก ยิ่ง Web ทำหน้าที่เป็นตัวกลางให้ Middle Ware เหล่านี้สามารถคุยกันได้ และมีประสิทธิภาพกว่าวิธีการเดิมๆ มาก

หากเรามองจากกรณีของ n-tier application จะพบว่า web service คือกลไกในการเข้าถึงบริการที่แต่ละ Middle Ware ให้บริการ การเข้าถึงจะอาศัย Listener และส่วนประกอบที่ระบุถึงบริการต่างๆ ที่รองรับการทำงาน โดยการทำงานจริงๆ นั้นก็ใช้วิธีการปกติของ Middle Ware นั้นๆ

### 2.3.2 พื้นฐานของ Web Service

พื้นฐานของ Web Service ก็คือ XML กับ HTTP ซึ่งจะพบว่า HTTP ก็เป็นที่รู้จักกันดี และไปได้ทั่วทุกแห่งที่มี internet ส่วน XML คือภาษาสากลที่คุณสามารถปรับแต่งได้ตามใจชอบเพื่อให้เกิดกิจกรรมระหว่าง client และบริการ หรือระหว่างส่วนประกอบต่างๆ เบื้องหลัง Web server ก็คือ ข้อความ XML จะถูกแปลงให้การขอบริการจาก Middle ware และผลที่ได้ก็จะแปลงกลับมาในรูปแบบ XML

ยกตัวอย่างให้เห็นง่ายๆ คุณต้องการให้เครื่อง PC อ่านค่าจาก serial port แล้วส่งไปประมวลผลบนเครื่อง UNIX แล้วส่งผลกลับมาแสดงบนจอ PC ถ้าเป็นเมื่อก่อน คุณก็คงต้องแปลงข้อมูลที่ได้ออกอยู่ในรูปของ ASCII แล้วส่งไปยัง UNIX พร้อมคำสั่งว่าให้ทำอะไร ในฝั่ง UNIX คุณก็ต้องมาแยกว่าอันไหนคือคำสั่ง อันไหนคือข้อมูล เมื่อประมวลผลแล้ว จะส่งกลับมาในรูปแบบไหน แล้วถ้าหากจะส่งไปหาเครื่องที่เป็น MAC ท่านจะต้องเขียนโปรแกรมเพิ่มในส่วนไหนบ้าง จะพบว่าเราต้องพัฒนากันเป็นคู่ๆ ไป และต้องนิยามในแต่ละฝั่งให้ชัดเจน แต่หากเป็น Web Service คุณจะพบว่า เราแปลงข้อมูลให้อยู่ในรูปแบบ XML แต่ละคุณก็ต้องการรู้แค่ มาตรฐาน XML ก็พอ แล้วต่างคนต่างก็เขียน Service ของตัวเอง ไม่ต้องกังวลเรื่องของการเชื่อมโยงอีกต่อไป และ Protocol ที่ส่งก็คือ HTTP นั่นเอง ถ้าท่านเชื่อมโยงกับ HTTP (หรือเว็บ) ได้ ท่านก็ใช้บริการทุกอย่างได้

แต่เดี๋ยวก่อนการเข้าถึงและการส่งงานนั้นยังเป็นเพียงโครงสร้างพื้นฐาน แต่ในความเป็นจริงยังมีอะไรมากกว่านั้น เช่น การค้นหา การทำธุรกรรม ความปลอดภัย การพิสูจน์ตัวตน และอื่นๆ อันเป็นบริการที่ทำให้เป็นบริการพื้นฐานจริงๆ

ระบบเพิ่มเติมที่ต้องมีและต้องรักษาความสะอาดและใช้งานง่ายไว้ด้วย พื้นฐานของ Web Service เต็มรูปแบบคือ XML + HTTP + SOAP + WSDL + UDDI หรือในระดับสูงกว่านั้น แต่ไม่ได้ถือเป็นสิ่งจำเป็นเสมอไปคือต้องเพิ่มเทคโนโลยี XAML, XLANG, XKMS, XFS เป็นต้น

### 2.3.3 เทคโนโลยีที่ใช้ในการพัฒนาเว็บเซอร์วิส

การทำงานของ Services ประกอบไปด้วย มาตรฐานหลัก 4 อย่าง ซึ่งสามารถอธิบายอย่างง่ายๆ ได้ดังนี้

#### XML (Extensible Markup Language)

เป็นภาษา Markup ที่เป็น text-based ซึ่งทำให้เป็น มาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็ว ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้าง เว็บเพจ ที่สามารถแสดงผลได้โดยโปรแกรมเบราว์เซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระ แล้วทำการส่ง XML ชุดนี้ไป ประมวลผลยังแอปพลิเคชันใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้

#### XML & HTML

HTML ภาษาที่ใช้ในการเขียน Web มากที่สุดนั้นเป็นเพราะมีรูปแบบที่ง่ายต่อการแสดงผลของ Browser เนื่องจาก มี tag ตามตัวที่สามารถบอกได้ว่าเมื่อเจอ tag นี้จะแสดงผลอย่างไร เช่น เมื่อเจอ tag <B>...</B> ในเอกสารก็ให้แสดงข้อความที่อยู่ระหว่าง tag เป็นตัวหนา แต่จะสังเกตได้ว่าคอมพิวเตอร์จะไม่เข้าใจว่าข้อความนั้นคืออะไร เพียงแต่รู้ว่าจะแสดงผลอย่างไร นั่นแสดงว่าไม่สามารถนำข้อมูลภายใน tag เหล่านี้ไปทำการประมวลใดๆ ได้เลย

XML เป็นภาษาที่มีลักษณะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อได้ เช่น

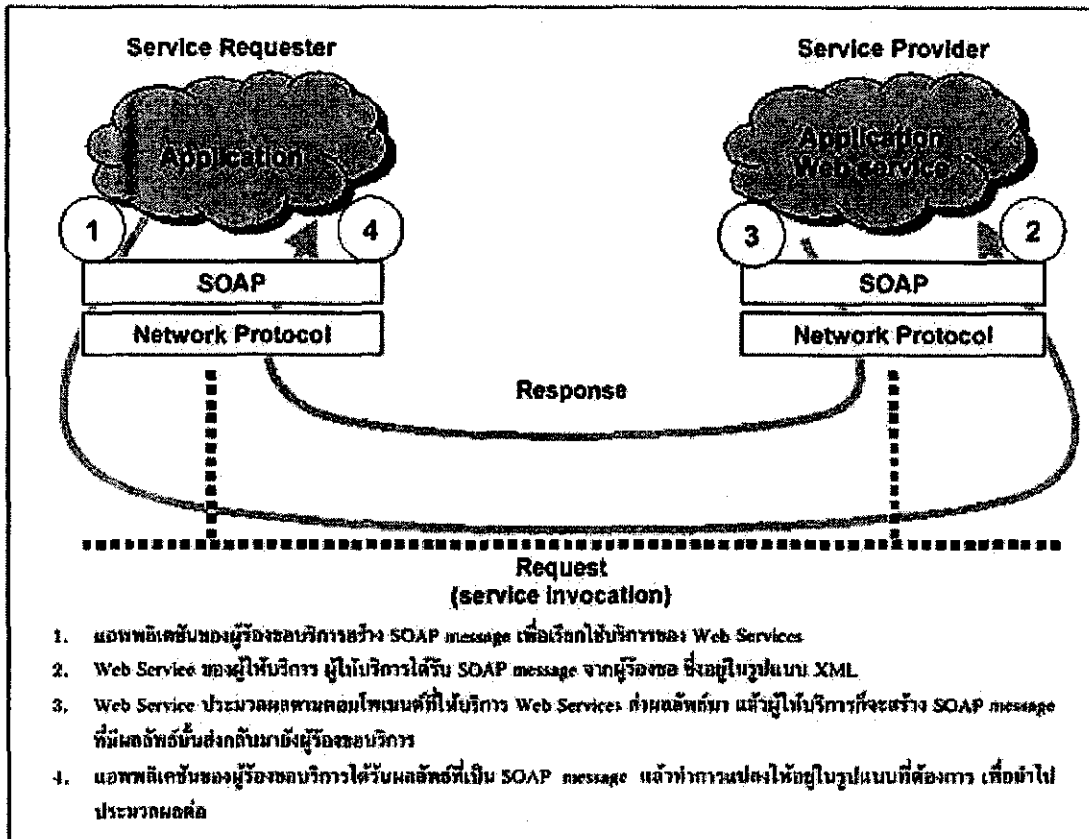
```
<ComputerBook>
  <book>
    <name>เว็บเซอร์วิส</name>
    <price>10.00$</price>
  </book>
  <book>
    <name>xml</name>
    <price>10.00$</price>
  </book>
</computerBook>
```

จากตัวอย่าง จะเห็นได้ว่ารูปแบบเอกสาร xml ไม่ได้บอกวิธีแสดงผลไว้ แต่เอกสารสามารถสื่อความหมายให้คอมพิวเตอร์เข้าใจได้ และนำค่าไปประมวลผลต่อได้ แต่ถ้าเราต้องการจะแสดงผลเราต้องใช้ควบคู่กันกับ HTML ข้อมูลที่มีโครงสร้างของภาษา XML จะถูกนำไปประมวลผลต่ออย่างอัตโนมัติได้อย่างง่ายดาย ภาษา XML จึงถูกนำมาใช้เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลของ Web Services

#### SOAP (Simple Object Access Protocol)

SOAP กลายเป็นสิ่งที่มีความสำคัญสำหรับ Web Services อย่างรวดเร็ว เป็นโปรโตคอลที่ผู้จัดหา Web Services เลือกใช้ที่จะส่ง message ระหว่าง Web Services SOAP เป็น Transport Protocol ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโปรโตคอลร่วมในการส่งผ่านเครือข่าย SOAP จะระบุวิธีในการเข้ารหัสส่วนหัว (Header Encoding) ของทั้ง HTTP และไฟล์ XML ไว้อย่างชัดเจนทั้งใน ส่วนของการติดต่อไปยังคอมพิวเตอร์อีกเครื่องหนึ่งและส่งผ่านข้อมูลไปให้ รวมถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย

SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol) และใช้ HTTP เป็นโปรโตคอลร่วม สำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนดเมสเสจจิงโปรโตคอล (Messaging Protocol) ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) เช่น ผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช้ RMI (Remote Method Invocation) ตามวิธีการของ โปรแกรมแบบออปเจ็ค บริษัทไมโครซอฟท์, ไอบีเอ็ม, โลดัส, ยูสเซอร์แลนด์ (UserLand) และ ดีเวลลอปเปอร์เมนเตอร์ (DeveloperMenter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้นโดยการพัฒนา SOAP แม้ว่า SOAP จะไม่ได้กำหนดเมสเสจจิงโปรโตคอล (Messaging Protocol) ไว้ แต่ SOAP ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจากมันได้ถูกใช้ในการพัฒนา SOA อย่างแพร่ หลายแล้วนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโปรโตคอลที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก่อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการแบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา



รูปที่ 2.3 XML message using SOAP

### ทำไมต้องเป็น SOAP

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่น โดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM,EJB หรือ CORBA นั้นไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล HTTP (เครือข่ายที่ปัจจุบันใช้งานแพร่หลาย และสะดวกมากที่สุดได้แก่เครือข่าย Internet ซึ่งให้การสื่อสารผ่านโปรโตคอล HTTP เป็นส่วนใหญ่)

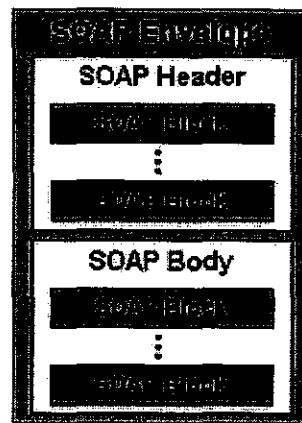
เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ยกเว้น CORBA ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy Server ด้วยเนื่องจากโดยปกติ เซอร์เวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอล HTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการเว็บเซอร์วิส ก็คือให้ทำงานอยู่บน โปรโตคอล HTTP ไปเลย ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

## โครงสร้างของ SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนๆของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้
3. SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส



รูปที่ 2.4 โครงสร้างของเอกสาร SOAP

ตัวอย่างเอกสาร SOAP อย่างง่ายของการสอบถามราคา ดอกกุหลาบ จำนวน 100 ดอก

```
<soap:Envelope>
  <soap:Body>
    <GetPrice>
      <Item>Rose</Item>
      <Quantity>100</ Quantity >
    </ GetPrice >
  </ soap:Body >
</ soap:Envelope >
```

เรามาดูอีกตัวอย่างหนึ่งของเอกสาร SOAP ที่จะแสดงในรูปด้านล่างเพื่อจะได้ทำความเข้าใจในส่วนต่างๆ ที่มีความซับซ้อนขึ้น แต่ก็อย่าท้อจนเกินไปที่จะทำความเข้าใจได้ดังรูปที่ 2 SOAP Envelope ต่อไปนี้

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <!--ข้อมูลในส่วนของ Header -->
    <i:local xmlns:i="http://www.i3t.or.th/ws/">
      <i:currency>Bath</i:currency>
    </i:local>
  </soap:Header>
  </soap:Body>
    <!--ข้อมูลในส่วนของ Body -->
    <GetPrice>
      <Item>Rose</Item>
      <Quantity>100</Quantity >
    </GetPrice >
  </soap:Body >
  <soap:Fault>
    <!--ข้อมูลของ SOAP ในการมีข้อผิดพลาด จาก SOAP Node -->
  </soap:Fault>
</soap:Envelope >

```

### คำอธิบาย

<Envelope>

จะเป็น Root ของเอกสาร XML SOAP เสมอ โดยจากตัวอย่างจะใช้ Namespaces จาก "http://schemas.xmlsoap.org/soap/envelope/" (Schema for the SOAP/1.1 envelope) ดังนั้น Element ต่างๆ ในตัวอย่างนี้ที่อยู่ในเอกสาร XML ส่วนที่เป็นมาตรฐานของ SOAP เราจะมี Namespaces ที่เราตั้งเป็น soap (<soap:Envelop>, <soap:Header>, <soap:Body>, <soap:Fault>

### Attribute ใน Envelop

Attribute encodingStyle ในตัวอย่างมีการอ้างการ encoding จาก

"http://schemas.xmlsoap.org/

soap/encoding/" ( Schema for the SOAP/1.1 encoding) จะใช้ในกรณีที่เรามีการใช้ Parameter หลายชนิดข้อมูลในเอกสาร (Data Type) เช่น Boolean, String, Integer เป็นต้น

<Header>

จากตัวอย่างมีการเพิ่มเนื้อหาของเอกสาร SOAP เข้าไปในส่วนของ Header ในที่นี้คือ สกุลเงิน (Currency) ซึ่งส่วนนี้จะใช้ในส่วนของ Application ไม่ได้เป็นส่วนของมาตรฐานของ SOAP แต่ผู้ใช้งานเป็นผู้กำหนดใช้เอง (User-defined)

### Attribute ใน Header

ใน <Header> Element อาจจะมีการใส่ Attribute mustUnderstand เพื่อให้ฝั่งที่รับเอกสาร SOAP ให้ทำการประมวลผลในส่วน Header ด้วยโดยกำหนดค่าเป็น Boolean เช่น

```
<i:local xmlns:i="http://www.i3t.or.th/ws/">
  <i:country mustUnderstand="1">Thailand</i:country>
  <i:currency mustUnderstand="0">Bath</i:currency>
</i:local>
```

ค่าของ mustUnderstand ถ้าเป็น "0" แสดงว่าไม่ต้องประมวลผล ถ้าเป็น "1" : true คือ จำเป็นต้องมีการประมวลผล (โดยค่า default = "0" : fault)

### <Body>

โดยปกติแล้วเอกสาร SOAP จะต้องมีส่วนของ Body เพราะเป็นส่วนเป็นเนื้อหาสาระจริงๆ ของ SOAP จากตัวอย่างเราต้องการสอบถามราคาของดอกกุหลาบจำนวน 100 ดอก ซึ่ง <GetPrice>, <Item> และ <Quantity>เป็น Element ที่ใช้งานใน Application ไม่ใช่มาตรฐานของ SOAP

### <Fault>

เป็นส่วนที่จะถูกใช้เมื่อมีข้อผิดพลาดในการประมวลผลของเอกสาร SOAP ซึ่งโดยปกติจะเห็นเฉพาะที่เป็นเอกสารตอบกลับเท่านั้น (Reply Message หรือ Answer Message) ซึ่งตัวอย่างจะเป็นเอกสารที่เป็นการร้องขอบริการ (Request Message) ซึ่งไม่น่าจะมีส่วนของ <Fault> Element แต่ที่เขียนไว้เพื่อให้เห็นภาพรวมของโครงสร้างเอกสาร SOAP ทั้งหมดก่อน ตัวอย่างข้อมูลใน <Fault> Element ที่มีการเกิดข้อผิดพลาด

```
<soap:Fault>
  <faultcode>Client</faultcode>
  <faultstring>Invalid Request</faultstring>
</soap:Fault>
```

### ตารางอธิบาย Fault Element

Sub Element	คำอธิบาย
<faultcode>	ชื่อของการเกิดข้อผิดพลาด (ดูตารางด้านล่าง)
<faultstring>	คำอธิบายของข้อผิดพลาด
<faultactor>	ชื่อของแหล่งที่เกิดข้อผิดพลาด
<detail>	ข้อมูลรายละเอียดเพิ่มเติมของข้อผิดพลาดนั้น

จะเห็นว่า SOAP Envelope ก็เป็นเอกสารที่อยู่ในรูปแบบ XML ทั่วไป แต่สิ่งที่ทำให้ SOAP มีความสามารถมากขึ้นก็คือ SOAP ได้มีการกำหนดโครงสร้างของเอกสารเป็นส่วนๆ โดยอธิบายว่าส่วนใดมีหน้าที่อะไร (What) มีข้อมูลอะไรอยู่ในส่วนนั้น และใคร (Who) คือผู้ที่จะต้องสนใจในส่วนนั้น ๆ และกล่าวถึงว่าส่วนใดจำเป็นที่ต้องมีในเอกสาร และส่วนใดเป็นส่วนที่เพิ่มเติมอาจจะมีหรือไม่มีส่วนนั้นก็ได้ขึ้นอยู่กับความจำเป็นในการใช้งาน ทำให้การพัฒนาโปรแกรมมีทิศทางที่ค่อนข้างชัดเจน แต่สามารถเพิ่มขยายได้ในอนาคตอีกด้วย

### SOAP Intermediaries

จากตัวอย่างเอกสาร SOAP นั้นจะเห็นได้ว่าในส่วนของ <Header> Element จะเป็นส่วนที่เราจะขยายสารสนเทศ (information) ใหม่ๆ เข้าไปใน SOAP ได้ซึ่งตัวอย่างเรามีการอธิบายให้ผู้รับเอกสารตัวอย่างนี้ให้ทราบว่าในการสอบถามราคานี้ให้ใช้เงินในสกุลเงินบาท เป็นต้น ซึ่งเราจะเรียกการขยายในส่วนนี้ว่าเป็นการขยายในแนวดิ่ง (Vertical extensibility) แต่ถ้าเราจะมาดูในด้านการขยายในแนวราบ (Horizontal extensibility) ของ SOAP นั่นก็คือทำอย่างไรที่จะให้มีการส่งเอกสาร SOAP ไปยังปลายทางโดยเส้นทางของการเดินทางของเอกสารอาจจะไม่เป็นลักษณะแบบจุดต่อจุด (Point-to-point) แต่อาจจะเป็นการส่งเอกสารตามเส้นทางโดยเอกสารยังคงเป็นเอกสารตัวเดิม

### ความจำเป็นในการใช้ SOAP Intermediary

SOAP Intermediaries จะทำหน้าที่เป็นตัวผ่าน หรือตัวกลางในการส่งเอกสาร SOAP จากต้นทาง ในที่นี้คือ Requestor ไปยัง Provider เพื่อสร้างกระบวนการบางอย่างกับเอกสาร ซึ่งในการสร้างเส้นทางในการส่งแบบนี้อาจมีจุดประสงค์ดังต่อไปนี้คือ

1. **Intermediary** ทำหน้าที่เป็น Crossing trust domains เนื่องจากการทำงานของ Web Services เป็นการเรียกใช้งานแบบระยะไกล หรือมีการทำงานแบบระบบกระจาย (Distributed System) ดังนั้นวิธีการในด้านความปลอดภัยจึงเป็นส่วนหนึ่งที่ต้องคำนึงถึง ซึ่งวิธีการหนึ่งก็คือให้มีการส่งเอกสารมายังจุดหนึ่งที่มีความน่าเชื่อถือได้ก่อน แล้วส่งต่อไปยังปลายทางอีกที่หนึ่ง ซึ่งหลักการใช้ Trust Domains เราอาจจะดูตัวอย่างง่ายๆ เช่น ในองค์กรเราอาจจะมีเครือข่าย Intranet อยู่ภายในองค์กร เครื่องคอมพิวเตอร์แต่ละเครื่องอาจจะสามารถติดต่อไปยัง Server ได้โดยตรง แต่ถ้าต้องการออกไปยัง Internet หรือ Extranet ที่อยู่นอกองค์กรจะต้องผ่าน Trust Domains ก่อน ในที่นี้อาจจะเป็น Firewalls หรือ Virtual Private Network (VPN) gateways เพื่อความปลอดภัย

2. **Intermediary** ทำหน้าที่เป็นตัวขยายการทำงานแบบระบบกระจาย (Distributed System) ถ้าเรามองดูเฉพาะแง่ของการขอใช้บริการจาก Requestor ไปยังผู้ให้บริการ Provider โดยผู้ให้บริการจะมีการกระบวนการทำงานบางอย่างภายในและตอบกลับไปยังผู้ขอใช้ ซึ่งการทำงานแบบนี้ก็อยู่บนพื้นฐานของ Request and Respond นั่นเอง แต่ถ้าระบบของเรามีการขยายใหญ่ขึ้น

สิ่งที่เราอาจจะต้องดูเพิ่มขึ้นก็คือความสามารถในการรองรับงานของผู้ให้บริการด้วย ดังนั้นก่อนที่จะมีการส่งเอกสาร SOAP ไปยังเซิร์ฟเวอร์ ของผู้ให้บริการนั้นอาจจะมีการผ่าน Intermediary Server ก่อนเพื่อทำหน้าที่เป็น Buffer รับเอกสาร SOAP มาพักไว้และส่งเอกสารให้ผู้ให้บริการ (Provider) อีกทอดหนึ่ง การทำงานเช่นนี้จะช่วยให้ Server ของ ผู้ให้บริการไม่รับภาระงานมากจนเกินไปในช่วงเวลาใดเวลาหนึ่งจนอาจทำให้เกิดความเสียหายได้ และ Intermediary อาจจะมีการรวบรวมเอกสารแล้วส่งมาให้ผู้ให้บริการเป็นลักษณะ Batch เพื่อเพิ่ม Performance การใช้งาน Intermediary ลักษณะนี้ผู้ขอใช้บริการอาจจะไม่เห็น Intermediary แต่จะมองเห็นว่าติดต่อกับผู้ให้บริการโดยตรง

3. การใช้งาน Intermediary ในลักษณะ Value-added ให้กับระบบ เช่น ใช้ Intermediary ในการเข้ารหัสข้อมูล (Encrypts) และเซ็นลายเซ็นอิเล็กทรอนิกส์ (Digital signs) ในการส่ง และรับ หรืออีกตัวอย่างหนึ่งเช่นต้องการเก็บข้อมูลเพิ่มนำไปวิเคราะห์การทำงานของระบบ โดยการรับ-ส่งเอกสารจะผ่าน Intermediary แต่ไม่ได้ทำการประมวลผลเอกสารแต่จะเก็บของข้อมูลบางอย่างไว้เท่านั้นเช่นนำไปวิเคราะห์ค่า QoS (Quality of Service) หรือ Bottleneck ของระบบ เป็นต้น

#### Intermediaries ใน SOAP

ในหัวข้อก่อนหน้านี้ได้อธิบายหลักการของการใช้ Intermediaries ไปแล้ว ดังนั้นในส่วนนี้จะกล่าวถึงว่าจะทราบได้อย่างไรว่าเอกสาร SOAP นั้นใครจะเป็นผู้ประมวลผล (ซึ่งจะอยู่ในส่วนของ <Body> Element ของเอกสาร)

ดังรูปที่แสดงการทำงานของ Intermediaries จะเห็นว่าเอกสาร SOAP จะถูกส่งจากผู้ขอใช้บริการผ่าน Intermediaries เป็นไปจนถึงปลายทางคือผู้ให้บริการ ดังนั้นเอกสาร SOAP ก็ถูกเปิดเผยทั้งหมดกับ Intermediaries ด้วย แต่ส่วนที่ Intermediaries สนใจจะอยู่เฉพาะส่วนที่เป็น Header เท่านั้น โดยใน Header จะมี Attribute หนึ่งเพื่อใช้ในการระบุว่าใครที่ควรจะทำางกับ Header นี้ ซึ่ง Attribute นั้นก็คือ actor โดยค่าที่ actor จะต้องใส่ก็คือ URI (Universal Resource Indicator) ของ Intermediary นั้นเอง ดังตัวอย่าง

```
<soap:Header>
  <i:Authentication
    xmlns:i="http://www.i3t.or.th/ws/security"
    soap:actor="urn:I3T:Security/GateWay"
    soap:mustUnderstand="1">
    <i:username>ACompany</i:username>
    <i:password>SOAPpath</i:password>
  </i:Authentication>
</soap:Header>
```

ดังนั้นเมื่อ Intermediary รับเอกสาร SOAP โดยถ้ามีชื่อ attribute actor ตรงกับของตัวเอง Intermediary จะเปลี่ยนเนื้อหาใน Header ที่ได้เพื่อส่งต่อไปยัง Intermediary อื่นหรือไปยังผู้

ให้บริการจุดประสงค์หลักที่มีการออกแบบการทำงานแบบนี้ เพื่อลดความซับซ้อนของเอกสารลง นอกจากนี้แล้วยังมี URI ของ attribute actor ชนิดพิเศษอีกหนึ่งค่าคือ "http://schemas.xmlsoap.org/soap/actor/next" เพื่อบอกว่าผู้รับเอกสาร SOAP ในจุดต่อไปจะเป็นผู้ที่ประมวลผลของเอกสาร ซึ่งกระบวนการนี้จะมีประโยชน์อย่างมากในการส่งเอกสาร SOAP เป็นทอดๆ (hop-by-hop)

### WSDL (Web Services Description Language)

WSDL (Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services ความต้องการของนิยามนี้เกี่ยวข้องกับความต้องการของ distributed system ที่จะกำหนด Interface Definition Language (IDL) โดยใช้ภาษา XML, WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัทไอบีเอ็ม, SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ใน ปัจจุบันคือ WSDL 1.1 (รายละเอียดเพิ่มเติมเกี่ยวกับ WSDL สามารถศึกษาเพิ่มเติมได้ที่ <http://www.w3c.org/TR/wsdl>)

WSDL คือ มาตรฐานสำหรับการประกาศ process ที่จำเป็นในการเรียกใช้เซอร์วิส SOAP (Simple Object Access Potocal)

### โครงสร้างเอกสาร WSDL

WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) version ที่มีอยู่ในปัจจุบัน คือ WSDL 1.1 ในการใช้งานจริง หากเราสร้างบริการ Web Services ก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับ Web Services อย่างอัตโนมัติ จุดภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของ Web Services มีจุดที่ควรรู้ ดังนี้

Element	Definition
<portType>	เป็นส่วนที่สำคัญที่สุดใน WSDL element อธิบาย operations ที่ web service มีให้บริการ และ messages ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียน โปรแกรม
<operation>	อธิบาย method ที่ให้บริการ Web Services หนึ่งจะมี method จำนวนกี่ method ก็ได้
<message>	อธิบาย data elements ของ operation แต่ละ message อาจมีมากกว่าหนึ่งส่วนเทียบได้กับ parameter ของ function ในการเขียน โปรแกรม
<types>	อธิบายชนิดข้อมูลที่ web service ใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล

<binding>	อธิบาย format ของ message และ protocol details ในแต่ละ port
<service>	สำหรับ web server จะมี Web Services จำนวนกี่บริการก็ได้ และ ชื่อ Web Services ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน

ตามทฤษฎีแล้ว ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการ Web Services ได้มากกว่า 1 บริการโดยแต่ละ Web Services จะมี port สื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

## WSDL Example

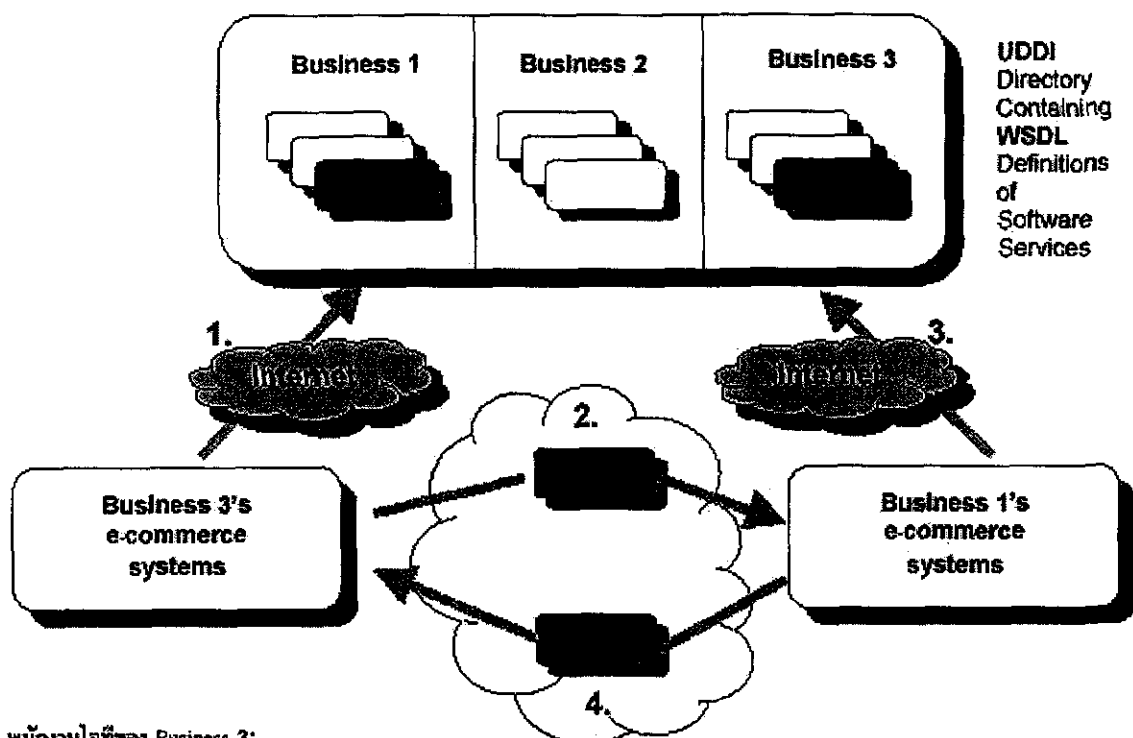
```

<wsdl:definitions name="nmtoken"? targetNamespace="uri">
  <import namespace="uri" location="uri"/> *
  <wsdl:documentation .... /> ?
  <wsdl:types> ?
    <wsdl:documentation .... /> ?
    <xsd:schema .... /> *
  </wsdl:types>
  <wsdl:message name="ncname"> *
    <wsdl:documentation .... /> ?
    <part name="ncname" element="qname"? type="qname"?/> *
  </wsdl:message>
  <wsdl:portType name="ncname"> *
    <wsdl:documentation .... /> ?
    <wsdl:operation name="ncname"> *
      <wsdl:documentation .... /> ?
      <wsdl:input message="qname"> ?
        <wsdl:documentation .... /> ?
      </wsdl:input>
      <wsdl:output message="qname"> ?
        <wsdl:documentation .... /> ?
      </wsdl:output>
      <wsdl:fault name="ncname" message="qname"> *
        <wsdl:documentation .... /> ?
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:serviceType name="ncname"> *
    <wsdl:portType name="qname"/> +
  </wsdl:serviceType>
  <wsdl:binding name="ncname" type="qname"> *
    <wsdl:documentation .... /> ?
    <!-- binding details --> *
    <wsdl:operation name="ncname"> *
      <wsdl:documentation .... /> ?
      <!-- binding details --> *
      <wsdl:input> ?
        <wsdl:documentation .... /> ?
        <!-- binding details -->
      </wsdl:input>
      <wsdl:output> ?
        <wsdl:documentation .... /> ?
        <!-- binding details --> *
      </wsdl:output>
      <wsdl:fault name="ncname"> *
        <wsdl:documentation .... /> ?
        <!-- binding details --> *
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ncname" serviceType="qname"> *
    <wsdl:documentation .... /> ?
    <wsdl:port name="ncname" binding="qname"> *
      <wsdl:documentation .... /> ?
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

### UDDI (Universal Description, Discovery and Integration protocol)

UDDI (Universal Description, Discovery, and Integration) เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI ใช้สำหรับค้นหา Service ที่ต้องการและเมื่อได้มาแล้ว UDDI ยังจัดหาข้อตกลงในวิธีการที่จะใช้งานเปรียบได้กับสมุดหน้าเหลือง เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟต์ และบริษัทอริบา (Ariba) ปัจจุบันมีบริษัทที่ร่วม กันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability



พนักงานไอทีของ Business 3:

1. ค้นหาข้อกำหนดการให้บริการใน WSDL ผ่านอินเทอร์เน็ตของ Business 1
2. ติดต่อผ่านอินเทอร์เน็ตของแอปพลิเคชันที่ให้บริการ (ด้วย SOAP) ผ่านทางอินเทอร์เน็ต
- 3,4. Business 1 รวมหรือบูรณาการกับ Business 3 ด้วยวิธีเดียวกัน

### รูปที่ 2.5 แสดง Web Services with SOAP, UDDI และ WSDL

จากมาตรฐานทั้ง 4 อย่างที่กล่าวข้างต้นสามารถสรุปลำดับขั้นตอนการทำงานของ Web Services ได้ดังนี้

1. Provider จัดทำระบบหรือบริการที่เป็น Web Services ขึ้นมา
2. ทำการลงทะเบียน Web Services กับหน่วยงานที่ให้บริการระบบ UDDI (หรือ Registry)
3. นำ WSDL ไฟล์ไปไว้ในระบบ UDDI ที่ได้ลงทะเบียนไว้
4. Requestor ทำการค้นหาระบบหรือบริการที่ต้องการจากระบบ UDDI

5. เมื่อ Requestor ได้พบระบบหรือบริการที่ต้องการจะนำไฟล์ WSDL ไปเรียนรู้วิธีการเรียกใช้ผ่านระบบของตน
6. Requestor ทำการติดต่อและเรียกใช้ระบบหรือบริการจาก Provider ได้โดยตรงผ่าน SOAP ในระบบของตน

การทำเว็บเซอร์วิส เป็นการแยกส่วนของข้อมูล (Data) และวิธีการทำงาน (Method) ออกจากกัน โดยที่ 1 เว็บเซอร์วิสอาจจะประกอบไปด้วย 1 วิธีการทำงาน หรือมากกว่าก็ได้ขึ้นอยู่กับความต้องการของเรา

ข้อดีอย่างหนึ่งของการแยกข้อมูล และวิธีการทำงานออกจากกันก็คือ จะช่วยให้เราสามารถปรับปรุง แก้ไขวิธีการทำงาน โดยที่เราไม่ต้องไปยุ่งเกี่ยวกับส่วนของข้อมูลแต่อย่างใด

คำว่า วิธีการทำงาน เปรียบเทียบได้ว่า การติดต่อกับฐานข้อมูล ก็ถือเป็น 1 วิธีการ, การแสดงเวลาในฝั่งเซิร์ฟเวอร์ ก็ถือเป็น 1 วิธีการ, การแสดงข้อความต่างๆ ก็ถือเป็น 1 วิธีการ, การบวกเลข ก็ถือเป็นอีก 1 วิธีการ ฯลฯ จะเห็นได้ว่าวิธีการทำงานดังกล่าวขึ้นอยู่กับว่า เว็บเซอร์วิสของเรามีจุดประสงค์ในการทำงานในด้านใด และเมื่อรวมกับการทำงานผ่านเว็บ จึงถูกเรียกว่า เว็บเซอร์วิส

ถ้าพิจารณาลักษณะการทำงานและความสามารถของ Web Services แล้ว เราสามารถสรุปได้ว่า Web Services ก็คือ วิวัฒนาการอีกก้าวหนึ่งของ Web Application นั่นเอง

#### 2.3.4 การประยุกต์ใช้ Web Services กับธุรกิจยุคใหม่

เราสามารถนำ Web Services มาประยุกต์ใช้ในธุรกิจ เพื่อตอบสนองต่อความต้องการบริการแบบ One Stop Service ได้อย่างง่ายดาย เนื่องจาก Web Services มีความสามารถในการแลกเปลี่ยน ข้อมูลและประสานการทำงานบนระบบที่ต่างกัน ได้ โดยการติดต่อกับผู้ใช้ผ่านทาง Web Application ซึ่งการประยุกต์ใช้ในลักษณะนี้ผู้ใช้ จะรู้เพียงว่า เมื่อเข้ามาที่เว็บไซต์ของเรา ก็จะได้รับบริการที่ครบวงจรเช่นบริการการท่องเที่ยว, บริการสถานที่พักผ่อนและสันทนาการ เป็นต้น

ในเรื่องของความปลอดภัย เนื่องจาก Web Service ทำงานอยู่บน Internet ซึ่งปัจจุบันมีเทคโนโลยีในการรักษาความปลอดภัยมากมายรองรับอยู่แล้ว และ Web Services สามารถผ่านระบบรักษาความปลอดภัย (Firewall) ได้เนื่องจาก SOAP ถูกส่งโดยผ่านโปรโตคอล HTTP นอกจากนี้ Web Services ยังมีระบบรักษาความปลอดภัยตามมาตรฐานของ PKI (Public Key Infrastructure) เช่น MD5 (Message Digest), SSL (Secure Socket Layer) และ PGP (Pretty Good

Privacy) ทำให้มั่นใจได้ว่าการใช้ Web Services เป็นเครื่องมือธุรกิจมีมาตรฐานในการรักษาความปลอดภัยรองรับ

ส่วนเครื่องมือที่ใช้ในการพัฒนา Web Services ก็มีอยู่มากมาย เนื่องจาก Web Services เป็น มาตรฐานที่พัฒนาได้ง่าย เพราะอยู่ในรูปแบบของข้อมูลตัวอักษรหรือ ภาษา XML ทำให้มีชุดเครื่องมือที่สนับสนุนการพัฒนา Web Services มากมาย ซึ่งสามารถแบ่งออกเป็น 4 กลุ่มใหญ่ได้ ดังนี้

1. ชุดเครื่องมือที่สนับสนุนโดย Microsoft ตาม Platform ของ Microsoft .NET

Framework

2. ชุดเครื่องมือที่สนับสนุนโดย Sun Microsystems ตาม Platform ของ Sun ONE (Sun

Open

Net Environment)

3. ชุดเครื่องมือที่สนับสนุนโดย IBM เช่น Web Services Toolkit

4. ชุดเครื่องมืออื่นๆที่สนับสนุน SOAP, XML ทั้งที่เป็น Commercial Product และ Open Source

## 2.4 เปรียบเทียบระหว่าง Web Application และ Web Services

เมื่อเราเข้าใจความหมายและการทำงาน Web Application และ Web Services แล้ว จะเห็นว่าเครื่องมือทั้งสองต่างใช้ HTTP Protocol หรือ อินเทอร์เน็ต เป็นช่องทางในการสื่อสารเหมือนกัน แต่มีวัตถุประสงค์ต่างกัน โดย Web Application ใช้เพื่อการแลกเปลี่ยนไฟล์ HTML ระหว่างเว็บเซิร์ฟเวอร์ แต่ Web Services เป็นการแลกเปลี่ยน “บริการ” (ก้อนโปรแกรม Software Components) ระหว่างระบบสารสนเทศผ่านเว็บเซิร์ฟเวอร์

ในเรื่องของความสามารถโดยส่วนใหญ่จะใช้ Web Application ในการติดต่อกับผู้ใช้ผ่านทาง Internet browser เพื่อนำเสนอข้อมูลและการทำงานต่างๆ ส่วน Web Services จะทำหน้าที่ในการติดต่อกับ เว็บเซิร์ฟเวอร์ เพื่อแลกเปลี่ยนข้อมูลและการทำงานหรือใช้บริการข้ามระบบกันโดยใช้ Web Application หรือ Application Interface ในการติดต่อกับผู้ใช้ นอกจากนี้ Web Services ยังสามารถทำงานกับระบบต่างๆ ได้มากกว่า 1 ระบบ ในขณะที่ Web Application ไม่สามารถทำได้โดยตรง ซึ่งสามารถสรุปการเปรียบเทียบได้ดังตาราง

หัวข้อ	Web Service	Web Application
การเชื่อมต่อ	program – program	human – program
ภาษาที่ใช้	XML	HTML
รายชื่อการให้บริการ	ค้นผ่าน UDDI	ค้นหาผ่าน search engine
ขอบเขตการใช้งาน	Business- to-Business (B2B)	Business-to-Customer (B2C)
โปรโตคอล (Protocol)	SOAP+HTTP	HTTP

ตารางที่ 1 แสดงการเปรียบเทียบระหว่าง Web Services กับ Web Application

หลังจากที่ได้ทำความเข้าใจความหมายและข้อเปรียบเทียบระหว่าง Web Application และ Web Services แล้ว ความแตกต่างระหว่างเครื่องมือทั้งสองจะเห็นได้ชัดในเรื่องของความสามารถ แต่ในการนำไปใช้งานจำเป็นจะต้องประยุกต์ใช้เครื่องมือทั้งสองชนิดร่วมกัน เพื่อให้เกิดประสิทธิภาพมากที่สุด

## บทที่ 3

# การออกแบบโปรแกรม

### 3.1 ความต้องการของระบบ

โครงการนี้มีจุดประสงค์เพื่อพัฒนาโปรแกรมประยุกต์สำหรับระบบบริหารงานโรงพยาบาล แผนกผู้ป่วยใน โดยในส่วนของระบบจะมีฐานข้อมูลซึ่งกรอกรการทำงานจะอยู่ในรูปแบบของแพทย์ผู้ให้การรักษาเข้ามาภายในห้องคนไข้แล้วก็จะใช้ Pocket PC 1 ตัวที่ใช้สำหรับบันทึกค่าที่ได้จากการตรวจผ่านทางหน้าจอ แล้วจากนั้นข้อมูลที่ได้ก็จะนำไปบันทึกลงฐานข้อมูลที่เก็บอยู่ในเซิร์ฟเวอร์ผ่านทางเครือข่ายไร้สาย โดยเมื่อแพทย์ต้องการยาหรือเวชภัณฑ์ก็จะส่งผ่าน Pocket PC ไปแสดงยังเครื่องของพยาบาลที่ห้องพยาบาล จากนั้นพยาบาลก็จะทำตามกระบวนการแล้วนำจ่ายยาให้แพทย์ตามที่ระบุไปใน message ซึ่งเมื่อทำงานไหนเสร็จแล้ว ก็จะมีการระบุสถานะว่าทำเสร็จแล้ว

พนักงานของโรงพยาบาลทุกคนจะได้รับ Username และ Password จากผู้ดูแลระบบ (Admin) ซึ่งพนักงานเหล่านี้สามารถล็อกอินเข้าไปเช็คข้อมูลของตนเอง รวมไปถึงสามารถดูข้อมูลคนไข้ที่เป็นคนไข้ที่ตนเองดูแลได้ผ่านทางหน้าเว็บที่ทางโรงพยาบาลได้จัดทำขึ้นมา ซึ่งจะช่วยให้แพทย์และพยาบาลสามารถทำงานได้รวดเร็วขึ้นเนื่องจากได้ทราบข้อมูลล่วงหน้า

ในส่วนของบุคคลภายนอก สามารถเข้ามาใช้บริการของโรงพยาบาลได้ เช่น เว็บไซต์ตอบคำถาม อ่านข่าวสารที่โรงพยาบาลได้จัดทำขึ้น เป็นต้น

### 3.2 อุปกรณ์ที่ใช้ในการพัฒนาโครงการ

อุปกรณ์ทางฮาร์ดแวร์ได้แก่

- Computer CPU Pentium 4 2.66 GHz, RAM 1 GB, Harddisk 40 GB
- Notebook CPU centino mobile 1.5 GHz, RAM 768 MB, Harddisk 40 GB
- Printer & Scanner
- Pocket PC

อุปกรณ์ทางซอฟต์แวร์ได้แก่

- OS : Microsoft Windows XP service pack 2
- DBMS : MySQL
- Microsoft Visual Studio.Net
- Rational Rose (UML) ใช้ใช้ในการออกแบบระบบวิเคราะห์ระบบ ซึ่งมี Use Case Diagram และ Sequence Diagram
- Notepad และ Editplus เป็นโปรแกรมช่วยในการออกแบบ HTML



ตาราง **USER\_PROFILE** เก็บรายชื่อและรหัสผ่านของผู้ใช้

ID คือ ลำดับที่ของผู้ใช้

FIRST\_NAME คือ ชื่อของผู้ใช้

LAST\_NAME คือ นามสกุลของผู้ใช้

USER\_NAME คือ ชื่อผู้ใช้ที่ใช้ในการล็อกอิน

PASSWORD คือ รหัสที่ใช้ในการล็อกอิน

STATUS คือ สถานะของผู้ใช้ (0 = Admin , 1 = พนักงาน รพ. , 2 = ผู้เยี่ยมชมเว็บไซต์ )

Field	Type	Attribute	Null	Key	Default	Extra
ID	Int(11)		No	Primary		Auto_increment
FIRST_NAME	Varchar(50)		No			
LAST_NAME	Varchar(50)					
USER_NAME	Varchar(20)					
PASSWORD	Varchar(32)					
STATUS	Char(1)					



CatID	CatName
-------	---------

ตาราง **CATEGORIES** เก็บรหัส และรายชื่อหมวดหมู่ความชำนาญการทางการแพทย์

CatID คือ รหัสของหมวดหมู่ความชำนาญการทางการแพทย์

CatName คือ รายชื่อหมวดหมู่ความชำนาญการทางการแพทย์

Field	Type	Attribute	Null	Key	Default	Extra
CatID	Int(10)	Unsigned	No	Primary		Auto_increment
CatName	Varchar(50)		No			

TABLE TBLDOCTOR

DoctorID	DOCTORNAME	ADDRESS	CatID	Picture
		Email	Phone	MobilePhone

ตาราง TBLDOCTOR เก็บรายละเอียดของพนักงานโรงพยาบาล

DoctorID คือ รหัสประจำตัวพนักงานโรงพยาบาล

DoctorName คือ ชื่อของพนักงานโรงพยาบาล

Address คือ ที่อยู่ของพนักงานที่สามารถติดต่อได้

Picture คือ รูปของพนักงาน

MobilePhone คือ เบอร์โทรศัพท์มือถือ

Phone คือ เบอร์โทรศัพท์บ้าน

Email คือ ที่อยู่อีเมลของพนักงานที่สามารถติดต่อได้

CatID คือ รหัสของหมวดหมู่ความชำนาญการทางการแพทย์  
(ซึ่งใช้เชื่อมโยงกับตาราง CATEGORIES)

Field	Type	Attribute	Null	Key	Default	Extra
DoctorID	Varchar(10)		No	Primary		
DoctorName	Varchar(50)		No			
Address	Varchar(100)		Yes		Null	
CatID	Int(10)	Unsigned	Yes	Foreign	Null	
Picture	Blob	Binary	Yes		Null	
MobilePhone	Varchar(15)		Yes		Null	
Phone	Varchar(15)		Yes		Null	
Email	Varchar(30)		Yes		Null	



UseCode	UseDetail	UsePerDay
---------	-----------	-----------

**ตาราง TBLUSE**      เก็บวิธีใช้ยาและเวชภัณฑ์

UseCode              คือ รหัสการใช้ยา

UseDetail            คือ รายละเอียดการใช้ยา

UsePerDay           คือ ปริมาณการใช้ต่อวัน

Field	Type	Attribute	Null	Key	Default	Extra
UseCode	Varchar(20)		No	Primary		
UseDetail	Varchar(200)		No			
UsePerDay	Varchar(50)		No		0	



DgID	DgName	UseCode	Properties	Concern
------	--------	---------	------------	---------

**ตาราง TBLDRUG**      เก็บทะเบียนยาและเวชภัณฑ์

DgID                    คือ รหัสยา

DgName                คือ ชื่อยา

UseCode                คือ รหัสการใช้ยา

Properties              คือ สรรพคุณทางยา

Concern                คือ รายละเอียดที่เกี่ยวข้อง เช่น ต้องกินให้หมด หรืออาจง่วง เป็นต้น

Field	Type	Attribute	Null	Key	Default	Extra
DgID	Varchar(20)		No	Primary		
DgName	Varchar(50)		No			
UseCode	Varchar(20)		No	Foreign		
Properties	Varchar(100)		Yes		Null	
Concern	Varchar(100)		Yes		Null	



PatientID	IDCarD	Sex	FName	Name	LName	Birthday
	Occupation	MarritalStatus	Weight	Height	BloodGroup	
	MobilePhone	Phone	Present_Address	Registered_Address		
	DoctorID	HereditiyID	AllertCollectID	ContactID		

**ตาราง TBLPATIENTBASICDATA** เก็บรายละเอียดเบื้องต้นของผู้ป่วย

PatientID	คือ รหัสผู้ป่วย
IDCard	คือ เลขที่บัตรประชาชน
Sex	คือ เพศ
FName	คือ ชื่อ
Name	คือ ชื่อเต็ม
LName	คือ นามสกุล
Birthday	คือ วันเกิด
BloodGroup	คือ กรุ๊ปเลือด
Height	คือ ส่วนสูง
Weight	คือ น้ำหนัก
MarritalStatus	คือ สถานภาพการสมรส
Occupation	คือ อาชีพ
Registered_Address	คือ ที่อยู่ตามที่ระบุในทะเบียนบ้าน
Present_Address	คือ ที่อยู่ปัจจุบัน
Phone	คือ เบอร์โทรศัพท์บ้าน

MobilePhone	คือ เบอร์โทรศัพท์มือถือ
ContactID	คือ รหัสของญาติที่เกี่ยวข้องกับผู้ป่วย
AllertCollectID	คือ รหัสของการแพ้ เช่น แพ้ยา อาหารทะเล เป็นต้น
HeredityID	คือ โรคทางกรรมพันธุ์
DoctorID	คือ รหัสพนักงานประจำโรงพยาบาล

Field	Type	Attribute	Null	Key	Default	Extra
PatientID	Varchar(10)		No	Primary		
IDCard	Varchar(20)		No			
Sex	Varchar(10)		No			
FName	Varchar(20)		No			
Name	Varchar(70)		No			
LName	Varchar(20)		No			
Birthday	Varchar(20)		No			
BloodGroup	Varchar(5)		No			
Height	Varchar(5)		No		0	
Weight	Varchar(5)		No		0	
MarritalStatus	Varchar(20)		No			
Occupation	Varchar(50)		No			
Registered_Address	Varchar(100)		No			
Field	Type	Attribute	Null	Key	Default	Extra
Present_Address	Varchar(100)		No			
Phone	Varchar(20)		Yes		Null	
MobilePhone	Varchar(20)		Yes		Null	
ContactID	Varchar(10)		No	Foreign		
AllertCollectID	Varchar(10)		Yes	Foreign	Null	
HeredityID	Varchar(10)		Yes	Foreign	Null	
DoctorID	Varchar(10)		No	Foreign		



AlertCollectID	AlertCollectName
----------------	------------------

**ตาราง TBLALLERTCOLLECT** เก็บรหัสและข้อมูลการแพ้ยาหรืออาหาร

AlertCollectID คือ รหัสการแพ้ยาหรืออาหาร

AlertCollectName คือ ชื่อของการแพ้ยาหรืออาหาร

Field	Type	Attribute	Null	Key	Default	Extra
AlertCollectID	Varchar(10)		No	Primary		
AlertCollectName	Varchar(100)		No			



CauseExitID	CauseExitName
-------------	---------------

**ตาราง TBLCAUSEEXIT** เก็บรหัสและสาเหตุการออกจากโรงพยาบาล

CauseExitID คือ รหัสของสาเหตุการออก

CauseExitName คือ สาเหตุการออก

Field	Type	Attribute	Null	Key	Default	Extra
CauseExitID	Varchar(10)		No	Primary	0	
CauseExitName	Varchar(50)		No			

**TBLCAUSE SICK**

CauseSickID	CauseSickName
-------------	---------------

**ตาราง TBLCAUSESICK** เก็บรหัสและสาเหตุที่คนไข้ป่วย

CauseSickID คือ รหัสของสาเหตุการป่วย

CauseSickName คือ สาเหตุการป่วย

Field	Type	Attribute	Null	Key	Default	Extra
CauseSickID	Varchar(10)		No	Primary	0	
CauseSickName	Varchar(50)		No			

**TBLDISEASE**

DiseaseID	DiseaseName	DGroupID
-----------	-------------	----------

**ตาราง TBLDISEASE** เก็บรหัสและข้อมูลการแพ้ยาหรืออาหาร

DiseaseID คือ รหัสโรค

DiseaseName คือ ชื่อโรค

DGroupID คือ รหัสประเภทโรค

(เอาไว้เชื่อมโยงกับตาราง TBLDISEASEGROUP)

Field	Type	Attribute	Null	Key	Default	Extra
DiseaseID	Varchar(10)		No	Primary		
DiseaseName	Varchar(50)		No			
DGroupID	Varchar(10)			Foreign		

**TBLDISEASEGROUP**

DGroupID	DGroupName
----------	------------

**ตาราง TBLDISEASEGROUP** เก็บรหัสและชื่อของกลุ่มโรค

DGroupID                      คือ รหัสประเภทโรค

DGroupName                 คือ ประเภทโรค

Field	Type	Attribute	Null	Key	Default	Extra
DGroupID	Varchar(10)		No	Primary		
DGroupName	Varchar(50)		No			

**TBLFORMULAR**

FormularID	FormularName	FGroupID	DgID	UseCode
		Concern	Properties	Dose

**ตาราง TBLFORMULAR** เก็บรายละเอียดเกี่ยวกับยาเพิ่มเติมจากตาราง TBLDRUG

FormularID                 คือ รหัสของสูตรยา

FormularName             คือ ชื่อสูตรยา

FGroupID                 คือ รหัสกลุ่มสูตรยา

DgID                        คือ รหัสยา (เอาไว้เชื่อมโยงกับตาราง TBLDRUG)

UseCode                    คือ รหัสการใช้ยา

Dose                        คือ ปริมาณยาที่กำหนดให้กิน

Properties                 คือ สรรพคุณทางยา

Concern                    คือ รายละเอียดเพิ่มเติม เช่น ง่ายง่าย เป็นต้น

Field	Type	Attribute	Null	Key	Default	Extra
FormularID	Varchar(10)		No	Primary		
FormularName	Varchar(50)		No			
FGroupID	Varchar(10)		Yes	Foreign		
DgID	Varchar(10)		No	Foreign		
UseCode	Varchar(20)		No	Foreign		
Dose	Int(10)		Yes			
Properties	Varchar(100)		Yes		0	
Concern	Varchar(50)		Yes			



FGroupID	FGroup
----------	--------

ตาราง TBLFORMULARGROUP เก็บรหัสและประเภทกลุ่มสูตรยา

FGroupID คือ รหัสประเภทสูตรยา

FGroup คือ ประเภทสูตรยา

Field	Type	Attribute	Null	Key	Default	Extra
FGroupID	Varchar(10)		No	Primary		
FGroup	Varchar(100)		No			



HeredityID	HeredityName
------------	--------------

**ตาราง TBLHEREDITY** เก็บรหัสและชื่อโรคทางกรรมพันธุ์

HeredityID คือ รหัสโรคทางกรรมพันธุ์

HeredityName คือ ชื่อโรคทางกรรมพันธุ์

Field	Type	Attribute	Null	Key	Default	Extra
HeredityID	Varchar(10)		No	Primary		
HeredityName	Varchar(50)		No			



ContactID	FName	Name	LName	Address	Phone
-----------	-------	------	-------	---------	-------

**ตาราง TBLPATIENTFOLK** เก็บรายละเอียดของญาติผู้ป่วย

ContactID คือ รหัสของญาติที่เกี่ยวข้องกับผู้ป่วย

FName คือ ชื่อ

Name คือ ชื่อเต็ม

LName คือ นามสกุลญาติ

Address คือ ที่อยู่ของญาติผู้ป่วย

Phone คือ หมายเลขโทรศัพท์ของญาติผู้ป่วย

Field	Type	Attribute	Null	Key	Default	Extra
ContactID	Varchar(10)		No	Primary		
FName	Varchar(50)		No			
Name	Varchar(70)		No			
LName	Varchar(50)		No			
Address	Varchar(255)		No			
Phone	Varchar(50)		No			



RoomTypeID	RoomTypeName
------------	--------------

**ตาราง TBLROOM** เก็บรหัสและชื่อประเภทห้องพักรักษาตัว

RoomTypeID คือ รหัสประเภทห้อง

RoomTypeName คือ ชื่อประเภทห้อง

Field	Type	Attribute	Null	Key	Default	Extra
RoomTypeID	Varchar(10)		No	Primary		
RoomTypeName	Varchar(50)		No			



ListID	PatientID	Date	Time	Degree	Pulse
Respiration	Systolic	Diastolic	BloodPress	Stools	Urine
State	Diagnosis	NotEat	DoctorID	MedicationRecord	Status

**ตาราง TBLTESTRECORD** เก็บรายละเอียดการรักษาในปัจจุบัน

ListID คือ ลำดับการรักษา

PatientID คือ รหัสผู้ป่วย (เอาไว้เชื่อมโยงกับตาราง TBLEASEGROUP)

Date คือ วันที่ทำการรักษา

Time คือ เวลา

Degree คือ อุณหภูมิร่างกาย

Pulse คือ จังหวะการเต้นของหัวใจ

Respiration คือ จังหวะการหายใจ

Systolic คือ การบีบตัวของหัวใจ

Diastolic คือ การขยายตัวตามปกติของหัวใจ

BloodPress คือ ความดันโลหิต

Stools	คือ ถ่ายอุจจาระ
Urine	คือ ปัสสาวะ
State	คือ อาการ
Diagnosis	คือ วินิจฉัย
NotEat	คือ ของที่ห้ามรับประทาน
DoctorID	คือ รหัสแพทย์ที่ให้การรักษา (เอาไว้เชื่อมโยงกับตาราง TBLDOCTOR)
MedicationRecord	คือ รายการยา/เวชภัณฑ์ที่หมอสั่ง
Status	คือ สถานะรายการสั่งยา (0 = อ่านแล้ว, 1 = ยังไม่ได้อ่าน)

Field	Type	Attribute	Null	Key	Default	Extra
ListID	Int(10)		No	Primary		Auto_Increment
PatientID	Varchar(10)		No	Foreign		
Date	Varchar(10)		No			
Time	Varchar(10)		No			
Degree	Varchar(10)		No			
Pulse	Varchar(10)		No			
Respiration	Varchar(10)		No			
Systolic	Varchar(10)		No			
Diastolic	Varchar(10)		No			
BloodPress	Varchar(10)		No			
Stools	Varchar(10)		No			
Urine	Varchar(10)		No			
State	Varchar(100)		No			
Diagnosis	Varchar(100)		No			
NotEat	Varchar(50)		No			
DoctorID	Varchar(10)		No	Foreign		
MedicationRecord	Varchar(250)		No			
Status	Int(11)		No			

### TBLTREATHISTORY

ListID	PatientID	DoctorID	DiseaseID	RoomNo
	CauseExitID	CauseSickID	RoomTypeID	Bed
			DateExit	DateAdmission

#### ตาราง TBLTREATHISTORY เก็บรายละเอียดประวัติการรักษา

ListID	คือ ลำดับการรักษา
PatientID	คือ รหัสผู้ป่วย (เอาไว้เชื่อมโยงกับตาราง TBLPATIENTBASICDATA)
DoctorID	คือ รหัสแพทย์ผู้ให้การรักษา (เอาไว้เชื่อมโยงกับตาราง TBLDOCTOR)
DiseaseID	คือ รหัสโรค (เอาไว้เชื่อมโยงกับตาราง TBLDISEASE)
RoomNo	คือ หมายเลขห้องที่เข้ารับการรักษา
Bed	คือ หมายเลขเตียง
RoomTypeID	คือ รหัสประเภทห้อง (เอาไว้เชื่อมโยงกับตาราง TBLROOM)
CauseSickID	คือ รหัสสาเหตุการป่วย (เอาไว้เชื่อมโยงกับตาราง TBLCAUSESICK)
CauseExitID	คือ รหัสสาเหตุการออก (เอาไว้เชื่อมโยงกับตาราง TBLCAUSEEXIT)
DateAdmission	คือ วันที่เข้ารับการรักษา
DateExit	คือ วันที่ออกจากโรงพยาบาล

Field	Type	Attribute	Null	Key	Default	Extra
ListID	Int(10)		No	Primary		Auto_Increment
PatientID	Varchar(10)		No	Foreign		
DoctorID	Varchar(10)		No	Foreign		
DiseaseID	Varchar(50)		No	Foreign		

RoomNo	Varchar(10)		No			
Bed	Varchar(10)		No			
RoomTypeID	Varchar(10)		No	Foreign		
CauseSickID	Varchar(50)		No	Foreign		
CauseExitID	Varchar(50)		No	Foreign		
DateAdmission	Varchar(50)		No			
DateExit	Varchar(50)		No			

จากความต้องการของโครงการ ได้พัฒนา 2 โปรแกรมประยุกต์ดังนี้

- โปรแกรมบน Pocket PC (Mobile Computing Application)
- โปรแกรม Web Application

### 3.4 โปรแกรมบน Pocket PC (Mobile Computing Application)

#### 3.4.1 ขอบเขตการทำงานของโปรแกรม

- 1.) เป็นส่วนที่ติดต่อกับแพทย์โดยตรง (User Interface) สามารถที่จะดูข้อมูลการรักษา, ประวัติของผู้ป่วย และบันทึกข้อมูลการรักษาผู้ป่วยได้
- 2.) สามารถเชื่อมต่อกับฐานข้อมูลของโรงพยาบาลได้ เพื่อดึงข้อมูลและบันทึกข้อมูลอย่างถูกต้อง

#### 3.4.2 การออกแบบโปรแกรม

- 1.) ส่วนของการติดต่อกับฐานข้อมูลของระบบจะติดต่อโดยผ่านทาง Web Service  
โปรแกรมบน PocketPC จะทำการติดต่อกับฐานข้อมูลของระบบได้ ผ่าน Web Service ซึ่ง Web Service จะใส่คำสั่ง SQL ไว้บนคอมโพเนนต์ (Component) ที่มีการทำงานเกี่ยวกับการค้นหา การบันทึกลงฐานข้อมูล ตัวอย่างเช่น

- 1.1) การตรวจสอบการใช้งานของแพทย์ จะต้องกรอกชื่อ (Username) และรหัสผ่าน (Password)

```
"SELECT * FROM tbldoctor WHERE "+"Username = '"+usr+"AND Password =
'+pwd+'"
```

ซึ่ง usr คือตัวแปรที่เก็บค่าชื่อผู้ใช้งานที่ป้อนเข้ามาทาง PocketPC และ pwd คือตัวแปรที่เก็บค่ารหัสผ่านที่ป้อนเข้ามาทาง PocketPC เช่นเดียวกัน และ tbldoctor คือชื่อตารางในฐานข้อมูล ส่วน Username และ Password เป็น field ของตาราง tbldoctor

สำหรับในขั้นตอนการกรอกชื่อแพทย์ผู้ใช้และรหัสผ่านนี้ จะทำให้เกิดผลลัพธ์ได้ดังนี้

1. กรณีไม่กรอก Username และไม่กรอก Password
2. กรณีกรอก Username ถูกแต่ Password ผิด
3. กรณีกรอก Username ผิดแต่ Password ถูก
4. กรณีกรอก Username ถูก Password ถูก

ทั้ง 3 กรณีแรกแกรมจะแจ้งว่ามีข้อผิดพลาด และไม่สามารถเข้าใช้งานระบบได้ ส่วนในกรณีที่ 4 จึงจะสามารถเข้าใช้งานระบบได้

```
<?xml version="1.0" encoding="utf-8" ?>
<boolean xmlns="http://tempuri.org/">true</boolean>
```

แสดงผลลัพธ์ที่ Web Service ทำการส่งค่ากลับมากรณี Username และ Password ถูกต้อง

```
<?xml version="1.0" encoding="utf-8" ?>
<boolean xmlns="http://tempuri.org/">false</boolean>
```

แสดงผลลัพธ์ที่ Web Service ทำการส่งค่ากลับมากรณีที่ไม่ถูกต้อง

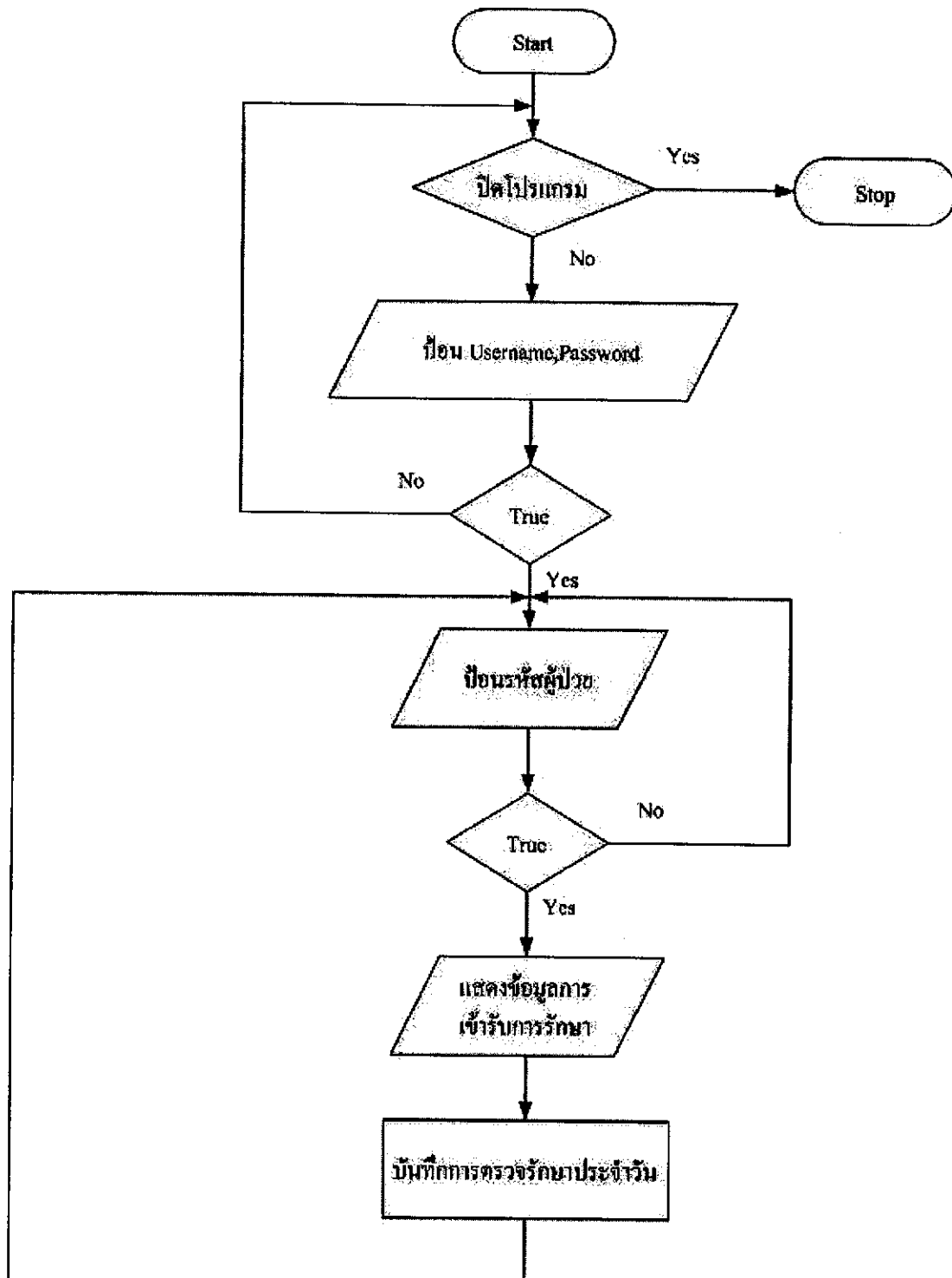
## 1.2) การบันทึกข้อมูลการตรวจรักษาลงฐานข้อมูล

```
"INSERT INTO tbltestrecord (PatientID,Date,Time)
VALUES ('"+PatientID+"','"+Date+"','"+Time+"")"
```

ซึ่ง PatientID,Date,Time คือ field ของตาราง tbltestrecord ในฐานข้อมูล ส่วน VALUES ('"+PatientID+"','"+Date+"','"+Time+"") เป็นตัวแปรที่เก็บค่าที่ป้อนเข้ามาทาง PocketPC ซึ่งจะบันทึกลงในตาราง tbltestrecord

ซึ่งการทำงานทั้งในส่วนของการดึงข้อมูลจากฐานข้อมูลออกมาดู และการบันทึกข้อมูลลงในฐานข้อมูลนี้ จะต้องทำผ่าน Web Service ทั้งหมด

FlowChart แสดงขั้นตอนการทำงานของ Pocket PC



รูปที่ 3.2 แสดงขั้นตอนการทำงานของ Pocket PC

## Service1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [GetRoomTypeName](#)
- [GetHistoryOrder](#)
- [GetPatient](#)
- [CheckUser](#)
- [CheckRoomNo](#)
- [GetHeredity](#)
- [GetDisease](#)
- [ClinicalRecord](#)
- [GetHistoryClinical](#)
- [CheckPatient](#)
- [GetAllertCollect](#)

รูปที่ 3.3 แสดงฟังก์ชันการติดต่อกับฐานข้อมูลทั้งหมด ทั้งการดึงข้อมูลและการบันทึกข้อมูล

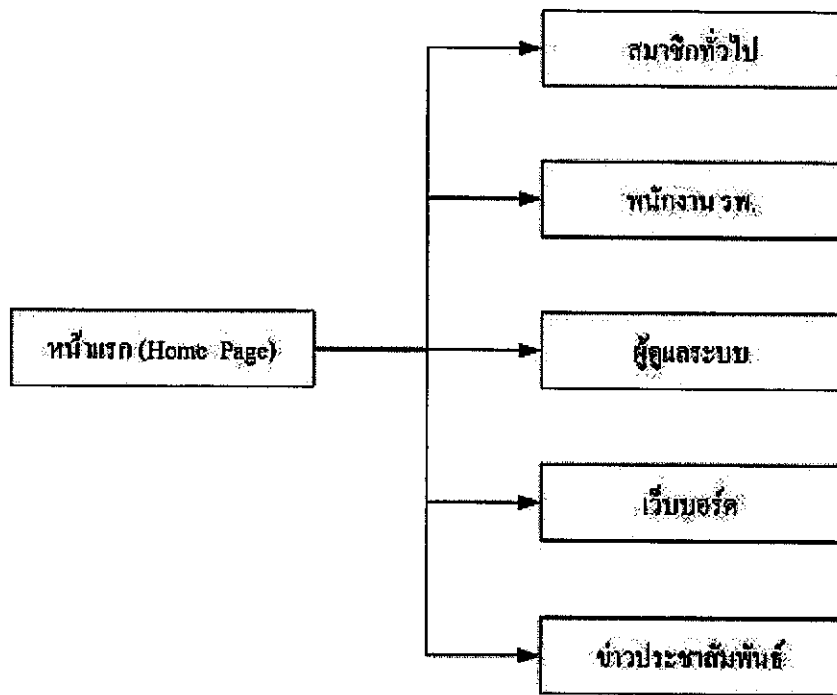
### 3.5 โปรแกรม Web Application

#### 3.5.1 ขอบเขตการทำงานของโปรแกรม

- 1.) บริการข้อมูลให้กับพนักงานผ่านทางกรลือกอินทางเว็บไซต์
- 2.) รับคำสั่งจาก Pocket PC กรณีต้องการยาหรือเวชภัณฑ์อื่นๆ
- 3.) บริการข่าวสารแก่บุคคลทั่วไปที่เข้ามาเยี่ยมชมเว็บไซต์ ผ่านทางเว็บบอร์ด และข่าวประชาสัมพันธ์
- 4.) แพทย์หรือพยาบาลสามารถค้นหาข้อมูลที่สำคัญต่อการรักษา จากฐานข้อมูลผ่านทางหน้าเว็บเพจได้ในส่วนที่โรงพยาบาลกำหนดเอาไว้

#### 3.5.2 การออกแบบโปรแกรม

การทำงานของเว็บไซต์นี้จะเริ่มจากหน้าแรก ในส่วนนี้ผู้ใช้ทั่วไปสามารถที่จะเข้ามาเยี่ยมชมเว็บไซต์ได้ แต่ในการทำงานบางส่วน ผู้ใช้จะต้องลงทะเบียนเพื่อที่จะเข้าไปเยี่ยมชม ซึ่งส่วนที่ต้องทำการล็อกอินเข้าไปดูข้อมูลได้ก็ ได้แก่ ในส่วนของพนักงานโรงพยาบาลสามารถล็อกอินเข้าไปดูเมนูต่างๆ ที่ได้จัดทำไว้ให้เฉพาะสำหรับพนักงาน และส่วนของผู้ดูแลระบบ (admin) ซึ่งจะทำงานในส่วนของการเพิ่ม แก้ไข ลบ ค้นหา ข้อมูลพนักงาน การจัดการเกี่ยวกับข่าวสาร และจัดหมวดหมู่ความเชี่ยวชาญได้ ซึ่งมีโครงสร้างของเว็บไซต์ดังรูปที่ 3.3 และมีขั้นตอนการทำงานในส่วนที่ต้องล็อกอินดังรูปที่ 3.4



รูปที่ 3.4 แสดงโครงสร้างของเว็บไซต์

### 3.5.2.1 ภาพรวมของระบบ Web Application

ประกอบไปด้วย การล็อกอินโดยผู้ใช้ การจัดการเกี่ยวกับผู้ป่วย และการจัดการเกี่ยวกับรายละเอียดของพนักงาน

การล็อกอินของผู้ใช้ระบบ ผู้ที่ล็อกอินจะต้องได้รับความเป็นสมาชิกก่อนถึงจะสามารถเยี่ยมชมเว็บได้ เพราะบางเว็บเพจอนุญาตให้เฉพาะที่เป็นสมาชิกเท่านั้นที่สามารถเยี่ยมชมได้ โดยการป้อนชื่อและรหัสผ่าน

#### การทำงานของระบบล็อกอินผู้ใช้ระบบมีดังนี้

ตรวจสอบผู้ใช่ว่าเป็นผู้ดูแลระบบหรือไม่ โดยรับชื่อและรหัสผ่านไปตรงกับตาราง user\_profile ว่ามีข้อมูลหรือไม่ ถ้าใช่ก็ไปทำงานในส่วนของผู้ดูแลระบบ หากไม่ก็ตรวจสอบต่อไปอีกว่าผู้ใช้เป็นพนักงานหรือไม่ ถ้าใช่ก็ทำในส่วนของพนักงาน ถ้าไม่ใช่ก็ถามต่อไปอีกว่าเป็นสมาชิกหรือไม่ ถ้าใช่ก็สามารถเข้าชมเว็บในส่วนที่จัดไว้สำหรับสมาชิกทั่วไปสามารถดูได้ ถ้าไม่ใช่ก็ถามต่อไปอีกว่าต้องการสมัครสมาชิกหรือไม่ ถ้าสมัครก็ทำการลงทะเบียน หากพบข้อมูลแสดงว่าผู้ใช้งานลงทะเบียนเรียบร้อยแล้ว ระบบจะเก็บชื่อไว้ในตัวแปร session

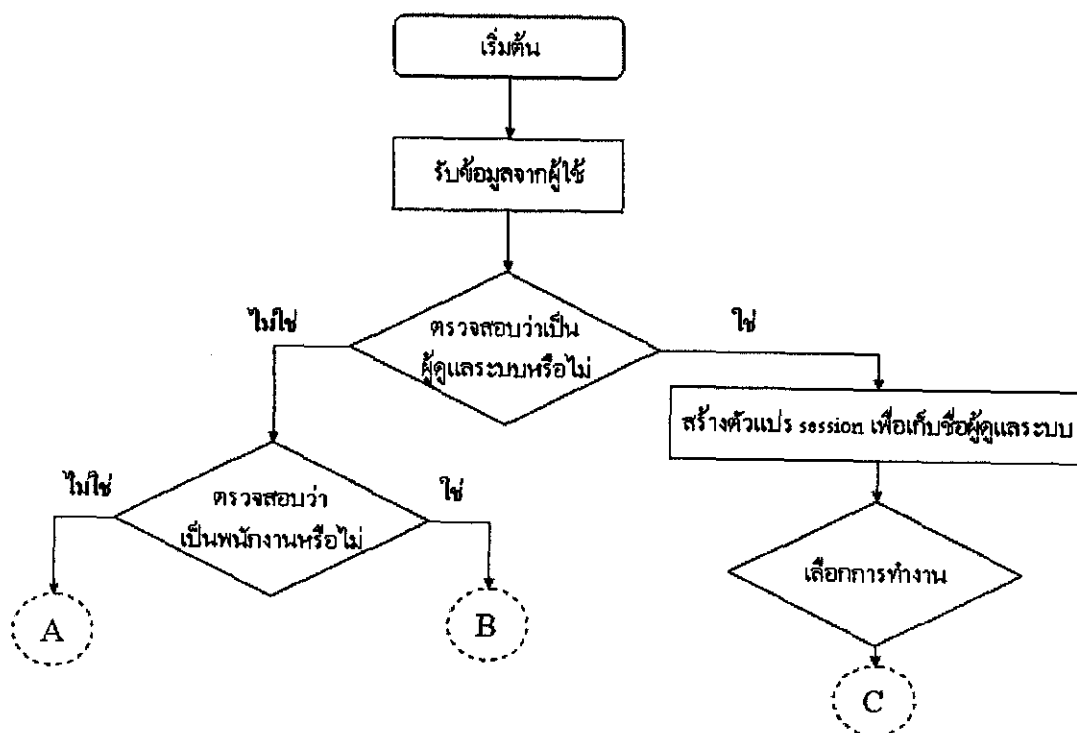
ข้อมูลที่ใช้ในการลงทะเบียนจะถูกเก็บลงฐานข้อมูล หากลืมรหัสผ่าน ก็เข้าไปที่เมนูลืมรหัสผ่านจากนั้นก็ป้อนคำตอบที่ได้ทำไว้ตอนลงทะเบียน ระบบจะนำคำตอบไปตรวจสอบกับ

ฐานข้อมูล ว่าชื่อผู้ใช้กับคำตอบตรงกันหรือไม่ หากตรงกันจะอนุญาตให้เปลี่ยนรหัสผ่านได้ เมื่อต้องการออกจากระบบก็คลิกที่เมนูออกจากระบบ ระบบจะทำลายตัวแปร session ที่สร้างไว้ในตอนที่ผู้ใช้ล็อกอินเข้ามา

#### ภาพการทำงาน

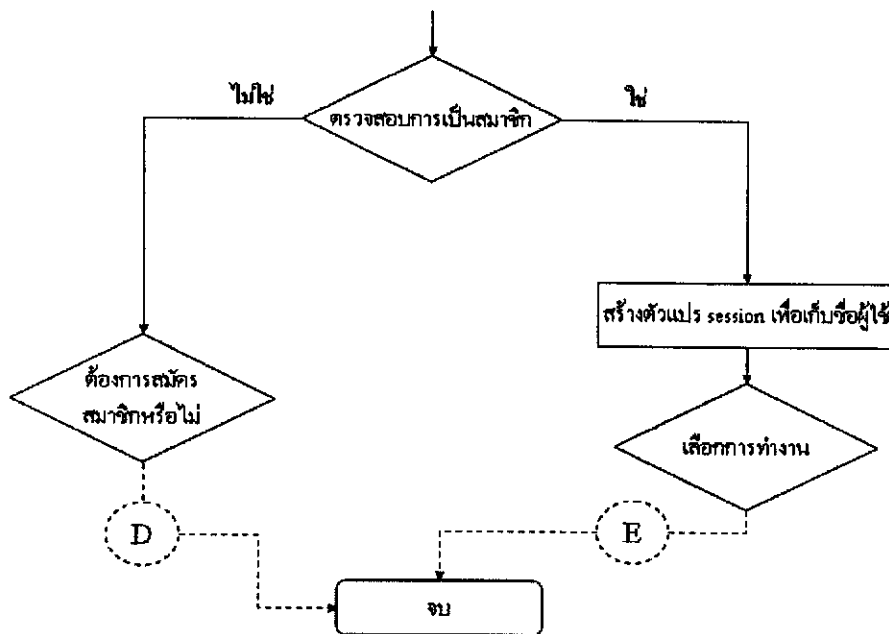
จากรูปจะเห็นว่ามียุคที่เชื่อมโยงไปยังส่วนต่างๆ โดยแต่ละจุดจะเชื่อมโยงไปยังแต่ละส่วนดังนี้

- ส่วน A เชื่อมโยงไปยังกรณีที่ไม่ใช่พนักงาน
- ส่วน B เชื่อมโยงไปยังกรณีที่เป็นพนักงาน
- ส่วน C เชื่อมโยงไปยังส่วนของการทำงานของผู้ดูแลระบบ
- ส่วน D เชื่อมโยงไปยังส่วนของการสมัครสมาชิก
- ส่วน E เชื่อมโยงไปยังส่วนของการทำงานของสมาชิกทั่วไป
- ส่วน F เชื่อมโยงไปยังส่วนตรวจสอบผู้ใช้และคำถามกรณีลืมรหัสผ่าน



รูปที่ 3.5 แสดงขั้นตอนการตรวจสอบผู้ใช้

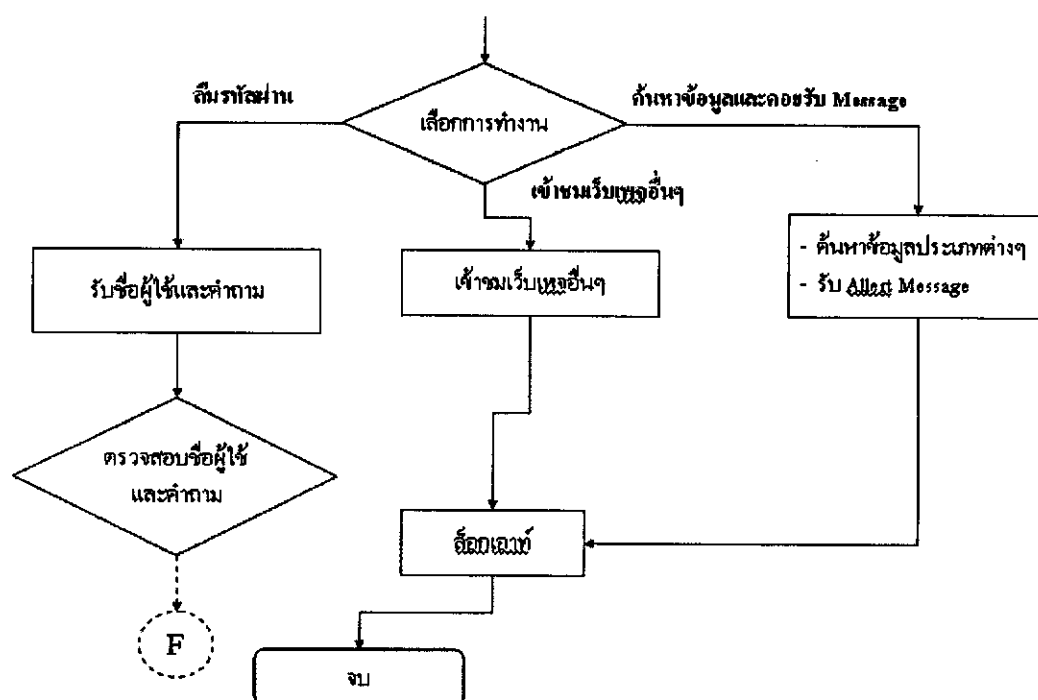
## ส่วน A กรณีไม่ใช้พนักงาน



รูปที่ 3.6 แสดงการตรวจสอบการเป็นสมาชิกทั่วไป

จากรูปที่ 3.6 หากผู้ที่ทำการล็อกอินนั้นไม่ใช่ทั้งผู้ดูแลระบบและพนักงานแล้วก็จะมีการตรวจสอบต่อไปว่าใช่สมาชิกทั่วไปหรือไม่ ถ้าใช่ก็จะสร้าง session แล้วไปทำงานในส่วน ของ E แต่ถ้าหากไม่ใช่ก็จะถามต่อไปอีกว่า ต้องการสมัครสมาชิกหรือไม่ ถ้าหากสมัครก็จะไป ทำในส่วน ของ D

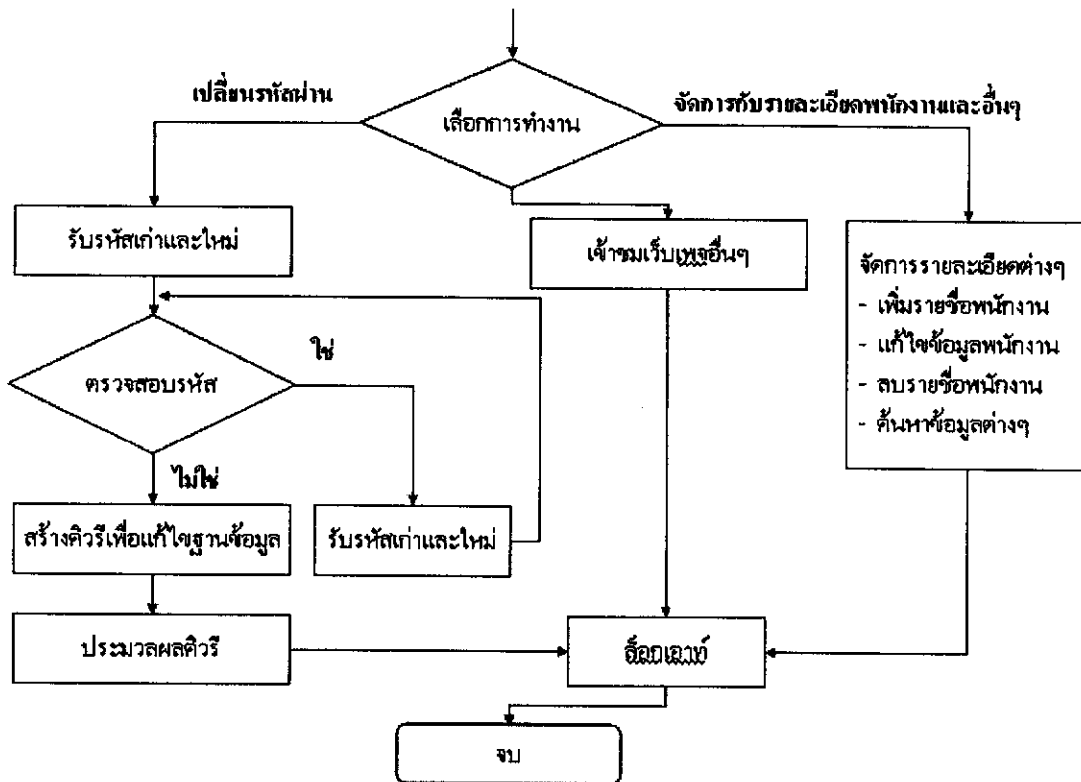
## ส่วน B กรณีที่เป็นพนักงานโรงพยาบาล



รูปที่ 3.7 แสดงการทำงานของผู้ใช้ที่เป็นพนักงาน โรงพยาบาล

จากรูปที่ 3.7 พนักงานบริษัทสามารถเลือกการทำงานได้โดยสามารถเปลี่ยนรหัสผ่านได้ และสามารถค้นหาข้อมูลและคอยรับ Message จาก PocketPC ได้ด้วย หรือจะเข้าเยี่ยมชมเว็บในส่วนอื่นๆ ก็ได้เช่นกัน

ส่วน C กรณีเป็นผู้ดูแลระบบ

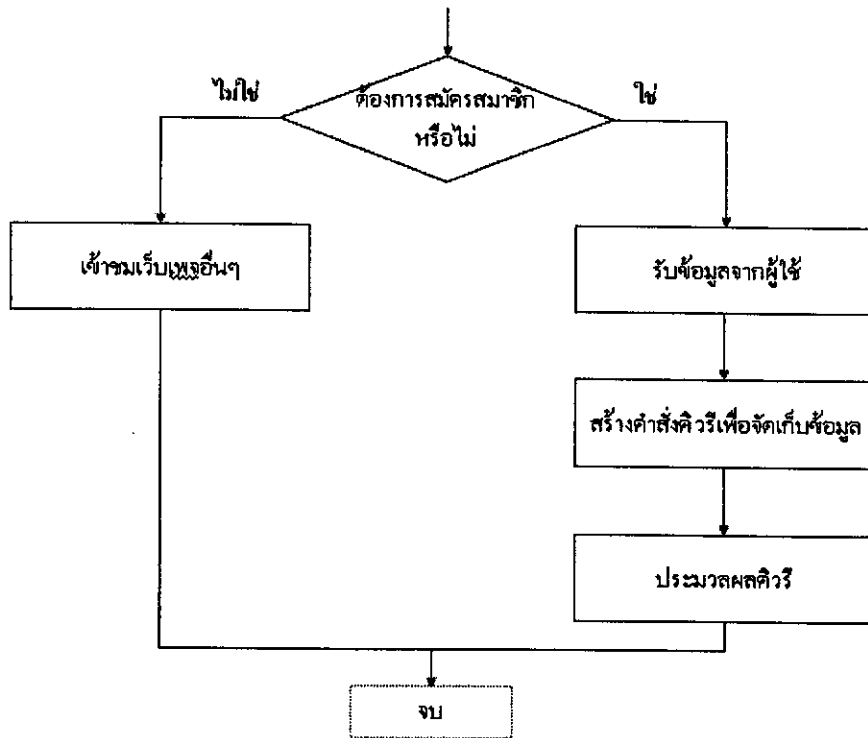


รูปที่ 3.8 แสดงการทำงานในส่วนของผู้ดูแลระบบ

จากรูปที่ 3.8 ผู้ดูแลระบบสามารถเลือกที่จะทำงาน โดยสามารถเปลี่ยนรหัสผ่าน ทั้งของตนเองและของผู้ใช้คนอื่น สามารถจัดการกับรายละเอียดของพนักงาน ไม่ว่าจะเป็น การเพิ่ม แก้ไข ลบ และค้นหาข้อมูลต่างๆ และยังสามารถเข้าชมเว็บส่วนต่างๆ ได้ด้วย

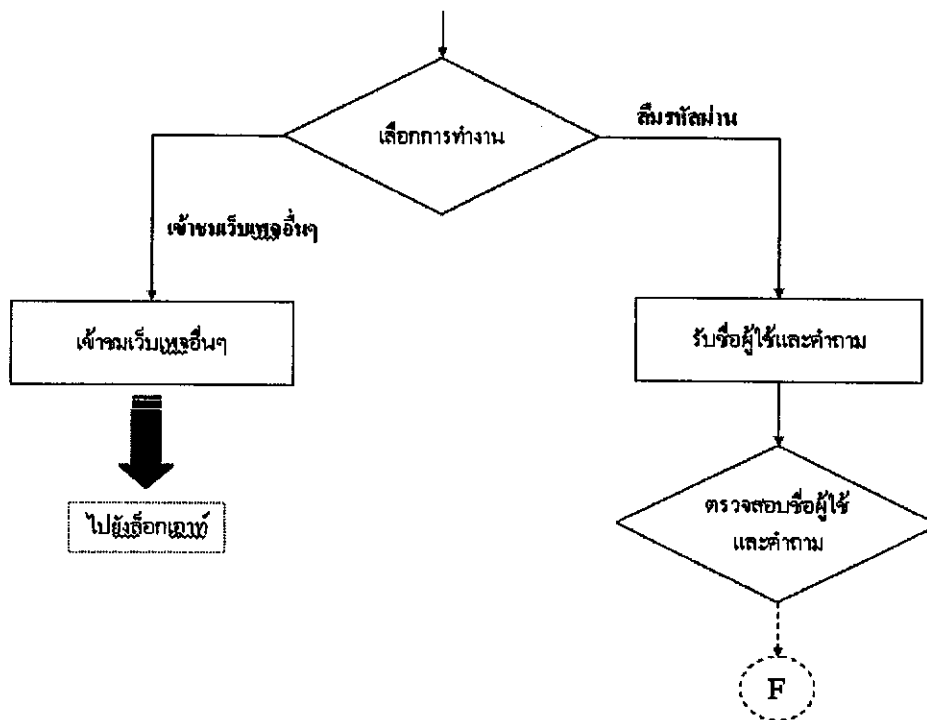
ส่วน D กรณีต้องการสมัครสมาชิก

หากผู้ใช้ยังไม่ได้ลงทะเบียนกับทางเว็บไซต์ก็ไม่สามารถใช้งานในบางส่วนได้ ดังนั้นจึงต้องมีการลงทะเบียน สมัครสมาชิก โดยถ้าหากจะสมัครก็ต้องทำการกรอกข้อมูลต่างๆ เพื่อที่จะใช้ในการประมวลผลคิวรี แล้วทำการเก็บลงฐานข้อมูล แต่ถ้าหากไม่ต้องการสมัครก็สามารถดูเว็บเพจหน้าอื่นๆ ที่ไม่มีการจำกัดสิทธิ์ได้ดังรูปที่ 3.9



รูปที่ 3.9 แสดงขั้นตอนการสมัครสมาชิกใหม่

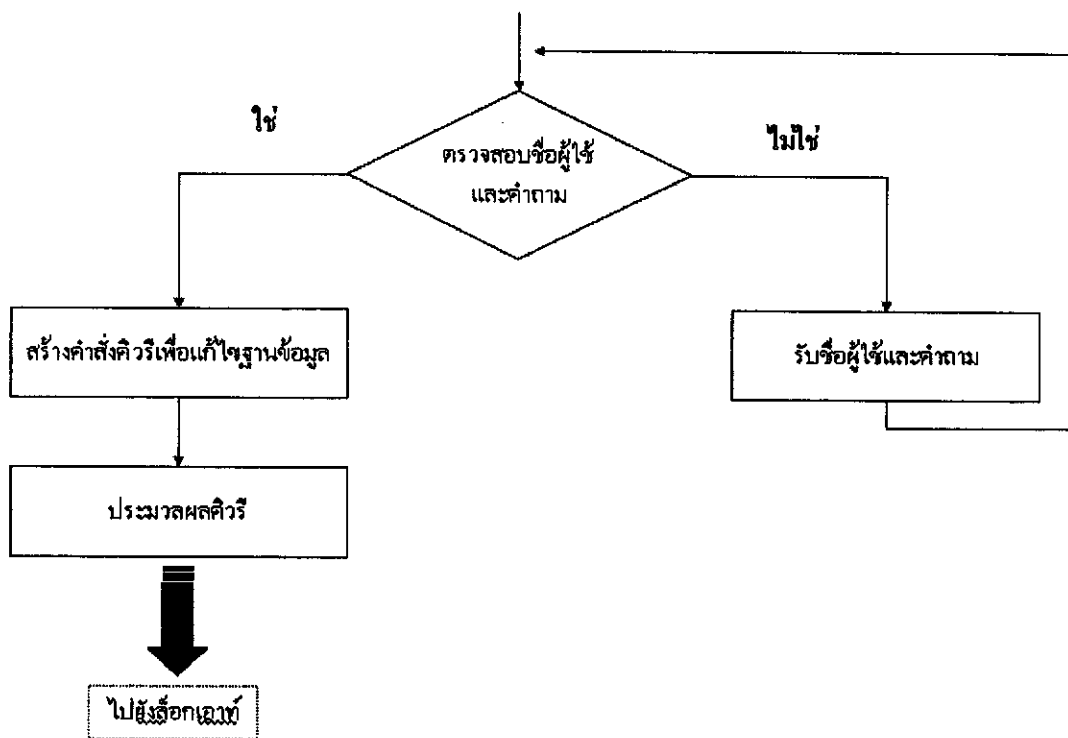
ส่วน E เป็นการทำงานของสมาชิกทั่วไป



รูปที่ 3.10 แสดงการทำงานของสมาชิกทั่วไป

สมาชิกทั่วไปสามารถที่จะเปลี่ยนรหัสผ่านได้หากลืมรหัสผ่าน โดยทำการป้อนชื่อผู้ใช้และคำตอบที่ได้จากการป้อนตอนสมัครสมาชิกใหม่ จากนั้นเมื่อตรวจสอบในส่วนของ F

## ส่วน F ส่วนตรวจสอบผู้ใช้และคำถามกรณีลืมรหัสผ่าน



รูปที่ 3.11 แสดงการทำงานกรณีลืมรหัสผ่าน

จากรูปที่ 3.11 เราจะให้ผู้ใช้ทำการป้อนชื่อผู้ใช้ และคำตอบที่ได้จากการสมัครในตอนแรก เพื่อที่จะเรียกเข้าไปตรวจสอบเพื่อที่จะเปลี่ยนรหัสผ่าน ถ้าหากว่าถูกต้องก็จะเรียกฟอร์มขึ้นมาเพื่อให้ทำการป้อนรหัสผ่านใหม่และยืนยัน จากนั้นก็จะทำการ update ฐานข้อมูลใหม่

### 3.5.2.2 การรวมระบบย่อยให้เป็นเว็บไซต์

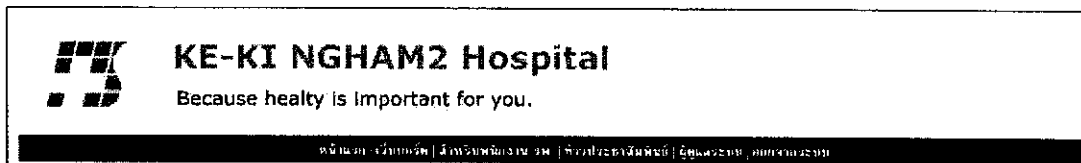
ในที่นี้จะกล่าวถึงหน้าของเว็บเพจที่มีการเชื่อมโยงไปยังส่วนต่างๆ ซึ่งมีไฟล์ที่เกี่ยวข้อง ได้แก่ index.php webboard.php officer.php news.php basic\_data\_menu.php

#### ไฟล์ index.php

เป็นหน้าแรกของเว็บไซต์ที่แสดงการทำงานส่วนต่างๆ ส่วนของการเชื่อมโยงไปยังไฟล์อื่นๆ ดังรูปที่ 3.12 ซึ่งประกอบด้วย

- เมนูหน้าแรก เชื่อมโยงไปยังไฟล์ index.php
- เมนูเว็บบอร์ด เชื่อมโยงไปยังไฟล์ webboard.php
- เมนูสำหรับพนักงาน รพ. เชื่อมโยงไปยังไฟล์ officer.php

- เมนูข่าวประชาสัมพันธ์ เชื่อมโยงไปยังไฟล์ news.php
- เมนูผู้ดูแลระบบ เชื่อมโยงไปยังไฟล์ basic\_data\_menu.php



### รูปที่ 3.12 แสดงเมนูในส่วนของหน้าแรกของเว็บไซต์

จากเมนูที่ได้แสดงดังรูปนั้น ได้มาจากการใช้ฟังก์ชัน `require_once("headeradmin.php");` ซึ่งโหลดขึ้นมาเพื่อนำมาใช้งานบนสคริปต์ ซึ่งไฟล์ `header.php` จะอยู่ในไดเรกทอรี `include` เป็นไฟล์ที่ใช้สร้างหัวของเมนูจาก code ข้างล่าง

```
<a class="cate" href="index.php"><b>หน้าแรก</b></a>
|<a class="cate" href="webboard.php"><b>เว็บบอร์ด</b></a>
|<a class="cate" href="officer.php"><b>สำหรับพนักงาน รพ.</b></a>
|<a class="cate" href="/news/news.php"><b>ข่าวประชาสัมพันธ์</b></a>
|<a class="cate" href="basic_data_menu.php"><b>ผู้ดูแลระบบ</b></a>
|<a class="cate" href="logout.php"><b>ออกจากระบบ</b></a>
```

ในรูปที่ 3.13 เป็นเมนูตรงกลางของไฟล์ `index.php` มีหน้าจอสําหรับทำการล็อกอิน เชื่อมโยงไปยังไฟล์ `process_login.php`

### รูปที่ 3.13 แสดงหน้าล็อกอิน

ในส่วนของการล็อกอิน ผู้ใช้จะป้อนชื่อและรหัสผ่านเข้าไปในฟอร์มซึ่งเมื่อคลิกปุ่ม Submit แล้วจะทำการส่งค่าไปยังไฟล์ `process_login.php`

```
<form action="process_login.php" method="post">
```

ภายในไฟล์ process\_login.php ก็มีการเชื่อมต่อกับฐานข้อมูลด้วยการ include ไฟล์ database.php จากนั้นค่อยตรวจสอบชื่อผู้ใช้และรหัสผ่าน โดยการ โหลดฟังก์ชัน require\_once

```
<?php
include("function.php");
include("database.php");

head_html("ตรวจสอบ session");
database_connect();


login($user_name,$pass);
header("Location: show.php");

end_head_html();
?>
```

เมื่อชื่อผู้ใช้และรหัสผ่านแล้วก็จะเกิดการสร้าง session ขึ้นมา โดยฟังก์ชัน login จะอยู่ในไฟล์ที่ชื่อว่า function.php

### ไฟล์ webboard.php

แบ่งออกเป็น 2 ส่วนคือส่วนที่เป็นกระทู้และส่วนที่เป็นคำตอบ โดยส่วนที่เป็นกระทู้จะทำหน้าที่ในการเก็บคำถามที่ผู้ใช้ป้อนเอาไว้ ภายในตัวกระทู้จะมีการเก็บรายละเอียดต่างๆ ได้แก่ หมายเลขกระทู้ หัวข้อกระทู้ ชื่อผู้ส่ง อีเมล วันเวลา และรายละเอียดของกระทู้ ในส่วนของคำตอบเช่นกันมีการเก็บหมายเลขคำตอบ หมายเลขกระทู้ ชื่อผู้ตอบ อีเมล วันเวลา และรายละเอียดของคำตอบ ดังรูปที่ 3.14

 <b>KE-KI NGHAM2 Hospital</b> Because healthy is important for you.				
00012	สมใจ	สมใจ	23-Feb-2006 17:23	0
00011	ขะขะขะ	พศุภ	23-Feb-2006 17:11	0
00010	อสมใจ	พศุภ	23-Feb-2006 17:10	0
00009	สมใจ	พศุภ	23-Feb-2006 17:10	0
00008	สมใจ	พศุภ	23-Feb-2006 17:07	0
00007	สมใจ	M.D	18-Feb-2006 19:06	2
00003	สมใจ	พศุภ	14-Feb-2006 02:00	2
00002	สมใจ	พศุภ	10-Feb-2006 21:17	3
00001	สมใจ	พศุภ	10-Feb-2006 21:13	1

รูปที่ 3.14 แสดงหน้าตาของเว็บบอร์ด

จากรูปที่ 3.10 เกิดจากการทำงานร่วมกันของไฟล์ต่างๆ ซึ่งได้แก่ webboard.php new\_discuss.php reply\_discuss.php discussion.php webboard\_db.php และ genhtml.php ซึ่งแต่ละไฟล์จะมีหน้าที่การทำงานดังนี้คือ

Webboard.php	ใช้แสดงหน้าเว็บบอร์ดหลัก
New_discuss.php	สร้างกระทู้ใหม่ และนำกระทู้ที่ได้เก็บไว้ในตาราง question ในฐานข้อมูล webboard
Reply_discuss.php	เพิ่มคำตอบลงในตาราง answer บนฐานข้อมูล webboard
Discussion.php	แสดงรายละเอียดของกระทู้และคำตอบ
Webboard_db.php	จัดการติดต่อกับฐานข้อมูล webboard รวมทั้งการสร้างคิวรีต่างๆ แล้วนำคิวรีเหล่านั้นไปประมวลผลด้วยฟังก์ชันที่เกี่ยวกับ MySQL
Genhtml.php	ทำหน้าที่กำหนดรูปแบบเอกสาร HTML ซึ่งเป็นโครงสร้างของตัวเว็บบอร์ด

### ไฟล์ officer.php

เป็นไฟล์ที่เอาไว้ตรวจสอบกรณีล็อกอินที่เป็นชื่อของพนักงาน เพื่อใช้ในการตรวจสอบสิทธิ์ก่อน จากนั้นก็จะมีเมนูย่อยด้านในอีกดังรูปที่ 3.15 ซึ่งประกอบด้วย

หน้าแรก | เว็บบอร์ด | สำหรับพนักงาน รพ. | ข่าวประชาสัมพันธ์ | ผู้ดูแลระบบ | ออกจากระบบ

รูปที่ 3.15 แสดงหน้าต่างในส่วนเมนูของพนักงาน

### ฟังก์ชัน login()

```
function login($user,$pass)
{
$str = "select * from user_profile where user_name = '$user' and password = '$pass'";
$ret = database_query($str);
$row = mysql_fetch_array($ret);
if(mysql_num_rows($ret) != 0) {
session_register("login");
if($row["STATUS"] == '0')
session_register("status");
}
else {
```

```

message_box("", "User name หรือ Password ผิด ไม่สามารถล็อกอิน
ได้", "javascript:history.back()");
exit;
}
}

```

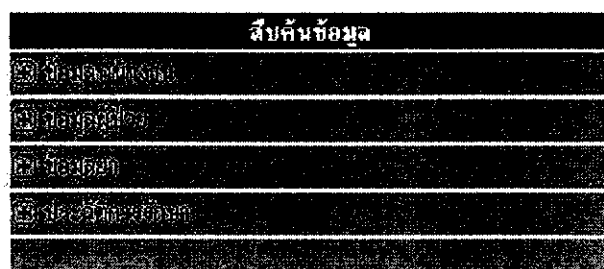
### ฟังก์ชัน check\_session()

```

function check_session() {
    session_start();
    if (!session_is_registered("login")) {
        message_box("", "คุณยังไม่ได้ล็อกอินเข้าระบบ", "index.php");
        exit;
    }
}

```

เมื่อเลือกเมนูค้นหาข้อมูลเราจะได้เมนูดังรูปที่ 3.16 ซึ่งเมนูเหล่านี้จะเชื่อมโยงไปยังไฟล์ต่างๆ



รูปที่ 3.16 แสดงเมนูหน้าค้นหาข้อมูล

จากนั้นเมื่อเข้าฟอร์มค้นหาเราก็จะมีการใช้ตัวแปรในเก็บค่าที่ป้อนไว้ในช่องค้นหา เพื่อจะส่งค่าไปทำการคิวรีในฐานข้อมูล

รูปที่ 3.17 แสดงฟอร์มในการค้นหาข้อมูลพนักงาน

ในการแสดงผลลัพธ์ที่ได้จากค้นหา ข้อมูลที่จะนำมาแสดงจะประกอบไปด้วยฟังก์ชันต่างๆ ดังนี้

### ส่วนของไฟล์ที่ Include เข้ามา

```
include("database.php");
include("function.php");
require_once("connect_db.php");
require_once("headeradmin.php");
```

### ฟังก์ชันที่ใช้ตรวจสอบสิทธิ์

```
check_session();
check_permit();
```

### ฟังก์ชันที่เกี่ยวข้องกับฐานข้อมูล

```
$dbname = "hospital";
$tablename = "tbldoctor";
mysql_select_db($dbname) or die("เลือกฐานข้อมูลไม่ได้");

$sql_search = "select * from $tablename where $type like '%$search%' order by
DoctorID,DoctorName";

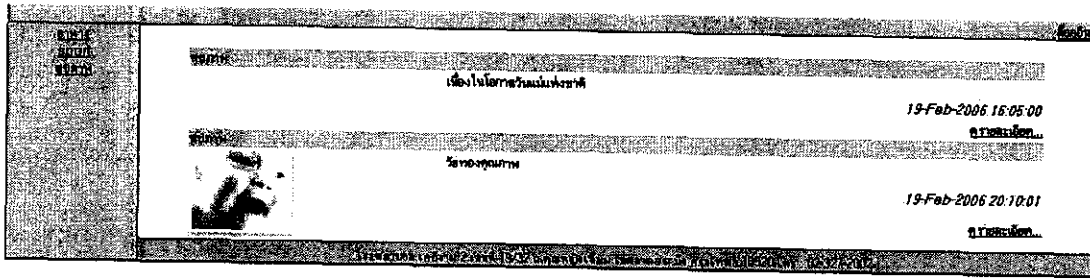
$db_query_search = mysql_db_query($dbname, $sql_search);
$num_rows_search = mysql_num_rows($db_query_search);
$num_fields = mysql_num_fields($db_query_search);
```

รหัส	ชื่อ	ที่อยู่	ยี่สิบสี่	รหัสประจำตัว	หมายเลขความเกี่ยวข้อง	เบอร์โทรศัพท์มือถือ	เบอร์โทรศัพท์บ้าน	อีเมล
45015342	วิไลลักษณ์ภักดิ์	208 ถนนเพชรฯ	ศาลเจ้า	๗๗๗	๒	091574893	023272002	mail5342@163.com

รูปที่ 3.18 แสดงผลลัพธ์ในการค้นหาข้อมูลพนักงาน

### ไฟล์ news.php

สร้างระบบการจัดการข้อมูลบนเว็บ ในที่นี้คือข่าวประชาสัมพันธ์ โดยโครงสร้างของระบบประกอบด้วย ส่วนของกลุ่มข่าว และส่วนของข่าว ในส่วนของกลุ่มข่าวจะรวมข่าวที่มีเนื้อหาสัมพันธ์กันมารวมกัน ส่วนข่าวจะแสดงรายละเอียดและเนื้อหาที่อยู่ภายในหัวข่าวนั้นๆ แสดงดังรูปที่ 3.19 โดยในส่วนนี้ admin หรือผู้ที่กำหนดสามารถเข้าไปสร้างหัวข้อได้ผ่านทางเมนูลืออกทางด้านขวา



รูปที่ 3.19 แสดงหน้าต่างในส่วนของข่าวประชาสัมพันธ์

ไฟล์ `basic_data_menu.php`

จะสร้างฟอร์มเพื่อให้ผู้ดูแลระบบได้ทำการล็อกอินเข้าไป ซึ่งในส่วนนี้เมื่อเข้าไปแล้ว จะมีเมนูต่างๆ ให้ผู้ดูแลระบบจัดการได้แก่ เพิ่มข้อมูลบุคลากร แก้ไขรายละเอียดบุคลากร และการจัดหมวดหมู่ความเชี่ยวชาญ และเมนูอื่นๆซึ่งจะได้อธิบายในแต่ละส่วนดังรูปที่จะแสดงต่อไปนี้

### 1.) เพิ่มข้อมูลบุคลากรใหม่

รูปที่ 3.20 แสดงหน้าต่างในส่วนของการเพิ่มบุคลากรใหม่

จากรูปที่ 3.20 ผู้ดูแลระบบสามารถกรอกเพิ่มข้อมูลบุคลากรของโรงพยาบาลเข้าไปเพิ่มในฐานข้อมูลได้โดยรายละเอียดต่างๆ จะแสดงในรูปซึ่งผลลัพธ์เกิดจากการเรียกไฟล์ `insertofficer_form.php`

## 2.) แก้ไขรายละเอียดบุคลากร

หลังจากที่ admin ทำการเพิ่มข้อมูลเข้าไปแล้วก็สามารถทำการแก้ไขได้ โดยเลือกที่เมนู แก้ไขรายละเอียดบุคลากร จากนั้นผลลัพธ์ที่ได้จะเกิดจากการเปิดไฟล์ modify\_form.php ดังรูปที่ 3.20 ซึ่งจะขึ้นรายชื่อของพนักงานให้ admin เลือกว่าจะแก้ไขใคร

เลือก	ID	ชื่อ-สกุล	อีเมล
<input type="checkbox"/>	0001	วิวิศ เพ็ญพนาพงษ์	s6015342@kmitl.ac.th
<input type="checkbox"/>	0003	ชัชชมน วัฒนศักดิ์	กชชชชชชช@hotmail.com
<input type="checkbox"/>	0002	จักริน ชินวัตร	chinnewai@hotmail.com
<input type="checkbox"/>	0005	วิวัฒน์ วัฒนศักดิ์	กชชชชชชช@hotmail.com

รูปที่ 3.21 แสดงรายชื่อบุคลากรที่ผู้ดูแลระบบสามารถแก้ไขหรือลบได้

หลังจากเลือก checkbox แล้วว่าจะเลือกใครที่ทำการคลิกที่ชื่อบุคคลนั้นแล้วโปรแกรมจะเรียกไฟล์ editofficer\_form.php เพื่อดึงข้อมูลของบุคลากรที่ต้องการแก้ไขขึ้นมา จากนั้น admin สามารถที่จะลบหรือแก้ไขได้ ดังรูปที่ 3.22

แก้ไข (ตัวกรอง: 0 บุคลากร)

ID:

ชื่อ-สกุล:

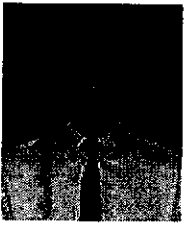
ความเชี่ยวชาญ:

ที่อยู่:

โทรศัพท์มือถือ:

โทรศัพท์บ้าน:

อีเมล:



รูปที่ 3.22 แสดงหน้าต่างในส่วนของการแก้ไขรายละเอียดบุคลากร

### 3.) การจัดหมวดหมู่ความเชี่ยวชาญ

ในส่วนนี้จะไม่ขออธิบายเนื่องจากการทำงานจะเหมือนกันหัวข้อที่ 2.) ต่างกันตรงที่ข้อมูลที่ใช้ป้อนเท่านั้น

#### ฟังก์ชันหลักที่ใช้ในแต่ละไฟล์

Connect_db()	ใช้สำหรับเชื่อม โยงฐานข้อมูล
Header_html()	ใช้แสดงส่วนหัวของจอภาพ
Change_password()	ใช้สำหรับเปลี่ยนรหัสผ่าน
Check_valid_user()	ใช้ตรวจสอบผู้ใช้ที่เข้ามานั้นเป็นคนที่ถูกต้องหรือไม่
Valid_email()	ใช้ตรวจสอบอีเมลว่ารับเข้ามาถูกต้องหรือไม่
Check_admin()	ใช้ตรวจสอบผู้ดูแลระบบว่าผ่านการล็อกอินถูกต้องหรือไม่
Filled_out()	ใช้ตรวจค่าในอาร์เรย์
Login()	ใช้ตรวจสอบผู้ใช้ที่ล็อกอินเข้ามาว่ามีข้อมูลในฐานข้อมูลหรือไม่
Register()	ใช้สำหรับลงทะเบียนสมาชิก
Insert_person()	ใช้เพิ่มรายชื่อพนักงานในตาราง tbldoctor
Delete_person()	ใช้ลบรายชื่อพนักงานในตาราง tbldoctor ตาม DoctorID
Edit_person()	ใช้แก้ไขรายละเอียดของพนักงานในตาราง tbldoctor
Show_data()	ใช้แสดงข้อมูลจากฐานข้อมูล "hospital"
Select_person()	ใช้เลือกข้อมูลของพนักงานจากฐานข้อมูลขึ้นมาตาม DoctorID ที่ระบุ เพื่อนำรายชื่อแต่ละคนไปจัดการ ลบ หรือแก้ไข
Select_cat	ใช้เลือกข้อมูลของพนักงานจากฐานข้อมูลขึ้นมาตามหมวดหมู่ความเชี่ยวชาญที่ระบุ เพื่อนำหมวดหมู่ความเชี่ยวชาญเหล่านั้นไปจัดการ ลบ หรือแก้ไข

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 สรุปความสามารถที่โปรแกรมทำได้

##### 4.1.1 โปรแกรมบน Pocket PC (Mobile Computing Application)

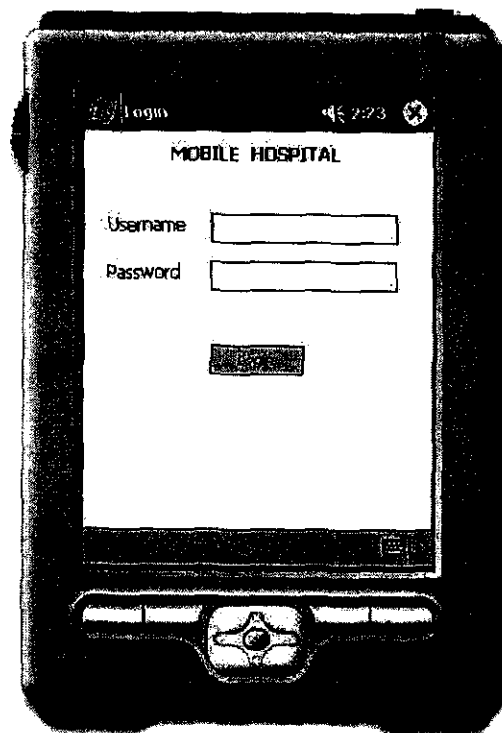
- ล็อกอินเข้าสู่โปรแกรม
- ดึงข้อมูลของผู้ป่วยที่จะทำการรักษา
- ป้อนข้อมูลใหม่ที่ได้จากการตรวจลงไปยังบันทึกในฐานข้อมูล

##### 4.1.2 โปรแกรม Web Application

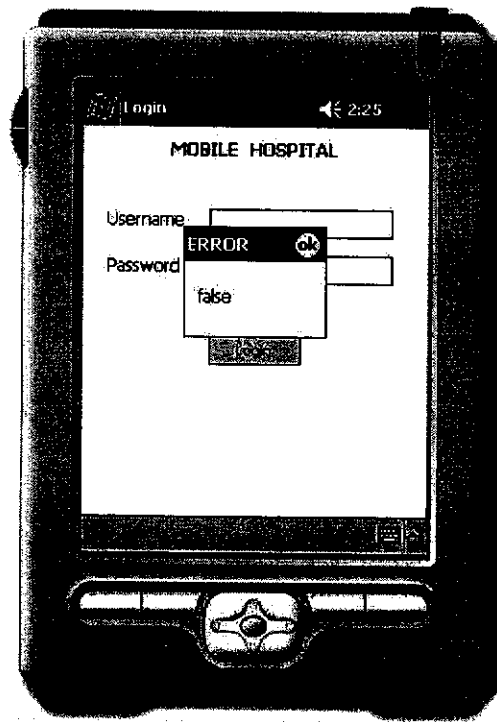
- ล็อกอินเข้าสู่ระบบทั้งสมาชิกทั่วไป พนักงาน และผู้ดูแลระบบ
- ผู้ดูแลระบบสามารถเพิ่ม ลบ แก้ไข รายละเอียดเกี่ยวกับพนักงานได้
- พนักงานสามารถล็อกอินเข้าไปในส่วนที่ได้รับอนุญาตได้ และสามารถใช้เมนูค้นหาเพื่อดูข้อมูลคนไข้ที่ทำการรักษาได้

#### 4.2 ผลการทำงานของโปรแกรม

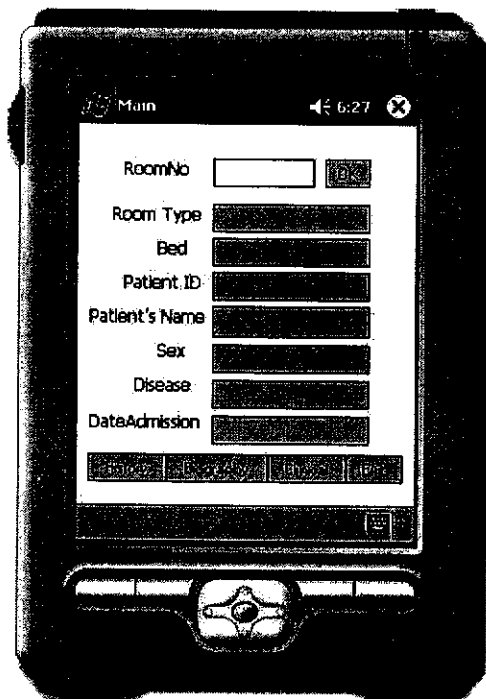
##### 4.2.1 โปรแกรมบน Pocket PC (Mobile Computing Application)



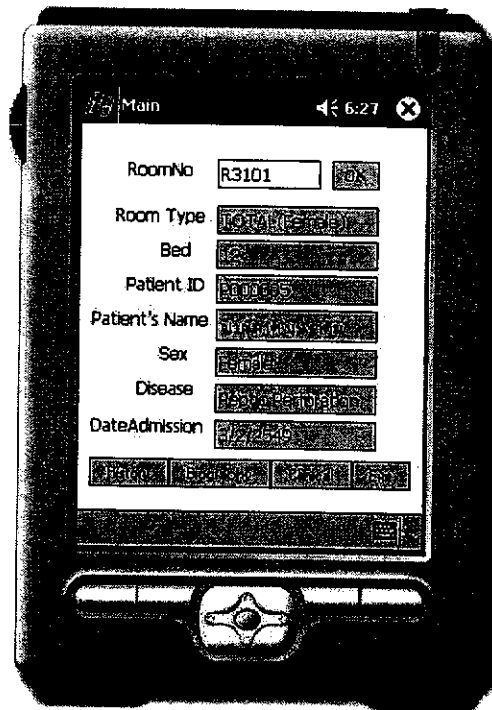
รูปที่ 4.1 แสดงหน้า Login ของ Pocket PC



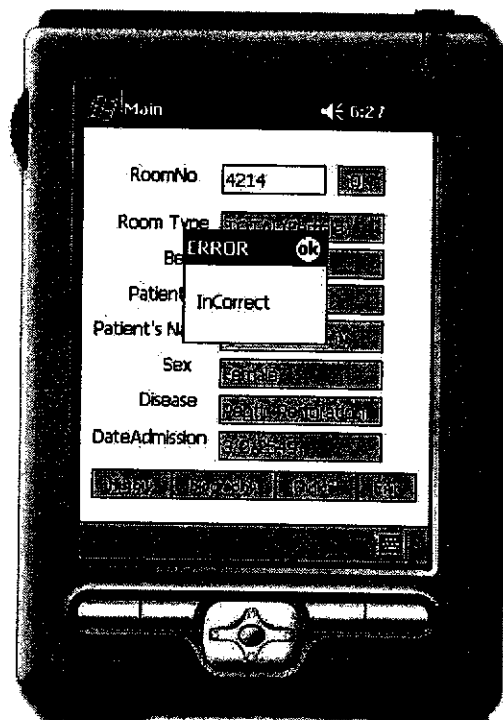
รูปที่ 4.2 แสดงหน้า Login ในกรณีที่เกิดการผิดพลาดในการ Login



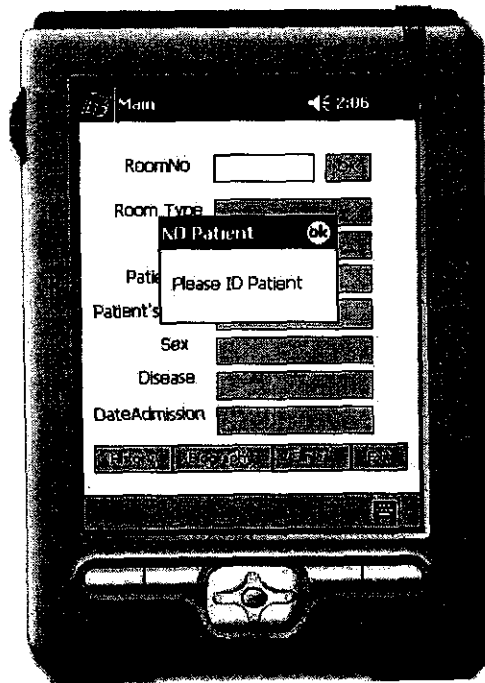
รูปที่ 4.3 หน้า Main ก่อนที่จะทำการป้อนหมายเลขห้อง



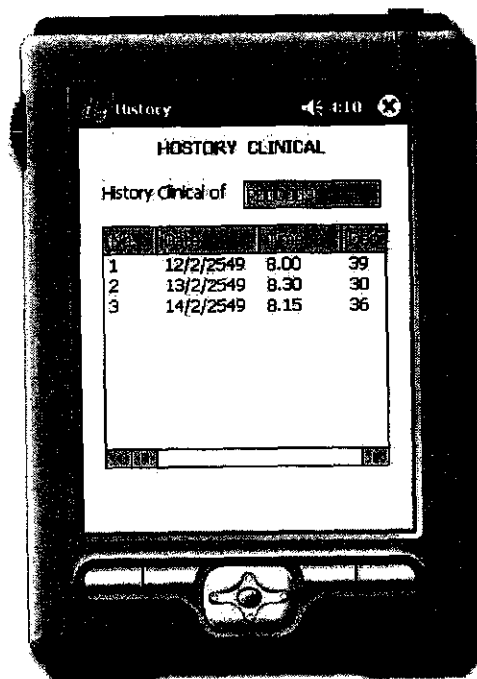
รูปที่ 4.4 แสดงหน้า Main เมื่อทำการป้อนหมายเลขห้องเรียบร้อยแล้ว และกดปุ่ม OK ก็จะได้  
ข้อมูลการเจ็บป่วยของผู้ป่วย



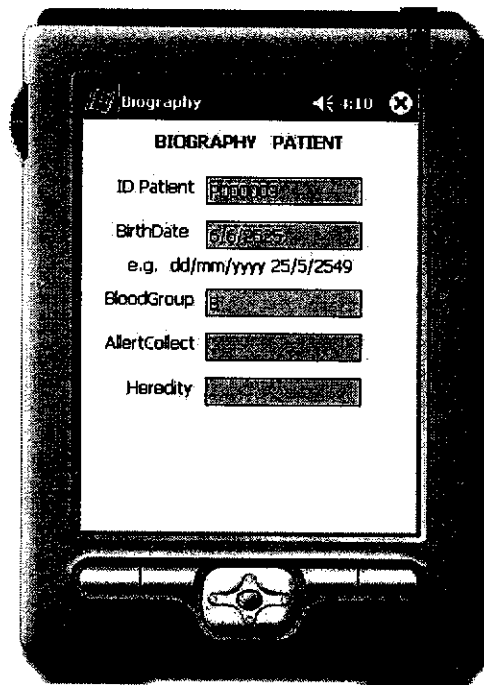
รูปที่ 4.5 แสดงข้อความเตือนกรณีที่ไม่ได้กรอกหมายเลขห้อง แล้วคลิก OK  
หรือหมายเลขห้องไม่มีอยู่จริง



รูปที่ 4.6 แสดงข้อความเตือนกรณีที่ไม่ได้กรอกหมายเลขห้อง แล้วคลิกปุ่ม History หรือ Biography

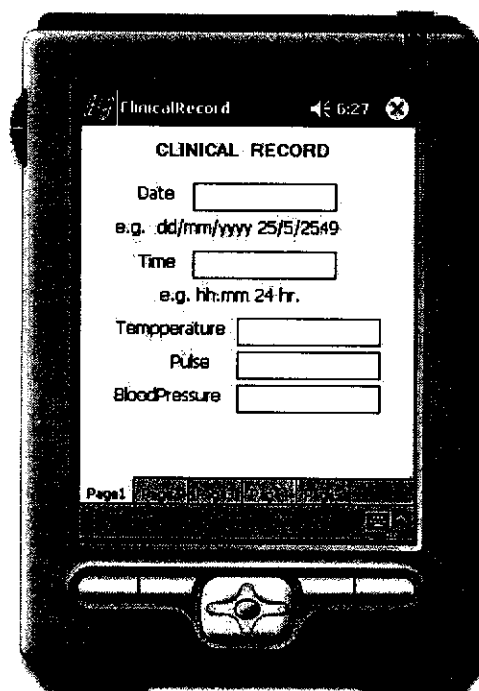


รูปที่ 4.7 เมื่อกดปุ่ม History ที่หน้า Main ก็จะแสดงหน้าข้อมูลการรักษาของผู้ป่วย



รูปที่ 4.8 เมื่อกดปุ่ม Biography ที่หน้า Main ก็จะแสดงหน้าข้อมูลส่วนตัวของผู้ป่วย

ต่อไปเมื่อกดปุ่ม Clinical ที่หน้า Main ก็จะเข้าสู่หน้า Clinical Record เพื่อทำการตรวจรักษา ซึ่งจะมีด้วยกันทั้งหมด 7 หน้า ดังต่อไปนี้



รูปที่ 4.9 หน้าแรกบันทึก วันที่ตรวจรักษา ,เวลาที่ตรวจ ,อุณหภูมิร่างกาย ,ชีพจร , ความดันโลหิต

ClinicalRecord 6:27

**CLINICAL RECORD**

Respirations

B.P. Systolic

Diastolic

Stools

Urine

Page2

รูปที่ 4.10 หน้าที 2 บันทึกการหายใจ ,อัตราการบีบของหัวใจ ,อัตราการพองของหัวใจ ,  
ตรวจอุจจาระ ,ตรวจปัสสาวะ

ClinicalRecord 6:27

**CLINICAL RECORD**

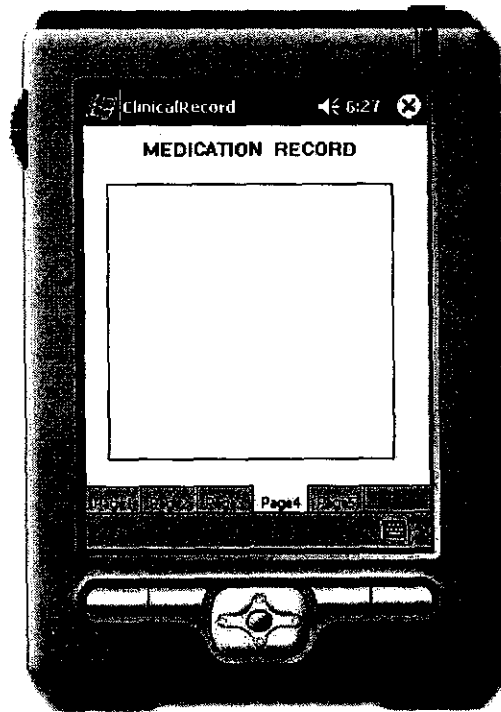
State

Diagnosis

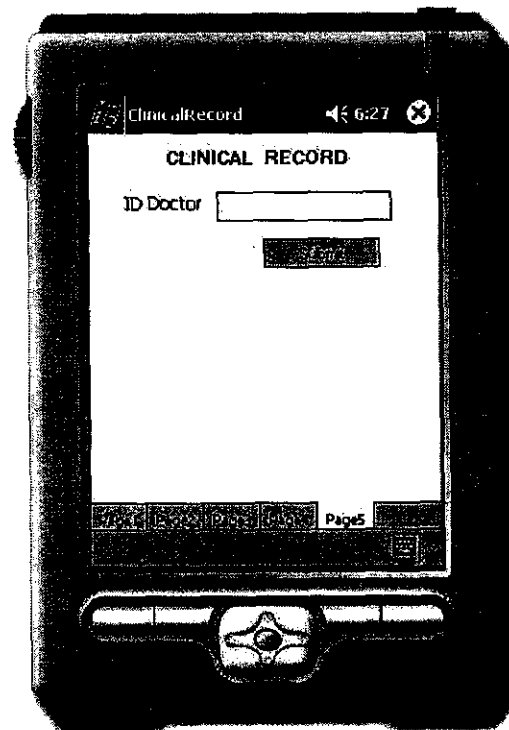
Don't eat

Page3

รูปที่ 4.11 หน้าที 3 บันทึกอาการของผู้ป่วย ,คำวินิจฉัย ,ของทีห้ามรับประทาน



รูปที่ 4.12 หน้าที 4 บันทึกการสั่งเวชภัณฑ์ทางการแพทย์



รูปที่ 4.13 หน้าที 5 ใส่รหัสของผู้ที่ทำการตรวจรักษา

## 4.2.2 โปรแกรม Web Application

- หน้าหลัก

ประกอบด้วยเมนูต่างๆ และที่ล็อกอินสำหรับผู้ใช้



**KE-KI NGHAM2 Hospital**

Because healthy is important for you.

หน้าแรก | เว็บบอร์ด | ข่าวประชาสัมพันธ์ | ติดต่อเรา | เกี่ยวกับเรา | ติดต่อเรา

โรงพยาบาลโพธาราม 2 เลขที่ 13/32 อ.ดอนเจดีย์ จ.เพชรบุรี โทร. 0 2327 2002

### รูปที่ 4.14 หน้าโฮมเพจ

จากรูปที่ 4.14 จะเห็นได้ว่าเป็นหน้าแรกของเว็บไซต์ โดยที่เห็นจะประกอบไปด้วยส่วนต่างๆ ซึ่งสามารถอธิบายคร่าวๆ ได้ดังนี้

1. ส่วนของการล็อกอิน : เอาไว้ให้ผู้ใช้ทำการล็อกอินก่อนที่จะเข้าใช้งานในส่วนต่างๆ ของเว็บที่มีการจำกัดสิทธิ์ผู้ใช้
2. ส่วนที่ช่วยในกรณีลืมรหัสผ่าน : ผู้ใช้แต่ละคนจะมีคำตอบที่ตนเองได้กรอกไว้ตอนสมัคร ซึ่งสามารถนำมาช่วยในกรณีที่เราจำรหัสผ่านไม่ได้ ก็ให้เข้าไปที่เมนูลืมรหัสผ่าน จากนั้นก็ป้อน Username และตอบคำถามก็สามารถเปลี่ยนรหัสผ่านใหม่ได้
3. ส่วนของการลงทะเบียน : ส่วนนี้จัดทำไว้ให้กับบุคคลทั่วไปสมัครเข้ามาใช้บริการเว็บ ซึ่งผู้สมัครต้องกรอกข้อมูลให้ครบก่อนถึงจะเป็นสมาชิกได้
4. ส่วนของเว็บบอร์ด : ส่วนนี้สามารถใช้งานได้สำหรับผู้ใช้ทุกประเภท แต่ต้องสมาชิกก่อนถึงจะใช้งานได้ ก็เป็นกระดานถามตอบกันในเรื่องปัญหา หรือสิ่งที่สนใจต่างๆ
5. ส่วนสำหรับพนักงานโรงพยาบาล : ในส่วนนี้จัดไว้ให้กับผู้ใช้ที่เป็นพนักงานของโรงพยาบาลเท่านั้น โดยภายในจะมีส่วนที่อำนวยความสะดวกด้านต่างๆ เช่น การค้นหาข้อมูล และหน้าที่สำคัญของส่วนนี้ก็คือคอยตรวจสอบว่าหมอได้ทำการสั่งยาหรือเวชภัณฑ์มาหรือเปล่า (หมอที่ใช้ Pocket PC) เมื่อมีคำสั่งมาพยาบาลจะต้องดูแล้วยทำการนำจ่ายยาหรือเวชภัณฑ์ต่อไป
6. ส่วนข่าวประชาสัมพันธ์ : เป็นหน้าที่เอาไว้สำหรับประกาศข่าวสารของทางโรงพยาบาลเพื่อให้คนที่เข้ามาเยี่ยมชมได้รับทราบ ส่วนนี้สามารถเข้าชมได้ทุกคน

7. ส่วนของผู้ดูแลระบบ : ในส่วนนี้สามารถจัดการในเรื่องของพนักงานโรงพยาบาล ไม่ว่าจะเป็นการเพิ่ม ลบ แก้ไข ค้นหาข้อมูลต่างๆ รวมถึงสามารถจัดการกับผู้ใช้แต่ละประเภทได้อีกด้วย

ต่อจากนี้เราก็จะมาทราบการทำงานของแต่ละส่วนกันว่ามีการทำงาน และผลการทำงานเป็นอย่างไรบ้าง

- ส่วนของการล็อกอิน

ใช้ในการล็อกอินเข้าสู่ระบบ โดยมีการจำกัดสิทธิ์ในแต่ละ User ผู้ใช้ที่ได้รับ Username และ Password แล้วก็ทำการป้อน จากนั้นระบบก็จะทำการตรวจสอบ ถ้าหากว่ามีตัวตนอยู่จริงก็จะให้ผ่านเข้าไปทำงานในแต่ละส่วนที่อนุญาตให้ทำ ถ้าหากไม่ใช่ผู้ใช้ก็ต้องสมัครสมาชิกก่อน

รูปที่ 4.15 หน้า User Login

- ส่วนที่ช่วยในกรณีลืมรหัสผ่าน

ใช้ในกรณีที่ผู้ใช้ที่สมัครสมาชิกเอาไว้แล้วแต่ลืมรหัสผ่าน สามารถเข้ามาส่วนนี้ได้เพื่อทำการเปลี่ยนรหัสผ่าน แต่ท่านต้องจำคำตอบที่เราใช้เป็นคีย์ในการที่จะตรวจสอบได้ด้วย ในตอนที่สมัครสมาชิกเราจะมีช่องให้ผู้ใช้ที่สมัครทำการป้อนคำตอบของคำถามที่เราได้ถาม ถ้าหากท่านตอบคำถามแล้วได้คำตอบที่ถูกต้องก็สามารถป้อนรหัสผ่านใหม่ได้ แต่ถ้าหากไม่ถูกต้องก็ไม่สามารถเปลี่ยนรหัสได้

รูปที่ 4.16 หน้าสำหรับตอบคำถามเพื่อใช้ในการเปลี่ยนรหัส

จากรูปที่ 4.16 ถ้าหากว่าการตรวจสอบถูกต้องท่านก็สามารถที่จะเข้าไปเปลี่ยนรหัสผ่านได้ โดยหน้าต่างของฟอร์มในการเปลี่ยนรหัสผ่านจะเป็นดังรูปที่ 4.17

รูปที่ 4.17 หน้าสำหรับเปลี่ยนรหัสผ่านกรณีลืมรหัสผ่าน

ในการเปลี่ยนรหัสผ่านแต่ละครั้งจะมีเงื่อนไขในการตรวจสอบ เช่น ต้องป้อนให้อยู่ในช่วงที่โปรแกรมกำหนด และรหัสผ่านจะต้องเหมือนกันทั้งสองช่องด้วย ซึ่งถ้าหากป้อนไม่ตรงตามเงื่อนไขที่ได้วางไว้ก็จะเกิดหน้าจอแสดงความผิดพลาดขึ้นมาดังรูปที่ 4.18 และ 4.19

รูปที่ 4.18 แสดงความผิดพลาดกรณีความยาวของรหัสผ่านไม่อยู่ในช่วงที่กำหนด

รูปที่ 4.19 แสดงความผิดพลาดกรณีรหัสผ่าน 2 ช่องไม่ตรงกัน

- ส่วนของการลงทะเบียน

ส่วนนี้จัดทำไว้ให้กับบุคคลทั่วไป สมัครเข้ามาใช้บริการเว็บ ซึ่งผู้สมัครต้องกรอกข้อมูลให้ครบก่อนถึงจะเป็นสมาชิกได้

**ลงทะเบียน**

อีเมล: s6015342@kmitl.ac.th

ชื่อจริง: ชยาภรณ์

นามสกุล: ภิญญศักดิ์

ชื่อผู้ใช้: s6015342

รหัสผ่าน: ●●●●●●

คำถาม: อาหารอะไรที่คุณชอบที่สุด?

คำตอบ: salad

สถานะ:  สมาชิกทั่วไป

รูปที่ 4.20 แสดงหน้าลงทะเบียนสมัครสมาชิกทั่วไป

เมื่อกรอกข้อมูลครบแล้วก็ให้คลิกที่ปุ่ม ลงทะเบียน จากนั้นระบบก็จะทำการบันทึกข้อมูลของสมาชิกคนนั้นลงฐานข้อมูล แล้วแจ้งผลการทำงานว่าเรียบร้อยแล้ว ดังรูปที่ 4.21

**เพิ่มรายชื่อผู้ใช้ในฐานข้อมูลเรียบร้อยแล้ว**

[กลับหน้าเดิม](#)

รูปที่ 4.21 แสดงผลลัพธ์การลงทะเบียนที่เสร็จสมบูรณ์

- ส่วนของเว็บบอร์ด

ส่วนนี้สามารถใช้งานได้สำหรับผู้ใช้ทุกประเภท แต่ต้องสมาชิกก่อนถึงจะใช้งานได้ ก็เป็นกระดานถามตอบกันในเรื่องปัญหา หรือสิ่งที่สนใจต่างๆ โดยประกอบไปด้วย 3 ส่วนใหญ่ๆ ได้แก่

1. หน้าแสดงกระทู้ทั้งหมด : ซึ่งเป็นหน้าแรกเมื่อเข้ามาเมนูเว็บบอร์ด จะแสดงกระทู้ต่างๆ ให้ผู้ใช้ได้เห็นแล้วสามารถเข้าไปอ่านและตอบคำถาม หรือตั้งกระทู้ใหม่ได้

KE-KI NGHAM2 Hospital			
Because healthy is important for you.			
00012	salad	salad	23-Feb-2006 17:23
00011	ชยาภรณ์	ชยาภรณ์	23-Feb-2006 17:11
00010	ภิญญศักดิ์	ชยาภรณ์	23-Feb-2006 17:10
00009	ชยาภรณ์	ชยาภรณ์	23-Feb-2006 17:10
00008	ชยาภรณ์	ชยาภรณ์	23-Feb-2006 17:07
00007	เป็นโรคนี้แล้ว	Mr.D	16-Feb-2006 19:06
00003	ขอคำแนะนำเรื่องพอกัน	ชยาภรณ์	14-Feb-2006 02:00
00002	ขอคำแนะนำเรื่องพอกัน	ชยาภรณ์	10-Feb-2006 21:17
00001	ขอคำแนะนำเรื่องพอกัน	ชยาภรณ์	10-Feb-2006 21:13

รูปที่ 4.22 หน้าแสดงกระทู้ทั้งหมดของเว็บบอร์ด

2. หน้าแสดงการตอบกระทู้ : ผู้ใช้สามารถคลิกเลือกดูหัวข้อของกระทู้แล้วเข้าไปอ่านข้อความที่มีคนอื่นตอบเอาไว้ได้ และท่านยังสามารถตอบคำถามเพิ่มเติมท้ายเข้าไปได้ด้วย

<b>สมัครสมาชิกเว็บไซต์กระทู้</b> ขอเชิญลงทะเบียนสมาชิกใหม่กับเว็บไซต์ของเรา ชื่อ : สมชาย นามสกุล : นามสกุล		
E-mail : neng_tun@hotmail.com	รหัสผ่าน : 1234567890	รหัสยืนยัน : 0987654321
* รหัสผ่านต้องเป็นตัวเลข 6 หลัก และต้องมีตัวอักษรด้วย * รหัสยืนยันต้องเป็นตัวเลข 6 หลัก และต้องมีตัวอักษรด้วย		
* รหัสผ่านต้องเป็นตัวเลข 6 หลัก และต้องมีตัวอักษรด้วย * รหัสยืนยันต้องเป็นตัวเลข 6 หลัก และต้องมีตัวอักษรด้วย		

\* ผู้ตอบ :

\* ชื่อ :

\* รายละเอียด :

OK Clear

รูปที่ 4.23 หน้าแสดงการตอบกระทู้

3. หน้าตั้งกระทู้ใหม่ : ผู้ใช้สามารถตั้งกระทู้ขึ้นมาใหม่ได้ โดยการเลือกที่เมนูตั้งกระทู้ใหม่ จากนั้นก็ป้อนรายละเอียด ที่สำคัญคือ ชื่อกระทู้ เจ้าของกระทู้ และรายละเอียด จากนั้นกระทู้ก็จะปรากฏที่หน้าแรกของเว็บบอร์ด

	<b>KE-KI NGHAM2 Hospital</b> Because healthy is important for you.
---	---

\* ชื่อกระทู้ :

\* หัวข้อกระทู้ :

\* ชื่อ :

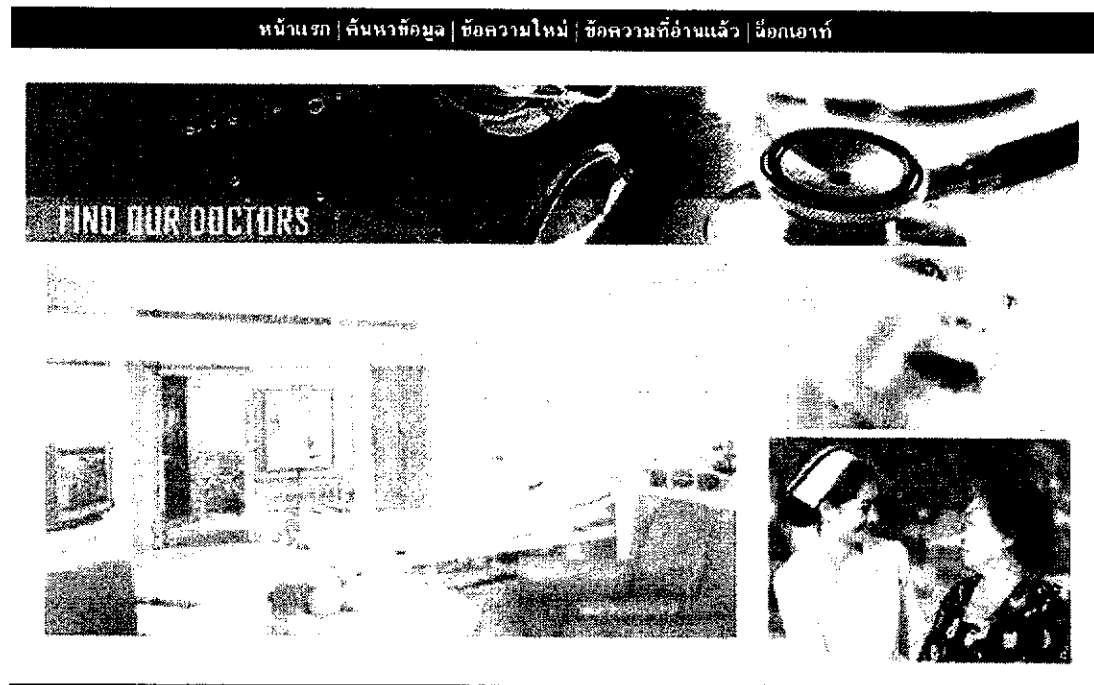
\* รายละเอียด :

OK Clear

รูปที่ 4.24 หน้าฟอร์มตั้งกระทู้ใหม่

- ส่วนสำหรับพนักงานโรงพยาบาล

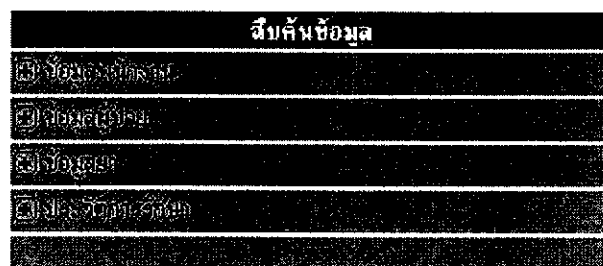
ในส่วนนี้จัดไว้ให้กับผู้ใช้ที่เป็นพนักงานของโรงพยาบาลเท่านั้น โดยภายในจะมี ส่วนที่อำนวยความสะดวกด้านต่างๆ เช่น การค้นหาข้อมูล และหน้าที่สำคัญของส่วนนี้ก็คือคอย ตรวจสอบว่าหมอได้ทำการสั่งยาหรือเวชภัณฑ์มาหรือเปล่า (หมอที่ใช้ Pocket PC) เมื่อมีคำสั่งมา โรงพยาบาลจะต้องดูแลแล้วทำการนำจ่ายยาหรือเวชภัณฑ์ต่อไป



รูปที่ 4.25 หน้าหลักของพนักงานโรงพยาบาล

จากรูปที่ 4.25 แสดงหน้าเมนูหลักของพนักงานโรงพยาบาล ที่สามารถที่จะค้นหาข้อมูล ตรวจสอบข้อความใหม่ และข้อความที่อ่านแล้ว ซึ่งสามารถอธิบายในแต่ละส่วนได้ดังนี้

1. ค้นหาข้อมูล : ใช้สำหรับให้พนักงานโรงพยาบาลค้นหาข้อมูล ซึ่งได้แก่ ข้อมูลพนักงาน ข้อมูลผู้ป่วย ข้อมูลยา และประวัติการรักษา โดยรูปแบบการทำงานจะแสดงตามรูปที่ 4.26



รูปที่ 4.26 หน้าค้นหาข้อมูล

หลังจากที่เราเลือกเมนูค้นหาที่เราต้องการแล้ว ก็จะปรากฏฟอร์มในการค้นหาข้อมูลขึ้นมาให้เราทำการป้อนค่าที่ต้องการค้นหาและคลิกที่ปุ่มค้นหา ก็จะได้ผลลัพธ์ออกมา ถ้าหากว่ามีค่าๆ นั้นอยู่ แต่หากไม่มีก็จะไม่ได้ผลลัพธ์ออกมา

สืบค้นข้อมูลพนักงาน		
ค้นหาจาก:	- รหัสพนักงาน	ค้นหาข้อมูล
ค้นหา:	DC001	

รูปที่ 4.27 หน้าป้อนค่าที่ต้องการค้นหา

รหัส	ชื่อ	ตำแหน่ง	เงินเดือน	วันลาพักร้อน	วันลาพักผ่อน	วันลาคลอด
DC001	น.ส. วิมล ใจดี	นักบัญชี	15,000	15	15	15

รูปที่ 4.28 ผลลัพธ์ที่แสดงออกมาจากการค้นหา

2. ข้อความใหม่ : หน้านี้เอาไว้สำหรับให้พนักงานของโรงพยาบาล ในการตรวจสอบว่ามีการสั่งจ่ายยา หรือเวชภัณฑ์อะไรแพทย์ผู้ทำการรักษาหรือไม่ โดยจะแสดงรายละเอียดว่ามีผู้ป่วยชื่ออะไร ห้องไหน เตียงอะไร ซึ่งเราสามารถคลิกที่เมนูลำดับที่ เพื่อให้แสดงรายละเอียดว่ามีคำสั่งอะไรบ้าง จากนั้นเมื่อทำการอ่านเสร็จ หน้าจอจะทำการ Refresh ตัวเองเพื่อเปลี่ยนสถานะของข้อความว่าเป็นอ่านแล้วไปแสดงอีกเมนูหนึ่งแทน โดยรูปแบบจะแสดงดังรูปที่ 4.29

ลำดับที่	ชื่อผู้ป่วย	ห้อง	เตียง	วันที่รับยา	ชื่อแพทย์	ชื่อเวชภัณฑ์	จำนวน	สถานะ
9	นางสาววิมลใจดี	B11005	-	-	-	ยังไม่ได้รับ	-	ยังไม่ได้รับ
10	นางสาววิมลใจดี	B11003	-	-	-	ยังไม่ได้รับ	-	ยังไม่ได้รับ
15	นางสาววิมลใจดี	B11003	-	-	-	ยังไม่ได้รับ	-	ยังไม่ได้รับ
35	นางสาววิมลใจดี	B21002	9	-	-	ยังไม่ได้รับ	-	ยังไม่ได้รับ
6	นางสาววิมลใจดี	B21006	7	-	-	ยังไม่ได้รับ	-	ยังไม่ได้รับ

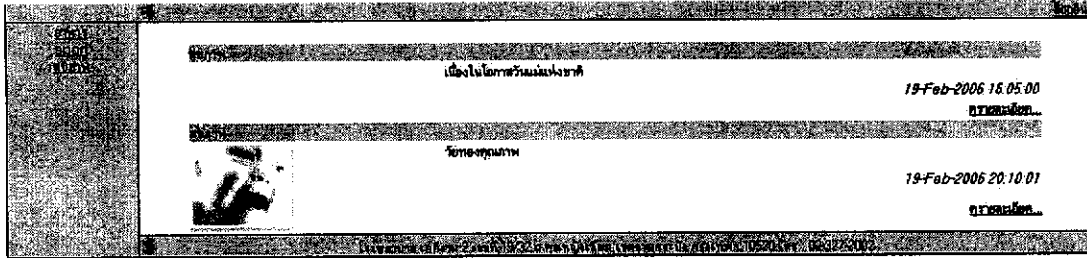
รวมจำนวน	#B2406
จำนวนเตียง	7
ประเภทห้อง	TOTAL(Male)
เตียงนอน	เตียงสำหรับชาย
วันที่	2/2/2549
เวลา	9.15
ชื่อแพทย์	
ยา/เวชภัณฑ์	VitaminB

รูปที่ 4.29 หน้าแสดงข้อความใหม่

3. ข้อความที่อ่านแล้ว : เอาไว้แสดงกรณีที่พนักงานลืม โดยจะแสดงข้อมูลเฉพาะที่สถานะเป็นอ่านแล้วเท่านั้น ส่วนการดูข้อมูลนั้นเหมือนกันกับรูปที่ 4.29

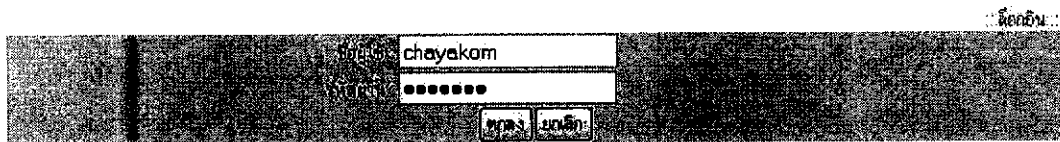
- ส่วนข่าวประชาสัมพันธ์

เป็นหน้าที่เอาไว้สำหรับประกาศข่าวสารของทาง โรงพยาบาลเพื่อให้คนที่เข้ามาเยี่ยมชมได้รับทราบ ส่วนนี้สามารถเข้าชมได้ทุกคน



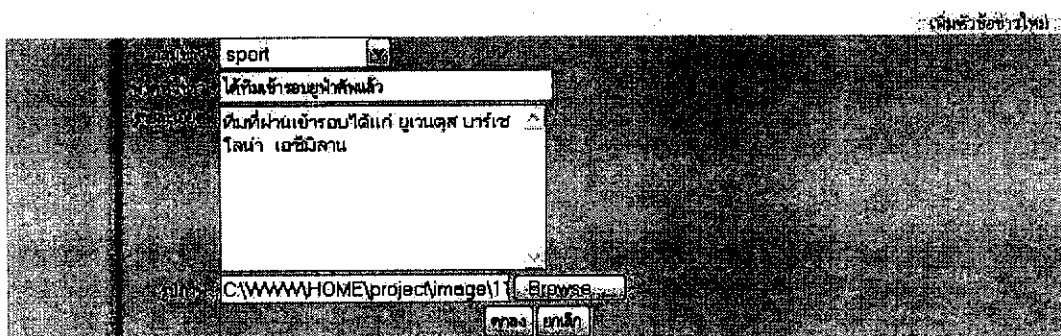
รูปที่ 4.30 หน้าข่าวประชาสัมพันธ์

การดูข้อมูลก็เพียงแค่คลิกที่รายละเอียดของแต่ละหัวข้อข่าวก็สามารถอ่านรายละเอียดได้ โดยในส่วนของด้านซ้าย จะเป็นเมนูเอาไว้สำหรับเลือกที่จะอ่านประเภทข่าวแบบไหน โดยในส่วนหัวข้อต่างๆ นั้นจะเกิดจากผู้ดูแลในส่วนของข่าวประชาสัมพันธ์เป็นคนตั้งขึ้นมา โดยจะต้องล็อกอินเข้าไปก่อนจึงสามารถสร้างได้ โดยรูปที่ 4.31 เป็นหน้าล็อกอินเพื่อเข้าไปตั้งหัวข้อข่าวหรือประเภทของข่าว



รูปที่ 4.31 หน้าล็อกอินของข่าวประชาสัมพันธ์

หลังจากที่ผู้ดูแลได้ทำการล็อกอินแล้วก็สามารถที่จะ เพิ่มกลุ่มข่าว ลบกลุ่มข่าว แก้ไขกลุ่มข่าว รวมไปถึงจัดการเรื่องหัวข้อข่าวได้อีกด้วย โดยในที่นี้จะยกตัวอย่างการสร้างหัวข้อข่าวใหม่ สามารถทำได้ดังรูปที่ 4.32



รูปที่ 4.32 สร้างหัวข้อข่าวใหม่

- ส่วนของผู้ดูแลระบบ

จำกัดไว้เฉพาะผู้ดูแลระบบเท่านั้นที่สามารถทำงานได้ โดยภายในจะประกอบไปด้วยเมนูในการจัดการเกี่ยวกับการเพิ่ม ลบ แก้ไข ข้อมูลพนักงาน การจัดหมวดหมู่ความเชี่ยวชาญ การค้นหาข้อมูล และการดูแลในเรื่องของผู้ใช้ทุกคนในระบบ ซึ่งจะได้กล่าวถึงในแต่ละส่วนต่อไป

1. เพิ่มข้อมูลบุคลากร : เข้าไปที่เมนูเพิ่มข้อมูลบุคลากร จากนั้นก็ทำการป้อนข้อมูลตามที่ได้ระบุไว้แล้วคลิกที่ปุ่มเพิ่มบุคลากร จะปรากฏข้อความว่ามีการเพิ่มบุคลากรลงฐานข้อมูลเรียบร้อยแล้วในรูปแบบที่ 4.34

**เพิ่มบุคลากรใหม่**

ID :

ชื่อ-สกุล :

ความเชี่ยวชาญ :

ที่อยู่ :

โทรศัพท์มือถือ :

โทรศัพท์บ้าน :

อีเมล :

ไฟล์รูปภาพ :

รูปที่ 4.33 แสดงการเพิ่มพนักงานใหม่



รูปที่ 4.34 แสดงการแจ้งผลลัพธ์หลังจากที่เพิ่มข้อมูลเรียบร้อยแล้ว

2. แก้ไข/ลบ รายละเอียดของพนักงาน : ผู้ดูแลระบบสามารถคลิกเลือกรายชื่อของพนักงานที่ต้องการแก้ไข หรือลบได้ โดยคลิกเลือกที่ชื่อถ้าหากต้องการแก้ไข หรือถ้าจะลบก็เลือกเช็คที่รายชื่อของพนักงานที่ต้องการลบ

เลือก	ID	ชื่อ-สกุล	E-mail
<input type="checkbox"/>	D001	วิฑิต เพ็ญนนท์พงษ์	s6015342@kmitp.ac.th
<input type="checkbox"/>	D003	พวงมณี พิภพพิทักษ์	pong_mani@hotmail.com
<input type="checkbox"/>	D002	ชัชวาล ชินวัตร	chinnawat@hotmail.com
<input type="checkbox"/>	D005	วิโรจน์ พงษ์สุวรรณ	wirod_pong@hotmail.com

รูปที่ 4.35 แสดงรายชื่อของพนักงานที่มีในฐานข้อมูล

จากรูปที่ 4.35 ถ้าผู้ดูแลระบบต้องการแก้ไขข้อมูลก็สามารถทำได้โดยการเลือกคลิกที่ชื่อของพนักงานที่ต้องการแก้ไขแล้วก็เข้าไปแก้ไขได้ตามรูปที่ 4.36

แก้ไขข้อมูลพนักงาน

ID:

ชื่อ-สกุล:


ความเชี่ยวชาญ:

ที่อยู่:

โทรศัพท์มือถือ:

โทรศัพท์บ้าน:

อีเมล:



รูปที่ 4.36 แสดงหน้าฟอร์มหลังจากที่ผู้ดูแลระบบทำการเลือกรายชื่อ

สำหรับการลบก็ทำเช่นเดียวกันแต่ต้องเลือก checkbox หน้ารหัสพนักงานแล้วก็คลิกปุ่มลบรายชื่อพนักงาน เพียงแค่นี้ก็สามารถลบรายชื่อออกจากฐานข้อมูลได้แล้ว



รูปที่ 4.37 แสดงหน้าฟอร์มการลบรายชื่อพนักงาน

3. การจัดหมวดหมู่ความเชี่ยวชาญ : การทำงานในส่วนนี้จะเหมือนกันกับการเพิ่มลบบข้อมูลบุคลากรคือสามารถเพิ่ม ลบ แก้ไข ได้

4. ค้นหาข้อมูล : ในส่วนนี้การทำงานจะเหมือนกันกับในส่วนของพนักงานโรงพยาบาลที่มีเมนูให้เลือกค้นหา ฟอร์มรับคำที่ต้องการค้นหา และแสดงผลลัพธ์ในการค้นหาออกมาที่หน้าจอ

5. เปลี่ยนรหัสผ่าน : ผู้ดูแลระบบสามารถจัดการ Username และ Password ของผู้ใช้ได้ทุกประเภท โดยสามารถเพิ่ม ลบ หรือแก้ไขข้อมูลได้ โดยแสดงดังรูปที่ 4.38

รหัสผ่าน	ชื่อจริง	ชื่อเล่น	ชื่อผู้ใช้	ชื่อรหัส	ประเภท	สถานะ
วิโรจน์	วิโรจน์	วโรด	dc001	พนักงาน	แก้ไข	ลบ
ทิพย์ธนา	พาเจริญ	admin	admin	ผู้ดูแล	แก้ไข	ลบ
ชนกคณ	ภิญญาศักดิ์	nengn	sonypl2	สมาชิก	แก้ไข	ลบ
วรินทร์	ศันญาเรือง	warin	123456	สมาชิก	แก้ไข	ลบ
สุรพงษ์	สมรัตน์	surapong	dc002	พนักงาน	แก้ไข	ลบ
สมส	หงษ์วงศ์	kmol	dc003	พนักงาน	แก้ไข	ลบ
สุวิธรา	สุเมธา	sumitta	dc004	พนักงาน	แก้ไข	ลบ
อนาเทพ	ศรีโสภกา	arnon	dc005	พนักงาน	แก้ไข	ลบ
สฤพร	สุรัตน์สุตธาน	sathapon	dc006	พนักงาน	แก้ไข	ลบ
วิหิต	เพ็ญนุชทรัพย์	withit	a	สมาชิก	แก้ไข	ลบ
ไชยา	ฉัตรชัย	chaiya	chaiya	สมาชิก	แก้ไข	ลบ
ชนกคณ	ภิญญาศักดิ์		sonypl2	สมาชิก	แก้ไข	ลบ
ชนกคณ	ภิญญาศักดิ์	john	sonypl2	สมาชิก	แก้ไข	ลบ

๑๘

<< เพิ่มผู้ใช้ใหม่ >>

รูปที่ 4.38 แสดงหน้าจัดการเกี่ยวกับผู้ใช้งานระบบ

จากรูปที่ 4.38 ผู้ดูแลระบบสามารถแก้ไขข้อมูลโดยสามารถที่จะ เปลี่ยนรหัสผ่าน โดยรับรหัสผ่านเก่า และรหัสผ่านใหม่ โดยมีเงื่อนไขในการตรวจสอบภายในนั้นด้วย โดยรูปที่แสดงการแก้ไขข้อมูลแสดงในรูปที่ 4.39

เปลี่ยนชื่อระบบ

ชื่อจริง:

นามสกุล:

ชื่อผู้ใช้:

รหัสผ่านเก่า:

รหัสผ่านใหม่:

สถานะ:  ▼

รูปที่ 4.39 แสดงการแก้ไขข้อมูลผู้ใช้ของผู้ดูแลระบบ

จากรูปที่ 4.39 ผู้ดูแลระบบสามารถเปลี่ยนสถานะของผู้ใช้ได้ว่าจะให้เป็น สมาชิกทั่วไป พนักงาน หรือผู้ดูแลระบบ ก็สามารทำได้ โดยในส่วนของกรลบบก็เพียงคลิกที่เมนูลบที่รายชื่อ นั้นได้เลย

## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 บทสรุป

จากสภาพการณ์ปัจจุบันประเทศไทยกำลังประสบปัญหาทางด้านสุขภาพเพิ่มมากขึ้น เมื่อเทียบกับเมื่อสมัยก่อน อาจสืบเนื่องมาจากมลพิษต่างๆ ที่เพิ่มมากยิ่งขึ้น ทำให้คนเรามีสุขภาพร่างกายที่ไม่แข็งแรงเหมือนแต่ก่อน จากปัญหาสุขภาพนั้นจึงได้มีการสร้างระบบการรักษาแบบใหม่ขึ้นมาแทนแบบเดิมที่ยังไม่ดีเท่าที่ควร ประกอบกับเทคโนโลยีการสื่อสารแบบไร้สายและอินเทอร์เน็ต (Wireless Communication & Internet) มีการเพิ่มขึ้นอย่างรวดเร็ว เราจึงได้นำเอาเทคโนโลยีเหล่านั้นมาประยุกต์ใช้กับระบบของโรงพยาบาล ที่ฟังของคนเจ็บไข้ได้ป่วย ทั้งนี้ก็เพื่ออำนวยความสะดวกในด้านสุขภาพของพวกเขาให้ได้รับการดูแลที่มีประสิทธิภาพขึ้น

จากปัญหาความไม่สะดวกในการรักษาทำให้เกิดระบบขึ้นมาใหม่ ซึ่งเราเรียกว่าระบบการจัดการโรงพยาบาลด้วยอุปกรณ์ประมวลผลแบบพกพา (Mobile Computing Application in Hospital) จึงเป็นทางเลือกหนึ่งเข้ามาช่วยให้เกิดการจัดการข้อมูลที่มีประสิทธิภาพมากยิ่งขึ้น เนื่องจากระบบบริหารงานแบบเดิมจะเน้นการจดบันทึกเป็นหลัก ซึ่งแน่นอนว่าต้องมีการผิดพลาดกันได้ ระบบนี้จึงใช้ข้อดีของเหล่านี้มาปรับปรุงให้ดีขึ้น ด้วยการใช้อุปกรณ์ประมวลผลพกพาที่เรียกว่า Pocket PC ทำการบันทึกลงฐานข้อมูลแทนการจดบันทึก โดย ผู้ใช้ซึ่งเป็นแพทย์จะทำการป้อนรหัสผ่านเพื่อตรวจสอบสิทธิ์ในการใช้ จากนั้นก็จะทำการเรียกดูข้อมูลของผู้ป่วยที่เขาจะทำการตรวจ จากนั้นก็ทำตามกระบวนการในการตรวจ ที่ได้ทำเป็นเมนูไว้ให้เลือกใช้ตามความเหมาะสม จากนั้นเมื่อป้อนเสร็จข้อมูลก็จะบันทึกลงฐานข้อมูล อีกส่วนหนึ่งแพทย์หรือพนักงานก็สามารถเข้าดูประวัติคนไข้หรือข้อมูลอื่นๆ ที่สามารถดูได้ผ่านทางเว็บไซต์ของทางโรงพยาบาลเอง ซึ่งการทำงานก็จะคล้ายกันคือ การตรวจสอบสิทธิ์ก่อนการใช้งาน และที่สำคัญที่สุดก็คือทั้ง Pocket PC และ เว็บจะต้องทำงานสัมพันธ์กันภายใต้ฐานข้อมูลเดียวกัน

### 5.2 วิจารณ์สิ่งที่ได้จากโครงการ

Pocket PC สามารถที่จะติดต่อกับฐานข้อมูลได้ แต่ด้วยหน้าจอที่จำกัดจึงต้องมีการคลิกดูหลายหน้าจึงจะทราบข้อมูลทั้งหมด รวมไปถึงการส่งข้อมูลระหว่าง Pocket PC กับคอมพิวเตอร์ที่ทำหน้าที่ในการรอรับ request ยังต้องใช้ Application เดียวกันในการพัฒนาอยู่ ซึ่งถ้าจะให้ดีควรสามารถส่งข้อมูลผ่านทางเว็บได้ด้วยจึงจะมีประสิทธิภาพกว่า

### 5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1. การหาข้อมูลจากโรงพยาบาลทำได้ไม่เต็มที่เท่าที่ควร เนื่องจากข้อมูลส่วนใหญ่เป็นความลับ จึงใช้การสอบถามกับพยาบาลที่รู้จักแทนการขอเอกสาร
2. โปรแกรมที่เขียนด้วย C# ไม่สามารถติดต่อกับฐานข้อมูลได้โดยตรง จึงต้องมีการใช้หลักการของ Web Service เข้าช่วย

### 5.4 แนวทางการพัฒนาต่อ

1. พัฒนาให้สามารถใช้งานร่วมกับแผนกผู้ป่วยนอก และสามารถเพิ่ม ลบ แก้ไข ข้อมูลทั้งในส่วนของผู้ป่วยในและผู้ป่วยนอกได้
2. ปรับปรุงฐานข้อมูลให้สมบูรณ์มากที่สุดและตกแต่งหน้าเว็บให้ดูทันสมัยมากยิ่งขึ้น
3. กรณีมีสาขาของโรงพยาบาล โปรแกรมสามารถที่จะติดต่อกับสาขาผู้ให้บริการได้

## บรรณานุกรม

- [1] eITCenter. "Microsoft eMbedded Visual Tools 3.0" Available:  
:http://www.eitcenter.com
- [2] กิตติศักดิ์ เจริญโภคานนท์ 2537 พัฒนาเว็บแอปพลิเคชัน PHP 4 By Example  
สำนักพิมพ์ ชัคเชส มีเดีย จำกัด กรุงเทพฯ
- [3] วรณิกา เนตรงาม 2544 พื้นฐานการเขียนสคริปต์และสร้าง Web Application ด้วย PHP &  
MySQL สำนักพิมพ์อินโฟเพลส กรุงเทพฯ
- [4] อานาจ มีมงคล การออกแบบและติดตั้งเครือข่าย Wireless LAN สำนักพิมพ์อินโฟเพลส  
กรุงเทพฯ
- [5] มณีโชติ สมานไทย 2544 สร้างเว็บเพจแบบมืออาชีพด้วย HTML สำนักพิมพ์อินโฟเพลส  
กรุงเทพฯ
- [6] ดร. สุภมิตร จิตตะยโสธร, "เอกสารประกอบการสัมมนา Database Design and SQL",  
(สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง)
- [7] ศุภชัย สมพานิช 2546 คู่มือการเขียนโปรแกรมและใช้งาน Visual C# .NET ฉบับสมบูรณ์  
พิมพ์ที่ บริษัท ด้านสุทธาการพิมพ์ จำกัด