

การพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต

Web-based Messenger

โดย

นาย พิบูล เทพบุตร

รหัส 42067017



H001848

อาจารย์ที่ปรึกษา

ดร. จันทร์บุรณธ์ สติตวิริยวงศ์

วัน เดือน ปี.....	15	ธ.ค.	2550
เลขทะเบียน.....	01848		
เลขเรียกหนังสือ.....	วศท. พ 6927	2544	
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต
นักศึกษา	นาย พิบูล เทพบุตร
อาจารย์ที่ปรึกษา	ดร. จันทร์บุรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ในปัจจุบัน โปรแกรมที่ใช้เพื่ออำนวยความสะดวกในการสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตเข้ามามีบทบาทสำคัญในระบบเครือข่ายอินเทอร์เน็ต ซึ่งแต่ละโปรแกรมอาจเกิดความยุ่งยากในการติดตั้งและใช้งานกับผู้ใช้ อีกทั้งการทำงานของ โปรแกรมเหล่านั้นยังขึ้นกับ Platform และเสี่ยงต่อการคุกคามของไวรัสคอมพิวเตอร์

โครงการพัฒนานี้มีจุดมุ่งหมายเพื่อทำระบบการสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตซึ่งได้ออกแบบระบบที่ทำให้ผู้ใช้ไม่จำเป็นต้องทำการปรับเปลี่ยนหรือติดตั้งโปรแกรมใด ๆ ในเครื่องคอมพิวเตอร์ของผู้ใช้, สามารถใช้งานระบบนี้ได้กับทุก Platform และ ทุกสถานที่ที่สามารถเชื่อมต่ออินเทอร์เน็ตได้ โดยใช้หลักการทำงานเชื่อมต่อซ็อกเก็ตผ่านการทำงานของเว็บเบราว์เซอร์ และ เซิร์ฟเวอร์เพื่อทำการติดต่อสื่อสารผ่านระบบเครือข่าย โดยหลักการดังกล่าวทำให้สามารถที่จะส่งข้อมูลระหว่างผู้ใช้ได้ และ ด้วยข้อจำกัดของเว็บเบราว์เซอร์และ Java Applet ทำให้ผู้ใช้ปลอดภัยต่อการคุกคามของไวรัสคอมพิวเตอร์

Title	Web based Messenger
Student	Mister Pibul Theppabutr
Advisor	Chanboon Sathitwiriawong, Ph.D.
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

ABSTRACT

Today, several messenger programs have become more important things in the Internet communication among users. But many times they are also difficult for users how to install them. More over, they are usually depending on platform and easy to infected by viruses.

Web-based Messenger project has designed to ease to use without installation. Web-based Messenger can run on every computer platform and every place that connected the Internet. Web-based Messenger has been developed using socket connection for inter process communication on web browser and communicate through server. These techniques have made the communication between uses through the Internet. Additional, because of the restrictions of web browser and Java Applet, Web-based Messenger has become secure system that cannot be infected by viruses.

กิตติกรรมประกาศ

ในการศึกษาและพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตได้สำเร็จลุล่วงไป ด้วยความสนับสนุนเป็นอย่างดีจากบุคคลหลายท่าน ที่ได้ให้คำแนะนำ ปรีกษา และ แนะนำแนวทางทำให้การศึกษาโครงการนี้บรรลุตามเป้าหมายที่ได้วางไว้ ผู้จัดทำจึงขอใคร่ขอบพระคุณ

1. มารดาผู้ให้กำเนิด เลี้ยงดูเอาใจใส่ดูแล เป็นที่ปรึกษาให้กำลังใจในการศึกษา และ ทำงาน ให้การอบรมให้ประพฤติในสิ่งที่ดีและถูกต้อง ตลอดจนการส่งเสริมในด้านการศึกษาเป็นอย่างดีที่สุด
2. ดร. จันทร์บุรณ สติศิววิริยวงศ์ อาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ คำปรึกษา ในการจัดทำโครงการ อีกทั้งยังเป็นผู้ที่มีความรู้ความสามารถอย่างแท้จริงที่ได้สั่งสอนให้ ความรู้แก่ผู้จัดทำในหลาย ๆ วิชาซึ่งถือเป็นแนวทางที่มีผลทำให้ผู้จัดทำมีความรู้ทั้งใน ด้านการปรับปรุงคุณภาพของงานจนถึงเรื่องความปลอดภัยของระบบการทำงานที่ดี
3. อาจารย์ทุก ๆ ท่านในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้า คุนทหารลาดกระบัง ที่ได้อบรมให้ความรู้ ทำให้สามารถนำความรู้ที่มีไปแก้ปัญหาหา แนวทางทำให้โครงการสำเร็จลุล่วงลงด้วยดี
4. นายมนต์ชัย พงนาสมสมาน สำหรับคำแนะนำที่ดีทำให้ผู้จัดเริ่มต้นการศึกษาที่ถูกแนว ทาง แนวคิดในการแก้ปัญหา อีกทั้งเป็นผู้แนะนำแหล่งความรู้ทำให้โครงการดำเนินไป ได้ด้วยดี

นายพิบูล เทพบุตร

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการพัฒนาระบบงาน.....	1
1.3 ขอบเขตของโครงการศึกษาและพัฒนาระบบงาน.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
2. หลักการพื้นฐานการส่งข้อมูลในระบบเครือข่าย.....	3
2.1 ทฤษฎีการส่งข้อมูลในระบบเครือข่าย.....	3
2.1.1 โพรโทคอลในระบบเครือข่าย.....	3
2.1.2 Internet Protocol.....	4
2.1.3 User Datagram Protocol.....	5
2.1.4 รูปแบบของไอพีแอดเดรส.....	6
2.1.5 Transmission Control Protocol.....	7
2.2 การส่งข้อมูลในระบบเครือข่ายด้วยซ็อกเก็ต.....	9
2.2.1 แนวคิดการเขียนโปรแกรมเครือข่าย.....	10
2.2.2 ขั้นตอนการทำงานของซ็อกเก็ต.....	10
2.2.2.1 สร้างซ็อกเก็ต.....	10
2.2.2.2 แอดเดรสของซ็อกเก็ต (Socket Address).....	12
2.2.2.3 การเชื่อมต่อซ็อกเก็ต.....	13
2.2.2.4 การรอการร้องขอจากซ็อกเก็ตไคลเอนต์ในกรณีของซ็อกเก็ตเซิร์ฟเวอร์.....	13

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ใดเห็นประโยชน์ในการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ดำเนินการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่

2.2.2.5 การร้องขอไปที่ซ็อกเก็ตเซิร์ฟเวอร์ในกรณีของซ็อกเก็ต โคลเอนต์.....	14
2.2.2.6 การตอบรับการเชื่อมต่อ	14
2.2.2.7 การยกเลิกการเชื่อมต่อ	14
3. การพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต	16
3.1 การวิเคราะห์และการออกแบบ.....	16
3.1.1 Context Diagram ของระบบ	16
3.1.2 Data Flow Diagram ของระบบ	16
3.1.2.1 การสมัครสมาชิกเพื่อเข้าใช้งานระบบ	16
3.1.2.2 การเข้าใช้งานระบบของสมาชิก.....	17
3.1.2.3 การส่งข้อความระหว่างผู้ใช้	18
3.1.2.4 การเพิ่มรายชื่อผู้ใช้อื่นที่ต้องการติดต่อในรายการติดต่อของผู้ใช้	18
3.1.2.5 การลบรายชื่อผู้ใช้อื่นนอกจากรายการติดต่อของผู้ใช้	19
3.1.2.6 การแก้ไขข้อมูลส่วนตัวของผู้ใช้	20
3.1.2.7 การยกเลิกการเป็นสมาชิก	21
3.1.3 System Data Flow Diagram.....	22
3.1.4 Entity Relationship ของระบบ.....	23
3.2 การพัฒนาโปรแกรม.....	25
3.2.1 การพัฒนาโปรแกรมฝั่งเซิร์ฟเวอร์.....	25
3.2.1.1 การพัฒนาโปรแกรมส่วนกลาง	25
3.2.1.2 การพัฒนา โปรแกรมเพื่อใช้ในการสื่อสารระหว่างผู้ใช้และเซิร์ฟเวอร์.....	29
3.2.2 การพัฒนาโปรแกรมฝั่งไคลเอนต์	30
3.2.3 ข้อตกลงในการส่งข้อมูล (Messaging Protocol)	33
3.2.4 การทำงานของระบบและ โพรโตคอลที่ใช้ในการสื่อสาร	34
3.3 ตารางฐานข้อมูลระบบ	44
3.4 ขั้นตอนการพัฒนาโปรแกรม	45
3.4.1 พัฒนาระบบ CGI เพื่อใช้ในการรับข้อมูลผู้ใช้ระบบ	45

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลระบบเห็นชอบที่จะเผยแพร่เอกสารนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่	
3.4.2 พัฒนาระบบส่วนกลาง	47
3.4.3 พัฒนาระบบติดต่อผู้ใช้ผ่าน socket บนเว็บเบราว์เซอร์	47
4. ผลการทดลอง.....	48
4.1 การทำงานฝั่งเซิร์ฟเวอร์.....	48
4.2 การทำงานของระบบในฝั่งไคลเอนต์.....	49
4.2.1 การสมัครสมาชิกเพื่อเข้าใช้งานระบบ.....	50
4.2.2 การเข้าใช้งานระบบ.....	51
4.2.2.1 การเพิ่มรายชื่อผู้ใช้คนอื่นเข้าไปในรายการติดต่อของผู้ใช้.....	54
4.2.2.2 การส่งข้อความเพื่อสื่อสารกับผู้อื่น.....	56
4.2.2.3 การลบรายชื่อผู้อื่นในรายการติดต่อของผู้ใช้.....	58
4.2.3 การแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ.....	58
4.2.4 การยกเลิกการเป็นสมาชิกระบบ	59
5. สรุปผลการพัฒนา.....	61
5.1 ความแตกต่างระหว่างระบบที่ได้พัฒนาขึ้นและระบบที่มีอยู่ในปัจจุบัน	61
5.2 ประโยชน์ที่ได้รับจากการทำงานของระบบ.....	62
5.3 ปัญหาและข้อจำกัดของระบบ	63
5.4 แนวทางแก้ไข.....	63
ภาคผนวก	64
1. วิธีการติดตั้งระบบ	64
2. วิธีการใช้งานระบบ	66
บรรณานุกรม.....	73
ประวัติผู้เขียน	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่	
3.1 แสดงความหมายของแต่ละส่วนของ โพรโตคอลฝั่งเซิร์ฟเวอร์.....	41
3.2 แสดงความหมายของแต่ละส่วนของ โพรโตคอลฝั่งไคลเอนต์	43
3.3 ข้อมูลผู้ใช้งานระบบ (contact)	44
3.4 ข้อมูลรายการติดต่อของผู้ใช้งานระบบ (contactlist)	45
3.5 ข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างผู้ใช้งานระบบ (message).....	45
5.1 แยกต่างระหว่างระบบที่ได้พัฒนาขึ้นและระบบที่มีอยู่ในปัจจุบัน	61



สารบัญภาพ

รูปที่	หน้า
2.1 แสดงชั้นของ โพรโตคอลในระบบเครือข่ายที่มีการใช้งาน TCP/IP	3
2.2 แสดง IP datagram layout	4
2.3 แสดง UDP layout.....	5
2.4 แสดงความสัมพันธ์ระหว่าง IP datagram และ UDP datagram.....	5
2.5 แสดงรูปแบบของ ไอพีแอดเรสคลาส C.....	6
2.6 แสดงรูปแบบของ TCP segment.....	7
2.7 แสดงความสัมพันธ์ระหว่าง IP datagram และ TCP segment	8
2.8 แสดงการแปลง IP แอดเรส 161.246.10.21 เป็นหมายเลขเครื่อง	12
2.9 แสดงการหาแอดเรสของเครื่องและแอดเรสของซ็อกเก็ต	12
2.10 แสดงโครงสร้างแอดเรสของซ็อกเก็ต	13
2.11 ขั้นตอนการทำงานของซ็อกเก็ต โคลเอนต์และซ็อกเก็ตเซิร์ฟเวอร์	15
3.1 Context Diagram ของระบบ	16
3.2 DFD แสดงการสมัครสมาชิก	16
3.3 แสดง DFD การเข้าใช้งานระบบ	17
3.4 แสดง DFD การใช้งานระบบสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต	18
3.5 แสดงการเพิ่มรายชื่อผู้ใช้อื่นที่ต้องการติดต่อในรายการติดต่อของผู้ใช้.....	18
3.6 แสดงการลบรายชื่อผู้ใช้อื่นออกจากรายการติดต่อของผู้ใช้.....	19
3.7 แสดงการแก้ไขข้อมูลส่วนตัวของผู้ใช้.....	20
3.8 แสดงการยกเลิกการเป็นสมาชิกของผู้ใช้.....	21
3.9 System Diagram.....	22
3.10 ER Diagram	23
3.11 การเชื่อมต่อระหว่าง Server Socket และ Client Socket	26
3.12 แสดงรูปแบบของ โพรโตคอลที่ใช้ในระบบ	33
4.1 แสดงการทำงานของโปรแกรมทำงานส่วนกลาง daemon.pl.....	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

หน้า

รูปที่

4.2 แสดงการเรียกใช้ระบบสื่อสารผ่านทางเว็บเบราว์เซอร์	49
4.3 แสดงหน้าจอสมัครสมาชิก.....	50
4.4 แสดงหน้าจอสมัครเสร็จสมบูรณ์พร้อมทั้งหมายเลขประจำตัวผู้ใช้.....	51
4.5 แสดงหน้าจอเพื่อให้ผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่าน.....	51
4.6 แสดงหน้าจอใช้ในการสื่อสาร	52
4.7 แสดงการทำงานของโปรแกรมส่วนกลางในการตอบรับการทำงานของผู้ใช้	53
4.8 แสดงการตอบรับผู้ใช้รายที่สองที่เข้าใช้ระบบของโปรแกรมส่วนกลาง	53
4.9 แสดงหน้าจอกำหนดหมายเลขประจำตัวผู้ใช้อื่นเพื่อเพิ่มในรายการติดต่อของผู้ใช้.....	54
4.10 แสดงหน้าจอเพื่อให้ผู้ใช้รายอื่นทราบว่าได้รับการเพิ่มชื่อเพื่อทำการติดต่อ.....	54
4.11 แสดงการเพิ่มรายชื่อผู้ใช้รายอื่นในระบบเข้ารายการติดต่อของผู้ใช้ได้สำเร็จ	55
4.12 แสดงหน้าจอการสื่อสารของผู้ใช้ที่ได้รับการเพิ่มรายชื่อในรายการติดต่อของผู้ใช้....	55
4.13 แสดงหน้าจอที่ใช้ในการส่งข้อความระหว่างผู้ใช้และผู้ใช้รายอื่นที่ต้องการติดต่อ	56
4.14 แสดงหน้าจอสื่อสารของผู้ใช้อื่นที่ผู้ใช้ติดต่อด้วย.....	56
4.15 แสดงหน้าจอสื่อสารข้อมูลที่สื่อสารกันทางฝั่งผู้ใช้อื่น	57
4.16 แสดงหน้าจอสื่อสารฝั่งผู้ใช้.....	57
4.17 แสดงข้อความเตือนการลบรายชื่อออกจากรายการติดต่อของผู้ใช้	58
4.18 แสดงหน้าจอข้อมูลส่วนตัวของผู้ใช้ระบบ.....	58
4.19 แสดงหน้าจอการแสดงการจัดเก็บข้อมูลที่ผู้ทำการเปลี่ยนแปลงสำเร็จ	59
4.20 แสดงหน้าจอยืนยันการยกเลิกการเป็นสมาชิกของระบบ.....	60
4.21 แสดงหน้าจอตอบรับการยกเลิกการเป็นสมาชิกของระบบ	60

ภาคผนวก

รูปที่

1 แสดงหน้าจอเริ่มดำเนินการทำงานของระบบ	65
2 แสดงหน้าจอสำหรับการสมัครสมาชิก	65

เอกสารนี้ 3 แสดงหน้าจอแจ้งหมายเลขประจำตัวสมาชิก.....

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

ภาคผนวก

รูปที่

4 แสดงหน้าจอเข้าระบบ	66
5 แสดงหน้าจอสำหรับการสื่อสารของผู้ใช้.....	67
6 แสดงหน้าจอสำหรับกรอกหมายเลขผู้ใช้อื่น.....	67
7 แสดงหน้าจอเมื่อผู้ใช้เพิ่มรายชื่อสำเร็จ.....	68
8 แสดงหน้าจอสำหรับการสื่อสาร	68
9 แสดงหน้าจอแสดงข้อมูลส่วนตัวของผู้ใช้อื่น	69
10 แสดงหน้าจอเพื่อยืนยันการลบ.....	69
11 แสดงหน้าจอเพื่อใช้ในการแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ.....	70
12 แสดงหน้าจอเพื่อใช้ในการยกเลิกการเป็นสมาชิกของผู้ใช้ระบบ.....	70

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันอินเทอร์เน็ตเข้ามามีบทบาทในชีวิตประจำวันมากขึ้น ทำให้การทำงาน, การค้นหาข้อมูล, การดำเนินธุรกิจ ฯลฯ ดำเนินไปได้ด้วยความสะดวกสบายมากขึ้นกว่าอดีตที่ผ่านมา ข้อดีข้อหนึ่งของระบบเครือข่ายอินเทอร์เน็ตที่เป็นที่ยอมรับกันว่ามีประโยชน์เป็นอย่างมากก็คือช่วยในการสื่อสารสะดวกสบาย ทำให้การใช้งานโปรแกรมที่ใช้ในการสื่อสารบนเครือข่ายอินเทอร์เน็ตเริ่มเข้ามามีบทบาทนอกเหนือจากการส่งข่าวสาร โดยการส่งไปรษณีย์อิเล็กทรอนิกส์ ซึ่งในปัจจุบันได้มีการพัฒนาโปรแกรมประยุกต์เพื่อใช้ในการสื่อสารออกมาหลายโปรแกรมด้วยกัน ตัวอย่างโปรแกรมที่ใช้ในการสื่อสารในปัจจุบัน เช่น โปรแกรม ICQ, MSN messenger, Yahoo Messenger ฯลฯ ซึ่งการที่จะทำให้สามารถใช้งานโปรแกรมสื่อสารดังกล่าวได้นั้น ผู้ใช้ต้องทำการติดตั้งโปรแกรมเหล่านั้นก่อนที่จะทำการใช้งานซึ่งอาจเป็นปัญหาหรือเกิดความยุ่งยากกับผู้ใช้งานที่ไม่มีความรู้ได้ และยังทำให้เกิดการสิ้นเปลืองทรัพยากรของเครื่องที่ใช้ในการสื่อสารอีกด้วย

1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

วัตถุประสงค์ในการพัฒนาระบบงานนี้คือการพัฒนากระบวนการสื่อสารที่ทำงานบนเครือข่ายอินเทอร์เน็ต เพื่อให้ผู้ใช้สามารถใช้งานระบบได้โดยไม่ต้องทำการติดตั้งโปรแกรมใช้งาน สามารถใช้งานได้ทุกสถานที่ที่มีการเชื่อมต่ออินเทอร์เน็ตและมีโปรแกรมเว็บเบราว์เซอร์ ซึ่งมีอยู่แล้วทุกเครื่องที่มีการใช้งานอินเทอร์เน็ต

1.3 ขอบเขตของโครงการศึกษาและพัฒนาระบบงาน

การศึกษาจะเริ่มจากการศึกษาโปรแกรมสื่อสารที่มีการใช้งานอยู่ในปัจจุบันว่าแต่ละโปรแกรมมีระบบการทำงานเป็นอย่างไรบ้าง แล้วพิจารณาถึงข้อดีข้อเสียของแต่ละระบบการทำงานเพื่อนำมาใช้ในการพัฒนาระบบต่อไป ระบบที่ทำการพัฒนาขึ้นนี้จะใช้เว็บเบราว์เซอร์ที่มีอยู่แล้ว เช่น Internet Explorer ที่มีทุกเครื่องที่ลง Windows หรือ Netscape ที่มีนิยมใช้งานในปัจจุบัน และเป็นที่คุ้นเคยของผู้ใช้ที่ใช้งานอินเทอร์เน็ตอยู่แล้ว ทำให้ผู้ใช้รู้สึกง่ายในการใช้งานระบบรวมถึงผู้ใช้

ไม่ต้องทำการติดตั้งโปรแกรมในเครื่องคอมพิวเตอร์แต่อย่างใด ทำให้ลดขั้นตอนที่ทำให้เกิดความยุ่งยากในการใช้งานของผู้ใช้ที่ไม่มีความรู้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่จะคาดว่าจะได้รับจากการพัฒนาโครงการนี้คือระบบสื่อสารที่สามารถติดต่อสื่อสารได้สะดวกโดยไม่เกิดความยุ่งยากในการใช้ต่อผู้ใช้งาน และ ผู้ใช้งานไม่ต้องทำการติดตั้งโปรแกรมใช้งานเหมือนดังเช่น โปรแกรมประยุกต์ที่ใช้งานอยู่ทั่วไปในปัจจุบัน ผู้ใช้เพียงแต่จำหมายเลขประจำตัวและรหัสผ่านของผู้ใช้เอง ผู้ใช้ก็สามารถที่จะทำการติดต่อสื่อสารได้ทุกสถานที่ไม่ว่าจะเป็นที่ทำงาน ที่บ้าน หรือ แม้แต่ร้านให้บริการอินเทอร์เน็ต โดยไม่จำเป็นต้องติดตั้งโปรแกรมสื่อสารแต่อย่างใด รวมถึงไม่ต้องจดจำหมายเลขของบุคคลอื่นที่ผู้ใช้งานระบบต้องการติดต่อด้วยเพราะระบบได้ทำการจัดเก็บข้อมูลข่าวสารต่าง ๆ ที่ผู้ใช้ต้องการไว้ที่เครื่องให้บริการ เพื่อความสะดวกในการใช้งาน

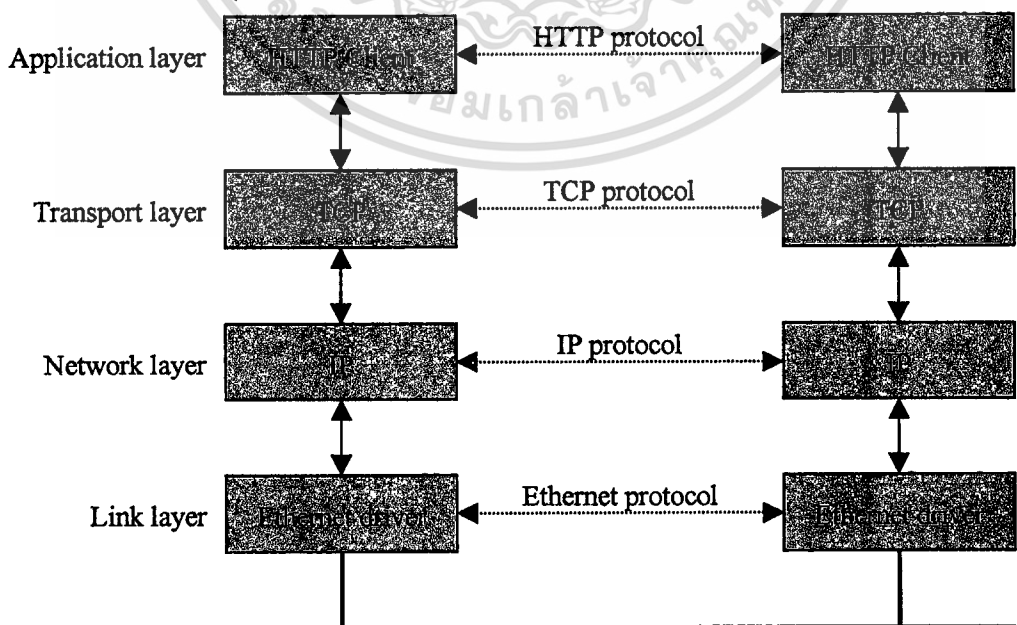
บทที่ 2

หลักการพื้นฐานการส่งข้อมูลในระบบเครือข่าย

2.1 ทฤษฎีการส่งข้อมูลในระบบเครือข่าย

2.1.1 โพรโทคอลในระบบเครือข่าย

ระบบเครือข่ายจะมีการใช้งาน โพรโทคอลที่ใช้ในการสื่อสารเป็นชั้น ๆ ซึ่งเราเรียกว่าสแตค (stack) แอปพลิเคชันจะติดต่อกับโพรโทคอลชั้นที่อยู่ด้านบนและมีการสื่อสารลงไปยังชั้นล่างไปยังระบบเครือข่าย รูปที่ 2.1 แสดงให้เห็นถึงชั้นของโพรโทคอลในระบบเครือข่ายท้องถิ่น (LAN) ที่มีการใช้งานโพรโทคอล TCP/IP แต่ละชั้นมีการทำงานติดต่อกันระหว่างชั้นที่ตรงกัน ในอีกฝั่งหนึ่งที่สอดคล้องกันในช่องทางการสื่อสาร โปรแกรมที่ทำหน้าที่เป็นผู้ให้บริการ (server program) ที่แสดงในรูปที่ 2.1 ด้านขวาจะรอรับสัญญาณตลอดเวลาในฝั่งหนึ่ง ในขณะที่โปรแกรมที่ทำหน้าที่ร้องขอบริการ (client program) ที่แสดงในรูปด้านซ้ายจะติดต่อเข้ามาที่โปรแกรมให้บริการเป็นครั้งคราวเพื่อแลกเปลี่ยนข้อมูลกัน การทำงานของโปรแกรมที่ทำหน้าที่ให้บริการเช่นการทำงานของ World Wide Web server และ โปรแกรมทำหน้าที่ร้องขอบริการก็คือเว็บเบราว์เซอร์ทำงานอยู่ที่เครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้งานอินเทอร์เน็ต



รูปที่ 2.1 ชั้นของโพรโทคอลในระบบเครือข่ายที่มีการใช้งาน TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ประโยชน์ด้านการค้า

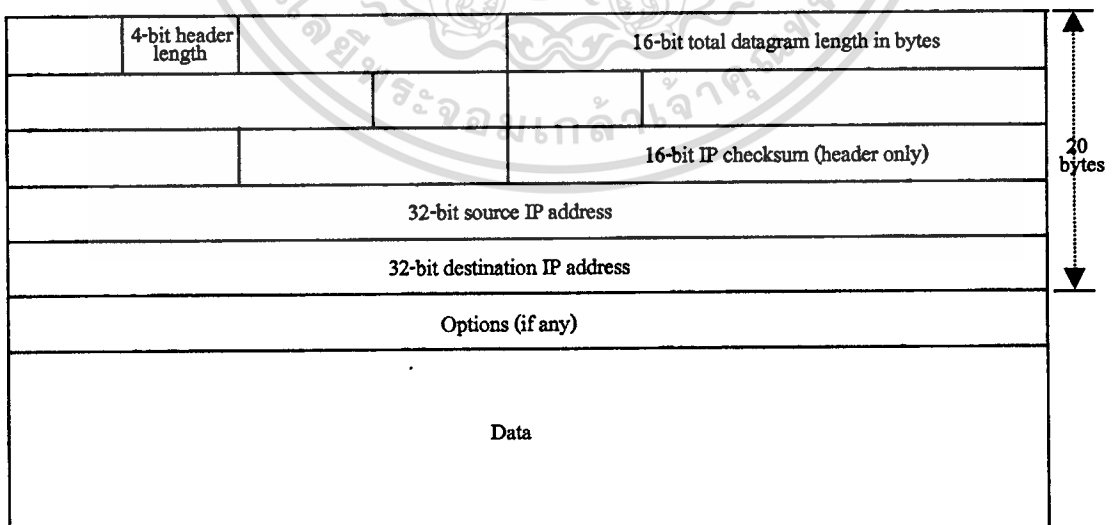
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Internet Protocol

Internet Protocol (IP) เป็นชั้นโพรโทคอลที่กำหนดแพ็กเก็ต (packet) ที่เรียกว่าค่าคำแกรม (datagrams) ซึ่งเป็นหน่วยข้อมูลพื้นฐานที่ใช้ในการสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต ซึ่งโดยทั่วไปแล้วแพ็กเก็ตเหล่านี้จะมีความยาวข้อมูลน้อยกว่า 1,000 ไบต์ ที่ใช้ในการสื่อสารกันเมื่อเปิดเว็บเพจ, ความเร็วโหลดเพิ่มข้อมูล หรือ ส่งเมล รูปที่ 2.2 แสดงให้เห็นถึงรูปแบบของไอพีค่าคำแกรม (IP datagram)

หมายเหตุ ไอพีค่าคำแกรมจะบรรจุแอดเดรสของเครื่องคอมพิวเตอร์ (address) เป็นข้อมูลขนาด 32 บิตทั้งส่วนต้นทาง (source) และส่วนปลายทาง (destination) ไอพีแอดเดรส (IP Address) เหล่านี้เป็นสิ่งที่บ่งบอกว่าเป็นหมายเลขประจำเครื่องคอมพิวเตอร์เฉพาะเครื่องที่ทำงานอยู่บนอินเทอร์เน็ต และ ถูกใช้งานโดยเราเตอร์ (router) ในการกำหนดทิศทางให้กับแต่ละค่าคำแกรมให้ส่งไปยังปลายทางได้ เราเตอร์จะไม่สนใจว่ามีอะไรบรรจุอยู่ในค่าคำแกรม เราเตอร์สนใจเพียงเลขแสดงที่อยู่ปลายทางที่ค่าคำแกรมจะเดินทางไปกับขนาดความยาวของค่าคำแกรมเท่านั้น งานของเราเตอร์คือการส่งค่าคำแกรมให้เร็วที่สุดเท่าที่จะทำได้

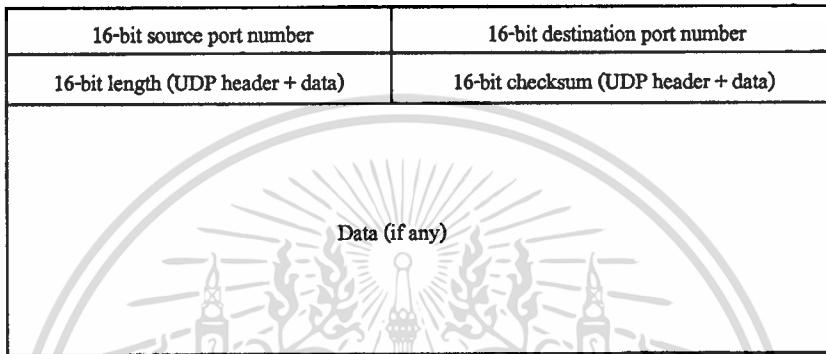
การทำงานชั้นของไอพีจะไม่บอกให้โปรแกรมที่ทำการส่งข้อมูลทราบว่า ได้ส่งถึงปลายทางสำเร็จหรือไม่ ซึ่งจะถือเป็นงานของชั้นโพรโทคอลชั้นบนถัดมาจากชั้นไอพี โปรแกรมที่ทำหน้าที่รับข้อมูลทำได้เพียงดูว่าเช็คซัม (checksum) ที่จะกำหนดได้ว่าส่วนหัวของค่าคำแกรมมีความผิดพลาดเกิดขึ้นหรือไม่



รูปที่ 2.2 IP datagram layout

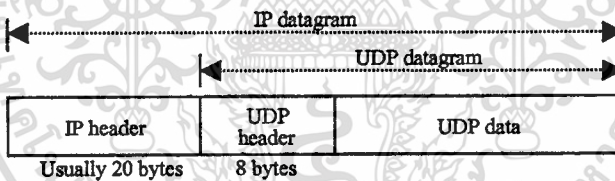
2.1.3 User Datagram Protocol

โพรโทคอล TCP/IP จริง ๆ แล้วควรจะเรียกว่า TCP/UDP/IP โพรโทคอล เพราะว่ามันรวม User Datagram Protocol ไว้ด้วยซึ่งมีการทำงานในระดับเดียวกับ TCP โพรโทคอลทุกโพรโทคอลที่ใช้ IP เป็นพื้นฐานในการส่งข้อมูล จะมีการเก็บข้อมูลส่วนหัว (header) และส่วนของข้อมูลอยู่ภายในบล็อกข้อมูลไอพี (IP data block) ดังในรูปที่ 2.3



รูปที่ 2.3 UDP layout

เค้าโครง UDP/IP ที่สมบูรณ์จะมีรูปแบบดังเช่นรูปที่ 2.4



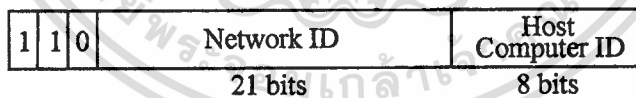
รูปที่ 2.4 ความสัมพันธ์ระหว่าง IP datagram และ UDP datagram

โพรโทคอล UDP เป็นเพียงส่วนย่อยของโพรโทคอล IP แต่แอปพลิเคชันไม่ได้ใช้โพรโทคอล IP โดยตรง เช่นเดียวกับโพรโทคอล IP โพรโทคอล UDP ไม่ได้บอกให้โปรแกรมที่ทำการส่งข้อมูลทราบว่าเค้าโครงถึงปลายทางหรือไม่ การตรวจสอบจะขึ้นกับแอปพลิเคชันเอง ยกตัวอย่างเช่น โปรแกรมที่ทำหน้าที่ส่งข้อมูลอาจต้องการให้โปรแกรมที่ทำหน้าที่รับข้อมูลส่งคำสั่งตอบสนอง (response) กลับมา และโปรแกรมที่ทำหน้าที่ส่งข้อมูลอาจส่งเค้าโครงไปอีกครั้งถ้าคำสั่งตอบสนองไม่ถึงโปรแกรมที่ทำหน้าที่ส่งภายใน 20 วินาทีเป็นต้น โพรโทคอล UDP เป็นโพรโทคอลที่เหมาะสมแก่การรับส่งข้อมูลข่าวสารที่มีขนาดสั้น และ ถูกใช้งานโดยระบบให้บริการโดเมนเนมบนอินเทอร์เน็ต (Internet Domain Name System (DNS)) โพรโทคอล UDP ถูกใช้ในการส่งข้อมูลทางเสียงและภาพในการถ่ายทอดสด

รูปที่ 2.3 แสดงให้เห็นว่าข้อมูลส่วนหัวของ UDP (UDP header) ได้บรรจุข้อมูลบางอย่างเพิ่มเติมเข้ามา กล่าวคือหมายเลขของช่องทางการสื่อสาร (port number) ของต้นทางและปลายทาง ซึ่งโปรแกรมประยุกต์ในแต่ละฝั่งการทำงานจะใช้หมายเลขดังกล่าว ซึ่งมีขนาด 16 บิต ยกตัวอย่างเช่น โปรแกรมที่ทำหน้าที่ร้องขอหรือ โปรแกรมไคลเอนต์อาจส่งข้อมูลคำสั่งแกรมโดยระบุหมายเลขช่องทางการสื่อสาร (port) ที่ฝั่งเซิร์ฟเวอร์เป็นหมายเลข 1700 โปรแกรมที่ทำหน้าที่ให้บริการหรือ โปรแกรมเซิร์ฟเวอร์จะรอรับข้อมูลคำสั่งแกรมที่ไคลเอนต์ส่งมาซึ่งจะระบุถึงหมายเลขช่องทางการสื่อสารปลายทางเป็นหมายเลข 1700 และ เมื่อเซิร์ฟเวอร์ได้รับข้อมูลที่มีหมายเลขดังกล่าว เซิร์ฟเวอร์ก็สามารถที่จะส่งข้อมูลตอบสนองกลับไปไคลเอนต์ได้ ซึ่งไคลเอนต์ก็จะรอรับข้อมูลคำสั่งแกรมอยู่ที่ระบุช่องทางการสื่อสารปลายทางหมายเลข 1701

2.1.4 รูปแบบของไอพีแอดเดรส

เราทราบอยู่แล้วว่าไอพีแอดเดรสเป็นเลขฐานสองที่มีขนาด 32 บิต ซึ่งอาจคิดได้ว่าหมายเลขจำนวน 2^{32} เป็นหมายเลขที่ใช้ในการระบุที่อยู่ของคอมพิวเตอร์แต่ละเครื่องที่ไม่ซ้ำกันที่ทำงานอยู่บนอินเทอร์เน็ต แต่นั่นไม่ใช่สิ่งที่คุณต้องทั้งหมด เพราะส่วนหนึ่งของหมายเลขไอพีแอดเดรสเป็นการระบุถึงเครือข่ายซึ่งมีโฮสหรือเครื่องคอมพิวเตอร์ทำงานอยู่ และ ส่วนหนึ่งใช้ระบุถึงเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่านั้น หมายเลขไอพีแอดเดรสส่วนใหญ่เป็นคลาส C ที่มีรูปแบบดังแสดงในรูปที่ 2.5



รูปที่ 2.5 รูปแบบของไอพีแอดเดรสคลาส C

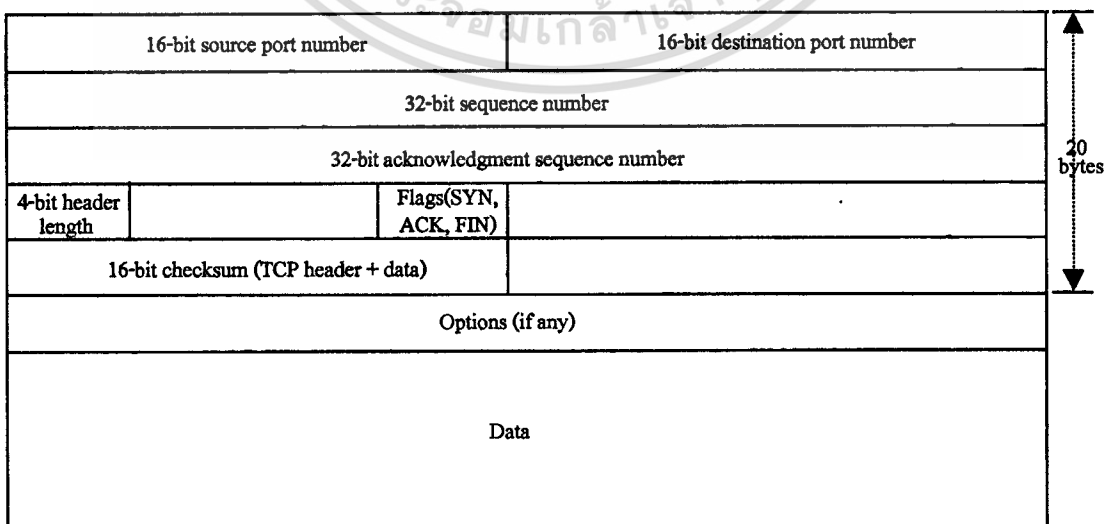
จากรูปแสดงให้เห็นว่าสามารถมีระบบเครือข่ายได้มากกว่า 2 ล้านเครือข่าย และ แต่ละเครือข่ายสามารถมีหมายเลขที่อยู่เครื่องได้ถึง 2^8 (256) เครื่อง

โดยทั่วไปแล้วไอพีแอดเดรสจะถูกเขียนเป็นรูปแบบที่เรียกว่า dotted-decimal โดยแบ่งออกเป็นสี่ส่วนแต่ละส่วนจะเป็นข้อมูลหนึ่งไบต์ ยกตัวอย่างเช่น ไอพีแอดเดรสคลาส C คือ 194.128.198.201 ในการทำงานของคอมพิวเตอร์ที่มีการใช้งานซีพียูของอินเทล ไบต์ข้อมูลที่อยู่ ถูกจัดเก็บเรียงจากบิตที่มีนัยสำคัญต่ำไปบิตที่มีนัยสำคัญสูง (low-order-to-the-left) หรือเรียกว่าการเรียงจากบิตที่น้อยไปบิตที่มาก (little-endian) ไม่อย่างนั้นถ้าเป็นบิตที่มีนัยสำคัญสูงไปบิตที่มีนัยสำคัญต่ำ (high-order-to-the-left) หรือเรียกว่าการเรียงจากบิตที่มากไปบิตที่น้อย (big-endian) อย่างไรก็ตามในปัจจุบันนี้มีการนำทั้งสองแบบมาใช้งานร่วมกันในหลายๆที่

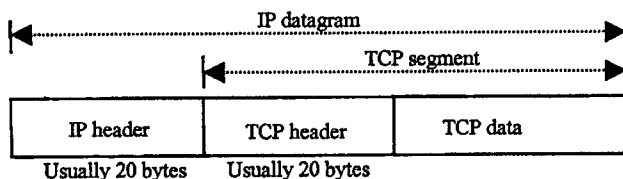
เรียงแบบ little-endian ในขณะที่คอมพิวเตอร์ส่วนใหญ่รวมถึงเครื่องที่คอมพิวเตอร์ที่ใช้ระบบ UNIX ซึ่งเป็นระบบแรกที่รองรับการทำงานของอินเทอร์เน็ต ไบต์ข้อมูลที่อยู่จะถูกจัดเก็บโดยเรียงจากบิตที่มีนัยสำคัญสูงไปบิตที่มีนัยสำคัญต่ำ (high-order-to-the-left) หรือเรียกว่าการเรียงแบบ big-endian เพราะว่าอินเทอร์เน็ตกำหนดให้มีมาตรฐานการแลกเปลี่ยนข้อมูลที่ไม่ขึ้นอยู่กับเครื่อง ไบต์ข้อมูลต้องถูกส่งที่มีการเรียงแบบของ big-endian ซึ่งหมายความว่าโปรแกรมที่ทำงานในเครื่องที่ใช้งานซีพียูของอินเทลต้องทำการแปลงข้อมูลระหว่างการเรียงไบต์ข้อมูลในระบบเครือข่าย (big-endian) กับการเรียงข้อมูลของเครื่องคอมพิวเตอร์ (little-endian) กฎนี้ใช้ทั้งในหมายเลขช่องทางการสื่อสาร (port number) ที่มีขนาด 2 ไบต์ และ ในไอพีแอดเดรสที่มีขนาด 4 ไบต์ด้วย

2.1.5 Transmission Control Protocol

เราทราบเกี่ยวกับข้อจำกัดของโพรโตคอล UDP สิ่งที่เราต้องการคือ โพรโตคอลที่รองรับการส่งข้อมูลของกลุ่มข้อมูลขนาดใหญ่โดยไม่เกิดข้อผิดพลาด เห็นได้ชัดว่าเราต้องการให้โปรแกรมที่รับข้อมูลสามารถที่จะรวมข้อมูลไบต์ให้ได้ลำดับการทำงานที่ถูกต้องตรงตามที่ได้ส่งมาถึงแม้ว่าแต่ละคำคำแกรมอาจมาถึงในลำดับที่ผิดไปก็ตาม โพรโตคอล TCP คือ โพรโตคอลที่เรากำลังพูดถึงและ โพรโตคอล TCP เป็นโพรโตคอลที่ใช้ในการส่งข้อมูลที่มีความสำคัญในการใช้งานอินเทอร์เน็ตทั้งหมด ซึ่งรวมถึงการใช้งานโพรโตคอล HTTP และ โพรโตคอลที่ใช้ในการส่งไฟล์ข้อมูล (File Transfer Protocol (FTP)) รูปที่ 2.6 แสดงให้เห็นถึงรูปแบบของ TCP segment (เราจะไม่เรียกว่าคำแกรม) ซึ่ง TCP segment จะบรรจุอยู่ใน IP datagram อีกทีหนึ่ง ดังแสดงในรูปที่ 2.7



รูปที่ 2.6 รูปแบบของ TCP segment



รูปที่ 2.7 ความสัมพันธ์ระหว่าง IP datagram และ TCP segment

โพรโตคอล TCP จะสร้างการเชื่อมต่อแบบ full-duplex, การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์สองเครื่องแบบ point-to-point และ โปรแกรมในเครื่องแต่ละฝั่งของการเชื่อมต่อจะมีการใช้งานช่องทางการสื่อสาร (port) ที่เป็นของตัวเอง การใช้งานร่วมกันระหว่าง IP address และ หมายเลข port จะเรียกว่า socket การเชื่อมต่อในครั้งแรกจะใช้เทคนิค three-way handshake โดย โปรแกรมที่เริ่มทำการเชื่อมต่อจะส่งเซ็กเมนต์ข้อมูลที่มีแฟล็ก SYN ไป แล้วโปรแกรมตอบสนองจะส่งเซ็กเมนต์ข้อมูลที่มีแฟล็ก SYN และ ACK กลับมาแล้ว โปรแกรมเริ่มการเชื่อมต่อก็จะส่งเซ็กเมนต์ข้อมูลที่มีแฟล็ก ACK อีกครั้ง

หลังจากการเชื่อมต่อถูกสร้างแล้ว แต่ละโปรแกรมสามารถจะส่งข้อมูลเรียงกันเป็นไบต์ต่อไบต์เป็นสาย (stream) ไปยังโปรแกรมที่อยู่อีกด้านหนึ่งของการเชื่อมต่อได้ โพรโตคอล TCP ใช้หมายเลขเรียงลำดับ (sequence number) กับแฟล็ก ACK ในการควบคุมการส่งข้อมูล โปรแกรมส่งข้อมูลไม่ต้องรอให้ข้อมูลแต่ละเซ็กเมนต์ให้ส่งสัญญาณตอบกลับ (acknowledgement) ว่าได้ถึงปลายทางแล้วแต่จะส่งเซ็กเมนต์ข้อมูลเป็นจำนวนมาก ๆ ก่อนแล้วทำการส่งสัญญาณตอบกลับครั้งเดียว ถ้าโปรแกรมรับข้อมูลต้องการส่งข้อมูลกลับไปให้โปรแกรมส่งข้อมูล โปรแกรมรับข้อมูลสามารถที่จะบรรจุสัญญาณตอบกลับและข้อมูลส่งออกไปที่โปรแกรมส่งข้อมูลภายในเซ็กเมนต์เดิมที่ส่งมาได้

2.2 การส่งข้อมูลในระบบเครือข่ายด้วยซ็อกเก็ต

การติดต่อสื่อสารผ่านระบบเครือข่ายระหว่างไคลเอนต์และเซิร์ฟเวอร์จะใช้สิ่งที่เรียกว่าซ็อกเก็ต (socket) เป็นตัวเชื่อมโยงให้ทั้งสองฝั่งรับส่งข้อมูลกันได้ โดยทำงานผ่านโพรโตคอล TCP/IP โพรโตคอล IP จะรองรับการทำงานส่วนที่ช่วยให้ข้อมูลจากเครื่องคอมพิวเตอร์เครื่องหนึ่งส่งไปยังอีกเครื่องหนึ่งในเครือข่ายได้ ส่วนโพรโตคอล TCP ช่วยสร้างความน่าเชื่อถือว่าข้อมูลสามารถส่งไปถึงผู้รับได้อย่างแน่นอน เพราะจะต้องมีการรับทราบด้วยกันทั้งสองฝ่ายก่อนรับส่งข้อมูลเสมอ ความสามารถของโพรโตคอล TCP/IP นี้ทำให้ความเร็วในการรับส่งข้อมูลลดลง เพราะต้องมีการตรวจสอบกันหลายขั้นตอนทั้งผู้ส่งและผู้รับ

ส่วนโพรโตคอล UDP ซึ่งทำงานรับส่งข้อมูลได้เหมือนกับโพรโตคอล TCP โดยตัดขั้นตอนการตรวจสอบบางอย่างระหว่างการรับส่งข้อมูลออกไป โพรโตคอลนี้จึงส่งข้อมูลได้รวดเร็วกว่าโพรโตคอล TCP แต่ขณะเดียวกันก็มีความน่าเชื่อถือในการทำงานน้อยกว่า เพราะไม่มีหลักประกันได้ว่า ข้อมูลที่ส่งไปแล้วนั้นถึงผู้รับจริงหรือไม่ เนื่องจากโพรโตคอล UDP ทำงานโดยไม่มีการตรวจสอบความพร้อมของผู้รับ

ไม่ว่าจะใช้โพรโตคอล TCP หรือ UDP ในการรับส่งข้อมูลก็ตาม ล้วนจำเป็นต้องใช้ช่องทางการสื่อสาร (port) สำหรับการติดต่อสื่อสาร พอร์ตเปรียบเสมือนช่องทางสำหรับรับส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ port จะใช้ตัวเลขขนาด 16 บิตกำกับไว้ ซึ่งหมายความว่าในแต่ละเครื่องจะมีหมายเลขพอร์ตให้ใช้งานได้ถึง 65536 port เริ่มจากพอร์ตหมายเลข 0 ถึง 65535 ในระดับการทำงานจริง ๆ แล้วพอร์ตเป็นเพียงตัวเลขที่ส่งไปพร้อมกับข้อมูล เพื่อให้ทราบได้ว่าข้อมูลนั้นทำงานกับพอร์ตหมายเลขอะไรในเซิร์ฟเวอร์ปลายทาง หมายเลขพอร์ตจึงเป็นหมายเลขที่ใช้แยกข้อมูลไม่ให้ปนกัน

ในระบบ UNIX มีการเก็บข้อมูลหมายเลขพอร์ตของแต่ละพอร์ตถูกใช้ทำงานอะไรไปแล้วบ้าง ข้อมูลนี้เก็บไว้ในไฟล์ /etc/services พอร์ตหมายเลข 1 ถึง 255 มีการสงวนไว้สำหรับบริการมาตรฐานที่ยอมรับเป็นสากล เช่น พอร์ตหมายเลข 80 สำหรับบริการ WWW, พอร์ตหมายเลข 25 สำหรับการส่งอีเมล เป็นต้น ในการเขียนโปรแกรมในการใช้งานซ็อกเก็ตซึ่งต้องมีการใช้งานพอร์ตนั้นจึงควรกำหนดหมายเลขพอร์ตที่ว่างอยู่ ตั้งแต่หมายเลข 1024 เพราะหมายเลขพอร์ตช่วงหมายเลข 1 ถึง 1023 นั้น เซิร์ฟเวอร์สงวนไว้สำหรับโปรแกรมและการทำงานของเซิร์ฟเวอร์เอง

2.2.1 แนวคิดการเขียนโปรแกรมเครือข่าย

หลักการพื้นฐานการรับส่งข้อมูลผ่านเครือข่ายจะใช้สิ่งที่เรียกว่าซ็อกเก็ต (socket) ซึ่งทั้งผู้รับและผู้ส่งจะต้องสร้างซ็อกเก็ตขึ้นมา ฝ่ายที่ต้องการข้อมูลจะต้องสั่งให้ซ็อกเก็ตของตนไปเชื่อมต่อกับซ็อกเก็ตของผู้รับ เมื่อซ็อกเก็ตของผู้รับยอมให้มีการเชื่อมต่อได้ ทั้งสองฝ่ายจึงจะเริ่มส่งข้อมูลระหว่างกันได้โดยทำงานผ่านซ็อกเก็ต การเขียน โปรแกรมใช้งานซ็อกเก็ตแบ่งเป็น 2 แบบคือ

- เขียน โปรแกรมทำหน้าที่เป็นเซิร์ฟเวอร์ซึ่งจะต้องสั่งให้ซ็อกเก็ตที่เขียนขึ้นอยู่ในสถานะรอการร้องขอติดต่อกับซ็อกเก็ตอื่น ซ็อกเก็ตที่ร้องขอเข้ามาอาจจะอยู่ในเครื่องเดียวกัน, เครื่องอื่น ๆ ในเครือข่ายเดียวกันหรือมาจากเครือข่ายอินเทอร์เน็ตก็ได้
- เขียน โปรแกรมทำหน้าที่เป็นไคลเอนต์ ซึ่งต้องสั่งให้ซ็อกเก็ตที่สร้างขึ้นทำการเชื่อมต่อไปยังซ็อกเก็ตในเซิร์ฟเวอร์เป้าหมายที่ต้องการ ซ็อกเก็ตเป้าหมายจะอยู่ในเครื่องเดียวกันหรือต่างเครื่องก็ได้ เมื่อซ็อกเก็ตในเซิร์ฟเวอร์ยอมรับการเชื่อมต่อแล้วจึงจะสามารถสั่งให้ซ็อกเก็ตร้องขอข้อมูลจากเซิร์ฟเวอร์ได้

2.2.2 ขั้นตอนการทำงานของซ็อกเก็ต

2.2.2.1 สร้างซ็อกเก็ต

ขั้นแรกของการสร้างซ็อกเก็ตจะต้องเรียกฟังก์ชันการสร้างซ็อกเก็ตคือ `socket()` เพื่อสร้างซ็อกเก็ตขึ้นมา ซึ่งซ็อกเก็ตที่สร้างแบ่งได้เป็นสองประเภทคือ

- **AF_UNIX** ถ้าสร้างซ็อกเก็ตโดยระบุเป็น **AF_UNIX** ในการเรียกฟังก์ชัน `socket()` เพื่อสร้างซ็อกเก็ต ซ็อกเก็ตที่สร้างขึ้นจะใช้ในการติดต่อกับซ็อกเก็ตอื่นภายในเครื่องเดียวกันเท่านั้น เพราะการอ้างถึงตำแหน่งซ็อกเก็ตจะใช้ชื่อ ไคลเอนต์เป็นตัวบอก
- **AF_INET** เป็นการสร้างซ็อกเก็ตเพื่อใช้ติดต่อกับซ็อกเก็ตอื่นที่อยู่ต่างเครื่อง, ต่างเซิร์ฟเวอร์ หรือผ่านเครือข่ายอินเทอร์เน็ต ซ็อกเก็ตชนิดนี้อ้างอิงตำแหน่งถึงกันด้วย IP แอดเดรส และ หมายเลขพอร์ต

ในการสร้างซ็อกเก็ตสิ่งที่จะต้องกำหนดให้กับฟังก์ชัน `socket()` ในการสร้างอีกอย่างคือการกำหนดรูปแบบการรับส่งข้อมูลผ่านซ็อกเก็ตซึ่งแบ่งได้สองแบบคือ

- **SOCKET_STREAM** ใช้กับโพรโทคอล TCP การรับส่งข้อมูลผ่านเครือข่ายด้วยซ็อกเก็ตแบบนี้มีความน่าเชื่อถือ เพราะก่อนรับส่งข้อมูลซ็อกเก็ตจะต้องตรวจสอบก่อนว่าทั้งสองฝ่ายพร้อมจึงจะเริ่มรับส่งข้อมูล การสื่อสารผ่านโพรโทคอล TCP จะต้องมีการ

เชื่อมต่อระหว่างผู้รับและผู้ส่งก่อนเสมอ จึงเรียกการทำงานของโพรโทคอลนี้ว่าเป็นแบบ connection-oriented

- **SOCK_DGRAM** ใช้กับโพรโทคอล UDP ซึ่งออกแบบนี้ส่งข้อมูลได้ไวกว่าโพรโทคอล TCP เพราะไม่มีการตรวจสอบความพร้อมของผู้รับ และ ไม่จำเป็นต้องสร้างการเชื่อมต่อระหว่างผู้รับกับผู้ส่ง จึงเรียกการทำงานของโพรโทคอล UDP ว่าเป็นแบบ connectionless การส่งข้อมูลผ่านโพรโทคอล UDP จึงไม่สามารถรับประกันได้ว่าข้อมูลที่ส่งแต่ละครั้งถึงผู้รับอย่างแน่นอน

การติดต่อสื่อสารจะต้องใช้ซ็อกเก็ตทั้งฝ่ายผู้ส่งและผู้รับ ผู้ส่งต้องสร้างซ็อกเก็ตขึ้นมาและส่งข้อมูลไปยังผู้รับผ่านซ็อกเก็ต ส่วนผู้รับต้องสร้างซ็อกเก็ตขึ้นมาเพื่อรับข้อมูล ในเครื่องหนึ่ง ๆ สามารถมีซ็อกเก็ตได้หลายตัวและทำงานพร้อมกันได้ เพราะซ็อกเก็ตจะมีพอร์ตทำงานแยกจากกันโดยอิสระ การแยกความแตกต่างว่าซ็อกเก็ตใดทำงานอะไรหรือรอข้อมูลจากซ็อกเก็ตไหนนั้น ซ็อกเก็ตแต่ละตัวจะมีหมายเลขประจำตัวกำกับไว้ ซึ่งได้จาก IP แอดเดรสของเครื่องที่ซ็อกเก็ตนั้นทำงานอยู่และหมายเลขพอร์ตที่ซ็อกเก็ตนั้นใช้งาน ซ็อกเก็ตจึงทำงานแยกจากกันอย่างอิสระไม่รบกวนซึ่งกันและกัน

socket (name, family, type, protocol)

name หมายถึงชื่อของซ็อกเก็ต
 family หมายถึงประเภทของซ็อกเก็ต (AF_INET หรือ AF_UNIX)
 type หมายถึงรูปการรับส่ง (SOCK_STREAM หรือ SOCK_DGRAM)
 protocol หมายถึงหมายเลขโพรโทคอลที่จะใช้กับซ็อกเก็ต

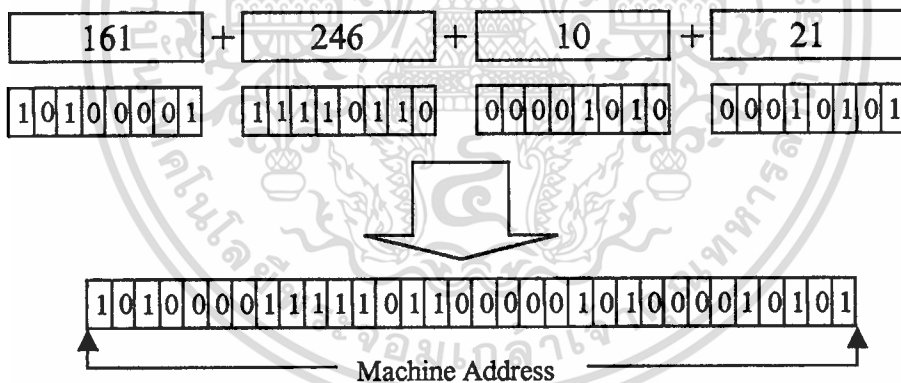
ในการเขียน โปรแกรมเพื่อทำงานเป็นเซิร์ฟเวอร์รอการร้องขอข้อมูลจากไคลเอนต์ โปรแกรมจะต้องสร้างซ็อกเก็ตขึ้นก่อน ซึ่งจะได้ซ็อกเก็ตเตรียมพร้อมสำหรับการติดต่อ

2.2.2.2 แอดเดรสของซ็อกเก็ต (Socket Address)

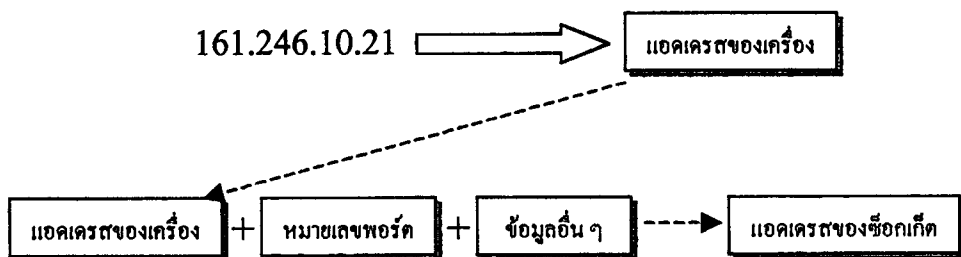
เนื่องจากซ็อกเก็ตต้องทำงานติดต่อกับซ็อกเก็ตด้วยกัน การที่เราจะบอกได้ว่าซ็อกเก็ตแต่ละตัวมาจากเครื่องไหนและทำงานกับพอร์ตไหนนั้น ซ็อกเก็ตแต่ละตัวจะต้องมีแอดเดรสประจำตัวเพื่อแจ้งแก่ซ็อกเก็ตที่กำลังคิดต่อซึ่งใช้กำกับในการรับส่งข้อมูล ทำให้ข้อมูลไปถึงซ็อกเก็ตที่ต้องการได้อย่างถูกต้อง

IP แอดเดรสที่มีขนาด 32 บิต ซึ่งเป็นตัวอ้างอิงตำแหน่งของแต่ละเครื่องในอินเทอร์เน็ต การเขียนเลข IP แอดเดรสตัวเลข 4 ตัวคั่นด้วยจุด เป็นเพียงการใช้งานเพื่อสื่อความหมายของเราเท่านั้น สำหรับในการทำงานของซ็อกเก็ต ในการอ้างถึง IP แอดเดรสใด ๆ ต้องแปลงเลข IP ทั้งสี่เป็นแอดเดรสของเครื่อง (Machine Address) ในการอ้างถึงตำแหน่งของซ็อกเก็ตเองหรือของซ็อกเก็ตตัวอื่นจะใช้แอดเดรสของซ็อกเก็ต

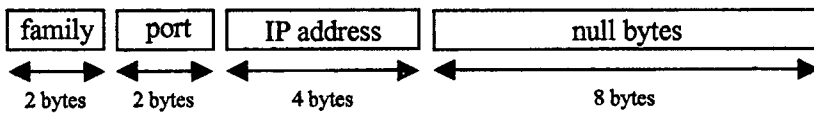
แอดเดรสของเครื่องเกิดจากการนำเอา IP แอดเดรสมาแปลงรูป ส่วนแอดเดรสของซ็อกเก็ตเกิดจากการเอาแอดเดรสของเครื่อง, หมายเลขพอร์ตที่ซ็อกเก็ตใช้ และข้อมูลอื่นอีกบางส่วน มารวมกันเป็นแอดเดรสของซ็อกเก็ตดังรูปที่ 2.9



รูปที่ 2.8 การแปลง IP แอดเดรส 161.246.10.21 เป็นหมายเลขเครื่อง



รูปที่ 2.9 การหาแอดเดรสของเครื่องและแอดเดรสของซ็อกเก็ต



รูปที่ 2.10 โครงสร้างแอดเดรสของซ็อกเก็ต

2.2.2.3 การเชื่อมต่อซ็อกเก็ต

หลังจากที่สร้างซ็อกเก็ตเพื่อใช้ในการเชื่อมต่อจะต้องกำหนดแอดเดรสให้กับซ็อกเก็ตที่สร้างขึ้น ขั้นตอนนี้จะใช้ฟังก์ชัน `bind()` ซึ่งหมายถึงการกำหนดเพื่อให้ซ็อกเก็ตที่ได้สร้างไว้มีแอดเดรสในการติดต่อสื่อสารกับซ็อกเก็ตอื่นได้

`bind(Socket, SocketAddress)`

Socket หมายถึงชื่อของซ็อกเก็ตที่สร้างไว้

SocketAddress หมายถึงแอดเดรสของซ็อกเก็ต

2.2.2.4 การรอการร้องขอจากซ็อกเก็ตไคลเอนต์ในกรณีของซ็อกเก็ตเซิร์ฟเวอร์

ในการเขียนโปรแกรมเพื่อทำงานเป็นเซิร์ฟเวอร์ ซ็อกเก็ตของเราจะต้องรอการร้องขอการเชื่อมต่อจากไคลเอนต์ การทำงานนี้ใช้คำสั่ง `listen()` ในขณะที่สร้างซ็อกเก็ต हमเราเลือกรูปแบบเป็น `SOCK_STREAM` แล้ว ซ็อกเก็ตจะมีความสามารถในการจัดคิวการร้องขอข้อมูล สำหรับการร้องขอเข้ามาที่ซ็อกเก็ตของเซิร์ฟเวอร์เข้ามาหลาย ๆ รายการ รายการใดที่ซ็อกเก็ตยังไม่สามารถรับไปทำงานได้ จะอยู่ในคิวรอการทำงานต่อไป

`listen(Socket, Buffer)`

Socket หมายถึงชื่อของซ็อกเก็ตที่สร้างไว้

Buffer หมายถึงจำนวนซ็อกเก็ตสูงสุดที่จะให้อยู่ในคิว

2.2.2.5 การร้องขอไปที่ซ็อกเก็ตเซิร์ฟเวอร์ในกรณีของซ็อกเก็ตโคลเอนต์

ฝั่งโคลเอนต์ต้องสร้างซ็อกเก็ตขึ้นก่อนแล้วกำหนดแอดเดรสให้กับซ็อกเก็ตที่สร้างขึ้น จากนั้นต้องกำหนดแอดเดรสของซ็อกเก็ตปลายทางที่จะเชื่อมต่อให้ซ็อกเก็ต แล้วจึงสามารถที่จะทำการเชื่อมต่อซ็อกเก็ตจากเครื่องโคลเอนต์ไปยังซ็อกเก็ตในเซิร์ฟเวอร์ได้ โดยการใช้ฟังก์ชัน `connect()` ที่ต้องกำหนดให้กับฟังก์ชัน `connect()` คือ ชื่อของซ็อกเก็ตในเครื่องโคลเอนต์และแอดเดรสของซ็อกเก็ตปลายทาง

```
connect(Socket, TargetSocketAddress)
```

Socket หมายถึงชื่อของซ็อกเก็ตที่สร้างไว้
TargetSocketAddress หมายถึงแอดเดรสของซ็อกเก็ตปลายทาง

2.2.2.6 การตอบรับการเชื่อมต่อ

เมื่อซ็อกเก็ตเซิร์ฟเวอร์ปลายทาง ได้รับสัญญาณขอเชื่อมต่อและยอมรับการเชื่อมต่อด้วยคำสั่ง `accept()` แล้ว ทั้งโคลเอนต์และเซิร์ฟเวอร์จึงเริ่มการรับส่งข้อมูลระหว่างกันได้

```
client_socket = accept(NewSocket, Socket)
```

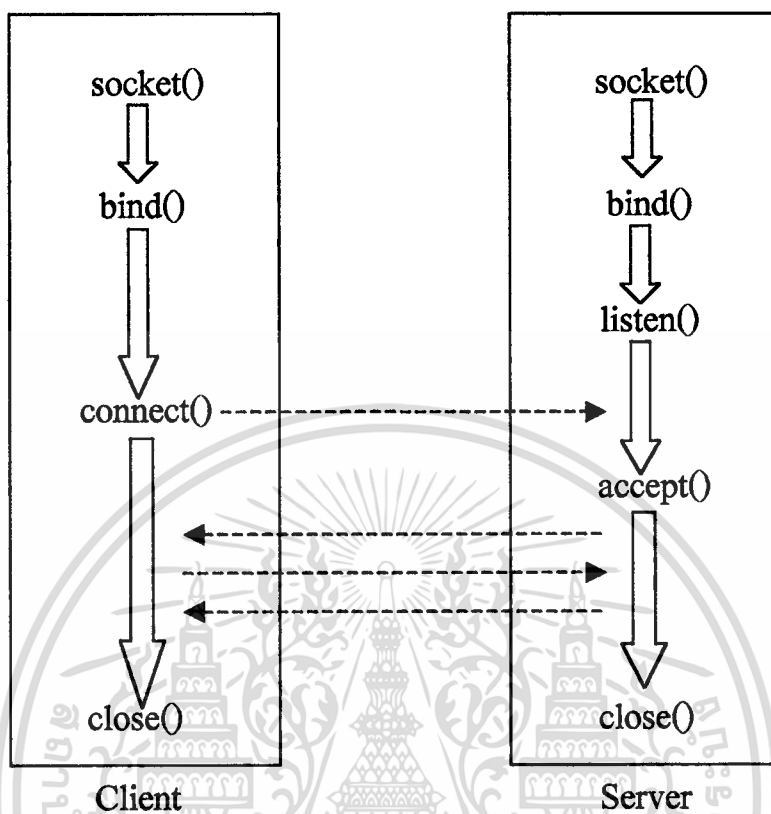
client_socket หมายถึงตัวแปรซึ่งมีค่าเป็นแอดเดรสของโคลเอนต์ที่เซิร์ฟเวอร์ตอบรับ (มีค่าเป็นแอดเดรสของซ็อกเก็ตตัวใหม่ NewSocket)
NewSocket หมายถึงซ็อกเก็ตตัวใหม่ที่ถูกสร้างขึ้นเพื่อติดต่อกับโคลเอนต์
Socket หมายถึงชื่อของซ็อกเก็ตที่สร้างไว้

2.2.2.7 การยกเลิกการเชื่อมต่อ

การรับส่งข้อมูลระหว่างโคลเอนต์กับเซิร์ฟเวอร์ จะกระทำผ่านซ็อกเก็ตได้ตลอดเวลา จนกว่าฝ่ายใดฝ่ายหนึ่งทำการตัดการเชื่อมต่อ ด้วยคำสั่ง `close()` การเชื่อมต่อระหว่างสองฝ่ายจึงจบลง

```
close(Socket)
```

Socket หมายถึงแอดเดรสของซ็อกเก็ตที่ต้องการตัดการเชื่อมต่อ



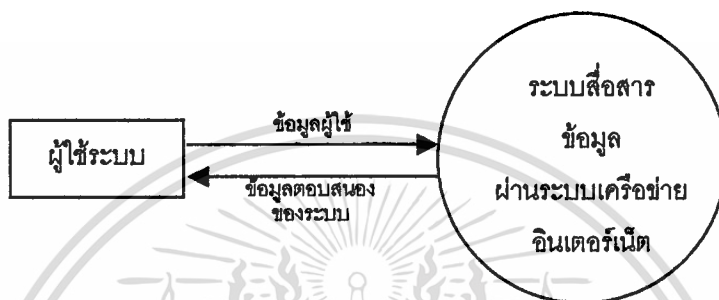
รูปที่ 2.11 ขั้นตอนการทำงานของซ็อกเก็ต โคลเฮนค์และซ็อกเก็ตเซิร์ฟเวอร์

บทที่ 3

การพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต

3.1 การวิเคราะห์และการออกแบบ

3.1.1 Context Diagram ของระบบ



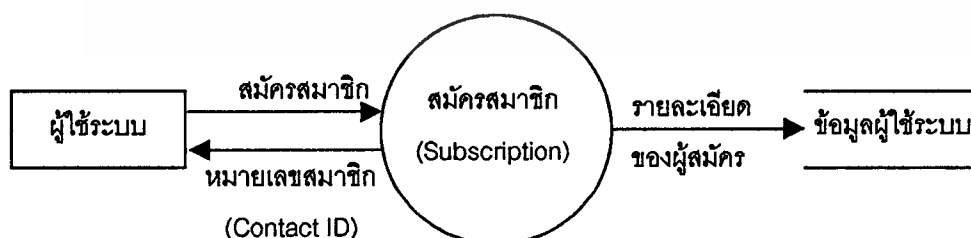
รูปที่ 3.1 Context Diagram ของระบบ

3.1.2 Data Flow Diagram ของระบบ

ระบบสามารถแยกเป็นส่วนต่างได้ตามลักษณะการทำงานได้ดังนี้

3.1.2.1 การสมัครสมาชิกเพื่อเข้าใช้งานระบบ

ในการสมัครสมาชิกของผู้ที่ต้องการใช้งานระบบสื่อสารผ่านระบบเครือข่ายจะสามารถทำได้โดยการเข้าไปยังเว็บไซต์แล้วทำการเลือกหัวข้อการสมัครสมาชิก (Subscription) แล้วทำการกรอกรายละเอียดเกี่ยวกับตัวผู้ใช้หลังจากนั้นระบบจะแจ้งให้ทราบถึงหมายเลขที่ใช้ในการเข้าระบบ (Contact ID) เพื่อใช้ในการเข้าระบบต่อไป



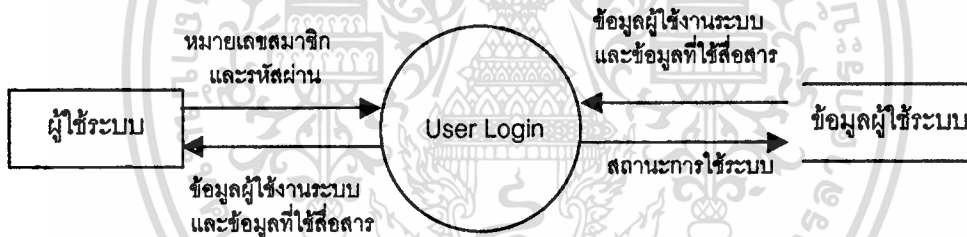
รูปที่ 3.2 DFD แสดงการสมัครสมาชิก

ขั้นตอนการทำงานของ Process สมัครงาน

1. รับข้อมูลจากผู้สมัคร
2. ตรวจสอบข้อมูลถ้าข้อมูลที่ผู้สมัครถูกต้องทำข้อ 3 ถ้าข้อมูลผิดพลาดเช่นกรอกข้อมูลไม่ครบ แจ้งให้ผู้สมัครทราบพร้อมแสดงหน้าจอเพื่อให้ผู้สมัครกรอกข้อมูลใหม่
3. จัดเก็บข้อมูลลงฐานข้อมูล
4. แจ้งหมายเลขประจำให้ผู้สมัครทราบเพื่อใช้ในการเข้าระบบต่อไป

3.1.2.2 การเข้าใช้งานระบบของสมาชิก

การเข้าใช้งานระบบ (User login) สามารถทำได้โดยการเข้าเว็บไซต์แล้วเลือกหัวข้อ User Login แล้วให้ผู้ใช้อกรอกหมายเลขสมาชิกที่ผู้ใช้ได้สมัครไว้แล้ว และรหัสผ่าน เพื่อใช้ในการตรวจสอบสิทธิการใช้งานระบบ



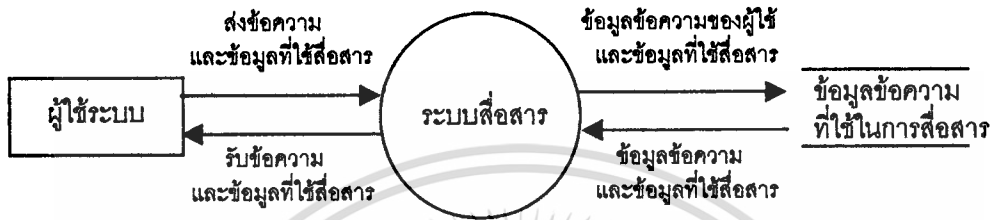
รูปที่ 3.3 แสดง DFD การเข้าใช้งานระบบ

ขั้นตอนการทำงานของ Process User Login

1. รับข้อมูลหมายเลขประจำตัว และ รหัสผ่านของผู้ใช้ระบบ
2. ทำการตรวจสอบหมายเลขประจำตัว และ รหัสผ่านของผู้ใช้ระบบ จากฐานข้อมูลผู้ใช้ระบบ ถ้าผิดพลาด เช่น ไม่มีหมายเลขประจำตัวดังกล่าว หรือ รหัสผ่านผิดพลาด แจ้งความผิดพลาดให้ผู้ใช้งานทราบ แล้วแสดงหน้าจอเพื่อให้ผู้ใช้ทำการกรอกหมายเลขประจำตัวและรหัสผ่านอีกครั้ง ถ้าหมายเลขและรหัสผ่านของผู้ใช้ถูกต้องทำข้อ 3
3. แสดงหน้าจอที่ใช้ในการติดต่อสื่อสาร เข้าสู่ Process การติดต่อกับเซิร์ฟเวอร์เพื่อสื่อสารข้อมูล

3.1.2.3 การส่งข้อความระหว่างผู้ใช้

ภายหลังจากที่ผู้ใช้ได้รับการตรวจสอบว่าเป็นผู้ที่ได้สมัครเป็นสมาชิกของระบบแล้ว ผู้ใช้จะเข้าสู่การทำงานของระบบสื่อสาร โดยมีการทำงานดังนี้



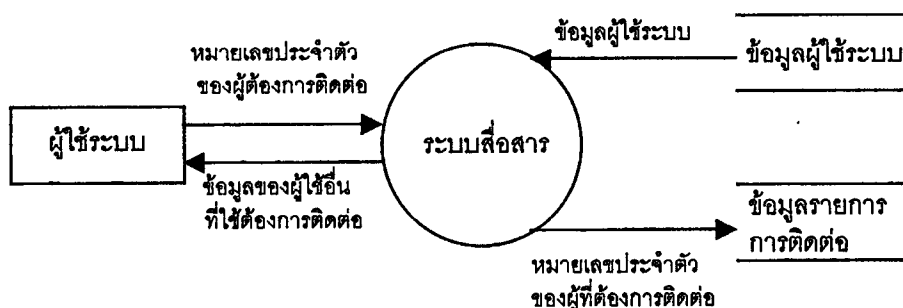
รูปที่ 3.4 แสดง DFD การใช้งานระบบสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต

ขั้นตอนการทำงานของ Process ระบบสื่อสารในการส่งข้อความระหว่างผู้ใช้ระบบ

1. รับข้อความที่รับจากผู้ใช้โดยตรงหรือข้อมูลที่ใช้ในการสื่อสาร (ระบบสื่อสารฝั่งไคลเอนต์ส่งข้อมูลเพื่อใช้ในการควบคุมการทำงานของระบบ)
2. ทำการแปลความหมายข้อมูลที่ได้รับจากการทำงานฝั่งไคลเอนต์ แล้วจัดเก็บข้อมูลลงฐานข้อมูล
3. อ่านข้อมูลที่เกี่ยวข้องกับผู้ใช้จากฐานข้อมูล
4. ส่งข้อมูลให้กับระบบฝั่งไคลเอนต์

3.1.2.4 การเพิ่มรายชื่อผู้ใช้อื่นที่ต้องการติดต่อในรายการติดต่อของผู้ใช้

ภายหลังจากที่ผู้ใช้ได้ทำการ login ใช้งานระบบได้แล้ว ผู้ใช้สามารถที่จะทำการเพิ่มรายชื่อของบุคคลอื่นที่เป็นสมาชิกเข้าไปภายในรายชื่อของผู้ที่ต้องการติดต่อสื่อสาร โดยการเลือกคำสั่งเพิ่มผู้ที่ต้องการติดต่อ (Add) โดยการกำหนดหมายเลขที่ต้องการติดต่อ



รูปที่ 3.5 แสดงการเพิ่มรายชื่อผู้ใช้อื่นที่ต้องการติดต่อในรายการติดต่อของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของ Process ระบบสื่อสารในกรณีผู้ใช้เพิ่มรายชื่อผู้ใช้อื่นที่ต้องการติดต่อในรายการติดต่อของผู้ใช้

1. รับข้อมูลหมายเลขประจำตัวของผู้ใช้อื่นที่ผู้ใช้ต้องการเพิ่มในรายการติดต่อ
2. ตรวจสอบจากฐานข้อมูลผู้ใช้ระบบว่ามีหมายเลขดังกล่าวหรือไม่ ถ้าไม่มีหมายเลขดังกล่าว ทำการแจ้งข้อผิดพลาดแก่ผู้ใช้ระบบ พร้อมให้ผู้ใช้ระบบกรอกข้อมูลใหม่ ถ้ามีหมายเลขผู้ใช้อย่างกล่าว ทำข้อ 3
3. เพิ่มข้อมูลรายการผู้ใช้ให้กับข้อมูลส่วนผู้ใช้และผู้ใช้ที่ถูกเพิ่ม
4. ส่งข้อมูลส่วนตัวให้ผู้ใช้ที่ถูกเพิ่ม เช่นหมายเลขประจำตัวผู้ใช้, ชื่อที่ใช้ในการสื่อสาร และสถานะการใช้งานระบบของผู้ใช้รายนั้น (online/offline) กลับไปให้ผู้ใช้ระบบที่ทำการเพิ่ม

3.1.2.5 การลบรายชื่อผู้ใช้อื่นนอกจากรายการติดต่อของผู้ใช้

ผู้ใช้สามารถทำการลบรายชื่อผู้ใช้รายอื่นที่ไม่ต้องการทำการติดต่อสื่อสารด้วยอีกต่อไป ออกไปจากรายการติดต่อผู้ใช้ได้โดยการเลือกที่รายชื่อของผู้ที่ต้องการลบจากนั้นเลือกที่คำสั่งลบ (Remove)



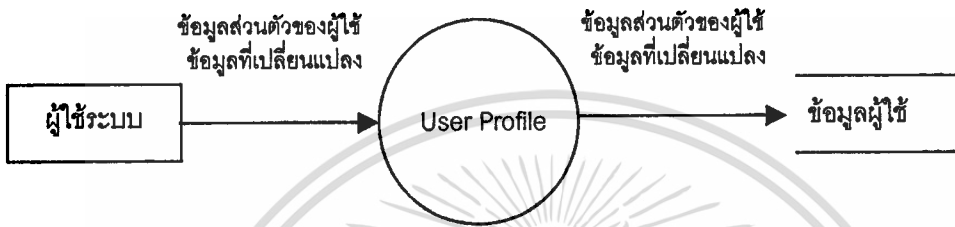
รูปที่ 3.6 แสดงการลบรายชื่อผู้ใช้อื่นนอกจากรายการติดต่อของผู้ใช้

ขั้นตอนการทำงานของ Process ระบบสื่อสารในกรณีผู้ใช้ลบรายชื่อผู้ใช้อื่นที่ต้องการติดต่อออกจากรายการติดต่อของผู้ใช้

1. รับข้อมูลหมายเลขประจำตัวผู้ใช้ระบบที่ผู้ต้องการลบออกจากรายการติดต่อของผู้ใช้
2. ทำการลบข้อมูลออกจากรายการติดต่อของผู้ใช้
3. แจ้งให้ผู้ใช้ทราบการลบข้อมูล

3.1.2.6 การแก้ไขข้อมูลส่วนตัวของผู้ใช้

ผู้ใช้สามารถที่จะแก้ไขข้อมูลส่วนตัวได้โดยการเลือกที่คำสั่ง User Profile ซึ่งผู้ใช้ต้องกรอกหมายเลขประจำตัวและรหัสผ่านก่อนการเข้าไปแก้ไขข้อมูล เพื่อเป็นการยืนยันตัวผู้ใช้งานว่าเป็นผู้ใช้งานระบบตัวจริง



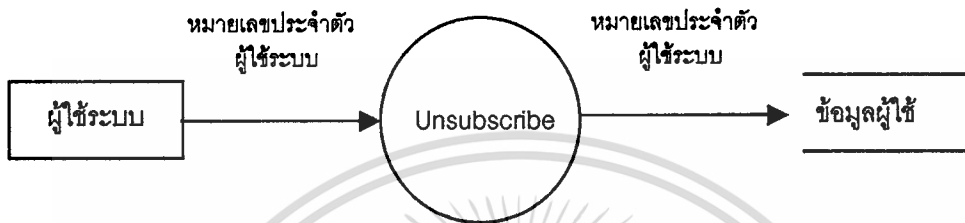
รูปที่ 3.7 แสดงการแก้ไขข้อมูลส่วนตัวของผู้ใช้

ขั้นตอนการทำงานของ Process User Profile

1. รับข้อมูลหมายเลขประจำตัวและรหัสผ่านของผู้ใช้ระบบ
2. ตรวจสอบหมายเลขประจำตัวและรหัสผ่านของผู้ใช้ระบบจากฐานข้อมูลผู้ใช้ ถ้าผิดพลาดแจ้งให้ผู้ใช้ทราบเช่น หมายเลขประจำตัวหรือรหัสผ่านผิดพร้อมแสดงหน้าจอให้ผู้ใช้ระบบกรอกหมายเลขประจำตัวและรหัสผ่านใหม่อีกครั้ง ถ้าหมายเลขประจำตัวและรหัสผ่านของผู้ใช้ถูกต้องทำข้อ 3
3. อ่านข้อมูลผู้ใช้ระบบจากฐานข้อมูลผู้ใช้
4. แสดงข้อมูลส่วนตัวของผู้ใช้ระบบ
5. รับข้อมูล que ผู้ใช้ทำการเปลี่ยนแปลง
6. ตรวจสอบข้อมูลกับฐานข้อมูลผู้ใช้ระบบ
7. เปลี่ยนแปลงข้อมูล que ผู้ใช้ทำการแก้ไขในฐานข้อมูลของผู้ใช้ระบบ

3.1.2.7 การยกเลิกการเป็นสมาชิก

ผู้ใช้สามารถทำการยกเลิกการเป็นสมาชิกได้โดยเลือกที่คำสั่ง User Unsubscription ซึ่งผู้ใช้ต้องกรอกหมายเลขประจำตัวของผู้ใช้และรหัสผ่านเพื่อยืนยันว่าเป็นผู้ใช้งานระบบที่แท้จริง

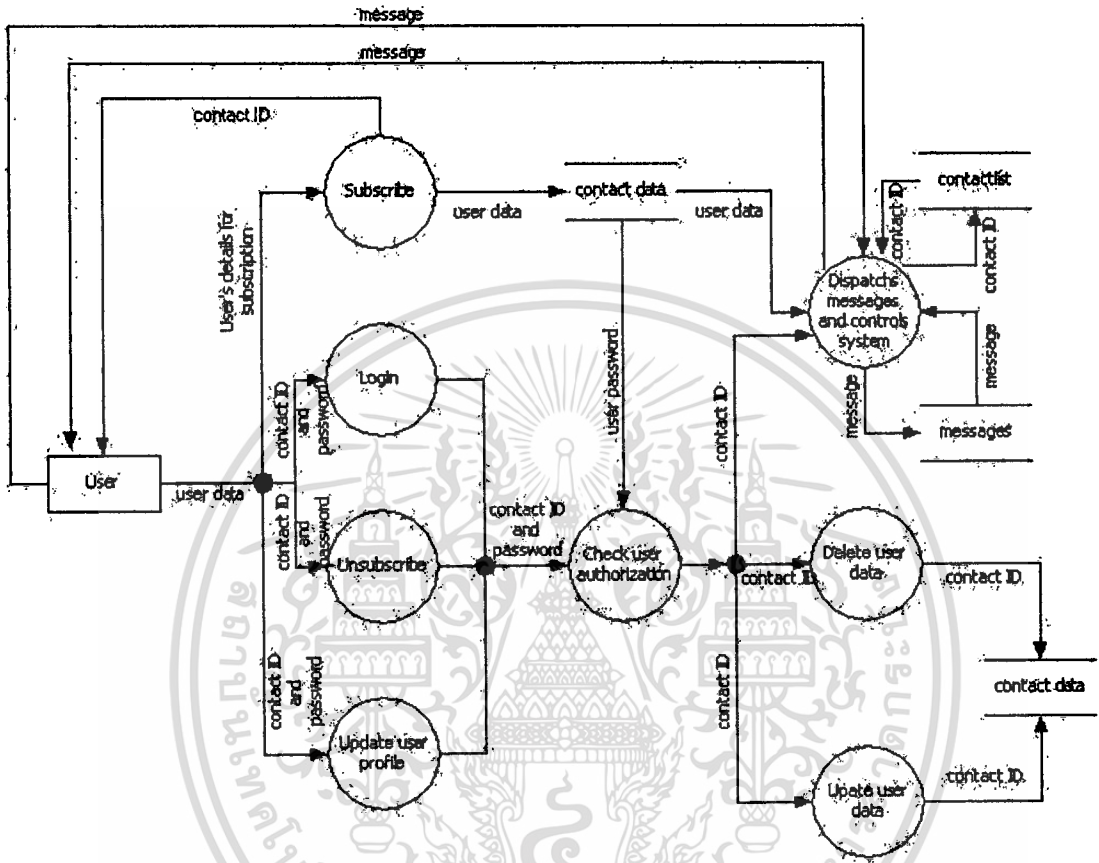


รูปที่ 3.8 แสดงการยกเลิกการเป็นสมาชิกของผู้ใช้

ขั้นตอนการทำงานของ Process Unsubscribe

1. รับข้อมูลหมายเลขประจำตัวและรหัสผ่านของผู้ใช้ระบบ
2. ตรวจสอบหมายเลขประจำตัวและรหัสผ่านจากฐานข้อมูลผู้ใช้ ถ้าผิดพลาดเช่น หมายเลขประจำตัวผู้ใช้ผิดหรือรหัสผ่านไม่ถูกต้องทำการแจ้งให้ผู้ใช้ทราบพร้อมแสดงหน้าจอให้ผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่านใหม่ ถ้าหมายเลขประจำตัวผู้ใช้และรหัสผ่านถูกต้องทำข้อ 3
3. แสดงหน้าจอเพื่อให้ผู้ใช้ยืนยันการยกเลิกการเป็นสมาชิก
4. ลบข้อมูลชื่อผู้ใช้ออกจากฐานข้อมูลผู้ใช้

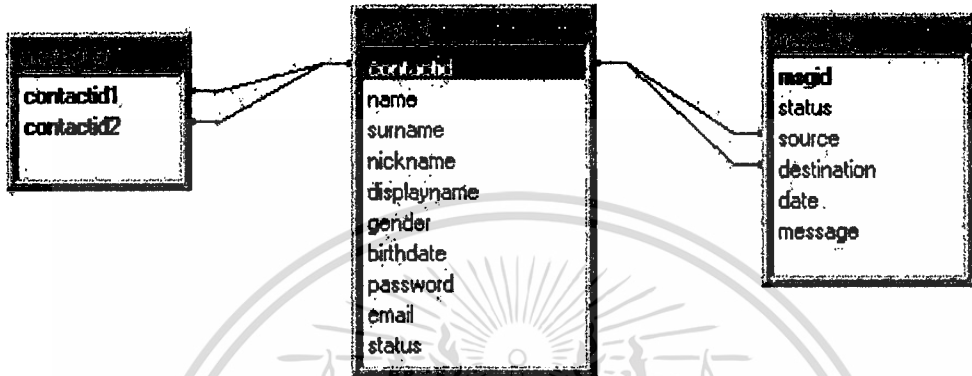
3.1.3 System Data Flow Diagram



รูปที่ 3.9 System Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 Entity Relationship ของระบบ



รูปที่ 3.10 ER Diagram

Entity contact หมายถึง ข้อมูลเกี่ยวกับผู้ใช้ระบบ ซึ่งจะประกอบด้วย Attributes ดังต่อไปนี้

- contactid หมายถึง หมายเลขประจำตัวผู้ใช้ระบบ และใช้ในการอ้างอิงในระบบ
- name หมายถึงชื่อของผู้ใช้ระบบ
- surname หมายถึงนามสกุลของผู้ใช้ระบบ
- nickname หมายถึงชื่อเล่นของผู้ใช้ระบบระบบ
- displayname หมายถึงชื่อที่ใช้แสดงในการใช้งานระบบของผู้ใช้ระบบ
- gender หมายถึงเพศของผู้ใช้ระบบ
- birthdate หมายถึงวันเดือนปีเกิดของผู้ใช้ระบบ
- password หมายถึงรหัสผ่านของผู้ใช้ (เข้ารหัสในการเก็บข้อมูลจริง)
- email หมายถึง E-mail ของผู้ใช้ระบบ
- status หมายถึงสถานะการใช้งานระบบของผู้ใช้ระบบ

Entity contactlist หมายถึง ข้อมูลที่เก็บรายการติดต่อของผู้ใช้ระบบ ประกอบด้วย Attributes ดังต่อไปนี้

- contactid1 หมายถึงหมายเลขประจำตัวของผู้ใช้ระบบ
- contactid2 หมายถึงหมายเลขประจำตัวผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้ระบบ

Entity message หมายถึง ข้อมูลที่เก็บข้อความที่ใช้ในการสื่อสารระหว่างผู้ใช้ และ ข้อมูลที่ใช้ในการควบคุมการทำงานในการสื่อสารของระบบ ประกอบด้วย Attributes ดังต่อไปนี้

- msgid หมายถึงหมายเลขประจำข้อความหรือข้อมูลระบบที่มีการติดต่อสื่อสารในระบบ
- status หมายถึงสถานะของคำสั่งที่ใช้ในการสื่อสารซึ่งต้องได้รับการแปลความหมายตามรูปแบบของโพรโตคอลที่ใช้งานในระบบ
- source หมายถึงหมายเลขประจำตัวผู้ใช้ที่ทำการส่งข้อความ หรือ ข้อมูลควบคุมในกรณีเซิร์ฟเวอร์ทำการส่งข้อมูลในระบบ
- destination หมายถึงหมายเลขประจำตัวผู้ใช้ที่เป็นผู้รับข้อความที่ส่งในการสื่อสาร
- date หมายถึงวันเวลาที่มีการส่งข้อความนั้น (Timestamp)
- message หมายถึงข้อความหรือข้อมูลที่ส่งระหว่างผู้ใช้ในระบบหรือระหว่างผู้ใช้กับเซิร์ฟเวอร์ ซึ่งจะต้องได้รับการแปลความหมายตามโพรโตคอลขึ้นอยู่กับหมายเลข status

3.2 การพัฒนาโปรแกรม

การพัฒนาโปรแกรมสื่อสารผ่านระบบเครือข่ายได้แบ่งเป็นส่วนใหญ่ดังนี้

3.2.1 การพัฒนาโปรแกรมฝั่งเซิร์ฟเวอร์

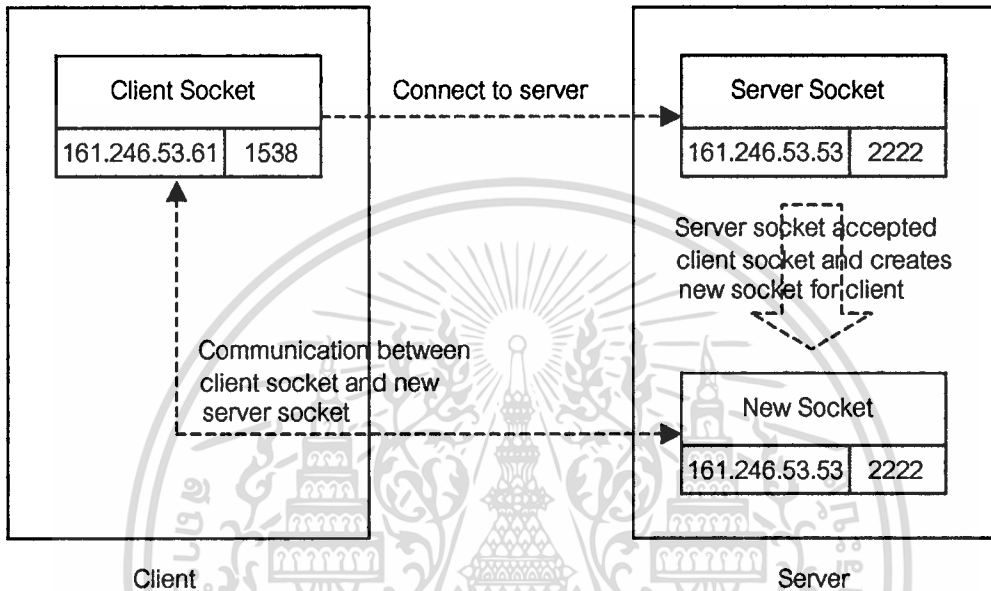
โปรแกรมฝั่งเซิร์ฟเวอร์ถือได้ว่าเป็นส่วนที่มีการทำงานที่สำคัญมาก อันเนื่องมาจากการทำงานของระบบต้องอ้างอิงการทำงานของฝั่งเซิร์ฟเวอร์เป็นส่วนใหญ่เพราะต้องการที่จะให้การใช้งานระบบของผู้ใช้มีปัญหาหรือเกิดความยุ่งยากในการทำงานให้น้อยที่สุดรวมถึงผู้ใช้ไม่ต้องยุ่งยากในการติดตั้งโปรแกรมหรือทำการปรับแต่งสภาพแวดล้อมของเครื่องคอมพิวเตอร์เพื่อให้ใช้งานระบบได้แต่อย่างใด จึงต้องเป็นภาระหน้าที่ที่จะต้องออกแบบและกำหนดการทำงานที่สามารถที่จะรองรับการทำงานของผู้ที่ต้องการใช้งานระบบได้ในคราวละหลาย ๆ คน

3.2.1.1 การพัฒนาโปรแกรมส่วนกลาง

โปรแกรมส่วนกลางเป็นโปรแกรมที่ทำหน้าที่ตอบรับการเข้าระบบ, การตรวจสอบการทำงานของผู้ใช้แต่ละคน และ ส่งต่อข้อมูลจากผู้ใช้รายหนึ่งไปยังผู้ใช้อีกรายหนึ่ง ซึ่งพัฒนาโปรแกรมได้ใช้ภาษา Perl ซึ่งเป็นภาษาที่สามารถทำงานในฝั่งเซิร์ฟเวอร์ได้เป็นอย่างดีและมีประสิทธิภาพในการใช้งานสูง โปรแกรมส่วนกลางจัดเก็บอยู่ในแฟ้มข้อมูลชื่อ `daemon.cgi` ซึ่งโปรแกรมส่วนกลางนี้ต้องทำงานอยู่ตลอดเวลาเพื่อรองรับการทำงานของผู้ใช้ ซึ่งในการทำงานจริงโปรแกรมส่วนกลางควรจะถูกเรียกให้ทำงานตั้งแต่ระบบปฏิบัติการเริ่มการทำงานเพราะทำให้ไม่ต้องทำการเรียกโปรแกรมให้ทำงานทุกครั้งที่มีการหยุดการทำงานของเซิร์ฟเวอร์แล้วเริ่มทำงานใหม่ การจะทำได้เช่นนั้นได้นั้นในระบบปฏิบัติการ Linux สามารถทำได้โดยกำหนดให้มีการเรียนใช้งานโปรแกรมในตอนเริ่มการทำงานของระบบด้วยการกำหนดการทำงานที่แฟ้มข้อมูล `/etc/rc.d/rc.local` ซึ่งในการเรียกใช้ควรกำหนดให้มีการทำงานเป็น Background Process ด้วยโดยการออกแบบการทำงานสามารถแบ่งเป็นขั้นตอนได้ดังนี้

- ทำหน้าที่ตอบรับการดำเนินงานของฝั่งไคลเอนต์โดยการทำหน้าที่ Server Socket โดยกำหนด port หมายเลข 2222 เพื่อรอรับการติดต่อของ Client Socket ในฝั่งไคลเอนต์
- หลังจากที่ได้รับ (Accept) การติดต่อจาก Client Socket จากฝั่งไคลเอนต์แล้ว โปรแกรมต้องสร้าง Socket ตัวใหม่เพื่อใช้ในการสื่อสารกับฝั่งไคลเอนต์ซึ่งต้องทำการแตกเป็น process ย่อยเพื่อให้เกิดความเป็นอิสระในการติดต่อส่งข้อมูลการสื่อสารระหว่างฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ ส่วนโปรแกรมส่วนกลาง (Parent

process) ก็จะรอการตอบรับการติดต่อจาก Client Socket จากฝั่งไคลเอนต์อื่นต่อไป ดังรูปที่ 3.11



รูปที่ 3.11 การเชื่อมต่อระหว่าง Server Socket และ Client Socket

- หลังจากที่ทำกาการแตกเป็น Process ย่อยเพื่อรองรับการทำงานกับ Client Socket แล้ว Socket ตัวใหม่จะทำหน้าที่สื่อสารกับ Client Socket โดยใช้ Protocol ที่สร้างขึ้นมาในการสื่อสาร (จะกล่าวถึงรายละเอียดของ Protocol ในหัวข้อถัดไป) พร้อมกับตรวจสอบการทำงานของไคลเอนต์ในการติดต่อกับฐานข้อมูลของระบบเพื่อเริ่มการทำงานของระบบสื่อสาร (Communication initialization) เช่นส่งข้อมูลที่จำเป็นในการสื่อสารได้แก่ ชื่อที่ใช้ในการสื่อสารของผู้ใช้และรายชื่อที่ผู้ใช้มีอยู่ในรายการติดต่อ พร้อมทั้งส่งข้อมูลแจ้งให้ผู้ใช้ที่เกี่ยวข้องกับผู้ใช้ที่อยู่ในสถานะ online ให้ทราบถึงการเข้าสู่สถานะ online ของผู้ใช้ ซึ่งการส่งข้อมูลเพื่อให้ผู้ใช้รายอื่นที่เกี่ยวข้องทราบนั้นทำได้โดยเพิ่มข้อมูลในตาราง MESSAGE ในฐานข้อมูลซึ่งตาราง MESSAGE เป็นตาราง Transaction ส่วนกลางที่ใช้ในการสื่อสารระหว่างผู้ใช้

- ในการทำงานต้องทำการแตกเป็น Process ย่อยอีกหนึ่ง Process เพื่อใช้ในการส่งข้อมูลส่งให้กับไคลเอนต์ซึ่งข้อมูลดังกล่าวได้จากตาราง MESSAGE ซึ่งความจำเป็นที่ต้องแตกการทำงานเป็นอีกหนึ่ง process นั้นเพราะเหตุว่าเมื่อ Server socket สร้าง socket ตัวใหม่ที่ใช้ในการติดต่อกับ Client Socket นั้นวงรอบ (Loop) ที่ใช้ในการสื่อสารข้อมูลระหว่างไคลเอนต์กับเซิร์ฟเวอร์จะดำเนินการทำงานต่อไปได้ก็ต่อเมื่อได้รับข้อมูลจากฝั่งไคลเอนต์ เมื่อไม่มีข้อมูลจากไคลเอนต์จะไม่เกิดการวนของรอบการทำงาน จึงต้องกำหนดให้เป็น process ย่อยอีกหนึ่ง process เพื่อที่จะทำให้การทำงานในการส่งข้อมูลที่ได้รับมาจากผู้ใช้อื่นไปยังผู้ใช้ฝั่งไคลเอนต์เป็นไปได้โดยอิสระและผู้ใช้สามารถที่จะได้รับข้อมูลหรือข้อความในทันทีที่มีข้อมูลเข้าสู่ระบบ พร้อมทั้งทำการลบข้อความที่ผู้ใช้ได้รับออกจากระบบ
- รับข้อมูลการเปลี่ยนแปลงจากผู้ใช้และส่งข้อมูลการเปลี่ยนแปลงให้ผู้ใช้ เช่น
 - แจ้งการเข้าสู่สถานะ online ของผู้ใช้ให้ผู้ใช้อื่นที่เกี่ยวข้องที่อยู่ในสถานะ online ทราบ โดยโปรแกรมส่วนกลางจะตรวจสอบข้อมูลของผู้ใช้อื่นที่เกี่ยวข้องและอยู่ในสถาน online จากฐานข้อมูล (ตาราง CONTACTLIST และตาราง CONTACT) แล้วทำการเพิ่มข้อมูลลงในตาราง MESSAGE เพื่อที่ผู้ใช้อื่นที่เกี่ยวข้องจะสร้างรับทราบได้
 - ผู้ใช้ทำการเพิ่มรายชื่อผู้ใช้อื่นเข้าสู่รายการติดต่อ ซึ่งระบบการทำงานของโปรแกรมส่วนกลางต้องทำการตรวจสอบว่าหมายเลขประจำตัวผู้ใช้ที่ผู้ใช้ต้องการเพิ่มนั้นมีอยู่ในระบบหรือไม่ (ตรวจสอบจากตาราง CONTACT) ถ้ามีอยู่จริงจึงจะทำการเพิ่มข้อมูลการติดต่อในฐานข้อมูล (ตาราง CONTACTLIST) แล้วแจ้งให้ผู้ใช้ทราบว่าได้ทำการเพิ่มข้อมูลเป็นผลสำเร็จ ซึ่งจะส่งข้อมูลไปให้กับระบบทางฝั่งไคลเอนต์ทราบว่าได้ทำการเพิ่มสำเร็จระบบในฝั่งไคลเอนต์จึงสามารถที่จะเพิ่มชื่อที่ใช้ในการติดต่อของผู้ใช้อื่น (ได้จากตาราง CONTACT) ที่โปรแกรมส่วนกลางส่งกลับมายังไคลเอนต์เข้าสู่รายการติดต่อของผู้ใช้ แต่หากไม่สามารถเพิ่มข้อมูลในฐานข้อมูลได้โปรแกรมส่วนกลางก็จะแจ้งให้ระบบทางฝั่งไคลเอนต์ทราบ เมื่อระบบทางฝั่งไคลเอนต์ทราบว่าไม่สามารถเพิ่มการทำงานได้ก็จะไม่ทำการเพิ่มรายชื่อนั้นในรายการติดต่อของผู้ใช้
 - แจ้งให้ผู้ใช้ทราบหลังจากเพิ่มรายชื่อผู้ใช้อื่น ในกรณีที่ผู้ใช้อื่นที่ผู้ใช้เพิ่มเข้าสู่รายการติดต่อไม่ได้อยู่ในสถานะ online

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แจ้งให้ผู้ใช้คนอื่นที่ผู้ใช้ได้ทำการเพิ่มเข้าสู่รายการติดต่อทราบในการนี้ผู้ใช้รายนั้นอยู่ในสถานะ online โดยการเพิ่มข้อมูลลงในตาราง MESSAGE ในฐานข้อมูล
- ผู้ใช้ลบรายชื่อผู้อื่นจากรายการการติดต่อของผู้ใช้ โปรแกรมส่วนกลางจะตรวจสอบและลบข้อมูลจากตาราง CONTACTLIST ในฐานข้อมูลพร้อมทั้งแจ้งการลบให้ผู้ใช้ทราบว่าได้ทำการลบสำเร็จเพื่อที่ระบบในฝั่งไคลเอนต์จะทำการลบชื่อออกจากรายการติดต่อของผู้ใช้ต่อไป และทำการส่งข้อมูลให้ระบบการทำงานฝั่งไคลเอนต์ผู้ใช้คนที่ถูกลบรายชื่อออกจากรายการติดต่อของผู้ใช้ทราบและทำการลบรายชื่อของผู้ใช้ออกจากรายการติดต่อของผู้ใช้รายนั้น การส่งข้อมูลทำได้โดยการเพิ่มข้อมูลที่ตาราง MESSAGE
- ผู้ใช้ระบบทำการส่งข้อมูลระหว่างผู้ใช้ระบบ ผู้ใช้จะส่งข้อมูลมาที่โปรแกรมส่วนกลางผ่านการทำงานของ Socket โดยใช้โปรโตคอลที่ได้กำหนดไว้มายังโปรแกรมส่วนกลาง เมื่อโปรแกรมส่วนกลางตรวจสอบว่าเป็นคำสั่งในการส่งข้อความก็จะทำการเพิ่มข้อมูลเข้าสู่ตาราง MESSAGE และทำการส่งต่อให้กับผู้ใช้คนอื่นที่ผู้ใช้ต้องการส่งข้อมูลให้
- ผู้ใช้ออกจากการทำงานของระบบ (Offline) เมื่อทราบว่าผู้ใช้ตัดการเชื่อมต่อเช่น ปิดเบราว์เซอร์ โปรแกรมส่วนกลางจะทราบถึงการขาดการเชื่อมต่อระหว่างฝั่งไคลเอนต์และเซิร์ฟเวอร์ โปรแกรมส่วนกลางจะทำการแก้ไขข้อมูลที่ใช้ในการแสดงสถานะการใช้งานระบบ (online/offline) โดยกำหนดให้ฟิลด์ status ในตาราง CONTACT มีค่าเท่ากับศูนย์ พร้อมทั้งแจ้งให้ผู้ใช้รายอื่นที่เกี่ยวข้องกับผู้ใช้ที่ offline ทราบถึงการ offline โดยโปรแกรมส่วนกลางจะตรวจสอบจากตาราง CONTACTLIST และ ตาราง CONTACT ในฐานข้อมูลพร้อมทั้งแจ้งให้ทราบโดยการเพิ่มข้อมูลการ offline ของผู้ใช้ที่ offline ในตาราง MESSAGE

หมายเหตุ การเพิ่มข้อมูลในตาราง MESSAGE ในฐานข้อมูลเพื่อใช้ในการสื่อสารระหว่างผู้ใช้ระบบกับผู้ใช้ระบบหรือระหว่างผู้ใช้ระบบกับโปรแกรมส่วนกลางนั้นจะต้องเป็นไปตามกฎการส่งข้อมูลหรือ Protocol ที่ได้กำหนดไว้สำหรับการส่งข้อมูลที่ได้กำหนดขึ้นซึ่งจะกล่าวถึงในหัวข้อถัดไป

3.2.1.2 การพัฒนาโปรแกรมเพื่อใช้ในการสื่อสารระหว่างผู้ใช้และเซิร์ฟเวอร์

การพัฒนาโปรแกรมเพื่อใช้ในการติดต่อระหว่างผู้ใช้กับระบบจะใช้โปรแกรมที่ทำงานแบบ CGI (Common Gateway Interface) ซึ่งพัฒนาโดยใช้ภาษา Perl และอยู่ในฝั่งเซิร์ฟเวอร์ผู้ใช้สามารถที่จะติดต่อผ่านทางเว็บเบราว์เซอร์เพื่อใช้ในการติดต่อส่งข้อมูลระหว่างเว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์ทำงานผ่านโพรโตคอล HTTP ผ่านระบบการทำงานของ HTML ซึ่ง HTML ได้จากการสร้างของโปรแกรม Perl การทำงานของระบบแบ่งได้ดังนี้

- โปรแกรม หน้าจอหลักจัดเก็บในชื่อ main.cgi ใช้ในการเรียกโปรแกรมส่วนการทำงานอื่น ๆ ของระบบ
- โปรแกรม User Login ใช้สำหรับรับข้อมูลจากผู้ใช้ในการเข้าใช้งานระบบ โดยจะกำหนดแบบฟอร์มให้ผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่านซึ่ง CGI จะรับพารามิเตอร์เป็นหมายเลขประจำตัวและรหัสผ่านของผู้ใช้เพื่อใช้ในการตรวจสอบจากฐานข้อมูลว่าเป็นผู้ใช้ตัวจริงหรือไม่ ซึ่งถ้าเป็นผู้ใช้ตัวจริงก็จะสร้างหน้าจอ HTML ซึ่งจะใช้เป็นหน้าจอที่ติดต่อกับโปรแกรมที่ใช้ทำหน้าที่ฝั่งไคลเอนต์ในการติดต่อระบบต่อไป โปรแกรมจะจัดเก็บในเพิ่มข้อมูลชื่อ login.cgi
- โปรแกรม User Subscription ใช้สำหรับการสมัครเข้าเป็นสมาชิกเพื่อใช้งานระบบ ซึ่งโปรแกรมจะกำหนดแบบฟอร์มให้ผู้ที่ต้องการสมัครกรอกรายละเอียดส่วนตัวเพื่อใช้ในการติดต่อ หลังจากกรอกข้อมูลผู้ใช้เสร็จเรียบร้อยแล้วและผู้ใช้ตอบตกลงสมัครสมาชิกแล้ว CGI จะรับข้อมูลที่ผู้ใช้ป้อนลงในฐานข้อมูลพร้อมทั้งกำหนดหมายเลขประจำตัวให้กับผู้สมัครเพื่อใช้ในการเข้าใช้งานระบบต่อไป โปรแกรมเก็บในเพิ่มข้อมูลชื่อ usersubscribe.cgi
- โปรแกรม User Unsubscription ใช้สำหรับการยกเลิกการเป็นสมาชิกของผู้ใช้ระบบ โดยที่ผู้ใช้ระบบที่ต้องการยกเลิกการเป็นสมาชิกต้องทำการกรอกหมายเลขประจำตัวและรหัสผ่านเพื่อเป็นการยืนยันตัวตนของผู้ใช้ ซึ่ง CGI จะรับข้อมูลหมายเลขประจำตัวและรหัสผ่านของผู้ใช้พร้อมทั้งรหัสผ่านของผู้ใช้เพื่อใช้ในการตรวจสอบกับฐานข้อมูลเมื่อโปรแกรมตรวจสอบแล้วว่าเป็นผู้ใช้ระบบตัวจริงที่ต้องการยกเลิกการเป็นสมาชิกก็จะสร้างหน้าจอเพื่อให้ผู้ใช้ยืนยันเมื่อผู้ใช้ยืนยันการยกเลิก CGI จะรับข้อมูลแล้วทำการลบข้อมูลผู้ใช้จากฐานข้อมูลที่อยู่ฝั่งเซิร์ฟเวอร์ โปรแกรมเก็บอยู่ในเพิ่มข้อมูล unsubscribe.cgi
- โปรแกรม User Profile ใช้สำหรับให้ผู้ใช้สามารถเปลี่ยนแปลงข้อมูลส่วนตัวโดยโปรแกรมจะสร้างแบบฟอร์มเพื่อแสดงข้อมูลส่วนตัวของผู้ใช้ หลังจากผู้ใช้ทำ

การแก้ไขข้อมูลส่วนตัวแล้ว CGI จะส่งข้อมูลแล้วทำการตรวจสอบข้อมูลที่มีการเปลี่ยนแปลงกับฐานข้อมูลแล้วทำการเปลี่ยนแปลงข้อมูลในฐานข้อมูล ซึ่งก่อนที่จะเข้าทำการเปลี่ยนแปลงข้อมูลได้นั้นผู้ใช้ต้องทำการกรอกหมายเลขประจำตัวและรหัสผ่านของผู้ใช้เพื่อยืนยันตัวตนจริงเมื่อ CGI รับข้อมูลและทำการตรวจสอบกับฐานข้อมูลในฝั่งเซิร์ฟเวอร์แล้วผู้ใช้จึงจะสามารถเข้าสู่แบบฟอร์มที่แสดงข้อมูลส่วนตัวของผู้ใช้และทำการเปลี่ยนแปลงได้ โปรแกรมเก็บอยู่ในแฟ้มข้อมูลชื่อ userprofile.cgi

3.2.2 การพัฒนาโปรแกรมฝั่งไคลเอนต์

การพัฒนาโปรแกรมในฝั่งไคลเอนต์เป็นโปรแกรมที่พัฒนาด้วย JAVA Applet ซึ่งถูกจัดเก็บอยู่ที่เซิร์ฟเวอร์และสามารถเรียกใช้ผ่านทางหน้าจอบริบทที่ถูกสร้างโดยโปรแกรม User Login เมื่อผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่านได้ถูกต้อง ซึ่งโปรแกรม Java Applet จะจัดเก็บในชื่อ Client.class ซึ่งเป็นโปรแกรมภาษา JAVA Applet ที่ได้รับการ compile ให้เป็น Byte code เรียบร้อยแล้วและเมื่อผู้ใช้เข้าสู่หน้าจอบริบทที่ใช้ในสื่อสารเว็บเบราว์เซอร์จะทำการ download แฟ้มข้อมูล Client.class มาที่เครื่องของผู้ใช้โดยอัตโนมัติซึ่ง โปรแกรม Client.class ทำหน้าที่ดังต่อไปนี้

- ทำการสร้าง Client Socket เพื่อใช้ในการเชื่อมต่อกับ Server Socket ในฝั่งไคลเอนต์ที่สร้างไว้รอรับการเชื่อมต่อโดยโปรแกรมส่วนกลางที่ฝั่งเซิร์ฟเวอร์
- หลังจากที่ทำการเชื่อมต่อ Socket สำเร็จแล้ว โปรแกรมจะทำการส่งข้อมูลเพื่อใช้ในการเริ่มต้นการทำงานของระบบสื่อสาร ตามรูปแบบของ Protocol ที่ได้กำหนดไว้สำหรับการสื่อสารของระบบนี้โดยเฉพาะ ซึ่งจะประกอบด้วยรหัสการเริ่มใช้งานระบบและหมายเลขประจำตัวของผู้ใช้เมื่อ โปรแกรมส่วนกลางในฝั่งเซิร์ฟเวอร์ได้รับข้อมูลดังกล่าวที่แสดงถึงการเริ่มต้นใช้งานระบบของผู้ใช้โปรแกรมส่วนกลาง จะทำการติดต่อกับฐานข้อมูลและทำการส่งข้อมูลที่จะใช้ในการสื่อสารเช่นชื่อของผู้ใช้ที่ใช้ในการสื่อสารและรายชื่อผู้ใช้อื่น ๆ ที่อยู่ในรายการติดต่อของผู้ใช้กลับมาที่โปรแกรมฝั่งไคลเอนต์ เมื่อโปรแกรมฝั่งไคลเอนต์ได้รับข้อมูลจากฝั่งไคลเอนต์ที่แสดงถึงรหัสของการเริ่มเข้าสู่ระบบสื่อสารสำเร็จ, ชื่อของผู้ใช้ระบบ และ รายชื่อของผู้ใช้อื่นที่อยู่ในรายการของผู้ใช้โปรแกรมฝั่งไคลเอนต์ก็จะปรับเปลี่ยนการทำงานเพื่อที่จะให้ผู้ใช้สามารถใช้งานระบบได้ต่อไป

- ในกรณีที่ผู้ใช้ต้องการเพิ่มรายชื่อของผู้ใช้อื่นที่อยู่ในระบบในรายการของผู้ใช้ก็สามารถทำได้โดยเลือกคำสั่งเพิ่มข้อมูลเข้าสู่รายการติดต่อโดยเลือกที่คำสั่ง (Add) ซึ่งโปรแกรมฝั่งไคลเอนต์จะทำการส่งข้อมูลตามรูปแบบ โพรโตคอลที่ได้กำหนดไว้โดยกำหนดในรูปแบบของการเพิ่มข้อมูลไปยังโปรแกรมส่วนกลางที่ทำงานในฝั่งเซิร์ฟเวอร์ เมื่อโปรแกรมในฝั่งเซิร์ฟเวอร์ได้รับข้อมูลทำการตรวจสอบการเพิ่มข้อมูลกับฐานข้อมูลก็จะส่งข้อมูลตามรูปแบบ โพรโตคอลที่ได้กำหนดไว้กลับมาที่โปรแกรมในฝั่งไคลเอนต์ว่าสามารถทำการเพิ่มได้หรือไม่ ถ้าไม่สามารถเพิ่มข้อมูลได้อาจเนื่องมาจากหมายเลขประจำตัวของผู้ใช้ระบบอื่นที่ผู้ใช้ต้องการเพิ่มในรายการติดต่อของผู้ใช้ไม่พบในระบบ โปรแกรมส่วนกลางก็จะส่งข้อความกลับมาที่โปรแกรมทางฝั่งไคลเอนต์ ซึ่ง โปรแกรมฝั่งไคลเอนต์ก็จะแจ้งให้ผู้ใช้ทราบต่อไป แต่ถ้าโปรแกรมส่วนกลางสามารถเพิ่มข้อมูลได้สำเร็จก็จะแจ้งกลับมาที่โปรแกรมฝั่งไคลเอนต์ทราบ โปรแกรมทางฝั่งไคลเอนต์ก็จะทำนำข้อมูลซึ่งได้แก่ หมายเลขประจำตัวของผู้ใช้และชื่อของผู้ใช้อื่นที่ผู้ใช้ต้องการติดต่อกับ เพิ่มเข้าที่รายการติดต่อของผู้ใช้
- ในกรณีที่ผู้ใช้ต้องการลบชื่อของผู้ใช้ในรายการติดต่อของผู้ใช้ เมื่อผู้ใช้เลือกคำสั่งลบ (Remove) โปรแกรมฝั่งไคลเอนต์จะส่งข้อมูลที่กำหนดตามรูปแบบ โพรโตคอลที่ได้กำหนดไว้โดยกำหนดเป็นข้อมูลที่แสดงการลบข้อมูลไปยังโปรแกรมส่วนกลาง เมื่อโปรแกรมที่ทำงานส่วนกลางได้รับข้อมูลและแปลความหมายว่าผู้ใช้ต้องการลบข้อมูลออกจากรายการติดต่อ โปรแกรมส่วนกลางจะทำการตรวจสอบจากฐานข้อมูลในอยู่ในเซิร์ฟเวอร์แล้วทำการลบข้อมูลในฐานข้อมูลแล้วส่งข้อมูลให้โปรแกรมในฝั่งไคลเอนต์ทราบว่าสามารถทำการลบข้อมูลได้หรือไม่ โดยส่งข้อมูลตาม โพรโตคอลที่ได้กำหนดไว้สำหรับการแจ้งให้ไคลเอนต์ทราบ เมื่อไคลเอนต์ได้รับข้อมูลและทำการแปลความหมาย ถ้าเป็นข้อมูลที่แสดงว่าโปรแกรมส่วนกลางไม่สามารถทำการลบได้ โปรแกรมฝั่งไคลเอนต์ก็จะทำการแจ้งให้ผู้ใช้ทราบว่าไม่สามารถลบข้อมูลได้ แต่ถ้าโปรแกรมส่วนกลางสามารถลบข้อมูลได้โปรแกรมฝั่งไคลเอนต์จึงจะสามารถลบรายชื่อผู้ใช้ระบบอื่นที่อยู่ในรายการติดต่อของผู้ใช้ได้
- เมื่อมีผู้ใช้อื่นที่อยู่ในรายการติดต่อ (ในฐานข้อมูล) ของผู้ใช้เข้าสู่สถานะ online (อยู่ในข้อมูลการติดต่อกับผู้ใช้แต่ไม่ได้เข้าใช้งานระบบและเพิ่งจะเข้าใช้งานระบบ จึงไม่ได้แสดงในรายการติดต่อใน โปรแกรมฝั่งไคลเอนต์) ซึ่งโปรแกรมส่วนกลาง

จะทำการแจ้งให้ผู้ใช้ทราบพร้อมทั้งส่งข้อมูลที่เกี่ยวข้องกับผู้ใช้ที่อยู่ในรายการติดต่อของผู้ใช้ ซึ่งได้แก่หมายเลขประจำตัว และ ชื่อของผู้ใช้ที่เข้าสู่สถานะ online ซึ่งโปรแกรมทางฝั่งไคลเอนต์จะใช้ในโปรแกรมแสดงในรายการติดต่อของผู้ใช้และใช้ในการส่งข้อมูลระหว่างผู้ใช้ต่อไป

- เมื่อผู้ใช้ที่อยู่ในรายการติดต่อของผู้ใช้ออกจากระบบการใช้งาน (อยู่ในสถานะ Offline และไม่ได้แสดงในรายการการติดต่อของผู้ใช้ในโปรแกรมฝั่งไคลเอนต์) โปรแกรมส่วนกลางซึ่งตรวจสอบการใช้งานระบบจะส่งข้อมูลเมื่อทราบว่าผู้ใช้ที่ออกจากระบบอยู่ในฐานข้อมูลที่เกี่ยวข้องกับการสื่อสารของผู้ใช้ก็จะแจ้งให้ผู้ใช้ทราบโดยส่งข้อมูลมาที่โปรแกรมฝั่งไคลเอนต์โดยระบุเป็นการแจ้งการ offline ของผู้ใช้ในรายการติดต่อของผู้ใช้ซึ่งประกอบด้วยหมายเลขประจำตัวของผู้ใช้ที่ออกจากระบบ ซึ่งเมื่อโปรแกรมฝั่งไคลเอนต์แปลความหมายและทราบว่าเป็นการออกจากระบบของผู้ใช้อื่นในระบบที่อยู่ในรายการติดต่อของผู้ใช้ โปรแกรมก็จะทำการลบออกจากรายการติดต่อที่แสดงไว้ที่โปรแกรม ทำให้ผู้ใช้ทราบว่าผู้ใช้รายนั้นได้ออกจากระบบไปแล้ว
- เมื่อผู้ใช้ถูกเพิ่มเข้าไปในรายการติดต่อของผู้ใช้ซึ่งผู้ใช้ที่ทำการเพิ่มก็จะทำการแจ้งให้โปรแกรมส่วนกลางในฝั่งเซิร์ฟเวอร์ทราบซึ่ง โปรแกรมก็จะแจ้งมาที่โปรแกรมในฝั่งไคลเอนต์ที่ผู้ใช้ใช้งานอยู่ได้รับทราบว่าผู้ใช้คนอื่นได้ทำการเพิ่มข้อมูลรายการติดต่อซึ่งหมายถึงผู้ใช้ ก็จะใช้ข้อมูลที่โปรแกรมส่วนกลางส่งมาให้ซึ่งโปรแกรมฝั่งไคลเอนต์ตีความตามความหมายโพรโตคอลที่กำหนดไว้ว่าเป็นการเพิ่มก็จะใช้ข้อมูลดังกล่าวซึ่งจะประกอบด้วย หมายเลขและชื่อที่ใช้ในการสื่อสารของผู้ใช้อื่นที่ทำการเพิ่มผู้ใช้ในรายการติดต่อ ซึ่งโปรแกรมฝั่งไคลเอนต์จะใช้ในการเพิ่มข้อมูลรายการติดต่อของผู้ใช้ พร้อมทั้งแสดงข้อความให้ผู้ใช้ทราบด้วย
- เมื่อผู้ใช้ถูกลบออกจากรายการติดต่อของผู้ใช้ซึ่งผู้ใช้ที่ทำการลบก็จะทำการติดต่อสื่อสารกับโปรแกรมส่วนกลางและระบุคำสั่งที่ส่งไปที่โปรแกรมส่วนกลางที่เซิร์ฟเวอร์ให้เซิร์ฟเวอร์ทราบว่าเป็นการลบข้อมูล โปรแกรมส่วนกลางก็จะติดต่อกับฐานข้อมูลที่อยู่ที่เซิร์ฟเวอร์เมื่อทำการลบข้อมูลได้แล้วจึงจะทำการส่งข้อมูลที่ระบุถึงการลบมายัง โปรแกรมฝั่งไคลเอนต์ผู้ใช้ซึ่งถูกลบทำงานอยู่ ซึ่งโปรแกรมฝั่งไคลเอนต์ก็จะทำการลบข้อมูลชื่อของผู้ใช้ที่ลบผู้ใช้ออกจากรายการติดต่อออกจากรายการติดต่อของผู้ใช้ที่ถูกลบ

- การส่งข้อความระหว่างผู้ใช้ระบบเมื่อผู้ส่งข้อความโดยเลือกรายชื่อผู้ใช้อื่นที่ต้องการส่งข้อความสื่อสารด้วยในรายการติดต่อของผู้ใช้ที่แสดงใน โปรแกรมฝั่งไคลเอนต์ แล้วเลือกคำสั่งส่งในหน้าจอสื่อสาร โปรแกรมฝั่งไคลเอนต์ก็จะทำการสร้างรูปแบบของข้อมูลตามโพรโตคอลที่ระบุเป็นการส่งข้อความไปที่โปรแกรมส่วนกลางที่ทำงานที่เซิร์ฟเวอร์ซึ่ง โปรแกรมส่วนกลางก็จะแปลความหมายและทำการส่งข้อมูลไปให้ผู้ใช้อื่นที่ผู้ใช้ต้องการส่งข้อมูลให้

3.2.3 ข้อตกลงในการส่งข้อมูล (Messaging Protocol)

ข้อตกลงหรือ โพรโตคอลที่ใช้ในการสื่อสารของระบบสื่อสารผ่านเครือข่ายอินเทอร์เน็ตที่ได้สร้างขึ้นนี้เป็นโพรโตคอลเฉพาะการทำงานของระบบนี้เพื่อให้การสื่อสารระหว่างโปรแกรมที่ทำงานที่ฝั่งเซิร์ฟเวอร์และ โปรแกรมที่ทำงานในฝั่งไคลเอนต์สามารถที่จะสื่อสารกันได้อย่างถูกต้องและเป็นไปตามความต้องการของการทำงานของระบบ โดยข้อตกลงหรือโพรโตคอลมีรูปแบบดังรูปที่ 3.12

Status	Source ID	Destination ID	Message
--------	-----------	----------------	---------

รูปที่ 3.12 แสดงรูปแบบของโพรโตคอลที่ใช้ในระบบ

ความหมาย

Status	หมายถึงสถานะของคำสั่งเป็นหมายเลขขนาดหนึ่งไบต์
Source ID	หมายถึงหมายเลขประจำตัวของผู้ส่งข้อมูลเป็นหมายเลขขนาด 10 digits
Destination ID	หมายถึงหมายเลขประจำตัวของผู้รับข้อมูลเป็นหมายเลขขนาด 10 digits
Message	หมายถึงข้อมูลซึ่งสามารถเปลี่ยนแปลงได้ตาม Status

3.2.4 การทำงานของระบบและโพรโตคอลที่ใช้ในการสื่อสาร

ภายหลังจากระบบในฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ได้ทำการเชื่อมต่อ socket กันเป็นที่เรียบร้อยแล้วระบบจะเริ่มการทำงานโดย

1. ระบบเริ่มการทำงานโดยฝั่งไคลเอนต์ส่งข้อมูลแสดงการร้องขอการเริ่มต้นการทำงานไปที่การทำงานฝั่งเซิร์ฟเวอร์โดยมีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งไคลเอนต์

Status	000
Source ID	หมายเลขประจำตัวผู้ใช้ระบบ
Destination ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Message	Null

หลังจากการทำงานฝั่งเซิร์ฟเวอร์ได้รับข้อมูลการเริ่มต้นระบบเซิร์ฟเวอร์ทำการอ่านข้อมูลผู้ใช้ระบบจากฐานข้อมูลตามหมายเลขประจำตัวของผู้ใช้ระบบ แล้วส่งข้อมูลกลับไปให้ฝั่งไคลเอนต์โดยมีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	000
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบ
Message	<ol style="list-style-type: none"> 1. ชื่อที่ใช้ในการสื่อสารของผู้ใช้ 2. รายการการติดต่อของผู้ใช้ประกอบด้วย <ol style="list-style-type: none"> 2.1 หมายเลขประจำตัวผู้ใช้ที่อยู่ในรายการการติดต่อ 2.2 ชื่อที่ใช้ในการสื่อสารของผู้ใช้นั้น 2.3 สถานะการใช้งานระบบ (online/offline) <p>หมายเหตุ แต่ละส่วนของ message จะขึ้นด้วยเครื่องหมาย เพื่อเป็นการแยกข้อมูล</p>

2. หลังจากที่ได้รับข้อมูลเพื่อใช้ในการเริ่มการทำงานแล้วของระบบแล้ว เซิร์ฟเวอร์ต้องทำการแจ้งให้ผู้ใช้ที่อยู่ในรายการติดต่อของผู้ใช้ทราบถึงสถานะของผู้ใช้ (online/offline)

2.1 เมื่อผู้ใช้เข้าสู่สถานะonline (เข้าใช้งานระบบ) มีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	200
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่เข้าสู่สถานะ online
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่อยู่ในรายการการติดต่อ
Message	ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่เข้าสู่สถานะ online

2.2 เมื่อผู้ใช้เข้าสู่สถานะ offline (ออกจากระบบ) มีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	201
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่เข้าสู่สถานะ offline
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่อยู่ในรายการการติดต่อ
Message	ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่เข้าสู่สถานะ offline

3. ในกรณีที่ผู้ใช้ต้องการเพิ่มรายชื่อผู้อื่นในรายการการติดต่อของผู้ใช้ หลังจากที่ใช้ทำการกรอกหมายเลขประจำตัวผู้อื่นที่ต้องการเพิ่มการทำงานฝั่งไคลเอนต์จะส่งข้อมูลไปที่ฝั่งเซิร์ฟเวอร์ โดยมีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งไคลเอนต์

Status	100
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Destination ID	หมายเลขประจำตัวผู้อื่นที่ผู้ใช้ต้องการเพิ่ม
Message	ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่ทำการเพิ่ม

หลังจากที่ฝั่งเซิร์ฟเวอร์ได้รับข้อมูลการเพิ่มจากฝั่งไคลเอนต์ เซิร์ฟเวอร์ทำการข้อมูลของผู้ใช้ตามหมายเลข Destination ID จากฐานข้อมูลผู้ใช้ระบบ แล้วทำการเพิ่มข้อมูลรายการการติดต่อของผู้ใช้และผู้ใช้อื่นที่ผู้ใช้ทำการเพิ่มในรายการการติดต่อ จัดเก็บข้อมูลลงฐานข้อมูลเพื่อให้ผู้ใช้คนอื่นที่ถูกรับทราบผ่านการอ่านข้อมูลจากฐานข้อมูลที่ใช้ในการเก็บข้อความ โดยมีรูปแบบโปรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	202
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Destination ID	หมายเลขประจำตัวผู้ใช้คนอื่นที่ผู้ใช้ต้องการเพิ่ม
Message	ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่ทำการเพิ่ม

หลังจากนั้นเซิร์ฟเวอร์จะทำการแจ้งให้ผู้ใช้ทราบว่าได้ทำการเพิ่มข้อมูลสำเร็จพร้อมทั้งส่งข้อมูลเกี่ยวกับผู้ใช้คนอื่นที่ผู้ใช้เพิ่มกลับไปให้ผู้ใช้ โดยมีรูปแบบโปรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	100
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Message	<ol style="list-style-type: none"> หมายเลขประจำตัวผู้ใช้ที่ถูกเพิ่ม ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่ผู้เพิ่ม สถานะการใช้งานระบบ (online/offline) หมายเหตุ แต่ละส่วนของ message จะขึ้นด้วยเครื่องหมาย เพื่อเป็นการแยกข้อมูล

ในกรณีหากเกิดความผิดพลาดเซิร์ฟเวอร์ไม่สามารถเพิ่มข้อมูลในฐานข้อมูลได้ เซิร์ฟเวอร์จะแจ้งให้ผู้ใช้ทราบ โดยมีรูปแบบโพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	503
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Message	ข้อความแสดงความผิดพลาดในที่นี้คือไม่สามารถเพิ่มข้อมูลหมายเลขนี้ได้

หรือในกรณีที่ผู้ใช้กรอกหมายเลขประจำตัวที่ไม่มีอยู่ในระบบ เซิร์ฟเวอร์จะแจ้งให้ผู้ใช้ทราบ โดยมีรูปแบบโพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	501
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Message	ข้อความแสดงความผิดพลาดในที่นี้คือไม่พบหมายเลขประจำตัวนี้

4. ในกรณีที่ผู้ใช้ต้องการลบรายชื่อผู้ใช้อื่นที่อยู่ในรายการการติดต่อของผู้ใช้ โดยการเลือกรายชื่อที่ต้องการลบ ระบบทำงานฝั่งไคลเอนต์จะส่งข้อมูลเพื่อแจ้งการลบให้เซิร์ฟเวอร์ทราบ โดยมีรูปแบบโพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งไคลเอนต์

Status	101
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการลบ
Destination ID	หมายเลขประจำตัวผู้ใช้อื่นที่ผู้ใช้ต้องการลบ
Message	Null

ระบบการทำงานฝั่งเซิร์ฟเวอร์เมื่อได้รับข้อมูลก็จะทำการลบข้อมูลออกจากฐานข้อมูลรายการ การติดต่อ ตามหมายเลขประจำตัวผู้ใช้ระบบแล้วแจ้งกลับไปไคลเอนต์ทราบตามรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	101
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการลบ
Message	หมายเลขประจำตัวผู้ใช้คนอื่นที่ถูกลบ

หลังจากที่แจ้งให้ไคลเอนต์ทราบการลบ โดยส่งข้อมูลไปยังไคลเอนต์ ซึ่งไคลเอนต์จะใช้ข้อมูล ดังกล่าวเพื่อปรับเปลี่ยนการทำงานของหน้าจอรายการการติดต่อของผู้ใช้แล้ว เซิร์ฟเวอร์ จะแจ้งให้ผู้ใช้ที่ถูกลบทราบผ่านทางฐานข้อมูล โดยเซิร์ฟเวอร์จะจัดเก็บข้อมูลลงฐานข้อมูลที่ ใช้ในการเก็บข้อความเพื่อใช้ในการส่งต่อผู้ใช้โดยมีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	203
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการลบ
Destination ID	หมายเลขประจำตัวผู้ใช้คนอื่นที่ผู้ใช้ต้องการลบ
Message	Null

ในกรณีที่เกิดความผิดพลาดระบบการทำงานฝั่งเซิร์ฟเวอร์ไม่สามารถทำการลบได้ เซิร์ฟเวอร์ จะแจ้งให้ไคลเอนต์ทราบโดยมีรูปแบบ โพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	504
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการเพิ่ม
Message	ข้อความแสดงความผิดพลาดในที่นี้คือไม่สามารถลบข้อมูลหมายเลขประจำตัวนี้ได้

5. ในกรณีที่ผู้ใช้ต้องการทราบรายละเอียดข้อมูลส่วนตัวของผู้ใช้ที่อยู่ในรายการการติดต่อของผู้ใช้โดยการเลือกที่รายชื่อและคลิกที่ปุ่ม Details ระบบการทำงานฝั่งไคลเอนต์จะส่งข้อมูลเพื่อร้องขอไปที่เซิร์ฟเวอร์โดยมีรูปแบบโปรโตคอลดังต่อไปนี้

ข้อมูลฝั่งไคลเอนต์

Status	120
Source ID	หมายเลขประจำตัวผู้ใช้ระบบ
Destination ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Message	หมายเลขประจำตัวผู้ใช้อื่นที่ต้องการทราบรายละเอียด

หลังจากที่เซิร์ฟเวอร์ได้รับข้อมูลการร้องขอรายละเอียดเกี่ยวกับตัวผู้ใช้อื่นที่ผู้ใช้ต้องการทราบ เซิร์ฟเวอร์จะทำการอ่านข้อมูลจากฐานข้อมูลผู้ใช้ระบบและส่งข้อมูลกลับไปให้ไคลเอนต์ตามรูปแบบโปรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	120
Source ID	0 (ศูนย์ หมายถึง เซิร์ฟเวอร์)
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบ
Message	<ol style="list-style-type: none"> 1. หมายเลขประจำตัวผู้ใช้ที่ต้องการทราบรายละเอียด 2. ชื่อ 3. นามสกุล 4. ชื่อเล่น 5. ชื่อที่ใช้ในการสื่อสาร 6. เพศ 7. วัน/เดือน/ปี เกิด 8. E-mail Address <p>หมายเหตุ แต่ละส่วนของ message จะขึ้นด้วยเครื่องหมาย เพื่อเป็นการแยกข้อมูล</p>

6. ในกรณีที่ผู้ใช้ส่งข้อความระหว่างกัน โดยการพิมพ์ข้อความที่หน้าจอส่งข้อความ ระบบการทำงานฝั่งไคลเอนต์จะส่งข้อมูลไปยังฝั่งเซิร์ฟเวอร์ โดยมีรูปแบบโพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งไคลเอนต์

Status	250
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการส่งข้อความ
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่รับข้อความ
Message	ข้อความที่ต้องการส่ง

เมื่อเซิร์ฟเวอร์ได้รับข้อมูลที่ไคลเอนต์ส่งมาเซิร์ฟเวอร์จะจัดเก็บข้อมูลดังกล่าวลงฐานข้อมูลที่ใช้ในการเก็บข้อความ คอยที่เซิร์ฟเวอร์ที่ทำงานเกี่ยวข้องกับผู้ใช้ที่เป็นผู้รับข้อมูลจะส่งข้อมูลไปให้ฝั่งไคลเอนต์ของผู้รับรายนั้น (ระบบมีการแตกการทำงานเป็น Process อีกระยะเพื่อใช้งานกับผู้ใช้แต่ละคน) โดยมีรูปแบบโพรโตคอลดังต่อไปนี้

ข้อมูลฝั่งเซิร์ฟเวอร์

Status	250
Source ID	หมายเลขประจำตัวผู้ใช้ระบบที่ทำการส่งข้อความ
Destination ID	หมายเลขประจำตัวผู้ใช้ระบบที่รับข้อความ
Message	ข้อความที่ต้องการส่ง

Status	ความหมาย	Source ID	Destination ID	Message
000	ตอบรับการติดต่อขอเริ่มต้นการทำงานจากระบบจากโปรแกรมฝั่งไคลเอน	ศูนย์ (หมายถึงเซิร์ฟเวอร์)	หมายเลขของผู้ใช้	ชื่อของผู้ใช้, หมายเลขประจำตัวและชื่อที่ใช้ในการสื่อสารของผู้ใช้อื่นที่เกี่ยวข้องกับผู้ใช้ในการติดต่อสื่อสาร
100	ทำการเพิ่มข้อมูลผู้ใช้อื่นเข้าสู่รายการติดต่อของผู้ใช้ได้ในฐานะข้อมูลสำเร็จ	ศูนย์ (หมายถึงเซิร์ฟเวอร์)	หมายเลขของผู้ใช้	หมายเลขประจำตัวและชื่อผู้ใช้อื่นที่ได้รับการเพิ่มในรายการติดต่อของผู้ใช้
101	ทำการลบข้อมูลผู้ใช้อื่นในรายการติดต่อผู้ใช้ในฐานะข้อมูลสำเร็จ	ศูนย์ (หมายถึงเซิร์ฟเวอร์)	หมายเลขของผู้ใช้	หมายเลขของผู้ใช้ที่ถูกลบจากรายการติดต่อของผู้ใช้
200	แจ้งให้ผู้ใช้ทราบว่าผู้ใช้ที่เกี่ยวข้องในรายการติดต่อของผู้ใช้เข้าสู่สถานะ online	หมายเลขประจำตัวผู้ใช้ระบบที่เข้าสู่สถานะ online	หมายเลขผู้ใช้ที่เกี่ยวข้องกับรายการติดต่อของผู้ใช้ที่เข้าสู่สถานะ online	ชื่อของผู้ใช้ที่เกี่ยวข้องกับรายการติดต่อของผู้ใช้ที่เข้าสู่สถานะ online
201	แจ้งให้ผู้ใช้ทราบว่าผู้ใช้ที่เกี่ยวข้องในรายการติดต่อของผู้ใช้ออกจากระบบ	หมายเลขประจำตัวผู้ใช้ระบบที่ออกจากระบบ	หมายเลขผู้ใช้ที่เกี่ยวข้องกับรายการติดต่อของผู้ใช้ที่ออกจากระบบ	ชื่อของผู้ใช้ที่เกี่ยวข้องกับรายการติดต่อของผู้ใช้ที่ออกจากระบบ

ตารางที่ 3.1 แสดงความหมายของแต่ละส่วนของโปรโตคอลฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Status	ความหมาย	Source ID	Destination ID	Message
202	แจ้งผู้ใช้ระบบว่ามี การเพิ่มชื่อของผู้ใช้ ในรายการติดต่อ ของผู้ใช้อีกรายหนึ่ง	หมายเลขของผู้ใช้ที่ ทำการเพิ่มข้อมูล	หมายเลขของผู้ใช้ที่ ถูกเพิ่มเข้าในราย การติดต่อของผู้ใช้ ที่ทำการเพิ่มข้อมูล	ชื่อของผู้ใช้ที่ทำการ เพิ่มข้อมูล
203	แจ้งผู้ใช้ว่าถูกลบ ออกจากรายการติด ต่อของผู้ใช้รายอื่น	หมายเลขของผู้ใช้ที่ ทำการลบ	หมายเลขของผู้ใช้ที่ ถูกลบ	-
250	ข้อมูลที่ส่งระหว่าง ผู้ใช้	หมายเลขผู้ใช้ที่ส่ง ข้อมูล	หมายเลขผู้ใช้ที่รับ ข้อมูล	ข้อความ
501	แจ้งให้ผู้ใช้ทราบว่า ไม่พบหมายเลขผู้ ใช้	ศูนย์ (หมายถึง เซิร์ฟเวอร์)	หมายเลขที่ทำการ เพิ่มข้อมูล	ข้อความแสดง ความผิดพลาด
502	แจ้งให้ผู้ใช้ทราบว่า หมายเลขที่ผู้ใช้ทำ การเพิ่ม ไม่ได้อยู่ใน สถานะ online	ศูนย์ (หมายถึง เซิร์ฟเวอร์)	หมายเลขที่ทำการ เพิ่มข้อมูล	ข้อความแสดง ความผิดพลาด
503	แจ้งให้ผู้ใช้ทราบว่า ไม่สามารถเพิ่มข้อ มูลในฐานข้อมูลได้	ศูนย์ (หมายถึง เซิร์ฟเวอร์)	หมายเลขที่ทำการ เพิ่มข้อมูล	ข้อความแสดง ความผิดพลาด
504	แจ้งให้ผู้ใช้ทราบว่า ไม่สามารถลบข้อ มูลในฐานข้อมูลได้	ศูนย์ (หมายถึง เซิร์ฟเวอร์)	หมายเลขที่ทำการ ลบข้อมูล	ข้อความแสดง ความผิดพลาด

ตารางที่ 3.1 แสดงความหมายของแต่ละส่วนของโปรโตคอลฝั่งเซิร์ฟเวอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Status	ความหมาย	Source ID	Destination ID	Message
000	แจ้งให้เซิร์ฟเวอร์ทราบถึงการเข้าระบบของไคลเอนต์	หมายเลขของผู้ใช้ที่ต้องการเข้าสู่ระบบ	ศูนย์ (หมายถึงเซิร์ฟเวอร์)	-
100	แจ้งให้เซิร์ฟเวอร์ทราบถึงการเพิ่มข้อมูลในรายการติดต่อ	หมายเลขของผู้ใช้ที่ทำการเพิ่มข้อมูล	หมายเลขของผู้ใช้ที่ถูกเพิ่มข้อมูลเข้ารายการติดต่อ	ชื่อที่ใช้ในการสื่อสารของผู้ใช้ที่ทำการเพิ่มข้อมูล
101	แจ้งให้เซิร์ฟเวอร์ทราบถึงการลบข้อมูลในรายการติดต่อ	หมายเลขของผู้ใช้ที่ทำการลบข้อมูล	หมายเลขของผู้ใช้ที่ถูกลบข้อมูลเข้ารายการติดต่อ	-
250	แจ้งให้เซิร์ฟเวอร์ทราบว่าเป็นการส่งข้อมูลระหว่างผู้ใช้	หมายเลขผู้ใช้ที่ส่งข้อมูล	หมายเลขผู้ใช้ที่รับข้อมูล	ข้อความ

ตารางที่ 3.2 แสดงความหมายของแต่ละส่วนของโปรโตคอลฝั่งไคลเอนต์

3.3 ตารางฐานข้อมูลระบบ

Field Name	Data Type	Length
contactid	int	10
name	varchar	40
surname	varchar	40
nickname	varchar	20
displayname	varchar	40
gender	char	1
birthdate	Date	-
password	varchar	20
email	varchar	70
status	char	1

ตารางที่ 3.3 ข้อมูลผู้ใช้งานระบบ (contact)

CONTACTID	หมายถึง หมายเลขประจำตัวผู้ใช้งานในระบบ
NAME	หมายถึง ชื่อของผู้ใช้ระบบ
SURNAME	หมายถึง นามสกุลของผู้ใช้ระบบ
NICKNAME	หมายถึง ชื่อเล่นของผู้ใช้ระบบ
DISPLAYNAME	หมายถึง ชื่อที่ใช้ในระบบ
GENDER	หมายถึง เพศของผู้ใช้ระบบ M หมายถึง ผู้ชาย F หมายถึง ผู้หญิง
BIRTHDATE	หมายถึง วัน/เดือน/ปี เกิดของผู้ใช้งานระบบ
PASSWORD	หมายถึง รหัสผ่านของผู้ใช้
STATUS	หมายถึง สถานการณ์ใช้งานระบบของผู้ใช้ 1 หมายถึง online 0 หมายถึง offline

Field	Type	Length
contactid1	Int	10
contactid2	Int	10

ตารางที่ 3.4 ข้อมูลรายการติดต่อของผู้ใช้งานระบบ (contactlist)

CONTACTID1 หมายถึง หมายเลขประจำตัวผู้ใช้ระบบ
 CONTACTID2 หมายถึง หมายเลขประจำตัวผู้ใช้ที่อยู่ในรายการติดต่อของ contactid1

Field	Type	Length
msgid	int	10
status	tinyint	3
source	int	10
destination	int	10
date	timestamp	14
message	varchar	255

ตารางที่ 3.5 ข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างผู้ใช้งานระบบ (message)

MSGID หมายถึง หมายเลขประจำข้อความหรือข้อมูล
 STATUS หมายถึง สถานะของคำสั่ง
 SOURCE หมายถึง หมายเลขประจำตัวของผู้ส่งข้อความหรือข้อมูล
 DESTINATION หมายถึง หมายเลขประจำตัวของผู้รับข้อความหรือข้อมูล
 DATE หมายถึง วันเวลาที่ส่งข้อความหรือข้อมูล
 MESSAGE หมายถึง ข้อความหรือข้อมูล

3.4 ขั้นตอนการพัฒนาระบบ

3.4.1 พัฒนาระบบ CGI เพื่อใช้ในการรับข้อมูลผู้ใช้ระบบ

ระบบปฏิบัติการ	Linux
ภาษาในการพัฒนา	PERL
เครื่องมือในการพัฒนา	EditPlus Text Editor v2.10

ประกอบด้วย 4 ส่วนคือ

Login.cgi	โปรแกรมที่ใช้ในการเข้าระบบของผู้ใช้ โดยโปรแกรมจะตรวจสอบข้อมูลการใช้งานกับฐานข้อมูล MySQL ซึ่งภายหลังจากที่ผู้ใช้เข้าระบบแล้ว โปรแกรมจะสร้างหน้าจอ HTML ที่มีการเชื่อมต่อกับ Java Applet (Client.jar) ซึ่งใช้ในการสร้างการเชื่อมระหว่างผู้ใช้ระบบและเซิร์ฟเวอร์ในการทำงานของระบบ
Subscribe.cgi	โปรแกรมที่ใช้ในการสมัครสมาชิก โดยโปรแกรมจะติดต่อกับฐานข้อมูล MySQL เพื่อจัดเก็บข้อมูลส่วนตัวของผู้ใช้ระบบ
unsubscribe.cgi	โปรแกรมที่ใช้ในการยกเลิกการเป็นสมาชิกของผู้ใช้ระบบ โดยโปรแกรมจะติดต่อเพื่อลบข้อมูลผู้ใช้ออกจากฐานข้อมูล MySQL
userprofile.cgi	โปรแกรมที่ใช้ในการแก้ไขข้อมูลส่วนตัวผู้ใช้ระบบ โดยโปรแกรมจะทำหน้าที่ติดต่อกับฐานข้อมูล MySQL เพื่อทำการเปลี่ยนแปลงข้อมูลส่วนตัวของผู้ใช้

3.4.2 พัฒนาระบบส่วนกลาง

การพัฒนาระบบส่วนกลางเพื่อให้เป็นระบบที่ทำงานฝั่งเซิร์ฟเวอร์เพื่อใช้ตอบรับการเชื่อมต่อ socket จากฝั่งไคลเอนต์ซึ่งใช้ในการสื่อสารและระบบจะมีหน้าที่การควบคุมระบบการทำงานหลักในการสื่อสารของผู้ใช้ระบบ

ระบบปฏิบัติการ Linux
 ภาษาในการพัฒนา PERL
 เครื่องมือในการพัฒนา EditPlus Text Editor v2.10

daemon.pl	เป็นโปรแกรมที่ใช้ในการเชื่อมต่อ socket จากเว็บเบราว์เซอร์ ซึ่งติดต่อกับ Java Applet (Client.jar) ในการส่งข้อมูลระหว่างผู้ใช้ระบบ และ เซิร์ฟเวอร์ซึ่งต้องมีการติดต่อกับฐานข้อมูล MySQL
-----------	---

3.4.3 พัฒนาระบบติดต่อผู้ใช้ผ่าน socket บนเว็บเบราว์เซอร์

การพัฒนาระบบส่วนนี้เพื่อใช้ในการเชื่อมต่อ socket จากผู้ใช้ระบบในฝั่งไคลเอนต์ (Java Applet) และ เซิร์ฟเวอร์ซึ่งใช้ในการติดต่อระหว่างผู้ใช้ระบบ

ระบบปฏิบัติการ Windows 2000
 ภาษาในการพัฒนา Java Applet
 เครื่องมือในการพัฒนา EditPlus Text Editor v2.10

Client.java	เป็นโปรแกรมที่ใช้ในการเชื่อมต่อ socket ใน Java Applet ซึ่งทำงานอยู่ในเว็บเบราว์เซอร์ ในการร้องขอการเชื่อมต่อไปยังเซิร์ฟเวอร์ ในการส่งข้อมูลระหว่างผู้ใช้ระบบในการใช้งานจริงจะนำ class ที่ได้จากการ compile Java Applet เก็บในไฟล์ Client.jar เพื่อใช้ในการติดต่อสื่อสารต่อไป
-------------	--

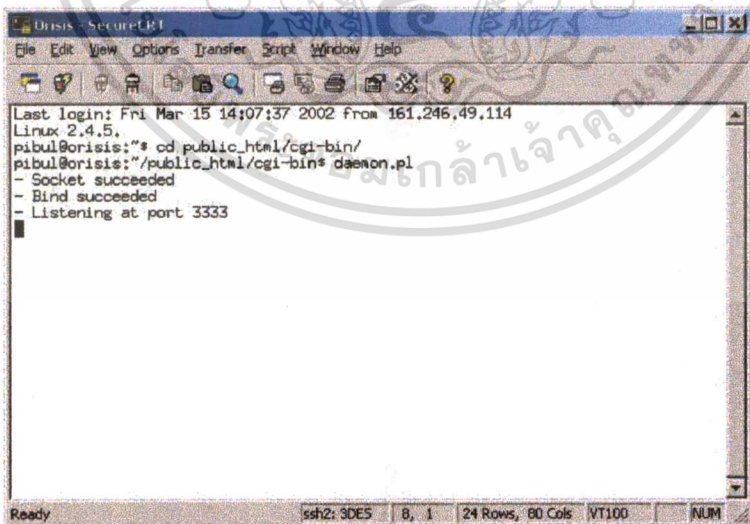
บทที่ 4

ผลการทดลอง

การทำงานของโปรแกรมระบบสื่อสารผ่านระบบเครือข่ายแบ่งออกเป็น 2 ส่วนการทำงานหลัก คือ

4.1 การทำงานฝั่งเซิร์ฟเวอร์

การทำงานของระบบสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตนั้นเริ่มต้นที่การทำงานของโปรแกรมจัดการข้อมูลที่อยู่ฝั่งเซิร์ฟเวอร์ (โปรแกรม daemon.cgi) ซึ่งเป็นโปรแกรมที่ทำหน้าที่ตรวจสอบและทำการส่งผ่านข้อมูลระหว่างผู้ใช้งานในระบบ (dispatcher program) ในการรันโปรแกรมหดดังกล่าวเพื่อให้ระบบทำงานได้นั้นต้องทำการรันโปรแกรมไว้ตลอดเวลา เพราะฉะนั้นจึงควรนำโปรแกรมไปรันไว้เป็น background process จึงควรนำไปกำหนดการทำงานไว้ที่ส่วนเริ่มต้นการทำงานของระบบ เช่นในระบบปฏิบัติการ Linux กำหนดไว้ที่ /etc/rc.d/rc.local เพื่อที่ในเวลาที่มีระบบมีการเปลี่ยนแปลงหรือหยุดลง แล้วต้องทำการเริ่มต้นการทำงานใหม่ก็จะทำให้ระบบการสื่อสารดำเนินการทำงานได้ต่อไป หรือการเรียกให้โปรแกรมทำงานจะเป็นดังรูปที่ 4.1

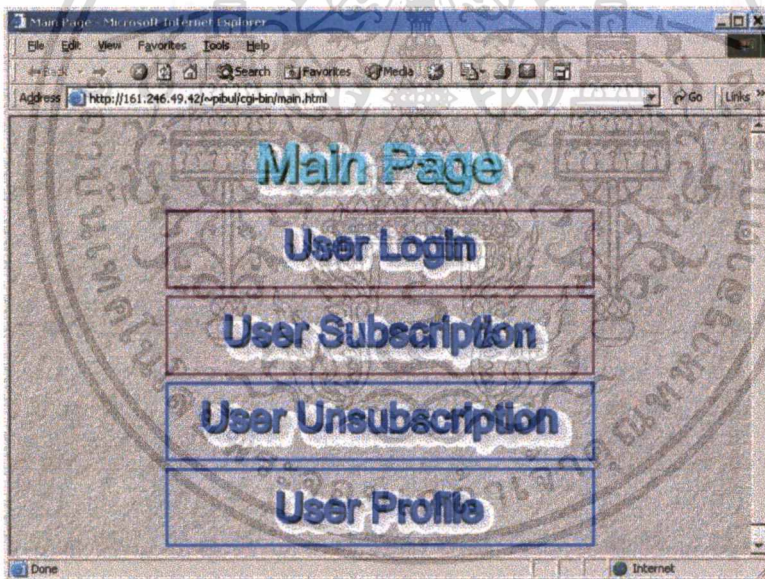


```
Orsis - SecureCRT
File Edit View Options Transfer Script Window Help
Last login: Fri Mar 15 14:07:37 2002 from 161.246.49.114
Linux 2.4.5.
pibul@osis:~$ cd public_html/cgi-bin/
pibul@osis:~/public_html/cgi-bin$ daemon.pl
- Socket succeeded
- Bind succeeded
- Listening at port 3333
Ready ssh2: 3DES 8, 1 24 Rows, 60 Cols VT100 NUM
```

รูปที่ 4.1 แสดงการทำงานของโปรแกรมทำงานส่วนกลาง daemon.pl

4.2 การทำงานของระบบในฝั่งไคลเอนต์

หลังจากที่ได้ทำการรันโปรแกรมส่วนการทำงานกลางเรียบร้อยแล้วการทำงานในฝั่งไคลเอนต์ ทำได้โดย ผู้ใช้เปิด web browser เพื่อใช้เข้าระบบโดยที่ผู้ใช้ไม่ต้องทำการติดตั้งโปรแกรมหรือเปลี่ยนแปลงใด ๆ ในระบบการทำงานในเครื่องคอมพิวเตอร์ของผู้ใช้แต่อย่างใด การเข้าใช้ระบบสื่อสารผ่านเครือข่ายกระทำได้โดยการเข้าเว็บไซต์ที่กำหนดไว้ โดยการพิมพ์ URL ที่ใช้เป็นเซิร์ฟเวอร์ในการทำงานของระบบโดยเรียกชื่อแฟ้มข้อมูล main.html ซึ่งเป็นโปรแกรมแสดงหน้าจอการทำงานหลัก (URL ในที่นี้ที่นี้ใช้ <http://161.246.49.42/~pibul/cgi-bin/main.pl> เพื่อทำการทดลองการทำงานชั่วคราว) หลังจากที่เราเรียกใช้งานโปรแกรมจะได้ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการเรียกใช้ระบบสื่อสารผ่านทางเว็บเบราว์เซอร์

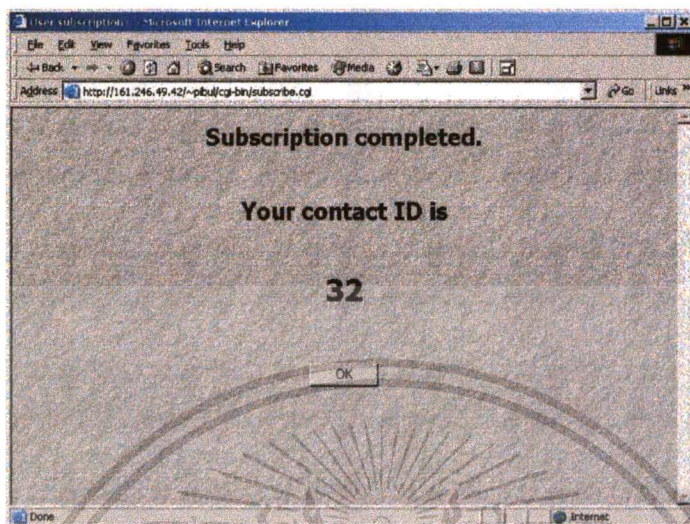
4.2.1 การสมัครสมาชิกเพื่อใช้งานระบบ

การสมัครสมาชิกทำได้หลังจากที่ผู้ใช้ได้เรียกหน้าจอหลักในการทำงานของระบบขึ้นมาทำงานแล้ว โดยเลือกที่หัวข้อ User Subscription เพื่อเข้าไปกรอกรายละเอียดเกี่ยวกับตัวผู้ที่ต้องการสมัครเป็นสมาชิก ดังรูปที่ 4.3

Name	ชื่อ
Surname	นามสกุล
Nick name	ชื่อเล่น
Display name	ชื่อที่แสดง
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Birthdate	14 April 1971
Password	
Password (re-type)	
E-mail Address	pibul@msn.com

รูปที่ 4.3 แสดงหน้าจอสมัครสมาชิก

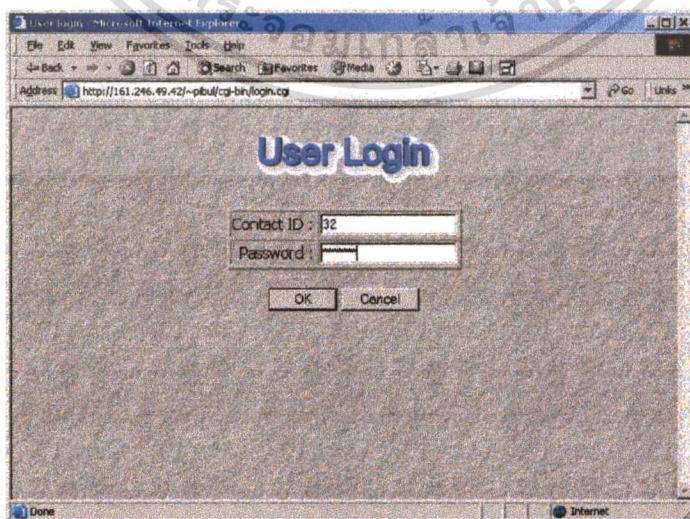
หลังจากที่ผู้ที่ต้องการสมัครเป็นสมาชิกทำการกรอกรายละเอียดทั้งหมดสมบูรณ์แล้วตอบตกลงโดยคลิกเมาส์ที่ปุ่ม OK เพื่อสมัครเป็นสมาชิกแล้วระบบจะตอบรับการสมัครโดยแสดงหน้าจอให้ผู้ต้องการใช้ระบบทราบถึงการสมัครว่าเสร็จเรียบร้อยแล้วพร้อมทั้งแสดงหมายเลขประจำตัวของผู้ใช้ระบบที่สามารถที่จะใช้ในการเข้าใช้ระบบสื่อสารผ่านระบบเครือข่าย และหมายเลขประจำตัวของใช้นี้จะใช้ในการอ้างถึงบุคคลหรือผู้ใช้อื่นที่อยู่ในระบบเพื่อใช้ในการติดต่อสื่อสารกันต่อไป ดังรูปที่ 4.4



รูปที่ 4.4 แสดงหน้าการสมัครเสร็จสมบูรณ์พร้อมทั้งหมายเลขประจำตัวผู้ใช้ เมื่อผู้ใช้ได้รับการตอบรับแล้วคลิกเมาส์ที่ปุ่ม OK ก็จะกลับมาที่หน้าจอหลักเพื่อทำงานต่อไป

4.2.2 การเข้าใช้งานระบบ

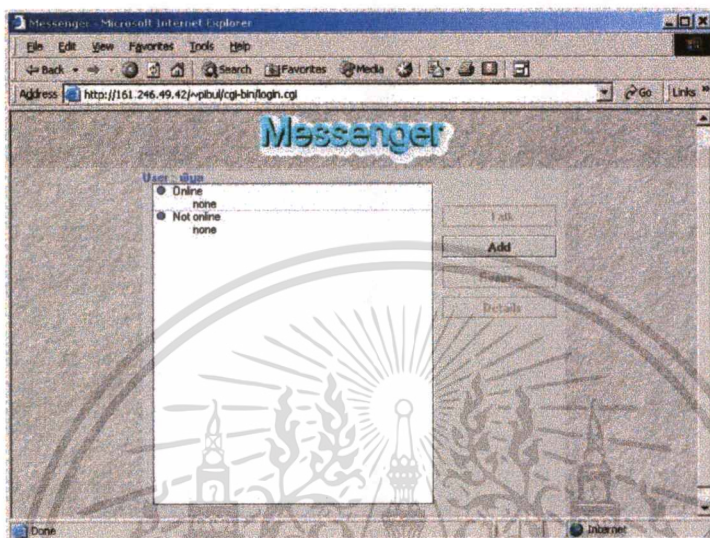
หลังจากที่ผู้ใช้ได้ทำการสมัครเป็นสมาชิกและได้รับหมายเลขประจำตัวแล้ว ผู้ใช้สามารถที่จะเข้าใช้งานระบบเพื่อติดต่อสื่อสารกับผู้อื่นผ่านระบบเครือข่ายอินเทอร์เน็ต ได้ โดยเลือกที่หัวข้อ User Login ในหน้าจอการทำงานหลัก หลังจากนั้นจะขึ้นหน้าจอให้ผู้ใช้ใส่หมายเลขประจำตัวผู้ใช้ และ รหัสผ่านดังรูปที่ 4.5



รูปที่ 4.5 แสดงหน้าจอเพื่อให้ผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

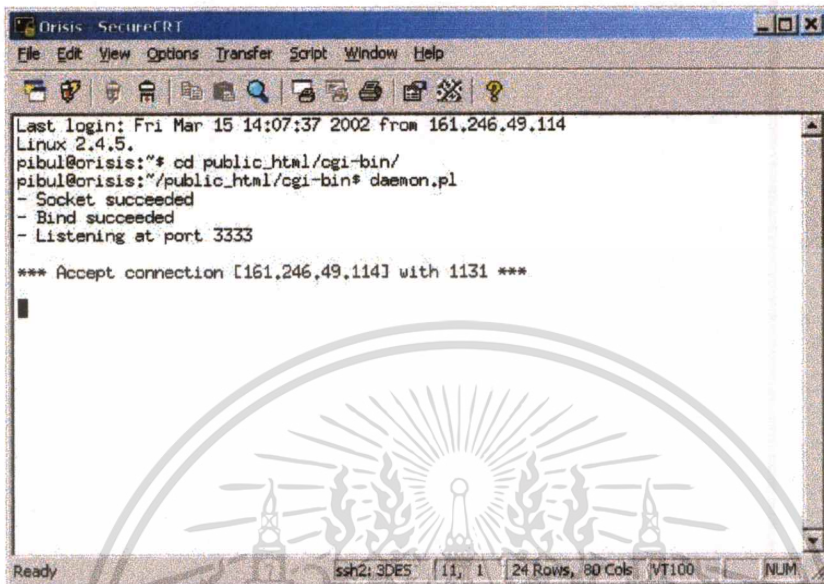
เมื่อผู้ใช้อกรอกหมายเลขและรหัสผ่านของผู้ใช้ได้ถูกต้องก็จะเข้าสู่หน้าจอที่ใช้ในการสื่อสาร ดังรูปที่ 4.6



รูปที่ 4.6 แสดงหน้าจอใช้ในการสื่อสาร

หน้าจอทางด้านซ้ายมือจะแสดงรายชื่อผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้ที่มีสถานะ online อยู่ด้านขวามือเป็นปุ่มคำสั่งในการทำงาน เช่นปุ่ม Talk ใช้เพื่อสื่อสารกับผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้โดยเลือกที่ชื่อที่ต้องการติดต่อนั้นคลิกเมาส์ที่ปุ่ม Talk เพื่อทำการสื่อสารกับผู้ใช้ที่ได้เลือกไว้, ปุ่ม Add ใช้ในการเพิ่มรายชื่อผู้ใช้อื่นที่ผู้ต้องการติดต่อเข้าไปในรายการติดต่อของผู้ใช้ทางซ้ายมือ, ปุ่ม Remove ใช้ในการลบรายชื่อผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้ทางซ้ายมือ เมื่อผู้ใช้ไม่ต้องการติดต่อสื่อสารกับผู้ใช้นั้นอีกต่อไป

หลังจากผู้ใช้อกรอกหมายเลขประจำตัวและรหัสผ่านในการเข้าสู่ระบบได้ถูกต้องและเข้าสู่หน้าจอที่ใช้ในการสื่อสารนั้นการทำงานของโปรแกรมส่วนกลางจะตอบรับการทำงานเพื่อที่จะเชื่อมโยงการติดต่อสื่อสารระหว่างระหว่างผู้ใช้งานระบบและระบบฐานข้อมูล และระหว่างผู้ใช้งานระบบและผู้ใช้อื่นที่ต้องการติดต่อสื่อสารกันต่อไป ดังรูปที่ 4.7



```

Orisis SecureCRT
File Edit View Options Transfer Script Window Help

Last login: Fri Mar 15 14:07:37 2002 from 161.246.49.114
Linux 2.4.5.
pibul@orisis:~* cd public_html/cgi-bin/
pibul@orisis:~/public_html/cgi-bin* daemon.pl
- Socket succeeded
- Bind succeeded
- Listening at port 3333

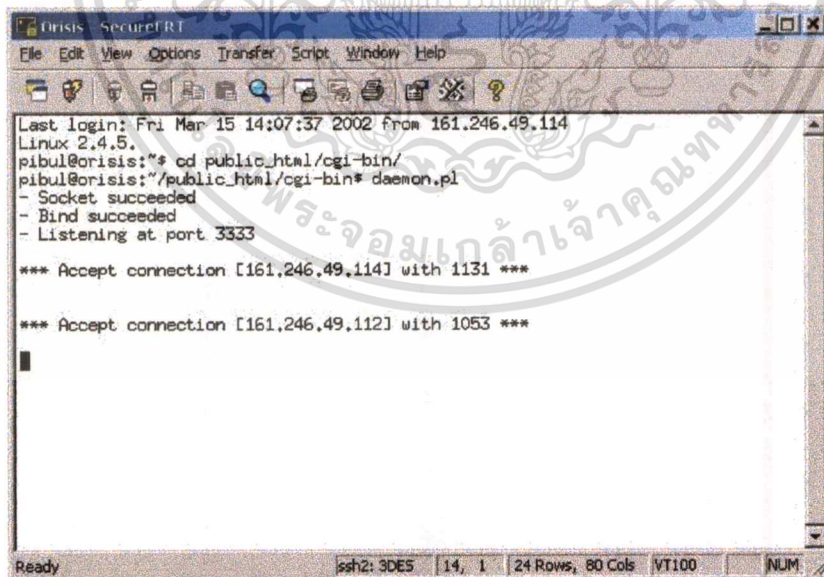
*** Accept connection [161,246,49,114] with 1131 ***

Ready ssh2: 3DES 11, 1 24 Rows, 80 Cols VT100 NUM

```

รูปที่ 4.7 แสดงการทำงานของโปรแกรมส่วนกลางในการตอบรับการทำงานของผู้ใช้

และเมื่อมีผู้รายอื่นเข้ามาในระบบ โปรแกรมส่วนกลางก็จะตอบรับเช่นเดียวกันดังรูปที่ 4.8



```

Orisis SecureCRT
File Edit View Options Transfer Script Window Help

Last login: Fri Mar 15 14:07:37 2002 from 161.246.49.114
Linux 2.4.5.
pibul@orisis:~* cd public_html/cgi-bin/
pibul@orisis:~/public_html/cgi-bin* daemon.pl
- Socket succeeded
- Bind succeeded
- Listening at port 3333

*** Accept connection [161,246,49,114] with 1131 ***

*** Accept connection [161,246,49,112] with 1053 ***

Ready ssh2: 3DES 14, 1 24 Rows, 80 Cols VT100 NUM

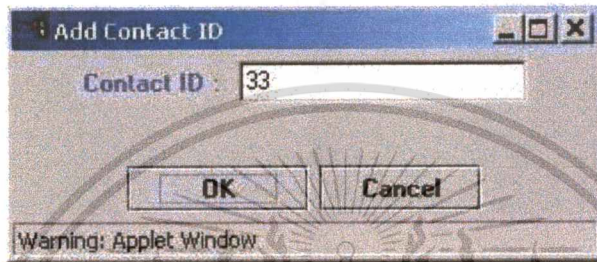
```

รูปที่ 4.8 แสดงการตอบรับผู้รายที่สองที่เข้าใช้ระบบของโปรแกรมส่วนกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

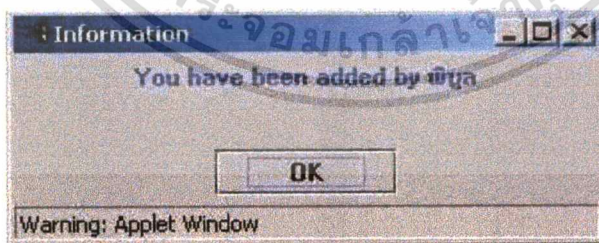
4.2.2.1 การเพิ่มรายชื่อผู้ใช้อื่นเข้าไปในรายการติดต่อของผู้ใช้

ในกรณีที่ผู้ใช้ต้องการติดต่อสื่อสารกับผู้ใ้รายอื่นในระบบ ผู้ใช้สามารถทำได้โดยคลิกเมาส์ที่ปุ่ม Add ในส่วนด้านขวามือ จะปรากฏหน้าจอเพื่อให้ผู้ใช้กรอกหมายเลขประจำตัวของผู้ใ้รายอื่นที่ผู้ใช้ต้องการติดต่อสื่อสารด้วย ดังรูปที่ 4.9 เมื่อระบบทำงานเสร็จจะเพิ่มรายชื่อผู้ใ้อื่นที่ผู้ใช้ต้องการติดต่อเข้าสู่รายการผู้ใ้ที่ต้องการติดต่อทางด้านซ้ายมือ



รูปที่ 4.9 แสดงหน้าจอกำหนดหมายเลขประจำตัวผู้ใ้อื่นเพื่อเพิ่มในรายการติดต่อของผู้ใ้

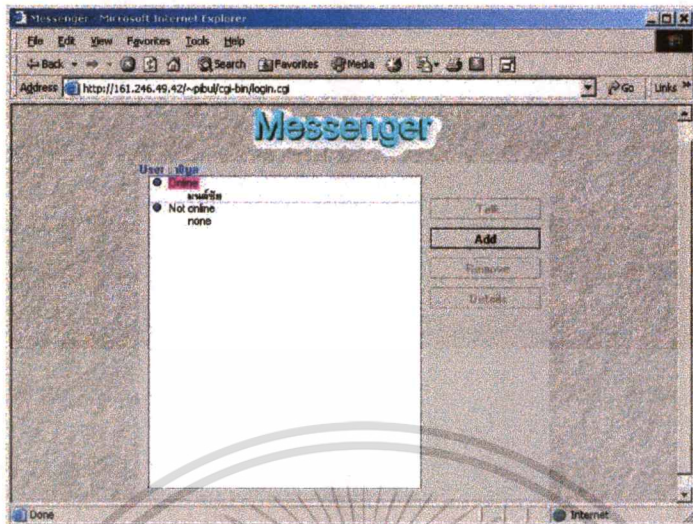
ภายหลังจากที่ทำการเพิ่มรายชื่อผู้ใ้รายอื่นที่ต้องการติดต่อในรายการติดต่อทางซ้ายมือสำเร็จก็จะมีชื่อที่ใ้ในการสื่อสารของผู้ใ้รายนั้นแสดงในรายการติดต่อของผู้ใ้ทางด้านซ้ายมือ พร้อมทั้งมีหน้าจอแสดงให้ผู้ใ้อื่นที่ถูกเพิ่มเข้าในรายการของผู้ใ้ทราบว่าผู้ใ้ได้เพิ่มรายชื่อเพื่อต้องการติดต่อกับผู้ใ้รายนั้นในกรณีที่ผู้ใ้รายนั้นมีสถานะ online ดังแสดงในรูปที่ 4.10



รูปที่ 4.10 แสดงหน้าจอเพื่อให้ผู้ใ้รายอื่นทราบว่าได้รับการเพิ่มชื่อเพื่อทำการติดต่อ

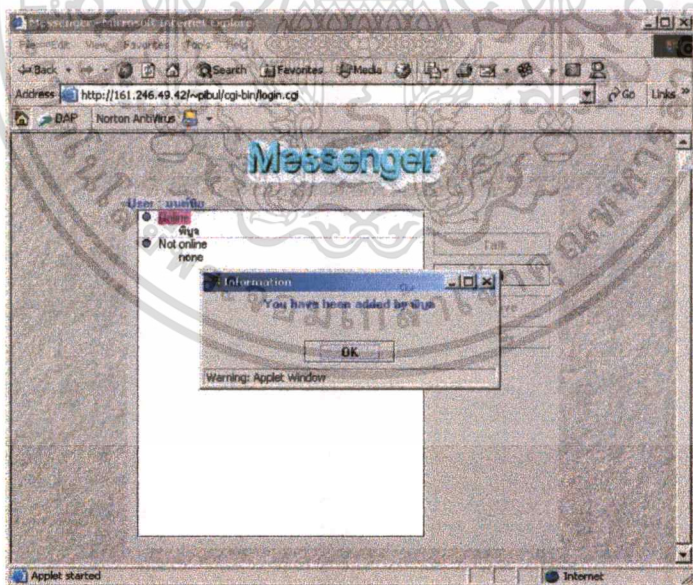
ภายหลังจากที่ผู้ใ้ (ในที่นี้ใ้ชื่อชิบูล) ทำการเพิ่มรายชื่อผู้ใ้รายอื่น (ในที่นี้ใ้ชื่อมนต์ชัย) ในการติดต่อเข้ากับรายการติดต่อของผู้ใ้ได้สำเร็จก็จะแสดงรายชื่อของผู้ใ้ที่ได้ทำการเพิ่มเข้ามานั้นในรายการทางด้านซ้ายมือ ดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงการเพิ่มรายชื่อผู้ใ้รายอื่นในระบบเข้ารายการติดต่อของผู้ใ้ได้สำเร็จ

เป็นการเพิ่มรายชื่อของผู้ใ้ (ในที่นี้ใ้ชื่อพิบูล) ในรายการติดต่อของผู้ใ้ที่ต้องการเพิ่ม (ในที่นี้ใ้ชื่อมนต์ชัย) โดยอัตโนมัติ ดังรูปที่ 4.12

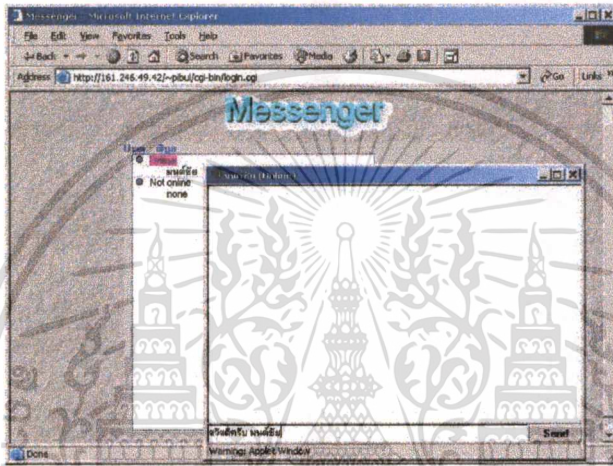


รูปที่ 4.12 แสดงหน้าจอการสื่อสารของผู้ใ้ที่ได้รับการเพิ่มรายชื่อในรายการติดต่อของผู้ใ้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

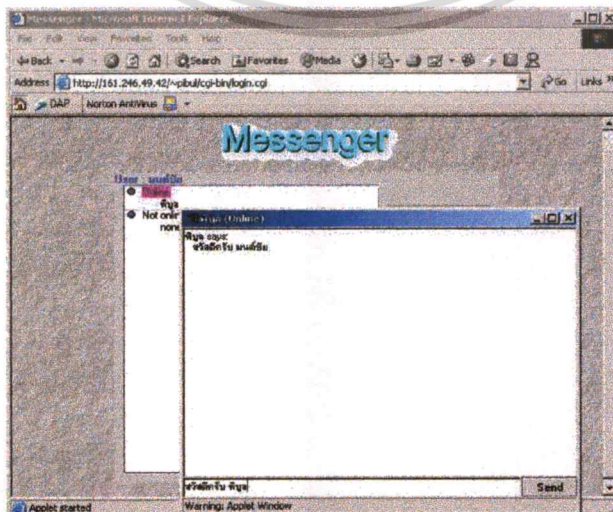
4.2.2.2 การส่งข้อความเพื่อสื่อสารกับผู้อื่น

ภายหลังจากที่ผู้ใช้ได้ทำการเพิ่มรายชื่อผู้ใช้รายอื่นเข้าไปในรายการติดต่อทางด้านซ้ายมือได้เป็นผลสำเร็จแล้ว ผู้ใช้สามารถที่จะส่งข้อความเพื่อสื่อสารกับผู้ใช้รายนั้นได้ โดยคลิกเมาส์เลือกที่ชื่อผู้ใช้ที่ต้องการติดต่อในรายการติดต่อทางด้านซ้ายมือแล้วคลิกเมาส์ที่ปุ่ม Talk หรือดับเบิลคลิกเมาส์ที่ชื่อที่ต้องการติดต่อ ซึ่งจะแสดงหน้าจอที่ใช้ในการสนทนาระหว่างผู้ใช้กับผู้ใช้ที่ต้องการติดต่อ ดังรูปที่ 4.13



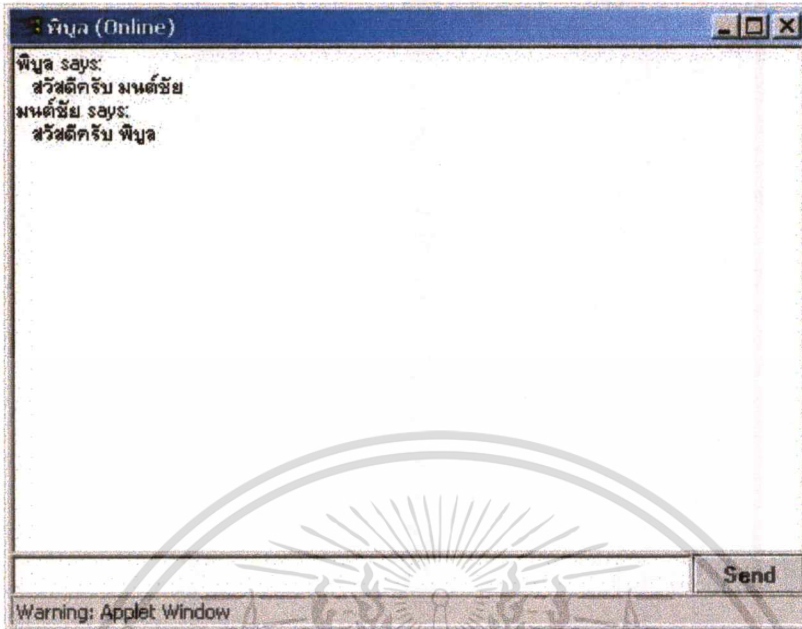
รูปที่ 4.13 แสดงหน้าจอที่ใช้ในการส่งข้อความระหว่างผู้ใช้และผู้ใช้รายอื่นที่ต้องการติดต่อ

ผู้ใช้สามารถที่จะพิมพ์ข้อความในช่องข้อความด้านล่างแล้วกดปุ่ม Enter หรือกดที่ปุ่ม Send เพื่อทำการส่งข้อความไปยังผู้ใช้รายอื่นที่ต้องการติดต่อได้ หลังจาก que ผู้ใช้ส่งข้อความไปยังผู้ใช้ที่ต้องการติดต่อ หน้าจอเพื่อใช้ในการสื่อสารของผู้ใช้ที่ต้องการติดต่อก็จะแสดงข้อความของทั้งผู้ใช้และผู้ใช้อื่นที่ติดต่อกับผู้ใช้ ดังแสดงในรูปที่ 4.14, รูปที่ 4.15 และ รูปที่ 4.16

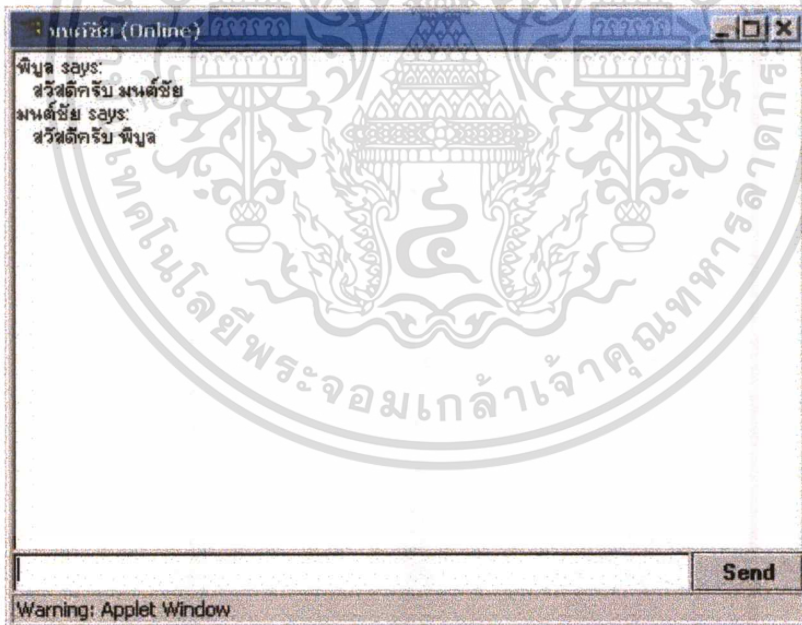


รูปที่ 4.14 แสดงหน้าจอสื่อสารของผู้ใช้ที่ผู้ใช้ติดต่อกับผู้ใช้รายอื่น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงหน้าจอสื่อสารข้อมูลที่สื่อสารฝั่งผู้อื่น (มนต์ชัย)

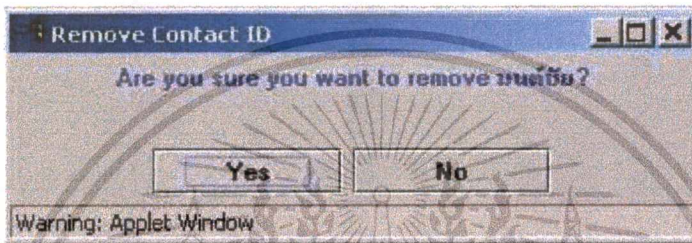


รูปที่ 4.16 แสดงหน้าจอสื่อสารฝั่งผู้ใช้ (พินุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 การลบรายชื่อผู้อื่นในรายการติดต่อของผู้ใช้

ผู้ใช้สามารถที่จะลบรายชื่อของผู้ใช้อื่นที่ไม่ต้องการติดต่อสื่อสารอีกต่อไปจากรายการติดต่อของผู้ใช้ทางด้านซ้ายมือได้ โดยการคลิกเมาส์เลือกที่รายชื่อที่ต้องการลบแล้วคลิกเมาส์ที่ปุ่ม Remove เพื่อทำการลบ ซึ่งจะมีหน้าจอแสดงเพื่อถามความแน่ใจของผู้ใช้ว่าต้องการลบออกจากรายการติดต่อของผู้ใช้หรือไม่เพื่อไม่ให้เกิดความผิดพลาดในการลบถ้าผู้ใช้ต้องการลบจริงๆ ทำได้โดยคลิกเมาส์เลือกที่ปุ่ม Yes ถ้าผู้ใช้ไม่ต้องการลบให้คลิกที่ปุ่ม No ดังรูปที่ 4.17



รูปที่ 4.17 แสดงข้อความเตือนการลบรายชื่อออกจากรายการติดต่อของผู้ใช้

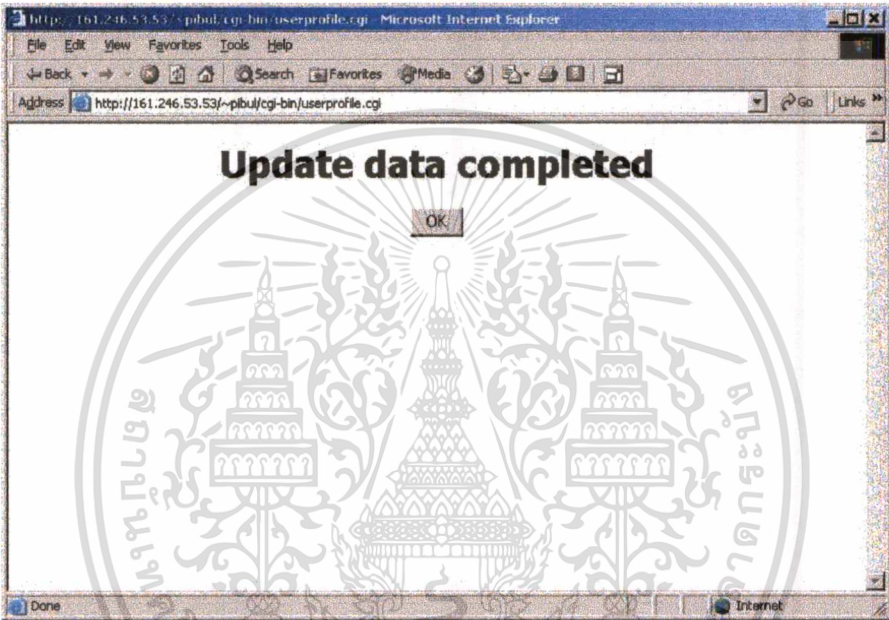
4.2.3 การแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ

ผู้ใช้สามารถที่จะแก้ไขข้อมูลส่วนตัวของผู้ใช้ได้โดยเลือกคำสั่ง User Profile ที่หน้าจอหลักซึ่งผู้ใช้ต้องทำการกรอกหมายเลขประจำตัวของผู้ใช้และรหัสผ่านเพื่อยืนยันว่าเป็นผู้ใช้ตัวจริงดังรูปที่ 4.18

รูปที่ 4.18 แสดงหน้าจอข้อมูลส่วนตัวของผู้ใช้ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

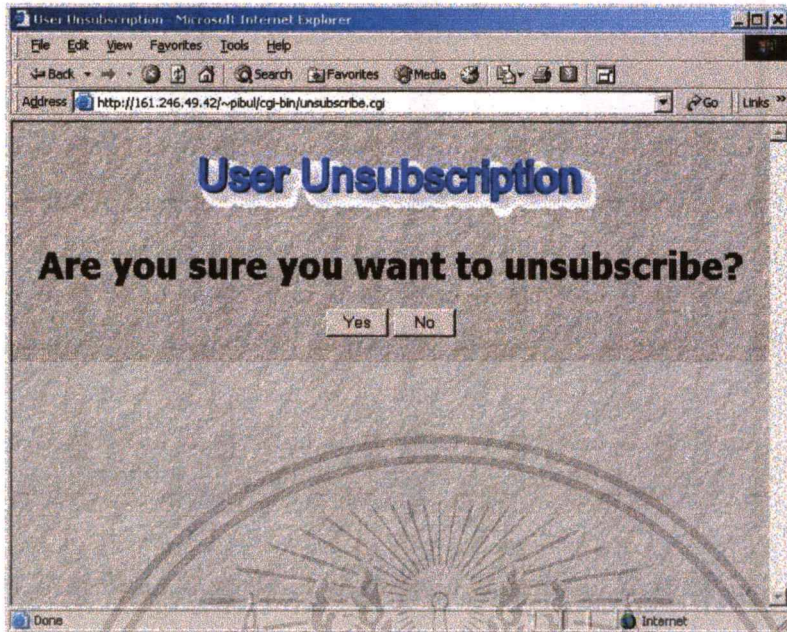
ภายหลังจากที่ผู้ใช้แก้ไขเปลี่ยนแปลงข้อมูลส่วนตัวเรียบร้อยแล้วทำการคลิกเมาส์ที่ปุ่ม Change เพื่อจัดเก็บข้อมูลที่ผู้ใช้ทำการเปลี่ยนแปลง และเมื่อระบบทำการจัดเก็บข้อมูลที่ผู้ใช้ทำการเปลี่ยนแปลงเรียบร้อยแล้ว ระบบจะแสดงหน้าจอให้ผู้ใช้ทราบว่าระบบได้ทำการเปลี่ยนแปลงข้อมูลเรียบร้อยแล้วหรือเกิดข้อผิดพลาดในการจัดเก็บ ดังรูปที่ 4.19 แสดงหน้าจอที่แสดงว่าหน้าการจัดเก็บข้อมูลที่ผู้ใช้ทำการแก้ไขเสร็จสิ้นสมบูรณ์แล้ว



รูปที่ 4.19 แสดงหน้าจอการแสดงผลการจัดเก็บข้อมูลที่ผู้ใช้ทำการเปลี่ยนแปลงสำเร็จ

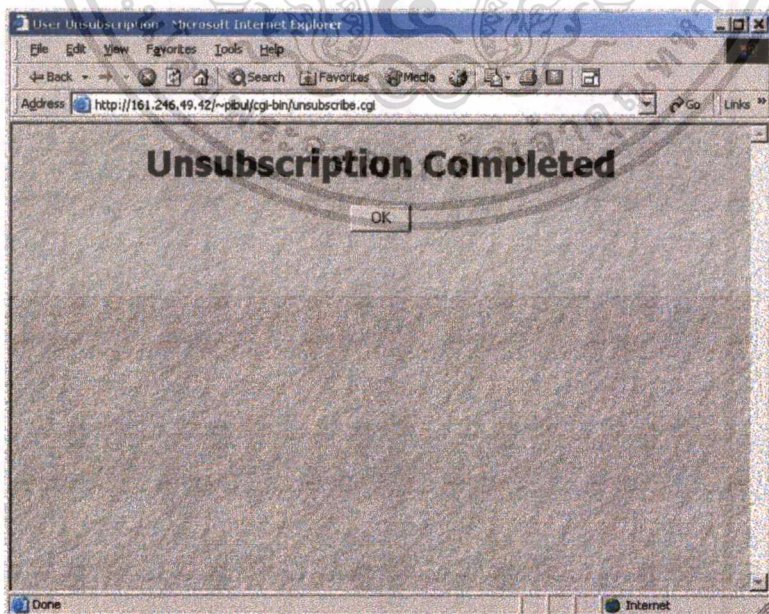
4.2.4 การยกเลิกการเป็นสมาชิกระบบ

ในกรณีที่ผู้ใช้ต้องการยกเลิกการเป็นสมาชิกระบบ อันเนื่องด้วยผู้ใช้ไม่ต้องการใช้งานระบบอีกต่อไป ผู้ใช้ก็สามารถที่จะทำได้โดยเลือกที่คำสั่ง User Unsubscription ที่หน้าจอหลัก ซึ่งผู้ใช้ต้องทำการกรอกหมายเลขประจำตัวของผู้ใช้และรหัสผ่านเพื่อยืนยันยังว่าเป็นผู้ใช้ตัวจริงที่ต้องการยกเลิกการเป็นสมาชิกของระบบ หลังจากผู้ใช้กรอกหมายเลขประจำตัวและรหัสผ่านได้ถูกต้องระบบจะแสดงหน้าจอเพื่อที่จะแสดงการยืนยันการยกเลิกการเป็นสมาชิกจากผู้ใช้ ถ้าผู้ใช้ต้องการยกเลิกการเป็นสมาชิกให้คลิกเมาส์ที่ปุ่ม Yes ถ้าผู้ใช้ไม่ต้องการยกเลิกให้คลิกเมาส์ที่ปุ่ม No ดังแสดงในรูปที่ 4.20



รูปที่ 4.20 แสดงหน้าจอขี้นยืนยันการยกเลิกการเป็นสมาชิกของระบบ

หลังจากที่ผู้ใช้ตอบตกลงที่จะยกเลิกการเป็นสมาชิกของระบบ ระบบจะแสดงหน้าจอให้ผู้ใช้ระบบทราบว่าได้ตอบรับการยกเลิกการเป็นสมาชิกของผู้ใช้เรียบร้อยแล้ว ดังแสดงในรูปที่ 4.21



รูปที่ 4.21 แสดงหน้าจอตอบรับการยกเลิกการเป็นสมาชิกของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการพัฒนา

5.1 ความแตกต่างระหว่างระบบที่ได้พัฒนาขึ้นและระบบที่มีอยู่ในปัจจุบัน

ระบบที่มีอยู่ในปัจจุบัน	ระบบ Web-based Messenger
ผู้ใช้ต้องทำการติดตั้งระบบในการใช้งาน	ผู้ไม่ต้องการติดตั้งระบบในการใช้งาน
ระบบต้องทำการติดตั้งเฉพาะ Platform ในการทำงาน	การทำงานของระบบไม่ขึ้นอยู่กับ Platform ในการใช้งานเพราะทำงานผ่านระบบ browser
ระบบมีการทำงานจาก Executable file ทำให้สามารถติดไวรัสได้	ไม่สามารถติดไวรัสได้เพราะระบบไม่สามารถติดต่อบนระบบไฟล์ที่อยู่ในเครื่องของผู้ใช้ได้
ระบบสามารถเปิดช่องทางการสื่อสารทำให้เสี่ยงต่อการติดไวรัสบางชนิดได้	ด้วยข้อจำกัดของ Java Applet ทำให้ระบบไม่สามารถเปิดช่องทางการสื่อสารเพื่อเป็นช่องทางติดไวรัสได้
ใช้ได้เฉพาะเครื่องที่ติดตั้ง โปรแกรมสื่อสารเท่านั้น	ใช้ได้ทุกเครื่องที่มีการติดต่อบนอินเทอร์เน็ตและมีเบราว์เซอร์ในการใช้งาน
ผู้ใช้ต้องทำการ Upgrade ระบบเอง ในกรณีที่ระบบมีการเปลี่ยนแปลง	เนื่องจากระบบทำงานจากส่วนกลางหรือเซิร์ฟเวอร์ผู้ใช้จึงไม่จำเป็นต้อง Upgrade ระบบเองในกรณีที่ระบบมีการเปลี่ยนแปลง

ตารางที่ 5.1 แสดงความแตกต่างระหว่างระบบที่ได้พัฒนาขึ้นและระบบที่มีอยู่ในปัจจุบัน

5.2 ประโยชน์ที่ได้รับจากการทำงานของระบบ

- ทำให้ผู้ใช้ระบบสามารถสื่อสารถึงกันได้ภายในระบบเครือข่ายอินเทอร์เน็ต
- ทำให้ผู้ใช้ระบบสามารถใช้งานได้ในทุก Platform เพราะระบบทำงานผ่านเว็บเบราว์เซอร์ และ Java Applet ซึ่งจากข้อดีของ Java Applet ที่สามารถใช้งานได้ในทุก Platform ทำให้ผู้ใช้ระบบที่ทำงานบนเครื่องต่าง Platform สามารถใช้งานระบบได้เช่นเดียวกัน
- ลดความยุ่งยากในการติดตั้ง โปรแกรมสื่อสารอันเนื่องมาจากการติดตั้งโปรแกรมที่ใช้ในการสื่อสาร
- ผู้ใช้สามารถมั่นใจได้ว่าการใช้งานระบบทำได้อย่างปลอดภัยไม่มีไวรัสที่สามารถที่จะแพร่ไปยังระบบของผู้ใช้ได้ อันเนื่องมาจากข้อจำกัดของเว็บเบราว์เซอร์และการทำงานของ Java Applet ที่ไม่สามารถติดต่อแก้ไขระบบเพิ่มข้อมูลภายในระบบของผู้ใช้ได้ และ ไม่สามารถเปิดการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ในระบบเครือข่ายได้นอกจากเชื่อมต่อกับเซิร์ฟเวอร์ที่เป็นต้นทางในการเรียกใช้ Java Applet นั้น ทำให้ไวรัสไม่สามารถติดต่อได้
- ลดการสิ้นเปลืองเนื้อที่จัดเก็บข้อมูลในระบบของผู้ใช้ เพราะผู้ใช้ระบบไม่จำเป็นต้องติดตั้งระบบการใช้งานภายในเครื่องของผู้ใช้ระบบ
- เป็นแนวทางในการศึกษาระบบที่มีการใช้การเชื่อมโยงข้อมูลระหว่างเครื่องคอมพิวเตอร์ในเครือข่ายโดยใช้ CGI และ Socket อีกทั้งทำให้ทราบถึงปัญหาต่าง ๆ ทำให้สามารถปรับเปลี่ยนการทำงานเพื่อพัฒนาระบบอื่นที่มีการใช้งานทางด้านนี้ได้ดีขึ้น

5.3 ปัญหาและข้อจำกัดของระบบ

- ระบบไม่สามารถที่จะทำการส่งแฟ้มข้อมูลจากผู้ใช้หนึ่งไปยังอีกผู้ใช้หนึ่งได้อันเนื่องมาจากข้อจำกัดของ Applet การอ่านข้อมูลที่เครื่องของผู้ใช้ระบบ
- ถ้ามีการใช้งานของผู้ใช้จำนวนมากอาจเกิดปัญหาคอขวดได้ เพราะเป็นการส่งข้อมูลผ่านเซิร์ฟเวอร์
- ระบบอาจมีการทำงานช้ากว่าระบบอื่นที่ต้องติดตั้งโปรแกรมในระยะการทำงานแรก ๆ เพราะระบบได้พัฒนาโดยใช้ Java Applet ซึ่งทำงานอยู่บนเบรดาเซอร์ จึงต้องทำการร้องขอข้อมูล Java Applet (download) จากเซิร์ฟเวอร์หลายไฟล์ แต่ระบบจะทำงานได้เร็วขึ้นในการทำงานครั้งต่อไปเพราะเว็บเบรดาเซอร์ได้จัดเก็บข้อมูลที่จำเป็นบางส่วนไว้ที่ cache ของเว็บเบรดาเซอร์ทำให้การทำงานครั้งต่อ ๆ ไปเร็วขึ้น
- ในบางหน่วยงานที่มีการห้ามการติดต่ออาจโดยใช้ Firewall ที่มีกำหนดห้ามการติดต่อผ่าน port การสื่อสารอาจทำให้ระบบไม่สามารถเชื่อมต่อกันได้

5.4 แนวทางแก้ไข

- ทำ Resource Replication โดยการเพิ่มจำนวนของเซิร์ฟเวอร์ในกรณีที่มีผู้ใช้งานจำนวนมากขึ้นเพื่อเป็นการกระจายการทำงานให้กับเซิร์ฟเวอร์หลาย ๆ เครื่องที่รองรับการทำงานพร้อม ๆ กัน (Load Balancing)
- ในการห้ามการสื่อสารผ่าน Port การสื่อสารซึ่งถือเป็นปัจจัยหลักในการเชื่อมโยงการทำงานระหว่างเครื่องคอมพิวเตอร์ในเครือข่ายผ่าน Socket นั้น อาจสามารถหลีกเลี่ยงการทำงานนี้ ไปใช้การส่งข้อมูลผ่านโพรโตคอล SOAP ซึ่งใช้ภาษา XML ในการส่งผ่านข้อมูลโดยใช้เว็บเบรดาเซอร์ได้

ภาคผนวก

1. วิธีการติดตั้งระบบ

โปรแกรมสื่อสารผ่านระบบเครือข่ายเป็นระบบที่มีการทำงานบนเซิร์ฟเวอร์ที่ใช้ระบบปฏิบัติการ Linux ซึ่งในการทำงานต้องทำการติดตั้งสิ่งต่อไปนี้ในระบบปฏิบัติการ Linux ด้วย

1. Apache Web Server
2. PERL
3. MySQL Database

ในการใช้งานฐานข้อมูล MySQL ต้องทำการสร้างฐานชื่อ Contact ข้อมูลเพื่อใช้ในการทำงานของระบบ โดยใช้คำสั่งดังต่อไปนี้

```
CREATE DATABASE Contact;
```

หลังจากที่สร้างฐานข้อมูล Contact แล้วต้องทำการสร้างตารางที่จำเป็นในการใช้งานระบบดังต่อไปนี้

- ตาราง contact โดยใช้คำสั่งดังต่อไปนี้

```
CREATE TABLE contact (  
    contactid int(10) unsigned NOT NULL auto_increment,  
    name varchar(40) default NULL,  
    surname varchar(40) default NULL,  
    nickname varchar(20) default NULL,  
    displayname varchar(40) NOT NULL default "",  
    gender char(1) default NULL,  
    birthdate date default NULL,  
    password varchar(20) NOT NULL default "",  
    email varchar(70) default NULL,  
    status tinyint(1) NOT NULL default '0',  
    PRIMARY KEY (contactid)  
) TYPE=MyISAM;
```

- ตาราง contactlist โดยใช้คำสั่งดังต่อไปนี้

```
CREATE TABLE contactlist (
    contactid1 int(10) unsigned NOT NULL default '0',
    contactid2 int(10) unsigned NOT NULL default '0',
    PRIMARY KEY (contactid1,contactid2)
) TYPE=MyISAM;
```

- ตาราง message โดยใช้คำสั่งดังต่อไปนี้

```
CREATE TABLE message (
    msgid int(10) unsigned NOT NULL auto_increment,
    status tinyint(3) unsigned zerofill NOT NULL default '000',
    source int(10) unsigned NOT NULL default '0',
    destination int(10) unsigned NOT NULL default '0',
    date timestamp(14) NOT NULL,
    message varchar(255) NOT NULL default "",
    PRIMARY KEY (msgid)
) TYPE=MyISAM;
```

4. ติดตั้งไฟล์ระบบ ซึ่งดังต่อไปนี้

- ทำการติดตั้ง โปรแกรม daemon.pl โดยการนำสำเนาไฟล์ไว้ที่ subdirectory /etc/ ของระบบปฏิบัติการ Linux แล้วทำการแก้ไขไฟล์ rc.local ใน subdirectory /etc/rc.d/ โดยการเพิ่มบรรทัดคำสั่งดังต่อไปนี้

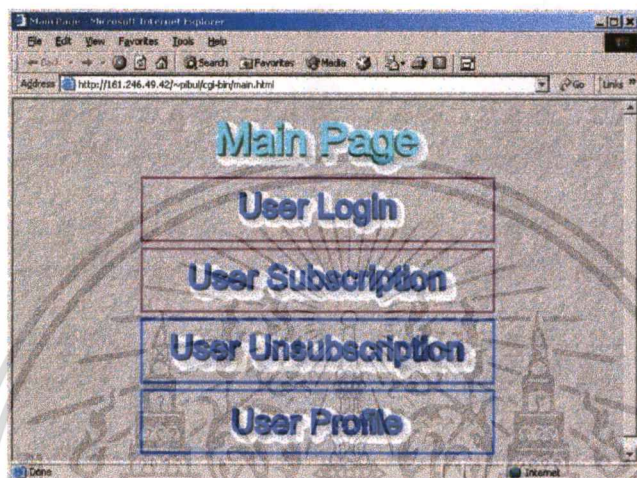

```
/etc/daemon.pl &
```
- ทำการติดตั้ง โปรแกรม CGI โดยการนำสำเนาเพิ่มดังต่อไปนี้ไว้ใน subdirectory CGI ของระบบ
 - login.cgi (เพิ่มข้อมูลที่ใช้ในการเข้าระบบของผู้ใช้ระบบ)
 - subscribe.cgi (เพิ่มข้อมูลที่ใช้ในการสมัครสมาชิกของผู้ใช้ระบบ)
 - unsubscribe.cgi (เพิ่มข้อมูลที่ใช้ในการยกเลิกการเป็นสมาชิกของผู้ใช้ระบบ)
 - userprofile.cgi (เพิ่มข้อมูลที่ใช้ในการแก้ไขข้อมูลส่วนตัวผู้ใช้ระบบ)
 - Client.jar (Java Applet ที่ใช้ในการติดต่อสื่อสารกับเซิร์ฟเวอร์)
- ทำการติดตั้งไฟล์ HTML (main.html) ไว้ที่ subdirectory /apache/htdocs/ ของ Apache Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วิธีการใช้งานระบบ

การใช้งานระบบทำได้ผ่านเว็บเบราว์เซอร์ โดยมีขั้นตอนการใช้งานดังต่อไปนี้

1. ผู้ใช้กำหนด Address เป็น URL ของเว็บไซต์ของระบบที่กำหนดไว้ (ในที่นี้กำหนดเป็น <http://161.246.49.42/~pibul/cgi-bin/main.html> ซึ่งใช้ในการทดลองการทำงานเท่านั้น) ดังรูปภาคผนวกที่ 1



รูปภาคผนวกที่ 1 แสดงหน้าจอเริ่มต้นการทำงานของระบบ

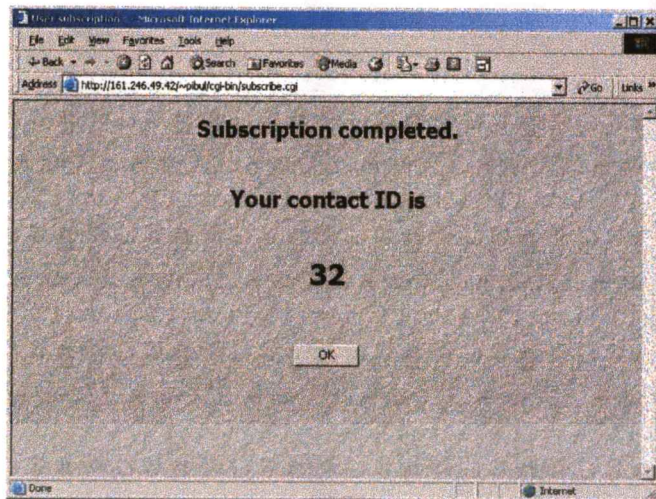
2. ผู้ใช้เลือกใช้หัวข้อ User Subscription ในการสมัครสมาชิกในกรณีผู้ใช้ยังไม่ได้เป็นสมาชิกของ โดยการคลิกเมาส์ที่หัวข้อ User Subscription ผู้ใช้ต้องทำการกรอกข้อมูลส่วนตัวผู้ใช้ ดังรูปภาคผนวกที่ 2

รูปภาคผนวกที่ 2 แสดงหน้าจอสำหรับการสมัครสมาชิก

หลังที่ผู้ใช้ทำการกรอกข้อมูลส่วนตัวผู้ใช้แล้วให้ผู้ใช้คลิกเมาส์ที่ปุ่ม OK เพื่อยืนยันการสมัครสมาชิก จากนั้นผู้ใช้จะได้รับแจ้งหมายเลขประจำตัวของผู้ใช้ ดังรูปภาคผนวกที่ 3

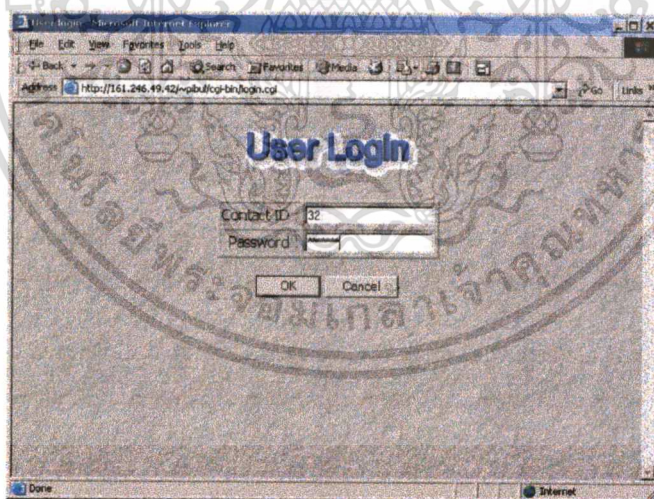
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ใช้งานไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพผนวกที่ 3 แสดงหน้าจอแจ้งหมายเลขประจำตัวสมาชิก
ซึ่งหมายเลขประจำตัวสมาชิกนี้ผู้ใช้จะต้องใช้ในการเข้าระบบ

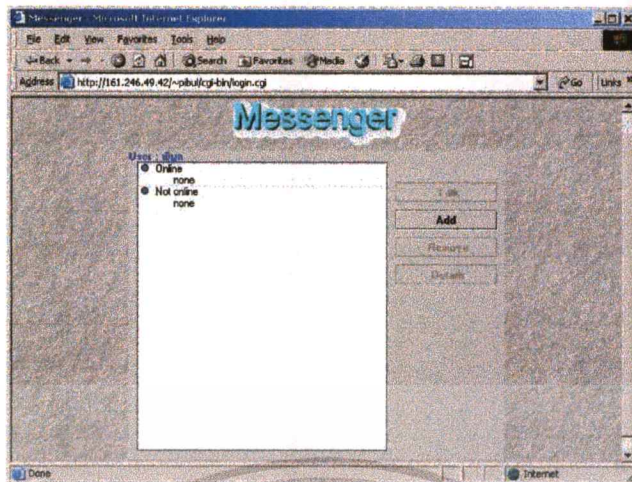
3. ในกรณีที่ผู้ใช้เป็นสมาชิกของระบบผู้ใช้สามารถเข้าใช้งานระบบได้โดยคลิกเมาส์เลือกที่หัวข้อ User Login ในหน้าจอเริ่มการทำงานของระบบ ซึ่งจะปรากฏหน้าจอสำหรับผู้กรอกหมายเลขประจำตัวและรหัสผ่านเพื่อเข้าระบบดังรูปภาพผนวกที่ 4



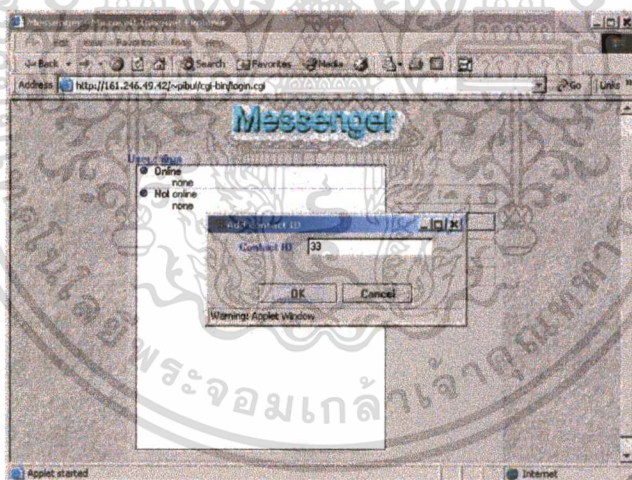
รูปภาพผนวกที่ 4 แสดงหน้าจอเข้าระบบ

หลังจากที่ผู้ใช้กรอกหมายเลขสมาชิกและรหัสผ่านได้ถูกต้องแล้วจะปรากฏหน้าจอสำหรับ
ใช้ในการสื่อสารของผู้ใช้ดังรูปภาพผนวกที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

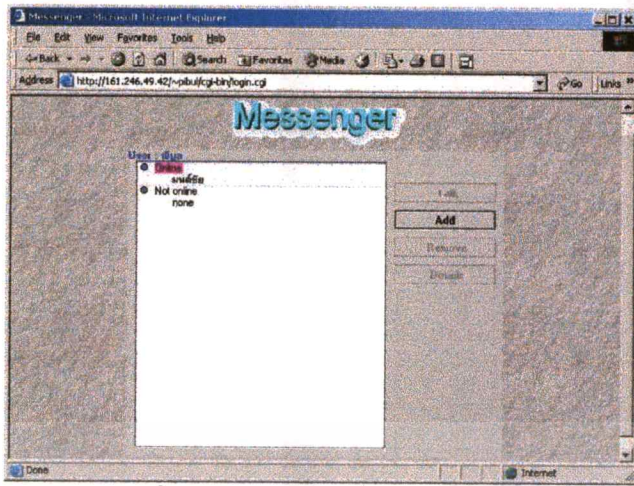


รูปภาพผนวกที่ 5 แสดงหน้าจอสำหรับใช้ในการสื่อสารของผู้ใช้ ผู้ใช้สามารถเพิ่มรายชื่อของผู้ที่ต้องการจะสื่อสารด้วยโดยคลิกเมาส์ที่ปุ่ม Add ในหน้าจอ สำหรับใช้ในการสื่อสาร จะปรากฏหน้าจอสำหรับกรอกหมายเลขประจำตัวผู้ใช้อื่นที่ผู้ใช้ ต้องการติดต่อสื่อสารด้วย ดังภาพผนวกที่ 6



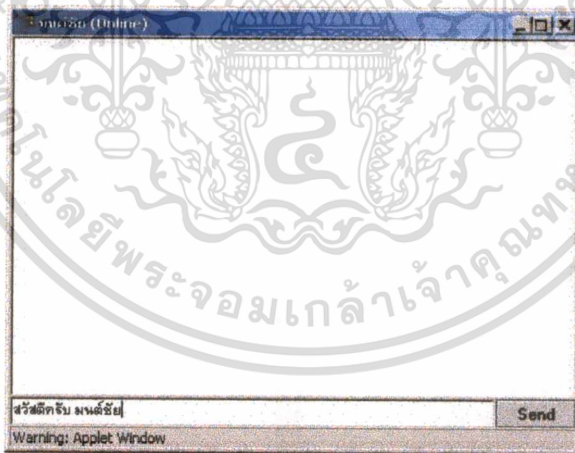
รูปภาพผนวกที่ 6 แสดงหน้าจอสำหรับกรอกหมายเลขผู้ใช้อื่น

ภายหลังจากที่ผู้ใช้กรอกหมายเลขผู้ใช้อื่นที่ต้องการติดต่อสื่อสารด้วยแล้ว จะปรากฏชื่อที่ใช้ในการสื่อสารของผู้ใช้อื่นในหน้าจอด้านซ้ายมือของผู้ใช้ ดังรูปภาพผนวกที่ 7



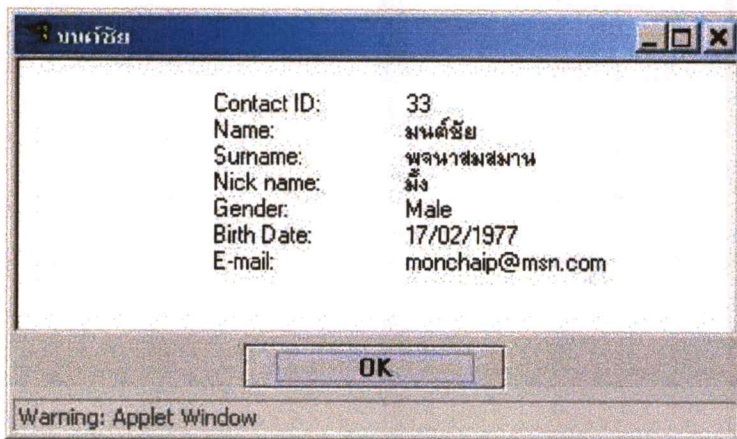
รูปภาคผนวกที่ 7 แสดงหน้าจอเมื่อผู้ใช้เพิ่มรายชื่อสำเร็จ

ผู้ใช้สามารถที่จะติดต่อสื่อสารกับผู้อื่นที่ผู้ใช้ได้เพิ่มรายชื่อเข้ามาโดยการคลิกเมาส์ที่ชื่อของผู้อื่นที่ผู้ใช้ต้องการสื่อสารด้วยในรายการติดต่อทางหน้าจอด้านซ้ายมือของผู้ใช้ แล้วคลิกเมาส์ที่ปุ่ม Talk (หรือคีย์เบิ้ลคลิกเมาส์ที่ชื่อผู้อื่นที่ผู้ใช้ต้องการสื่อสารด้วย) ซึ่งจะปรากฏหน้าจอสำหรับใช้ในการสื่อสารกับผู้อื่นรายนั้น ดังรูปภาคผนวกที่ 8



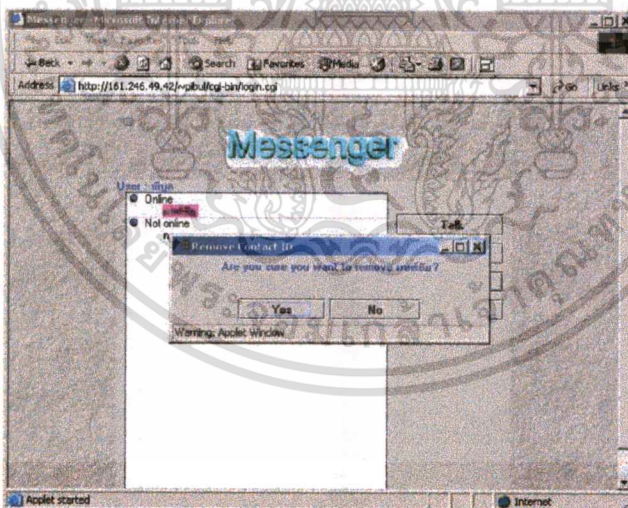
รูปภาคผนวกที่ 8 แสดงหน้าจอสำหรับใช้ในการสื่อสาร

ในกรณีที่ผู้ใช้ต้องการทราบถึงข้อมูลส่วนตัวของผู้อื่นที่อยู่ในรายการติดต่อของผู้ใช้สามารถทำได้โดยคลิกเมาส์ที่ชื่อของผู้ใช้ที่ต้องการทราบข้อมูลส่วนตัวแล้วคลิกเมาส์ที่ปุ่ม Details จะปรากฏหน้าจอแสดงข้อมูลส่วนตัวของผู้ใช้รายนั้นดังรูปภาคผนวกที่ 9



รูปภาพผนวกที่ 9 แสดงหน้าจอแสดงข้อมูลส่วนตัวของผู้ใช้อื่น

ในกรณีที่ผู้ใช้ไม่ต้องการติดต่อกับผู้ใช้อื่นที่มีรายชื่ออยู่ในรายการติดต่อของผู้ใช้ก็ต่อไป ผู้ใช้ก็สามารถที่จะลบรายชื่อของผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้ออกไปได้โดยการคลิกเมาส์เลือกผู้ใช้อื่นที่อยู่ในรายการติดต่อของผู้ใช้ที่ผู้ใช้ต้องการลบ แล้วคลิกเมาส์ที่ปุ่ม Remove ซึ่งจะปรากฏหน้าจอเพื่อขอการยืนยันการลบของผู้ใช้ ดังรูปภาพผนวกที่ 10

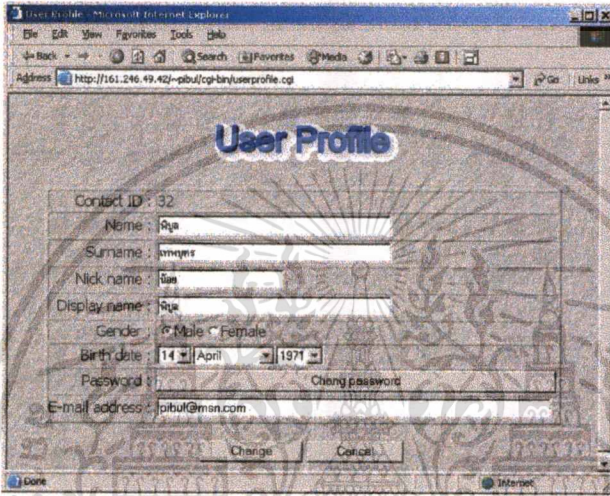


รูปภาพผนวกที่ 10 แสดงหน้าจอเพื่อยืนยันการลบ

เมื่อผู้ใช้ยืนยันการลบโดยการคลิกเมาส์ที่ปุ่ม Yes ผู้ใช้รายนั้นจะถูกลบออกจากรายการติดต่อของผู้ใช้ ซึ่งจะทำให้ผู้ใช้ไม่สามารถติดต่อกับผู้ใช้รายนั้นได้อีกต่อไปจนกว่าจะมีการเพิ่มรายชื่อผู้ใช้รายนั้นใหม่ในกรณีที่ผู้ใช้ต้องการติดต่ออีกครั้ง

4. การแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ

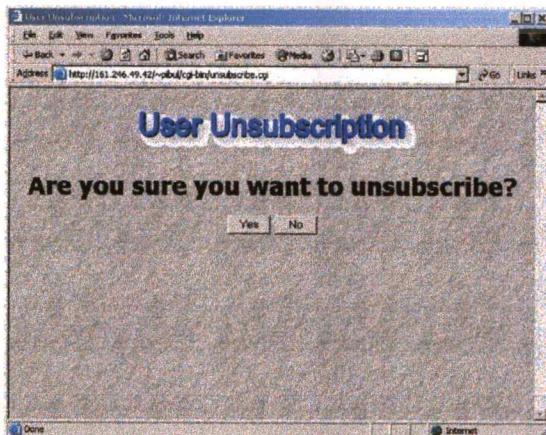
ผู้ใช้งานระบบสามารถที่จะทำการแก้ไขข้อมูลส่วนตัวได้โดยการคลิกเมาส์เลือกหัวข้อ User Profile ในหน้าจอเริ่มต้นการทำงานซึ่งจะปรากฏหน้าจอสำหรับการแก้ไขข้อมูลส่วนตัวของผู้ใช้ดังรูปภาคผนวกที่ 11 ซึ่งในการเข้าสู่หน้าจอแก้ไขข้อมูลส่วนตัวของผู้ใช้นั้นผู้ใช้จำเป็นต้องกรอกหมายเลขสมาชิกและรหัสผ่านเพื่อยืนยันการเป็นสมาชิกก่อนจึงจะสามารถทำการแก้ไขข้อมูลได้



รูปภาคผนวกที่ 11 แสดงหน้าจอเพื่อใช้ในการแก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ

5. การยกเลิกการเป็นสมาชิกระบบ

ในกรณีที่ผู้ใช้งานระบบต้องการยกเลิกการเป็นสมาชิกระบบผู้ใช้สามารถทำได้โดยคลิกเมาส์เลือกที่หัวข้อ User Unsubscription ที่หน้าจอเริ่มต้นการทำงาน ซึ่งผู้ใช้ต้องกรอกหมายเลขประจำตัวและรหัสผ่านเพื่อยืนยันการยกเลิกการเป็นสมาชิกซึ่งจะปรากฏหน้าจอเพื่อให้ผู้ใช้ยืนยันการเป็นสมาชิกของผู้ใช้ดังรูปภาคผนวกที่ 12



รูปภาคผนวกที่ 12 แสดงหน้าจอเพื่อใช้ในการยกเลิกการเป็นสมาชิกของผู้ใช้ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิได้อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้นั้นยืนยันการยกเลิกสมาชิกโดยการคลิกเมาส์ที่ปุ่ม Yes จะถือว่าผู้ใช้นั้นไม่ได้เป็นสมาชิกของระบบและผู้ใช้นั้นจะไม่สามารถเข้าระบบได้โดยใช้หมายเลขประจำตัวของผู้ใช้อีกต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- กิตติ ภัคคีวัฒนะกุล. 2542. **JAVA ฉบับโปรแกรมเมอร์.** กรุงเทพฯ : เคทีพี คอมพ์ แอนด์ คอนซัลท์.
- ฉลองชัย จงประเสริฐพร และ วรวิภา ท่าพระนา. 2452. **CGI WEB Programming การพัฒนาโปรแกรมใช้งานบนเครือข่ายอินเทอร์เน็ต.** กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- คณพล กิ่งสุคนธ์. 2542. **การเขียนโปรแกรมด้วยภาษา JAVA.** กรุงเทพฯ : ส.เอเชียเพลส.
- ทรงเกียรติ ภาวดี. 2542. **เริ่มเขียนสคริปต์ด้วยภาษา PERL.** กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- ทรงเกียรติ ภาวดี. 2542. **แกะรอย CGI เพื่อเขียนสคริปต์เรียกเพจเจอร์/มือถือผ่านเว็บ.** กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- วีรศักดิ์ ชิงถาวร. 2545. **Java Programming.** เล่มที่ 2. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- Coulouris, George et. al. 1999. **Distributed Systems Concepts and Design.** 2nd ed. USA : Addison-Wesley.
- Halsall, Fred. 1995. **Data Communications, Computer Networks and Open Systems.** 4th ed. USA : Addison-Wesley.
- Quigley, Ellie. 1997. **PERL by Example.** 2nd ed. USA : Prentice Hall PTR Upper Saddle River
- Spcll, Brctt. 2000. **Professional Java Programming.** USA : Wrox Prcss Ltd.
- Wong, Clinton. 1997. **Web Client Programming.** USA : O'Reilly & Associates, Inc.

ประวัติผู้เขียน

ชื่อผู้เขียน	นาย พิบูล เทพบุตร
วันเดือนปีเกิด	14 เมษายน 2514
สถานที่เกิด	จังหวัดนครราชสีมา
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (คณิตศาสตร์)
สถาบันที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2539

