

ระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์
และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์

A system for Making Connection between Mobile Phone and Computer
and between Computers



เลขหมู่.....
เลขทะเบียน.....49981
วัน,เดือน,ปี 16 เม.ย. 2547

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้า
ปีการศึกษา 2545
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์

A System for Making Connections between Mobile Phones and Computers and between Computers

คณะผู้จัดทำ นาย กมลวัฒน์ เอี่ยมแจ้ง

นางสาว ขวัญเดือน แซ่หลิว



..... อาจารย์ที่ปรึกษา
(ดร.วิศิษฐ์ หิรัญกิตติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์

นาย กมลวัฒน์ เอี่ยมแจ้ง 42010003

นางสาว ขวัญเดือน แชนท์ 42010036

คร. วิศิษฐ์ หิรัญกิตติ อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการสร้าง และพัฒนาระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ โดยมีเป้าหมายให้โทรศัพท์มือถือ และเครื่องคอมพิวเตอร์ส่วนบุคคลสามารถทำการติดต่อแลกเปลี่ยน โค้ดโปรแกรม ไฟล์ข้อมูล พื้นที่หน่วยความจำบนฮาร์ดดิสก์ เครื่องพิมพ์ และใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุดกับผู้ใช้เครื่องข่าย ด้วยเหตุผลที่โปรแกรมจาวาสามารถรันได้บนทุกระบบปฏิบัติการ รวมทั้งบนอุปกรณ์สื่อสารขนาดเล็กเช่น โทรศัพท์มือถือ ได้ด้วย โครงการนี้ จึงเลือกใช้ภาษาจาวาในการพัฒนาระบบ

ปฏิญานิพนธ์นี้มุ่งเน้นการสร้างและการพัฒนาระบบซึ่งสามารถแลกเปลี่ยน โค้ดโปรแกรมแลกเปลี่ยนการรัน โปรแกรมระหว่างเครื่องคอมพิวเตอร์ และใช้ทรัพยากรร่วมกันได้ โดยได้มีการออกแบบโปรโตคอลขึ้นมาใหม่โดยคัดแปลงจากโปรโตคอล HTTP เพื่อใช้ติดต่อกันภายในระบบ จากการทดลองกับโปรโตคอลนี้ระบบสามารถทำงานได้บรรลุวัตถุประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A system for Making Connection between Mobile Phone and Computer and between Computers

Kamolwat Eiamchaeng

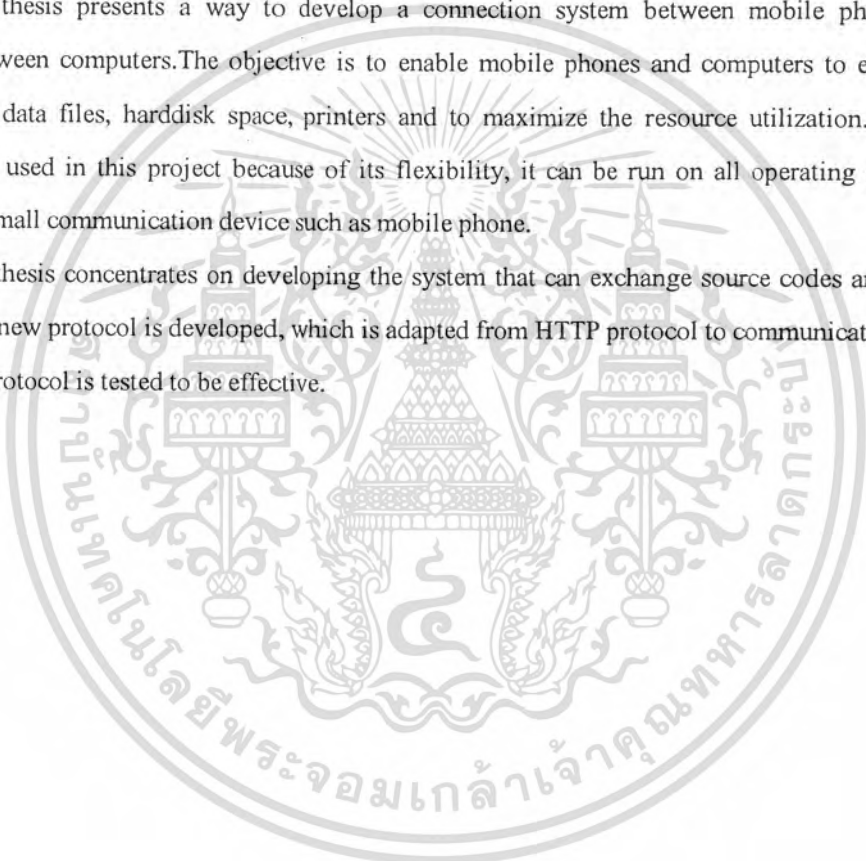
Kwanduen Saelew

Dr. Visit Hirankitti Advisor

ABSTRACT

This thesis presents a way to develop a connection system between mobile phone and computer, between computers. The objective is to enable mobile phones and computers to exchange source codes, data files, harddisk space, printers and to maximize the resource utilization. Java is selected to be used in this project because of its flexibility, it can be run on all operating systems, including on small communication device such as mobile phone.

This thesis concentrates on developing the system that can exchange source codes and share resources so a new protocol is developed, which is adapted from HTTP protocol to communicate within system. This protocol is tested to be effective.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดีหากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์วิศิษฐ์ หิรัญกิตติ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างยิ่ง

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

กมลวัฒน์ เอี่ยมแจ้ง
ขวัญเดือน แซ่หลิว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 JAVA	4
2.1 แพลตฟอร์มจาวา	4
2.2 จาวาเทคโนโลยีไมโครเอ็ดจัน (J2ME™ Technology)	5
2.2.1 คอนฟิกูเรชันใน J2ME	5
2.2.2 J2ME สำหรับอุปกรณ์ไร้สาย	7
2.2.3 ความต้องการของระบบ	7
2.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME	7
2.3.1 ไลบรารีของ CLCD	7
2.3.2 ชั้นคลาสของ J2SE	8
2.3.3 คลาสที่มีเฉพาะใน CLDC	8
2.3.4 ประเภทของข้อมูลพื้นฐาน	8
2.3.5 ไลบรารีของ MIDP	9
2.3.6 คลาสโปรแกรมจัดการแอปพลิเคชัน	9
2.3.7 คลาสของส่วนติดต่อกราฟิกกับผู้ใช้ (GUI Class)	9
2.3.8 คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)	9
2.3.9 คลาสของเครือข่าย (Network Class)	9
2.4 MIDlet	9
2.5 วงจรการทำงานของMIDlet	10
บทที่ 3 ภาพรวมของระบบ	12
3.1 การติดต่อโดยรวมของภายในระบบ	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1	การติดต่อกับเซิร์ฟเวอร์กลางเพื่อเข้าสู่ระบบ	12
3.1.2	การติดต่อกับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูลเพื่อจัดเก็บข้อมูล	13
3.1.3	การทำงานของเครื่องไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล	15
3.1.4	การทำงานของเครื่องไคลเอนต์ที่เป็นโทรศัพท์มือถือ	15
3.2	ไคอะแกรมขั้นตอนการติดต่อของไคลเอนต์ในระบบ	16
3.2.1	ไคอะแกรมแสดงขั้นตอนการติดต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์กลาง	16
3.2.2	ไคอะแกรมแสดงขั้นตอนการติดต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์ที่ให้พื้นที่ในการสำรองข้อมูล	16
3.2.3	ไคอะแกรมแสดงขั้นตอนการติดต่อในกรณีที่เครื่องไคลเอนต์เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล	17
3.2.4	ไคอะแกรมแสดงขั้นตอนการติดต่อในกรณีที่เครื่องไคลเอนต์เป็นโทรศัพท์มือถือ	17
บทที่ 4	การเชื่อมต่อเครือข่าย โพรโตคอล และการออกแบบ	18
4.1	รูปแบบการเชื่อมต่อ	18
4.1.1	การติดต่อผ่าน TCP	19
4.1.2	รับส่ง Datagram Packets ผ่าน UDP	19
4.2	โพรโตคอล HTTP	19
4.3	การออกแบบการเชื่อมต่อ และ โพรโตคอลของระบบ	20
4.3.1	การเลือกรูปแบบการเชื่อมต่อที่จะใช้	20
4.3.2	การออกแบบโพรโตคอล	21
4.3.2.1	รูปแบบของฟอร์ม	21
4.3.2.2	FMP Header	22
4.3.2.3	FMP Method	23
4.3.3	FMP Status	29
4.3.3.1	ตอบรับ (OK)	29
4.3.3.2	ล้มเหลว (FAIL)	29
4.3.3.3	ตอบสนอง (ACK)	29
บทที่ 5	เซิร์ฟเวอร์	30
5.1	เซิร์ฟเวอร์หลัก	30
5.1.1	การออกแบบฐานข้อมูลของเซิร์ฟเวอร์หลัก	30
5.1.2	โปรแกรมช่วยจัดการข้อมูลบนฐานข้อมูล	31
5.1.2.1	การสร้างข้อมูลของยูสเซอร์ใหม่ที่ทำกรลงทะเบียนเพื่อเข้าสู่ระบบ	32
5.1.2.2	การค้นหาข้อมูลของยูสเซอร์ที่ทำกรลงทะเบียนแล้ว	33
5.1.2.3	การแก้ไขข้อมูลของยูสเซอร์ที่ทำกรลงทะเบียนแล้ว	34
5.1.2.4	การแสดงยูสเซอร์เนม และ ไอพีแอดเดรสของยูสเซอร์ที่ลงทะเบียนแล้ว	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานับ ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 เซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล	36
5.2.1 การออกแบบฐานข้อมูลของเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูลเพื่อจัดเก็บข้อมูล	37
5.2.1.1 ตารางข้อมูลการใช้พื้นที่ของแต่ละยูสเซอร์	37
5.2.1.2 ตารางข้อมูลขนาดพื้นที่ที่เปิดให้บริการของ authority แต่ละประเภท	38
5.2.2 โปรแกรมช่วยจัดการข้อมูลบนฐานข้อมูล	39
บทที่ 6 การออกแบบระบบไฟล์บน โทรศัพท์มือถือ	40
6.1 Record Management Systems	40
6.2 การออกแบบระบบไฟล์บน MIDP	41
6.2.1 หลักการในการทำ Record Store ให้เป็นไฟล์	41
6.2.2 ประเภทของไฟล์	41
6.2.2.1 Class File	42
6.2.2.2 Class Bin File	43
6.2.2.3 Class Folder	44
6.2.2.4 Class VirtualFile	45
บทที่ 7 การพัฒนาโปรแกรมส่วนไคลเอนต์	47
7.1 การเข้าสู่ระบบจัดการไฟล์	49
7.1.1 กรณีที่ไคลเอนต์เป็นเครื่องคอมพิวเตอร์	49
7.1.2 กรณีที่ไคลเอนต์เป็น โทรศัพท์มือถือ	49
7.2 การจัดการไฟล์	50
7.3 การส่งพิมพ์งาน โดยผ่านเครื่องคอมพิวเตอร์เครื่องอื่นที่อยู่ในระบบ	53
7.4 การส่งคอมมานด์ไลน์ไปยังเครื่องคอมพิวเตอร์	53
บรรณานุกรม	54
ภาคผนวก	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2-1 แสดงแพลตฟอร์มจาวา 2 ทั้ง 3 รุ่น ที่ใช้งานอยู่ในขณะนี้	4
รูปที่ 2-1 แสดงแพลตฟอร์มจาวา 2 ทั้ง 3 รุ่น ที่ใช้งานอยู่ในขณะนี้	6
รูปที่ 2-3 วงจรการทำงานของ MIDlet	11
รูปที่ 3-1 แสดงการติดต่อจากไคลเอนต์ไปยังเซิร์ฟเวอร์กลางเพื่อขอเข้าสู่ระบบ	12
รูปที่ 3-2 แสดงการตรวจสอบข้อมูลของไคลเอนต์ที่ทำการขอเข้าสู่ระบบ	13
รูปที่ 3-3 แสดงการส่งไอพีแอสเครสของไคลเอนต์ที่ออนไลน์อยู่ในขณะนั้นไปยังไคลเอนต์ที่ขอเข้าสู่ระบบ	13
รูปที่ 3-4 แสดงติดต่อกับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล	14
รูปที่ 3-5 แสดงการตรวจสอบพื้นที่ที่ให้บริการของแต่ละไคลเอนต์ก่อนจัดเก็บข้อมูล	14
รูปที่ 3-6 แสดงการส่งผลการจัดเก็บข้อมูลกลับไปยังเครื่องไคลเอนต์	15
รูปที่ 3-7 แสดงการเรียกใช้บริการระหว่างเครื่องไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์	15
รูปที่ 3-8 แสดงการเรียกใช้บริการระหว่างเครื่องไคลเอนต์ที่เป็นโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์	16
รูปที่ 3-9 แสดงขั้นตอนการเชื่อมต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์กลาง	16
รูปที่ 3-10 แสดงขั้นตอนการเชื่อมต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์ ที่ให้บริการพื้นที่ในการสำรองข้อมูล	17
รูปที่ 3-11 แสดงขั้นตอนการเชื่อมต่อในกรณีที่เครื่องไคลเอนต์เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล	17
รูปที่ 3-12 แสดงขั้นตอนการเชื่อมต่อในกรณีที่เครื่องไคลเอนต์เป็นโทรศัพท์มือถือ	17
รูปที่ 4-1 แสดงขั้นตอนการติดต่อกันผ่านซอกเกต	18
รูปที่ 4-2 แสดงขั้นตอนการติดต่อกันผ่าน UDP	19
รูปที่ 4-3 แสดงรูปแบบของ HTTP request	20
รูปที่ 4-4 แสดงรูปแบบของ HTTP response	20
รูปที่ 4-5 แสดงรูปแบบ request และ response ของ FMP	21
รูปที่ 4-6 แสดงตัวอย่างของรูปแบบในการร้องขอเป็นแพ็กเกจ	22
รูปที่ 4-7 แสดง Header ทั้งหมดของ FMP	22
รูปที่ 4-8 แสดงการขอติดต่อกับไคลเอนต์อื่น	23
รูปที่ 4-9 แสดงการออกไปโฟลเดอร์แม่	23
รูปที่ 4-10 แสดงการเปิดโฟลเดอร์	24
รูปที่ 4-11 แสดงการลบไฟล์	24
รูปที่ 4-12 แสดงการเปลี่ยนชื่อไฟล์	24
รูปที่ 4-13 แสดงการคัดลอกไฟล์	25

รูปที่ 4-13 แสดงการคัดลอกไฟล์ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-14 แสดงการย้ายไฟล์	25
รูปที่ 4-15 แสดงการดูคุณสมบัติของไฟล์	26
รูปที่ 4-16 แสดงการส่งไฟล์ให้เซิร์ฟเวอร์	26
รูปที่ 4-17 แสดงการรับไฟล์ให้เซิร์ฟเวอร์	26
รูปที่ 4-18 แสดงการรันคอมมานด์ไฟล์จากเซิร์ฟเวอร์	27
รูปที่ 4-19 แสดงการพิมพ์ไฟล์ที่เซิร์ฟเวอร์	27
รูปที่ 4-20 แสดงการชิง โคร โนซ์กับเซิร์ฟเวอร์	27
รูปที่ 4-21 แสดงการเข้าระบบ	28
รูปที่ 4-22 แสดงการออกจากระบบ	28
รูปที่ 4-23 แสดงการลงทะเบียน	28
รูปที่ 5-1 แสดงการติดต่อโดยรวมระหว่างเซิร์ฟเวอร์หลัก กับไคลเอนต์	30
รูปที่ 5-2 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลัก	31
รูปที่ 5-3 แสดงยูสเซอร์อินเตอร์เฟซของ โปรแกรมช่วยจัดการบนฐานข้อมูล	32
รูปที่ 5-4 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลักหลังจากทำการ create ยูสเซอร์ใหม่	32
รูปที่ 5-5 แสดงการกรอกยูสเซอร์เนมเพื่อค้นหาข้อมูล	33
รูปที่ 5-6 แสดงผลของการค้นหาข้อมูล	34
รูปที่ 5-7 แสดงการกรอกข้อมูลที่ต้องการแก้ไข	34
รูปที่ 5-8 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลักหลังจากทำการแก้ไขข้อมูลของยูสเซอร์ Air	35
รูปที่ 5-9 แสดงผลของการ List ชื่อ และ ไอพีแอสเครสทั้งหมดที่ลงทะเบียนกับระบบ	36
รูปที่ 5-10 แสดงการติดต่อโดยรวมระหว่างเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูลกับไคลเอนต์	37
รูปที่ 5-11 แสดงตาราง Storage_DB บนฐานข้อมูลของเซิร์ฟเวอร์ที่ให้บริการสำรองพื้นที่	38
รูปที่ 5-12 แสดงตาราง TYPE บนฐานข้อมูลของเซิร์ฟเวอร์ที่ให้บริการสำรองพื้นที่	38
รูปที่ 5-13 แสดงยูสเซอร์อินเตอร์เฟซของ โปรแกรมช่วยจัดการบนฐานข้อมูล	39
รูปที่ 6-1 ตารางแสดงการเก็บข้อมูลของ RecordStore	40
รูปที่ 6-2 แสดงการใช้ Record Store เก็บข้อมูลระบบไฟล์	41
รูปที่ 6-3 แสดงโครงสร้างของระบบไฟล์บน โทรศัพท์มือถือ	42
รูปที่ 6-4 แสดงคลาส File	42
รูปที่ 6-5 แสดงคลาส BinFile	43
รูปที่ 6-6 แสดงคลาส Folder	44
รูปที่ 6-7 แสดงคลาส VirtualFile	45
รูปที่ 7-1 แสดงการทำงานของไคลเอนต์	47
รูปที่ 7-2 แสดงการทำงานของไคลเอนต์ส่วนเซิร์ฟเวอร์	48
รูปที่ 7-3 แสดงขั้นตอนการเริ่มเข้าสู่ระบบของไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์ให้นำไปใช้ประโยชน์ด้านการ	49

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7-4 แสดงขั้นตอนการเริ่มเข้าสู่ระบบของไคลเอนต์ที่เป็นโทรศัพท์มือถือ	50
รูปที่ 7-5 แสดงการทำงานส่วนจัดการไฟล์	50
รูปที่ 7-6 แสดงส่วนที่ติดต่อกับผู้ใช้งานส่วนจัดการไฟล์บนเครื่องคอมพิวเตอร์	51
รูปที่ 7-7 ส่วนที่ติดต่อกับผู้ใช้งานส่วนจัดการไฟล์บนโทรศัพท์มือถือ	52
รูปที่ 7-8 แสดงส่วนที่ติดต่อกับผู้ใช้งานสังคมนาฬ	53
รูปที่ 7-9 แสดงผลจากการรับจาวาไฟล์	54



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบัน โทรศัพท์มือถือได้เข้ามาเป็นส่วนหนึ่งในชีวิตประจำวันของผู้คนทั่วไปในสังคม ซึ่งอาจกล่าวได้ว่าเป็นปัจจัยที่ 5 ของมนุษย์แล้วและในอนาคตความสามารถของโทรศัพท์มือถือมีแนวโน้มที่จะถูกพัฒนาให้มีความสามารถเพิ่มมากขึ้นตามเทคโนโลยีที่เปลี่ยนแปลงไป เช่น ความเร็วในการรับส่งข้อมูล ความละเอียดของจอภาพ และความเร็วในการประมวลผล ซึ่งจะทำให้โทรศัพท์มือถือไม่ได้เป็นเพียงแค่โทรศัพท์มือถืออีกต่อไป แต่ข้อจำกัดที่สำคัญของโทรศัพท์มือถือก็คือขนาดของหน่วยความจำที่ใช้ในการเก็บข้อมูลมีขนาดเล็ก และความเร็วในการประมวลผลนั้นต่ำ เพราะในอนาคตแนวโน้มของโทรศัพท์มือถือจะมีขนาดเล็กลง ทำให้การเพิ่มจำนวนหน่วยความจำในตัวโทรศัพท์มือถือ หรือการพัฒนาการประมวลผลบน โทรศัพท์มือถือทำได้ยากยิ่งขึ้น และมีราคาสูง ซึ่งจะส่งผลให้เกิดข้อจำกัดในการใช้งานขึ้นมาเพราะการเก็บข้อมูล หรือการสร้างบริการการทำงาน จะถูกจำกัดตามไปด้วย

ประกอบกับในปัจจุบันนั้นเครื่องคอมพิวเตอร์ส่วนบุคคลมีกระจายอยู่ทั่วไปตามที่อยู่อาศัย และสำนักงานต่างๆ ซึ่งเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น ไม่มีข้อจำกัดเหมือน โทรศัพท์มือถือ กล่าวคือมีหน่วยความจำขนาดใหญ่ การประมวลผลมีความรวดเร็วและราคาถูกกว่าอุปกรณ์ของโทรศัพท์มือถือถ้าเทียบกันในเชิงของประสิทธิภาพกับราคา แต่ก็มีข้อเสียในแง่ของขนาดที่ไม่สะดวกในการพกพาเพื่อใช้งาน รวมทั้งข้อจำกัดของระบบปฏิบัติการที่ทำให้บางครั้งระบบปฏิบัติการที่ต่างกันไม่สามารถทำงานร่วมกันได้

ด้วยเหตุผลข้างต้นถ้าสามารถสร้างระบบที่โทรศัพท์มือถือ และเครื่องคอมพิวเตอร์ส่วนบุคคลสามารถนำจุดเด่นของตนมาใช้งานร่วมกันได้ และลดข้อจืดจางของแต่ละฝ่าย ซึ่งสามารถทำงานผ่านโทรศัพท์มือถือได้โดยไม่ต้องไปตั้งงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล นับเป็นการอำนวยความสะดวกให้แก่ผู้ใช้ซึ่งไม่จำเป็นที่จะต้องอยู่หน้าเครื่องคอมพิวเตอร์เพื่อที่จะทำงานต่างๆ และยังเป็นการแบ่งปันทรัพยากรที่มีอยู่ให้เกิดประสิทธิภาพสูงสุดอีกทางหนึ่ง

1.2 วัตถุประสงค์ของโครงการ

- สร้างระบบเชื่อมต่อระหว่างโทรศัพท์มือถือและเครื่องคอมพิวเตอร์ส่วนบุคคล
- สร้างระบบเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลทั้งที่มีระบบปฏิบัติการเหมือนกัน และต่างกัน
- สร้างบริการในรูปแบบของ File Manager เพื่อใช้กับระบบที่พัฒนาขึ้นมา
- สร้างบริการในรูปแบบของ Command Line เพื่อใช้กับระบบที่พัฒนาขึ้นมา

เอกสารนี้เป็นสร้างบริการการเรียกใช้พรินเตอร์ผ่านระบบที่พัฒนาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างโปรแกรมการจัดการ server กลาง และ server storage

1.3 ขอบเขตของโครงการ

โครงการที่ได้ทำขึ้นมาต้องการที่จะศึกษาการสร้างระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลกับเครื่องคอมพิวเตอร์ส่วนบุคคล โดยมีเป้าหมายให้สามารถเขียนโปรแกรมภาษาจาวาเพื่อสร้างระบบการติดต่อในลักษณะของ client และ server

สร้างระบบการส่งงานระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลโดยจะมีบริการประเภทไฟล์เมเนเจอร์อันได้แก่ การสร้าง folder การ copy ไฟล์ การ delete ไฟล์ การเปลี่ยนชื่อไฟล์ การ upload ไฟล์ การ download ไฟล์ การเรียกดู properties ไฟล์

สร้างระบบการส่งงานระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลโดยจะมีบริการส่งไฟล์ไปประมวลผลยังเครื่องอื่นๆ โดยอยู่ในรูปแบบของ command line รวมทั้งบริการในการส่งไฟล์ไป print ผ่านเครื่องอื่นๆที่มีพรินเตอร์ติดตั้งอยู่

สร้างโปรแกรมการจัดการการเก็บข้อมูลของ user ในระบบ และโปรแกรมการจัดการการควบคุมพื้นที่ในการรับฝากข้อมูลในเครื่อง server กลาง และ server storage

ซึ่งโปรแกรมที่ได้พัฒนาขึ้นมาอาจยังไม่สามารถนำไปใช้งานจริงได้ เนื่องจากการพัฒนาระบบขึ้นมาเพื่อศึกษาการติดต่อระหว่างโทรศัพท์มือถือ กับเครื่องคอมพิวเตอร์ส่วนบุคคล และระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล เพื่อทำการโอนย้ายไฟล์ ส่งคำสั่งไปประมวลผล และการเรียกใช้งานพรินเตอร์นั้นยังคงมีข้อจำกัดทางด้านความปลอดภัย ทำให้การนำไปใช้งานจริงอาจเกิดความไม่ปลอดภัยขึ้นกับข้อมูลได้ ซึ่งการพัฒนางานวิจัยนี้ต้องการที่จะเป็นแนวทางให้แก่ผู้ที่สนใจได้ศึกษา และสามารถนำความรู้นี้ไปประยุกต์ใช้ในการสร้างระบบติดต่อระหว่าง โทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และระหว่างเครื่องคอมพิวเตอร์เองได้ต่อไปในอนาคต

1.4 วิธีการดำเนินงาน

ขั้นตอนในการดำเนินงาน โครงการเริ่มตั้งแต่ศึกษารายละเอียดของภาษาจาวาทั้งในส่วนของ J2ME และ J2EE แล้วจึงเริ่มทำการทดลองเขียน โปรแกรมการติดต่อระหว่าง โทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล จากนั้นทำการออกแบบโปรโตคอลที่จะใช้ในการติดต่อของระบบ เมื่อทำการติดต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์ส่วนบุคคล และระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลได้แล้ว จึงทำการเขียนโปรแกรมในส่วนของการให้บริการด้านไฟล์เมเนเจอร์ทั้งบน โทรศัพท์มือถือ และบนเครื่องคอมพิวเตอร์ส่วนบุคคล จากนั้นทำการเขียนโปรแกรมในส่วนของการบริการการส่งไฟล์ไปประมวลผลบนเครื่องอื่นๆ และบริการการเรียกใช้พรินเตอร์ตามลำดับ เมื่อบริการต่างๆทำงานได้แล้วจึงเริ่มทำการเขียน server กลาง และโปรแกรมการจัดการบน server กลางที่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้เก็บข้อมูลของ user จากนั้นก็เริ่มเขียน server storage และ โปรแกรมการจัดการบน server storage ที่
จะใช้ในการรับฝากข้อมูลจาก user แล้วจึงทำการเชื่อมต่อ โปรแกรมทั้งหมดเป็นระบบ และทำการทดสอบ
ใช้งานจริง พร้อมทั้งหาข้อบกพร่องเพื่อนำกลับมาแก้ไขระบบให้ดีขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

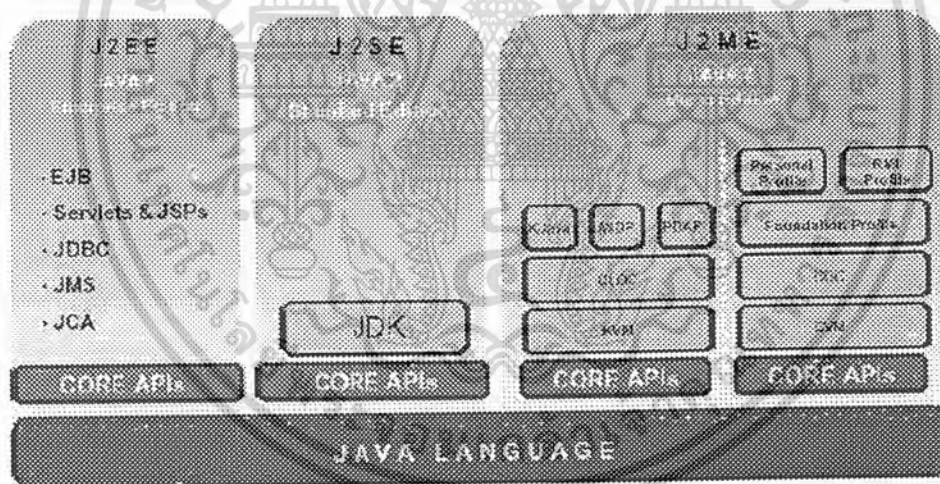
JAVA

2.1 แพลตฟอร์มจาวา

ในช่วงเวลาระยะหลังที่ผ่านมาจาวาได้กลายเป็นแพลตฟอร์มการพัฒนาแอปพลิเคชันแบบเชิงวัตถุ (object-oriented) ในอุปกรณ์และงานต่าง ๆ อย่างเต็มตัว นับตั้งแต่แอปพลิเคชันเซิร์ฟเวอร์ระดับองค์กร เครื่องคอมพิวเตอร์แบบตั้งโต๊ะทั่วไป จนถึงแอปพลิเคชันฝังตัวสำหรับอุปกรณ์ขนาดเล็ก

แพลตฟอร์มจาวา 2 ที่ใช้งานอยู่ในขณะนี้ 3 รุ่นด้วยกัน แต่ละรุ่นเน้นนำไปใช้กับเฉพาะกลุ่มแอปพลิเคชัน ดังนี้

- Java 2 Enterprise Edition (J2EE) ใช้งานกับแอปพลิเคชันบนเครื่องเซิร์ฟเวอร์สำหรับองค์กรที่รองรับระบบงานใหญ่ ๆ และโคลเอนต์จำนวนมาก
- Java 2 Standard Edition (J2SE) ใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะทั่วไป
- Java 2 Micro Edition (J2ME) ใช้งานกับแอปพลิเคชันรุ่นใหม่ซึ่งเน้นกลุ่มผู้ใช้ อุปกรณ์ อิเล็กทรอนิกส์และอุปกรณ์ฝังตัว (eMbedded)



รูปที่ 2-1 แสดงแพลตฟอร์มจาวา 2 ทั้ง 3 รุ่น ที่ใช้งานอยู่ในขณะนี้

จาวาแต่ละรุ่นจะมีเวอร์ชวลแมชีน (Virtual achine) เฉพาะซึ่งปรับแต่งเป็นพิเศษ เพื่อให้สนับสนุนการทำงานของแอปพลิเคชันที่จะนำไปใช้ได้ดียิ่งขึ้น HotSpot VM เป็น Virtual machine ของจาวาที่ปรับแต่งเพื่อเพิ่มประสิทธิภาพของแอปพลิเคชันบนเครื่องเซิร์ฟเวอร์ที่เขียนด้วย J2EE

JVM เป็น Virtual machine สำหรับจาวาทั่วไปที่ปรับแต่งเพื่อใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ นอกจากนี้ HotSpot VM ยังสามารถทำงานเข้ากับแอปพลิเคชันที่เขียนด้วย JSE

ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Virtual Machine ที่ออกแบบมาเพื่อใช้งานกับ J2ME มี 2 ประเภทด้วยกัน คือ C Virtual Machine <CVM> และ K Virtual Machine <KVM> ซึ่งมีขนาดกะทัดรัดและใช้ทรัพยากรของระบบน้อยกว่า HotSpot VM และ JVM

2.2 จาวาทูโมโครเอดิชัน (J2ME™ Technology)

J2ME เป็นแพลตฟอร์มจาวาที่ออกแบบมาเพื่อใช้กับแอปพลิเคชันที่ทำงานบนอุปกรณ์ขนาดเล็ก เช่น โทรศัพท์เคลื่อนที่ พีดีเอ โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต โทรศัพท์ดิจิทัลขนาดเล็ก อุปกรณ์บันทึกและระบบนำทางในรถยนต์ สวิตช์ในระบบเครือข่าย คอมพิวเตอร์ของเครื่องใช้อำนวยความสะดวกภายในบ้าน ฯลฯ

J2ME ได้นำโครงสร้างแบบโมดูลที่มีความยืดหยุ่นสูงเข้ามาใช้ เพื่อให้สามารถสนับสนุนการทำงานอุปกรณ์หลากหลายประเภท รูปที่ 2.1 แสดงให้เห็นว่า J2ME กำหนดชั้นของซอฟต์แวร์ไว้ 3 เลเยอร์ด้วยกัน โดยเลเยอร์ทั้งหมดจะอยู่เหนือชั้นระบบปฏิบัติการของอุปกรณ์ ดังนี้

- เลเยอร์ Java Virtual Machine เป็นเลเยอร์ของ Java Virtual Machine ปรับแต่งให้เข้ากับระบบปฏิบัติการของอุปกรณ์ และรองรับแต่ละ configuration ของ J2ME จากรูปที่ 2.1 Virtual Machine ของ J2ME ได้แก่ CVM และ KVM
- เลเยอร์ Configuration เป็นเลเยอร์ของคอนฟิกูเรชันของ J2ME ซึ่งกำหนดคลาสไลบรารี อุปกรณ์ใช้งานทั่วไปหรือกลุ่มอุปกรณ์ที่มีความต้องการหน่วยความจำและหน่วยประมวลผลใกล้เคียงกัน จากรูปที่ 2.1 Configuration ใน J2ME มี 2 ประเภทด้วยกัน คือ CDC < Connected Device Configuration > และ CLDC < Connected Limited Device Configuration >
- เลเยอร์ Profile เป็นเลเยอร์ที่สร้างเหนือเลเยอร์คอนฟิกูเรชัน โดยกำหนดคลาสไลบรารีเพื่อสนองต่อความต้องการของตลาดเฉพาะกลุ่ม จากรูปที่ 2.1 ตัวอย่างของ Profile ในเลเยอร์นี้ ได้แก่ PDAP, MIDP, Foundation Profile และ Personal Profile

2.2.1 คอนฟิกูเรชันใน J2ME

คอนฟิกูเรชันและโพรไฟล์เป็นองค์ประกอบหลักของ J2ME โดยมีจุดสำคัญ คือ เพื่อปรับแต่ง virtual machine และคลาสไลบรารีให้เหมาะสมกับอุปกรณ์แต่ละประเภท คอนฟิกูเรชัน คือชุดที่มีคุณสมบัติขั้นต่ำของ Java Virtual Machine และจาวาคลาสสำหรับอุปกรณ์แต่ละประเภท เป็นตัวแทนของอุปกรณ์ที่ลักษณะคล้ายคลึงกัน อาจกล่าวได้ว่า คอนฟิกูเรชันเป็นตัวกำหนดคุณสมบัติหรือไลบรารีกำหนดใน คอนฟิกูเรชันจะมีในทุกอุปกรณ์ที่อยู่ในกลุ่มเดียวกัน

ปัจจุบัน คอนฟิกูเรชันใน J2ME แบ่งเป็น 2 ประเภทคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) โดยมีเป้าหมายที่กลุ่มอุปกรณ์ 2 ประเภทซึ่งมีความต้องการหน่วยความจำและหน่วยประมวลผลพื้นฐานใกล้เคียงกัน

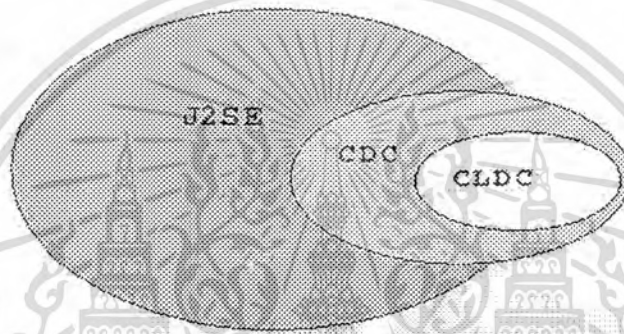
- CDC เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่น ติดตั้งตายตัว และใช้เชื่อมต่อข้อมูล โดยปรกติ

มักมีส่วนติดต่อกับผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2-16 เมกกะไบต์ ใช้

หน่วยประมวลผลแบบ 32 บิต หรือมากกว่าเชื่อมต่อเครือข่ายที่มีแบนด์วิดธ์สูงอย่างต่อเนื่อง

เนื่องโดยอาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ได้แก่ โทรศัพท์มือถือขนาดเล็ก อินเทอร์เน็ตทีวี โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง อุปกรณ์บันทึกและระบบนำทางในรถยนต์

- CLDC เป็นอุปกรณ์ส่วนบุคคล พกพาได้ และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้แบบง่าย ๆ (เมื่อเทียบกับระบบบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ) มีหน่วยความจำประมาณ 128 กิโลไบต์ – 1 เมกกะไบต์ ใช้หน่วยประมวลผลแบบ 16 หรือ 32 บิต เชื่อมต่อกับเครือข่ายที่มีแบนด์วิดธ์ต่ำเป็นระยะเวลาสั้น ๆ โดยไม่อาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์เคลื่อนที่แบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องปาล์มโอเอสแบบพกพา



รูปที่ 2-2 แสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีใน CDC และ CLDC

ในชั้นของ Configuration มีคลาส 2 ประเภทด้วยกัน คือ คลาสที่นำมาจาก J2SE และที่ออกแบบเฉพาะอุปกรณ์ขนาดเล็ก คลาสที่นำมาจาก J2SE จะมีคุณสมบัติอย่างเดียวกันกับคลาสใน J2SE หรือเป็นซับคลาสของ J2SE เช่น แพ็กเกจ java.lang java.io และ java.util จากรูปที่ 2-2 แสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีของ CDC และ CLDC จากภาพจะเห็นว่าคลาสใน CLDC ส่วนใหญ่ทำงานร่วมกับคลาสใน CDC ได้ดีเช่นเดียวกับกรณีของคลาสใน CLDC

ขณะเดียวกันคลาสที่ไม่ได้นำมาจาก J2SE และออกแบบเพื่อใช้เฉพาะอุปกรณ์มักทำงานร่วมกับ J2SE ได้ไม่ค่อยดี ใน CLDC คลาสเหล่านี้จะอยู่ในกลุ่มกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) โดยระบุไว้ในแพ็คเกจ javax.microedition.io

2.2.2 J2ME สำหรับอุปกรณ์ไร้สาย

J2ME ได้ให้กำเนิดแอปพลิเคชันยุคใหม่บนอุปกรณ์ไร้สาย ช่วยให้เกมแบบหลายผู้เล่นที่ทำงานผ่านอินเทอร์เน็ตการทำธุรกรรมทางโทรศัพท์มือถือ แอปพลิเคชันสำหรับองค์กรทั้งไคลเอ็นท์และเซิร์ฟเวอร์ เกิดขึ้นได้บนโทรศัพท์เคลื่อนที่และเพจเจอร์รับส่งข้อความ MIDP CLDC และ KVM ได้กลายเป็นรากฐานในการพัฒนาจาวาแอปพลิเคชันบนอุปกรณ์ไร้สายยุคใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลขเซอร์ฮาร์ดแวร์ MID หมายถึง ตัวโทรศัพท์เคลื่อนที่ (เช่น โมโตโรลา รุ่น iDEN 3000) หรือเพจเจอร์รับส่งข้อความ (เช่น RIM รุ่น Blzckberry 950)
- เลขเซอร์ซอฟต์แวร์ของระบบที่ติดตั้งมากับอุปกรณ์ หมายถึง ระบบปฏิบัติการและไลบรารีของระบบที่บริษัทผู้ผลิตให้มา
- เลขเซอร์ KVM เป็นส่วนที่เตรียม runtime environment ไว้ให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
- เลขเซอร์ CLDC เป็นส่วนที่เตรียม API หลักของจาวาให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
- เลขเซอร์ MIDP เป็นส่วนที่เตรียมไลบรารีสำหรับส่วนติดต่อกราฟฟิกกับผู้ใช้ และหน่วยเก็บข้อมูล persistent storage ระบบเครือข่าย และโหมดเมอร์

นอกจากคลาสไลบรารีสำหรับ MIDP อาจเตรียมคลาสไลบรารีเฉพาะอุปกรณ์ไว้ให้นักพัฒนา เพื่อดึงความสามารถของฟังก์ชันที่มีอยู่แล้วไปใช้งานให้เกิดประโยชน์สูงสุด อาทิ การโทรศัพท์ การแชร์ข้อมูลกับแอปพลิเคชันที่ติดตั้งมาในเครื่อง (เช่น ปฏิทิน สมุดจดที่อยู่) การตรวจสอบข้อมูลอุปกรณ์ที่ผู้ผลิตเตรียมไว้มาใช้งาน จะช่วยเพิ่มความสามารถแก่แอปพลิเคชันบนอุปกรณ์ไร้สาย แต่ไม่สามารถเคลื่อนย้ายไปสู่อุปกรณ์อื่นที่ใช้ MIDP ได้ เนื่องจากคลาสที่นำมาใช้ออกนอกเหนือขอบเขตของ MIDP

2.2.3 ความต้องการของระบบ

อุปกรณ์ไร้สายจะทำงานสนับสนุน J2ME ได้ดีเมื่อมีคุณสมบัติตรงตามข้อกำหนด หากต้องการให้ KVM ทำงานได้อย่างมีประสิทธิภาพ โดยใช้ไลบรารี CLDC จะต้องมีความสัมพันธ์ของระบบขั้นต่ำดังนี้

- มีหน่วยความจำ 160 - 512 กิโลไบต์ สำหรับสร้างแพลตฟอร์มจาวา
- มีหน่วยประมวลผลแบบ 16-32 บิตความเร็ว 25 เมกกะเฮิรซ์
- ใช้พลังงานน้อย โดยมากมักทำงานโดยใช้แบตเตอรี่
- เชื่อมต่อกับเครือข่ายได้ในช่วงสั้นๆ (อาศัยระบบไร้สายเป็นส่วนใหญ่) แบนด์วิดธ์ค่อนข้างจำกัด (ความเร็ว 9600 ไบต์ต่อวินาทีหรือน้อยกว่า)
- มีหน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวา รันไทม์และหน่วยความจำของออบเจกต์

2.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME

2.3.1 ไลบรารีของ CLCD

ไลบรารีของ CLDC เป็นไลบรารีของเครือข่ายและระบบชั้นสูงที่ไม่จำกัดเฉพาะอุปกรณ์ใด อุปกรณ์หนึ่ง ประกอบด้วยคลาส 2 ประเภท คือ คลาสที่เป็นซัพเซตของ J2SE และคลาสที่เกี่ยวข้องกับการติดต่อสื่อสารทั่วไป (Generic Connection Framework) ของ CLDC

2.3.2 ซับคลาสของ J2SE

คลาสประเภทแรกของ CLDC เป็นซับคลาสของไลบรารี J2SE กำหนดไว้ในแพ็คเกจ `java.lang`, `java.util` และ `java.io` โดยคลาสระบบและคลาสชนิดข้อมูลเข้ากันได้กับ J2SE และ J2EE และเพื่อให้เข้าเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันได้และเคลื่อนย้ายข้ามแพลตฟอร์มได้โดยสะดวก คลาสประเภทนี้จะใช้ชื่อแพ็คเกจตามชื่อคลาสอย่างเดียวกันใน J2SE หรือตามชื่อซัพคลาสของคลาสใน J2SE ที่เกี่ยวเนื่องกัน โดยไม่มีการเพิ่มเมธอด public หรือ protected หรือ ฟิวด์ซึ่งไม่มีในคลาสที่เกี่ยวข้องกันใน J2SE ความหมายของคลาสและเมธอดจึงไม่เปลี่ยนไป

2.3.3 คลาสที่มีเฉพาะใน CLDC

คลาสประเภทที่สองของ CLDC กำหนดไว้ในแพ็คเกจ javax.microedition.io คลาสเหล่านี้มีเฉพาะใน CLDC จึงไม่สามารถทำงานร่วมกับไลบรารีของ J2SE ได้ เป็นคลาสชั้นสูงเกี่ยวกับเรื่องทั่วไปของเครือข่าย กำหนดไว้ในแพ็คเกจ java.io และ java.net คลาสประเภทนี้จะเตรียมกรอบการติดต่อสื่อสารผ่านเครือข่ายให้กับอุปกรณ์ที่สนับสนุน J2ME โดยกรอบการสื่อสารนี้มักเรียกกันว่า Generic Connection Framework คลาสส่วนใหญ่ได้แก่อินเตอร์เฟซที่แทนการสื่อสารรูปแบบต่างๆ เช่น ซอกเกต (Socket) คำสั่งแกรม (Datagram) ซีเรียล (Serial) และ http ขึ้นอยู่กับว่าผู้ผลิตอุปกรณ์หรือผู้ให้บริการเครือข่ายจะติดตั้งอินเตอร์เฟซการเชื่อมต่อเหล่านี้เพียงบางตัวหรือครบทั้งหมด เมื่อพิจารณาจากความสามารถของอุปกรณ์และเครือข่าย

2.3.4 ประเภทของข้อมูลพื้นฐาน

CLDC สนับสนุนเฉพาะซัพเซตของชนิดข้อมูลที่นำมาจาก J2SE ซึ่งได้แก่ byte short int long char และ boolean แต่ไม่สนับสนุนข้อมูลประเภท float และ double เนื่องจากสาเหตุ 2 ประการ คือ อุปกรณ์เป้าหมายที่นำ CLDC ไปใช้ ส่วนใหญ่ไม่มีฮาร์ดแวร์รองรับตัวเลขทศนิยม และหากจะใช้ซอฟต์แวร์เข้าช่วยก็จะต้องเสียค่าใช้จ่ายสูงมาก

นอกจากนี้ CLDC ยังกำหนดคลาส type wrapper สำหรับข้อมูลทุกประเภท ได้แก่

- Java.lang.Boolean
- Java.lang.Byte
- Java.lang.Character
- Java.lang.Integer
- Java.lang.Long
- Java.lang.Short

คลาส type wrapper กำหนดไว้ในจาวา เนื่องจากจาวามีระบบย่อยหลายระบบซึ่งทำงานได้เฉพาะกับออบเจกต์เท่านั้นในกรณีนี้ คุณสามารถสร้างออบเจกต์โดยใช้คลาส wrapper ซึ่งเก็บชนิดของข้อมูลที่นำมาจาก J2SE เอาไว้

2.3.5 ไลบรารีของ MIDP

ขณะที่ไลบรารีของ CLDC ช่วยสร้างฟังก์ชันที่ไม่จำกัดเฉพาะอุปกรณ์บางชนิด ไลบรารีของ MIDP กลับตรงข้ามได้แก่ การจัดการแอปพลิเคชันบนอุปกรณ์ ส่วนติดต่อกราฟฟิกกับผู้ใช้ทั้งแบบพื้นฐานและซับซ้อน พื้นที่เก็บข้อมูลแบบคงตัว (persistent storage) และความสามารถเพิ่มเติมบนเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 คลาสโปรแกรมจัดการแอปพลิเคชัน

คลาสที่ติดต่อกับโปรแกรมจัดการแอปพลิเคชันบนอุปกรณ์ถูกกำหนดไว้ในแพ็คเกจ `java.microedition.midlet` แอปพลิเคชันทั้งหลายที่เขียนใน MIDP จะต้องขยายคลาส `MIDlet` ที่อยู่ในแพ็คเกจออกไป และจะต้องนำเมธอดทั้ง 3 ซึ่งได้แก่ `startApp()` `pauseApp()` และ `destroyApp()` เข้ามาใช้

2.3.7 คลาสของส่วนติดต่อกราฟิกกับผู้ใช้ (GUI Class)

ชุดเครื่องมือ `Abstract Windowing Toolkit` ใน J2SE ออกแบบมาเพื่อใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะและไม่สามารถนำมาใช้กับอุปกรณ์ไร้สายได้ เนื่องจากต้องการทรัพยากรประเภทหน่วยความจำค่อนข้างมาก MIDP มีวิธีการที่ต่างกันในการกำหนดไลบรารีสำหรับแพ็คเกจส่วนติดต่อกราฟิกกับผู้ใช้ ด้วยการใช้ API ชั้นสูงซึ่งเน้นความสามารถในการเคลื่อนย้ายข้ามอุปกรณ์ และ API ชั้นพื้นฐานซึ่งเน้นองค์ประกอบกราฟิกเฉพาะอุปกรณ์และ `input event` ทั่วไป คลาสที่เชื่อมต่อกับส่วนติดต่อกราฟิกกับผู้ใช้ และ `event-handling` กำหนดไว้ในแพ็คเกจ `java.microedition.lcdui` `Screen` ถือเป็นซูเปอร์คลาสของคอมโพเนนต์ส่วนติดต่อกับผู้ใช้โดยใช้ API ชั้นสูง ประกอบไปด้วย `Alert`, `Form`, `List`, `Textbox` ฯลฯ

`Canvas` และ `Graphic` เป็นคลาสหลักของ API ระดับต่ำ (Low-Level) แอปพลิเคชันเกมเป็นแอปพลิเคชันที่ใช้ API ระดับต่ำในการสร้างส่วนติดต่อกับผู้ใช้

2.3.8 คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)

ในบางครั้งแอปพลิเคชันที่เขียนบน MIDP จำเป็นจะต้องเก็บข้อมูลไว้บนอุปกรณ์อย่างถาวร คลาสที่กำหนดไว้ในแพ็คเกจ `java.microedition.rms` ให้กลไกการเก็บข้อมูลถาวรที่เรียกว่า `RecordStore` โดยยอมให้แอปพลิเคชันเขียนลบและปรับปรุงบันทึกข้อมูลในหน่วยเก็บข้อมูลถาวรในอุปกรณ์

2.3.9 คลาสของเครือข่าย (Network Class)

แม้ว่ากรอบการติดต่อสื่อสารทั่วไปที่กำหนดไว้ใน CLDC จะประกอบไปด้วยชุดของอินเทอร์เฟซการเชื่อมต่อเครือข่าย แต่ก็ไม่มีโพรโตคอลอยู่เบื้องหลังอินเทอร์เฟซการเชื่อมต่อจริงๆ หากแต่ปล่อยให้มันเป็นหน้าที่ของ MIDP

ในบรรดาอินเทอร์เฟซการเชื่อมต่อเครือข่ายเหล่านี้ `HttpConnection` ถือเป็นอินเทอร์เฟซหลักที่ต้องมีใน MIDP เสมอคลาสของอินเทอร์เฟซเหล่านี้ กำหนดไว้ในแพ็คเกจ `java.microedition.io`

2.4 MIDlet

`MIDlet` (“มิดเล็ต”) หมายถึง แอปพลิเคชันบน MIDP `MIDlet` มีส่วนคล้ายกับจาวาแอปเพล็ต แม้จะไม่มีเมธอด `main()` แต่ `MIDlet` ก็นำคลาส `javax.microedition.midlet.MIDlet` ตลอดจนเมธอดทั้ง 3 [`startApp()` `pauseApp()` และ `destroyApp()`] เข้ามาใช้ นอกจากนี้ `MIDlet` ยังกำหนด constructor แบบ `public` ที่ไม่มีอาร์กิวเมนต์ใด ๆ อีกด้วย

เราสามารถให้คำนิยามของคลาส `javax.microedition.midlet.MIDlet` ได้ดังนี้

เอกสารนี้เป็นเอกสารต้นฉบับของคลาส `MIDlet` extends `Object` เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

protected MIDlet ()
protected abstract void startApp () throws MIDletStateChangeException
protected abstract void pauseApp ()
protected abstract void destroyApp (boolean unconditional)
    throws MIDletStateChangeException
public final String getAppProperty (String key)
public final void notifyDestroyed ()
public final void notifyPaused ()
public final String getAppProperty (String key)
public final void resumeRequest ()
}

```

คลาส MIDlet ระบุเมธอดที่สามารถเรียกใช้โดยซอฟต์แวร์จัดการแอปพลิเคชัน (AMS) เพื่อสั่งให้แอปพลิเคชัน MIDlet เริ่มต้นและหยุดทำงาน

2.5 วงจรการทำงานของ MIDlet

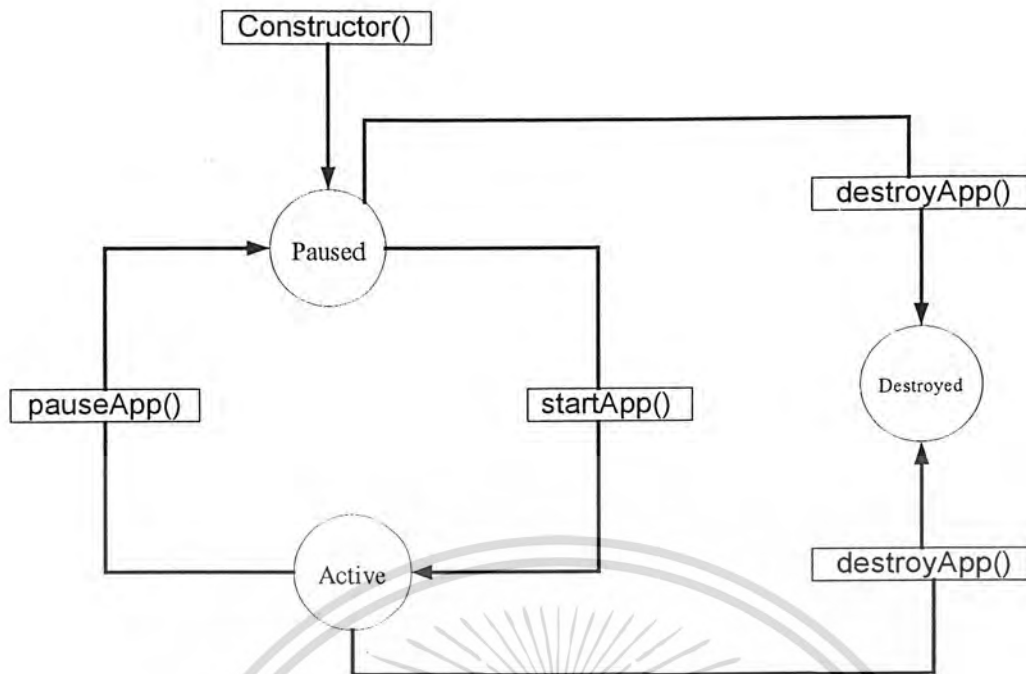
การกระทำของ MIDlet ประกอบไปด้วย 3 สถานะ คือ กำลังทำงาน หยุดชั่วคราวและถูกทำลาย ออสซีซอฟต์แวร์จัดการแอปพลิเคชันเป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่ง ควบคุมด้วยเมธอด StartApp () pauseApp() และ destroyApp()) ที่มาพร้อมกับ MIDlet รูปที่ 2-3 แสดงให้เห็นจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกใช้เมธอดข้างต้น

เมื่อ MIDlet พร้อมสั่งกระทำ ซอฟต์แวร์จัดการแอปพลิเคชันจะสร้างตัวอย่าง MIDlet ขึ้นมาก่อน โดยใช้ constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆ โดย MIDlet จะอยู่สถานะหยุดชั่วคราว

จากนั้นซอฟต์แวร์จัดการแอปพลิเคชันจะเรียกเมธอด startApp() ขึ้นมา และ MIDlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน

เมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการให้ MIDlet ทำงานต่อไป ก็จะเรียกเมธอด pauseApp() จากนั้น MIDlet จะหยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว คั้นทรัพยากรที่ดึงมาใช้งานและเข้าสู่ภาวะไม่ทำงาน MIDlet สามารถกลับไปอยู่ที่สถานะกำลังทำงาน ได้เมื่อซอฟต์แวร์จัดการแอปพลิเคชันเรียกเมธอด startApp() ขึ้นมาท้ายที่สุดเมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการเคลียร์หน่วยความจำเพื่อให้โปรแกรมอื่นได้ใช้งาน ก็จะส่งสัญญาณเตรียมทำลาย MIDlet ที่ด้วยการเรียกเมธอด destroyApp () และเข้าสู่สถานะถูกทำลาย ซึ่ง MIDlet จะปล่อยทรัพยากรทั้งหมด ทำการจัดเก็บข้อมูลถาวรต่างๆ และหยุดการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 วงจรการทำงานของ MIDlet

หาก MIDlet อยู่ระหว่างสักระยะทำการขึ้นคอนสำคัญอยู่ ก็อาจร้องขอไม่ให้เข้าสู่สถานะถูกทำลายได้โดยเรียกใช้ MIDletStateChangeException อย่างไรก็ตาม ซอฟต์แวร์จัดการแอปพลิเคชันอาจปฏิเสธหรือยินยอมตามคำร้องขอก็ได้ ตัวแปรบูลีน unconditional ในลายเซ็นของเมธอด destroyApp () จะเป็นตัวกำกับว่าการร้องขอนี้สมควรหรือไม่ ถ้าตัวบ่งชี้นี้มีค่าเท่ากับ true การร้องขอก็เป็นอันลุล่วง ตรงกันข้ามหากมีค่าเท่ากับ false ก็จะได้รับคำตอบรับและจะเรียกเมธอด destroyApp () ขึ้นมาใหม่ในคราวต่อไป

หาก MIDlet ต้องการเข้าสู่สถานะหยุดชั่วคราวหรือถูกสทำลายด้วยตัวเอง ก็สามารถทำได้โดยเรียกเมธอด notifyPause() หรือ notifyDestroy() ตามลำดับ โดยเมธอดทั้ง 2 จะแจ้งให้ซอฟต์แวร์จัดการแอปพลิเคชันทราบว่า MIDlet ได้เข้าสู่สถานะหยุดชั่วคราว/ถูกทำลายแล้ว ในกรณีนี้ซอฟต์แวร์จัดการแอปพลิเคชันจะไม่เรียกเมธอด pauseApp() หรือ destroyApp() อีก ทั้งนี้ MIDlet ควรจะล้างการใช้ทรัพยากรก่อนส่งสัญญาณให้ซอฟต์แวร์จัดการแอปพลิเคชันทราบ

MIDlet จะเข้าสู่สถานะหยุดทำงานได้ก็ต่อเมื่อผ่านสถานะกำลังทำงานมาก่อนแล้ว ในทางตรงกันข้าม หากต้องการเข้าสู่สถานะถูกทำลาย สามารถเข้าได้โดยตรงจากทั้งสถานะหยุดทำงานชั่วคราวและสถานะกำลังทำงาน นอกจากนี้ยังสามารถเข้าสู่สถานะหยุดการทำงานได้ในขณะที่กำลังทำงาน หรือเมื่อได้รับคำสั่งจากซอฟต์แวร์จัดการแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

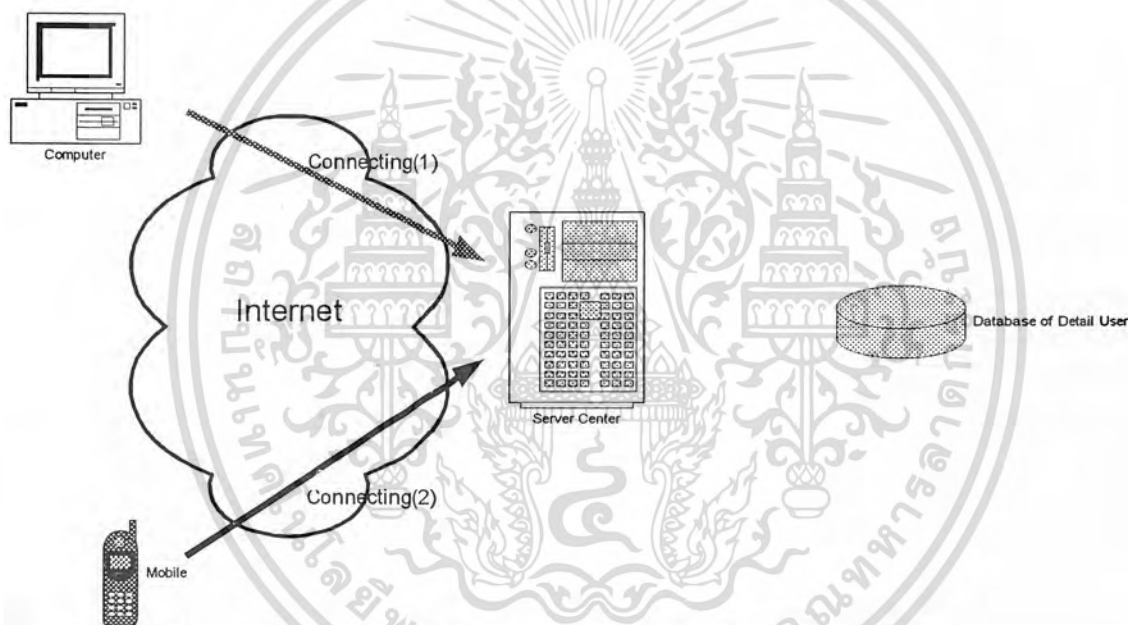
ภาพรวมของระบบ

3.1 การติดต่อโดยรวมของภายในระบบ

ภาพรวมการทำงานของระบบเชื่อมต่อระหว่างโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์และเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ จะแบ่งเป็นการทำงานหลักๆ ได้ดังนี้

3.1.1 การติดต่อกับเซิร์ฟเวอร์กลางเพื่อเข้าสู่ระบบ

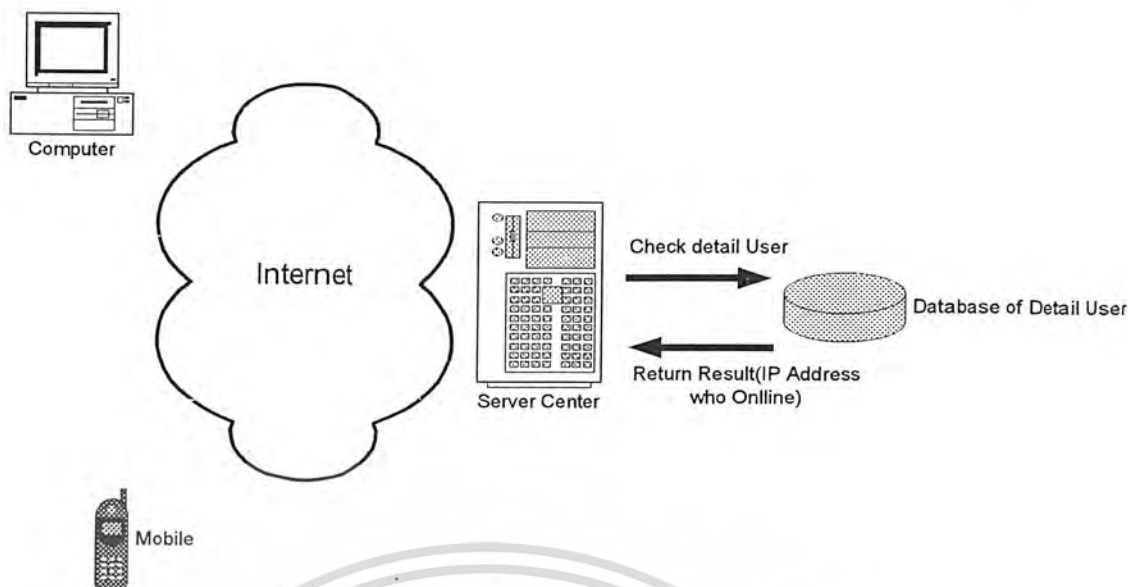
เมื่อไคลเอนต์ซึ่งคือ โทรศัพท์มือถือ หรือเครื่องคอมพิวเตอร์ส่วนบุคคลต้องการเข้าสู่ระบบ จะต้องทำการติดต่อ (Connection) ไปยังเซิร์ฟเวอร์กลางก่อนเพื่อยืนยันการเข้าสู่ระบบของไคลเอนต์เอง (Login) โดยไคลเอนต์จะต้องเคยลงทะเบียนกับระบบมาก่อนจึงจะเข้าสู่ระบบได้



รูปที่ 3-1 แสดงการติดต่อจากไคลเอนต์ไปยังเซิร์ฟเวอร์กลางเพื่อขอเข้าสู่ระบบ

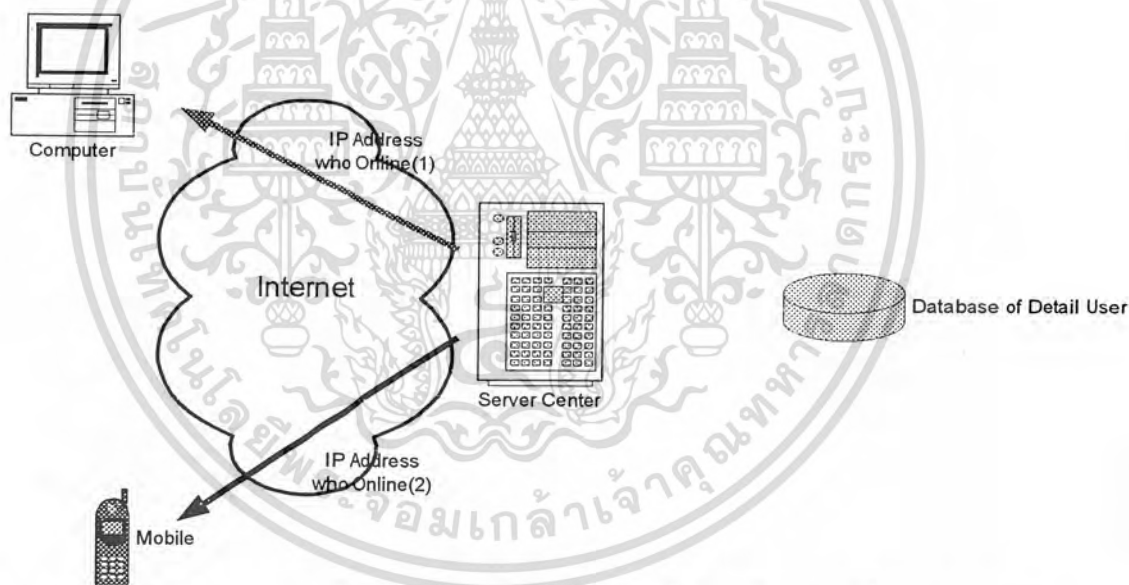
เซิร์ฟเวอร์หลักจะทำการตรวจสอบข้อมูลของไคลเอนต์โดยเปรียบเทียบกับข้อมูลที่ถูกลงทะเบียนไว้ในดาต้าเบส ถ้าชื่อของไคลเอนต์ที่ติดต่อเข้ามาถูกต้องก็จะทำการเปลี่ยนสเตตัส (status) ในข้อมูลของไคลเอนต์ที่ติดต่อเข้ามาจากออฟไลน์ (offline) เป็นออนไลน์ (online) พร้อมกับดึงไอพีแอดเดรสของเครื่องไคลเอนต์อื่นๆที่ออนไลน์อยู่ในขณะนั้นจากดาต้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-2 แสดงการตรวจสอบข้อมูลของไคลเอนต์ที่ทำการขอเข้าสู่ระบบ

เซิร์ฟเวอร์หลักจะทำการส่งไอพีแอดเดรส (IP Address) ของไคลเอนต์ที่ออนไลน์อยู่ในขณะนั้นกับไปยังไคลเอนต์ที่ติดต่อเข้ามา เพื่อให้ไคลเอนต์นำไปใช้ในการติดต่อไปยังไคลเอนต์อื่นๆ

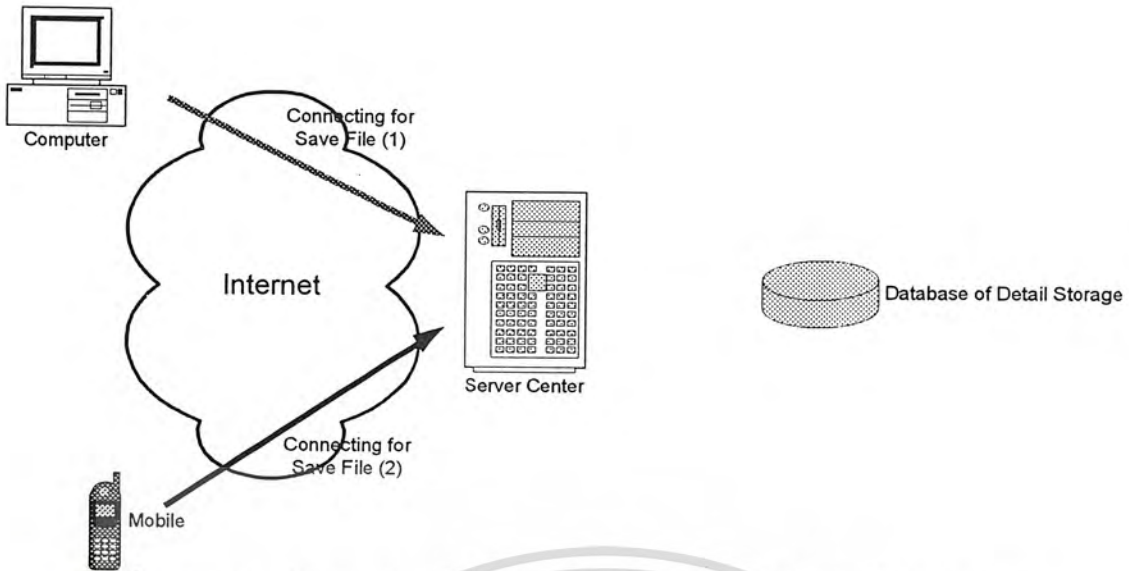


รูปที่ 3-3 แสดงการส่งไอพีแอดเดรสของไคลเอนต์ที่ออนไลน์อยู่ในขณะนั้นไปยังไคลเอนต์ที่ขอเข้าสู่ระบบ

3.1.2 การติดต่อกับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล

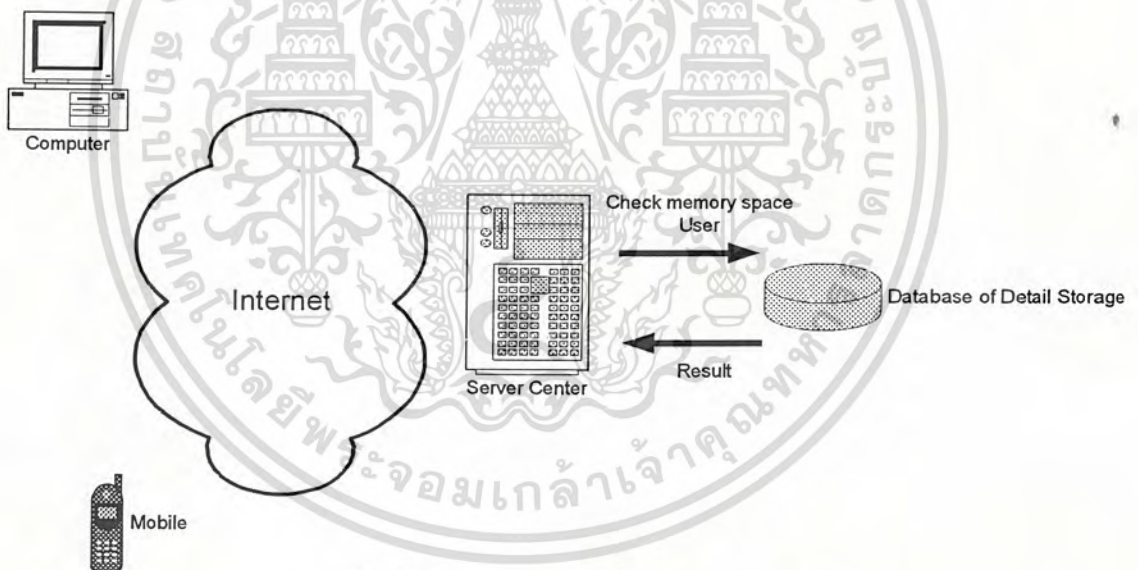
เมื่อไคลเอนต์ซึ่งคือโทรศัพท์มือถือ หรือเครื่องคอมพิวเตอร์ส่วนบุคคลต้องการจัดเก็บข้อมูล (data) ไว้ยังเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล ไคลเอนต์จะต้องทำการติดต่อพร้อมทั้งส่งข้อมูลไปยังเซิร์ฟเวอร์ที่ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 แสดงติดต่อกับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล

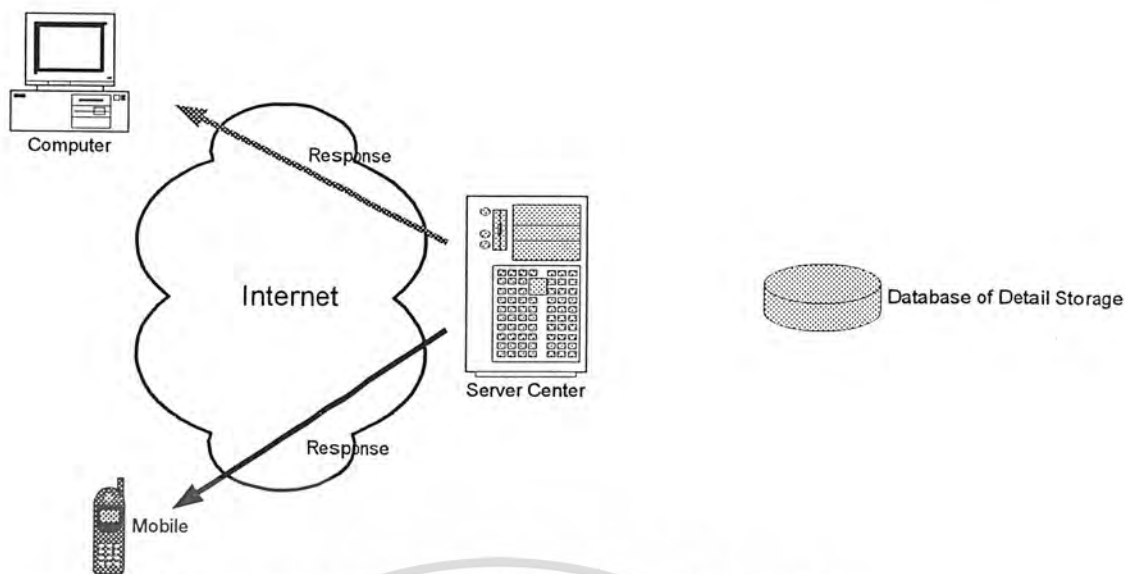
ซึ่งก่อนจะทำการจัดเก็บข้อมูลเซิร์ฟเวอร์กลางจะต้องทำการตรวจสอบพื้นที่ที่ให้บริการของไคลเอ็นต์นั้นๆก่อนว่ามีพื้นที่เพียงพอหรือไม่ในการจัดเก็บข้อมูล โดยขนาดของพื้นที่ที่ให้บริการของไคลเอ็นต์แต่ละคนจะขึ้นอยู่กับลำดับความสำคัญของไคลเอ็นต์เอง (authority)



รูปที่ 3-5 แสดงการตรวจสอบพื้นที่ที่ให้บริการของแต่ละไคลเอ็นต์ก่อนจัดเก็บข้อมูล

จากนั้นจึงส่งผลการจัดเก็บข้อมูลว่าสำเร็จหรือไม่ไปยังไคลเอ็นต์ที่ส่งข้อมูลมาจัดเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

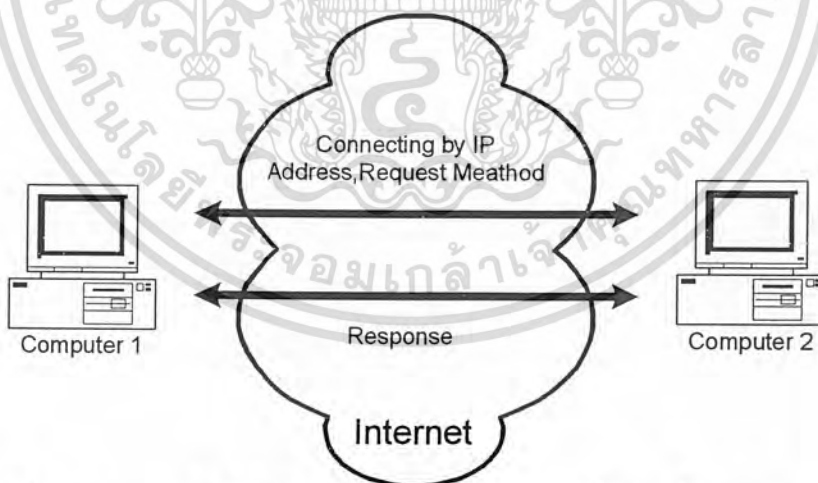


รูปที่ 3-6 แสดงการส่งผลการจัดเก็บข้อมูลกลับไปยังเครื่องไคลเอนต์

3.1.3 การทำงานของเครื่องไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล

ในกรณีที่ไคลเอนต์เป็นเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น เครื่องคอมพิวเตอร์แต่ละเครื่องจะทำงานคล้ายกับเซิร์ฟเวอร์ขนาดเล็กที่จะรอรับการติดต่อจากไคลเอนต์อื่นๆ และจะใช้ไอพีแอดเดรสในการติดต่อ โดยจะใช้การส่งข้อมูลไปในรูปของคำสั่งร้องขอ (Request Method) แบบต่างๆ ในการร้องขอใช้บริการจากไคลเอนต์อื่นๆ

เมื่อไคลเอนต์ทำตามคำสั่งร้องขอแล้ว จะส่งผลกลับไปในรูปแบบของคำตอบสนองแบบต่างๆ (Response) เพื่อแจ้งถึงผลการทำงานตามที่ถูกร้องขอมา



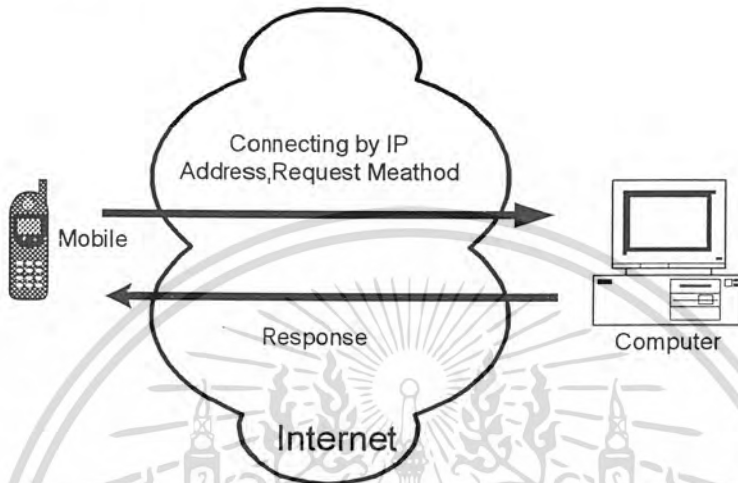
รูปที่ 3-7 แสดงการเรียกใช้บริการระหว่างเครื่องไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์

3.1.4 การทำงานของเครื่องไคลเอนต์ที่เป็นโทรศัพท์มือถือ

ในกรณีที่ไคลเอนต์เป็นโทรศัพท์มือถือ นั้น เนื่องจากโทรศัพท์มือถือแต่ละเครื่องมีข้อจำกัดด้านจำนวนหน่วยความจำและความสามารถในการประมวลผลทำให้ไม่สามารถทำงานในลักษณะของเซิร์ฟเวอร์ขนาดเล็ก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อรอรับการติดต่อจากไคลเอนต์อื่นๆ ไม่ได้เหมือนกรณีของเครื่องคอมพิวเตอร์ส่วนบุคคล ดังนั้นจึงทำได้แค่ติดต่อไปยังเครื่องไคลเอนต์อื่นๆ แต่จะมีหลักการการทำงานเหมือนกับกรณีของเครื่องคอมพิวเตอร์ส่วนบุคคล กล่าวคือใช้ไอทีแอดเดรสในการติดต่อ โดยจะใช้การส่งข้อมูลไปในรูปของคำสั่งร้องขอ (Request Method) แบบต่างๆ ในการร้องขอใช้บริการจากไคลเอนต์อื่นๆ

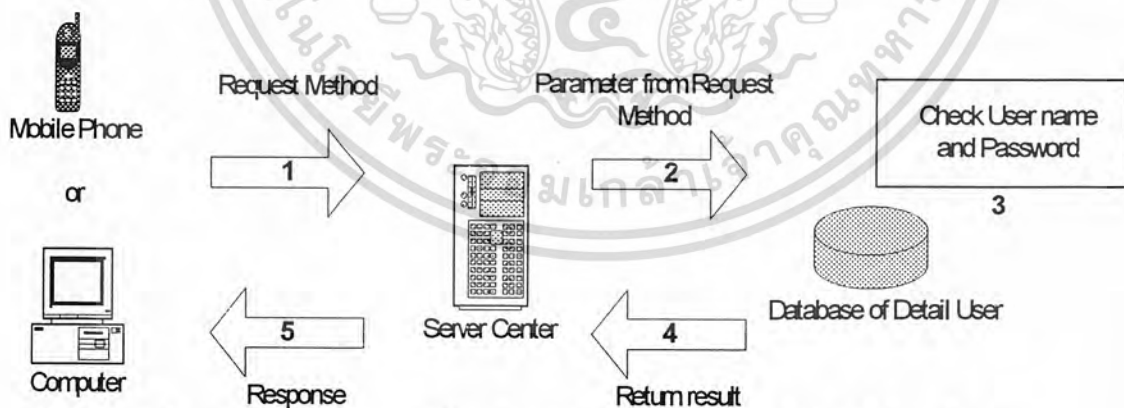
เมื่อไคลเอนต์ทำตามคำสั่งร้องขอแล้ว จะส่งผลกลับไปในรูปของผลตอบสนองแบบต่างๆ (Response) เพื่อแจ้งถึงผลการทำงานตามที่ถูกร้องขอมา



รูปที่ 3-8 แสดงการเรียกใช้บริการระหว่างเครื่องไคลเอนต์ที่เป็นโทรศัพท์มือถือกับเครื่องคอมพิวเตอร์

3.2 ไตอะแกรมขั้นตอนการติดต่อของไคลเอนต์ในระบบ

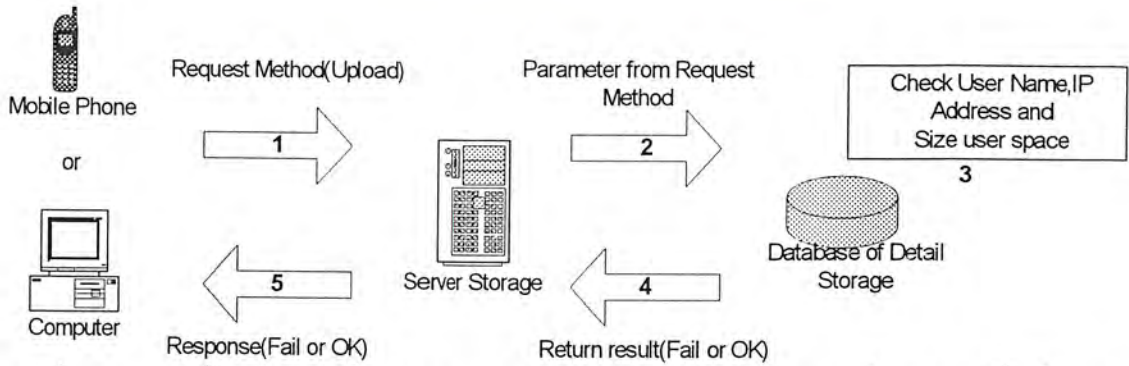
3.2.1 ไตอะแกรมแสดงขั้นตอนการติดต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์กลาง



รูปที่ 3-9 แสดงขั้นตอนการเชื่อมต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์กลาง

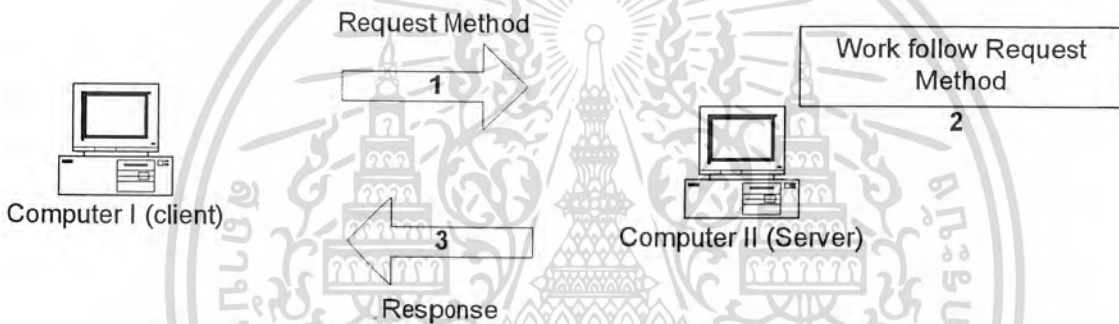
3.2.2 ไตอะแกรมแสดงขั้นตอนการติดต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



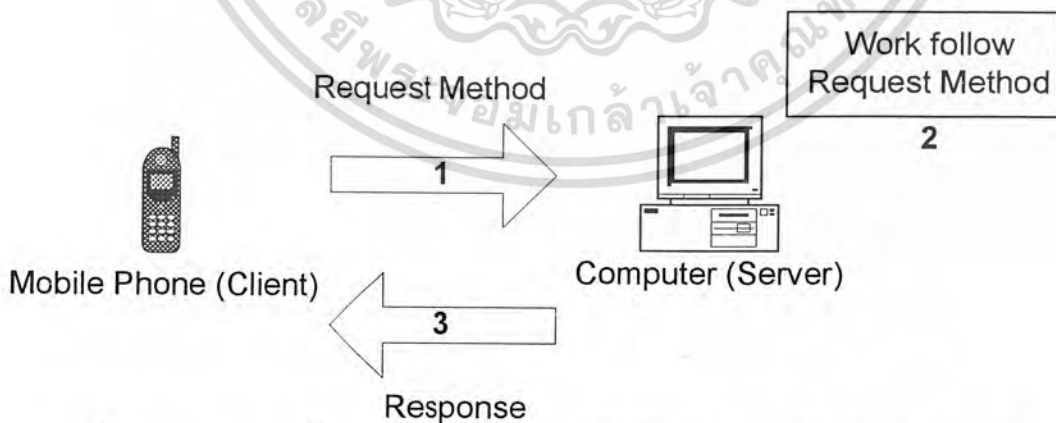
รูปที่ 3-10 แสดงขั้นตอนการเชื่อมต่อระหว่างเครื่องไคลเอนต์กับเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล

3.2.3 ไดอะแกรมแสดงขั้นตอนการติดต่อในกรณีที่เครื่องไคลเอนต์เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล



รูปที่ 3-11 แสดงขั้นตอนการเชื่อมต่อในกรณีที่เครื่องไคลเอนต์เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล

3.2.4 ไดอะแกรมแสดงขั้นตอนการติดต่อในกรณีที่เครื่องไคลเอนต์เป็นโทรศัพท์มือถือ



รูปที่ 3-12 แสดงแสดงขั้นตอนการติดต่อในกรณีที่เครื่องไคลเอนต์เป็นโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การเชื่อมต่อเครือข่าย โพรโทคอล และการออกแบบ

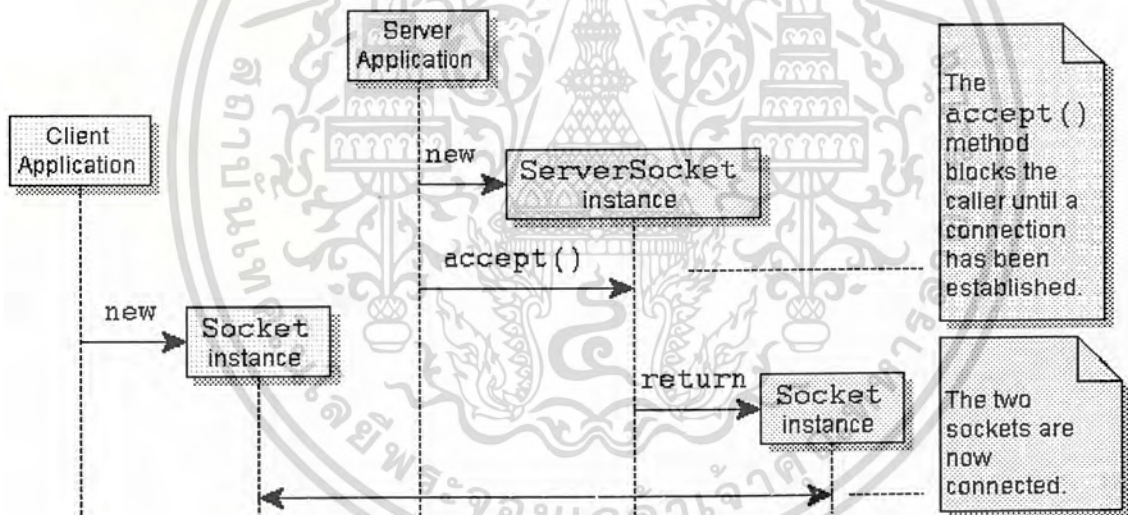
การติดต่อกันระหว่างเครื่องคอมพิวเตอร์สามารถทำได้ง่าย เมื่อติดต่อผ่านอินเทอร์เน็ต โดยใช้ IP address เป็นที่อยู่ที่ใช้ในการติดต่อ ทำให้มีความสะดวกขึ้นในการใช้งาน

4.1 รูปแบบการเชื่อมต่อ

การรับส่งข้อมูลผ่านเครือข่าย สามารถทำการสื่อสารกันระหว่างเครื่องกันได้ มี 2 แบบ คือ

4.1.1 การติดต่อบน TCP

การสื่อสารแบบนี้จะให้เซิร์ฟเวอร์เปิดพอร์ตรอรับไคลเอนต์ ส่วนไคลเอนต์ก็จะเปิดซอกเกต(socket) เพื่อติดต่อขอเชื่อมต่อกับเซิร์ฟเวอร์ ซอกเกตจะเชื่อมต่อไว้ตลอดเวลา (connection-oriented) จนกว่าจะไม่ต้องการรับส่งข้อมูลแล้ว ก็ไคลเอนต์จะต้องปิดการเชื่อมต่อซอกเกตลงเอง



รูปที่ 4-1 แสดงขั้นตอนการติดต่อกันผ่านซอกเกต

คำสั่งที่ใช้เปิดซอกเกต สำหรับ ไคลเอนต์ จะใช้ IP address และ พอร์ต

```
Socket connection = new Socket(IPaddress, 6000);
```

คำสั่งที่ใช้เปิดเซิร์ฟเวอร์ซอกเกต สำหรับ เซิร์ฟเวอร์

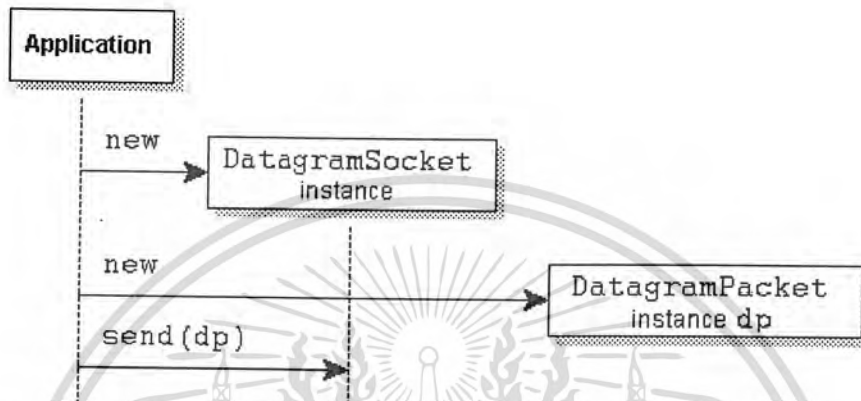
```
server = new ServerSocket( 6000 );
```

```
connection = server.accept();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 รับส่ง Datagram Packets ผ่าน UDP

การส่งข้อมูลแบบนี้ เซิร์ฟเวอร์จะเปิดพอร์ตรอรับเช่นกัน และไคลเอนต์จะสร้าง DatagramPacket เพื่อใส่ข้อมูลแล้วส่งออกไป จึงไม่จำเป็นต้องเชื่อมต่อกันตลอดเวลา (connectionless) จึงดีสำหรับการส่งข้อมูลไม่มาก และส่งไม่บ่อย แต่อาจจะไม่สามารถรับประกันได้ว่าเซิร์ฟเวอร์จะได้รับทุกแพ็คเกจ ที่ไคลเอนต์ส่งไปให้



รูปที่ 4-2 แสดงขั้นตอนการติดต่อกันผ่านUDP

คำสั่งที่ใช้ส่ง packet สำหรับไคลเอนต์

```

DatagramSocket connection = new DatagramSocket( 6000 );
DatagramPacket packet = new DatagramPacket( data , data.length);
Connection.send( packet );
  
```

คำสั่งที่ใช้เปิดพอร์ตรอรับ packet สำหรับเซิร์ฟเวอร์

```

DatagramSocket server = new DatagramSocket( 6000 );
Byte data[] = new byte[100];
rcvPacket = DatagramPacket( data , data.length )
server.receive( rcvPacket );
  
```

4.2 โพรโทคอล HTTP

HTTP (Hypertext Transfer Protocol) เป็นโพรโทคอลที่เชื่อมต่อกันแบบ Socket แล้วรอให้ไคลเอนต์ส่งคำร้องขอข้อมูลไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์ก็จะไปหาข้อมูลที่ไคลเอนต์ต้องการ ซึ่งไม่ว่าจะพบหรือไม่พบข้อมูลที่ข้อมูลก็ตามเซิร์ฟเวอร์จะต้องส่งข้อมูลตอบสนองกลับมาไคลเอนต์เสมอและสุดท้ายการเชื่อมต่อจะถูกตัดขาด หรือปลดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GET URI HTTP/1.0
User-Agent :
Accept :
Accept-Language :
Accept-Charset :
Cache-Control :
Message in block of data

รูปที่ 4-3 แสดงรูปแบบของ HTTP request

HTTP/1.0 200 OK
Date : Fri, 15 March 02.05.69
Content Length :
Content-Type :
Message in block of data

รูปที่ 4-4 แสดงรูปแบบของ HTTP response

มี headers ที่สำคัญๆ ดังนี้

- Date : วันที่เวลาที่ส่งถูกส่ง
- Server : ชื่อเซิร์ฟเวอร์
- Content type : ชื่อชนิดของไฟล์
- Content-length : ความยาวของข้อมูล
- Last modified : เวลาที่ถูกปรับปรุงครั้งสุดท้าย

4.3 การออกแบบการเชื่อมต่อ และ โพรโตคอลของระบบ

4.3.1 การเลือกรูปแบบการเชื่อมต่อที่จะใช้

ระบบนี้ต้องการการเชื่อมต่อที่สามารถรองรับการรับส่งข้อมูลขนาดใหญ่ที่เป็นไฟล์ได้ เพราะฉะนั้นความถูกต้อง และความน่าเชื่อถือก็มีความสำคัญพอสมควร ดังนั้นจึงต้องเปรียบเทียบว่าการเชื่อมต่อแบบไหนที่มีความเหมาะสมที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของแบบ socket

- มีการควบคุมการไหลของข้อมูลให้สมดุล ไม่เกิด Overflow
- มีความน่าเชื่อถือ เพราะ อยู่บนโพรโทคอล TCP ที่สามารถจะตรวจเช็คข้อมูลผิดพลาด หรือมีแพ็คเกจหายกลางทาง หรือไม่ ถ้าหากมีก็จะส่งส่วนที่ผิดพลาดนั้นกลับมาแก้ไข ได้เองอัตโนมัติ

ข้อเสียของแบบ socket

- กินทรัพยากรเพราะต้องเปิดชอกเกตค้างไว้ตลอดเวลาที่ใช้งาน ถึงแม้ว่าจะไม่มีการส่งข้อมูลระหว่างกัน

ข้อดีของแบบ UDP

- เหมาะกับการส่งข้อมูลขนาดเล็กๆ ที่ไม่มีความจำเป็นที่จะต้องตรวจสอบความถูกต้องมากนัก เช่น Chat

ข้อเสียของแบบ UDP

- ไม่มีการตรวจสอบข้อมูลที่ได้รับ จึงไม่เหมาะกับการส่งข้อมูลที่มีขนาดใหญ่อย่างไฟล์ และมีความสำคัญเพราะ มีความเสี่ยงพอสมควรที่แพ็คเกจที่ถูกส่งออกไปอาจจะสูญหายได้

จากข้อดีข้อเสียของการเชื่อมต่อทั้งสองแบบ สรุปได้ว่า Socket เหมาะสมระบบนี้มากกว่า เพราะระบบนี้เน้นที่ความถูกต้อง การรับส่งไฟล์ที่ไม่มีข้อผิดพลาดใดๆเกิดขึ้น

4.3.2 การออกแบบโพรโทคอล

จากการที่ศึกษา โพรโทคอลหลายๆแบบแล้ว HTTP เป็นที่มีความเหมาะสมที่จะนำมาเป็นต้นแบบที่จะนำมาพัฒนาเป็น โพรโทคอลของระบบเป็น FMP (File Manager Protocol) ซึ่งได้นำโครงสร้างของ HTTP มาเป็นแนวทางกำหนด

4.3.2.1 รูปแบบของฟอร์ม

FMP/1.0 Request Form ①	FMP/1.0 Response Form
Method :	Status :
User-Agent : Apps / MIDlet	User-Agent : Apps / MIDlet
Header	② Header
Path :	Path :
Name :	Name :
Type :	Type :
Length :	Length :
Parameter :	Parameter :
Body	③ Body
Byte[] data	Byte[] data

รูปที่ 4-5 แสดงรูปแบบ request และ response ของ FMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแบ่งได้เป็น 3 ส่วนหลักๆ คือ

1. Request หรือ Response Line จะเป็น บอกว่าเป็น Request หรือ Response และมี User agent และ method หรือ status มาด้วยเสมอ แบ่ง
2. Header ในส่วนนี้จะ เป็น optional (อาจจะมีหรือไม่มีขึ้นอยู่กับ method)
3. ข้อมูล ไฟล์ที่จะส่งไปให้อีกไคลเอนต์หนึ่ง

วิธีรวม Header ทั้งหมดมาเป็นแพ็คเกจจะทำโดยนำมาต่อกัน โดยคั่นแต่ละ Header ด้วย “\n” และแบ่ง Header กับ Body ด้วย “\n\n” แต่ไม่จำเป็นต้องตัด String เอง เพราะสามารถเรียก get method จาก คลาส FMP Request และ FMP Response ได้เลย

Method:upload	\n	User-Agent:MIDlet	\n
Name:doc.html	\n	MIMEtype:text/html	\n
Length:3409	\n	LastModified:15Mar200316:23:02	\n
\n	Byte[] data	\n	

รูปที่ 4-6 แสดงตัวอย่างของรูปแบบในการร้องขอเป็นแพ็คเกจ

4.3.2.2 FMP Header

- User-Agent บอก agent ที่ไคลเอนต์ใช้เป็นแบบ Application (เครื่องคอมพิวเตอร์) หรือ MIDlet (โทรศัพท์มือถือ)
- Path : ใช้ในการอ้างที่อยู่ของไฟล์ หรือ ชื่อ โฟลเดอร์
- Name : ชื่อไฟล์
- MIMEtype : ชื่อชนิดของ ไฟล์สำหรับเก็บข้อมูลบน โทรศัพท์มือถือ
- Length : ความยาวของไฟล์ที่แนบมาด้วย
- Parameter : ตัวแปรเพิ่มเติมตามที่ request หรือ response ต้องการ
- Method : คำสั่งที่ใช้ร้องขอให้เซิร์ฟเวอร์ทำ
- Status : ผลแสดงสถานะจากการทำงาน request method นั้นว่าสำเร็จ หรือไม่ หากทำไม่สำเร็จ ก็จะมีข้อความให้แสดงผลว่าการทำงานผิดพลาดที่ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Headers	Request	Response
User-agent	✓	✓
Method	✓	
Path	✓	✓
Name	✓	✓
MIME type	✓	✓
Length	✓	✓
Parameter	✓	✓
Status		✓

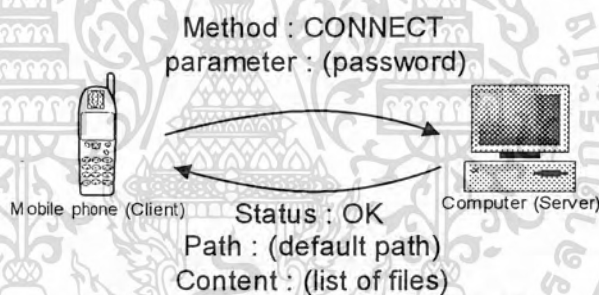
รูปที่ 4-7 แสดง Header ทั้งหมดของ FMP

4.3.2.3 FMP Method

Request Method มีการใช้งานขอบริการต่างๆ ดังนี้

- การขอติดต่อไคลเอนต์อื่น (connect)

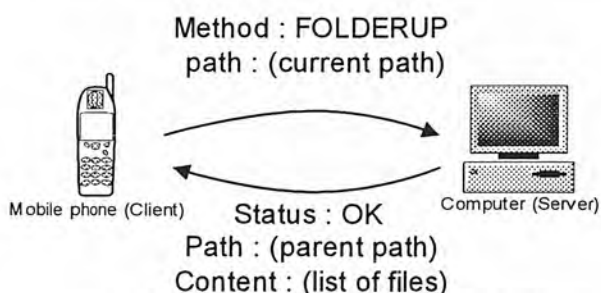
method connect ขอใช้บริการเซิร์ฟเวอร์ โดยใช้ IP address ที่ได้จากเซิร์ฟเวอร์กลางมาใช้ แล้วเปิด ซอกเก็ตติดต่อกัน โดยเซิร์ฟเวอร์จะถามหาพาสเวิร์ด ถ้าถูกต้องจะส่งรายชื่อไฟล์ในโฟลเดอร์ที่เซิร์ฟเวอร์กลับมาให้



รูปที่ 4-8 แสดงการขอติดต่อกับไคลเอนต์อื่น

- การออกไปยังโฟลเดอร์แม่ (Folder Up)

ไคลเอนต์จะส่ง request folderup พร้อมกับไครกทอรีปัจจุบันที่เปิดอยู่ไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปให้เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

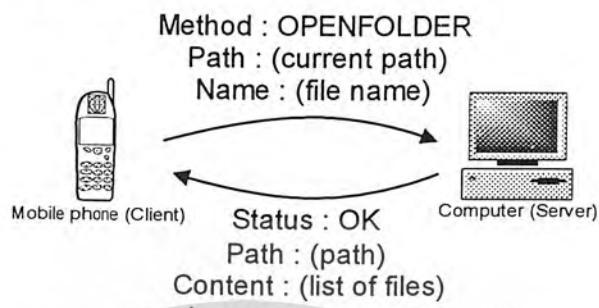


รูปที่ 4-9 แสดงการออกไปโฟลเดอร์แม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในระบบเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเปิดโฟลเดอร์ (Open Folder)

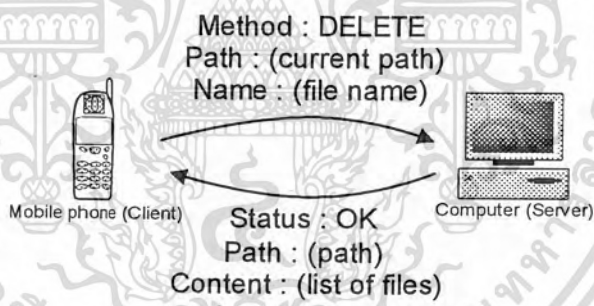
ไคลเอนต์จะส่ง request openfolder พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่และ ชื่อโฟลเดอร์ที่ต้องการเปิดไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่



รูปที่ 4-10 แสดงการเปิดโฟลเดอร์

- การลบไฟล์ (Delete File)

ไคลเอนต์จะส่ง request delete พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่และ ชื่อไฟล์ที่ต้องการจะลบออก เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

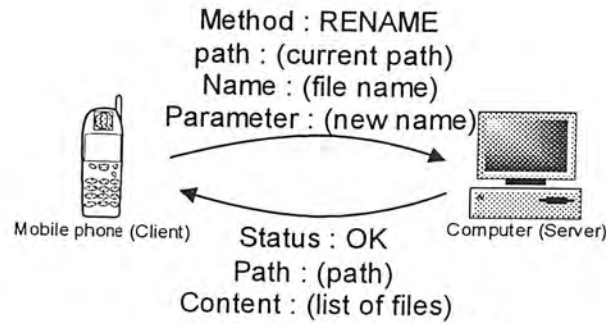


รูปที่ 4-11 แสดงการลบไฟล์

- การเปลี่ยนชื่อ (Rename)

ไคลเอนต์จะส่ง request rename พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ ชื่อไฟล์ที่ต้องการจะเปลี่ยนชื่อ และ ชื่อไฟล์ใหม่ไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

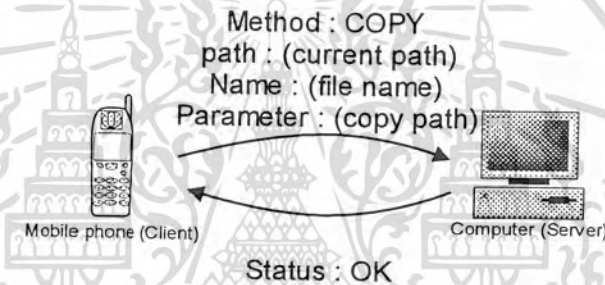
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-12 แสดงการเปลี่ยนชื่อไฟล์

- การคัดลอกไฟล์ (Copy)

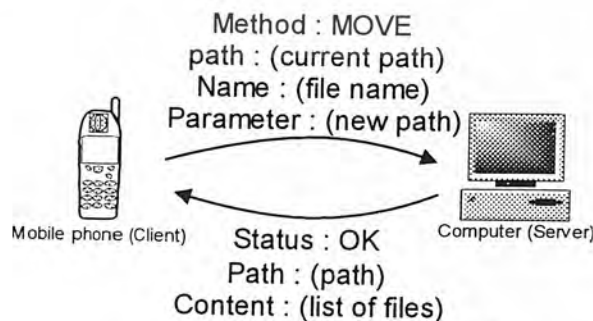
ไคลเอนต์จะส่ง request copy พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ ชื่อไฟล์ที่ต้องการจะคัดลอก และ ชื่อพาธใหม่ที่ต้องการจะวางไปให้ เซิร์ฟเวอร์จะส่งสถานะว่าทำงานได้สำเร็จหรือไม่



รูปที่ 4-13 แสดงการคัดลอกไฟล์

- การย้ายไฟล์ (Move)

ไคลเอนต์จะส่ง request move พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ ชื่อไฟล์ที่ต้องการจะย้าย และ ชื่อพาธใหม่ที่ต้องการจะวางไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปให้ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

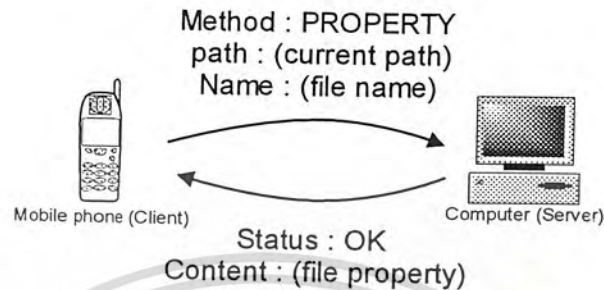


รูปที่ 4-14 แสดงการคัดลอกไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การดูคุณสมบัติของไฟล์ (Property)

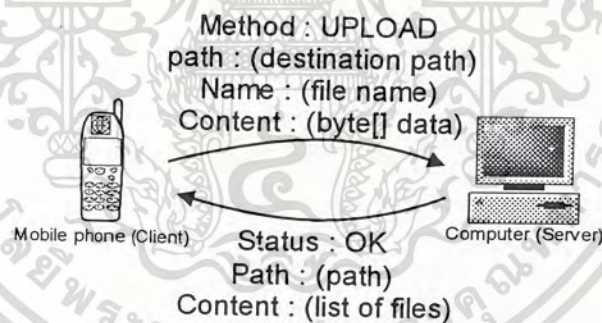
ไคลเอนต์จะส่ง request property พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ และชื่อไฟล์ที่ต้องการจะดูคุณสมบัติไปให้ เซิร์ฟเวอร์จะส่งคุณสมบัติของไฟล์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่



รูปที่ 4-15 แสดงการดูคุณสมบัติของไฟล์

- การส่งไฟล์ให้เซิร์ฟเวอร์ (Upload)

ไคลเอนต์จะส่ง request upload พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ของเซิร์ฟเวอร์ ชื่อไฟล์ที่ต้องการจะส่งไป และ ไฟล์ส่งไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

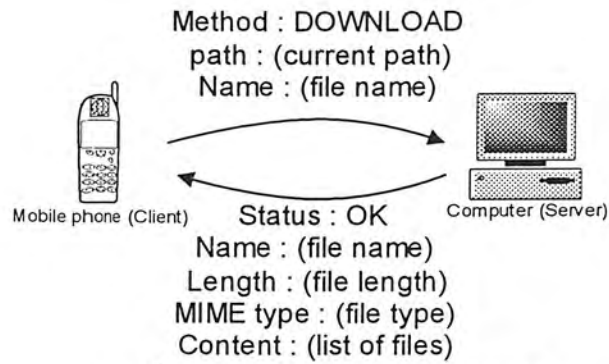


รูปที่ 4-16 แสดงการส่งไฟล์ให้เซิร์ฟเวอร์

- การรับไฟล์จากเซิร์ฟเวอร์ (Download)

ไคลเอนต์จะส่ง request download พร้อมกับไคเรกทอรีปัจจุบันที่เปิดอยู่ของเซิร์ฟเวอร์ ชื่อไฟล์ที่ต้องการ และ ชื่อพารามิเตอร์เซิร์ฟเวอร์ที่ต้องการจะวางไปให้ เซิร์ฟเวอร์จะส่งรายชื่อของไฟล์ในโฟลเดอร์นั้นกลับไปที่ เพื่ออัปเดตจากรายชื่อที่แสดงอยู่

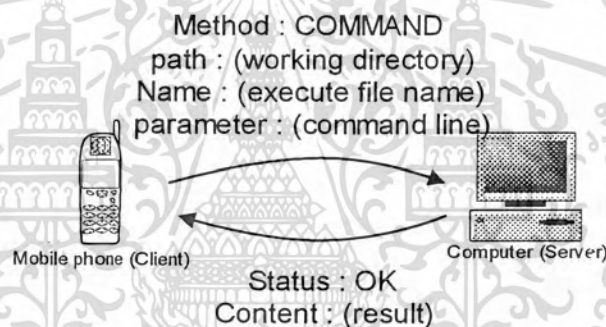
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-17 แสดงการรับไฟล์ให้เซิร์ฟเวอร์

- การรันคอมมานด์ไลน์ (command)

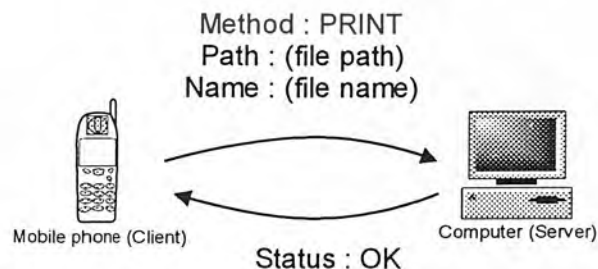
ไคลเอนต์จะส่ง request command พร้อมกับไดเรกทอรีที่ใช้รัน ชื่อไฟล์ที่ต้องการรัน หรือ คอมมานด์จากผู้ใช้โดยตรง เซิร์ฟเวอร์จะส่งผลการรันที่เป็น Output Stream ของ Process กลับไปให้ เพื่อแสดงผล



รูปที่ 4-18 แสดงการรับไฟล์จากเซิร์ฟเวอร์

- การสั่งพิมพ์ (print)

ไคลเอนต์จะส่ง request print พร้อมกับไดเรกทอรีที่ใช้รัน ชื่อไฟล์ที่ต้องการพิมพ์ เซิร์ฟเวอร์จะส่งผลการดำเนินงานกลับมาว่าสำเร็จหรือไม่

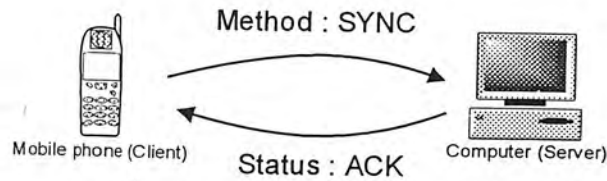


รูปที่ 4-19 แสดงการรับไฟล์จากเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การซิงโครไนซ์ (sync)

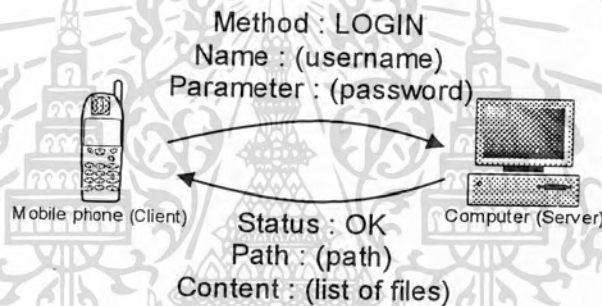
ไคลเอนต์จะส่ง request sync ไป เซิร์ฟเวอร์จะตอบ acknowledge กลับมา เมื่อนี้จะใช้สำหรับโทรศัพท์มือถือเท่านั้น เพื่อให้ไม่ให้เกิดการเชื่อมต่อก่อนถ้าไม่มีการส่งข้อมูลระหว่างกัน



รูปที่ 4-20 แสดงการซิงโครไนซ์กับเซิร์ฟเวอร์

- การเข้าระบบ (Login)

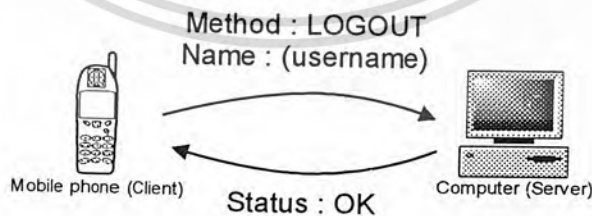
ไคลเอนต์จะส่ง request login ไป เซิร์ฟเวอร์จะตอบกลับพร้อมกับรายชื่อของไฟล์ ถ้าหาก username และ password ถูกต้อง



รูปที่ 4-17 แสดงการเข้าระบบ

- การออกจากระบบ (Logout)

ไคลเอนต์จะส่ง request logout ไป พร้อมกับชื่อผู้ใช้ เซิร์ฟเวอร์จะตอบรับ ถ้าผู้ใช้ออกจากระบบสำเร็จ

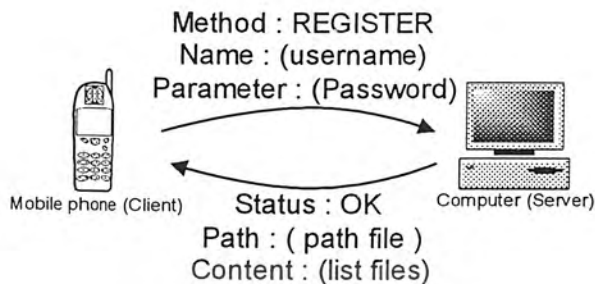


รูปที่ 4-18 แสดงการออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การลงทะเบียน (Register)

ไคลเอนต์จะส่ง request register ไป เซิร์ฟเวอร์จะตอบรับ ถ้าผู้ใช้ลงทะเบียนกับระบบได้สำเร็จ



รูปที่ 4-19 แสดงการลงทะเบียน

4.3.3 FMP Status

มีสถานะที่เซิร์ฟเวอร์จะตอบกลับมา 3 แบบ ได้แก่

4.3.3.1 ตอรับ (OK)

คำสั่งที่ได้ส่ง request มาร้องขอให้ทำงาน สามารถทำได้สำเร็จ ไม่มีควมผิดพลาดใดๆ และสามารถให้บริการต่อไปได้อย่างปกติ

4.3.3.2 ล้มเหลว (FAIL)

คำสั่งที่ได้ส่ง request มาร้องขอให้ทำงาน ไม่สามารถทำได้ เกิดความผิดพลาดบางอย่าง และอาจจะไม่สามารถให้บริการได้ต่อ มีข้อความแสดงบอกความผิดพลาดที่เกิดขึ้นส่งกลับไปด้วย

4.3.3.3 ตอบสนอง (ACK)

ใช้ในการตอบสนองต่อ SYNC เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

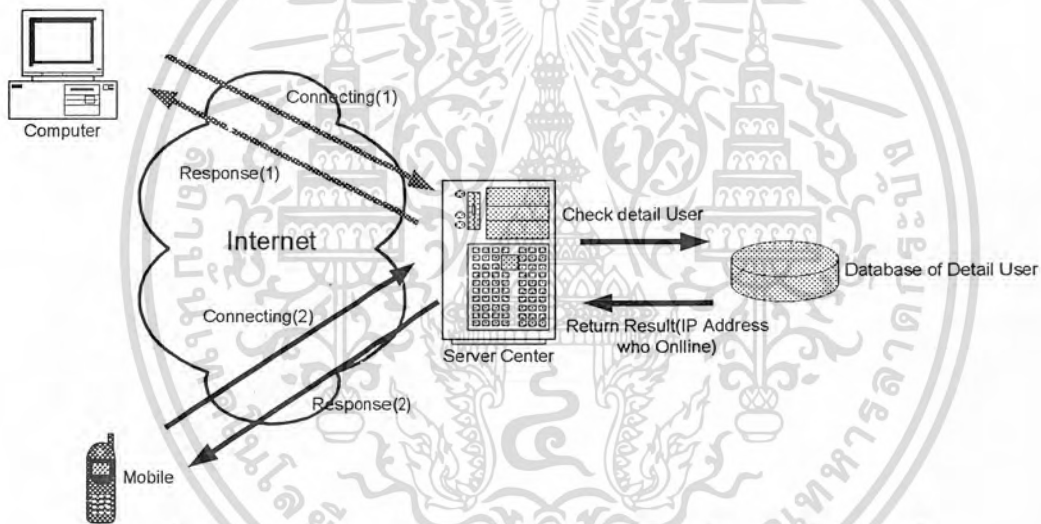
บทที่ 5

เซิร์ฟเวอร์

ในการพัฒนาระบบเชื่อมต่อระหว่างโทรศัพท์มือถือ กับเครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์นั้น ได้สร้างโปรแกรมเพื่อรองรับการติดต่อจากไคลเอนต์ ซึ่งจะมีการทำงานที่คล้ายคลึงกับเซิร์ฟเวอร์ โดยแบ่งออกเป็น

5.1 เซิร์ฟเวอร์หลัก

เป็นเซิร์ฟเวอร์ที่จะรองรับการติดต่อจากไคลเอนต์ที่ต้องการร้องขอเข้าสู่ระบบ ร้องขอออกจากระบบ และร้องขอลงทะเบียนเข้าสู่ระบบ ซึ่งทำให้เซิร์ฟเวอร์หลักนี้จะต้องทำการออกแบบฐานข้อมูลเพื่อใช้เก็บข้อมูลของยูสเซอร์แต่ละคน



รูปที่ 5.1 แสดงการติดต่อโดยรวมระหว่างเซิร์ฟเวอร์หลัก กับไคลเอนต์

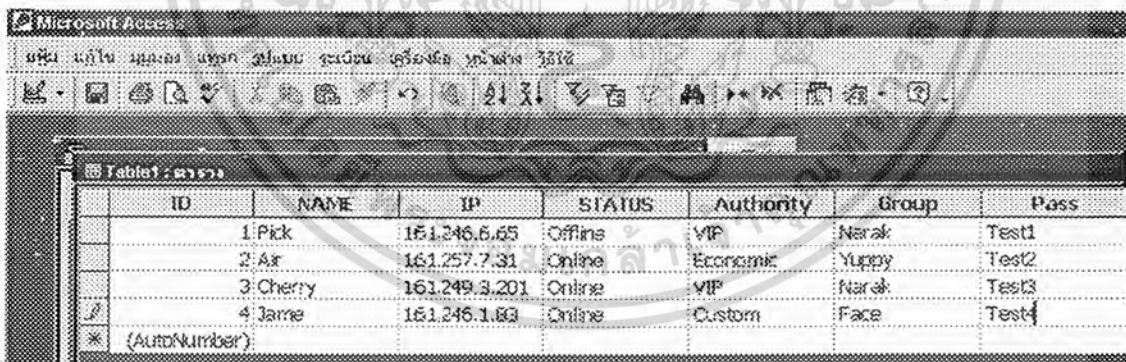
5.1.1 การออกแบบฐานข้อมูลของเซิร์ฟเวอร์หลัก

เนื่องจากเซิร์ฟเวอร์หลักต้องเก็บข้อมูลของยูสเซอร์แต่ละคน ซึ่งข้อมูลแต่ละอันจะมีผลต่อการเรียกใช้บริการต่างๆของระบบ ดังนั้นจึงต้องมีการออกแบบฐานข้อมูลให้ครอบคลุมประเภทของข้อมูลที่จะต้องถูกเรียกใช้ในบริการต่างๆให้มากที่สุด โดยในโครงการนี้เลือกใช้ Microsoft Access ในการสร้างฐานข้อมูล ซึ่งจะแบ่งออกเป็นแต่ละคอลัมน์ (culum) ดังนี้

- ID เป็นคอลัมน์ที่กำหนดให้เป็น primary key ซึ่งจะทำการเพิ่มค่าอัตโนมัติทุกครั้งที่มีการสร้าง row ของยูสเซอร์ใหม่
- NAME เป็นคอลัมน์ที่จะเก็บยูสเซอร์เนม (User name) ของยูสเซอร์ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำออกเผยแพร่ไปใช้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังอาจมีการเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้เข้าสู่ระบบ

- IP เป็นคอลัมน์ที่จะเก็บไอพีแอสเครสของยูสเซอร์เนมแต่ละอันที่ทำการติดต่อเข้ามาเพื่อร้องขอเข้าสู่ระบบ ในกรณีที่เป็นยูสเซอร์เนมใหม่ที่เพิ่งทำการลงทะเบียนเพื่อเข้าสู่ระบบ ค่าในคอลัมน์นี้จะกำหนดค่าไว้ให้เป็น 161.246.6.300 ซึ่งเป็นหมายเลขที่เกินกว่าที่ใช้กันในระบบทั่วไปในปัจจุบัน
- STATUS เป็นคอลัมน์ที่จะเก็บสถานะการเข้าสู่ระบบของยูสเซอร์เนมแต่ละอัน โดยจะเปลี่ยนจาก Offline ไปเป็น Online เมื่อยูสเซอร์เนมนั้นร้องขอเข้าสู่ระบบ และจะเปลี่ยนจาก Online ไปเป็น Offline เมื่อยูสเซอร์เนมนั้นทำการร้องขอออกจากระบบ
- Authority เป็นคอลัมน์ที่จะเก็บลำดับค่าของสำคัญของยูสเซอร์เนมแต่ละอัน โดยในระบบที่พัฒนาขึ้นมาจะมีผลในด้านของการเรียกใช้บริการพื้นที่ในการสำรองข้อมูลเพื่อจัดเก็บข้อมูล กล่าวคือ Authority ที่แตกต่างกันก็จะมีขนาดของพื้นที่ที่ให้บริการแตกต่างกัน
- GROUP เป็นคอลัมน์ที่จะเก็บชื่อของกลุ่ม (group) ซึ่งยูสเซอร์แต่ละคนสามารถกำหนดยูสเซอร์เนมได้เองในขั้นตอนการลงทะเบียนเพื่อขอเข้าสู่ระบบ โดยในระบบที่พัฒนาขึ้นมาจะมีผลในขั้นตอนการส่งไอพีแอสเครสของยูสเซอร์อื่นๆที่อยู่ในระบบขณะนั้น กล่าวคือถ้ายูสเซอร์มีกรุปไอพีแอสเครสที่ส่งไปให้ก็จะเป็น ไอพีแอสเครสของยูสเซอร์ที่มีกรุปชื่อเดียวกัน แต่ในกรณีที่ยูสเซอร์ไม่มีกรุปไอพีแอสเครสที่จะส่งไปให้ก็จะเป็น ไอพีแอสเครสของยูสเซอร์ที่ไม่มีกรุปเช่นเดียวกัน



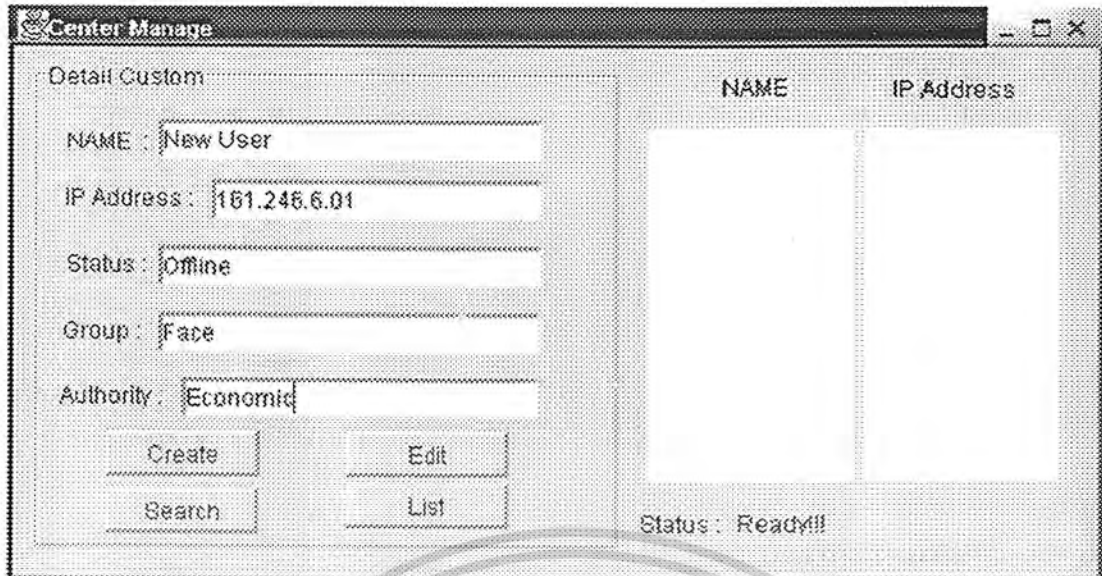
ID	NAME	IP	STATUS	Authority	Group	Pass
1	Pick	161.246.6.65	Offline	VIP	Narak	Test1
2	Ax	161.257.7.31	Online	Economic	Yuppy	Test2
3	Cherry	161.249.3.201	Online	VIP	Narak	Test3
4	Jame	161.246.1.83	Online	Custom	Face	Test4

รูปที่ 5-2 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลัก

5.1.2 โปรแกรมช่วยจัดการข้อมูลบนฐานข้อมูล

เนื่องจากโครงการนี้เลือกใช้ Microsoft Access ในการสร้างฐานข้อมูล ทำให้ผู้ที่ทำการแก้ไข หรือเพิ่มเติมข้อมูลในฐานข้อมูลนั้นจะต้องมีความรู้ในการใช้งาน Microsoft Access ในระดับหนึ่ง เพื่อให้สะดวกกับการใช้งานการพัฒนาระบบนี้จึงสร้างโปรแกรมขนาดเล็กขึ้นมาเพื่อจัด

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-3 แสดงยูสเซอร์อินเทอร์เน็ตเฟซของโปรแกรมช่วยจัดการบนฐานข้อมูล

โดยมีการทำงานหลักๆดังต่อไปนี้

5.1.2.1 การสร้างข้อมูลของยูสเซอร์ใหม่ที่ทำกรลงทะเบียนเพื่อเข้าสู่ระบบ

เมื่อทำการคลิกที่ปุ่ม Create โปรแกรมจะทำการสร้างข้อมูลของยูสเซอร์ลงบนฐานข้อมูล โดยนำค่าพารามิเตอร์จากยูสเซอร์อินเทอร์เน็ตเฟซ (User interface) ไปเป็นข้อมูลในคอลัมน์ต่างๆ ด้วยคำสั่ง SQL ดังนี้

```
String sql = "INSERT INTO Server_DB(NAME,IP,STATUS,Authority,[GROUP])" (1)
```

```
VALUES(""+NameT+"",""+IpT+"','Online','"+AuthorityT+"','"+GroupT+"");
```

```
stmt.executeUpdate(sql); (2)
```

ในบรรทัดที่ (3) เป็นคำสั่ง SQL ที่จะทำการเพิ่ม row ลงไปยังฐานข้อมูลโดย NameT,IpT,AuthorityT,GroupT เป็นค่าพารามิเตอร์ที่รับมาจากยูสเซอร์อินเทอร์เน็ตเฟซ ถ้าทำการสร้างข้อมูลของยูสเซอร์สำเร็จ แถบสถานะบาร์จะปรากฏข้อความ "Create Complete..." แต่ถ้าสร้างไม่สำเร็จจะปรากฏข้อความ "User name Exist..."

The screenshot shows a Microsoft Access window displaying a table named "Table1: users". The table has the following columns: ID, NAME, IP, STATUS, Authority, Group, and Pass. The data in the table is as follows:

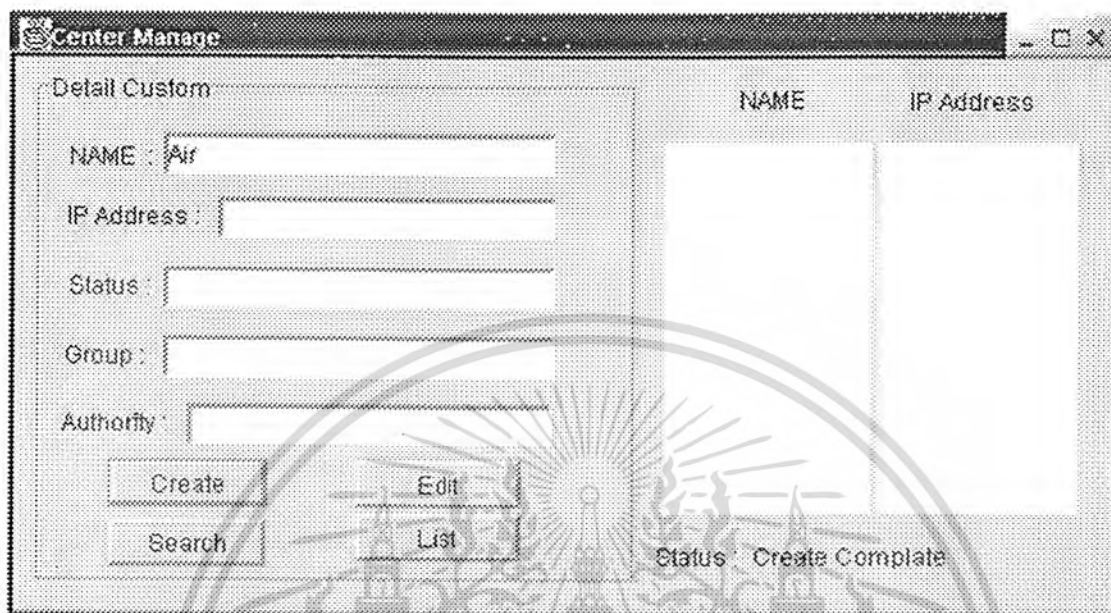
ID	NAME	IP	STATUS	Authority	Group	Pass
1	Pick	161.246.6.65	Offline	VIP	Narak	Test1
2	Air	161.257.7.31	Online	Economic	Yuppy	Test2
3	Cherry	161.249.3.201	Online	VIP	Narak	Test3
4	Jame	161.246.1.83	Online	Custom	Face	Test4
5	New User	161.246.6.01	Online	Economic	Face	

รูปที่ 5-4 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลัก

หลังจากทำการ create ยูสเซอร์ใหม่

5.1.2.2 การค้นหาข้อมูลของยูสเซอร์ที่ทำการลงทะเบียนแล้ว

การเรียกใช้การทำงานการค้นหาข้อมูลของยูสเซอร์ที่ทำการลงทะเบียนแล้วนั้น จะต้องใส่ยูสเซอร์เนมของยูสเซอร์ที่ต้องการจะค้นหาแล้วจึงคลิกที่ปุ่ม Edit



รูปที่ 5-5 แสดงการกรอกยูสเซอร์เนมเพื่อค้นหาข้อมูล

โปรแกรมจะทำการค้นหาข้อมูลของยูสเซอร์จากฐานข้อมูล โดยนำค่าพารามิเตอร์จากยูสเซอร์อินเทอร์เฟซ (User interface) ไปเป็นข้อมูลในการค้นหาด้วยคำสั่ง SQL ต่อไปนี้

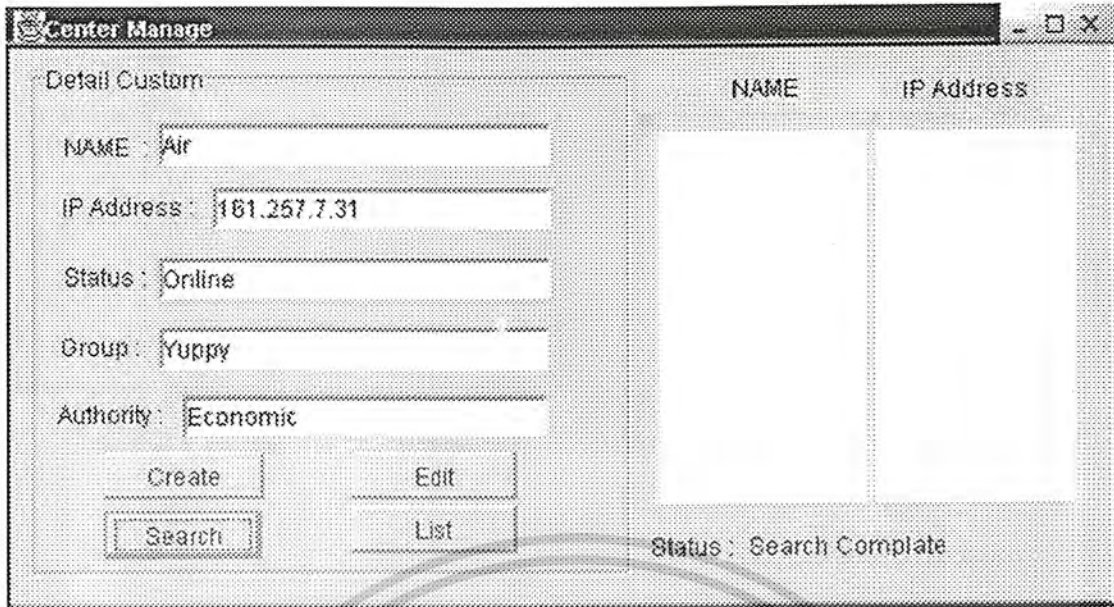
```
String sql = "Select * FROM Server_DB" (1)
```

```
WHERE NAME = "+NameT+" "; (2)
```

```
l= stmt.executeQuery(sql);
```

ในบรรทัดที่ (3) เป็นคำสั่ง SQL ที่จะทำการค้นหา row ของยูสเซอร์ที่ต้องการจากฐานข้อมูลโดย NameT เป็นค่าพารามิเตอร์ที่รับมาจากยูสเซอร์อินเทอร์เฟซเพื่อใช้เป็นข้อมูลในการหา row ที่มียูสเซอร์เนมเดียวกัน ถ้าโปรแกรมทำการค้นหาข้อมูลของยูสเซอร์สำเร็จ แถบสถานะบาร์จะปรากฏข้อความ "Search Complete..." แต่ถ้าค้นหาไม่พองจะปรากฏข้อความ "Not Found..."

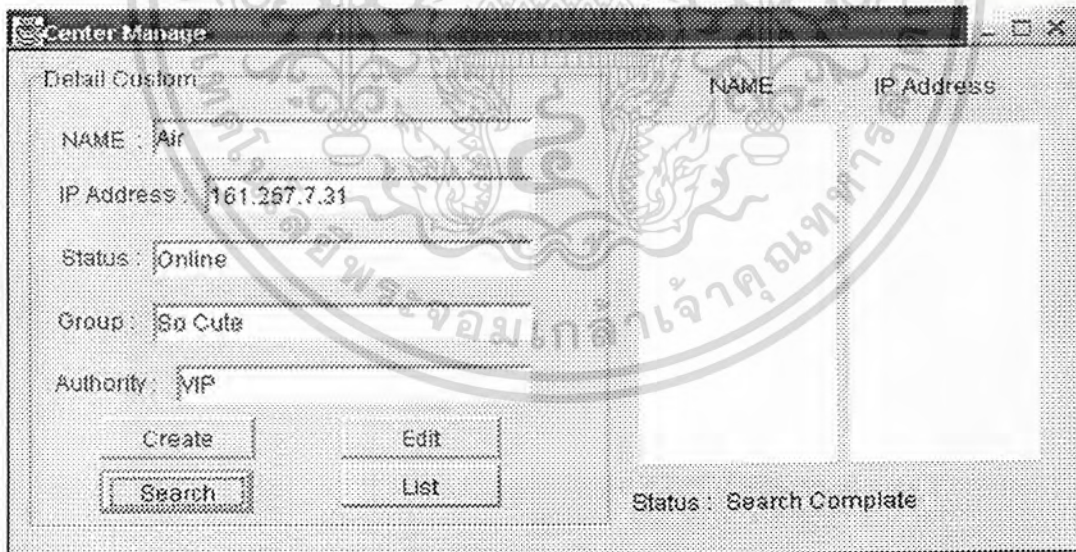
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 แสดงผลของการค้นหาข้อมูล

5.1.2.3 การแก้ไขข้อมูลของยูสเซอร์ที่ทำการลงทะเบียนแล้ว

การเรียกใช้การทำงานการแก้ไขข้อมูลของยูสเซอร์ที่ทำการลงทะเบียนแล้วนั้น ถ้าทราบยูสเซอร์เนมก็สามารถแก้ไขได้โดยคลิกกรอกรายละเอียดลงไปยังยูสเซอร์อินเตอร์เฟซ แล้วคลิกที่ปุ่ม Edit แต่เพื่อความปลอดภัยควรทำการค้นหาข้อมูลขึ้นมาก่อนที่จะทำการแก้ไข



รูปที่ 5-7 แสดงการกรอกข้อมูลที่ต้องการแก้ไข

โปรแกรมจะทำการค้นหาข้อมูลของยูสเซอร์จากฐานข้อมูล โดยนำค่าพารามิเตอร์จากยูสเซอร์อินเตอร์เฟซ (User interface) ในส่วนของยูสเซอร์เนมไปเป็นข้อมูลในการค้นหาข้อมูลตามคำสั่ง SQL ต่อไปนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น String sql = "Select * FROM Server_DB"; อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการแก้ไข (1) ไปใช้

ResultSet j = null;

(2)

```
j = stmt.executeQuery(sql);
```

```
int i=1;
```

```
while(j.next())
```

```
{
    if(NameT.equals(j.getString("NAME"))) {i = -1;}
}
```

ในบรรทัดที่ (1) เป็นคำสั่ง SQL ที่จะทำการเลือก row ทั้งหมดจากตาราง Server_DB m ที่อยู่บนฐานข้อมูลโดยตัวแปร j บรรทัด (2) เป็นตัวแปรประเภทกลุ่มที่มีลักษณะต่อกันไปเรื่อย โดยมีฟังก์ชันการเรียกค่าของตัวแปรตัวต่อไปในกลุ่มด้วยคำสั่ง next() บรรทัด(3) จะเป็นการ execute คำสั่ง SQL จากนั้นจึงวนลูปเพื่อหา row ที่มียูสเซอร์เนมเดียวกับค่าในพารามิเตอร์ NameT

จากนั้นจึงทำการแก้ไขข้อมูลของยูสเซอร์เนมที่ต้องการตามคำสั่ง SQL ดังต่อไปนี้

```
if(i == -1)
```

```
{
    sql = "UPDATE Server_DB SET
```

```
NAME="+NameT+",STATUS="+StatusT+",Authority="+AuthorityT+",
```

```
[ GROUP="+GroupT+" WHERE IP = "+IpT+"";
```

```
stmt.executeUpdate(sql);
```

```
frame.Show_Status("Edit Complete");
```

```
}else
```

```
{ frame.Show_Status("IP not found");
```

บรรทัด (1) เป็นคำสั่ง SQL ที่จะทำการค้นหา row ของยูสเซอร์ที่ต้องการจากฐานเป็นแล้วทำการแก้ไขข้อมูลใน row นั้นใหม่ ซึ่งถ้าโปรแกรมทำการแก้ไขข้อมูลของยูสเซอร์สำเร็จ แถบสถานะบาร์จะปรากฏข้อความ “Edit Complete...” แต่ถ้าค้นหาไม่พบจะปรากฏข้อความ “Not Found...”

ID	NAME	IP	STATUS	Authority	Group	Pass
1	Pick	151.245.6.65	Offline	VIP	Narak	Test1
2	Air	161.257.7.31	Online	VIP	So Cute	Test2
3	Cherry	151.245.3.201	Online	VIP	Narak	Test3
4	Jame	161.245.1.83	Online	Custom	Face	Test4
5	New User	151.245.6.01	Online	Economic	Face	

รูปที่ 5-8 แสดงตาราง Server_DB บนฐานข้อมูลของเซิร์ฟเวอร์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.4 การแสดงยูสเซอร์เนม และไอพีแอสเครสของยูสเซอร์ที่ลงทะเบียนแล้ว

การเรียกใช้การทำงานการแสดงผลข้อมูลของยูสเซอร์ที่ทำการลงทะเบียนแล้วทั้งหมดนั้น จะทำงานเมื่อคลิกที่ปุ่ม List แล้วโปรแกรมจะทำงานดังนี้

```
String sql ="Select * FROM Server_DB ";
```

 (1)

```
ResultSet l = null;
```

```
l = stmt.executeQuery(sql);
```

 (2)

```
while(l.next())
```

```
{
```

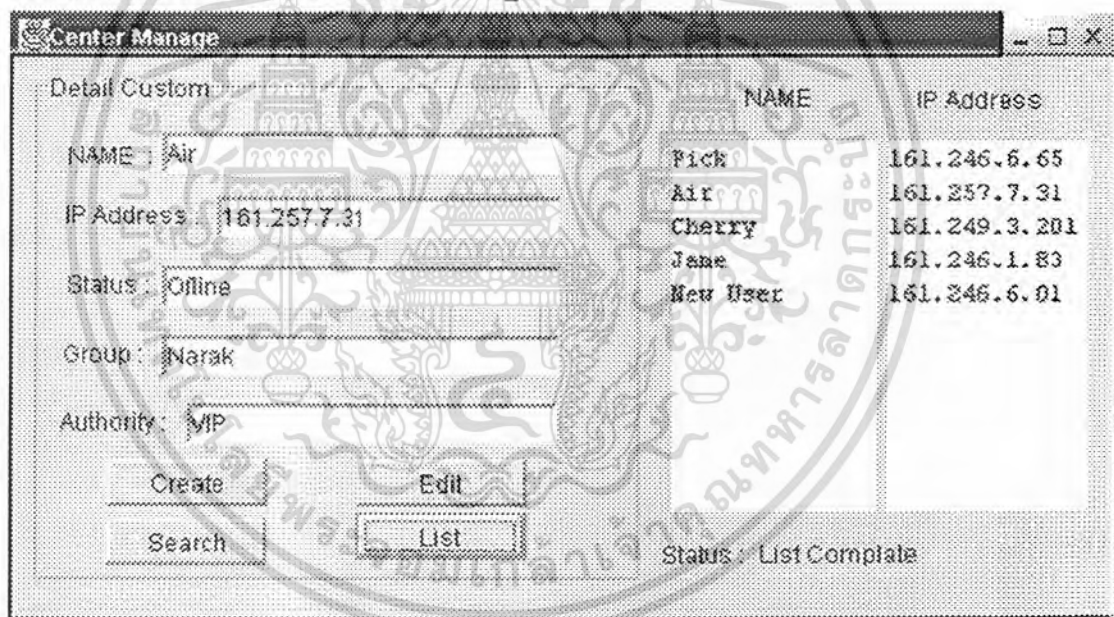
```
frame.Show_List(l.getString("NAME"),l.getString("IP"));
```

 (3)

```
}
```

```
frame.Show_Status("List Complete");
```

ในบรรทัดที่ (1) เป็นคำสั่ง SQL ที่จะทำการเลือก row ทั้งหมดจากตาราง Server_DB ที่อยู่บนฐานข้อมูล บรรทัด(2)จะเป็นการ excute คำสั่ง SQL บรรทัด (3) ทำการแสดงผลยูสเซอร์เนม และไอพีแอสเครสทั้งหมดที่มีในตาราง Server_DB

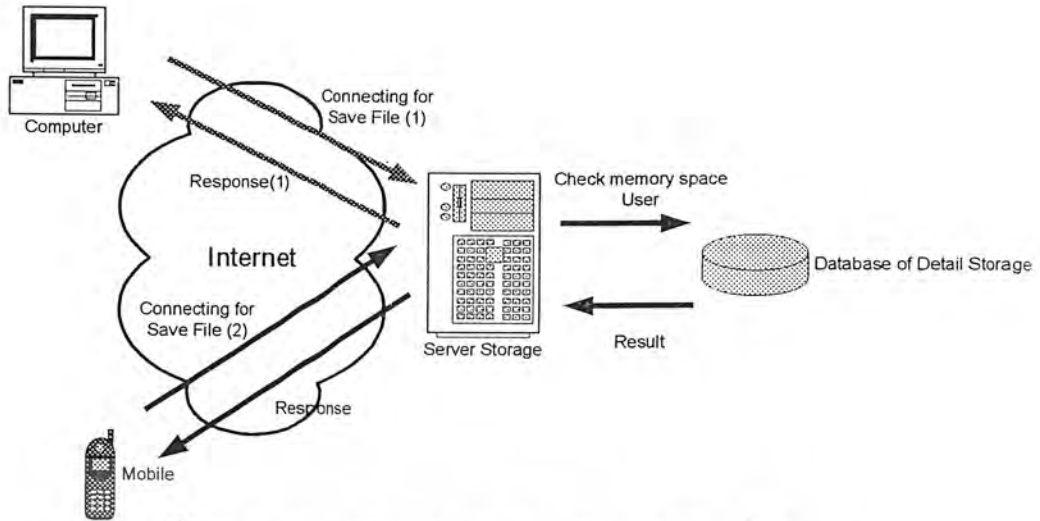


รูปที่ 5-9 แสดงผลของการ List ชื่อ และไอพีแอสเครสทั้งหมดที่ลงทะเบียนกับระบบ

5.2 เซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูลเพื่อจัดเก็บข้อมูล

เป็นเซิร์ฟเวอร์ที่จะรองรับการติดต่อจากไคลเอนต์ที่ต้องการใช้บริการจัดเก็บข้อมูลบนพื้นที่ที่ให้บริการสำรองข้อมูล ซึ่งทำให้เซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูลนี้จะต้องทำการออกแบบฐานข้อมูลเพื่อใช้เก็บข้อมูลการใช้พื้นที่ของยูสเซอร์แต่ละคน รวมทั้งเก็บข้อมูลของขนาดพื้นที่ที่ให้บริการของแต่ละประเภท authority

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดงการติดต่อโดยรวมระหว่างเซิร์ฟเวอร์ที่ให้บริการ
พื้นที่ในการสำรองข้อมูล กับไคลเอนต์

5.2.1 การออกแบบฐานข้อมูลของเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล

เนื่องจากเป็นเซิร์ฟเวอร์ที่ให้บริการ ในด้านพื้นที่สำหรับสำรองข้อมูล ดังนั้นการออกแบบฐานข้อมูลจึงควรคำนึงถึงข้อมูลเกี่ยวกับการขอใช้พื้นที่ในการจัดเก็บข้อมูล และประเภทของยูสเซอร์ที่จะมาขอใช้บริการ ซึ่งสามารถออกแบบได้เป็นสองตารางดังนี้

5.2.2.1 ตารางข้อมูลการใช้พื้นที่ของแต่ละยูสเซอร์

เป็นตารางที่จะเก็บข้อมูลของผู้ที่เข้ามาติดต่อ และขอใช้บริการสำรองพื้นที่เพื่อการจัดเก็บข้อมูล ซึ่งประกอบด้วยคอลัมน์เช่นมีดังต่อไปนี้

- ID เป็นคอลัมน์ที่กำหนดให้เป็น primary key ซึ่งจะทำการเพิ่มค่าอัตโนมัติทุกครั้งที่มีการสร้าง row ของยูสเซอร์ใหม่

- NAME เป็นคอลัมน์ที่เก็บยูสเซอร์เนม (User name) ของยูสเซอร์ ซึ่งควรจะเป็นชื่อเดียวกับตอนลงทะเบียนเพื่อร้องขอเข้าสู่ระบบที่เซิร์ฟเวอร์กลาง

- IP เป็นคอลัมน์ที่เก็บไอพีแอสแตรสของยูสเซอร์เนมแต่ละอันที่ทำการติดต่อเข้ามาเพื่อร้องขอเข้าสู่ระบบ ในกรณีที่ป็นยูสเซอร์เนมใหม่ที่เพิ่งทำการลงทะเบียนเพื่อเข้าสู่ระบบ ค่าในคอลัมน์นี้จะกำหนดค่าไว้ให้เป็น 161.246.6.300 ซึ่งเป็นหมายเลขที่เกินกว่าที่ใช้กันในระบบทั่วไปในปัจจุบัน

- SIZE เป็นคอลัมน์ที่เก็บขนาดของพื้นที่ที่ยูสเซอร์แต่ละคนได้ใช้เก็บข้อมูลบนเครื่องเซิร์ฟเวอร์ที่ให้บริการพื้นที่ในการสำรองข้อมูล เพื่อจัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ID	NAME	IP	SIZE	Pass
1	Pick	161.246.6.65	275	Test1
2	Air	161.257.7.31	1520	Test2
3	Jame	161.246.1.83	682	Test4
(AutoNumber)				

รูปที่ 5-11 แสดงตาราง *Storage_DB* บนฐานข้อมูลของ
เซิร์ฟเวอร์ที่ให้บริการสำรองพื้นที่

5.2.1.2 ตารางข้อมูลขนาดพื้นที่ที่เปิดให้บริการของ authority แต่ละประเภท

โดยมีรายละเอียดในแต่ละคอลัมน์ดังนี้

- ID เป็นคอลัมน์ที่กำหนดให้เป็น primary key ซึ่งจะทำการเพิ่มค่าอัตโนมัติทุกครั้งที่มีการสร้าง row ของประเภท authority ใหม่

- TYPE เป็นคอลัมน์ที่เก็บประเภทของ authority

- TYPE_SIZE เป็นคอลัมน์ที่เก็บขนาดสูงสุดที่เซิร์ฟเวอร์ให้บริการสำหรับ authority แต่ละประเภท

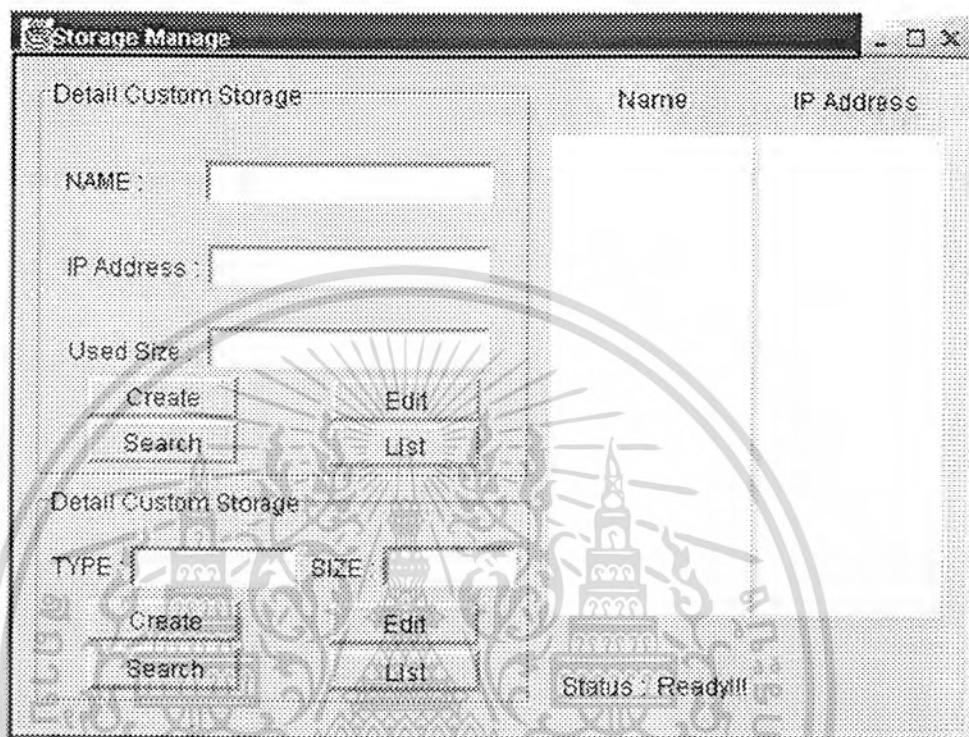
ID	TYPE	TYPE_SIZE
1	VIP	2200
2	Economic	2000
3	Custom	1800
(AutoNumber)		

รูปที่ 5-12 แสดงตาราง *TYPE* บนฐานข้อมูลของ
เซิร์ฟเวอร์ที่ให้บริการสำรองพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 โปรแกรมช่วยจัดการข้อมูลบนฐานข้อมูล

เช่นเดียวกับเซิร์ฟเวอร์หลัก เพื่อให้สะดวกกับการใช้งานการพัฒนาระบบนี้จึงสร้างโปรแกรมขนาดเล็กขึ้นมาเพื่อจัดการกับข้อมูลบนฐานข้อมูล โดยมีลักษณะการใช้งาน โปรแกรม ดังรูปที่ 5-13



รูปที่ 5-13 แสดงยูสเซอร์อินเตอร์เฟซของโปรแกรมช่วยจัดการบนฐานข้อมูล

ซึ่งลักษณะการทำงาน การเขียนโปรแกรม และการเขียนคำสั่ง SQL นั้นมีความคล้ายคลึงกับโปรแกรมช่วยจัดการข้อมูลบนฐานข้อมูลของเซิร์ฟเวอร์หลักดังนั้นจึงไม่ขอกล่าวถึง ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

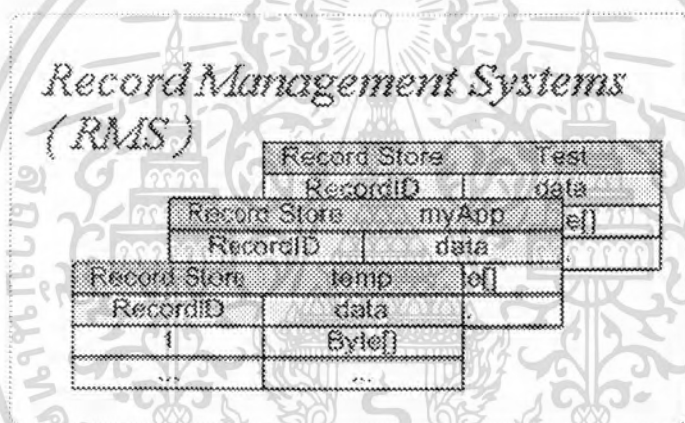
บทที่ 6

การออกแบบระบบไฟล์บนโทรศัพท์มือถือ

ในปัจจุบันนี้ โทรศัพท์มือถือมีความสามารถที่จะจัดเก็บข้อมูลแบบถาวรลงในหน่วยความจำบน (ไม่มีไฟเลี้ยงแล้วข้อมูลไม่หาย) เป็นประโยชน์ต่อการเขียนโปรแกรมทำให้ใช้ประโยชน์จากโปรแกรมได้สูงสุด

6.1 Record Management Systems

ระบบจัดการข้อมูล (Record Management System – RMS) รูปแบบของการเก็บข้อมูลจะเป็นการเก็บข้อมูลลงตาราง1มิติ และใช้ recordID อ้างอิงเพื่อเข้าถึงข้อมูลนั้น และชื่อของแต่ละ Record Store ต้องไม่ซ้ำกันด้วย



รูปที่ 6-1 ตารางแสดงการเก็บข้อมูลของ RecordStore

การใช้งาน RecordStore มีฟังก์ชันการใช้งานดังนี้

1. สร้างหรือเปิด RecordStore ขึ้นมา โดยใช้คำสั่ง

```
RecordStore rs = RecordStore.openRecordStore(rsName, true);
```

2. เมื่อเปิด RecordStore แล้ว จึงสามารถเก็บข้อมูลลงไปได้ ข้อมูลที่เก็บต้องเป็นอาร์เรย์ของไบต์เสมอ เพราะฉะนั้นถ้าต้องการเก็บข้อมูลในรูปแบบอื่นๆ หรือเก็บข้อมูลหลายๆชนิดในเรคอร์ดเดียวกัน จะต้องแปลงข้อมูลนั้นให้เป็นอาร์เรย์ของไบต์ก่อน แล้วจึงเก็บลงเรคคอร์ดตามปกติ ดังนี้

```
ByteArrayOutputStream bout = new ByteArrayOutputStream();
```

```
DataOutputStream dout = new DataOutputStream(bout);
```

```
dout.writeInt(type); // integer
```

```
dout.writeUTF(name); // String
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dout.writeLong(created); // long
dout.flush();
byte[] data = bout.toByteArray();
rs.addRecord( data, 0, data.length);
```

3. การลบเรคคอร์ดออกจะใช้ RecordID อ่างเรคคอร์ดที่ต้องการจะลบ โดยใช้คำสั่ง

```
rs.deleteRecord( recordID );
```

เมื่อลบแล้วข้อมูลที่ RecordID นั้นจะหายไป แต่ข้อมูลอยู่หลัง RecordID นั้นจะไม่เลื่อนขึ้นไปด้วย ซึ่งอาจจะทำให้เกิดความผิดพลาดได้ง่าย

4. เมื่อ ไม่ต้องการใช้งาน RecordStore แล้ว ควรจะปิด RecordStore ด้วยคำสั่ง

```
rs.closeRecordStore();
```

เพื่อไม่ให้ RecordStore ยังใช้ทรัพยากรโดยเปล่าประโยชน์ และประหยัดหน่วยความจำบนโทรศัพท์มือถือที่มีไม่มากอยู่แล้ว

เนื่องจากการเก็บข้อมูลเก็บต้องอยู่ในรูปของอาร์เรย์ ของไบต์ การใช้งานจึงไม่เหมาะสำหรับให้ผู้ใช้สามารถใช้ได้โดยตรง ส่วนใหญ่จะใช้สำหรับเขียน log file ไว้ใช้สำหรับแต่ละโปรแกรมเท่านั้น

6.2 การออกแบบระบบไฟล์บน MIDP

จาก RMS API ที่ MIDP มีให้ใช้เก็บข้อมูลได้ จึงสามารถออกแบบให้ Record Store สามารถจัดการได้ง่ายขึ้นเป็น ระบบไฟล์

6.2.1 หลักการในการทำ Record Store ให้เป็นไฟล์

FileID	RecordID	Data
n	2n-1	File property : name, type, length, lastModified
	2n	File Content
1	1	Binary File 1 property
	2	Binary File 1 content
2	3	Folder 1 property
	4	Folder 1 content
3	5	Virtual File 1 property
	6	Virtual File 1 content
...

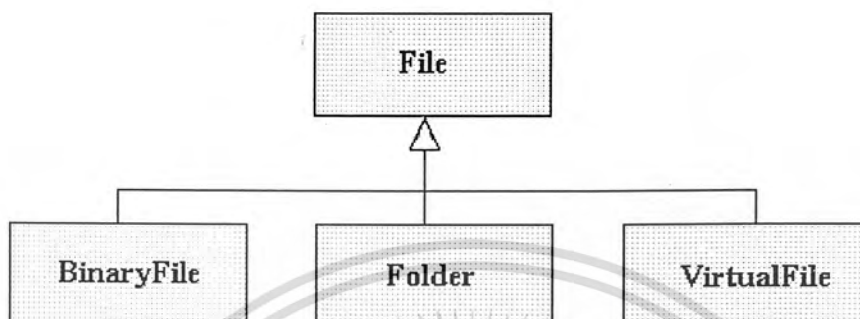
รูปที่ 6-2 แสดงการใช้ Record Store เก็บข้อมูลระบบไฟล์

File จะใช้พื้นที่ในการเก็บข้อมูล 2 เรคคอร์ด เรคคอร์ดแรกจะใช้เก็บคุณสมบัติของไฟล์ไว้ใช้ในการอ่านข้อมูลสำหรับโพลเดอร์เพื่อแสดงรายชื่อไฟล์ทั้งหมดที่มีอยู่ใน Record

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า. ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.2 ประเภทของไฟล์

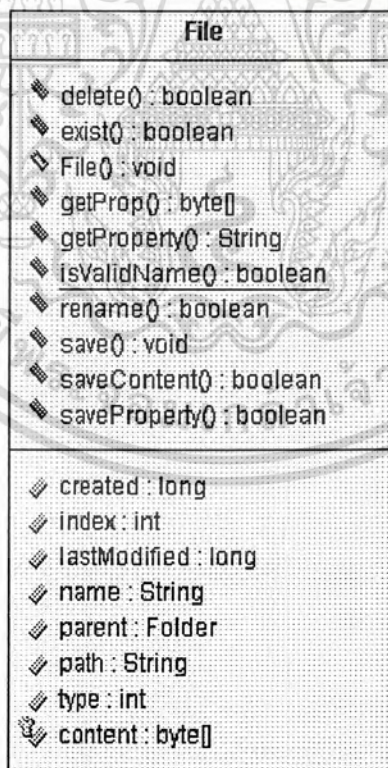
ระบบนี้จะจำลองมาจากระบบไฟล์บนเครื่องคอมพิวเตอร์ ซึ่งเป็นที่คุ้นเคยสำหรับผู้ใช้อุปกรณ์แล้ว แต่มีการเพิ่มเติมไฟล์ลักษณะพิเศษอีกแบบหนึ่งเข้าไปด้วย ซึ่งจะสามารถโครงสร้างของไฟล์นี้แสดงได้ดังรูปข้างล่างนี้



รูปที่ 6-3 แสดงโครงสร้างของระบบไฟล์บนโทรศัพท์มือถือ

6.2.2.1 Class File

File เป็นคลาสต้นแบบที่มีความสามารถพื้นฐานของไฟล์ทั้งหมด



รูปที่ 6-4 แสดงคลาส File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attribute

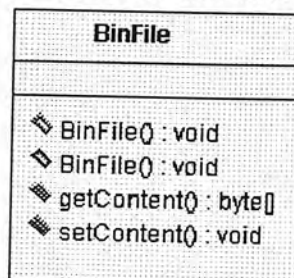
- Name : ชื่อของไฟล์
- Parent : ชื่อโฟลเดอร์แม่ ของไฟล์
- Path : ชื่อพารของไฟล์
- Type : ชนิดของไฟล์ มีค่าเป็น `BINARY_FILE`, `VIRTUAL_FILE`, `FOLDER` หรือ `EMPTY`
- Index : RecordID ที่ไฟล์นี้ถูกอ่านขึ้นมาจาก Record Store เพื่อเอาไว้อ้างอิงเมื่อต้องการจะเก็บทับซ้ำลงไปเดิม ลงใน Record Store
- Created : วันเวลาที่ไฟล์ถูกสร้างขึ้นมา
- LastModified : วันเวลาที่ไฟล์ถูกแก้ไขครั้งสุดท้ายขึ้นมา
- Content : เนื้อไฟล์จริง ซึ่งจะเก็บค่านีกลงใน record ช่องถัดไปต่อจาก property

Method

- Delete() : ลบไฟล์ โดยการเปลี่ยน type ให้เป็น `EMPTY_FILE` แล้วเรียกให้โฟลเดอร์ parent เก็บค่าลง Record Store แล้วจึงอัปเดตรายชื่อไฟล์ทั้งหมดในโฟลเดอร์
- Exist() : เช็คไฟล์นี้ยังมีอยู่หรือไม่ โดยดูจาก type
- GetProp() : ให้ property ที่พร้อมเก็บลงหน่วยความจำ
- GetProperty() : ให้ property ทั้งหมดของไฟล์ออกมาเป็น String เพื่อแสดงผล
- isValidName() : เช็คความถูกต้องของชื่อไฟล์ ว่ามีตัวอักษรพิเศษอยู่ในชื่อหรือไม่ ตัวอักษรพิเศษมีดังนี้ “/”, “\”, “:”, “*”, “”, “?”, “|”, “<”, “>”
- rename() : เปลี่ยนชื่อไฟล์
- save() : เก็บข้อมูลทั้ง property และ content ลงหน่วยความจำ
- saveContent() : เก็บข้อมูล content ลงหน่วยความจำ
- saveProperty() : เก็บข้อมูล property ลงหน่วยความจำ

6.2.2.2 Class Bin File

สืบทอดมาจาก Class File เก็บ content เป็นไฟล์ข้อมูลจริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 6-5 แสดงคลาส BinFile ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Method

- getContent() คืนค่าของ content หรือ เนื้อข้อมูล
- setContent() ตั้งค่าของ content หรือ เนื้อข้อมูล

6.2.2.3 Class Folder

สืบทวงมาจาก Class File เก็บ content เป็นชื่อของ Record Store ที่ใช้เปิดจริง ไม่ใช่ชื่อไฟล์ เพราะชื่อ Record Store ต้องไม่ซ้ำกันเลย แต่ถ้าใช้ชื่อไฟล์เคอร์แทนก็อาจจะซ้ำกับไฟล์เคอร์อื่นๆที่อยู่คนละ directory ได้



รูปที่ 6-6 แสดงคลาส Folder

Attribute

- rs : Record Store ของไฟล์เคอร์
- rsName : ชื่อ Record Store ที่ใช้ในการ openRecordStore อาจจะไม่เหมือนชื่อของไฟล์เคอร์ก็ได้ ถ้าหากชื่อนี้เคยถูกใช้ในไฟล์เคอร์อื่นแล้ว
- count : จำนวนไฟล์ทั้งหมดในไฟล์เคอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อวัตถุประสงค์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาหรือข้อมูลของเอกสารนี้ส่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- nameList : รายชื่อของไฟล์ทั้งหมด
- typeList : รายชื่อของชนิดไฟล์ทั้งหมด

- files : อาร์เรย์ของไฟล์ทั้งหมดที่อ่านขึ้นมาได้จาก Record Store

Method

- addFile(File) : เก็บไฟล์เพิ่มลงในโพลเดอร์
- close() : ปิดโพลเดอร์เพื่อปิด Record Store
- deleteFile() : ลบไฟล์ออกจากโพลเดอร์
- getData(int) : ให้ข้อมูลใน Record ที่ต้องการ
- getList() : ให้รายชื่อไฟล์ทั้งหมดในโพลเดอร์
- getNumFile() : ให้จำนวนไฟล์ทั้งหมดในโพลเดอร์
- nameExists() : เช็คว่ามีชื่อไฟล์ที่เป็นชนิดเดียวกันอยู่ในโพลเดอร์หรือไม่
- saveContent(File) : เก็บข้อมูลที่รับเข้ามาลงในหน่วยความจำ
- saveProperty(File) : เก็บข้อมูลที่รับเข้ามาลงในหน่วยความจำ
- rsNameExist() : เช็คว่าชื่อ Record Store นี้เคยถูกสร้างหรือไม่
- createFolder() : สร้างโพลเดอร์ใหม่
- searchEmptyRecord() : หา record ที่ว่างอยู่ใน Record Store
- updateList() : เปลี่ยนแปลงข้อมูลในโพลเดอร์เมื่อมีการเปลี่ยนแปลงเกิดขึ้น

6.2.2.4 Class VirtualFile

สืบทอดมาจาก Class File เป็นไฟล์เสมือนที่สามารถจัดการได้เหมือนไฟล์ปกติแต่ไม่ได้เก็บไฟล์จริงๆไว้ในโทรศัพท์มือถือ แต่เก็บไว้ที่เครื่องคอมพิวเตอร์เครื่องอื่นที่สามารถติดต่อได้ คล้ายกับ shortcut บน Windows

VirtualFile
<ul style="list-style-type: none"> ✚ getContent() : byte[] ✚ setHost() : void ✚ setLocation() : void ✚ VirtualFile() : void ✚ VirtualFile() : void ✚ getHost() : String ✚ getLocation() : String
<ul style="list-style-type: none"> ✚ host : String ✚ location : String

รูปที่ 6-7 แสดงคลาส VirtualFile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attribute

- host : IP address ของ โฮสต์ที่ไฟล์อยู่
- location : ที่อยู่ของไฟล์บน โฮสต์

Method

- getContent() คืนค่าของ host และ location เป็นอาร์เรย์ ของไบต์เพื่อ เก็บลงหน่วยความจำ
- getLocation() คืนค่าของ location
- getHost() คืนค่าของ IP address ของ host
- setLocation() ตั้งค่า location
- setHost() ตั้งค่า IP address ของ host



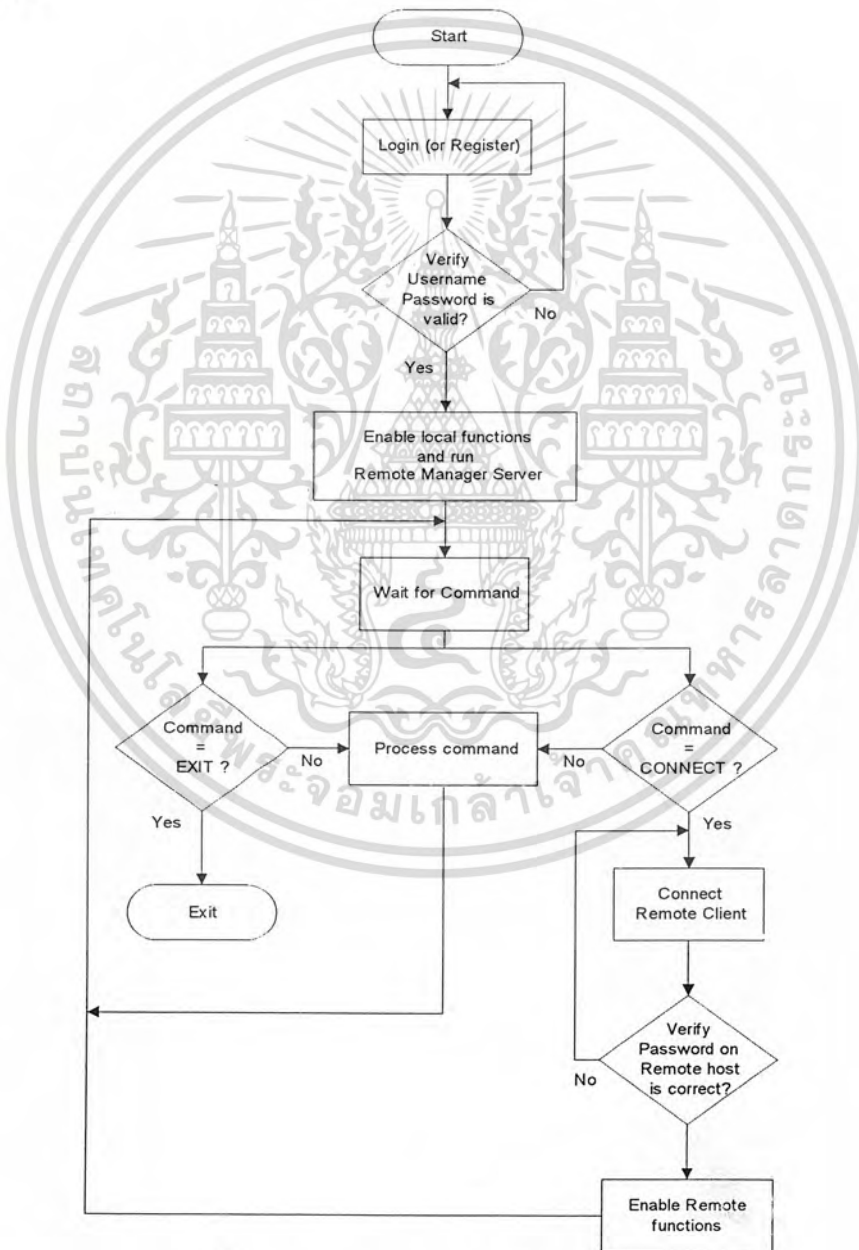
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การพัฒนาโปรแกรมส่วนไคลเอนต์

การบริหารจัดการไฟล์เป็นการให้บริการกับโทรศัพท์มือถือหรือเครื่องคอมพิวเตอร์ ไปควบคุมจัดการไฟล์ที่เครื่องคอมพิวเตอร์ที่ออนไลน์อยู่ในระบบได้ เพื่ออำนวยความสะดวกให้กับผู้ใช้สามารถเข้ามาใช้งานได้ไม่ว่าจะอยู่ที่ไหนก็ตาม

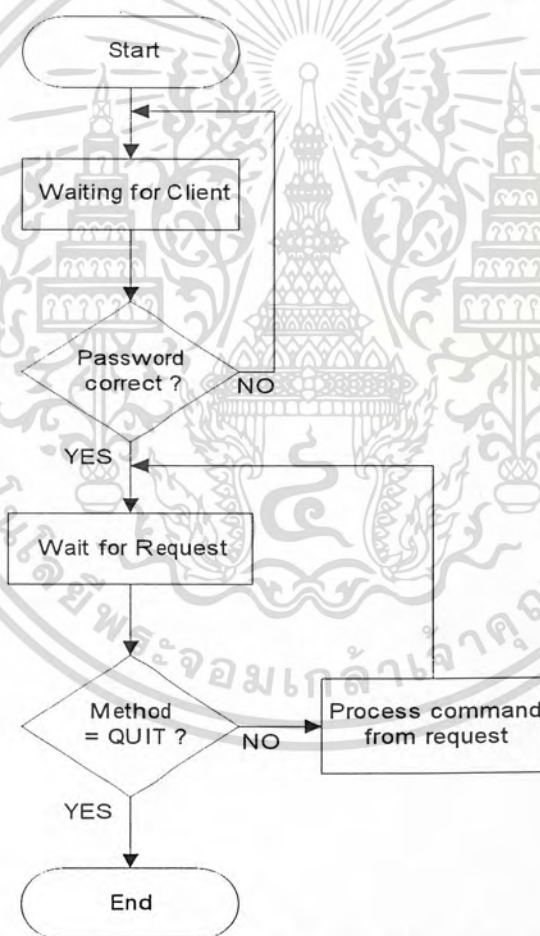
การทำงานของโปรแกรมที่อยู่บนเครื่องคอมพิวเตอร์ส่วนของไคลเอนต์ สามารถอธิบายได้ตามแผนภาพข้างล่างนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 7-1 แสดงการทำงานของไคลเอนต์ นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อเปิดโปรแกรมแล้ว ผู้ใช้ต้องเข้าสู่ระบบโดยใช้ username และ password ถ้าใช้งานเป็นครั้งแรกต้องสมัครสมาชิกก่อน
2. ถ้า password ถูกต้อง ก็จะสามารถใช้งานฟังก์ชันที่ทำงานบนเครื่องได้ และถ้าเป็นไคลเอนต์แบบแอปพลิเคชัน(ทำงานบนเครื่องคอมพิวเตอร์) จะรันเซิร์ฟเวอร์ไว้รอรับไคลเอนต์อื่นที่จะติดต่อเข้ามา
3. รอรับคำสั่งจากผู้ใช้ แล้วทำงานตามคำสั่งนั้น ถ้าคำสั่งเป็น CONNECT ไคลเอนต์ที่ติดต่อจะตรวจสอบ Password ว่าถูกต้องแล้ว จะสามารถใช้งานฟังก์ชันที่ทำงานบนเครื่องไคลเอนต์ที่ติดต่อได้
4. จบการทำงานเมื่อคำสั่งเป็น EXIT

การทำงานในส่วนของเซิร์ฟเวอร์ มีการทำงานหลักๆ ตามแผนภาพข้างล่างนี้



รูปที่ 7-2 แสดงการทำงานของไคลเอนต์ส่วนเซิร์ฟเวอร์

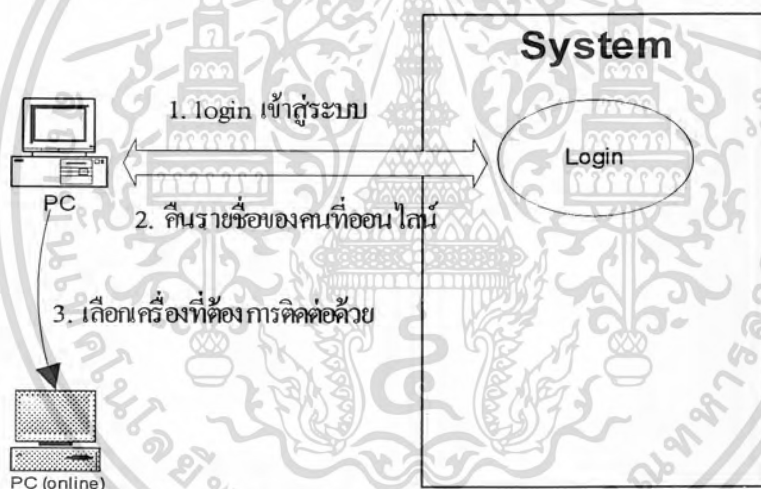
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เซิร์ฟเวอร์จะทำงานเป็น Thread ให้สามารถทำงานหลักไปพร้อมกับ การเปิดชอกเกตรอรับ ไคลเอนต์เข้ามาติดต่อ
2. เมื่อมีไคลเอนต์ติดต่อเข้ามาแล้ว ก็จะตรวจสอบ Password แล้วส่งผลกลับไปบอก ถ้าถูกต้องจะสร้างอีก Thread หนึ่งเพื่อรองรับการทำงานของไคลเอนต์นั้น แล้วเซิร์ฟเวอร์หลักก็กลับไปรอรับไคลเอนต์เหมือนเดิม

7.1 การเข้าสู่ระบบจัดการไฟล์

7.1.1 กรณีที่ไคลเอนต์เป็นเครื่องคอมพิวเตอร์

ผู้ใช้จะต้องเข้าสู่ระบบ (Login) ก่อนเสมอ โดยใช้ username และ password เพื่อให้ไคลเอนต์อื่นสามารถมองเห็นเครื่องผู้ใช้ได้เช่นกัน หลังจากนั้นเซิร์ฟเวอร์จะส่งรายชื่อและ IP address ของเครื่องอื่นที่ออนไลน์อยู่กลับมา แล้วเลือกเครื่องคอมพิวเตอร์ที่ต้องการติดต่อกับจากรายชื่อ โปรแกรมก็จะใช้ IP address ติดต่อกับไคลเอนต์นั่นเอง



รูปที่ 7-3 แสดงขั้นตอนการเริ่มเข้าสู่ระบบของไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์

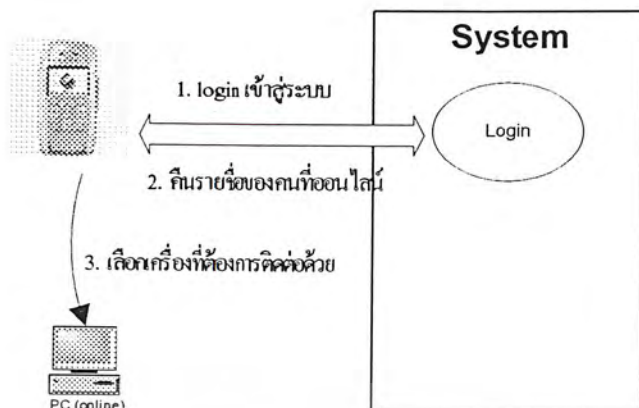
หลังจากนั้น ไคลเอนต์ที่เราติดต่อกำลังหาหาเซิร์ฟเวอร์ ถ้าหาเซิร์ฟเวอร์ถูกต้องเราก็จะติดต่อกับไคลเอนต์นั้นขอใช้บริการได้แล้ว

แต่ถ้าผู้ที่ยังไม่เคยสมัครสมาชิกใช้งานมาก่อน ก็ต้องใช้ username และ password มาลงทะเบียนกับเซิร์ฟเวอร์ก่อน โดยที่ขั้นตอนหลังจากนั้นเหมือน Login ทุกอย่าง

7.1.2 กรณีที่ไคลเอนต์เป็นโทรศัพท์มือถือ

เนื่องจากโทรศัพท์มือถือไม่มีความสามารถเป็นเซิร์ฟเวอร์ได้ การเข้าสู่ระบบของโทรศัพท์มือถือจึงไม่จำเป็นต้องใช้ username และ password ในการติดต่อเซิร์ฟเวอร์กลาง เพราะฉะนั้นในกรณีที่โทรศัพท์มือถือจะคืนรายชื่อและ IP address ทั้งหมดได้เลย

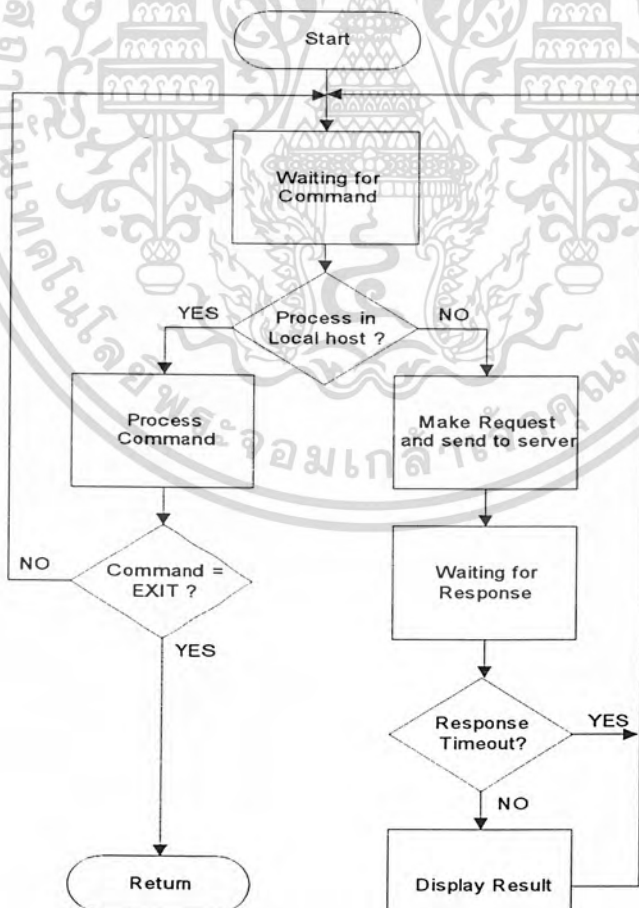
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนลิขสิทธิ์ในชื่อของสถาบันนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-4 แสดงขั้นตอนการเริ่มเข้าสู่ระบบของไคลเอนต์ที่เป็นโทรศัพท์มือถือ

7.2 การจัดการไฟล์

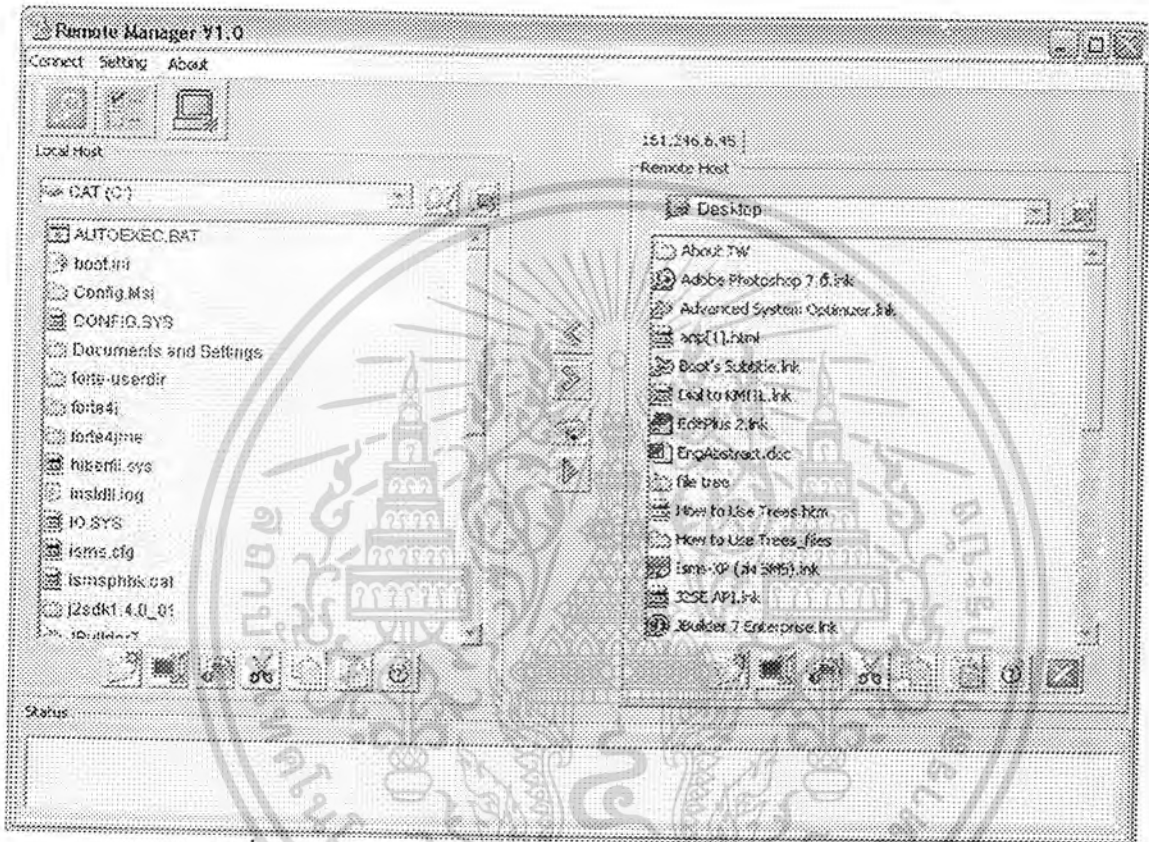
โปรแกรมในส่วนนี้มีการทำงานคล้ายกับโปรแกรม FTP ทั่วไปที่สามารถจัดการไฟล์ได้ โปรแกรมจะแบ่งการทำงานเป็น 2 ส่วนด้วยกัน คือ ส่วนที่จัดการไฟล์บนเครื่องไคลเอนต์ และจัดการไฟล์บนเครื่องเซิร์ฟเวอร์ แสดงเป็นแผนภาพการทำงานได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 7-5 แสดงการทำงานส่วนจัดการไฟล์
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุตบแต่งสิ่งเนื้อหาและต้องยกย่องถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่จัดการไฟล์ที่บนเครื่องตัวเอง จะทำงานตามที่วิธีที่ระบบไฟล์มีไว้ให้แล้ว ทำงานจัดการไฟล์ปกติ และส่วนที่จัดการไฟล์ที่เซิร์ฟเวอร์ เมื่อได้รับคำสั่งแล้วจะสร้าง Request Form ตามโปรโตคอล FMP ที่ได้ออกแบบไว้ในบทที่ 4 แล้วจะส่งไปให้เซิร์ฟเวอร์ดำเนินการ แล้วรอรับผลออกมาแสดงบนโปรแกรม

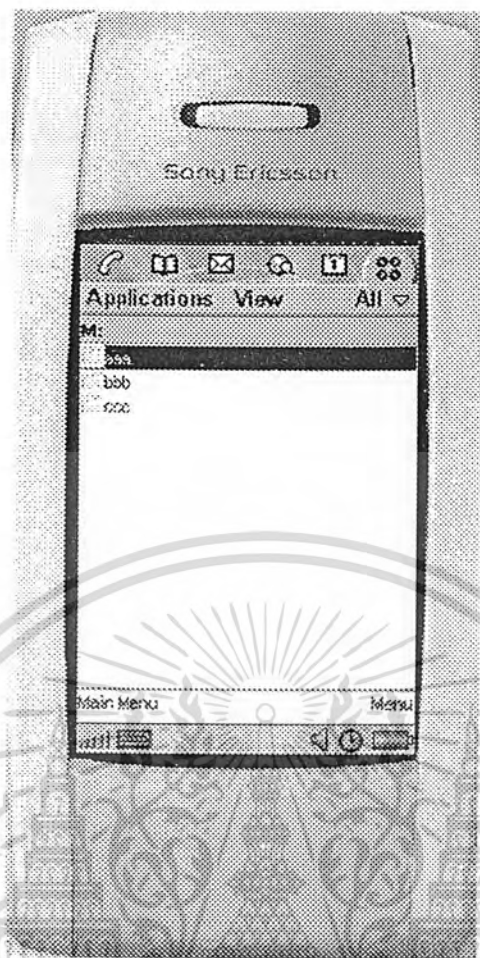
ส่วนที่ทำงานบนไคลเอนต์ จะเป็นการพัฒนาโปรแกรมส่วนที่ติดต่อกับระบบไฟล์บนเครื่อง ซึ่งจะต้องแสดงออกมาเป็นส่วนติดต่อกับผู้ใช้งาน ให้สามารถใช้ได้สะดวก คังรูปข้างล่าง



รูปที่ 7-6 แสดงส่วนที่ติดต่อกับผู้ใช้งานส่วนจัดการไฟล์บนเครื่องคอมพิวเตอร์

จากรูปแสดงหน้าจอจัดการไฟล์ ด้านซ้ายเป็นส่วนที่แสดงไฟล์ของฝั่งบนเครื่องตัวเอง และด้านขวาเป็นไฟล์บนเครื่องเซิร์ฟเวอร์ที่เราติดต่อด้วย มีปุ่มสั่งงานต่างๆวางไว้อยู่ได้รายชื่อไฟล์ของแต่ละฝั่ง ให้เลือกใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-7 ส่วนที่ติดต่อกับผู้ใช้งานส่วนจัดการไฟล์บนโทรศัพท์มือถือ

ส่วนที่ติดต่อกับผู้ใช้ของโทรศัพท์มือถือได้พัฒนาเป็นหน้าจอให้เลือกแบบไฟล์โฟลเดอร์ตามที่ได้ออกแบบไว้ในบทที่ 6 แล้ว ให้สามารถใช้งานได้ใกล้เคียงกับเครื่องคอมพิวเตอร์ฟังก์ชันการทำงานในส่วนจัดการไฟล์นี้ จะมีฟังก์ชันหลักๆดังนี้

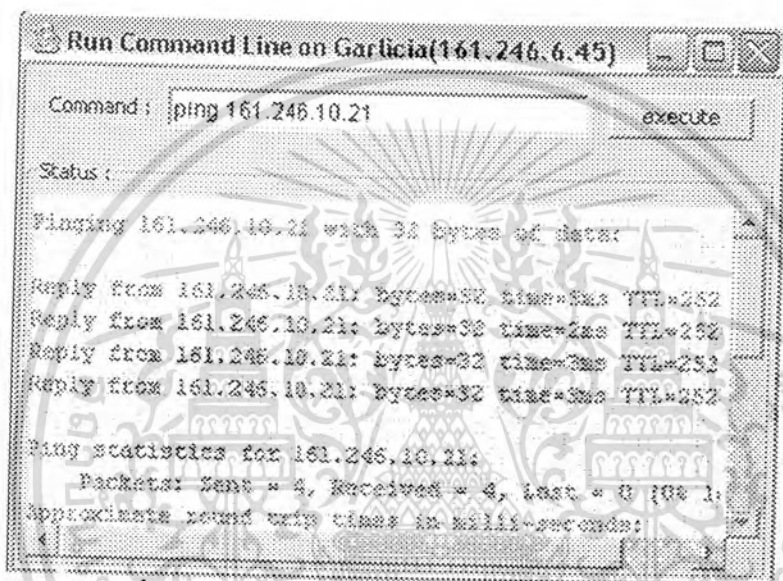
- Delete ลบไฟล์หรือโฟลเดอร์ (สามารถลบทั้งซบโฟลเดอร์ได้)
- Open Folder เปิด โฟลเดอร์ขึ้นมาอ่านรายชื่อไฟล์ข้างใน
- Folder Up ออกไปยังโฟลเดอร์แม่
- Copy คัดลอกไฟล์ไปอีกที่หนึ่ง
- Property คุณสมบัติของไฟล์
- Upload ส่งไฟล์ขึ้นให้เซิร์ฟเวอร์
- Download รับไฟล์จากเซิร์ฟเวอร์

7.3 การสั่งพิมพ์งานโดยผ่านเครื่องคอมพิวเตอร์เครื่องอื่นที่อยู่ในระบบ ำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่โทรศัพท์มือถือไม่สามารถต่อพรินเตอร์ได้จึงไม่สามารถส่งงานพิมพ์เอกสารที่ต้องการออกมาได้อย่างเครื่องคอมพิวเตอร์ จึงได้พัฒนาให้โคลเอนต์สามารถส่งพิมพ์เอกสารที่อยู่บนเครื่องส่งไปที่เซิร์ฟเวอร์พิมพ์ออกมาทางพรินเตอร์ได้ โดยให้ โคลเอนต์ส่งไฟล์ที่จะพิมพ์ออกมาไปที่เซิร์ฟเวอร์เก็บเอกสารเป็นไฟล์ชั่วคราวไว้ก่อน เมื่อพิมพ์แล้วจึงค่อยลบทิ้ง

7.4 การส่งคอมมานด์ไลน์ไปยังเครื่องคอมพิวเตอร์

โคลเอนต์จะสามารถส่งงานเครื่องคอมพิวเตอร์โดยใช้ คอมมานด์ไลน์ เหมือนสั่งรันบนเครื่องโดยตรง แล้วเซิร์ฟเวอร์จะส่งผลของการทำงานนั้น มาเป็นตัวอักษรเหมือนหน้าจอเทอร์มินอลจริง



```

Run Command Line on Garcia(161.246.6.45)
Command: ping 161.246.10.21
execute

Status:
Pinging 161.246.10.21 with 32 bytes of data:
Reply from 161.246.10.21: bytes=32 time=2ms TTL=252
Reply from 161.246.10.21: bytes=32 time=2ms TTL=252
Reply from 161.246.10.21: bytes=32 time=3ms TTL=252
Reply from 161.246.10.21: bytes=32 time=3ms TTL=252

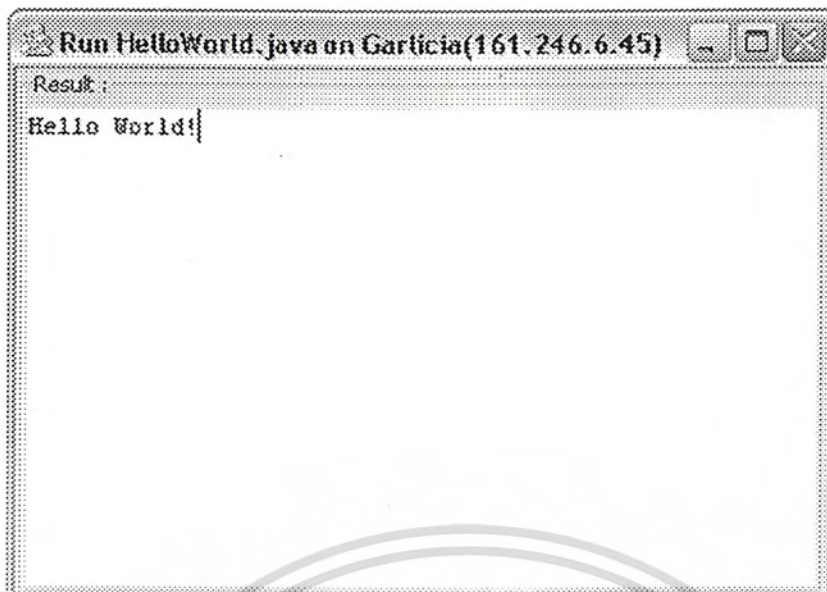
Ping statistics for 161.246.10.21:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    approximate round trip times in milliseconds:
  
```

รูปที่ 7-8 แสดงส่วนที่ติดต่อกับผู้ใช้งานสั่งคอมมานด์

7.5 การส่งคอมไพล์และรันจาวาไฟล์

โคลเอนต์จะสามารถส่งเครื่องคอมพิวเตอร์คอมไพล์แล้วรันซอร์สโค้ดจาวาที่อยู่บนเครื่อง แล้วเซิร์ฟเวอร์จะส่งผลของการทำงานนั้น มาเป็นตัวอักษรเหมือนหน้าจอเทอร์มินอลจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-9 แสดงผลจากการรันจาวาไฟล์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Jean J. Labrosse : MicroC/OS-II The Real-Time Kernel : Miller Freeman, 1999
- [2] Deitel, Harvey M. : Java How To Program : Prentice Hall, 2001
- [3] วีระศักดิ์ ชิงถาวร. 2545. Java Programming Volume. เล่มที่ 1. กรุงเทพฯ : ซีเอ็ดยูเคชั่น
- [4] วีระศักดิ์ ชิงถาวร. 2545. Java Programming Volume. เล่มที่ 2. กรุงเทพฯ : ซีเอ็ดยูเคชั่น
- [5] John W. Muchow : CORE J2ME Technology and MIDP : 2002 Sun Microsystems Inc.
- [6] <http://www.sun.com/> Organize That Build Java and J2ME
- [7] <http://gear.kku.ac.th/~thana/> Thai Researcher about JAVA
- [8] <http://www.corej2me.com/> J2ME Community site
- [9] <http://www.j2me.org> J2ME Community site
- [10] <http://www.microjava.com/> J2ME Community site
- [11] <http://www.apl.jhu.edu/~hall/java/> Java Programming Resources
- [12] <http://www.midletzone.com/> Thai Java Mobile Community



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

การสร้างแพ็คเกจและจัดเตรียมแอปพลิเคชันสำหรับ MIDlet

การสร้างแพ็คเกจ และจัดเตรียมแอปพลิเคชัน MIDP ใน J2ME นั้น ประการแรกไฟล์คลาสทั้งหมดของแอปพลิเคชัน MIDlet จะต้องเก็บอยู่ในไฟล์ JAR เพียงไฟล์เดียว ทั้งนี้ไฟล์หนึ่งๆอาจมีแอปพลิเคชันได้มากกว่าหนึ่งแอปพลิเคชัน กลุ่มของแอปพลิเคชันนี้เรียกว่า “ชุดโปรแกรม MIDlet” (MIDlet suit)

ประการต่อมาไฟล์ JAR สามารถดาวน์โหลด และติดตั้งบนอุปกรณ์ไร้สายผ่านสายเคเบิลอนุกรมที่ต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคล หรือผ่านเครือข่ายไร้สายก็ได้ เมื่อติดตั้งไฟล์ JAR แล้วผู้ใช้งานสามารถเลือก และสั่งกระทำแอปพลิเคชันจากเมนูเริ่มต้นได้

ไฟล์ที่เกี่ยวข้องในการสร้าง และจัดเตรียมแพ็คเกจของ MIDlet ได้แก่ไฟล์ manifest และไฟล์ application descriptor โดยนำไฟล์ manifest มาใช้อธิบายเนื้อหาข้างในของไฟล์ JAR ของชุดโปรแกรม MIDlet ส่วนไฟล์ application descriptor นำมาใช้อธิบายเนื้อหารายการเมนู ตำแหน่งไฟล์ JAR ในชุดโปรแกรม MIDlet ฯลฯ

Manifest

ในชุดโปรแกรม MIDlet จะต้องมีไฟล์ manifest รวมอยู่ด้วยเสมอเพื่ออธิบายเนื้อหาในไฟล์ JAR และข้อมูลอื่นๆเช่น ชื่อ รุ่นและผู้ผลิตโปรแกรม MIDlet

ไฟล์ manifest ประกอบด้วยรายการแอตทริบิวต์ MIDlet ที่แสดงในลักษณะคู่ซึ่งของชื่อ และค่า ซึ่งค้นกลางด้วยเครื่องหมาย : การใช้แอตทริบิวต์เหล่านี้จะช่วยให้นักพัฒนาสามารถอธิบายให้ผู้ใช้ใช้งานทราบว่า มี MIDlet ไคบ้างในชุดโปรแกรม MIDlet

ชื่อแอตทริบิวต์	รายละเอียด
MIDlet-Name	ชื่อของชุดโปรแกรม MIDlet
MIDlet-Version	รุ่นของชุดโปรแกรม MIDlet
MIDlet-Vendor	ผู้ผลิตชุดโปรแกรม MIDlet
MIDlet-Icon	ชื่อไฟล์ PNG ในไฟล์ JAR ใช้แทนชุดโปรแกรม MIDlet ไฟล์นี้จะใช้เมื่อซอฟต์แวร์จัดการแอปพลิเคชันแสดงไอคอนเพื่อระบุชุดโปรแกรมบนหน้าจอ
MIDlet-Description	คำอธิบายรายละเอียดของชุดโปรแกรม MIDlet
MIDlet-Info-URL	URL สำหรับค้นคำอธิบายของชุดโปรแกรม MIDlet เพิ่มเติม
MIDlet-<n>	ชื่อ ไอคอน และคลาสของ MIDlet ลำดับที่ n ใน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ไฟล์ JAR คำนวณด้วยเครื่องหมาย , โดยค่าต่ำสุดของ n เริ่มจาก 1 และไล่ไปตามลำดับ Name เพื่อใช้จำแนก MIDlet ให้ผู้ใช้งานทราบ Icon เป็นชื่อของคลาสที่ขยายคลาสของ MIDlet สำหรับ MIDlet ลำดับที่ n โดยคลาสจะต้องมี constructor แบบ no-args ที่ประกาศเป็นแบบ public
MIDlet-Jar-URL	URL ที่สามารถดาวน์โหลดไฟล์ JAR ได้
MIDlet-Jar-Size	ขนาดของไฟล์ JAR เป็นไบต์
MIDlet-Data-Size	ขนาดขั้นต่ำของพื้นที่เก็บข้อมูลแบบคงตัวที่ MIDlet ใช้ มีหน่วยเป็นไบต์ อุปกรณ์อาจจัดสรรเนื้อที่เพิ่มเติมให้ทั้งนี้ขึ้นอยู่กับอุปกรณ์ ค่าโดยปริยายคือ 0
MicroEdition-Profile	โพรไฟล์ของ J2ME ที่ต้องการ ใช้ฟอร์แมต และค่าเดียวกับพรีอเพอร์ตี microedition.profiles (เช่น MIDP-1.0)
MicroEdition-Configuration	คอนฟิกูเรชันของ J2ME ที่ต้องการ ใช้ฟอร์แมต และค่าเดียวกับพรีอเพอร์ตี microedition.configuration (เช่น CLDC-1.0)

ตาราง แอตทริบิวต์ต่างๆของไฟล์ manifest

ตัวอย่าง ไฟล์ manifest(MANIFEST.MF) ของโปรแกรม HelloWorld

MIDlet-Name : Hello World

MIDlet-Vendor : Sam Publishing

MIDlet-Version : 1.

MIDlet-1 :HelloWorld, /Icon.png,HelloWorld

MicroEdition-Configuration : CLDC-1.0

MIDlet-Data-Size: 0

ชื่อชุด โปรแกรม MIDlet คือ Hello World ผู้ผลิตคือ Sam Publishing และรุ่นของแพ็คเกจคือ 1.0 มีเฉพาะแอตทริบิวต์ MIDlet-1 เนื่องจากมี MIDlet เพียงตัวเดียวในชุดโปรแกรมนี้ หากมีมากกว่าหนึ่งตัว จะต้องใส่รายการ MIDlet-n เพิ่มเข้าไปโดย n ได้แก่ เลขลำดับต่อมาของโปรแกรม MIDlet อื่นๆ ในพิวต์ MIDlet-1 ค่าตัวแรกของ HelloWorld หมายถึง ชื่อของ MIDlet โดย /Icon.png คือ พารามิเตอร์ของไฟล์ไอคอน ลำดับที่สองคือชื่อคลาสของ MIDlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างแพ็คเกจแอปพลิเคชัน MIDlet

ไฟล์ JAR ทั่วไปในชุดโปรแกรม MIDlet จะประกอบไปด้วยไฟล์คลาสทั้งหมดในแอปพลิเคชัน MIDlet ไฟล์ทรัพยากร และไฟล์ manifest สำหรับไฟล์คลาสจะต้องคอมไพล์ไปก่อน ไฟล์ทรัพยากรมักจะเป็นไฟล์ข้อมูล และรูปภาพต่างๆที่ MIDlet มาใช้ขณะทำงาน

MIDlet ต่างๆในชุดโปรแกรม MIDlet อาจใช้ไฟล์คลาสร่วมกันได้ ทำให้คุณสามารถสร้างโปรแกรม MIDlet หลายๆโปรแกรม และเก็บไว้ในแพ็คเกจเดียวกัน โดยออกแบบให้ MIDlet ต่างๆในแอปพลิเคชันใช้ไฟล์คลาส และไฟล์ทรัพยากรร่วมกัน เพื่อลดขนาดของแพ็คเกจลงได้

ตัวอย่างการรวม MIDlet ต่างๆเข้าไปไว้ในไฟล์ JAR

```
c:> c:\jdk1.3\bin\jar cvmf MAMIFEST.MF HelloWorld.jar HelloWorld.class icon.png
```

jar เป็นโปรแกรมยูทิลิตี้ที่มากับ JDK1.3 ตัวเลือก cvmf นำสั่งมีความหมายดังนี้ c หมายถึงการสร้างไฟล์ใหม่ v หมายถึง ให้แสดงข้อความออกแบบไม่ย่อ m หมายถึง ให้อ่านข้อมูล manifest และ f ระบุชื่อไฟล์ JAR ที่ต้องการ

Application Descriptor

ไฟล์ application descriptor จะมีฟอร์แมตเหมือนกับไฟล์ manifest ทว่ามีวัตถุประสงค์การใช้งานที่แตกต่างกัน คือไฟล์ manifest ใช้เพื่อสร้างแพ็คเกจของชุดโปรแกรม MIDlet ขณะที่ไฟล์ application descriptor ใช้เพื่อจัดเตรียมชุดโปรแกรม MIDlet โดยเฉพาะในกรณีที่ต้องการแจกจ่ายแอปพลิเคชันแบบไร้สาย หรือที่เรียกว่า Over The-Air (OTA) deployment

ก่อนที่จะดาวน์โหลดไฟล์ JAR ลงในอุปกรณ์ โปรแกรมจัดการแอปพลิเคชัน (AMS) จะอ่านไฟล์ application descriptor เพื่อตรวจให้แน่ใจว่าชุดโปรแกรม MIDlet สามารถใช้ได้กับอุปกรณ์ที่จะติดตั้งแอปพลิเคชันนั้นๆ ซึ่งจะเป็นประโยชน์ในกรณีที่เครือข่ายไร้สายที่โรมมิ่งแบนด์วิดธ์ต่ำ หรือไม่เสถียรภาพ

แอตทริบิวต์ที่กำหนดไว้ล่วงหน้า

แอตทริบิวต์ที่จำเป็นต้องมีในไฟล์ application descriptor นั้นมี 7 แอตทริบิวต์คือ

MIDlet-Name

MIDlet-Version

MIDlet-Vendor

MIDlet-Jar-URL

MIDlet-Jar-Size

MicroEdition-Profile

MicroEdition-Configuration

และอีก 4 แอตทริบิวต์ที่อาจมี หรือไม่มีก็ได้คือ

MIDlet-Description

MIDlet-Icon

MIDlet-Info-URL

MIDlet-Data-Size

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนจะดาวน์โหลดไฟล์ JAR ลงในอุปกรณ์ โปรแกรมจัดการแอปพลิเคชัน (AMS) จะตรวจสอบแอตทริบิวต์บังคับทั้ง 7 ในไฟล์ application descriptor เพื่อดูว่าแอปพลิเคชันเหมาะสมกับอุปกรณ์หรือไม่ เช่นถ้าอุปกรณ์ไม่สนับสนุนโพรไฟล์ หรือรุ่นของโพรไฟล์ที่กำหนดไว้ ก็จะไม่ดาวน์โหลดไฟล์ JAR มา และหากขนาดของไฟล์ JAR ที่ระบุในแอตทริบิวต์ MIDlet-Jar-Size มีขนาดใหญ่เกินไปสำหรับอุปกรณ์นั้นๆก็จะไม่ดาวน์โหลดไฟล์ JAR เช่นกัน

ถึงแม้ว่าไฟล์ application descriptor และไฟล์ manifest จะมีวัตถุประสงค์ต่างกัน ทว่าค่าแอตทริบิวต์ MIDlet -Name, MIDlet -Version, MIDlet -Vendor ในไฟล์ทั้งสองจะต้องเหมือนกัน มิฉะนั้นจะไม่สามารถทำการดาวน์โหลดชุดโปรแกรม MIDlet ได้

การกำหนดแอตทริบิวต์ใหม่

นอกจากแอตทริบิวต์ที่กำหนดไว้แล้วยังสามารถกำหนดแอตทริบิวต์ขึ้นเพื่อใช้งานเองได้ในแอปพลิเคชัน โดยพิวด์ที่ผู้ใช้กำหนดขึ้นเองจะเริ่มต้นด้วย MIDlet- ไม่ได้

ค่าแอตทริบิวต์ทั้งหมดในไฟล์ manifest และไฟล์ application descriptor จะถูกอ่านโดยโปรแกรม MIDlet ขณะทำงาน เช่นค่าแอตทริบิวต์เฉพาะแอปพลิเคชัน (Language-Support Target-Device Display-Width และ Display-Height) จะถูกใช้เป็นพารามิเตอร์สำหรับคอนฟิกูเรชันค่า Display-Width และ Display-Height โปรแกรม MIDlet จะใช้ค่าพารามิเตอร์ที่อ่านค่าได้ในขณะทำงานในการปรับตัวเองให้เข้ากับอุปกรณ์ปลายทางประเภทต่างๆ สิ่งที่ต้องทำคือสร้างไฟล์ application descriptor เฉพาะอุปกรณ์สำหรับอุปกรณ์ปลายทางทุกประเภทเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้