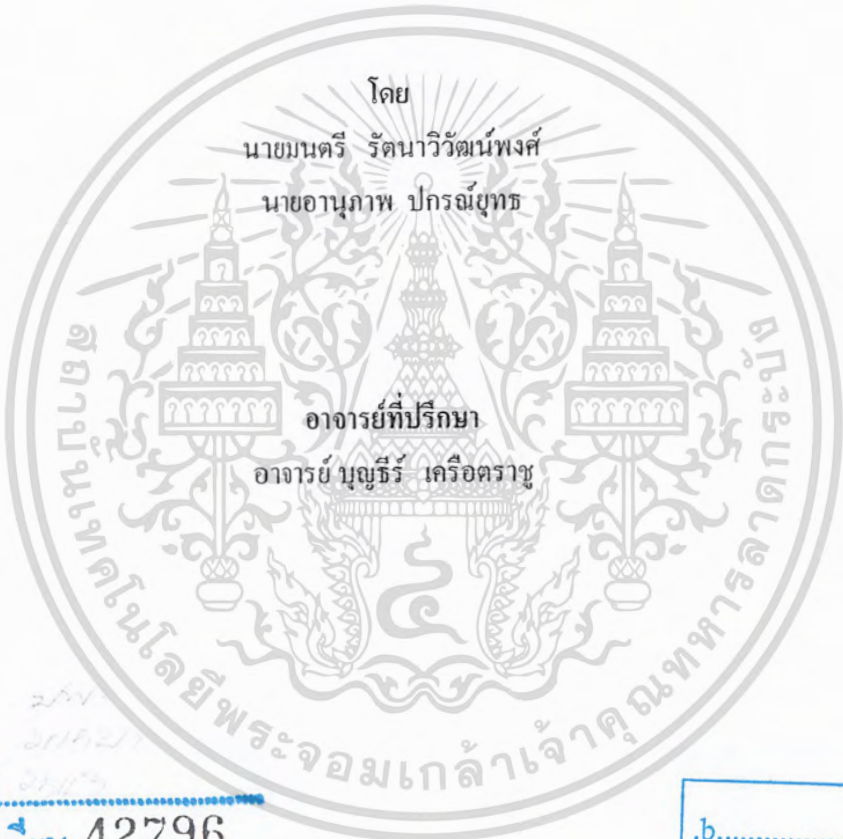


การประยุกต์ใช้ WAP สำหรับทำรายการธุรกิจ  
WAP Application for transaction



โดย  
นายมนตรี รัตนาวีวัฒน์พงศ์  
นายอนุภาพ ปกรณ์ยุทธ์  
อาจารย์ที่ปรึกษา  
อาจารย์ บุญธีร์ เครือคราช

เลขที่.....  
เลขทะเบียน 42796  
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปฏิญานินพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Handwritten signature or mark

ปริญญาโท ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้ WAP สำหรับทำรายการธุรกิจ

WAP Application for transaction

ผู้จัดทำ

1. นายมนตรี รัตนวิวัฒน์พงศ์ รหัสประจำตัว 40010579
2. นายอนุภาพ ปกรณ์ยุทธ รหัสประจำตัว 40011001



( ผศ. ดร. บุญธิร์ เครือตราฐ )

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การประยุกต์ใช้ WAP สำหรับทำรายการธุรกิจ

นายมนตรี รัตนาวิวัฒน์พงศ์ 40010579

นายอานุภาพ ปกรณ์ยุทธ 40011001

รศ.ดร. บุญธีร์ เครือตราฐ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2543

### บทคัดย่อ

โครงการนี้ เป็นการศึกษาเกี่ยวกับเทคโนโลยีเว็บ และภาษาจาวา เนื่องจากผู้จัดทำได้ตระหนักว่าเว็บเป็นเทคโนโลยี ที่เป็นประโยชน์ และคาดว่าจะใช้กันอย่างกว้างขวางในอนาคตอันใกล้นี้ ด้วยคุณสมบัติหลายๆประการ โครงการนี้ ได้นำเอาความรู้ที่ได้จากการศึกษา มาทำการพัฒนาเป็นโปรแกรมเพื่อนำไปใช้ประโยชน์ต่อไป

โครงการนี้ จะเป็นการพัฒนาโปรแกรม ในลักษณะของเซิร์ฟเวอร์ สำหรับให้บริการเกี่ยวกับการทำรายการสินค้า โดยใช้ Microsoft Access ในการจัดเก็บข้อมูล ซึ่งผู้ใช้บริการสามารถลงทะเบียนผ่านทางเว็บ และทำการ Update ข้อมูล โดยเว็บ นอกจากนี้ยังมีบริการส่ง message ระหว่างผู้ใช้และบริการตรวจสอบจดหมาย จาก POP3 Mail Server ผ่านทางโทรศัพท์มือถือ ที่มีความสามารถในการใช้งาน เว็บ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## WAP Application for transaction

Mr. Montri Rattanawiwatpong 40010579

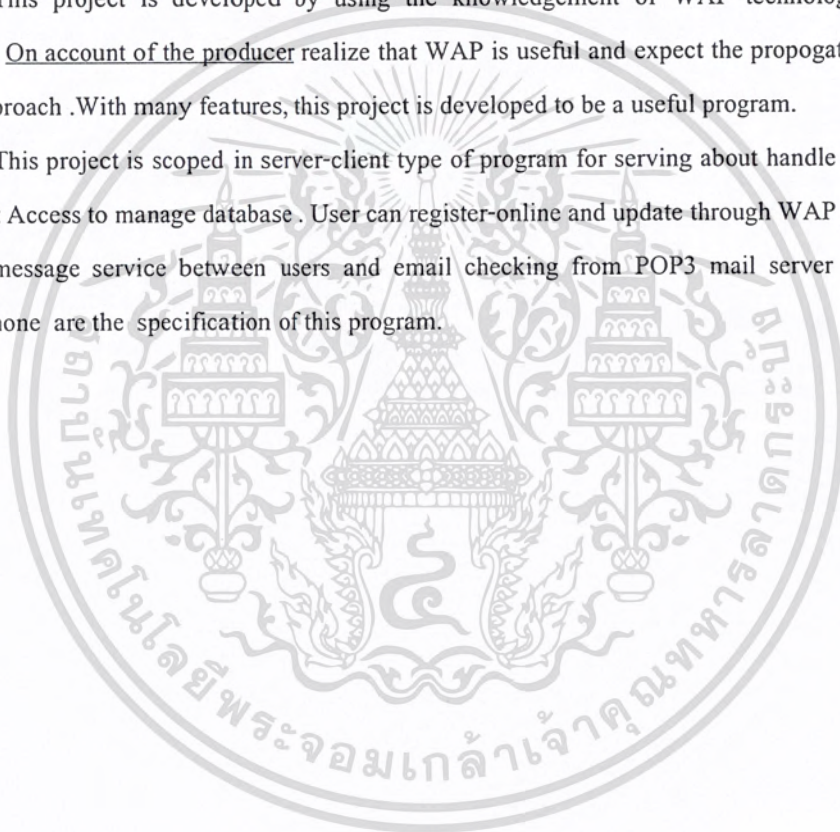
Mr. Arnupap Pakornyudh 40011001

Assoc. Prof. Dr. Boontee Klautrachu Advisor

### Abstract

This project is developed by using the knowledgement of WAP technology and JAVA language. On account of the producer realize that WAP is useful and expect the propogation of it in the future approach .With many features, this project is developed to be a useful program.

This project is scoped in server-client type of program for serving about handle stock by using Microsoft Access to manage database . User can register-online and update through WAP . In addition , sending message service between users and email checking from POP3 mail server through WAP moblie phone are the specification of this program.



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	VIII

### บทที่

1. บทนำ	1
2. ทฤษฎีและหลักการ	2
2.1 แอป	2
2.1.1 ความรู้พื้นฐานเกี่ยวกับเว็บ	2
2.1.2 แบบโครงสร้างของเว็บ	3
2.1.3 WAP Protocol Stack	4
2.1.4 WAP Application Environment	5
2.1.5 Wireless Session Protocol	5
2.1.6 Wireless Transaction Protocol	6
2.1.7 Wireless Transport Layer Security	6
2.1.8 Wireless Datagram Protocol	7
2.2 WAP Application Programming	7
2.3 Java Servlet	8
2.3.1 การนำ servlet ไปใช้งาน	8
2.3.2 Servlet Life Cycle	8
2.3.3 การเขียนโปรแกรม servlet	9
2.4 WEB Server	9
2.5 WAP Server	9
2.6 ระบบฐานข้อมูล ( Database System )	11
2.6.1 Database Management System ( DBMS )	11
2.6.2 JDBC ( Java Database Connectivity )	11
2.7 POP 3 Mail Server	16
2.7.1 หลักการทำงานของ POP3 Mail Server	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ ( ต่อ )

	หน้า
2.7.2 คำสั่งมาตรฐาน สำหรับใช้ติดต่อกับ POP3 Mail Server	17
2.8 Socket Programming	18
2.8.1 Internet Protocol ( IP )	18
2.8.2 User Datagram Protocol ( UDP )	19
3. การดำเนินงาน	20
3.1 ระบบการทำรายการสินค้า	20
3.1.1 การวิเคราะห์และการออกแบบระบบการทำรายการสินค้า	20
3.1.1.1 หลักการ	20
3.1.1.2 ขอบเขตการให้บริการ	20
3.1.1.3 กิจกรรมต่างๆ	20
3.1.1.4 การลงทะเบียนคลังสินค้า	21
3.1.1.5 การให้อนุญาตทำการ	21
3.1.1.6 การนำสินค้าเข้า	21
3.1.1.7 การนำสินค้าออก	21
3.1.1.8 การดูสินค้าคงเหลือ	21
3.1.1.9 การแจ้งเตือนรูปแบบการนำสินค้าออก	21
3.1.1.10 การแจ้งเตือนสินค้าหมด	22
3.1.2 การทำการรายการสินค้า	22
3.1.2.1 การกรอกรายละเอียดของบริษัท และ Admin ของบริษัท	22
3.1.2.2 กรอกรายละเอียดต่าง ๆ เกี่ยวกับตัวสินค้า ตามขั้นตอน	24
3.1.2.3 ผู้ใช้รายการสินค้าทำการกรอกข้อมูลส่วนตัว	29
3.1.2.4 ผู้ที่จะเช็คสินค้า ส่งข้อความเพื่อขออนุญาต จาก Admin ของ บริษัท	29
3.2 Check Mail	36
3.2.1 ลักษณะทั่วไปของโปรแกรมตรวจสอบจดหมายอิเล็กทรอนิกส์	36
3.2.2 ส่วนประกอบของ Check Mail System	36
3.2.3 รูปแบบการทำงานของ การ ตรวจสอบจดหมาย มีดังนี้	38
3.2.4 การเขียนโปรแกรมสำหรับ Check Mail System	39
3.2.5 การทดสอบนำ Check Mail System มาใช้งาน	41
4. เครื่องมือที่ใช้ในโครงการ	43
4.1 Java Servlet Development Kit	43

## สารบัญ ( ต่อ )

	หน้า
4.1.2 การติดตั้ง Java Servlet Development Kit	43
4.1.3 การใช้งานโปรแกรม servletrunner	43
4.2 Nokia Wap Toolkit	44
4.2.1 ส่วนประกอบของ Nokia WAP Toolkit	44
4.2.2 Toolkit หลักที่จำเป็นต่อการใช้งาน	45
4.2.2.1 Phone Simulator	45
4.2.2.2 Editor	46
4.2.2.3 Debugging Aids	46
4.2.3 การติดต่อระหว่าง Phone Simulator กับเซิร์ฟเวอร์	47
4.2.4 การใช้งานชุดโปรแกรม Nokia WAP Toolkit	48
4.2.4.1 การใช้งานชุดโปรแกรม Nokia WAP Toolkit	48
4.2.4.2 ขั้นตอนการติดตั้งชุดโปรแกรม Nokia WAP Toolkit	48
4.3 Microsoft Access 97	48
4.3.1 การติดตั้งโปรแกรม Microsoft Access 97 สามารถทำได้โดย	48
4.3.2 การติดตั้ง ODBC Driver	49
4.4 Tomcat Web Server	51
4.5 EditPlus 2.1a	52
5. สรุปและวิจารณ์ผลการทดลอง	52
เอกสารอ้างอิง	53
ภาคผนวก	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 ตัวอย่างเครือข่าย WAP	3
รูปที่ 2.2 WAP Model	4
รูปที่ 2.3 WAP Protocol Stack	5
รูปที่ 2.4 การติดต่อ JDBC โดยรูปแบบที่1 JDBC-ODBC Bridge	12
รูปที่ 2.5 การติดต่อ JDBC โดยรูปแบบที่2 Native-API, partly Java driver	14
รูปที่ 2.6 การติดต่อ JDBC โดยรูปแบบที่3 Network-protocol, all-Java driver	14
รูปที่ 2.7 การติดต่อ JDBC โดยรูปแบบที่4 Native-protocol, all-Java driver	15
รูปที่ 3.1	23
รูปที่ 3.2 หน้าจอการกรอกรายละเอียดเบื้องต้นของสินค้า	25
รูปที่ 3.3 หน้าจอการกรอกรายละเอียดของแต่ละ Attribute	26
รูปที่ 3.4 หน้าจอการกรอกรหัส Attribute	27
รูปที่ 3.5 รูปการใส่ข้อมูล	28
รูปที่ 3.6 หน้าจอการ Sign in ผ่านทาง WAP	29
รูปที่ 3.7 การส่งข้อความหา Admin เพื่อขอ Authorize	29
รูปที่ 3.8 หน้าจอการให้ Authorization	30
รูปที่ 3.9 หน้าจอการเลือกสินค้าที่จะทำรายการ	30
รูปที่ 3.10 การเลือกประเภทของสินค้านั้น ๆ	30
รูปที่ 3.11 เมนูเลือกทำรายการ	31
รูปที่ 3.12 หน้าจอเตือนสินค้าไม่พอ	32
รูปที่ 3.13 หน้าจอเตือนเรื่อง FIFO	32
รูปที่ 3.14 หน้าจอเตือนสินค้าใกล้หมด	33
รูปที่ 3.15 หน้าจอหลังกด View	33
รูปที่ 3.16 หน้าจอแสดงจำนวนข้อความ	33
รูปที่ 3.17 หน้าจอการเลือกจัดการกับข้อความ	34
รูปที่ 3.18 หน้าจอเลือกอ่านข้อความ	34
รูปที่ 3.19 หน้าจอแสดงผลการอ่านข้อความ	34
รูปที่ 3.20 หน้าจอแบบฟอร์มการส่งข้อความ	35
รูปที่ 3.21 หน้าจอการเลือกลบข้อความ	35
รูปที่ 3.22 แสดง Class Diagram ของ Check Mail System	37

## สารบัญรูป ( ต่อ )

รูปที่ 3.23 แสดงลำดับการทำงาน ( Sequence Diagram )	39
รูปที่ 3.24 แสดงหน้าจอแรกที่เข้าสู่โปรแกรมตรวจสอบจดหมาย	41
รูปที่ 3.25 รูปแสดงการใส่ข้อมูลของผู้ใช้	41
รูปที่ 3.26 เลือก Go!! เพื่อให้ MainMail ทำการติดต่อกับเซิร์ฟเวอร์	42
รูปที่ 3.27 รูปหน้าจอแสดงตัวเลือกสำหรับเลือกข้อความ	42
รูปที่ 3.28 ตัวอย่างแสดงข้อความที่ไม่สามารถอ่านได้	42
รูปที่ 3.29 ตัวอย่างข้อความที่สามารถอ่านได้	42
รูปที่ 4.1 แสดง Blueprint Phone Simulator	45
รูปที่ 4.2 Nokia 7110 Phone Simulator	45
รูปที่ 4.3 แสดงรูป Editor	46
รูปที่ 4.4 แสดงการติดต่อระหว่าง โคลนเข้ากับเซิร์ฟเวอร์ 3 วิธี	47
รูปที่ 4.5 แสดงการเรียกติดตั้ง ODBC Driver	49
รูปที่ 4.6 การเพิ่มฐานข้อมูลสำหรับการใช้งาน	49
รูปที่ 4.7 เลือกสร้าง Microsoft Access Driver สำหรับสร้างฐานข้อมูลใหม่	50
รูปที่ 4.8 การตั้งชื่อฐานข้อมูลใหม่	50
รูปที่ 4.9 การเลือกชื่อไฟล์ ฐานข้อมูล	51

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 ตาราง Admin	23
ตารางที่ 3.2 ตาราง Username_Password	23
ตารางที่ 3.3 ตาราง S0011001_Message	24
ตารางที่ 3.4 ตารางตัวอย่าง KMITL	25
ตารางที่ 3.5 ตารางตัวอย่าง KMITL_Pen	26
ตารางที่ 3.6 ตารางตัวอย่าง KMITL_Pen_Brand	27
ตารางที่ 3.7 ตารางตัวอย่าง KMITL_Pen_Red_Pentel	29
ตารางที่ 3.8 ตัวอย่างตาราง KMITL_Pen_Red_Pentel หลังจากทำการ In	31



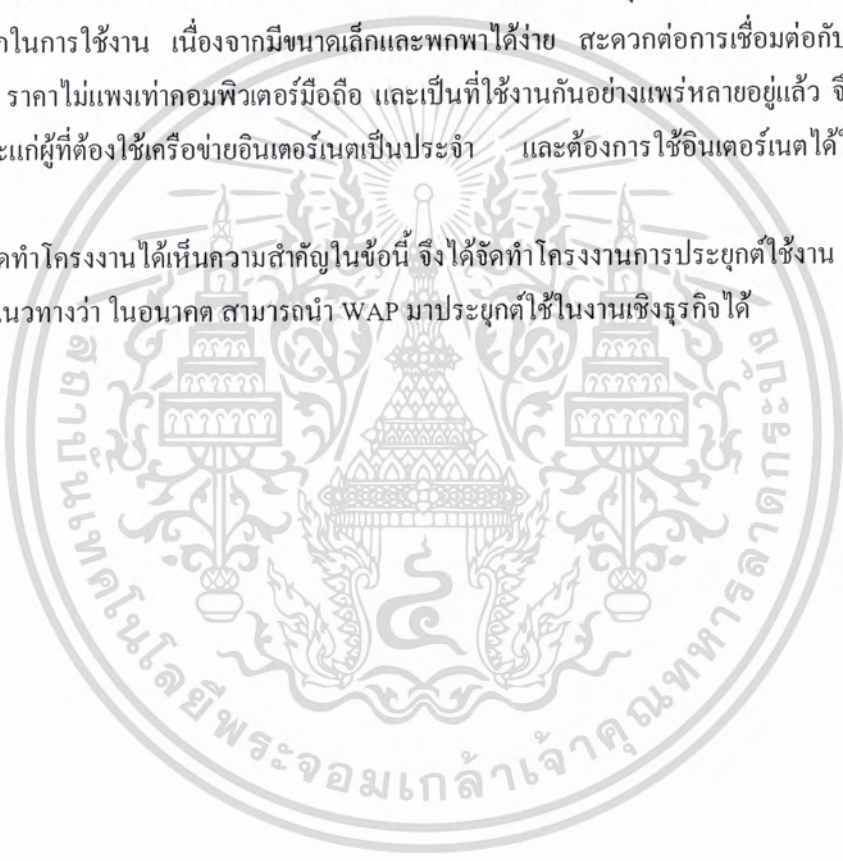
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

ในปัจจุบันนี้ การใช้อินเทอร์เน็ตเป็นที่นิยมกันอย่างแพร่หลาย ทั้งในการติดต่อสื่อสาร การเสาะหาข้อมูลต่างๆ และอื่นๆอีกมากมาย เนื่องจากเป็นระบบเครือข่ายที่สามารถติดต่อสื่อสารกันได้ทั่วโลก แต่การใช้อินเทอร์เน็ตโดยทั่วไป จะต้องใช้ผ่านเครื่องคอมพิวเตอร์ซึ่งไม่สะดวกต่อการเคลื่อนย้าย สำหรับคอมพิวเตอร์มือถือ ( โน้ตบุค ) ถึงแม้จะสามารถนำไปใช้ในสถานที่ต่างๆได้ แต่ก็ยังมีราคาแพง ไม่สะดวกต่อการพกพา และไม่สะดวกต่อการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต จึงได้มีการพัฒนาระบบเวปขึ้นมา เพื่อใช้ในการใช้งานระบบอินเทอร์เน็ตด้วยโทรศัพท์มือถือ หรืออุปกรณ์สื่อสารไร้สายอื่นๆ ซึ่งมีความสะดวกในการใช้งาน เนื่องจากมีขนาดเล็กและพกพาได้ง่าย สะดวกต่อการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต ราคาไม่แพงเท่าคอมพิวเตอร์มือถือ และเป็นที่ใช้ร่วมกันอย่างแพร่หลายอยู่แล้ว จึงเป็นทางเลือกที่เหมาะสมแก่ผู้ที่ใช้เครือข่ายอินเทอร์เน็ตเป็นประจำ และต้องการใช้อินเทอร์เน็ตได้ในทุกสถานที่

ทางผู้จัดทำโครงการได้เห็นความสำคัญในข้อนี้ จึงได้จัดทำโครงการการประยุกต์ใช้งาน WAP ขึ้นมา เพื่อ เป็นแนวทางว่า ในอนาคต สามารถนำ WAP มาประยุกต์ใช้ในงานเชิงธุรกิจได้



## บทที่ 2

### ทฤษฎีและหลักการ

โครงการนี้ ต้องใช้ทฤษฎีและหลักการต่างๆดังนี้

#### 2.1 แอป

##### 2.1.1 ความรู้พื้นฐานเกี่ยวกับแอป

WAP มีชื่อเต็ม ๆ อย่างเป็นทางการว่า “Wireless Application Protocol” ซึ่งเป็นมาตรฐานเปิดที่ใช้ในการรับส่งข้อมูลบนระบบเครือข่ายโทรศัพท์เคลื่อนที่รวมถึง Telephony Services ซึ่งก็คือบริการทางโทรศัพท์แบบอื่น ๆ บนโทรศัพท์เคลื่อนที่และอุปกรณ์พกพาแบบไร้สาย เช่น PDA, Palm และ เพจเจอร์ เป็นต้น

เนื่องจาก WAP เป็นมาตรฐานเปิดจึงทำให้อุปกรณ์ที่ผลิตขึ้นมาตามมาตรฐานของ WAP ทุกชนิดสามารถสื่อสารกับอุปกรณ์หรือเครือข่ายที่ใช้โปรโตคอลต่างกัน ได้ เช่น TCP/IP, UDP/IP นอกจากนี้ WAP ยังสามารถทำงานได้บนระบบต่าง ๆ มากมาย ไม่ว่าจะเป็น GSM, CDMA, PDC ฯลฯ

WAP Forum ได้ถูกจัดตั้งขึ้นโดยบริษัทใหญ่ 4 บริษัท คือ Nokia, Ericsson, Motorola, Phone.com เมื่อปี ค.ศ. 1997 โดยมีจุดมุ่งหมายคือ

- นำเสนอมาตรฐานที่เหมาะสม
- ไม่ขัดกับมาตรฐานของเครือข่ายไร้สายเพียงระบบใดระบบหนึ่งเท่านั้น
- ใช้ได้กับมาตรฐานการสื่อสารมากมายหลายชนิด
- เปิดกว้างสำหรับทุกคน
- ใช้ได้กับอุปกรณ์มากมายหลายชนิด
- รองรับเครือข่าย และมาตรฐานการสื่อสารที่จะเกิดขึ้นใหม่ในอนาคต

ลักษณะของ Waplications เราอาจแบ่งได้เป็น 3 ลักษณะใหญ่ ๆ คือ

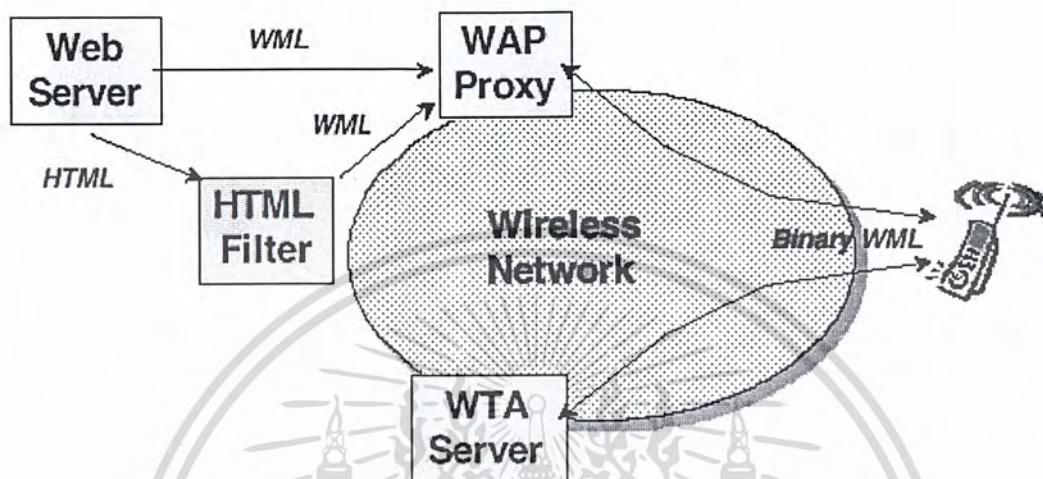
- Information Service  
เป็นการเข้าถึงข้อมูล โดยไม่มีการทำ Transaction นั่นก็คือการเข้าไปอ่านข้อมูลต่าง ๆ (Read Only) เช่นเดียวกับการอ่านข้อมูลบนอินเทอร์เน็ต โดยจะทำการผ่าน WAP Browser โดยใช้มาตรฐานต่าง ๆ ของอินเทอร์เน็ตเป็นหลัก
- Transaction Service  
เป็น Application ที่มีจะเกิดขึ้นหลังจากการทำ Information Service คือมีการทำรายการบางอย่างด้วย เช่น การจองตั๋วหนัง การโอนเงิน การทำ E-commerce
- Corporate IT Applications  
เป็นการเข้าไปสู่ระบบ IT ขององค์กรต่าง ๆ ซึ่งจะเป็นการเข้าไปสู่เครือข่ายที่ปิดเช่น เข้าไปในระบบการสั่งซื้อขายสินค้าของบริษัทใดบริษัทหนึ่ง หรือการเข้าไปในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีเมลล์ของบริษัท ซึ่ง Application แบบนี้ค่อนข้างจะต้องมีความปลอดภัยสูงกว่าในสองประเภทแรก

### 2.1.2 แบบโครงสร้างของ WAP

โครงสร้างของเครือข่าย WAP จะมีทั้งส่วนที่เป็น Fixed Network ซึ่งเหมือนกับเครือข่าย WWW ทั่วไป และแบบที่เป็น Wireless Network ซึ่งเป็นเครือข่ายของโทรศัพท์เคลื่อนที่



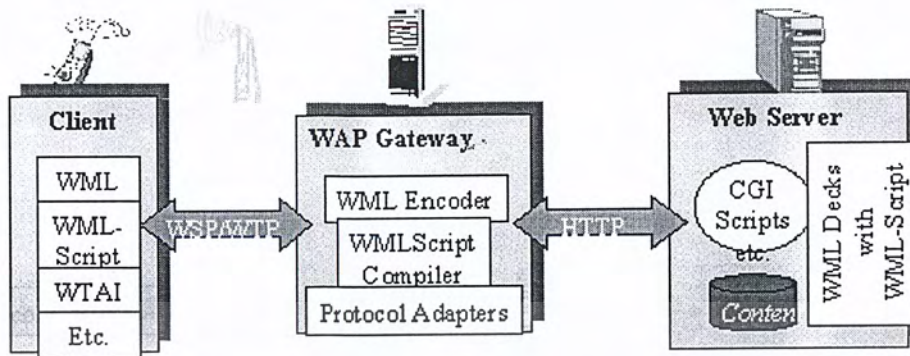
รูปที่ 2.1 ตัวอย่างเครือข่าย WAP

WAP Gateway จะเป็นประตูที่ทำหน้าที่เชื่อม โยงระหว่างเครือข่ายทั้งสองแบบ โดยมีหน้าที่หลักคือ

1. เป็นตัวกำหนดว่าอุปกรณ์ WAP ใช้มาตรฐานอะไรในการเชื่อมต่อ โดยขึ้นอยู่กับ Network Layer ข้างล่าง Network Layer เป็นแบบที่สามารถรองรับ IP ได้ ก็จะใช้ UDP/IP ไม่เช่นนั้นก็จะใช้ WDP
2. ทำหน้าที่ในการแปลงโปรโตคอลที่แตกต่างกันระหว่างเครือข่ายแบบ Fixed Network และ Wireless Network เพื่อให้การรับส่งข้อมูลกันเป็นไปได้ ตัวอย่างเช่น การแปลง WSP ซึ่งอยู่บนเครือข่ายไร้สายให้เป็น HTTP เพื่อส่งข้อมูลไปยังตัว Server ซึ่งหลังจากที่ Server ได้ทำการประมวลผลเรียบร้อยแล้ว ก็จะส่ง HTTP กลับมาที่ WAP Gateway แล้ว WAP Gateway จะทำหน้าที่แปลง HTTP กลับไปเป็น WSP เพื่อส่งข้อมูลไปยังโทรศัพท์เคลื่อนที่
3. ทำหน้าที่ในการ Cache ข้อมูลเกี่ยวกับความสามารถและความต้องการพิเศษ (Capability and Preference Information หรือ CPI) ของอุปกรณ์ WAP เพื่อที่จะทำให้การทำงานเร็วขึ้นโดยใช้ CPI เป็นเสมือน Profile และจะทำการอัปเดตเมื่ออุปกรณ์ WAP เป็นผู้กำหนดเท่านั้น และ Gateway จะต้องทำการส่ง CPI ไปยัง Origin Server ด้วยเพื่อเป็นการทำให้มันใจว่าฟังก์ชันใดบ้างที่สามารถทำงานได้ และฟังก์ชันใดบ้างที่ไม่สามารถทำงานกับอุปกรณ์ WAP นั้น ๆ ได้
4. ทำหน้าที่ในการจัดการเรื่องความปลอดภัย โดยการใช้ WTLS บน WAP Stack
5. ทำหน้าที่ในการ Encode และ Decode ข้อมูลเนื่องจากการส่งข้อมูลในเครือข่ายโทรศัพท์เคลื่อนที่ ซึ่งมีแบนด์วิธต่ำต้องมีการบีบอัดขนาดของข้อมูลลงก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทำหน้าที่ในการแปลงโปรโตคอลอื่น ๆ นอกจาก HTTP ให้เป็น WSP เพื่อให้ใช้งานได้กับบางโปรโตคอลเช่น POP3, SMTP, MAPI, IMAP4 เพื่อใช้ใน Application อื่น ๆ เช่น การรับส่งอีเมลเป็นต้น



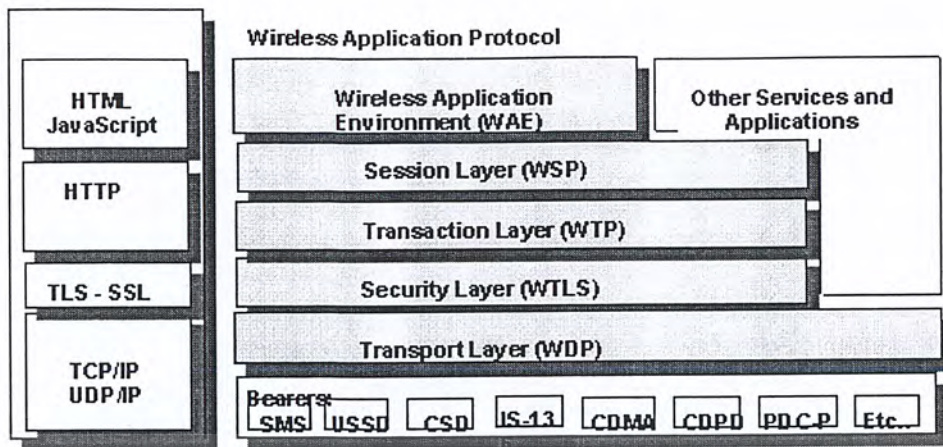
รูปที่ 2.2 WAP Model

### 2.1.3 WAP Protocol Stack

WAP Protocol Stack ถูกสร้างขึ้นมาเพื่อเป็นมาตรฐานสำหรับการสื่อสารในระบบ WAP และเพื่อทำให้การพัฒนาที่มีความเป็นมาตรฐาน จึงใช้ OSI model (Open System Interconnection model) เป็นพื้นฐาน แบ่งได้เป็น 5 ระดับ ดังนี้คือ

1. ระดับ Application – Wireless Application Environment (WAE) เป็นตัวกำหนดสิ่งแวดล้อมของ Application เช่น Format ต่าง ๆ รวมถึงตัว WAP Browser ด้วย
2. ระดับ Session – Wireless Session Protocol (WSP) เป็นตัวกำหนดกฎเกณฑ์ในการรับส่งข้อมูลเพื่อให้ฟังก์ชันต่าง ๆ สามารถทำงานได้อย่างถูกต้อง
3. ระดับ Transmission – Wireless Transaction Protocol (WTP) ทำหน้าที่เพื่อความน่าเชื่อถือในการสื่อสารข้อมูล
4. ระดับ Security – Wireless Transport Layer Security (WTLS) ทำหน้าที่ควบคุมความปลอดภัยในการรับส่งข้อมูล
5. ระดับ Transport – Wireless Datagram Protocol (WDP) ทำให้ Wapplication สามารถทำงานได้บนเครือข่ายโทรศัพท์เคลื่อนที่ ที่มีมาตรฐานแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 WAP Protocol Stack

#### 2.1.4 WAP Application Environment (WAE)

WAE เป็นระดับที่กำหนดขอบเขตความสามารถของ Wapplication โดยจะเป็นตัวกำหนดรูปแบบของข้อมูลต่าง ๆ และส่วนของการติดต่อกับผู้ใช้ โดย WAE นั้นจะแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ User Agents และ อีกส่วนคือ ส่วนของ Service และ Format ต่าง ๆ ดังนี้

1. Service & Format – ได้แก่ Format ต่าง ๆ ที่ใช้ใน Wapplication ซึ่ง หลัก ๆ ก็ได้แก่ WML (Wireless Markup Language) ซึ่งมีพื้นฐานมาจาก XML โดย Tag ของมันจะมีรูปแบบคล้ายกันกับ HTML ที่ใช้ใน Web page ทั่วไป WmlScript ซึ่ง ใช้ JavaScript เป็นพื้นฐานเพื่อเพิ่มขีดความสามารถให้กับ WML นอกจากนี้ยังมี URL(Universal Resource Locators) เพื่อใช้เป็นตัวเข้าสู่ Content Server
2. WML User Agent – คือ WAP Browser ซึ่งเป็นส่วนที่ควบคุมการทำ User Interface โดยการแปลง WML และ WMLScript ให้แสดงผลปรากฏบนหน้าจอของโทรศัพท์มือถือได้

#### 2.1.5 Wireless Session Protocol (WSP)

ทำหน้าที่เหมือนกับ HTTP ในระบบเครือข่ายอินเทอร์เน็ต แต่ถูกแปลงให้เป็นไบนารี เพื่อให้สามารถรองรับได้กับเครือข่ายของโทรศัพท์มือถือซึ่งมีแบนด์วิดธ์ต่ำ โดยมีฟังก์ชันบางอย่างที่ถูกเพิ่มมาจาก HTTP เช่น

1. Header Encoding – เป็นการบีบอัดข้อมูลเพื่อให้มีขนาดเล็กลง แม้แต่ค่า Header ก็ต้องมีขนาดเล็กลงด้วยเพื่อเป็นการเพิ่มประสิทธิภาพในการรับส่งข้อมูลในเครือข่ายที่มีแบนด์วิดธ์ต่ำ
2. Capability Negotiation – เป็นกลไกที่ WAP Gateway ใช้ในการพิจารณาว่าไคลเอนต์ที่ส่งสัญญาณ Request มาสามารถรองรับออปชันและฟังก์ชันของ WAP ได้หรือไม่ และยังใช้สำหรับการแลกเปลี่ยน Header Code Page อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Header Caching – ทำให้ไม่ต้องมีการส่ง Header ซ้ำ ๆ ไปทุก ๆ ครั้งที่มีการ Request หลัง จาก Session ระหว่างไคลเอนต์กับเซิร์ฟเวอร์ได้ถูกสร้างขึ้นแล้ว
4. Long-lives Session – ทำให้ Session มีระยะเวลายาวนานขึ้น รวมทั้งยังสนับสนุนการ Resume และ Suspend ซึ่งจำเป็นมากสำหรับโทรศัพท์มือถือซึ่งมักจะมีช่วงสัญญาณขาดหาย เป็นช่วง ๆ โดย ถ้าสามารถ Resume ได้ก็จะไม่ต้องมีการทำ Capability Negotiation ใหม่

### 2.1.6 Wireless Transaction Protocol (WTP)

WTP เป็นตัวควบคุมการรับส่งข้อมูล ทำให้เกิดความน่าเชื่อถือในการสื่อสาร โดยจะจัดให้มีการส่งข้อมูลใหม่เมื่อเกิดการสูญหายระหว่างส่ง ควบคุมไม่ให้เกิดความซ้ำซ้อน โดยจะมองว่าการส่งข้อมูลแต่ละครั้งเป็น Transaction

WAP Forum ได้กำหนดระดับของ Transaction ไว้ 3 ระดับคือ

Class 0: Unreliable one – way request ใน Transaction ระดับนี้จะไม่มีการส่งข้อมูลใหม่ถ้าเกิดการสูญหาย

Class 1: Reliable one-way requests ใน Transaction ระดับนี้ผู้รับจะทำการส่ง Acknowledgement กลับมายังผู้ส่ง ซึ่งในระดับนี้จะทำให้เกิดความมั่นใจได้ว่าข้อมูลจะต้องถึงมือผู้รับ

Class 2:Reliable two-way request-reply transaction ระดับนี้จะมีการป้องกันการส่งข้อมูลซ้ำซ้อนที่เกิดขึ้นใน Class 1 เนื่องจากในกรณีที่ Acknowledgement เกิดการสูญหาย โดยวิธีการแก้ปัญหาก็คือ เพิ่มฟังก์ชันของการ Response และ Acknowledgement จากผู้ขอ กลับไปยังผู้รับ เพื่อยืนยันว่าได้รับข้อมูลเรียบร้อยแล้ว

### 2.1.7 Wireless Transport Layer Security (WTLS)

เป็น Layer ที่ทำหน้าที่เกี่ยวกับการควบคุมความปลอดภัยในการรับส่งข้อมูล โดยถูกพัฒนามาจาก Transport Layer Security ซึ่งปรับปรุงให้ใช้ได้กับเครือข่ายที่มีแบนด์วิธต่ำโดยเฉพาะ ซึ่งจะเป็นอุปสรรคที่สามารถจะเลือกใช้หรือไม่ก็ได้ เนื่องจาก Application บางอย่างก็ไม่จำเป็นต้องมีความปลอดภัยสูง

WTLS แบ่งได้เป็น

1. Data Integrity หมายถึง ความถูกต้องของข้อมูล เพื่อให้แน่ใจได้ว่าข้อมูลจะไม่ถูกแก้ไข
2. Privacy หมายถึง การเข้ารหัส เพื่อป้องกันการไม่ให้ผู้ที่ลักลอบขโมยข้อมูลสามารถเข้าใจข้อมูลได้
3. Authentication เป็น ฟังก์ชันที่จะทำหน้าที่ตรวจสอบผู้ที่เข้ามาใช้ Wapplication บน Server
4. Denial – of – service คือ จะไม่ทำการประมวลผลข้อมูลที่ไม่ได้ถูกตรวจสอบอย่างถูกต้อง

ในการส่งข้อมูลนั้น เราจะใช้พอร์ตที่แตกต่างกันระหว่าง Application ที่ต้องใช้ระบบควบคุมความปลอดภัย กับ Application ที่ไม่ต้องการใช้ เช่น ใช้พอร์ต 9202 สำหรับ Secured Connectionless และ 9200 สำหรับ Unsecured Connectionless

WTLS จะเป็นการควบคุมความปลอดภัยระหว่าง โคลเอนต์ กับ WAP Gateway ซึ่งเป็นส่วนของ Wireless Network เท่านั้น ไม่ได้ครอบคลุมไปถึงเครือข่ายแบบ Fixed Network ดังนั้น ในการใช้งานจริงต้องใช้ WTLS ควบคู่ไปกับ SSL (Secure Sockets Layer) ด้วย

### 2.1.8 Wireless Datagram Protocol (WDP)

เครือข่ายโทรศัพท์เคลื่อนที่ในปัจจุบันนั้นมีหลายระบบ ดังนั้น WDP จึงถูกออกแบบขึ้นมาเพื่อช่วยให้ Wapplication สามารถทำงานได้บนระบบที่แตกต่างกันดังกล่าว โดยมีหลักการก็คือ อินเทอร์เน็ตให้กับระดับที่สูงกว่าจะไม่มีเปลี่ยนแปลงไม่ว่า Network layer หรือ Bearer จะเป็นอะไรก็ตาม WDP จะปรับตัวเองให้เข้ากับ Bearer โดยถ้า Bearer สามารถรองรับ TCP/IP ได้ก็จะใช้ UDP (User Datagram Protocol) นั่นก็คือจะเป็น UDP/IP แต่ถ้า Bearer ไม่สามารถรองรับ TCP/IP ได้ก็จะใช้ WDP แทน

### 2.2 Web Application Programming

จากที่ได้กล่าวมาในหัวข้อที่ 1 หลักการทำงานของ WAP Application คือ โคลเอนต์ส่งคำร้องขอไปที่เซิร์ฟเวอร์ แล้วรอรับ WML Page จากเซิร์ฟเวอร์ เพื่อนำมาแสดงผลให้ผู้ใช้ ที่มาของ WML Page เราสามารถแบ่งเป็น 2 ประเภทใหญ่ได้คือ

1. WML Page ที่ถูกเก็บอยู่ในไฟล์ในเครื่องเซิร์ฟเวอร์อยู่แล้ว เมื่อโคลเอนต์ต้องการรับ WML Page ใด ก็ส่งข้อความร้องขอ ที่ระบุ WML Page นั้นมาที่เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับคำร้องขอ WML Page ก็จะทำการค้นหา และส่ง WML Page ไปให้โคลเอนต์ตามต้องการ หรือถ้าไม่สามารถหา WML Page ที่โคลเอนต์ต้องการได้ ก็จะแจ้งข้อความแสดงข้อผิดพลาดไปให้โคลเอนต์ WML Page รูปแบบนี้ เรียกว่าเป็นแบบ Static Content
2. WML Page แบบที่ถูกสร้างขึ้น เมื่อมีคำร้องขอมาจากโคลเอนต์ โดยเนื้อหาใน WML Page นั้น จะถูกสร้างขึ้น โดยขึ้นกับข้อมูลที่โคลเอนต์ส่งมาด้วย ซึ่งจะเรียกว่าเป็นแบบ Dynamic Content

การที่ เซิร์ฟเวอร์ จะสามารถสร้าง WML Page แบบที่ 2 ได้ นั้น เครื่องเซิร์ฟเวอร์จำเป็นต้องมีความสามารถ ที่เรียกว่า Server – Side Application ได้ ซึ่งสามารถใช้เทคนิคต่างๆ ดังต่อไปนี้

- Common Gateway Interface
- Server Extension APIs
- Active Server Pages
- Server-Side JavaScript
- Java Servlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่จะใช้งาน Java Servlet ได้นั้น จำเป็นจะต้องมีการติดตั้งโปรแกรม ที่รองรับความสามารถของ Servlet นี้ ซึ่งอาจเป็น Web Server ที่มีความสามารถในการรองรับความสามารถนี้ในตัวอยู่แล้ว เช่น Tomcat Web Server , Java Web Server หรืออาจจะอยู่ในรูปของ Plug – In ที่นำมาใช้ร่วมกับ Web Server เช่น J-Serv ซึ่งนำมาใช้กับ Apache Web Server

### 2.3 Java Servlet

ในที่นี้ เป็นการนำโปรแกรม ที่เขียนขึ้นด้วยภาษาจาวา มาทำการประมวลผล และใช้งานโดยเซิร์ฟเวอร์ เพื่อใช้เพิ่มความสามารถในการทำงาน , คำานวน ,และสร้างผลลัพธ์ให้เซิร์ฟเวอร์

Java Servlet เป็น web-scripting technology ที่มีลักษณะคล้ายกับ Netscape server-side JavaScript (SSJS) หรือ Microsoft Active Server Pages (ASP) หรือ Common Gateway Interface แต่มีความแตกต่างกันที่ Java Servlet เน้นการใช้งานของภาษาจาวา ซึ่งเป็นภาษาที่เน้นเรื่องการทำงานในลักษณะที่เป็นออบเจ็ค (object-oriented style) ซึ่งช่วยทำให้ง่ายต่อการพัฒนาโครงการใหญ่ ๆ ตลอดจนสามารถนำส่วนประกอบต่าง ๆ กลับมาใช้ได้อีก (software reusable) จุดเด่นที่สำคัญของ Java Servlet คือสามารถทำงาน ได้โดยไม่ขึ้นอยู่กับผู้ผลิตซอฟต์แวร์รายใดรายหนึ่ง โดยเฉพาะ ซึ่งโดยทั่วไปเทคโนโลยีต่าง ๆ มักจะออกมาในลักษณะของผลิตภัณฑ์จากบริษัทผู้ผลิตแห่งใดแห่งหนึ่ง แต่ Java Servlet ใช้ลักษณะของข้อกำหนด ซึ่งกำหนดโดยบริษัท Sun Microsystems ดังนั้นผู้ผลิตซอฟต์แวร์จึงสามารถอ้างอิงข้อกำหนดที่กำหนดขึ้น ผลิต Java Servlet Runner ( โปรแกรม ที่ทำให้ Java Servlet สามารถใช้งานได้ ) ขึ้นมาใช้กับแพลตฟอร์มใดก็ได้

#### 2.3.1 การนำ servlet ไปใช้งาน

โปรแกรม Servlet เป็นโปรแกรม ที่ทำงานภายใต้พื้นฐานของการทำงานแบบ โคลเอนท์ – เซิร์ฟเวอร์ โดยจะทำงานอยู่บนฝั่งของเซิร์ฟเวอร์ ซึ่งความสามารถที่ Java Servlet ให้กับ Server ก็คือ

1. ประมวลผลข้อมูล ที่มาพร้อมคำร้องขอ ในรูปแบบของ HTTPS ที่อยู่ใน HTML ซึ่งอาจจะเป็นการประมวลข้อมูลการซื้อขาย , การใช้บัตรเครดิต ซึ่งจากความสามารถนี้ อาจนำ Servlet มาใช้ในการประมวลผลข้อมูล ที่รับเข้ามาจากโคลเอนท์ได้
2. นำมาประยุกต์เพื่อช่วยในเรื่อง การทำงานร่วมกัน เนื่องจาก Servlet สามารถรองรับคำร้องขอของโคลเอนท์ได้หลายคำร้องขอพร้อมกัน และ servlet สามารถนำคำร้องขอที่มาจากแต่ละโคลเอนท์ มาประมวลผลได้พร้อมกัน
3. สามารถนำ servlet มาส่งต่อ คำร้องขอที่ได้รับเข้ามา ไปสู่ เซิร์ฟเวอร์ หรือ servlet อื่นๆอีกได้ ซึ่งความสามารถนี้ สามารถนำมาใช้ในการแบ่งงานแบบกระจาย เพื่อลดภาระของเซิร์ฟเวอร์แต่ละเครื่องได้
4. สามารถนำมาประยุกต์ใช้ในงานประเภท โปรแกรมตัวแทนได้ โดยโปรแกรมตัวแทน ซึ่งถูกสร้างจากแต่ละ servlet จะสามารถรับส่งข้อมูลถึงกันได้

#### 2.3.2 Servlet Life Cycles

Servlet จะมี 3 สถานะคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Init เมื่อ Servlet ถูกเรียกใช้เป็นการครั้งแรก จะทำการ Load ตัวเองเข้าสู่หน่วยความจำหลักของเซิร์ฟเวอร์ และจัดการกำหนดค่าเริ่มต้นบางอย่าง ตามแต่ที่จะมีกำหนดไว้ใน init( )
2. Service เมื่อทำการกำหนดค่าเริ่มต้นเรียบร้อยแล้ว ก็จะเข้าสู่สถานะการให้บริการ ซึ่งบริการต่างๆ ก็จะขึ้นอยู่กับโปรแกรม ที่ผู้เขียนกำหนดขึ้น
3. Destroy เมื่อผู้ควบคุมเซิร์ฟเวอร์ ต้องการยกเลิกการให้บริการ หรือต้องการเปลี่ยนแปลงการให้บริการของ Servlet นั้น ก็จะทำการทำลายทิ้ง โดยเรียก destroy( )

### 2.3.3 การเขียนโปรแกรม Servlet

เนื่องจาก Java Servlet เป็นการนำภาษา จาวา มาใช้เพิ่มความสามารถของเซิร์ฟเวอร์ จึงมีลักษณะการเขียนโปรแกรม เหมือนภาษาจาวาโดยทั่วไป ซึ่งส่วนสำคัญที่ต้องมีในทุกโปรแกรม servlet ก็คือ ส่วนควบคุมการติดต่อระหว่าง Servlet กับผู้ใช้ ( โคลเอนท์ ) ด้วยการนำ `javax.servlet.Servlet` และ `javax.servlet.http.HttpServlet` มาใช้งาน

วิธีที่ใช้ในการติดต่อกับโคลเอนท์ จะมีอยู่ 4 วิธี

โปรแกรมที่รองรับ Servlet จะต้องมีวิธีการรองรับ จัดการกับ method ที่สำคัญ 4 method คือ

- doGet สำหรับรองรับ คำร้องขอที่เป็นแบบ GET
- doPost สำหรับรองรับการร้องขอแบบ POST
- doPut สำหรับรองรับการร้องขอแบบ PUT
- doDelete สำหรับรองรับการร้องขอแบบ DELETE

โดยมา เราจะใช้ doPost เนื่องจากเป็นวิธีที่สะดวก และปลอดภัย

### 2.4 Web Server

เป็นตัวโปรแกรม ซึ่งทำหน้าที่รองรับคำร้องขอของโคลเอนท์ ซึ่งหน้าที่หลักแรกเริ่มของ Web Server ถูกออกแบบมา เพื่อให้จัดการกับคำร้องขอ ที่ต้องการผลลัพธ์ในรูปแบบของ HTML แต่ก็สามารถตอบสนองคำร้อง เป็นผลลัพธ์ในรูปแบบของ WML ได้เช่นกัน

ในการใช้งานกับ WAP , WEB Server จะต้องทำงานร่วมกับ WAP Gateway โดยเป็นเสมือนตัวคอยผลิตผลลัพธ์เริ่มแรก ซึ่งอยู่ในรูปแบบของ WML แต่มี Header เป็น HTTP แล้วส่งผลลัพธ์นั้น ไปให้ WAP Gateway ที่ได้อธิบายไปในหัวข้อที่ 1 เพื่อทำการส่งต่อผลลัพธ์ไปให้โคลเอนท์ในที่สุด

เนื่องจากโครงการนี้ มีการใช้ Java Servlet เป็นส่วนสำคัญของโครงการ Web Server ที่เลือกใช้ จึงจำเป็นต้องมีความสามารถในการรองรับ Java Servlet ซึ่งโครงการนี้ ได้เลือกใช้ Tomcat ซึ่งจะได้กล่าวอีกครั้ง ในบทที่ 5 หัวข้อที่ xxx

### 2.5 WAP Server

มีลักษณะ และหน้าที่การทำงาน คล้ายกับ WAP Gateway และ Web Server รวมกัน คือมีทั้งความสามารถในการติดต่อกับคำร้องขอ และส่งผลลัพธ์กลับไปให้โคลเอนท์ได้โดยตรง ซึ่งเป็นหน้าที่และความสามารถของ WAP Gateway และมีความสามารถในการจัดเตรียม WML Page ซึ่งอาจจะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ Static หรือ Dynamic ดังที่ได้อธิบายในหัวข้อที่ 3 ไว้แล้ว ซึ่งในส่วนนี้ เป็นหน้าที่ และความสามารถของ Web Server

ชุดโปรแกรม WAP Server ที่ถูกพัฒนาขึ้น และสามารถนำมาใช้งานได้ โดยไม่ต้องเสียค่าใช้จ่ายใดๆ มีเช่น

Nokia WAP Server

Ericsson WAP Server

Motolora WAP Server

## 2.6 ระบบฐานข้อมูล (Database System)

จากปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูล ได้ก่อให้เกิดการจัดเก็บข้อมูลในรูปแบบใหม่ขึ้น ที่เรียกว่าฐานข้อมูล (Database) การจัดเก็บข้อมูลในฐานข้อมูลนี้จะแตกต่างจากการจัดเก็บข้อมูลแบบแฟ้มข้อมูล เนื่องจากฐานข้อมูลเป็นการนำเอาข้อมูลต่าง ๆ ที่มีความสัมพันธ์กัน ซึ่งแต่เดิมจัดเก็บอยู่ในแต่ละแฟ้มข้อมูลมาจัดเก็บไว้ในที่เดียวกัน เช่น ข้อมูลพนักงาน สินค้าคงคลัง พนักงานขาย และลูกค้า ซึ่งแต่เดิมถูกจัดเก็บอยู่ในรูปของแฟ้มข้อมูลของฝ่ายต่าง ๆ ได้ถูกนำมาจัดเก็บรวมกันไว้ภายในฐานข้อมูลเดียว ซึ่งเป็นฐานข้อมูลรวมของบริษัท ส่งผลให้แต่ละฝ่ายสามารถใช้ข้อมูลร่วมกัน และสามารถแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูลได้

ข้อมูลต่าง ๆ ที่ถูกจัดเก็บเป็นฐานข้อมูล นอกจากจะต้องเป็นข้อมูลที่มีความสัมพันธ์กันแล้ว ยังจะต้องเป็นข้อมูลที่ใช้สนับสนุนการดำเนินงานอย่างน้อยอย่างใดอย่างหนึ่งขององค์กร ดังนั้นจึงอาจกล่าวได้ว่าแต่ละฐานข้อมูลจะเทียบเท่ากับระบบแฟ้มข้อมูล 1 ระบบ และจะเรียกฐานข้อมูลที่จัดทำขึ้นเพื่อสนับสนุนการดำเนินงานอย่างใดอย่างหนึ่งนั้นว่า “ระบบฐานข้อมูล” (Database System) เช่น ระบบฐานข้อมูลเงินเดือน ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่าง ๆ ที่สนับสนุนการคำนวณเงินเดือน หรือระบบฐานข้อมูลประชากร ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่าง ๆ ที่สนับสนุนการจัดทำสำมะโนประชากร ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่าง ๆ ที่สนับสนุนการจัดทำสำมะโนประชากร เป็นต้น

การติดต่อกับข้อมูลภายในฐานข้อมูลของผู้ใช้ จะต้องกระทำผ่านโปรแกรมที่มีชื่อว่าโปรแกรม Database Management System (DBMS) โดยหน้าที่หลักของ โปรแกรม DBMS ได้แก่ การทำให้การเรียกใช้ฐานข้อมูล เป็นอิสระจากส่วนของ Hardware หรือกล่าวอีกนัยหนึ่ง โปรแกรม DBMS จะมีหน้าที่ในการจัดการและควบคุมความถูกต้อง ความซ้ำซ้อนและความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูลแทนโปรแกรมเมอร์ ส่งผลให้ผู้ใช้สามารถที่จะเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยไม่ต้องทราบถึงโครงสร้างทางกายภาพของข้อมูลในระดับที่ลึกเช่นเดียวกับโปรแกรมเมอร์ เนื่องจากโปรแกรม DBMS นี้ จะมีส่วนของ Query Language ซึ่งเป็นภาษาที่ประกอบด้วยคำสั่งต่าง ๆ ที่ใช้ในการจัดการ และเรียกใช้ข้อมูลจากฐานข้อมูล ซึ่งสามารถนำไปใช้ร่วมกับภาษาคอมพิวเตอร์อื่น ๆ เพื่อพัฒนาเป็นโปรแกรมที่ใช้สำหรับเรียกใช้ข้อมูลจากฐานข้อมูลมาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.1 Database Management System (DBMS)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DLL หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปลเป็นการกระทำต่าง ๆ ภายใต้อำนาจคำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป

โปรแกรม DBMS นี้ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาทางด้าน Data Independence ที่ไม่มีในระบบแฟ้มข้อมูล ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และตัวข้อมูลภายในฐานข้อมูล กล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นอยู่กับรูปแบบ (Platform) ของตัว Hardware ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ส่งผลให้ผู้ใช้สามารถเรียกใช้ฐานข้อมูล โดยไม่จำเป็นต้องทราบถึงประเภทของข้อมูล หรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของ Field นั้น

ลำดับหน้าที่ของ DBMS มีดังนี้

1. ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
2. ทำหน้าที่ในการนำคำสั่งต่าง ๆ ซึ่งได้รับการแปลแล้ว ไปสั่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) การเพิ่มข้อมูล (Add) เป็นต้น
3. ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่ามีคำสั่งใดที่สามารถทำงานได้ และคำสั่งใดไม่สามารถทำงานได้
4. ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
5. ทำหน้าที่เก็บรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้มักจะถูกเรียกว่า “ข้อมูลของข้อมูล” (Metadata)
6. ทำหน้าที่ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

### 2.6.2 JDBC (Java Database Connectivity)

JDBC (Java Database Connectivity) ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystems ซึ่งก็คือฟังก์ชันมาตรฐาน หรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล นักพัฒนาสามารถใช้ JDBC API และยังสามารถประกอบด้วยแพ็คเกจอื่น ๆ ด้วย ซึ่งนำเสนอในรูปแบบฟังก์ชันพิเศษ โดยทั่วไปการใช้ SQL database ในการติดต่อกับเฟรมเวิร์ก

(framework) เพื่อพัฒนามาตรฐานในการติดต่อส่วนบนสุดของชนิดต่างๆ ของ database connectivity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

modules ซึ่งก็คือมาตรฐานของ ANSI SQL-2 Entry level database เพราะว่า relational database เกือบทั้งหมดในปัจจุบันใช้มาตรฐานของ SQL-2 Entry level

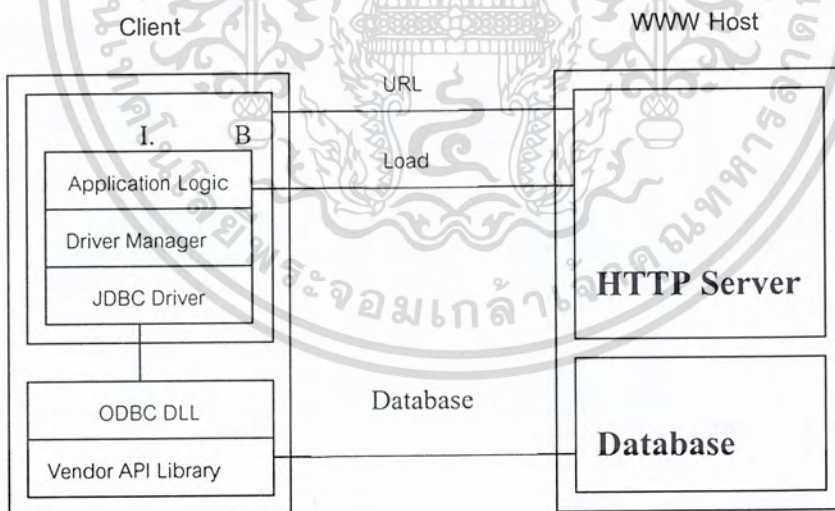
JDBC สร้างระดับการเชื่อมต่อเพื่อการสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC (Open Database Connectivity ของบริษัทไมโครซอฟท์) หลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบ X Window โครงสร้างของ JDBC

โครงสร้างของการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลัก คือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูป ระดับบนสุด JDBC API เป็นระดับฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับล่าง JDBC Driver (มีไดร์ฟเวอร์ที่ต่างกันอยู่ 4 ชนิด)

รายละเอียดการทำงานของไดร์ฟเวอร์แต่ละชนิดอธิบายได้ดังนี้

### 1. JDBC/ODBC bridge

JDBC/ODBC bridge ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูล ได้โดยผ่านทางการทำงานของ ODBC โดยนำข้อดีของ ODBC-enabled data sources ที่มีใช้อยู่โดยทั่วไปอย่างมาก ผังไคลเอนท์ จาวาแอปเพล็ตหรือจาวาแอปพลิเคชัน จะถูกเขียนโดยใช้ JDBC API บริดจ์จะทำการแปลงโดยการเรียกใช้ JDBC ไปยัง ODBC และส่งค่า ODBC Driver ที่เหมาะสมสำหรับ back-end database



รูปที่ 2.4 การติดต่อ JDBC โดยรูปแบบที่ JDBC-ODBC Bridge

ข้อดีของบริดจ์ ทำให้แอปพลิเคชันสามารถติดต่อกับฐานข้อมูลได้อย่างง่ายดาย โดยจากผู้ผลิตที่มีมากมาย โดยสามารถเลือก ODBC Driver ที่เหมาะสมอย่างไรก็ตามการติดต่อกับฐานข้อมูลประเภทนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

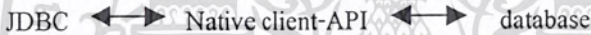
ต้องพิจารณาค่าใช้จ่าย (overhead) และความซับซ้อน (complexity) เพราะว่าการเรียกใช้จะมีลำดับดังนี้  
คือ



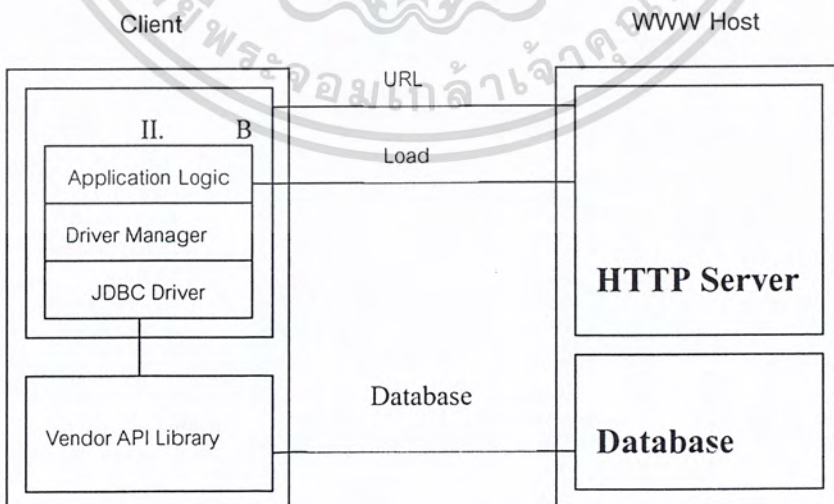
2. Native-API, partly Java driver

Native-API, partly Java driver ใช้ Vendor Library ในการแปลง JDBC function ไปยังคุณลักษณะของการใช้ภาษาไทยในการ Query เช่น ไลบรารีสำหรับออร์เคิล คือ ocilib การติดต่อ JDBC ประเภทนี้จะเหมือนกับ JDBC/ODBC Bridge ก็ต้องติดตั้งโค้ดในเครื่องไคลเอนต์

ข้อดีของไดรฟ์เวอร์ชนิดนี้คือ สามารถใช้ Native-API Driver ติดต่อกับฐานข้อมูลนั้นโดยตรง โดยผ่าน โพรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะกับการเชื่อมต่อนฐานข้อมูลแบบ Two-tier Client Server นอกจากนี้ยังเร็วกว่า ไดรฟ์เวอร์แบบแรก เพราะว่ามีเลขอร์พิเศษของการแปลงเป็น ODBC ถูกจำกัดออกไป



แต่มีบางส่วนของ Native-API Driver ถูกเขียนจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้ และไม่สามารถเชื่อมต่อนฐานข้อมูลข้ามชนิดกันได้



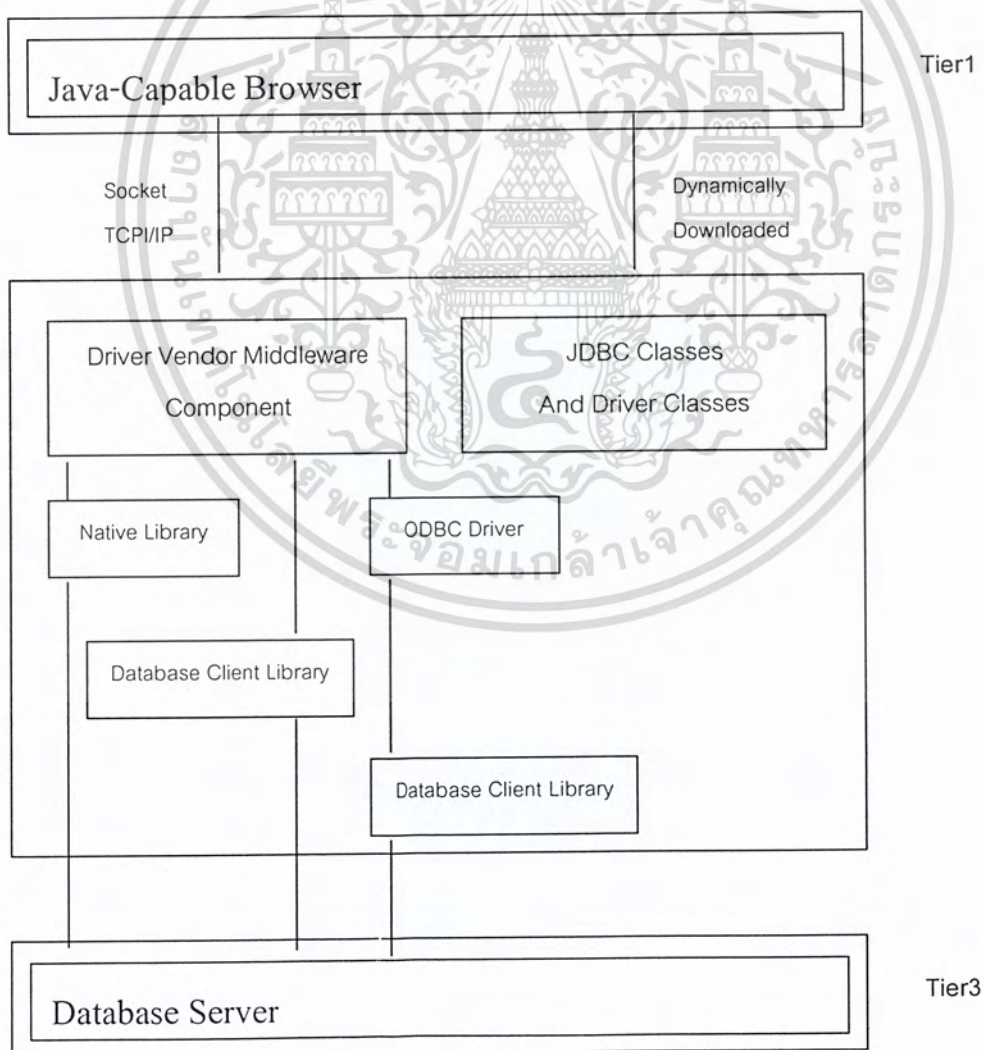
รูปที่ 2.5 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API, partly Java driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. Network-protocol, all-Java driver

Network-protocol, all-Java driver จะทำการแปลงการเรียกใช้ JDBC ให้อยู่ในรูปของเน็ตเวิร์ค โพรโตคอลร่วม (DBMS-independent network Protocol) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปเฉพาะของแต่ละฐานข้อมูล (database-specific API) บนเซิร์ฟเวอร์นั้นๆ รูปแบบการเชื่อมต่อจะเป็นลักษณะของ three tier ไดรฟ์เวอร์ชนิดนี้จะทำการเอ็กซิคิวท์บนไคลเอนต์ และส่งคำสั่ง SQL ไปยังเน็ตเวิร์ค เมื่อเซิร์ฟเวอร์ได้รับข้อมูล ก็จะจัดการเชื่อมต่อที่มีมาไปยังฐานข้อมูล

ไดรฟ์เวอร์ประเภทนี้เหมาะสมมากสำหรับ ระบบเครือข่ายอินเทอร์เน็ต-อินทราเน็ต และการทำงานที่มีผู้ใช้หลายคน เพราะมีความยืดหยุ่นคล่องตัวที่สุด เพราะเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไดรฟ์เวอร์ร่วมที่เขียนจากภาษาอื่น ซึ่งต้องติดตั้งเฉพาะฝั่งไคลเอนต์เท่านั้น และสามารถรันบนระบบใดก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine)



รูปที่ 2.6 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver

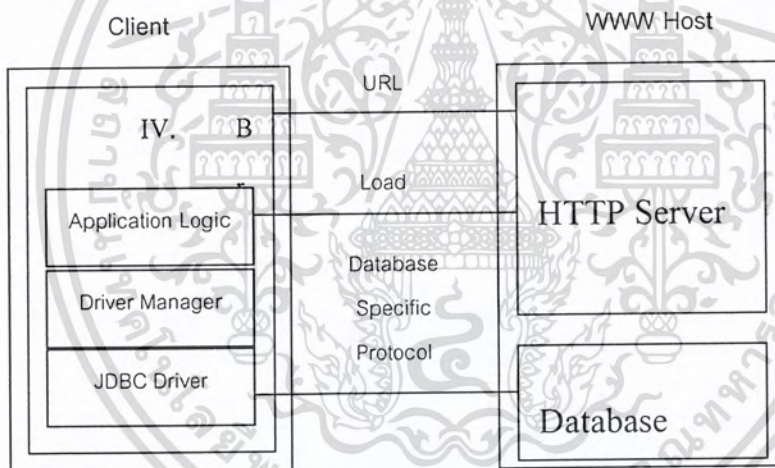
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. Native-protocol, all-Java driver

Native-protocol, all-Java driver จะแปลงการเรียกใช้คำสั่งของ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์คโพรโทคอลเฉพาะของฐานข้อมูลนั้นโดยตรง เพราะใช้ไคร์ฟเวอร์ของตัวเองแทนจำหน่ายฐานข้อมูลไคร์ฟเวอร์เหล่านี้สามารถเขียนในภาษาจาวา และติดต่อกับแอปพลิเคชันแบบ just-in-time เพราะว่าไคร์ฟเวอร์เหล่านี้จะแปลง JDBC ตรงไปยัง native protocol โดยปราศจากการใช้ ODBC หรือ Native APIs ซึ่งสามารถจัดหามาสำหรับการติดต่อกับฐานข้อมูลที่มีประสิทธิภาพสูง

ข้อดีของไคร์ฟเวอร์ชนิดนี้คือ ไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กรและไม่มีคามจำเป็นต้องติดตั้งไคร์ฟเวอร์ตัวกลาง

JDBC Driver ชนิดที่ 3 และ 4 คือไคร์ฟเวอร์ที่คาดว่าจะเป็สิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคต เพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและคล่องตัวมากกว่า



รูปที่ 2.7

การติดต่อ JDBC โดยรูปแบบที่ 4 Native-protocol, all-Java driver

ออราเกิลมี JDBC Driver 2 ประเภท คือ

1. JDBC Thin เป็น JDBC Driver ชนิดที่ 4 ซึ่งใช้ซ็อกเก็ต (socket) เชื่อมต่อโดยตรงกับออราเกิล โดยผ่านโพรโทคอล TCP/IP การติดต่อทำโดยภาษาจาวา ได้แก่ จาวาแอปพลิเคชันและจาวาแอปพลิเคชัน ทำให้ไคร์ฟเวอร์ชนิดนี้มีคุณสมบัติคือ ไม่ขึ้นกับแพลตฟอร์ม (platform-independent)
2. JDBC OCI เป็น JDBC Driver ชนิดที่ 2 ไคร์ฟเวอร์ชนิดนี้ใช้กับจาวาแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 POP 3 Mail Server ( rfc 1939 )

การรับจดหมายอิเล็กทรอนิกส์ ( Email ) แบบเก่า ใช้ข้อตกลงมาตรฐานที่เรียกว่า Simple Transfer Mail Protocol ( STMP ) ซึ่งในระหว่างที่ผู้ใช้อ่านเมล์ เครื่องคอมพิวเตอร์ของผู้ใช้ จำเป็นต้องติดต่อกับ Mail Server อยู่ตลอดเวลา ซึ่งทำให้สิ้นเปลืองทรัพยากร และเวลาในการรับส่งข้อมูล รวมทั้งทรัพยากร ในการรักษาการติดต่อระหว่างเครื่องคอมพิวเตอร์ของผู้ใช้ กับ Mail Server อีกด้วย จึงได้มีการคิดค้น วิธีการอ่านเมล์แบบอื่นขึ้น ซึ่งมีชื่อเรียกว่า Post Office Protocol ( POP ) ซึ่งในปัจจุบัน มี Version ที่ใช้กันอย่างแพร่หลายคือ Version 3 หรือ POP3 ซึ่งข้อกำหนดต่างๆ ได้ถูกกำหนดไว้ใน rfc 1939 กำหนดโดย xxxxA

### 2.7.1 หลักการทำงานของ POP3 คือ

1. โปรแกรมรับจดหมาย ที่ติดตั้งอยู่ในเครื่องคอมพิวเตอร์ของผู้ใช้ จะทำการสร้างการติดต่อกับ Mail Server ที่ให้บริการ POP3 ได้
2. เมื่อสร้างการติดต่อสำเร็จ โปรแกรมจะทำการส่งข้อมูลสำหรับการตรวจสอบผู้ใช้แต่ละคน
3. เมื่อทำการตรวจสอบผู้ใช้เรียบร้อยแล้ว โปรแกรมจะทำการอ่านจดหมาย ที่อยู่ใน Mail Server ทั้งหมด มาเก็บไว้ที่ตัวโปรแกรมรับจดหมาย แล้วลบจดหมาย ใน Mail Server นั้นไป แล้วตัดการติดต่อกับ Mail Server
4. โปรแกรมจะจัดการกับจดหมาย ที่อ่านมาเก็บไว้

จะเห็นได้ว่า ด้วยวิธีนี้ ทำให้การติดต่อกับ Mail Server ใช้เวลา และใช้ความสามารถของ Mail Server ไม่มาก ซึ่งจะมีผลคือทั้ง Mail Server และ ผู้ใช้ ที่ต้องการตรวจสอบจดหมาย คือประหยัดเวลาขึ้น Operation หลัก ที่ทำกับจดหมายคือ

- ตรวจสอบขนาดเนื้อที่ ที่ใช้จัดเก็บจดหมายแต่ละฉบับ
- เปิดอ่านจดหมายแต่ละฉบับ
- ลบจดหมาย

ในโครงการนี้ จะใช้การติดต่อกับ POP3 Mail Server แบบ Socket ซึ่งจะได้อธิบายในหัวข้อถัดไป การทดลองติดต่อกับ POP3 Mail Server อาจทำได้โดยใช้โปรแกรม Telnet ติดต่อไปยัง Host Address ของ POP3 Mail Server ที่พอร์ตที่ให้บริการ ซึ่งโดยมากมักนิยมใช้พอร์ตที่ 110

เมื่อติดต่อกับ POP3 Mail Server จะมีลักษณะดังนี้

```
+OK POP3 Chaokhun.kmitl.ac.th v7.64 server ready
```

```
user a0010579
```

```
+OK User name accepted, password please
```

```
pass a123456
```

```
+OK Mailbox open, 1 messages
```

```
stat
```

```
+OK 1:3564
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

list l

+OK l 3564

retr

-ERR Missing message number argument

quit

+OK Sayonara

ผลตอบสนอง จาก Mail Server จะมี 2 ประเภทคือ

1. คำร้องขอจากผู้ใ้สามารถตอบสนองได้ จะมี +OK ขึ้นต้น แล้วตามด้วยผลลัพธ์ตามคำสั่ง
2. คำร้องขอจากผู้ใ้ ไม่สามารถตอบสนองได้ จะมี -ERR ขึ้นต้น ซึ่งการเกิดข้อผิดพลาดนี้ อาจเกิด ขึ้นได้หลายกรณี เช่น มีการใส่ค่าพารามิเตอร์ไม่ถูกต้อง

response ต่างๆ ที่ส่งมาจาก Mail Server จะมีรหัสลงท้ายเป็น "\r\n" หรือ "\r\n.\r\n" เสมอ

### 2.7.2 คำสั่งมาตรฐาน สำหรับใช้ติดต่อกับ POP3 Mail Server มีดังต่อไปนี้

#### 1. STAT

เป็นคำสั่ง ที่เรียกตรวจสอบจำนวนจดหมาย ที่มีอยู่ในตู้จดหมาย รวมทั้งขนาดเนื้อที่ ที่ใช้จัดเก็บ จดหมาย

#### 2. LIST

เป็นคำสั่ง ให้ Mail Server แสดงหมายเลขของจดหมายที่อยู่ในตู้จดหมาย พร้อมทั้งขนาดเนื้อที่ ที่ใช้จัดเก็บจดหมาย โดยถ้ามีการระบุหมายเลขของจดหมาย ก็จะแสดงเฉพาะ หมายเลข และ ขนาดเนื้อที่ที่ใช้จัดเก็บจดหมายฉบับนั้น แต่ถ้าไม่ระบุหมายเลขของจดหมาย ก็จะแสดงหมายเลขของจดหมายแต่ละฉบับ พร้อมทั้งขนาดเนื้อที่ที่ใช้จัดเก็บจดหมายแต่ละฉบับ

#### 3. RETR

เป็นคำสั่งเรียกดูข้อความเนื้อหา ภายในจดหมายฉบับนั้นๆ คำสั่งนี้ จึงต้องมีการระบุหมายเลข ของจดหมายที่ต้องการอ่านข้อความเสมอ การใช้คำสั่งนี้ ควรจะระวังว่า จะมีการแนบไฟล์มา พร้อมกับจดหมายฉบับนั้นด้วยหรือไม่ เพราะคำสั่ง RETR นี้ จะเปิดอ่านและแสดงเนื้อหาข้าง ในทั้งหมด โดยไม่ตรวจสอบว่า มีไฟล์แนบมาด้วยหรือไม่ ซึ่งถ้ามีไฟล์แนบมาด้วย ข้อความที่ แสดงออกมา ก็จะอยู่ในรูปของรหัสแอสกีจะเห็นว่า ในส่วนของ Status: จะเป็นส่วนที่บอก สถานะว่า จดหมายฉบับนี้ เคยถูกเปิดอ เนไปหรือไม่ ซึ่งจดหมายฉบับนี้ยังไม่ถูกเปิดอ่าน ถ้าถูก เปิดอ่านแล้ว จะมีผลลัพธ์เป็น Status: RO

#### 4. DELE

เป็นคำสั่งลบจดหมาย ซึ่งต้องมีการระบุหมายเลขของจดหมาย ที่ต้องการลบ เมื่อมีการใช้คำสั่ง นี้ Mail Server จะทำการทำเครื่องหมายไว้ที่จดหมายฉบับนั้น แต่ยังไม่ทำการลบจดหมายไป ในทันที การลบจดหมายจะเกิดขึ้น เมื่อมีการใช้คำสั่ง UPDATE หรือ QUIT ซึ่งจะได้อธิบาย หัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. NOOP

เป็นคำสั่งยืนยันการติดต่อกับ Mail Server เพื่อป้องกันการถูกยกเลิกการติดต่อจาก Mail Server เนื่องจาก Time Out

## 6. RSET

เป็นคำสั่ง ยกเลิกการลบจดหมาย ซึ่งเกิดจากการใช้คำสั่ง DELE แต่จะใช้ได้ ถ้ายังไม่มีเรียกใช้คำสั่ง UPDATE

## 7. QUIT

คำสั่ง ยกเลิกการติดต่อกับ Mail Server

## 8. UPDATE State

เมื่อมีการเรียกคำสั่ง QUIT ผู้จดหมายนั้น จะเข้าสู่สถานะ UPDATE ก็จะทำการประมวลผล และจัดการกับจดหมายต่างๆ ตามคำสั่ง ที่ถูกเรียกใช้ก่อนมีการเรียกใช้คำสั่ง UPDATE

## 2.8 Socket Programming

การรับส่งข้อมูลบนระบบเครือข่าย ซึ่งรวมถึงอินเทอร์เน็ต สามารถทำได้หลายวิธี วิธีหนึ่ง ที่ง่าย ไม่ซับซ้อน และสะดวกรวดเร็วคือ การติดต่อรับส่งข้อมูลแบบ Socket ซึ่งจะทำให้สามารถส่งชุดข้อมูล (Datagram) ด้วย Socket ทำได้ 2 วิธี

### 2.8.1 Internet Protocol (IP)

ในโครงงานนี้ จะใช้ Socket แบบ Internet Protocol (IP) ซึ่งจะรับ และส่งข้อมูลแบบ “IP Datagram” ซึ่งมีหลักการทำงาน ในการติดต่อบนระบบเครือข่าย คือ

- ทำการสร้างการติดต่อ ระหว่างเครื่องเทอร์มินอล 2 เครื่อง หรือมากกว่า แล้วแต่กรณี โดยอาจใช้การติดต่อแบบ โคลเอนท์ – เซิร์ฟเวอร์ คือมีเครื่องเทอร์มินอลหนึ่ง ทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ โดยทำการรอการติดต่อจากเครื่องเทอร์มินอลอีกเครื่อง ซึ่งในที่นี้หมายถึงโคลเอนท์ โดยจะรอตรวจสอบการติดต่อเข้ามา ที่พอร์ตที่กำหนด
- เมื่อเครื่องโคลเอนท์ติดต่อเข้ามาหาเซิร์ฟเวอร์ ที่พอร์ตที่เซิร์ฟเวอร์อยู่ ก็จะเกิดการติดต่อแบบถาวร ( เป็นแบบ Virtual Circuit ) แล้วจึงทำการรับส่งข้อมูลกัน
- เมื่อต้องการยกเลิกการติดต่อ ก็จะทำการตัดการติดต่อกัน Virtual Circuit ที่สร้างขึ้นมา ก็จะถูกทำลายไปด้วย

จากรูปแบบการทำงานที่ได้อธิบายข้างต้น จะเห็นว่า ความแน่นอนของระบบจะมีสูง เนื่องจากมีการสร้างการติดต่อแบบ Virtual Circuit การติดต่อ Socket แบบ Internet Protocol จึงได้รับความนิยมในการใช้งานสูง เนื่องจากมีความถูกต้อง น่าเชื่อถือสูง แต่ก็มีข้อเสียตรงที่ ต้องมีการสร้างการเชื่อมต่อแบบ Virtual Circuit ทุกครั้ง ที่มีการติดต่อระหว่าง โคลเอนท์ และ เซิร์ฟเวอร์ ซึ่งจะเป็นการเสียพอร์ตการติดต่อของทั้งโคลเอนท์และเซิร์ฟเวอร์ไป

การเขียนโปรแกรมควบคุมการสื่อสารรับส่งข้อมูลบน Socket แบบ IP โดยใช้ภาษาจาวา ทำโดยสร้างออปเจกต์สำหรับติดต่อ ซึ่งออปเจกต์นี้ เป็นส่วนขยายมาจาก Socket Class มีวิธีการดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ติดต่อเข้าเซิร์ฟเวอร์ ทำได้โดยการกำหนดให้ connection เป็น ออบเจกต์ ที่เป็นตัวติดต่อกับ เซิร์ฟเวอร์ โดยที่อยู่ หรือชื่อของ เซิร์ฟเวอร์ในที่นี้คือ MailServer และติดต่อกับ เซิร์ฟเวอร์ที่พอร์ตที่อยู่ในตัวแปร port

```
connection = new Socket(MailServer,port);
```

- เมื่อติดต่อเข้าเซิร์ฟเวอร์ได้ ก็ทำการเตรียม ออบเจกต์ สำหรับรับส่งข้อมูลซึ่งทำได้โดยสร้างออบเจกต์ ที่มาจาก DataInputStream Class และ DataOutputStream Class

```
inStream = new DataInputStream(connection.getInputStream());
```

```
outStream = new DataOutputStream(connection.getOutputStream());
```

- ส่วนที่ควบคุมการรับส่งข้อมูล ซึ่งแบ่งได้เป็น 2 ส่วนคือ

- ส่วนรับข้อมูล

```
ServerResponse = inStream.readLine();
```

- ส่วนส่งข้อมูล

```
outStream.writeBytes(InputCommand+"\r\n");
```

- เมื่อต้องการยกเลิกการติดต่อ ใช้คำสั่ง

```
connection.close();
```

### 2.8.2 User Datagram Protocol ( UDP )

เป็นการติดต่อระหว่างเครื่องคอมพิวเตอร์เทอร์มินอล 2 เครื่องหรือมากกว่า โดยไม่มีการสร้างการเชื่อมต่อแบบ Virtual Circuit แต่จะใช้การส่งที่เรียกว่าแบบ Datagram ขึ้นมาแทน ซึ่งมีข้อดีที่ไม่ต้องเสียช่องการสื่อสาร ในการสร้าง Virtual Circuit มากนัก แต่ก็มีข้อเสียที่ ข้อมูลที่รับส่ง อาจมีการสูญหาย หรืออาจมีความเสียหายได้ วิธีการติดต่อแบบนี้ จึงไม่ค่อยได้รับความนิยมมากนัก เราจึงไม่ขอกล่าวถึงรายละเอียด

## บทที่ 3

### การดำเนินงาน

ในโครงการนี้ จะประกอบไปด้วย โปรแกรมประยุกต์สำหรับใช้งานในทางธุรกิจคือ

1. ระบบการทำรายการสินค้า
2. Check Mail

#### 3.1 ระบบการทำรายการสินค้า

##### 3.1.1 การวิเคราะห์และการออกแบบระบบการทำรายการสินค้า

###### 3.1.1.1 หลักการ

เนื่องจาก WAP ใช้งานได้กับอุปกรณ์ ไร้สาย ( Wireless Device ) โดยได้พัฒนาขึ้นมาจากมาตรฐาน Internet เหมือนการให้บริการผ่านเว็บทั่วไปคือมี Web Server คอยให้บริการ ดังนั้นเราจึงใช้ประโยชน์จากจุดนี้มาเพื่อช่วยในการทำรายการสินค้าแบบออนไลน์ เพื่อความสะดวกและรวดเร็ว ให้ก่อเกิดประโยชน์ในเชิงธุรกิจให้มากที่สุด

###### 3.1.1.2 ขอบเขตการให้บริการ

ระบบการทำรายการสินค้าผ่าน WAP นั้นจะให้ก่อนจะใช้บริการได้นั้นต้องทำการลงทะเบียนคลังสินค้าผ่านทางเว็บ โดยกำหนดจำนวนและประเภทสินค้าขึ้น และยังต้องกำหนดรูปแบบการทำคลังของสินค้าเพื่ออำนวยความสะดวกกับการทำรายการในภายหลัง ส่วนผู้ที่ทำการบันทึกรายการได้นั้นต้องทำการลงทะเบียนและขออนุญาตจาก Admin หรือ ผู้ที่ทำการสมัครในครั้งแรกนั้นก่อน จึงจะสามารถทำรายการได้ หลังจากนั้น ผู้ที่เป็น Admin สามารถจะดูรายละเอียดการเปลี่ยนแปลงคลังสินค้าได้โดยผ่านทางเว็บ นอกจากนี้ยังมีบริการรับส่งข้อความระหว่าง ผู้ใช้บริการด้วยกันด้วย

###### 3.1.1.3 กิจกรรมต่างๆ

กิจกรรมต่าง ๆ ของระบบการทำรายการสินค้านี้มีดังต่อไปนี้

1. การลงทะเบียนคลังสินค้า
2. การลงทะเบียนผู้ทำรายการ
3. การให้อินพุตทำรายการ
4. การนำสินค้าเข้า
5. การนำสินค้าออก
6. การดูสินค้าคงเหลือ
7. การแจ้งเตือนรูปแบบการนำสินค้าออก
8. การแจ้งเตือนสินค้าหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 3.1.1.4 การลงทะเบียนคลังสินค้า

ผู้ที่มีความประสงค์ ในการใช้ระบบต้องการลงทะเบียนเสียก่อน โดยผู้ลงทะเบียนจะถือเป็น Admin ของบริษัทนั้น หลังจากกรอกข้อมูลส่วนตัวแล้ว ต้องกรอกรายละเอียดสินค้าที่มีในคลังไม่ว่าจะเป็นการแบ่งประเภทของสินค้า และประเภทของสินค้าที่มี เพื่อนำไปแยกเป็นตารางของสินค้าประเภทต่าง ๆ ได้อย่างถูกต้อง และท้ายที่สุดซึ่งเป็นส่วนสำคัญก็คือการระบุ column ต่าง ๆ ที่จำเป็นในการทำการขายสินค้า ซึ่งขึ้นตามแต่ลักษณะการออกแบบลักษณะการบันทึกของบริษัทนั้น ๆ

#### 3.1.1.5 การให้อินทิติทำการขาย

เมื่อ Admin ของบริษัทได้รับข้อความขออนุญาตทำการขายโดยผ่านทางระบบรับส่งข้อความแล้ว จะทำการพิจารณา แล้วจึงให้อินทิติทำการขายสินค้าได้ ทั้งนี้ Admin ของบริษัทมีสิทธิที่จะทำการยกเลิกการให้สิทธิทำการขายสินค้า (disable) หรือให้สิทธิ (enable) เมื่อไรก็ได้

#### 3.1.1.6 การนำสินค้าเข้า

ผู้ที่ทำการบันทึกการขายสินค้าจะทำการบันทึกการขายสินค้าโดยต้องใส่ค่าต่าง ๆ ที่กำหนดไว้ให้ครบ โดยเมื่อทำการนำสินค้าเข้าเรียบร้อยแล้วข้อมูลต่าง ๆ จะถูกเก็บอยู่ในฐานข้อมูลตามตารางของประเภทสินค้านั้น ๆ ซึ่งจะถูกรับที่วันเวลาจำนวนสินค้าคงเหลือ ลำดับการนำเข้า และผู้ทำการขายสินค้าลงไปด้วย

#### 3.1.1.7 การนำสินค้าออก

การนำสินค้าออกจะถูกบันทึกเช่นเดียวกันกับการนำสินค้าเข้า แต่การนำสินค้าออกจะมีข้อกำหนดต่าง ๆ มากกว่าโดยจะมีการแจ้งเตือนในกรณีทีต่าง ๆ กัน เมื่อเกิดเหตุการณ์ตรงตามเงื่อนไขที่กำหนดไว้

#### 3.1.1.8 การดูสินค้าคงเหลือ

เป็นฟังก์ชันเสริมที่ช่วยสำหรับการนำสินค้าออก โดยจะเป็นการตรวจสอบดูสินค้าประเภทนั้น ๆ ที่ยังคงเหลืออยู่ในคลัง โดยเรียงลำดับตามการนำเข้าของสินค้า

#### 3.1.1.9 การแจ้งเตือนรูปแบบการนำสินค้าออก

ในขั้นตอนลงทะเบียนคลังสินค้าในตอนแรก สินค้าแต่ละประเภทจะถูกกำหนดรูปแบบการนำสินค้าออกว่าเป็นแบบ FIFO (FIRST-IN-FIRST-OUT) หรือแบบ LIFO (LAST-IN-FIRST-OUT) หรือแบบที่ไม่พิจารณา (NONE) ซึ่งในขั้นตอนการนำสินค้าออก ถ้ามีการนำสินค้าออกผิดรูปแบบ ก็จะมีการแจ้งเตือนให้ผู้ทำการขายได้ทราบ หลังจากนั้นผู้ที่ทำการขายจะสามารถทำการพิจารณาว่าควรจะนำสินค้านั้นออกหรือไม่อย่างไร โดยลำดับการนำสินค้าเข้านั้นจะดูได้จาก Rank ซึ่งจะเป็นตัวบ่งบอกว่าสินค้าชุดใดถูกนำเข้ามาก่อนหลังชุดใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณี FIFO ถ้า หมายเลข Rank ของสินค้าชุดที่นำออกไม่มีค่าน้อยที่สุดเมื่อเทียบกับสินค้าชุดอื่น ที่เหลือ จะต้องทำการแจ้งเตือน

กรณี LIFO ถ้า หมายเลข Rank ของสินค้าชุดที่นำออกไม่มีค่ามากที่สุดเมื่อเทียบกับสินค้าชุดอื่น ที่เหลือ จะต้องทำการแจ้งเตือน

กรณี NONE จะไม่มีการแจ้งเตือนใด ๆ

### 3.1.1.10 การแจ้งเตือนสินค้าหมด

ในขั้นตอนลงทะเบียนคลังสินค้าในตอนแรก สินค้าแต่ละประเภทจะถูกกำหนดค่าสินค้าคงเหลือที่จะต้องมีการแจ้งเตือน (Alert) คือทุกครั้งที่มีการนำสินค้าออก ก็จะทำการเปรียบเทียบค่าสินค้าคงเหลือทั้งหมดของสินค้าประเภทนั้น ๆ (Total) โดยถ้ามีค่าน้อยกว่าหรือเท่ากับค่า Alert ที่ได้ตั้งไว้ ก็จะมีการแจ้งเตือน เพื่อให้มีการนำเข้ามาใหม่

### 3.1.2 การทำรายการสินค้า

ลำดับขั้นตอนในการทำรายการสินค้านี้

1. กรอกรายละเอียดของบริษัท และ Admin ของบริษัท
2. กรอกรายละเอียดต่าง ๆ เกี่ยวกับตัวสินค้า ตามขั้นตอน
3. ผู้ที่จะเช็คสินค้าทำการกรอกรายละเอียดส่วนตัว
4. ผู้ที่จะเช็คสินค้า ส่งข้อความเพื่อขออนุญาต จาก Admin ของ บริษัท
5. Admin ของบริษัท ทำการ Enable Account ของผู้เช็คสินค้า
6. ผู้ที่จะเช็คสินค้าสามารถทำรายการต่าง ๆ ได้โดย login โดยใช้ Username แล้ว Password ของตนเอง

#### 3.1.2.1 การกรอกรายละเอียดของบริษัท และ Admin ของบริษัท

ในขั้นตอนแรกสุดของการทำรายการสินค้านั้น Admin ที่ทำหน้าที่คุมระบบเช็ครายการสินค้านั้นจะต้องทำการกรอกรายละเอียดข้อมูลต่าง ๆ ของบริษัท และตนเอง ผ่านทางเว็บ ดังรูปที่ xxx

ซึ่งข้อมูลที่ต้องกรอกในหน้าจอนี้ ได้แก่ ชื่อบริษัท Username Password และชื่อของ Admin สถานที่ตั้งของบริษัท เบอร์ติดต่อ และ อีเมล จำนวนชนิดของสินค้า และสุดท้าย คือ คำชี้แจงเพิ่มเติมอื่น ๆ

Page Signin - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost:8080/serviet/Page12n

Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft RealPlayer Plus Window

COMPANY: KMITL

ADMIN USERNAME: S0011001

ADMIN PASSWORD: knot

FIRSTNAME: ARNUPAP

LASTNAME: PAKORNYUDH

ADDRESS: 123....

TELEPHONE: 4242424

E-MAIL: s0011001@kmitl.ac.th

NUMBER OF GOODS: 4

COMMENT:

Done Local intranet zone

## รูปที่ 3.1

หลังจากที่คลิกปุ่ม "SUBMIT" ในหน้าจอข้างต้นแล้ว ข้อมูลทั้งหมดจะถูกนำไปเก็บในตารางชื่อ Admin และ ตาราง Username\_Password ดังรูป

Company	Username	Password	Firstname	Lastname	Address	Telephone	Email	Goods	Comment
KMITL	S0011001	knot	ARNUPAP	PAKORNYUDH	123.....	4242424	s0011001@ce.kmitl.ac.th	4	-

ตารางที่ 3.1 ตาราง Admin

Username	Password	Company	Status	Firstname	Lastname
S0011001	knot	KMITL	Admin	ARNUPAP	PAKORNYUDH
S0010579	au	KMITL	Authorize	Montri	Rattana

ตารางที่ 3.2 ตาราง Username\_Password

ตาราง Admin จะเป็นตารางที่ใช้เก็บรายละเอียดเบื้องต้นของ Admin แต่ละบริษัท รวมไปถึงรายละเอียดของ บริษัทด้วย ส่วนตาราง Username\_Password จะเป็นตารางที่ใช้เก็บข้อมูลเกี่ยวกับ User เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละคนซึ่งได้แก่ Admin ของแต่ละบริษัท รวมไปถึงผู้ทำการเช็คสินค้าแต่ละบริษัทด้วย ข้อมูลในตารางนี้ได้แก่ Username Password ชื่อบริษัท สถานะ และ ชื่อนามสกุล

และ สุดท้ายโปรแกรมจะทำการสร้างตารางขึ้นอีกตารางหนึ่งตามชื่อ Username ของ Admin ที่เราได้กรอกไว้ เป็นตารางชื่อ Username\_Message คือ ถ้า Username คือ S0011001 ตารางนี้ก็จะชื่อว่า S0011001\_Message ซึ่งจะไว้ใช้สำหรับจัดการเรื่อง Message ต่าง ๆ ซึ่งจะกล่าวในภายหลัง

MessageDateTime	Sender	Read	New	Subject	Message
-----------------	--------	------	-----	---------	---------

ตารางที่ 3.3 ตาราง S0011001\_Message

### 3.1.2.2 กรอกรายละเอียดต่าง ๆ เกี่ยวกับตัวสินค้า ตามขั้นตอน

สำหรับขั้นตอนนี้จะค่อนข้างมีความยุ่งยากสักเล็กน้อย ในหน้าจอแรกจะให้ผู้ใส่ กรอกรายละเอียดเบื้องต้นของสินค้า ซึ่งจำนวนสินค้าจะตามที่ได้ระบุไว้ในหน้าจอที่แล้ว ในช่องต่าง ๆ ที่ให้กรอกจะมีความหมายดังต่อไปนี้

1. GOODSNAME – ชื่อของสินค้า

2. NO OF ATTRIBUTE – คือจำนวนสิ่งที่ใช้แยกความแตกต่างหรือชนิดย่อยของสินค้า เช่น ถ้าสินค้า เรามีหลายสี หลายขนาด หลายยี่ห้อ เราอาจจะให้มีค่าเท่ากับ 3 ซึ่งหมายถึง Color Size Brand

3. SORT – เป็นระบบช่วยสำหรับการนำสินค้าออก ซึ่ง จะมีให้เลือก 3 แบบ คือ

- FIFO (FIRST – IN – FIRST – OUT) หมายถึง คอยเตือนให้สินค้าชุดที่ถูกนำเข้ามาก่อนควรจะต้องถูกนำออกไปก่อน

- LIFO (LAST – IN – FIRST – OUT) หมายถึง คอยเตือนให้สินค้าชุดที่ถูกนำเข้ามาทีหลังควรจะต้องถูกนำออกไปก่อน

- NONE หมายถึง จะไม่มีการเตือนใด ๆ

4. ALERT – เป็นระบบช่วยอีกอย่าง ซึ่งจะคอยเตือนเมื่อนำสินค้าออกแล้วระดับสินค้าที่คงเหลืออยู่ในคลังเหลือถึงระดับที่กำหนด จะมีการเตือนเพื่อให้ทำการสั่งสินค้าเข้าเพิ่มอีก

KNOT PROJECT - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost:8080/Servlet/Page2n

Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft RealPlayer Plus Window

GOODSNAME1: Pen  
NUMBER OF ATTRIBUTE: 2 SORT: FIFO ALERT WHEN: 4000

GOODSNAME2: Pencil  
NUMBER OF ATTRIBUTE: 3 SORT: LIFO ALERT WHEN: 3000

GOODSNAME3: Rubber  
NUMBER OF ATTRIBUTE: 2 SORT: NONE ALERT WHEN: 3500

GOODSNAME4: Ruler  
NUMBER OF ATTRIBUTE: 2 SORT: FIFO ALERT WHEN: 4000

CHECK YOUR DATA BEFORE CLICK SUBMIT

Done Local intranet zone

### รูปที่ 3.2 หน้าจอการกรอกรายละเอียดเบื้องต้นของสินค้า

หลังจากที่กดปุ่ม “SUBMIT” ในหน้านี้แล้ว ข้อมูลต่างๆ ที่ได้ทำการใส่เข้าไปจะถูกนำไปสร้างเป็นตาราง ตามชื่อบริษัท ซึ่งมีทั้งหมด 4 คอลัมน์ ดังตาราง xxx

Goods	NoAttribute	Sort	Alert
Pen	2	FIFO	4000
Pencil	3	LIFO	3000
Rubber	2	NONE	3500
Ruler	2	FIFO	4000

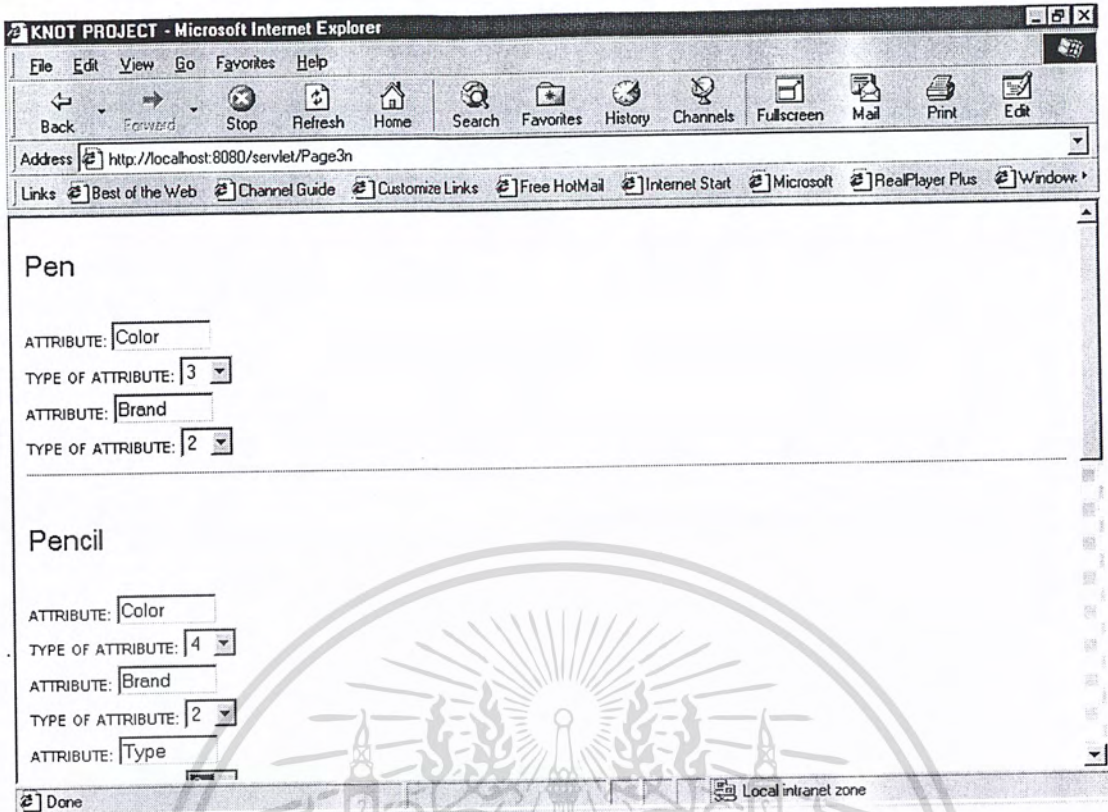
ตารางที่ 3.4 ตารางตัวอย่าง KMITL

หน้าถัดมาจะเป็นการกรอกรายละเอียดของแต่ละ Attribute ซึ่งจะมีอยู่ 2 ช่องให้กรอกคือ

1. ATTRIBUTE – หมายถึง ชื่อของ Attribute นั้น ๆ

2. TYPE OF ATTRIBUTE – หมายถึง ประเภทย่อยของ ATTRIBUTE เหล่านั้น เช่น ถ้าเราให้ ATTRIBUTE เป็น Color และ TYPE OF ATTRIBUTE เป็น 3 ก็อาจจะหมายถึง มีสีอยู่ 3 สี คือ Red Blue Green

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 หน้าจอการกรอกรายละเอียดของแต่ละ Attribute

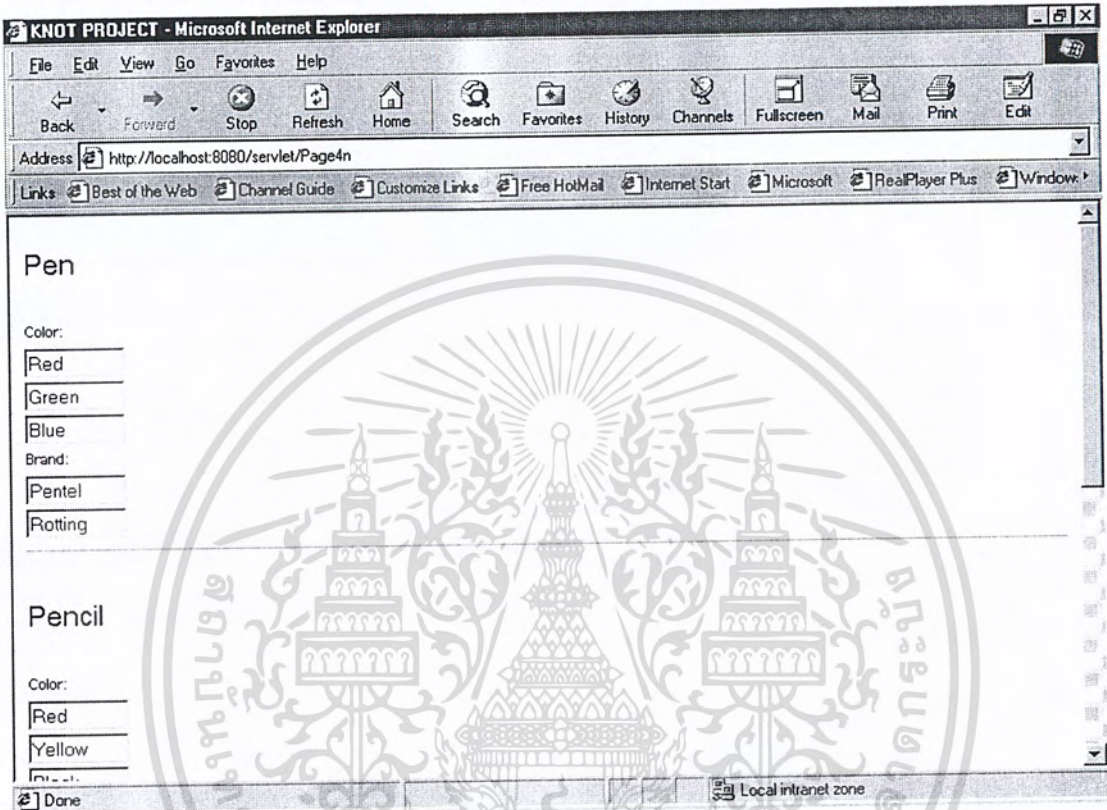
หลังจากที่กดปุ่ม “SUBMIT” ข้อมูลต่างๆ จะถูกนำไปสร้างเป็นตารางตามชื่อของบริษัท และชนิดของสินค้า ซึ่งตารางนี้จะเก็บ ชื่อ Attribute และ จำนวนชนิดของ Attribute ตามที่ได้ระบุไว้

Attribute	NoKind
Color	3
Brand	2

ตารางที่ 3.5 ตารางตัวอย่าง KMITL\_Pen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าถัดมาจะเป็นหน้าจอสำหรับใส่ข้อมูลเกี่ยวกับ Attribute แต่ละตัวว่ามีประเภทย่อยคืออะไรบ้าง เช่นถ้า ชื่อ Attribute ย่อยที่ได้ระบุไว้ในหน้าที่ แล้ว เป็น Brand ซึ่งหมายถึงยี่ห้อ ของสินค้า สำหรับหน้านี้เราก็จะใส่ว่าสินค้านั้นมี Brand อะไรบ้าง



รูปที่ 3.4 หน้าจอกรกรอกชื่อ Attribute

สำหรับข้อมูลในหน้านี้จะถูกนำไปสร้างเป็นตารางตามชื่อของบริษัท ชื่อสินค้า และชื่อ Attribute ซึ่งในตารางจะระบุถึงประเภทของ Attribute ว่ามีอะไรบ้าง

Kind
Pentel
Rotting

ตารางที่ 3.6 ตารางตัวอย่าง KMITL\_Pen\_Brand

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหน้าจอดีไปจะเป็นหน้าจอสุดท้ายในขั้นตอนกรอกรายละเอียดของสินค้า ซึ่งในแต่ละ Field ของ หน้าจอ นี้จะถูกนำไปสร้างเป็นคอลัมน์ต่าง ๆ ของตารางแสดงรายละเอียดการเปลี่ยนแปลง หมุนเวียนของ สินค้าแต่ละชนิด ซึ่งคอลัมน์พื้นฐานของตารางนี้ได้แก่

TransactionDateTime – ใช้เก็บวันและเวลาที่ทำการรายการสินค้าตัวนี้

InOut - เป็นคอลัมน์ที่ระบุจำนวนสินค้าที่ถูกนำเข้า หรือนำออก

Available – จะบ่งบอกว่าสินค้าชุดนั้น ๆ เหลืออยู่อีกเท่าไร

Total – จะบ่งบอกว่าสินค้าประเภทนั้น ๆ เหลืออยู่ในคลังอีกทั้งหมดเท่าไร

Rank – ลำดับการเข้ามาของสินค้า

Stocker – ชื่อผู้นำเข้า/นำออก สินค้า

ซึ่ง field ที่จะให้กรอกสำหรับหน้าจอ นี้จะมีอยู่ 2 ประเภทคือ

1. SPECIFIC SIGN – ซึ่งจะกลายเป็นคอลัมน์ที่เป็นตัวบ่งชี้ความแตกต่างของสินค้าแต่ละชุดว่าเป็นชุดไหน เช่น อาจจะเป็น Serial number หรือ Part number ของสินค้า

2. DYNAMIC VALUE – จะเป็นคอลัมน์ซึ่งสำหรับระบุค่าต่าง ๆ ที่จำเป็นสำหรับการทำบัญชีอื่น ๆ ตามแต่รูปแบบของแต่ละบริษัท เช่น ในบางบริษัท ในการนำสินค้าเข้าออกจะต้องมีการระบุถึงราคาของสินค้าที่ทำการนำเข้าหรือนำออกด้วย ดังนั้น ใน field นี้เราอาจจะให้เป็น PriceIn และ PriceOut ซึ่งหมายถึงราคานำเข้า/นำ ออกสินค้า ซึ่งผู้ที่ทำการเช็คสินค้าจะต้องทำการระบุลงไปด้วย

Input Type - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost:8080/servlet/Page5n

Links Best of the Web Channel Guide Customize Links Free HotMail Internet Start Microsoft RealPlayer Plus Window

SPECIFIC SIGN:

PartNo

Serial

Location

DYNAMIC VALUE:

PriceIn

PriceOut

CHECK YOUR DATA BEFORE CLICK SUBMIT

SUBMIT

Done Input Type - Microsoft Internet Explorer

รูปที่ 3.5 รูปการใส่ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

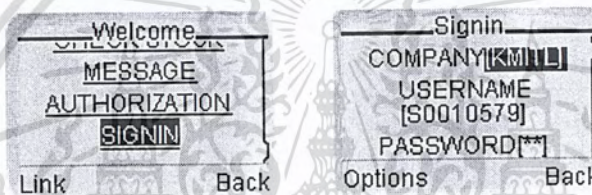
ข้อมูลต่าง ๆ ที่ได้ส่งไปจะถูกนำไปสร้างเป็นตารางตามชื่อบริษัท ชื่อสินค้า และ Attribute ต่าง ๆ ของสินค้า ซึ่งมีคอลัมน์ต่าง ๆ ดังตัวอย่าง ๆ

TransactionDate	Time	PartNo	Serial	Location	InOut	PriceIn	PriceOut	Available	Total	Rank	Stocker
-----------------	------	--------	--------	----------	-------	---------	----------	-----------	-------	------	---------

ตารางที่ 3.7 ตารางตัวอย่าง KMITL\_Pen\_Red\_Pentel

### 3.1.2.3 ผู้ใช้รายการสินค้าทำการกรอกข้อมูลส่วนตัว

ก่อนที่ผู้ซื้อรายการสินค้าจะสามารถทำการได้นั้น จะต้องทำการลงทะเบียนก่อน โดยทำการผ่านมือถือระบบ WAP ซึ่งจากหน้าจอเมนูหลัก เลือกลิงค์โยงไปที่ “SIGNIN” แล้วทำการกรอกข้อมูลส่วนตัวคือ บริษัท Username Password และ ชื่อ-นามสกุล

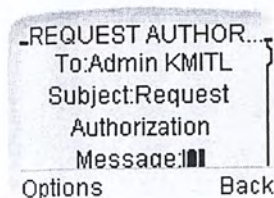


รูปที่ 3.6 หน้าจอการ Sign in ผ่านทาง WAP

หลังจากนั้นข้อมูลดังกล่าวจะถูกนำไปเก็บไว้ใน ตาราง Username\_Password โดยมีสถานะเป็น Awaiting คือยังไม่สามารถทำการเช็คสินค้าได้ นอกจากนี้ Username จะถูกนำไปสร้างเป็น ตาราง Username\_Message เพื่อไว้ใช้เก็บ Message ต่าง ๆ

### 3.1.2.4 ผู้ที่จะเช็คสินค้า ส่งข้อความเพื่อขออนุญาต จาก Admin ของ บริษัท

ต่อมาหน้าจอของมือถือจะปรากฏแบบฟอร์มการส่ง Message ซึ่ง จะเป็นการส่ง Message ให้กับ Admin ของบริษัทนั้น ๆ ซึ่งผู้ที่ทำการเช็ครายการสินค้าต้อง ส่งข้อความไปเพื่อขออนุญาตจาก Admin ของบริษัท

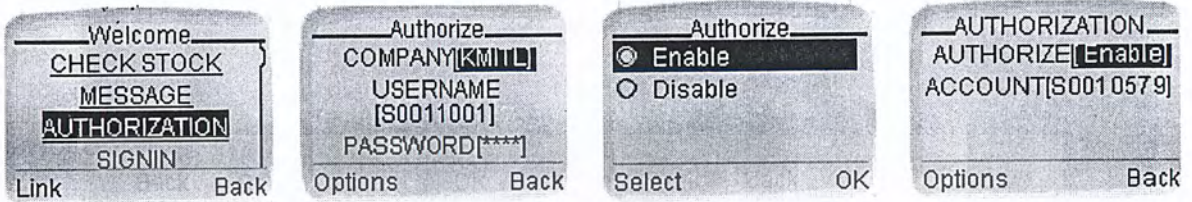


รูปที่ 3.7 การส่งข้อความหา Admin เพื่อขอ Authorize

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

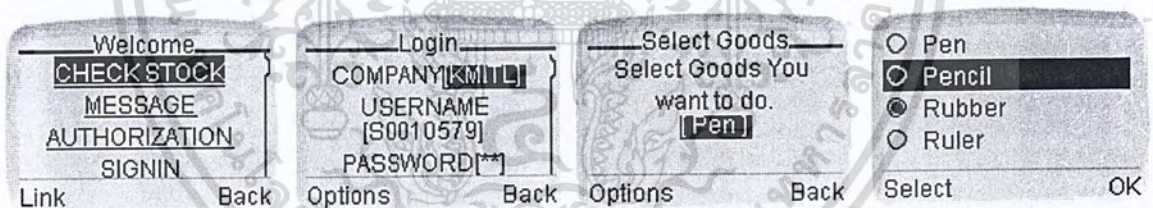
4. Admin ของบริษัท ทำการ Enable Account ของผู้เช็คสินค้า

ขั้นตอนนี้จะเกิดขึ้นเมื่อ Admin ของบริษัท เข้ามาตรวจเช็ค Message ของคนแล้วพบว่ามีความ  
ขออนุญาตทำการเช็คสินค้า โดย Admin จะกลับเข้าไปเลือก Authorize แล้วทำการ Enable Account ของ  
ผู้ทำรายการสินค้าที่ต้องการ



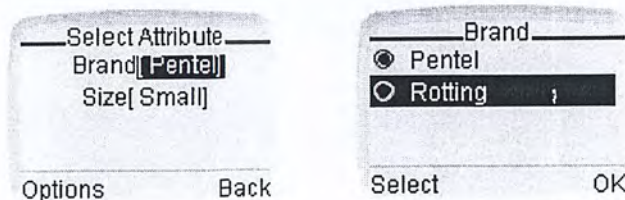
รูปที่ 3.8 หน้าจอการให้ Authorization

5. ผู้ที่จะเช็คสินค้าสามารถทำรายการต่างๆได้โดย login โดยใช้ Username แล้ว Password ของตนเอง  
ซึ่ง Username และ Password ที่ใช้ Login เข้าไปจะถูกนำไปตรวจสอบว่า Username นี้ได้รับ  
Authorize แล้วหรือไม่ ถ้าได้ Username นี้ ได้ถูกทำการ Enable เรียบร้อยแล้ว ก็จะสามารถเข้าสู่หน้า  
จอต่อไปเพื่อจะทำการเลือกชนิดของสินค้าได้



รูปที่ 3.9 หน้าจอการเลือกสินค้าที่จะทำรายการ

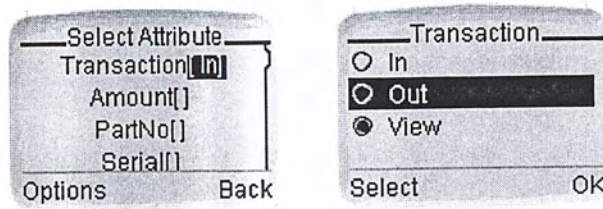
หลังจากเลือกชนิดสินค้าได้แล้ว จะเข้าสู่หน้าจอถัดไปเพื่อเลือก ประเภทของสินค้านั้น ๆ ตาม  
Attribute ที่มีอยู่ดังรูป



รูปที่ 3.10 การเลือกประเภทของสินค้านั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่เลือกสินค้าและประเภทของสินค้าได้เรียบร้อยแล้ว ก็จะเข้าสู่การทำรายการสินค้าประเภทนั้น ๆ ซึ่งเราสามารถทำการได้ 3 รายการคือ นำสินค้าเข้า นำสินค้าออก และดูรายการสินค้าคงเหลือ



รูปที่ 3.11 เมนูเลือกทำรายการ

### 1. การนำสินค้าเข้า

การนำสินค้าเข้าทำได้โดยเลือก field Transaction ให้เป็น "In" แล้วทำการใส่รายละเอียดต่าง ๆ ลงไป หลังจากนั้น ข้อมูลต่าง ๆ ที่ได้ใส่ลงไปนั้น จะถูกนำไปเก็บในฐานข้อมูลในตารางตาม ประเภทของสินค้านั้น ๆ ซึ่ง ในตารางนั้นจะเก็บ วันเวลาที่ทำการ (TransactionDateTime) จำนวนสินค้าที่เข้า (InOut) จำนวนสินค้าที่นำออกได้ (Available) จำนวนสินค้าทั้งหมด (Total) ลำดับนำเข้าสินค้า (Rank) และผู้ทำรายการ (Stocker) และค่าต่าง ๆ ที่ได้ใส่ลงไป

TransactionDateTime	PartNo	Serial	Location	InOut	PriceIn	PriceOut	Available	Total	Rank	Stocker
Mon Mar 19 00:25:13 GMT+07:00 2001	1A	1000	BKK	5000	2.50	3.00	5000	5000	1	S0010579
Mon Mar 19 00:25:53 GMT+07:00 2001	2A	1001	BKK	5000	2.50	3.00	5000	10000	2	S0010579
Mon Mar 19 00:26:40 GMT+07:00 2001	1B	2000	CHA	5000	2.50	3.00	5000	15000	3	S0010579

ตารางที่ 3.8 ตัวอย่างตาราง KMITL\_Pen\_Red\_Pentel หลังจากทำการ In

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การนำสินค้าออก

การนำสินค้าออกทำได้โดยเลือก field Transaction ให้เป็น “Out” แล้วทำการระบุ ส่วนเฉพาะของสินค้าที่ต้องการจะนำออก และจำนวนที่ต้องการนำออก ซึ่ง ค่าต่าง ๆ จะ ถูกนำไปเก็บในฐานข้อมูลในตารางตามประเภทของสินค้านั้น ๆ เช่น เดียวกันกับการนำ สินค้าเข้า

การนำสินค้าออกจะมีจุดที่ละเอียดอ่อนกว่าการนำสินค้าเข้าก็คือ การนำสินค้าออกนั้น สินค้าที่นำออกจะถูกเช็คว่ามีเหลือพอจะนำออกหรือไม่ (Available) ถ้าไม่พอ ก็จะมีการ แจ้งเตือนว่า ทำรายการผิดพลาด หรือในกรณีที่ได้อัปเดตระบบการนำสินค้าออกเป็นแบบ FIFO (FIRST – IN – FIRST – OUT) ก็จะมีการแจ้งเตือนในกรณีที่สินค้าที่นำออกไม่ใช่ ชุดแรกที่ถูกลำเข้า หรือถ้าเป็นระบบการนำสินค้าออกแบบ LIFO (LAST – IN –FIRST – OUT) ก็จะมีการแจ้งเตือนในกรณีที่สินค้าที่นำออกไม่ใช่ชุดสุดท้ายที่ถูกลำเข้า



รูปที่ 3.12 หน้าจอเตือนสินค้าไม่พอ

รูปที่ 3.13 หน้าจอเตือนเรื่อง FIFO

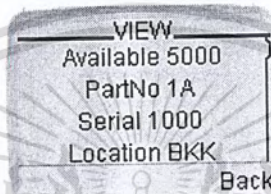
นอกจากนี้เมื่อได้ทำการนำสินค้าออกแล้ว ยังจะมีระบบแจ้งเตือนสินค้าหมด ในกรณีที่เรารั้ง ค่า ALERT ไว้ในตอนแรก ซึ่งโปรแกรมจะทำการเปรียบเทียบค่าของสินค้าคงเหลือ (Total) กับ จำนวนที่ต้องแจ้งเตือน (ALERT) ที่ได้ตั้งไว้ ถ้าสินค้าคงเหลือถึงระดับแจ้งเตือน ก็จะมีการแจ้งเตือนกับผู้ที่ทำรายการสินค้า ให้ทำการนำเข้ามาสินค้าเพิ่มเติมด้วย



รูปที่ 3.14 หน้าจอเตือนสินค้าใกล้หมด

### 3. ดูรายการสินค้าคงเหลือ

การดูรายการสินค้าคงเหลือทำได้โดยเลือก field Transaction ให้เป็น “View” โดยการ View นั้นจะแสดงถึงจำนวนสินค้าที่สามารถนำออก (Available) แต่ละชุดที่ยังคงเหลือในคลัง โดยจะเรียงลำดับตามการนำเข้าของสินค้า



รูปที่ 3.15 หน้าจอหลังกด View

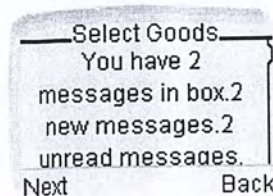
### การรับส่งข้อความ

Application อีกอย่างใน โครงงานนี้ก็คือ การรับส่งข้อความ ซึ่งการรับส่งข้อความนี้ได้ถูกกล่าวถึงมาแล้วครั้งหนึ่งใน ส่วนของการขอ Authorize สำหรับผู้ที่ต้องการทำการทำรายการสินค้า โดยผู้ที่ต้องการทำรายการสินค้าจะส่งข้อความเพื่อขออนุญาตทำการจาก Admin ของบริษัท

การรับส่งข้อความนี้เป็น Application ที่มีอยู่แล้วสำหรับ Admin ของแต่ละบริษัทหลังจากที่ได้ลงทะเบียนในหน้าแรกของการกรอกข้อมูลรายละเอียดของบริษัท แต่สำหรับผู้ใช้รายอื่นจะเข้ามาใช้ได้ โดยต้องทำการลงทะเบียนก่อน โดยทำการผ่านมือถือระบบ WAP ซึ่งจากหน้าจอเมนูหลัก เลือกลิงค์โยงไปที่ “SIGNIN” แล้วทำการกรอกข้อมูลส่วนตัวคือ บริษัท Username Password และ ชื่อ-นามสกุล

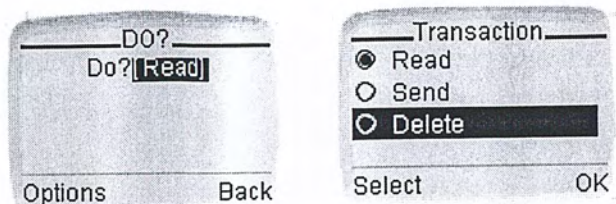
การเข้ามาใช้งานส่วนของการรับส่งข้อความทำได้โดยผ่านมือถือระบบ WAP ซึ่งจากหน้าจอเมนูหลัก เลือกลิงค์โยงไปที่ “MESSAGE” หลังจากนั้นจะปรากฏหน้าจอให้ใส่ Username และ Password ซึ่ง Username และ Password จะถูกนำไปตรวจสอบ จากนั้น หน้าจอของโทรศัพท์จะปรากฏข้อความแสดงจำนวนข้อความที่มีอยู่ทั้งหมด จำนวนข้อความใหม่ และจำนวนข้อความที่ยังไม่ถูกอ่าน

เมื่อกดปุ่ม “Next” เพื่อไปยังหน้าถัดไป จะมี option ให้เลือก 3 ชนิดคือ อ่านข้อความ ส่งข้อความ และลบข้อความ



รูปที่ 3.16 หน้าจอแสดงจำนวนข้อความ

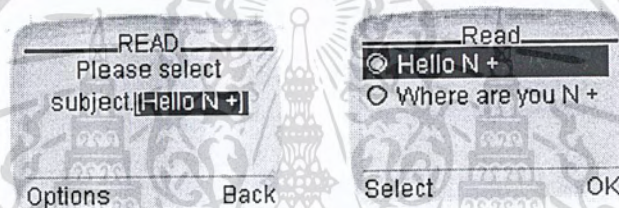
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 หน้าจอการเลือกจัดการกับข้อความ

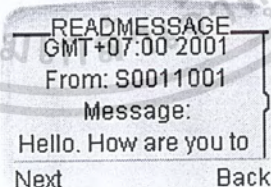
### 1. การอ่านข้อความ

ทำได้โดยการเลือก option เป็น “Read” หลังจากกดปุ่ม next แล้ว จากนั้นจะมี หัวเรื่อง ของข้อความมาให้เลือก ซึ่งจะกำกับด้วยถ้าเป็นข้อความใหม่จะปรากฏตัวอักษร “N” ต่อท้าย และถ้าข้อความนั้นยังไม่ถูกอ่าน จะปรากฏ อักษร “+” ต่อท้ายด้วย



รูปที่ 3.18 หน้าจอเลือกอ่านข้อความ

หลังจากเลือกข้อความแล้ว ผลที่ได้จะมี วันเวลาที่ข้อความนั้นส่งมา ชื่อผู้ส่ง และ เนื้อ  
ความ



รูปที่ 3.19 หน้าจอแสดงผลการอ่านข้อความ

### 2. การส่งข้อความ

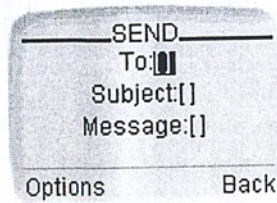
ทำได้โดยการเลือก option เป็น “Send” หลังจากกดปุ่ม next แล้ว จากนั้นจะมี แบบฟอร์มให้กรอกข้อความลงไปซึ่งได้แก่

To – หมายถึง ชื่อของผู้รับ ซึ่งจะต้องตรงกับ Username ของผู้รับ

Subject – หมายถึง หัวเรื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

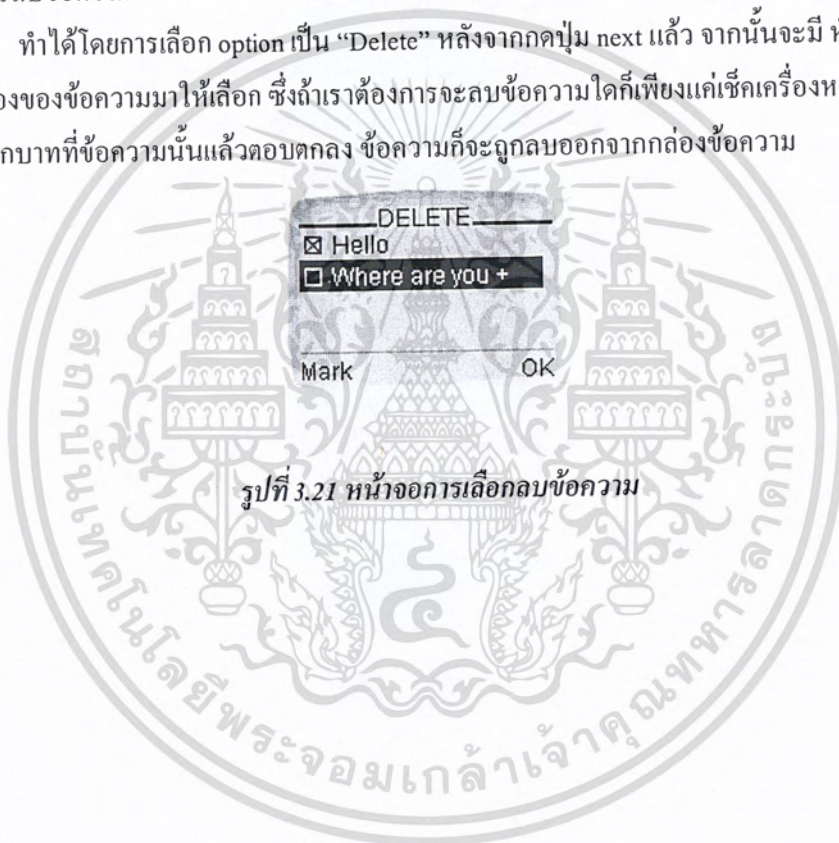
Message – หมายถึง ตัวเนื้อหาความ



รูปที่ 3.20 หน้าจอแบบฟอร์มการส่งข้อความ

### 3. การลบข้อความ

ทำได้โดยการเลือก option เป็น “Delete” หลังจากกดปุ่ม next แล้ว จากนั้นจะมี หัวเรื่องของข้อความมาให้เลือก ซึ่งถ้าเราต้องการจะลบข้อความใดก็เพียงแค่เช็คเครื่องหมายกากบาทที่ข้อความนั้นแล้วตอบตกลง ข้อความก็จะถูกลบออกจากกล่องข้อความ



รูปที่ 3.21 หน้าจอการเลือกลบข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 Check Mail

เป็นโปรแกรมสำหรับ ตรวจสอบ Email จาก Mail Server ที่รองรับการให้บริการผู้จดหมายแบบ POP 3 Server ซึ่งรายละเอียดของ POP 3 Server ได้อธิบายไว้แล้วในบทที่ 2 หัวข้อที่ Xxa

#### 3.2.1 ลักษณะทั่วไปของโปรแกรมตรวจสอบจดหมายอิเล็กทรอนิกส์

โดยทั่วไป โปรแกรมตรวจสอบจดหมายอิเล็กทรอนิกส์ จะมีลักษณะการทำงานคือ

1. ติดต่อเข้าเซิร์ฟเวอร์ ที่ให้บริการรับส่งจดหมายอิเล็กทรอนิกส์ และแสดงความเป็นผู้ใช้ด้วยการใส่ ชื่อรหัสผู้ใช้ และรหัสผ่านของผู้ใช้
2. ทำการอ่าน หรือเขียนและส่งจดหมาย รวมทั้งลบจดหมาย ที่อยู่ในตู้จดหมาย
3. ยกเลิกการติดต่อ

ลักษณะของข้อความของจดหมายอิเล็กทรอนิกส์ จะมีส่วนที่สำคัญคือ

Subject: เป็นส่วนที่บอกหัวข้อ ของจดหมายว่า จดหมายฉบับนี้ มีข้อความเกี่ยวกับอะไร

From: เป็นการระบุว่า ใครเป็นผู้ส่งจดหมายฉบับนี้

Date: ระบุวันที่จดหมายนี้ถูกส่งมาถึงเซิร์ฟเวอร์ ที่ให้บริการรับส่งจดหมาย

Status: ระบุสถานะของจดหมาย ว่าเคยถูกเปิดอ่านแล้วหรือไม่

Message: เป็นส่วนของข้อความของจดหมาย

#### 3.2.2 ส่วนประกอบของ Check Mail System

สำหรับโครงการนี้ โปรแกรมตรวจสอบและอ่านจดหมายอิเล็กทรอนิกส์มีรูปแบบการทำงานที่สามารถอธิบายในรูปแบบของ ออปเจกต์ได้ โดย Check Mail System ประกอบด้วย 3 Class คือ

- MainMail Class ทำหน้าที่เป็นตัวควบคุมหลัก ของระบบการตรวจสอบจดหมาย โดยจะทำหน้าที่รับคำร้องขอจากไคลเอนท์ แล้วทำการวิเคราะห์ และเรียกใช้บริการจากออปเจกต์ต่างๆ หรือจากตัวมันเอง

มีบริการคือ

doPost ทำหน้าที่รับคำร้องขอจากไคลเอนท์ และทำการวิเคราะห์ เพื่อตัดสินใจว่า จะต้องทำงานอะไรต่อไป

PrepareList1 ทำหน้าที่เตรียมค่านวนหาจำนวน Deck ที่ต้องมี เนื่องจากข้อจำกัดทางด้านหน่วยแสดงผล ที่เป็นหน้าจอโทรศัพท์มือถือ จึงกำหนดให้ 1 Deck จะแสดงจดหมายได้ 5 ฉบับ ทำให้ต้องมีส่วนค่านวนหาจำนวน Deck ที่ต้องใช้ จะรองรับจดหมาย 5 ฉบับ

PrepareList2 ทำการเตรียม การเขียนผลลัพธ์ที่เป็นหน้าจอแสดงจดหมาย และข้อความข้างในจดหมายต่างๆ

- MailConnection Class ทำหน้าที่เป็นตัวกลาง ในการติดต่อระหว่าง ออปเจกต์ MainMail กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เซิร์ฟเวอร์ของผู้จดหมาย ( POP3 Server )

มีบริการคือ

MailConnection เป็น Constructor ที่จัดการสร้างการติดต่อเริ่มต้นระหว่างแอปเจกต์ MainMail Connection กับ POP3 Mail Server โดยสร้างการติดต่อแบบ Socket IP

DoCommand ทำหน้าที่รับคำสั่ง ที่ส่งมาจาก MainMail แล้วทำการส่งต่อคำสั่ง ไปยัง Mail Server แล้วรอผลลัพธ์จากทาง Mail Server ที่ส่งกลับมา

GetMail ทำหน้าที่ ดึงจดหมายจาก Mail Server ตามหมายเลข ที่ส่งมาจาก Mail Connection

CountMail ทำหน้าที่นับจำนวนจดหมาย ที่อยู่ในผู้จดหมาย

ParseAnswer ทำหน้าที่ตรวจสอบ เนื้อหาจดหมาย แล้วแยกว่า จดหมายส่งมาจากใคร มีหัวข้ออะไร ส่งมาวันที่เท่าไร และมีส่วนที่เป็นข้อความจดหมายจริงๆอะไรบ้าง

IsNew ทำหน้าที่ตรวจสอบว่า จดหมายฉบับที่ต้องการตรวจสอบ เป็นจดหมายที่เคยถูกเปิดอ่านแล้วหรือไม่

- CardWriter Class เป็นแอปเจกต์ ที่มีหน้าที่สร้าง WML Page ขึ้น แล้วส่งต่อไปให้ โคลอนท์

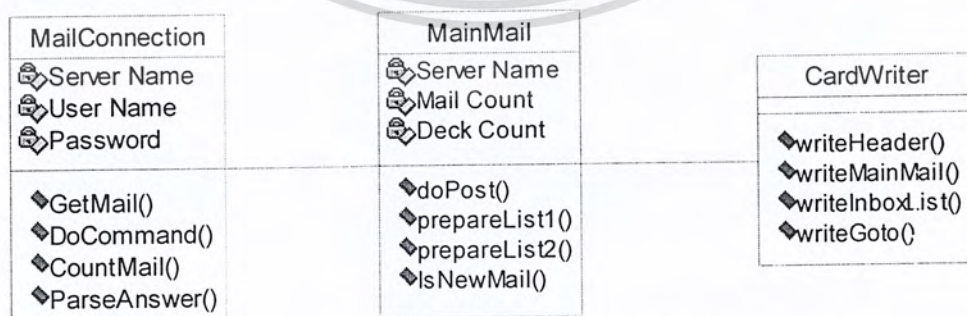
มีบริการคือ

WriteMainMail ทำหน้าที่ สร้าง WML Page ที่เป็นส่วนหน้าจอแรกเริ่ม เมื่อเข้าสู่โปรแกรม ตรวจสอบจดหมาย

WriteInboxList ทำหน้าที่สร้าง WML Page สำหรับ แสดง

WriteHeader ทำหน้าที่สร้าง Header ให้กับทุก WML Page

รูป Class Diagram ของ Check Mail System สามารถแสดงได้ดังนี้



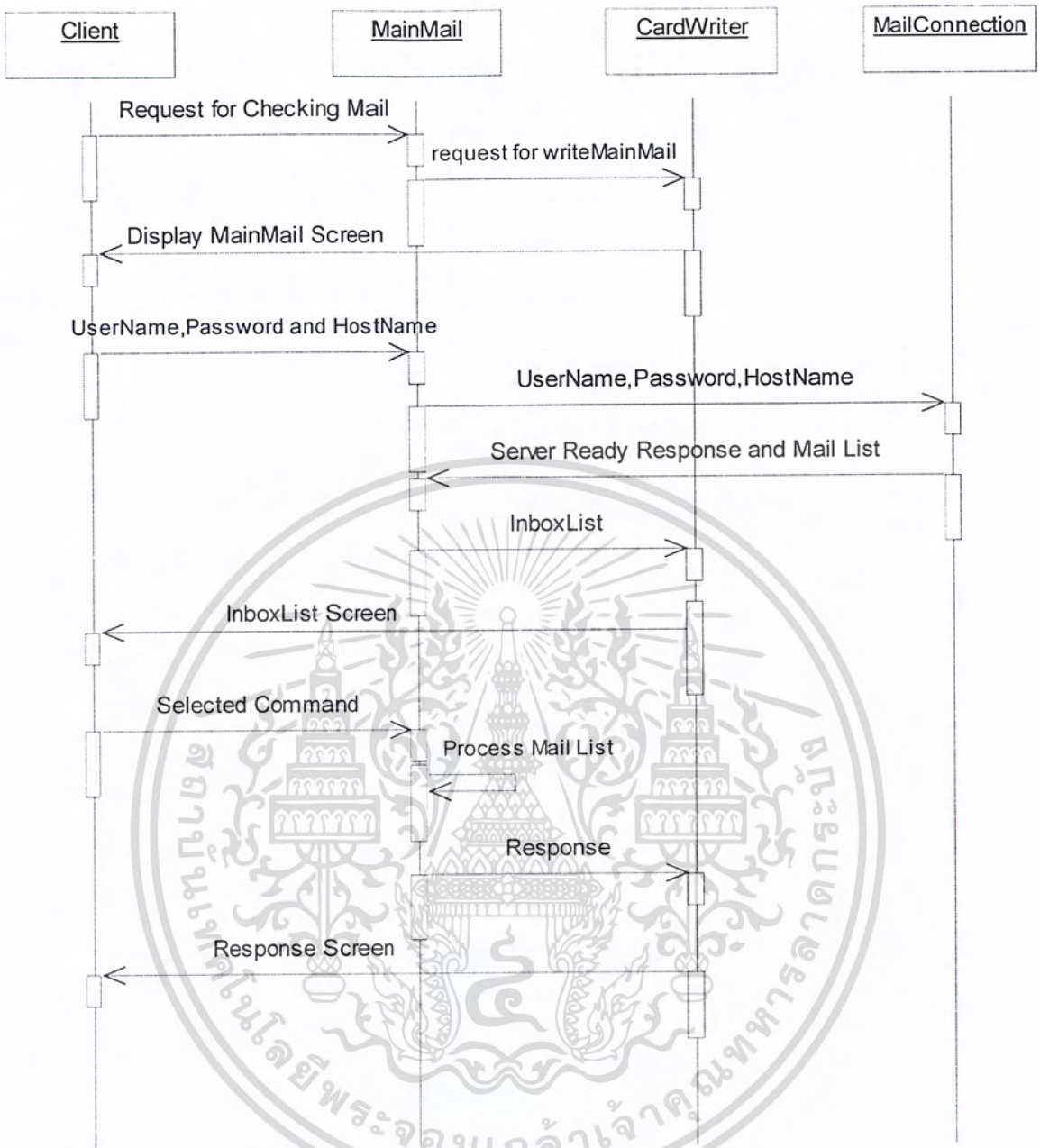
รูปที่ 3.22 แสดง Class Diagram ของ Check Mail System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 รูปแบบการทำงานของ การ ตรวจสอบจดหมาย มีดังนี้

1. ผู้ใช้ เลือกเรียกการทำงานตรวจสอบจดหมาย จาก WML Page หลักของโครงการ
2. ผู้ใช้ ใส่ ชื่อผู้ใช้ ( User Name ) , รหัสผ่าน ( Password ) และเลือกเซิร์ฟเวอร์ที่ให้บริการจดหมาย ( Mail Server )
3. เมื่อ MainMail ได้รับข้อมูลจาก ข้อ 2 แล้ว ก็จะเรียกใช้บริการจาก MailConnector เพื่อสร้างการติดต่อกับ เซิร์ฟเวอร์ที่ให้บริการจดหมาย ดังที่ผู้ใช้ได้เลือกไว้ ผ่านทาง Socket ซึ่งใช้หลักการ ดังที่ได้อธิบายไว้ ในบทที่ 2 หัวข้อที่ Xxa แล้วรอให้เซิร์ฟเวอร์ที่ให้บริการจดหมาย ตอบรับกลับมา
4. เมื่อสามารถสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ผู้จดหมายได้แล้ว MailConnector ก็จะใช้ชื่อผู้ใช้ และรหัสผ่าน ที่ผู้ใช้กำหนด ทำการขอเข้าใช้บริการ ( Log In ) แล้วรอการตรวจสอบรายชื่อผู้ใช้กับรหัสผ่าน จากฝั่งเซิร์ฟเวอร์ผู้จดหมาย
5. เมื่อเซิร์ฟเวอร์ผู้จดหมายทำการตรวจสอบรายชื่อผู้ใช้ และรหัสผ่าน และให้อนุญาตให้เข้าถึง Mail Box ได้แล้ว MailConnector ก็จะทำการส่งคำสั่ง เพื่อตรวจสอบจำนวนจดหมาย ที่มีอยู่ในผู้จดหมาย , ทำการอ่านจดหมายทั้งหมด มาไว้ในหน่วยความจำ , ตรวจสอบว่า มีจดหมายที่เข้ามาใหม่กี่ฉบับ โดยในขั้นตอนการอ่านจดหมายเข้ามาเก็บไว้ในหน่วยความจำนั้น จะมีการตรวจสอบว่า ขนาดเนื้อหา หรือข้อความในจดหมายนั้น มีขนาดใหญ่เพียงใด รวมทั้งตรวจสอบว่า มีการส่งแนบไฟล์อื่นๆที่ไม่ใช่ข้อความ มาพร้อมกับจดหมายด้วยหรือไม่ เพราะ อุปกรณ์สื่อสารไร้สายที่ผู้ใช้ ใช้ในการติดต่อ มักมีขนาดหน้าจอลดแสดงผล รวมทั้งหน่วยความจำที่จำกัด หากมีการแนบไฟล์มาในจดหมาย หรือข้อความมีขนาดใหญ่เกินไป หรืออยู่ในรูปแบบที่ไม่เหมาะสม MailConnector ก็จะไม่เก็บ จดหมายฉบับนั้นไว้ โดยหากผู้ใช้ จะเลือกอ่านจดหมายฉบับนั้น จะมีข้อความแจ้งให้ ผู้ใช้ เปิดอ่านจดหมายฉบับนั้น โดยใช้เครื่องคอมพิวเตอร์ส่วนบุคคลธรรมดาแทน
6. เมื่อ MailConnector ทำการอ่านจดหมายที่อยู่ใน Mail Box จนครบแล้ว ก็จะทำการยกเลิกการติดต่อกับเซิร์ฟเวอร์ผู้จดหมาย เพื่อลดจำนวนขั้นตอนการติดต่อ
7. จากข้อ 6 MailConnector จะทำการส่งชุดจดหมาย ( Mail List ) ที่ได้จากการอ่านจากเซิร์ฟเวอร์ผู้จดหมาย มาทำการจัดให้อยู่ในรูปแบบ ที่เหมาะกับการแสดงผล ในอุปกรณ์สื่อสารไร้สายโดยทั่วไป

ในการแสดงลำดับการทำงาน จะมีไคลเอนท์ ซึ่งหมายถึงผู้ใช้ ที่ใช้อุปกรณ์สื่อสารไร้ล : ย ในการติดต่อกับระบบ และสามารถแสดงผลภาพลำดับการทำงานได้ตามรูปที่ ๒๒



รูปที่ 3.23 แสดงลำดับการทำงาน (Sequence Diagram)

### 3.2.4 การเขียนโปรแกรมสำหรับ Check Mail System

ในการเขียนโปรแกรม สำหรับตรวจสอบจดหมายนั้น ต้องใช้ความรู้เรื่อง Socket Programming และ POP 3 Server รวมทั้ง Java Servlet ซึ่งทั้งหมดนี้ ได้อธิบายไว้แล้ว ในบทที่ 2 หัวข้อที่ Xxa ซึ่งสามารถอธิบายได้ โดยพิจารณาออปเจกต์ MainMail เป็นหลักดังนี้

#### ออปเจกต์ MainMail

ดังที่ได้แสดงไว้ใน Object Diagram จะเห็นได้ว่า MainMail มี การทำงานที่สำคัญคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. doPost เป็นส่วนสำคัญ ในการรองรับคำร้องขอจากไคลเอนต์ ดังที่ได้อธิบายไว้ในบทที่ 2 หัวข้อที่ xxx ซึ่งใน MainMail นี้ เมื่อได้รับคำร้องขอจาก ไคลเอนต์ แล้ว ก็จะทำการตรวจสอบ ค่าพารามิเตอร์บางค่า ที่ส่งมาจากไคลเอนต์ เพื่อทราบข้อมูลที่จำเป็น ซึ่งข้อมูลที่จำเป็นในที่นี้คือ “DoWhat”

การจะรับค่าพารามิเตอร์ ในที่นี้ ใช้คำสั่งของ Session ดังนี้

```
String DoWhat = req.getParameter("DoWhat");
```

เมื่อ doPost รับค่าพารามิเตอร์ DoWhat มาแล้ว ก็จะทำการตรวจสอบว่า จะต้องเรียกคำสั่ง หรือทำอะไรต่อไป โดยค่าพารามิเตอร์ของ DoWhat ที่เป็นไปได้มีดังนี้

- writeMainMail

เมื่อค่าพารามิเตอร์ของ DoWhat มีค่าเป็น “writeMainMail” doPost ก็จะทำการเรียก writeMainMail Method ซึ่งเป็นส่วน ที่ใช้เตรียม WML Page ซึ่งเป็นหน้าแรก ที่เริ่มเข้าสู่การให้บริการตรวจสอบจดหมาย ดังรูปที่ xxx ซึ่ง writeMainMail นี้ จะเรียกใช้ writeHeader ซึ่งเป็น method ที่จะถูกเรียกใช้ทุกครั้ง ที่มีการเตรียม WML Page writeMainMail method จะมีลักษณะ โครงสร้างดังนี้

```
public void writeMainMail(ServletOutputStream out) throws IOException
```

```
{
```

```
    writeHeader(out);
```

```
    //
```

```
    //
```

```
}
```

- CheckMail1 เรียก prepareList1 method และ prepareList2 method เพื่อทำการสร้าง WML Page สำหรับอ่านข้อความของจดหมาย

```
else if (DoWhat.equals("CheckMail1"))
```

```
{
```

```
    PrepareList1(MailCount,session);
```

```
    String CurrentDeckString = req.getParameter("CurrentDeck");
```

```
    Integer CurrentDeckInteger = Integer.decode(CurrentDeckString);
```

```
    int CurrentDeck = CurrentDeckInteger.intValue();
```

```
    try
```

```
    { PrepareList2(out,InboxList,session,CurrentDeck); }
```

```
    catch (Exception Err)
```

```
    { System.out.println("Error = "+Err); }
```

```
}
```

-- CheckMail2 เรียก prepareList2 สำหรับสร้าง WML Page ที่ไม่ใช่ชุดจดหมายข้อความ 1- 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (DoWhat.equals("CheckMail2"))
{
    String CurrentDeckString = req.getParameter("CurrentDeck");
    Integer CurrentDeckInteger = Integer.decode(CurrentDeckString);
    int CurrentDeck = CurrentDeckInteger.intValue();

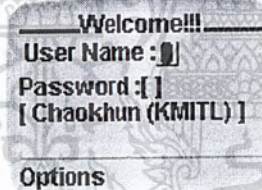
    try
    { PrepareList2(out,InboxList,session,CurrentDeck); }
    catch (Exception Err)
    { System.out.println("Error = "+Err); }
}

```

- prepareList1 จะทำการคำนวณหา จำนวนของ Deck ที่ต้องมี , จัดเก็บจำนวนจดหมายทั้งหมด ลงในตัวแปรของ session , จำนวนจดหมายที่มีในแต่ละ Deck
- prepareList2 จะทำการเตรียมเนื้อหา สำหรับการแสดงข้อความจดหมายแต่ละฉบับ

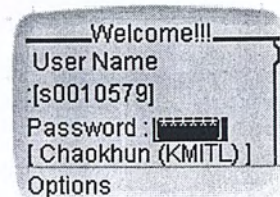
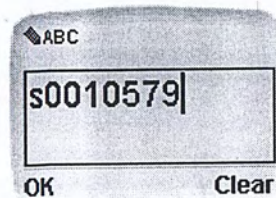
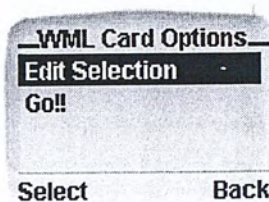
### 3.2.5 การทดสอบนำ Check Mail System มาใช้งาน

1. เมื่อเรียก Check Mail จะเข้าสู่หน้าจอหลัก



รูปที่ 3.24 แสดงหน้าจอแรกที่เข้าสู่โปรแกรมตรวจสอบจดหมาย

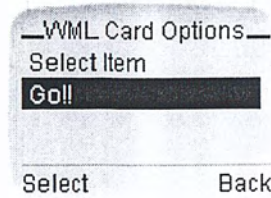
2. ทำการใส่ User Name , Password และ เลือก Mail Server



รูปที่ 3.25 รูปแสดงการใส่ข้อมูลของผู้ใช้

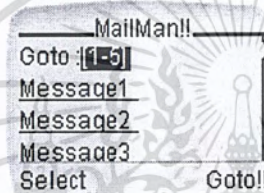
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการเลือก Options และ Go!! เพื่อสั่งให้ MainMail รับผิดชอบจาก เซิร์ฟเวอร์



รูปที่ 3.26 เลือก Go!! เพื่อให้ MainMail ทำการติดต่อกับเซิร์ฟเวอร์

4. จะขึ้นหน้าจอแรกของตู้จดหมาย โดยการส่ง WML Page จาก เซิร์ฟเวอร์มายังโคลเอนท์ จะส่งมาทีละ 5 การ์ด ซึ่งจะส่งถึงได้จากตรงช่องที่ให้เลือกหมายเลขจดหมาย จะเป็น 1-5 และลำดับต่อไปก็จะเป็น 6-10



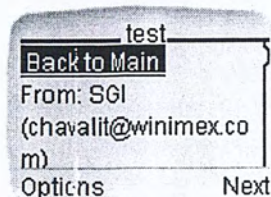
รูปที่ 3.27 รูปหน้าจอแสดงตัวเลือกสำหรับเลือกข้อความ

5. ตัวอย่างข้อความ ที่ไม่สามารถนำมาแสดงบนหน้าจอโทรศัพท์มือถือได้ เนื่องจากจดหมายมีขนาดใหญ่เกินไป ซึ่งในกรณี ผู้ใช้ต้องใช้เครื่องคอมพิวเตอร์ธรรมดา ในการเปิดอ่านจดหมาย



รูปที่ 3.28 ตัวอย่างแสดงข้อความที่ไม่สามารถอ่านได้

6. หากเป็นข้อความที่สามารถอ่านได้จะขึ้นข้อความดังรูป



รูปที่ 3.29 ตัวอย่างข้อความที่สามารถอ่านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### เครื่องมือที่ใช้ในการทำโครงการ

โครงการนี้ ต้องใช้เครื่องมือต่างๆ และมีวิธีการใช้ ดังต่อไปนี้

#### 4.1 Java Servlet Development Kit

เป็นชุดโปรแกรม สำหรับช่วยพัฒนา Java Servlet ซึ่งอธิบายในบทที่ 2 ผลิตและพัฒนาโดยบริษัท Sun Microsystem

##### 4.1.1 ส่วนประกอบของชุด Java Servlet Development Kit

ชุด Java Servlet Development Kit จะมีส่วนประกอบที่สำคัญคือ

- Class File ที่ใช้เป็น javax.servlet
- โปรแกรม Servletrunner
- เอกสารประกอบการใช้งาน
- ไฟล์ตัวอย่าง

##### 4.1.2 การติดตั้ง Java Servlet Development Kit

การติดตั้ง Java Servlet Development Kit นั้น สามารถทำได้โดยจัดเตรียมโปรแกรมสำหรับติดตั้ง ซึ่งสามารถดึงข้อมูลมาได้จากที่อยู่

<http://jserv.javasoft.com>

- ไฟล์ที่จัดเตรียมมา จะสามารถ Unpack ได้ทันที โดยเรียกไฟล์นั้น
- เมื่อติดตั้งเรียบร้อยแล้ว ต้องกำหนด Class Path สำหรับใช้ในการ Compile ซึ่งสามารถทำได้โดยใช้คำสั่ง เช่น
 

```
set classpath = c:\jsdk2.0\lib\jsdk2.jar
```
- กำหนด path สำหรับเรียกใช้ Servletrunner เช่น
 

```
C:\> set PATH=C:\jsdk\bin;%PATH%
```

##### 4.1.3 การใช้งานโปรแกรม servletrunner

เมื่อติดตั้งชุด Java Servlet Development Kit เรียบร้อยแล้ว สามารถทดลองใช้โปรแกรม Servletrunner ได้โดยเรียกคำสั่ง servletrunner เช่น

```
c:\>servletrunner
```

โดยผู้ใช้สามารถกำหนดค่าบางอย่างได้โดยใช้ตัวเลือกดังต่อไปนี้ หากไม่มีการกำหนดตัวเลือก ค่าเริ่มต้นจะเป็นดังต่อไปนี้

- p port เป็นหมายเลขของพอร์ต ที่รอการติดต่อเข้ามาของไคลเอนท์ ซึ่งโดยปกติ จะกำหนดค่าเป็นพอร์ตที่ 8080

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- b backlog เป็นค่าสูงสุด สำหรับเก็บ log file สำหรับการติดต่อที่เข้ามาจากไคลเอนท์ ปกติจะกำหนดค่าไว้ที่ 50
- m max เป็นค่าจำนวนการติดต่อจากไคลเอนท์สูงสุด ที่เซิร์ฟเวอร์จะสามารถรองรับได้ ปกติจะกำหนดค่าไว้ที่ 50
- t timeout เป็นค่าเวลาสูงสุด ที่ถ้าไคลเอนท์ไม่มีการติดต่อเข้ามา เซิร์ฟเวอร์จะยกเลิกการติดต่อกับไคลเอนท์นั้น มีหน่วยเป็นมิลลิวินาที ค่าปกติที่กำหนดไว้เป็น 5000 มิลลิวินาที
- d dir เป็นที่อยู่ชื่อของไดเรกทอรี ที่จัดเก็บไฟล์ servlet ไว้
- s filename เป็นชื่อของเก็บข้อมูลของการทำงานของ servlet ขณะที่เราเรียกใช้งาน servletrunner โดยปกติ จะมีค่าเท่ากับ (<servletdir>/servlets.properties)

ชุดโปรแกรม Java Servlet Development Kit ซึ่งในที่นี้ ขอเรียกว่า JSDK สามารถนำมาใช้ในการพัฒนาและทดสอบ servlet ในแบบ standalone คือทั้งไคลเอนท์ และ เซิร์ฟเวอร์ เป็นเครื่องคอมพิวเตอร์เครื่องเดียวกันได้ โดยการเรียกใช้โปรแกรมเซิร์ฟเวอร์อย่างง่าย คือ Servletrunner ในการทดสอบ servlet ก่อนที่จะนำ servlet ที่พัฒนา มาใช้งานจริงกับ Web Server

## 4.2 Nokia Wap Toolkit

เป็นชุดโปรแกรม ที่ผลิตโดยบริษัท Nokia สำหรับใช้พัฒนา และช่วยอำนวยความสะดวกในการเขียน WML Page ต่างๆ รวมไปถึงการเขียน WML Script และการวาดรูป ที่มีส่วนขยายเป็น wbmp

### 4.2.1 ส่วนประกอบของ Nokia WAP Toolkit

Nokia WAP Toolkit จะมีส่วนประกอบคือ

- ส่วน Toolkit หลัก ซึ่งจะมีส่วนประกอบย่อยคือ
  - WAP Browser
  - WAP Phone Simulator
  - WML , WML Script , WBMP Editor
  - WAP Protocol Stack , HTTP and files access modules
  - Debugging Views
  - WAP Server Simulator for device that require a connection to a WAP Gateway
  - WAP Push Simulator ( Blueprint Phone Simulator Only )
- เครื่องมือช่วยในการติดตั้ง โปรแกรม รวมทั้งไฟล์ตัวอย่างในรูปแบบของ WML , WML Script
- XML Library
- เอกสารคู่มือการใช้งาน ซึ่งอยู่ในรูปแบบของ PDF ( Portable Document Format )

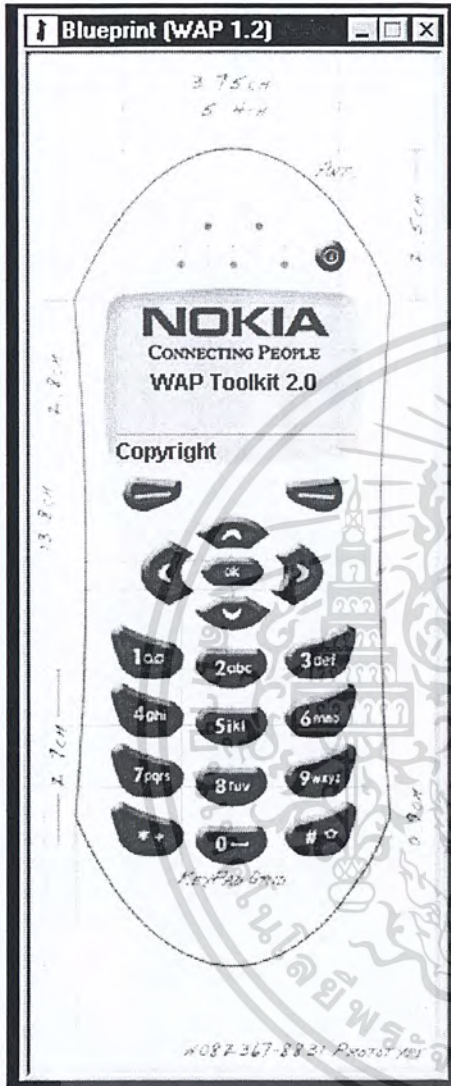
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.2 Toolkit หลักที่จำเป็นต่อการใช้งาน

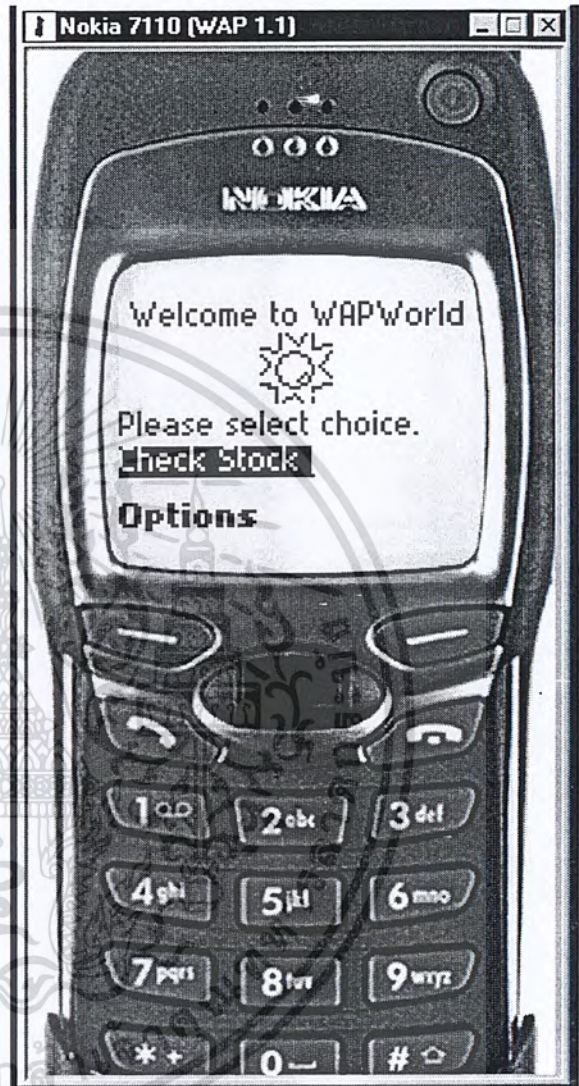
ส่วน Tool Kit หลัก มีส่วนที่ผู้ใช้ควรทำความคุ้นเคย เพราะจำเป็นต่อการใช้งานคือ

### 4.2.2.1 Phone Simulator

Phone Simulator เป็นโปรแกรมจำลองโทรศัพท์มือถือ ซึ่งสามารถใช้งาน WAP ได้



รูปที่ 4.1 แสดง Blueprint Phone Simulator



รูปที่ 4.2 Nokia 7110 Phone Simulator

โดย Phone Simulator ทั้ง 2 แบบ มีความแตกต่างกันคือ

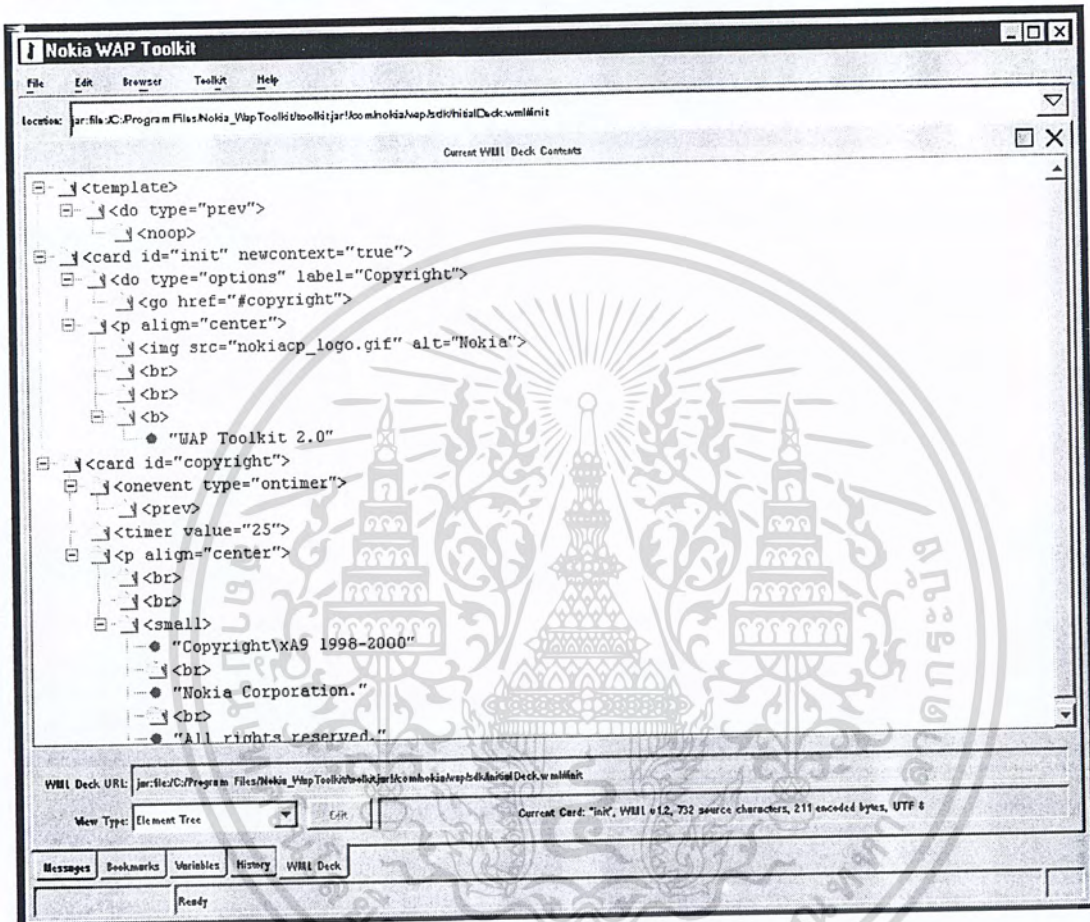
1. Blueprint Simulator มีลักษณะดังรูปที่ 4.1 โดยมีคุณสมบัติสามารถรองรับ WML Page Version 1.2 ได้ Blueprint Simulator นี้ ไม่มีการสร้างขึ้น เพื่อใช้งานจริง โดยเป็นเพียงแบบจำลอง สำหรับทดสอบโปรแกรม และ WML Page เท่านั้น ซึ่งในการใช้ทดสอบ Blueprint จะมีความสามารถ และความยืดหยุ่นมากกว่า Nokia 7110 Simulator ที่จะอธิบายในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Nokia 7110 Simulator มีลักษณะดังรูปที่ 4.2 มีคุณสมบัติรองรับ WML Page Version 1.1 มีความสามารถ และหน่วยความจำ เท่ารุ่นที่ผลิตเป็นโทรศัพท์เคลื่อนที่ ที่วางขายจริง ซึ่งจะมีข้อจำกัดด้านความยืดหยุ่น สะดวกในการใช้งาน ซึ่งน้อยกว่า Blueprint Simulator

#### 4.2.2.2 Editor

เป็นส่วนช่วยในการเขียน WML Page , WML Script มีลักษณะดังรูปที่ 4.3



รูปที่ 4.3 แสดงรูป Editor

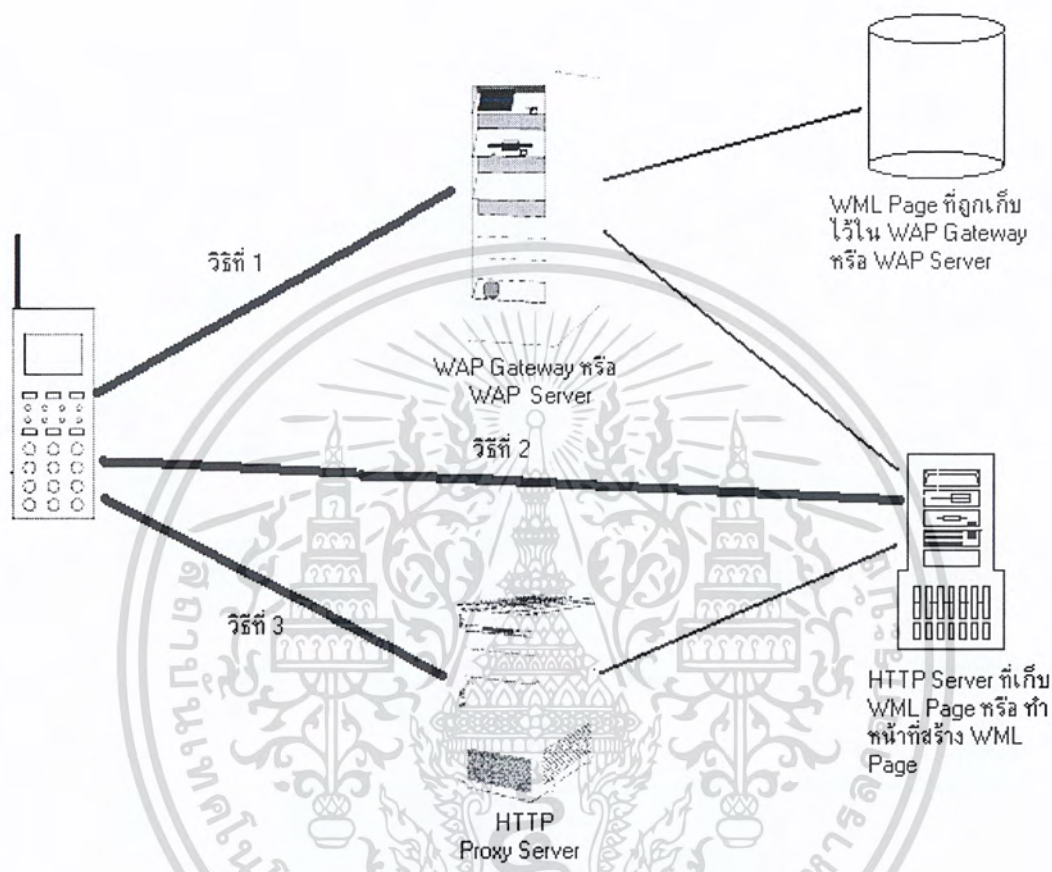
#### 4.2.2.3 Debugging Aids

ซึ่งจะมี History View , Session View และ Variable View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 การติดต่อระหว่าง Phone Simulator กับเซิร์ฟเวอร์

เพื่อความเข้าใจในการนำ Nokia WAP Toolkit มาใช้งาน ผู้ใช้ควรจะต้องเข้าใจหลักการทำงานของ Phone Simulation ของ Nokia WAP Toolkit ซึ่ง Phone Simulator ซึ่งเปรียบเสมือนเป็น ออปเจกต์หนึ่งในระบบ มีการติดต่อกับ Server ได้โดยวิธีต่างๆ 3 วิธี



รูปที่ 4.4 แสดงการติดต่อระหว่าง โคลนกับเซิร์ฟเวอร์ 3 วิธี

โดยวิธีติดต่อกับ เซิร์ฟเวอร์ ทั้ง 3 วิธีเป็นดังนี้

1. วิธีปกติ คือ โทรศัพท์มือถือจำลองที่ใช้ทดสอบ ทำการติดต่อกับ WAP Gateway หรือ WAP Server เพื่อส่งร้องขอ WML Page วิธีนี้ จะมีหลักการทำงานเหมือนการใช้งานโทรศัพท์เคลื่อนที่ ติดต่อใช้งาน WAP จริงๆ วิธีนี้ สามารถใช้กับโทรศัพท์มือถือจำลอง ทั้ง 2 แบบ ดังที่อธิบายไว้ข้างต้น
2. โดยตรง คือ แบบจำลองโทรศัพท์มือถือ ติดต่อเข้าหา HTTP Server โดยตรง ซึ่งวิธีนี้ จะสามารถใช้ได้กับโทรศัพท์มือถือจำลอง แบบ Blueprint Simulator เท่านั้น
3. โดยอ้อม มีลักษณะคล้ายกับแบบโดยตรง (แบบที่ 2 ) คือ การร้องขอ ไม่ต้องผ่าน WAP Gateway หรือ WAP Server แต่ติดต่อเข้าหา HTTP Proxy Server เลย ซึ่งการติดต่อแบบนี้สามารถใช้ได้กับโทรศัพท์มือถือจำลอง แบบ Blueprint Simulator เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.4 การใช้งานชุดโปรแกรม Nokia WAP Toolkit

##### 4.2.4.1 การใช้งานชุดโปรแกรม Nokia WAP Toolkit

ชุดโปรแกรม Nokia WAP Toolkit ต้องการความสามารถของเครื่องคอมพิวเตอร์ ขั้นต่ำดังต่อไปนี้

- เพื่อจะเรียกใช้งาน Nokia WAP Toolkit เครื่องคอมพิวเตอร์จะต้องมีคุณสมบัติคือ
  - มีระบบปฏิบัติการ Windows 95 , Windows 98 หรือ Windows NT ที่มี Service Pack3 อยู่ด้วย
  - หน่วยประมวลผล ที่มีความเร็วเทียบเท่า หรือสูงกว่า หน่วยประมวลผล เพนเทียม ความเร็ว 266 MHz
  - ฮาร์ดไดรฟ์มีเนื้อที่ว่าง อย่างน้อย 20 Mb.
  - มีหน่วยความจำหลัก อย่างน้อย 64 Mb.
  - หน่วยแสดงผล ซึ่งประกอบด้วย การ์ดแสดงผล และจอคอมพิวเตอร์สำหรับแสดงผล มีความสามารถในการแสดงสี อย่างต่ำ 65,000 สี
- มีชุด Java Runtime Environment รุ่น 1.2.2 หรือรุ่นที่ใหม่กว่า ซึ่ง Java Runtime Environment นี้สามารถหาได้จาก <http://www.javasoft.com/products/jdk/1.2/jre/index.html>
- หากต้องการทดสอบใช้บริการต่างๆ ที่มีอยู่ บนระบบเครือข่ายอินเทอร์เน็ต จะต้องมีการเตรียมการเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ตไว้
- ถ้าหากต้องการทดสอบบริการ ซึ่งพัฒนาขึ้นเอง จำเป็นต้องมี HTTP1.1 Web Server ด้วย

##### 4.2.4.2 ชุดโปรแกรม Nokia WAP Toolkit มีขั้นตอนการติดตั้งดังต่อไปนี้

1. จัดเตรียมโปรแกรมสำหรับติดตั้ง โดยสามารถดึงโปรแกรมมาได้จาก <http://www.forum.nokia.com>
2. ไฟล์ที่จัดได้จากข้อ 1 จะอยู่ในรูปของไฟล์ที่ถูกบีบอัดข้อมูล ( Zip File ) ซึ่งต้องทำการขยาย ( Unzip ) โดยโปรแกรม
3. เมื่อขยายข้อมูลแล้ว ก็เรียกใช้ไฟล์ Setup.exe เพื่อทำการติดตั้งโปรแกรม ซึ่งจะมีลำดับการทำงานของ setup.exe ดังนี้

#### 4.3 Microsoft Access 97

โปรแกรม Microsoft Access 97 เป็นส่วนหนึ่งของชุดโปรแกรม Microsoft Office 97 ซึ่งทำหน้าที่เป็น Database Management System ดังที่ได้กล่าวไปในบทที่ 2 หัวข้อที่ Xxa

##### 4.3.1 การติดตั้งโปรแกรม Microsoft Access 97 สามารถทำได้โดย

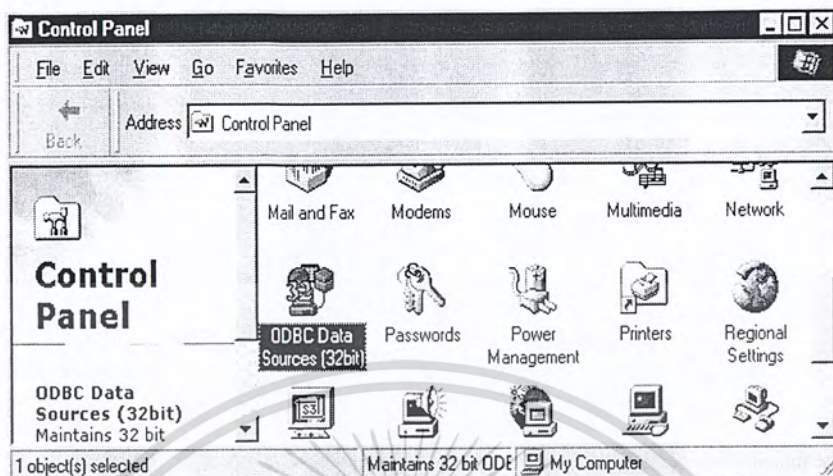
1. เรียกไฟล์ Setup.exe ซึ่งอยู่ในแผ่นซีดีรอม Microsoft Office
2. ทำการกำหนดค่าต่างๆ ตามที่จำเป็น
3. ทำการติดตั้ง ODBC Driver สำหรับ ฐานข้อมูล ที่สร้างขึ้น โดย Microsoft Access 97 นี้ เพื่อให้ Java Servlet สามารถติดต่อกับฐานข้อมูลได้ โดยผ่าน JDBC:ODBC...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 การติดตั้ง ODBC Driver

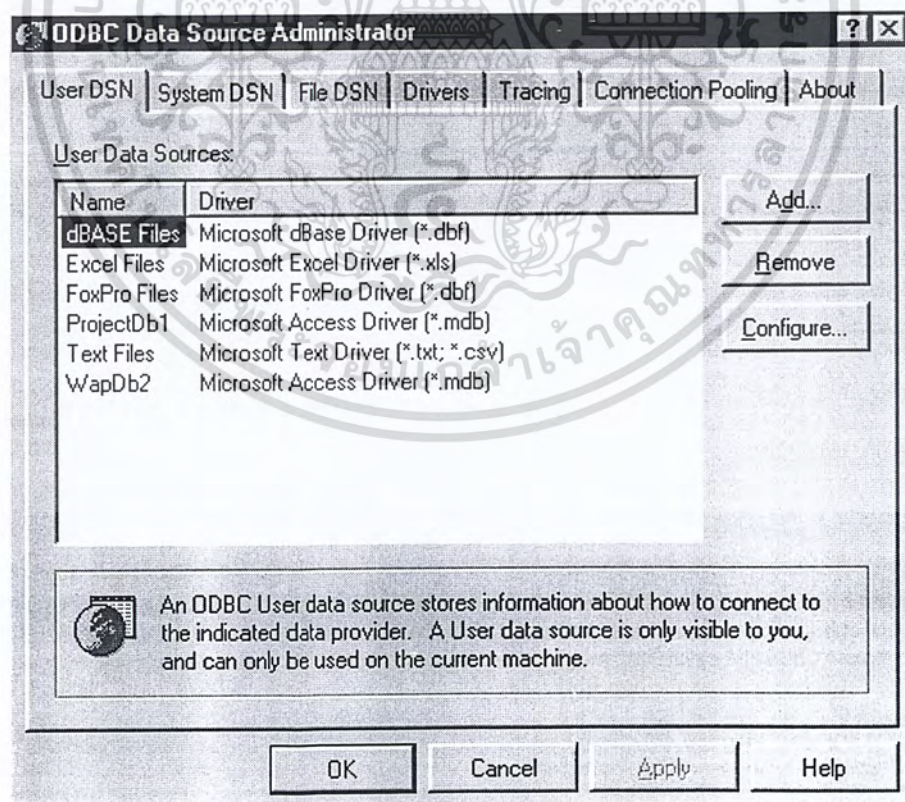
การติดตั้ง ODBC Driver ของ Microsoft Access 97 สามารถทำได้ ดังต่อไปนี้

#### 1. เรียก ODBC Data Sources (32 Bit)



รูปที่ 4.5 แสดงการเรียกติดตั้ง ODBC Driver

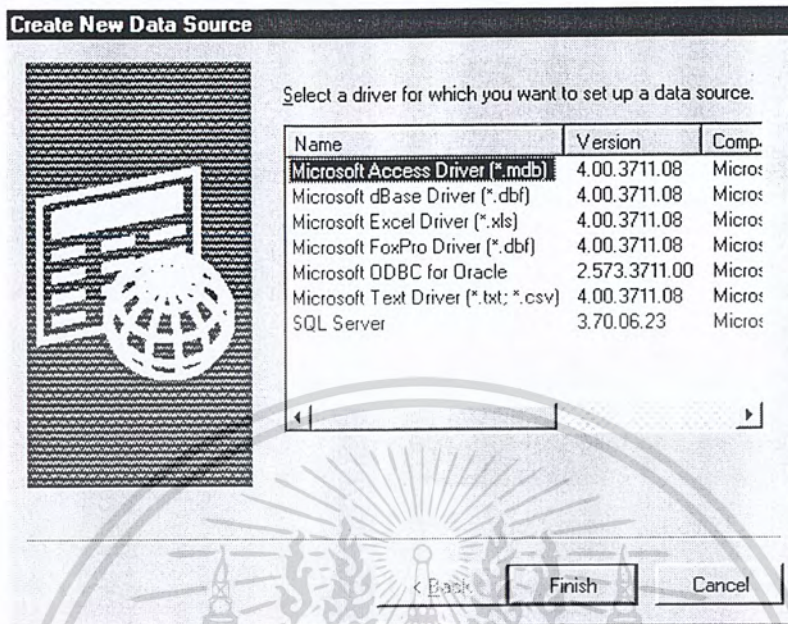
#### 5. เลือก Add



รูปที่ 4.6 การเพิ่มฐานข้อมูลสำหรับการใช้งาน

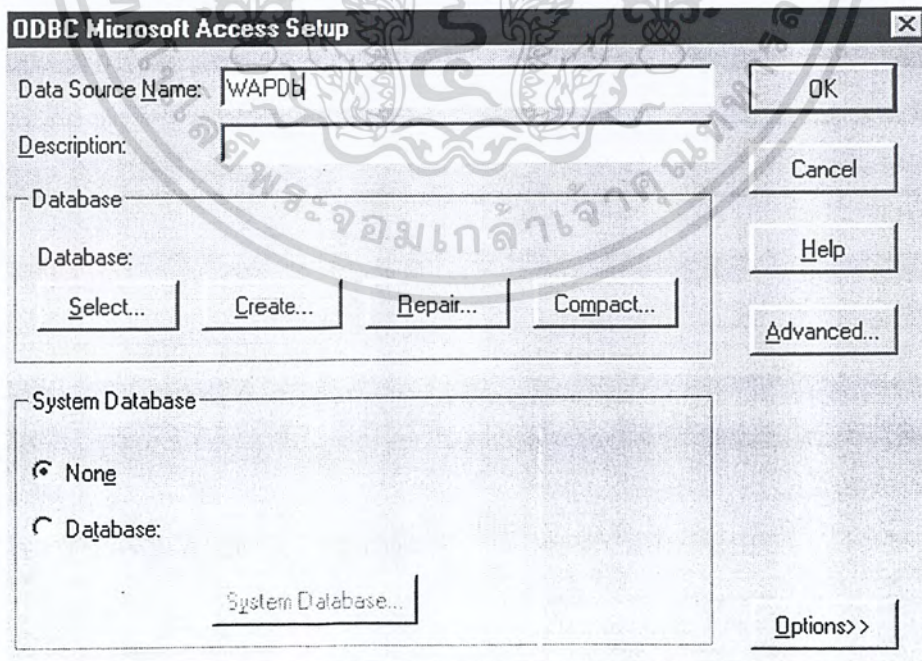
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. เลือก Microsoft Access Driver (\*.mdb)



รูปที่ 4.7 เลือกสร้าง Microsoft Access Driver สำหรับสร้างฐานข้อมูลใหม่

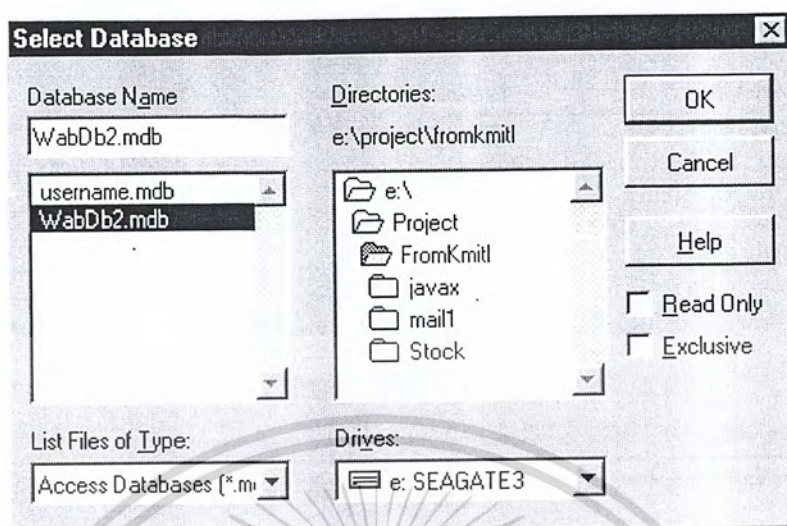
## 7. กำหนด Data Source Name



รูปที่ 4.8 การตั้งชื่อฐานข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8. เลือกไฟล์ Database ที่ต้องการ



รูปที่ 4.9 การเลือกชื่อไฟล์ฐานข้อมูล

### 4.4 Tomcat Web Server

เป็นโปรแกรมควบคุมเครื่องเซิร์ฟเวอร์ ให้สามารถรองรับ คำร้องขอต่างๆ ที่มาจากไคลเอนท์ และสามารถส่ง WML Page ที่ไคลเอนท์ต้องการ กลับไปให้ไคลเอนท์ได้ด้วย โดยเนื่องจากโครงการนี้ใช้ Java Servlet ซึ่งได้อธิบายไปในบทที่ 2 หัวข้อที่ Xxa ทำให้โปรแกรม Tomcat Web Server นี้ต้องมีความสามารถในการรองรับ Java ได้ด้วย ( Java - Enable ) โดยโปรแกรม Tomcat Web Server มีเป็นการพัฒนามาจากโปรแกรม Apache Web Server และ Apache J Serv ขั้นตอนการติดตั้ง Tomcat Web Server สามารถทำได้โดย

1. Download โปรแกรม
2. ติดตั้งโปรแกรม
3. กำหนดค่าเริ่มต้นใช้งานต่างๆ รวมทั้ง ใดเรกทอรี ที่ใช้เก็บไฟล์ กลาสต่างๆ

### 4.5 EditPlus 2.1a

โปรแกรม EditPlus ซึ่งนำมาใช้ในโครงการนี้ เป็นโปรแกรมแก้ไขไฟล์ต้นฉบับ ( Source Code ) รวมทั้งข้อความ, ไฟล์ HTML , ไฟล์โปรแกรมสำหรับภาษาต่างๆ ตามที่ผู้ใช้ต้องการ ซึ่งสามารถดาวน์โหลดโปรแกรม EditPlus ได้จากที่อยู่ <http://www.editplus.com> รายละเอียดการใช้งาน จะได้อธิบายไว้ในคู่มือการติดตั้งและใช้งาน

## บทที่ 5

### สรุป และวิจารณ์ ผลการทดลอง

#### สรุปและวิจารณ์ผลการทำงาน

จากระบบของโปรแกรม ที่ได้ทำการออกแบบและพัฒนาขึ้นมา นั้น เป็นระบบที่ผู้ใช้สามารถทำการ การสินค้าต่างๆ ได้อย่างถูกต้อง และในส่วนของการรับและอ่านจดหมายอิเล็กทรอนิกส์ สามารถแสดงข้อความ ได้ดี แม้ว่าต้องมีการเปลี่ยนแปลงสัญลักษณ์บางอย่างไปบ้าง

จากการที่ทำการพัฒนาโปรแกรมตามที่ได้ออกแบบมานั้น ผู้จัดทำประสบปัญหาบางประการ ซึ่งโดยมากจะอยู่ในขั้นตอนของการศึกษาเกี่ยวกับเซิร์ฟเลต ไม่ว่าจะเป็นการติดตั้ง หรือการใช้งาน รวมทั้งการเขียน โปรแกรมเซิร์ฟเลต ซึ่งจำเป็นต้องใช้ความชำนาญในการเขียนโปรแกรมด้วยภาษาจาวา แต่ชิ้นงานที่ได้ออกมา ก็มีคุณสมบัติตามที่ต้องการ แต่อาจยังไม่สมบูรณ์ และมีข้อผิดพลาดอยู่บ้าง หากจะมีการนำไปชิ้นงานที่มากขึ้น ไปใช้งานจริงในเชิงธุรกิจ ยังจำเป็นต้องมีการพัฒนาปรับปรุงแก้ไขต่อไป

แนวทางในการพัฒนาต่อไป

ในส่วนของการตรวจสอบรายการสินค้านั้น ยังขาดส่วนที่จำเป็นในระบบการเงินอยู่หลายอย่าง ซึ่งถ้าต้องการจะพัฒนาต่อไป ก็อาจจะต้องการศึกษาความต้องการในการใช้งานจริง และจากระบบบัญชีเพิ่มเติมในบางส่วน นอกจากนี้ ควรจะมีการปรับปรุงฐานข้อมูลที่ใช้ ให้มีความสามารถในการทำงานกับภาษาจาวามากกว่านี้ เช่น ใช้ชุดโปรแกรมฐานข้อมูล Oracle ในการจัดเก็บข้อมูล รวมทั้งควรพัฒนาส่วนติดต่อกับผู้ใช้ เพื่อให้ผู้ใช้ สามารถใช้งานระบบได้ง่ายและสะดวกขึ้น ในส่วนของการตรวจสอบจดหมายอิเล็กทรอนิกส์ มีข้อจำกัดในการแสดงผล ซึ่งขึ้นกับขนาดหน่วยแสดงผลของอุปกรณ์สื่อสารไร้สายที่นำมาใช้งานกับระบบ , ข้อจำกัด ในการรับจดหมายอิเล็กทรอนิกส์ ที่มีการแนบไฟล์มาด้วย , ข้อจำกัดที่มาจากภาษา WML ที่ใช้ เนื่องจากสัญลักษณ์บางอย่าง ที่อยู่ในข้อความในจดหมาย จะมีผลกับภาษา WML และข้อจำกัดที่สามารถรับจดหมายได้จาก POP3 Mail Server ได้ประเภทเดียว

## เอกสารอ้างอิง

- [1 ] Karl Moss , “Java Servlets” , McGraw – Hill , 1999
- [2] Java Distributed Objects , Bill McCarty and Luke Cassady – Dorion , 1<sup>st</sup> ed. , Sams , 1998
- [3] Jason Hunter and William Crawford , Java Servlet Programming , O’REILLY , 1998
- [4] Steve Mann , Programming Application with the Wireless Application Protocol The Complete Developers’ Guide , Wiley Computer Publishing , 2000



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

### ภาษา WML

ภาษา WML ( Wireless Markup Language ) เป็นภาษาที่ถูกออกแบบมา เพื่อให้สามารถจัดการกับการแสดงผลบนอุปกรณ์สื่อสารไร้สาย โดยลักษณะภาษา WML จะมีความคล้ายกันกับภาษา HTML ( Hyper Text Markup Language ) ซึ่งมีลักษณะสำคัญคือมี แท็ก ( Tag ) 2 ประเภท

1. แท็กว่าง ( Empty Tag )
2. แท็ก ที่มีข้อมูล

ซึ่ง แท็กแบบที่มีข้อมูลนี้ จะมี แอททริบิวต์ ( Attribute ) ต่างๆ ซึ่งจะกล่าวไว้ใน ภาคผนวก ข. ในภาคผนวก ก. นี้ จะเป็นการให้ตัวอย่างการเขียนภาษา WML เบื้องต้น เพื่อแสดงผลบนหน้าจออุปกรณ์สื่อสารไร้สาย ซึ่งในที่นี้ใช้ Blue Print 1.1 ของ Nokia WAP Toolkit

รูปแบบโครงสร้างของ ภาษา WML เป็นดังนี้

```
<?xml version = "1.0" ?>
```

```
<!DOCTYPE WML PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
http://www.wapforum.org/DTD/wml\_1.1.xml >
```

```
<wml>
```

```
<head>
```

```
· head information
```

```
</head>
```

```
<template>
```

```
· template definition
```

```
</template>
```

```
<card>
```

```
· card definition
```

```
</card>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

optional additional cards

```
</card>
```

```
</wml>
```

ตัวอย่างการแสดงผลเพื่อแสดงข้อความอย่างง่าย

ตัวอย่างที่ ป.ป

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="card1" title="Card #1">
```

```
<p align="center">
```

```
<big><b>First Card</b></big>
```

```
</p>
```

```
</card>
```

```
</wml>
```



ตัวอย่างที่ ป.ป

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="card1" title="Card #1">
```

```
<p align="center">
```

```
<a title = "Card 2" href = "#card2" >
```

```
Card 2
```

```
</a>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</p>
</card>

<card id="card2" title="Card #2">
  <p align="center">
    <a title = "Card 1" href = "#card1" >
      Card 1
    </a>
  </p>
</card>
</wml>

```

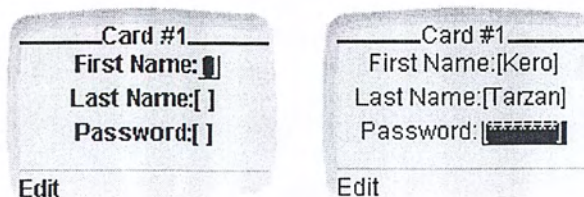
ตัวอย่างที่ ป.ป

```

<wml>
  <card id="card1" title="Card #1">
    <p align="center">
      First Name:
      <input name = "fname"
        maxlength = "15" /> <br/>
      Last Name:
      <input name = "lname"
        maxlength = "15" /> <br/>
      Password:
      <input name = "passw"
        maxlength = "8" type = "password" /> <br/>
    </p>
  </card>
</wml>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

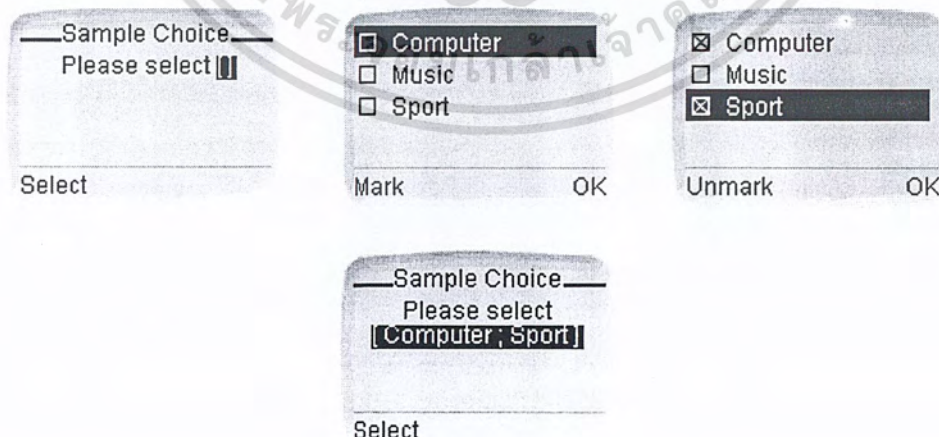


รูปที่ ป.ป

จะเห็นว่าส่วนที่เป็นข้อมูล password หรือรหัสผ่านนี้ เราสามารถกำหนดค่าให้ปรากฏบนหน้าจอแสดงผลเป็น \*\*\* ได้ ด้วยการกำหนดประเภทของข้อมูลให้เป็น password คือ type = "password"

ตัวอย่างที่ ป.ป

```
<wml>
<card id="card1" title="Sample Choice">
  <p align="center">
    <b> Please select </b>
    <select name = "select1" multiple = "true" >
      <option value = "computer" > Computer </option>
      <option value = "music" > Music </option>
      <option value = "spoart" > Sport </option>
    </select>
  </p>
</card>
</wml>
```



รูปที่ ป.ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ป.ป จะเห็นว่า เมื่อยังไม่มีการทำการเลือกตัวเลือกข้างใน ช่องข้อมูลจะยังว่างอยู่ แต่เมื่อมีการเลือกตัวเลือกแล้ว ก็จะมีข้อมูลที่ผู้ใช้เลือก ปรากฏขึ้น ตัวเลือกนี้ สามารถกำหนดได้ว่า ต้องการให้เป็นแบบเลือกครั้งเดียว ได้หลายตัวเลือก หรือ แต่ละครั้งที่เลือก สามารถเลือกได้เพียงตัวเลือกเดียว

### ตัวอย่างที่ ป.ป

```
<wml>
<card id="card1" >
  <p align="center">
    <big><b>Welcome To Mail Manager</b></big>
  </p>

  <do type = "unknown" label = "Check Mail">
    <go href = "http://127.0.0.1:8080/servlet/MainMail1" method = "post" >
      <postfield name = "DoWhat" value = "writeMainMail" />
    </go>
  </do>
</card>
</wml>
```



ในตัวอย่างนี้ เป็นการเขียน WML ให้สามารถติดต่อกับ servlet ซึ่งเป็นส่วนสำคัญของโครงการนี้ โดยสามารถเขียน WML ในส่วนที่ติดต่อกับ servlet ได้คือ

```
<go href = "http://127.0.0.1:8080/servlet/MainMail1" method = "post" >
  <postfield name = "DoWhat" value = "writeMainMail" />
</go>
```

ซึ่ง <go href = "http://127.0.0.1:8080/servlet/MainMail1" method = "post" > จะทำการติดต่อกับเซิร์ฟเวอร์ ที่ไคลเรททอรี ที่กำหนด คือ servlet ซึ่งทางฝั่งเครื่องเซิร์ฟเวอร์ จะเป็นผู้กำหนดเอง ว่าไคลเรททอรี servlet จะอยู่ที่ตำแหน่งใด ของเครื่องเซิร์ฟเวอร์ และเนื่องจากการติดต่อกับ เซิร์ฟเวอร์ มักจะต้องมีการส่งข้อมูลไปด้วย ซึ่งสามารถทำได้โดยใช้ส่วนขยายของ <go> คือ

```
<postfield name = "DoWhat" value = "writeMainMail" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้ เป็นการส่งข้อมูล ที่กำหนดชื่อเป็น DoWhat โดยส่วนข้อมูลจะมีค่าเป็น writeMainMail  
 ในส่วนของข้อมูล เราสามารถกำหนดขึ้นเอง หรือส่งค่าของตัวแปรได้ ซึ่งถ้ากำหนดค่าข้อมูล โดยตรง  
 จะมีลักษณะเป็น

```
<postfield name = "DoWhat" value = "writeMainMail" />
```

แต่ถ้าต้องการส่งค่าของตัวแปร สามารถกำหนดได้ดังนี้

```
<postfield name = "DoWhat" value = "$Var1" />
```

เมื่อ Var1 เป็นชื่อของตัวแปร ที่ต้องการส่งข้อมูลไปยังเครื่องเซิร์ฟเวอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

## สรุปไวยากรณ์และเงื่อนไขของภาษา WML

## Header

```
<?xml version = "1.0" ?>
```

## Decks and Cards

```
<wml xml:lang = "lang" >
```

```
content
```

```
</wml>
```

```
<card id = "name" title = "label" newcontext = "boolean" ordered = "boolean"
onenterforward = "href" onenterbackward = "href" ontimer = "href" >
```

```
content
```

```
</card>
```

```
<template onenterforward = "href" onenterbackward = "href" ontimer = "href"
```

```
content
```

```
</template>
```

```
<head> content </head>
```

```
<access domain = "domain" path = "path" />
```

```
<meta content = "value" scheme = "format" />
```

## Events

```
<do type = "type" label = "label" name = "name" optional = "boolean">
```

```
task
```

```
</do>
```

```
<onevent type = "type"> task </onevent>
```

```
<postfield name = "value" value = "value" >
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Tasks

```
<go href = "href" sendreferer = "boolean" method = "method"
  accept-charset = "charset" >
  content
</go>
```

```
<prev> content </prev>
```

```
<noop/>
```

```
<refresh> content </refresh>
```

## User Input

```
<input name = "variable" title = "label" type = "type" value = "value"
  value = "default" format = "mask" emptyok = "boolean" size = "n"
  maxlength = "n" tabindex = "n" />
```

```
<select title = "label" multiple = "boolean" name = "variable" value = "default"
  iname = "index_var" ivalue = "default" tabindex = "n" >
  content
</select>
```

```
<option title = "label" value = "value" onpick = "href">
  content
</option>
```

```
<optgroup title = "label" > content </optgroup>
```

```
<fieldset title = "label" > content </fieldset>
```

## Anchors

```
<anchor title = "label"> task text </anchor>
```

```
<a title = "label"> href = "href" text </a>
```

## Image

```
<img alt = "text" src = "url" localsrc = "icon" align = "alignment"
  height = "n" width = "n" vspace = "n" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Timers

<timer = name = “variable” value = “value” />

## Variables

<setvar name = “name” value = “value” />

## Layout and Formatting

<br>

<p align = “alignment” mode = “wrapmode” />

<table title = “value” align = “alignment” columns= “number” />

<tr> </tr>

<td> text layout img anchor a </td>

<em> text </em>

<strong> text </strong>

<b> text </b>

<i> text </i>

<u> text </u>

<big> text </big>

<small> text </small>

## Special Characters

&quot;            qutotaion mark

&amp;            ampersand

&apos;            apostrophe

&lt;                less than

&gt;                greater than

&nbsp;                non-breaking space

&shy;            soft hyphen ( discretionary hyphen )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค.

### สรุปคำศัพท์เฉพาะ

#### American Standard Code for Information Interchange ( ASCII )

เป็นมาตรฐานที่กำหนดโดยจีน เพื่อใช้สำหรับกำหนดรูปแบบของอักขระตัวอักษรต่างๆ โดยมีทั้งตัวอักษรตัวเล็กและตัวใหญ่ รหัสสำหรับการควบคุม และอื่นๆ

#### Attribute

องค์ประกอบสำหรับใช้เป็นเงื่อนไข โดยมักถูกนำมาประกอบรวมกัน เพื่อสร้างเป็น Element

#### Bandwidth

เป็นความสามารถในการส่งข้อมูลได้ ของตัวกลางของระบบสื่อสาร ไร้สาย ถ้าในระบบอนาล็อก จะวัดจากความถี่ แต่ถ้าเป็นระบบดิจิทัล จะวัดจากจำนวนบิต ที่สามารถส่งได้ต่อ 1 วินาที

#### Byte

เป็นกลุ่มของข้อมูลหลายบิต ที่มีความเกี่ยวข้องกัน อยู่รวมกันเป็น Byte

#### Bytecode

เป็นข้อมูล ที่ถูกเข้ารหัส ข้อมูล ซึ่งประกอบไปด้วย ออปโค้ด (Opcode) และ โอเปอเรนด์ (Operand) หรือที่เรียกว่าชุดคำสั่ง และข้อมูล ซึ่งการเข้ารหัสข้อมูลต่างๆนี้ เพื่อทำให้สามารถใช้งานกับแพลตฟอร์มของแต่ละเครื่องคอมพิวเตอร์ ที่ต้องการนำข้อมูลที่เข้ารหัสนั้นไปใช้ได้

#### Card

เป็นหน่วยย่อยที่สุด สำหรับการแสดงผลของ WML Page ซึ่งเนื้อหาภายใน อาจประกอบไปด้วย คำสั่งเพื่อรอการป้อนข้อมูลจากผู้ใช้ หรือคำสั่งสำหรับแสดงผลข้อมูลต่างๆ

#### Client

ในที่นี้หมายถึง อุปกรณ์ หรือ โปรแกรมประยุกต์ใดๆ ที่มีการสร้างการติดต่อกับเซิร์ฟเวอร์

#### Common Gateway Interface ( CGI )

ภาษาที่ใช้ในการกำหนดรูปแบบของเนื้อหา บน Web Site

#### Deck

ชุดของ Card ซึ่งอาจมีเพียง 1 Card หรือมากกว่า 1 Card มารวมกัน , WML Deck สามารถถูกเรียกได้ว่า เป็น XML Document ชนิดหนึ่งได้

#### Document Type Definition ( DTD )

ส่วนที่บอกข้อมูลประเภทของเอกสาร ซึ่งอาจมีการนำ DTD ชนิดหนึ่ง มาทำการต่อเชื่อมกับ DTD อีกชนิดหนึ่งได้

#### Element

ส่วนที่บอกข้อมูลต่างๆ ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Extensible Markup Language ( XML )

XML เป็นข้อตกลง ที่ถูกกำหนดโดยองค์การ WWW ซึ่งจะเป็นข้อตกลง เกี่ยวกับการกำหนด Markup Language ต่างๆ ที่ใช้บนระบบเครือข่ายอินเทอร์เน็ต

### Hyper Text Tranfer Protocol ( HTTP )

Hyper Text Tranfer Protocol ( HTTP ) เป็นโปรโตคอลพื้นฐาน ที่ใช้สำหรับการใช้งานอินเทอร์เน็ต ด้วย WWW ซึ่งรายละเอียดจะเกี่ยวกับการกำหนดรูปแบบข้อมูล และการรูปแบบการรับส่งข้อมูล

### Resource

ในที่นี้จะหมายถึงวัตถุที่อยู่บนระบบเครือข่าย หรือบริการที่สามารถเรียกใช้ได้ บนระบบเครือข่าย Resource นี้ อาจถูกจัดเตรียมไว้ในหลายรูปแบบ เช่น มีหลายภาษา , รูปแบบ , ขนาดที่แสดง

### Server

หมายถึงเครื่องคอมพิวเตอร์ หรือวัตถุใดๆบนระบบเครือข่าย ที่รอคอยการติดต่อเข้ามาของไคลเอนท์ โดย Server สามารถตรวจสอบและรับการติดต่อ หรือปฏิเสธการติดต่อจากไคลเอนท์ใดๆได้

### Tag

Tag คือ generic term of language element discriptor ในภาษา WML เนื้อหาและข้อมูลต่างๆ จะต้องถูกกำหนดด้วย Tag เสมอ Tag จะมีลักษณะคือมีเครื่องหมาย < ตามด้วย ชื่อของ Tag และ เครื่องหมาย > ปิดท้าย โดยปกติ การใช้ Tag จะใช้เป็นคู่ คือมี Tag เริ่มต้น และ Tag ปิดท้าย

### Terminal

เครื่องมือ ซึ่งมีความสามารถในการเป็น User Agent สำหรับผู้ใช้ สำหรับการใช้งานเป็นไคลเอนท์ ซึ่งรวมถึงความสามารถในการร้องขอ และรับข้อมูลต่างๆ Terminal ยังอาจถูกเรียกว่าเป็น Mobile Station หรือ Mobile Terminal ได้อีกด้วย

### Uniform Resource Locator ( URL )

เป็นที่อยู่บนอินเทอร์เน็ต ซึ่งมีการอ้างอิงด้วยข้อมูลที่ถูเก็บไว้ในระบบเครือข่ายอินเทอร์เน็ต

### User

บุคคล ที่ทำการติดต่อกับ User Agent เพื่อรับข้อมูลด้วยการอ่าน , ฟัง หรืออื่นๆ

### User Agent

เครื่องมือ หรืออาจรวมถึงโปรแกรมใดๆ ที่จะทำการแปลง ข้อมูลที่ถูกเขียนขึ้นด้วยภาษา WML , WML Script หรือ Resource อื่นๆ ซึ่ง User Agent นี้ อาจเป็น เบราวเซอร์ต่างๆ ที่มีความสามารถในการแปลงข้อมูลที่ถูกเขียนขึ้นด้วยภาษา WML อีกด้วย

### Web Server

Server ที่เก็บข้อมูล หรือสามารถสร้างข้อมูล รวมทั้ง Resource ต่างๆเองได้ เพื่อรองรับความต้องการของ User โดย Web Server อาจถูกเรียกว่าเป็น Origin Server หรือ HTTP Server ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Wireless Application Protocol ( WAP )

Wireless Application Protocol ( WAP ) เป็นข้อกำหนด ที่กำหนดขอบเขตการทำงาน รวมทั้ง โพรโตคอลต่างๆที่ใช้สำหรับระบบเครือข่าย รวมไปถึงโพรโตคอลที่ใช้ในการติดต่อสื่อสารกับอุปกรณ์สื่อสารไร้สาย โทรศัพท์มือถือ ( Mobile Phone ) , วิทยุติดตามตัว ( Pager ) , Personal Digital Assistance ( PDA )

### Wireless Markup Language ( WML )

ภาษา สำหรับกำหนดรูปแบบการแสดงผล ซึ่งถูกสร้างและกำหนดรูปแบบด้วย XML โดยหลักสำคัญสำหรับภาษา WML คือถูกออกแบบให้เข้ากับอุปกรณ์สื่อสารไร้สายต่างๆ ที่มีความเร็วในการรับส่งข้อมูลต่ำ WML เป็นภาษาที่ถูกใช้กับ WAP

### Wireless Markup Language Script ( WMLScript )

Wireless Markup Language Script ๖ ( WMLScript ) เป็น ภาษาสคริปต์ที่ถูกใช้สำหรับกำหนดการทำงานของอุปกรณ์โทรศัพท์มือถือ โดยเป็นส่วนขยาย ที่ถูกอ้างอิงและพัฒนามาจาก JavaScript



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ง.

## คำสั่งมาตรฐานของ POP3 Mail Server

คำสั่งมาตรฐาน สำหรับใช้ติดต่อกับ POP3 Mail Server มีดังต่อไปนี้

STAT

เป็นคำสั่ง ที่เรียกตรวจสอบจำนวนจดหมาย ที่มีอยู่ในตู้จดหมาย รวมทั้งขนาดเนื้อที่ ที่ใช้จัดเก็บจดหมาย

คำสั่ง STAT มีรูปแบบการใช้งานดังนี้

STAT

จะได้ผลลัพธ์ +OK nn mm

โดย nn คือตัวเลข แสดงจำนวน จดหมายที่มีอยู่

mm คือตัวเลข แสดงขนาดของเนื้อที่ทั้งหมด ที่ใช้ในการจัดเก็บจดหมาย

LIST

เป็นคำสั่ง ให้ Mail Server แสดงหมายเลขของจดหมายที่อยู่ในตู้จดหมาย พร้อมทั้งขนาดเนื้อที่ ที่ใช้จัดเก็บจดหมาย โดยถ้ามีการระบุหมายเลขของจดหมาย ก็จะแสดงเฉพาะ หมายเลข และขนาดเนื้อที่ที่ใช้จัดเก็บจดหมายฉบับนั้น แต่ถ้าไม่ระบุหมายเลขของจดหมาย ก็จะแสดงหมายเลขของจดหมายแต่ละฉบับ พร้อมทั้งขนาดเนื้อที่ที่ใช้จัดเก็บจดหมายแต่ละฉบับ

คำสั่ง LIST มีรูปแบบการใช้งาน 2 แบบคือ

1. LIST

จะให้ผลลัพธ์เป็น

+OK Mailbox scan listing follows

1 3564

2 2346

...

...

2. LIST n

เมื่อ n เป็นหมายเลขของจดหมาย และเมื่อป้อนคำสั่ง list 1 ไป จะได้ผลลัพธ์ คือ

list 1

+OK 1 3564

RETR

เป็นคำสั่งเรียกดูข้อความเนื้อหา n Resource

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้จะหมายถึงวัตถุที่อยู่บนระบบเครือข่าย หรือบริการที่สามารถเรียกใช้ได้ บนระบบเครือข่าย Resource นี้ อาจถูกจัดเตรียม ไว้ในหลายรูปแบบ เช่น มีหลายภาษา , รูปแบบ , ขนาดที่แสดง Server

หมายถึงเครื่องคอมพิวเตอร์ หรือวัตถุใดๆบนระบบเครือข่าย ที่รอคอยการติดต่อเข้ามาของไคลเอนท์ โดย Server สามารถตรวจสอบและรับการติดต่อ หรือปฏิเสธการติดต่อจากไคลเอนท์ใดๆได้ Tag.

Tag คือ generic term of language element descriptor ในภาษา WML เนื้อหาและข้อมูลต่างๆ จะต้องถูกกำหนดด้วย Tag เสมอ Tag จะมีลักษณะคือมีเครื่องหมาย < ตามด้วย ชื่อของ Tag และ เครื่องหมาย > ปิดท้าย โดยปกติ การใช้ Tag จะใช้เป็นคู่ คือมี Tag เริ่มต้น และ Tag ปิดท้าย

ขยในจดหมายฉบับนั้นๆ คำสั่งนี้ จึงต้องมีการระบุหมาย

เลขของจดหมายที่ต้องการอ่านข้อความเสมอ การใช้คำสั่งนี้ ควรจะระวังว่า จะมีการแนบไฟล์มา พร้อมกับจดหมายฉบับนั้นด้วยหรือไม่ เพราะคำสั่ง RETR นี้ จะเปิดอ่านและแสดงเนื้อหาข้างในทั้งหมด โดยไม่ตรวจสอบว่า มีไฟล์แนบมาด้วยหรือไม่ ซึ่งถ้ามีไฟล์แนบมาด้วย ข้อความที่แสดงออกมา ก็จะอยู่ในรูปของรหัสแอสกี

คำสั่ง RETR มีรูปแบบการใช้งานคือ

RETR n เมื่อ n เป็นหมายเลขของจดหมาย

retr 2

+OK 745 octets

Received: from diamond.ce.kmitl.ac.th (s0010579@diamond.ce.kmitl.ac.th [161.246.4.3])

by Chaokhun.kmitl.ac.th (8.9.1a/8.9.1) with ESMTP id NAA03723

for <s0010579@kmitl.ac.th>; Sun, 18 Mar 2001 13:41:09 +0700 (GMT)

Received: from localhost (s0010579@localhost)

by diamond.ce.kmitl.ac.th (8.11.1/8.11.1) with ESMTP id f2I6fWk01574

for <s0010579@kmitl.ac.th>; Sun, 18 Mar 2001 13:41:32 +0700 (TST)

Date: Sun, 18 Mar 2001 13:41:32 +0700 (TST)

From: Montri Rattanawiwatpong <s0010579@ce.kmitl.ac.th>

To: <s0010579@kmitl.ac.th>

Subject: Test1

Message-ID: <Pine.GHP.4.31.0103181341220.1571-100000@diamond.ce.kmitl.ac.th>

MIME-Version: 1.0

Content-Type: TEXT/PLAIN; charset=US-ASCII

Content-Length: 16

Status:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This is test1

จะเห็นว่า ในส่วนของ Status: จะเป็นส่วนที่บอกสถานะว่า จดหมายฉบับนี้ เคยถูกเปิดอ่านไปหรือไม่ ซึ่งจดหมายฉบับนี้ยังไม่ถูกเปิดอ่าน ถ้าถูกเปิดอ่านแล้ว จะมีผลลัพธ์เป็น

Status: RO

DELE

เป็นคำสั่งลบจดหมาย ซึ่งต้องมีการระบุหมายเลขของจดหมาย ที่ต้องการลบ เมื่อมีการใช้คำสั่งนี้ Mail

Server จะทำการทำเครื่องหมายไว้ที่จดหมายฉบับนั้น แต่ยังไม่ทำการลบจดหมายไปในทันที การ

ลบจด

หมายจะเกิดขึ้น เมื่อมีการใช้คำสั่ง UPDATE หรือ QUIT ซึ่งจะได้อธิบายหัวข้อถัดไป

คำสั่ง DELE มีรูปแบบการใช้งานคือ

DELE n เมื่อ n เป็นหมายเลขของจดหมาย

เมื่อทดลองเรียกคำสั่ง จะได้ผลลัพธ์ดังนี้

DELE 2

+OK Message deleted

NOOP

เป็นคำสั่งยืนยันการติดต่อกับ Mail Server เพื่อป้องกันการถูกยกเลิกการติดต่อจาก Mail Server

เนื่องจาก Time Out

คำสั่ง NOOP มีรูปแบบการใช้งานคือ

NOOP

ควรจะได้ผลลัพธ์เป็น +OK เสมอเช่น

NOOP

+OK No-op to you too!

RSET

เป็นคำสั่ง ยกเลิกการลบจดหมาย ซึ่งเกิดจากการใช้คำสั่ง DELE แต่จะใช้ได้ ถ้ายังไม่มีกรเรียก

คำสั่ง UPDATE

คำสั่ง RSET มีรูปแบบการใช้งานคือ

RSET

เมื่อทดลองเรียกใช้คำสั่ง RSET จะได้ผลลัพธ์ดังนี้

RSET

+OK Reset state

QUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง ยกเลิกการติดต่อกับ Mail Server  
 คำสั่ง QUIT มีรูปแบบคือการใช้งานคือ  
 QUIT  
 และจะได้ผลลัพธ์คือ  
 +OK Sayonara

UPDATE State

เมื่อมีการเรียกคำสั่ง QUIT ซึ่งเจตเจหมายนั้น จะเข้าสู่สถานะ UPDATE คือจะทำการประมวลผล และจัดการกับจดหมายต่างๆ ตามคำสั่ง ที่ถูกเรียกใช้ก่อนมีการเรียกใช้คำสั่ง UPDATE สำหรับรายละเอียดทั้งหมด สำหรับ POP3 Mail Server สามารถค้นหาได้จากเอกสารข้อกำหนด rfc 1939 ซึ่งสามารถหาได้ ด้วยการ ใช้ Search Engine ต่างๆ และค้นหาคำว่า “rfc 1939”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้