

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบสั่งอาหารผ่านมือถือ

FOOD ORDER APPLICATION ON MOBILE



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

๒๕๖

๑๖/๑๒/๕๘

๒๕๖๘

ปีการศึกษา ๒๕๖๘

b.....	11565913
i.....	

เลขหมู่.....เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขทะเบียน.....59381.....ไปรษณีย์.....ทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัน,เดือน,ปี..... 2 ส.ย. 2549

FOOD ORDER APPLICATION ON MOBILE



TONG KAN DAOPRADUBWONG

**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2005**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ

ระบบสั่งอาหารผ่านมือถือ

FOOD ORDER APPLICATION ON MOBILE

ชื่อนักศึกษา

นาย ต้องกานต์ ดาวประดับวงษ์

45050476

ภาควิชา

คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา

อ.สายชล ใจเย็น

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2548

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	ดร.นवलสวาท หิรัญสกลวงศ์	
กรรมการ	อ. ศังกรศรีณีย์ ถ่องจุผล	
กรรมการและอาจารย์ที่ปรึกษา	อ.สายชล ใจเย็น	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ระบบสั่งอาหารผ่านมือถือ	
ชื่อนักศึกษา	นายต้องกานต์ ดาวประดับวงษ์	45050476
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2548	
อาจารย์ที่ปรึกษา	อ.สายชล ใจเย็น	

บทคัดย่อ

ปัญหาพิเศษนี้เสนอแนวทางในการพัฒนาซอฟต์แวร์บนโทรศัพท์มือถือ โดยจัดทำระบบสั่งอาหารผ่านทางโทรศัพท์มือถือ ระบบประกอบไปด้วยโปรแกรมบนมือถือใช้ J2ME ในการพัฒนา, เซิร์ฟเวอร์ ใช้ Tomcat, ฐานข้อมูลใช้ Microsoft Access, หน้าจอแสดงผลที่ร้านอาหารใช้ JSP ในการพัฒนา และทดสอบระบบโดยใช้ Emulator WTK22 ของโทรศัพท์มือถือ Sony Ericsson รุ่น K700i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	FOOD ORDER APPLICATION ON MOBILE	
Students	Mr. Tongkan Daopradubwong	45050476
Degree	Bachelor of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Academic Year	2005	
Special Project Advisor	Saichon Jaiyen	

ABSTRACT

This special problem is a development of software for ordering food via mobile phone. This software is developed using J2ME, TOMCAT, Microsoft Access and JSP. It is tested using Emulator WTK22 of Sony Ericsson (K700i).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องระบบการส่งอาหารผ่านมือถือสามารถสำเร็จ ลุล่วงได้ด้วยดี ด้วยความช่วยเหลือ และความร่วมมือจากหลายๆท่าน คณะผู้จัดทำต้องขอขอบพระคุณบุคคลที่มีส่วนช่วยให้การทำงานครั้งนี้เสร็จไปได้ด้วยดี คือ อาจารย์สายชล ใจเย็น อาจารย์ที่ปรึกษาปัญหาพิเศษฉบับนี้ อาจารย์ศังกรศรีณย์ ล่องซุผล และดร.นवलสวาท หิรัญสกุลวงศ์ ที่กรุณาให้คำแนะนำแนวคิดต่างๆในการทำปัญหาพิเศษนี้ดูแลเอาใจใส่และให้การสนับสนุนทางด้านซอฟต์แวร์ รวมทั้งเป็นผู้ตรวจความถูกต้องของโครงการปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบคุณบุคคลที่เกี่ยวข้องทุกฝ่าย ที่ทำให้การทำปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี รวมทั้งบิดา มารดา ผู้ให้กำเนิด อบรมเลี้ยงดู ตลอดจนให้ความสนับสนุนด้านกำลังใจและทุนทรัพย์

คณะผู้จัดทำ

มีนาคม 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปัญหา	1
1.4 ขั้นตอนในการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 เทคโนโลยีที่ใช้สำหรับแอปพลิเคชันบนมือถือ	3
2.1.1 Java Server Pages (JSP)	3
- JSP Containers	3
- การติดตั้ง Tomcat	3
- Web Applications	4
- ส่วนประกอบของ Web Application	5
- Integration with Tomcat	6
- การทำงานของ JSP	6
2.1.2 J2ME	7
- คอนพิทเจอร์ชั้นใน J2ME	8
- J2ME สำหรับอุปกรณ์ไร้สาย	9
- ความต้องการของระบบ	10
2.1.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME	10
- ไลบรารีของ CLDC	10
- ชั้นคลาสของ J2SE	10
- คลาสที่มีเฉพาะใน CLDC	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประเภทของข้อมูลพื้นฐาน	11
- ไบรารีของ MIDP	11
- คลาสโปรแกรมจัดการแอปพลิเคชัน	12
- คลาสของส่วนติดต่อกราฟฟิกับผู้ใช้ (GUI Class)	12
- คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)	12
- คลาสของเครือข่าย (Network Class)	12
2.1.4 MIDlets	12
2.1.5 การเชื่อมต่อ J2ME กับเครือข่าย	14
- Generic Connection Framework	15
- รูปแบบคำสั่งที่ใช้สร้างการเชื่อมต่อ	16
- วิธีการเชื่อมต่อเครือข่ายแบบ HTTP	17
2.2 เทคโนโลยีอื่นๆ	18
2.2.1 MySQL	18
- คุณสมบัติเด่นของ MySQL	18
- สถาปัตยกรรมของ MySQL	19
- หลักการทำงานของ MySQL	20
- วิธีการเชื่อมต่อจาก ไคลเอนต์ เข้าสู่ เซิร์ฟเวอร์	20
- ความสามารถของ MySQL	21
2.2.2 JDBC	22
- หลักการทำงานของ JDBC	23
- รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC	23
- โครงสร้างของ JDBC	25
- รูปแบบของ JDBC ไดรเวอร์ (JDBC drivers)	26
บทที่ 3 ขั้นตอนและวิธีการดำเนินการ	29
3.1 ขั้นตอนการวิเคราะห์ระบบการทำงาน	29
3.2 ขั้นตอนการออกแบบ โครงสร้างของระบบ	30
3.2.1 ยูสเคส ไดอะแกรม	30
- แผนภาพยูสเคส ไดอะแกรม	31
3.2.2 สเตท ไดอะแกรม	32
- แผนภาพสเตท ไดอะแกรม	33
3.2.3 คลาส ไดอะแกรม	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การเก็บข้อมูลลงฐานข้อมูล	36
3.2.5 ตัวอย่าง Interface โทรศัพท์มือถือ	37
- ตัวอย่างหน้าจอเมนูหลักของโทรศัพท์มือถือ	37
- ตัวอย่างหน้าจอเมนูอาหารของโทรศัพท์มือถือ	38
บทที่ 4 ผลการทดลอง	39
4.1 ลักษณะของระบบการสั่งอาหารผ่านมือถือ	39
4.1.1 การสั่งอาหารบนมือถือ	39
4.1.2 การแสดงผลข้อมูลที่ร้าน	50
บทที่ 5 บทวิจารณ์และสรุป	58
5.1 บทวิจารณ์และสรุป	58
5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางในการแก้ปัญหา	58
5.3 ข้อจำกัดของระบบ	58
5.4 แนวทางการพัฒนา	58
บรรณานุกรม	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 2.1 การใช้งาน jsp	7
รูปที่ 2.2 แสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีใน CDC และ CLDC	9
รูปที่ 2.3 วงจรการทำงานของ MIDlet	13
รูปที่ 2.4 อินเทอร์เน็ตใน GCF	15
รูปที่ 2.5 สถาปัตยกรรมของ MySQL	19
รูปที่ 2.6 การทำงานของ JDBC	23
รูปที่ 2.7 Two – Tiers Model	24
รูปที่ 2.8 Three – Tiers Model	25
รูปที่ 2.9 ระดับการเชื่อมต่อของ JDBC API	25
รูปที่ 2.10 JDBC driver ประเภทที่ 3 และ 4 ที่ใช้ pure Java JDBC driver	26
รูปที่ 2.11 JDBC driver ประเภทที่ 1 และ 2	27
รูปที่ 3.1 ยูสเคส ไคอะแกรม	30
รูปที่ 3.2 สเตท ไคอะแกรม	32
รูปที่ 3.3 คลาส ไคอะแกรม	35
รูปที่ 3.4 ตัวอย่างหน้าจอเมนูหลักของโทรศัพท์มือถือ	37
รูปที่ 3.5 ตัวอย่างหน้าจอเมนูอาหารของโทรศัพท์มือถือ	38
รูปที่ 4.1 หน้าแรก	39
รูปที่ 4.2 หน้าจอเมนูหลัก	40
รูปที่ 4.3 การพิมพ์ภาษา	41
รูปที่ 4.4 การพิมพ์ตัวเลข	41
รูปที่ 4.5 Register complete	42
รูปที่ 4.6 หน้าจอ Information	43
รูปที่ 4.7 หน้าจอเมนูอาหาร	44
รูปที่ 4.8 หน้าจอเมนูอาหาร(ต่อ)	44
รูปที่ 4.9 หน้าจอแสดงรูปอาหาร	45
รูปที่ 4.10 หน้าจอราคาอาหาร	46
รูปที่ 4.11 หน้าจอระบุการส่งอาหาร	47
รูปที่ 4.12 หน้าจอกรอก id, password	48
รูปที่ 4.13 หน้าจอระบุเวลา	48
รูปที่ 4.14 แสดงข้อมูลการสั่งแบบ Delivery	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 แสดงข้อมูลการสั่งแบบจองโต๊ะ	49
รูปที่ 4.16 หน้าจอหลัก	50
รูปที่ 4.17 หน้าจอแสดงรายการสั่งอาหารทั้งหมด	51
รูปที่ 4.18 หน้าจอแสดงรายการสั่งอาหารแบบ Delivery	52
รูปที่ 4.19 หน้าจอแสดงรายการสั่งอาหารแบบ จองโต๊ะ	53
รูปที่ 4.20 หน้าจอแสดงใบเสร็จจตุกคำ	54
รูปที่ 4.21 รายงานการสั่งอาหารแบบ Delivery	55
รูปที่ 4.22 รายงานการสั่งอาหารแบบ จองโต๊ะ	56
รูปที่ 4.23 รายงานการใบเสร็จจตุกคำ	57



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันโทรศัพท์มือถือได้มีการใช้งานอย่างแพร่หลายและมีแนวโน้มที่จะเพิ่มขึ้นอย่างรวดเร็ว จึงเกิดการพัฒนาศามารถของโทรศัพท์มือถือทางด้านต่างๆมากมาย เพื่อความสะดวกสบายของผู้ใช้โทรศัพท์มือถือ หนึ่งในนั้นคือการพัฒนาด้าน Mobile Application

Food Order Application On Mobile เป็นการพัฒนา Mobile Application ในการสั่งและจองอาหารผ่านทางมือถือ ซึ่งก่อให้เกิดความสะดวกสบายแก่ผู้ใช้งาน สามารถที่จะดำเนินการสั่งอาหารได้เองโดยไม่ต้องติดต่อผ่านโอเปอเรเตอร์ ของร้านอาหารและลูกค้ายังสามารถเห็นรายละเอียดเมนูอาหารได้ ส่วนการจองอาหารก็สามารถสั่งจองอาหารและกำหนดเวลาที่จะไปถึงร้านได้ เมื่อไปถึงก็สามารถรับประทานอาหารหรือรับอาหารกลับไปได้ทันที อีกทั้งยังแสดงราคาสุทธิของอาหารที่สั่งไปทั้งหมด ในแง่ของเจ้าของธุรกิจยังสามารถลดต้นทุนในการจ้างพนักงาน ในส่วนของโอเปอเรเตอร์เพื่อให้พนักงานสามารถทำงานในส่วนที่จำเป็นได้อย่างมีประสิทธิภาพ กล่าวได้ว่าระบบ Food Order Application On Mobile เป็นระบบที่อำนวยความสะดวกแก่ชีวิตคนเมืองในปัจจุบันได้เป็นอย่างดี

1.2 วัตถุประสงค์

วัตถุประสงค์ของปัญหาพิเศษนี้คือ การพัฒนาระบบสั่งอาหารผ่านทางโทรศัพท์มือถือให้สามารถสั่งอาหารได้โดยเห็นข้อมูลอาหารทั้งหมดช่วยในการตัดสินใจของผู้ใช้โทรศัพท์มือถือ และช่วยอำนวยความสะดวกแก่ผู้ใช้โทรศัพท์มือถืออีกด้วย

1.3 ขอบเขตของปัญหา

ขอบเขตของระบบ Food Order Application On Mobile แบ่งได้เป็น 4 ส่วน

1. ส่วน Application ในการสั่งอาหารบนมือถือ โดยมีรายการอาหารให้เลือกซึ่งอาศัยการ download / upload ข้อมูลผ่าน Protocol HTTP
2. ส่วน Server ติดต่อ Database ให้บริการสำหรับ Client หรือเครื่องโทรศัพท์มือถือและ Application ที่ร้าน download / upload ข้อมูล ต่างๆ
3. ส่วน Application ที่ร้านซึ่งจะแสดงผล ข้อมูลลูกค้า รายการอาหารที่สั่ง
4. ส่วน Database เป็นฐานข้อมูลเก็บรายการอาหาร ข้อมูลลูกค้า Server สามารถจัดการฐานข้อมูลได้

1.4 ขั้นตอนในการดำเนินงาน

1. ศึกษาการใช้งานโปรแกรม J2ME
2. ศึกษาการใช้งาน Tomcat
3. จัดทำ Application บนโทรศัพท์มือถือโดยใช้ โปรแกรม J2ME
4. จัดทำ Server
5. จัดทำ Data Base เป็นฐานข้อมูลของรายการอาหาร
6. ใช้ Emulator ทดสอบระบบและแก้ไขข้อผิดพลาดที่เกิดขึ้น
7. Download และใช้งานได้จริง
8. จัดทำรายงานและเตรียมการนำเสนอ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบในการให้บริการบนโทรศัพท์มือถือเพื่อสั่งหรือจองอาหาร
2. ได้รับความรู้ในการพัฒนา Mobile Application โดยใช้ J2ME
3. ผู้ใช้โทรศัพท์มือถือที่ใช้ Food Order Application On Mobile ได้รับความสะดวก รวดเร็วในการสั่งอาหารในช่วงโมงเร่งรีบ
4. เจ้าของธุรกิจสามารถลดต้นทุนในการจ้างพนักงาน

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

ฮาร์ดแวร์ที่ใช้ในการทำปัญหาพิเศษ

1. ซีพียู ความเร็ว 1.2 GHz ขึ้นไป
2. ขนาดหน่วยความจำ 128 MB ขึ้นไป
3. ฮาร์ดดิสก์
4. โทรศัพท์มือถือ Sony Ericsson K700 I

ซอฟต์แวร์ที่ใช้ในการทำปัญหาพิเศษ

1. Jakarta Tomcat
2. Java Development Kit (JDK)
3. J2ME
4. Java 2 Micro Edition Wireless Toolkit
5. My SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เทคโนโลยีที่ใช้สำหรับแอปพลิเคชันบนมือถือ

2.1.1 Java Server Pages (JSP)

Java Server Pages (JSP) เป็น web-scripting เทคโนโลยีคล้ายกับ Netscape server-side JavaScript (SSJS) หรือ Microsoft Active Server Pages (ASP) แต่ผิดกันตรงที่หัวใจของ JSP คือ Java ซึ่งเป็นภาษาที่คอนเซ็ปต์หลักอยู่ที่ออบเจกต์ (object-oriented style) ซึ่งช่วยทำให้ง่ายต่อการพัฒนาในโปรเจกต์ใหญ่ ๆ ตลอดจนสามารถนำส่วนประกอบต่าง ๆ กลับมาใช้ได้อีก (software reusable) จุดเด่นที่สำคัญของ JSP คือสามารถทำงานได้โดยไม่ขึ้นอยู่กับผู้ผลิตซอฟต์แวร์รายใดรายหนึ่งโดยเฉพาะ ซึ่งโดยทั่วไปเทคโนโลยีต่าง ๆ มักจะออกมาในลักษณะของผลิตภัณฑ์จากบริษัทผู้ผลิตแห่งใดแห่งหนึ่ง แต่ JSP ใช้ลักษณะของ specification ซึ่งกำหนดโดย Sun Microsystems ดังนั้นผู้ผลิตซอฟต์แวร์จึงสามารถอ้างอิง specification ที่กำหนดขึ้น ผลิต JSP Container (ตัวที่ใช้ในการรัน JSP) ขึ้นมาใช้กับแพลตฟอร์มใดก็ได้

JSP Containers

JSP Pages (ไฟล์ที่เขียนขึ้นโดยใช้ JSP script และลงท้ายด้วย .jsp) จะถูก run โดย JSP Container ซึ่งมักจะเป็นส่วนประกอบที่อยู่ใน Webserver หรือเป็นตัวแอดฮอนใน Application Server. โดยทั่วไป JSP Container จะเป็นตัวรับ request จาก client ส่งผ่านไปยัง JSP Page และส่งค่าที่ได้จากการประมวลผลโดย JSP Page กลับไปยัง client. JSP Container ที่ใช้กันอยู่มีมาจากหลายค่าย ยกตัวอย่างเช่น GNU JSP, Expresso, Tomcat Jakarta, Resin, Weblogic เป็นต้น JSP Container ที่เราจะใช้กันคือ Jasper ซึ่งเป็น JSP Container ที่อยู่ใน Tomcat Servlet Engine จากค่าย apache (www.apache.org) ซึ่งทาง Sun ใช้เป็นตัวอ้างอิงในวงการ JSP โดยในอนาคต Tomcat จะเป็นตัวที่ใช้ในการรัน Servlet แทน Jserv ซึ่งใช้กันอยู่ในปัจจุบัน

การติดตั้ง Tomcat

1. Java Development Kit

โปรแกรม Tomcat ถูกเขียนขึ้นโดยใช้ Java ซึ่งจะ Run กับ Java Development Kit (JDK) เวอร์ชัน 1.1 และ 1.2 ดังนั้นก่อนจะ Run เราจะต้องติดตั้ง JDK1.xxx ตามแพลตฟอร์มที่เราใช้เสียก่อน สำหรับ Window และ Unix ให้ไปดาวน์โหลดที่ <http://java.sun.com/jdk/>. (บางครั้ง JDK อาจถูกเรียกว่า SDK) สำหรับ Linux ให้ไปที่ <http://www.blackdown.org/>. สมมุติว่าเราใช้ Window และติดตั้ง JDK1.2.2 ไว้ที่ D:\jdk1.2.2 ให้เซต PATH=d:\jdk1.2.2\bin\ โดยใช้คำสั่งใน Dos (set PATH=d:\jdk1.2.2\bin;) หรือ คลิกขวาที่ My Computer แล้วเลือกที่ Environment แท็บ ที่เราต้องเซต PATH เพราะ Tomcat จะรันโดยใช้คำสั่ง java.exe ซึ่งอยู่ใน bin ไดรректорรี่ของ JDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Jakarta Tomcat

ทาง apache ยังคงทำการพัฒนา Tomcat ให้มีประสิทธิภาพมากขึ้นเพื่อใช้เป็นมาตรฐานในอนาคต ดังนั้นถ้าเข้าไปที่เว็บไซต์ในส่วนของการ Download เราจะเห็นตัวเลือกในการ Download ได้คร่าว ๆ เป็นสองแบบ คือ แบบ sourcefiles ที่ต้องมา Compile เองโดยใช้เครื่องมือช่วยที่เรียกว่า Ant ซึ่งเป็นส่วนที่นักพัฒนามิชอบ Download กัน กับแบบที่ Compile แล้ว (binary) พร้อมทั้งจะรันซึ่งเหมาะสำหรับคนที่อยากนำ Tomcat มาใช้งานอย่างเดียว และเป็นอย่างที่เราจะใช้ ดังนั้น ให้ไปที่ <http://jakarta.apache.org/downloads/binindex.html> แล้วทำการติดตั้งสมมุติว่าเราใส่ Tomcat ไว้ที่ D:\jakarta-tomcat ให้เปิด Command Prompt (MSDos) จาก Application Menu หรือใช้ start->run->type cmd แล้วไปที่ไดเรกทอรีที่เป็น bin ซึ่งในตัวอย่างก็คือ D:\jakarta-tomcat\bin แล้ว Run โดยใช้คำสั่ง D:\jakarta-tomcat\bin> startup หลังจากนั้นให้เปิด browser เพื่อเช็คว่า Tomcat ทำงานถูกต้องหรือเปล่า โดยไปที่ <http://127.0.0.1:8080/examples/jsp/Note>: <http://127.0.0.1:8080> คือ ให้ browser ส่ง request ไปที่เครื่องตัวเอง (localhost) ที่พอร์ต 8080 (Tomcat default port) ถ้าจะปิด Tomcat ก็ให้พิมพ์ D:\jakarta-tomcat\bin> shutdown ลงไปแทน

Web Applications

ก่อนที่เราจะเริ่มเขียน JSP สิ่งหนึ่งที่เราควรทำความเข้าใจคือโครงสร้างของไดเรกทอรีต่าง ๆ สำหรับไฟล์ของเราใน Tomcat ในสมัยก่อนเราไม่ค่อยได้ให้ความสนใจกับการจัดเรียงไฟล์ต่าง ๆ ซักเท่าไร ข้อเสียที่เห็นได้ชัดคือการยากในการจัดเก็บ และการยากในการย้ายไฟล์จาก Server หนึ่ง ไปยังอีก Server หนึ่ง เพราะโดยทั่วไปแต่ละ webserver จะมีลักษณะการจัดเก็บไฟล์ไม่เหมือนกัน แต่เมื่อ Java Servlet Specification เวอร์ชัน 2.2 ออกมา Server ที่ใช้ Run Servlet v.2.2 จะถูกบังคับให้มีข้อกำหนดสนับสนุนการจัดเก็บไฟล์แบบหนึ่ง ที่เรียกว่า Web - Application

Web application คือกลุ่มของไดเรกทอรีและไฟล์ที่อาจจะประกอบด้วย html, jsp, servlet, javabean และอื่นๆ ซึ่งอยู่รวมกันในลักษณะของระบบไฟล์ (file system) หรือถูกอัดอยู่ในไฟล์เดียวกันโดยจะเรียกว่า Web Archive (.war) ไฟล์ ซึ่งมีประโยชน์ในการโยกย้ายและติดตั้งจาก Server หนึ่ง ไปยังอีกเซฟ Server หนึ่ง Tomcat เป็น Servlet Engine ที่สนับสนุน Java Servlet API v.2.2 ดังนั้นไฟล์ JSP ที่เราเขียน จะต้องถูกจัดอยู่ในรูปแบบ Web- Application นี้ด้วย

ถ้าใครเคยคลุกคลีกับ webserver จะเห็นว่ามีการใช้ไดเรกทอรีในการจัดกลุ่มของไฟล์ไว้ด้วยกันตามหน้าที่ ซึ่งทาง Tomcat ก็ใช้หลักการเดียวกันในการจัดการ web application หลักการก็คือ แต่ละ web application จะถูกจัดให้อยู่ในไดเรกทอรีของตัวเอง ที่เรียกว่า context path ยกตัวอย่างเช่น ถ้า web application ของเราถูกแมปกับ context path ที่ชื่อ webapplication1 เวลาดูจาก URL ก็จะเป็นเป็น <http://www.mycompany.com/webapplication1/> เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างที่กล่าวมาข้างต้นว่า web application คือกลุ่มของไคลเอนต์และไฟล์ที่ถูกจัดอยู่รวมกันเป็นชั้น ๆ ดังนั้นถ้าเรามีไฟล์ที่ชื่อ helloworld.html :ซึ่งอยู่บนสุด (top-level) ของไคลเอนต์ใน web application เราจะสามารถเรียกออกมาดูได้โดยใช้ URL เท่ากับ `http://www.mycompany.com/helloworld.html`

ในกรณีที่ไฟล์ไม่ได้อยู่บนสุดของไคลเอนต์ใน web application เราก็ยังสามารถเรียกออกมาได้โดยอ้างอิงไปที่ไคลเอนต์ย่อยนั้น ยกตัวอย่างเช่น ถ้ากลุ่มไฟล์ JSP ถูกเก็บไว้ในไคลเอนต์ถัดลงมาที่ชื่อ *.jsp เราสามารถที่จะเรียกไฟล์มาดูโดยใช้ URL เท่ากับ `http://www.mycompany.com/ /jsp/xxx.jsp`

ส่วนประกอบของ Web Application

web application หนึ่ง ๆ อาจประกอบด้วย

- 1) Servlet
- 2) JavaServer Pages
- 3) Utility Classes
- 4) ไฟล์ปกติ เช่น HTML, images, sounds
- 5) JavaBean

แต่ละอย่างจะถูกเก็บไว้ในไคลเอนต์ต่อไปนี้

1. root directory หรือส่วนบนสุด : ไฟล์ *.html, *.jsp มักจะถูกเก็บไว้ส่วนบนสุดของ web application แต่อาจจะมีไคลเอนต์ย่อยลงมาจาก root เพื่อเก็บ ไฟล์ images ก็ได้

2. WEB-INF/web.xml : ไฟล์ web.xml เป็นส่วนที่ใช้ในการบรรยายส่วนประกอบต่าง ๆ (Web Application Deployment Descriptor) รวมไปถึงการป้อนค่าเริ่มต้นให้กับโปรแกรมต่าง ๆ ที่บรรจุอยู่ใน web application นี้ยกตัวอย่างเช่น เราสามารถป้อนค่าเริ่มต้นให้กับ Servlet ได้โดยใส่ค่าเข้าไปใน XML Tag ตรงส่วนที่เรียกว่า <init-param> เป็นต้น

3. WEB-INF/classes/* : เป็นส่วนที่ใช้ในการเก็บไฟล์ *.class ทั้งหมดที่ใช้ใน web application ซึ่งอาจจะเป็นทั้ง Servlet หรือ ไฟล์ class ธรรมดาที่ไม่ได้อยู่ในรูปของ .jar ก็ได้ ยกตัวอย่างเช่น ถ้าไฟล์ของเราอยู่ใน package ชื่อ com.mycompany.mypackage.MyServlet ไฟล์นี้ก็จะถูกเก็บอยู่ในไคลเอนต์ WEB-INF/classes/com/mycom

pany/mypackage/MyServlet.class

4. WEB-INF/lib/*.jar : ไคลเอนต์นี้ใช้เก็บไฟล์ .jar ที่เราต้องการใช้สำหรับ web application ซึ่งอาจจะเป็นส่วนที่ได้มาจาก third party หรือเป็นพวก Database driver ก็ได้ ประโยชน์อย่างหนึ่งที่เราเห็นได้ในการเก็บไลบรารีต่างๆ ไว้ที่ไคลเอนต์นี้คือ การง่ายต่อการติดตั้ง เหตุผลคือหลังจาก web application ของเราถูกใส่เข้าไปใน server แล้ว ตัว server จะโหลด Load ไฟล์ที่อยู่ในไคลเอนต์นี้โดยอัตโนมัติ เราจึงไม่ต้องทำการเซต CLASSPATH ให้ยุ่งยากตัวอย่างง่าย ๆ ของโครงสร้าง web application อาจเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/index.html
 /howto.jsp
 /feedback.jsp
 /images/banner.gif
 /images/jumping.gif
 /WEB-INF/web.xml
 /WEB-INF/lib/jspbean.jar
 /WEB-INF/classes/com/mycompany/servlets/MyServlet.class
 /WEB-INF/classes/com/mycompany/util/MyUtil.class

Integration with Tomcat

การติดตั้ง web application สามารถทำได้ 3 วิธีคือ

1. นำ web application ที่อยู่ในรูปของกลุ่มไครเรททอรี่ (ยังไม่ได้ถูกบีบอัด) ไปใส่ไว้ที่ \$TOMCAT_HOME/webapps/. กรณีนี้ Tomcat จะจัด context path ให้ web application เองโดยอิงจากชื่อ subdirectory ที่บรรจุกลุ่มไครเรททอรี่ของ web application ไว้ ยกตัวอย่างเช่น เราใส่ web application ไว้ที่ \$TOMCAT_HOME/webapps/myapp/ ดังนั้น context path ที่จะใช้ติดต่อกับ web application นี้ก็คือ myapp วิธีนี้เป็นวิธีที่ง่ายที่สุด และเหมาะสำหรับการพัฒนา ก่อนนำไปใช้จริง

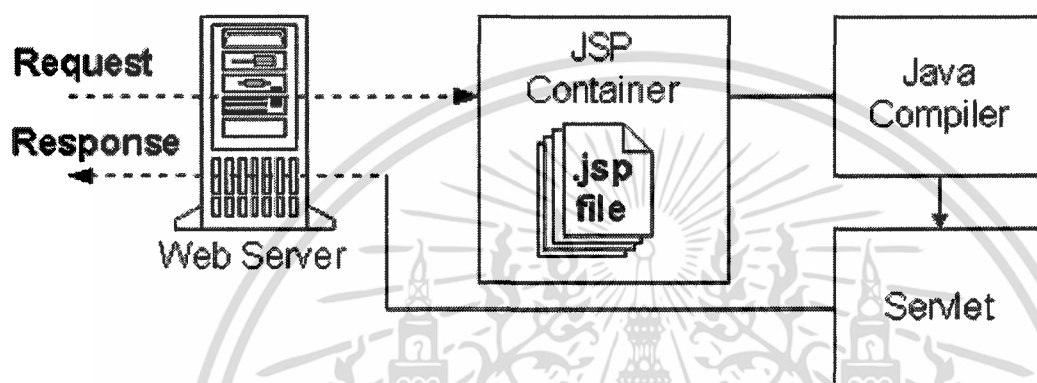
2. นำ web application archive (.war ไฟล์) ไปใส่ไว้ที่ \$TOMCAT_HOME/webapps/. วิธีนี้เหมาะสำหรับการติดตั้ง web application ใน production server ที่ใช้งานจริง โดย Tomcat จะแกะและขยายไฟล์ทั้งหมดที่อยู่ใน .war ไฟล์ออกมาแล้วทำการโหลดเข้าหน่วยความจำ

3. ในกรณีที่ไครเรททอรี่ที่เราพัฒนา web application ไม่ได้อยู่ที่ \$TOMCAT_HOME/webapps/ เราก็สามารถบอก Tomcat ให้โหลด web application นี้ได้โดยเข้าไปแก้ที่ \$TOMCAT_HOME/conf/server.xml ในส่วนของ <Context> ให้ชี้ไปที่ไครเรททอรี่ที่เราเก็บ webapplication ไว้ เช่น ถ้าเราเก็บ web application ไว้ที่ D:\MyWebApplication\ แล้วอยากให้ MyWebApp เป็น context path ในการชี้ไปที่ web application ของเราก็แค่ Add<Contextpath="/MyWebApp"docBase="D:\MyWebApp" debug="0"reloadable="true"/> ไว้ที่บรรทัดถัดมาจาก Context path อันสุดท้ายที่มีอยู่แล้ว เป็นต้น (สมมุติว่าเราติดตั้ง Tomcat ไว้ที่ ไครเรททอรี่ D:\jakarta-tomcat ดังนั้น \$TOMCAT_HOME จะเป็น D:\jakarta-tomcat)

การทำงานของ JSP

หลังจาก request จาก client ส่งมาถึงเซฟเวอร์, JSP Container จะทำการแปลงไฟล์ JSP ให้กลายเป็น Servlet source ไฟล์ (ในกรณีที่ JSP ไฟล์ดังกล่าวถูกเรียกเป็นครั้งแรก) ซึ่งไฟล์ Servlet source ที่ได้จะถูก Compile เป็น .class เพื่อใช้ในการประมวลผล request ของ client แล้วส่งกลับไปให้ Server ในรูปของ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

outputStream ซึ่งถูกส่งไปที่ client ในท้ายสุด หลังจากนั้น ถ้า JSP ไฟล์ดังกล่าวถูกเรียกอีก การแปลงไฟล์ หรือ Compile จะไม่เกิดขึ้น เพราะ JSP Container จะใช้ไฟล์ .class ที่เก็บไว้แล้วประมวลผลแทน ข้อสังเกต คือ การเรียกไฟล์ JSP ครั้งแรกจะรู้สึกช้า แต่ครั้งถัดไป ๆ จะเห็นได้ว่าเร็วขึ้นมาก เพราะได้มีการลด ขั้นตอนต่างๆที่เสียเวลาไป ใน JSP Specification 1.1 ยังมีการแนะนำให้ JSP Engine รองรับการโหลดไฟล์ JSP ขึ้นมา Compile ใหม่แบบอัตโนมัติ (Reloading) ซึ่งมักจะเกิดขึ้นในกรณีที่ JSP ไฟล์ได้มีแก้ไข แล้ว จะต้องมีการ Compile ไฟล์ Servlet ที่กำลังใช้งานอยู่ใหม่



รูปที่ 2.1 การใช้งาน jsp

จากข้อความที่อธิบายข้างต้น เราสามารถแบ่งระยะเวลาของไฟล์ JSP หนึ่งๆ ออกเป็นสองช่วงคือ

1. Translation Time คือช่วงเวลาที่ JSP ไฟล์ถูกแปลงให้กลายเป็น Servlet ไฟล์และถูก Compile ให้กลายเป็น .class ไฟล์ ซึ่งจะเกิดก่อนการรับ request จาก client ครั้งแรก
2. Client Request Time คือช่วงเวลาที่ .class ของ JSP ไฟล์ ทำการรับ request จากแต่ละ client แล้วทำการประมวลผล

2.1.2 J2ME

J2ME เป็นแพลตฟอร์ม Java ที่ออกแบบมาเพื่อใช้งานกับ Application ที่ทำงานบนอุปกรณ์ขนาดเล็ก เช่น โทรศัพท์เคลื่อนที่ พีดีเอ โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต โทรศัพท์ดิจิทัลขนาดเล็ก อุปกรณ์บันทึกและระบบนำทางในรถยนต์ สวิตช์ในระบบเครือข่าย คอมพิวเตอร์ของเครื่องอำนวยความสะดวกภายในบ้าน ฯลฯ

J2ME ได้นำโครงสร้างแบบ Module ที่มีความยืดหยุ่นสูงเข้ามาใช้ เพื่อให้สามารถสนับสนุนการทำงานอุปกรณ์หลากหลายประเภท J2ME กำหนดชิ้นการทำงานของ Software ไว้ 3 เลเยอร์ด้วยกัน โดยเลเยอร์ทั้งหมดจะอยู่เหนือชั้นระบบปฏิบัติการของอุปกรณ์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลขอร์ Java Virtual Machine เป็นเลขอร์ของ Java Virtual Machine ปรับแต่งให้เข้ากับระบบปฏิบัติการของอุปกรณ์ และรองรับแต่ละ configuration ของ J2ME จากรูปที่ 2.1 Virtual Machine ของ J2ME ได้แก่ CVM และ KVM
- เลขอร์ Configuration เป็นเลขอร์ของคอนฟิกูเรชันของ J2ME ซึ่งกำหนดคลาสไลบรารี อุปกรณ์ใช้งานทั่วไปหรือกลุ่มอุปกรณ์ที่มีความต้องการหน่วยความจำและหน่วยประมวลผลใกล้เคียงกัน จากรูปที่ 2.1 Configuration ใน J2ME มี 2 ประเภทด้วยกัน คือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration)
- เลขอร์ Profile เป็นเลขอร์ที่สร้างเหนือเลขอร์ Configuration โดยกำหนดคลาสไลบรารี เพื่อสนองตอบความต้องการของตลาดเฉพาะกลุ่ม จากรูปที่ 2.1 ตัวอย่างของ Profile ในเลขอร์นี้ได้แก่ PDAP, MIDP, Foundation Profile และ Personal Profile

Configuration ใน J2ME

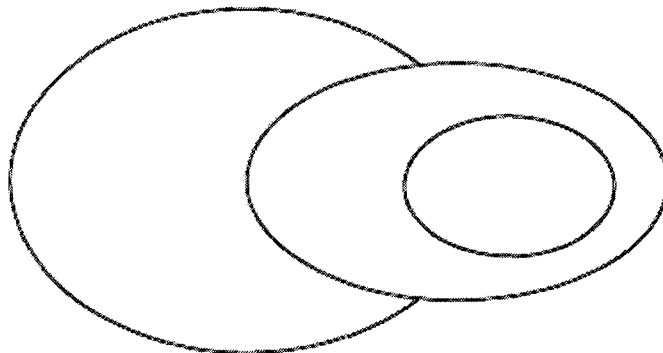
Configuration และ Profile เป็นองค์ประกอบหลักของ J2ME โดยมีจุดสำคัญ คือ เพื่อปรับแต่ง Virtual Machine และคลาสไลบรารีให้เหมาะสมกับอุปกรณ์แต่ละประเภท Configuration คือ ชุดคำสั่งที่มีคุณสมบัติขั้นต่ำของ Java Virtual Machine และจาวาคลาสสำหรับอุปกรณ์แต่ละประเภท เป็นตัวแทนของอุปกรณ์ที่ลักษณะคล้ายคลึงกัน อาจกล่าวได้ว่า Configuration เป็นตัวกำหนดคุณสมบัติหรือไลบรารีกำหนด Configuration จะมีในอุปกรณ์ที่อยู่ในกลุ่มเดียวกัน

ปัจจุบัน Configuration ใน J2ME แบ่งเป็น 2 ประเภท คือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration) โดยมีเป้าหมายที่กลุ่มอุปกรณ์ 2 ประเภทซึ่งมีความต้องการหน่วยความจำและหน่วยประมวลผลพื้นฐานใกล้เคียงกัน

- CDC เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่น ติดตั้งตายตัว และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2-16 เม็กกะไบต์ ใช้หน่วยประมวลผลแบบ 32 บิต หรือมากกว่า เชื่อมต่อเครือข่ายที่มีแบนด์วิดธ์สูงอย่างต่อเนื่อง โดยอาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์มือถือขนาดเล็ก อินเทอร์เน็ตทีวี โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง อุปกรณ์บันทึกและระบบนำทางในรถยนต์

- CLDC เป็นอุปกรณ์ส่วนบุคคล พกพาได้ และใช้เชื่อมต่อข้อมูล โดยปรกติมักมีส่วนติดต่อกับผู้ใช้แบบง่ายๆ (เมื่อเปรียบเทียบกับระบบบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะ) มีหน่วยความจำประมาณ 128 กิโลไบต์ – 1 เม็กกะไบต์ ใช้หน่วยประมวลผลแบบ 16 หรือ 32 บิต เชื่อมต่อกับเครือข่ายที่มีแบนด์วิดธ์ต่ำเป็นระยะเวลานานๆ โดยไม่อาศัยพอร์ต TCP/IP ตัวอย่างอุปกรณ์ประเภทนี้ ได้แก่ โทรศัพท์เคลื่อนที่แบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องปาล์ม ไอเอสแบบพกพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีใน CDC และ CLDC

ในชั้นของ Configuration มีคลาส 2 ประเภทด้วยกัน คือ คลาสที่นำมาจาก J2SE และที่ออกแบบเฉพาะอุปกรณ์ขนาดเล็ก คลาสที่นำมาจาก J2SE จะมีคุณสมบัติอย่างเดียวกันกับคลาสใน J2SE หรือ Subclass ของ J2SE เช่น แพ็กเกจ java.lang java.io และ java.util จะแสดงความสัมพันธ์ระหว่าง J2SE และคลาสไลบรารีของ CDC และ CLDC จากภาพจะเห็นว่าคลาสใน CLDC ส่วนใหญ่ทำงานร่วมกับคลาส CDC ได้ดีเช่นเดียวกับกรณีของคลาสใน CLDC

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

J2ME สำหรับอุปกรณ์ไร้สาย

J2ME ได้ให้กำเนิดแอปพลิเคชันยุคใหม่บนอุปกรณ์ไร้สาย ช่วยให้เกมแบบหลายผู้เล่นที่ทำงานผ่านอินเทอร์เน็ต การทำธุรกรรมทางการเงินบนโทรศัพท์มือถือ แอปพลิเคชันสำหรับองค์กรทั้งไคลเอ็นท์และเซิร์ฟเวอร์ เกิดขึ้นได้บนโทรศัพท์เคลื่อนที่ และเพจเจอร์รับส่งข้อความ MIDP CLDC และ KVM ได้กลายมาเป็นรากฐานในการพัฒนาแอปพลิเคชันบนอุปกรณ์ไร้สายยุคใหม่

สามารถแบ่งโครงสร้างออกเป็นเลเยอร์ต่างๆ จากล่างขึ้นบน ดังนี้

- เลเยอร์ฮาร์ดแวร์ MID หมายถึง ตัวโทรศัพท์เคลื่อนที่ (เช่น Nokia6600) หรือเพจเจอร์รับส่ง ข้อความ (เช่น RIM รุ่น Blackberry950)
- เลเยอร์ซอฟต์แวร์ของระบบที่ติดตั้งมากับอุปกรณ์ หมายถึง ระบบปฏิบัติการและไลบรารีของระบบที่บริษัทผู้ผลิตให้มา
- เลเยอร์ KVM เป็นส่วนที่เตรียม runtime environment ไว้ให้แอปพลิเคชันบนอุปกรณ์ไร้สาย
- เลเยอร์ CLDC เป็นส่วนที่เตรียม API หลักของจาวาให้แอปพลิเคชันบนอุปกรณ์ไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลขอร์ MIDP (Mobile Information Device Profile) เป็นส่วนที่เตรียมไลบรารีสำหรับส่วนติดต่อกราฟฟิกับผู้ใช้ (User Interface) และหน่วยเก็บข้อมูล persistent storage ระบบเครือข่าย และไทม์เมอร์

นอกจากคลาสไลบรารีสำหรับ MIDP อาจเตรียมคลาสไลบรารีเฉพาะอุปกรณ์ไว้ให้นักพัฒนา เพื่อตีความสามารถของฟังก์ชันที่มีอยู่แล้วไปใช้งานให้เกิดประโยชน์สูงสุด อาทิ การโทรศัพท์ การแชร์ข้อมูลกับแอปพลิเคชันที่ติดตั้งมาในเครื่อง (เช่น ปฏิทิน สมุดจดที่อยู่) การตรวจสอบข้อมูลอุปกรณ์ที่ผู้ผลิตเตรียมไว้มาใช้งาน จะช่วยเพิ่มความสามารถแก่แอปพลิเคชันบนอุปกรณ์ไร้สาย แต่ไม่สามารถเคลื่อนย้ายไปสู่อุปกรณ์อื่นที่ใช้ MIDP ได้ เนื่องจากคลาสที่นำมาใช้อยู่นอกเหนือขอบเขตของ MIDP

ความต้องการของระบบ

อุปกรณ์ไร้สายจะทำงานสนับสนุน J2ME ได้ดีเมื่อมีคุณสมบัติตรงตามข้อกำหนด หากต้องการให้ KVM ทำงานได้อย่างมีประสิทธิภาพ โดยใช้ไลบรารี CLDC จะต้องมีความสัมพันธ์ของระบบขั้นต่ำ ดังนี้

- มีหน่วยความจำขนาด 160 – 512 กิโลไบต์ สำหรับสร้างแพลตฟอร์มจาวา
- มีหน่วยประมวลผลแบบ 16 – 32 บิต ความเร็ว 25 เมกกะเฮิร์ตซ์
- ใช้พลังงานน้อย โดยมากมักทำงานโดยใช้แบตเตอรี่
- เชื่อมต่อกับเครือข่ายได้ในช่วงสั้นๆ (อาศัยระบบไร้สายเป็นส่วนใหญ่) แบนด์วิดท์ค่อนข้างจำกัด (ความเร็ว 9600 ไบต์ต่อวินาทีหรือน้อยกว่า)
- มีหน่วยความจำชั่วคราวขนาด 32 กิโลไบต์ สำหรับเก็บจาวา รันไทม์หน่วยความจำออบเจกต์

2.1.3 หลักการพัฒนาแอปพลิเคชันบน MIDP ด้วย J2ME

ไลบรารีของ CLDC

ไลบรารีของ CLDC เป็นไลบรารีของเครือข่ายและระบบขั้นสูงที่ไม่จำกัดเฉพาะอุปกรณ์ใดอุปกรณ์หนึ่ง ประกอบด้วยคลาส 2 ประเภท คือ คลาสที่เป็นชั้นเซตของ J2SE และคลาสที่เกี่ยวกับกรอบการติดต่อสื่อสารทั่วไป (Generic Connection Framework) ของ CLDC

ชั้นคลาสของ J2SE

คลาสประเภทแรกของ CLDC เป็นชั้นคลาสของไลบรารี J2SE กำหนดไว้ในแพ็คเกจ java.lang, java.io และ java.util โดยคลาสระบบและคลาสชนิดข้อมูลเข้ากันได้กับ J2SE และ J2EE และเพื่อให้เข้ากันได้และเคลื่อนย้ายข้ามแพลตฟอร์มได้โดยสะดวก คลาสประเภทนี้จะใช้ชื่อแพ็คเกจตามชื่อคลาสอย่างเดียวกับใน J2SE หรือตามชื่อชั้นคลาสของคลาสใน J2SE ที่เกี่ยวเนื่องกัน โดยไม่มีการเพิ่มเมธอด public หรือ protected หรือฟิลด์ซึ่งไม่มีในคลาสที่เกี่ยวข้องกันใน J2SE ความหมายของคลาสและเมธอดจึงไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสที่มีเฉพาะใน CLDC

คลาสประเภทที่ 2 ของ CLDC กำหนดไว้ในแพ็คเกจ javax.microedition.io คลาสเหล่านี้มีเฉพาะใน CLDC จึงไม่สามารถทำงานร่วมกับไลบรารีของ J2SE ได้ เป็นคลาสชั้นสูงเกี่ยวกับเรื่องทั่วไปของเครือข่าย กำหนดไว้ในแพ็คเกจ java.io และ java.net คลาสประเภทนี้จะเตรียมกรอบการติดต่อสื่อสารผ่านเครือข่าย ให้กับอุปกรณ์ที่สนับสนุน J2ME โดยกรอบการสื่อสารนี้มักเรียกว่า Generic Connection Framework คลาสส่วนใหญ่ได้แก่อินเตอร์เฟซที่แทนการสื่อสารรูปแบบต่างๆ เช่น ซอกเกต (Socket) ดาต้าแกรม (Datagram) ซีเรียล (Serial) และ HTTP ขึ้นอยู่กับว่าผู้ผลิตอุปกรณ์หรือผู้ให้บริการเครือข่าย จะติดตั้งอินเตอร์เฟซการเชื่อมต่อเหล่านี้เพียงบางตัวหรือครบทั้งหมด เมื่อพิจารณาจากความสามารรถของอุปกรณ์และเครือข่าย

ประเภทของข้อมูลพื้นฐาน

CLDC สนับสนุนเฉพาะขอบเขตของชนิดข้อมูลที่นำมาจาก J2SE ซึ่งได้แก่ byte short int long char และ boolean แต่ไม่สนับสนุนข้อมูลประเภท float และ double (ยกเว้น CLDC 1.1 จะสนับสนุนข้อมูลประเภท float) เนื่องจากสาเหตุ 2 ประการ คือ อุปกรณ์เป้าหมายที่นำ CLDC ไปใช้ ส่วนใหญ่ไม่มีฮาร์ดแวร์รองรับเลขทศนิยม และหากใช้จะใช้ซอฟต์แวร์เข้าช่วยก็จะต้องเสียค่าใช้จ่ายสูงมาก

นอกจากนี้ CLDC ยังกำหนดคลาส type wrapper สำหรับข้อมูลทุกประเภท ได้แก่

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

คลาส type wrapper กำหนดไว้ในจาวา เนื่องจากจาวามีระบบย่อยหลายระบบซึ่งทำงานได้เฉพาะออบเจกต์เท่านั้น ในกรณีนี้ คุณสามารถสร้างออบเจกต์โดยใช้คลาส type wrapper ซึ่งเก็บชนิดของข้อมูลที่นำมาจาก J2SE เอาไว้

ไลบรารีของ MIDP

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

คลาสโปรแกรมจัดการแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

คลาสของส่วนติดต่อกราฟฟิกับผู้ใช้ (GUI Class)

ชุดเครื่องมือ `Abstract Windowing Toolkit` ใน `J2SE` ออกแบบมาเพื่อใช้งานกับแอปพลิเคชันบนเครื่องคอมพิวเตอร์แบบตั้งโต๊ะและไม่สามารถนำมาใช้กับอุปกรณ์ไร้สายได้ เนื่องจากความต้องการทรัพยากรประเภทหน่วยความจำค่อนข้างมาก MIDP มีวิธีการที่ต่างกันในการกำหนดไลบรารีสำหรับแพ็คเกจส่วนติดต่อกราฟฟิกับผู้ใช้ ด้วยการใช้อ API ชั้นสูงซึ่งเน้นความสามารถในการเคลื่อนย้ายข้ามอุปกรณ์ และ API ชั้นพื้นฐานซึ่งเน้นองค์ประกอบทางกราฟฟิเฉพาะอุปกรณ์ และ `input event` ทั่วไป คลาสที่เชื่อมต่อกับส่วนติดต่อกราฟฟิกับผู้ใช้ และ `event-handling` กำหนดไว้ในแพ็คเกจ `java.microedition.lcdui` `Screen` ถือเป็นซูเปอร์คลาสของคอมโพเนนต์ส่วนติดต่อกับผู้ใช้โดยใช้อ API ชั้นสูง ประกอบไปด้วย `Alert` `Form` `List` `Textbox` ฯลฯ

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

คลาสของพื้นที่เก็บข้อมูลแบบคงตัว (Persistent Storage Class)

ในบางครั้งแอปพลิเคชันที่เขียนบน MID จำเป็นจะต้องเก็บข้อมูลไว้บนอุปกรณ์อย่างถาวร คลาสที่กำหนดไว้ในแพ็คเกจ `java.microedition.rms` ให้กลไกการเก็บข้อมูลถาวรที่เรียกว่า `RecordStore` โดยยอมให้แอปพลิเคชันเขียนลบและปรับปรุงบันทึกข้อมูลใหม่ในหน่วยเก็บข้อมูลถาวรของอุปกรณ์

คลาสของเครือข่าย (Network Class)

แม้ว่ากรอบการติดต่อสื่อสารทั่วไปที่กำหนดไว้ใน CLDC จะประกอบไปด้วยชุดของอินเทอร์เน็ตเฟซการเชื่อมต่อเครือข่าย แต่ก็ไม่มีโพรโตคอลอยู่เบื้องหลังอินเทอร์เน็ตเฟซการเชื่อมต่อจริงๆ หากแต่ปล่อยให้ เป็นหน้าที่ของ MIDP

ในบรรดาอินเทอร์เน็ตเฟซการเชื่อมต่อเครือข่ายเหล่านี้ `Httpconnection` ถือเป็นอินเทอร์เน็ตเฟซหลักที่ต้องมีใน MIDP เสมอ คลาสของอินเทอร์เน็ตเฟซเหล่านี้ กำหนดไว้ในแพ็คเกจ `java.microedition.io`

2.1.4 MIDlets

`MIDlet` หมายถึง แอปพลิเคชันบน MIDP `MIDlet` มีส่วนคล้ายกับจาวาแอปเพล็ต แม้จะไม่มีเมธอด `main()` แต่ `MIDlet` ก็นำคลาส `javax.microedition.midlet.MIDlet` ตลอดจนเมธอดทั้ง 3 `startApp()` `pauseApp()`

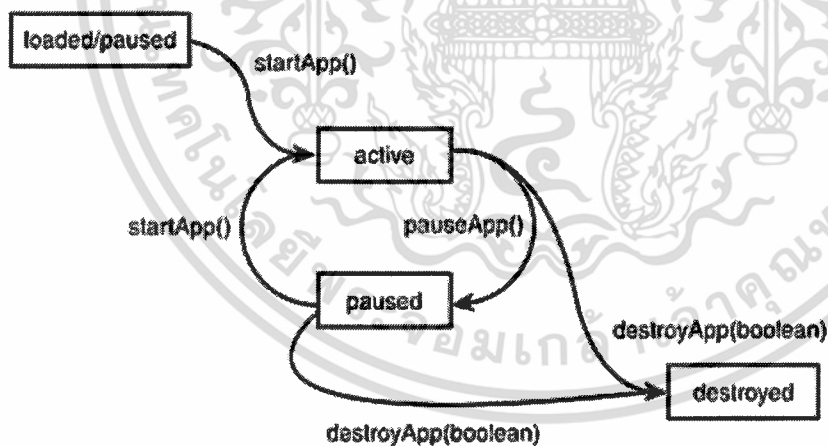
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

destroyApp() เข้ามาใช้ นอกจากนี้ MIDlet ยังกำหนด constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆ อีกด้วย

เราสามารถให้คำนิยามของคลาส javax.microedition.midlet.MIDlet ได้ดังนี้

```
public abstract คลาส MIDlet extends Object {
    protected MIDlet()
    protected abstract void startApp() throws MIDletStateChangeException
    protected abstract void pauseApp()
    protected abstract void destroyApp() (Boolean unconditional)
        throws MIDletStateChangeException
    public final String getAppProperty(String key)
    public final void notifyDestroyed()
    public final void notifyPaused()
    public final String getAppProperty(String key)
    public final void resumeRequest() }
```

คลาส MIDlet ระบุเมธอดที่สามารถเรียกใช้โดยซอฟต์แวร์จัดการแอปพลิเคชัน (AMS แอปพลิเคชัน Management System) เพื่อสั่งให้แอปพลิเคชัน MIDlet เริ่มต้นและหยุดทำงาน



รูปที่ 2-3 วงจรการทำงานของ MIDlet

การกระทำการของ MIDlet ประกอบไปด้วย 4 สถานะ คือ บรรลุ กำลังทำงาน หยุดชั่วคราว และถูกทำลาย อาศัยซอฟต์แวร์จัดการแอปพลิเคชันเป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่ง ควบคุมด้วยเมธอด startApp() pauseApp() และ destroyApp() ที่มาพร้อมกับ MIDlet รูปที่ 2-3 แสดงให้เห็นจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกเมธอดข้างต้น

จากรูปเมื่อ MIDlet ถูกบรรจุลงในอุปกรณ์และ constructor ถูกเรียกแล้ว มันก็จะเข้าสู่สถานะบรรลุ ซึ่งต้องเกิดขึ้นทุกครั้งก่อนที่ซอฟต์แวร์จัดการแอปพลิเคชันจะเริ่มแอปพลิเคชันจะเรียกเมธอดstartApp() ขึ้นมา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ MIDlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน

เมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการให้ MIDlet ทำงานต่อไป ก็จะเรียกเมธอด `pauseApp()` จากนั้น MIDlet จะหยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว คินทรัพยากรที่ดึงมาใช้งานและเข้าสู่ภาวะไม่ทำงาน MIDlet สามารถกลับไปอยู่ที่สถานะกำลังทำงานได้ เมื่อซอฟต์แวร์จัดการแอปพลิเคชันเรียกเมธอด `startApp()` ขึ้นมา ทำยที่สุดเมื่อซอฟต์แวร์จัดการแอปพลิเคชันไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการเคลียร์หน่วยความจำเพื่อให้โปรแกรมอื่นได้ใช้งาน ก็จะส่งสัญญาณเตรียมทำลาย MIDlet ทิ้ง ด้วยการเรียกเมธอด `destroyApp()` และเข้าสู่สถานะถูกทำลายซึ่ง MIDlet จะปล่อยทรัพยากรทั้งหมด ทำการจัดเก็บข้อมูลถาวรต่างๆ และหยุดการทำงานทั้งหมดหาก MIDlet อยู่ระหว่างสังเคราะห์การขึ้นตอนสำคัญอยู่ ก็อาจร้องขอไม่ให้เข้าสู่สถานะถูกทำลายได้โดยเรียกใช้ `MIDletStateChangeException` อย่างไรก็ตาม ซอฟต์แวร์จัดการแอปพลิเคชันอาจปฏิเสธหรือยินยอมตามคำร้องขอนี้ก็ได้ ตัวแปรบูลีน `unconditional` ในลายเซ็นของเมธอด `destroyApp()` จะเป็นตัวกำกับว่าคำร้องขอนี้สมควรหรือไม่ ถ้าตัวบ่งชี้มีค่าเท่ากับ `true` คำร้องขอก็เป็นอันตกลงไป ตรงกันข้าม หากมีค่า `false` ก็จะได้รับคำตอบรับและจะเรียกเมธอด `destroyApp()` ขึ้นมาใหม่ในคราวต่อไป

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

MIDlet จะเข้าสู่สถานะหยุดทำงานได้ก็ต่อเมื่อผ่านสถานะกำลังทำงานมาก่อนแล้ว ในทางตรงกันข้ามหากต้องการเข้าสู่สถานะถูกทำลาย สามารถเข้าได้โดยตรงจากทั้งสถานะหยุดทำงานชั่วคราว และสถานะกำลังทำงาน นอกจากนี้ยังสามารถเข้าสู่สถานะหยุดการทำงานได้ในขณะที่กำลังทำงาน หรือเมื่อได้รับคำสั่งจากซอฟต์แวร์จัดการแอปพลิเคชัน

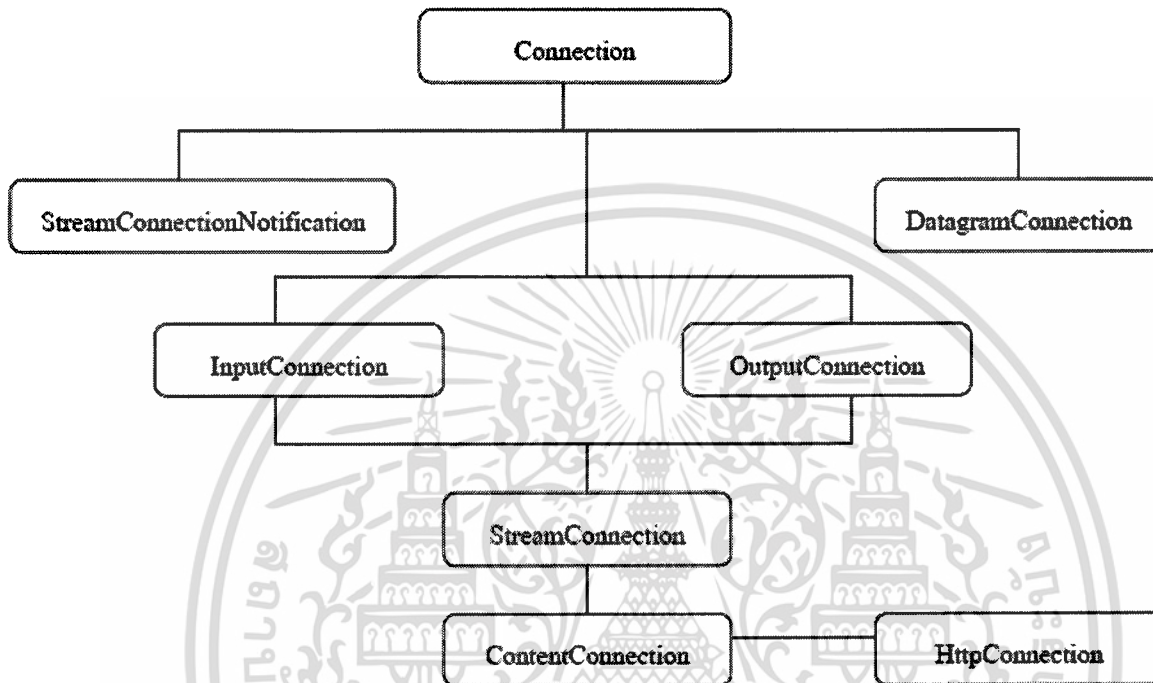
2.1.5 การเชื่อมต่อ J2ME กับเครือข่าย

การเชื่อมต่อเครือข่ายของ J2ME จะต้องมีความยืดหยุ่นสูงเพื่อรองรับอุปกรณ์หลากหลายประเภท ขณะเดียวกันก็ต้องสนับสนุนเฉพาะอุปกรณ์ด้วย โดย J2ME ได้ดึงเอาบางส่วนของแพ็คเกจ `java.io` และแพ็คเกจ `java.network` มาใช้และได้กำหนดขอบเขตการเชื่อมต่อกับระบบเครือข่ายขึ้นมาใหม่โดยเรียกว่า Generic Connection Framework (GCF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Generic Connection Framework

Generic Connection Framework (GCF) คือโครงสร้างทั่วไปของการเชื่อมต่อเครือข่ายที่มีความยืดหยุ่นต่อความหลากหลายในรูปแบบการเชื่อมต่อ GCF ประกอบด้วย 1 คลาส คือ คลาส Connector และ 7 อินเทอร์เน็ตเฟส นอกจากนี้ MIDP ยังได้กำหนดอินเทอร์เน็ตเฟสเพิ่มอีก 1 อินเทอร์เน็ตเฟส คือ HttpURLConnection ซึ่งทั้งหมดนี้ถูกบรรจุอยู่ในแพ็คเกจ javax.microedition.io



รูปที่ 2.4 อินเทอร์เน็ตเฟสใน GCF

อินเทอร์เน็ตเฟสแต่ละตัวมีหน้าที่ต่างกันดังนี้

- Connection การเชื่อมต่อแบบพื้นฐาน เช่น เปิดหรือปิดการเชื่อมต่อ
- InputConnection การเชื่อมต่อเพื่ออ่านข้อมูล
- OutputConnection การเชื่อมต่อเพื่อเขียนข้อมูล
- StreamConnection รวมการเชื่อมต่อทั้งเพื่ออ่านข้อมูลและเขียนข้อมูล ใช้ในการเชื่อมต่อแบบ Socket
- ContentConnection สืบทอดจากอินเทอร์เน็ตเฟส StreamConnection ใช้จัดการกับข้อมูลผ่านการเชื่อมต่อแบบ HTTP
- StreamConnection Notification รอให้การเชื่อมต่อกับเครือข่ายสำเร็จและคืนออบเจกต์ของอินเทอร์เน็ตเฟส StreamConnection
- DatagramConnection การเชื่อมต่อแบบ Datagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่งที่ใช้สร้างการเชื่อมต่อ

การสร้างการเชื่อมต่อเครือข่ายทุกรูปแบบเริ่มต้นจากเมธอดในคลาส

Connection

```
Connection.open(String connection_string)
```

Connection_string มีรูปแบบการคล้ายคลึงกับรูปแบบของ URL ที่ใช้อยู่ทั่วไปคือ

{ โพรโตคอล } :: { ปลายทาง } ({ พารามิเตอร์ })

- โพรโตคอลคือ รูปแบบการเชื่อมต่อซึ่งสามารถกำหนดได้หลายแบบดังนี้

คำ	ประเภทการเชื่อมต่อ
file	ไฟล์ IO
comm	การเชื่อมต่อผ่านเทอร์มินัล
socket	การเชื่อมต่อแบบ Socket
datagram	การเชื่อมต่อกับเว็บเซิร์ฟเวอร์
http	การเชื่อมต่อกับเว็บเซิร์ฟเวอร์

- ปลายทาง อาจจะเป็นชื่อของ host หรือชื่อไฟล์ก็ได้ เป็นต้น
- พารามิเตอร์ คือ ข้อมูลเพิ่มเติมในการเชื่อมต่อ เป็นตัวเลือกที่ไม่ต้องใส่ก็ได้

ตัวอย่างด้านล่างแสดงการใช้เมธอด open() เพื่อสร้างการเชื่อมต่อแบบต่างๆ โดยการกำหนดโปรโตคอลที่ต่างกันออกไปดังนี้

```
//การเชื่อมต่อแบบ http
Connection.open("http://www.monster.com");

//การเชื่อมต่อแบบ sockets
Connection.open("socket://localhost:80");

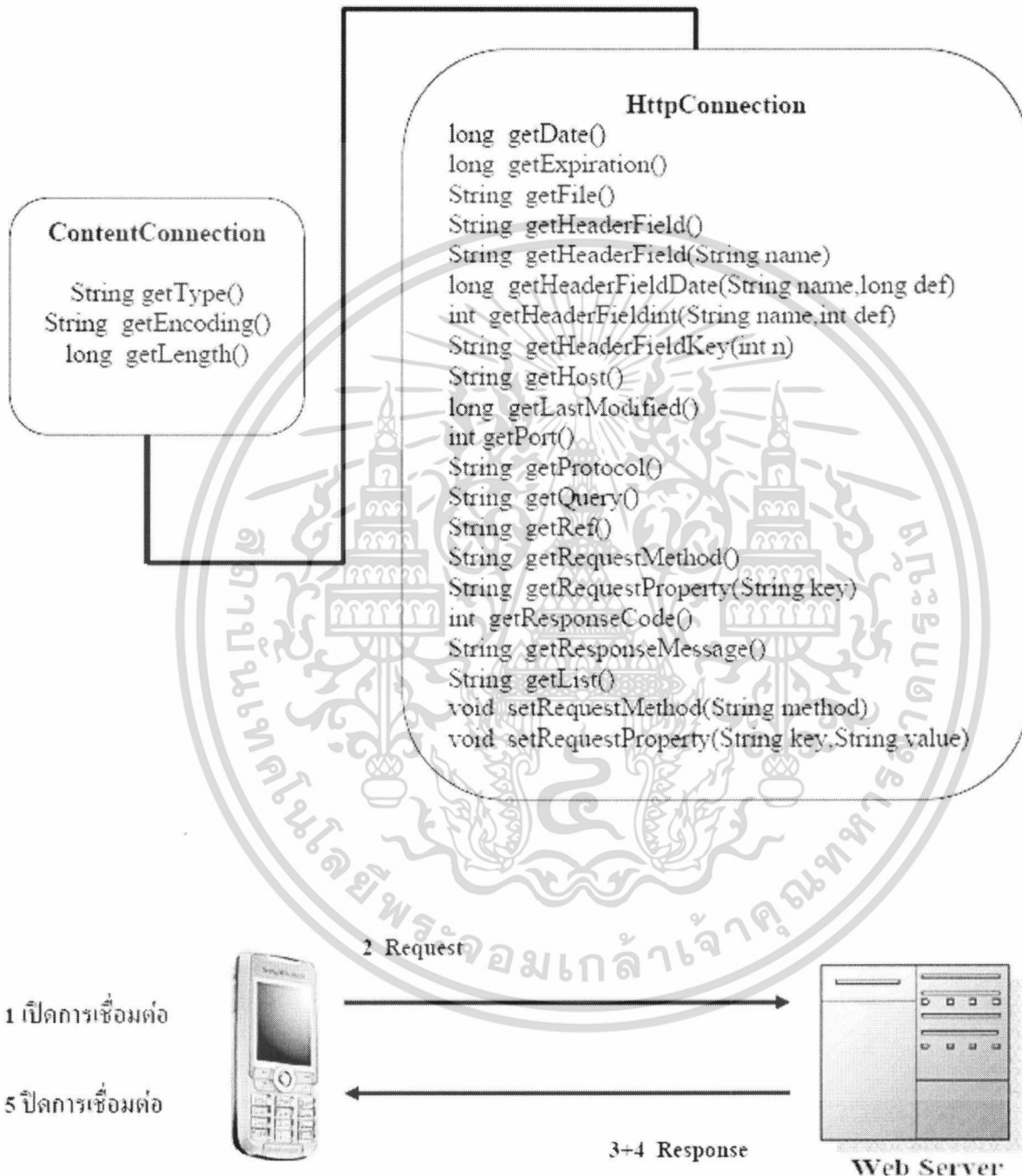
//การเชื่อมต่อแบบ Datagrams
Connection.open("datagram://http://www.moster.com:9000");

//การเชื่อมต่อกับ File:
Connection.open("file://foo.dat");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเชื่อมต่อเครือข่ายแบบ HTTP

การเชื่อมต่อเครือข่ายแบบ HTTP ใช้ในการเชื่อมต่อระหว่างโทรศัพท์มือถือและเว็บเซิร์ฟเวอร์ โดยมีการกำหนดให้อินเตอร์เฟซ HttpURLConnection เป็นมาตรฐานของอุปกรณ์ประเภท MIDP เพื่อรองรับการเชื่อมต่อเครือข่ายผ่านทาง HTTP



การเชื่อมต่อแบบ HTTP

1. เปิดการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ (สถานะ HttpURLConnection เป็น Setup)
2. โทรศัพท์มือถือส่งการร้องขอ (Request) ไปที่เว็บเซิร์ฟเวอร์โดยใช้วิธี GET, HEAD หรือ เอกสารนี้เป็นเอกสารที่ส่งมโนเวสสำหรับการแข่งขันเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปเซบระเยชนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST (สถานะของ HttpURLConnection เป็น Connected)

วิธีการร้องขอ	รายละเอียด
GET	ใช้เรียกข้อมูลจากเว็บเซิร์ฟเวอร์
POST	ใช้ส่งข้อมูลกลับไปยังเว็บเซิร์ฟเวอร์
HEAD	ใช้เรียกดูเฉพาะส่วนหัว (header) ของข้อมูลจากเว็บเซิร์ฟเวอร์

- เว็บเซิร์ฟเวอร์ตอบรับการร้องขอ (สถานะของ HttpURLConnection เป็น Connected)
- เว็บเซิร์ฟเวอร์จัดการกับคำร้องขอและส่งผลการร้องขอกลับไปโทศัพท์มือถือ (สถานะของ HttpURLConnection เป็น Connected)
- ปิดการเชื่อมต่อ (สถานะของ HttpURLConnection เป็น Closed)

2.2 เทคโนโลยีอื่นๆ

2.2.1 MySQL

มายเอสคิวแอล(MySQL) จัดเป็นระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS:Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบันตัวหนึ่ง สาเหตุก็เพราะ มายเอสคิวแอลเป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูงและมีความสามารถในการด้านของความเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมาก ทั้งยังสนับสนุนการทำงานบนระบบปฏิบัติการที่หลากหลาย ไม่ว่าจะเป็น ยูนิกซ์(Unix), โอเอสทู(OS/2), แมค โอเอส(Mac OS), ลินุกซ์(Linux) หรือวินโดวส์(Windows) ก็ตาม นอกจากนี้มายเอสคิวแอล ยังสามารถใช้งานร่วมกับ เว็บเดเวลอปเมนต์แพลตฟอร์ม(Web Development Platform)ทั้งหลาย ไม่ว่าจะเป็น ซี(C), ซีพลัสพลัส(C++), จาวา(Java), เพิร์ล(Perl), พีเอชพี(PHP), เอเอสพี(ASP) หรือ เจเอสพี(JSP) จึงเป็นสาเหตุให้ มายเอสคิวแอล ได้รับความนิยมมากในปัจจุบัน

คุณสมบัติเด่นของ มายเอสคิวแอล

- มายเอสคิวแอล เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งจะทำการเก็บข้อมูล แยกเป็นตารางแทนที่จะเก็บข้อมูลไว้รวม ๆ กันเป็นก้อนใหญ่ไว้ในที่หนึ่ง ซึ่งสิ่งนี้ได้เพิ่มความเร็วและ ความยืดหยุ่นในการใช้งานฐานข้อมูล ตารางเหล่านี้จะเชื่อมกันโดยการกำหนดความสัมพันธ์ให้แต่ละ ตาราง ซึ่งจะทำให้สามารถรวมข้อมูลจากหลาย ๆ ตารางได้
- มายเอสคิวแอล ใช้ภาษา SQL (Structured Query Language) เป็นพื้นฐานในการกระทำต่าง ๆ กับฐานข้อมูล ซึ่งภาษา SQL นี้เป็นภาษามาตรฐานในการติดต่อกับฐานข้อมูลอยู่แล้ว ทำให้ผู้ใช้สามารถเรียนรู้การใช้งาน มายเอสคิวแอลได้อย่างง่ายดายและรวดเร็วมายเอสคิวแอล เป็นซอฟต์แวร์แบบเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปไซ่ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

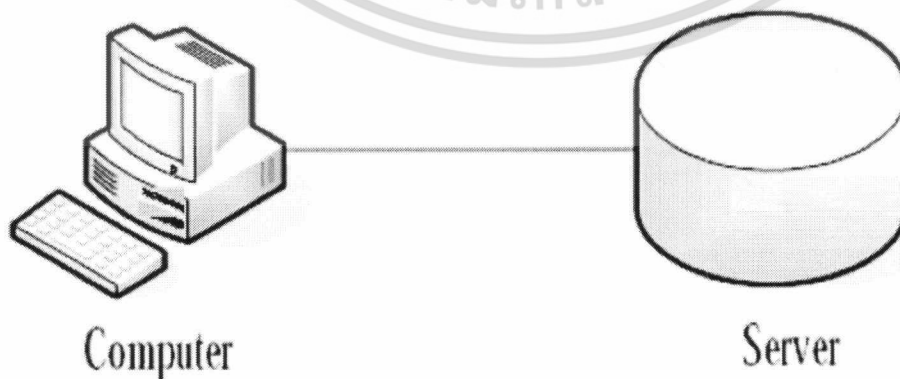
เปิด กล่าวคือ ใคร ๆ ก็ตามต่างก็มีสิทธิ์ใช้ มายเอสคิวแอล ได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ ทั้งสิ้น อีกทั้งผู้ใช้สามารถเรียนรู้การทำงานของ มายเอสคิวแอล ได้

จากซอร์ซ โค้ด และสามารถทำการแก้ไขซอร์ซโค้ดนั้นเพื่อให้ มายเอสคิวแอล มีความเหมาะสมกับความต้องการของตนได้

- มายเอสคิวแอล มีความเร็วสูงในการเข้าถึงฐานข้อมูล
- สามารถใช้ มายเอสคิวแอล ได้ในหลาย ๆ ระบบปฏิบัติการ เช่น ลินุกซ์, ยูนิกซ์, วินโดวส์
- มายเอสคิวแอล ง่ายต่อการเรียนรู้และใช้งาน

สถาปัตยกรรมของมายเอสคิวแอล

สถาปัตยกรรม หรือโครงสร้างภายในของ มายเอสคิวแอล ก็คือ การออกแบบการทำงานในลักษณะของไคลเอนต์/เซิร์ฟเวอร์(Client/Server) ซึ่งประกอบด้วยส่วนหลักๆ 2 ส่วนคือ ส่วนของผู้ให้บริการ (เซิร์ฟเวอร์) และส่วนผู้ใช้บริการ(ไคลเอนต์) โดยแต่ละส่วนก็จะมีโปรแกรมการทำงานตามหน้าที่ของตน ส่วนผู้ให้บริการ หรือเซิร์ฟเวอร์ จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูลในที่นี้ก็คือ ตัว มายเอสคิวแอลเซิร์ฟเวอร์นั่นเองและเป็นที่ยึดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้มีทั้งข้อมูลที่จำเป็นสำหรับการทำงานกับระบบฐานข้อมูลและข้อมูลที่เกิดจากที่ผู้ใช้แต่ละคนสร้างขึ้นมา ส่วนผู้ใช้บริการ หรือ ไคลเอนต์ ก็คือผู้ใช้นั่นเอง โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ มายเอสคิวแอล ไคลเอนต์, แอคเซส (Access), Web development Platform ต่างเช่น Java, Perl, PHP, ASP,JSP Computer Serverรูปภาพ 2.5 สถาปัตยกรรมของ MySQL 5.1.4.3 หลักการทำงานของ มายเอสคิวแอล 1. ที่ฝั่งของ เซิร์ฟเวอร์ จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการจาก ไคลเอนต์ 2. เมื่อมีการร้องขอการใช้บริการเข้ามา เซิร์ฟเวอร์ จะทำการตรวจสอบตามวิธีของตน เช่นอาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ มายเอสคิวแอล สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ ไคลเอนต์ ใดๆ ในระบบที่เข้าใช้บริการอีกด้วย



รูปที่ 2.5 สถาปัตยกรรมของ MySQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของ มายเอสคิวแอล

1.ที่ฝั่งของ เซิร์ฟเวอร์ จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการให้บริการจาก ไคลเอนต์

2.เมื่อมีการร้องขอการให้บริการเข้ามา เซิร์ฟเวอร์ จะทำการตรวจสอบตามวิธีของตน เช่น อาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ มายเอสคิวแอล สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธไคลเอนต์ใดๆ ในระบบที่เข้าใช้บริการอีกด้วย

3.ถ้าผ่านการตรวจสอบ เซิร์ฟเวอร์ ก็จะอนุมัติการให้บริการแก่ ไคลเอนต์ ที่ร้องขอใช้บริการนั้นต่อไปและถ้าในกรณีที่ไม่ได้รับการอนุมัติเซิร์ฟเวอร์ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไปให้ ไคลเอนต์ ที่ร้องขอบริการนั้น เครื่องคอมพิวเตอร์3607 ทำหน้าที่เป็น ไคลเอนต์ หรือ เซิร์ฟเวอร์ อาจอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบตามปกติถ้าเป็นการทำงานในลักษณะ เว็บพื้นฐาน(Web-based) มีการใช้ฐานข้อมูลขนาดไม่ใหญ่มาก ตัว มายเอสคิวแอล เซิร์ฟเวอร์และ ไคลเอนต์ มักทำงานอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน (เช่น เนื้อที่ฮาร์ดดิสก์, หน่วยความจำหลัก เป็นต้น) มากพอสมควร แต่สำหรับการทำงานจริง (RealworldApplication) ก็มักจะแยก ไคลเอนต์ และ เซิร์ฟเวอร์ ออกเป็นคนละเครื่องกัน เพราะสามารถรองรับการทำงานได้ดีกว่ามากกว่าดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับทำงานเครือข่าย จะต้องคำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะให้ระบบมีการทำงานรองรับการให้บริการผู้ใช้ อย่างมีประสิทธิภาพและข้อมูลมีความปลอดภัยมากที่สุด

วิธีการเชื่อมต่อจาก ไคลเอนต์ เข้าสู่เซิร์ฟเวอร์

จะกล่าวถึงลักษณะวิธีการเชื่อมต่อ 2 แบบ คือ แบบเนทีฟ(Native) และแบบผ่านตัวกลาง ดังนี้

1.แบบ Native เป็นที่นิยมกันมากในกรณีที่ ระบบปฏิบัติการของ มายเอสคิวแอล เซิร์ฟเวอร์ เป็น ยูนิคต์เป็นวิธีการเชื่อมต่อที่มีวิธีการทำงานเร็วที่สุด เพราะทำงานกันภายใน โดยลักษณะการทำงานประเภทนี้ได้แก่การใช้งาน มายเอสคิวแอล ร่วมกับ Web Development Platform ทั้งหมด ที่ต้องมีการ Modify ภายใน เช่นPHP

2.แบบผ่านตัวกลาง ในที่นี้จะกล่าวถึงรูปแบบที่นิยมใช้งานมากที่สุดก็คือ ODBC (Open DataBase Connectivity) ซึ่งส่วนใหญ่ใช้กับ เซิร์ฟเวอร์ ที่เป็น Windows Platform เป็นระบบปฏิบัติการ การทำงานประเภทนี้อาจมีการทำงานที่ช้ากว่าแบบ Native เพราะการทำงานในแต่ละครั้งระหว่าง ไคลเอนต์ และ เซิร์ฟเวอร์ ต้องผ่านตัวกลางก่อน แต่ ODBC ก็ถือว่ามีข้อได้เปรียบในเรื่องฐานผู้ใช้ Windows Platform มากกว่า และด้วย ODBC ทำให้สามารถใช้เครื่องมือไคลเอนต์ เดเวลลอปเมนต์ (Client Development Tools) ยอดนิยมได้มากกว่า เช่น Access, VB, ASP, JSP เพื่อเชื่อมต่อเข้าหา มายเอสคิวแอล เซิร์ฟเวอร์ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบการทำงานระหว่างแบบเนทีฟ และแบบผ่านตัวกลาง จะเห็นได้ว่าแบบผ่านตัวกลางจะใช้ขั้นตอนมากกว่าแบบ เนทีฟ ถึง 2 เท่า แต่มีข้อดีข้อด้อยแตกต่างกันสรุปได้ดังนี้

แบบ Native	แบบผ่านตัวกลาง
1. มีการทำงานรวดเร็วกว่า เพราะสื่อสารกันภายใน	1. มีการทำงานช้ากว่าเพราะมีการทำงานเพิ่มขึ้นอีก 1 ขั้นตอน
2. ไคลเอนต์ ที่จะมาเชื่อมต่อต้องมีคำสั่งส่วนของ โปรแกรมของ เซิร์ฟเวอร์ บางส่วนไว้สำหรับการใช้งาน หมายถึงเราต้องปรับปรุง ไคลเอนต์ เพิ่มเติม	2. ไม่ต้องปรับปรุง ไคลเอนต์ เพิ่มเติม เพียงแค่สนับสนุน ODBC ก็สามารทำงานได้แล้ว
3. ส่วนใหญ่มักไม่มีข้อจำกัดในการใช้งาน	3. มีข้อจำกัดขึ้นอยู่กับตัวกลางหรือ ODBC ที่ใช้
4. Server Platform ส่วนใหญ่มักเป็น ยูนิกซ์	4. ส่วนใหญ่เป็น Windows Platform
5. ไคลเอนต์ ส่วนใหญ่ใช้งานในลักษณะ Web-based เช่น Java Perl PHP เป็นต้น	5. รองรับทั้ง Web-based หรือ Client Development Tools อื่นๆ เช่น Access, VB, ASP, JSP

ตาราง เปรียบเทียบการทำงานระหว่างแบบเนทีฟ และแบบผ่านตัวกลาง

ความสามารถของ มายเอสคิวแอล

ความสามารถของ มายเอสคิวแอล โดยทั่วไปจะครอบคลุมความต้องการของผู้ใช้เพียงพอ แต่ถ้านำไปเปรียบเทียบกับฐานข้อมูลที่ขายกันในท้องตลาดแล้ว (Commercial) อาจแตกต่างกันมากปกติผลิตภัณฑ์ที่เป็น Commercial เหล่านั้น มักจะมีความสามารถต่างๆที่เกินความจำเป็นของผู้ใช้เสมอ สิ่งที่เกิดความจำเป็นเหล่านี้ถือเป็นการสูญเปลืองของผลิตภัณฑ์ เพราะทำขึ้นมาแต่ไม่ได้ถูกนำไปใช้งาน หรือใช้แต่ไม่เต็มความสามารถนอกจากนี้ก็อาจทำให้ผลิตภัณฑ์มีขนาดใหญ่ขึ้น อีกทั้งราคาก็สูงตามไปด้วย ซึ่งสำหรับ มายเอสคิวแอล แล้วจะมีสามารถที่ครอบคลุมความต้องการของผู้ใช้ ไม่มีอะไรที่เกินความจำเป็น ทั้งนี้อาจสรุปความสามารถเด่นๆได้ดังนี้

- มายเอสคิวแอล จัดเป็นระบบฐานข้อมูลประเภท SQL-based ผู้ใช้หรือผู้พัฒนาสามารถใช้คำสั่ง SQL ในการสั่ง หรือใช้งานกับ มายเอสคิวแอล เซิร์ฟเวอร์ ได้โดยไม่ต้องศึกษาเพิ่มเติมแต่อย่างใด ซึ่งความสามารถนี้ถือเป็นแนวโน้มในการจัดการฐานข้อมูลในปัจจุบัน
- สนับสนุนการใช้งานสำหรับตัวประมวลผลกลาง (CPU: Central Processing Unit) หลายตัว
- การทำงานแบบมัลติเทร็ด (Multi-threaded) ใช้ Kernel Threads
- สนับสนุน API เพื่อใช้งานกับ Development Platform ต่างๆ มากมาย ไม่ว่าจะเป็น C, C++, Eiffel, Java, Perl, PHP, Python, หรือ Tcl และนอกจากนี้ยังสามารถใช้งานร่วมกับ ODBC (Open

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นใบโฆษณาระยะสั้นในการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DataBaseConnectivity) ซึ่งทำให้สามารถใช้งานกับเครื่องมืออื่นบน Windows Platform ได้เช่น Access เป็นต้น

• มายเอสคิวเอล สามารถทำงานได้บนระบบปฏิบัติการหลายตัวหลายค่าย ไม่ว่าจะเป็น AIX,BSD/OS,DEC Unix, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2, SGI Irix,Solaris, SunOS, SCO OpenServer, SCO Unixware, Tru64 Unix, Windows Platform ทำให้ผู้ใช้สามารถทำการย้ายหรือปรับขนาดของระบบขึ้นไปได้ในกรณีที่ต้องการขยายขนาดของข้อมูล หรือมีความต้องการทรัพยากรเพิ่มมากขึ้น

• ประเภทของข้อมูล 3607 ใช้ได้ใน มายเอสคิวเอล ได้แก่ ตัวเลข(ทั้งแบบคิดและไม่คิดเครื่องหมาย) ขนาด 1, 2, 3, 4 และ 8 ไบท์ FLOAT, DOUBLE, CHAR, VARCHAR, TEXT,BLOB, DATE, TIME,DATETIME, TIMESTAMP, YEAR, SET และ ENUM

• สนับสนุน GROUP BY และ ORDER BY clauses และ Group Functions ได้แก่ COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX(), และ MIN() - สนับสนุน Left OUTERJOIN และ Right OUTER JOIN - การกำหนดสิทธิ์และรหัสผ่าน ให้มีความปลอดภัย ความยืดหยุ่นสูง สามารถกำหนดเครื่องและ/หรือผู้ใช้ ในการเข้าถึงข้อมูลได้ มีการเข้ารหัสข้อมูล (Encryption) สำหรับรหัสผ่านของผู้ใช้ด้วย ทำให้ผู้ใช้นั้นเฝ้าข้อมูลจะมีความปลอดภัย ไม่มีใครทำการเข้าถึงข้อมูลได้หากไม่ได้รับ อนุญาต

• สามารถทำดัชนี(Index) ได้สูงที่สุดถึง 32 ดัชนีในแต่ละตารางข้อมูล โดยที่ในแต่ละดัชนีสามารถใช้ฟิลด์ได้ ตั้งแต่ 1-16 ฟิลด์

• สามารถรองรับข้อมูลขนาดใหญ่ เช่น ข้อมูลระดับล้านระเบียน ซึ่งในปัจจุบัน มายเอสคิวเอล สามารถรองรับข้อมูลได้ในระดับ 60,000 ตารางข้อมูล และ 5 ล้านระเบียน

• สนับสนุนรูปแบบของภาษา (Character Set) หลายชนิด เช่น ISO-8859-1 (latin1), big5, ujis และอื่นๆ ทำให้สามารถทำการจัดเรียงข้อมูล (Sort) หรือกำหนดการแสดงผลข้อผิดพลาด (Error Message) ได้ตามรูปแบบภาษาที่ต้องการ

• เครื่องที่ทำหน้าที่เป็นผู้ให้บริการ สามารถเชื่อมเข้าสู่ มายเอสคิวเอล เซิร์ฟเวอร์ โดยการที่ใช้ที่ซีพี/ไอพีซ็อกเก็ต(TCP/IP Socket), ยูนิคซ์ ซ็อกเก็ต (Unix Socket) หรือ เนมไปป์(Name Pipes(NT))

2.2.2 JDBC

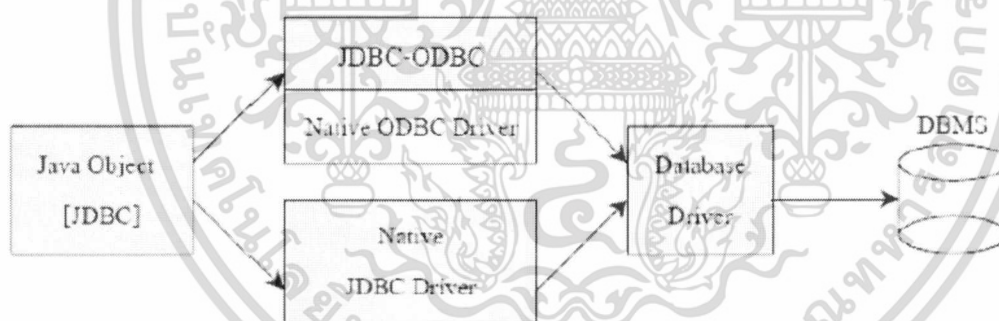
JDBC (Java Database Connectivity) ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystemsซึ่งก็คือฟังก์ชันมาตรฐานหรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล, นักพัฒนาสามารถใช้ JDBC API และยังสามารถประกอบด้วย packages อื่นๆด้วย ซึ่งนำเสนอในรูปแบบฟังก์ชันพิเศษหรือhigher level API เพื่อเขียนโปรแกรมอิสระในการกำหนด DBMS หรือ database connectivitymechanism ในการเริ่มใช้งาน, โดยทั่วไปการใช้ SQL database ในการติดต่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนักผู้ดูแลระบบไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(interface) กับ framework เพื่อที่จะจัดมาตรฐานในการติดต่อในส่วนบนสุดของชนิดต่างๆ ของ database connectivity modules ซึ่งก็คือมาตรฐานของ ANSI SQL-2 Entry level database เพราะว่า Relational databases เกือบจะทั้งหมดในปัจจุบันใช้มาตรฐานของ SQL-2 Entry level JDBC สร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC Database Connectivity ของบริษัทไมโครซอฟต์ ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN) แต่ตามหลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-level Interface ของระบบ X-Windows และ JDBC driver ต้องเข้ากันได้กับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้น

หลักการการทำงานของ JDBC

มีหน้าที่หลัก 3 อย่าง คือ

1. สร้างจุดเชื่อมต่อกับฐานข้อมูลที่ต้องการเชื่อมต่อด้วย
2. ตั้งคำสั่ง SQL (SQL Statement) ไปยังฐานข้อมูลนั้น
3. รับผลจากคำสั่งที่ส่งไปในข้อ 2



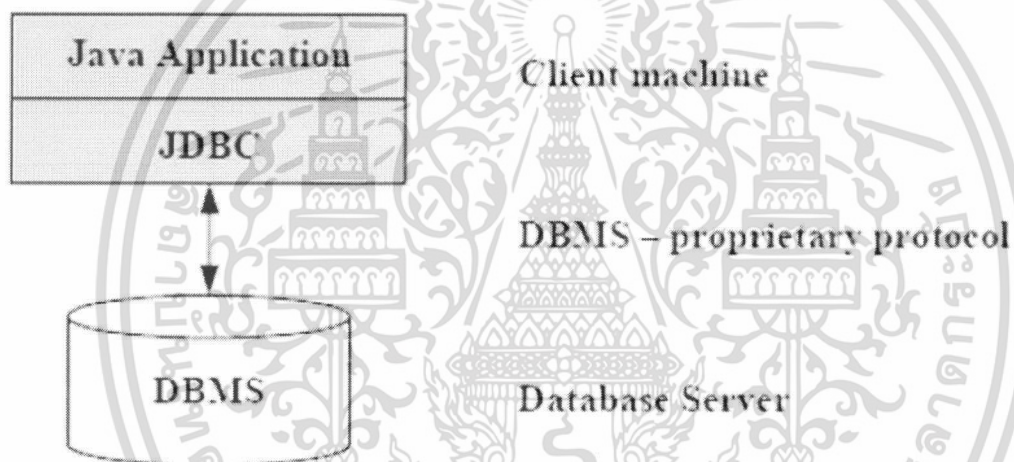
รูปที่.2.6 การทำงานของ JDBC

จากภาพ จาวาออบเจกต์ (Java Object) จะร้องขอไปยัง JDBC Class ซึ่งจะเป็นตัวเรียกไปยัง JDBC Driver หรือ JDBC-ODBC Bridge ที่กำหนด จากนั้นเน็ตเวิร์กไดรเวอร์ (Native Driver) จะเรียกตัวไดรเวอร์ของฐานข้อมูลจริง ๆ เพื่อเชื่อมต่อไปยังฐานข้อมูลอย่างสมบูรณ์

รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-Tiers model และ Three-Tiers model เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

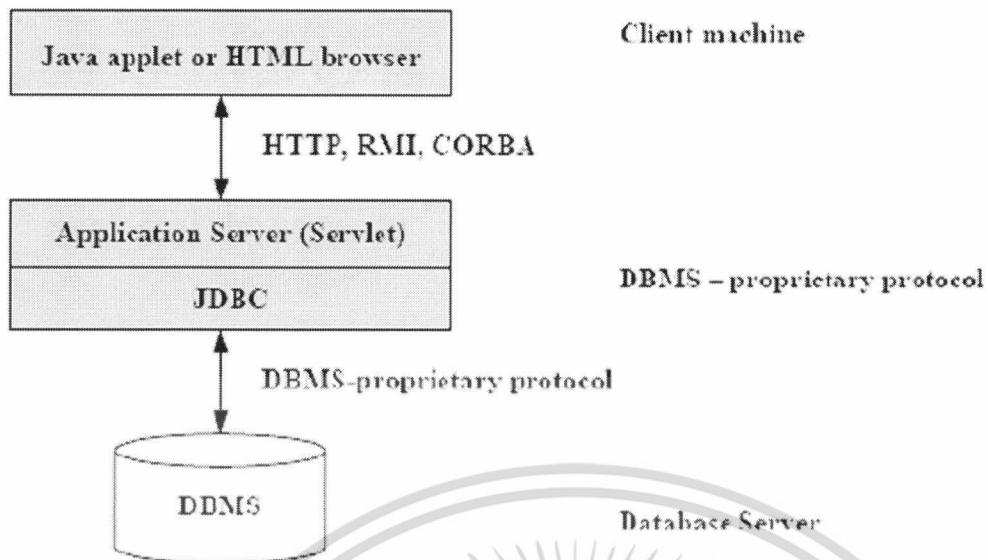
• Two – Tiers model จาว่าแอปเพล็ต (Java applet) หรือจาว่าแอปพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรง จึงมีความจำเป็นที่โปรแกรมจาว่าต้องการ JDBC ไดรเวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ คำสั่งในการค้นคืนข้อมูลในรูปของภาษาSQL (Structured Query Language) จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้งานข้อมูลดังกล่าวนี้ ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้สามารถเชื่อมต่อผ่านระบบเน็ตเวิร์ก (Network) รูปแบบ Two – Tiers นี้ใช้หลักการทำงานเช่นเดียวกับรูปแบบไคลเอนต์/เซิร์ฟเวอร์ที่เราารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้ คือไคลเอนต์และเครื่องคอมพิวเตอร์ ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครือข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two-Tiers มักเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร



รูปที่ 2.7 Two – Tiers Model

• Three – Tiers Model คำสั่งค้นต่างๆจากผู้ใช้งานจะถูกส่งไปที่ Middle Tier หรือส่วนกลางของการบริการ เสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปที่ระบบจัดการ ฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่ง ต่อไปให้ผู้ใช้งานในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ต ซึ่งประกอบด้วยเครื่อง คอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอนต์และเซิร์ฟเวอร์ฐานข้อมูลสามารถพูดคุยกันได้ การปรับเปลี่ยนระบบ คอมพิวเตอร์ของไคลเอนต์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบ ซึ่งกันและกันเกิดขึ้น

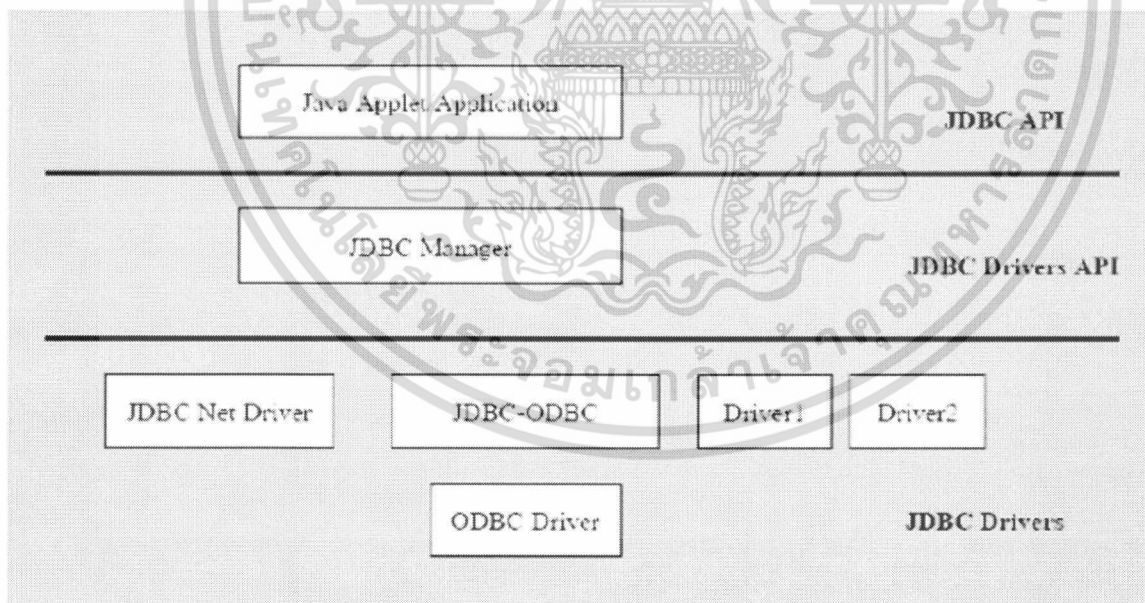
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Three - Tiers Model

โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายในJDBC ประกอบด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver



รูปที่ 2.9 ระดับการเชื่อมต่อของ JDBC API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JDBC API ประกอบด้วยส่วนติดต่อ (Interface) 2 ส่วนที่สำคัญ ส่วนแรกคือ JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่ผู้เขียนโปรแกรมประยุกต์(application writers) ส่วนที่สองคือ JDBC driver API ในระดับ lower-level สำหรับผู้เขียนไดรเวอร์ (driver writers)

รูปแบบของ JDBC ไดรเวอร์ (JDBC drivers)

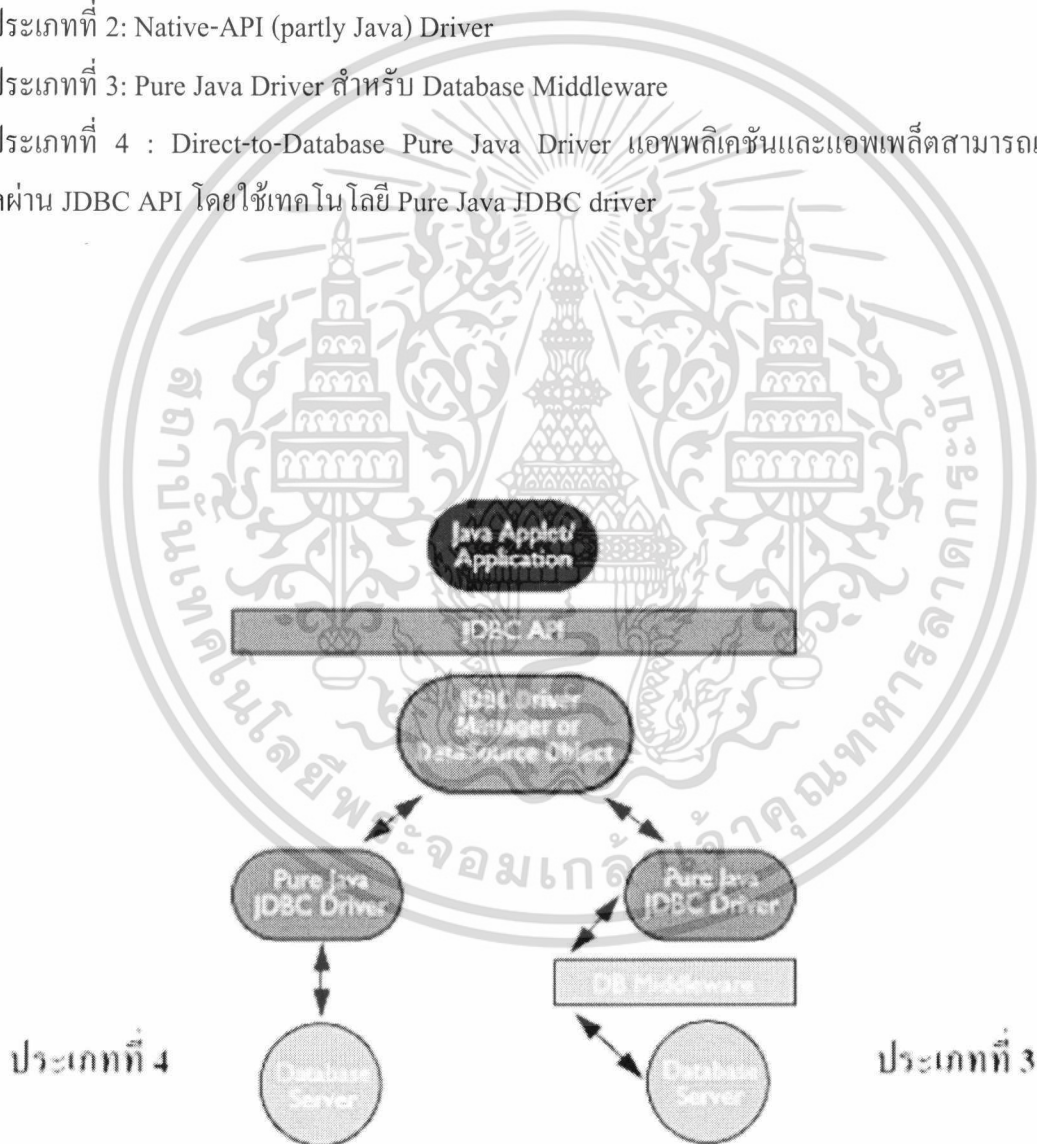
JDBC ไดรเวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆของ JDBC โดยเทคโนโลยี JDBC drivers ถูกแบ่งออกเป็น 4 ประเภทด้วยกันคือ

ประเภทที่ 1: JDBC-ODBC Bridge plus ODBC Driver

ประเภทที่ 2: Native-API (partly Java) Driver

ประเภทที่ 3: Pure Java Driver สำหรับ Database Middleware

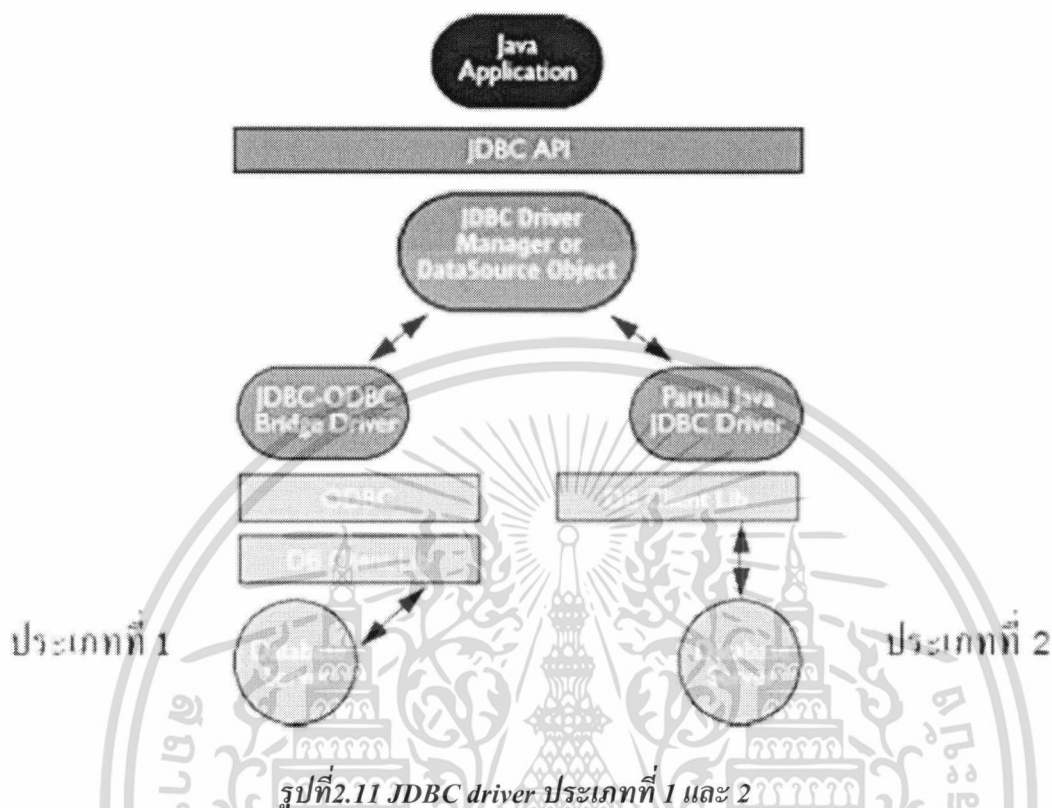
ประเภทที่ 4 : Direct-to-Database Pure Java Driver แอปพลิเคชันและแอปเพล็ตสามารถเข้าถึงฐานข้อมูลผ่าน JDBC API โดยใช้เทคโนโลยี Pure Java JDBC driver



รูปที่ 2.10 JDBC driver ประเภทที่ 3 และ 4 ที่ใช้ pure Java JDBC driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JDBC Connectivity โดยใช้ ODBC ไดรเวอร์ (ODBC drivers) และ libraries ของฐานข้อมูลในเครื่องไคลเอ็นต์ที่มีอยู่แล้ว



ประเภท 1 (Type1: JDBC-ODBC Bridge)

มักเรียกกันสั้นๆว่า JDBC-ODBC บริดจ์ หมายถึง JDBC ที่ทำงานอยู่บนชั้นการสื่อสารมิดเดิลแวร์ที่ชื่อ ODBC อีกทีหนึ่ง ถูกใช้สำหรับการศึกษาวิธีการเขียนโปรแกรมจาวา เพื่อติดต่อกับ JDBC เหมาะกับระบบงานที่มี ODBC ในการทำงานอยู่แล้ว

- สำหรับระบบงานที่ทั้งหมดทำงานอยู่บนพื้นฐานของแพลตฟอร์มวินโดวส์
- ไม่เหมาะกับการนำไปใช้งานบนระบบงานที่มีขนาดใหญ่ๆเนื่องจากทำให้เกิดความซ้ำในกาการทำงานและประสิทธิภาพในการทำงานที่ไม่ดี
- มีข้อมูลในส่วน โอเวอร์เฮด(Overhead) สูง เนื่องจากต้องมีส่วนในการติดต่อระหว่าง JDBC และ ODBC เพิ่มเติม
- ไม่สนับสนุนความสามารถทั้งหมดของมาตรฐาน JDBC เนื่องจากข้อจำกัดของ ODBC ที่มีรูปแบบการทำงานน้อยกว่า JDBC

ประเภท 2 (Type2: Native-API Driver)

สำหรับ JDBC ประเภทนี้ ตัวไดรเวอร์ (มิดเดิลแวร์ของแต่ละผลิตภัณฑ์มักจะใช้คำว่าไดรเวอร์แทนมิดเดิลแวร์ตัวนั้นๆ) จะถูกโหลดไว้ที่ไคลเอ็นต์หรือ Tier ส่วนติดต่อกับผู้ใช้

- มีประสิทธิภาพในการดีกว่าประเภท 1 เมื่อเปรียบเทียบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งในการติดต่อกับเซิร์ฟเวอร์ จะเป็นคำสั่งที่ดีที่สุดสำหรับเซิร์ฟเวอร์นั้นๆ โดยเฉพาะ ทำการทำงานโดยรวมดีกว่า ข้อดีของการนำไปใช้งาน
- ผู้ใช้โปรแกรมในส่วนไคลเอ็นต์ยังต้องการไดรเวอร์ สำหรับเซิร์ฟเวอร์โดยเฉพาะ
- เมื่อมีการเปลี่ยนแปลงเซิร์ฟเวอร์เป็นผลิตภัณฑ์ตัวอื่น โปรแกรมในส่วนไคลเอ็นต์ต้องมีการเปลี่ยนแปลงและคอมไพล์ใหม่เสมอ

ประเภท 3 (Type 3: Pure Java Driver)

ต้องมีแอปพลิเคชันเซิร์ฟเวอร์ทำงานร่วมอยู่ในระบบ

- เป็นระบบที่มีประสิทธิภาพในการทำงานที่ดีกว่าประเภทที่ 1 และ 2 เมื่อเปรียบเทียบกัน
- เหมาะสำหรับองค์กรที่มีเซิร์ฟเวอร์ทางด้านฐานข้อมูลที่หลากหลายรูปแบบ
- การทำงานของไคลเอ็นต์ไม่จำเป็นต้องติดตั้ง JDBC ไดรเวอร์ไว้ในทุกตัว
- การติดตั้งและการดูแลระบบไคลเอ็นต์ทำได้ง่าย และสะดวก
- ยังต้องการ ไดรเวอร์สำหรับแต่ละผลิตภัณฑ์ เพื่อติดตั้งไว้ที่แอปพลิเคชันเซิร์ฟเวอร์

ประเภท 4 (Type 4: Direct-to-DB)

ลักษณะการทำงานของ JDBC ในประเภทที่ 4 มีการทำงานโดยการส่งคำสั่ง SQL จากไคลเอ็นต์ไปที่เซิร์ฟเวอร์ฐานข้อมูล โดยตรงผ่านระบบเครือข่าย

- มีประสิทธิภาพในการทำงานดีที่สุดในเมื่อเทียบกับทุกประเภทที่กล่าวมา
- มีความยุ่งยากในการพัฒนาโปรแกรมเพื่อทำงานในฝั่งไคลเอ็นต์ เพราะผู้พัฒนาต้องเรียนรู้การทำงานเซิร์ฟเวอร์แต่ละตัวก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนและวิธีการดำเนินการ

ในการทำปัญหาพิเศษเรื่อง Food Order Application On Mobile มีกระบวนการกิจกรรมตามขั้นตอนและวิธีการดำเนินงาน ดังต่อไปนี้

1. ขั้นตอนการวิเคราะห์ระบบการทำงาน
2. ขั้นตอนการออกแบบโครงสร้างของระบบ

สำหรับในแต่ละขั้นตอนและวิธีดำเนินงานดังกล่าว มีรายละเอียดการทำงานดังนี้

3.1 ขั้นตอนการวิเคราะห์ระบบการทำงาน

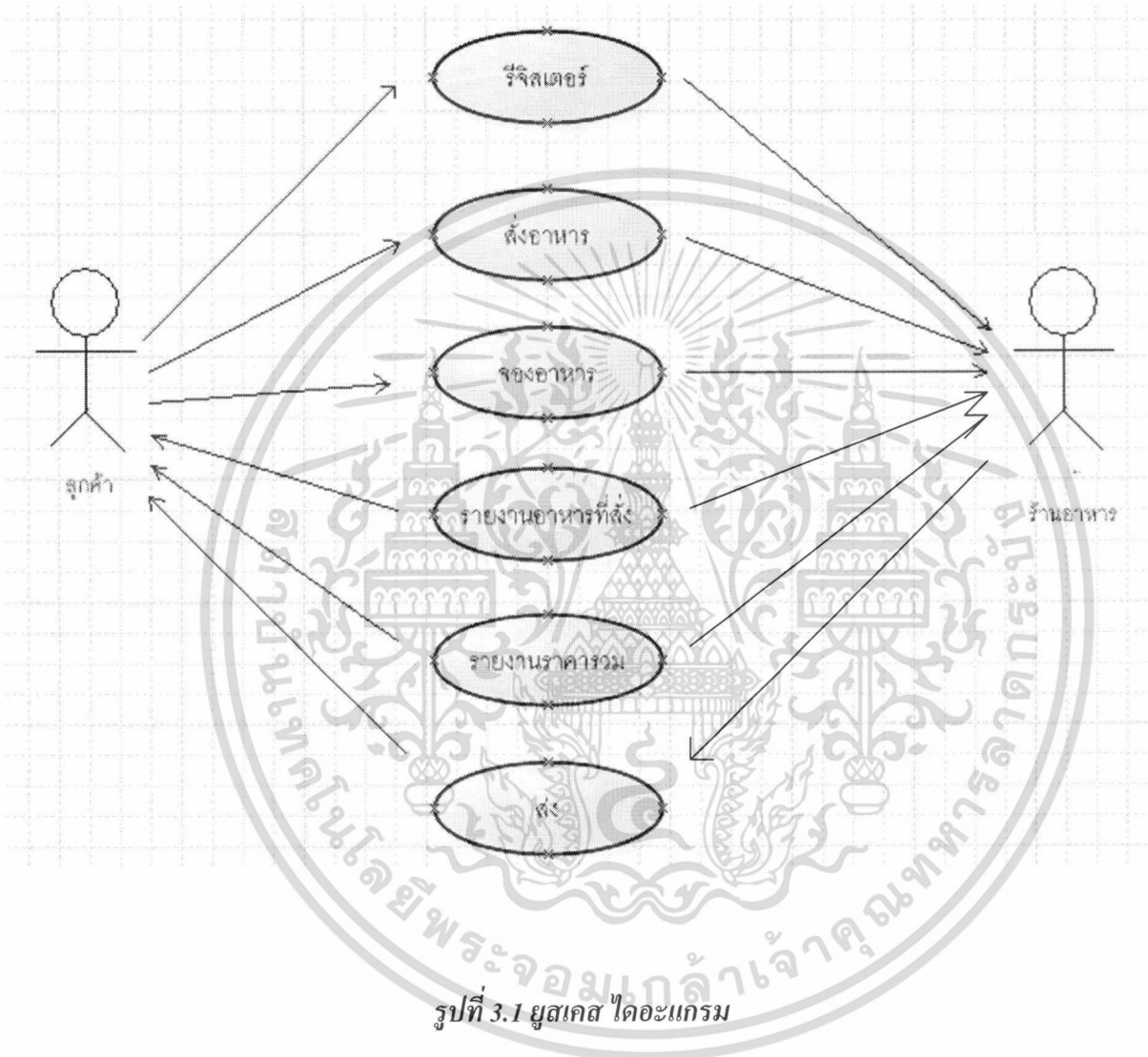
ปัญหาพิเศษนี้ มีความต้องการสร้างแอปพลิเคชันในการสั่งอาหารผ่านมือถือ หน้าหลักของโปรแกรมมีดังนี้

1. สามารถทำการรีจิสเตอร์เพื่อเก็บข้อมูล ชื่อ ที่อยู่ ของลูกค้า แล้วกำหนด รหัสให้ลูกค้าได้ในการสั่ง โดยลูกค้าต้องจดจำรหัสลูกค้าที่ให้ ไปเพื่อใช้สั่งอาหารและข้อมูลต่างของลูกค้าจะถูกนำไปเก็บลงฐานข้อมูลไว้ใช้ในการอ้างอิงต่อไป
2. มีรายการเมนูหลักให้เลือกประเภทอาหาร คาว หวาน เครื่องดื่ม ซึ่งในแต่ละเมนูจะมีรูป ชื่ออาหาร และราคาของอาหารอยู่ และจะรวมราคาอาหารให้ด้วย เมื่อมีการยืนยันการสั่งอาหาร โดยจะมีการทบทวน รายการอาหารให้ลูกค้าด้วย
3. เมื่อสั่งอาหารระบบจะถามรหัสลูกค้าที่ได้ให้ไปแล้วเพื่อให้ระบบที่ส่งอาหาร โดยจะมีให้เลือก 3 กรณี
 - สั่งตามที่อยู่ที่รีจิสเตอร์ไว้ คือ เมื่อมีการสั่งอาหารจะส่งไปยังที่อยู่รีจิสเตอร์ไว้ในตอนแรก
 - ระบบที่ส่งเอง คือ ให้ลูกค้าสามารถกรอกที่อยู่ที่จะส่งอาหารได้
 - สั่งจองเพื่อไปรับที่ร้านคือให้ลูกค้าสามารถจองอาหารเพื่อความรวดเร็วโดยสามารถ กำหนดเวลาที่จะไปถึงที่ร้านได้
4. เมื่อมีการส่งข้อมูลมา Server ก็จะทำการติดต่อกับ Database เพื่อดึงข้อมูลต่างๆ แล้วส่งต่อไปยังแอปพลิเคชัน ที่ร้าน
5. แอปพลิเคชันที่ร้านจะแสดงผลรายการอาหาร ชื่อลูกค้า รหัสลูกค้า ที่อยู่ และราคารวมของอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการออกแบบโครงสร้างของระบบ

3.2.1 ยูสเคส ไดอะแกรม



รูปที่ 3.1 ยูสเคส ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

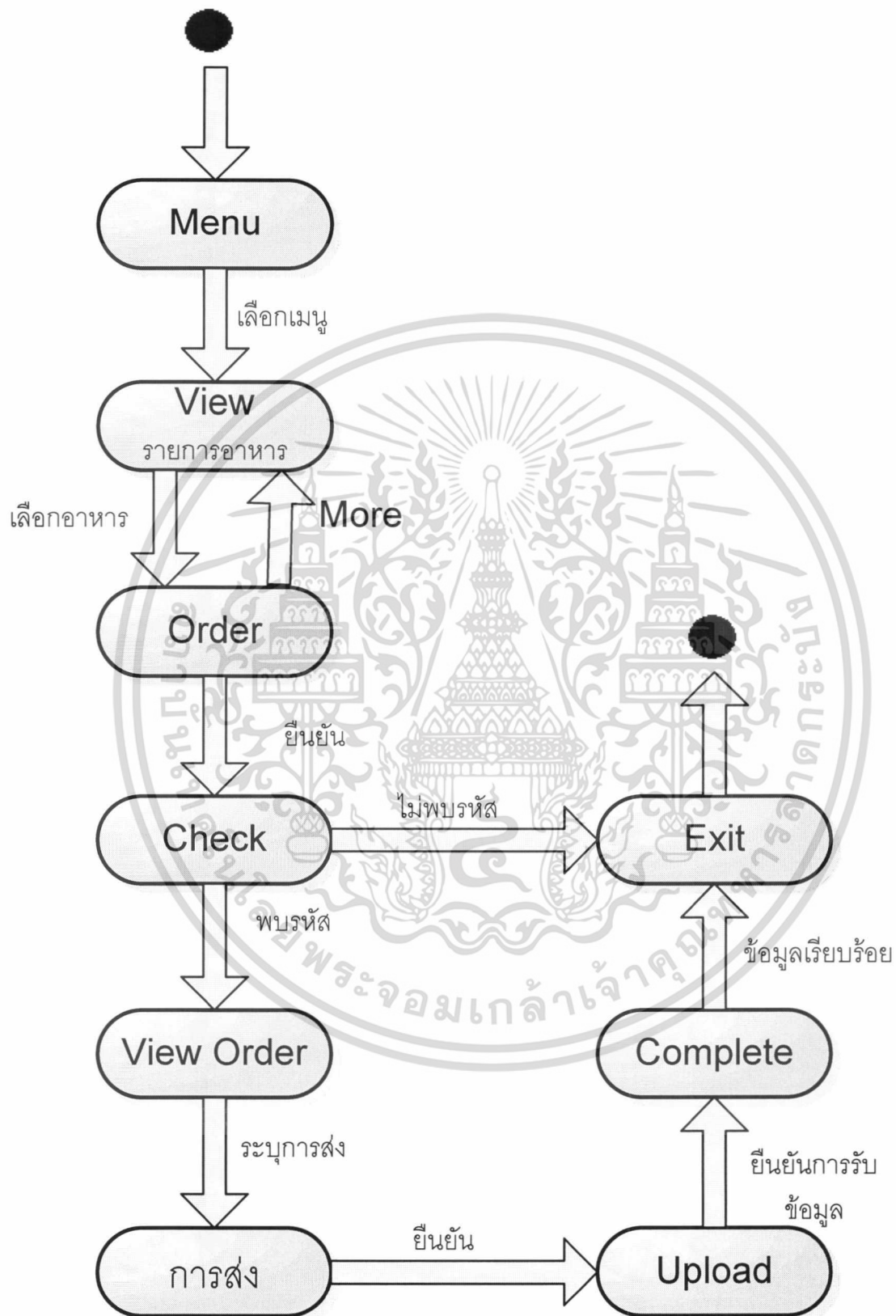
แผนภาพยูสเคส ไดอะแกรม

จากยูสเคส ไดอะแกรม แสดงถึงกิจกรรมต่างๆของระบบ ที่เกิดขึ้นระหว่าง ลูกค้า กับ ร้านอาหาร ซึ่งประกอบด้วย ยูสเคส ทั้งหมด 6 ยูสเคสดังนี้

1. **รีจิสเตอร์** บอกถึงกิจกรรมที่ลูกค้าต้องการสมัคร หรือลงทะเบียนเพื่อเก็บข้อมูลลูกค้าใหม่ผ่านทางแอปพลิเคชัน เพื่อเป็นฐานข้อมูลของตัวเองโดยลูกค้าต้องกรอกข้อมูล ชื่อ ที่อยู่ เบอร์โทรศัพท์ แล้วระบบจะทำการสร้าง รหัสลูกค้าให้เองโดยอัตโนมัติเพื่อที่ลูกค้าสามารถที่จะสั่งอาหารได้กับทุกๆเครื่องที่มีแอปพลิเคชัน ลูกค้าต้องจดจำรหัสนั้นไว้เพื่อเป็นตัวแทนของข้อมูลลูกค้าเองและใช้ในการสั่งอาหารผ่านทางมือถือในครั้งต่อไป โดยไม่ต้องรีจิสเตอร์ซ้ำอีกครั้ง
2. **สั่งอาหาร** บอกถึงกิจกรรมที่ลูกค้าสั่งอาหารผ่านระบบ โดยที่แอปพลิเคชันจะมีเมนูอาหารให้เลือก ทั้งหมด 3 ประเภทคือ อาหารคาว ของหวานและเครื่องดื่ม โดยที่แต่ละเมนูจะมีรายการอาหารมากมายพร้อมทั้งมีราคากำกับไว้ด้านหลัง ลูกค้าจะต้องเลือกอาหารผ่านทางเมนูโดยการกด Soft key เพื่อติกรายการอาหารทีละรายการ เมื่อลูกค้าเลือกอาหารจนครบแล้วจะต้องกรอกรหัสลูกค้าเพื่อยืนยันการสั่งอาหาร แล้วระบบจะให้ลูกค้าระบุสถานที่สั่งอาหาร โดยมีให้เลือก 2 แบบ
 - สั่งตามที่อยู่ที่รีจิสเตอร์ไว้แล้ว โดยไม่ต้องกรอกข้อมูลใดๆระบบจะทำการค้นที่อยู่ลูกค้าจากฐานข้อมูลเองแล้วแจ้งให้ร้านอาหารทราบ
 - ระบุสถานที่ส่ง ลูกค้าจะต้องกรอกสถานที่สั่งอาหารหลังจากนั้นทำการ อัปโหลดเพื่อแจ้งให้ทางร้านอาหารทราบเพื่อทำการส่งอาหารต่อไป
3. **จองอาหาร** บอกถึงกิจกรรมที่ลูกค้าจองอาหารผ่านระบบ ในกรณีที่ลูกค้าต้องการจองอาหารก็จะต้องทำการเลือกรายการอาหารทางเมนู เมื่อลูกค้าทำการเลือกอาหารจนครบแล้วจะต้องกรอกรหัสลูกค้าเพื่อยืนยันการสั่งอาหาร แล้วระบบจะให้กรอกรหัสลูกค้าเพื่อยืนยันตัวตนลูกค้า และลูกค้าจะต้องระบุเวลาที่จะไปถึงที่ร้านอาหารเพื่อรับอาหารกลับบ้าน
4. **รายงานอาหารที่สั่ง** บอกถึงกิจกรรมของระบบที่แสดงผลรายการอาหารที่ลูกค้าสั่งเพื่อให้ลูกค้าตรวจสอบ รายการอาหารก่อนที่จะทำการอัปโหลดข้อมูลเพื่อให้ลูกค้าตรวจทาน ระบบยังทำการแสดงผลรายการอาหารแจ้งไปยังที่ร้านอาหารด้วยเพื่อการดำเนินงานทางร้านต่อไป
5. **รายงานราคารวม** บอกถึงกิจกรรมของระบบที่แสดงราคาสุทธิทั้งหมดของอาหารที่ลูกค้าสั่งไปจะแสดงผลที่มีมือถือลูกค้า และแสดงผลที่ร้านอาหารด้วยเช่นกัน
6. **ส่ง** บอกถึงกิจกรรมของร้านอาหารที่จัดทำอาหารตามรายการอาหารต่างๆ แล้วดำเนินการส่งอาหารไปให้ลูกค้าตามที่อยู่ที่ลูกค้าระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 สเตทไดอะแกรม



รูปที่ 3.2 สเตทไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพสเตทไดอะแกรม

จากสเตท ไดอะแกรม แสดงการเปลี่ยนสถานะของระบบ เมื่อมีการกระทำต่างๆเข้ามาภายในระบบสั่งอาหารผ่านมือถือ ซึ่งประกอบไปด้วย 9 สเตท ดังนี้

1. Menu เป็นการเปลี่ยนแปลงสเตทเมื่อแอปพลิเคชันถูกโหลดลงโทรศัพท์มือถือเรียบร้อยแล้ว และมีการเปิดใช้ จะเข้ามาสู่สเตทนี้คือ มีเมนูให้เลือกว่าจะดำเนินการอะไร

- รีจิสเตอร์
- เมนูอาหาร
- เมนูของหวาน
- เมนูเครื่องดื่ม
- สั่งอาหาร
- Exit

2. View รายการอาหาร เข้าสู่สเตทนี้เมื่อมีการเลือกเมนูอาหาร แล้วจะปรากฏรายการอาหารต่างๆมากมายตามเมนูที่เลือกไป ทั้งอาหารคาว อาหารหวาน และเครื่องดื่ม ลูกค้าน่าเลือกอาหารทีละเมนู คือ ถ้าเข้าเมนูอาหารคาวก็จะมีแต่รายการอาหารคาวทั้งหมด ถ้าเข้าเมนูอาหารหวานก็จะมีแต่อาหารหวานทั้งหมด เมื่อเลือกอาหารครบแล้วกดยืนยันจะแสดงรายการอาหารที่เลือกไปทั้งหมดพร้อมทั้งแสดง ราคาสุทธิของแต่ละเมนู

3. Order จะเข้าสู่สเตทนี้ เมื่อลูกค้าเลือกอาหารที่จะสั่งจากรายการอาหารเรียบร้อยแล้ว ระบบก็จะให้ลูกค้าใส่รหัสลูกค้า เพื่อตรวจสอบข้อมูลของลูกค้า แล้วกดปุ่ม send ระบบจะส่งข้อมูลไปยังเซิร์ฟเวอร์ เพื่อทำขั้นตอนต่อไป

4. Check เมื่อทำการกรอกรหัสลูกค้าแล้วจากข้อ 3 จะต้องทำการตรวจสอบจากเซิร์ฟเวอร์ก่อนว่ามีรหัส ลูกค้านี้หรือไม่ถ้ามีก็จะทำการเปลี่ยนสถานะสเตทต่อไป แต่ถ้าไม่มีก็จะทำการออกจากระบบมาสู่หน้า เมนูหลักพร้อมแจ้งผลให้ลูกค้าทราบ

5. View Order เมื่อตรวจสอบรหัสลูกค้าแล้วว่ามีลูกค้า รหัสนี้อยู่ ก็จะมีการแสดงผลรายการอาหารอีกครั้งแต่ครั้งนี้คือรายการอาหารทั้งหมดที่ลูกค้าเลือกไว้แล้ว พร้อมด้วยราคาอาหารสุทธิ ถ้าลูกค้าพอใจกับ อาหารที่สั่งไปและราคา ก็จะดำเนินการต่อไปถ้าไม่พอใจก็สามารถออกจากระบบหรือกลับไปเปลี่ยนแปลงรายการอาหารใหม่ได้

6. การส่ง เข้าสู่สเตทการส่งเมื่อลูกค้าพึงพอใจที่จะสั่งอาหารแล้ว ก็จะให้ลูกค้าระบุสถานที่ที่จะส่งอาหาร โดยมีให้เลือกทั้งหมด 3 แบบ แบบที่1. ส่งตามที่อยู่ที่รีจิสเตอร์ไว้ 2. ส่งตามที่อยู่โดยระบุสถานที่ใหม่แบบนี้ลูกค้าต้องกรอกสถานที่ใหม่ที่ต้องการ 3.จองอาหารเพื่อไปรับที่ร้านแบบนี้ลูกค้าต้องระบุเวลาที่จะไปรับอาหารที่แน่นอน

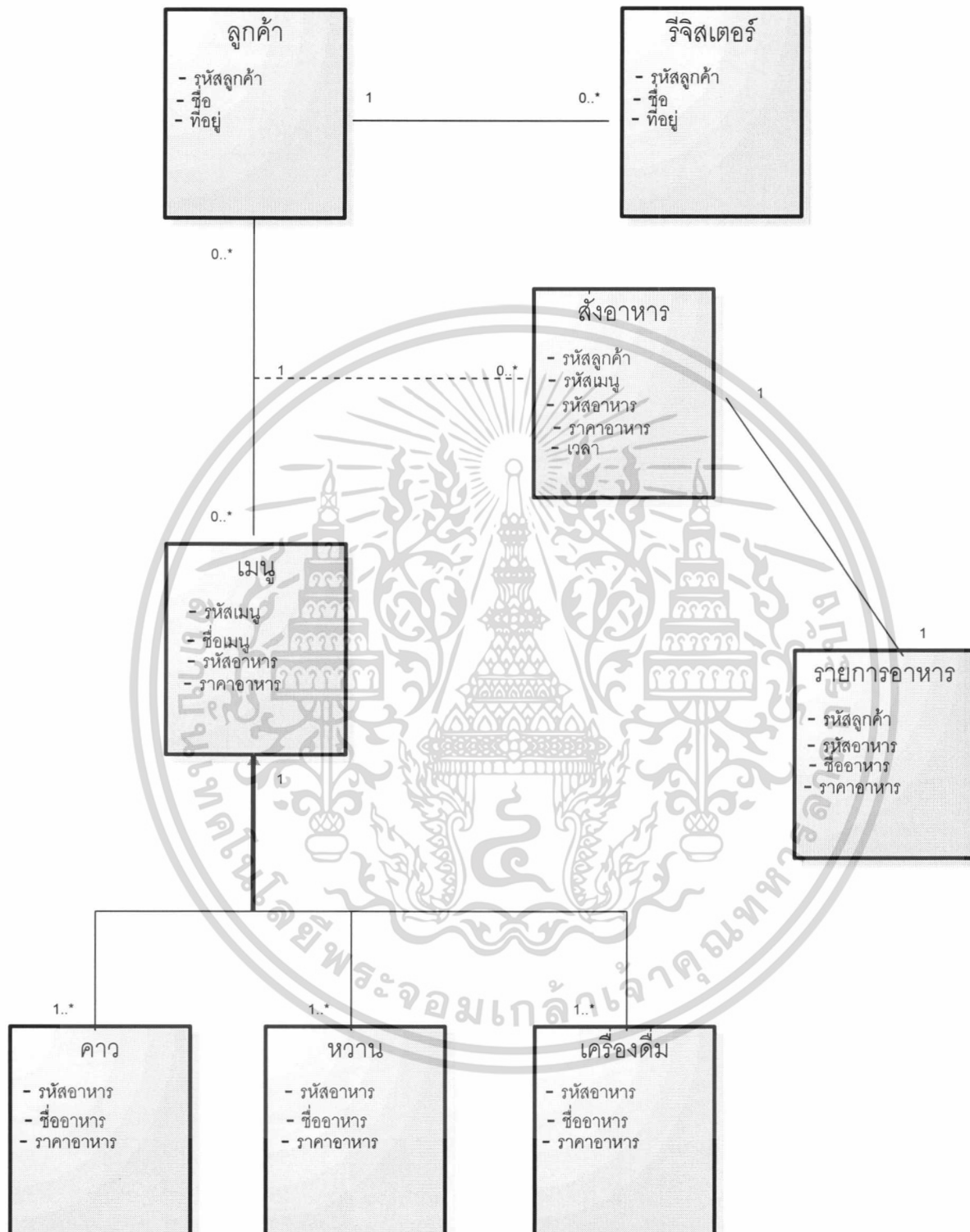
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. **Upload** เข้าสู่สถานะนี้เมื่อลูกค้าดำเนินการต่างๆข้างต้นเรียบร้อยแล้วก็จะส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อให้เซิร์ฟเวอร์จัดการส่งข้อมูลต่อไปยังร้านอาหารอีกทีหนึ่ง
8. **Complete** เมื่อข้อมูลถูกส่งไปเรียบร้อยแล้วและไม่มีปัญหาใดๆเซิร์ฟเวอร์จะแจ้งกลับมา
9. **Exit** ออกจากระบบกลับมายังหน้าเมนูหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 คลาสไดอะแกรม



รูปที่ 3.3 คลาส ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การเก็บข้อมูลลงฐานข้อมูล

เนื่องจากต้องมีการเก็บข้อมูลลงฐานข้อมูลจึงได้ออกแบบ โครงสร้างหลักที่ใช้เก็บข้อมูลของระบบ ดังต่อไปนี้

1. ตารางข้อมูลลูกค้า ใช้เก็บข้อมูลลูกค้าเพื่อใช้ดำเนินงานต่างๆ โดยจะเก็บข้อมูลของลูกค้า ต่างๆ ได้แก่ รหัสลูกค้า, ชื่อลูกค้า, ที่อยู่, เบอร์โทรศัพท์

ชื่อตาราง Customer

ชื่อคีย์หลัก ID Customer

ตารางที่ 3.1 ตารางข้อมูลลูกค้า

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	คีย์
1	ID Customer	Text	รหัสลูกค้า	P.K.
2	Name	Text	ชื่อลูกค้า	
3	Address	Text	ที่อยู่ลูกค้า	
4	PhoneNumber	Number	เบอร์โทรศัพท์ลูกค้า	

2. ตารางรายการอาหารคาว ใช้เก็บข้อมูลอาหารคาวเพื่อใช้ดำเนินงานต่างๆ โดยจะเก็บข้อมูลของอาหารคาวต่างๆ ได้แก่ รหัสอาหาร, ชื่ออาหาร, ราคา

ชื่อตาราง Food

ชื่อคีย์หลัก ID Food

ตารางที่ 3.2 ตารางรายการอาหารคาว

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	คีย์
1	ID Food	Text	รหัสอาหาร	P.K.
2	Name	Text	ชื่ออาหาร	
3	Price	Number	ราคา	

3. ตารางรายการของหวาน ใช้เก็บข้อมูลของหวานเพื่อใช้ดำเนินงานต่างๆ โดยจะเก็บข้อมูลของของหวานต่างๆ ได้แก่ รหัสของหวาน, ชื่อของหวาน, ราคา

ชื่อตาราง Refreshments

ชื่อคีย์หลัก ID Refreshments

ตารางที่ 3.3 ตารางรายการของหวาน

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	คีย์
1	ID Refreshments	Text	รหัสของหวาน	P.K.
2	Name	Text	ชื่อของหวาน	
3	Price	Number	ราคา	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตารางรายการเครื่องดื่ม ใช้เก็บข้อมูลเครื่องดื่มเพื่อใช้ดำเนินงานต่างๆ โดยจะเก็บข้อมูลของเครื่องดื่มต่างๆ ได้แก่ รหัสเครื่องดื่ม,ชื่อเครื่องดื่ม,ราคา

ชื่อตาราง Drink

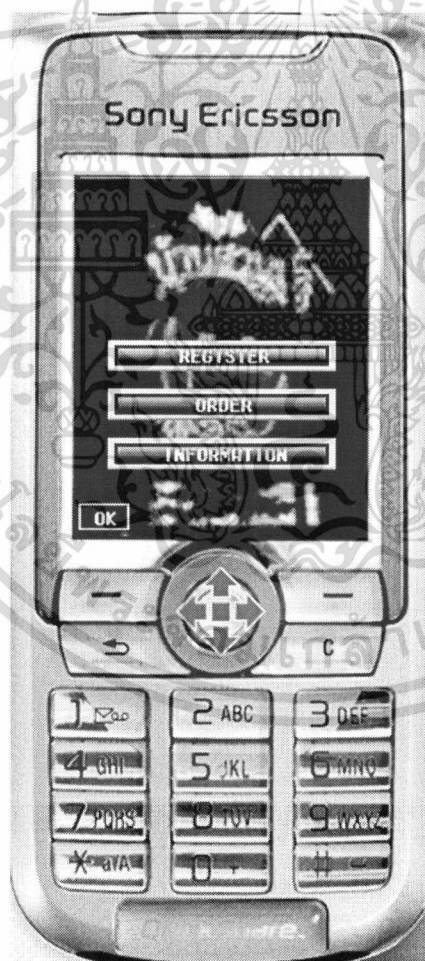
ชื่อคีย์หลัก ID Drink

ตารางที่ 3.4 ตารางรายการเครื่องดื่ม

ลำดับที่	ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย	คีย์
1	ID Drink	Text	รหัสเครื่องดื่ม	P.K.
2	Name	Text	ชื่อเครื่องดื่ม	
3	Price	Number	ราคา	

3.2.5 ตัวอย่าง Interface โทรศัพท์มือถือ

1. ตัวอย่างหน้าจอเมนูหลักของโทรศัพท์มือถือ



รูปที่ 3.4 ตัวอย่างหน้าจอเมนูหลักของโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตัวอย่างหน้าจอเมนูอาหารของโทรศัพท์มือถือ



รูปที่ 3.5 ตัวอย่างหน้าจอเมนูอาหารของโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

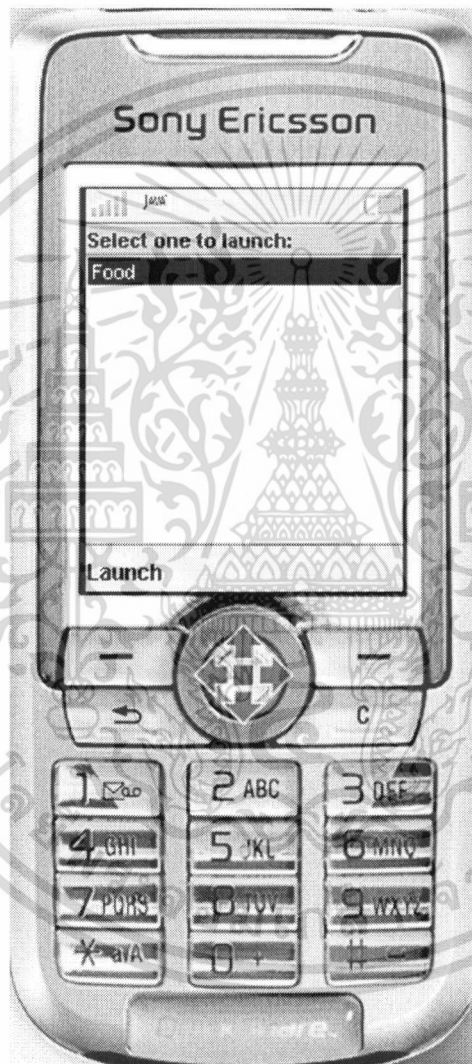
บทที่ 4

ผลการทดลอง

4.1 ลักษณะของระบบการสั่งอาหารผ่านมือถือ

4.1.1 การสั่งอาหารบนมือถือ

ลักษณะหน้าจอการสั่งอาหารบนมือถือ



รูปที่ 4.1 หน้าแรก

จากรูปที่ 4.1 เป็นหน้าเริ่มแรกของการเลือกโปรแกรม เมื่อผู้ใช้ได้ดาวน์โหลดโปรแกรมลงบนมือถือแล้วก็จะปรากฏตัวชื่อโปรแกรมขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าจอเมนูหลัก

จากรูปที่ 4.2 เป็นหน้าจอให้ผู้เลือกใช้การทำงาน ผู้ใช้สามารถเลื่อนปุ่มขึ้น-ลง ตามลูกศรได้ตามต้องการ แล้วกดปุ่ม OK

- REGISTER เพื่อกรอกข้อมูลส่วนตัวของผู้ใช้ซึ่งจะถูกเก็บเป็นข้อมูลไว้ที่ server
- ORDER เพื่อทำการสั่งอาหาร
- INFORMATION เป็นข้อมูลการใช้งานเพื่อช่วยให้ผู้ใช้งาน โปรแกรมนี้ใช้ได้ง่ายขึ้น

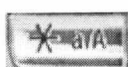
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 การพิมพ์ภาษา

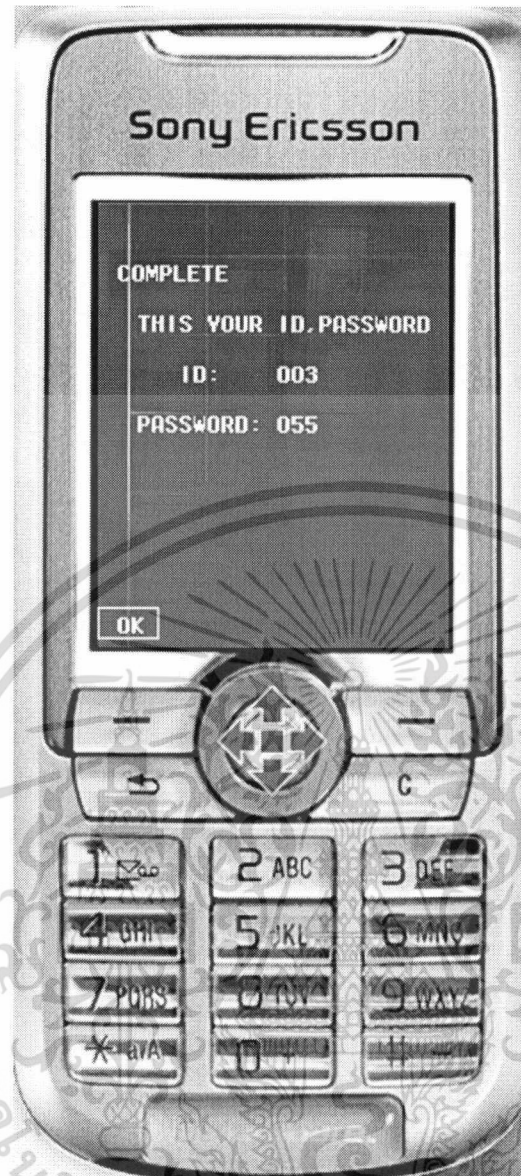
รูปที่ 4.4 การพิมพ์ตัวเลข

จากรูปที่ 4.3 เป็นหน้าจอในส่วนของการ REGISTER ผู้ใช้ต้องกรอกข้อมูลต่างๆ ให้ครบ โดยข้อมูลเหล่านี้จะถูกส่งไปยัง server เพื่อเก็บเป็นข้อมูล เราสามารถเปลี่ยน โหมด ภาษาหรือตัวเลขได้โดยกดปุ่ม



เมื่อกรอกข้อมูลเรียบร้อยแล้วให้กด SEND เพื่อส่งข้อมูลไปยัง Server

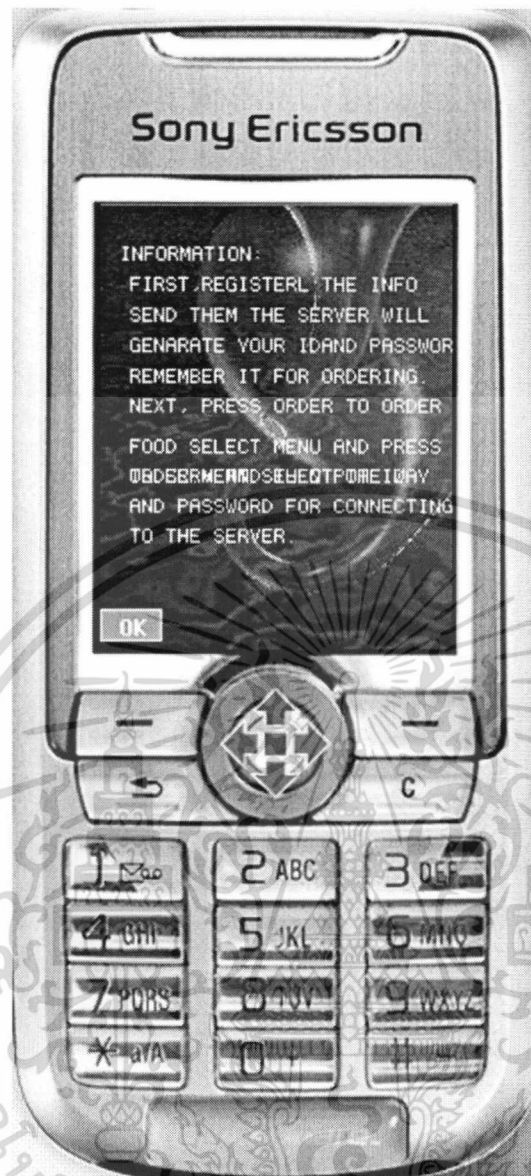
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Register complete

เมื่อ Server รับข้อมูลเรียบร้อยแล้วจะจัดเก็บข้อมูลลง Database แล้วทำการ Generate ค่า id และ password จากนั้นจะส่งข้อมูลกลับมาแสดงที่หน้าจอ ดังรูปที่ 4.5 ผู้ใช้ต้องจดจำ id password นี้ไว้เพื่อใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



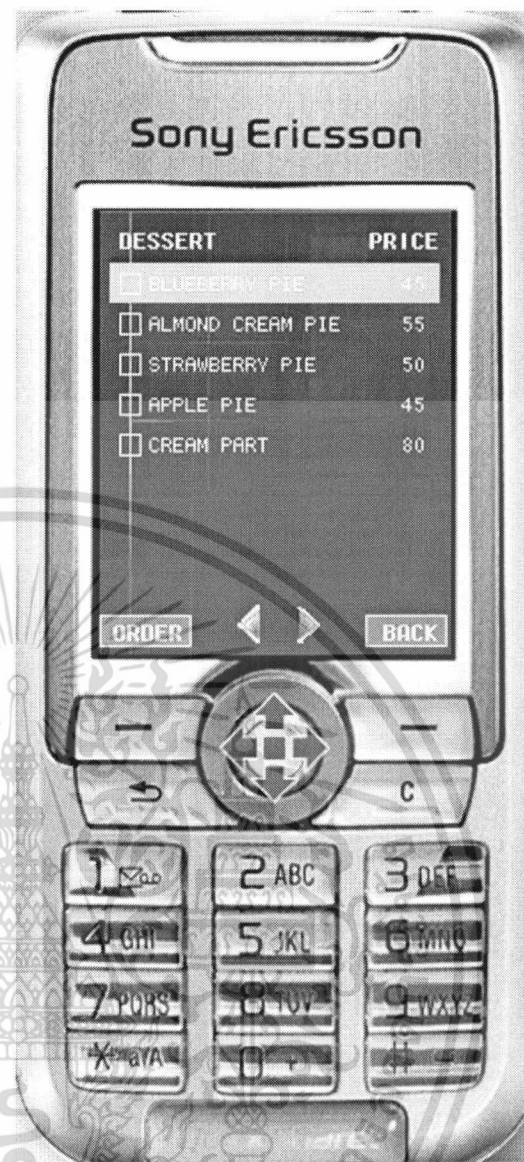
รูปที่ 4.6 หน้าจอ Information

จากรูปที่ 4.6 เป็นหน้าจอในส่วนของ INFORMATION ซึ่งจะบอกข้อมูลเกี่ยวกับการใช้งานโปรแกรมนี้เพื่อช่วยให้ผู้ใช้สามารถใช้งานได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 หน้าจอเมนูอาหาร



รูปที่ 4.8 หน้าจอเมนูอาหาร(ต่อ)

จากรูปที่ 4.7 และ 4.8 เป็นหน้าจอในส่วนของ ORDER เป็นหน้าจอแสดงรายชื่ออาหารและราคามีทั้งหมด 3 หน้า โดยสามารถเปลี่ยนหน้ารายการอาหารได้โดยกดปุ่มตามลูกศรที่ชี้ถ้าต้องการสั่งอาหารหรือดูรูปอาหารก็ให้กดปุ่ม SOFT KEY สีแดงก็จะแสดงหน้าจอรูปอาหารและถามจำนวนที่ต้องดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 หน้าจอแสดงรูปอาหาร

รูปที่ 4.9 หน้าจอแสดงรูปอาหารและถามจำนวนอาหารที่ต้องการเท่าไรผู้ใช้สามารถกด ขึ้น-ลง เพื่อระบุจำนวนอาหารที่ต้องการแล้วกด OK

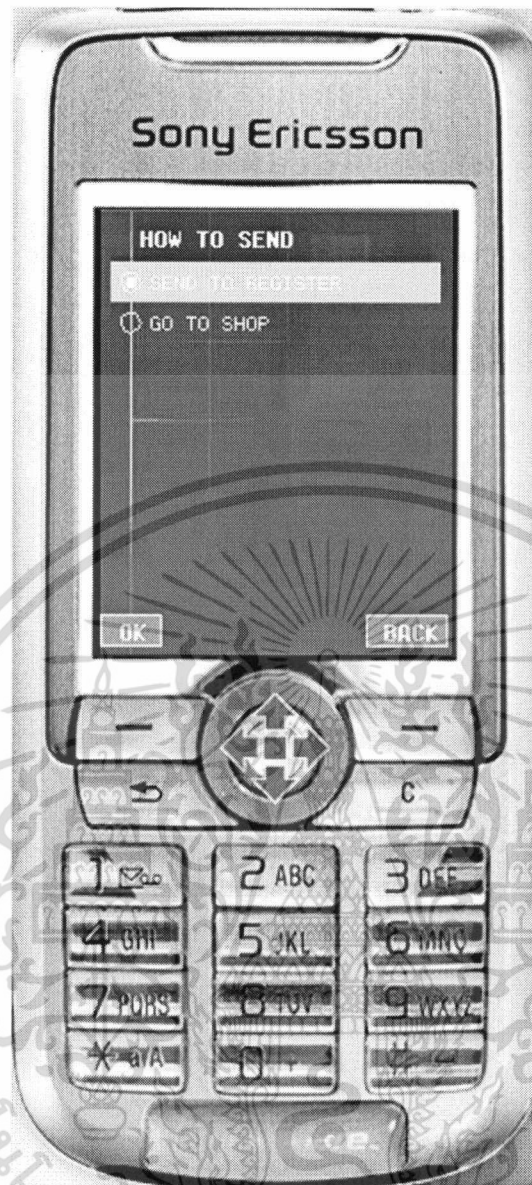
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าจอราคาอาหาร

รูปที่ 4.10 หน้าจอแสดงจำนวนอาหารทั้งหมดและรายการอาหารทั้งหมด หลังจากที่ผู้ใช้สั่งอาหารเรียบร้อยแล้วกดปุ่ม ORDER มาจากหน้าเดิมหากผู้ใช้พอใจกับราคาก็กด SUBMIT เพื่อทำงานต่อ แต่หากต้องการเลือกอาหารใหม่ก็ทำได้โดยกด BACK กลับไปยังหน้าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 หน้าจอระบบการส่งอาหาร

รูปที่ 4.11 เป็นหน้าจอแสดงให้ผู้เลือกใช้วิธีการส่งอาหาร โดยมีให้เลือก 2 วิธีคือ

- SEND TO REGISTER คือให้ส่งตามสถานที่ๆ REGISTER ไว้
- GO TO SHOP คือสั่งไว้เพื่อมารับประทานที่ร้าน โดยระยะเวลาที่มาถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 หน้าจอกรอก id, password

รูปที่ 4.13 หน้าจอระบุเวลา

ไม่ว่าจะเลือกวิธีการส่งอาหาร ใดๆ ก็จะทำให้เข้ามาสู่หน้าจอ ดังรูปที่ 4.12 เพื่อระบุ id และ password ของตนเองจากนั้นกดปุ่ม OK ถ้าเป็นการส่งอาหารแบบ GO TO SHOP ก็จะเข้าสู่หน้าจอ ดังรูปที่ 4.13 เพื่อบอกเวลาที่เข้ามาถึงร้าน แต่ถ้าเป็น SEND TO REGISTER ก็จะไม่มีหน้านี้ เมื่อกรอกทุกอย่างเรียบร้อยแล้วทำการกด OK ก็จะแสดงผลหน้าจอสุดท้าย ดังรูป 4.14 และ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงข้อมูลการสั่งแบบ Delivery

รูปที่ 4.15 แสดงข้อมูลการสั่งแบบจองโต๊ะ

เป็นหน้าจอแสดงผลข้อมูลทั้งหมดอีกครั้งเพื่อให้ผู้ใช้ทบทวนในการตัดสินใจ ว่าข้อมูลทุกอย่างถูกต้องหรือไม่สำหรับผู้ใช้ที่เลือกวิธีการส่งอาหาร แบบ SEND TO REGISTER จะแสดงผลหน้าจอ ดังรูปที่ 4.14 ผู้ใช้ที่เลือกวิธีการส่งอาหาร แบบ GO TO SHOP จะแสดงผลหน้าจอ ดังรูปที่ 4.15 หากผู้ใช้พอใจก็กด SEND เพื่อส่งข้อมูลไปยัง Server

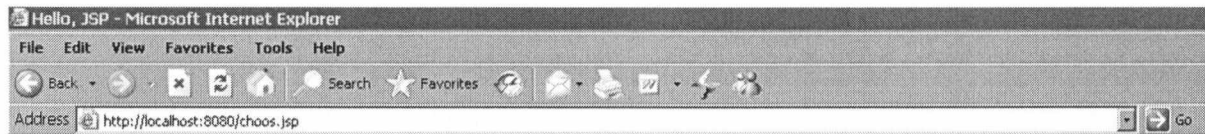
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การแสดงผลข้อมูลที่ร้าน



รูปที่ 4.16 แสดงผลหน้าจอหลักที่ร้านอาหาร เมื่อมีผู้สั่งอาหารเข้ามา โดยจะแสดงรหัสลูกค้าเรียงลำดับตามเวลาการสั่งอาหาร และบอกวิธีการสั่งอาหารของลูกค้าด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



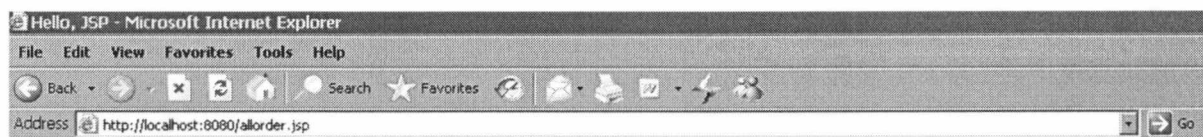
Food Order Application On Mobile

Delivery
 จองโต๊ะ
 back

รูปที่ 4.17 หน้าจอแสดงรายการสั่งอาหารทั้งหมด

รูปที่ 4.17 จากรูปที่ 4.16 เมื่อมีการกดปุ่ม OK ที่รายการอาหารที่สั่งทั้งหมด ก็จะเข้าสู่หน้านี้เพื่อให้ผู้ใช้เลือกว่าจะดูรายการอาหารที่ถูกสั่งมาทั้งหมด แบบ Delivery หรือ จองโต๊ะ เพื่อทำการพิมพ์ใบรายงานที่จะส่งไปให้พ่อครัวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Food Order Application On Mobile

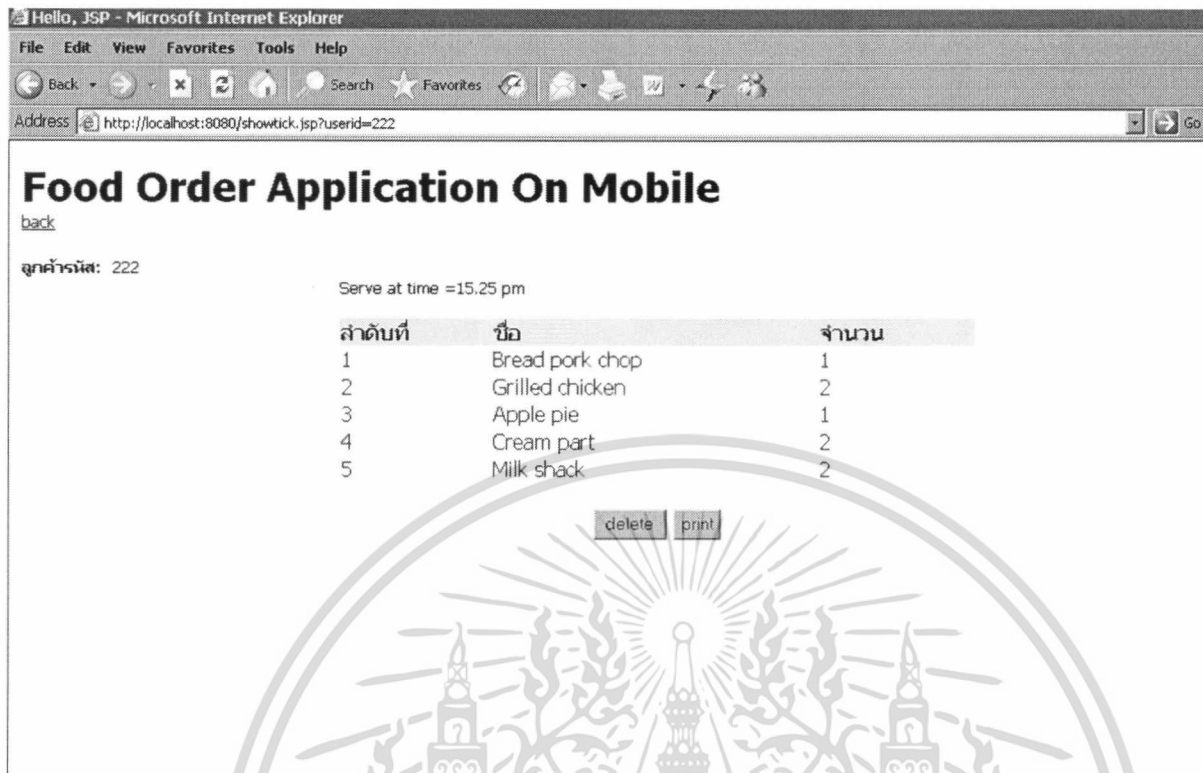
[back](#)

ลำดับที่	ชื่อ	จำนวน
1	Porkburger	2
2	Blueberry pie	2
3	Strawberry pie	1

รูปที่ 4.18 หน้าจอแสดงรายการสั่งอาหารแบบ Delivery

จากรูปที่ 4.17 เมื่อมีการกด Delivery จะแสดงหน้าจอรายการอาหารที่ยังไม่ได้ทำของลูกค้าที่สั่งอาหารแบบ Delivery (4.18) ทั้งหมดจากนั้นพนักงานจะสั่งพิมพ์ข้อมูล โดยกดปุ่ม print เพื่อนำรายงาน (4.21) ไปให้พ่อครัวจัดทำอาหารต่อไป หลังจากพิมพ์ข้อมูลเรียบร้อยแล้วจะต้องลบรายการอาหารทิ้งเพื่อเคลียร์หน้าจอหรือรับรายการอาหารที่จะเข้ามาต่อไป โดยกดปุ่ม delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าจอแสดงรายการสั่งอาหารแบบ จองโต๊ะ

จากรูปที่ 4.17 เมื่อมีการกด จองโต๊ะ จะแสดงหน้าจอรายการอาหารที่ยังไม่ได้ทำของลูกค้าที่สั่งอาหารแบบ จองโต๊ะ (4.19) โดยจะแสดงรหัสลูกค้าและเวลาที่ลูกค้าจะมาถึงร้านของลูกค้ารหัสนั้น จากนั้นพนักงานจะสั่งพิมพ์ข้อมูลโดยกดปุ่ม print เพื่อนำรายงาน (4.22) ไปให้พ่อครัวจัดทำอาหารต่อไป หลังจากพิมพ์ข้อมูลเรียบร้อยแล้วจะต้องลบรายการอาหารทิ้งเพื่อเคลียร์หน้าจอรับรายการอาหารที่จะเข้ามาต่อไป โดยกดปุ่ม delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hello, JSP - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address http://localhost:8080/showBill.jsp?userid=222 Go

Food Order Application On Mobile

[back](#)

ใบเสร็จลูกค้ารหัส: 222
Serve at time =15.25 pm

ใบรายการสินค้า

ชื่อ : TONGKAN
นามสกุล : DAOPRADUBWONG
ที่อยู่ : BANGKOK
เบอร์โทรศัพท์ : 014762367

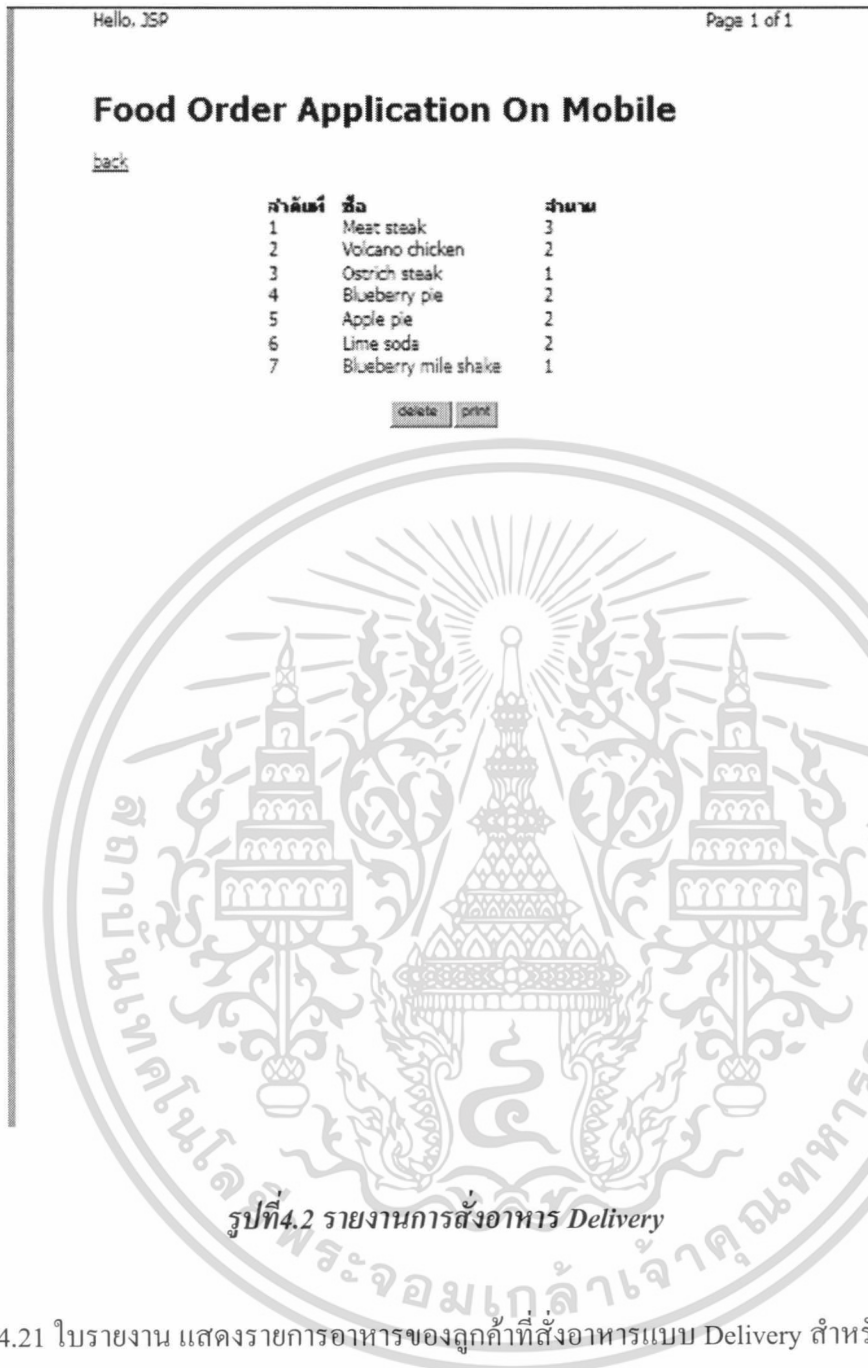
ลำดับที่	ชื่อ	ราคา/หน่วย	จำนวน	ราคารวม
1	Bread pork chop	70	1	70
2	Grilled chicken	80	2	160
3	Apple pie	50	1	50
4	Cream part	45	2	90
5	Milk shack	30	2	60
	รวม		8	430

delete print

รูปที่ 4.20 หน้าจอแสดงใบเสร็จลูกค้า

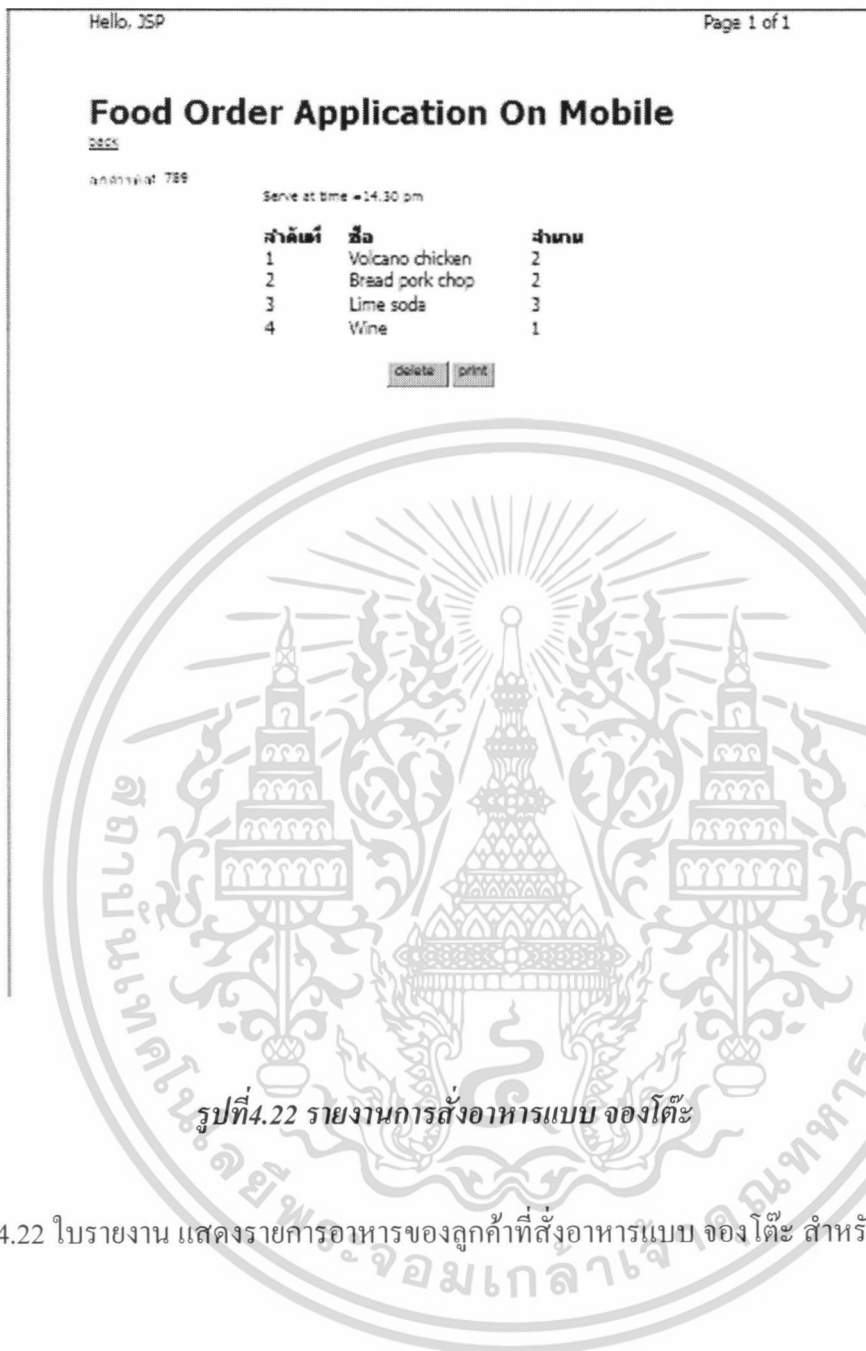
จากรูปที่ 4.16 เมื่อกรอกรหัสลูกค้าลงในช่อง **ใบเสร็จรับเงินลูกค้ารหัส**; แล้วกด OK ก็จะขึ้นหน้าจอใบเสร็จรับเงินลูกค้ารหัสตามที่กรอกไป กดปุ่ม print เพื่อพิมพ์ใบเสร็จรับเงินเพื่อให้ลูกค้าชำระเงิน หลังจากมีการชำระเงินเรียบร้อยแล้วก็จะต้องลบรายการใบเสร็จลูกค้า โดยกดปุ่ม delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 ใบรายงาน แสดงรายการอาหารของลูกค้าที่สั่งอาหารแบบ Delivery สำหรับพ่อครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 รายงานการสั่งอาหารแบบ จองโต๊ะ

รูปที่ 4.22 ใบรายงาน แสดงรายการอาหารของลูกค้าที่สั่งอาหารแบบ จองโต๊ะ สำหรับพ่อครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hello, JSP Page 1 of 1

Food Order Application On Mobile

๕๕๕


เบอร์โทรจองโต๊ะ: ๕๕๕

ใบรายการสินค้า

ชื่อ : SOMKEAT
 นามสกุล : USABAY
 ที่อยู่ : BANGKOK
 เบอร์โทรศัพท์ : 094532381

ลำดับที่	ชื่อ	ราคา/หน่วย	จำนวน	ราคารวม
1	Meat steak	95	3	285
2	Volcano chicken	85	2	170
3	Ostrich steak	90	1	90
4	Blueberry pie	40	2	80
5	Apple pie	50	2	100
6	Lime soda	30	2	60
7	Blueberry mile shake	45	1	45
	รวม		13	830

delete drink



รูปที่ 4.23 รายงานการใบเสร็จลูกค้า

รูปที่ 4.23 ใบรายงาน แสดงใบเสร็จรับเงินของลูกค้าที่สั่งอาหาร ออกเพื่อให้ลูกค้าชำระเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์และสรุป

สามารถนำเทคโนโลยีหลายๆ ชนิดมาประยุกต์ใช้ในการสร้างระบบสั่งอาหารผ่านมือถือได้อย่างประสบความสำเร็จ ทั้งส่วนของการสั่งอาหารที่มือถือและ การแสดงผลที่ร้านผ่านการเชื่อมต่อกับ Server ได้เป็นแนวคิดการสั่งอาหารแบบใหม่ซึ่งยังไม่มีใครทำ

5.2 ปัญหาที่เกิดขึ้นในการพัฒนาและแนวทางในการแก้ปัญหา

1. รูปที่จัดหามา นั้นไม่สวยเท่าที่ควรเนื่องจากไม่ได้มีการออกแบบรูปเอง แต่ใช้การโหลดมาจาก Internet ทำให้รูปไม่ค่อยตรง กับคอนเซ็ปต์มากนักหากต้องการรูปที่สวยงามและตรงคอนเซ็ปต์จริงๆ ควรจะมีการออกแบบรูปเอง

5.3 ข้อจำกัดของระบบ

1. เนื่องจากการเขียนโปรแกรมลงมือถือ นั้นจะจำกัดรุ่นของมือถือเนื่องจาก API และขนาดหน้าจอ และปุ่มกดของมือถือแต่ละรุ่น ไม่เหมือนกันจึงทำให้โปรแกรมนี้ใช้งานได้เฉพาะมือถือบางรุ่นเท่านั้น

2. ข้อจำกัดทางภาษาคือสามารถใช้ได้เฉพาะภาษาอังกฤษเท่านั้นเนื่องจากการแสดงผลภาษาไทยนั้นมีความยุ่งยากโดยจะต้องทำการวาดขึ้นเองหรือใช้รูปในการแสดงผล

5.4 แนวทางการพัฒนา

1. สามารถเพิ่มในส่วนของกราฟฟิกของหน้าจอแต่ละหน้าจอให้มีความทันสมัยและสวยงามขึ้นได้

2. สามารถเพิ่มรายการอาหารต่างๆ ได้โดยการจัดทำโปรแกรมขึ้นใหม่แล้วเพิ่มข้อมูลลง Database แล้วโหลดโปรแกรมใหม่ได้

3. อาจเพิ่มส่วนของการสั่งอาหารให้สามารถสั่งอาหาร แล้วส่งไปที่ไหนก็ได้ โดยไม่จำเป็นต้องส่งไปที่ที่ได้ REGISTER ไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] สาริต ชัยวิวัฒน์ตระกูล, “เก่ง JSP ให้ครบสูตร”, “ดวงกมลสมัย”, 2002, 1-464
- [2] กาญจนา ตันวิสุทธิ, “เขียนเกมและโปรแกรมบนมือถือ J2ME”, “ไอดีซีๆ”, 2004, 1-288
- [3] สุวัฒน์ ปุณณชัยยะ , ดัน คัตช์สุทธิวงษ์, สุพจน์ ปุณณชัยยะ, “เปิดโลก TCP/IP และโปรโตคอลของอินเทอร์เน็ต”, “โปรวิชั่น”, 2003, 1-59
- [4] <http://java.sun.com/>
- [5] <http://www.gamedev.net/>
- [6] <http://www.thaigamedevx.com/>
- [7] <http://www.midletzone.com/>
- [8] <http://www.narisa.com/>
- [9] <http://www.thaidev.com/>
- [10] <http://www.mysql.com/>
- [11] <http://www.developer.com/>
- [12] <http://www.pantip.com/tech/developer/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้