

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการเตือนการนัดหมายและการตรวจสอบปริมาณน้ำมันบน
โทรศัพท์มือถือ

APPOINTMENT AND FUEL CHECKING ALERT SYSTEM ON
MOBILE PHONE



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เลขที่.....

เลขทะเบียน.....593971

วัน, เดือน, ปี.....2549

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีคนนำไปใช้

**APPOINTMENT AND FUEL CHECKING ALERT SYSTEM ON
MOBILE PHONE**

The seal of King Mongkut's Institute of Technology Ladkrabang is a circular emblem. It features a central five-tiered umbrella (parasol) with a sunburst at the top. The umbrella is flanked by two ornate lamps. The entire design is set against a background of stylized floral and geometric patterns. The Thai text "สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง" is inscribed around the perimeter of the seal.

**JARUWAN TANOMSILP
NATTAWAN RUJIRANICHAPAT
WINWAROT SUNSANAPITTAYAKORN**

**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 ระบบการเตือนการนัดหมายและการตรวจสอบปริมาณน้ำมันบนโทรศัพท์มือถือ
APPOINTMENT AND FUEL CHECKING ALERT SYSTEM ON MOBILE
PHONE

ชื่อนักศึกษา นางสาวจรรวณ ถนอมศิลป์ 45050463

นางสาวณัฐวรรณ รุจิราภิภัทร 45050474

นางสาววินารท สันตะปะพิทยากร 45050518

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ดร.นวลสวาท หิรัญสกุลวงศ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ทำปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ปีการศึกษา 2548

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	อ.วีระชัย ต้นยะสิทธิ์	
กรรมการ	ผศ.ดร.กรรณ ประทุมรักษ์	
กรรมการและอาจารย์ที่ปรึกษา	ดร.นวลสวาท หิรัญสกุลวงศ์	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ระบบการเตือนการนัดหมายและการตรวจสอบปริมาณน้ำมันบนโทรศัพท์มือถือ	
ชื่อนักศึกษา	นางสาวจารุวรรณ ถนอมศิลป์	45050463
	นางสาวณัฐวรรณ รุจิราณิชาภัทร	45050474
	นางสาววินวาท ศันสนะพิทยากร	45050518
ปริญญา	วิทยาศาสตร์บัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2548	
อาจารย์ที่ปรึกษา	ดร.นवलสวาท ธีรบุญสกุลสงฆ์	

บทคัดย่อ

ปัญหาพิเศษนี้เป็นการพัฒนาระบบการนัดหมายและระบบตรวจเช็คน้ำมันบนโทรศัพท์มือถือ ระบบการนัดหมายมีฟังก์ชันและรูปแบบการแจ้งเตือนที่มากกว่าแอปพลิเคชันอื่น ส่วนระบบการตรวจเช็คน้ำมันนี้จะให้ข้อมูลแก่ผู้ใช่ว่ารถยนต์จะสามารถเคลื่อนที่ไปได้กี่กิโลเมตร โดยใช้น้ำมันที่เหลืออยู่ และยังสามารถแจ้งเตือนผู้ใช้ให้ทำการเติมน้ำมันได้อีกด้วย แอปพลิเคชันนี้พัฒนาโดยใช้ UML, J2ME (Java 2 Micro Edition) และเทคโนโลยี Record Store ซึ่งทำการทดสอบโปรแกรมบน Emulator และบนโทรศัพท์มือถือ Nokia 7210

Special Project Title	APPOINTMENT AND FUEL CHECKING ALERT SYSTEM ON MOBILE PHONE		
Students	Miss.Jaruwan Tanomsilp	45050463	
	Miss.Nattawan Rujiranichapat	45050474	
	Miss.Winwarot Sansanapittayakorn	45050518	
Degree	Bachelor of Science		
Department	Mathematics and Computer Science, Faculty of Science		
Programme	Computer Science		
Academic Year	2005		
Special Project Advisor	Dr.Nualsawat Hiransakolwong		

ABSTRACT

This special project is a development of Fuel Checking Alert and Appointment System on mobilephone. The Appointment System consists of functions and alert methods more than other applications. Fuel Checking Alert System helps user to forecast how many kilometers that car can go using the left fuel and can warn driver for refilling. This application is developed using UML and J2ME (JAVA 2 MICRO EDITION). It uses Record Store Technology to keep database. It was tested on emulator and Nokia 7210.

กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลุล่วงได้ คณะผู้จัดทำขอกราบขอบพระคุณ ดร.นวลสวาท หิรัญสกุลวงศ์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณ อ.วีระชัย ดันยะสิทธิ ประธานกรรมการสอบปัญหาพิเศษ และ ผศ.ดร. กรกช ประชุมรักย์ กรรมการสอบปัญหาพิเศษ ที่ให้ความกรุณาในการแก้ไขข้อบกพร่องต่าง ๆ ของปัญหาพิเศษ รวมทั้งอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้ ซึ่งทำให้ทางคณะผู้จัดทำสามารถนำความรู้ที่ได้รับมาแก้ไขปัญหาดัง ๆ ที่เกิดขึ้นจนปัญหาพิเศษนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณเจ้าหน้าที่ห้องคอมพิวเตอร์ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ทุกท่าน ที่ให้ความช่วยเหลือด้านทรัพยากรคอมพิวเตอร์และระเบียบวินัยอันดีแก่คณะผู้จัดทำ รวมทั้งผู้มีพระคุณทุกท่านที่มีได้เอื้อยนามไว้ ณ ที่นี้ด้วย

นอกจากนี้คณะผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ที่ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนกระทั่งการทำปัญหาพิเศษครั้งนี้สำเร็จได้ด้วยดี รวมทั้งเพื่อน ๆ และน้อง ๆ ทุกคนที่ได้ให้ความช่วยเหลือในด้านต่าง ๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2549

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 ขอบเขตของการศึกษา.....	2
1.4 ขั้นตอนการดำเนินงาน.....	3
1.5 แผนการดำเนินงาน.....	4
1.6 ประโยชน์ที่คาดว่าจะได้รับจากการศึกษา.....	5
1.7 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ.....	5

บทที่ 2 ความหมายและทฤษฎีที่เกี่ยวข้อง..... 6

2.1 ภาษาจาวา.....	6
2.2 ภาษา Java 2 Micro Edition (J2ME).....	7
2.2.1 โครงสร้างของ J2ME.....	8
2.2.2 การแบ่งกลุ่มอุปกรณ์ตาม Configuration.....	9
2.2.3 Profile.....	10
2.2.4 หน่วยความจำ.....	11
2.2.5 MIDlet.....	13
2.2.5.1 สถานะการทำงานของ MIDlet.....	13
2.2.5.2 ขั้นตอนการพัฒนา MIDlet.....	14
2.2.5.3 MIDlet Suite.....	14
2.2.5.4 Manifest.....	14
2.2.5.5 Packaging MIDlet Application.....	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.5.6 ตัวอย่างการพัฒนา MIDlet	17
2.2.5.7 MIDlet GUI.....	19
2.2.5.8 MIDlet RMS(Record Management System)	20
2.3 การเขียน Diagram โดยใช้ The Unified Modeling Language	22
2.3.1 Diagram ที่ใช้ในโครงการปัญหาพิเศษ	23
2.3.2 วิธีการสร้างและรันโปรแกรม	27
2.3.2.1 การสร้าง Project ใหม่	28
2.3.2.2 การเขียน Code และเก็บไว้ในไดเรกทอรี src.....	29
2.3.2.3 การคอมไพล์และตรวจสอบ	30
2.3.2.4 การทดสอบการทำงานบนอิมูเลเตอร์.....	30
บทที่ 3 สถาปัตยกรรมทางซอฟต์แวร์.....	32
3.1 Use Case Diagram.....	32
3.1.1 คำอธิบาย Use Case ของระบบบันทึกและเตือนการนัดหมาย.....	33
3.1.2 คำอธิบาย Use Case ของระบบการเตือนการใช้น้ำมัน.....	34
3.2 Class Diagram	35
3.2.1 คำอธิบายความหมายของ Class Diagram	35
3.3 Activity ของระบบบันทึกและเตือนการนัดหมายบนโทรศัพท์มือถือ.....	36
3.4 Activity ของระบบการเตือนการใช้น้ำมันบนโทรศัพท์มือถือ.....	41
3.5 การออกแบบ User Interface ของระบบ	45
บทที่ 4 ผลการทดลองและการวิเคราะห์ข้อมูล	47
4.1 ขั้นตอนการเข้าสู่โปรแกรม	47
4.1.1 โปรแกรมการเตือนการใช้น้ำมัน	47
4.1.2 โปรแกรมการเตือนการนัดหมาย.....	52

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะ	58
5.1 สรุปผล.....	58
5.2 ข้อเสนอแนะ	58
บรรณานุกรม	60
ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้ในการพัฒนา.....	61



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 Method ของ MIDlet.....	13
ก.1 แสดงรายละเอียดในโฟลเดอร์ต่างๆ	70



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 Java 2 Platform.....	7
2.2 โครงสร้างของ J2ME.....	8
2.3 ความสัมพันธ์ระหว่าง J2SE และ J2ME Configuration	10
2.4 MIDP Architecture	11
2.5 แสดงส่วนประกอบภายใน RecordStore.....	12
2.6 MIDlet Life Cycle	13
2.7 ขั้นตอนการพัฒนา MIDlet.....	14
2.8 แสดงการสืบทอดการทำงานจาก Displayable	20
2.9 โครงสร้างของ UML.....	23
2.10 สัญลักษณ์ Actor และ Use Case	23
2.11 ความสัมพันธ์ระหว่าง Actor กับ Use Case.....	24
2.12 โครงสร้างของ Class.....	24
2.13 ตัวอย่าง Class Diagram.....	25
2.14 สัญลักษณ์ Object Lifeline, Activation, Message, Return Message	25
2.15 ความสัมพันธ์ระหว่าง Actor User กับ Object Menu	26
2.16 ตัวอย่าง Activity Diagram	26
2.17 โปรแกรม KToolbar ใน J2ME Wireless Toolkit	27
2.18 หน้าจอการเริ่มต้นใช้งานโปรแกรม KToolbar.....	27
2.19 หน้าจอการสร้าง Project ใหม่.....	28
2.20 หน้าจอแสดงค่าเริ่มต้นของ Project.....	28
2.21 แสดงรายละเอียดโฟลเดอร์ต่างๆใน Project ตัวอย่างที่สร้างขึ้น.....	29
2.22 การเก็บ Code ไว้ในไดเรกทอรี src	29
2.23 หน้าจอแสดงการคอมไพล์และตรวจสอบเสร็จสิ้นสมบูรณ์	30
2.24 การเลือก Emulator.....	31
2.25 ผลลัพธ์ที่ได้จากการทำงานของ Helloworld.java.....	31
3.1 Use Case Diagram ของระบบบันทึกและเตือนการนัดหมายบน โทรศัพท์มือถือ	33
3.2 Use Case Diagram ของระบบการเตือนการใช้น้ำมันบน โทรศัพท์มือถือ	34
3.3 Class Diagram ของระบบบันทึกและเตือนการนัดหมายบน โทรศัพท์มือถือ	35
3.4 Activity Diagram การเพิ่มข้อมูล.....	36

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.5 Activity Diagram การแก้ไขข้อมูล.....	37
3.6 Activity Diagram การลบข้อมูล.....	38
3.7 Activity Diagram การแสดงข้อมูล.....	39
3.8 Activity Diagram การเตือน	40
3.9 Activity Diagram การเพิ่มข้อมูลสมรรถนะของรถ.....	41
3.10 Activity Diagram การเพิ่มข้อมูลการเติมน้ำมัน.....	42
3.11 Activity Diagram การเพิ่มข้อมูลการใช้รถประจำวัน.....	43
3.12 Activity Diagram การตั้งเตือนให้กรอกข้อมูล	44
3.13 Activity Diagram การลบข้อมูลที่บันทึกทั้งหมด	45
3.14 แสดง User Interface ของหน้าจอหลัก (MAIN MENU)	45
3.15 แสดง User Interface แสดงหน้าจอการนัดหมาย	46
3.16 แสดง User Interface แสดงหน้าจอการตั้งปลุก	46
4.1 แสดง User Interface แสดงหน้าจอการตั้งปลุก.....	47
4.2 หน้าจอหลักของโปรแกรมตรวจสอบปริมาณน้ำมัน	47
4.3 เลือกเมนูสมรรถนะรถ	48
4.4 หน้าจอแสดงให้กรอกรายละเอียดสมรรถนะรถ	48
4.5 หน้าจอแสดงเมื่อทำการกรอกรายละเอียดข้อมูลเรียบร้อยแล้ว	49
4.6 เลือกเมนูข้อมูลน้ำมัน	49
4.7 หน้าจอแสดงให้กรอกรายละเอียดข้อมูลน้ำมัน	49
4.8 เลือกเมนูข้อมูลการใช้รถ.....	50
4.9 แสดงหน้าจอให้กรอกรายละเอียดการใช้รถประจำวัน	50
4.10 แสดงผลลัพธ์หลังจากใส่รายละเอียดการใช้รถประจำวันเรียบร้อยแล้ว	50
4.11 เลือกเมนูตั้งเตือนใส่ข้อมูล	51
4.12 แสดงหน้าจอการตั้งเตือนให้กรอกรายละเอียดการใช้รถประจำวัน.....	51
4.13 แสดงหน้าจอเตือนให้ผู้ใช้ทำการกรอกข้อมูลการใช้รถประจำวัน	51
4.14 เลือกเมนูลบทั้งหมด.....	52
4.15 แสดงหน้าจอเมื่อเลือกเมนูลบทั้งหมด.....	52
4.16 แสดงหน้าจอหลัก	52
4.17 หน้าจอแสดง ตารางนัดหมายเก่า	53

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.18 หน้าจอสร้างการนัดหมาย	53
4.19 หน้าจอเลือกชนิดของการเตือน	54
4.20 หน้าจอสถานะการบันทึกพร้อมรายละเอียด	54
4.21 หน้าจอแสดงนัดหมายทั้งหมดเรียงตามลำดับ	54
4.22 หน้าจอปฏิทินรายเดือน	55
4.23 หน้าจอรายการหลัก	55
4.24 หน้าจอกรอกเวลาปลุก	56
4.25 ประเภทของการปลุก	56
4.26 หน้าจอแสดงสถานการณ์บันทึก	57
4.27 การปลุก	57
ก.1 การเตรียมพร้อมก่อนทำการติดตั้งซอฟต์แวร์	61
ก.2 ยอมรับเงื่อนไขในการติดตั้งซอฟต์แวร์ J2SE SDK	62
ก.3 การเลือกไฟเจอร์และกำหนดไคเร็คทอรี	62
ก.4 การติดตั้งไฟเจอร์	63
ก.5 เลือกไฟเจอร์และกำหนดไคเร็คทอรี	63
ก.6 ติดตั้ง Plug-in ให้กับบราวเซอร์	64
ก.7 การติดตั้งไฟเจอร์	64
ก.8 เสร็จสิ้นการติดตั้ง J2SE SDK	65
ก.9 เตรียมไฟล์ต่างๆก่อนติดตั้งซอฟต์แวร์	66
ก.10 หน้าจอต้อนรับสู่การติดตั้งซอฟต์แวร์	66
ก.11 ข้อตกลงในการติดตั้งซอฟต์แวร์ J2MEWTK	67
ก.12 โฟลเดอร์ที่ได้ติดตั้ง J2SDK	67
ก.13 โฟลเดอร์ที่ต้องการติดตั้ง WTK	68
ก.14 ตั้งชื่อโฟลเดอร์ที่ต้องการเก็บโปรแกรม	68
ก.15 ทำการติดตั้งซอฟต์แวร์	69
ก.16 เสร็จสิ้นการติดตั้งซอฟต์แวร์ J2MEWTK	69
ก.17 เตรียมไฟล์เพื่อทำการติดตั้ง	71
ก.18 หน้าจอ Introduction	71
ก.19 ข้อตกลงในการติดตั้งซอฟต์แวร์ Nokia 7210 Emulator	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ก.20 สอบถามการเป็นสมาชิก.....	72
ก.21 การเลือกวิธีการเชื่อมต่ออินเทอร์เน็ต.....	73
ก.22 กรอกข้อมูล Username และ Password	73
ก.23 กรอก Serial Number.....	74
ก.24 กำหนดโฟลเดอร์ที่ต้องการติดตั้ง.....	74
ก.25 รายละเอียดการติดตั้งซอฟต์แวร์.....	75
ก.26 เสร็จสิ้นการติดตั้งซอฟต์แวร์ Nokia 7210 Emulator	75



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการสื่อสารแบบไร้สายนั้นได้เข้ามามีอิทธิพลต่อชีวิตประจำวันของคนอย่างแพร่หลาย โดยจะเห็นได้จากบุคคลแทบจะทุกคนล้วนแต่ต้องมีโทรศัพท์มือถือพกติดตัวตลอดเวลา และจะเห็นได้จากสื่อโฆษณาหรือแม้แต่ตัวแทนจำหน่ายได้ทำการแข่งขันกันอย่างหนัก เพื่อที่จะเพิ่มยอดขายโทรศัพท์มือถือให้มากขึ้น จึงเห็นได้ว่า ไม่ว่าจะเป็นการติดต่อสื่อสารระหว่างบุคคล การนัดหมายหรือแม้แต่โปรแกรมช่วยเหลือต่างๆบนโทรศัพท์มือถือจะเข้ามามีบทบาทมากขึ้น

โดยซอฟต์แวร์ส่วนใหญ่ที่ติดตั้งบนโทรศัพท์มือถือที่ผู้ใช้เลือกใช้งานส่วนมากนั้นคือโปรแกรมช่วยเตือนความจำ หรือโปรแกรมนัดหมาย ผู้จัดทำจึงได้นำโปรแกรมนัดหมายมาพัฒนาให้มีความสามารถมากขึ้น เนื่องจากได้เล็งเห็นว่าโปรแกรกดังกล่าวยังมีความสามารถไม่เพียงพอตามที่ผู้ใช้งานต้องการ ความสามารถดังกล่าวนี้คือ ผู้ใช้สามารถตั้งวันและเวลาในการปลุกเพียงครั้งเดียวเพื่อใช้สำหรับวันที่ต้องการปลุกโดยไม่จำเป็นต้องตั้งใหม่ทุกครั้ง เป็นต้น

และนอกจากนี้ผู้จัดทำได้เล็งเห็นว่าในปัจจุบันราคของน้ำมันมีการเปลี่ยนแปลงอย่างรวดเร็ว ผู้ที่ใช้รถยนต์พาหนะที่ใช้น้ำมันเป็นเชื้อเพลิงในการขับเคลื่อนเกิดการไม่สะดวกในการประมาณหรือคำนวณค่าน้ำมันที่ใช้ไปของระยะทางที่ใช้ยานพาหนะที่ใช้ไปจึงได้จัดทำซอฟต์แวร์คำนวณการใช้น้ำมันบนโทรศัพท์มือถือขึ้นมา เพื่อช่วยในการคำนวณการใช้น้ำมันเพื่อช่วยประมาณระยะทางที่รถสามารถวิ่งได้ ประมาณจำนวนน้ำมันคงเหลือและสามารถทำการเตือนเมื่อมีปริมาณน้ำมันคงเหลือในถังจำนวนน้อย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อสร้างแอปพลิเคชันการเตือนการนัดหมายให้มีรูปแบบในการตั้งเตือนที่มีความหลากหลายให้มากกว่าแอปพลิเคชันที่มีอยู่ในปัจจุบัน

1.2.2 เพื่อสร้างแอปพลิเคชันการเตือนการใช้น้ำมันเพื่อให้มีการใช้งานโทรศัพท์มือถือในรูปแบบใหม่ๆเกิดขึ้น

1.2.3 เพื่อศึกษาภาษา Java บน Platform J2ME และเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือด้วยภาษา Java

1.2.4 เพื่อศึกษาวิธีการออกแบบด้วยภาษา UML และพัฒนา Mobile Application

1.2.5 เพื่อให้แอปพลิเคชันนี้เป็นแนวทางในการพัฒนา Mobile Application ใหม่ๆหรือเป็นแนวทางในการปรับปรุง Mobile Application เดิมที่มีอยู่ให้มีประสิทธิภาพและความสามารถใหม่ๆเพิ่มขึ้น

1.3 ขอบเขตของการศึกษา

โครงการปัญหาพิเศษนี้จะดำเนินการพัฒนาซอฟต์แวร์สำหรับเตือนการนัดหมายและเตือนการใช้น้ำมันให้สามารถทำงานได้บนโทรศัพท์มือถือซึ่งใช้โปรแกรมภาษา J2ME เป็นภาษาหลักในการพัฒนา โดยซอฟต์แวร์ที่พัฒนามีหน้าที่หลักดังนี้

1. สามารถบันทึกข้อมูลของการนัดหมาย
2. สามารถเรียกดูการนัดหมายที่บันทึกไว้ได้จากคำสั่งแสดง หรือ จากปฏิทินที่สามารถ-3. -
3. แสดงวันที่มีการนัดหมายได้ชัดเจน
4. สามารถกำหนดวันในการเรียกเตือนได้ โดยความสามารถพิเศษในการกำหนดเป็นช่วงวันได้ เช่น ปลุกกราชอาทิตย์ หรือ ปลุกเพียงเสาร์-อาทิตย์ ก็ได้
5. สามารถประมาณระยะทางที่รถสามารถวิ่งได้
6. สามารถทำการเตือนได้เมื่อมีจำนวนน้ำมันคงเหลือในถัง

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ระบุปัญหาและกำหนดขอบเขตของโครงการ

1.4.2 ศึกษาแนวทาง เทคนิค และเครื่องมือที่จำเป็นในการพัฒนาซอฟต์แวร์บน
โทรศัพท์มือถือ

1) JAVA 2 MICRO EDITION (J2ME)

2) MOBILEPHONE EMULATOR

1.4.3 ออกแบบซอฟต์แวร์ และส่วนติดต่อผู้ใช้ วิเคราะห์และเลือกอัลกอริทึมที่เหมาะสม
พัฒนาซอฟต์แวร์ตามที่ได้ออกแบบไว้

1.4.4 ทดสอบซอฟต์แวร์ที่พัฒนาขึ้นเพื่อสังเกตข้อผิดพลาด ทำการปรับปรุงและแก้ไขให้มี
ประสิทธิภาพ

1.4.5 สรุปและวิเคราะห์ปัญหาพร้อมทั้งทำเอกสารประกอบโครงการปัญหาพิเศษ

1.5 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ภาคเรียนที่ 1 (ปี 2548)								ภาคเรียนที่ 2 (ปี 2549)		
	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	มกราคม	กุมภาพันธ์	มีนาคม	
ระบุปัญหาและกำหนดขอบเขตของโครงการ											
ศึกษาแนวทางการ เทคนิค และเครื่องมือที่จำเป็นในการพัฒนาซอฟต์แวร์บนโทรศัพท์มือถือ											
ออกแบบซอฟต์แวร์ และส่วนติดต่อผู้ใช้ วิเคราะห์และเลือกอัลกอริทึมที่เหมาะสม											
พัฒนาซอฟต์แวร์ตามที่ได้ออกแบบไว้											
ทดสอบซอฟต์แวร์ที่พัฒนาขึ้นเพื่อสังเกตข้อผิดพลาด เพื่อทำการปรับปรุงและแก้ไขให้มีประสิทธิภาพมากขึ้น											
สรุปและวิเคราะห์ปัญหาพร้อมทั้งทำเอกสารประกอบโครงการปัญหาพิเศษ											

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่ควรนำออกนอกหน่วยงานโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่คาดว่าจะได้รับจากการศึกษา

- 1.6.1 เข้าใจวิธีการพัฒนาซอฟต์แวร์เพื่อใช้งานบนโทรศัพท์มือถือ
- 1.6.2 เข้าใจถึงโครงสร้างการทำงานของโทรศัพท์มือถือ
- 1.6.3 เกิดความชำนาญในการพัฒนาภาษา J2ME และการใช้เครื่องมือช่วยพัฒนาซอฟต์แวร์
- 1.6.4 เกิดทักษะในการเขียนและออกแบบซอฟต์แวร์
- 1.6.5 นำซอฟต์แวร์ที่ได้ทำการสร้างแล้วไปใช้งานจริงได้

1.7 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

- 1.7.1 โปรแกรมอิมูเลเตอร์ของโทรศัพท์มือถือ
- 1.7.2 Java 2 Standard Edition (J2SE SDK)
- 1.7.3 J2ME Wireless Toolkit (J2ME WTK)
- 1.7.4 โทรศัพท์มือถือ

บทที่ 2

ความหมายและทฤษฎีที่เกี่ยวข้อง

ตามขอบเขตของโครงการและหน้าที่ของซอฟต์แวร์ที่อธิบายในบทที่แล้ว การพัฒนาซอฟต์แวร์ของโครงการจึงต้องอาศัยเทคโนโลยีที่เกี่ยวข้องดังนี้

- ภาษาจาวาเวอร์ชัน J2ME
- MOBILEPHONE Emulator
- Java 2 Micro Edition Wireless Toolkit
- การเขียน Diagram โดยใช้ The Unified Modeling Language

2.1 ภาษาจาวา

ภาษาจาวา (Java Language) คือ ภาษาคอมพิวเตอร์ที่ใช้สำหรับสร้างโปรแกรมคอมพิวเตอร์ ถูกคิดค้นโดยบริษัท Sun Microsystems โดยภาษานี้มีจุดเด่นที่สามารถทำงานบนสถานะแวดล้อมได้ทุกระบบปฏิบัติการ โดยโปรแกรมนั้นจะต้องผ่านการคอมไพล์ (Compile) ด้วยคอมไพเลอร์จากซอร์สโค้ดที่สร้างจากภาษาจาวาจนได้ไบต์โค้ดออกมา

Sun Microsystems ได้ออกเวอร์ชันของ Java 2 ออกมา 3 รุ่น คือ

1. J2SE (Java 2 Standard Edition)

มีจุดประสงค์เพื่อการใช้งานบนเครื่อง Desktop ทั่วไป ใช้ในการสร้าง Application แบบ Standalone เช่น Java Application, Java Applet เป็นต้น

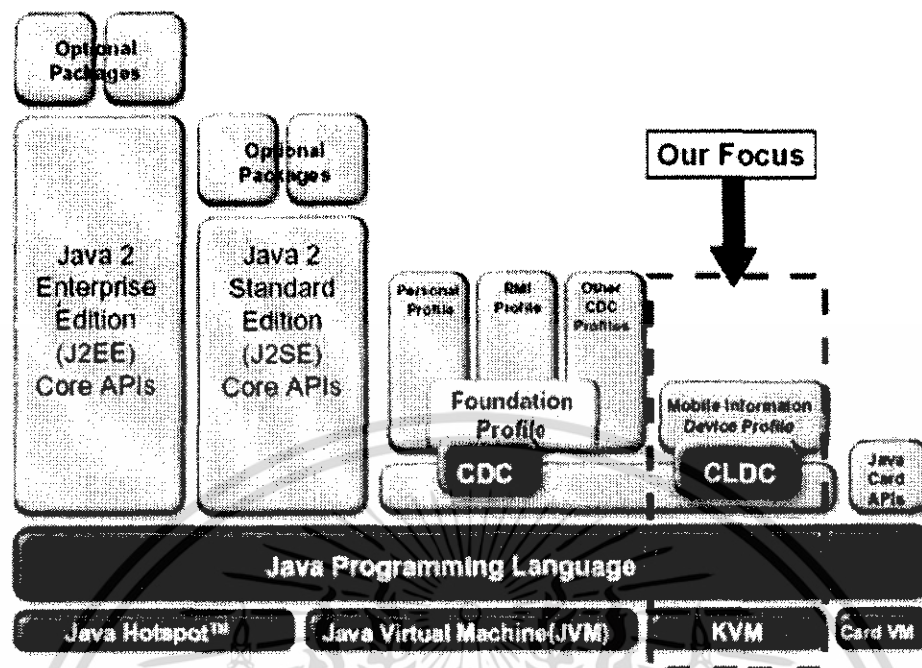
2. J2EE (Java 2 Enterprise Edition)

มีจุดประสงค์เพื่อการใช้งานในระบบงานใหญ่ๆ โดยขยายความสามารถจากตัว J2SE เพื่อให้สามารถรองรับการทำงานแบบ Server Side ให้สามารถรองรับการใช้งานจาก Client จำนวนมากได้ เช่น การสร้างซอฟต์แวร์ในรูปแบบ Java Servlet, JSP เป็นต้น

3. J2ME (Java 2 Micro Edition)

มีจุดประสงค์เพื่อใช้งานบนอุปกรณ์ขนาดเล็ก ซึ่งมีทรัพยากรจำกัด เช่น ขนาดหน่วยความจำ ความสามารถในการประมวลผล เป็นต้น โดยอุปกรณ์ที่มีทรัพยากรจำกัดเหล่านี้ก็ได้แก่ Palm, Pocket PC, โทรศัพท์มือถือ เป็นต้น โดยใน J2ME จะตัดฟังก์ชันที่ไม่จำเป็นออกไป เพื่อให้สามารถทำงานได้ในสถานะที่มีทรัพยากรจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 Java 2 Platform

ในบทนี้จะกล่าวถึงรายละเอียดของภาษา J2ME ซึ่งเป็นภาษาที่เหมาะสมกับการทำโครงการพิเศษนี้

2.2 ภาษา Java 2 Micro Edition (J2ME)

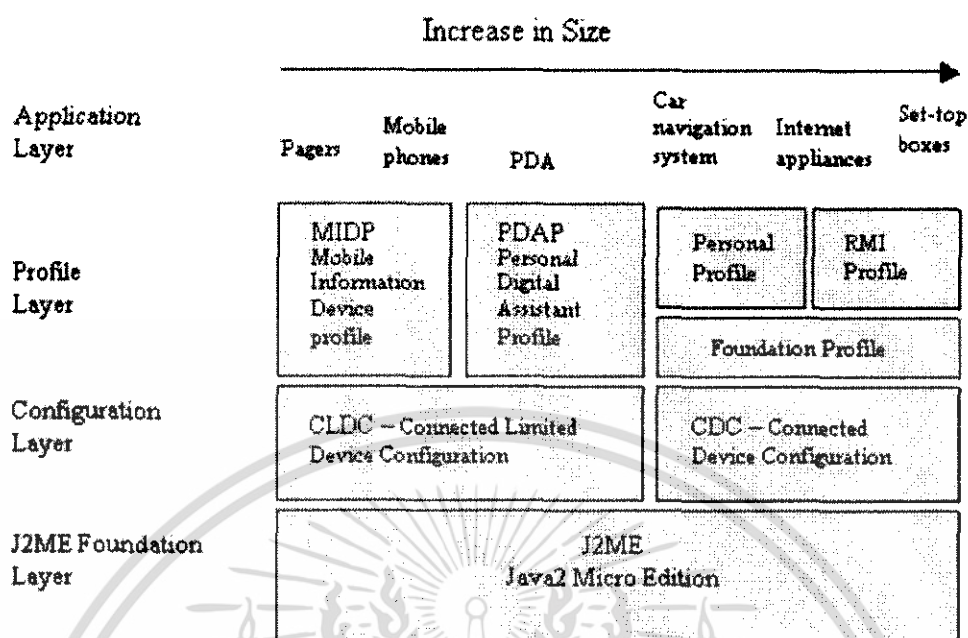
แต่ละเวอร์ชันของภาษาจาวานั้นจะมี Virtual Machine (VM) แตกต่างกันไปตามการใช้งาน เพื่อรองรับงานที่แตกต่างกัน จากรูปที่ 2.1 จะเห็นว่า Hotspot VM เป็น Virtual Machine ที่ใช้รองรับงานที่เป็นในลักษณะของ Client – Server ของ J2EE หรือเหมาะกับการรองรับการทำงานบน Desktop ของ J2SE

ส่วน Virtual Machine สำหรับ J2ME นั้นจะมีสอง Virtual Machine คือ Classic Virtual Machine (CVM) กับ Kilo Virtual Machine (KVM) ซึ่งเป็น Virtual Machine ที่มีขนาดเล็ก และใช้ทรัพยากรน้อยกว่า Virtual Machine ที่กล่าวมาข้างต้น โดย Virtual Machine ใน J2ME จะใช้ Configuration ในการแยก Virtual Machine

J2ME เป็น Platform หนึ่งของจาวา ซึ่งออกแบบมาเพื่อใช้กับอุปกรณ์ขนาดเล็ก และมีข้อจำกัดเรื่องทรัพยากร โครงสร้างของ J2ME จะแบ่งออกเป็น 3 Software Layer ซึ่งถูกสร้างบน native operating system ของแต่ละอุปกรณ์นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 โครงสร้างของ J2ME



รูปที่ 2.2 โครงสร้างของ J2ME

Layer 1 : J2ME Foundation Layer/Java Virtual Machine Layer

เป็น Layer สำหรับการสร้าง Java Virtual Machine ต้องปรับให้เหมาะสมตาม host operating system ของอุปกรณ์แต่ละชนิด และสนับสนุนแต่ละ J2ME configuration จากรูปที่ 2.1 จะเห็นว่า Virtual Machine สำหรับ J2ME คือ CVM (Classic Virtual Machine) และ KVM (Kilo Virtual Machine) โดยที่

- CVM ใช้สำหรับ 32-bit architectures และ memory ขนาดใหญ่
- KVM ใช้สำหรับ 16-bit or 32-bit architectures และมี memory จำกัด

Layer 2 : Configuration Layer

Configuration จะระบุตัว Class Libraries ตามกลุ่มของชนิดอุปกรณ์ภายใต้พื้นฐานความต้องการของ memory และหน่วยประมวลผล จากรูปที่ 2.2 Configuration ของ J2ME มี 2 แบบคือ CDC (Connected Device Configuration) และ CLDC (Connected Limited Device Configuration)

Layer 3 : Profile Layer

เป็น Layer ที่อยู่เหนือ Configuration โดยได้นิยาม Class Libraries ต่างๆไว้สำหรับ Market Segment แต่ละแบบ ตัวอย่างของ Profile ใน Layer นี้ ได้แก่ PDAP, MIDP, RMI และ Personal Profile

2.2.2 การแบ่งกลุ่มอุปกรณ์ตาม Configuration

Configuration เป็นตัวระบุ Virtual Machine และ Class Libraries พื้นฐานซึ่งจะมีเหมือนกันในทุกอุปกรณ์ที่ถูกจัดอยู่ในกลุ่มเดียวกัน โดย Configuration ใน J2ME จะแบ่งกลุ่มของอุปกรณ์ โดยใช้คุณสมบัติของหน่วยความจำ การแสดงผล และความสามารถในการประมวลผลเป็นตัวกำหนด ดังนี้

1. CDC (Connected Device Configuration)

คุณสมบัติของอุปกรณ์ในกลุ่ม ได้แก่

- มีหน่วยความจำตั้งแต่ 2-16 MB
- มีหน่วยประมวลผลขนาด 32 บิต เป็นอย่างน้อย
- ความเร็วในการเชื่อมต่อเครือข่ายค่อนข้างสูง

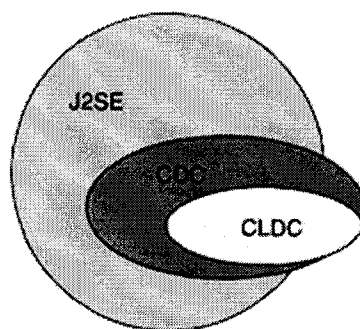
ตัวอย่างของอุปกรณ์ในกลุ่มนี้ ได้แก่ Pocket PC, TV Set-Top Box, Internet TV, Internet-enabled screenphones, highend communicators and car entertainment/navigation systems

2. CLDC (Connected Limited Device Configuration)

คุณสมบัติของอุปกรณ์ในกลุ่ม ได้แก่

- มีหน่วยความจำ 160-512 KB โดยควรมีหน่วยความจำแบบ Non-Volatile Memory (หน่วยความจำที่สามารถเก็บข้อมูลถาวรเอาไว้แม้ว่าจะปิดเครื่องหรือแบตเตอรี่หมดก็ตาม จึงสามารถนำหน่วยความจำประเภทนี้มาเก็บไคลบารี และ VM ซึ่งจำเป็นต้องใช้ทุกครั้งได้) อย่างน้อย 128 KB สำหรับไคลบารีของ CLDC และ Virtual Machine (VM) และควรมีหน่วยความจำแบบ Volatile Memory (ข้อมูลใดๆก็ตามที่เก็บใน Volatile Memory จะหายไปถ้ามีการเปิดเครื่องใหม่ ดังนั้นจึงถูกนำมาใช้เก็บข้อมูลชั่วคราวในขณะรันโปรแกรม) อย่างน้อย 32 KB สำหรับ VM ในการใช้รันโปรแกรม มีหน่วยประมวลผลขนาด 16-32 บิต ซึ่งมีความเร็วอย่างน้อย 25 MHz มีข้อจำกัดในการแสดงผล ความเร็วในการเชื่อมต่อเครือข่ายค่อนข้างต่ำ

ตัวอย่างของอุปกรณ์กลุ่มนี้ ได้แก่ low-end cell phone, two-way pager และ Palm OS handhelds โดย CLDC จะเป็นsubsetของ CDC ดังรูปที่ 2.3



รูปที่ 2.3 ความสัมพันธ์ระหว่าง J2SE และ J2ME Configuration

จากรูปที่ 2.3 ความสัมพันธ์ระหว่าง Class Libraries ของ J2SE และ J2ME Configuration สามารถอธิบายได้ว่า ฟังก์ชันต่างๆ ของ CDC และ CLDC ส่วนใหญ่จะสืบทอดมาจาก J2SE และมีส่วนที่เพิ่มเติม เพื่อให้เหมาะสมกับการทำงานบนอุปกรณ์ขนาดเล็กที่มีทรัพยากรจำกัด

2.2.3 Profile

Profile เป็นตัวกำหนดชุดฟังก์ชันการทำงานที่เหมาะสมกับงานแต่ละแบบ โดย Profile จะถูก Implement เหนือระดับ Configuration ดังรูปที่ 2.2 Profile จะประกอบด้วยชุด APIs ใช้งานในอุปกรณ์หลากหลายดังแสดงใน Application layer รูปที่ 2.2

Profile ที่สำคัญคือ Mobile Information Device Profile (MIDP) และ Foundation Profile (FP) โดย MIDP เป็น Profile ที่พัฒนามาบน CLDC ส่วน Foundation Profile จะเป็นของ CDC

Mobile Information Device Profile (MIDP)

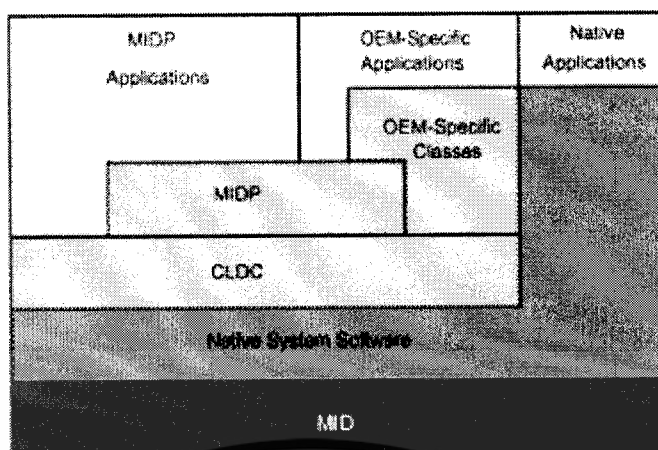
MIDP คือประเภทของอุปกรณ์ที่มีคุณสมบัติ เช่น มีส่วนแสดงผลขนาดเล็ก (min. 96×54 pixels), มี touch screen หรือ keypad, สามารถ connect mobile network ด้วย bandwidth ที่จำกัด โดย MIDP จะประกอบไปด้วย APIs ที่ทำหน้าที่ดังต่อไปนี้

- กำหนด และควบคุม Application
- แสดงผล text, graphics และตอบสนองต่อ user events
- จัดเก็บข้อมูลในฐานข้อมูลอย่างง่าย
- การเชื่อมต่อ Network จะทำได้โดยผ่าน subset ของ HTTP
- มี Timer notifications

โครงสร้าง MIDP มีความสัมพันธ์และการทำงานร่วมกับ CLDC, KVM ใน J2ME

Architecture ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 MIDP Architecture

โครงสร้างที่เห็นดังรูปที่ 2.4 แบ่งออกเป็น 5 Layer ดังนี้

- MID hardware layer ได้แก่ โทรศัพท์มือถือ หรือ pager
- Native system software layer คือส่วนที่เป็น Native OS library ซึ่งจัดเตรียมโดยผู้ผลิต
- KVM layer คือส่วนที่จัดเตรียม runtime environment สำหรับ Java Application
- CLDC layer คือส่วนที่จัดเตรียม GUI library, persistent storage library, networking library และ timer class ซึ่งส่วนนี้ทางผู้ผลิตอุปกรณ์แต่ละรายจะจัดเตรียม Class Libraries ซึ่งมีคุณสมบัติเฉพาะกับอุปกรณ์รุ่นนั้นๆ ได้เช่น Telephony, sharing data with native application เช่น build in calendar, address book เป็นต้น

2.2.4 หน่วยความจำ

ในโทรศัพท์มือถือมักจะใช้หน่วยความจำอยู่ 2 ประเภท ประเภทแรกใช้ในการทำงานของแอปพลิเคชัน อีกประเภทใช้สำหรับเก็บข้อมูลซึ่งจะเก็บได้คงทนถาวรกว่าหน่วยความจำประเภทแรก โดยผู้ใช้สามารถเก็บข้อมูลส่วนตัว ค่าเริ่มต้น หรือข้อมูลอื่นๆ ที่เปลี่ยนแปลงได้ในหน่วยความจำประเภทหลังนี้

โดยคำนิยามของ “ฐานข้อมูล” หมายถึง คลังข้อมูลที่เปลี่ยนแปลงแก้ไขได้ มักมีขนาดใหญ่ และต้องอาศัยชุดคำสั่ง สำหรับจัดการหรือเข้าถึงฐานข้อมูลนั้น J2ME MIDP มีระบบจัดการฐานข้อมูลแบบ Record ง่ายๆ เรียกว่า “Record Management System (RMS)”

โครงสร้างของ RMS ประกอบด้วย 2 ส่วน คือ ตารางในฐานข้อมูล เรียกว่า Record Store และ แต่ละรายการข้อมูลใน Record Store จะเรียกว่า “Record” ส่วนชุดคำสั่ง API สำหรับสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าถึง และจัดการกับเรคคอร์ดต่างๆ นั้นจะอยู่ในแพ็คเกจ javax.microedition.rms โดยตัวข้อมูลจริงจะเก็บแยกกันอยู่คนละ Record และ Record ที่มีข้อมูลเกี่ยวข้องกันจะเก็บรวมใน RecordStore ถ้าเทียบกับฐานข้อมูลที่ใช้กันบนเครื่องคอมพิวเตอร์แล้ว RecordStore ก็คือ ตาราง (Table) ส่วน Record คือข้อมูลแต่ละแถวในตารางนั่นเอง โดย MIDlet จะใช้งาน RecordStore ที่อยู่ภายใน MIDlet Suite เดียวกันได้นั่น

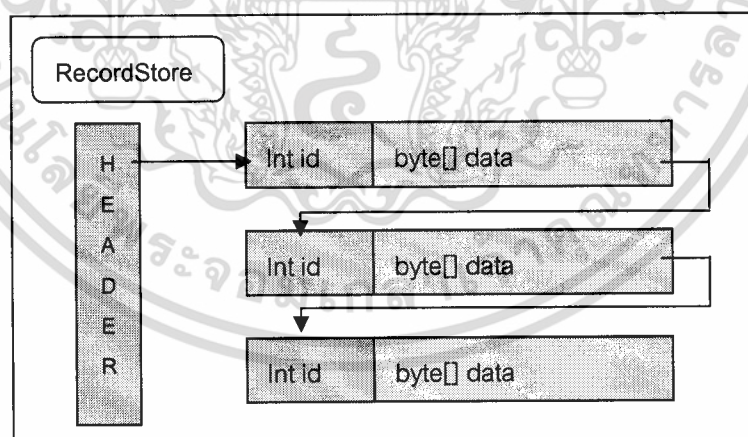
โครงสร้างของ RecordStore จะถูกแบ่งออกเป็นส่วนตัวและส่วนข้อมูล แต่ละ RecordStore จะต้องมีส่วนหัวอย่างน้อย 1 ข้อมูลเสมอ กรณีที่ระบบไม่มีหน่วยความจำเหลือพอสำหรับส่วนตัว ก็จะไม่สามารถสร้าง RecordStore เพิ่มได้ ข้อมูลส่วนตัวจะประกอบด้วย

- จำนวน Record ทั้งหมดเริ่มจาก 0
- หมายเลขรุ่นโดยจะเป็นจำนวนเต็มที่มีมากกว่า 0 และเพิ่มขึ้นทุกครั้งที่มีการแก้ไข

RecordStore

- เวลาครั้งล่าสุดที่มีการเปลี่ยนแปลง
- ค่า RecordID ต่อไปสำหรับการเพิ่ม Record เข้าใน RecordStore ซึ่งเพิ่มขึ้นทีละ 1

ส่วนข้อมูลใน RecordStore จะเก็บเป็น Record แต่ละ Record จะเชื่อมกันและมี pointer ชี้ไปยัง Record ถัดไป ซึ่ง Record แรกจะเชื่อมกับข้อมูลส่วนตัว ดังรูปที่ 2.5



รูปที่ 2.5 แสดงส่วนประกอบภายใน RecordStore

2.2.5 MIDlet

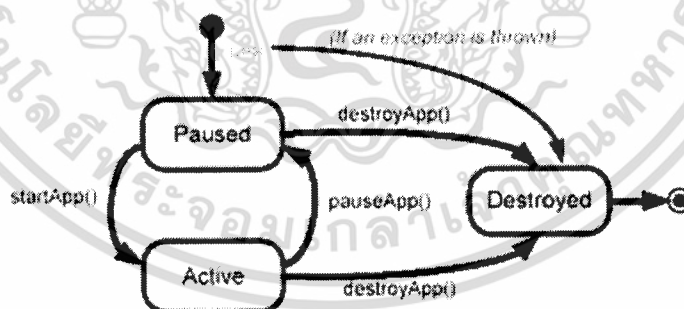
MIDlet เป็น Application ที่สร้างขึ้นสำหรับทำงานบนอุปกรณ์ประเภท MIDP ซึ่งลักษณะจะคล้ายกับ Java Applet ที่ทำงานบนบราวเซอร์ ซึ่งในการสร้าง MIDlet ทุกครั้งจะต้องมีการสืบทอด Class `javax.microedition.midlet.MIDlet` โดย Class นี้จะประกอบไปด้วย Method `startApp()`, `pauseApp()` และ `destroyApp()` ซึ่งใช้ในการควบคุมสถานะการทำงานของ MIDlet

2.2.5.1 สถานะการทำงานของ MIDlet

วงจรการทำงานของ MIDlet จะต้อง extend abstract MIDlet class ซึ่งมี 3 method ซึ่งแต่ละ MIDlet ต้องทำการ override เพื่อให้วงจรการทำงานเป็นไปอย่างสมบูรณ์ โดยทั้ง 3 method ได้แก่

ตารางที่ 2.1 Method ของ MIDlet

Method Name	การใช้งาน
StartApp	จัดการทรัพยากรในระบบเพื่อกำหนดการเริ่มต้นทำงาน
PauseApp	หยุดการทำงานชั่วคราวเพื่อย้ายการทำงานไปส่วนอื่นๆ
DestroyApp	ยกเลิกทรัพยากรที่จัดสรรและหยุดการทำงานของ MIDlet

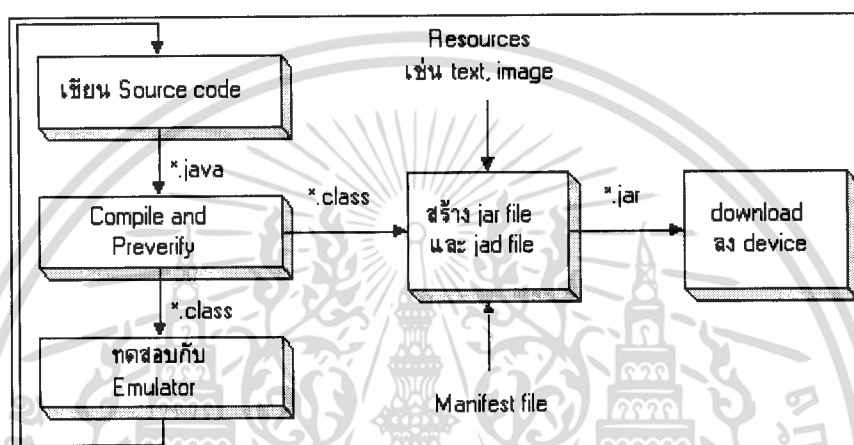


รูปที่ 2.6 MIDlet Life Cycle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.2 ขั้นตอนการพัฒนา MIDlet

1. เขียน source code
2. Compile Java code
3. Run Class ที่ได้จากการ Compile ผ่าน preverify tool
4. นำ Class ที่ผ่านการตรวจสอบจากการ verify และไฟล์ทรัพยากรทั้งหมดใส่ลงใน JAR file
5. ทดสอบโปรแกรมโดยใช้ emulator
6. นำ Application ติดตั้งลงในอุปกรณ์



รูปที่ 2.7 ขั้นตอนการพัฒนา MIDlet

2.2.5.3 MIDlet Suite

MIDlet Application เมื่อถูกสร้างขึ้นต้องนำมา package รวมกันเป็น JAR file 1 file โดยสามารถมี MIDlet Application หลายตัวใน JAR file เดียวกันได้ เราจะเรียกกลุ่มของ MIDlet Application นี้ว่า “MIDlet Suite” หลังจากทำ Packaging แล้วจะสามารถดาวน์โหลดและติดตั้งลงในอุปกรณ์ได้โดยผ่านทาง serial cable ต่อเข้าโดยตรงจากคอมพิวเตอร์ หรือจาก wireless network (OTA : Over-The-Air)

เมื่อติดตั้ง JAR file ลงบนอุปกรณ์แล้วจะปรากฏเมนูสำหรับแต่ละ MIDlet ที่อยู่ใน MIDlet Suite นั้นๆ ทำให้สามารถเลือกที่จะใช้งาน MIDlet แต่ละตัวแยกกันได้

2.2.5.4 Manifest

Manifest file เป็น file หนึ่งที่จะต้องถูก package อยู่ใน JAR file ของ MIDlet Suite มีเพื่อเป็นตัวบอกว่าใน JAR file นั้น ๆ มี content อะไร และประกอบด้วยข้อมูลอะไรบ้าง เช่น name, version, vendor ของ MIDlet Suite

ตัวอย่างของ attribute list ของ Manifest file

MIDlet-Name: HelloMidlet
 MIDlet-Vendor: MidletZone.com
 MIDlet-Version: 1.0
 MIDlet-1: HelloMidlet, /Icon.png, HelloMidlet
 MicroEdition-Configuration: CLDC-1.0
 MicroEdition-Profile: MIDP-1.0
 MIDlet-Data-Size: 0

2.2.5.5 Packaging MIDlet Application

MIDlet เมื่อสร้างขึ้นจะต้องนำมารวมกันทำเป็น package สำหรับดาวน์โหลดไปใช้งาน ซึ่งเรียกว่า การทำ Packaging ผลลัพธ์ของการทำ packaging จะเรียกว่า MIDlet Suite ซึ่งจะประกอบด้วย file 2 file ต่อไปนี้

1. JAR file : ประกอบด้วย class file ของ MIDlet (ที่ทำการคอมไพล์และ preverify แล้ว) และไฟล์ทรัพยากรต่าง ๆ เช่น ไฟล์รูปภาพ, audio file เป็นต้น

โดยปกติ JAR file ของ MIDlet Suite จะประกอบด้วย class file ทั้งหมดของ MIDlet application รวมไปถึงไฟล์ทรัพยากร และ Manifest file โดยที่

- ตัว class file จะต้องถูกทำ preverify ก่อน ส่วนไฟล์ทรัพยากรจะถูกใช้โดย MIDlet ตอน runtime
- แต่ละ MIDlet ที่อยู่ใน MIDlet Suite เดียวกันสามารถ share class file กันได้ ถ้าเราสร้าง MIDlet หลายๆ ตัวโดยทำเป็น package เดียวกันจะประหยัดขนาด package ที่ต้องดาวน์โหลดได้มากเพราะว่าสามารถใช้วิธีการ share file ได้
- ถ้าเราต้องใช้ Third party class library ภายใน MIDlet เช่น library สำหรับการคำนวณ floating point เป็นต้น เราจะต้อง include ตัว Third party class file ลงใน JAR file ด้วย ถ้าหากว่า device ที่ใช้งานไม่ได้ถูก preload ตัว class file นี้ไว้ใน device อยู่แล้วจากผู้ผลิต เราสามารถ package MIDlet ลงใน JAR file โดยใช้คำสั่งได้ ดังนี้

```
jar cvmf MANIFEST.MF HelloMidlet.jar HelloMidlet.class icon.png
```

2. JAD file : เป็น file ที่ทำหน้าที่เป็น "Application Descriptor" ลักษณะของ file format จะคล้าย Manifest file ที่อยู่ใน .jar file แต่ใช้คนละจุดประสงค์โดย Manifest file จะใช้สำหรับ packaging MIDlet Suite แต่ jad file จะใช้ตอนนำ MIDlet ลงอุปกรณ์โดยเฉพาะทำ OTA (Over-The-Air) deployment process

JAD file จะประกอบด้วย 7 attribute หลักคือ

MIDlet-Name
MIDlet-Version
MIDlet-Vendor
MIDlet-Jar-URL
MIDlet-Jar-Size
MicroEdition-Profile
MicroEdition-Configuration

ส่วนที่เป็น Optional ได้แก่

MIDlet-Description
MIDlet-Icon
MIDlet-Info-URL
MIDlet-Date-Size

ก่อน MIDlet Suite จะถูกดาวน์โหลดลงอุปกรณ์จะมีตัวที่เรียกว่า Application Management Software คอยตรวจสอบค่า attribute ใน jad file 7 ค่าแรก เพื่อดูว่าเหมาะสมกับอุปกรณ์นั้นหรือไม่ ตัวอย่างเช่นถ้าอุปกรณ์ไม่สนับสนุน version ของ profile ซึ่งอยู่ใน attribute "MicroEdition-Profile" ตัว JAR file ก็จะไม่ถูกดาวน์โหลด หรือถ้าขนาดของ JAR file ที่อยู่ใน MIDlet-Jar-Size ใหญ่เกินกว่าที่ device จะรองรับได้ก็จะไม่ถูกโหลดมา

ตัวอย่าง HelloMidlet.jad file

MIDlet-Name: HelloMidlet
MIDlet-Version: 1.0
MIDlet-Vendor: MidletZone.com
MIDlet-1: HelloMidlet, /Icon.png, HelloMidlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDlet-Data-Size: 0
 MIDlet-Description: Our First MIDlet Example
 MIDlet-Jar-Size:1510
 MIDlet-Jar-URL: HelloMidlet.jar
 MicroEdition-Configuration: CLDC-1.0
 MicroEdition-Profile: MIDP-1.0
 Language-Support: English
 Target-Device: Motorola Condor
 Display-Width: 140
 Display-Height: 100

2.2.5.6 ตัวอย่างการพัฒนา MIDlet

เป็นตัวอย่างการแสดงผลการเขียนอย่างง่ายซึ่งเป็นโปรแกรมแสดงข้อความว่า Hello Midlet world โดยในแต่ละบรรทัดจะมีคำอธิบายโปรแกรมและจะมีการแสดงผลของโปรแกรม

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

Public class HelloMidlet extends MIDlet implements CommandListener
{
    //ส่วนของการ Initialize ตัวแปร Display ของ MIDlet
    private Display midletDisplay;

    //Initialize ตัวแปรสำหรับ doneCommand
    private Command doneCommand;

    public HelloMidlet()
    {

        //สร้าง display จาก static display object โดยเรียก method getDisplay
        midletDisplay = Display.getDisplay(this);
    }
    
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//Initialize the doneCommand

doneCommand = new Command("DONE",Command.SCREEN,1);

}

/*สร้าง TextBox มีข้อความ Hello Midlet World และ command listener */

public void startApp()
{

//สร้าง TextBox มีข้อความ "Hello MIDlet World"
TextBox textbox = new TextBox("Hello Midlet", "Hello Midlet World!",256,0)

//เพิ่ม done Command ลงใน TextBox
textbox.addCommand(doneCommand);

//set ค่า command listener สำหรับ textbox ที่สร้างขึ้น
textbox.setCommandListener((CommandListener)this);

//set ค่า current display ไปที่ textbox screen
midletDisplay.setCurrent(textBox);

}

/*PauseApp ใช้เพื่อ suspend background activities และปล่อยการใช้งาน resource ให้กับ
device เมื่อไม่มีการใช้งาน MIDlet */

Public void pauseApp()
{}

/*DestroyApp ใหหยุดการทำงานของ background activities และปล่อย resource ให้กับ
device */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public void destroyApp(Boolean unconditional)
```

```
{}
```

```
/*commandAction method ใช้คู่กับ CommandListener เพื่อ handle done action */
```

```
Public void commandAction(Command command,Displayable screen)
```

```
//ถ้ามีการทำ doneCommand
```

```
If(command == doneCommand)
```

```
{
```

```
//ให้เรียก destroyApp method
```

```
destroyApp(false);
```

```
//Notify ไปยัง MIDlet platform ว่า MIDlet ทำงานจบแล้ว
```

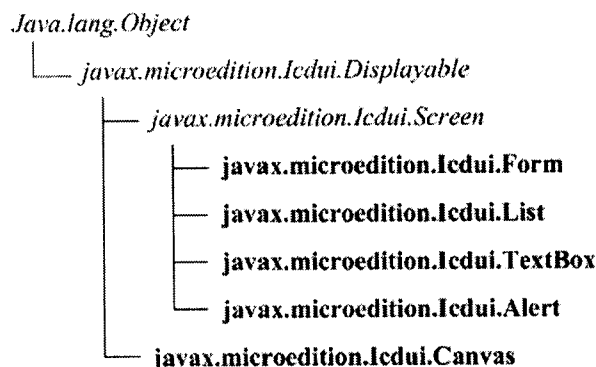
```
notifyDestroyed();
```

```
}
```

```
}
```

2.2.5.7 MIDlet GUI

MIDlet จะมีชนิดของ Displayable คือการจัดหน้าจอ โดยจะมีให้เลือกอยู่ 5 อย่าง คือ Canvas ,Form ,List ,Textbox และ Alert ซึ่งแต่ละรูปแบบจะมีวิธีการใช้งานที่แตกต่างกันออกไป โดย Canvas จะสืบทอดมาจาก Displayable โดยตรง ส่วนForm ,List ,TextBox และ Alert จะสืบทอดมาจาก Screen ซึ่ง Screen จะสืบทอดมาจาก Displayable อีกทีโดยจะแสดงดังรูป



รูปที่ 2.8 แสดงการสืบทอดการทำงานจาก Displayable

ใน Displayable จะมี Method ที่ใช้ในการจัดการ Command ดังนี้

1. void addCommand(Command cmd) ใช้ในการเพิ่ม Command เข้าไปในหน้านั้นๆ
2. void removeCommand(Command cmd) ใช้ในการลบ Command ออกจากหน้านั้นๆ
3. void setCommandListener(CommandListener l) ใช้ในการ Set Listener เพื่อบอกว่าถ้ามี Command Event เข้ามาให้ทำงานที่ใด

2.2.5.8 MIDlet RMS(Record Management System)

RMS เป็น set ของ class และ Interface ที่ช่วยในการรองรับระบบฐานข้อมูลอย่างง่ายที่ใช้สำหรับเก็บข้อมูลใน MIDlet data

RecordStore Class

RecordStore Class เป็นตัวแทนของ record store ซึ่งมันเป็น binary file และจะถูกเก็บในหน่วยความจำของ MIDP device ใน RecordStore Class จะมี method ดังนี้

openRecordStore()	ใช้สำหรับเปิด record store
closeRecordStore()	ใช้สำหรับปิด record store
deleteRecordStore()	ใช้สำหรับลบ record store
getName()	ใช้สำหรับดึงชื่อ record store
getNameRecordStore()	ใช้สำหรับดึงหมายเลขของ record ที่อยู่ใน record store
addRecord ()	ใช้สำหรับเพิ่ม record ใน record store
getRecord()	ใช้สำหรับดึง record ใน record store
deleteRecord()	ใช้สำหรับลบ record ใน record store
enumerateRecords()	ใช้สำหรับดึงกลุ่มข้อมูลใน record store

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RMS Interfaces

นอกจาก class RecordStore แล้ว javax.microedition.rms แพ็คเก็จยังมี interface ที่เป็นประโยชน์อีกหลายตัวดังนี้

RecordEnumeration	ใช้สำหรับเก็บกลุ่มของ record store เพื่อใช้สำหรับระบุถึงแต่ละ record ที่อยู่ใน record store
RecordComparator	ใช้เปรียบเทียบระหว่าง 2 record
RecordFillter	ใช้สำหรับเลื่อน record ไปยัง record ที่ต้องการ
RecordListener	ใช้สำหรับรอฟังเหตุการณ์ เมื่อมีการเพิ่ม เปลี่ยนหรือลบ record

ตัวอย่างการใช้งาน RMS

การทำงานของ record stores มีลักษณะคล้ายกับการทำงานของฐานข้อมูลซึ่งมีการเปิดและการเปิด record store อีกทั้งยังมีการเพิ่ม,ลบและดึงข้อมูลจาก record ซึ่งจะแสดงตามตัวอย่างต่อไปนี้

1. การเปิด Record store

ทุก MIDlet ที่ใช้ RMS ต้องเปิด record store โดยเรียกใช้ method openRecordStore() เราสามารถสร้าง record store ใหม่และเปิดขึ้นมาใช้งานดังตัวอย่างนี้โดยจะสร้าง record store ใหม่ชื่อว่า myDBfile.db โดยการตั้งชื่อของ record store จะต้องไม่เกิน 32 ตัวอักษร

```
RecordStore db = RecordStore.openRecordStore("myDBfile",true);
```

2. การเพิ่ม Record ใหม่

การเพิ่ม record ใหม่ที่ record store โดยเรียกใช้ method addRecord()

```
int id = 0;
```

```
try{
```

```
id = recordStore.addRecord(bytes,0,bytes,length);
```

```
}
```

```
catch(RecordStoreException e){
```

```
e.printStackTrace();
```

```
}
```

3. การเรียกใช้ Record

การดึง record จาก record store โดยเรียกใช้ method getRecord()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

byte[] recordData = null;
try{
    recordData = recordStore.getRecord(id);
}
catch(RecordStoreException e){
    e.printStackTrace();
}

```

4. การลบ Record

การลบ record ที่ต้องการนั้นเราต้องรู้ ID ของ record ที่เราต้องการลบ เมื่อเรารู้ ID แล้วเราสามารถลบ record ได้โดยใช้ method deleteRecord()

```

try{
    recordStore.deleteRecord(id);
}
catch(RecordStoreException e){
    e.printStackTrace();
}

```

5. การปิด Record

เมื่อใช้งานเสร็จแล้วเราก็ต้องปิด record โดยใช้ method closeRecordStore()

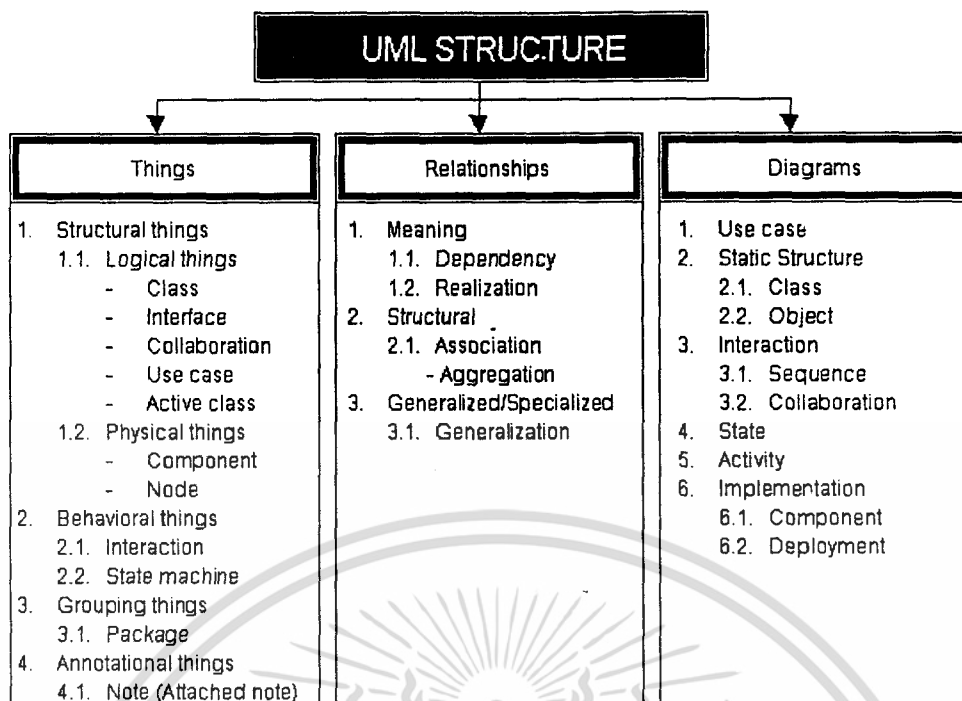
```

recordStore.closeRecordStore();

```

2.3 การเขียน Diagram โดยใช้ The Unified Modeling Language

The Unified Modeling Language (UML) คือ สัญลักษณ์มาตรฐานที่ใช้ในวิธีการออกแบบการพัฒนาเชิงวัตถุ (Object-oriented Design Methodology) เป็นภาษาที่แสดงให้เห็นถึงรายละเอียดโครงสร้างต่างๆของระบบที่จะพัฒนา โดยแสดงในรูปแบบของ Diagram เพื่อให้เป็นสื่อระหว่างระบบกับผู้อ่าน ตรวจสอบความถูกต้องของระบบ จำลองการทำงาน และในบางครั้งให้เห็นเพื่อที่ตัดการทำงานบางอย่างที่ซับซ้อนออกได้ด้วย



รูปที่ 2.9 โครงสร้างของ UML

2.3.1 Diagram ที่ใช้ในโครงการปัญหาพิเศษ

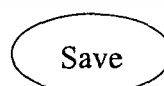
- 1) **Use Case Diagram** : Diagram แสดงให้เห็นพฤติกรรมของระบบ และความสัมพันธ์ระหว่างแหล่งข้อมูลภายนอกกับพฤติกรรมต่างๆที่เกิดขึ้นในระบบ เรียกว่าเป็น Diagram ที่ทำหน้าที่รวมความต้องการของระบบ

ลักษณะพิเศษของ Use Case Diagram

- เป็นเทคนิคในการสร้างแบบจำลองเพื่อใช้อธิบายหน้าที่ของระบบใหม่หรือระบบปัจจุบัน
- กระบวนการสร้าง Use case เป็นแบบ Iteration
- ความต้องการของระบบจะได้จาก ลูกค้า/ผู้ใช้ และผู้พัฒนาระบบ
- องค์ประกอบจะมี Use case, Actor, Use case Relation และระบบ



User

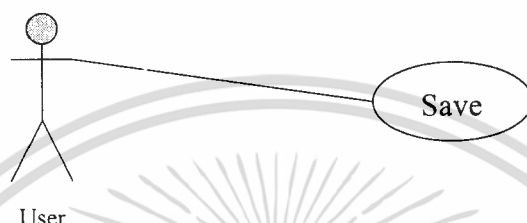


รูปที่ 2.10 สัญลักษณ์ Actor และ Use Case

Actor: แหล่งข้อมูลภายนอก เวลาเขียนต้องมีชื่อกำกับ

Use Case : ส่วนพฤติกรรมต่างๆ ที่เกิดในระบบ โดยสัญลักษณ์จะเขียนพฤติกรรมกำกับไว้ในแต่ละ Use Case

ความสัมพันธ์ระหว่าง Actor กับ Use Case จะใช้เส้นตรงแสดงว่ามีการรับ-ส่งข้อมูลกัน



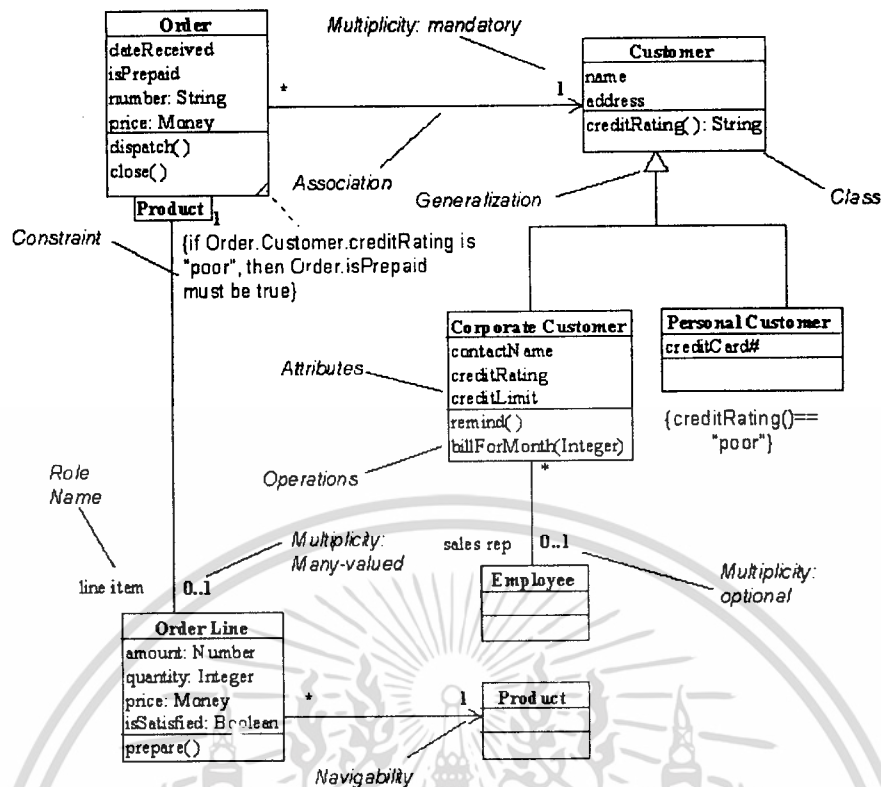
รูปที่ 2.11 ความสัมพันธ์ระหว่าง Actor กับ Use Case

- 2) **Class Diagram :** Diagram อธิบายโครงสร้างของ Class ประกอบด้วย Class และความสัมพันธ์ต่างๆ ระหว่าง Class เช่น Dependency, generalization, association เป็นต้น Class Diagram ยังสามารถทำการแสดงรายละเอียดภายใน Class แต่ละ Class ได้ว่ามี Method อะไรบ้าง Field และ Attribute เป็นอย่างไร โดยมีลักษณะดังรูปที่ 2.10

ชื่อ Class
Attribute : type
Method(argument) : type

รูปที่ 2.12 โครงสร้างของ Class

การอธิบายลักษณะความสัมพันธ์ของ Class จะใช้เส้นตรง โดยแต่ละเส้นจะมีชื่อความสัมพันธ์ และตัวบอกลักษณะความสัมพันธ์ (one-to-one, many-to-one, many-to-many)



รูปที่ 2.13 ตัวอย่าง Class Diagram

- 3) **Sequence Diagram** : Diagram ซึ่งแสดงปฏิสัมพันธ์(Interaction) ระหว่าง Object ตามลำดับของเหตุการณ์ที่เกิดขึ้น ณ เวลาที่กำหนด message ที่เกิดขึ้นระหว่าง class จะสามารถนำไปสู่การสร้าง method ใน class ที่เกี่ยวข้องได้



รูปที่ 2.14 สัญลักษณ์ Object Lifeline, Activation, Message, Return Message

Object Lifeline : Object ที่มีส่วนเกี่ยวข้องกับระบบในช่วงเวลาหนึ่ง

Activation : ช่วงเวลาที่ Object หรือ Actor มีส่วนเกี่ยวข้องกับระบบ

Message : การติดต่อระหว่าง Object เพื่อให้ได้ข้อมูลที่ต้องการ

Return Message : การติดต่อระหว่าง Object เพื่อสำหรับการส่งค่ากลับ

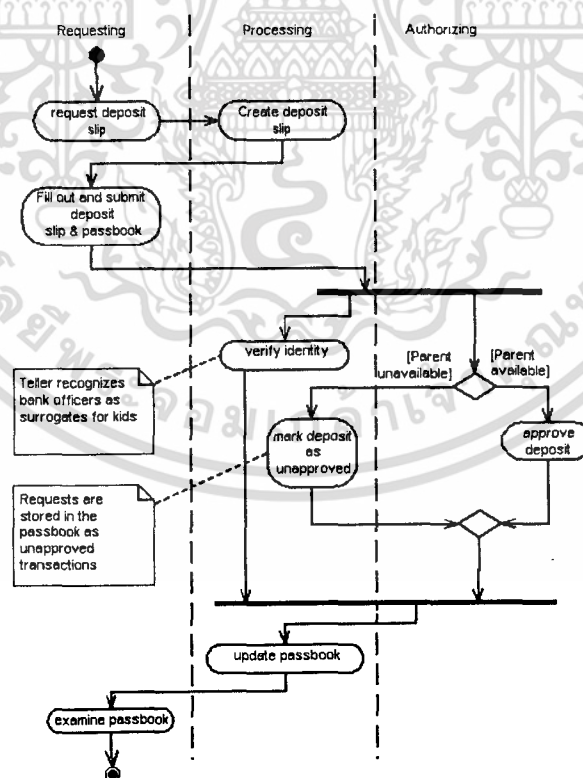
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่าง Actor กับ Object จะใช้สัญลักษณ์จากรูปที่ 2.12 มาเขียนรวมกัน



รูปที่ 2.15 ความสัมพันธ์ระหว่าง Actor User กับ Object Menu

4) **Activity Diagram** : Diagram ซึ่งอธิบายนั้นจะเป็นกิจกรรมที่แสดงการปฏิบัติงาน เน้นหนักไปที่การไหลของกิจกรรมในกระบวนการหนึ่งๆ

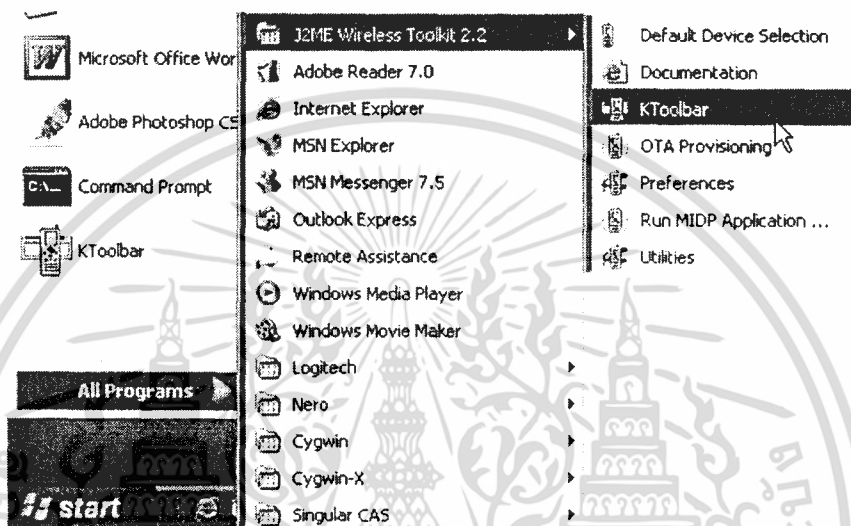


รูปที่ 2.16 ตัวอย่าง Activity Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

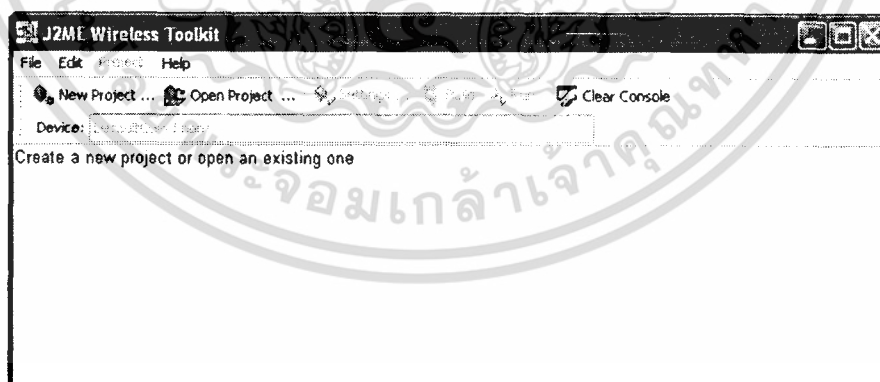
2.3.2 วิธีการสร้างและรันโปรแกรม

ก่อนอื่นต้องเปิดโปรแกรม KToolbar ซึ่งเป็นเครื่องมือที่ช่วยเพิ่มความสะดวกในการเขียนโปรแกรมด้วย J2ME ได้ง่ายขึ้นเป็นเครื่องมือที่ติดตั้งมาพร้อมกับ J2ME WTK ซึ่งใช้สำหรับการพัฒนาซอฟต์แวร์ ซึ่งประกอบไปด้วยการคอมไพล์, การทำแพ็คเกจ และการรันโปรแกรม โดยการใช้งานโปรแกรม KToolbar นั้นสามารถทำการเรียกผ่านทาง Command Line หรือในเมนู Start > Programs > J2ME Wireless Toolkit 2.2 > KToolbar



รูปที่ 2.17 การเรียกใช้โปรแกรม KToolbar ใน J2ME Wireless Toolkit

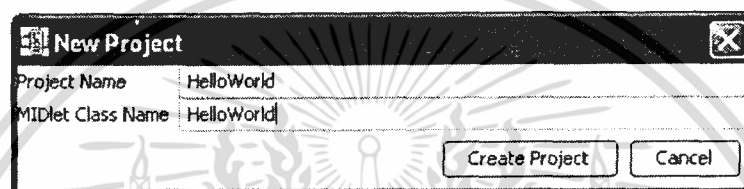
เมื่อทำการเปิดโปรแกรม KToolbar ขึ้นมาจะเริ่มต้นการทำงาน จะแสดงหน้าจอให้เห็นดังรูปที่ 2.41



รูปที่ 2.18 หน้าจอการเริ่มต้นใช้งานโปรแกรม KToolbar

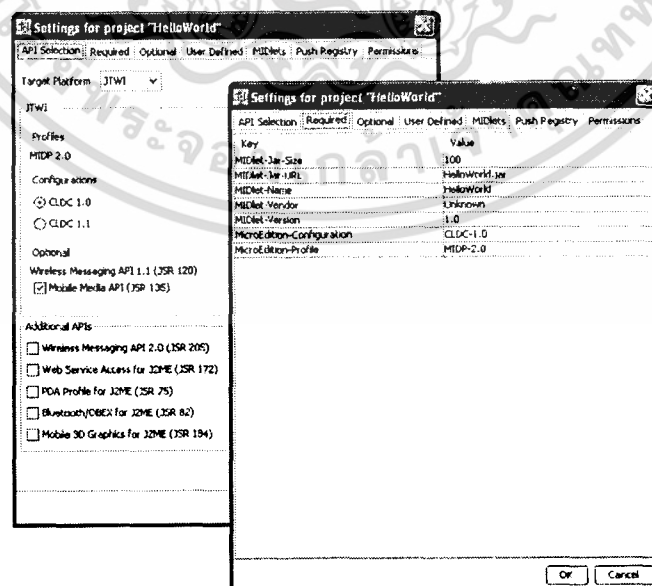
2.3.2.1 การสร้าง Project ใหม่

- 1) เมื่อทำการเปิดโปรแกรม KToolbar ขึ้นมาแล้วให้คลิกที่ปุ่ม New Project เพื่อทำการสร้าง Project ขึ้นมาใหม่
- 2) ตั้งชื่อ Project ในช่อง Project Name
- 3) กำหนดชื่อคลาสในช่อง MIDlet Class Name
- 4) คลิกปุ่ม Create Project จะมีหน้าต่างแสดงรายละเอียด Project ปรากฏขึ้นมา ในหน้าต่างนี้เราสามารถเลือกได้ว่า MIDlet ของเราจะใช้ CLDC เวอร์ชันใด



รูปที่ 2.19 หน้าจอการสร้าง Project ใหม่

- 5) คลิก Tab Required เพื่อดูรายละเอียดอื่นๆที่จำเป็นว่าถูกต้องหรือไม่ ซึ่งสามารถเปลี่ยนค่าต่างๆเหล่านี้ได้
- 6) คลิกปุ่ม OK เพื่อปิดหน้าต่างนี้ลงไปในเนื่องจากโดยปกติเราไม่จำเป็นต้องแก้ไขค่าเหล่านี้



รูปที่ 2.20 หน้าจอแสดงค่าเริ่มต้นของ Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

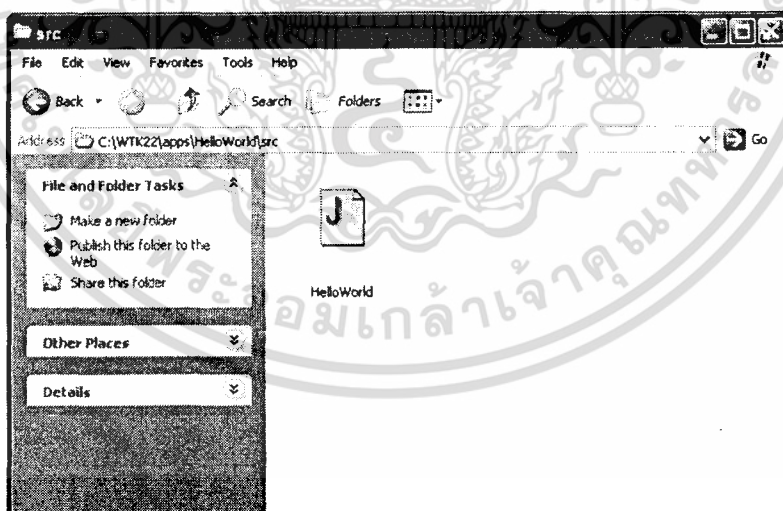
- 7) ที่หน้าจอหลักจะแสดงรายละเอียดเกี่ยวกับ Project ว่าจะต้องเก็บ Code, File Resource และ File Library ไว้ที่ใด



รูปที่ 2.21 หน้าจอแสดงรายละเอียดโฟลเดอร์ต่างๆ ใน Project ตัวอย่างที่สร้างขึ้น

2.3.2.2 การเขียน Code และเก็บไว้ในไดเรกทอรี src

ขั้นตอนนี้นำ Code ที่เขียนไว้แล้ว มาเก็บไว้ในไดเรกทอรี src ซึ่งแสดงไว้ที่หน้าต่างหลัก (C:\WTK22\apps\HelloWorld\src) ดังรูปที่ 2.45



รูปที่ 2.22 การเก็บ Code ไว้ในไดเรกทอรี src

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.3 การคอมไพล์และตรวจสอบ

การคอมไพล์โปรแกรมสามารถทำได้โดยคลิกปุ่ม Build ที่หน้าจอหลักของ J2ME WTK เพื่อให้ J2ME WTK คอมไพล์ไฟล์ .java (จากตัวอย่างคือ HelloWorld.java) เพื่อให้ได้ไฟล์ .class ออกมา (จากตัวอย่างจะได้ HelloWorld.class) และตรวจสอบไฟล์ .class ที่ได้ต่อไป ซึ่งจะมีหน้าต่างแสดงสถานะของการทำงานปรากฏขึ้นมา

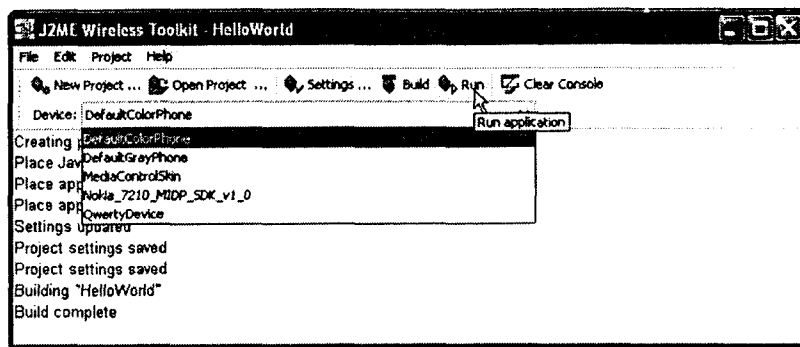
เมื่อเสร็จสิ้นและไม่มีข้อผิดพลาดของโปรแกรมจะมีข้อความว่า Build Complete แสดงขึ้นมา



รูปที่ 2.23 หน้าจอแสดงการคอมไพล์และตรวจสอบเสร็จสิ้นสมบูรณ์

2.3.2.4 การทดสอบการทำงานบนอิมูเลเตอร์

- 1) คลิกเลือกอุปกรณ์ ซึ่งจะมี Emulator ให้เลือกดังนี้
 - DefaultColorPhone : โทรศัพท์ที่แสดงผลเป็นสีโดยทั่วไป
 - DefaultGrayPhone : โทรศัพท์ที่แสดงผลเป็นสีเทาในระดับต่างๆ ด้วยสีเทา 256 เฉด
 - MediaControlSkin : โทรศัพท์ที่มีการควบคุมการเล่นของ Audio และ Video
 - Nokia_7210_MIDP_SDK_v1_0 : โทรศัพท์โนเกีย 7210
 - QwertyDevice : อุปกรณ์พกพาที่มีคีย์บอร์ดแบบเป็นพิมพ์มาตรฐาน
- 2) คลิกปุ่ม Run เพื่อทดสอบการทำงานของ MIDlet ซึ่งจะได้ผลการทำงานของ MIDlet ดังรูป



รูปที่ 2.24 การเลือก Emulator



รูปที่ 2.25 ผลลัพธ์ที่ได้จากการทำงานของ Helloworld.java

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

สถาปัตยกรรมทางซอฟต์แวร์

การพัฒนาระบบเตือนการนัดหมายและการใช้น้ำมันบนโทรศัพท์มือถือเป็นการพัฒนาซอฟต์แวร์บนโทรศัพท์มือถือซึ่งเป็นอุปกรณ์ขนาดเล็ก อีกทั้งยังมีทรัพยากร เช่น การแสดงผล ขนาดของหน่วยความจำ และ ความสามารถในการประมวลผล จำกัด ดังนั้นจึงต้องคำนึงถึงความสามารถในการเก็บข้อมูลเพื่อนำข้อมูลเหล่านั้นไปใช้งาน และคำนึงถึง ความสะดวกสบายในการใช้งานของผู้ใช้เป็นหลัก

การออกแบบซอฟต์แวร์นี้ได้ใช้ซอฟต์แวร์ Microsoft Visio 2003 มาช่วยในการเขียน Diagram แบบ UML โดยซอฟต์แวร์ Microsoft Visio 2003 นี้เป็นเครื่องมือช่วยในการสร้าง Diagram ชนิดต่างๆ เพื่อความสะดวกต่อการใช้งานของระบบงาน และได้ออกแบบ Use case diagram , Class diagram , Sequence diagram , Activity diagram ตามลำดับ

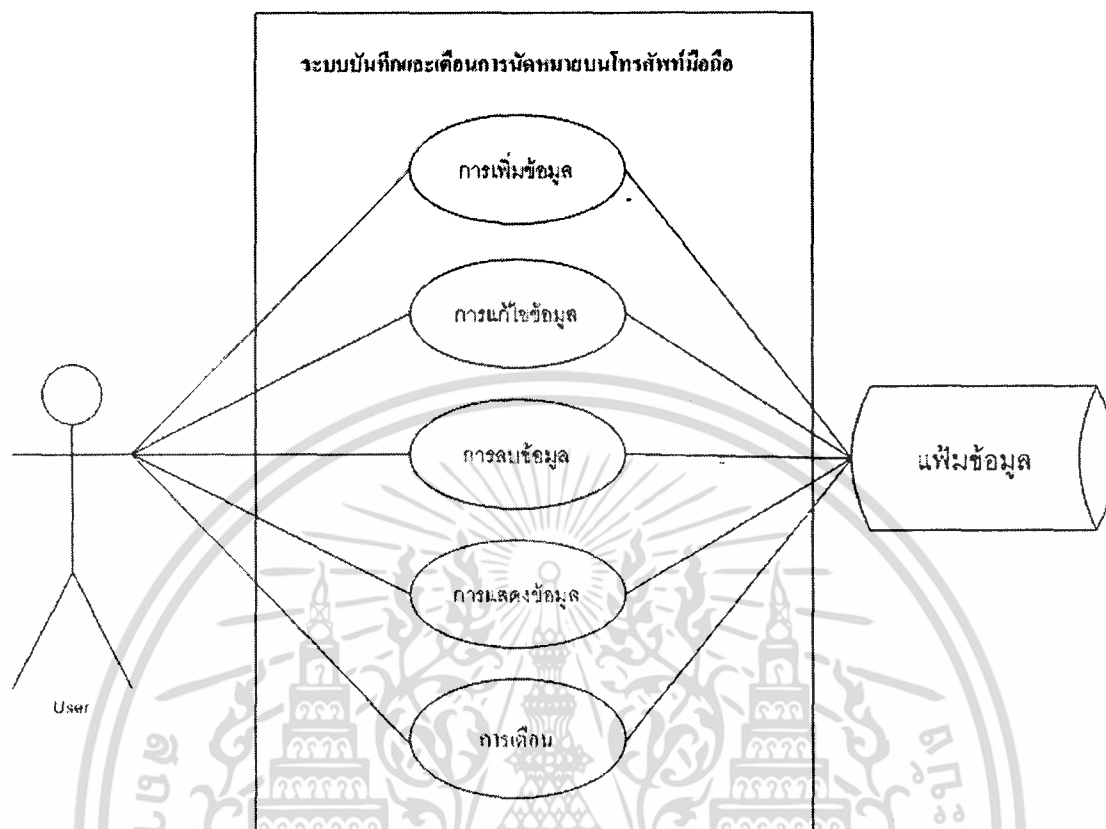
3.1 Use Case Diagram

จากการวิเคราะห์ระบบจากความต้องการทำให้ได้มาซึ่งการออกแบบ Use Case Diagram ซึ่งประกอบไปด้วย Actor และ Use Case ดังต่อไปนี้

- **Actor** คือผู้ใช้โทรศัพท์มือถือที่มีซอฟต์แวร์ระบบการบันทึกและการเตือนการนัดหมาย
- **Use Case** โดยประกอบไปด้วย 5 Use Case ดังนี้
 - **การเพิ่มข้อมูล** คือ Use Case ที่ใช้สำหรับการเพิ่มข้อมูลการบันทึกหรือการเตือนเข้าสู่ระบบ
 - **การแสดงผลข้อมูล** คือ Use Case ที่ใช้สำหรับการแสดงข้อมูลการบันทึกหรือการเตือนที่มีอยู่ในระบบ
 - **การลบข้อมูล** คือ Use Case ที่ใช้สำหรับการลบข้อมูลการบันทึกหรือการเตือนที่มีอยู่ออกจากระบบ
 - **การแก้ไขข้อมูล** คือ Use Case ที่ใช้สำหรับการแก้ไขข้อมูลการบันทึกหรือการเตือนที่มีอยู่ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Actor และ Use Case ที่มีอยู่ในระบบ ทำให้สามารถนำมาเขียนเป็น Use case Diagram ได้ดังนี้



รูปที่ 3.1 Use Case Diagram ของระบบบันทึกและเตือนการนัดหมายบนโทรศัพท์มือถือ

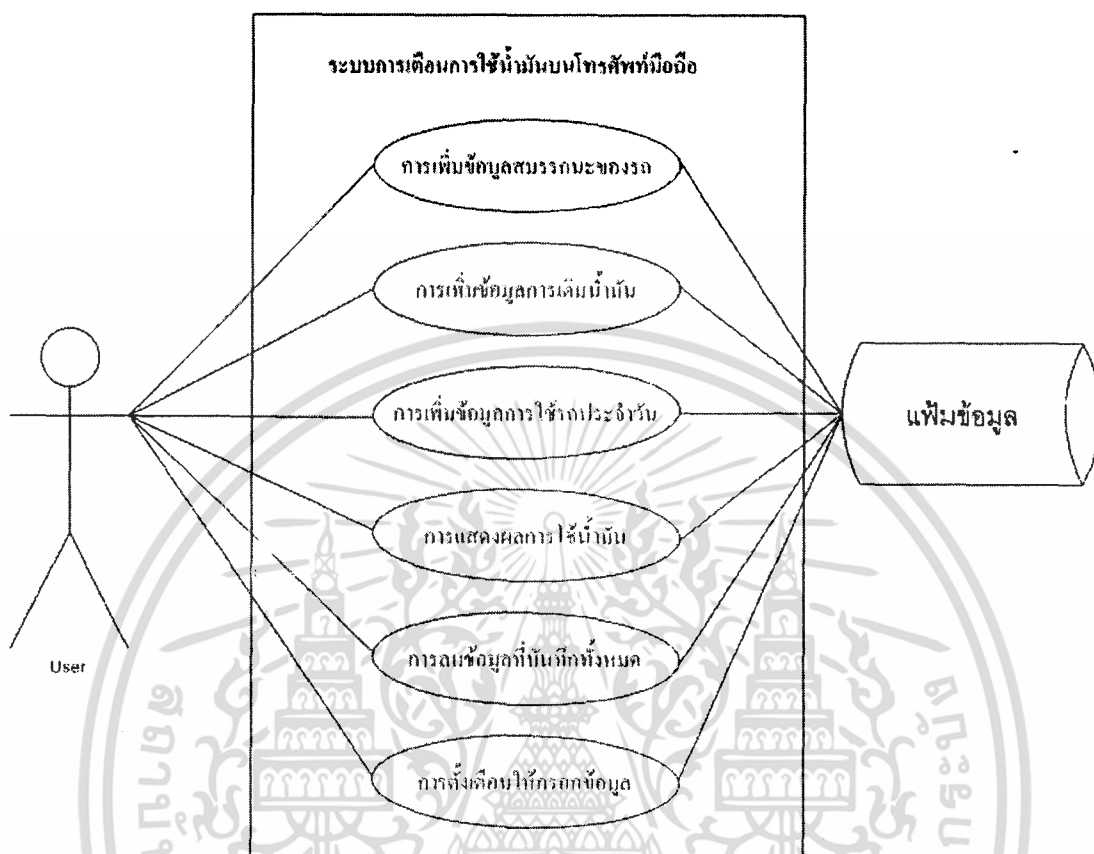
3.1.1 คำอธิบาย Use Case Diagram ของระบบบันทึกและเตือนการนัดหมาย

ระบบบันทึกและเตือนการนัดหมายบนโทรศัพท์มือถือมีความสามารถดังนี้

1. Use case : การเพิ่มข้อมูล ทำหน้าที่บันทึกการนัดหมายของผู้ใช้ลงไป ในฐานข้อมูลของโทรศัพท์มือถือ โดยมีฟังก์ชันต่างๆบันทึกข้อมูลลงไป โดยข้อมูลนั้นประกอบไปด้วย วัน/เดือน/ปีที่มีการนัดหมาย, เวลาที่ต้องการเตือนในการนัดหมาย, รายละเอียดของการนัดหมาย, หมวดของการนัดหมาย, รายละเอียดการตั้งเตือน
2. Use case : การแก้ไขข้อมูล ทำหน้าที่แก้ไขข้อมูลที่มีการบันทึกอยู่ในฐานข้อมูล โดยผู้ใช้สามารถเลือกรายการที่ต้องการแก้ไขได้ที่ละรายการ
3. Use case : การลบข้อมูล ทำหน้าที่ลบข้อมูลที่มีอยู่ในฐานข้อมูล โดยสามารถทำการลบได้ที่ละรายการและลบได้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Use case : การแสดงข้อมูล ทำหน้าที่แสดงการนัดหมายที่มีบันทึกอยู่ในฐานข้อมูลโดยผู้ใช้งานสามารถเลือกดูรายละเอียดได้ที่สกรายการ



รูปที่ 3.2 Use Case Diagram ของระบบการเตือนการใช้น้ำมันบนโทรศัพท์มือถือ

3.1.2 คำอธิบาย Use Case Diagram ของระบบการเตือนการใช้น้ำมัน

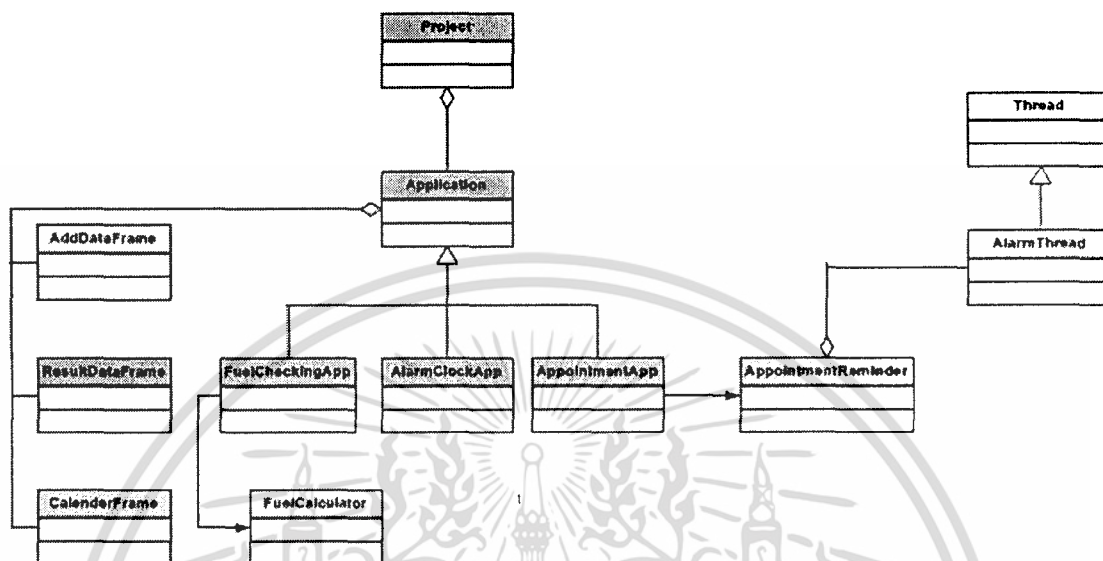
ระบบการเตือนการใช้น้ำมันบนโทรศัพท์มือถือนี้มีความสามารถดังนี้

1. Use case : การเพิ่มข้อมูลสมรรถนะของรถ ทำหน้าที่บันทึกข้อมูลสมรรถนะของรถของผู้ใช้ลงไปฐานข้อมูลของโทรศัพท์มือถือ
2. Use case : การเพิ่มข้อมูลการเติมน้ำมัน ทำหน้าที่บันทึกข้อมูล รายละเอียดของการเติมน้ำมัน
3. Use case : การแสดงผลการใช้น้ำมัน ทำหน้าที่แสดงรายละเอียดการใช้น้ำมันของผู้ใช้ออกมาทางหน้าจอ
4. Use case : การลบข้อมูลที่บันทึกทั้งหมด ทำหน้าที่ลบข้อมูลที่มีอยู่ในฐานข้อมูลทั้งหมด
5. Use case : การตั้งเตือนให้กรอกข้อมูล ทำหน้าที่บันทึกข้อมูลการเตือนในเวลาที่ใช้กำหนดและเตือนเมื่อถึงเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 Class Diagram

ระบบบันทึกข้อมูลและเตือนการนัดหมายบนโทรศัพท์มือถือ



รูปที่ 3.3 Class Diagram ของระบบบันทึกและเตือนการนัดหมายบน โทรศัพท์มือถือ

3.2.1 คำอธิบายความหมายของ Class Diagram

1. Class Project เป็นคลาสในส่วนของการทำงานของระบบ เป็นส่วนหลักของระบบที่ทำหน้าที่ควบคุมและประสานการทำงานระหว่างออปเจ็กต์ต่างๆที่มีอยู่ในระบบ
2. Class Application เป็นองค์ประกอบของโปรเจกต์โดย 1 Application จะประกอบขึ้นจาก class หน้าจอต่างได้ที่ใช้รับข้อมูล , แสดงผล หรือ แสดงปฏิทิน
3. Class AddDataFrame หมายถึง class ที่ประเภทหน้าจอที่รับข้อมูล
4. Class ResultDataFrame หมายถึง class ที่ประเภทหน้าแสดงผลข้อมูล
5. Class CalenderFrame หมายถึง class ที่ประเภทหน้าปฏิทิน
6. Class FuelCheckingApp หมายถึง application คำนวณค่าน้ำมัน
7. Class AlarmClockApp เป็นคลาสในส่วนของการเตือนหรือตั้งปลุก ผู้ใช้สามารถเพิ่มเติมแก้ไข และลบการตั้งเตือนได้โดยผ่านออปเจ็กต์ของคลาสนี้
8. Class AppointmentApp เป็นคลาสในส่วนของหน้าจอการนัดหมาย โดยผู้ใช้สามารถเพิ่มเติมแก้ไข และลบการตั้งเตือนได้โดยผ่านออปเจ็กต์ของคลาสนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Class FuelCalculator

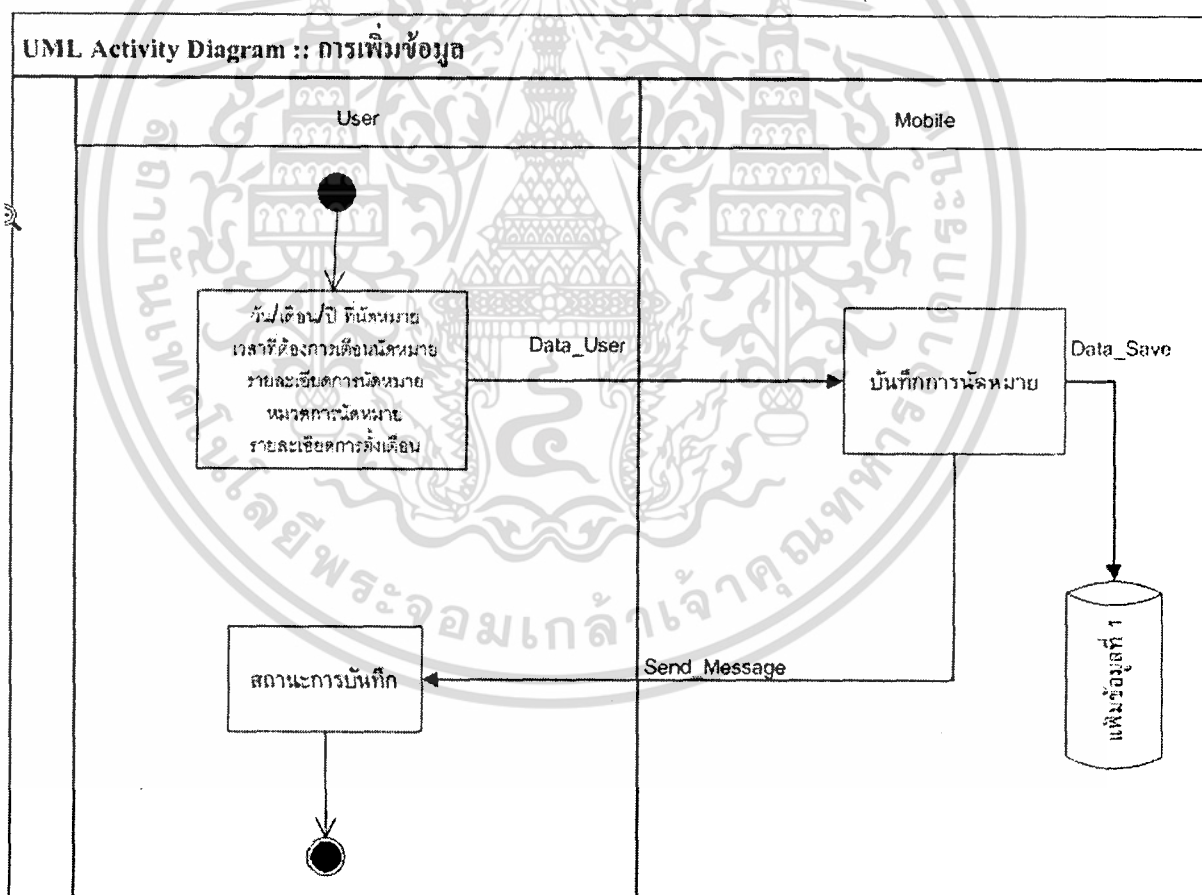
10. Class AlarmThread คลาสจำพวก Thread ที่ใช้ในการตั้งเตือน

11. Class AppointmentReminder คลาสหลักที่ทำหน้าที่ควบคุมจัดการการนัดหมายต่างๆ ประกอบไปด้วย Thread ที่ใช้ในการตั้งเตือน

3.3 Activity Diagram ของระบบบันทึกและเตือนการนัดหมายบนโทรศัพท์มือถือ

1. Use Case : การเพิ่มข้อมูล

ในส่วนของการเพิ่มข้อมูลการนัดหมายนั้น User จะทำการกรอกรายละเอียดของน้ำมันประกอบด้วยวัน/เดือน/ปีที่มีการนัดหมาย, เวลาที่ต้องการเตือนในการนัดหมาย, รายละเอียดของการนัดหมาย, หมวดของการนัดหมาย, รายละเอียดการตั้งเตือน จากนั้นระบบจะทำการบันทึกข้อมูลการนัดหมายแต่ละครั้งไว้ หากการบันทึกเสร็จเรียบร้อยแล้วจะแสดงข้อความว่า “Add Complete”

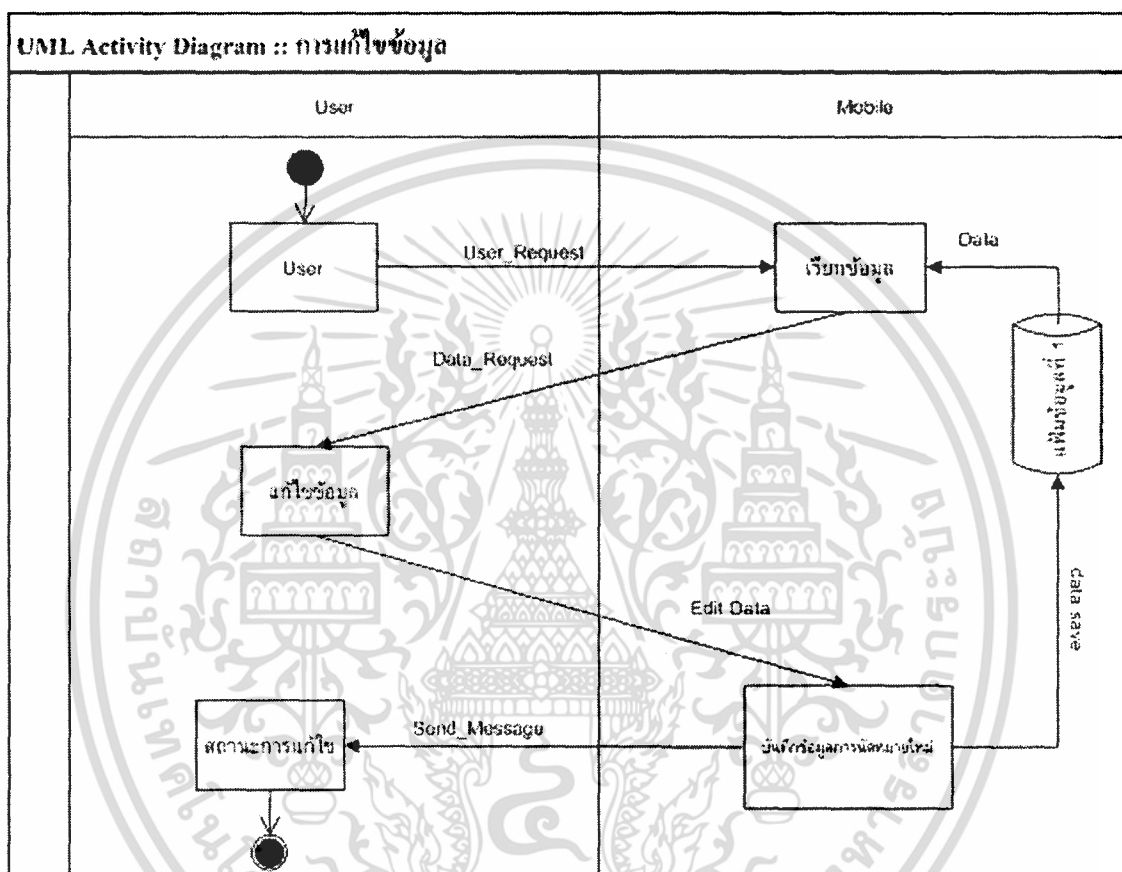


รูปที่ 3.4 Activity Diagram การเพิ่มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Use Case : การแก้ไขข้อมูล

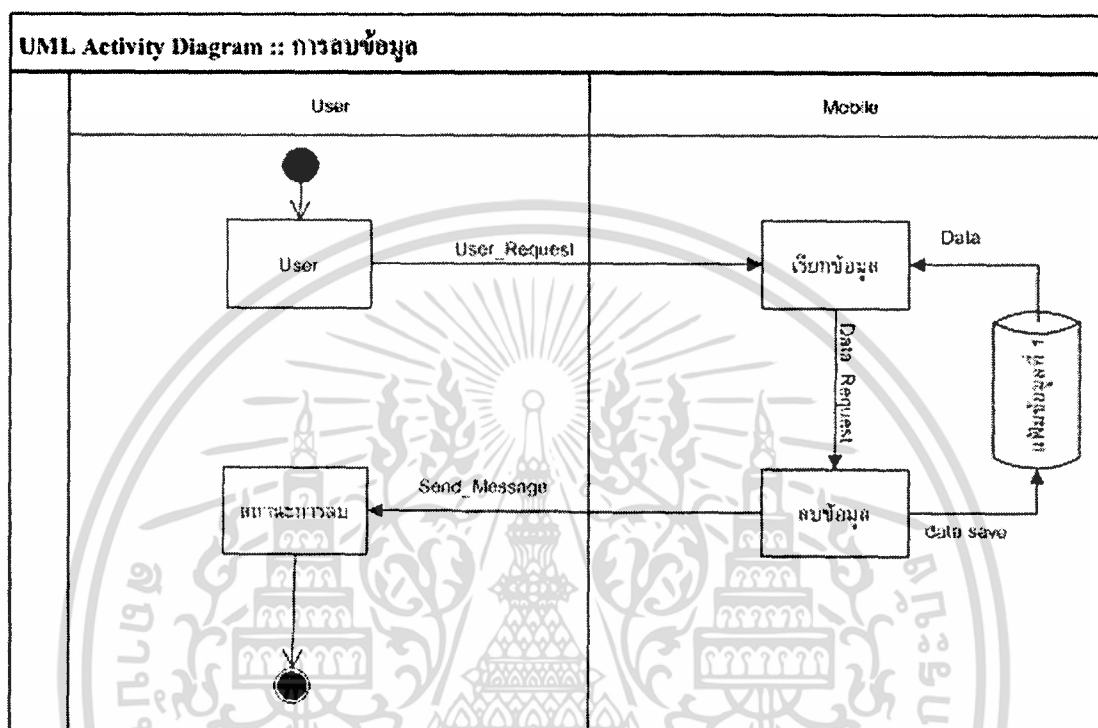
ในส่วนของการแก้ไขข้อมูล User จะต้องทำการร้องขอข้อมูลที่ต้องการแก้ไข จากฐานข้อมูล จากนั้น User จะทำการแก้ไขข้อมูล และระบบจะบันทึกข้อมูลที่ทำการแก้ไขลงในฐานข้อมูล เมื่อบันทึกเสร็จเรียบร้อยแล้วจะแสดงข้อความว่า “Delete Complete”



รูปที่ 3.5 Activity Diagram การแก้ไขข้อมูล

3. Use Case : การลบข้อมูล

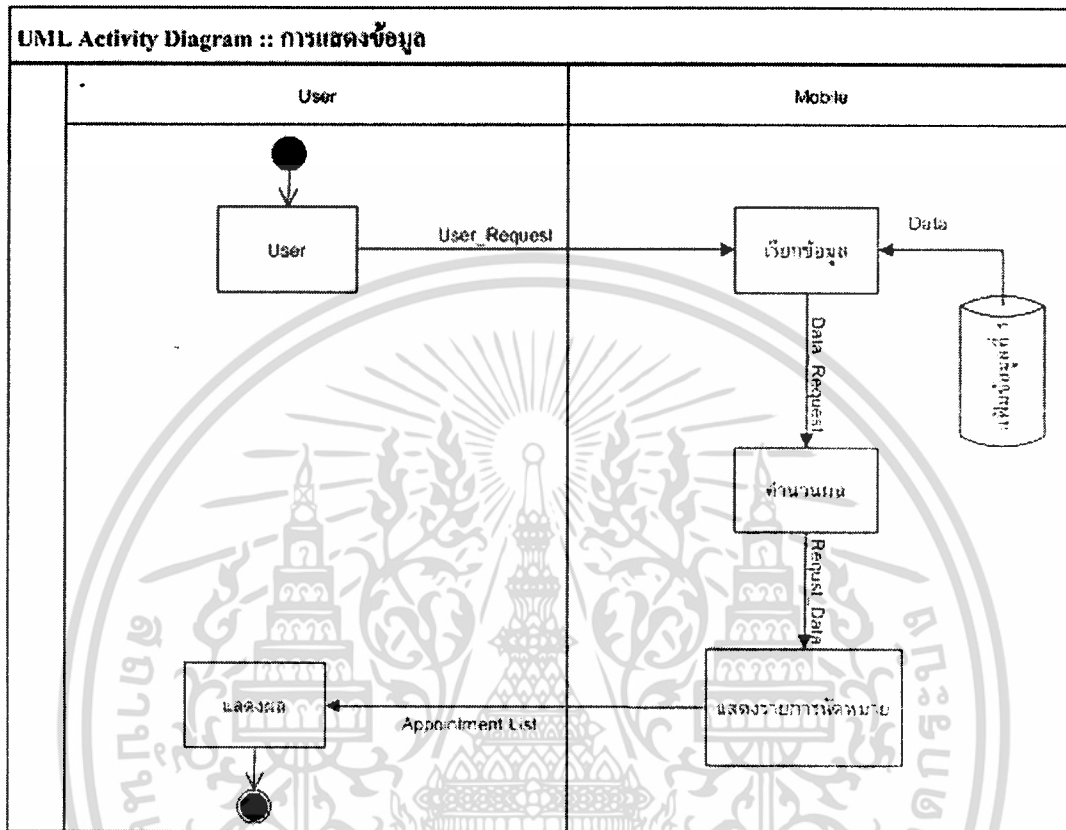
ในส่วนของการแก้ไขข้อมูล User จะต้องทำการร้องขอข้อมูลที่ต้องการลบจากฐานข้อมูล จากนั้นระบบจะทำการลบข้อมูล และบันทึกการลบข้อมูลลงในฐานข้อมูล เมื่อบันทึกเสร็จเรียบร้อยแล้วจะแสดงข้อความว่า “Delete Complete”



รูปที่ 3.6 Activity Diagram การลบข้อมูล

4. Use Case : การแสดงข้อมูล

เมื่อ User ต้องการเรียกดูข้อมูลที่เกี่ยวข้องกับข้อมูล ระบบจะแสดงผลอยู่ในรูปของข้อมูลที่มีการคำนวณผลแล้วและจัดทำอยู่ในรูปแบบรายการนัดหมาย

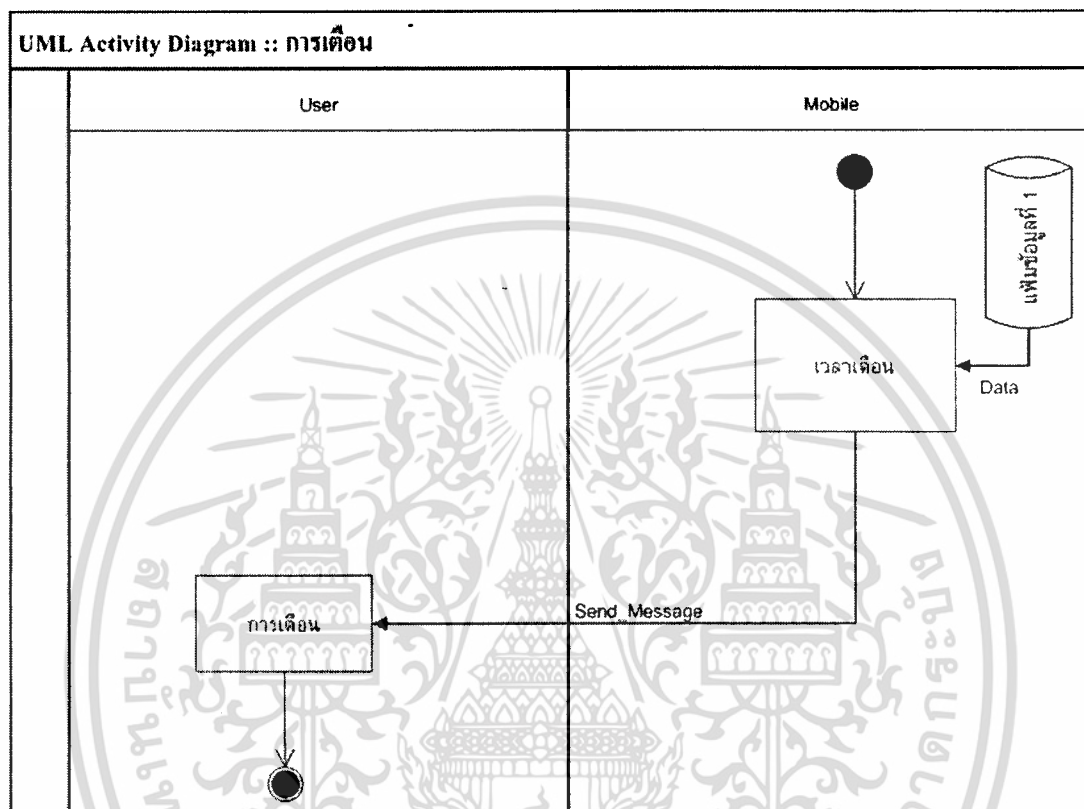


รูปที่ 3.7 Activity Diagram การแสดงข้อมูล

5. Use Case : การเตือน

เมื่อถึงเวลาที่ผู้ใช้ได้กำหนดไว้ในการเตือน ระบบจะทำการเตือนให้ผู้ใช้ได้

ทราบ



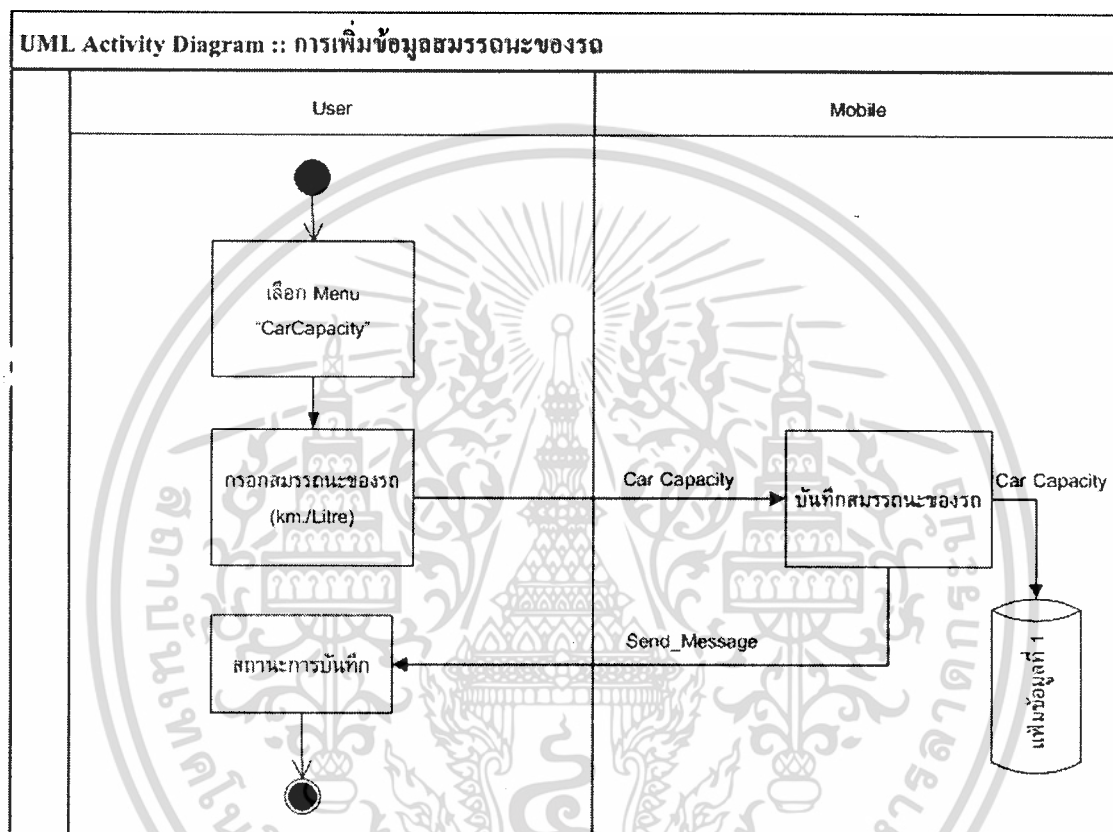
รูปที่ 3.8 Activity Diagram การเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Activity Diagram ของระบบการเตือนการใช้น้ำมันบนโทรศัพท์มือถือ

1. Use Case : การเพิ่มข้อมูลสมรรถนะของรถ

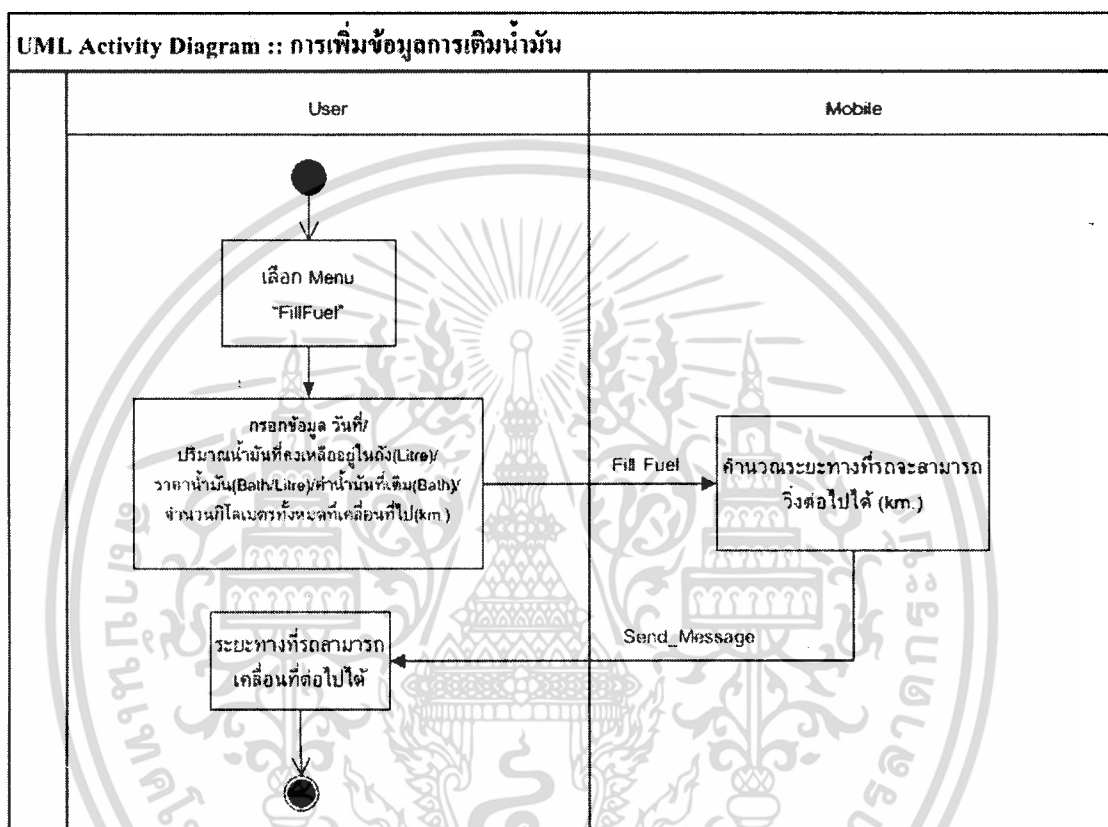
ในส่วนของการเพิ่มข้อมูลสมรรถนะของรถนั้น User จะทำการกรอกสมรรถนะของรถ จากนั้นระบบจะทำการบันทึกข้อมูลสมรรถนะของรถไว้ หากการบันทึกเสร็จเรียบร้อยแล้ว จะแสดงข้อความว่า “Add Complete”



รูปที่ 3.9 Activity Diagram การเพิ่มข้อมูลสมรรถนะของรถ

2. Use Case : การเพิ่มข้อมูลการเติมน้ำมัน

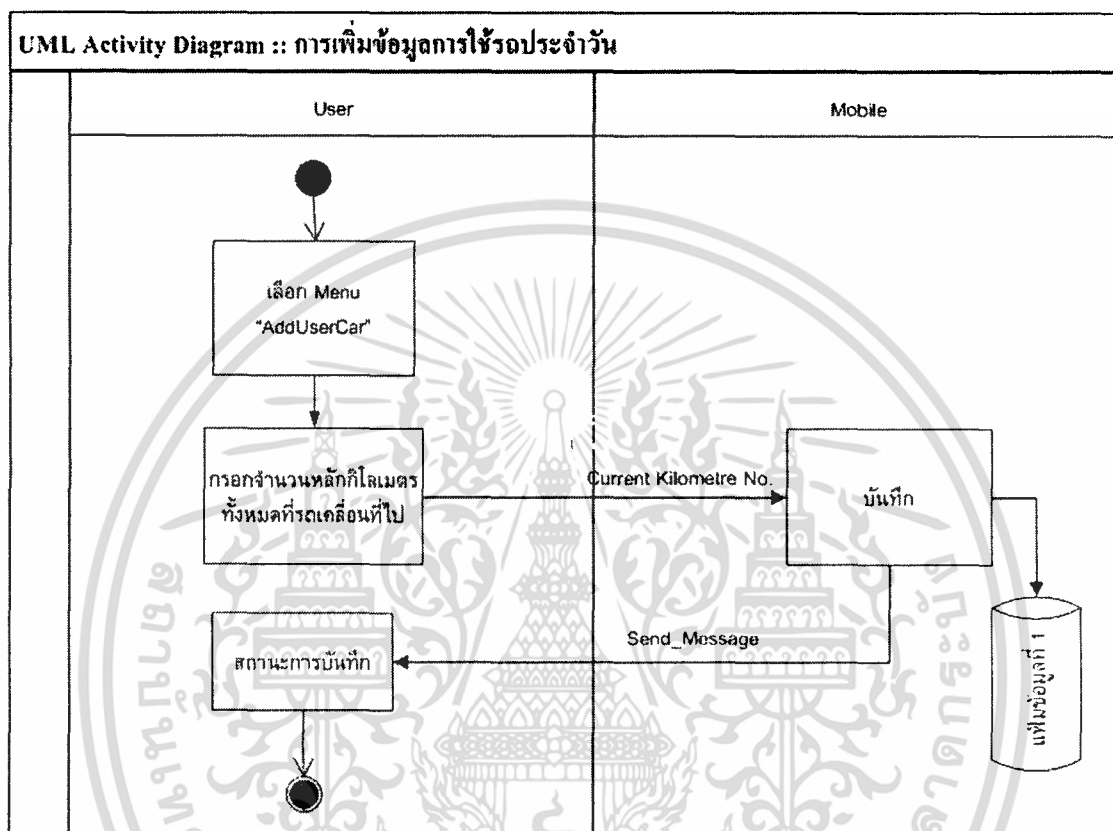
ในส่วนของการเพิ่มข้อมูลการเติมน้ำมันนั้น User จะทำการกรอกข้อมูลการเติมน้ำมันประกอบไปด้วย วันที่เติมน้ำมัน, ปริมาณน้ำมันที่คงเหลืออยู่ในถัง, ราคาน้ำมัน, ค่าน้ำมันที่เติม, จำนวน กิโลเมตรทั้งหมดที่เคลื่อนที่ไป จากนั้นระบบจะทำการคำนวณระยะทางที่รถจะสามารถวิ่งต่อไปได้ จากนั้นระบบจะแสดงข้อมูลที่คำนวณได้ขึ้นมาทางหน้าจอ



รูปที่ 3.10 Activity Diagram การเพิ่มข้อมูลการเติมน้ำมัน

3. Use Case : การเพิ่มข้อมูลการใช้รถประจำวัน

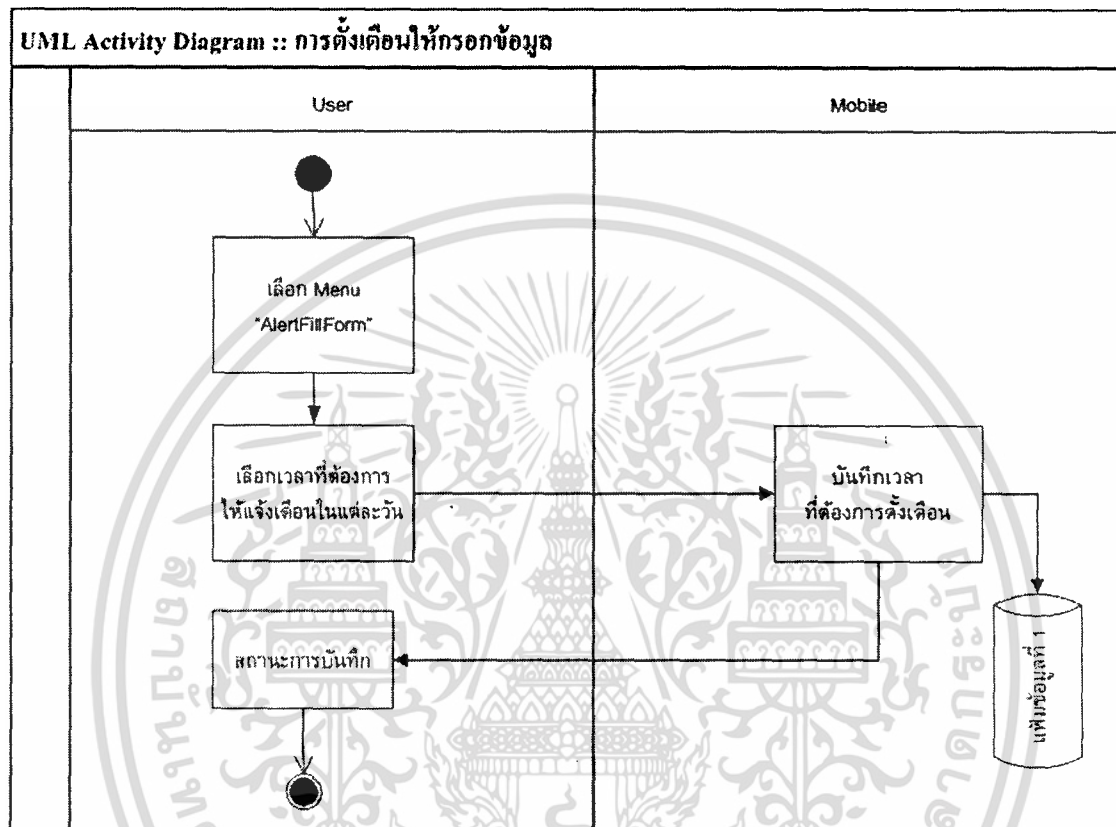
ในส่วนของการเพิ่มข้อมูลสมรรถนะของรถนั้น User จะทำการกรอกจำนวนหลักกิโลเมตรทั้งหมดที่รถเคลื่อนที่ไป จากนั้นระบบจะทำการบันทึกข้อมูลไว้ หากการบันทึกเสร็จเรียบร้อยแล้วจะแสดงข้อความว่า “Add Complete”



รูปที่ 3.11 Activity Diagram การเพิ่มข้อมูลการใช้รถประจำวัน

4. Use Case : การตั้งเตือนให้กรอกข้อมูล

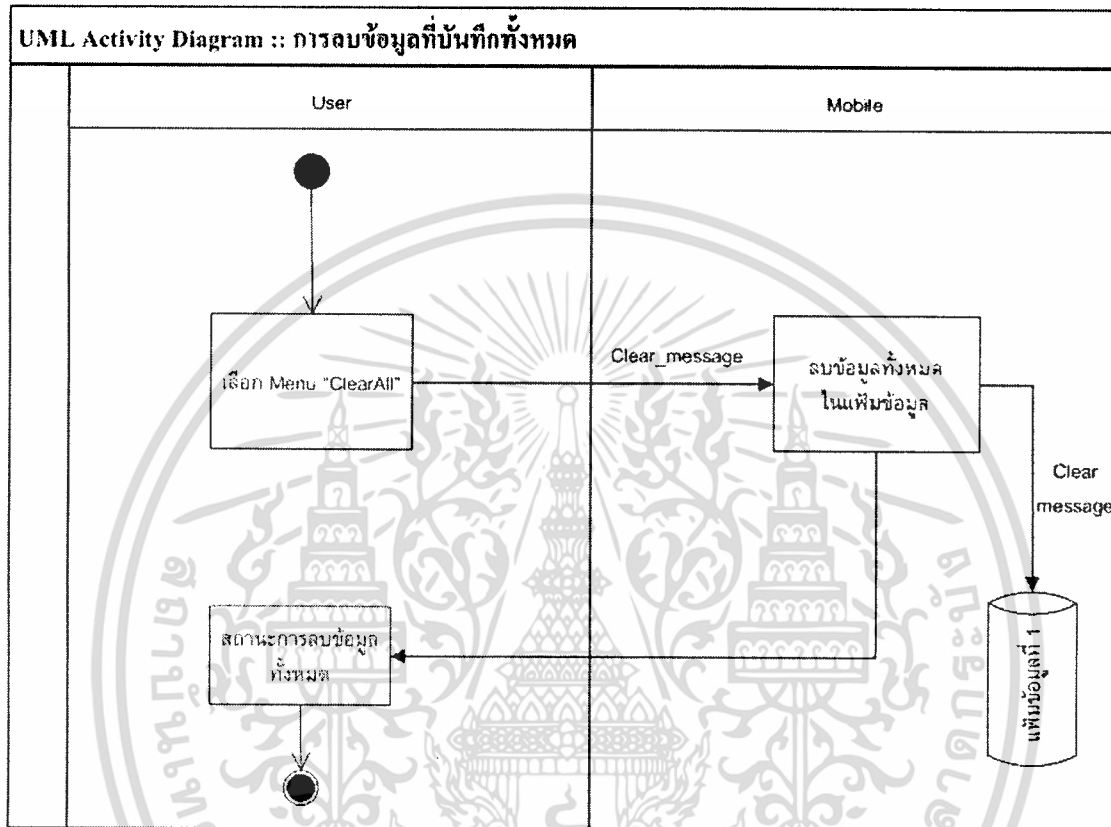
ในส่วนของการตั้งเตือนให้กรอกข้อมูลนั้น User จะทำการกรอกเวลาที่ต้องการให้แจ้งเตือนในแต่ละวัน จากนั้นระบบจะทำการบันทึกข้อมูลไว้ หากการบันทึกเสร็จเรียบร้อยแล้วจะแสดงข้อความว่า “Add Complete”



รูปที่ 3.12 Activity Diagram การตั้งเตือนให้กรอกข้อมูล

5. Use Case : การลบข้อมูลที่บันทึกทั้งหมด

ในส่วนของการลบข้อมูล เมื่อ User ทำการลบข้อมูลที่บันทึกทั้งหมด ระบบจะทำการลบข้อมูลทั้งหมดที่มีอยู่ในฐานข้อมูลและเมื่อทำการลบเรียบร้อยแล้วจะแสดงข้อความว่า “Delete Complete”



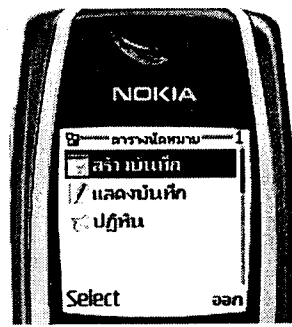
รูปที่ 3.13 Activity Diagram การลบข้อมูลที่บันทึกทั้งหมด

3.5 การออกแบบ User Interface ของระบบ

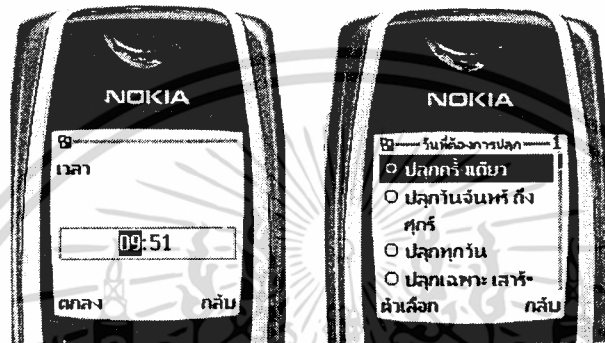


รูปที่ 3.14 แสดง User Interface ของหน้าจอหลัก (MAIN MENU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดง User Interface แสดงหน้าจอการนัดหมาย



รูปที่ 3.16 แสดง User Interface แสดงหน้าจอการตั้งปลูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและการวิเคราะห์ข้อมูล

4.1 ขั้นตอนการเข้าสู่โปรแกรม

การใช้งาน โปรแกรมการเตือนการใช้น้ำมันและการนัดหมายบนโทรศัพท์มือถือจะงานผ่านทาง MOBILEPHONE EMULATOR โดยจะมีไอคอนหน้าจอหลักปรากฏบนหน้าจอ ดังรูป

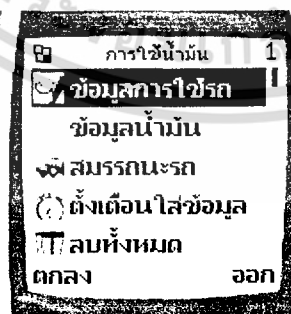
หน้าจอหลักจะมีรายการให้เลือกว่าต้องการใช้โปรแกรมการเตือนประเภทใด ให้กดปุ่มเลื่อนเพื่อเลือกโปรแกรมการเตือนประเภทที่ต้องการแล้วกดปุ่มเลือก เพื่อเลือกโปรแกรมนั้น



รูปที่ 4.1 หน้าจอหลัก

4.1.1 โปรแกรมการเตือนการใช้น้ำมัน

- 1) เมื่อเลื่อนแถบเลือกโปรแกรมการตรวจสอบปริมาณน้ำมันจะปรากฏเมนู ดังรูป



รูปที่ 4.2 หน้าจอหลักของโปรแกรมตรวจสอบปริมาณน้ำมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของ หน้าจอหลักในโปรแกรมการตรวจสอบปริมาณน้ำมัน

ข้อมูลการใช้รถ : เมนูที่ให้กรอกรายละเอียดการใช้รถ

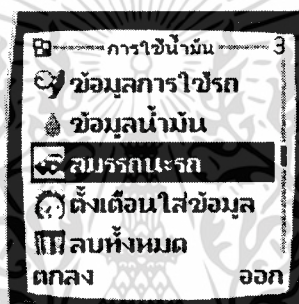
ข้อมูลน้ำมัน: เมนูที่ให้กรอกรายละเอียดการเติมน้ำมัน

สมรรถนะรถ : เมนูที่ให้กรอกรายละเอียดของสมรรถนะรถ

ตั้งเตือนใส่ข้อมูล : เมนูที่ให้กรอกรายละเอียดการตั้งเตือนให้กรอกข้อมูล

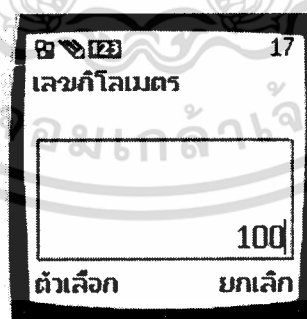
ลบทั้งหมด : เมนูที่ทำการล้างข้อมูลการตรวจสอบปริมาณน้ำมันทั้งหมดที่มีในเครื่อง

- 2) ผู้ใช้จะต้องทำการกรอกข้อมูลเกี่ยวกับรถก่อน โดยจะต้องเข้าไปที่เมนูสมรรถนะรถ



รูปที่ 4.3 เลือกเมนูสมรรถนะรถ

ใส่ข้อมูลสมรรถนะรถ หน่วยเป็นกิโลเมตรต่อลิตร เมื่อเข้าเมื่อดังกล่าวจะปรากฏหน้าจอ ดังรูป ให้ใส่รายละเอียดของสมรรถนะรถ



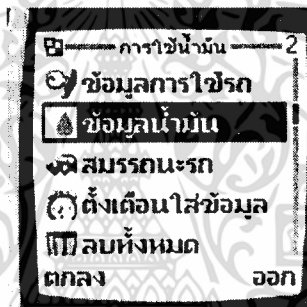
รูปที่ 4.4 หน้าจอแสดงให้กรอกรายละเอียดสมรรถนะรถ

เมื่อกรอกข้อมูลรายละเอียดสมรรถนะรถเรียบร้อยแล้วจะมีหน้าจอแสดงว่าได้ทำการใส่ข้อมูลเรียบร้อยแล้วดังรูป



รูปที่ 4.5 หน้าจอแสดงเมื่อทำการกรอกรายละเอียดข้อมูลเรียบร้อยแล้ว

- 4) เมื่อผู้ใช้ได้กรอกรายละเอียดสมรรถนะรถไปแล้วจะต้องทำการกรอกข้อมูลของการเติมน้ำมัน โปรแกรมจึงจะทำงานได้ โดยเข้าไปที่เมนูข้อมูลน้ำมัน



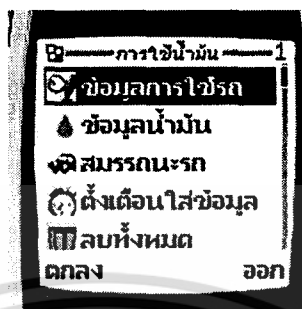
รูปที่ 4.6 เลือกรายการข้อมูลน้ำมัน

จะปรากฏหน้าจอหน้ากรอกข้อมูลดังรูป

ข้อมูลน้ำมัน	
น้ำมันคงเหลือ(ลิตร):	[0]
ราคาน้ำมัน(บ./ล.):	[25.50]
จำนวนที่เติม(บาท):	[2550]
เลขกิโลก่อนเติม:	[0]
ตัวเลือก	กลับ

รูปที่ 4.7 หน้าจอแสดงให้กรอกรายละเอียดข้อมูลน้ำมัน

- 5) หลังจากผู้ใช้ได้กรอกข้อมูลตามสองข้อข้างบนแล้ว โปรแกรมจะเริ่มทำงานเมื่อผู้ใช้ทำการกรอกข้อมูลการใช้รถประจำวัน หรือทุกครั้งก่อนใช้รถ ให้ผู้ใช้เปิดโปรแกรมเข้ามากรอก โดยเลือกที่เมนูข้อมูลการใช้รถ



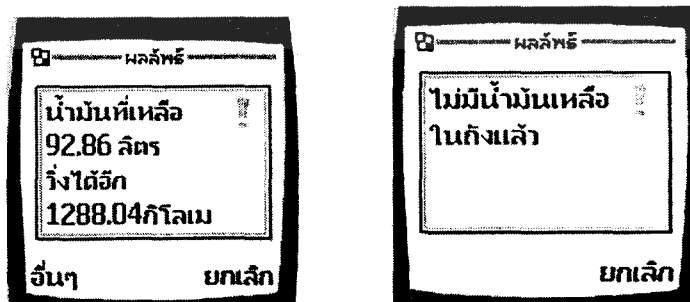
รูปที่ 4.8 เลือกเมนูข้อมูลการใช้รถ

เมื่อเข้ามาจะปรากฏหน้าจอให้ใส่ข้อมูลดังรูป



รูปที่ 4.9 แสดงหน้าจอให้กรอกรายละเอียดการใช้รถประจำวัน

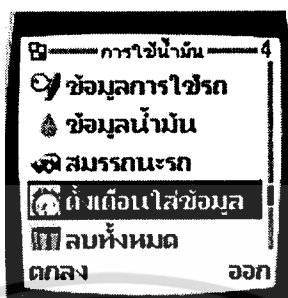
เมื่อกรอกรายละเอียดการใช้รถประจำวันโดยกรอกเลขกิโลเมตรรถลงไป โปรแกรมจะทำการประมวลผลคำนวณ รายละเอียดการใช้น้ำมันให้โดยจะมีผลลัพธ์เดือนออกทางหน้าจอ โดยจะแจ้งเตือนว่าผู้ใช้จะสามารถขับไปได้เป็นระยะทางกี่กิโลเมตร และเหลือน้ำมันในถังอีกเท่าไร



รูปที่ 4.10 แสดงผลลัพธ์หลังจากได้รายละเอียดการใช้รถประจำวันเรียบร้อยแล้ว

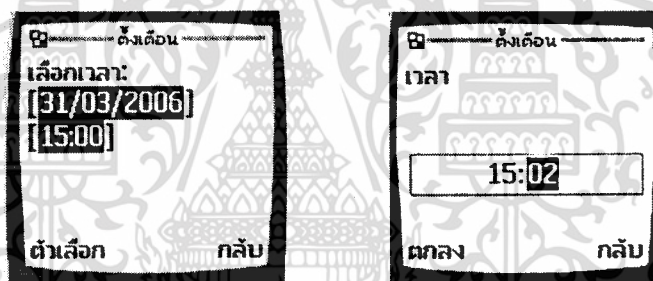
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) เพื่อป้องกันการลืมการกรอกข้อมูลการใช้รถประจำวัน โปรแกรมจะมีเมนูตั้งเตือนใส่ข้อมูล ให้ผู้ใช้ได้ทำการตั้งเวลา



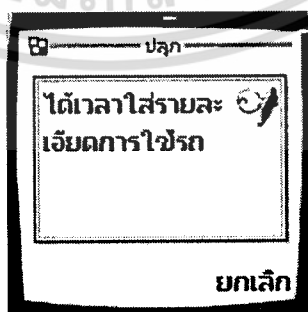
รูปที่ 4.11 เลือกเมนูตั้งเตือนใส่ข้อมูล

เพื่อให้โปรแกรมเตือนให้ผู้ใช้ทำการกรอกข้อมูลการใช้รถประจำวันทุกวัน โดยเมื่อทำการตั้งเวลาแล้ว โปรแกรมจะปลุกให้ผู้ใช้ทำการกรอกข้อมูลตามเวลาที่ผู้ใช้เลือก



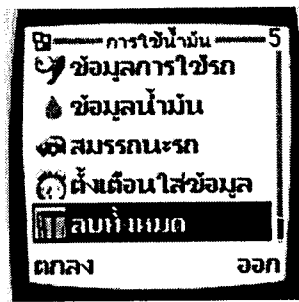
รูปที่ 4.12 แสดงหน้าจอการตั้งเตือนให้กรอกรายละเอียดการใช้รถประจำวัน

เมื่อถึงเวลาที่ผู้ใช้ได้ทำการตั้งให้ทำการใส่ข้อมูลจะปรากฏหน้าจอเตือนให้ผู้ใช้ใส่ข้อมูลดังรูป



รูปที่ 4.13 แสดงหน้าจอเตือนให้ผู้ใช้ทำการกรอกข้อมูลการใช้รถประจำวัน

7) หากต้องการล้างข้อมูลทั้งหมดให้เลือกเมนูลบทั้งหมด เพื่อล้างข้อมูลที่มีในเครื่อง



รูปที่ 4.14 เลือกเมนูลบทั้งหมด

เมื่อเลือกเมนูนี้จะปรากฏผลลัพธ์ทางหน้าจอแสดงให้เห็นว่าได้ทำการลบข้อมูลทั้งหมดแล้ว



รูปที่ 4.15 แสดงหน้าจอเมื่อเลือกเมนูลบทั้งหมด

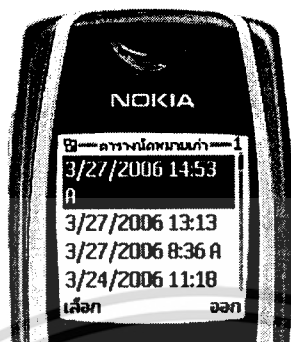
4.1.2 โปรแกรมการเตือนการนัดหมาย



รูปที่ 4.16 แสดงหน้าจอหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือก “นัดหมาย” จากเมนูหลักแล้ว จะพบหน้าจอแสดง “ตารางนัดหมายเก่า” ซึ่งเป็นนัดหมายที่ผู้ใช้ได้ทำการบันทึกไว้แล้วทั้งหมด โดยผู้ใช้สามารถ แก้ไข , ลบ นัดหมายได้โดยเลือกปุ่ม “เลือก” แต่หากผู้ใช้ต้องการสร้างนัดหมายให้เลือกปุ่ม “ออก ”



รูปที่ 4.17 หน้าจอแสดง ตารางนัดหมายเก่า

การสร้างการนัดหมาย

เมื่อเลือก นัดหมาย → ตารางนัดหมายเก่า → ออก จะพบหน้าจอ “ตารางนัดหมาย ” และผู้ใช้สามารถเลือกทำการการ สร้างบันทึก ,แสดงบันทึก,ปฏิทิน ให้ผู้ใช้เลือก สร้างบันทึก จะพบหน้าจอกรอกข้อมูลการนัดหมาย ดังรูป



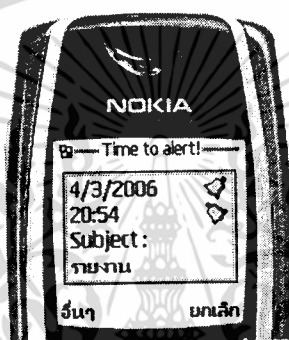
รูปที่ 4.18 หน้าจอสร้างการนัดหมาย

โดยกรอก วัน-เดือน-ปี และ เวลา ที่มีการนัดหมาย จากนั้นเลือก ชนิดของการเตือน โดยสามารถเลือกได้มากกว่า 1 ตัวเลือก



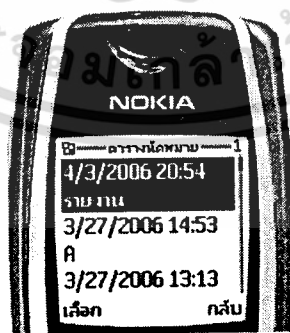
รูปที่ 4.19 หน้าจอเลือกชนิดของการเตือน

จากนั้นกรอก สถานที่นัดหมาย , ชื่อเรื่อง แล้วเลือกปุ่ม “ตัวเลือก” อีกครั้งแล้วเลือก “บันทึก” จะแสดงหน้าจอสถานะการบันทึกพร้อมรายละเอียด ดังรูป



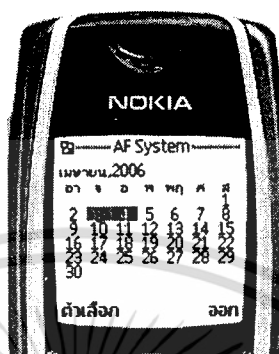
รูปที่ 4.20 หน้าจอสถานะการบันทึกพร้อมรายละเอียด

ผู้ใช้สามารถเข้าไปดูนัดหมายที่บันทึกไว้ได้ จาก เมนู “แสดงบันทึก” หรือ “ปฏิทิน” ก็ได้ หากเลือก เมนู “แสดงบันทึก” จะพบหน้าจอแสดงนัดหมายทั้งหมดเป็นรายการเรียงตามลำดับ ผู้ใช้สามารถเข้าไปรายละเอียดการนัดหมาย และสามารถแก้ไขหรือ ลบ ได้ ดังรูป



รูปที่ 4.21 หน้าจอแสดงนัดหมายทั้งหมดเรียงตามลำดับ

หากเลือก เมนู “ปฏิทิน” จะพบหน้าจอปฏิทินรายเดือน ซึ่งแสดงวันที่ที่มีการนัดหมายอยู่ภายในกรอบสีส้ม ส่วนภายในกรอบสีฟ้า แสดงวันที่ปัจจุบัน ผู้ใช้สามารถเข้าไปในวันที่ที่มีการนัดหมายเพื่อดูการนัดหมายทั้งหมดของวันนั้นได้ และสามารถแก้ไข หรือลบ ได้ ดังรูป



รูปที่ 4.22 หน้าจอปฏิทินรายเดือน

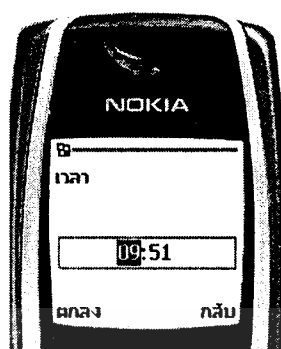
4.1.3 โปรแกรมนาฬิกาปลุก

เมื่อเลือก “นาฬิกาปลุก” จากเมนูหลักแล้ว



รูปที่ 4.23 หน้าจอรายการหลัก

จากนั้นให้ผู้ผู้ใช้กรอกเวลาปลูก แล้วกด ตกลง

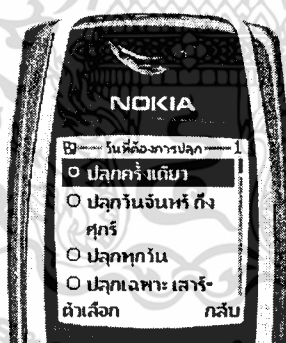


รูปที่ 4.24 หน้าจกรอกเวลาปลูก

จากนั้นเลือก ประเภทการปลูก

ผู้ใ้สามารถใช้เลือกวันที่ต้องการปลูกได้ เลือกจาก

- ปลูกเพียงครั้งเดียวและวันนั้นวันเดียว
- ปลูกทุกวันจันทร์ ถึง ศุกร์
- ปลูกทุกวัน
- ปลูกเฉพาะวัน เสาร์-อาทิตย์



รูปที่ 4.25 ประเภทของการปลูก

เมื่อกรอกข้อมูลครบ กด “บันทึก” จะพบหน้าจอ แสดงสถานะการบันทึก



รูปที่ 4.26 หน้าจอแสดงสถานะการบันทึก

เมื่อถึงเวลาปลุกจะแสดงหน้าจอ ดังรูป



รูปที่ 4.27 การปลุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปัจจุบันโทรศัพท์มือถือเข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก นอกจากความสามารถทั่วไปของมือถือเช่น รับสาย, โทรออก, ส่งข้อความได้แล้ว เรายังสามารถเพิ่มความสามารถให้กับมือถือโดยการใช้โปรแกรมการเตือนการใช้น้ำมันและการนัดหมายบนโทรศัพท์มือถือ ซึ่งทำมือถือเป็นเหมือนผู้จัดการส่วนตัว คอยจัดการเกี่ยวกับ นัดหมาย , เตือนความจำ แบบเลือกเวลาเตือนและเลือกความถี่ในการเตือน ได้อีกด้วย และจะแสดงวันที่มีการนัดหมายบนปฏิทินเพื่อให้ผู้ใช้เห็นเด่นชัดทำให้ง่ายต่อการเข้าใจ นอกจากความสามารถพิเศษเหล่านี้แล้ว โปรแกรมการเตือนการใช้น้ำมันและการนัดหมายบนโทรศัพท์มือถือ นี้จะทำให้มือถือสามารถเตือนการใช้น้ำมันให้กับผู้ใช้รถ เพื่อช่วยเตือนการเติมน้ำมันครั้งต่อไป ป้องกันการลืมและลดการเกิดปัญหาน้ำมันหมดก่อนถึงปลายทางได้อีกด้วย โดยส่วนนี้เป็นส่วนพิเศษที่ได้พัฒนาขึ้นใหม่ ข้อมูลในโปรแกรมนี้สามารถ แก้ไข , ลบ ได้ตามความต้องการของผู้ใช้

5.2 ข้อเสนอแนะ

ระบบนี้อาจมีการพัฒนาแอปพลิเคชันในอนาคต ดังนี้

1. ในส่วนการกรอกข้อมูลสมรรถนะของรถยนต์ให้ดึงข้อมูลจากอินเทอร์เน็ตแทนการกรอกข้อมูลจากผู้ใช้ โดยให้ผู้ใช้เลือกเพียงยี่ห้อและรุ่นของรถเท่านั้นเพื่อความสะดวกในการใช้งาน
2. ในส่วนการพิมพ์ข้อความในการนัดหมายอาจจะมีเครื่องมือคอยช่วยเหลือในการพิมพ์โดยมีข้อความซึ่งเป็นชนิดของการนัดหมายที่มักใช้โดยทั่วไปไว้ให้เลือกใช้ได้เลย ประหยัดเวลา และสะดวกกับผู้ใช้เป็นอย่างมาก
3. ในการเตือนของการนัดหมายและการใช้น้ำมัน ผู้ใช้สามารถเลือกเสียงปลุกได้ตามต้องการ เพื่อแสดงความต่างกัน เมื่อผู้ใช้ได้ยินเสียงจะรู้ทันทีว่าเป็นการเตือนการนัดหมายหรือเตือนการใช้น้ำมัน
4. เพิ่มกราฟฟิค บนหน้าจอแต่ละหน้าตามความเหมาะสมเพื่อความสวยงาม ชวนมอง และดึงดูดความสนใจของผู้ใช้มากขึ้น แต่ไม่ควรมากจนเกินไปควรคำนึงถึงขนาดของจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. อาจมีการพัฒนาโปรแกรมใหม่ขึ้นมาเกี่ยวกับการแจ้งเตือนเมื่อน้ำมันใกล้หมดถัง และบอก
ระยะทางที่สามารถวิ่งได้ก่อนน้ำมันจะหมด เพื่อเตือนผู้ขับรถให้รีบเติมน้ำมัน

6. อาจพัฒนาเป็นโปรแกรมหาปั้มน้ำมันที่ใกล้ที่สุดโดยใช้ GPRS ควบคู่ไปกับโปรแกรมแจ้ง
เตือนเมื่อน้ำมันใกล้หมดถัง เพื่อช่วยในการตัดสินใจ และ เพิ่มความสะดวก ได้อีกด้วย



บรรณานุกรม

กาญจนา ตันวิสุทธิ. 2547. เขียนเกมและโปรแกรมบนมือถือ J2ME. นนทบุรี : ไอดีซี อินโฟ
ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

ธวัชชัย ชูลีพงศ์เชษฐ์ และนพดล นวลพรหม. 2549. โปรแกรมตารางนัดหมายบน
โทรศัพท์มือถือ. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์

คอมพิวเตอร์,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปิยะนันท์ สมบัติวิชาธร, วราวุธ วัฒนศิริ และสุรศักดิ์ เหลืองอากุล 2549. การพัฒนา
ซอฟต์แวร์เพื่อการจัดการหุ้นส่วนบุคคลบนเครื่อง PDA . วิทยานิพนธ์วิทยาศาสตร
มหาบัณฑิต สาขาวิชาวิทยาศาสตร์ คอมพิวเตอร์,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง



ภาคผนวก ก

การติดตั้งเครื่องมือที่ใช้ในการพัฒนา

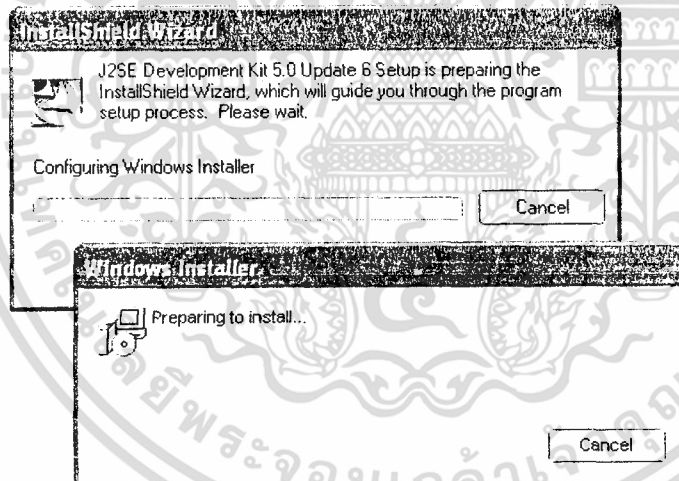
เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์บนโทรศัพท์มือถือ ประกอบไปด้วย

- 1) Java 2 Standard Edition (J2SE SDK)
- 2) J2ME Wireless Toolkit (J2MEWTK)
- 3) MOBILEPHONE Emulator

วิธีการดาวน์โหลดและติดตั้ง Java 2 Standard Edition (J2SE SDK)

เครื่องมือในการพัฒนา Java 2 Standard Edition (J2SE SDK) เวอร์ชัน 1.5 โดยสามารถเข้าไปดาวน์โหลดได้ที่ <http://java.sun.com/j2se/1.5.0/download.jsp> เมื่อดาวน์โหลดเรียบร้อยแล้วสามารถทำการติดตั้งได้ตามขั้นตอนดังนี้

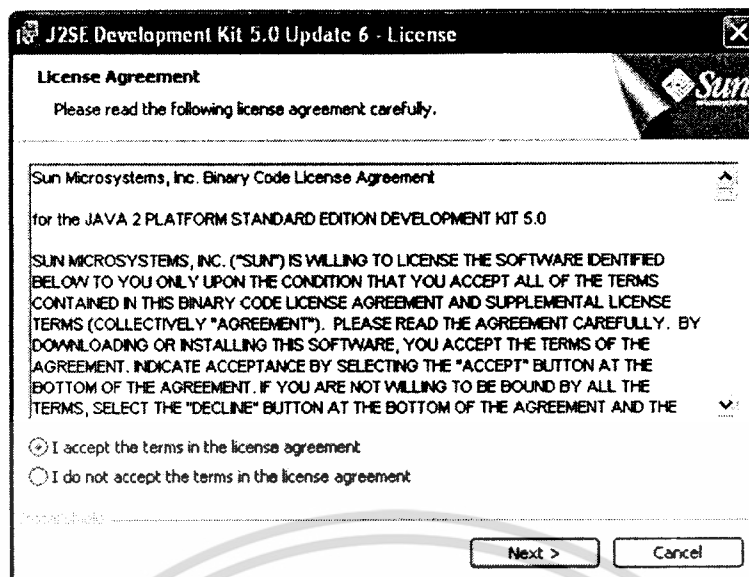
- 1) ดับเบิลคลิกไฟล์ `jdk-1_5_0_06-windows-i586-p.exe` ที่ได้อัปโหลดมาเพื่อทำการติดตั้งโปรแกรมลงในเครื่องคอมพิวเตอร์



รูปที่ ก.1 การเตรียมพร้อมก่อนทำการติดตั้งซอฟต์แวร์

- 2) คลิกเลือก I accept the terms in the license agreement
- 3) คลิกปุ่ม Next > เพื่อทำการติดตั้ง

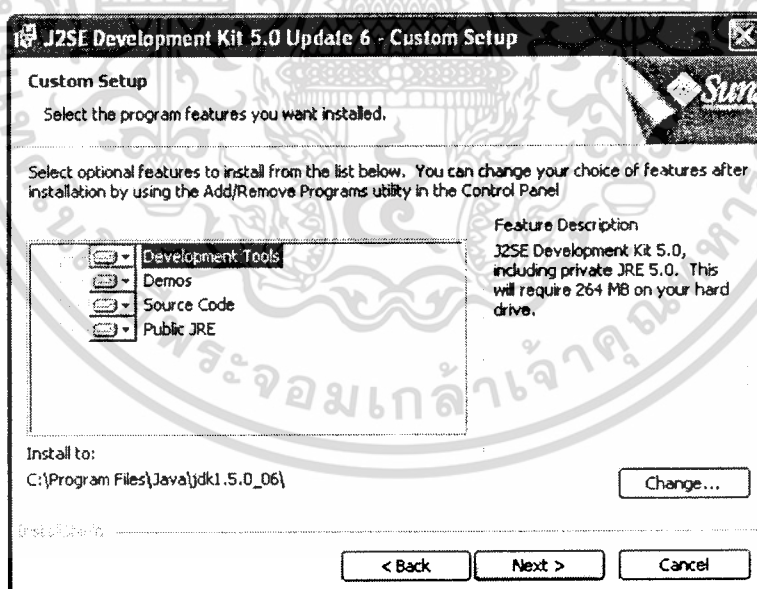
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 ขอมรับเงื่อนไขในการติดตั้งซอฟต์แวร์ J2SE SDK

- 4) เลือก Development Tools ซึ่งโดยปกติจะเป็นค่าที่ถูกกำหนดไว้อยู่แล้ว
- 5) กำหนดโฟลเดอร์ที่ต้องการติดตั้ง โดยค่าปกติคือ C:\Program Files\Java\jdk1.5.0_06
- 6) คลิกปุ่ม Next > เพื่อทำการติดตั้ง

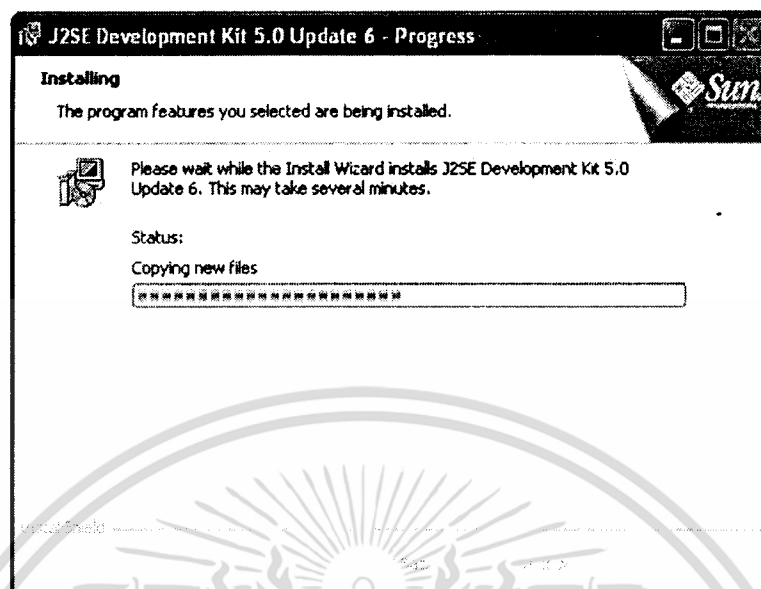
C:\Program



รูปที่ ก.3 การเลือกฟีเจอร์และกำหนดไดเรกทอรี

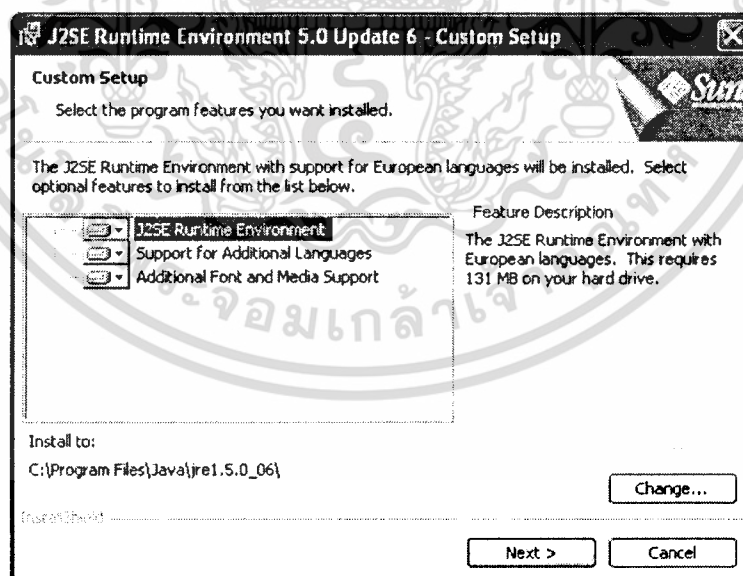
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) รอการติดตั้งพีเจอร์ที่เลือกไว้



รูปที่ ก.4 การติดตั้งพีเจอร์

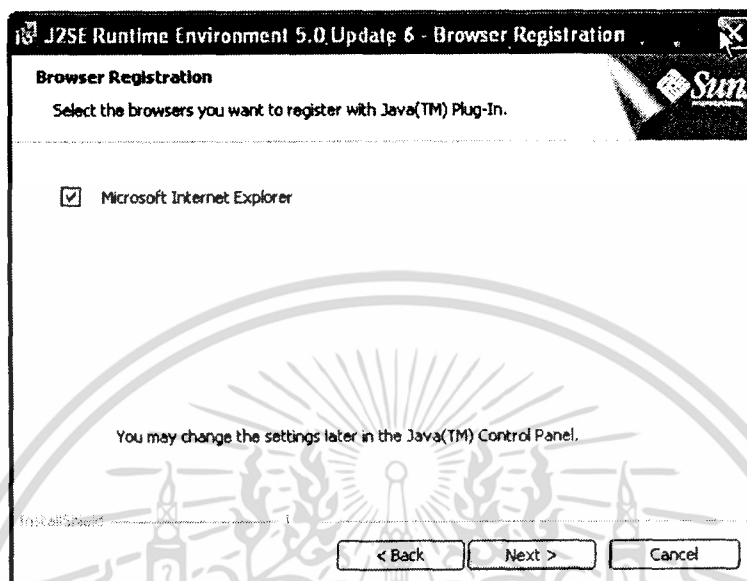
- 8) เลือก J2SE Runtime Environment ซึ่งโดยปกติเป็นค่าที่ถูกกำหนดไว้แล้ว
- 9) กำหนดโฟลเดอร์ที่ต้องการติดตั้ง โดยค่าปกติคือ C:\Program Files\Java\jre1.5.0_06
- 10) คลิกปุ่ม Next > เพื่อทำการติดตั้ง



รูปที่ ก.5 เลือกพีเจอร์และกำหนดไดเรกทอรี

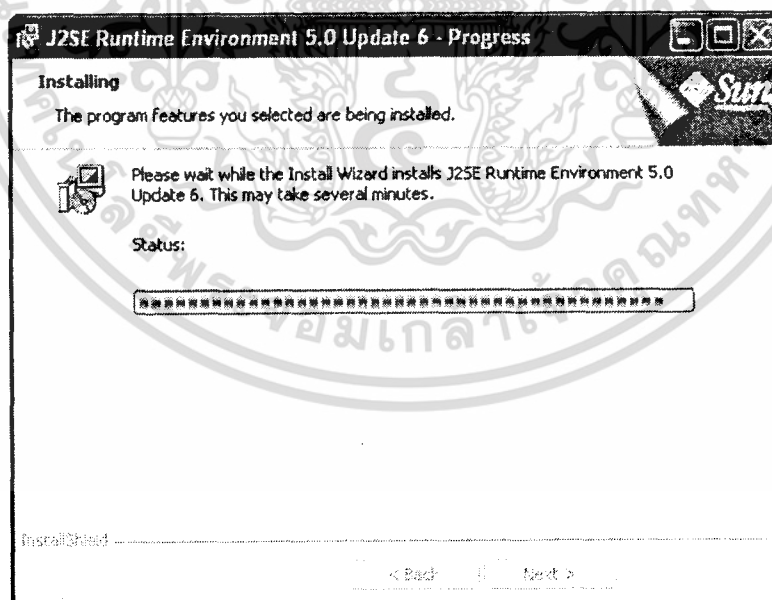
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 11) คลิกให้ปรากฏเครื่องหมายถูกหน้า Microsoft Internet Explorer เพื่อให้ Internet Explorer สามารถรัน Java Applets ได้
- 12) คลิกปุ่ม Next > เพื่อทำการติดตั้ง



รูปที่ ก.6 ติดตั้ง Plug-in ให้กับเบราว์เซอร์

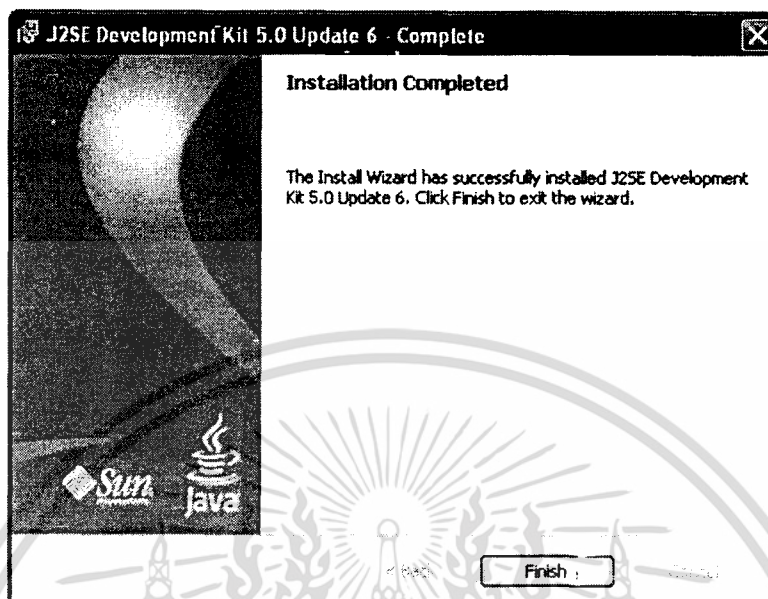
- 13) รอการติดตั้งไฟล์ที่เลือกไว้



รูปที่ ก.7 การติดตั้งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14) คลิกปุ่ม Finish เมื่อเสร็จสิ้นการติดตั้งซอฟต์แวร์

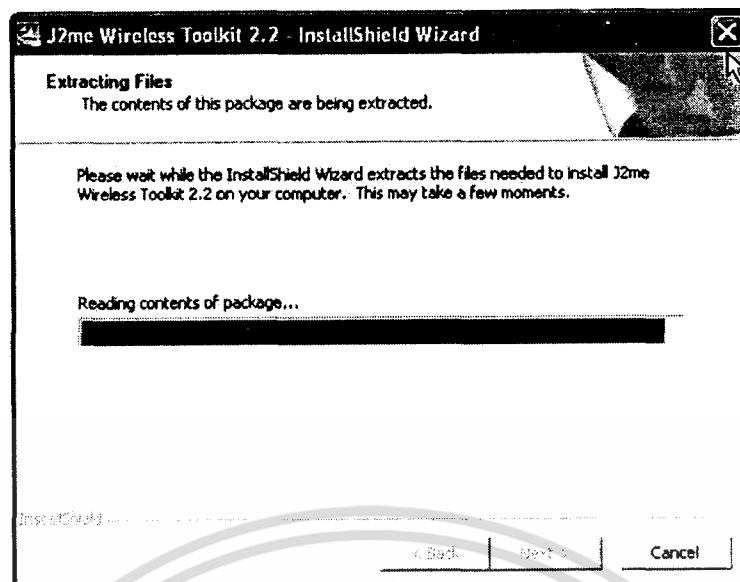


รูปที่ ก.8 เสร็จสิ้นการติดตั้ง J2SE SDK

วิธีการดาวน์โหลดและติดตั้ง J2ME Wireless Toolkit (J2MEWTK)

เครื่องมือในการพัฒนา J2ME Wireless Toolkit (J2MEWTK) สามารถเข้าไปดาวน์โหลดได้ที่ <http://java.sun.com/products/sjwtoolkit/> เมื่อดาวน์โหลดเรียบร้อยแล้ว สามารถทำการติดตั้งได้ตามขั้นตอนดังนี้

- 1) ดับเบิลคลิกไฟล์ `j2me_wireless_toolkit-2_2-windows.exe` ที่ได้ดาวน์โหลดมาเพื่อทำการติดตั้งโปรแกรมลงในเครื่องคอมพิวเตอร์



รูปที่ ก.9 เตรียมไฟล์ต่างๆก่อนติดตั้งซอฟต์แวร์

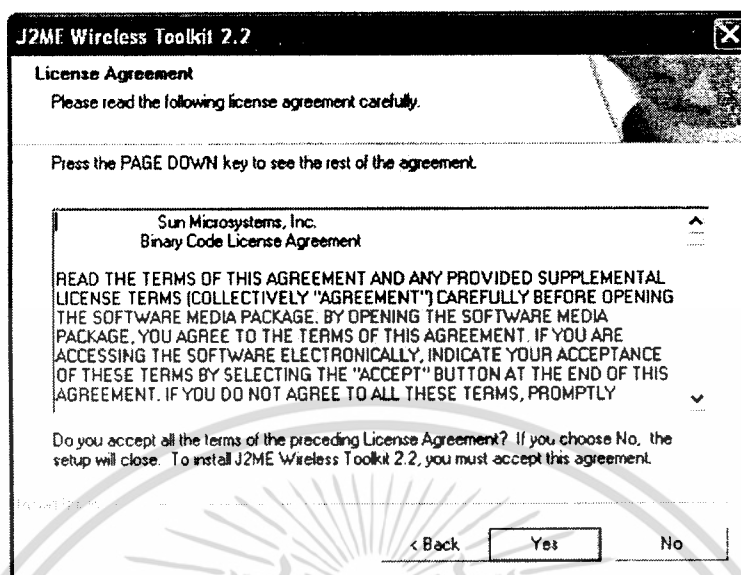
2) คลิกปุ่ม Next > เพื่อไปยังหน้าจอถัดไป



รูปที่ ก.10 หน้าจอต้อนรับสู่การติดตั้งซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) คลิกปุ่ม Yes เพื่อยอมรับข้อตกลงในการติดตั้งซอฟต์แวร์

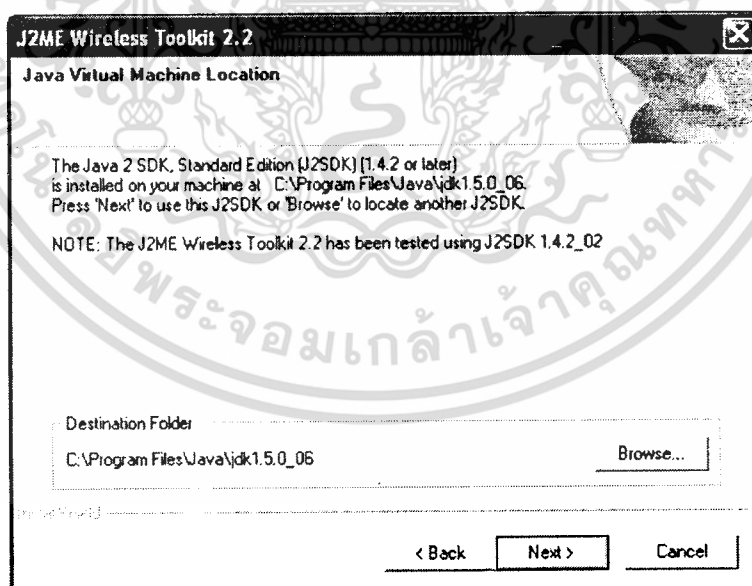


รูปที่ ก.11 ข้อตกลงในการติดตั้งซอฟต์แวร์ J2MEWTK

- 4) กำหนดโฟลเดอร์ที่ได้ติดตั้ง J2SDK โดยค่าปกติคือ
Files\Java\jdk1.5.0_06

C:\Program

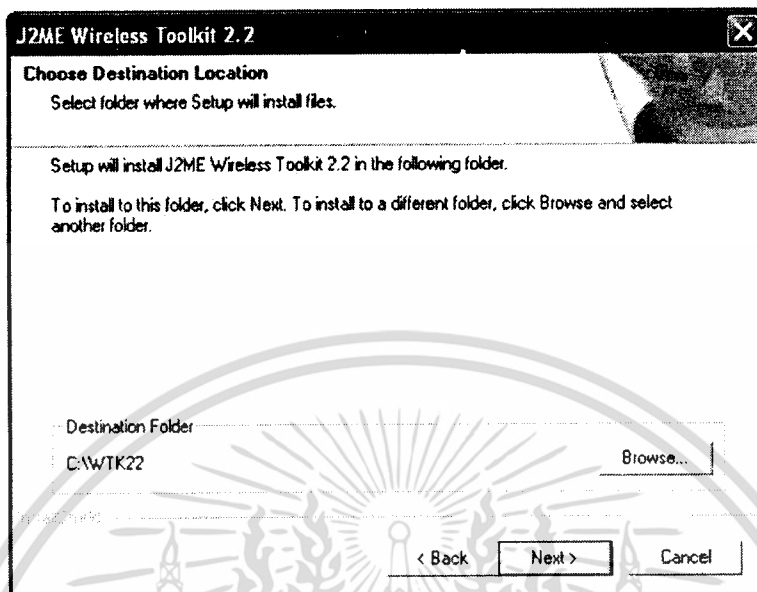
- 5) คลิกปุ่ม Next >



รูปที่ ก.12 โฟลเดอร์ที่ได้ติดตั้ง J2SDK

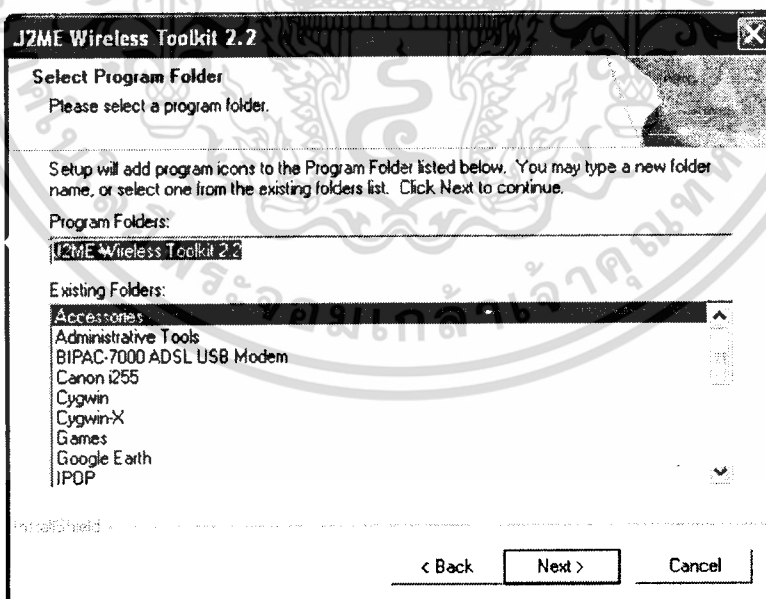
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) กำหนดโฟลเดอร์ที่ต้องการจะติดตั้ง WTK โดยค่าปกติคือ C:\WTK22
- 7) คลิกปุ่ม Next >



รูปที่ ก.13 โฟลเดอร์ที่ต้องการติดตั้ง WTK

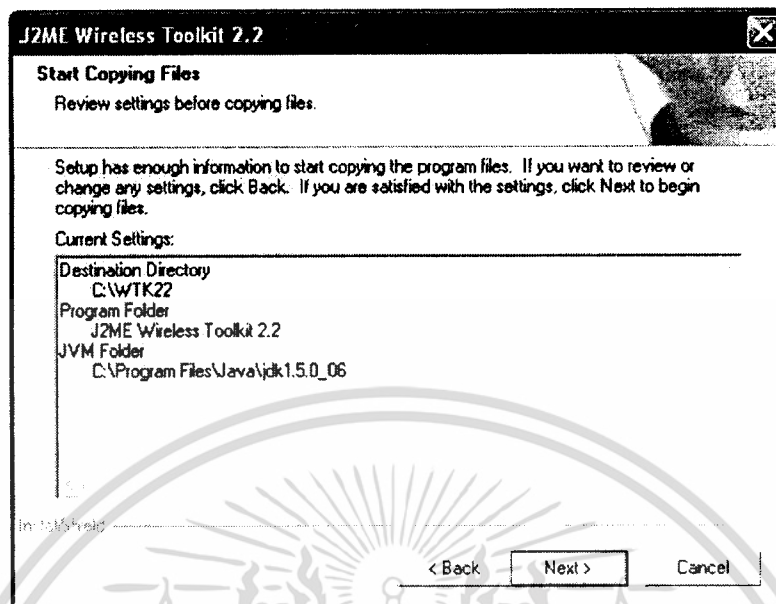
- 8) ตั้งชื่อโฟลเดอร์ที่เก็บโปรแกรม โดยค่าปกติคือ J2ME Wireless Toolkit 2.2
- 9) คลิกปุ่ม Next >



รูปที่ ก.14 ตั้งชื่อโฟลเดอร์ที่ต้องการเก็บโปรแกรม

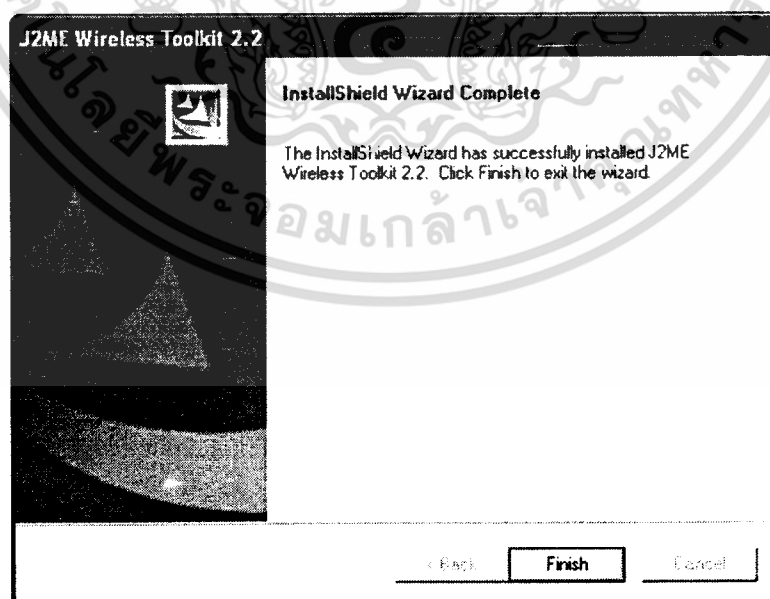
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) คลิกปุ่ม Next > เพื่อทำการติดตั้งซอฟต์แวร์



รูปที่ ก.15 ทำการติดตั้งซอฟต์แวร์

11) คลิกปุ่ม Finish เมื่อเสร็จสิ้นการติดตั้งซอฟต์แวร์



รูปที่ ก.16 เสร็จสิ้นการติดตั้งซอฟต์แวร์ J2MEWTK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการติดตั้งซอฟต์แวร์เรียบร้อยแล้ว J2MEWTK จะถูกติดตั้งอยู่ที่โฟลเดอร์ C:\\WTK22 ซึ่งประกอบไปด้วยโฟลเดอร์ต่างๆ แสดงเป็นตารางดังรูป 2.1

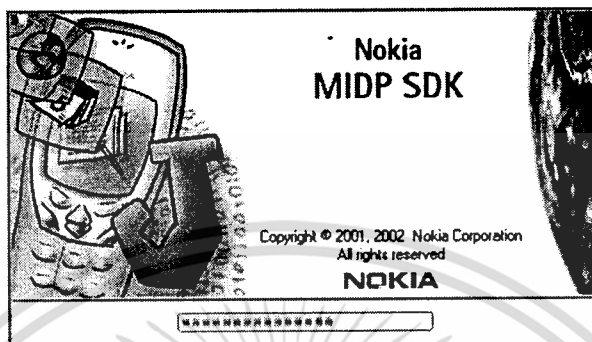
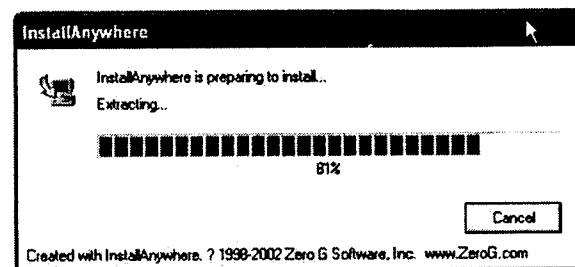
ตารางที่ ก.1 แสดงรายละเอียดในโฟลเดอร์ต่างๆ

โฟลเดอร์	คำอธิบาย
appdb\	เก็บข้อมูล RMS
apps\	เก็บตัวอย่าง MIDlet ที่มาพร้อมกับการติดตั้ง และโปรเจกต์ที่สร้างจาก KToolbar
bin\	ไฟล์ *.bat และไฟล์ *.exe สำหรับเครื่องมือต่างๆ
docs\	เอกสารการใช้งาน WTK และ API ของ MIDP
libs\	API ของ MIDP (อยู่ใน midpapi.zip)
wtklib\	เก็บคุณสมบัติสำหรับแบบจำลองโทรศัพท์มือถือ

วิธีการดาวน์โหลดและติดตั้ง Emulator

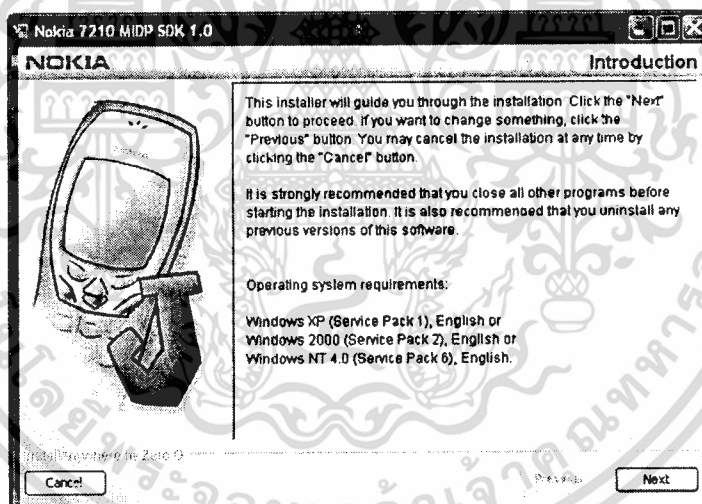
อีมูเลเตอร์ที่จะใช้ในการพัฒนาคือ Nokia รุ่น 7210 โดยสามารถทำการดาวน์โหลดซอฟต์แวร์ได้ที่ <http://www.forum.nokia.com/main/1,6566,030,00.html> เมื่อทำการดาวน์โหลดเรียบร้อยแล้วสามารถทำการเพิ่มอีมูเลเตอร์ที่ต้องการทดสอบเข้าไปใน WTK โดยนำไปใส่ไว้ในโฟลเดอร์ C:\\WTK22\\wtklib\\devices โดยสามารถทำได้ตามขั้นตอนดังนี้

- 1) แยกไฟล์ n7210_jme_sdk_v1_0.zip ที่ดาวน์โหลดมาในโฟลเดอร์ที่ต้องการ
- 2) ดับเบิลคลิกที่ไฟล์ setup.exe เพื่อทำการติดตั้งซอฟต์แวร์



รูปที่ ก.17 เตรียมไฟล์เพื่อทำการติดตั้ง

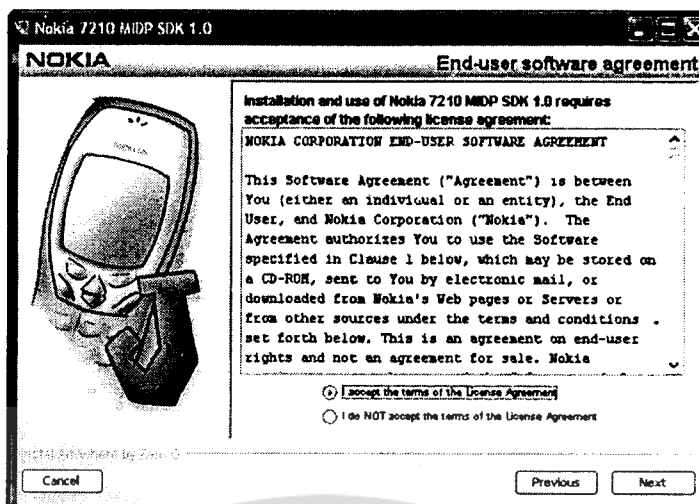
3) คลิกปุ่ม Next >



รูปที่ ก.18 หน้าจอ Introduction

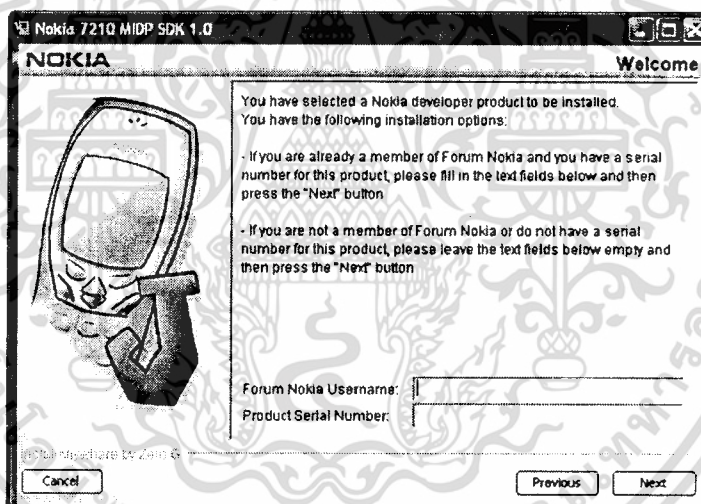
- 4) เลือก "I accept the terms of the License Agreement" เพื่อยอมรับเงื่อนไขในการติดตั้งซอฟต์แวร์
- 5) คลิกปุ่ม Next >

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.19 ข้อตกลงในการติดตั้งซอฟต์แวร์ Nokia 7210 Emulator

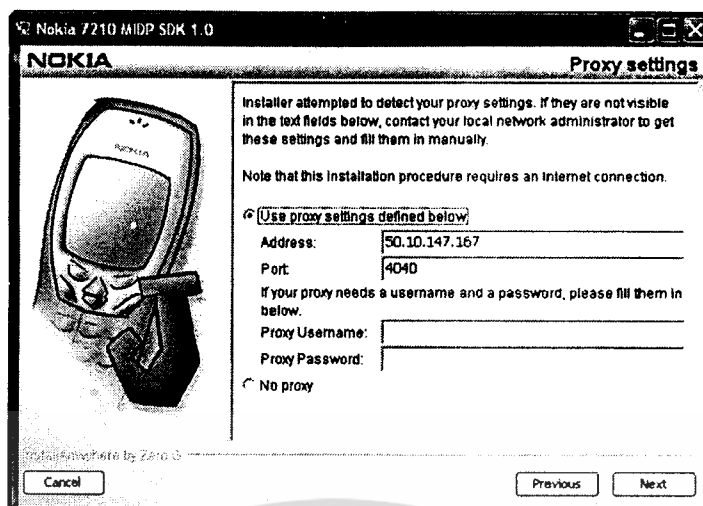
- 6) คลิกปุ่ม Next > (โดยสามารถปล่อยช่อง Forum Nokia Username และ Product Serial Number ให้อยู่ว่างไว้ได้)



รูปที่ ก.20 สอบถามการเป็นสมาชิก

- 7) เลือกว่ามีการเชื่อมต่ออินเทอร์เน็ตแบบใด ซึ่งถ้ามีการใช้ Proxy โปรแกรมจะค้นหารายละเอียดการใช้ให้อัตโนมัติ
- 8) คลิกปุ่ม Next >

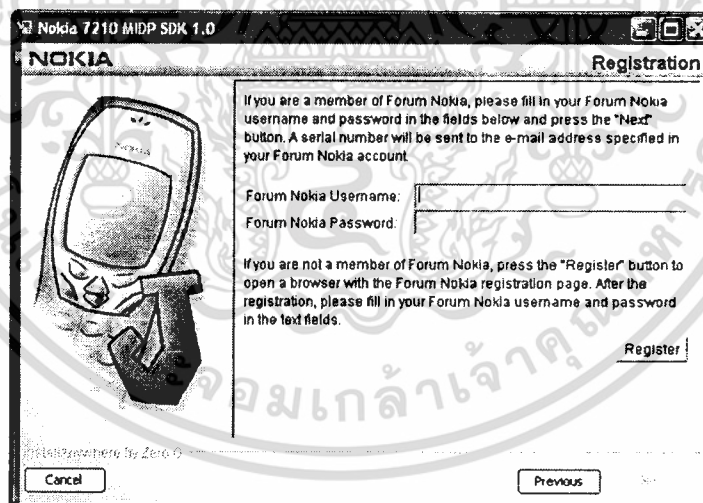
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.21 การเลือกวิธีการเชื่อมต่ออินเทอร์เน็ต

- 9) กรอกข้อมูล Username และ Password โดยถ้าเป็นสมาชิก Forum ของ Nokia อยู่แล้วสามารถใช้ Username และ Password เดิมได้เลย แต่ถ้าไม่ได้เป็นสมาชิก สามารถสมัครสมาชิกได้โดยกดปุ่ม Register เมื่อทำการกรอก Username และ Password เรียบร้อยแล้ว

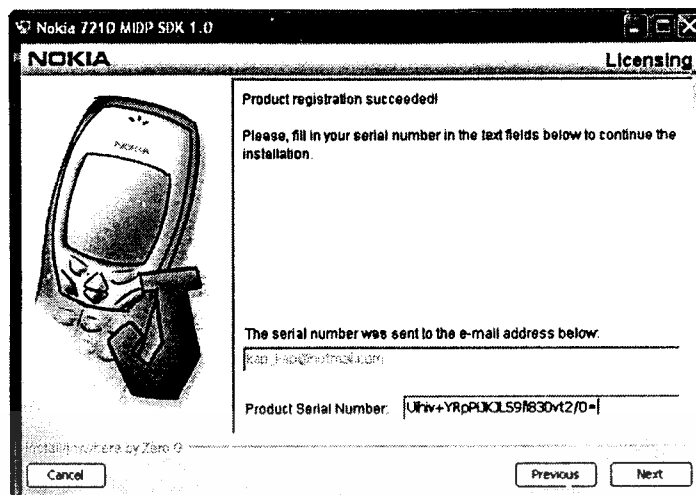
- 10) คลิกปุ่ม Next >



รูปที่ ก.22 กรอกข้อมูล Username และ Password

- 11) กรอกหมายเลข Serial โดยสามารถรับหมายเลข Serial ได้จากอีเมลล์ที่กรอกไว้
- 12) คลิกปุ่ม Next >

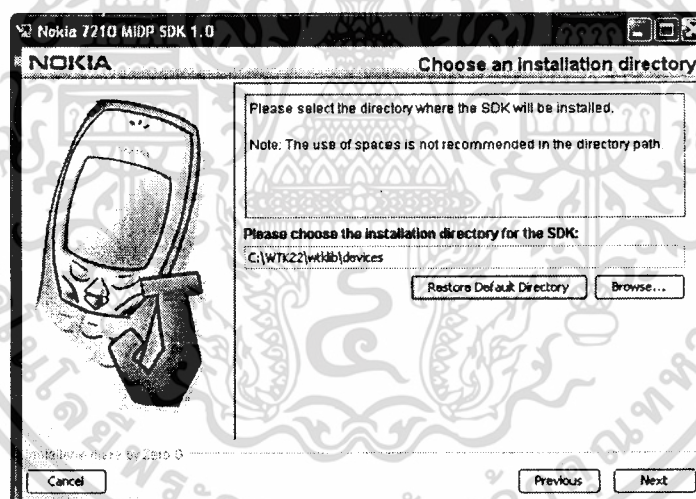
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.23 กรอก Serial Number

- 13) เลือกโฟลเดอร์เพื่อเก็บไฟล์ที่ได้จากการติดตั้ง เนื่องจากการใช้งานร่วมกับ WTK
จึงต้องเก็บไฟล์ไว้ที่ C:\WTK22\wtklib\devices

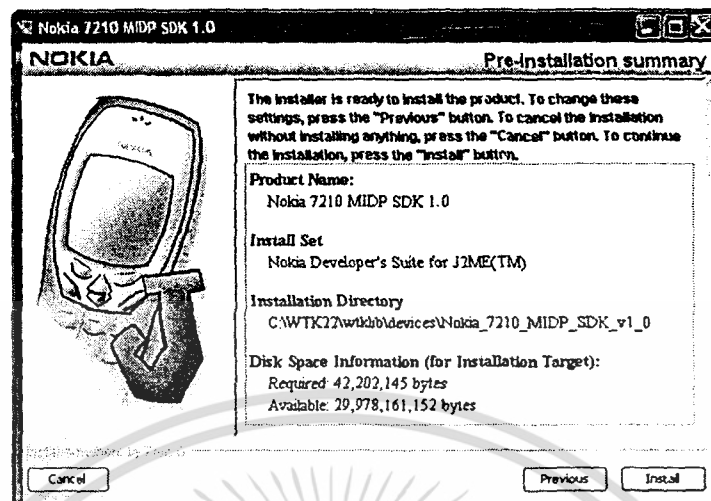
- 14) คลิกปุ่ม Next >



รูปที่ ก.24 กำหนดโฟลเดอร์ที่ต้องการติดตั้ง

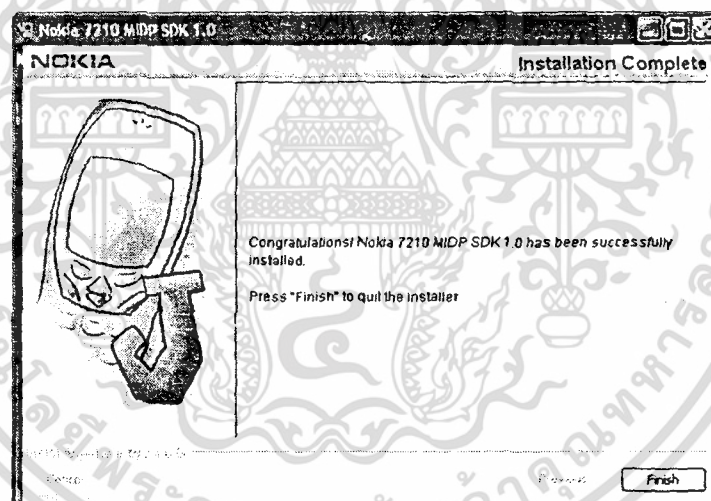
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15) คลิกปุ่ม Install เพื่อทำการติดตั้งซอฟต์แวร์



รูปที่ ก.25 รายละเอียดการติดตั้งซอฟต์แวร์

16) คลิกปุ่ม Finish เมื่อเสร็จสิ้นการติดตั้งซอฟต์แวร์ Nokia 7210 Emulator



รูปที่ ก.26 เสร็จสิ้นการติดตั้งซอฟต์แวร์ Nokia 7210 Emulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการดาวน์โหลดแอปพลิเคชันลงในโทรศัพท์มือถือ

การดาวน์โหลดโปรแกรมลงมือถือสามารถทำได้ 3 วิธี

1. ดาวน์โหลดโดยใช้สาย data link
2. ดาวน์โหลดโดยใช้ port Infrared
3. ดาวน์โหลดโดยใช้ การเชื่อมต่อแบบ GPRS

ในที่นี้ผู้จัดทำขอยกตัวอย่างวิธีการดาวน์โหลดโดยใช้ การเชื่อมต่อแบบ GPRS เราต้องทำการ upload ไฟล์ .jar และ .jad ไปไว้ที่ server โดยไฟล์ทั้งสองนี้มาจาก

C:\WTK21\apps\FuelApp\bin จากนั้นทำการเชื่อมต่อ GPRS จากมือถือ โดยเข้าไปที่เมนู

บริการ >> บัญชี >> บัญชีใหม่ >> พิมพ์ address ของ ไฟล์ที่เราเก็บไว้บน server จากนั้นตั้งชื่อบัญชี >> จัดเก็บ จากนั้นจะปรากฏชื่อบัญชี >> ไปที่ มือถือจะทำการเชื่อมต่อ GPRS >> เลือกดาวน์โหลด แอปพลิเคชัน >> เมื่อดาวน์โหลดเรียบร้อยแล้วให้เลือก จัดเก็บ

เมื่อต้องการเรียกใช้ โปรแกรมประยุกต์นี้ ก็สามารถเข้าไปที่ Application ก็จะพบ

