

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ
Intelligent Bangkok Traffic Router (Front-end)



ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
 ภาควิชาวิศวกรรมคอมพิวเตอร์
 คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ปีการศึกษา 2546

เอกสารนี้...
 เลขที่... 55116
 เลขทะเบียน... 2546
 วัน,เดือน,ปี...
 รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป...
 ไม่ว่าการอื่น...
 ...

ระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ

Intelligent Bangkok Traffic Router



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ

Intelligent Bangkok Traffic Router (Front-end)

คณะผู้จัดทำ 1. นายวุฒินัย กาญจนสร รหัสประจำตัว 43010414
2. นายสันหิ ภัทรวิทย์ รหัสประจำตัว 43010460



สุวิมล กิตติขจร

อาจารย์ที่ปรึกษา

(ดร.สุรินทร์

กิตติขจรกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ (ส่วนแสดงผล)

นายวุฒินัย กาญจนสร 43010414
 นายสันห์ ภัทรวิทย์ 43010460
 ดร.สุรินทร์ กิตติธรรมกุล อาจารย์ที่ปรึกษา
 ปีการศึกษา 2546

บทคัดย่อ

เนื่องจากปัจจุบันเส้นทางการเดินทางสัญจรทางบกภายในกรุงเทพมหานครมีความสลับซับซ้อนมาก และนับยิ่งก็ยิ่งเพิ่มมากขึ้นด้วย ซึ่งเกิดเป็นปัญหาสำหรับผู้ที่ต้องการจะเดินทางในเขตของกรุงเทพมหานครมาก ไม่ว่าจะเป็นนักท่องเที่ยวที่เพิ่งจะเคยเดินทางเข้ามาและยังไม่คุ้นกับสภาพเส้นทางหรือไม่รู้จักเส้นทางเลย รวมทั้งผู้ที่ใช้เส้นทางอยู่เป็นประจำก็อาจจะประสบกับปัญหาการจราจรติดขัดได้ และเป็นการยากที่จะหลีกเลี่ยงสภาพการจราจรแบบนั้น จึงจำเป็นต้องมีเครื่องมือสักชิ้นที่ช่วยอำนวยความสะดวกในค่านี้นี้ และให้ข้อมูลที่จำเป็นเพื่อช่วยในการตัดสินใจและเลือกเส้นทางด้วย ซึ่งจะต้องใช้ได้ทุกที่และทุกเวลาด้วย

ระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ (ส่วนแสดงผล) ก็เป็นส่วนหนึ่งเครื่องมือหนึ่งที่จะช่วยแก้ปัญหาเหล่านี้ได้ ซึ่งจะต้องทำงานร่วมกับระบบค้นหาเส้นทางอัจฉริยะในกรุงเทพฯ (ส่วนประมวลผล) ด้วย โดยที่เครื่องมือนี้จะช่วยในการให้ข้อมูลสภาพเส้นทางและสภาพการจราจรในปัจจุบันภาพแผนที่ เป็นที่รวบรวมข้อมูลข่าวสารเกี่ยวกับการจราจร รวมทั้งยังช่วยในการตัดสินใจเลือกเส้นทางตามความต้องการของผู้ใช้อีกด้วย โดยที่เครื่องมือนี้จะใช้งานผ่านทางอินเทอร์เน็ตทางเครื่องคอมพิวเตอร์ทั่วไป และผ่านทางโทรศัพท์เคลื่อนที่ที่สามารถใช้งานจาวาแอปพลิเคชันและ WAP ได้

ปัญญานิพนธ์ฉบับนี้ได้รวบรวมถึงทฤษฎีที่เกี่ยวข้องกับโครงการ วิธีการออกแบบ เครื่องมือที่ใช้ในการพัฒนาโครงการ ลักษณะการทำงานของโปรแกรม ผลลัพธ์สุดท้ายของโครงการ รวมทั้งปัญหาที่เกิดขึ้นและวิธีการแก้ปัญหาเหล่านั้นๆ

Intelligent Bangkok Traffic Router

Mr.Wutthinai Kanjanasorn

Mr.San Pattharavit

Dr. Surin Kittitornkun Advisor

Academic Year 2003

Abstract

Due to increasing complexity of Bangkok road traffic system, it may cause problems, long travel delay, and eventually traffic jam to many commuters in Bangkok. The problem can be more severe to the tourists and new Bangkok residence who have no experiences in Bangkok traffic. For daily commuters, any tool that can help, decide, and guide how to reach their destinations anytime would be very thankful.

The Bangkok Traffic Router System (BTRS) is one of many tools to alleviate this problem. Only its user interface (front-end) system is described in this project report while the routing engine (back end) is in another one. BTRS can adaptively find the near-optimal routes from source to destination based on the current traffic information. The user can select suggested routes based on his/her criteria. In order to provide services to both static users, who are about to commute, and mobile users on the roads, BTRS can be accessed via normal webpage, and via mobile phones.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษาและคอยดูแลจากหลายๆ ฝ่ายด้วยกัน โดยเฉพาะท่านอาจารย์ที่ปรึกษาที่ให้โอกาสข้าพเจ้าได้ทำปริญญาบัตรฉบับนี้ คอยให้ความเอาใจใส่ คำแนะนำและให้ความช่วยเหลือโดยเสมอมา คือ ดร.สุรินทร์ กิตติธรรมกุล ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวกเพื่อให้การวิจัยและพัฒนาเป็นไปได้ดีด้วยความสะดวก รวมทั้งบริการอินเทอร์เน็ตที่มีความเร็วสูงสำหรับการค้นคว้าหาข้อมูลต่างเพื่อใช้ค้นคว้าหาความรู้และประกอบการพัฒนาโปรแกรมนี้ด้วย

ขอขอบคุณเพื่อนในห้องวิจัยมัลติมีเดียที่คอยให้ความช่วยเหลือและเป็นกำลังใจในการทำงานอย่างยิ่ง

และสุดท้ายต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้าได้มีวันนี้ นั่นคือ บิดาและมารดาอันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดูและคอยอบรมสั่งสอนข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้

นายวุดินัย กาญจนสร
นายสันต์ ภัทรวิทย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูปภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
1.5 ระบบทั้งหมดโดยรวม	3
บทที่ 2 ทฤษฎีและเทคโนโลยีพื้นฐาน	4
2.1 JSP	4
2.1.1 Java Servlet	4
2.1.2 จาภาษา Java Servlet มาเป็น JSP	4
2.1.3 ข้อดีของ JSP	6
2.1.4 โครงสร้างและขั้นตอนการทำงานของ JSP	6
2.2 Java Applet	8
2.2.1 เบื้องต้น	8
2.2.2 กลาส์ Applet	9
2.2.3 Applet Life Cycle	10
2.3 J2ME	12
2.3.1 เบื้องต้น	12
2.3.2 แพลตฟอร์มของจาวาทูโมบายล์อิดิตชั่น	13
2.3.3 MIDP	14
บทที่ 3 เครื่องมือที่ใช้ในการพัฒนาระบบ	15
3.1 Apache Tomcat Webserver 5.0.16	15
3.2 MySQL 4.016 และ MySQL Front 2.5	15
3.3 J2SE และ J2ME	15
3.4 J2ME Wireless Toolkit (WTK)	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Editor	16
บทที่ 4 การออกแบบและพัฒนาระบบ	17
4.1 การออกแบบตารางฐานข้อมูล	17
4.1.1 ตาราง Member	17
4.1.2 ตาราง Matching	17
4.1.3 ตาราง Report	18
4.1.4 ตาราง District	18
4.1.5 ตาราง Board_quest	18
4.1.6 ตาราง Board_ans	19
4.1.7 ตาราง Mapinfo	19
4.2 ขั้นตอนการทำงานส่วนแสดงผลของเว็บไซต์	20
4.2.1 หน้าแรกหรือโฮมเพจ (home.jsp)	21
4.2.2 หน้าสมัครสมาชิกใหม่ (register.jsp)	22
4.2.3 หน้าแก้ไขข้อมูลส่วนตัวของสมาชิก (edit.jsp และ edit_cont.jsp)	23
4.2.4 หน้าล็อกอินหรือหน้าเข้าสู่ระบบสมาชิก (login.jsp)	23
4.2.5 หน้าล็อกเอาต์ออกจากระบบ (logout.jsp)	24
4.2.6 หน้าแสดงแผนที่และรายละเอียดของเส้นทางกรรจราร (map.jsp และ member_map.jsp)	24
4.2.7 หน้าค้นหาเส้นทางกรรจราร (routing.jsp และ member_routing.jsp)	25
4.2.8 หน้าแสดงรายงานอุบัติเหตุและสิ่งกีดขวางกรรจราร (report.jsp และ member_report.jsp)	26
4.2.9 หน้าแสดงรายงานสถิติขออนุญาตนำจล.100 (radio.jsp และ member_radio.jsp)	27
4.2.10 หน้าต้อนรับเข้าสู่ระบบสมาชิกหลังทำการล็อกอิน (member_section.jsp)	28
4.2.11 หน้าแสดงรายชื่อกระทู้ทั้งหมดของเว็บบอร์ด (webboard.jsp)	28
4.2.12 หน้าสร้างกระทู้ใหม่หรือหน้าโพสต์ (addnew.jsp)	29
4.2.13 หน้าแสดงและตอบกระทู้ (view.jsp)	32
4.2.14 หน้าแสดงความผิดพลาดของเซิร์ฟเวอร์ (error.jsp)	34
4.2.15 ไฟล์เก็บส่วนหัวและแถบเมนูด้านข้าง (header.jsp, member_header.jsp และ member_left.jsp)	34
4.2.16 โปรแกรม Java Applet ที่ใช้แสดงแผนที่ (Map.class)	35

4.3 การทำงานของแอปพลิเคชันค้นหาเส้นทางบนโทรศัพท์เคลื่อนที่	36
4.3.1 การทำงานของแอปพลิเคชัน โดยรวม	36
4.3.2 การทำงานของบริการค้นหาเส้นทางจราจร	37
4.3.3 การทำงานของบริการรายงานสภาพการจราจร	39
4.3.4 การทำงานของบริการแก้ไขข้อมูลสมาชิก	41
4.4 การออกแบบแอปพลิเคชันบนโทรศัพท์เคลื่อนที่	42
4.4.1 คลาส TrafficReport	42
4.4.2 คลาส UserScreen	44
4.4.3 คลาส LoginScreen	44
4.4.4 คลาส MemberScreen	45
4.4.5 คลาส NonmemberScreen	45
4.4.6 คลาส SearchsourceScreen	45
4.4.7 คลาส SearchdestScreen	46
4.4.8 คลาส CostScreen	46
4.4.9 คลาส ReportareaScreen	47
4.4.10 คลาส ReportrangeScreen	47
4.4.11 คลาส ResponseScreen	47
4.4.12 คลาส EdituserScreen	48
4.4.13 คลาส EditdistrictScreen	48
4.5 การออกแบบไฟล์ประมวลผลบนฝั่งเซิร์ฟเวอร์ของโทรศัพท์เคลื่อนที่	48
4.5.1 ไฟล์ Server_Authen.jsp	49
4.5.2 ไฟล์ Server_EditSearch.jsp	49
4.5.3 ไฟล์ Server_RouteSearch.jsp	50
4.5.4 ไฟล์ Server_RouteJunction.jsp	50
4.5.5 ไฟล์ Server_RouteResponse.jsp	51
4.5.6 ไฟล์ Server_AreaSearch.jsp	51
4.5.7 ไฟล์ Server_RangeSearch.jsp	51
4.5.8 ไฟล์ Server_ReportResponse.jsp	52
4.5.9 ไฟล์ Server_EditUpdate.jsp	52
บทที่ 5 ผลการทดลอง/ทดสอบ	54
5.1 ผลการทดลองในส่วนของเว็บไซต์	54
5.2 ผลการทดลองแอปพลิเคชันบนโทรศัพท์เคลื่อนที่	64
5.2.1 ผลการทดลองการเข้าสู่ระบบ	65
5.2.2 ผลการทดลองบริการค้นหาเส้นทาง	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 ผลการทดลองบริการรายงานสภาพการจราจร	69
5.2.4 ผลการทดลองบริการแก้ไขข้อมูลสมาชิก	70
บทที่ 6 บทวิจารณ์และสรุป	72
6.1 บทวิจารณ์และสรุปส่วนฐานข้อมูล	73
6.2 บทวิจารณ์และสรุปส่วนเว็บไซต์	73
6.3 บทวิจารณ์และสรุปส่วนแอปพลิเคชันบนโทรศัพท์เคลื่อนที่	73
ภาคผนวก ก อักษรย่อ	74
ภาคผนวก ข ซอร์ซโค้ดของ applet (Map.jsp)	75
บรรณานุกรม	91



สารบัญตาราง

	หน้า
ตารางที่ 2-1 แสดง packet ของ JMP MIDP	14
ตารางที่ 4-1 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Member	17
ตารางที่ 4-2 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Matching	18
ตารางที่ 4-3 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Report	18
ตารางที่ 4-4 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Report	18
ตารางที่ 4-5 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest	19
ตารางที่ 4-6 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest	19
ตารางที่ 4-7 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest	20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1-1 แสดงระบบทั้งหมดโดยรวม	3
รูปที่ 2-1 โครงสร้างและขั้นตอนการประมวลผลไฟล์ JSP	7
รูปที่ 2-2 ขั้นตอนการประมวลผลไฟล์ JSP ในช่วง translation	8
รูปที่ 2-3 แสดงการเปลี่ยนสถานะของ applet	11
รูปที่ 2-4 แสดงการเปรียบเทียบภาษาจาวาเวอร์ชันต่างๆ	13
รูปที่ 2-5 แสดงสถาปัตยกรรม J2ME และสถาปัตยกรรม MIDP	13
รูปที่ 4-1 แสดงลำดับการใช้งานของเว็บเพจ	21
รูปที่ 4-2 แสดงหน้าตาของ home.jsp	21
รูปที่ 4-3 แสดงไฟล์ชาร์ตของ register.jsp	22
รูปที่ 4-4 แสดงไฟล์ชาร์ตของ edit.jsp และ edit_cont.jsp	23
รูปที่ 4-5 แสดงไฟล์ชาร์ตของ map.jsp และ member_map.jsp	25
รูปที่ 4-6 แสดงไฟล์ชาร์ตของ routing.jsp และ member_routing.jsp	26
รูปที่ 4-7 แสดงไฟล์ชาร์ตของ report.jsp และ member_report.jsp	27
รูปที่ 4-8 แสดง plug in ของ Windows Media Player	28
รูปที่ 4-9 แสดงไฟล์ชาร์ตของ webboard.jsp	29
รูปที่ 4-10 แสดงไฟล์ชาร์ตของ addnew.jsp	32
รูปที่ 4-11 แสดงไฟล์ชาร์ตของ view.jsp	33
รูปที่ 4-12 หน้าตาของ member_header.jsp	34
รูปที่ 4-13 หน้าตาของ header.jsp	34
รูปที่ 4-14 หน้าตาของ member_left.jsp	34
รูปที่ 4-15 รูปแบบคู่อันดับ XY ของโหนด	35
รูปที่ 4-16 แสดงไฟล์ชาร์ต การทำงานของระบบโดยรวม	37
รูปที่ 4-17 แสดงไฟล์ชาร์ตการทำงานของบริการค้นหาเส้นทางจราจร	38
รูปที่ 4-18 แสดงไฟล์ชาร์ตการทำงานของบริการรายงานสภาพจราจร	40
รูปที่ 4-19 แสดงไฟล์ชาร์ตการทำงานของบริการแก้ไขข้อมูลสมาชิก	41
รูปที่ 5-1 แสดงผลการเริ่มเรียกใช้งาน home.jsp	54
รูปที่ 5-2 แสดงผลการเรียก Register.jsp	55
รูปที่ 5-3 หน้าแสดงผลเมื่อเกิดข้อผิดพลาดขึ้น error.jsp	56
รูปที่ 5-4 แสดงผลการเรียก login.jsp	56
รูปที่ 5-5 แสดงผลการเมื่อผ่านการล็อกอิน login.jsp	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5-6 แสดงผลเมื่อต้องการแก้ไขข้อมูล edit.jsp	57
รูปที่ 5-7 แสดงผลเมื่อแก้ไขข้อมูลเรียบร้อยแล้ว	58
รูปที่ 5-8 แสดงผลเมื่อต้องการแก้ไขต่อ edit_cont.jsp	58
รูปที่ 5-9 และ 5-10 แสดงผลเมื่อสมาชิกต้องการรายงานจราจร member_report.jsp	59
รูปที่ 5-11 แสดงผลเมื่อเรียก Live Radio Report member_radio.jsp	60
รูปที่ 5-12 แสดงผลเมื่อเรียก webboard webboard.jsp	61
รูปที่ 5-13 แสดงผลเมื่อเรียกต้องการสร้างกระทู้ใหม่ addnew.jsp	62
รูปที่ 5-14 แสดงผลเมื่อเรียกดูกระทู้ view.jsp	62
รูปที่ 5-15 แสดงผลการวาดแผนที่พร้อมกับแสดงรายละเอียดเส้นทาง member_map.jsp	63
รูปที่ 5-16 แสดงหน้าขั้นตอนการเลือกประเภทผู้ใช้งาน	64
รูปที่ 5-17 แสดงหน้าขั้นตอน การเข้าสู่ระบบของสมาชิก และข้อผิดพลาดที่เกิดขึ้น	65
รูปที่ 5-18 แสดงหน้าขั้นตอน การเข้าสู่ระบบของผู้ใช้บริการทั่วไป	66
รูปที่ 5-19 แสดงหน้าขั้นตอนของบริการค้นหาเส้นทางของสมาชิก	67
รูปที่ 5-20 แสดงหน้าขั้นตอนบริการรายงานสภาพการจราจรของสมาชิก	68
รูปที่ 5-21 แสดงหน้าขั้นตอนบริการแก้ไขข้อมูลของสมาชิก	70



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันเส้นทางจราจรในเขตกรุงเทพมหานครนั้นมีความซับซ้อนมากขึ้น ทำให้ยากต่อการเดินทาง โดยเฉพาะอย่างยิ่งสำหรับคนต่างจังหวัดและนักท่องเที่ยวหรือแม้กระทั่งคนในเขตพื้นที่ที่ท่องเที่ยว การที่จะรู้ถึงรายละเอียดสภาพการจราจรในขณะเวลานั้นๆทำได้ค่อนข้างลำบาก และการที่จะเลือกใช้เส้นทางในการเดินทางเพื่อความสะดวกรวดเร็วนั้นก็ย่อมยากลำบากตามไปด้วย ถ้ามีเครื่องมือที่จะมาช่วยอำนวยความสะดวกที่จะสามารถช่วยค้นหาเส้นทางในการเดินทางเพื่อความรวดเร็วในการเดินทางและบอกระยะทางที่ใช้ในการเดินทางที่สั้นที่สุดได้ หรือรายงานสภาพการจราจรขณะนั้น ก็จะทำให้ผู้ที่ไม่คุ้นเคยกับเส้นทางจราจรสามารถเดินทางไปยังจุดหมายปลายทางได้ด้วยความสะดวก รวดเร็ว และประหยัดค่าใช้จ่าย

และในปัจจุบันอินเทอร์เน็ตได้เข้ามามีบทบาทในชีวิตประจำวันของเราค่อนข้างสูง หรือแทบจะว่าเป็นสิ่งอำนวยความสะดวกที่จำเป็นสำหรับชีวิตประจำวันก็ว่าได้ เพราะอินเทอร์เน็ตเป็นสื่อที่ช่วยให้สามารถทำการติดต่อสื่อสารได้อย่างรวดเร็ว สะดวกสบายและมีข้อมูลข่าวสารอยู่มากมาย ด้วยเหตุนี้จึงได้พัฒนาโปรแกรมการค้นหาเส้นทางและให้บริการข้อมูลข่าวสารการจราจรโดยผ่านระบบอินเทอร์เน็ต ซึ่งช่วยให้ผู้ที่ใช้สามารถค้นหาและศึกษาเส้นทางได้ก่อนที่จะเดินทางมายังเขตกรุงเทพมหานครฯ โดยไม่ต้องสนกับเส้นทางอื่น

ยิ่งไปกว่านั้นเครื่องมือสื่อสารอย่างโทรศัพท์เคลื่อนที่ก็ได้รับความนิยมมากขึ้น โดยเข้ามามีบทบาทอย่างมากในชีวิตประจำวัน ซึ่งในปัจจุบันนี้แทบทุกคนจะมีโทรศัพท์เคลื่อนที่เป็นของตัวเอง ดังนั้นถ้าหากว่าเราสามารถนำโทรศัพท์เคลื่อนที่ในการค้นหาเส้นทางในการเดินทางได้ในขณะที่เรากำลังอยู่ระหว่างการเดินทางนั้น ก็จะเป็นการเพิ่มความยืดหยุ่นมากขึ้นในการใช้งาน ก็สามารถทำการค้นหาเส้นทางได้ทุกขณะเวลาและสถานที่

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อให้ข้อมูลข่าวสารด้านการจราจรภายในเขตกรุงเทพมหานครฯ ทั้งด้านข้อมูลแผนที่และสภาพการจราจร
2. เพื่อช่วยตัดสินใจเลือกเส้นทางในการเดินทางจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งที่สะดวกต่อผู้ต้องการเดินทางได้อย่างมีประสิทธิภาพ ทั้งด้านระยะทาง ความสะดวกรวดเร็วหรือเวลาในการเดินทาง และค่าใช้จ่ายที่ใช้ในการเดินทาง ผ่านทางอินเทอร์เน็ตและโทรศัพท์เคลื่อนที่
3. เพื่อให้ผู้ที่ต้องการ ข้อมูลข่าวสารด้านเส้นทางจราจรสามารถทำการค้นหาข้อมูลที่ต้องการได้ทุกที่ทุกเวลา โดยผ่านทางโทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อเป็นกระดานสนทนาแลกเปลี่ยนข่าวสารและข้อคิดเห็นด้านการจราจรแก่สมาชิก

1.3 ขอบเขตของโครงการ

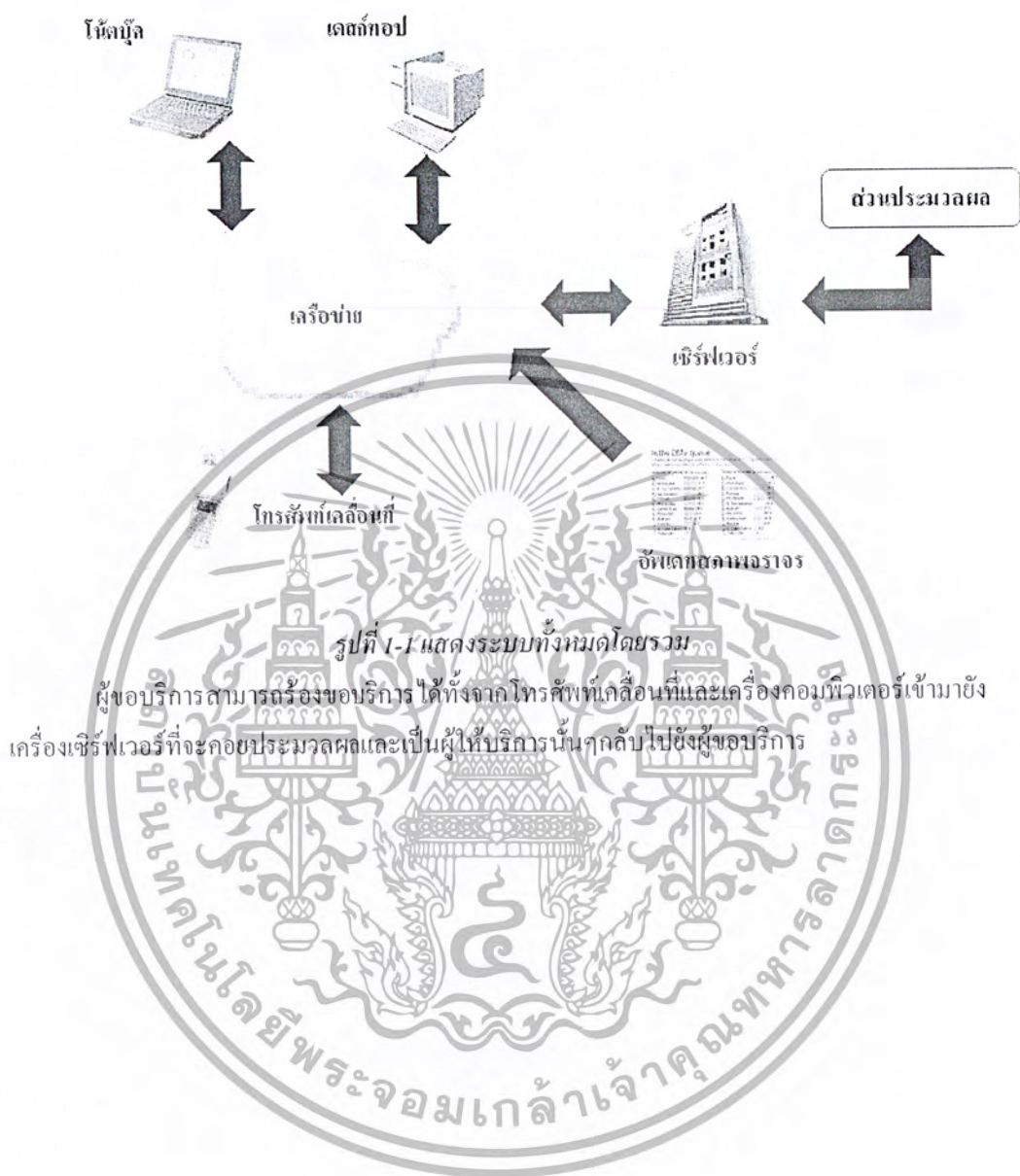
1. สามารถทำการค้นหาและแสดงผลเส้นทางจราจรจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้ โดยสามารถระบุตำแหน่งนั้นๆ ได้ทั้งชื่อถนน ชื่อแยกและสถานที่ โดยที่ตัวแปรการค้นหา นั้นขึ้นอยู่กับระยะทางที่สั้นที่สุด, ระยะเวลาที่ใช้ในการเดินทางหรือค่าใช้จ่ายที่ใช้ในการเดินทาง
2. ระบบสมาชิกช่วยเก็บข้อมูลของสมาชิกแต่ละคน เพื่อความรวดเร็วในการใช้งาน เช่น จุดเพื่อการค้นหาเส้นทางทั้ง 2 จุดที่สมาชิกได้เคยให้ไว้ และข้อมูลอื่นๆของสมาชิกที่จำเป็น
3. สามารถขอข้อมูลและแสดงผลสภาพเส้นทางจราจรรวม รวมทั้งค้นหาเส้นทางจราจรระหว่างตำแหน่ง 2 ตำแหน่ง ได้ทั้งทางเครื่องคอมพิวเตอร์และโทรศัพท์เคลื่อนที่
4. กระดานข่าวรวบรวมข้อมูลข่าวสารด้านการจราจร (Webboard) แก่สมาชิก
5. ฟังสถานีวิทยุ จส.100 ออนไลน์ทางอินเทอร์เน็ต
6. มีการเชื่อมโยง (Link) สำหรับเชื่อมโยงไปหาเว็บไซต์อื่นที่เกี่ยวข้องกับการจราจร
7. สามารถแสดงรายละเอียดเส้นทางจราจรได้ทุกๆเส้นทาง ทั้งสภาพการจราจรและรายงานอุบัติเหตุ
8. สามารถแสดงแผนที่การจราจรภายในเขตกรุงเทพมหานครได้ พร้อมด้วยรายละเอียดของสภาพการจราจร ณ บริเวณนั้นๆ
9. สำหรับการใช้งานด้านแผนที่นั้น สามารถเลือกแสดงข้อมูลจากแผนที่ได้โดยการเลื่อนเมาส์ไปบนตำแหน่งที่ต้องการ

1.4 วิธีการดำเนินงาน

1. ค้นหาข้อมูลที่จะใช้ในการทำงานและระบบที่มีการใช้งานจริงอยู่แล้ว
2. นำข้อมูลที่ได้มาวิเคราะห์หาความต้องการของระบบ ออกแบบระบบงานและจัดหาเครื่องมือที่เหมาะสมกับงาน
3. พัฒนาโปรแกรม
4. นำโปรแกรมมาทดสอบใช้กับข้อมูลจริงและหาข้อผิดพลาดเพื่อปรับปรุงแก้ไขให้ถูกต้อง
5. จัดทำเอกสารประกอบ
6. สรุปผลและนำเสนอโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ระบบทั้งหมดโดยรวม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีพื้นฐาน

ในส่วนของการใช้งานนั้นจะแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ส่วนที่ทำงานบนเครื่องคอมพิวเตอร์ และ โทรศัพท์เคลื่อนที่ ดังนั้นจึงต้องมีเครื่องมือที่ใช้ในการพัฒนาต่างกัน มีดังนี้คือ 1) สคริปต์จาวาที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันบนเครื่องคอมพิวเตอร์ (JSP) 2) โปรแกรมจาวาขนาดเล็กที่สามารถถูกรันบนเบราว์เซอร์ของเครื่องคอมพิวเตอร์ได้ (จาวาแอปพเล็ท Java Applet) และ 3) ภาษายาวาที่ใช้พัฒนาแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ (J2ME)

2.1 JSP

2.1.1 Java Servlet

Java Servlet คือ “โปรแกรม” ที่เขียนขึ้นมาเป็นคลาสของภาษายาวาเพื่อทำงานทางฝั่งเซิร์ฟเวอร์ ทำนองเดียวกับ ASP หรือ PHP ซึ่งรันทางฝั่งเซิร์ฟเวอร์เช่นกัน การนำจาวาเซิร์ฟเล็ตมาใช้ประโยชน์จึงไม่แตกต่างกับ ASP หรือ PHP คือ ใช้สำหรับเขียนโปรแกรมในลักษณะเดียวกับ CGI เพื่อทำหน้าที่อ่านข้อมูลที่ได้รับมาจากผู้ชมเว็บไซต์ แล้วเอาข้อมูลนั้นมาประมวลผล จากนั้นจึงส่งผลลัพธ์กลับไปให้แก่ผู้ชมเว็บนั่นเอง

เนื่องจากจาวาเซิร์ฟเล็ตมีจุดเด่นที่สำคัญมากมาย เช่น มีประสิทธิภาพและความเร็วสูงในการทำงาน, สามารถปรับปรุงแก้ไขและพัฒนาได้ง่ายเพราะใช้ภาษายาวาซึ่งเป็นภาษาเชิงวัตถุ (Object-Oriented Language) ในการพัฒนา เป็นต้น จึงเป็นอีกทางเลือกหนึ่งในการพัฒนา CGI และได้รับความนิยมนำมาพัฒนาเว็บแอปพลิเคชันต่างๆ (web application)

2.1.2 จากจาวา Java Servlet มาเป็น JSP

JSP ย่อมาจากจาวา Java Servlet Page เป็นเทคโนโลยีที่ใช้ “สคริปต์” ในการพัฒนาเว็บแอปพลิเคชัน เพื่อทำงานทางฝั่งเซิร์ฟเวอร์ (server-side script) และส่งผลลัพธ์กลับมายังเว็บเบราว์เซอร์เป็นภาษา HTML เหมือนกับเทคโนโลยีอื่นๆ เช่น ASP, PHP, Cold Fusion เป็นต้น

การเขียนสคริปต์ JSP จะใช้ภาษายาวาเป็นหลัก ซึ่งเป็นภาษาที่ได้รับความนิยมอย่างมากภาษาหนึ่ง เนื่องจากมีคุณลักษณะทางภาษาเชิงวัตถุที่มีเทคนิคช่วยในการเขียนโปรแกรมได้ง่ายขึ้น มีความสามารถในการนำส่วนประกอบหรือคอมโพเนนต์ต่างๆ (component) กลับมาใช้งานได้อีก ซึ่งเป็นประโยชน์อย่างยิ่งโดยเฉพาะในการพัฒนาโปรแกรมขนาดใหญ่

ทั้ง JSP และ Java Servlet ล้วนพัฒนาขึ้น โดยบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) เพื่อเอาไว้ใช้สำหรับการสร้างเว็บแอปพลิเคชันด้วยภาษายาวาเช่นเดียวกัน แต่ข้อแตกต่างระหว่าง JSP กับ จาวาเซิร์ฟเล็ตก็คือ JSP เป็นสคริปต์ ฉะนั้นเวลาพัฒนาเว็บแอปพลิเคชันเราจะสามารถเขียนแต่กัคำสั่งของ

JSP แทรกลงไปในบริเวณที่ต้องการ ภายในเอกสาร HTML ได้ทันที โดยการเขียนแท็กเปิด แล้วตามด้วยชอร์ชโค้ด JSP และปิดด้วยแท็กปิด ในลักษณะเดียวกับสคริปต์ ASP, PHP หรือ JavaScript ตัวอย่างเช่น ถ้าต้องการสร้างเว็บแอปพลิเคชันง่ายๆเพื่อส่งพิมพ์ข้อความ “Java Servlet และ JSP” โดยใช้ JSP จะเขียนสคริปต์ออกมาได้ดังนี้

```
<html>
<body>
    <%
        out.println("Java Servlet และ JSP")ว
    %>
</body>
</html>
```

ที่นี้ทางเซิร์ฟเล็ตซึ่งเป็นโปรแกรม ไม่มีชอร์ชโค้ดกันบ้าง เราไม่สามารถเขียนเซิร์ฟเล็ตด้วยวิธีแทรกชอร์ชโค้ดเข้าไปโดยตรงแบบเดียวกับ JSP แต่ต้องนำแท็กคำสั่งของเอชทีเอ็มแอล (HTML) ทั้งหมด มาเขียนในลักษณะคลาส (class) ของภาษาจาวา ฉะนั้นจากตัวอย่างเดิมเดิมที่เขียนด้วย JSP ถ้าเราเปลี่ยนมาเขียนเว็บแอปพลิเคชันเป็นจาวาเซิร์ฟเล็ตแทน จะได้ชอร์ชโค้ดดังนี้

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class hello extends HttpServlet
{
    public void doGet (HttpServletRequest request,HttpServletResponse response)
    throws ServletException, IOException
    {
        reponse.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("Java Servlets และ JSP");
        out.priontln("</body></html>");
    }
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปรียบเทียบกันดูจะเห็นได้ว่าการเขียน JSP ง่ายกว่าจาวาเซิร์ฟเล็ตมาก รวมทั้งการตรวจสอบแก้ไขก็ทำได้สะดวกกว่า ถ้าจะพูดกันตามความเป็นจริงแล้ว JSP ก็คือเซิร์ฟเล็ตที่เขียนออกมาในรูปแบบที่ง่ายขึ้นนั่นเอง สคริปต์ JSP จึงได้รับความนิยมในการใช้สำหรับพัฒนาเว็บแอปพลิเคชันมากกว่าจาวาเซิร์ฟเล็ต

อย่างไรก็ตาม ในขั้นตอนการทำงานจริงนั้น คอมไพเลอร์ก็ยังคงแปลง JSP ไปเป็นจาวาเซิร์ฟเล็ตก่อนอยู่ดี จึงจะสามารถทำงานได้

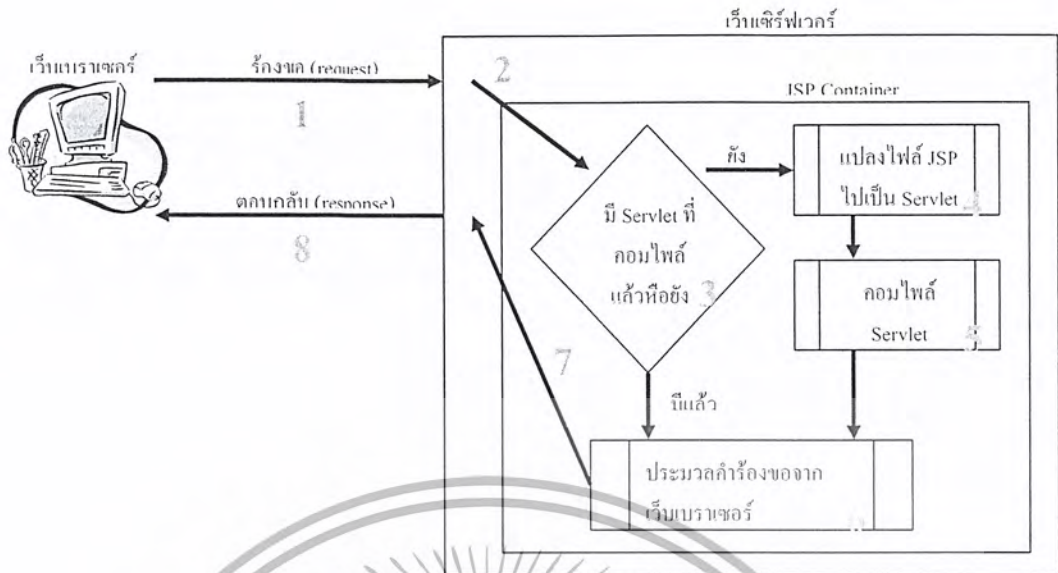
2.1.3 ข้อดีของ JSP

การพัฒนาเว็บแอปพลิเคชันด้วย JSP มีข้อดีต่างๆมากมายดังนี้

- ทำงานโดยไม่ยึดติดแพลตฟอร์มใดๆ คือ JSP ได้สืบทอดคุณสมบัติเด่นของจาวามาอย่างเต็มเปี่ยม คือ การทำงานโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใดๆ ดังนั้นเมื่อพัฒนาในแพลตฟอร์มหนึ่งก็สามารถย้ายไปใช้งานกับแพลตฟอร์มอื่นได้ไม่ยาก
- ใช้งานจาวาเอพีไอ (Java API) ได้หลากหลาย ซึ่งจาวาเอพีไอคือกลุ่มของคลาสที่ช่วยอำนวยความสะดวกในการใช้งานต่างๆ
- นำคอมโพเนนต์กลับมาใช้ได้อีก ไม่ต้องเสียเวลาสร้างใหม่ คือ เราสามารถนำจาวาบีน (Java Bean) มาใช้งานร่วมกับสคริปต์ JSP เพราะจาวาบีนเป็นคอมโพเนนต์ที่เขียนขึ้นมาเพื่อใช้สำหรับทำงานหรือทำหน้าที่อย่างใดอย่างหนึ่ง และสามารถนำกลับมาใช้งานได้เสมอ ดังนั้นเราไม่ต้องเสียเวลาเขียนสคริปต์ JSP เพื่อทำงานนั้นทุกครั้ง จึงพัฒนาเว็บไซต์ได้เร็วขึ้น
- มีความยืดหยุ่นในการใช้งาน คือ ในการเขียนสคริปต์ JSP เราสามารถกำหนดเท็กใหม่ขึ้นมาใช้งานให้เหมาะสมกับความต้องการได้ นอกจากนี้ภาษาที่ใช้เขียนสคริปต์ JSP ไม่ได้จำกัดเฉพาะจาวาเท่านั้น ตามหลักการแล้วเรายังสามารถนำไปใช้งานร่วมกับเอ็กซ์เอ็มแอล (XML) ได้อย่างมีประสิทธิภาพอีกด้วย
- ความปลอดภัย เนื่องจาก JSP มีระบบการจัดการข้อผิดพลาดต่างๆ ไม่ว่าจะเป็นข้อผิดพลาดที่เกิดขึ้นในระหว่างกรเขียนสคริปต์ หรือข้อผิดพลาดที่เกิดขึ้นเมื่อนำสคริปต์ JSP ไปใช้งานจริงล้วนสามารถตรวจสอบและรายงานข้อผิดพลาดได้ทันที

2.1.4 โครงสร้างและขั้นตอนการทำงานของ JSP

สิ่งที่มีบทบาทสำคัญในการทำงานของ JSP ได้แก่ JSP Container (หรือเรียกอีกอย่างหนึ่งว่า JSP Engine) ซึ่งเป็นส่วนประกอบสำคัญที่อยู่ในเว็บเซิร์ฟเวอร์ เพราะทำหน้าที่ควบคุมและประมวลผลไฟล์ JSP ที่มีการร้องขอ (request) เข้ามา และตอบสนอง (response) คำร้องขอนั้นกลับไปยังไคลเอนต์ ขั้นตอนการประมวลผลไฟล์ JSP ทั้งหมดแบ่งเป็น 8 ขั้นตอนดังที่ดังนี้



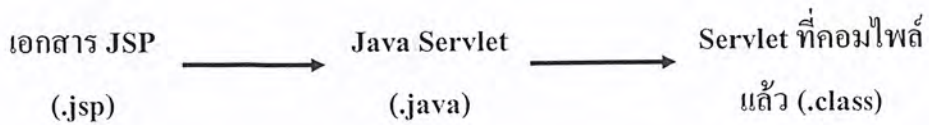
รูปที่ 2-1 โครงสร้างและขั้นตอนการประมวลผลไฟล์ JSP

- 1) ผู้ใช้คลิกเอนต์ส่งคำร้องขอเอกสาร JSP ไปที่เว็บเซิร์ฟเวอร์
- 2) เว็บเซิร์ฟเวอร์ตรวจสอบคำร้องขอ พบว่าเป็นไฟล์ JSP จึงส่งต่อไปให้แก่ JSP Container
- 3) JSP Container ตรวจสอบว่าไฟล์ JSP ที่ร้องขอมานั้นเคยแปลงเป็น Servlet หรือคอมไพล์เป็นไฟล์ .class แล้วหรือยัง ถ้ายังไม่มีการกระทำขั้นตอนที่ 4) ต่อ แต่ถ้ามีอยู่แล้วก็จะตรวจสอบอีกว่า หลังจากแปลงไฟล์เป็น JSP เป็น Servlet และคอมไพล์เป็นไฟล์ .class ครึ่งล่าสุดแล้ว ไฟล์ JSP นั้นมีการเปลี่ยนแปลงแก้ไขหรือเปล่า ถ้ามีการแก้ไขก็จะกระทำขั้นตอนที่ 4) ต่อเช่นกัน แต่ถ้าไม่มีการแก้ไข แสดงว่าไฟล์ JSP นั้นยังคงเดิมไม่เปลี่ยนแปลง จึงไม่มีความจำเป็นที่จะต้องแปลงเป็น Servlet และคอมไพล์ใหม่ ก็ข้ามไปยังขั้นตอนข้อ 6) ได้เลย
- 4) JSP Container แปลงไฟล์ JSP เป็น Java Servlet
- 5) JSP Container คอมไพล์ไฟล์ Java Servlet เป็นไฟล์ .class
- 6) JSP Container ประมวลผลคำร้องขอนั้น
- 7) JSP Container ส่งผลลัพธ์ที่ได้จากการประมวลผลให้แก่เว็บเซิร์ฟเวอร์
- 8) เว็บเซิร์ฟเวอร์ส่งผลลัพธ์นั้นไปยังไคลเอนต์หรือเว็บเบราว์เซอร์อีกทอดหนึ่ง

จากขั้นตอนการประมวลผลไฟล์ JSP ที่แจกแจงข้างต้น สามารถแบ่งออกได้เป็น 2 ช่วงหลักๆ คือ ช่วง translation และช่วง execution โดยช่วง translation ได้แก่ขั้นตอนข้อ 4) และขั้นตอนข้อ 5) ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการแปลงเอกสาร JSP (ไฟล์ .jsp) ให้เป็น Servlet (ไฟล์ .java) จากนั้นก็จะคอมไพล์ไฟล์ Servlet ให้เป็น .class ดังแสดงในรูปที่ 3.3.2



รูปที่ 2-2 ขั้นตอนการประมวลผลไฟล์ JSP ในช่วง translation

ส่วนช่วง execution ได้แก่ขั้นตอนข้อ 6) ซึ่งเป็นการนำเอาไฟล์ .class ที่ได้จากการคอมไพล์ มาประมวลผลหรือทำงานตามคำสั่งของจากไคลเอนต์นั่นเอง

ปกติแล้วกระบวนการทำงานในช่วง translation จะกินเวลาพอสมควร แต่โชคคือว่าช่วง translation จะไม่เกิดขึ้นทุกครั้งที่มีการร้องขอไฟล์ เพราะตราบโคที่ไฟล์ JSP ต้นฉบับไม่มีการเปลี่ยนแปลงอะไร เมื่อมีการร้องขอไฟล์เข้ามาใหม่ ก็ย่อมไม่มีความจำเป็นที่จะแปลงไฟล์เป็น Servlet และคอมไพล์เป็นไฟล์ .class อีก ระบบจะเข้าสู่ช่วง execution ทันทีโดยใช้ไฟล์ .class ที่มีอยู่แล้ว การทำงานจึงรวดเร็วยิ่งขึ้น แต่ถ้ามีการเปลี่ยนแปลงหรือแก้ไขไฟล์ JSP ใหม่ ก็จะต้องเข้าสู่กระบวนการ translation ใหม่ทุกครั้ง

สรุปว่ากระบวนการ translation มีโอกาสจะเกิดขึ้นได้ 2 กรณี กรณีแรกคือ ไฟล์ JSP ที่ร้องขอมาเป็นไฟล์ใหม่ที่ยังไม่เคยแปลงหรือคอมไพล์มาก่อน กับอีกกรณีคือ ไฟล์ JSP ที่ร้องขอมาเคยผ่านการแปลงและคอมไพล์มาแล้ว แต่ภายหลังนั้นมีการเปลี่ยนแปลงแก้ไขไฟล์ JSP นั้นไปจากเดิม

2.2 Java Applet

2.2.1 เบื้องต้น

Applet คือ โปรแกรมขนาดเล็กที่สร้างขึ้นด้วยภาษาจาวา สามารถถูกเรียกจากใน HTML page ให้ทำงานเป็นส่วนหนึ่งของเว็บเพจนั้น กล่าวอีกนัยหนึ่งได้ว่าเป็นโปรแกรมที่ถูกส่งไปกับ HTML page เพื่อไปทำงานภายใต้เบราว์เซอร์ที่มีตัวแปลภาษาจาวา (Java Interpreter) บนเครื่องที่รับ HTML page นั้นๆ ไปสังเกตว่าภาษาจาวาที่เป็นแอปพลิเคชันจะทำงานภายใต้ Java Interpreter โดยไม่จำเป็นต้องอาศัยโปรแกรมอื่น (จึงเรียกว่าเป็น standalone โปรแกรม) และสามารถควบคุมการดำเนินงานของตัวเองได้ แต่โปรแกรมที่เป็น applet จะทำงานภายใต้เบราว์เซอร์โดยไม่สามารถควบคุมการดำเนินงานของตัวเองได้ทั้งหมด

คลาส Applet เก็บอยู่ในแพ็คเกจ java.applet ที่มากับ JDK (Java Development Kit) ใช้สำหรับสร้าง applet ทุก โปรแกรมของ applet ต้องมีประโยค “import java.applet.*;” เพื่อให้มองเห็นคลาส applet และประโยค “import java.awt.*” เพื่อให้คอมไพเลอร์มองเห็นแพ็คเกจที่เรียกว่า Abstract Window Toolkit (AWT) ซึ่งเป็นที่รวบรวมของคลาสเกี่ยวกับกราฟิกและ GUI (Graphic User Interface) โดยทั่วไปเราจะไม่สร้าง instance ของคลาส applet ขึ้นใช้โดยตรง เพราะจะได้ applet ที่ไม่สามารถทำอะไรได้เลย แต่จะใช้งานคลาส Applet โดยขยายไปเป็นคลาสลูก โดยทำการ override methods บางตัวให้ applet ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นทำงานตามที่เราร้องการ สังเกตว่าโปรแกรมของ applet ไม่จำเป็นต้องมี main() เป็นจุดเริ่มต้นการทำงาน (มีก็ได้แต่จะไม่ถูกเรียกใช้) เพราะ applet มีช่วงชีวิตที่ถูกควบคุมโดยเบราว์เซอร์

โปรแกรมของ applet เมื่อถูกคอมไพล์แล้วจะได้ไฟล์เป็น .class (อาจจะมีย่อยไฟล์ขึ้นอยู่กบวิธีการเขียน) แต่จะไม่สามารถทำงานได้โดย Java Interpreter (Java.exe) การเรียกให้ applet ทำงานนั้นต้องทำงานจากภายใน HTML page โดยการใช้ Applet Tag และ applet นั้นจะทำงานภายในสภาพแวดล้อมของเบราว์เซอร์ อย่างเช่น Netscape Navigator , Microsoft Internet Explorer หรือ Hot Java ซึ่งทั้งหมดนั้นเป็นเบราว์เซอร์ที่มี Java Interpreter อยู่ภายใน หรือโปรแกรมสำหรับใช้ทดสอบ applet ที่มากับ JDK หรือ appletviewer

ตัวอย่างการสร้าง applet โดยมีโปรแกรมของ applet อยู่ในไฟล์ HelloA.java ดังนี้

```
import java.awt.*;
import java.applet.*;

public class HelloA extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("Hello",25,50);
    }
}
```

เมื่อโปรแกรมถูกคอมไพล์แล้วจะได้ไฟล์ HelloA.class นอกจากนี้จะต้องมี HTML Page ที่มี Applet Tag ที่เรียกว่า HelloA.class ให้ทำงานอยู่ในไฟล์ HelloA.html ดังนี้

```
<html>
<body>
    This is an Applet.
    <applet code="HelloA.class" width=200 height=200></applet>
</body>
</html>
```

2.2.2 คลาส Applet

คลาส applet จะถูกกำหนดไว้ใน java.applet.apckage ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

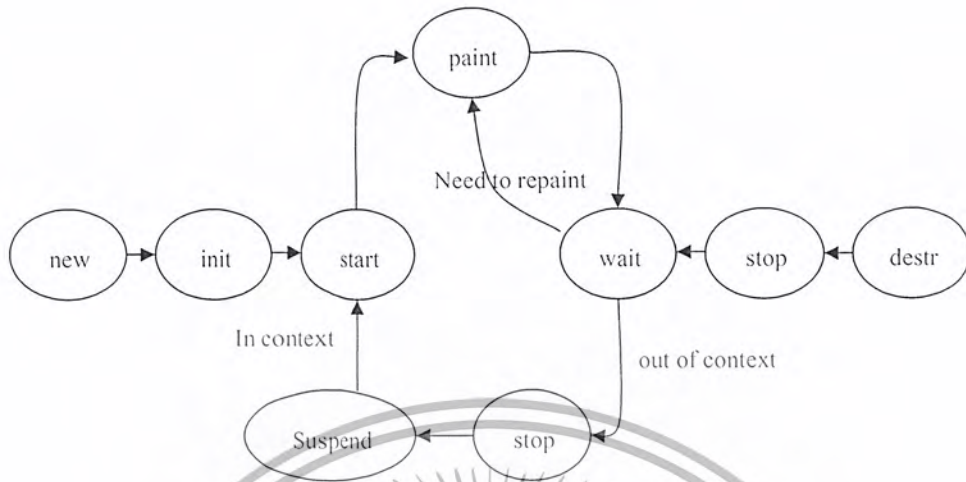
Public class Applet extends Panel
{
    public void destroy();
    public AppletContext getAppletContext();
    public String getAppletInfo();
    public AudioClip getAudioClip(URL url);
    public AudioClip getAudioClip(URL url,String name);
    public URL getCodeBase();
    public URL getDocumentBase();
    public Image getImage(URL url);
    public Image getImage(URL url,String name);
    public Locale getLocale(); //Overrides Component
    public String getParameter(String name);
    public String[][] getParameterInfo();
    public void init();
    public Boolean isActive();
    public void play(URL url);
    public void play(URL url,String name);
    public void resize(int width,int height); // Override Component
    public void resize(Dimension d)// Overrides Component
    public final void setStub(AppletStub stub);
    public void showStatus(String msg);
    public void start();
    public void stop();
}

```

2.2.3 Applet Life Cycle

ช่วงชีวิตของ applet แบ่งออกเป็นสถานะต่างๆ ได้ดังนี้ init, start, paint, wait, suspended, stop และ destroy ทุกๆ applet จะดำเนินการผ่านสถานะเหล่านี้ โดยบางส่วนจะเป็นไปอย่างอัตโนมัติและบางส่วนอยู่ในการควบคุมของผู้ใช้ที่เรียกดู applet นั้น ช่วงชีวิตของ applet แสดงเป็นรูปการเปลี่ยน

สถานะได้ดังนี้



รูปที่ 2-3 แสดงการเปลี่ยนสถานะของ applet

จากรูปเมื่อเริ่มต้น applet ถูกสร้าง โดยการ new() เหมือน instances ที่ทั่วไป ก็จะเริ่มทำงานทันทีคือ

1. เข้าสู่สถานะ init แล้วผ่านเข้าสู่สถานะ start จากนั้นจะเข้าสู่สถานะ paint และเข้าสู่สถานะ wait ซึ่งจะหยุดรอเหตุการณ์ (Event) หรือสัญญาณจากผู้ใช้ (คืออินพุตที่รับเข้าไปทางเมาส์ คีย์บอร์ด หรือ GUN) ที่จะเข้ามาให้เกิดการทำงานของ method ที่กำหนดไว้สำหรับจัดการกับเหตุการณ์นั้น เมื่อเสร็จแล้วจะกลับเข้าสู่สถานะ wait เพื่อรอเหตุการณ์อื่นต่อไป
2. ระหว่างที่ applet อยู่ในสถานะ wait อยู่ หากผู้ใช้เปลี่ยนแปลงเว็บเพจจนทำให้
 - a. พื้นที่แสดงผลของ applet นั้นต้องมีการถูกวาดใหม่ เบราเซอร์จะออกคำสั่งให้ applet ทำการวาดอีกครั้งโดยเรียก repaint() ซึ่งจะส่งผลให้ applet เข้าสู่สถานะ paint และทำการวาดพื้นที่แสดงผลของ applet นั้นอีกครั้ง เมื่อเสร็จแล้วจะกลับเข้าสู่สถานะ visit เพื่อรอเหตุการณ์อื่นต่อไป
 - b. พื้นที่แสดงผลของ applet นั้นไม่ปรากฏในเว็บเพจปัจจุบัน จะทำให้เบราเซอร์ออกคำสั่งให้ applet ผ่านเข้าสู่สถานะ stop และไปสู่สถานะ suspended ซึ่ง applet จะหยุดรอในสถานะนี้โดยไม่ยอมรับอินพุตจากผู้ใช้ และไม่ได้ตอบใดๆจนกว่าพื้นที่แสดงผลของ applet นั้นจะถูกทำให้กลับมาอีกครั้ง เบราเซอร์ก็จะออกคำสั่งให้ applet เข้าสู่สถานะ start, paint และ wait เพื่อเริ่มทำงานอีก
3. ระหว่างที่ applet อยู่ในสถานะ wait หากผู้ใช้กดปุ่มปิดของเบราเซอร์เพื่อหยุดการทำงาน จะทำให้ applet เข้าสู่สถานะ stop และไปสู่สถานะ destr แล้วสิ้นสุดการทำงาน

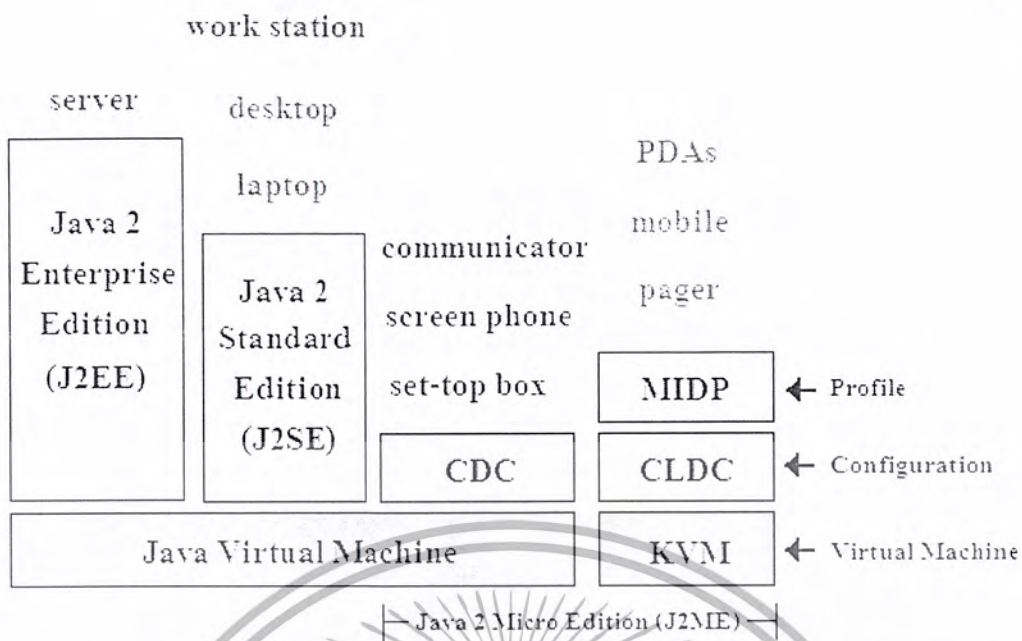
ในการสร้างโปรแกรมของ applet เราสามารถกำหนดรูปแบบการทำงานในแต่ละสถานะต่างๆที่กล่าวมา เมื่อ applet ดำเนินการผ่านคลาส Applet ก็จะมี methods ชื่อเหมือนกับสถานะคือ init(), start(), paint(), stop() และ destroy() สำหรับกำหนดการทำงานเมื่อตอน applet นั้นเข้าสู่สถานะนั้นๆ ถ้าตรวจสอบดูจะพบว่า paint() ไม่ได้ถูกกำหนดไว้ในคลาส applet แต่ได้รับการถ่ายทอดมาจากคลาส Component และสังเกตว่าจะไม่มี method สำหรับสถานะ suspend และ wait เนื่องจากการเข้าและออกจากสถานะ suspended นั้นอยู่ในการควบคุมของเบราวเซอร์

2.3 J2ME

2.3.1 เบื้องต้น

J2ME (Java 2 Micro Edition) เป็นภาษาจาวาเวอร์ชันปัจจุบันที่พัฒนาโดยบริษัท Sun Microsystems ออกแบบเพื่อติดตั้ง ใช้งานกับแอปพลิเคชันในอุปกรณ์ที่มีขนาดเล็กและมีทรัพยากรของระบบจำกัด เช่น โทรศัพท์มือถือและคอมพิวเตอร์ขนาดเล็กแบบพกพา (Personal Digital Assistants) เป็นต้น J2ME เป็นภาษาเชิงวัตถุที่มีจุดเด่นตรงที่สามารถนำไปรันในหลายๆ ระบบปฏิบัติการได้ เพราะในแพลตฟอร์มของ J2ME จะมี Java Virtual Machine ที่ทำหน้าที่แปลความหมายไบต์โค้ดหรือคลาสจาวาที่ถูกคอมไพล์แล้วไปเป็นภาษาเครื่องของแต่ละระบบปฏิบัติการอีกที

J2ME เป็นภาษาจาวามีขนาดเล็กที่สุด เพื่อที่จะสามารถติดตั้งบนอุปกรณ์ขนาดเล็กได้ กลุ่มคลาสของ J2ME แบ่งได้เป็น 2 กลุ่ม โดยกลุ่มแรกเป็นคลาสที่พัฒนาขึ้นมาใหม่ และกลุ่มที่สองเป็นคลาสบางส่วนที่นำมาจาก J2SE สาเหตุที่ไม่สามารถนำคลาสทั้งหมดของ J2SE มาใช้งานได้เพราะคลาสบางตัวต้องใช้ทรัพยากรของระบบมากและมีรูปแบบที่ไม่เหมาะสมกับอุปกรณ์ เช่น คลาส AWT (Abstract Window Toolkit) ซึ่งออกแบบมาเพื่อใช้งานกับระบบปฏิบัติการวิโดวส์ เป็นต้น



รูปที่ 2-4 แสดงการเปรียบเทียบภาษาจาวาเวอร์ชันต่างๆ J2EE (Java 2 Enterprise Edition) ทำงานบนเครื่องเซิร์ฟเวอร์ ใช้สำหรับองค์กรที่รองรับระบบงานใหญ่ และมีไคลเอนต์จำนวนมาก J2SE (Java 2 Standard Edition) ทำงานบนเครื่องคอมพิวเตอร์ประเภทเดสก์ทอปและเวิร์กสเตชัน และ J2ME (Java 2 Micro Edition) เป็นภาษาจาวาเวอร์ชันเล็กที่สุด ทำงานบนอุปกรณ์ประเภทมือถือ (Hand-Held Devices) เช่น โทรศัพท์มือถือและคอมพิวเตอร์ขนาดพกพา (Personal Digital Assistants) สถาปัตยกรรม J2ME และ MIDP

Profile	MID Profile
Configuration	CLDC
Java Virtual Machine	K Virtual Machine
Host Operating System	Host Operating System

รูปที่ 2-5 แสดงสถาปัตยกรรม J2ME และสถาปัตยกรรม MIDP

2.3.2 แพลตฟอร์มของจาวาโทโมบายส์อีดีชัน

จะแบ่งเป็น 4 ส่วนดังนี้

- 1) Profile เป็นกลุ่มคลาสไลบรารีที่รองรับข้อแตกต่างของอุปกรณ์ เช่น Profile ของอุปกรณ์ประเภทมือถือคือ MIDP (Mobile Information Device Profile) จะเป็นกลุ่มคลาสไลบรารีต่างๆที่รองรับ

การพัฒนาแอปพลิเคชันบนอุปกรณ์ประเภทมือถือ ประกอบด้วย UI (User Interface), Input/Output, Event Handling, Persistent Storage, Networking และ Timers

2) **Configuration** เป็นกลุ่มคลาสไลบรารีพื้นฐานสำหรับอุปกรณ์ประเภทมือถือทุกประเภท การที่จะให้กลุ่มคลาสนี้เคลือบคลุมทุกอุปกรณ์นั้น จำเป็นต้องนำอุปกรณ์มาจัดกลุ่มโดยใช้คุณสมบัติพื้นฐานของอุปกรณ์ คือ หน่วยความจำ การแสดงผล การเชื่อมต่อสัญญาณ และการใช้พลังงาน ปัจจุบัน Configuration ใน J2ME แบ่งเป็น 2 ประเภท คือ Connect Device Configuration (CDC) เป็นอุปกรณ์ใช้ร่วมกับอุปกรณ์อื่น ติดตั้งตายตัวและใช้เชื่อมต่อข้อมูล โดยปรกติก็มีส่วนติดต่อผู้ใช้หลากหลายแบบ มีหน่วยความจำประมาณ 2 – 16 เมกกะไบต์ ตัวอย่างอุปกรณ์ประเภทนี้ได้แก่ โทรศัพท์มือถือขนาดเล็ก โทรศัพท์พร้อมจอภาพที่เชื่อมต่อกับอินเทอร์เน็ต อุปกรณ์สื่อสารที่มีความซับซ้อนสูง และ Connected Limited Device Configuration (CLDC) ซึ่งเป็นกลุ่มคลาสอุปกรณ์ส่วนบุคคล พกพาได้ และใช้เชื่อมต่อข้อมูล โดยปรกติจะมีส่วนติดต่อผู้ใช้ได้ง่าย มีหน่วยความจำประมาณ 128 กิโลไบต์ – 1 เมกกะไบต์ ตัวอย่างอุปกรณ์ประเภทนี้ได้แก่ โทรศัพท์มือถือแบบไม่ซับซ้อนมากนัก เพจเจอร์รับส่งข้อความ เครื่องคอมพิวเตอร์แบบพกพา

3) **Java Virtual Machine** โปรแกรมภาษาทุกตัวจะทำงานภายใต้ JVM เสมอ โดยเมื่อเราคอมไพล์โปรแกรมจาวาเป็นคลาสแล้ว JVM จะทำหน้าที่แปลงคลาสนั้นไปเป็นภาษาเครื่องและทำงานต่อไป ใน J2ME จะใช้ Configuration เป็นตัวกำหนด JVM โดยกลุ่ม CDC ใช้กลุ่มคลาสชุดเดียวกับ J2SE ทำให้สามารถใช้ JVM ได้แต่สำหรับกลุ่ม CLDC ไม่ได้ใช้กลุ่มคลาสของ J2SE ทั้งหมดทำให้ต้องพัฒนา JVM ขึ้นมาใหม่เรียกว่า K Virtual Machine (KVM) ซึ่งมีคุณสมบัติเป็น JVM ที่ใช้งานในอุปกรณ์ประเภทมือถือ

4) **Host Operating System** คือ ระบบปฏิบัติการของอุปกรณ์แต่ละตัว

2.3.3 MIDP

MIDP คือ Profile สำหรับพัฒนาแอปพลิเคชันบนอุปกรณ์ประเภทมือถือ ซึ่งในที่นี้เราจะใช้เวอร์ชัน 1.0 และใช้ KVM เป็น Virtual Machine แทน JVM-MIDP ประกอบด้วยแพ็คเกจทั้งหมด 7 แพ็คเกจ ดังนี้

แพ็คเกจ	คำอธิบาย
java.io	กลุ่มคลาสที่มาจาก J2SE สำหรับส่งข้อมูล (Input/Output)
java.lang	กลุ่มคลาสที่มาจาก J2SE ที่เกี่ยวกับจาวา
java.util	กลุ่มคลาสที่มาจาก J2SE ประกอบด้วยคลาสยูทิลิตี้ต่างๆ
javax.microedition.io	กลุ่มคลาสสำหรับรับส่งข้อมูลผ่านระบบเน็ตเวิร์ก
javax.microedition.lcdui	กลุ่มคลาสของคอม โพนেন্টต่างๆ
javax.microedition.midlet	กลุ่มคลาสสำหรับแอปพลิเคชันของ J2SE
javax.microedition.rms	กลุ่มคลาสสำหรับจัดการข้อมูลบนหน่วยความจำถาวรของอุปกรณ์

ตารางที่ 2-1 แสดง packet ของ JMP MIDP

บทที่ 3

เครื่องมือที่ใช้ในการพัฒนาระบบ

3.1 Apache Tomcat Webserver 5.0.16

เป็นเว็บเซิร์ฟเวอร์ที่รองรับการทำงานของ JSP คือ สามารถติดต่อกับจาวาคอมไพเลอร์เพื่อทำการแปลงไฟล์ซึ่งก็คือสคริปต์ JSP ให้กลายเป็น HTML เพื่อส่งให้กับผู้ร้องขอเว็บเพจนั้นๆ และยังแปลงเป็นไฟล์ class เพื่อเก็บไว้ประมวลผลในลำดับต่อไป

ซึ่งเวอร์ชันนี้เป็นตัวใหม่ล่าสุดที่สามารถระบุถึงปัญหาของการเขียนสคริปต์ไม่ถูกต้องได้ดีกว่าเวอร์ชัน 4 มาก

3.2 MySQL 4.016 และ MySQL Front 2.5

MySQL 4.016 เป็น Database Server ที่สามารถใช้งานร่วมกับภาษาจาวาได้เป็นอย่างดี รวมทั้งสคริปต์ JSP ด้วย สามารถใช้งานได้หลายแพลตฟอร์มด้วยกันไม่ว่าจะเป็น Linux หรือ Windowa และยังใช้งานได้ง่าย และมีขนาดเล็กไม่มีความยุ่งยากซับซ้อนในการใช้งาน เหมาะกับขนาดของระบบที่กำลังพัฒนาซึ่งมีขนาดค่อนข้างเล็กนี้ แต่ MySQL มีรูปแบบการใช้งานในลักษณะของ text mode

MySQL Front 2.5 เป็นตัวช่วยเหลือเพื่อให้งาน MySQL เป็นไปได้อย่างขึ้น คือ มีการแปลงจากอินเตอร์เฟซ (interface) ที่เป็น text mode ให้อยู่ในรูปแบบของกราฟิกได้เป็นอย่างดี

3.3 J2SE และ J2ME

เป็นคอมไพเลอร์และ interpreter ให้กับภาษาจาวาต่างๆ (ในที่นี้คือ Java Applet) รวมทั้ง JSP ด้วย ทั้งนี้ J2SE จะถูกใช้งานในฝั่งของเซิร์ฟเวอร์ที่รองรับการทำงานของไคลเอนต์ที่เป็นเครื่องคอมพิวเตอร์ และจะใช้ J2ME ในกรณีที่ไคลเอนต์ที่ติดต่อเข้ามาเป็นโทรศัพท์เคลื่อนที่

3.4 J2ME Wireless Toolkit (WTK)

เป็นชุดเครื่องมือในการพัฒนาแอปพลิเคชันบนโทรศัพท์เคลื่อนที่โดยเฉพาะ เวอร์ชันที่ใช้คือ 1.0.4 โดยมีรูปแบบการพัฒนาเป็น โปรเจกต์ ซึ่งภายในหนึ่ง โปรเจกต์จะประกอบด้วยขั้นตอนการสร้างโปรเจกต์ การคอมไพล์ การทดสอบการทำงานและการทำแพ็คเกจ

สิ่งที่ WTK ไม่มีคือเอดิเตอร์เพื่อเขียนโค้ดจึงต้องไปทำในเอดิเตอร์ตัวอื่น แล้วนำไปใส่ไว้ในโฟลเดอร์ของโปรเจกต์ อีกครั้งหนึ่ง

3.5 Editor

Macromedia Dreamweaver MX 2004

ใช้ในการพัฒนาส่วนที่เป็นไฟล์ JSP ซึ่งรองรับไฟล์ที่เกี่ยวกับเว็บทุกชนิด ช่วยสร้างอินเทอร์เน็ตเฟสของคำสั่งและสคริปต์ต่างๆให้ใช้งานได้ รวมทั้งยังมีเครื่องมือที่ใช้ในการทดสอบผลของการเขียนสคริปต์หรือโค้คนั้นๆอีกด้วย

Editplus

ใช้ในขั้นตอนการเขียนภาษาจาวาต่างๆ ทั้งที่เป็นแอปพลิเคชันบนโทรศัพท์เคลื่อนที่และที่เป็น Applet เป็น editor ที่มีขนาดเล็ก และรองรับการเขียนโค้ดจาวา คือ สามารถให้สีของ syntax ต่างๆได้ รวมทั้งยังช่วยในการจับคู่วงเล็บและเพิ่มชุดคำสั่งบางคำสั่งให้อย่างอัตโนมัติขณะเขียน โค้ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบและพัฒนาระบบ
(ส่วนการแสดงผล)

4.1 การออกแบบตารางฐานข้อมูล

ฐานข้อมูลที่ใช้ในระบบของส่วนการแสดงผลนี้ใช้ MySQL เป็นฐานข้อมูล ซึ่งประกอบไปด้วยตารางทั้งหมด 7 ตารางคือ

4.1.1 ตาราง Member

เป็นตารางที่เก็บรายละเอียดของสมาชิกของระบบ โดยมีฟิลด์ต่างๆ ดังนี้คือ

ฟิลด์	รายละเอียด
id	หมายเลขลำดับสมาชิก
usr	เป็นชื่อที่ใช้ในการล็อกอิน (login) เข้าสู่ระบบสมาชิก
pass	รหัสที่ใช้การเข้ารับบริการของระบบ
name	ชื่อและนามสกุลของสมาชิก
email	อีเมลที่ใช้ในการติดต่อ
district	เขตพื้นที่ของกรุงเทพมหานครฯ เช่น ลาดกระบัง, พระโขนง
address	ที่อยู่ของสมาชิก
mobile	หมายเลขโทรศัพท์เคลื่อนที่ของสมาชิก
Provider	ผู้ให้บริการโทรศัพท์เคลื่อนที่
Mo_usr	ชื่อที่ใช้ในการล็อกอินของโทรศัพท์เพื่อส่ง SMS
Mo_pass	รหัสผ่านที่ใช้เพื่อส่ง SMS
Start1	จุดเริ่มต้นที่ 1 ที่สมาชิกต้องการให้ระบบค้นหาเส้นทางให้
dest1	จุดสิ้นสุดที่ 1 ที่สมาชิกต้องการให้ระบบค้นหาเส้นทางให้
Start2	จุดเริ่มต้นที่ 2 ที่สมาชิกต้องการให้ระบบค้นหาเส้นทางให้
Dest2	จุดสิ้นสุดที่ 2 ที่สมาชิกต้องการให้ระบบค้นหาเส้นทางให้

ตารางที่ 4-1 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Member

4.1.2 ตาราง Matching

เป็นตารางที่ใช้ทำการเปรียบเทียบเพื่อแปลงจากชื่อบริเวณต่างๆ ซึ่งเป็นได้ทั้งชื่อถนน, สถานที่ และชื่อแยก ให้กลายเป็นชื่อแยกทั้งหมดซึ่งก็คือจุดที่สามารถเกิดการเกี่ยวเปลี่ยนแปลงเส้นทางการเดินทางได้ เช่น บริเวณยูเทิร์น (U-turn) ของถนน และสี่แยก

เนื่องจากว่าในการทำงานของส่วนที่เป็นระบบค้นหาเส้นทางอัจฉริยะส่วนประมวลผลนั้น จะเป็นการคำนวณจาก โหนด (node) ดังกล่าวทั้งสิ้น ดังนั้นเพื่อความยืดหยุ่นในการใช้งานของผู้ขอบริการ ให้มีความหลากหลายจึงต้องทำการแปลงก่อน แล้วยังส่งต่อให้ส่วนประมวลผลทำงานต่อไป

ฟิลด์	รายละเอียด
word	ชื่อบริเวณซึ่งเป็นได้ทั้งชื่อถนน, สถานที่และชื่อแยก
inter1	เป็นชื่อที่ใช้ในการล็อกอิน (login) เข้าสู่ระบบสมาชิก
inter2	รหัสที่ใช้การเข้ารับบริการของระบบ

ตารางที่ 4-2 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Matching

4.1.3 ตาราง Report

เป็นตารางที่ใช้ในการรายงานอุบัติเหตุหรือเกิดสิ่งกีดขวางการจราจรที่เกิดขึ้น โดยการค้นหาจะ เริ่มจากชื่อถนนหรือชื่อเขตของกรุงเทพมหานคร

ฟิลด์	รายละเอียด
street	ชื่อถนนที่เกิดอุบัติเหตุหรือเกิดสิ่งกีดขวางการจราจร
district	ชื่อเขตที่เกิดอุบัติเหตุหรือเกิดสิ่งกีดขวางการจราจร
comment	รายงานอุบัติเหตุ หรือเกิดสิ่งกีดขวางการจราจร
date	วันที่ที่เกิดรายงานนั้นๆ

ตารางที่ 4-3 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Report

4.1.4 ตาราง District

เก็บรายชื่อของเขตต่างๆภายในกรุงเทพมหานคร เพื่อการเปลี่ยนแปลงในภายหลัง เช่น การเพิ่มหรือลบเขต จะสามารถทำได้ง่ายขึ้นโดยกระทำที่จุดเดียว

ฟิลด์	รายละเอียด
id	หมายเลขลำดับของเขตภายในกรุงเทพมหานคร โดยเรียงตามตัวอักษร
distr_name	รายชื่อเขตของกรุงเทพมหานคร

ตารางที่ 4-4 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Report

4.1.5 ตาราง Board_quest

ใช้เก็บรายละเอียดของเว็บบอร์ด ซึ่งตารางนี้จะเก็บเกี่ยวกับกระทู้หลัก (หัวข้อกระทู้) ทั้งหมดที่มีของเว็บบอร์ด แต่จะไม่ได้ทำการเก็บเกี่ยวกับรายละเอียดของการตอบกระทู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟิลด์	รายละเอียด
id	หมายเลขลำดับของหัวข้อกระทู้ โดยเรียงตามเวลาที่สร้างก่อนหลัง
topic	ชื่อหัวข้อกระทู้
name	ชื่อผู้ตั้งกระทู้ (ชื่อที่ใช้ในการล็อกอินของสมาชิก)
email	อีเมลของผู้ตั้งกระทู้ที่ได้ให้ไว้ในฐานข้อมูลสมาชิก
detail	รายละเอียดของหัวข้อกระทู้
ip	หมายเลข ไอพีแอดเดรส (IP address) ของเครื่องที่ทำการสร้างกระทู้
datepost	วันที่และเวลาที่สร้างกระทู้
lastpost	วันที่และเวลาครั้งล่าสุดที่ได้มีการเปลี่ยนแปลงกระทู้ (ตอบหรือสร้าง)
Lastname	ชื่อล็อกอินของผู้ที่ทำการเปลี่ยนแปลงครั้งล่าสุด
View	จำนวนของผู้ที่เข้ามาดูกระทู้
ans	จำนวนผู้ทำการตอบกระทู้

ตารางที่ 4-5 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest

4.1.6 ตาราง Board_ans

ใช้เก็บรายละเอียดของเว็บบอร์ด ซึ่งตารางนี้จะเก็บเกี่ยวกับการโพสต์ตอบกระทู้ โดยเชื่อมโยงกับตาราง Board_quest โดยใช้หมายเลข id_quest ของตารางนี้ให้ตรงกับหมายเลข id ในตาราง Board_quest หากตรงกันหมายถึงคำตอบกระทู้นี้เป็นของกระทู้นั้นๆ

ฟิลด์	รายละเอียด
id_ans	หมายเลขของคำตอบกระทู้
id_quest	หมายเลขของกระทู้ซึ่งตรงกันกับของตาราง Board_quest
name	ชื่อล็อกอินของผู้ตอบกระทู้
email	อีเมลของผู้ตอบกระทู้ที่ได้ให้ไว้ในฐานข้อมูล
ip	หมายเลข ไอพีแอดเดรส (IP address) ของเครื่องที่ทำการตอบกระทู้
datepost	วันที่และเวลาที่ได้ทำการตอบกระทู้

ตารางที่ 4-6 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest

4.1.7 ตาราง Mapinfo

เป็นตารางที่จะถูกเรียกใช้งานโดย Apple เพื่อใช้ในการแสดงแผนที่ และสร้างจุดที่แสดงให้เห็นถึงบริเวณที่มีโหนด (จุดที่สามารถเกิดการเลี้ยวเพื่อเปลี่ยนเส้นทางการเดินทางได้)

โดยที่ apple จะนำค่า x และ y จากฐานข้อมูลมาทำการวาด โหนด ถ้า apple ทำการแสดงผลภาพใดอยู่ก็จะทำการวาด โหนดที่มีฟิลด์ image ตรงกับชื่อรูปนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟิลด์	รายละเอียด
id	หมายเลขของโหนด
node	ชื่อโหนด
x_coord	ค่าตามแกน x ของโหนดเรียงจากซ้ายไปขวาโดยเริ่มที่ 0 มีค่าถึง 600
y_coord	ค่าตามแกน y ของโหนดเรียงจากบนลงล่างโดยเริ่มที่ 0 มีค่าถึง 600
junction	จำนวนแยกเส้นทางที่สามารถไปได้ของโหนด (ยูเทิร์นมีค่าเป็น 0)
image	ชื่อรูปภาพของแผนที่ที่จะทำการเรียกใน applet
district	ชื่อเขตของบริเวณที่โหนดนี้อยู่

ตารางที่ 4-7 แสดงชื่อฟิลด์และสิ่งที่เก็บภายในตาราง Board_quest

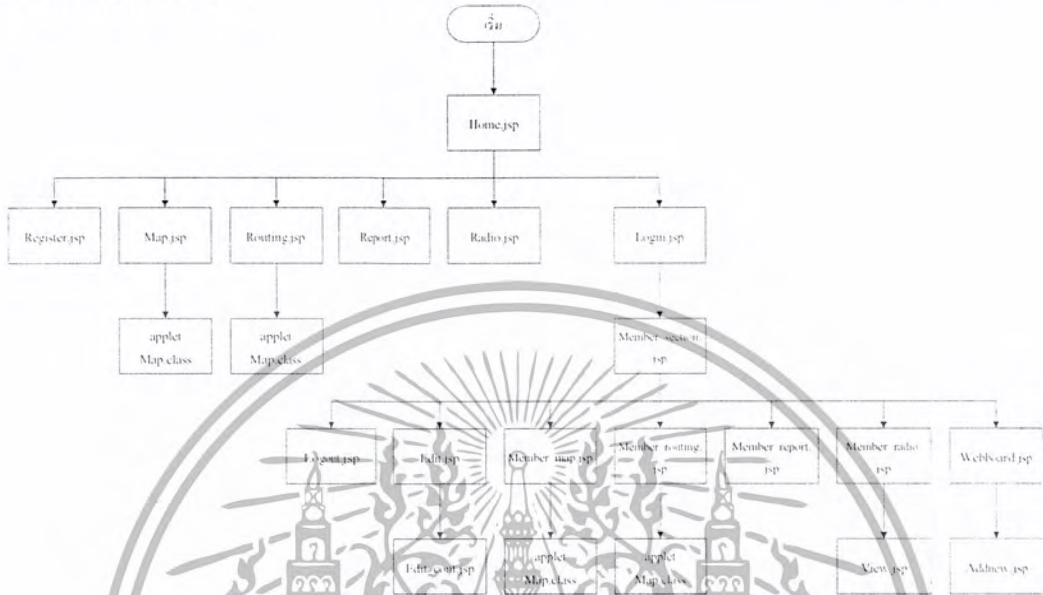
4.2 ขั้นตอนการทำงานส่วนแสดงผลของเว็บไซต์

การทำงานของส่วนเว็บไซต์จะอยู่บนเครื่องคอมพิวเตอร์ โดยใช้สคริปต์ JSP ในการเขียน โดยใช้ Tomcat เป็นเว็บเซิร์ฟเวอร์ และติดต่อกับฐานข้อมูลซึ่งก็คือ MySQL โดยผ่านทาง JDBC โดยไดรเวอร์ (Driver) ที่ชื่อว่า "org.gjt.mm.mysql.Driver" ซึ่งเครื่องที่เป็นเว็บเซิร์ฟเวอร์และเซิร์ฟเวอร์ฐานข้อมูลเป็นเครื่องเดียวกัน ลักษณะการทำงานจะแบ่งเป็นแต่ละเพจดังนี้ คือ

1. หน้าแรกหรือโฮมเพจ (home.jsp)
2. หน้าสมัครสมาชิกใหม่ (register.jsp)
3. หน้าแก้ไขข้อมูลส่วนตัวของสมาชิก (edit.jsp และ edit_cont.jsp)
4. หน้าล็อกอินหรือหน้าเข้าสู่ระบบสมาชิก (login.jsp)
5. หน้าสื่อกเองต่อออกจากระบบ (logout.jsp)
6. หน้าแสดงแผนที่และรายละเอียดของเส้นทางจราจร (imap.jsp และ member_map.jsp)
7. หน้าค้นหาเส้นทางจราจร (routing.jsp และ member_routing.jsp)
8. หน้าแสดงรายงานอุบัติเหตุและสิ่งกีดขวางจราจร (report.jsp และ member_report.jsp)
9. หน้าแสดงรายงานสดวิทยุออนไลน์ จส.100 (radio.jsp และ member_radio.jsp)
10. หน้าต้อนรับเข้าสู่ระบบสมาชิกหลังทำการล็อกอิน (member_section.jsp)
11. หน้าแสดงรายชื่อกระทู้ทั้งหมดของเว็บบอร์ด (webboard.jsp)
12. หน้าสร้างกระทู้ใหม่หรือหน้าโพสต์ (addnew.jsp)
13. หน้าแสดงและตอบกระทู้ (view.jsp)
14. หน้าแสดงความคิดเห็นของเซิร์ฟเวอร์ (error.jsp)
15. ไฟล์เก็บส่วนหัวและแถบเมนูด้านข้าง (header.jsp, member_header.jsp และ member_left.jsp)
16. โปรแกรม Java Applet ที่ใช้แสดงแผนที่ (Map.class)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ขึ้นต้นด้วย “member” ทั้งหมดและหน้าที่เกี่ยวกับเว็บบอร์ดจะมีตรวจสอบก่อนการเข้ามาดูทุกครั้งว่าบุคคลนั้นๆ ได้ผ่านการล็อกอินมาอย่างถูกต้องแล้วหรือยัง และจะมีการเรียกใช้งาน member_left.jsp และ member_header.jsp ซึ่งเป็นส่วนเมนูที่อยู่ทางด้านซ้ายและด้านบนของเว็บเพจตามลำดับ ส่วนหน้าเว็บเพจอื่นๆจะเรียกใช้ header.jsp ทุกๆเพจ ดังนั้นการที่จะแก้หน้าตาของเว็บเพจจึงสามารถกระทำได้ที่เดียว



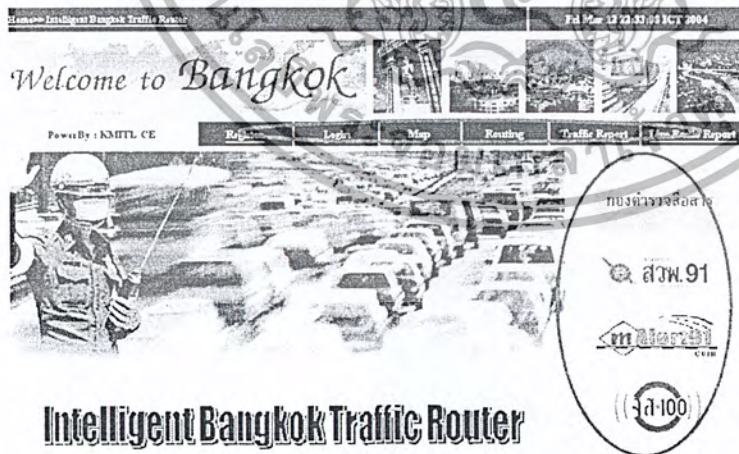
รูปที่ 4-1/แสดงลำดับการใช้งานของเว็บเพจ

และเมื่อเกิดข้อผิดพลาดขึ้น คำรับทุกๆเพจจะเรียก ไปยัง error.jsp

4.2.1 หน้าแรกหรือโฮมเพจ (home.jsp)

มีการแสดงลิงค์ไปยังเพจต่างๆภายในเว็บไซต์ และยังรวบรวมลิงค์ที่เกี่ยวข้องกับกิจการจราจร

ภายในเขตกรุงเทพมหานครด้วย



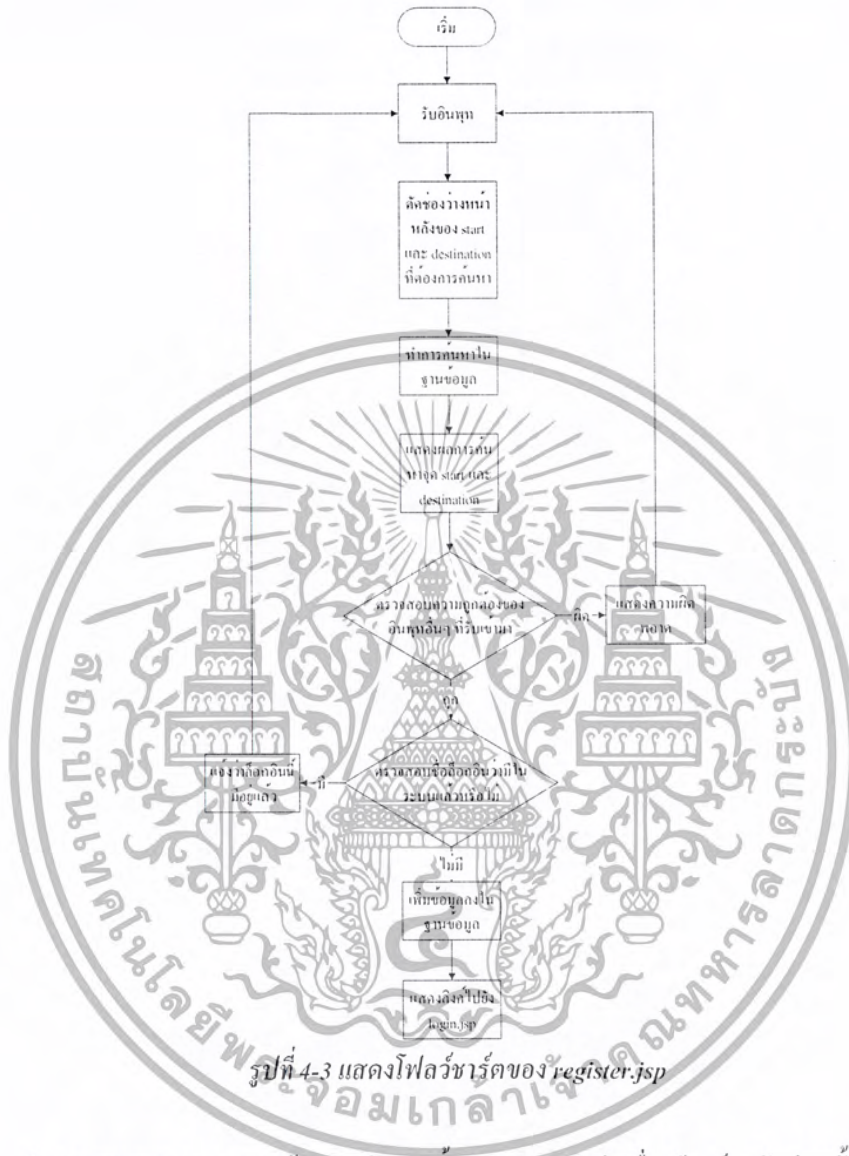
ระบบที่แนะนำทางจราจรในกรุงเทพฯ

รูปที่ 4-2 แสดงหน้าตาของ home.jsp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 หน้าสมัครสมาชิกใหม่ (register.jsp)

เป็นที่ให้สมาชิกทำการกรอกข้อมูลเพื่อนำข้อมูลไปลงในฐานข้อมูลในตารางที่ชื่อ member



รูปที่ 4-3 แสดงโฟลว์ชาร์ตของ register.jsp

ในการตรวจสอบความถูกต้องของอินพุตนั้นจะตรวจสอบว่า ชื่อ, อีเมล, รหัสผ่านทั้ง 2 ครั้ง และ ชื่อถือถืออินต้องอยู่ในรูปแบบที่ถูกต้อง คือ ต้องมีการป้อนและไม่ประกอบด้วยอักขระพิเศษ รวมทั้งตรวจสอบว่าจุด start และ destination ที่เป็นคู่เดียวกันนั้นต้องไม่ใช่จุดเดียวกัน และเมื่อมีการกรอกอินพุตของคู่ใดแล้ว อินพุตคู่อื่นๆจะต้องมีการกรอกด้วย

ซึ่งขั้นตอนนี้จะใช้เทคนิคของตัวแปรเซสชัน (session parameter) คือ จะใช้หมายเลขเซสชันของ http connection เป็นเหมือนกับหมายเลขตัวแทนในการติดต่อกับเซิร์ฟเวอร์ แล้วเก็บข้อมูลในฝั่งเซิร์ฟเวอร์สำหรับผู้ที่ผ่านการล็อกอินอย่างถูกต้องโดยใช้หมายเลขเซสชันเป็นตัวบ่งบอกว่าต้องเก็บที่ตัวแปรใด

ดังนั้นในการตรวจสอบสิทธิของสมาชิกจะใช้ตัวแปรเซสชันที่เก็บในฝั่งเซิร์ฟเวอร์เป็นตัวตรวจสอบ (ผู้ที่ผ่านการล็อกอินจะมีตัวแปร ผู้ที่ไม่ผ่านจะไม่มี) ซึ่งโค้ดส่วนนี้จะเป็นดังนี้

```
if (session.getAttribute("auth") == null)
{
    response.sendRedirect("login.jsp");
    return;
}
```

4.2.5 หน้าล็อกเอาต์จากระบบ (logout.jsp)

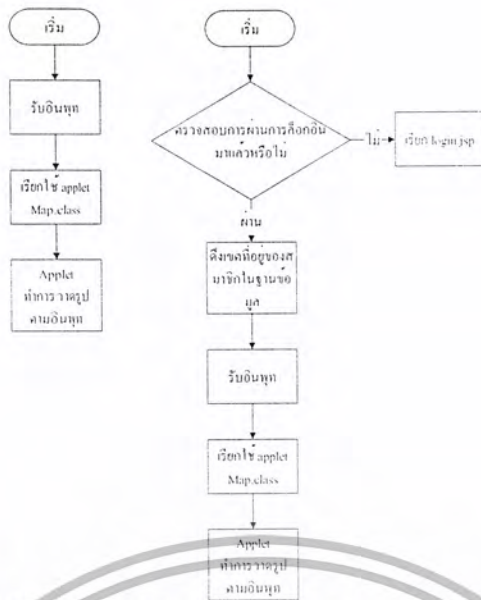
เป็นการลบตัวแปรเซสชันออกจากฝั่งเซิร์ฟเวอร์ทั้งหมด โดยใช้หมายเลขเซสชันเป็นตัวบ่งบอกเช่นเดียวกับกับขั้นตอนล็อกอิน

และสำหรับการที่สมาชิกที่ผ่านการล็อกอินแล้ว แต่ไม่ได้ทำการติดต่อกับระบบเป็นเวลานาน (20 นาที) จะทำให้เซสชันนั้นปิดตัวลงไป จะเสมือนกับว่าสมาชิกได้ทำการล็อกเอาต์ออกไปแล้ว ซึ่งส่วนนี้จะใช้คำสั่ง

```
<%
    session.invalidate();
    response.sendRedirect("home.jsp");
%>
```

4.2.6 หน้าแสดงแผนที่และรายละเอียดของเส้นทางจราจร (map.jsp และ member_map.jsp)

จะแสดงแผนที่พร้อมด้วยสภาพการจราจรให้กับผู้ใช้ดู โดยต้องให้ผู้ใช้ทำการเลือกเขตที่ต้องการจะดูก่อน แต่สำหรับสมาชิกจะแสดงแผนที่ของเขตที่ตรงกับในฟิลด์ district ในตาราง member ในฐานข้อมูลเลย โดยแผนที่นั้นจะใช้ applet เป็นตัววาด (Map.class) และสำหรับ applet เพื่อนำมาใส่ไปวางไว้บนตำแหน่งที่เป็นโหนด ก็จะแสดงรายละเอียดของโหนดนั้นๆออกมา



รูปที่ 4-5 แสดงโฟลว์ชาร์ตของ map.jsp และ member_map.jsp

4.2.7 หน้าค้นหาเส้นทางจราจร (routing.jsp และ member_routing.jsp)

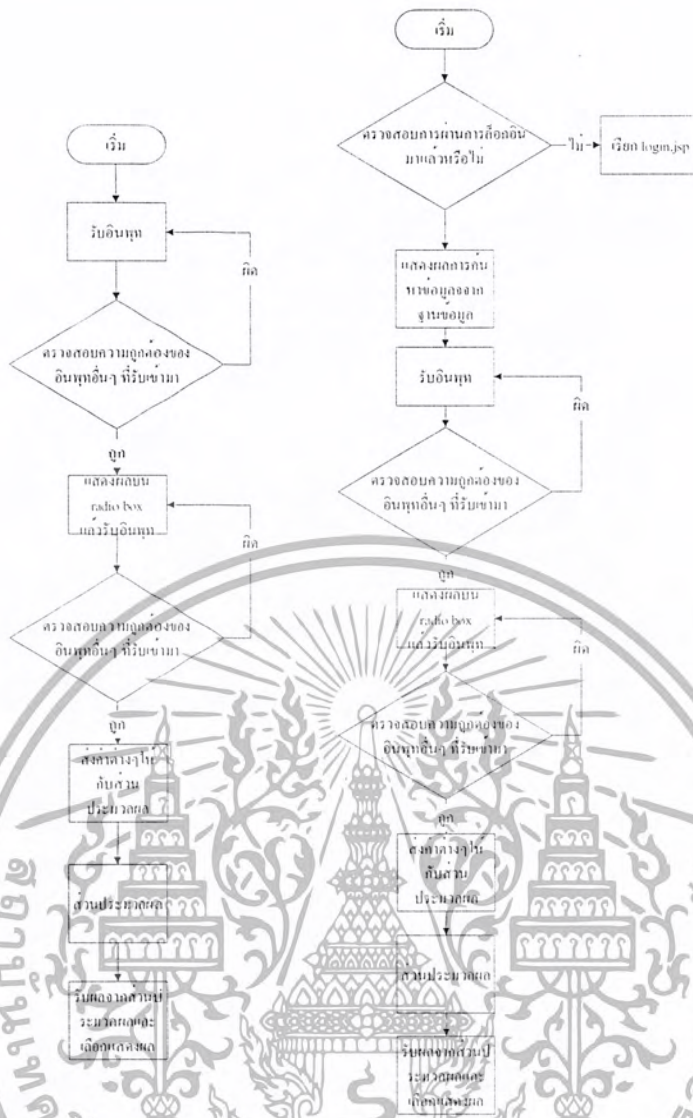
ให้ผู้ใช้ได้ทำการค้นหาบริเวณที่ต้องการก่อน โดยประเภทของบริเวณจะเป็นได้คือ ชื่อถนน, ชื่อโหนดและสถานที่ เช่น ถ้าทำการค้นหาคำว่า "ราช" ผลที่ได้จากการค้นหาก็คือ ราชประสงค์ ราชดำริ ราชดำเนิน เป็นต้น และผลการค้นหาจะแสดงออกมาเป็น radio box ให้ทำการเลือกว่าจะเลือกจากจุดเริ่มต้นใดไปยังจุดสิ้นสุดใด แต่สำหรับสมาชิกจะแสดงจุดเริ่มต้นและสิ้นสุดที่มีอยู่ในฐานข้อมูลเลย โดยไม่ต้องผ่านขั้นตอนการค้นหา

ซึ่งขั้นตอนต่อไปก็จะทำการเปลี่ยนจากจุดหรือบริเวณที่รับเข้ามาเป็น โหนดทั้งหมด เช่น หากเลือกเป็นชื่อถนนจะได้โหนด 2 โหนดด้วยกัน คือ แยกที่ถนนนั้นๆ ต้นสุด โดยที่การเปลี่ยนจากจุดหรือบริเวณเป็น โหนดนั้นจะใช้ข้อมูลในตาราง matching ในฐานข้อมูล (เปลี่ยนจากฟิลด์ word ไปเป็นฟิลด์ inter1 และ inter2) ดังนั้นจะเห็นได้ว่าจากจุดหรือบริเวณ 2 จุด (คือต้นทางและปลายทาง) จะได้เป็นโหนดทั้งหมด 4 โหนด

การจับคู่โหนดนั้นก็จะจับคู่ได้ทั้งหมด 4 แบบด้วยกัน แต่สำหรับส่วนประมวลผลนั้นจะทำงานได้แค่ทีละคู่เท่านั้น ดังนั้นจึงต้องส่งให้แก่ส่วนประมวลผลทีละคู่ และจะได้ผลการค้นหา 4 ผลด้วยกัน จากนั้นจึงตรวจสอบว่าเส้นทางใดมีผลตรงกับความต้องการของผู้ใช้มากที่สุด จึงเลือกผลนั้นมาทำการแสดงผลเพียงผลเดียว

ผู้ใช้สามารถเลือกรูปแบบการค้นหาได้ทั้งสิ้น 3 รูปแบบ คือ ระยะทางที่สั้นที่สุด ค่าใช้จ่ายที่ใช้ในการเดินทางน้อยที่สุด และระยะเวลาในการเดินทางน้อยที่สุด โดยทั้ง 3 แบบนี้สามารถนำมาใช้รวมกันอย่างไรก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 แสดงโฟลว์ชาร์ตของ routing.jsp และ member_routing.jsp

สำหรับขั้นตอนการตรวจสอบอินพุตครั้งแรกนั้นจะดูว่าผู้ใช้ได้พิมพ์ชื่อเข้าหรือยัง ถ้าพิมพ์เข้ามาจะตัดช่องว่างด้านหน้าและหลังของค่าที่รับเข้ามาก่อน แล้วค่อยนำมาค้นหาในฐานข้อมูล สำหรับการตรวจสอบอินพุตส่วนที่ดังนั้นจะดูว่าค่าจาก radio box ที่รับเข้ามา คือจุดเริ่มต้นและสิ้นสุดเป็นจุดเดียวกันหรือไม่ ถ้าเป็นจุดเดียวกันต้องทำการเลือกใหม่

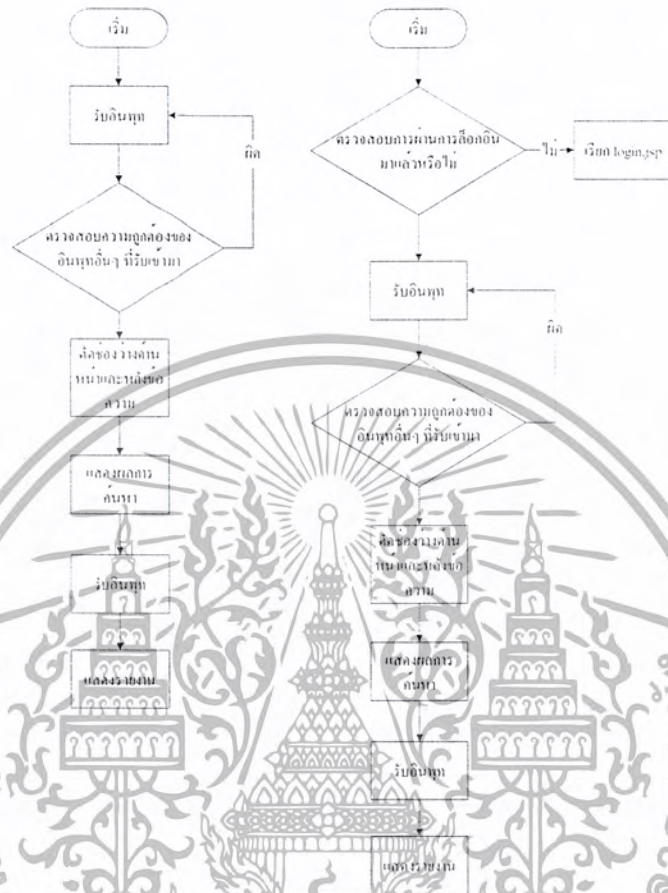
4.2.8 หน้าแสดงรายงานอุบัติเหตุและสิ่งกีดขวางการจราจร (report.jsp และ member_report.jsp)

ให้ผู้ใช้ทำการค้นหาบริเวณที่ต้องการดูรายงานอุบัติเหตุหรือสิ่งกีดขวางการจราจรก่อน โดยบริเวณนั้นต้องเป็นได้ทั้งชื่อเขตและชื่อถนน คือถ้าเลือกค้นหาแบบชื่อเขต เมื่อทำการค้นหาแล้วผลที่จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาจะเป็นรายชื่อถนนที่อยู่ในเขตนั้นๆและมีรายงานเข้ามาทั้งหมด แต่ถ้าเลือกเป็นชื่อถนน ผลที่ได้จากการค้นหาจะเป็นเพียงถนนที่ค้นเจอเท่านั้น

จากนั้นผลของการค้นหาจะแสดงออกมาเป็น check box เพื่อให้เลือกแสดงรายงานเฉพาะบริเวณที่ต้องการเท่านั้น



รูปที่ 4-7 แสดงไวยากรณ์ของ report.jsp และ member_report.jsp

4.2.9 หน้าแสดงรายงานสดวิทยุออนไลน์ จส.100 (radio.jsp และ member_radio.jsp)

เรียกใช้ plugin ของ Windows Media Player ใน Microsoft Internet Explorer มาทำการเล่น URL ของสตรีมที่เป็นรายงานสดของวิทยุ จส.100

โดยมีคำสั่งดังนี้คือ

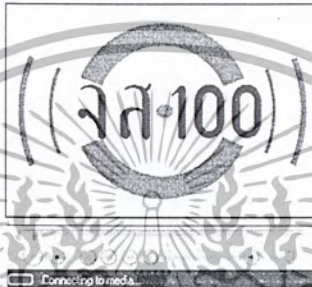
```
<Embed SRC="mms://live.virginradiothailand.com/105.5"
Width="337"
Height="70"
AutoStart="false"
TYPE="application/x-mplayer2"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pluginspage="http://www.microsoft.com/Windows/MediaPlayer/"
Name="objMediaPlayer"
autoSize="false"
showdisplay="false"
displaySize="0"
center="true"
ShowControls="true"
ShowStatusBar="true"
AnimationAtStart="false" border="2"></embed>

```



รูปที่ 4-8 แสดง plug in ของ Windows Media Player

4.2.10 หน้าต้อนรับเข้าสู่ระบบสมาชิกหลังทำการล็อกอิน (member_section.jsp) เป็นหน้าต้อนรับสมาชิกผู้ผ่านการล็อกอินเข้าสู่ระบบอย่างถูกต้องแล้วเท่านั้น

4.2.11 หน้าแสดงรายชื่อกระทู้ทั้งหมดของเว็บบอร์ด (webboard.jsp)

แสดงหัวข้อกระทู้ทั้งหมดที่มี โดยจะแบ่งแสดงเป็นหน้าๆ หน้าละ 10 หัวข้อกระทู้ ดังนั้นจึงต้องคำนวณก่อนว่าจะมีจำนวนหน้าทั้งหมดกี่หน้า

```
int total_page = (int) Math.ceil((double) total_row / (double) row_page);
```

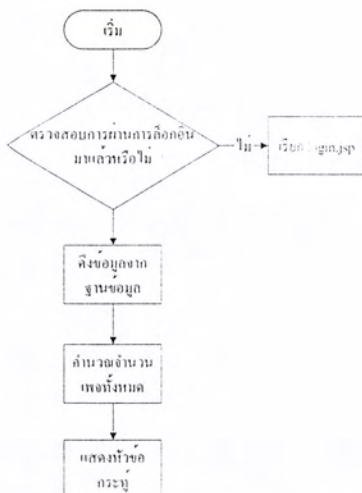
จากนั้นทำการคำนวณต่อวัน จะต้องดึงข้อมูลจากรุ่นข้อมูลเข้ามาตั้งแต่กระทู้ที่เท่าไรถึงเท่าไร โดยการหาจุดเริ่มต้นการ

```
start = (screen - 1) * row_page
```

screen คือ หมายเลขหน้าที่ต้องการเปิดดู จากนั้นก็ทำการดึงข้อมูลออกมาจำนวน 10 กระทู้

ทั้งนี้หมายเลขก่อนหลังของกระทู้จะต้องเรียงตามวันเวลาที่สร้างกระทู้คือ สร้างก่อนจะมี

หมายเลขต่ำกว่า



รูปที่ 4-9 แสดงไพล์ชาร์ตของ webboard.jsp

4.2.12 หน้าสร้างกระทู้ใหม่หรือหน้าโพสท์ (addnew.jsp)

สำหรับสมาชิกเพื่อทำการสร้างกระทู้ใหม่ โดยจะมีการเก็บวันเวลาที่โพสท์ หมายเลข IP address ของเครื่องที่ใช้ติดต่อเข้ามาด้วย ส่วนข้อมูลอื่นจะเก็บจากค่าในฐานข้อมูลของสมาชิกคนนั้นๆ เช่น อีเมลแอดเดรส

โดยที่เพจนี้จะมีการเรียกเมธอด (Method) ชื่อ filter_topic และ filter_comment ด้วย โดยที่ภายในเมธอดนี้จะมีรายละเอียดดังนี้

```

public static String filter_topic(String input)
{
    StringBuffer temp = new StringBuffer(input.length());
    char c;
    int j=1;
    for (int i=0;i<input.length();i++)
    {
        c = input.charAt(i);
        if (c == '<') temp.append("&lt;");
        else if (c == '>') temp.append("&gt;");
        else if (c == "\") temp.append("&#039;");
        else if (c == "") temp.append("&quot;");
        else if (c == '&') temp.append("&amp;");
        else if (c == "\") temp.append("&#92;");
        else temp.append(c);
    }
}
  
```

```

        if (j == 30)
        {
            temp.append("<br>");
            j = 0;
        }
        j++;
    }
    return(temp.toString());
}

```

```

public static String filter_comment(String input)
{
    StringBuffer temp = new StringBuffer(input.length());
    char c;
    int j=1;
    for (int i=0;i<input.length();i++)
    {
        c = input.charAt(i);
        if (c == '<') temp.append("&lt;");
        else if (c == '>') temp.append("&gt;");
        else if (c == '"') temp.append("&#039;");
        else if (c == "'") temp.append("&quot;");
        else if (c == '&') temp.append("&amp;");
        else if (c == '\\') temp.append("&#92;");
        else temp.append(c);
    }

    if (j == 50)
    {
        temp.append("<br>");
        j = 0;
    }

    if (c == '\n')
        j = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        j++;
    }
    return(temp.toString());
}

```

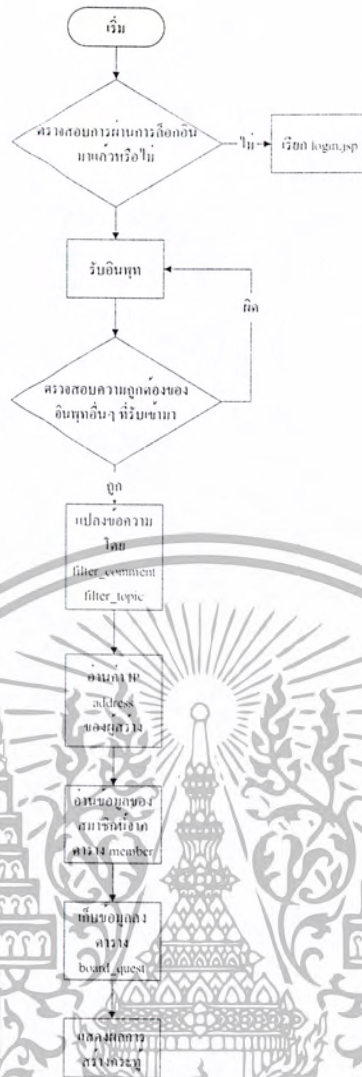
คือมีหน้าที่ในการแปลงข้อความที่รับมาจากสมาชิก แล้วทำการเปลี่ยนจากอักขระพิเศษที่จะทำให้เกิดปัญหาหรือเป็นอันตรายแก่เซิร์ฟเวอร์ได้เป็นอักขระอื่นๆดังนี้คือ

<	เป็น	<
>	เป็น	>
\	เป็น	'
"	เป็น	"
&	เป็น	&
\\	เป็น	\

และจะมีการตัดข้อความยาวของตัวอักษร คือ เมื่อข้อความมีความยาวเกิน 50 ตัวอักษรและ 30 ตัวอักษร สำหรับ filter_comment และ filter_topic ตามลำดับ โดยไม่มีการเว้นวรรคหรือขึ้นบรรทัดใหม่ จะมีการแทรก “
” ลงไปในข้อความนั้นๆด้วย เพื่อให้รูปแบบตอนแสดงผลของตัวอักษรมีรูปแบบที่ไม่เป็นปัญหา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-10 แสดงโฟลว์ชาร์ตของ addnew.jsp

ในการตรวจสอบอินพุตนั้นจะแค่ดูว่าสมาชิกได้มีการพิมพ์ชื่อกระทู้และรายละเอียดลงมาหรือไม่

4.2.13 หน้าแสดงและตอบกระทู้ (View.jsp)

เมื่อสมาชิกทำการคลิกเมาส์ที่ชื่อกระทู้ภายในหน้า webboard.jsp ก็จะทำให้การเปิดเข้ามาดูว่าภายในกระทู้มีอะไรอยู่บ้าง และสามารถทำการตอบกระทู้นั้นๆ ได้

เพงนี้จะทำการเรียกเมธอด filter_comment เช่นเดียวกับกับ addnew.jsp และเรียกเมธอด newline เพิ่มขึ้นมา ซึ่งมีการทำงานดังนี้

```

public static String newline(String input)
{
    StringBuffer temp = new StringBuffer();
    StringTokenizer st = new StringTokenizer(input, "\n");
    while(st.hasMoreTokens())
        temp.append(st.nextToken() + "<br>");
    return(temp.toString());
}

```

คือเนื่องจากการรับข้อความเข้ามาทาง textfield ของเพจ HTML เมื่อเกิดการเว้นบรรทัดขึ้นจะแทนด้วยอักขระ "\n" ซึ่งการที่เราจะนำข้อความนี้ไปแสดงบนเพจ HTML จะต้องทำการแปลงให้เป็น
 ก่อน ซึ่งเป็นคำสั่งเว้นบรรทัดจริงๆ ดังนั้นจึงต้องเรียกเมธอดนี้ก่อนการแสดงผล



รูปที่ 4-11 แสดงโฟลว์ชาร์ตของ view.jsp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.14 หน้าแสดงความคิดพลาดของเซิร์ฟเวอร์ (error.jsp)

หน้านี้จะถูกแสดงขึ้นมาเมื่อเกิดความคิดพลาดจิงเว็บเซิร์ฟเวอร์ขึ้น โดยเว็บเซิร์ฟจะเรียกขึ้นมาเองเลย เพื่อเป็นการแจ้งข้อผิดพลาด และปกปิดค่าตัวแปร (parameter) ต่างๆที่ไม่ควรจะให้ผู้ใช้งานเห็นด้วยคำสั่ง

```
<%@ errorPage="error.jsp">
```

4.2.15 ไฟล์เก็บส่วนหัวและแถบเมนูด้านข้าง (header.jsp, member_header.jsp และ

member_left.jsp)

เพื่อการแก้ถึงที่อาจเกิดขึ้นภายหลังด้วยความง่ายและสะดวก จึงได้ทำการรวมลิงค์ที่ต้องมีในแต่ละเพจที่เหมือนกันไว้ในไฟล์เดียว ด้วยคำสั่ง

```
<jsp:include page="header.jsp"/>
```

```
<jsp:include page="member_header.jsp"/>
```

```
<jsp:include page="member_left.jsp"/>
```

โดยที่ไฟล์ header.jsp จะถูกเรียกใช้โดยเพจที่ผู้ใช้ไม่ได้ผ่านการล็อกอินเข้าสู่ระบบ ส่วน member_header.jsp และ member_left.jsp จะถูกเรียกใช้ในเพจที่ผู้ใช้เป็นสมาชิกและผ่านการล็อกอินแล้ว



รูปที่ 4-12 หน้าตาของ member_header.jsp



รูปที่ 4-13 หน้าตาของ header.jsp



รูปที่ 4-14 หน้าตาของ member_left.jsp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.16 โปรแกรม Java Applet ที่ใช้แสดงแผนที่ (Map.class)

ใช้ในการแสดงและคอยกำกับการแสดงของส่วนที่เป็นรูปแผนที่ โดยเฉพาะ คือ จะทำการเลือกรูปภาพไปไว้ 8 ทิศทาง เลือกรูปภาพที่จะนำมาแสดง ทำการวาดรูปหรือเพิ่มจุดที่เป็น โหนดลงในแผนที่ และจากนั้นก็ทำการแสดงรายละเอียดของบริเวณที่ผู้ใช้งานต้องการ

Applet นี้ถูกแบ่งแยกการทำงานออกเป็นหลายๆส่วนซึ่งทำงานไม่เหมือนกัน แต่เกี่ยวข้องกัน โดยแบ่งเป็นเมธอดที่ ซึ่งมีเมธอดที่สำคัญดังนี้คือ

- CountXY ใช้ในการนับจำนวนของโหนดที่มีอยู่ในภาพที่กำลังแสดงอยู่ เพื่อให้โปรแกรมสามารถทำงานได้ถูกต้อง โดยจะเป็นตัวกำหนดจำนวนการวนลูป (For loop) เพื่อการวาดโหนดในรูปนั้นๆ
- getX ใช้ในการดึงค่าคู่ลำดับแกน X ของโหนดแต่ละโหนดในภาพจากฐานข้อมูลที่กำลังแสดงอยู่ เพื่อใช้ในการวาดจุดโหนด
- getY ใช้ในการดึงค่าคู่ลำดับแกน Y ของโหนดแต่ละโหนดในภาพจากฐานข้อมูลที่กำลังแสดงอยู่ เพื่อใช้ในการวาดจุดโหนด
- shownode ใช้ในการแสดงรายละเอียดของโหนดนั้นๆ เมื่อนำมาใส่ไปวางไว้ยังตำแหน่งที่เป็นโหนด
- getPic ใช้ในการคัดลอกส่วนที่เป็น JSP คือ เมื่อผู้ใช้งานทำการเลือกเขตที่จะแสดงรูปแผนที่ เมธอดนี้ จะทำการหาจากฐานข้อมูลว่า จะต้องแสดงรูปใด
- กลุ่มเมธอดที่ใช้แสดงสถานะเมื่อทำการเลื่อนเมาส์
- และกลุ่มเมธอดที่ใช้ในการเปลี่ยนภาพที่แสดงไปในทิศทางต่างๆ

และสำหรับชื่อไฟล์ที่เป็นรูปภาพแผนที่จะต้องทำการเก็บนำลักษณะคู่อันดับ XY เช่น "e10044600n134600s4.jpg" โดยที่ตัวเลขจุดแรกที่อยู่ทางซ้ายมือ นั่นคือค่าของแกน X และถัดมาทางขวามือค่าแกน Y โดยที่เมื่อทำการเปลี่ยนภาพที่จะแสดงนั้นคนเหล่านี้จะบวกหรือลบด้วย 2400 (เมื่อต้องการภาพที่อยู่ด้านบนจะบวกแกน Y ด้วย 2400 เช่นกันทางด้านล่างจะลบด้วย 2400 และสำหรับเมื่อต้องการจะเลื่อนไปทางด้านซ้ายจะลบค่าแกน X ด้วย 2400 เช่นกันทางด้านขวาจะบวกด้วย 2400)

แต่คู่อันดับของโหนดแต่ละโหนดนั้นจะต่างกัน คือ แต่ละภาพจะไม่ขึ้นต่อกัน เช่น ภาพ ก. จะมีคู่อันดับ (x,y) ตั้งแต่ (0,0) ถึง (600,600) และภาพ ข. ก็จะมีตั้งแต่ (0,0) ถึง (600,600) เหมือนกัน

(0,0)



(600,600)

รูปที่ 4-15 รูปแบบคู่อันดับ XY ของโหนด

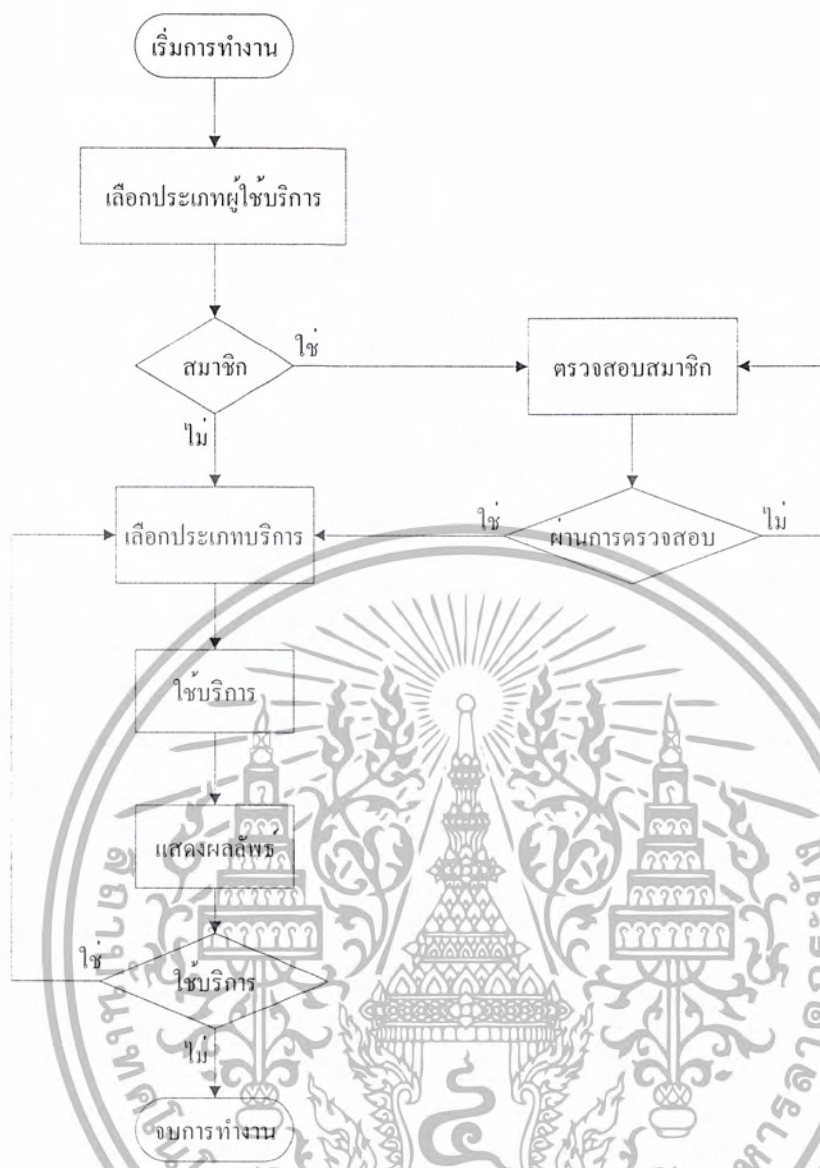
4.3 การทำงานของแอปพลิเคชันค้นหาเส้นทางบนโทรศัพท์เคลื่อนที่

การทำงานของแอปพลิเคชันจะกล่าวถึง ในลักษณะของกิจกรรมที่ผู้ใช้งานกระทำกับตัวระบบ โดยมีเซิร์ฟเวอร์และฐานข้อมูลเป็นส่วนประกอบในการทำงานด้วย ซึ่งสามารถแยกได้ 4 ประเภท ได้แก่ การทำงานของแอปพลิเคชันโดยรวม, การทำงานของบริการค้นหาเส้นทางจราจร, การทำงานของบริการรายงานสภาพการจราจร และการทำงานของบริการแก้ไขข้อมูลสมาชิก

4.3.1 การทำงานของแอปพลิเคชันโดยรวม

ขั้นตอนการทำงานของแอปพลิเคชัน โดยรวมจะเป็นไปตามโฟลว์ชาร์ตของการทำงาน ดังรูปที่ 4-16 ซึ่งมีขั้นตอนดังนี้

- เมื่อผู้ใช้บริการรันแอปพลิเคชัน ขั้นตอนแรกจะต้องเลือก ประเภทของผู้ใช้งาน ได้แก่ สมาชิก หรือ ผู้ใช้งานทั่วไป เพราะสมาชิกจะแตกต่างกับผู้ใช้งานทั่วไปตรงที่ มีข้อมูลต้นทาง, ปลายทาง และข้อมูลเขตรายงานที่เป็นประโยชน์แก่สมาชิกในบริการค้นหาเส้นทาง, บริการรายงานสภาพการจราจร และบริการแก้ไขข้อมูลสมาชิก
- ถ้าผู้ใช้บริการเป็นสมาชิก จะมีการพิสูจน์การมีตัวตนของสมาชิก โดยให้สมาชิกระบุชื่อสมาชิก และรหัสผ่าน ในขั้นตอนนี้จะมีการติดต่อและส่งข้อมูลร้องขอไปยังเซิร์ฟเวอร์ เพื่อตรวจสอบ และเซิร์ฟเวอร์จะส่งผลลัพธ์กลับมายังแอปพลิเคชัน
- ถ้าผลลัพธ์ที่ได้ยืนยันว่าเป็นสมาชิกจริง ผู้ใช้จะเข้าสู่การให้บริการในฐานะสมาชิก ถ้าผลลัพธ์ที่ได้ระบุว่าไม่มีชื่อสมาชิกหรือรหัสผ่านที่ตรงกับฐานข้อมูล โปรแกรมจะให้ผู้ใช้ระบุชื่อสมาชิก และรหัสผ่านอีกครั้ง
- ก่อนที่สมาชิกจะเข้าสู่การให้บริการ โปรแกรมจะทำการติดต่อกับเซิร์ฟเวอร์อีกครั้งเพื่อร้องขอข้อมูลทั้งหมดของสมาชิก และนำมาเก็บไว้ในออเรย์สตรึง
- ทั้งผู้ใช้งานทั่วไปและสมาชิกที่ผ่านการตรวจสอบแล้วจะเข้ามาสู่การรับบริการ ที่มีประเภทของบริการให้เลือก โดยสำหรับผู้ใช้งานทั่วไปจะมีบริการค้นหาเส้นทางและบริการรายงานสภาพการจราจร และสำหรับสมาชิกจะเพิ่มบริการแก้ไขข้อมูลสมาชิกอีก 1 บริการ
- เมื่อผู้ใช้งานเลือกบริการแล้ว จะใช้บริการต่างๆ เพื่อให้ได้ผลลัพธ์ของบริการ โดยสำหรับสมาชิกจะมีข้อมูลต้นทาง ปลายทางและ เขตรายงาน ที่เป็นประโยชน์กับการใช้บริการ
- หลังจากที่ได้รับผลลัพธ์ของบริการแล้ว ผู้ใช้สามารถที่จะเข้าใช้บริการได้อีก หรือจะจบการทำงานแอปพลิเคชันก็ได้



รูปที่ 4-16 แสดงโฟลว์ชาร์ต การทำงานของระบบโดยรวม

4.3.2 การทำงานของบริการค้นหาเส้นทางจราจร

ขั้นตอนการทำงานของบริการค้นหาเส้นทาง จะรวมการทำงานของผู้ใช้ที่เป็นสมาชิกกับผู้ใช้ทั่วไปเข้าด้วยกัน ซึ่งสมาชิกจะมีต้นทางกับปลายทางให้อยู่แล้ว โดยการทำงานจะเป็นไปตามโฟลว์ชาร์ตดังรูปที่ 4-17 ซึ่งมีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทันทีซึ่งโปรแกรมจะดึงข้อมูลต้นทางจากอาเรย์มาแสดงผลเป็นลำดับรายการ หรือจะค้นหา รายการต้นทางใหม่ก็ได้

- การค้นหาคือการติดต่อและส่งข้อมูลไปยังเซิร์ฟเวอร์ ถ้าการค้นหาสำเร็จเซิร์ฟเวอร์จะส่ง ข้อมูลที่เป็นลำดับรายการต้นทางให้ผู้ใช้เลือก
- ถ้าหากไม่พบรายการ โปรแกรมจะให้ผู้ใช้ระบุคีย์เวิร์ดและค้นหารายการต้นทางใหม่
- ผู้ใช้งานสามารถเลือกที่จะค้นหารายการได้มากกว่า 1 ครั้ง
- เมื่อผู้ใช้เลือกต้นทางได้แล้ว โปรแกรมจะนำมาสู่ขั้นตอนการเลือกปลายทาง โดยมีขั้นตอนการ ติดต่อเหมือนกับเลือก และค้นหาปลายทาง เช่นกัน และสำหรับสมาชิก ชื่อปลายทางจะมีให้ อยู่แล้วสามารถเลือกได้ทันที
- ขั้นตอนต่อมาคือการเลือกปัจจัยในการค้นหาเส้นทาง ซึ่งมีอยู่ 3 ประเภท คือ ผู้ใช้ต้องการเส้นทาง ที่มีค่าใช้จ่าย เช่น เสียค่าทางด่วน, ค่าน้ำมัน น้อยที่สุด เป็นต้น ผู้ใช้ต้องการเส้นทางที่ใช้เวลาใน การเดินทางน้อยที่สุด หรือ ผู้ใช้จะต้องการเส้นทางที่มีระยะทางสั้นที่สุด โดยสามารถเลือกได้ มากกว่า 1 ปัจจัย
- หลังจากนั้นแอปพลิเคชันจะทำการส่งข้อมูล ต้นทางและปลายทาง ไปคำนวณที่ฝั่งเซิร์ฟเวอร์
- เซิร์ฟเวอร์คำนวณหาเส้นทางที่ดีที่สุดแล้วจึงส่งผลลัพธ์กลับมาให้ฝั่งไคลเอนต์
- หลังจากที่ทราบผลลัพธ์แล้ว ผู้ใช้สามารถที่จะใช้บริการ ได้อีก หรือจะจบการทำงาน แอปพลิเคชันก็ได้

4.3.3 การทำงานของบริการรายงานสภาพการจราจร

ขั้นตอนการทำงานของบริการรายงานสภาพการจราจร จะรวมการทำงานของผู้ใช้ที่เป็นสมาชิกกับ ผู้ใช้ทั่วไปเข้าด้วยกัน ซึ่งสมาชิกจะมีเขตรายงานให้อยู่แล้ว โดยการทำงานจะเป็นไปตามโฟลว์ชาร์ตดังรูป ที่ 4-18 ซึ่งมีขั้นตอนดังนี้

- จุดเริ่มต้นของการใช้บริการ คือ การเลือกเขตการ โดยถ้าเป็นผู้ใช้งานทั่วไปต้องค้นหารายการ เขตโดยอาศัยคีย์เวิร์ดของการค้นหา สำหรับสมาชิก ชื่อเขตจะมีให้อยู่แล้วสามารถเลือกได้ทันที ซึ่งโปรแกรมจะดึงข้อมูลเขตรายงานจากอาเรย์มาแสดงผลเป็นลำดับรายการ หรือจะค้นหารายการ เขตใหม่ก็ได้
- การค้นหาคือการติดต่อและส่งข้อมูลไปยังเซิร์ฟเวอร์ ถ้าการค้นหาสำเร็จเซิร์ฟเวอร์จะส่ง ข้อมูลที่เป็นลำดับรายการเขตให้ผู้ใช้เลือก
- ถ้าหากไม่พบรายการ โปรแกรมจะให้ผู้ใช้ระบุคีย์เวิร์ดและค้นหารายการเขตใหม่
- ผู้ใช้งานสามารถเลือกที่จะค้นหารายการได้มากกว่า 1 ครั้ง
- เมื่อผู้ใช้เลือกเขตรายงานได้แล้ว โปรแกรมจะมีการติดต่อกับเซิร์ฟเวอร์อีกครั้งเพื่อขอข้อมูลพื้นที่ ต่างๆ ในเขตรายงาน ซึ่งจะนำมาสู่ขั้นตอนการเลือกพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขั้นตอนต่อมาคือการร้องขอข้อมูลรายงานสภาพจราจรในพื้นที่ที่ได้เลือกไว้ แอปพลิเคชันจะทำการส่งข้อมูลพื้นที่รายงานไปที่ฝั่งเซิร์ฟเวอร์เพื่อให้ได้ผลลัพธ์ออกมา
- เซิร์ฟเวอร์ค้นหาหารายงานสภาพจราจรจากฐานข้อมูลแล้วจึงส่งผลลัพธ์การรายงานกลับมาให้ที่ฝั่งไคลเอนต์
- หลังจากที่ทราบผลลัพธ์แล้ว ผู้ใช้สามารถที่จะเข้าใช้บริการได้อีก หรือจะจบการทำงาน แอปพลิเคชันก็ได้

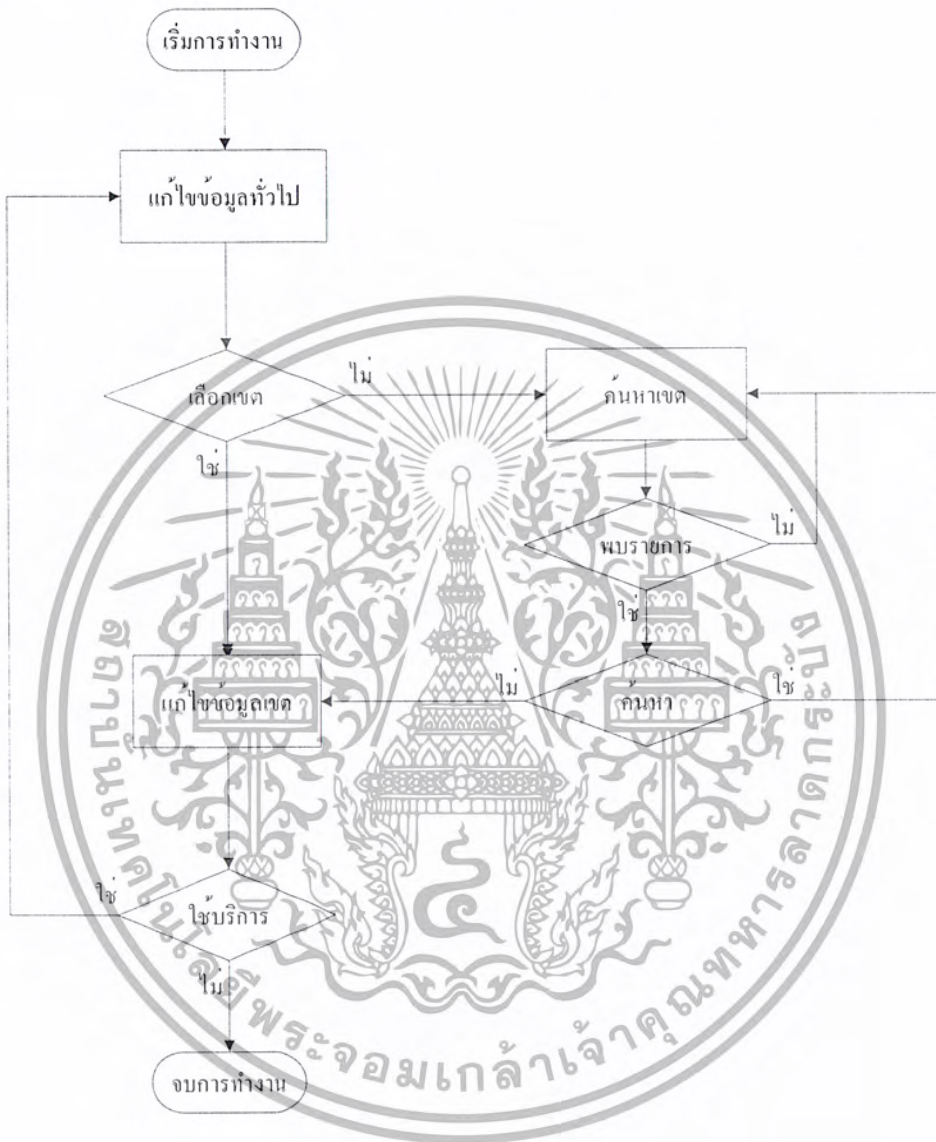


รูปที่ 4-18 แสดงโฟลว์ชาร์ตการทำงานของบริการรายงานสภาพจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การทำงานของบริการแก้ไขข้อมูลสมาชิก

ขั้นตอนการทำงานของบริการแก้ไขข้อมูล จะเป็นบริการที่เพิ่มเข้ามาสำหรับสมาชิกเท่านั้น โดยการทำงานจะเป็นไปตามโฟลว์ชาร์ตดังรูปที่ 4-19 ซึ่งมีขั้นตอนดังนี้



รูปที่4-19 แสดงโฟลว์ชาร์ตการทำงานของบริการแก้ไขข้อมูลสมาชิก

- จุดเริ่มต้นของการใช้บริการ คือ การแก้ไขข้อมูลทั่วไปในแต่ละฟิลด์ ซึ่งประกอบด้วย ข้อมูลชื่อสมาชิก, รหัสผ่าน, ชื่อสมาชิกจริง, อีเมล, ที่อยู่, ชื่อผู้ให้บริการ, เบอร์โทรศัพท์เคลื่อนที่, ชื่อผู้ใช้งานโทรศัพท์, รหัสผ่านโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นเป็นการแก้ไขข้อมูลเขตของสมาชิกซึ่งอาจจะต้องมีการค้นหาเพื่อหารายการเขตโดยอาศัยคีย์เวิร์ดของการค้นหา สำหรับสมาชิก ชื่อเขตจะมีให้อยู่แล้วสามารถเลือกได้ทันทีซึ่งโปรแกรมจะดึงข้อมูลเขตรายงานจากอาเรย์มาแสดงผลเป็นลำดับรายการ หรือจะค้นหารายการเขตใหม่ก็ได้
- การค้นหาคือการติดต่อและส่งข้อมูลคีย์เวิร์ดไปยังเซิร์ฟเวอร์ ถ้าการค้นหาสำเร็จเซิร์ฟเวอร์จะส่งข้อมูลที่ เป็นลำดับรายการเขตให้ผู้ใช้เลือก
- ถ้าหากไม่พบรายการ โปรแกรมจะให้ผู้ใช้นุคีย์เวิร์ดและค้นหาใหม่
- ผู้ใช้งานสามารถเลือกที่จะค้นหาได้รายการมากกว่า 1 ครั้ง
- เมื่อผู้ใช้แก้ไขข้อมูลทุกอย่างครบแล้ว โปรแกรมจะมีการติดต่อกับเซิร์ฟเวอร์เพื่อส่งข้อมูลทั้งหมด ไปอัปเดตลงฐานข้อมูล

หลังจากที่แก้ไขข้อมูลเสร็จแล้ว ผู้ใช้สามารถที่จะเข้าใช้บริการได้อีก หรือจะจบการทำงานแอปพลิเคชันก็ได้

4.4 การออกแบบแอปพลิเคชันบนโทรศัพท์เคลื่อนที่

แอปพลิเคชันจะถูกเขียนขึ้นและใช้ในฝั่งไคลเอนต์โดยภาษาที่ใช้หลักการของ Object Orientation ซึ่งมีการเขียนคลาสออกเป็นหลายคลาส เมธอดและตัวแปรต่างๆ จะถูกเรียกใช้ผ่านออบเจกต์ที่ได้ประกาศไว้ การออกแบบคลาสจะออกแบบตามประเภทของหน้าการแสดงผล เช่น คลาส UserScreen สำหรับแสดงหน้าการเลือกประเภทของผู้ใช้บริการ เป็นต้น ทั้งนี้เพื่อแยกหน้าที่การทำงานในแต่ละหน้าอย่างชัดเจน ซึ่งประกอบด้วยคลาสทั้งหมด ดังนี้

4.4.1 คลาส TrafficReport

เป็นคลาสหลักที่มีการประกาศเมธอดต่างๆ ที่ใช้สำหรับประมวลผล และการแสดงผลออกทางจอภาพของโทรศัพท์เคลื่อนที่ โดยการแสดงผลจะใช้การประกาศออบเจกต์ของคลาสอื่นๆ ร่วมกับคลาส Display ด้วย เมธอดต่างๆ ในคลาส TrafficReport มีดังนี้

- เมธอด TrafficReport() เป็นคอนสตรัคเตอร์ของคลาสที่ไม่มีการใช้งาน
- เมธอด StartApp() เป็นเมธอดของ MIDlet ใช้เพื่อจองทรัพยากรของระบบ และถือเป็นจุดเริ่มของโปรแกรมด้วย เมื่อ MIDlet เริ่มทำงานเมธอดนี้จะถูกเรียกขึ้นมา
- เมธอด PauseApp() เป็นเมธอดของ MIDlet ที่บอก MIDlet ให้พักการทำงานชั่วคราวและคืนทรัพยากรสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมธอด DestroyApp() เป็นเมธอดของ MIDlet เพื่อจบการทำงานของ MIDlet และคืนทรัพยากรทั้งหมดกลับสู่ระบบ
- เมธอด goUser() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส UserScreen เพื่อแสดงหน้าการเลือกประเภทผู้ใช้บริการ ออกทางจอภาพ
- เมธอด goLogin() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส LoginScreen เพื่อแสดงหน้าการตรวจสอบความเป็นสมาชิกของผู้ใช้งานระบบ ออกทางจอภาพ
- เมธอด goMember() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส MemberScreen เพื่อแสดงหน้าการเลือกประเภทของบริการสำหรับสมาชิก ออกทางจอภาพ โดยในคลาสนี้จะมีการติดต่อกับเซิร์ฟเวอร์เพื่อร้องขอข้อมูลของสมาชิก
- เมธอด goNonmember() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส NonmemberScreen เพื่อแสดงหน้าการเลือกประเภทของบริการสำหรับผู้ทั่วไปที่ไม่ใช่สมาชิก ออกทางจอภาพ
- เมธอด goSearchsource() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส SearchsourceScreen เพื่อแสดงหน้าการเลือกและค้นหาเส้นทาง ออกทางจอภาพ
- เมธอด goSearchdest() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส SearchdestScreen เพื่อแสดงหน้าการเลือกและค้นหาปลายทาง ออกทางจอภาพ
- เมธอด goCost() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส CostScreen เพื่อแสดงหน้าการเลือกปัจจัยในการค้นหาเส้นทาง ออกทางจอภาพ
- เมธอด goReport() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส ReportareaScreen เพื่อแสดงหน้าการเลือกและค้นหาเขตรายงาน ออกทางจอภาพ
- เมธอด goRange() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส ReportrangeScreen เพื่อแสดงหน้าการเลือกพื้นที่ของเขตรายงาน ออกทางจอภาพ
- เมธอด goResponse(String input) เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส ResponseScreen เพื่อแสดงผลลัพธ์เนื่องจากการค้นหาเส้นทางและการรายงานสภาพจราจรในพื้นที่เขตออกทางจอภาพ
- เมธอด goEdituser() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส EdituserScreen เพื่อแสดงหน้าการแก้ไขข้อมูลของสมาชิก ออกทางจอภาพ
- เมธอด goEditdistrict() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส EditdistrictScreen เพื่อแสดงหน้าการแก้ไขข้อมูลเขตรายงาน ออกทางจอภาพ
- เมธอด goEditok() เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส Alert เพื่อแสดงหน้าจอการแก้ไขข้อมูลสมาชิกสำเร็จ โดยจะแสดงออกมาชั่วระยะเวลาหนึ่งก่อนที่จะเข้าสู่หน้าประเภทของบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมธอด `goError(String flag)` เป็นเมธอดที่มีการประกาศออบเจกต์ของคลาส `Alert` เพื่อแสดงหน้าความผิดพลาดอันเนื่องมาจากบริการต่างๆ ออกทางจอภาพ โดยในหน้าของความผิดพลาดจะแสดงในช่วงระยะเวลาหนึ่งก่อนที่จะเข้าสู่หน้าประเภทของบริการ
- เมธอด `goConnect(String input)` เป็นเมธอดที่ใช้สำหรับส่งข้อมูลร้องขอและรับข้อมูลตอบรับต่างๆ ระหว่างผู้ใช้งาน โทรศัพท์เคลื่อนที่และเครื่องเซิร์ฟเวอร์ผู้ให้บริการ

4.4.2 คลาส `UserScreen`

เป็นคลาสเพื่อแสดงประเภทของผู้ใช้บริการ คือ ผู้ใช้บริการที่เป็นสมาชิก และผู้ให้บริการทั่วไป สืบทอดมาจากคลาส `List` ที่แสดงเป็นลำดับรายการให้ผู้ใช้งานเลือกได้ ประกอบด้วยตัวแปรและเมธอดต่างๆ ที่สำคัญ ดังนี้

- ตัวแปร `userStatus` เป็นตัวแปรชนิด บูลีน ใช้สำหรับเก็บสถานะของผู้ใช้ ปกติเมื่อโปรแกรมทำงานค่า `userStatus` จะถูกกำหนดให้เป็น `false` ซึ่งจะหมายถึงผู้ใช้งานในขณะนี้เป็นผู้ใช้งานทั่วไป และ `userStatus` จะถูกกำหนดเป็น `true` สำหรับสมาชิก
- เมธอด `UserScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ลำดับรายการเลือกและ ปุ่ม `Select`
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `UserScreen`

4.4.3 คลาส `LoginScreen`

เป็นคลาสที่แสดงหน้าของกรเข้าสู่ระบบของสมาชิก สืบทอดคุณสมบัติมาจากคลาส `Form` โดยคลาส `LoginScreen` จะมีตัวแปรและเมธอดที่สำคัญ ดังนี้

- ตัวแปร `usr_pass` เป็นตัวแปรชนิดสตริง ใช้สำหรับเก็บข้อมูลชื่อและรหัสผ่านของผู้ใช้ เพื่อใช้ในการร้องขอข้อมูลสมาชิกที่เครื่องเซิร์ฟเวอร์
- เมธอด `LoginScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ `Form` ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลชื่อผู้ใช้, ฟิลด์ข้อมูลรหัสผ่าน ปุ่ม `Select` และปุ่ม `Back`
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `LoginScreen`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4 คลาส MemberScreen

เป็นคลาสแสดงหน้าบริการสำหรับสมาชิก ที่สืบทอดมาจากคลาส List ได้แก่ บริการค้นหาเส้นทาง, บริการรายงานสภาพจราจร และบริการการแก้ไขข้อมูลสมาชิก ซึ่งเป็นบริการเฉพาะสำหรับสมาชิกเท่านั้น คลาส MemberScreen จะมีตัวแปรและเมธอดที่สำคัญดังนี้

- ตัวแปร infoList เป็นตัวแปรชนิดอาเรย์ แบบสตริง ขนาด 14 ช่องที่ใช้สำหรับเก็บข้อมูลต่างๆ ของสมาชิก ได้แก่ ชื่อสมาชิก, รหัสผ่าน, ชื่อจริง, อีเมลล์, ที่อยู่, ชื่อเขตรายงาน, ผู้ให้บริการ โทรศัพท์, เบอร์โทรศัพท์, ชื่อผู้ใช้โทรศัพท์, รหัสผ่านโทรศัพท์, ต้นทาง1, ต้นทาง2, ปลายทาง1 และ ปลายทาง2 สำหรับข้อมูลเขตรายงาน, ต้นทางและปลายทาง จะช่วยเพิ่มความสะดวกให้กับสมาชิกในกรณีที่ต้องการค้นหาเส้นทางจราจรที่ไปเป็นประจำ หรือต้องการทราบรายงานสภาพจราจรในเขตพื้นที่รายงานเดิม สมาชิกไม่ต้องเสียเวลากับการค้นหารายชื่อ
- เมธอด MemberScreen(TrafficReport midlet, String input) เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ List ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, รายการบริการต่างๆ และ ปุ่ม Select รวมถึงการนำผลลัพธ์ของข้อมูลสมาชิกที่ได้จากเครื่องเซิร์ฟเวอร์มาเก็บลงในอาเรย์
- เมธอด commandAction(Command c, Displayable d) เป็นเมธอดที่มีอยู่ในแพ็คเกจของ j2me อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส MemberScreen

4.4.5 คลาส NonmemberScreen

เป็นคลาสแสดงหน้าบริการสำหรับผู้ใช้งานทั่วไป สืบทอดมาจากคลาส List ได้แก่ บริการค้นหาเส้นทาง, บริการรายงานสภาพจราจร NonmemberScreen จะมีเมธอดที่สำคัญดังนี้

- เมธอด NonmemberScreen(TrafficReport midlet) เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ List ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, รายการบริการสำหรับผู้ใช้งานทั่วไป
- เมธอด commandAction(Command c, Displayable d) เป็นเมธอดที่มีอยู่ในแพ็คเกจของ j2me อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส NonmemberScreen

4.4.6 คลาส SearchsourceScreen

เป็นคลาสแสดงหน้าบริการค้นหาต้นทางจราจร สืบทอดมาจากคลาส Form เพื่อวางคอมโพเนนต์ชนิด Item ได้หลากหลาย SearchsourceScreen จะมีตัวแปรและเมธอดที่สำคัญดังนี้

- ตัวแปร sourceList เป็นตัวแปรชนิดอาเรย์ แบบสตริง ขนาด 50 ช่องใช้สำหรับเก็บข้อมูลรายชื่อต้นทางที่ค้นหาเพื่อเตรียมมาวางเป็นลำดับรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแปร `routeArray` เป็นตัวแปรชนิดอาร์เรย์ แบบสตริง ขนาด 5 ช่อง ใช้สำหรับเก็บข้อมูล ต้นทาง, ปลายทางและปัจจัยในการค้นหาเส้นทาง เพื่อใช้ในการค้นหาเส้นทางจากต้นทางไปยัง ปลายทางโดยคำนึงถึงปัจจัยที่เลือก โดยจะส่งเป็นข้อมูลร้องขอไปยังเครื่องเซิร์ฟเวอร์
- เมธอด `SearchsourceScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้าง องค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลสำหรับใช้ ค้นหารายการชื่อต้นทาง, คอมโพเนนต์ ประเภทแสดงลำดับรายการ, ปุ่ม Select, ปุ่ม Search และ ปุ่ม Back
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `SearchsourceScreen`

4.4.7 คลาส `SearchdestScreen`

เป็นคลาสแสดงหน้าบริการค้นหาปลายทางจราจร สืบต่อมาจากคลาส Form เพื่อวางคอมโพเนนต์ชนิด Item ได้หลากหลาย `SearchdestScreen` จะมีตัวแปรและเมธอดที่สำคัญดังนี้

- ตัวแปร `destList` เป็นตัวแปรชนิดอาร์เรย์ แบบสตริง ขนาด 50 ช่อง ใช้สำหรับเก็บข้อมูลรายชื่อ ปลายทางที่ค้นหาเพื่อเตรียมวางเป็นลำดับรายการ
- เมธอด `SearchdestScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้าง องค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลสำหรับใช้ ค้นหาชื่อปลายทาง, คอมโพเนนต์ประเภทลำดับรายการ สำหรับแสดงรายการชื่อปลายทาง, ปุ่ม Select, ปุ่ม Search และปุ่ม Back
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `SearchdestScreen`

4.4.8 คลาส `CostScreen`

เป็นคลาสแสดงหน้ารายการเลือกปัจจัยในการค้นหาเส้นทาง ได้แก่ ค่าใช้จ่าย, เวลา และระยะทาง ตามความต้องการในการค้นหาเส้นทางที่แตกต่างกันของผู้ใช้บริการ สืบต่อมาจากคลาส List โดยมี เมธอดที่สำคัญดังนี้

- เมธอด `CostScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, คอมโพเนนต์ประเภท Choice Group สำหรับแสดงรายการปัจจัยที่มีผลต่อการค้นหาเส้นทาง, ปุ่ม Route และปุ่ม Back
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `CostScreen`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.9 คลาส ReportareaScreen

เป็นคลาสที่ใช้สำหรับการแสดงผลหน้ารายการเพื่อให้ผู้ใช้งานสามารถเลือกเขตที่ต้องการทราบสภาพจราจรในขณะนั้น สืบทอดมาจากคลาส Form มีตัวแปรและเมธอดที่สำคัญ ดังนี้

- ตัวแปร `areaList` เป็นตัวแปรอาร์เรย์แบบสตริงขนาด 50 ช่อง สำหรับเก็บรายการเขตที่ค้นหาได้ เพื่อนำไปแสดงผล โดยข้อมูลที่ได้อาจได้มาจากการร้องขอที่เครื่องเซิร์ฟเวอร์
- เมธอด `ReportareaScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, คอมโพเนนต์ Choice Group แสดงลำดับรายการเขต, ฟิลด์ข้อมูลรหัสผ่าน ปุ่ม Select, ปุ่ม Search และปุ่ม Back
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `ReportareaScreen`

4.4.10 คลาส ReportrangeScreen

เป็นคลาสสำหรับการแสดงผลหน้ารายการพื้นที่ที่มีการรายงานสภาพจราจรในขณะนั้น โดยจะแสดงในลักษณะของลำดับรายการ มีคุณสมบัติสืบทอดมาจากคลาส List มีตัวแปรและเมธอดที่สำคัญดังนี้

- เมธอด `ReportrangeScreen(TrafficReport midlet, String input)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ List ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลผลลัพธ์, ปุ่ม Report และปุ่ม Back รวมถึงขั้นตอนการนำข้อมูลผลลัพธ์มาใส่ในฟิลด์ข้อมูล
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `ReportrangeScreen`

4.4.11 คลาส ResponseScreen

เป็นคลาสที่ใช้ในการแสดงผลพื้นที่เนื่องจากการค้นหาเส้นทางหรือการรายงานสภาพการจราจร โดยจะรับข้อมูลเข้ามาและนำข้อมูลไปแสดงผลออกทางจอภาพ คลาส `ResponseScreen` จะมีคุณสมบัติสืบทอดมาจากคลาส Form โดยจะมีเมธอด ต่างๆ ที่สำคัญดังนี้

- เมธอด `ResponseScreen(TrafficReport midlet)` เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลผลลัพธ์, ปุ่ม Service และปุ่ม Back
- เมธอด `commandAction(Command c, Displayable d)` เป็นเมธอดที่มีอยู่ในแพ็คเกจของ `j2me` อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส `ResponseScreen`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.12 คลาส EdituserScreen

เป็นคลาสใช้สำหรับสมาชิกในกรณีที่สมาชิกเรียกใช้บริการการแก้ไขข้อมูล โดยคลาสจะเป็นส่วนแสดงผลหน้าฟิลด์ข้อมูลต่างๆ เกี่ยวกับสมาชิกทั้งหมด เช่น ข้อมูลชื่อ, ข้อมูลที่อยู่ และข้อมูลชื่ออีเมลล์ ทั้งนี้เพื่อให้สมาชิกสามารถแก้ไขข้อมูลบนโทรศัพท์เคลื่อนที่ได้ มีตัวแปรและเมธอดต่างๆ ที่สำคัญ ดังนี้

- ตัวแปร pmet เป็นตัวแปรอารีย์ ชนิดสตริงขนาด 13 ช่อง ใช้สำหรับเก็บข้อมูลใหม่ที่สมาชิกแก้ไขแล้ว ยกเว้นข้อมูลเขตรายงานที่สมาชิกจะต้องเลือกในหน้าแก้ไขข้อมูลเขตรายงานซึ่งแยกออกไปต่างหาก ก่อนที่จะส่งไปเปลี่ยนแปลงที่ฐานข้อมูลบนเครื่องเซิร์ฟเวอร์
- เมธอด EdituserScreen(TrafficReport midlet) เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลค้นหารายการเขต, คอมโพเนนต์ลำดับรายการ ปุ่ม Next และปุ่ม Back
- เมธอด commandAction(Command c, Displayable d) เป็นเมธอดที่มีอยู่ในแพ็คเกจของ j2me อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส EdituserScreen

4.4.13 คลาส EditdistrictScreen

เป็นคลาสใช้สำหรับสมาชิกในกรณีที่สมาชิกเรียกใช้บริการการแก้ไขข้อมูล โดยคลาสจะเป็นส่วนแสดงผลหน้าฟิลด์ข้อมูลเฉพาะข้อมูลเขตรายงานเท่านั้น เนื่องจากจะมีลำดับรายการเขตให้สมาชิกสามารถเลือกได้ ซึ่งระยะเวลาในการพิมพ์ชื่อที่ถูกคือ คลาส EditdistrictScreen สืบทอดมาจากคลาส Form โดยมีเมธอดต่างๆ ที่สำคัญ ดังนี้

- เมธอด EditdistrictScreen(TrafficReport midlet) เป็นคอนสตรัคเตอร์สำหรับสร้างองค์ประกอบของ Form ในการแสดงผล ได้แก่ หัวข้อของหน้าแสดงผล, ฟิลด์ข้อมูลค้นหารายการเขตรายงาน, คอมโพเนนต์ลำดับรายการเขต, ปุ่ม Edit, ปุ่ม Search และปุ่ม Back
- เมธอด commandAction(Command c, Displayable d) เป็นเมธอดที่มีอยู่ในแพ็คเกจของ j2me อยู่แล้วใช้ตรวจสอบเหตุการณ์ของการกดปุ่ม ในคลาส EditdistrictScreen

4.5 การออกแบบไฟล์ประมวลผลบนฝั่งเซิร์ฟเวอร์ของโทรศัพท์เคลื่อนที่

ที่ฝั่งเซิร์ฟเวอร์จะเขียนเป็นไฟล์ JSP ทั้งหมด 9 ไฟล์ โดยแต่ละไฟล์จะทำหน้าที่ในการติดต่อไปยังฐานข้อมูล ด้วยคำสั่งในการค้นหาข้อมูล ที่แตกต่างกัน จากนั้นจะนำค่าข้อมูลที่ได้ส่งกลับไปฝั่งไคลเอนต์เพื่อใช้ประมวลผลต่อไป ไฟล์ต่างๆ มีดังนี้

4.5.1 ไฟล์ Server_Authen.jsp

เป็นไฟล์สำหรับการตรวจสอบความเป็นสมาชิกของผู้ใช้งาน ทำหน้าที่ในการรับข้อมูล ชื่อสมาชิกและรหัสผ่าน และใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อสมาชิก และฟิลด์รหัสผ่านตรงกับข้อมูลที่รับเข้ามา ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะมีค่ารีเทิร์นกลับมาเป็นเลข 1 เซิร์ฟเวอร์จะส่งค่ากลับไปเป็นสตริงว่า "ok" ถ้าค้นหาไม่พบ ผลลัพธ์ที่ได้จะมีค่ารีเทิร์นกลับมาเป็นเลขอื่นที่ไม่ใช่เลข 1 และเซิร์ฟเวอร์จะส่งค่ากลับไปเป็นสตริงว่า "no"

```
ResultSet myresult = stmt.executeQuery("SELECT count(*) as num FROM member WHERE
usr='"+requirer+"' and pass='"+reqpass+"'"); //set sql command to search username and password field
while(myresult.next()) {
    if(myresult.getInt("num")==1)
        out.println("ok"); //search complete return ok
    else
        out.println("no"); //search fail return no
}
```

4.5.2 ไฟล์ Server_EditSearch.jsp

เป็นไฟล์สำหรับการค้นหาข้อมูลต่างๆ ที่เกี่ยวกับสมาชิก เพื่อใช้ในบริการประเภทต่างๆ เช่น บริการค้นหาเส้นทาง จะมีข้อมูลต้นทางและปลายทางปรากฏให้สมาชิกได้เลือกอยู่แล้ว เป็นการประหยัดเวลาในการเลือกรายการต้นทางและปลายทาง ไฟล์จะทำหน้าที่ในการรับข้อมูล ชื่อสมาชิกและรหัสผ่าน จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อสมาชิกและฟิลด์รหัสผ่านตรงกับข้อมูลที่รับเข้ามา ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของข้อมูลสมาชิกส่งกลับไปฝั่งไคลเอนต์เพื่อนำไปใช้งานต่อไป ถ้าค้นหาไม่พบผลลัพธ์จะเป็นอย่างอื่นที่ไม่ใช่ข้อมูลสมาชิก

```
ResultSet myresult = stmt.executeQuery("SELECT * FROM member WHERE usr='"+requirer+"' and
pass='"+reqpass+"'"); //set sql command to search member information
if(myresult != null) {
    while(myresult.next()) {
        output = myresult.getString("usr")+','+ //get information in each field and-
myresult.getString("pass")+','+ //concatenate with the original output
myresult.getString("name")+','+
myresult.getString("email")+','+
myresult.getString("district")+','+
myresult.getString("address")+','+

```

```

myresult.getString("mobile")+','+
myresult.getString("provider")+','+
myresult.getString("mo_usr")+','+
myresult.getString("mo_pass")+','+
myresult.getString("start1")+','+
myresult.getString("dest1")+','+
myresult.getString("start2")+','+
myresult.getString("dest2")+','+$";
}
out.println(output); //return to client
}

```

4.5.3 ไฟล์ Server_RouteSearch.jsp

เป็นไฟล์สำหรับการค้นหารายการข้อมูล ต้นทางหรือปลายทาง ในกรณีที่ผู้ใช้บริการไม่ทราบชื่อของต้นทางหรือปลายทางที่ต้องการ ไฟล์จะทำหน้าที่ในการรับข้อมูลที่เป็นคีย์เวิร์ดในการค้นหา จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหารายการต้นทางหรือปลายทางที่มีตัวอักษรในข้อมูลตรงกับคีย์เวิร์ด ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของข้อมูลรายการต้นทางหรือปลายทาง ส่งกลับไปที่ฝั่งไคลเอนต์เพื่อนำไปใช้ผู้ใช้งานได้เลือก ถ้าค้นหาไม่พบผลลัพธ์จะเป็นอย่างอื่นที่ไม่ใช่ข้อมูลต้นทางและปลายทาง

```

ResultSet myresult = stmt.executeQuery("SELECT * FROM matching WHERE word like
%" + reqsearch + "%"); //set sql command to search anywhere matched keyword
while(myresult.next()){
    output = output + myresult.getString("word") + ','; //concatenate result with original output
}
output = output + '$';
out.println(output); //return to client

```

4.5.4 ไฟล์ Server_RouteJunction.jsp

เป็นไฟล์สำหรับการค้นหาข้อมูลแยกของถนนหรือเรียกว่า โหนด เพื่อใช้ในการค้นหาเส้นทางหลังจากที่ได้ต้นทางหรือปลายทางแล้วระบบจะติดต่อมายังไฟล์ โดยไฟล์จะทำหน้าที่ในการรับข้อมูลชื่อต้นทางหรือปลายทาง จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อต้นทางหรือปลายทางตรงกับข้อมูลที่รับเข้ามา ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของข้อมูลแยกหรือ โหนดของต้นทางหรือปลายทาง ส่งกลับไปที่ฝั่งไคลเอนต์เพื่อนำไปใช้งานต่อไป ถ้าค้นหาไม่พบผลลัพธ์จะเป็น

อย่างอื่นที่ไม่ใช่ข้อมูลสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ResultSet myresult = stmt.executeQuery("SELECT * FROM matching WHERE word='"+reqjunc+"'");
while(myresult.next()){           //set sql command to get junction nearby places
    output = output + myresult.getString("inter1") + ',' + myresult.getString("inter2") + ',';
}
//get junction and concatenate with the original output
output = output + '$';           //return to client
out.println(output);

```

4.5.5 ไฟล์ Server_RouteResponse.jsp

เป็นไฟล์สำหรับการส่งข้อมูลไปยังโปรแกรมค้นหาเส้นทาง ไฟล์จะทำหน้าที่ในการรับข้อมูล ชื่อต้นทางและปลายทาง ได้แก่ ต้นทาง1, ต้นทาง2, ปลายทาง1, ปลายทาง2 และ ปัจจัยการค้นหาเส้นทาง จากนั้นใช้จาวาบิน ในการติดต่อไปยังโปรแกรมค้นหาเส้นทาง เมื่อการค้นหาเสร็จสิ้นระบบจะได้เส้นทางที่ดีที่สุดส่งกลับไปยังฝั่งไคลเอนต์ ซึ่งเป็นเส้นทางที่ผู้ใช้สามารถเดินทางได้อย่างมีประสิทธิภาพ

4.5.6 ไฟล์ Server_AreaSearch.jsp

เป็นไฟล์สำหรับการค้นหาข้อมูลเขตพื้นที่รายงาน เพื่อใช้ในบริการ รายงานสภาพจราจรและ บริการแก้ไขข้อมูลสมาชิก ซึ่งต้องมีการค้นหารายชื่อเขตที่ต้องการ ไฟล์จะทำหน้าที่ในการรับข้อมูลคีย์เวิร์ดเข้ามา จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาในแถวข้อมูลที่มีตัวอักษรตรงกับคีย์เวิร์ด ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของข้อมูลรายการเขตส่งกลับไปฝั่งไคลเอนต์เพื่อนำไปเลือกพื้นที่รายงานต่อไป ถ้าค้นหาไม่พบผลลัพธ์จะเป็นอย่างอื่นที่ไม่ใช่ข้อมูลเขต

```

ResultSet myresult = stmt.executeQuery("SELECT distr_name FROM district WHERE distr_name like
"+"%"+reqarea+"%");           //set sql command to search area matched keyword
while(myresult.next()){
    output = output + myresult.getString("distr_name") + ','; //concatenate areas with the-
}
//original output
output = output + ",$";           //return to client

```

4.5.7 ไฟล์ Server_RangeSearch.jsp

เป็นไฟล์สำหรับการค้นหาข้อมูลพื้นที่ในเขตรายงาน จะทำงานหลังจากที่ผู้ใช้งานได้เลือกเขตแล้ว พื้นที่รายงานจะเป็นส่วนหนึ่งของเขตรายงาน เนื่องจากเขต 1 เขตมีขนาดใหญ่ โดยไฟล์จะทำหน้าที่ในการรับข้อมูลชื่อเขตรายงาน จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อเขตตรงกับข้อมูลที่รับเข้ามา ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของรายการพื้นที่ต่างๆ ในเขตที่การ

รายงานสภาพจราจร และจะส่งกลับไปฝั่งไคลเอนต์ให้ผู้ใช้ได้เลือกต่อไป ถ้าค้นหาไม่พบผลลัพธ์จะเป็นอย่างอื่นที่ไม่ใช่ข้อมูลรายการพื้นที่

```
ResultSet myresult = stmt.executeQuery("SELECT * FROM report WHERE district='"+reqarea+"'");
while(myresult.next()){
    // set sql command to search area
    output = output + myresult.getString("street") + ','; //get the ranges in the area
}
output = output + "$"; //return to client
```

4.5.8 ไฟล์ Server_ReportResponse.jsp

เป็นไฟล์สำหรับการค้นหาข้อมูลรายละเอียดสภาพการจราจรในเขตพื้นที่ที่เลือกไว้ ไฟล์จะทำหน้าที่ในการรับข้อมูลชื่อพื้นที่รายงานเข้ามา จากนั้นใช้คำสั่งค้นหาในฐานข้อมูล ค้นหาแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อพื้นที่ตรงกับข้อมูลที่รับเข้ามา ถ้าค้นหาเจอ ผลลัพธ์ที่ได้จะเป็นสตริงของข้อมูลรายละเอียดการรายงานสภาพการจราจรในพื้นที่บริเวณนั้น เช่น เกิดอุบัติเหตุ หรือ มีสิ่งก่อสร้าง เป็นต้น ข้อมูลจะถูกส่งกลับไปฝั่งไคลเอนต์เพื่อนำไปแสดงผลออกทางจอภาพ และถ้าค้นหาไม่พบผลลัพธ์จะเป็นอย่างอื่นที่ไม่ใช่ข้อมูลรายละเอียดการรายงานสภาพการจราจร

```
ResultSet myresult = stmt.executeQuery("SELECT * FROM report WHERE street='"+reqstr+"'");
while(myresult.next()){
    // set sql command to get report by range in the area
    output = myresult.getString("date")+','+myresult.getString("comment");
}
//get report
output = output + "$"; //return to client
```

4.5.9 ไฟล์ Server_EditUpdate.jsp

เป็นไฟล์สำหรับการแก้ไขข้อมูลในฐานข้อมูลของสมาชิก โดยไฟล์จะทำหน้าที่ในการรับข้อมูลทั้งหมดของสมาชิก จากนั้นใช้คำสั่งอัปเดตฐานข้อมูล อัปเดตแถวของข้อมูลที่มีข้อมูลในฟิลด์ชื่อสมาชิกตรงกับข้อมูลที่รับเข้ามา ถ้าการอัปเดตสำเร็จ ผลลัพธ์ที่ได้จะเป็นเลข 1 ถ้าค้นหาไม่พบผลลัพธ์จะเป็นเลขอื่นที่ไม่ใช่เลข 1 และจะส่งกลับไปฝั่งไคลเอนต์เพื่อใช้งานต่อไป

```
String sql = "UPDATE member SET
usr='"+requser+"',pass='"+reqpass+"',name='"+reqname+"',email='"+reqemail+"',district='"+reqdistr+"',
address='"+reqaddr+"',mobile='"+reqmob+"',provider='"+reqprov+"',mo_usr='"+reqmours+"',
mo_pass='"+reqmopss+"' WHERE usr='"+requser+"'"; //set sql to go update member information
int result = stmt.executeUpdate(sql);
```

```
if(result == 1)                                //update succeeded
    out.println("ok");
else                                            //update not succeeded
    out.println("no");
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง/ทดสอบ

5.1 ผลการทดลองในส่วนของเว็บไซต์

เมื่อทำการเรียกเว็บไซต์หรือโฮมเพจขึ้นมาจะได้ผลดังนี้

<http://161.246.6.11:8080/home.jsp>



สามารถใช้ลิงก์ทุกลิงก์ได้ไม่มีปัญหาเกิดขึ้น และสำหรับการเรียกเพจโดยตรงไปยังเพจอื่นๆที่สมาชิกต้องทำการล็อกอินก่อนจะไม่สามารถทำได้ เพจจะทำการเรียกไปยัง login.jsp โดยอัตโนมัติ คือ จะไม่ให้เข้าถ้าไม่ผ่านการล็อกอิน

ทั้งนี้ URL ที่เรียกพอร์ต 8080 นั้นเป็นการตั้งค่าของเว็บเซิร์ฟเวอร์ให้เปิดเป็นพอร์ตสำหรับการใช้งานเว็บนี้เท่านั้น เพื่อไม่ให้ชนกับการใช้งานของพอร์ตอื่นๆ (แทนที่จะเป็นพอร์ต 80) แต่สามารถเปลี่ยนแปลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home>> Intelligent Bangkok Traffic Raster Sat Mar 13 23:27:04 ICT 2004

Welcome to Bangkok

Power By : KMITL CE

Register Login Map Routing Traffic Report Live Radio Report

สมัครสมาชิก

Login: * Eng

Password: * Eng

Re-type password: * Eng

Name: * Eng

Email: * Eng

District: * Eng

Address: * Eng

Mobile Phone#: * Eng

Phone's provider: DTAC AIS Orange Hutch * Eng

Phone's user: * Eng

Phone's password: * Eng

Start 1:

Destination 1:

Start 2:

Destination 2:

Map Search: * Eng

Start 1: Destination 1: Start 2: Destination 2:

word 2: Lay 1 Lay 2

word 1: Lay 1 Lay 2

Submit Reset

รูปที่ 5-2 แสดงผลการเรียก Register.jsp

การกรอกข้อมูลนั้น อินพุตจะต้องอยู่ในขอบเขตที่กำหนดไว้เท่านั้น (ต้องระบุฟิลด์ที่มีเครื่องหมาย *, ข้อความต้องไม่มีอักขระพิเศษ และจุด start และ destination ต้องไม่เป็นจุดเดียวกัน หากมีการเลือก start และ destination คู่ใด ก็จะต้องมีการเลือกครบคู่ด้วย) ถึงจะผ่านการสมัครสมาชิกได้ และหากเกิดข้อผิดพลาดขึ้นไปในเพจใดก็จะทำการเรียกไปยัง error.jsp เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ของภัยเกิดเหตุขุดของ

Internal Server Error

รูปที่ 5-3 หน้าแสดงผลเมื่อเกิดข้อผิดพลาดขึ้น error.jsp

หน้าล็อกอินจะทำการตรวจสอบชื่อล็อกอินและรหัสผ่านจากฐานข้อมูล โดยที่ชื่อล็อกอินนั้นจะไม่ขึ้นกับตัวอักษรตัวเล็กหรือใหญ่ (non case-sensitive)



รูปที่ 5-4 แสดงผลการเรียก login.jsp

รูปที่ 5-4 แสดงผลการเรียก login.jsp

และเมื่อสมาชิกทำการล็อกอินเข้าสู่ระบบอย่างถูกต้องก็จะได้ผลดังนี้



รูปที่ 5-5 แสดงผลการเมื่อผ่านการล็อกอิน login.jsp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสมาชิกต้องการจะแก้ไขข้อมูลส่วนตัวของตนเอง ที่เคยให้ไว้กับระบบในขั้นตอนการสมัครสมาชิกที่ผ่านมาจะเป็นดังนี้

Home> Intelligent Bangkok Traffic Router Sun Mar 14 01:48:00 ICT 2004

Welcome to Bangkok

PowerBy : KMITL CE

[Register](#) [Logout](#) [Map](#) [Routing](#) [Traffic Report](#) [Live Radio Report](#)

แก้ไขข้อมูลสมาชิก

Login: test

Password: Eng

Re-type password: Eng

Name: Eng

Email: Eng

District: Eng

Address: Eng

Mobile Phone#: Eng

Phone's provider: DTAC AIS Orange Hutch

Phone's user: Eng

Phone's password: Eng

Start to Destination: #1

word1: Eng

word2: Eng

Start to Destination: #2

word1: Eng

word2: Eng

รูปที่ 5-6 แสดงผลเมื่อต้องการแก้ไขข้อมูล `edit.jsp`

คือจะทำการแสดงข้อมูลเดิมขึ้นมาให้ทำทราบก่อน และเมื่อคลิกปุ่ม “Reset” ข้อมูลที่กำลังแก้ไขอยู่ก็จะกลับมาเป็นข้อมูลตั้งต้นเดิมในฐานะข้อมูล และหากแก้ไขข้อมูลเรียบร้อยแล้วจะได้ผลลัพธ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-7 แสดงผลเมื่อแก้ไขข้อมูลเรียบร้อยแล้ว

หากสมาชิกยังต้องการแก้ไขข้อมูลในส่วนของคุณเริ่มต้นและปลายทางก็จะต้องกดปุ่ม “change” และได้ผลลัพธ์ดังนี้



รูปที่ 5-8 แสดงผลเมื่อต้องการแก้ไขต่อ edit_cont.jsp

ซึ่งในส่วนที่เป็น radio box นั้นจะเป็นผลจากการค้นหาจุดเริ่มต้นและสิ้นสุด และสมาชิกจะต้องเลือกจุดทั้งสองจุดเป็นคนละที่กันเท่านั้น จึงจะผ่านการแก้ไขข้อมูลได้ และก็ต้องเลือกด้วยว่าจะแก้ไขข้อมูลของจุดเริ่มต้นและสิ้นสุดคู่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสมาชิกทำการค้นหารายงานอุบัติเหตุและสภาพการจราจรจะได้ผลลัพธ์ดังนี้

Home>> Intelligent Bangkok Traffic Reporter Sun Mar 14 02:11:54 ICT 2004

Welcome to *Bangkok*

PowerBy : KMITL CE

Register | Logout | Map | Routing | Traffic Report | Live Radio Report

Edit your information

Map

Routing

Traffic Report

Live Radio

Webboard

Logout

รายงานสภาพจราจร

District : Lat Krabang Search

Grand Palace

Chalong Krung - Lat Krabang

OK

Home>> Intelligent Bangkok Traffic Reporter Sun Mar 14 02:13:46 ICT 2004

Welcome to *Bangkok*

PowerBy : KMITL CE

Register | Logout | Map | Routing | Traffic Report | Live Radio Report

Edit your information

Map

Routing

Traffic Report

Live Radio

Webboard

Logout

รายงานสภาพการจราจร

District : Lat Krabang Search

ถนน - ท.ก.	รายละเอียด
Chalong Krung - Lat Krabang	very slow traffic

รูปที่ 5-9 และ 5-10 แสดงผลเมื่อสมาชิกต้องการรายงานจราจร member_report.jsp

โดยที่มีความแตกต่างจากผู้ที่ไม่ได้ล็อกอินเข้ามาตรงที่ ในส่วนข้อความที่ต้องการค้นหาจะไม่มีข้อความใดๆ ขึ้นมาให้ก่อนเลย จะต้องทำการป้อนอินพุทหรือบริเวณที่ต้องการค้นหาเอง (ข้อความที่ขึ้นมาให้เป็นบริเวณที่สมาชิกอยู่) และจะไม่มีแถบเมนูทางด้านซ้ายมือ

เมื่อสมาชิกต้องการฟังรายงานสดของ จส.100 ผ่านทางอินเทอร์เน็ตก็จะ ได้ผลลัพธ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home >> Intelligent Bangkok Traffic Router Sun Mar 14 02:15:57 ICT 2004

Welcome to Bangkok

Powered By : KMITL CE

[Register](#) | [Log out](#) | [Map](#) | [Routing](#) | [Traffic Report](#) | [Live Radio Report](#)

PowerBy : KMITL CE

Edit your information

Map

Routing

Traffic Report

Live Radio

Webboard

Logout

Live Radio Report



รูปที่ 5-11 แสดงผลเมื่อเรียก Live Radio Report member_radio.jsp

ต่างจากผู้ที่ไม่ผ่านการล็อกอินเข้ามาตรงที่จะ ไม่มีแถบเมนูทางด้านซ้ายมือ

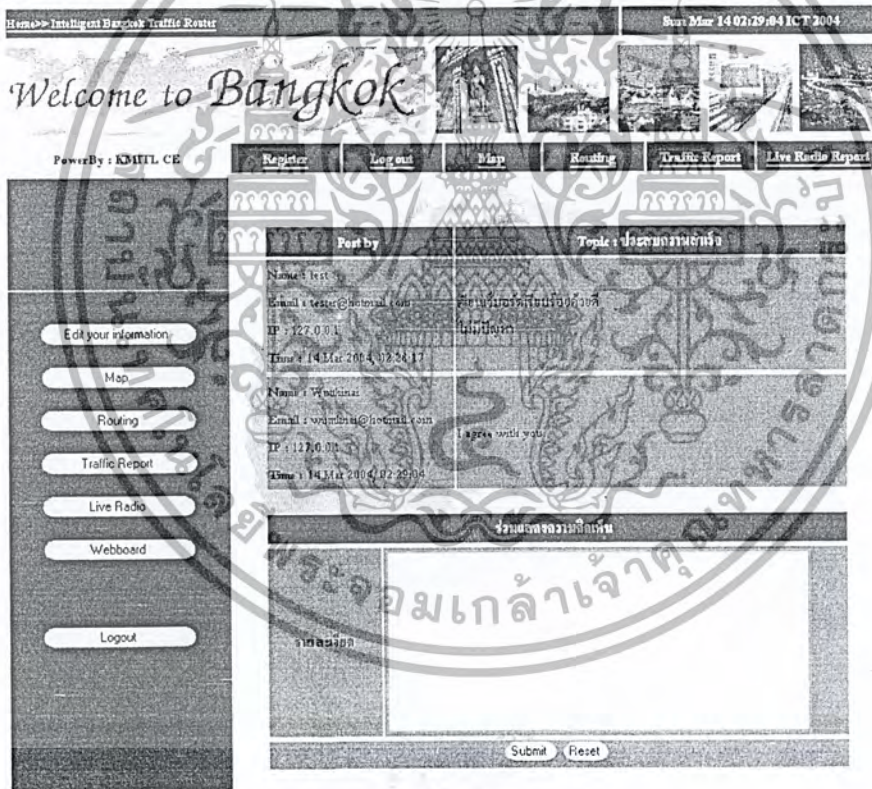
เมื่อสมาชิกต้องการจะแลกเปลี่ยนข่าวสารกับสมาชิกคนอื่นๆของระบบ ก็จะสามารถเข้าไปยังเว็บบอร์ดกระดานสนทนาซึ่งมีหน้าตาดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-13 แสดงผลเมื่อเรียกต้องการสร้างกระทู้ใหม่! `addnew.jsp`

และเมื่อสมาชิกต้องการจะตอบกระทู้หรือดูข้อความรายละเอียดของกระทู้ ก็จะได้แสดงออกมาได้ดังนี้



รูปที่ 5-14 แสดงผลเมื่อเรียกดูกระทู้ `view.jsp`

สำหรับการตอบและสร้างกระทู้นั้นจะสามารถพิมพ์ข้อความใดๆเข้ามาได้ทั้งสิ้น และจะทำการแสดงผลออกมาได้อย่างเหมือนกับที่ป้อนเข้าไป ไม่ว่าข้อความนั้นๆจะเป็น HTML tag ก็ตาม จะมีสิ่งเดียวที่ต่างกันก็คือ การเว้นบรรทัดของข้อความ หากสิ่งที่ป้อนเข้ามาไม่มีการขึ้นบรรทัดใหม่เป็นระยะยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับต่อไปคือการแสดงรูปภาพแผนที่เมื่อผู้ใช้งานต้องการที่จะทราบเส้นทางการเดินทาง และจะแสดงพร้อมทั้งสภาพเส้นทางซึ่งจะแสดงออกมาเมื่อนำเมาส์ไปวางไว้บนตำแหน่งที่เป็นจุดสีดำในรูป (จุดสีดำคือจุดที่เป็นโหนด)



รูปที่ 5-15 แสดงผลการวาดเส้นทางที่พร้อมกับแสดงรายละเอียดเส้นทาง member_map.jsp

โดยที่การแสดงผลสภาพเส้นทางและรายละเอียดต่าง ๆ นั้นจะแสดงออกมาทางแถบสถานะของ Windows Internet Explorer เมื่อนำเมาส์ไปวางบนตำแหน่งที่มีข้อมูลอยู่ในฐานข้อมูลเท่านั้น

5.2 ผลการทดลองแอปพลิเคชันบนโทรศัพท์เคลื่อนที่

การทดลองจะเป็นไปในลักษณะของกิจกรรมที่กระทำระหว่างผู้ใช้งานกับระบบ คือ การเข้าสู่ระบบ, การค้นหาเส้นทาง, การรายงานสภาพการจราจร และการแก้ไขข้อมูลสมาชิก เมื่อผู้ใช้งานแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TrafficReport โปรแกรมจะเรียกหน้าผู้ใช้ขึ้นมาให้เลือกประเภทของผู้ใช้บริการ ถ้าผู้ใช้เลือกในประเภท “member” จะต้องมีชื่อสมาชิกและรหัสผ่าน (ผู้ใช้เป็นสมาชิกของระบบ)



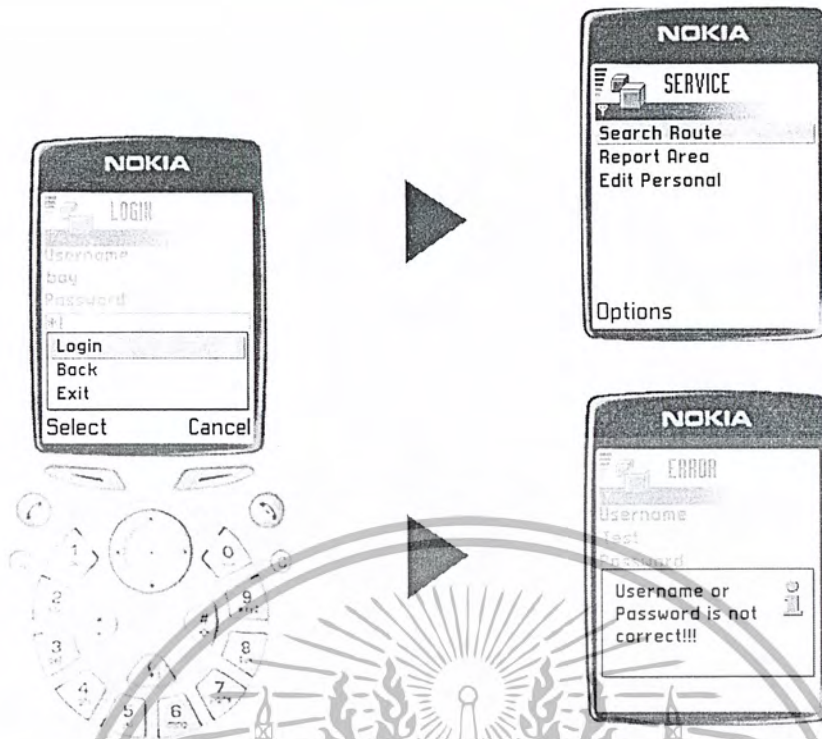
5.2.1 ผลการทดลองการเข้าสู่ระบบ

การเข้าสู่ระบบของสมาชิกและผู้ใช้งานทั่วไปมีความแตกต่างกันใน 1 บริการ คือ บริการการแก้ไขข้อมูลสมาชิก

- เข้าสู่ระบบสมาชิก

สมาชิกจะระบุชื่อสมาชิกลงในฟิลด์ Username และรหัสผ่านลงในฟิลด์ Password จากนั้นกดปุ่ม Login โปรแกรมจะทำการส่งข้อมูลเหล่านั้นไปที่ฝั่งเซิร์ฟเวอร์ตรวจสอบกับฐานข้อมูล ถ้าการตรวจสอบสำเร็จ โปรแกรมจะเรียกหน้าบริการสำหรับสมาชิกออกแสดงผลทางจอโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



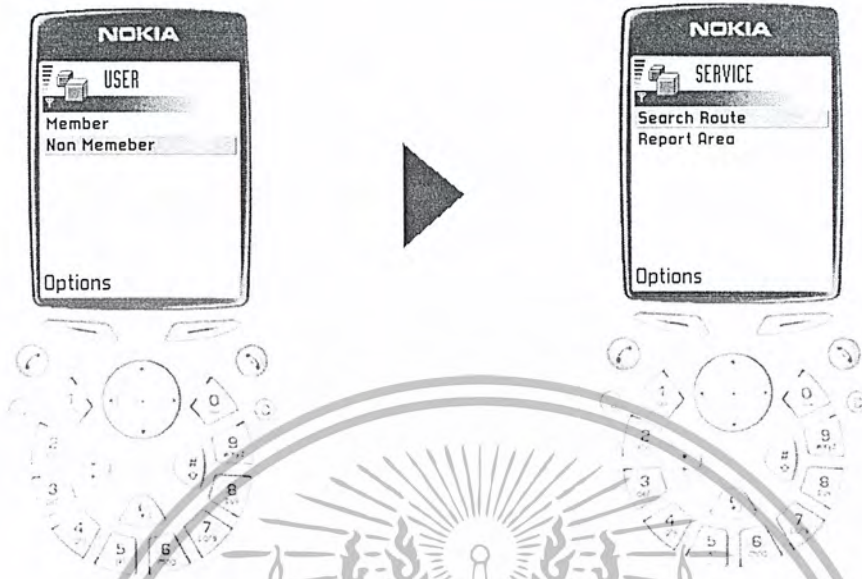
รูปที่ 5-17 แสดงหน้าขั้นตอน การเข้าสู่ระบบของสมาชิก และข้อผิดพลาดที่เกิดขึ้น

ถ้าหากข้อมูลไม่ตรงกัน โปรแกรมจะเรียกหน้าแสดงข้อความผิดพลาดเนื่องจากการตรวจสอบขึ้นมา ซึ่งหน้าแสดงข้อความผิดพลาดจะถูกกำหนดให้แสดงไว้เป็นระยะเวลา 5 วินาทีก่อนที่จะกลับไปยังหน้าเข้าสู่ระบบอีกครั้ง

- เข้าสู่ระบบผู้ใช้ทั่วไป

สำหรับผู้ใช้งานทั่วไป ไม่ได้เป็นสมาชิกของระบบ จะสามารถเข้าใช้บริการได้โดยไม่ต้องมีการพิสูจน์ แต่จะมีแค่บริการค้นหาเส้นทางและบริการรายงานสภาพการจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



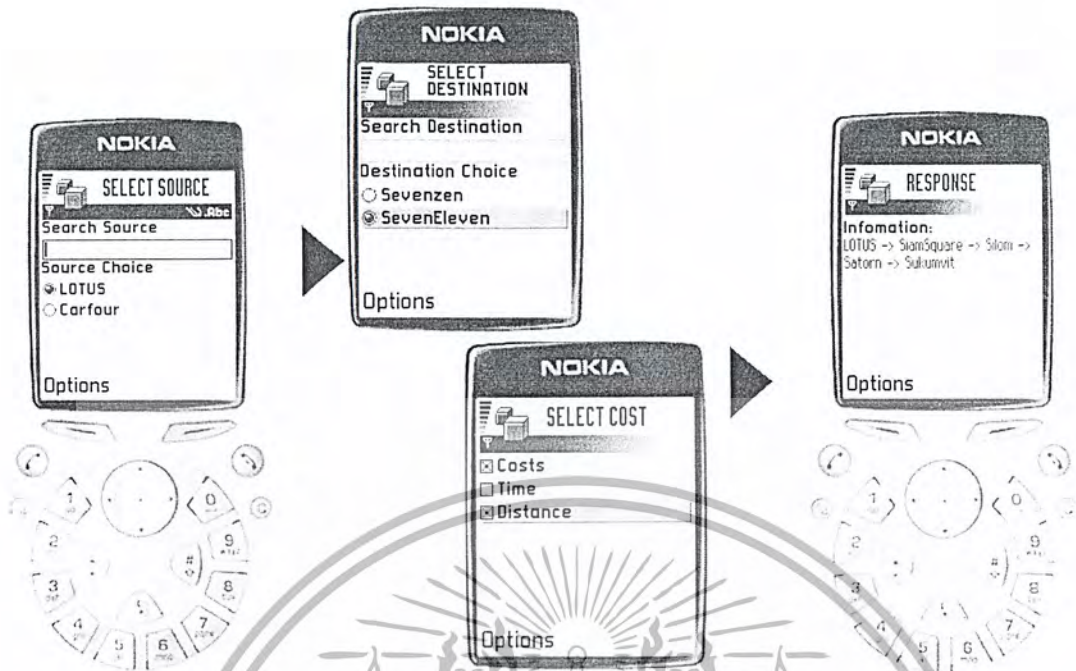
รูปที่ 5-18 แสดงหน้าขั้นตอน การเข้าสู่ระบบของผู้ใช้บริการทั่วไป

5.2.2 ผลการทดลองบริการค้นหาเส้นทาง

- บริการค้นหาเส้นทางสำหรับสมาชิก

บริการค้นหาเส้นทางของสมาชิกจะมีข้อมูลรายการให้อยู่ก่อนแล้ว เนื่องจากเมื่อครั้งที่ผู้ใช้บริการได้มีความยินดีที่จะสมัครเป็นสมาชิกผ่านเว็บไซต์ ระบบจะให้กรอกข้อมูลต่างๆ เกี่ยวกับตัวผู้ใช้ รวมถึงข้อมูลเส้นทาง 2 รายการ และข้อมูลปลายทางอีก 2 รายการ รวมถึงข้อมูลเขตรายงานด้วย แต่ถ้าสมาชิกมีความต้องการค้นหาเส้นทางและปลายทางใหม่ก็สามารถค้นหารายการได้ ตัวอย่างเช่น ในหน้าเลือกค้นหา สมาชิกสามารถเลือกห้างสรรพสินค้าโลตัส (LOTUS) ซึ่งมีอยู่ในรายการแล้วได้ทันที จากนั้นโปรแกรมจะเข้าสู่หน้าเลือกปลายทาง แต่สมาชิกไม่ต้องการปลายทางที่มีอยู่แล้วจึงค้นหาใหม่ โดยต้องการคำว่าเซเว่นอีเลเว่น (SevenEleven) จึงระบุคีย์เวิร์ดว่า “Seven” จากนั้นกดปุ่ม Search โปรแกรมจะทำการส่งข้อมูลเหล่านั้นไปที่ฝั่งเซิร์ฟเวอร์ค้นหารายการก็ฐานข้อมูล เจอคำว่า “SevenEleven” และ “Sevenzen” จากนั้นจึงส่งรายการกลับมา โปรแกรมจะเรียกหน้ารายการค้นหาที่นั้นออกแสดงผลทางจอโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-19 แสดงหน้าจอขั้นตอนของบริการค้นหาเส้นทางของสมาชิก

หลังจากที่เลือกต้นทางและปลายทางแล้ว โปรแกรมจะเรียกหนึ่งปัจจัยในการค้นหาเส้นทางออกมาให้ผู้ใช้งานเลือก ซึ่งสามารถเลือกได้มากกว่า 1 ประเภท ตัวอย่างเช่น สมาชิกเลือก ปัจจัยค่าใช้จ่าย และระยะทาง เมื่อกดปุ่ม Route โปรแกรมจะส่งข้อมูลค้นหาเส้นทางที่มีความหมายว่า "ค้นหาเส้นทางจากโลตัสไปยังเซเวนอีเลเวน โดยคำนวณจากระยะทางที่สั้นที่สุดและประหยัดค่าใช้จ่ายที่สุด"

เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จแล้วจึงส่งข้อมูลกลับมา โปรแกรมจะเรียกหน้าจอแสดงผลลัพธ์ ออกมา ซึ่งเป็นเส้นทางที่ดีที่สุด โดยระหว่างต้นทางและปลายทางจะผ่าน โหนดหรือแยกต่างๆ

- บริการค้นหาเส้นทางสำหรับผู้ทั่วไป

บริการค้นหาเส้นทางสำหรับผู้ทั่วไปจะมีสิ่งที่แตกต่างกับบริการแบบสมาชิก คือ ผู้ใช้งานจะต้องระบุ 키워ดที่ใช้ในการค้นหาเองทุกครั้ง เพราะไม่มีข้อมูลต้นทางและปลายทางแบบบริการของสมาชิก ตัวอย่างเช่น จะค้นหาต้นทางชื่อ โลตัส ต้องระบุ 키워ดเป็นคำว่า "LOT" และกดปุ่มค้นหา จากนั้นค้นหาปลายทางชื่อ เซเวนอีเลเวน จะต้องระบุ 키워ดเป็นคำว่า "Seven" และกดปุ่มค้นหา แต่ขั้นตอนการเลือกปัจจัยค้นหาเส้นทางและกดปุ่มค้นหาเส้นทางจะเหมือนกันกับบริการของสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 ผลการทดลองบริการรายงานสภาพการจราจร

- บริการรายงานสภาพการจราจรของสมาชิก

บริการรายงานสภาพการจราจรของสมาชิกจะมีข้อมูลรายการให้อยู่ก่อนแล้วเช่นเดียวกับบริการการค้นหาเส้นทางจราจร สมาชิกมีความต้องการเขตรายงานสภาพการจราจรใหม่ก็สามารถค้นหารายการได้ ตัวอย่างเช่น ในหน้าเลือกเขตรายงาน สมาชิกสามารถเลือกเขตบางกะปิ (Bang_Kapi) ซึ่งมีอยู่ในรายการแล้วได้ทันที แต่สมาชิก ต้องการทราบพื้นที่รายงานบนถนนพระรามที่ 2 ในเขตบางบอน จึงระบุคีย์เวิร์ดเป็นตัวอักษรว่า 'B' จากนั้นจะมีการส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อขอข้อมูลรายการเขตรายงานซึ่งมีอยู่หลายเขต เช่น บางบอน, บางกะปิ, บางแค และบางเขน เป็นต้น เมื่อสมาชิกเลือก เขตรายงานแล้ว โปรแกรมจะมีการส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อขอข้อมูลรายการพื้นที่ในเขตรายงาน ได้แก่พื้นที่บริเวณอ่อนนุชและพระรามที่ 2 และส่งผลลัพธ์กลับมา



รูปที่ 5-20 แสดงหน้าขั้นตอนบริการรายงานสภาพการจราจรของสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

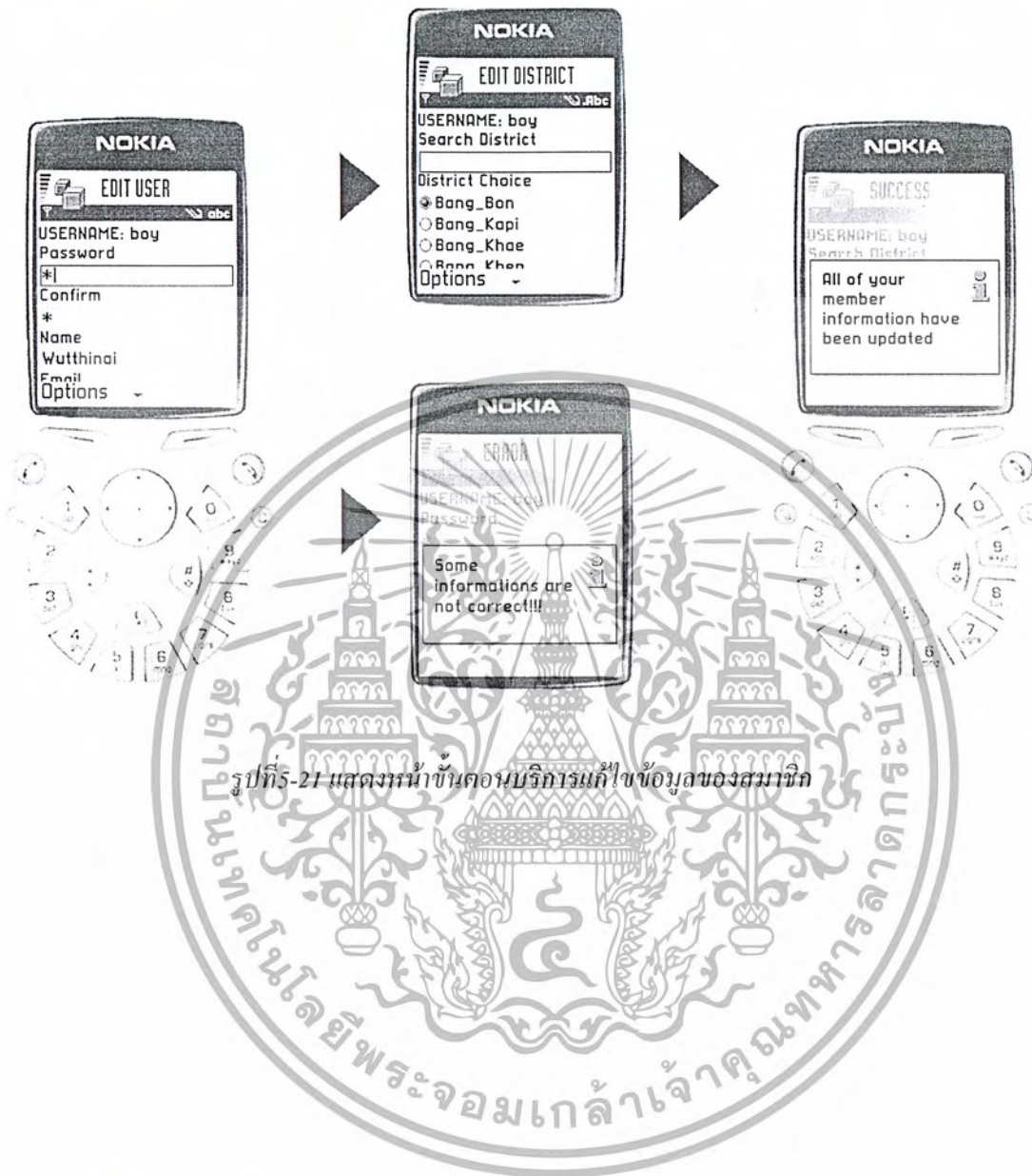
โปรแกรมจะเข้าสู่หน้าเลือกพื้นที่ในเขตรายงานนั้นๆ ซึ่งสมาชิกสามารถเลือกพื้นที่แล้วรายงานสภาพจราจรได้
ครั้งละ 1 พื้นที่ เช่น เลือกพื้นที่ถนนพระรามที่ 2 กดปุ่ม Report โปรแกรมจะทำการส่งข้อมูลเหล่านั้นไปที่ฝั่ง
เซิร์ฟเวอร์ค้นหาข้อมูลพื้นฐานข้อมูล พบว่ามีกรรายงานบนถนนพระรามที่ 2 จากนั้นจึงส่งรายละเอียดสภาพ
จราจรบนพื้นที่นั้นกลับมา โปรแกรมจะเรียกหน้าแสดงผลพร้อมออกทางจอโทรศัพท์

• บริการรายงานสภาพการจราจรของผู้ใช้ทั่วไป

บริการรายงานสภาพการจราจรสำหรับผู้ทั่วไปจะมีสิ่งที่แตกต่างกับบริการแบบสมาชิก คือ
ผู้ใช้งานจะต้องระบุคีย์เวิร์ดที่ใช้ในการค้นหาเองทุกครั้ง เช่นเดียวกับการค้นหาเส้นทาง เพราะไม่มีข้อมูลเขต
รายงานแบบบริการของสมาชิก ตัวอย่างเช่น จะค้นหาเขตบางบอน ต้องระบุคีย์เวิร์ดเป็นตัวอักษรว่า ‘B’ และ
กดปุ่มค้นหา ถ้าหากจะค้นหาเขตอื่นชื่อชุดคิด จะต้องระบุคีย์เวิร์ดเป็นคำว่า ‘Du’ และกดปุ่มค้นหา แต่ขั้นตอน
การเลือกพื้นที่รายงานในเขตและกดปุ่มรายงานจะเหมือนกันกับบริการของสมาชิก

5.2.4 ผลการทดสอบบริการแก้ไขข้อมูลสมาชิก

บริการแก้ไขข้อมูลมีเฉพาะสมาชิกเท่านั้น เมื่อสมาชิกเลือกใช้บริการ โปรแกรมจะเรียกหน้าแก้ไข
ข้อมูลทั่วไปขึ้นมา สมาชิกจะสามารถแก้ไขข้อมูลส่วนตัว เช่น รหัสผ่าน, ชื่อจริง, ที่อยู่ และเบอร์โทรศัพท์ เป็น
ต้น แต่จะไม่สามารถแก้ไขชื่อสมาชิกของระบบได้ จากนั้นจะเข้าสู่หน้าแก้ไขข้อมูลเขต ซึ่งสมาชิกสามารถ
แก้ไขข้อมูล โดยการค้นหาชื่อเขตรายงานใหม่ได้ ตัวอย่างเช่น ผู้ใช้ชื่อสมาชิกว่า ‘boy’ แก้ไขรหัสผ่านจาก
เดิมคือ ‘boy’ กลายเป็น ‘w’ และเปลี่ยนอีเมลจาก Wuthinai@hotmail.com เป็น boy@hotmail.com แทน โดย
ลงค่าข้อมูลอื่นๆ ไว้ และก่อนกดปุ่ม Next เพื่อข้ามไปทำหน้าถัดไป สมาชิกจะต้องยืนยันรหัสผ่านที่เปลี่ยน
ใหม่ด้วย ไม่งั้นนั้นจะเกิดข้อผิดพลาดขึ้น จากนั้นสมาชิก ‘boy’ จะแก้ไขข้อมูลเขตโดยเปลี่ยนจากบางกะปิ
เป็นบางแค จึงระบุคีย์เวิร์ดเป็นตัวอักษรว่า ‘B’ โปรแกรมจะติดต่อร้องขอข้อมูลรายการที่มีค่าตรงกับคีย์เวิร์ด
และได้ผลลัพธ์เป็นรายการเขต เช่น บางบอน, บางแค, บางกะปิ เป็นต้น จากนั้นส่งผลลัพธ์กลับมา เมื่อสมาชิก
เลือกเขตรายงานแล้ว กดปุ่ม Edit ข้อมูลจากโทรศัพท์จะถูกส่งไปอัปเดตที่เครื่องเซิร์ฟเวอร์ และสมาชิกจะใช้
ข้อมูลนั้นตลอดจนมีการแก้ไขใหม่อีกครั้ง



รูปที่ 5-21 แสดงหน้าขั้นตอนบริการแก้ไขข้อมูลของสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และสรุป

6.1 บทวิจารณ์และสรุปส่วนฐานข้อมูล

ข้อมูลในฐานข้อมูลที่มีอยู่นั้น มีข้อมูลของเส้นทางการจราจรภายในเขตกรุงเทพมหานครไม่เพียงพออย่างมาก ซึ่งการที่จะได้ฐานข้อมูลมานั้นจะต้องเสียเงินหรือสามารถติดต่อกับผู้ที่มีข้อมูลด้านนี้โดยตรง ซึ่งทำไม่ได้ เพราะเนื่องจากผู้ที่มีข้อมูลนี้ไม่สามารถให้นำมาทดลองใช้งานได้ เนื่องจากเป็นข้อมูลทางพาณิชย์

ซึ่งข้อมูลที่ต้องการนั้นคือ จะต้องสามารถนำมาเขียนเป็นแผนที่เองได้เลย (เป็นข้อมูลแบบสมบูรณ์) หมายถึงถนนแต่ละเส้นจะต้องมีจุดคู่ลำดับ X และ Y ทุกๆจุดที่มีการเลี้ยว (เมื่อถนนไม่ได้เป็นเส้นตรง) จะต้องมีการบันทึกกำกับถนนอย่างถูกต้อง มีข้อมูลของถนน เช่น จำนวนช่องทางการจราจร ค่าผ่านทาง ระยะทางและทิศทางการเดินทาง เป็นต้น รวมถึงมีการระบุถึงตำแหน่งยูเทิร์นกลับรถของถนนด้วย ซึ่งข้อมูลส่วนนี้ถ้าได้มา จะสามารถนำไปวาดแผนที่ด้วย GIS (Geographic Information System) โดยผลิตภัณฑ์ตัวนี้ได้มีการศึกษาและถูกเลือกที่จะใช้แล้ว ก็คือ JShape นั่นเอง ซึ่งประมวลผลไฟล์ข้อมูลที่เป็น “.shp”

การแก้ไขปัญหาข้อมูลด้านการขาดแคลนฐานข้อมูลนี้ ทำได้เพียงแก้เพื่อให้สามารถนำไปใช้งานได้เท่านั้น และเป็นเพียงบางส่วนของเขตลาดกระบังเท่านั้น คือแก้ไขโดยการ

1. ทำการสำรวจพื้นที่อย่างง่ายด้วยตัวเอง
2. นำภาพแผนที่ที่มีอยู่แล้วในอินเทอร์เน็ตมาทำการแก้ไขปรับปรุงเพื่อให้สามารถนำไปใช้งานได้
3. นำแผนที่จริงของเขตกรุงเทพมหานครที่มีความละเอียดของข้อมูลมากกว่ามาทำการเปรียบเทียบกับรูปภาพที่ได้มา
4. เก็บข้อมูลรูปภาพในบริเวณที่เป็นโหนดของรูปนั้นๆด้วยคู่อันดับ X และ Y ด้วยการวัด

ผลที่ได้จากการแก้ไขก็คือ สามารถจะนำข้อมูลมาทำการเพิ่มจุดสีดำในบริเวณที่เป็นโหนดลงบนแผนที่ที่ได้ แต่ไม่สามารถที่จะวาดเส้นถนนทั้งเส้นที่ต้องการได้ เช่น ถนนที่มีความคับคั่งในการสัญจรมากควรจะมีสีเป็นสีแดง แต่ไม่สามารถทำได้ และแผนที่ที่ได้มาไม่มีกอร์นขนาดของถนนหรือจำนวนช่องทางการเดินทาง รวมทั้งไม่มีทิศทางการเดินทางด้วย

และปัญหาอีกอย่างก็คือ ระบบค้นหาเส้นทางอัจฉริยะ(ส่วนแสดงผล) จะใช้ฐานข้อมูลเดียวกันกับของระบบค้นหาเส้นทางอัจฉริยะ(ส่วนประมวลผล) ซึ่งข้อมูลที่ใช้ร่วมกันไม่สามารถนำมาใช้ประโยชน์ทั้ง 2 ฝ่าย (ข้อมูลที่ใช้ไม่เหมือนกัน) ซึ่งก็คือ ส่วนแสดงผลต้องการความทับซ้อนของการเดินทางของถนนทั้งสาย แต่ส่วนประมวลผลจะใช้เวลาทับซ้อนระหว่างโหนด แต่ส่วนแสดงผลก็สามารถนำข้อมูลนี้ไปแสดงได้ ซึ่งจะได้การแสดงผลที่ดูค่อนข้างจะแปลกตา

6.2 บทวิจารณ์และสรุปส่วนเว็บไซต์

ปัญหาที่เกิดขึ้นโดยตรงกับเว็บไซต์โดยตรงนั้นไม่มี จะมีก็เพียงแค่ส่วนที่ขาดข้อมูลในการวาดเส้นทางเท่านั้น

ปัญหาอีกอย่างก็คือ ตัวกระผมผู้พัฒนาเองต้องการที่จะรวมส่วนแสดงผลและส่วนประมวลของหน้าเว็บเพจเข้าด้วยกันเป็นไฟล์เดียว ทำให้ชุดคำสั่งโค้ดค่อนข้างจะมีความยุ่งยากซับซ้อน แต่ก็จะได้ประโยชน์คือการแก้ไขสามารถที่จะกระทำได้ในจุดเดียว

และสุดท้ายคือเนื่องจากระบบค้นหาเส้นทางอัจฉริยะ(ส่วนแสดงผลและประมวลผล) จะต้องทำการติดต่อสื่อสารกันได้ในระดับหนึ่ง ซึ่งผู้ที่พัฒนามาของทั้ง 2 ฝ่ายนั้นเป็นคนละฝ่าย ไม่ได้พัฒนาเพื่อนำมาใช้งานร่วมกันอย่างจริงจัง ทำให้เกิดปัญหาในการสื่อสารบ้าง คือ ค่อนข้างจะยุ่งยากเท่านั้น

6.3 บทวิจารณ์และสรุปส่วนแอปพลิเคชันบนโทรศัพท์เคลื่อนที่

ปัญหาส่วนใหญ่ที่เกิดขึ้นบนโทรศัพท์เคลื่อนที่ คือ ทรัพยากรของระบบที่มีขนาดจำกัด เช่น หน่วยความจำ หน่วยประมวล และขนาดหน้าจอแสดงผล เป็นต้น ประกอบกับคอมพิวเตอร์ต่างๆ ของ j2me ที่ส่วนใหญ่จะเป็นคอมพิวเตอร์อย่างง่ายและไม่ซับซ้อน จึงทำให้การทำงานของแอปพลิเคชันไม่มีประสิทธิภาพเท่าที่ควร แต่อย่างไรก็ตามบทบาทการใช้งานแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เริ่มมีมากขึ้นในอนาคตอาจจะมีการพัฒนาทรัพยากรต่างๆ ของระบบให้มีประสิทธิภาพสูงขึ้น

ปัญหาอีกอย่าง คือ การนำไปใช้งานจริงซึ่งในบางครั้งแอปพลิเคชันขาดความมีเสถียรภาพใช้งานในการติดต่อกับเครื่องเซิร์ฟเวอร์ ทั้งนี้อาจจะขึ้นอยู่กับระบบของผู้ให้บริการ โทรศัพท์ หรือเป็นเพราะการทำงานของแอปพลิเคชันก็เป็นได้

ภาคผนวก ก

อักษรย่อ

ASP	Active Server Page
CDC	Connect Device Configuration
CGI	Common Gateway Interface
CLDC	Connected, Limited Device Configuration
HTML	HyperText Markup Language
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JSP	Java Server Page
JVM	Java Virtual Machine
KVM	K Virtual Machine
MIDP	Mobile Information Device Profile
PDA	Personal Digital (Data) Assistant
PHP	PHP Hypertext Preprocessor
SMS	Short messaging service
WAP	Wireless Application Protocol



ภาคผนวก ข

ซอร์ซโค้ดของ applet (Map.jsp)

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import java.util.*;
import java.io.*;
import java.sql.*;
import java.text.*;

public class Map1 extends Applet implements MouseListener,MouseMotionListener
{
    private Image i;
    private int x,y,clickx,clicky,picx,picy,maxpicx,maxpicy,minpicx,minpicy;
    private int count = 0;
    private int x_coord[] = new int[100];
    private int y_coord[] = new int[100];
    String temp = "",getdistrict = "";

    public int CountXY(String image) throws ClassNotFoundException,SQLException
    {
        int count_xy = 0;

        try
        {
            Class.forName("org.gjt.mm.mysql.Driver");
            Connection mycon =
DriverManager.getConnection("jdbc:mysql://localhost/myproject?user=root&password=231025231025");
            Statement stmt = mycon.createStatement();

```

```

        ResultSet rs = stmt.executeQuery("select count(*) as num from mapinfo where
image = '"+image+"' ");

        while (rs.next())
        {

            count_xy = rs.getInt("num");

        }

        rs.close();
        stmt.close();
        mycon.close();

    }
    catch(Exception ex)
    {
        count_xy = 0;
        return count_xy;
    }
}

public int[] getX(String image) throws ClassNotFoundException,SQLException
{
    int x[];
    x = new int[100];
    int i = 0;

    try
    {

        Class.forName("org.gjt.mm.mysql.Driver");

        Connection mycon =
        DriverManager.getConnection("jdbc:mysql://localhost/myproject?user=root&password=231025231025");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Statement stmt = mycon.createStatement();

ResultSet rs = stmt.executeQuery("select x_coord from mapinfo where image =
"+image+" ");

while (rs.next())
{
    x[i] = rs.getInt("x_coord") - 5;
    i++;
}

rs.close();
stmt.close();
mycon.close();
}
catch(Exception ex)
return x;
}

public int[] getY(String image) throws ClassNotFoundException,SQLException
{
    int y[];
    y = new int[100];
    int i = 0;

    try
    {
        Class.forName("org.gjt.mm.mysql.Driver");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Connection mycon =
DriverManager.getConnection("jdbc:mysql://localhost/myproject?user=root&password=231025231025");

        Statement stmt = mycon.createStatement();

        ResultSet rs = stmt.executeQuery("select y_coord from mapinfo where image =
"+image+" ");

        while (rs.next())
        {

            y[i] = rs.getInt("y_coord") - 5;
            i++;
        }

        rs.close();
        stmt.close();
        mycon.close();
    } catch (Exception ex)
    {
        return y;
    }

    public String shownode (String image,int x,int y) throws ClassNotFoundException,SQLException
    {

        String node = "";

        try
        {

            Class.forName("org.gjt.mm.mysql.Driver");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Connection mycon =
DriverManager.getConnection("jdbc:mysql://localhost/myproject?user=root&password=231025231025");
        Statement stmt = mycon.createStatement();

        ResultSet rs = stmt.executeQuery("select node from mapinfo where image =
""+image+" && x_coord > +(x-5)+" && x_coord < +(x+5)+" && y_coord > +(y-5)+" && y_coord <
""+(y+5)""");

        while (rs.next())
        {
                node = rs.getString("node");
        }

        rs.close();
        stmt.close();
        mycon.close();
    } catch (Exception ex)
    {
        return node;
    }
}

public String getPic(String district) throws ClassNotFoundException, SQLException
{
    try
    {
        Class.forName("org.gjt.mm.mysql.Driver");
        Connection mycon =
DriverManager.getConnection("jdbc:mysql://localhost/myproject?user=root&password=231025231025");
        Statement stmt = mycon.createStatement();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ResultSet rs = stmt.executeQuery("select image from mapinfo where
default_district = '"+district+"' ");

        while (rs.next())
        {

            temp = rs.getString("image");

        }

        rs.close();
        stmt.close();
        mycon.close();

    }
    catch (Exception ex)
    {

    }

    if (temp.equals(""))
    {
        temp = "e10047000n1346000s4.jpg";
    }

    return temp;

}

public void init()
{

    //picx = 10047000;
    //picy = 1346000;
    String temp = "";
    getdistrict = getParameter("map_district");
    try

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        temp = getPic(getdistrict);
        picx = Integer.parseInt(temp.substring(1,9));
        picy = Integer.parseInt(temp.substring(10,17));
    }
catch(Exception ex)
    {
        picx = 10047000;
        picy = 1346000;
    }
maxpicx = 10047000;
maxpicy = 1346000;
minpicx = 10044600;
minpicy = 1343600;
i = getImage(getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
addMouseListener(this);
addMouseMotionListener(this);
}

public void paint (Graphics g)
{
    try
    {
        count = CountXY("e"+picx+"n"+picy+"s4");
        x_coord = getX("e"+picx+"n"+picy+"s4");
        y_coord = getY("e"+picx+"n"+picy+"s4");
    }
    catch (Exception e)
    {
        count = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    g.drawImage(i,0,0,this);
    //g.drawString(temp,10,10);
    //g.drawString("e"+picx+"n"+picy+"s4",10,50);
    //String temp1 = Integer.toString(count);
    //g.drawString(temp1,10,10);
    //String temp2 = Integer.toString(x[0]);
    //g.drawString(temp2,10,70);
    //String temp3 = Integer.toString(y[0]);
    //g.drawString(temp3,10,70);

    for (int i=0;i<count;i++)
    {
        g.fillOval(x_coord[i],y_coord[i],10,10);
    }
}

public void mousePressed(MouseEvent e) {}
public void mouseReleased(MouseEvent e) {}
public void mouseExited(MouseEvent e)
{
    showStatus("Pointer outside applet");
}

public void mouseEntered(MouseEvent e) {}
public void mouseClicked(MouseEvent e)
{
    clickx = e.getX();
    clicky = e.getY();
    showStatus("clicked at "+clickx+","+clicky);
    //move left
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((clickx < 20) && (clicky>290) && (clicky<310))
{
    if (picx == minpicx)
    {
        showStatus("Can't move left");
    }
    else
    {
        showStatus("click move left");
        MoveLeft();
    }
}
//move right
if((clickx >580) && (clicky>290) && (clicky<310))
{
    if (picx == maxpicx)
        showStatus("Can't move right");
    else
        showStatus("click move right");
        MoveRight();
}
//move up
if((x < 310) && (x>290) && (y<20))
{
    if (picy == maxpicy)
    {
        showStatus("Can't move up");
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
    {
        showStatus("click move up");
        MoveUp();
    }
}
//move down
if((x < 310) && (x>290) && (y>580))
{
    if (picy == minpicy)
    {
        showStatus("Can't move down");
    }
    else
    {
        showStatus("click move down");
        MoveDown();
    }
}
//move up left
if((x <20) && (y<20))
{
    if ((picx == minpicx) || (picy == maxpicy))
    {
        showStatus("Can't move up left");
    }
    else
    {
        showStatus("click move up left");
        MoveUpLeft();
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

}

//move down left
if ((x < 20) && (y>580))
{
    if ((picx == minpicx) || (picy == minpicy))
    {
        showStatus("Can't move down left");
    }
    else
    {
        showStatus("click move down left");
        MoveDownLeft();
    }
}

//move up right
if ((x > 580) && (y<20))
{
    if ((picx == maxpicx) || (picy == maxpicy))
    {
        showStatus("Can't move up right");
    }
    else
    {
        showStatus("click move up right");
        MoveUpRight();
    }
}

}

//move down right
if ((x > 580) && (y>580))
{

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if ((picx == maxpicx) || (picy == minpicy))
        {
            showStatus("Can't move down left");
        }
    else
    {
        showStatus("click move down right");
        MoveDownRight();
    }
}

public void mouseDragged(MouseEvent e) {}
public void mouseMoved(MouseEvent e)
{
    x = e.getX();
    y = e.getY();
    showStatus(x+" "+y);
    try
    {
        showStatus(shownode("e"+picx+"n"+picy+"s4",x,y));
    }
    catch(Exception ex)
    {
    }
}

if((x < 20) && (y>290) && (y<310))
{
    showStatus("Move left");
}

if((x > 580) && (y>290) && (y<310))
{

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        showStatus("Move right");
    }
    if((x < 310) && (x>290) && (y<20))
    {
        showStatus("Move up");
    }
    if((x < 310) && (x>290) && (y>580))
    {
        showStatus("Move down");
    }
    if((x < 20) && (y<20))
    {
        showStatus("Move up-left");
    }
    if((x < 20) && (y>580))
    {
        showStatus("Move down-left");
    }
    if((x > 580) && (y<20))
    {
        showStatus("Move up-right");
    }
    if((x > 580) && (y>580))
    {
        showStatus("Move down-right");
    }
}

public void MoveLeft()
{
    picx = picx - 2400;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveRight()
    {
        picx = picx + 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveUp()
    {
        picy = picy + 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveDown()
    {
        picy = picy - 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveUpLeft()
    {
        picx = picx - 2400;
        picy = picy + 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveDownLeft()
    {
        picx = picx - 2400;

```



```

        picy = picy - 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveUpRight()
    {
        picx = picx + 2400;
        picy = picy + 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
    public void MoveDownRight()
    {
        picx = picx + 2400;
        picy = picy - 2400;
        i = getImage (getCodeBase(),"pic/e"+picx+"n"+picy+"s4.jpg");
        repaint();
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Forum Nokia : “A Network MIDlet Example:Fruit Machine”, Version 0.9.2, Date : 17 December 2001
- [2] John W.Muchow : “core J2ME Technology & MIDP”, Sun Microsystems Press Java Series
- [3] Yu Feng, Jun Zhu แปลและเรียบเรียงโดย จันทรมาศ ศาณะเสน พันธุ์คำ : “J2ME Wireless Java Programming with Java2 Micro Edition”, Corporation Limited, 2545
- [4] Deitel™ : “Java HowTo Program”, Fourth Edition, Prentice Hall
- [5] ดร.วีระศักดิ์ ชิงฉาวร : “Fundamental of Java Programming Volume 2”, บริษัทซีเอ็ดจำกัด มหาชน, สงวนลิขสิทธิ์ พ.ศ. 2543
- [6] ทรงเกียรติ ภาวดี : “เก่ง J2ME ให้ครบสูตร”, Witty Group
- [7] กิตติ ภัคดีวัฒนสกุล : “ถ่มถึร์ Java เล่ม 1”, พิมพ์ครั้งที่ 2 ฉบับแก้ไขปรับปรุงเพิ่มเติม, KTP Comp & Consult
- [8] กวิณัฐ มาศสุวรรณ : “JAVA 2 MICRO EDITION J2ME คู่มือสำหรับเริ่มต้น พัฒนาจากฉบับมือถือ”, บริษัท เอ.อาร์. อินฟอรมชัน แอนคด์, พิมพ์เลขัน จำกัด, 2537
- [9] สลาติต ชัยวิวัฒน์ตระกูล : “เก่ง JSP ให้ครบสูตร”, Witty Group, พิมพ์ครั้งที่ 1 ธันวาคม 2545

