สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บ

A WEB-BASED MULTI-AIRLINES CUSTOMER SERVICE SYSTEM DEVELOPMENT USING ORACLE 10G

นายภักดี ยาวิชัย

นายอริยะ ชมะวิต

เลขหมู่.....
เลขทะเบียน 62376
วัน,เดือน,ปี 16 ส.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

การออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บ

A WEB-BASED MULTI-AIRLINES CUSTOMER SERVICE SYSTEM DEVELOPMENT USING ORACLE 10G

โดย

นายภักดี ยาวิชัย

นายอริยะ ชมะวิต

อาจารย์ที่ปรึกษา

รศ.ดร. ศุภมิตร จิตตะยโศธร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บ

A WEB-BASED MULTI-AIRLINES CUSTOMER SERVICE SYSTEM DEVELOPMENT USING ORACLE 10G

ผู้จัดทำ

1. นายภักดี ยาวิชัย รหัสนักศึกษา 45010571

2. นายอริยะ ชมะวิต รหัสนักศึกษา 45010946

อาจารย์ที่ปรึกษา

(รศ.ดร. ศุภมิตร จิตตะยโศธร)

การออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บ

นายภักดี ยาวิชัย 45010571
นายอริยะ ชมะวิต 45010946
รศ.ดร. ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการการออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บแอพพลิเคชั่น โดยอ้างอิงจาก เว็บไซต์ สตาร์แอร์ลิแอนท์(Star Alliance) ซึ่งให้บริการลูกค้าในด้านการค้นหาเส้นทางและตารางเวลาของแต่ละสายการบินทั่วโลก โดยในเว็บแอพพลิเคชันนี้จะสามารถที่จะทำการค้นหาเส้นทางของการเดินทาง ทำการจองตั๋วการเดินทางและอื่นๆ โดยลักษณะเด่นของวิธีการที่นำเสนอในวิทยานิพนธ์นี้คือจะเน้นขั้นตอนของการพัฒนาที่สามารถนำมาบำรุงรักษาหรือพัฒนาต่อได้ง่ายโดยยึดหลักการพัฒนาตามกระบวนการการพัฒนาเว็บแอพลิเคชันและเว็บเซอร์วิส ซึ่งในวิทยานิพนธ์เล่มนี้ได้แสดงผลของการทำธุรกรรมต่างๆบนเว็บแอพพลิเคชันของการจองตั๋วหลายสายการบินผ่านทางเว็บแอพพลิเคชันอีกด้วย

A WEB-BASED MULTI-AIRLINES CUSTUMER SERVICE SYSTEM DEVELOPMENT USING ORACLE 10G

Phakdee Yavichai 45010571

Ariya Chamavit 45010946

Assoc. Prof. Dr. Suphamit Chittayasothorn Advisor

Academic Year 2004

ABSTRACT

This thesis proposes the design and development of a prototype Web-based Multi-Airlines Customer Service System. The system imitates the Star Alliance Web-site. This prototype have many features such as searching and flight reservation . A salient feature of the project is to design a web-application which is easy to develop and maintain based on web-application and web-service good practices. This thesis presents the result of transaction processing on Web-based Multi-Airlines Customer Service System.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก รศ.ดร. ศุภมิตร จิตตะยโศธร ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกคนที่ให้คำแนะนำต่างๆ โดยเฉพาะ พี่ๆ และเพื่อนที่ห้องวิจัยอีเอสแอล(Embedded System Lab)ทุกคน ที่ให้คำปรึกษาและคอยช่วยเหลือในการใช้ห้องวิจัยและให้กำลังใจในการทำปริญญานิพนธ์ฉบับนี้ตลอดมา

ขอบคุณ google.com และเว็บไซต์ต่างๆ ที่เป็นแหล่งความรู้ที่ดี และทำให้การทำงานชิ้นนี้สะดวกมากยิ่งขึ้น

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

นายภักดี ยาวิชัย

Thank you my Project Advisor, Assoc. Prof. Suphamit Chittayasothorn. I feel so appreciate for your suggestion that I can recognize what your deeply means. You are my impressive.

Thanks Computer Engineering Department. To put me in the way to reach for the star. You give me precious moment.

Thanks my second home, ECC-603 Embedded System Lab. It is always a good day when I step in this room. I will remember that esl36 used to my own.

And the last two persons that I never forget, My parents You are my greatest willpower. I love you all of my heart.

นายอริยะ ชมะวิต

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความสำคัญและที่มาของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....1

1.3 ขอบเขตของโครงงาน.....2

1.4 วิธีการดำเนินการ.....3

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

1.6 ส่วนประกอบของปริญญานิพนธ์.....4

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการวิจัย และระบบโอเอฟดีเอ็ม.....5

2.1 แนวทางการพัฒนาเว็บแอพพลิเคชัน.....5

2.1.1. รูปแบบการพัฒนาเว็บไซต์.....5

2.1.2. สถาปัตยกรรมในการติดตั้งเว็บไซต์.....6

2.1.3.ภาษายูเอ็มแอลกับเว็บ.....8

2.1.4. กระบวนการทางวิศวกรรมซอฟต์แวร์อาร์ยูพี (Rational Unified Process).....19

2.1.5. รูปแบบสถาปัตยกรรมของเว็บแอพพลิเคชัน.....21

2.2. เทคโนโลยีดอทเน็ต (.NET).....22

2.2.1. ภาพรวมของเทคโนโลยีดอทเน็ต (.NET).....22

2.2.2. ประโยชน์ของเทคโนโลยีดอทเน็ต.....22

2.2.3.ผลิตภัณฑ์และเทคโนโลยีของ Microsoft ในปัจจุบันที่มีการใช้ .NET.....23

2.3. เทคโนโลยีเว็บเซอร์วิส (Web Service).....24

2.3.1. ภาพรวมของเทคโนโลยี Web Service.....24

2.3.2. โครงสร้างขั้นพื้นฐานของเทคโนโลยีเว็บเซอร์วิส (Web services).....24

สารบัญ (ต่อ)

2.3.3. ลักษณะการเชื่อมต่อของเทคโนโลยีเว็บเซอร์วิส (Web services).....26

2.3.4. เทคโนโลยีเซอร์วิส (Web services) ในมุมมองทางธุรกิจ.....27

2.3.5. เปรียบเทียบเว็บไซต์กับเอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web service).....28

2.3.6. ตัวอย่างผลิตภัณฑ์ของไมโครซอร์ฟMicrosoft
ที่มีการเรียกใช้เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services).....28

2.4. ไมโครซอร์ฟ ดอทเน็ตเฟรมเวิร์ก (Microsoft .NET Framework).....29

2.4.1. ความหมายของเฟรมเวิร์ก (Framework) และแพลตฟอร์ม (Platform).....29

2.4.2. โครงสร้างของดอทเน็ตเฟรมเวิร์ก (.NET Framework).....30

2.5. เอเอสพีดอทเน็ต (ASP.NET).....32

2.5.1. ภาพรวมของ เอเอสพีดอทเน็ต (ASP.NET).....32

2.5.2. คุณสมบัติ ข้อดีและส่วนที่ปรับปรุงจากเอเอสพี (ASP)
ของเอเอสพี ดอทเน็ต (ASP.NET).....32

2.6. Visual Studio .NET.....33

2.6.1. ภาพรวมของวิชวลสตูดิโอ (Visual Studio .NET).....33

2.6.2. คุณสมบัติใหม่ของวิชวลสตูดิโอ 2003(Visual Studio .NET 2003).....35

2.7. Oracle Developer Tools for Visual Studio .NET.....36

2.7.1. ส่วนประกอบต่างๆ ของ Oracle Developer Tools Visual Studio .NET.....36

2.8. โครงสร้างข้อมูลแบบกราฟ (Graph).....41

2.8.1. ประเภทของกราฟ 42

2.9. การค้นหาข้อมูล (searching).....44

2.9.1. การค้นหาแบบไบล์ด (Blind Search).....44

2.9.2. การค้นหาแบบฮิวริสติก(Heuristic Search).....48

2.10.การบำรุงรักษาซอฟต์แวร์(software Maintenance).....48

2.10.1. ความหมายและภาพรวม.....48

2.10.2. Level of maintenance.....49

2.10.3. รูปแบบของการ Maintenance.....49

2.10.4 ปัจจัยของการ Maintenance ที่จะส่งผลกระทบ.....49

2.11. โพรโตคอล Secure Socket Layer.....50

2.12. ทูเฟสคอมมิท โพรโตคอล Two-Phase Commit Protocol.....50

สารบัญ (ต่อ)

หน้า

บทที่ 3 ขั้นตอนในการออกแบบและสร้างเว็บแอพพลิเคชันการจองตั๋วเครื่องบินหลายสายการบิน

3.1. ขั้นตอนการเก็บความต้องการและยูสเคส.....52

3.2. ขั้นตอนการวิเคราะห์ (Analysis).....55

3.2.1. แผนภาพแสดงลำดับการทำงาน.....55

3.2.2. แผนภาพแสดงการทำงานร่วมกัน.....62

3.3. ขั้นตอนการออกแบบ (Design).....67

3.3.1. แผนภาพคลาส (Class Diagram).....68

3.3.2. การออกแบบเว็บเซอร์วิส.....74

3.3.3. การไหลของระบบหน้าเว็บเพจ.....80

3.3.4. การออกแบบฐานข้อมูลของระบบนำส่งเอกสาร.....80

3.4. ขั้นตอนการสร้าง (Implement).....82

3.4.1. การสร้างในส่วนของไคลเอ็นต์เพจ.....82

3.4.2. การสร้างในส่วนของเซิร์ฟเวอร์เพจ.....85

3.4.3. การสร้างในส่วนของคอมโพแนนท์ที่เทียร์กลาง.....86

บทที่ 4 ผลของการทำงานส่วนต่างๆของเว็บแอพพลิเคชัน

4.1 ส่วนของการทำการจองตั๋วเครื่องบิน.....87

4.2 ส่วนของการแก้ไขหรือยกเลิกตั๋วที่ได้จองไปแล้ว.....94

4.3 ส่วนของการสมัครสมาชิก.....98

4.4 ส่วนของการเข้าสู่ระบบสมาชิก.....100

4.5 ส่วนของการแก้ไขข้อมูลสมาชิก.....102

4.6 ส่วนตรวจสอบประวัติเที่ยวบินที่เคยทำการจองไปแล้ว.....102

4.7 ทำการตรวจสอบสถานะของเที่ยวบินขาเข้า-ออก ในแต่ละสนามบิน.....103

4.8 การอัพเดทที่นั่งและสถานะการบินของสายการบินพันธมิตร.....104

4.9 การจัดการฐานข้อมูลผ่านทางหน้าเว็บเพจ.....105

บทที่ 5 บทวิจารณ์และสรุป

5.1 บทสรุป.....111

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....111

5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....111

สารบัญ (ต่อ)

หน้า

5.4 แนวทางการพัฒนาต่อ.....112

บรรณานุกรม.....113

สารบัญตาราง

ตารางที่ หน้า

2.1. แสดงการเปรียบเทียบการค้นหาระหว่างแบบการค้นหาแนวลึกก่อน
กับการค้นหาแนวกว้างก่อน.....47

3.1. แสดงการอธิบายคำศัพท์ของยูสเคสในเว็บแอพพลิเคชัน.....53

3.2. อธิบายการทำงานแต่ละตารางในฐานข้อมูล.....80

สารบัญรูป

รูปที่ หน้า

2.1. สถาปัตยกรรมของเว็บแบบซิงเกิลเทียร์(Single-Tier).....6
2.2. สถาปัตยกรรมของเว็บแบบทูเทียร์(Two-Tier).....7
2.3. สถาปัตยกรรมของเว็บแบบทรีเทียร์(Three-Tier).....8
2.4. สัญลักษณ์ที่ใช้แทน ยูสเคส.....9
2.5. สัญลักษณ์ที่ใช้แทน แอคเตอร์.....9
2.6. ตัวอย่างแผนภาพยูสเคส.....10
2.7. ตัวอย่างแผนภาพแสดงลำดับการทำงาน.....11
2.8. ตัวอย่างแผนภาพแสดงการทำงานร่วมกัน.....12
2.9. ตัวอย่าง ความสัมพันธ์แบบมีส่วนร่วม (Association).....13
2.10. ตัวอย่าง ความสัมพันธ์แบบเป็นส่วนหนึ่ง (Aggregation).....13
2.11.ตัวอย่างการกำหนดความสัมพันธ์แบบสืบทอดคุณสมบัติเพื่อเป็นการเจาะจงซับคลาสให้มีความเฉพาะมากขึ้น.....14
2.12. ตัวอย่างไคลเอ็นต์เพจ.....14
2.13. ตัวอย่างฟอร์ม.....15
2.14. ตัวอย่างเซิร์ฟเวอร์เพจ.....15
2.15 ตัวอย่างคลาสคอมโพเนนท์ซึ่งทำงานอยู่ที่เทียร์กลาง.....16
2.16. ตัวอย่างแผนภาพสถานะ.....16
2.17. ตัวอย่างการออกแบบสถาปัตยกรรมของซอฟต์แวร์เป็นแบบระดับชั้น.....17
2.18. ตัวอย่างแผนภาพการติดตั้งการใช้งาน.....18
2.19. ตัวอย่างแผนภาพกิจกรรม.....18
2.20.แสดงโครงสร้างพื้นฐานของเทคโนโลยี Web Service.....25
2.21 แสดงลักษณะการเชื่อมต่อของเทคโนโลยี Web Service.....26
2.22 แสดงโครงสร้างและส่วนประกอบของ .NET Framework.....30
2.23. แสดงส่วนที่ .NET Framework มีหน้าที่รับผิดชอบ.....31
2.24. แสดง Oracle Explorer.....37
2.25. แสดง Designers and Wizards.....37
2.26. แสดง Automatic Code Generation.....38
2.27. แสดง PL/SQL Editor.....39

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.28. แสดง Stored Procedure Testing.....39
2.29. แสดง Oracle Data Windows.....40
2.30. แสดง SQL Query Windows.....40
2.31. แสดง Integrated Help System.....41
2.32. แสดงตัวอย่างของ Undirect graph.....42
2.33. แสดงตัวอย่างของ Direct graph.....43
2.34. แสดงตัวอย่างของ Cyclic graph.....43
2.35. ลำดับการเดินทางบนโหนดของการค้นหาแบบลึกก่อนบนโครงสร้างต้นไม้.....45
2.36. โครงสร้างข้อมูลแบบกราฟ.....45
2.37. ลำดับการค้นหาแบบกว้างก่อนบนโครงสร้างต้นไม้.....46
2.38. โครงสร้างข้อมูลแบบกราฟ.....46
2.39. แสดงการทำงานของทูเฟสคอมมิท.....50
3.1. ตัวอย่างของยูสเคสของระบบของตั๋วเครื่องบินหลายสายการบิน.....53
3.2. แผนภาพแสดงลำดับการทำงานของการค้นหาเที่ยวบิน.....55
3.3. แผนภาพแสดงลำดับการทำงานของการจองตั๋วเครื่องบิน.....56
3.4. แผนภาพแสดงลำดับการทำงานของการชำระเงินของการจองตั๋ว.....57
3.5. แผนภาพแสดงลำดับการทำงานของการเปลี่ยนแปลงเที่ยวบิน.....58
3.6. แผนภาพแสดงลำดับการทำงานของการยกเลิกเที่ยวบิน.....59
3.7. แผนภาพแสดงลำดับของการสมัครสมาชิก.....60
3.8. แผนภาพแสดงลำดับการทำงานของการอัพเดทจำนวนที่นั่งจากสายการบินพันธมิตร.....61
3.9. แผนภาพแสดงการทำงานร่วมกันของการค้นหาสายการบิน.....62
3.10. แผนภาพแสดงการทำงานร่วมกันของการจองเที่ยวบิน.....63
3.11. แผนภาพแสดงการทำงานร่วมกันของการชำระตั๋วโดยสาร.....63
3.12. แผนภาพแสดงการทำงานร่วมกันของการเปลี่ยนแปลงเที่ยวบิน.....64
3.13. แผนภาพแสดงการทำงานร่วมกันของยกเลิกเที่ยวบิน.....65
3.14. แผนภาพแสดงการทำงานร่วมกันของการสมัครสมาชิก.....66
3.15. แผนภาพแสดงการทำงานร่วมกันของการเข้าสู่ระบบสมาชิก.....66
3.16. แผนภาพแสดงการทำงานร่วมกันของตรวจสอบประวัติการเดินทาง.....67
3.17 แผนภาพคลาสแสดงการเข้าสู่ระบบสมาชิก.....68
3.18 แผนภาพคลาสแสดงการค้นหาเที่ยวบิน.....69

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.19 แผนภาพแสดงคลาสของการทำการจองตั๋วเครื่องบิน.....70
3.20 แผนภาพคลาสแสดงการเปลี่ยนแปลงเที่ยวบิน.....71
3.21 แผนภาพคลาสแสดงการชำระเงิน.....73
3.22. แสดงการให้บริการเว็บเซอร์วิสในการให้บริการบนเว็บจองตั๋วหลายสายการบิน.....74
3.23. แสดงการให้บริการเว็บเซอร์วิสในกรณีการจองตั๋วสายการบิน.....75
3.24. แสดงการให้บริการเว็บเซอร์วิสในกรณีการยกเลิกการจองตั๋วสายการบิน.....76
3.25. แสดงการให้บริการเว็บเซอร์วิสในกรณีการเปลี่ยนวันเวลาและสายการบิน
จากการจองตั๋วบินหลายสายการบิน.....77
3.26. แสดงการให้บริการเว็บเซอร์วิสในกรณีการตรวจสอบข้อมูลที่นั่งว่าง.....78
3.27. แสดงการให้บริการเว็บเซอร์วิสในกรณีการชำระเงินผ่านบัตรเครดิต.....79
3.28. แสดงการให้บริการเว็บเซอร์วิสในกรณีการอัพเดทข้อมูลเวลาการบินเข้า-ออกจริงจาก
สนามบินของสายการบินพันธมิตร.....79
3.29. แสดงการไหลเวียนของหน้าเพจของเว็บแอพพลิเคชันของการจองตั๋วหลายสายการบิน.....80
3.30 แสดงERDiagram ของระบบการจองตั๋วเครื่องบินหลายสายการบิน.....81
3.31. แสดงการสร้างฟอร์มบนไคลเอ็นต์เพจในวิชวลสตูดิโอดอทเน็ต ในส่วนหน้าเพจของการ
ค้นหา.....82
3.32. ฟอร์มสำเร็จรูปที่สามารถให้เรียกใช้บนไคลเอ็นต์เพจได้.....83
3.33. แสดงโคดที่วิชวลสตูดิโอดอทเน็ตทำการสร้างให้ในการสร้างฟอร์มต่างๆ
ในส่วนหน้าเพจของการค้นหา.....84
3.34 แสดงการตัวอย่างหน้าไคลเอ็นต์เพจของการชำระเงิน.....84
3.35. แสดงส่วนโคดของเซิร์ฟเวอร์เพจของการค้นหาเที่ยวบิน.....85
3.36. แสดงความสัมพันธ์ระหว่างส่วนโคดและหน้าฟอร์มของ
เซิร์ฟเวอร์เพจของการค้นหาเที่ยวบิน.....85
3.37. แสดงเมธอดของการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูลในส่วนของคอมโพแนนท์เทียร์กลาง.....86
4.1. ส่วนของหน้าเว็บก่อนทำการค้นหาเส้นทาง.....87
4.2. ในกรณีเลือกขาไปเพียงอย่างเดียว.....88
4.3. ในกรณีเลือกขาไปและขากลับ.....88
4.4. ฟอร์มใส่รายละเอียดของเที่ยวบินที่ต้องการค้นหา.....88
4.5. ฟอร์มชื่อเมืองต้นทางและปลายทางที่ต้องการ.....88
4.6. แสดงผลของการค้นหาทั้งขาไปและขากลับ.....89

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.7. การแสดงผลของการค้นหาแบบหลายเที่ยวบินในหนึ่งการเดินทาง.....89
4.8. ลิงค์ของรายละเอียดการเดินทางในการเดินทางแต่ละเส้นทาง.....90
4.9. แสดงรายละเอียดการเดินทางในการเดินทางแต่ละเส้นทาง.....90
4.10. แสดงปุ่มของการเลือกการเดินทาง.....91
4.11. แสดงผลหลังจากการเลือก โดยแสดงแต่ละเที่ยวบินของการเดินทางครั้งหนึ่ง.....91
4.12. แสดงผลของเที่ยวบินและราคารวมของการเลือกการเดินทาง.....91
4.13 แสดงผลหน้าฟอร์มของการกรอกเพื่อยืนยันสิทธ์ในการจอง.....92
4.14. แสดงการตอบกลับไปให้กับผู้ใช้ว่าการจองได้เสร็จสิ้นแล้ว.....92
4.15. แสดงรายละเอียดต่างๆของเที่ยวบินที่ได้ถูกจองไปแล้วทั้งหมด.....93
4.16. แสดงฟอร์มการจ่ายตังก์ผ่านทางบัตรเครดิต.....93
4.17. หน้าเพจแสดงขั้นตอนการชำระเงินที่เสร็จสมบูรณ์.....94
4.18. แสดงข้อมูลต่างๆของตั๋วที่ได้จองไปแล้วรวมทั้งหมายเลข Booking Reference.....94
4.19. ฟอร์มกรอกหมายเลขตั๋วเพื่อทำการตรวจสอบรายละเอียดของเที่ยวบิน
รวมทั้งการเปลี่ยนหรือยกเลิกเที่ยวบินที่ได้ทำการจองไปแล้ว.....95
4.20. หน้าเพจของการเปลี่ยนแปลงเที่ยวบิน โดยให้ทำการเลือกเงื่อนไขใหม่.....95
4.21. หน้าเพจแสดงเที่ยวบินที่สามารถทำการเปลี่ยนได้.....96
4.22. ผลลัพท์เมื่อได้ทำการเปลี่ยนแปลงเที่ยวบินเสร็จแล้ว.....96
4.23. แสดงหน้าเพจของการยืนยันการยกเลิกเที่ยวบินและเหตุผลของการยกเลิกเที่ยวบิน.....97
4.24. แสดงหน้าเพจของผลลัพท์ของการสั่งยกเลิกเที่ยวบิน.....97
4.25. แสดงการตอบกลับไปให้กับผู้ใช้ว่าการยกเลิกเที่ยวบินได้เสร็จสิ้นแล้ว.....98
4.26. แสดงแบบฟอร์มของการสมัครเป็นสมาชิก.....98
4.27. แสดงการตรวจสอบเมื่อไม่กรอก Username และ Password.....99
4.28. แสดงการตรวจสอบเมื่อไม่กรอก Password..... 99
4.29. แสดงการตรวจสอบเมื่อ Password นั้นไม่ตรงกัน.....99
4.30. แสดงการตรวจสอบเมื่อชื่อสมาชิก (username) นี้ได้ถูกทำการสมัครไปแล้ว.....100
4.31. แสดงผลของการสิ้นสุดสมัครสมาชิก.....100
4.32. เมื่อลืมรหัสผ่านให้ทำการกรอกชื่อสมาชิกแล้วกดปุ่ม Forget Password.....101
4.33. ระบบจะทำการส่งเมลล์ให้กับสมาชิกเพื่อบอกรหัสสมาชิก.....101
4.34. ฟอร์มของการเปลี่ยนแปลงรายละเอียดของสมาชิก.....102

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.35 แสดงการเดินทางทั้งหมดของสมาชิกรวมทั้งไมล์สะสม.....103
4.36. แสดงฟอร์มการเลือกสนามบินที่ต้องการตรวจสอบเที่ยวบิน เข้า-ออก.....103
4.37. แสดงผลของการเลือกสนามบินที่เข้า-ออก.....103
4.38. แสดงฟอร์มของการอัพเดทจำนวนที่นั่งของแต่ละสายการบิน.....104
4.39. แสดงฟอร์มของการอัพเดทสถานะเที่ยวบินของแต่ละสนามบิน.....105
4.40. แสดงหน้าหลักของ Oracle Enterprise Manager.....106
4.41. แสดงหน้าเว็บของการวัดประสิทธิภาพต่างๆที่เกิดขึ้นบนเซิร์ฟเวอร์ฐานข้อมูล.....107
4.42. แสดงหน้าเว็บของการจัดการกับโครงสร้างของตาราง.....107
4.43. แสดงหน้าเว็บของการจัดการเกี่ยวกับรูปแบบข้อมูลของโครงสร้างตาราง.....108
4.44. แสดงหน้าเว็บของการจัดการเกี่ยวกับเทเบิลสเปซ.....108
4.45. แสดงหน้าเว็บของการจัดการเกี่ยวกับยูสเซอร์.....109
4.46. แสดงหน้าเว็บของการจัดการเกี่ยวกับรายละเอียดต่างๆของยูสเซอร์.....109
4.47. แสดงหน้าเว็บเพจของเครื่องมือ ISQL plus.....110
4.48. แสดงการเรียกใช้คำสั่งผ่านเครื่องมือ ISQL plus.....110

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันเมื่ออินเทอร์เน็ตมีการใช้งานอย่างกว้างขวางขึ้น ความต้องการประยุกต์ใช้สิ่งใหม่ ๆ บนอินเทอร์เน็ตจึงได้รับการพัฒนามากขึ้น โดยเฉพาะอย่างยิ่งการใช้อินเทอร์เน็ตเพื่ออำนวยความสะดวกสบายในการกระทำธุรกรรมต่างๆ อีกทั้งยังช่วยลดเวลาและความยุ่งยากในการดำเนินการลงอย่างมาก

เว็บแอพพลิเคชั่น(Web Application) ก็เป็นอีกผลพลอยได้หนึ่งจากเทคโนโลยีของอินเทอร์เน็ต ที่มีการพัฒนาจากผู้พัฒนาต่างๆมากมาย โดยเว็บแอพพลิเคชัน (Web Application) ในปัจจุบันนั้นมีขนาดที่ใหญ่และซับซ้อนมากยิ่งขึ้น ส่งผลให้นักพัฒนาต้องทำงานกันเป็นทีม แต่เมื่อทีมมีขนาดใหญ่มากขึ้นรวมทั้งการผลัดเปลี่ยนของนักพัฒนาที่เพิ่มมากขึ้นทุกๆปี ก็อาจจะทำให้นักพัฒนาเกิดความสับสนและความไม่สอดคล้องกันในการพัฒนาเว็บแอพพลิเคชัน (Web Application) นี้ได้ ปัญหาเหล่านี้จึงจำเป็นต้องได้รับการแก้ไข

ระบบการจองตั๋วหลายสายการบิน ก็เป็นอีกตัวอย่างหนึ่งที่มีการนำเทคโนโลยีทางด้านอินเทอร์เน็ตเข้ามาช่วยอำนวยความสะดวก โดยจะช่วยลดความยุ่งยากในการค้นหาเส้นทางการเดินทาง และสั่งจองตั๋วของแต่ละสายการบิน โดยหากสามารถเชื่อมต่อกับอินเตอร์เน็ตได้ ก็สามารถค้นหาข้อมูลที่ต้องการเหล่านี้ได้ตลอดเวลา ทั้งนี้ยังช่วยสั่งจองตั๋วโดยสารให้ได้ โดยไม่ต้องเสียเวลาในการเดินทางอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

ปริญญานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาการพัฒนาเว็บแอพพลิเคชั่น(Web Application) ที่ง่ายต่อการดูแลรักษาและการเพิ่มเติมต่อในระบบให้กับผู้พัฒนา โดยผู้พัฒนาที่จะพัฒนาต่อนั้นสามารถที่จะทำความเข้าใจส่วนต่างๆของระบบได้ แม้ว่าจะไม่เคยพัฒนาระบบนี้มาก่อนก็ตาม

โดยเว็บแอพพลิเคชั่นที่ได้ทำการศึกษานี้จะต้องมีคุณสมบัติในการทำการจองตั๋วหลายสายการบิน เช่น การค้นหาเส้นทางการบินของสายการบินหรือการทำการจองผ่านหลายสายการบินเพร้อมกันโดยใช้เทคโนโลยีเว็บเซอร์วิส โดยการศึกษาและพัฒนาครั้งนี้ได้มีการนำเทคโนโลยีดอทเน็ต(.NET Technology) มาทำงานร่วมกันกับ ระบบการจัดการดาต้าเบสหรือดีบีเอ็มเอสโอราเคิล เท็นจี (DBMS Oracle 10g)

ดังนั้นในปริญญานิพนธ์ฉบับนี้จึงนำเสนอวิธีการพัฒนาเว็บแอพพลิเคชั่นของระบบจองตั๋วเครื่องบินที่สามารถที่จะทำการดูแลรักษาได้ง่าย โดยมีการนำเอาเทคโนโลยีดอทเน็ท(.NET Technology) มาทำงานร่วมกันกับ ระบบการจัดการดาต้าเบสหรือดีบีเอ็มเอส โอราเคิล เท็นจี (DBMS Oracle 10g)

1.3 ขอบเขตของโครงงาน

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการการออกแบบและพัฒนาต้นแบบระบบการจองตั๋วหลายสายการบินผ่านเว็บแอพพลิเคชั่น โดยอ้างอิงจาก เว็บไซต์ สตาร์อาไลนแอนซ์(Star Alliance) ซึ่งให้บริการลูกค้าในด้านการค้นหาเส้นทางและตารางเวลาของแต่ละสายการบินทั่วโลก โดยในโครงงานนี้ได้มีการเพิ่มคุณสมบัติต่างๆเข้าไปโดยคุณสมบัติของเว็บไซต์หลักๆจะมีดังนี้คือ

สามารถค้นหาเส้นทางในการเดินทางจากสายการบินต่างๆ ที่เข้าร่วมบริการ โดยผู้ใช้บริการอาจจะค้นหาจากต้นทางและปลายทางที่ต้องการเดินทาง, ช่วงเวลาในการเดินทางไปและกลับรวมทั้งเงื่อนไขต่างๆ ที่ต้องการ เช่น ราคา, ระยะเวลาของการเดินทาง โดยผลลัพธ์ที่ได้นั้นจะสามารถที่จะบอกรายละเอียดในการเดินทาง ไม่ว่าจะเป็นสายการบินที่ต้องเดินทางในแต่ละรอบ, ช่วงเวลาของการเดินทาง, ราคาตั๋วของการเดินทาง และแสดงรายละเอียดโดยเรียงจากเงื่อนไขที่ผู้ใช้ต้องการได้ เช่น ค้นหาเส้นทางที่ราคาค่าเดินทางน้อยที่สุด หรือเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด ได้

สามารถที่จะทำการเลือกเส้นทางและทำการสั่งจองตั๋วเครื่องบินได้โดยผ่านทางเว็บไซต์ได้เลย ตรงจุดนี้จะมีการนำเทคโนโลยีของเว็บเซอร์วิส เข้ามาจัดการในการสั่งจองไปที่เว็บไซต์ของสายการบินต่างๆ

สามารถที่จะทำการบำรุงและรักษาตัวเว็บไซต์ได้ง่าย โดยอาจให้นักพัฒนาคนอื่นที่เข้ามาพัฒนาระบบ สามารถที่จะเข้าใจและทำการ ปรับปรุงและแก้ไขส่วนต่างๆได้ แม้ว่านักพัฒนาคนนั้นจะไม่ได้เป็นผู้ที่พัฒนาระบบนี้ขึ้นมาเองตั้งแต่ต้นก็ตาม เช่น การ เพิ่ม-ลดสายการบินที่ร่วมบริการ โดยมุมมองของการพัฒนาเว็บไซต์นี้ จะเป็นในลักษณะของการพัฒนาเว็บไซต์ให้กับองค์กรที่ให้บริการจองตั๋วเครื่องบินในการเดินทางท่องเที่ยวหรือวาระต่างๆ โดยที่จะสามารถนำกลับมาพัฒนาได้ตามคำร้องขอขององค์กรที่ให้บริการนั้นๆ และทางผู้พัฒนาจะต้องสามารถที่จะทำการปรับเปลี่ยนและแก้ไข ตัวเว็บไซต์ตัวนี้ได้ตามคำร้องขอ แม้ว่าจะเคยหรือไม่เคยพัฒนาระบบนี้มาก่อนก็ตาม

1.4 วิธีการดำเนินการ

1. ศึกษาเกี่ยวกับกระบวนการและวิธีการพัฒนาเว็บแอพพลิเคชั่น เพื่อให้เข้าใจถึงขั้นตอนต่าง ในกระบวนการ เพื่อที่จะนำมาประยุกต์ใช้ในพัฒนาแวพแอพพลิเคชั่นต้นแบบ
2. ศึกษาเกี่ยวกับกระบวนการของการพัฒนาเทคโนโลยีเว็บเซอร์วิส
3. ศึกษาเกี่ยวกับทฤษฎีการค้นหา(Theory Search)แบบต่างๆ
4. ศึกษาการใช้งานของกลุ่มเครื่องมือ วิชวลสตูดิโอ ดอท เน็ต(Visual Studio .NET) ศึกษาการใช้งานของระบบการจัดการดาต้าเบสโอราเคิล เท็นจี (DBMS Oracle 10g)
5. ศึกษาการเขียนโปรแกรมในรูปแบบ ดาต้าเบสแอพพลิเคชั่น(Database Application)
6. ศึกษาการเขียนโปรแกรมโดยใช้ภาษา เอเอสพี ดอทเน็ต ASP.NET และ วิชวลเบสิก ดอท เน็ต (Visual Basic .NET)
7. ศึกษาเกี่ยวกับระบบการจัดการข้อมูลในตารางการบินของสายการบิน
8. ศึกษาขั้นตอนการทำ ทู เฟส คอมมิท (Two Phase Commit) เพื่อนำไปใช้ในการจองที่นั่งของเครื่องบินกับทางสายการบินร่วมต่างๆ
9. วิเคราะห์ผลของระบบการสั่งจองที่นั่งและการบำรุงรักษาเว็บแอพพริเคชั่น รวมทั้งทำการแก้ไขส่วนที่ผิดพลาดเพื่อให้ระบบสามารถทำงานได้อย่างสมบูรณ์ที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจในกระบวนการพัฒนาเว็บแอพพลิเคชั่น
2. ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการของการพัฒนาเทคโนโลยีเว็บเซอร์วิส
3. ได้รับความรู้ความเข้าใจแกี่ยวกับกระบวนการของการพัฒนาเทคโนโลยีเว็บเซอร์วิส
4. ได้รับความรู้ความเข้าใจแกี่ยวกับทฤษฎีการค้นหา(Theory Search)แบบต่างๆ
5. ได้รับความรู้ความเข้าใจในการใช้งานของกลุ่มเครื่องมือ วิชวลสตูดิโอ ดอท เน็ต(Visual Studio .NET
6. ได้รับความรู้ความเข้าใจในการใช้งานของระบบการจัดการดาต้าเบสโอราเคิล เท็นจี (DBMS Oracle 10g)
7. ได้รับความรู้ความเข้าใจในการเขียนโปรแกรมในรูปแบบดาต้าเบสแอพพลิเคชั่น (Database Application)
8. ได้รับความรู้ความเข้าใจในการเขียนโปรแกรมโดยใช้ภาษา เอเอสพี ดอทเน็ต ASP.NET และ วิชวลเบสิก ดอท เน็ต (Visual Basic .NET)
9. ได้รับความรู้ความเข้าใจเกี่ยวกับระบบการจัดการข้อมูลในตารางการบินของสายการบิน
10. ได้รับความรู้ความเข้าใจในขั้นตอนการทำ ทู เฟส คอมมิท (Two Phase Commit)

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วย ขั้นตอนการพัฒนาเว็บแอพพลิเคชัน เทคโนโลยีดอทเน็ต เทคโนโลยีเว็บเซอร์วิส ภาษาเอเอสพีดอทเน็ตจากไมโครซอร์ฟดอทเน็ตเฟรมเวิร์ก วิชวลสตูดิโอดอทเน็ต ออราเคิลดีวีลอปเปอร์ทูลสำหรับวิชวลสตูดิโอดอทเน็ต ทฤษฎีการค้นหาของกราฟ และการทำทูเฟส คอมมิทด้วยเวปเซอร์วิส

บทที่ 3 จะเป็นส่วนของการออกแบบเว็บแอพพลิเคชันและเว็บเซอร์วิสตามคุณสมบัติต่างๆ ของเว็บการจองตั๋วหลายสายการบิน

บทที่ 4 จะเป็นส่วนของการทดลองใช้งานของตัวเว็บแอพพลิเคชันของการจองตั๋วหลายสายการบิน

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการวิจัย ไม่ว่าจะเป็นพื้นฐานของแนวทางการออกแบบเว็บแอพลิเคชัน เทคโนโลยีเว็บเซอร์วิส โครงสร้างดอทเน็ตเทคโนโลยี ทฤษฎีการค้นหาเส้นทางต่างๆ รวมทั้งเครื่องมือต่างๆที่ใช้ในการศึกษานี้ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา และประเมินประสิทธิภาพของการพัฒนาเว็บแอพพลิที่ทำการทดลองนี้

2.1. แนวทางการพัฒนาเว็บแอพพลิเคชัน

ในปัจจุบันเมื่ออินเทอร์เน็ตมีการใช้งานอย่างกว้างขวางขึ้น ความต้องการประยุกต์ใช้สิ่งใหม่ ๆ บนอินเทอร์เน็ตจึงได้รับการพัฒนามากขึ้น รวมทั้งเว็บไซต์ ก็เป็นอีกหนึ่งเทคโนโลยีที่มีการพัฒนาอย่างต่อเนื่องเพราะฉะนั้นการให้บริการเว็บในปัจจุบันไม่เพียงแต่ความหลากหลายของข้อมูล และรูปแบบที่ให้บริการเท่านั้น ตัวเว็บไซต์เองก็มีขนาดใหญ่และมีความซับซ้อนมากขึ้นอีกด้วย

2.1.1. รูปแบบการพัฒนาเว็บไซต์

สำหรับการพัฒนาเว็บนั้นเราสามารถแบ่งระดับออกได้เป็นสามระดับคือ

- การพัฒนาเว็บสำหรับให้บริการข้อมูล
- การพัฒนาเว็บแบบโต้ตอบได้
- การพัฒนาเว็บแอพพลิเคชัน

2.1.1.1. การพัฒนาเว็บสำหรับให้บริการข้อมูล

เป็นการสร้างเว็บไซต์ที่มีเป้าหมายหลักสำหรับการให้บริการข้อมูล ซึ่งเป็นการให้บริการแบบทางเดียว โดยผู้พัฒนาสามารถสร้างเว็บไซต์ขึ้นมาได้ในทันทีโดยที่ไม่ต้องผ่านการกระบวนการออกแบบ แต่ถ้าเว็บมีขนาดที่ใหญ่มากก็อาจจะจำเป็นที่ต้องมีการออกแบบโครงสร้างเว็บไซต์ โดยในการออกแบบเว็บไซต์ประเภทนี้ผู้ออกแบบจะต้องเน้นไปที่ความสะดวกของผู้ใช้ในการเข้าถึงข้อมูล ซึ่งจำเป็นต้องออกแบบโครงสร้างของเว็บให้สามารถเข้าถึงแต่ละส่วนได้ง่าย และ ไม่ซับซ้อนจนเกินไป

2.1.1.2. การพัฒนาเว็บแบบโต้ตอบได้

เป็นการพัฒนาเว็บไซต์ที่มีความสามารถในการรับข้อมูลจากผู้ใช้เข้ามาประมวลผล หรือนำมาเก็บไว้เพื่อนำมาใช้ในอนาคต พร้อมทั้งมีการแสดงหน้า เอชทีเอ็มแอลแบบไดนามิกตามข้อมูลที่ผู้ใช้ป้อนเข้ามา โดยทั่วไปแล้วการพัฒนาเว็บแบบนี้จะต้องมีฟังก์ชันการให้บริการข้อมูล

อยู่ด้วยเป็นหลัก แต่จะมีการเพิ่มความสามารถในการรับข้อมูลจากผู้ใช้ เช่น บอร์ดแสดงความคิดเห็น (Web Board) หรือการฝากข้อความ ส่วนใหญ่แล้วการพัฒนาเว็บแบบนี้จะมีการประมวลผลและการเก็บข้อมูลที่ไม่ซับซ้อนมาก โปรแกรมที่สร้างนั้นก็มีขนาดไม่ใหญ่และไม่จำเป็นต้องใช้ระบบฐานข้อมูล การออกแบบเว็บประเภทนี้นั้น ผู้ออกแบบจะต้องคำนึงถึงการออกแบบคล้ายกับการพัฒนาเว็บสำหรับให้บริการข้อมูลรวมทั้งยังต้องมีการทำโมเดลแสดงอัลกอริทึมเพื่อแสดงการทำงานของโปรแกรมที่มีอยู่ในเว็บเพื่อให้สามารถกลับมาแก้ไขโปรแกรมได้ง่ายอีกด้วย

2.1.1.3. การพัฒนาเว็บแอพพลิเคชัน

เป็นการพัฒนาเว็บไซต์ที่มีความซับซ้อนมากที่สุด เมื่อเทียบกับการพัฒนาจากทั้งสองแบบที่กล่าวมา โดยเว็บแอพพลิเคชันนั้นจะประกอบไปด้วยส่วนสำหรับให้บริการข้อมูล ส่วนที่โต้ตอบกับผู้ใช้ และส่วนที่ใช้ในการทำการประมวลผล โดยทางผู้พัฒนานั้นอาจจะต้องใช้ภาษาในการเขียนโปรแกรมเฉพาะเพื่อให้สามารถควบคุมการประมวลผลได้ง่าย นอกจากนี้แล้วผู้พัฒนายังต้องคำนึงการติดต่อสื่อสารกับภายนอก และส่วนการเก็บข้อมูลที่มีความซับซ้อน ซึ่งบางครั้งอาจจะจำเป็นต้องมีการใช้ ระบบการจัดการฐานข้อมูลมาช่วยด้วย สำหรับการพัฒนาเว็บแอพพลิเคชันผู้พัฒนาจะต้องทำการโมเดลที่แสดงการทำงานของเว็บแอพพลิเคชันเพิ่มขึ้นมา โดยต้องมีการออกแบบ โครงสร้างส่วนประกอบ และการออกแบบฐานข้อมูลด้วย

2.1.2. สถาปัตยกรรมในการติดตั้งเว็บไซต์

เราสามารถที่จะทำการแบ่งสถาปัตยกรรมในการติดตั้งเว็บไซต์ออกได้เป็น 3 แบบ คือ

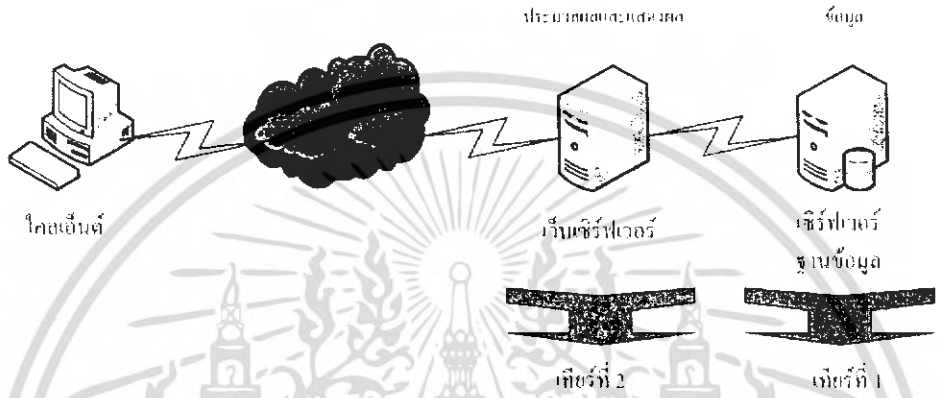
2.1.2.1. สถาปัตยกรรมแบบซิงเกิลเทียร์(Single-Tier)

เป็นสถาปัตยกรรมที่ประกอบไปด้วยเว็บเซิร์ฟเวอร์ และไคลเอ็นต์ โดยเว็บเซิร์ฟเวอร์นั้นจะทำหน้าที่ในการให้บริการข้อมูลของเว็บรวมทั้งการประมวลต่างๆ ส่วนไคลเอ็นต์นั้นจะแสดงผลต่อผู้ใช้และอาจจะทำการส่งข้อมูลจากผู้ใช้กลับไปให้เซิร์ฟเวอร์เพื่อประมวลผลและเก็บไว้ด้วย ดังรูปที่ 2.1.

รูปที่2.1. สถาปัตยกรรมของเว็บแบบซิงเกิลเทียร์(Single-Tier)

2.1.2.2. สถาปัตยกรรมแบบทูเทียร์(Two-Tier)

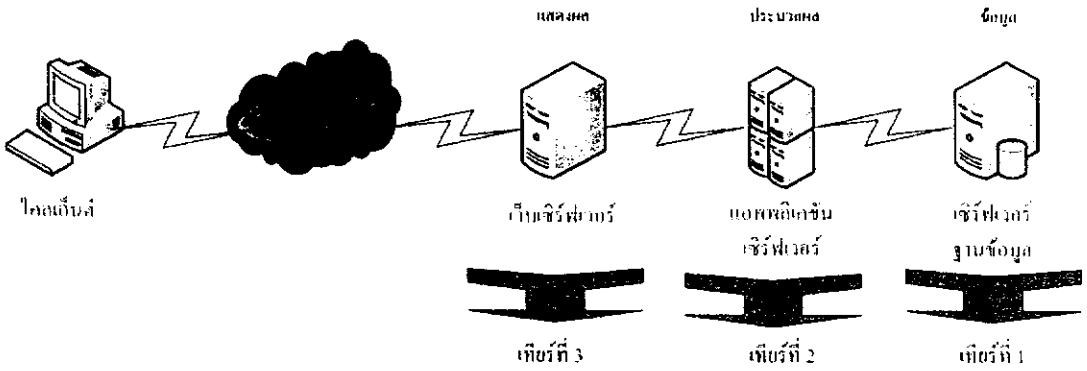
เป็นสถาปัตยกรรมที่ประกอบไปด้วยเว็บเซิร์ฟเวอร์ ไคลเอ็นต์รวมทั้งมีการเพิ่มในส่วนของระบบฐานข้อมูล ซึ่งเหมาะสำหรับเว็บแอพพลิเคชันที่เน้นการเก็บข้อมูลของผู้ใช้เป็นหลัก ดังรูปที่ 2.2.



รูปที่2.2. สถาปัตยกรรมของเว็บแบบทูเทียร์(Two-Tier)

2.1.2.3. สถาปัตยกรรมแบบทรีเทียร์(Three-Tier)

เป็นสถาปัตยกรรมที่มีการแยกระบบออกเป็น ระบบฐานข้อมูล แอพพลิเคชันเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์และไคลเอ็นต์ โดยในส่วนของแอพพลิเคชันเซิร์ฟเวอร์จะทำหน้าที่ในการประมวลผลหรือทำธุรกรรมต่างๆ เป็นหลัก ส่วนเว็บเซิร์ฟเวอร์นั้นจะนำผลลัพธ์ที่ได้จากการประมวลผลจากแอพพลิเคชันเซิร์ฟเวอร์มาจัดหน้าตา เพื่อส่งไปให้ยังไคลเอ็นต์แสดงผลต่อไป ดังรูปที่ 2.3. เนื่องจากสถาปัตยกรรมทรีเทียร์จะมีการแบ่งส่วนประมวลผลและส่วนของการแสดงผลแยกออกจากกันอย่างชัดเจน จึงทำให้ผู้พัฒนาสามารถที่จะทำการเปลี่ยนแปลงและแก้ไขส่วนประมวลผลได้โดยไม่กระทบต่อส่วนแสดงผล หรือมีการเปลี่ยนแปลงหน้าตาส่วนแสดงผลก็จะไม่มีผลกระทบต่อส่วนประมวลผลเป็นต้น นอกจากนี้สถาปัตยกรรมแบบทรีเทียร์นี้ยังสนับสนุนความสามารถในการขยายระบบ และความสามารถในการแบ่งภาระงานอีกด้วย



รูปที่ 2.3. สถาปัตยกรรมของเว็บแบบทรีเทียร์(Three-Tier)

2.1.3.ภาษายูเอ็มแอลกับเว็บ

ภาษายูเอ็มแอล (Unified Modeling Language) ได้รับการพัฒนาขึ้นโดยบริษัท เรชันแนล ซอฟต์แวร์ (Rational Software) ซึ่งเป็นภาษาที่ใช้ในการอธิบายซอฟต์แวร์ โดยจะมีการนำภาษายูเอ็มแอลนี้มาใช้ตั้งแต่ เริ่มพัฒนาซอฟต์แวร์ คือ การเก็บความต้องการ การวิเคราะห์ การออกแบบ พัฒนา จนไปถึงขั้นตอนของการทดสอบ โดยส่วนประกอบหลักของภาษายูเอ็มแอลคือ แผนภาพ (Diagram) และ สัญลักษณ์(Notation)ที่มีความหมาย

ถึงแม้ว่าการประยุกต์ใช้ภาษายูเอ็มแอลกับการพัฒนาเว็บไซต์นั้นจะไม่สามารถทำได้โดยตรง แต่ถ้าผู้พัฒนาทราบถึงวิธีทำความเข้าใจและตีความหมายของสัญลักษณ์ในภาษายูเอ็มแอลให้ตรงกับส่วนประกอบของเว็บไซต์ก็สามารถทำได้ไม่ยาก โดยผู้พัฒนาจะต้องมีความรู้ความเข้าใจในตัวเทคโนโลยีที่ใช้ในการพัฒนาเว็บไซต์นั้น อีกทั้งต้องทำความเข้าใจกับ แนวความคิด ความหมาย และสัญลักษณ์ของภาษายูเอ็มแอลพื้นฐาน จากนั้นจึงทำความเข้าใจว่าสัญลักษณ์และความหมายที่มีให้ในภาษายูเอ็มแอล นั้นจะนำไปโมเดลส่วนประกอบของเว็บไซต์ได้อย่างไร

โดยในตัวภาษายูเอ็มแอลก็มีแผนภาพมากมายคือ

2.1.3.1. แผนภาพยูสเคส(Use Case diagram)

เป็นแผนภาพที่ใช้ในการแสดงความต้องการในแง่ของฟังก์ชันการทำงาน (Functional requirements) ที่สำคัญของระบบ และอาจจะรวมถึงความต้องการที่ไม่เป็นฟังก์ชัน(Nonfunctional requirements) ที่สำคัญด้วย ส่วนแผนภาพยูสเคสในมุมมองของผู้ใช้นั้นจะเป็นแผนภาพที่แสดงฟังก์ชันการทำงานหลักของซอฟต์แวร์ ทีมพัฒนาสามารถใช้แผนภาพยูสเคสนี้เป็นเอกสารหลักในการสื่อสารกับผู้ใช้ซอฟต์แวร์ และยังสามารถใช้สื่อสารระหว่างสมาชิกในทีมพัฒนาได้อีกด้วย

สำหรับในแผนภาพยูสเคสจะประกอบไปด้วยส่วนที่สำคัญอยู่ 3 ส่วน คือ ยูสเคส(use case), แอคเตอร์(actor), และความสำคัญระหว่างยูสเคส กับ แอคเตอร์

2.1.3.1.1. ยูสเคส จะเป็นตัวบ่งบอกถึงฟังก์ชันที่สำคัญซึ่งระบบสามารถทำได้ โดยการเขียนข้อความภายในยูสเคสนั้น ควรที่จะสื่อภาพให้เห็นได้ว่ายูสเคสนั้นเกี่ยวข้องอะไรกับการทำซอฟต์แวร์ และโดยทั่วไปแล้วข้อความในยูสเคสนั้นควรที่จะเป็นคำกริยา เช่น ยูสเคสสำหรับการจองตั๋วเครื่องบิน,เปลี่ยนแปลงเที่ยวบิน เป็นต้น โดยสัญลักษณ์จะเป็นดังรูปที่ 2.4.

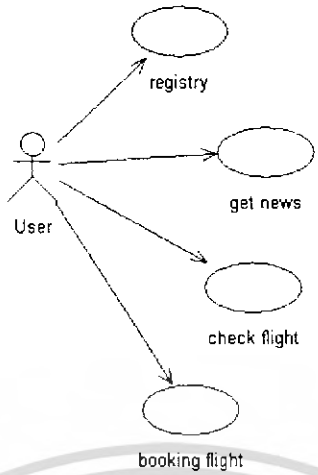
รูปที่2.4. สัญลักษณ์ที่ใช้แทน ยูสเคส

2.1.3.1.2. แอคเตอร์ เป็นตัวบ่งบอกถึงสิ่งแวดล้อมภายนอกที่เกี่ยวข้องกับซอฟต์แวร์ เช่น ลูกค้า,ผู้ใช้งานระบบ หรือ ผู้ดูแลระบบเป็นต้น นอกจากนี้แอคเตอร์ไม่จำเป็นที่จะต้องเป็นมนุษย์เท่านั้น แต่ยังสามารถที่จะยังเป็นระบบได้ หรือเป็นอุปกรณ์ที่อยู่นอกซอฟต์แวร์ได้ โดยสัญลักษณ์จะเป็นดังรูปที่ 2.5.

รูปที่ 2.5. สัญลักษณ์ที่ใช้แทน แอคเตอร์

2.1.3.1.3. ความสัมพันธ์ระหว่างยูสเคส กับ แอคเตอร์ จะใช้เส้นตรงลากจาก แอคเตอร์มายังยูสเคสที่เกี่ยวข้อง โดยความสำคัญระหว่างยูสเคสกับแอคเตอร์นั้นเป็นความสัมพันธ์แบบแมนนีทูแมนนี (many-to-many) คือแอคเตอร์หนึ่งอาจจะมีเกี่ยวข้องกับหลายยูสเคสได้ และยูสเคสหนึ่งยูสเคสก็อาจจะเกี่ยวข้องกับแอคเตอร์หลายแอคเตอร์ได้ ดังรูป2.6.

นอกจากนี้แล้วการกำหนดความสัมพันธ์ระหว่างยูสเคสนั้นก็ยังสามารถทำได้ เพื่อให้เป็นการกำหนดรายละเอียดของยูสเคสภายในมุมมองของทีมพัฒนา โดยอาจจะมีการสืบทอดคุณสมบัติให้กับยูสเคสจากตัวหนึ่งสู่อีกตัวหนึ่งได้

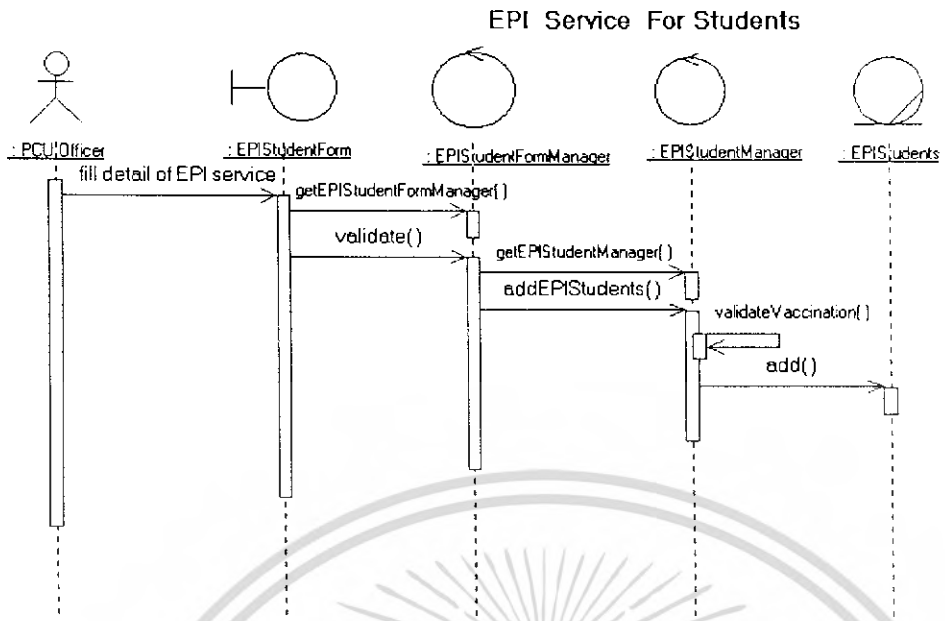


รูปที่ 2.6. ตัวอย่างแผนภาพยูสเคส

2.1.3.2. แผนภาพแสดงการทำงาน(Scenario diagram)

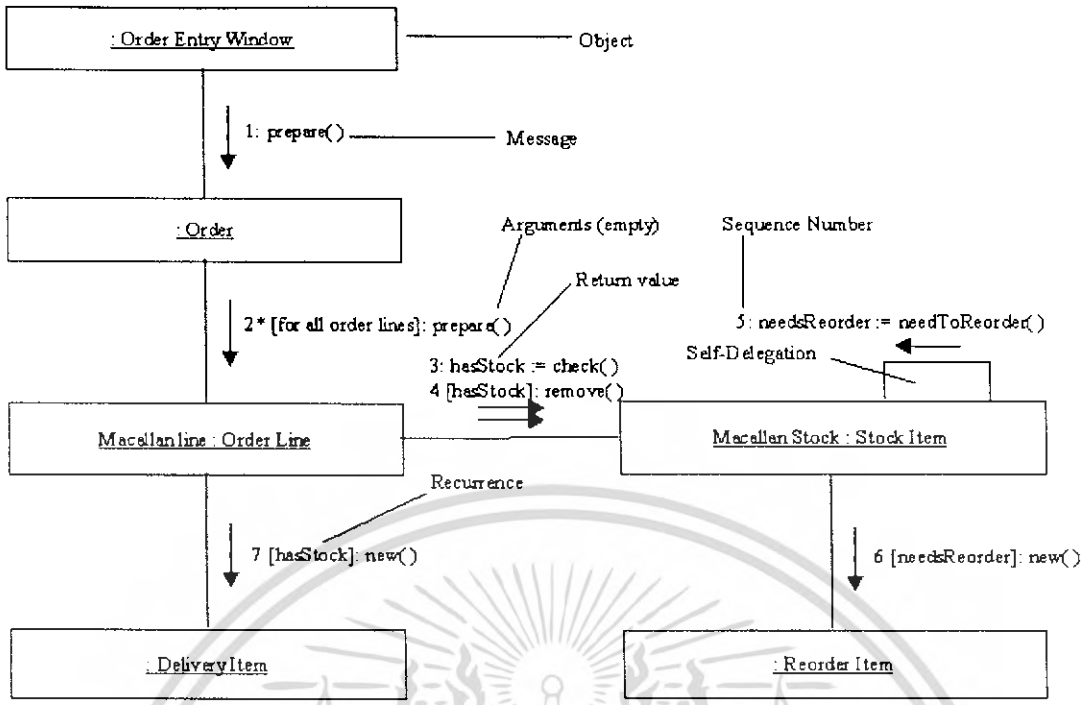
เป็นขั้นตอนของการที่จะอธิบายขั้นตอนการทำงานในกรณีหนึ่งของยูสเคส โดยจะเลือกเหตุการณ์ที่สำคัญและมีบทบาทหลักต่อยูสเคส เพื่อมาอธิบายให้ทั้งผู้ใช้และทีมพัฒนาซอฟต์แวร์เข้าใจ ซึ่งการอธิบายการทำงานของยูสเคสในกรณีหนึ่งนั้นสามารถทำได้โดยใช้แผนภาพที่ภาษายูเอ็มแอลมีให้ซึ่งประกอบด้วย **แผนภาพแสดงลำดับการทำงาน (Sequence Diagram)** และ **แผนภาพการทำงานร่วมกัน (Collaboration Diagram)**

2.1.3.1.1 . แผนภาพแสดงลำดับการทำงาน (Sequence Diagram) นั้นจะแสดงการลำดับขั้นตอนของการ และอาจจะมีเงื่อนไขทางด้านเวลาเข้ามาเกี่ยวข้องด้วย สำหรับส่วนประกอบภายในก็จะประกอบไปด้วยส่วน แอคเตอร์, คลาสหรือออบเจ็กต์ และแมสเซจ (Message) ที่ไหลอยู่ภายในซอฟต์แวร์ โดยแมสเซจที่อยู่ด้านบนนั้นจะเป็นแมสเซจที่เกิดขึ้นก่อนแมสเซจที่อยู่ด้านล่าง และแมสเซจนั้นจะลากจากแอคเตอร์ คลาสหรือออบเจ็กต์ที่เป็นต้นกำเนิดไปยังปลายทางเพื่อบ่งบอกผู้เรียกและผู้ให้บริการ ดังรูป 2.7.



รูปที่ 2.7. ตัวอย่างแผนภาพแสดงลำดับการทำงาน

2.1.3.1.2. แผนภาพการทำงานร่วมกัน มีวัตถุประสงค์เพื่อดูใช้แสดงการทำงานสำหรับยูสเคสในกรณีหนึ่งเช่นเดียวกับแผนภาพแสดงลำดับการทำงาน แต่จะแตกต่างกันที่ การนำเสนอของแผนภาพแสดงการทำงานร่วมกันนั้นจะเน้นที่การแสดงโครงสร้างในการทำงานภายในยูสเคสมากกว่าที่จะแสดงลำดับ แต่กระนั้นแล้ว ในแผนภาพแสดงการทำงานร่วมกันนี้ก็มีการแสดงลำดับของการทำงานไว้ด้วยเช่นกัน สำหรับภายในแผนภาพแสดงการทำงานร่วมกันนั้นก็มีส่วนประกอบคือ แอคเตอร์, คลาสหรือออบเจ็กต์ ,ความสัมพันธ์ระหว่างกัน และแมสเซจ (Message) โดยที่แมสเซจนั้นจะมีการบ่งบอกลำดับการเกิดขึ้นด้วยว่าแมสเซจใดเกิดขึ้นก่อนแมสเซจใด ดังรูป 2.8.



รูปที่ 2.8. ตัวอย่างแผนภาพแสดงการทำงานร่วมกัน

2.1.3.3. แผนภาพออบเจ็กต์ (Object diagram)

แผนภาพออบเจ็กต์นั้นเป็นแผนภาพที่ใช้ในการแสดงอินสแตนซ์ (Instance) ของส่วนประกอบภายในซอฟต์แวร์ โดยส่วนใหญ่แล้วการพัฒนาซอฟต์แวร์จะไม่ได้เขียนแผนภาพออบเจ็กต์ขึ้นมาโดยตรง แต่จะใช้แผนภาพแสดงการทำงานร่วมกันแทน ทั้งนี้เนื่องจากแผนภาพแสดงการทำงานร่วมกันนั้นจะใช้ในการนำเสนอการสื่อสารระหว่างออบเจ็กต์ภายในซอฟต์แวร์อยู่แล้ว ส่งผลให้เครื่องมือในการเขียนแผนภาพของยูเอ็มแอลบางตัวไม่สนับสนุนการเขียนแผนภาพออบเจ็กต์

สำหรับส่วนประกอบภายในแผนภาพนั้นก็จะประกอบด้วย สัญลักษณ์แทนออบเจ็กต์ ซึ่งจะเป็นกล่องสี่เหลี่ยมพร้อมด้วยชื่อของออบเจ็กต์ซึ่งอยู่ภายใน และความสัมพันธ์ระหว่างออบเจ็กต์ โดยความสัมพันธ์ระหว่างออบเจ็กต์ที่สำคัญในแผนภาพออบเจ็กต์นั้นประกอบด้วย

2.1.3.3.1. ความสัมพันธ์แบบมีส่วนร่วม (Association)

เป็นความสัมพันธ์ระหว่างออบเจ็กต์ที่เกิดขึ้นเนื่องจากออบเจ็กต์ 2 ออบเจ็กต์มีความเกี่ยวข้องกัน ดังเช่นมีการเรียกใช้บริการ หรือมีการส่งแมซเสจถึงกันเป็นต้น

สำหรับสัญลักษณ์ที่ใช้แทนความสัมพันธ์ระหว่างออบเจ็กต์แบบมีส่วนร่วมนั้นจะเป็นเส้นตรงที่ลากระหว่างออบเจ็กต์ที่สัมพันธ์กัน นอกจากนั้นยังสามารถที่จะแสดงทิศทางในการ

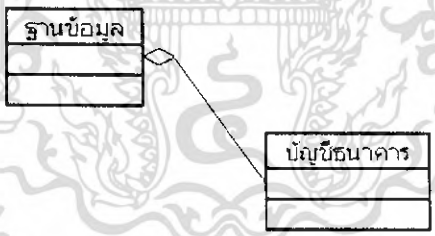
เรียกใช้บริการหรือส่งแมสเซจได้ด้วยการเติมหัวลูกศร โดยออบเจ็กต์ต้นทางจะเป็นผู้เรียกใช้บริการหรือเป็นผู้ส่งแมสเซจไปให้ ดังรูป 2.9.

รูปที่ 2.9. ตัวอย่าง ความสัมพันธ์แบบมีส่วนร่วม (Association)

2.1.3.3.2. ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)

เป็นความสัมพันธ์ที่ออบเจ็กต์หนึ่งแสดงความเป็นเจ้าของกับอีกออบเจ็กต์หนึ่ง นั่นคือ นอกจากออบเจ็กต์หนึ่งที่มีความสัมพันธ์แบบเป็นส่วนหนึ่งนั้นจะมีการเรียกใช้บริการหรือมีการแลกเปลี่ยนแมสเซจกันแล้ว ออบเจ็กต์ที่เป็นเจ้าของจะมีหน้าที่ที่จะต้องรับผิดชอบในการสร้างออบเจ็กต์ที่เป็นส่วนประกอบอีกด้วย

สำหรับสัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบเป็นส่วนหนึ่งนั้นจะเป็นเส้นตรงที่ลากระหว่างออบเจ็กต์เช่นกัน แต่ปลายเส้นที่ออบเจ็กต์ที่เป็นเจ้าของจะมีสัญลักษณ์ของสี่เหลี่ยมปรากฏอยู่ด้วย เพื่อใช้ในการแสดงความเป็นเจ้าของ ดังรูป 2.10.



รูปที่ 2.10. ตัวอย่าง ความสัมพันธ์แบบเป็นส่วนหนึ่ง (Aggregation)

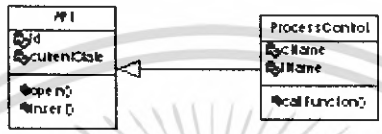
2.1.3.4 . แผนภาพคลาส (Class diagram)

เป็นแผนภาพคลาสนี้จะเป็นแผนภาพหลักสำหรับกระบวนการออกแบบและพัฒนา รวมถึงการทดสอบด้วย โดยแผนภาพคลาสนั้นประกอบไปด้วยคลาส และความสัมพันธ์ระหว่างคลาส สำหรับสัญลักษณ์ที่ใช้แทนคลาสนั้นจะเป็นสี่เหลี่ยม ซึ่งจะแบ่งออกเป็นสามส่วนคือ ชื่อของคลาส เมธอด (Method) ซึ่งเป็นส่วนที่จะบ่งบอกถึงพฤติกรรมของคลาส และแอททริบิวต์ (Attribute) ซึ่งจะบ่งบอกถึงข้อมูลที่เก็บอยู่ในคลาส และความสัมพันธ์ของคลาสนั้นก็จะคล้ายกับความสัมพันธ์ของออบเจ็กต์ โดยความสัมพันธ์ระหว่างคลาสที่มีความสัมพันธ์คือ

- ความสัมพันธ์แบบมีส่วนร่วม (Association)

- ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)
- ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Generalization หรือ Inheritance)

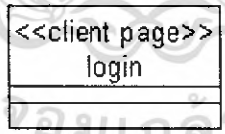
โดยความสัมพันธ์แบบสืบทอดคุณสมบัตินั้นเป็นความสัมพันธ์ที่เกิดขึ้นเมื่อมีคลาสหนึ่งสืบทอด โดยเราจะเรียกคลาสที่ให้การสืบทอดคุณสมบัติว่า ซูเปอร์คลาส (Super class) และเรียกคลาสที่ทำการสืบทอดคุณสมบัติมาว่าซับคลาส (Sub class) เนื่องจากซับคลาสนั้นจะสืบทอดคุณสมบัติจาก ซูเปอร์คลาส และจะส่งผลให้ซับคลาสนั้นมีคุณสมบัติทุกอย่างเหมือนกับซูเปอร์คลาส และยังสามารถที่จะทำการเพิ่มคุณสมบัติเข้าไปได้ตามที่ต้องการอีกด้วย ดังรูป 2.11.



รูปที่ 2.11. ตัวอย่างการกำหนดความสัมพันธ์แบบสืบทอดคุณสมบัติเพื่อเป็นการเจาะจงซับคลาสให้มีความเฉพาะมากขึ้น

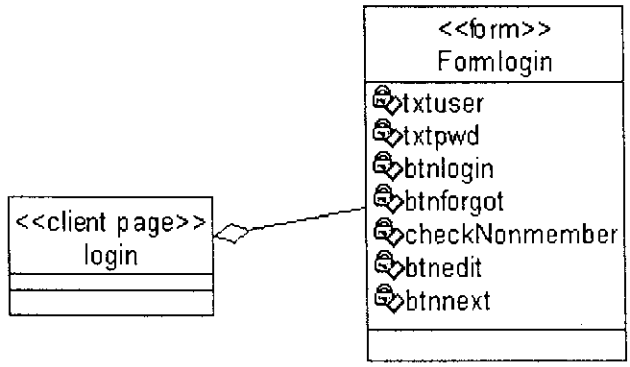
แต่ในการออกแบบแผนภาพคลาสของเว็บแอพพลิเคชันนั้น จะมีคลาสต่างๆที่เกี่ยวข้องกับเว็บแอพพลิเคชัน คือ

- **ไคลเอ็นต์เพจ (client page)** เป็นคลาสที่ใช้แทนเพจฝั่งไคลแอนต์ซึ่งโดยทั่วไปจะสืบทอดมาจากบราวเซอร์คลาสในขั้นตอนของการวิเคราะห์ โดยไคลเอ็นต์เพจนี้สามารถที่จะลิงค์ไปยังไคลเอ็นต์เพจอื่นๆ หรือเว็บเซิร์ฟเวอร์อื่นๆได้ สามารถมีฟอร์ม หรือสคริปและออบเจ็กต์ที่ฝั่งไคลเอ็นต์ ดังรูป 2.12.



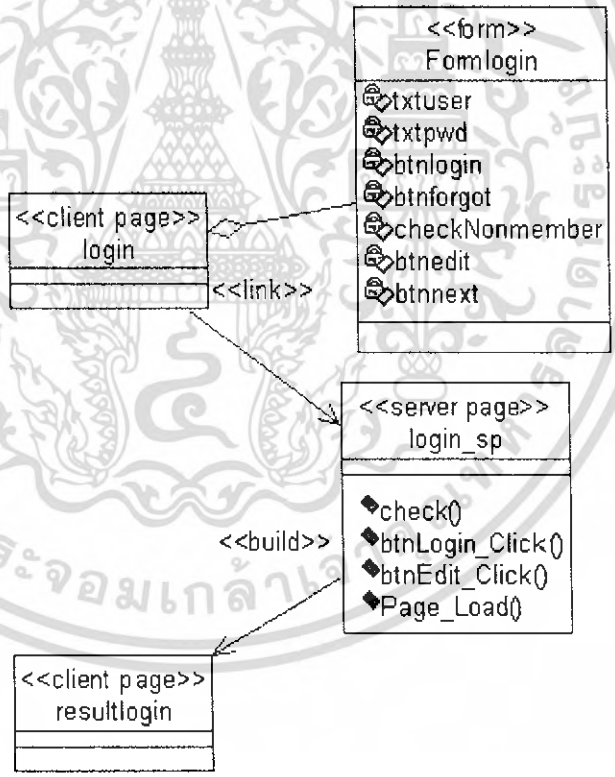
รูปที่2.12. ตัวอย่างไคลเอ็นต์เพจ

- **ฟอร์ม(Form)** โดยทั่วไปคลาสฟอร์มจะประกอบเป็นส่วนหนึ่งของไคลเอ็นต์เพจ โดยคลาสฟอร์มนั้นจะใช้เพื่อรับข้อมูลของผู้ใช้ ซึ่งภายในฟอร์มจะมีออบเจ็กต์สำหรับรับข้อมูลเช่น ฟิลด์และปุ่มซึ่งใช้ในการส่งข้อมูลและเรียกให้เซิร์ฟเวอร์เพจทำงานได้ ดังรูป 2.13.



รูปที่2.13. ตัวอย่างฟอร์ม

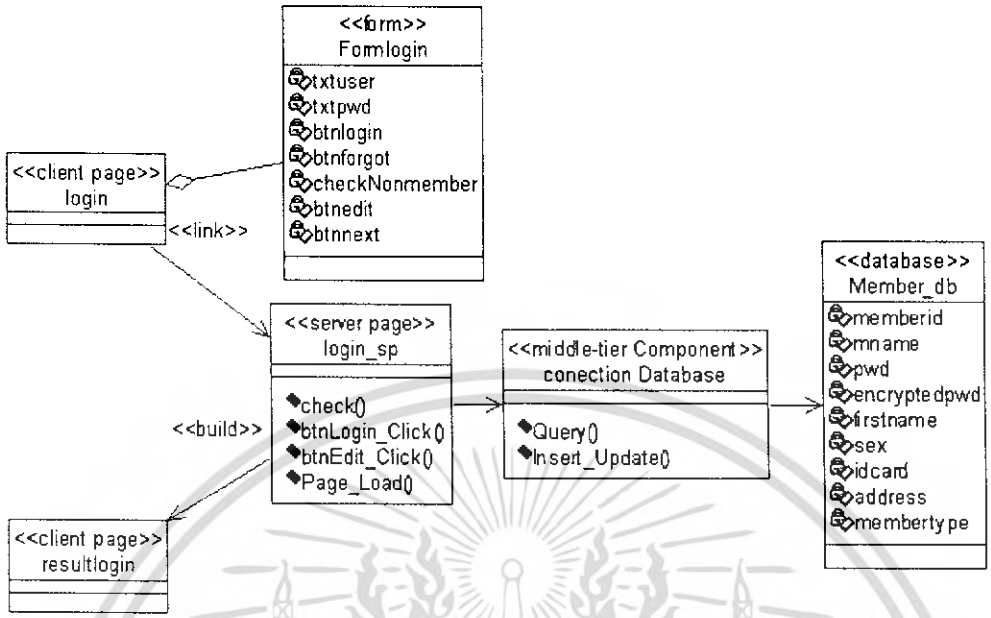
- เซิร์ฟเวอร์เพจ (server page) ซึ่งในการกำหนดเซิร์ฟเวอร์เพจนั้น เราสามารถที่จะกำหนดเซิร์ฟเวอร์เพจเป็นคลาสๆหนึ่งในเว็บแอพพลิเคชันได้ โดยคลาสเซิร์ฟเวอร์เพจนี้จะมีความสัมพันธ์กับไคลเอ็นต์ในกรณีที่เรียกใช้งาน และอาจจะมีการไปสร้างคลาสที่เป็นคอมโพแนนท์ที่เทียร์กลาง ดังรูป 2.14.



รูปที่2.14. ตัวอย่างเซิร์ฟเวอร์เพจ

- คลาสคอมโพแนนท์ซึ่งทำงานอยู่ที่เทียร์กลาง (Middle-Tier Component) เป็นคลาสที่ใช้ในการสร้างออบเจ็กต์เมื่อเซิร์ฟเวอร์เพจถูกเรียกใช้งานเพื่อประมวลผล หรือเมื่อต้องการติดต่อกับฐานข้อมูล ซึ่งการนำเอาคลาสคอมโพแนนท์ซึ่งทำงานอยู่ที่เทียร์กลาง (Middle-Tier-Component) นำไปใช้ในการออกแบบเว็บแอพพลิเคชันนี้จะช่วยให้ผู้พัฒนาสามารถที่จะทำการเปลี่ยนแปลงคลาสคอมโพแนนท์ที่ทำงานที่เทียร์กลางนี้ได้โดยจะมีผลกระทบกับเซิร์ฟเวอร์เพจ

นั้นน้อยมาก ซึ่งจะเหมาะกับการทำเว็บแอพพลิเคชันที่คำนึงถึงหลักการของการบำรุงรักษา (Maintenance) ดังรูป 2.15.

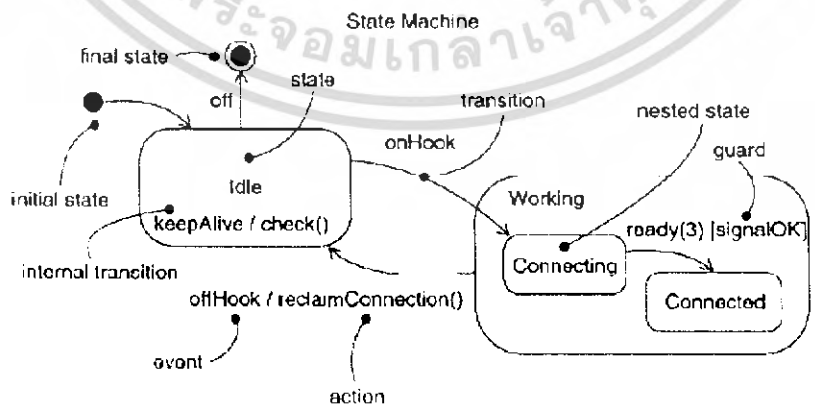


รูปที่2.15 ตัวอย่างคลาสคอมโพเนนท์ซึ่งทำงานอยู่ที่เทียร์กลาง

2.1.3.5. แผนภาพสถานะการทำงาน (State Diagram)

เป็นแผนภาพที่ใช้ในการอธิบายพฤติกรรมของซอฟต์แวร์ ส่วนประกอบภายใน หรือแม้แต่คลาสในแผนภาพของภาษายูเอ็มแอล โดยแผนภาพสถานะนี้จะเน้นที่การนำเสนอการทำงานภายใต้สถานะการทำงานของซอฟต์แวร์ทั้งหมดในทุกกรณีที่มี

สำหรับส่วนประกอบในแผนภาพสถานะนั้นประกอบไปด้วย จุดเริ่มต้น จุดจบ และสถานะ เส้นทางการเปลี่ยนแปลงสถานะตามเหตุการณ์ที่มีเข้ามาและการกำหนดเงื่อนไขสำหรับการเปลี่ยนสถานะ ดังรูป 2.16.



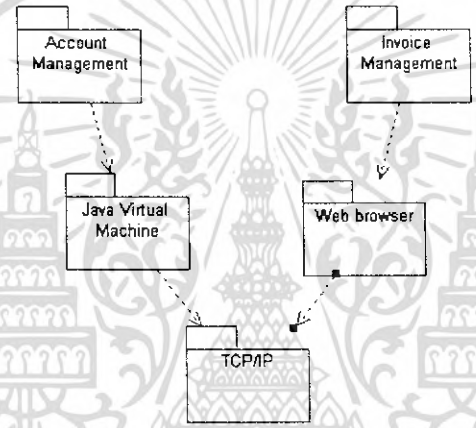
รูปที่2.16. ตัวอย่างแผนภาพสถานะ

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.1.3.5. แผนภาพคอมโพแนนท์ (Component diagram)

เป็นแผนภาพที่ใช้ในการอธิบายส่วนประกอบที่จะมีในระบบโดยเฉพาะส่วนของซอฟต์แวร์ สำหรับคอมโพแนนท์ในระบบนั้นจะหมายถึงระบบย่อยหรือแพกเกจ เราสามารถกล่าวได้ว่าคอมโพแนนท์นั้นเกิดขึ้นจากการรวมคลาสหลายๆคลาสที่มีหน้าที่หรือความรับผิดชอบคล้ายกันมาอยู่รวมกันในคอมแนนท์เดียวกันเพื่อให้ง่ายต่อการจัดการและดูแล

แผนภาพคอมโพแนนท์นอกจากจะใช้ในการแสดงโครงสร้างและส่วนประกอบของซอฟต์แวร์แล้ว ยังใช้ในการนำเสนอสถาปัตยกรรมของซอฟต์แวร์ด้วย ทั้งนี้เนื่องจากแผนภาพคอมโพแนนท์นี้จะมีการนำเสนอส่วนประกอบของซอฟร์แวร์ในระดับสูงขึ้นมา ทำให้ผู้พัฒนาสามารถทำความเข้าใจกับสถาปัตยกรรมของซอฟต์แวร์ได้ง่ายกว่า ดังรูป 2.17.

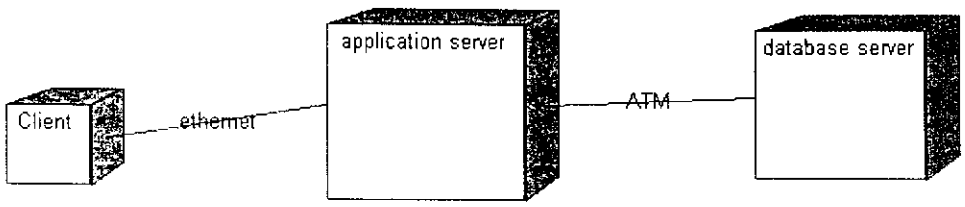


รูปที่ 2.17. ตัวอย่างการออกแบบสถาปัตยกรรมของซอฟต์แวร์เป็นแบบระดับชั้น

2.1.3.6. แผนภาพการติดตั้งใช้งาน (Deployment diagram)

เป็นแผนภาพที่นำเสนอส่วนประกอบของระบบในส่วนที่เป็นฮาร์ดแวร์ รวมถึงการเชื่อมต่อระหว่างกันด้วย สำหรับส่วนประกอบที่สำคัญของแผนภาพการติดตั้งใช้งานนั้นก็คือหน่วยประมวลผลหรือโนด (node) และการเชื่อมต่อระหว่างโนด สำหรับโนดนั้นจะเป็นที่บ่งบอกถึง อุปกรณ์ที่มีความสามารถในการประมวลผลที่มีอยู่ในระบบ ไม่ว่าจะเป็นคอมพิวเตอร์เซิร์ฟเวอร์ และอาจจะรวมถึงอุปกรณ์ต่อพ่วง ดังเช่น พรินเตอร์ หรือสแกนเนอร์ ดังรูป 2.18.

62376

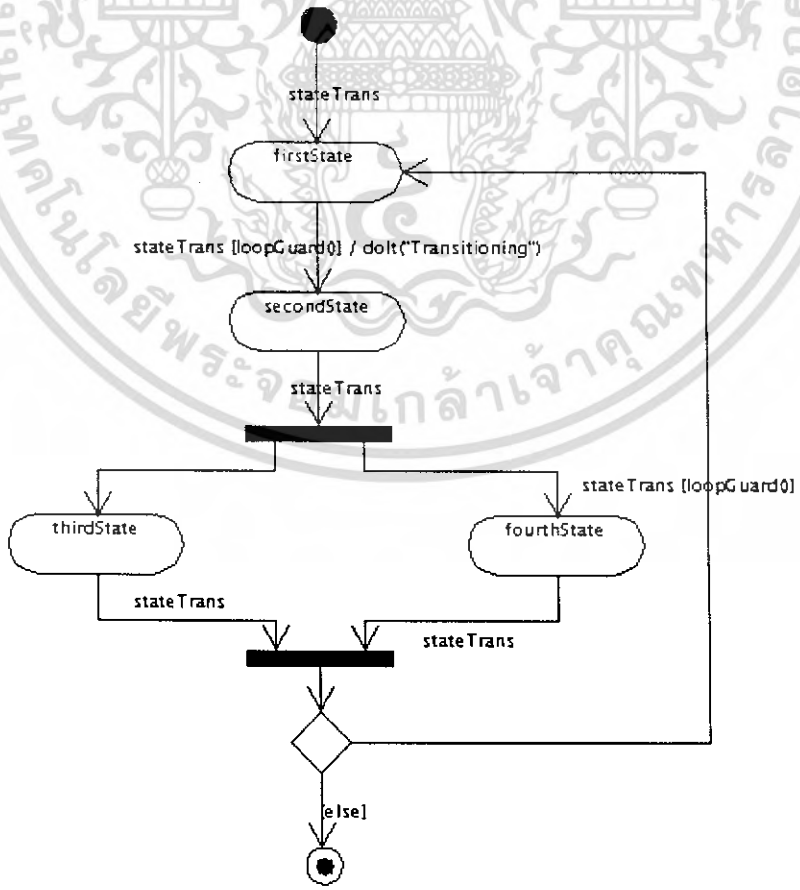


รูปที่ 2.18. ตัวอย่างแผนภาพการติดตั้งการใช้งาน

2.1.3.7. แผนภาพกิจกรรม (Activity diagram)

เป็นแผนภาพที่ใช้ในการแสดงพฤติกรรมของซอฟต์แวร์ในแง่ของลำดับการทำงาน เราอาจจะกล่าวได้ว่าแผนภาพกิจกรรมในโมเดลยูเอ็มแอลนี้เป็นแผนภาพโฟลว์ชาร์ต(Flow chart) แบบหนึ่ง ที่ได้มีการขยายเพิ่มคุณสมบัติขึ้นเพื่อให้เหมาะสมกับการโมเดลซอฟต์แวร์ในปัจจุบัน

โดยส่วนประกอบที่สำคัญในแผนภาพกิจกรรมนั้นก็คือ จุดเริ่มต้น, จุดจบ, กิจกรรม ,เงื่อนไข และเส้นทางแสดงลำดับการทำงาน สำหรับการสร้างแผนภาพกิจกรรมนั้นจะคล้ายกับแผนภาพโฟลว์ชาร์ตมาก ดังนั้นผู้ที่มีคุ้นเคยกับแผนภาพโฟลว์ชาร์ตอยู่แล้วก็สามารถวาดแผนภาพกิจกรรมได้โดยใช้หลักการเดียวกัน ดังรูป 2.19.



รูปที่ 2.19. ตัวอย่างแผนภาพกิจกรรม

2.1.4. กระบวนการทางวิศวกรรมซอฟต์แวร์อาร์ยูพี (Rational Unified Process)

เป็นกระบวนการทางวิศวกรรมซอฟร์แวร์ที่พัฒนาโดยใช้ยูสเคส เป็นตัวขับเคลื่อน (Use Case Driven Development Approach) เป็นการพัฒนาโดยมีสถาปัตยกรรมเป็นศูนย์กลาง (Architecture Centric Development Approach) และมีการพัฒนาเป็นรอบและการพัฒนาแบบก้าวหน้า (Iterative and Incremental)ซึ่งเป็นคุณสมบัติที่ดีในการพัฒนาซอฟต์แวร์ โดยขั้นตอนการทำงานหลักๆนั้นจะมีดังต่อไปนี้คือ

- ขั้นตอนการทำงานสำหรับการโมเดลทางธุรกิจ (Business modeling workflow)
- ขั้นตอนการทำงานที่เกียวข้องกับความต้องการ (Requirements workflow)
- ขั้นตอนการทำงานสำหรับการวิเคราะห์และออกแบบ (Analysis & Design workflow)
- ขั้นตอนการทำงาน ในการพัฒนาและสร้าง(Implementation workflow)
- ขั้นตอนการทำงานในการทดสอบ (Test workflow)
- ขั้นตอนการทำงานในการติดตั้งและใช้งาน (Deployment workflow)

2.1.4.1. ขั้นตอนการทำงานสำหรับการโมเดลทางธุรกิจ (Business modeling workflow)

คือการกำหนดขั้นตอนการทำงานร่วมกันระหว่างทีมพัฒนาทางธุรกิจกับซอฟต์แวร์ เพื่อให้เกิดความสอดคล้องในการพัฒนาและตรงกับความต้องการในการดำเนินธุรกิจมากที่สุด โดยในขั้นตอนการทำงานของโมเดลธุรกิจนั้นจะเกี่ยวข้องกับการจัดทำเอกสารโดยใช้สิ่งที่เรียกว่ายูสเคสทางธุรกิจ (Business Use case) ซึ่งโมเดลนี้จะช่วยเพิ่มความเข้าใจระหว่างลูกค้ากับทีมพัฒนาในกระบวนการทางธุรกิจที่จำเป็นต้องได้รับการสร้างขึ้นเป็นซอฟต์แวร์ให้ดีขึ้น ส่วนยูสเคสทางธุรกิจจะถูกวิเคราะห์เพื่อให้ทีมพัฒนาเข้าใจว่าซอฟต์แวร์จะถูกนำไปใช้เพื่อการสนับสนุนการทำงานในธุรกิจอย่างไรซึ่งเราเรียกเอกสารนี้ว่าออบเจ็กต์โมเดลสำหรับธุรกิจ (Business Object-Model) อย่างไรก็ตามไม่จำเป็นที่โครงสร้างการพัฒนาซอฟต์แวร์ทุกโครงการจะต้องดำเนินการขั้นตอนการทำงานเพื่อสร้างโมเดลทางธุรกิจ โดยขึ้นอยู่กับรูปแบบของซอฟต์แวร์ และองค์กรที่จะใช้งานซอฟต์แวร์

2.1.4.2. ขั้นตอนการทำงานที่เกี่ยวข้องกับความต้องการ (Requirements workflow)

เป็นขั้นตอนที่จะช่วยให้ทีมพัฒนารู้ว่าอะไรคือสิ่งที่ระบบซอฟต์แวร์ต้องทำ อีกทั้งยังช่วยให้ทีมพัฒนากับลูกค้าสามารถทำความตกลงในรายละเอียดของซอฟต์แวร์ที่จะพัฒนาขึ้น ในขั้นตอนนี้ทีมพัฒนาจะต้องค้นหา จัดโครงสร้าง และบันทึกความต้องการที่เป็นฟังก์ชันการทำงานและเงื่อนไขต่างๆ ไว้เป็นเอกสารเพื่อการติดตามและการนำกลับมาใช้เมื่อจำเป็น

ในขั้นตอนเกี่ยวกับความต้องการ จะมีการสร้างเอกสารวิสัยทัศน์ (Vision document) ขึ้น มีการกำหนดแอคเตอร์ซึ่งจะหมายถึงผู้ใช้ซอฟต์แวร์และระบบอื่นๆ ที่จะต้องติดต่อซอฟต์แวร์ที่จะพัฒนาขึ้น มีการกำหนดยูสเคสซึ่งจะนำเสนอถึงพฤติกรรมที่ซอฟต์แวร์จะทำงาน

หลังจากที่ได้ทำการกำหนดยูสเคสแล้วขั้นต่อมาทีมพฒนาจะทำการกำหนดรายละเอียดของแต่ละยูสเคส คำอธิบายยูสเคสจะแสดงให้เห็นว่าซอฟต์แวร์จะติดต่อกับแอคเตอร์อย่างไรทีละขั้นตอน ส่วนความต้องการที่มาได้เป็นฟังก์ชันการทำงานนั้นจะได้รับการกำหนดและบันทึกไว้ด้วย

2.1.4.3. ขั้นตอนการทำงานสำหรับการวิเคราะห์และออกแบบ (Analysis & Design workflow)

วัตถุประสงค์ของขั้นตอนการทำงานสำหรับการวิเคราะห์และออกแบบนั้นก็เพื่อแสดงให้เห็นว่าซอฟต์แวร์จะมีการทำงานอย่างไรซึ่งรายละเอียดและข้อมูลที่ได้ถูกนำไปใช้ในขั้นตอนของการสร้างต่อไป และผลของขั้นตอนในการวิเคราะห์และออกแบบนี้จะได้เป็นโมเดล ซึ่งเรียกว่าโมเดลการออกแบบที่จะเป็นต้นแบบในการเขียนโค้ด

โมเดลการออกแบบนั้นประกอบไปด้วยคลาสที่ได้มีการจัดโครงสร้างให้อยู่ร่วมกันเป็นแพกเกจ (Package) หรือระบบย่อย (Subsystem) และกิจกรรมในการออกแบบนั้นก็จะเน้นที่แผนภาพสัญลักษณ์ที่จะแสดงถึงสถาปัตยกรรมของซอฟต์แวร์ ซึ่งจากสถาปัตยกรรมนี้จะทำให้ทีมพัฒนาสามารถมองเห็นรายละเอียดที่จะต้องเพิ่มหรือเสริมขึ้นมาเพื่อให้ซอฟต์แวร์สามารถทำงานได้ตามที่ผู้ใช้ต้องการ

2.1.4.3. ขั้นตอนการทำงาน ในการพัฒนาและสร้าง (Implementation workflow)

วัตถุประสงค์ของขั้นตอนการทำงาน ในการพัฒนาและสร้างนั้น ก็เพื่อการกำหนดโครงสร้างของโค้ด ในมุมมองของการสร้างระบบย่อยในระดับต่างๆของซอฟต์แวร์ , เพื่อสร้างคลาสและออบเจ็กต์ของ คอมโพแนนท์, เพื่อทดสอบคอมโพแนนท์ในระดับหน่อยย่อย, เพื่อทำการรวมส่วนประกอบที่ได้รับการพัฒนาของนักพัฒนาแต่ละคนหรือแต่ละทีมเข้ามาเป็นซอฟต์แวร์ที่สามารถทำงานได้

ในอาร์ยูพี(Ration Unified Process) คอมโพแนนท์จะถูกกำหนดโครงสร้างให้อยู่ร่วมกันในระบบย่อยของการพัฒนาและสร้าง (Implementation Subsystem) ในการบริหารโครงการระดับย่อยที่ถูกสร้างขึ้นเหล่านี้ควรจะถูกเก็บโดยโครงสร้างไดเร็กทอรีทั้งนี้เพื่อให้มีการจัดโครงสร้างที่ง่ายต่อการใช้งานและแก้ไขรวมทั้งยังสามารถใช้ในการเก็บข้อมูลที่มีความสำคัญต่อการจัดการได้ด้วย

2.1.4.3. ขั้นตอนการทำงานในการทดสอบ (Test workflow)

เนื่องจากกระบวนการ อาร์ยูพี(Ration Unified Process) นั้นจะเป็นกระบวนการที่มีการพัฒนาเป็นรอบๆ เพราะฉะนั้นในแต่ละช่วงของการพัฒนาก็จะถูกทำการทดสอบอยู่เป็นระยะ โดยในกระบวนการของการทดสอบนั้น จะประกอบไปด้วย การวางแผน การออกแบบ การสร้างและการลงมือทดสอบและประเมินผล นอกจากนี้แล้วในกระบวนการยังอธิบายว่าทีมพัฒนาจะใช้

เครื่องมือในการทดสอบซอฟต์แวร์เมื่อไหร่และอย่างไร ซึ่งเครื่องมือในการทดสอบเป็นสิ่งสำคัญในกระบวนการพัฒนาแบบเป็นรอบ

2.1.4.4 ขั้นตอนการทำงานในการติดตั้งและใช้งาน (Deployment workflow)

วัตถุประสงค์ของขั้นตอนการทำงานเพื่อติดตั้งและใช้งานนั้นก็เพื่อทำให้ผลิตภัณฑ์ซอฟต์แวร์สามารถทำงานได้ถูกต้องและผู้ใช้พึงพอใจ โดยในขั้นตอนการติดตั้งและการใช้งานนี้จะประกอบด้วยกิจกรรมย่อยต่างๆ เช่น การเปิดตัวซอฟต์แวร์ต่อผู้ใช้ที่อยู่ภายนอกทีมพัฒนา, การจัดหน้าของตัวซอฟต์แวร์เพื่อเตรียมส่งมอบ, การเผยแพร่ซอฟต์แวร์ การให้ความช่วยเหลือต่อผู้ใช้ในการใช้งานซอฟต์แวร์ บางครั้งขั้นตอนของการทดสอบจะประกอบไปด้วยกิจกรรมในการวางแผนและทดสอบซอฟต์แวร์ในขั้นเบต้า , การทำการย้ายซอฟต์แวร์และข้อมูลที่มีอยู่เพื่อให้พร้อมกับซอฟต์แวร์ใหม่ หรือทำการตกลงเพื่อรับมอบระบบจากลูกค้า

2.1.5 รูปแบบสถาปัตยกรรมของเว็บแอพพลิเคชัน (Web Application Architecture Pattern)

ในการพัฒนาซอฟต์แวร์ทั่วไปนั้น ผู้พัฒนามีทางเลือกในการกำหนดและเลือกสถาปัตยกรรมสำหรับซอฟต์แวร์ได้อย่างหลากหลาย เช่น รูปแบบสถาปัตยกรรมสำหรับควบคุม รูปแบบสถาปัตยกรรมความปลอดภัย แต่สำหรับการพัฒนาเว็บแอพพลิเคชันนั้น ทางเลือกของสถาปัตยกรรมที่จะใช้นั้นจะขึ้นอยู่กับเทคโนโลยีและผลิตภัณฑ์ที่ใช้ และด้วยคุณสมบัติของเว็บแอพพลิเคชันที่ให้ผู้ใช้สามารถประมวลผลและแลกเปลี่ยนข้อมูลกับเว็บเซิร์ฟเวอร์ได้ ดังนั้นเราจึงสามารถที่จะกำหนดส่วนประกอบที่สำคัญของเว็บแอพพลิเคชันได้ คือ โปรแกรมเว็บบราวเซอร์ที่ไคลเอ็นต์ เว็บเซิร์ฟเวอร์ แอพพลิเคชันเซิร์ฟเวอร์ และอาจจะประกอบด้วย เซิร์ฟเวอร์ฐานข้อมูลเพิ่มขึ้นมาในระบบงานที่เกี่ยวข้องกับการเก็บข้อมูลเป็นจำนวนมาก

โดยรูปแบบของสถาปัตยกรรมที่เหมาะสำหรับการประยุกต์ใช้ในเว็บแอพพลิเคชันนั้นสามารถแบ่งออกได้เป็น 3 รูปแบบ คือ ทินเว็บไคลเอ็นต์ (Thin Web Client) ทริกเว็บไคลเอ็นต์ (Thick Web Client) และเว็บเดริเวอร์รี ไคลเอ็นต์ (Web Delivery Client) โดยรูปแบบสถาปัตยกรรมทั้ง 3 รูปแบบนี้จะเกี่ยวข้องกับตำแหน่งของส่วนประกอบที่เกี่ยวข้องกับการประมวลผลเป็นหลัก ซึ่งรูปแบบแต่ละแบบก็มีข้อดีข้อเสียต่างกันไป

2.1.5.1 รูปแบบสถาปัตยกรรมทินเว็บไคลเอ็นต์ (Thin Web Client)

เป็นรูปแบบสถาปัตยกรรมเว็บแอพพลิเคชันที่ส่วนใหญ่เลือก ทั้งนี้เป็นรูปแบบสถาปัตยกรรมที่ไคลเอ็นต์ไม่จำเป็นต้องมีความสามารถมากนัก เนื่องจากผู้ใช้จะใช้ไคลเอ็นต์เป็นเพียงช่องทางสำหรับรับและส่งข้อมูลเท่านั้น ส่วนการประมวลผลทั้งหมดจะทำที่เซิร์ฟเวอร์

2.1.5.2 รูปแบบสถาปัตยกรรมทริกเว็บไคลเอ็นต์ (Thick Web Client)

เป็นรูปแบบสถาปัตยกรรมที่เริ่มมีการประมวลผลบางส่วนกระทำที่ฝั่งไคลเอ็นต์ ทั้งนี้ไคลเอ็นต์จะต้องมีความสามารถในการรันภาษาสคริปหรือออบเจ็กต์ของจาวา และ แอคทีฟเอ็ก (ActiveX) สำหรับการสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์นั้นยังคงทำผ่านโพรโตคอลเอชทีทีพี (HTTP) ปกติ

2.1.5.3 รูปแบบสถาปัตยกรรมเว็บเดริเวอร์รี ไคลเอ็นต์ (Web Delivery Client)

เป็นรูปแบบสถาปัตยกรรมที่มีการใช้โพรโตคอลเอชทีทีพี เพียงการสื่อสารในครั้งแรกเท่านั้นแต่การสื่อสารหลังจากนั้นที่เกี่ยวข้องกับการประมวลผลอาจจะกระทำผ่านโพรโตคอลอื่น เช่น ไอไอโอพี (IIOP) หรือ อาร์เอ็มไอ (RMI) ของจาวา รูปแบบการทำงานแบบนี้จะเป็นการทำงานของออบเจ็กต์แบบกระจาย โดยโปรแกรมเว็บบราวเซอร์จะทำหน้าที่เป็นช่องทางเพื่อให้ออบเจ็กต์ทำงานเท่านั้น

2.2. เทคโนโลยีดอทเน็ต (.NET)

2.2.1. ภาพรวมของเทคโนโลยีดอทเน็ต (.NET)

ดอทเน็ต (.NET) เป็นกลยุทธ์ของทางบริษัทไมโครซอร์ฟ (Microsoft) ที่คิดจะทำการเชื่อมต่อข้อมูล, ผู้คน, ระบบและอุปกรณ์ผ่านซอฟต์แวร์เว็บเซอร์วิส (Web Service) ทำให้ทีมงานสามารถสื่อสารและทำงานร่วมกันได้อย่างมีประสิทธิภาพ

เทคโนโลยีดอทเน็ต .NET ถูกรวมอยู่ในผลิตภัณฑ์ทั้งหมดของไมโครซอร์ฟ (Microsoft) เพื่อเพิ่มความสามารถในการสร้าง, การจัดการ และติดต่อได้อย่างรวดเร็ว เพิ่มแนวทางด้านความปลอดภัย (security-enhanced solutions) ด้วยเว็บเซอร์วิส Web Service

แนวทางการเชื่อมต่อดอทเน็ต (.NET-connected solutions) เป็นการรวมระบบธุรกิจให้สามารถทำงานได้เร็วขึ้น มีความว่องไว กระฉับกระเฉง และช่วยให้เรารู้ถึงการเปลี่ยนแปลงของข้อมูลได้ทุกที่ ทุกเวลา และทุกอุปกรณ์

2.2.2. ประโยชน์ของเทคโนโลยีดอทเน็ต(.NET)

2.2.2.1. ประโยชน์ทางด้านองค์กร (.NET benefits organizations)

ดอทเน็ต (.NET) จะลบข้อจำกัดในเรื่องโพรไพรอิทาลี่ เทคโนโลยี (proprietary technology) ทำให้มีความยืดหยุ่นสามารถช่วยขยายการติดต่อระหว่างระบบ IT ที่มีในปัจจุบันเป็นการสร้างคลื่นลูกใหม่ของเทคโนโลยีคอมพิวเตอร์ดอทเน็ต .NET และเว็บเซอร์วิส (Web services) ช่วยองค์กรลดค่าใช้จ่ายในการติดต่อกับระบบ เพิ่มยอดขายโดยช่วยให้พนักงานเข้าถึง

ข้อมูลที่ถูกต้องในเวลาและสถานที่ที่เขาต้องการ ช่วยให้นักพัฒนาสร้างโซลูชัน (solution) ทางธุรกิจได้อย่างรวดเร็วและมีค่าใช้จ่ายต่ำ

2.2.2.2. ประโยชน์ในส่วนบุคคล (.NET benefits individuals)

ช่วยให้ได้รับข้อมูลตามความต้องการของแต่ละบุคคลมากขึ้น ซึ่งสามารถรองรับฮาร์ดแวร์ได้หลากหลาย เช่น Pocket PCs, Smart phones, laptops เป็นต้น ผู้ใช้โมบาย คอมพิวติ้ง (mobile computing) สามารถรับข้อมูลได้ทุกๆ ที่

2.2.2.3. ประโยชน์ในด้านการพัฒนา (.NET benefits developers)

มีการรวมเอากลุ่มของ tools เพื่อใช้ในการสร้าง Web services ด้วย Visual Studio .NET และ .NET Framework ซึ่งพัฒนาได้ในหลาย platform เป็นหนทางช่วยผู้พัฒนาให้พัฒนา Web service ได้ดี , รวดเร็วและมีประสิทธิภาพมากที่สุด

2.2.3. ผลิตภัณฑ์และเทคโนโลยีของ Microsoft ในปัจจุบันที่มีการใช้ .NET

2.2.3.1. .NET Passport

เป็น Web-based service ที่ออกแบบมาเพื่อใช้สำหรับการล็อกอิน (login) เข้าสู่เว็บไซต์ต่างๆ ได้อย่างง่ายดายและรวดเร็ว โดยตัวพาสปอร์ต (passport) นี้ ทำให้เว็บไซต์ที่รองรับ .NET Passport สามารถให้สิทธิ์แก่ผู้ใช้จาก single set of sign-in credentials ทำให้ผู้ใช้ไม่จำเป็นต้องจดจำชื่อผู้ใช้ (username) และรหัสผ่าน (password) จำนวนมาก

ไมโครซอร์ฟดอทเน็ต (Microsoft .NET) มีทุกๆ สิ่งที่จำเป็นในการพัฒนา service-based IT architecture:

- servers to host Web services
- development tools เพื่อใช้สร้าง Web services
- applications เพื่อที่จะเรียกใช้เว็บเซอร์วิส Web services เทคโนโลยีดอทเน็ต .NET

มีการรองรับไว้ในผลิตภัณฑ์ตระกูลไมโครซอร์ฟ (Microsoft) ทั้งหมด เช่น Windows Server System, Windows XP desktop operating system และ Microsoft Office System

2.2.3.2. ตัวอย่างองค์กรที่มีการใช้เทคโนโลยี .NET

องค์กรทั่วโลกมีการนำเทคโนโลยีดอทเน็ต (.NET) และเว็บเซอร์วิส (Web service) ไปใช้เพื่อช่วยเชื่อมต่อธุรกิจและทำให้การสื่อสาร การทำงานร่วมกันมีประสิทธิภาพมากยิ่งขึ้น ตัวอย่างเช่น Honeywell, GlaxoSmithKline, Sony, Dollar Rent A Car, Farmers Insurance และ the United States Postal Service ที่เป็นองค์กรที่มีการใช้เทคโนโลยี .NET ในปัจจุบัน

2.3. เทคโนโลยีเว็บเซอร์วิส (Web Service)

2.3.1. ภาพรวมของเทคโนโลยี Web Service

เว็บเซอร์วิส (Web Service) คือ วิธีการนำกลับมาใช้ของแอพพลิเคชัน (reusable application) ที่ช่วยให้แอพพลิเคชัน ทำงานร่วมกันได้ด้วยการแลกเปลี่ยนเมสเซช (message) ผ่านทางโปรโตคอลมาตรฐานของการติดต่อกันในอินเทอร์เน็ต (standard internet communication protocol) เช่น

- XML (Extensible Markup Language)
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)

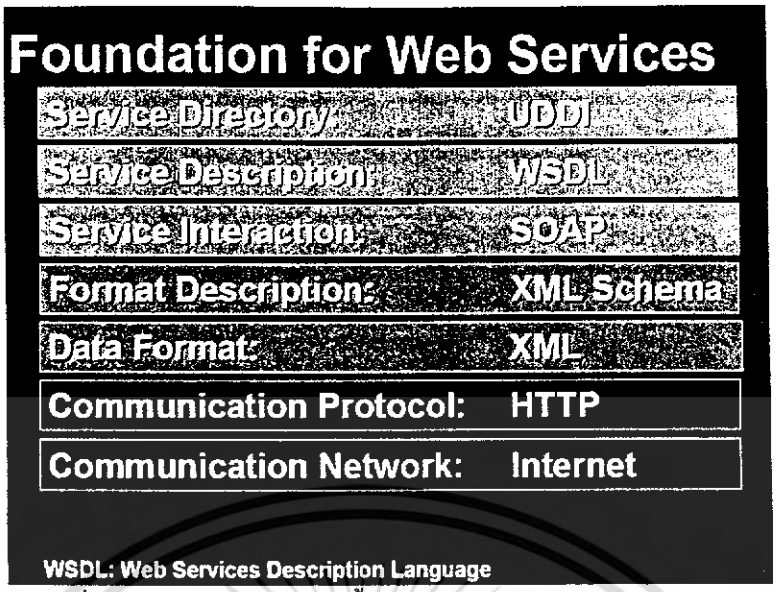
โดยโปรโตคอลเหล่านี้จะทำให้คอมพิวเตอร์สามารถทำงานร่วมกันข้ามภาษาและข้ามแพลตฟอร์มได้ เป็นเว็บแอพพลิเคชัน (Web application) ที่ไม่มียูสเซอร์อินเทอร์เฟส (user interface)

Web services เป็นการปฏิวัติวิธีการสื่อสารกันระหว่างแอพพลิเคชัน (applications) แลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ต (internet) โดยการใช้ universal data format ทำให้เว็บเซอร์วิส (Web services) สามารถสื่อสารข้ามแพลตฟอร์ม (platform) และระบบปฏิบัติการ และไม่ขึ้นกับภาษาที่ใช้เขียนแอพพลิเคชัน *(applications)*

แต่ละ เว็บเซอร์วิส (Web services) จะเป็น discrete unit of code ที่รองรับงานที่จำกัด แม้ว่า เว็บเซอร์วิส (Web services) จะเป็นอิสระต่อกัน แต่มันก็สามารถรวมตัวกันเป็นกลุ่มแบบหลวมๆ เพื่อที่จะทำงานเฉพาะอย่างได้

2.3.2. โครงสร้างขั้นพื้นฐานของเทคโนโลยีเว็บเซอร์วิส (Web services)

โครงสร้างพื้นฐานของเทคโนโลยี Web Service ประกอบไปด้วยส่วนต่างๆ ดังรูป 2.20.



รูปที่ 2.20. แสดงโครงสร้างพื้นฐานของเทคโนโลยี Web Service

2.3.2.1.เอ็กซ์เอ็มแอล XML (Extensible Markup Language)

เอ็กซ์เอ็มแอล (Extensible Markup Language) คือมาตรฐานภาษาอธิบายข้อมูลที่ใช้กันทั่วโลกสำหรับการใช้ข้อมูลทางธุรกิจร่วมกันบนเว็บ โดยไม่ต้องกังวลเรื่องโปรแกรม เครือข่ายคอมพิวเตอร์ โครงสร้างข้อมูลและระบบปฏิบัติการที่เข้ากันไม่ได้ โดยที่ เอ็กซ์เอ็มแอล (Extensible Markup Language) ทำให้สามารถใช้ข้อมูลร่วมกันได้ง่ายขึ้น

เอ็กซ์เอ็มแอล (Extensible Markup Language) ใช้แทก (tag) มาตรฐานในการระบุโครงสร้างและเนื้อหาของแฟ้ม การใช้แทกเอ็กซ์เอ็มแอล (tag XML) เหมือนกันในทุกแฟ้มจะช่วยให้คุณสามารถทำดัชนี ค้นหา รวม และนำข้อมูลที่ใช้ข้อความเป็นพื้นฐานมาใช้ใหม่ได้อย่างมีประสิทธิภาพ และเนื่องจากเอ็กซ์เอ็มแอล (Extensible Markup Language) ใช้ข้อความเป็นพื้นฐานและไม่จำกัดสำหรับภาษาคอมพิวเตอร์ส่วนตัวเท่านั้น จึงสามารถแลกเปลี่ยนข้อมูลระหว่างระบบที่โดยปกติไม่เข้ากันได้

2.3.2.2. เอสโอเอพี SOAP (Simple Object Access Protocol)

เอสโอเอพี (SOAP) คือ XML-based messaging technology เป็นมาตรฐานสำหรับเมสเซจของเอ็กซ์เอ็มแอล (XML messaging) ถูกพัฒนาให้เป็นมาตรฐานโดยดับบลิวทรีซี (W3C) ซึ่งกำหนด rule ที่สำคัญบนเว็บเซอร์วิส (Web services), วิธีการการรวม web service เข้าไปใน application และการสื่อสารระหว่างกัน

2.3.2.3. ดับบลิวเอสดีแอล WSDL (Web Services Description Language)

ดับบลิวเอสดีแอล (WSDL (Whiz-Dull)) คือ XML-based service description ที่อธิบายว่าหากต้องการใช้ เว็บเซอร์วิส (Web services) ต้องทำการติดต่ออย่างไร ต้องมีรูปแบบของเมสเซจ

(message formats) เป็นอย่างไร เป็นตัวอธิบาย public interface สำหรับเว็บเซอร์วิส (Web services) เมื่อไคลเอ็นต์ ทำการติดต่อมายังเว็บเซอร์วิส (Web services) สามารถอ่านดับบลิวเอสดีแอล (WSDL) เพื่อที่จะรับรู้ว่ามีฟังก์ชัน (function) ใดบ้างที่ให้บริการบน (server)

2.3.2.4. ยูดีดีไอ UDDI (Universal Description, Discovery, and Integration)

ยูดีดีไอUDDI เป็น public registry ไม่เสียค่าใช้จ่าย เป็นที่ๆ ผู้คนสามารถเผยแพร่และสอบถามข้อมูลของเว็บเซอร์วิส (Web services) ได้ อธิบายวิธีการมาตรฐานในการเก็บและรับข้อมูลที่เกี่ยวกับเว็บเซอร์วิส (Web services)

2.3.3. ลักษณะการเชื่อมต่อของเทคโนโลยีเว็บเซอร์วิส (Web services)

การเชื่อมต่อของเว็บเซอร์วิส (Web services) มีได้หลากหลายแบบ ดังรูป 2.21.

รูปที่ 2.21. แสดงลักษณะการเชื่อมต่อของเทคโนโลยี Web Service

2.3.3.1. ไคลเอ็นต์กับไคลเอ็นต์ (Client-to-client)

สมาร์ทไคลเอ็นต์ Smart clients สามารถเป็นโฮส (host) และประยุกต์ใช้เว็บเซอร์วิส (Web services) เพื่อทำการแชร์ข้อมูลได้ทุกที่ ทุกเวลา

สมาร์ทไคลเอ็นต์ (Smart client) คือไคลเอ็นต์แอพพลิเคชัน (client applications) ที่มีการใช้บริการเว็บเซอร์วิส (Web services) โดยจะอยู่บนฮาร์ดแวร์ของผู้ใช้ เช่น desktop PCs, laptops, Pocket PCs และ Smart phones มันสามารถทำการจัดการ ปรับใช้ได้ง่ายโดยการใช้ทรัพยากรบนอุปกรณ์ และสามารถติดต่อกับ distributed data sources ได้

สมาร์ทไคลเอ็นต์แอพพลิเคชัน (Smart client applications) สามารถทำงานได้ไม่ว่าจะมีการติดต่อกับอินเทอร์เน็ตหรือไม่ ตัวอย่างเช่น Microsoft Money , Microsoft Outlook ต่างกับ "browser-based" applications เช่น Amazon.Com หรือ eBay.com

2.3.3.2. ไคลเอ็นต์กับเซิร์ฟเวอร์ (Client-to-Server)

เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services) สามารถแชร์ข้อมูลจากเซอร์เวอร์แอพพลิเคชัน (server application) ไปยังเครื่องคอมพิวเตอร์ desktop หรือ mobile computing device ผ่านอินเทอร์เน็ตได้

2.3.3.3. เซิร์ฟเวอร์กับเซิร์ฟเวอร์ (Server-to-Server)

เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services) ให้บริการด้าน common interface ระหว่าง application ที่มีการรันอยู่ตามสภาพแวดล้อมของแต่ละ server

2.3.3.4. เซอร์วิสกับเซอร์วิส (Service-to-service)

เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services) สามารถทำงานร่วมกันได้เป็นลำดับเพื่อที่จะทำการสร้าง operation ที่ซับซ้อนขึ้น

2.3.4. เทคโนโลยีเซอร์วิส (Web services) ในมุมมองทางธุรกิจ

ผู้นำกลุ่มธุรกิจกล่าวว่าเซอร์วิส (Web services) คือวิธีที่ช่วยให้ธุรกิจติดต่อกับลูกค้า, พันธมิตรธุรกิจ, ผู้ผลิตและลูกจ้างได้อย่างมีประสิทธิภาพ เหมือนเป็นการปลดล็อคข้อมูลเพราะข้อมูลจะวิ่งไปได้ทุกๆ ที่ หากมีผู้ต้องการ ลดเวลาในการพัฒนาและค่าใช้จ่ายสำหรับโครงการใหม่

จากมุมมองทางธุรกิจเว็บซอร์วิส (Web services) ไม่เพียงแต่เชื่อมต่อระบบ แต่มันยังช่วยเชื่อมต่อผู้คนเข้ากับข้อมูลพวกเค้าต้องการ ด้วย software applications ที่พวกเค้าใช้อยู่

ผลิตภัณฑ์ที่ไมโครซอร์ฟ (Microsoft) ได้เสนอเพื่อช่วยให้องค์กรและแต่ละบุคคลได้รับประโยชน์จาก **Web service-based connectivity** มีตัวอย่างเช่น

- Microsoft Visual Studio .NET 2003 ซึ่งเป็น developer tools
- Windows Server System ซึ่งเป็น hosts Web services

และ desktop applications ที่เราคุ้นเคยอย่าง

- Microsoft Office System ซึ่งเป็นระบบที่จะเรียกใช้ Web services.

2.3.5. เปรียบเทียบเว็บไซต์กับเอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web service)

เว็บไซต์มักเกี่ยวข้องกับการนำเสนอข้อมูลให้กับผู้ใช้ โดยเป็นเครื่องมือในการสื่อสารสำหรับเซิร์ฟเวอร์ไปยังผู้ใช้ แต่ในทางตรงกันข้าม บริการเอ็กซ์เอ็มแอลเว็บ (XML Web) เป็นวิธีที่แอพพลิเคชันจะสามารถโต้ตอบกับแอพพลิเคชันอื่นได้โดยตรง ไม่ว่าจะเป็นแอพพลิเคชันที่อยู่บนเซิร์ฟเวอร์เดียวกัน หรืออยู่บนระบบที่ห่างไกลออกไป ก็สามารถสื่อสารกันได้โดยใช้อินเทอร์เน็ต และระบบข่าวสารเอ็กซ์เอ็มแอล (XML) และเอสโอเอพี (SOAP)

พิจารณาจากตัวอย่างง่ายๆ คือ หากคุณมีระบบสินค้าคงคลัง แต่ไม่ได้เชื่อมโยงเข้ากับระบบใดเลย ก็เท่ากับว่าคุณค่าของระบบนั้นมีไม่มาก แม้จะสามารถติดตามสินค้าคงคลังได้ แต่ต้องดำเนินการเป็นจำนวนมาก และสิ่งที่ระบบสามารถทำงานได้ด้วยตัวเองโดยไม่ต้องเชื่อมโยงกับระบบอื่นนั้นก็ค่อนข้างจำกัด เพราะทุกรายการสินค้าที่ขายไม่เพียงแต่จะต้องป้อนเข้าสู่ระบบสินค้าคงคลังเท่านั้น แต่ต้องกระจายไปสู่ระบบบัญชีและบันทึกในบัญชีของลูกค้าและยังจำเป็นต้องทำบันทึกเพื่อเตรียมสั่งสินค้าเพิ่มจากซับไพลเออร์ (supplier) อีกด้วย สัดส่วนระหว่างต้นทุนและประโยชน์ที่ได้รับนั้นดูแล้วอาจไม่น่าดึงดูดใจ เนื่องจากสิ่งที่ได้รับจากระบบนั้นใกล้เคียงกับค่าใช้จ่ายที่ต้องเสียไป

แต่ถ้าคุณทำการเชื่อมโยงระบบนี้เข้ากับระบบบัญชี ประโยชน์ที่ได้รับจะเพิ่มมากขึ้น เพราะเมื่อมีการซื้อ-ขายเกิดขึ้น การทำงานที่สัมพันธ์กันระหว่างสินค้าคงคลังและกระแสเงินสดของคุณ จะเกิดขึ้นต่อเนื่องในขั้นตอนเดียว และเมื่อเพิ่มการเชื่อมโยงเข้าสู่ระบบจัดการคลังสินค้า, ระบบสั่งซื้อสินค้าของลูกค้า, ระบบสั่งซื้อสินค้าจาก supplier และเชื่อมต่อกับบริษัทจัดส่งสินค้า ซึ่งเมื่อถึงจุดนี้ ระบบจัดการสินค้าคงคลังของคุณจะมีคุณประโยชน์อย่างแท้จริงคือ คุณสามารถจัดการธุรกิจของคุณได้อย่างครบถ้วน โดยทำงานกับแต่ละธุรกรรมเพียงครั้งเดียวเท่านั้น แทนที่จะต้องดำเนินการกับทุกๆ ระบบที่เกี่ยวข้อง ปริมาณงานจึงลดลง และเกิดข้อผิดพลาดน้อยลง

2.3.6. ตัวอย่างผลิตภัณฑ์ของไมโครซอร์ฟ Microsoft ที่มีการเรียกใช้เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services)

2.3.6.1. ออฟฟิต 2003 (Office 2003)

มีการสนับสนุนเอ็กซ์เอ็มแอล (XML) ในออฟฟิตโปรเฟสชินอล (Office Professional Edition 2003) ซึ่ง

- ช่วยให้บริษัท พัฒนาโซลูชัน (solution) ที่สร้างมาตรฐานให้กับการสร้างและการเก็บรวบรวมข้อมูลซึ่งอยู่ในรูปแฟ้มไบนารีในคอมพิวเตอร์และในแฟ้มที่ใช้ร่วมกันต่างๆ องค์กรธุรกิจสามารถช่วยลดการซ้ำซ้อนของข้อมูลและงาน
- ช่วยให้พนักงานเข้าถึงและวิเคราะห์ข้อมูลจากระบบธุรกิจที่หลากหลายได้ เช่น โปรแกรมประยุกต์เกี่ยวกับการบริหารสัมพันธภาพกับลูกค้า (CRM) ระบบการ

วางแผนทรัพยากรสำหรับองค์กร (ERP) หรือโปรแกรมประยุกต์ในสายงานธุรกิจอื่นๆ

- บริษัทสามารถดึงข้อมูลเอ็กเอ็มแอล (XML) จากแหล่งต่างๆ มายังเอกสารฐานข้อมูล และงานนำเสนอได้

วิศวกรสนามนำคอมพิวเตอร์โน้ตบุ้คที่ติดตั้งอินโฟพาส (InfoPath 2003) ซึ่งเป็นโปรแกรมป้อนข้อมูลโดยใช้ฟอร์มเป็นพื้นฐานมาใช้ในการรวบรวมข้อมูลเกี่ยวกับสถานะของอุปกรณ์ เช่น ปั๊มหรือเครื่องกำเนิดไฟฟ้า หลังจากที่รวบรวมแล้ว จะกรอกข้อมูลนี้ลงในใบสั่งงานสำหรับการซ่อมบำรุง และสร้างแฟ้มประวัติสำหรับสินทรัพย์แต่ละรายการ

ผู้บริหารด้านการเงินที่ต้องการสร้างรายงานรวมจากหลายแผนกในบริษัทจะส่งออกตัวเลขรายได้จากแต่ละแผนกไปยังกระดาษคำนวณ Excel 2003 ไม่ว่าผู้จัดการแผนกจะเก็บตัวเลขเหล่านี้ไว้ที่ใด การรวมข้อมูลก็จะเป็นไปอย่างราบรื่นเพราะมีการติดตามตำแหน่งข้อมูลด้วยเอ็กเอ็มแอล (XML) อยู่เสมอ

นักวิเคราะห์งานวิจัยใช้ Word 2003 เพื่อสร้างรายงานแบบกำหนดเองเกี่ยวกับโทรศัพท์เคลื่อนที่รุ่นล่าสุดสำหรับลูกค้า นักวิเคราะห์รวบรวบข้อมูลที่มี tag XML จากแหล่งข้อมูลภายในและภายนอกหลายแหล่งลงในแฟ้มเดียว หลังจากที่แฟ้มถูกสร้างขึ้น จะสามารถปรับปรุงเอกสารได้ทุกเวลาด้วยการฟื้นฟูข้อมูลในแฟ้มและสร้างรายงานใหม่

2.4. ไมโครซอร์ฟ ดอทเน็ตเฟรมเวิร์ก (Microsoft .NET Framework)

2.4.1. ความหมายของเฟรมเวิร์ก (Framework) และแพลตฟอร์ม (Platform)

2.4.1.1. แอพพลิเคชัน เฟรมเวิร์ก (Application Framework)

Application Framework หมายถึง ชุดของ library หรือ class ที่ใช้สำหรับสร้างโครงสร้างมาตรฐานของ Application ของแต่ละ OS โดยมี reusable code จำนวนมากใน framework เพื่อประหยัดเวลาของผู้พัฒนาโปรแกรม ทำให้ไม่ต้องเขียนโค้ดมาตรฐานทุกๆ ครั้งที่จะมีการพัฒนา application ใหม่ๆ

2.4.1.2. แพลตฟอร์ม (Platform)

แพลตฟอร์ม (Platform) หมายถึง ระบบคอมพิวเตอร์ที่แอพพลิเคชันใช้รันอยู่ในปัจจุบัน ซึ่งประกอบไปด้วย ระบบปฏิบัติการ (Operating System) และฮาร์ดแวร์ ระบบปฏิบัติการจะคอยส่งชุดคำสั่งไปยังไมโครโปรเซสเซอร์ให้ประมวลผล โดยระบบปฏิบัติการก็จะถูกออกแบบมาเพื่อใช้กับไมโครโปรเซสเซอร์แต่ละรุ่นตามสถาปัตยกรรม

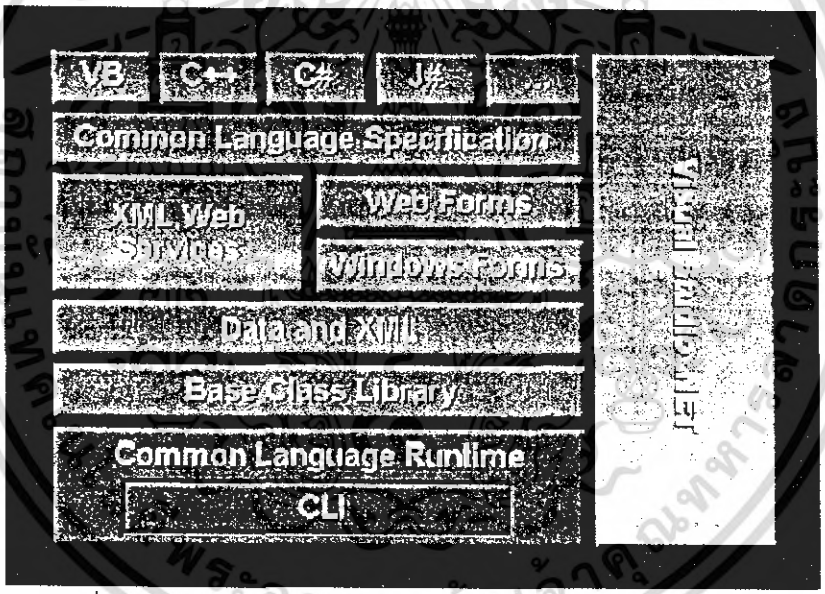
2.4.2.โครงสร้างของดอทเน็ตเฟรมเวิร์ก (.NET Framework)

2.4.2.1. ภาพรวมโครงสร้างของดอทเน็ตเฟรมเวิร์ก (.NET Framework)

ดอทเน็ตเฟรมเวิร์ก (.NET Framework) คือสภาพแวดล้อมในการพัฒนาโปรแกรมด้วยเทคโนโลยี ที่เชื่อมต่อระหว่างระบบปฏิบัติการของเครื่องคอมพิวเตอร์กับ tools ที่เราใช้ในการพัฒนา

เราสามารถพัฒนาโปรแกรมที่ทำงานเสถียรและไม่เปลืองทรัพยากรของระบบได้ โดยแทบไม่ต้องรู้เทคนิคการเขียนโปรแกรมที่เกี่ยวข้องใดๆ เลย ทั้งนี้ก็เพราะ .NET Framework จะคอยจัดการให้เราเอง

ดอทเน็ตเฟรมเวิร์ก (.NET Framework) เป็นคอมโพแนนท์ (component) ภายในวินโดว์ (Windows) ที่รองรับการสร้างและรันแอพพลิเคชันที่มีการเรียกใช้เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web Services) อย่างสมบูรณ์ ดังรูป 2.22.



รูปที่2.22. แสดงโครงสร้างและส่วนประกอบของ .NET Framework

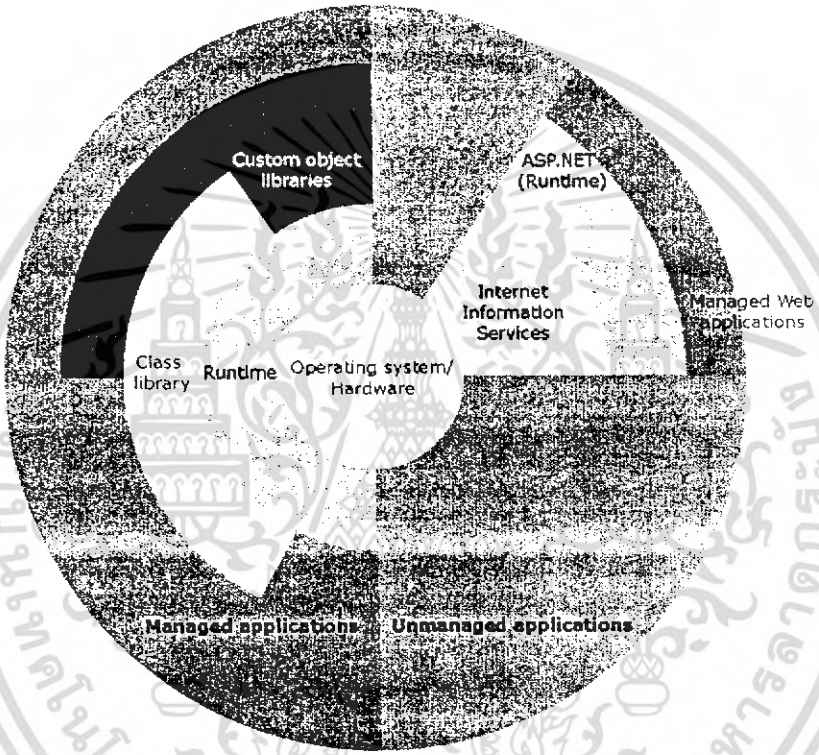
ดอทเน็ตเฟรมเวิร์ก (.NET Framework) ใช้คลาสไลบารี่ร่วมกัน โดยสร้างชุดของ API ขึ้นมาให้สามารถทำการสืบทอด (inherit) , (error handling) และการดีบัก (debugging) ข้ามภาษาได้ ไม่มีการแบ่งแยกว่า C++ พัฒนาโดยใช้ MFC Java พัฒนาโดยใช้ WFC

ดอทเน็ตเฟรมเวิร์ก 1.1 (.NET Framework 1.1) พัฒนาขึ้นจากดอทเน็ตเฟรมเวิร์ก 1.0 (.NET Framework 1.0) โดยบรรจุความสามารถใหม่ๆ, เสริมการทำงานให้ดีขึ้น และปรับปรุงทางด้านความสามารถในการขยายระบบ, ความน่าเชื่อถือ, เทคโนโลยีด้านความปลอดภัยและประสิทธิภาพ ตัวอย่างเช่น เพื่อวางใจในการทำงานของ application ได้มากขึ้น ผู้บริหารระบบจะ

สามารถกำหนดสิทธิการทำงานให้กับแอพพลิเคชัน เอเอสพีดอทเน็ต และเว็บเซอร์วิส (Web services) ได้ จึงเกิดความยืดหยุ่นมากขึ้น และเพิ่มระดับการป้องกันจากผู้ใช้ที่ประสงค์ร้าย

2.4.2.2. Common language runtime

มีหน้าที่รับผิดชอบเรื่องการจัดการเมโมรี่ , เริ่ม หยุดโพรเซส และเทรด (thread) , ควบคุมนโยบายด้านความปลอดภัยและข้อจำกัดที่คอมโพแนนท์หนึ่งๆ จะมีต่อ คอมโพแนนท์อื่นๆ โดยจะเรียกโค้ดที่ถูกกำหนดไว้ในรันทาม (run time) ว่า managed code และเรียกโค้ดที่ไม่ถูกกำหนดไว้ใน รันทาม (run time) ว่า unmanaged code ดังรูป 2.23.



รูปที่ 2.23. แสดงส่วนที่ .NET Framework มีหน้าที่รับผิดชอบ

โปรแกรมดอทเน็ต (.NET) ใดๆ จะถูกคอมไพล์เป็นภาษากลาง (Common Language) ซึ่งมีชื่อเรียกว่า Microsoft Intermediate Language (MSIL หรือ IL) ก่อนเสมอ ทำให้ในทีมพัฒนาเดียวกันสามารถเลือกภาษาที่แตกต่างกันได้ ผู้ที่มารับดูแลโปรแกรมต่อสามารถต่อเติมงานอื่นๆ ได้โดยไม่จำเป็นต้องใช้ภาษาเดิม ก่อนที่ common language runtime จะแปลง IL ให้เป็นภาษาเครื่อง

2.4.2.3. คลาสไลบารี่ (Class library)

เป็น Object-oriented class library ที่นำกลับมาใช้ได้ ก่อให้เกิดรูปแบบที่แน่นอนในการเข้าถึงฟังก์ชันต่างๆ สามารถนำไปพัฒนา application ในส่วนที่เป็นมาตรฐานต่างๆ ได้ จึงช่วยลด

ความจำเป็นในการเรียนรู้ ในการเขียนแอพพลิเคชัน แบ่งออกเป็นกลุ่มๆ ตามหน้าที่ที่เกี่ยวข้อง เช่นการจัดการกับ string , การเก็บข้อมูล , การติดต่อกับฐานข้อมูล , การติดต่อไฟล์ เป็นต้น ซึ่งจะลดเวลาในการเรียนรู้ feature ใหม่ๆ ของ .NET Framework

2.5. เอเอสพีดอทเน็ต (ASP.NET)

2.5.1. ภาพรวมของ เอเอสพีดอทเน็ต (ASP.NET)

เป็นเว็บโปรแกรมมิ่งเทคโนโลยี (web programming technology) ถูกสร้างอยู่บนพื้นฐานคลาสของดอทเน็ทเฟรมเวิร์ก (.NET Framework) ในรูปของชุดคอนโทรลและโครงสร้างพื้นฐานที่ง่ายต่อการสร้างเว็บแอพพลิเคชัน (web application) และ เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services)

2.5.2. คุณสมบัติ ข้อดีและส่วนที่ปรับปรุงจากเอเอสพี (ASP) ของเอเอสพี ดอทเน็ต (ASP.NET)

เทคโนโลยี เอเอสพี ดอทเน็ต (ASP.NET)ได้มีการปรับปรุงโครงสร้างการทำงานใหม่ จึงทำให้การใช้งานต่างไปจากเอเอสพี (ASP) โดย เอเอสพี ดอทเน็ต (ASP.NET)มีลักษณะดังต่อไปนี้

- ในการเขียนสคริปเอเอสพี ดอทเน็ต (ASP.NET)ไม่จำเป็นต้องศึกษาภาษาเอชทีเอ็มแอล (HTML) มากนัก ศึกษาเพียงเบื้องต้นว่าแท็ก (tag) คืออะไร มีหน้าตาอย่างไร <html>,<head>,<body>,<form> ใช้ทำอะไร วิธีสร้างฟอร์มเพื่อรับข้อมูลจากผู้ใช้ทำอย่างไรก็เพียงพอแล้ว
- ในการพัฒนาเว็บฟอร์ม (Web Forms) ส่วนติดต่อผู้ใช้แต่ละตัว (User Interface Elements) จะถูกสร้างมาจาก Control ที่สามารถใช้วิธี drag&drop ได้ การที่ใช้คอนโทล ทำให้ third parties สามารถสร้างคอนโทล ใหม่ขึ้นมาได้
- ในการเขียนโค้ดส่วนติดต่อผู้ใช้กับโค้ดควบคุมการทำงานจะถูกแยกออกจากกัน (อยู่ในไฟล์เดียวกันหรือคนละไฟล์ก็ได้) สามารถเขียนโค้ดขึ้นมาตอบสนองต่อเหตุการณ์ที่เกิดขึ้นกับคอนโทลแต่ละตัวได้ ทำให้ง่ายต่อการปรับปรุงหรือหาข้อผิดพลาด
- ในส่วนวิธีติดต่อกับผู้ใช้มีการใช้ เซิร์ฟเวอร์คอนโทลซึ่งเป็นคอนโทรลที่ถูกประมวลผลบนฝั่งเซิร์ฟเวอร์ ทำให้ควบคุมและเขียนโปรแกรมเพื่อใช้งานได้ง่าย เมื่อต้องการข้อมูลที่ผู้ใช้กรอกเข้ามาทางช่องรับข้อมูล ก็สามารถอ่านข้อมูลมาจากคอนโทรลได้โดยตรง ด้วยการเขียนโค้ดขึ้นมารับค่าพรอพเพอตี้ของคอนโทรลตัวที่เราต้องการ ในทางตรงกันข้ามหากต้องการแสดงข้อมูลกลับไปให้ผู้ใช้ ก็เพียงเขียนโค้ดกำหนดค่าพรอพเพอตี้ให้กับคอนโทรลตัวนั้นๆ
- สามารถนำข้อมูลจากฐานข้อมูลมาแสดงในตารางให้ผู้ใช้ดูได้ โดยไม่ต้องรู้วิธีการสร้างตารางด้วยเอชทีเอ็มแอล (HTML) เลย เพราะมีคอนโทรลมาช่วยทำหน้าที่นี้ให้

- การหาข้อผิดพลาดของ script (Debug) ให้ข้อมูลที่เพียงพอและค่อนข้างละเอียด โดยจะมีคำอธิบายข้อผิดพลาด โค้ดบริเวณที่มีข้อผิดพลาดพร้อมเลขบรรทัด ตลอดจนชื่อไฟล์และหมายเลขบรรทัดที่ผิดอีกครั้งหนึ่ง ทำให้เราเข้าใจข้อผิดพลาดและไปยังจุดผิดได้เร็วกว่าเดิม
- ASP.NET พยายามให้โปรแกรมเมอร์พัฒนาโปรแกรมในลักษณะ event-driven มากกว่าแบบ web scripting โดยเฟรมเวิร์กจะรวมเทคโนโลยีที่มีในปัจจุบัน เช่น JavaScript และคอมโพแนนท์ภายในให้อยู่ในลักษณะของ **"view state"** คือ เปลี่ยนแปลงจาก persistent state เป็น stateless web environment

2.6. Visual Studio .NET

2.6.1. ภาพรวมของวิชวลสตูดิโอ (Visual Studio .NET)

ไมโครซอร์ฟ วิชวลสตูดิโอ (Microsoft Visual Studio .NET) คือไอดีอี (integrated development environment) ที่พัฒนาโดยไมโครซอร์ฟเป็น general-purpose development tool สำหรับการสร้าง enterprise-grade Microsoft .NET application ให้สามารถเชื่อมต่อกันได้ โดยผู้ที่ออกแบบมีจุดประสงค์สำคัญดังต่อไปนี้

- รองรับ เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services) คือ แอพพลิเคชันสามารถรับการร้องขอหรือข้อมูล โดยใช้ XML บน HTTP ซึ่ง เอ็กซ์เอ็มแอลเว็บเซอร์วิส (XML Web services) สามารถถูกเข้าถึงได้ด้วยทุกๆ ภาษา และทุกๆ operating system
- โปรแกรมเมอร์สร้างแอพพลิเคชันที่สามารถเข้าถึงข้อมูลได้ทุกที่ , ทุกเวลา และทุกอุปกรณ์ของคอมพิวเตอร์หรือเซิร์ฟเวอร์หลัก
- โปรแกรมเมอร์สามารถสร้างแอพพลิเคชันโดยใช้ประโยชน์ของ platform ได้อย่างเต็มที่มีฟังก์ชันการใช้งาน , ความน่าเชื่อถือ , ประสิทธิภาพและความปลอดภัยสูงสุด
- รองรับ tool และ component จาก third-party vendor ให้โปรแกรมเมอร์สามารถทำงานเสร็จได้อย่างรวดเร็ว

2.6.1.1.ไอดีอี (Integrated Development Environment)

ไอดีอี (Integrated Development Environment (IDE)) คือซอฟต์แวร์ที่ช่วยโปรแกรมเมอร์ในการพัฒนาโปรแกรมประกอบไปด้วย source code editor, compiler หรือ interpreter, build-automation tools , graphical user interface builder และ debugger

สำหรับการพัฒนาโปรแกรมแบบ object oriented แล้วจะมีการรวมเอา class browser, object inspector และ class hierarchy diagram ไว้ด้วย

2.6.1.2. คอมไพลเลอร์ (Compiler)

Compiler คือโปรแกรมที่ทำหน้าที่แปลโปรแกรมภาษาต่างๆ ให้กลายเป็นภาษาเครื่อง โดยผลลัพธ์ที่ได้จากการคอมไพล์เรียกว่า object module

Compiler ส่วนใหญ่จะแบ่งออกเป็นสองส่วนคือ front end ซึ่งมีหน้าที่แปล source language ให้เป็น intermediate code ส่วนที่สองคือ back end ที่จะมุ่งไปที่การสร้าง output ที่ถูกต้องและมีประสิทธิภาพ ทำให้มีประโยชน์ในเรื่องสามารถใช้เบ็กเอ็น (back end) อันเดียวกับหลายๆ source language ได้ และสามารถใช้หลายๆ back end เพื่อสร้างหลายๆแฟลตฟอร์ม ได้

Compiler front end มีขั้นตอนดังต่อไปนี้

- Lexical analysis
- Syntax analysis
- Semantic analysis
- Intermediate representation

Compile back end มีขั้นตอนดังต่อไปนี้

- Compiler Analysis
- Optimization
- Code generation

Compiler มีสองชนิดคือ native compiler กับ cross-compile

2.6.1.2.1. เนทีฟ คอมไพลเลอร์ (Native Compiler)

เนทีฟ คอมไพลเลอร์ (Native Compiler) คือ คอมไพลเลอร์ ที่สร้างโค้ดเพื่อที่จะรันบนคอมพิวเตอร์และ OS (หรือ platform) เดียวกันกับที่คอมไพลเลอร์รันอยู่

2.6.1.2.2. ครอส คอมไพลเลอร์ (Cross Compiler)

ครอส คอมไพลเลอร์ (Cross Compiler) คือคอมไพลเลอร์ ที่สร้างโค้ดเพื่อที่จะรันบนแพลตฟอร์มอื่นที่ต่างออกไป โดยจะมีประโยชน์มากเมื่อมีการนำฮาร์ดแวร์ แพลตฟอร์มใหม่มาในครั้งแรก

2.6.1.3. Debugger

Debugger เป็นโปรแกรมที่ใช้สำหรับค้นหาข้อผิดพลาดในการทำงานจาก source code ที่ถูกเขียนขึ้นมา เมื่อโปรแกรมเกิดการชนหรือมีบัก debugger จะแสดงตำแหน่งของโค้ดนั้นๆ ขึ้นมา (หากเป็นแบบ source-level debugger หรือ symbolic debugger)

โดยปกติแล้ว Debugger จะมีฟังก์ชันที่ซับซ้อน เช่น สามารถรันโปรแกรมแบบ step by step (single-stepping), stopping (breaking) คือหยุดโปรแกรมเพื่อตรวจสอบสถานะปัจจุบัน และติดตามค่าของตัวแปร เป็นต้น

2.6.2. คุณสมบัติใหม่ของวิชวลสตูดิโอ 2003(*Visual Studio .NET 2003*)

คุณสมบัติใหม่และคุณสมบัติสำคัญๆ ของ Visual Studio .NET 2003 มีดังต่อไปนี้

- สนับสนุน ดอทเน็ทเฟรมเวิร์ก (.NET Framework) 1.1 อย่างเต็มรูปแบบ
- สนับสนุนมาตรฐานล่าสุดของ เว็บซอร์วิส (Web services)

รวมถึงการกำหนดเส้นทางสื่อสาร, ความปลอดภัย, และการแนบข้อมูลให้กับ Web Services Enhancements

- สนับสนุนอุปกรณ์มือถือและ Embedded Device

Visual Studio .NET 2003 สนับสนุน .NET Compact Framework อย่างเต็มรูปแบบ ทำให้อุปกรณ์มือถือ เช่น Pocket PC ก้าวไปสู่การใช้เทคโนโลยี .NET ง่ายขึ้น ต่างจากเดิมที่ต้องการ tool kit แยกต่างหาก

- ช่วยให้การนำแอพพลิเคชัน ไปใช้งานและบำรุงรักษาง่ายขึ้น

วิชวลสตูดิโอ (Visual Studio .NET) 2003 จะแก้ปัญหาส่วนใหญ่ที่เกี่ยวข้องกับการนำแอพพลิเคชัน ไปใช้ และขจัดปัญหาความขัดแย้งของเวอร์ชันของ DLL

- ปรับปรุงการทำงานของ Visual Basic .Net Upgrade Wizard

ด้วยเครื่องมือและคำแนะนำที่เตรียมไว้ จะช่วยให้คุณสามารถอัพเกรดชุดคำสั่งและ แอพพลิเคชันจาก Visual Basic 6.0 ได้อย่างง่ายดาย

- สภาพแวดล้อมในการพัฒนาที่เป็นหนึ่งเดียว

วิชวลสตูดิโอ (Visual Studio .NET) 2003 รวมสภาพแวดล้อมในการทำงานเข้ากับเครื่องมือที่จำเป็นสำหรับการพัฒนา เพื่อรองรับกับแอพพลิเคชันทุกรูปแบบ เครื่องมือออกแบบที่ใช้งานร่วมกันยังทำให้พัฒนา application บน Windows และอุปกรณ์มือถือได้ในรูปแบบเดียวกันอีกด้วย ผลที่ได้ก็คือ นักพัฒนาสามารถนำทักษะมาใช้กับงานได้หลายรูปแบบและเลือกใช้ภาษาโปรแกรมได้มากมาย

ด้วยการสนับสนุนภาษาโปรแกรมต่างๆ มากมาย จึงสามารถเขียนโปรแกรมด้วยภาษาที่ต้องการ และรวมเข้ากับโปรแกรมที่เขียนขึ้นด้วยภาษาใดก็ได้ นักพัฒนาจึงสามารถใช้ความชำนาญที่มีอยู่ในปัจจุบันทำงานต่อไปโดยไม่ต้องฝึกอบรมใหม่

- HTML Designer

พัฒนา HTML, ASP.NET Web Forms โดยไม่ต้องเขียนชุดคำสั่ง HTML หรือสคริปต์ Microsoft Windows Forms , Web Forms และ Mobile Web Forms

- การพัฒนาสำหรับ server

Server Explorer และเครื่องมืออื่นๆ ได้รวมองค์ประกอบของ application ที่อยู่บน server เพื่อเรียกใช้เครื่องมือสำหรับจัดการ server, event logs, database และ Web services ได้อย่างสะดวก

- Visual Studio .NET Debugger

  ใช้งาน remote debugger ได้

- Dynamic Help และ Task List

จะช่วยให้จัดการขั้นตอนการพัฒนาได้อย่างเป็นระบบ โดยใส่หมายเหตุประกอบชุดคำสั่ง , การตรวจสอบและเข้าถึงข้อผิดพลาดและเตือนได้อย่างรวดเร็ว

- Visual Basic .NET

สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP) อย่างเต็มรูปแบบ รวมถึงคุณสมบัติการถ่ายทอด (Inheritance), การจัดการ exception แบบมีโครงสร้างและ free-threading

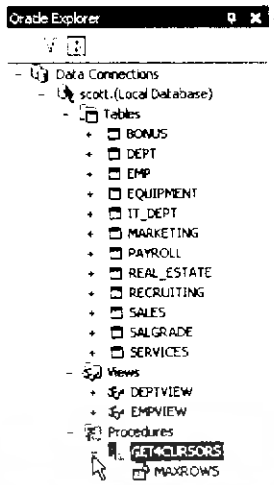
2.7. Oracle Developer Tools for Visual Studio .NET

2.7.1. ส่วนประกอบต่างๆ ของ Oracle Developer Tools Visual Studio .NET

เป็นเครื่องมือแอดอิน (add-in) ของออราเคิลที่ช่วยให้นักพัฒนาที่ใช้ วิชวลสตูดิโดดอทเน็ต (Visual Studio .NET) สามารถพัฒนาโปรแกรมได้รวดเร็วยิ่งขึ้น โดยมีส่วนประกอบต่างๆ ดังนี้

2.7.1.1. Oracle Explorer

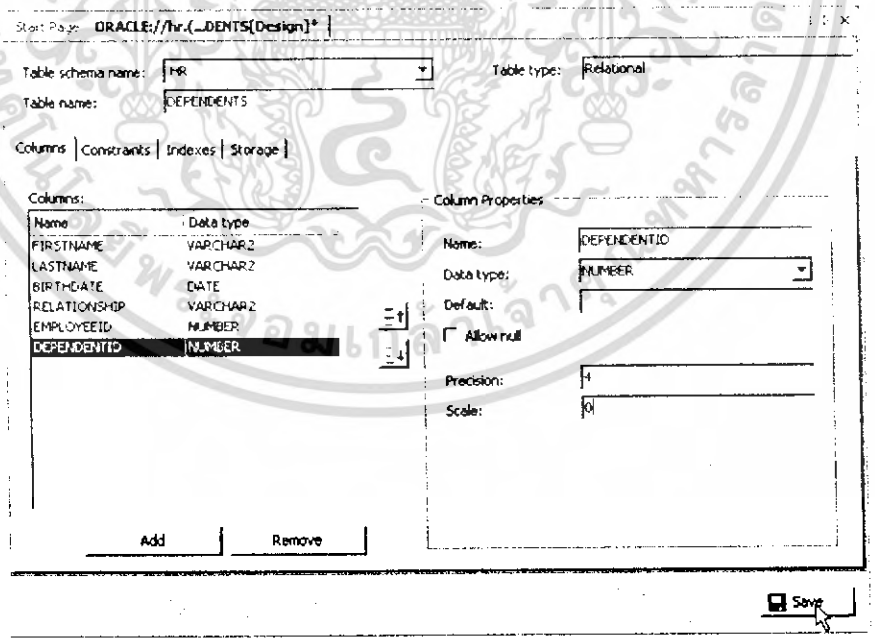
เป็นหน้าต่าง tree control ที่แสดงให้เห็นโครงสร้างของ Oracle Schema เช่น tables , stored procedures เมื่อ schema object ถูกเลือกจะมีการแสดง metadata ไว้ที่ "properties pane" ดังรูป 2.24.



รูปที่ 2.24. แสดง Oracle Explorer

2.7.1.2. Designers and Wizards

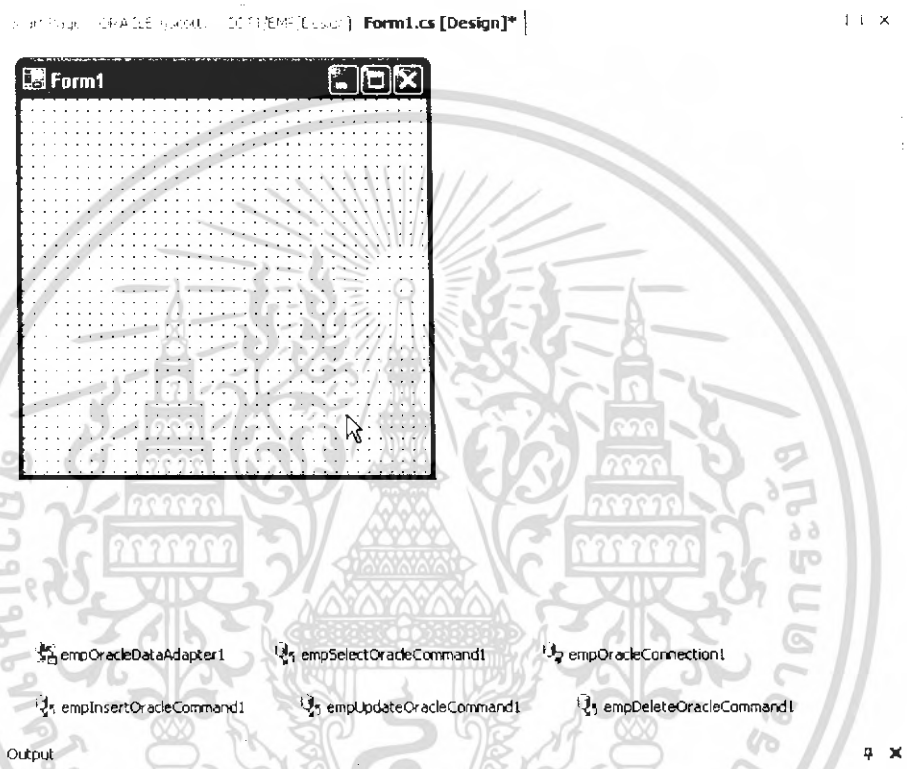
เป็นเครื่องมือ (wizard) ที่ทำให้การสร้างหรือเปลี่ยนแปลง database objects เช่น tables, views, stored procedures, stored functions, PL/SQL packages, sequences, indexes, constraints, triggers, synonyms และอื่นๆ ทำได้ง่ายขึ้นในรูปแบบ GUI ไม่จำเป็นต้องหาจาก datatype หรือจำ SQL syntax สามารถเลือก column ต่างๆ ได้ง่ายๆ จาก table name และ name of column เลือก datatype ได้จาก drop down box เมื่อกด "save" จะปรากฏ schema object ที่ Oracle explorer ดังรูป 2.25.



รูปที่ 2.25. แสดง Designers and Wizards

2.7.1.3. Automatic Code Generation

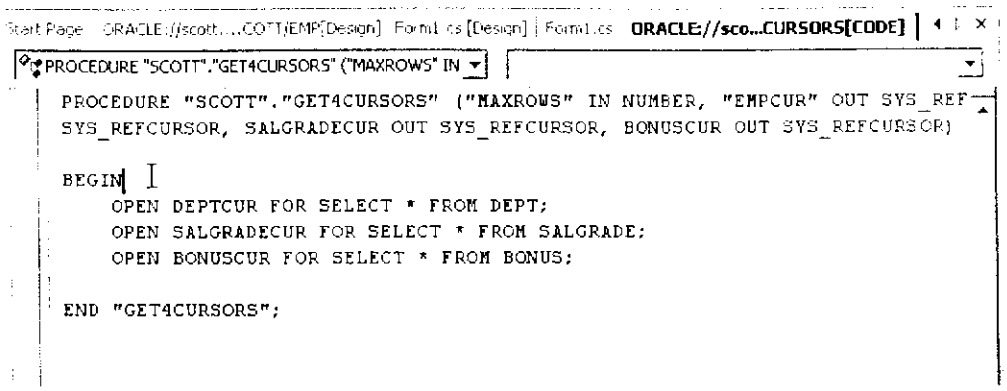
เมื่อทำการ drag&drop ตัว schema object จาก Oracle Explorer ไปบน application's form มันจะทำการสร้างโค้ดในส่วนของการ SELECT , UPDATE , INSERT, และ DELETE ของโครงสร้าง object นั้นให้ โดยมีคลาส OracleDataAdapter เป็นตัวเชื่อมต่อ User Interface element (เช่น datagrid) บน application's form ไปยัง Oracle database ด้วยการเพิ่มโค้ดเพียงเล็กน้อย โดยมีลักษณะ ดังรูป 2.26.



รูปที่2.26 แสดง Automatic Code Generation

2.7.1.4. PL/SQL Editor

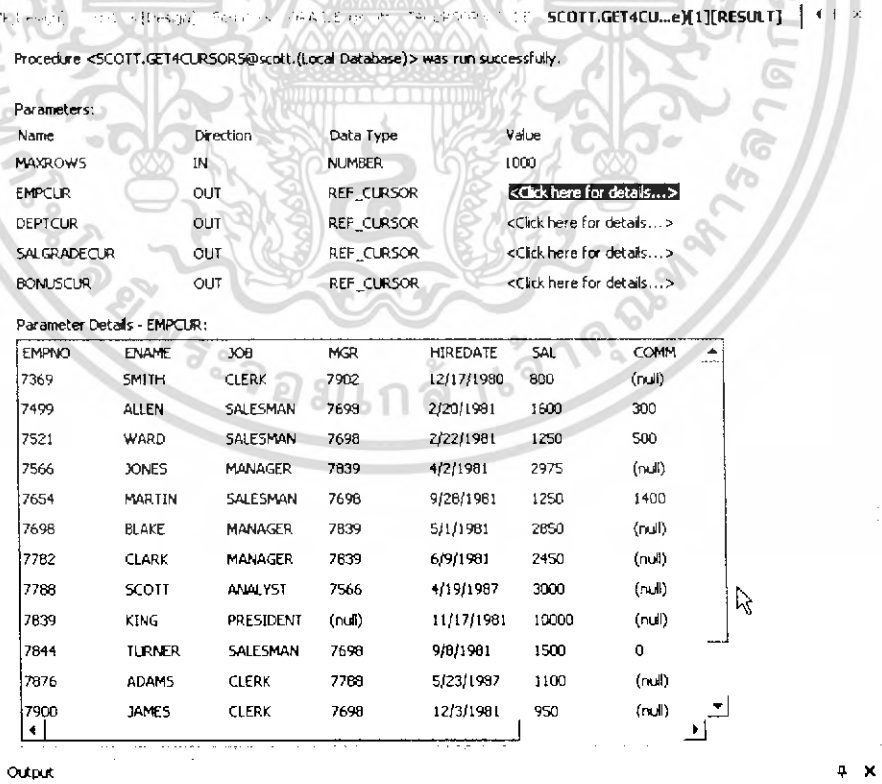
มีการรวม development environment สำหรับ Oracle developers ที่ต้องการพัฒนา stored procedure หน้าต่าง PL/SQL Editor มีการใช้สีและ auto completion เพื่อความสะดวก รวดเร็วในการพัฒนา application หากทำการ compile แล้วเกิด error จะมีข้อมูลนั้นอยู่ใน error task list และสามารถคลิกเพื่อที่จะไปยังบรรทัดที่เกิด error ได้ ดังรูป 2.27.



รูปที่ 2.27. แสดง PL/SQL Editor

2.7.1.5. Stored Procedure Testing

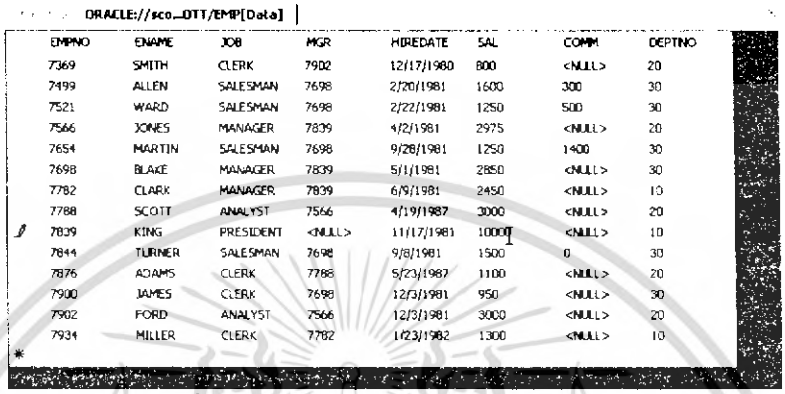
สามารถทดสอบ stored procedures และ function ได้อย่างรวดเร็ว โดยการกด "Run" ที่ Oracle Explorer (โดยใส่ parameters ที่ต้องการไปด้วย) output ที่ออกมาอยู่ในรูปแบบที่อ่านง่าย สำหรับ output ที่ซับซ้อนเมื่อคลิกที่ลิงค์แล้ว จะเปิดตารางที่ประกอบด้วยข้อมูลที่ซับซ้อนออกมาให้สามารถอ่านได้ง่าย ดังรูป 2.28.



รูปที่ 2.28. แสดง Stored Procedure Testing

2.7.1.6. Oracle Data Windows

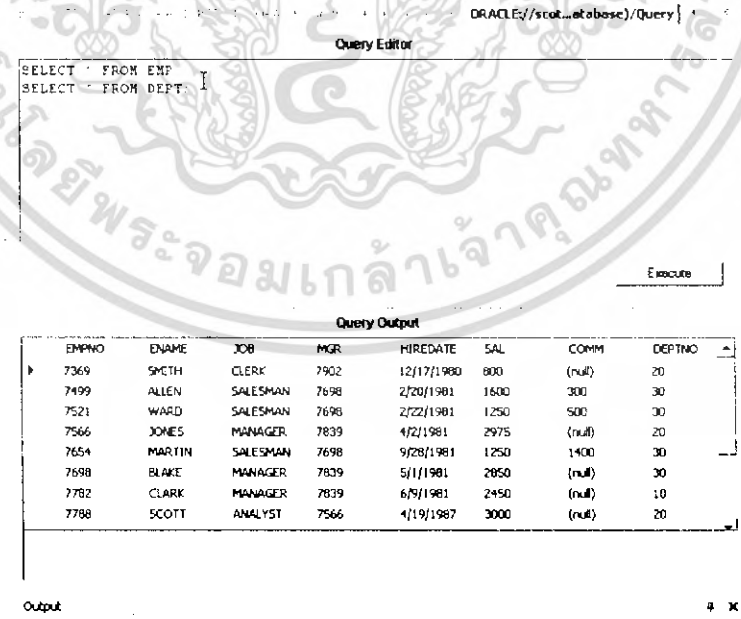
แสดง table หรือ view data ที่ง่ายต่อการอ่าน และอนุญาตให้ผู้ใช้สามารถ INSERT, UPDATE , DELETE ข้อมูลได้โดยไม่ต้องมี Visual Studio .NET development environment ถ้าตารางมีขนาดใหญ่ ก็สามารถที่จะเลือกข้ามไปที่ row นั้นๆ ได้ ดังรูป 2.29.



รูปที่ 2.29. แสดง Oracle Data Windows

2.7.1.7. SQL Query Windows

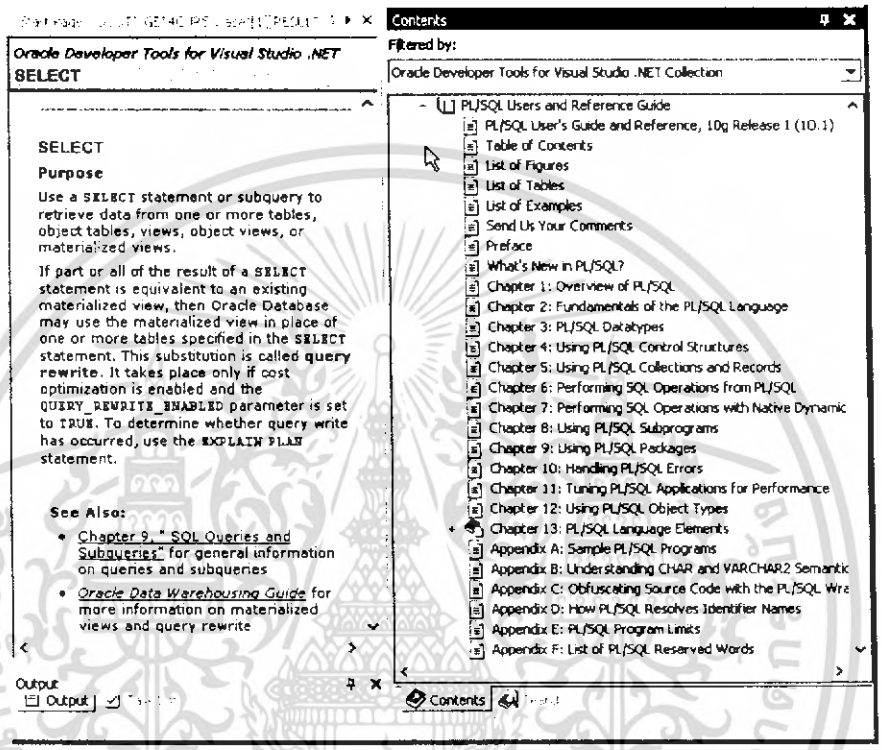
ใช้สำหรับ Execute ad-hoc SQL statement หรือ script เพื่อความสะดวกในการตรวจสอบความถูกต้องของผลลัพธ์ที่ถูก query ออกมา สามารถใช้คำสั่งแบบ multiple statement ได้ (output ใหม่จะต่อท้าย output เก่า) ดังรูป 2.30.



รูปที่ 2.30. แสดง SQL Query Windows

2.7.1.8. Integrated Help System

มีการรวบรวมเอกสารสำคัญของ Oracle ให้อยู่ในรูปแบบของ Visual Studio help format เช่น The SQL Language Reference Guide, the PL/SQL User and Reference Guide, และ the Error Messages and Codes และรวมความสามารถของ Visual Studio context sensitive help ไว้ด้วย ดังรูป 2.31.



รูปที่ 2.31. แสดง Integrated Help System

2.8. โครงสร้างข้อมูลแบบกราฟ (Graph)

กราฟ(Graph) เป็นโครงสร้างข้อมูลที่ใช้แสดงความสัมพันธ์ระหว่างออบเจ็กต์ (object) โดยจะประกอบไปด้วยกลุ่มของโหนด (vertex) ที่ใช้แทนออบเจ็กต์และกลุ่มของเส้นเชื่อม (Edge) ระหว่างโหนดกรณีที่ออบเจ็กต์ตั้งแต่ 2 ออบเจ็กต์ขึ้นไปมีความสัมพันธ์กัน ก็จะมีเส้นเชื่อมระหว่างออบเจ็กต์หรือโหนดเหล่านั้น

โครงสร้างของกราฟนั้นจะประกอบไปด้วยโหนดและเส้นเชื่อมระหว่างโหนด (กรณีที่โหนดแต่ละโหนดนั้นมีความสัมพันธ์กัน) โดยสามารถเขียนแทนด้วยสัญลักษณ์ต่อไปนี้

G = (V,E) เมื่อ G คือกราฟ

V คือกลุ่มของโหนด

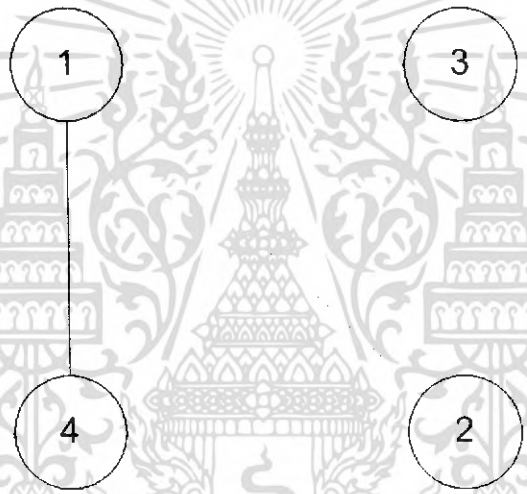
E คือเส้นเชื่อมระหว่างโหนด

2.8.1. ประเภทของกราฟ

ในส่วนของประเภทของกราฟนั้นจะแบ่งออกได้ป็น 3 ประเภทหลักๆ คือ Direct graph , Undirect graph และ Cyclie graph

2.8.1.1. Undirect graph

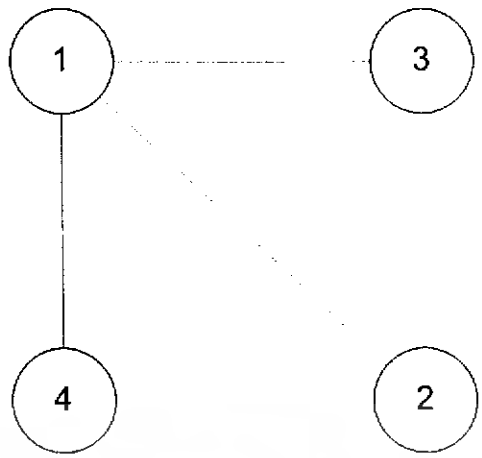
Undirect graph นี้มีลักษณะเป็นกราฟที่มีเส้นที่เชื่อมต่อระหว่างโหนดในแต่ละโหนด โดยเส้นที่เชื่อมต่อนี้จะไม่มีการแสดงทิศทางของการเชื่อมต่อ ดังรูป 2.32.



รูปที่ 2.32. แสดงตัวอย่างของ Undirect graph

2.8.1.2. Direct graph

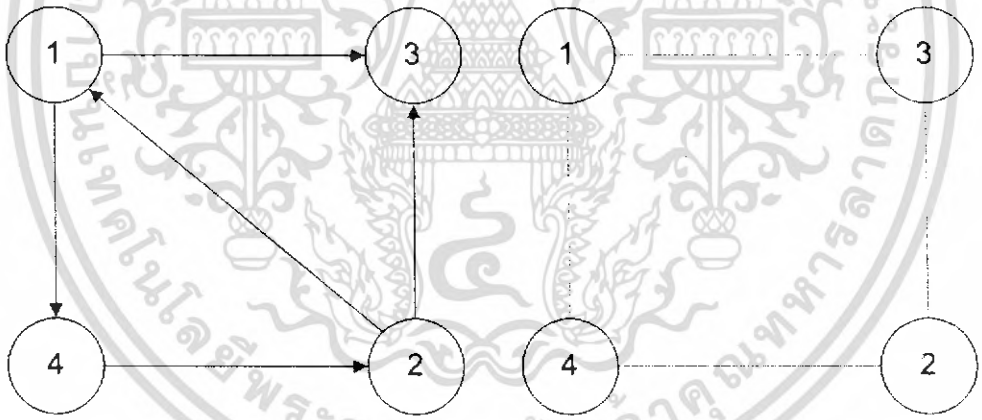
Direct graph หรืออีกชื่อหนึ่งคือ Digraph จะมีลักษณะเป็นกราฟที่มีเส้นเชื่อมต่อระหว่างโหนดในแต่ละโหนด โดยเส้นเชื่อมของ Direct graph นี้จะไม่เหมือนกับเส้นเชื่อมของ Undirect graph ตรงที่เส้นเชื่อมของ Direct graph นั้นจะมีการแสดงทิศทางของเส้นเชื่อมต่อด้วย ดังรูป 2.33.



รูปที่2.33. แสดงตัวอย่างของ Direct graph

2.8.1.3. Cyclic graph

Cyclic graph นี้จะเป็นกราฟที่เส้นทาง (path) เกิดจากเส้นเชื่อมระหว่างโหนดที่มีลักษณะเป็นวงจรปิด (Cycle) คือหมายถึงมีโหนดต้นทางและโหนดปลายทางเป็นโหนดเดียวกัน โดย Cyclic graph สามารถเป็นได้ทั้ง Direct graph และ Undirect graph ดังรูป 2.34.



รูปที่2.34 แสดงตัวอย่างของ Cyclic graph

2.9. การค้นหาข้อมูล (searching)

การค้นหาคำตอบ หรือการค้นหาข้อมูลในทางคอมพิวเตอร์มักจะกระทำบนโครงสร้างข้อมูลแบบต้นไม้ และกราฟ ทั้งนี้เพราะโครงสร้างข้อมูลในลักษณะนี้สามารถทำให้การค้นหาทำได้สะดวกและสามารถพลิกแพลงการค้นหาได้ง่าย ดังนั้นในการค้นหาข้อมูลที่มีขนาดใหญ่นั้น ก่อนการค้นหา หรือระหว่างการค้นหา ข้อมูลที่จะถูกค้นจะต้องถูกจัดให้อยู่ในรูปแบบของต้นไม้หรือกราฟเท่านั้น การค้นหาข้อมูลบนโครงสร้างต้นไม้และกราฟสามารถจำ าแนกได้ 2 แบบคือ การค้นหาแบบไบล์ด(Blind Search) และการค้นหาแบบฮิวริสติก(Heuristic Search)

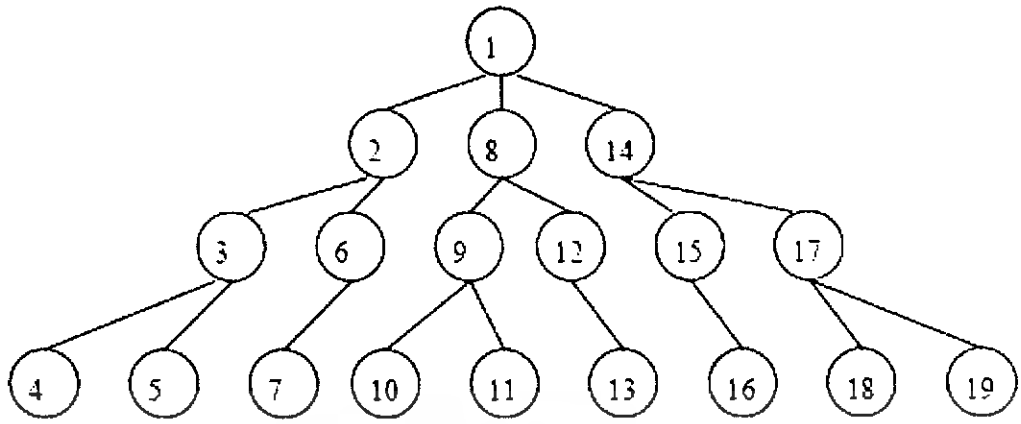
2.9.1. การค้นหาแบบไบล์ด (Blind Search)

การค้นหาแบบไบล์ด(Blind search) เป็นการค้นหาแบบที่เดินทางจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง โดยอาศัยทิศทางเป็นตัวกำหนดการค้นหา และไม่ต้องมีข้อมูลอะไรมาช่วยเสริมในการตัดสินใจว่าจะเดินทางต่อไปอย่างไร หรือกล่าวอย่างง่าย ๆ คือการจะหยิบข้อมูลใดมาช่วยในการค้นหาต่อไปนั้นจะไม่ต้องอาศัยข้อมูลใด ๆ ทั้งสิ้น นอกจากทิศทางซึ่งเป็นรูปแบบตายตัว ซึ่งการค้นหาแบบไบล์ดสามารถแบ่งย่อยได้ดังนี้ คือ การค้นหาทั้หมด และการค้นหาบางส่วน

- **การค้นหาทั้งหมด(exhaustive search)** คือ การค้นหาทั้งหมดของปริภูมิสถานะ
- **การค้นหาบางส่วน (partial search)** การค้นหาเพียงบางส่วนของปริภูมิสถานะ ซึ่งในความเป็นจริงการค้นหาส่วนมากใช้การค้นหาเฉพาะบางส่วนเท่านั้นเนื่องจากปริภูมิสถานะมักมีขนาดใหญ่ ทำให้ไม่สามารถค้นหาได้ทั้งหมด ดังนั้นจึงมีความเป็นไปได้ว่าคำตอบที่ได้อาจไม่ใช่คำตอบที่ดีที่สุด ซึ่งการค้นหาแบบนี้สามารถแบ่งได้เป็น 2 ประเภทคือ การค้นหาแบบลึกก่อน(Depth first search) และการค้นหาแบบกว้างก่อน (Breadth first search)

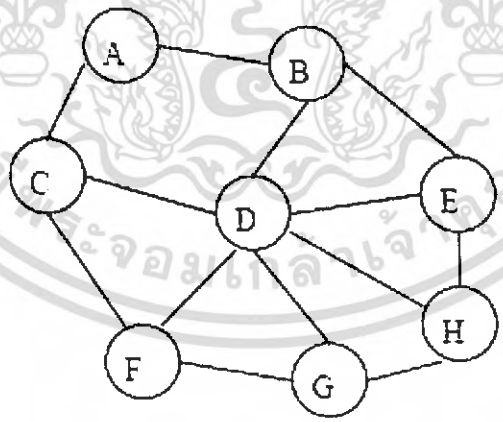
2.9.1.1. การค้นหาแบบลึกก่อน(Depth first search)

การค้นหาแบบลึกก่อนเป็นการค้นหาที่กำหนดทิศทางจากรูปของ โครงสร้างต้นไม้ ที่เริ่มต้นจากโหนดราก(Root node) ที่อยู่บนสุด แล้วเดินลงมาให้ลึกที่สุด เมื่อถึงโหนดล่างสุด (Terminal node) ให้ย้อนขึ้นมาที่จุดสูงสุดของกิ่งเดียวกันที่มีกิ่งแยกและยังไม่ได้เดินผ่าน แล้วเริ่มเดินลงจนถึงโหนดลึกสุดอีก ทำเช่นนี้สลับไปเรื่อยจนพบโหนดที่ต้องการหาหรือสำรวจครบทุกโหนดแล้วตามรูปที่ 2.35 การค้นหาแบบลึกก่อนจะมีลำดับการเดินตามโหนดดังตัวเลขที่กำกับไว้ในแต่ละโหนด



รูปที่2.35. ลำดับการเดินทางบนโหนดของการค้นหาแบบลึกก่อนบนโครงสร้างต้นไม้

โครงสร้างข้อมูลที่ใช้สำหรับการค้นหานี้ยังที่จะสามารถใช้กับโครงสร้างกราฟได้ด้วย โดยอาศัยหลักการเดียวกัน แต่สำหรับการเดินทางบนกราฟนั้นจะไม่มีโหนดลึกสุดดังนั้นการเดินทางบนกราฟจะต้องปรับวิธีการเป็นดังนี้ โดยเริ่มจากโหนดเริ่มต้น จากนั้นให้นำโหนดที่อยู่ติดกับโหนดที่กำลังสำรวจอยู่ (ที่ยังไม่ได้ทำการสำรวจและยังไม่ได้อยู่ในแสต็กมาใส่แสต็ก) มาเก็บไว้ในสแต็กเมื่อสำรวจโหนดนั้นเสร็จ ให้พอพ (pop) ตัวบนสุดของโหนดออกมาทำการสำรวจ แล้วนำโหนดข้างเคียงทั้งหมดที่ยังไม่ได้สำรวจมาต่อท้ายแสต็ก แล้วพอพตัวบนสุดออกมาสำรวจ ทำเช่นนี้เรื่อย ๆ จนกระทั่งพบโหนดที่ต้องการ หรือสำรวจครบทุกโหนด ดังรูปที่ 2.36.



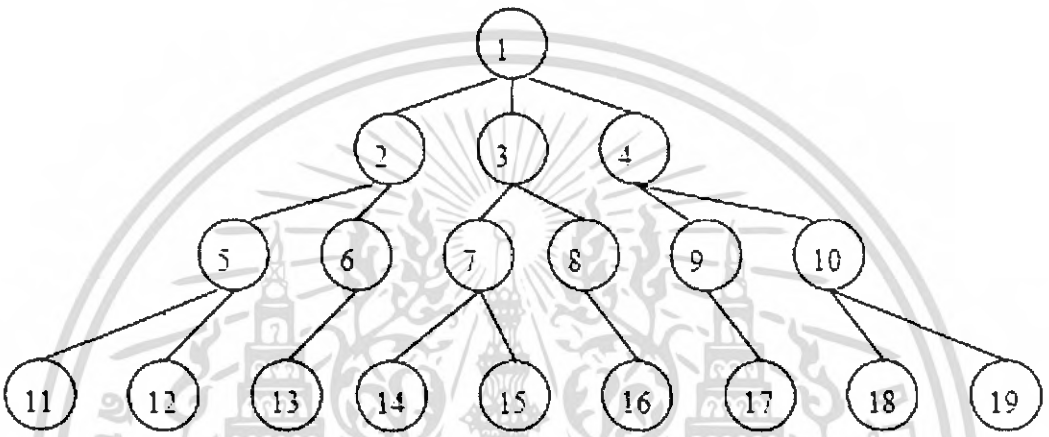
รูปที่ 2.36. โครงสร้างข้อมูลแบบกราฟ

การสำรวจจะเริ่มต้นที่ A และนำโหนดข้างเคียง B และ C มาเก็บไว้ในแสต็ก เมื่อสำรวจ Aเสร็จพอพข้อมูลจากแสต็กออกมาได้ C ทำการสำรวจ C และนำโหนดข้างเคียงกับ C ที่ยังไม่ได้ทำการสำรวจและยังไม่ได้อยู่ในแสต็กมาใส่แสต็ก D และ F พุช(Push) ใส่แสต็ก ดังนั้นในแสต็กตอนนี้มี B D F อยู่ เมื่อสำรวจ C เสร็จ พอพ F ออกมาทำการสำรวจ แล้วนำโหนดข้างเคียงที่ยัง

ไม่ได้สำรวจและยังไม่ได้อยู่ในแสต็กมาใส่แสต็ก ซึ่งก็คือ G ดังนั้นข้อมูลในแสต็กจะเป็น B D G ทำเช่นนี้ไปเรื่อย ๆ จนจบการทำงานก็จะได้ลำดับการสำรวจคือ (A C F G H E D B)

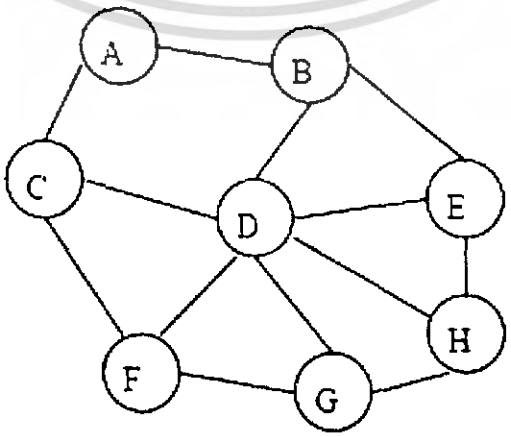
2.9.1.2. การค้นหาแบบกว้างก่อน (Breadth first search)

การค้นหาแบบกว้างก่อนนี้จะเป็นการกำหนดทิศทางการค้นหาแบบที่แต่ละระดับของโครงสร้างต้นไม้โดยเริ่มจากโหนดราก(ระดับที่ 0) แล้วลงมาระดับที่ 1 จากซ้ายไปขวา เมื่อเสร็จระดับที่ 1 ไประดับที่ 2จากซ้ายไปขวาเช่นกัน ทำเช่นนี้เรื่อย ๆ จนพบโหนดที่ต้องการ ดังรูปที่ 2.37.



รูปที่ 2.37. ลำดับการค้นหาแบบกว้างก่อนบนโครงสร้างต้นไม้

สำหรับการค้นหาแบบกว้างก่อนบนโครงสร้างต้นไม้ จะอาศัยโครงสร้างข้อมูลแบบคิว (Queue) มาช่วย และด้วยวิธีการเช่นเดียวกับการค้นหาแบบลึกก่อนคือ ให้เริ่มต้นสำรวจที่โหนดเริ่มต้น แล้วนำโหนดข้างเคียงเก็บไว้ในคิว เมื่อสำรวจโหนดเริ่มต้นเสร็จ ให้นำข้อมูลในคิวออกมาสำรวจ แล้วนำโหนดข้างเคียงที่ยังไม่ได้สำรวจและไม่ได้อยู่ในคิวใส่คิวไว้ ทำเช่นนี้ไปเรื่อย ๆ จนพบโหนดที่ต้องการ หรือเมื่อสำรวจครบทุกโหนด ดังรูปที่ 2.38.



รูปที่ 2.38. โครงสร้างข้อมูลแบบกราฟ

การสำรวจเริ่มต้นที่ A นำโหนดข้างเคียง B C ไว้ในคิว เมื่อสำรวจ A เสร็จ นำข้อมูลในคิวคือ Bออกมาสำรวจ แล้วนำข้อมูลข้างเคียงคือ D E ใส่คิว ตอนนี้คิวจะมี B D E อยู่ แล้วนำ B ออกมาสำรวจทำเช่นนี้เรื่อย ๆ จะได้ลำดับการสำรวจข้อมูลคือ (A B C D E F G H)

การค้นหาแนวลึกก่อนและการค้นหาแบบแนวกว้างก่อนนั้น มีข้อที่แตกต่างกันหลายประการดังตารางที่ 2.1.

ตารางที่ 2.1. แสดงการเปรียบเทียบการค้นหาระหว่างแบบการค้นหาแนวลึกก่อนกับการค้นหาแนวกว้างก่อน

การค้นหาแนวลึกก่อน (Depth first search)	**การค้นหาแนวกว้างก่อน (Breadth first search)**
1.ใช้หน่วยความจำน้อยกว่า เพราะว่าสถานะในเส้นทางค้นหาปัจจุบันเท่านั้นที่ถูกเก็บ(ในขณะใดๆ จะเก็บเส้นทางเดียว พอจะไปเส้นทางอื่น เส้นทางที่ผ่านมาก็ไม่จำเป็นต้องเก็บ)	1.ใช้หน่วยความจำมาก เพราะต้องเก็บสถานะไว้ทุกตัวเพื่อหาเส้นทางจากสถานะเริ่มต้นไปหาคำตอบ
2. อาจจะติดเส้นทางที่ลึกมากโดยไม่พบคำตอบ เช่นในกรณีที่เส้นทางนั้นไม่มีคำตอบและเป็นเส้นทางที่ยาวไม่สิ้นสุด จะทำไม่สามารถไปเส้นทางอื่นได้	2. จะไม่ติดเส้นทางที่ลึกมาก ๆ โดยไม่พบคำตอบ
3. ถ้าคำตอบอยู่ในระดับ n+1 สถานะอื่นทุกตัวที่ระดับ 1ถึงระดับ n ไม่จำเป็นต้องถูกกระจายจนหมด	3. ถ้าคำตอบอยู่ในระดับ n+1 สถานะทุกตัวที่ระดับ 1ถึงระดับ n จะต้องถูกกระจายจนหมด ทำให้มีสถานะที่ไม่จำเป็นในเส้นทางที่จะไปสู่คำตอบถูกกระจายออกด้วย
4. เมื่อพบคำตอบไม่สามารถรับประกันได้ว่าเส้นที่ได้เป็นเส้นทางที่สั้นที่สุดหรือไม่	4. ถ้ามีคำตอบจะรับประกันได้ว่าจะพบคำตอบแน่ ๆ และจะได้เส้นทางสั้นที่สุดด้วย

2.9.2. การค้นหาแบบฮิวริสติก(Heuristic Search)

การค้นหาคำตอบอาศัยวิธีการทางฮิวริสติก (heuristic search) มีความความแตกต่างจากการค้นหาข้อมูลแบบธรรมดา ตรงที่การค้นหาข้อมูลธรรมดา ผู้ที่ทำการค้นข้อมูลจะต้องตรวจสอบข้อมูลทีละตัวทุกตัวจนครบ แต่ฮิวริสติกจะไม่ลงไปดู ข้อมูลทุกตัว วิธีการนี้จะเลือกได้คำตอบที่เหมาะสมให้กับการค้นหา ซึ่งมีข้อดีคือ สามารถทำการ ค้นหาคำตอบจาก ข้อมูลที่มีขนาดใหญ่มาก ๆ ได้ แต่มีข้อเสียคือคำตอบที่ได้เป็นเพียงคำตอบที่ดี เท่านั้นไม่แน่ว่าจะดีที่สุด แต่เนื่องจากว่าปัญหาในบางลักษณะนั้นใหญ่มาก และเป็นไปไม่ได้ที่จะทำ การค้นหาด้วยวิธีธรรมดากระบวนการของฮิวริสติกจึงเป็นสิ่งที่จำเป็นในเรื่องของฮิวริสติกนั้น นอกจากจะมีการค้นหาแบบฮิวริสติกแล้ว ยังมีอีกสิ่งหนึ่งที่สำคัญคือ ฮิวริสติกฟังก์ชัน (heuristic function) ซึ่งหมายถึงฟังก์ชันที่ทำหน้าที่ในการวัดขนาดของความเป็น ไปได้ในการแก้ปัญหาซึ่งจะแสดงด้วยตัวเลข วิธีการดังกล่าวจะกระทำ าได้โดยการพิจารณาถึงวิธีการ (aspects) ต่าง ๆ ที่ใช้ในการแก้ปัญหา ณ สถานะหนึ่งว่าจะสามารถแก้ปัญหาได้ตามที่ต้องการหรือไม่ โดยกำหนดเป็นน้ำหนักที่ให้กับการแก้ปัญหาของแต่ละวิธี น้ำหนักเหล่านี้จะถูกแสดงด้วยตัวเลขที่กำกับไว้กับโหนดต่าง ๆ ในกระบวนการ ค้นหา และค่าเหล่านี้จะเป็นตัวที่ใช้ในการประมาณความเป็นไปได้ว่าเส้นทางที่ผ่านโหนดนั้นจะมี ความเป็นไปได้ในการนำไปสู่หนทางการแก้ปัญหาได้มากน้อยแค่ไหนจุดประสงค์ที่แท้จริงของฮิวริสติก ฟังก์ชันก็คือ การกำกับทิศทางของกระบวนการค้นหาเพื่อให้อยู่ในทิศทางที่ได้ประโยชน์สูงสุด โดยการบอกว่าเราควรเลือกเดินเส้นทางไหนก่อน ในกรณีที่มีเส้น ทางมากกว่าหนึ่งเส้นทางต้องเลือกกระบวนการค้นหาแบบฮิวริสติก โดยปกติแล้วจะต้องอาศัยฮิวริสติกฟังก์ชัน ทำให้การแก้ปัญหาหนึ่ง ๆ จะดีหรือไม่ ก็ขึ้นอยู่กับฮิวริสติกฟังก์ชัน ดังนั้นการค้นหาแบบนี้จึงไม่มีอะไรเป็นหลักประกันว่าจะได้สิ่งที่ไม่ดีออกมาด้วยเหตุนี้เอง เราจึงเรียกการ ค้นหาแบบฮิวริสติกนี้ว่า Weak Methods หรือจะกล่าวอีกนัยหนึ่งคือ Weak Methods เป็นกระบวนการควบคุมโดยทั่วไป (general-purpose control stategies)

2.10. การบำรุงรักษาซอฟต์แวร์ (software Maintenance)

2.10.1. ความหมายและภาพรวม

IEEE Computer Society ได้ให้ความหมายคำว่า software maintenance ไว้ว่า คือการที่จะทำให้ ระบบของ software นั้นสามารถทำการ ตรวจสอบความถูกต้อง, พัฒนาประสิทธิภาพ, หรือปรับเปลี่ยนสภาพแวดล้อม ได้ หรือให้ ระบบนั้นสามารถที่จะคงสภาพที่ให้สามารถทำงานได้อย่างสภาพปกติ

2.10.2. Level of maintenance

Level 1 Full maintenance: จะมีการ ปรับปรุง เปลี่ยนแปลงระบบได้, มีการ training ให้กับผู้พัฒนา, มีการทำ document อย่างเต็มรูปแบบ, มี Help desk ไว้คอยช่วยเหลือและมี การส่ง team เพื่อช่วย support เช่น maintainer นั้นจะต้องทำการแสดงฟังก์ชันทั้งหมดของการ maintenance software และทำการแก้ปัญหาที่เกิดขึ้นทั้งหมด รวมทั้งปรับปรุงระบบตามการร้องขอของ user

Level 2 Corrective maintenance: จะคล้ายกับ Full maintenance แต่ว่าจะไม่มีการปรับปรุงเปลี่ยนแปลงระบบ จะมีเพียงการทำ corrective maintenance เท่านั้นจะไม่มีการทำการปรับปรุงเปลี่ยนแปลงระบบตามคำขอที่ต้องการจาก user อีก

Level 3 Limited corrective maintenance: จะคล้ายกับ Corrective maintenance แต่ว่าจะไม่มีการแก้ปัญหาที่มีความสำคัญสูงๆ เท่านั้น

Level 4 Limited software configuration management: จะทำการจำกัด function ที่จะทำการเปลี่ยนแปลงหรือแก้ไขกับระบบ เหมาะกับองค์กรที่มีงบประมาณน้อยมาก และถ้าเกิดงบประมาณมากขึ้น ก็สามารถเพิ่มระดับการ maintenance เพื่อเพิ่มประสิทธิภาพได้

2.10.3. รูปแบบของการ Maintenance

2.10.3.1 Corrective Maintenance คือการตรวจสอบความถูกต้องของระบบ และทำการแก้ไขข้อผิดพลาดที่เกิดขึ้นกับตัวระบบในการทำกิจกรรมต่างๆ บนระบบ

2.10.3.2 Adaptive Maintenance จะมีการทำการปรับปรุงเปลี่ยนแปลงค่าสภาพแวดล้อมที่มีอยู่ในระบบ

2.10.3.3 Perfective Maintenance จะทำการปรับปรุงระบบให้เกิดประสิทธิภาพสูงขึ้น โดยอาจจะเพิ่มคุณสมบัติต่างๆเพื่อให้ระบบมีประสิทธิภาพมากขึ้นได้

2.10.4 ปัจจัยของการ Maintenance ที่จะส่งผลกระทบ

Development process ก็คือ การที่รวมกระบวนการในการพัฒนา software และการที่จะทำการกำหนดและวางความสามารถของการ maintenance ไว้ ณ กระบวนการไหนที่จะให้เกิดประสิทธิภาพสูงสุด

Documentation คือการจัดการทำเอกสารต่างๆ จะช่วยลดต้นทุนเมื่อมีการเปลี่ยนแปลงหรือการปรับปรุงเกิดขึ้นกับระบบ รวมทั้งไม่ให้นักพัฒนารุ่นต่อไปต้องเสียเวลาในการที่จะรับช่วงต่อหรือทำการแก้ไขระบบ

Program Comprehensive คือความเข้าใจในตัวระบบ ถ้าเกิดผู้ที่จะรับผิดชอบนั้นไม่มีความเข้าใจในระบบ อาจเกิดจากไม่มีความรู้ความเข้าใจในระบบ, มีข้อมูลเกี่ยวกับระบบไม่

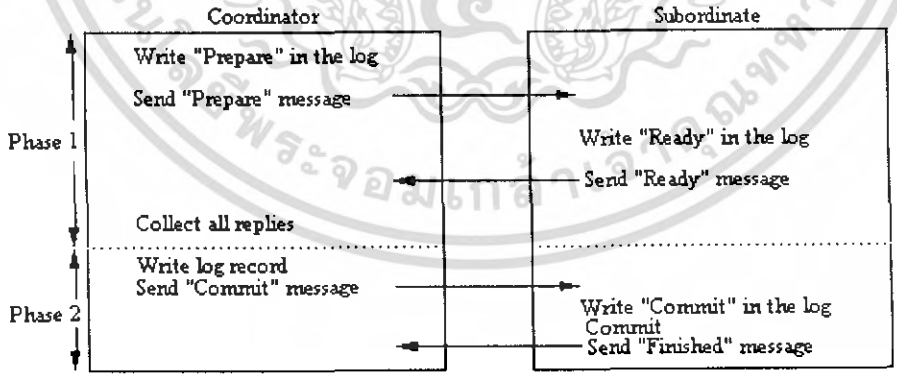
เพียงพอ เพราะฉะนั้นจะต้องมีการเลือกบุคคลที่จะมา maintenance ให้เหมาะสม รวมทั้งมีข้อมูลที่เพียงพอกับระบบ

2.11. โปรโตคอล Secure Socket Layer

SSL หรือ Secure Socket Layer คือ โพรโตคอล (ภาษาที่มีการกำหนดรูปแบบไว้อย่างเป็นกฎเกณฑ์เพื่อการติดต่อสื่อสารของระบบเครือข่าย) ที่ใช้บนเว็บสำหรับสร้างความปลอดภัยในการรับส่งข้อมูลระหว่างเว็บบราวเซอร์ และ เว็บเซิร์ฟเวอร์ โดยมีกลไก คือ การตรวจสอบซึ่งกันและกันระหว่างเว็บบราวเซอร์ และเว็บเซิร์ฟเวอร์ว่าผู้ใช้ทั้งสองฝ่ายเป็นผู้มีสิทธิ์จริง (Authorized parties) แล้วจึงทำการเข้ารหัสข้อมูล (Data Encryption) ที่รับส่งระหว่าง บราวเซอร์และเซิร์ฟเวอร์ เพื่อทำให้ข้อมูลเป็นความลับไม่สามารถถูกลักลอบอ่านได้โดยง่ายจากผู้ที่ไม่มีสิทธิ์ (ในกระบวนการนี้ใบรับรองดิจิตอลจะถูกใช้ในการตรวจสอบ) โดยข้อมูลที่ถูกส่งนั้นจะถูกเข้ารหัสอยู่ในชั้นเครือข่าย (Network) ซึ่งอยู่ระหว่างชั้นโปรแกรมประยุกต์ (Application) กับชั้นทรานสปอร์ต ในโปรโตคอล TCP/IP ตัวอย่างที่นำ SSL มาใช้ ได้แก่ การชำระเงินด้วยบัตรเครดิตโดยการส่งข้อมูลบัตรเครดิตผ่านทางอินเทอร์เน็ต

2.12. ทูเฟสคอมมิท โพรโตคอล Two-Phase Commit Protocol

เป็นโพรโตคอลที่ใช้ดูแลการ commit ให้ทุกๆ site ทำการ commit พร้อมกัน โดยแต่ละ site จะต้องมี Transaction Manager ที่จะคอยดูแลเรื่องการ commit



รูปที่ 2.39. แสดงการทำงานของทูเฟสคอมมิท

ขั้นตอนการทำงาน ดังรูปที่ 2.39. มีขั้นตอนทำงานดังนี้คือ

เฟส 1

- Coordinator ทำการเขียนข้อความ "prepare to commit" ลงใน log file เพื่อเริ่มโปรโตคอล
- ทำการส่งข้อความให้กับแต่ละ participant ที่เกี่ยวข้อง
- ถ้า participant พร้อมที่จะ commit มันจะทำการเขียนข้อความ "ready" ลง log file แล้วส่ง ข้อความ "ready" ไปยัง Coordinator โดยถ้าตัดสินใจจะ commit หรือ abort แล้วจะเปลี่ยนกลับไม่ได้
- Coordinator จะรับข้อความจากแต่ละ participant เพื่อตัดสินใจว่าจะทำ Global Commit หรือ Global Abort

เฟส 2

- โดย Coordinator จะตัดสินใจเป็น Global Abort หากมี participant ใดๆ ตอบกลับเป็น Abort และจะตัดสินใจเป็น Global Commit เมื่อทุกๆ participant ตอบกลับเป็น Commit
- เมื่อตัดสินใจได้แล้วจะส่งข้อความออกไปยังแต่ละ participant
- เมื่อ Coordinator รับ ACK แล้วจึงจะค่อยลงมือเอา record ลง log file ว่า "commit complete" เพื่อป้องกันเรื่อง failure (Failure Detection)
- กรณีที่ participant เกิดล่ม เมื่อกลับเข้าสู่สภาพพร้อมทำงานแล้วจะต้องส่งข้อความไปถาม Coordinator อีกทีว่าเป็น Global Commit หรือ Global Abort

บทที่ 3
การออกแบบระบบการจองตั๋วหลายสายการบิน

ก่อนการพัฒนาเว็บแอพพลิเคชันต้นแบบที่มีความซับซ้อนนั้น ทางทีมของผู้พัฒนาควรที่จะต้องมีการออกแบบโครงสร้างต่างๆให้กับระบบหรือเว็บแอพพลิเคชันที่พัฒนา ก่อนที่จะทำการพัฒนา ซึ่งจะช่วยให้ทีมของผู้พัฒนาสามารถที่จะทำความเข้าใจในตัวระบบและสามารถที่จะทำการพัฒนาหรือแบ่งงานกันทำได้อย่างง่ายมากขึ้นรวมทั้งยังสามารถที่จะทำการบำรุงรักษาในบางส่วนของระบบได้อย่างง่ายและไม่ซับซ้อนอีกด้วย

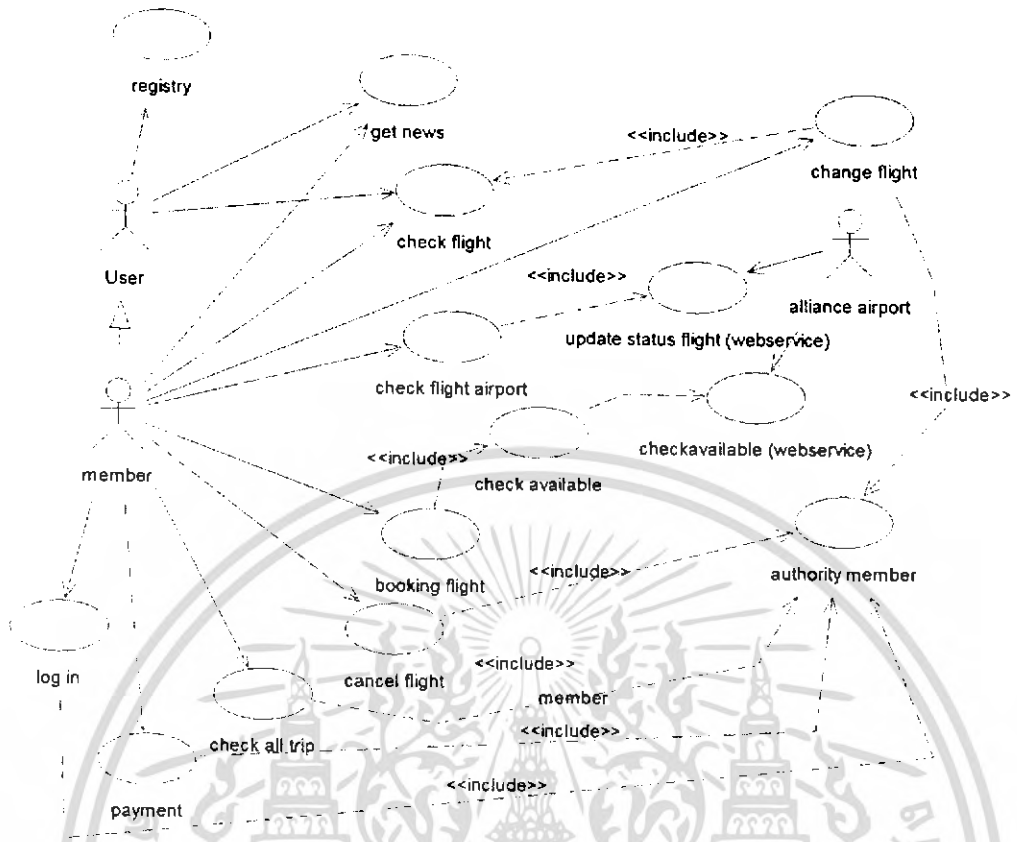
ในบทที่ 3 นี้เราจะกล่าวถึงขั้นตอนการออกแบบก่อนการพัฒนาต้นแบบ ไม่ว่าจะเป็นขั้นตอนของการค้นหาความต้องการ ขั้นตอนการวิเคราะห์และออกแบบ รวมไปถึงการทำการสร้าง (Implement) โดยที่ใช้วิชวลสตูดิโอดอทเน็ต

3.1. ขั้นตอนการเก็บความต้องการและยูสเคส

หัวข้อนี้จะแสดงในส่วนของขั้นตอนการเก็บความต้องการของระบบการจองตั๋วเครื่องบินหลายสายการบินโดยการใช้ยูสเคสเป็นเครื่องมือ โดยความต้องการของเว็บแอพพลิเคชันนี้จะได้รับการบันทึกให้อยู่ในรูปแบบของการโมเดล เอกสารหรือคำอธิบายที่บ่งบอกว่าเว็บแอพพลิเคชันการจองระบบตั๋วเครื่องบินหลายสายการบินนี้ สามารถทำสิ่งใดได้บ้าง

สำหรับการค้นหาและบันทึกความต้องการนั้น ขั้นตอนและกิจกรรมหลักจะอยู่ที่การค้นหาว่าผู้ใช้ต้องการสิ่งใด และเว็บแอพพลิเคชันนี้จะต้องทำอะไรได้บ้าง และเมื่อเราได้รับความต้องการสำหรับเว็บแอพพลิเคชันมาแล้ว ขั้นตอนต่อไปนั้นก็คือเราจะต้องทำการเปลี่ยนความต้องการที่ได้มานั้นให้อยู่ในรูปของยูสเคส พร้อมทั้งยังนำเสนอเงื่อนไขและข้อจำกัดต่างๆของเว็บแอพพลิเคชันนี้

ในการทำการทดลองนี้จะเห็นได้ว่าเราทำการนำความต้องการที่คิดว่าควรจะมีในเว็บแอพพลิเคชันของการจองตั๋วเครื่องบินมาเขียนหรือโมเดลในรูปของยูสเคสดังรูปที่ 3.1.



รูปที่ 3.1. ตัวอย่างของยูสเคสของระบบของตั๋วเครื่องบินหลายสายการบิน

นอกจากนี้แล้วในการใช้ยูสเคสนั้นจะต้องมีการอธิบายความหมายของคำศัพท์ที่มีอยู่ในยูสเคสเพื่อให้นักพัฒนาที่จะมาพัฒนาต่อหรือพัฒนาร่วมกันนั้นสามารถที่จะเข้าใจและเกิดความสอดคล้องกันในกระบวนการพัฒนาได้ โดยการอธิบายคำศัพท์นั้นอาจจะเป็นการอธิบายภายใต้รูปแบบที่ไม่เป็นทางการ คือการแสดงคำศัพท์ พร้อมคำอธิบาย ดังตารางที่ 3.1.

ตารางที่ 3.1. แสดงการอธิบายคำศัพท์ของยูสเคสในเว็บแอพพลิเคชัน

คำศัพท์	คำอธิบาย
User	เป็นบุคคลใดๆก็ได้ ที่สามารถที่จะเข้ามาที่เว็บไซต์ได้ โดยสามารถที่จะอ่านข่าวสารในตัวเว็บได้ รวมทั้งการค้นหาเส้นทางในการเดินทางของประเทศที่จะไปได้อีกด้วย
Member	เป็น user ที่ได้ผ่านขั้นตอนของการสมัครมาแล้วโดยที่จะสามารถทำกิจกรรมต่างๆบนเว็บไซต์ได้เหมือนกันกับ user รวมทั้งยังสามารถที่จะทำการเก็บสะสมระยะทางที่ได้ทำการ

	เดินทางรวมถึงยังมีสิทธิที่จะได้รับโปรโมชันต่างๆในตัวเว็บไซต์ได้
Alliance airport	เป็นสายการบินพันธมิตร ที่จะคอยอัพเดทข้อมูลของที่นั่งของแต่ละเที่ยวบินที่เป็นพันธมิตรกันหรืออัพเดทสถานะการบินในแต่ละสนามบิน
Registry	เป็นขั้นตอนในการสมัครสมาชิกกับทางเว็บไซต์
Get news	เป็นข้อมูลข่าวสารต่างๆที่จะนำขึ้นมานำเสนอบนหน้าเว็บไซต์
Check flight	เป็นการค้นหาเที่ยวบินที่ต้องการจะทำการเดินทางระหว่างประเทศ
Check flight airport	เป็นการตรวจสอบสถานะของเที่ยวบินที่จะทำการออกหรือเข้ามาในสนามบินพันธมิตรต่างๆในแต่ละสนามบิน
Check available	เป็นการตรวจสอบที่นั่งว่ายังมีเหลือหรือพอเพียงต่อการจองของเที่ยวบินนั้นๆหรือไม่
Booking flight	เป็นการจองตั๋วโดยสารของเที่ยวบินที่ได้ทำการเลือกไว้แล้ว
Cancel flight	เป็นการยกเลิกเที่ยวบินที่ได้ทำการจองไปแล้ว
Check all trip	เป็นการตรวจสอบประวัติการเดินทางของสมาชิกทั้งหมด
Payment	เป็นการชำระเงินหลังจากที่ได้ทำการจองไปแล้ว
Log in	เป็นการเข้าสู่ระบบของสมาชิก
Update status airport (webservice)	เป็นการอัพเดทสถานะการบินของสนามบินที่เป็นพันธมิตรกันว่าเที่ยวบินไหนได้ทำการออกหรือลงมายังสนามบินนั้นแล้วหรือยัง
Check available (webservice)	เป็นการตรวจสอบและอัพเดทจำนวนที่นั่งของสนามบินพันธมิตรว่ามีเพียงพอหรือลดน้อยลงเท่าใด
Change flight	เป็นการทำการเปลี่ยนแปลงเที่ยวบินที่ได้ทำการจองไปแล้ว
Authority member	เป็นการตรวจสอบความถูกต้องของการเป็นสมาชิก

3.2. ขั้นตอนการวิเคราะห์ (Analysis)

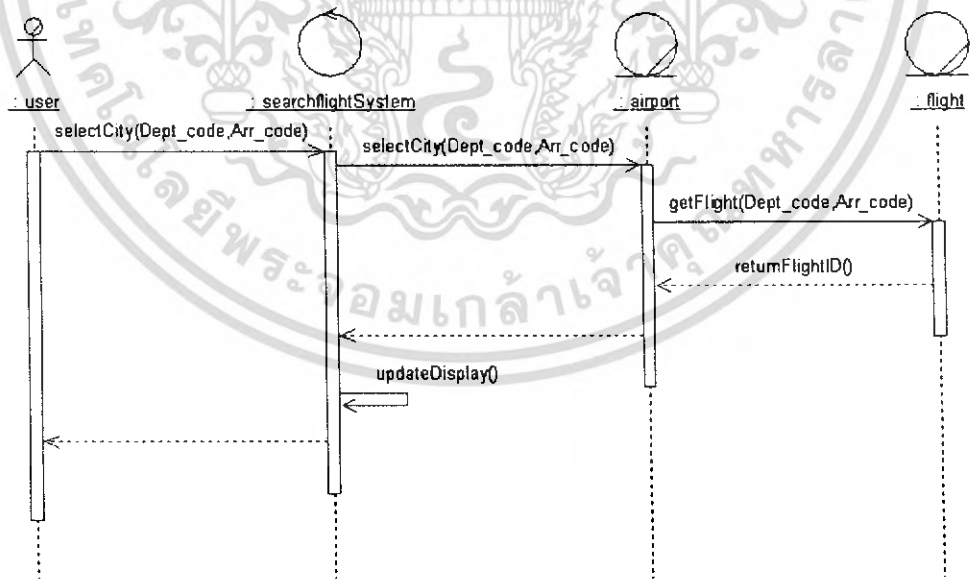
ในขั้นตอนนี้จะประกอบไปด้วยกิจกรรมในการเปลี่ยนแปลงความต้องการของซอฟต์แวร์ให้อยู่ในรูปแบบที่สามารถนำไปใช้ในการออกแบบและสร้างเป็นซอฟต์แวร์จริง

โดยจุดประสงค์ของขั้นตอนการวิเคราะห์นี้ คือ การที่จะทำให้ผู้พัฒนาสามารถที่จะเข้าใจในความต้องการได้อย่างลึกซึ้งมากยิ่งขึ้นซึ่งจะทำให้ผู้พัฒนาสามารถที่จะปรับปรุงและแก้ไขได้ง่าย และในขั้นตอนของการวิเคราะห์นั้นจะใช้ข้อมูลที่ได้จากขั้นตอนของการค้นหาและบันทึกความต้องการมาทำการวิเคราะห์เพื่อค้นหาความต้องการที่ละเอียดมากยิ่งขึ้นภายใต้เงื่อนไขและประเด็นที่สำคัญต่างๆ โดยจะนำโมเดลต่างๆ มาช่วยในการวิเคราะห์ความต้องการที่ละเอียดขึ้นคือแผนภาพที่แสดงการทำงานร่วมกันและแผนภาพแสดงลำดับการทำงาน

ซึ่งในการทำแผนภาพแสดงการทำงานร่วมกันและแผนภาพแสดงลำดับการทำงานนั้นจะแสดงมาจากยูสเคสที่ได้กำหนดความต้องการของระบบการจองตั๋วเครื่องบินหลายสายการบิน ซึ่งจากการวิเคราะห์ระบบนี้แล้วพบว่าเว็บแอพพลิเคชันการจองตั๋วเครื่องบินหลายสายการบินนี้มียูสเคสที่สำคัญคือ การทำการจองตั๋วเครื่องบิน, การค้นหาเที่ยวบิน, การเปลี่ยนแปลงและยกเลิกเที่ยวบิน, การสมัครสมาชิกการตรวจสอบประวัติการเดินทาง, การชำระเงิน และการอัพเดทที่นั่งของสายการบินพันธมิตร ซึ่งจะนำมาเขียนในรูปแบบของแผนภาพแสดงลำดับการทำงานและแผนภาพแสดงการทำงานร่วมกัน

3.2.1. แผนภาพแสดงลำดับการทำงาน

3.2.1.1. ส่วนของการค้นหา (Searching)



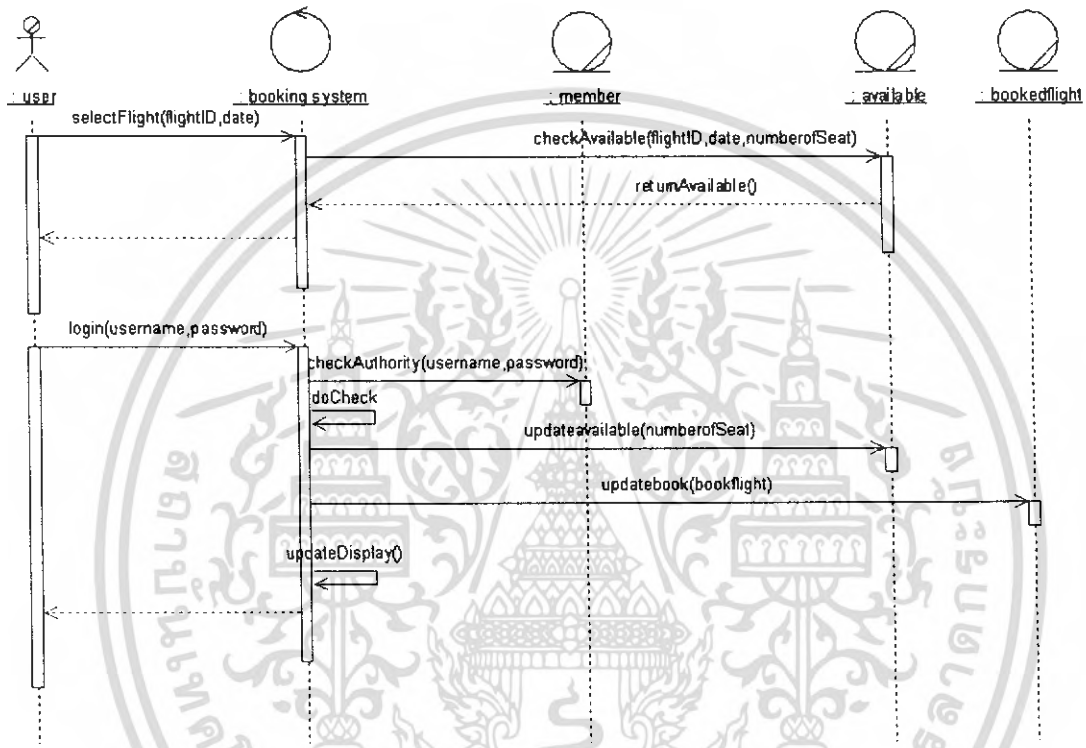
รูปที่ 3.2. แผนภาพแสดงลำดับการทำงานของการค้นหาเที่ยวบิน

ในขั้นตอนของการจองในรูปที่ 3.2. นั้นมีลำดับของการทำงานดังนี้คือ

- ผู้ใช้จะทำการเลือกเมืองที่ต้องการไปแล้วต้นทางที่จะไป

- เมื่อเข้าสู่ระบบการค้นหา ระบบจะทำการไปเรียกรหัสเมืองทั้งขาเข้า-ขาออก และไปทำการเลือกเที่ยวบินที่สามารถเดินทางจากต้นทางไปยังปลายทางได้
- เมื่อได้ผลของเที่ยวบินมาแล้วระบบก็จะแสดงเที่ยวบินที่เป็นไปได้ให้กับผู้ใช้ได้เลือกต่อไป

3.2.1.2. ส่วนของการจอง (Booking)



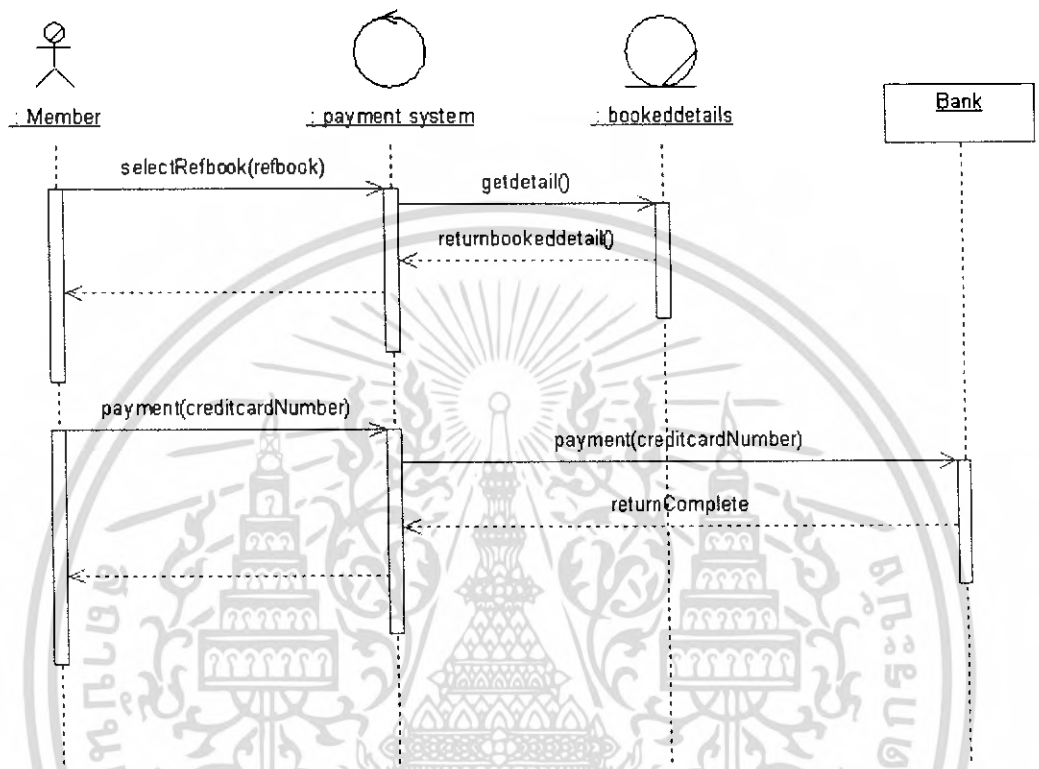
รูปที่ 3.3. แผนภาพแสดงลำดับการทำงานของการจองตั๋วเครื่องบิน

ในขั้นตอนของการจองในรูปที่ 3.3. นั้นมีขั้นตอนการทำงานดังนี้

- ผู้ใช้ทำการเลือกเที่ยวบินที่ได้ผลมาจากการค้นหานั้นโดยจะส่งหมายเลขเที่ยวบินที่ทำการเลือกกับวันที่ทำการเลือกไปให้กับระบบของการจอง
- ระบบการจองจะทำการไปตรวจสอบหมายเลขที่นั่งของหมายเลขเที่ยวบินในวันนั้นๆ
- เมื่อตรวจสอบและพบว่ามีที่ว่างแล้วนั้นก็จะทำการแจ้งกลับไปให้ผู้ใช้เพื่อรอการยืนยัน
- ผู้ใช้ทำการยืนยันโดยการกรอกหมายเลขสมาชิกและรหัสสมาชิก
- ระบบของการจองจะทำการตรวจสอบความเป็นสมาชิกของผู้ใช้ ถ้าผ่านระบบของการจองก็จะส่งข้อมูลไปทำการอัพเดทจำนวนที่นั่ง (available)และเพิ่มหมายเลขของตั๋วที่ (bookedflight)

- เมื่อทำการจองเสร็จเรียบร้อยแล้วก็จะแสดงผลออกทางหน้าจอว่ากระบวนการจองนั้นเสร็จสิ้นแล้ว

3.2.1.3. ส่วนของการชำระเงิน (Payment)

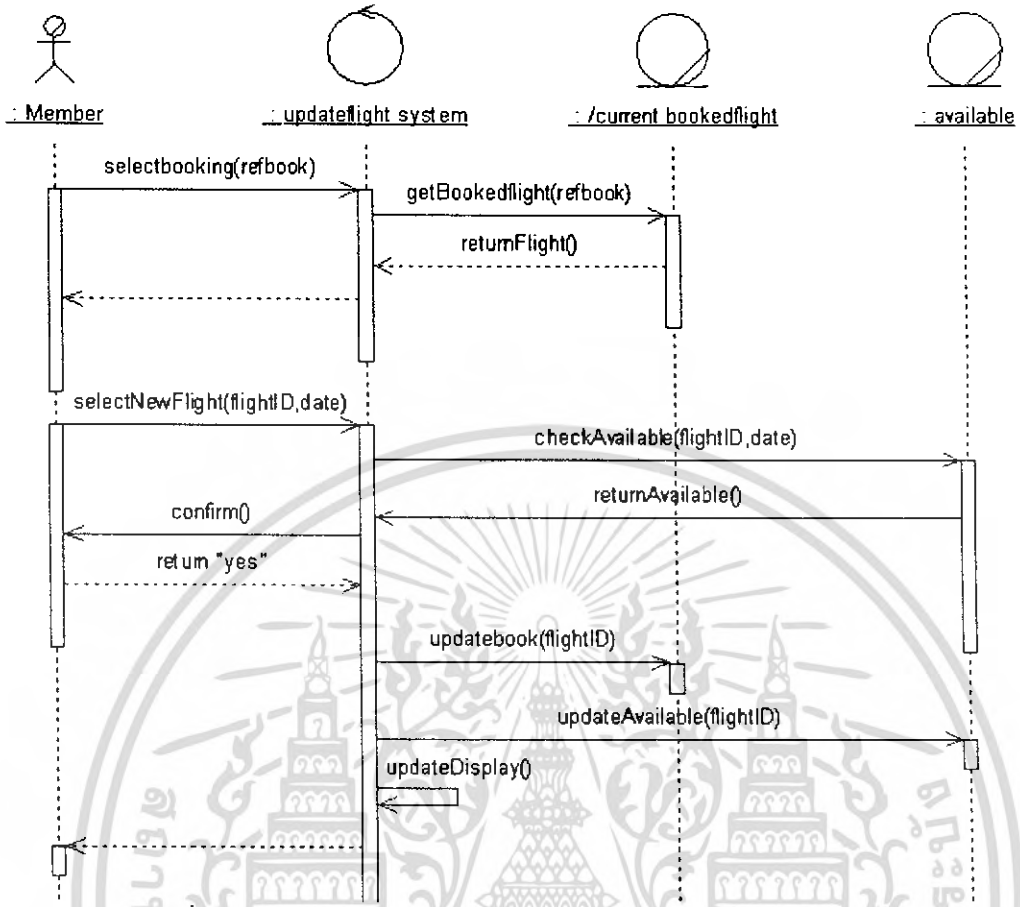


รูปที่ 3.4. แผนภาพแสดงลำดับการทำงานของการชำระเงินของการจองตั๋ว

ในขั้นตอนของการจองในรูปที่ 3.4. นั้นมีขั้นตอนการทำงานดังนี้

- สมาชิกที่ได้ทำการจองไปแล้วนั้นจะมีหมายเลข Booking Reference เพื่อเอาไปใช้อ้างอิงสำหรับการทำธุรกรรมต่างๆ ซึ่งการชำระเงินนี้ สมาชิกต้องกรอกหมายเลข Booking reference ให้กับระบบการชำระเงินก่อน
- ระบบจะทำการค้นหารายละเอียดของการเดินทางจากหมายเลข Booking reference ใน Bookeddetails จะทำให้ระบบและผู้ชำระเงินทราบถึงราคาที่ต้องชำระ
- สมาชิกต้องทำการกรอกหมายเลขบัตรเครดิตของตัวเองให้กับระบบการชำระเงิน
- ระบบการชำระเงินก็จะไปเรียกใช้บริการของการหักเงินผ่านบัญชีของธนาคารและส่งผลคืนกับมาให้ระบบการชำระเงินว่าเสร็จสิ้นแล้ว

3.2.1.4. ส่วนของการเปลี่ยนแปลงเที่ยวบิน

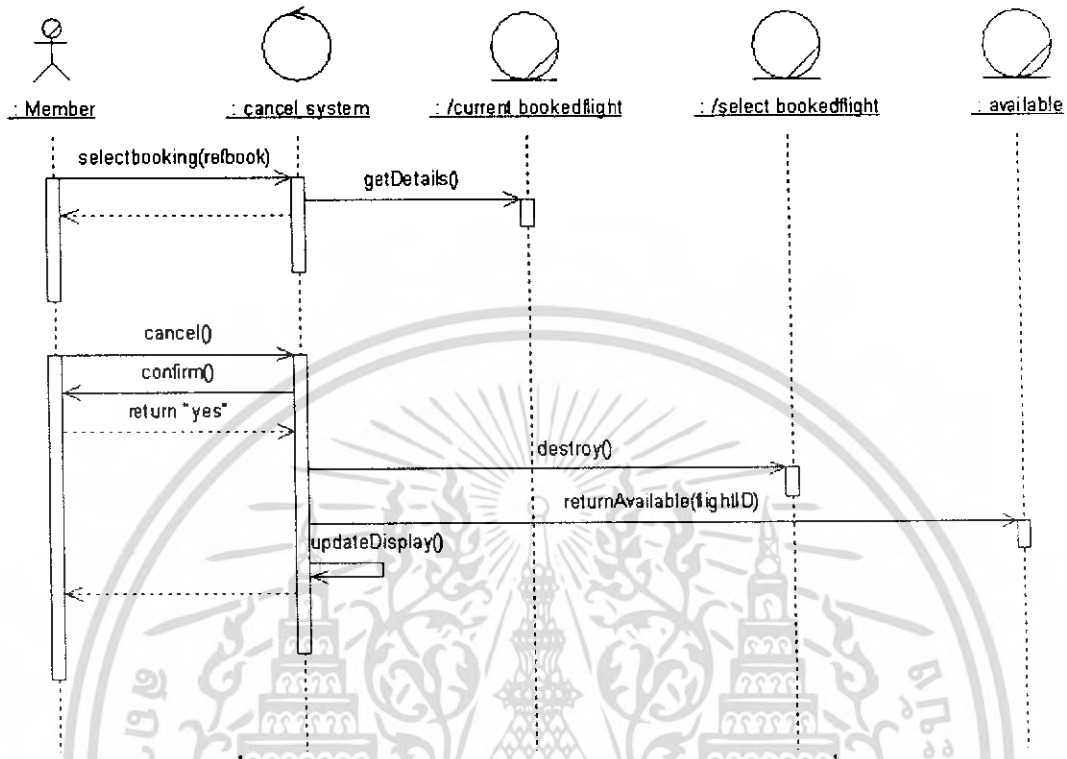


รูปที่ 3.5. แผนภาพแสดงลำดับการทำงานของการเปลี่ยนแปลงเที่ยวบิน

ในขั้นตอนของการจองในรูปที่ 3.5. นั้นมีขั้นตอนการทำงานดังนี้

- สมาชิกจะต้องทำการกรอกหมายเลขของ Booking reference ให้กับระบบของการเปลี่ยนแปลง
- ระบบของการเปลี่ยนแปลงจะทำการค้นหาเที่ยวบินที่ได้ทำการจองไว้จากหมายเลข Booking reference ที่ได้กรอกเข้ามาให้ระบบและแสดงให้กับสมาชิก
- สมาชิกจะทำการเลือกเที่ยวบินใหม่โดยจะส่งค่าของหมายเลขเที่ยวบินใหม่กับวันที่ที่ทำการเปลี่ยนแปลงเที่ยวบินให้กับระบบเปลี่ยนแปลงเที่ยวบิน
- ระบบการเปลี่ยนแปลงเที่ยวบินจำทำการตรวจสอบจำนวนที่นั่งของเที่ยวบินและส่งผลของของการตรวจสอบมาให้กับระบบของการเปลี่ยนแปลงแล้วระบบจะทำการถามถึงการยืนยันการเปลี่ยนแปลงกับสมาชิก
- เมื่อสมาชิกทำการยืนยันการเปลี่ยนแปลง ระบบการเปลี่ยนแปลงจะทำการอัพเดทเที่ยวบินใหม่ที่ทำการเปลี่ยนใน Bookedflight และ available และทำการส่งผล
- กลับมาให้กับสมาชิก

3.2.1.5. ส่วนของการยกเลิกเที่ยวบิน

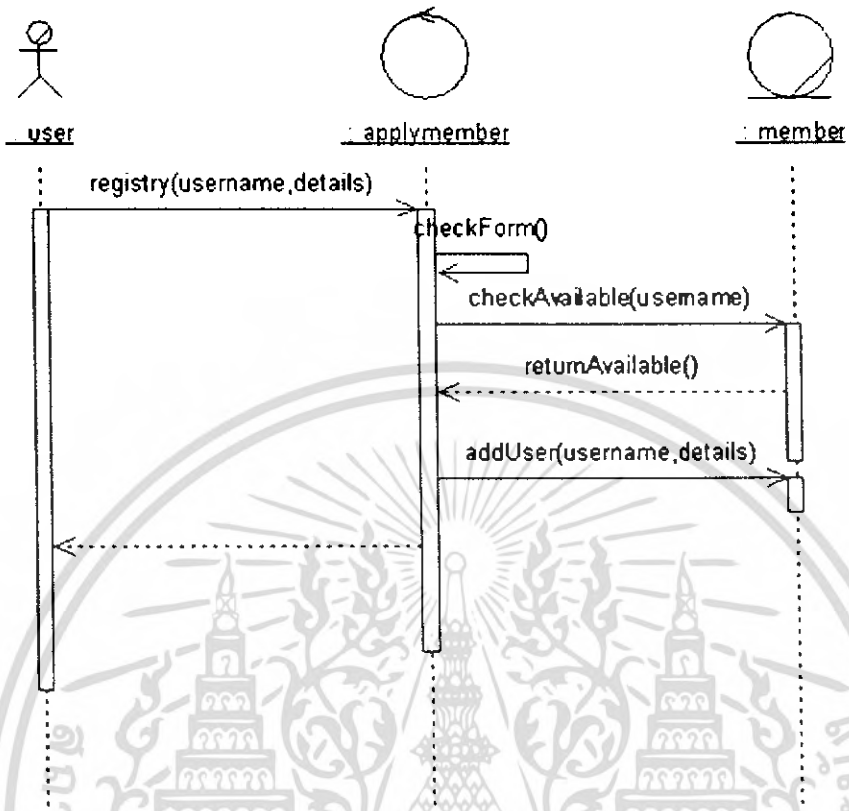


รูปที่ 3.6. แผนภาพแสดงลำดับการทำงานของการยกเลิกเที่ยวบิน

ในขั้นตอนของการจองในรูปที่ 3.6. นั้นมีขั้นตอนการทำงานดังนี้

- สมาชิกจะต้องทำการกรอกหมายเลขของ Booking Reference ให้กับระบบของการยกเลิกเที่ยวบิน
- ระบบของการยกเลิกจะทำการค้นหารายละเอียดของเที่ยวบินที่ได้ทำการจองไว้จากหมายเลข Booking reference ที่ได้กรอกเข้ามาให้ระบบและแสดงให้กับสมาชิก
- สมาชิกสั่งทำการยกเลิกเที่ยวบินโดยบอกไปที่ระบบการยกเลิก
- ระบบการยกเลิกจะรอการยืนยันการยกเลิกอีกครั้งหนึ่ง
- เมื่อสมาชิกทำการยืนยันการยกเลิกแล้วระบบการยกเลิกจะไปทำการอัพเดทหรือยกเลิกตั๋วของการเดินทางและที่นั่งของเที่ยวบินนั้นบน Bookedflight และ available ตามลำดับ และจะส่งผลของการยกเลิกเที่ยวบินให้กับสมาชิก

3.2.1.6. ส่วนของการสมัครสมาชิก

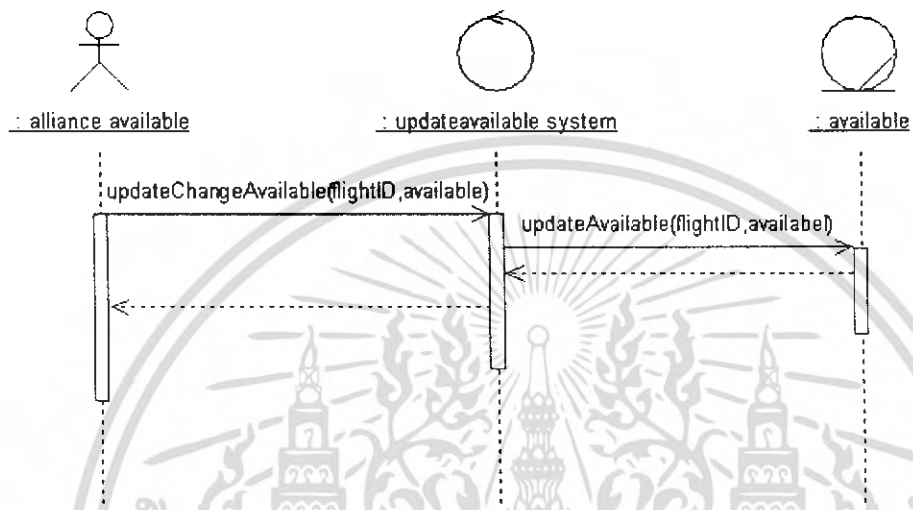


รูปที่ 3.7. แผนภาพแสดงลำดับของการสมัครสมาชิก

ในขั้นตอนของการจองในรูปที่ 3.7. นั้นมีขั้นตอนการทำงานดังนี้

- ผู้ใช้จะทำการกรอกชื่อสมาชิกพร้อมรายละเอียดต่างๆที่เกี่ยวข้องกับสมาชิกให้กับระบบการสมัครสมาชิก
- ระบบสมัครสมาชิกจะทำการตรวจสอบชื่อของผู้ใช้ว่ามีการซ้ำกันหรือไม่ ถ้าไม่ระบบก็จะทำการเพิ่มสมาชิกให้กับ Member และส่งผลของการสมัครสมาชิกกลับไปให้ผู้ใช้

3.2.1.7. ส่วนของการอัพเดทของจำนวนที่นั่งจากสายการบินพันธมิตร



รูปที่ 3.8. แผนภาพแสดงลำดับการทำงานของการอัพเดทจำนวนที่นั่งจากสายการบินพันธมิตร

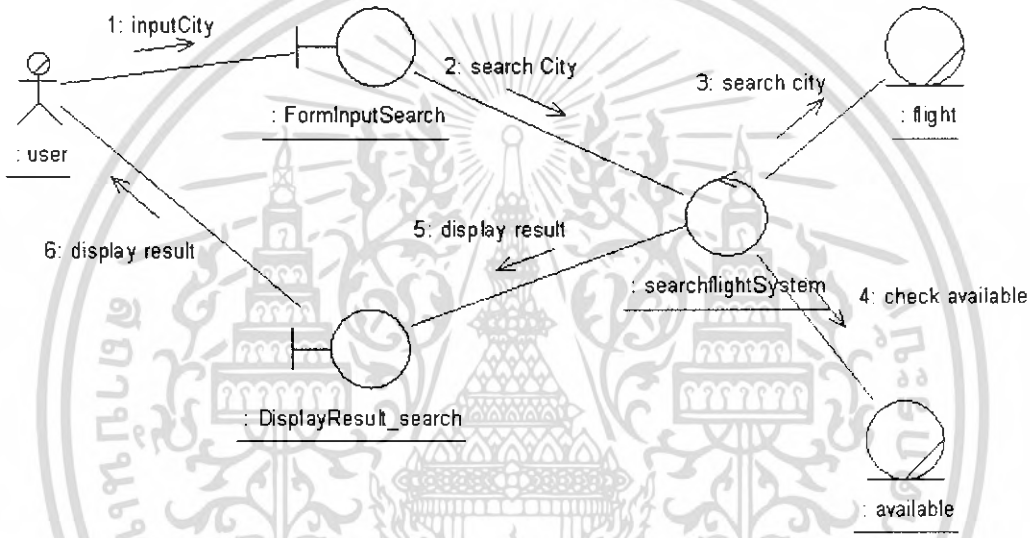
ในขั้นตอนของการจองในรูปที่ 3.8. นั้นมีขั้นตอนการทำงานดังนี้

- สายการบินพันธมิตรทำการอัพเดทเที่ยวบินที่มีการเปลี่ยนแปลงโดยมีการเรียกใช้เทคโนโลยีเว็บเซอร์วิสเพื่อมาทำการอัพเดทข้อมูลลงในเว็บแอพพลิเคชัน ในส่วนของจำนวนที่นั่ง
- เมื่อทำการอัพเดทเสร็จแล้วก็จะส่งค่ากลับไปที่สายการบินพันธมิตรว่าการทำงานเสร็จสมบูรณ์แล้ว

3.2.2. แผนภาพแสดงการทำงานร่วมกัน

3.2.2.1. ส่วนของการค้นหา (Searching)

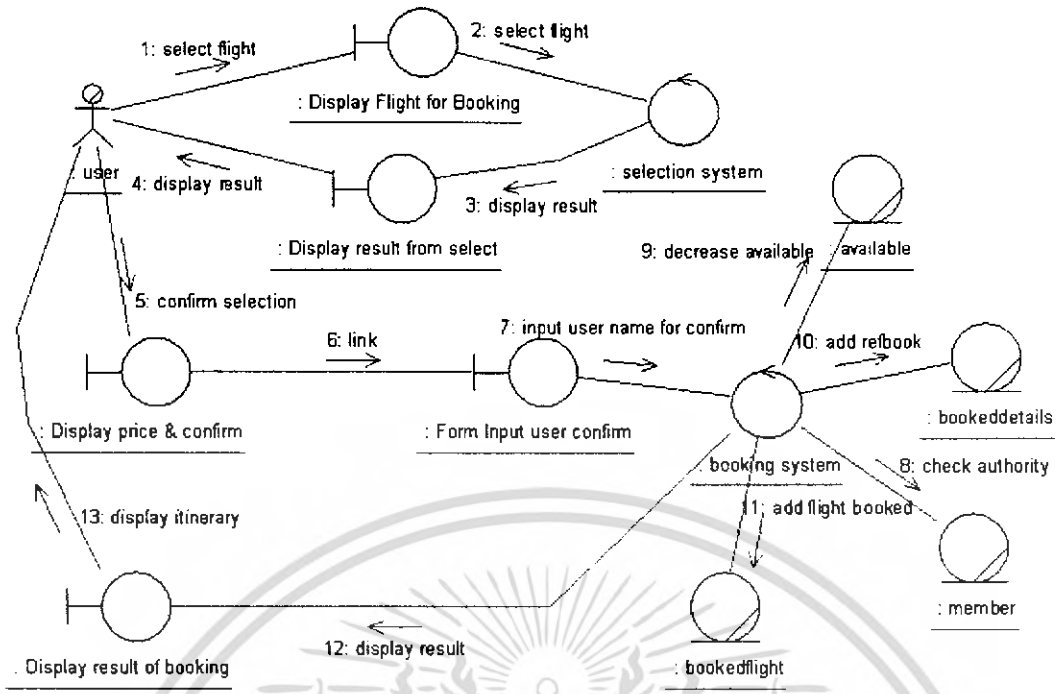
คำอธิบายลำดับการไหลของเหตุการณ์สำหรับการค้นหาสายการบินนี้ คือ ผู้ใช้จะทำการเลือกเมืองที่ต้องการโดยกรอกลงฟอร์มเพื่อเลือกเมืองต้น- ปลายทางที่ต้องการและรายละเอียดต่างๆ ระบบจะทำการค้นหาและตรวจสอบจำนวนที่นั่งสำหรับเที่ยวบินที่เพียงพอ แล้วจึงส่งค่ากลับมาที่หน้าจอให้กับผู้ใช้ต่อไป ดังรูปที่ 3.9.



รูปที่ 3.9. แผนภาพแสดงการทำงานร่วมกันของการค้นหาสายการบิน

3.2.2.2. ส่วนของการจองตั๋วโดยสาร

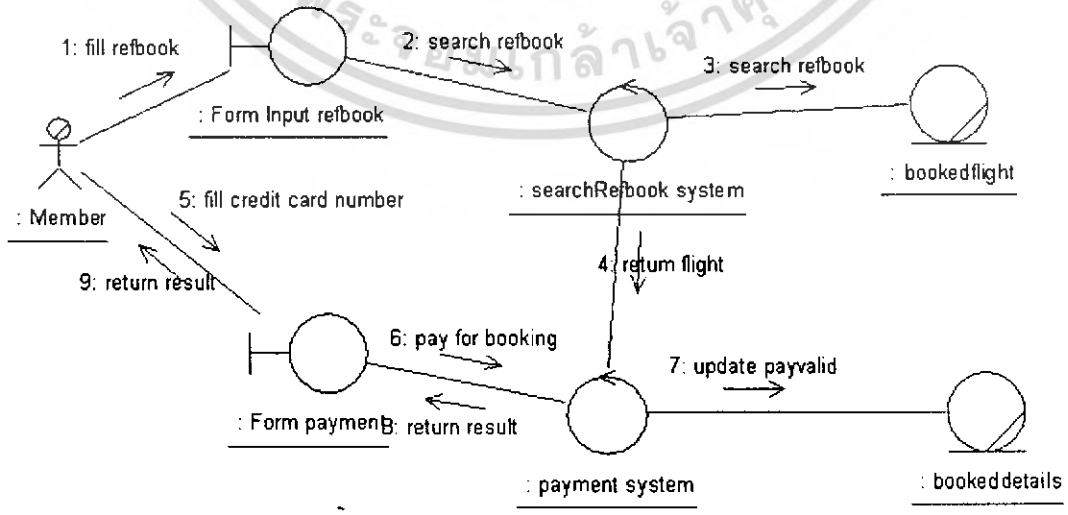
คำอธิบายลำดับการไหลของเหตุการณ์การจองตั๋วโดยสารนั้น คือ ผู้ใช้หรือสมาชิกจะทำการเลือกเที่ยวบินที่ต้องการจากผลของการค้นหา แล้วระบบจะทำการส่งผลของการเลือกกลับไปให้กลับผู้ใช้เพื่อขอยืนยันการเลือกเที่ยวบินนั้นๆ และเมื่อผู้ใช้ทำการยืนยันการเลือกแล้วก็จะเข้าไปสู่หน้าของการกรอกชื่อและรหัสผ่านของผู้ใช้เพื่อทำการยืนยันสิทธิ์ของผู้ที่จะทำการจองให้ระบบโดยระบบการจองก็จะทำการตรวจสอบสิทธิ์ของผู้ใช้ก่อนแล้วค่อยไปทำการตรวจจำนวนที่นั่งของเที่ยวบินนั้นๆ เมื่อการตรวจสอบผ่านแล้ว ระบบการจองก็จะทำการจองโดยทำการสร้างหมายเลขตั๋ว (Booking Reference) แล้วเพิ่มเข้าไปในฐานข้อมูลของ Bookeddetails และ Bookedflight แล้วจึงส่งผลของการทำการจองกลับออกมาให้กับผู้ใช้ทางหน้าเว็บอีกครั้ง ดังรูปที่ 3.10.



รูปที่ 3.10. แผนภาพแสดงการทำงานร่วมกันของการจองเที่ยวบิน

3.2.2.3. ส่วนของการชำระเงิน (payment)

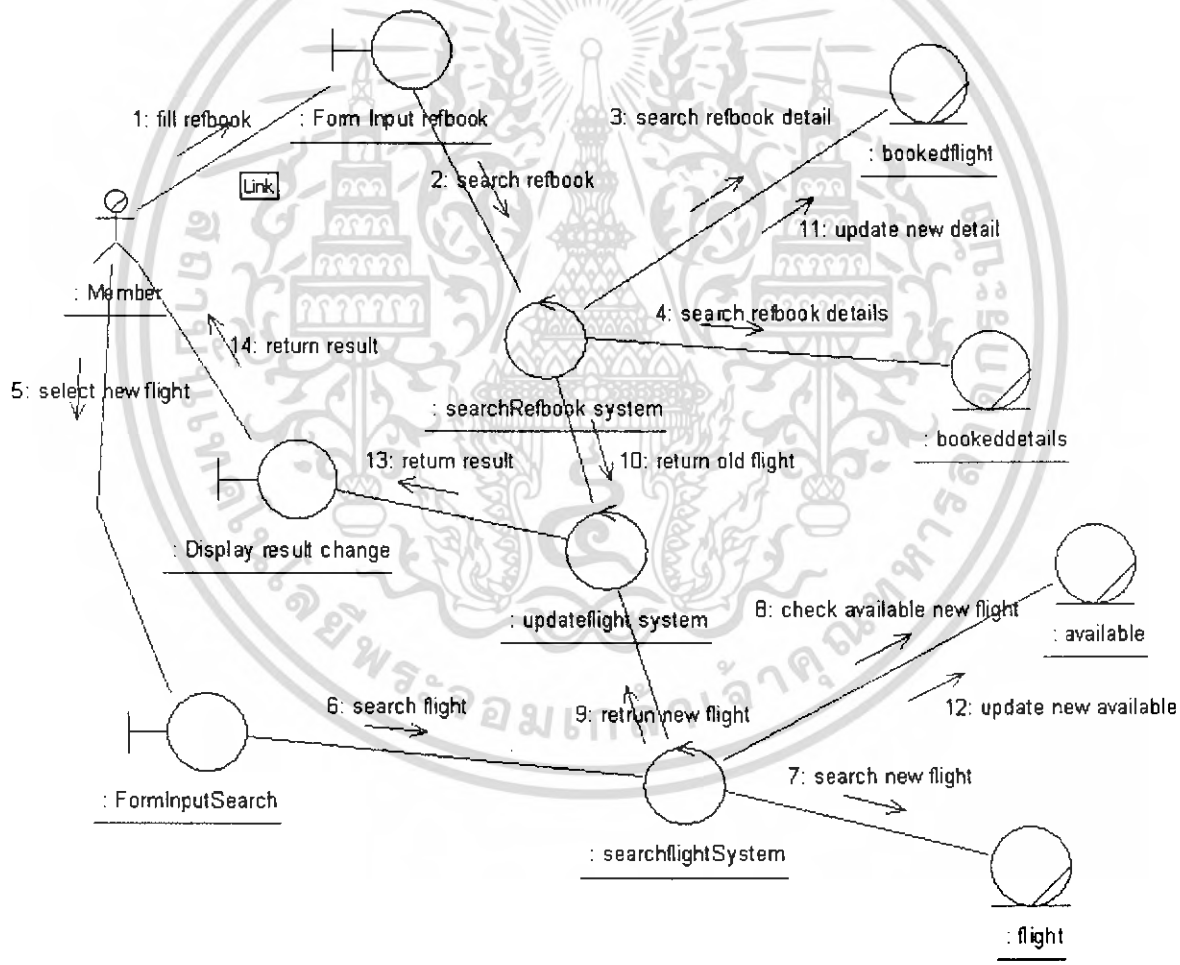
คำอธิบายลำดับการไหลของเหตุการณ์การชำระเงินนั้น คือ ขั้นตอนแรกสมาชิกจะทำการกรอกหมายเลขตั๋วหรือ Booking Reference เพื่อให้ระบบทำการค้นหาเที่ยวบินและรายละเอียดของเที่ยวบิน แล้วสมาชิกก็จะทำการใส่หมายเลขของบัตรเครดิตการ์ดเพื่อให้ระบบทำการชำระเงินผ่านทางบริการของธนาคารและเมื่อทำการชำระเสร็จเรียบร้อยแล้วระบบการชำระเงินจะทำการอัพเดทของตั๋วที่ได้ทำการจ่ายเงินไว้แล้ว และส่งผลลัพธ์ของการชำระเงินให้กับสมาชิกบนหน้าเว็บอีกครั้ง ดังรูปที่ 3.11.



รูปที่ 3.11. แผนภาพแสดงการทำงานร่วมกันของการชำระตั๋วโดยสาร

3.2.2.4. ส่วนของการเปลี่ยนแปลงเที่ยวบิน (Changing flight)

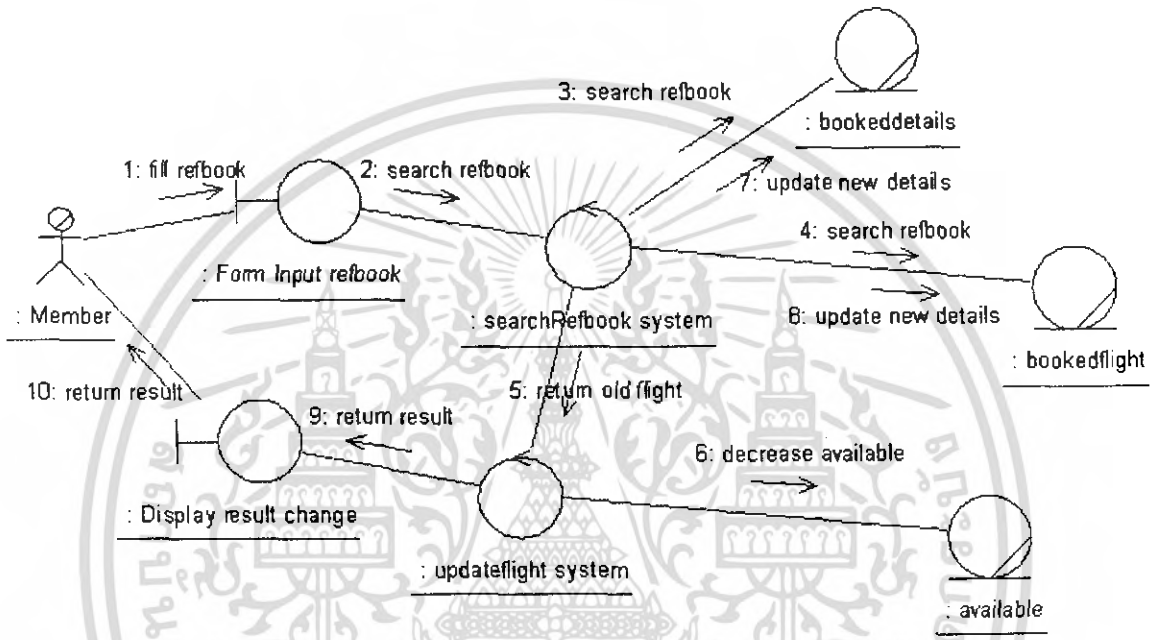
คำอธิบายลำดับการไหลของเหตุการณ์การเปลี่ยนแปลงเที่ยวบินนั้น คือขั้นตอนแรก สมาชิกจะทำการกรอกหมายเลขตั๋วหรือ Booking Reference เพื่อให้ระบบทำการค้นหาเที่ยวบินและรายละเอียดของเที่ยวบินที่ทำการจองไว้ก่อนหน้านี้ แล้วสมาชิกจะทำการค้นหาเที่ยวบินอีกครั้งโดยใส่เงื่อนไขที่ต้องการใหม่แต่ไม่สามารถที่จะเปลี่ยนแปลงต้นทางและปลายทางของเที่ยวบินได้ ต่อจากนั้นระบบจะทำการค้นหาเที่ยวบินที่เหมาะสมให้ใหม่อีกครั้งโดยจะต้องไปค้นหาเที่ยวบินและทำการตรวจสอบที่นั่งของเที่ยวบินที่เหมาะสม เมื่อสมาชิกเลือกเที่ยวบินใหม่แล้วระบบก็จะทำการอัพเดทจำนวนที่นั่งรวมทั้งรายละเอียดของเที่ยวบินที่เปลี่ยนไปโดยหมายเลขตั๋วหรือ Booking Referenceยังคงเป็นหมายเลขเดิม และทำการส่งผลลัพธ์ไปให้กับสมาชิกอีกครั้งหนึ่ง ดังรูปที่ 3.12.



รูปที่ 3.12. แผนภาพแสดงการทำงานร่วมกันของการเปลี่ยนแปลงเที่ยวบิน

3.2.2.5. ส่วนของการยกเลิกเที่ยวบิน (Cancel flight)

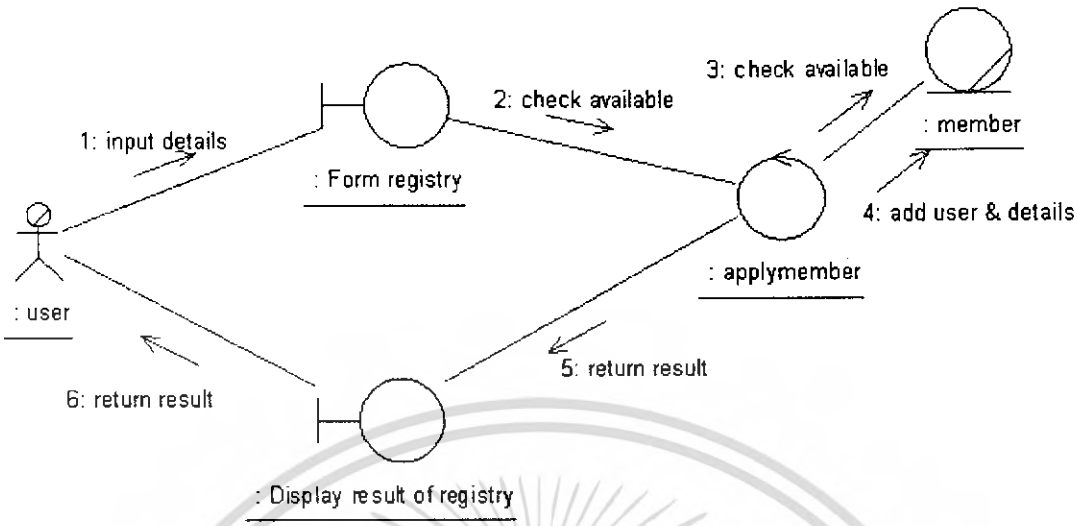
คำอธิบายลำดับการไหลของเหตุการณ์การยกเลิกเที่ยวบินนั้น คือขั้นตอนแรกสมาชิกจะทำการกรอกหมายเลขตั๋วหรือ Booking Reference เพื่อให้ระบบทำการค้นหาเที่ยวบินและรายละเอียดของเที่ยวบินที่ทำการจองไว้ก่อนหน้านี้ แล้วสมาชิกจะทำการยกเลิกเที่ยวบินที่ได้ทำการจองไปแล้วนั้น ต่อจากนั้นระบบก็จะทำการอัพเดทจำนวนที่นั่งรวมทั้งรายละเอียดของเที่ยวบินที่เปลี่ยนไป และทำการส่งผลลัพธ์ไปให้กับสมาชิกอีกครั้งหนึ่ง ดังรูปที่ 3.13.



รูปที่ 3.13. แผนภาพแสดงการทำงานร่วมกันของการยกเลิกเที่ยวบิน

3.2.2.6. ส่วนของการสมัครสมาชิก (Registry)

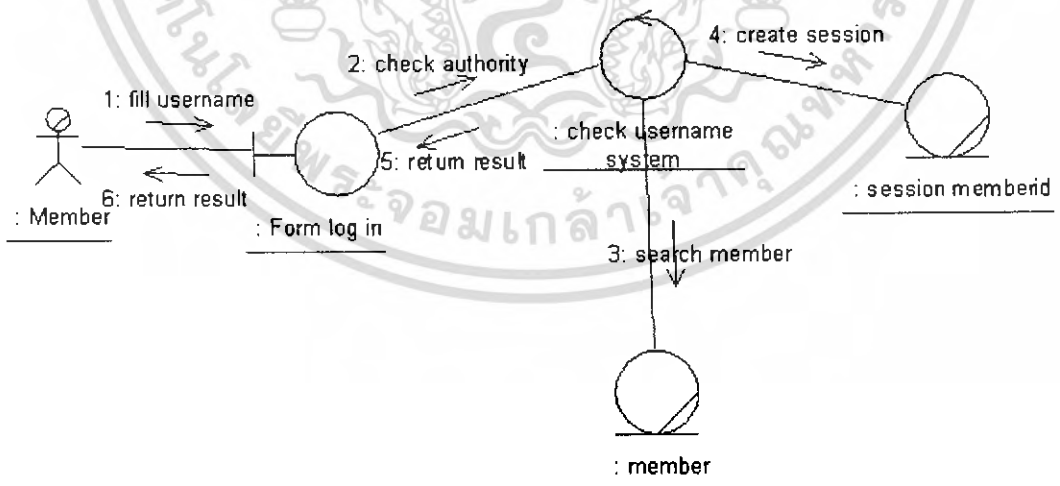
คำอธิบายลำดับการไหลของเหตุการณ์การสมัครสมาชิกนั้น คือขั้นตอนแรกผู้ใช้จะทำการกรอกข้อมูลรายละเอียดต่างๆรวมทั้งชื่อของผู้ใช้และรหัสผ่านที่ผู้ใช้ได้ตั้งขึ้น ระบบการสมัครจะทำการตรวจสอบว่ามีสมาชิกคนไหนเคยใช้ชื่อนี้หรือไม่ ถ้าไม่มีก็จะทำการ เพิ่มชื่อสมาชิกพร้อมรายละเอียดต่างๆลงในส่วนของสมาชิกและจะส่งผลกลับไปให้กับผู้ใช้อีกครั้งหนึ่ง ดังรูปที่ 3.14.



รูปที่ 3.14. แผนภาพแสดงการทำงานร่วมกันของการสมัครสมาชิก

3.2.2.7. ส่วนของการเข้าสู่ระบบสมาชิก (log in)

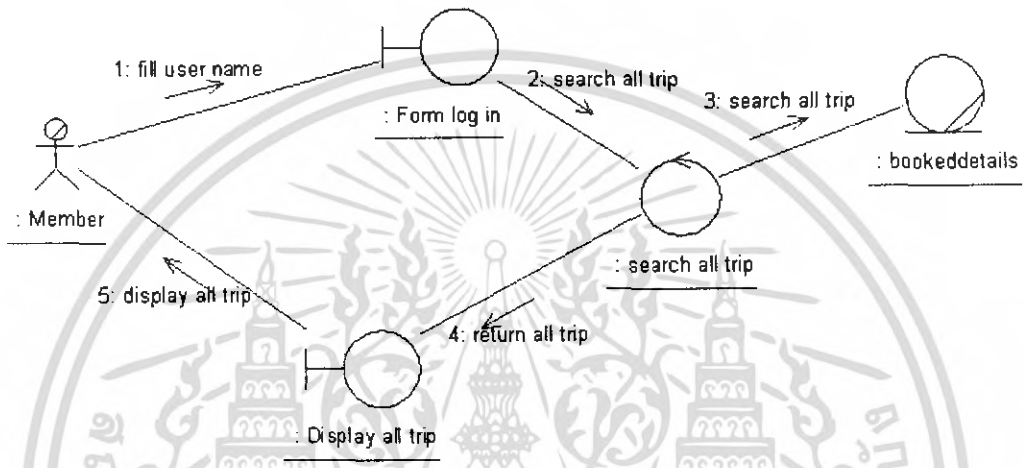
คำอธิบายลำดับการไหลของเหตุการณ์การเข้าสู่ระบบสมาชิกนั้น คือ ขั้นตอนแรกสมาชิกจะทำการใส่ชื่อสมาชิกและรหัสสมาชิกผ่านหน้าเว็บล็อกอินแล้ว ระบบจะทำการตรวจสอบชื่อและรหัสผ่านกับทางฐานข้อมูลและทำการสร้าง เซสชั่นไอดี (Session memberID) ของสมาชิกไว้ โดยขั้นตอนสุดท้ายก็จะแสดงผลลัพธ์ของการล็อกอินกลับไปให้กลับสมาชิกอีกครั้ง ดังรูปที่ 3.15



รูปที่ 3.15. แผนภาพแสดงการทำงานร่วมกันของการเข้าสู่ระบบสมาชิก

3.2.2.8. ส่วนของการตรวจสอบประวัติการเดินทาง (Checking all trip)

คำอธิบายลำดับการไหลของเหตุการณ์การตรวจสอบประวัติการเดินทางนั้น คือ ขั้นตอนแรกสมาชิกจะทำการใส่ชื่อสมาชิกและรหัสผ่านเข้าไปสู่ระบบการตรวจสอบประวัติการเดินทาง ต่อจากนั้นระบบจะทำการตรวจสอบประวัติการเดินทางจากชื่อของสมาชิกในฐานข้อมูลที่เก็บรายละเอียดของการจองที่ผ่านมา และก็จะทำการแสดงประวัติของการเดินทางที่ผ่านมานั้นให้กับสมาชิกที่ทำการตรวจสอบประวัติการเดินทาง ดังรูปที่ 3.16.



รูปที่ 3.16. แผนภาพแสดงการทำงานร่วมกันของการตรวจสอบประวัติการเดินทาง

3.3. ขั้นตอนการออกแบบ (Design)

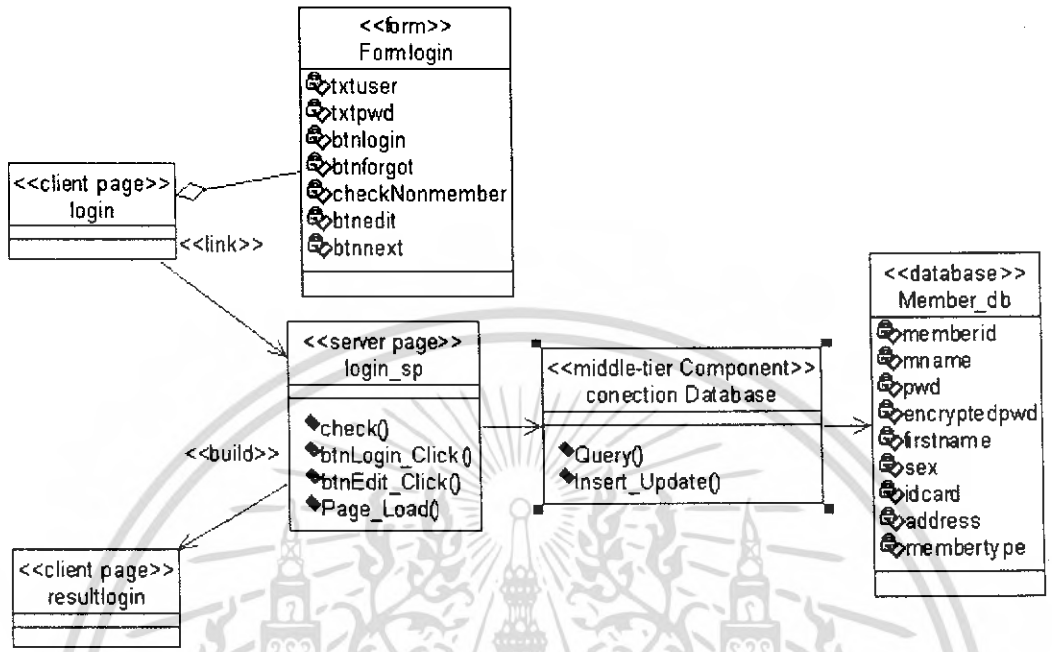
ขั้นตอนของการออกแบบเว็บแอพพลิเคชันนี้นั้นผู้พัฒนาจะต้องทำความเข้าใจกับความต้องการของระบบอย่างละเอียดก่อนจากขั้นตอนของการวิเคราะห์ และจะต้องแปลงความต้องการเหล่านั้นให้สามารถนำไปสร้างเป็นเว็บแอพพลิเคชั้นที่ใช้ทำงานจริงได้โดยโมเดลที่จะมาช่วยในขั้นตอนการออกแบบนี้เป็น แผนภาพแสดงความสัมพันธ์ระหว่างคลาส

โดยในการออกแบบเว็บแอพพลิเคชันของการจองตั๋วเครื่องบินนี้ได้ออกแบบภายใต้สถาปัตยกรรมแบบ Thin web client ซึ่งการออกแบบส่วนใหญ่จะอยู่ที่ส่วนของเซิร์ฟเวอร์เพจเป็นหลัก โดยเทคโนโลยีที่เกียวข้องก็จะเป็นเทคโนโลยีฝั่งไคลเอ็นต์, สคริปฝั่งเซิร์ฟเวอร์, เทคโนโลยีของคอมโพเนนท์ที่เทียร์กลางและเทคโนโลยีของระบบฐานข้อมูลในการออกแบบ

โดยเราจะทำการยกตัวอย่างเพียงบางระบบที่มีความสำคัญและเป็นส่วนหลักของเว็บแอพพลิเคชันการจองตั๋วเครื่องบินหลายสายการบิน

3.3.1. แผนภาพคลาส (Class Diagram)

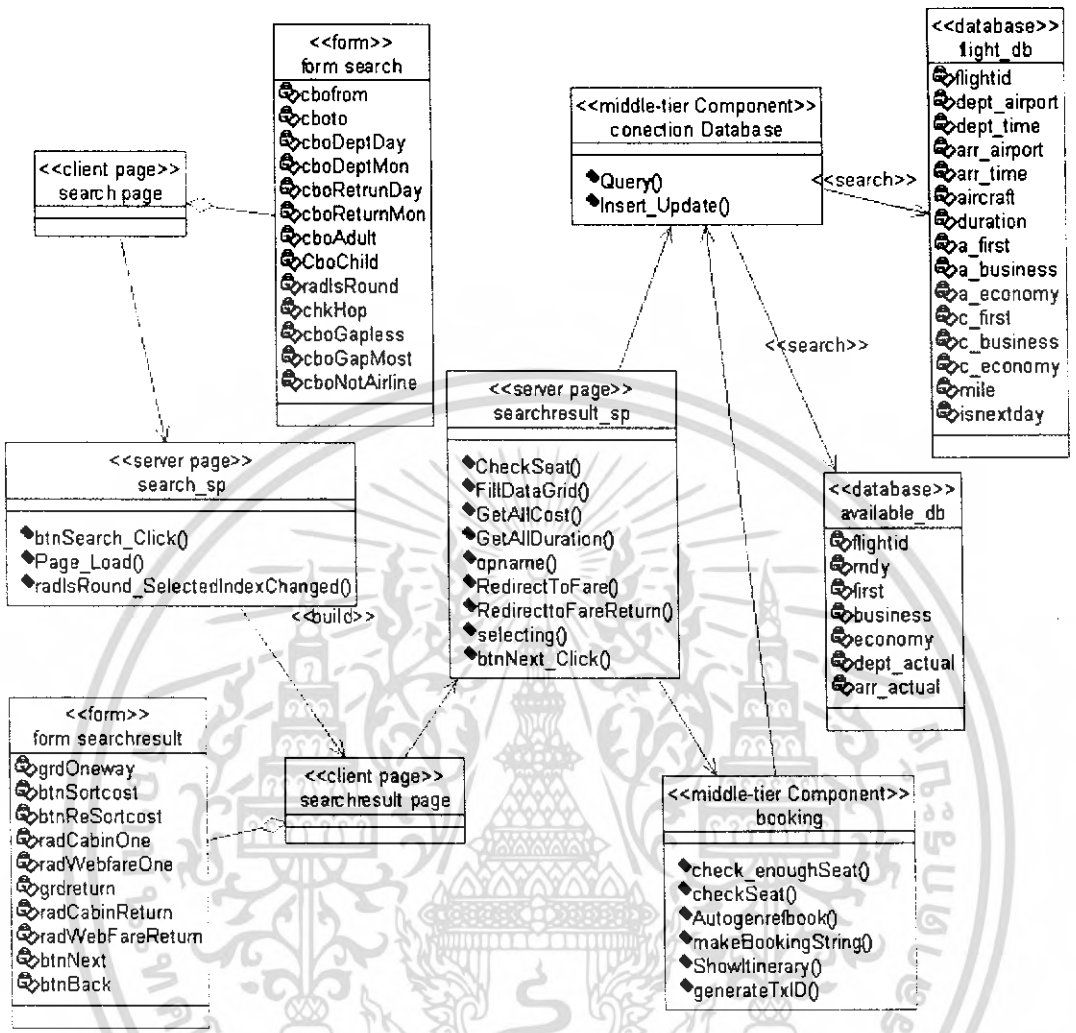
3.3.1.1 ขั้นตอนการเข้าสู่ระบบสมาชิก



รูปที่ 3.17. แผนภาพคลาสแสดงการเข้าสู่ระบบสมาชิก

จากรูปที่ 3.17 นั้น สามารถอธิบายได้ดังนี้คือ คลาสไดลเอ็นต์เพจ Login ซึ่งจะประกอบไปด้วยฟอร์ม FormLogin และมีหน้าเซิร์ฟเวอร์เพจ Login_sp ที่จะจัดการในเรื่องการตรวจสอบและเพิ่มสมาชิกไปที่ ฐานข้อมูล member_db โดยเรียกใช้คอมโพแนนท์เทียร์กลางที่ชื่อว่า Connection Database

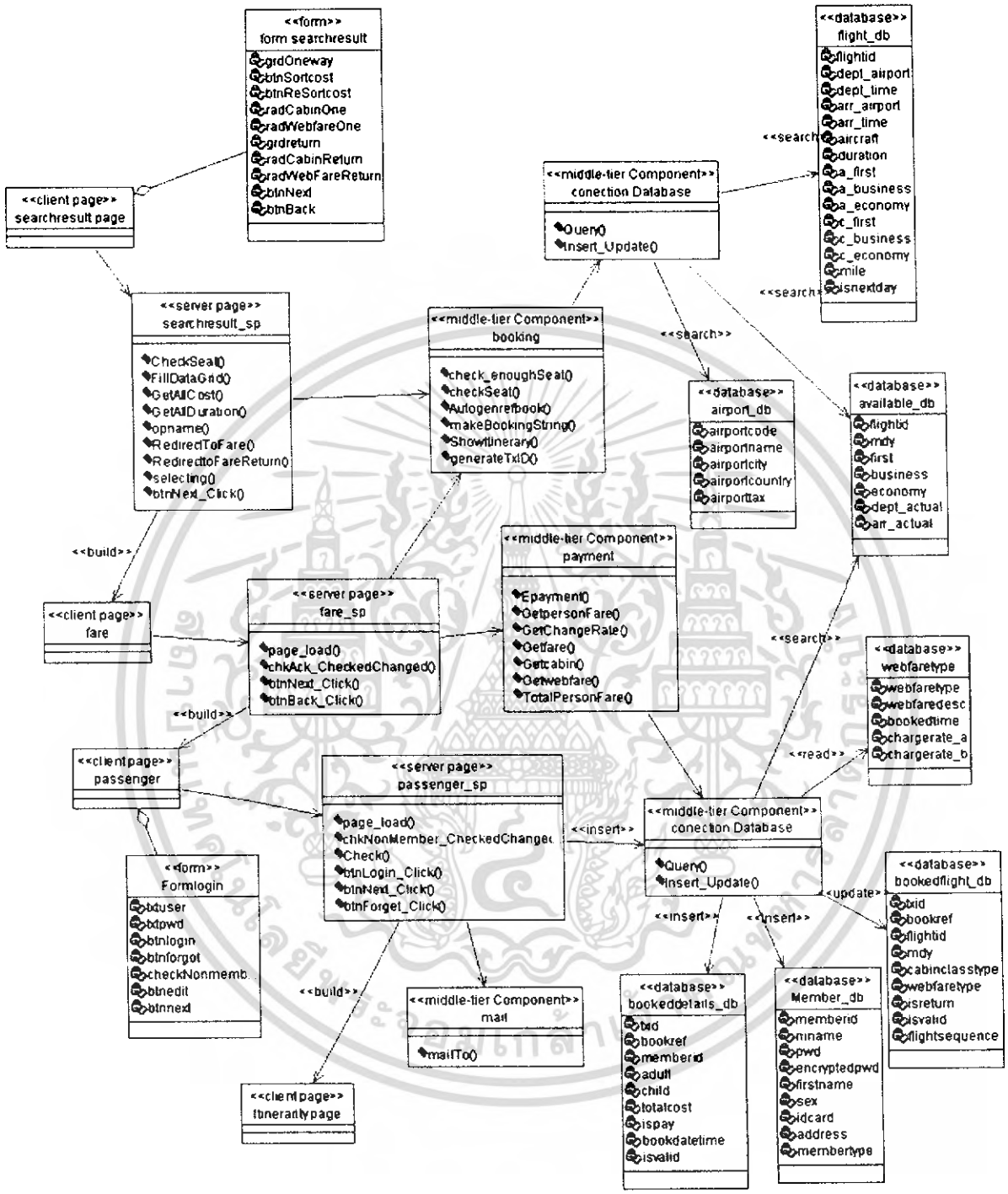
3.3.1.2 ขั้นตอนการค้นหาเที่ยวบิน



รูปที่ 3.18. แผนภาพคลาสแสดงการค้นหาเที่ยวบิน

จากรูปที่ 3.18 นั้น สามารถอธิบายได้ดังนี้ คือ ในคลาสไคลเอ็นต์เพจ search page จะประกอบไปด้วยฟอร์ม Form search และมีหน้าเซิร์ฟเวอร์เพจ search_sp ซึ่งจะทำการสร้างหน้าคลาสไคลเอ็นต์เพจ searchresult page ซึ่งจะประกอบไปด้วยฟอร์ม Form searchresult และมีหน้าเซิร์ฟเวอร์เพจ searchresult_sp ซึ่งจะทำการค้นหาเที่ยวบิน, ตรวจสอบจำนวนที่นั่ง,จัดรูปแบบการแสดงผล และการตรวจสอบราคาในเที่ยวบินต่างๆ โดยจะมีการเรียกใช้ฐานข้อมูล flight_db และ available_db โดยเรียกใช้คอมโพแนนท์เทียร์กลางชื่อ Connection Database และ Booking

3.3.1.3. ขั้นตอนของการทำการจองเที่ยวบิน



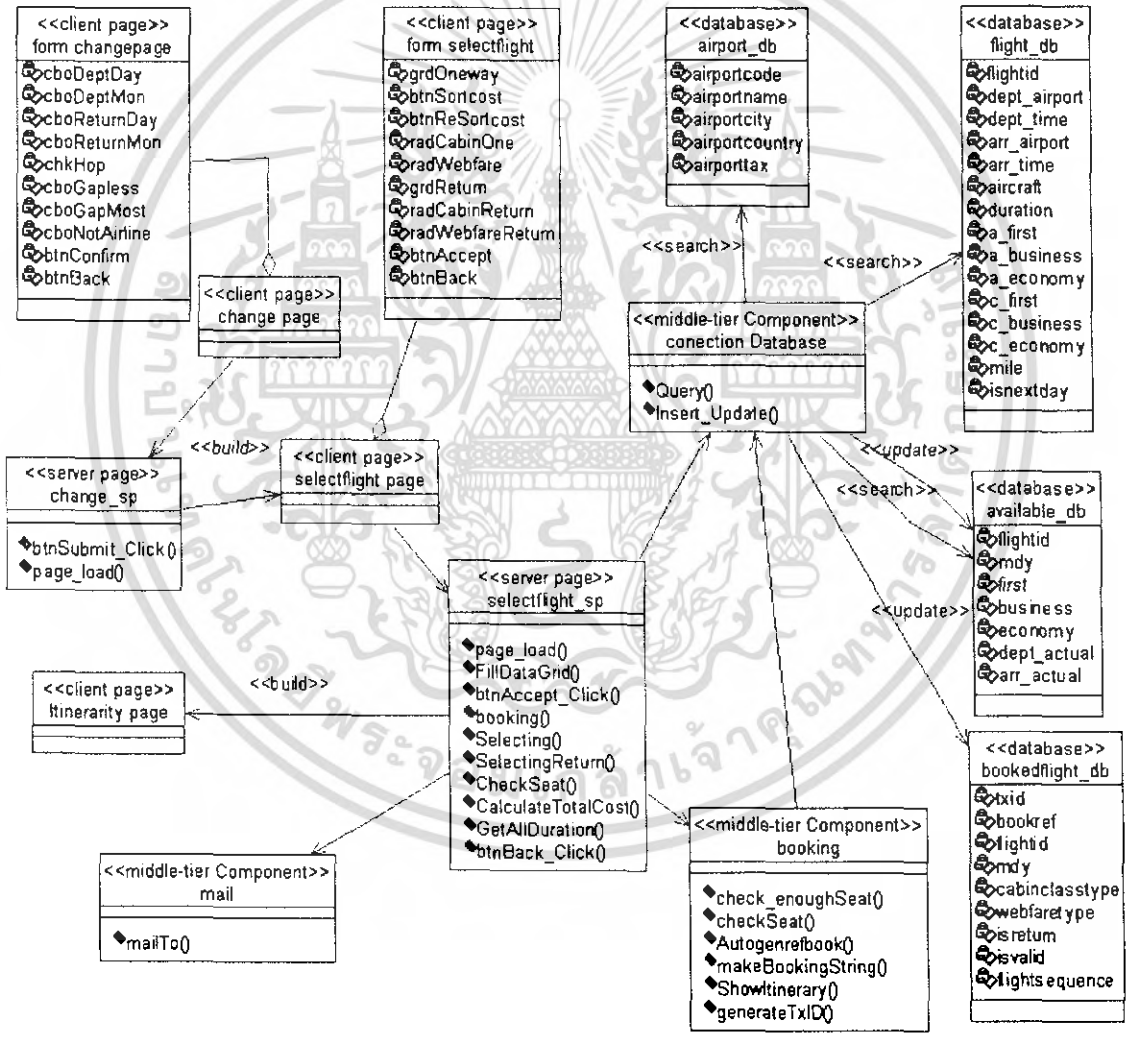
รูปที่ 3.19. แผนภาพแสดงคลาสของการทำการจองตั๋วเครื่องบิน

จากรูปที่ 3.19 นั้น สามารถอธิบายได้ดังนี้ คือ ในคลาสไคลเอ็นต์เพจ searchresult page จะประกอบไปด้วยฟอร์ม Form searchresult และมีหน้าเซิร์ฟเวอร์เพจ searchresult_sp ซึ่งจะทำการเรียกใช้งานคอมโพแนนท์เทียร์กลางของ booking และ connection database เพื่อทำการค้นหา

ในตัวฐานข้อมูลของ flight_db ,available_db, airport_db ในการที่จะทำการค้นหาเที่ยวบิน, ตรวจสอบจำนวนที่นั่ง,จัดรูปแบบการแสดงผล และการตรวจสอบราคาในเที่ยวบินต่างๆ

นอกจากนี้แล้วยังมีสร้างหน้าคลาสไคลเอ็นต์เพจ fare page ซึ่งมีหน้าเซิร์ฟเวอร์เพจ fare_sp ที่มีการเรียกใช้คอมโพแนนท์เทียร์กลางของ booking และ payment แล้วจะทำการสร้างคลาสไคลเอ็นต์เพจ passenger ขึ้นซึ่งจะประกอบไปด้วยฟอร์ม Formlogin โดยมีเซิร์ฟเวอร์เพจเป็น passenger_sp ที่จะทำการเพิ่มข้อมูลเมื่อมีการจองผ่านทางคอมโพแนนท์เทียร์กลางที่ชื่อ connection database และ mail

3.3.1.4. ขั้นตอนของการเปลี่ยนแปลงเที่ยวบิน



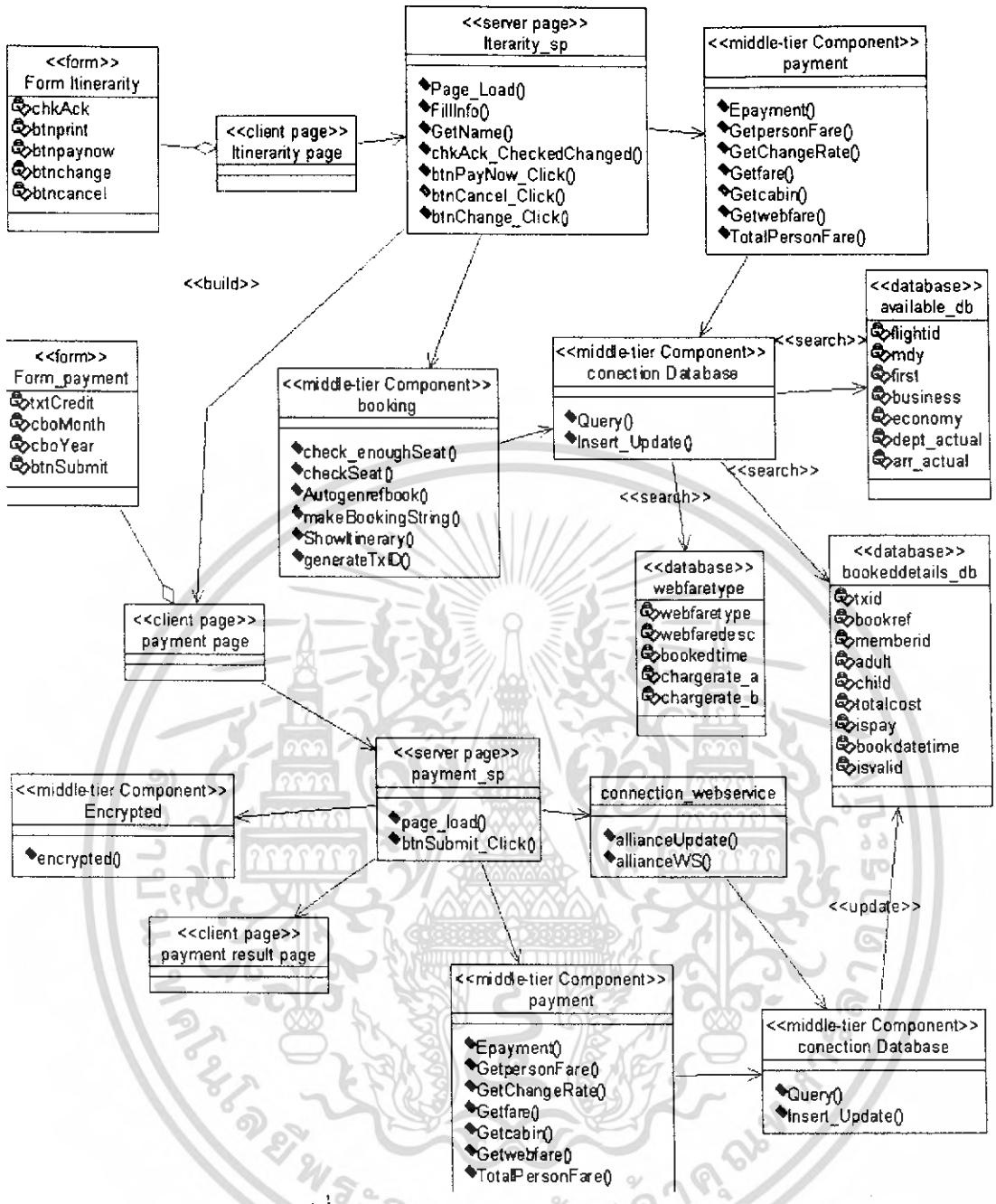
รูปที่ 3.20. แผนภาพคลาสแสดงการเปลี่ยนแปลงเที่ยวบิน

จากรูปที่ 3.20 นั้นสามารถอธิบายได้ดังนี้ คือ ในคลาสไคลเอ็นต์เพจ change page ซึ่งประกอบไปด้วยฟอร์ม form changepage จะทำการไปเรียกคลาสเซิร์ฟเวอร์เพจ change_sp ซึ่งจะมีการสร้างหน้าคลาสไคลเอ็นต์เพจ selectflight page ซึ่งประกอบไปด้วยฟอร์ม form selectflight

และมีการเรียกใช้เซิร์ฟเวอร์เพจ selectflight_sp ซึ่งจะทำการจัดผลของการแสดงหน้าผลลัพธ์ ทำการค้นหาเที่ยวบิน ตรวจสอบที่นั่งของเที่ยวบิน การคำนวณราคาของเที่ยวบินทั้งหมด โดยมีการเรียกใช้งานคอมโพแนนท์เทียร์กลางของ booking เพื่อทำการจองตั๋วโดยสาร, connection database เพื่อทำการค้นหาและอัพเดทค่าใหม่ให้กับฐานข้อมูลflight_db, available_db, bookedflight_db, airport_db และ mail เพื่อใช้ในการส่งเมลล์ให้ไปบอกกับผู้ใช้งานบนระบบ

3.3.1.5. ขั้นตอนของการชำระเงิน

จากรูปที่ 3.21 นั้นสามารถอธิบายได้ดังนี้ คือ ในคลาสไคลเอ็นต์เพจ itinerarity page ซึ่งประกอบไปด้วยฟอร์ม form itinerarity จะมีการเรียกใช้เซิร์ฟเวอร์เพจ Iterarity_sp ซึ่งจะมีการไปเรียกใช้คอมโพแนนท์เทียร์กลางของ payment และ booking ซึ่งจะมีการทำการตรวจสอบราคา ทำการสร้างหมายเลขตั๋วเครื่องบิน ทำการชำระเงินด้วยการเรียกใช้บริการเว็บเซอร์วิส การคำนวณค่าตั๋วโดยสาร และจะทำการสร้าง คลาสไคลเอ็นต์เพจ payment page ซึ่งประกอบไปด้วยฟอร์ม form payment และมีการเรียกใช้เซิร์ฟเวอร์เพจของ payment_sp ซึ่งจะมีการไปเรียกใช้คอมโพแนนท์เทียร์กลางของ connection_webservice , paymentและconnection database เพื่อใช้ในการทำการตรวจสอบราคาของตั๋วโดยสาร,การชำระเงิน,และอัพเดทฐานข้อมูลหลังจากการชำระเงินเสร็จแล้ว



รูปที่ 3.21. แผนภาพคลาสแสดงการชำระเงิน

3.3.2. การออกแบบเว็บเซอร์วิส

โดยระบบการออกแบบเว็บเซอร์วิสที่ได้ทำการออกแบบ ดังรูปที่ 3.22. ไว้มีดังนี้ คือ

SkyHigh Web service ซึ่งประกอบไปด้วย

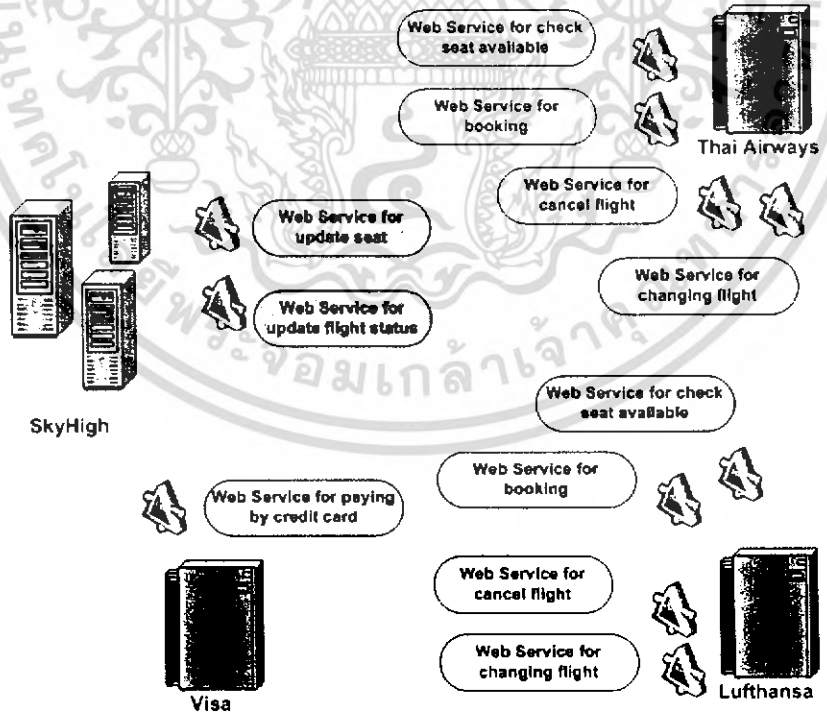
- เว็บเซอร์วิสสำหรับการอัพเดทข้อมูลที่นั่งว่าง
- เว็บเซอร์วิสสำหรับการอัพเดทข้อมูลเวลาการบินเข้า-ออกจริงจากสนามบินของสายการบินพันธมิตร

Alliance Web service ซึ่งประกอบไปด้วย

- เว็บเซอร์วิสสำหรับการตรวจสอบข้อมูลที่นั่งว่าง
- เว็บเซอร์วิสสำหรับการจองตั๋วสายการบิน
- เว็บเซอร์วิสสำหรับยกเลิกการจองตั๋วสายการบิน
- เว็บเซอร์วิสสำหรับการเปลี่ยนวันเวลาและสายการบินจากการจองตั๋วสายการบิน

Credit Card Owner Web service ซึ่งประกอบไปด้วย

- เว็บเซอร์วิสสำหรับการชำระเงินผ่านบัตรเครดิต

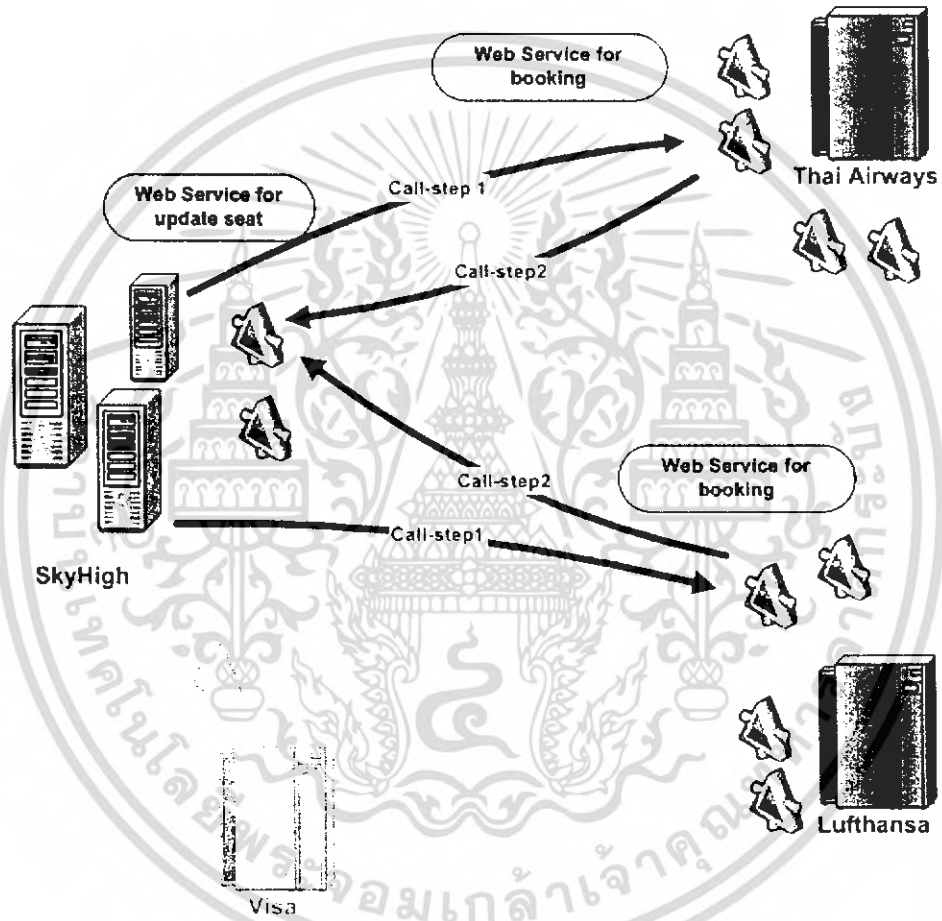


รูปที่ 3.22. แสดงการให้บริการเว็บเซอร์วิสในการให้บริการบนเว็บจองตั๋วหลายสายการบิน

การทำงานของเว็บเซอร์วิสในกรณีต่างๆในระบบนั้นมีดังต่อไปนี้คือ

3.3.2.1. กรณีการจองตั๋วสายการบิน

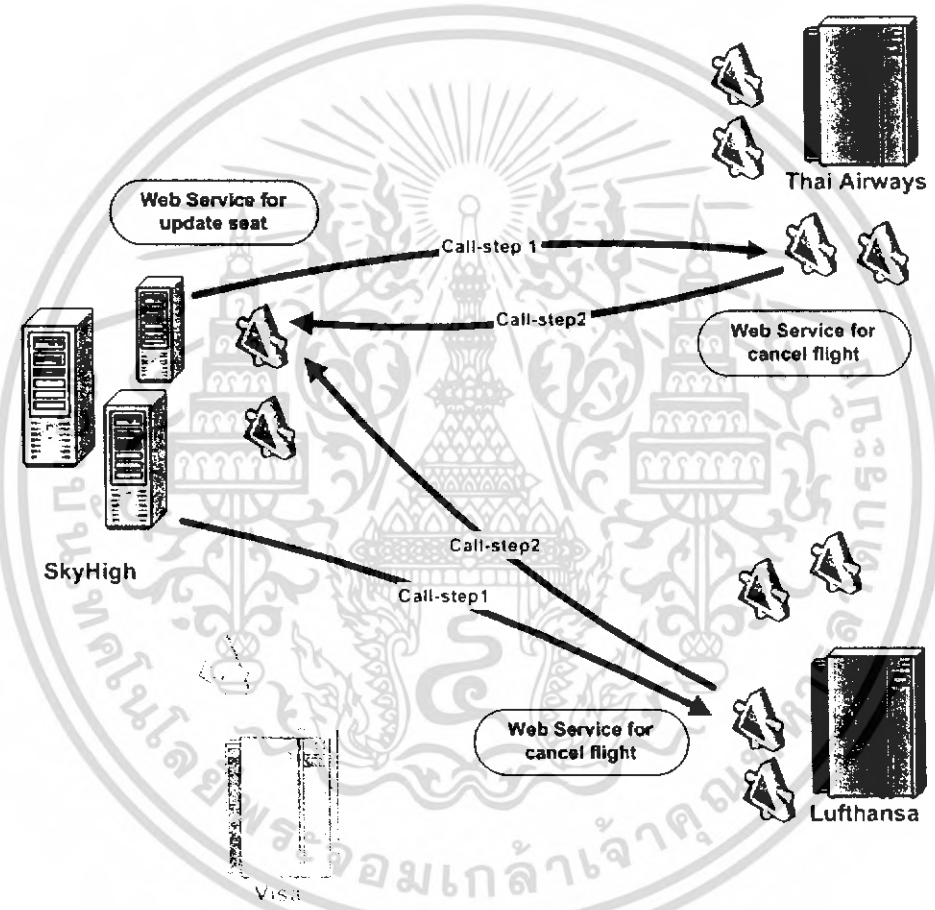
เมื่อผู้ใช้สั่งจองตั๋วเครื่องบิน ระบบจะทำการไปเรียกเว็บเซอร์วิสการจองตั๋วของของสายการบินที่เกี่ยวข้อง โดยเมื่อเว็บเซอร์วิสทำการจองเรียบร้อยแล้ว จะมีการคำนวณจำนวนที่นั่งว่างของสายการบินนั้นๆ แล้วส่งมาอัพเดทที่ฝั่ง SkyHigh ด้วย ดังรูป 3.23.



รูปที่ 3.23. แสดงการให้บริการเว็บเซอร์วิสในกรณีการจองตั๋วสายการบิน

3.3.2.2. กรณีการยกเลิกการจองตั๋วสายการบิน

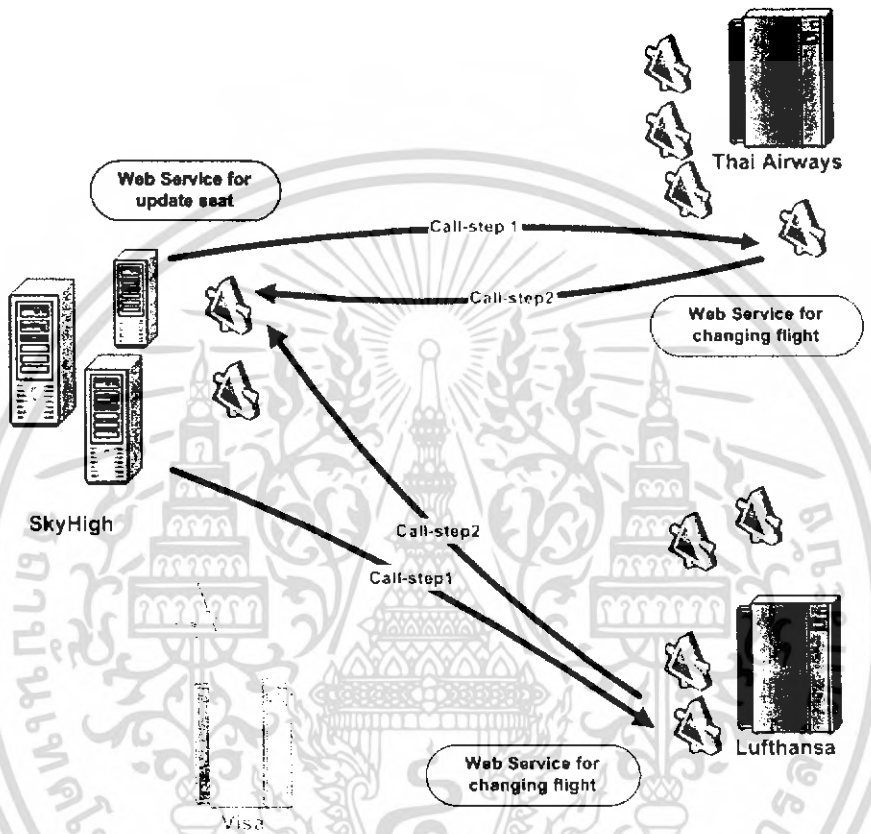
เมื่อผู้ใช้สั่งยกเลิกการจองตั๋วเครื่องบิน ระบบจะทำการไปเรียกเว็บเซอร์วิสของการยกเลิกการจองตั๋วของของสายการบินที่เกี่ยวข้อง โดยเมื่อเว็บเซอร์วิสทำการยกเลิกการจองเรียบร้อยแล้วจะมีการคำนวณจำนวนที่นั่งว่างของสายการบินนั้นๆ แล้วส่งมาอัพเดทที่ฝั่ง SkyHigh ด้วย ดังรูป 3.24.



รูปที่ 3.24. แสดงการให้บริการเว็บเซอร์วิสในกรณีการยกเลิกการจองตั๋วสายการบิน

3.3.2.3. กรณีการเปลี่ยนวันเวลาและสายการบินจากการจองตั๋วสายการบิน

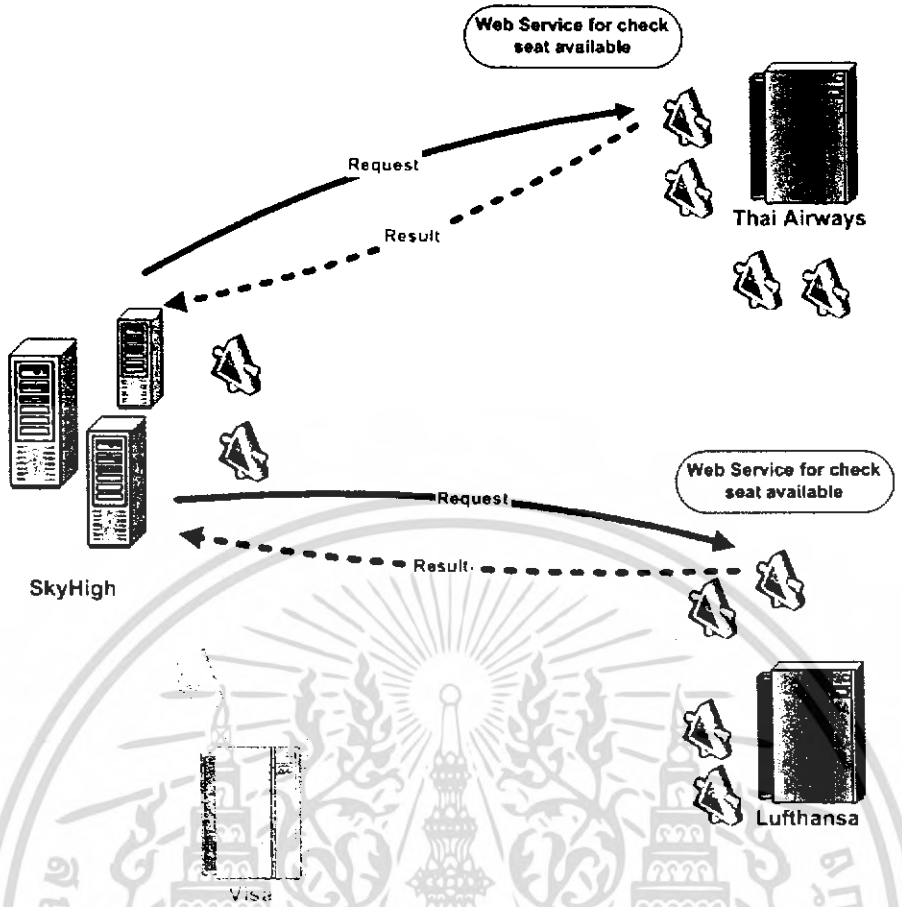
เมื่อผู้ใช้สั่งเปลี่ยนวันเวลาและสายการบินจากการจองตั๋วสายการบิน ระบบจะทำการไปเรียกเว็บเซอร์วิสของการเปลี่ยนวันเวลาและสายการบินจากการจองตั๋วสายการบินที่เกี่ยวข้อง โดยเมื่อเว็บเซอร์วิสทำเปลี่ยนการจองเรียบร้อยแล้ว จะมีการคำนวณจำนวนที่นั่งว่างของสายการบินนั้นๆ แล้วส่งมาอัพเดทที่ฝั่ง SkyHigh ด้วย ดังรูปที่ 3.25.



รูปที่ 3.25. แสดงการให้บริการเว็บเซอร์วิสในกรณีการเปลี่ยนวันเวลาและสายการบินจากการจองตั๋วสายการบิน

3.3.2.4. กรณีการตรวจสอบข้อมูลที่นั่งว่าง

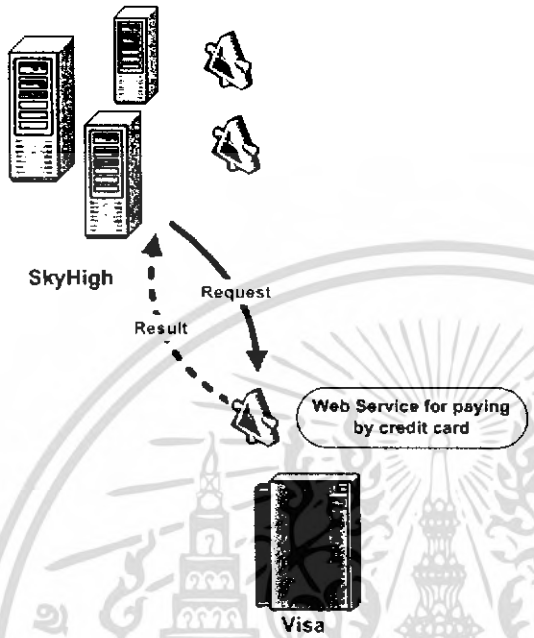
เมื่อผู้ใช้ทำการเลือกสายการบิน ระบบจะทำการไปเรียกเว็บเซอร์วิสของการตรวจสอบข้อมูลที่นั่งว่างจากสายการบินที่เกี่ยวข้อง แล้วส่งผลลัพธ์กลับมายังฝั่ง SkyHigh ดังรูปที่ 3.26.



รูปที่ 3.26. แสดงการให้บริการเว็บเซอร์วิสในกรณีการตรวจสอบข้อมูลที่นั่งว่าง

3.3.2.5. กรณีการชำระเงินผ่านบัตรเครดิต

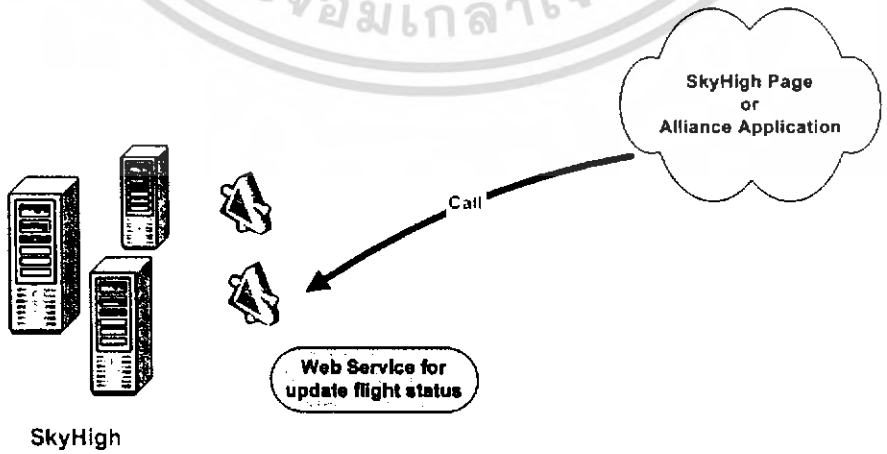
เมื่อผู้ใช้ทำการชำระเงินค่าตั๋วเครื่องบิน ระบบจะทำการเรียกเว็บเซอร์วิสของการชำระเงินผ่านบัตรเครดิตจากผู้ให้บริการการชำระเงินผ่านบัตรเครดิต แล้วส่งผลลัพธ์กลับมายังฝั่ง SkyHigh ว่าการชำระเงินเสร็จสมบูรณ์หรือไม่ ดังรูปที่ 3.27.



รูปที่ 3.27. แสดงการให้บริการเว็บเซอร์วิสในกรณีการชำระเงินผ่านบัตรเครดิต

3.3.2.6. กรณีการอัพเดทข้อมูลเวลาการบินเข้า-ออกจริงจากสนามบินของสายการบินพันธมิตร

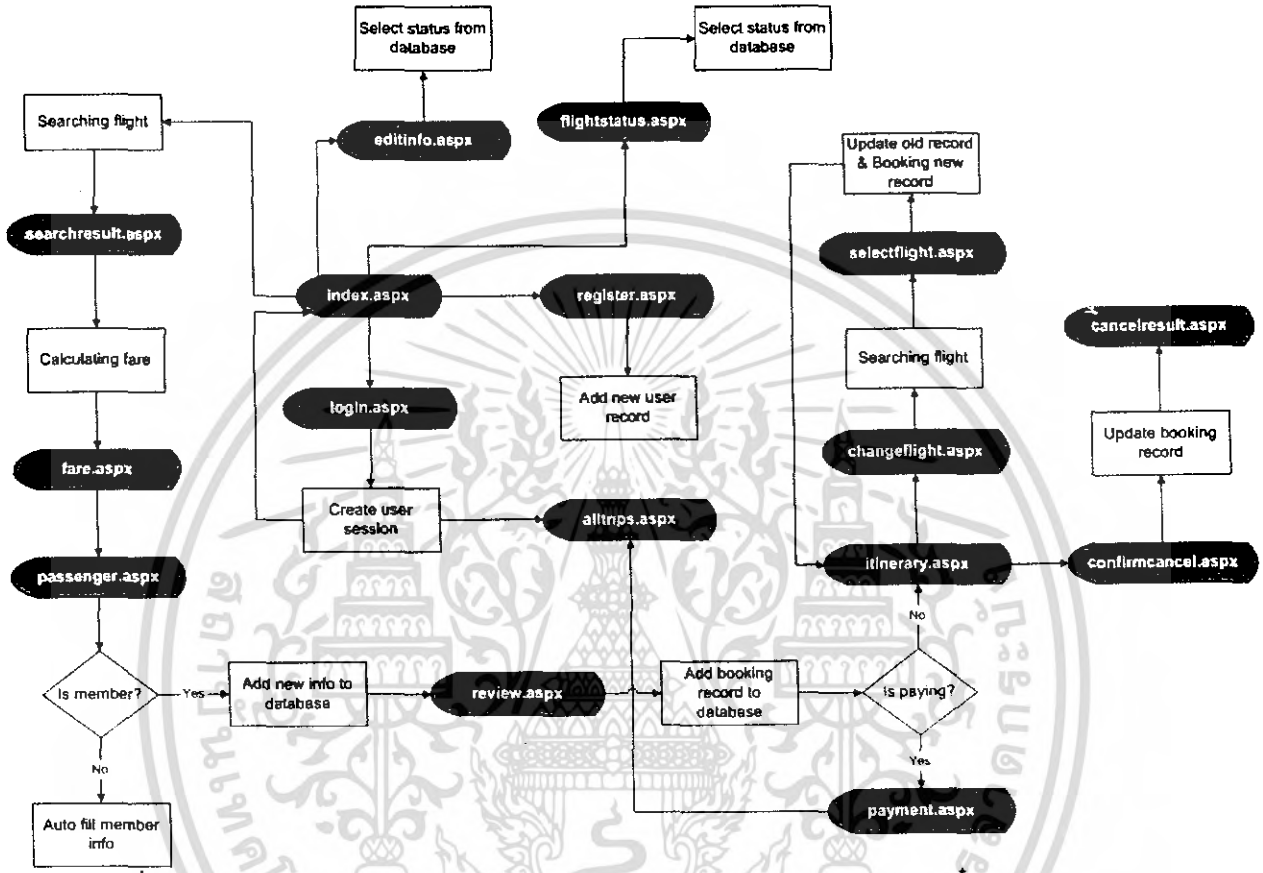
เมื่อมีการบินเข้า-ออกจากสนามบินของสายการบินพันธมิตร จะมีการอัพเดทข้อมูลผ่านเว็บเซอร์วิสของการอัพเดทข้อมูลเวลาการบินเข้า-ออกจริงจากสนามบินของสายการบินพันธมิตร โดยจะถูกเรียกใช้จากแอพพลิเคชันของสายการบินพันธมิตรก็ได้ หรือจะทำการอัพเดทผ่านทางหน้าเว็บเพจที่สร้างเตรียมไว้ก็ได้ ดังรูปที่ 3.28.



รูปที่ 3.28. แสดงการให้บริการเว็บเซอร์วิสในกรณีการอัพเดทข้อมูลเวลาการบินเข้า-ออกจริงจากสนามบินของสายการบินพันธมิตร

3.3.3. การไหลของระบบหน้าเว็บเพจ

ในระบบของเว็บแอพพลิเคชันของการจองตั๋วเครื่องบินผ่านหลายสายการบินนี้มีการไหลเวียนของการทำงานในส่วนหน้าเพจต่างๆดังรูปที่ 2.29. ดังนี้คือ



รูปที่ 3.29. แสดงการไหลเวียนของหน้าเพจของเว็บแอพพลิเคชันของการจองตั๋วหลายสายการบิน

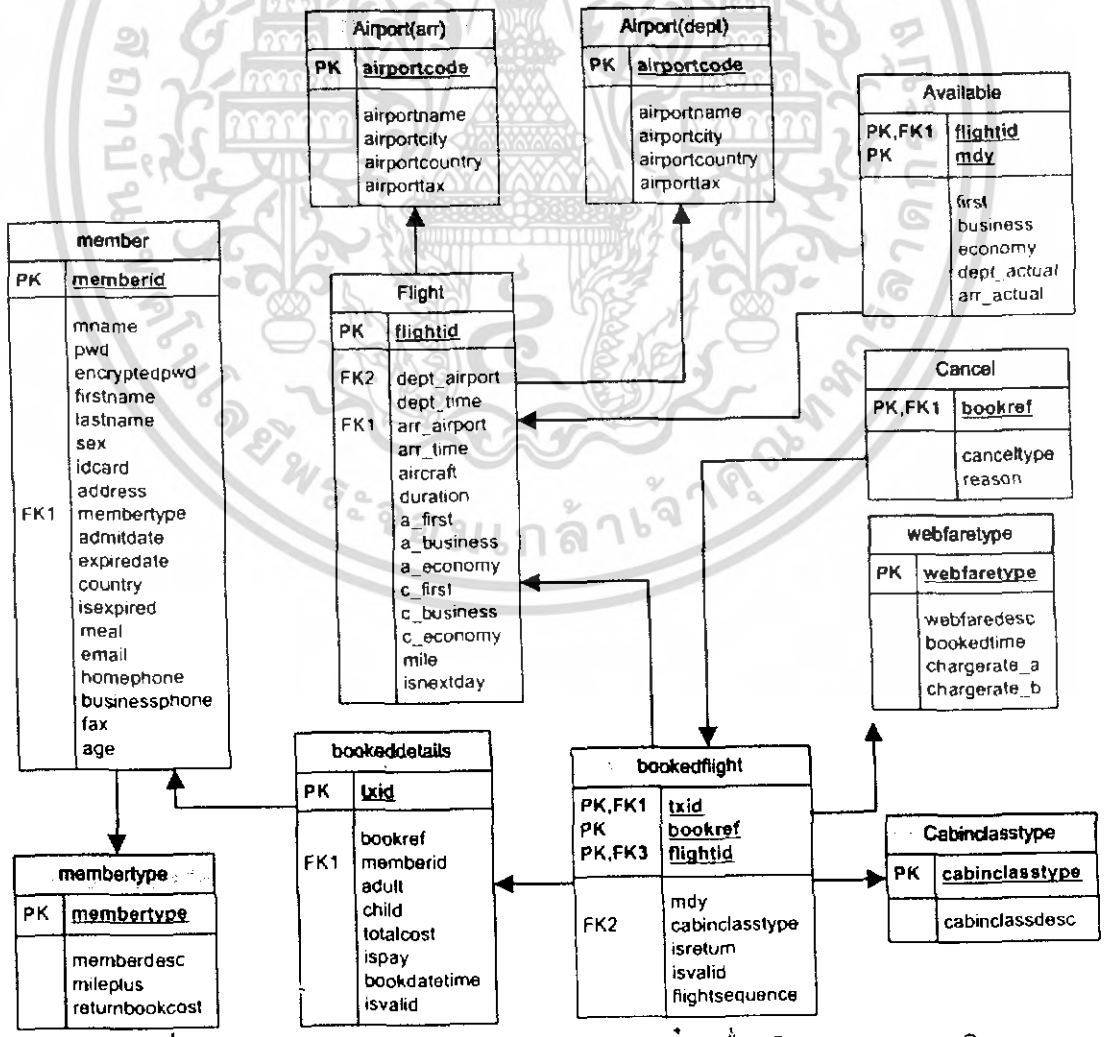
3.3.4. การออกแบบฐานข้อมูลของระบบนำส่งเอกสาร

ในER diagram ในรูปที่ 3.30. นี้ จะแสดงความสัมพันธ์ภายในฐานข้อมูลโดยเป็นความสัมพันธ์ของตารางต่างๆ ซึ่งตารางในฐานข้อมูลก็มีดังต่อไปนี้ คือ

ตารางที่ 3.2. อธิบายการทำงานแต่ละตารางในฐานข้อมูล

ชื่อตาราง	คำอธิบาย
Airport	เป็นตารางที่เก็บรหัส,ชื่อ ของสนามบินรวมทั้งชื่อเมืองและประเทศที่ตั้งของสนามบินนั้น
Available	เป็นตารางที่เก็บจำนวนที่นั่งของเที่ยวบินในวันใดวันหนึ่งและจะเก็บที่นั่งในหลายระดับ
Flight	เป็นตารางแสดงรายละเอียดของเที่ยวบินว่าออกและเข้าที่เมืองไหน

	สนามบินไหน บอกระยะทางและวลาของการเดินทาง
Cancel	เป็นตารางที่เก็บจำนวนหมายเลขตั๋วที่ทำการยกเลิกรวมทั้งเหตุผลที่ทำการยกเลิก เพื่อนำเอามาวิเคราะห์ในภายหลังได้
Webfaretype	เป็นตารางเก็บรูปแบบของการจองตั๋วซึ่งก็จะมีค่าบริการที่ไม่เท่ากัน
Member	เป็นตารางเก็บรายละเอียดต่างๆของสมาชิก เช่น ชื่อสมาชิก รหัสผ่าน ที่อยู่ อีเมลล์ วันหมดอายุ เป็นต้น
Membertype	เป็นตารางเก็บประเภทของสมาชิก
Bookeddetails	เป็นตารางเก็บรายละเอียดของหมายเลขตั๋วเครื่องบินเช่น ว่ามีจำนวนที่นั่งเท่าไหร่เป็นของสมาชิกคนใด และได้ทำการชำระเงินหรือยัง เป็นต้น
Bookedflight	เป็นตารางเก็บรายละเอียดของหมายเลขตั๋วบินที่เกี่ยวข้องกับเที่ยวบินที่ต้องขึ้นในตั๋วใบนึง เช่น เที่ยวบินที่จองไว้ วันที่
Cabinclasstype	เป็นตารางเก็บรายละเอียดของประเภทของระดับของตั๋วของที่นั่งบนเครื่องบิน



รูปที่ 3.30 แสดง ER Diagram ของระบบการจองตั๋วเครื่องบินหลายสายการบิน

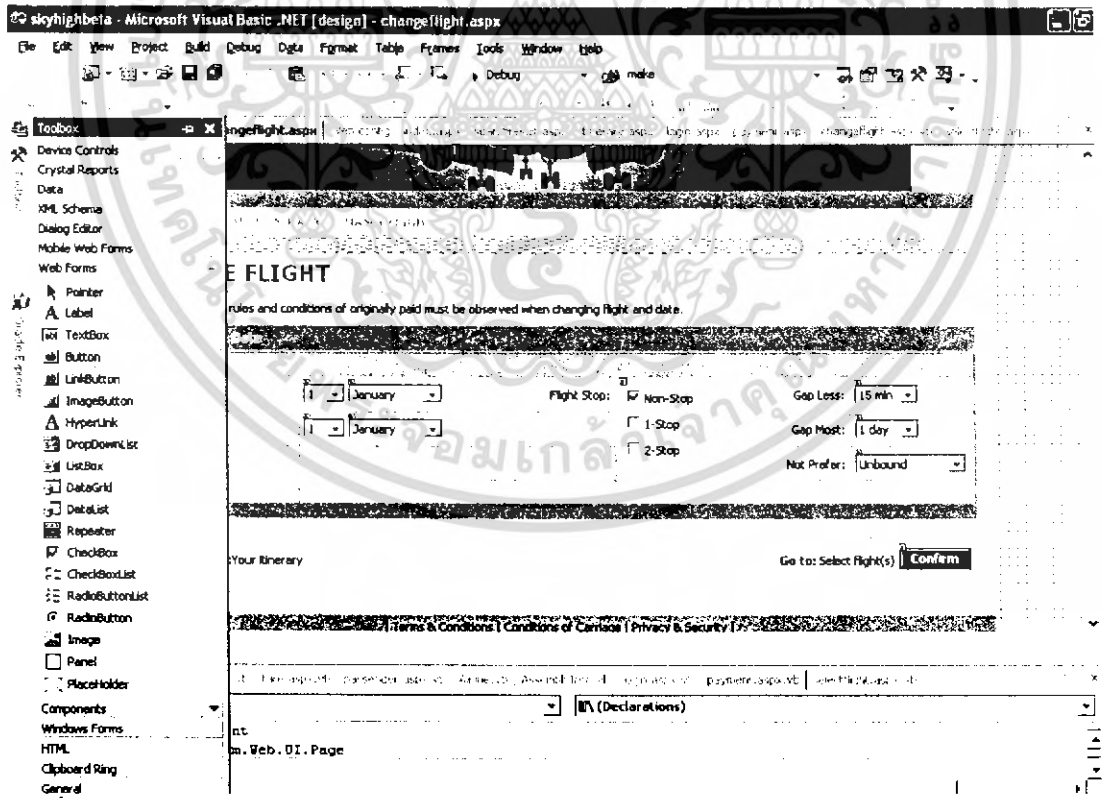
3.4. ขั้นตอนการสร้าง (Implement)

ขั้นตอนของการสร้างเว็บแอพพลิเคชันนี้ เราจะทำการสร้างโดยอิงจากแผนภาพคลาส และ ER diagram ที่ได้ทำมาในขั้นตอนของการออกแบบ โดยจะต้องมีการสร้างในหลายๆส่วน ด้วยกันไม่ว่าจะเป็นในส่วนของ ไคลเอ็นต์เพจ เซิร์ฟเวอร์เพจ และคอมโพแนนท์ที่เทียร์กลาง

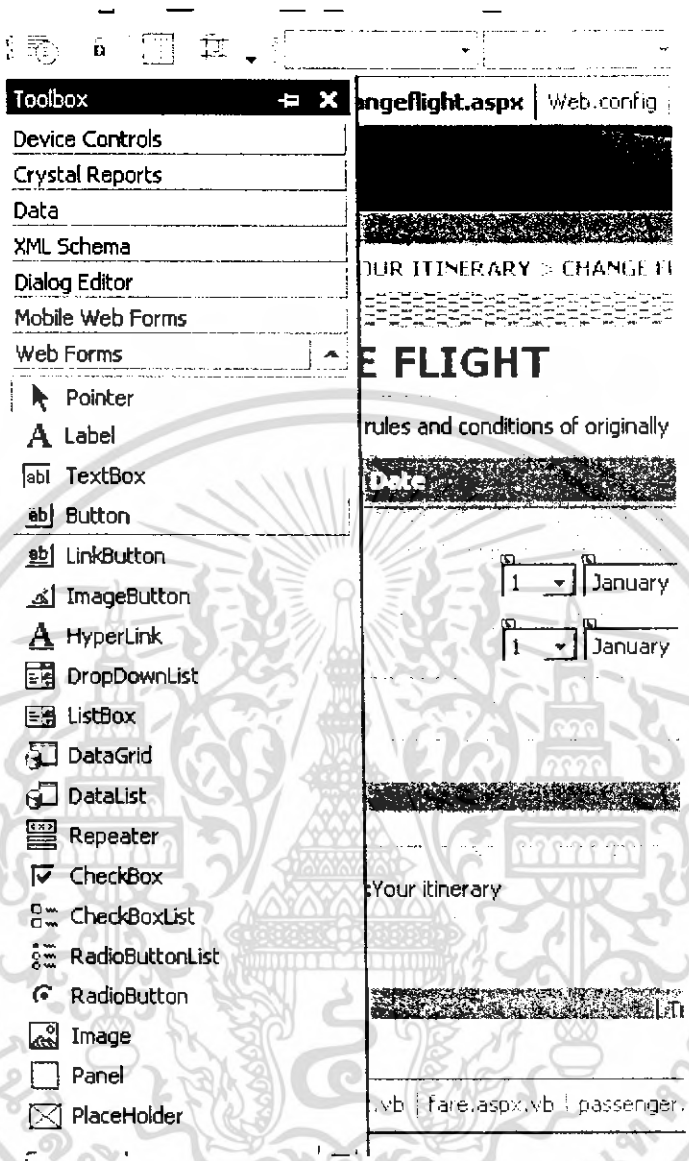
3.4.1. การสร้างในส่วนของไคลเอ็นต์เพจ

ไคลเอ็นต์เพจนี้ส่วนใหญ่จะต้องประกอบไปด้วยฟอร์มต่างๆ เช่น ปุ่ม ฟิลด์รับข้อความ ตาต้ากริดที่ใช้แสดงข้อมูล เป็นต้น ซึ่งการสร้างไคลเอ็นต์เพจที่ประกอบไปด้วยฟอร์มนี้เราจะมีการเรียกใช้ทูลวิชวลสตูดิโอดอทเน็ต (Visual Studio .NET) มาช่วยในการสร้างไคลเอ็นต์เพจและฟอร์ม

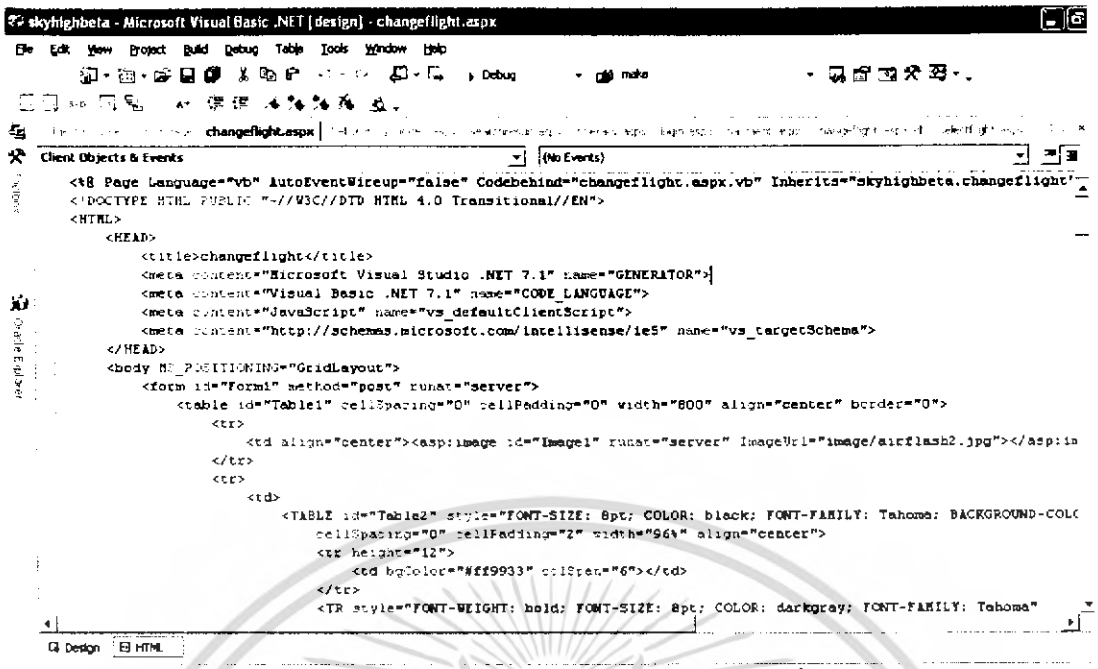
เนื่องจาก ทูลวิชวลสตูดิโอดอทเน็ต (Visual Studio .NET) จะมีฟอร์มที่เรียกใช้สำเร็จรูปมากมายซึ่งสามารถดึงมาใช้ได้บนหน้าไคลเอ็นต์เพจ แล้วทูลวิชวลสตูดิโอดอทเน็ต (Visual Studio .NET)จะทำการสร้างโค้ดของการสร้างฟอร์มให้โดยอัตโนมัติ ดังรูปที่ 3.31 – 3.34.



รูปที่ 3.31. แสดงการสร้างฟอร์มบนไคลเอ็นต์เพจในวิชวลสตูดิโอดอทเน็ต ในส่วนหน้าเพจของการค้นหา



รูปที่ 3.32. ฟอร์มสำเร็จรูปที่สามารถให้เรียกใช้บนไคลเอ็นต์เพจได้



รูปที่ 3.33. แสดงโค้ดที่วิชวลสตูดิโอดอทเน็ตทำการสร้างให้ในการสร้างฟอร์มต่างๆ ในส่วนหน้าเพจของการค้นหา

รูปที่ 3.34 แสดงการตัวอย่างหน้าไคลเอ็นต์เพจของการชำระเงิน

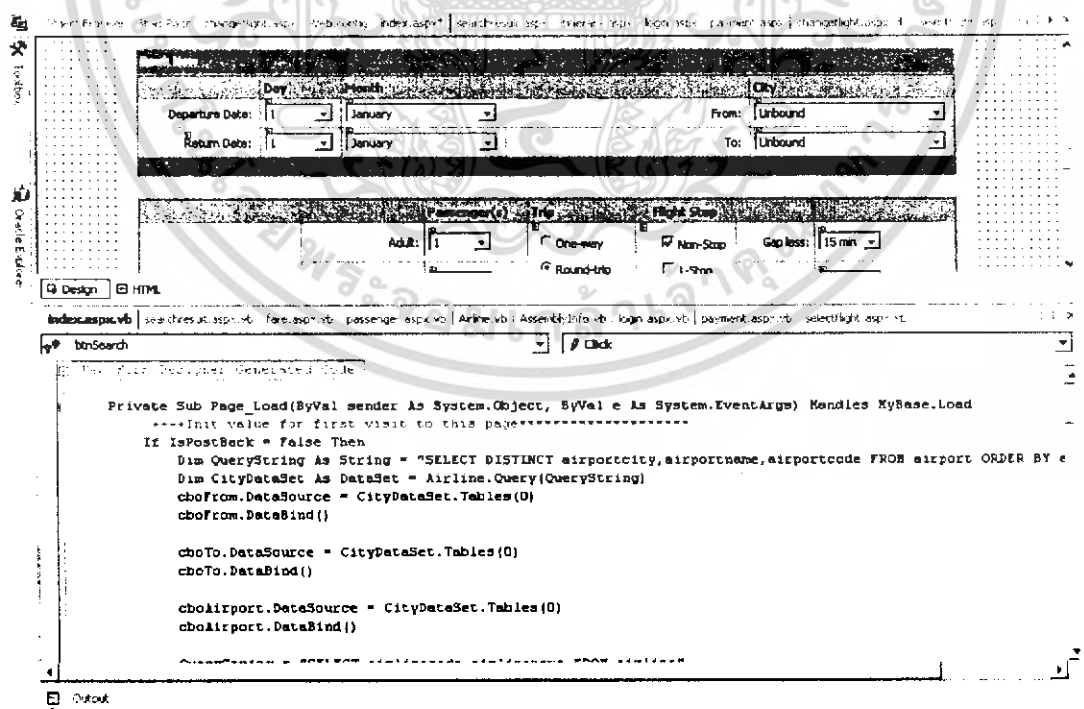
ในส่วนของไคลเอ็นต์เพจนี้จะมีรูปแบบไฟล์ประเภท ".aspx" ซึ่งสามารถใช้บราวเซอร์ทั่วไป ในการอ่านได้ โดยผ่านโปรโตคอลเอชทีทีพี (HTTP)

3.4.2. การสร้างในส่วนของเซิร์ฟเวอร์เพจ

เซิร์ฟเวอร์เพจนี้จะเป็นเพจที่มีการประมวลและทำงานอยู่บนฝั่งของเซิร์ฟเวอร์ซึ่งในวิชวลสตูดิโอดอทเน็ตนี้จะทำการสร้างเซิร์ฟเวอร์เพจได้จากไคลเอ็นต์เพจที่ทำการสร้างมาก่อนหน้านี้ ในรูปแบบของไฟล์จะเป็นแบบ ".aspx.vb" ในเซิร์ฟเวอร์เพจนี้อาจจะประกอบไปด้วยเมธอดที่เกิดการกระทำต่างๆบนฟอร์มซึ่งเมธอดบางเมธอดนั้นจะมีการไปเรียกเมธอดหรือคลาสในส่วนของคอมโพแนนท์เทียร์กลางมาใช้ก็ได้ ดังรูปที่ 3.35 และ 3.36

```
index.aspx.vb | ...
(Page Events)                                   Load
Public Class index
    Inherits System.Web.UI.Page

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        '+++Init value for first visit to this page+++++++++++++++++++++
        If IsPostBack = False Then
            Dim QueryString As String = "SELECT DISTINCT airportcity,airportname,airportcode FROM airport ORDER BY
            Dim CityDataSet As DataSet = Airline.Query(QueryString)
            cboFrom.DataSource = CityDataSet.Tables(0)
            cboFrom.DataBind()

            cboTo.DataSource = CityDataSet.Tables(0)
            cboTo.DataBind()

            cboAirport.DataSource = CityDataSet.Tables(0)
            cboAirport.DataBind()

            QueryString = "SELECT airlinecode,airlinename FROM airline"
            Dim AirlineDataSet As DataSet = Airline.Query(QueryString)
            Dim NotSelect As DataRow = AirlineDataSet.Tables(0).NewRow()
            NotSelect.Item("airlinecode") = "null"
            NotSelect.Item("airlinename") = "=Not Select="
            AirlineDataSet.Tables(0).Rows.Add(NotSelect)
            Dim AirlineView As DataView
            AirlineView = AirlineDataSet.Tables(0).DefaultView
```

รูปที่ 3.35. แสดงส่วนโค้ดของเซิร์ฟเวอร์เพจของการค้นหาเที่ยวบิน



รูปที่ 3.36. แสดงความสัมพันธ์ระหว่างส่วนโค้ดและหน้าฟอร์มของเซิร์ฟเวอร์เพจของการค้นหาเที่ยวบิน

3.4.3. การสร้างในส่วนของคอมโพแนนท์ที่เทียร์กลาง

ส่วนของคอมโพแนนท์ที่เทียร์กลางนี้จะเป็นเหมือนคลาสที่เป็นคนละส่วนกับเซิร์ฟเวอร์เพจ โดยที่เซิร์ฟเวอร์เพจอาจจะเรียกใช้งานในส่วนของคอมโพแนนท์ที่เทียร์กลางนี้ได้ เช่น เซิร์ฟเวอร์เพจของการค้นหาจะไปเรียกคอมโพแนนท์เทียรก์กลางที่ทำหน้าที่ในการติดต่อกับระบบฐานข้อมูลเพื่อทำการค้าหาข้อมูลในฐานข้อมูล เป็นต้น และคลาสของคอมโพแนนท์เทียร์กลางนี้จะมีประเภทของไฟล์เป็น ".vb" ดังรูปที่ 3.37

```
Airline                                                    Query
Imports System.Text
Imports System.Random
Imports System.Web.Mail
Imports Oracle.DataAccess.Client
Imports System.Security.Cryptography

Public Class Airline
    Inherits System.Web.UI.Page

    Shared Function Query(ByVal QueryString As String) As DataSet
        'for any "SELECT" from alliance's tables ++++++++++++++
        Dim ConnectionString As String = "Data Source=(DESCRIPTION=(ADDRESS_LIST=" & _
                                         "(ADDRESS=(PROTOCOL=TCP)(HOST=DJTONG)(PORT=1521)))" & _
                                         "(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=AIRLINE)));" & _
                                         "User Id=alliance;Password=alliance;"

        Dim OraConnection As New OracleConnection(ConnectionString)
        Dim QueryAdapter As OracleDataAdapter = New OracleDataAdapter(QueryString, OraConnection)

        Dim QueryDataSet As New DataSet
        QueryAdapter.Fill(QueryDataSet)

        Return QueryDataSet

    End Function
```

รูปที่ 3.37. แสดงเมธอดของการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูลในส่วนของคอมโพแนนท์เทียร์กลาง

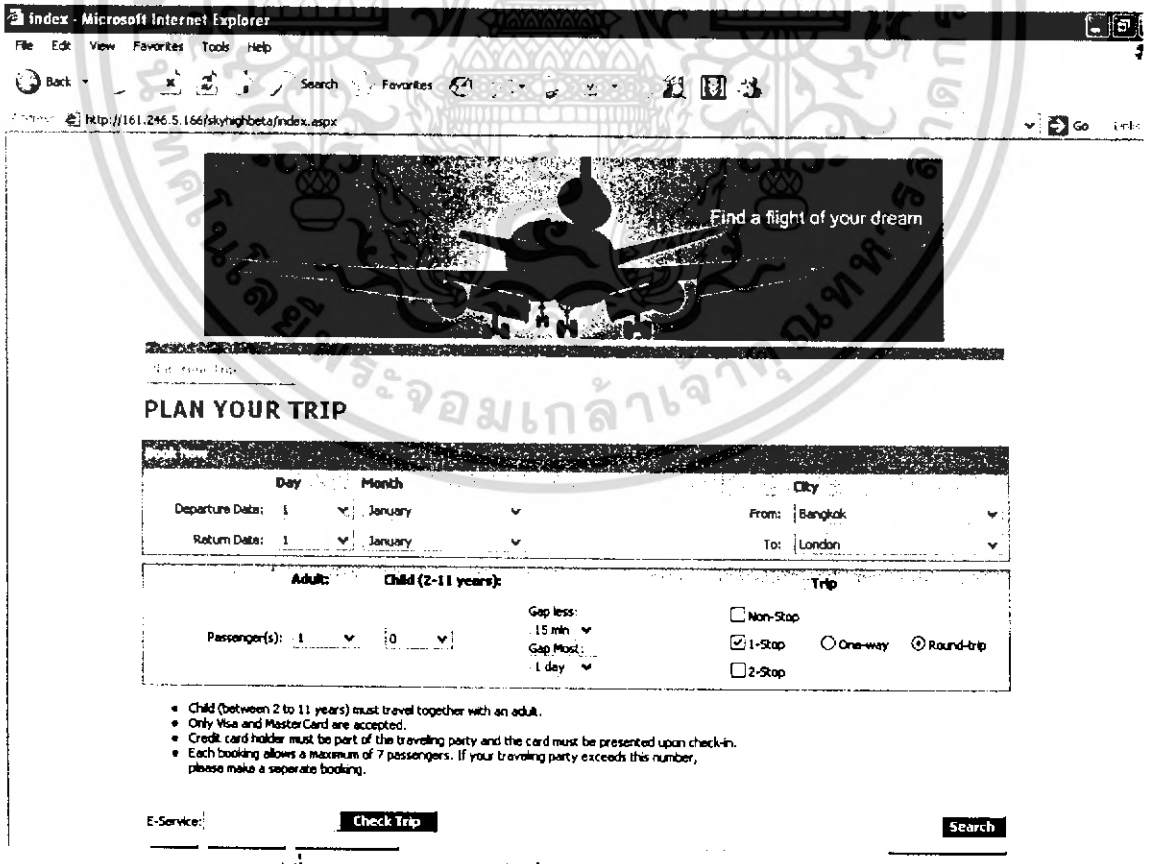
บทที่ 4
ผลของการทำงานส่วนต่างๆของเว็บแอพพลิเคชัน

ในการทดลองบนเว็บแอพพลิเคชันต้นแบบของการจองตั๋วเครื่องบินหลายสายการบินนี้ ประกอบด้วยส่วนที่สำคัญหลายส่วนด้วยกันคือ

4.1 ส่วนของการทำการจองตั๋วเครื่องบิน

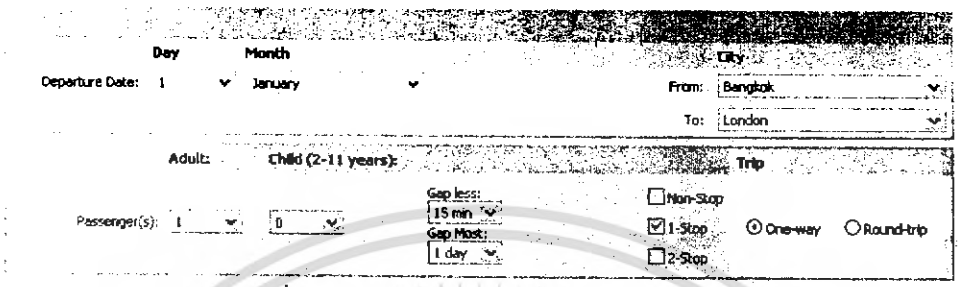
ส่วนหน้าแรกของการจองตั๋วนั้นจะมีคุณสมบัติให้ได้เลือกทำหลายคุณสมบัติด้วยกันคือ สามารถล็อกอินผ่านหน้าเว็บ ทำการเปลี่ยนแปลงและยกเลิกเที่ยวบินที่ทำการจองไปแล้ว โดยส่วนหลักของหน้าของขั้นตอนก่อนการทำการจอง คือ การค้นหาสายการบินที่ต้องการ

โดยผู้ใช้งานจะต้องกรอกชื่อเมืองที่เครื่องบินจะออกและเมืองที่กำลังจะไป เลือกได้ว่าจะไปอย่างเดียวหรือทั้งไปทั้งกลับ วันที่ทั้งขาไปและขากลับ จำนวนผู้โดยสารทั้งในส่วนของผู้ใหญ่และเด็ก,จำนวนครั้งของการต่อขึ้นเครื่องไปยังเมืองเป้าหมาย และช่วงเวลาที่ใช้ในการรอในการต่อเครื่องในแต่ละครั้ง (gap) ดังรูปที่ 4.1.

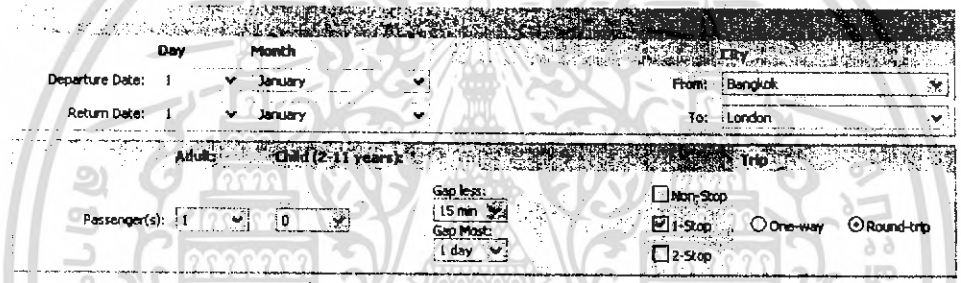


รูปที่ 4.1. ส่วนของหน้าเว็บก่อนทำการค้นหาเส้นทาง

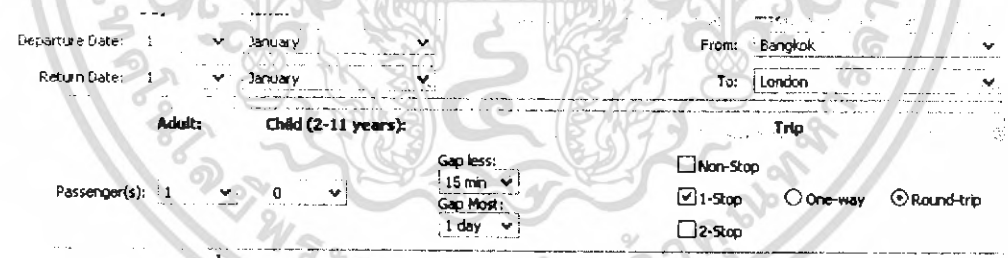
ในส่วนหน้าแรกก่อนการทำการค้นหานี้สามารถที่จะกำหนดได้ว่า จะไปแบบเที่ยวเดียว หรือไปทั้งขาไปและกลับ โดยถ้าเลือกไปอย่างเดียวนั้น ก็จะเห็นฟอร์มในส่วนของวันที่ขาไปเพียงฟอร์มเดียว ดังรูปที่ 4.2. – 4.5.



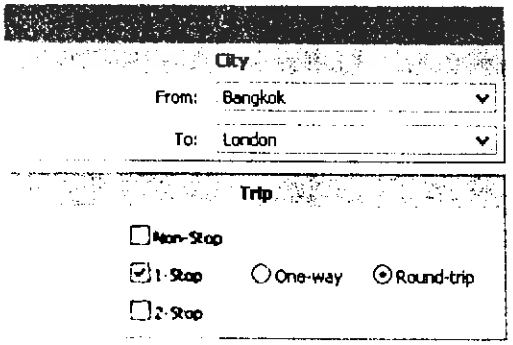
รูปที่4.2. ในกรณีเลือกขาไปเพียงอย่างเดียว



รูปที่4.3. ในกรณีเลือกขาไปและขากลับ

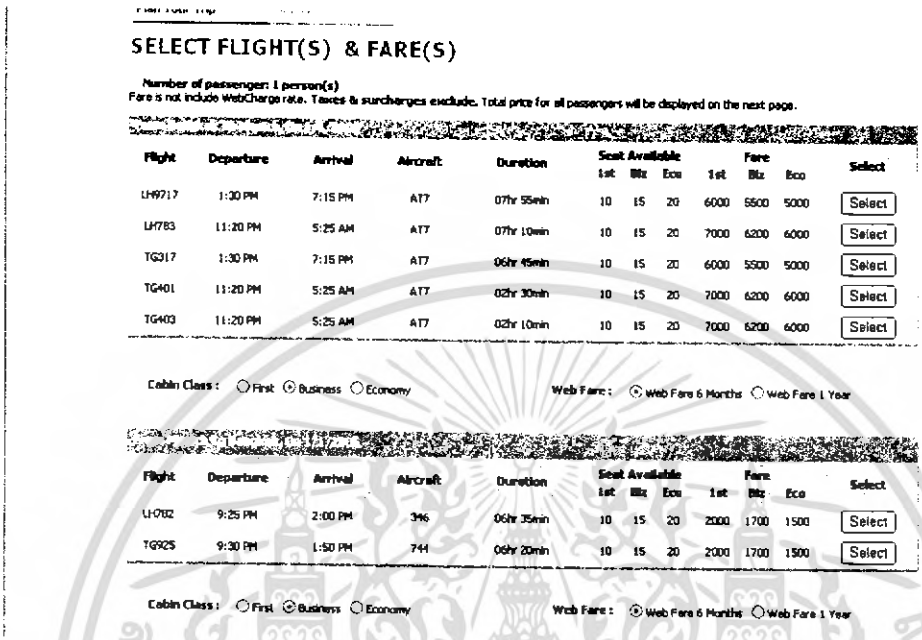


รูปที่ 4.4. ฟอร์มใส่รายละเอียดของเที่ยวบินที่ต้องการค้นหา



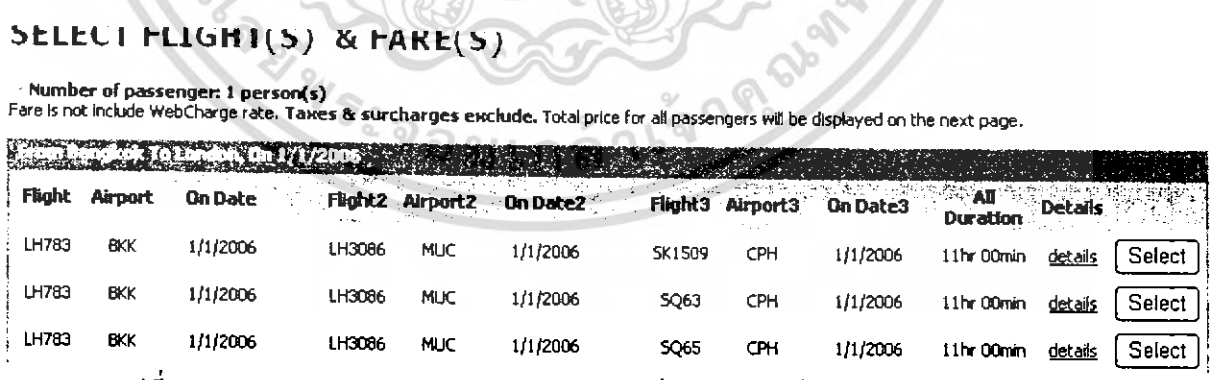
รูปที่ 4.5. ฟอร์มชื่อเมืองต้นทางและปลายทางที่ต้องการ

เมื่อกดปุ่ม "next" หลังจากกรอกรายละเอียดของเที่ยวบินที่ต้องการแล้วก็จะมายังหน้าผลลัพธ์ของการค้นหาเที่ยวบิน โดยผลของการแสดงนั้น ก็จะแสดงทั้งขาไปและขากลับ ดังรูปที่ 4.6.



รูปที่ 4.6. แสดงผลของการค้นหาทั้งขาไปและขากลับ

นอกจากนี้แล้วการแสดงผลยังสามารถที่จะแสดงผลได้หลายโหนดของการเดินทางโดยสามารถเข้าลิงค์ ดูรายละเอียด (details) เพื่อดูรายละเอียดของการเดินทางในหนึ่งครั้งนั้นว่ามีการเดินทางด้วยเที่ยวบินใดบ้าง ดังรูปที่ 4.7. – 4.9.



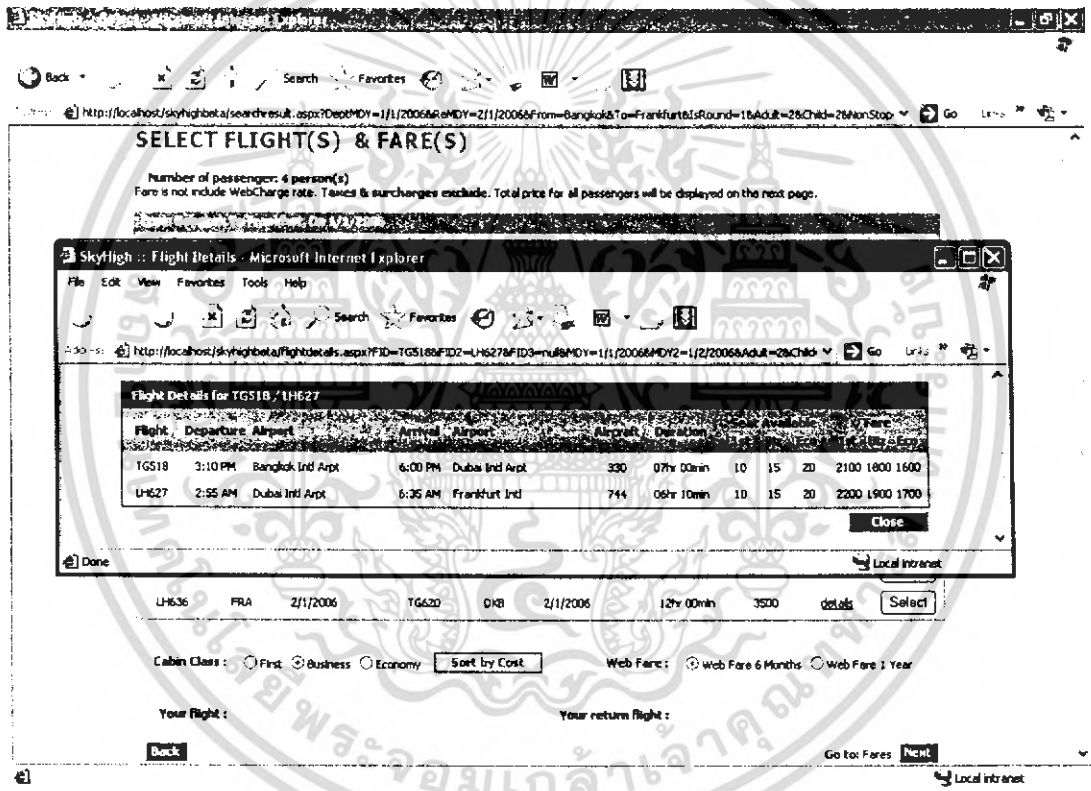
รูปที่ 4.7. การแสดงผลของการค้นหาแบบหลายเที่ยวบินในหนึ่งการเดินทาง

:(S)

es exclude. Total price for all passengers will be displayed on the next page.

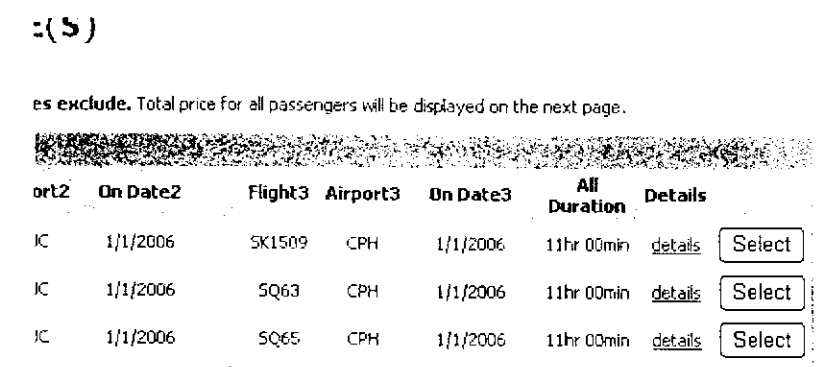
ort2	On Date2	Flight3	Airport3	On Date3	All Duration	Details	
JC	1/1/2006	SK1509	CPH	1/1/2006	11hr 00min	details	Select
JC	1/1/2006	SQ63	CPH	1/1/2006	11hr 00min	details	Select
JC	1/1/2006	SQ65	CPH	1/1/2006	11hr 00min	details	Select

รูปที่ 4.8. ลิงค์ของรายละเอียดการเดินทางในการเดินทางแต่ละเส้นทาง



รูปที่ 4.9. แสดงรายละเอียดการเดินทางในการเดินทางแต่ละเส้นทาง

ในการเลือกเส้นทางแต่ละเส้นทางนั้นสามารถทำการเลือกเส้นทางที่ต้องการได้โดยกดที่ปุ่ม “Select” แล้วหน้าเว็บจะทำการ รีเฟรซ เพื่อแสดงผลของการเลือกเที่ยวบินไหนบ้าง ดังรูปที่ 4.10 และ 4.11.



es exclude. Total price for all passengers will be displayed on the next page.

ort2	On Date2	Flight3	Airport3	On Date3	All Duration	Details	
IC	1/1/2006	SK1509	CPH	1/1/2006	11hr 00min	details	Select
IC	1/1/2006	SQ63	CPH	1/1/2006	11hr 00min	details	Select
IC	1/1/2006	SQ65	CPH	1/1/2006	11hr 00min	details	Select

รูปที่ 4.10. แสดงปุ่มของการเลือกการเดินทาง

Cabin Class : First Business Economy Web Fare : Web Fare 6 Months Web Fare 1 Year

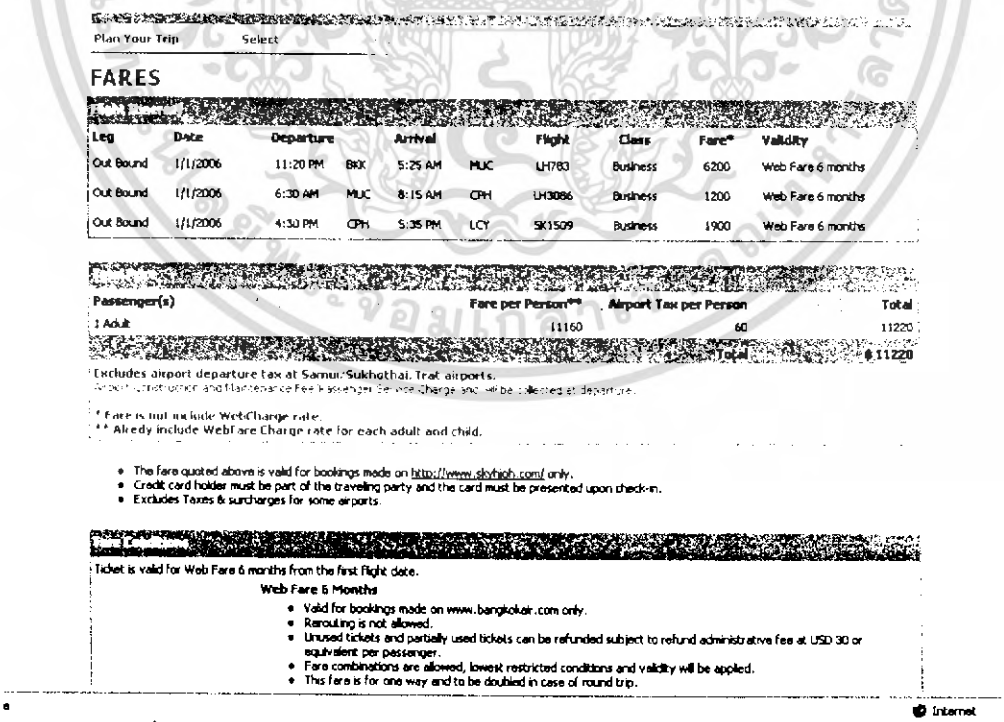
Your flight : You selected LH783 / LH3086 / SK1509

Back Go to: Fares Next

- X = Not have this cabin class
- F = Seat availability is full.

รูปที่ 4.11. แสดงผลหลังจากการเลือก โดยแสดงแต่ละเที่ยวบินของการเดินทางครั้งหนึ่ง

เมื่อทำการกด “Next” แล้วจะมายังหน้าของการแสดงผลลัพธ์ทั้งหมดของการเลือกเที่ยวบินที่จะเดินทาง ราคารวมทั้งหมด และข้อชี้แจงต่างๆ ดังรูปที่ 4.12.



Plan Your Trip Select

FARES

Leg	Date	Departure		Arrival		Flight	Class	Fare*	Validity
Out Bound	1/1/2006	11:20 PM	BKK	5:25 AM	MUC	LH783	Business	6200	Web Fare 6 months
Out Bound	1/1/2006	6:30 AM	MUC	8:15 AM	CPH	LH3086	Business	1200	Web Fare 6 months
Out Bound	1/1/2006	4:30 PM	CPH	5:35 PM	LCY	SK1509	Business	1900	Web Fare 6 months

Passenger(s)	Fare per Person**	Airport Tax per Person	Total
1 Adult	11160	60	11220
		Total	$ 11220

Excludes airport departure tax at Samui/Sukhothai, Trat airports.

* Fare is not include WebCharge rate.
** Already include WebFare Charge rate for each adult and child.

- The fare quoted above is valid for bookings made on http://www.skyhigh.com/ only.
- Credit card holder must be part of the traveling party and the card must be presented upon check-in.
- Excludes Taxes & surcharges for some airports.

Ticket is valid for Web Fare 6 months from the first flight date.

Web Fare 6 Months

- Valid for bookings made on www.bangkokair.com only.
- Rerouting is not allowed.
- Unused tickets and partially used tickets can be refunded subject to refund administrative fee at USD 30 or equivalent per passenger.
- Fare combinations are allowed, lowest restricted conditions and validity will be applied.
- This fare is for one way and to be doubled in case of round trip.

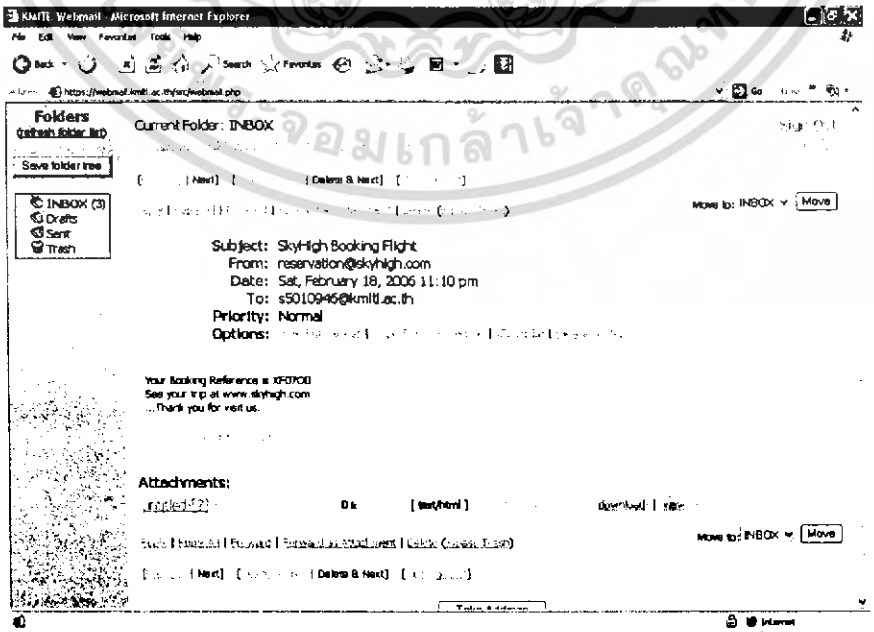
Internet

รูปที่ 4.12. แสดงผลของเที่ยวบินและราคารวมของการเลือกการเดินทาง

เมื่อทำการตอบรับ และกดไปที่ปุ่ม "next" แล้วจะเข้าสู่หน้าของการ ยืนยันการทำการจองโดยต้องทำการกรอกชื่อสมาชิก (user name)และรหัสผ่าน (password) แต่ถ้าไม่ได้เป็นสมาชิกกับทางเว็บก็สามารถที่จะทำการแจ้งชื่อ และประวัติส่วนตัวผ่านทางหน้านี้ได้เลย ดังรูปที่ 4.13.

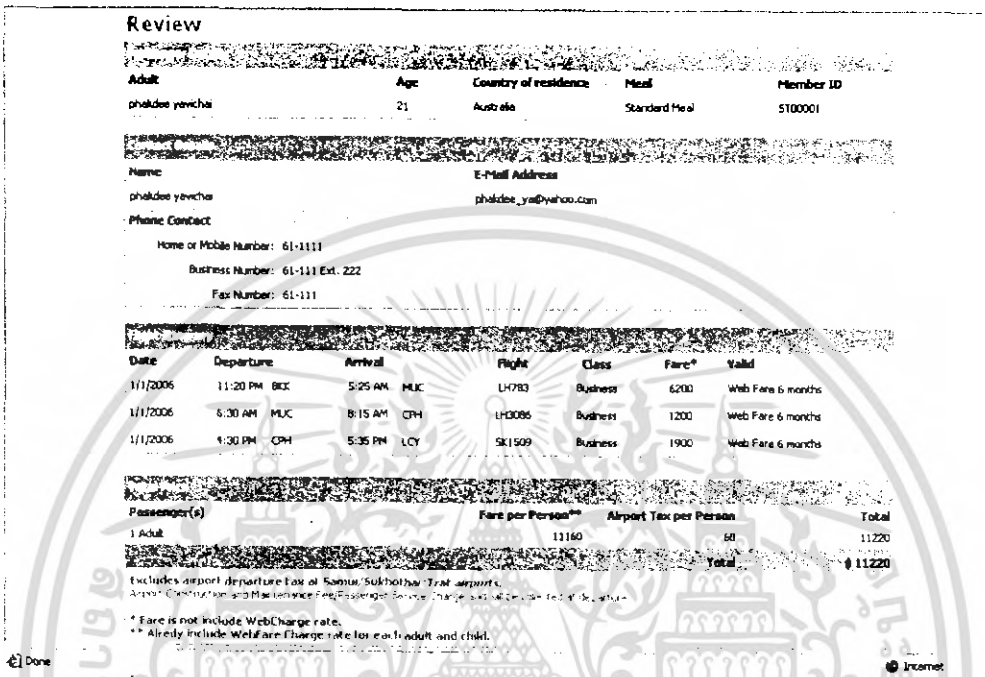
รูปที่ 4.13. แสดงผลหน้าฟอร์มของการกรอกเพื่อขืนยันสิทธิ์ในการจอง

เมื่อกระบวนการจองเสร็จสมบูรณ์แล้วนั้น จะมีการส่งอีเมล์ไปให้กับผู้ใช้เพื่อบอกว่ากระบวนการของการจองได้เสร็จสมบูรณ์แล้ว ดังรูปที่ 4..14.

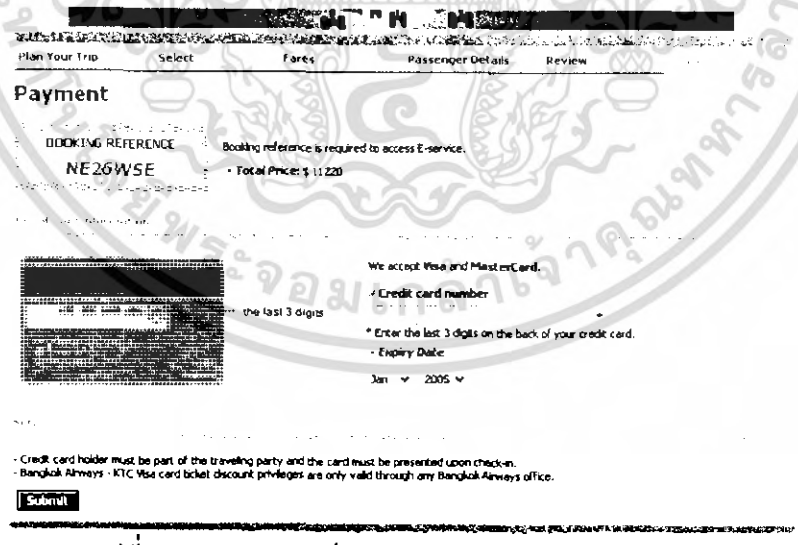


รูปที่ 4.14. แสดงการตอบกลับไปให้กับผู้ใช้ว่าการจองได้เสร็จสิ้นแล้ว

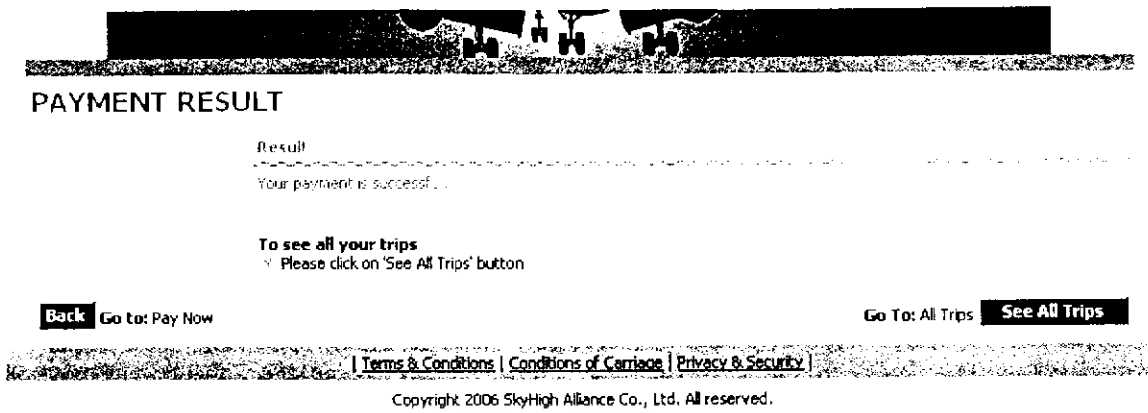
ในขั้นตอนต่อไปจะเป็นหน้าในส่วนของการทบทวน (Review) เพื่อแสดงรายละเอียดต่างๆของเที่ยวบินที่ได้ถูกจองไปแล้วทั้งหมด รวมทั้งยังสามารถที่จะเลือกชำระค่าตั๋วโดยสารได้เลยทันทีทางบัตรเครดิต หรือจะจ่ายทีหลังก็ได้ ดังรูปที่ 4.15. – 4.17.



รูปที่ 4.15. แสดงรายละเอียดต่างๆของเที่ยวบินที่ได้ถูกจองไปแล้วทั้งหมด

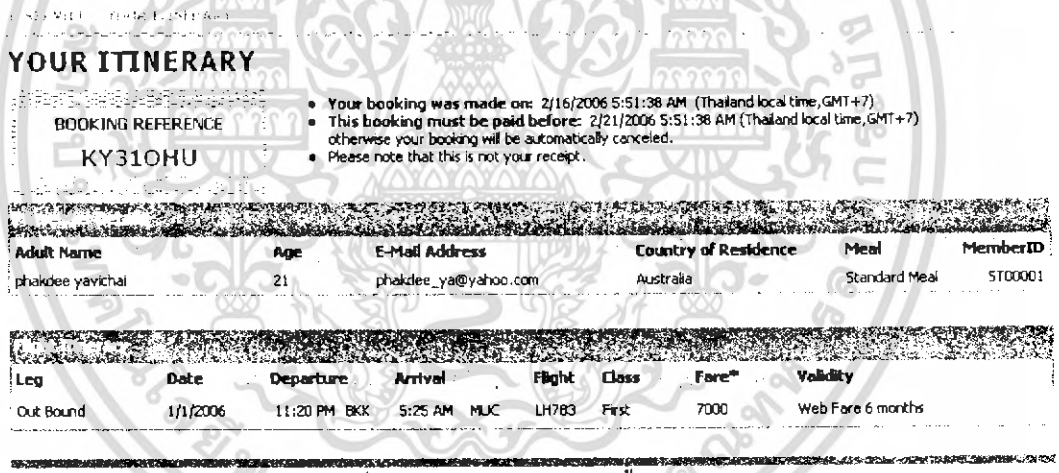


รูปที่ 4.16. แสดงฟอร์มการชำระเงินผ่านทางบัตรเครดิต



รูปที่ 4.17. หน้าเพจแสดงขั้นตอนการชำระเงินที่เสร็จสมบูรณ์

ถ้าทำการเลือกจ่ายแบบจ่ายทีหลังนั้นระบบจะทำการแจ้งหมายเลข Booking Reference ที่ถูกสร้าง ขึ้นมาเพื่อใช้ในการทำธุรกรรมต่างๆ ไม่ว่าจะเป็นชำระเงิน เปลี่ยนแปลงเที่ยวบิน หรือยกเลิกเที่ยวบิน ดังรูปที่ 4.18.



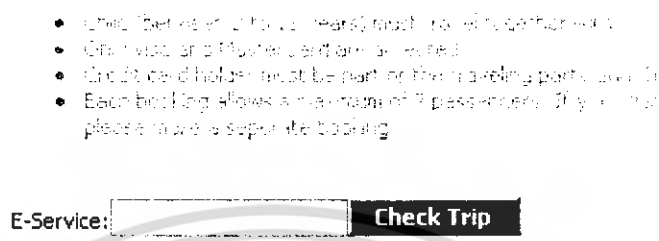
รูปที่ 4.18. แสดงข้อมูลต่างๆของตั๋วที่ได้จองไปแล้วรวมทั้งหมายเลข Booking Reference

4.2 ส่วนของการแก้ไขหรือยกเลิกตั๋วที่ได้จองไปแล้ว

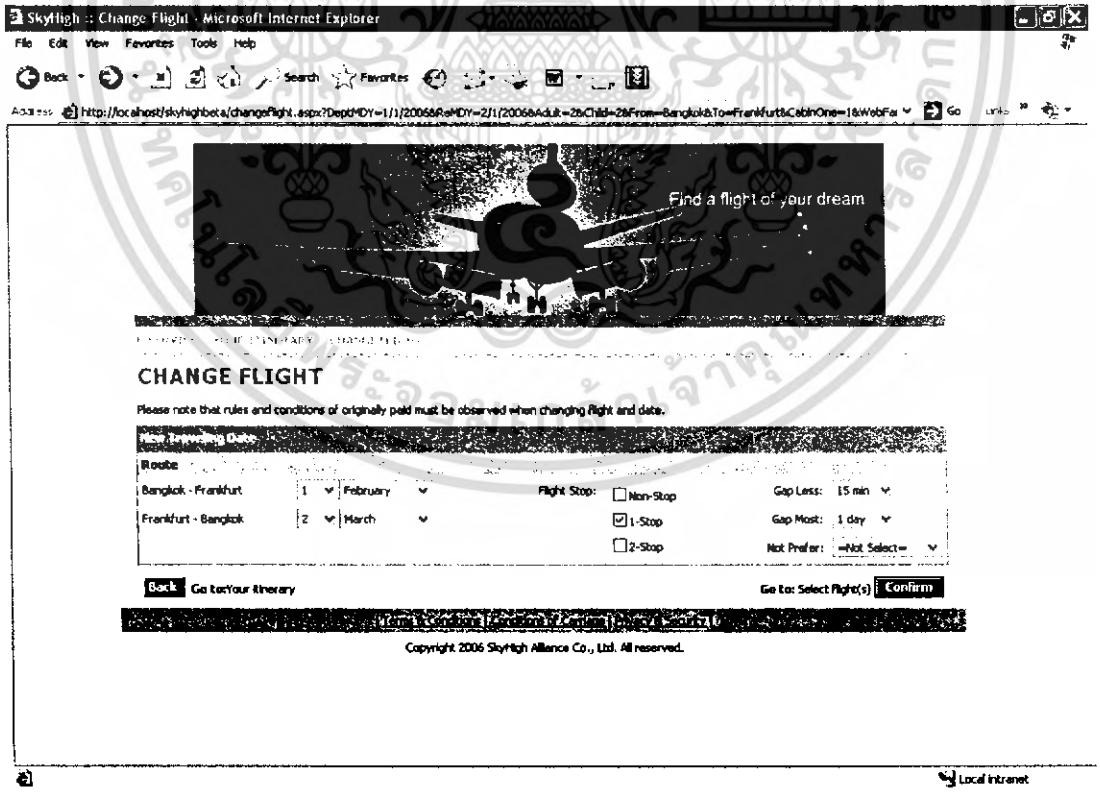
เมื่อสมาชิกที่ได้ทำการจองตั๋วเครื่องบินไปแล้ว แต่กับมีความต้องการที่จะเปลี่ยนแปลงหรือยกเลิกเที่ยวบินต่างๆ ที่ได้จองไว้นั้น ผู้ใช้ก็สามารถที่จะทำการยกเลิกหรือเปลี่ยนแปลงเที่ยวบินได้

โดยการจะทำการเปลี่ยนแปลงนั้นผู้ใช้จะต้องกรอกหมายเลขตั๋วที่ได้รับตอนที่ได้ทำการจองเสร็จแล้ว (Booking Reference) ลงในระบบ E-service (โดยหมายเลข Booking Reference นี้จะเป็นหมายเลขที่ สร้างขึ้นมาตอนที่ได้ทำการจองแล้ว ซึ่งเป็นแหมายเลขที่ใช้แทนการจองในครั้งๆหนึ่ง) และเมื่อทำการใส่หมายเลขตั๋วแล้ว ระบบจะแสดงหน้าเพจของฟอร์มที่จะให้ทำการ

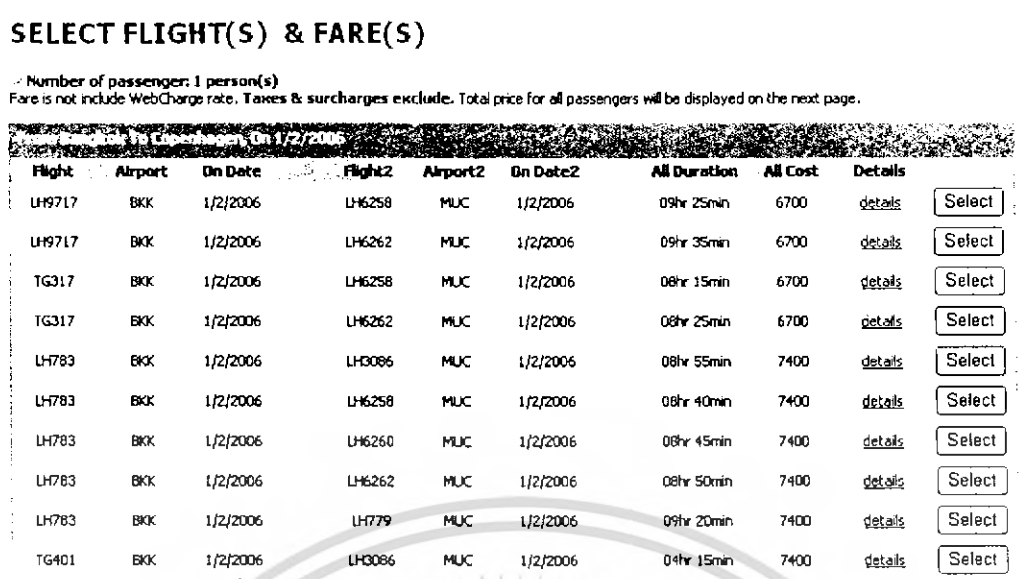
เลือกเที่ยวบินใหม่ โดยมีข้อแม้ว่า เที่ยวบินที่สามารถเลือกใหม่ได้นั้น จะต้องยังคงเป็นเที่ยวบินที่บินจากต้นทางและปลายทางเดิม แต่สามารถเปลี่ยนวันและรายละเอียดอื่นได้เหมือนตอนเลือกใหม่ ดังรูปที่ 4.19. – 4.22.



รูปที่4.19. ฟอร์มกรอกหมายเลขตั๋วเพื่อทำการตรวจสอบรายละเอียดของเที่ยวบิน รวมทั้งการเปลี่ยนหรือยกเลิกเที่ยวบินที่ได้ทำการจองไปแล้ว



รูปที่ 4.20. หน้าเพจของการเปลี่ยนแปลงเที่ยวบินโดยให้ทำการเลือกเงื่อนไขใหม่

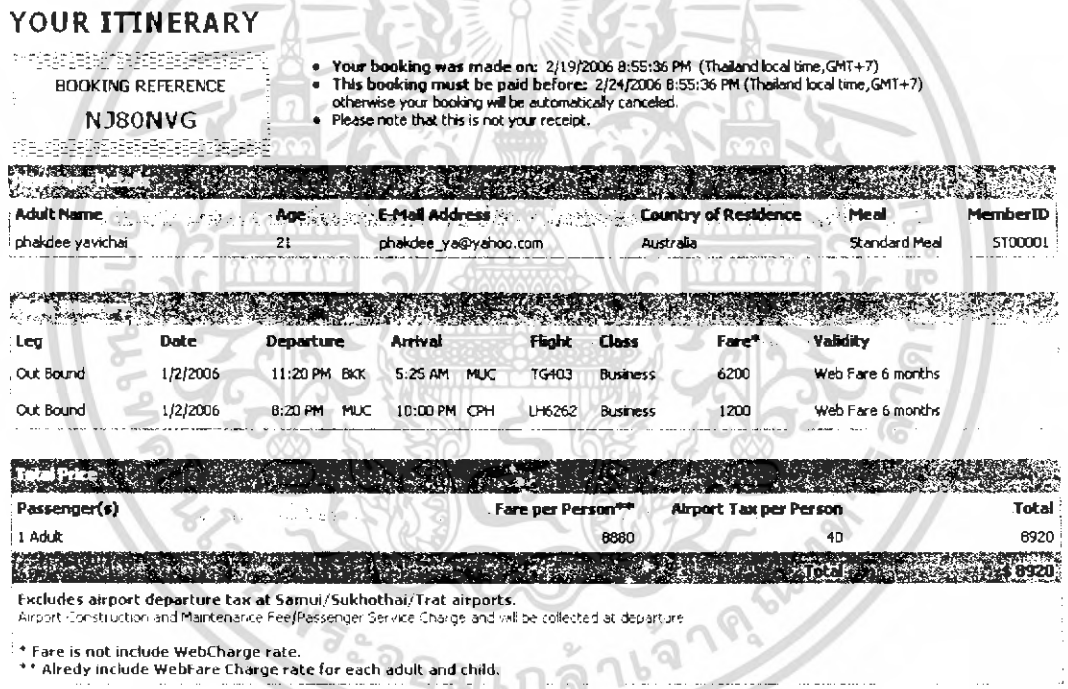


SELECT FLIGHT(S) & FARE(S)

Number of passenger: 1 person(s)
Fare is not include WebCharge rate. Taxes & surcharges exclude. Total price for all passengers will be displayed on the next page.

Flight	Airport	On Date	Flight2	Airport2	On Date2	All Duration	All Cost	Details	
LH9717	BKK	1/2/2006	LH6258	MUC	1/2/2006	09hr 25min	6700	details	Select
LH9717	BKK	1/2/2006	LH6262	MUC	1/2/2006	09hr 35min	6700	details	Select
TG317	BKK	1/2/2006	LH6258	MUC	1/2/2006	08hr 15min	6700	details	Select
TG317	BKK	1/2/2006	LH6262	MUC	1/2/2006	08hr 25min	6700	details	Select
LH783	BKK	1/2/2006	LH3086	MUC	1/2/2006	08hr 55min	7400	details	Select
LH783	BKK	1/2/2006	LH6258	MUC	1/2/2006	08hr 40min	7400	details	Select
LH783	BKK	1/2/2006	LH6260	MUC	1/2/2006	08hr 45min	7400	details	Select
LH783	BKK	1/2/2006	LH6262	MUC	1/2/2006	08hr 50min	7400	details	Select
LH783	BKK	1/2/2006	LH779	MUC	1/2/2006	09hr 20min	7400	details	Select
TG401	BKK	1/2/2006	LH3086	MUC	1/2/2006	04hr 15min	7400	details	Select

รูปที่ 4.21. หน้าเพจแสดงเที่ยวบินที่สามารถทำการเปลี่ยนได้



YOUR ITINERARY

BOOKING REFERENCE
NJ80NVG

- Your booking was made on: 2/19/2006 8:55:36 PM (Thailand local time,GMT+7)
- This booking must be paid before: 2/24/2006 8:55:36 PM (Thailand local time,GMT+7) otherwise your booking will be automatically canceled.
- Please note that this is not your receipt.

Adult Name	Age	E-Mail Address	Country of Residence	Meal	MemberID
phakdee yavichai	21	phakdee_ya@yahoo.com	Australia	Standard Meal	ST00001

Leg	Date	Departure	Arrival	Flight	Class	Fare*	Validity
Out Bound	1/2/2006	11:20 PM BKK	5:25 AM MUC	TG403	Business	6200	Web Fare 6 months
Out Bound	1/2/2006	8:20 PM MUC	10:00 PM CPH	LH6262	Business	1200	Web Fare 6 months

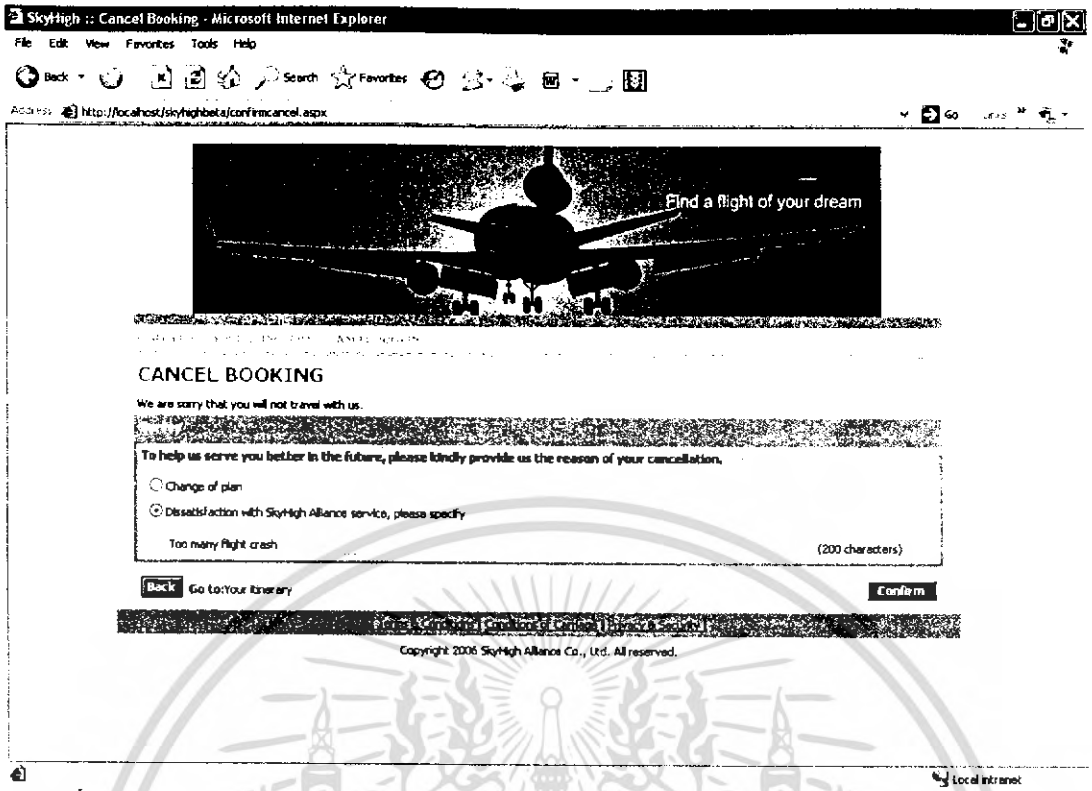
Passenger(s)	Fare per Person**	Airport Tax per Person	Total
1 Adult	6880	40	6920
		Total	6920

Excludes airport departure tax at Samui/Sukhothai/Trat airports.
Airport Construction and Maintenance Fee/Passenger Service Charge and will be collected at departure

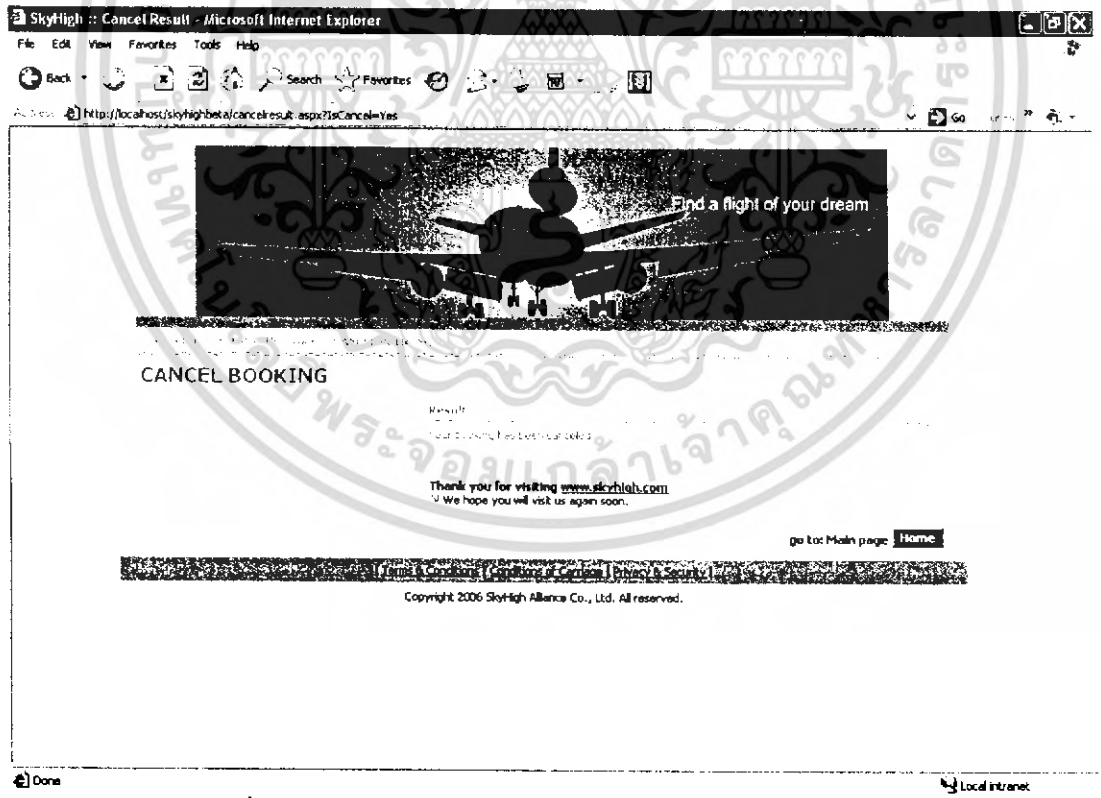
* Fare is not include WebCharge rate.
** Already include WebFare Charge rate for each adult and child.

รูปที่ 4.22. ผลลัพธ์เมื่อได้ทำการเปลี่ยนแปลงเที่ยวบินเสร็จแล้ว

ส่วนในขั้นตอนของการยกเลิกเที่ยวบินนั้น ต้องทำการใส่หมายเลขตั๋ว(Booking Referenee) ที่ระบบ E-service เหมือนกับการทำการเปลี่ยนเที่ยวบิน แล้วระบบจะแสดงหน้าจอของการยืนยันการยกเลิกเที่ยวบินที่ได้จองไว้ โดยสมาชิกจะต้องใส่เหตุผลของการยกเลิกด้วย โดยประโยชน์ของการใส่เหตุผลนั้น จะทำให้ระบบสามารถนำเอาเหตุผลของการยกเลิก มาทำการวิเคราะห์ได้อีกครั้งหนึ่ง ดังรูปที่ 4.23. และ 4.24.

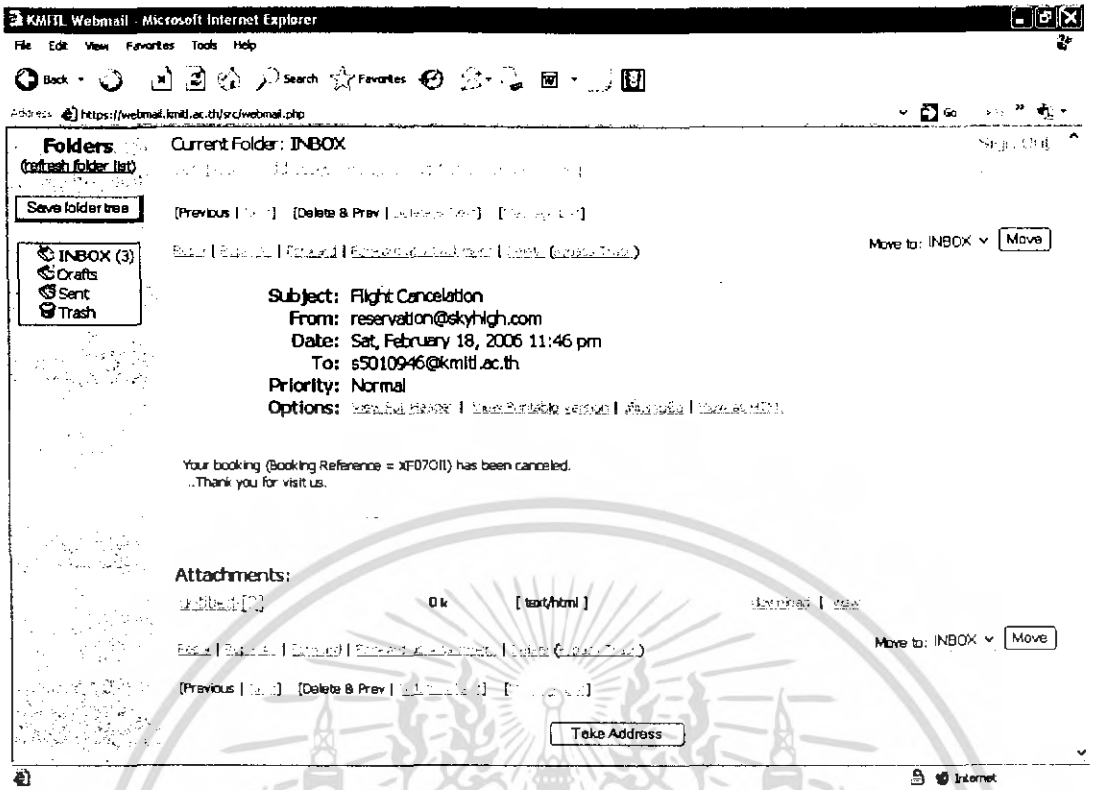


รูปที่ 4.23. แสดงหน้าเพจของการยืนยันการยกเลิกเที่ยวบินและเหตุผลของการยกเลิกเที่ยวบิน



รูปที่ 4.24. แสดงหน้าเพจของผลลัพธ์ของการสั่งยกเลิกเที่ยวบิน

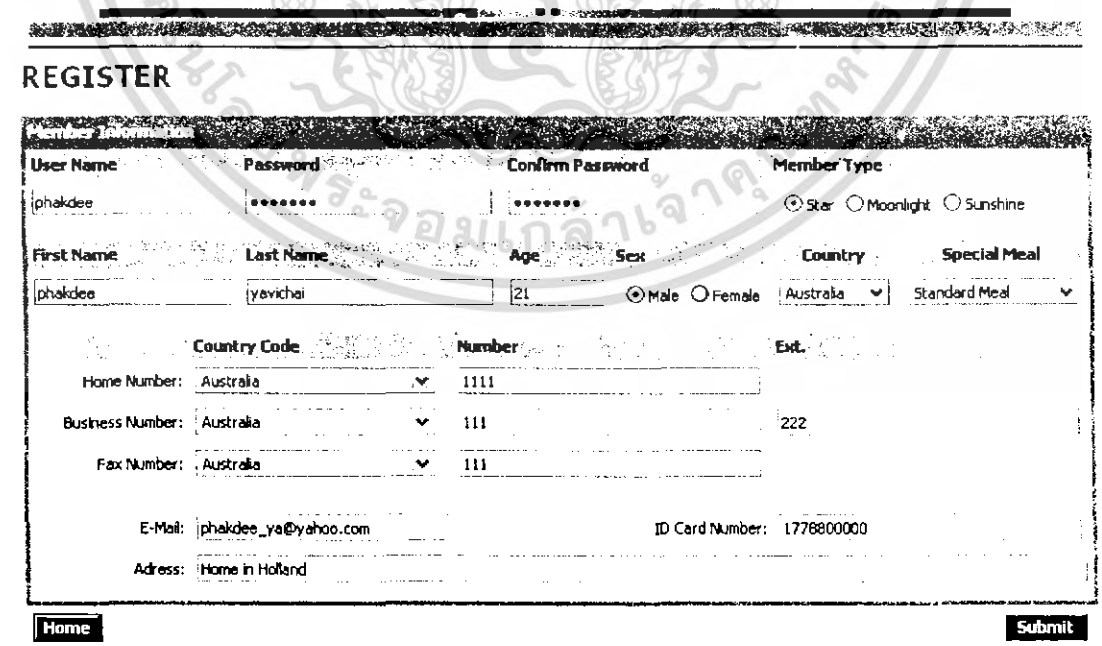
เมื่อทำการยกเลิกเที่ยวบินเสร็จเรียบร้อยแล้วทางระบบจะทำการส่งอีเมล์เพื่อไปบอกกับสมาชิกว่ากระบวนการที่ทำการยกเลิกเที่ยวบินนั้นเสร็จสมบูรณ์แล้ว ดังรูปที่ 4.25.



รูปที่ 4.25. แสดงการตอบกลับไปให้กับผู้ใช้ว่าการยกเลิกเที่ยวบินได้เสร็จสิ้นแล้ว

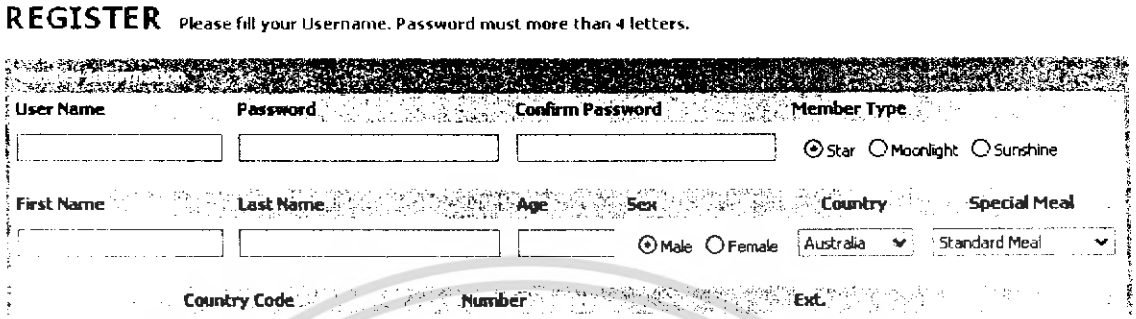
4.3 ส่วนของการสมัครสมาชิก

ในขั้นตอนของการสมัครสมาชิกกับระบบนั้นสามารถที่จะเข้ามาทำการทำการสมัครสมาชิกได้โดยต้องกรอกฟอร์มรายละเอียดของผู้ใช้เพื่อทำการเก็บลงฐานข้อมูลสมาชิก ดังรูปที่ 4.26.



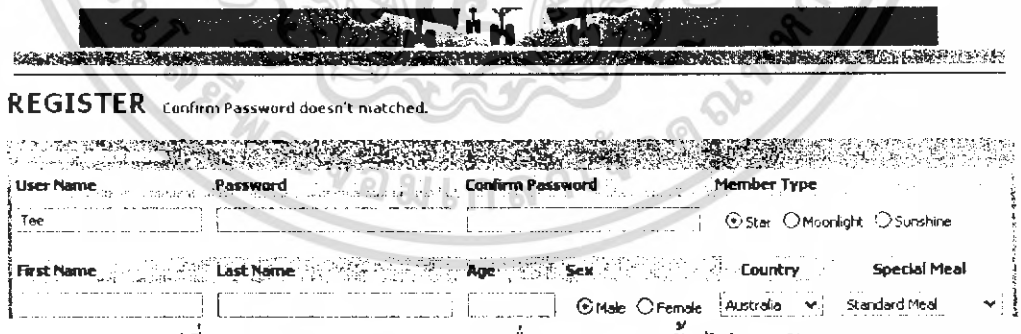
รูปที่ 4.26. แสดงแบบฟอร์มของการสมัครเป็นสมาชิก

โดยการกรอกนั้นจะมีการตรวจสอบข้อความที่ทำการกรอกด้วยไม่ว่าจะเป็นการกรอก User name ซ้ำ ,รหัสผ่านที่ตั้งไม่ตรงกัน ,กรอก User name น้อยกว่าสี่ตัว หรือไม่ได้ทำการกรอกรหัสผ่าน ดังรูปที่ 4.27. - 4.31.

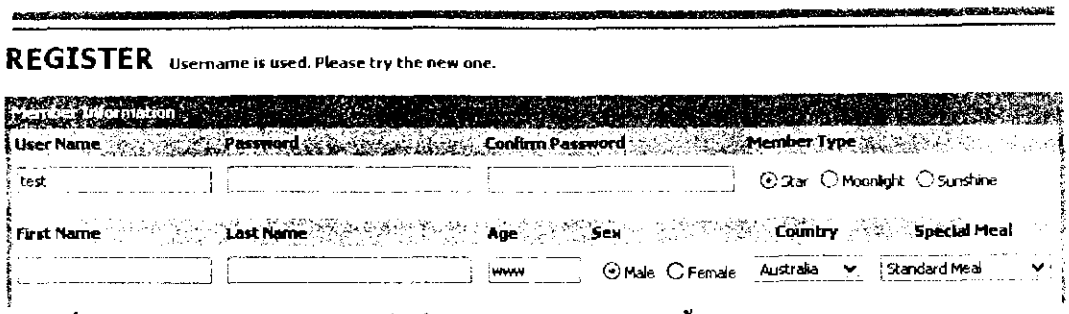


รูปที่ 4.27. แสดงการตรวจสอบเมื่อไม่กรอก Username และ Password

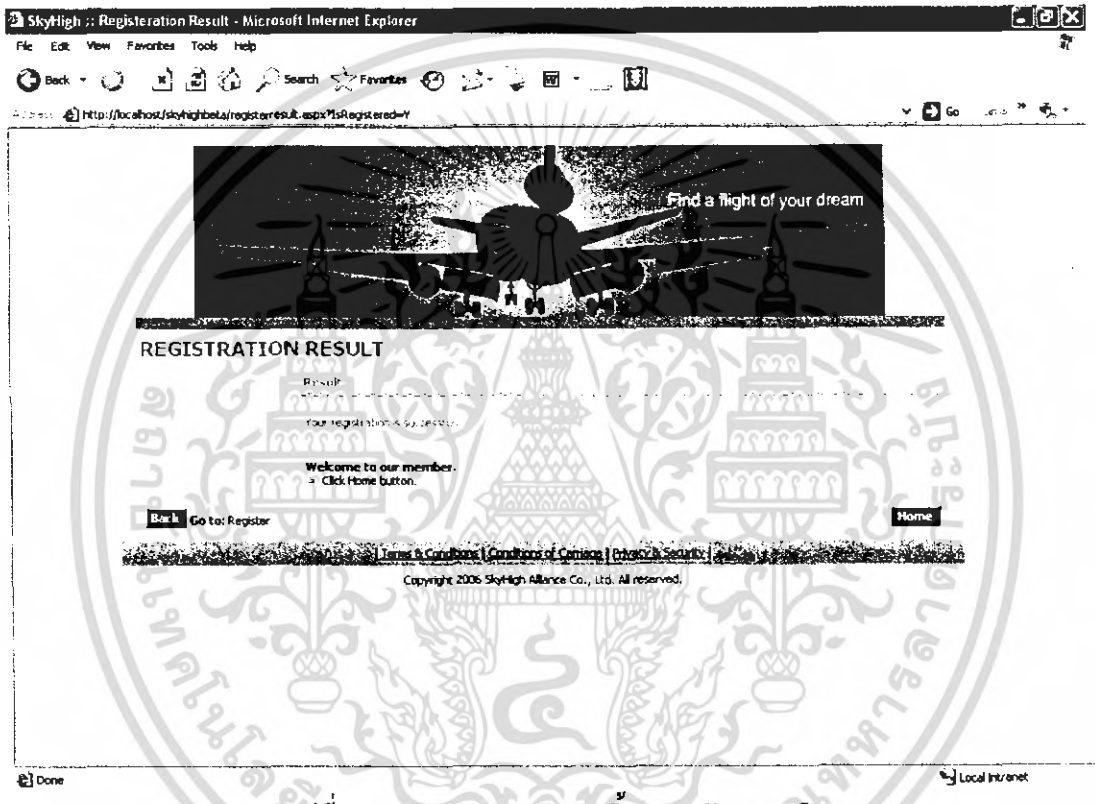
รูปที่ 4.28. แสดงการตรวจสอบเมื่อไม่กรอก Password



รูปที่ 4.29. แสดงการตรวจสอบเมื่อ Password นั้นไม่ตรงกัน



รูปที่ 4.30. แสดงการตรวจสอบเมื่อชื่อสมาชิก (username) นี้ได้ถูกทำการสมัครไปแล้ว

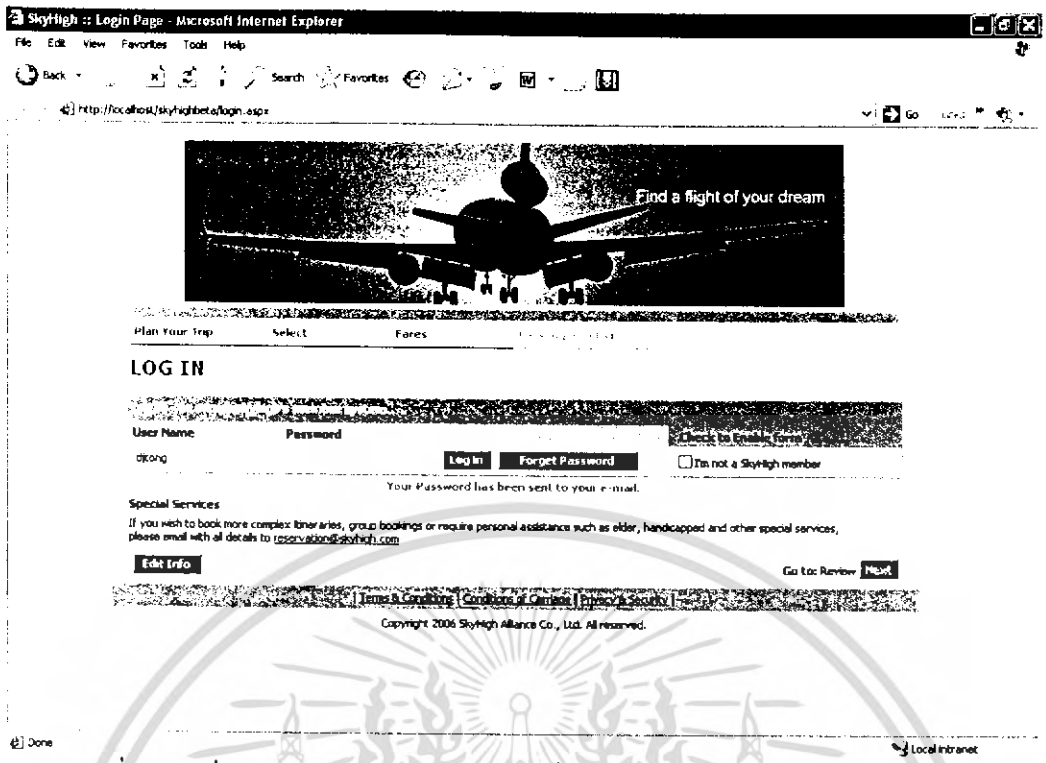


รูปที่ 4.31. แสดงผลของการสิ้นสุดสมัครสมาชิก

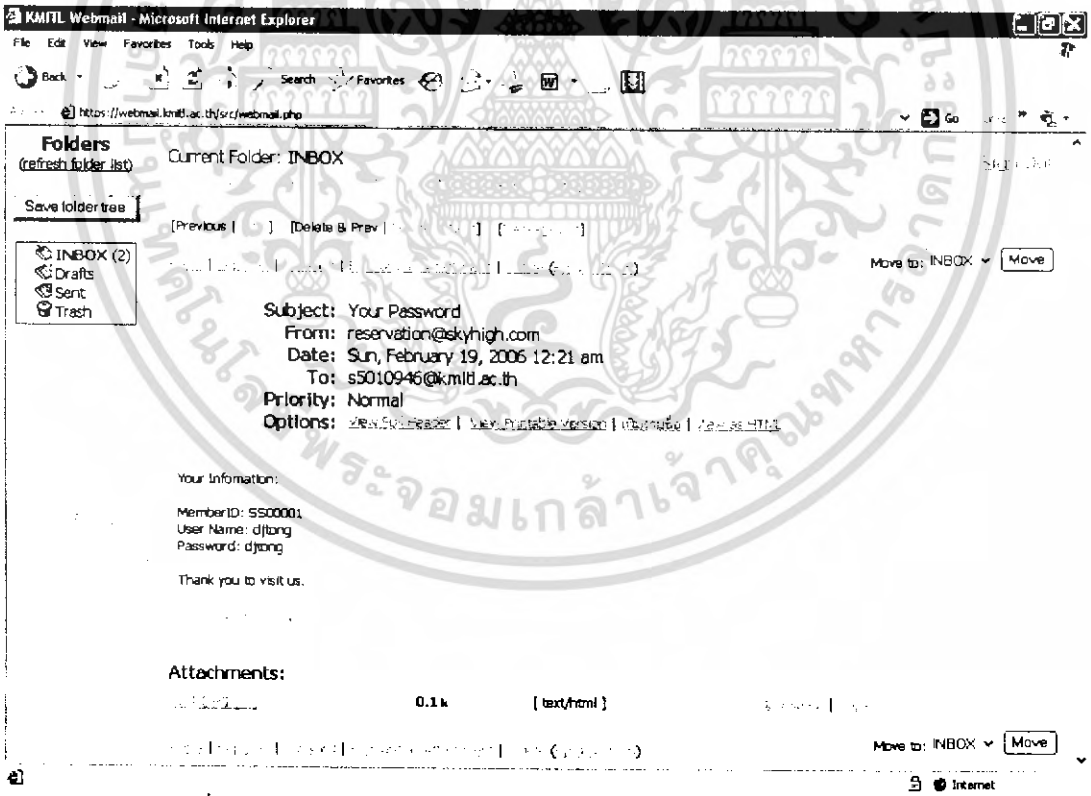
4.4 ส่วนของการเข้าสู่ระบบสมาชิก

เมื่อได้ทำการสมัครสมาชิกแล้ว ในครั้งต่อๆไป เมื่อผู้ใช้ต้องการที่จะเข้าสู่ระบบสมาชิกก็สามารถทำการเข้าสู่ระบบสมาชิก (Log-in) โดยเข้าสู่หน้า(Log-in)และทำการใส่ชื่อผู้ใช้และรหัสผ่านของผู้ใช้ เมื่อชื่อสมาชิกและรหัสถูกต้อง ระบบก็จะทำการสร้างเซสชั่นให้กับสมาชิกเพื่อทำธุรกรรมต่างๆบนเว็บได้ต่อไป

แต่เมื่อผู้ใช้นั้นลืมรหัสผ่านก็สามารถที่จะกรอกชื่อสมาชิกและกดปุ่ม "Forget Password" ระบบจะทำการส่งอีเมล์ไปเจ้าของชื่อสมาชิกนั้นเพื่อบอกรหัสสมาชิก ดังรูปที่ 4.32. และ 4.33.



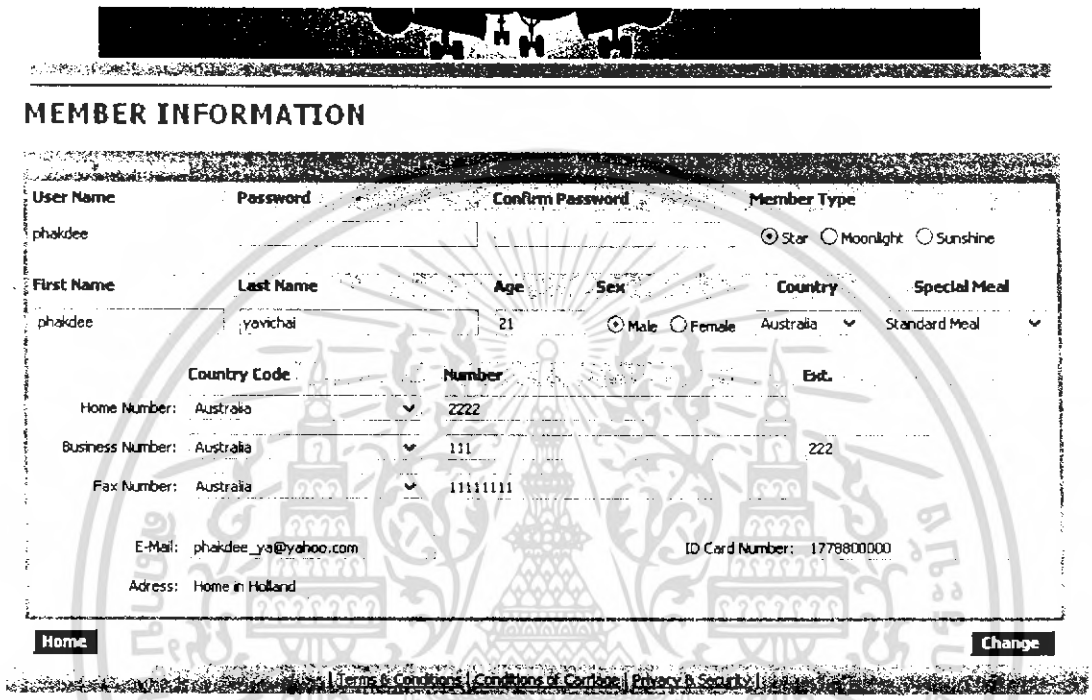
รูปที่ 4.32. เมื่อลืมรหัสผ่านให้ทำการกรอกชื่อสมาชิกแล้วกดปุ่ม Forget Password



รูปที่ 4.33. ระบบจะทำการส่งอีเมล์ให้กับสมาชิกเพื่อบอกรหัสสมาชิก

4.5 ส่วนของการแก้ไขข้อมูลสมาชิก

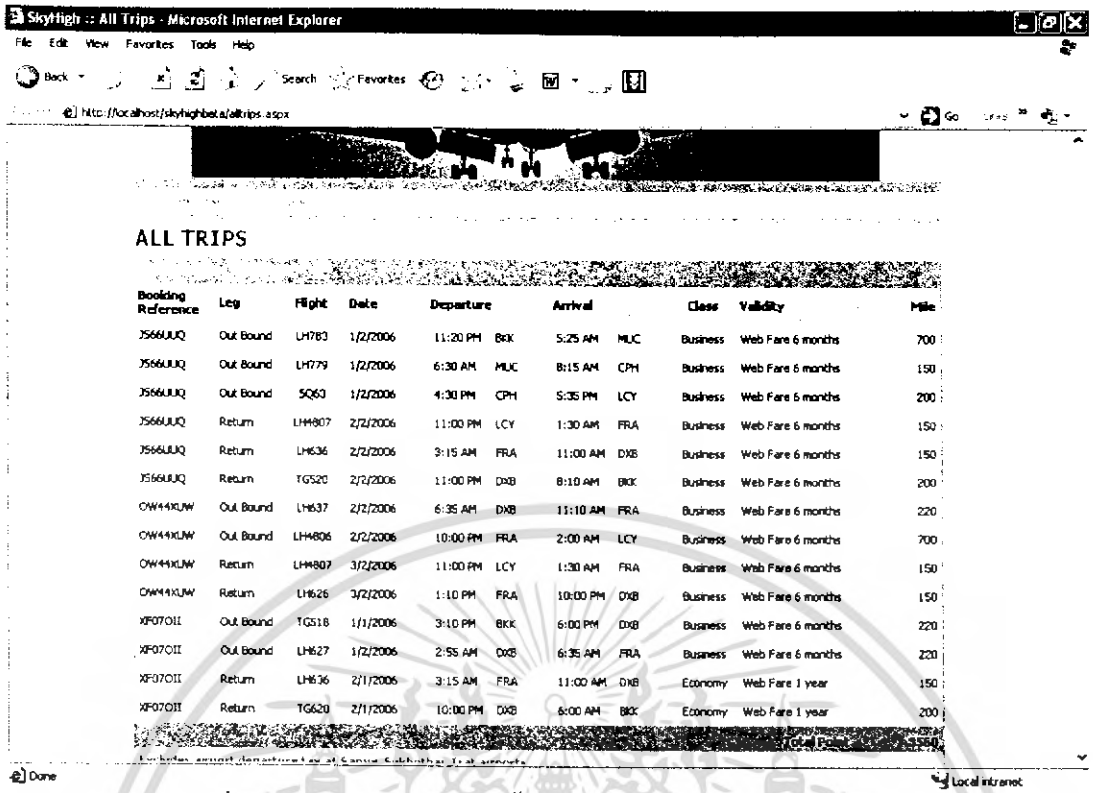
สมาชิกสามารถที่จะทำการแก้ไขรายละเอียดข้อมูลต่างๆของสมาชิกหลังจากที่ได้ทำการสมัครไปแล้วโดยจะต้องทำการเข้าสู่ระบบของสมาชิกก่อนแล้วจึงจะสามารถที่จะทำการเปลี่ยนแปลงข้อมูลของสมาชิกเองได้ ดังรูปที่ 4.34.



รูปที่ 4.34. ฟอร์มของการเปลี่ยนแปลงรายละเอียดของสมาชิก

4.6 ส่วนตรวจสอบประวัติเที่ยวบินที่เคยทำการจองไปแล้ว

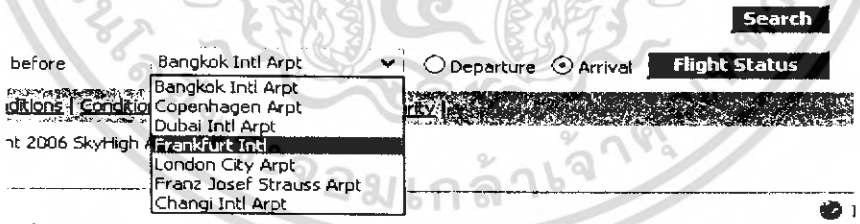
สามารถที่จะทำการตรวจสอบประวัติการเดินทางของสมาชิกที่ได้ทำการเดินทางทั้งหมดที่ผ่านมาเพื่อที่จะดูค่าไมล์เดินทางที่สะสมในการเดินทางทั้งหมดที่ผ่านมา ดังรูปที่ 4.35.



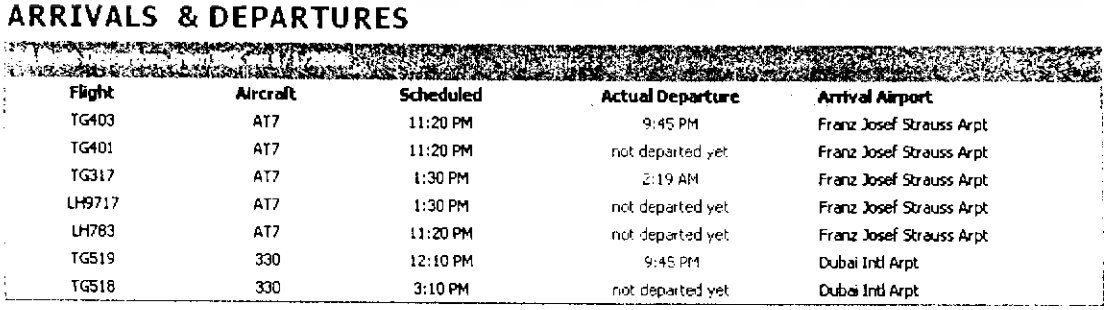
รูปที่ 4.35. แสดงการเดินทางทั้งหมดของสมาชิกรวมทั้งไมล์สะสม

4.7 ทำการตรวจสอบสถานะของเที่ยวบินขาเข้า-ออก ในแต่ละสนามบิน

สามารถที่จะทำการตรวจสอบสถานะการเข้า-ออกของเที่ยวบินต่างๆที่เข้าหรือออกที่สนามบินว่าเที่ยวบินนั้นได้ออกจากสนามบินนั้นไปหรือยัง หรือออกเวลาใดบ้าง โดยทำการเลือกสนามบินขาเข้า-ขาออก ได้ ดังรูปที่ 3.36. และ 3.37.



รูปที่ 4.36. แสดงฟอร์มการเลือกสนามบินที่ต้องการตรวจสอบเที่ยวบิน เข้า-ออก



รูปที่4.37. แสดงผลของการเลือกสนามบินที่เข้า-ออก

4.8 การอัพเดทที่นั่งและสถานะการบินของสายการบินพันธมิตร

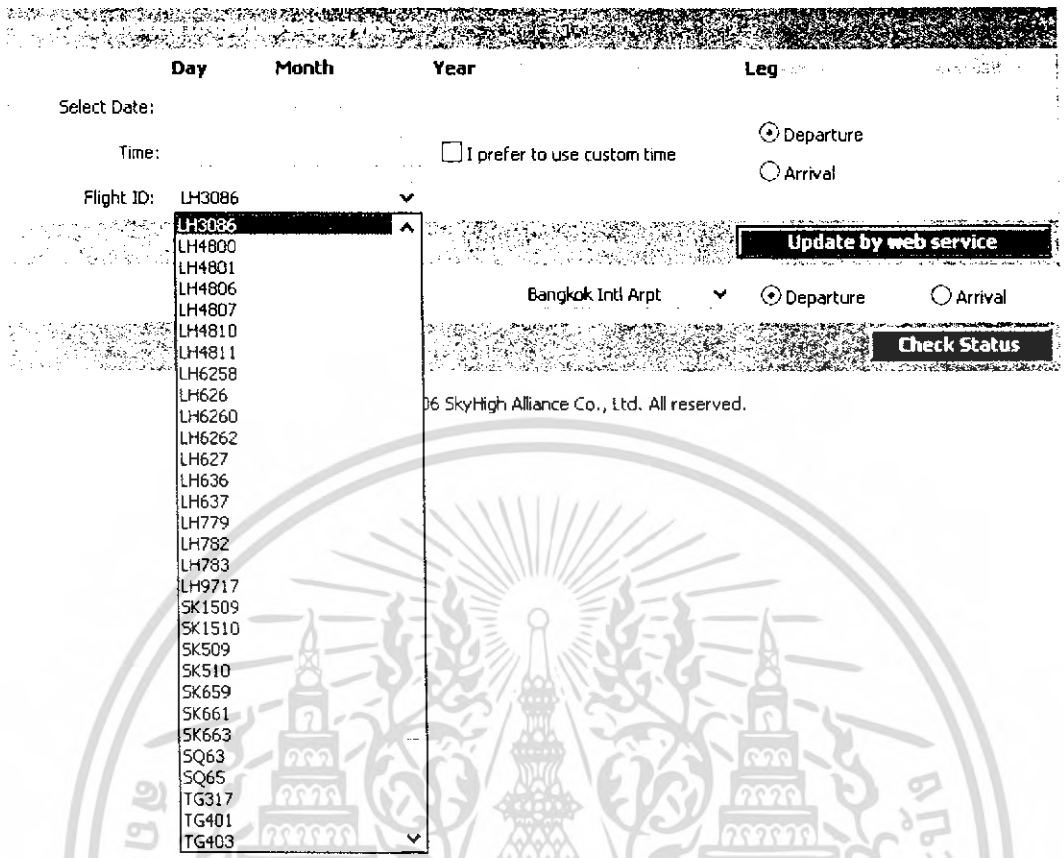
สายการบินพันธมิตรนั้นสามารถที่จะทำการอัพเดทจำนวนที่นั่งของเที่ยวบินของสายการบินตัวเองให้กับทางเว็บของการจองตั๋วโดยสารเครื่องบินหลายสายการบินได้ โดยการเรียกใช้เทคโนโลยีเว็บเซอร์วิส

การอัพเดทของสายการบินนั้นก็จะทำการอัพเดทจำนวนที่นั่งผ่านทางฟอร์มของการอัพเดทที่นั่งของทางเว็บการจองตั๋วโดยสารเครื่องบินหลายสายการบิน ดังรูปที่ 3.38.

รูปที่ 4.38. แสดงฟอร์มของการอัพเดทจำนวนที่นั่งของแต่ละสายการบิน

นอกจากการอัพเดทจำนวนที่นั่งแล้ว สนามบินพันธมิตรก็ยังสามารถที่จะทำการอัพเดทสถานะเที่ยวบินที่เป็นพันธมิตรว่า เที่ยวบินไหนที่บินออกหรือเข้าที่สนามบินที่เป็นพันธมิตรโดยมีการเรียกใช้เทคโนโลยีเว็บเซอร์วิส เช่นกัน

การอัพเดทสถานะเที่ยวบินนั้น จะทำการอัพเดทสถานะเที่ยวบินผ่านทางฟอร์มของการอัพเดทสถานะเที่ยวบินของเว็บการจองตั๋วโดยสารเครื่องบินหลายสายการบิน อีกเช่นกัน ดังรูปที่ 4.39.

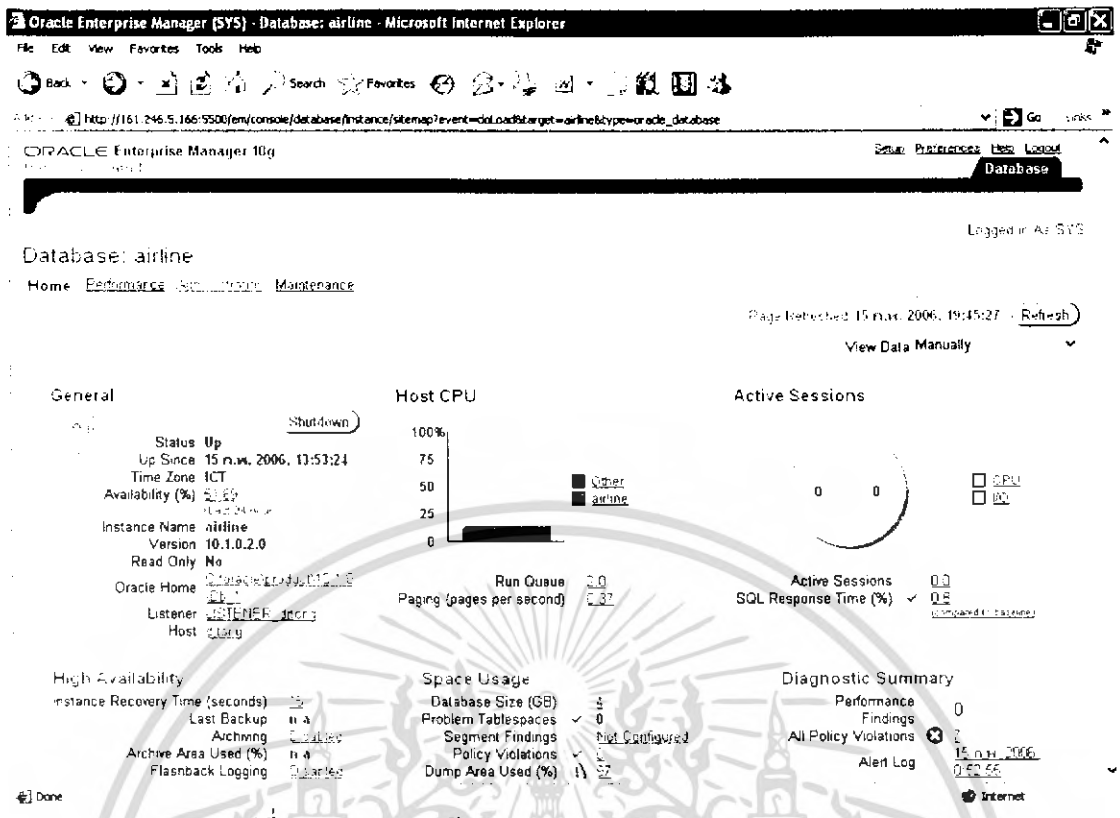


รูปที่ 4.39. แสดงฟอร์มของการอัพเดทสถานะเที่ยวบินของแต่ละสนามบิน

4.9 การจัดการฐานข้อมูลผ่านทางหน้าเว็บเพจ

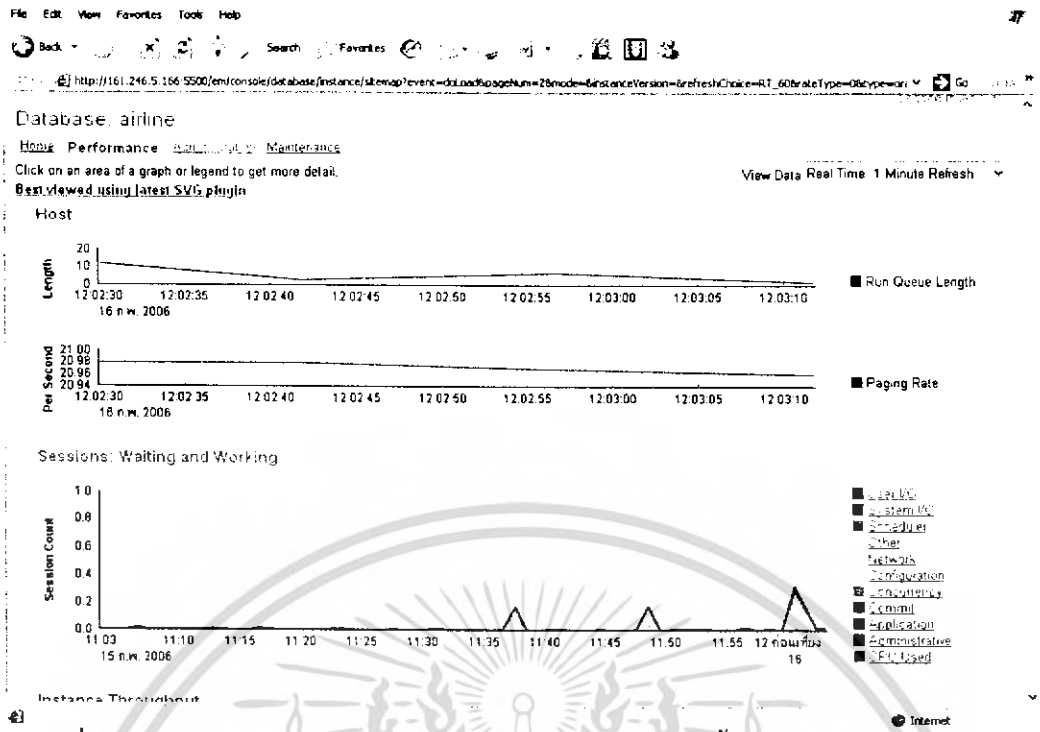
ในตัวการจัดการฐานข้อมูลออราเคิลเท็นจีนั้นได้มีเครื่องมือที่เรียกว่าออราเคิลเอ็นเทอร์ไพร์แมนเนเจอร์ (Oracle Enterprise Manger 10g) ที่จะช่วยให้แอดมินของระบบฐานข้อมูลของเซิร์ฟเวอร์ฐานข้อมูลมีความสะดวกและง่ายต่อการจัดการต่างๆกับระบบเซิร์ฟเวอร์ฐานข้อมูลโดยผ่านทางเว็บเพจที่ทางออราเคิลได้พัฒนาขึ้น ซึ่งจะทำให้แอดมินของระบบฐานข้อมูลของเซิร์ฟเวอร์ฐานข้อมูลสามารถเข้ามาจัดการได้จากทุกสถานที่ ที่สามารถเรียกใช้อินเทอร์เน็ตและมีเว็บบราวเซอร์

โดยเมื่อเข้ามาที่หน้าแรกนั้นจะมีการแสดงสถานะของการทำงานต่างๆ บนระบบเซิร์ฟเวอร์ฐานข้อมูล เช่น พื้นที่ที่เหลือและที่ถูกใช้งาน, จำนวนโฮสซีพียูที่ใช้ไป, การเตือนต่างๆที่เกิดขึ้นในระบบ ดังรูปที่ 3.40.



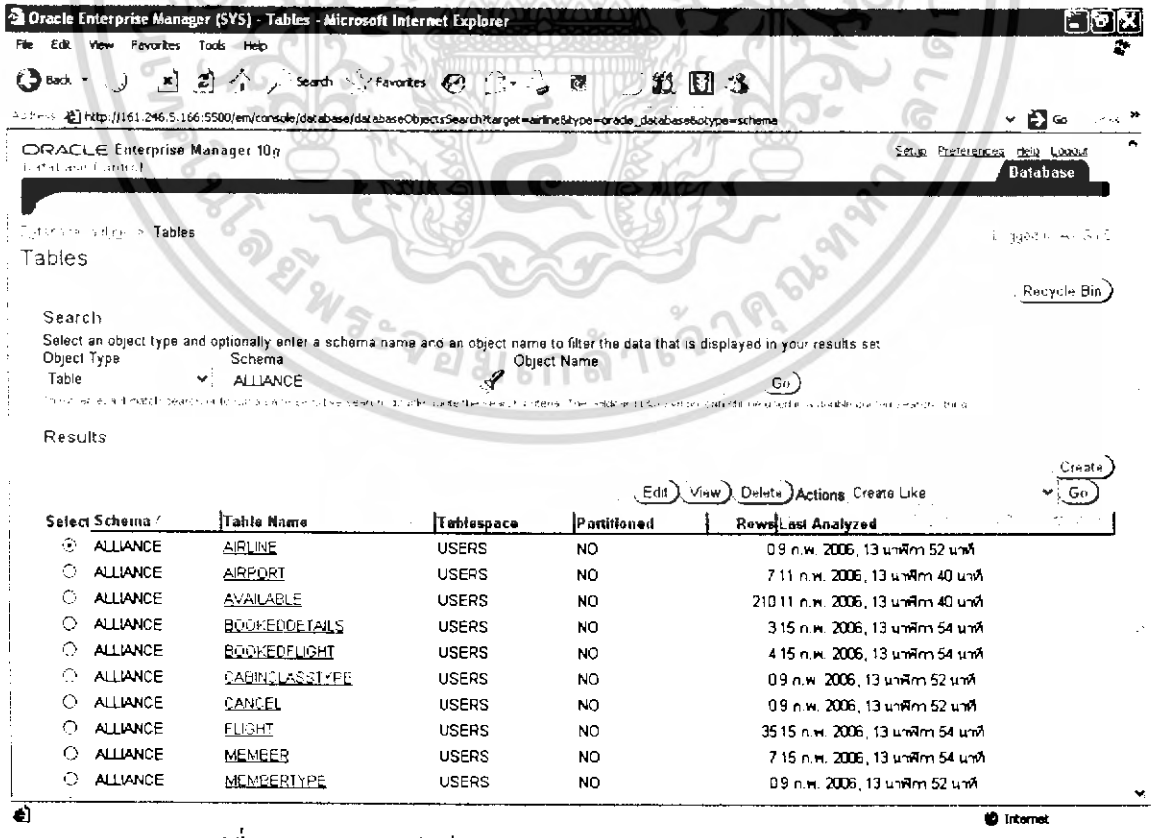
รูปที่ 4.40. แสดงหน้าหลักของ Oracle Enterprise Manager

ใน Oracle Enterprise Manager มีความสามารถที่จะจัดการระบบฐานข้อมูลในส่วนต่างๆ ได้อย่างมากมาย แต่เราจะทำการทดลองและยกตัวอย่างในส่วนที่มีความสำคัญเท่านั้น โดยหนึ่งในความสามารถนั้นก็คือ การตรวจสอบประสิทธิภาพการใช้งาน โดยสามารถที่จะวัดหรือตรวจสอบประสิทธิภาพได้อย่างน้อยที่สุดคือ ทุกๆ 15 นาทีโดยที่แสดงผลออกมาเป็นรูปแบบของกราฟที่สามารถที่จะเข้าใจได้ง่าย ดังรูปที่ 3.41.

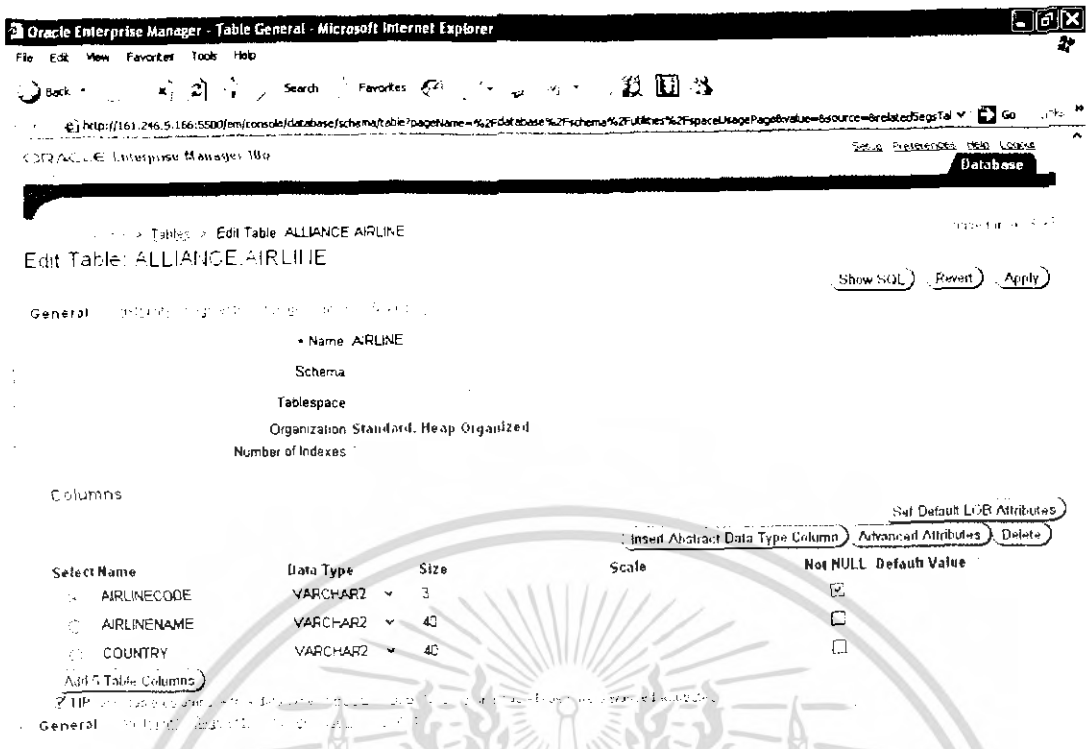


รูปที่ 4.41. แสดงหน้าเว็บของการวัดประสิทธิภาพต่างๆที่เกิดขึ้นบนเซิร์ฟเวอร์ฐานข้อมูล

สามารถที่จะทำการสร้าง ลบ แก้ไข โครงสร้างของตารางได้ อีกทั้งยังสามารถที่จะแก้ไข Constraint ,Data type ของโครงสร้างบนตารางได้ ดังรูปที่ 3.42.และ 3.43.

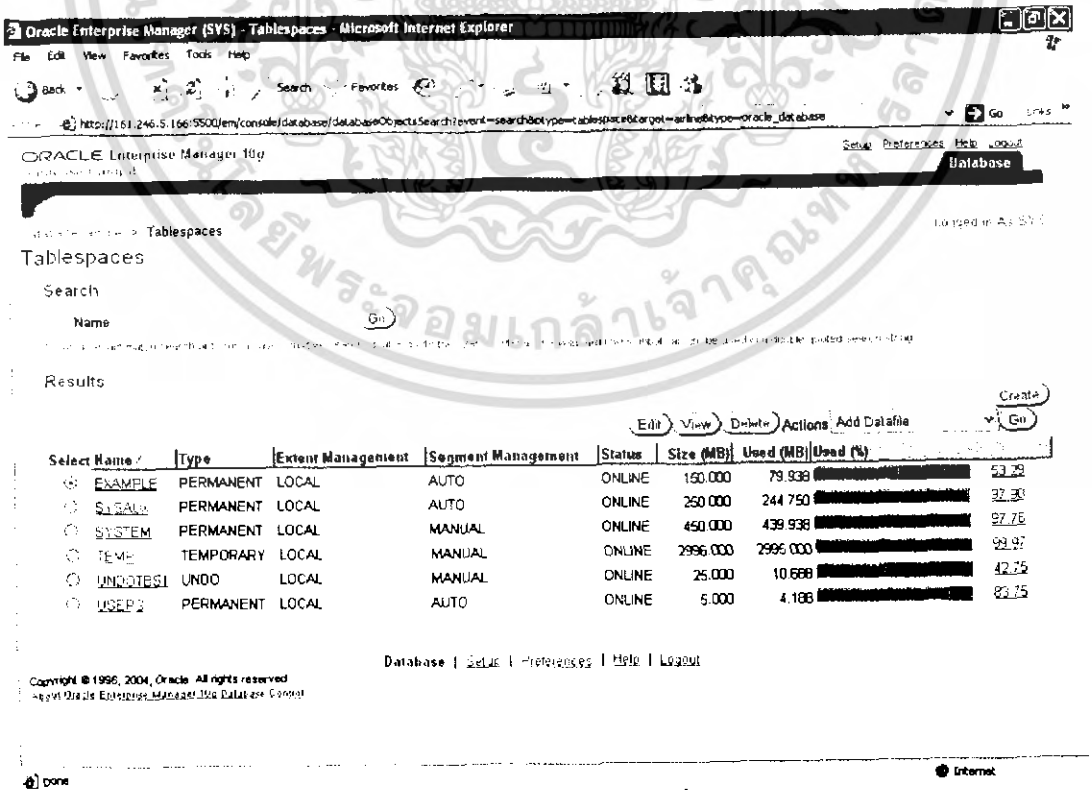


รูปที่ 4.42. แสดงหน้าเว็บของการจัดการกับโครงสร้างของตาราง



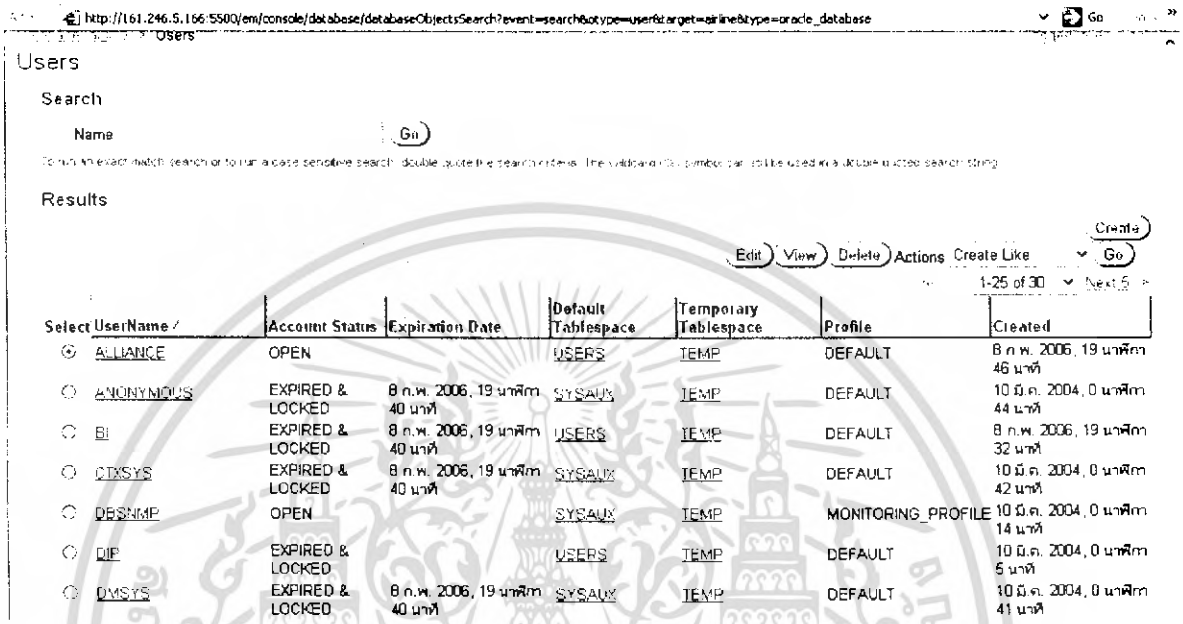
รูปที่4.43. แสดงหน้าเว็บของการจัดการเกี่ยวกับรูปแบบข้อมูลของโครงสร้างตาราง

สามารถที่จะจัดการและดูแลพื้นที่ ที่ใช้เก็บข้อมูลต่างๆ ของผู้ใช้แต่ละคนได้ โดยสามารถจำกัดพื้นที่ในแต่ละตารางได้รวมทั้งกำหนดไฟล์ที่ใช้เก็บจริงๆในเครื่องเซิร์ฟเวอร์ได้ ดังรูปที่ 3.44

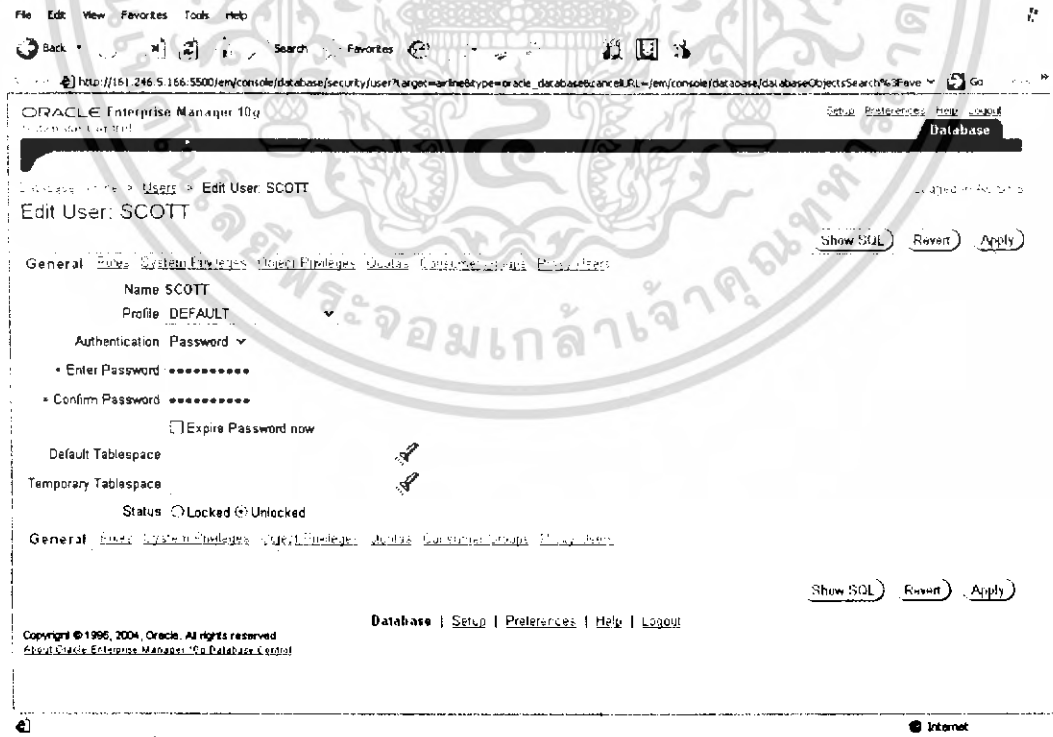


รูปที่ 4.44. แสดงหน้าเว็บของการจัดการเกี่ยวกับเทเบิลสเปซ

สามารถที่จะจัดการเกี่ยวกับยูสเซอร์ต่างๆ ได้ไม่ว่าจะเป็นการ เพิ่ม ลด หรือแก้ไขข้อมูลเกี่ยวกับยูสเซอร์ รวมทั้งสามารถที่จะจำกัดสิทธิ์และกำหนดบทบาทให้แต่ละยูสเซอร์ได้ดังรูปที่ 3.45.และ 3.46.

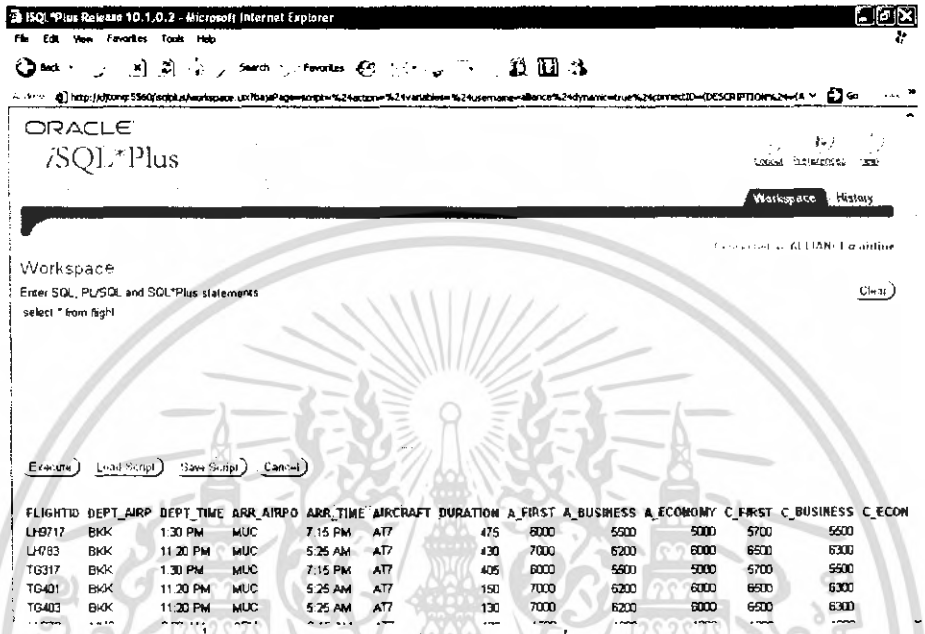


รูปที่ 3.45. แสดงหน้าเว็บของการจัดการเกี่ยวกับยูสเซอร์

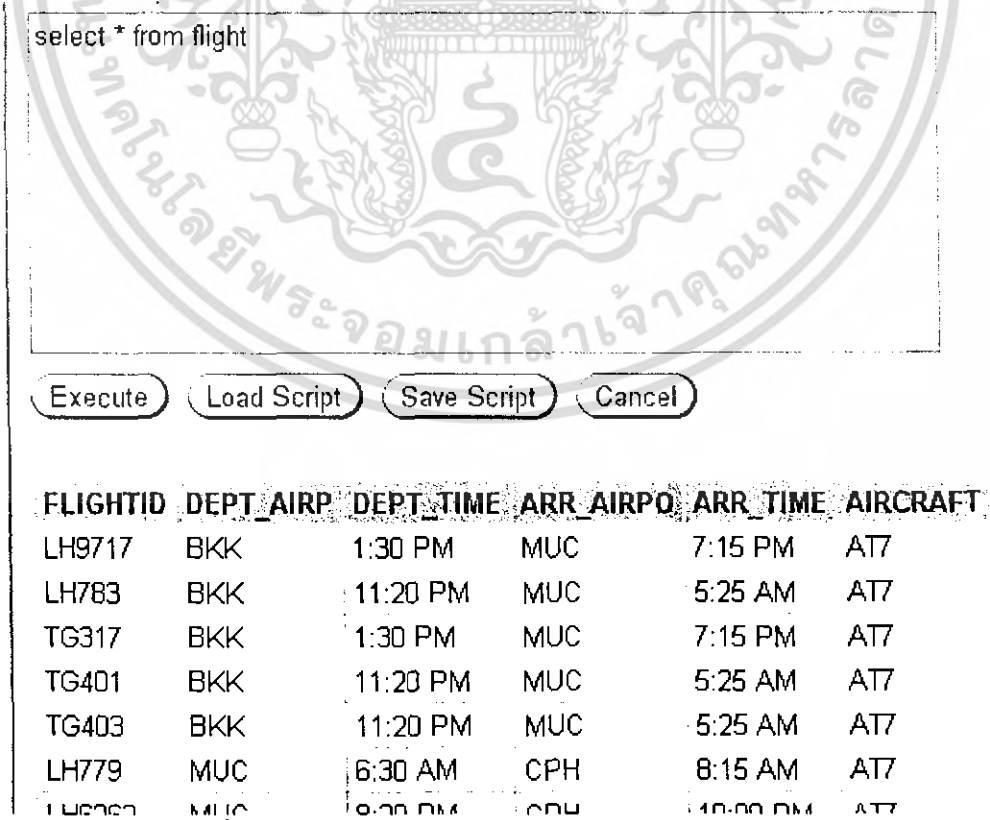


รูปที่4.46. แสดงหน้าเว็บของการจัดการเกี่ยวกับรายละเอียดต่างๆของยูสเซอร์

นอกจากเครื่องมือ Oracle Enterprise Manager แล้วยังมีเครื่องมืออีกตัวหนึ่งคือ iSQL plus ที่ช่วยในการจัดการระบบฐานข้อมูลโดยใช้คำสั่ง SQL ผ่านทางเว็บเพจเพื่อไปจัดการกับฐานข้อมูลต่างๆที่ต้องการซึ่งจะทำให้แอดมินของระบบฐานข้อมูลของเซิร์ฟเวอร์ฐานข้อมูลสามารถเข้ามาจัดการได้จากทุกที่ ที่สามารถเรียกใช้อินเทอร์เน็ตได้และมีเว็บบราวเซอร์ ดังรูปที่ 3.47. และ 3.48.



รูปที่ 4.47. แสดงหน้าเว็บเพจของเครื่องมือ iSQL plus



รูปที่ 4.48. แสดงการเรียกใช้คำสั่งผ่านเครื่องมือ iSQL plus

บทที่ 5
บทวิจารณ์และสรุป

5.1 บทสรุป

ในปัจจุบันเมื่ออินเทอร์เน็ตมีการใช้งานอย่างกว้างขวางขึ้น เว็บแอพพลิเคชั่น (Web Application) ก็เป็นอีกสิ่งหนึ่งที่ได้รับผลกระทบจากเทคโนโลยีของอินเทอร์เน็ต โดยเว็บแอพพลิเคชัน (Web Application) ในปัจจุบันมีขนาดที่ใหญ่และซับซ้อนมากยิ่งขึ้น ส่งผลให้นักพัฒนาต้องทำงานกันเป็นทีม แต่เมื่อทีมมีขนาดใหญ่มากขึ้นรวมทั้งการผลัดเปลี่ยนของนักพัฒนาที่เพิ่มมากขึ้นทุกๆปี ก็อาจจะทำให้นักพัฒนาเกิดความสับสนและความไม่สอดคล้องกันในการพัฒนาเว็บแอพพลิเคชัน (Web Application) นี้ได้

เพราะฉะนั้น การผลิตเว็บแอพพลิเคชันในปัจจุบันนั้นจะต้องคำนึงถึงการบำรุงรักษาหลังจากที่ได้ทำการติดตั้ง หรือใช้งานไปแล้ว โดยจะต้องมีการวางแผนการพัฒนาตามรูปแบบของการพัฒนาเว็บแอพพลิเคชัน และจะต้องศึกษาการใช้งานของเทคโนโลยี หรือวิธีการพัฒนาต่างๆที่เกี่ยวข้องกับการพัฒนาเว็บแอพพลิเคชันนี้

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน

เมื่อได้ทำการพัฒนาเว็บแอพพลิเคชันตามรูปแบบของการพัฒนาเว็บแอพพลิเคชัน โดยที่คำนึงถึงการบำรุงรักษา รวมทั้งคุณสมบัติต่างๆที่จะต้องมีในตัวเว็บแอพพลิเคชันของการจองตั๋วเครื่องบินหลายสายการบินแล้ว การที่จะพัฒนาต่อหรือแก้ไขส่วนต่างๆนั้นมีความสะดวกมากขึ้นกว่าเดิม โดยผู้พัฒนาที่จะพัฒนาต่อนั้น สามารถที่จะทำความเข้าใจส่วนต่างๆของตัวเว็บแอพพลิเคชันนี้ได้ สามารถที่จะทำการเพิ่มเติมจากคุณสมบัติเดิมของตัวเว็บแอพพลิเคชันนี้ได้ อีกทั้งยังสามารถที่จะทำการปรับเปลี่ยนส่วนของเทคโนโลยีที่เกี่ยวข้องในการพัฒนา ที่มักจะมีการปรับเปลี่ยนหรือพัฒนาขึ้นอย่างรวดเร็ว อีกทั้ง เมื่อทำการพัฒนาโดยยึดหลักปฏิบัติในการพัฒนาที่ดี (Best practice) ของแต่ละเทคโนโลยีที่ทำการพัฒนานั้นแล้ว จะทำให้ได้เว็บแอพพลิเคชันต้นแบบ ที่เกิดจากการพัฒนาร่วมกันจากหลายๆเทคโนโลยีที่เกี่ยวข้องกัน

5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1. การจัดการเรื่องการบำรุงรักษา (Maintenance) ในส่วนของการดีไซน์ยูสเซอร์อินเทอร์เฟสนั้นยังคงมีความซับซ้อนและการจัดการได้ค่อนข้างยาก ซึ่งแนวทางการแก้ไขปัญหานี้คือ

ต้องมีการแยกส่วนของ รูปที่จะใช้บนส่วนของหน้าเว็บเพจรวมทั้งมีการจัดการฟอร์มต่างบนหน้าเว็บเพจในรูปแบบที่เป็นแบบไดนามิค

2. การติดตั้งของตัวบริหารการจัดการฐานข้อมูลนั้น ใช้ทรัพยากรของระบบค่อนข้างสูง และอาจที่จะมีปัญหาได้ถ้าไม่มีการเตรียมเครื่องให้พร้อมก่อนที่จะทำการติดตั้งระบบการจัดการฐานข้อมูล แก้ไขได้โดยศึกษาข้อมูลและจัดเตรียมเครื่องให้พร้อม สำหรับความต้องการใช้ทรัพยากรของตัวบริหารการจัดการฐานข้อมูล
3. การเชื่อมต่อกันระหว่างเว็บเซิร์ฟเวอร์กับดาต้าเบสเซิร์ฟเวอร์ในการทดลองนั้น อาจจะมีความผิดพลาดเกิดขึ้นได้ ทำให้เวลาเริ่มใช้งานทุกครั้งต้องมีการปรับค่าและทดลองก่อนใช้งาน การแก้ไขทำได้โดยพยายามให้ผู้ใช้งานมีความเข้าใจการทำงานต่างๆ และคอยอัพเดทตัวระบบเซิร์ฟบ่อยๆเพื่ออุดช่องโหว่และทำให้ผู้พัฒนาสามารถทำการพัฒนาได้อย่างรวดเร็วยิ่งขึ้น

5.4 แนวทางการพัฒนาต่อ

1. พัฒนาในส่วนของการบำรุงรักษา (Maintenance) ของยูสเซอร์อินเทอร์เฟสบนหน้าเว็บเพจ เพื่อให้ง่ายสำหรับผู้พัฒนาที่ต้องการที่จะเปลี่ยนแปลงหรือปรับเปลี่ยนการออกแบบหน้าเว็บเพจ
2. พัฒนาส่วนของเว็บของธนาคารที่ให้บริการเทคโนโลยีเว็บเซอร์วิสเพื่อให้บริการกับการทำการจองตั๋วของเครื่องบินผ่านทางหน้าเว็บเพจ รวมทั้งคำนึงถึงความปลอดภัยในการใช้หมายเลขของบัตรเครดิตด้วย
3. พัฒนาให้ตัวเว็บแอพพลิเคชันนี้สามารถให้บริการเว็บเซอร์วิสกับเว็บไซต์อื่นๆได้รวมทั้งสามารถที่จะออกแบบส่วนของเทคโนโลยีเว็บเซอร์วิสให้สามารถทำการบำรุงรักษา (Maintenance)ได้ง่าย
4. พัฒนาการจองที่นั่งให้ผู้ใช้สามารถทำการเลือกตำแหน่งที่นั่งได้ โดยจะต้องทำการเพิ่มในส่วนยูสเซอร์อินเตอร์เฟสให้มีรูปที่นั่งให้ผู้ใช้เลือกได้ ออกแบบฐานข้อมูลเพื่อรองรับข้อมูลการจองที่นั่งของเครื่องบินในแต่ละแบบ

บรรณานุกรม

ตัวอย่างเอกสารอ้างอิงที่เป็น Web-site

[1] Microsoft Thailand Homepage [Online]. Available
:http://www.microsoft.com/thailand/default.asp.

[2] Microsoft .NET Homepage
:http://www.microsoft.com/net/default.mspx.

[3] Microsoft .NET Frequency Asked Questions [Online]. Available
:http://www.microsoft.com/net/basics_faq.mspx.

[4] Web Services and Others Distributed Technologies Developer Center [Online]. Available
:http://msdn.microsoft.com/webservices/.

[5] Microsoft Visual Studio Developer Center [Online]. Available
:http://msdn.microsoft.com/vstudio/.

[6] Microsoft .NET Framework Developer Center [Online]. Available
:http://msdn.microsoft.com/netframework/.

[7] Microsoft ASP.NET Developer Center [Online]. Available
:http://msdn.microsoft.com/asp.net/.

[8] Microsoft Visual Basic.NET Developer Center [Online]. Available
:http://msdn.microsoft.com/vbasic/.

[9] Best Practices and sample coding for Visual Basic .NET developer [Online]. Available
:http//msdn.microsoft.com/vbasic/using/bestpractices/default.aspx.

[10].NET Developer Center for using Oracle [Online]. Available
:http://www.oracle.com/technology/tech/dotnet/index.html.

[11]Oracle Developer Tools for Visual Studio .NET Center [Online]. Available
:http://www.oracle.com/technology/tech/dotnet/tools/index.html.

[12]Oracle Data Provider for .NET Center [Online]. Available
:http://www.oracle.com/technology/tech/windows/odpnet/index.html.

[13]AI-Dept Blind-search
http://ai-depot.com/Tutorial/PathFinding-Blind.html

ตัวอย่างเอกสารอ้างอิงที่เป็นหนังสือ

[14] ทวีชัย หงษ์สุมาลย์, สงวนชัย สุวรรณชีวะศิริ 2546 **อินไซท์ ASP.NET ฉบับสมบูรณ์** สำนักพิมพ์โปรวิชั่น กรุงเทพฯ

[15] มณีโชติ สมานไทย 2546 **การเขียนโค้ด ASP.NET ฉบับสมบูรณ์** สำนักพิมพ์ Infopress Developer Book กรุงเทพฯ

[16] ศุภชัย จิระรังสินี, ขจรศักดิ์ สังข์เจริญ 2547 **เรียนรู้ Oracle database 10g และภาษา SQL** สำนักพิมพ์เอส เค ซีรี่ย์ กรุงเทพฯ

[17] อำไพ สินลิขิตกุล 2546 **การประยุกต์และออกแบบฐานข้อมูลด้วย Oracle และ SQL Server** สำนักพิมพ์ซีเอ็ดยูเคชั่น กรุงเทพฯ

[18] Thomas M. Pigoski **Practical Software Maintenace** published by John Wiley & Sons,Inc

[19] ผู้ช่วยศาสตราจารย์อภิเนตร อูนากูล 2546 **กระบวนการและวิธีการพัฒนาเว็บแอพพลิเคชันโดยใช้ UML**

[19] ทรงลักษณ์ พิริยะไพโรจน์,สุมนา เกษมสวัสดิ์2544 **เรียนลัด Data Structure ด้วย Visual Basic**