

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ระบบจองตั๋วรถโดยสารประจำทาง

BUS TICKET RESERVATION SYSTEM



H004868



จพ.
ร 314 ร
2550

เลขหมู่.....
เลขทะเบียน..... 04868
วัน,เดือน,ปี..... - 9 ต.ค. 2551

.b.1197820x.....
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ระบบที่ทางศึกษาคณะฯ นั้น ไปอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาคเรียนที่ 2 ปีการศึกษา 2550
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BUS TICKET RESERVATION SYSTEM



**A SYSTEM DEVELOPMENT PROJECT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY**

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
2/ 2007



COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำเห็นประโยชน์ของการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ระบบของตัวรอดโดยสารประจำทาง
นักศึกษา	นายรังสรรค์ สาริกะภูติ
รหัสนักศึกษา	48066824
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	ผศ.ดร.ธนารัตน์ ชลิดาพงศ์

บทคัดย่อ

การจองตัวรอดโดยสารประจำทางผ่านทางอินเทอร์เน็ตโดยใช้เทคโนโลยีเว็บเซอร์วิสเข้ามาช่วย เป็นการนำเอาเทคโนโลยีสมัยใหม่เข้ามาประยุกต์ใช้ โดยความสามารถของระบบนั้นเพื่ออำนวยความสะดวกแก่ผู้โดยสาร ทำให้สามารถจองตัวรอดโดยสารประจำทางผ่านทางอินเทอร์เน็ต โดยมีรายละเอียดดังนี้คือ การลงทะเบียนเพื่อเป็นสมาชิก การเปลี่ยนแปลงแก้ไขข้อมูลสมาชิก การดูรายละเอียดของรถโดยสาร การจอง การยกเลิก และค้นหาอัตราค่ารถโดยสาร

โดยในบทความนี้จะกล่าวถึงการวิเคราะห์และออกแบบระบบรวมถึงรายละเอียดการพัฒนา ระบบดังกล่าว อ้างอิงตามวิธีการเชิงวัตถุ Unified Modeling Language (UML) ซึ่งใช้เป็นเครื่องมือในการสร้างแบบจำลองระบบ เพื่อจำลองฟังก์ชันการทำงานหลักและรายละเอียดการทำงานของฟังก์ชัน และใช้ NetBeans IDE 5.5.1 เป็นเครื่องมือในการพัฒนาระบบ

Title	Bus Ticket Reservation System
Student	Mr. Rangsak Sarikabhuti
Student ID.	48066824
Degree	Master of Science
Programme	Information Science
Academic Year	2007
Advisor	Asst.Prof. Dr.Thanarat Chalidabhongse

ABSTRACT

Bus Ticket Reservation System through web service is a project developed by using JAVA language programming. The ability of the system has been designed for passengers' convenience. With this system, passengers can reserve their bus tickets on the internet per following procedures; registering member information, checking the bus schedule and fare, making and canceling their reservation.

In this article, we will be discussing about the system analysis and the designation for the system development referred by the Unified Modeling Language (UML). It is the equipment in creating the system models for seeking the main function and its details. Furthermore, the NetBeans IDE 5.5.1 is used to be the tool for the system development.

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ ผศ.ดร.ธนรัตน์ ชลิตาพงศ์ อาจารย์ที่ปรึกษาของโครงการพัฒนาระบบงาน ที่กรุณาให้ความรู้และคำแนะนำอันเป็นประโยชน์ ต่อการพัฒนาโครงการนี้ ตลอดจนตรวจสอบแก้ไขจนกระทั่งโครงการสำเร็จลุล่วง

ขอขอบคุณบุคลากรของคณะเทคโนโลยีสารสนเทศทุกท่าน ที่ให้ความช่วยเหลือเรื่องเอกสารและให้ความอนุเคราะห์ตลอดมา

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ สาขาวิชาเทคโนโลยีสารสนเทศ ที่ได้ให้คำแนะนำช่วยเหลือผู้จัดทำ ในเรื่องแนวทางการเขียนและแก้ไขโปรแกรมให้สามารถทำงานได้ประสบผลสำเร็จ

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องด้วยดีเสมอมา ทำให้สามารถทำโครงการสำเร็จลุล่วงได้ด้วยดี

คุณค่าและประโยชน์อันพึงมาจากโครงการพัฒนาระบบงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

รังสรรค์ สาริกะภูติ

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการพัฒนาระบบ.....	1
1.3 ขอบเขตของการพัฒนาระบบ.....	2
1.4 ขั้นตอนของการพัฒนาระบบ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 รายละเอียดของแต่ละบท.....	3
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง.....	4
2.1 เว็บเซอร์วิส (Web Services).....	4
2.1.1 สถาปัตยกรรม Service Oriented Architecture (SOA)	4
2.1.2 เว็บเซอร์วิสสแต็ก (Web Services Stack)	6
2.1.3 XML (eXtensible Markup Language)	7
2.1.4 SOAP (Simple Object Access Protocol)	8
2.1.5 WSDL (Web Service Description Language)	9
2.1.6 UDDI (Universal Description, Discovery and Integration)	10
2.2 UML (Unified Modeling Language).....	10
2.3 Java 2 Platform, Enterprise Edition (J2EE™)	13
2.3.1 Java Application Server)	14
2.3.2 Java Servlets	16
2.3.3 JavaServer Pages (JSP)	16
2.3.4 Enterprise JavaBean (EJB)	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สารบัญ (ต่อ)

	หน้า
2.4 Java API for XML	18
2.4.1 JAXP	19
2.4.2 JAXB	21
2.4.3 SAAJ	21
2.4.4 JAX-RPC	22
2.4.5 JAXR	23
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	25
3.1 ความต้องการของระบบ.....	25
3.2 การออกแบบระบบ.....	25
3.3 การออกแบบเชิงวัตถุโดยใช้ UML.....	26
3.3.1 Use Case Diagram	26
3.3.2 Use Case Description	28
3.3.3 Class Diagram	33
3.3.4 Sequence Diagram	36
3.3.5 Activity Diagram	43
บทที่ 4 การออกแบบฐานข้อมูล.....	45
4.1 ฐานข้อมูลของเว็บเบราว์เซอร์.....	45
4.2 ฐานข้อมูลของเว็บเซอร์วิสธนาคาร.....	47
บทที่ 5 การพัฒนาระบบ.....	49
5.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ.....	49
5.1.1 ฮาร์ดแวร์	49
5.1.2 ซอฟต์แวร์และเครื่องมือ	49
5.2 รายละเอียดการทำงานของระบบ.....	49
5.3 รายละเอียดของการพัฒนาระบบ.....	50

สารบัญ (ต่อ)

	หน้า
5.3.1 หน้าจอส่วนเว็บเพจของบริษัท	50
บทที่ 6 บทสรุป.....	58
6.1 สรุปผลการออกแบบและพัฒนาระบบ	58
6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ	58
6.3 ปัญหาและอุปสรรคในการพัฒนาระบบ	58
6.4 ข้อเสนอแนะและแนวทางการพัฒนาเพิ่มเติม	59
บรรณานุกรม.....	60
ประวัติผู้เขียน.....	61



สารบัญตาราง

ตารางที่	หน้า
2.1 ส่วนประกอบเอกสาร WDSL.....	10
2.2 ชุดคำสั่งใน J2EE.....	13
3.1 แสดงคำอธิบายยูสเคสของการสมัครสมาชิกระบบเพื่อเข้าใช้บริการ.....	28
3.2 แสดงคำอธิบายยูสเคสแสดงรายละเอียดข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูลผู้ใช้บริการ... 28	
3.3 แสดงคำอธิบายยูสเคสที่ใช้แสดงรายละเอียดข้อมูลของตารางการเดินรถ.....	29
3.4 แสดงคำอธิบายยูสเคสของการจองตั๋วโดยสาร.....	29
3.5 แสดงคำอธิบายยูสเคสยกเลิกการจองตั๋วโดยสาร.....	30
3.6 แสดงคำอธิบายยูสเคสตรวจสอบสถานะของการจองตั๋ว.....	31
3.7 แสดงคำอธิบายยูสเคสตรวจสอบหมายเลขบัญชี.....	31
3.8 แสดงคำอธิบายยูสเคสที่ใช้สำหรับหักยอดเงินในบัญชี.....	32
3.9 แสดงคำอธิบายยูสเคสที่ใช้สำหรับคืนยอดเงินในบัญชี.....	32
3.10 แสดงคำอธิบายยูสเคสที่ใช้สำหรับตรวจสอบยอดเงินคงเหลือในบัญชี.....	33
3.11 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	37
3.12 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	37
3.13 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	38
3.14 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	39
3.15 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	40
3.16 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	41
3.17 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	42
3.18 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไคอะแกรม.....	43
4.1 รายละเอียดของตาราง Route	46
4.2 รายละเอียดของตาราง Schedule	46
4.3 รายละเอียดของตาราง Bus_Operate	46
4.4 รายละเอียดของตาราง Ticket	46
4.5 รายละเอียดของตาราง Member	46
4.6 รายละเอียดของตาราง Bustype	47
4.7 รายละเอียดของตาราง Bus	47
4.8 รายละเอียดของตาราง Driver	47

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.9 รายละเอียดของตาราง Card_Type	47
4.10 รายละเอียดของตาราง Credit_Card	48



สารบัญรูป

รูปที่	หน้า
2.1 Service Oriented Architecture (SOA) Model.....	5
2.2 Web Services Stack	6
2.3 Soap Message	8
2.4 ลักษณะการทำงาน Soap Message	9
2.5 กระบวนการ Forward Engineering.....	11
2.6 กระบวนการ Reverse Engineering	12
2.7 โครงสร้างของ Java Application Server	15
2.8 วงจรชีวิตของ Servlets	17
2.9 โครงสร้างของ SAX APIs.....	19
2.10 โครงสร้างของ DOM APIs	20
2.11 โครงสร้างของ XSLT APIs.....	21
2.12 การทำงานของ JAX-RPC.....	22
2.13 สถาปัตยกรรม JAXR.....	24
3.1 ยูสเคสไดอะแกรมของระบบฝั่งเว็บแอปพลิเคชัน.....	26
3.2 ยูสเคสไดอะแกรมของฝั่งเว็บเซอร์วิสใช้ในการตัดเงินและคืนเงินให้กับผู้ใช้บริการ.....	27
3.3 คลาสไดอะแกรมของระบบจองตั๋วรถเดินทางฝั่งเว็บแอปพลิเคชัน.....	34
3.4 คลาสไดอะแกรมของระบบจองตั๋วรถเดินทางฝั่งเว็บเซอร์วิส.....	35
3.5 ซีเควนซ์ไดอะแกรมของการ Register	36
3.6 ซีเควนซ์ไดอะแกรมของการ UpdateData	37
3.7 ซีเควนซ์ไดอะแกรมของ Schedule.....	38
3.8 ซีเควนซ์ไดอะแกรมของการ CheckReserveStatus	39
3.9 ซีเควนซ์ไดอะแกรมของการ ReserveTicket – Valid Account.....	40
3.10 ซีเควนซ์ไดอะแกรมของการ ReserveTicket – Invalid Account	41
3.11 ซีเควนซ์ไดอะแกรมของการ CancelTicket	42

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.12 ซีเควนซ์ไคอะแกรมของ Web Service Payment.....	43
3.13 แอคทิวิตีไคอะแกรมของการจองตั๋วรถโดยสารประจำทาง.....	44
4.1 E-R Diagram ของระบบจองตั๋วรถโดยสารประจำทาง.....	45
4.2 E-R Diagram ของฝั่งเว็บเซอร์วิสธนาคาร.....	47
5.1 หน้าจอหลักของระบบ.....	51
5.2 หน้าจอลงทะเบียนข้อมูลสำหรับสมาชิก.....	52
5.3 หน้าจอตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน.....	52
5.4 หน้าจอแสดงผลการสมัครสมาชิก.....	53
5.5 หน้าจอสำหรับแก้ไข เปลี่ยนแปลงข้อมูลสมาชิก.....	53
5.6 หน้าจอแสดงผลการแก้ไขข้อมูลสมาชิก.....	54
5.7 หน้าจอการสืบค้นตารางการเดินทาง.....	54
5.8 หน้าจอแสดงผลการสืบค้นตารางการเดินทาง.....	55
5.9 หน้าจอการสำรองที่นั่ง.....	55
5.10 หน้าจอยืนยันการสำรองที่นั่ง.....	56
5.11 หน้าจอยกเลิกการสำรองที่นั่ง.....	56
5.12 หน้าจอการสืบค้นราคาค่าโดยสาร.....	57
5.13 หน้าจอแสดงผลการสืบค้นราคาค่าโดยสาร.....	57

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการค้นหาข้อมูลและการสั่งซื้อสินค้าผ่านทางเครือข่ายอินเทอร์เน็ตกำลังเป็นที่นิยมเป็นอย่างมากเนื่องจากสะดวกสบาย ประหยัดเวลา และสามารถให้บริการได้ตลอดเวลา เพราะในปัจจุบันมีร้านค้าทางอินเทอร์เน็ตให้บริการมากมาย ซึ่งระบบดังกล่าวนี้จะช่วยลดปัญหาความแออัดของผู้โดยสารที่ต้องเสียเวลาในการต่อแถวเพื่อซื้อบัตรโดยสาร หรือปัญหาความล่าช้าในการใช้โทรศัพท์ ผู้เขียนจึงมีแนวความคิดว่าหากเราสามารถทำให้ผู้ที่ต้องการซื้อบัตรโดยสาร สามารถเข้าถึงข้อมูลรายละเอียดของเที่ยวการเดินทางได้ง่าย ผู้โดยสารก็สามารถจองบัตรโดยสารได้อย่างสะดวก รวดเร็ว และตรงกับความต้องการของตนเองได้อย่างถูกต้องแน่นอน

เทคโนโลยีเว็บเซอร์วิสถูกคิดค้นขึ้นเพื่อใช้เป็นตัวกลางในการแลกเปลี่ยนข้อมูลกันระหว่างแอปพลิเคชัน ซึ่งเว็บเซอร์วิสคือซอฟต์แวร์โปรแกรมที่ใช้ Extensible Markup Language (XML) ซึ่งมีความสามารถในการเคลื่อนย้ายข้อมูลได้ง่าย (Data Portable) คือสามารถใช้ข้อมูลได้หลากหลาย มีความเป็นมาตรฐานและโปรแกรมส่วนใหญ่สามารถตีความข้อมูลได้ คุณสมบัติเหล่านี้ทำให้ภาษา XML สามารถใช้ร่วมกับเทคโนโลยีเว็บเซอร์วิสที่กล่าวมาข้างต้นได้เป็นอย่างดี ดังนั้นจึงทำให้เว็บเซอร์วิสปราศจากข้อจำกัดต่างๆ เช่น ฮาร์ดแวร์หรือแพลตฟอร์มของแต่ละแอปพลิเคชันที่ใช้งาน โดยการแลกเปลี่ยนข้อมูลของเว็บเซอร์วิสจะกระทำผ่านทางอินเทอร์เน็ต โพรโตคอล โดยจะทำการเปิดให้บริการต่างๆ กับแอปพลิเคชันที่มีการร้องขอ นอกจากนี้ในการพัฒนาระบบในโครงการนี้จะใช้เทคโนโลยีภาษาในการสร้างซึ่งภาษาจาวานั้นก็มีลักษณะเด่นในการเคลื่อนย้ายโค้ดได้ง่าย (Code Portable) คือสามารถนำไปใช้งานที่ใดก็ได้ที่มีการติดตั้ง Java Virtual Machine (JVM) ซึ่งเมื่อใช้งานความสามารถของ XML และภาษาจาวาร่วมกัน จะทำให้การออกแบบเว็บเซอร์วิสสามารถใช้ประโยชน์ได้เป็นอย่างดี

1.2 วัตถุประสงค์ของการพัฒนาระบบ

1. เพื่อศึกษาความเป็นไปได้ในการนำเทคโนโลยีเว็บเซอร์วิสมาประยุกต์ใช้ในการให้บริการโอนเงินของธนาคาร
2. เพื่อเพิ่มประสิทธิภาพในการทำงานในด้านการอำนวยความสะดวกให้กับผู้โดยสารในการใช้บริการระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของการพัฒนาระบบ

ระบบจองตั๋วรถโดยสารประจำทางจัดทำขึ้นเพื่อ

1. ให้บริการแสดงข้อมูลรายละเอียดของตารางการเดินรถ
2. ให้บริการจองตั๋วรถโดยสาร
3. ให้บริการชำระเงินตั๋วรถโดยสาร

ระบบจองตั๋วรถโดยสารประจำทางประกอบด้วยส่วนงานต่างๆ ดังต่อไปนี้

1. ระบบเว็บแอปพลิเคชันของผู้ให้บริการหลัก
โดยให้บริการแสดงข้อมูลรายละเอียดของตารางการเดินรถ ให้บริการจองตั๋วรถโดยสาร และให้บริการชำระเงินตั๋วรถโดยสารผ่านระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร
2. ระบบเว็บเซอร์วิสของผู้ให้บริการธนาคาร

1.4 ขั้นตอนของการพัฒนาระบบ

1. ศึกษาและรวบรวมข้อมูลเพื่อใช้ในการพัฒนาระบบ
2. ศึกษาเทคโนโลยีเว็บเซอร์วิส
3. รวบรวมและวิเคราะห์ความต้องการของระบบ
4. ออกแบบระบบ
5. พัฒนาระบบ
6. ทดสอบระบบ
7. สรุปผลการศึกษาและเสนอแนะข้อคิดเห็น จากการศึกษาและพัฒนาระบบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ลูกค้าได้รับความสะดวกสบาย ประหยัด และสามารถให้บริการได้ตลอดเวลา
2. เพิ่มช่องทางการจำหน่ายสินค้าเพื่อเพิ่มรายได้
3. ได้รับความรู้เกี่ยวกับเทคโนโลยีเว็บเซอร์วิสและภาษาจาวา
4. ได้นำทฤษฎีเกี่ยวกับการ โปรแกรมเชิงวัตถุและ Unified Modeling Language (UML) มาประยุกต์ใช้ในการออกแบบและพัฒนาระบบจริง
5. เป็นแนวทางในการพัฒนาระบบเว็บเซอร์วิสและเว็บแอปพลิเคชันที่มีความซับซ้อนมากขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 รายละเอียดของแต่ละบท

โครงการฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บทด้วยกัน คือ

บทที่ 1 กล่าวถึงความเป็นมาของโครงการ วัตถุประสงค์ในการพัฒนาโครงการ ขอบเขตของโครงการ ขั้นตอนการดำเนินโครงการ ประโยชน์ที่คาดว่าจะได้รับ และรายละเอียดของแต่ละบท

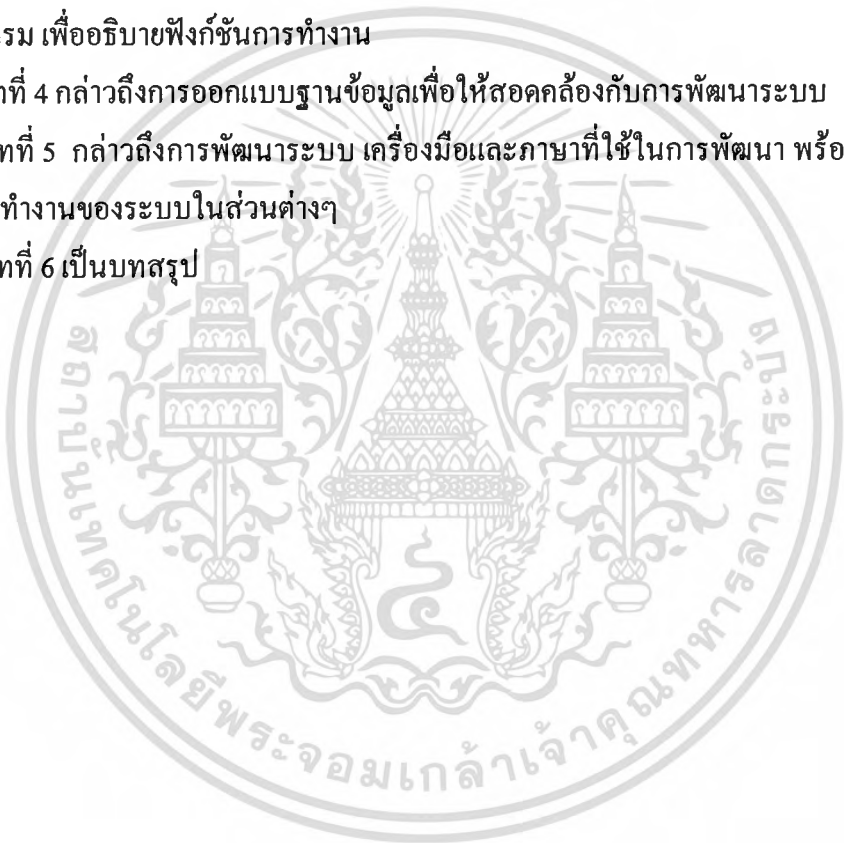
บทที่ 2 กล่าวถึงทฤษฎีและเทคโนโลยีที่เกี่ยวข้องในการพัฒนาระบบ

บทที่ 3 กล่าวถึงการวิเคราะห์และออกแบบระบบ โดยใช้วิธีการเชิงวัตถุ แสดงสถาปัตยกรรมของระบบ ยูสเคสไดอะแกรม คลาสไดอะแกรม ซีเควนซ์ไดอะแกรม และแอกทิวิตีไดอะแกรม เพื่ออธิบายฟังก์ชันการทำงาน

บทที่ 4 กล่าวถึงการออกแบบฐานข้อมูลเพื่อให้สอดคล้องกับการพัฒนาระบบ

บทที่ 5 กล่าวถึงการพัฒนาเว็บ เครื่องมือและภาษาที่ใช้ในการพัฒนา พร้อมทั้งแสดงหน้าจอการทำงานของระบบในส่วนต่างๆ

บทที่ 6 เป็นบทสรุป



บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 เว็บเซอร์วิส (Web Services)

การสื่อสารระหว่างระบบที่มีความแตกต่างกันหรือการรวมระบบเข้าด้วยกัน นับเป็นประเด็นสำคัญที่จะทำให้ ระบบหรือแอปพลิเคชันติดต่อสื่อสาร แลกเปลี่ยนข้อมูลระหว่างกันได้ เนื่องจากทุกวันนี้แต่ละองค์กรต่างมีสารสนเทศเป็นของตนเองทำให้เกิดความแตกต่างกันทั้งทางด้านฮาร์ดแวร์ ซอฟต์แวร์ เมื่อมีการติดต่อสื่อสารกันทำให้เกิดความยากในการรวมระบบหรือติดต่อสื่อสารกันเนื่องมาจากแต่ละระบบต่างมีวิธีการติดต่อสื่อสาร มีระบบสนับสนุนภาษาในการเขียน โปรแกรมที่แตกต่างกัน

เว็บเซอร์วิสเป็นเทคโนโลยีที่ช่วยให้แอปพลิเคชันสามารถติดต่อสื่อสารถึงกันผ่านเครือข่ายโดยไม่ขึ้นอยู่กับแพลตฟอร์มหรือภาษาที่ใช้เขียน โปรแกรม (Programming Language)

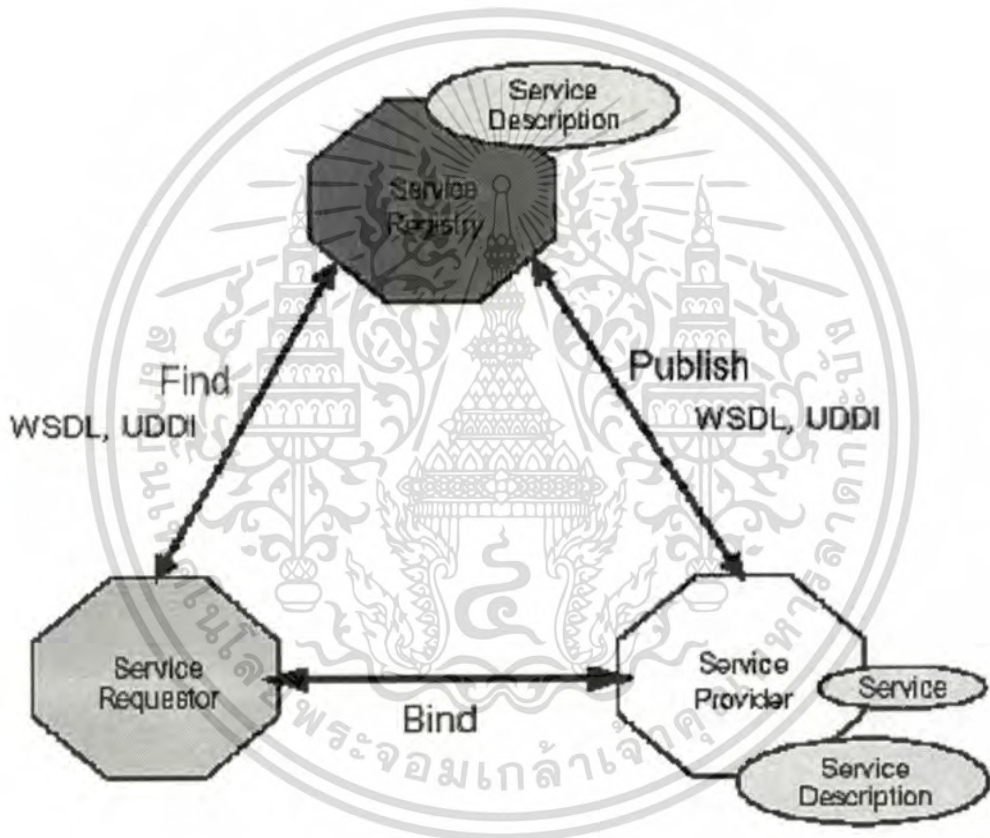
ลักษณะของเว็บเซอร์วิส (Character of Web Services)

1. เว็บเซอร์วิสมีลักษณะอยู่ในรูปแบบของทรัพยากรเว็บ (Web Resource) ซึ่งสามารถเข้าถึงได้จากแพลตฟอร์มใดๆ ก็ได้ โดยอาศัยโปรโตคอลพื้นฐาน เช่น HTTP
2. เว็บเซอร์วิสจัดเตรียมส่วนอินเตอร์เฟซ ซึ่งสามารถเรียกใช้ได้จากโปรแกรมใดๆ
3. เว็บเซอร์วิสโดยทั่วไปจะมีการจดทะเบียนของบริการที่ web service register ซึ่งจะเป็นหน่วยที่ให้ผู้บริการสามารถเข้ามาค้นหาบริการที่ต้องการ
4. เว็บเซอร์วิสสนับสนุนการเชื่อมต่อระหว่างระบบแบบ loosely couple ซึ่งมีการสื่อสารแลกเปลี่ยนข้อมูลระหว่างกัน โดยใช้ข้อความแบบ XML

2.1.1 สถาปัตยกรรม Service Oriented Architecture (SOA)

เว็บเซอร์วิสมีพื้นฐานมาจากสถาปัตยกรรม Service Oriented Architecture (SOA) ซึ่งสถาปัตยกรรม SOA จะเข้ามาช่วยให้ระบบใหม่สามารถกับระบบที่มีอยู่เดิม (Legacy System) ได้ และระบบที่มีความแตกต่างกันทางด้านฮาร์ดแวร์ ซอฟต์แวร์สามารถสื่อสารกันได้ (Heterogeneous) สถาปัตยกรรม Service Oriented Architecture จะแสดงถึงความสัมพันธ์ของ 3 ส่วนด้วยกัน คือ ผู้ร้องขอบริการ ผู้ให้บริการ และผู้จดทะเบียนบริการ (Kreger.2001)

- ผู้ร้องขอบริการ (Service Requester) ทำหน้าที่เป็นเซิร์ฟเวอร์ให้บริการเว็บเซอร์วิสคอมพิวเตอร์ โดยที่คอมพิวเตอร์ที่จะให้บริการ จะมีการเผยแพร่ (Publish) ลักษณะการให้บริการและวิธีการเรียกใช้ เช่น เอกสาร WSDL ไปลงทะเบียนไว้กับนายหน้าซื้อขาย (Service broker)
- ผู้ให้บริการ (Service Provider) ทำหน้าที่เรียกใช้บริการจากผู้ให้บริการ ถ้าหากยังไม่รู้วิธีการเรียกใช้ หรือลักษณะการให้บริการ ก็สามารถทำการค้นหา (Find) ได้จากนายหน้าซื้อขาย
- ผู้จดทะเบียนบริการ (Service Registry) ทำหน้าที่เป็นตัวกลางระหว่างผู้ร้องขอบริการและผู้ให้บริการ โดยให้ผู้ให้บริการมาประกาศบริการที่ตนเองได้สร้างขึ้น และให้ผู้ร้องขอบริการมาค้นหาบริการที่ต้องการ



รูปที่ 2.1 Service Oriented Architecture (SOA) Model

ดังรูปที่ 2.1 มีขั้นตอนการทำงานพื้นฐาน (Operation) 3 ขั้นตอน ดังนี้

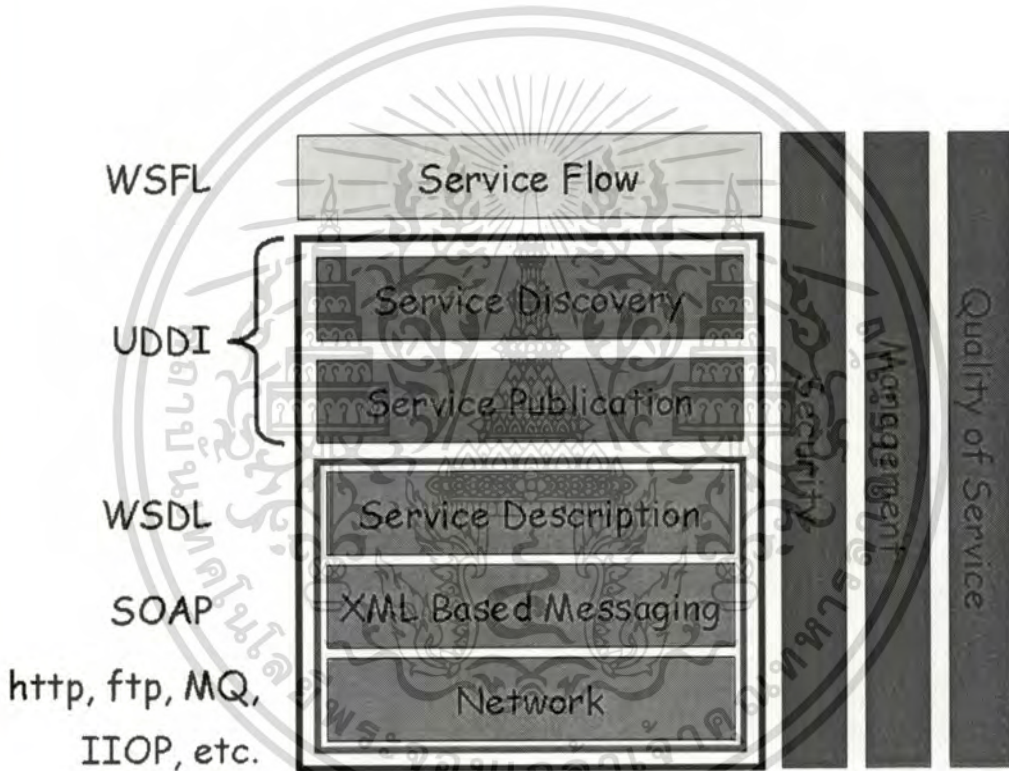
1. Publish เริ่มจากผู้ให้บริการสร้างเว็บเซอร์วิสเสร็จแล้วประกาศ (Publish) ไปยังผู้จดทะเบียนบริการ ซึ่งผู้จดทะเบียนบริการจะบันทึกเก็บไว้ในไดเรกทอรีของบริการ (Directory Service)
2. Find ผู้ที่ร้องขอบริการก็จะเข้ามาค้นหาบริการจากผู้จดทะเบียนบริการ
3. Bind คือสถานะที่ผู้ร้องขอบริการเชื่อมต่อกับผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเซอร์วิสจะทำให้การทำงานมีประโยชน์และมีประสิทธิภาพมากขึ้น โดยโปรแกรมของเว็บเซอร์วิสจะมีลักษณะหน่วยย่อยที่มีการกำหนดโครงสร้างไว้อย่างดี และซ่อนฟังก์ชันการทำงานภายในไว้และเปิดให้โปรแกรมอื่นเรียกใช้งานได้เฉพาะบางส่วน ซึ่งทำให้เกิดการทำงานร่วมกันระหว่างโปรแกรมหรือระบบที่เป็นไปในลักษณะต่อกันอย่างหลวมๆ (Loosely Coupled)

2.1.2 เว็บเซอร์วิสสแตค (Web Services Stack)

ในการทำงานทั้ง 3 นั้น (Publish, Find , Bind) อาศัยเว็บเซอร์วิสสแตค ซึ่งเป็นมาตรฐานในการติดต่อสื่อสารเว็บเซอร์วิส (Kreger.2001)



รูปที่ 2.2 Web Services Stack

- Layer Network เว็บเซอร์วิสจะถูกเรียกใช้บริการจากผู้ร้องขอ หรือ Requester ผ่านเครือข่ายโดยอาศัยอินเทอร์เน็ตโปรโตคอล อาทิ SMTP, FTP, MQ โดยมี HTTP เป็นโปรโตคอลมาตรฐาน
- Layer XML-Base Messaging เป็นการสื่อสารระหว่างเว็บเซอร์วิสและแอปพลิเคชันมีลักษณะ XML-Base Protocol ซึ่งโปรโตคอลที่ถูกนำมาใช้คือ SOAP (Simple Object Access Protocol) เพื่อแลกเปลี่ยนข้อมูลในสภาพแวดล้อมแบบกระจาย (Distribute

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกพันไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Layer Service Description เป็นส่วนที่อธิบายถึงรายละเอียด ลักษณะการให้บริการของเว็บเซอร์วิส โดยใช้มาตรฐาน WSDL (Web Service Description Language)
- Layer Service Publication & Discovery เป็นขั้นตอนในการประกาศและค้นหาเซอร์วิสที่มีในเว็บ โดยอาศัยมาตรฐาน UDDI (Universal Description, Discovery and Integration) เพื่อให้ผู้บริการใช้ในการเผยแพร่ข้อมูลการให้บริการเว็บเซอร์วิส และให้ผู้ขอใช้บริการใช้ค้นหาเว็บเซอร์วิสที่ต้องการ

เทคโนโลยีที่เกี่ยวข้องกับ SOA ในการ Implement เว็บเซอร์วิสมีดังนี้

- XML (eXtensible Markup Language)
- SOAP (Simple Object Access Protocol)
- WSDL (Web Service Description Language)
- UDDI (Universal Description, Discovery and Integration)

2.1.3 XML (eXtensible Markup Language)

XML หรือ eXtensible Markup Language เป็นภาษาที่เป็น text-based ซึ่งทำให้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ต XML เป็นฟอร์แมตที่อธิบายถึงรายละเอียดของโครงสร้างและแบบของข้อมูลเป็นภาษาหรือชุดคำสั่งมรการพัฒนาและมีศักยภาพในส่วนของโครงสร้างข้อมูล ด้วย XML จะทำให้การจัดการข้อมูลหรือเรียกใช้ข้อมูลจากแอปพลิเคชันต่างๆ เข้าสู่มาตรฐานเดียวกัน ผู้ที่ทำหน้าที่รับผิดชอบและกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้างเว็บเพจที่สามารถแสดงผลได้โดยโปรแกรมบราวเซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระแล้วทำการส่ง XML ชุดนี้ไปประมวลผลยังแอปพลิเคชันใดๆ ที่สามารถใช้ข้อมูล XML นั้น XML เป็นภาษาที่มีลักษณะเป็น tags คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อได้ ลักษณะโครงสร้างของเอกสาร XML แสดงได้ดังนี้

```
<?xmlversion="1.0" encoding="windows-874"?>
```

```
<root>
```

```
  <element>
```

```
    <tag></tag>
```

เอกสาร</element>สารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่</root>ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML Document Type Definition

DTD คือเพิ่มข้อมูลซึ่งบรรจุข้อกำหนด และกฎเกณฑ์ของเอกสาร ชุดข้อกำหนดเหล่านี้ใช้สำหรับกำหนดรูปแบบของ element ตัวอย่างเช่น หากต้องการเอกสารที่มี element <LIST> ที่มี element <ITEM> บรรจุอยู่ภายใน ข้อกำหนดในเพิ่มข้อมูล DTD จะมีรูปแบบดังนี้

```
<!ELEMENT item (#data)>
```

```
<!ELEMENT list (item)>
```

สามารถอธิบายได้ดังนี้คือ element items ใช้สำหรับบรรจุข้อความใดๆ และ element list บรรจุ element item อื่นๆ

เอกสาร XML มีส่วนประกอบที่สำคัญอยู่ 3 ส่วน ซึ่ง 2 ส่วน เป็นสิ่งที่จำเป็น ในขณะที่อีกส่วนเป็นทางเลือกที่จะมีหรือไม่มีก็ได้

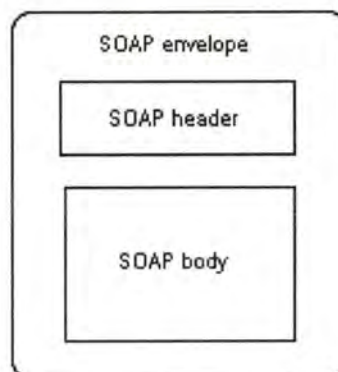
ส่วนที่ 1 คือส่วนเนื้อหา หรือ Content ซึ่งทำให้เอกสารมีข้อมูลสำหรับผู้อ่าน ส่วนของเนื้อหาสามารถเป็นได้ทั้งข้อความและรูปภาพ

ส่วนที่ 2 คือ กฎเกณฑ์และข้อกำหนดโครงสร้างของเอกสาร ในที่นี้คือไฟล์ DTD ซึ่งถือเป็นทางเลือกที่จะมีหรือไม่มีก็ได้

ส่วนที่ 3 คือ Style Sheet เป็นข้อกำหนดสำหรับการแสดงผลลัพธ์ในที่นี้คือไฟล์ XSL

2.1.4 SOAP (Simple Object Access Protocol)

SOAP เป็น XML-based โพรโทคอล และใช้ HTTP เป็นโพรโทคอลร่วมสำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจาย (Distributed Environment) โดยมีบริษัท ไมโครซอฟต์ ไอบีเอ็ม โสดีส และอีกกว่า 30 บริษัทเข้าร่วมและจัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น เพื่อให้ SOAP เป็น โพรโทคอลที่ไม่มีใครเป็นเจ้าของและเป็นโพรโทคอลที่ทำงานร่วมกับโพรโทคอลอื่นๆ ได้ ส่วนการพัฒนา ก็สามารถกระทำได้อย่างอิสระตามแพลตฟอร์มของระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object Model) และภาษาที่ใช้ในการเขียนโปรแกรมของผู้พัฒนา



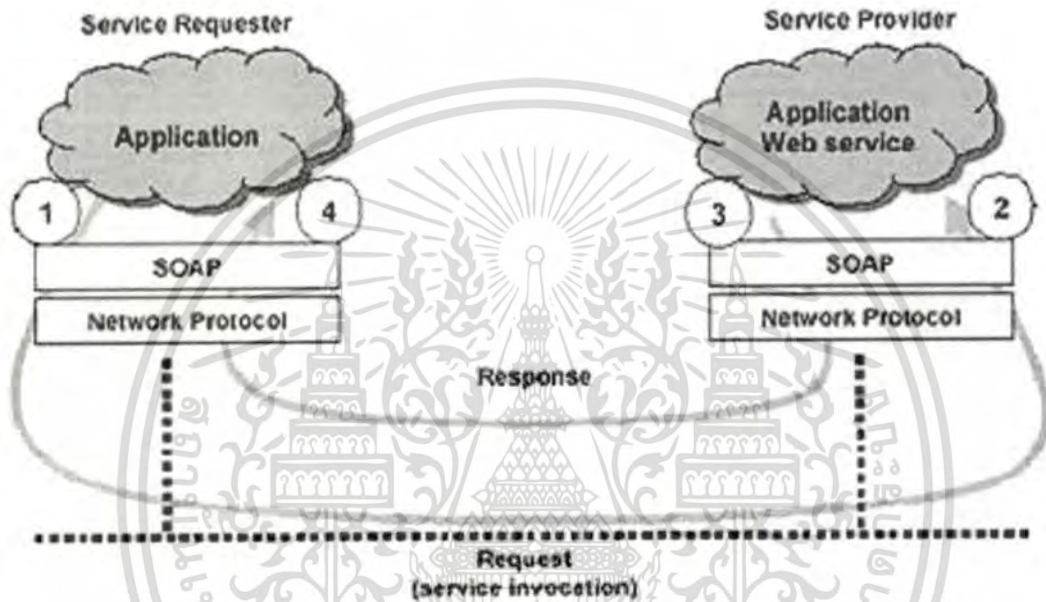
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแต่งแก้ไขเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 Soap Message

โครงสร้างของ SOAP

เอกสาร SOAP มีโครงสร้างในรูปแบบ XML ซึ่งสามารถแบ่งออกเป็น 3 ส่วนคือ

- SOAP Envelope เป็นส่วนที่ห่อหุ้มส่วนอื่นๆ ของเอกสารทั้งหมดเปรียบเสมือนซองจดหมาย
- SOAP Header เป็นส่วนที่เพิ่มเติมของเอกสาร (Extension) ซึ่งจะมีหรือไม่มีก็ได้
- SOAP Body เป็นส่วนเนื้อหา ที่ต้องการสื่อสารกันเปรียบเสมือนข้อความในจดหมาย



รูปที่ 2.4 ลักษณะการทำงาน Soap Message

ลักษณะการทำงานคือ

1. ผู้ร้องขอบริการจะสร้าง SOAP message และส่งผ่านเครือข่ายไปยังผู้ให้บริการ
2. ผู้ให้บริการได้รับ SOAP message
3. เว็บเซอร์วิสประมวลผลและส่งผลกลับในรูปแบบของ SOAP message

2.1.5 WSDL (Web Service Description Language)

เอกสาร WSDL เป็นข้อกำหนดที่สร้างขึ้นมาให้เป็นมาตรฐานเพื่อใช้อธิบายและประกาศถึงรูปแบบ คุณลักษณะ รายละเอียดของเว็บเซอร์วิส เช่น ชื่อของ Web Services ชื่อ URL ของ Web Services และฟังก์ชันการทำงานของ Web Services เอกสาร WSDL อยู่บนพื้นฐานของภาษา XML เช่นเดียวกับ SOAP โดย WSDL ถูกพัฒนาโดย ไมโครซอฟต์ อารีบา และ ไอบีเอ็ม ในเวอร์ชัน 1.1 และถูกประกาศให้เป็นมาตรฐานโดย W3C

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สงวนสิทธิ์ในเนื้อหาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ WSDL

WSDL เป็นภาษาที่อยู่ในการดูแลขององค์กร W3C (World Wide Web Consortium) ในเวอร์ชันที่มีอยู่ปัจจุบันคือ WSDL 1.1 เอกสาร WSDL มีส่วนประกอบที่สำคัญ ดังแสดงในตารางที่ 2.1 (W3.ORG.2001)

ตารางที่ 2.1 ส่วนประกอบเอกสาร WSDL

Element	Definition
<portType>	เป็นส่วนที่สำคัญที่สุดใน WSDL element อธิบาย operation ที่เว็บเซอร์วิสมีให้บริการ และ message ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียน โปรแกรม
<operation>	อธิบาย method ที่ให้บริการ เว็บเซอร์วิสหนึ่งจะมี method จำนวนกี่ method ก็ได้
<message>	อธิบาย data element ของ operation แต่ละ message อาจมีมากกว่าหนึ่ง message เทียบได้กับ parameter ของ function ในการเขียน โปรแกรม
<types>	อธิบายชนิดข้อมูลที่เว็บเซอร์วิสใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล
<binding>	อธิบาย format ของ message และ protocol details ในแต่ละ port
<service>	สำหรับเว็บเซอร์วิสจะมีเว็บเซอร์วิสจำนวนกี่บริการก็ได้ และชื่อเว็บเซอร์วิสก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีซ้ำกัน

2.1.6 UDDI (Universal Description, Discovery and Integration)

UDDI เปรียบเสมือนสมุดหน้าเหลืองที่ให้ข้อมูลว่า แต่ละธุรกิจให้บริการเว็บเซอร์วิสใดบ้าง หรือเปรียบเสมือนตัวแทนผู้ให้บริการ (Broker) UDDI ใช้สำหรับค้นหาเว็บเซอร์วิสที่ต้องการและเมื่อได้มาแล้ว UDDI ยังจัดหาข้อตกลงในวิธีการที่จะใช้งานเปรียบได้กับสมุดหน้าเหลือง เป็นมาตรฐานที่จัดตั้งขึ้น โดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟต์ และบริษัทอริบา ปัจจุบันมีบริษัทที่ร่วมกันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐาน UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B Interoperability

2.2 UML (Unified Modeling Language)

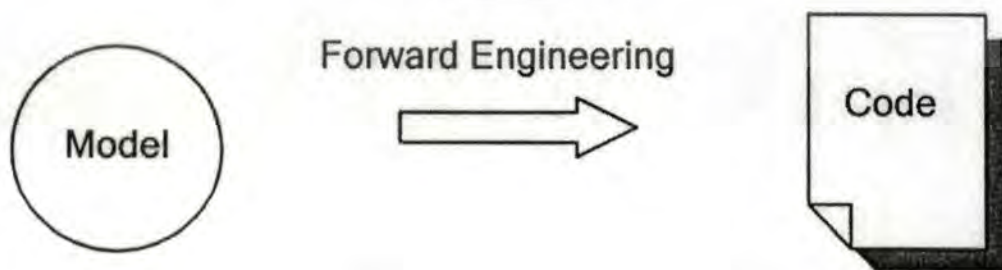
UML เป็นภาษาที่ใช้ในอธิบายถึงแบบจำลองโดยใช้ภาพสัญลักษณ์ในการอธิบายถึงระบบงาน เป็นมาตรฐานในการสร้างแบบพิมพ์เขียว (Blueprint) ให้กับระบบงาน โดย UML ช่วยสร้างมุมมองต่อระบบ การกำหนดรายละเอียด สร้างระบบงานและจัดทำเอกสารอ้างอิงแก่ระบบ

(สราวุธ อ้อยศรีสกุล, 2544)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

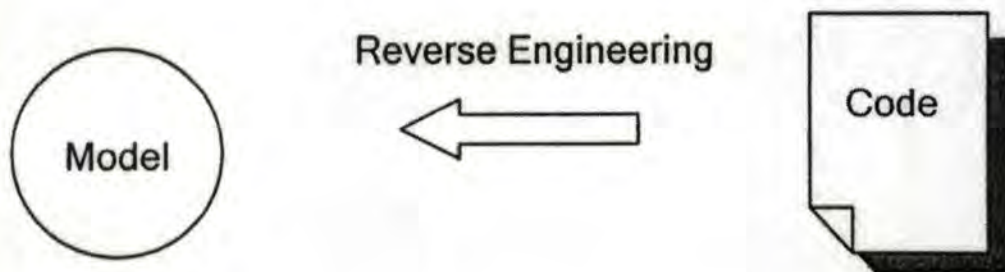
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ช่วยสร้างมุมมองต่างๆ ต่อระบบ ในการพัฒนาระบบนั้นย่อมมีบุคคลที่ต้องติดต่อ เกี่ยวข้องกับการพัฒนาระบบหลายฝ่ายด้วยกัน เช่น ผู้ใช้ (User) นักวิเคราะห์และแก็บระบบ (System Analyst) โปรแกรมเมอร์ (Programmer) แต่ละคนมีมุมมองต่อระบบที่ แตกต่างกันไป ทำให้เกิดช่องว่าง หรือความเข้าใจที่อาจไม่ถูกต้องตรงกัน เช่น นัก ออกแบบระบบอาจคิดแบบจำลองระบบขึ้นมาคร่าวๆ ในใจ แต่การอธิบายให้ผู้ฟัง อาจค่อนข้างลำบาก หรือมีความเข้าใจไม่ตรงกัน รายละเอียดบางอย่างไม่สามารถอธิบาย ได้ด้วยการดูโค้ดของโปรแกรม เช่น การอธิบายถึงระบบที่มีการสร้างคลาส และมีการ ถ่ายทอดคุณสมบัติให้แก่กัน (Inheritance)
- ช่วยกำหนดรายละเอียดของระบบ UML ช่วยในการสร้างแบบจำลองระบบให้มีความ ชัดเจน ถูกต้อง ไม่กำกวม โดยการมีภาพสัญลักษณ์ ซึ่งอธิบายเจตจำนงว่าอธิบายเจตจำนงนั้นๆ คือ อะไร สามารถใช้คำอธิบาย หรือหมายเหตุเพื่อช่วยอธิบายรายละเอียดเพิ่มเติมเพื่อความ เข้าใจที่ถูกต้องตรงกัน
- ช่วยในการสร้างระบบงานเพื่อใช้งานจริง นอกจาก UML จะใช้ในการอธิบายแบบจำลอง แล้ว UML ยังช่วยในการเขียนโค้ด โดยอัตโนมัติ กล่าวคือเราสามารถนำแบบจำลองจาก UML มาแปลงเป็นซอร์สโค้ดได้ โดยเรียกการเปลี่ยนแปลงจากแบบจำลองไปเป็น โค้ด โปรแกรมลักษณะเช่นนี้ว่า “Forward Engineering” ในทางกลับกันนอกจากจะแปลง แบบจำลองไปเป็น โค้ด โปรแกรม ยังมีวิธีในการแปลงจาก โค้ด โปรแกรมให้กลับมาเป็น ภาษา UML อีกด้วย โดยเรียกกระบวนการนี้ว่า “Reverse Engineering” ทั้งนี้ทั้งนั้น กระบวนการทั้งสองต้องขึ้นอยู่กับเครื่องมือ (Tool) ที่จะนำมาใช้ด้วยว่ามีคุณสมบัติในการ ทำ Forward หรือ Reverse Engineering หรือไม่



รูปที่ 2.5 กระบวนการ Forward Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 กระบวนการ Reverse Engineering

- ช่วยในการจัดทำเอกสารอ้างอิงแก่ระบบ เมื่อมีการพัฒนาระบบงาน ควรจะมีการจัดทำเอกสารอธิบายประกอบในขั้นตอนต่างๆ ของการพัฒนาเพื่อให้ผู้ใช้ระบบหรือผู้ที่ทำการพัฒนาต่อสามารถเข้าใจถึงขั้นตอนในการพัฒนาระบบได้อย่างถูกต้องเป็นระบบ

ประเภทของ UML Diagram

UML ประกอบไปด้วยไคอะแกรมต่างๆ หลายไคอะแกรมซึ่งใช้อธิบายแบบจำลองในแง่มุมต่างๆ กันไป ดังนี้

- Use Case Diagram แสดงถึงภาพรวมของระบบ ผู้ที่ปฏิสัมพันธ์กับระบบ และฟังก์ชันในระบบที่มีให้ใช้งาน แสดงความสัมพันธ์ระหว่าง Actor กับ Use case โดยมีองค์ประกอบสำคัญ 2 ส่วนคือ Actor หมายถึงผู้ที่มีปฏิสัมพันธ์กับระบบ และ Use case หมายถึงฟังก์ชันภายในระบบ
- Class Diagram แสดงถึงคลาสภายในระบบว่าในระบบหนึ่งๆ นั้นประกอบด้วยคลาสอะไรบ้าง และแต่ละคลาสมีความสัมพันธ์กันอย่างไร
- Sequence Diagram เป็นมุมมองที่แสดงความสัมพันธ์ระหว่างออบเจ็กต์ต่างๆ ว่ามีปฏิสัมพันธ์กันอย่างไรบ้าง โดยไคอะแกรมจะแสดงการติดต่อระหว่างออบเจ็กต์ในแนวนอนหรือแกน X ส่วนแนวตั้ง หรือแกน Y จะแทนช่วงเวลา
- Collaboration Diagram เป็นไคอะแกรมที่แสดงปฏิสัมพันธ์ระหว่างออบเจ็กต์เช่นเดียวกับ Sequence Diagram แต่ต่างกันตรงที่ Collaboration Diagram จะเน้นที่ข้อความ (Message) ที่สื่อสารกันระหว่างออบเจ็กต์
- State Chart Diagram แสดงการเปลี่ยนสถานภาพ (State) ของออบเจ็กต์ตั้งแต่เริ่มต้นจนถึงสิ้นสุดการเปลี่ยนแปลง ว่าในรอบหนึ่งๆ (Sequence) ออบเจ็กต์มีการเปลี่ยนแปลงไปอย่างไรบ้าง

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

- Component Diagram เป็นไดอะแกรมที่แสดงองค์ประกอบของระบบที่มีตัวตนจริงๆ (Component) สามารถจับต้องได้ เช่น ไฟล์ข้อมูล ไฟล์ไลบรารี และเอกสารการใช้งานระบบ เป็นต้น ซึ่งจะช่วยให้ผู้พัฒนาได้เห็นโครงสร้างที่ชัดเจนของระบบ เพื่อเป็นแนวในการพัฒนาให้ได้ระบบอย่างที่ต้องการ
- Deployment Diagram เป็นไดอะแกรมที่เกี่ยวข้องกับส่วนของฮาร์ดแวร์ ที่จะแสดงว่าการเซตอัพฮาร์ดแวร์ในขั้นตอนสุดท้ายจะเป็นอย่างไร

2.3 Java 2 Platform, Enterprise Edition (J2EE™)

Java 2 Platform, Enterprise Edition (J2EE™) คือ กลุ่มของเทคโนโลยีภาษาจาวาที่จำเป็นสำหรับการพัฒนาโปรแกรมประยุกต์ด้วยภาษาจาวาให้ใช้งานได้ในระดับองค์กร (นรินทร์ โอพารากิจอนันต์.2547:9)

แต่เดิมนั้นภาษาจาวาถูกใช้ในการสร้างโปรแกรมประยุกต์ส่วนบุคคลเป็นหลัก ต่อมาเมื่อจาวาถูกนำไปใช้ในองค์กรมากขึ้น ผู้พัฒนาจาวาจำเป็นต้องเพิ่มชุดคำสั่งใหม่ๆ เข้าไปเพื่อตอบสนองความต้องการใช้งานระดับองค์กร เมื่อชุดคำสั่งที่เพิ่มเข้าไปมีจำนวนมากขึ้นเรื่อยๆ จึงได้มีการจัดระเบียบชุดคำสั่งภาษาจาวาเสียใหม่ โดยแบ่งเทคโนโลยีภาษาจาวาออกเป็น 3 ระดับดังนี้

1. J2SE (Java 2 Standard Edition) หรือภาษาจาวาระดับมาตรฐาน คือภาษาจาวาดั้งเดิมที่มีแต่ชุดคำสั่งปกติสำหรับการสร้างโปรแกรมประยุกต์ส่วนบุคคล
2. J2EE (Java 2 Enterprise Edition) หรือ จาวาระดับองค์กร คือ จาวาที่มีชุดคำสั่งการสร้างโปรแกรมประยุกต์ให้ใช้งานได้ในระดับองค์กร
3. J2ME (Java 2 Micro Edition) หรือ จาวาระดับจิ๋ว คือ จาวาที่มีชุดคำสั่งเพื่อการสร้างโปรแกรมประยุกต์สำหรับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก

J2EE ถือได้ว่าเป็นส่วนขยายของ J2SE ด้วย กล่าวคือ J2EE รวมเอาทั้งหมดของ J2SE เข้าไปในตัวมันแล้วเพิ่มชุดคำสั่งที่จำเป็นสำหรับการพัฒนาโปรแกรมประยุกต์ให้ใช้งานได้ในระดับองค์กรเข้าไป

ตารางที่ 2.2 ชุดคำสั่งใน J2EE

ชื่อชุดคำสั่ง	หน้าที่และประโยชน์ใช้สอย
RMI-IIOP	ใช้สำหรับการสื่อสารหรือการเรียกเมธอดทางไกลผ่านระบบเครือข่าย
Java IDL	ภาษากลางสำหรับสื่อสารกับโปรแกรมประยุกต์ระดับองค์กรที่พัฒนาขึ้นด้วยเทคโนโลยีอื่น
JNDI	ระบบการเรียกชื่อของบริการหรือวัตถุที่มีอยู่บนระบบเครือข่าย
JDBC	ใช้เชื่อมการติดต่อกับฐานข้อมูล
JMS	ใช้ประสานงานหรือสื่อสารระหว่างโปรแกรมประยุกต์ด้วยกัน

ตารางที่ 2.2 (ต่อ)

Java Mail	ใช้ติดต่อกับเมลเซิร์ฟเวอร์ในองค์กร
Java Servlets	บริการเว็บบนฝั่งเซิร์ฟเวอร์
JSP	บริการเว็บบนฝั่งเซิร์ฟเวอร์ที่ดัดแปลงมาจาก Java Servlets อีกทีหนึ่ง โดยมีความง่ายในการพัฒนามากกว่า
JTA	ชุดคำสั่งสำหรับการบริหารจัดการธุรกรรม
JTS	บริหารธุรกรรม
EJB	โครงสร้างโปรแกรมประยุกต์แบบคอมโพเนนท์
JAXP	ชุดคำสั่งสำหรับจัดการไฟล์ XML
JAAS	ชุดคำสั่งสำหรับระบบรักษาความปลอดภัย
JCA	ใช้ติดต่อกับระบบอื่นๆ ที่มีอยู่ในเครือข่ายขององค์กร

2.3.1 Java Application Server

การนิยามคำว่า J2EE ขึ้นมาไม่ได้เป็นแค่การจัดกลุ่มชุดคำสั่งให้เป็นระบบเท่านั้น แต่ได้มีการสร้างนวัตกรรมของการพัฒนาโปรแกรมระดับองค์กรด้วย กล่าวคือมีการสร้างแนวคิดเรื่อง Application Server ขึ้นมา (นรินทร์ โอพารากิจนันต์.2547:9)

แต่เดิมการเขียนโปรแกรมประยุกต์ให้ใช้งานได้ระดับองค์กรเป็นเรื่องยากมาก เพราะนอกจากจะต้องเขียนคำสั่งให้โปรแกรมทำงานได้อย่างที่เราต้องการได้แล้ว โปรแกรมเมอร์ระดับองค์กรยังต้องเขียนคำสั่งให้โปรแกรมทำงานได้ในระบบเครือข่ายอีกด้วย ซึ่งเป็นเรื่องที่ยากมาก

แนวคิดเรื่องการพัฒนา Application Server ทำให้การพัฒนาโปรแกรมประยุกต์ระดับองค์กรทำได้ง่ายขึ้น เพราะ Application Server จะรับผิดชอบเกี่ยวกับการจัดการเครือข่ายทั้งหมดแทนเรา ทำให้โปรแกรมเราทำงานบนเครือข่ายคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ ถือได้ว่า Application Server เป็นหัวใจสำคัญของ J2EE

โครงสร้างของ Java Application Server

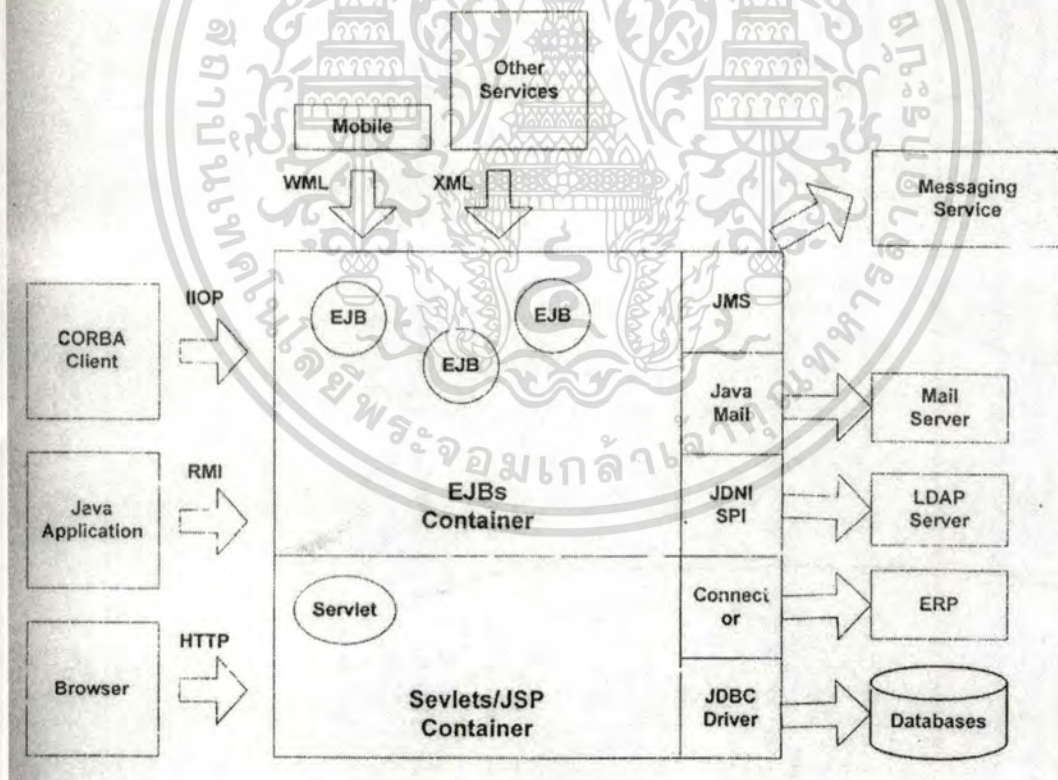
Java Application Server มีลักษณะคล้ายกับ Java Virtual Machine ในจาวาระดับมาตรฐาน กล่าวคือเป็นสิ่งที่โปรแกรมภาษาจาวาที่เราเขียนขึ้นทำงานบนตัวมันเองที่ อาจกล่าวได้ว่า Java Application Server เป็นตัวห่อหุ้มโปรแกรมภาษาจาวาระดับองค์กรที่เราเขียนเอาไว้จากสภาวะแวดล้อมภายนอก มันทำหน้าที่เป็นตัวกลางในการติดต่อสื่อสารกับภายนอกแทนโปรแกรมของเราแบบเดียวกับที่ Java Virtual Machine ที่คั่นกลางระหว่างโปรแกรมภาษาจาวาของเรากับระบบปฏิบัติการของเครื่อง

จาวาแอปพลิเคชันเซิร์ฟเวอร์ สามารถทำงานได้มากกว่า Java Virtual Machine เพราะในระดับองค์กร นอกจากโปรแกรมประยุกต์จะต้องติดต่อกับระบบปฏิบัติการแล้ว มันยังต้องติดต่อกับสิ่งอื่นอีกมากมาย ซึ่งส่วนมากเป็นการติดต่อผ่านทางระบบเครือข่ายขององค์กร เช่น บางครั้งโปรแกรมต้องดึงข้อมูลจากฐานข้อมูลที่อยู่บนเครื่องคอมพิวเตอร์อีกหนึ่งที่ห่างออกไปบน

เครือข่ายองค์กร เป็นต้น ตัวอย่างของผลิตภัณฑ์สำเร็จรูปที่ผลิตออกมา เช่น WebLogic ของ BEA, WebSphere ของ IBM, Sun Java System Application Server ของ SUN

คุณสมบัติเด่นของ Java Application Server

- สนับสนุนมาตรฐาน COBRA
- สนับสนุนแนวคิดของ Web Application
- ใช้หลักการของคอมโพเนนต์
- สนับสนุนแนวคิดของบริการผ่านเว็บ (Web Services)
- มีการบริการหน่วยความจำที่มีประสิทธิภาพ
- มีการจัดการด้านธุรกรรมข้อมูล
- มีระบบรักษาความปลอดภัยและระบบจัดการผู้ใช้



รูปที่ 2.7 โครงสร้างของ Java Application Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 Java Servlets

Java Servlets เป็นชุดคำสั่งของ J2EE ซึ่งใช้ในการสร้างโปรแกรมบนฝั่ง Server ที่ใช้ Browser เป็น Client เป็นการใช้ภาษาจาวาในการสร้างโปรแกรมที่มีลักษณะเป็น Web application นั้นเอง (นรินทร์ โอพารกิจอนันต์.2547:215)

โปรแกรมบนฝั่ง Server ที่เขียนด้วย Java Servlets เราเรียกว่า Servlet คำว่า Servlet มีความหมายว่า Server ขนาดเล็ก ซึ่งเป็นการตั้งชื่อให้สอดคล้องกับ Java Applet ซึ่งเป็นโปรแกรมภาษาจาวาขนาดเล็กที่ทำงานบนฝั่ง Client

Servlet ไม่ได้รันอยู่บน Java Virtual Machine แบบที่โปรแกรมภาษาจาวาทัวไปใช้ แต่มันรันอยู่บน Java Virtual Machine พิเศษที่เรียกว่า Servlet Container ซึ่งมีคุณสมบัติพิเศษคือทำหน้าที่เกี่ยวกับการติดต่อสื่อสารกับ Browser ให้กับ Servlet Servlet Container มีทั้งแบบที่สมบูรณ์ในตัวมันเองและแบบที่ต้องเกาะอยู่บน Web Server อีกที่ ตัวอย่างที่ได้รับความนิยมมากที่สุดคือ Tomcat

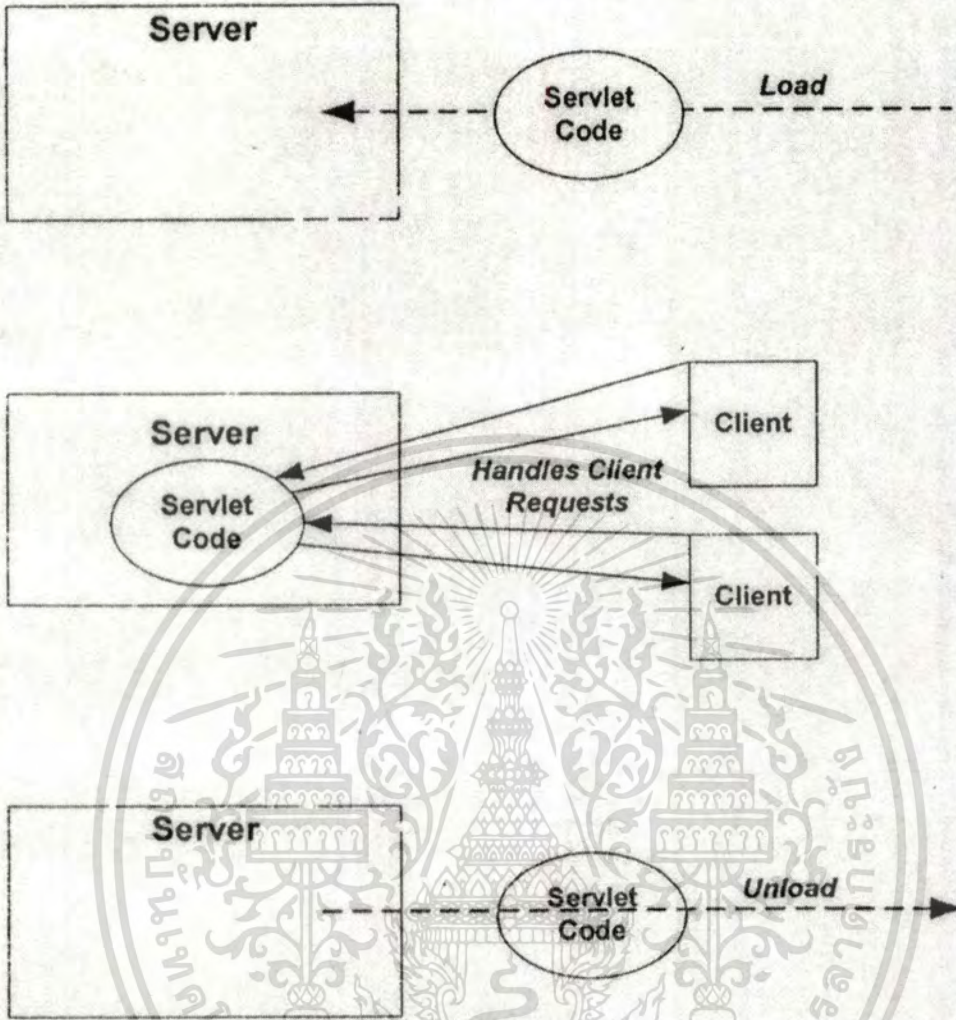
หน้าที่การทำงานของ Servlets

หลังจากที่เราติดตั้ง Servlet ไว้ใน Servlet Container แล้ว มันจะยังไม่ถูกโหลดเข้าไปในหน่วยความจำหลักของเครื่อง การโหลด Servlet ลงหน่วยความจำจะเกิดขึ้นก็ต่อเมื่อมี Browser เรียกว่า Servlet นั้นเป็นครั้งแรก และเมื่อมีการโหลด Servlet ลงหน่วยความจำแล้ว มันจะค้างอยู่ในหน่วยความจำตลอดเวลาที่ Servlet Container ยังทำงานอยู่ เวลาที่มี Browser เรียก Servlet นั้นซ้ำมันจะทำงานได้เลยไม่ต้องมีการโหลดใหม่ ซึ่งทำให้การเข้าถึง Servlet ครั้งต่อไปทำงานได้เร็วขึ้น

2.3.3 JavaServer Pages (JSP)

JavaServer Pages คือการใช้ภาษาจาวาในการสร้าง Web Application ที่มีเนื้อหาไม่ตายตัว โดยใช้ก๊อปปี้สำหรับแทรก Code ภาษาจาวาลงใน html page ซึ่งเมื่อตอนถูกเรียกใช้ page นั้นจะถูกเปลี่ยนไปเป็น Servlet ดังนั้น JSP จะสามารถรับ HTTP requests ได้เช่นเดียวกับ Servlet แต่การสร้าง JSP นั้นสามารถสร้างได้ง่ายกว่าการสร้าง Servlet (นรินทร์ โอพารกิจอนันต์.2547:235)

โดยทั่วไปเรามักให้ JSP ทำหน้าที่ presentation logic ส่วนงาน business logic ถูกส่งต่อไปให้ enterprise java beans ทำ



รูปที่ 2.8 วงจรชีวิตของ Servlets

2.3.4 Enterprise JavaBean (EJB)

Enterprise JavaBean (EJB) หรือ บีน คือโปรแกรมภาษาจาวาขนาดเล็ก (คอมโพเนนต์) บนเครื่อง Server ที่เขียนขึ้นมาเพื่อทำงานเฉพาะเจาะจงอย่างใดอย่างหนึ่ง บีนมีความเป็นอิสระในตัวเอง สามารถทำงานอยู่อย่างโดดเดี่ยวบน Server หรือทำงานประสานกับบีนด้วยกันเองก็ได้ (นรินทร์ โอพารากิจอนันต์.2547:28)

ประเภทของบีน

1. บีนบริการ (Session Bean)

ถูกสร้างขึ้นมาเพื่อทำงานบางอย่างให้กับผู้ใช้ฝั่ง Client ที่ติดต่อเข้ามายัง Server แบ่งเป็น

● บีนบริการแบบคงสถานะ (Stateful Bean) คืออินสแตนซ์ของบีนแบบนี้จะถูกสร้างขึ้นเพื่อจัดการกับผู้ใช้ที่เข้ามาแบบตัวต่อตัว สามารถจดจำสถานะต่างๆ ของผู้ใช้ได้

- บีนบริการแบบไม่คงสถานะ (Stateless Bean) คือบีนบริการที่ไม่มีการจดจำสถานะต่างๆ ของผู้ใช้ที่เข้ามา ดังนั้นในเวลาเดียวกันบีนบริการแบบไม่คงสถานะสามารถให้บริการผู้ใช้ได้มากกว่า 1 คน

2. บีนวัตถุ (Entity Bean)

บีนวัตถุ คือ บีนที่สามารถคงสถานะของบีนไว้ได้ตลอด แม้บีนจะถูกทำลายไปแล้ว

โดย Server จะอาศัยฐานข้อมูลในการเก็บสถานะของบีนเหล่านี้ แบ่งออกเป็น 2 แบบคือ

- บีนวัตถุแบบที่บีนเป็นผู้จัดการ (Bean-Managed Persistence) โดยโปรแกรมเมอร์จะต้องเป็นผู้เขียนคำสั่งติดต่อกับฐานข้อมูลเอง
 - บีนวัตถุแบบที่คอนเทนเนอร์เป็นผู้จัดการ (Container-Managed Entity Bean)
- ## 3. บีนที่ทำงานด้วยเมสเสจ (Message-Driven Bean) โดยจะทำงานก็ต่อเมื่อได้รับ Message จาก Message-Oriented-Middleware (ตัวกลางที่ช่วยในการทำงานให้คอมพิวเตอร์ต่างระบบสามารถคุยกันได้)

สำหรับบีนบริการและบีนวัตถุ ทุกๆ คอมโพเนนต์จะต้องประกอบด้วย

1. Home interface ใช้ประกาศเมธอดมาตรฐานที่ Client เรียกใช้เวลาติดต่อเข้ามา
2. Interface ใช้เป็นตัวกลางให้ Client เรียก Business ของบีน

2.4 Java API for XML

Java API for XML เป็น API ภาษาจาวาที่ใช้ในการติดต่อกับเอกสาร XML แบ่งออกเป็น 2 ประเภทคือ (Sun Microsystem. 2004)

- Document-Oriented คือ API ที่ใช้ในการจัดการโดยตรงกับเอกสาร XML เช่น JAXP JAXB และ SAAJ
- Procedure-Oriented คือ API ที่ใช้ในการจัดการกระบวนการงาน เช่น JAX-RPC และ JAXR

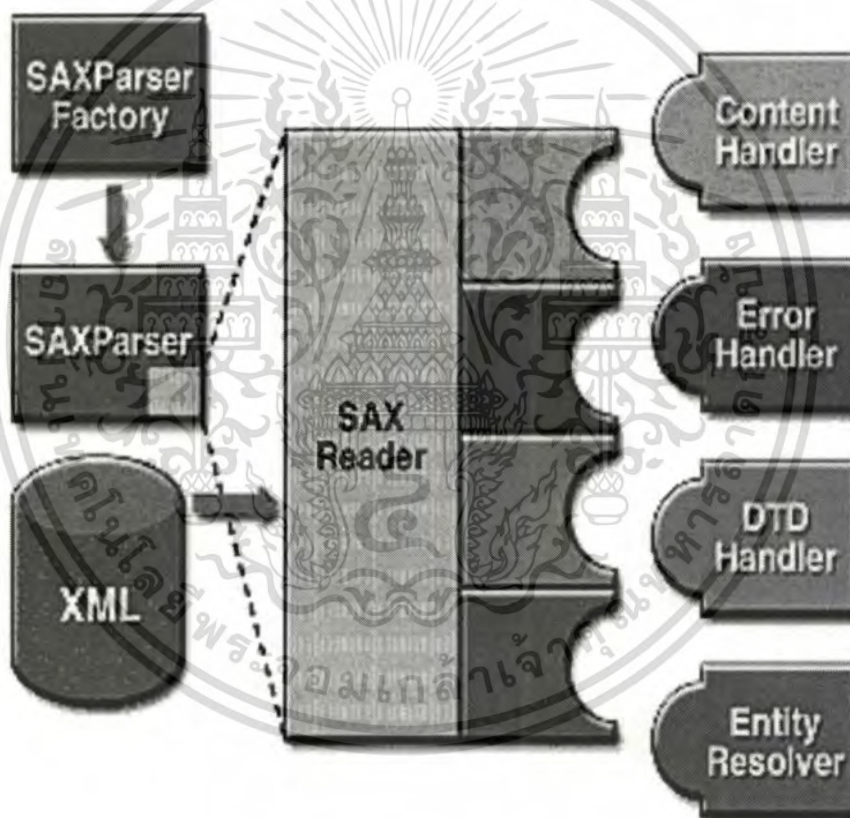
ลักษณะเด่นของ Java API for XML คือสามารถทำงานร่วมกับมาตรฐานอุตสาหกรรม ซึ่งได้ถูกกำหนดขึ้นมาก่อนแล้วโดยกลุ่มหน่วยงานที่กำหนดมาตรฐานต่างๆ เช่น World Wide Web Consortium (W3C) และ Organization for the Advancement of Structure Information Standards (OASIS) นอกจากนี้แล้ว Java API for XML จะมีความยืดหยุ่นในการใช้งาน เช่น โค้ด JAXP สามารถถูกเรียกใช้งานได้หลากหลายจากเครื่องมือประมวลผล XML ซึ่งทำให้ผู้พัฒนาสามารถเขียนโปรแกรมได้เหมาะสมพอดีกับระบบของตน

2.4.1 JAXP

Java API for XML Processing เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาให้ทำงานประมวลผลข้อมูลของเอกสาร XML ได้ โดยสามารถใช้งานร่วมกับ SAX DOM และ XSLT ดังนี้

- SAX APIs

เริ่มจากการสร้าง SAXParser Object จาก SAXParserFactory Object ซึ่งตัว SAXParser Object นี้จะห่อหุ้ม SAXReader Object ไว้ภายใน และเมื่อเรียก SAXParser ทำการเรียกเมธอด parse() ก็จะทำให้ SAXReader Object ไปเรียกเมธอดที่ได้ทำการอิมพลิเมนต์ไว้ ซึ่งเมธอดต่าง ๆ นั้นอิมพลิเมนต์ขึ้นจาก ContentHandler Interface, ErrorHandler Interface, DTDHandler Interface และ EntityResolver Interface ดังรูปที่ 2.9

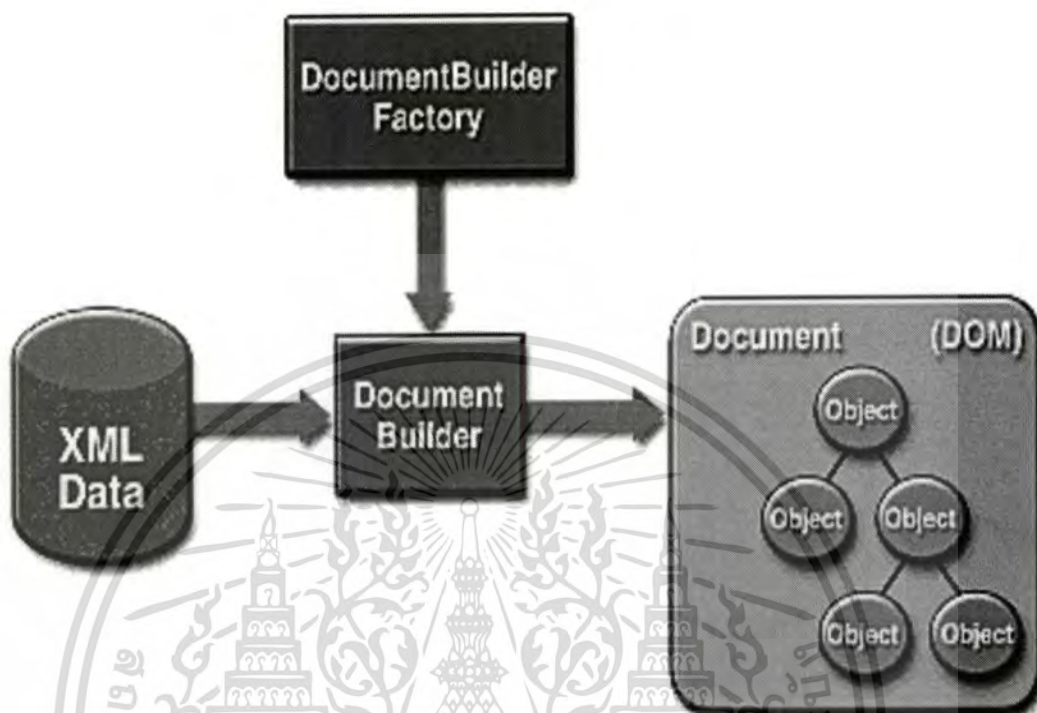


รูปที่ 2.9 โครงสร้างของ SAX APIs

- DOM APIs

เริ่มจากการใช้ DocumentBuilderFactory Object ไป get DocumentBuilder Object ออกมาเพื่อที่จะไปใช้ในการสร้างเอกสารที่สอดคล้องกับข้อกำหนดของ DOM (คือการนำ DocumentBuilder มาสร้างเอกสารใหม่ที่ทำการอิมพลิเมนต์ org.w3c.dom.Document Interface นั้นเอง) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นก็เรียกเมธอด `parse()` เพื่อสร้างเอกสาร โดยใช้ข้อมูล XML ที่มีอยู่ ผลที่ได้ คือ DOM Tree ทั้งนี้ DOM APIs ดีกว่า SAX APIs ตรงที่เราสามารถแก้ไขและเพิ่ม XML Node ได้

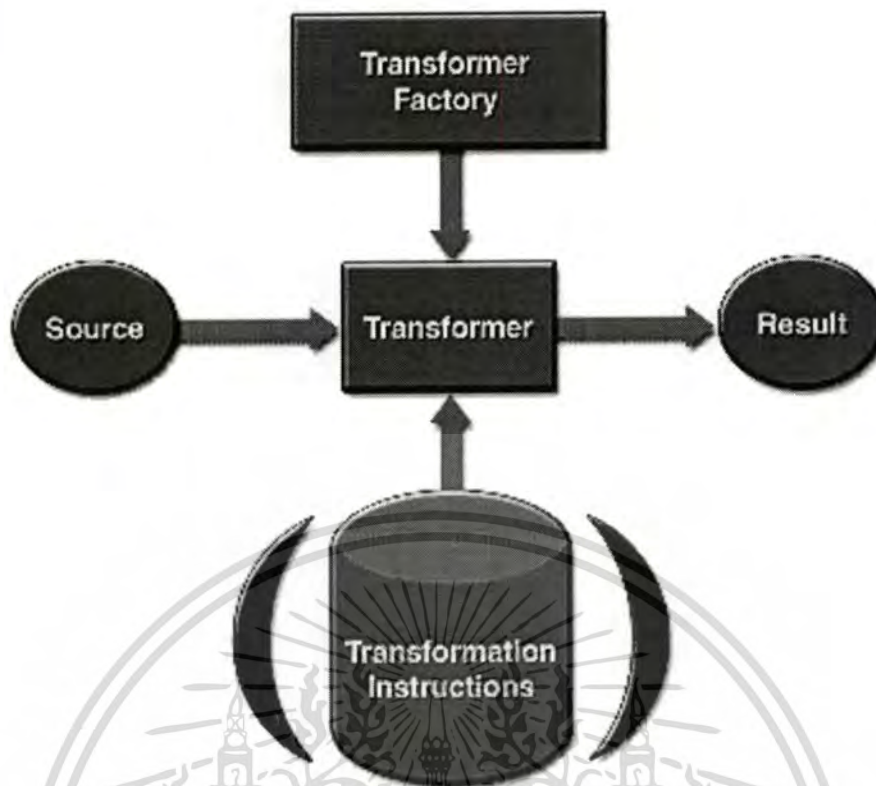


รูปที่ 2.10 โครงสร้างของ DOM APIs

- XSLT APIs

Extensible Stylesheet Language for Transformation หรือ XSLT เป็นภาษาที่ใช้ในการแปลงเอกสาร XML ให้เป็นรูปแบบอื่นๆ การทำงานของ XSLT APIs ของภาษาจาวาเริ่มจากการสร้าง Transformer Object จาก TransformationFactory Object และรับอินพุตเข้ามา ซึ่ง Input นี้ อาจจะมาจากการอ่านของ SAX หรือ DOM ก็ได้ รวมทั้งอาจเป็น Inputstream ก็ได้เช่นกัน เมื่อ Transformer Object เกิดขึ้นมานั้น ก็จะมีการสร้าง Transformer Instructions ด้วย เพื่อใช้ในการทำงานสร้างเป็นเอสต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 โครงสร้างของ XSLT APIs

2.4.2 JAXB

Java APIs for XML Binding เป็น APIs ที่ใช้ในการเขียนโปรแกรมภาษาจาวาเพื่อทำการขีดเหนี่ยว (binding) ซึ่งก็คือการสร้างจาวาคลาสจากเค้าร่าง XML นั้นเอง โดยการเรียกใช้ เมธอด Unmarshalling และเมธอด Marshalling ซึ่งเมธอด Unmarshalling คือการเฝ้า จากเอกสาร XML ไปเป็นวัตถุซึ่งมีลักษณะเป็น Content Tree ส่วนเมธอด Marshalling ก็คือการเฝ้า จากวัตถุ Java Content Tree ไปเป็นเอกสาร XML สำหรับการทำงานของ XSLT APIs นี้จะทำงานโดยซ่อนรายละเอียดการติดต่อกับ SAX และ DOM ทำให้ผู้พัฒนาได้รับความสะดวกเป็นอย่างมาก

2.4.3 SAAJ

SOAP with Attachments API for Java เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการส่งข้อความ SOAP ไปในเครือข่ายอินเทอร์เน็ต ซึ่ง SAAJ นี้เป็นการทำงานเบื้องหลังของการอิมพลีเมนต์ JAX-RPC และ JAXR โดยทำให้ผู้พัฒนาสามารถเขียนโปรแกรมเพื่อส่งข้อความ SOAP ได้โดยตรง โดยไม่ต้องผ่านกระบวนการ JAX-RPC โดย SAAJ APIs นี้สอดคล้องกับข้อกำหนดของ SOAP เวอร์ชัน 1.1

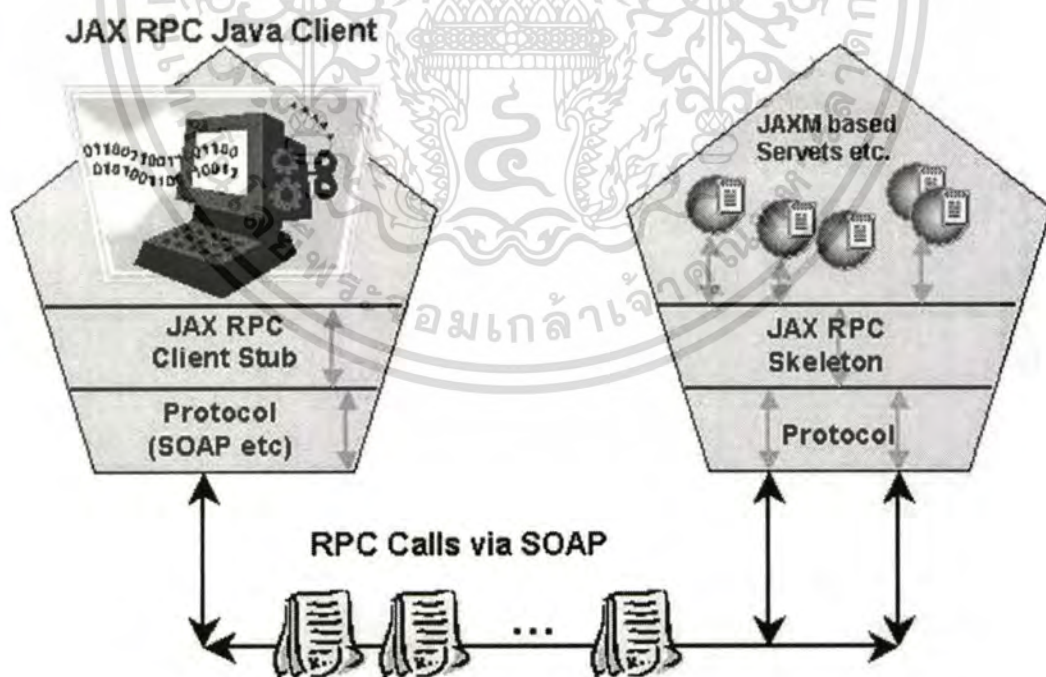
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดของ SAAJ เวอร์ชัน 1.2 นี้ได้กำหนดไว้ในแพ็คเกจ javax.xml.soap ซึ่งประกอบด้วย APIs ที่ใช้ในการสร้างและนำข้อมูลใส่ข้อความ SOAP และการรับ-ส่ง สำหรับการอิมพลิเมนต์นั้นมีคลาสที่ต้องใช้บ่อยๆ 2 ตัว คือ SOAPConnection กับ SOAPMessage

2.4.4 JAX-RPC

Java API for XML-based RPC เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการเรียกใช้งานเว็บเซอร์วิส โดยเป็นการเรียกใช้งานกระบวนการระยะไกล (Remote Procedure Calling) โดย JAX-RPC นี้เป็นส่วนที่สำคัญหลักของการพัฒนาจาวาเว็บเซอร์วิสเนื่องจากเป็นตัวสำคัญที่ทำให้เกิดการทำงานร่วมกันข้ามระบบ (Interoperability) ในการทำงานข้ามแพลตฟอร์ม โดยอาศัยพื้นฐานของ XML WSDL และ SOAP

โดยปกติแล้ว การพัฒนาโปรแกรมเพื่อเรียกใช้งาน Procedure ข้ามเครื่องจะมีขั้นตอนที่ยุ่งยากซับซ้อน แต่ JAX-RPC ได้ทำการอิมพลิเมนต์ ขั้นตอนเหล่านั้นไว้แล้ว เช่น การทำ Marshal/Unmarshal การสร้างคลาสในระดับล่างสำหรับไคลเอนต์ และเซิร์ฟเวอร์ (Stubs และ Ties) การสร้างหรือตีความ SOAPMessage (SAAJ เองก็เป็นการทำงานเบื้องหลังของ JAX-RPC เช่นกัน) marshal



รูปที่ 2.12 การทำงานของ JAX-RPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างเว็บเซอร์วิสที่สามารถเรียกใช้งานผ่าน RPC ได้นั้นมี 2 ขั้นตอนคือสร้างส่วนต่อประสาน และสร้างคลาสที่อิมพลิเมนต์ส่วนต่อประสาน นั้นขึ้นมา และคลาสตัวนี้ก็ถูกไคลเอนต์มาเรียกใช้เมธอดของมันในภายหลัง ซึ่ง JAX-RPC ไม่ได้บังคับว่าไคลเอนต์ที่ติดต่อเข้ามาเรียกใช้เว็บเซอร์วิสดังกล่าว ต้องรันอยู่บนแพลตฟอร์มของจาวาเท่านั้น

ในการทำงานร่วมกับ XML และ WSDL นั้น ในการทำงานเบื้องหลังของ JAX-RPC จะมีการแม็ประหว่างแบบชนิดข้อมูลของ XML กับ แบบชนิดข้อมูลของภาษาจาวา ซึ่งผู้พัฒนาไม่จำเป็นต้องรู้วิธีการแม็ปนั้น เพียงแต่ต้องระวังว่าแบบชนิดข้อมูลประเภทใดบ้างในภาษาจาวา ที่สามารถใช้เป็นพารามิเตอร์และค่าที่คืนกลับมาของเมธอดที่สนับสนุนการทำงานของ JAX-RPC โดยแบบชนิดข้อมูลที่สนับสนุนมีดังนี้

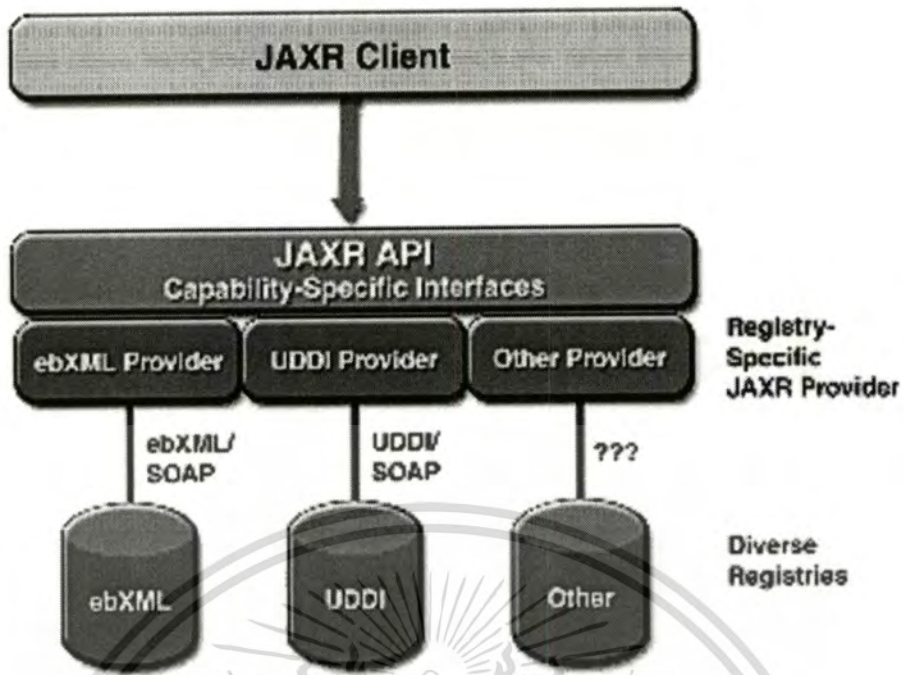
- J2SE SDK Classes
- Primitives
- Arrays
- Value Types
- JavaBeans Components

2.4.5 JAXR

Java API for XML Registries เป็น API ที่ใช้สำหรับเขียนโปรแกรมภาษาจาวาเพื่อทำการติดต่อกับ Registries Server เช่นการเขียนโปรแกรมเพื่อ Publish บริการตัวเองไปยัง Registries Server หรือการเขียนโปรแกรมเพื่อค้นหาบริการของคนอื่น ตัวอย่างของ Registries Server เช่น UDDI และ ebXML Registries

สถาปัตยกรรมของ JAXR ประกอบด้วย

- JAXR Client เป็นโปรแกรมฝั่งไคลเอนต์ที่มีการเรียกใช้งาน Registries โดยผ่านทาง JAXR Provider (โดยโปรแกรมฝั่งไคลเอนต์นั้นมีการเรียกใช้ JAXR)
- JAXR Provider เป็นการอิมพลิเมนต์ JAXR API เพื่อเรียกใช้งาน JAXR Provider หรือกลุ่มของ JAXR Provider ที่เป็นไปตามมาตรฐานใดมาตรฐานหนึ่ง



รูปที่ 2.13 สถาปัตยกรรม JAXR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 ความต้องการของระบบ

เนื่องจากในปัจจุบันการซื้อตั๋วรถประจำทางมีข้อเสียคือต้องเสียเวลาในการเข้าคิว และถ้าหากที่นั่งเต็มลูกค้าต้องเสียเวลารอค้นต่อไปเรื่อยๆ ซึ่งอาจจะไม่ใช่คั่นที่เราต้อง การการซื้อออนไลน์ผ่านเว็บบราวเซอร์ลูกค้าสามารถเลือกที่นั่งต่างๆ ได้ตามต้องการ โดยระบบจะมีลักษณะดังนี้

- สามารถเลือกจองตั๋วรถประจำทางได้ทุกคั่นทุกรอบทุกที่นั่งเหมือนกับเดินไปซื้อเองที่เคาเตอร์จองตั๋ว
- สามารถจองตั๋วล่วงหน้าได้
- สามารถดูรายละเอียดเกี่ยวกับรถประจำทาง
- สามารถแก้ไข เปลี่ยนแปลง ยกเลิกการจองได้
- สามารถชำระค่าบริการได้ผ่านทางธนาคาร

3.2 การออกแบบระบบ

ในการออกแบบระบบของตั๋วรถประจำทางผ่านอินเทอร์เน็ตสามารถแบ่งออกได้ 2 ระบบย่อยๆ ดังนี้

1. ระบบของฝั่งเว็บแอปพลิเคชัน (web application) ใช้เป็นอินเทอร์เน็ตเฟสระหว่างผู้ใช้กับระบบ ซึ่งภายในระบบใช้สำหรับการประมวลผลการทำงานด้านต่างๆ และเก็บข้อมูลของระบบ
2. ระบบของเว็บเซิร์ฟเวอร์ ใช้ในการประมวลผลด้านการเงินเพื่อใช้สำหรับติดต่อกับธนาคาร ซึ่งประกอบ ด้วย ส่วนการตรวจสอบยอดหรือเครดิตของผู้ใช้ระบบ ทำการหักบัญชีเมื่อมีการจองตั๋ว และคืนเงินเมื่อมีการยกเลิกการจอง

รายละเอียดการทำงานของระบบสามารถอธิบายได้ดังนี้

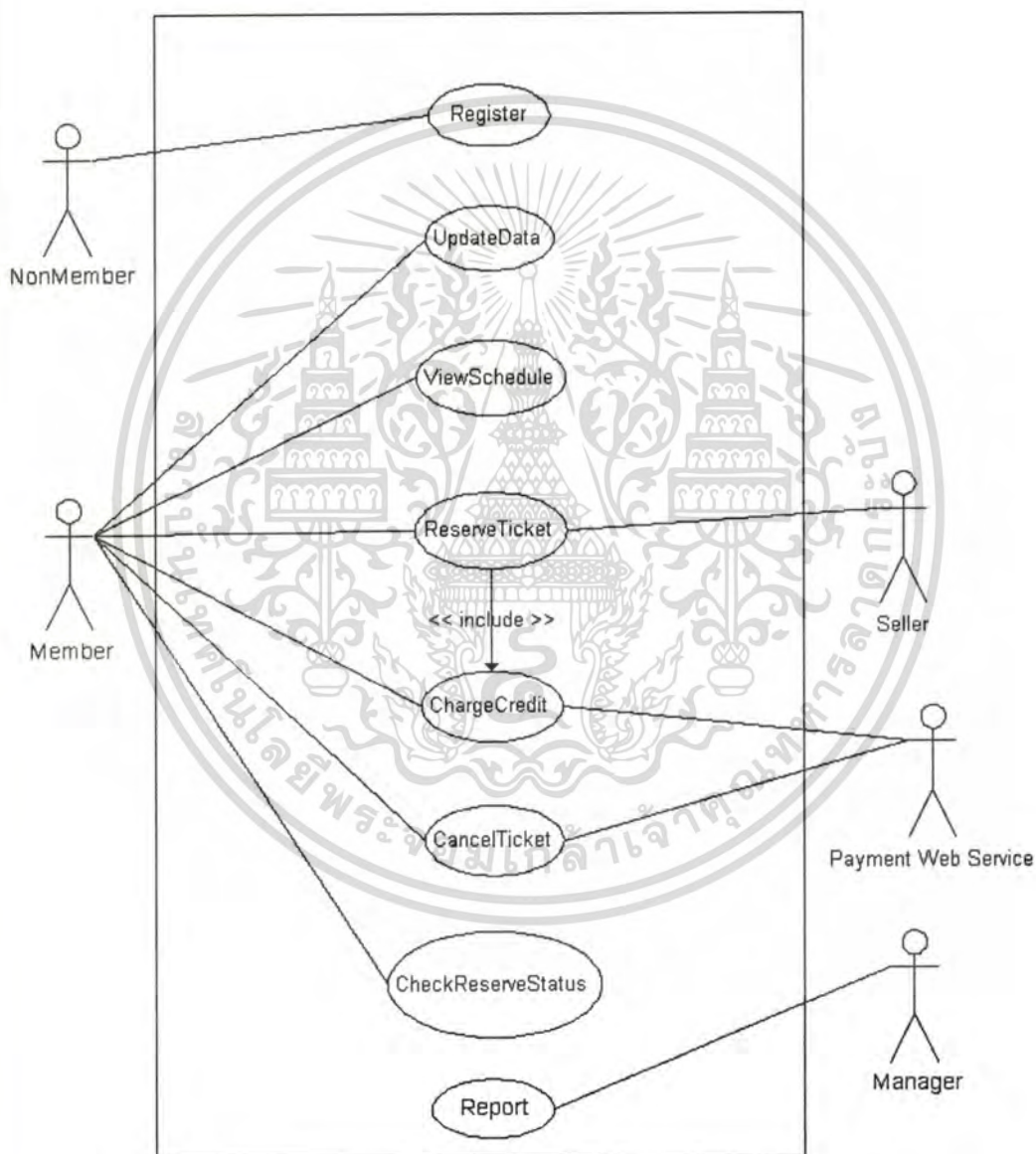
1. เมื่อลูกค้าต้องการจองตั๋วรถประจำทางผ่านอินเทอร์เน็ตจะโดยใช้คอมพิวเตอร์เชื่อมต่อเข้ามายังระบบเพื่อดูข้อมูลของรอบการเดินทางต่างๆ ของรถโดยสาร เมื่อเลือกรอบการเดินทางและที่นั่ง ได้แล้วที่ต้องการได้แล้วระบบจะส่งรายการทั้งหมดกลับ โดยจะระบุตำแหน่งที่ถูกซื้อไว้แล้ว ตำแหน่งที่ถูกคนอื่นจองอยู่ ตำแหน่งที่เราเลือกไว้ และตำแหน่งที่ว่างทั้งหมด เมื่อเลือกที่นั่ง ได้แล้วกดตกลงระบบจะถามบัญชีธนาคารหรือหมายเลขบัตรเครดิต วันที่หมดอายุ รหัสหลังบัตร และเบอร์โทรศัพท์มือถือ

เอกสารนี้เป็นเอกสาร 2. ที่ระบบทำการตรวจสอบยอดเงิน เมื่อระบบตรวจสอบเรียบร้อยแล้วจะส่งการคำนวณว่ากรณีใดๆทั้งสิ้น หมายเลขรายการและเอกสารที่ใช้ในการพิมพ์ตั๋วผ่านทางแอปพลิเคชันที่มีการนำไปใช้

3.3 การออกแบบเชิงวัตถุโดยใช้ UML

การออกแบบระบบของตัวรถประจำทางผ่านอินเทอร์เน็ตใช้ UML เป็นโมเดลในการอธิบายการทำงาน ซึ่งมองสิ่งต่างๆ ในระบบเป็นวัตถุ จากการวิเคราะห์ระบบสามารถยูสเคสไดอะแกรมที่ใช้แสดงถึงภาพรวมของระบบ ที่จะเกิดขึ้นภายในระบบว่า Actor และ Use Case ใดบ้างที่มีความสัมพันธ์กัน

3.3.1 Use Case Diagram

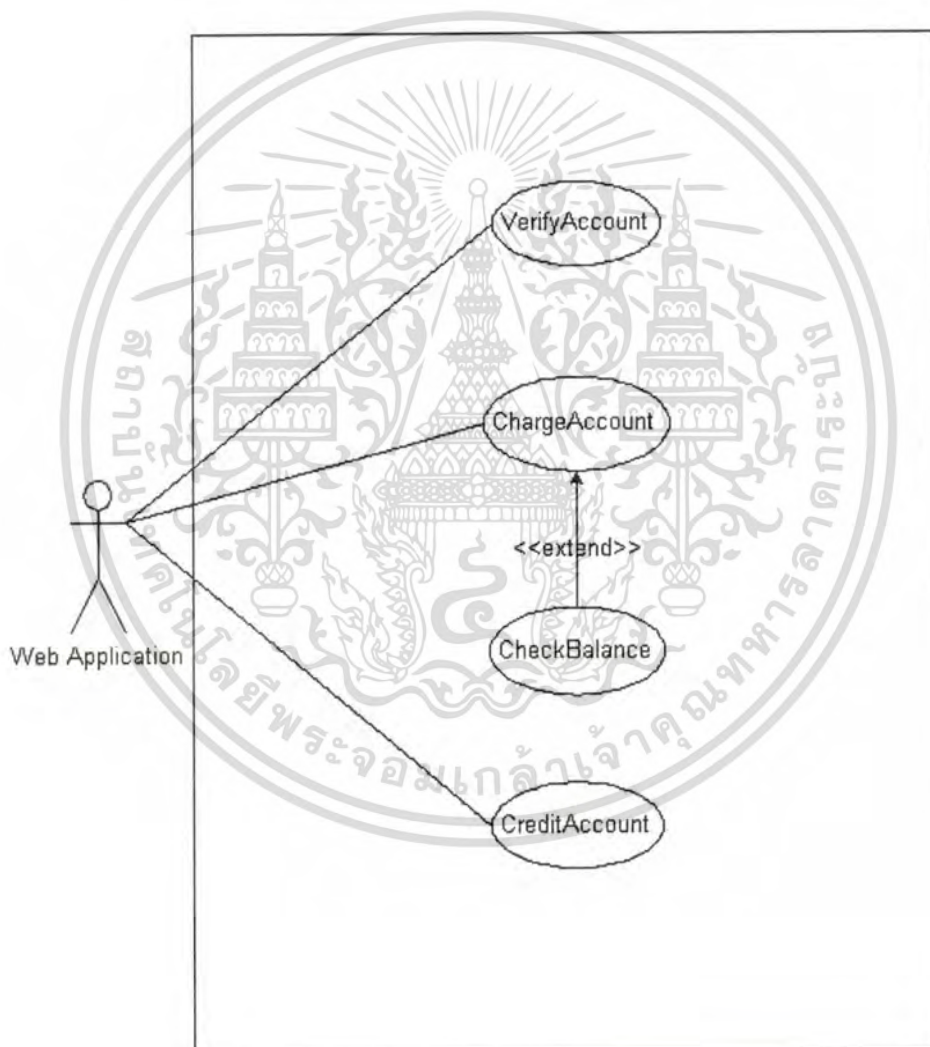


รูปที่ 3.1 ยูสเคสไดอะแกรมของระบบฝั่งเว็บแอปพลิเคชัน

โดยจะแสดงถึงการทำงานและภาพรวมของระบบทั้งฝั่งแม่ข่ายและลูกข่าย โดยมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่ Register ใช้สำหรับผู้ที่สนใจเข้าเป็นสมาชิกเข้าลงทะเบียนระบบเพื่อขอใช้บริการ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. UpdateData ใช้สำหรับแสดงรายละเอียดของข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูล
ผู้ใช้บริการ
3. ViewSchedule ใช้สำหรับแสดงรายละเอียดข้อมูลของตารางการเดินทาง
4. ReserveTicket ใช้สำหรับจองตั๋วรถโดยสาร
5. CancelTicket ใช้สำหรับยกเลิกการจองตั๋วรถโดยสาร
6. CheckReserveStatus ใช้ตรวจสอบสถานะของการจองตั๋ว
7. ChargeCredit ใช้สำหรับตรวจสอบและตัดเงินออกจากบัญชี
8. Report ใช้สำหรับแสดงผลรายงานต่างๆ



รูปที่ 3.2 ยูสเคสโคอะแกรมของฝั่งเว็บเซอร์วิสใช้ในการตัดเงินและคืนเงินให้กับ
ผู้ใช้บริการ

ยูสเคสโคอะแกรมสำหรับการหักเงินเวลาซื้อตั๋วและการคืนเงินในกรณียกเลิกการเดินทาง โดยจะ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ใช้เว็บเซอร์วิส โดยมียูสเคสดังต่อไปนี้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. VerifyAccount ใช้สำหรับตรวจสอบว่าหมายเลขบัญชีนั้นถูกต้องหรือไม่
2. ChargeAccount ใช้สำหรับหักยอดเงินในบัญชี ในกรณีที่มีการซื้อตั๋วรถประจำทาง
3. CreditAccount ใช้สำหรับคืนยอดเงินในบัญชีในกรณีที่มีการยกเลิกตั๋วรถประจำทาง
4. CheckBalance ใช้สำหรับตรวจสอบว่ามียอดเงินคงเหลือในบัญชีเพียงพอกับค่าตั๋วหรือไม่

3.3.2 Use Case Description

ตารางที่ 3.1 แสดงคำอธิบายยูสเคสของการสมัครสมาชิกระบบเพื่อเข้าใช้บริการ

Name:	Register
Brief Description:	เป็นยูสเคสใช้สำหรับผู้ที่สนใจเข้าเป็นสมาชิก
Actor:	NonMember
Precondition:	-
Basic Flows:	<ol style="list-style-type: none"> 1) ระบบแสดงแบบฟอร์มเพื่อกรอกรายละเอียด 2) ผู้ใช้บริการกรอกข้อมูล 3) ระบบตรวจสอบเพื่อไม่ให้ชื่อ user ซ้ำ 4) ผู้ใช้บริการกดปุ่มสมัครสมาชิก
Alternative Flows:	1a) NonMember ไม่ทำการกรอกและจบการทำงานของ Use Case นี้
Post Conditions:	ระบบทำการสร้างสมาชิก

ตารางที่ 3.2 แสดงคำอธิบายยูสเคสแสดงรายละเอียดข้อมูลเพื่อทำการเปลี่ยนแปลงข้อมูล
ผู้ให้บริการ

Name:	UpdateData
Brief Description:	เป็นยูสเคสสำหรับให้สมาชิกทำการเปลี่ยนแปลงข้อมูลสมาชิก
Actor:	Member
Precondition:	ต้องผ่านการสมัครสมาชิกมาเรียบร้อยแล้ว

ตารางที่ 3.2 (ต่อ)

Basic Flows:	1) ระบบแสดงแบบฟอร์มเพื่อกรอกรายละเอียด 2) ผู้ใช้บริการเปลี่ยนแปลงข้อมูล 3) ผู้ใช้ตรวจสอบความถูกต้องของรายละเอียดที่กรอก 4) ผู้ใช้บริการคณุ่มแก้ไขข้อมูลสมาชิก
Alternative Flows:	1a) Member ไม่ทำการเปลี่ยนแปลงและจบการทำงานของ Use Case นี้
Post Conditions:	ระบบทำการเปลี่ยนแปลงข้อมูล

ตารางที่ 3.3 แสดงคำอธิบายยูสเคสที่ใช้แสดงรายละเอียดข้อมูลของตารางการเดินรถ

Name:	ViewSchedule
Brief Description:	เป็นยูสเคสใช้สำหรับแสดงรายละเอียดข้อมูลของตารางการเดินรถ
Actor:	Member
Precondition:	ต้องผ่านการสมัครสมาชิกมาเรียบร้อยแล้ว
Basic Flows:	1) ผู้ใช้เลือกเส้นทางและวันเดินทาง 2) ระบบแสดงรายละเอียดต่างๆ ของเที่ยวรถ
Alternative Flows:	1a) ผู้ใช้เลือกเส้นทางและวันเดินทางอีกครั้ง 1b) ไม่มีข้อมูลตามวันที่ที่แจ้งระบบแสดงข้อความเตือนไม่มีข้อมูลในวันที่ดังกล่าวในฐานข้อมูล
Post Conditions:	-

ตารางที่ 3.4 แสดงคำอธิบายยูสเคสของการจองตั๋วโดยสาร

Name:	ReserveTicket
-------	---------------

ตารางที่ 3.4 (ต่อ)

Brief Description:	เป็นยูสเคสสำหรับให้สมาชิกทำการจองตั๋วโดยสารโดยมีการตรวจสอบยอดเงินสมาชิกก่อน
Actor:	Member
Precondition:	ต้องผ่านการเลือกเที่ยวการเดินทางมาก่อน
Basic Flows:	<ol style="list-style-type: none"> 1) แสดงรายละเอียดเที่ยวการเดินทาง 2) สมาชิกเลือกเที่ยวการเดินทางจากตารางเที่ยวการเดินทาง 3) สมาชิกเลือกเมนูจองตั๋ว 4) ระบบทำการเรียกใช้ Use Case "Check Balance" 5) เว็บเซอร์วิสทำการตรวจสอบยอดคงเหลือจากธนาคารที่ให้บริการ 6) เว็บเซอร์วิสส่งผลตอบกลับเพื่อยืนยันว่าสมาชิกนั้นมียอดเงินคงเหลือเพียงพอ 7) สมาชิกเลือกที่นั่ง 8) สมาชิกยืนยันการจอง 9) ระบบทำการติดต่อเว็บเซอร์วิสเพื่อติดต่อกับธนาคาร 10) เว็บเซอร์วิสติดต่อกับธนาคารเพื่อทำการหักเงินของสมาชิก 11) ระบบบันทึกข้อมูลการจองตั๋ว
Alternative Flows:	<ol style="list-style-type: none"> 3a) สมาชิกไม่ทำการจองและจบการทำงานของ Use Case นี้ 6a) ยอดเงินคงเหลือไม่เพียงพอสำหรับการจอง ระบบจะแจ้ง error message และกลับไปทำข้อ 2 8a) สมาชิกไม่ยืนยันการจองจบการทำงานของ Use Case นี้
Post Conditions:	ระบบทำการสร้างตั๋วการเดินทางและเปลี่ยนสถานะเป็น Yes

ตารางที่ 3.5 แสดงคำอธิบายยูสเคสยกเลิกการจองตั๋วโดยสาร

Name:	CancelTicket
Brief Description:	เป็นยูสเคสใช้สำหรับยกเลิกการจองตั๋วโดยสาร
Actor:	Member
Precondition:	สมาชิกทำการจองตั๋วโดยสาร โดยสารเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 (ต่อ)

Basic Flows:	1) สมาชิกเลือกเมนูยกเลิกการจองตัว 2) ระบบทำการเรียกใช้ Use Case "Deposit" 3) สมาชิกยืนยันการยกเลิกการจอง 4) ระบบทำการติดต่อเว็บเซอร์วิสเพื่อติดต่อกับธนาคาร 5) เว็บเซอร์วิสติดต่อกับธนาคารเพื่อทำการฝากเงินของสมาชิก กลับคืน 6) ระบบบันทึกข้อมูลการยกเลิกการจองตัว
Alternative Flows:	1a) สมาชิกไม่ทำการยกเลิกการจองและจบการทำงานของ Use Case นี้ 3a) สมาชิกไม่ยืนยันการจองจบการทำงานของ Use Case นี้
Post Conditions:	ระบบทำการยกเลิกการจองการเดินทาง

ตารางที่ 3.6 แสดงคำอธิบายยูสเคสตรวจสอบสถานะของการจองตัว

Name:	CheckReserveStatus
Brief Description:	เป็นยูสเคสใช้ตรวจสอบสถานะของการจองตัว
Actor:	Member
Precondition:	สมาชิกทำการจองตัวโดยสาร โดยสารเรียบร้อยแล้ว
Basic Flows:	1) สมาชิกเลือกเมนูตรวจสอบสถานะการจองตัว 2) ระบบทำการแสดงข้อมูลของการการเดินทาง
Alternative Flows:	-
Post Conditions:	-

ตารางที่ 3.7 แสดงคำอธิบายยูสเคสตรวจสอบหมายเลขบัญชี

Name:	VerifyAccount
Brief Description:	เป็นยูสเคสใช้สำหรับตรวจสอบว่าหมายเลขบัญชีนั้นถูกต้องหรือไม่

ตารางที่ 3.7 (ต่อ)

Actor:	System
Precondition:	สมาชิกทำการจองตั๋วโดยสาร
Basic Flows:	1) ระบบส่งข้อมูลเข้ามา 2) เว็บเซอร์วิส ทำการตรวจสอบ Account จากธนาคารที่ให้บริการ 3) เว็บเซอร์วิสส่งผลตอบกลับเพื่อยืนยันว่าสมาชิกนั้นมี Account อยู่หรือไม่ 4) ระบบส่งข้อมูลกลับไปยัง use case "ReserveTicket"
Alternative Flows:	2a) ระบบของธนาคารไม่ทำงาน
Post Conditions:	-

ตารางที่ 3.8 แสดงคำอธิบายยูสเคสที่ใช้สำหรับหักยอดเงินในบัญชี

Name:	Withdraw
Brief Description:	เป็นยูสเคสใช้สำหรับหักยอดเงินในบัญชี ในกรณีที่มีการซื้อตั๋วรถประจำทาง
Actor:	System
Precondition:	สมาชิกทำการจองตั๋วโดยสาร
Basic Flows:	1) ระบบส่งข้อมูลเข้ามา 2) เว็บเซอร์วิส ทำการหักยอดเงินในบัญชี จากธนาคารที่ให้บริการ 3) เว็บเซอร์วิสส่งผลตอบกลับเพื่อยืนยันผลการหักบัญชี 4) ระบบส่งข้อมูลกลับไปยัง use case "ReserveTicket"
Alternative Flows:	2a) ระบบของธนาคารไม่ทำงาน
Post Conditions:	-

ตารางที่ 3.9 แสดงคำอธิบายยูสเคสที่ใช้สำหรับคืนยอดเงินในบัญชี

Name:	Deposit
Brief Description:	เป็นยูสเคสที่ใช้สำหรับคืนยอดเงินในบัญชี
Actor:	System

ตารางที่ 3.9 (ต่อ)

Precondition:	สมาชิกทำการยกเลิกการจองตั๋วโดยสาร
Basic Flows:	1) ระบบส่งข้อมูลเข้ามา 2) เว็บเซอร์วิส ทำการโอนเงินเข้าบัญชีของธนาคารที่ให้บริการ 3) เว็บเซอร์วิสส่งผลตอบกลับเพื่อยืนยันผลการโอนเงิน 4) ระบบส่งข้อมูลกลับไปยัง use case "CancelTicket"
Alternative Flows:	2a) ระบบของธนาคารไม่ทำงาน
Post Conditions:	-

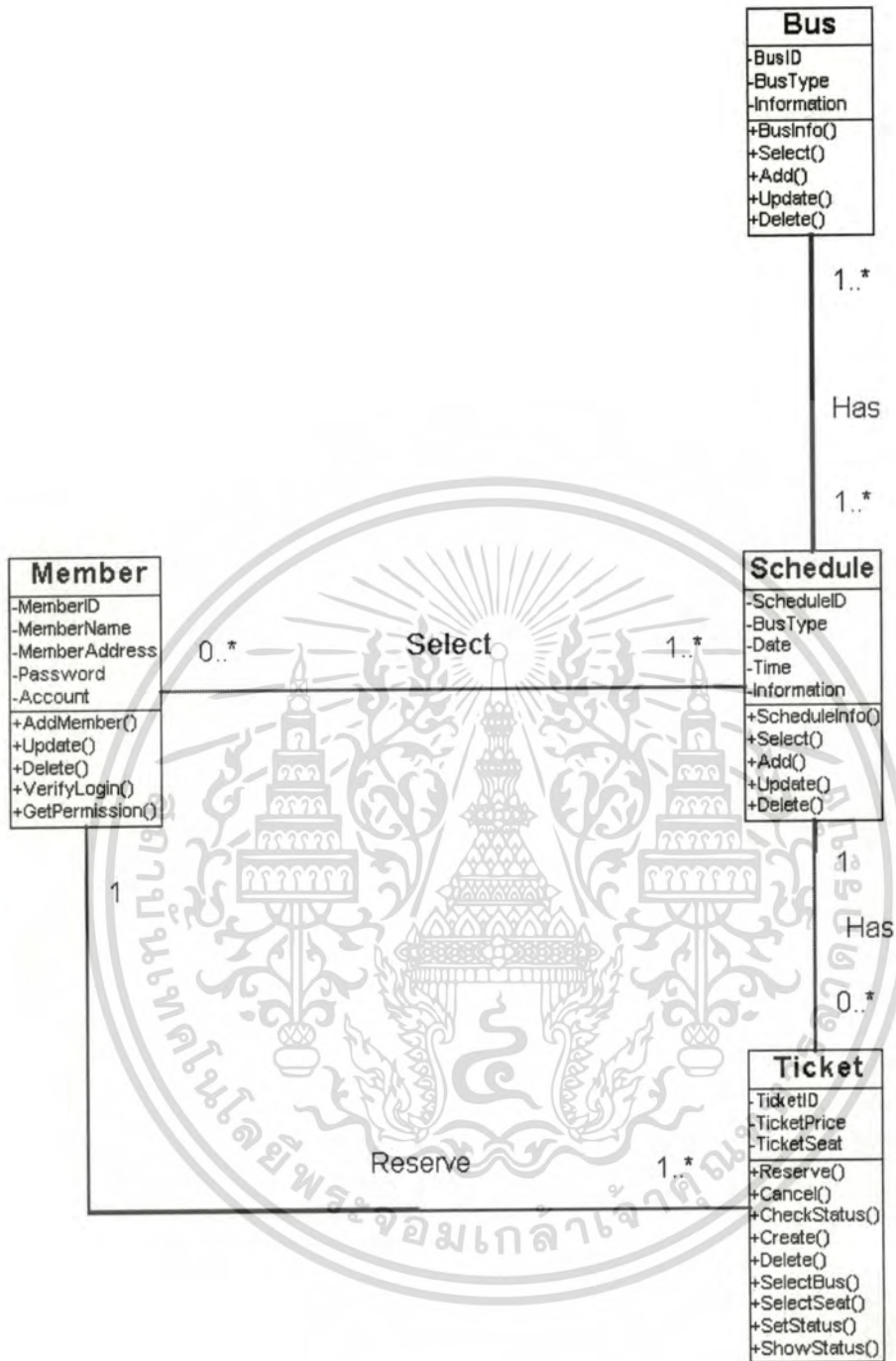
ตารางที่ 3.10 แสดงคำอธิบายยูสเคสที่ใช้สำหรับตรวจสอบยอดเงินคงเหลือในบัญชี

Name:	CheckBalance
Brief Description:	เป็นยูสเคสใช้สำหรับตรวจสอบว่ามียอดเงินคงเหลือในบัญชี
Actor:	System
Precondition:	สมาชิกทำการจองตั๋วโดยสาร
Basic Flows:	1) ระบบส่งข้อมูลเข้ามา 2) เว็บเซอร์วิส ทำการตรวจสอบยอดเงินจากธนาคารที่ให้บริการ 3) เว็บเซอร์วิสส่งผลตอบกลับเพื่อยืนยันว่าสมาชิกนั้นมีจำนวนเงินในบัญชีเพียงพอหรือไม่ 4) ระบบส่งข้อมูลกลับไปยัง use case "ReserveTicket"
Alternative Flows:	2a) ระบบของธนาคารไม่ทำงาน
Post Conditions:	-

3.3.3 Class Diagram

เป็นการแสดงความสัมพันธ์ระหว่างที่มีอยู่ในระบบ ซึ่งมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 คลาสไดอะแกรมของระบบจองตั๋วรถเดินทางฝั่งเว็บแอปพลิเคชัน

คลาสไดอะแกรมเป็นแผนภาพแสดงความสัมพันธ์ของคลาสทั้งหมดในระบบ จากการวิเคราะห์ระบบของตั๋วรถเดินทางฝั่งเว็บเบราว์เซอร์ ได้คลาสดังต่อไปนี้

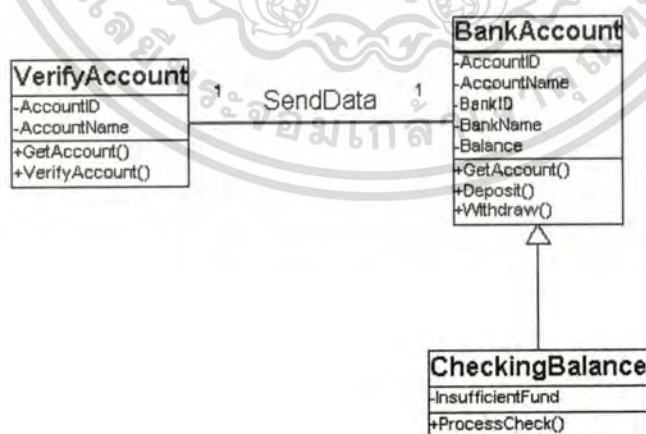
- Member หมายถึง บุคคล
- Bus หมายถึง รถประจำทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิได้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Schedule หมายถึง ตารางการเดินทาง
- Ticket หมายถึง ตั๋วรถโดยสาร

จากรูปที่ 3.3 คลาสไดอะแกรมจะแสดงให้เห็นถึงคลาสที่มีในระบบรวมถึงการกำหนดคุณสมบัติ เมธอดของคลาส และความสัมพันธ์ระหว่างคลาส ซึ่งแต่ละคลาสมีความสัมพันธ์ดังนี้

- คลาส Member มีความสัมพันธ์กับคลาส Schedule คือ ลูกค้า 1 คนสามารถมีตารางการเดินทางได้หลายตาราง
- คลาส Member มีความสัมพันธ์กับคลาส Ticket คือ ลูกค้า 1 คนสามารถมีตั๋วการเดินทางได้หลายตัว
- คลาส Schedule มีความสัมพันธ์กับคลาส Member คือ ตาราง 1 ตารางสามารถมีลูกค้าได้หลายคน หรือไม่มีก็ได้
- คลาส Schedule มีความสัมพันธ์กับคลาส Bus คือ ตาราง 1 ตารางมีรถประจำทางใช้ได้ 1 คัน หรือหลายคันก็ได้
- คลาส Schedule มีความสัมพันธ์กับคลาส Ticket คือ ตาราง 1 ตารางสามารถมีตั๋วได้หลายใบ หรือไม่มีก็ได้
- คลาส Bus มีความสัมพันธ์กับคลาส Schedule คือ รถประจำทาง 1 คัน มีตาราง 1 ตารางการเดินทางหลายตาราง หรือหลายตารางก็ได้
- คลาส Ticket มีความสัมพันธ์กับคลาส Member คือ ตัว 1 ใบถูกจองโดยลูกค้า 1 คน
- คลาส Ticket มีความสัมพันธ์กับคลาส Schedule คือ ตัว 1 ใบมีได้ 1 ตารางการเดินทาง



รูปที่ 3.4 คลาสไดอะแกรมของระบบจองตั๋วรถเดินทางฝั่งเว็บเซอร์วิส

คลาสดิอะแกรมเป็นแผนภาพแสดงความสัมพันธ์ของคลาสทั้งหมดในระบบ จากการ

วิเคราะห์ระบบของตั๋วรถเดินทางฝั่งเว็บเซอร์วิส ได้คลาสดังต่อไปนี้ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดก็ตาม ลีกรักษาฉบับนี้ไว้เพื่อประโยชน์ของอาจารย์จนถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- VerifyAccount หมายถึง ตรวจสอบบัญชี

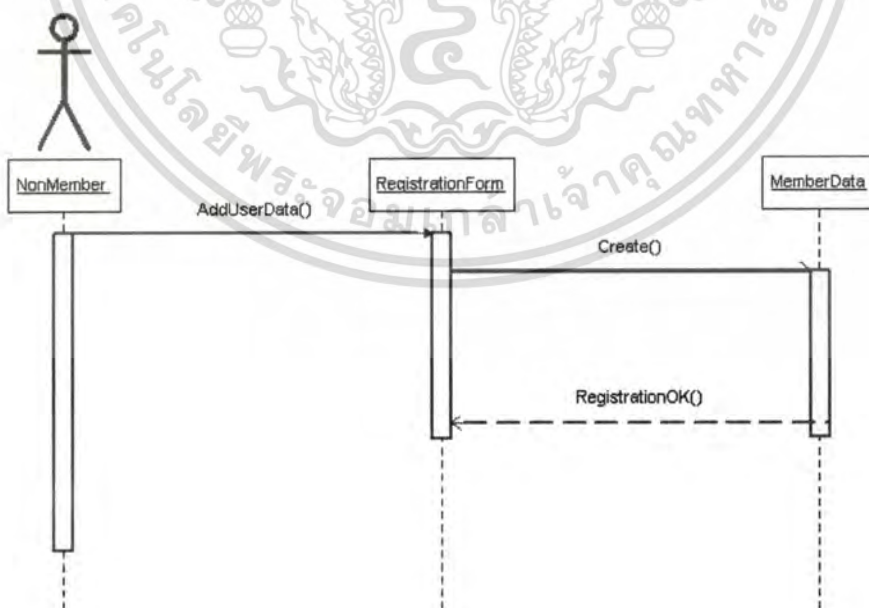
- BankAccount หมายถึง บัญชีธนาคาร
- CheckingBalance หมายถึง ตรวจสอบยอดคงเหลือ

จากรูปที่ 3.4 คลาสไดอะแกรมจะแสดงให้เห็นถึงคลาสที่มีในระบบรวมถึงการกำหนดคุณสมบัติ เมธอดของคลาส และความสัมพันธ์ระหว่างคลาส ซึ่งแต่ละคลาสมีความสัมพันธ์ดังนี้

- คลาส VerifyAccount มีความสัมพันธ์กับคลาส BankAccount คือ บัญชี 1 บัญชีส่งข้อมูลไปยังบัญชีธนาคาร 1 บัญชี
- คลาส BankAccount มีความสัมพันธ์กับคลาส VerifyAccount คือ บัญชีธนาคาร 1 บัญชี ถูกส่งข้อมูลไปบัญชี 1 บัญชี
- คลาส BankAccount มีความสัมพันธ์กับคลาส CheckingBalance แบบ Generalization

3.3.4 Sequence Diagram

Sequence Diagram เป็นแผนภาพที่ใช้แสดงถึงปฏิสัมพันธ์ระหว่างวัตถุในระบบ ซึ่งจะแสดงลำดับขั้นตอนการทำงานของงานนั้นๆ โดยแสดงในรูปแบบของการส่ง message ถึงกันระหว่างวัตถุ ลำดับเวลา ก่อนหลัง แสดงจากบนลงล่าง และเส้นที่ลากเชื่อมระหว่างวัตถุจะแสดงการรับส่ง message ระหว่างวัตถุ



รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของการ Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์โคอะแกรม

Name:	Register
Brief	
Description:	เป็นซีเควนซ์โคอะแกรมของการสมัครสมาชิก
Actor:	NonMember
Precondition:	-
Basic Flows:	1. RegistrationForm รับข้อมูล user เข้ามาและเรียกใช้งาน MemberData 2. เรียกใช้งานเมธอด create ของ MemberData
Alternative Flows:	-
Post Conditions:	1. Return ค่าผลลัพธ์ผ่านทางเมธอด RegistrationOK



รูปที่ 3.6 ซีเควนซ์โคอะแกรมของการ UpdateData

ตารางที่ 3.12 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์โคอะแกรม

Name:	UpdateData
Brief	
Description:	เป็นซีเควนซ์โคอะแกรมของการเปลี่ยนแปลงข้อมูลสมาชิก
Actor:	Member
Precondition:	ต้องเป็นสมาชิกของระบบเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในระบบเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.12 (ต่อ)

Basic Flows:	1. UserForm รับข้อมูล Member เข้ามาและเรียกใช้งาน UpdateData 2. เรียกใช้งานเมธอด Update ของ MemberData
Alternative Flows:	-
Post Conditions:	1. Return ค่าผลลัพธ์ผ่านทางเมธอด UpdateOK

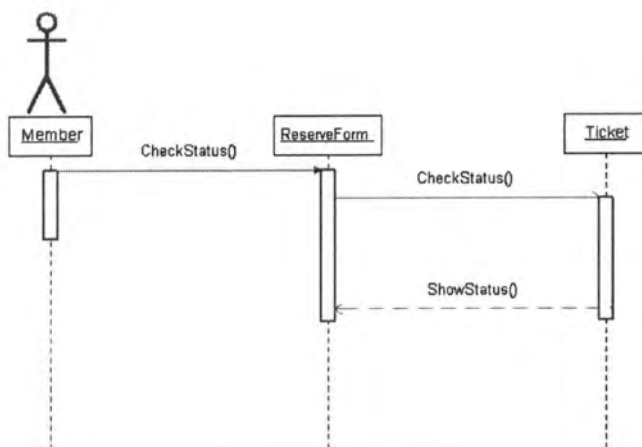


รูปที่ 3.7 ซีควেনซ์ไดอะแกรมของ Schedule

ตารางที่ 3.13 แสดงคำอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรม

Name:	Schedule
Brief Description:	เป็นซีควেনซ์ไดอะแกรมของการดูตารางการเดินทาง
Actor:	Member
Precondition:	ต้องเป็นสมาชิกของระบบเรียบร้อยแล้ว
Basic Flows:	1. ScheduleForm รับข้อมูล Member เข้ามาและเรียกใช้งาน ViewSchedule 2. เรียกใช้งานเมธอด ShowSchedule ของ Schedule
Alternative Flows:	-
Post Conditions:	1. Return ค่าผลลัพธ์ผ่านทางเมธอด ShowSchedule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

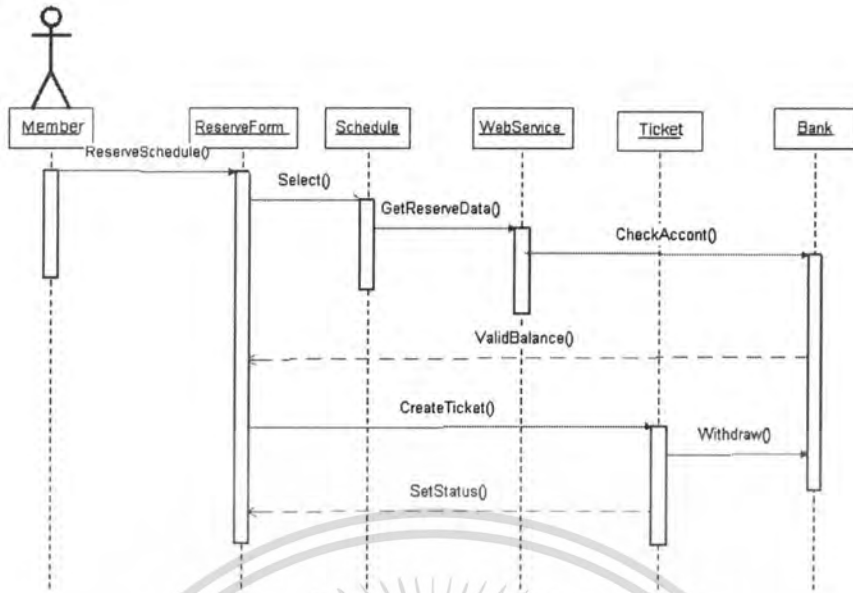


รูปที่ 3.8 ซีควেনซ์ไดอะแกรมของการ CheckReserveStatus

ตารางที่ 3.14 แสดงคำอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรม

Name:	CheckReserveStatus
Brief	
Description:	เป็นซีควেনซ์ไดอะแกรมของการตรวจสอบผลการจอง
Actor:	Member
Precondition:	ต้องจองตั๋วโดยสารเรียบร้อยแล้ว
Basic Flows:	<ol style="list-style-type: none"> 1. ReserveForm รับข้อมูล Member เข้ามาและเรียกใช้งาน CheckStatus 2. เรียกใช้งานเมธอด CheckStatus ของ Ticket
Alternative Flows:	-
Post Conditions:	1. Return ค่าผลลัพธ์ผ่านทางเมธอด ShowStatus

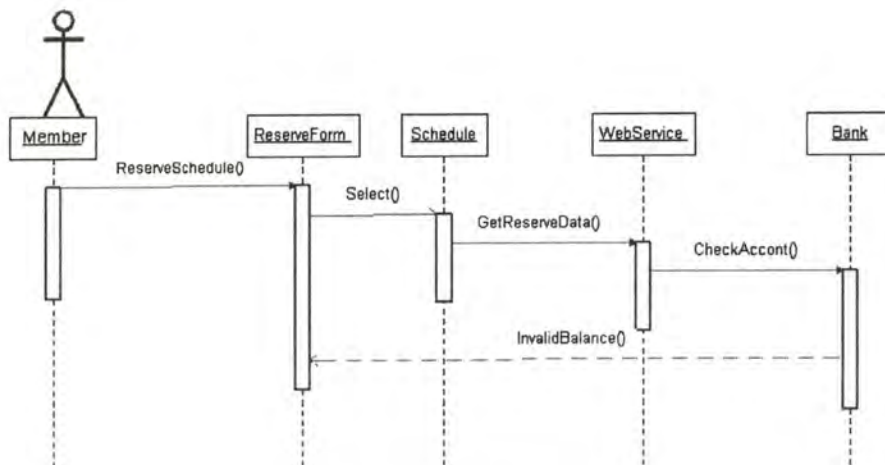
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ซีควเอนซ์ไดอะแกรมของการ ReserveTicket - Valid Account

ตารางที่ 3.15 แสดงคำอธิบายรายละเอียดประกอบซีควเอนซ์ไดอะแกรม

Name:	ReserveTicket - Valid Account
Brief	เป็นซีควเอนซ์ไดอะแกรมของการจองตั๋วโดยสารกรณีบัญชี
Description:	ถูกต้อง
Actor:	Member
Precondition:	ต้องเป็นสมาชิกของระบบเรียบร้อยแล้ว
Basic Flows:	<ol style="list-style-type: none"> 1. ReserveForm รับข้อมูล Member เข้ามาและเรียกใช้งาน ReserveSchedule 2. เรียกใช้งานเมธอด Select ของ Schedule 3. WebService เรียกใช้งานเมธอด GetReserveData และทำการตรวจสอบบัญชีผ่านทางเมธอด CheckAccount 4. Return ค่าผลลัพธ์ผ่านทางเมธอด ValidBalance 5. Ticket เรียกใช้งานเมธอด CreateTicket และทำการตัดเงินจากบัญชีผ่านทางเมธอด Withdraw
Alternative Flows:	-
Post	
Conditions:	1. Ticket return ค่าผลลัพธ์ผ่านทางเมธอด SetStatus

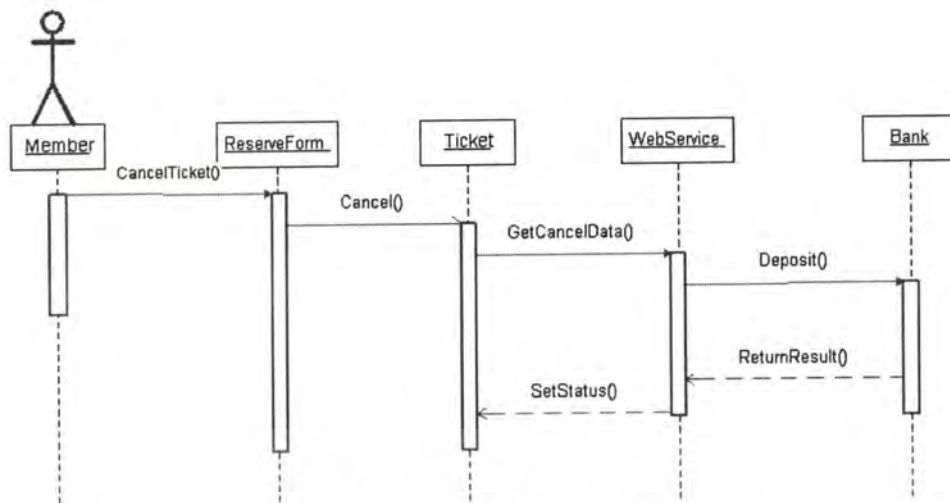


รูปที่ 3.10 ซีเควนซ์ไดอะแกรมของการ ReserveTicket – Invalid Account

ตารางที่ 3.16 แสดงคำอธิบายรายละเอียดประกอบซีเควนซ์ไดอะแกรม

Name:	ReserveTicket - Invalid Account
Brief	เป็นซีเควนซ์ไดอะแกรมของการจองตั๋วโดยสารกรณีบัญชีไม่
Description:	ถูกต้อง
Actor:	Member
Precondition:	ต้องเป็นสมาชิกของระบบเรียบร้อยแล้ว
Basic Flows:	<ol style="list-style-type: none"> 1. ReserveForm รับข้อมูล Member เข้ามาและเรียกใช้งาน ReserveSchedule 2. เรียกใช้งานเมธอด Select ของ Schedule 3. WebService เรียกใช้งานเมธอด GetReserveData และทำการตรวจสอบบัญชีผ่านทางเมธอด CheckAccount
Alternative Flows:	-
Post Conditions:	1. Bank return ค่าผลลัพธ์ผ่านทางเมธอด InvalidBalance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

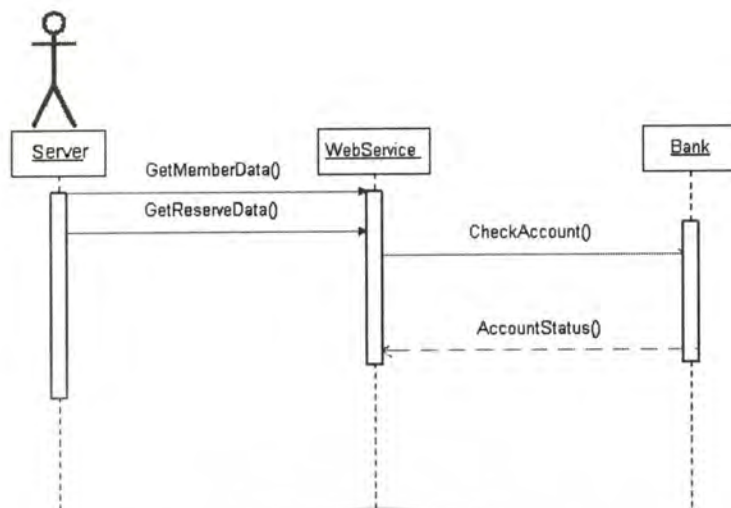


รูปที่ 3.11 ซีควেনซ์ไดอะแกรมของการ CancelTicket

ตารางที่ 3.17 แสดงคำอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรม

Name:	CancelTicket
Brief	
Description:	เป็นซีควেনซ์ไดอะแกรมของการยกเลิกตั๋วโดยสาร
Actor:	Member
Precondition:	ต้องทำการจองตั๋วรถโดยสารเรียบร้อยแล้ว
Basic Flows:	<ol style="list-style-type: none"> 1. ReserveForm รับข้อมูล Member เข้ามาและเรียกใช้งาน CancelTicket 2. เรียกใช้งานเมธอด Cancel ของ Ticket 3. WebService เรียกใช้งานเมธอด GetCancelData และทำการฝากเงินเข้าบัญชีผ่านทางเมธอด Deposit 4. Bank return ผลลัพธ์ผ่านทางเมธอด ReturnResult
Alternative Flows:	-
Post Conditions:	1. WebService return ผลลัพธ์ผ่านทางเมธอด SetStatus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ซีควেনซ์ไดอะแกรมของ Web Service Payment

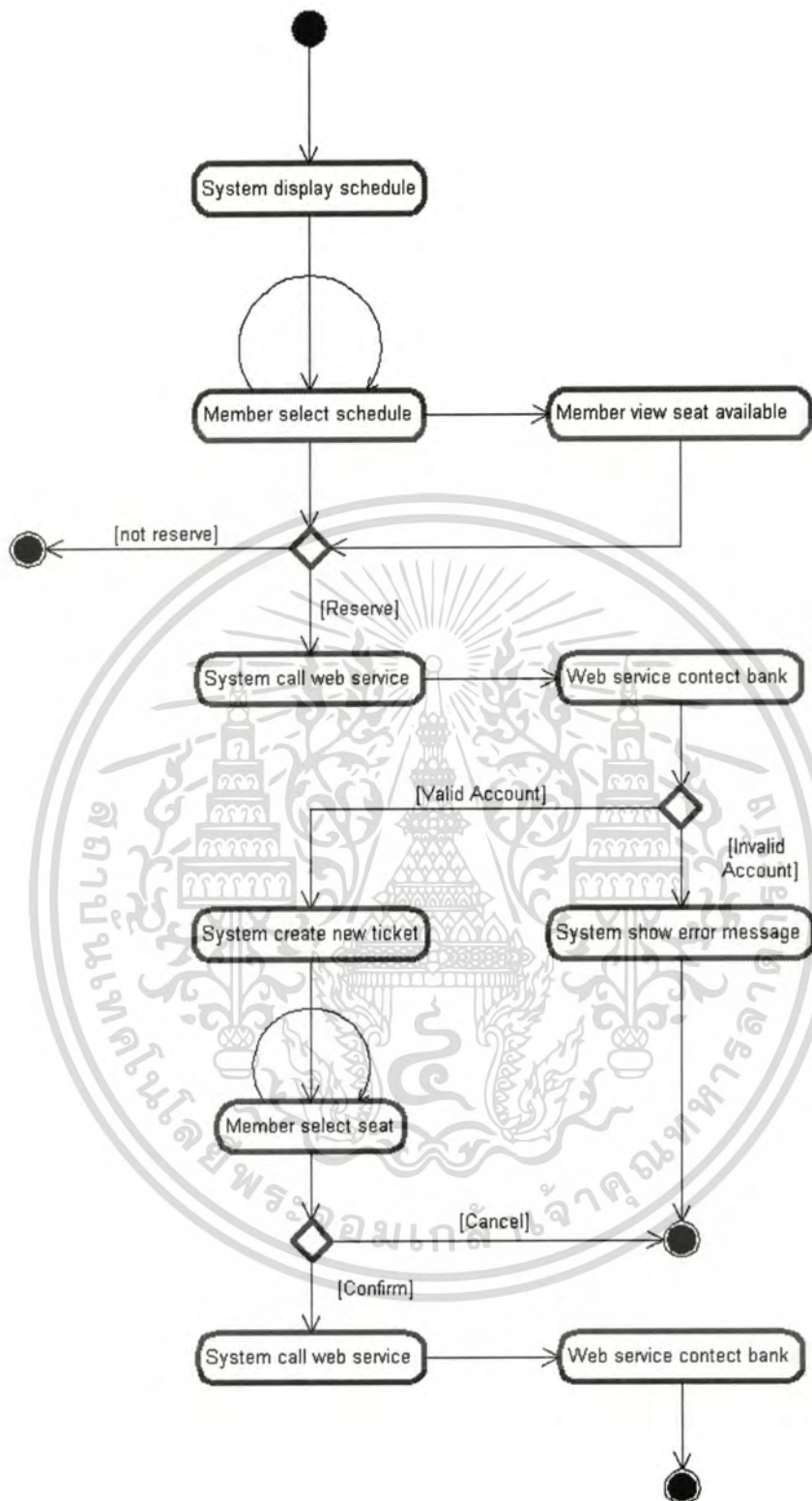
ตารางที่ 3.18 แสดงคำอธิบายรายละเอียดประกอบซีควেনซ์ไดอะแกรม

Name:	Web Service Payment
Brief	
Description:	เป็นซีควেনซ์ไดอะแกรมของการจ่ายเงิน
Actor:	Server
Precondition:	ต้องเป็นลูกค้าของธนาคาร
Basic Flows:	1. WebService รับข้อมูลเข้ามาโดยเรียกใช้งานเมธอด GetMemberData และ GetReserveData 2. เรียกใช้งานเมธอด CheckAccount ของ Bank
Alternative Flows:	-
Post Conditions:	1. Return ค่าผลลัพธ์ผ่านทางเมธอด AccountStatus

3.3.5 Activity Diagram

เป็นการแสดงถึงกิจกรรมที่เกิดขึ้นในระบบ ซึ่งจากรูปเป็นการแสดงกิจกรรมที่เกิดขึ้นของการจองตั๋วรถโดยสารประจำทางรวมถึงเงื่อนไขต่างๆ ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แอกทिवิตีไดอะแกรมของการจองตั๋วรถโดยสารประจำทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบฐานข้อมูล

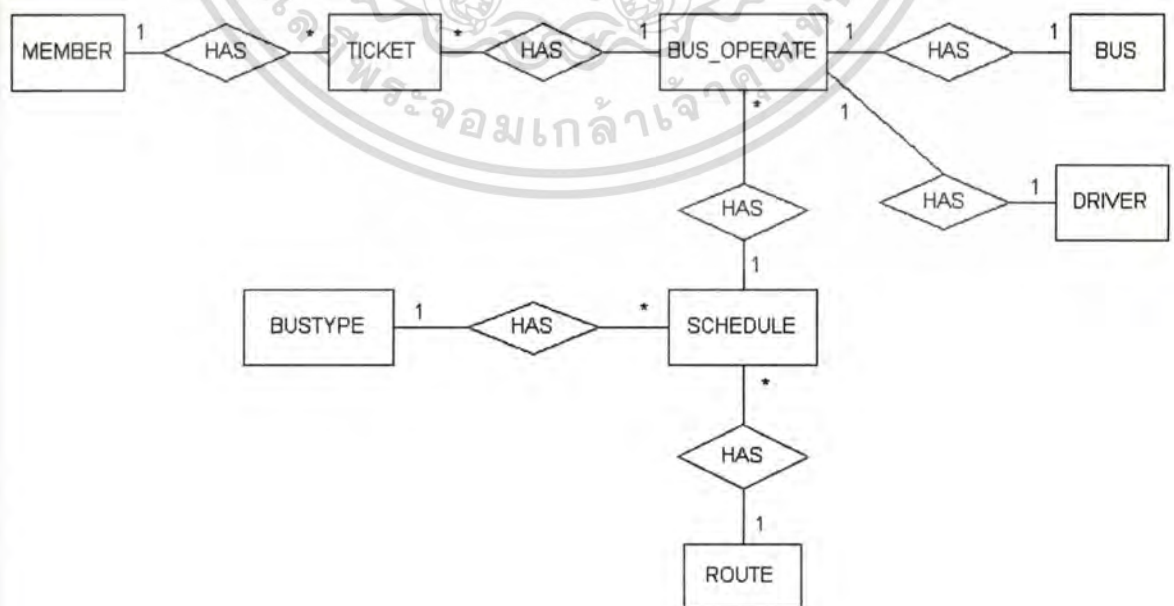
ในการออกแบบฐานข้อมูลขึ้นใช้งานในระบบจำเป็น ต้องอาศัยแบบจำลองข้อมูล เพื่อนำเสนอรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับข้อมูลในฐานข้อมูลที่ออกแบบ โดยการออกแบบฐานข้อมูลได้นำทฤษฎีโคอะแกรม มาทำการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ (Relation Database) โดยแสดงในรูปแบบของแผนภาพความสัมพันธ์ระหว่างเอนทิตี (Entity Relationship Diagram (E-R Diagram))

โดยได้แยกออกเป็น 2 ส่วนด้วยกันคือส่วน ของเว็บเบราว์เซอร์ และส่วนของเว็บเซิร์ฟเวอร์

4.1 ฐานข้อมูลของเว็บเบราว์เซอร์

สำหรับฐานข้อมูลของเว็บเบราว์เซอร์ ประกอบด้วยตารางต่างๆ ดังต่อไปนี้

1. Member เป็นตารางที่ใช้สำหรับเก็บข้อมูลสมาชิก
2. Passenger_Record เป็นตารางที่ใช้สำหรับเก็บข้อมูลการสำรองที่นั่ง
3. Bus_Operate เป็นตารางที่ใช้สำหรับเก็บข้อมูลตารางการเดินรถ
4. Schedule เป็นตารางที่ใช้สำหรับเก็บข้อมูลรถ
5. Fare เป็นตารางที่ใช้สำหรับเก็บข้อมูลอัตราค่าโดยสาร



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ได้ หากมีการนำออกไปใช้โดยไม่ได้รับอนุญาต เจ้าของลิขสิทธิ์ขอสงวนสิทธิ์ในการดำเนินคดีตามกฎหมายที่เกี่ยวข้อง

มีรายละเอียดดังนี้

ตารางที่ 4.1 รายละเอียดของตาราง Route

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Route	รหัสเส้นทาง	VARCHAR(5)	PK	
Origin	ต้นทาง	VARCHAR(20)		
Destination	ปลายทาง	VARCHAR(20)		

ตารางที่ 4.2 รายละเอียดของตาราง Schedule

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Schedule	รหัสตารางเดินรถ	VARCHAR(5)	PK	
Route	รหัสเส้นทาง	VARCHAR(5)	PK, FK	ROUTE
Dept time	เวลาออกเดินทาง	TIME		
Arrv time	เวลาถึงที่หมาย	TIME		
Bustype	รหัสประเภทรถ	VARCHAR(5)	FK	BUSTYPE
Fare	ราคาค่าโดยสาร	INTEGER(5)		
Distance	ระยะทาง	INTEGER(5)		
Duration	ระยะเวลาเดินทาง	INTEGER(5)		

ตารางที่ 4.3 รายละเอียดของตาราง Bus_Operate

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Schedule	รหัสตารางเดินรถ	VARCHAR(5)	PK, FK	SCHEDULE
Date	วันที่เดินทาง	DATE	PK	
Available_seat	จำนวนที่นั่งที่เหลือ	INTEGER(5)		
Bus_no	รหัสรถประจำทาง	VARCHAR(5)	FK	BUS
Driver_no	รหัสคนขับรถโดยสาร	VARCHAR(5)	FK	DRIVER

ตารางที่ 4.4 รายละเอียดของตาราง Ticket

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Schedule	รหัสตารางเดินรถ	VARCHAR(5)	PK, FK	SCHEDULE
Date	วันที่เดินทาง	DATE	PK, FK	BUS_OPERATE
Member_no	รหัสสมาชิก	VARCHAR(5)	FK	MEMBER
Number_seat	รหัสที่นั่ง	VARCHAR(5)		

ตารางที่ 4.5 รายละเอียดของตาราง Member

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Member_no	รหัสสมาชิก	VARCHAR(5)	PK	
Username	ชื่อ	VARCHAR(10)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 (ต่อ)

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Password	รหัสผ่าน	VARCHAR(10)		
Name	ชื่อลูกค้า	VARCHAR(10)		
Surname	นามสกุล	VARCHAR(10)		
Phone_no	หมายเลขโทรศัพท์	VARCHAR(10)		
Email	อีเมล	VARCHAR(10)		

ตารางที่ 4.6 รายละเอียดของตาราง Bustype

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Bustype	รหัสประเภทรถ	VARCHAR(5)	PK	
Description	คำอธิบาย	VARCHAR(10)		

ตารางที่ 4.7 รายละเอียดของตาราง Bus

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Bus_no	รหัสรถประจำทาง	VARCHAR(5)	PK	
Bus_name	ชื่อรถประจำทาง	VARCHAR(20)		

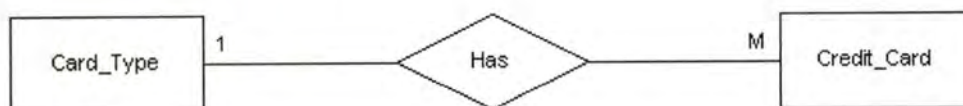
ตารางที่ 4.8 รายละเอียดของตาราง Driver

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Driver_no	รหัสคนขับรถโดยสาร	VARCHAR(5)	PK	
Driver_name	ชื่อคนขับรถประจำทาง	VARCHAR(20)		

4.2 ฐานข้อมูลของเว็บเซอร์วิสธนาคาร

สำหรับฐานข้อมูลของเว็บเบราร์เซอร์ ประกอบด้วยตารางต่างๆ ดังต่อไปนี้

1. Card_Type เป็นตารางที่ใช้สำหรับเก็บประเภทบัตรเครดิต
2. Credit_Card เป็นตารางที่ใช้สำหรับเก็บข้อมูลบัตรเครดิต เช่น รหัสบัตรและวันหมดอายุบัตรเครดิต



รูปที่ 4.2 E-R Diagram ของฝั่งเว็บเซอร์วิสธนาคาร

มีรายละเอียดดังนี้

ตารางที่ 4.9 รายละเอียดของตาราง Card_Type

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Cardtypeid	รหัสประเภทบัตร	VARCHAR(5)	PK	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัย กรุณาติดต่อฝ่ายบริการลูกค้า โทร. 1676 หรือ 02-2542400

ตารางที่ 4.9 (ต่อ)

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
Description	คำอธิบาย	VARCHAR(10)		

ตารางที่ 4.10 รายละเอียดของตาราง Credit_Card

ชื่อ	คำอธิบาย	ชนิดข้อมูล	ชนิดคีย์	ตารางที่อ้างอิง
CardId	รหัสบัตรเครดิต	VARCHAR(16)	PK	
CardtypeId	รหัสประเภทบัตร เช่น 1 คือ VISA	VARCHAR(5)		
ExpireDate	วันหมดอายุบัตร	DATE		



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบ

จากการวิเคราะห์และออกแบบระบบบริการจัดหาผู้เชี่ยวชาญในบทที่ผ่านมา ทำให้ทราบถึงภาพรวมของระบบ รวมถึงขั้นตอนการทำงาน เพื่อเป็นแนวทางในการพัฒนาระบบสามารถอธิบายเป็นหัวข้อ ดังนี้

5.1 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ

การพัฒนาระบบในโครงการนี้ ใช้ฮาร์ดแวร์ ซอฟต์แวร์และเครื่องมือในการพัฒนาดังนี้

5.1.1 ฮาร์ดแวร์

ฮาร์ดแวร์ที่ใช้ ในการทดสอบและพัฒนาระบบประกอบด้วย

- เครื่องคอมพิวเตอร์พีซีสำหรับให้บริการระบบ

5.1.2 ซอฟต์แวร์และเครื่องมือ

ซอฟต์แวร์และเครื่องมือที่ใช้ ในการทดสอบและพัฒนาระบบประกอบด้วย

- ระบบปฏิบัติการ : Windows XP Professional
- เว็บเบราว์เซอร์ : Internet Explorer 7.0
- โปรแกรมจัดการฐานข้อมูล : Microsoft SQL Server 2005 ใช้สำหรับเครื่องคอมพิวเตอร์พีซีที่ให้บริการระบบ เพื่อจัดเก็บและจัดการกับข้อมูลในระบบฐานข้อมูล
- เครื่องมือที่ใช้ในการพัฒนาโปรแกรม : NetBeans IDE 5.5.1

5.2 รายละเอียดการทำงานของระบบ

โครงการพัฒนาระบบสืบค้นตารางการเดินทางและสำรองที่นั่งของรถประจำทางโดยใช้เว็บเซอร์วิสมีระบบเว็บแอปพลิเคชันบัสเทลเวล เป็นระบบเว็บแอปพลิเคชันที่ใช้ติดต่อกับผู้ใช้งานระบบโดยตรง มีรายละเอียดหน้าจอการทำงานดังต่อไปนี้

1. หน้าจอหลักของระบบเว็บแอปพลิเคชันบัสเทลเวล
2. หน้าจอตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน
3. หน้าจอลงทะเบียนข้อมูลสำหรับสมาชิก
4. หน้าจอแสดงผลการสมัครสมาชิก

เอกสารนี้เป็นเอกสารสำหรับแก้ไข เปลี่ยนแปลงข้อมูลสมาชิกนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หน้าจอแสดงผลการแก้ไขข้อมูลสมาชิก
7. หน้าจอการสืบค้นตารางการเดินทาง
8. หน้าจอแสดงผลการสืบค้นตารางการเดินทาง
9. หน้าจอการสำรองที่นั่ง
10. หน้าจอยืนยันการสำรองที่นั่ง
11. หน้าจอการสืบค้นราคาค่าโดยสาร
12. หน้าจอแสดงผลการสืบค้นราคาค่าโดยสาร

หน้าจอหลักของระบบเว็บแอปพลิเคชันบัตรเที่ยว

5.3 รายละเอียดของการพัฒนาระบบ

5.3.1 หน้าจอส่วนเว็บเพจของบริษัท

รูปที่ 5.1 แสดงหน้าจอหลักของเว็บเพจเมื่อสมาชิกเข้าใช้บริการ จะประกอบด้วยเมนูต่างๆ แสดงให้เห็นชัดเจน ซึ่งเมนูหลักจะประกอบไปด้วย

1. หน้าแรก แสดงหน้าจอหลักของเว็บเพจ
2. รู้จักเรา แสดงรายละเอียดต่างๆ ของบริษัท เช่น ที่อยู่ หมายเลขโทรศัพท์ ช่องทางการชำระเงิน เป็นต้น
3. วิธีการจองและชำระเงิน แสดงวิธีการจองและช่องทางการชำระเงิน
4. เส้นทางและผู้ให้บริการ แสดงเส้นทางการเดินทางและผู้ให้บริการ
5. ตรวจสอบข้อมูลการจอง แสดงข้อมูลการจอง
6. คำถามที่พบบ่อย แสดงคำถามที่ผู้ให้บริการถามบ่อยครั้ง
7. ติดต่อเรา แสดงช่องทางการติดต่อบริษัท

Bus Travel

ตลอด 24 ชั่วโมง
 โทร → 02 123 4567

หน้าแรก
รู้จักเรา
วิธีการจองและชำระเงิน
เส้นทางและคู่มือบริการ
ตรวจสอบผลการจอง
คำถามที่พบบ่อย
ติดต่อเรา

ค้นหาเที่ยวรถ

เดินทาง: ไปกลับ เที่ยวเดียว

จังหวัดต้นทาง:

จังหวัดปลายทาง:

ยินดีต้อนรับ: Guest [Logout](#)


วันที่ออกเดินทาง:

วันที่เดินทางกลับ:

จำนวนผู้โดยสาร: คน

มาตรฐานรถ:

ถ้าหิวสมัคร
 Username:
 Password:



ใหม่! แคนท์บริการเดินทาง
สถานีขนส่งสายใต้ใหม่ ด.บวรราชธานี
ชนนี

ข่าวสาร

สถานีขนส่งสายใต้ใหม่พร้อมเปิดให้บริการแล้ว

คำแนะนำในการจองตั๋ว
 และนโยบายประกันคุณภาพ

รูปที่ 5.1 หน้าจอหลักของระบบ

กรณีที่ลูกค้าเข้ามาใช้งานเว็บเพจไม่เคยลงทะเบียนเก็บข้อมูลมาก่อน จะต้องลงทะเบียนเพื่อเก็บข้อมูลของลูกค้าในระบบ ดังรูปที่ 5.2

กรณีที่ลูกค้าเคยลงทะเบียนไว้แล้ว จะสามารถเข้าใช้งานระบบเพื่อส่งข้อมูลรายละเอียดความต้องการได้ทันที โดยระบบจะตรวจสอบชื่อผู้ใช้งานและรหัสผ่านเพื่อยืนยันความเป็นตัวจริงในการเข้าใช้งาน ดังรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมัครสมาชิก

ข้อมูลส่วนตัว

ชื่อ *

นามสกุล *

ที่อยู่ *

หมายเลขโทรศัพท์ * (กรุณากรอกเฉพาะตัวเลข เช่น 0811234567)

อีเมล *

ข้อมูลสมาชิก

Username *

(4 - 20 ตัวอักษร)

Password *

(4 - 20 ตัวอักษร)

Confirm Password

เงื่อนไขการสมัครโดยผ่านทางเว็บไซต์

- ผู้ใช้บริการได้ตรวจสอบข้อมูล ข้อกำหนด เงื่อนไข รูปแบบ ค่าบริการ ค่าธรรมเนียม และรายละเอียดอื่น ๆ โดยถี่ถ้วน แล้วจึงได้ตกลงทำรายการตามวิธีการที่บริษัทกำหนด และ/หรือ ผู้ใช้บริการได้ตรวจสอบรายละเอียดเพิ่มเติมจากรายการและผู้ให้บริการเดินรถที่เลือกใช้บริการ เช่น การให้บริการ การให้บริการ และเงื่อนไขอื่น ๆ ที่ปรากฏอยู่ในเว็บไซต์ รวมถึงข้อกำหนดและเงื่อนไขของ ผู้ให้บริการเดินรถที่เลือกใช้บริการเรียบร้อยแล้ว
- การสำรองที่นั่งจะได้รับบริการขึ้นรถหลังจากเกิดการชำระค่าโดยสารเต็มจำนวน และบริษัทขอสงวนสิทธิ์ยกเลิกรายการจองซึ่งไม่ชำระค่าโดยสารภายในระยะเวลาที่กำหนด ชำระค่าโดยสารไม่ครบถ้วน หรือทำรายการชำระเงิน

รูปที่ 5.2 หน้าจอลงทะเบียนข้อมูลสำหรับสมาชิก

จังหวัดปลายทาง:

วันที่ขึ้นรถ: มีนาคม 2551

เวลาที่ขึ้นรถ: มีนาคม 2551

จำนวนผู้โดยสาร: คน

หมวดรถ:

สำหรับสมาชิก
Username:
Password:

สมัครสมาชิก ฟรี! หรือรับสิทธิพิเศษมากมาย
* สมัครสมาชิก * สิทธิพิเศษ

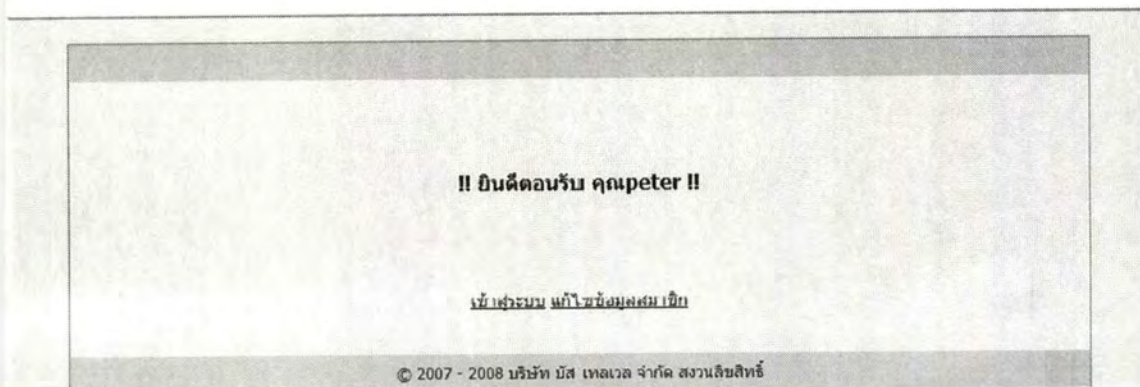
ใหม่! แทนที่การเดินทาง
สถานีขนส่งสายใต้ใหม่ ถ.บรมราชชนนี
ขณะนี้

* สถานีขนส่งสายใต้ใหม่พร้อมเปิดให้บริการแล้ว

**คำแนะนำในการจองตั๋ว
และนโยบายประกันคุณภาพ**

เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 5.3 หน้าจอตรวจเช็คข้อมูลผู้ใช้งานและรหัสผ่านหน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกรอกชื่อและรหัสผ่านจะแสดงข้อความต้อนรับ ดังรูปที่ 5.4



รูปที่ 5.4 หน้าจอแสดงผลการสมัครสมาชิก

เมื่อสมาชิกลงทะเบียนเรียบร้อยแล้ว สมาชิกสามารถเลือกทำการเปลี่ยนแปลงรายละเอียดข้อมูลต่างๆ ของสมาชิกได้ ดังรูปที่ 5.5

รูปที่ 5.5 หน้าจอสำหรับแก้ไข เปลี่ยนแปลงข้อมูลสมาชิก

หลังจากแก้ไขข้อมูลสมาชิกเรียบร้อยแล้ว จะแสดงข้อความเพื่อยืนยันการเปลี่ยนแปลง ดังรูป
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ที่ 5.6
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

!! แกะไข่มูลสมาชิกเรียบร้อยแล้ว !!

[หน้าหลัก](#) [เข้าสู่ระบบ](#)

© 2007 - 2008 บริษัท บัส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.6 หน้าจอแสดงผลการแก้ไขข้อมูลสมาชิก

สมาชิกสามารถเลือกค้นหาเที่ยวรถ เส้นทางการเดินทาง ราคา และรายละเอียดต่างๆ ของรถโดยสารที่ให้บริการได้ ดังรูปที่ 5.7

The screenshot shows the 'Bus Travel' website interface. At the top, there is a navigation menu with links: หน้าแรก, บริการเรา, วิธีการจองและชำระเงิน, เส้นทางและรถให้บริการ, ตรวจสอบผลการจอง, ค่าตอบแทนบ่อย, ติดต่อเรา. The main content area is titled 'ค้นหาเที่ยวรถ' (Search for bus routes). It features a search form with the following fields:

- เส้นทาง: ไปกลับ เที่ยวเดียว
- จังหวัดต้นทาง:
- จังหวัดปลายทาง:
- วันออกเดินทาง:
- วันที่เดินทางกลับ:
- จำนวนผู้โดยสาร: คน
- มาตรฐานรถ:

 Below the form is a 'ค้นหา' (Search) button. To the right of the form is a large image of a white bus. Below the image, there is a section for 'ข่าวสาร' (News) with the text: 'ใหม่! แสนที่ทางเดินทาง สถานีขนส่งสายใต้ใหม่ ด.บรมราชชนนี ธานี' and '* สถานีขนส่งสายใต้ใหม่พร้อมเปิดให้บริการแล้ว'. At the bottom of the search form area, there is a button that says 'คำแนะนำในการจองตั๋ว และนโยบายประกันคุณภาพ'.

รูปที่ 5.7 หน้าจอการสืบค้นตารางการเดินทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอแสดงรายละเอียดต่างๆ ของการสืบค้นข้อมูลของรถประจำทาง ดังรูปที่ 5.8

รายละเอียดการเดินทางไป							
ต้นทาง	ปลายทาง	มาตรฐานรถ	จำนวนที่นั่ง	ราคา	วันรถออก	เวลารถออก	ถึงปลายทางโดยประมาณ
○ กรุงเทพมหานคร	จันทบุรี	ม1ข	40	193	01/03/2008	05.00	08.30
○ กรุงเทพมหานคร	จันทบุรี	ม1ข	40	193	01/03/2008	07.30	11.00
○ กรุงเทพมหานคร	จันทบุรี	ม1ข	40	193	01/03/2008	12.00	15.30
○ กรุงเทพมหานคร	จันทบุรี	ม1ข	40	193	01/03/2008	17.00	20.30
○ กรุงเทพมหานคร	จันทบุรี	ม1ข	40	193	01/03/2008	20.00	23.30

รายละเอียดการเดินทางกลับ							
ต้นทาง	ปลายทาง	มาตรฐานรถ	จำนวนที่นั่ง	ราคา	วันรถออก	เวลารถออก	ถึงปลายทางโดยประมาณ
○ จันทบุรี	กรุงเทพมหานคร	ม1ข	40	193	2/03/2008	07.30	11.00
○ จันทบุรี	กรุงเทพมหานคร	ม1ข	40	193	2/03/2008	11.45	15.15
○ จันทบุรี	กรุงเทพมหานคร	ม1ข	40	193	2/03/2008	18.45	22.15

[ขึ้นคอนต่อไป](#)

เลือกเส้นทางใหม่

© 2007 - 2008 บริษัท บัส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.8 หน้าจอแสดงผลการสืบค้นตารางการเดินทาง

หน้าจอแสดงผลการสำรองที่นั่ง ประกอบไปด้วยข้อมูลลูกค้า และรายละเอียดการเดินทาง ดังรูปที่ 5.9

ผลการสำรองที่นั่ง

ข้อมูลลูกค้า		
คุณ: สมชาย แซ่ตั้ง		
ที่อยู่: 99/99 บางเขน		
เบอร์โทรศัพท์: 0811234567		
Email: test@hotmail.com		
รายละเอียดการเดินทางไป		
ประเภทรถโดยสาร	ต้นทาง	ปลายทาง
ม1ข	กรุงเทพมหานคร	จันทบุรี
	วันที่ 01/03/2008 เวลา 05.00	วันที่ 01/03/2008 เวลา 08.30
รายละเอียดการเดินทางกลับ		
ประเภทรถโดยสาร	ต้นทาง	ปลายทาง
ม1ข	จันทบุรี	กรุงเทพมหานคร
	วันที่ 2/03/2008 เวลา 07.30	วันที่ 2/03/2008 เวลา 11.00
ราคาค่าโดยสาร 386 บาท		

[ขึ้นคอนต่อไป](#)

[กลับหน้าหลัก](#) [Logout](#)

© 2007 - 2008 บริษัท บัส เทลเวล จำกัด สงวนลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 5.9 หน้าจอการสำรองที่นั่ง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากยืนยันการจองเรียบร้อยแล้วทำการจ่ายเงินค่าเดินทางด้วยบัตรเครดิต ดังรูปที่

5.10

กรุณากรอกรายละเอียดบัตรเครดิต

Card Type *

Card Number *

Verify Code *

Expire Date *

[กลับหน้าหลัก Logout](#)

© 2007 - 2008 บริษัท บีส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.10 หน้าจอยืนยันการสำรองที่นั่ง

เมื่อสมาชิกทำการสำรองที่นั่งเรียบร้อยแล้ว แต่เมื่อสมาชิกเปลี่ยนความต้องการ สมาชิกสามารถยกเลิกการสำรองที่นั่งได้ ดังรูปที่ 5.11

ยกเลิกการจองที่นั่ง

ข้อมูลลูกค้า

คุณ: สมชาย แซ่ตั้ง

ที่อยู่: 99/99 บางเขน

เบอร์โทรศัพท์: 0811234567

Email: test@hotmail.com

ประเภทโดยสาร	ต้นทาง	ปลายทาง
<input type="radio"/> ม1ข	กรุงเทพมหานคร วันที่ 01/03/2008 เวลา 05.00	จิ่นหนิง วันที่ 01/03/2008 เวลา 08.30

ประเภทโดยสาร	ต้นทาง	ปลายทาง
<input type="radio"/> ม1ข	จิ่นหนิง วันที่ 2/03/2008 เวลา 07.30	กรุงเทพมหานคร วันที่ 2/03/2008 เวลา 11.00

[กลับหน้าหลัก](#)

© 2007 - 2008 บริษัท บีส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.11 หน้าจอยกเลิกการสำรองที่นั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาชิกสามารถสืบค้นอัตราค่าโดยสารและรายละเอียดต่างๆ โดยเลือกต้นทางและปลายทางที่สมาชิกต้องการเดินทางไป แสดงได้ดังรูปที่ 5.12-5.13

สืบค้นอัตราค่าโดยสาร

โปรดเลือกเส้นทาง

ต้นทาง กรุงเทพมหานคร *
 ปลายทาง เชียงใหม่ *

© 2007 - 2008 บริษัท บัส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.12 หน้าจอการสืบค้นราคาค่าโดยสาร

อัตราค่าโดยสาร

ประเภทรถ	ต้นทาง	ปลายทาง	อัตราค่าโดยสาร (ราคา)	เวลาารถออก
พิเศษ24	กรุงเทพมหานคร	เชียงใหม่	805	09.00
ม1พ	กรุงเทพมหานคร	เชียงใหม่	605	20.45
พิเศษ24	กรุงเทพมหานคร	เชียงใหม่	805	19.00
พิเศษ24	กรุงเทพมหานคร	เชียงใหม่	805	20.00
พิเศษ24	กรุงเทพมหานคร	เชียงใหม่	805	21.30

คลิกที่พิกัด สืบค้นอัตราค่าโดยสาร

© 2007 - 2008 บริษัท บัส เทลเวล จำกัด สงวนลิขสิทธิ์

รูปที่ 5.13 หน้าจอแสดงผลการสืบค้นราคาค่าโดยสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

6.1 สรุปผลการออกแบบและพัฒนาระบบ

โครงการพัฒนาระบบนี้มีจุดประสงค์เพื่อศึกษาการนำเทคโนโลยีเว็บเซอร์วิสมาประยุกต์เพื่อใช้งานในการให้บริการสืบค้นข้อมูลและสำรองที่หนึ่งของรถโดยสารประจำทาง และเพิ่มประสิทธิภาพในการทำงาน ในด้านการอำนวยความสะดวกให้ผู้โดยสาร ทำให้การบริหารระบบงานมีประสิทธิภาพและช่วยลดเวลาในการทำงานที่ไม่จำเป็นลงได้ ระบบการจองตั๋วรถโดยสารประจำทางผ่านทางอินเทอร์เน็ต โดยใช้เว็บเซอร์วิสเข้ามาช่วยนั้น เป็นอีกทางเลือกหนึ่งที่น่าสนใจ เพราะเป็นการนำเทคโนโลยีที่มีอยู่ในปัจจุบันมาประยุกต์ใช้ได้อย่างมีประสิทธิภาพ และเหมาะสมกับวิถีชีวิตของมนุษย์ในปัจจุบัน เนื่องจากในยุคนี้ผู้คนมีความเร่งรีบและต้องแข่งขันกันอยู่ตลอดเวลา ดังนั้นระบบนี้จึงช่วยอำนวยความสะดวกแก่ผู้ที่ต้องการซื้อตั๋วโดยสาร โดยไม่ต้องเดินทางเพื่อไปยื่นต่อแถวจองตั๋วอีกต่อไป

ในการพัฒนาโครงการนี้ได้ทำการวิเคราะห์และออกแบบระบบ โดยใช้วิธีการเชิงวัตถุ Unified Modeling Language (UML) และทำการพัฒนาระบบโดยใช้ภาษา Java เข้ามาช่วยในการพัฒนาระบบและใช้ฐานข้อมูล SQL Server 2005 ในการเก็บข้อมูล

6.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ

ประโยชน์ที่ผู้พัฒนาระบบได้รับ สรุปได้ดังนี้

1. ได้รับความรู้ในการเขียนโปรแกรมโดยใช้ภาษาจาวาและเทคโนโลยีเชิงวัตถุ
2. ได้รับความรู้ในเทคโนโลยีใหม่และเกิดทักษะในการพัฒนาระบบ
3. ได้รับความรู้ในเรื่องการวิเคราะห์และออกแบบระบบโดยใช้ UML

6.3 ปัญหาและอุปสรรคในการพัฒนาระบบ

ระหว่างการพัฒนาได้พบปัญหาและอุปสรรคในเรื่องของเทคโนโลยีและพื้นฐานความรู้ที่จำเป็น รวมทั้งต้องใช้ความรู้พื้นฐานในเรื่องการเขียนโปรแกรมด้วยภาษาจาวา เทคโนโลยีในการทำเว็บเซอร์วิส และเทคโนโลยีเชิงวัตถุ ทำให้เป็นอุปสรรคต่อการออกแบบและพัฒนาระบบทำให้ต้องใช้เวลาในการศึกษา

6.4 ข้อเสนอแนะและแนวทางการพัฒนาเพิ่มเติม

1. ควรเพิ่มเติมความสามารถในการรักษาความปลอดภัยของข้อมูล ที่ส่งระหว่างผู้เรียกใช้บริการเว็บเซอร์วิส และผู้ให้บริการเว็บเซอร์วิส เพื่อป้องกันผู้ที่ไม่เกี่ยวข้องกับระบบเข้าถึงข้อมูลได้ เช่น การเข้ารหัสและถอดรหัสข้อมูล เป็นต้น
2. เพิ่มช่องทางการให้บริการเพื่อให้ลูกค้าสามารถรับบริการผ่านทางอุปกรณ์อื่นๆ ได้ เช่น โทรศัพท์มือถือ
3. ควรนำข้อมูลลูกค้าในระบบมาทำการวิเคราะห์ข้อมูลเพิ่มเติม เพื่อออกรายงานนำเสนอแก่ผู้บริการ



บรรณานุกรม

นรินทร์ โอพารกิจอนันต์. 2547. **J2EE & XML** จาวาระดับองค์กร. กรุงเทพฯ : เติชชุกิดอทเน็ต.

สรารุช อ้อยศรีสกุล. 2544. **ถอดรหัส. NET+Web Services**. กรุงเทพฯ: วิตตี้ กรู๊ป.

Heather Kreger. 2001. **Web Service Conceptual Architecture (WSCA1.0)**. [Online].

Available : <http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>.

Sun Microsystems. 2004. **The Java™ Web Services Tutorial**.

[Online]. Available : <http://java.sun.com/webservices/docs/1.4/tutorial/doc/index/html>.

W3.org. 2001. **Web Service Description Guide Thailand (WSDL) 1.1**. [Online]. Available :

<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.



ประวัติผู้เขียน

ชื่อผู้เขียน

นายรังสรรค์ สาริกะภูติ

วุฒิการศึกษาระดับปริญญาตรี

วท.บ. (วิทยาการคอมพิวเตอร์)

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร

ลาดกระบัง

การทำงาน

บริษัท โพรเกรส ซอฟต์แวร์ จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้