

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ.

การพัฒนาเว็บไซต์ระบบโรงแรมโดยใช้เทคโนโลยี
เอ็นเตอร์ไพร์สจาวาเบินส์

Development of Web Based Hotel System Using Enterprise JavaBeans
Technology

โดย

นายจักรินทร์ วิทยุศิลป์

รหัส 45066012



H002170

อาจารย์ที่ปรึกษา

ดร. พรฤดี เนติโสภาคกุล

วัน เดือน ปี.....	0 6 ก.พ. 2550
เลขทะเบียน.....	02170
เลขเรียกหนังสือ.....	วพ.ค. 2546
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจธ."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาเว็บไซต์ระบบโรงแรมโดยใช้เทคโนโลยี อินเทอร์เน็ตไร้สาย
นักศึกษา	นายจักรินทร์ วัฒนศิริศิลป์
อาจารย์ที่ปรึกษา	ดร. พรฤดี เนติโสภาคกุล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ในปัจจุบันธุรกิจโรงแรมเป็นธุรกิจที่ต้องมีการแข่งขันสูง อีกทั้งรัฐบาลได้มีนโยบายที่ต้องการให้นักท่องเที่ยวทั้งใน และนอกประเทศได้เข้ามาท่องเที่ยวในประเทศไทย ทำให้ธุรกิจโรงแรมต้องมีการปรับเปลี่ยนแนวทางการให้บริการเพื่อตอบสนองนโยบายของรัฐบาล รวมถึงการเพิ่มประสิทธิภาพในการให้บริการแก่นักท่องเที่ยวที่เพิ่มขึ้น แนวทางหนึ่งที่สามารถทำได้คือการจัดทำเว็บไซต์ที่สามารถทำการจองห้องพัก หรือการตรวจสอบราคาห้องพัก ซึ่งสามารถเข้าใช้บริการได้ทุกที่ที่มีอินเทอร์เน็ต แนวทางนี้จะส่งผลให้นักท่องเที่ยวทั่วโลกสามารถเข้ามาใช้บริการของโรงแรมได้ และยังช่วยเพิ่มผลกำไรให้กับทางโรงแรมอีกเช่นกัน

Title	Development of Web Based Hotel System Using Enterprise JavaBeans Technology
Student	Mr. Jugarin Viboonsin
Advisor	Dr. Ponrudee Netisopakul
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

ABSTRACT

Hotel business has strong competition. Moreover, Thai government promoted tourism by implementing policies to attract tourist from inside and outside country. Therefore, hotel business has to improve administrative operation to support government policies and to increase service efficiency. One approach is to implement a web site for general information and reservation system. This paper develops web based hotel system for the hotel business using Enterprise JavaBeans Technology.

กิตติกรรมประกาศ

การดำเนินการโครงการพัฒนาระบบงานฉบับนี้ได้สำเร็จลุล่วงเป็นอย่างดี เนื่องจากได้รับความกรุณาจากอาจารย์ที่ปรึกษา ดร.พรฤดี เนติโสภาคกุล ในการให้คำปรึกษา ให้คำแนะนำ และให้ความรู้แก่ผู้เขียนมาโดยตลอด อีกทั้งยังช่วยกรุณาตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ในแต่ละขั้นต้นในการดำเนินงานเป็นอย่างดี ผู้เขียนขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูง

ขอขอบคุณเพื่อนร่วมงานทุก ๆ ท่าน และเพื่อนร่วมรุ่นในการให้การสนับสนุน ให้ความช่วยเหลือ และเป็นกำลังใจในการศึกษาและดำเนินงานมาเป็นอย่างดี

ผู้เขียนขอกราบขอบพระคุณบิดา มารดา และญาติ ๆ ทุกท่าน ที่ได้อบรม และให้การสนับสนุนในทุก ๆ เรื่อง อีกทั้งยังเป็นกำลังใจให้โดยตลอด จนทำให้โครงการพัฒนาระบบงานนี้สำเร็จลุล่วงไปได้อย่างสมบูรณ์

นายจักรินทร์ วิบูลย์ศิลป์

กุมภาพันธ์ 2546

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่	
1. บทนำ.....	1
1.1 ปัญหา การวิเคราะห์ และการออกแบบระบบ.....	1
1.2 วัตถุประสงค์ของการศึกษา และออกแบบ.....	1
1.3 ขอบเขต และวิธีการศึกษา.....	2
1.4 แผนการดำเนินการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
2. เทคโนโลยีที่ใช้ในระบบ.....	3
2.1 เอ็นเตอร์ไพรส์จาวาปิ่นส์.....	3
2.2 Struts Framework.....	14
3. การวิเคราะห์ และออกแบบระบบ.....	16
3.1 Use Case Diagram.....	16
3.2 Use Case Specification.....	17
3.3 Class Diagram.....	36
3.4 Sequence Diagram.....	50
3.5 เครื่องมือที่เลือกใช้ในการพัฒนาระบบ.....	72
4. การวิเคราะห์ และออกแบบระบบฐานข้อมูล.....	73
4.1 ลักษณะการออกแบบฐานข้อมูล.....	73

สารบัญ (ต่อ)

	หน้า
บทที่	
4.2 Data Dictionary.....	75
5. การพัฒนาระบบงาน.....	85
5.1 การออกแบบหน้าจอ.....	85
6. สรุป และข้อเสนอแนะ.....	117
6.1 ข้อเสนอแนะ.....	117
6.2 ประโยชน์ที่คาดว่าจะได้รับ.....	118
บรรณานุกรม.....	119
ประวัติผู้เขียน.....	120

สารบัญตาราง

หน้า

ตารางที่

4.1 ตาราง RATE_TYPE.....	75
4.2 ตาราง SEASON.....	75
4.3 ตาราง RATE_PRICE.....	76
4.4 ตาราง RATE.....	76
4.5 ตาราง RATEPLAN.....	77
4.6 ตาราง SELL_SEQUENCE.....	77
4.7 ตาราง STATUS_TYPE.....	77
4.8 ตาราง PRICE_TYPE.....	78
4.9 ตาราง CREDIT_CARD.....	78
4.10 ตาราง POLICY.....	78
4.11 ตาราง POLICY_CREDIT_CARD.....	79
4.12 ตาราง ROOMTYPE.....	79
4.13 ตาราง BEDTYPE.....	80
4.14 ตาราง ROOM_STATUS.....	80
4.15 ตาราง RESERVATION.....	81
4.16 ตาราง GUEST_INFO.....	82
4.17 ตาราง AVERAGE_DAILY_RATE.....	83
4.18 ตาราง RESERVE_RATE.....	84
4.19 ตาราง RESERVE_SEASON.....	84

สารบัญรูป

หน้า

รูปที่

2.1 ระบบ J2EE based.....	3
2.2 EJB component.....	9
2.3 Bean instances.....	10
2.4 การติดต่อผ่านระบบเครือข่าย.....	10
2.5 แสดงตัวอย่างการทำงานเมื่อ Client เรียกใช้ EJB.....	11
2.6 Struts Architecture.....	14
3.1 Use Case Diagram ของ Hotel System.....	16
3.2 Class Diagram ใน Property Package (Business Service Layer).....	37
3.3 Class Diagram ใน Rate Plan Package (Business Service Layer).....	38
3.4 Class Diagram ใน Report Package (Business Service Layer).....	39
3.5 Class Diagram ของ package reservation (Business Service Layer).....	40
3.6 Class Diagram ของ Reservation Package (Business Service Layer) (ต่อ).....	40
3.7 Class Diagram ในกลุ่มของ Manager (Business Service Layer).....	41
3.8 Class Diagram ในกลุ่มของ Data Access Object (Business Service Layer).....	42
3.9 Class Diagram ในกลุ่มของ EJB Helper (Business Service Layer).....	43
3.10 Class Diagram ในกลุ่มของ Action Form (Common Business Object Layer)..	44
3.11 Class Diagram ใน Common Package (Common Business Object Layer).....	45
3.12 Class Diagram ใน Property Package (Common Business Object Layer).....	46
3.13 Class Diagram ใน Rate Plan Package (Common Business Object Layer).....	47
3.14 Class Diagram ใน Report Package (Common Business Object Layer).....	48
3.15 Class Diagram ใน Reservation Package (Common Business Object Layer)...	49
3.16 View Rate Plan Sequence Diagram.....	51
3.17 View Rate Plan Sequence Diagram (ต่อ).....	52

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.18	Make Reservation Sequence Diagram : Make Stay Info.....	54
3.19	Make Reservation Sequence Diagram : Make Stay Info (ต่อ).....	55
3.20	Make Reservation Sequence Diagram : Select Room Availability.....	57
3.21	Make Reservation Sequence Diagram : Make Guest Info.....	59
3.22	Make Reservation Sequence Diagram : Confirm Reservation.....	61
3.23	Make Reservation Sequence Diagram : Confirm Reservation (ต่อ).....	62
3.24	Manage Room Type Sequence Diagram : List Room Type.....	64
3.25	Manage Room Type Sequence Diagram : List Room Type (ต่อ).....	65
3.26	Manage Room Type Sequence Diagram : Add Room Type.....	67
3.27	Manage Room Type Sequence Diagram : Add Room Type (ต่อ).....	68
3.28	Manage Room Type Sequence Diagram : Edit Room Type.....	70
3.29	Manage Room Type Sequence Diagram : Edit Room Type (ต่อ).....	71
4.1	Database Model Diagram.....	74
5.1	หน้าจอ Login ของผู้ดูแลระบบ.....	85
5.2	หน้าจอ Menu เมื่อเข้าสู่ระบบ.....	86
5.3	หน้าจอ Menu เพื่อเข้าไปจัดการข้อมูลในส่วนของ Property.....	87
5.4	แสดงหน้าจอแรกในส่วนของการจัดการข้อมูลของ Room Type.....	88
5.5	แสดงหน้าจอเมื่อต้องการเพิ่ม Room Type.....	89
5.6	แสดงหน้าจอเมื่อต้องการแก้ไขข้อมูลของ Room Type.....	90
5.7	หน้าจอ Menu เพื่อเข้าไปจัดการข้อมูลในส่วนของ Rate Plan.....	91
5.8	แสดงหน้าจอเพื่อจัดการ Sell Sequence ของ Room Type.....	92
5.9	แสดงหน้าจอเพื่อจัดการ Season ของระบบ.....	93
5.10	แสดงหน้าจอเพื่อจัดการ Rate ของแต่ละห้องในแต่ละ Season.....	94

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.11 หน้าจอ Menu เพื่อเข้าไป Update ข้อมูลเกี่ยวกับการจอง.....	95
5.12 แสดงหน้าจอ Update Inventory จากระบบอื่น.....	96
5.13 แสดงหน้าจอขึ้นชั้นการ Update Inventory จากระบบอื่น.....	97
5.14 แสดงหน้าในการค้นหาการจองของ Customer เพื่อทำการแก้ไขข้อมูล.....	98
5.15 แสดงหน้าในการแก้ไขข้อมูลของ Customer.....	99
5.16 หน้าจอ Menu เพื่อแสดง Report ต่าง ๆ.....	100
5.17 แสดงหน้าจอของ Reservation Report.....	101
5.18 แสดงหน้าจอของ Reservation Report Detail.....	102
5.19 แสดงหน้าจอของ Monthly Summary Report.....	103
5.20 แสดงหน้าจอของ Room Inventory Report.....	104
5.21 แสดงหน้าจอของ Arrival Report.....	105
5.22 แสดงหน้าจอแรกในส่วนของ Customer.....	106
5.23 แสดงหน้าจอ Menu ของ Customer.....	106
5.24 แสดงหน้าจอ View Rate ของ Customer.....	107
5.25 แสดงหน้าจอ Stay Information ของ Customer.....	108
5.26 แสดงหน้าจอ Room Availability ของ Customer.....	109
5.27 แสดงหน้าจอ Guest Information ของ Customer.....	107
5.28 แสดงหน้าจอ Confirmation ของ Customer.....	108
5.29 แสดงหน้าจอการจองห้องเสร็จสมบูรณ์ของ Customer.....	112
5.30 แสดงหน้าจอการค้นหาข้อมูลการจอง.....	113
5.31 แสดงหน้าจอผลลัพธ์ของการค้นหาข้อมูลการจอง.....	114
5.32 แสดงหน้าจอเมื่อ Customer เลือกที่จะแก้ไขข้อมูล หรือยกเลิกการจอง.....	115

สารบัญรูป (ต่อ)

หน้า

รูปที่

5.33	แสดงหน้าจอการแก้ไขการจองห้องเสร็จสมบูรณ์ของ Customer.....	116
------	---	-----



บทที่ 1

บทนำ

1.1 ปัญหา การวิเคราะห์ และออกแบบระบบ

สังคมในปัจจุบันเป็นสังคมที่เรียกว่า สังคมสารสนเทศ (Information Society) มีการคมนาคม การสื่อสารที่เจริญ และมีการไหลเวียนของข่าวสารมากขึ้น เนื่องจากระบบสื่อสารที่รวดเร็ว จึงทำให้สามารถได้รับข้อมูลข่าวสารจากซีกโลกหนึ่งไปยังอีกซีกโลกหนึ่งได้ ระบบสื่อสารที่ว่าเป็นส่วนสำคัญส่วนหนึ่งของระบบสารสนเทศ (Information System) ในขณะเดียวกันการแข่งขันในวงธุรกิจก็มีความเข้มข้นมากขึ้น จึงต้องอาศัยเครื่องมือช่วยกลั่นกรองข้อมูลข่าวสารประมวลผลอย่างเป็นระบบ และสามารถนำไปใช้ได้อย่างรวดเร็ว

หากจะมองถึงนโยบายหนึ่งของทางรัฐบาลในขณะนี้ นั่นคือ Unseen in Thailand ที่เร่งส่งเสริมการท่องเที่ยวในประเทศไทย ซึ่งจะส่งผลโดยตรงถึงธุรกิจโรงแรม ทำให้ต้องมีการปรับเปลี่ยนแนวทางการให้บริการเพื่อตอบสนองนโยบายของรัฐบาล รวมถึงการเพิ่มประสิทธิภาพในการให้บริการแก่นักท่องเที่ยวที่เพิ่มขึ้น

แนวทางหนึ่งที่สามารถช่วยเพิ่มผลกำไรให้กับทางโรงแรมได้นั้นคือ จัดทำเว็บไซต์ที่ให้ความสะดวกสบายแก่ผู้ที่เข้ามาใช้บริการ ทั้งในเรื่องของการตรวจสอบราคาห้องพัก การสำรองห้องพัก ซึ่งสามารถเข้าใช้บริการได้ทุกมุมโลกผ่านเครือข่ายอินเทอร์เน็ต ทำให้นักท่องเที่ยวที่ต้องการเข้ามาท่องเที่ยวในประเทศไทยสามารถที่จะเข้ามาใช้บริการของทางโรงแรมได้ และยังช่วยเพิ่มผลกำไรให้กับทางโรงแรมอีกเช่นกัน

1.2 วัตถุประสงค์ของการศึกษา และออกแบบ

1.2.1 เพื่อออกแบบ และพัฒนาระบบโรงแรมผ่านระบบอินเทอร์เน็ต ซึ่งช่วยอำนวยความสะดวกแก่นักท่องเที่ยวทั้งใน และนอกประเทศ

1.2.2 เพื่อสร้างระบบสารสนเทศที่มีความพร้อม และสะดวกในการเข้าถึงข้อมูล

1.2.3 เพื่อสนับสนุนการนำเทคโนโลยีอินเทอร์เน็ตไปสร้างเป็นสัมมาประยุกต์ใช้งาน

1.2.4 เพื่อนำระบบสารสนเทศมาเพิ่มศักยภาพในงานด้านการบริการ

1.3 ขอบเขต และวิธีการศึกษา

การพัฒนาเว็บไซต์ระบบโรงแรมโดยใช้เทคโนโลยีอินเทอร์เน็ตโพรสจาวาบีเอส เป็นการพัฒนาเว็บไซต์ในการให้บริการต่าง ๆ ของโรงแรมเช่นการเข้ามาชมประเภท และราคาห้องพัก การสำรองห้องพัก เป็นต้น โดยจะแบ่งกลุ่มของการใช้งานออกเป็น 2 กลุ่มคือ กลุ่มของผู้ใช้บริการทั่วไป และกลุ่มของผู้ดูแลระบบ ซึ่งเทคโนโลยีที่ใช้ในการเข้าถึงข้อมูลนั้นจะใช้เทคโนโลยีของจาวาที่เรียกว่า อินเทอร์เน็ตโพรสจาวาบีเอส และในส่วนของเว็บแอปพลิเคชันได้ใช้เทคโนโลยีของ Java Server Pages (JSP) ร่วมกับ Struts framework

1.4 แผนการดำเนินการศึกษา

- 1.4.1 เก็บข้อมูลเกี่ยวกับการจัดการทางด้านโรงแรม
- 1.4.2 ศึกษาข้อมูลเพื่อเลือกฟังก์ชันการทำงานที่เหมาะสม
- 1.4.3 วิเคราะห์ และออกแบบระบบ
- 1.4.4 ศึกษากระบวนการจัดการฐานข้อมูลอราเคิล 9i เพื่อนำมาใช้เป็นฐานข้อมูลของระบบ และศึกษาวิธีที่จะนำอินเทอร์เน็ตโพรสจาวาบีเอส เข้ามาติดต่อ และดึงข้อมูลจากระบบฐานข้อมูล
- 1.4.5 พัฒนาเว็บไซต์ระบบโรงแรม โดยใช้เทคโนโลยีอินเทอร์เน็ตโพรสจาวาบีเอส
- 1.4.6 ทดสอบ และแก้ไขโปรแกรม ตลอดจนปรับปรุงฐานข้อมูล
- 1.4.7 สรุปผล และข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

การพัฒนาเว็บไซต์ระบบโรงแรมโดยใช้เทคโนโลยีอินเทอร์เน็ตโพรสจาวาบีเอส เป็นการนำเอาความรู้ความเข้าใจในการบริหารงานระบบโรงแรม เพื่ออำนวยความสะดวกแก่นักท่องเที่ยวทั้งใน และนอกประเทศ ซึ่งนักท่องเที่ยวสามารถเข้ามาตรวจสอบห้องพัก และราคาห้องพัก หรือการสำรองห้องพักได้อย่างง่ายดาย สะดวก และรวดเร็ว ทั้งนี้ยังเป็นการเพิ่มทางเลือกในการเข้ามาใช้บริการของทางโรงแรม ส่งผลให้โรงแรมมีผลกำไรที่เพิ่มขึ้น จากจำนวนนักท่องเที่ยวที่เพิ่มขึ้นนั่นเอง

ในส่วนของผู้พัฒนานั้นเป็นการเพิ่มทางเลือก และวิธีการจัดการกับข้อมูลในฐานข้อมูล โดยใช้เทคโนโลยีของจาวาที่เรียกว่า อินเทอร์เน็ตโพรสจาวาบีเอส ที่ช่วยในการเข้าถึงข้อมูล รวมถึงได้ใช้ Framework ที่มีประสิทธิภาพในการพัฒนาเว็บไซต์นั้นคือ Struts framework ซึ่งทำงานร่วมกับ Java Server Pages (JSP)

บทที่ 2

เทคโนโลยีที่ใช้ในระบบ

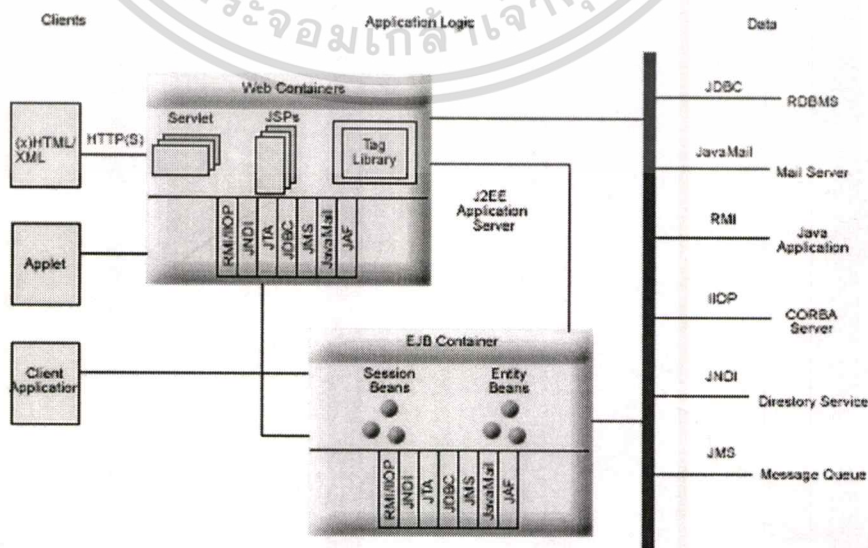
2.1 เอนเตอร์ไพรส์จาวาเบินส์ (วีระศักดิ์ ชิงदार และคณะ. 2546.)

2.1.1 ความเป็นมาของเอนเตอร์ไพรส์จาวาเบินส์

Java 2 Platform, Enterprise Edition ถูกเสนอขึ้นเพื่อลดความยุ่งยากในการสร้างโปรแกรมในระดับ enterprise โดยกำหนดมาตรฐานสำหรับโปรแกรม รวมทั้งสภาพแวดล้อมสำหรับการทำงาน รวมเรียกว่า J2EE architecture ประกอบด้วยข้อกำหนดของ application server ซึ่งมีบริการมาตรฐานสนับสนุนการทำงานของ application ทำให้ผู้เขียนโปรแกรมสามารถมุ่งความสนใจไปที่ business logic ไม่ต้องสร้างโปรแกรมเกี่ยวกับบริการ และการติดต่อกับระบบอื่นเอง

J2EE servers มีลักษณะคล้ายกับ application servers ของระบบ web-based และระบบ n-tier แต่มีข้อกำหนดว่า J2EE servers ต้องสร้างสภาพแวดล้อมสำหรับให้ J2EE application ทำงาน เรียกว่า J2EE containers และต้องมีบริการเกี่ยวกับ transactions, instance management, persistence และ security รวมทั้งบริการเกี่ยวกับการติดต่อทาง network กับระบบอื่น

J2EE servers ต้องมี J2EE containers อย่างน้อย 2 ประเภท คือ web container สำหรับทำงานเกี่ยวกับ servlet/jsp และ EJB container สำหรับทำงานเกี่ยวกับ Enterprise JavaBeans (EJB) ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ระบบ J2EE based

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 J2EE Application Components

Component คือผลลัพธ์หน่วยหนึ่งของการคอมไพล์โปรแกรมบางส่วน ซึ่งเป็นไฟล์ ๆ หนึ่งที่สามารถนำมาใช้งานร่วมกับโปรแกรมอื่นได้โดยไม่ต้องถูกคอมไพล์ใหม่ จึงสะดวกในการนำไปใช้งาน

J2EE application components คือ components ประเภทหนึ่ง แบ่งได้เป็น 2 แบบคือ

2.1.2.1 Web Components คือ components ที่ต้องทำงานในสภาวะแวดล้อมที่เรียกว่า web container สามารถรับและโต้ตอบกับ HTTP requests แบ่งออกเป็น 2 ประเภทคือ

Servlets เป็นโปรแกรม components ที่ต้องทำงานในสภาวะแวดล้อมที่เรียกว่า web container ซึ่งอาจจะสร้างขึ้นโดย web server หรือ J2EE servers เรามักใช้ servlet เป็นตัวกลางที่รับ HTTP requests มาจาก browsers แล้วเรียกต่อไปที่ enterprise javabeans เมื่อได้คำตอบกลับมาแล้วจะสร้างเป็น html/xml และส่งกลับไปที่ browsers ด้วยวิธีนี้จะทำให้ browsers สามารถเรียกใช้ enterprise javabeans ได้

Java Server Pages (JSP) เป็นกลไก สำหรับแทรก java โปรแกรมลงใน html ซึ่งเมื่อตอนถูกเรียกใช้ จะถูกเปลี่ยนเป็น servlets ดังนั้น JSP จะสามารถรับ HTTP requests และตอบกลับเป็น html/xml ได้คล้ายกลับ servlet แต่เราสามารถสร้าง html/xml ได้ง่ายกว่าการเขียนโปรแกรม servlets ให้สร้าง page โดยตรง โดยทั่วไปเรามักใช้ JSP ทำหน้าที่ presentation logic ส่วนงาน business logic ถูกส่งต่อไปให้ enterprise javabeans ทำ

2.1.2.2 Enterprise JavaBeans (EJB) เป็น components ที่จะถูกสร้างและทำงานในสภาวะแวดล้อมที่เรียกว่า EJB container ซึ่งสร้างขึ้นโดย J2EE servers

ภาษา Java มี model ของโปรแกรมที่เรียกว่า JavaBean สำหรับสร้าง components ที่จะถูกนำไปใช้อีกได้ EJB เป็น components ประเภทหนึ่งที่สร้างขึ้นตาม model ขึ้นตัวของ java bean แต่มีข้อกำหนดมากกว่าเพื่อที่จะทำงานได้ใน J2EE servers และเหมาะสำหรับเป็นโปรแกรมที่ทำงานธุรกิจ สรุปได้ว่า java bean มีข้อแตกต่างจาก EJB ดังนี้

- java bean เป็น components ที่สามารถทำงานได้ใน JVM คล้ายกับโปรแกรม java แต่ EJB จะต้องทำงานอยู่ใน EJB container ของ J2EE servers เท่านั้น
- java bean มี methods ที่เรียกว่า accessor / mutator สำหรับใช้อ่านหรือเขียนค่าที่เป็น properties แต่ EJB ต้องมี remote interface สำหรับ methods ที่เป็น accessor/mutator
- java bean มี event model สำหรับรับและส่ง event แต่ EJB จะติดต่อกับ EJB อื่น โดยการเรียก methods ผ่านทาง remote/local interfaces ไม่ใช่ event โดยตรง

2.1.3 EJB Containers

EJB Containers คือโปรแกรมที่สร้างสถานะแวดล้อมสำหรับให้ EJB ทำงาน และมีบริการมาตรฐานสำหรับสำหรับให้ EJB เรียกใช้งานได้ ข้อกำหนดของ J2EE กำหนดว่า EJB containers ต้องไม่ถูกมองเห็นจาก clients หมายความว่า client สามารถเข้าถึง EJB ได้ โดยเหมือนกับว่า clients ไม่จำเป็นต้องทราบถึงการมีอยู่ของ EJB containers เลย ด้วยข้อกำหนดแบบนี้จะทำให้ clients สามารถเข้าถึง EJB ใด ๆ ด้วยวิธีเหมือนกันหมด ไม่ว่าจะเป็น J2EE server ที่ผลิตโดยบริษัทใด บริการมาตรฐานที่ EJB containers ต้องมีให้แก่ EJB คือ

2.1.3.1 Transaction Controls

โดยปกติการควบคุม transaction จะใช้ begin กับ end ล้อมลำดับของประโยคที่ต้องถูกทำให้เสร็จทั้งหมด หากประโยคใดประโยคหนึ่งล้มเหลวจะต้องล้มเลิกทั้งหมด เราสามารถเขียนโปรแกรมให้ EJB ควบคุม transaction เอง แต่จะเป็นงานที่ค่อนข้างยุ่งยาก ดังนั้น EJB containers จึงมีบริการ transaction controls ให้แก่ EJB โดย bean providers สามารถระบุ transactions แบบที่ EJB นั้นต้องการได้แบบ declarative คือไม่ได้ระบุลงในโปรแกรมของ EJB แต่ระบุอยู่ใน deployment descriptor ทำให้ application assembler สามารถนำ EJB นั้นไปใช้ใน applications ที่แตกต่างกันโดยมี transaction ตามที่ต้องการได้ EJB Containers มี transaction controls 6 แบบให้เลือกใช้คือ

Not supported หมายถึง EJB นั้นไม่สนับสนุน transactions และไม่สามารถถูกเรียกใช้จากใน transaction ที่มีอยู่ก่อน หาก client เรียกใช้ EJB นี้จากใน transaction ที่มีอยู่ก่อน transaction นั้นจะถูกหยุดไว้ก่อน จนเมื่อ EJB นั้นทำงานเสร็จ transaction ก็จะเริ่มต่อไป

Bean Managed หมายถึง EJB นั้นจะทำการควบคุม transaction เอง

Required ถ้า client เรียกใช้ EJB จาก transaction ที่มีอยู่ก่อน จะทำให้ EJB นั้นทำงานอยู่ใน transaction context ของ client แต่ถ้า client ไม่มี transaction มาก่อน EJB จะต้องสร้าง transaction ใหม่ขึ้นมาเอง

Supports ถ้า client เรียกใช้ EJB จาก transaction ที่มีอยู่ก่อน จะทำให้ EJB นั้นทำงานอยู่ใน transaction context ของ client แต่ถ้า client ไม่มี transaction มาก่อน EJB จะไม่สร้าง transaction ใหม่

Required new หมายถึง EJB จะสร้าง transaction ใหม่เสมอ หาก client เรียกใช้ EJB จาก transaction ที่มีอยู่ก่อน จะทำให้ transaction เดิมนั้นถูกหยุดไว้จนกว่า EJB จะทำงานเสร็จ

Mandatory หมายถึง client จะต้องมีการมี transaction เปิดอยู่ก่อนจะเรียกใช้ EJB นี้ หากไม่มีจะเกิด error ขึ้นตอนโปรแกรมทำงาน

2..1.3.1.1 Isolation Level

ในการทำ transaction ย่อมต้องเกิดกรณีที่มีการใช้ข้อมูลพร้อมกัน ดังนั้นจึงอาจจะเกิดปัญหาดังต่อไปนี้

Dirty read จะเกิดขึ้นเมื่อ transaction หนึ่งสามารถอ่าน uncommitted change ซึ่งทำโดย transaction ที่ 2 ได้ ปัญหาคือถ้า transaction ที่ 2 rollback ข้อมูลที่ transaction แรกอ่านไปจะ invalid เพราะการ rollback จะ undo การเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูล

Nonrepeatable read จะเกิดขึ้นเมื่อข้อมูลที่ transaction หนึ่งอ่านได้ในภายหลังมีค่าไม่เท่ากับข้อมูลที่อ่านได้ในครั้งแรก ซึ่งสาเหตุที่ทำให้การอ่านข้อมูลในครั้งแรกและครั้งหลังไม่เท่ากันนั้นมาจากการที่ transaction อื่นสามารถเปลี่ยนแปลงข้อมูลที่ transaction แรกอ่านไปก่อนที่ transaction จะสิ้นสุดลง

Phantom read จะเกิดขึ้นเมื่อ new record ที่ถูก insert ลงฐานข้อมูลโดย transaction หนึ่งสามารถถูกอ่านได้จาก transaction ที่ 2 ซึ่งเริ่มต้นก่อน transaction แรก ปัญหาคือการค้นหาข้อมูลของ transaction ที่สองในครั้งแรกและครั้งหลังจะมีค่าต่างกัน เพราะในครั้งหลังผลของการค้นหาวouldรวม record ที่ transaction ที่หนึ่งเพิ่งจะ insert ลงไปด้วย

ในการจัดการปัญหาเหล่านี้ EJB จึงได้สร้าง Isolation Level ขึ้นมา และแบ่งออกเป็น 4 ระดับ โดยมีรายละเอียดดังนี้

Read uncommitted จะยอมให้ transaction สามารถอ่าน uncommitted change ของ transaction อื่นได้ ดังนั้น Dirty read, Nonrepeatable read และ Phantom read สามารถเกิดขึ้นได้ isolation level นี้ใช้ได้เฉพาะ transaction ที่อ้างถึงข้อมูลที่เป็น read-only

Read committed จะยอมให้ transaction หนึ่งสามารถอ่านข้อมูลของ transaction อื่นได้เมื่อ transaction อื่นนั้นได้ commit ข้อมูลนั้นหรือ rollback แล้ว ทำให้ dirty read จะไม่เกิดขึ้น แต่ Nonrepeatable read และ Phantom read ยังสามารถเกิดขึ้นได้ ฐานข้อมูลส่วนใหญ่จะกำหนดให้ isolation level นี้เป็น default

Repeatable read จะไม่ยอมให้ transaction หนึ่งอ่านข้อมูลของ transaction อื่นจนกว่า transaction อื่นนั้นจะ commit หรือ rollback สำหรับ transaction ที่ใช้ isolation level นี้จะมั่นใจได้ว่าการอ่านข้อมูลแต่ละครั้ง ค่าที่ได้จะเท่ากัน ดังนั้น Dirty read และ Nonrepeatable read จะไม่เกิดขึ้น แต่ Phantom read จะยังสามารถเกิดขึ้นได้

Serializable จะมีข้อกำหนดที่เข้มงวดที่สุด กล่าวคือในขณะที่ transaction ยังไม่ commit หรือ rollback transaction อื่นจะไม่สามารถอ่านหรือเขียนข้อมูลที่ transaction นั้นอ้างถึงได้ ดังนั้น Dirty read, Nonrepeatable read และ Phantom read จะไม่เกิดขึ้น

2.1.3.2 Management of Multiple Instances

EJB มักจะถูกเขียนให้เป็นโปรแกรม single-threaded คือที่เวลาหนึ่งจะถูกใช้งานโดย client หนึ่งเท่านั้น อย่างไรก็ตามมี EJB บางประเภท เช่น stateless session bean ที่อาจถูกใช้งานจากหลาย ๆ clients พร้อมกัน จึงเป็นหน้าที่ของ EJB containers ที่จะต้องรักษาจำนวน EJB สำหรับรองรับการทำงาน of clients ให้พอเหมาะ เพื่อไม่ให้สิ้นเปลืองทรัพยากรของระบบมากเกินไป J2EE มีข้อกำหนดว่า EJB containers ต้องสามารถให้บริการเหล่านี้ได้

Instance Passivation หาก J2EE server ต้องการใช้ทรัพยากรบางอย่าง เช่น หน่วยความจำ หรือ database connection แต่ขณะนั้นไม่สามารถใช้งานได้ เนื่องจากมี EJB บางตัวจับจองอยู่ กรณีเช่นนี้ EJB containers อาจเลือกทำ passivation EJB (หรือเรียกว่า swap out) บางตัวที่ไม่ถูกใช้งานในตอนนั้น คือเขียนค่าสถานะของ EJB นั้นไปที่หน่วยความจำถาวร เพื่อนำทรัพยากรที่ EJB นั้นจับจองอยู่ไปใช้งานอื่นก่อน จนเมื่อ EJB นั้นถูกเรียกใช้งานอีก จึงสร้าง instance ของ EJB นั้นแล้วอ่านค่าจากหน่วยความจำถาวรมากำหนดเป็นสถานะของ EJB จึงสามารถทำงานต่อจากเมื่อตอนก่อนถูกทำ passivation สังเกตว่า EJB containers จะทำ passivation EJB เองเมื่อจำเป็นโดยอัตโนมัติ แต่สำหรับ entity bean จะมีกลไกให้ผู้เขียนโปรแกรมควบคุมการทำ passivation ด้วย

Instance Pooling หมายถึง EJB container ต้องสามารถควบคุมให้ clients หลาย ๆ ตัวใช้งาน EJB ตัวเดียวกันได้ (ยกเว้น statefull session bean จะถูกใช้งานโดย client เดียว) เพื่อไม่ต้องให้ J2EE server เสียเวลาสร้าง และทำลาย instance ของ EJB ที่ถูกใช้งานบ่อย ๆ EJB containers จะสร้าง instance pool สำหรับเก็บ instance ของ EJB ที่ต้องถูกใช้งานบ่อย ๆ เมื่อมี client ขอใช้บริการ EJB นั้น ก็จะถูกดึงออกจาก pool มาทำงาน และเมื่อทำงานเสร็จแล้วก็เก็บกลับเข้า pool เพื่อถูกใช้อีกในภายหลัง

Database Connection Pooling เป็นการสร้าง pool สำหรับเก็บ database connection คล้ายกับ instance pooling ทำให้เมื่อ EJB หนึ่งต้องการติดต่อกับ database server ก็ไม่ต้องสร้าง connection เอง แต่สามารถขอ connection จาก pool ไปใช้ และเมื่อใช้เสร็จก็นำมาคืน connection pool ช่วยให้โปรแกรมทำงานเร็วขึ้นอย่างมาก เนื่องจาก EJB มักจะใช้ database connection เพียงช่วงเวลาสั้น ๆ แต่บ่อย ๆ หากต้องสร้างและทำลาย database connection ในทุกครั้งที่อ้างถึง database จะเสียเวลามาก

2.1.3.3 Catching

มี EJB บางประเภท เช่น entity beans ที่ต้องอ่านข้อมูลจาก database มาเป็นค่าสถานะของมัน และอาจจะต้องเขียนค่าสถานะนั้นกลับไปใน database เพื่อเก็บไว้อย่างถาวรบ่อย ๆ กรณีนี้ EJB containers ต้องสามารถทำ catching ข้อมูลเหล่านั้นเพื่อเพิ่มประสิทธิภาพในการทำงาน เนื่องจากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านหรือเขียนข้อมูลจาก database หรือไฟล์ หากทำเพียงจำนวนเล็กน้อยแต่บ่อย ๆ จะเสียเวลามากกว่าทำจำนวนมากแต่น้อยครั้ง EJB containers จึงควรมีหน่วยความจำเป็นตัวกลางสำหรับรองรับการอ่าน หรือเขียนบ่อย ๆ

2.1.3.4 Method Invocation Optimization

ข้อกำหนดของ EJB รุ่น 1.x ระบุว่า EJB จะต้องมีการมี remote interface สำหรับให้ clients เรียกใช้งานแบบ remote คือเรียกผ่านทาง rmi-iiop protocol ทำให้ EJB รุ่นเก่าต้องถูกเรียกใช้งานอย่าง remote เท่านั้น แม้ว่า clients อาจจะอยู่บนเครื่องเดียวกัน แต่เริ่มจาก EJB รุ่น 2.0 มีทางเลือกกว่า EJB อาจจะมี local interface เพิ่มขึ้นมา สำหรับให้ clients ที่อยู่บนเครื่องเดียวกันเรียกใช้ได้ด้วย ซึ่งจะใช้การเรียก method ในเครื่องเดียวกันที่ทำงานได้เร็วกว่า โดย EJB containers จะเป็นผู้เลือกวิธีให้ คือถ้า clients อยู่ต่างเครื่องจะใช้ rmi-iiop โดย EJB นั้นต้องมี remote interface แต่ถ้า clients อยู่เครื่องเดียวกันจะใช้ method call ธรรมดา โดย EJB นั้นต้องมี local interface

2.1.3.5 Support for Persistence

Persistence คือ กลไกสำหรับทำให้วัตถุ (เช่น ตัวแปร หรือ EJB เป็นต้น) สามารถรักษาสถานะของมันในช่วงชีวิตไปได้ สมมติเช่น EJB ตัวหนึ่งมีค่าสถานะของมันค่าหนึ่ง แต่ EJB จะถูกทำลายด้วยเหตุผลบางอย่างหากเราต้องการคงค่าสถานะนั้นไว้ก็ต้องเขียนใส่หน่วยความจำเพื่อว่าเมื่อ EJB ถูกสร้างขึ้นมาก็สามารถอ่านค่าที่เขียนไว้มาเป็นสถานะของมันได้ ทำให้ EJB นี้สามารถทำงานต่อจากจุดก่อนถูกทำลายไปได้ EJB ประเภท entity beans มักต้องเป็น EJB ที่คงค่าสถานะของมันได้ เราอาจเขียนโปรแกรม entity beans ให้จัดการทำ persistence เองเรียกว่า *bean manage persistence* หรืออาจจะใช้บริการของ EJB containers คือให้ EJB containers จัดการให้ก็ได้ เรียกว่า *container manage persistence* ตามข้อกำหนดของ EJB รุ่น 1.x ไม่ได้ระบุว่า EJB containers ต้องสนับสนุน persistence แม้ว่า J2EE server ของบางบริษัทจะสามารถให้บริการนี้ได้ แต่นับจากรุ่น 2.0 ไปมีข้อบังคับว่า EJB containers จะต้องสนับสนุน persistence ของ EJB

2.1.3.6 Support for Access Control List

เนื่องจาก EJB อาจจะถูกเรียกใช้จาก clients ที่อยู่ใน J2EE server เดียวกันหรืออยู่ต่าง J2EE server ก็ได้ ถ้าเป็น clients ที่อยู่ใน J2EE server เดียวกันก็มักจะไม่มีปัญหาเกี่ยวกับการเรียกใช้ EJB โดยไม่มีสิทธิ์ เพราะ deployer เป็นผู้ควบคุมว่าจะมี components ใดถูก deployed ใน server นั้นบ้าง แต่ถ้าเป็น clients ที่อยู่ต่าง J2EE server จำเป็นต้องมีการตรวจสอบสิทธิ์การเข้าใช้งาน EJB โดย EJB containers จะมี access control list เก็บชื่อ และรหัสผ่านของผู้มีสิทธิ์ใช้งาน EJB ใน server นั้น deployer สามารถกำหนด access control list ให้แก่ EJB แต่ละตัว หรือแม้แต่บาง methods ของ EJB ตัวใดตัวหนึ่ง

2.1.4 EJB Components

ในต้นปี 2545 J2EE รุ่น 1.3.1 ใช้ข้อกำหนด EJB รุ่น 2.0 ซึ่งกำหนดว่า EJB มีอยู่ 3 ประเภท คือ session beans, entity beans และ message driven beans ในที่นี้จะขอกล่าวเฉพาะ session และ entity beans

Session beans คือส่วนของ session ที่เป็นตัวแทนในการเรียกใช้ application หรือฐานข้อมูลสำหรับกิจกรรมหนึ่ง ๆ อาจแบ่งออกเป็น 2 ประเภทคือ stateless และ statefull นอกจากนี้ session beans ยังอาจมี option กำหนดให้เป็น session synchronization ได้ด้วย

Entity beans คือตัวแทนของข้อมูลในฐานข้อมูลทำให้สามารถ insert, update และ delete ข้อมูลได้ อาจแบ่งออกเป็น 2 ประเภทคือ bean managed persistence และ container managed persistence

สำหรับ session beans และ entity beans แล้วทุก ๆ EJB components จะต้องประกอบด้วย

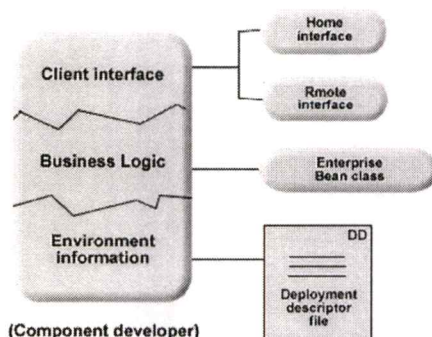
1. Client Interfaces เป็นข้อมูลให้ client อ้างถึง EJB นั้น ประกอบด้วย 2 interfaces คือ
 - *home interfaces* ระบุ methods เพื่อควบคุมวัฏจักรของ EJB
 - *remote interfaces* ระบุ business methods ของ EJB

Bean developers เป็นผู้กำหนด interfaces ทั้งสองนี้ ส่วนคลาสที่ implements interfaces เหล่านี้จะถูก bean container สร้างให้เมื่อตอน deployment

2. Business Logic เป็นคลาสของ EJB ที่ implements business logic methods ซึ่งกำหนดใน remote interface โดยปกติ bean developers เป็นผู้สร้างคลาสนี้พร้อม ๆ กับ client interfaces

3. Environment Information คือ deployment descriptor ซึ่งเป็นไฟล์ xml ที่ระบุข้อมูลสถานะแวดล้อมที่ต้องการให้ EJB นี้ทำงาน และบริการที่ EJB นั้นต้องการใช้ โดยปกติ bean developers เป็นผู้สร้าง deployment descriptor ก่อนไว้แล้ว แต่ไฟล์นี้อาจถูกเปลี่ยนแปลงได้โดย application assemblers และ deployers descriptor

ซึ่งทั้ง 3 ส่วนแสดงได้ดังในรูปที่ 2.2

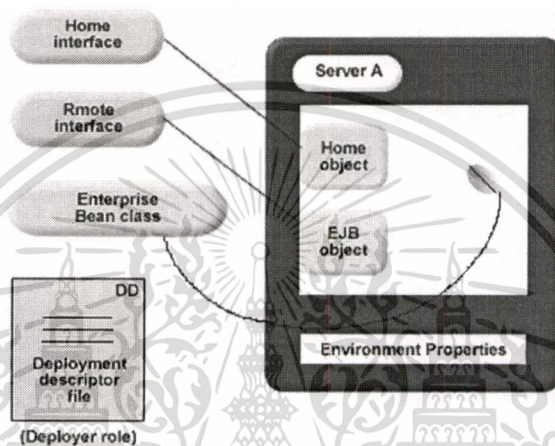


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 2.2 EJB component อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตอนที่ EJB ถูกใช้งานบน J2EE server นั้น EJB container จะสร้างคลาสที่ implements remote และ home interface แล้วสร้างเป็น interfaces ดังนี้

- Home object เป็น instance ของคลาสที่ implements home interface
- EJB object เป็น instance ของคลาสที่ implements remote interface

ส่วน interface ของคลาส EJB (เรียกว่า bean instance) จะถูกสร้างเมื่อต้องถูกใช้งาน คือเมื่อมี clients อ้างถึง ดังในรูปที่ 2.3

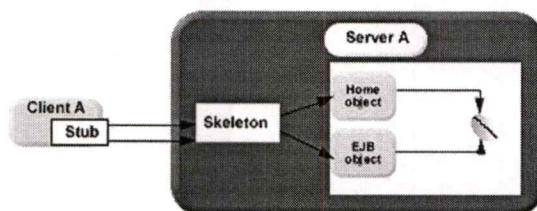


รูปที่ 2.3 Bean instances

หน้าที่ของ home object คือเป็นผู้ผลิตที่สามารถสร้างและทำลาย bean instance โดยที่ home object จะต้องถูกติดตั้งไว้กับ name service ให้ client เรียกขอ reference ไปใช้งานได้ ดังนั้น bean instance จะถูกสร้างขึ้นหรือถูกทำลายโดยขึ้นอยู่กับโปรแกรม client แต่ในกรณีที่มี clients จำนวนมากเรียกใช้งาน bean instance ตัวหนึ่ง และหาก bean instance ให้บริการไม่ทัน ก็ต้องมีการรอคอย EJB container สามารถสร้าง bean instance ของคลาส EJB นั้นเพิ่มขึ้นมาเพื่อช่วยทำงานก็ได้ เราเรียกบริการแบบนี้ของ EJB container ว่า EJB instance lifecycle management

สำหรับ remote interface หนึ่งจะถูก EJB containers นำไปสร้างคลาส 2 ประเภทคือ

- stub/skeleton เป็นคลาสสำหรับให้ clients ใช้ติดต่อผ่านระบบเครือข่ายมาที่ home object และ EJB object ดังในรูปที่ 2.4



รูปที่ 2.4 การติดต่อผ่านระบบเครือข่าย

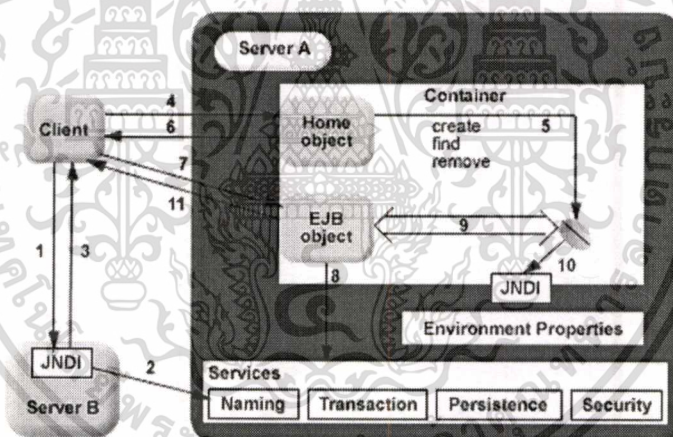
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสเกี่ยวกับ stub (อาจมีเพียงหนึ่งคลาสหรือมากกว่าหนึ่งคลาส ขึ้นอยู่กับผู้ผลิต EJB container) จะถูก download ผ่านระบบเครือข่ายไปให้ clients

- คลาสที่จะนำไปสร้างเป็น EJB objects ซึ่งหน้าที่ของ EJB objects คือเป็นตัวแทนระหว่าง clients กับ bean instance คือเมื่อ clients เรียกไปที่ methods ของ bean instance ที่จริงแล้วเป็นการเรียกไปที่ EJB object ที่เป็นเช่นนี้เพื่อให้ EJB containers สามารถแทรกบริการลงไป method call นั้นก่อนจะส่งต่อไปที่ bean instance ข้อกำหนดของ J2EE ระบุว่า clients ไม่สามารถเรียกไปที่ methods ของ bean instances ได้โดยตรง

2.1.5 การทำงานของ EJB

เพื่ออธิบายการทำงานร่วมกันระหว่าง home object, EJB object และ bean instance เราจะแสดงตัวอย่างการทำงานของ client หนึ่งที่เรียก method หนึ่งของ EJB ดังในรูปที่ 2.5



รูปที่ 2.5 แสดงตัวอย่างการทำงานเมื่อ Client เรียกใช้ EJB

กรณีนี้ถือว่า EJB นั้นถูก deployed ไว้บน server A เรียบร้อยแล้ว ส่วน client เป็น J2EE application client ที่ทำงานอยู่ใน application client container บนเครื่องอีกเครื่องหนึ่ง ซึ่งมี server B ให้บริการ name and directory service โดยอาจจะอยู่บนเครื่องเดียวกันกับ client หรือต่างเครื่องก็ได้ แต่ทั้งหมดอยู่ในเครือข่ายเดียวกัน เมื่อ client จะเรียกใช้ EJB ต้องมีขั้นตอนดังนี้

1. Client ติดต่อไปที่ server B เพื่อขอ reference ของ home object ของ EJB ที่จะใช้งาน
2. Server B จะติดต่อไปยัง server A เพื่อขอ reference ของ home object (ซึ่งถูกทำ binding ไว้กับ name server ตั้งแต่ตอนที่ deploy EJB)
3. เมื่อได้ reference นั้นแล้ว server B จะส่งไปให้ client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Client ใช้ reference ที่ได้เรียกไปที่ home object เพื่อสั่งให้สร้าง (create) bean instance
5. Home object ทำการสร้าง bean instance
6. Home object ส่ง reference ของ EJB object สำหรับ bean instance ที่สร้างขึ้นให้แก่ client
7. Client เรียก business method ไปที่ EJB object
8. EJB object จะแทรกบริการของ EJB container เข้าไปในการเรียก method นั้น ในขั้นตอนนี้อาจต้องสอบถามค่าบางอย่างเช่น database หรือ URL หรืออาจจะขอ references ของบริการที่จะใช้จาก name server
9. EJB object เรียกไปที่ method ของ bean instance และรับค่าที่ถูกส่งกลับมา
10. ระหว่างที่ bean instance ทำงานอาจต้องการสอบถามค่าเกี่ยวกับสถานะแวดล้อมการทำงานจาก name server
11. EJB object ส่งผลลัพธ์ได้ไปให้ client

โปรแกรม client อาจทำงานอยู่ต่างเครื่อง หรือเครื่องเดียวกันกับ J2EE server ก็ได้ วิธีที่ client จะติดต่อกับ EJB จะคล้ายกันคือ ถ้าเป็นกรณีอยู่ต่างเครื่อง client จะต้องเลือกใช้ remote interface ซึ่งจะมี stub/skeleton ในการติดต่อผ่านระบบเครือข่าย แต่ถ้าอยู่ในเครื่องเดียวกันก็อาจจะเลือกใช้ local interface ซึ่งจะเป็นการเรียกใช้ method ในเครื่องเดียวกันจึงทำงานได้เร็วกว่า

2.1.6 สรุป

เทคโนโลยีเอ็นเตอร์ไพรส์จาวาเป็นเทคโนโลยีที่อยู่ในฝั่งของ server ที่คอยจัดการกับฐานข้อมูล โดยสิ่งที่ได้เปรียบของมันก็คือการง่ายในการพัฒนา และการจัดการในเรื่องของ transaction เนื่องจากเอ็นเตอร์ไพรส์จาวาถูกออกแบบมาให้เหมาะสมกับผลิตภัณฑ์ของฐานข้อมูลที่รองรับมาตรฐานนี้ ซึ่งในปัจจุบันแทบทุกผลิตภัณฑ์สามารถที่จะให้เทคโนโลยีนี้ใช้ได้ ในส่วนของระบบนั้นได้ใช้เอ็นเตอร์ไพรส์จาวา 2 แบบคือ Session Beans และ Entity Beans โดยที่ Session Beans ทำหน้าที่พัฒนาในส่วนของ Business Logic และ Entity Beans จะคอยติดต่อกับฐานข้อมูลเพื่อดึงข้อมูลออกมาทำงาน สำหรับในเรื่องของ transaction เอ็นเตอร์ไพรส์จาวาสามารถที่จะกำหนด transaction control และ isolation level ได้ทำให้การจัดการในเรื่องของ transaction เป็นไปได้ง่ายขึ้นเนื่องจากผู้พัฒนาไม่จำเป็นต้องเขียนโปรแกรมควบคุม transaction เอง แต่ในระบบของเอ็นเตอร์ไพรส์จาวา EJB Container จะเป็นผู้ดูแลทั้งหมด

2.2 Struts Framework (Falkner, J. et al. 2001.)

2.2.1 ความเป็นมาของ Struts Framework

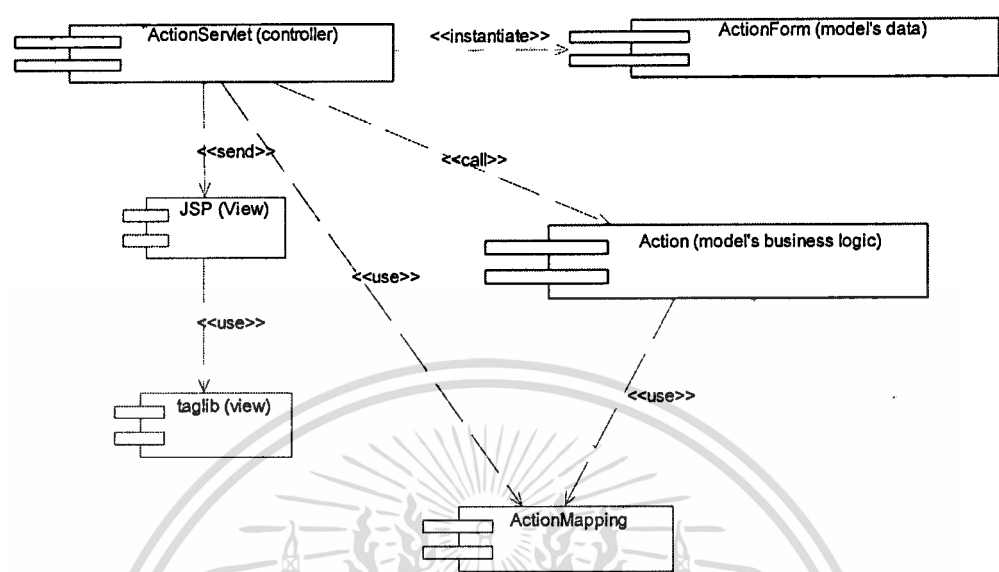
Struts เป็น open-source web application framework ซึ่งอยู่บนพื้นฐานของ MVC design pattern สร้างขึ้นมาเพื่อทำงานร่วมกับ Servlet และ JSP API ซึ่งสามารถนำไปพัฒนา web application ที่มีความซับซ้อน เมื่อพัฒนาด้วย framework นี้จะสามารถ reuse และมีการบำรุงรักษา ในภายหลังได้ง่าย Struts framework เป็นส่วนหนึ่งของ Jakarta Project โดยการดูแลของ Apache Software Foundation สามารถ download ได้ที่ <http://jarkata.apache.org/struts>

แต่เดิมการพัฒนา web application โดยการใช้ JSP และ Servlet นั้น ผู้พัฒนาจะต้องเขียน Servlet เพื่อรับข้อมูลจาก request และทำการ redirect ไปยังหน้า JSP ที่ต้องการซึ่งการทำงานต่าง ๆ เหล่านี้นั้นอยู่ในไฟล์ ๆ เดียว หากเป็นเช่นนี้การ reuse จะเป็นไปได้ยาก รวมถึงการบำรุงรักษาก็เช่นกัน Struts จึงเข้ามาช่วยแก้ปัญหาในเรื่องนี้

บริการที่ Struts framework จัดการมีดังนี้

1. เป็น Servlet ซึ่งทำหน้าที่เป็น controller ใน MVC model
2. ควบคุมการเกิดขึ้นของ JSP
3. การพัฒนาของ JDBC ในการกำหนด datasource และ database connection pool
4. กลไกการตรวจจับความผิดพลาดซึ่งจะดึงเอาข้อความการผิดพลาดที่ระบุไว้ในไฟล์ application resource มาแสดง
5. XML parsing
6. เครื่องมือที่ช่วยในการ upload file
7. เครื่องมือที่ช่วยในการทำ Logging

2.2.2 Struts Architecture



รูปที่ 2.6 Struts Architecture

ในส่วนของการ presentation tier (view) ของ application ที่สร้างบนพื้นฐานของ Struts จะใช้ Struts tag libraries (taglibs) โดยที่ request ที่มาจาก client จะถูกส่งไปยัง servlet ซึ่งเรียกว่า ActionServlet มันมีหน้าที่เป็น controller สำหรับ application ที่ใช้ Struts ทุก ๆ request ที่ต้องผ่านเข้าไปยัง framework นี้จะถูกส่งไปยัง ActionServlet ซึ่งมันจะส่งข้อมูลไปเก็บไว้ที่ ActionForm ActionForm คือ JavaBean ซึ่งเป็นตัวแทนข้อมูลที่ input เข้ามาจาก component ในส่วนของ view ข้อมูลที่ส่งมากับ request เพื่อไปเก็บยัง ActionForm นั้นสามารถที่จะทำการตรวจสอบข้อผิดพลาดได้เมื่อ user กด submit

ActionServlet ถูกกำหนดการทำงานจาก ActionMapping ซึ่ง ActionMapping เป็น object ที่ทำการ map URL ซึ่งกำหนดโดยผู้พัฒนา การกำหนดของ ActionServlet และ ActionMapping จะอยู่ไฟล์ที่เป็นรูปแบบของ XML

สำหรับ application ที่การประมวลผล request คือ Action class ซึ่งเป็น model ใน MVC architecture Action class สามารถที่จะตรวจสอบความถูกต้องของข้อมูลที่ user ส่งมา และถ้าเกิดข้อผิดพลาดระหว่างการทำงาน Action class สามารถที่จะสร้าง error object แล้วส่งไปกับ HTTP แต่ถ้าหากการทำงานนั้นเสร็จสมบูรณ์ Action class จะส่ง ActionForward object ซึ่งเป็นตัวบอกว่าจะให้แสดงหน้า JSP ใด ไปยัง controller จากนั้นหน้า JSP ที่ถูกระบุไว้จะแสดงขึ้นมา

2.2.3 สรุป

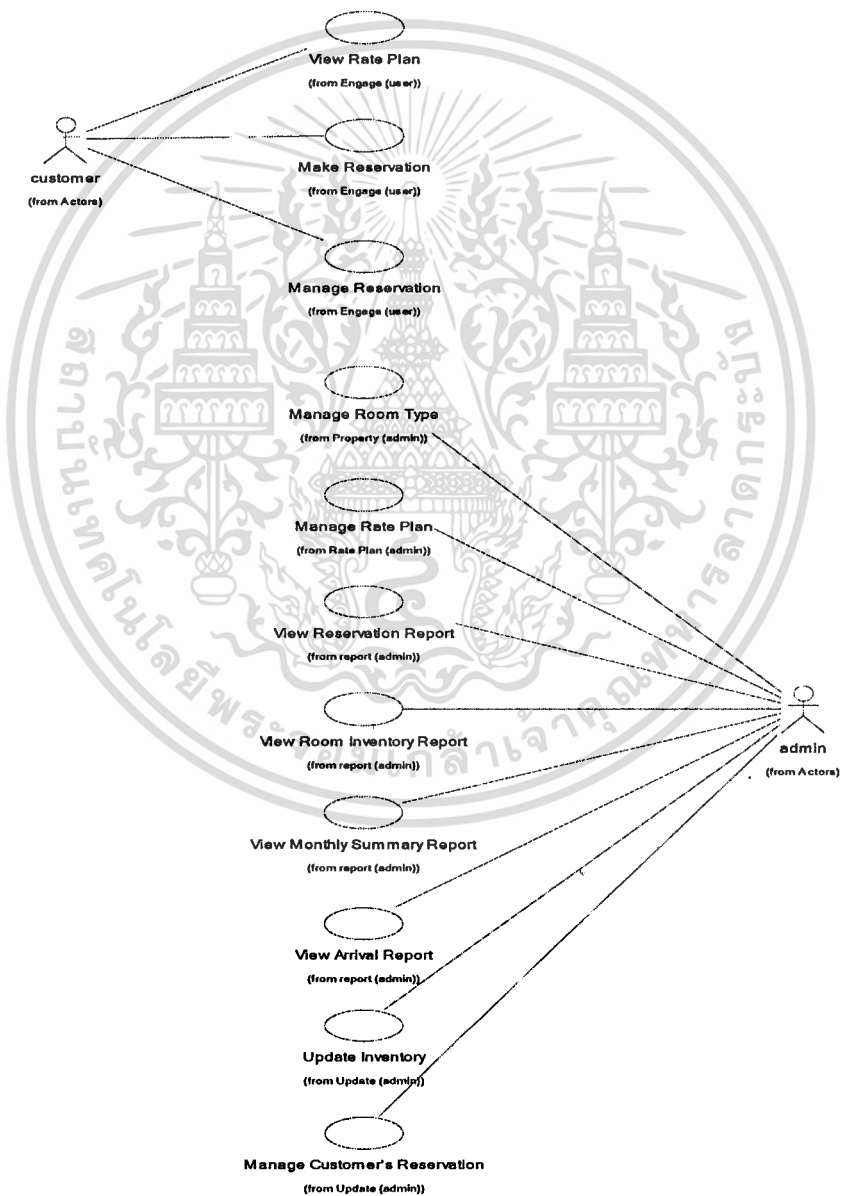
จากที่ได้กล่าวมาแล้วจะเห็นได้ว่า application ที่ใช้ Struts framework มีการทำงานที่แยกเป็นส่วนอย่างชัดเจนจึงเป็นสิ่งที่ง่ายที่จะทำความเข้าใจ และนำไปพัฒนาเป็น web application ที่ซับซ้อน เนื่องจากมีความสามารถที่จะ reuse ได้ดี และหากมีปัญหาหรือต้องเพิ่มเติมความสามารถที่จะแก้ไขได้ และรวดเร็ว



บทที่ 3

การวิเคราะห์ และออกแบบระบบ

3.1 Use Case Diagram



รูปที่ 3.1 Use Case Diagram ของ Hotel System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 Use Case Specification

1. View Rate Plan

1.1 Brief Use Case

use case นี้อนุญาตให้ user ที่เป็น customer ดูรายละเอียดของ rate plan ของแต่ละประเภทของห้อง ซึ่งแสดงราคาห้องทั้ง 5 ฤดู

1.2 Flow of Events

1.2.1 Basic Flow

use case นี้จะเริ่มทำงานเมื่อ user ตัดสินใจที่ดูรายละเอียด rate plan ของแต่ละห้อง

1. Retrieve Data: ระบบจะดึงข้อมูลราคาทั้งหมดของแต่ละห้องใน 5 ฤดู ซึ่งข้อมูลเหล่านี้จะอยู่ใน Manage Rate Plan use case

2. Display Rate: ระบบจะแสดงราคากลับไปยัง user

use case นี้จะเสร็จสิ้นเมื่อได้แสดงราคาจนหมดทุกห้องทุกฤดูแล้ว

1.2.2 Alternative Flows

1.2.2.1 No rate Plan

ถ้าหากในขณะนั้นไม่มี rate plan ใดเลย ระบบจะต้องแสดงข้อความแก่ user ว่า
“Sorry, there is no rate plan available. ”

1.3 Special Requirement

ไม่มี

1.4 Pre-Condition

ไม่มี

1.5 Post- Condition

ไม่มี

1.6 Extension Points

ไม่มี

2. Make Reservation

2.1 Brief Use Case

use case นี้อนุญาตให้ user ที่เป็น customer สามารถที่จะเข้ามาเพื่อที่จะสำรองห้องพักในแบบออนไลน์ได้ โดยที่ระบบจะตรวจสอบ และเลือกห้องที่เหมาะสม หลังจากที่ user ยืนยันที่จะสำรองห้องพักนั้น ๆ แล้ว ระบบจะสร้าง confirmation no. ให้อัตโนมัติ

2.2 Flow of Events

2.2.1 Basic Flow

user case จะเริ่มทำงานเมื่อ user ตัดสินใจที่จะสำรองห้องพัก

1. Display Form: ระบบจะแสดง stay information form ให้กับ user
2. Enter Value: user จะกรอก stay information ของตน จากนั้นจึงส่ง request ไปยังระบบเพื่อให้ระบบตรวจสอบ และแสดงห้องที่เหมาะสมออกมา
3. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอกใน stay information
 - ระยะเวลาในการเข้าพักต้องระบุ
 - จำนวนห้องที่ต้องการต้องถูกระบุ
 - จำนวนผู้ใหญ่ และเด็กต้องถูกระบุ
4. Search for Room Availability: ระบบจะค้นหาห้อง และ rate plan ที่เหมาะสมกับข้อมูลที่รับมา
5. Select Room and Rate: ระบบจะแสดงรายการของห้องที่เหมาะสมขึ้นมาให้ จากนั้น user จึงเลือกห้องจากรายการนั้น
6. Display Form: ระบบจะแสดง guest information form ให้กับ user
7. Enter Value: user จะกรอกข้อมูลของตน และส่ง request ไปยังระบบเพื่อทำงานในลำดับต่อไป
8. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอกใน guest information
 - ชื่อ, นามสกุล, อีเมล และหมายเลขโทรศัพท์ต้องถูกระบุ
 - หมายเลขบัตร credit ต้องถูกระบุ
9. Preview Data: ระบบจะแสดงข้อมูลที่ผ่านมาทั้งหมดให้ user
10. Confirm Reservation: user ทำการยืนยันที่จะสำรองห้องพัก
11. Generate Confirmation No.: ระบบจะสร้าง confirmation no. และแสดงข้อมูลในการสำรองห้องพักให้กับ user เพื่อพิมพ์เก็บไว้เป็นหลักฐาน

- Confirmation No. จะประกอบไปด้วยตัวเลข 9 หลัก ซึ่งแต่ละหลักจะมีความหมายดังต่อไปนี้

103020001 = หลักแรกเป็นตัวบอกว่าเป็น confirmation no. (1)

หลักที่สอง และสามเป็นเลขของปี (03)

หลักที่สี่ และห้าเป็นเลขของเดือน (02)

สี่หลักที่เหลือเป็นจำนวนที่เพิ่มขึ้นเรื่อย ๆ (0001)

เมื่อขึ้นเดือนถัดไประบบจะเริ่มนับหมายเลขสี่หลักสุดท้ายใหม่ (0001)

2.2.2 Alternative Flows

2.2.2.1 Stay Information Not Complete

ถ้าใน step ที่ 3 user ไม่ได้กรอกข้อมูลที่จำเป็นครบทั้งหมด ระบบจะต้องเตือน user ให้กรอกข้อมูลให้ครบถ้วน

2.2.2.2 Guest Information Not Complete

ถ้าใน step ที่ 8 user ไม่ได้กรอกข้อมูลที่จำเป็นครบทั้งหมด ระบบจะต้องเตือน user ให้กรอกข้อมูลให้ครบถ้วน

2.2.2.3 No Room Availability

ถ้าใน step ที่ 4 ไม่มีห้องที่เหมาะสมเลยในขณะนั้น ระบบต้องแสดงข้อความบอก user ว่า “Sorry, there is no room available.” จากนั้นระบบจะให้ user กรอกข้อมูลใหม่อีกครั้ง

2.2.2.4 Edit Reservation Information

user สามารถที่จะกลับไปแก้ไขข้อมูลจาก step ที่ผ่านมาของการสำรองห้องพักได้

2.3 Special Requirement

1. ในการแสดงรายการห้องที่เหมาะสมจะแสดง 15 ห้องต่อหน้า

2. การเลือกห้อง

2.1 เปรียบเทียบจำนวนของผู้เข้าพักทั้งหมด กับจำนวนสูงสุดที่ห้องพักแต่ละห้องรับได้

- ถ้าจำนวนของผู้เข้าพักทั้งหมดน้อยกว่าจำนวนสูงสุดที่ห้องพักแต่ละห้องรับได้ จะไปยัง step ที่สอง
- ถ้าจำนวนของผู้เข้าพักทั้งหมดมากกว่าจำนวนสูงสุดที่ห้องพักแต่ละห้องรับได้ จะไปยัง alternative flows ข้อ 2.2.2.3

3. การนับจำนวน Extra Person

Extra Person = จำนวนของผู้เข้าพักทั้งหมด – (จำนวนสูงสุดที่ห้องพักแต่ละห้องรับได้ – จำนวนเตียงเสริมที่ห้องแต่ละห้องมี)

2.4 Pre-Condition

ไม่มี

2.5 Post-Condition

ไม่มี

2.6 Extension Points

ไม่มี

3. Manage Reservation

3.1 Brief Use Case

use case นี้อนุญาตให้ user ที่เป็น customer สามารถที่จะเข้ามาแก้ไข และยกเลิกการสำรองห้องพัก ซึ่งการที่จะทำเช่นนี้ได้มันจะต้องเข้ามาแก้ไขก่อนที่จะเลยกำหนดวันคืนห้อง ถ้า user เลือกที่จะยกเลิกห้องพัก ระบบจะทำการสร้าง cancellation no. ใหม่ให้กับ user

3.2 Flow of Events

3.2.1 Basic Flow

use case นี้จะเริ่มทำงานเมื่อ user ตัดสินใจที่จะแก้ไข หรือยกเลิกห้องพัก

1. Display Reservation: ระบบจะแสดงหน้าสำหรับให้ user ใส่ข้อมูลเพื่อค้นหาใบจองที่ต้องการ
2. Enter Value: user ใส่ confirmation no. หรือ credit card no. และนามสกุล และส่ง request ไปยังระบบ
3. Search for Reservation: ระบบจะทำการค้นหาใบจองที่ตรงกับข้อมูลที่ user กรอกเข้ามา
4. Display Reservation List: ระบบจะแสดงรายการการจองแก่ user
 - ถ้า user เลือกที่จะแก้ไข Amendment Reservation Information sub-flow จะเริ่มทำงาน
 - ถ้า user เลือกที่จะยกเลิกการจองห้องพัก Cancellation Reservation Information sub-flow จะเริ่มทำงาน

use case นี้จะสิ้นสุดเมื่อ sub-flow ได้ทำงานเสร็จสมบูรณ์

3.2.2 Sub-Flows

3.2.2.1 Amendment Reservation Information

sub-flow นี้จะเริ่มทำงานเมื่อ user เลือกที่จะแก้ไขข้อมูลใบจองห้องพักซึ่งสามารถแก้ไขได้เฉพาะในส่วนของ guest information

1. Display Reservation: ระบบจะดึงข้อมูลของใบจองนั้น ๆ เพื่อให้ user ทำการแก้ไข
2. Edit Value: user ทำการแก้ไขข้อมูลการจองห้องพัก และส่ง request เพื่อให้ระบบทำการ update ข้อมูลใหม่
3. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
Guest Information
 - ชื่อ, นามสกุล, อีเมล และหมายเลขโทรศัพท์ที่ต้องถูกระบุ
 - หมายเลขบัตร credit ต้องถูกระบุ
4. Preview Data: ระบบจะแสดง reservation information ที่ user ได้เลือกไว้ให้กับ user เพื่อให้ user สามารถที่จะพิมพ์เก็บไว้เพื่อเป็นหลักฐาน

3.2.2.2 Cancel Reservation Information

sub-flow นี้จะทำงานเมื่อ user ตัดสินใจที่จะยกเลิกการจองห้องพัก

1. Retrieve Data: ระบบจะดึงข้อมูลการจองให้แก่ user
2. Cancel Reservation: user ส่ง request เพื่อยกเลิกการจองห้อง
3. Generate Cancellation No.: ระบบจะทำการยกเลิกใบจองนั้น ๆ และสร้าง cancellation no.
 - Cancellation No. จะประกอบไปด้วยตัวเลข 9 หลัก ซึ่งแต่ละหลักจะมีความหมายดังตัวอย่างนี้
 903020001 = หลักแรกเป็นตัวย่อว่าเป็น cancellation no. (9)
 หลักที่สอง และสามเป็นเลขของปี (03)
 หลักที่สี่ และห้าเป็นเลขของเดือน (02)
 สี่หลักที่เหลือเป็นจำนวนที่เพิ่มขึ้นเรื่อย ๆ (0001)
 เมื่อขึ้นเดือนถัดไประบบจะเริ่มนับหมายเลขสี่หลักสุดท้ายใหม่ (0001)
4. Preview Cancellation No.: ระบบจะแสดงหน้าที่บอก cancellation no. แก่ user เพื่อให้ user เก็บเป็นหลักฐานไว้

3.2.3 Alternative Flows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.1 Reservation Information Not Complete

ถ้าใน step ที่ 3 ของ amendment reservation information นั้น user ได้กรอกข้อมูลที่ไม่ครบถ้วน ระบบจะต้องเตือน user เพื่อให้ user กรอกให้ครบ

3.2.3.2 Back to Previous Page

หลังจากใน step ที่ 4 ของ amendment reservation information นั้น user สามารถกลับไปใส่ข้อมูลใหม่เพื่อค้นหาห้องที่เหมาะสมได้อีก

3.3 Special Requirement

ระบบจะไม่แสดงข้อมูลการจอง เมื่อวันที่ที่เข้าไปทำการแก้ไขนั้นอยู่ระหว่างช่วงของการพัก หรือเลยกำหนดการพักไปแล้ว

3.4 Pre-Condition

ไม่มี

3.5 Post-Condition

ไม่มี

3.6 Extension Points

ไม่มี

4. Manage Room Type

4.1 Brief Use Case

use case นี้อนุญาตให้ administrator เพิ่มหรือปรับเปลี่ยนรายละเอียดของประเภทห้องได้

4.2 Flow of Events

4.2.1 Basic Flow

use case นี้จะเริ่มทำงานก็ต่อเมื่อ administrator ตัดสินใจที่จะเพิ่ม หรือแก้ไขข้อมูลประเภทห้อง

1. ระบบจะต้องดึงข้อมูลของประเภทของห้องทั้งหมดออกมาให้ administrator

- ถ้า administrator เลือกที่จะเพิ่มประเภทของห้อง Add Room Type sub-flow จะเริ่มทำงาน
- ถ้า administrator เลือกที่จะแก้ไขรายละเอียดของประเภทห้อง Edit Room Type sub-flow จะเริ่มทำงาน

use case นี้จะทำงานเสร็จเมื่อ sub-flow ทำงานเสร็จสมบูรณ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 Sub-Flows

4.2.2.1 Add Room Type

sub-flow นี้จะเริ่มทำงานเมื่อ administrator ตัดสินใจที่จะเพิ่มประเภทของห้องพัก

1. Display Form: ระบบแสดงหน้าเพื่อให้ administrator ได้กรอกข้อมูลของประเภทของห้องพักใหม่
2. Enter Room: administrator กรอกข้อมูลประเภทของห้องพัก และส่ง request เพื่อให้ระบบ update ประเภทห้องพักใหม่
3. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
 - รหัสห้อง, ชื่อห้อง, วันเริ่มใช้งานได้, จำนวนห้องที่เปิด, จำนวนสูงสุดที่ให้ user พักได้
4. Confirm Update Data: ระบบทำการ update ข้อมูล และแสดงหน้าของการ update successful

sub-flow นี้จะทำงานเสร็จเมื่อระบบแสดงข้อความว่า update successfully

4.2.2.2 Edit Room Type

sub-flow นี้จะเริ่มทำงานเมื่อ administrator ตัดสินใจที่จะแก้ไขรายละเอียดของประเภทของห้องพัก

1. Retrieve Data: ระบบจะดึงข้อมูลของประเภทของห้องออกมาทั้งหมดให้ administrator
2. Edit Value: administrator แก้ไขข้อมูล และส่ง request เพื่อให้ระบบได้ update ข้อมูลใหม่
3. Verify data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
 - รหัสห้อง, ชื่อห้อง, วันเริ่มใช้งานได้, จำนวนห้องที่เปิด, จำนวนสูงสุดที่ให้ user พักได้
4. Confirm Updated Data: ระบบทำการ update ข้อมูล และแสดงหน้าของการ update successful

sub-flow นี้จะทำงานเสร็จเมื่อระบบแสดงข้อความว่า update successfully

4.2.3 Alternative Flows

4.2.3.1 Requested Cancel

1. Edit Room Type

ในขณะที่อยู่ในส่วนของการแก้ไขข้อมูลห้องนั้น administrator สามารถที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะยกเลิกการแก้ไขนั้นได้ทุกเมื่อ

2. Add Room Type

ในขณะที่อยู่ในส่วนของการเพิ่มประเภทของห้องนั้น administrator สามารถที่จะยกเลิกการแก้ไขนั้นได้ทุกเมื่อ

4.2.3.2 Room Type Information Not Complete

ใน step ที่ 4 ของ sub-flow ของการเพิ่ม และแก้ไขประเภทของห้องนั้น ถ้า administrator กรอกข้อมูลที่จำเป็นไม่ครบถ้วนนั้น ระบบจะต้องเตือน และให้ administrator ได้กรอกข้อมูลให้ครบ

4.2.3.3 Display Inactive Room Type

เมื่อระบบแสดงรายชื่อของประเภทของห้องนั้น administrator สามารถที่จะ request เพื่อให้แสดงรายชื่อของประเภทของห้องพักที่ไม่ active (ห้องที่มีวันสิ้นสุดเลขไปจากปัจจุบันแล้ว) ซึ่งโดยปกติแล้วห้องที่ไม่ active จะไม่แสดงขึ้นมา

4.3 Special Requirement

ถ้า administrator ได้กรอกวันสิ้นสุดการใช้งานของห้องนั้น เมื่อผ่านวันนั้นไปห้องประเภทนี้จะเป็นห้องที่ไม่ active โดยทันที

4.4 Pre-Condition

administrator จะต้องทำการ login ก่อนที่จะเข้ามาในระบบ

4.5 Post- Condition

ไม่มี

4.6 Extension Points

ไม่มี

5. Manage Rate Plan

5.1 Brief Use Case

use case นี้อนุญาตให้ administrator เข้ามาแก้ไขข้อมูลราคาของแต่ละห้องในแต่ละฤดู

5.2 Flow of Events

5.2.1 Basic Flow

use case นี้จะเริ่มทำงานเมื่อ administrator เลือกที่จะเข้ามาจัดการ rate plan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้า administrator เลือกที่จะแก้ไขลำดับการขายของห้องพัก Edit Sell Sequence sub-flow จะเริ่มทำงาน
- ถ้า administrator เลือกที่จะแก้ไขข้อมูลของฤดู Edit Season sub-flow จะเริ่มทำงาน
- ถ้า administrator เลือกที่จะแก้ไขข้อมูลของราคา Edit Rate sub-flow จะเริ่มทำงาน

use case นี้จะทำงานเสร็จเมื่อ sub-flow ทำงานเสร็จสมบูรณ์แล้ว

5.2.2 Sub-Flows

5.2.2.1 Edit Sell Sequence

sub-flow นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะแก้ไขรายละเอียดลำดับการขายของแต่ละห้อง

1. Display Room Type: ระบบดึงรายการของห้อง และแสดงให้กับ administrator จากนั้น administrator จึงระบุลำดับการขายให้แก่แต่ละห้อง และส่ง request เพื่อให้ระบบทำการบันทึก step นี้

2. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
- Sell sequence ไม่สามารถซ้ำกันได้
 - Sell sequence ไม่สามารถใส่เกินจำนวนห้องที่แสดงขึ้นมา

3. Save Data: ระบบทำการบันทึกลงฐานข้อมูล
sub-flow นี้จะสิ้นสุดเมื่อระบบได้บันทึกการแก้ไขเรียบร้อยแล้ว

5.2.2.2 Edit Season

sub-flow นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะแก้ไขรายละเอียดข้อมูลของฤดูกาล

1. Display Season Form: ระบบดึงข้อมูลของฤดูกาลขาย และแสดงให้กับ administrator จากนั้น administrator ทำการกำหนดฤดูกาลต่าง ๆ และส่ง request เพื่อให้ระบบทำการบันทึก

2. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
- ช่วงระยะเวลาของฤดูกาลขายจะต้องถูกระบุ
 - ช่วงเวลาของฤดูกาลขายจะต้องไม่ซ้อนทับกัน
 - ฤดูกาลทั้งหมดจะต้องต่อเนื่องกัน ไม่สามารถที่จะมีช่องว่างได้

3. Save Data: ระบบทำการบันทึกลงฐานข้อมูล

sub-flow นี้จะสิ้นสุดเมื่อระบบทำการบันทึกการแก้ไขฤดูกาลขายเรียบร้อยแล้ว

5.2.2.3 Edit Rate

sub-flow นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะแก้ไขราคาของห้องพักในแต่ละฤดูกาลขาย

1. Display Rate: ระบบดึงราคาที่ได้สร้างไว้แล้วทั้งหมดขึ้นมาแสดงแก่ administrator
2. Edit Value: administrator ทำการแก้ไขราคา และ request ไปยังระบบเพื่อให้ระบบทำการบันทึก
3. Verify data: ระบบจะตรวจสอบข้อมูลที่ต้องกรอก
 - ราคาห้องของทุกฤดูกาลขายจะต้องถูกระบุจนครบ
 - ถ้า administrator เลือกประเภทของราคาเป็นช่องว่าง แสดงว่าราคาในส่วนนั้นต้องถูกระบุ
 - ถ้า administrator เลือกประเภทของราคาเป็น CP (complementary) ราคาในส่วนนั้นไม่จำเป็นต้องระบุ
 - ถ้า administrator เลือกประเภทของราคาเป็น NA (not applicable) ราคาในส่วนนั้นไม่จำเป็นต้องถูกระบุ
4. Save Data: ระบบทำการบันทึกข้อมูล
- sub-flow นี้จะสิ้นสุดเมื่อระบบทำการบันทึกราคาจนเสร็จสิ้นแล้ว

5.2.3 Alternative Flows

5.2.3.1 Rate Plan Not Complete

ถ้า administrator ใส่ข้อมูลที่จำเป็นไม่ครบ ระบบจะต้องเตือน administrator ให้ใส่ให้ครบ

5.3 Special Requirement

administrator จะต้องทำตาม step ที่ต้องกำหนด sell sequence ของแต่ละห้อง กำหนดช่วงฤดูของการขาย และกำหนดราคาขายของห้องแต่ละฤดู

5.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

5.5 Post- Condition

ไม่มี

5.6 Extension Points

ไม่มี

6. View Reservation Report

6.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถเข้าไปดูรายงานการจองห้องพัก แต่ละรายการจะประกอบด้วย confirmation no. ประเภท ชื่อผู้จอง วันมาถึง วันออก ซึ่งทั้งหมดนี้จะแสดงเป็นรายการออกมาให้กับ administrator ได้เลือก

6.2 Flow of Events

6.2.1 Basic Flow

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะต้องการดูรายงานการจองห้องพัก

1. Display Form: ระบบจะแสดง form สำหรับให้ administrator ใช้ในการค้นหา
2. Enter Data: administrator เลือกสถานะ และรายละเอียดอื่น ๆ จากนั้นจึงส่ง request เพื่อให้ระบบประมวลผลและไปสู่ step ต่อไป
3. Verify data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องการออก
 - administrator ต้องระบุสถานะของใบจองที่ต้องการค้นหา
 - administrator สามารถระบุสถานะร่วมกับ cancellation no. หรือ confirmation no. เพื่อใช้ในการค้นหาได้
 - ถ้า administrator เลือกสถานะ amendment หรือ all unprinted แล้วนั้น administrator จะไม่สามารถใส่ cancellation no. และ confirmation no. ได้
 - administrator สามารถระบุสถานะใบจอง และนามสกุลผู้จองเพื่อใช้ในการค้นหาได้
 - administrator สามารถระบุสถานะใบจอง นามสกุลผู้จอง และช่วงของระยะเวลาเพื่อใช้ในการค้นหาได้
 - administrator สามารถระบุสถานะใบจอง และช่วงของระยะเวลาเพื่อใช้ในการค้นหาได้
4. Retrieve Data: ระบบดึงข้อมูลที่ตรงตามที่ administrator ระบุไว้ขึ้นมา
 - ระบบจะแสดงโดยเรียงตามวันเวลาของการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Selects Reservation: administrator ส่ง request ไปยังระบบเพื่อให้ระบบแสดงรายละเอียดของใบจองนั้น ๆ

6. Retrieve Data: ระบบดึงข้อมูลของใบจองนั้น ๆ และแสดงให้กับ administrator use case นี้จะสิ้นสุดเมื่อระบบได้แสดงรายละเอียดของใบจองที่ administrator เลือกจนเสร็จสิ้นแล้ว

6.2.2 Alternative Flows

6.2.2.1 Delete Reservation

administrator สามารถที่จะลบรายการใด ๆ โดยที่ระบบจะลบข้อมูลนั้น ๆ ออก จากฐานข้อมูล และแสดงรายการที่เปลี่ยนไปให้กับ administrator

6.2.2.2 No Search Result

ถ้าใน step ที่ 2 ไม่สามารถหารายการใดที่ตรงตามที่ administrator ค้นหา ระบบ ต้องแสดงข้อความว่า “Sorry, there is no search result, please search again.”

6.2.2.3 Request Sorting

administrator สามารถที่จะเลือกให้ระบบเรียงข้อมูลตามวันเวลาการจอง confirmation no. หรือ cancellation no. ได้

6.3 Special Requirement

ในส่วนของการแสดงรายการนั้นให้แสดง 15 รายการต่อหนึ่งหน้า

6.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

6.5 Post- Condition

ไม่มี

6.6 Extension Points

ไม่มี

7. View Room Inventory Report

7.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถที่จะดูรายงานของจำนวนห้อง คงเหลือในแต่ละวัน โดยการระบุของเขตของวันเวลาเพื่อแสดงรายงานออกมา

7.2 Flow of Events

7.2.1 Basic Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะดูรายงานของจำนวนห้องที่ถูกใช้ไป

1. Display Form: ระบบจะแสดง form เพื่อให้ administrator ได้ใส่ข้อมูลที่ใช้สำหรับค้นหา
2. Enter Data: administrator เลือกห้อง และช่วงระยะเวลา จากนั้นจึง request ไปยังระบบเพื่อให้ระบบแสดงรายงานออกมา
3. Retrieve Data: ระบบดึงรายละเอียดของจำนวนออกมา และแสดงให้ administrator

use case นี้จะสิ้นสุดเมื่อระบบได้แสดงรายงานแก่ administrator เสร็จสิ้นแล้ว

7.2.2 Alternative Flows

7.2.2.1 Request Sorting

administrator สามารถเลือกให้ระบบเรียงข้อมูลตาม วันเวลา หรือรหัสของห้อง

7.3 Special Requirement

ในส่วนของการแสดงรายการนั้นให้แสดง 15 รายการต่อหนึ่งหน้า

7.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

7.5 Post-Condition

ไม่มี

7.6 Extension Points

ไม่มี

8. View Monthly Summary Report

8.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถดูรายงานการขายเป็นรายเดือน และรายงานการขายตั้งแต่ต้นปีจนถึงวันปัจจุบัน ซึ่งสามารถเทียบได้กับช่วงเวลาเดียวกันของปีที่ผ่านมาได้

8.2 Flow of Events

8.2.1 Basic Flow

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะดูรายงานการขาย

ประจำเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1. Display Form: ระบบแสดง form เพื่อให้ administrator ได้ใส่ข้อมูลเพื่อใช้เป็น ตัวเปรียบเทียบในการค้นหา
 - 2. Enter Value: administrator เลือกวันที่ต้องการ และ request ไปยังระบบเพื่อให้ ระบบแสดงรายงานออกมา
 - 3. Retrieve Data: ระบบดึงข้อมูลเกี่ยวกับรายงานที่ administrator ต้องการออกมา โดยที่แต่ละค่ามีความหมายดังนี้
 - Net Room Night คือจำนวนห้องที่ถูกจองในเดือนนั้น เวลานั้น
 - % total room night คือจำนวนห้องที่ถูกจองคิดเป็นเปอร์เซ็นต์จากห้อง ทั้งหมดในเดือนนั้น เวลานั้น $(\text{Net Room Night} / \text{Total}) * 100$
 - Net Revenue คือรายได้รวมจากการจองห้องในเดือนนั้น เวลานั้น
 - Average Rate คือรายได้รวมหารด้วยจำนวนห้องที่ถูกจองในเดือนนั้น
 - Room Denied คือจำนวนห้องที่ถูกยกเลิกการจอง
- use case นี้จะสิ้นสุดเมื่อระบบได้แสดงรายงานจนเสร็จสิ้นแล้ว

8.2.2 Alternative Flows

ไม่มี

8.3 Special Requirement

ไม่มี

8.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

8.5 Post- Condition

ไม่มี

8.6 Extension Points

ไม่มี

9. View Arrival Report

9.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถดูรายงานเกี่ยวกับการเข้าพักของ ลูกค้าโดยการระบุวันในการค้นหา รายละเอียดของรายงานจะรวมถึงระยะเวลาในการเข้าพักที่วัน คิดเป็นราคาเท่าไร และใช้ credit card อะไร

9.2 Flow of Events

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2.1 Basic Flow

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะดูรายงานการเข้าพัก

1. Display Form: ระบบแสดง form เพื่อให้ administrator ได้ใส่ข้อมูลเพื่อเอาไว้เปรียบเทียบในการค้นหา

2. Enter Value: administrator ระบุวันเวลาการเข้ามาพัก และส่ง request เพื่อให้ระบบแสดงรายงานออกมา

3. Retrieve Data: ระบบดึงข้อมูลขึ้นมา และแสดงผลลัพธ์กลับไปยัง administrator โดยที่แต่ละค่ามีความหมายดังนี้

- Guarantee คือประเภท และหมายเลขของบัตรเครดิตที่ใช้
- Average Length of Stay คือผลรวมของระยะเวลาการเข้าพักหารด้วยจำนวนห้องทั้งหมดที่เข้าพักในวันนั้น
- Total Estimated Revenue คือรายได้รวมทั้งหมดของทุกห้องที่เข้าพักวันนั้น $([LOS1 * Rate1 * No. of rooms1] + [LOS2 * Rate2 * No. of rooms2] + \dots)$
- Estimated Average Room Type คือผลรวมของราคาแต่ละห้องหารด้วยจำนวนห้องทั้งหมด
- Total Rooms Arriving คือจำนวนห้องที่มาถึงในวันนั้น

use case นี้จะสิ้นสุดเมื่อระบบได้แสดงรายงานจนเสร็จสิ้นแล้ว

9.2.2 Alternative Flows

ไม่มี

9.3 Special Requirement

ไม่มี

9.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

9.5 Post-Condition

ไม่มี

9.6 Extension Points

ไม่มี

10. Update Inventory

10.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถเข้ามาแก้ไขจำนวนห้องพักคงเหลือ เมื่อเกิดการจอง หรือการยกเลิกจากระบบอื่น

10.2 Flow of Events

10.2.1 Basic Flow

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะแก้ไขจำนวนห้องพักคงเหลือ

- ถ้า administrator เลือกที่จะจองห้องพัก Reserve Room sub-flow จะเริ่มงาน
- ถ้า administrator เลือกที่จะยกเลิกห้องพัก Cancel Room sub-flow จะเริ่มทำงาน

use case นี้จะสิ้นสุดเมื่อ sub-flow ได้ทำงานเสร็จสิ้นแล้ว

10.2.2 Sub-Flows

10.2.2.1 Reserve Room

sub-flow นี้เริ่มทำงานเมื่อ administrator เลือกที่จะทำการจองห้อง

1. Display Form: ระบบแสดง form ในการจองห้องแก่ administrator
 2. Enter Data: administrator ระบุระยะเวลาการเข้าพัก รหัสของห้อง และจำนวนห้องที่ต้องการ จากนั้นส่ง request ไปเพื่อให้ระบบทำการจองห้องพัก
 3. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
 - ระยะเวลาการเข้าพักต้องถูกระบุ
 - รหัสของห้องพักต้องถูกระบุ
 - จำนวนห้องต้องถูกระบุ
 4. Confirm Updated Date: ระบบจะแสดงรายละเอียดที่เลือกไว้เพื่อให้ administrator ได้ตรวจสอบ จากนั้นจึงส่ง request เพื่อให้ระบบทำการบันทึก
 5. Save Data: ระบบบันทึกข้อมูล และแสดงผลการบันทึกให้กับ administrator
- sub-flow นี้จะสิ้นสุดเมื่อระบบแสดงข้อความการบันทึกเสร็จสมบูรณ์แล้ว

10.2.2.2 Cancel Room

sub-flow นี้เริ่มทำงานเมื่อ administrator เลือกที่จะยกเลิกห้องพัก

1. Display Form: ระบบแสดง form ในการยกเลิกห้องแก่ administrator
2. Enter Data: ระบุระยะเวลา รหัสของห้อง และจำนวนห้อง จากนั้นส่ง request

ไป เพื่อให้ระบบทำการยกเลิกห้องพักนั้น ๆ

3. Verify Data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก

- ระยะเวลาการเข้าพักต้องถูกระบุ
- รหัสของห้องพักต้องถูกระบุ
- จำนวนห้องต้องถูกระบุ

4. Confirm Updated Date: ระบบจะแสดงรายละเอียดที่เลือกไว้เพื่อให้ administrator ได้ตรวจสอบ จากนั้นจึงส่ง request เพื่อให้ระบบทำการบันทึก

5. Save Data: ระบบบันทึกข้อมูล และแสดงผลการบันทึกให้กับ administrator sub-flow นี้จะสิ้นสุดเมื่อระบบแสดงข้อความการบันทึกเสร็จสมบูรณ์แล้ว

10.2.3 Alternative Flows

10.2.3.1 Updated Inventory Information Not Complete

ถ้าใน step ที่ 3 ของทั้งสอง sub-flow มีข้อมูลที่จำเป็นบ้างตัวไม่ได้ระบุ ระบบจะต้องเตือน administrator ให้ใส่ข้อมูลให้ครบ

10.3 Special Requirement

ไม่มี

10.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

10.5 Post- Condition

ไม่มี

10.6 Extension Points

ไม่มี

11. Manage Customer’s Reservation

11.1 Brief Use Case

use case นี้อนุญาตให้ administrator สามารถเข้ามาแก้ไขข้อมูลของการจอง และยกเลิกการจองห้องพักได้

11.2 Flow of Events

11.2.1 Basic Flow

use case นี้เริ่มทำงานเมื่อ administrator ตัดสินใจที่จะยกเลิก หรือแก้ไขข้อมูลการ

จอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้า administrator เลือกที่จะยกเลิกการจองนั้น ๆ Cancel Reservation sub-flow จะเริ่มทำงาน
 - ถ้า administrator เลือกที่จะแก้ไขข้อมูลการจอง Amend Reservation sub-flow จะเริ่มทำงาน
- use case นี้จะสิ้นสุดเมื่อ sub-flow ทำงานจนเสร็จสิ้นแล้ว

11.2.2 Sub-Flows

11.2.2.1 Cancel Reservation

sub-flow นี้เริ่มทำงานเมื่อ administrator เลือกที่จะแก้ไขข้อมูลการจองห้องพัก

1. Display Form: ระบบจะแสดง form เพื่อให้ administrator ได้ใส่ข้อมูลเพื่อใช้ในการค้นหา
 2. Enter Value: administrator ต้องระบุนามสกุลของลูกค้า และ confirmation no. ของใบจองนั้น ๆ จากนั้นจึงส่ง request เพื่อให้ระบบทำการค้นหา
 3. Retrieve Data: ระบบดึงข้อมูลการจองขึ้นมาให้กับ administrator
 4. Cancel Reservation: administrator ส่ง requests ไปยังระบบเพื่อให้ระบบยกเลิกใบจองนั้น ๆ
 5. Generate Cancellation No.: ระบบทำการยกเลิกใบจองนั้น ๆ และสร้าง cancellation no. ขึ้นมา
 6. Confirm Cancel: ระบบแสดงข้อความการยกเลิกแก่ administrator
- sub-flow นี้จะสิ้นสุดเมื่อระบบได้แสดงข้อความการยกเลิกแก่ administrator เรียบร้อยแล้ว

11.2.2.2 Amend Reservation

sub-flow นี้เริ่มทำงานเมื่อ administrator เลือกที่จะแก้ไขข้อมูลของใบจองนั้น ๆ ซึ่งสามารถแก้ไขได้เฉพาะข้อมูลในส่วน of guest information

1. Display Form: ระบบแสดง form เพื่อให้ administrator ได้ใส่ข้อมูลในการค้นหา
2. Enter Value: administrator ต้องระบุนามสกุลของลูกค้า และ confirmation no. ของใบจองนั้น ๆ จากนั้นจึงส่ง request เพื่อให้ระบบทำการค้นหา
3. Retrieve Data: ระบบดึงข้อมูลการจองขึ้นมาให้กับ administrator เพื่อให้ administrator ทำการแก้ไข
4. Edit Value: administrator ทำการแก้ไขข้อมูลการจอง

- 5. Update Data: administrator ส่ง request เพื่อให้ระบบทำการบันทึกข้อมูล
- 6. Verify data: ระบบจะตรวจสอบข้อมูลที่จำเป็นต้องกรอก
 - Guest Information
 - ชื่อ, นามสกุล, อีเมล และหมายเลขโทรศัพท์ที่ต้องถูกระบุ
 - หมายเลข credit card ต้องถูกระบุ
- 7. Preview Reservation Information: ระบบแสดงข้อมูลที่ administrator ได้แก้ไข

ไป

sub-flow นี้จะสิ้นสุดเมื่อระบบได้แสดงข้อมูลที่ administrator ได้แก้ไขไปแล้ว

11.2.3 Alternative Flows

11.2.3.1 Reservation Information Not Complete

ถ้าใน step ที่ 6 ของ sub-flow amend reservation นั้น administrator ไม่ได้ระบุข้อมูลที่จำเป็นครบถ้วนแล้วนั้น ระบบจะต้องเตือน administrator และให้ administrator กลับมาใส่ข้อมูลให้ครบถ้วน

11.3 Special Requirement

ไม่มี

11.4 Pre-Condition

administrator ต้อง login ก่อนเข้าใช้ระบบ

11.5 Post- Condition

ไม่มี

11.6 Extension Points

ไม่มี

3.3 Class Diagram

การออกแบบนั้นจะแบ่งออกเป็น 3 Layer คือ Application, Business Service และ Common Business Object Layer ซึ่งแต่ละ layer มีรายละเอียดดังนี้

3.3.1 Application Layer

ในส่วนของ Application Layer ของระบบนี้จะเป็นกลุ่มของหน้าเว็บเพจทั้งหมด ซึ่งจะแบ่งออกเป็น package ในลักษณะดังนี้

- user
- reservation
- rate plan
- admin
 - property
 - rate plan
 - report
 - reservation

หน้าเว็บเหล่านี้เรียกอีกอย่างได้ว่าเป็น Boundary ดังนั้นจึงไม่ขอแสดง class diagram ของ layer นี้

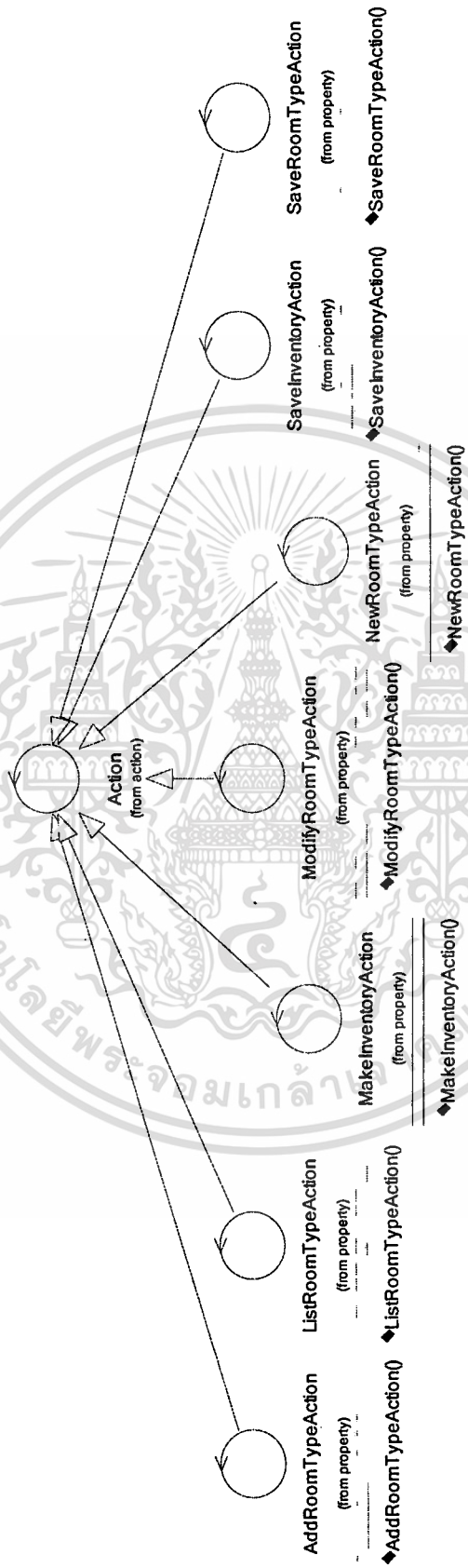
3.3.2 Business Service Layer

ในส่วนของ Business Service Layer ของระบบนี้จะประกอบไปด้วย class ในหลาย ๆ กลุ่มคือ Action, Manager, Data Access Object (DAO), EJB Helper ซึ่งรายละเอียดของแต่ละกลุ่มเป็นดังนี้

1. Action

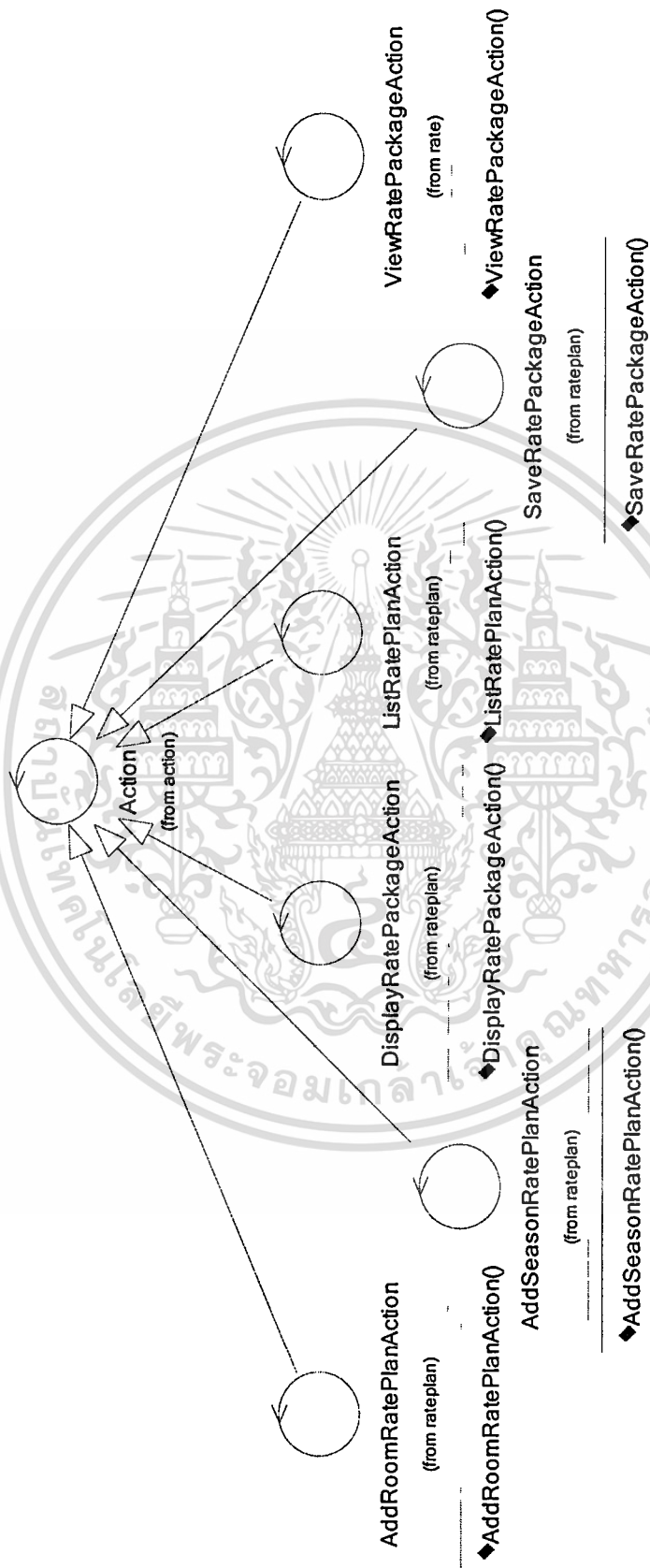
Class ในกลุ่มนี้ทำหน้าที่เป็น Controller ซึ่งจะแบ่งออกเป็น package ต่าง ๆ คือ Property Package, Rate Plan Package, Report Package และ Reservation Package ดังแสดงในรูปที่ 3.2 – 3.6

Property Package



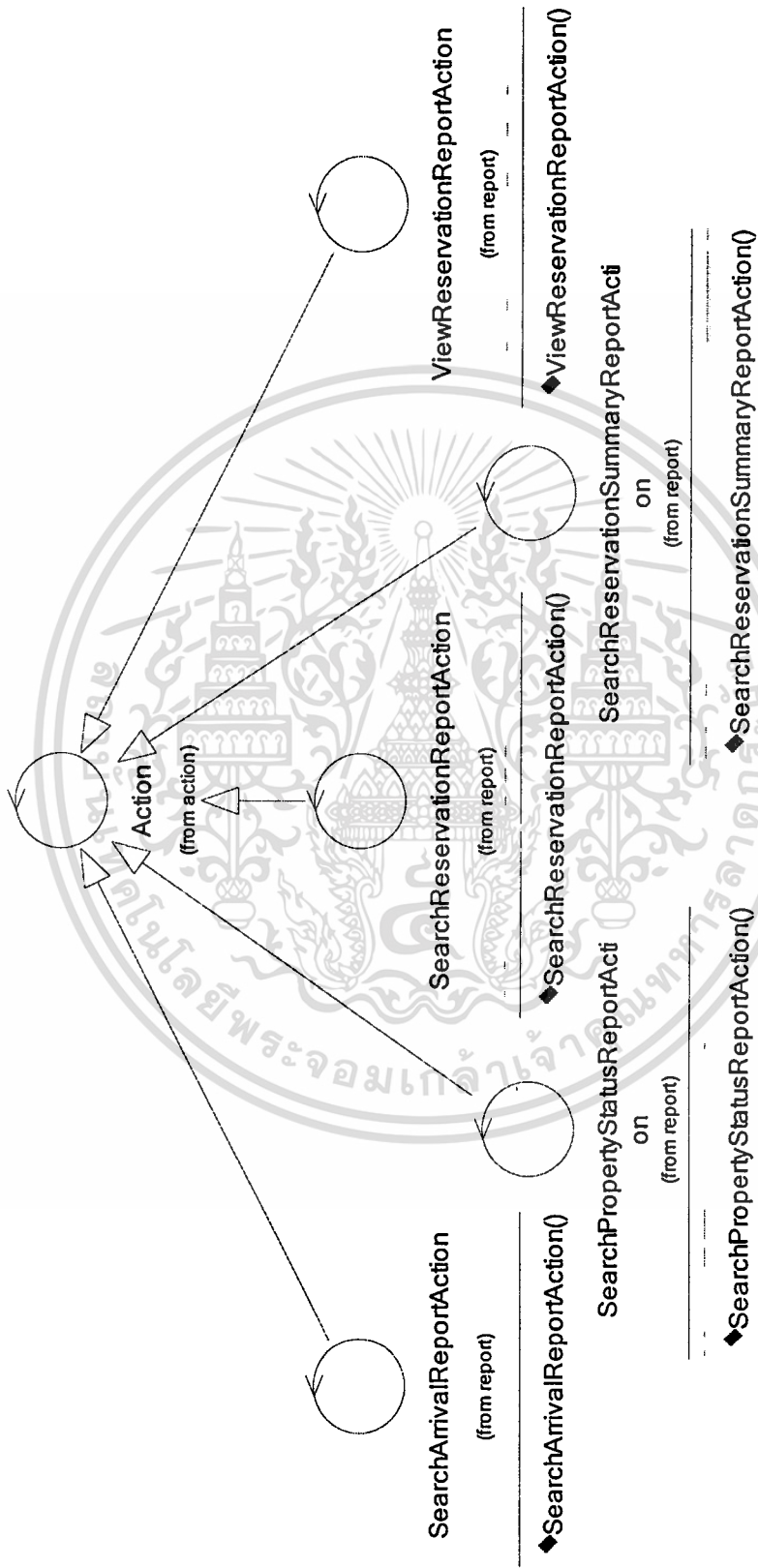
รูปที่ 3.2 Class Diagram ใน Property Package (Business Service Layer)

Rate Plan Package



รูปที่ 3.3 Class Diagram ใน Rate Plan Package (Business Service Layer)

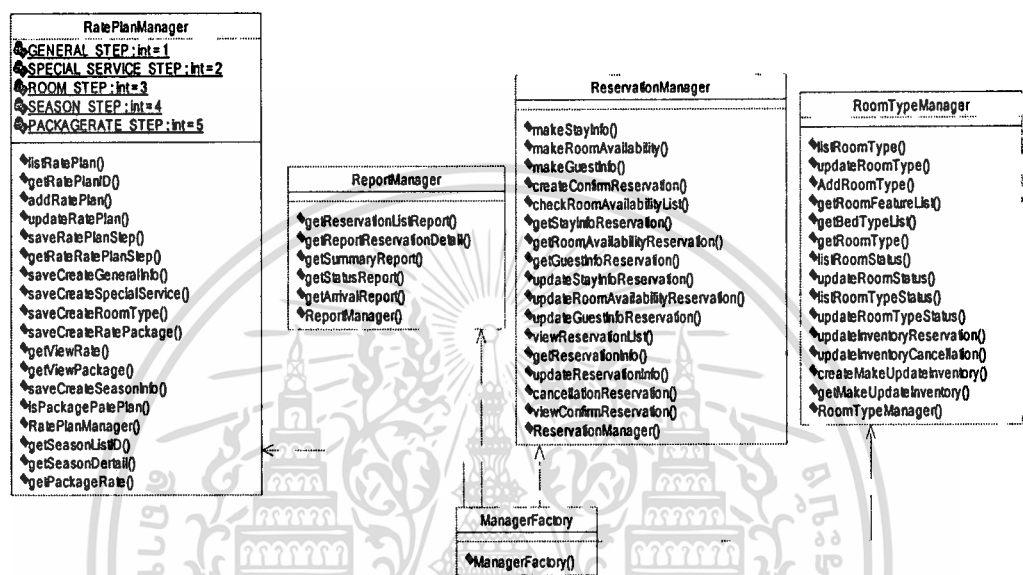
Report Package



รูปที่ 3.4 Class Diagram ใน Report Package (Business Service Layer)

2. Manager

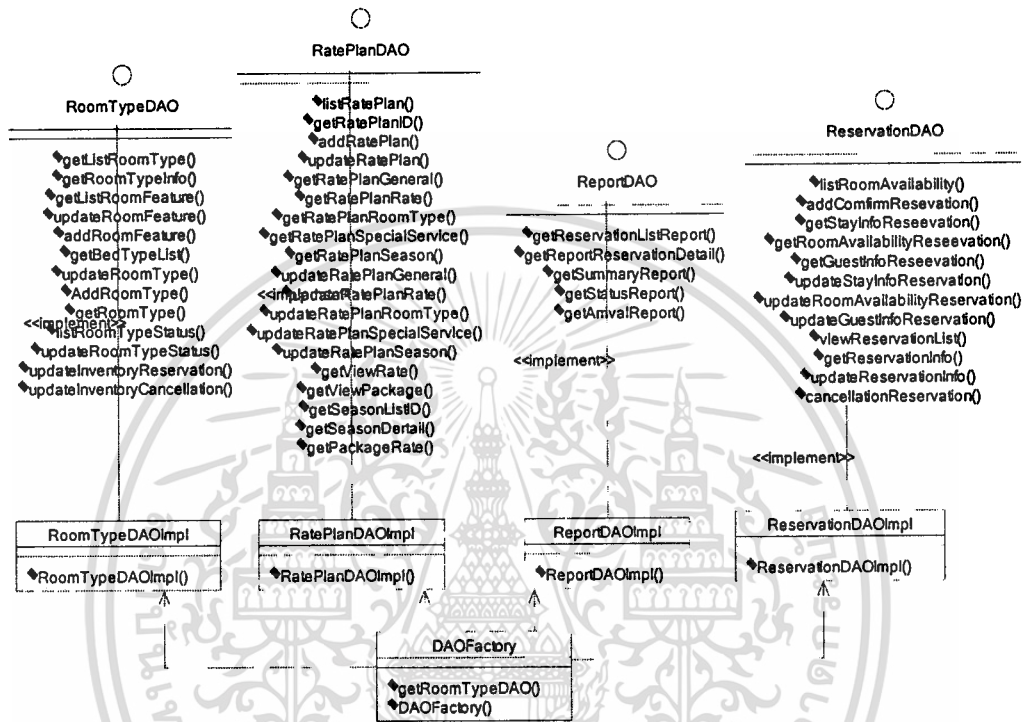
Class ในกลุ่มนี้เป็น class ที่ทำหน้าที่คอยจัดการไม่ว่าจะเป็นเรื่องของการเรียก method ที่ทำงานอย่างใดอย่างหนึ่งซึ่ง method นั้นอาจจะอยู่ในclass ในกลุ่มอื่น ๆ ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 Class Diagram ในกลุ่มของ Manager (Business Service Layer)

3. Data Access Object (DAO)

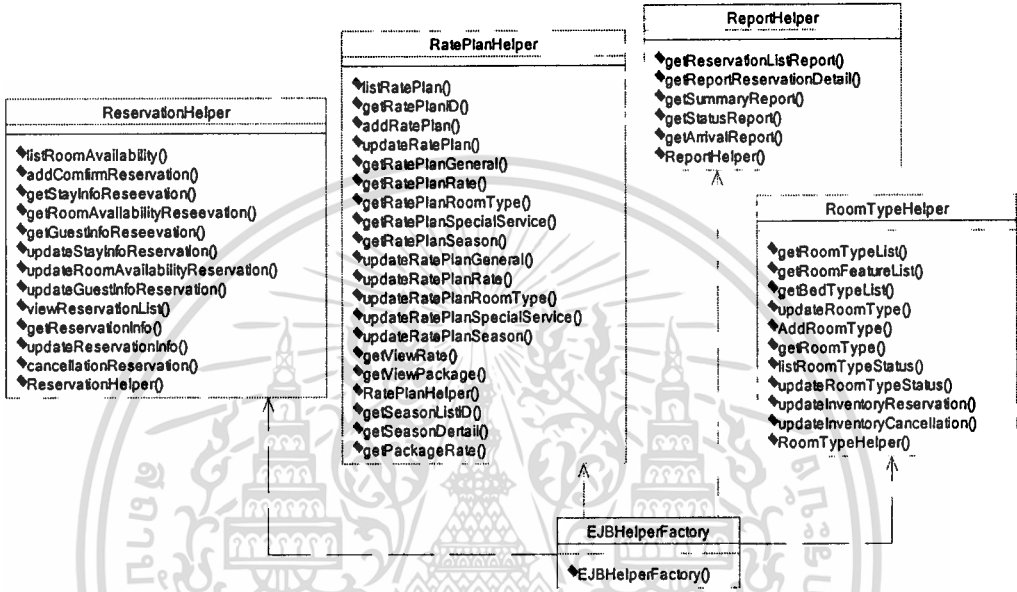
Class ในกลุ่มนี้เป็น class ที่ทำหน้าที่ ในการส่ง Entity object ไปยัง class ในกลุ่มของ EJB Helper ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 Class Diagram ในกลุ่มของ Data Access Object (Business Service Layer)

4. EJB Helper

Class ในกลุ่มนี้เป็นตัวช่วยในการติดต่อไปยัง EJB โดยใช้ JNDI ในการอ้างถึง EJB ที่ต้องการ ซึ่ง EJB ที่ class ในกลุ่มนี้เรียกนั้นจะเป็น EJB ชนิด Session Beans ดังแสดงในรูป 3.9



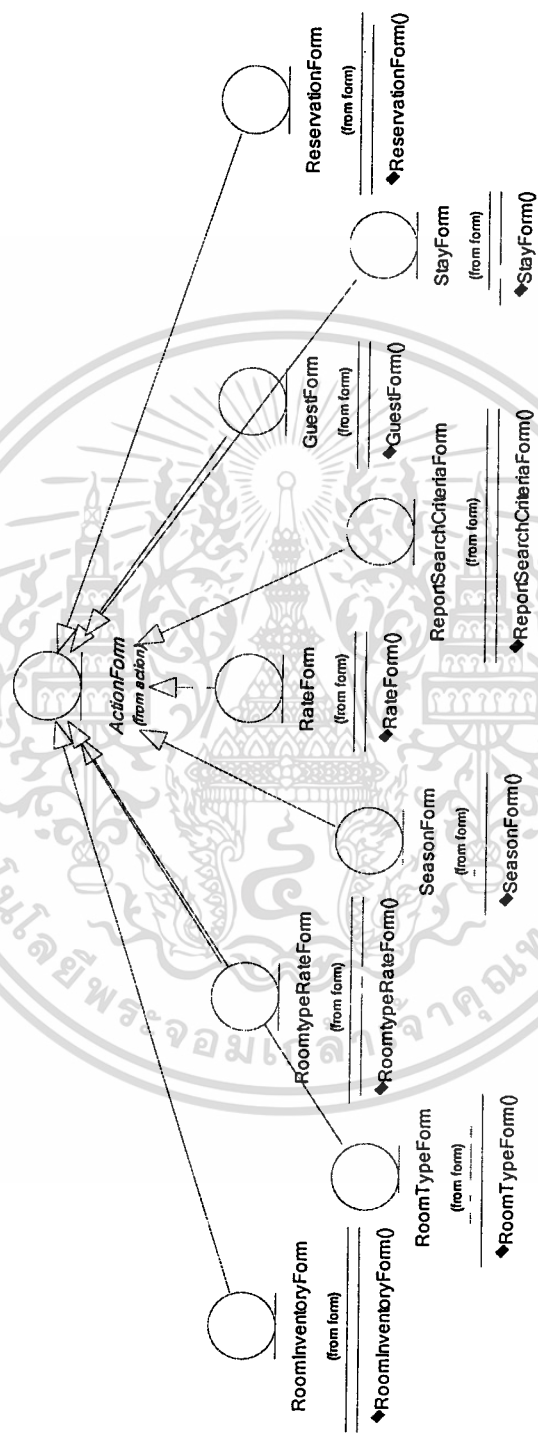
รูปที่ 3.9 Class Diagram ในกลุ่มของ EJB Helper (Business Service Layer)

3.3.3 Common Business Object Layer

ในส่วนของ Common Business Object ของระบบนี้เป็นกลุ่มของ class ที่ทำหน้าที่เก็บตัวข้อมูล หรือเรียกอีกอย่างว่า Entity จะประกอบไปด้วย class สองกลุ่มคือ Action Form และ Data Transfer Object (DTO) ซึ่งรายละเอียดของแต่ละกลุ่มเป็นดังนี้

1. Action Form

Class ในกลุ่มนี้ทำหน้าที่รับค่าข้อมูลจาก form ในหน้าเว็บเพจ เพื่อให้คลาสใน Business Service Layer ได้นำมาจัดการต่อไป ดังแสดงในรูป 3.10



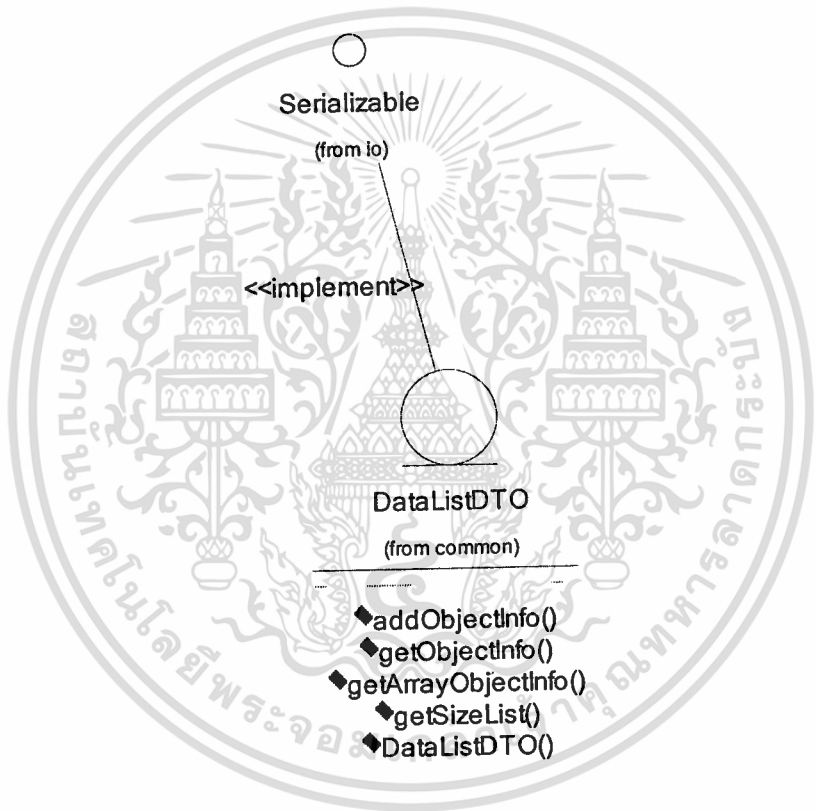
รูปที่ 3.10 Class Diagram ในกลุ่มของ Action Form (Common Business Object Layer)

2. Data Transfer Object (DTO)

Class ในกลุ่มนี้ทำหน้าที่เก็บค่าข้อมูลต่าง ๆ ซึ่งข้อมูลดังกล่าวนั้นอาจจะได้รับมาจากการคัดลอกข้อมูลของ class ในกลุ่ม Action Form หรืออาจจะได้มาจากการดึงข้อมูลของ Entity Beans หลังจากนั้น class ในกลุ่มนี้จะถูกส่ง request ออกไปยังหน้าเว็บเพจ เพื่อนำข้อมูลที่มีอยู่แสดงให้กับ administrator หรือ user ต่อไป

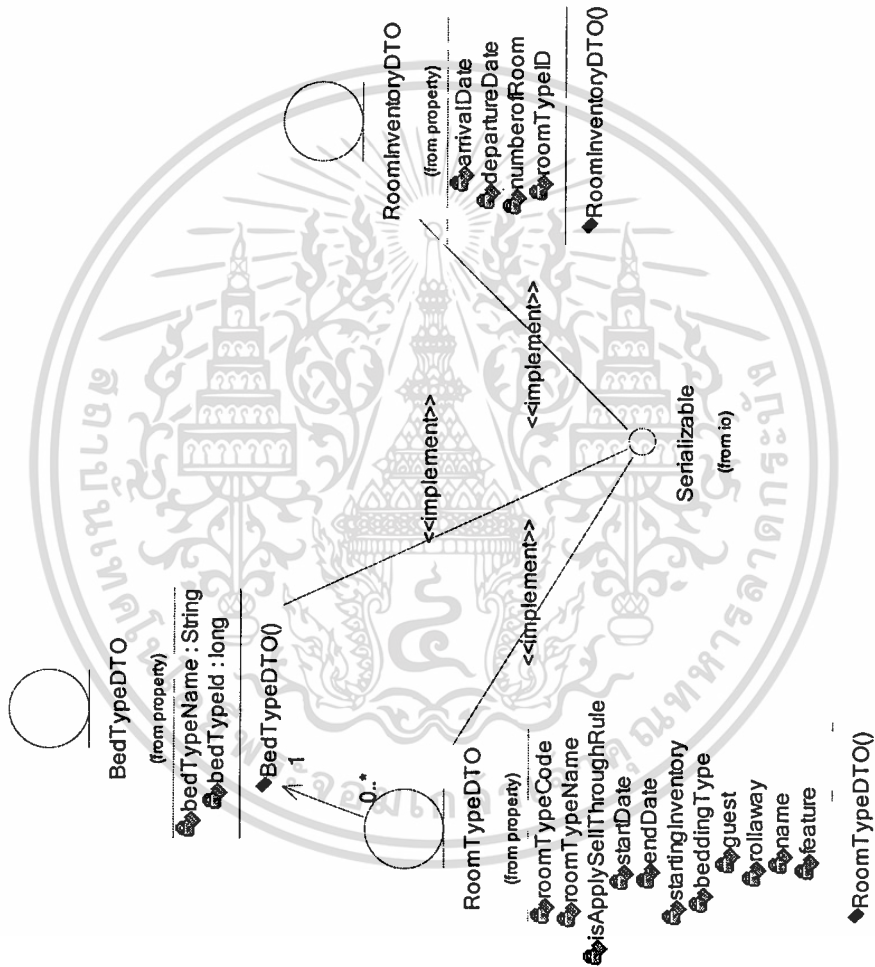
Class ในกลุ่มนี้จะแบ่งย่อยออกเป็น 5 package ดังนี้คือ

Common Package



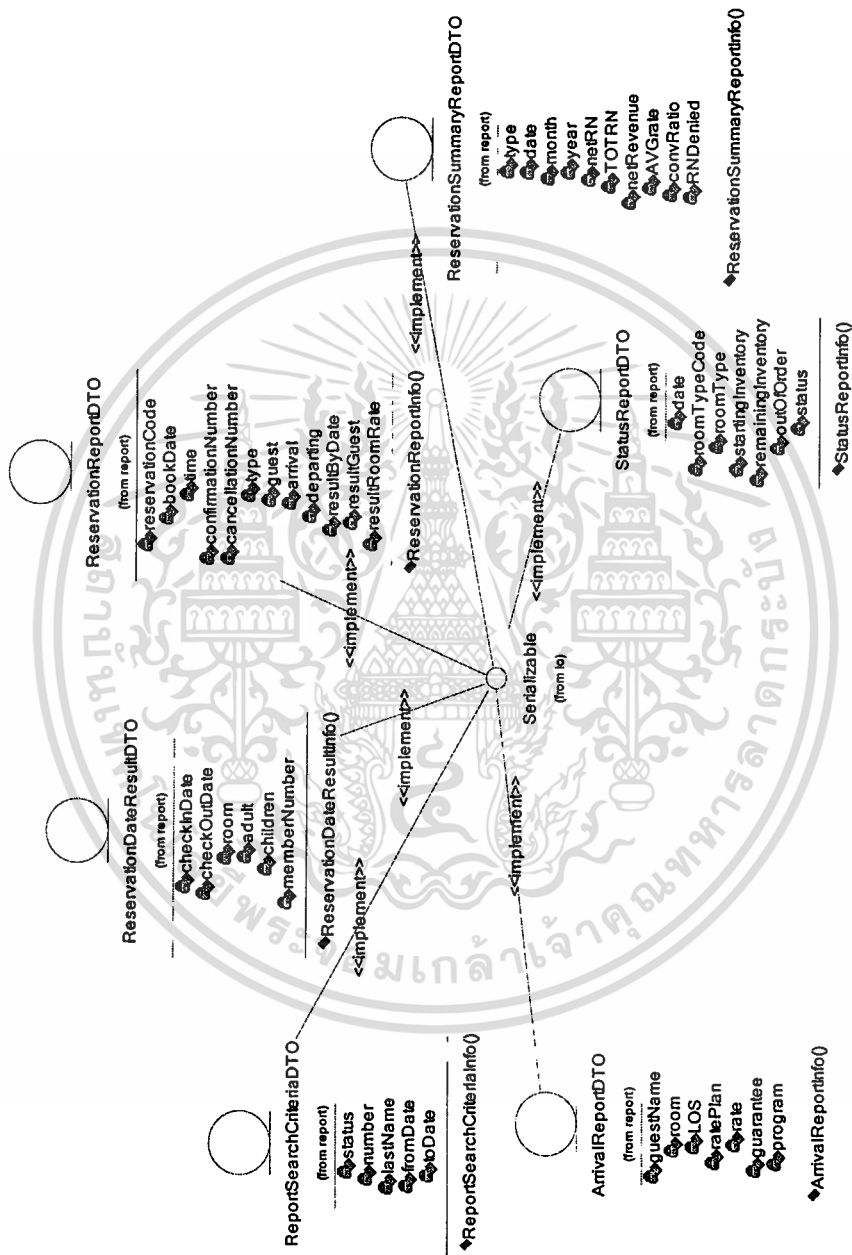
รูปที่ 3.11 Class Diagram ใน Common Package (Common Business Object Layer)

Property Package



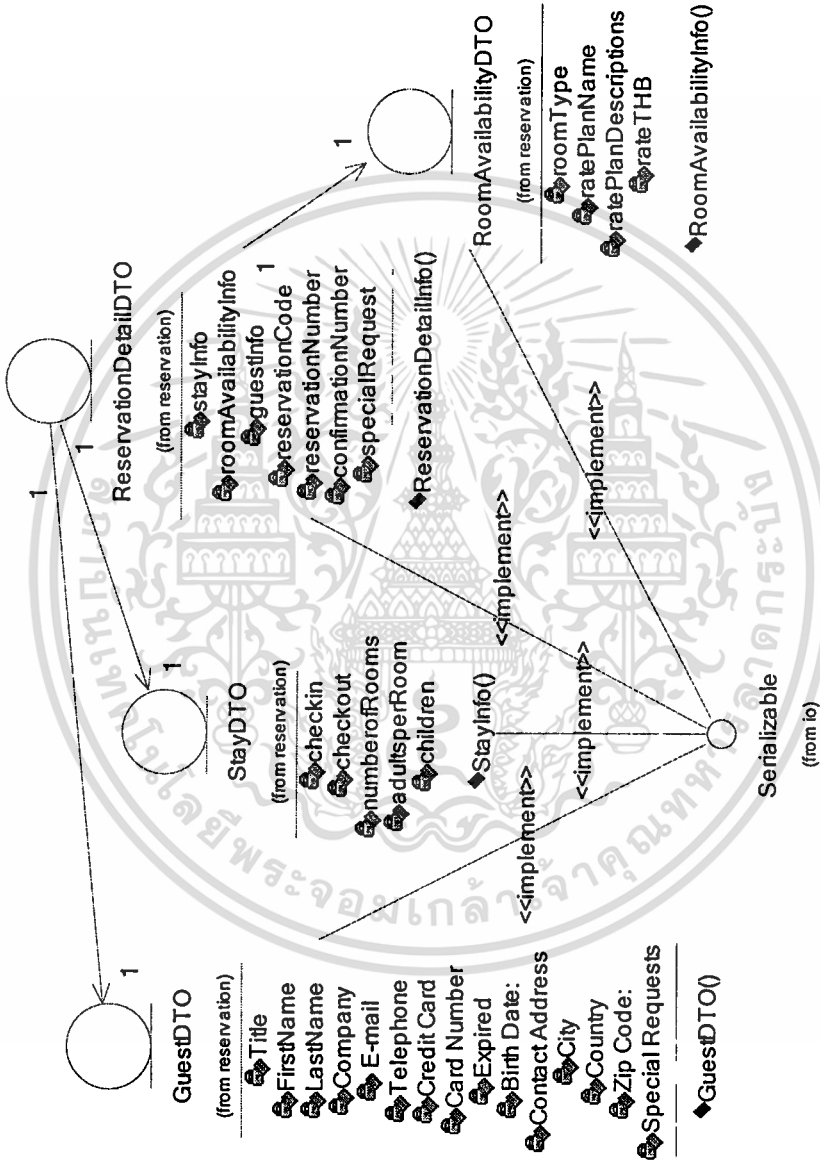
รูปที่ 3.12 Class Diagram ใน Property Package (Common Business Object Layer)

Report Package



รูปที่ 3.14 Class Diagram ใน Report Package (Common Business Object Layer)

Reservation Package



รูปที่ 3.15 Class Diagram ใน Reservation Package (Common Business Object Layer)

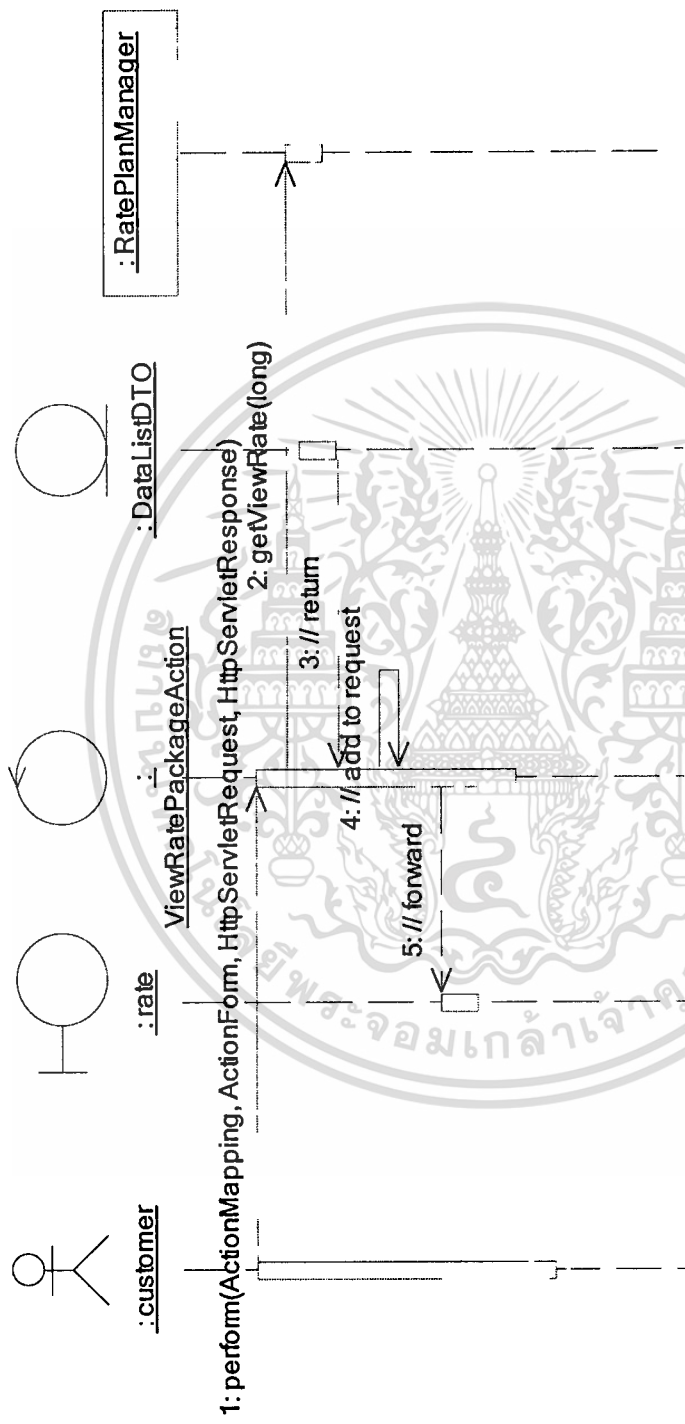
3.4 Sequence Diagram

Sequence diagram จะแสดงการทำงานของโปรแกรมว่าเกิดการทำงานจากส่วนใดก่อน เรียงลำดับกันไป โดยจะแบ่งออกเป็นกลุ่ม ๆ ตาม use case diagram ดังนี้ View Rate Plan, Make Reservation, Manage Reservation, Manage Room Type, Manage Rate Plan, View Reservation Report, View Room Inventory Report, View Monthly Summary Report, View Arrival Report, Update Inventory, Manage Customer's Reservation

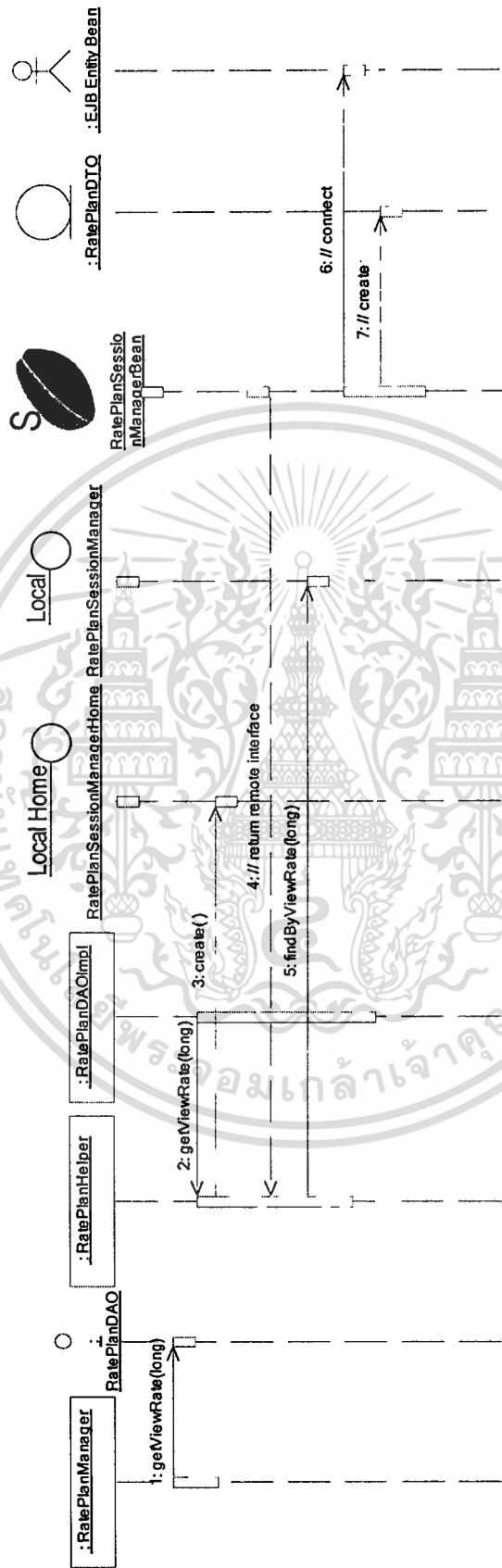
ในที่นี้จะขอแสดงรายละเอียดของ View Rate Plan, Make Reservation และ Manage Room Type เท่านั้นซึ่งรายละเอียดแสดงในรูปที่ 3.16 – 3.29



View Rate Plan Sequence Diagram



รูปที่ 3.16 View Rate Plan Sequence Diagram



รูปที่ 3.17 View Rate Plan Sequence Diagram (ต่อ)

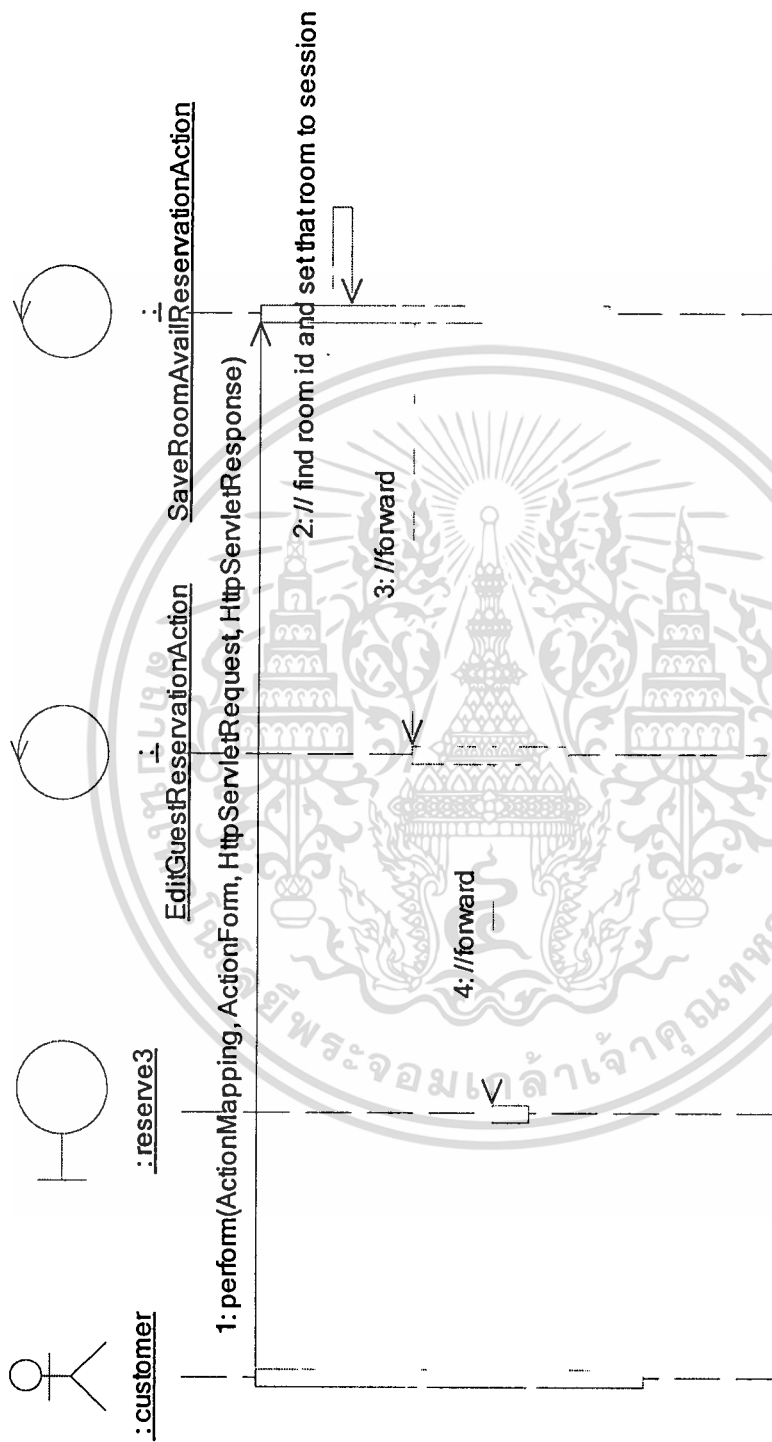
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.16 และ 3.17 เริ่มจากเมื่อ Customer ต้องการที่จะดูรายละเอียดของราคา Customer จะเรียก class ที่ชื่อว่า ViewRatePackageAction จากนั้น class นี้จะเรียก method getViewRate() จาก class RatePlanManager จากนั้น RatePlanManager จะเรียก method getViewRate() จาก interface class RatePlanDAO ซึ่งถูก implement โดย class RatePlanDAOImpl ซึ่งจะเรียก method getViewRate() จาก class RatePlanHelper จากนั้น RatePlanHelper จะทำการสร้าง connection ไปเรียก Session Bean ที่ชื่อว่า RatePlanSessionManager ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ RatePlanSessionManager จากนั้นจึงเรียก method findByViewRate() ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อดึงข้อมูลจากฐานข้อมูลผ่าน Entity Bean และเก็บข้อมูลที่ได้ไว้ใน class RatePlanDTO เมื่อได้ข้อมูลมา class นี้จะเก็บ object ของตัวมันเองไว้ใน class DataListDTO ซึ่งจะถูกส่งออกมาผ่าน request มายังหน้า rate.jsp เพื่อแสดงผล



จากรูปที่ 3.18 และ 3.19 Customer ต้องการที่จะสำรองห้องพักจึงเรียก class EditStayReservationAction ซึ่ง class นี้จะ forward ไปยังหน้า reserve1.jsp เพื่อให้ Customer ใส่ข้อมูล เมื่อ Customer ใส่ข้อมูลเสร็จเรียบร้อยแล้วก็จะเรียก class SaveStayReservationAction จากนั้น class นี้จะดึงค่าข้อมูลที่ Customer กรอกไว้ผ่าน class StayForm และทำการ initiate class StayDTO เพื่อทำการ copy ค่าจาก StayForm ให้กับ StayDTO ซึ่งเป็น method copyProperties() จาก class ServiceInfoUtil เมื่อ copy ได้แล้วจึงเก็บ object ของ StayDTO ไว้ใน session แล้ว forward ไปเรียก class EditRoomAvailReservationAction จากนั้น class นี้จะเรียก method checkRoomAvailabilityList() จาก class ReservationManager ที่คลาสนี้จะทำการดึง object StayDTO ที่เก็บไว้ใน session ออกมาเพื่อเป็น parameter ให้กับ method listRoomAvailability() ของ interface class ResevationDAO ซึ่งถูก implement โดย class ReservationDAOImpl ที่ class นี้จะเรียก method listRoomAvailability() ที่อยู่ใน class ReservationHelper ซึ่งจะทำการ สร้าง connection ไปเรียก Session Bean ที่ชื่อว่า ReservationSessionManager ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ ReservationSessionManager จากนั้นจึงเรียก method findByRoomAvailabilityList() ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อดึงข้อมูลจากฐานข้อมูลผ่าน Entity Bean และเก็บข้อมูลที่ได้ไว้ที่ class RoomAvailabilityDTO เมื่อได้ข้อมูลมา class นี้จะเก็บ object ของตัวมันเองไว้ที่ class DataListDTO ซึ่งจะถูกส่งออกมาผ่าน request มายังหน้า reserve2.jsp เพื่อแสดงผล

Select Room Availability

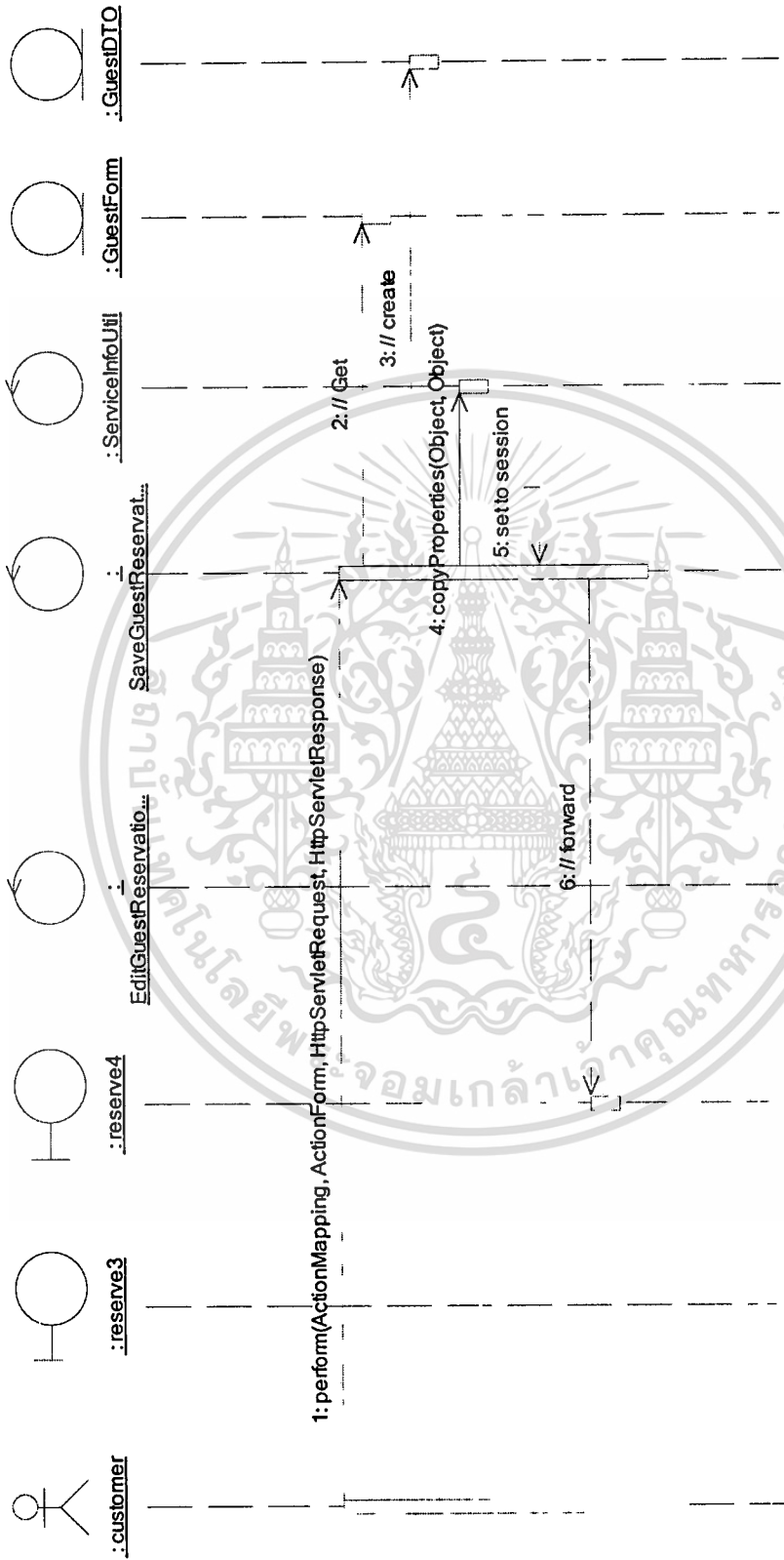


รูปที่ 3.20 Make Reservation Sequence Diagram : Select Room Availability

จากรูปที่ 3.20 Customer ทำการเลือกห้องที่ตนเองต้องการเมื่อเลือกได้ระบบจะเรียกให้ class `SaveRoomAvailReservationAction` ทำงานจากนั้นจะมีการค้นหา room object ที่ Customer ได้เลือกไว้จาก list ของ room object เมื่อเจอจึงเก็บ object นั้นลง session ไว้ จากนั้นจึง forward ไปยัง class `EditGuestReservationAction` ซึ่งที่ class นี้จะ forward ออกไปยังหน้า `reserve3.jsp` อีกที่หนึ่ง



Make Guest Info

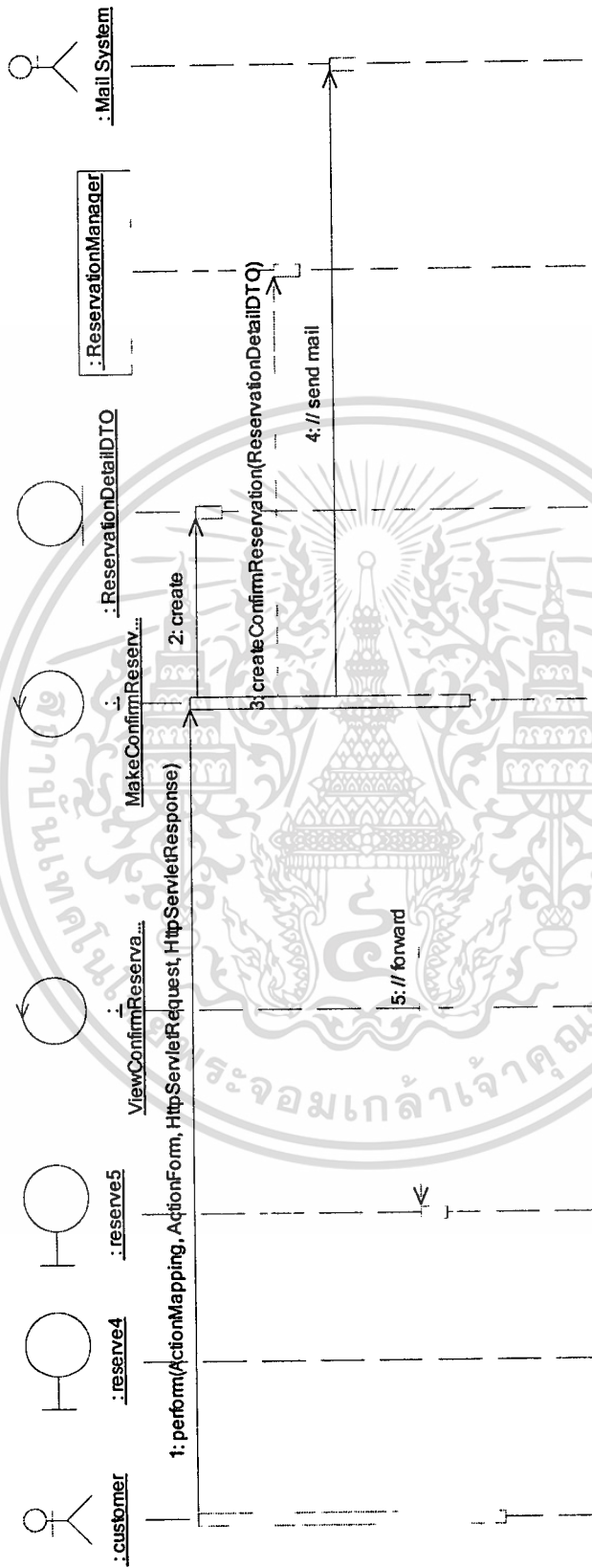


รูปที่ 3.21 Make Reservation Sequence Diagram : Make Guest Info

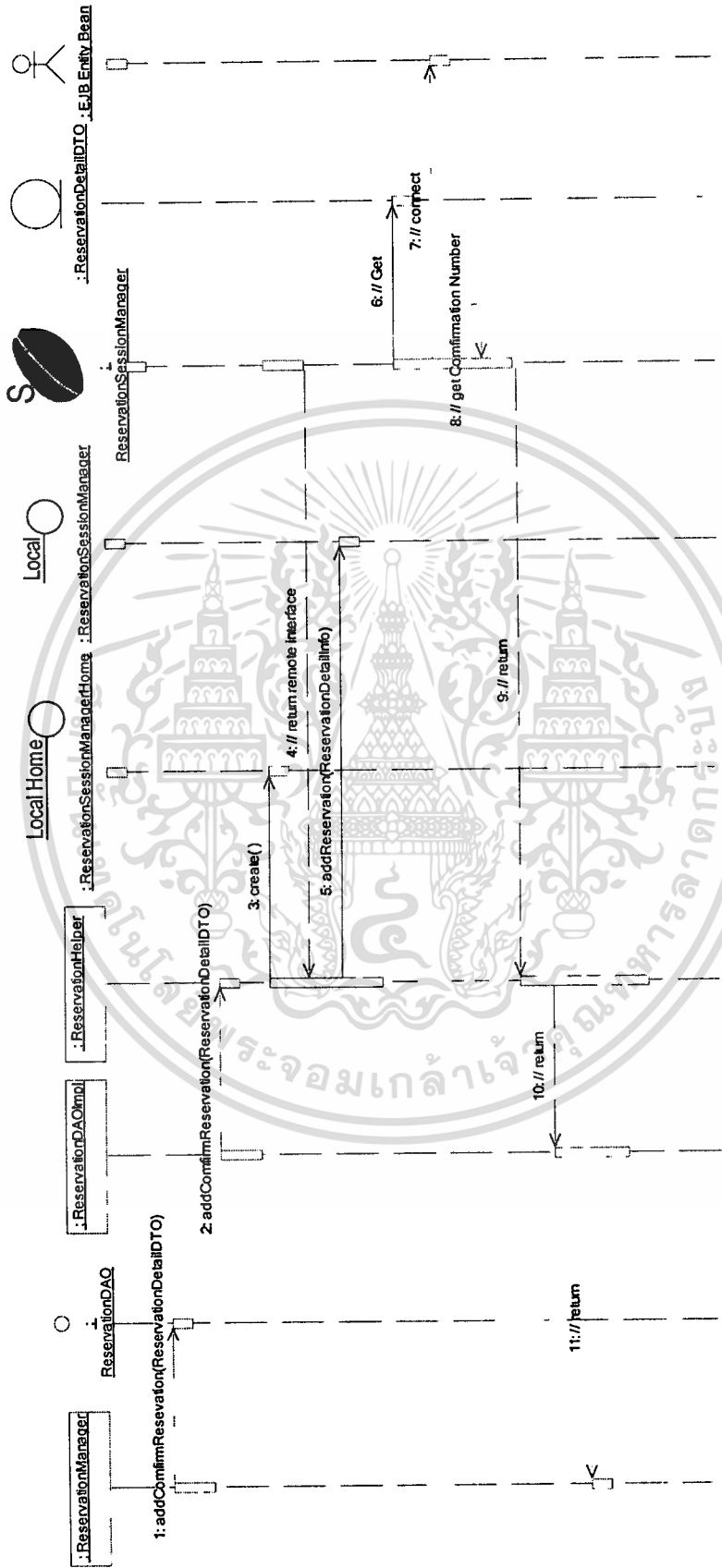
จากรูปที่ 3.21 เมื่อ Customer ทำการกรอกข้อมูลเกี่ยวกับตนเองเสร็จแล้วนั้นเมื่อทำการส่งข้อมูล class `SaveGuestReservationAction` จะถูกเรียกให้ทำงาน จากนั้นจะทำการดึงข้อมูลที่ Customer ได้กรอกเข้ามาซึ่งเก็บไว้ใน class `GuestForm` จากนั้นจึง instantiate class `GuestDTO` แล้วจึง copy ข้อมูลที่อยู่ใน object `GuestForm` มาเก็บไว้ที่ object `GuestDTO` ซึ่งกระทำผ่าน method `copyProperties()` ของ class `ServiceInfoUtil` จากนั้นจึงเก็บ object `GuestDTO` ใน session แล้วจึง forward ออกหน้า `reserve4.jsp`



Confirm Reservation



รูปที่ 3.22 Make Reservation Sequence Diagram : Confirm Reservation



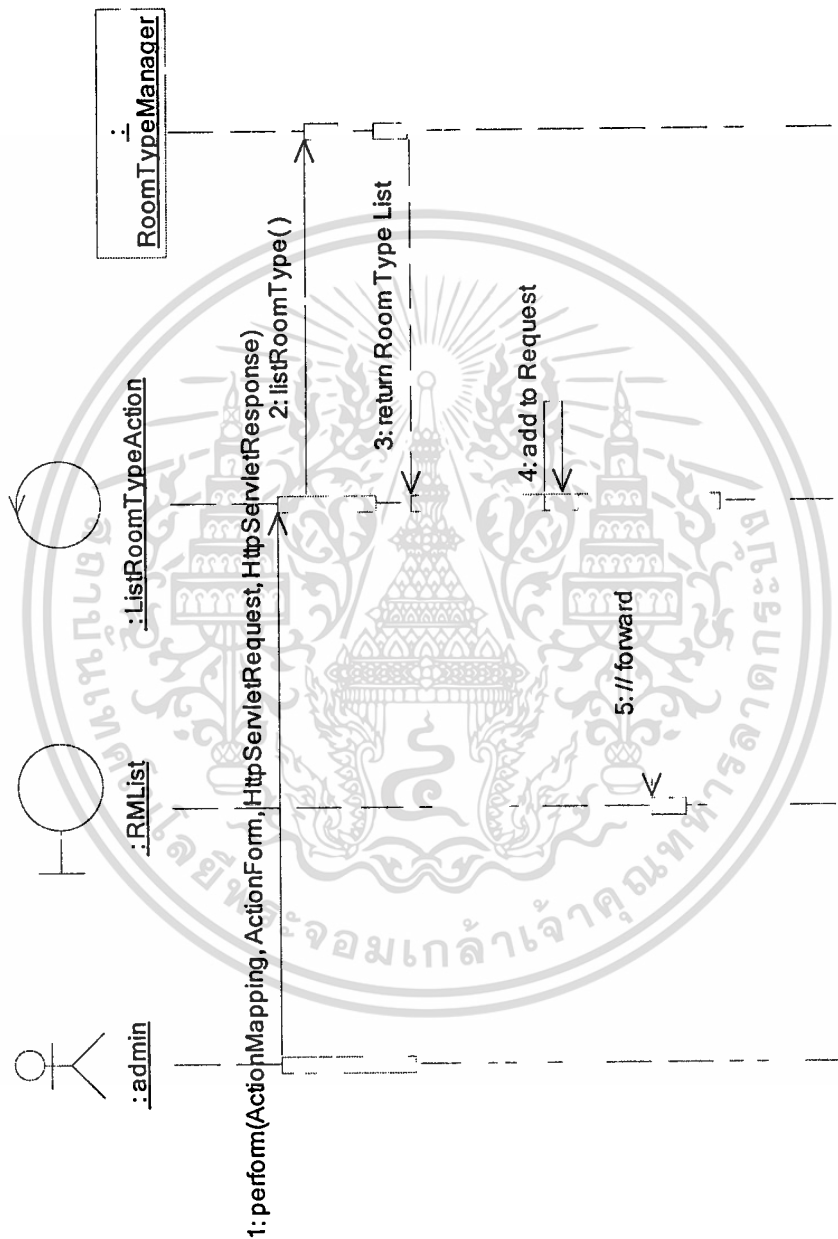
รูปที่ 3.23 Make Reservation Sequence Diagram : Confirm Reservation (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

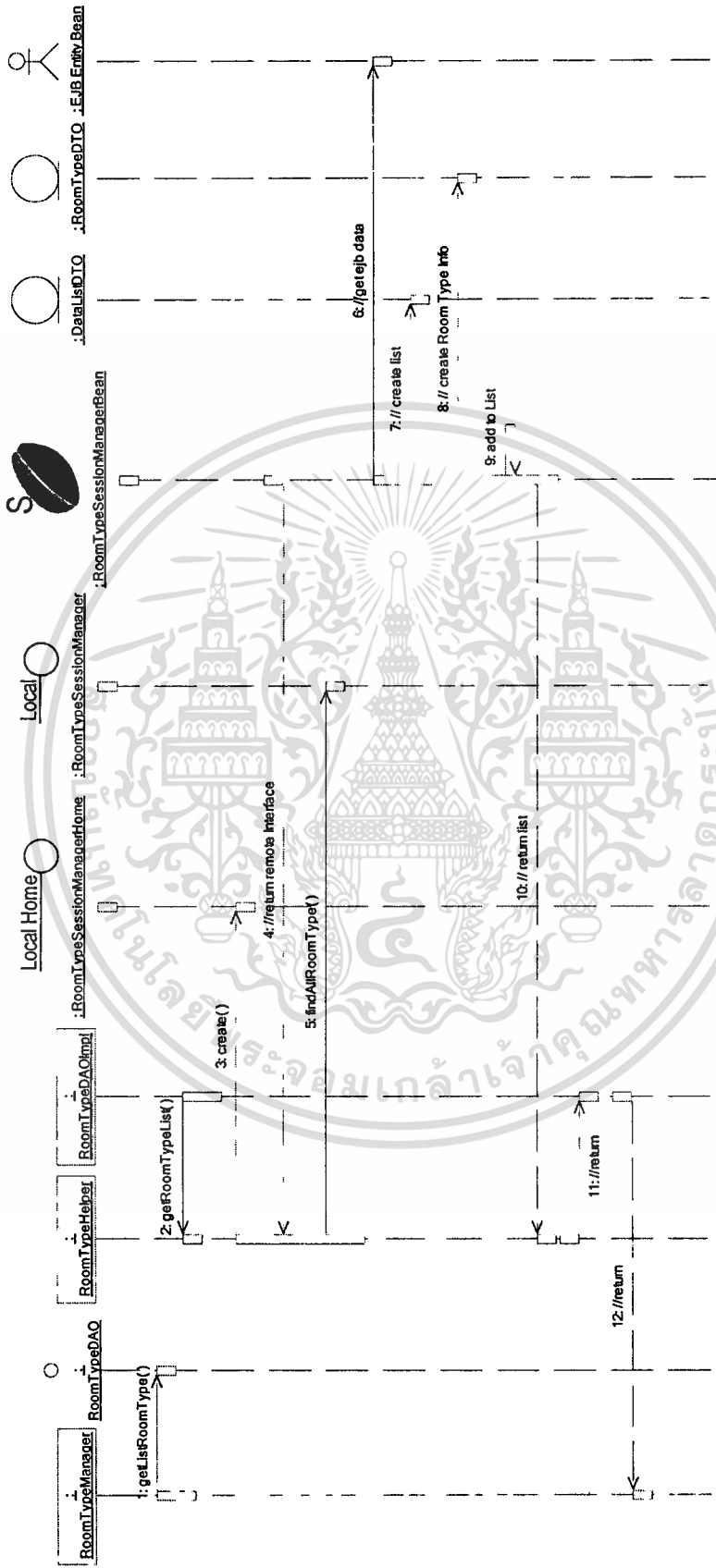
จากรูปที่ 3.22 และ 3.23 Customer ทำการยืนยันการจองห้องพักจึงเรียก class MakeConfirmReservationAction ซึ่ง class นี้จะทำการนำ object ต่าง ๆ ที่เก็บไว้ใน session มาเก็บไว้ใน class ReservationDetailDTO จากนั้นจึงเรียก method createConfirmReservation() จาก class ReservationManager ที่คลาสนี้จะเรียก method addConfirmReservation() จาก interface class ReservationDAO ซึ่งถูก implement ด้วย class ReservationDAOImpl ที่ class นี้จะเรียก method addConfirmReservation() ที่ class ReservationHelper ซึ่งจะทำการ สร้าง connection ไปเรียก Session Bean ที่ชื่อว่า ReservationSessionManager ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ ReservationSessionManager จากนั้นจึงเรียก method addReservation() ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อบันทึกข้อมูลลงฐานข้อมูลผ่าน Entity Bean และคืน confirmation number ที่เกิดจากการจองส่งกลับไป จากนั้นระบบจะทำการส่ง e-mail ไปให้กับ Customer และ forward มายังหน้า reserve5.jsp เพื่อแสดงผล

Manage Room Type Sequence Diagram

List Room Type



รูปที่ 3.24 Manage Room Type Sequence Diagram : List Room Type

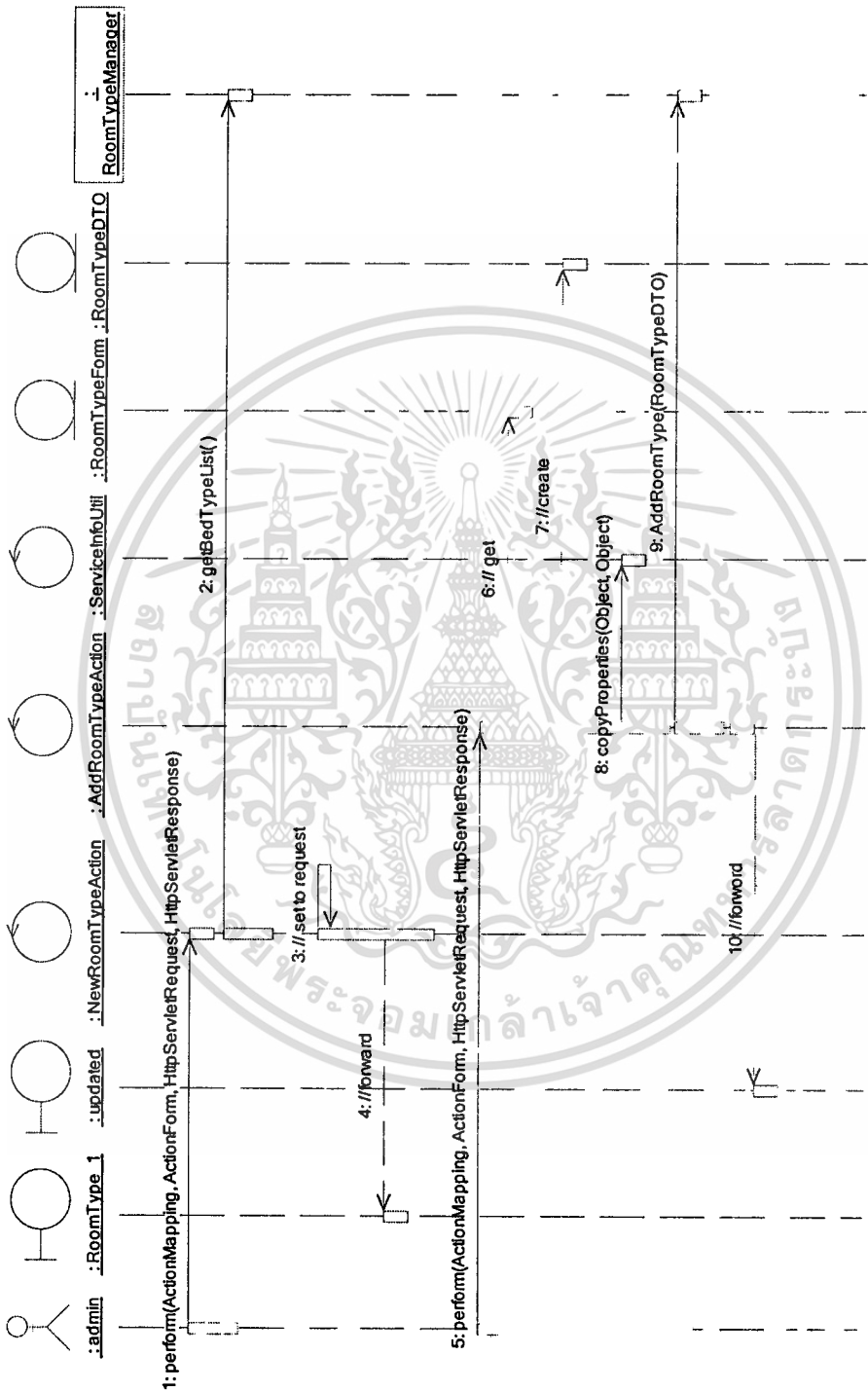


รูปที่ 3.25 Manage Room Type Sequence Diagram : List Room Type (ต่อ)

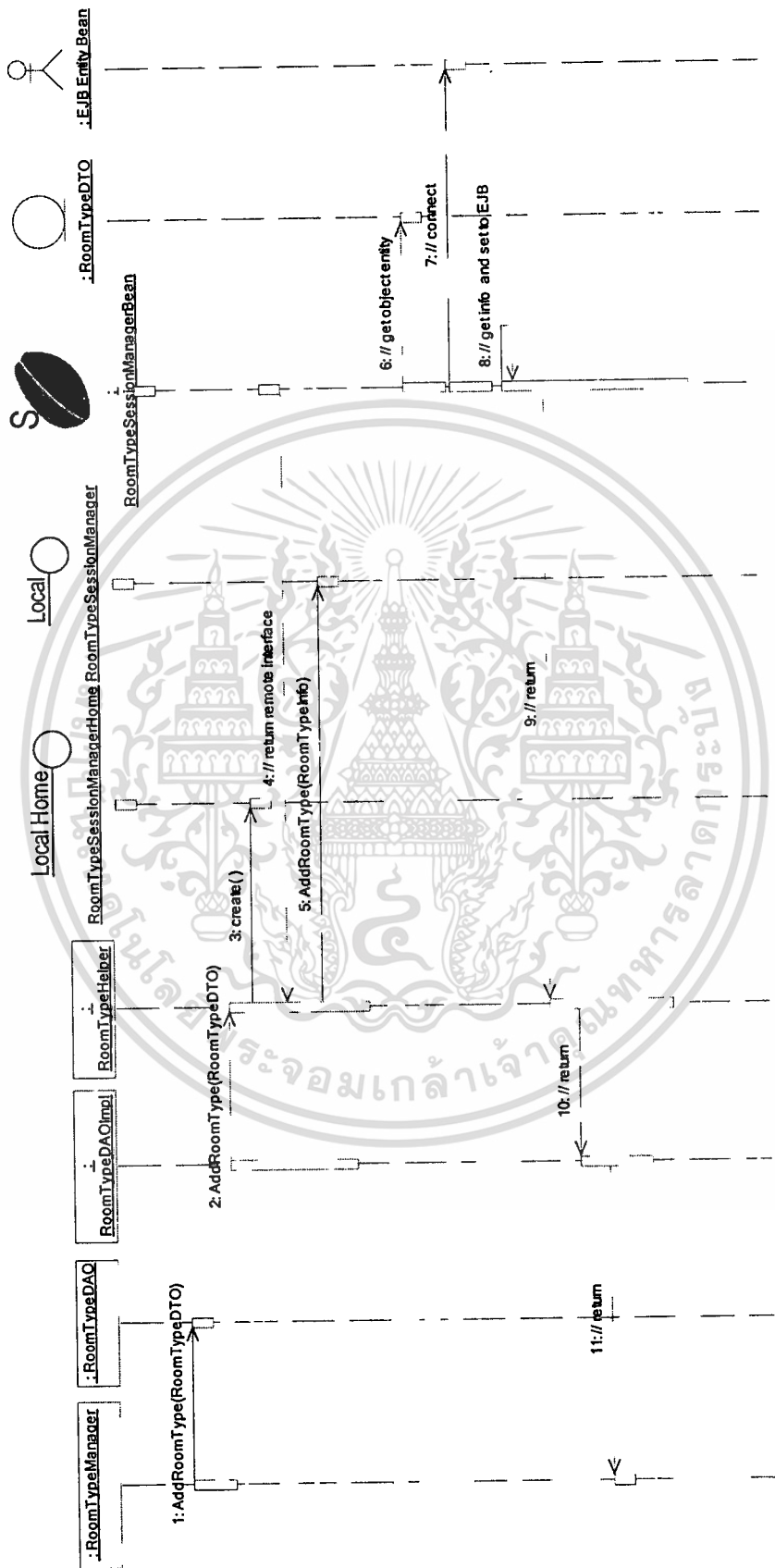
จากรูปที่ 3.24 และ 3.25 admin ต้องการที่จะเข้าไปจัดการข้อมูลเกี่ยวกับห้องพัก จึงเรียก class ListRoomTypeAction ซึ่ง class นี้จะเรียก method listRoomType() จาก class RoomTypeManager ที่คลาสนี้จะเรียก method getRoomTypeList() จาก interface class RoomTypeDAO ซึ่งถูก implement ด้วย class RoomTypeDAOImpl ที่ class นี้จะเรียก method getRoomTypeList() ที่ class RoomTypeHelper ซึ่งจะทำการ สร้าง connection ไปเรียก Session Bean ที่ชื่อว่า RoomTypeSessionManager ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ RoomTypeSessionManager จากนั้นจึงเรียก method findAllRoomType() ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อดึงข้อมูลจากฐานข้อมูลผ่าน Entity Bean และทำการเก็บข้อมูลที่ได้ลงใน class RoomTypeDTO แล้วจึงเก็บ object ของ class RoomTypeDTO ลงใน class DataListDTO และส่ง และเก็บไว้ใน request เพื่อส่งออกไปยังหน้า RMList.jsp



Add Room Type



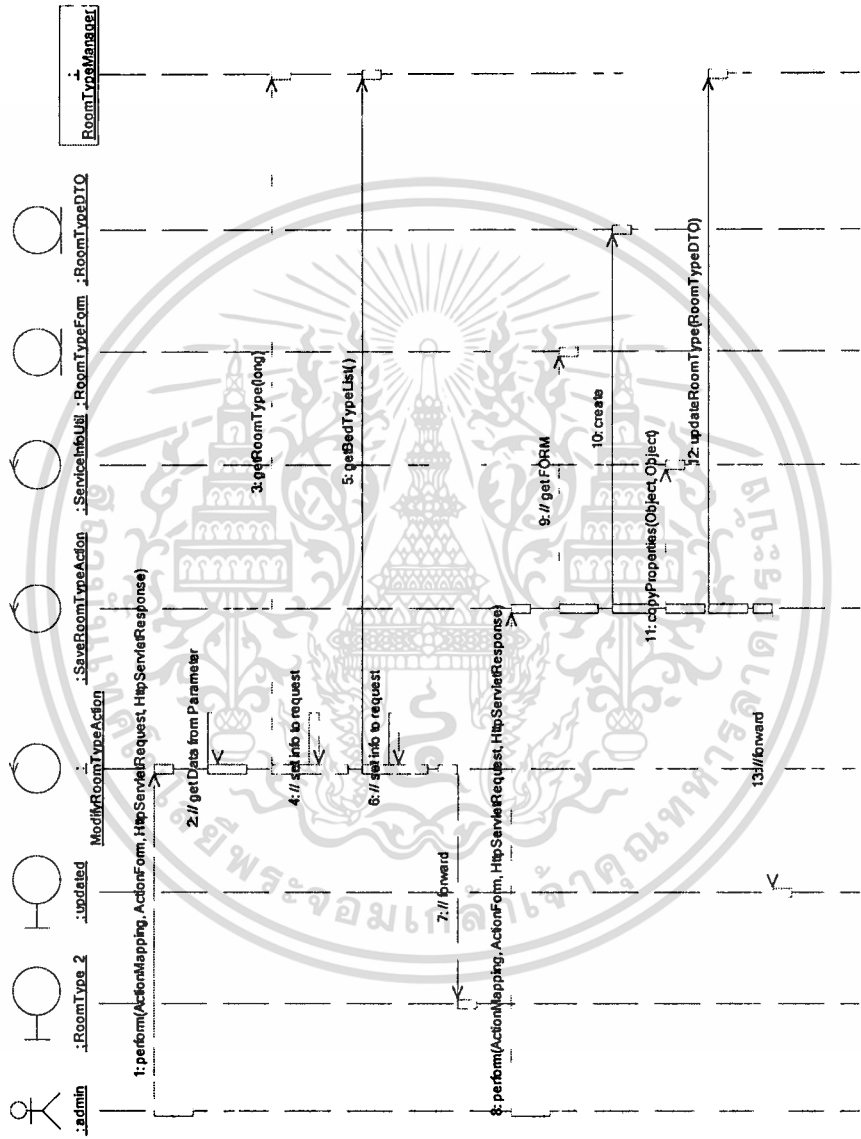
รูปที่ 3.26 Manage Room Type Sequence Diagram : Add Room Type



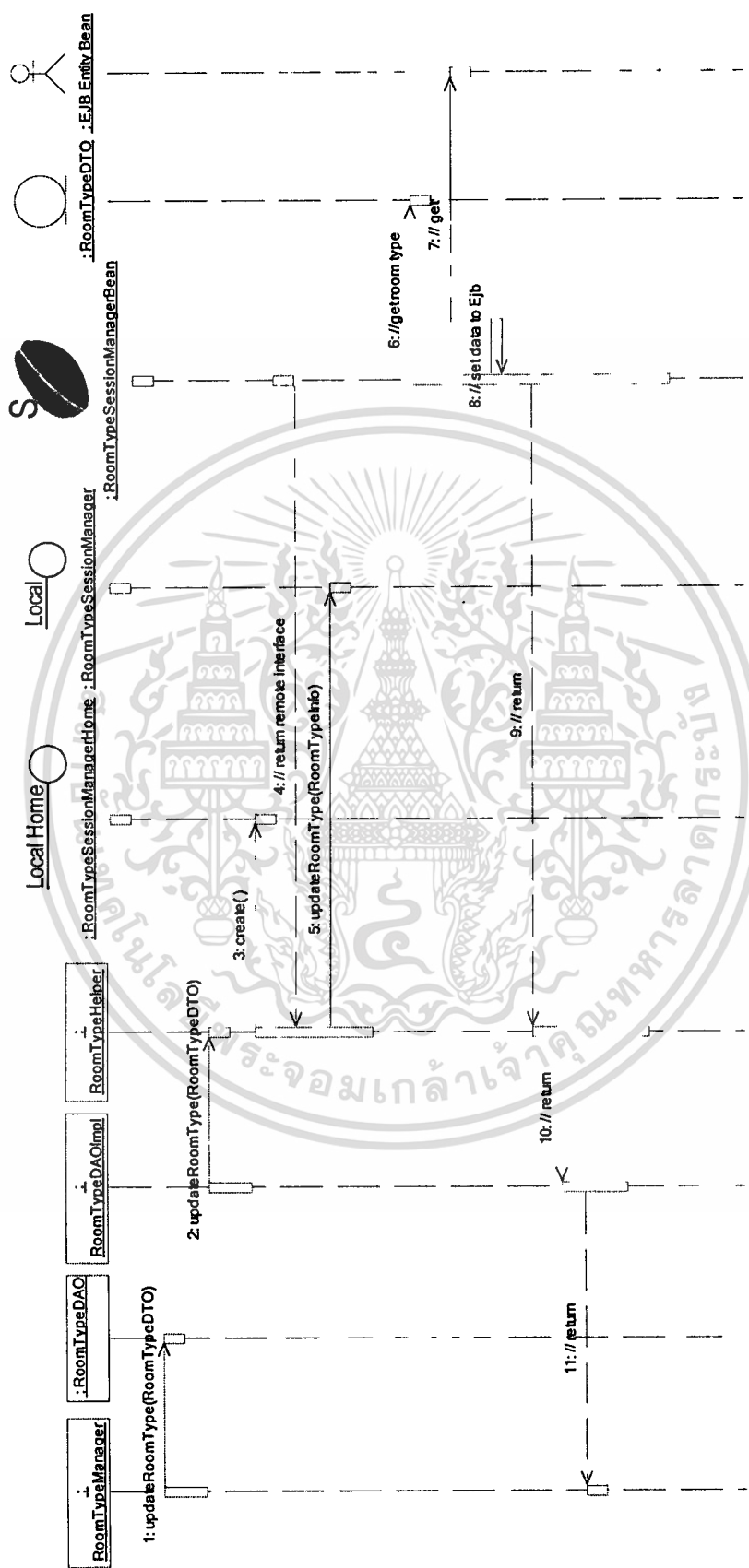
รูปที่ 3.27 Manage Room Type Sequence Diagram : Add Room Type (ต่อ)

จากรูปที่ 3.26 และ 3.27 admin ต้องการที่จะเพิ่มประเภทของห้องพักจึงเรียก class `NewRoomTypeAction` ที่คลาสนี้จะเรียก method `getBedTypeList()` ที่อยู่ใน class `RoomTypeManager` เพื่อดึงเอาข้อมูลเกี่ยวกับประเภทเตียงนอนออกมาหลังจากนั้นจึง forward ออกไปยังหน้า `RoomType_1.jsp` เมื่อ admin ได้กรอกข้อมูลจนครบจึงเรียก class `AddRoomTypeAction` ที่ class นี้จะทำการดึงข้อมูลที่ admin ได้กรอกมาผ่าน class `RoomTypeForm` จากนั้นทำการ instantiate class `RoomTypeDTO` แล้วทำการเรียก method `copyProperties()` จาก class `ServiceInfoUtil` เพื่อคัดลอกค่าจาก `RoomTypeForm` ไปยัง `RoomTypeDTO` แล้วจึงเรียก method `AddRoomType()` จาก class `RoomTypeManager` ที่คลาสนี้จะเรียก method `addRoomType()` จาก interface class `RoomTypeDAO` ซึ่งถูก implement ด้วย class `RoomTypeDAOImpl` ที่ class นี้จะเรียก method `addRoomType()` ที่ class `RoomTypeHelper` ซึ่งจะทำการสร้าง connection ไปเรียก Session Bean ที่ชื่อว่า `RoomTypeSessionManager` ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ `RoomTypeSessionManager` จากนั้นจึงเรียก method `addRoomType()` ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อบันทึกข้อมูลลงฐานข้อมูลผ่าน Entity Bean จากนั้นระบบจะทำการ forward มายังหน้า `updated.jsp` เพื่อแสดงผล

Edit Room Type



รูปที่ 3.28 Manage Room Type Sequence Diagram : Edit Room Type



รูปที่ 3.29 Manage Room Type Sequence Diagram : Edit Room Type (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.28 และ 3.29 admin ต้องการที่จะแก้ไขข้อมูลของห้องพักจึงเรียก class ModifyRoomTypeAction ที่คลาสนี้จะเรียก method getRoomType() และ getBedType() ที่อยู่ใน class RoomTypeManager เพื่อดึงเอาข้อมูลเกี่ยวกับห้องพักนั้น ๆ และประเภทเตียงนอนออกมาหลังจากนั้นจึง forward ออกไปยังหน้า RoomType_2.jsp เมื่อ admin ได้แก้ไขข้อมูลจนครบจึงเรียก class SaveRoomTypeAction ที่ class นี้จะทำการดึงข้อมูลที่ admin ได้กรอกมาผ่าน class RoomTypeForm จากนั้นทำการ instantiate class RoomTypeDTO แล้วทำการเรียก method copyProperties() จาก class ServiceInfoUtil เพื่อคัดลอกค่าจาก RoomTypeForm ไปยัง RoomTypeDTO แล้วจึงเรียก method updateRoomType() จาก class RoomTypeManager ที่คลาสนี้จะเรียก method updateRoomType() จาก interface class RoomTypeDAO ซึ่งถูก implement ด้วย class RoomTypeDAOImpl ที่ class นี้จะเรียก method updateRoomType() ที่ class RoomTypeHelper ซึ่งจะทำการสร้าง connection ไปเรียก Session Bean ที่ชื่อว่า RoomTypeSessionManager ซึ่งสิ่งที่ได้กลับมาคือ object ของ remote interface ที่ชื่อ RoomTypeSessionManager จากนั้นจึงเรียก method updateRoomType() ที่ remote interface ตัวนั้น ซึ่งจะทำการติดต่อเพื่อบันทึกข้อมูลลงฐานข้อมูลผ่าน Entity Bean จากนั้นระบบจะทำการ forward มายังหน้า updated.jsp เพื่อแสดงผล

3.5 เครื่องมือที่เลือกใช้ในการพัฒนาระบบ

หลังจากที่ได้ทำการวิเคราะห์ และออกแบบระบบจนเสร็จแล้วนั้นเป็นได้ว่าระบบนี้เป็นการพัฒนาในรูปแบบของ web application โดยใช้เทคโนโลยี JSP ซึ่งใช้ภาษาจาวาในการ implement อีกทั้งจำเป็นต้องสร้าง EJB เพื่อทำการติดต่อกับฐานข้อมูล ดังนั้นเครื่องมือ และอุปกรณ์ที่เลือกใช้ในการพัฒนาระบบมีดังนี้

1. Server (OS : Windows 2000 Server/Windows XP Pro)
2. Cilent (OS : Windows 2000 Pro/Windows XP Home/Pro)
3. J2SDK 1.4.2_03
4. WebSphere Application Server 5 fixed pack 2
5. IBM WebSphere Studio Application Developer 5.1
6. Oracle 9i Release 2
7. Macromedia Dreamweaver MX
8. Adobe Photoshop 7

บทที่ 4

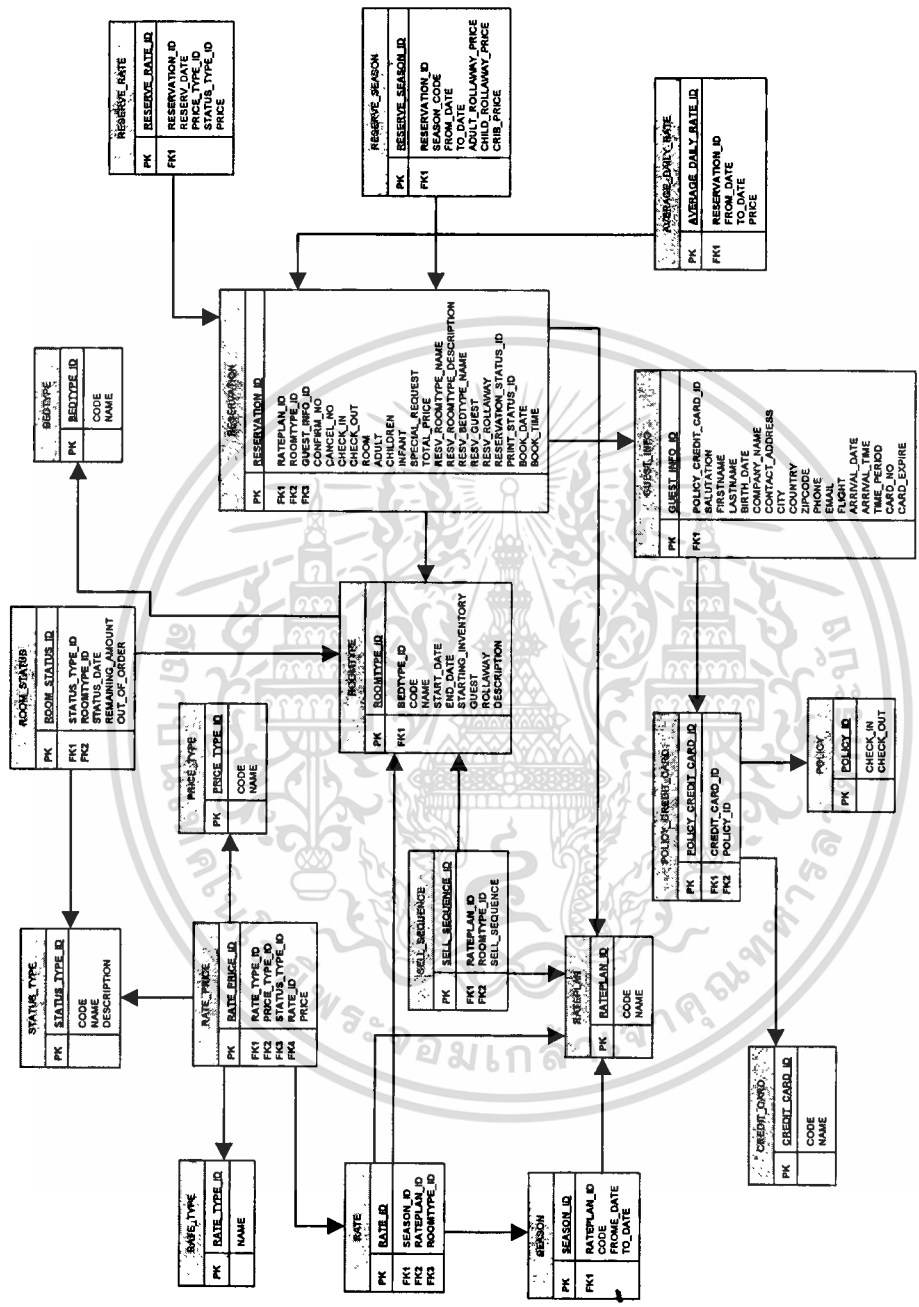
การวิเคราะห์ และออกแบบฐานข้อมูล

4.1 ลักษณะการออกแบบฐานข้อมูล

แบบจำลองข้อมูล คือแบบจำลองที่ใช้สำหรับอธิบายถึงโครงสร้าง และความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล จากรูปแบบที่เป็นแนวความคิดที่ยากแก่การทำความเข้าใจให้อยู่ในรูปแบบที่สามารถเข้าใจ และจับต้องได้ง่ายขึ้น ดังนั้นแบบจำลองของฐานข้อมูล จึงมักถูกนำไปใช้อธิบายถึงโครงสร้างของฐานข้อมูลประเภทต่าง ๆ

จากการศึกษา และวิเคราะห์ปัญหาทั้งหมดของระบบโรงแรมจึงได้แบบจำลองของข้อมูล ดังแสดงในรูปที่ 4.1





รูปที่ 4.1 Database Model Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 เป็นมุมมองที่มีต่อข้อมูลในสถาปัตยกรรมในระดับ Internal ซึ่งสามารถแปลงเป็นข้อมูลในระดับกายภาพเพื่อใช้ในการจัดเก็บข้อมูลต่อไป

4.2 Data Dictionary

ตารางที่ 4.1 ตาราง RATE_TYPE

Table Name : RATE_TYPE				
Primary Key : RATE_TYPE_ID				
ลำดับที่	Attribute	Type	Width	Description
1	RATE_TYPE_ID	Number	3,0	หมายเลขประเภทของอัตรา
2	NAME	VarChar2	64	ชื่อของประเภทราคา

ตารางที่ 4.2 ตาราง SEASON

Table Name : SEASON				
Primary Key : SEASON_ID				
Foreign Key : RATEPLAN_ID reference RATEPLAN table				
ลำดับที่	Attribute	Type	Width	Description
1	SEASON_ID	Number	10,0	หมายเลขของฤดูกาล
2	CODE	VarChar2	16	ชื่อของฤดูกาล
3	RATEPLAN_ID	Number	3,0	หมายเลขของแผนราคา
4	FROM_DATE	Date	-	วันเริ่มต้นของฤดู
5	TO_DATE	Date	-	วันสิ้นสุดของฤดู

ตารางที่ 4.3 ตาราง RATE_PRICE

Table Name : RATE_PRICE				
Primary Key : RATE_PRICE_ID				
Foreign Key : RATE_ID reference RATE table				
Foreign Key : RATE_TYPE_ID reference RATE_TYPE table				
Foreign Key : PRICE_TYPE_ID reference PRICE_TYPE table				
Foreign Key : STATUS_TYPE_ID reference STATUS_TYPE table				
ลำดับที่	Attribute	Type	Width	Description
1	RATE_PRICE_ID	Number	10,0	หมายเลขของอัตราราคา
2	RATE_ID	Number	10,0	หมายเลขของอัตรา
3	RATE_TYPE_ID	Number	3,0	หมายเลขประเภทของอัตรา
4	PRICE_TYPE_ID	Number	3,0	หมายเลขประเภทของราคา
5	STATUS_TYPE_ID	Number	3,0	หมายเลขประเภทของสถานะ
6	PRICE	Number	9,2	ราคา

ตารางที่ 4.4 ตาราง RATE

Table Name : RATE				
Primary Key : RATE_ID				
Foreign Key : RATEPLAN_ID reference RATEPLAN table				
Foreign Key : SEASON_ID reference SEASON table				
Foreign Key : ROOMTYPE_ID reference ROOMTYPE table				
ลำดับที่	Attribute	Type	Width	Description
1	RATE_ID	Number	10,0	หมายเลขของอัตรา
2	RATEPLAN_ID	Number	3,0	หมายเลขของแผนราคา
3	SEASON_ID	Number	10,0	หมายเลขฤดูกาล
4	ROOMTYPE_ID	Number	10,0	หมายเลขประเภทของห้องพัก

ตารางที่ 4.5 ตาราง RATEPLAN

Table Name : RATEPLAN				
Primary Key : RATEPLAN_ID				
ลำดับที่	Attribute	Type	Width	Description
1	RATEPLAN_ID	Number	3,0	หมายเลขของแผนราคา
2	CODE	VarChar2	16	รหัสของแผนราคา
3	NAME	VarChar2	64	ชื่อของแผนราคา

ตารางที่ 4.6 ตาราง SELL_SEQUENCE

Table Name : SELL_SEQUENCE				
Primary Key : SELL_SEQUENCE_ID				
Foreign Key : RATEPLAN_ID reference RATEPLAN table				
Foreign Key : ROOMTYPE_ID reference ROOMTYPE table				
ลำดับที่	Attribute	Type	Width	Description
1	SELL_SEQUENCE_ID	Number	10,0	หมายเลขของลำดับการขาย
2	RATEPLAN_ID	Number	3,0	หมายเลขของแผนราคา
3	ROOMTYPE_ID	Number	10,0	หมายเลขประเภทของห้องพัก
4	SELL_SEQUENCE	Number	2,0	ลำดับการขาย

ตารางที่ 4.7 ตาราง STATUS_TYPE

Table Name : STATUS_TYPE				
Primary Key : STATUS_TYPE_ID				
ลำดับที่	Attribute	Type	Width	Description
1	STATUS_TYPE_ID	Number	3,0	หมายเลขประเภทของสถานะ
2	CODE	VarChar2	16	รหัสของสถานะ
3	NAME	VarChar2	64	ชื่อของสถานะ
4	DESCRIPTION	VarChar2	256	คำอธิบายสถานะ

ตารางที่ 4.8 ตาราง PRICE_TYPE

Table Name : PRICE_TYPE				
Primary Key : PRICE_TYPE_ID				
ลำดับที่	Attribute	Type	Width	Description
1	PRICE_TYPE_ID	Number	3,0	หมายเลขประเภทของราคา
2	CODE	VarChar2	16	รหัสของประเภทของราคา
3	NAME	VarChar2	64	ชื่อของประเภทของราคา

ตารางที่ 4.9 ตาราง CREDIT_CARD

Table Name : CREDIT_CARD				
Primary Key : CREDIT_CARD_ID				
ลำดับที่	Attribute	Type	Width	Description
1	CREDIT_CARD_ID	Number	3,0	หมายเลขของบัตรเครดิต
2	CODE	VarChar2	16	ชื่อของบัตรเครดิต
3	NAME	VarChar2	64	ชื่อเต็มของบัตรเครดิต

ตารางที่ 4.10 ตาราง POLICY

Table Name : POLICY				
Primary Key : POLICY_ID				
ลำดับที่	Attribute	Type	Width	Description
1	POLICY_ID	Number	3,0	หมายเลขของนโยบาย
2	CHECK_IN	Date	-	เวลาเข้า
3	CHECK_OUT	Date	-	เวลาออก

ตารางที่ 4.11 ตาราง POLICY_CREDIT_CARD

Table Name : POLICY_CREDIT_CARD				
Primary Key : POLICY_CREDIT_CARD_ID				
ลำดับที่	Attribute	Type	Width	Description
1	POLICY_CREDIT_CARD_ID	Number	3,0	หมายเลขของนโยบายบัตรเครดิต
2	POLICY_ID	Number	3,0	หมายเลขของนโยบาย
3	CREDIT_CARD_ID	Number	3,0	หมายเลขของบัตรเครดิต

ตารางที่ 4.12 ตาราง ROOMTYPE

Table Name : ROOMTYPE				
Primary Key : ROOMTYPE_ID				
Foreign Key : BEDTYPE_ID reference BEDTYPE table				
ลำดับที่	Attribute	Type	Width	Description
1	ROOMTYPE_ID	Number	10,0	หมายเลขของประเภทห้องพัก
2	CODE	VarChar2	16	รหัสย่อของประเภทห้องพัก
3	NAME	VarChar2	32	ชื่อเต็มของประเภทห้องพัก
4	START_DATE	Date	-	วันเริ่มต้นใช้ห้อง
5	END_DATE	Date	-	วันสิ้นสุดใช้ห้อง
6	STARTING_INVENTORY	Number	3,0	จำนวนห้อง
7	BEDTYPE_ID	Number	3,0	หมายเลขของประเภทเตียง
8	GUEST	Number	3,0	จำนวนคนพักสูงสุด
9	ROLLAWAY	Number	3,0	จำนวนเตียงเสริม
10	DESCRIPTION	VarChar2	512	คำอธิบาย

ตารางที่ 4.13 ตาราง BEDTYPE

Table Name : BEDTYPE				
Primary Key : BEDTYPE_ID				
ลำดับที่	Attribute	Type	Width	Description
1	BEDTYPE_ID	Number	3,0	หมายเลขของประเภทเตียง
2	CODE	VarChar2	16	รหัสย่อของประเภทเตียง
3	NAME	VarChar2	32	ชื่อของประเภทเตียง

ตารางที่ 4.14 ตาราง ROOM_STATUS

Table Name : ROOM_STATUS				
Primary Key : ROOM_STATUS_ID				
Foreign Key : ROOMTYPE_ID reference ROOMTYPE table				
Foreign Key : STATUS_TYPE_ID reference STATUS_TYPE table				
ลำดับที่	Attribute	Type	Width	Description
1	ROOM_STATUS_ID	Number	10,0	หมายเลขของสถานะห้อง
2	STATUS_DATE	Date	-	วันเวลา
3	ROOMTYPE_ID	Number	32	หมายเลขของประเภทห้องพัก
4	REMAINING_AMOUNT	Number	3,0	จำนวนห้องที่เหลือ
5	OUT_OF_ORDER	Number	3,0	สถานะถ้าห้องหมด
6	STATUS_TYPE_ID	Number	3,0	หมายเลขประเภทของสถานะ

ตารางที่ 4.15 ตาราง RESERVATION

Table Name : RESERVATION				
Primary Key : RESERVATION_ID				
Foreign Key : RATEPLAN_ID reference RATEPLAN table				
Foreign Key : ROOMTYPE_ID reference ROOMTYPE table				
Foreign Key : GUEST_INFO_ID reference GUEST_INFO table				
ลำดับที่	Attribute	Type	Width	Description
1	RESERVATION_ID	Number	10,0	หมายเลขของการทำ การจอง
2	RATEPLAN_ID	Number	3,0	หมายเลขของแผน ราคา
3	ROOMTYPE_ID	Number	10,0	หมายเลขของประเภท ห้องพัก
4	GUEST_INFO_ID	Number	10,0	หมายเลขของข้อมูลผู้ พัก
5	CONFIRM_NO	VarChar2	32	หมายเลขการจอง
6	CANCEL_NO	VarChar2	32	หมายเลขการยกเลิก
7	CHECK_IN	Date	-	วัน เวลาที่เข้าห้องพัก
8	CHECK_OUT	Date	-	วัน เวลาที่ออกจาก ห้องพัก
9	ROOM	Number	3,0	จำนวนห้องที่จอง
10	ADULT	Number	3,0	จำนวนผู้ใหญ่ที่เข้าพัก
11	CHILDREN	Number	3,0	จำนวนเด็กที่เข้าพัก
12	INFANT	Number	3,0	จำนวนเด็กที่อายุต่ำ กว่า 4 ปี
13	SPECIAL_REQUEST	VarChar2	512	สิ่งที่ต้องการเพิ่มเติม
14	TOTAL_PRICE	Number	16,2	ราคารวม
15	RESV_ROOMTYPE_NAME	VarChar2	32	ชื่อห้อง
16	RESV_ROOMTYPE_DESCRIPTION	VarChar2	512	คำอธิบายของห้องนั้น

17	RESV_BEDTYPE_NAME	VarChar2	32	ประเภทของเตียงของ ห้องนั้น
18	RESV_GUEST	Number	3,0	จำนวนคนพักสูงสุด ของห้อง
19	RESV_ROLLAWAY	Number	3,0	จำนวนเตียงของห้องที่ จอง
20	RESERVATION_STATUS_ID	Number	3,0	สถานะใบจอง
21	PRINT_STATUS_ID	Number	3,0	สถานะในการถูกพิมพ์
22	BOOK_DATE	Date	-	วันจอง
23	BOOK_TIME	Date	-	เวลาจอง

ตารางที่ 4.16 ตาราง GUEST_INFO

Table Name : GUEST_INFO				
Primary Key : GUEST_INFO_ID				
Foreign Key : POLICY_CREDIT_CARD_ID reference POLICY_CREDIT_CARD table				
ลำดับที่	Attribute	Type	Width	Description
1	GUEST_INFO_ID	Number	10,0	หมายเลขของข้อมูลผู้พัก
2	SALUTATION	VarChar2	16	คำนำหน้าชื่อ
3	FIRSTNAME	VarChar2	32	ชื่อ
4	LASTNAME	VarChar2	32	นามสกุล
5	BIRTH_DATE	Date	-	วันเกิด
6	COMPANY_NAME	VarChar2	32	ชื่อบริษัท
7	CONTACT_ADDRESS	VarChar2	128	ที่อยู่ติดต่อได้
8	CITY	VarChar2	32	เมือง
9	COUNTRY	VarChar2	32	ประเทศ
10	ZIPCODE	VarChar2	16	รหัสไปรษณีย์
11	PHONE	VarChar2	16	หมายเลขโทรศัพท์
12	EMAIL	VarChar2	32	จดหมายอิเล็กทรอนิกส์

13	FLIGHT	VarChar2	16	สายการบิน
14	ARRIVAL_DATE	Date	-	วันที่มาถึง
15	ARRIVAL_TIME	VarChar2	16	เวลาที่มาถึง
16	TIME_PERIOD	VarChar2	16	ช่วงของเวลา
17	POLICY_CREDIT_CARD_ID	Number	3,0	หมายเลขของนโยบายบัตรเครดิต
18	CARD_NO	VarChar2	32	รหัสบัตรเครดิต
29	CARD_EXPIRE	Date	-	วันหมดอายุของบัตรเครดิต

ตารางที่ 4.17 ตาราง AVERAGE_DAILY_RATE

Table Name : AVERAGE_DAILY_RATE				
Primary Key : AVERAGE_DAILY_RATE_ID				
Foreign Key : RESERVATION_ID reference RESERVATION table				
ลำดับที่	Attribute	Type	Width	Description
1	AVERAGE_DAILY_RATE_ID	Number	10,0	หมายเลขของข้อมูลราคาเฉลี่ย
2	RESERVATION_ID	Number	10,0	หมายเลขของการทำ การจอง
3	ROOMTYPE_ID	Number	10,0	หมายเลขของประเภทห้องพัก
4	FROM_DATE	Date	-	วันเริ่มต้น
5	TO_DATE	Date	-	วันสิ้นสุด
6	PRICE	Number	9,2	ราคาเฉลี่ย

ตารางที่ 4.18 ตาราง RESERVE_RATE

Table Name : RESERVE_RATE				
Primary Key : RESERVE_RATE_ID				
Foreign Key : RESERVATION_ID reference RESERVATION table				
ลำดับที่	Attribute	Type	Width	Description
1	RESERVE_RATE_ID	Number	10,0	หมายเลขของอัตราราคา
2	RESERVATION_ID	Number	10,0	หมายเลขของการทำ การจอง
3	RESERVE_DATE	Date	-	วันเวลาการจอง
4	PRICE_TYPE_ID	Number	3,0	วันเริ่มต้น
5	STATUS_TYPE_ID	Number	3,0	วันสิ้นสุด
6	PRICE	Number	9,2	ราคาเฉลี่ย

ตารางที่ 4.19 ตาราง RESERVE_SEASON

Table Name : RESERVE_SEASON				
Primary Key : RESERVE_SEASON_ID				
Foreign Key : RESERVATION_ID reference RESERVATION table				
ลำดับที่	Attribute	Type	Width	Description
1	RESERVE_SEASON_ID	Number	10,0	หมายเลขของฤดูกาลที่จอง
2	RESERVATION_ID	Number	10,0	หมายเลขของการทำ การจอง
3	SEASON_CODE	VarChar2	16	วันเวลาการจอง
4	FROM_DATE	Date	-	วันเริ่มต้น
5	TO_DATE	Date	-	วันสิ้นสุด
6	ROOM_RATE_PRICE	Number	9,2	ราคาห้อง
7	ADULT_ROLLAWAY_PRICE	Number	9,2	ราคาเตียงเสริมของผู้ใหญ่
8	CHILD_ROLLAWAY_PRICE	Number	9,2	ราคาเตียงเสริมของเด็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

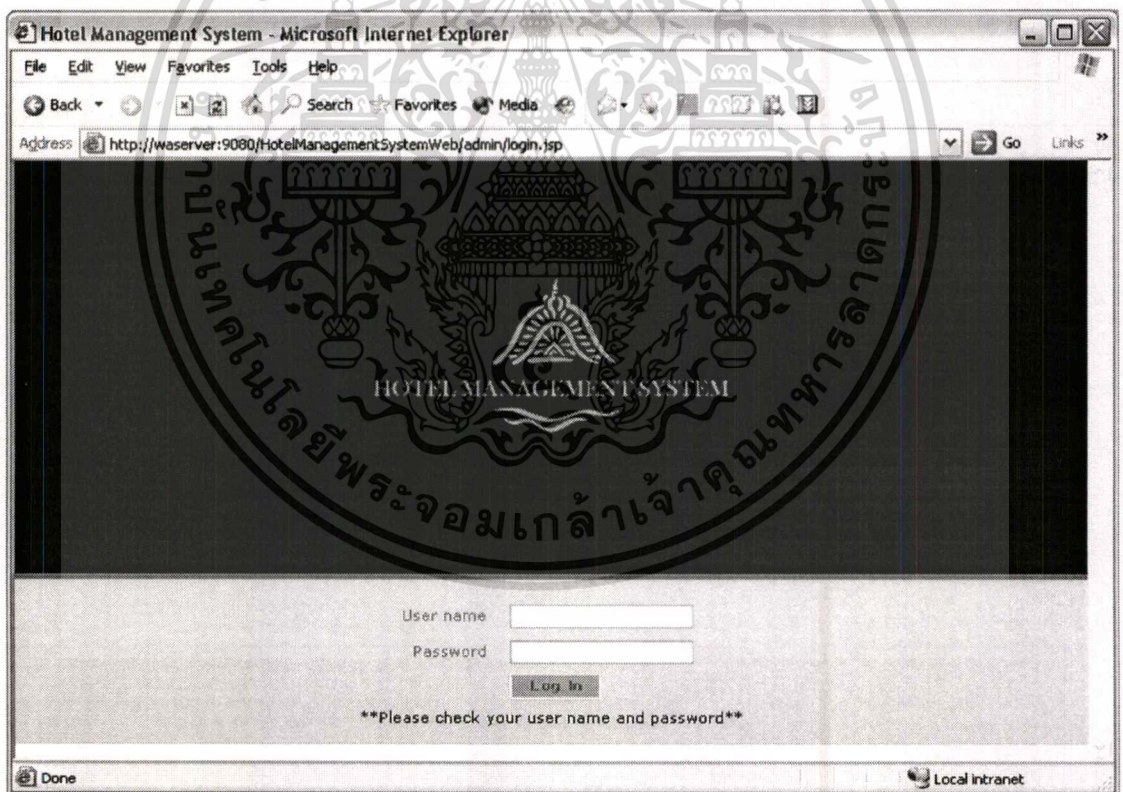
บทที่ 5

การพัฒนาระบบงาน

5.1 การออกแบบหน้าจอ

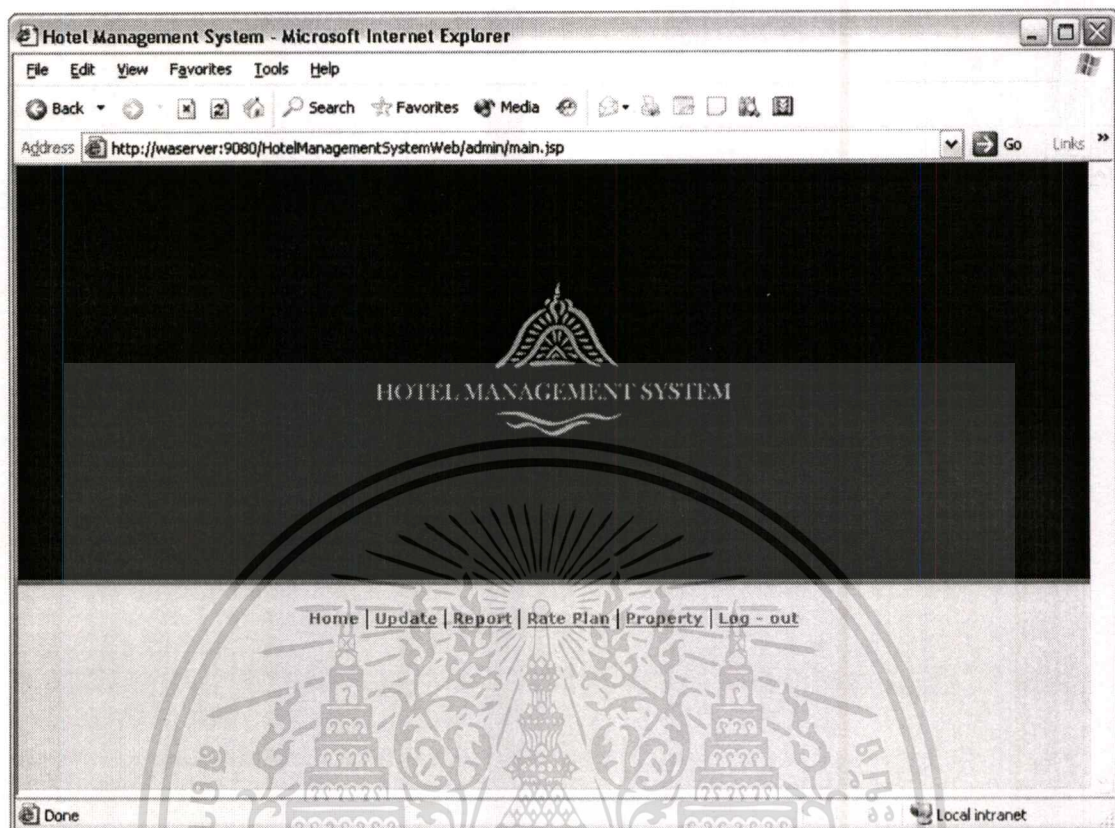
การออกแบบหน้าจอ นั้นจะเป็นการออกแบบเว็บเพจ โดยจะแบ่งเป็น 2 ส่วนคือ ส่วนของ Administrator และ ส่วนของ Customer แต่ละส่วนมีรายละเอียดดังนี้

Administrator



รูปที่ 5.1 หน้าจอ Login ของผู้ดูแลระบบ

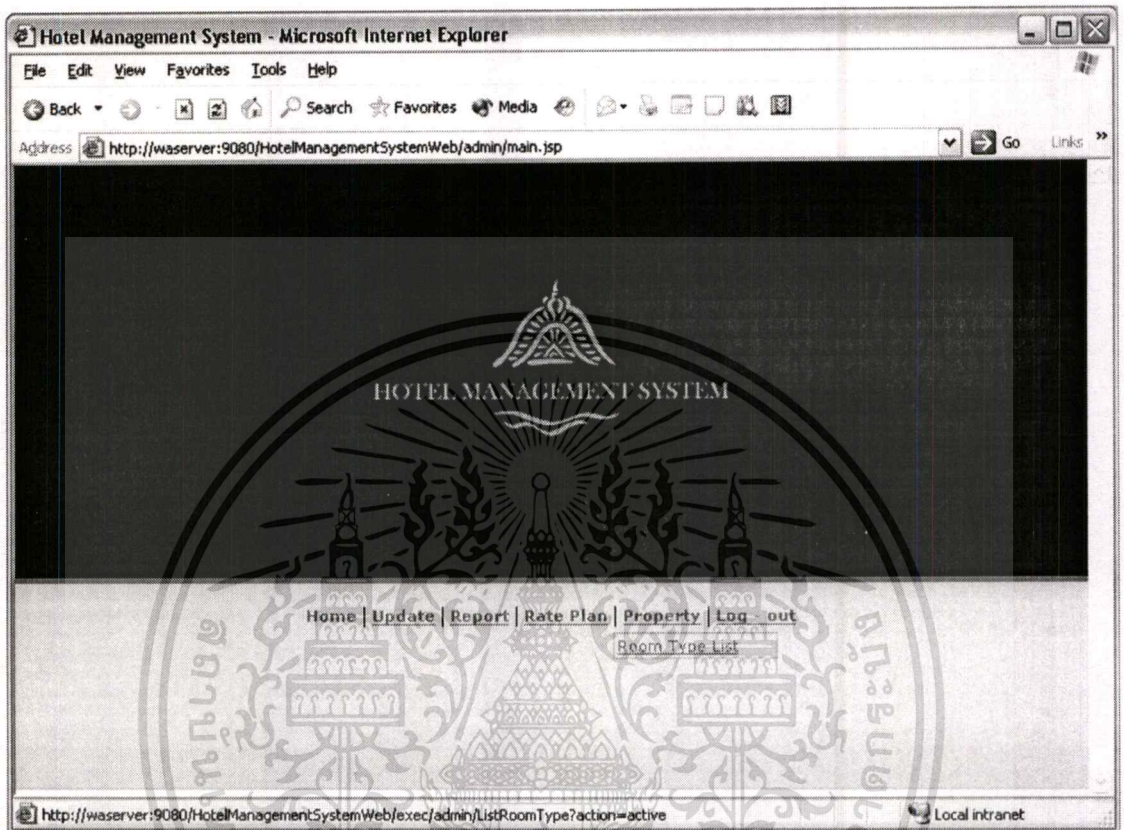
จากรูปที่ 5.1 นั้นเป็นหน้าจอแรกเมื่อเข้ามาสู่ส่วนของผู้ดูแลระบบ ซึ่งผู้ดูแลระบบจะต้องใส่ User name และ Password ให้ถูกต้องเพื่อเข้าไปจัดการระบบต่อไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 หน้าจอ Menu เมื่อเข้าสู่ระบบ

จากรูปที่ 5.2 เมื่อผู้ดูแลระบบ Login เข้ามาในระบบอย่างถูกต้องแล้ว ระบบจะแสดงหน้า Menu เพื่อให้ผู้ดูแลระบบเลือกที่จะทำงานต่อไป

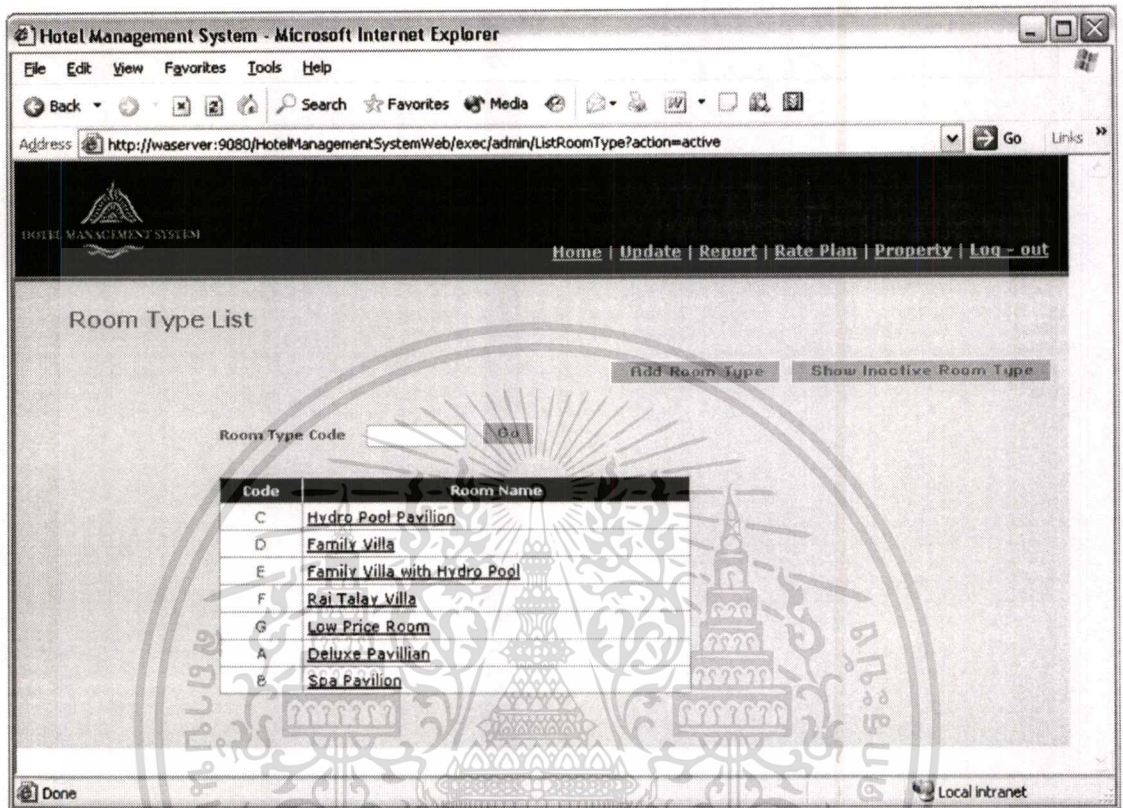
Administrator : Property



รูปที่ 5.3 หน้าจอ Menu เพื่อเข้าไปจัดการข้อมูลในส่วนของ Property

จากรูปที่ 5.3 แสดงให้เห็น Menu ที่จะเข้าไปจัดการข้อมูลในส่วนของ Property

Administrator : Property : Room Type List

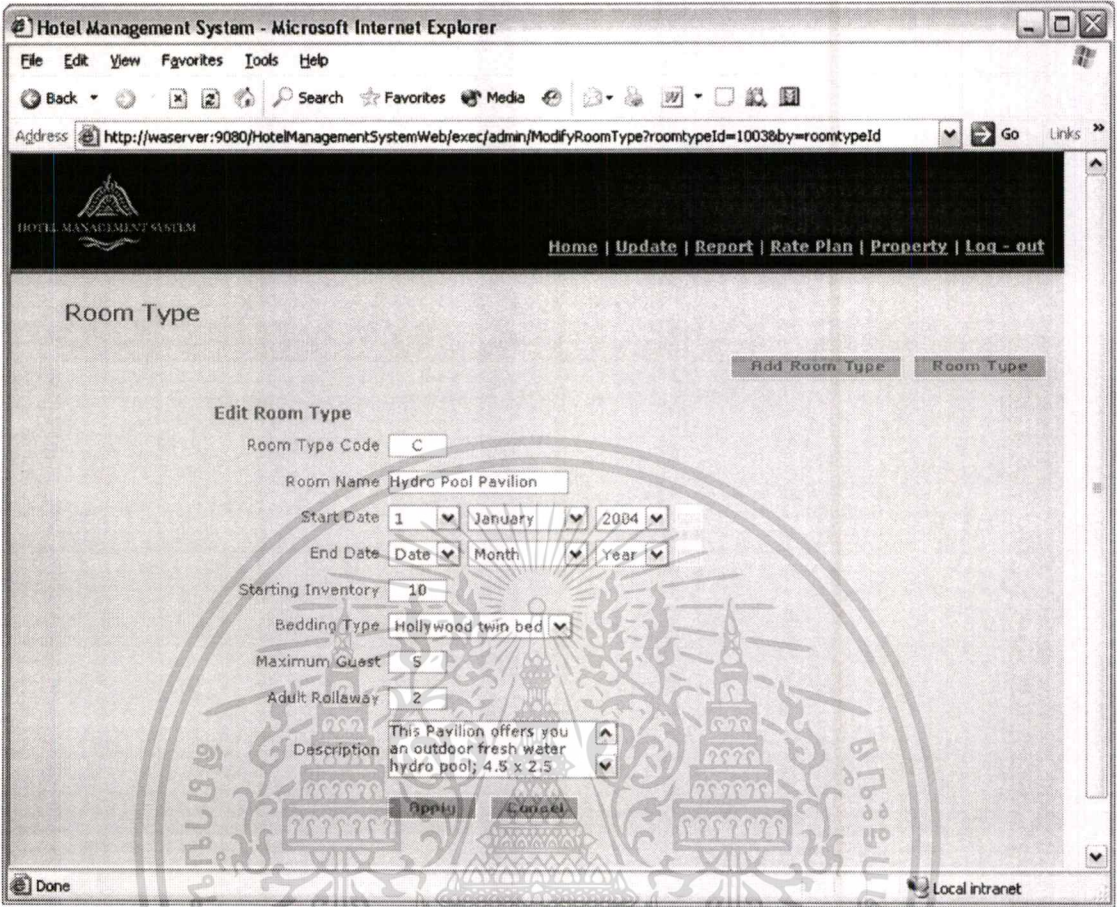


รูปที่ 5.4 แสดงหน้าจอแรกในส่วนของการจัดการข้อมูลของ Room Type

จากรูปที่ 5.4 เป็นหน้าจอที่แสดง Room Type ทั้งหมดที่มีอยู่ในระบบ ซึ่งในหน้านี้จะแสดงชื่อของห้อง และรหัสของห้องนั้นให้กับผู้ดูแลระบบ

รูปที่ 5.5 แสดงหน้าจอเมื่อต้องการเพิ่ม Room Type

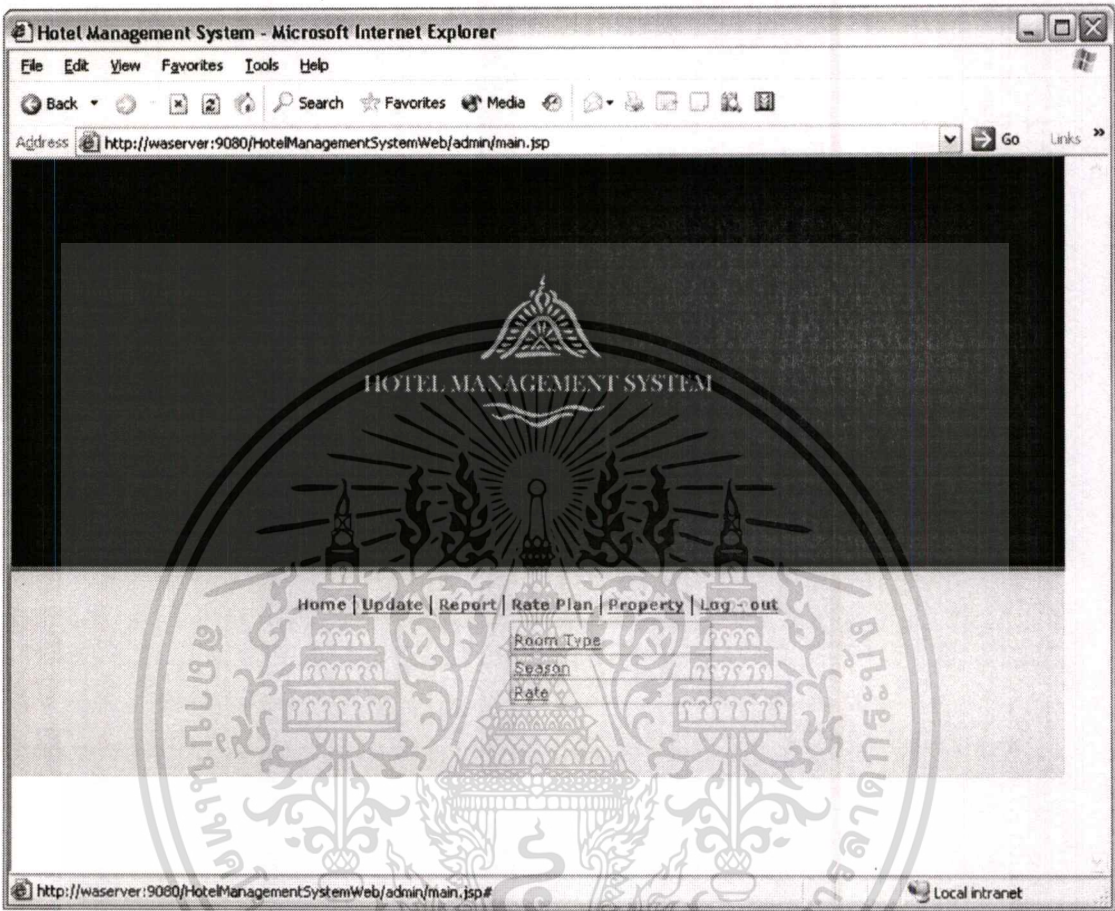
การรูปที่ 5.5 เมื่อผู้ดูแลระบบต้องการเพิ่ม Room Type ระบบจะแสดงหน้าจอขึ้นมา ซึ่งข้อมูลที่จำเป็นต้องระบุในแต่และ Room Type นั้นประกอบด้วย รหัสห้องพัก ชื่อห้องพัก วันเริ่มต้น วันสิ้นสุด จำนวนห้องทั้งหมดที่มี ประเภทของเตียงนอน จำนวนคนสูงสุดที่พักได้ จำนวนเตียงเสริมที่มีได้ และรายเอียดของห้อง



รูปที่ 5.6 แสดงหน้าจอเมื่อต้องการแก้ไขข้อมูลของ Room Type

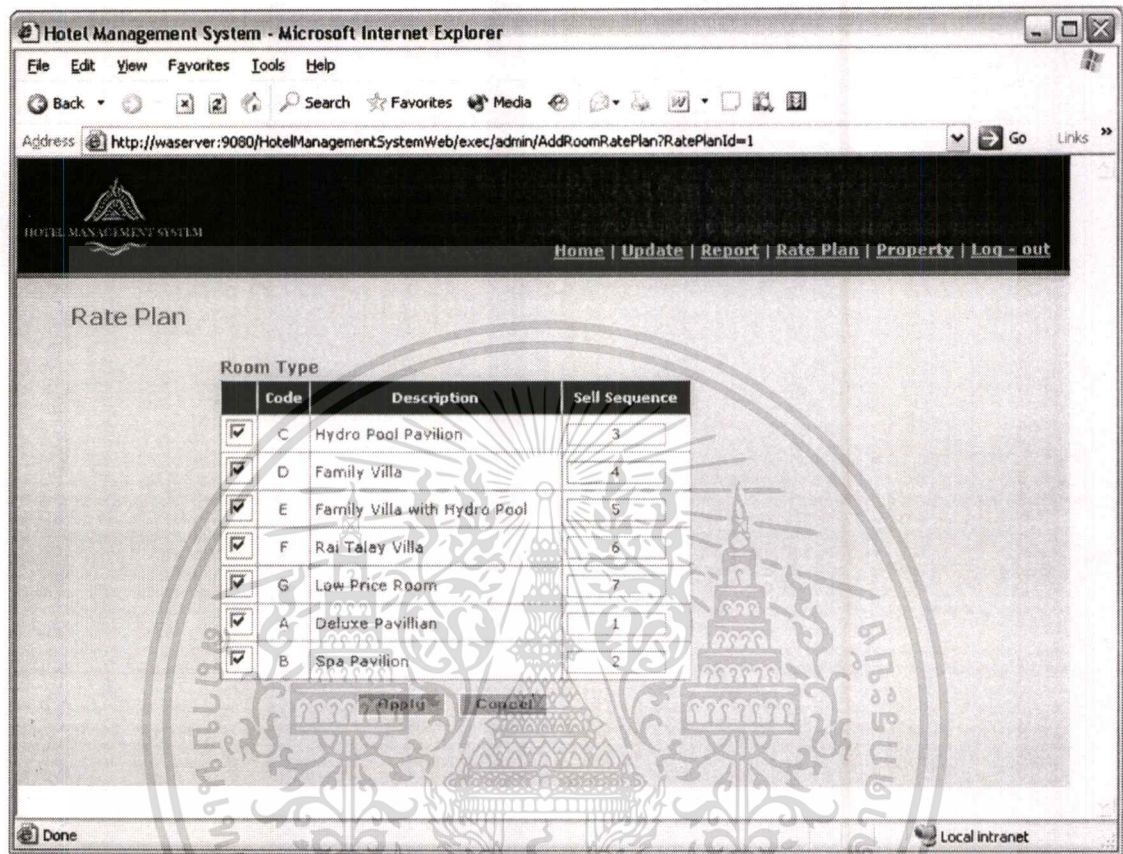
จากรูปที่ 5.6 จะเป็นหน้าจอเมื่อผู้ดูแลระบบต้องการเข้ามาแก้ไขข้อมูลของ Room Type ที่มีอยู่แล้ว

Administrator : Rate Plan



รูปที่ 5.7 หน้าจอ Menu เพื่อเข้าไปจัดการข้อมูลในส่วนของ Rate Plan

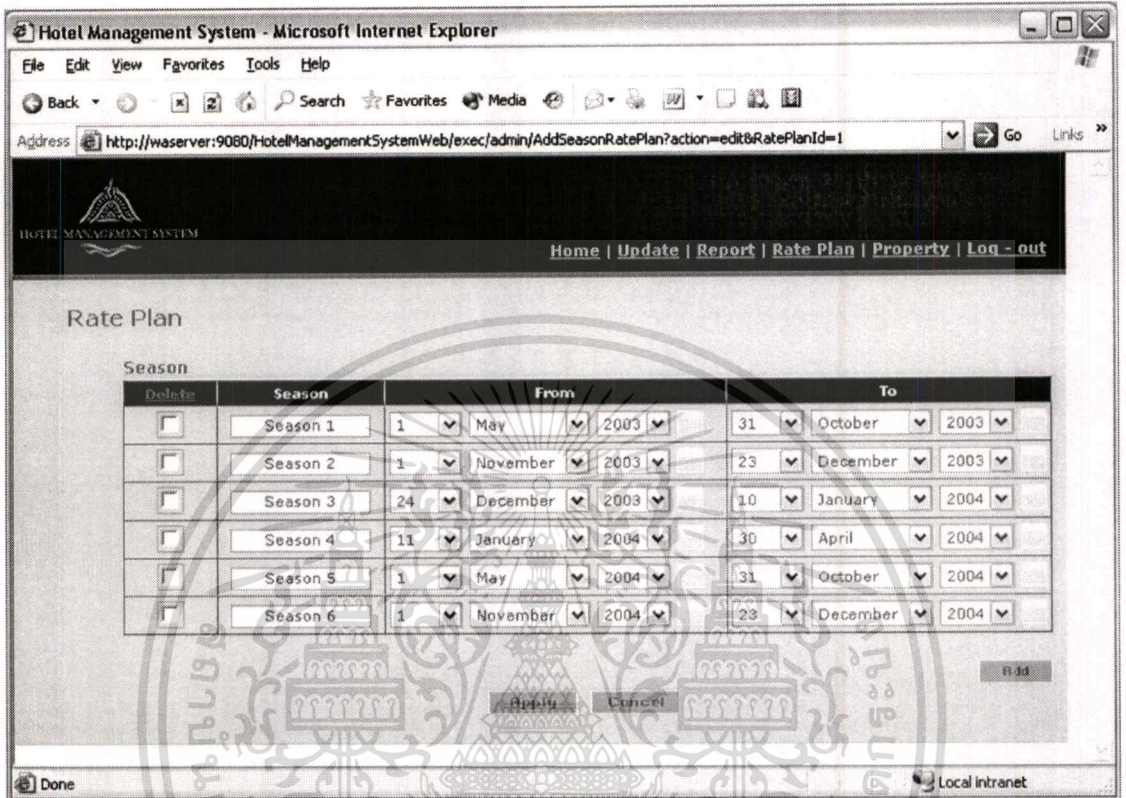
Administrator : Rate Plan : Room Type



รูปที่ 5.8 แสดงหน้าจอเพื่อจัดการ Sell Sequence ของ Room Type

จากรูปที่ 5.8 จะเป็นหน้าจอที่ผู้ดูแลระบบจะทำการเลือก Room Type ที่จะนำออกไปให้ Customer เลือกได้ โดยดูจากถ้า Room Type ใดถูก check อยู่จะต้องถูกระบุลำดับการขายด้วย Room Type นั้นก็จะสามารถให้ Customer เลือกที่จะพักได้

Administrator : Rate Plan : Season



Delete	Season	From	To
☐	Season 1	1 May 2003	31 October 2003
☐	Season 2	1 November 2003	23 December 2003
☐	Season 3	24 December 2003	10 January 2004
☐	Season 4	11 January 2004	30 April 2004
☐	Season 5	1 May 2004	31 October 2004
☐	Season 6	1 November 2004	23 December 2004

Buttons: Add, Apply, Cancel

รูปที่ 5.9 แสดงหน้าจอเพื่อจัดการ Season ของระบบ

จากรูปที่ 5.9 แสดงหน้าจอเพื่อให้ผู้ดูแลระบบได้เข้ามาจัดการข้อมูลของ Season ต่าง ๆ ที่ทางโรงแรมเปิดให้บริการ ซึ่งรอยต่อของแต่ละ Season จะต้องต่อเนื่องกัน

Administrator : Rate Plan : Rate

Hotel Management System - Microsoft Internet Explorer

Address: <http://waserver:9080/HotelManagementSystemWeb/exec/admin/DisplayRatePackage?RatePlanId=1>

Home | Update | Report | Rate Plan | Property | Log - out

Rate Plan

Rate

Season: 01 May 03 - 31 Oct 03

Code	Room Name	Rate Per Room	Extra Person	Child
C	Hydro Pool Pavilion	23,000	7,800	800
D	Family Villa	28,000	7,800	800
E	Family Villa with Hydro Pool	32,000	7,800	800
F	Rai Talay Villa	34,000	7,800	800
G	Low Price Room	3,200	CP	CP
A	Deluxe Pavillion	16,000	7,800	800
B	Spa Pavilion	18,000	7,800	800

Apply Cancel

Done Local intranet

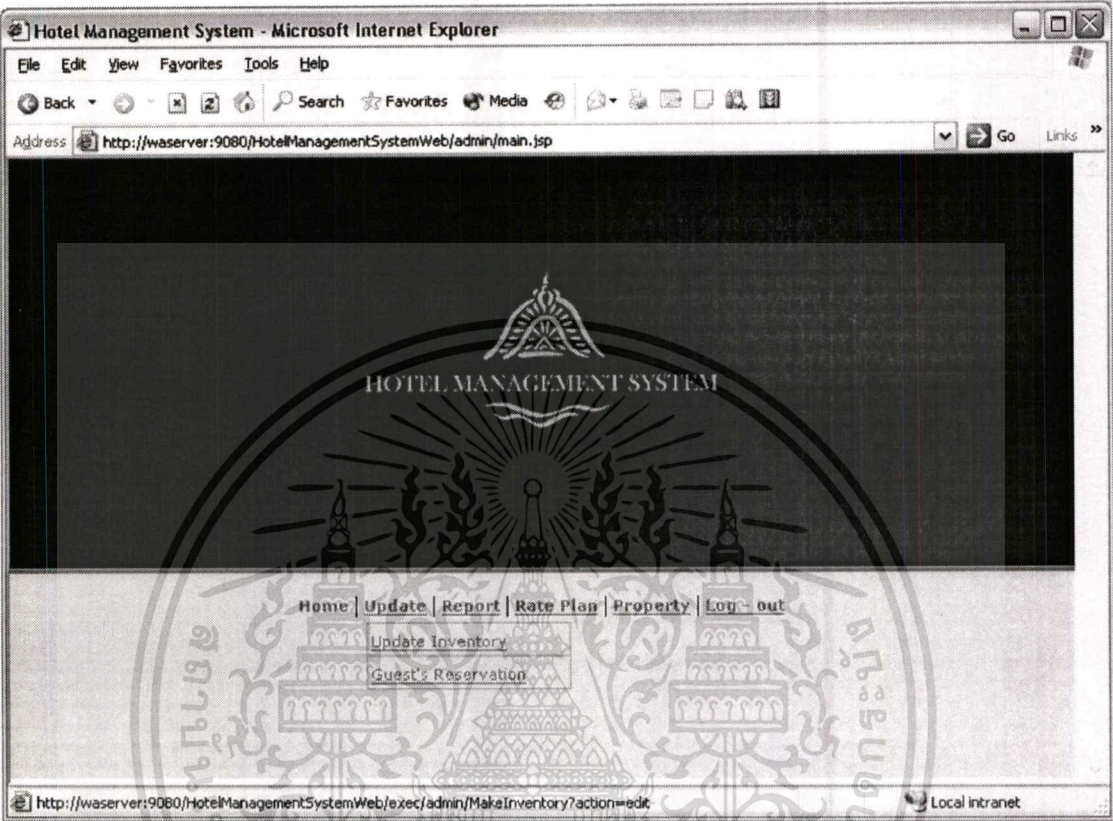
รูปที่ 5.10 แสดงหน้าจอเพื่อจัดการ Rate ของแต่ละห้องในแต่ละ Season

จากรูปที่ 5.10 เป็นหน้าจอที่ผู้ดูแลระบบเข้ามาเพื่อระบบราคาในแกห้องแต่ละห้อง ในแต่ละ Season ซึ่งราคาจะประกอบไปด้วยราคาของห้อง ราคาเตียงเสริมที่เป็นผู้ใหญ่ และราคาเตียงเสริมที่เป็นเด็ก ซึ่งราคาเตียงเสริมของทั้ง 2 แบบนั้นจะมี option ดังนี้

1. - (Regular Price) คือ option ที่ต้องคิดราคาตามที่กำหนด
2. CP (Complementary) คือ option ที่ไม่คิดราคา
3. NA (Not Applicable) คือ option ที่ไม่อนุญาตให้มีเตียงเสริมได้

ในการระบุราคนั้นต้องระบุราคาให้ครบทุกห้อง และทุก Season เพื่อให้ระบบทำงานได้อย่างสมบูรณ์

Administrator : Update



รูปที่ 5.11 หน้าจอ Menu เพื่อเข้าไป Update ข้อมูลเกี่ยวกับการจอง

Administrator : Update : Update Inventory

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print

Address <http://waserver:9080/HotelManagementSystemWeb/exec/admin/MakeInventory?action=edit> Go Links

HOTEL MANAGEMENT SYSTEM

[Home](#) | [Update](#) | [Report](#) | [Rate Plan](#) | [Property](#) | [Log - out](#)

Update Inventory

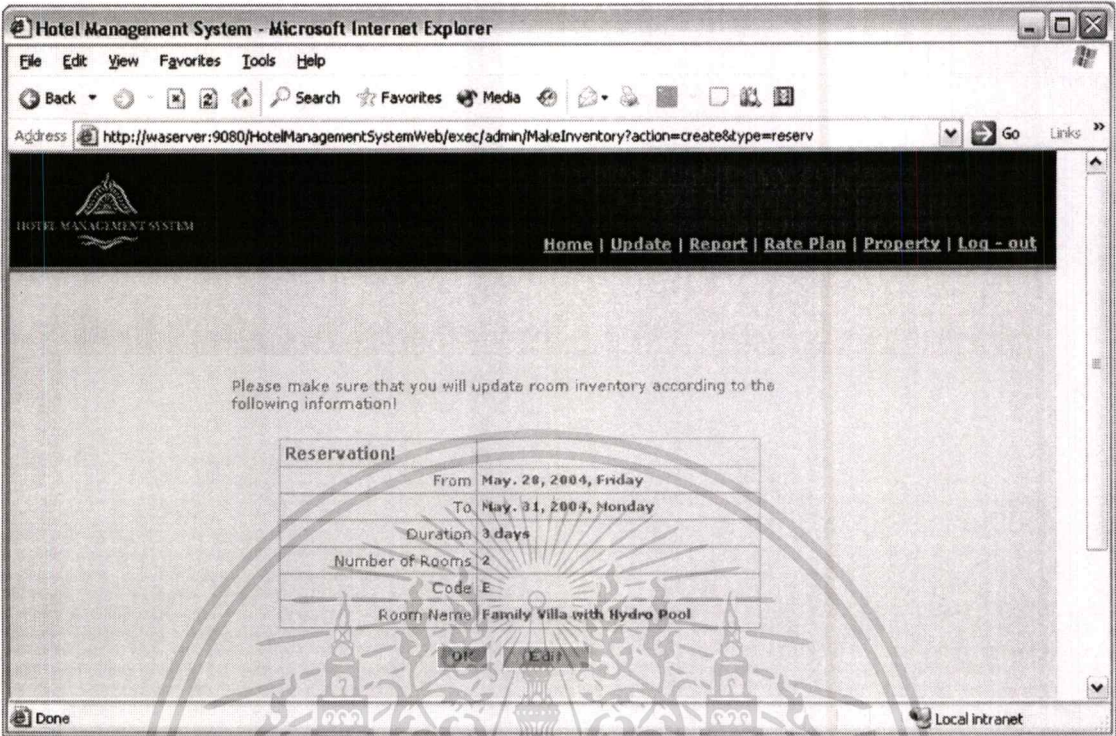
Arrival	Date	Month	Year
Departure	Date	Month	Year
Room Type Code	C		
Number of Rooms			

[Reservation](#) [Cancellation](#)

Done Local intranet

รูปที่ 5.12 แสดงหน้าจอ Update Inventory จากระบบอื่น

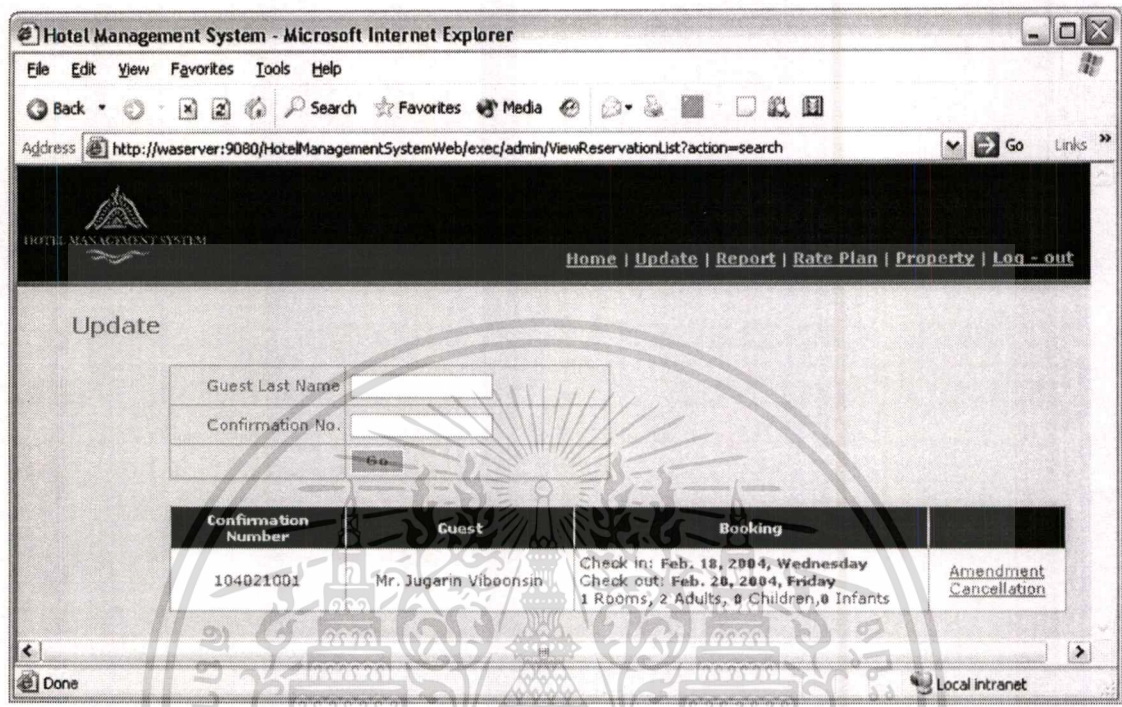
จากรูปที่ 5.12 เป็นหน้าจอที่ผู้ดูแลระบบเข้ามาเพื่อ Update Inventory ซึ่งเป็นข้อมูลจากระบบอื่น โดยการระบุขอบเขตของวันเข้าพัก รหัสของห้องพัก และจำนวนห้อง จากนั้นจึงเลือกว่าการ Update นี้เป็นการจองห้อง หรือยกเลิกห้อง



รูปที่ 5.13 แสดงหน้ายืนยันการ Update Inventory จากระบบอื่น

รูปที่ 5.13 แสดงหน้าจอหลังจากที่ได้รับข้อมูลมาจากหน้าก่อนหน้านี้ เพื่อยืนยันการ Update

Administrator : Update : Guest's Reservation



รูปที่ 5.14 แสดงหน้าในการค้นหาการจองของ Customer เพื่อทำการแก้ไขข้อมูล

จากรูปที่ 5.14 เป็นหน้าจอที่ให้ผู้ดูแลระบบเข้ามาค้นหาการจองห้องพักของ Customer เพื่อทำการ Update ข้อมูลต่าง ๆ หรือยกเลิกการจองนั้น โดยการระบุนามสกุล และหมายเลขการยืนยันจอง

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/admin/ViewAmendment?index=0 Go Links

HOTEL MANAGEMENT SYSTEM

Home | Update | Report | Rate Plan | Property | Log - out

Amendment

Check in*	Feb. 18, 2004
Check out*	Feb. 20, 2004
Number of Rooms*	1
Adults per Room*	2
Child per Room (4 - 12 yrs)	0
Infants per Room (under 4 yrs)	0

Room and Rate :

Spa Pavilion
 Total Price : THB 36,000
 Average Daily Rate : THB 18,000
 Feb. 18, 2004 - Feb. 20, 2004; THB 18,000

Guest Information :

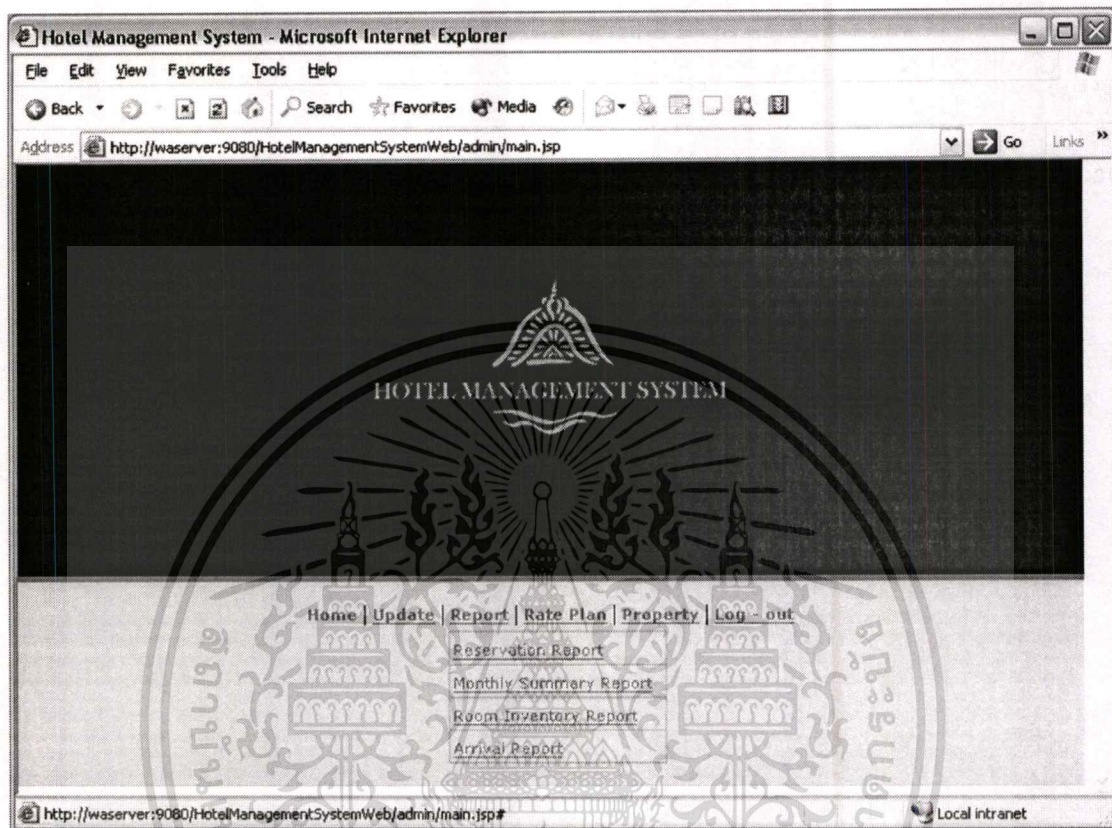
Title*	Mr. ▼
First Name*	Jugarin
Last Name*	Viboonsin
Company	Test Company

Local intranet

รูปที่ 5.15 แสดงหน้าในการแก้ไขข้อมูลของ Customer

จากรูปที่ 5.15 แสดงให้เห็นหน้าจอที่ผู้ดูแลระบบสามารถเข้ามาแก้ไขข้อมูลต่าง ๆ ของ Customer ซึ่งสามารถแก้ไขได้เฉพาะข้อมูลที่เป็นรายละเอียดของ Customer เท่านั้น ไม่สามารถแก้ไขข้อมูลการจองห้องพักได้

Administrator : Report



รูปที่ 5.16 หน้าจอ Menu เพื่อแสดง Report ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Administrator : Report : Reservation Report

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/admin/SearchReservationReport?action=search Go Links

HOTEL MANAGEMENT SYSTEM

Home | Update | Report | Rate Plan | Property | Log - out

Reservation Report

Status All Unprinted

Confirmation No.

Guest Last Name

From 1 February 2004

To 29 February 2004

Go

Display 1 - 7 total 7 Page 1

Del	Book Date	Time	Confirmation/ Cancellation No.	Type	Guest	Arrival	Departing
☐	07 Feb 04	13:14:55	104021000	Cancel	Jugarin Viboonsin	11 Feb 04	13 Feb 04
☐	08 Feb 04	2:6:10	104021001	New	Jugarin Viboonsin	18 Feb 04	20 Feb 04
☐	08 Feb 04	23:22:30	104021002	New	Jugarin Viboonsin	26 Feb 04	29 Feb 04
☐	08 Feb 04	23:46:11	104021003	New	Jugarin Viboonsin	29 Feb 04	10 Mar 04
☐	08 Feb 04	23:55:15	104021004	New	Jugarin Viboonsin	16 Feb 04	18 Feb 04
☐	09 Feb 04	0:4:13	104021005	New	Tammy Youw	15 Apr 04	17 Apr 04
☐	09 Feb 04	0:12:52	104021006	New	Tammy Youw	23 Mar 04	25 Mar 04

Display 1 - 7 total 7 Page 1

Local intranet

รูปที่ 5.17 แสดงหน้าจอของ Reservation Report

จากรูปที่ 5.17 แสดงหน้าจอการออก Report เกี่ยวกับการจอง โดยที่ผู้ดูแลระบบจะระบุข้อมูลที่ใช้ในการค้นหา จากนั้นระบบจะแสดงรายการการจองห้องพักที่ตรงกับเงื่อนไขที่ผู้ดูแลระบบได้ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print

Address <http://waserver:9080/HotelManagementSystemWeb/exec/admin/ViewReservationReport?resId=1004> Go Links

HOTEL MANAGEMENT SYSTEM

[Home](#) | [Update](#) | [Report](#) | [Rate Plan](#) | [Property](#) | [Log - out](#)

Reservation Report

Stay Information			
Confirmation No.	: 104021004	Cancellation No.	:
Arrives	: Feb. 16 2004, Monday	Departs	: Feb. 18 2004, Wednesday
LOS	: 2	Rooms	: 1
Rate Plan(s)	:	Adults	: 1
Children	: 0	Infants	: 0

Room Information				
Room	Bedding Type	Room Description	Average Daily Rate Per Room	Total For Room Rate Only
Deluxe Pavillion	King size bed	Located between two sublime beaches, Nam Mao and Rallay is the 77 rooms pavilion fully equipped and designed to provide spacious accommodation with the aspect selection of garden view, pool side, or cliff and partial sea view provided.	Feb. 16, 2004 - Feb. 18, 2004 16,000	32,000

Guest Information	
Contact Guest	Mr. Jugarin Viboonsin
Company	Test Company
Email	j_viboonsin@yahoo.com
Telephone	023226114
Credit Card	MASTER CARD
Card Number	xxxxxxxxxx7208
Expired	July 2015
Flight No.	
Arrival Date	
Time	
Birth Date	
Contact Address	
City	
Country	

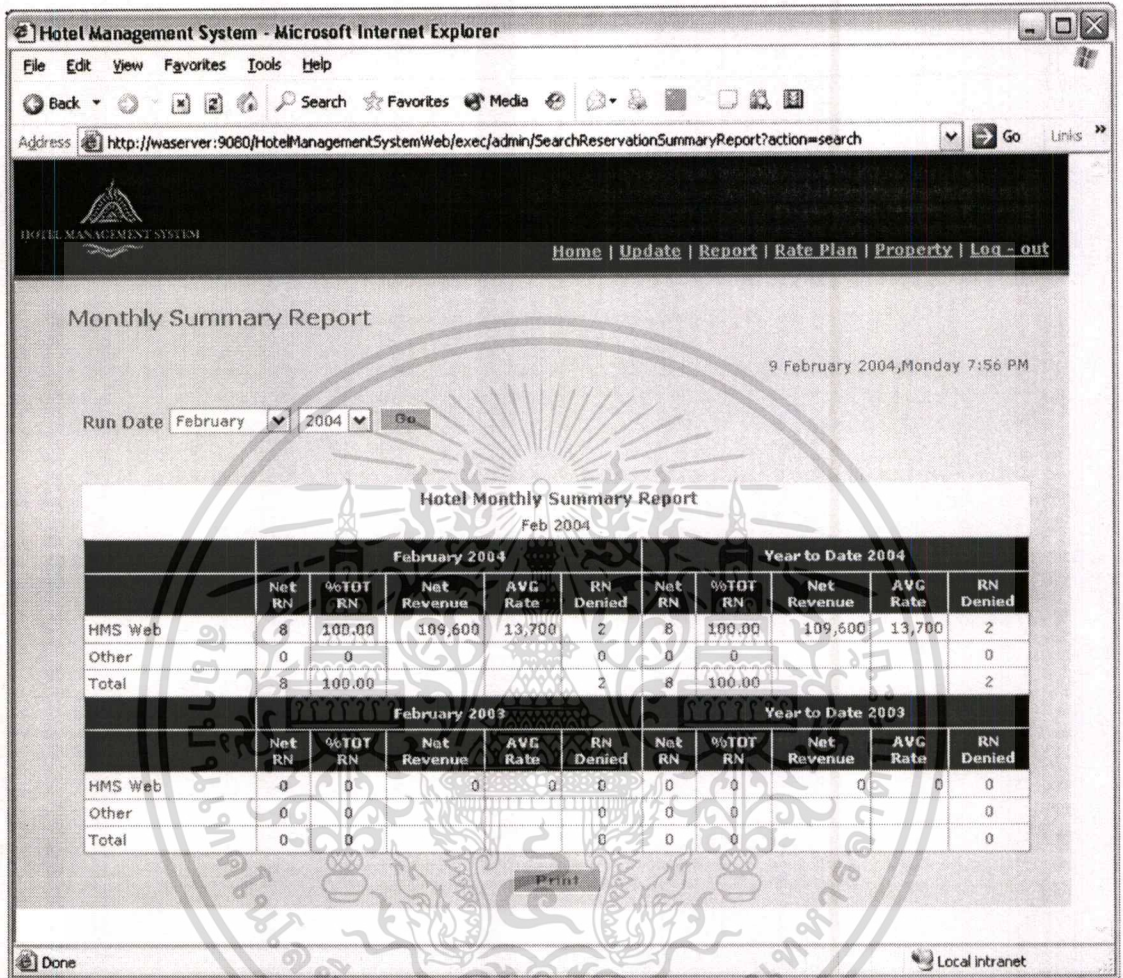
Local intranet

รูปที่ 5.18 แสดงหน้าจอของ Reservation Report Detail

จากรูปที่ 5.18 เกิดจากผู้ดูแลระบบต้องการที่จะดูรายละเอียดเกี่ยวกับการจอง ซึ่งมาจาก รายการการจองห้องพักจากหน้าที่ผ่านมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Administrator : Report : Monthly Summary Report



รูปที่ 5.19 แสดงหน้าจอของ Monthly Summary Report

จากรูปที่ 5.19 แสดงหน้าจอเกี่ยวกับการแสดง Report รายรับเป็นรายเดือนซึ่งแบ่งเป็น 4 ส่วนคือแสดงรายรับของเดือนที่ระบุ แสดงรายรับตั้งแต่ต้นปีจนถึงเดือนที่ระบุ แสดงรายรับของเดือนที่ระบุแต่เป็นของปีที่ผ่านมา และแสดงรายรับตั้งแต่ต้นปีจนถึงเดือนที่ระบุแต่เป็นของปีที่ผ่านมา

Administrator : Report : Room Inventory Report

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/admin/SearchPropertyStatusReport?action=search&sortBy=date& Page 1

Home | Update | Report | Rate Plan | Property | Log - out

Room Inventory Report

9 February 2004,Monday 8:14 PM

Room Name Deluxe Pavillian

From 10 February 2004

To 19 February 2004

Go

Display 1 - 10 total 10

Hotel Inventory Report

Deluxe Pavillian

10 Feb 2004 - 19 Feb 2004

Date	Code	Room Name	Starting Inventory	Remaining Inventory
Feb. 10, 2004, Tuesday	A	Deluxe Pavillian	10	10
Feb. 11, 2004, Wednesday	A	Deluxe Pavillian	10	10
Feb. 12, 2004, Thursday	A	Deluxe Pavillian	10	10
Feb. 13, 2004, Friday	A	Deluxe Pavillian	10	10
Feb. 14, 2004, Saturday	A	Deluxe Pavillian	10	10
Feb. 15, 2004, Sunday	A	Deluxe Pavillian	10	10
Feb. 16, 2004, Monday	A	Deluxe Pavillian	10	9
Feb. 17, 2004, Tuesday	A	Deluxe Pavillian	10	9
Feb. 18, 2004, Wednesday	A	Deluxe Pavillian	10	10
Feb. 19, 2004, Thursday	A	Deluxe Pavillian	10	10

Display 1 - 10 total 10

Print

Page 1

Done Local intranet

รูปที่ 5.20 แสดงหน้าจอของ Room Inventory Report

จากรูปที่ 5.20 เป็นการแสดง Report เกี่ยวกับจำนวนห้องที่เหลือในแต่ละวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Administrator : Report : Arrival Report

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media

Address <http://waserver:9080/HotelManagementSystemWeb/exec/admin/SearchArrivalReport?action=search> Go Links

HOTEL MANAGEMENT SYSTEM

[Home](#) | [Update](#) | [Report](#) | [Rate Plan](#) | [Property](#) | [Log - out](#)

Arrival Report

9 February 2004, Monday 8:26 PM

Arrival 18 February 2004 Go

Hotel Arrival Report

18 Feb 2004

Guest Name	Room	LOS	Rate	Flight	Guaranteed
Viboonsin, Jugarin	B/Spa Pavilion	2	18,000	TG 891/18 Feb 04 9:00 PM	MC/5577721906867208 Nov 04
Total Rooms Arriving					1
Average LOS					2
Total Estimated Revenue					36,000
Estimated ADR					18,000

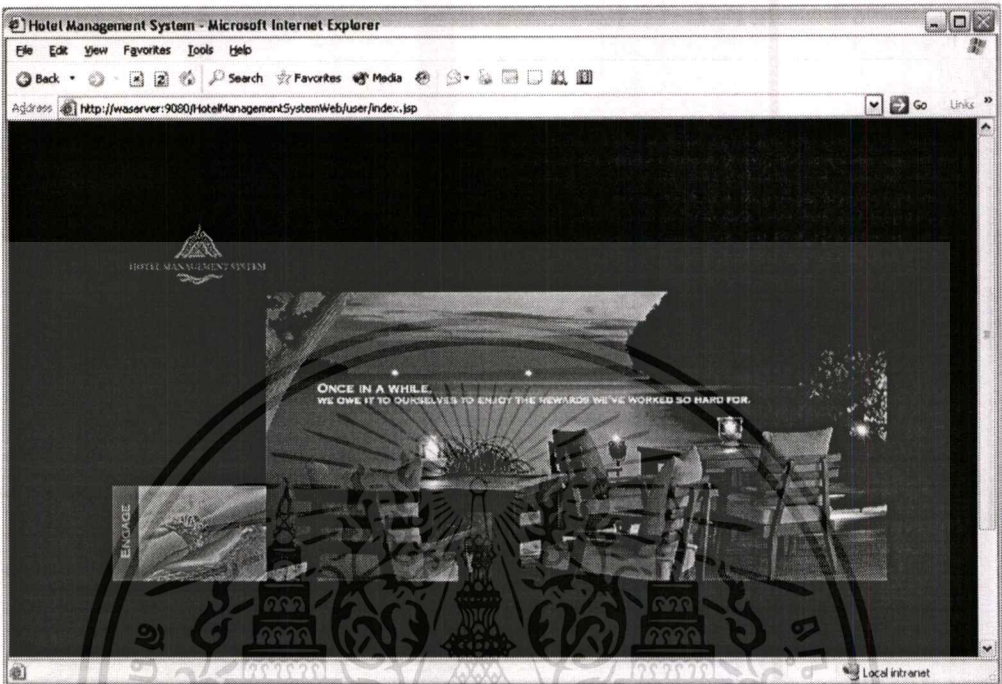
Print

Done Local intranet

รูปที่ 5.21 แสดงหน้าจอของ Arrival Report

จากรูปที่ 5.21 เป็นการแสดง Report เกี่ยวกับการมาถึงในวันที่ระบุ

Customer



รูปที่ 5.22 แสดงหน้าจอแรกในส่วนของ Customer



รูปที่ 5.23 แสดงหน้าจอ Menu ของ Customer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Customer : View Rate

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/user/ViewRatePackage Go Links

HOTEL MANAGEMENT SYSTEM

Engage

Home | Rate | Reservation

Hotel Management System Rate effective January 2004

Room Type	01 May 03	01 Nov 03	24 Dec 03	11 Jan 04	01 May 04	01 Nov 04
	31 Oct 03	23 Dec 03	10 Jan 04	30 Apr 04	31 Oct 04	23 Dec 04
Hydro Pool Pavilion	20,000	20,000	24,000	24,000	24,000	24,000
Family Villa	28,000	28,000	28,000	28,000	28,000	28,000
Family Villa with Hydro Pool	32,000	32,000	32,000	32,000	32,000	32,000
Rai Talar Villa	34,000	38,000	36,000	38,000	38,000	38,000
Low Price Room	3,200	3,200	3,200	3,200	3,200	3,200
Deluxe Pavilion	16,000	16,000	16,000	16,000	16,000	16,000
Spa Pavilion	18,000	17,600	17,000	18,000	18,000	18,000

All rates are quoted in Thai Baht and subject to 10% service charge and 7% tax.
Rates are inclusive of breakfast.
Rates are subject to change without notice.

Local intranet

รูปที่ 5.24 แสดงหน้าจอ View Rate ของ Customer

Customer : Reservation

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/user/EditStayReservation Go Links

HOTEL MANAGEMENT SYSTEM Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Stay Info 2 Room Availability 3 Guest Info 4 Confirmation

Please enter the following information to find your preferences.

Check in:* 18 March 2004

Check out:* 20 March 2004

Number of Rooms:* 2

Adults per Room:* 2

Child per Room (4 - 12 yrs): 2

Infants per Room (under 4 yrs): Select

Next

Local intranet

รูปที่ 5.25 แสดงหน้าจอ Stay Information ของ Customer

จากรูปที่ 5.25 แสดงหน้าจอแรกเมื่อ Customer เข้ามาเพื่อที่จะจองห้องพัก โดยที่ Customer ต้องระบุขอบเขตของการเข้าพัก จำนวนห้องที่ต้องการ จำนวนผู้ใหญ่ จำนวนเด็ก และจำนวนเด็กที่อายุต่ำกว่า 4 ปี

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://waserver:9080/HotelManagementSystemWeb/exec/user/SaveStayReservation> Go Links

Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Stay Info 2 Room Availability 3 Guest Info 4 Confirmation

5 of 6 page 1 | 2 Stay Information:

Room	Bed Type	Room Description	Rate/THB
Deluxe Pavilion	King size bed	Located between two sublime beaches, Nana Mao and Railay is the 77 rooms pavilion fully equipped and designed to provide spacious accommodation with the aspect selection of garden view, pool-side, or cliff and partial sea view provided.	16,000 Next>>
Spa Pavilion	King size bed	Located in Railay and Phra Nang Zones, the Spa Pavilion gives you more privacy and recreation by providing an additional private garden equipped with an outdoor Jacuzzi and BBQ terrace.	18,000 Next>>
Hydro Pool Pavilion	Hollywood twin bed	This Pavilion offers you an outdoor fresh water hydro pool 1.5 x 2.5 meters in size where you can indulge yourself privately in a landscaped enclosed	24,000 Next>>

Check in: 18 March 2004

Check out: 20 March 2004

Number of Rooms: 2

Adults per Room: 2

Child per Room (4 - 12 yrs): 2

Infants per Room (under 4 yrs): Select

[Change>>](#)

Local intranet

รูปที่ 5.26 แสดงหน้าจอ Room Availability ของ Customer

จากรูปที่ 5.26 แสดงหน้าจอหลังจากที่ Customer ได้ระบุ Stay Information และระบบได้นำข้อมูลนั้นไปทำการเลือกห้องที่เหมาะสมออกมาให้กับ Customer

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Links

Address http://waserver:9080/HotelManagementSystemWeb/exec/user/SaveRoomAvailReservation?index=1002&rateplanId=1 Go

Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Stay Info 2 Room Availability 3 Guest Info 4 Confirmation

Please enter the following information for your best accommodation.

Stay Information:

Check in: 18 March 2004

Check out: 20 March 2004

Number of Rooms: 2

Adults per Room: 2

Child per Room (4 - 12 yrs): 2

Infants per Room (under 4 yrs): Select

[Change >>](#)

Room and Rate:

Spa Pavilion

Total Price: THB 72,000

Average Daily Rate: THB 18,000

Mar. 18, 2004 - Mar. 20, 2004: THB 18,000

[Change >>](#)

*Title: Mr.

* First Name: Jugarin

* Last Name: Viboonsin

Company: KMITL

* E-mail: j_viboonsin@yahoo.com

* Telephone: 023226114

* Credit Card: MASTER CARD

* Card Number: 5577721906867208

* Expired: October 2006

Please note >> Your card must be valid for your entire stay.

Flight Number: TG 764

Arrival Date: 18 March 2004

Time: 6:00 PM

Birth Date: 7 June 1980

Contact Address: 153/114 On Nut Rd. Pr

City/ Country/ Zip: Bangkok Thailand 10250

Special Requests: Something special

Next

Local intranet

รูปที่ 5.27 แสดงหน้าจอ Guest Information ของ Customer

จากรูปที่ 5.27 Customer ทำการกรอกข้อมูลของตัวเอง ซึ่งในส่วนที่มีเครื่องหมาย (*) Customer จำเป็นต้องกรอกให้ครบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/user/SaveGuestReservation Go Links

Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Stay Info 2 Room Availability 3 Guest Info 4 Confirmation

Please review the information. If the information is correct please click Reserve.

Stay Information:

Check in: Mar. 18, 2004, Thursday
Check out: Mar. 20, 2004, Saturday
2 Rooms, 2 Adults, 2 Children, 0 Infants

Room and Rate:

Spa Pavilion
Total Price: THB 72,000
Average Daily Rate: THB 18,000
Mar. 18, 2004 - Mar. 20, 2004: THB 18,000

Guest Information:

Name: Mr. Jugarin Viboonsin
Company: KMTEL
E-mail: j.viboonsin@yahoo.com
Telephone: 023226114

Guaranteed by: MASTER CARD
Number: xxxxxxxxxxxx7200
Expired: October 2006

Flight Number: TG 764
Arrival Date: Mar. 18, 2004, Thursday
Time: 5:00 PM

Birth Date: Jun. 07, 1980, Saturday
Contact Address: 153/114 On Nut Rd. Pravee
City/ Country/ Zip Code: Bangkok / Thailand / 10250

Special Request:
Something special

Stay Information:

Check in: 18 March 2004
Check out: 20 March 2004

Number of Rooms: 2
Adults per Room: 2
Child per Room: 2 (4 - 12 yrs)
Infants per Room: Select (under 4 yrs)
[Change>>](#)

Room and Rate:

Spa Pavilion
Total Price: THB 72,000
Average Daily Rate: THB 18,000
Mar. 18, 2004 - Mar. 20, 2004: THB 18,000
[Change>>](#)

Guest Information:

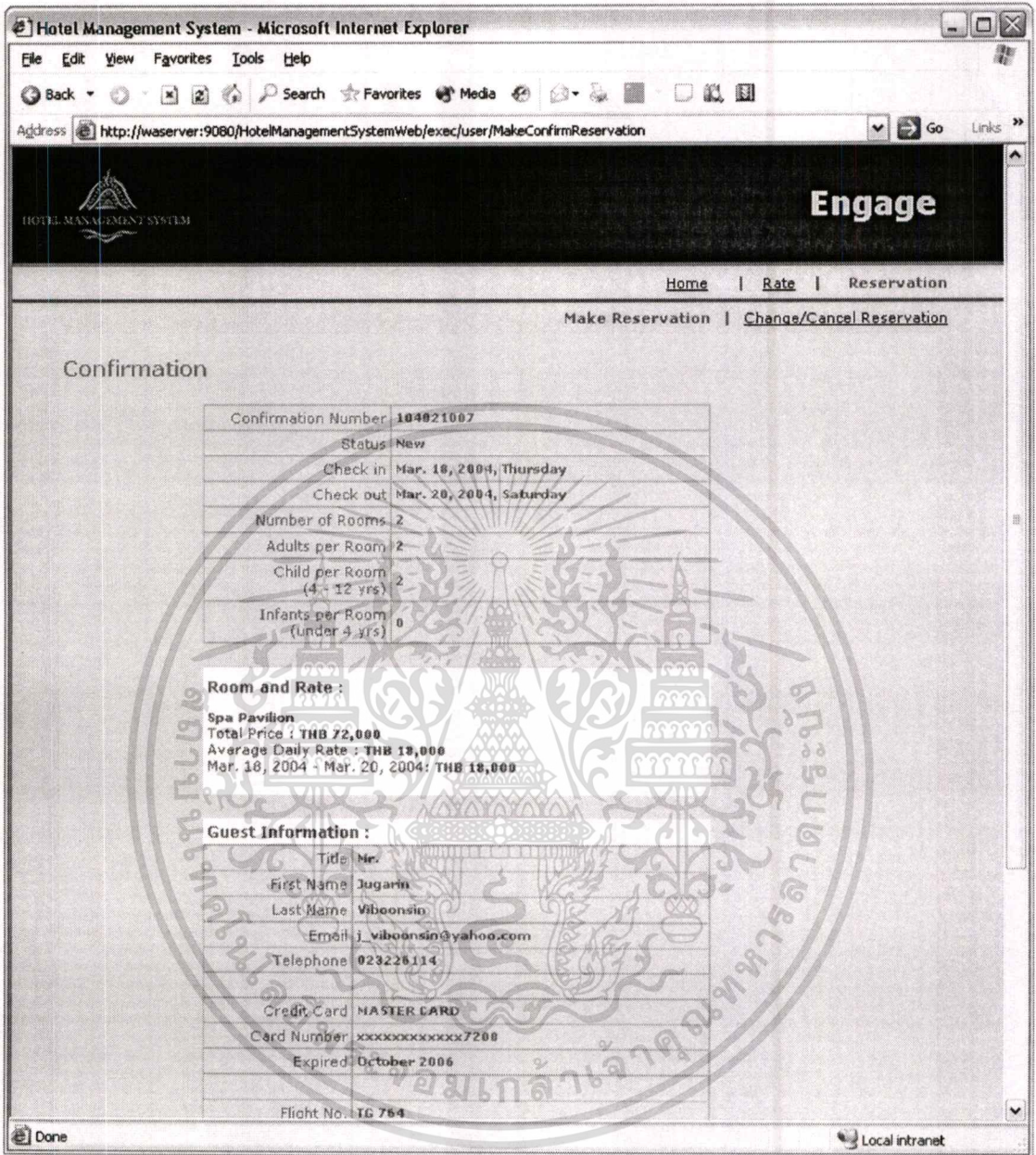
Mr. Jugarin Viboonsin
E-mail: j.viboonsin@yahoo.com
Telephone: 023226114
[Change>>](#)

[Reserve](#)

Done Local intranet

รูปที่ 5.28 แสดงหน้าจอ Confirmation ของ Customer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.29 แสดงหน้าจอการจองห้องเสร็จสมบูรณ์ของ Customer

จากรูปที่ 5.29 Customer ควรจะพิมพ์ข้อมูลนี้เก็บไว้เป็นหลักฐานกับทางโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Customer : Change/Cancel Reservation

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address <http://waserver:9080/HotelManagementSystemWeb/exec/user/SearchReservation> Go Links

Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Search 2 Reservation List 3 Change/Cancel

Please enter your information below, then click GO

Credit Card Number:

Last Name:

Or you may enter the Confirmation Number and the Last Name to go directly to the specific reservation

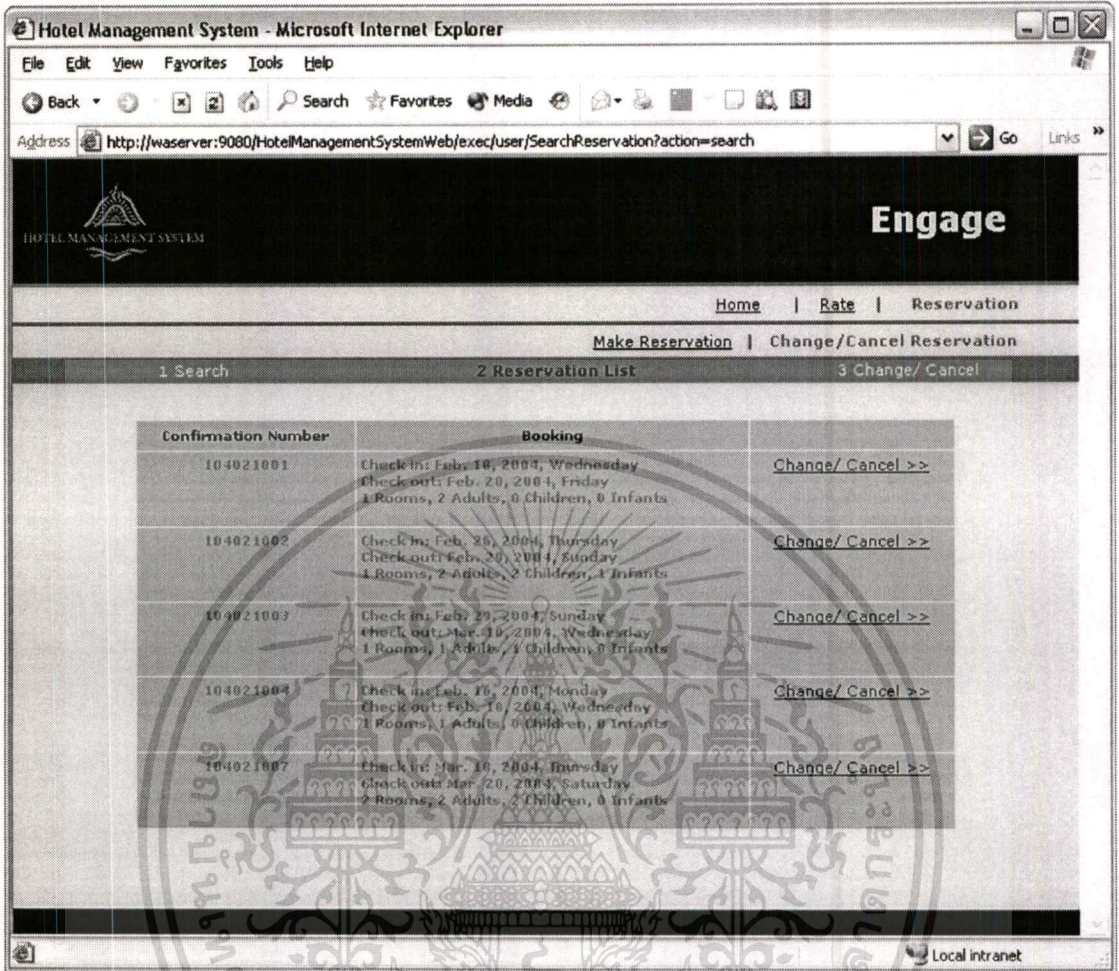
Confirmation Number:

Last Name:

Local intranet

รูปที่ 5.30 แสดงหน้าจอการค้นหาข้อมูลการจอง

จากรูปที่ 5.30 Customer สามารถเลือกได้ว่าจะค้นหาด้วยวิธีใด คืออาจจะค้นหาด้วย Credit Card Number และนามสกุล หรือ Confirmation Number และนามสกุล



รูปที่ 5.31 แสดงหน้าจอผลลัพธ์ของการค้นหาข้อมูลการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print

Address http://waserver:9080/HotelManagementSystemWeb/exec/User/AmendReservationInfo?index=4&type=creditcardno Go Links

Engage

Home | Rate | Reservation

Make Reservation | Change/Cancel Reservation

1 Search 2 Reservation List 3 Change/Cancel

Check in*	Mar. 18, 2004
Check out*	Mar. 20, 2004
Number of Rooms*	2
Adults per Room*	2
Child per Room (4 - 12 yrs)	2
Infants per Room (under 4 yrs)	0

Room and Rate :

Spa Pavilion
 Total Price : THB 72,000
 Average Daily Rate : THB 18,000
 Mar. 18, 2004 - Mar. 20, 2004: THB 18,000

Guest Information :

Title	Mr. ▼
First Name*	Dugarin
Last Name*	Viboonsin
Company	KMITL
Email*	j_viboonsin@y...

Local intranet

รูปที่ 5.32 แสดงหน้าจอเมื่อ Customer เลือกที่จะแก้ไขข้อมูล หรือยกเลิกการจอง

จากรูปที่ 5.32 Customer สามารถแก้ไขข้อมูลของตัวเองได้เท่านั้น และจากหน้าจอนี้สามารถที่จะยกเลิกการจองครั้งนี้ได้

Hotel Management System - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Media

Address http://waserver:9080/HotelManagementSystemWeb/exec/user/EditReservation Go Links

Engage

Home | Rate | Reservation

[Make Reservation](#) | [Change/Cancel Reservation](#)

Change/Cancel Reservation

Confirmation Number	104021007
Status	Changed
Check in	Mar. 18, 2004, Thursday
Check out	Mar. 20, 2004, Saturday
Number of Rooms	2
Adults per Room	2
Child per Room	2
Infants	0

Room and Rate :

Spa Pavilion
 Total Price : THB 72,000
 Average Daily Rate : THB 18,000
 Mar. 18, 2004 - Mar. 20, 2004: THB 18,000

Guest Information :

Title	Mr.
First Name	Jugarin
Last Name	Viboonsin
Email	j.viboonsin@yahoo.com
Telephone	028226114

Done Local Intranet

รูปที่ 5.33 แสดงหน้าจอการแก้ไขการจองห้องเสร็จสมบูรณ์ของ Customer

จากรูปที่ 5.33 Customer ควรจะพิมพ์ข้อมูลนี้เก็บไว้เป็นหลักฐานกับทางโรงแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุป และข้อเสนอแนะ

การพัฒนาเว็บไซต์ระบบโรงแรมโดยใช้เทคโนโลยีอินเทอร์เน็ตพร็อทจาวาเบินส์ มีวัตถุประสงค์เพื่อออกแบบ และพัฒนาระบบโรงแรมผ่านระบบอินเทอร์เน็ต ซึ่งช่วยอำนวยความสะดวกให้กับนักท่องเที่ยวทั้งใน และนอกประเทศ อีกทั้งยังนำเอาระบบสารสนเทศมาช่วยในเรื่องของการบริการ และมีความสะดวกในเรื่องของการเข้าถึงข้อมูลเนื่องจากได้นำเอาเทคโนโลยีอินเทอร์เน็ตพร็อทจาวาเบินส์มาช่วยในการพัฒนา

สำหรับเหตุผลที่เลือกพัฒนาระบบโรงแรมขึ้นมาเป็นเพราะว่าในขณะนี้รัฐบาลได้มีนโยบายในการส่งเสริมการท่องเที่ยวภายในประเทศ ซึ่งจะส่งผลโดยตรงถึงธุรกิจโรงแรม ทำให้ต้องมีการปรับเปลี่ยนแนวทางการให้บริการเพื่อตอบสนองนโยบายของรัฐบาล รวมถึงการเพิ่มประสิทธิภาพในการให้บริการแก่นักท่องเที่ยวที่เพิ่มขึ้น

6.1 ข้อเสนอแนะ

เว็บไซต์ระบบโรงแรมที่ได้พัฒนาขึ้นมา เป็นส่วนหนึ่งของความพยายามที่จะนำเทคโนโลยีอินเทอร์เน็ตพร็อทจาวาเบินส์มาใช้ให้เกิดประโยชน์ ซึ่งเน้นในส่วนของการสำรองห้องพัก การออกรายงานแก่ผู้ดูแลระบบ การจัดการห้องพัก เป็นต้น ดังนั้นหากจะนำไปใช้งานให้ได้อย่างมีประสิทธิภาพนั้นควรจะต้องมีการออกแบบเว็บไซต์ให้ดูมีสีสัน และน่าสนใจยิ่งขึ้น รวมถึงการเพิ่มเติมรายละเอียดในส่วนอื่น ๆ เพื่อนำเสนอความน่าสนใจของโรงแรม อาทิเช่นการมีโปรโมชั่นต่าง ๆ การมีเฟ็กเกจการท่องเที่ยว หรือการนำเสนอรูปแบบของห้องพักผ่านทางเว็บไซต์

อีกทั้งในส่วนของผู้ดูแลระบบจำเป็นต้องมีการตรวจสอบในเรื่องของความปลอดภัย เพื่อป้องกันไม่ให้ผู้อื่นเข้าไปแก้ไขข้อมูลได้ เพราะระบบโรงแรมที่ได้พัฒนาขึ้นมาได้ออกแบบในส่วนของผู้ดูแลระบบ แต่ไม่ได้จัดการในเรื่องของความปลอดภัยในส่วนนี้ เนื่องจากต้องการนำเสนอในส่วนของการสำรองห้องพัก การจัดการห้องพัก และการออกรายงานเพื่อดูผลของธุรกิจ

6.2 ประโยชน์ที่คาดว่าจะได้รับ

6.2.1 เพิ่มทางเลือกในการสำรองห้องพักให้กับนักท่องเที่ยว เนื่องจากเมื่ออยู่บนอินเทอร์เน็ตแล้วนั้นนักท่องเที่ยวก็สามารถที่จะเข้ามาใช้บริการของทางโรงแรมได้จากทุกที่ จึงเป็นการเพิ่ม

6.2.2 เพิ่มทางเลือกในการทำงานกับฐานข้อมูลโดยใช้เอ็นเตอร์ไพรส์จาวาบีเอส เนื่องจากการทำงานกับฐานข้อมูลแต่เดิมอาจจะมีการฝัง query statement ไว้ร่วมกับตัวโปรแกรม หรืออาจจะเขียนเป็น store procedure แล้วเรียกใช้ การใช้เอ็นเตอร์ไพรส์จาวาบีเอสจึงเป็นอีกทางเลือกหนึ่งของผู้พัฒนาโปรแกรมภาษาจาวาที่จะใช้ในการจัดการกับฐานข้อมูล

6.2.3 การจัดการในเรื่องของห้องพักเป็นไปได้ง่าย เพราะการทำงานอยู่ในรูปแบบของเว็บเพจซึ่งเห็นการเปลี่ยนแปลงได้ทันที รวมถึงยังสามารถเข้ามาจัดการได้จากทุกที่

6.2.4 การสำรองห้องพักเป็นไปได้สะดวก และรวดเร็ว เนื่องจากการสำรองห้องพักมีขั้นตอนที่แน่นอน และมีการแจ้งผลในทุกขั้นตอนที่ได้ทำผ่านมา และเมื่อมีการยืนยันการสำรองห้องพักระบบจะทำการส่ง e-mail แจ้งการยืนยันนั้นต่อลูกค้าด้วย

6.2.5 มีการแสดงรายงานต่าง ๆ ที่ผู้ดูแลระบบสามารถนำมาวิเคราะห์เพื่อปรับเปลี่ยนกลยุทธ์ เพื่อนำมาเพิ่มผลกำไรแก่ทางโรงแรมต่อไป

บรรณานุกรม

โอภาส เอี่ยมศิริวงศ์. 2545. การออกแบบและจัดการฐานข้อมูล. กรุงเทพฯ: ซีเอ็ดบุ๊คเซ็น.

ญาติ กาชัย. 2546. จัดการระบบฐานข้อมูลอย่างมืออาชีพ Oracle DBA. นนทบุรี:

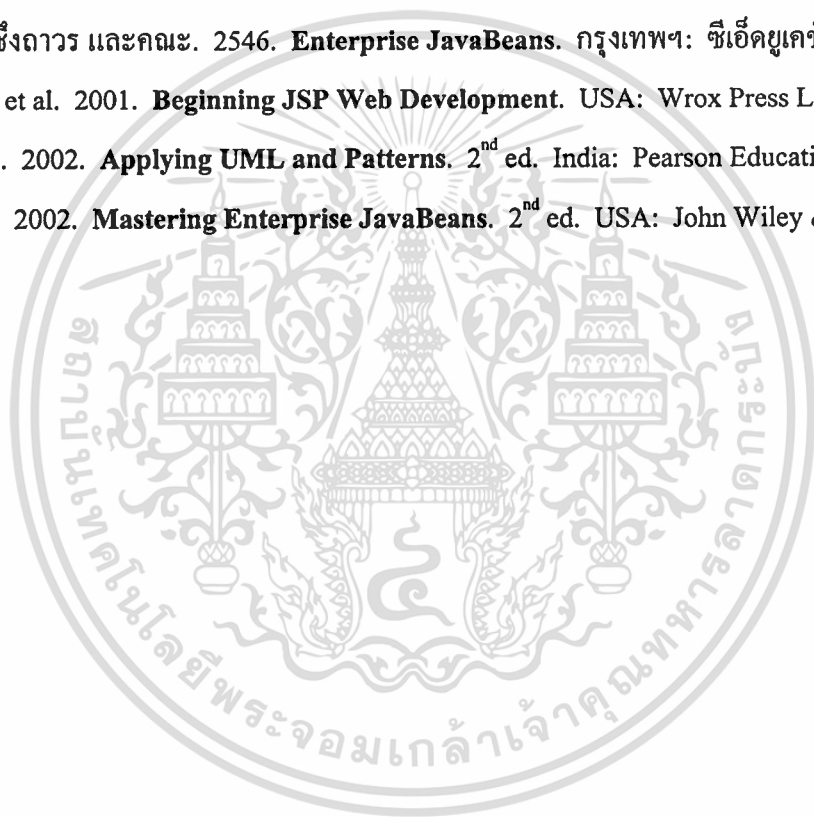
อินโฟเพรส.

วีระศักดิ์ ชิงदार และคณะ. 2546. **Enterprise JavaBeans**. กรุงเทพฯ: ซีเอ็ดบุ๊คเซ็น.

Falkner, J. et al. 2001. **Beginning JSP Web Development**. USA: Wrox Press Ltd.

Larman, C. 2002. **Applying UML and Patterns**. 2nd ed. India: Pearson Education.

Roman, E. 2002. **Mastering Enterprise JavaBeans**. 2nd ed. USA: John Wiley & Sons, Inc.



ประวัติผู้เขียน

ชื่อผู้เขียน	นายจักรินทร์ วินุทธ์ศิลป์
วันเดือนปีเกิด	7 มิถุนายน 2523
สถานที่เกิด	นครศรีธรรมราช
มัธยมศึกษา	โรงเรียนกัลยาณีศรีธรรมราชศึกษา
ปริญญาตรี	สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2544