

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

ระบบจัดการคลังข้อสอบและการสอบ

Test and Test Bank Management System



ดร. จันทร์บุรณ์ สถิตวิริยวงศ์



H001860

รายงานนี้เป็นส่วนหนึ่งของโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบจัดการคลังข้อสอบและการสอบ
นักศึกษา	นาย บุญชัย แซ่ปู้
อาจารย์ที่ปรึกษา	ดร. จันทร์บุรณีย์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

โครงการพัฒนาระบบจัดการคลังข้อสอบและการสอบ มีวัตถุประสงค์เพื่อพัฒนาเครื่องมือส่งเสริมการเรียนการสอนและเพิ่มประสิทธิภาพในการวัดผล โดยใช้เทคโนโลยีเชิงวัตถุในการพัฒนา, ใช้ Unified Modeling Language(UML) ในการออกแบบโมเดล, ใช้ JavaScript ในการออกแบบหน้าจอการติดต่อกับผู้ใช้ และใช้ PL/SQL Gateway ในการเรียกใช้ Stored Procedure จากฐานข้อมูลที่พัฒนาแอปพลิเคชันทางด้าน Server

การทำงานของระบบคือให้บริการในการสอบและการประเมินผล ผู้สอนสามารถสร้างข้อสอบและกำหนดคุณสมบัติต่างๆของข้อสอบเช่นประเภทคำถาม, ระดับความยากง่ายของคำถาม และนักเรียนจะสามารถเข้ามาสอบภายในระบบเพื่อวัดผล ผลจากการพัฒนาระบบเชิงวัตถุทำให้ได้ระบบที่มีการพัฒนาเป็นมาตรฐานและสะดวกในการดูแลปรับปรุง นอกจากนี้ประโยชน์ที่จะได้รับคือเข้าใจถึงขั้นตอนการพัฒนาในระบบในเชิงวัตถุ และเข้าใจปัญหาที่เกิดขึ้นทำให้ช่วยลดขั้นตอนการทำงานที่ซ้ำซ้อนของระบบเดิม

Title	Test and Test Bank Management System
Student	Mr. Boonchai Saeboo
Advisor	Dr. Chanboon Sathitwiriya Wong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

ABSTRACT

The purpose of Test and Test Bank Management System Development project is to develop a tool to enhance teaching, teaching and effectiveness of result measurement. The design and development is based on the object oriented technology. Unified Modeling Language is used to design models. The user interface is written using java and server application is developed using PL/SQL stored procedure and invoked through PL/SQL Gateway.

The main function of the tool is to provide service in testing and grading. Instructors can build tests and define attributes, for example, category and difficulty level to them. Students can take tests and have them graded by the tool. The object-oriented design provide a standardized development and ease of maintenance. Further benefits are the insight of object oriented development methodology and the understanding of arising problems which reduces redundant steps of old system.

กิตติกรรมประกาศ

ในโครงการฉบับนี้ ที่สำเร็จลุล่วงไปด้วยดีนั้น เนื่องด้วยได้รับคำแนะนำ และความช่วยเหลือสนับสนุน รวมถึงกำลังใจที่ดี ทั้งนี้ผู้จัดทำขอกล่าวขอบคุณกับบุคคลและกลุ่มบุคคลต่างๆ ดังนี้

บิดามารดา ที่เป็นผู้ให้ทุกอย่างในชีวิต รวมถึงกำลังใจที่มีให้ตลอดมา

ดร. จันทร์บุรณีย์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษาวิชาโครงการพัฒนาระบบงาน ขอขอบคุณสำหรับคำแนะนำ ข้อเสนอแนะ และการสนับสนุนการทำโครงการ

คุณสุจิต ฟ้าสว่าง ที่คอยให้ความสนับสนุนทางการศึกษาตลอดมา

พี่จรรยา ฟ้าสว่าง พี่ชลิตา ที่คอยให้ข้อเสนอแนะและเป็นที่ปรึกษาในการจัดทำโครงการ

คุณกิตติยา ที่ให้คำปรึกษาแนะนำตลอดการทำโครงการ



บุญชัย แซ่บู๊
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ.....	1
1.1 วัตถุประสงค์.....	1
1.2 ขอบเขตของการพัฒนาระบบ.....	2
1.3 หลักวิธีที่ใช้ในการพัฒนาระบบ.....	2
1.4 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. การพัฒนา Stored Procedure ด้วย Java.....	4
2.1 Java Stored Procedure.....	4
2.2 JServer JVM และส่วนประกอบ.....	8
2.3 ขั้นตอนในการพัฒนา Stored Procedure.....	12
3. การวิเคราะห์และออกแบบระบบจัดการคลังข้อสอบและการสอบ.....	31
3.1 ขอบเขตการทำงานของระบบจัดการคลังข้อสอบและการสอบ.....	31
3.2 ขั้นตอนการทำงาน.....	32
3.3 การวิเคราะห์และออกแบบระบบ.....	33
4. การสร้างฐานข้อมูลระบบระบบจัดการคลังข้อสอบและการสอบ.....	49
4.1 การแปลงแบบจำลองเป็นฐานข้อมูล.....	49
4.2 รายละเอียดข้อมูลที่จัดเก็บในระบบ.....	49
5. การพัฒนาระบบงาน.....	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
5.1 การพัฒนาแอปพลิเคชันใช้งาน.....	57
5.2 ขั้นตอนและวิธีการพัฒนาแอปพลิเคชัน.....	58
5.3 การทดสอบระบบ.....	64
5.4 ผลการทดสอบระบบ.....	66
6. บทสรุป.....	68
6.1 ผลการพัฒนาระบบงาน.....	68
6.2 ประโยชน์ที่ได้รับ.....	68
6.3 ข้อเสนอแนะ.....	69
บรรณานุกรม.....	70
ภาคผนวก ก.....	71
ประวัติผู้เขียน.....	84

สารบัญตาราง

	หน้า
ตารางที่	
2.1 รายละเอียดของ Option ของ loadjava.....	15
3.1 แอททริบิวต์ของคลาส Site.....	45
3.2 แอททริบิวต์ของคลาส TestBank.....	46
3.3 แอททริบิวต์ของคลาส Question.....	46
3.4 แอททริบิวต์ของคลาส Answer.....	46
3.5 แอททริบิวต์ของคลาส Learner.....	46
3.6 แอททริบิวต์ของคลาส Test.....	47
3.7 แอททริบิวต์ของคลาส TestGrade.....	48
3.8 แอททริบิวต์ของคลาส TestScore.....	48
3.9 แอททริบิวต์ของคลาส TestLevel.....	48
4.1 Site	50
4.2 UserSite.....	51
4.3 SessionSite.....	51
4.4 TestbankSite.....	52
4.5 Question.....	52
4.6 Answer.....	53
4.7 Test.....	53
4.8 TestLevel.....	54
4.9 TestGrade.....	55
4.10 TestScore.....	55
4.11 TestFixedQuestion.....	55
4.12 TakenTest.....	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

รูปที่	หน้า
2.1 ระบบ Client/Server แบบ 2 tier.....	5
2.2 การเรียกใช้ Stored Procedure.....	5
2.3 ส่วนประกอบหลักของ JServer JVM.....	9
2.4 การเปรียบเทียบระหว่าง Interpreter และ JServer Accelerator.....	12
2.5 การโหลดไฟล์จาวาเข้าสู่ฐานข้อมูล.....	13
2.6 การเรียกใช้ Java Method.....	17
3.1 แอคทิวิตีไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบ.....	34
3.2 แอคทิวิตีไดอะแกรมการเข้าสอบของนักเรียน.....	34
3.3 ยูสเคสไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบ.....	35
3.4 ซีเควนซ์ไดอะแกรมของยูสเคส Create Site.....	37
3.5 ซีเควนซ์ไดอะแกรมของยูสเคส Create User.....	38
3.6 ซีเควนซ์ไดอะแกรมของยูสเคส Create Test Bank.....	38
3.7 ซีเควนซ์ไดอะแกรมของยูสเคส Create Test.....	39
3.8 ซีเควนซ์ไดอะแกรมของยูสเคส Take a Test.....	39
3.9 ซีเควนซ์ไดอะแกรมของยูสเคส Grade a Test.....	40
3.10 ซีเควนซ์ไดอะแกรมของยูสเคส View Score.....	40
3.11 สแตทไดอะแกรมแสดงสถานะของผู้สอบ.....	41
3.12 สแตทไดอะแกรมแสดงสถานะของนักเรียน.....	41
3.13 คลาสไดอะแกรมระบบจัดการคลังข้อสอบและการสอบในขั้นตอนการวิเคราะห์.....	42
3.14 คลาสไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบในขั้นตอนการออกแบบ...	44
5.1 หน้าจอในส่วน Gateway Configuration Menu.....	59
5.2 หน้าจอส่วนการเลือกวิธีการสร้าง DAD.....	59
5.3 หน้าจอในส่วนการสร้าง DAD.....	60
ก.1 หน้าจอการ Login เข้าสู่ระบบ.....	71
ก.2 หน้าจอการสร้าง Site ใหม่.....	72
ก.3 หน้าจอรายการหลักของระบบ.....	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่

ก.4	หน้าจอการจัดการ Site.....	73
ก.5	หน้าจอการจัดการนักเรียน.....	74
ก.6	หน้าจอการเพิ่มข้อมูลนักเรียน.....	75
ก.7	หน้าจอการจัดการคลังข้อสอบ.....	76
ก.8	หน้าจอการแก้ไขคลังข้อสอบ.....	77
ก.9	หน้าจอการจัดการคำถาม.....	77
ก.10	หน้าจอการเพิ่มคำถาม.....	78
ก.11	หน้าจอการป้อนข้อมูลคำถาม.....	78
ก.12	หน้าจอการเพิ่มคลังข้อสอบ.....	79
ก.13	หน้าจอการจัดการข้อสอบ.....	80
ก.14	หน้าจอการเพิ่มข้อสอบ.....	80
ก.15	หน้าจอการ Login เข้าสู่ระบบของนักเรียน.....	81
ก.16	หน้าจอแสดงรายชื่อข้อสอบ.....	82
ก.17	หน้าจอการสอบ.....	82
ก.18	หน้าจอสิ้นสุดการสอบ.....	83
ก.19	หน้าจอแสดงประวัติการสอบ.....	83

บทที่ 1

บทนำ

ในปัจจุบันการแข่งขันในธุรกิจมีสูงมาก หลายๆองค์กรมีการพัฒนาอยู่ตลอดเวลาเพื่อทัดเทียมคู่แข่งกันในเชิงธุรกิจ เทคโนโลยีสารสนเทศ (Information Technology) เป็นสิ่งที่จำเป็นสำหรับการพัฒนาขององค์กร ดังนั้นการพัฒนาระบบสารสนเทศจึงมีผลต่อการดำเนินการขององค์กรเช่นกัน แต่การพัฒนาระบบสารสนเทศมีหลายแนวทางและประกอบกับการแข่งขันทางธุรกิจมีการเปลี่ยนแปลงตลอดเวลา แนวทางที่จะใช้ในการพัฒนาระบบสารสนเทศจึงควรเป็นแนวทางที่สามารถรองรับการเปลี่ยนแปลงที่เกิดขึ้นได้(รวมถึงเครื่องมือที่จะนำมาใช้ในการพัฒนาระบบตามแนวทางด้วย) การพัฒนาระบบสารสนเทศคือการนำระบบสารสนเทศมาแก้ไขปัญหาหรือเพิ่มประสิทธิภาพในองค์กร ประกอบด้วยขั้นตอนหลักๆดังนี้ การวิเคราะห์ระบบ การสร้างแบบจำลอง การออกแบบระบบ การพัฒนาระบบ การทดสอบ และการบำรุงดูแลรักษาระบบ

การพัฒนาระบบซอฟต์แวร์ในลักษณะเดิมที่เป็นการพัฒนาในลักษณะ Function-Oriented เป็นการพัฒนาที่ขาดความยืดหยุ่น เนื่องจากเป็นการพัฒนาซอฟต์แวร์ที่อิงกับวิธีการทำงานของแต่ละองค์กรเป็นหลัก ทำให้ระบบซอฟต์แวร์ที่พัฒนากลายเป็นซอฟต์แวร์ที่ใช้ได้เฉพาะภายในองค์กรที่พัฒนาแล้วเท่านั้น ไม่สามารถนำระบบดังกล่าวไปใช้ภายในองค์กรอื่นๆได้ ถึงแม้จะการทำงานในลักษณะเดียวกันก็ตาม และหากมีการเปลี่ยนแปลงแก้ไขบางส่วนในระบบ ก็จะมีผลกระทบต่อส่วนต่างๆในระบบด้วย ทำให้การดูแลรักษาระบบซอฟต์แวร์เป็นไปด้วยความยากลำบากและสิ้นเปลืองค่าใช้จ่ายในการดูแลสูง ดังนั้นแนวทางการพัฒนาในปัจจุบันจึงมีการนำแนวคิดของการพัฒนาในเชิงวัตถุ (Object Oriented) มาใช้ เพื่อแก้ไขปัญหาที่เกิดขึ้นกับแนวการพัฒนาระบบในลักษณะเดิม

ระบบจัดการคลังข้อสอบและการสอบ (Test and Test Bank Management System) ถูกพัฒนาโดยใช้แนวการพัฒนาระบบในเชิงวัตถุ เพื่อช่วยเพิ่มประสิทธิภาพในการจัดการคลังข้อสอบและการสอบให้ดีขึ้นจากเดิม อีกทั้งยังช่วยแก้ไขปัญหาเดิมที่เคยมีในระบบให้ ด้วยการนำเทคโนโลยีสารสนเทศมาใช้ในการพัฒนาระบบ

1.1 วัตถุประสงค์

เพื่อให้ระบบที่พัฒนาเป็นตามความต้องการของผู้ใช้ จึงมีวัตถุประสงค์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (1) เพื่อพัฒนาเครื่องมือส่งเสริมการเรียนการสอน ให้ผู้สอนสามารถสร้างแบบฝึกหัดหรือข้อสอบให้นักเรียนได้อย่างสะดวก
- (2) ลดขั้นตอนในการทำงานด้านการวัดผลการเรียน ระบบจะสามารถทำการตรวจสอบได้ด้วยตัวเอง เป็นการลดภาระในการทำงานให้กับผู้สอน
- (3) เพื่อเกิดประสิทธิภาพวัดผลในการสอบ โดยจะกำหนดให้ระบบทำการสร้างข้อสอบชุดใหม่ทุกครั้ง เมื่อมีการสอบเกิดขึ้น
- (4) เพิ่มความยืดหยุ่นของเวลา ในการเรียนรู้ ทบทวนและฝึกฝนของนักเรียนให้มากขึ้น นักเรียนสามารถเข้าสู่ระบบเพื่อทดสอบความรู้ได้ตามเวลาและสถานที่ที่ต้องการ

1.2 ขอบเขตของการพัฒนาระบบ

ระบบจัดการคลังข้อสอบและการสอบเป็นระบบงานที่จัดทำขึ้นเพื่อให้บริการด้านจัดการข้อสอบให้แก่ผู้สอน ขั้นตอนการทำงานที่สำคัญประกอบด้วยงานหลักๆ ดังนี้

- (1) ผู้สอนสร้าง Site ภายในระบบ เพื่อเตรียมสำหรับการจัดการคลังข้อสอบ
- (2) ผู้สอนสร้างคลังข้อสอบเก็บไว้ภายใน Site
- (3) ผู้สอนสร้างข้อสอบโดยนำคำถามมาจากคลังข้อสอบ
- (4) ผู้สอนสร้างรายชื่อนักเรียนและกำหนดข้อสอบที่จะให้นักเรียนทำการสอบ
- (5) นักเรียนเข้าสู่ระบบเพื่อทำการสอบ
- (6) ระบบทำการตรวจสอบผลสอบ
- (7) ระบบทำการประกาศผลสอบ

1.3 หลักวิธีที่ใช้ในการพัฒนาระบบ

ขั้นตอนการดำเนินงานในการวิเคราะห์และออกแบบระบบจัดการคลังข้อสอบและการสอบ จะใช้วิธีเชิงวัตถุ (Object-Oriented Methodology) สำหรับภาษาที่ใช้ออกแบบโมเดลคือ Unified Modeling Language (UML)

1.4 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในการศึกษาวิเคราะห์และพัฒนาระบบงานนี้ มีเครื่องมือที่ใช้ คือ

- (1) CASE Tool ใช้สำหรับการสร้าง Model ที่มีชื่อว่า Rational Rose 2000
- (2) ระบบจัดการฐานข้อมูล Oracle Version 8.1.7

- (3) เครื่องมือสำหรับสร้างหน้าจอการทำงานของแอปพลิเคชัน คือ Macromedia Dreamweaver 4.0

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ในการพัฒนาระบบจัดการคลังข้อสอบและการโดยใช้แนวทางเชิงวัตถุและ CASE Tool คาดว่าจะได้รับประโยชน์ดังนี้

- (1) มีการนำแนวคิดเชิงวัตถุมาใช้ในการพัฒนาระบบ เพื่อให้มีความยืดหยุ่นและง่ายต่อการดูแลรักษา
- (2) ทำให้เข้าใจขั้นตอนในการนำ CASE Tool มาใช้ในการวิเคราะห์และออกแบบระบบ
- (3) ลดขั้นตอนการทำงานที่ซ้ำซ้อนในบางขั้นตอนของระบบเดิม
- (4) ได้ระบบจัดการคลังข้อสอบที่ทั้งผู้สอนและนักเรียนได้รับความสะดวกทั้ง 2 ฝ่าย อีกทั้งยังช่วยเพิ่มประสิทธิภาพและความยืดหยุ่นในการสอบ



บทที่ 2

การพัฒนา Stored Procedure ด้วย Java

2.1 Java Stored Procedure

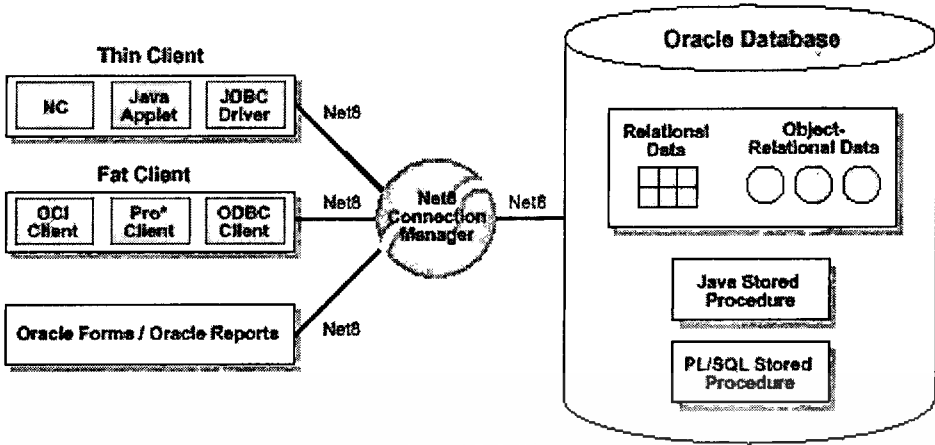
Oracle RDBMS ทำงานร่วมกับแอปพลิเคชัน Java โดยสามารถทำ query ที่ซับซ้อน และมี view ที่แตกต่างกันได้ เมื่อ client ส่งคำสั่งในรูปแบบ data query จะถูกปฏิบัติงานและได้รับผลลัพธ์ทันที

แนวคิด Java มีลักษณะเด่นหลายประการ เช่น ให้คุณสร้างแอปพลิเคชันโดยใช้ off-the-shelf software component (JavaBeans) ที่จัดการหน่วยความจำอย่างอัตโนมัติและปลอดภัย โดยทำงานร่วมกับ RDBMS และ Java ยังสามารถกระจายส่วนประกอบของแอปพลิเคชันไปตามเน็ตเวิร์กแบบ transparent Java และ RDBMS ทำให้สามารถสร้างแอปพลิเคชันได้เร็วจากการนำ component มาประกอบกัน และแอปพลิเคชันแบบ network-centric สามารถพลิกโฉมรูปแบบแอปพลิเคชันให้เหมาะสมกับแนวโน้มของธุรกิจได้ คือย้ายการเก็บแอปพลิเคชันจากเครื่อง desktop ไปไว้ใน network-centric server แล้วเรียกใช้งานผ่านระบบเครือข่าย และที่สำคัญ คุณสามารถใช้แอปพลิเคชันและข้อมูลจากเครื่อง client ใดๆก็ได้

รูปที่ 2.1 แสดงระบบ Client/Server แบบ 2 tier ซึ่ง client สามารถใช้ Java stored procedure เหมือนกับเรียกใช้งาน PL/SQL stored procedure และแสดง Net8 Connection Manager ทำหน้าที่รองรับการเชื่อมต่อจาก client หลายลักษณะและหลายตัว เป็นการเชื่อมต่อเดียวสู่ฐานข้อมูล ทำให้ RDBMS สามารถรองรับการใช้งานพร้อมกันจากผู้ใช้จำนวนมากได้

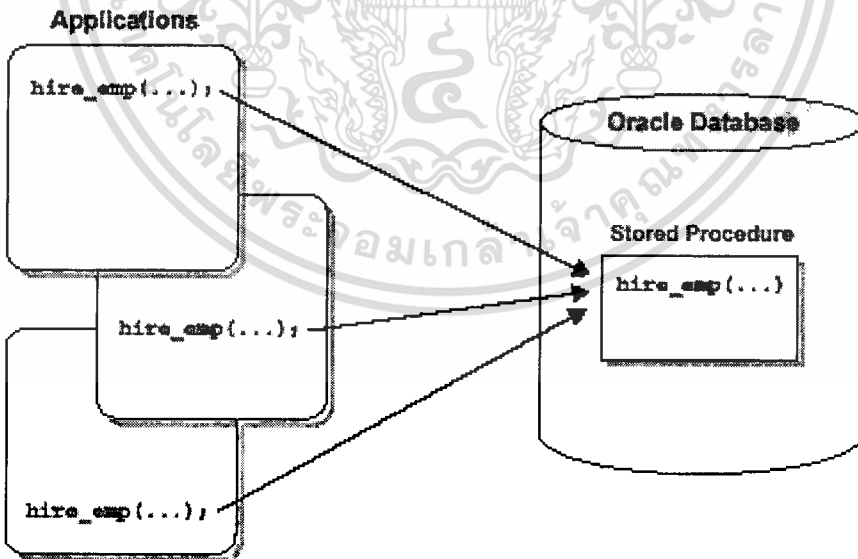
Java Stored Procedure คือการใช้งาน Java ที่เก็บอยู่ในฐานข้อมูล Oracle เช่นเดียวกับ PL/SQL Stored Procedure แต่แตกต่างที่การใช้ภาษาในการทำงานของ Stored Procedure และ Java ยังเป็นภาษาที่มีการใช้งานอย่างกว้างขวางในอุตสาหกรรมซอฟต์แวร์ ทำให้การใช้ Java ในฐานข้อมูล Oracle อำนวยความสะดวกแก่นักพัฒนาในกรณีที่ต้องการใช้ Java ในการทำงาน

การประกาศ Method ของ Java ทำโดยเขียน Call Specifications (Call Spec) เพื่อจับคู่ระหว่างชื่อ Method ของ Java, ชนิดตัวแปรของ Parameter และค่าที่ส่งคืน ซึ่งไม่เหมือนกับการใช้ Wrapper ที่ห่อหุ้ม Procedure ทั้งหมดก่อนที่จะส่งไป Layer ถัดไปเพื่อเรียกใช้งาน Call Spec จึงทำให้ระบบ Run-Time ทำงานได้โดยเสีย Overhead น้อยมาก



รูปที่ 2.1 ระบบ Client/Server แบบ 2 tier
(Tom. 1999 : 1-2)

เมื่อแอปพลิเคชันของ Client เรียกใช้งาน Stored procedure จะรับคำสั่ง แล้วไปอ้างอิงกับ Class ของ Java และคืนค่าผลลัพธ์แบบ Java มาให้ รูปที่ 2.2 แสดง Stored procedure กำลังถูกเรียกใช้งานโดยหลายๆแอปพลิเคชัน



รูปที่ 2.2 การเรียกใช้ Stored Procedure
(Tom. 1999 : 1-3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากต้องการให้สามารถใช้งาน Java Stored Procedure ได้ จะต้องมีารติดตั้ง Oracle8i พร้อมด้วย JServer ก่อน เพื่อให้ฐานข้อมูล Oracle สามารถใช้งานJava ได้นอกจาก Java Stored Procedure เช่น JDBC, SQLJ, และ COBRA/EJB

การติดตั้ง JServer สามารถทำได้ 3 แบบ:

- ทำการติดตั้ง Oracle8i แบบ Typical หรือ แบบ Minimal: ในการติดตั้ง Oracle8i ไม่ว่าจะ เป็นแบบ Typical หรือแบบ Minimal จะทำการติดตั้ง JServer ให้โดยอัตโนมัติ
- ทำการติดตั้ง Oracle8i แบบ Custom: ระบุตัวเลือก JServer ขณะติดตั้ง Oracle8i แบบ Custom
- ติดตั้งแบบ Manual: หากยังไม่มีารติดตั้ง JServer พร้อมกับการติดตั้ง Oracle8i เราสามารถทำการติดตั้ง JServer ที่ Database โดยทำการใช้สคริปต์ initjvm.sql

การนำ Java มาใช้ใน Oracle จะสามารถทำได้ตั้งแต่เวอร์ชัน Oracle8i(8.1.x) เป็นต้นไป ในการใช้ Java Stored Procedure จะสามารถปฏิบัติในการทำงานได้ 3 สภาพแวดล้อมดังนี้

1. การเรียกใช้งานในลักษณะเช่นเดียวกับ SQL Function และ Procedure

Function และ Procedure คือชื่อของ Block ที่ครอบคลุมลำดับของชุดคำสั่ง นำมาประกอบกันเพื่อสร้าง Module หรือแอปพลิเคชันทั่วไป

โดยทั่วไป เราใช้ Procedure เพื่อกระทำงานอย่างหนึ่ง และใช้ Function เพื่อคำนวณค่าใดค่าหนึ่ง ดังนั้นหากต้องการใช้ Method ของ Java โดยไม่ต้องการค่าส่งกลับมา ให้ใช้ Call Spec ของ Procedure และหากต้องการค่าคืนกลับมา ให้ใช้ Call Spec ของ Function

Function และ Procedure ของ PL/SQL ในระดับสูงสุดและ Package เท่านั้นที่ใช้งานแบบ Call Spec ได้ ในการสร้างทำโดยใช้คำสั่ง CREATE FUNCTION, CREATE PROCEDURE และ CREATE PACKAGE ซึ่งจะถูกรวบรวมในฐานข้อมูลที่สามารถใช้งานได้ทั่วไป

Method ของ Java ที่ประกาศเป็น Function หรือ Procedure สามารถเรียกใช้ได้จาก

- คำสั่ง SQL DML (INSERT, UPDATE, DELETE และ SELECT)
- คำสั่ง SQL CALL
- PL/SQL block, subprogram และ package

2. การเรียกใช้ในส่วนของ Database Trigger

Database Trigger คือ Stored Procedure ที่ทำงานเฉพาะใน Table หรือ View ที่กำหนด จะทำงานเองเมื่อคำสั่ง DML ที่กำหนดได้ทำการเปลี่ยนแปลง Table หรือ View นั้น

Trigger มี 3 ส่วนคือ Triggering Event (DML Operation), Optional Trigger Constraint และ Trigger Action เมื่อเกิด Event นั้นขึ้น Trigger จะไปเรียกใช้ Method ของ Java นั้น (Call Spec) เพื่อทำงานต่อไป

เราสามารถสร้าง Database Trigger ขึ้นมาเองโดยคำสั่ง CREATE TRIGGER เพื่อจัดการ RDBMS ได้ตามต้องการ

3. การเรียกใช้ในส่วน Method ของ Object

ประเภทของ SQL Object คือการรวมประเภทของข้อมูลโดยผู้กำหนดเอง เป็นกลุ่มของ ตัวแปรและการทำงาน (Method) สร้างโดยใช้คำสั่ง CREATE ... OBJECT สร้าง Template ที่ กำหนด Attribute และพฤติกรรมที่จำเป็น ในการเรียกใช้จะต้องให้ค่าแล้วสร้าง Object นั้นขึ้นมา และสามารถสร้างได้จำนวนมากตามต้องการ

ในงานที่มีลักษณะมีรายการหลายการ เราสามารถสร้าง Object Type ที่เป็น Array และ Nested Table ได้อีกด้วย

2.1.1 Java Schema Object Type

สิ่งที่แตกต่างจากการใช้งาน Java ปกติคือการใช้ JVM(Java Virtual Machine) ในการ คอมไพล์ไฟล์ Java เพื่อทำการ Load เป็น Object ไปไว้ยังฐานข้อมูล ซึ่งสามารถแบ่งประเภทของ Object เป็น 3 ประเภทดังนี้

- JAVA SOURCE: ตัวโปรแกรมต้นฉบับของคลาสจาวา
- JAVA CLASS: ตัวคลาสจาวาที่ผ่านการคอมไพล์เรียบร้อยแล้ว
- JAVA RESOURCE: คือ ไฟล์อื่นๆที่อาจจะมีการเรียกใช้โดยคลาสจาวา เช่น ไฟล์ข้อมูล ต่างๆ

2.1.2 ข้อดีของ Stored Procedure

- ด้านความสามารถ (Performance) Stored Procedure นั้น Compile เพียงครั้งเดียว แล้ว จัดเก็บไว้ในรูปแบบที่เรียกใช้งานได้ทันที จึงเรียกใช้งานได้เร็วและมีประสิทธิภาพ โดย Load เข้าสู่หน่วยความจำและ Share แก่ผู้ใช้ได้ ทำให้ใช้หน่วยความจำและ Overhead น้อย การจัดกลุ่มของชุดคำสั่งเพื่อทำงานเป็นหน่วยเดียว ทำให้ลดการสื่อสารในระบบเครือข่ายลง และเวลารอการตอบกลับก็ลดลงด้วย การทำงานของคำสั่ง ทำที่ Server ไม่ต้องส่งข้อมูลไปมาระหว่าง Client/Server ขณะที่ปฏิบัติงาน เมื่อสิ้นสุด การทำงานจึงส่งผลลัพธ์กับไปยัง Client นั้น ทำให้สามารถทำงานได้เร็วกว่า
- ด้านผลงานและความง่ายในการใช้งาน (Productivity and Ease of Use) ลดการเขียน โปรแกรมที่ซ้ำซ้อน และเพิ่มปริมาณผลงานมากขึ้น และยังใช้งาน RDBMS ได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

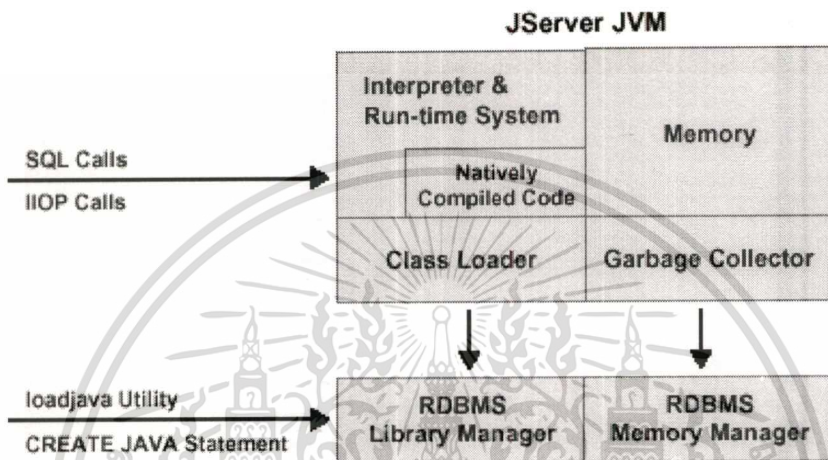
- อีกด้วย เราสามารถใช้ Java Integrated Development Environment (IDE) ทั่วไปในการสร้าง Stored Procedure ใหม่หรือนำมาจากระบบอื่น และยังคงใช้ได้จาก Interface มาตรฐานของ Java เช่น JDBC, COBRA และ EJB โดยการเขียนโปรแกรม Interface และ Development Tool เช่น SQLJ, OCI, Pro*C/C++ และ JDeveloper
- **ความสามารถในการขยายระบบ (Scalability)** การทำงานของแอปพลิเคชันนั้นแยกกัน และจะขึ้นต่อกันโดยอัตโนมัติเมื่อกำลังทำงานที่เกี่ยวข้องกัน การ Share หน่วยความจำของ Multi-Threaded Server (MTS) ทำให้ Oracle8i สามารถรองรับการใช้งานพร้อมกันของผู้ใช้มากกว่า 10,000 ราย และมี Net8 Connection Manager ทำ Multiplex การเชื่อมต่อด้วย
 - **การดูแลและบำรุงรักษา (Maintainability)** เมื่อมีการตรวจสอบความถูกต้องแล้ว สามารถใช้งาน Stored Procedure โดยแอปพลิเคชันได้อย่างมั่นใจ และถ้ามี Stored Procedure ใดเปลี่ยนแปลง ก็มีผลเฉพาะ Stored Procedure นั้น ไม่เกี่ยวกับแอปพลิเคชัน ทำให้ดูแลทำได้โดยง่าย และการดูแล Procedure ที่ Server เพียงจุดเดียวย่อมง่ายกว่าต้องดูแลที่ทุกๆ Client
 - **การทำงานระหว่าง PL/SQL และ Java Stored Procedure (Interoperability)** เป็นไปตาม Java Language Specification สามารถใช้ประโยชน์และข้อดีของภาษา Object-Oriented ทุกประการ PL/SQL สามารถเข้าถึงข้อมูลของ Oracle ได้เต็มที่ และ Procedure ที่เขียนโดย PL/SQL ก็ยังสามารถเขียนได้โดย Java โปรแกรมเมอร์ที่ถนัด PL/SQL หรือ Java ก็เลือกใช้ได้ตามต้องการ แอปพลิเคชัน Java สามารถเรียกใช้ PL/SQL Stored Procedure ได้โดยใช้ JDBC และแอปพลิเคชัน PL/SQL ก็ยังสามารถเรียกใช้ Java Stored Procedure ได้โดยตรง
 - **การทำสำเนา Stored Procedure (Replication)** สามารถก๊อปปี้ Stored Procedure จากฐานข้อมูล Oracle8i ไปใช้ที่อื่นได้
 - **ความปลอดภัย (Security)** Oracle JServer ใช้ Java 2 Security ป้องกัน Java Virtual Machine ทุก Class ถูกเก็บในฐานข้อมูลที่ปลอดภัย ผู้ใช้งานต้องมี Permission ที่เหมาะสมจึงจะใช้งานได้ (มีการ Grant EXECUTE Privilege ให้กับผู้ใช้งาน)

2.2 JServer JVM และส่วนประกอบ

JServer Java Virtual Machine (JVM) มีการทำงานตามสภาพแวดล้อมที่กำหนดไว้ใน Java 2-Compliant โดยจะทำงานใน Process Space และ Address Space เดียวกันกับ RDBMS Kernel มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้หน่วยความจำร่วมกันและสามารถเข้าถึงข้อมูลได้โดยตรง จึงเป็นการใช้หน่วยความจำที่เหมาะสมและเพิ่มความเร็วในการทำงานได้

นอกจากนี้ยังรองรับ Core Java Class Library เช่น java.lang, java.io, java.net, java.math และ java.util โดยแสดงในรูปที่ 2.3



รูปที่ 2.3 ส่วนประกอบหลักของ JServer JVM
(Tom. 1999 : 1-9)

JServer JVM ได้รวม Namespace มาตรฐานของ Java ไว้ใน RDBMS แล้ว ทำให้โปรแกรม Java สามารถใช้ Java Object ที่เก็บในฐานข้อมูล Oracle ได้

2.2.1 การเปรียบเทียบระหว่าง JServer JVM และ Client JVM

- **Method Main()** แอปพลิเคชัน Java ในฝั่ง Client เป็น Top-Level Method (main()) กำหนด Profile ของแอปพลิเคชัน ในขณะที่แอปพลิเคชัน Java ในฝั่ง Server ไม่มี Inner Loop จะถูกบังคับโดยแต่ละ Client อย่างอิสระ แต่ละ Client เมื่อเริ่ม Session จะไปเรียก Module จากฝั่ง Server ผ่าน Top-Level Entry Point และเมื่อจบ Session Server จะซ่อนการจัดการ Session และ Network และการ Share ทรัพยากรจากโปรแกรม Java
- **GUI Server** ไม่สามารถให้บริการด้าน GUI ได้ แต่ให้วิธีการที่จะใช้ GUI ได้ เช่น JServer JVM ไม่มี Basic GUI Component ที่มีอยู่ใน JDK's Abstract Windowing

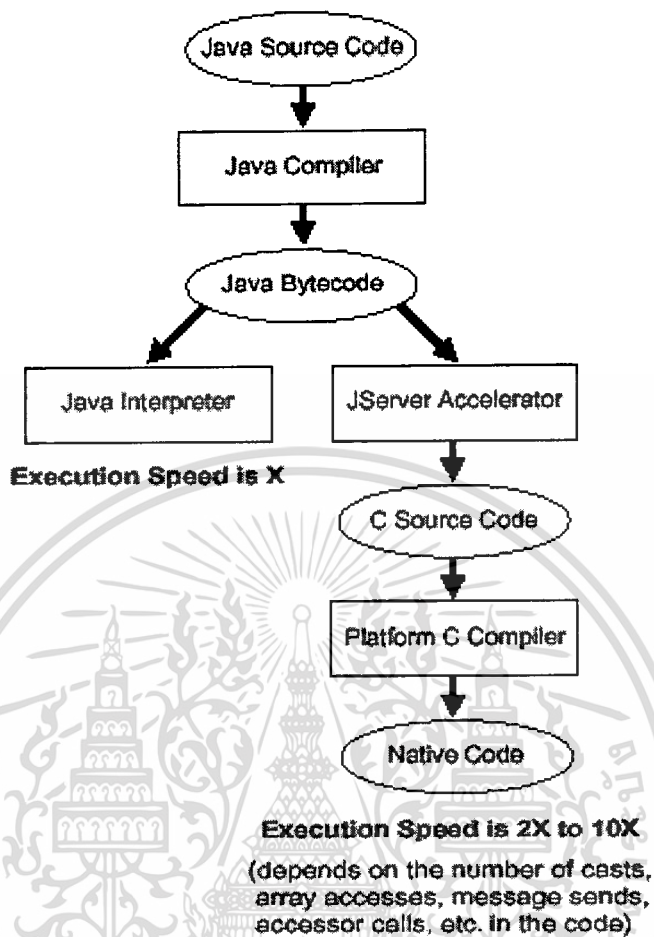
Toolkit (AWT) แต่มี AWT Java Class ทั้งหมดใน Server ดังนั้นโปรแกรมสามารถใช้ Function ของ AWT ได้ แต่จะต้องไม่ไปจัดการในส่วน GUI ของทาง Server

- **IDE** โปรแกรมเมอร์สามารถเขียนและทดสอบแอปพลิเคชัน โดย IDE ที่ชอบได้ แล้วนำมาใช้ทำงานภายใน RDBMS และเนื่องจาก Binary File ของ Java สามารถเข้ากันได้กับ IDE ทั่วไป ดังนั้นเมื่อนำ Java Class มาไว้ที่ Server จึงไม่จำเป็นต้องนำ Java Source File มาใส่ไว้ในฐานข้อมูลด้วย แต่กลับสามารถจัดการดูแลแอปพลิเคชัน Java ที่อยู่ใน Server โดยใช้การพัฒนาของ IDE ที่ฝั่ง Client
- **Multi-Threading** โปรแกรม Java Multi-Threaded สามารถทำงานบน Oracle8i ได้ทันทีโดยไม่จำเป็นต้องแก้ไข แต่ใน Server, Java Thread ไม่ได้เพิ่มการทำงานพร้อมกันและความเร็วซึ่งมีจะผลเฉพาะกรณีที่เป็น MTS Mode เท่านั้นและขึ้นอยู่กับจำนวนโปรเซสเซอร์ที่ถูกใช้โดย RDBMS และการปรับแต่งต่างๆ

2.2.2 ส่วนประกอบหลักของ JServer JVM

- **Library Manager** ในการจัดเก็บ Java Class ลงในฐานข้อมูล จะต้องใช้คำสั่ง loadjava ซึ่งจะไปใช้คำสั่งในส่วนของ SQL เป็นคำสั่ง CREATE JAVA ทำงานให้ต่ออีกระดับหนึ่ง เมื่อมีการใช้คำสั่ง CREATE JAVE {SOURCE | CLASS | RESOURCE} library manager จะทำการ load Java source, class หรือ resource file เข้าไปในฐานข้อมูล
- **Memory Manager** การจัดการพื้นที่ในการทำงานโดยอัตโนมัติเป็นจุดเด่นประการหนึ่งของ Java โดยเฉพาะระบบ Run-Time ที่ใช้ Garbage Collection (การคืนหน่วยความจำจาก Object ที่ไม่ใช้งานแล้ว) โดยใช้เทคนิคการปรับแต่งช่วงอายุของ Object เพื่อทำการเก็บ Object ที่ยังน่าจะใช้งานต่อไว้ในพื้นที่หน่วยความจำที่เหมาะสม
- **Compiler** JServer JVM ได้มีการรวม Java 2 Compiler มาตรฐาน (JDK 1.2)เอาไว้ เมื่อมีการสั่ง CREATE JAVA SOURCE จะมีการแปล Java Source File ให้อยู่ในรูปแบบ Architecture-Neutral ที่เป็นคำสั่งขนาด 1 ไบต์ ที่เรียกว่า Bytecode ประกอบด้วย Opcode และ Operand ซึ่งเป็นไปตามมาตรฐานของ Java และถูกใช้งานโดย Interpreter เวลาถูกใช้งาน
- **Interpreter** ใช้ในการแปลคำสั่งในโปรแกรม Java ให้ทำงาน ถ้าต้องการความเร็วเพิ่มขึ้น จะต้องให้ Interpreter ทำงานบน Multi-Thread Server ที่จะจัดการ Session และ Schedule ที่จะสั่งให้โปรแกรม Java ทำงาน

- **Class Loader** ในการตอบรับการขอบริการจากระบบ Run-Time, Java Class Loader จะกำหนดตำแหน่ง, การ Load และเริ่มทำงาน Java Class ที่อยู่ในฐานข้อมูล จะมีการอ่าน Class และสร้างโครงสร้างข้อมูล ข้อมูลที่ไม่มีการเปลี่ยนแปลงหรือมีขนาดใหญ่ จะถูก Load ครั้งเดียว ทำให้ใช้หน่วยความจำน้อยต่อการทำงานภายใน Session และจะมีการติดต่อข้อมูลอ้างอิงภายนอกเมื่อจำเป็น และ Compile Java Class File ที่ต้อง Compile ใหม่ให้โดยอัตโนมัติ
- **Verifier** ป้องกันการใช้ Java Class File ที่ไม่ถูกต้องโดยไม่ตั้งใจ ซึ่งจะทำให้แอปพลิเคชันและข้อมูลผิดพลาดได้
- **JDBC Internal Driver** ผัง Server JDBC คือชุดของ Java Class ที่สามารถให้โปรแกรมของ Vendor เข้าถึงฐานข้อมูลได้ มีการกำหนดขึ้นโดย Sun Microsystem ต่อจาก ODBC (Open Database Connectivity) และ X/Open SQL CLI (Call Level Interface) การใช้ Low-Level Entry Point ทำให้ JDBC สามารถ Run ใน RDBMS ได้โดยตรง จึงมีความเร็วสูงในการที่เข้าถึงข้อมูลจาก Java Stored Procedure
- **SQLJ Translator** ผัง Server SQLJ ทำให้สามารถรวมคำสั่ง SQL กับโปรแกรม Java ได้ มีความกระชับกว่า JDBC และน่าเชื่อถือกว่าในด้านการวิเคราะห์ทางสถิติและการตรวจสอบชนิดของข้อมูลเพราะ SQLJ Processor เป็น โปรแกรมที่สร้างด้วย Java SQLJ สามารถ Run ใน RDBMS โดยตรงได้ดีกว่า และ SQLJ ยังสามารถรวม Query, DML, DDL, คำสั่งควบคุม Transaction และเรียก Stored Procedure ได้
- **JServer Accelerator** เป็นตัวคอมไพล์ Native Code ที่ทำให้การทำงานของโปรแกรม Java เร็วขึ้น โดยการแยก Overhead ของ Interpreter ออกมา แล้วแปลเป็นภาษา C ซึ่งจะถูกใช้งานโดยคอมไพเลอร์ของภาษา C ใน Platform นั้น ภายใน Shared Library ที่ JServer สามารถ Load ได้ แม้ว่าโปรแกรม Java ที่ Load ไว้ในฐานข้อมูลจะเป็นแบบ Interpret แต่ก็ยังใช้ Native Compiler ได้ Core JDK Class และ Oracle Class ก็ใช้ Native Compiler ตามรูปที่ 2.4 แสดง Code ที่ใช้ Native Compiler สามารถทำงานได้เร็วกว่า Code แบบ Interpret สูงสุดถึง 10 เท่า ดังนั้นถ้าโปรแกรมใช้ Native Code มากขึ้น ก็สามารถทำงานได้เร็วขึ้น



รูปที่ 2.4 การเปรียบเทียบระหว่าง Interpreter และ JServer Accelerator
(Tom. 1999 : 1-14)

2.3 ขั้นตอนในการพัฒนา Stored Procedure

ในการสร้าง Java stored procedure มีขั้นตอนดังที่จะแสดงเป็นตัวอย่างต่อไปนี้

ขั้นที่ 1: ทำการสร้าง Java Class

ใช้ IDE ใดๆ สร้าง Class หรือนำ Class ที่มีอยู่แล้วมาใช้ IDE ที่ใช้ได้ ได้แก่ Visual Cafe ของ Symantec, JDeveloper ของ Oracle และ JBuilder ของ Borland

ในตัวอย่างด้านล่างเป็นการสร้าง Class Oscar มี Method เดียวชื่อว่า Quote() ซึ่งจะคืน Quotation จาก Oscar Wilde

```
public class Oscar {
    // return a quotation from Oscar Wilde
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public static String quote() {
    return "I can resist everything except temptation.";
}
}

```

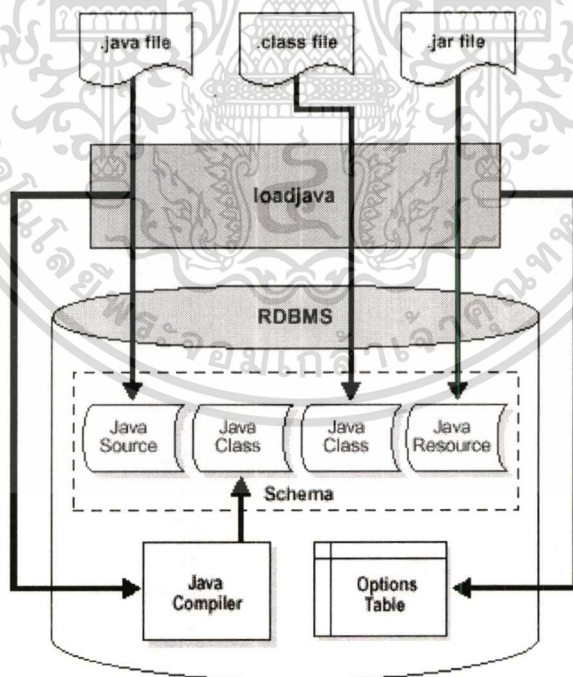
จากนั้นให้ทำการใช้ JDK Java Compiler ของ Sun Microsystems เพื่อทำการ Compile ไฟล์ Java ให้เป็น ไฟล์ Class Oscar บน Client(ขณะ Compile จะต้องระบุชื่อไฟล์เป็นตัวอักษรใหญ่เล็ก ให้ถูกต้อง)

```
javac Oscar.java
```

ผลลัพธ์ได้ Java Binary File (Oscar.class)

ขั้นที่ 2: Load และ Resolve Java Class

ก่อนที่เราจะสามารถเรียกใช้ Java Stored Procedure เราต้องมีการ Load Class ที่เป็นไฟล์ เข้าสู่ฐานข้อมูล Oracle ให้เป็น Schema Object ก่อน โดยใช้ loadjava ดังแสดงในรูปที่ 2.5 loadjava สามารถเรียกใช้ Java Compiler ของ JVM ซึ่งสามารถ Compile ไฟล์ Java ให้เป็น Class ได้



รูปที่ 2.5 การโหลดไฟล์จาวาเข้าสู่ฐานข้อมูล

(Tom. 1999 : 2-2)

สำหรับในแต่ละ Class เราสามารถ Load ไฟล์ที่เป็นต้นฉบับหรือที่เป็นไฟล์ Class ก็ได้ แต่ไม่สามารถ Load พร้อมกันได้ทั้ง 2 ไฟล์ ถ้ามีการสร้าง Class เรียบร้อยทางด้าน Client เราสามารถใช้ loadjava ในการ Upload ไปยังฐานข้อมูล หรือ Upload ไฟล์ต้นฉบับไปยังฐานข้อมูล และให้ JServer JVM ทำการ Compile

Class ทั้งหมดใน Java จะต้อง Load ก่อนที่จะมีการอ้างอิงจากภายนอกจะมีการ Resolve โดยการ Resolve คือการตรวจสอบว่ามี Class ที่มีการอ้างอิงหรือไม่ หากยังไม่มีการ Resolve มาก่อน Class ที่ Load จะมีสถานะเป็น Invalid

ที่ Command Line ให้ระบุชื่อไฟล์ต้นฉบับ, Class และ Resource ในการ Load เข้าสู่ฐานข้อมูลดัง Syntax ด้านล่าง

```
loadjava {-user | -u} username/password[@database]
[-option_name -option_name ...] file_name file_name ...
```

ที่ Option_name สามารถกำหนดค่าได้ต่างๆดังนี้

```
{ {a | andresolve}
| debug
| {d | definer}
| {e | encoding} encoding_scheme_name
| {f | force}
| {g | grant} username[,username]...
| {o | oci8}
| {h | help}
| oracleresolver
| noverify
| {r | resolve}
| {R | resolver} "resolver_spec"
| {S | schema} schema_name
| {s | synonym}
| {t | thin}
| {v | verbose} }
```

ในรายการของ Option จะต้องแบ่งด้วยการเว้นวรรค

```
-force, -resolve, -thin // ผิด
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-force -resolve -thin // ถูก
 แต่ในรายการของผู้ใช้ จะแบ่งด้วยเครื่องหมาย Commas
 SCOTT, UTIL, BLAKE // ผิด
 SCOTT,UTIL,BLAKE // ถูก

ตารางที่ 2.1 รายละเอียดของ Option ของ loadjava

Option	คำอธิบาย
andresolve	ทำการ Compile ไฟล์ต้นฉบับและทำการ Resolve แต่ละ Class ที่ทำการ Load
debug	สร้างและแสดงข้อมูลการ Debug
definer	ระบุสิทธิ์ในการเรียกใช้ Method ของ Class
encoding	กำหนดข้อมูลในการเข้ารหัสของไฟล์ต้นฉบับ
force	ระบุให้ทำการ Load Class โดยไม่สนใจว่าเคยทำการ Load มาก่อน
grant	ให้สิทธิ์ในการ EXECUTE Class ที่ทำการ Load กับ User ที่กำหนดเอาไว้
help	แสดงหน้าจอช่วยเหลือของ loadjava
oci8	จัดการให้ loadjava ติดต่อสื่อสารกับฐานข้อมูล โดยใช้ JDBC OCI Driver
noverify	กำหนดไม่ให้ Verifier ขณะ Load Class
oracleresolver	ทำการเชื่อมโยง Class ที่สร้างกับ Resolver Spec เดิมที่มี
resolve	ทำการ Resolve Class
resolver	ทำการเชื่อมโยง Class ที่สร้างกับ Resolver Spec ที่กำหนด
schema	Load Class ไปยัง Schema ที่กำหนด ค่า Default คือ Schema ที่ทำการ Logon
synonym	สร้าง Synonym สำหรับ Class ที่ Load ทำให้สามารถเรียกใช้ Class ได้จาก Schema อื่น
thin	จัดการให้ loadjava ติดต่อสื่อสารกับฐานข้อมูล โดยใช้ JDBC Thin Driver จากทางด้าน Client
verbose	ให้ loadjava แสดงข้อความขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับพารามิเตอร์ -user จะเป็นการระบุชื่อผู้ใช้,รหัสผ่าน,และฐานข้อมูลตามรูปแบบด้านล่าง

```
username/password[@database]
```

ตัวอย่างด้านล่างเป็นการใช้ loadjava ในการติดต่อไปยังฐานข้อมูลโดยใช้ thin JDBC

Driver ทำการ Load 2 Class และทำการ Resolve

```
loadjava -user scott/tiger@dbhost:5521:orcl -thin -resolve Agent.class
```

```
Alerter.class
```

ตัวอย่างต่อไปเป็นการใช้ loadjava ติดต่อฐานข้อมูลด้วยฐานข้อมูลที่เป็น Default โดยใช้ JDBC OCI Driver ทำการ Load จากไฟล์ JAR ไปยัง Schema BLAKE และทำการ Resolve

```
loadjava -user scott/tiger -resolve -schema BLAKE serverObjs.jar
```

ตัวอย่างสุดท้ายเป็นการใช้ loadjava เพิ่ม NILES และ FORD ลงในรายการผู้ใช้ที่สามารถเรียก Manager.class ทำงานได้

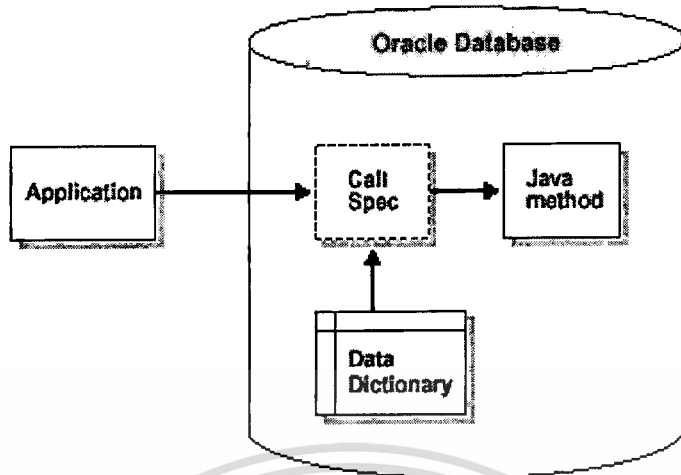
```
loadjava -user scott/tiger@localhost:5521:orcl -thin -grant NILES,FORD
```

```
Manager.class
```

ขั้นที่ 3: ประกาศ Java Class

เมื่อทำการ Load Class เรียบร้อย ขั้นตอนต่อไปคือการเขียน Procedure หรือ Function Call Spec โดยใช้ คำสั่ง CREATE FUNCTION หรือ CREATE PROCEDURE ของ SQL

เราจะใช้ Function กับ Method ที่มีการส่งค่ากลับคืน และใช้ Procedure กับ Method ที่ไม่มีการส่งค่ากลับ(ส่งกลับเป็น void) ภายใน Function หรือ Procedure จะมีอนุประโยค LANGUAGE JAVA อยู่ ในอนุประโยคจะบันทึกชื่อ Method, ชนิดข้อมูลของพารามิเตอร์, และชนิดของค่าที่ส่งกลับ



รูปที่ 2.6 การเรียกใช้ Java Method

(Tom. 1999 : 3-2)

ในรูปที่ 2.6 จะแสดงให้เห็นการเรียกใช้ Java Method ผ่าน Call Spec คือการอ้างอิงชื่อของ Call Spec ระบบจะมองหาการนิยาม Call Spec ใน Data Dictionary จากนั้นจะทำการเรียกใช้ Java Method ที่กำหนดไว้ใน Call Spec โดยที่ Call Spec และ Java Method จะต้องอยู่ใน Schema เดียวกัน(ยกเว้น Java Method ที่มีการสร้าง PUBLIC Synonym)

เราสามารถนิยาม Call Spec โดยใช้ Syntax ดังนี้

```

CREATE [OR REPLACE] { PROCEDURE procedure_name [(param[, param]...)]
| FUNCTION function_name [(param[, param]...)] RETURN sql_type}
[AUTHID {DEFINER | CURRENT_USER}]
[PARALLEL_ENABLE]
[DETERMINISTIC]
{IS | AS} LANGUAGE JAVA
NAME 'method_fullname (java_type_fullname[, java_type_fullname]...)
[return java_type_fullname]';
  
```

โดยที่ param ประกอบด้วย Syntax

```
parameter_name [IN | OUT | IN OUT] sql_type
```

อนุประโยค AUTHID ระบุว่า Stored Procedure จะถูกเรียกใช้โดยสิทธิ์ของผู้นิยามหรือของผู้เรียก

PARALLEL_ENABLE ใช้ประกาศว่า Stored Function สามารถใช้ใน Session ของ Slave กรณีมีการทำงานแบบ Parallel DML

DETERMINISTIC ช่วยในการหลีกเลี่ยงการเรียกใช้ Function ซ้ำซ้อน ถ้า Stored Function ถูกเรียกเหมือน Stored Function ก่อนหน้านี้ด้วยพารามิเตอร์ที่เหมือนกัน ตัว Optimizer จะนำผลลัพธ์จากการทำงานก่อนหน้านี้ให้

อนุประโยค NAME จะระบุ Java Method ที่เรียกใช้ ชื่อและพารามิเตอร์ Call Spec จะถูก Map โดยตำแหน่ง หาก Java Method นั้นไม่มีพารามิเตอร์ ให้ใส่เป็นรายการพารามิเตอร์ว่างๆของ Java Method แต่สำหรับ Procedure กับ Function ที่สร้างไม่ต้องระบุ

ในตัวอย่างด้านล่างใช้ SQL*Plus ติดต่อกับฐานข้อมูลและสร้าง Call Spec ให้กับ Method quote();

```
SQL> connect scott/tiger
```

```
SQL> CREATE FUNCTION oscar_quote RETURN VARCHAR2
```

```
2 AS LANGUAGE JAVA
```

```
3 NAME 'Oscar.quote() return java.lang.String'
```

ขั้นที่ 4 : เรียกใช้ Stored Procedure

หลังจาก load และ ประกาศ Java stored procedure แล้ว ก็จะสามารถเรียกใช้ได้ในรูปแบบ context ต่างๆดังนี้

การเรียกใช้ Java จาก the Top Level

คำสั่ง CALL เป็นคำสั่ง SQL ที่ทำให้เรียกใช้ Method ของ Java ที่ประกาศไว้ใน PL/SQL Package หรือใน SQL Object Type

ใน SQL*Plus สามารถใช้คำสั่ง CALL ได้ โดยมี syntax ดังนี้

```
CALL [schema_name.][{package_name | object_type_name}][@dblink_name]
```

```
{ procedure_name ([param[, param]...])
```

```
| function_name ([param[, param]...]) INTO :host_variable};
```

โดยที่ param นั้นย่อมาจาก syntax ดังต่อไปนี้

```
{literal | :host_variable}
```

Host Variables (Variables ที่มีการประกาศในสภาพแวดล้อมที่ PL/SQL ทำงานอยู่ แต่ไม่ใช่ตัวแปรที่ PL/SQL ประกาศ) ต้องใส่ : ไว้ข้างหน้าด้วย ตัวอย่างต่อไปนี้แสดงว่าตัวแปรของ Host ไม่สามารถปรากฏขึ้นสองครั้งในคำสั่ง CALL และ Subprogram ที่ไม่มี Parameter ก็เรียกใช้โดยไม่มี Parameter เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL swap(:x, :x); -- ไม่ถูกต้อง, Host Variables ซ้ำซ้อน

CALL balance() INTO :current_balance; -- () required

บน Server การแสดงผลที่เป็น Default Output คือ Trace File ไม่ใช่หน้าจอ เช่น System.out System.err ก็อยู่ในรูปของ Current Trace Files ในการ Redirect Output ไปเป็น SQL*Plus Text Buffer ทำได้โดยใช้ Procedure ชื่อ set_output() ที่อยู่ใน Package DBMS_JAVA ดังตัวอย่าง

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> CALL dbms_java.set_output(2000);
```

ขนาดของ Buffer ที่น้อยที่สุดตาม Default คือ 2,000 ไบต์ และขนาดของ Buffer ที่มากที่สุดตาม Default คือ 1,000,000 ไบต์ ในตัวอย่างข้างล่าง ขนาดของ Buffer ถูกตั้งให้มีขนาดใหญ่ขึ้นเป็น 5,000 ไบต์

```
SQL> SET SERVEROUTPUT ON SIZE 5000
```

```
SQL> CALL dbms_java.set_output(5000);
```

Output จะถูกพิมพ์ออกมาเมื่อ Stored Procedure ทำงานเสร็จสิ้นแล้ว

ตัวอย่างที่ 1 ในตัวอย่างข้างล่าง Method main มีการรับชื่อของ Table ในฐานข้อมูล (เช่น 'emp') และเงื่อนไข WHERE clause (เช่น 'sal > 1500') ถ้าเราไม่กำหนดเงื่อนไขเหล่านี้แล้วมีการใช้ Method deletes ตั้งลบ Row ทั้งหมดจะถูกลบทุก Row แต่ถ้าใช้เงื่อนไขด้วยจะลบเฉพาะบาง Row ที่เป็นไปตามเงื่อนไข

```
import java.sql.*;
```

```
import oracle.jdbc.driver.*;
```

```
public class Deleter {
```

```
    public static void main (String[] args) throws SQLException {
```

```
        Connection conn =
```

```
            DriverManager.getConnection("jdbc:default:connection:");
```

```
        String sql = "DELETE FROM " + args[0];
```

```
        if (args.length > 1) sql += " WHERE " + args[1];
```

```
        try {
```

```
            Statement stmt = conn.createStatement();
```

```
            stmt.executeUpdate(sql);
```

```
            stmt.close();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    } catch (SQLException e) {System.err.println(e.getMessage());}
}
}

```

Method main สามารถใช้ Argument ได้ทั้ง 1 และ 2 ตัว DEFAULT Clause ที่ถูกใช้จะแปรเปลี่ยนตามจำนวน Argument แล้วส่งไปยัง PL/SQL Subprogram แต่อย่างไรก็ตาม Clause นั้นใช้ไม่ได้ใน Call Spec ดังนั้นจึงต้องใช้วิธี Overload 2 Packaged Procedure (ไม่สามารถ Overload Top-Level Procedure ได้) ดังตัวอย่าง

```

CREATE OR REPLACE PACKAGE pkg AS
  PROCEDURE delete_rows (table_name VARCHAR2);
  PROCEDURE delete_rows (table_name VARCHAR2, condition VARCHAR2);
END;

```

```

CREATE OR REPLACE PACKAGE BODY pkg AS
  PROCEDURE delete_rows (table_name VARCHAR2)
  AS LANGUAGE JAVA
  NAME 'Deleter.main(java.lang.String[])';
  PROCEDURE delete_rows (table_name VARCHAR2, condition VARCHAR2)
  AS LANGUAGE JAVA
  NAME 'Deleter.main(java.lang.String[])';
END;

```

ขณะนี้ก็สามารถที่จะเรียก procedure delete_rows

```
SQL> CALL pkg.delete_rows('emp', 'sal > 1500');
```

Call completed.

```
SQL> SELECT ename, sal FROM emp;
```

ENAME	SAL
SMITH	800
WARD	1250
MARTIN	1250
TURNER	1500

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ADAMS          1100
JAMES          950
MILLER         1300
```

7 rows selected.

ตัวอย่างที่ 2 สมมุติว่า Java Class ที่สามารถทำงานได้ ถูกจัดเก็บไว้ในฐานข้อมูล Oracle เรียบร้อยแล้ว

```
public class Fibonacci {
    public static int fib (int n) {
        if (n == 1 || n == 2)
            return 1;
        else
            return fib(n - 1) + fib(n - 2);
    }
}
```

Class Fibonacci มี Method เดียว ชื่อว่า fib ซึ่งจะคืนค่าลำดับที่ของตัวเลข Fibonacci ซึ่งลำดับเป็น 1, 1, 2, 3, 5, 8, 13, 21, ... ซึ่งใช้ครั้งแรกเป็นแบบการเติบโตของฝูงกระต่าย เป็นแบบ Recursive เลขลำดับแต่ละตัวเกิดขึ้นทุกวินาที เป็นยอดรวมของ 2 ตัวล่าสุด เพราะว่า Method fib คืนค่าผลลัพธ์ออกมา ดังนั้นจึงประกาศเป็น Function

```
CREATE OR REPLACE FUNCTION fib (n NUMBER) RETURN NUMBER
AS LANGUAGE JAVA
NAME 'Fibonacci.fib(int) return int';
```

ต่อไปก็กำหนด 2 ตัวแปร Host ของ SQL*Plus แล้วให้ค่าเริ่มต้นกับตัวแปรตัวหนึ่ง

```
SQL> VARIABLE n NUMBER
SQL> VARIABLE f NUMBER
SQL> EXECUTE :n := 7;
```

PL/SQL procedure successfully completed.

หลังจากนั้นก็พร้อมที่จะเรียก Function fib และต้องระวังไว้เสมอว่า CALL Statement, Host Variables ต้องนำหน้าด้วยเครื่องหมาย :

```
SQL> CALL fib(:n) INTO :f;
Call completed.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SQL> PRINT f
```

```
F
```

```
-----
```

```
13
```

การเรียกใช้ Java จาก Database Triggers

Database Trigger เป็น Stored Program ที่รวมไปกับ Table หรือ View ที่กำหนดเจาะจง Oracle จะสั่งให้ Trigger ทำงานอัตโนมัติเมื่อมีการใช้คำสั่ง DML ปฏิบัติงานแล้วมีผลต่อ Table หรือ View นั้น

Trigger มีอยู่ 3 ส่วนคือ 1. Triggering Event (DML Operation), 2. Optional Trigger Constraint และ 3. Trigger Action เมื่อ Event นั้นเกิดขึ้น Trigger จะเริ่มทำงาน และ PL/SQL Block หรือ CALL Statement จะทำงานตาม Action Statement Trigger จะทำงานครั้งเดียวก่อนหรือหลัง Event นั้น ส่วน Row Trigger จะทำงานกับแต่ละ Row ที่มีผลจาก Event นั้น

ภายใน Database Trigger สามารถอ้างอิงค่าใหม่และค่าเดิมของ Row ที่กำลังเปลี่ยนแปลงได้ ใช้ชื่อคู่ที่ว่า New และ Old ใน Trigger-Action Block หรือ คำสั่ง CALL ชื่อของ Column ต้องขึ้นด้วย :New หรือ :Old

การสร้าง Database Trigger ต้องใช้คำสั่ง SQL CREATE TRIGGER

ตัวอย่างที่ 1 สมมุติว่าต้องการสร้าง Database Trigger ซึ่งใช้ Java Class ต่อไปนี้เพื่อบันทึก Out-Of-Range Salary ที่เพิ่มขึ้น

```
import java.sql.*;
import java.io.*;
import oracle.jdbc.driver.*;

public class DBTrigger {

    public static void logSal (int empID, float oldSal, float newSal)
        throws SQLException {

        Connection conn =
            DriverManager.getConnection("jdbc:default:connection:");
        String sql = "INSERT INTO sal_audit VALUES (?, ?, ?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setInt(1, empID);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pstmt.setFloat(2, oldSal);
pstmt.setFloat(3, newSal);
pstmt.executeUpdate();
pstmt.close();
} catch (SQLException e) {System.err.println(e.getMessage());}
}
}

```

Class DBTrigger มี Method เดียวทำการ Insert Row สู่ Table Sal_Audit ของฐานข้อมูล เพราะว่า logSal เป็น Method ที่ไม่ต้องส่งค่ากลับ จึงประกาศเป็น Procedure

```

CREATE OR REPLACE PROCEDURE log_sal (
emp_id NUMBER, old_sal NUMBER, new_sal NUMBER)
AS LANGUAGE JAVA
NAME 'DBTrigger.logSal(int, float, float)';

```

ต่อไปก็สร้าง Table Sal_Audit ดังต่อไปนี้

```

CREATE TABLE sal_audit (
empno NUMBER,
oldsal NUMBER,
newsal NUMBER);

```

และในขั้นสุดท้าย ทำการสร้าง Database Trigger ซึ่งจะทำงานเมื่อ Salary เพิ่มขึ้นมากกว่า 20%

```

CREATE OR REPLACE TRIGGER sal_trig
AFTER UPDATE OF sal ON emp
FOR EACH ROW
WHEN (new.sal > 1.2 * old.sal)
CALL log_sal(:new.empno, :old.sal, :new.sal);

```

เมื่อใช้คำสั่ง UPDATE ตามข้างล่าง จะ update ทุกๆ rows ใน Table Emp สำหรับแต่ละ Row ที่ตามตามเงื่อนไข WHERE Clause ของ Trigger โดย Trigger จะทำงานและ Java Method จะ Insert Row ใน Table Sal_Audit

```
SQL> UPDATE emp SET sal = sal + 300;
```

```
SQL> SELECT * FROM sal_audit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EMPNO	OLDSAL	NEWSAL
-----	-----	-----
7369	800	1100
7521	1250	1550
7654	1250	1550
7876	1100	1400
7900	950	1250
7934	1300	1600

6 rows selected.

ตัวอย่างที่ 2 สมมติว่าจะสร้าง Trigger ซึ่ง Insert Rows ใน View ดังต่อไปนี้

```
CREATE VIEW emps AS
SELECT empno, ename, 'Sales' AS dname FROM sales
UNION ALL
SELECT empno, ename, 'Marketing' AS dname FROM mktg;
```

สร้าง Table Sales และ Mktg ในฐานะข้อมูล

```
CREATE TABLE sales (empno NUMBER(4), ename VARCHAR2(10));
CREATE TABLE mktg (empno NUMBER(4), ename VARCHAR2(10));
```

จะต้องสร้าง INSTEAD OF Trigger แทนเพราะว่า Row ไม่สามารถถูก Insert สู่วิวที่ใช้

Operator เช่น UNION ALL ได้ Trigger จะ Insert Row สู่วิว Base Tables.

ตัวอย่างที่แล้ว ได้ทำการ Add Java Method สู่วิว Class DBTrigger

```
public static void addEmp (
    int empNo, String empName, String deptName)
throws SQLException {
    Connection conn =
        DriverManager.getConnection("jdbc:default:connection:");
    String tabName = (deptName.equals("Sales") ? "sales" : "mktg");
    String sql = "INSERT INTO " + tabName + " VALUES (?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, empNo);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pstmt.setString(2, empName);
pstmt.executeUpdate();
pstmt.close();
} catch (SQLException e) {System.err.println(e.getMessage());}
}

```

Method addEmp ทำการ Insert Row ลงสู่ Table Sales หรือ Mktg ขึ้นอยู่กับค่าของ Parameter deptName มีการเขียน Call Spec ให้กับ Method นี้ดังนี้

```

CREATE OR REPLACE PROCEDURE add_emp (
emp_no NUMBER, emp_name VARCHAR2, dept_name VARCHAR2)
AS LANGUAGE JAVA
NAME 'DBTrigger.addEmp(int, java.lang.String, java.lang.String)';

```

ต่อไปให้ทำการสร้าง INSTEAD OF Trigger

```

CREATE OR REPLACE TRIGGER emps_trig
INSTEAD OF INSERT ON emps
FOR EACH ROW
CALL add_emp(:new.empno, :new.ename, :new.dname);

```

เมื่อสั่งคำสั่ง INSERT ทำงานแต่ละครั้งตามข้างล่าง Trigger จะทำงานให้ Java Method Insert Row ลงสู่ Base Table ที่เหมาะสม

```

SQL> INSERT INTO emps VALUES (8001, 'Chand', 'Sales');
SQL> INSERT INTO emps VALUES (8002, 'Van Horn', 'Sales');
SQL> INSERT INTO emps VALUES (8003, 'Waters', 'Sales');
SQL> INSERT INTO emps VALUES (8004, 'Bellock', 'Marketing');
SQL> INSERT INTO emps VALUES (8005, 'Perez', 'Marketing');
SQL> INSERT INTO emps VALUES (8006, 'Foucault', 'Marketing');
SQL> SELECT * FROM sales;

```

EMPNO	ENAME
8001	Chand
8002	Van Horn
8003	Waters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SQL> SELECT * FROM mktg;
```

EMPNO	ENAME
-----	-----
8004	Bellock
8005	Perez
8006	Foucault

```
SQL> SELECT * FROM emps;
```

EMPNO	ENAME	DNAME
-----	-----	-----
8001	Chand	Sales
8002	Van Horn	Sales
8003	Waters	Sales
8004	Bellock	Marketing
8005	Perez	Marketing
8006	Foucault	Marketing

การเรียกใช้ Java จาก SQL DML

ถ้าประกาศ Java Method เป็น Function จะสามารถเรียกใช้ได้เลยจากคำสั่ง SQL SELECT, INSERT, UPDATE และ DELETE เช่นสมมุติว่า Java Class ที่สามารถทำงานได้ ซึ่งจัดเก็บไว้ในฐานข้อมูล Oracle

```
public class Formatter {
    public static String formatEmp (String empName, String jobTitle) {
        empName = empName.substring(0,1).toUpperCase() +
            empName.substring(1).toLowerCase();
        jobTitle = jobTitle.toLowerCase();
        if (jobTitle.equals("analyst"))
            return (new String(empName + " is an exempt analyst"));
        else
            return (new String(empName + " is a non-exempt " + jobTitle));
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้าง Class Formatter มี Method เดียวชื่อว่า formatEmp โดยจะคืนข้อมูล String ที่จัดรูปแบบแล้ว ประกอบด้วยชื่อของ Staffer และ Job Status เริ่มด้วยสร้าง Call Spec ให้กับ Method นี้ดังนี้

```
CREATE OR REPLACE FUNCTION format_emp (ename VARCHAR2, job
VARCHAR2)
RETURN VARCHAR2
AS LANGUAGE JAVA
NAME 'Formatter.formatEmp (java.lang.String, java.lang.String)
return java.lang.String';
```

แล้วเรียกใช้ Function format_emp เพื่อจัดรูปแบบรายละเอียดของกลุ่มของ Employee

```
SQL> SELECT format_emp(ename, job) AS "Employees" FROM emp
2 WHERE job NOT IN ('MANAGER', 'PRESIDENT') ORDER BY ename;
```

Employees

Adams is a non-exempt clerk

Allen is a non-exempt salesman

Ford is an exempt analyst

James is a non-exempt clerk

Martin is a non-exempt salesman

Miller is a non-exempt clerk

Scott is an exempt analyst

Smith is a non-exempt clerk

Turner is a non-exempt salesman

Ward is a non-exempt salesman

เพื่อให้สามารถถูกเรียกใช้ได้จากคำสั่ง SQL DML Method ของ Java ต้องยอมปฏิบัติตามกฎ Purity ดังต่อไปนี้ ซึ่งมีความหมายในการควบคุมผลกระทบ

- เมื่อเรียกใช้จากคำสั่ง SELECT หรือ Parallelized INSERT, UPDATE หรือ DELETE Method จะไม่สามารถเปลี่ยนแปลง Table ใดๆของฐานข้อมูลได้

- เมื่อเรียกใช้จากคำสั่ง INSERT, UPDATE หรือ DELETE Method จะไม่สามารถ Query หรือเปลี่ยนแปลง Table ใดๆของฐานข้อมูล ที่ถูกเปลี่ยนแปลงไปแล้วจากคำสั่งนั้น
- เมื่อเรียกใช้จากคำสั่ง SELECT, INSERT, UPDATE หรือ DELETE Method จะไม่สามารถทำงานคำสั่ง SQL Transaction Control (เช่น COMMIT), คำสั่ง Session Control Statement (เช่น SET ROLE) หรือคำสั่ง System Control (เช่น ALTER SYSTEM) ได้ และยังไม่สามารถทำงานคำสั่ง DDL (เช่น CREATE) ด้วยเพราะว่ามันจะปฏิบัติตาม Automatic Commit

ถ้าหากคำสั่ง SQL ที่อยู่ใน Method ไม่เป็นไปตามกฎนี้ จะเกิด Error ขึ้นในระหว่าง Run Time

การเรียกใช้ Java จาก PL/SQL

เราสามารถเรียกใช้ Java Stored Procedure จาก PL/SQL Block ใดๆ, Subprogram ใดๆ หรือ Package ใดๆ ตัวอย่างเช่น สมมุติ Java Class ที่สามารถทำงานได้แล้วนี้ ถูกจัดเก็บไว้ในฐานข้อมูลของ Oracle

```
import java.sql.*;
import oracle.jdbc.driver.*;
public class Adjuster {
    public static void raiseSalary (int empNo, float percent)
    throws SQLException {
        Connection conn =
            DriverManager.getConnection("jdbc:default:connection");
        String sql = "UPDATE emp SET sal = sal * ? WHERE empno = ?";
        try {
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setFloat(1, (1 + percent / 100));
            pstmt.setInt(2, empNo);
            pstmt.executeUpdate();
            pstmt.close();
        } catch (SQLException e) {System.err.println(e.getMessage());}
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

สร้าง Class Adjuster มี Method เดียว ทำงานขึ้นเงินเดือนให้กับ Employee คนหนึ่งจาก
เปอร์เซ็นต์ที่รับค่ามา

เพราะว่า raiseSalary เป็น Method ที่ไม่ต้องส่งค่ากลับ จึงประกาศเป็น Procedure ดังต่อไปนี้

```
CREATE OR REPLACE PROCEDURE raise_salary (empno NUMBER, pct
NUMBER)
AS LANGUAGE JAVA
NAME 'Adjuster.raiseSalary(int, float)';
```

ตัวอย่างต่อไปนี้ สามารถเรียกใช้ Procedure raise_salary จาก Anonymous PL/SQL Block

```
DECLARE
emp_id NUMBER;
percent NUMBER;
BEGIN
-- get values for emp_id and percent
raise_salary(emp_id, percent);
...
END;
```

ตัวอย่างต่อไป จะเรียกใช้ Function row_count จาก Stand-Alone PL/SQL Stored

Procedure

```
CREATE PROCEDURE calc_bonus (emp_id NUMBER, bonus OUT NUMBER) AS
emp_count NUMBER;
...
BEGIN
emp_count := row_count('emp');
...
END;
```

ตัวอย่างสุดท้าย จะเป็นการเรียกใช้ Method raise_sal ของ Object Type Employee จาก

Anonymous PL/SQL Block

```
DECLARE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
emp_id NUMBER(4);  
v emp_type;  
BEGIN  
-- assign a value to emp_id  
SELECT VALUE(e) INTO v FROM emps e WHERE empno = emp_id;  
v.raise_sal(500);  
UPDATE emps e SET e = v WHERE empno = emp_id;  
...  
END;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบจัดการคลังข้อสอบและการสอบ

ระบบจัดการคลังข้อสอบและการสอบถูกออกแบบเพื่ออำนวยความสะดวกให้กับการทดสอบและประเมินผล จากเดิมในระบบปัจจุบันที่มีลักษณะการทำงานคือ ผู้สอนทำการสร้างข้อสอบในแต่ละวิชาและแจกข้อสอบให้กับนักเรียนเพื่อทำการทดสอบ เมื่อนักเรียนทำข้อสอบเสร็จก็จะส่งข้อสอบพร้อมคำตอบกลับมายังผู้สอน จากนั้นผู้สอนจะทำการตรวจข้อสอบพร้อมทั้งประกาศผลสอบให้นักเรียนทราบ แต่ในปัจจุบันข้อสอบที่สร้างมาจะเป็นข้อสอบที่คำถามเป็นรูปแบบคงที่ไม่มีการเปลี่ยนแปลงคำถามในการสอบวิชาเดิม นอกจากผู้สอนจะสร้างข้อสอบใหม่ขึ้นมาทดแทนทำให้การประเมินผลขาดประสิทธิภาพ เนื่องจากนักเรียนอาจจำคำถามเดิมๆที่เคยสอบมาได้ หากเป็นนักเรียนที่เคยสอบวิชานี้มาก่อน อีกทั้งในการตรวจข้อสอบจะต้องใช้แรงงานคนในการตรวจถึงแม้จะเป็นข้อสอบปรนัยก็ตาม นอกจากนี้ในการทดสอบแต่ละครั้งจะต้องเตรียมเอกสารในส่วนข้อสอบและกระดาษคำตอบ ซึ่งทำให้สูญเสียค่าใช้จ่ายทางด้านเอกสารจำนวนมาก ดังนั้นถ้ามีการพัฒนาระบบให้รองรับปัญหาต่างๆแล้วจะทำให้การทดสอบมีประสิทธิภาพ และยังคงค่าใช้จ่ายด้านเอกสารตลอดจนการตรวจข้อสอบให้เป็นไปอย่างถูกต้องและรวดเร็ว

3.1 ขอบเขตการทำงานของระบบจัดการคลังข้อสอบและการสอบ

ระบบจัดการคลังข้อสอบและการสอบเป็นระบบที่จัดทำขึ้นเพื่อให้บริการด้านการทดสอบและประเมินผล โดยสามารถสร้างข้อสอบต่างๆในแต่ละวิชา ตามแต่ที่ผู้สอนต้องการ ขั้นตอนการทำงานที่สำคัญประกอบด้วยงานหลักๆ ดังนี้

- ให้ผู้สอนทำการสร้าง Site ขึ้นมาในระบบ เพื่อให้สามารถสร้างคลังข้อสอบใน Site ของตนเองได้ จากนั้นจึงทำการสร้างข้อสอบ
- กำหนดคุณสมบัติต่างๆของข้อสอบได้ตามความเหมาะสม เช่น ลักษณะการแสดงผลการสอบ, วิธีการให้คะแนน, หลักการกำหนดคำถามในข้อสอบ
- กำหนดประเภทของคำถามและคำตอบ รวมทั้งการกำหนดระดับความยากง่ายของคำถาม

ข้อกำหนดของระบบคือ

- ผู้สอนจะต้องสร้าง User และ Password ของนักเรียนที่จะให้ทำการสอบใน Site

- ผู้สอนจะกำหนดผู้ที่มีสิทธิ์เข้าสอบในแต่ละข้อสอบ
- ผู้สอนจะกำหนดวิธีการแสดงผลการสอบและเกณฑ์การให้คะแนนของแต่ละข้อสอบ

3.2 ขั้นตอนการทำงาน

ขั้นตอนการทำงานสำหรับระบบจัดการคลังข้อสอบและการสอบประกอบด้วยหน้าที่การทำงานดังนี้

(1) สร้าง Site

เมื่อผู้สอนต้องการใช้งานระบบ จะต้องทำการสร้าง Site ขึ้นมาก่อน เพื่อรองรับการทำงานในการสร้างข้อสอบของแต่ละวิชา

(2) สร้าง User

ผู้สอนทำการสร้าง User, Password ให้กับนักเรียนที่จะเข้ามาสอบภายใน Site

(3) สร้างคลังข้อสอบ

ผู้สอนสามารถสร้างคลังข้อสอบ เพื่อนำไปใช้ในการสร้างข้อสอบ

(4) สร้างข้อสอบ

ผู้สอนจะสร้างข้อสอบ โดยคำถามภายในข้อสอบจะนำมาจากคลังข้อสอบที่มีการสร้างมาก่อนหน้านี้ โดยอาจกำหนดคำถามแต่ละคำถามในข้อสอบเป็นรูปแบบคงที่หรือให้ทำการสุ่มคำถามมาจากคลังข้อสอบก็ได้

(5) กำหนดผู้มีสิทธิ์สอบ

ข้อสอบที่สร้างขึ้นจะต้องกำหนดผู้มีสิทธิ์เข้าสอบ(ขึ้นอยู่กับข้อกำหนดตอนสร้างข้อสอบ) โดยนำ User ของนักเรียนที่สร้างขึ้นมากำหนดว่าใครมีสิทธิ์ในการสอบ

(6) ทำการสอบ

นักเรียนที่ได้รับ User, Password จะทำการล็อกเข้าสู่ระบบเพื่อทำการสอบ

(7) ตรวจสอบข้อสอบ

กรณีที่นักเรียนทำการสอบเสร็จแล้ว ระบบจะทำการตรวจสอบข้อสอบหรือให้ผู้สอนมาทำการตรวจสอบข้อสอบ(ขึ้นอยู่กับข้อกำหนดตอนสร้างข้อสอบ)

(8) ดูผลการสอบ

เมื่อการตรวจสอบข้อสอบเสร็จสิ้น ระบบจะทำการประกาศผลสอบ นักเรียนที่เข้าสอบจะสามารถดูผลสอบของตัวเองได้ แต่ไม่สามารถดูผลสอบของผู้อื่นได้

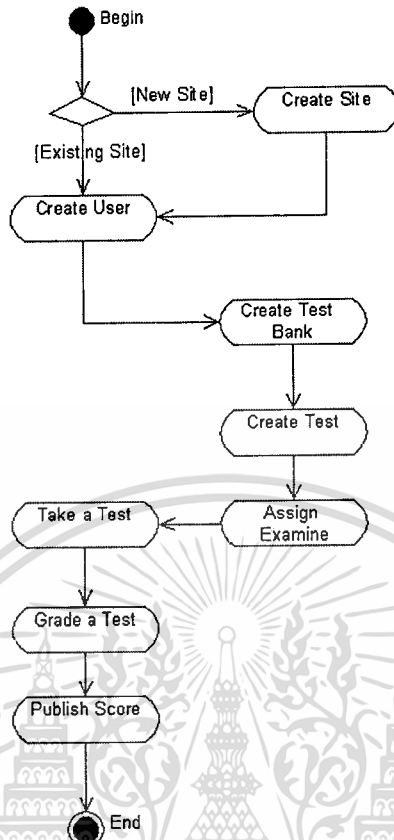
3.3 การวิเคราะห์และออกแบบระบบ

ก่อนที่จะเริ่มทำการวิเคราะห์และออกแบบระบบ สิ่งสำคัญคือต้องเข้าใจว่าการวิเคราะห์ระบบคืออะไรและการออกแบบระบบคืออะไรเสียก่อน โดยการวิเคราะห์ระบบคือการเข้าใจและรับทราบถึงปัญหาที่เกิดขึ้นภายในระบบ รวมถึงความต้องการของผู้ใช้งานระบบ ส่วนการออกแบบระบบคือการหาวิธีการแก้ไขปัญหาที่เกิดขึ้นและตอบสนองผู้ใช้งานที่ต้องการให้ระบบสามารถทำอะไรได้บ้าง โดยจะสามารถวิเคราะห์ระบบได้ตามขั้นตอนดังต่อไปนี้

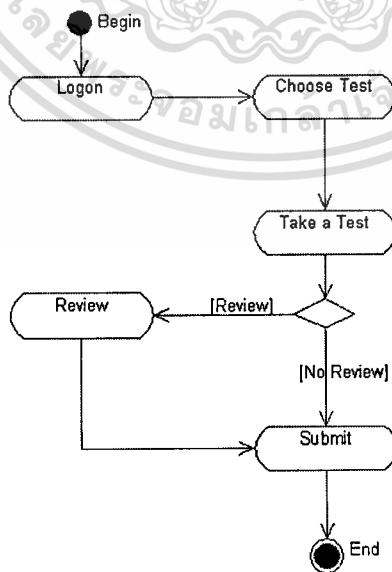
3.3.1 กำหนดแอกเตอร์ จากการวิเคราะห์ระบบจัดการคลังข้อสอบและการสอบ ในขั้นต้นจะค้นหาแอกเตอร์ของระบบก่อน จากการศึกษาพบว่าผู้ใช้ระบบคือ ผู้สอน และนักเรียน ดังนั้นแอกเตอร์ของระบบ คือ

- (1) ผู้สอน (Instructor) ทำหน้าที่ในการสร้าง Site, คลังข้อสอบ และข้อสอบให้นักเรียนทำการสอบ ตลอดจนทำหน้าที่ในการประเมินผลการสอบ
- (2) นักเรียน (Learner) ทำหน้าที่เข้าสอบข้อสอบต่างๆภายในในระบบ

3.3.2 กำหนดแบบจำลองของกระบวนการทางธุรกิจ (Business Process Model) สำหรับการออกแบบ Business Process Model ในขั้นตอนนี้จะใช้เอกทวิดีโคอะแกรมเพื่อแสดงการทำงานของระบบ ดังแสดงในรูปที่ 3.1 โดยมีขั้นตอนการทำงานคือ ผู้สอนเริ่มใช้งานระบบ หากเป็นการใช้งานครั้งแรกก็ต้องทำการสร้าง Site ก่อน จากนั้นจึงจะสามารถสร้าง User, Password สำหรับนักเรียนที่จะเข้าสอบภายใน Site ได้ ขั้นตอนต่อมาคือสร้างคลังคำถามเพื่อเตรียมไว้สำหรับการสร้างข้อสอบ เมื่อสร้างคลังข้อสอบเสร็จผู้สอนก็จะสร้างข้อสอบโดยใช้คำถามที่สร้างเตรียมไว้จากคลังข้อสอบและทำการกำหนดผู้มีสิทธิ์เข้าสอบในข้อสอบที่สร้าง ส่วนต่อไปจะเป็นการเข้าสอบของนักเรียน เมื่อนักเรียนทำการสอบเสร็จเรียบร้อย ระบบก็จะทำการตรวจข้อสอบและประกาศผลการสอบให้ทราบ



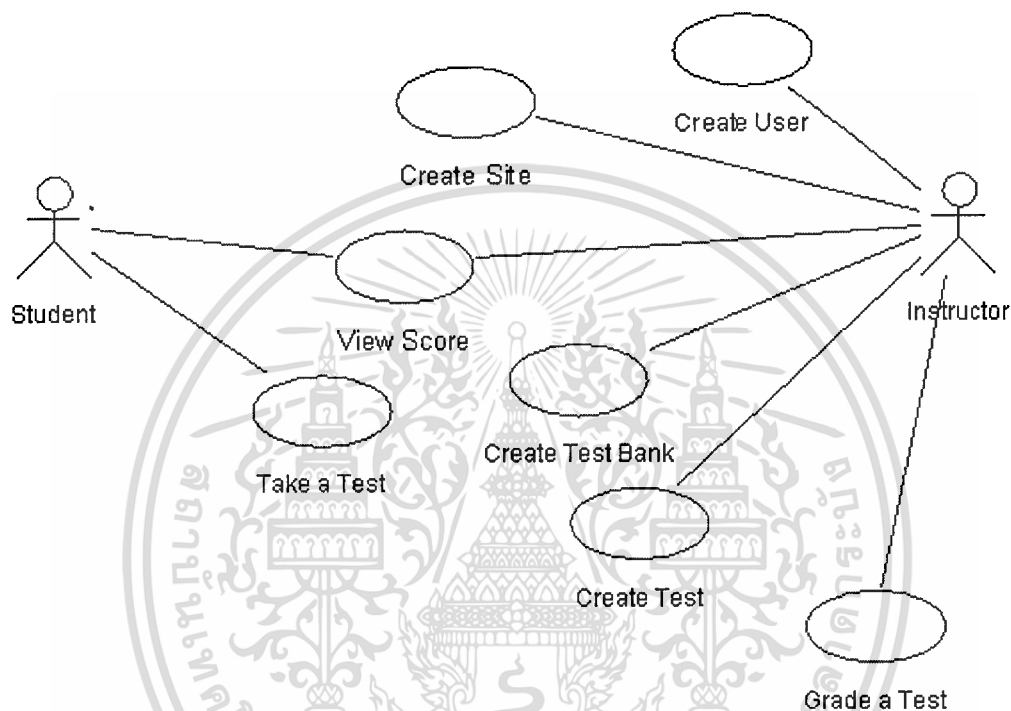
รูปที่ 3.1 แอคทีวิตี้ไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบ



รูปที่ 3.2 แอคทีวิตี้ไดอะแกรมการเข้าสอบของนักเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การสร้างยูสเคสของระบบ ในขั้นตอนนี้จะเป็นการสร้างยูสเคสไดอะแกรมแสดงการทำงานระหว่างผู้ใช้กับระบบ โดยแสดงถึงการติดต่อระหว่างผู้ใช้และงานของระบบที่ผู้ใช้ต้องการจากระบบ ตัวอย่างเช่นผู้สอนทำการสร้างข้อสอบ หรือนักเรียนทำการสอบและดูผลสอบ ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 ยูสเคสไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบ

จากยูสเคสไดอะแกรมที่ได้จากการศึกษาความต้องการของระบบเป็นเวลาหลายรอบเพื่อให้มีความถูกต้องสมบูรณ์ มีรายละเอียดแต่ละยูสเคสดังต่อไปนี้

(1) ชื่อยูสเคส : Create Site คือยูสเคสแสดงขั้นตอนการสร้าง Site

รายละเอียด :

1. เลือกรายการสร้าง Site ใหม่
2. ระบุชื่อ Site
3. ระบุชื่อ, รหัสสำหรับผู้สอนเพื่อใช้ในการดูแล Site

(2) ชื่อยูสเคส : Create User คือยูสเคสแสดงการสร้าง User

รายละเอียด :

1. ผู้สอนสร้าง User และกำหนด Password สำหรับนักเรียนเพื่อใช้ในการสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณีมีนักเรียนใหม่ จะต้องทำการสร้าง User ให้กับนักเรียน

(3) ชื่อยูสเทส : Create Test Bank คือ ยูสเทสแสดงการคลังข้อสอบ

รายละเอียด :

1. กำหนดชื่อคลังข้อสอบและคำอธิบาย
2. สร้างคำถามเพิ่มเข้าไปยังคลังข้อสอบ
3. กำหนดคำถามและวิธีการตอบคำถาม
4. กำหนดคำตอบของคำถามแต่ละข้อ

(4) ชื่อยูสเทส : Create Test คือ ยูสเทสการสร้างข้อสอบ

รายละเอียด :

1. กำหนดชื่อข้อสอบและคำอธิบาย
2. ระบุวิธีการนำคำถามจากคลังข้อสอบและจำนวนคำถามในข้อสอบ
3. ระบุเกณฑ์การให้คะแนนของข้อสอบ
4. ระบุจำนวนครั้งสูงสุดที่สามารถเข้าสอบ
5. ระบุเวลาที่ใช้ในการสอบ

(5) ชื่อยูสเทส : Take a Test คือ ยูสเทสการสอบของนักเรียน

รายละเอียด :

1. นักเรียนเลือกข้อสอบที่ต้องการสอบ
2. นักเรียนทำการสอบ
3. เมื่อสอบเสร็จ จะสามารถตรวจทานข้อสอบก่อนส่ง
4. ส่งข้อสอบ

(6) ชื่อยูสเทส : Grade a Test คือ ยูสเทสการตรวจข้อสอบของผู้สอน

รายละเอียด :

1. ผู้สอนเลือกข้อสอบที่ต้องการตรวจ
2. ผู้สอนตรวจข้อสอบ
3. กำหนดคะแนนให้แก่ นักเรียนที่ทำการสอบ
4. ส่งผลการตรวจข้อสอบ

(7) ชื่อยูสเทส : View Score คือ ยูสเทสแสดงผลการสอบ

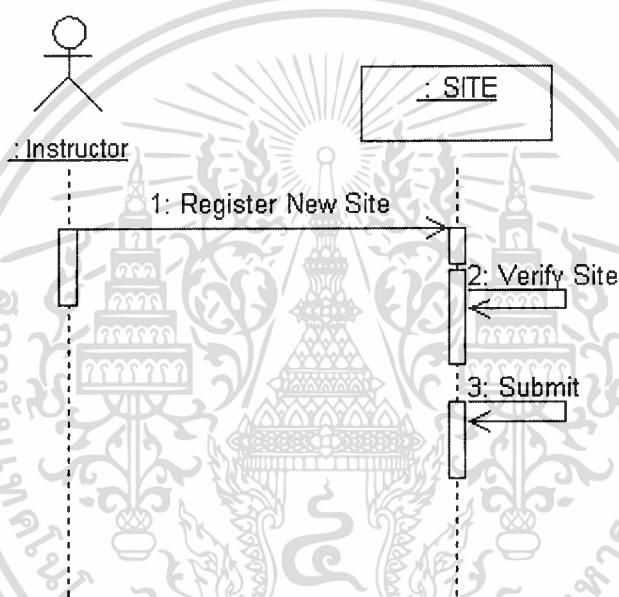
รายละเอียด :

1. นักเรียนหรือผู้สอนเลือกข้อสอบที่ต้องการดูผลการสอบ
2. ดูผลการสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

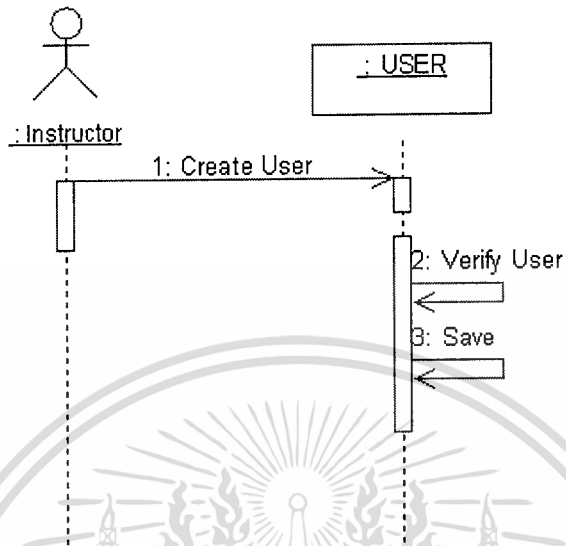
3.3.4 การสร้างซีเควนซ์และสเตทไดอะแกรมของระบบ ในขั้นตอนนี้จะนำยูสเคสที่ได้มาวิเคราะห์ต่อเพื่อสร้างซีเควนซ์ไดอะแกรมและสเตทไดอะแกรม โดยซีเควนซ์ไดอะแกรมเป็นการแสดงลำดับเหตุการณ์ที่เกิดขึ้นภายในยูสเคสแต่ละตัว และใช้สเตทไดอะแกรมแสดงสถานะของผู้สอนและนักเรียนเมื่อเข้ามาในระบบ โดยรายละเอียดของไดอะแกรมจะแสดงในแต่ละหัวข้อดังต่อไปนี้

(1) ซีเควนซ์ไดอะแกรมของยูสเคส Create Site



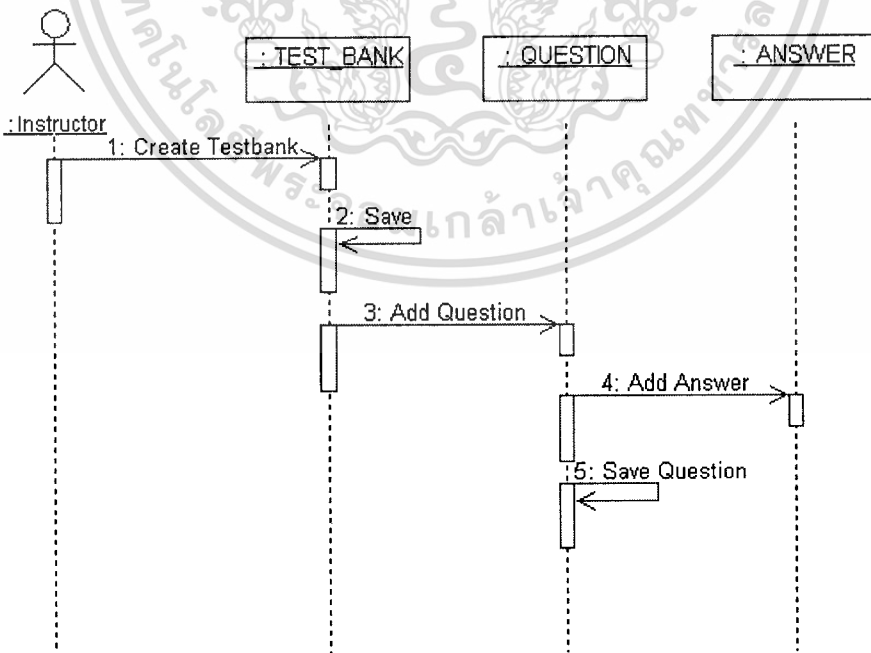
รูปที่ 3.4 ซีเควนซ์ไดอะแกรมของยูสเคส Create Site

(2) ซีเควนซ์ไดอะแกรมของยูสเคส Create User



รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของยูสเคส Create User

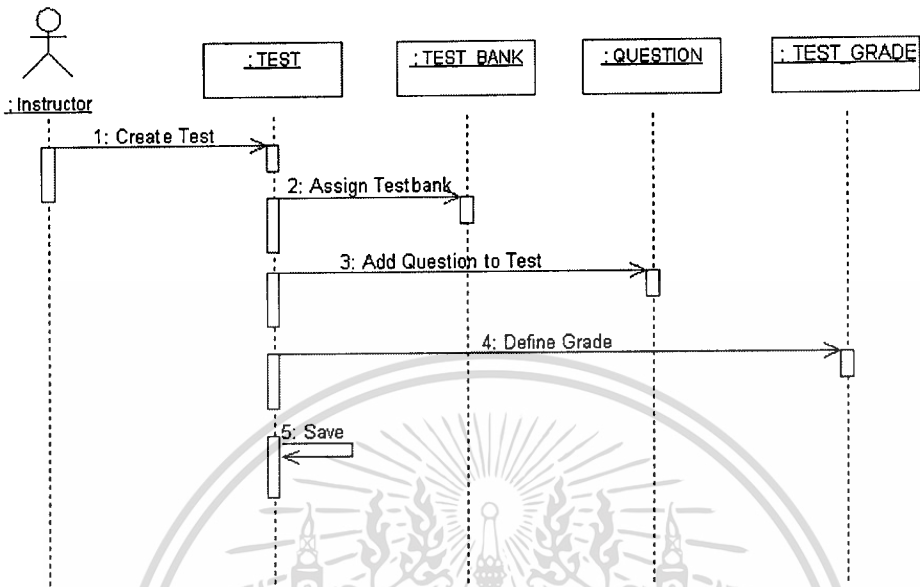
(3) ซีเควนซ์ไดอะแกรมของยูสเคส Create Test Bank



รูปที่ 3.6 ซีเควนซ์ไดอะแกรมของยูสเคส Create Test Bank

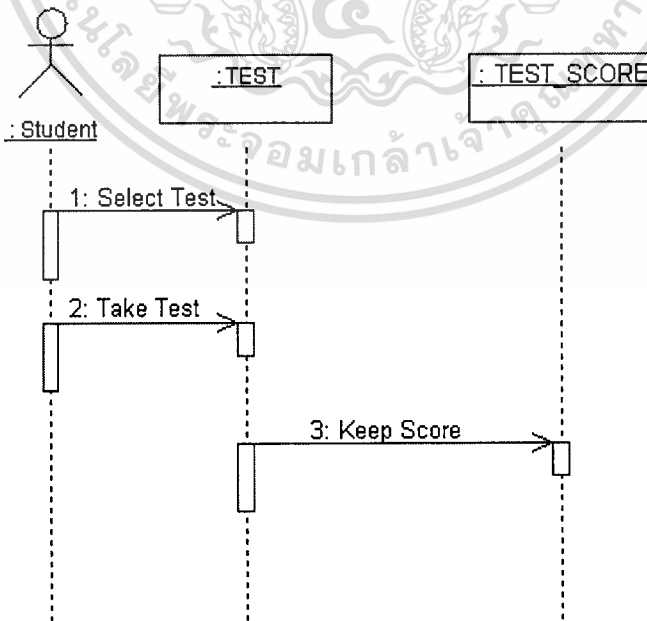
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4) ซีเควนซ์ไดอะแกรมของยูสเคส Create Test



รูปที่ 3.7 ซีเควนซ์ไดอะแกรมของยูสเคส Create Test

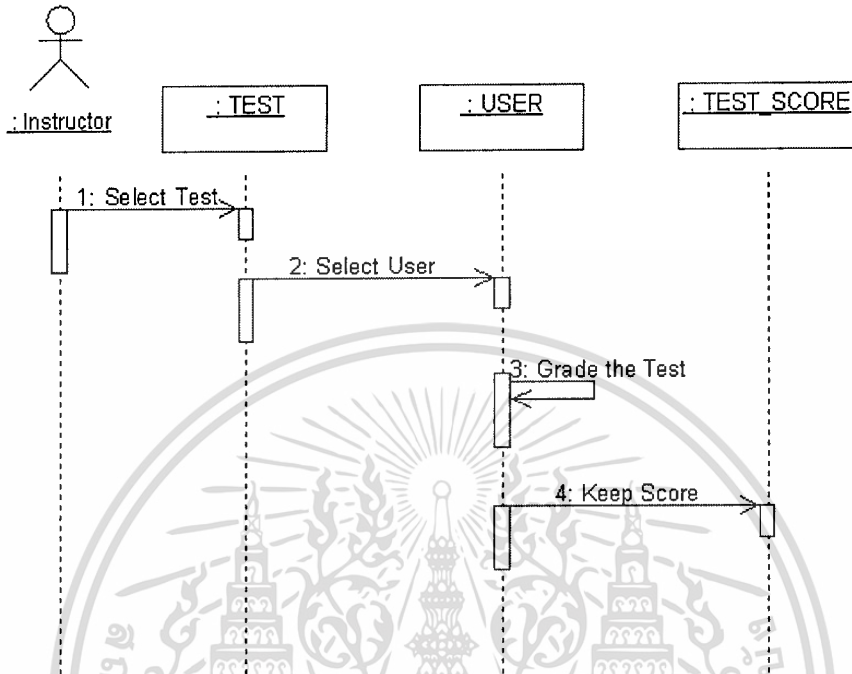
(5) ซีเควนซ์ไดอะแกรมของยูสเคส Take a Test



รูปที่ 3.8 ซีเควนซ์ไดอะแกรมของยูสเคส Take a Test

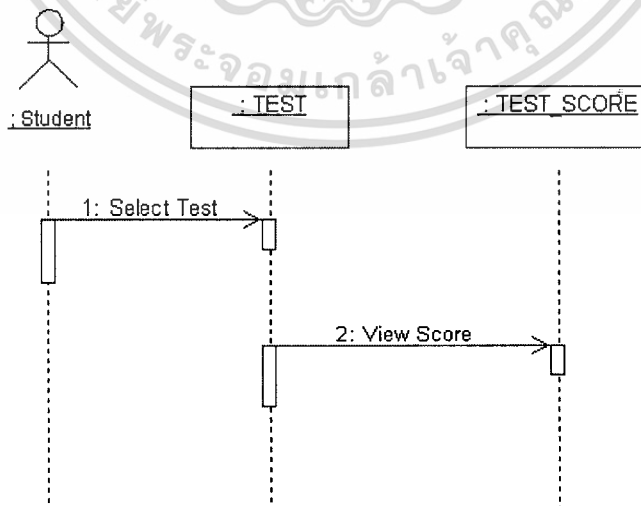
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(6) ซีเควนซ์ไดอะแกรมของยูสเคส Grade a Test



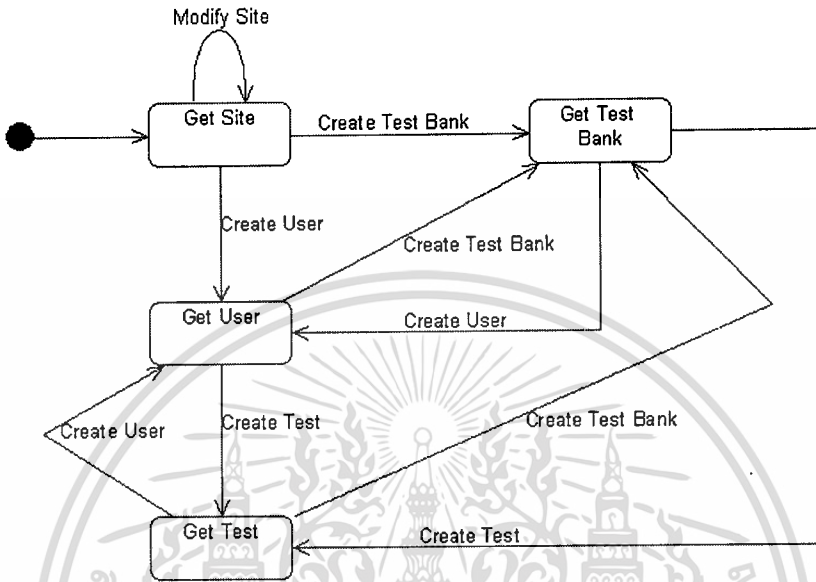
รูปที่ 3.9 ซีเควนซ์ไดอะแกรมของยูสเคส Grade a Test

(7) ซีเควนซ์ไดอะแกรมของยูสเคส View Score



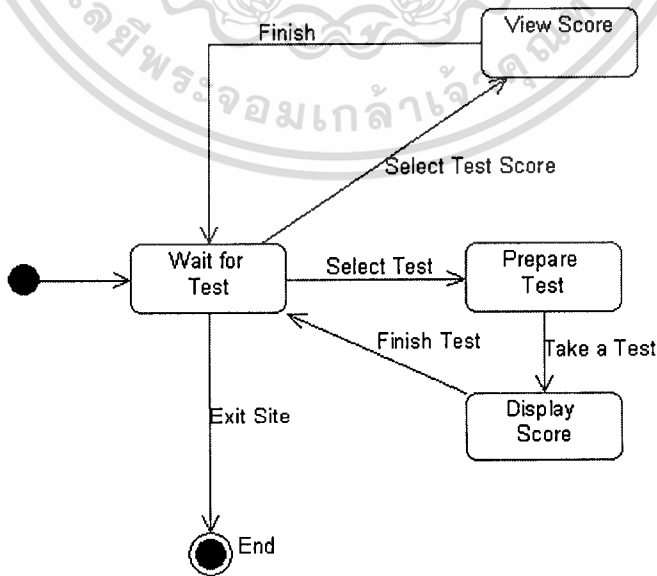
รูปที่ 3.10 ซีเควนซ์ไดอะแกรมของยูสเคส View Score

(8) สเตทไดอะแกรมแสดงสถานะของผู้สอน



รูปที่ 3.11 สเตทไดอะแกรมแสดงสถานะของผู้สอน

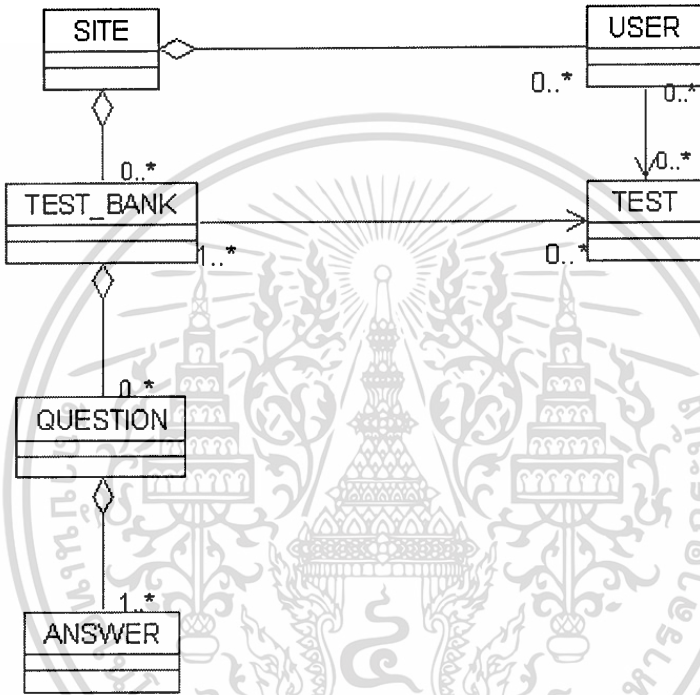
(9) สเตทไดอะแกรมแสดงสถานะของนักเรียน



รูปที่ 3.12 สเตทไดอะแกรมแสดงสถานะของนักเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 การสร้างคลาสไต่อะแกรมของระบบ จากการวิเคราะห์และออกแบบระบบจัดการคลังข้อสอบและการสอบโดยกำหนดโมเดลของยูสเคสไต่อะแกรมขึ้นมา นั้น ในขั้นตอนต่อไปคือการสร้างคลาสไต่อะแกรมของระบบ จากการวิเคราะห์ระบบเดิมที่มีอยู่ในปัจจุบัน ทำให้ได้คลาสไต่อะแกรมของระบบแสดงดังรูปที่ 3.13



รูปที่ 3.13 คลาสไต่อะแกรมระบบจัดการคลังข้อสอบและการสอบในขั้นตอนการวิเคราะห์

จากคลาสไต่อะแกรมในขั้นตอนนี้ สามารถกำหนดความสัมพันธ์ระหว่างคลาสภายในระบบได้ดังต่อไปนี้

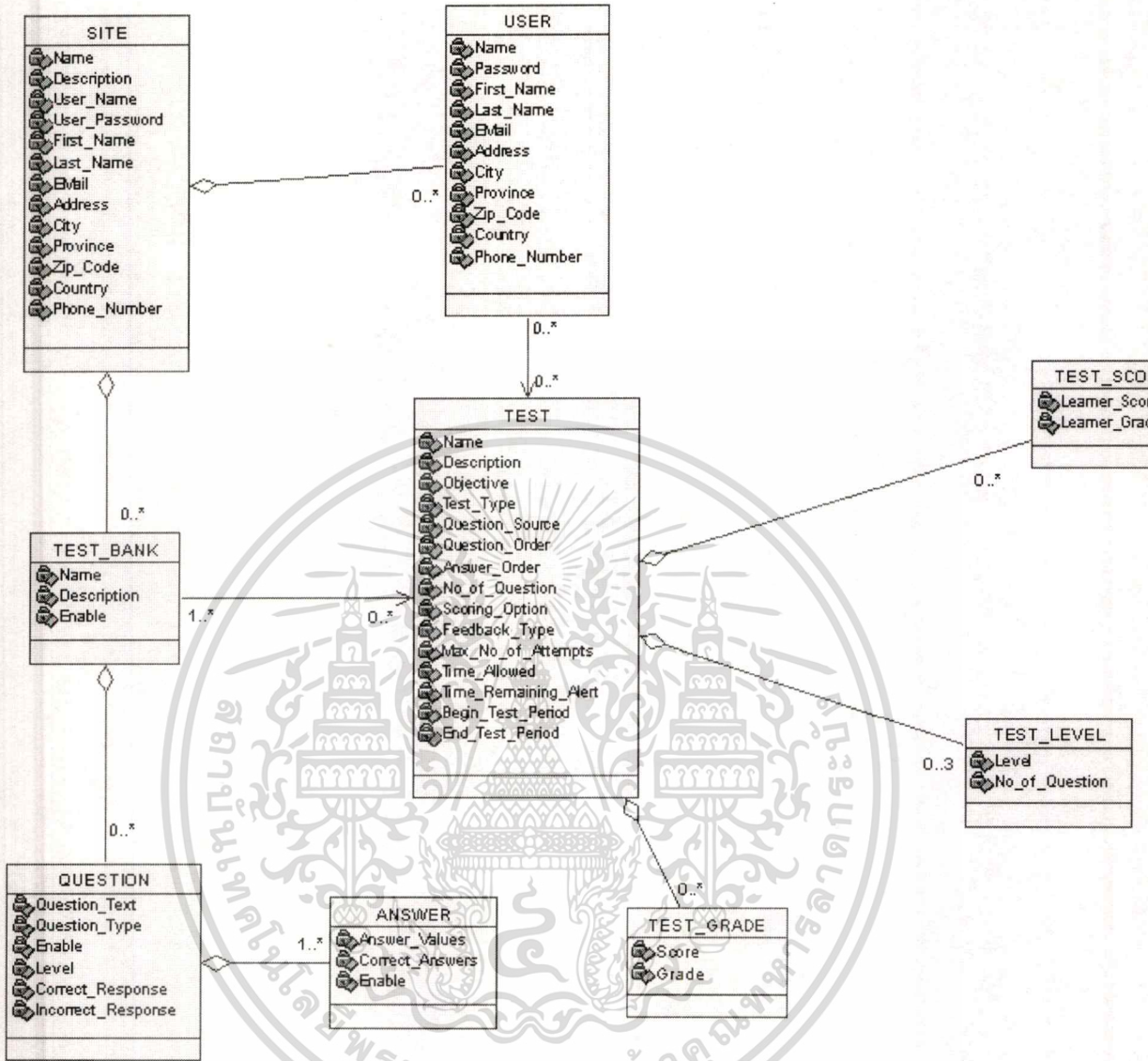
- (1) Site กับ Learner การทำข้อสอบของนักเรียนนั้นจะต้องทำการสร้าง User, Password ของนักเรียนภายใน Site ก่อน จึงจะสามารถเข้าทำข้อสอบได้ ใน Site จะสามารถมีนักเรียนกี่คนก็ได้ ดังนั้นความสัมพันธ์ระหว่าง Site และ Learner จึงเป็นแบบ Aggregation
- (2) Site กับ TestBank ใน Site จะสามารถสร้างคลังข้อสอบขึ้นมาเพื่อเป็นแหล่งที่มาของการสร้างข้อสอบ และจะมีได้หลายๆคลังข้อสอบใน 1 Site จึงทำให้เกิดความสัมพันธ์ระหว่าง Site และ TestBank เป็นแบบ Aggregation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (3) TestBank กับ Question ในคลังข้อสอบจะมีการสร้างคำถามต่างๆในแต่ละประเภทเพื่อใช้ในการสร้างข้อสอบ ทำให้เกิดความสัมพันธ์แบบ Aggregation ระหว่าง TestBank และ Question
- (4) Question กับ Answer เนื่องจากการสร้างคำถามในแต่ละครั้งจะต้องมีการระบุคำตอบ โดยอาจมีตัวเลือกที่เป็นคำตอบถูกต้องเพียง 1 คำตอบหรือหลายๆคำตอบ หรือเป็นการตอบแบบเติมคำก็ได้ ดังนั้นใน 1 คำถามจะมีได้หลายๆตัวเลือกและหลายๆคำตอบ จึงทำให้เกิดความสัมพันธ์แบบ Aggregation ระหว่าง Question และ Answer
- (5) Learner กับ Test หลังจากที่มีการสร้าง User, Password ให้กับนักเรียนภายใน Site เรียบร้อยแล้ว หากต้องการให้นักเรียนทำการสอบจะต้องทำการกำหนดข้อสอบที่นักเรียนสามารถสอบได้ นักเรียน 1 คนจะสอบข้อสอบได้หลายชุดแล้วแต่ผู้สอนกำหนด ในขณะที่ข้อสอบ 1 ชุดก็สามารถมีนักเรียนเข้าสอบได้หลายคนเช่นกัน ดังนั้นจึงเกิดความสัมพันธ์แบบ Association ระหว่าง Learner และ Test
- (6) Test กับ TestBank ในการสร้างข้อสอบแต่ละชุดนั้น จะต้องมีการกำหนดคำถามภายในข้อสอบ แต่คำถามที่จะนำมาไว้ในข้อสอบจะมีการสร้างไว้ใน TestBank เรียบร้อยแล้ว ดังนั้นการกำหนดคำถามในข้อสอบจะเป็นการกำหนดจาก TestBank ในข้อสอบ 1 ชุดจะกำหนดคำถามจาก TestBank ได้เพียง 1 เท่านั้น แต่ TestBank จะเป็นแหล่งคำถามให้กับข้อสอบได้หลายๆชุด จึงทำให้เกิดความสัมพันธ์ระหว่าง Test และ TestBank แบบ Association

3.3.6 การออกแบบระบบ จากคลาสที่ได้ในขั้นตอนการวิเคราะห์จะนำมาออกแบบระบบต่อไป โดยนำคลาสเหล่านั้นมาออกแบบให้เหมาะสมกับการสร้างระบบ

3.3.6.1 การออกแบบคลาสเพิ่มเติม จากความสัมพันธ์ของคลาสไดอะแกรมในรูปแบบที่ 3.13 คลาสที่มีในขั้นตอนการวิเคราะห์นั้น ไม่เพียงพอสำหรับทำงานของระบบ ดังนั้นจึงต้องมีการออกแบบคลาสเพิ่มเติมจากเดิม เพื่อให้รองรับการทำงานของระบบได้ คลาสที่มีการออกแบบเพิ่มเติมจากเดิมได้แก่ คลาส TestScore, คลาส TestLevel และ TestGrade เป็นคลาสที่เก็บข้อมูลเกี่ยวกับการทำข้อสอบ, การกำหนดระดับคำถามในข้อสอบและระดับการให้คะแนนในแต่ละข้อสอบ หลังจากออกแบบคลาสเพิ่มเติมแล้วจะได้คลาสไดอะแกรมที่มีความชัดเจนขึ้น ดังแสดงในรูปที่ 3.5



รูปที่ 3.14 คลาสไดอะแกรมของระบบจัดการคลังข้อสอบและการสอบในขั้นตอนการออกแบบ

3.3.6.2 การออกแบบแอททริบิวต์ ในขั้นตอนนี้จะออกแบบแอททริบิวต์เพื่อนำไปใช้ในขั้นตอนการพัฒนาระบบ โดยขั้นตอนการออกแบบได้แก่ การกำหนดชนิดข้อมูล การกำหนดขนาดข้อมูล และคำอธิบายของแอททริบิวต์ จากคลาสไดอะแกรมดังแสดงในรูปที่ 3.14 มีแอททริบิวต์ดังต่อไปนี้

- (1) คลาส Site คือ ข้อมูลของ Site ที่ใช้ในการสอบ ซึ่งมีแอททริบิวต์แสดงดังตารางที่ 3.1
- (2) คลาส TestBank คือ ข้อมูลคลังข้อสอบ ซึ่งมีแอททริบิวต์แสดงดังตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (3) คลาส Question คือ ข้อมูลรายละเอียดของคำถาม มีแอททริบิวต์แสดงดังตารางที่ 3.3
- (4) คลาส Answer คือ ข้อมูลคำตอบของแต่ละคำถาม มีแอททริบิวต์แสดงดังตารางที่ 3.4
- (5) คลาส Learner คือ ข้อมูลของนักเรียน มีแอททริบิวต์แสดงดังตารางที่ 3.5
- (6) คลาส Test คือ ข้อมูลข้อสอบ มีแอททริบิวต์แสดงดังตารางที่ 3.6
- (7) คลาส TestGrade คือ ข้อมูลการให้คะแนนของแต่ละข้อสอบ มีแอททริบิวต์แสดงดังตารางที่ 3.7
- (8) คลาส TestScore คือ ข้อมูลผลการสอบ มีแอททริบิวต์แสดงดังตารางที่ 3.8
- (9) คลาส TestLevel คือ ข้อมูลการกำหนดระดับคำถามในข้อสอบ มีแอททริบิวต์แสดงดังตารางที่ 3.9

ตารางที่ 3.1 แอททริบิวต์ของคลาส Site

แอททริบิวต์	ชนิดข้อมูล	Contents
SiteName	Varchar2(50)	ชื่อ Site
SiteDescription	Varchar2(300)	คำอธิบาย
UserName	Varchar2(15)	ชื่อที่ใช้ในการ log เข้าสู่ระบบของผู้สอน
Password	Varchar2(15)	รหัสผ่านของผู้สอน
FirstName	Varchar2(15)	ชื่อผู้สอน
LastName	Varchar2(15)	นามสกุลผู้สอน
Email	Varchar2(30)	อีเมลของผู้สอน
Address	Varchar2(100)	ที่อยู่
City	Varchar2(20)	เมือง
Province	Varchar2(20)	รัฐ
ZipCode	Varchar2(15)	รหัสไปรษณีย์
Country	Varchar2(20)	ประเทศ
PhoneNumber	Varchar2(20)	เบอร์โทรศัพท์

ตารางที่ 3.2 แอททริบิวต์ของคลาส TestBank

แอททริบิวต์	ชนิดข้อมูล	Contents
Name	Varchar2(15)	ชื่อคลังข้อสอบ
Description	Varchar2(300)	คำอธิบาย
Enable	Varchar2(1)	สถานะการใช้งานคลังข้อสอบ

ตารางที่ 3.3 แอททริบิวต์ของคลาส Question

แอททริบิวต์	ชนิดข้อมูล	Contents
QuestionText	Varchar2(500)	ข้อความคำถาม
QuestionType	Varchar2(1)	ชนิดคำถาม
Enable	Varchar2(1)	สถานะการใช้งานคำถาม
Level	Varchar2(1)	ระดับความยากง่ายของคำถาม
CorrectResponse	Varchar2(300)	ข้อความตอบรับกรณีตอบถูก
IncorrectResponse	Varchar2(300)	ข้อความตอบรับกรณีตอบผิด

ตารางที่ 3.4 แอททริบิวต์ของคลาส Answer

แอททริบิวต์	ชนิดข้อมูล	Contents
AnswerValues	Varchar2(300)	ข้อความคำตอบ
CorrectAnswers	Varchar2(1)	สถานะคำตอบ(ถูก/ไม่ถูก)
Enable	Varchar2(1)	สถานะการใช้งานคำตอบ

ตารางที่ 3.5 แอททริบิวต์ของคลาส Learner

แอททริบิวต์	ชนิดข้อมูล	Contents
Name	Varchar2(30)	ชื่อที่ใช้ในการ log ผู้ระบบของนักเรียน
Password	Varchar2(10)	รหัสผ่านของนักเรียน
FirstName	Varchar2(15)	ชื่อนักเรียน
LastName	Varchar2(15)	นามสกุลนักเรียน
EMail	Varchar2(30)	อีเมลล์ของนักเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 แอททริบิวต์ของคลาส Learner(ต่อ)

แอททริบิวต์	ชนิดข้อมูล	Contents
Address	Varchar2(100)	ที่อยู่
City	Varchar2(20)	เมือง
Province	Varchar2(20)	รัฐ
ZipCode	Varchar2(15)	รหัสไปรษณีย์
Country	Varchar2(20)	ประเทศ
PhoneNumber	Varchar2(20)	เบอร์โทรศัพท์

ตารางที่ 3.6 แอททริบิวต์ของคลาส Test

แอททริบิวต์	ชนิดข้อมูล	Contents
Name	Varchar2(30)	ชื่อข้อสอบ
Description	Varchar2(300)	คำอธิบายข้อสอบ
Objective	Varchar2(100)	เป้าหมายข้อสอบ
TestType	Varchar2(1)	ชนิดข้อสอบ
QuestionSource	Varchar2(1)	วิธีการตั้งคำถามจากคลังคำถาม
QuestionOrder	Varchar2(1)	วิธีลำดับคำถามในข้อสอบ
AnswerOrder	Varchar2(1)	วิธีการเรียงลำดับคำตอบ
NoofQuestion	Number(4)	จำนวนคำถามในข้อสอบ
ScoringOption	Varchar2(1)	ประเภทการให้คะแนน
FeedbackType	Varchar2(1)	ชนิดการโต้ตอบเมื่อทำข้อสอบเสร็จ
MaxNoofAttempts	Number(3)	จำนวนครั้งสูงสุดที่สามารถเข้าสอบ
TimeAllowed	Number(4)	กำหนดเวลาที่ใช้ในการสอบ
TimeRemainingAlert	Number(4)	เวลาที่แจ้งเตือนเมื่อเหลือเวลาในการสอบ
BeginTestPeriod	Date	วันเวลาเริ่มต้นที่อนุญาตให้สอบ
EndTestPeriod	Date	วันเวลาสิ้นสุดที่อนุญาตให้สอบ

ตารางที่ 3.7 แอททริบิวต์ของคลาส TestGrade

แอททริบิวต์	ชนิดข้อมูล	Contents
Score	Number(3)	คะแนน
Grade	Varchar2(10)	เกรดที่จะได้เมื่อได้คะแนนสูงกว่าแอททริบิวต์ Score

ตารางที่ 3.8 แอททริบิวต์ของคลาส TestScore

แอททริบิวต์	ชนิดข้อมูล	Contents
LearnerScore	Number(3)	คะแนนที่สอบได้
LearnerGrade	Varchar2(10)	เกรดที่ได้

ตารางที่ 3.9 แอททริบิวต์ของคลาส TestLevel

แอททริบิวต์	ชนิดข้อมูล	Contents
Level	Varchar2(1)	ระดับความยากของคำถาม
NoofQuestion	Number(3)	จำนวนคำถามในแต่ละระดับ

บทที่ 4

การสร้างฐานข้อมูลระบบระบบจัดการคลังข้อสอบและการสอบ

หลังจากทำการรวบรวมข้อมูลต่างๆที่ได้มาจากการวิเคราะห์และออกแบบของระบบจัดการคลังข้อสอบและการสอบแล้ว ขั้นตอนต่อไปคือการนำข้อมูลเหล่านั้นมาใช้ในการออกแบบฐานข้อมูลและแอปพลิเคชัน

4.1 การแปลงแบบจำลองเป็นฐานข้อมูล

ด้วยแนวทางการออกแบบระบบเชิงวัตถุ ตัวแอปพลิเคชันจะทำการติดต่อกับข้อมูลที่เก็บในฐานข้อมูลเชิงสัมพันธ์ แต่การออกแบบเชิงวัตถุจะไม่สามารถนำมาใช้กับข้อมูลที่อยู่ในตารางฐานข้อมูลได้ จึงต้องใช้วิธี Forward Engineering ทำการสร้างตารางฐานข้อมูลจากออบเจกต์โมเดลฐานข้อมูลเชิงสัมพันธ์ประกอบด้วยตารางที่มีแถวและคอลัมน์ ซึ่งหากเปรียบเทียบกับในออบเจกต์โมเดลก็คือคลาส แต่ละแถวในตารางที่ประกอบด้วยข้อมูลก็จะเปรียบเสมือนอินสแตนซ์ (Instance) ของคลาส ส่วนคอลัมน์ก็เปรียบเสมือนแอททริบิวต์ (Attribute) และ Procedure, Function ก็เปรียบเสมือนกับเมธอด (Method) ดังนั้นจึงจำเป็นต้องทำการแปลงคลาสไคอะแกรมที่สร้างขึ้นให้อยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์โดยการทำ Forward Engineering ซึ่งสามารถทำได้ผ่านตัว CASE Tool Rational Rose โดยจะสร้างไฟล์นามสกุล .SQL ขึ้นมาเป็น Script แล้วนำไฟล์ที่ได้มา Run ที่ฐานข้อมูลของ Oracle เพื่อสร้างเป็นตาราง

4.2 รายละเอียดข้อมูลที่จัดเก็บในระบบ

จากการทำ Forward Engineering เพื่อสร้างตารางในฐานข้อมูลนั้น ตารางที่ได้ยังมีความไม่สมบูรณ์อยู่มาก จึงต้องมีการเพิ่มเติมแอททริบิวต์ต่างๆลงในตารางและสร้างตารางใหม่ขึ้นมาเพื่อความสมบูรณ์ของข้อมูลในระบบ จากนั้นก็ทำการ Normalization เพื่อลดความซ้ำซ้อนของข้อมูลจนได้ตารางที่พร้อมใช้งาน คือ

- (1) ตาราง Site
- (2) ตาราง UserSite
- (3) ตาราง SessionSite

- (4) ตาราง TestbankSite
- (5) ตาราง Question
- (6) ตาราง Answer
- (7) ตาราง Test
- (8) ตาราง TestLevel
- (9) ตาราง TestGrade
- (10) ตาราง TestScore
- (11) ตาราง TestFixedQuestion
- (12) ตาราง TakenTest

ซึ่งมีรายละเอียดของตารางต่างๆ ดังตารางที่ 4.1 - 4.12

ตารางที่ 4.1 Site

Table Name : Site					
Description : ข้อมูลหลักของ Site ที่มีในระบบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
SiteName	ชื่อ Site	Varchar2	50	PK	
SiteDesc	คำอธิบาย Site	Varchar2	300		
Username	ชื่อที่ใช้ในการ log เข้าสู่ระบบของผู้สอน	Varchar2	15		
Password	รหัสผ่านของผู้สอน	Varchar2	15		
FirstName	ชื่อผู้สอน	Varchar2	15		
LastName	นามสกุลผู้สอน	Varchar2	15		
Email	อีเมลของผู้สอน	Varchar2	30		
Address	ที่อยู่	Varchar2	100		
City	เมือง	Varchar2	20		
Province	รัฐ	Varchar2	20		
ZipCode	รหัสไปรษณีย์	Varchar2	15		
Country	ประเทศ	Varchar2	20		
PhoneNumber	เบอร์โทรศัพท์	Varchar2	20		

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาตจากทางผู้จัดทำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 UserSite

Table Name : UserSite					
Description : ข้อมูลหลักของนักเรียนในแต่ละ Site					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
UserId	รหัสนักเรียน	Varchar2	10	PK	
SiteName	ชื่อ Site	Varchar2	50	FK	Site
UserName	ชื่อที่ใช้ในการ log เข้าสู่ระบบของนักเรียน	Varchar2	15		
Password	รหัสผ่าน	Varchar2	15		
FirstName	ชื่อนักเรียน	Varchar2	15		
LastName	นามสกุลนักเรียน	Varchar2	15		
EEmail	อีเมลล์ของนักเรียน	Varchar2	30		
Address	ที่อยู่	Varchar2	100		
City	เมือง	Varchar2	20		
Province	รัฐ	Varchar2	20		
ZipCode	รหัสไปรษณีย์	Varchar2	15		
Country	ประเทศ	Varchar2	20		
PhoneNumber	เบอร์โทรศัพท์	Varchar2	20		

ตารางที่ 4.3 SessionSite

Table Name : SessionSite					
Description : ข้อมูลหลักการทำงานของผู้ใช้ภายใน Site					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
SessionId	รหัส Session	Number	10	PK	
SiteName	ชื่อ Site	Varchar2	50	FK	Site
UserName	ชื่อนักเรียน	Varchar2	15	FK	Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 TestbankSite

Table Name : TestbankSite					
Description : ข้อมูลหลักคลังข้อสอบในแต่ละ Site					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestbankId	รหัสคลังข้อสอบ	Number	10	PK	
SiteName	ชื่อ Site	Varchar2	50	FK	Site
Name	ชื่อคลังข้อสอบ	Varchar2	15		
Description	คำอธิบายคลังข้อสอบ	Varchar2	300		
Enable	สถานะการใช้งานคลังข้อสอบ	Varchar2	1		

ตารางที่ 4.5 Question

Table Name : Question					
Description : ข้อมูลหลักคำถามในคลังข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
QuestionId	รหัสคำถาม	Number	10	PK	
TestbankId	รหัสคลังข้อสอบ	Number	10	FK	TestbankSite
QuestionText	ข้อความคำถาม	Varchar2	500		
QuestionType	ชนิดคำถาม	Varchar2	1		
Enable	สถานะการใช้งานคำถาม	Varchar2	1		
Level	ระดับความยากง่ายของคำถาม	Varchar2	1		
CorrectResponse	ข้อความตอบรับกรณีตอบถูก	Varchar2	300		
IncorrectResponse	ข้อความตอบรับกรณีตอบผิด	Varchar2	300		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 Answer

Table Name : Answer					
Description : ข้อมูลหลักคำตอบของแต่ละคำถาม					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
AnswerId	รหัสคำตอบ	Number	10	PK	
QuestionId	รหัสคำถาม	Number	10	FK	Question
AnswerValue	ข้อความคำตอบ	Varchar2	300		
CorrectAnswer	สถานะคำตอบ(ถูก/ไม่ถูก)	Varchar2	1		
Enable	สถานะการใช้งานคำตอบ	Varchar2	1		

ตารางที่ 4.7 Test

Table Name : Test					
Description : ข้อมูลหลักข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK	
TestbankId	รหัสคลังข้อสอบ	Number	10	FK	TestbankSite
Name	ชื่อข้อสอบ	Varchar2	30		
Description	คำอธิบายข้อสอบ	Varchar2	300		
Objective	เป้าหมายข้อสอบ	Varchar2	100		
TestType	ชนิดข้อสอบ	Varchar2	1		
QuestionSource	วิธีการตั้งคำถามจากคลังคำถาม	Varchar2	1		
QuestionOrder	วิธีการเรียงลำดับคำถามในข้อสอบ	Varchar2	1		
AnswerOrder	วิธีการเรียงลำดับคำตอบ	Varchar2	1		
NoofQuestion	จำนวนคำถามในข้อสอบ	Number	4		
ScoringOption	ประเภทการให้คะแนน	Varchar2	1		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 Test(ต่อ)

FeedbackType	ชนิดการโต้ตอบเมื่อทำ ข้อสอบเสร็จ	Varchar2	1		
MaxNoofAttempts	จำนวนครั้งสูงสุดที่ สามารถเข้าสอบ	Number	3		
TimeAllowed	กำหนดเวลาที่ใช้ในการ สอบ	Number	4		
TimeRemaining Alert	เวลาที่แจ้งเตือนเมื่อเหลือ เวลาในการสอบ	Number	4		
BeginTestPeriod	วันเวลาเริ่มต้นที่อนุญาต ให้สอบ	Date			
EndTestPeriod	วันเวลาสิ้นสุดที่อนุญาต ให้สอบ	Date			

ตารางที่ 4.8 TestLevel

Table Name : TestLevel					
Description : ข้อมูลหลักการกำหนดจำนวนคำถามแต่ละระดับในข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK,FK	Test
Level	ระดับความยากง่ายของ คำถาม	Varchar2	1	PK	
No_of_Question	จำนวนคำถามในแต่ละ ระดับ	Number	3		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 TestGrade

Table Name : TestGrade					
Description : ข้อมูลหลักการกำหนดเกรดในข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK,FK	Test
Grade	เกรดที่จะได้เมื่อได้คะแนนสูงกว่าเอททรี บิวต์ Score	Varchar2	10	PK	
Score	คะแนน	Number	3		

ตารางที่ 4.10 TestScore

Table Name : TestScore					
Description : ข้อมูลหลักผลการสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK,FK	Test
UserId	รหัสนักเรียน	Varchar2	10	PK,FK	UserSite
Score	คะแนนที่สอบได้	Number	3		
Grade	เกรดที่ได้	Varchar2	10		

ตารางที่ 4.11 TestFixedQuestion

Table Name : TestFixedQuestion					
Description : ข้อมูลหลักคำถามในแต่ละข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK,FK	Test
QuestionId	รหัสคำถาม	Number	10	PK,FK	Question

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 TakenTest

Table Name : TakenTest					
Description : ข้อมูลหลักการทำข้อสอบ					
Attribute Name	Contents	Type	Length	Key	Foreign Key Referenced Table
TestId	รหัสข้อสอบ	Number	10	PK,FK	Test
QuestionId	รหัสคำถาม	Number	10	PK,FK	Question
UserId	รหัสนักเรียน	Varchar2	10	PK,FK	UserSite
AnswerValue	ข้อความคำตอบ	Varchar2	500		



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบงาน

เมื่อผ่านขั้นตอนการวิเคราะห์และออกแบบระบบเรียบร้อยแล้ว ขั้นตอนต่อไปในการพัฒนาระบบจัดการคลังข้อสอบและการสอบ ก็คือนำสิ่งที่ได้ออกแบบมาทำการสร้างเป็นแอปพลิเคชันให้สามารถใช้งานได้ตามต้องการ

5.1 การพัฒนาแอปพลิเคชันใช้งาน

ในการพัฒนาแอปพลิเคชันระบบจัดการคลังข้อสอบและการสอบนั้น ได้ทำการสร้างแบบฟอร์มต่างๆในการเพิ่ม, ปรับปรุงแก้ไข, และลบข้อมูลต่างๆในระบบเช่นข้อมูลของ Site, คลังข้อสอบ, ข้อสอบ โดยเครื่องมือที่ใช้ในการพัฒนามีดังนี้คือ

- (1) ระบบจัดการฐานข้อมูล Oracle Version 8.1.7
- (2) เครื่องมือสำหรับสร้างหน้าจอการทำงานของแอปพลิเคชัน คือ Macromedia Dreamweaver 4.0

แอปพลิเคชันระบบจัดการคลังข้อสอบและการสอบจะถูกพัฒนาและเก็บไว้ภายในฐานข้อมูล Oracle ในลักษณะของ Stored Procedure ซึ่งเมื่อมีการเริ่มทำงานก็จะมีการติดต่อมายังฐานข้อมูลเพื่อให้ Stored Procedure ทำงาน และแสดงหน้าจอติดต่อการทำงานไปยัง Browser ของผู้ใช้ แต่การจะใช้งานในลักษณะดังกล่าว จะต้องมีการติดตั้ง PL/SQL Gateway ลงไปยังฐานข้อมูลเสียก่อน โดย PL/SQL Gateway จะเป็นตัวเชื่อมต่อระหว่าง Web Server และฐานข้อมูล ทำให้สามารถส่งการทำงานไปมาได้

จากที่กล่าวมาข้างต้นว่าการทำงานทุกส่วนของแอปพลิเคชันจะเก็บในฐานข้อมูลในลักษณะ Stored Procedure นั้น แต่ก็ยังสามารถแบ่งการทำงานของแอปพลิเคชันเป็นส่วนหลักๆได้ดังนี้

- (1) การทำงานในส่วนการจัดเก็บข้อมูลของระบบ ในส่วนนี้จะพัฒนาด้วยภาษา PL/SQL และเนื่องจาก Stored Procedure ก็ถูกสร้างด้วย PL/SQL เช่นกัน ทำให้การทำงานดังกล่าวสามารถทำงานได้อย่างรวดเร็ว เพราะเป็นการทำงานภายในตัวฐานข้อมูลเอง
- (2) การทำงานในส่วนการแสดงผลหน้าจอของระบบ ในส่วนนี้จะพัฒนาด้วยภาษา HTML และ JavaScript โดยจะพัฒนาด้วย Macromedia Dreamweaver 4.0 ก่อนจะนำมาเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังฐานข้อมูล แต่ในตัวอย่างข้อมูลไม่สามารถใช้ภาษาเหล่านี้ในการพัฒนาได้โดยตรง ดังนั้นจะต้องมีการใช้ Built-In Package ที่ชื่อว่า HTP (เป็นชุด Subprogram ที่มีการติดตั้งมาพร้อมกับฐานข้อมูล) เพื่อทำการเรียกใช้อีกครั้งหนึ่ง

5.2 ขั้นตอนและวิธีการพัฒนาแอปพลิเคชัน

5.2.1 ติดตั้งฐานข้อมูล Oracle Version 8.1.7

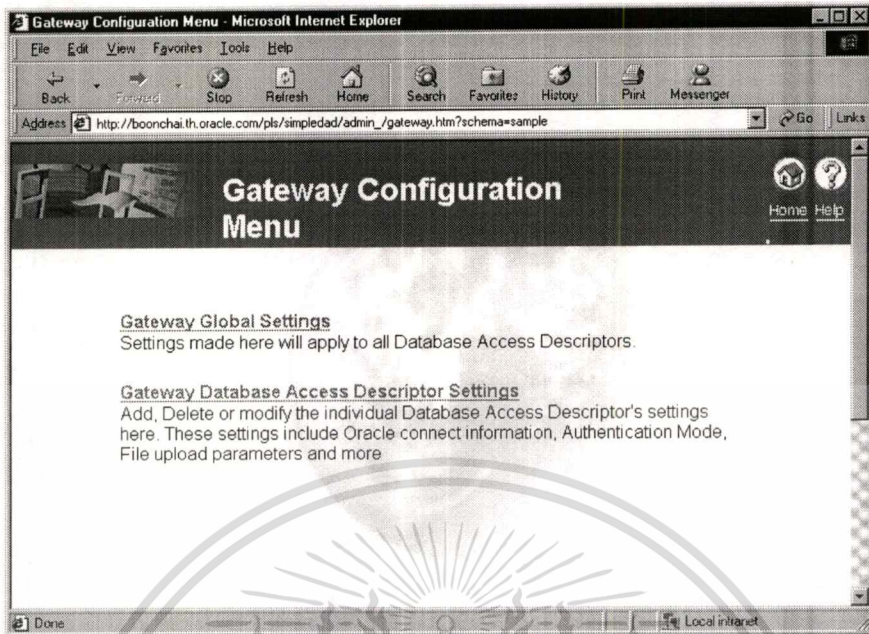
ให้ทำการติดตั้งฐานข้อมูล Oracle Version 8.1.7 เพื่อใช้เก็บข้อมูลทั้งในส่วนของแอปพลิเคชันและของระบบ จากนั้นทำการติดตั้ง PL/SQL Gateway โดยการ Run Script File ชื่อ ooload.sql ซึ่งเก็บอยู่ในไดเรกทอรี <IAS_ROOT>/Apache/modplsql/owa(IAS_ROOT คือ ไดเรกทอรีที่มีการติดตั้งฐานข้อมูล)

5.2.2 กำหนดการติดต่อในส่วน PL/SQL Gateway

เมื่อติดตั้ง PL/SQL Gateway เรียบร้อยแล้ว ขั้นตอนต่อไปก็คือการกำหนดส่วนการติดต่อจาก Web Server ผ่าน PL/SQL Gateway ไปยังฐานข้อมูล โดยจะต้องสร้าง DAD(Database Access Descriptor) เพื่อระบุวิธีการเรียกใช้ Stored Procedure สำหรับการสร้าง DAD ให้ไปยัง Gateway Configuration Menu ด้วยการระบุ URL ดังนี้

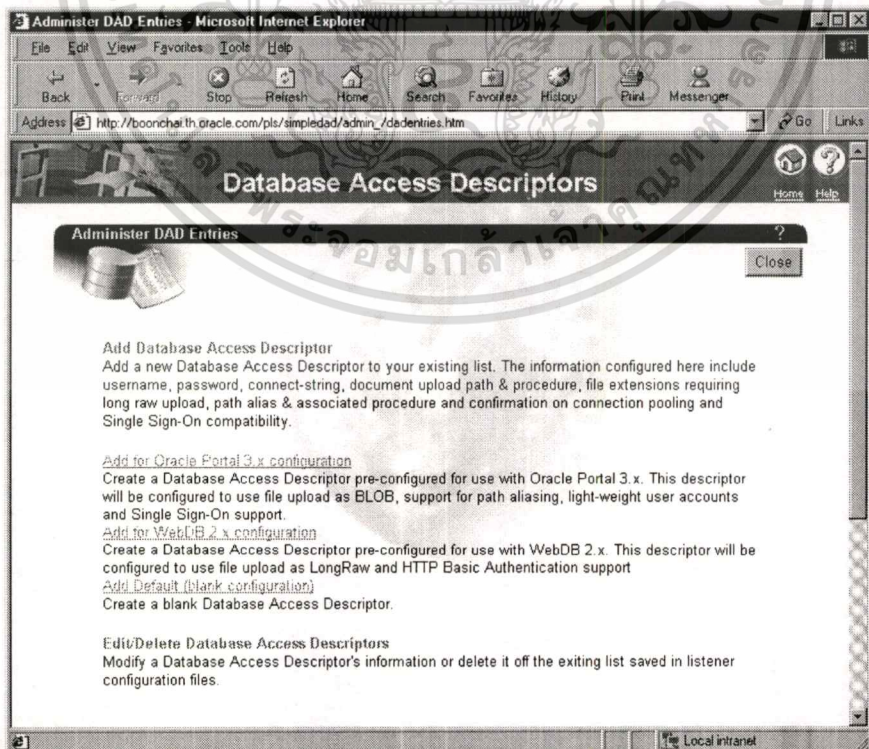
http://<hostname>:<port>/pls/admin/_gateway.htm

โดยที่ hostname คือชื่อเครื่องที่มีการติดตั้งฐานข้อมูล ส่วน port จะเป็นหมายเลข 80 แล้วแต่การกำหนดตอนการติดตั้งฐานข้อมูล เมื่อระบุ URL แล้ว Browser จะแสดงหน้าจอตั้งรูปที่ 5.1



รูปที่ 5.1 หน้าจอในส่วน Gateway Configuration Menu

จากรูปที่ 5.1 ให้คลิกที่ Gateway Database Access Descriptor Settings เพื่อเข้าสู่หน้าจอการเลือกวิธีการสร้าง DAD ดังรูปที่ 5.2



รูปที่ 5.2 หน้าจอส่วนการเลือกวิธีการสร้าง DAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้คลิกเลือกลักษณะ DAD ที่ต้องการสร้าง สำหรับการพัฒนานี้จะสร้าง DAD สำหรับ Portal ดังนั้นให้คลิกที่ Add for Oracle Portal 3.x configuration เพื่อทำการสร้าง DAD ดังรูปที่ 5.3

รูปที่ 5.3 หน้าจอในส่วนการสร้าง DAD

สำหรับการสร้าง DAD นั้น ข้อมูลที่จำเป็นต้องระบุได้แก่

- Database Access Descriptor Name: ชื่อ DAD ที่ใช้ในการติดต่อระหว่าง Web Server และฐานข้อมูล
- Schema Name: ชื่อเจ้าของ DAD
- Oracle User Name: ชื่อผู้ใช้ในการติดต่อกับฐานข้อมูล
- Oracle Password: รหัสผ่านของผู้ใช้ในการติดต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกมัดให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Oracle Connect String: ชื่อฐานข้อมูลที่ติดต่อ สำหรับข้อมูลที่เหลือนอกจากนั้น ไม่ต้องกำหนดค่าใหม่ ให้ใช้ค่าเดิมที่ระบบกำหนดมา เมื่อสร้าง DAD เสร็จเรียบร้อยแล้ว ก็จะสามารถสร้างแอปพลิเคชันจากฐานข้อมูลได้

5.2.3 การพัฒนาแอปพลิเคชัน

ในการพัฒนาแอปพลิเคชันระบบจัดการคลังข้อสอบและการสอบสามารถแบ่งเป็น 2 ระบบย่อยได้แก่

5.2.3.1 แอปพลิเคชันส่วนการจัดการคลังข้อสอบและการสอบ ผู้สอนจะเป็นผู้ใช้งานในแอปพลิเคชันส่วนนี้ ดังนั้นในส่วนนี้จะประกอบด้วย

- Site Management เป็นการพัฒนาในส่วนการจัดการดูแล Site ในส่วนนี้จะทำงานในการปรับปรุงข้อมูลทั่วไปของ Site เช่นคำอธิบาย Site, ชื่อผู้สอนและรหัสผ่าน
- User Management เป็นการพัฒนาในส่วนการจัดการนักเรียน ในส่วนนี้จะทำงานในการสร้าง, ลบและปรับปรุงแก้ไขข้อมูลของนักเรียนใน Site เช่นชื่อนักเรียนและรหัสผ่าน
- Test Bank Management เป็นการพัฒนาในส่วนการจัดการคลังข้อสอบ ในส่วนนี้จะทำงานในการสร้าง, ลบและปรับปรุงแก้ไขข้อมูลของคลังข้อสอบใน Site โดยสามารถสร้างคลังข้อสอบได้หลายๆคลังข้อสอบ และสามารถสร้างคำถามเก็บไว้ภายในคลังข้อสอบได้ สำหรับคำถามที่สามารถสร้างได้จะมีอยู่ 3 ประเภทคือ Fill in The Blank, Single Answer, Multiple Answers คำถามทั้ง 3 ประเภทสามารถกำหนดคุณสมบัติหลักๆ ได้ดังนี้
 - ตัวข้อความคำถาม(Question Text)
 - ระดับความยากของคำถาม(Level) มีให้เลือกได้ 3 ระดับ Easy, Medium และ Difficult
 - คำตอบ(Answers Values) สำหรับคำถามประเภท Fill in The Blank ข้อมูลในส่วนนี้จะถือเป็นคำตอบที่ถูกต้องทั้งหมด ขณะที่คำถามประเภท Single Answer และ Multiple Answers จะต้องมีการระบุคำตอบที่ถูกต้องด้วย แต่คำถามทั้ง 3 ประเภทสามารถมีคำตอบได้หลายๆคำตอบใน 1 คำถาม แต่ไม่จำเป็นต้องเป็นคำตอบที่ถูกต้องทั้งหมด(ยกเว้น Fill in The Blank ที่กล่าวมาแล้ว) นอกจากนี้สำหรับ Single Answer จะมีคำตอบที่ถูกต้องได้เพียง 1 คำตอบเท่านั้น(ตัวแอปพลิเคชันจะกำหนดให้เลือกคำตอบที่ถูกต้องได้เพียง 1 คำตอบ กรณีที่เป็นคำถามประเภท Single Answer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามทั้ง 3 ประเภทมีความแตกต่างกันที่ลักษณะการตอบคำถาม แต่เกณฑ์การให้คะแนน, การสุ่มคำถามจากข้อสอบจะ ไม่มีความแตกต่าง

- Test Management เป็นการพัฒนาในส่วนการจัดการข้อสอบ ในส่วนนี้จะแบ่งการทำงานเป็นส่วนย่อยๆ ดังนี้
 - การสร้าง, ลบและปรับปรุงแก้ไขข้อมูลของข้อสอบใน Site ข้อสอบที่สร้างจะมีคุณสมบัติหลักๆ ดังนี้
 - ประเภทข้อสอบ(Test Type) มี 2 ประเภทคือ Auto Grade ที่ตัวข้อสอบจะสามารถตรวจข้อสอบได้โดยอัตโนมัติและ Manual Grade ที่ผู้สอนจะต้องตรวจข้อสอบด้วยตัวเอง
 - คลังข้อสอบ(Test Bank) เป็นการกำหนดคลังข้อสอบที่จะนำคำถามมาใช้ภายในข้อสอบ
 - แหล่งคำถาม(Question Source) มี 2 ประเภทคือ Pre-Selected ที่จะให้ผู้สอนกำหนดคำถามในข้อสอบก่อน ซึ่งจะทำให้นักเรียนได้รับข้อสอบที่เหมือนกันทุกคนและ อีกประเภทคือ Random ที่จะมีการกำหนดคำถามในข้อสอบด้วยระบบ โดยจะมีการสุ่มคำถามจากคลังข้อสอบใหม่ทุกครั้งที่มีนักเรียนเข้าสอบ ไม่ว่าจะเป็นคำถามประเภทใดใน 3 ประเภทก็ตาม(สำหรับการกำหนดประเภทของคำถามอยู่ในส่วนของ Test Bank Management)
 - ลำดับคำถาม(Question Order) มี 2 ประเภทคือ Fixed ที่คำถามจะเรียงลำดับเหมือนกันทุกครั้งที่เข้าสอบและ Random ที่คำถามจะมีการสลับลำดับทุกครั้งที่เข้าสอบ
 - ลำดับคำตอบ(Answer Order) มี 2 ประเภทคือ Fixed ที่คำตอบในแต่ละคำถามจะเรียงลำดับเหมือนกันทุกครั้งที่เข้าสอบและ Random ที่คำตอบในแต่ละคำถามจะมีการสลับลำดับทุกครั้งที่เข้าสอบ
 - จำนวนคำถาม(Number of Question) ที่มีในข้อสอบ จะต้องระบุถ้าหากในส่วนแหล่งคำถาม(Question Source) กำหนดค่าเป็น Random
 - อัตราส่วนของจำนวนคำถามแต่ละระดับในข้อสอบ(Test Level) จะต้องระบุถ้าหากในส่วนแหล่งคำถาม(Question Source) กำหนดค่าเป็น Random ในส่วนนี้จะเป็นการกำหนดอัตราส่วนว่าข้อสอบที่สร้างจะให้มียจำนวนคำถามในแต่ละระดับกี่ข้อ ตัวอย่างเช่นมีคำถามระดับง่าย 30%, ระดับกลาง 40%, ระดับยาก 30% จากนั้นระบบจะคำนวณและกำหนดจำนวนคำถามในแต่ละระดับให้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัต โนมัตติ(สำหรับการกำหนดระดับความยากง่ายของคำถามอยู่ในส่วนของ Test Bank Management)

- การให้คะแนน(Scoring Option) เป็นการกำหนดวิธีการให้คะแนนของข้อสอบ มี 2 ประเภทคือ Not Scored คือเป็นข้อสอบที่ไม่คิดคะแนนหลังการสอบ และ Scored ที่จะมีการคิดคะแนนหลังการสอบ สำหรับการคิดคะแนนจะคิดเป็น เปอร์เซ็นต์ของจำนวนคำถามที่ตอบถูกต้องเมื่อเทียบกับจำนวนคำถามทั้งหมด ในข้อสอบ สำหรับคำถามที่นำมาจากคลังข้อสอบทั้ง 3 ประเภท(Fill in The Blank, Single Answer, Multiple Answers) จะไม่มีความแตกต่างกันเมื่อนำมาใช้ ในการคิดคะแนน
- การกำหนดเกรด(Grade Definition) เมื่อสอบได้คะแนนเกินที่กำหนด ข้อสอบ 1 ชุดจะสามารถกำหนดเกรดได้หลายๆเกรดในระดับคะแนนที่แตกต่างกัน แต่ถ้าส่วนการให้คะแนนกำหนดเป็น Not Scored ก็จะไม่นำส่วนการกำหนดเกรด ไปใช้งาน
- จำนวนครั้งที่สามารถเข้าสอบ(Maximum Number of Attempts) หากไม่กำหนด นักเรียนจะเข้าสอบกี่ครั้งก็ได้
- เวลาที่ใช้ในการสอบ(Time Allowed) กำหนดเป็นนาที ถ้าไม่กำหนดจะไม่มี การจำกัดเวลาในการสอบ
- วันเริ่มต้นที่กำหนดให้สามารถเข้าสอบ(Begin Test Period) ระบุรูปแบบวันที่ เป็น วัน/เดือน/ปี ถ้าไม่กำหนดนักเรียนจะเข้าสอบวันไหนก็ได้
- วันสุดท้ายที่กำหนดให้สามารถเข้าสอบ(End Test Period) ระบุรูปแบบวันที่เป็น วัน/เดือน/ปี ถ้าไม่กำหนดนักเรียนจะเข้าสอบวันไหนก็ได้
- การกำหนดนักเรียนที่จะสามารถเข้าสอบ ให้เลือกรายชื่อนักเรียนที่ต้องการให้เข้า สอบในข้อสอบแต่ละชุด(รายละเอียดการจัดการในส่วนนักเรียนอยู่ในส่วนของ User Management)
- การตรวจข้อสอบ จะใช้กรณีข้อสอบประเภท Manual Grade ที่ต้องให้ผู้สอนตรวจ ด้วยตัวเอง หากเป็นข้อสอบประเภท Auto Grade จะไม่มีการใช้งานในส่วนนี้
- การดูผลการสอบ ของแต่ละข้อสอบว่ามีนักเรียนคนใดเข้าสอบบ้าง ในวันและเวลา ใด ได้คะแนนและเกรดอะไร

5.2.3.2 แอปพลิเคชันส่วนการเข้าสอบ นักเรียนจะเป็นผู้ใช้งานในแอปพลิเคชันส่วนนี้

ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Take Test เป็นการเข้าสอบในแต่ละข้อสอบที่ต้องการ นักเรียนจะสามารถเลือกสอบเฉพาะข้อสอบที่ผู้สอนกำหนดให้เท่านั้น
- View Score เป็นการดูผลการสอบในแต่ละข้อสอบ แต่จะดูได้เฉพาะคะแนนและเกรดของตัวเอง

5.2.4 ส่วนแอปพลิเคชันที่ยังไม่ได้พัฒนา

จากการออกแบบระบบจนถึงขั้นตอนการพัฒนาแอปพลิเคชันนั้น ยังขาดส่วนการทำงานบางส่วนที่ยังไม่ได้พัฒนาในปัจจุบันดังนี้

- ประเภทการตอบสนองของข้อสอบ(Feedback Type) เป็นการระบุการตอบสนองของข้อสอบเมื่อนักเรียนทำข้อสอบเสร็จว่าจะให้มีการเฉลยคำตอบหลังการสอบหรือไม่ ในส่วนงานนี้ยังรวมไปถึงการกำหนดข้อความในแต่ละคำถามที่จะแสดงเมื่อตอบคำถามถูกหรือผิดอีกด้วย

5.2.5 ปัญหาที่พบในการพัฒนาด้วย Java Stored Procedure

เมื่อเริ่มต้นพัฒนาแอปพลิเคชันนั้น ได้มีการพัฒนาด้วย Java Stored Procedure แต่การพัฒนาด้วย Java Stored Procedure นั้นมีความล่าช้า เนื่องจากประสบปัญหาต่างๆดังนี้

- ตัวภาษา Java มีความซับซ้อนและต้องใช้เวลาในการเรียนรู้จนจึงจะสามารถใช้งานได้
- การสร้าง Stored Procedure ด้วย Java ต้องมีการติดตั้งในส่วนของ JVM(Java Virtual Machine) ลงไปในฐานข้อมูลและกำหนดรายละเอียดต่างๆเพิ่มเติม ซึ่งหากกำหนดรายละเอียดไม่ครบ จะทำให้การสร้าง Java Stored Procedure เกิดปัญหาได้
- ตัว Java Stored Procedure ไม่สามารถเรียกใช้จาก Web Server ได้โดยตรง จะต้องมีการสร้างส่วน Header ด้วย PL/SQL เสียก่อน

จากปัญหาดังกล่าว ทำให้การพัฒนาแอปพลิเคชันด้วย Java Stored Procedure มีการเปลี่ยนการพัฒนามาเป็น PL/SQL Stored Procedure แทน เนื่องจาก PL/SQL เป็นภาษาที่มีการใช้งานในฐานข้อมูล Oracle มาแต่เดิม จึงสามารถสร้างและเรียกใช้ PL/SQL Stored Procedure ได้โดยตรง

5.3 การทดสอบระบบ

ในการทดสอบระบบนั้น ได้กำหนดผู้ใช้งานระบบไว้ 2 กลุ่มคือ ผู้สอนและนักเรียน โดยการทดสอบมีลำดับดังนี้

ส่วนการใช้งานของผู้สอน

1. ผู้สอนสร้าง Site ขึ้นมาภายในระบบ กำหนดรายละเอียดในส่วนชื่อผู้สอน รหัสผ่าน
2. ผู้สอนเพิ่มรายชื่อนักเรียนเข้าไปใน Site กำหนดรายละเอียดในส่วนชื่อนักเรียน รหัสผ่าน
3. ผู้สอนสร้างคลังข้อสอบ จากนั้นทำการสร้างคำถามจำนวน 50 ข้อเข้าไปในคลังข้อสอบ โดยคำถามที่สร้างจะประกอบด้วย
 - คำถามประเภท Fill in The Blank 10 ข้อ แบ่งเป็นระดับ Easy 5 ข้อ, ระดับ Medium 3 ข้อ, ระดับ Difficult 2 ข้อ
 - คำถามประเภท Single Answer 20 ข้อ แบ่งเป็นระดับ Easy 10 ข้อ, ระดับ Medium 5 ข้อ, ระดับ Difficult 5 ข้อ
 - คำถามประเภท Multiple Answers 20 ข้อ แบ่งเป็นระดับ Easy 10 ข้อ, ระดับ Medium 7 ข้อ, ระดับ Difficult 3 ข้อ
4. ผู้สอนสร้างข้อสอบ 4 ข้อสอบ แต่ละข้อสอบมีลักษณะดังนี้
 - ข้อสอบ A 20 ข้อ ให้ผู้สอนเป็นผู้ระบุคำถามที่ใช้ในข้อสอบ ให้ทำการตรวจข้อสอบโดยอัตโนมัติ กำหนดคะแนนและเกรดที่จะได้รับ ไม่จำกัดเวลาและจำนวนครั้งที่เข้าสอบ
 - ข้อสอบ B 30 ข้อ ให้ระบบสุ่มคำถามให้ข้อสอบ เป็นคำถามระดับง่าย 50%, ระดับกลาง 30%, ระดับยาก 20% ให้ทำการตรวจข้อสอบโดยอัตโนมัติ กำหนดคะแนนและเกรดที่จะได้รับ จำกัดเวลาในการสอบ 10 นาทีและให้สอบได้เพียง 1 ครั้ง
 - ข้อสอบ C 5 ข้อ ให้ผู้สอนเป็นผู้ระบุคำถามที่ใช้ในข้อสอบ และการตรวจข้อสอบโดยผู้สอน จำกัดเวลาในการสอบ 90 นาทีและให้สอบได้เพียง 1 ครั้ง
 - ข้อสอบ D 10 ข้อ ให้ระบบสุ่มคำถามให้ข้อสอบ เป็นคำถามระดับง่าย 50%, ระดับกลาง 30%, ระดับยาก 20% และการตรวจข้อสอบโดยผู้สอน จำกัดเวลาในการสอบ 90 นาทีและให้สอบได้เพียง 1 ครั้ง
5. ผู้สอนกำหนดรายชื่อนักเรียนให้สามารถเข้าสอบข้อสอบแต่ละชุด
6. ผู้สอน Logout ออกจากระบบ

ส่วนการใช้งานของนักเรียน

1. นักเรียนทำการเข้าสู่ระบบด้วยชื่อและรหัสผ่านที่ผู้สอนจัดให้
2. นักเรียนเลือกข้อสอบและทำการสอบข้อสอบแต่ละชุด
3. นักเรียนดูผลการสอบจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นักเรียน Logout ออกจากระบบ

5.4 ผลการทดสอบระบบ

สำหรับผลการทดสอบระบบมีรายละเอียดดังนี้

ส่วนการใช้งานของผู้สอน

- การสร้าง Site สามารถสร้างและเก็บข้อมูลส่วนต่างๆ ได้ถูกต้อง ระบบมีการตรวจสอบ หากไม่ระบุชื่อผู้สอนและรหัสผ่านที่ใช้ในการเข้าสู่ระบบ จะมีการแจ้งข้อความแสดงข้อผิดพลาดที่หน้าจอ
- การเพิ่มรายชื่อนักเรียนสามารถสร้างและเก็บข้อมูลส่วนต่างๆ ได้ถูกต้อง ระบบมีการตรวจสอบหากไม่ระบุชื่อนักเรียนและรหัสผ่านที่ใช้ในการเข้าสู่ระบบ จะมีการแจ้งข้อความแสดงข้อผิดพลาดที่หน้าจอ
- การสร้างคลังข้อสอบและคำถามภายในคลังข้อสอบสามารถสร้างและเก็บข้อมูลส่วนต่างๆ ได้ถูกต้อง ระบบจะมีการให้ระบุประเภทของคำถามก่อน เพื่อแสดงหน้าจอการสร้างคำถามของแต่ละประเภท หากขาดข้อมูลในส่วนที่จำเป็นต่อระบบ จะมีการแจ้งข้อความแสดงข้อผิดพลาดที่หน้าจอ
- การสร้างข้อสอบสามารถสร้างและเก็บข้อมูลส่วนต่างๆ ได้ถูกต้อง แต่ขาดการตรวจสอบในส่วนความสัมพันธ์ของข้อมูล ตัวอย่างเช่นเมื่อกำหนดแหล่งคำถามเป็น Random ผู้สอนก็ยังสามารถกำหนดคำถามที่ใช้ข้อสอบ ซึ่งไม่มีความจำเป็นเพราะระบุแหล่งคำถามเป็น Random ไปแล้ว
- การกำหนดรายชื่อนักเรียนให้สามารถเข้าสอบในแต่ละข้อสอบ สามารถเก็บข้อมูลส่วนต่างๆ ได้ถูกต้อง

ส่วนการใช้งานของนักเรียน

- ข้อสอบ A สามารถทำการสอบได้ คำถามในข้อสอบเป็นไปตามที่ผู้สอนระบุเอาไว้ และแสดงคะแนนกับเกรดหลังการสอบได้ถูกต้อง สามารถเข้าสอบกี่ครั้งก็ได้ตามความต้องการ
- ข้อสอบ B สามารถทำการสอบได้ คำถามในข้อสอบมีการสุ่มมาจากคลังข้อสอบ เมื่อทำข้อสอบครบ 10 นาที ระบบจะปิดหน้าจอการสอบและทำการคิดคะแนนให้ทันที หากทำการสอบอีกครั้ง ระบบจะแจ้งข้อความให้ทราบว่าข้อสอบชุดนี้มีการสอบครบตามจำนวนครั้งที่ระบุแล้ว

- ข้อสอบ C สามารถทำการสอบได้ คำถามในข้อสอบเป็นไปตามที่ผู้สอนระบุเอาไว้ แต่เมื่อสอบเสร็จ ระบบจะแจ้งว่าข้อสอบชุดนี้ต้องตรวจคะแนนโดยผู้สอน ดังนั้นจึงไม่มีคะแนนและเกรดปรากฏหลังการสอบ หากทำการสอบอีกครั้ง ระบบจะแจ้งข้อความให้ทราบว่าข้อสอบชุดนี้มีการสอบครบตามจำนวนครั้งที่ระบุแล้ว
- ข้อสอบ D สามารถทำการสอบได้ คำถามในข้อสอบมีการสุ่มมาจากคลังข้อสอบ แต่คำถามที่สุ่มมาจะมีประเภทอื่นนอกเหนือจาก Fill in The Blank ด้วย เมื่อทำข้อสอบเสร็จ ตัวแอปพลิเคชันจะเกิดปัญหาขึ้น มีข้อความผิดพลาดแจ้งให้ทราบว่า ไม่สามารถบันทึกข้อมูลได้

จากการทดสอบระบบจะได้ข้อสรุปดังนี้

- ในการทำงานส่วนการใช้งานของผู้สอน ไม่พบปัญหามากนัก เพราะการใช้งานในส่วนนี้เป็นการเพิ่ม, ลบและปรับปรุงข้อมูลเท่านั้น ไม่มีกระบวนการทำงานที่ซับซ้อนมาก ส่วนที่ควรปรับปรุงแก้ไขคือการตรวจสอบความสัมพันธ์ของข้อมูลในระดับแอปพลิเคชัน เพื่อป้องกันไม่ให้ผู้ใช้งานระบบต้องใส่ข้อมูลที่ไม่ได้ใช้งานเข้าไปในระบบ
- ในการทำงานส่วนการใช้งานของนักเรียน จะพบปัญหาเมื่อข้อสอบเป็นประเภท Manual Grade และแหล่งคำถามเป็น Random เพราะการสร้างชุดคำถามให้กับข้อสอบประเภทนี้ควรจะเป็นประเภท Fill in The Blank เท่านั้น แต่ตัวระบบจะนำคำถามประเภทอื่นมาใช้ในข้อสอบประเภทนี้ด้วย ทำให้เกิดปัญหาในการบันทึกข้อมูลขึ้น วิธีแก้ไขคือ หากเป็นข้อสอบประเภท Manual Grade ให้ดึงคำถามประเภท Fill in The Blank ขึ้นมาเท่านั้น

บทที่ 6

บทสรุป

จากการศึกษาการวิเคราะห์และออกแบบระบบจัดการคลังข้อสอบและการสอบโดยใช้เทคโนโลยีเชิงวัตุนั้นประกอบด้วยขั้นตอนต่างๆในการพัฒนา เริ่มตั้งแต่การกำหนดวัตถุประสงค์ของการพัฒนาระบบ, การวิเคราะห์ระบบ, การออกแบบและการนำระบบไปใช้งาน โดยใช้ CASE Tool ระบบจัดการฐานข้อมูล และ Tool ต่างๆในการพัฒนาระบบ ทำให้สามารถสรุปผลการพัฒนาระบบทั้งหมดได้ดังนี้

6.1 ผลการพัฒนาระบบงาน

- (1) ในการออกแบบระบบโดยใช้ CASE Tools ช่วยให้การออกแบบระบบมีความสะดวกรวดเร็ว โดยผู้ที่เข้ามาพัฒนาระบบสามารถศึกษาระบบงานเดิมได้จากยูสเคสไดอะแกรมเพื่อพัฒนาระบบต่อไปและช่วยลดข้อผิดพลาดต่างๆที่เกิดขึ้นได้
- (2) การออกแบบระบบโดยใช้ Rational Rose 2000 จะสามารถสร้าง Code พื้นฐานของแอปพลิเคชันได้จาก Class Diagram เป็นภาษาต่างๆ ได้โดยไม่ต้องเสียเวลาในการเขียน โปรแกรมใหม่ ทำให้ลดเวลาในการพัฒนาโปรแกรมให้น้อยลง
- (3) การพัฒนาระบบโดยใช้เทคโนโลยีเชิงวัตถุ เป็นการนำทฤษฎีที่ศึกษามาใช้ในการพัฒนาระบบงานตามขั้นตอนต่างๆ ช่วยให้การพัฒนาเป็นไปตามขั้นตอนและมีรูปแบบการพัฒนาเป็นมาตรฐานทำให้การดูแลปรับปรุงมีความสะดวก
- (4) การพัฒนาแอปพลิเคชันบนเว็บ ช่วยให้มีความเข้าใจในการทำงานของ Internet ได้ชัดเจนขึ้น ทั้งในส่วนของตัว Server ตลอดจนภาษาต่างๆที่ใช้เช่น HTML, JavaScript

6.2 ประโยชน์ที่ได้รับ

จากการศึกษาระบบจัดการคลังข้อสอบและการสอบ ทำให้ทราบถึงขั้นตอนการทำงานของ การออกข้อสอบตลอดจนการให้คะแนนและการประกาศผลสอบ ซึ่งจะช่วยให้เห็นปัญหาที่เกิดขึ้นในระบบเดิมที่มีอยู่และความต้องการใหม่ๆของระบบ แล้วจึงทำการพัฒนาระบบเพื่อแก้ไขปัญหาในระบบเดิมและรองรับความต้องการใหม่ๆที่เกิดขึ้น นอกจากนี้ยังได้ความรู้จากการใช้

เครื่องมือต่างๆในการพัฒนาระบบเช่น Rational Rose 2000, Oracle 8.1.7, Macromedia Dreamweaver รวมทั้งได้เรียนรู้การพัฒนาแอปพลิเคชันบนเว็บด้วยภาษา JavaScript อีกด้วย

6.3 ข้อเสนอแนะ

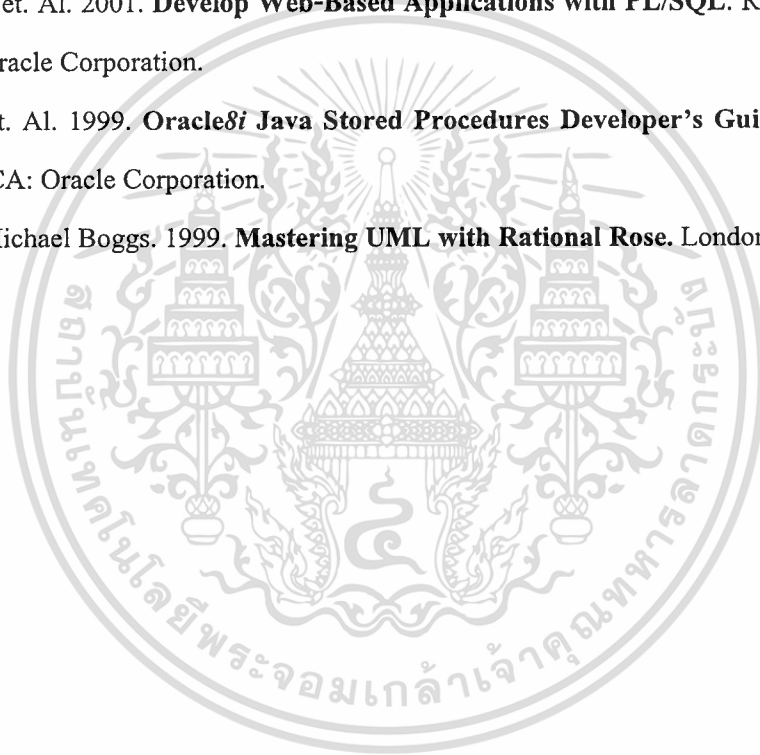
การพัฒนาระบบโดยใช้เทคโนโลยีเชิงวัตุนั้น ยังเป็นแนวทางใหม่ที่ไม่มีการใช้โดยทั่วไป แต่เป็นแนวการพัฒนาที่มีประสิทธิภาพ ช่วยให้การวิเคราะห์และออกแบบระบบใกล้เคียงกับความเป็นจริง ทำให้สามารถนำไปใช้กับงานที่มีลักษณะใกล้เคียงกันได้ ดังนั้นจึงเหมาะสำหรับระบบที่ต้องมีการเปลี่ยนแปลงอยู่เสมอ หากใช้การพัฒนาระบบแบบเดิมก็จะต้องใช้เวลาในการพัฒนานานกว่าการใช้เทคโนโลยีเชิงวัตุนั้น

นอกจากนี้การสร้างแอปพลิเคชันบนเว็บจะช่วยให้การใช้งานแอปพลิเคชันมีความสะดวกมากขึ้น สามารถใช้งานจากที่ใดก็ได้ และการดูแลแอปพลิเคชันก็มีความสะดวก เนื่องจากอยู่ที่ Server เพียงที่เดียว ไม่จำเป็นต้องติดตั้งแอปพลิเคชันที่เครื่องผู้ใช้ทุกคน แต่ก็ต้องระมัดระวังในเรื่องของความปลอดภัยรวมทั้งความเร็วของระบบเครือข่ายที่ใช้ด้วยเช่นกัน



บรรณานุกรม

- Jonathan Lewis. 2001. **Practical Oracle8i**. Upper Saddle River, NJ: Prentice Hall.
- Larman, Craig. 1997. **Applying UML and Patterns**. Upper Saddle River, NJ: Prentice Hall.
- Nigel McFarlane. et. Al. 1999. **Professional JavaScript**. Birmingham: Wrox Press.
- Quatrani, Terry. 1999. **Visual Modeling With Rational Rose 2000 and UML**. Reading, MA: Addison Wesley Longman.
- Shena Deuchars. et. Al. 2001. **Develop Web-Based Applications with PL/SQL**. Redwood City, CA: Oracle Corporation.
- Tom Portfolio. et. Al. 1999. **Oracle8i Java Stored Procedures Developer's Guide**. Redwood City, CA: Oracle Corporation.
- Wendy Boggs, Michael Boggs. 1999. **Mastering UML with Rational Rose**. London: Sybex



ภาคผนวก ก.

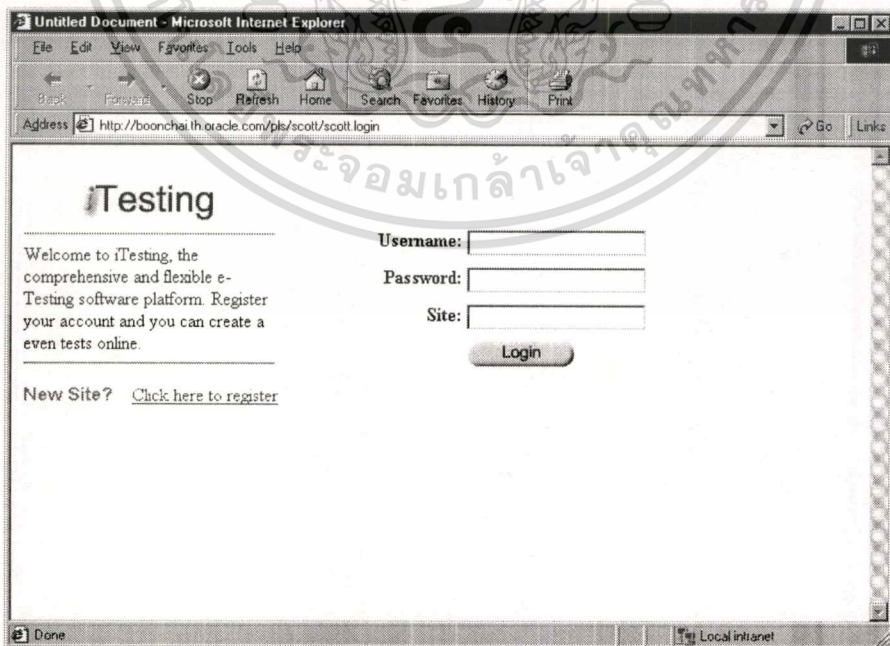
วิธีการติดตั้งและใช้งานระบบ

ก.1 วิธีการเข้าสู่ระบบ

การทำงานกับแอปพลิเคชันนี้ ผู้ใช้งานไม่มีความจำเป็นต้องติดตั้งซอฟต์แวร์ใดๆเพิ่มเติม เพื่อให้สามารถใช้งานแอปพลิเคชัน เพียงมี Browser ติดตั้งอยู่ภายในเครื่องก็จะสามารถทำงานกับแอปพลิเคชันนี้ได้ โดยขั้นตอนแรกให้เรียก Browser ทำงาน จากนั้นทำการระบุ URL ของแอปพลิเคชันระบบจัดการคลังข้อสอบและการสอบ เมื่อเข้าสู่แอปพลิเคชันจะปรากฏหน้าจอ ดังรูปที่ ก.1 ซึ่งเป็นรูปภาพที่แสดงการ Login เข้าสู่ระบบ

เมื่อปรากฏหน้าจอ Login ดังที่แสดงในรูปที่ ก.1 ผู้ใช้จะต้องระบุชื่อผู้ใช้งาน (Username), รหัสผ่าน (Password) และ Site เมื่อระบุข้อมูลถูกต้องเรียบร้อยแล้ว ให้กดปุ่ม ก็จะ สามารถเข้าสู่ระบบได้

หากเป็นผู้สอนที่ไม่เคยใช้งานระบบมาก่อนจะต้องทำการสร้าง Site ขึ้นมาใหม่ โดยต้องทำการลงทะเบียนข้อมูลเพื่อสร้าง Site จึงจะสามารถเข้าทำงานภายในระบบได้



รูปที่ ก.1 หน้าจอการ Login เข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลงทะเบียนข้อมูลเพื่อสร้าง Site ใหม่ ให้คลิกไปยัง [Click here to register](#) เพื่อเข้าสู่หน้า
จอการสร้าง Site ใหม่ดังแสดงในรูปที่ ก.2

Microsoft Internet Explorer
Address: http://boonchai.th.oracle.com/pls/soai/CreateSite

Create New Site

Instruction

Fill in the fields below and click Submit to create a new site. The fields marked with * are required. The User Name you select identifies you within iTesting and must be entered, along with the password you choose, each time you login to iTesting.

*Site Name:

Site Description:

*Username:

*Password:

*Confirm Password:

*First Name:

*Last Name:

*E-Mail:

Address:

City:

State/Province:

Zip/Postal Code:

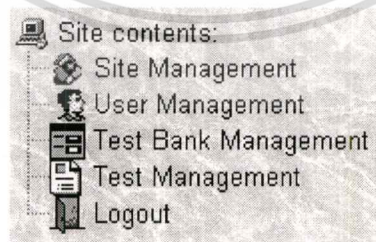
Country:

Phone Number:

รูปที่ ก.2 หน้าจอการสร้าง Site ใหม่

ก.2 รายการหลัก

รายการหลักที่ใช้สำหรับเลือกหัวข้อการทำงานในระบบจัดการคลังข้อสอบและการสอบ มี
รายการแสดงดังรูปที่ ก.3



รูปที่ ก.3 หน้าจอรายการหลักของระบบ

จากหน้ารายการหลักในการทำงาน สามารถแยกส่วนการทำงานออกเป็น 5 รายการดังนี้

- ส่วนการจัดการ Site(Site Management) ใช้สำหรับแก้ไขข้อมูลของ Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนการจัดการผู้ใช้(User Management) ใช้สำหรับเพิ่ม,ลบและแก้ไขข้อมูลนักเรียนภายใน Site
- ส่วนการจัดการคลังข้อสอบ(Test Bank Management) ใช้สำหรับเพิ่ม,ลบและแก้ไขข้อมูลคลังข้อสอบรวมทั้งคำถามในคลังข้อสอบ
- ส่วนการจัดการข้อสอบ(Test Management) ใช้สำหรับเพิ่ม,ลบและแก้ไขข้อมูลข้อสอบ
- ออกจากระบบ(Logout) ใช้เมื่อต้องการออกจากระบบ

ก.3 วิธีการใช้งานระบบจัดการคลังข้อสอบและการสอบ

เมื่อเข้าสู่หน้าจอรายการหลักของระบบเรียบร้อยแล้ว ก็จะสามารถทำรายการต่างๆที่มีในระบบดังนี้


- ส่วนการจัดการ Site(Site Management) เมื่อเลือกทำรายการจะปรากฏหน้าจอดังรูปที่

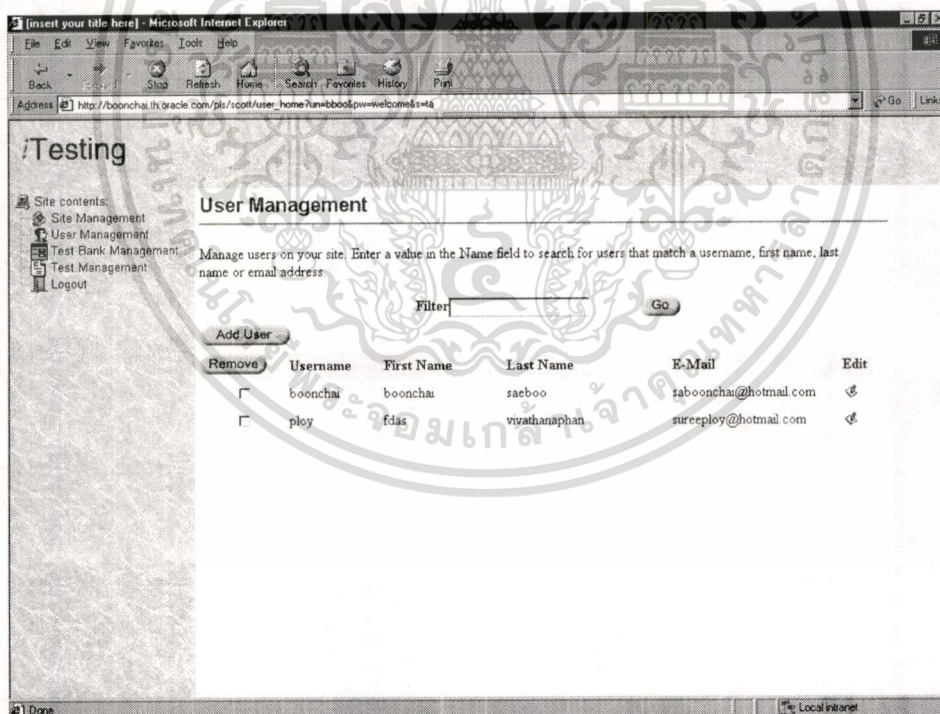
ก.4 มีขั้นตอนการทำงานดังนี้

1. ป้อนข้อมูลของ Site ที่ต้องการเปลี่ยนแปลง
2. กดปุ่ม Submit เพื่อยืนยันการเปลี่ยนแปลง หากต้องการยกเลิกให้กดปุ่ม Cancel

รูปที่ ก.4 หน้าจอการจัดการ Site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนการจัดการผู้ใช้ (User Management) ใช้สำหรับเพิ่ม,ลบและแก้ไขข้อมูลนักเรียนภายใน Site เมื่อเลือกทำรายการจะปรากฏหน้าจอดังรูปที่ ก.5 มีขั้นตอนการทำงานดังนี้
 1. ระบุชื่อนักเรียนในช่องข้อมูล Filter และกดปุ่ม Go เพื่อแสดงข้อมูลผู้ใช้เฉพาะที่ต้องการ
 2. กรณีที่ต้องการลบข้อมูลนักเรียนใน Site ให้ทำเครื่องหมาย ด้านหน้านักเรียนที่ต้องการ และทำการกดปุ่ม Remove
 3. กรณีที่ต้องการแก้ไขข้อมูลนักเรียนใน Site ให้คลิกที่  ของนักเรียนที่ต้องการ
 4. เมื่อต้องการเพิ่มข้อมูลนักเรียนคนใหม่เข้าไปใน Site ให้กดปุ่ม Add User จะปรากฏหน้าจอดังรูปที่ ก.6
 5. ป้อนข้อมูลของนักเรียนคนใหม่เข้าสู่ระบบ
 6. กดปุ่ม Save เพื่อยืนยันการเปลี่ยนแปลง หากต้องการยกเลิกให้กดปุ่ม Cancel



รูปที่ ก.5 หน้าจอการจัดการนักเรียน

Microsoft Internet Explorer
 Address: http://boonchai.th.oracle.com/pls/scott/user_home?un=bbotpw=welcome&st=

Testing

Site contents
 Site Management
 User Management
 Test Bank Management
 Test Management
 Logout

Add User


Fill in the fields below to add a user account for your site. The fields marked with * are required.

*Username:
 *Password:
 *Confirm Password:
 *First Name:
 *Last Name:
 *E-Mail:
 Address:
 City:
 State/Province:
 Zip/Postal Code:
 Country:
 Phone Number:

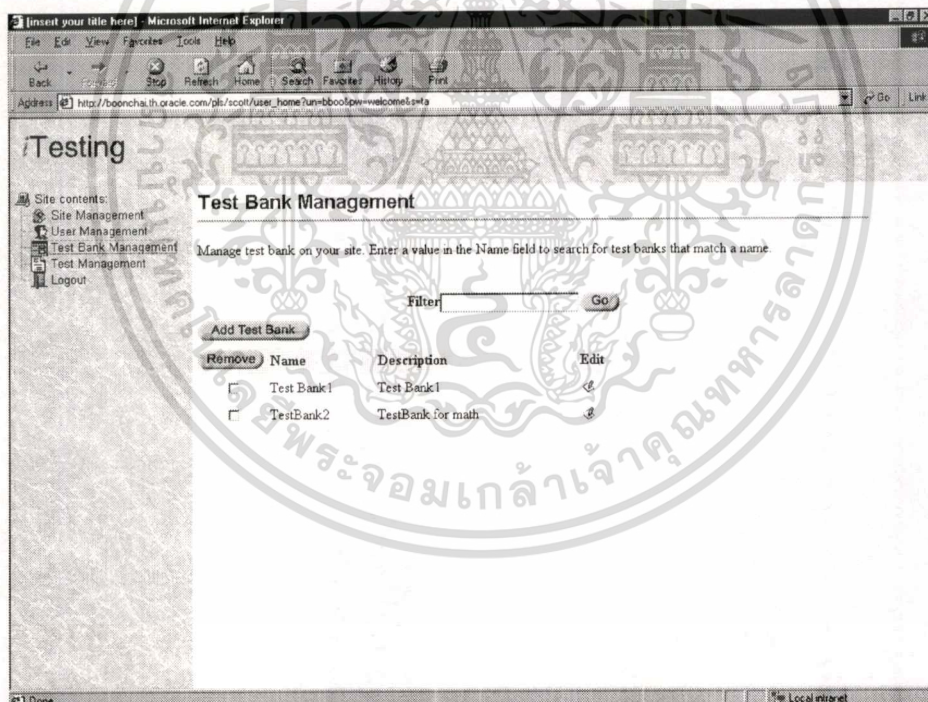
Click "Save" to finish. Click "Cancel" to abort this process.

Save Cancel

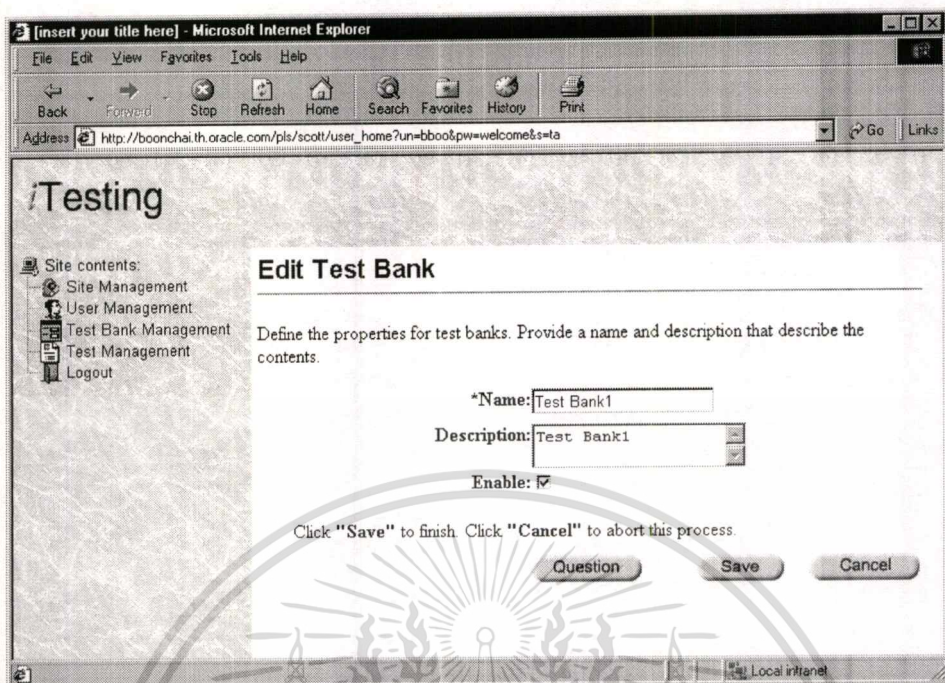
รูปที่ ก.6 หน้าจอการเพิ่มข้อมูลนักเรียน

- ส่วนการจัดการคลังข้อสอบ(**Test Bank Management**) ใช้สำหรับเพิ่ม,ลบและแก้ไข ข้อมูลคลังข้อสอบและคำถามในคลัง เมื่อเลือกทำรายการจะปรากฏหน้าจอดังรูปที่ ก.7 มีขั้นตอนการทำงานดังนี้
 1. ระบุข้อคลังข้อสอบในช่องข้อมูล Filter และกดปุ่ม Go เพื่อแสดงข้อมูลคลังข้อสอบเฉพาะที่ต้องการ
 2. กรณีที่ต้องการลบข้อมูลคลังข้อสอบใน Site ให้ทำเครื่องหมาย ด้านหน้าคลังข้อสอบที่ต้องการ และทำการกดปุ่ม Remove
 3. กรณีที่ต้องการแก้ไขข้อมูลคลังข้อสอบใน Site หรือเพิ่มคำถามในคลัง ให้คลิกที่  ของคลังข้อสอบที่ต้องการ จะปรากฏหน้าจอดังรูปที่ ก.8 เพื่อแก้ไขข้อมูลคลังข้อสอบ หากต้องการเพิ่มคำถามใหม่ให้กับคลังข้อสอบ ให้ทำตามขั้นตอนดังต่อไปนี้
 - 3.1. กดปุ่ม Question เพื่อเข้าสู่หน้าจอการจัดการคำถามดังรูปที่ ก.9
 - 3.2. กดปุ่ม Add Question เพื่อเข้าสู่หน้าจอการเพิ่มคำถามดังรูปที่ ก.10

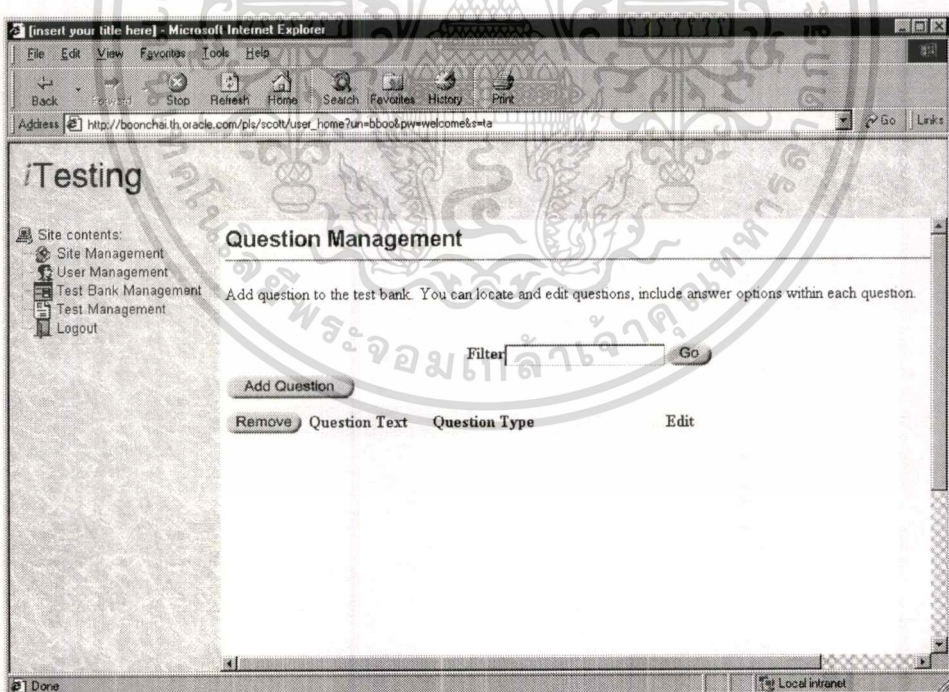
- 3.3. เลือกประเภทคำถามที่ต้องการสร้างจากที่มี 3 ประเภทได้แก่ เติมคำในช่องว่าง(Fill in The Blank), ปรนัย 1 คำตอบ(Single Choice)และปรนัยหลายคำตอบ(Multiple Choice) จากนั้นกดปุ่ม Next จะปรากฏหน้าจอจดังรูปที่ ก.11
- 3.4. ป้อนข้อมูลคำถามใหม่เข้าสู่คลังข้อสอบได้แก่ ตัวคำถาม(Question Text), ระดับความยากของคำถาม(Level), คำตอบต่างๆ(Answers Values)
- 3.5. กดปุ่ม Save เพื่อยืนยันการเปลี่ยนแปลง หากต้องการยกเลิกให้กดปุ่ม Cancel
4. เมื่อต้องการเพิ่มข้อมูลคลังข้อสอบใหม่เข้าไปใน Site ให้กดปุ่ม Add Test Bank จะปรากฏหน้าจอจดังรูปที่ ก.12
5. ป้อนข้อมูลคลังข้อสอบใหม่เข้าสู่ระบบ
6. กดปุ่ม Save เพื่อยืนยันการเปลี่ยนแปลง หากต้องการยกเลิกให้กดปุ่ม Cancel



รูปที่ ก.7 หน้าจอการจัดการคลังข้อสอบ

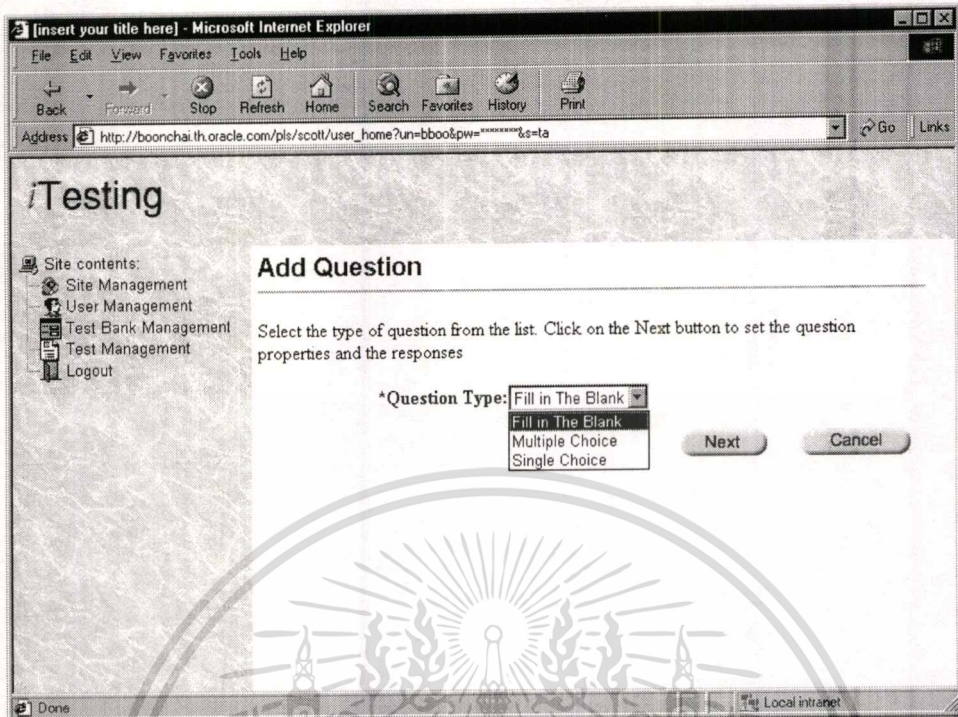


รูปที่ ก.8 หน้าจอการแก้ไขคลังข้อสอบ

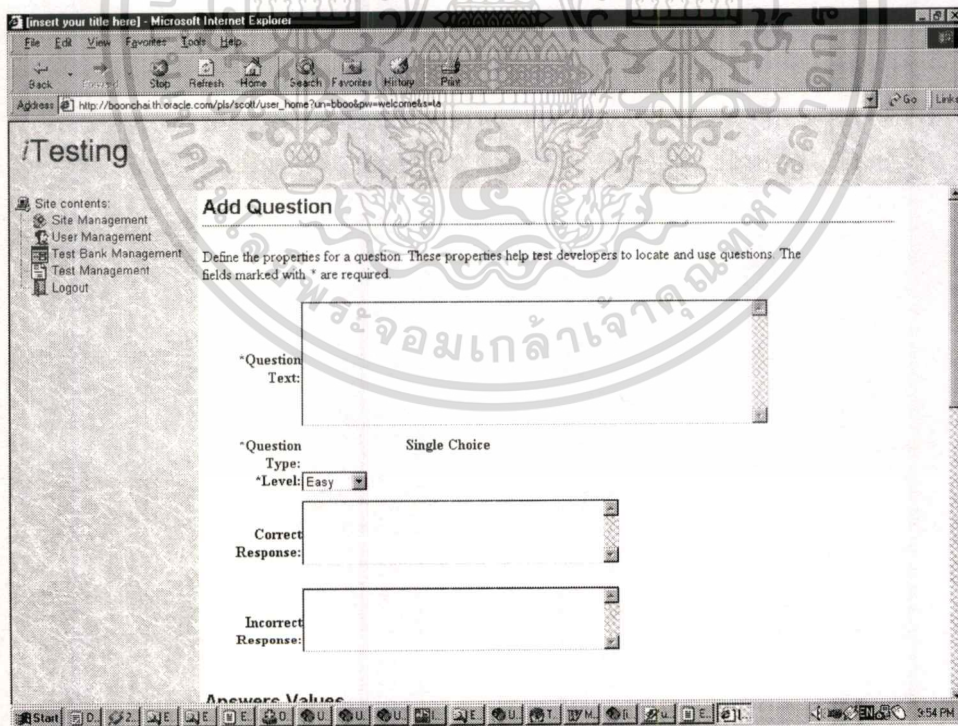


รูปที่ ก.9 หน้าจอการจัดการคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

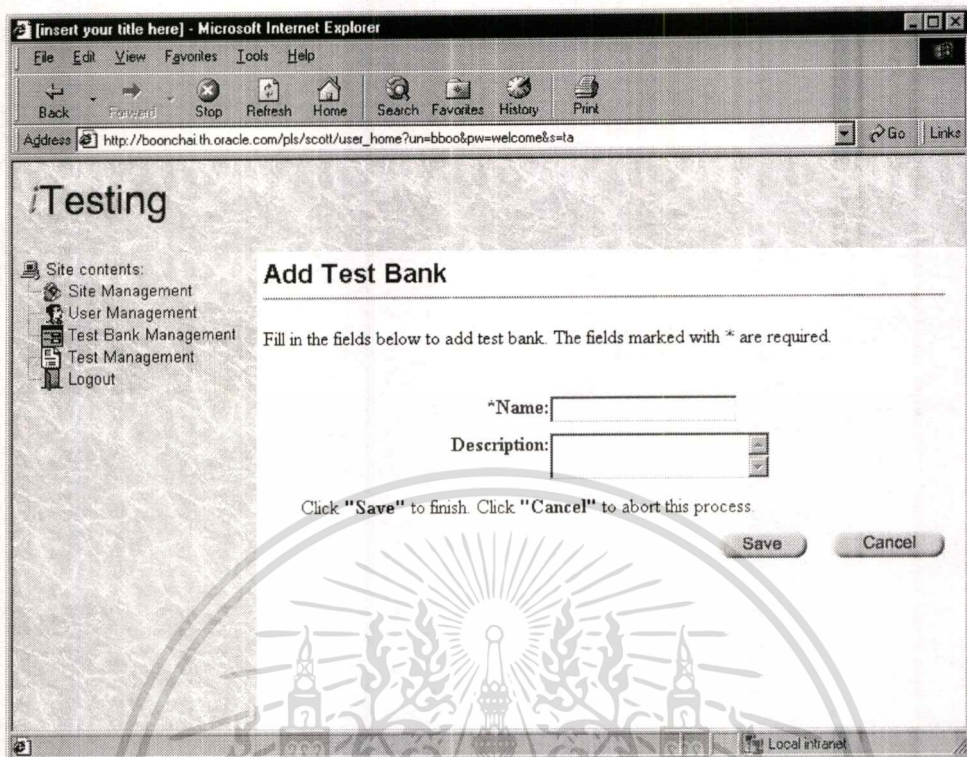


รูปที่ ก.10 หน้าจอการเพิ่มคำถาม







รูปที่ ก.11 หน้าจอการป้อนข้อมูลคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

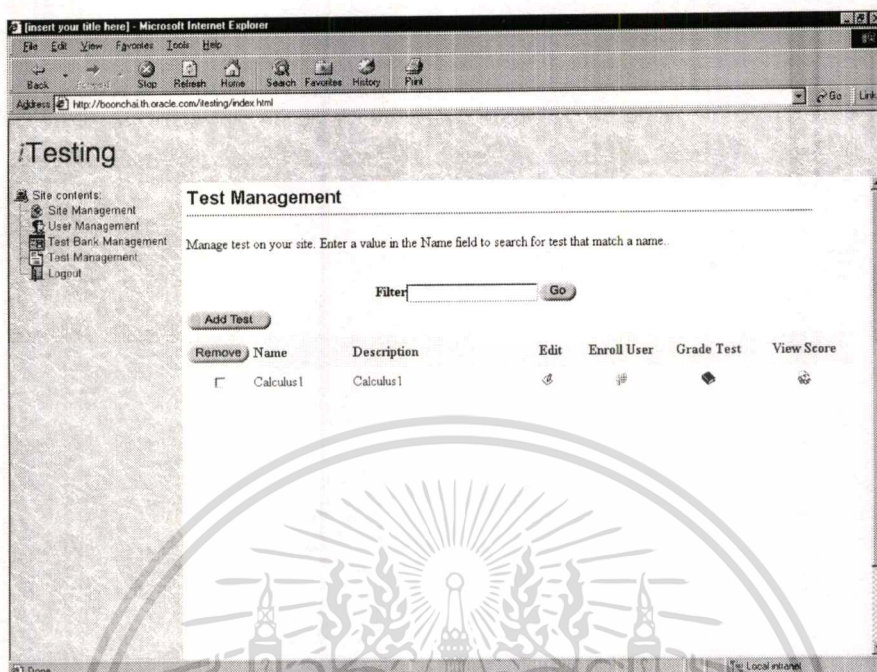


รูปที่ ก.12 หน้าจอการเพิ่มคลังข้อสอบ

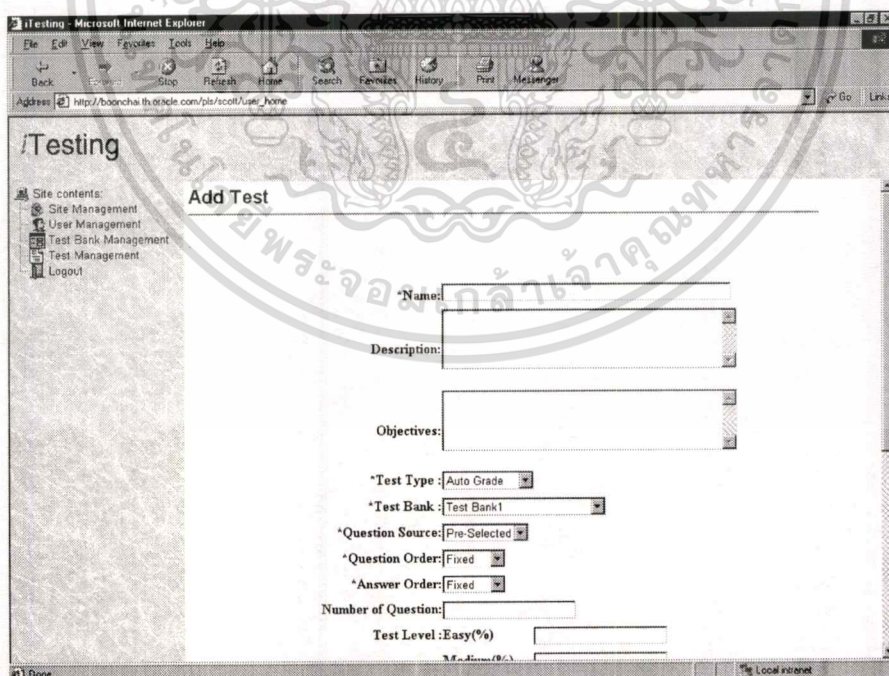
- ส่วนการจัดการข้อสอบ(Test Management) ใช้สำหรับเพิ่ม,ลบและแก้ไขข้อสอบ เมื่อเลือกทำรายการจะปรากฏหน้าจอจดังรูปที่ ก.13 มีขั้นตอนการทำงานดังนี้
 1. ระบุชื่อข้อสอบในช่องข้อมูล Fiter และกดปุ่ม Go เพื่อแสดงข้อสอบเฉพาะที่ต้องการ
 2. กรณีที่ต้องการลบข้อสอบใน Site ให้ทำเครื่องหมาย ด้านหน้าข้อสอบที่ต้องการ และทำการกดปุ่ม Remove
 3. กรณีที่ต้องการแก้ไขข้อสอบใน Site ให้คลิกที่  ของข้อสอบที่ต้องการ
 4. กรณีที่ต้องการกำหนดนักเรียนที่มีสิทธิ์เข้าสอบในข้อสอบ ให้คลิกที่  ของข้อสอบที่ต้องการ
 5. กรณีที่ต้องการตรวจข้อสอบ ให้คลิกที่  ของข้อสอบที่ต้องการ
 6. กรณีที่ต้องการดูผลการสอบของนักเรียน ให้คลิกที่  ของข้อสอบที่ต้องการ
 7. เมื่อต้องการเพิ่มข้อสอบใหม่เข้าไปใน Site ให้กดปุ่ม Add Test จะปรากฏหน้าจอจดังรูปที่ ก.14
 8. ป้อนข้อมูลคลังข้อสอบใหม่เข้าสู่ระบบ
 9. กดปุ่ม Save เพื่อยืนยันการเปลี่ยนแปลง หากต้องการยกเลิกให้กดปุ่ม Cancel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.13 หน้าจอการจัดการข้อสอบ



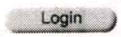
รูปที่ ก.14 หน้าจอการเพิ่มข้อสอบ

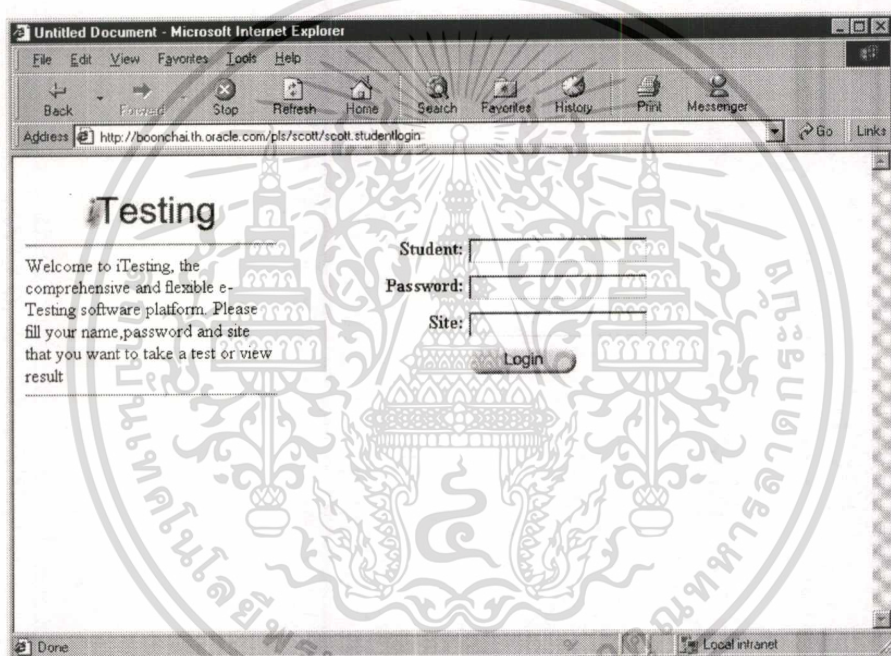
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออกจากระบบ(Logout) ใช้เมื่อต้องการออกจากระบบโดยคลิกที่ 

ก.4 การเข้าสอบในระบบ

นักเรียนที่ต้องการสอบให้ทำการระบุ URL ของแอปพลิเคชันส่วนการเข้าสอบ เมื่อเข้าสู่แอปพลิเคชันจะปรากฏหน้าจอตั้งรูปที่ ก.15 ซึ่งเป็นรูปภาพที่แสดงการ Login เข้าสู่ระบบ

เมื่อปรากฏหน้าจอ Login ดังที่แสดงในรูปที่ ก.15 นักเรียนจะต้องระบุชื่อ (Student), รหัสผ่าน (Password) และ Site เมื่อระบุข้อมูลถูกต้องเรียบร้อยแล้ว ให้กดปุ่ม  ก็จะ สามารถเข้าสู่ระบบได้

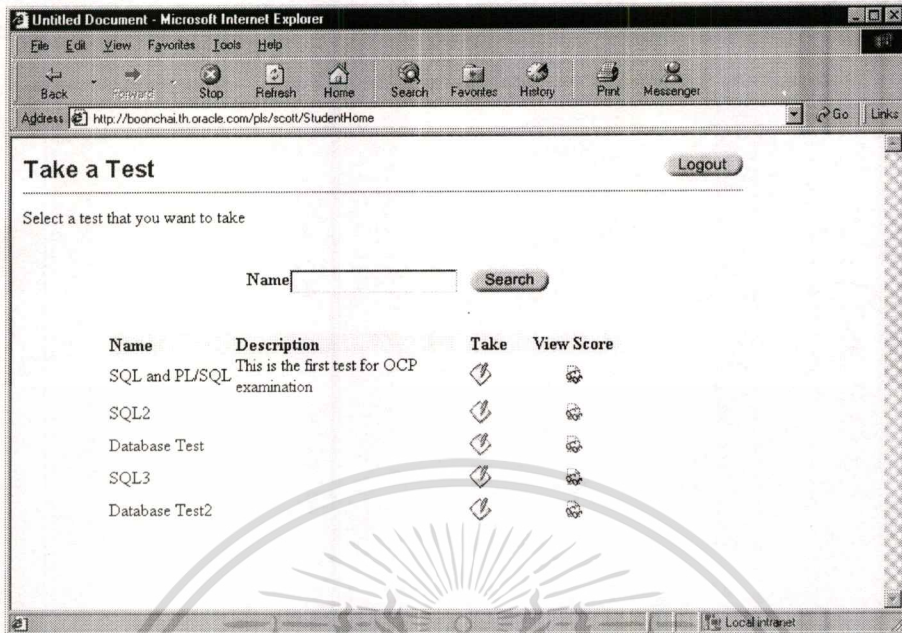


รูปที่ ก.15 หน้าจอการ Login เข้าสู่ระบบของนักเรียน

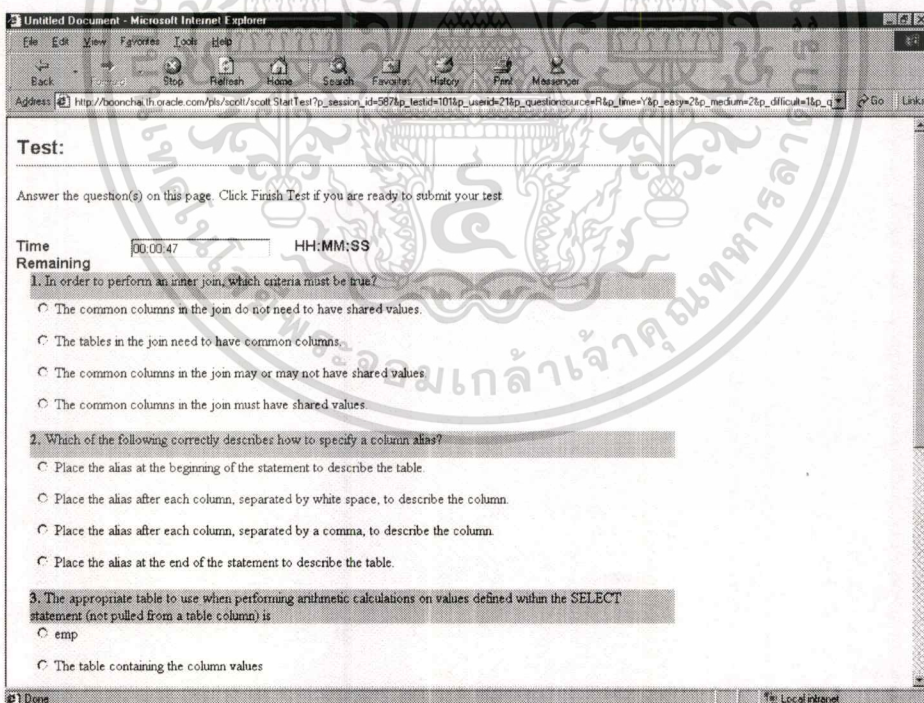
เมื่อเข้าสู่ระบบเรียบร้อยแล้วจะปรากฏหน้าจอตั้งรูปที่ ก.16 ซึ่งแสดงรายชื่อข้อสอบโดยสามารถทำรายการได้ต่างๆดังนี้

- ส่วนการสอบ(Take) เมื่อเลือกทำรายการจะปรากฏหน้าจอการทำข้อสอบดังรูปที่ ก.17 เมื่อทำข้อสอบเสร็จ กรณีที่เป็นข้อสอบประเภท Auto Grade ข้อสอบจะแสดงผลการสอบพร้อมเกรดที่ได้รับดังรูปที่ ก.18
- ส่วนการดูผลสอบ เมื่อเลือกทำรายการจะปรากฏหน้าจอแสดงประวัติผลการสอบที่ผ่านมาดังรูปที่ ก.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

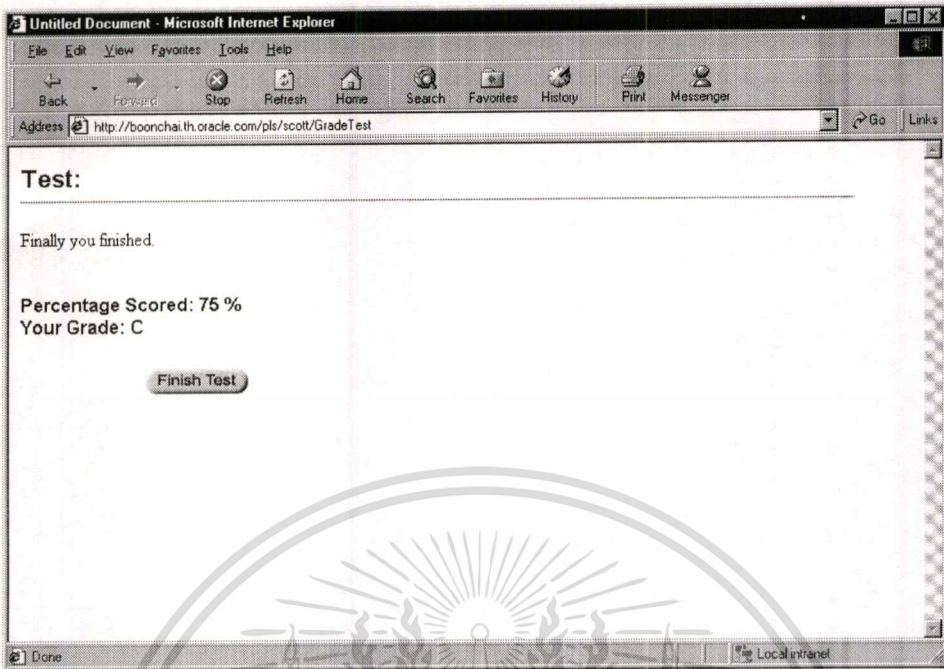


รูปที่ ก.16 หน้าจอแสดงรายชื่อข้อสอบ

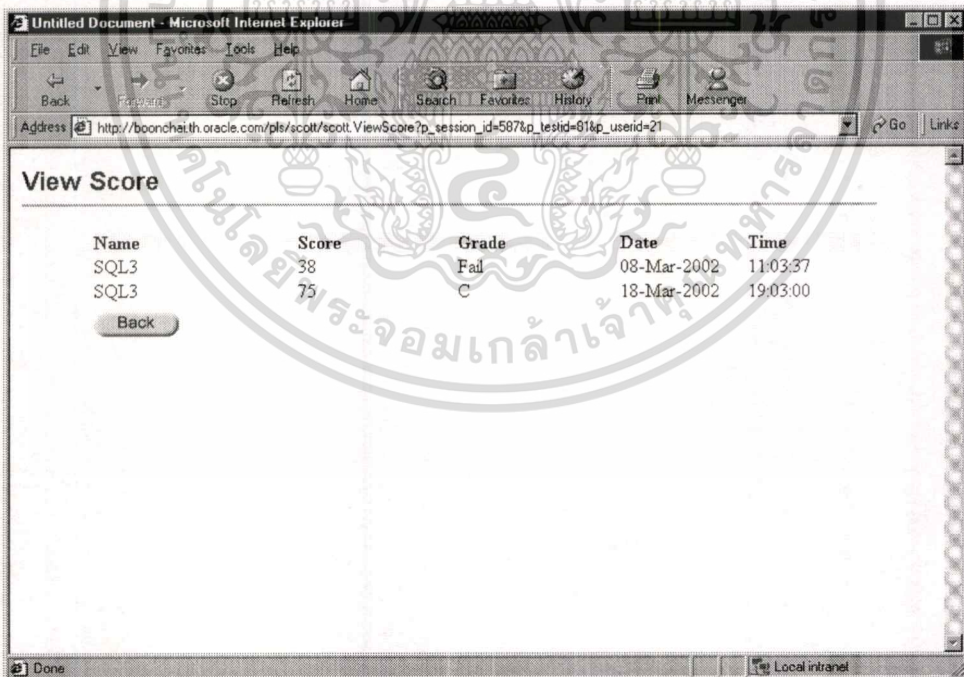


รูปที่ ก.17 หน้าจอการสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.18 หน้าจอสิ้นสุดการสอบ



รูปที่ ก.19 หน้าจอแสดงประวัติการสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายบุญชัย แซ่บู้
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	วิทยาศาสตร์บัณฑิต คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง
ตำแหน่งหน้าที่	Education Consultant
สถานที่ทำงาน	บริษัทออราเคิล คอร์ปอเรชั่น (ประเทศไทย) จำกัด

