สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องมือสำหรับตรวจสอบประสิทธิภาพของเซิร์ฟเวอร์ในด้านงานบริการฐานข้อมูล

TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE



TO45823

โดย

นายณัฐพงศ์ พงศ์แพทย์พินิจ

นายภัทรเศรษฐ์ กอทวีศักดิ์

ปพ.
ฬ 336ก
2544

เลขหมู่.....
เลขทะเบียน 45823
วัน, เดือน, ปี 18 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือสำหรับตรวจสอบประสิทธิภาพของเซิร์ฟเวอร์ในด้านงานบริการฐานข้อมูล

TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE



โดย

นายณัฐพงศ์ พงศ์แพทย์พินิจ

นายภัทรเศรษฐ์ กอทวีศักดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE

MR NUTTAPONG PONGPADPINIT

MR PATTARASETH KOTHAVEESAK

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULLFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF INFORMATION ENGINEERING

DEPARTMENT OF INFORMATION ENGINEERING, FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบเสนอปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	เครื่องมือสำหรับตรวจสอบประสิทธิภาพของเซิร์ฟเวอร์ในด้านงานบริการฐานข้อมูล TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE
นักศึกษา	นายณัฐพงศ์ พงศ์แพทย์พินิจ นายภัทรเศรษฐ์ กอทวีศักดิ์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์สุธีรา พันธุ์ธีรานุรักษ์ อาจารย์ภุชงค์ หงษ์สุวรรณ
ระดับการศึกษา	วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา	เทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(อาจารย์สุธีรา พันธุ์ธีรานุรักษ์)
อาจารย์ผู้ควบคุมวิทยานิพนธ์

(อาจารย์ภุชงค์ หงษ์สุวรรณ)
อาจารย์ผู้ควบคุมวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องมือสำหรับตรวจสอบประสิทธิภาพของเซิร์ฟเวอร์ในด้านงานบริการฐานข้อมูล
TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE

นักศึกษา นายณัฐพงศ์ พงศ์แพทย์พินิจ
นายภัทรเศรษฐ์ กอทวีศักดิ์
อาจารย์ที่ปรึกษา อาจารย์สุธีรา พันธุ์ธีรานุรักษ์
อาจารย์ภุชงค์ หงษ์สุวรรณ

ระดับการศึกษา วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา เทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
ปีการศึกษา 2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้กล่าวถึง เครื่องมือที่ใช้ในการทดสอบประสิทธิภาพการให้บริการของเซิรฟเวอร์ในด้านของฐานข้อมูล

ในปัจจุบันที่มีการใช้เทคโนโลยีแบบไคลแอนต์-เซิรฟเวอร์กันอย่างแพร่หลาย เครื่องคอมพิวเตอร์แบบเซิรฟเวอร์ได้มีการใช้งานเพื่อให้บริการมากขึ้น แต่เราจะสามารถทราบได้อย่างไรว่าเครื่องเซิรฟเวอร์ของเรานั้นจะสามารถให้บริการเป็นที่น่าพึงพอใจมากเพียงใด จึงมีการพัฒนาเครื่องมือทางซอฟแวร์ขึ้นเพื่อใช้วัดประสิทธิภาพการทำงานของเครื่องเซิรฟเวอร์ก่อนนำไปใช้งานจริง

เครื่องมือที่พัฒนาขึ้นนี้จะใช้ทดสอบประสิทธิภาพการให้บริการของเซิรฟเวอร์ในด้านของฐานข้อมูลเป็นหลัก ซึ่งเครื่องมือนี้สามารถใช้ได้กับระบบจัดการฐานข้อมูล (DBMS) เกือบทุกประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	:	TOOLS FOR SERVER PERFORMANCE TESTING IN DATABASE SERVICE
Student name	:	MR NUTTAPONG PONGPADPINIT
		MR PATTARASETH KOTHAVEESAK
Advisor	:	MISS SUTERA PANTERANUWAT
		MR PUCHONG HONGSUWAN
Course	:	BACHELOR OF INFORMATION ENGINEERING
Department	:	INFORMATION ENGINEERING
Faculty	:	ENGINEERING
Year	:	2001

ABSTRACT

This thesis is about tool using for testing performance of server in database service. Today, client-server technology is spreading rapidly as use servers in service purposes. Nevertheless, how can you know that the server can service satisfactorily? So we had developed a software tool for using in this purpose – test server performance – that is the objective of this tool.

The developed tool is using for testing server performance focus on database service. This tool can apply for almost Database Management System (DBMS).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำไม่มีคำกล่าวใดๆที่จะบ่งบอกถึงความกรุณา และความอนุเคราะห์ของ อาจารย์สุธีรา พันธุ์ธีรานุรักษ์และอาจารย์ภุชงค์ หงษ์สุวรรณ อาจารย์ที่ปรึกษาโครงงานพิเศษนี้ได้ ตลอดระยะเวลาของงานศึกษานี้ คำแนะนำและการสั่งสอนที่มีค่ามากมายของท่านทั้งสองได้ชี้นำให้ คณะผู้จัดทำสามารถผ่านอุปสรรคในการทำโครงงานได้อย่างง่ายดาย และยังทำให้คณะผู้จัดทำทราบ ถึงวิธีในการทำงาน ซึ่งเป็นสิ่งสำคัญที่คณะผู้จัดทำได้รับในการศึกษา ณ สถาบันเทคโนโลยีพระจอม เกล้าเจ้าคุณทหารลาดกระบังแห่งนี้ คณะผู้จัดทำขอกล่าวคำขอบคุณ ด้วยความซาบซึ้ง และนับถือแด่ อาจารย์สุธีรา พันธุ์ธีรานุรักษ์และอาจารย์ภุชงค์ หงษ์สุวรรณ

ขอขอบคุณอาจารย์ทุกท่าน ที่สั่งสอนและถ่ายทอดวิชาความรู้อันเป็นพื้นฐานให้กับ คณะผู้จัดได้ใช้ประโยชน์ในการศึกษา กระทำโครงงานนี้ให้สำเร็จลุล่วงโดยสะดวก และยังเป็น ประโยชน์ต่อไปในอนาคตของคณะผู้จัดทำอีกด้วย

และไม่เคยลืมความห่วงใย ความหวังดี และความจริงใจของเพื่อนที่ให้ความช่วยเหลือ คณะผู้จัดทำอย่างบรรยายมิได้ ทำให้ผู้จัดทำผ่านอุปสรรคมากมายใน 4 ปีนี้มาได้ ขอขอบคุณทุกคนจาก ใจจริง

คำขอบคุณอันพิเศษสุดขอมอบให้กับครอบครัวที่น่ารักของผู้จัดทำ ที่คอยอยู่เคียงข้าง ให้ความรัก ความห่วงใย และกำลังใจอย่างไม่เคยขาด ขอบคุณทุกคนที่ให้ความช่วยเหลือแม้จะไม่ได้ กล่าวถึงในที่นี้ ขอบคุณผู้จัดทำที่ไม่เคยยอมแพ้ และก้าวมาถึงจุดนี้ได้

นายณัฐพงศ์ พงศ์แพทย์พินิจ
นายภัทรเศรษฐ์ กอทวีศักดิ์
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	เรื่อง	หน้า
	ปกใน (ภาษาไทย)	ก
	ปกใน (ภาษาอังกฤษ)	ข
	ใบอนุมัติปริญญานิพนธ์	ค
	บทคัดย่อภาษาไทย	ง
	บทคัดย่อภาษาอังกฤษ	จ
	กิตติกรรมประกาศ	ฉ
	สารบัญ	ช
	สารบัญรูป	ฌ
1	บทนำ	1
	1.1 แนวคิดและที่มาของปัญหา	1
	1.2 วัตถุประสงค์	2
	1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
2	Object oriented และ UML	3
	2.1 ขอบเขต (Domain)	3
	2.2 วัตถุ (Object)	3
	2.3 คลาส (Class)	4
	2.4 Abstractions	5
	2.5 ประเภทของ Attributes และ Functions	7
	2.6 ความสัมพันธ์ระหว่างคลาส	8
	2.7 ข้อดีของการพัฒนาเชิงวัตถุ	8
	2.8 UML	9
	2.9 แผนภาพยูสเคส	10
	2.10 แผนภาพคลาส	12
	2.11 แผนภาพสถานะ	14
	2.12 แผนภาพกิจกรรม	15
	2.13 แผนภาพแสดงลำดับขั้นตอนการทำงาน	15
	2.14 แผนภาพ deployment	16
	2.15 แผนภาพคอมโพแนนท์	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	เรื่อง	หน้า
3	การออกแบบ	20
	3.1 ผลการออกแบบ	20
4	ผลการทดลอง	29
	4.1 วิธีการทำงานของโปรแกรม	29
	4.2 ตัวอย่างของเครื่องมือ	31
5	ปัญหาที่เกิดขึ้นและแนวทางในการพัฒนาต่อ	38
	5.1 ปัญหาที่พบในการทำงาน	38
	5.2 แนวทางในการพัฒนาต่อ	38
	ภาคผนวก	
	บรรณานุกรม	

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	ชื่อรูป	หน้า
1.1	แสดงรูปแบบการทำงานของเครื่องมือ	1
2.1	แสดงสัญลักษณ์ของคลาส	4
2.2	แสดงแผนภาพต่างๆใน UML	10
2.3	แสดงแผนภาพยูสเคส	11
2.4	แสดงแผนภาพลำดับขั้นตอนการทำงานที่ใช้สัญลักษณ์ของสถานะ	16
2.5	Class diagram ที่จำลองการแบ่งย่อยระบบ	18
2.6	Drill Down Engineering	19
3.1	แสดงไคลแอนต์อินเทอร์วิว	20
3.2	แสดงแผนภาพ High level use case	21
3.3	แสดงแผนภาพยูสเคสระดับ 1 ของชุดคำสั่ง	22
3.4	แสดงแผนภาพยูสเคสระดับ 1 ของการส่งผลลัพธ์	23
3.5	แสดงแผนภาพยูสเคสระดับ 1 ของการแสดงข้อมูล	24
3.6	แสดงแผนภาพสถานะของการสร้างตัวแปรในการสร้างกราฟ	25
3.7	แสดงแผนภาพสถานะของการเก็บค่าของเวลาการทำงาน	25
3.8	แสดงแผนภาพสถานะของการแสดงผล	26
3.9	แสดงแผนภาพคลาสของระบบ	27
3.10	แสดงแผนภาพกิจกรรมของระบบ	28
4.1	ขั้นตอนการทำงานของโปรแกรม	29
4.2	แสดงคอมโพแนนท์ที่ใช้ในการพัฒนาเครื่องมือ	32
4.3	แสดงหน้าจอเริ่มต้นของเครื่องมือ	33
4.4	แสดงหน้าจอการรับสคริปต์ที่ใช้ทดสอบและdelay time	34
4.5	แสดงหน้าจอการรับสคริปต์จากไฟล์ที่มีอยู่แล้ว	34
4.6	แสดงหน้าจอการรับจำนวนไคลแอนต์ของแต่ละสคริปต์และแสดงข้อมูลของสคริปต์ที่เก็บมา	35
4.7	แสดงหน้าจอการทำการทดสอบประสิทธิภาพของเซิร์ฟเวอร์	36

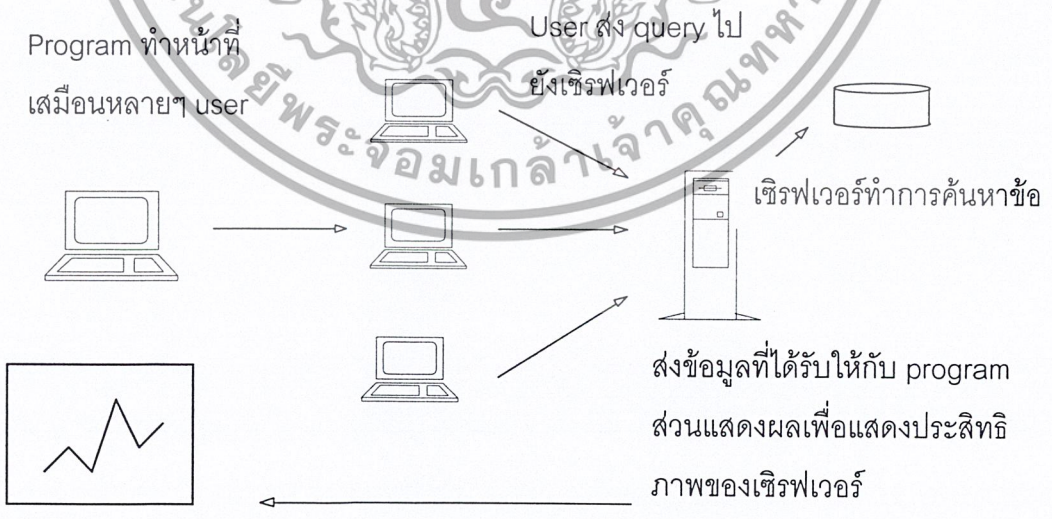
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

ในปัจจุบันปัญหา Server down ส่วนใหญ่มีสาเหตุมาจาก การที่มีผู้ใช้งานจำนวนมากใช้ฐานข้อมูลของ Server ในช่วงเวลาที่พร้อมๆ กัน จึงมีแนวคิดที่จะทำเครื่องมือขึ้นมาไว้คอยตรวจสอบประสิทธิภาพของ Server ในการใช้ฐานข้อมูลจากผู้ใช้งานหลายๆคนพร้อม ๆ กัน เพื่อให้เจ้าของ Server ได้ทราบถึงขีดจำกัดในการรองรับผู้ใช้ และจะได้ทำการป้องกัน หรือแก้ไข ให้ Server มีประสิทธิภาพเพียงพอต่อผู้ใช้งาน

เครื่องมือที่จะจัดทำนั้นเป็น Software โดยหลักการทั่วไปของตัว Program คือ จะทำการจำลองตัวเองให้เป็นกลุ่มของผู้ใช้งาน (Visual user) ขึ้นมาจำนวนหนึ่งแล้วทำการติดต่อไปยังตัว Server พร้อมๆ กัน หรือช่วงเวลาที่ใกล้เคียงกัน (ทำให้ Server เหมือนถูกเรียกใช้งานจากหลาย User พร้อมกัน) เพื่อใช้งานทางด้านฐานข้อมูล หลังจากนั้นตัว Program จะทำการตรวจสอบและเก็บข้อมูลต่างๆที่มีการตอบรับ ในแต่ละ Visual User ที่ทำการติดต่อไปยังเครื่อง Server และจะนำข้อมูลที่ได้รับมาต่างๆ เหล่านี้มาทำการวิเคราะห์ถึงประสิทธิภาพของ Server เพื่อหาจำนวน User ที่ทำให้เกิดการ Over load ของ Server เมื่อทุกVisual User ทำการติดต่อเรียบร้อยจะทำการแสดงผลต่างๆ ออกมาให้ทราบทางหน้าจอ



รูปภาพ 1.1 แสดงรูปแบบการทำงานของเครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

- สามารถนำไปทดสอบและใช้งานจริงได้
- สามารถวัดความแตกต่างระหว่างระบบจัดการฐานข้อมูลในการจัดการฐานข้อมูลได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- เข้าใจการทำงานแบบ multithreading
- เข้าใจการติดต่อกับฐานข้อมูลโดยใช้ ODBC
- ฝึกการทำงานเป็นทีม
- สามารถนำโปรแกรมที่พัฒนาขึ้นมาไปใช้งานได้จริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2

Object Oriented และ UML

Object Oriented คืออะไร

การพัฒนาระบบเชิงวัตถุ (Object Oriented) เป็นแนวคิดและบรรทัดฐานในการวิเคราะห์และออกแบบระบบคอมพิวเตอร์และซอฟต์แวร์ โดยการใช้ Object เป็นตัวหลักเพื่อการพิจารณา ซึ่งจะส่งผลให้การวิเคราะห์และออกแบบระบบ ทำได้ง่ายและมีความถูกต้องยิ่งขึ้น

2.1 ขอบเขต (Domain)

จะเห็นได้ว่าในโลกของเรามีวัตถุอยู่มากมายดังนั้นเราจำเป็นต้องให้กรอบของสิ่งที่เราต้องการพิจารณาหรือสนใจ เพราะ เราไม่สามารถสนใจในทุก ๆ วัตถุในโลกในเวลาเดียวกันได้ และในขณะเดียวกันเราไม่สามารถให้ความสนใจกับทุก ๆ ความสัมพันธ์และทุก ๆ กิจกรรมหรือการกระทำที่เกิดขึ้นได้กรอบของความสนใจที่ว่านี้ เราเรียกมันว่า Domain

ใน Domain หนึ่ง ๆ นั้นสามารถมี วัตถุ ได้ตั้งแต่ 2 ตัวขึ้นไป จนถึงนับไม่ถ้วนในขณะเดียวกันวัตถุ หนึ่ง ๆ สามารถอยู่ใน Domain ได้มากกว่า 1 Domain เช่นกัน

2.2 วัตถุ (Object)

ในโลกของความเป็นจริง เราจะพบเห็นวัตถุ (Object) ต่าง ๆ ไม่ว่าจะเป็นวัตถุที่เราสามารถจับต้องได้ (Tangible Objects) เช่น คอมพิวเตอร์ แป้นพิมพ์ หนังสือ คน เป็นต้น และวัตถุที่ไม่สามารถจับต้องได้ (Intangible Objects) เช่น เวลา ความรู้ ในโลกของเรามีสิ่งที่เกิดจากวัตถุ ต่าง ๆ เช่น กิจกรรม (Activities) ความเคลื่อนไหว (Movement) หรือ การกระทำ (Actions)

สามารถแบ่งการทำงานของวัตถุได้เป็น 2 แบบ

- พาสสิฟ (Passive Object)

  วัตถุที่มีการทำงานแบบพาสสิฟ จะมีบริการไว้ให้วัตถุอื่น ๆ เรียกใช้งาน

- แอกทีฟ (Active Object)

  วัตถุที่มีการทำงานแบบแอกทีฟ จะทำหน้าที่เรียกใช้บริการจากวัตถุที่ทำงานแบบพาสสิฟ

กิจกรรมต่าง ๆ ที่เกิดขึ้นระหว่างวัตถุล้วนแล้วเกิดจากการมีความสัมพันธ์ (Relationship) และการมีปฏิสัมพันธ์(Interaction) ตั้งแต่วัตถุ 2 ตัวขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสัมพันธ์ (Relationship)

  คือ ความเกี่ยวข้องกันหรือระหว่าง วัตถุตั้งแต่ 2 ตัวขึ้นไป เช่น ความเป็น แม่-ลูก ความเป็นเจ้าของการมีอยู่

- ปฏิสัมพันธ์ (Interaction)

  คือ การกระทำใด ๆ ที่เกิดขึ้นระหว่าง วัตถุตั้งแต่ 2 ตัวขึ้นไป เช่น การสร้าง การเปลี่ยนแปลง การกระตุ้น เป็นต้น ซึ่ง Interaction นี้เองที่ทำให้เกิดกิจกรรม ต่าง ๆ

- แอบสทแรคชัน (Abstraction)

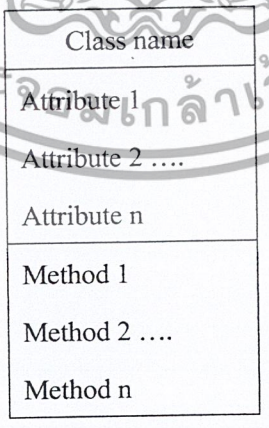
  คือ กระบวนการในการให้แนวคิดกับวัตถุจนเกิดเป็นคลาส

- อินซแทซ์ทิเอชัน (Instantiation)

  คือ กระบวนการของการทำให้เกิดวัตถุจากคลาสที่เราสร้างขึ้นมาบางครั้งจะเรียกวัตถุที่เกิดขึ้นในคอมพิวเตอร์ว่า อินสแทนซ์ (Instance) เพราะเป็นวัตถุที่เกิดจากกระบวนการ อินสทแรกชัน

2.3 คลาส (Class)

คลาสคือ การแยกแยะเอกลักษณ์ของคุณสมบัติพื้นฐานจากกลุ่มของวัตถุที่เหมือนกัน หรืออาจจะกล่าวได้ว่าคลาสคือชนิดของวัตถุ คลาสจะกำหนดแอทริบิวต์และเมธอดของวัตถุ ซึ่งเป็นอินสแทนซ์



รูปภาพ 2.1 แสดงสัญลักษณ์ของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 แอททริบิวต์ (Attribute)

หมายถึง คุณสมบัติต่าง ๆ ที่ใช้บรรยายวัตถุ โดยคุณสมบัติเหล่านี้จะอยู่ภายในขอบเขตของที่เราสนใจ โดยทั่วไปจะเก็บเอาไว้เพื่อใช้ในการประมวลผล เก็บไว้ให้ผู้อื่นใช้งาน เก็บไว้แสดงสถานะของตัววัตถุเอง เพื่อตอบสนองกับเหตุการณ์ภายนอก หรือเก็บเอาไว้เฉย ๆ

ในการเก็บแอททริบิวต์ของวัตถุผู้ออกแบบโปรแกรมมักจะซ่อนเอาไว้ไม่สามารถมองเห็นได้จากผู้ใช้งานภายนอกเนื่องจากการเข้าถึงแอททริบิวต์โดยตรง เปรียบเสมือนการเข้าถึงโครงสร้างของวัตถุ

2.3.2 เมธอดหรือพฤติกรรม (Method or function)

หมายถึง ความสามารถในการทำกิจกรรมของวัตถุ ที่มีไว้เพื่อให้วัตถุอื่น ๆ ในขอบเขตที่เราสนใจ สามารถเรียกใช้หรือกระตุ้นให้เกิดได้ เมธอดสามารถแบ่งออกได้เป็น 3 แบบคือ Simple, Automation และ Continuous

รูปแบบพื้นฐาน (Simple) เป็นวัตถุที่ให้บริการกับการร้องขอเข้ามาและไม่ค้างค่าในหน่วยความจำไว้สำหรับให้บริการในครั้งต่อไป การกระทำเป็นหนึ่งเดียวและสมบูรณ์ในตัว เรียกวัตถุที่มีเฉพาะเมธอดนี้ว่า วัตถุแบบ Primitive

รูปแบบเครื่องจักร (Automation) วัตถุชนิดนี้จะมีกลุ่มของสถานะที่มีขอบเขต วัตถุหนึ่งอาจจะมีสถานะ 1 หรือมากกว่าในเวลาใด ๆ เนื่องจากวัตถุแบบนี้ทำงานแบบ State Machine เพื่อการตอบสนองกับเหตุการณ์ที่เข้ามาจากสถาพแวดล้อมภายนอกจึงเรียกว่า วัตถุแบบรีแอกทีฟ (Reactive Object)

รูปแบบไม่จำกัด (Continous) วัตถุชนิดนี้จะมีกลุ่มของสถานะที่ไม่จำกัดและไม่มีขอบเขต โดยการทำงานของวัตถุในปัจจุบันจะขึ้นอยู่กับการทำงานของเมธอดและอินพุทก่อนหน้า ซึ่งความเกี่ยวข้องระหว่างสถานะนี้อาจจะเป็นความสัมพันธ์แบบต่อเนื่อง

2.4 Abstractions

กระบวนการในการให้ Concept กับวัตถุต่างๆ ในโลกความเป็นจริง เพื่อสร้าง class นั้นเราเรียกว่า Abstractions ซึ่งแบ่งออกได้เป็น 4 กระบวนการย่อยๆ คือ

1. Classification Abstraction
2. Aggregation Abstraction
3. Generalization Abstraction
4. Association Abstraction

การใช้งาน Abstraction เพื่อการวิเคราะห์ Problem domain นั้น ไม่ได้มีข้อกำหนดตายตัวว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Problem domain หนึ่งๆ จะต้องใช้ทุก Abstraction หรือจะใช้ Abstraction ตามลำดับก่อนหรือหลังอย่างไร แต่ส่วนมากแล้ว Classification มักจะเป็น Abstraction แรกที่ถูกใช้ หลังจากนั้นจะขึ้นอยู่กับมุมมองและวิจารณญาณของผู้วิเคราะห์ออกแบบระบบ ที่จะใช้ Abstraction ใดที่เหมาะสมเพื่อการวิเคราะห์ Problem domain และจากประสบการณ์ของผู้เขียน พบว่า Abstraction หนึ่งๆ นั้นมักจะถูกใช้มากกว่าหนึ่งครั้งเสมอ เพื่อการวิเคราะห์ Problem domain เพียงหนึ่งเดียว

2.4.1 Classification Abstraction

Classification Abstraction เป็นกระบวนการที่ใช้เพื่อแยกประเภท (Classify) วัตถุต่างๆ ที่อยู่ใน domain และให้ Concept กับ Object ต่างๆ เหล่านั้น เพื่อให้ได้ "Class พื้นฐาน" (Fundamental Classes) ที่ต้องการ

เราถือว่า Classification เป็น Abstraction ที่สำคัญที่สุด เพราะ Class เกิดขึ้นด้วย Classification Abstraction ถ้าหากว่า Class พื้นฐานที่เกิดขึ้นในขั้นตอนนี้เกิดข้อผิดพลาด การสร้าง Class ใหม่ๆ ด้วยกระบวนการ Aggregation และ Generalization และการสร้างความสัมพันธ์ระหว่าง Class ต่างๆ ด้วย Association ย่อมจะมีความผิดพลาดด้วยเช่นกัน

2.4.2 Aggregation Abstraction

Aggregation Abstraction คือกระบวนการที่นำเอา Class พื้นฐาน มารวมกันหรือประกอบกัน (Aggregate) เพื่อให้เกิดเป็น Class ที่ใหญ่ขึ้น หรือซับซ้อนขึ้น โดย Class พื้นฐานดังกล่าวคือ Class ที่สร้างขึ้นในขั้นตอน Classification นั่นเอง

การทำ Aggregation คือการตอบคำถามว่าใน Class กลุ่มหนึ่งนั้น เราสามารถที่จะนำมันมารวมกัน (Compose) เพื่อให้เกิด Class ใหม่ ที่มี concept ใหม่ ได้หรือไม่อย่างไร หรือในทางกลับกัน ก็คือการตอบคำถามว่า มี Class ใดบ้างที่สามารถแบ่งย่อย (Decompose) ออกเป็นส่วนประกอบต่างๆซึ่งมี Concept ต่างไปจากเดิมได้

2.4.3 Generalization Abstraction

Generalization Abstraction คือกรระบวนการในการนำ Class ที่มีลักษณะเหมือนหรือคล้ายคลึงกันหรือมีคุณสมบัติอย่างใดอย่างหนึ่งร่วมกัน (General) มาจัดหมวดหมู่ไว้เป็น Class เดียวกัน กระบวนการการย้อนกลับของ Generalization Abstraction เรียกว่า Specialization ซึ่ง Specialization คือการตอบคำถามว่าใน Class หนึ่งๆ นั้นสามารถจำแนกเป็น Class อะไรได้บ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 Association Abstraction

Association Abstraction คือกระบวนการในการสร้างความสัมพันธ์ระหว่าง Class ต่างๆ ใน Problem domain ที่เราสนใจ ความสัมพันธ์ดังกล่าวคือความสัมพันธ์ที่ไม่สามารถอธิบายได้ Aggregation (ความสัมพันธ์ในเชิงการรวมกัน การประกอบกัน หรือการแบ่งแยกส่วนประกอบ) หรือ Generalization (ความสัมพันธ์ในเชิง การจัดประเภทจัดหมวดหมู่ หรือการจำแนก) แต่ Association เป็นการอธิบายความสัมพันธ์ของ Class ในเชิงกิจกรรม เช่น การใช้งาน การขับเคลื่อน การบริโภค เป็นต้น หรือความสัมพันธ์ในเชิงสถิตย์ เช่น การเป็นเจ้าของ การให้กำเนิด การผลิต เป็นต้น

2.5 ประเภทของ Attributes และ Functions

การจำแนกประเภทของ Attributes และ Functions ตามความสามารถในการเห็นและการเข้าถึง Attributes และ Functions เหล่านั้น ความสามารถนี้เรียกว่า Visibility

การจำแนกประเภทของ Attributes และ Functions ของ Class ใดๆ จะใช้ Visibility ของ Attributes และ Functions เป็นหลักเกณฑ์ สามารถแบ่งได้ 3 ประเภทดังนี้

2.5.1 Private Attributes and Functions

คือ Attributes และ Function ที่ไม่สามารถเห็นได้เลยจากภายนอก หนทางเดียวที่จะเข้าถึง Attributes เหล่านี้ได้คือต้องผ่านทาง Function ที่ Class มีไว้เพื่อการเข้าถึง Attributes เหล่านี้เท่านั้น Private Function มีไว้เพื่อใช้งานภายในตัววัตถุของ Class เท่านั้น ไม่สามารถถูกเรียกใช้โดยตรงจากภายนอกได้

2.5.2 Protected Attributes and Functions

คือ Attributes และ Function ที่ไม่สามารถเห็นได้จากภายนอกแต่เป็นส่วนที่สามารถส่งต่อให้ Inherited Class ได้เท่านั้น

2.5.3 Public Attributes และ Functions

คือ Attributes และ Functions ที่สามารถมองเห็นได้ และสามารถเรียกใช้ได้โดยตรงจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความสัมพันธ์ระหว่างคลาส

สามารถกำหนดความสัมพันธ์ระหว่างคลาสด้วยเหตุผลใดเหตุผลหนึ่ง เช่น เป็นสิ่งที่บ่งบอกถึงชนิดที่เหมือนกัน แสดงความเกี่ยวข้องกันระหว่างคลาส สามารถสรุปความสัมพันธ์ระหว่างคลาสได้ 3 แบบใหญ่

2.6.1 แอซโซซิเอชัน (Association)

ความสัมพันธ์แบบแอซโซซิเอชัน คือการที่วัตถุหนึ่งใช้บริการของอีกวัตถุแต่ว่าไม่ได้เป็นเจ้าของวัตถุนั้น

2.6.2 การสืบทอดคุณสมบัติ (Inheritance)

การที่คลาสหนึ่งเป็นความเฉพาะของอีกคลาสหนึ่งโดยใช้ความสัมพันธ์ที่เรียกว่า การสืบทอดคุณสมบัติ (Inheritance หรือ Generalization) หมายความว่าคลาสลูกจะมีคุณสมบัติทุกอย่างที่คลาสพ่อแม่มี คลาสลูกอาจเพิ่มคุณสมบัติเข้าไปโดยการเพิ่มแอททริบิวต์และเมธอดเข้าไป

2.6.3 ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (Aggregation)

จะใช้กี่ต่อเมื่อมีวัตถุหนึ่งบรรจุอีกวัตถุหนึ่งไว้ทั้งในทางตรรกและทางกายภาพ คลาสที่ใหญ่กว่าเรียกว่าเจ้าของ (Owner หรือ Whole) ส่วนคลาสที่เล็กกว่านั้นเรียกว่าคลาสคอมโพเนนท์ (Component หรือ Owned หรือ Part)

2.7 ข้อดีของการพัฒนาเชิงวัตถุ

- นักพัฒนาสามารถ โมเดลระบบได้ครบถ้วนมากขึ้น
- ช่วยเพิ่มความเข้าใจในโมเดลของปัญหา
- ช่วยเพิ่มเสถียรภาพของการเปลี่ยนแปลง
- มีโมเดลสนับสนุนของการนำกลับมาใช้ใหม่ (Reuse)
- สนับสนุนการเปลี่ยนแปลงขนาดของระบบ (Scalability)
- สนับสนุนการออกแบบระบบที่เชื่อถือได้และมีความปลอดภัยสูง (Reliability and Safety)
- สนับสนุนการทำงานพร้อมกัน (Concurrency)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

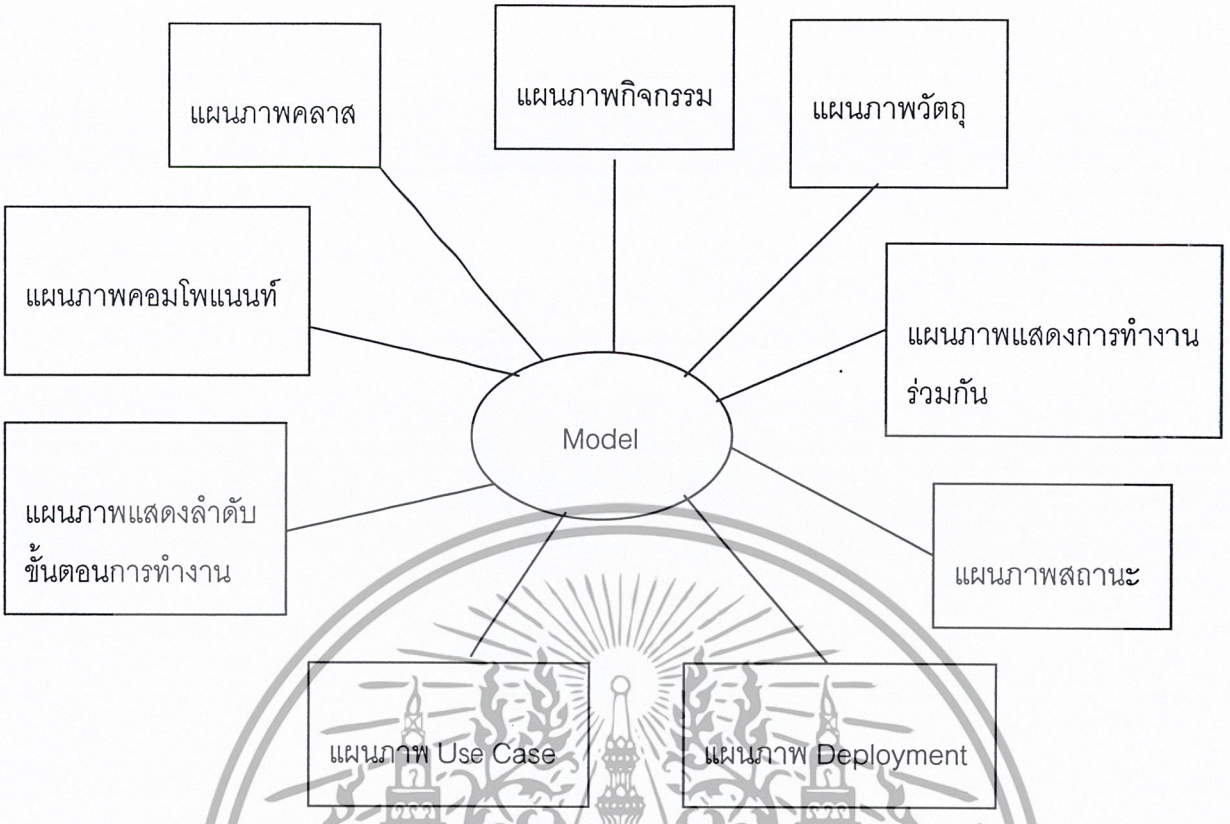
2.8 UML

UML หรือ Unified Modeling Language เป็นภาษาในการโมเดลมาตรฐาน เป็นโมเดลสำหรับการพัฒนาเชิงวัตถุ เกิดจากการพัฒนาร่วมกันของ Grady Booch, Ivar Jacobson และ Jim Rumbaugh ซึ่งเป็นผู้นำเทคโนโลยีทางด้านวัตถุ UML ได้รวมแนวความคิดของวิธีต่างๆ เข้าด้วยกัน จุดประสงค์เพื่อต้องการสร้างโมเดลในการพัฒนาที่เข้าใจและสร้างได้ง่าย แต่สามารถนำไปใช้ได้กับทุกระบบ UML เองนั้นยังได้รับการสนับสนุนจากผู้นำในอุตสาหกรรมด้านคอมพิวเตอร์หลายๆ รายเช่น HP, ICON Computer, IBM, I-Logix, Microsoft, ObjecTime, oracle, Unisys เป็นต้น

ใน UML มีโมเดลสื่อสารด้วยภาพสำหรับระบบหลายๆ โมเดล โดยแต่ละโมเดลจะแสดงมุมมองต่อระบบไม่เหมือนกัน ซึ่งประกอบด้วย

- แผนภาพยูสเคส (ยูสเคส Diagram) แสดงการติดต่อระหว่างระบบกับผู้ใช้
- แผนภาพคลาส(Class Diagram) แสดง โครงสร้างทางตรรกของระบบ
- แผนภาพวัตถุ(Object Diagram) แสดงวัตถุและความสัมพันธ์ของระบบ
- แผนภาพสถานะ(State Diagram) แสดงพฤติการณ์ของระบบ
- แผนภาพคอมโพแนนท์(Component Diagram) แสดง โครงสร้างทางกายภาพของซอฟแวร์
- แผนภาพ Deployment(Deployment Diagram) แสดงการติดต่อระหว่างซอฟแวร์กับฮาร์ดแวร์
- แผนภาพแสดงลำดับขั้นตอนการทำงาน(Sequence Diagram) แสดงพฤติการณ์ของระบบ
- แผนภาพกิจกรรม(Activity Diagram) แสดงการไหลของอีเวนต์(Event) ในยูสเคส
- แผนภาพแสดงการทำงานร่วมกัน(Collaboration Diagram) แสดงการทำงานร่วมกันของสิ่งต่างๆ ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 2.2 แสดงแผนภาพต่าง ๆ ใน UML

2.9 แผนภาพ ยูสเคส

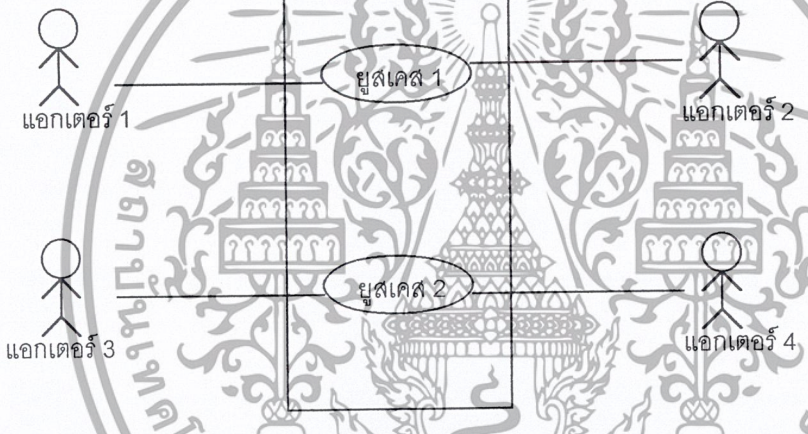
แผนภาพ ยูสเคส เป็นแผนภาพสำหรับแสดงการติดต่อระหว่างระบบกับผู้ใช้งานหรือแอคเตอร์ (actor) แสดงการไหลของอีเวนต์ แผนภาพ ยูสเคส จะแสดงมุมมองต่อฟังก์ชันการทำงานหลักๆ ของระบบซึ่งผู้พัฒนาระบบสามารถใช้แผนภาพ ยูสเคส ในการติดต่อสื่อสารกับผู้ใช้ระบบได้ แผนภาพ ยูสเคส อาจเป็นแผนหลักในการพัฒนาระบบเนื่องจากได้รวมความต้องการและรายละเอียดการทำงานไว้ในแผนภาพแล้ว

แผนภาพยูสเคสยังได้รวมความต้องการหรือรายละเอียดในการทำงานหรือ Scenarios ของระบบไว้ภายในแล้วซึ่ง Scenarios จะแสดงรายละเอียดลำดับในการติดต่อสื่อสารระหว่างวัตถุในระบบ Scenarios เป็น อินสแตนท์ ของยูสเคสเปรียบเหมือนกับวัตถุเป็นอินสแตนท์ของคลาสนั่นเอง ดังนั้นแผนภาพยูสเคสจึงเหมือนแผนภาพของคลาสที่จะแสดงโครงสร้างทางตรรกของ Scenarios Scenarios เป็นเครื่องมือที่มีประโยชน์ใช้ในความต้องการของระบบเนื่องจาก Scenarios จะกำหนดรูปแบบในการพัฒนารวมทั้งการแสดงมุมมองของระบบจากนักคอมพิวเตอร์ และแสดงรายละเอียดความต้องการของระบบ รวมทั้งคุณลักษณะและการพัฒนาระบบได้อย่างครบถ้วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยูสเคสจะอธิบายทุกๆกรณีที่เกิดขึ้นได้ทั้งหมด ในขณะที่ Scenarios จะหมายถึงเหตุการณ์จริงที่เกิดขึ้นภายใต้เงื่อนไขต่างๆ ของยูสเคสนั้น ซึ่งอาจจะไม่ได้เกิดทุกๆกรณีที่ระบุไว้ในยูสเคสก็ได้

จุดประสงค์หลักของการเขียนยูสเคสไดอะแกรม ก็เพื่อเล่าเรื่องราวของ Problem domain ทั้งหมดว่า มีส่วนประกอบอะไรบ้างและเกี่ยวพันกันจนกลายเป็นระบบได้อย่างไร การเขียนยูสเคสไดอะแกรม จะช่วยให้ผู้พัฒณาระบบสามารถแยกแยะได้ว่าจะมีกิจกรรมอะไรที่น่าจะเกิดขึ้นในระบบบ้าง ยิ่งไปกว่านั้นยูสเคสไดอะแกรมถือเป็นไดอะแกรมพื้นฐาน ซึ่งมีขีดความสามารถในการอธิบายสิ่งต่างๆ ด้วยรูปภาพที่ไม่ซับซ้อน ยูสเคสไดอะแกรมจะออกมาอย่างมีประสิทธิภาพหากถูกเขียนด้วยผู้ที่มีความเข้าใจใน Problem domain หรือเนื้องานตรงนั้นจริงๆ



รูปภาพ 2.3 แผนภาพยูสเคส

2.9.1 บทบาทของ Scenarios ในการกำหนดเมธอด

แผนภาพสถานะนั้นเป็นโมเดลที่ทำให้สามารถกำหนดและลงมือเขียนโค้ดสำหรับวัตถุได้ดังที่ผ่านมา Scenarios นั้นเหมือนกับเป็นแผนภาพที่หลักในการวิเคราะห์ระบบ ส่วนแผนภาพสถานะนั้นเป็นโมเดลที่แสดงการทำงานของวัตถุ

Scenario เป็นรายละเอียดของ ยูสเคส โดยโมเดลของ Scenario จะแสดงแมซเสจตามลำดับรวมทั้งโครงสร้างของวัตถุ ข้อดีของ Scenario คือช่วยเพิ่มความเข้าใจในตัวระบบและทำให้ทั้งผู้ชำนาญและผู้ใช้ระบบสามารถเข้าใจระบบจาก Scenario ได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 แสดง Scenarios พร้อมด้วยเงื่อนไขทางเวลา

การแสดง Scenarios พร้อมเงื่อนไขทางเวลาสามารถทำได้ 2 วิธี คือการใช้แผนภาพเวลาและการใช้แผนภาพแสดงลำดับขั้นตอนการทำงานซึ่งแผนภาพเวลานั้นมีประโยชน์อย่างมากสำหรับระบบที่ต้องการมีการติดตามดูการทำงานตามลำดับเวลาเป็นสำคัญ

2.9.3 Uses และ Extends

แต่ละยูสเคสภายในระบบเองก็สามารถมีความสัมพันธ์ซึ่งกันและกันได้ โดยความสัมพันธ์ของยูสเคสนั้นสามารถแบ่งออกได้เป็น 2 แบบคือ

1.Uses

หมายถึงการที่ยูสเคสหนึ่ง เรียกใช้งานยูสเคสอีกอันหนึ่ง คล้ายๆกับการเรียกใช้งานโปรแกรมย่อยโดยโปรแกรมหลัก

2.Extends

หมายถึงการที่ยูสเคสหนึ่งไปมีผลต่อการทำงานตามปกติของอีกยูสเคสหนึ่ง นั่นหมายถึงว่ายูสเคสที่มา Extend นั้นจะมีผลทำให้การดำเนินการของยูสเคสที่ถูก Extend มาราบกวน หรือมีการสะดุดหรือมีการเปลี่ยนกิจกรรมไป

2.10 แผนภาพคลาส

แผนภาพของคลาสเป็นแผนภาพที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบ โดยวิธีเชิงวัตถุ เนื่องจากแผนภาพของคลาสจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย จะเป็นโครงสร้างหลักของระบบและสามารถแยกรายละเอียดด้วยวิธีเชิงวัตถุได้โดยที่แผนภาพของคลาสอาจมีลักษณะคล้ายกับแผนภาพวัตถุแต่ว่าแผนภาพของคลาสไม่ได้แสดงอินสแทนซ์หรือวัตถุทั้งหมดจะเน้นที่การแสดงโครงสร้างมากกว่าความสำคัญของคลาส ใช้กำหนดความสัมพันธ์และการเกี่ยวข้องของคลาส

แผนภาพของคลาสเป็นแผนภาพที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเชิงวัตถุ เนื่องจากแผนภาพของคลาสจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย แผนภาพของคลาสจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด (Decomposite) ด้วยวิธีเชิงวัตถุ

หลังจากเราได้วัตถุและความสัมพันธ์ เราสามารถโมเดลวัตถุให้กลายเป็นคลาสได้โดยดูจากวัตถุใดที่มี “ชนิด” เหมือนกันก็สามารถโมเดลให้เป็นคลาสเดียวกันได้ หลังจากได้คลาสแล้วก็สามารถกำหนดความสัมพันธ์พร้อมเขียนแผนภาพของกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพคลาส คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแง่ต่างๆ (Relationship) ระหว่างคลาสเล่านั้น ซึ่งมีความสัมพันธ์ที่กล่าวถึงในคลาสไดอะแกรมนี้ ถือเป็นความสัมพันธ์เชิงสถิตย์ (Static Relationship) หมายถึงความสัมพันธ์ที่มีอยู่แล้วเป็นปกติในระหว่างคลาสต่างๆ ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรมต่างๆ ซึ่งเรียกว่า ความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship)

ความสัมพันธ์เชิงสถิตย์นั้นมักจะไม่แปรสภาพความสัมพันธ์ไปตามเวลามากนัก หรือเราอาจเรียกว่า เป็นความสัมพันธ์ที่คงทน

ความสัมพันธ์เชิงกิจกรรมมักจะแปรรูปความสัมพันธ์ไปตามเวลา หรือเป็นความสัมพันธ์ที่ไม่คงทน

2.10.1 การวิเคราะห์การส่งแมสเซจ

ระบบปฏิบัติการค่าโพรเซสหนึ่งต้องการติดต่อไปยังอีกโพรเซสหนึ่งสามารถทำได้หลายวิธี อาจใช้วิธีแบ่งหน่วยความจำ หรือวิธีการส่งแมสเซสระหว่างโพรเซสก็ได้

2.10.2 การกำหนดโอเปอเรชัน

ส่วนใหญ่แล้วโอเปอเรชันของคลาสจะเกี่ยวข้องกับการเปลี่ยนสถานะดังนั้นการเขียนแผนภาพสถานะนั้นจึงมีความสัมพันธ์กับโอเปอเรชันนั้นและโอเปอเรชันยังเกี่ยวข้องกับการรับแมสเซจและอีแวนต์ด้วย

2.10.3 ความสัมพันธ์ระหว่างวัตถุ

ในระบบที่พัฒนาด้วยวิธีเชิงวัตถุนั้นมีความสัมพันธ์ระหว่างวัตถุใหญ่ ๆ อยู่ดังนี้คือ

- แอซโซซิเอชัน (Association)
- ความเป็นส่วนหนึ่งของ (Aggregation)
- คอมโพรซิชัน (Composition)
- การสืบทอดคุณสมบัติ (Generalization)

แอซโซซิเอชัน

ความสัมพันธ์แบบแอซโซซิเอชันนั้นเป็นคลาสแบบสองทิศทาง จะเกิดขึ้นเมื่อวัตถุของคลาสสองวัตถุมีความเกี่ยวข้องกันหรือมีการสื่อสารถึงกัน

แอสโซซิเอทีฟคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสโซซิเอทีฟคลาสเป็นคลาสที่เกิดจากความสัมพันธ์ระหว่างสองวัตถุ เช่น แอสโซซิเอทีฟคลาสในระบบการทำงานแบบกระจายคือคลาสแมสเซจ

ความสัมพันธ์แบบเป็นส่วนหนึ่งของและคอมโพรซิชัน

ความสัมพันธ์แบบเป็นส่วนหนึ่งของ เป็นความสัมพันธ์แบบแอสโซซิเอชัน จะหมายถึงคลาสหนึ่งประกอบด้วยอีกคลาสหนึ่งทั้งในทางตรรกและในทางกายภาพ

สำหรับความสัมพันธ์แบบคอมโพซิชันนั้นเป็นชนิดพิเศษของความสัมพันธ์แบบเป็นส่วนหนึ่งของคือนอกจากวัตถุที่เป็นเจ้าของ (Composite Object) นั้นจะประกอบด้วยวัตถุส่วนประกอบ (Component Object) ที่อยู่ภายในแล้ว วัตถุที่เป็นเจ้าของยังมีหน้าที่รับผิดชอบวัตถุส่วนประกอบที่ตัวเองเป็นเจ้าของอยู่ด้วย เช่น วัตถุเจ้าของนั้นมีหน้าที่ในการสร้างและทำลายวัตถุส่วนประกอบ

ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Generalization Relationship)

ความสัมพันธ์แบบสืบทอดคุณสมบัติเป็นความสัมพันธ์ระหว่างคลาสที่เกิดขึ้น เมื่อคลาสมีการสืบทอดคุณสมบัติมาจากอีกคลาสหนึ่ง สามารถใช้ความสัมพันธ์แบบการสืบทอดคุณสมบัติในการจัดลำดับขั้นของคลาสได้

ความสัมพันธ์แบบการสืบทอดคุณสมบัติเป็นความสัมพันธ์ระหว่างคลาสที่เกิดขึ้นเมื่อคลาสหนึ่งมีการสืบทอดคุณสมบัติมาจากอีกคลาสหนึ่ง เราสามารถใช้ความสัมพันธ์แบบการสืบทอดคุณสมบัตินี้ในการจัดลำดับขั้นของคลาสได้ โดยคลาสที่อยู่ด้านบนของลำดับชั้นนั้นเราจะเรียกว่าซูเปอร์คลาสซึ่งเป็นคลาสที่สืบทอดคุณสมบัติและพฤติกรรมกับคลาสที่อยู่ระดับล่างของลำดับชั้นซึ่งเราเรียกว่าสับคลาส

2.11 แผนภาพสถานะ

การทำงานของระบบต่างๆ ไม่ว่าจะเป็นระบบภายในคอมพิวเตอร์ หรือระบบในโลกของความเป็นจริงก็ตาม สิ่งที่เคลื่อนไหวหรือการเปลี่ยนแปลงใดๆ ที่เกิดขึ้นในระบบนั้น เรียกว่า กิจกรรม (Activity) ซึ่งกิจกรรมนั้นเกิดขึ้นจากออบเจกต์หรือวัตถุต่างๆ ในระบบกระทำต่อกัน สิ่งที่เราใช้เพื่อบรรยายกิจกรรมโดยรวมที่เกิดขึ้นในระบบนั่นก็คือ Sequence Diagram แต่เมื่อพิจารณาเข้าไปในรายละเอียดของกิจกรรมที่เกิดขึ้นจะพบว่า กิจกรรมโดยรวมของระบบเกิดจากกิจกรรมย่อยของวัตถุแต่ละตัวรวมกันนั่นเอง กลไกที่ทำให้ระบบมีกิจกรรมการส่งก็คือการส่งแมสเซจ หรือในทาง Object Oriented ก็คือการเรียกใช้ฟังก์ชั่นของวัตถุนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรมที่เกิดขึ้นในวัตถุนั้นเกิดจาก 2 สิ่งประกอบกัน นั่นคือ สถานะ (State) และการเปลี่ยนสถานะ (Transition) การที่วัตถุใดๆ เปลี่ยนจากสถานะที่ 1 ไปยังสถานะที่ 2 จะทำให้เกิดกิจกรรม หรือส่วนของกิจกรรมขึ้นในตัววัตถุนั้น

2.11.1 การสืบทอดคุณสมบัติของโมเดลสถานะ

โมเดลสถานะของระบบที่ออกแบบด้วยวิธีเชิงวัตถุสามารถสืบทอดคุณสมบัติได้เพื่อให้เข้ากับคุณสมบัติของความสามารถในการในกลับมาใช้ใหม่ นอกจากนั้นการสืบทอดคุณสมบัติของโมเดลยังช่วยเพิ่มความยืดยุ่นในการสร้างแผนภาพสถานะด้วย

2.12 แผนภาพกิจกรรม

แผนภาพกิจกรรมนั้นจะแสดงการทำงานของระบบโดยพิจารณาตามกิจกรรมที่เกิดขึ้น ซึ่งจะแสดงแตกต่างจากแผนภาพสถานะที่จะแสดงสถานะของระบบ

2.13 แผนภาพแสดงลำดับขั้นตอนการทำงาน

การจำลองภาพในเชิงกิจกรรม (Dynamic View) ด้วย Sequence Diagram

การสร้างแบบจำลองเชิงกิจกรรม (Dynamic Model หรือ Behavioral Model) ของ Problem Domain ซึ่งก็คือการจำลองกระบวนการที่ทำให้เกิดกิจกรรมของระบบ เกิดจากชุดของกิจกรรม ซึ่งกิจกรรมหนึ่งๆนั้นเกิดจากการที่วัตถุหนึ่งโต้ตอบกับอีกวัตถุหนึ่ง

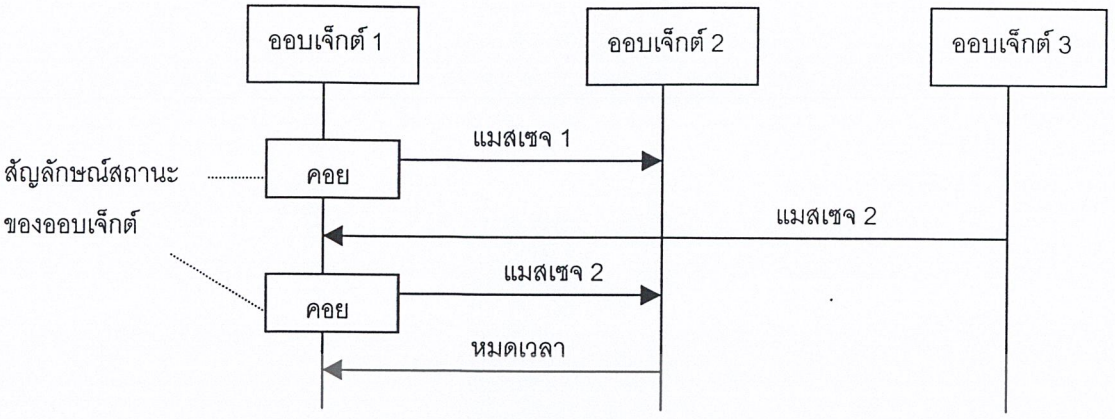
Dynamic Model คือการจำลองกิจกรรมและลำดับของกิจกรรมที่เกิดขึ้นหรืออาจเกิดขึ้นใน Problem Domain ให้อยู่ในรูปที่สามารถเข้าใจได้โดยง่าย

Sequence Diagram เป็น Diagram ที่ประกอบด้วยคลาส หรือวัตถุ เส้นที่ใช้เพื่อแสดงลำดับเวลาและเส้นที่ใช้เพื่อแสดงกิจกรรมที่เกิดขึ้นจากวัตถุหรือคลาสในไดอะแกรม

Sequence Diagram จะมีบทบาทและมีประโยชน์ในการช่วยพิจารณาว่าใน Class Diagram ที่เราสร้างขึ้นมีฟังก์ชั่นใดของคลาสใดขาดหายไป หรือควรเพิ่มเติมเข้ามาหรือไม่ เพียงใด ซึ่งมีส่วนช่วยให้คลาสต่างๆ ที่เราสร้างขึ้นใน Class Diagram มีความสมบูรณ์มากขึ้น

จะแสดงลำดับของแมซเสจระหว่างวัตถุ โดยเส้นในแนวตั้งแสดงวัตถุ โดยมีชื่อของวัตถุที่อาจจะเขียนอยู่ด้านบนหรือล่างของเส้น ส่วนเส้นในแนวนอนคือแมซเสจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 2.4 แสดงแผนภาพลำดับการขั้นตอนการทำงานที่ใช้สัญลักษณ์ของสถานะ

2.14 แผนภาพ deployment

สำหรับการออกแบบนั้นใน UML มีไดอะแกรมสำหรับรองรับคือ แผนภาพ Deployment ที่จะแสดงโครงสร้างทางด้านฮาร์ดแวร์ของระบบเป็นหลัก ส่วนแผนภาพคอมโพแนนท์จะแสดงโครงสร้างทางซอฟต์แวร์

ใน แผนภาพ Deployment นั้นจะเป็นการอธิบายโครงสร้างทางด้านฮาร์ดแวร์ ดังนั้นในไดอะแกรมจึงประกอบด้วยโนด ที่ใช้สำหรับแทนหน่วยประมวลผล และการเชื่อมต่อระหว่างโนดทางด้านกายภาพ ดังเช่นการเชื่อมต่อผ่านโมเด็ม หรือการเชื่อมต่อผ่านเน็ตเวิร์คอีเทอร์เน็ตเป็นต้น

การวิเคราะห์และออกแบบระบบด้วยหลักการ Object Oriented Analysis and Design นั้น จะถือว่าทุกๆ ส่วนประกอบของระบบงานนั้นเป็นวัตถุตัวหนึ่งเสมอ ซึ่งวัตถุนั้น ไม่ได้หมายถึงเฉพาะส่วนที่อยู่ใน Application หรือ Software เท่านั้น แต่ยังหมายรวมถึง เครื่องคอมพิวเตอร์ หรือ ฮาร์ดแวร์ ซึ่งถือเป็นข้อได้เปรียบของการออกแบบระบบด้วย Object Oriented Analysis and Design เพราะเราสามารถที่จะออกแบบซอฟต์แวร์และฮาร์ดแวร์ โดยไม่ต้องเปลี่ยนหลักการที่ใช้เพื่อการออกแบบ การออกแบบในส่วนของฮาร์ดแวร์ของระบบนั้นเรียกว่า System Architecture Design

เครื่องมือที่ใช้ในการทำ System Architecture Design นั้นเรียกว่า แผนภาพ Deployment ซึ่ง แผนภาพ Deployment เป็นไดอะแกรมที่มีลักษณะเหมือนกับ Class Diagram โดยส่วนประกอนทางฮาร์ดแวร์ (Hardware Module) ตัวหนึ่งๆใน แผนภาพ Deployment ก็จะเปรียบได้กับคลาสหนึ่ง ใน แผนภาพ Deployment นั่นเอง

หลักการทำ System Architecture Design โดยแผนภาพ Deployment

การทำ System Architecture Design หรือการออกแบบระบบฮาร์ดแวร์ด้วยแผนภาพ Deployment นั้น ต้องมีความสัมพันธ์กับ Application Architecture Design ซึ่งออกแบบด้วย Component

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

Diagram คือแต่ละ Software Component ต้องอยู่ภายใน Hardware Module ใดโมดูลหนึ่งเสมอ จะต้องไม่มี Software Component ใดขาดหายไป

ดังนั้นเมื่อสร้างแผนภาพ Deployment สิ่งที่ต้องทำไปพร้อมๆ กันคือ พิจารณาว่าจะนำ Software Component ไปไว้ใน Hardware Module ตัวใด และต้องพิจารณาด้วยว่า เทคโนโลยีหรือรูปแบบของฮาร์ดแวร์หรือระบบ Network ที่จะใช้นั้น เหมาะสมกับระบบงานที่เราจะสร้างหรือไม่

ข้อพึงตระหนักที่สำคัญที่สุดในการทำ System Architecture Design ก็คือ การออกแบบระบบฮาร์ดแวร์ เป็นการออกแบบที่ไม่ควรเกิดข้อผิดพลาด หรือถ้าเกิดก็ควรจะน้อยที่สุด เพราะถึงแม้ว่าจะยึดถือ หลักการวิเคราะห์และออกแบบระบบแบบ Spiral Process Model แต่การแก้ไขระบบฮาร์ดแวร์ย่อมหมายถึงการซื้อหรือเปลี่ยนอุปกรณ์ฮาร์ดแวร์ ซึ่งบางกรณีต้องใช้การลงทุนที่มีมูลค่าสูง

2.15 แผนภาพคอมโพแนนท์

แผนภาพคอมโพแนนท์ นั้นใช้อธิบายโครงสร้างทางด้านซอฟต์แวร์รวมทั้งยังอาจจะอธิบายโครงสร้างของโค้ดด้วย ในไดอะแกรมนี้จึงประกอบด้วยแพคเกจ(Package), ทรัค(Task), และอาจรวมถึงคลาส (Class) ด้วย

แผนภาพคอมโพแนนท์เป็น Static Diagram ที่ใช้สำหรับจำลองลักษณะทางกายภาพของ Object-Oriented System โดยจะแสดงให้เห็นถึง "ส่วนประกอบทาง Software" (Software Components – เรียกสั้นๆ ว่า Components) ต่างๆ ของระบบ รวมถึงความสัมพันธ์ระหว่างคอมโพแนนท์ต่างๆ ด้วย แผนภาพคอมโพแนนท์จะมีความสัมพันธ์กับ Class Diagram เพราะในแต่ละคอมโพแนนท์นั้น จะประกอบไปด้วยคลาส ที่อยู่ในระบบตั้งแต่ 1 คลาส ขึ้นไป

หลังจากที่ได้ทำ Refinement กับไดอะแกรมต่างๆ แล้ว สิ่งที่ต้องทำในขั้นตอนถัดมาคือ การออกแบบโครงสร้างของซอฟต์แวร์หรือ Application ที่เราจะสร้างต่อไป โดยใช้ไดอะแกรมต่างๆ เป็นวัตถุดิบและแนวทาง เรียกกระบวนการนี้ว่า Application Architecture Design

Application Architecture Design คือกระบวนการที่ดึงเอาผลลัพธ์ต่างๆ ที่ได้จารการทำ Object-Oriented Analysis ซึ่งได้แก่ไดอะแกรมต่างๆที่ผ่านการทำ Refinement แล้ว มาเป็นวัตถุดิบในการออกแบบส่วนประกอบของ Application ที่จะมีในระบบ

คอมโพแนนท์แต่ละตัวนั้นสามารถแบ่งย่อยออกเป็น คอมโพแนนท์ย่อยๆ ได้เช่นเดียวกัน ซึ่งเมื่อพิจารณาโดยละเอียดแล้ว แผนภาพคอมโพแนนท์ก็จะสามารถจำลองลักษณะการทำงานที่มี Routine ย่อยๆ อยู่ในโปรแกรมหลักได้

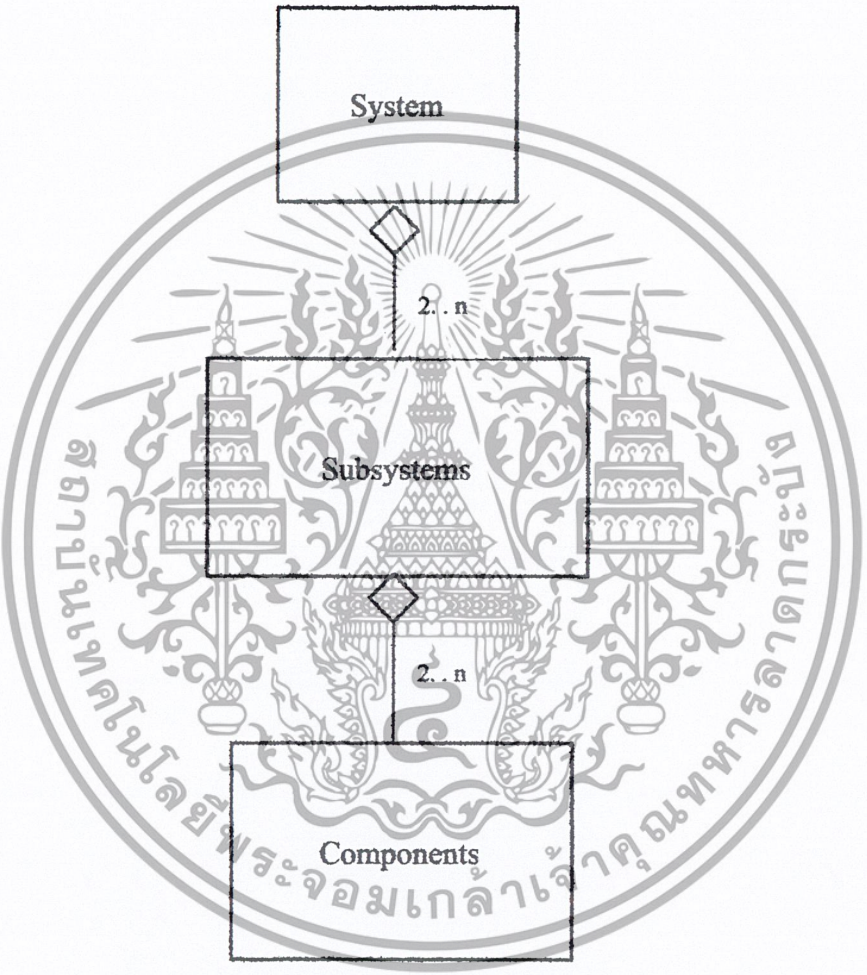
ประโยชน์ที่สำคัญของแผนภาพคอมโพแนนท์ คือ สามารถแบ่งระบบงานขนาดใหญ่ (System) ออกเป็นระบบย่อยๆ (Subsystem) ซึ่งแต่ละระบบย่อยๆ ก็จะมีคอมโพแนนท์ต่างๆ ประกอบอยู่ การแบ่งระบบออกเป็นระบบย่อยๆ เป็นที่ยอมรับโดยทั่วไปแล้วว่า สามารถทำให้การพัฒนาระบบมีประสิทธิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

45823

ภาพ และสนับสนุนหลักการพัฒนาระบบงานแบบเป็นทีมงานที่สามารถแบ่งส่วนย่อยต่างๆ ให้แต่ละส่วนงานย่อยไปรับผิดชอบได้

สามารถแสดงการแบ่งระบบออกเป็นระบบย่อยๆ และคอมโพแนนท์ได้ดัง Class Diagram ที่แสดงอยู่ในรูป

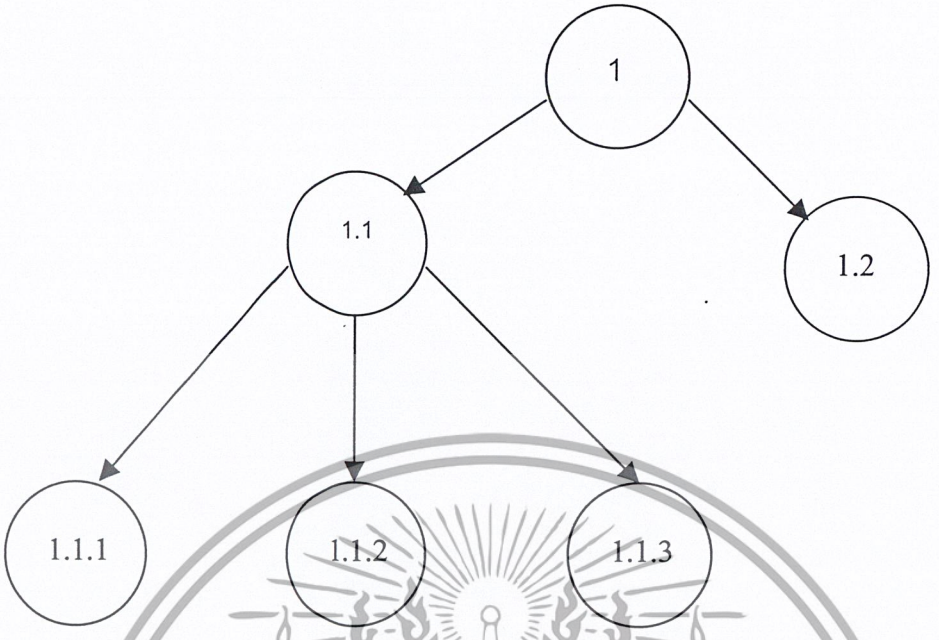


รูปภาพ 2.5 Class diagram ที่จำลองการแบ่งย่อยระบบ

System Decomposition

การแบ่งระบบใหญ่ออกเป็นระบบย่อยๆ หรือที่เรียกว่า System Decomposition นั้น มีด้วยกันหลายวิธี แต่วิธีที่ใช้ในหนังสือเล่มนี้ก็คือ การแบ่งย่อยระบบแบบ Drill Down คือพยายามแบ่งระบบใหญ่ออกเป็นระบบย่อยๆ โดยทำการขุดเจาะลงไปในระบบย่อยแต่ละส่วนไปจนกระทั่งถึงขั้นละเอียดที่สุด หรือละเอียดจนกว่าที่ทีมงานเห็นสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Drill Down Engineering

การแบ่งย่อยระบบนั้นสามารถทำได้หลายรูปแบบ สำหรับวิธีการ Drill Down Engineering จะแบ่งระบบใหญ่ออกเป็น 3 ส่วนหลักๆ ดังนี้

1. Presentation Logic System

Subsystem ในส่วนนี้ เป็น Subsystem ที่เกี่ยวข้องกับระบบย่อยที่ติดต่อกับผู้ใช้งานระบบโดยตรง ซึ่งได้แก่ ส่วนที่ติดต่อกับผู้ใช้ (User Interface) ส่วนของการแสดงผลลัพธ์ (Output) และส่วนของการนำเข้า (Input)

2. Working Logic Subsystem หรือ Business Logic Subsystem

Working Logic Subsystem คือส่วนของระบบที่เกี่ยวข้องกับการทำงานที่เกิดขึ้นจริงๆ ใน CPU ของเครื่องคอมพิวเตอร์ อาทิ ส่วนของการคำนวณ กลไกการจัดดึงข้อมูลที่ได้รับมาเพื่อมาใช้งาน กลไกการบันทึกข้อมูล เป็นต้น

3. Database Logic Subsystem

Database Logic Subsystem จำลองภาพของ Data Item ต่างๆ ที่ถูกจัดเก็บอยู่ในสื่อบันทึกข้อมูล ไม่ว่าจะเป็นในรูปของ Database หรือ File ก็ตาม ซึ่งการออกแบบ Database Logic Subsystem นี้จะมีส่วนสัมพันธ์กับ Persistent Data Design

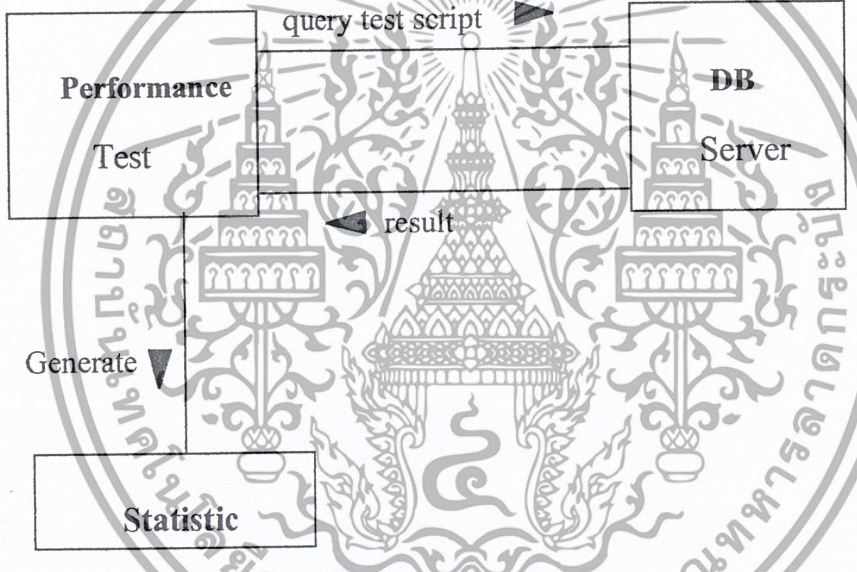
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 ผลการออกแบบ

จากโครงสร้างและขอบเขตของปัญหาดังที่ได้กล่าวไว้ในบทที่ 1 ได้ทำการวิเคราะห์และออกแบบโปรแกรมโดยวิธีการของ "การพัฒนาระบบเชิงวัตถุ" ซึ่งได้ส่วนต่างดังต่อไปนี้

3.1.1 Client interview



รูปภาพ 3.1 แสดงไคลแอนต์อินเทอร์วิว

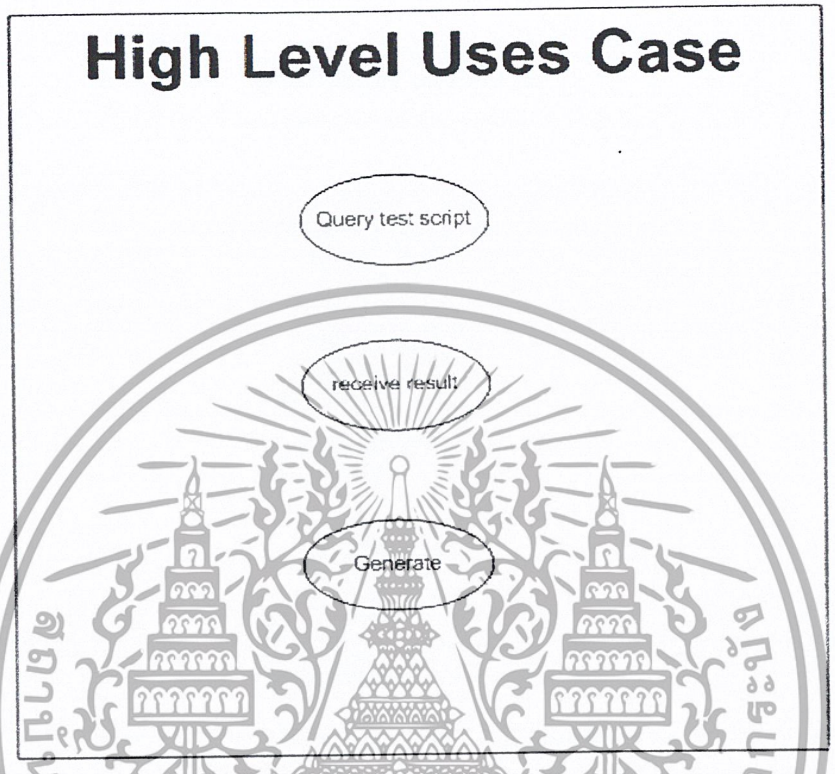
จากมุมมองของผู้ใช้งานจะเห็นส่วนประกอบที่สำคัญภายในระบบ 3 ส่วน คือ

1. ตัวโปรแกรม (Performance test)
2. เซิร์ฟเวอร์ของฐานข้อมูล (Database Server) เป็นส่วนที่ต้องทำการทดสอบ
3. ตัวเก็บข้อมูล (Statistic)

โครงสร้างการทำงานโดยรวมที่ได้จาก Client interview คือเริ่มจากตัวโปรแกรมส่งชุดคำสั่ง (query test script) ต่าง ๆ ไปยังเซิร์ฟเวอร์ของฐานข้อมูลเมื่อตัวเซิร์ฟเวอร์ทำตามคำสั่งเรียบร้อยแล้วก็จะส่งผลลัพธ์ (result) กลับมายังตัวโปรแกรมซึ่งตัวโปรแกรมจะทำการประมวลผลถึงขีดความสามารถของเซิร์ฟเวอร์ของฐานข้อมูล และจะทำการสร้างและแสดงข้อมูลต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 High level use case



รูปภาพ 3.2 แสดงแผนภาพ High level use case

High level use case นั้นสามารถหาได้จาก Client interview โดยพิจารณาจากการกระทำกันระหว่างวัตถุใน Client Interveiw ซึ่งเป็นการกระทำที่ไม่ได้เกิดจากผู้ใช้งาน High level use case ในที่นี้จะประกอบไปด้วยกระบวนการต่าง ๆ ดังนี้

ส่งชุดคำสั่ง (query test script) เป็นการกระทำระหว่าง ตัวโปรแกรม และ เซิร์ฟเวอร์ของฐานข้อมูล

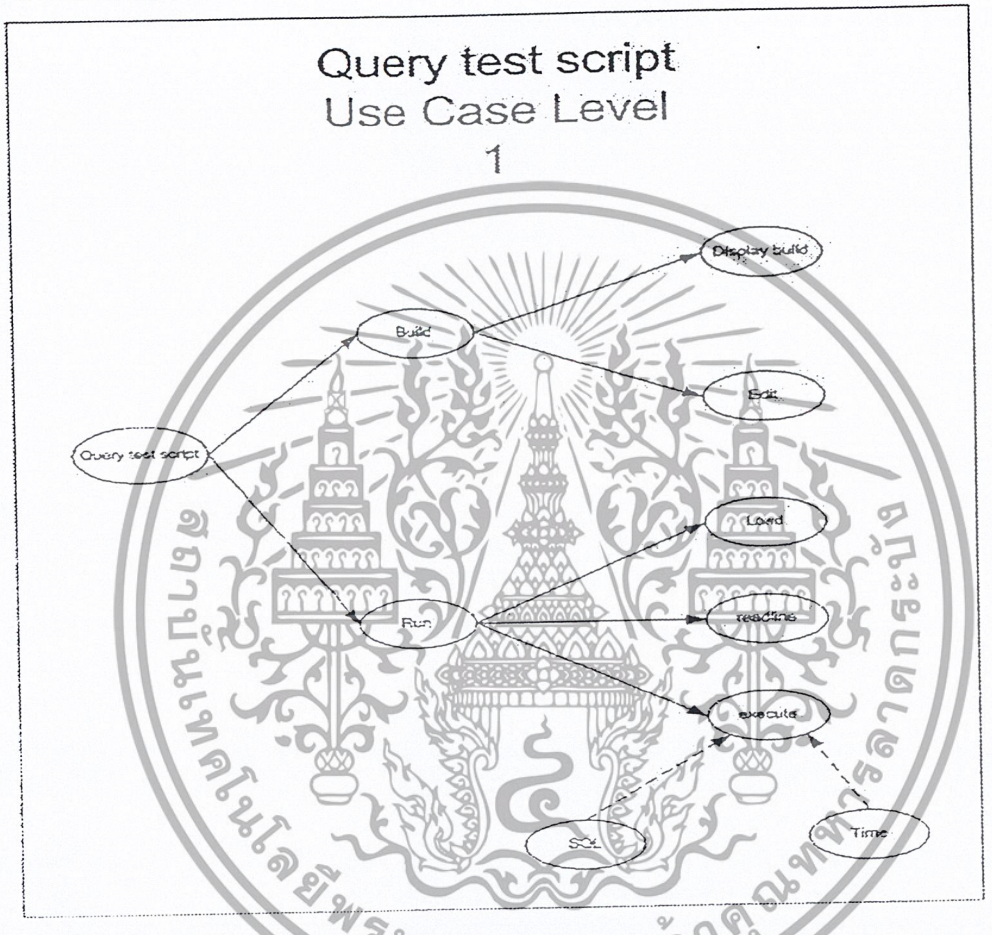
ส่งผลลัพธ์ (result) เป็นการกระทำระหว่าง เซิร์ฟเวอร์ของฐานข้อมูลและ ตัวโปรแกรม

การสร้างแสดงข้อมูล (generate) เป็นการกระทำระหว่าง เซิร์ฟเวอร์ของฐานข้อมูลและ ตัวเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ยูสเคส ระดับ 1

หลังจากการเขียน high level use case แล้วก็จะทำการแตกย่อยส่วนประกอบภายในของแต่ละยูสเคสออกมาเพื่อให้เห็นถึงการทำงานภายในที่ละเอียดมากยิ่งขึ้นซึ่งเป็นการทำยูสเคสอีกระดับหนึ่งที่ย่อยออกมา



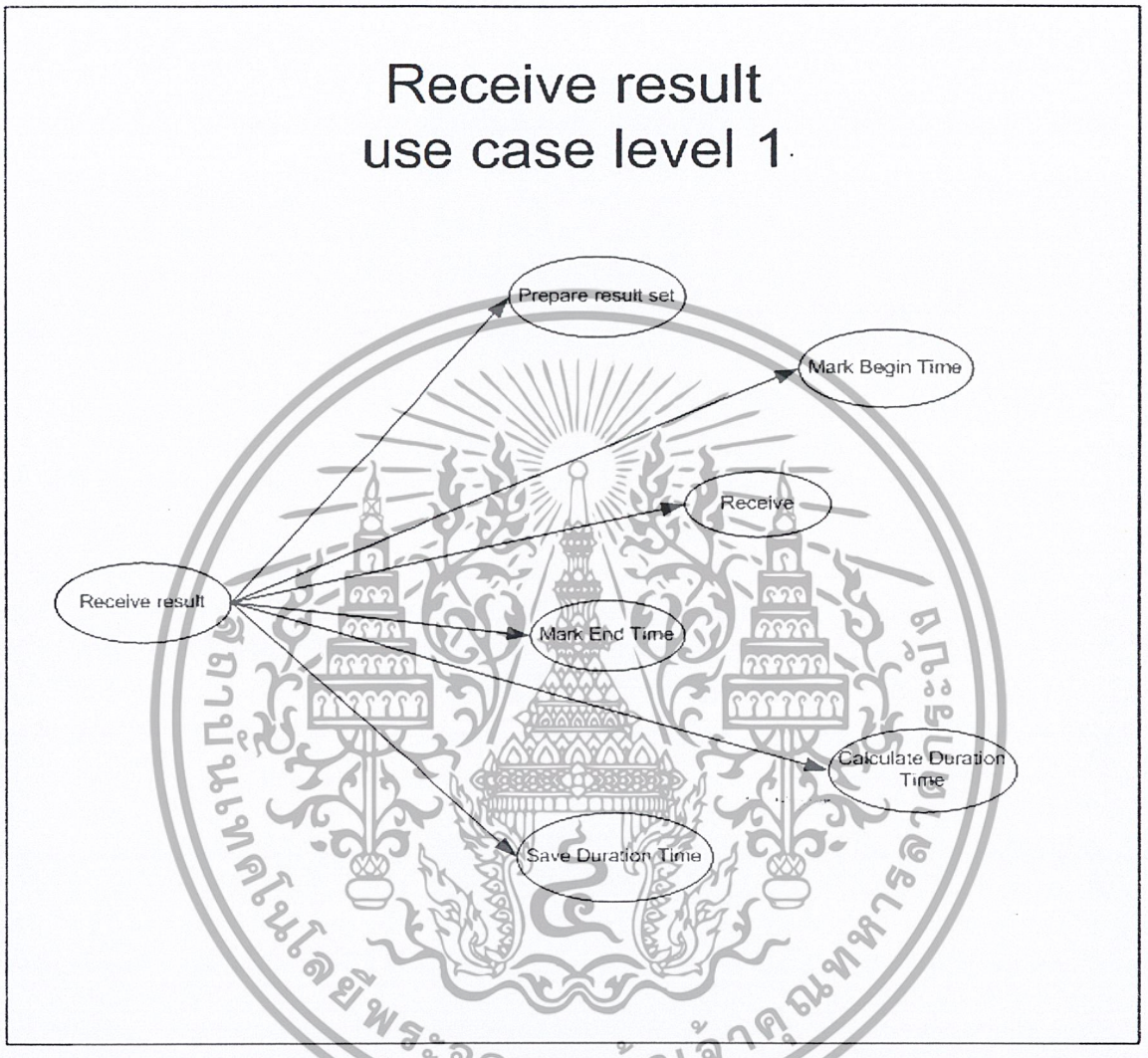
รูปภาพ 3.3 แสดงแผนภาพยูสเคส ระดับ 1 ของชุดคำสั่ง

ส่งชุดคำสั่ง (query test script) จะแยกย่อยออกได้เป็น 2 การกระทำใหญ่ ๆ การสร้าง (Build) และการดำเนินการ(Run)

ในการสร้างนั้นจะมีหน้าที่คือเริ่มจากแสดงหน้าจอ(Display) เพื่อติดต่อกับผู้ใช้งานรับข้อมูลต่าง ๆ รวมถึงการแก้ไข(Edit) การดำเนินงานต่าง ๆ

การดำเนินการจะเริ่มตั้งแต่การเก็บรวบรวมข้อมูลของผู้ใช้ (Load) โดยการอ่าน (Read Line) แล้วส่งต่อไปยังการจัดการ(Execute) ซึ่งจะมีการใส่คำสั่งของเอสคิวแอล (SQL) และการดำเนินการจับเวลา (time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

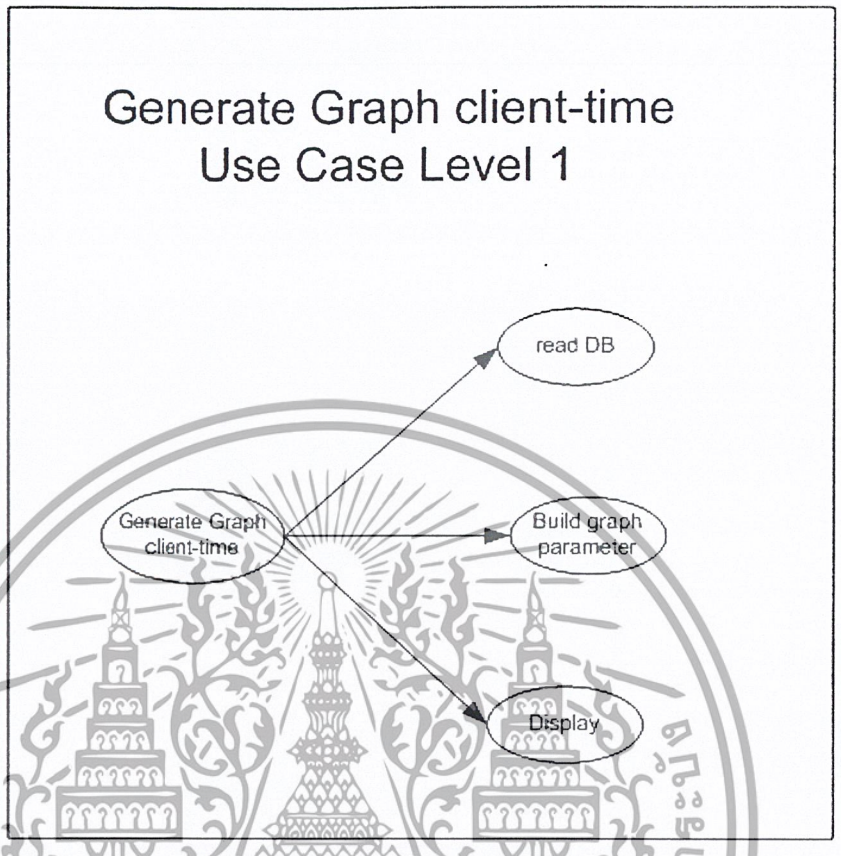


รูปภาพ 3.4 แสดงแผนภาพยูสเคส ระดับ 1 ของการส่งผลลัพธ์

ส่งผลลัพธ์ (result) เมื่อเซิร์ฟเวอร์ของฐานข้อมูลส่งการติดต่อกลับมาตัวโปรแกรมจะทำกระบวนการต่าง ๆ ดังนี้

สร้างตัวรับข้อมูล(Perpare result set) ดึงเวลาเริ่มต้น(Mark begin time) รับข้อมูล(Receive) หยุดเวลา (Mark end time) คำนวณช่วงเวลา(Calculate Duration time) และ เก็บค่าช่วงเวลา(Save Duration time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

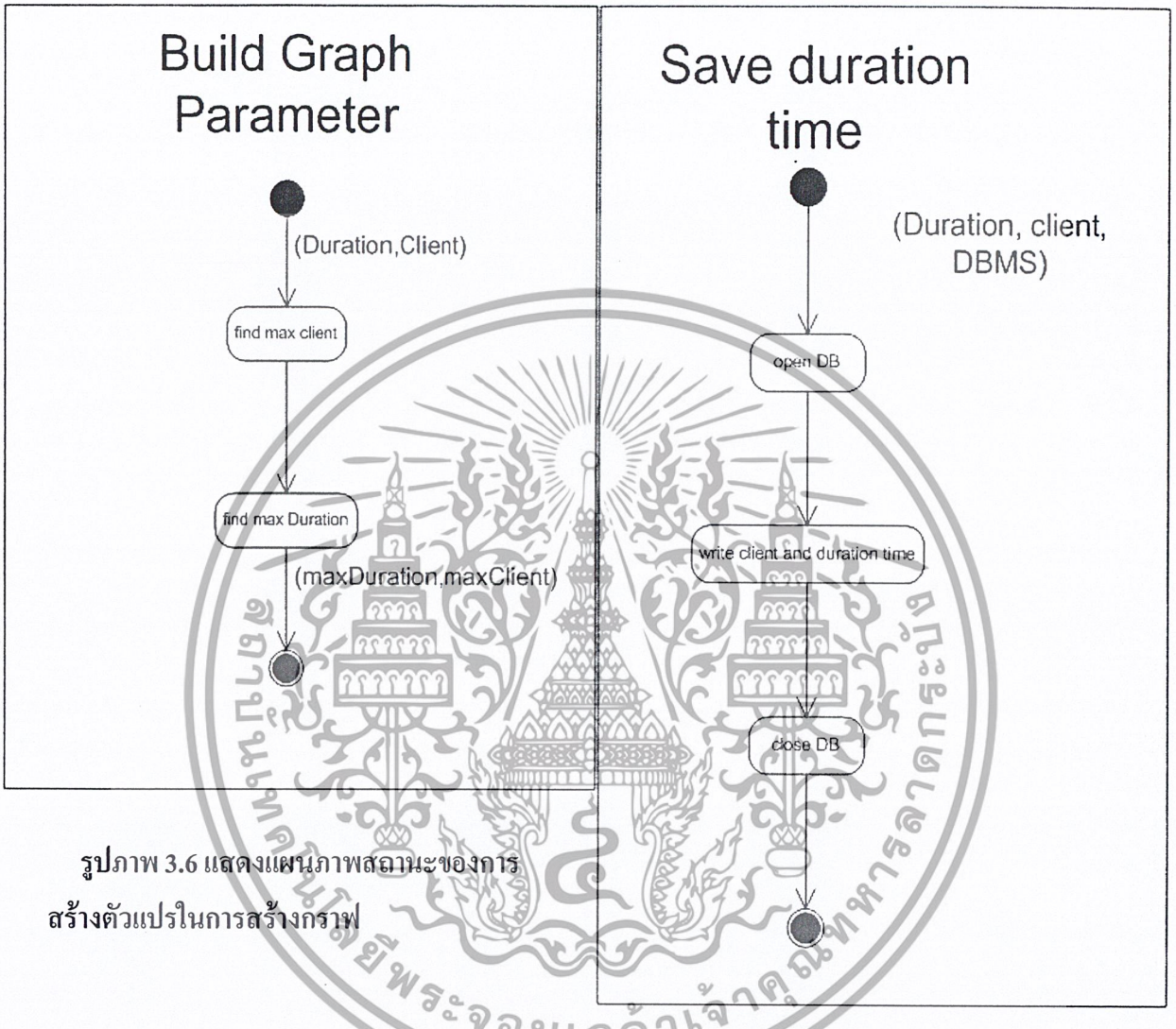


รูปภาพ 3.5 แสดงแผนภาพยูสเคส ระดับ 1 ของการแสดงข้อมูล

การสร้างแสดงข้อมูล (generate)จะเป็นการแสดงข้อมูลต่าง ๆ ในรูปของกราฟและตารางข้อมูล โดยกระบวนการเริ่มจาก อ่านข้อมูลต่างๆที่เก็บเอาไว้ (read DB) สร้างตัวแปรกราฟ (Build graph parameter) แสดงผล(Display)

หลังจากนั้นจะทำการกระจายยูสเคสที่ยังไม่ละเอียดเพียงพอต่อการที่จะนำไปเขียนโปรแกรม หรือทำความเข้าใจโดยวิธีการสร้าง Activity Diagram โดยได้แสดงไว้ในรูปที่ 3.6 - 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 3.6 แสดงแผนภาพสถานะของการสร้างตัวแปรในการสร้างกราฟ

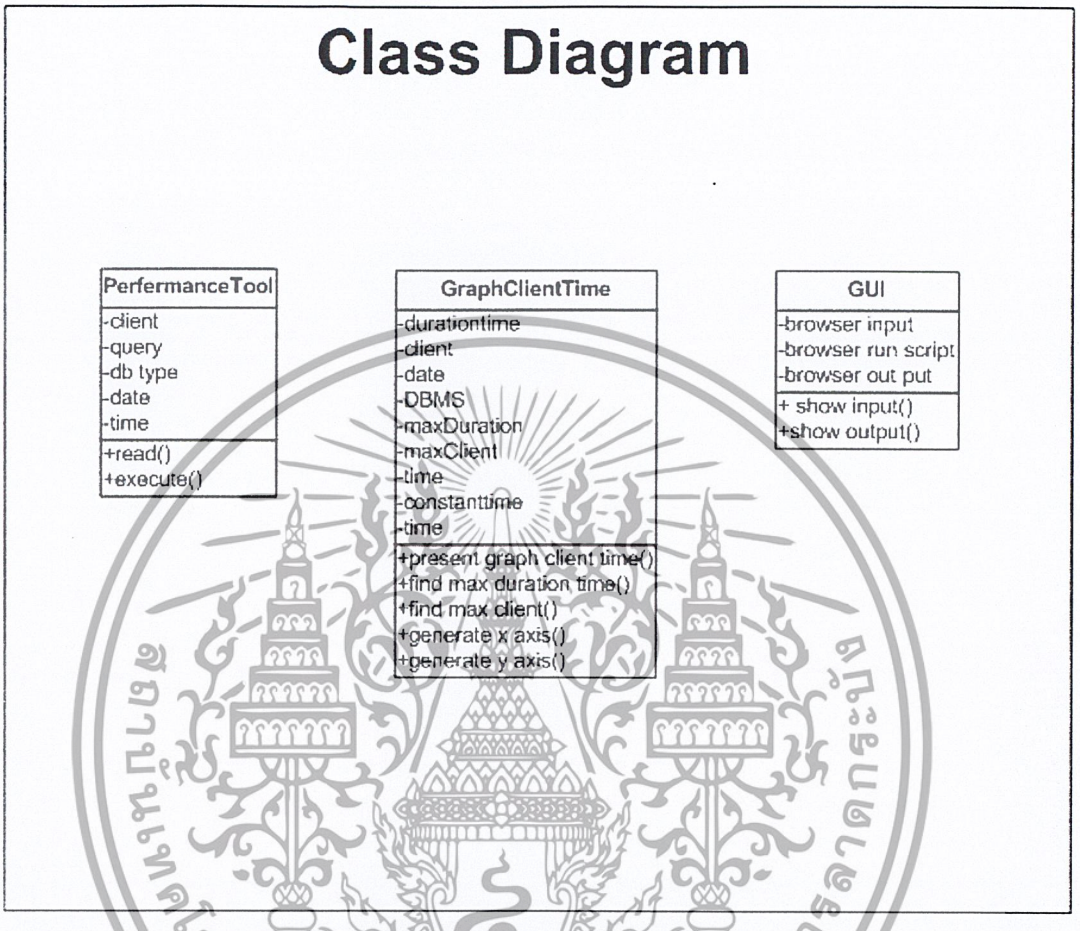
รูปภาพ 3.7 แสดงแผนภาพสถานะของการเก็บค่าของเวลาการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 3.8 แสดงแผนภาพสถานะของการแสดงผล

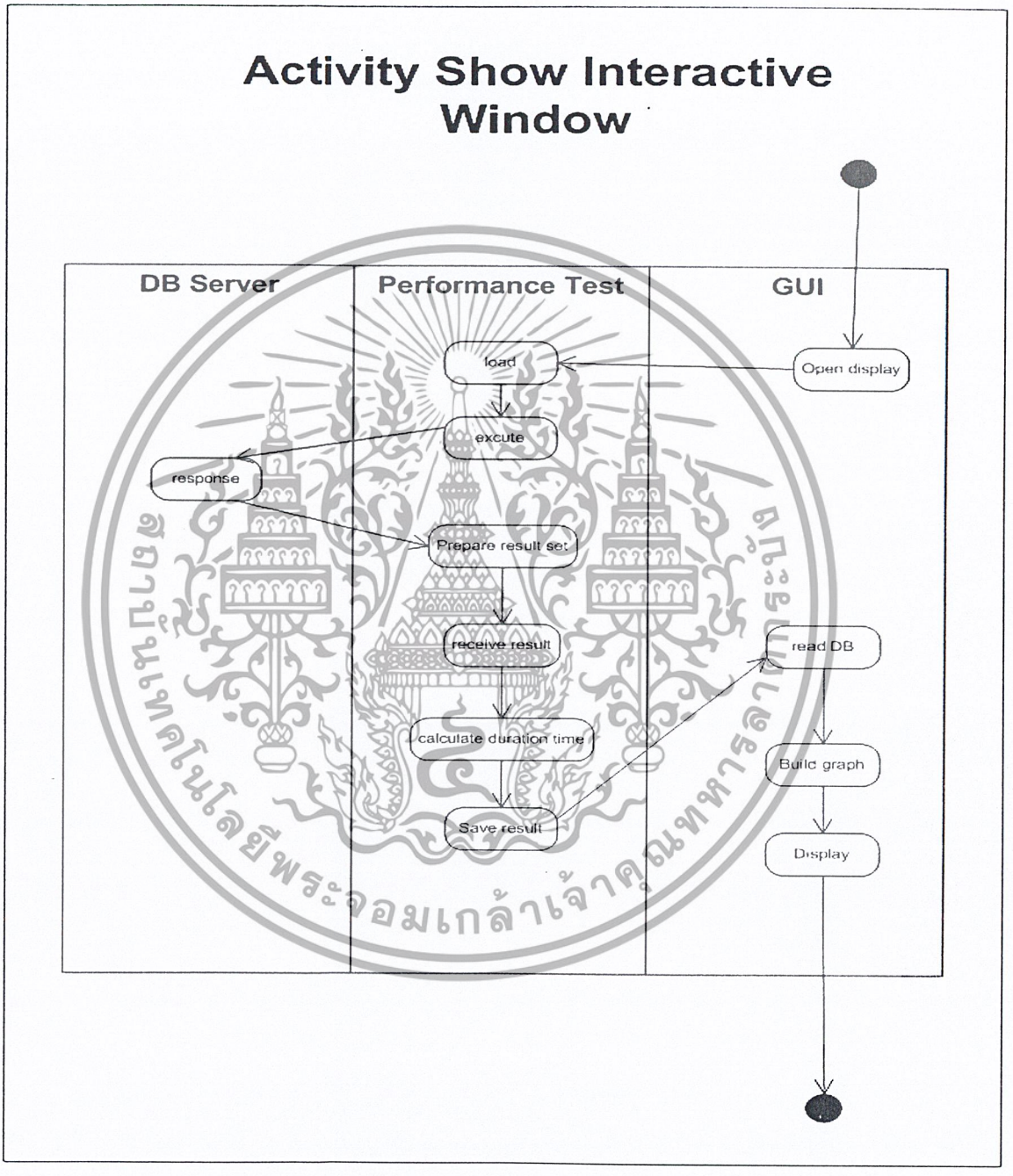
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 3.9 แสดงแผนภาพคลาสของระบบ

ในรูปที่ 3.9 เป็นการแสดงภาพของคลาสไดอะแกรม ในที่นี้จะประกอบไปด้วย คลาส 3 คลาส ได้แก่ คลาสของตัวโปรแกรม(Performance Tool) คลาสของหน้าจอ(GUI) และคลาสของการแสดงผล (Graph Client-Time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพ 3.10 แสดงแผนภาพกิจกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 วิธีการทำงานของโปรแกรม

โปรแกรมจะมีการทำงานแบ่งออกเป็น 2 ส่วนคือ

1. ส่วน Master
2. ส่วน Slave

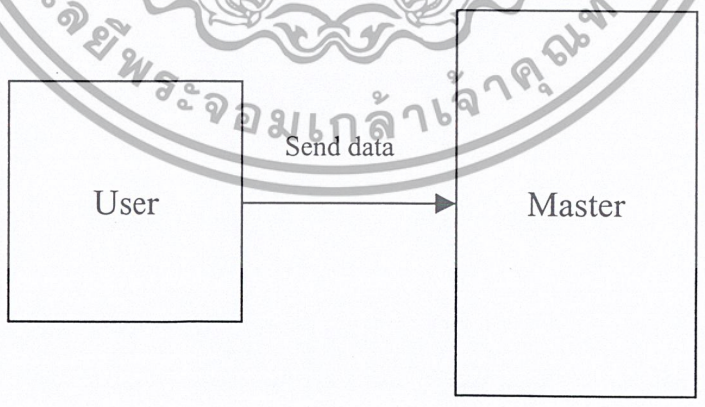
ในส่วนของ master จะมีหน้าที่รับข้อมูลที่จะใช้ทำการทดสอบจากผู้ใช้ระบบ ซึ่งจะข้อมูล เช่น ชื่อของฐานข้อมูล , จำนวนที่จะทดสอบ , คำสั่งที่จะใช้ทดสอบ , เวลาที่กำหนดในการทำงาน เป็นต้น ซึ่งเมื่อได้รับข้อมูลแล้ว ส่วน master จะทำการส่งข้อมูลไปควบคุมส่วน slave ให้เปิดพอร์ทตามที่ master ต้องการ เมื่อได้แล้วจะส่งข้อมูลที่จำเป็นต้องใช้ในการติดต่อให้กับ slave ซึ่ง slave จะไม่ทำการติดต่อโดยทันทีแต่จะรอคำสั่งจาก master ก่อน

เมื่อ master ทำการควบคุมให้ slave เปิดพอร์ทครบแล้วจึงจะทำการควบคุมให้ slave ทั้งหมดเริ่มการทำงานพร้อมกัน

เมื่อทำงานเสร็จล้ว slave แต่ละตัวจะทำการส่งข้อมูลเวลาที่ใช้คืนให้กับส่วน master ซึ่ง master จะนำข้อมูลที่ได้นี้ไปประมวลผลและนำมาแสดงเป็นกราฟต่อ ไป

รูปแบบการทำงานจะแสดงดังรูปต่อไปนี้

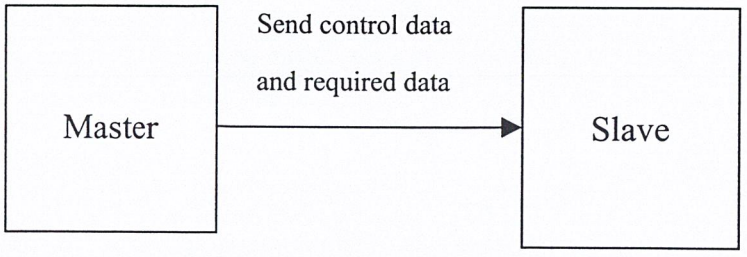
1.



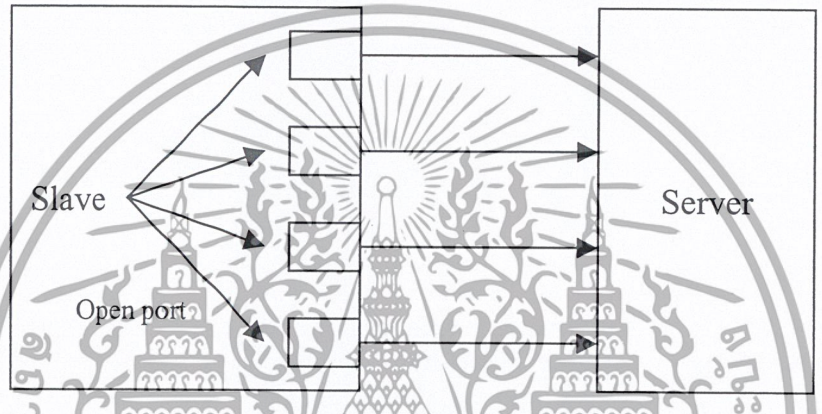
รูปที่ 4.1 ขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

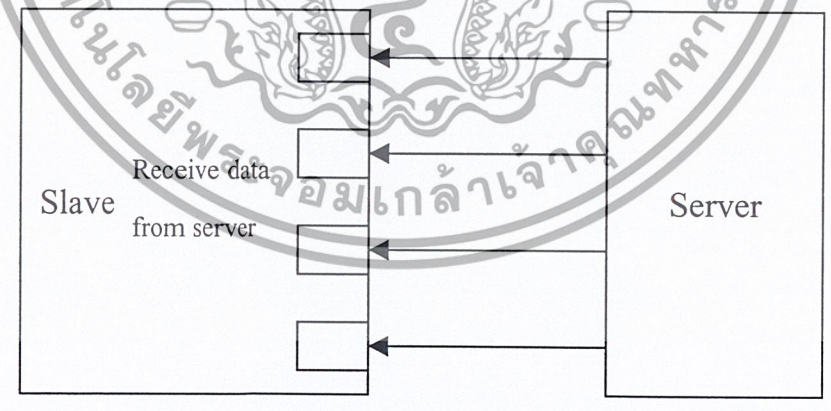
2.



3.



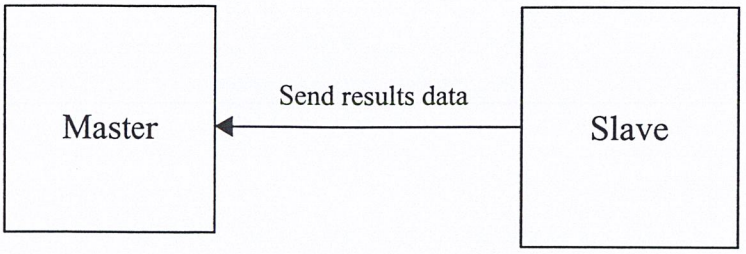
4.



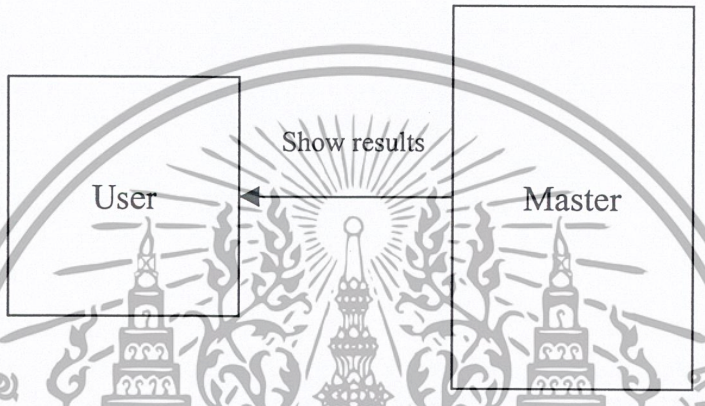
รูปที่ 4.1(ต่อ) ขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.



6.



รูปที่ 4.1(ต่อ) ขั้นตอนการทำงานของโปรแกรม

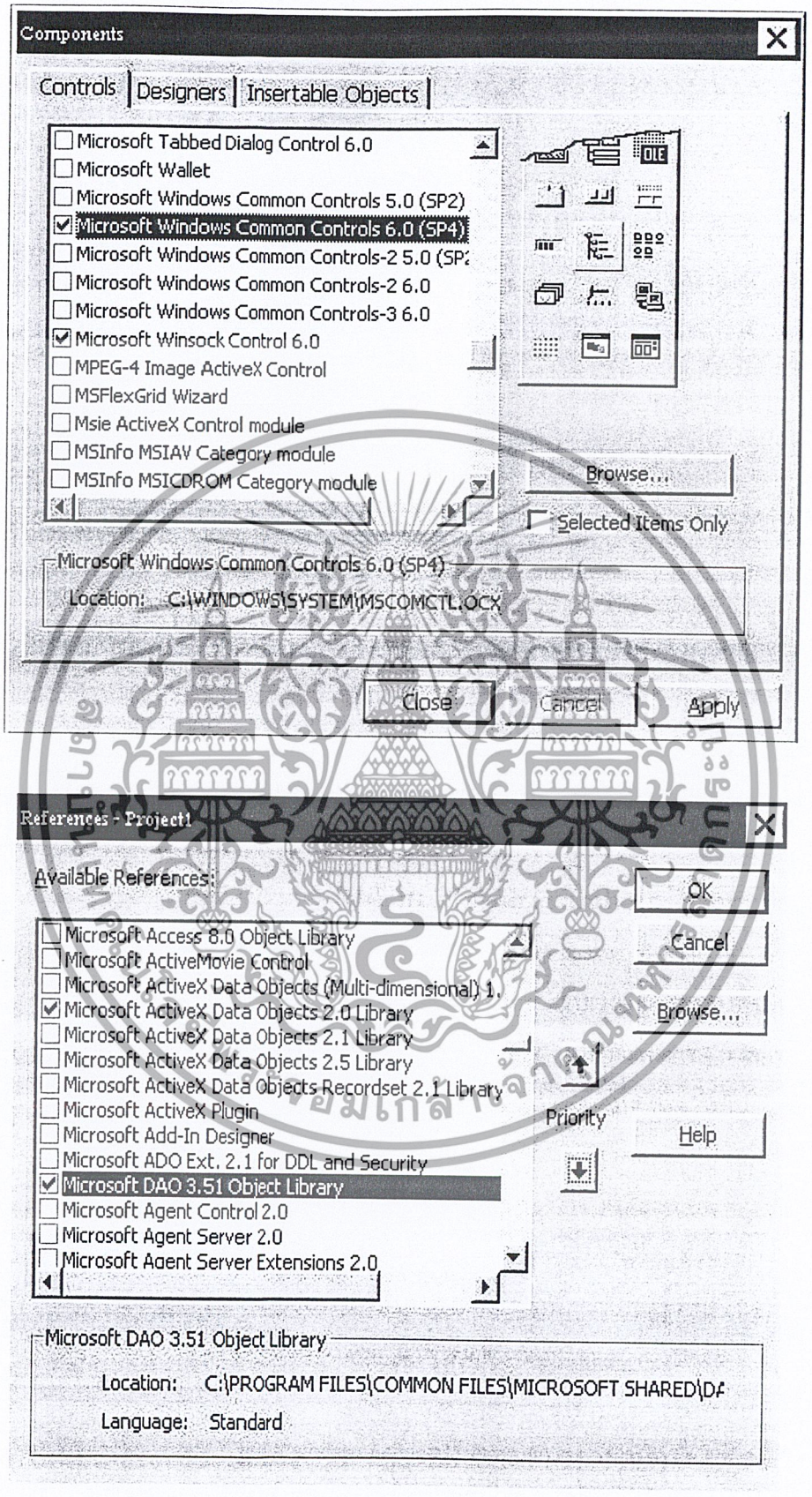
4.2 ตัวอย่างของเครื่องมือ

ที่จะแสดงต่อไปนี้เป็นตัวอย่างของโปรแกรมซึ่งพัฒนาขึ้นโดยใช้ภาษา Visual Basic 6.0 และส่วนหลักของโปรแกรมที่ใช้ในการจำลองการทำการทดสอบประสิทธิภาพของเซิร์ฟเวอร์ (การให้ master ควบคุมการทำงานของ slave เพื่อส่งการร้องขอไปยังเซิร์ฟเวอร์โดยใช้พอร์ทการส่งข้อมูลที่แตกต่างกันเสมือนผู้ใช้งานคนละกลุ่มกัน) จะใช้คอมโพแนนท์ Microsoft Winsock Control 6.0 และ Microsoft Windows Common Controls 6.0 (SP4)

ส่วนของโปรแกรมที่ใช้ในการสร้างการติดต่อและติดต่อกับฐานข้อมูลจะใช้คอมโพแนนท์ Microsoft ActiveX Data Objects 2.0 Library และ Microsoft DAO 3.5 Object Library

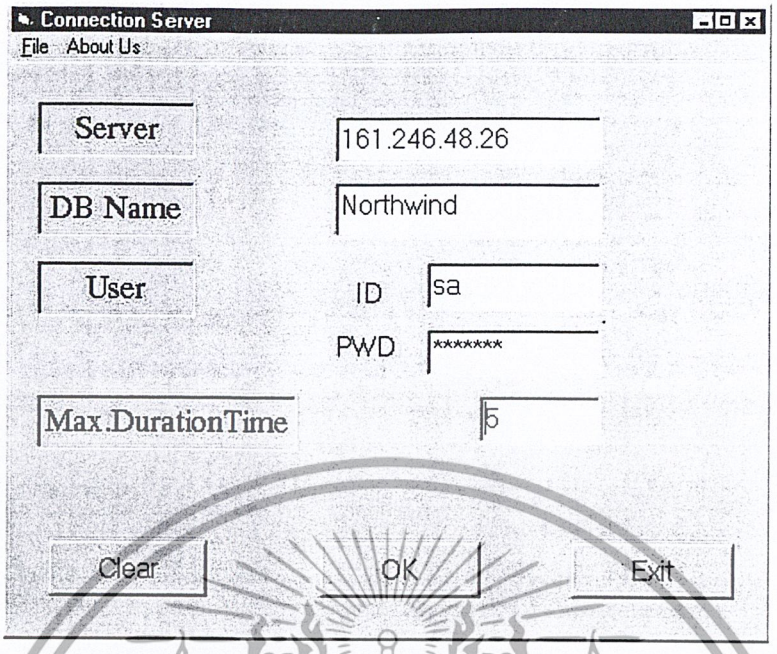
ส่วนของโปรแกรมที่ใช้ในการสร้างกราฟจะใช้คอมโพแนนท์ IOcomp plot pack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงคอมโพแนนท์ที่ใช้ในการพัฒนาเครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงหน้าจอเริ่มต้นของเครื่องมือ

ในรูปที่ 4.3 จะแสดงหน้าจอเริ่มต้นจะมีการรับข้อมูลดังนี้

- Server รับชื่อของเซิร์ฟเวอร์
- DB Name รับชื่อของฐานข้อมูล
- User ID รับ user id ที่ใช้ในการติดต่อฐานข้อมูล
- User PWD รับ user password ที่ใช้ในการติดต่อฐานข้อมูล
- Max.DurationTime รับเวลาสูงสุดที่กำหนดจะใช้ในการทดสอบ

ข้อมูลที่รับเข้ามาจะอยู่ในส่วน master ซึ่ง master จะส่ง Server name, DB name, UserID, UserPassword ไปยังส่วน slave เพื่อใช้ในการติดต่อกับฐานข้อมูล

เมื่อคลิกที่ปุ่ม Clear จะทำการลบ text box ทั้งหมด

เมื่อคลิกที่ปุ่ม OK จะทำการรับข้อมูลและไปยัง form การรับสคริปต์และ delay time

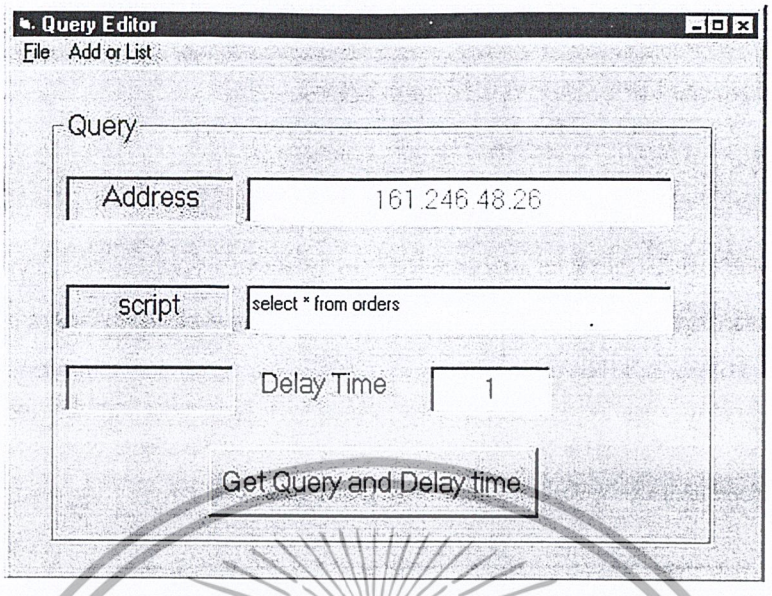
เมื่อคลิกที่ปุ่ม Exit จะทำการออกจากโปรแกรม

ในรูปที่ 4.4 แสดงหน้าจอการรับสคริปต์ที่ใช้ทดสอบและdelay time

- Address แสดงแอดเดรสของเซิร์ฟเวอร์
- Script รับสคริปต์ที่ใช้ในการทำการทดสอบ
- Delay Time รับค่าหน่วงเวลาเป็นมิลลิวินาที

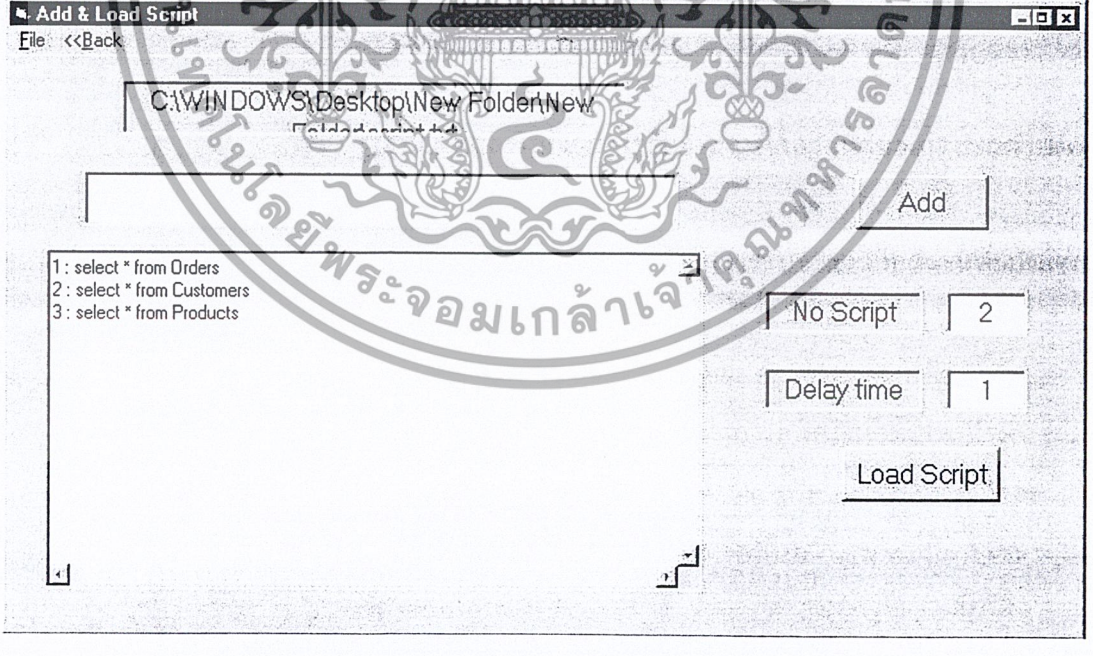
เมื่อคลิกที่ปุ่ม get query and delay time จะทำการรับสคริปต์และค่าหน่วงเวลาและไปยังหน้าจอการรับจำนวนผู้ใช้ดังในรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงหน้าจอการรับสคริปต์ที่ใช้ทดสอบและdelay time

ในส่วนของเมนูบาร์ Add or List จะใช้ในการรับสคริปต์จากไฟล์ที่มีอยู่แล้วซึ่งมีหน้าจอแสดงในรูปที่ 4.5



รูปที่ 4.5 แสดงหน้าจอการรับสคริปต์จากไฟล์ที่มีอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

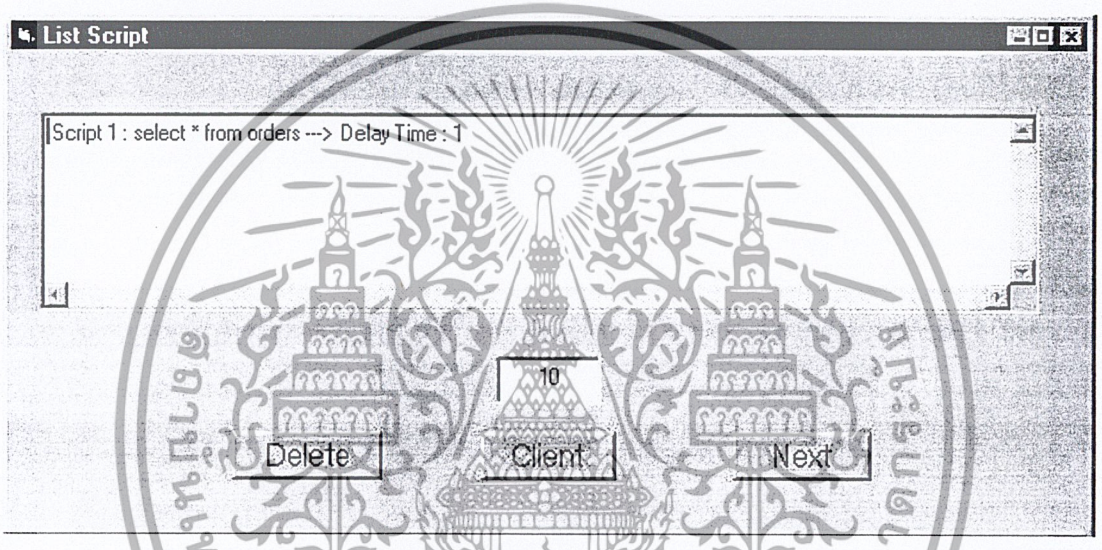
ในรูปที่ 4.5แสดงหน้าจอการรับสคริปต์จากไฟล์ที่มีอยู่แล้ว

- No Script รับค่าหมายเลขของสคริปต์นั้นๆ
- Delay time รับค่าหน่วงเวลาของสคริปต์

เมื่อคลิกที่ปุ่ม Add จะทำการเก็บสคริปต์ใน text box เข้าไปในลิสต์

เมื่อคลิกที่ปุ่ม Load Script จะรับค่าของสคริปต์พร้อมด้วย Delay time

ในส่วนของเมนูบาร์ Back จะใช้กลับไปยังหน้าจอในรูปที่ 4.4 .



รูปที่ 4.6 แสดงหน้าจอการรับจำนวนไคลแอนต์ของแต่ละสคริปต์และแสดงข้อมูลของสคริปต์ที่เก็บมา

ในรูปที่ 4.6 แสดงหน้าจอการรับจำนวนไคลแอนต์ของแต่ละสคริปต์และแสดงข้อมูลของสคริปต์ที่เก็บมา

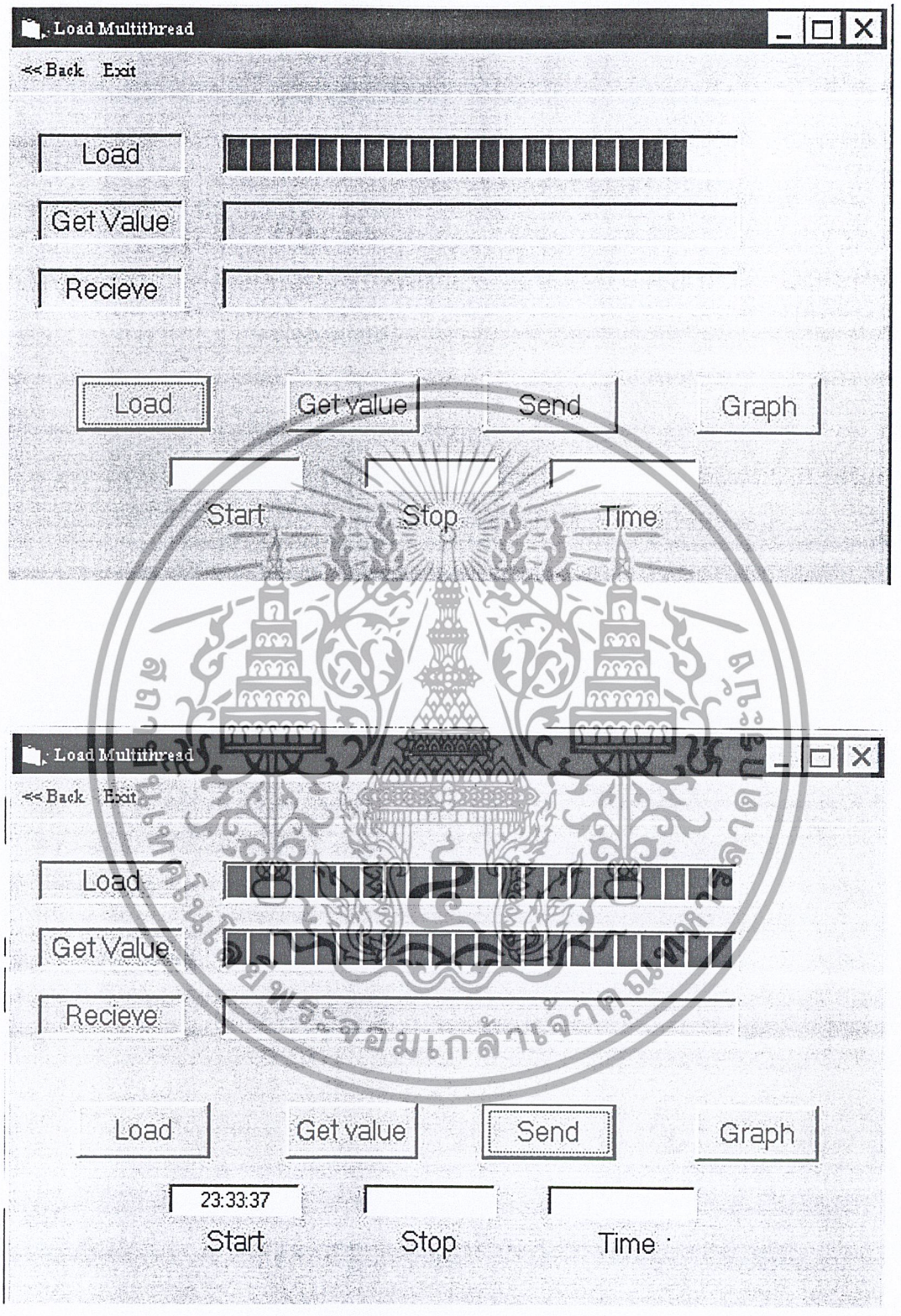
ใน text box เหนือปุ่ม Client จะใช้รับค่าจำนวนไคลแอนต์

เมื่อคลิกที่ปุ่ม Delete จะทำการลบค่าใน text box

เมื่อคลิกที่ปุ่ม Client จะทำการเก็บจำนวนไคลแอนต์ของสคริปต์ที่แสดง

เมื่อคลิกที่ปุ่ม Next จะทำการเก็บจำนวนไคลแอนต์ของสคริปต์ที่แสดงและไปยังหน้าจอเพื่อทำการทดสอบดังในรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงหน้าจอการทำการทดสอบประสิทธิภาพของเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4.7 แสดงหน้าจอการทำการทดสอบประสิทธิภาพของเซิร์ฟเวอร์ในส่วนนี้จะมีการนำข้อมูลจากผู้ใช้มาเก็บในโปรแกรมส่วน master และทำการส่งข้อมูลจากส่วน master ไปยังส่วน slave แล้วจึงส่งข้อมูลออกไปทดสอบ

ในโพรเกสบาร์ Load จะแสดงความคืบหน้าในการเก็บข้อมูลที่รับมาเข้าสู่ส่วน master เมื่อคลิกที่ปุ่ม Load

ในโพรเกสบาร์ Get Value จะแสดงความคืบหน้าในการส่งข้อมูลจากส่วน master ไปยังส่วน slave เพื่อเตรียมพร้อมทำการทดสอบเมื่อคลิกที่ปุ่ม Get Value

ในโพรเกสบาร์ Receive จะแสดงความคืบหน้าในการทดสอบโดยจะแอกทีพเมื่อทำการเริ่มทดสอบโดยคลิกที่ปุ่ม Send

เมื่อคลิกที่ปุ่ม Graph จะทำการแสดงผลการทดสอบในรูปของกราฟ

Text box Start จะแสดงเวลาจริงที่เริ่มทดสอบ

Text box Stop จะแสดงเวลาจริงที่เสร็จสิ้นการทดสอบ

Text box Time จะแสดงเวลาทั้งหมดที่ใช้ในการทดสอบ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ปัญหาที่เกิดขึ้นและแนวทางการพัฒนาต่อ

5.1ปัญหาที่พบในการทำงาน

ปัญหาของการทำงานแบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

1) ส่วนของฮาร์ดแวร์

เนื่องจากโปรแกรมที่จะต้องทำการเขียนขึ้นมานั้นต้องทำการติดต่อกับระบบฐานข้อมูลเซิรฟเวอร์ดังนั้นเมื่อทำการเขียนโปรแกรมขึ้นมาจึงต้องทำการหาเซิรฟเวอร์ที่มีระบบฐานข้อมูลเพื่อทำการทดสอบโปรแกรม

2) ส่วนของซอร์ฟแวร์

ประสบปัญหาการนำ Visual Basic 6.0 มาออกแบบการทำงานแบบ Multithreading ของโปรแกรม ซึ่งส่วนนี้ตัวโปรแกรมไม่ได้มีไว้ให้และการเขียนโปรแกรมให้ทำงานแบบ Multithreading โดยใช้ Visual Basic 6.0 เป็นเรื่องที่ซับซ้อนซึ่งผู้ศึกษาได้ศึกษาและทดลองหลายๆ วิธีแล้ว แต่ยังไม่ได้ผลทดสอบที่น่าพอใจ ผู้ศึกษาจึงได้เปลี่ยนวิธีการเขียนโปรแกรม แต่มีรูปแบบการทำงานของโปรแกรมในการทดสอบเหมือนเดิม

จากภาษาของคอมพิวเตอร์ที่เลือกใช้ ซึ่งคือตัว Visual Basic 6.0 นั้น ไม่ได้ถูกออกแบบเพื่อจุดประสงค์ทางด้านคณิตศาสตร์โดยตรง จึงไม่มีเครื่องมือสำหรับการคำนวณหาจุดตัดของกราฟได้ ดังนั้นในการคำนวณหาจุดตัดของกราฟจึงจำเป็นต้องทำการเขียนโปรแกรมในส่วนนี้เพิ่มขึ้นมา

โปรแกรมภาษา Visual Basic 6.0 ไม่ได้ออกแบบเพื่อจุดประสงค์ในการใช้งานด้าน Multithreading ดังนั้นถ้าต้องการพัฒนาต่อไปควรใช้โปรแกรมภาษาอื่นที่รองรับการทำงาน Multithreading เช่น Java , C

5.2 แนวทางการพัฒนาต่อ

จากปัญหาที่พบในเครื่องมือที่พัฒนาขึ้นนี้เราอาจมีการพัฒนาต่อให้เครื่องมือทำงานในโครงข่ายข้อมูลใดๆก็ได้ อาจมีการเปลี่ยนซอฟแวร์ที่ใช้พัฒนาให้มีความเหมาะสมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

การวิเคราะห์และออกแบบด้วย UML , สัลยุทธ์ สว่างวรรณ

การเขียนโปรแกรมด้วย Visual Basic 6.0 ฉบับโปรแกรมเมอร์, จำลอง ครุอุตสาหะ

การเขียนโปรแกรมด้วย Visual Basic 6.0 ฉบับฐานข้อมูล, จำลอง ครุอุตสาหะ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้